Studies on the control quality improvement for multirotor flying robot

WOJCIECH GIERNACKI, STANISŁAW GARDECKI, JAROSŁAW GOŚLIŃSKI
Institute of Control and Information Engineering
Poznan University of Technology
ul. Piotrowo 3A, 60-965 Poznan
POLAND
wojciech.giernacki@put.poznan.pl   http://d1.cie.put.poznan.pl

Abstract: - In this paper research results on control quality improvement by the use of particle swarm optimization algorithm (PSO) was described as an alternative for the classic PID tuning. For the problem of multi-rotor flying robot control, test studies were conducted using a specially for this purpose designed and performed measuring testbed. This solution allows to obtain the angle as a feedback in flying robot stabilizing loop and it was compared with the reference angle measured at the test stand. In the first part of the paper robot construction, test stand and tests related with the calibration and correctness of indications were described. In the second part, a method of the PID controller tuning by particle swarm optimization algorithm, their subsequent verification and fine tuning were presented. The results obtained with PSO algorithm were compared to results of empirical tuning of the PID controller.

Key-Words: - multirotor flying robot, quadrotor, particle swarm optimization algorithm, PID controller tuning, test stand, flying robot stabilization, quality control improvement

1 Introduction

In multi-rotor flying robots constructions for the stabilization of platform in the air one may use several types of controllers, including PID [1]. The choice of controller parameters sets depends mainly on the dynamics of the flying platform. In detail it depends from the length of robot arms, as well as from drive units [2]. Further factors which play a major role in the stabilization of the control object, are: a main control loop update rate and a measurements correctness obtained from the on-board sensor technology. Most of amateur construction flies on controller parameters selected in empirical way, but such a solution brings not exactly desired effect: stable and fully controlled flight of flying platform. Often in a practice constructions with such a control stabilize poorly in the air – a large drift may be observed. Such a behavior forces in the effect to frequent correction of robot position by the operator, which impairs the maneuvering precision and contributes to reduced of the time that platform can remain in the air. The excessive maneuver entails an increase of energy consumption from the batteries than the calm, smooth flight. Due to the above-mentioned problems one is looking for effective methods of controllers tuning [3],[4],[5] which would be an alternative to the commonly used, classic PID controller tuning based on empirical knowledge of the operator and platform designer.

The novelty and contribution of the authors of this study is to propose and implement two (varied in terms of the desired effect: two different characteristics of the flight) methods for calculating the cost function in the problem of the quality improvement of the PID type controllers tuning (by the use of the particle swarm optimization algorithm - PSO) for drives of multi-rotor flying robot. For now these scientists have focused mainly on the search for a one, universal type of controllers tuning for any task [6] or on the use of the PSO algorithm for other purposes related to the flying robots such as i.e. a path planning [7].

2 Problem Formulation

This article was written primarily to show the effectiveness of proposed method for the PID controller tuning – through a combination of the simulation results with real tests on the flying robot located in a specially designed and built test stand. As a flying platform for the tests one used four rotor flying robot Falcon (next generation of the Hornet robot described in detail in [8]) and shown in Figure 1. A detailed description of the mathematical model and the equations describing the dynamics of the Falcon robot may be found in [9]. In Figure 2 it may be seen a block diagram with controllers used to control the orientation and rotational speed of quadrotor.

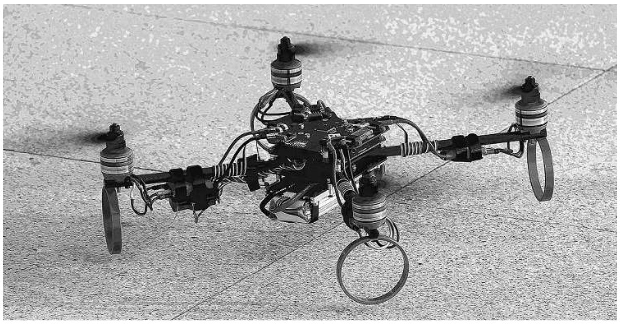

Fig. 1 Multi-rotor flying robot Falcon

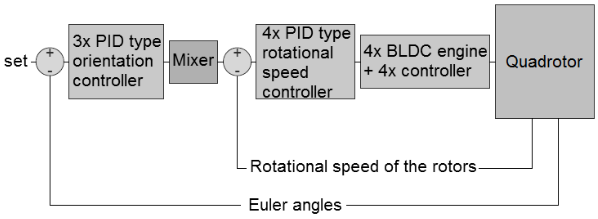

Fig. 2 A simplified block diagram of the Falcon robot control system

Falcon is a robot based on the cross frame. At the end of each arm a drive unit responsible for the movement of the robot in the air is located. In the central part of the platform on-board controller responsible among others for the stabilization of robot in the air is mounted. To supply the platform in energy a lithium-polymer battery is screwed under the robot.

