Optimization pitch angle controller of rocket system using improved differential evolution algorithm

Dwi Lastomo[a, 1,*], Herlambang Setiadi[b, 2], Muhammad Ruswandi Djalal[c, 3]

[a] UPMB, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[b] School of Information Technology & Electrical Engineering, The University of Queensland, Brisbane, Australia
[c] Department of Mechanical Engineering, State Polytechnic of Ujung Pandang, Makassar, Indonesia
[1] dtomo23@gmail.com*; [2] h.setiadi@uq.edu.au; [3] wandi@poliupg.ac.id
* corresponding author

ARTICLE INFO

ABSTRACT

Pitch angle of rocket system is the important parts of the rocket. This part corresponds to the movement of the rocket system. Rocket system is fell into multi-input and multi-output (MIMO) system. The most challenge factor in MIMO system is designing the controller, if the design is not appropriate, it may lead to the unstable condition. Hence, appropriate and robust control design is inevitable. This paper introduces PID controller as pitch angle control of rocket system. Furthermore, PID controller parameter is optimized using improved differential evolution algorithm (IDEA). To analyze the performance of rocket system, time domain simulation is implemented. From the simulation result, it is found by using PID controller based on IDEA, the performance of pitch angle of rocket system is enhanced.

I. Introduction

The development of multi-input and multi-output (MIMO) systems increases significantly in last few decade. One of the types of MIMO input that becomes popular is rocket system. Rocket system was first used in early 1232 when the Chinese used an uncontrolled missile to defending city of Peiping from the Mongol onslaught [1]. Since then, the application of rocket system has been growing fast and a lot of improvement has been made such as controlling the speed and the high [1]. In general, a rocket system comprising of rides, propulsion, propellant, aerodynamics, trajectories, guidance, controller and telemetry [2]. For the rocket system to operate properly, each subsystem must work synergistically to achieve the expected performance [2]. The rocket movement control purpose is to allow the rocket to move according to the target [2]. Broadly speaking the rocket's movement is determined by the rocket fins [2]. The fins controller is corresponded to pitch fins controller, yaw fins controller, and rolling fins controller. One of the most important is pitch fins controller or pitch angle controller of the rocket system.

PID controller is the most formidable controller in the last few decade. PID controller can provide a fast response with smaller overshoot. However, due to increasing development of technology lead to increasing complexity of the system. It might worse when the system is MIMO system such as rocket system. Hence, intelligent controller design based on a nature-inspired algorithm or metaheuristic algorithm can be the solution to handle the complexity of MIMO system.

Metaheuristic algorithm can be classified into three group: biological inspiration, physical inspiration and social inspiration [3]. The example of social inspiration is tabu search algorithm and imperialist competitive algorithm [4]. Simulated annealing algorithm is one of the examples of physical inspiration. The most commonly used is a biological inspiration, there is some algorithm based on biological inspiration such as ant colony optimization [5], particle swarm optimization [6, 7], artificial bee colony [8] and differential evolution algorithm [9]. Differential evolution algorithm itself has solved tons of optimization problems. However, there is some handicap of the differential

evolution algorithm in term of the mutation. Hence, the improved approach of differential evolution algorithm can be the solution to address the mutation problems.

This paper introduces the application of PID controller as pitch angle controller of the rocket system. To find the best coordination, the PID controller is tuned using improved differential evolution algorithm. The rest of the paper is organized as follows: Section II briefly explain dynamic modeling of rocket system and PID controller. Differential evolution algorithm, improved differential evolution algorithm, and tuning procedure are given in section III. Section IV provides the result and discussion of investigated system. Finally, Section V highlights the conclusions, contribution, and future direction.

