Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 40 (2019) 156–169

Transportation Research Procedia

www.elsevier.com/locate/procedia

13th International Scientific Conference on Sustainable, Modern and Safe Transport (TRANSCOM 2019), High Tatras, Novy Smokovec – Grand Hotel Bellevue, Slovak Republic, May 29-31, 2019

Optimizing PID controller gains to model the performance of a quadcopter

Leszek Cedro[a,*], Krzysztof Wieczorkowski[a]

[a] *Department of Automation and Robotics, Kielce University of Technology, 25-314 Kielce al. Tysiącletnia Państwa Polskiego 7*

Abstract

This article deals with the quadcopter dynamics modelled by tuning the PID controller gains. The model was derived using Newtons's equations of motion. Wolfram Mathematica was employed to run simulations and compare different optimization methods for PID tuning.

Keywords: Quadcopter, PID controller, simulation, optimization method;

1. Introduction

Recent years have seen growing interest in the use of drones. This has coincided with the development of new technologies to design and control such systems. Drones that use more than two rotors to generate lift are referred to as multicopters. This article analyzes the dynamics of an unmanned aerial vehicle (UAV) with four rotors, i.e., quadcopter [1]. Appropriate controllers are required to control such a multidimensional system. Research in this area has included analyzing the use of linear quadratic regulators (LQR) [3], [10]. There are also studies on the use of sliding mode controllers [2], [8] and fuzzy logic controllers [14]. The most popular, however, are PID controllers [13]. Despite their common use in quadcopters [4], [5], it is still problematic to choose appropriate PID settings. One way to deal with this problem is to optimize the control criteria, e.g., error integral. Most studies on the subject have

* Corresponding author.
E-mail address: lcedro@tu.kielce.pl

10.1016/j.trpro.2019.07.026

been performed using computer programs such as MATLAB [4], [5], [6] or LabVIEW [9]. This article proposes employing Wolfram Mathematica to find the optimal PID settings for a quadcopter by minimizing the error integral. The quadcopter control system uses four PID controllers: one to control the altitude and three to control the attitude, i.e., the roll, pitch and yaw. The minimization functions available in Wolfram Mathematica are 'FindMinimum' and 'NMinimize'. The main difference between them lies in the fact that one searches for a local minimum and the other is responsible for finding a global minimum of a constrained function. The most popular algorithms implemented in Wolfram Mathematica are: 'PrincipalAxis', 'Automatic', and 'ConjugateGradient'. Constrained optimization methods include 'NelderMead', 'RandomSearch', 'SimulatedAnnealing', and 'DifferentialEvolution' [7]. Details of these optimization methods are provided in [15].

Nomenclature

x, y, z	position coordinates
φ, θ, ψ	roll, pitch and yaw about the respective coordinate axes
$\mathbf{R}_x$	rotation matrix for a rotation by angle φ about the X axis
$\mathbf{R}_y$	rotation matrix for a rotation by angle θ about the Y axis
$\mathbf{R}_z$	rotation matrix for a rotation by angle ψ about the Z axis
$\mathbf{\Omega}$	angular velocity matrix in the local coordinate system
$\mathbf{\Theta}$	angular velocity matrix in the auxiliary coordinate system
m	quadcopter mass
m_w	mass of propellers and rotors
$\mathbf{a}$	linear acceleration matrix
g	gravity accelelation
$\mathbf{F}$	vector of sum of thrust of all motors
$\mathbf{F}_{\mathrm{g}}$	vector of gravitational force
$\mathbf{F}_{\mathrm{Co}}$	Coriolis forces vector from rotations of the entire quadcopter
$\mathbf{F}_{\mathrm{Cw}}$	Coriolis forces vector from rotors rotations of the quadcopter
$\mathbf{v}$	linear velocity matrix
$\mathbf{\Omega}_{\mathrm{w}}$	sum of rotors's angular velocity matrix
$\mathbf{L}$	angular momentum vector
$\boldsymbol{\tau}$	vector of thrust moments
$\mathbf{I}$	matrix of inertia moments
J_x, J_y, J_z	moments of inertia about the respective coordinate axes
$\boldsymbol{\tau}_{\mathrm{gyro}}$	gyroscopic effect
l	distance between the i-th motor and the centre of gravity of the drone
T_i	thrust force of the i-th motor
M_i	thrust moment of the i-th motor
ω_i	angular velocity of the i-th motor
b	the gain coefficient
d	the gain coefficient
l	distance between the i-th motor and the centre of gravity of the drone
K_{Pi}	proportional gains
K_{Ii}	integral gains.
K_{Di}	derivative gains
ρ	penalty coefficient
U_{ri}	coefficient of the i-th steady-state signal

