Simulation of Cascade Control for Two Rotor Aerodynamical System Using FOPID and PID Controllers

Jakub Żegleń-Włodarczyk
AGH University of Krakow, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering

Abstract: Two Rotor Aerodynamical System is a non-linear system in which cross-coupling occurs. In this specific case the individual rotors of the system affect both the system angles: azimuth and pitch. Therefore, both values must be used in each feedback loop. This makes it possible to use cascade control. It was decided to use FOPID as primary controllers and PID as secondary. For comparison purposes, an analogous control system with only PIDs was prepared. The coefficients were determined using the GWO algorithm. All simulations were performed in the MATLAB/Simulink environment. In addition to comparing the values of the cost function, the execution time of individual controllers was also checked.

Keywords: FOPID, PID, TRAS, Two Rotor Aerodynamical System, optimization, GWO algorithm, ORA approximation, execution time

1. Introduction

Recently, airspace is becoming an increasingly discussed topic. In addition to the well-known planes and helicopters, drones are becoming an increasingly popular topic. They make it possible to carry small objects [1] or search for people [2]. The Two Rotor Aerodynamical System is a system that has two rotors arranged similarly to a helicopter. In addition, it is tethered, which allows to safely control the azimuth and pitch angles. The angle of attack of the rotors is constant [3], thus the control is only possible via the DC motors. It is a non-linear cross-coupling system [4], which means that each rotor affects both measured angles.

Choosing the type of controller is one of the most important decisions. There are many possibilities, eg. PID [5], LQR [6], Model Predictive Control (MPC) [7], fuzzy neural network control [8]. The opportunity to compare some options with each other may help with decision. PID, which is one of the most popular and robust controllers [9], can be a good determinant.

Four controllers are used to control the TRAS system – two for each angle (due to the presence of cross-coupling). This multiplies the number of coefficients that need to be chosen. Various optimization algorithms are used to find the best values. Methods based on real examples from nature are very popular. These include Whale Optimization Algorithm (WOA) [10], Firefly Algorithm (FA) [11], Grey Wolf Optimizer (GWO) [12].

The concept of fractional calculus dates back to the 17th century, when the derivative $\alpha = 1/2$ was described by Leibnitz in a letter to L'Hospital [13]. It is a field of mathematics that requires a lot of computing power, which is why it has been developed only in recent years thanks to computers. It enables a more precise description of various dependencies, including also controllers such as PID, from which the Fractional Order PID – FOPID was created.

In the TRAS system the cross-coupling phenomenon appears, which requires the use of two controllers (one for each output) to calculate control values for a single input. In this article it was decided to arrange them in a cascade manner. FOPID controllers serve as a primary part in the control loop and PID as an secondary. For comparison purposes, a second variant of analogous control was prepared. It only contains PID controllers. This will allow to check if the cascade approach is sufficient and whether the cooperation between FOPID and PID controllers will give better results than the reference control version.

The next chapter describes the PID and FOPID controllers. Later section outline Grey Wolf Optimizer method. The following chapter describes the mathematical model of the TRAS system and the implementation of the simulation. At the end, the results of the simulation are presented together with the execution times. The article ends with a summary with conclusions.

Autor korespondujący:
Jakub Żegleń-Włodarczyk, zeglen@agh.edu.pl

Artykuł recenzowany
nadesłany 18.11.2024 r., przyjęty do druku 14.04.2025 r.

2. PID and FOPID Controller

The FOPID controller has two additional parameters in relation to the PID. This is because fractional calculus assumes

that the derivatives may be of an non-integer order. Therefore, the transfer function of the FOPID controller takes the form as shown in equation (1):

$$G_c(s) = K_P + \frac{K_I}{s^{\lambda}} + K_D s^{\mu} \tag{1}$$

where: K_P, K_I, K_D – respectively coefficients of proportional, integral, derivative action; λ – order of integral; μ – differentiation part.