II. Fundamental Theory

A. Dynamic Model of Rocket

A dynamic representation of rocket can be represented as 4 order system. The state space representation of rocket movement dynamic is stated in (1) [2].

$$
\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & U_0 & 0 \\ M_u + M_{\dot{w}}X_u & M_w + M_{\dot{w}}Z_w & M_q + M_{\dot{w}}U_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} + M_w Z_{\delta e} \\ 0 \end{bmatrix} \delta_e \qquad (1)
$$

B. PID Controller

The most important part of the system is the controller. The controller function is to reduce error signal caused by interference. If the controller cannot deal with the error signal, it might lead to deteriorated performance and unstable condition of the system. Hence an appropriate and robust controller is crucial. Proportional, integral and derivative is one of the most popular controllers among industrial practitioners [10-13].

The proportional controller output is related to the product gain and input signal Fig. 1 shows the block diagram of the relationship between the magnitude of the error and the magnitude of proportional controller output. The error signal represents the difference between the reference signal and the feedback signal from the system [10, 13].
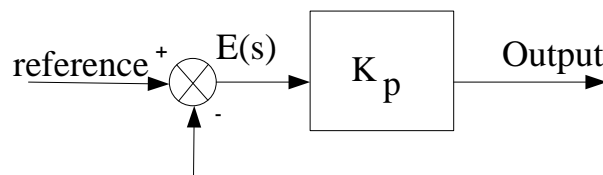

Fig. 1. Block diagram of proportional controller [10, 13].

The purpose of the integral controller is to produce a system response with zero steady state error. With an integral controller, the system response can be enhanced due to solid-state error to zero from the integrator. Fig. 2 shows the block diagram of the relationship between the error signal and the magnitude of integral output. The error signal consists of the reference signal and the actual signal [10, 13].
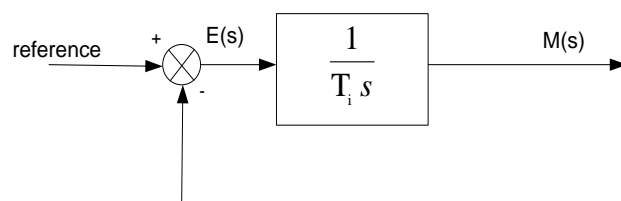

Fig. 2. Block diagram of integral controller [10, 13].

Derivative controller output has the same properties as derivatives operations. When the input does not change, the output of the controller also does not change, whereas if the input signal changes suddenly and ascending (step function), the output produces an impulse-shaped signal. If the input signal changes slowly (ramp function), the output is a large step function corresponded with the rising speed of the ramp function and the differential constant factor T_d. Fig. 3 illustrates the block diagram of the relationship between the error signals to the controller output [10, 13].
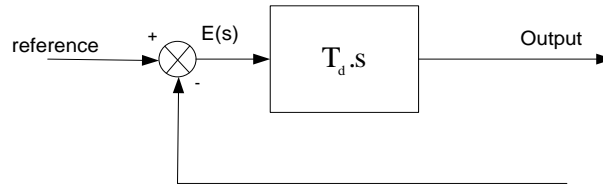

Fig. 3. Block diagram of the integral controller.

Each of the disadvantages and advantages of the proportional, integral and derivative controller can be handled by combining all three in parallel connection. The purpose of PID controller is to produce large initial changes, accelerate the reaction system and eliminating offset. Fig. 4 depictions a block diagram of the PID controller [10-14].
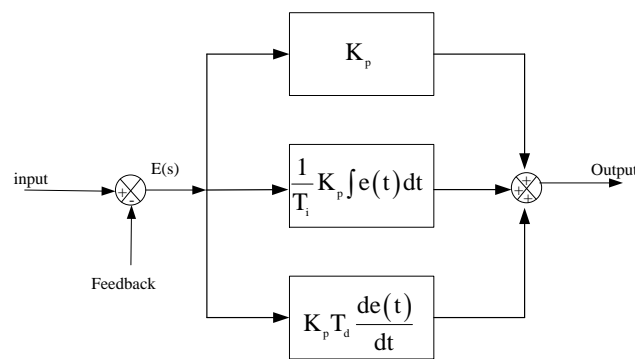

Fig. 4. Block diagram of PID controller [10-14].

