ECONOMIC SIGNIFICANCE AND EFFICIENCY OF CONTROL SYSTEM DESIGN OF MULTIROTOR UAV BY USING PYTHON OPENCV

Volodya BARSEGHYAN
Ph.D. in Economics, NPUA, Faculty of Engineering Economics and Management
Nerses NERSISYAN
PhD student, NPUA Faculty of Control systems and their elements
Avetik BASKOVCHYAN
PhD student at EUA Faculty of Information Technologies, “Automation systems”
Tariel SIMONYAN
Master student, NPUA Faculty of Control systems and their elements
Yerevan, Armenia

Keywords: quadcopter, controller, machine vision, object, image, recognition

Introduction. Multirotor drones are an integral part of the development of the modern world. This article presents a UAV control model using the Python programming language, which is based on object recognition using the OpenCV [C.B.Kadua, C.Y.Patilb,2017, 737-746] library and, according to the obtained results, determining the direction of movement of the quadrupole by the system itself. The work was performed according to the parameters of the DJI TELLO quadcopter and the effective operation of the algorithm was demonstrated in real use.

In order to correctly obtain the direction and speed of the multi-rotor UAV, in our case, the four-propeller movement, 3 parameters are checked: roll, pitch, yaw - Figure 1, the control performed by proportional–integral–derivative (PID) through the regulator.

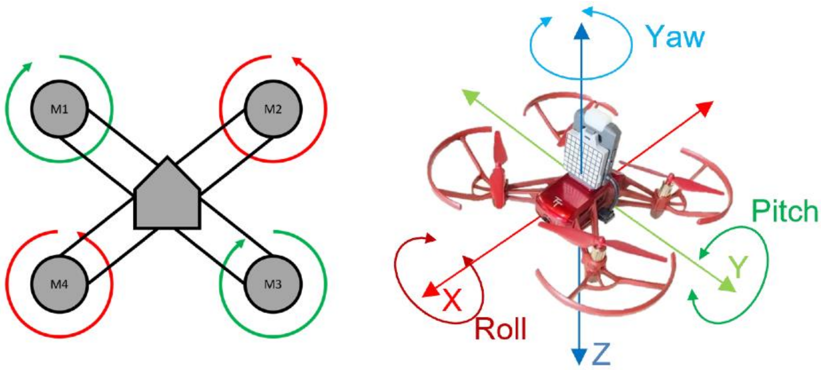

Figure 1 . roll,pitch,yaw angels for Drone

As can be seen from Figure 1, in order to correctly adjust the movement of the quadrotor, it is necessary to ensure the correct rotation with respect to the corresponding axis by correctly adjusting the rotation speed of the motors, the values of which the quadrotor takes from the output of the controller.

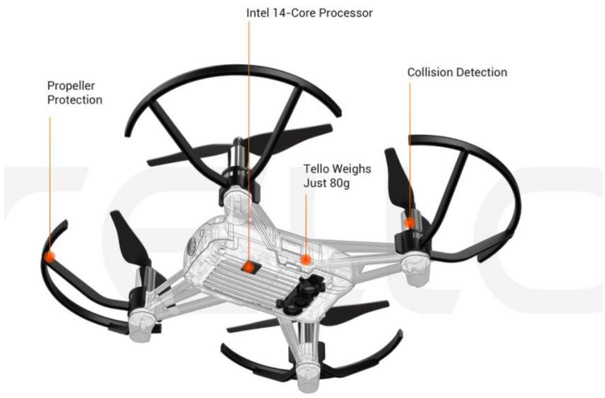

Figure 2. DJI Tello Quadcopter

Figure 2 shows the general view of the Tello quadcopter. As can be seen from Figure 2, the quadcopter has a front camera, through which it becomes possible to use the OpenCV library for object recognition and coordinate transfer.

