CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.3 / 2024 / pp.205-217
https://doi.org/10.51537/chaos.1396823

Bio-Inspired Jumping Spider Optimization for Controller Tuning/Parameter Estimation of an Uncertain Aerodynamic MIMO System

David Mohammed Ezekiel [*,1], **Ravi Samikannu** [*,2] **and Oduetse Matsebe** [β,3]

*Department of Electrical and Communications Systems Engineering, Botswana International University of Science & Technology (BIUST), Private bag 16, Palapye, Botswana, βDepartment of Mechanical Energy and Industrial Engineering, Botswana International University of Science & Technology (BIUST), Private bag 16, Palapye, Botswana.

ABSTRACT The practical near impossibility of empirical attempts in estimating optimal controller gains makes the use of metaheuristics strategies inevitable to automatically obtain these gains by an iterative heuristic simulation procedure. The convergence of the gain values to the local or global solutions occur with ease. In designing controllers for the Twin-Rotor MIMO System (TRMS) Jumping Spider Optimization Algorithm (JSOA), a novel neoteric population-based bio-inspired metaheuristic approach is used to obtain optimum values for the Proportional Integral and Derivative (PID) controllers. With the k_p, k_i, k_d controller gains as the decision variables, the JSOA solution to a nonlinear multi-objective optimization problem subject to some intrinsic constraints spawned optimal values for the controllers' variables. Counter to other algorithms (deterministic and stochastic) that get caught in local minima, JSOA evolved a solution after searchingly rummaging the entire solution search space in a vectorized fashion for an optimal value. Compared with several other versatile controllers (using GA, PSO, Pattern Search, and Simulated Annealing), statistical results obtained showed JSOA technique provided a unique solution and found the gains of the PID controllers marginally in relation to the others (optimization methods).

KEYWORDS
Jumping Spider Optimization Algorithm (JSOA), Meta-heuristics, Optimization, PID, Intelligent control, Dynamic system, Nonlinear system, Linearization, Pitch and Yaw, Twin-Rotor MIMO System (TRMS)

INTRODUCTION

Recent advances in computerized/computing technology have revolutionized the aerospace industry, putting flying vessels and equipment at the cutting-edge. Flying vessels or maneuvering vehicles designed for Vertical Take-Off and Landing (VTOL) like helicopters, drones, Unmanned Aerial Vehicles (UAVs) are advantageous over fixed-wing types (e.g., airplanes) in that they can maneuver and hover around in confined and limited spaces (Toha and Tokhi 2010). To carry out research on the helicopters and drones, control laboratories around the world are equipped with a laboratory-scaled version of the helicopter model, the TRMS. Essentially, it is an electromechanical, electrodynamic, and aerodynamic equipment prototype which models the dynamics of a typical true-life helicopter (Ezekiel *et al.* 2020a; Mones and Diaa-Eldeen 2017; Toha and Tokhi 2011). It is a unique equipment with fewer Degrees of Freedom (DoF) than a true-life helicopter, and its Angle Of Attack (AOA) in piercing through space and air is fixed (see Table 1 for comparison). Since it is an equipment fixed to the workbench in the laboratory, it most important function is to develop control strategies to control and maintain the testbed in a hovering posture (Ezekiel *et al.* 2020b; Choudhary 2017), signifying a helicopter position when airlifting humans and equipment during emergency rescues, etc.

The challenge to control multivariable systems (SIMO, MISO, MIMO systems) has attracted numerous researches over the past few centuries. This is owing to the problem of cross-linkages or dynamic couplings which are significant, and pairing issues between the input-output variables/channels. The best practice is not to trivialize these coupling and dynamic effects in modeling

Manuscript received: 27 November 2023,
Revised: 31 January 2024,
Accepted: 7 March 2024.

[1]ed18100190@studentmail.biust.ac.bw (**Corresponding author**)
[2]ravis@biust.ac.bw
[3]matsebeo@biust.ac.bw

Table 1 Similar and contrasting features between a Helicopter and its Prototype/Model (Basturk 2006)

Attributes	Helicopter	TRMS
Locus of pivot point	Main rotor head	Three-fifth way between twin-rotor
Vertical control	Main rotor's pitch angle	Main rotor Speed control
Horizontal control	Tail rotor's yaw angle	Tail rotor Speed control
Degrees-of-Freedom	3-DOF yaw, pitch, and roll axes	1 or 2-DOF pitch and/or yaw
Angle of Attack (AOA)	AOA changes (variable)	Fixed AOA

and design because in so doing results in degraded controller's performance and in unrealized control objectives (Ezekiel *et al.* 2020b; Choudhary 2017; Raghavan and Thomas 2017). The all-purpose PID controllers uniquely designed for linear systems are applicable for only SISO systems, which may not be adequate for multivariable MIMO systems due to the aforementioned problem involved, even if the channels are decoupled. Hence the need for an efficient strategy to handle these effects.

Poised with this challenge, therefore, authors and researchers have proposed a vast number of methodologies using Classical, Optimal, Robust, Adaptive, and Intelligent control strategies. The investigation of deadbeat control as a strategy capable of controlling the TRMS as a decoupled subsystem (of vertical and horizontal) was carried out in (Wen and Lu 2008). Simulations were performed on the decoupled system, which is a linearized system. It can be inferred that errors due to approximations were systematically introduced on the system due to decoupling as reflected in the final system's simulations results. Ghellab et al. designed a multistage fuzzy gain-scheduled feedback linearization-based controller using the nonlinear TRMS model to stabilize the beam of the TRMS to its horizontal posture (Ghellab *et al.* 2018). The effects of cross-couplings were incorporated in the design of the control laws. The proposed controller was implemented real-time. However, the use of output feedback linearization inherently introduced disturbance as presented in the simulations results obtained therein.

