Automated System of the Variable-Pitch Propeller Rotation Control

Mikola Yeremiyev, Alexander Karatanov, Dmitriy Kritskiy and Olha Pohudina

July 22, 2019

AUTOMATED SYSTEM OF THE VARIABLE-PITCH PROPELLER ROTATION CONTROL

Yeremiyev M.B., Karatanov O.V., Kritsky D.N., Pogudina O.K.
National Aerospace University «KhAI»
Kharkiv, Ukraine
d.krickiy@khai.edu

Abstract: *The different variants of the implementation of the proportional-integral-derivative controllers are considered in the article, the results of the experiment carried out on the model for the P-controller, the PD-controller, the PID-controller and the non-controller system used for automating the operation of an unmanned aerial vehicle; it has been proved that the use of the PID-controller that processes the data from the interference sensor, capable of maintaining a constant value of engine speed when changing the angle of the screw, also experimentally the coefficients for each component of the controller were found.*

Keywords: *PID, multicopter, arduino, preservation of a given number of engine revolutions, variable-pitch propeller.*

INTRODUCTION

Formulation of the problem. There is a wide range of problems in the development of small unmanned aerial vehicles (SUAV), which use the convertiplane scheme. It raises the question of waiting for a long time for the internal combustion engine (ICE) to trigger signals so in this case it is impossible to stabilize SUAV.

The next question to be solved is the maintenance of the required lift during the change of the helicopter scheme to the airplane scheme and vice versa. One of the solutions for this problem is to use the variable-pitch propeller, which allows not to reduce or add gas to the engine, but to change the step of the screw pitch. Thus, the engine turn must be supported constantly, that requires the development of the PID-regulator.

Analysis of recent research and publications. Despite the new findings in management theory that have been received by researchers in recent years all around the world, PID-controllers are still the most widely used regulators in the industry. The reason for such high popularity is the simplicity of the construction and industrial use, clarity of functioning, suitability for solving most practical problems and low cost [1, 2].

Thus, in [3-5], PID-regulators were researched for the angle stabilization of a multicopter, stabilization in space and height relatively to the sea level. Also, in these works, the methods of implementation and configuration of the PID-controllers are considered.

In [6], the development and optimization of the PID-controller is used to provide the required characteristics of the transient process of the ICE and to ensure that the actual rotational speed of the engine meets the desired speed in steady state. The mathematical calculation of the most optimal PID-control coefficients and the influence of the coefficients on the work of the system was demonstrated.

The article [7] describes the operation and implementation of a discrete PID-controller. It presents the equations of the regulator in the finite difference, which were taken as the basis for the development of the required controller.

The aim of the study. Development of the automated system to maintain specified engine speed when using an alternating step screw.

MAIN PART

A proportional–integral–derivative controller (PID-controller or three-term controller) is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value *e(t)* as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name [8].

The block diagram of the system with the PID-controller is shown in Fig. 1.

The scheme contains the following notation:

- *x(t)* – sets the effect, in this case – given turns that comes to the receiver;
- *y(t)* – real turns;
- *e(t)* – deviation (or error) – is calculated as the difference between the given action and the feedback signal;
- *u(t)* – controlling influence – the result of the work of the PID-controller;
- δ – delta function – control signal transformed with a drive unit;
- *g* – perturbation – the result of the step changing of the screw;
- $\otimes$ – adder – a device that adds input signals (on the first adder the lower part is painted because the input feedback signal is negative);

- the block marked as "▶" is a proportional link that only multiplies the deviation e(t) by the coefficient;
- the block marked as "∫" – the integrator – accumulates the error signal and multiplies by the coefficient;
- the block marked as "$\frac{d}{dt}$" – differentiator – multiplies the difference of the output signal with the output signal at the previous moment.

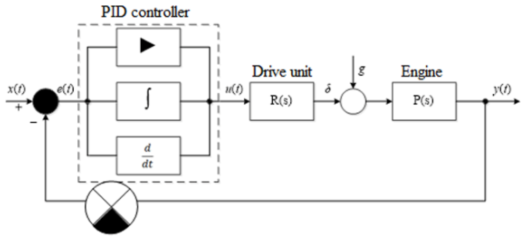

Fig. 1. System block diagram

In the illustrated scheme, the differential element operates only with the feedback signal from the device. This signal is subtracted from the control signal, and the resulting difference is considered to be a control error. This difference is applied to the input of proportional and integral elements. The signals, received at theiк outputs, are added and form a control signal.

