EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Fixed-structure heading-autopilot controller design using meta-heuristics

Nattapong Ruenruedeepan[1)], Pakin Champasak[2)], Natee Panagant[1)], Nantiwat Pholdee[1)] and Sujin Bureerat*[1)]

[1)]Sustainable and Infrastructure Research and Development Center, Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
[2)]Department of Mechanical and Aerospace Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok 10800, Thailand

Received 20 July 2023
Revised 16 October 2023
Accepted 8 November 2023

Abstract

This work presents an alternative efficient means to synthesise a fixed-structure autopilot controller that is both robust and optimal using meta-heuristics (MHs). The problem is aimed at finding controllers in several sections with the objective of minimising integral square error, subject to several constraints to ensure a robust, precise, and rapid reference tracking control system. An optimum control problem was posed while several MHs were employed to solve the problem, and their performances were investigated. Based on the results, a Self-Adaptive Differential Evolution (JADE) was found to be the most efficient algorithm. The study presents a simple but effective tool for designing a robust and optimum autopilot flight controller. It also explores the performance of several MHs in the new optimisation design field of robust and optimal flight control systems.

Keywords: Evolutionary algorithms, Flight dynamics and control, Multidisciplinary optimisation, Fixed-structure control, Robust and optimum control

1. Introduction

Aviation has become an integral part of various aspects of human life, which holds immense potential to evolve into the field of automatic Unmanned Aerial Vehicles (UAVs) in the near future for a myriad of applications, including agriculture [1], mapping [2], military operations [3], surveying [4], and more[5]. As UAVs can operate without pilots, they are a safer choice for hazardous conditions. Consequently, research on UAV design has gained immense popularity worldwide, with trajectory/path optimisation being one of the most sought-after research topics to ensure optimum mission conditions. Since UAVs do not require a pilot on board, the control system plays a crucial role in their performance, requiring high trajectory/path tracking efficiency, robustness against uncertainties, and disturbances throughout their flight envelope [6, 7].

Flight control is typically divided into two loops: the inner loop and the outer loop. The outer loop, or autopilot, guides the UAV along a predefined trajectory, ensuring mission accuracy and efficiency [8, 9]. Meanwhile, the inner loop controls the UAV's actuators to stabilize the vehicle and precisely track reference signals with high speed and accuracy [10, 11]. When designing the UAV control system, the inner loop, being the fastest loop, is conventionally designed first, followed by the outer loop, which is slower. Control design techniques can vary from classical methods such as root locus [12] and loop shaping [13] to more modern approaches like Linear-Quadratic Regulator (LQR) [14], Linear-Quadratic-Gaussian [15, 16], H-infinity control [17, 18], and Mu-synthesis [19]. Among these techniques, H-infinity and Mu-synthesis are considered the most advanced, as they can optimise performance while providing robustness against disturbances and uncertainties. However, a major challenge of H-infinity and Mu-synthesis is that all requirements must be transformed into weighting factors, leading to an increasing number of factors as the requirements increase. Moreover, tuning these weighting factors requires experience, and the process can be somewhat intuitive [20]. Traditional H-infinity synthesis does not allow for the imposition of structure on the controller, often resulting in controllers with high-order dynamics that are difficult to map to specific real-world control architectures.

On the other hand, fixed-structure or fixed-order controllers are sometime considered more practical due to their straightforward implementation and ease of interpretation [21, 22]. However, synthesising such a controller requires optimisation tools, and the controller synthesis optimisation problem is a non-convex constrained optimisation problem [23, 24]. Although non-smooth optimisation techniques [25] can deal with such a problem and are popular, they might be inefficient when dealing with several constraints due to the complexity of gradient approximation, and they might get trapped in local optima, particularly for achieving optimal performance while also providing robustness against disturbances and uncertainties. Therefore, metaheuristic algorithms, also known as evolutionary algorithms, have been recently considered as an alternative solver [22], such as genetic algorithm (GA) [26, 27], particle swarm algorithm (PSO) [28, 29], and differential evolutionary algorithm (DE) [24]. Compared with gradient-based optimisers, metaheuristics have the advantage of not requiring function derivatives, enabling them to deal with any kind of optimisation problem. However, they have some drawbacks, including a slow convergence rate and a lack of search consistency. Therefore, numerous metaheuristics have been developed over the last decade [30-39] while, to our best knowledge, only some classical

*Corresponding author.
Email address: sujbur@kku.ac.th
doi: 10.14456/easr.2024.4

metaheuristics [26, 27, 29] have been applied to the problem of fixed-structure controller design. In this regard, exploring the developed metaheuristics for the problem of fixed-structure controller design, such as heading autopilot, is an interesting topic. Moreover, investigating the performance of several metaheuristics might contribute to further research on the applications of metaheuristics for the problem of fixed-structure controller design.

This study introduces applications of MHs for a fixed-structure autopilot controller, considering optimum robustness, precision, and rapid reference tracking requirements. An optimisation problem is developed whilst being subject to a number of constraints in order to achieve a precise and rapid reference tracking controller for autopilot flight control synthesis. Several meta-heuristics are utilised to solve this optimisation problem while their performance in solving the autopilot flight control synthesis problem is investigated. The remainder of this paper is organised as follows: Section 2 outlines the formulation of the heading control autopilot optimisation problem, Section 3 describes the numerical experiment, Section 4 presents the results and discussion, and Section 5 provides the conclusion.