The article presents test results obtained by simulations conducted on the mathematical model [9] and with the use of data obtained from the real Falcon robot. In order to find the PID controller sets particle swarm optimization algorithm has been applied. After receiving of the initial sets, numerous tests were conducted with the use of flying platform – in order to improve the received parameters. By such an action one may verify and improve the reflection of mathematical model in relation to the real robot [9], as well as better and faster select the mentioned PID controller parameters sets to a specific, other than that presented flying platform. Such researches will allow in a future time for more accurate representation of the simulated object behavior by the use of a more precise mathematical model, which will open the possibility of using in platform control strictly analytical methods, such as already proposed coefficient diagram method – described in detail in [10]. It is very desirable from the perspective of potential use of multi-rotor flying robots in patrol tasks in semi-autonomous and autonomous modes.

3 Test stand and experiments

To ensure the repeatability of recorded tests results, as well as to provide necessary security, a test stand was designed and build (Fig.3). The movement of robot located in the test stand was restricted to one axis. Due to the type of power supply used in the robot (battery), one introduced an additional security – the exclusion of full rotation possibility in the aforementioned axis. Without such a solution in the case when robot will fall in to a spin state in unlimited axis – turning robot off would be very difficult and dangerous. Introduced restrictions do not affect on the proper work of robot, and thus on the conducted experiments which are related to the platform stabilization and to the selection of controller sets.

Developed test stand consists of a rigid base, made of aluminum profiles with a cross section 32x32 mm. Profiles have been twisted with M8 screws using triangular brackets, ensuring the right angle of connected elements. This solution enables (if necessary) to a fast modification of the test stand construction and adaptation to the other, new requirements. Two right angled profiles with a length of 800 mm have been installed to the base. At their ends bearings were fitted to enable assembling of gripping elements of the flying robot arms. To increase the stiffness of the whole stand, as well as to reduce the possibility of robot overturning, crossbar between mentioned arms was installed. Test stand was tared using a spirit level and adjustable feets. The base has been further laden with concrete block.

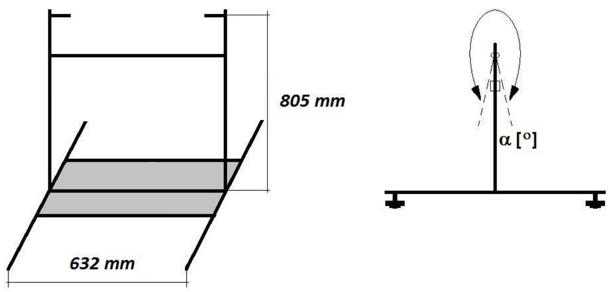

Fig. 3 Outline drawings of the test stand

Mounting the robot to the test stand was carried out from two sides with aluminum handles and M3 screws. Handles were set on steel axes, inserted into the aforementioned bearings. To the one of the axes

ISBN: 978-1-61804-279-8

a magnetic encoder was fastened. Thus the control of data correctness received from on-board sensor technology was enabled. This ensured also generation of a reference signal of robot yaw angle (α). To work with encoder one programmed a printed circuit board based on the STM32 processor. Data received from the encoder were converted and transferred via USB to a computer. Due to the synchronization of robot controller board with the system on the test stand, there was a possibility to compare of the received data. A computer application supervises the whole system and captured the data received during the test at a frequency of 400 Hz. The application has a possibility of online visualization of generated data in the form of numbers and graphs. The program has also the ability to archive data for subsequent simulations and comparisons. After the robot was installed on the test stand, it was balanced so that its horizontal plane was parallel to the base. The readings obtained from the encoder and board sensor technology were calibrated to show the zero in robot equilibrium.

