Journal of Robotics and Control (JRC)
Volume 3, Issue 2, March 2022
ISSN: 2715-5072 DOI: 10.18196/jrc.v3i2.14180

Mini Drone Linear and Nonlinear Controller System Design and Analyzing

Esraa H. Kadhim [1], Ahmad T. Abdulsadda [2]
[1,2] Department of Communication Engineering, Engineering Technical Collage / Al-Najaf, Al-Furat Al-Awsat Technical University (ATU), Najaf, Iraq
E-mail: [1] esraa.kadhim.ms.ectn@student.atu.edu.iq, [2] coj.abdulsad@atu.edu.iq

***Abstract*—Choosing the mini-drone for a specific payload for designing purposes is one of the most challenging for both cost and design purposes. It is important to develop and analyze the flight control systems of the quadcopter-type Parrot mini drone and how to make the drones more tolerant of adverse weather conditions. The main problem with any quadcopter is that it loses its balance when exposed to any external influence, even if that influence is weak. Where the controller is the most important part of the drone, six plane controllers cover the six degrees of freedom (6dof) in the movement of the drone. In our research, we have improved the height controller in the drone, thus improving the altitude controller by using (PD) and increasing the values of (Kp and Kd) in the altitude controller of the Parrot Mini Drone Mambo to make it more bearable to external influence and to maintain its altitude. We assumed that the aircraft was exposed to bad weather conditions, such as snowfall and dust, which led to an increase in the speed at which the drone fell. We also increased the free fall constant of the object in the simulation design of the drone from (-9.81 m/s^2 to -12.81 m/s^2) and used Matlab R2021a Simulink to undertake the tuning of the (Kp and Kd) values. This study yielded good results, as illustrated in the results section. Therefore, this research paper suggests adopting the PD controller in the altitude controller and the new values of Kp and Kd to make the drone more tolerant of weather conditions. We tested these results in practice and got good results.**

Keywords—PID controller; quadcopter; Matlab-Simulink; Altitude controller.

I. INTRODUCTION

Recently, UAVs, particularly quadcopters, have elicited the attention of people all over the world, including researchers, students, and technology enthusiasts or hobbyists. Responding to this extraordinary popularity, researchers have created a plethora of novel control algorithms, ranging from the model-based controller to the model-free controller, to effectively and efficiently control the quadcopter system.

A large number of users have used this design to study the characteristics and components of drones and to develop and use them in different fields. The most famous of these researches are: Several deep learning architectures were used in this paper to identify the quadcopter UAV system. Overall, the CNN-LSTM model has been found to outperform all other architectures, with average tested MSE and MAE values of 0.0002 and 0.0030, respectively [1],[2]. A paper proposed a UAV-based smart healthcare scheme for COVID-19 monitoring, cleansing, social distancing, data study, and statistics group in the control area. The frame collects information via wearable sensors, drive sensors deployed in battered areas, or thermal appearance processing [3],[4].

In the other paper, the observer (linear parameter-varying (LPV)) was deployed on a Parrot® Rolling Spider mini drone, and a series of flying tests were performed to evaluate the (Fault Detection and Diagnosis (FDD)) competencies in real-time using the onboard processing power. Flight tests validated the simulation results and demonstrated that the sliding style observer could provide reliable fault rebuilding for quadrotor mini-drone organizations [5]. The modified adaptive sliding approach algorithm was developed in the other paper using a version law based on the Lyapunov strength approach, which allowed the controller's nonlinear adaptive performance to compensate for disturbances and parameter perturbations. Matlab simulations are used to validate the utility of the suggested regulator strategy in comparison to the old approach [6]. The sensors, such as ultrasonic and barometric pressure sensors, as well as their data, played a vital part in calculating the altitude of the Parrot Mambo micro drone in the other study. Utilized Simulink software and block sets like the Simulink support package for Parrot micro drones to keep the drone at a constant height. Apart from the hardware and software descriptions, the drone's equipment, capabilities, and performance have also been discussed [7].

The Vortex Ring State (VRS) and Windmill-Brake State (WBS) have been examined in the context of quadcopters in the other work. Following that, wind tunnel tests were used to develop a quadcopter model that is independent of the floppy load and blade disk sizes. A basic model was then developed for trajectory de-signs. Thereafter, the GPOPS-II program was used as an arithmetical solver to construct optimum 2D and 3D descent trajectories due to the challenging optimum issue aimed at minimal time path design. Finally, conducted flying tests were conducted to demonstrate that the VRS is current in quadcopters. Further claimed that the flight fluxes might stand decreased by raising the plane speed of the blade floppy.