2. Formulation of an optimisation problem for UAV heading autopilot

2.1 Optimization problem

An optimization problem represents a unique class of mathematical challenges where the goal is to identify a specific set of design variables that optimize an objective function while meeting a set of predefined constraints. This problem type is pervasive in various fields, including engineering, economics, and data science, as it enables the efficient allocation of resources, the enhancement of processes, and the discovery of optimal solutions.

The mathematical formulation of an optimization problem can be expressed as follows:

$Min\text{: } or\ Max\text{: } f(\boldsymbol{x})$

Subjected to

$g_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, n_g$

$h_i(\boldsymbol{x}) = 0, i = 1, \ldots, n_h$

$\boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}$

In this context, the following mathematical notations are generally used:

- The vector $\boldsymbol{x} = [x_1, \ldots, x_{nv}]^T$ represents a set of design variables.
- The function $f(\boldsymbol{x})$ represents the objective function, which we aim to either minimize or maximize. The $g_i(\boldsymbol{x})$ represents the i^{th} inequality constraint function, ensuring that specific conditions are satisfied
- The $h_i(\boldsymbol{x})$ represents the i^{th} equality constraint function, which must hold true.
- The vectors $\boldsymbol{l_b}$ and $\boldsymbol{u_b}$ correspond to lower and upper bounds for the design variables.

These elements collectively define the framework of an optimization problem, which seeks to discover the most favorable configuration of design variables $\boldsymbol{x}$ that achieves the desired objective, while complying with both inequality and equality constraints.

2.2 UAV heading autopilot controller model

The prototype UAV used in this study was designed in our laboratory, as illustrated in Figure 1. Its specifications are shown in Table 1.

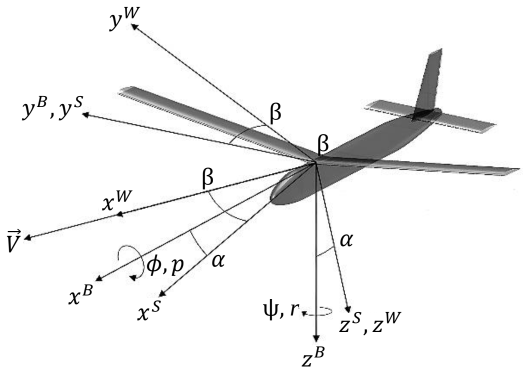


Figure 1 The UAV used in this study

Table 1 The UAV specification

Variable	Value	Variable	Value
wingspan (m)	2.25025	Horizontal tail incidence (deg)	-0.2841
wing aspect ratio	14	Horizontal tail surfaces (m^2)	0.10075
wing taper ratio	1	Horizontal tail incidence (deg)	-0.2841
wing dihedral (deg)	7.61167	Vertical tail aspect ratio	1.88389
wing incidence (deg)	3.9877	Vertical tail taper ratio	0.6
wing translation in x axis (m)	0.22638	Vertical tail surface (m^2)	0.05043
wing sweep angle (deg)	0	fuel CG placement in x axis (m)	0.09334
Horizontal tail aspect ratio	4.47037	designed angle of attack (deg)	2.97251
Horizontal tail taper ratio	1	fuselage length (m)	1.5
Horizontal tail dihedral (deg)	0		

The lateral linear state space model of the UAV can be obtained using Equation (1). The lateral linear state space model includes five state variables: sideslip angle, roll rate, roll angle, yaw rate and yaw angle, denoted as $\mathbf{x} = [\beta, p, \phi, r, \psi]^T$, while the control inputs are aileron deflection and rudder deflection, denoted as, $\mathbf{u} = [\delta_a, \delta_r]^T$. The heading autopilot block diagram for the proposed UAV is shown in Figure 2.

$$\begin{Bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{\phi} \\ \dot{r} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} Y_v & \dfrac{Y_p}{u_0} & \dfrac{g_0 \cos \Theta_0}{u_0} & \dfrac{Y_r - u_0}{u_0} & 0 \\ \left(\dfrac{L_v + i_x N_v}{1 - i_x i_z}\right) u_0 & \dfrac{L_p + i_x N_p}{1 - i_x i_z} & 0 & \dfrac{L_r + i_x N_r}{1 - i_x i_z} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \left(\dfrac{N_v + i_z L_v}{1 - i_x i_z}\right) u_0 & \dfrac{N_p + i_z L_p}{1 - i_x i_z} & 0 & \dfrac{N_r + i_z L_r}{1 - i_x i_z} & 0 \\ 0 & 0 & 0 & 1 & 0 \end{Bmatrix} \begin{Bmatrix} \beta \\ p \\ \phi \\ r \\ \psi \end{Bmatrix} + \begin{bmatrix} \dfrac{Y_{\delta_a}}{u_0} & \dfrac{Y_{\delta_r}}{u_0} \\ \dfrac{L_{\delta_a} + i_x N_{\delta_a}}{1 - i_x i_z} & \dfrac{L_{\delta_r} + i_x N_{\delta_r}}{1 - i_x i_z} \\ 0 & 0 \\ \dfrac{N_{\delta_a} + i_z L_{\delta_a}}{1 - i_x i_z} & \dfrac{N_{\delta_r} + i_z L_{\delta_r}}{1 - i_x i_z} \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta_a \\ \delta_r \end{Bmatrix} \quad (1)$$

where

$$\mathbf{A} = \begin{bmatrix} -0.121 & -0.009 & 0.656 & -0.990 & 0 \\ -2526.655 & -694.515 & 0 & 279.392 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 27.479 & 4.201 & 0 & -3.341 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.001 & -0.001 \\ 75.150 & -0.947 \\ 0 & 0 \\ -0.695 & 0.160 \\ 0 & 0 \end{bmatrix}$$