The research tests started from the verification of readings correctness received from the controller board and test stand in the terms of robot real tilt. In the first test, robot was inclined by 5 degrees and then left free to return to balance. Further tests relied on tilting robot increasingly – due to 10, 20, 25, 45 degrees and mixed combination of angles values. One of recorded tests is presented in Figure 4.

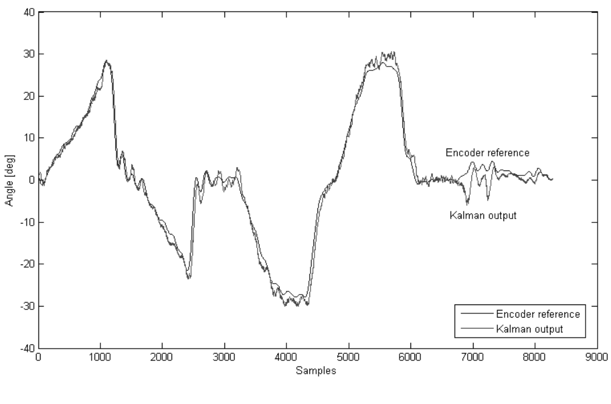

Fig. 4 Test of tilting the robot from the balance

From the obtained test results it may be seen the fit of encoder reference signal to Kalman output signal. In all conducted tests the data recorded from encoder coincided with a set angle, while those data received from the controlled board have some margin of error. This is the result of errors and operations executed in the controller board. In order to close the feedback from the state, robot uses the IMU (Inertial Measurement Unit).

The IMU consists among other things from the gyroscopes and accelerometers thanks to which it is possible to stabilize the robot in the air. In the following tests and experiments described further in the article, time characteristic obtained from a magnetic encoder was used as a reference.

4 Tests with the PSO algorithm

Developed in year 1995 by R. Kenedy and E. Eberhart particle swarm optimization algorithm [11], was inspired by the herd behavior of animals (insects, fish, birds) that are influenced by the behavior of single individuals (particles) which communicate between each other in the search for optimum. It is used for many years successfully in control engineering and robotics for solving optimization problems [12],[13],[14]. Algorithm's simplicity and high efficiency at the same time is also used for tuning of the PID controllers. Therefore the idea of the PSO algorithm application for such a complex issue as stabilization in the air of such a complicated organism as a four rotor flying robot (characterized by multi-dimensional control structure and high dynamics), was born.

A detailed description of the PSO algorithm and its modifications are available in a number of publications [15],[16]. In the main variants algorithm uses the fact that each individual knows its position, remembers the best positions achieved so far, communicates with other individuals in order to collectively search the space of possible positions in the aim to find the optimum. In the quest appears also significant, physical characteristic – the speed of the individual and the fact that in every community of individuals must be a leader with the best fit – who gives direction to change to the whole population.

The mechanism of the algorithm is based on the principle that up to a specified stop condition (for example – arbitrarily set number of iterations of the algorithm) speed is calculated for each individual from the population. His position is changed in accordance with the calculated velocity. On this base a value of optimized function is calculated and it is compared to the best one obtained so far by the individual. If the new position is better than the best previous position of the individual, it replaces the previous, and is compared to the best obtained globally. If it is also better from it, it remains the best position found globally and gives a new direction of change.

ISBN: 978-1-61804-279-8

In the tests results described further object model approved for the study concerned the Euler angles and angular velocities in the axes X, Y and Z (respectively: phi, theta and psi). The model was considered at the speed constraints – after crossing the maximum values in the loop saturation was activated. An example of code syntax for the speed controller of the X axis is given:

```
e_fi=fi_z-fi(k+1);
Pid_fi=Kp_fi*e_fi+sum_fi*Ki_fi+Kd_fi*(
e_fi-e_fi_p);
sum_fi=sum_fi+e_fi;
e_fi_p=e_fi;

         if sum_fi>100
            sum_fi=100;
         end
         if sum_fi<-100
            sum_fi=-100;
         end

w2(k+1)=THRO-Pid_fi+Pid_psi;
w4(k+1)=THRO+Pid_fi+Pid_psi;
w1(k+1)=THRO-Pid_theta-Pid_psi;
w3(k+1)=THRO+Pid_theta-Pid_psi;
```

where K_p, K_i, K_d are gains of the PID controller blocks, while THRO is an offset for the maintenance of flight altitude; w1, w2, w3, w4 are rotors set speeds at the adoption of simplification that set values are obtained immediately after their set (due to the absence of rotation control).