Methodology. This article presents a UAV control model in real conditions, based on the DJI Tello drone, for decision making by comparing the results obtained through real-time sensor values. The correct UAV control model and the correct use of the values obtained from the sensors are of great economic importance, the model presented in this article can be applied in various fields to perform analytical analyzes of image recognition and recognition results. The clear results of real-time work are presented in Figure 10. The used methodology for UAV control was chosen by using the values obtained from the drone's camera to recognize the object through the PID controller, and performing analytical calculations by moving in the direction of the recognized object, the results of which have been widely used in many fields: urban planning, agriculture, etc. A feedback system such as a PID controller includes a feedback control system. This system evaluates a feedback variable using a fixed point to generate an error signal. Based on this, it changes the output of the system. This procedure will continue until the error reaches zero, otherwise the value of the feedback variable becomes equivalent to a fixed point. For object detection the program takes images from camera and transferring to array. It is containing number of pixels and creating binary image.

Literature review. The results presented in the paper are widespread and sufficiently used in other works. The model used in agriculture [Rejeb, Abdolahi, 2022,372-398] is significantly cheaper than the model presented in this paper, considering the low cost of the DJI TELLO drone at $90. In the model presented by [Pathak, Zang, 2022, 40-48], a manual control model of the UAV is used for crop research, in our model, the control is performed automatically. To create UAV control model [L. Salih, K. Gaeid, 2010, 3660-3667] with PID controller was used Matlab Simulink package. PI and PID controllers created.

Scientific novelty The control model introduced in this article is an automated system corresponding to the parameters of a real drone, which detects, recognizes and collects the necessary information about the target object without intervention. The presented model is adapted to a cheap UAV**.**

PID controller As already mentioned, the DJI TELLO quadcopter was used as the main UAV during the work, the control system used includes the PID controller, Figure 5 shows the main block diagram describing the operation of the PID system:

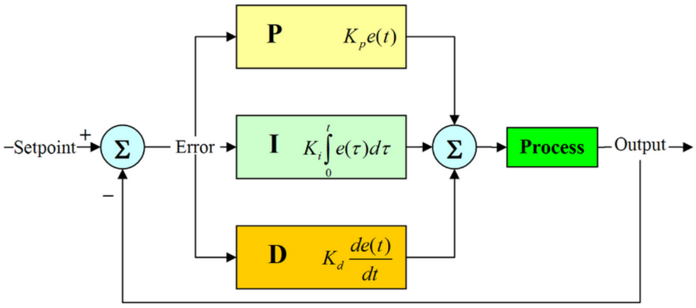

Figure 5. Block diagram of PID controller

The PID controller shown in Fig. 5 in Python for the Tello quadcopter looks like Fig. 6:

```
for i in range(0,ns):
    e[i] = sp[i] - pv[i]
    if i >= 1:  # calculate starting on second cycle
        dpv[i] = (pv[i]-pv[i-1])/delta_t
        ie[i] = ie[i-1] + e[i] * delta_t
    P[i] = Kc * e[i]
    I[i] = Kc/tauI * ie[i]
    D[i] = - Kc * tauD * dpv[i]
    op[i] = op[0] + P[i] + I[i] + D[i]
    if op[i] > op_hi:  # check upper limit
        op[i] = op_hi
        ie[i] = ie[i] - e[i] * delta_t # anti-reset windup
    if op[i] < op_lo:  # check lower limit
        op[i] = op_lo
        ie[i] = ie[i] - e[i] * delta_t # anti-reset windup
    y = odeint(process,pv[i],[0,delta_t],args=(op[i],Kp,taup))
    pv[i+1] = y[-1]
op[ns] = op[ns-1]
ie[ns] = ie[ns-1]
P[ns] = P[ns-1]
I[ns] = I[ns-1]
D[ns] = D[ns-1]
```

Figure 6. Representation of the PID controller in Python language

Figure 6 shows the representation of the PID controller in the Python language, as a result of which the corresponding parameters of the PID controller are automatically calculated, assigned to the variables P, I, D. The variables P,I,D are given as input values to the quadrupole. Calculation formulas are presented below (1).

$$P = Kc * e$$
$$I = Kc/tauI * Ie \qquad (1)$$
$$D = Kc * tauD * dpv$$

Where: P,I,D are respectively the searchable values of Proportional, Integral, Differential variables, *Kc* proportional coefficient, *tauI* integrating component coefficient, *tauD* differentiating component coefficient, *e* error, *Ie* error integral, *dpv* process variable differential.