III. Methodology

A. Differential Evolution Algorithm (DEA)

Differential evolution algorithm (DEA) is soft computing technique based on recurrence cycle of recombination and selection to lead the population into global optimum values. This algorithm first introduces by Storn and Prince in 1995 [15]. DEA as several steps for finding optimal value namely initialization, mutation, recombination, crossover and selection.

The initialization process starts by putting a random number in a 2 dimension area with lower and upper limit boundary. Each initial vector value is generated irregularly on the boundary. The population initialization can be described using (2) [16].

$$x_{j,i,g} = rand_j(0,1) \times (b_{j,u} - b_{j,l}) + b_{j,l} \tag{2}$$

Where $b_{j,u}$ and $b_{j,l}$ are upper and lower limit of vector j. While $x_{j,i,g}$ is vector value of I on j parameter and g generation. $rand_j(0,1)$ is random value between 0 and 1.

After going through the initialization process, the next steps are mutation and recombination process. The DEA mutation and recombination is performed by combining the differential vectors of the first and second vector randomly choose with the third vector to obtain a mutant vector. This process can be described using (3) [16].

$$v_{i,g} = x_{r0} + F \times (x_{rl,g} - x_{r2,g}) \tag{3}$$

Where $v_{i,g}$ and F are mutant vector and real number with ranges 0 to 1 respectively. While $x_{rl,g}$, $x_{r2,g}$ and x_{r0} are a chosen random vector. In DEA, crossovers are used to complement the search strategy for differential mutation. The function of crossover is forming a trial vector of the duplicated parameter values of two distinct vector. This process can be evaluated using (4) [16].

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} \ if \ (rand_j(0,1) < Cr \ or \ j = j_{rand}) \\ another \ x_{j,i,g} \end{cases} \tag{4}$$

The limit of Cr's crossover probability value is 0 to 1 and Cr is the value set by the user to control the division of the duplicated parameter values from mutant. $rand_j$ is a random value that will give a decision of a vector, whether the vector needs to be through a crossover process or not. If the value of the $rand_j$ vector (0, 1) is less than Cr, then the value of the vector to be duplicated in the trial vector is the mutant vector. If the process goes the opposite of the $rand_j$ value over Cr then the vector value to be duplicated in the trial vector is the initial vector [16].

The next process is selection, the purpose of selection is to determine the vector that will become a member of the population for the next iteration process. If trial vector $u_{i,g}$ has an objective function value equal to or smaller than the target vector $x_{i,g}$ then during the next iteration it will replace the target vector. If the process is reversed, trial vector $u_{i,g}$ has an objective function value greater than the target vector $x_{i,g}$, then in the next process the target vector remains a member [16].

B. Improved Differential Evolution Algorithm (IDEA)

The expectation of Improved Differential Evolution Algorithm (IDEA) is to obtain better optimization result. The IDEA focus is on the mutation process done more based on the increasing particle displacement restrictions along with the increase in the number of iterations, so IDEA particle velocity each iteration decreases and gradually the particle shift tends to decrease [17]. DEA focuses on the great value of particle displacement per iteration so that the value of particle velocity that occurs tends to remain. Particle velocity in the DEA process tends to produce a static particle gap still. Optimum global and local optimum DEA can differentiate as well as apply the best fitness value on each iteration [17]. Assuming that a unit of the best particle in an iteration ever moves from a particular particle in an iteration ever moves from a particular particle to another particle. If the optimum global position and the optimum local position are not complicated, then this not a problem. But if the system parameters to be tested quite a lot, where it complicates the global position of optimum and local optimum on the constraints of multi-parameters, another problem will emerge [17].