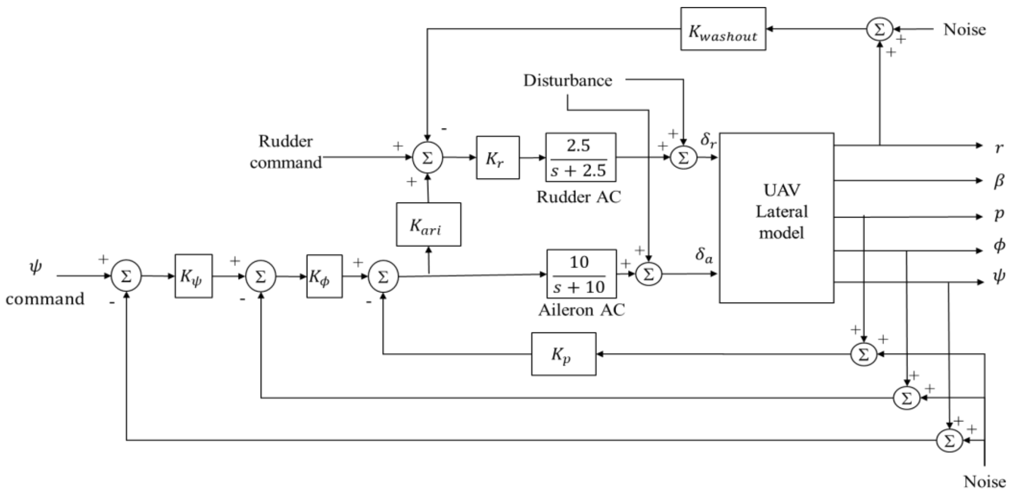


Figure 2 Flight control model

2.3 Heading Autopilot optimisation problem

The heading autopilot control structure present in this work is a conventional control structure as shown in Figure 2. To obtain a robust and optimum control system, the optimisation problem is formulated to find all possible controllers, as shown in Figure 2. The controllers are formulated as follows: $k_{washout} = \frac{x_1 s}{x_1 s + x_2}$, $k_p = \frac{x_8}{x_9}$, $k_{ari} = \frac{x_3}{x_4}$, $k_r = \frac{x_5 s + x_6}{x_7 s}$, $k_\phi = \frac{x_{10} s + x_{11}}{x_{12} s}$, $k_\psi = \frac{x_{13} U_0}{x_{14} g}$ where $\{x\} = [x_1, x_2, \dots, x_{14}]^T$ is the set or vector of design variables while all of them are within the domain $[1 \times 10^{-4}, 1]$. This means that all controller coefficients can be within the interval of $[1, 1 \times 10^4]$. The objective function is to minimise the integral square error, subject to several constraints, taking into account the robustness, precision, and rapidity of the control system. The proposed constraints include the response limits of all state variables, stability margin, sensitivity and complementary sensitivity function. The formulation of the proposed optimisation problem is expressed as follows:

$$min: f(\{x\}) = \int_{t=0}^{t} e^2(t) dt$$

Subjected to:

$g_1 = stability\ margin\ under\ 20\%\ uncertainty > 1$,

$g_2 = \%overshoot = \left\{100 \times max\left(\frac{y(t) - y(0)}{y(\propto) - y(0)}\right)\right\} \leq 5$,

$g_3 = rise\ time = t_{@0.9y(\propto)} - t_{@0.1y(\propto)} \leq 15$,

$g_4 = settling\ time = min\left\{t > 0; \frac{|y(t^*) - y(\propto)|}{y(\propto)} \leq 0.02 \text{ for all } t^* \geq t\right\} \leq 30$,

$g_5 = p \leq 60\ deg/s$,

$g_6 = r \leq 10\ deg/s$,

$g_7 = \beta \leq 10\ deg$,

$g_8 = \emptyset \leq 45\ deg$,

$g_9 = \delta_a \leq 45\ deg$,

$g_{10} = \delta_r \leq 30\ deg,$
$g_{11} = max|\boldsymbol{S}(j\omega)| \leq 1.4,$
$g_{12} = max|\boldsymbol{T}(j\omega)| \leq 1.4$

where $\int_{t=0}^{t} e^2(t)dt$ is the integral square error of heading step response while the robust stability margin (g_1) is calculated using MATLAB function *robstab*. The $\boldsymbol{S}$ and $\boldsymbol{T}$ refer to sensitivity and complementary sensitivity functions, respectively, which are due to input disturbance and output noise at the plant's input and output, as shown in Figure 2. Their maximum magnitudes in the frequency response are limited to 1.4 based on the reference [40].

3. Numerical experiment

To obtain a robust, precise, and rapid autopilot control system, the proposed optimisation problem is solved using several metaheuristic algorithms. MH is a non-gradient-based optimiser inspired by natural phenomena such as genetic evolution, food finding of animals or insects, mathematical or physical laws, etc. The general optimisation search process of MHs is illustrated in Figure 3. Most, if not all, MHs start their search with an initial population, a set of design solutions. The population is then improved in some ways depending on the philosophical ideas of the metaheuristics, which usually consist of the reproduction and selection stages. The population of solutions are repeatedly improved until it gets close to the optimum and reaches a termination criterion, which usually is the maximum number of function evaluations.