Object recognition using the OpenCV library The OpenCV library makes it possible to implement object recognition and tracking using the Python language [Culjak, 2018]. The code implemented is shown in figure 7:

```
def findObjects(img,imgContour):
    global dir
    contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        areaMin = cv2.getTrackbarPos("Area", "Parameters")
        if area > areaMin:
            cv2.drawContours(imgContour, cnt, -1, (255, 0, 255), 7)
            peri = cv2.arcLength(cnt, True)
            approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
            #print(len(approx))
            x , y , w, h = cv2.boundingRect(approx)
            cx = int(x + (w / 2))  # CENTER X OF THE OBJECT
            cy = int(y + (h / 2))  # CENTER X OF THE OBJECT

            if (cx <int(frameWidth/2)-deadZone):
                cv2.putText(imgContour, " GO LEFT " , (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(0,int(frameHeight/2-deadZone)),(int(frameWidth/2) \
                                            -deadZone,int(frameHeight/2)+deadZone),(0,0,255),cv2.FILLED)
                dir = 1
            elif (cx > int(frameWidth / 2) + deadZone):
                cv2.putText(imgContour, " GO RIGHT ", (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2+deadZone),int(frameHeight/2-deadZone)), \
                        (frameWidth,int(frameHeight/2)+deadZone),(0,0,255),cv2.FILLED)
                dir = 2
            elif (cy < int(frameHeight / 2) - deadZone):
                cv2.putText(imgContour, " GO UP ", (20, 50), cv2.FONT_HERSHEY_COMPLEX,1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2-deadZone),0),(int(frameWidth/2+deadZone), \
                                                  int(frameHeight/2)-deadZone),(0,0,255),cv2.FILLED)
                dir = 3
            elif (cy > int(frameHeight / 2) + deadZone):
                cv2.putText(imgContour, " GO DOWN ", (20, 50), cv2.FONT_HERSHEY_COMPLEX, 1,(0, 0, 255), 3)
                cv2.rectangle(imgContour,(int(frameWidth/2-deadZone), \
                int(frameHeight/2)+deadZone),(int(frameWidth/2+deadZone),frameHeight),(0,0,255),cv2.FILLED)
                dir = 4
            else: dir=0
```

Figure 7. Presentation of Object Recognition Algorithm in Python Language

As can be seen from Figure 7, as a result of the work of the FindObject function, the searchable object is detected and presented on the coordinate plane through the “scale” and “imgArray” variables. Figure 8 shows the implementation of the function of the algorithm for determining the speed and direction of movement of the quadrotor using the PID controller based on the corresponding coordinates obtained as a result of the detection of the found object using the Python language.

```
def trackObjcet(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
```

Figure 8. The algorithm for determining the direction and speed of the quadrotor

The block diagram of the designed feedback system [F. Ahmad Khan, Sh. Nisar, 2018,16-24] is shown in Figure 9 below.

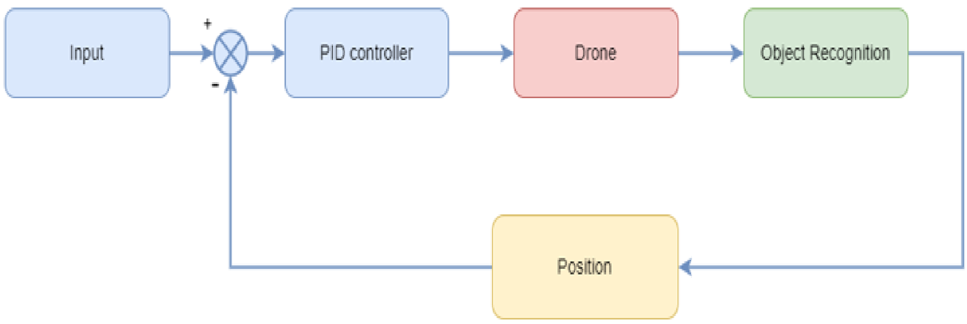

Figure 9. General block diagram of quadrotor control

The general block diagram shows the following blocks:
- Desired distance
- PID controller

• Object recognition from which the available distance is obtained

Results obtained. The results of the development and testing of the DJI TELLO quadcopter control system using the Python language and in real conditions are presented below. Figure 10 shows the operation of the object recognition algorithm, which provides as a result the values of the input variables for the operation of the code: “center” and “Area”.