To the PSO algorithm in the PID controller tuning, following simulation parameters were used: the amount of particles: 15, coefficients: c1=1.8, c2=1.8, V_{max}=4, V_{min}=-4, ksi=0.73, omega=1.2. Parameters: *ksi* and *omega* are used for mutation of particle swarm optimization algorithm. All the PSO algorithm parameters were chosen in accordance with the information from [11] as well as with the recommendations from the Internet forum: *swarmintelligence.org*. For example the amount of particles should be chosen from the numerical range (10, 20). Due to the limited space in this article, only the most important parts of the PID controller tuning process by the use of the PSO algorithm, were shown. Full step-by-step description of the procedure with the flow chart, as well as the code implementation may be found in the planned, extended version of the article or may be obtain after direct contact with the authors of this paper.

In the first test results were obtained for a single PID controller (equivalent of researches on a test bench with one available rotary axis) and they varied between themselves depending on the method of control cost calculation – defined respectively as:

a) quality=quality+|regulator_pfi->e|

where *regulator_phi->e* is a control error, and *quality* is calculated iteratively at each step.
As a result of the PSO algorithm application following PID controller sets were obtained:
K_p=19,0204, K_i=0.001, K_d=200 at control cost of K=16.3079.
For distortion levels definied as:

```
if  k>3300&&k<3320
       fi(k+1) = 0.2;
end
if k>3580&&k<3600
       fi(k+1) =-0.2;
end
```

time characteristics from Figures 5 and 6 were obtained.

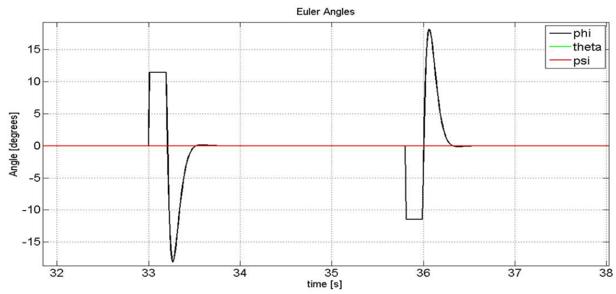

Fig. 5 Step changes of *phi* angle due to the changes of rotors velocity *w2* and *w4* (test no.1, point a)

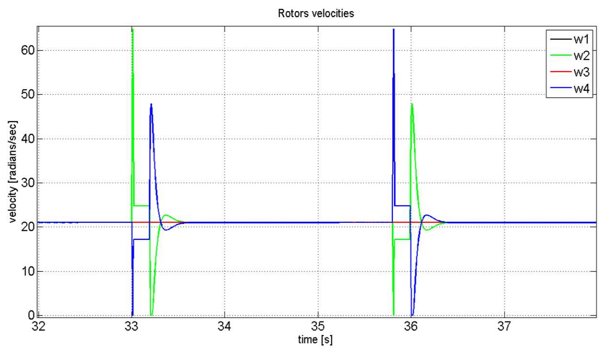

Fig. 6 Time characteristics of rotors velocities (test no.1, point a)

b) quality=quality+(regulator_pfi->e)^2

Following optimization results were obtained:
K_p=0.556886, K_i=0.001, K_d=144.193 at control cost K=1.82818, what for similar distortion levels as at point b) allowed to register time characteristics from Figures 7 and 8.

ISBN: 978-1-61804-279-8

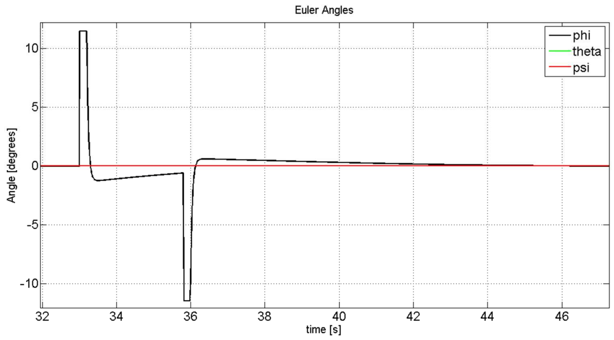

Fig. 7 Step changes of *phi* angle due to the changes of rotors velocity *w2* and *w4* (test no.1, point b)