2. Mathematical model of the quadrocopter dynamics

The mathematical model was derived using Newton's equations of motion. Since there are no constraints for the analyzed unmanned aerial vehicle, i.e., the quadcopter (Fig. 1), it is assumed to have six degrees of freedom (DOF).

Fig. 1. Quadcopter.

The coordinates of the drone altitude and attitude are given in the form of vector **q** (1). The first three parameters correspond to the drone position in space and the other three refer to its rotations.

$$\mathbf{q} = \begin{bmatrix} x & y & z & \varphi & \theta & \psi \end{bmatrix} \tag{1}$$

The first step is the theoretical calculation of the angular velocity in the local coordinate system like based on [6]:

$$\mathbf{\Omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x \mathbf{R}_y \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{2}$$

which may be written in matrix form:

$$\mathbf{\Omega} = \mathbf{P}_1^2 \,\dot{\mathbf{\Theta}} \tag{3}$$

where:

$$\mathbf{P}_1^2 = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \tag{4}$$

The next step is to calculate the inverse of matrix $\mathbf{P}_1^2$

$$\mathbf{P}_2^1 = \left(\mathbf{P}_1^2\right)^{-1} = \begin{bmatrix} 1 & \sin\varphi\,\mathrm{tg}\theta & \cos\varphi\,\mathrm{tg}\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{bmatrix} \tag{5}$$

Finally, having already P_1^2 matrix, depending on the angular speed of the auxiliary coordinate system, Euler rates may be defined:

$$\dot{\mathbf{\Theta}} = \mathbf{P}_2^1\,\mathbf{\Omega} \tag{6}$$

Another issue is to define linear acceleration using Newton's second law of motion. The general form of the equation for linear acceleration may be written:

$$ma = \mathbf{F} - \mathbf{F}_g - \mathbf{F}_{Co} - \mathbf{F}_{Cw} \tag{7}$$

As may be seen, above equation consists of a constraint which is the force decreased by the gravitational force and the Coriolis forces. The force of gravity may be explained as projection of force from local coordinate system to the new local coordinate system. Gravity acceleration is multiplied by the total mass of quadcopter and rotation matrix P_1^2 (8).

$$\mathbf{F}_g = \mathbf{P}_2^1\,m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -mg\sin\theta \\ mg\sin\varphi\cos\theta \\ mg\cos\varphi\cos\theta \end{bmatrix} \tag{8}$$

The first of Coriolis forces results from the rotation of the entire quadcopter:

$$\mathbf{F}_{Co} = 2m\;\mathbf{\Omega}\times\mathbf{v} = 2m\begin{bmatrix} v_z\omega_y - v_y\omega_z \\ v_x\omega_z - v_z\omega_x \\ v_y\omega_x - v_x\omega_y \end{bmatrix} \tag{9}$$

while the second from the rotation of rotors:

$$\mathbf{F}_{Cw} = 2m_w\mathbf{\Omega}_w \times \mathbf{v} = 2m_w\begin{bmatrix} -v_y(\omega_1+\omega_2+\omega_3+\omega_4) \\ v_x(\omega_1+\omega_2+\omega_3+\omega_4) \\ 0 \end{bmatrix} \tag{10}$$

where $\mathbf{\Omega}_w$ may be explained:

$$\mathbf{\Omega}_w = \begin{bmatrix} 0 \\ 0 \\ \omega_1+\omega_2+\omega_3+\omega_4 \end{bmatrix} \tag{11}$$

The lift (thrust) **F**, being a sum of thrust of all motors , is generated vertically upwards along the drone's vertical axis Z; it is described by the following formula (12).