The PID uses three main management methods, which are explained below.

The proportional or P-controller gives an output that is proportional to the current error *e(t)*. It compares the scheduled (set) value with the actual value (or the value of the feedback process). The resulting error is multiplied by a proportional constant to obtain the value of the control signal. In the case if the value of the error is zero, the output signal of the controller is also zero.

Such a controller requires a manual reset, as it never reaches a steady state. It provides stable work, but accumulates the error. The reaction rate increases with the increase of the proportional constant k_p (Fig. 2).

Due to the limitations of this control method, where there is always a shift between the process variable and a given value, an I-controller is required, which provides the necessary actions to resolve the error of the steady state. It integrates an error during the time until the error value reaches zero and contains a value for the control device in which the error becomes zero.

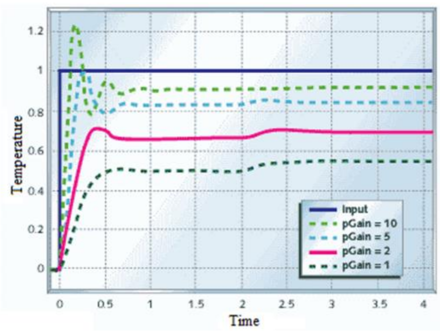

Fig. 2. The P-controller behavior chart, where pGain is the k_p value

Integral control reduces the output control signal when a negative error occurs. It limits the speed of the reaction and affects the stability of the system. The reaction rate is increased by reducing the integral gain k_i (Fig. 3).

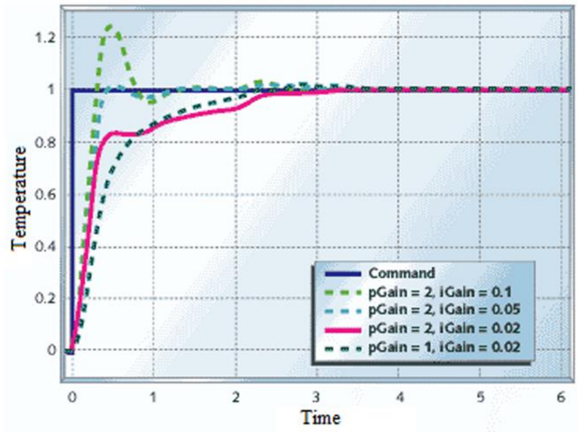

Fig. 3. The PI-controller behavior chart, where iGain is the k_i value

From the picture above, it can be noticed that with decreasing the I-controller gain, the error decreases. In most cases, PI control is used when a high-speed response is required.

When using the PI-controller, the output of the I controller is limited to some range to overcome the integral conditions, when the integral output increases even at zero error conditions because of non-linearity.

The I-controller cannot predict the future behavior of the error. D-controller overcomes this problem by "predicting" information about the future behavior of the system. Its output depends on the rate of change in error over time, multiplied by a constant derivative, which increases the system response [9].

Fig. 4 shows how the system responds to the PID-control. As can be seen from the graphs, this approach significantly improves the quality of system management compared to the PI-controller.

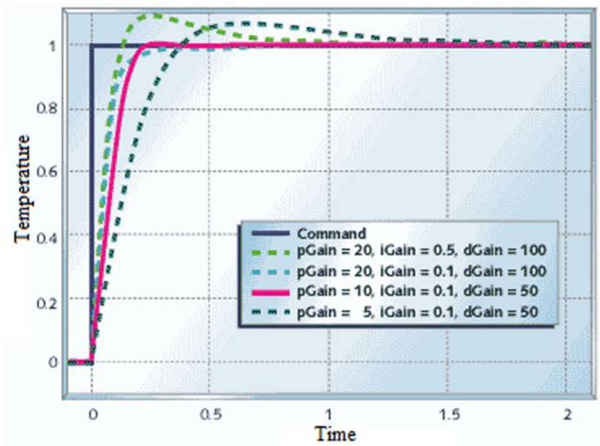

Fig. 4. The behavior of the PID-controller, where dGain is the k_d value

Adding a component of the D-controller can increase the stability of the system by compensating the phase delay caused by the I-controller. Increasing the differential control factor increases the reaction rate.

From the last graph, it is noticeable that by combining these three components, the desired response for the system can be got. Different manufacturers are developing a variety of PID-controllers [10].