As an ideal falling trajectory, a helix-type trajectory is used [8]. Low-cost instruments, such as a 10-DOF Mems

(Micro-electro-mechanical systems), IMU (Inertial Measurement Unit), and a LIDAR (light detection and ranging) were fitted on a minor unmanned rotorcraft in other research and synchronized at a 10-Hz measurement rate to estimate the location of the platform and its space from a hitch or a landing field. Kalman filtering was used to correct the IMU data for systematic errors (bias) and dimension noise, as well as to obtain predicted locations from the accelerometer data. The technique was created on an aboard microprocessor (Arduino Mega 2560), and it enables low-cost hardware applications of many sensors for usage in aerospace requests [7].

The other study looks at a proportional and derivative PD controller that uses a quadrotor UAV to regulate the adjustment of the quadrotor UAV while in flight. To be stable and have high performance, the PD controller's gain parameters, the proportional gain Kp, and the derivative gain Kd is used. Unmanned aerial vehicles (UAVs) are becoming more popular, and they come in a wide variety of sizes and designs. The quadcopter settles the time of roll, pitch, and yaw system after incorporating PD controllers into the systems. After the research, the simulation results and a comparison of X, Y, and Yaw control approaches are shown. Plemented. The optimum estimate technique, which was built on an aboard microprocessor (Arduino Mega 2560), enables low-cost hare operations of many sensors for usage in a variety of requests [10], [11].

In other studies, the controller has been tweaked to handle the tracking trajectory problem. The primary idea behind this control system is to allow the robot to trace the target trajectory with the least amount of error possible. The robustness and effectiveness of the created method, as well as the responsiveness of the suggested sliding mode controller, are demonstrated using simulation results produced using MATLAB software [10], [12]–[14]. In [15]–[18], the outcomes of the autonomous swarming flights in the open air are discussed. The designed mini-drone is small in size, with a wheelbase of 130 mm and a mass of 76 g, and it comes equipped with all of the sensors required for autonomous flying. The suggested controller compensates for nonlinearity in dynamics, allowing for accurate velocity control. Furthermore, the results of the swarming flight tests revealed that the produced mini-drones and the suggested controller perform flawlessly under real-world flying circumstances. Another author found the results impressive: using the audio signal's Mel-frequency cepstral coefficients (MFCCs) and various support vector machine (SVM) classifiers, it was possible to achieve a minimum classification accuracy of 98% in the detection of the specific payload class carried by the drone with an acquisition time of only 0.25 s; the performance improved when longer acquisition times were used [19]. The key references that the author used are [20]–[23]. Moreover, some studies created an embedded system for a quadrotor UAV flight controller. The controller was built with readily available low-cost components, open hardware design, and open software, allowing users to test and implement new control algorithms, which distinguish it from the most prevalent alternatives on the market. A sensing system was created for taking and recording the quadrotor's odometer. An architecture for sending angular velocity instructions to the motors through the PWM was designed, and everything was processed everything on a Raspberry Pi 3 [24-26].

Many research studies focus on improving the design of the control system in drones because these aircraft reach dangerous places that humans cannot reach. During their flight, they are exposed to different and dangerous weather conditions. In addition, the drone system under actuated is challenging to control.

In this paper, we will improve the control system for determining the position and altitude of the aircraft (PD) by making the aircraft maintain its stability even after exposure to bad weather conditions such as falling dust or snow on it.

II. MATERIALS AND METHOD

A. *Material*

In this research paper, we use a mini drone called Parrot mini drone-Mabo (Fig. 1). The Mambo is controlled by a computer running the PyParrot interface through a Wi-Fi or Bluetooth connection. A built Simulink model is utilized to simulate the desired flight route for this study. This program enables simulated runs with various parameters to identify the Parrot mini drone's intended response. This is performed by controlling the Simulink model's numerous subsystems [4].

Fig. 1. Parrot mini-drone fly

Maintaining control of a UAV is necessary for a variety of reasons. UAVs must have fewer independent control inputs than grades of freedom, which causes a controller difficulty when tiresome to retain control of wholly six degrees of freedom. This opens up the possibility of including design elements to regulate the axes, as well as yaw, pitch, and roll. Fig. 2 shows a simple block chart of the needed inputs and wanted outputs that a controller will require to successfully manage a UAV.

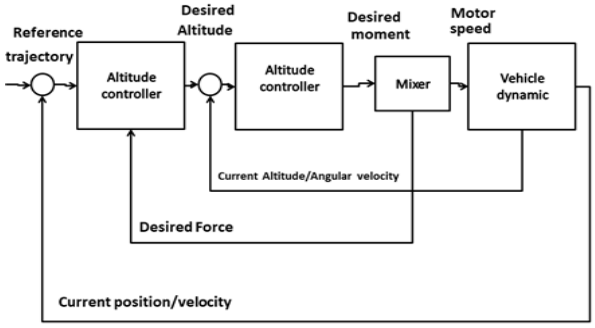

Fig. 2. Quadrotor control system design.