Due to the use of fractional order calculus, s should be of a non-integer order. For this purpose in this article the Oustaloup Recursive Approximation method was used, which is done in the frequency range of ω [14]:

Symbols needed to describe the system

- J_v – sum of moments of inertia relative to the horizontal axis
- J_h – sum of moments of inertia relative to the vertical axis
- α_v – pitch angle of the beam
- α_h – azimuth angle of the beam
- Ω_h – angular velocity of the beam around the vertical axis
- Ω_v – angular velocity around the horizontal axis
- ω_v – angular velocity of the main rotor
- ω_h – angular velocity of the tail rotor
- l_m – length of the main part of the beam
- l_t – the length of the tail part of the beam
- $F_v(\omega_m)$ – dependence of the propulsive force on the angular velocity of the rotor
- $F_h(\omega_t)$ – dependence of the propulsive force on the angular velocity of the tail rotor
- f_v – friction coefficient in the horizontal axis
- f_h – friction coefficient in the vertical axis
- U_h – horizontal DC motor PWM control input
- U_v – vertical DC motor PWM voltage control input
- K_h – horizontal angular momentum
- K_v – vertical angular momentum
- m_t – mass of the tail part of the beam
- m_{tr} – mass of the tail motor with tail rotor
- m_{ts} – mass of the tail shield
- m_m – mass of the main part of the beam
- m_{mr} – mass of the main DC motor with main rotor
- m_{ms} – mass of the main shield
- m_b – mass of the counter weight beam
- m_{cb} – mass of the counter weight
- l_t – length of the tail part of the beam
- l_m – length of the main part of the beam
- l_b – length of the counter weight beam
- l_{cb} – distance between the counter weight and the joint
- r_{ms} – radius of the main shield
- r_{ts} – radius of the tail shield
- k_{hv} – constant
- k_{vh} – constant
- a_1 – constant
- a_2 – constant

$$s^{\lambda} \approx C_o \Pi \frac{s+\omega_k'}{s+\omega_k} \tag{2}$$

where each coefficient is as below:

$$\omega_k' = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+0.5(1-r)}{2N+1}},$$

$$\omega_k = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+N+0.5(1+r)}{2N+1}}, \tag{3}$$

$$C_o = \left(\frac{\omega_h}{\omega_b}\right)^{\frac{-r}{2}} \Pi \frac{\omega_k}{s+\omega'}.$$

where $r \in \Re$ and r is in the range $[-1, 1]$, (ω_b, ω_h) are the pulsation interval, N is degree of the approximation.

3. Grey Wolf Optimizer

The behavior of grey wolves was the inspiration for the Grey Wolf Optimizer algorithm [15]. Two aspects were taken into account: hierarchy and hunting method.

The GWO algorithm distinguishes four subgroups in its hierarchy:
- alpha – leader,
- beta – individuals supporting alpha,
- delta – scouts, hunters, etc.,
- omega – the weakest individuals.

Furthermore, the hunting process has been divided into three steps:
- tracking and approaching a victim,
- circling and harassing,
- attacking the victim.

The equations (4)–(6) describe how the algorithm is performed:

$$D = \left|C \cdot X_v(t) - X(t)\right| \quad C = 2 \cdot r_2 \tag{4}$$

$$X(t+1) = X_v(t) - A \cdot D \quad A = 2a \cdot r_1 - a \tag{5}$$

$$X(t+1) = \frac{X_\alpha + X_\beta + X_\delta}{3}, \tag{6}$$

where X is the position of wolf, X_α, X_β and X_δ specify the location vector of the alpha, beta and delta, X_v is the location of prey, t is the current iteration, r_1, $r_2 \in [0, 1]$ are random numbers, coefficient a decrease during calculations from 2 to 0.

The algorithm ends the calculation after reaching the maximum number of iterations.

4. Mathematical model of the Two Rotor Aerodynamical System

Two Rotor Aerodynamical System is a laboratory model of a miniature tethered helicopter (Fig. 1). It is a system that can be classified as non-linear MIMO (Multiple Input Multiple Output) [16].