The above information should be considered when designing own controller.

Another important step in the creation of an automated control system for variable-pitch propeller rotation is to build a layout for the controller development and testing.

The test bench consists of the following elements:

- 3D model for fixing on a plane, considered earlier;
- electric motor;
- set with screw of variable step;
- infrared obstacle sensor YL-63;
- Arduino Mega 2560 board;
- ZIPPY battery;
- Turnigy 9X transmitter and receiver – for manual change of speed and tilt angle of propeller blades;
- micro-USB cable;
- electronic controller;
- clamp for the model mounting.

A model consisting of two fastened parts, pre-designed in the SolidWorks system and printed on a 3D printer (Fig. 5-6), was developed for fixing the engine on a flat surface.

The obstacle sensor should be described separately, as during the test there were new difficulties associated with it, and the performance of the entire system depends on it.

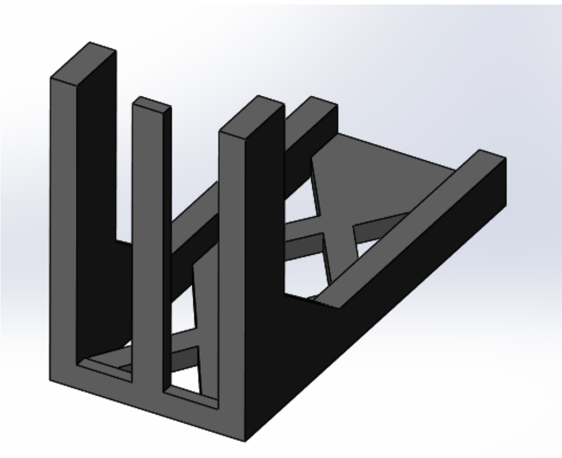

Fig. 5. The first part for fixing to the plane

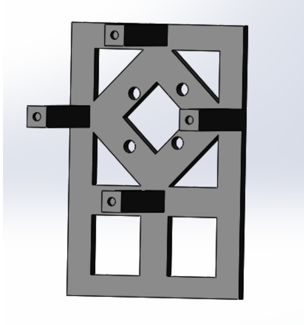

Fig. 6. The second part for mounting the engine with the variable-pitch propeller

The contactless sensor YL-63 detects objects in the range of distances from almost zero and to the set limit without coming into contact with them. Different manufacturers assign different names to the same device. Some of them calls the sensor as YL-63 other FC-51. The sensor is designed for use when information about the distance to an object is not needed, but only about its presence or absence. The registration deadline depends on the setting. The YL-63 sensor has a discrete output. This optical sensor records the intensity increase of reflected infrared radiation in a controlled space. The change in reflected radiation occurs because of moving parts of the mechanisms or the movement of surrounding objects.

The device contains a source of infrared radiation and a photodetector. The radiation reflects the obstacles and is recorded by the photodetector. It transmits a signal to the comparator LM393, which is configured to operate at a certain level of photodetector illumination. The comparator generates a signal at the output of the YL-63 sensor of a low or high logic level.

The optical sensor YL-63 belongs to the diffusing class. The name of the group of sensors arose due to the reflection of the radiation on the sets of directions – the radiation diffusion by the reflecting surface.

The device is used to determine the photodetector illumination. Since the YL-63 captures reflected radiation, there is an error of the distance measurement, caused by different reflecting abilities of object surfaces made of various materials.

The scheme, that describes the connection of the elements with the board, is shown in Fig. 7.

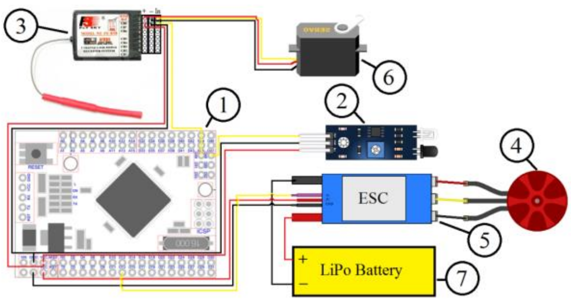

Fig. 7. Connection scheme of elements with Arduino board:
1 – Arduino Mega 2560 board, 2 – obstacle sensor, 3 – receiver, 4 – electric motor, 5 – electronic speed controller, 6 – servo drive, 7 – battery

The resulting test bench with the designation of the constituent elements is shown in Fig. 8.