$$\mathbf{F} = \begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_1+T_2+T_3+T_4 \end{bmatrix} \tag{12}$$

Another steps is calculate the angular acceleration. Mathematical description for this effect is possible thanks to Euler equation of a rigid body motion. The general form of the Euler equation may be defined as:

$$\boldsymbol{\tau} = \frac{d\mathbf{L}}{dt} + \mathbf{\Omega} \times \mathbf{L} \tag{13}$$

For a rigid body is also rightly that:

$$\mathbf{L} = \mathbf{I}\mathbf{\Omega} \tag{14}$$

where **I** is described by matrix:

$$\mathbf{I} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{15}$$

Substituting then (13) to (14) (assuming the constancy of inertia moment) following equation may be obtained:

$$\boldsymbol{\tau} = \mathbf{I}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times (\mathbf{I}\boldsymbol{\Omega}) \tag{16}$$

However, due to the gyroscopic effect [11], [12] caused by the motors and rotors, equation (16) must be modified. Gyroscopic effect is described by the basic equation of the gyroscope:

$$\boldsymbol{\tau}_{gyro} = \boldsymbol{\Omega} \times \mathbf{L}_p \tag{17}$$

Angular momentum vector is formed by multiplying the moment of inertia of the rotor (I_w) by the rotor's angular velocity in the axis where the rotation occurs ($\boldsymbol{\Omega}_w$):

$$\mathbf{L}_p = I_w \boldsymbol{\Omega}_w = \begin{bmatrix} 0 \\ 0 \\ I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \end{bmatrix} \tag{18}$$

Knowing the $\mathbf{L}_p$ and $\boldsymbol{\Omega}$, equation (17) may be expanded:

$$\boldsymbol{\tau}_{gyro} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \end{bmatrix} = \begin{bmatrix} \omega_y I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ -\omega_x I_w(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ 0 \end{bmatrix} \tag{19}$$

With the final form of gyroscopic torques vector, Euler's equation for an object (13) may be upgraded and described by the following formula:

$$\boldsymbol{\tau} = \mathbf{I}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times (\mathbf{I}\boldsymbol{\Omega}) + \boldsymbol{\tau}_{gyro} \tag{20}$$

In this equation the sought quantity is the angular acceleration, so after transformations it may be written that:

$$\dot{\mathbf{\Omega}} = \mathbf{I}^{-1}(\boldsymbol{\tau} - \mathbf{\Omega} \times (\mathbf{I}\mathbf{\Omega}) - \boldsymbol{\tau}_{gyro}) \tag{21}$$

For the configuration coordinates responsible for the rotation by φ, θ and ψ, the moments $\boldsymbol{\tau}$ are expressed in the form of moments M_φ, M_θ and M_ψ, (22) – (24), respectively:

$$M_\varphi = -T_1 l + T_3 l \tag{22}$$

$$M_\theta = T_2 l - T_4 l \tag{23}$$

$$M_\psi = \sum_{i=1}^{4} M_i \tag{24}$$

These relationships are illustrated in Fig. 2.

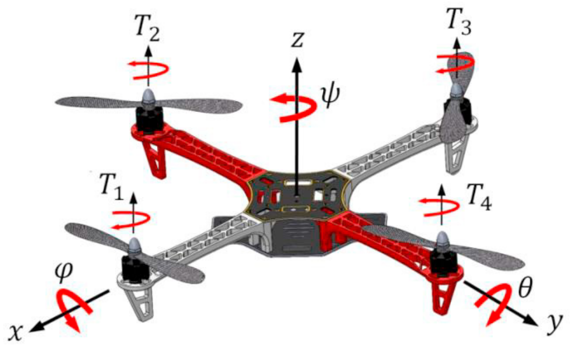

Fig. 2. Quadcopter's coordinate system.