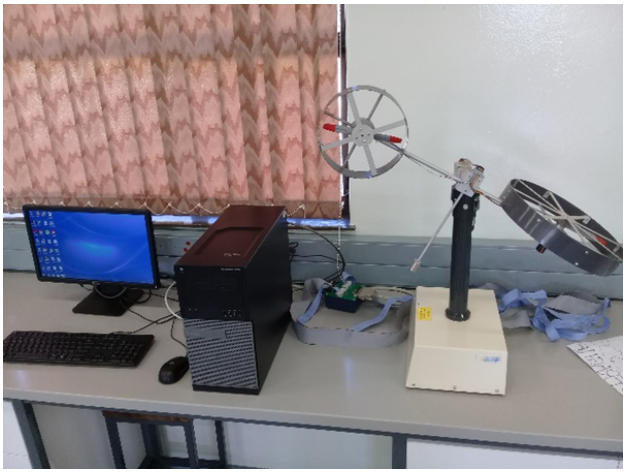
The robust and adaptive Sliding Mode Control (SMC) is a controller designed on 2 sliding control surfaces on the control plane (Mondal 2012a; Butt and Aschemann 2015; Mondal and Mahanta 2012; Mondal 2012b; Dimassi *et al.* 2019). The main benefits of SMC as a controller are robust system robustness against nonlinearities, modeling errors, and parametric uncertainty (variations) as well as being very effective in rejecting disturbances. In (Mondal and Mahanta 2012), a terminal sliding mode controller was implemented on the linearized nonlinear TRMS plant. In (Mondal 2012a; Mondal and Mahanta 2012; Mondal 2012b), the 2nd order SMC was proposed to control the TRMS plant while a MIMO Integral SMC of a 2-DOF helicopter was proposed in (Butt and Aschemann 2015). A Fuzzy-Sliding and Fuzzy-Integral-Sliding Controller (FS/FISC) for the TRMS were presented in (Mondal 2012a). The main challenge in the use of SMC is the chattering phenomenon that exists in the control signals owing to the discontinuous control action that may excite un-modeled high-frequency dynamics powerful enough to damage the actuators. Although the undesired effects of chattering were reduced to a barest in simulations in (Mondal 2012a; Mondal and Mahanta 2012), yet the effect still persisted and the control signal was still discontinuous and not smooth enough for practical purposes.

Parallel distributed fuzzy LQR controllers were utilized in (Tao *et al.* 2010) to accommodate different regions of operation for the decoupled system to separately control the pitch and yaw angles in pure simulations without any real-time applications on the system. Particle Swarm Optimisation (PSO) invented by Kennedy & Eberhard in the 90's, one of the most recent intelligent optimization strategies, was employed to augment the tuning process of the popular PID controller in (Toha and Tokhi 2011; Al-Mahturi and Wahid 2017). It is a population-based stochastic search global optimisation technique inspired by nature. It is based on simulating the phenomenon of a swarm of schools of fishes or flocks of birds competing for food (Toha and Tokhi 2011). PSO is designed to operate on a swarm of particles where each particle represents a candidate solution to the optimization problem (Al-Mahturi and Wahid 2017). These particles are arranged in an n-dimensional search space and a randomized selection of positions and velocities for their best values. The flying experience of each particle swarm automatically adjusts the velocity of the individual alongside the other particles in the swarm. The drawback to this technique is that it cannot directly handle multi-objective optimization problems and may not converge for large parametric modeling (large variables).

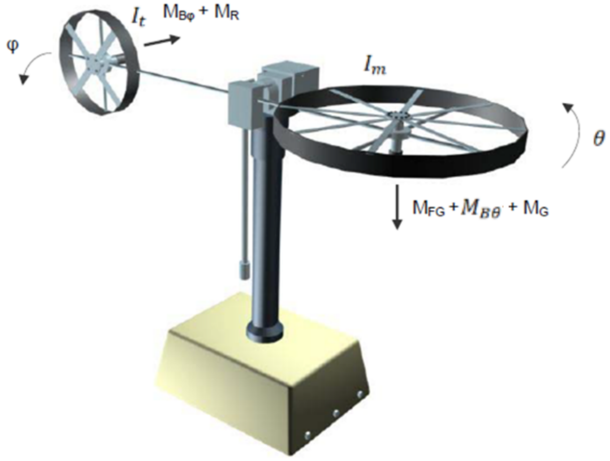
Juang et al. proposed an evolutionary algorithm methodology using Real-coded Genetic Algorithm (R-GA) for the parametrized modeling and optimization of PID controller gains to control the TRMS (Juang *et al.* 2008). The simulations results were not satisfactorily optimal due to the bang-bang inputs used and the highly oscillatory results obtained. Intelligent systems are employed to intelligently tune or determine the best possible values of the system parameters (decision variables) for optimum system performance. Some of the techniques used have been mentioned above already under "Intelligence control schemes". Mostly these schemes are nature-inspired or biologically inspired (bio-inspired) from living organisms or natural processes. In this paper, a Multi-Objective GA (MOGA) algorithm is employed for tuning custom-designed PID controller parameters for optimized performance. The preponderating advantage of the simple design reposes/stands in the compromise on the various control performance requirements or preconditions. The paper organization is as follows: Section 1 introduces the TRMS and control strategies developed by researchers; Section 2 reviews related literature; Section 3 focuses on the mathematical modelling; while section 4 discusses the intelligent bio-inspired JSOA-PID controller design and tuning process utilized; Section 5 presents and discusses simulation and experimental results obtained and the further discussions of the results; with a conclusion drawn at the end in Section 6.

MATHEMATICAL/ANALYTICAL MODELLING OF THE TRMS

The mathematical models used are based on the rotations of the asymmetrically weight-distributed beam of the TRMS owing to the unequal weights of the main and tail rotor assemblages placed end to end. The TRMS as a dynamic system is modelled (i.e. described by equations) using the phenomenological model (Instruments 2000; Agrawal 2013).



(a) A setup of the TRMS in the Measurements Control and Instrumentation Laboratory of the Botswana International University of Science and Technology (BIUST)



(b) Phenomenological model of the TRMS (Instruments 2006)

Figure 1 (a) A setup of the TRMS and (b) Phenomenological model of the TRMS

Equations of Motion governing the TRMS Based on Newton-Euler Approach

Let u_1, u_2 represent the two DC Stepper motors input voltages (for the pitch and yaw channels) respectively and τ_1, τ_2 represent their corresponding output torques. The torque-developed equations due to the resulting rotations can be modelled as:

$$\begin{aligned} \tau_1 &= \frac{K_1}{T_{11}s + T_{10}} u_1 \\ \tau_2 &= \frac{K_2}{T_{21}s + T_{20}} u_2 \end{aligned} \tag{1}$$

where $K_1, K_2, T_{10}, T_{11}, T_{21}, T_{20}$ are some lumped-parameter motor constants (motor torque constants, back emf constants, etc.) that have been determined experimentally (Instruments 2006).