The focus of this study is on the Parrot mini drone-Mabo control implementation. Simulink is used to create the programmed controller based on a Parrot model. The block diagram for the procedure for each flight alteration made by

the Parrot mini drone Mabo is shown in Fig. 3. Two control rings, an external loop, and an internal loop flow continually into apiece throughout the system. The system inputs are the location reference, estimated yaw, yaw reference, and altitude reference. The Simulink simulation's state estimator is divided hooked on numerous filter blocks. A complementary filter and a Kalman filter remain employed [4].

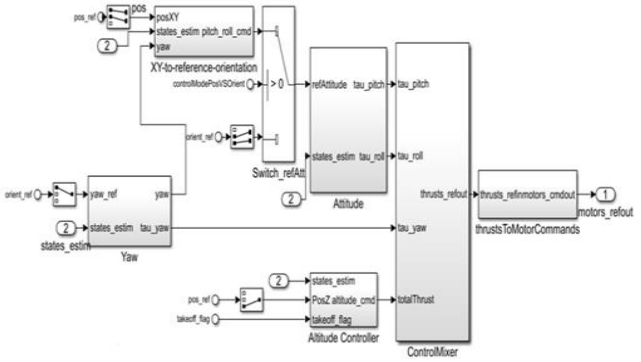

Fig. 3. controller subsystem [11]

To identify inaccuracies, it compares the reference signals generated by the path planning algorithm to the estimated states. These are fed into the PID controllers, which generate the commands for the actuators. The signals are then sent to the pitch/roll (or attitude) internal loop controller by the X-Y position outer loop controller. There is also a yaw controller and a height controller that work independently of these controllers. A total of six PID controllers control the position and attitude of the micro drone. Fig. 4 and Fig. 5 illustrate how to set up the altitude controller as PID. In this approach, the proportional gain is multiplied by the altitude error generated from the sonar sensor, while the derivative gain is multiplied by the rate of altitude of the gyroscope, which is a less noisy signal than the ultrasound signals. It is important to note that the z-axis in the coordinate system of the drone points downwards, which means the altitude value in the control system will always have a negative sign in front of it (expressed in meters) [2].

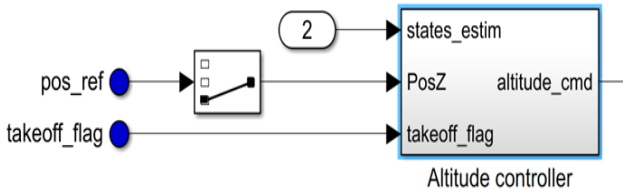

Fig. 4. Altitude controller block [27].

B. The Mathematical Model of PID (Proportional Integrated and Derivative) and PD (Proportional and Integrated) controller

1) PD controller

It is a series controller, proportional and derivative controller. If we assume that we have the system shown in Fig. 6, the PD controller is connected in series with the system.

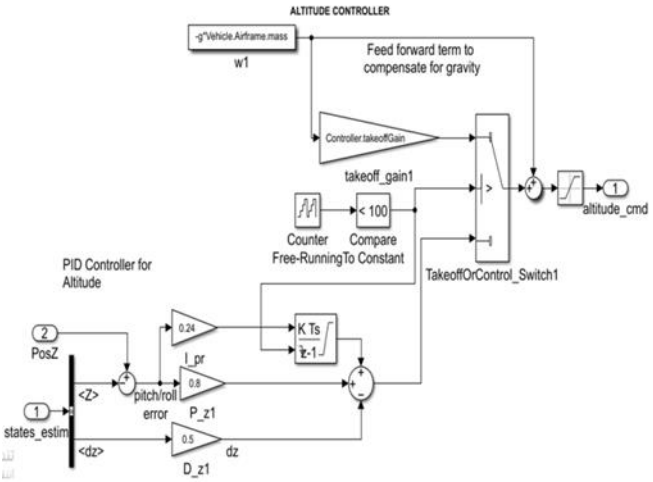

Fig. 5. Altitude controller structure [11]

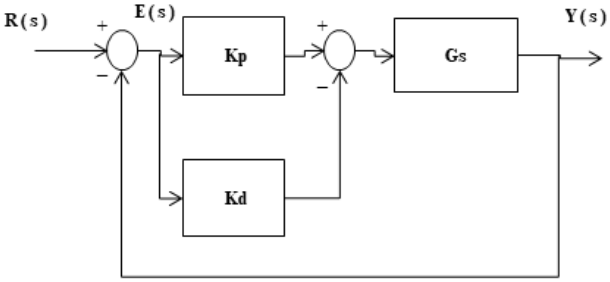

Fig. 6. PD Controller block diagram.