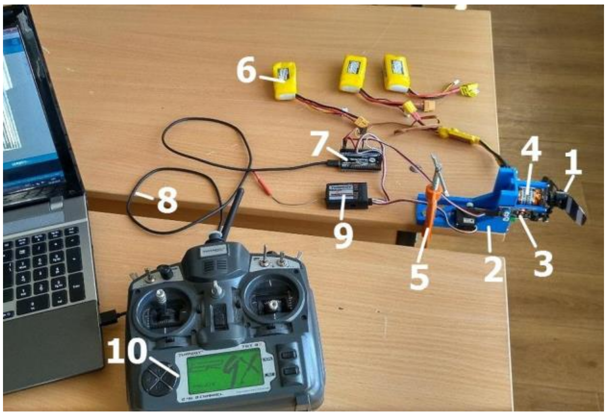

Fig. 8. Ready-to-work test bench: 1 – variable-pitch propeller, 2 – printed 3D-model, 3 – obstacle sensor, 4 – electric motor, 5 – clamp, 6 – battery, 7 – Arduino board, 8 – micro-USB cable, 9 – receiver, 10 – transmitter

The sensor model described above has shown poor results during the testing stage. The reason is that the YL-63 is very sensitive to noise, and gives false testimony if the additional sources of lighting appear. Therefore, it was decided to replace it with the KY-032 model.

KY-032 allows to detect an object without using a mechanical contact with it, which makes it possible to determine the presence of an object at a certain distance. With KY-032 you can set the absence or presence of obstacles at a specified distance or closer. It is not possible to determine exactly the distance to the obstacle. Another application of the module is the rev counter or speed meter. In the case of cyclic displacement, the module is used to determine the speed and position of the moving part. The KY-032 sensor reacts to reflected infrared radiation in a controlled area. To measure the speed of rotation of the element it should be painted in white and black stripes. The radiation of the sensor is directed to the alternating strips. When rotation from the output of the module will receive pulses, the frequency of which describes the speed of rotation. It is used in automatic systems, including using microcontrollers, for example, for mobile robots.

The module includes two semiconductor photovoltaic devices: an infrared radiation LED and a photodetector. The radiation of the LED is reflected with the obstacle and comes to the photodetector. The radiation of the KY-032 sensor is modulated at 38 kHz for the photodetector, which is great for fulfilling the task.

Next, after constructing the test bench, there is a program implementation of the PID-controller.

To program the microcontroller the C/C++ programming language and the Arduino development environment was used.

The program that performs the controller function consists of four parts, each of which is executed in a separate stream. Figure 9 shows a block diagram of the program algorithm.

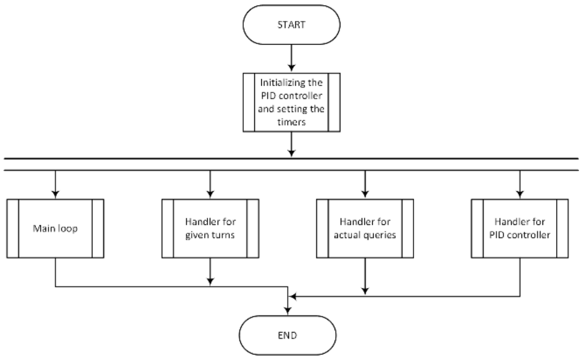

Fig. 9. Block diagram of the program algorithm

Fig. 9 shows that the first thread executes the main program cycle. It displays the values of the given and actual speed of rotation for plotting, as well as provides new values of engine speeds, which corresponds to the calculated control signal.

In the second thread, the timer interrupt function is executed to obtain the specified turn speed. The value, entered on the input, is read and filtered, taking into account the previous values of the formula:

$$y_i = y_{i-1} + \frac{x_i - y_{i-1}}{Tu}, \quad (1)$$

where $Tu = 2n$;

y_{i-1} – the speed value obtained on the previous iteration;

x_i – the input value.

Also, in this handler there is a check of the signal front and inverting the timer flag, which is responsible for this check. It is necessary for the handler to work on both fronts.

In the third thread, processing of the timer interruption is performed to obtain real values of the number of revolutions. The same actions as in the second thread are executed, except the ones related to the front of the signal, since in this case the handler should operate only one front which is determined at the beginning.

In the fourth stream a handler is executed for the PID-controller, which operates at a frequency of 100 Hz.