Summing moments and forces acting on the system about the pitch axis for vertical motion yields the general form of the torque-developed equation of motion about the pitch plane (Raghavan and Thomas 2017; Instruments 2006; Chaudhary and Kumar 2019a,b; Sodhi and Kar 2014) as:

$$I_m\ddot{\theta} = M_m - M_{FF} - M_{B\theta} - M_G \tag{2}$$

$$M_m = \alpha_1\tau_1^2 + b_1\tau_1 \quad \text{(nonlinear static characteristic)} \tag{3}$$

$$M_{B\theta} = B_{1\theta}\dot{\theta} - B_{2\theta}\sin(2\theta)\dot{\varphi}^2 \quad \text{(friction forces momentum)} \tag{4}$$

$$M_{FF} = M_g\sin(\theta) \quad \text{(gravity momentum)} \tag{5}$$

$$M_G = K_{gy}M_m\dot{\varphi}\cos(\theta) \quad \text{(gyroscopic momentum)} \tag{6}$$

Equations (3-6) are the momentum equations acting on the vertical plane.

Similarly, summing moments and forces acting on the system about the yaw axis yields the general form of the developed torque equation for motion about the yaw plane (Raghavan and Thomas 2017; Instruments 2006; Chaudhary and Kumar 2019a,b) as:

$$I_t\ddot{\phi} = M_t - M_{B\phi} - M_{CR} \tag{7}$$

where

$$M_t = \alpha_2\tau_2^2 + b_2\tau_2 \quad \text{- nonlinear static characteristics} \tag{8}$$

$$M_{B\phi} = B_1\phi\phi + B_2\theta\text{sign}\dot{\phi} \quad \text{- fric. forces momentum} \tag{9}$$

$$M_{CR} = K_c\left(\frac{T_os + 1}{T_ps + 1}\right)M_m \quad \text{- approx. cross-reactn momentum} \tag{10}$$

Where equations (8-10) are the momentums acting on the horizontal plane due to the rotational dynamics of the twin rotors.

Substituting equations (3-6) into (2) and (8-10) into (7) result in 2 differential algebraic equations with $\ddot{\theta}$ and $\ddot{\phi}$ numerically given by:

$$\begin{aligned} \ddot{\theta} &= \frac{1}{I_m}\Big(\alpha_1\tau_1^2 + b_1\tau_1 - Mg\sin\theta \\ &\quad - \left(B_{1\theta}\dot{\theta} - B_{2\theta}\sin(2\theta)\dot{\phi}^2\right) \\ &\quad - K_{gy}M_m\dot{\phi}\cos\theta\Big) \\ &= \frac{1}{I_m}\Big(\alpha_1\tau_1^2 + b_1\tau_1 - Mg\sin\theta \\ &\quad - B_{1\theta}\dot{\theta} + B_{2\theta}\sin(2\theta)\dot{\phi}^2 \\ &\quad - K_{gy}(\alpha_1\tau_1^2 + b_1\tau_1)\dot{\phi}\cos\theta\Big) \end{aligned} \tag{11}$$

$$\begin{aligned} \ddot{\phi} &= \frac{1}{I_t}\Big(\alpha_2\tau_2^2 + b_2\tau_2 - B_{1\phi}\dot{\phi} \\ &\quad - \frac{K_c(T_Ds + 1)}{T_Ps + 1}M_m\Big) \\ &= \frac{1}{I_t}\Big(\alpha_2\tau_2^2 + b_2\tau_2 - B_{1\phi}\dot{\phi} \\ &\quad - \frac{K_c(T_Ds + 1)}{T_Ps + 1}(\alpha_1\tau_1^2 + b_1\tau_1)\Big) \end{aligned} \tag{12}$$

These wholly define the equations of motion for the TRMS. The MATLAB/Simulink implementation of these equations for modeling, simulations, and control purposes is shown below:

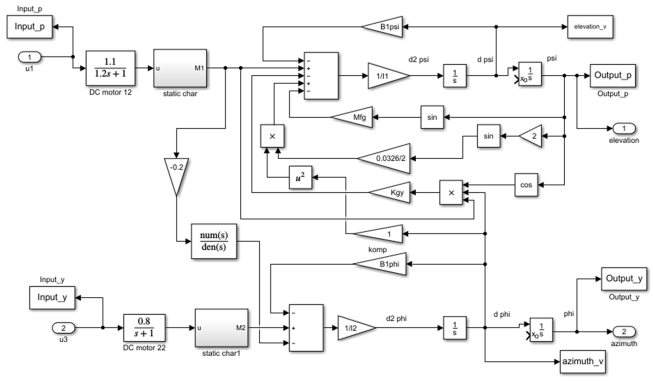


Figure 2 TRMS full implementation in Simulink

State-Space Models

The dynamic state of the linear plant given by the state-space representation (Chaudhary and Kumar 2019a) using dynamic equation is:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \tag{13}$$

The state vector x, input vector u, and output vector y for the system are given by equations (14-16)

$$x = \begin{bmatrix} \theta & \phi & \dot{\theta} & \dot{\phi} & \tau_1 & \tau_2 & M_{CR} \end{bmatrix}^T \tag{14}$$

$$u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T \tag{15}$$

$$y = \begin{bmatrix} \theta & \phi \end{bmatrix}^T \tag{16}$$

Modelling of the 2-DOF TRMS

Using the state, input and output vectors (equations (14-16)), the complete 2-DOF TRMS dynamic equations for the system can be modelled and in state-variable form by:

$$\begin{cases} \frac{d}{dt}\dot{\theta} = \dot{\theta} \\ \frac{d}{dt}\dot{\phi} = \dot{\phi} \\ \frac{d}{dt}\dot{\theta} = -\frac{Mg}{I_m}\sin\theta - \frac{B_{1\theta}}{I_m}\dot{\theta} + \frac{B_{2\theta}}{I_m}\sin(2\theta)\dot{\phi}^2 \\ \qquad + \left(1 - K_{gy}\dot{\phi}\cos\theta\right)\frac{\alpha_1\tau_1^2}{I_m} + \left(1 - K_{gy}\dot{\phi}\cos\theta\right)\frac{b_1\tau_1}{I_m} \\ \frac{d}{dt}\dot{\phi} = -\frac{B_{1\phi}}{I_t}\dot{\phi} + \left(1 - \frac{K_c(T_Ds+1)}{T_ps+1}\right)\frac{\alpha_2\tau_1^2}{I_t} \\ \qquad + \left(1 - \frac{K_c(T_Ds+1)}{T_ps+1}\right)\frac{b_2\tau_1}{I_t} \\ \frac{d}{dt}\tau_1 = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{K_1}{T_{11}}u_1 \\ \frac{d}{dt}\tau_2 = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{K_2}{T_{21}}u_2 \\ \frac{d}{dt}M_{CR} = \frac{d}{dt}\left(\frac{K_c(T_os+1)}{T_ps+1}\alpha_1\tau_1^2\right) + \frac{d}{dt}\left(\frac{K_c(T_os+1)}{T_ps+1}b_1\tau_1\right) \\ \qquad = 2\alpha_1\frac{K_c(T_os+1)}{T_ps+1}\tau_1\dot{\tau}_1 + b_1\frac{K_c(T_os+1)}{T_ps+1}\dot{\tau}_1 \end{cases} \tag{17}$$