There are two rotors in the system. One rotates around the vertical axis, the other around the horizontal – like helicopters. However, unlike them, there is no possible way to adjust the angle of attack of the rotors, only the spin speed. Both rotors are connected by a beam that can rotate and change its angular position around a second vertical beam that serves as the connection to the base. With the use of encoders it is possible to obtain the values of horizontal angle and vertical angle.

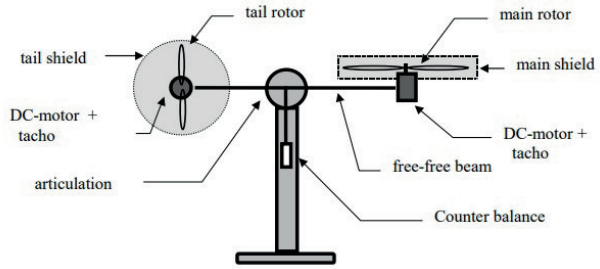

Fig. 1. Aerodynamical model of TRAS [16]
Rys. 1. Model aerodynamiczny systemu TRAS [16]

For the preparation of the mathematical model, it was assumed that [16]:
– the behavior of the propeller can be described by the first order differential equations,
– friction in the system can be classified as viscous type,
– the behavior of the propeller-air subsystem can be described in accordance with the assumptions of the flow theory.

Ultimately, the TRAS system can be described by the equations (7)–(10) – as shown below [16]:

$$\frac{d\omega_v}{dt} = \frac{l_m F_v(\omega_m) - \Omega_v k_v + g\left((A-B)\cos\alpha_v - C\sin\alpha_v\right)}{J_v} \cdots \\ \frac{-\frac{1}{2}\Omega_h^2 (A+B+C)\sin 2\alpha_v U_h k_{hv} + U_h k_{hv} - a_1 \Omega_v abs(\omega_v)}{J_v} \tag{7}$$

$$\frac{d\alpha_v}{dt} = \Omega_v \tag{8}$$

$$\frac{dK_h}{dt} = \frac{M_h}{J_h} = \frac{l_t F_h(\omega_t)\cos\alpha_v - \Omega_h k_h + U_v k_{vh}}{D\sin^2\alpha_v + E\cos^2\alpha_v + F} \cdots \\ \frac{-a_2 \Omega_h \mathrm{abs}(\omega_v)}{D\sin^2\alpha_v + E\cos^2\alpha_v + F} \tag{9}$$

$$\frac{d\alpha_h}{dt} = \Omega_h = \frac{K_h}{J_h \alpha_v} \tag{10}$$

where:

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right) l_t$$

$$B = \left(\frac{m_m}{2} + m_{mr} + m_{ms}\right) l_m$$

$$C = \frac{m_b}{2} l_b + m_{cb} l_{cb}$$

$$E = \left(\frac{m_m}{3} + m_{mr} + m_{ms}\right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts}\right) l_t^2$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2$$

5. Implementation of simulation

The cross-coupling phenomenon appears in the TRAS, which means that both position angles should be taken into account for each control input to the system.

In the article, it was decided to test the interaction of PID and FOPID controllers in a cascade control system. For this purpose, a simulation in the MATLAB/Simulink environment was prepared (Fig. 2). To implement PID controllers, default PID blocks were used with the filter coefficient value set to 100. The model of the TRAS system (the grey block in the middle of the figure – provided by INTECO) takes the control values for both rotors as input. The output from the system are azimuth angle, pitch angle, azimuth RPM, pitch RPM.