The problem can be solved by applying a velocity particle size constraint progressively, where there is a regulation at the particle gap is in the area that has the potential to become global optimum. This idea has already been implemented by IDEA with the aim of manipulating the mutation process in such a way. The mathematical model of the mutation process can be explained using (5) and 6 [17].

$$v_{i,g}(r) = x_{best}(r) \times \lambda + x_{r0,g}(r) \times (\lambda - 1) + F \times \left(x_{r1,g}(r) - x_{r2,g}(r)\right) \tag{5}$$

$$\lambda = \frac{iter \ max - r}{iter \ max} \tag{6}$$

Where r is an iteration, as the iteration progresses the lambda value will decrease. The purpose of this process is the best solution $x_{best}(r)$ to have more influent role than random solutions $(x_{r0,g}(r))$. The role of the random solution increases with the increasing iteration value ad as the role diminishes best solution $(x_{best}(r))$. Improvement of this algorithm aims is to make the balance between exploration and exploitation [17].

C. Tuning Pitch Angle Controller using IDEA

The pitch angle controller of rocket system is PID controller. The IDEA is used to optimized the parameter of PID controller which are gain constant (*Kp*), integral constant (*Ki*), derivative constant (*Kd*) and filter constant (*Kfilter*). Integral of absolute error (IAE), integral of squared-error (ISE) and integral of time-weighted-squared-error (ISSE) are a performance index criterion that is commonly used to design a controller on a system [18, 19]. Three performance index criteria have their respective strengths and weaknesses [18, 19]. The weakness of the IAE and ISE is that the performance index can generate a response with relatively small overshoot but steady-state time is longer due to weighting for all errors and independent of time [18, 19]. Although the ITSE performance index criteria can overcome the weaknesses of the ISE performance index criteria but cannot be ascertained as meeting the desired stability limits [18, 19]. Hence, integral time multiply by absolute error (ITAE) is used as index criteria to evaluate the performance of the proposed method. The mathematical representation of ITAE can be described using (7) while the block diagram of optimization scheme is illustrated in Fig. 5 [19].

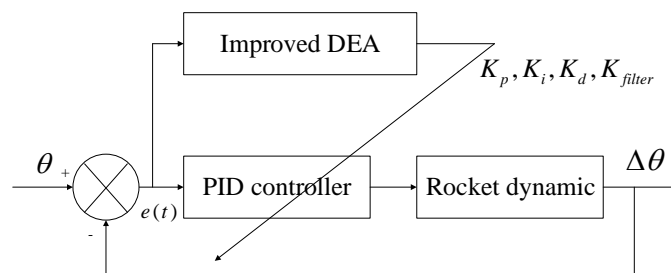$$ITAE = \sum \int_0^{t1} t|\Delta theta(t, X)|dt \qquad (7)$$

Fig. 5. Block diagram of the optimized method.

IV. Results and Discussions

A fourth-order dynamic rocket is chosen as the test system. The case study was carried on in MATLAB/SIMULINK environment. PID controller was chosen as the pitch angle control of the rocket system. IDEA was then used to optimize a parameter of PID controller. The parameter of rocket system is shown in Table 1. While parameter of IDEA used in this paper for various simulations was illustrated in Table 2. Fig. 6 depiction the convergence curves of IDEA and it was clearly shown that after 39 iterations, IDEA found the convergence value. Furthermore, Table 3 shows the optimized parameter of PID controller.

Table 1. Rocket system parameters.

Parameter	Value	Parameter	Value	Parameter	Value
X_u	-0.101	Z_u	-0.2731	U_0	170
Z_w	-1.4552	$Z_{\delta e}$	-16.7885	g	9.81
M_q	-0.6255	M_u	0		
M_w	-3.5227	$M_{\delta e}$	-71.1916		
$M_{\dot{w}}$	-0.0088	$X_{\delta e}$	0		