or,

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ -\frac{Mg}{I_m}\sin x_1 - \frac{B_{1x_1}}{I_m}x_2 + \frac{B_{2x_1}}{I_m}\sin(2x_1)x_4^2 \\ \quad + \frac{b_1}{I_m}x_6 - K_{gy}\frac{b_1}{I_m}\cos x_1 x_4 x_6 + \frac{\alpha_1}{I_m}x_6^2 \\ \quad - K_{gy}\frac{\alpha_1}{I_m}\cos x_1 x_4 x_6^2 \\ -\frac{B_{1x_3}}{I_t}x_4 + \left(1 - \frac{K_c(T_Ds+1)}{T_ps+1}\right)\frac{b_2}{I_t}x_6 \\ \quad + \left(1 - \frac{K_c(T_Ds+1)}{T_ps+1}\right)\frac{\alpha_2}{I_t}x_6^2 \\ -\frac{T_{10}}{T_{11}}x_6 \\ -\frac{T_{20}}{T_{21}}x_7 \\ 2\alpha_1\frac{K_c(T_os+1)}{T_ps+1}x_6\dot{x}_6 + \frac{K_c(T_os+1)}{T_ps+1}b_1\dot{x}_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{K_1}{T_{11}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_2}{T_{21}} \end{bmatrix}^T u \tag{18}$$

$$y = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$$

INTELLIGENT CONTROL DESIGN

Bio-Inspired/Nature-Inspired Algorithm Design Based on a novel metaheuristic Jumping Spider Optimization Algorithm (JSOA)

In this research, the novel metaheuristic technique called Jumping Spider Optimization Algorithm (JSOA) is used as the optimization strategy for controllers' tuning. It is a Biologically-Inspired and Nature-Inspired optimization strategy inspired by the hunting habits of Arachnida Salticidae spider species (Peraza-Vázquez *et al.* 2022). JSOA mimics the behaviour of spiders in nature, modelling how it hunts for food/prey using a search, persecution, and jumping prowess and artistry (skills) to catch and kill (or prey) for its meal. Just like other evolutionary metaheuristics, this strategic hunting scheme or ruse of search, persecution, and jumping are harnessed to strike a balance over the entire solution space (search space) between exploitation and exploration. This is intended in solving a global optimization problem.

In this study, JSOA is used as an optimization algorithm to parametrically tune the PID controllers' gains in a decentralized control system architecture of the underactuated TRMS plant. In addition to the algorithm globally converging for the multi-objective problem, its performance is tested by comparing against several notable, popular, and well-established metaheuristics algorithms of Genetic Algorithm (GA), Pattern Search, Simulated Annealing, and Particle Swarm Optimization (PSO). The results (in tables and graphs) revealed that the proposed algorithm outperforms the aforementioned algorithms and is capable of solving real-world problems that can be considered very challenging with unknown search or solution spaces.

In the TRMS controller design, an optimization problem is setup where the gains of the PID controllers are set as the parameters to be optimized by the JSOA metaheuristic optimizer.

Modelling of the JSOA Optimization Algorithm

The proposed JSOA considers four hunting strategies of the spiders: attacking by persecution, search, jumping on prey and pheromone rate.

Attacking by Persecution The stealthy jumping to catch movements made by the spider in attacking its prey is modelled using Newton's third equation for linear or rectilinear motion. The spider moves along its coordinate axis with increasing or decreasing velocity at constant/uniform velocity linearly with time and given by:

$$x_i = \frac{1}{2}at^2 + v_o t \tag{19}$$

where x_i is the position of the ith follower spider, v_o, a, and t follow the usual conventional definition for the initial speed, acceleration, and time respectively. The optimization procedure is an iterative procedure where each iteration $x_i(t)$ is defined in terms of position with disparity in iterations equalling 1. The initial velocity $v_o = 0$.

$$\vec{x}_i(t+1) = \vec{x}_i(t) - \vec{x}_r(t) \tag{20}$$

where $\vec{x}_i(t+1)$ is the updated displacement of the jumping spider Search Agent (SA) of $(t+1)$ generation, $\vec{x}_i(t)$ is the prevailing or current ith SA in the tth generation, and $\vec{x}_r(t)$ is the rth SA for a randomly selected $r \neq i$ where r is a randomized integer in the interval between 1 and the maximum size of SAs. This is depicted as shown in Figure 3.

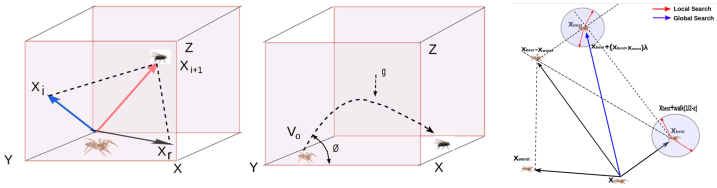


Figure 3 (a) Jumping Spider performing persecution (b) Jumping on prey under a projectile-like motion (c) The local and global search vectorizations (Peraza-Vázquez *et al.* 2022)

Jumping on Prey The jumping spider attacks and pounces on its prey in a projectile motion fashion. It is decomposed into its horizontal and vertical components and given by:

$$\begin{cases} \vec{x}_i = v_o(a)t\vec{i} \\ \dfrac{dx}{dt} = \vec{V}_x = v_o(a)\vec{i} \end{cases} \tag{21}$$

$$\begin{cases} \vec{y}_l = \left(v_o \sin(a)t - \dfrac{1}{2}gt^2\right)\vec{i} \\ \dfrac{dy}{dt} = \vec{V}_y = (v_o \sin(a) - gt)\,\vec{i} \end{cases} \tag{22}$$