$$oversG(s) = \frac{w_n^2}{s^{2+2\zeta w_{ns}}} \quad (1)$$

Where G(s) is the transfer function of the system, ζ is the damping ratio, and wn is the natural frequency.

$$Gc(s)K_{p+}K_{d^s} \quad (2)$$

Where Gc(s) is the transfer function of the controller, KP and KD are constant values (gain).

$$GT(s) = Gc(s)G(s) = \frac{w_{nKds+}^2 w_n^2 Kp}{s^2 + 2\zeta w_{n^s}} \quad (3)$$

Where GT(s) is the total transfer function.

$$\begin{aligned} \frac{Y(s)}{U(s)} &= \frac{Gc(s)G(s)}{1 + Gc(s)G(s)} \\ &= \frac{w_n^2 Kds + w_n^2 Kp}{s^2 + (2\zeta w_n + w_n^2 Kd)s + w_n^2 Kp} \\ &= \frac{w_n^2}{s^2 + 2\zeta wns + w_n^2} \end{aligned} \quad (4)$$

Where the Y(s) is the output signal and the U(s) is the input signal.

To control the damping coefficient and natural frequency, three variables (Kp, Kd, Ki) were selected in three equations to allow for full control over the system [28].

2) PID Controller

It is a cascade controller, proportional, integrated, and derivative controller. If we assume that we have the system shown in Fig. 7, the PID controller is connected in series with the system.

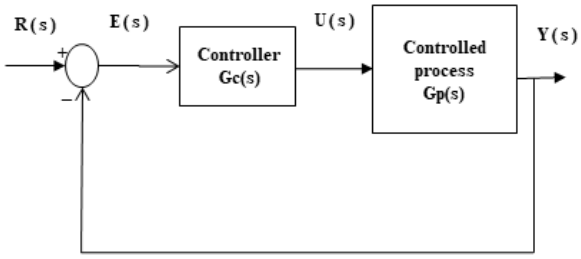

Fig. 7 PID Controller block diagram.

$$G(s) = \frac{w_n^2}{s^{2+2\zeta w_{ns}}} \quad (5)$$

$$Gc(s) = K_{p+}K_{d^s} + \frac{Ki}{s} \quad (6)$$

$$GT = Gc(s)G(s) = \frac{w_{nKds^2+}^2 w_n^2 Kp\, s + w_n^2 Ki}{s^3 + 2\zeta w_{n^s}\, s^2} \quad (7)$$

$$\frac{Y(s)}{U(s)} = \frac{Gc(s)G(s)}{1 + Gc(s)G(s)} = \frac{w_n^2 Kds^2 + w_n^2 Kps + w_n^2 Ki}{s^3 + (2\zeta w_n + w_n^2 Kd)s^2 + w_n^2 Kp} = \frac{w_n^2}{s^3 + (a + 2\zeta wn)s^2 + (2a\zeta w_n + w_n^2)s + aw_n^2} \quad (8)$$

To control the damping coefficient and natural frequency. Three variables are to be selected (Kp, Kd, Ki) in three equations. Thus, we have full control over the system [28].

C. Methods

In this paper, we developed and improved the altitude control structure shown in Fig. 4 and Fig. 5. These PID controllers contain force and torque commands as outputs, which are then communicated to the mix motor algorithm (MMA) (Fig. 3), which generates the required motor thrusts and converts the orders into motor speeds.

We replaced the PID controller with a type PI controller and then a type controller PD in the first step. We found the best of the three types in system stability, ease of design, and cost reduction. The controller was simplified, as shown in Fig. 8. The "auto-tuning" method found the value of Kp and Kd were found by the "auto-tuning" method.

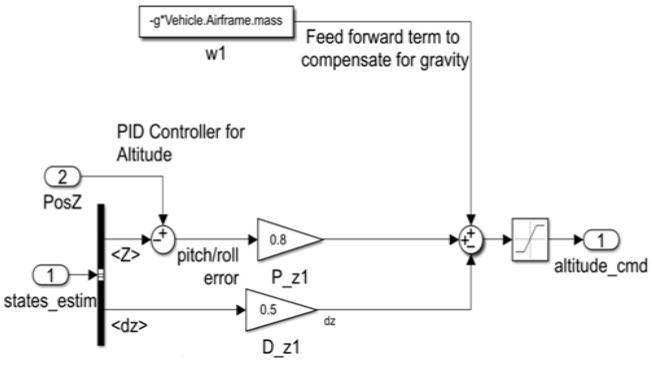

Fig. 8. Altitude controller [12].