The TRAS simulation model provided by INTECO was used in default configuration, which means that the values were assumed according to the instruction [16]:

$m_t = 0.032$ kg	$lt = 0.216$ m
$m_{tr} = 0.225$ kg	$lm = 0.202$ m
$m_{ts} = 0.061$ kg	$lb = 0.145$ m
$m_m = 0.03$ kg	$lcb = 0.15$ m
$m_{mr} = 0.252$ kg	$rms = 0.145$ m
$m_{ms} = 0.083$ kg	$rts = 0.1$ m
$m_b = 0.1$ kg	
$m_{cb} = 0.0256$ kg	

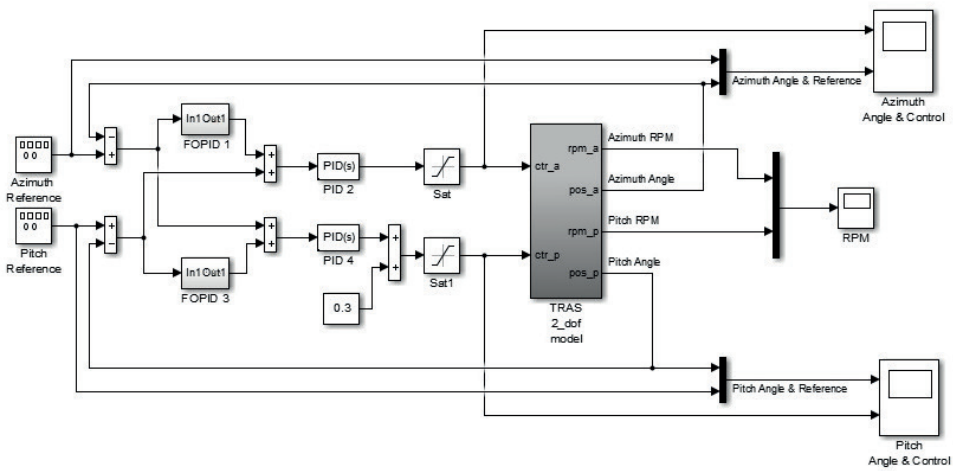

Fig. 2. Simulation model of the TRAS with cascade control system
Rys. 2. Model symulacyjny systemu TRAS z kaskadowym systemem sterowania

Tab. 1. Input parameters for GWO algorithm
Tab. 1. Parametry wejściowe dla algorytmu GWO

Parameters	Values
Number of wolves in herd	5
Maximum number of iterations	200

It should be assumed that:
e_a – difference between the set point and the actual value of the azimuth angle,
e_p – difference between the set point and the actual value of the pitch angle.

For the tail rotor, the primary controller is FOPID, to which the e_a signal is provided as input. The output value is summed with the signal e_p, which ultimately is the input for the secondary PID controller. The situation is analogous for the main rotor. FOPID again is the primary controller, but this time the input signal is e_p. The output value is summed with e_a, which then goes to the secondary PID controller.

For comparison purposes, an identical simulation was prepared in which the FOPID1 and FOPID3 blocks were replaced respectively by PID1 and PID3.

In order to determine the coefficients, the GWO algorithm was used, which needed input parameters (Table 1). It was also necessary to prepare initial range of the sought coefficients – as presented in (11).

$$\begin{aligned} K_P &= [0:50] \\ K_I &= [0:50] \\ K_D &= [0:50] \\ \lambda &= [-1:0] \\ \mu &= [0:1] \end{aligned} \tag{11}$$

Equation (12) was used as a cost function:

$$J = \int_0^\infty \left(\left| e_a(t) \right| + \left| e_p(t) \right| \right) dt \tag{12}$$

In all calls to the GWO algorithm, the main goal was to reduce the difference between the received values and the references – marked in blue on the graphs in the Figures 3 and 4.

6. Results of simulation

In this article the Two Rotor Aerodynamical System was simulated in two variants, both with cascade control system:
- Variant 1 – this is a version for comparison with four PID controllers,
- Variant 2 – version in which both primary controllers are FOPID, while PID act as secondary controllers.

The GWO algorithm was used to find the coefficients for the controllers. Their values are presented in Table 2.