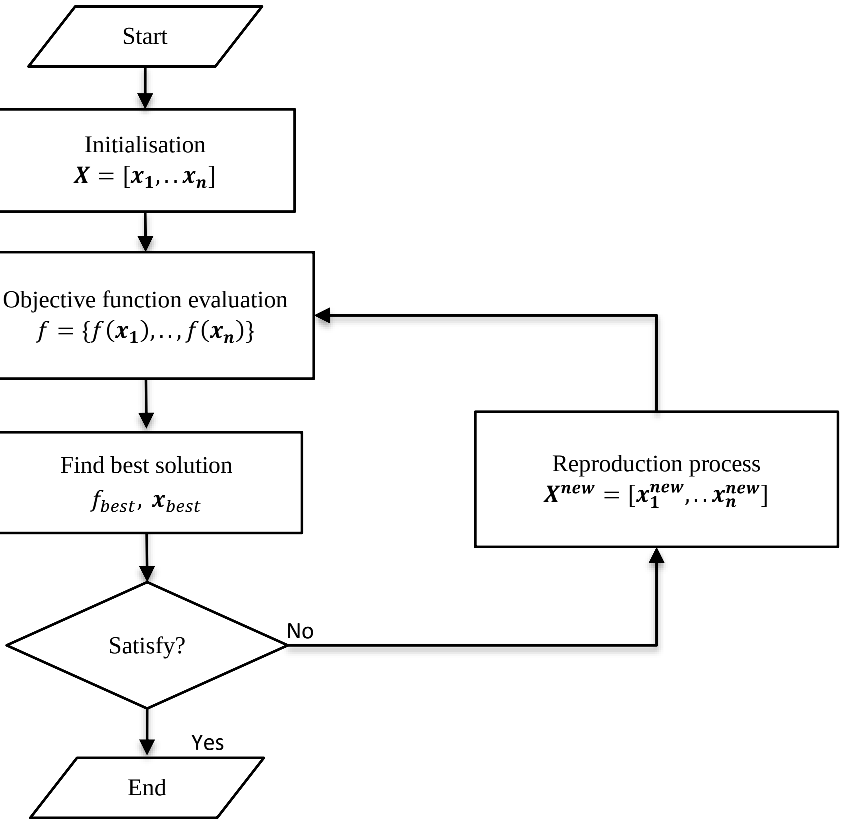


Figure 3 Shown general search process of MHs

To verify the performance and compare it with state-of-the-art metaheuristics, three groups of metaheuristics were selected. These groups consist of classical metaheuristics, newly developed metaheuristics, and the top-ranked metaheuristics from the Congress on Evolutionary Computation (CEC) competitions. The metaheuristic algorithms used in this study and their optimisation parameter settings are detailed as follows:

1. Differential evolution (DE) [30] ; DE/best/2/bin strategy was used. A scaling factor, crossover rate and probability of choosing elements of mutant vectors are 0.5, 0.7, and 0.8 respectively.
2. Particle swarm optimisation (PSO) [31] ; The starting inertia weight, ending inertia weight, cognitive learning factor, and social learning factor are assigned as 0. 5, 0.01, 2.8 and 1.3 respectively.
3. Real-code ant colony optimisation (ACOR) [32] ; The parameter settings are q = 0.2, and ξ = 1.
4. Artificial Rabbits Optimisation (ARO) [33] ; Original code from [33] is used. No parameter setting required
5. Dandelion Optimizer (DO) [34] ; Original code from [34] is used. No parameter setting required
6. Cheetah Optimizer (CO) [35] ; Original code from [35] is used. Number of search agents in a group is set to be 2.
7. Pelican Optimisation Algorithm (POA) [36] ; Original code from [36] is used. No parameter setting required
8. Self Adaptive Differential Evolution (JADE) [37]; the process optimisation parameters are iterative self-adaption
9. Success-History Based Adaptive Differential Evolution (SHADE) [38] ; the process optimisation parameters are iterative self-adaption
10. SHADE with Linear Population Size Reduction (L-SHADE) [39] ; the process optimisation parameters are iterative self-adaption

The proposed optimisation problem was solved using each MH optimiser for 20 optimisation runs. The population size and number of iterations were set to 30 and 100, respectively, and the total number of function evaluations was limited to 3,000 for each run. In addition, the control system obtained from the best run of the best MHs is verified with the popular MATLAB control tuning toolbox "Systune" command.

4. Results and discussions

After conducting ten optimisation runs of each MH to solve the proposed heading autopilot, the results were compared based on the objective function values presented in Table 2. Note that only algorithms that could find feasible solutions are reported in the table. The mean value of the objective function was used to evaluate the convergence rate and consistency of the algorithms, while the number of successful runs was used to measure their search consistency. In the event that two algorithms had the same number of successful runs, their consistency performance was evaluated using the standard deviation (STD) of the objective function values. Only the algorithm that found the feasible solution at least twice was allowed to have mean and STD values.