From the above equations, it is clear that the effectiveness of the quadcopter control is affected by the thrust force generated along the Z axis F_z and the rotational moments M_φ, M_θ and M_ψ. These quantities are dependent on the square of the angular velocity of each motor. The thrust of the i-th motor T_i is calculated using the angular velocity of the i-th rotor as squared rotors angular velocities and the gain coefficient b:

$$T_i = b\omega_i^2 \tag{25}$$

The moment on the i-th motor shaft M_i can be calculated from the angular velocity of the i-th rotor (26)

$$M_i = d\omega_i^2 \tag{26}$$

After substituting (25) and (26) in Equations (22) – (24), we obtain

$$\boldsymbol{\tau} = \begin{bmatrix} M_\varphi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} lb\left(-\omega_1^2 + \omega_3^2\right) \\ lb\left(\omega_2^2 - \omega_4^2\right) \\ d\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{bmatrix} \tag{27}$$

Having the successive elements of model and forces one may to write a complete model of a flying object. The first part refers to a linear acceleration, which is defined in the local coordinate system:

$$\begin{aligned} \mathbf{a} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{b}{m}\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \end{bmatrix} - \begin{bmatrix} -g\sin\theta \\ g\sin\varphi\cos\theta \\ g\cos\varphi\cos\theta \end{bmatrix} - 2\begin{bmatrix} v_z\omega_y - v_y\omega_z \\ v_x\omega_z - v_z\omega_x \\ v_y\omega_x - v_x\omega_y \end{bmatrix} + \\ -2\frac{m_w}{m}\begin{bmatrix} -v_y(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ v_x(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ 0 \end{bmatrix} \end{aligned} \tag{28}$$

The second part are Euler rates, which will be used in order to determine changes of quadcopter local system orientation in relation to the auxiliary system:

$$\dot{\Theta} = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x + \omega_y \sin\varphi\,\mathrm{tg}\theta + \omega_z \cos\varphi\,\mathrm{tg}\theta \\ \omega_y \cos\varphi - \omega_z \sin\varphi \\ \omega_y \sin\varphi\sec\theta + \omega_z \cos\varphi\sec\theta \end{bmatrix} \tag{29}$$

The last part relates directly to angular accelerations:

$$\dot{\mathbf{\Omega}} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \frac{lb}{J_x}\left(-\omega_1^2+\omega_3^2\right) \\ \frac{lb}{J_y}\left(\omega_2^2-\omega_4^2\right) \\ \frac{d}{J_x}\left(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2\right) \end{bmatrix} - \begin{bmatrix} \frac{1}{J_x}\left(\omega_y\omega_z(J_z-J_y)\right) \\ \frac{1}{J_y}\left(\omega_x\omega_z(J_x-J_z)\right) \\ \frac{1}{J_z}\left(\omega_x\omega_y(J_y-J_x)\right) \end{bmatrix} + \\ - \begin{bmatrix} \frac{1}{J_x}\left(\omega_y I_w(\omega_1+\omega_2+\omega_3+\omega_4)\right) \\ \frac{1}{J_y}\left(-\omega_x I_w(\omega_1+\omega_2+\omega_3+\omega_4)\right) \\ 0 \end{bmatrix} \tag{30}$$

3. Mathematical model of the quadrocopter dynamics

3.1. PID control system

Equations (28) – (30) show that the physical control of a quadcopter involves properly changing the angular velocities of the rotors. In relationships (28), the thrust force is assumed to be provided by all the four motors. In practice, the four controllers are implemented to control: the altitude z (by controlling the force F_z), the roll φ (by controlling the moment M_φ), the pitch θ (by controlling the moment M_θ), and the yaw ψ (by controlling the moment M_ψ). The equations for the particular PID controllers can be written as (31) – (34):

$$U_1 = K_{P1}e_1 + K_{I1}\int e_1(\tau)d\tau - K_{D1}\dot{z} \tag{31}$$

$$U_2 = K_{P2}e_2 + K_{I2}\int e_2(\tau)d\tau - K_{D2}\dot{\varphi} \tag{32}$$

$$U_3 = K_{P3}e_3 + K_{I3}\int e_3(\tau)d\tau - K_{D3}\dot{\theta} \tag{33}$$

$$U_4 = K_{P4}e_4 + K_{I4}\int e_4(\tau)d\tau - K_{D4}\dot{\psi} \tag{34}$$

The control errors are: the altitude error e_1 (35), the roll attitude error e_2 (36), the pitch attitude error e_3 (37), and the yaw attitude error e_4 (38).

$$e_1 = z_z - z \tag{35}$$

$$e_2 = \varphi_z - \varphi \tag{36}$$

$$e_3 = \theta_z - \theta \tag{37}$$

$$e_4 = \psi_z - \psi \tag{38}$$

where:
z_z, φ_z, θ_z , ψ_z – setpoints.