Table 2. IDEA parameters

Parameter	Value
CR	0.8
Number of population	40
F	7
Number of optimized parameters	4
Max iteration	50
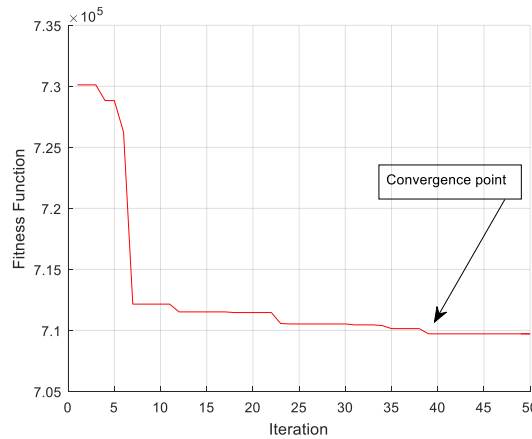
Fig. 6. Convergence graph of IDEA

Table 3. Optimized parameters

Parameter	Value
Kp	1.9922
Ki	14.975
Kd	-0.626
Kfilter	2.0591

To analyze the performance of the rocket system, time domain simulation was carried out by giving 45-degree step input as the reference. In this section, the comparison between rocket system with conventional PID controller and optimal PID controller based on IDEA were considered. Fig 7 shows the oscillatory condition of rocket system while Table 4 illustrates the overshoot and settling time of the case study. It was clearly shown from Fig. 7 and Table 4 that optimal PID based on IDEA had the better oscillatory condition indicated by smaller overshoot and fastest settling time. Even thought, the oscillatory condition of system with optimal PID controller did not increased significantly, however on the rocket system, small movement enhancement are essential to reduce movement deviation of the rocket and the precision of the rocket when fall to the target.

Table 4. Overshoot and settling time of investigated system.

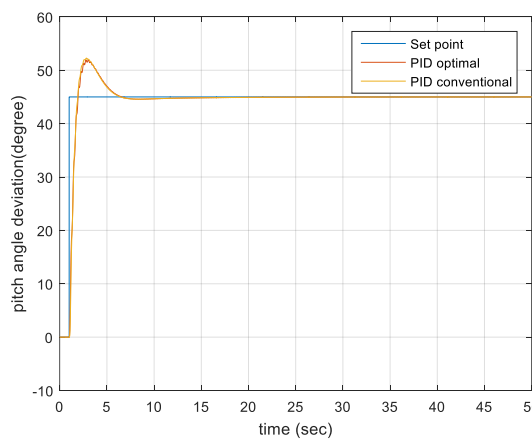|	Conventional PID	Optimal PID
Overshoot (degree)	52.29	51.97
Settling time (s)	20	12

Fig. 7. The oscillatory condition of pitch angle rocket system.

Table 5 shows the comparison between the proposed method with existing system. From the table it was shown that the proposed method could enhance the performance significantly indicated by smallert overshoot and fastest settling time. The proposed method could reduce the overshoot and settling time around 10 degree and 68 second

Table 5. Overshoot and settling time of investigated system.

	Existing system	Optimal PID
Overshoot (degree)	62	51.97
Settling time (s)	80	12

V. Conclusions

This paper presents the intelligent approach design based on IDEA to optimized pitch angle controller of the rocket system. From the investigated case study, it was found that IDEA can enhance the oscillatory condition of rocket angle indicated by smaller overshoot and fastest settling time of the rocket angle deviation.

Acknowledgment

References

[1] M. S. George, "Missile guidance and control systems," *New,* vol. 1, pp. 61-62, 2004.

[2] F. Mangkusasmito, W. Wahyudi, and B. Setiyono, "Sistem kendali posisi sudut angguk untuk roket RKX-300 dengan metode kendali linear quadratic regulator (LQR) dan pole placement," *TRANSMISI,* vol. 15, pp. 99-106, 2013.

[3] H. Setiadi and K. O. Jones, "Power System Design using Firefly Algorithm for Dynamic Stability Enhancement," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 1, pp. 446-455, 2016.

[4] M. R. Djalal, A. Imran, and I. Robandi, "Optimal placement and tuning power system stabilizer using Participation Factor and Imperialist Competitive Algorithm in 150 kV South of Sulawesi system," in *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 147-152.