The second method to increase the efficiency of the system was to apply a disturbance such as dust or snow on the vehicle, thereby increasing the value of the block named (g*vehicle*mass) representing the mass of the vehicle. In doing so, the system became unstable, which led us to tune the value of Kp and Kd until we could improve performance through trial and error. We eventually obtained a good result as the system was able to remain stable even as it was affected by the bad weather. Fig. 9 summarizes the steps of the work that was undertaken.

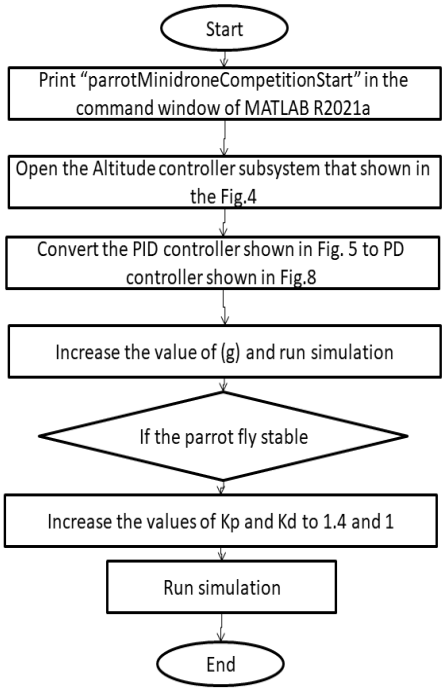

Fig. 9. Workflow algorithm

Tuning the PD controller: a linear model is required to tune the controllers since nonlinear models, notwithstanding their simulation accuracy, are not ideal for controller design. The height controller will be tuned in this article using Simulink's 'PD tuner' tool. The controller simply indicates the height to climb or drop using a positive or negative command. The linearized controller model used for tuning is shown in Fig. 10 [29].

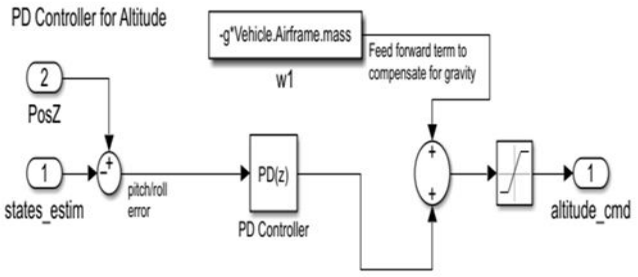

Fig. 10. Tuning is done via a Simplified Altitude Controller.

By opening the PD block, the 'autotuner' is launched. It linearizes the control loop. The program then shows the linearized version's closed-loop response, allowing you to tweak the system's reaction time and transient behavior (see Fig. 11).

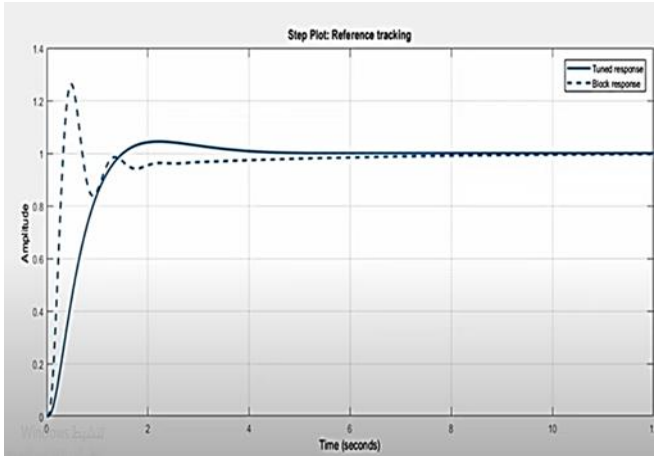

Fig. 11. PID Tuner App on Simulink

Due to the elimination of nonlinear components, the dashed line of the response signal does not have the same simulation behavior as the solid line, but it is still useful for tweaking purposes. Following gain selection and brief hardware testing, it became clear that the hardware does not behave as planned as it is unable to take off correctly. In this scenario, the issue has an impact on the feedforward term. If it is too low, the algorithm assumes that the drone's weight is lower than it actually is or that the thrust is greater than it actually is. As a result of the reduced proportional route, the controller has more trouble controlling the remaining weight, and it is unable to lift off. The drone can eventually take off if the value is increased by approximately 25% [30].