Where equations (21) and (22) represent the horizontal and vertical components respectively. Therefore the equation for the trajectory is given as:

$$y = x\tan(a) - \frac{gx^2}{2V_o^2\cos^2(a)} \tag{23}$$

The trajectory w.r.t the iterations in the succeeding generations is therefore given by:

$$\vec{x}_l(t+1) = \vec{x}_l(t)\tan(a) - \frac{g\vec{x}_l^2(t)}{2V_o^2\cos^2(a)} \tag{24}$$

where $\vec{x}_l(t+1)$= the new displacement/position of the new search agent, $\vec{x}_l(t)$= the current ith search agent, $\vec{V}_x$=100 mm/sec, g=9.80665 m/s^2 is the acceleration due to gravity and a is the angle determined by a randomly generated ϕ angle lying between the interval (0, 1).

$$a = \phi\pi/180 \tag{25}$$

Searching for Prey JSOA in searching for prey executes a random search within the search space in order to locate its prey. Both the local and global search models have been developed and used in the JSOA approach. The local search is modelled as:

$$\vec{x}_l(t+1) = \vec{x}_{\text{best}}(t) + \text{walk}\left(\frac{1}{2} - \epsilon\right) \tag{26}$$

where $\vec{x_{best}}(t)$ is the best SA/solution from the preceding generation (or iteration), walk is a pseudo-random number (PRBS) generated with a Gaussian distribution in the interval $[-2, 2]$, and ϵ is a PRBS generated in the interval $[0, 1]$. The formulation for the global search model is given by:

$$\vec{x}_i(t+1) = \vec{x_{best}}(t) + \beta\left(\vec{x_{best}}(t) - \vec{x_{worst}}(t)\right) \tag{27}$$

where $\vec{x_{worst}}(t)$ is the worst SA (or solution) found in the preceding generation/ iteration, β is a Cauchy random number with mean $\mu = 0$ and variance $\sigma = 1$.

Pheromone's Rate Pheromone is a chemical substance/secretion secreted externally by the body of an individual jumping spider which is being perceived by the olfactory lobes of another individual that influences the physiology or behavior of the other individual animal of the same species. Pheromone is not unique to the jumping spider alone but a biological process shared with many animals particularly insects or arachnids. The modeling of the rate of pheromone production in jumping spider is given as:

$$\text{pheromone}(i) = \frac{\text{Fitness}_{best} - \text{Fitness}(i)}{\text{Fitness}_{best} - \text{Fitness}_{worst}} \tag{28}$$

where Fitness_{best}, Fitness_{worst} are the best and worst fitness values in the current generation/iteration respectively. The fitness values are normalized in the interval $(0, 1)$, the bounds corresponding to the worst and best pheromone rates respectively. For low or subjacent pheromone rates (pheromone$(i) \leq 0.3$) the position/displacement is recalculated by:

$$\vec{x}_l(t) = \vec{x}_{\text{best}}(t) + \gamma\left(\vec{x}_{r_1}(t) - (-1)^{\sigma}\vec{x}_{r_2}(t)\right) \tag{29}$$

where $\vec{x}_l(t)$ is the SA (but this time with low pheromone rate for updating, r_1, r_2 are random integers numbers randomized within the interval of $[1, \text{maximum size of SAs}]$, $r_1 \neq r_2$, $\vec{x}_{r_1}(t), \vec{x}_{r_2}(t)$ represent the r_1th r_2th search agents randomly selected, σ is binary random number generated i.e. $\sigma \in [0, 1]$.

SIMULATIONS RESULTS AND DISCUSSIONS

Design, modeling, simulations, and algorithms optimizations were carried out. The following results were obtained as explained in this section.

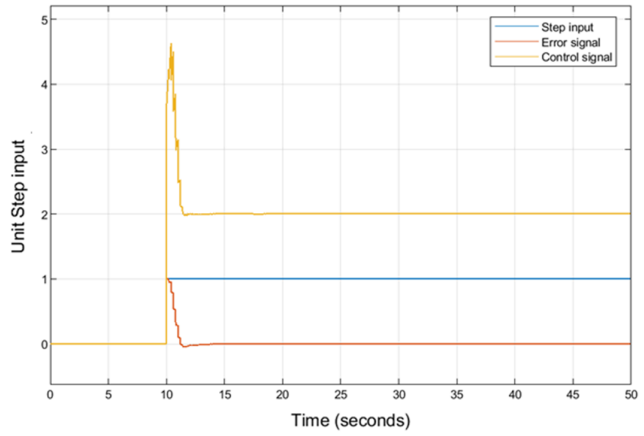


Figure 4 Control & error signals compared against step input signal

The PID controller configuration shows the control action of the controller in response to non-zero input and the error (Figure 4). The basic task of intelligent control is the controller-tuning design problem where satisfactory values for the adjustable parameters like the proportional gain, normalized gain, and integral time must be found to satisfy or achieve the desired closed loop requirements for a given system or process.

Results from Linearization of the TRMS Plant

Using the formulations presented in Klee and Allen (2018) for linearizing a nonlinear system about an operating point based on the Jacobian matrix of the vector-valued function $f(t, \underline{x}, \underline{u})$ which defines the state derivatives in MATLAB/SIMULINK software, the following useful results were directly obtained:

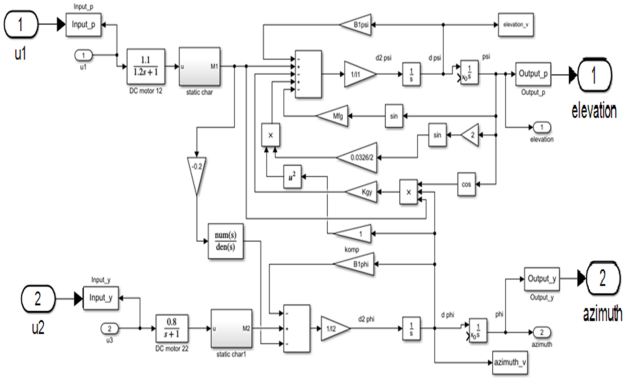


Figure 5 TRMS Simulink Nonlinear model prepared for linearization

Note: u_1, u_2 are the step inputs to the TRMS system while elevation and azimuth are the pitch & yaw outputs. The A, B, C, D system matrices were obtained to be:

$$
A = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-4.706 & 0 & -0.08824 & 0 & 1.246 & 0 & 0 \\
0 & 0 & 0 & -5 & 1.482 & 3.6 & 18.75 \\
0 & 0 & 0 & 0 & -0.8333 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -0.01694 & 0 & -0.5
\end{bmatrix},
$$

$$
B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix},
$$

$$
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},
$$

$$
D = \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\tag{30}
$$

Representing the system matrices in transfer function transform the continuous-time state-space system matrices to the following equations in the frequency-domain:

$$
\begin{cases}
G_{11} = \dfrac{1.246}{s^3 + 0.9215s^2 + 4.78s + 3.922} \\
G_{21} = \dfrac{1.482s + 0.4234}{s^4 + 6.333s^3 + 7.083s^2 + 2.083s} \\
G_{12} = 0 \\
G_{22} = \dfrac{3.6}{s^3 + 6s^2 + 5s}
\end{cases}
\tag{31}
$$