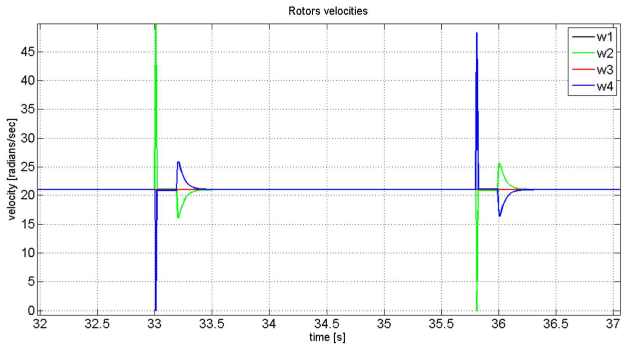

Fig. 8 Time characteristics of rotors velocities (test no.1, point b)

In the second test time characteristics were recorded for controllers in three axes (X, Y, Z) - fully reflecting the behavior of a flying robot. Two ways of cost function calculation were introduced:

a) quality=quality+(regulator_phi->e)^2+ (regulator_theta->e)^2+ (regultor_psi->e)^2
b) quality=quality+|regulator_phi->e|+ |regulator_theta->e|+|regulator_psi->e|.

As the results of the PSO algorithm (number of iterations equal to 500) following control cost were recorded:
- in test no.2, point a: K=13.7603,
- in test no.2, point b: K=79.1884.

In test no.2, point a, following controller sets:
Kp_fi=0.29559,
Ki_fi=0.000290527,
Kd_fi=119.234,

Kp_theta=0.0392697,
Ki_theta=5.01708*10^(-5),
Kd_theta=85.8451,

Kp_psi=2.75618,
Ki_psi=0.000340892,
Kd_psi=141.76,

and time characteristics from Figures 9-10 were obtained.

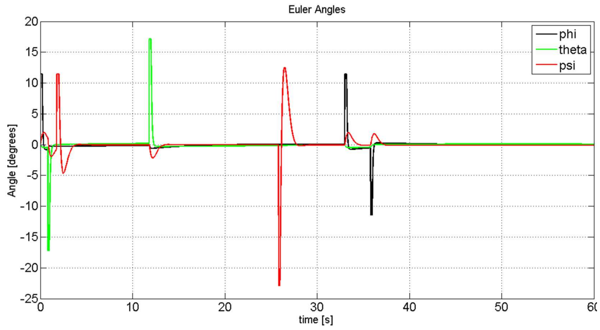

Fig. 9 Step changes of *phi*, *theta* and *psi* angles due to the changes of rotors velocity (test no.2, point a)

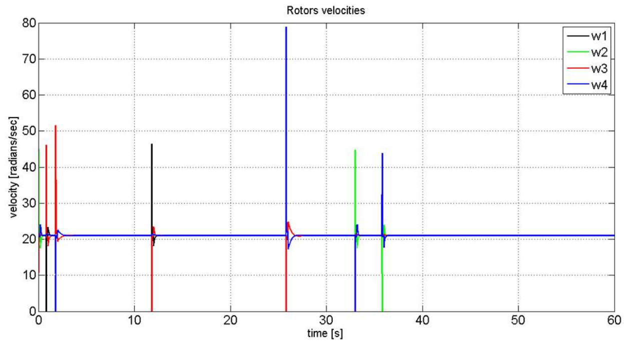

Fig. 10 Time characteristics of rotors velocities (test no.2, point a)

5 Conclusion

From the research results it is known that the use of a swarm algorithm to determine parameters of the PID controller and their further improvement in the real object may become an alternative for the selection of controller parameters in empirical way (often randomly or from the operator's or platform designer's experience). Achievement of better control results depends mainly from the used method of cost function calculating. Comparison of test results (test no.1, point a and b) proves that in addition to the analysis of the cost function value, one need to follow in detail the recorded time characteristics and decide whether more care about the dynamic control (test no.1, point a) with large changes of Eulers angles and fast time of equilibrium obtain, or by minimization of the cost function as in the second method (test no.1, point b) one has on the aim to achieve a smooth balance in the longer time.