Figure 3 shows the azimuth angle graph of the TRAS system. The blue line indicates the reference the system is trying to reach. The green color shows the behavior of variant 1 with only PID controllers, while the red color shows the response of the system containing the FOPID controllers. Variant 1 reaches the first setpoint slightly earlier. After that, both versions behave very similarly until the 20th second, when the

Tab. 2. Coefficients for all variants
Tab. 2. Współczynniki wszystkich wariantów

Variant 1	P	I	λ	D	μ
PID 1	213.75	0.96	—	148.89	—
PID 2	2593.34	0.93	—	0.64	—
PID 3	110.59	6.55	—	1.39	—
PID 4	57.01	34.38	—	86.85	—
Variant 2	**P**	**I**	**λ**	**D**	**μ**
FOPID 1	19.30	184.33	–0.0089	15.80	0.027
PID 2	739.63	2.23	—	610.29	—
FOPID 3	30.10	19.83	–0.17	1036.41	0.24
PID 4	16.51	25.74	—	10.62	—

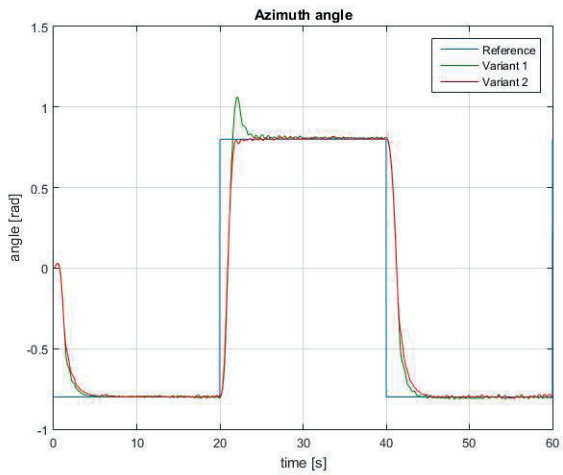

Fig. 3. TRAS - plot for azimuth angle response
Rys. 3. TRAS - wykres pomiaru kąta azymutowego

setpoint changes. Here are the biggest differences. For variant 1 overshoot can be observed. In variant 2 this phenomenon does not occur and it approaches the set value as dynamically as the version marked with green color. As a result, it takes variant 1 more than twice as long to reach the setpoint than for second control version (marked as red). Another difference becomes apparent after the 40th second, when the setpoint changes again. The differences are very similar to those at the very beginning of the chart. However, it is worth noting that for variant 1 there are slightly stronger oscillations around the reference.

The pitch angle behavior is illustrated by the Figure 4. The greatest differences between the actions of both variants are visible when the first maximum is reached. The behavior is very similar for about first 2 seconds. However, then it can be seen that variant 1 starts to steadily reach values above the set value. When reaching the maximum, the graph briefly drops below the set value and only then stabilizes. Variant 2 is approaching the reference already in the middle of the climb to the maximum. For most of the time both variants follow the setpoint very well. The only differences can be seen for a while after the 20th second and after reaching the second minimum. In the first case, the sudden change in the azimuth angle setpoint caused the problem. Variant 1 copes with such an issue worse – there are some deviations. On the other hand second control version keeps up with the reference. In the second case for variant 1, approximately 10 seconds have elapsed since the setpoint was changed abruptly for the azimuth angle. Therefore, it can be

Tab. 3. Values of cost function (12)
Tab. 3. Wartości funkcji kosztu (12)

	Cost Function
Variant 1	294.16
Variant 2	284.94

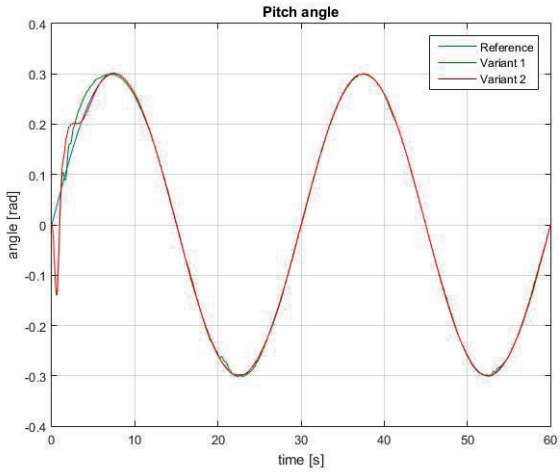

Fig. 4. TRAS – plot for pitch angle response
Rys. 4. TRAS – wykres pomiaru kąta nachylenia

concluded that option 1 is a bit worse at reaching the minimum, because also in the first case the deviations lasted after reaching the minimum.