The given revolutions come at the input in the range from 0 to 1000, and the information on actual turns is got in the form of a period in the range of 6000 to 1000 units. In this regard, the value of real revolutions must be calculated with the formula:

$$V_R = \frac{65535}{V_{get}/80},$$

where V_{get} is the actual speed value.

After bringing to one range, the calculation of inconsistency is performed as the difference between the given and real speed values:

$$e = V - V_R,$$

Further, the received error value is taken into account in the calculation of the following:

$$P(t) = k_p \cdot e(t),$$

$$I(t) = I(t-1) + k_i \cdot e(t),$$

$$D(t) = k_d \cdot (e(t) - e(t-1)),$$

where k_p, k_i i k_d are the coefficients of P, I and D regulation.

Signal of control influence is calculated by the formula:

$$u = V + P + I + D,$$

As soon as the output signal was calculated, it must be translated into the range of values with which the electronic controller is used (in this case, from 1000 to 1400 units):

$$\omega = \frac{u}{3} + 1000,$$

where ω is the speed of rotation to be transmitted to the engine in the main loop.

The PID controller parameters adjustment (coefficients k_p, k_i and k_d).

At first it is necessary to set the coefficients k_i and k_d to zero. The coefficient of the proportional component must be set to an initial value equal to "1". If the system goes very slowly to the line, then k_p must be increased, if it begins to oscillate, then k_p must be reduced. Starting with a small k_p value, it should be increased in 10 times until the fluctuation begins. Then the value of the coefficient is reduced, but not in 10 times, but in 2. And so on, until the fluctuation is not stopped. This describes searches for the desired value, initially using large steps.

The value of the coefficient of integral component k_i must be a small relative to k_p. It is recommended to take a number from 0.0001 to 0.01 as the initial value of k_i. The procedure for finding the coefficient k_i is exactly the same as the proportional coefficient (initially large steps, and then small). The excessive coefficient k_i also manifests itself in the appearance of oscillations.

To find k_d coefficient, you must set its initial value to 0. Next, the value of the proportional k_p ratio must be set to small (for example, k_p = 1). It is important to choose such value of k_p that the system at zero k_d does not oscillate. Next, you need to set a small initial value of k_d (for example, k_d = 0,1).

The k_d factor should be increased until false oscillations are caused by small noise. In this case, the oscillations from the too large coefficient occur faster than the oscillations from the insufficient-exact coefficient. It is recommended to set a factor of half or a quarter from the one at which fluctuations start. The main thing in this process is to ensure in time that the system behavior is adequate [11].

To set up the controller, a bench was created consisting of test equipment, the system and its communication with the computer. The input of the system was influenced in the form of a change in the given speed and step of the propeller, and the analysis of the response of the system was carried out. In the Fig. 8 the first thread executes the main program cycle. It displays the values of the given and actual speed of rotation for plotting, and also sets the new value of engine speeds, which corresponds to the calculated control signal.

The application testing was carried out during the setup of the PID-controller. This is due to the fact that the configuration requires a ready-to-use and properly running software implementation. Fig. 10 shows a graph of the system behavior without PID-control.

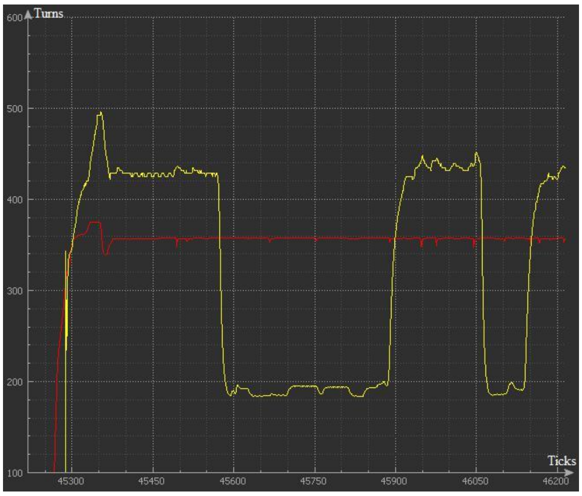

Fig. 10. Graph of the system behavior without the PID-control

On the graphic, the red color indicates the change of the given speed, and yellow – the actual turns. In the figure the leaps of the actual speed values can be noticed, which appear as a result of a change in the step of the screw: when the screw is loaded, the turns are falling, and when load relieves – sharply increasing.

Fig. 11 shows a graph that describes changes in system behavior as a result of adding the proportional control. The P-control coefficient k_p = 3.7 was chosen as the most optimal.