As can be seen from the 4 transfer functions obtained (equation 31) for our particular TRMS plant, there are main rotor - yaw plane or axis i.e., G_{21} interactions or cross couplings but there seems to be no interaction at all between the tail rotor and the pitch axis or pitch plane, i.e., G_{12}. There is therefore the need to design a decoupler to eliminate or reduce to the barest the G_{21} interactions offered by the tail rotor (powered by its DC motor) which serves as disturbance to the movements/ rotations about the yaw axis.

Let G_m and G_t represent the pitch & yaw planes' transfer functions respectively of the decoupled system. Making substitutions for $G_{11}, G_{12}, G_{21}, G_{22}$ to obtain G_m and G_t respectively, in MATLAB, as:

$$\begin{cases} G_m = \dfrac{1.246}{s^3 + 0.9215s^2 + 4.78s + 3.922} \\ G_t = \dfrac{3.66}{s^3 + 6s^2 + 45s} \end{cases} \tag{32}$$

Optimizing the pitch model of the TRMS is tasking and very involved because of the system complexity and the cross-correlational existing between the main and tail rotors' effects.

Results from Decoupling the TRMS

Accordingly, from equation 34 it follows that

$$\begin{cases} G_{\text{inv}11}(s) = 0.8026s^3 + 0.74s^2 + 3.2s + 3.144 \\ G_{\text{inv}12}(s) = 0 \\ G_{\text{inv}21}(s) = \dfrac{-0.3304s^7 - 2.381s^6 - 5.711s^5 - 13.74s^4 - 19.18s^3}{s^4 + 6.333s^3 + 7.083s^2 + 2.083s + 2.637 \cdot 10^{-13}} \\ \qquad\qquad + \dfrac{-10.96s^2 - 1.851s + 1.982 \cdot 10^{-12}}{s^4 + 6.333s^3 + 7.083s^2 + 2.083s + 2.637 \cdot 10^{-13}} \\ G_{\text{inv}22}(s) = 0.2778s^3 + 1.667s^2 + 1.389s \end{cases} \tag{33}$$

Using the simplified decoupling technique/methodology based on the generalized procedure of matrix inversion formula (Yang *et al.* 2016), the decoupling plant transfer matrix $G_R(s)$ as presented in (Yang *et al.* 2016) is expressed as:

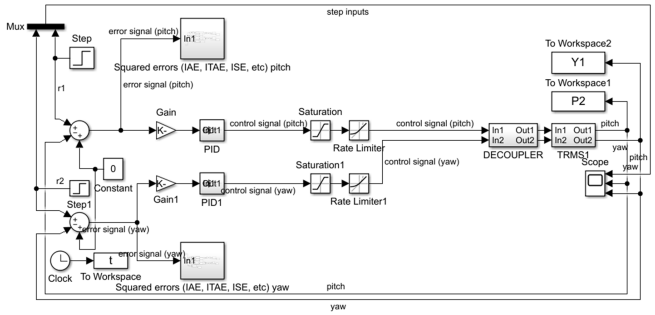
$$G_R(s) = \begin{bmatrix} G_m(s) & 0 \\ 0 & G_t(s) \end{bmatrix} \tag{34}$$

where $G_m(s)$, $G_t(s)$ are the decoupled transfer functions for the main and tail rotors subsystems respectively and given by equation (32). To obtain the decoupling matrix, we apply equations given in (Yang *et al.* 2016) to get:

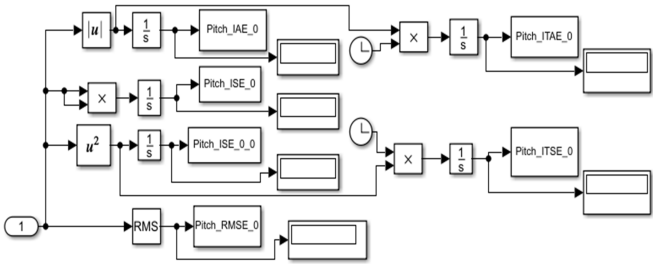
$$G_D(s) = \begin{bmatrix} G_{D11}(s) & G_{D12}(s) \\ G_{D21}(s) & G_{D22}(s) \end{bmatrix} \tag{35}$$

So that

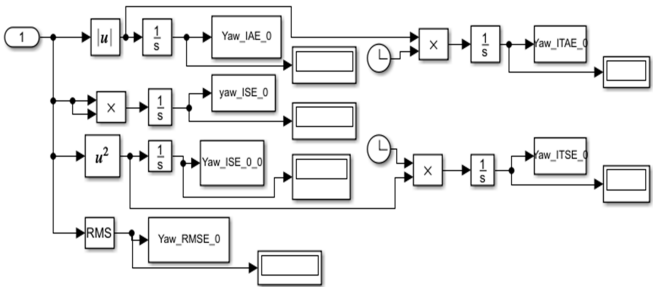
$$\begin{cases} G_{D11}(s) = \dfrac{s^3 + 0.9215s^2 + 4.78s + 3.922}{s^3 + 0.9215s^2 + 4.78s + 3.922} = 1 \\ G_{D12}(s) = 0 \\ G_{D21}(s) = \dfrac{-0.4117s^7 - 2.967s^6 - 7.116s^5 - 17.12s^4 - 23.9s^3}{s^7 + 7.255s^6 + 17.7s^5 + 42.8s^4 + 60.61s^3} \ldots, \\ \qquad\qquad \dfrac{-13.65s^2 - 2.306s + 2.47 \times 10^{-12}}{37.74s^2 + 8.17s + 1.034 \times 10^{-12}} \\ G_{D22}(s) = \dfrac{s^3 + 6s^2 + 5s}{s^3 + 6s^2 + 5s} = 1 \end{cases} \tag{36}$$