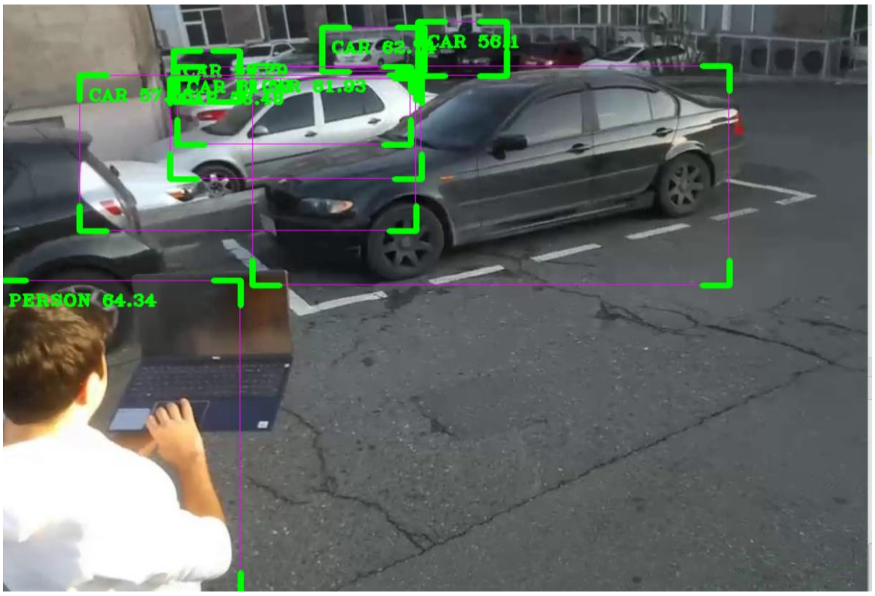

Figure 10. The result of the work of the object recognition algorithm

Figure 11 shows the output characteristic of the PID controller compared to the ideal input value.

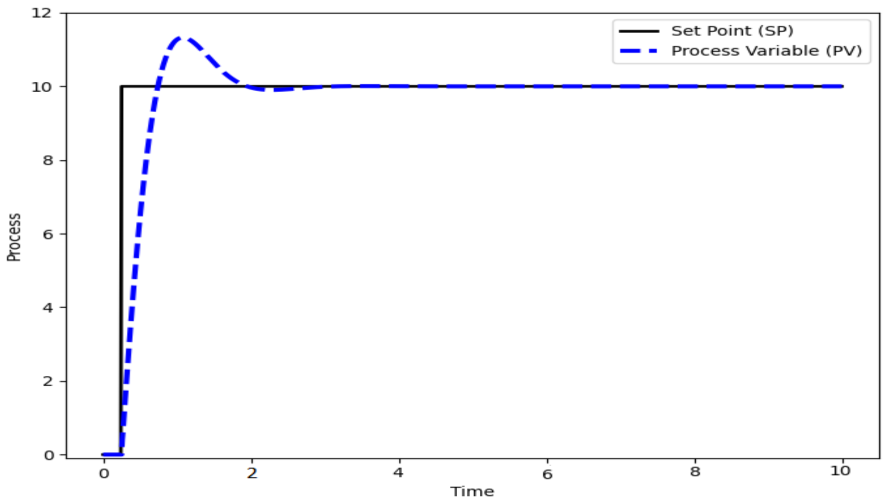

Figure 11. The transient process graph of the system

As can be seen from Figure 11, the system settling time is 1.74s and has about 1.3% overshoot. Such results demonstrate the high level of efficiency of the system. Figure 12 shows the graphic description of the individual parameters of the used HID controller during the flight.