The values of the cost function (12) are shown in table 3. Variant 2 (with FOPID controllers) achieved a much better result, which was already indicated by the previous charts.

Figures 5 and 6 respectively show the execution time of the controllers for variant 2 for the tail and main rotor control loops. Simulations were performed with fixed step 0.02 on a computer with Intel (R) Core (TM) i5-4460 CPU @ 3.20 GHz processor. It can be observed that the operation time of PID controllers is definitely shorter than that of FOPID controllers. The difference is one order of magnitude for both control loops for variant 2. Figure 7 shows the execution time of the entire control loops. It can be seen that the control for the main rotor is computed slightly faster than for the tail rotor.

Tab. 4. Average execution time of controllers and whole control loops
Tab. 4. Średni czas wykonania regulatorów i całych pętli sterowania

Controller/Control Loop	Average execution time
FOPID 1	1.6138e-07
PID 1	2.0591e-08
Tail Rotor control loop	3.3394e-07
FOPID 3	1.2268e-07
PID 4	2.0228e-08
Main Rotor control loop	1.8860e-07

Tab. 5. Maximum execution time of controllers and whole control loops
Tab. 5. Maksymalny czas wykonania regulatorów i całych pętli sterowania

Controller/Control Loop	Maximum execution time
FOPID 1	7.0250e-07
PID 1	3.2625e-07
Tail Rotor control loop	9.9125e-07
FOPID 3	5.3500e-07
PID 4	2.5375e-07
Main Rotor control loop	6.0750e-07

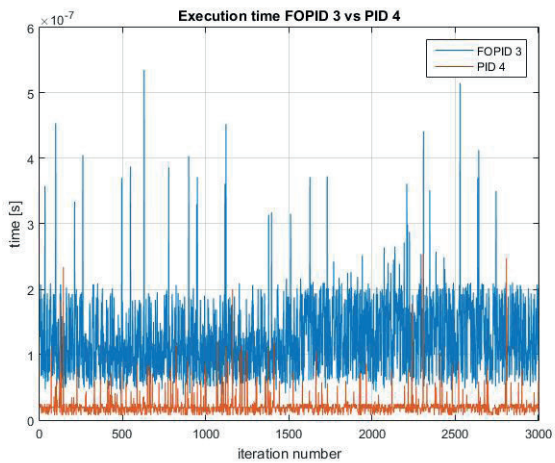

Fig. 6. Execution time: FOPID 3 vs PID 4
Rys. 6. Czas wykonania: FOPID 3 vs PID 4

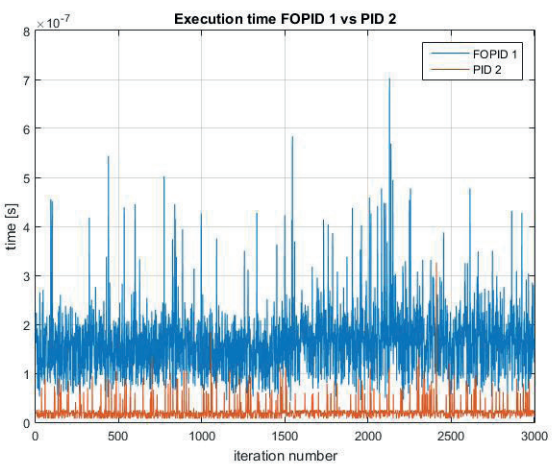

Fig. 5. Execution time: FOPID 1 vs PID 2
Rys. 5. Czas wykonania: FOPID 1 vs PID 2