As the force F_z and the moments M_φ, M_θ and M_ψ cannot be controlled directly, it is essential to determine the angular velocity of each motor by solving the matrix equation (39). We then obtain solutions (40) – (43).

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -bl & 0 & bl & 0 \\ 0 & bl & 0 & -bl \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{39}$$

$$\omega_1 = \sqrt{\frac{U_1}{4b} - \frac{U_2}{2bl} + \frac{U_4}{4d}} \tag{40}$$

$$\omega_2 = \sqrt{\frac{U_1}{4b} + \frac{U_3}{2bl} - \frac{U_4}{4d}} \tag{41}$$

$$\omega_3 = \sqrt{\frac{U_1}{4b} + \frac{U_2}{2bl} + \frac{U_4}{4d}} \tag{42}$$

$$\omega_4 = \sqrt{\frac{U_1}{4b} - \frac{U_3}{2bl} - \frac{U_4}{4d}} \tag{43}$$

The diagram in Fig. 3 illustrates how the combination of control signals affects the angular velocities of the motors ω_1, ω_2, ω_3 and ω_4.

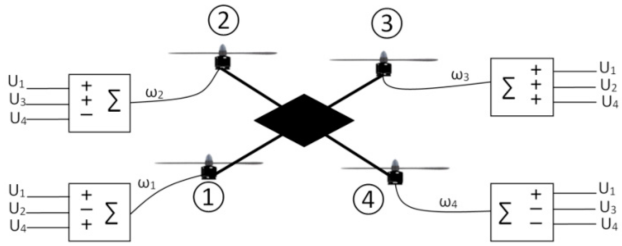

Fig. 3. Control signals U affecting the performance of the motors.

3.2. Optimization of the PID controller gains

Tuning each PID controller for a quadcopter consists in finding its gains: K_{Pi}, K_{Ii}, K_{Di}. Equations (28) – (30) indicate that the drone control model is multidimensional, i.e., with multiple inputs and outputs [16]. The diagram in Fig. 4 shows the tuning of a PID controller.

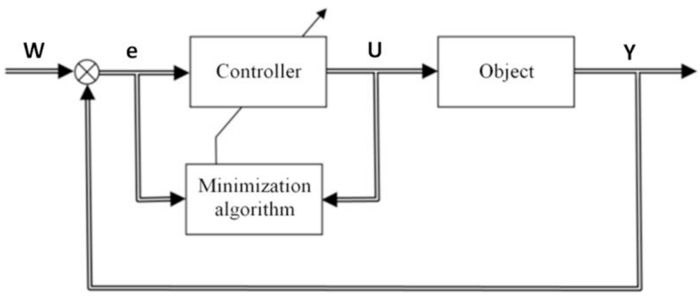

Fig. 4. Block diagram of the PID controller with the minimization algorithm.

The minimization algorithm aims at finding optimal PID control settings at which the control quality indicator is the lowest. In this case, the quality indicator is the error integral for the controlled coordinates z, φ, θ and ψ. The quality indicator is calculated by multiplying the input signal by the penalty coefficient ρ to reduce the input signal level. Ultimately, the quality indicator is written as (44).

$$\int_{t_0}^{t_k} |e_i| + |\rho(U_i - U_{ri})| dt \tag{44}$$

The script developed in Wolfram Mathematica aimed to minimize this indicator by finding appropriate PID settings. Several penalty coefficients were analyzed, but only used a few of them permited to complete the minimization function. This article presents only the best results obtained for ρ = 1. The program was used to test the most common methods for finding a minimum. Unconstrained procedures proved to be unsuitable because some gains were negative. When methods with constraints were used, the gains were satisfactory. However constraints must have been very precisely defined. In this case: from 5 to 17 for the first proportional terms and from 1 to 10 for the other proportional terms, from 0.1 to 0.5 for the integral and from 0.1 to 2 for the first derivative terms and from 0.1 to 2 for the other derivative terms. The simulations showed that the best results were reported for 'Random Search' with the penalty coefficient being ρ = 1 (Table 1)