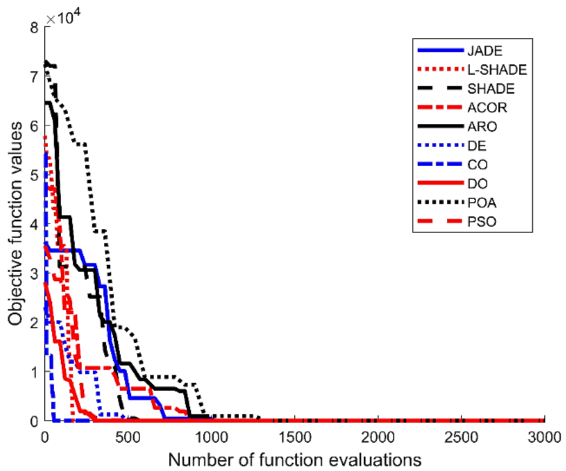
The results in Table 2 indicate that JADE outperformed all other MHs in terms of convergence rate, followed by L-SHADE and SHADE, which ranked second and third, respectively. In terms of search consistency, Only DE, ACOR, ARO, POA, JADE, SHADE and L-SHADE could produce 20 out of 20 feasible solutions. However, when considering the standard deviation (STD) values, JADE still had the highest variability, indicating that their performance was less consistent compared to other MHs. Among the 10 MHs, L-SHADE achieved the lowest objective function value, indicating its superior optimisation capability.

In Figure 4, the search history of the best run among all the Metaheuristics (MHs) is depicted. The graph highlights CO as the fastest in converging, followed by L-SHADE and DO in that order. However, examining the final state in Figure 4b reveals that L-SHADE achieved the lowest objective function value, trailed by ARO and DO. This ranking aligns with the minimum values listed in Table 2. Although Figure 4 may not position JADE as outstanding, it remains the most efficient choice based on the comprehensive results presented in Table 2. It's important to note that Figure 4 represents only the best outcome from 20 optimization runs. Given the inherent lack of search consistency in Metaheuristics, relying solely on a single best solution doesn't accurately evaluate the MHs performance. In this context, JADE emerges as the most reliable option for solving the proposed autopilot control optimization design. This conclusion is supported by JADE consistently obtaining the lowest average and standard deviation of objective function values.

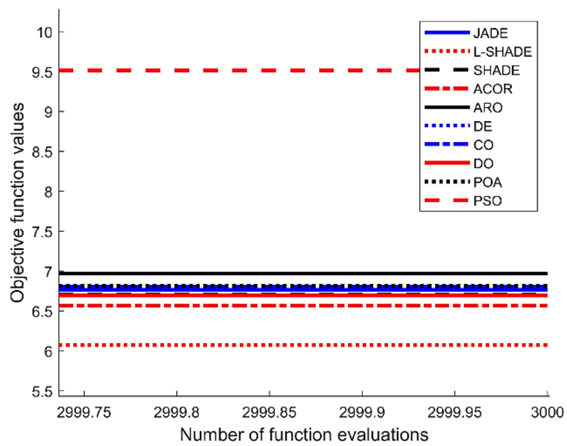
Overall, among the 10 MHs evaluated in this study for the proposed autopilot control optimisation design that takes into account robustness, precision, and rapid trajectory tracking control, JADE is found to be the most efficient MH, followed by L-SHADE and SHADE as the second and third best, respectively.

Table 2 Optimum results

MHs	Mean	STD	Min	Max	No. of successful runs
DE	10.783	3.863	6.812	15.884	20
ACOR	9.429	1.220	6.569	10.473	20
PSO	17.796	6.971	9.514	23.920	19
ARO	10.502	4.800	6.968	24.068	20
CO	11.259	5.002	6.805	23.020	17
DO	11.602	6.795	6.696	26.427	15
POA	12.482	3.563	6.813	18.833	20
JADE	7.617	0.700	6.765	9.318	20
SHADE	9.114	3.568	6.704	23.768	20
L-SHADE	7.865	0.843	6.072	8.840	20



(a) original



(b) zoom in on Figure 4(a)

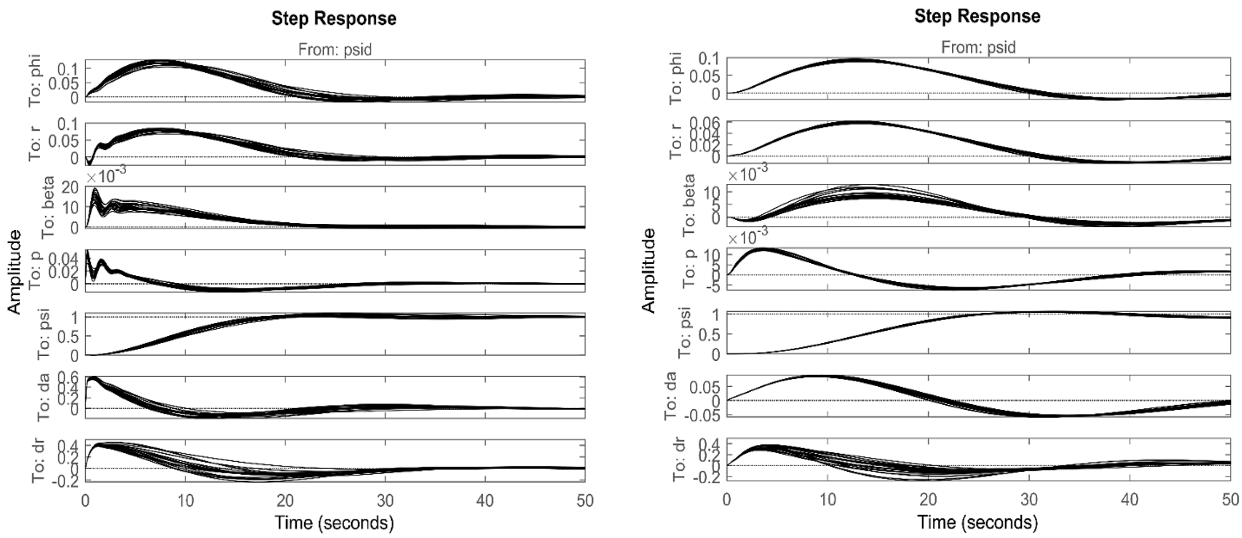
Figure 4 Search history of the best run of the top three best MHs, (a) original, and (b) zoom in