III. RESULTS AND DISCUSSION

A. *RESULTS*

In the reference conditions when the drone flies in suitable weather, we obtained the following results (Fig. 12):

Kp = 0.8, Kd = 0.3, ζ=0.707, wn= 190 Hz

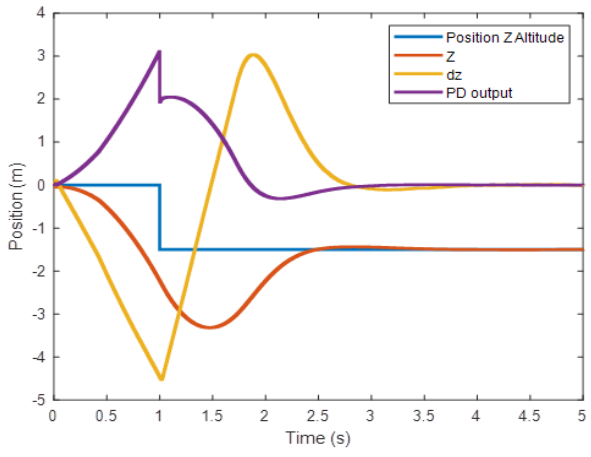

Fig. 12. Reference results of PD controller

We increased the drone's weight to impose the falling of dust or snow due to bad weather conditions by changing the value in the block (-g*vehicle. airframe. mass). This is the plane's weight multiplied by the body's free fall constant (g = 9.81 m/s^2). The negative sign indicates that the body is increasing.

We increased the value of the constant (g) to -12.81 to impose an increase in the weight of the aircraft. This resulted in the system becoming unstable, and we attained the results shown in Fig. 13.

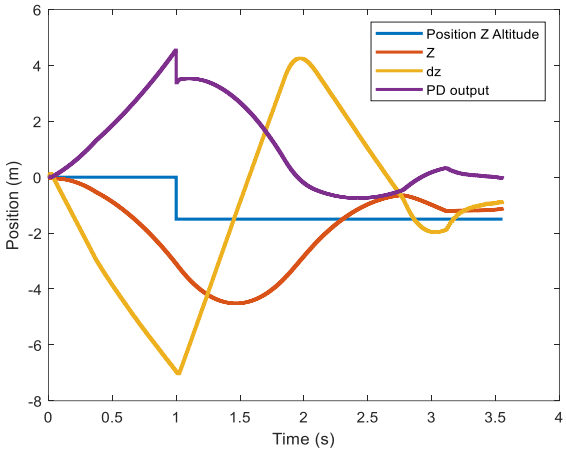

Fig. 13. As a result of the disturbance, the drone falls down after 3.5 seconds of flying.

To improve the system's response, we changed the values of (Kp) and (Kd) via the 'autotuner' method. The system responded the best when (Kp) was 1.4 and (Kd) was 1 (see Fig. 14 and Fig. 15). We can summarize the results as shown in Table I.

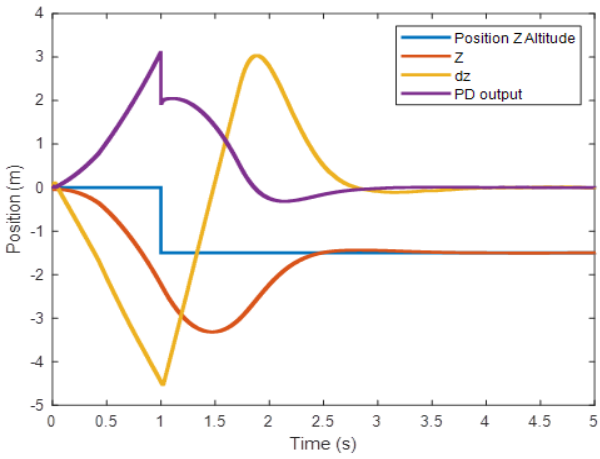

Fig. 14. The new result with disturbance after converting the value of Kp and Kd.

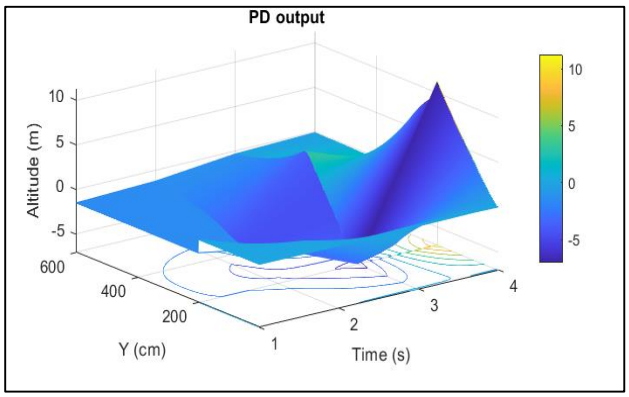

Fig. 15. Illustrate the end result

TABLE I. EXPLAIN THE RESULT

	Type of control-ler	Kp	Ki	Kd	System response	-g*vehicle.airframe.mass
1	PID	0.8	0.24	0.5	Stable	-9.81* vehicle.airframe.mass
2	PD	0.8	0	0.3	Stable	-9.81* vehicle.airframe.mass
3	PD	0.8	0	0.3	Unstable	-12.81* vehicle.airframe.mass
4	PD	1.4	0	1	Stable	-12.81* vehicle.airframe.mass