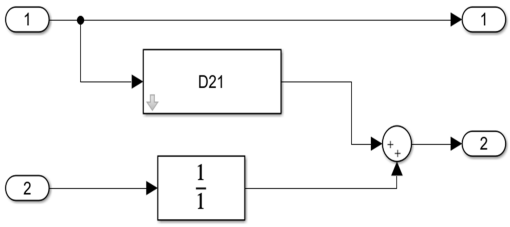
(a) Control system designed for optimizing with JSOA intelligent scheme



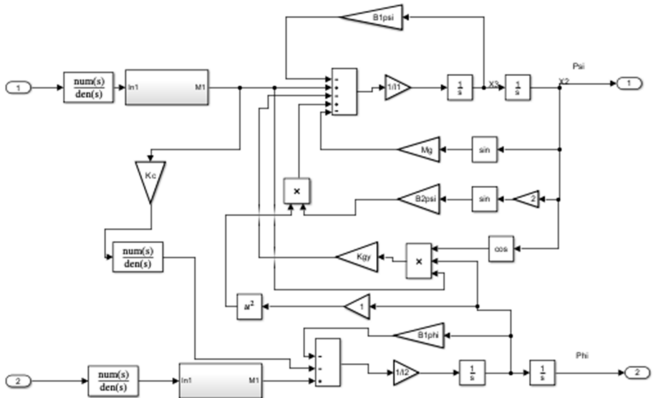
(b) Squared errors (IAE, ITAE, ISE, etc.) pitch



(c) Squared errors (IAE, ITAE, ISE, etc.) yaw



(d) Decoupler



(e) TRMS1

Figure 6 (a) Control system designed for optimizing with JSOA intelligent scheme, (b) Squared errors (IAE, ITAE, ISE, etc.) pitch, (c) Squared errors (IAE, ITAE, ISE, etc.) yaw, (d) Decoupler, (e) TRMS1

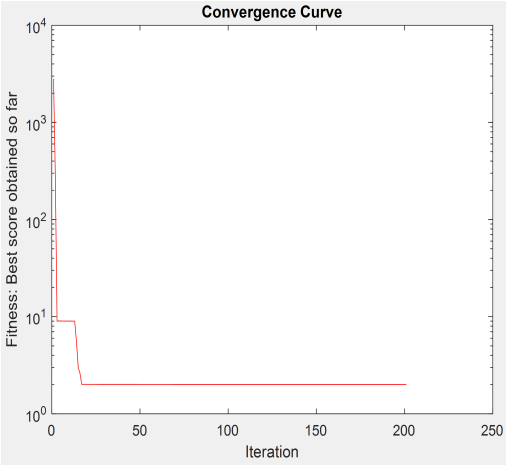
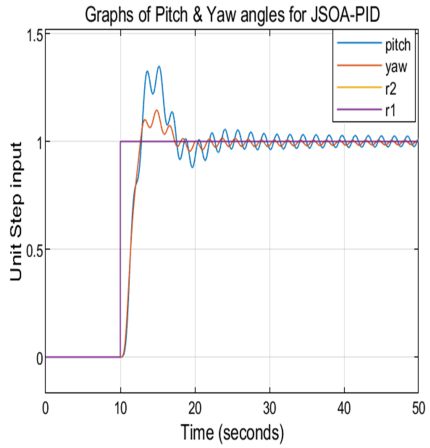
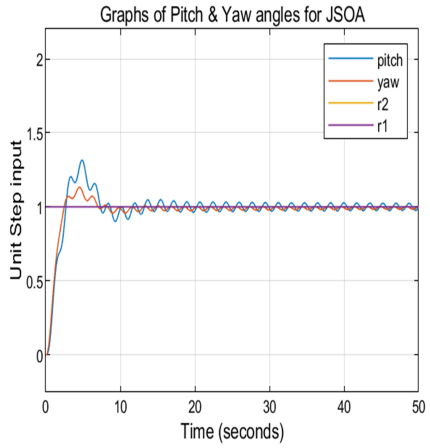


Figure 7 JSOA-PID Optimization convergence graph for 30 search agents (jumping spiders) and 200 generations

JSOA-PID Optimization Simulations Results



(a) With dead time



(b) Without dead time

Figure 8 Set-point tracking responses of pitch and yaw angles using JSOA-based controllers (a) With dead time (b) Without dead time

Multi-objective optimization algorithm was written and used with the JSOA code for faster convergence. JSOA is a slow converging procedure taking several hours or even days to converge depending on the size of the design variables, search agents/ number of iterations, population size, and other design settings used. An exponential relationship exists between number of search agents or population size and time to converge (the larger the population size, the more time required to converge). Also, another factor here is the size of the population. The visible vibrations or ripples seen on the plot resulted from the fact that the plant (i.e., TRMS) is a very stiff system as well as the significant cross-couplings. For this design, 30 search agents were employed.

Table 2 Baseline JSOA-PID controller's parameters/gains values

Parameters	K_p	K_i	K_d
Main	-1.2171	-1.2432	-2.3345
Tail	-2.3897	-0.0485	-2.026

Six (6) other methods or strategies were compared against JSOA strategy.

Performance Indices

The parameter tuning rules are based on the following performance measures: Integral of the Absolute Error (IAE), Integral of Time Absolute Error (ITAE), Integral Squared Error (ISE), Integral Time Squared Error (ITSE), and Root Mean Squared Error (RMSE). These measures or indicators are indices that show or indicate how well the system performed during the execution/run).

Objective Function

The objective function used is given by:

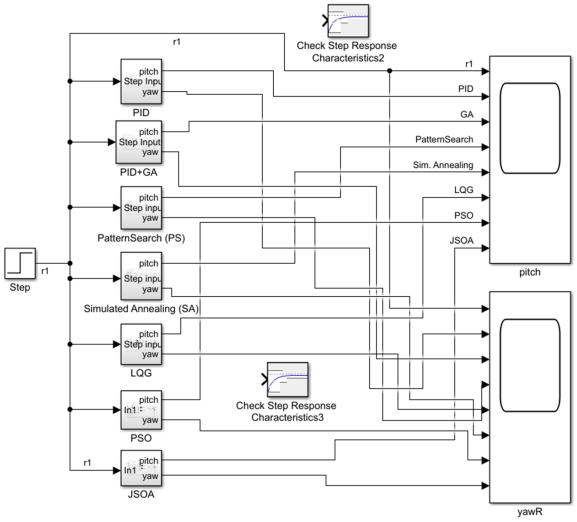
$$J(K_p, K_i, K_d, \theta, \phi) = 2 * (\omega_1 * \text{pitch (ISE)} + \omega_2 * \text{yaw (ISE)}) \quad (37)$$

where ω_1, ω_2 are some carefully chosen weighting coefficients. In this study, $\omega_1 = \omega_2 = 0.5$.