Fig. 11. The system behavior during operation of the P-controller

As can be seen from the figure above, real revolutions have become closer to the given values.

The result of adding a differential component can be seen in Fig. 12.

Fig. 12. The result of the PD-controller operation

For the coefficient of differential control, the maximum value that causes unnecessary hesitation was k_d = 15. The last stage is the inclusion of an integral component that should lead to the fact that the real values of revolutions will match the given. The most optimal value of the integral control coefficient was k_i = 0.017. The result of the PID-control is shown in Fig. 13.

From the figure above it is noticeable that the PID-controller copes with its task. Sharp leaps in the actual revolutions caused by the relief/load of the propeller have significantly decreased, compared with the system behavior without control and quickly come to the norm, thanks to PID-controller.

Fig. 13. Graph of the system behavior with PID-control

As a result, the following values of control coefficients were obtained:

- k_p = 3,7;
- k_i = 0,017;
- k_d = 15.

The obtained values are more effective, because the engine operation with the variable-pitch propeller rotation control shows more smooth transitions when changing the step.

Conclusion

In this article the basic principles of development of the automated variable-pitch propeller rotation control system are researched.

In the course of the study, the program code was developed in the C programming language for the Arduino board, which performs the function of the PID-controller, as well as the bench for testing and demonstrating the performance of the program. Additionally, the article discusses the controllers setting. The input of the system was influenced and, as a result of the response analysis, the coefficients for PID-control were selected.

As a result, after the completion of the work, the PID-controller was implemented, which supports the engine turns according to the given value. The results of the study will be useful when designing a multicopter with an internal combustion engine, which, at this stage of the development of the UAV, exists only in the form of prototype samples.

References

[1] Visioli, A. (2012). Research Trends for PID Controllers. *University of Brescia*, pp. 144-148. Available at: https://dspace.cvut.cz/bitstream/handle/10467/66989/1656-1488-1-PB.pdf.

[2] PID-regulators. Available at: http://www.bookasutp.ru/Chapter5_1.aspx.

[3] Gopalakrishnan, E. (2017). Quadcopter flight mechanics model and control algorithms. *Czech Technical University*, 69 p.

[4] Jiřinec, T. (2011) Stabilization and control of unmanned quadcopter. Master's thesis. *Czech technical university in Prague*, 105 p.

[5] Liang, O. (2018). Quadcopter pid explained. Available at: https://oscarliang.com/quadcopter-pid-explained-tuning/.

[6] Malkhede, D.N. Dhariwal, H.C., Joshi, M.C. (2010). On optimization of the pid governor for diesel engine, Mathematical Modelling and Analysis. *Indian Institute of Technology*, 135-150 pp. Available at: https://www.tandfonline.com/doi/pdf/10.1080/13926292.2002.9637186.

[7] PID regulators - for dummies practitioners (2012). Available at: http://we.easyelectronics.ru/Theory/pid-regulyatory--dlya-chaynikov-praktikov.html.

[8] Araki, M. (2012). PID control. *Control systems, robotics, and automation*, Vol. 2. 7 p. Available at: http://www.eolss.net/ebooks/Sample%20Chapters/C18/E6-43-03-03.pdf.

[9] Agarwal, T. (2017). The Working Principle of a PID Controller for Beginners. *Elprocus*, Available at: https://www.elprocus.com/the-working-of-a-pid-controller/.

[10] Wescott, T. (2016). PID Without a PhD. *Wescott Design* Services, 30 p. Available at: http://www.wescottdesign.com/articles/pid/pidWithoutAPhd.pdf.

[11] Karpov, V.E. (2012) Lax PID control. 34 p. Available at: http://robofob.ru/materials/articles/pages/Karpov_mobline1.pdf.

[12] Kritskiy, D., Karatanov, A., Koba, S., & Druzhinin, E. (2018). Increasing the reliability of drones due to the use of quaternions in motion. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT).

[13] Kritsky, D., Druzhinin, E., Pogudina, O., & Kritskaya, O. (2018). Decision Making by the Analysis of Project Risks Based on the FMEA Method. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT).

[14] Kritsky, D. N., Druzhinin, E. A., Pogudina, O. K., & Kritskaya, O. S. (2018). A Method for Assessing the Impact of Technical Risks on the Aerospace Product Development Projects. *Advances in Intelligent Systems and Computing Advances in Intelligent Systems and Computing III*.