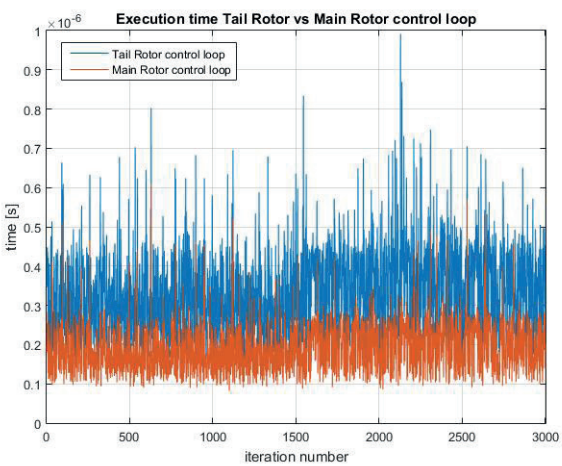

Fig. 7. Execution time of control loops: tail rotor vs main rotor
Rys. 7. Czas wykonania pętli sterujących: wirnik ogonowy vs główny wirnik

Tables 4 and 5 present average and maximum execution time values for variant 2. The table 4 confirms the observations from the 5–7 plots. However, in the case of the maximum values the differences between the individual controllers are not so large. In extreme cases PID controllers are only 2 times faster than FOPID. Nevertheless, all controllers perform their calculations fast enough to meet the requirements for real-time systems.

7. Conclusions

It can be concluded from the article that in the case of cascade control FOPID and PID controllers can cooperate with each other well enough that they achieve better results than the version with classic counterparts only. In the case of the pitch angle for the second control version it was even possible to remove the deviations caused by the sudden change of the azimuth angle setpoint. Coefficients of individual controllers were determined by the GWO algorithm. Many simulations were carried out to find the best values. Additionally, the execution times of all controllers were measured. As expected, the classic versions were definitely faster. However, in the case of the maximum values the differences were smaller. Ultimately both controllers met the requirements of real-time systems.

The author plans to conduct experiments on the TRAS system from INTECO in order to be able to confront the conclusions of the simulations with reality. Furthermore, it would be worthwhile to also perform comparisons with PI-D controllers.