[5] B. P. Hadhi, H. Setiadi, and I. Robandi, "Optimisasi Pengaturan Frekuensi Sistem Interarea Menggunakan Algoritma Particle Swarm Optimization (PSO) dan Ant Colony Optimization (ACO)," in *The 14th Seminar on Intelligent Technology And Its Applications (SITIA 2013)*, Surabaya Indonesia, 2013, pp. 120-123.

[6] A. T. Al-Awami, Y. Abdel-Magid, and M. Abido, "A particle-swarm-based approach of power system stability enhancement with unified power flow controller," *International Journal of Electrical Power & Energy Systems,* vol. 29, pp. 251-259, 2007.

[7] A. Thobirin and I. T. R. Yanto, "Automatic differentiation based for particle swarm optimization Steepest descent direction," *International Journal of Advances in Intelligent Informatics,* vol. 1, pp. 90-97, 2015.

[8] F. S. Abu-Mouti and M. E. El-Hawary, "Optimal Distributed Generation Allocation and Sizing in Distribution Systems via Artificial Bee Colony Algorithm," *IEEE Transactions on Power Delivery,* vol. 26, pp. 2090-2101, 2011.

[9] A. Sharma and B. K. Panigrahi, "Framework arrangement of directional relays in meshed networks based on differential evolution algorithm," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*, 2016, pp. 1-6.

[10] M. A. Fassamsi, "Optimal Tuning PID Superconducting Magnetic Energy Storage (SMES) menggunakan Imperialist Competitive Algorithm (ICA) untuk meredam osilasi daya pada sistem kelistrikan Jawa Bali 500 kV," Undergraduate thesis, Department of Electrical Engineering, Sepuluh Nopember Institute Technology Sepuluh Nopember, Surabaya, 2011.

[11] H Setiadi, W. K. Kautsar, A Swandaru, and I Robandi, "Optimal Tunning PID Controler for Inter Area Using Imperialist Competitive Algorthm (ICA)," in *International Seminar on Applied Technology, Science, and Arts 2011*, Surabaya, 2011.

[12] R. W. W. Atmaja, S. Herlambang, and I. Robandi, "Optimal Design of PID Controller in Interconnected Load Frequency Control using Hybrid Differential Evolution–Particle Swarm Optimization Algorithm," in *Seminar on Intelligent Technology and Its Applications 2014*, 2014.

[13] D. Lastomo, H. Setiadi, and M. R. Djalal, "Design Controller of Pendulum System using Imperialist Competitive Algorithm," *Jurnal Intek Politeknik Negeri Ujung Pandang,* vol. 4, pp. 53-59, 2017.

[14] C. Pradipta, F. Arazzi, H. Setiadi, and I. Robandi, "Koordinasi Kontroler PID dan Thyristor Controlled Phase Shifter (TCPS) Pada Load Frequency Control (LFC) menggunakan Imperialist Competitive Algorithm (ICA)," in *SITIA 2013-Smart and Intelligent Technology Application for Distributed Generation Systems*, 2014.

[15] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[16] J. H. Van Sickel, K. Y. Lee, and J. S. Heo, "Differential evolution and its applications to power plant control," in *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, 2007, pp. 1-6.

[17] M. Nayak, K. Krishnanand, and P. Rout, "Modified differential evolution optimization algorithm for multi-constraint optimal power flow," in *Energy, Automation, and Signal (ICEAS), 2011 International Conference on*, 2011, pp. 1-7.

[18] R. Krohling, H. Jaschek, and J. Rey, "Designing PI/PID controllers for a motion control system based on genetic algorithms," in *Intelligent Control, 1997. Proceedings of the 1997 IEEE International Symposium on*, 1997, pp. 125-130.

[19] M. Pujiantara and M. Abdillah, "Adaptive pi controller berbasis bat algorithm (ba) dan extreme learning machine (elm) untuk kontrol kecepatan permanent magnet synchronous motor (pmsm)," *JAVA Journal of Electrical and Electronics Engineering,* vol. 9, 2011.