B. DISCUSSION

Unmanned aerial vehicles (UAVs) may be a valuable asset in search and release missions. However, to realize their full potential, all parameters that can touch the flight of UAVs must be properly accounted for, such as the excellence of sensory operations (which can vary depending on the location of the UAVs). Most previous studies in the literature focused on the parts of the controllers in drones, especially the altitude controller. Many researchers, as explained in the introduction chapter, employed many modern and complex techniques. In this research, we used a parrot mini drone in our experiment, where we tested the performance of the plane using PD, and after subjecting the drone to some disturbance and changing the values (Kp and Kd) by auto-tuning in the simulation, we obtained good results by applying that in the MATLAB simulation. Fig. 12 shows the altitude of aircraft Z and the estimation altitude dZ as well as the output of PD, where it can be noted that the drone is flying at a fixed altitude until the end of the specified implementation time, but when we introduce disturbance in the altitude controller (increase g to -12.81), the drone falls down after 3.5 seconds of flying, as shown in Fig. 13. Fig.14 presents the results obtained after a change of values. Kp and Kd are exactly identical to the original results before the disturbance was added. Therefore, we suggest changing the values of the altitude controller to the new values to make the drone more stable in bad weather.

IV. CONCLUSIONS

Our modification of the PD controller shows that our results can be enhanced. This research uses a dynamic model of a quadcopter-type Parrot mini drone Mabo to construct a durable cascade PD control technique. The key benefit of the cascade PD control scheme is its high tolerance to external disturbances. In addition, the efficacy of the developed controller was demonstrated by comparing conventional and cascade PID to PD control systems. To summarize, the cascade PD control approach gives the quadcopter system a significant performance gain. The focus of future research will be to build on the other controllers in the dynamic system of the quadcopter so that the quadcopter system's resilience and performance against parameter uncertainty and external disturbances may be increased.

REFERENCE

[1] B. P. Amiruddin, E. Iskandar, A. Fatoni, and A. Santoso, "Deep Learning based System Identification of Quadcopter Unmanned Aerial Vehicle," *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2021, pp. 165-169.

[2] P. Ceppi, "Model-based Design of a Line-tracking Algorithm for a Low-cost Mini Drone through Vision-based Control," *Diss. University of Illinois at Chicago*, 2020.

[3] A. Kumar, K. Sharma, H. Singh, and S. Gupta, "A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 1–19, 2021, doi: 10.1016/j.future.2020.08.046.

[4] S. DeBock "Guidance, Navigation, And Control of a Quadrotor Drone with PID Controls," Diss. Monterey, CA; Naval Postgraduate School, 2020.

[5] S. Waitman, H. Alwi, and C. Edwards, "Flight evaluation of simultaneous actuator/sensor fault reconstruction on a quadrotor minidrone," *IET Control Theory Appl.*, vol. 15, no. 16, pp. 2095–2110, 2021, doi: 10.1049/cth2.12180.

[6] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Mini-drone quadrotor altitude control using characteristic ratio assignment PD tuning approach," *Lect. Notes Eng. Comput. Sci.*, vol. 2019, pp. 337–341, 2019.

[7] C. C. Veedhi, "Estimation of altitude using ultrasonic and pressure sensors," *Thesis of Blekinge Institute of Technology*, 2020.

[8] A. Talaeizadeh, D. Antunes, H. N. Pishkenari, and A. Alasty, "Optimal-time quadcopter descent trajectories avoiding the vortex ring and autorotation states," *Mechatronics*, vol. 68, p. 102362, 2020, doi: 10.1016/j.mechatronics.2020.102362.

[9] P. J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The Navigation and Control technology inside the AR.Drone micro UAV," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 1477–1484, 2011, doi: 10.3182/20110828-6-IT-1002.02327.

[10] A. Shamshirgaran, H. Javidi and D. Simon, "Evolutionary Algorithms for Multi-Objective optimization of Drone Controller Parameters," *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 1049-1055, doi: 10.1109/CCTA48906.2021.9658828.

[11] W. M. Thet, M. Myint, and E. E. Khin, "Modelling and Control of Quadrotor Control System using MATLAB/Simulink," *Int. J. Sci. Eng. Appl.*, vol. 7, no. 7, pp. 125–129, 2018, doi: 10.7753/ijsea0707.1002.