Furthermore, the control system optimised using metaheuristics (MHs) was validated using the Systune [22] command from the MATLAB control tuning toolbox, whereas the results are presented in Table 3. The optimum controller and its performance parameters are displayed, with Figures 4 and 5 illustrating the step response and step disturbance response of the control system obtained from the best run of JADE and Systune. The analysis of Table 3 revealed that the control system obtained from Systune violated several constraints, such as rise time, settling time, overshoot percentage, sensitivity, and complimentary sensitivity gain limit, whereas the control system obtained from JADE satisfied all requirements and performed better performance and robustness in disturbance.

Figure 5 illustrates the step response plot under 20% uncertainty, which indicates that both JADE and Systune obtained control systems that remain stable under the defined uncertainty. However, Systune gives a slower response to the input. Figure 6 depicts the response of the step disturbance injected at the aircraft input, and it was observed that the control systems obtained from both JADE and Systune were able to reject the disturbance in both inputs. Nonetheless, Systune obtained much higher peak responses for all state variables, implying that although the response of the control system from Systune seems stable, the flying quality is worse and may lead to stalling in real-life situations.

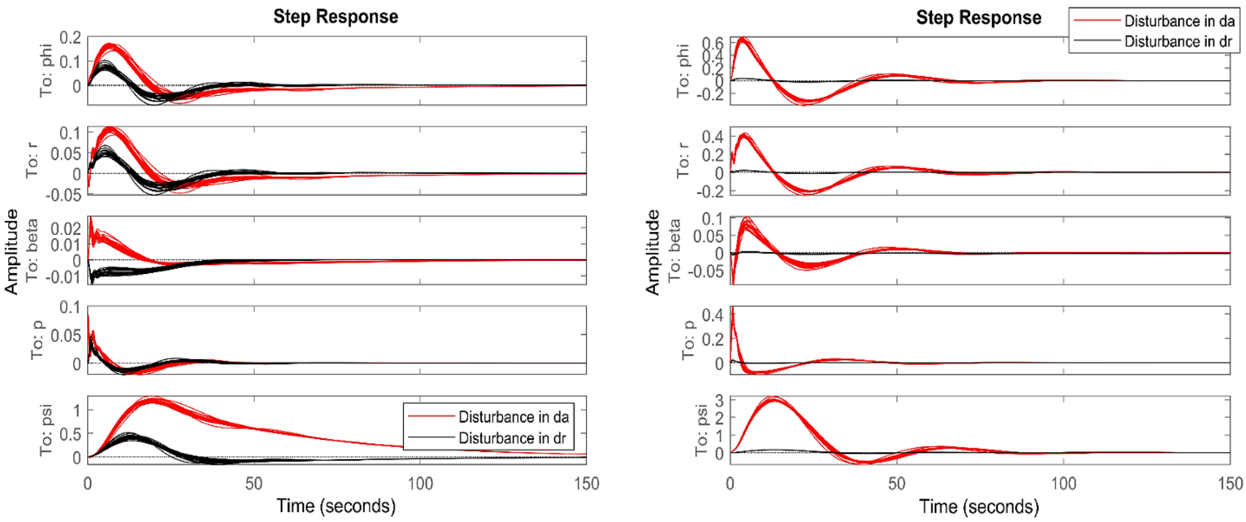
It is important to note in this study that JADE and Systune utilised different formulations of the optimisation problem in tuning the control system. Systune herein employed the default MATLAB tuning algorithm, which converts each tuning requirement into normalised values of objective and constraint functions and solves such a problem by a non-smooth optimisation technique. However, it is well known that this approach might be inefficient when dealing with several constraints and might get trapped in local optima.

Furthermore, this study shows that a recently introduced MH, whether it is metaphoric or not, cannot be overlooked as JADE can outperform L-SHADE, the CEC winner, which many believe to be the state-of-the-art metaheuristic. This is confirmed by another study [41], as the sine-cosine algorithm can surpass conventional optimisers and metaheuristics.