Table 1. PID gains obtained using the RandomSearch constrained method

Method	Penalty coefficient ρ	Output	K_P	K_I	K_D	Quality indicator
Random Search	1	z	11.31334	0.19547	1.24053	0.47815
		φ	1.16725	0.41442	0.34797	0.00572
		θ	3.78533	0.24066	0.38672	0.00147
		ψ	3.55551	0.32859	0.25760	0.00351

4. Simulation results

Wolfram Mathematica was used to perform simulations for the parameters provided in Table 2. The simulation time was 3 seconds.

$$x(0) = 0.1,\ x'(0) = 0,\ y(0) = 0.1,\ y'(0) = 0,\ z(0)=5,\ z'(0) = 0,\ \varphi(0) = 0,\ \theta(0) = 0,\ \psi(0) = 0,\ \omega_x(0)=0, \omega_y(0)=0,\ \omega_z(0)=0,\ i_{e1}(0)=0,\ i_{e2}(0)=0,\ i_{e3}(0)=0,\ i_{e4}(0)=0, \tag{45}$$

Table 2. Parameters of the quadcopter control model

Parameter	Value	Unit
m	0.65	kg
m_w	0.01	kg
l	0.23	m
g	9.81	kg m/s^2
I_w	0.000065	kg m^2
J_x	0.0075	kg m^2
J_y	0.0075	kg m^2
J_z	0.0013	kg m^2

The setpoints for the variables z, φ , θ and ψ are shown in Table 3.

Table 3. Setpoints

$z_z(t)$ [m]	$\varphi_z(t)$ [rad]	$\theta_z(t)$ [rad]	$\psi_z(t)$ [rad]
6	0.04	0.05	0.1

After introducing the calculated values of the PID gains in equations (28) – (30), we plotted the process variables. Figure 5a) illustrates the three position coordinates x, y and z changing with time. Figure 5b), on the other hand, shows the relationship between the three angles φ, θ and ψ and time.

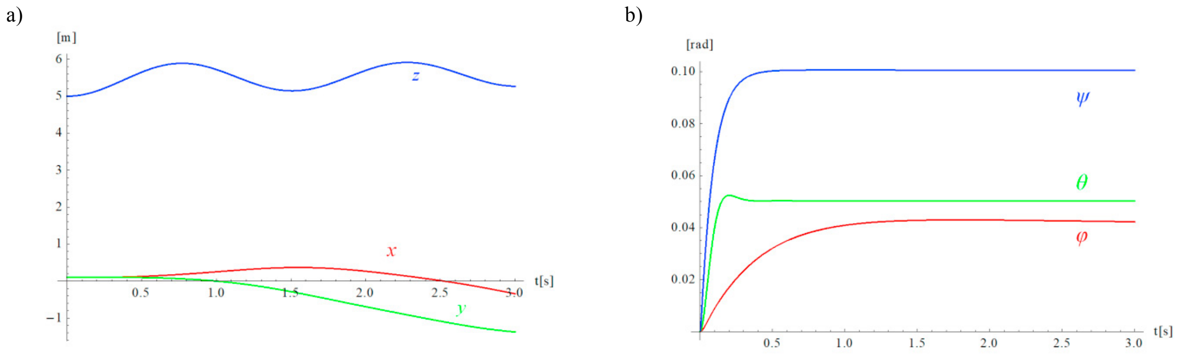

Fig. 5. Drone altitude after PID tuning: a) position in space, b) rotations in space

5. Conclusion

The article has shown that the proposed method for tuning the PID gains works correctly. The process requires applying the quality indicator. Not all Wolfram Mathematica optimization procedures turned out to be suitable for finding the right minimum. Different constrained and unconstrained optimization methods were analyzed and simulation was employed to compare them. The simulations showed how the penalty coefficient ρ affected the optimization results. For the PID-controlled plant, i.e., the quadcopter, the best results were reported when the 'RandomSearch' constrained minimization method was used and when ρ was equal to 1. The proposed criterion worked well for many gains. The penalty term was particularly important for the input signal. When the quality criterion was considered without the input signal or the penalty coefficient, the optimization results were unsatisfactory because of significant overshoots and oscillations. This procedure can be successfully used to optimize gains for a higher number of controllers. Mathematica proves to be one of the best modelling programs, which ensures fast optimization of complex criteria.