Bibliography

1. Khosravi M., Enayati S., Saeedi H., Pishro-Nik H., *Multi--Purpose Drones for Coverage and Transport Applications*, "IEEE Transactions on Wireless Communications", Vol. 20, No. 6, 2021, 3974–3987, DOI: 10.1109/TWC.2021.3054748.
2. Nožica D., Blažević D., Keser T., *Unmanned Aerial Vehicle Swarm Uses Wi-Fi to Search for Stranded People in Remote Areas Embedded devices as active scanners in search for Wi-Fi-enabled mobile phones*, 10th Mediterranean Conference on Embedded Computing (MECO), 2021, DOI: 10.1109/MECO52532.2021.9460147.
3. Ahmad M., Ali A., Choudhry M.A., *Fixed-Structure H_∞ Controller Design for Two-Rotor Aerodynamical System* (*TRAS*), "Arabian Journal for Science and Engineering", Vol. 41, No. 9, 2016, 3619–3630, DOI: 10.1007/s13369-016-2232-1.
4. Almtireen N., Elmoaqet H., Ryalat M., *Linearized Modelling and Control for a Twin Rotor System*, "Automatic Control and Computer Sciences", Vol. 52, 2018, 539–551, DOI: 10.3103/S0146411618060020.
5. Ajangnay M.O., *Optimal PID controller parameters for vector control of induction motors*, SPEEDAM 2010, 959–965, DOI: 10.1109/SPEEDAM.2010.5545043.
6. Kaijuka P., Dixon R., Ward C.P., Dutta S., *LQR Control Applied to a Novel Track Switch Actuator*, UKACC 12th International Conference on Control (CONTROL), 2018, 385–385, DOI: 10.1109/CONTROL.2018.8516839.
7. Zeilinger M.N., Jones C.N., Raimondo D.M., Morari M., *Real-time MPC – Stability through robust MPC design*, Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 2009, 3980–3986, DOI: 10.1109/CDC.2009.5400903.
8. Sekine S., Nishimura M., *Application of fuzzy neural network control to automatic train operation*, Proceedings of 1995 IEEE International Conference on Fuzzy Systems, Vol. 5, 1995, 39–40, DOI: 10.1109/FUZZY.1995.409917.
9. Bennett S., *Development of the PID controller*, "IEEE Control Systems Magazine", Vol. 13, No. 6, 1993, 58–62, DOI: 10.1109/37.248006.
10. Zhai Q., Xia X., Feng S., Huang M., *Optimization Design of LQR Controller Based on Improved Whale Optimization Algorithm*, 2020 3rd International Conference on Information and Computer Technologies (ICICT), 2020, 380–384, DOI: 10.1109/ICICT50521.2020.00067.
11. Sarangi S.K., Panda R., Priyadarshini S., Sarangi A., *A new modified firefly algorithm for function optimization*, 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, 2944–2949, DOI: 10.1109/ICEEOT.2016.7755239.
12. Singh K., Singh A.V., Khatri S.K., *Advanced Gameplay Strategy Based on Grey Wolf Optimization*, 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, 183–185, DOI: 10.1109/ISCON47742.2019.9036295.
13. Katugampola U., *A New Approach to Generalized Fractional Derivatives*, "Bulletin of Mathematical Analysis and Applications", Vol. 6, No. 4, 2014, 1–15, DOI: 10.48550/arXiv.1106.0965.
14. Oustaloup A., Levron F., Mathieu B., Nanot F.M., *Frequency-band complex noninteger differentiator: characterization and synthesis*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 47, No. 1, 2000, 25–39, DOI: 10.1109/81.817385.
15. Anbumani K., Ranihemamalini R., Pechinathan G., *GWO based tuning of PID controller for a heat exchanger process*, Third International Conference on Sensing, Signal Processing and Security (ICSSS), 2017, DOI: 10.1109/SSPS.2017.8071631.
16. INTECO *Two Rotor Aerodynamical System*, User's Manual.

Symulacja sterowania kaskadowego dla dwurotorowego systemu aerodynamicznego z wykorzystaniem regulatorów FOPID oraz PID

Streszczenie: Dwurotorowy system aerodynamiczny to nieliniowy system, w którym występuje sprzężenie krzyżowe. W przypadku tego układu poszczególne wirniki wpływają na obie mierzone wartości: kąt azymutowy i kąt nachylenia. W związku z tym obie wartości muszą być używane w każdej pętli sprzężenia zwrotnego. Umożliwia to realizację kaskadowego układu sterowania. Zdecydowano się na zastosowanie FOPID jako regulatorów nadrzędnych i PID jako regulatorów podrzędnych. W celach porównawczych przygotowano analogiczny układ regulacji z samymi PID. Współczynniki wyznaczono za pomocą algorytmu Grey Wolf Optimizer (GWO). Wszystkie symulacje wykonano w środowisku MATLAB/Simulink. Oprócz porównania wartości funkcji kosztu sprawdzono również czas wykonania poszczególnych regulatorów.

Słowa kluczowe: FOPID, PID, TRAS, dwurotorowy system aerodynamiczny, optymalizacja, algorytm GWO, aproksymacja ORA, czas wykonania

Jakub Żegleń-Włodarczyk, PhD

zeglen@agh.edu.pl
ORCID: 000-0003-1857-1461

He obtained MSc in Automatics and Robotics in 2018 and PhD in Automatics and Robotics in 2024. Currently working at AGH University of Krakow in Department of Automatics and Robotics. Speaker at many international conferences, such as MMAR, Automation, ICFDA and PCC. His articles have been published in prestigious scientific journals. His research interests include PLCs, fractional control and fractional order controllers, especially FOPID. Since 2016 he has been gaining professional experience in international companies. In his free time he expands his knowledge with information on innovative technological solutions in motor sports, such as Formula 1 or World Endurance Championship.