(a) JADE (b) Systune

Figure 5 Step response of the system with 20% uncertainty obtained from (a) the best run of JADE, (b) Systune



(a) JADE (b) Systune

Figure 6 Response of step disturbance at the input of the system with 20% uncertainty obtained from (a) the best run of JADE, (b) Systune

Table 3 Optimum controller and performance of the control system obtained from the best run of JADE and Systune, (** denote that the constraints are violated)

Parameter	JADE	Systune
k_{wash}	$\frac{0.7555s}{0.7555\,s + 0.009994}$	$\frac{14.47s}{14.47s + 1}$
k_p	1.779	0.868
k_{ari}	0.7936	0.492
k_r	$\frac{0.6083s + 0.1378}{0.7457s}$	$\frac{35.7s + 4.05 \times 10^{-6}}{s}$
k_{ϕ}	$\frac{0.9526s + 0.02174}{0.2408s}$	$\frac{0.0189s + 0.201}{s}$
k_{ψ}	0.1686	0.09649
$Rise\ time\ (\leq 15)$	12.1548	15.8939**
$Setting\ time\ (\leq 30)$	26.7879	72.2897**
$\%\ overshoot\ \ (\leq 5)$	2.8306	5.4021**
$Stability\ margin\ \ (\geq 1)$	1.3707	1.8477
$\max(p)\ \ (\leq 60\ deg/s)$	2.5803	0.7183
$\max(r)\ (\leq 10\ deg/s)$	4.5074	3.3322
$\max(\beta)\ (\leq 10\ deg)$	0.8227	0.5014
$\max(\phi)\ (\leq 45\ deg)$	6.8701	5.1394
$\max(\delta_a)\ (\leq 45\ deg)$	32.8051	4.9804
$\max(\delta_r)\ (\leq 30\ deg)$	17.9026	18.7187
$max\lvert S(j\omega)\rvert\ \ (\leq 1.4)$	1.3994	1.8294**
$max\lvert T(j\omega)\rvert\ \ (\leq 1.4)$	1.3920	2.0264**

5. Conclusions

In this study, metaheuristic algorithms were successfully applied to solve the proposed autopilot control optimisation design problem. The problem aimed to find controllers in several sections with the objective of minimising integral square error, subject to several constraints to ensure a robust, precise, and rapid reference tracking control system. Several MHs were employed to solve the problem, and their performances were investigated. Based on the results, JADE was found to be the most efficient algorithm. The study presents a simple but effective tool for designing a robust and optimum autopilot flight controller, and it also explores the performance of several MHs in the field of new optimisation design problems. Developing an efficient optimisation framework for autopilot control design can significantly improve the accuracy and reliability of autonomous systems in aviation. Although the proposed technique can achieve optimal performance, and robustness against uncertainties, it is inherently linear control design techniques, best suited for linear trim conditions. When the flight envelope significantly differs from the linear regime, the design system must account for a linear parameter-varying system. Future work will focus on performance enhancement of MHs, particularly JADE, for this type of control design. This concept will also be implemented on other complex control systems for aircraft.

6. Acknowledgements

The authors are grateful for financial support from the National Research Council of Thailand, Grant No. N42A650549.

7. References

[1] Bouguettaya A, Zarzour H, Kechida A, Taberkit AM. Deep learning techniques to classify agricultural crops through UAV imagery: a review. Neural Comput Applic. 2022;34(12):9511-36.

[2] Vong CN, Conway LS, Feng A, Zhou J, Kitchen NR, Sudduth KA. Corn emergence uniformity estimation and mapping using UAV imagery and deep learning. Comput Electron Agric. 2022;198:107008.

[3] Biswas K. Military Aviation Principles. In: Dekoulis E, editor. Military Engineering. London: IntechOpen; 2019. p. 1-25.

[4] Bravo-Mosquera PD, Botero-Bolivar L, Acevedo-Giraldo D, Cerón-Muñoz HD. Aerodynamic design analysis of a UAV for superficial research of volcanic environments. Aerosp Sci Technol. 2017;70:600-14.

[5] Ahmed F, Mohanta JC, Keshari A, Yadav PS. Recent advances in unmanned aerial vehicles: a review. Arab J Sci Eng. 2022;47(7):7963-84.

[6] Oyana SNO, Li J, Usman M. Three-layer multi-uavs path planning based on ROBL-MFO. Guid Navigat Control. 2022;02(03):2250017.

[7] Lu H, Zhen Y, Hao M. Nonlinear autopilot design for fixed-wing UAV using disturbance observer based backstepping. 2020 Chinese Automation Congress (CAC); 2020 Nov 6-8; Shanghai, China. USA: IEEE; 2020. p. 4423-8.

[8] Zhang W, Jia J, Zhou S, Guo K, Yu X, Zhang Y. A safety planning and control architecture applied to a quadrotor autopilot. IEEE Robot Autom Lett. 2023;8(2):680-7.

[9] Zhao YY, Yang Z, Kong WR, Piao HY, Huang JC, Lv XF, et al. Hybrid gradient vector fields for path-following guidance. Def Technol. 2023;28:165-82.

[10] Yang X, An X, Wu Y, Ma F, Li B. Adaptive super-twisting sliding mode back-stepping control for hypersonic flight vehicle with impact angle constraint and autopilot dynamics. 17th International Conference on Control & Automation (ICCA); 2022 Jun 27-30; Naples, Italy. USA: IEEE; 2022. p. 38-43.

[11] Yao Y, Deng Z, Zhang X, Lv C. Design of a quadrotor control software experimental validation platform based on real-time hardware-in-the-loop simulation. 10th International Conference on Educational and Information Technology (ICEIT); 2021 Jan 18-20; Chengdu, China. USA: IEEE; 2021. p. 239-43.