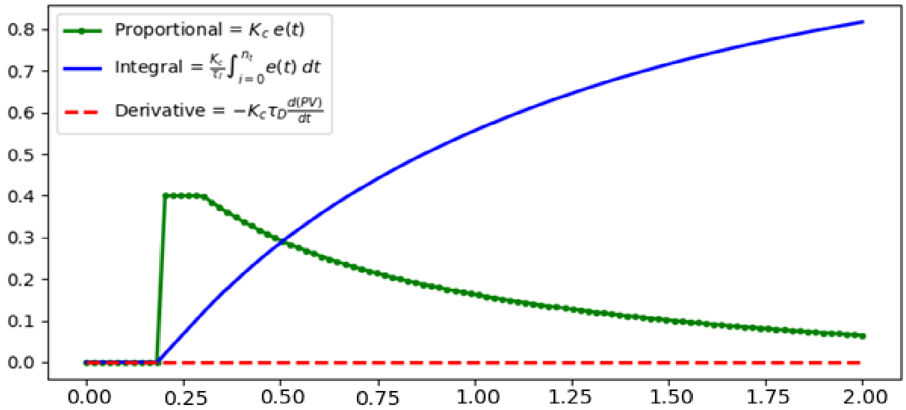

Figure 12. Operation of individual components of the PID controller

From the graphs shown in Figure 12, it becomes clear that during the test, the P and I parameters of the P,I,D parameters of the HID controller were activated, which was sufficient to ensure the stable operation of the four-propeller. The calculation formulas of the parameters are shown in Figure 12

Conclusion The work was done in accordance with the parameters of the real model of the quadcopter, through an algorithmic solution of OpenCV library object recognition. As a result, a mathematical description of the system equipped with an PID controller was shown using the Python language, which adjusts the position and the corresponding speed of the quadcopter in a short time interval to maintain the distance from the recognized object. The obtained results are sufficient for use in the field of agriculture any other economical processes.

References

1. C.B.Kadua, C.Y.Patilb "Design and Implementation of Stable PID Controller for Interacting Level Control System", 7th International Conference on Communication, Computing and Virtualization 2016, pp 737-746, 2016
2. I.Culjak, D. Abram, T. Pribanic, etc, "A brief introduction to OpenCV", 2012 Proceedings of the 35th International Convention MIPRO, pp 26-31, January 2012
3. S. Ch. De Silva, M. Phlernjai, S. Rianmora , "Inverted Docking Station: A Conceptual Design for a Battery-Swapping Platform for Quadrotor UAVs" , pp 55-71,February 2022
4. https://m.dji.com/product/tello
5. Chandan G, A. Jain, Harsh Jain, Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV", 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp 1305-1309, July 2018

6. F. Ahmad Khan, Sh. Nisar, "Design and analysis of feedback control system", 2018 International Conference on Information and Communications Technology, p 16-24, March 2018
7. H. Pathak et al., "A review of unmanned aerial vehicle-based methods for plant stand count evaluation in row crops", CEA,Volume 198, pp 40-48, July 2022, 107064
8. Ab. Rejeb, Al. Abdollahi, K. Rejeb, H. Treiblmaier "Drones in agriculture: A review and bibliometric analysis", Computers and Electronics in Agriculture,Volume 198, p 372, July 2022,
9. M V Subramanyam, P V, K Satyaprasad, G. krishna " Study on PID Controller Design and Performance Based on Tuning Techniques", International Conference on Control, Instrumentation, Communication, Computational Technologies At: NI University, Nagercoil. July 2014
10. A.Salih, M. Moghavvemi, and others "Flight PID controller design for a UAV quadrotor", Scientific Research and Essays Vol. 5(23), pp. 3660-3667, 4 December, 2010

Volodya BARSEGHYAN, Nerses NERSISYAN, Avetik BASKOVCHYAN, Tariel SIMONYAN

Economic significance and efficiency of control system design of Multirotor UAV by using Python OpenCV

Key words: quadcopter, controller, machine vision, object, image, recognition

This article presents the implementation of a control system for a multirotor unmanned aerial vehicle with a controller and an object recognition system based on Python programming language. The DJI Tello drone was used as the main drone to get accurate real-world values and see the operation behavior in practice. Multirotor drones are an integral part of the development of the modern world. This article presents a control model for an unmanned aerial vehicle using the Python programming language, which is based on object recognition using the OpenCV library and, based on the results, determining the trajectory of the quadrupole by the system itself. The work was carried out in accordance with the parameters of the DJI TELLO drone and the efficient operation of the algorithm in real conditions was demonstrated. To get the direction and speed of a multi-rotor unmanned aerial vehicle, in our case a four-rotor, 3 parameters are checked: roll, pitch, yaw, the latter is controlled using a comparative, integrator-differentiating controller.