[12] B. Alkhlidi, A. T. Abdulsadda, and A. Al Bakri, "Optimal robotic path planning using intelligent search algorithms," *J. Robot. Control*, vol. 2, no. 6, pp. 519–526, 2021, doi: 10.18196/jrc.26132.

[13] C. Ben Jabeur and H. Seddik, "Optimized Neural Networks-PID Controller with Wind Rejection Strategy for a Quad-Rotor," *J. Robot. Control*, vol. 3, no. 1, pp. 62–72, 2022, doi: 10.18196/jrc.v3i1.11660.

[14] A. E. M. Redha, R. B. Abduljabbar, and M. S. Naghmash, "Drone Altitude Control Using Proportional Integral Derivative Technique and Recycled Carbon Fiber Structure," *Lecture Notes in Networks and Systems*, pp. 55–67, Aug. 2021.

[15] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build Your Own Quadrotor: Open-Source Projects on Unmanned Aerial Vehicles," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 33–45, 2012, doi: 10.1109/MRA.2012.2205629.

[16] D. Lee and D. H. Shim, "Design and Validation of Low-cost Flight Control Computer for Multi-rotor UAVs," *J. Korean Soc. Aeronaut. Sp. Sci.*, vol. 45, no. 5, pp. 401–408, 2017, doi: 10.5139/jksas.2017.45.5.401.

[17] D. Lee, H. Lee, J. Lee, and D. H. Shim, "Design, implementation, and flight tests of a feedback linearization controller for multirotor UAVs," *Int. J. Aeronaut. Sp. Sci.*, vol. 18, no. 4, pp. 740–756, 2017, doi: 10.5139/IJASS.2017.18.4.740.

[18] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2992–2997, 2011, doi: 10.1109/ICRA.2011.5980229.

[19] O. A. Ibrahim, S. Sciancalepore, and R. Di Pietro, "Noise2Weight: On detecting payload weight from drones acoustic emissions," Future Generation Computer Systems, Apr. 2022.

[20] Z. Uddin, M. Altaf, M. Bilal, L. Nkenyereye, and A. K. Bashir, "Amateur Drones Detection: A machine learning approach utilizing the acoustic signals in the presence of strong interference," *Comput. Commun.*, vol. 154, pp. 236–245, 2020, doi: 10.1016/j.comcom.2020.02.065.

[21] S. Sciancalepore, O. A. Ibrahim, G. Oligeri, and R. Di Pietro, "Detecting drones status via encrypted traffic analysis," *Proceedings of the ACM Workshop on Wireless Security and Machine Learning - WiseML 2019*, pp. 67–72, 2019, doi: 10.1145/3324921.3328791.

[22] U. Seidaliyeva, M. Alduraibi, L. Ilipbayeva, and A. Almagambetov, "Detection of loaded and unloaded UAV using deep neural network," *Proc. - 4th IEEE Int. Conf. Robot. Comput. IRC 2020*, pp. 490–494, 2020, doi: 10.1109/IRC.2020.00093.

[23] I. Djurek, A. Petosic, S. Grubesa, and M. Suhanek, "Analysis of a Quadcopter's Acoustic Signature in Different Flight Regimes," *IEEE*

Access, vol. 8, pp. 10662–10670, 2020, doi: 10.1109/ACCESS.2020.2965177.

[24] S. Madruga, A. Tavares, A. Brito and T. Nascimento, "A Project of an Embedded Control System for Autonomous Quadrotor UAVs," *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, 2018, pp. 483-489, doi: 10.1109/LARS/SBR/WRE.2018.00091.

[25] S. Bouabdallah, A. Noth and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, pp. 2451-2456, vol. 3, doi: 10.1109/IROS.2004.1389776.

[26] A. Benini, A. Mancini, and S. Longhi, "An IMU/UWB/vision-based extended kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network," *J. Intell. Robot. Syst. Theory Appl.*, vol. 70, no. 1–4, pp. 461–476, 2013, doi: 10.1007/s10846-012-9742-1.

[27] "Simulink Support Package for Parrot Minidrones - File Exchange - MATLAB Central." https://ch.mathworks.com/matlabcentral/fileexchange/63318-simulink-support-package-for-parrot-minidrones (accessed Feb. 19, 2022).

[28] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall, 1991.

[29] D. Xue, *Modeling and Simulation with Simulink®: For Engineering and Information Systems*, Walter de Gruyter GmbH & Co KG, 2022.

[30] C. S. Veerappan, P. K. K. Loh, and R. J. Chennattu, "Smart Drone Controller Framework—Toward an Internet of Drones," Studies in Computational Intelligence, pp. 1–14, 2022.