[12] Evans WR. Control system synthesis by root locus method. Trans AIEE. 1950;69(1):66-9.
[13] McFarlane D, Glover K. A loop-shaping design procedure using H/sub infinity/synthesis. IEEE Trans Autom Control. 1992;37(6):759-69.
[14] Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. Automatica. 2002;38(1):3-20.
[15] Athans M. The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. IEEE Trans Autom Control. 1971;16(6):529-52.
[16] Mokhtari A, Benallegue A, Belaidi A. Polynomial linear quadratic Gaussian and sliding mode observer for a quadrotor unmanned aerial vehicle. J Robot Mechatron. 2005;17(4):483-95.
[17] Nichols RA, Reichert RT, Rugh WJ. Gain scheduling for H-infinity controllers: a flight control example. IEEE Trans Control Syst Technol. 1993;1(2):69-79.
[18] Latif Z, Shahzad A, Bhatti AI, Whidborne JF, Samar R. Autonomous landing of an UAV Using H∞ based model predictive control. Drones. 2022;6(12):416.
[19] Alfi A, Khosravi A, Lari A. Swarm-based structure-specified controller design for bilateral transparent teleoperation systems via synthesis. IMA J Math Control Inf. 2014;31(1):111-36.
[20] Lundstrøm P, Skogestad S, Wang ZQ. Performance weight selection for H-infinity and μ-control methods. Trans Inst Meas Control. 1991;13(5):241-52.
[21] Alfi A, Bakhshi A, Yousefi M, Talebi HA. Design and implementation of robust-fixed structure controller for telerobotic systems. J Intell Robot Syst. 2016;83(2):253-69.
[22] Feyel P. Evolutionary fixed-structure controller tuning against multiple requirements. IFAC-PapersOnLine. 2016;49(5):345-52.
[23] Alaviyan Shahri ES, Alfi A, Tenreiro Machado JA. Fractional fixed-structure H∞ controller design using Augmented Lagrangian Particle Swarm Optimization with Fractional Order Velocity. Appl Soft Comput. 2019;77:688-95.
[24] Feyel P, Duc G, Sandou G. Evolutionary fixed-structure μ-synthesis. IEEE Symposium on Computational Intelligence in Control and Automation (CICA); 2014 Dec 9-12; Orlando, USA. USA: IEEE; 2014. p. 1-8.
[25] Gahinet P, Apkarian P. Decentralized and fixed-structure H∞ control in MATLAB. 50th IEEE Conference on Decision and Control and European Control Conference; 2011 Dec 12-15; Orlando, USA. USA: IEEE; 2011. p. 8205-10.
[26] El-Mahallawy AA, Yousef HA, El-Singaby MI, Madkour AA, Youssef AM. Robust flight control system design using H∞ loop-shaping and recessive trait crossover genetic algorithm. Expert Syst Appl. 2011;38(1):169-74.
[27] Qing-Tang F, Zhi-Hong J, Ji-Hong Z, Shi-Qian L. Fixed structure flight control law design based on genetic algorithm. The Proceedings of the Multiconference on Computational Engineering in Systems Applications; 2006 Oct 4-6; Beijing, China. USA: IEEE; 2006. p. 659-62.
[28] Sedehi AG, Alfi A. Swarm-based robust fixed-structure controller design for buck converter using Kharitonov approach: design and experiment. Int J Dynam Control. 2022;10(4):1251-64.
[29] Maruta I, Kim TH, Sugie T. Synthesis of fixed-structure H∞ controllers via constrained particle swarm optimization. IFAC Proc Vol. 2008;41(2):7843-8.
[30] Gao Z, Xiao T, Fan W. Hybrid differential evolution and Nelder–Mead algorithm with re-optimization. Soft Comput. 2011;15(3):581-94.
[31] Schutte JF, Groenwold AA. Sizing design of truss structures using particle swarms. Struct Multidisc Optim. 2003;25(4):261-9.
[32] Pholdee N, Bureerat S. Hybrid real-code ant colony optimisation for constrained mechanical design. Int J Syst Sci. 2016;47(2):474-91.
[33] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell. 2022;114:105082.
[34] Zhao S, Zhang T, Ma S, Chen M. Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. Eng Appl Artif Intell. 2022;114:105075.
[35] Akbari MA, Zare M, Azizipanah-abarghooee R, Mirjalili S, Deriche M. The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. Sci Rep. 2022;12(1):10953.
[36] Trojovský P, Dehghani M. Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. Sensors. 2022;22(3):855.
[37] Zhang J, Sanderson AC. JADE: Self-adaptive differential evolution with fast and reliable convergence performance. IEEE Congress on Evolutionary Computation; 2007 Sep 25-28; Singapore. USA: IEEE; 2007. p. 2251-8.
[38] Tanabe R, Fukunaga A. Success-history based parameter adaptation for Differential Evolution. IEEE Congress on Evolutionary Computation; 2013 Jun 20-23; Cancun, Mexico. USA: IEEE; 2013. p. 71-8.
[39] Tanabe R, Fukunaga A. Improving the search performance of SHADE using linear population size reduction. IEEE Congress on Evolutionary Computation (CEC); 2014 Jul 6-11; Beijing, China. USA: IEEE; 2014. p. 1658-65.
[40] Garpinger O, Hägglund T. Software-based optimal PID design with robustness and noise sensitivity constraints. J Process Control. 2015;33:90-101.
[41] Biertümpfel F, Pholdee N, Bureerat S, Pfifer H. Adaptive boundary sine cosine optimizer with population reduction for robustness analysis of finite time horizon systems. Appl Soft Comput. 2021;113:107900.