Controller settings chosen in the proposed manner (test no.2, point a) stabilize robot more better in the air than other approaches to this issue. This fact allows to authors of this paper to start a next phase

ISBN: 978-1-61804-279-8

of works – to develop a system for patrol flights in semi-autonomous and full-autonomous mode.

References:

[1] J. Li, Y. Li, Dynamic Analysis and PID Control for a Quadrotor, *2011 International Conference on Mechatronics and Automation (ICMA)*, Beijing, 7-10 August 2011, pp. 573–578.

[2] St. Gardecki, A. Kasiński, Testing and Selection of Electrical Actuators for Multi-rotor Flying Robot, *PAK*, No.1, 2012, pp.80-83.

[3] C. Coza, C.J.B. Macnab, A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization, *2006 NAFIPS Annual meeting of the North American*, Montreal, 3-6 June 2006, pp.454-458.

[4] C. Nicol, C.J.B. Macnab, A. Ramirez-Serrano, Robust Neural Network Control of a Quadrotor Helicopter, *2008 CCECE Canadian Conference on Electrical and Computer Engineering*, Niagara Falls, 4-7 May 2008, pp. 1233-1238.

[5] Y. Xu, C. Tong, Quantitative Feedback Control of a Quadrotor, *2012 IEEE International Symposium on Industrial Electronics (ISIE)*, Hangzhou, 28-31 May 2012, pp. 1309-1314.

[6] M.A. Ayoubi, E. Ting, An Optimal Fuzzy-PID Controller for Aircraft Pitch Control, *AIAA Infotech@Aerospace 2012*, Garden Grove, California, 19-21 June 2012, doi: 10.2514/6.2012-2580.

[7] Z. Hongguo, Z. Changwen, H. Xiaohui, L. Xiang, Path planner for unmanned aerial vehicles based on modified PSO algorithm, *International Conference on Information and Automation ICIA 2008*, Hunan, 20-23 June 2008, pp. 541-544.

[8] St. Gardecki, A. Kasiński, Multirotor Flying Robot Hornet, *Scientific Papers of the Warsaw University of Technology*. *Electronics,* Vol.175, No.2, 2010, pp. 655-664.

[9] St. Gardecki, W. Giernacki, J. Gośliński, A. Kasiński, An Adequate Mathematical Model of Four-rotor Flying Robot in the Context of Control Simulations, *Journal of Automation, Mobile Robotics & Intelligent Systems*, Vol.8, No.2 2014, pp. 9-16.

[10] S. Manabe, Coefficient Diagram Method, *14th IFAC Symposium on Automatic Control in Aerospace*, Seoul, 1998, pp. 199-210.

[11] J. Kennedy, R. Eberhart, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, Vol.4, 1995, pp. 1942–1948.

[12] D. Gong, L. Lu, M. Li, Robot Path Planning in Uncertain Environments Based on Particle Swarm Optimization, *CEC '09 IEEE Congress on Evolutionary Computation*, Trondheim, 18-21 May 2009, pp. 2127-2134.

[13] X. Luo, Sh. Shan, Sh. Wang, Particle-swarm Optimization Algorithm for Model Predictive Control of MIMO with Constraints, *2012 10th World Congress on Intelligent Control and Automation (WCICA)*, Beijing, 6-8 July 2012, pp. 2576-2581.

[14] T. Adel, C. Abdelkader, A Particle Swarm Optimization Approach for Optimum Design of PID Controller for Nonlinear Systems, *2013 International Conference on Electrical Engineering and Software Applications (ICEESA)*, Hammamet, 21-23 March 2013, pp. 1-4.

[15] A. Ltaief, A. Chaari, A. Taeib, PID Control Based on Modified Particle Swarm Optimization for Nonlinear Process, 2013 *World Congress on Computer and Information Technology (WCCIT)*, Sousse, 22-24 June 2013, pp. 1-5.

[16] M. Ebadzadeh, A. Nickabadi, R. Safabakhch, A Novel Particle Swarm Optimisation Algorithm with Adaptive Inertia Weight, *Journal of Applied Soft Computing*, Vol.11, 2011, pp. 3658-3670.