References

[1] Audronis T., Building Multicopter Video Drones, Wydawnictwo Packt Publishing, Birmingham 2014 r., ISBN: 9781782175445

[2] Ben Ammar N., Bouallegue S., Haggege J., Modeling and sliding mode control of a quadrotor unmanned aerial vehicle, 3rd International Conference on Automation, Control, Engineering and Computer Science, (2016 r.) pp. 834-840

[3] Bouabdallah, S., Noth A., Siegwart R., PID vs LQ control techniques applied to an indoor micro quadrotor, IEEE/RSJ International Conference on Intelligent Robots and Systems, (2014 r.) pp. 2451 – 2456.

[4] Cheng L. L., Liu H. B., Examples of Quadrocopter control on ROS, IEEE 9th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2015 r.

[5] Florek M., Huba M., Duchon F., Sovcik J., Kajan M., Comparing approaches to quadrocopter control, 23rd International Conference on Robotics in Alpe-Adria-Danube Region, Słowacja 2014 r.

[6] Gardecki S., Giernacki W., Gośliński J., Kasiński A., An adequate mathematical model of four-rotor flying robot in the context of control simulations, [in:] Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 8, no. 2 (2014 r.), pp. 9-16, ISSN: 1897-8649

[7] Gliński H., Grzymkowski R., Kapusta A., Słota D., Mathematica 8, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2012 r., ISBN: 978-83-62652-33-4

[8] Herrera M., Chamorro W., Gómez A. P., Camacho O., Sliding mode control An approach to control a quadrotor, Conference: APCASE 2015, At Quito, Ecuador, (2015 r.)

[9] Holonec R., Copindean R., Dragan F., Rapolti L., Self-quided AR Drone using LabVIEW, [in:] Acta Electrotehnica Vol. 57 No. 5 (2016 r.) , pp. 600-603, ISSN: 1841-3323

[10] Janecki D., Globally stable and exponentially convergent adaptive control, [in:] International Journal of Control, Vol. 43 No. 2 (1996 r.) , pp. 601-613, ISSN: 0020-7179

[11] Koruba Z., Control and correction of a gyroscopic platform mounted in a flying object, [in:] Journal of Theoretical and Applied Mechanics, vol. 45, no. 1, (2007 r.), pp.41–51, ISSN: 1429-2955

[12] Koruba Z., Dziopa Z., Krzysztofik I., Dynamics and control of a gyroscope-stabilized platform in a self-propelled anti-aircraft system, [in:] Journal of Theoretical and Applied Mechanics, vol. 48, no 1 (2010 r.), pp. 5 –26, ISSN: 1429-2955

[13] Laski P., Fractional-order feedback control of a pneumatic servo-drive, [in:] Bulletin of the Polish Academy of Sciences: Technical Sciences, Vol. 67, No. 1 (2019 r.), pp. 53-59, ISSN: 0239-7528

[14] Rabhi A., Chadli M., and Pegard C., Robust Fuzzy Control for Stabilization of a Quadrotor, 15th International Conference on Advanced Robotics (ICAR), (2011 r.), pp. 471 - 475

[15] Seidler J., Badach A., Molisz W., Metody rozwiązywania zadań optymalizacji, Wydawnictwa Naukowo-Techniczne, Warszawa 1980 r., ISBN: 83-204-0256-5

[16] Víteček A., Cedro L., Farana R., Modelowanie matematyczne: podstawy, Wydawnictwo Politechniki Świętokrzyskiej, Kielce 2010 r., ISBN: 978-33-88906-28-2