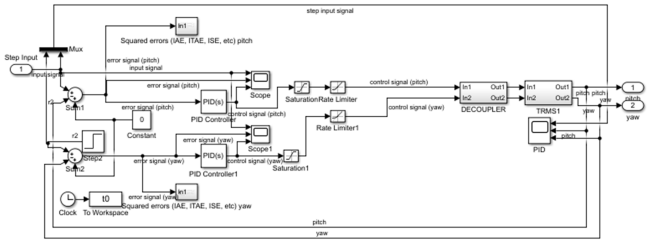
Fitness Function

The error signal minimization is achieved by the use of the PID controller via performance index minimization in the objective function value J. And we know that the smaller the J value of performance index of the corresponding search agent, the fitter the search agent will be and vice versa. Hence, J varies inversely to the fitness of the chromosomes. Therefore, the fitness of the chromosomes is defined as:

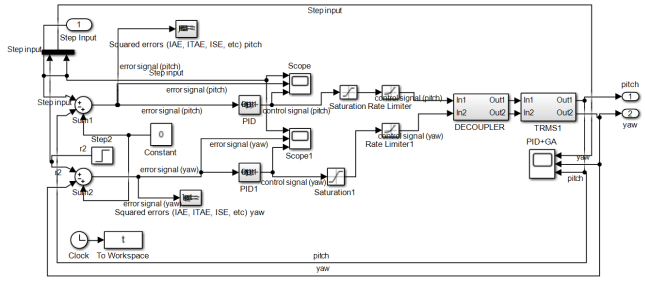
$$\text{Fitness function} = \frac{1}{J} = \frac{1}{2} * \frac{1}{\omega_1 * \text{pitch(ISE)} + \omega_2 * \text{yaw(ISE)}} \quad (38)$$



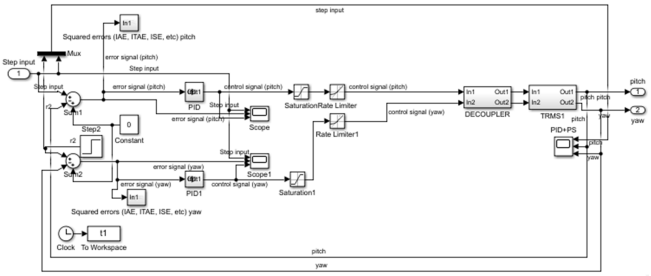
(a) Combined Simulink Model



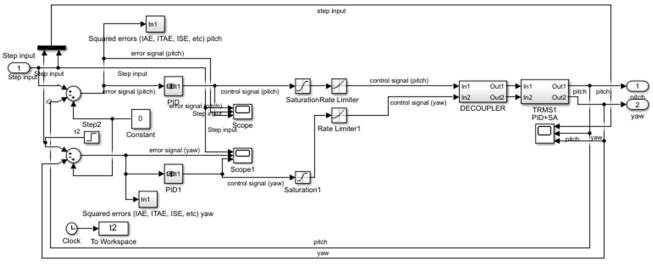
(b) PID Controllers



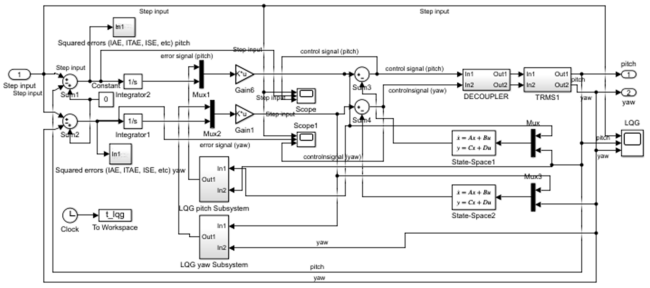
(c) PID+GA Controllers



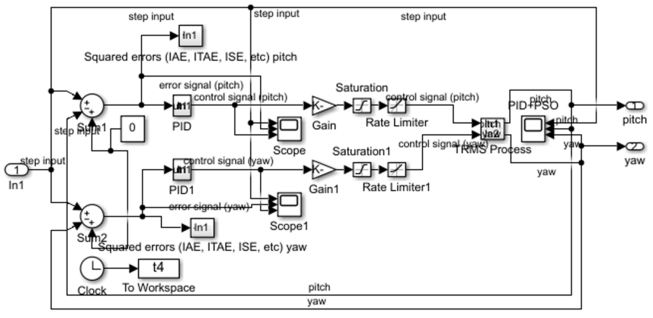
(d) PID+Pattern Search Controllers



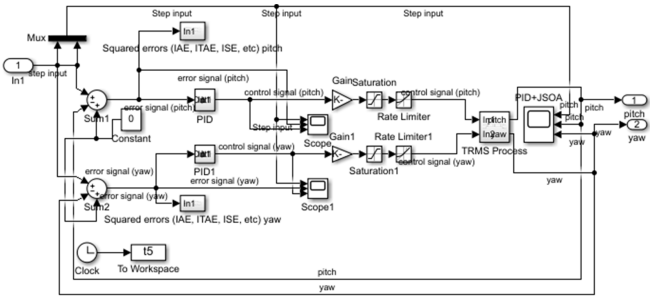
(e) PID+Simulated Annealing Controllers



(f) LQG Controllers

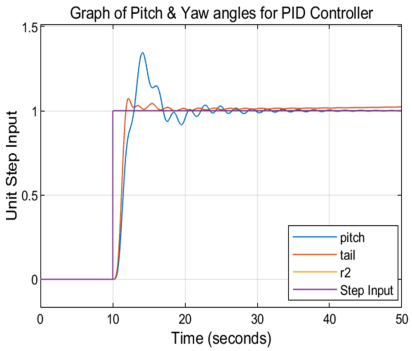


(g) PID+PSO Controllers

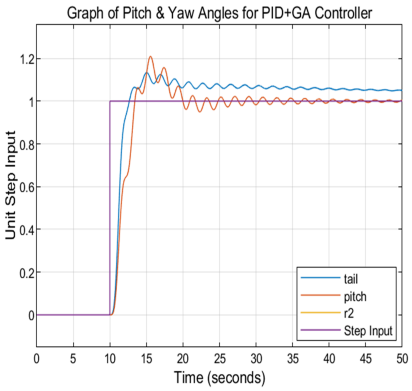


(h) PID+JSOA-based Controllers

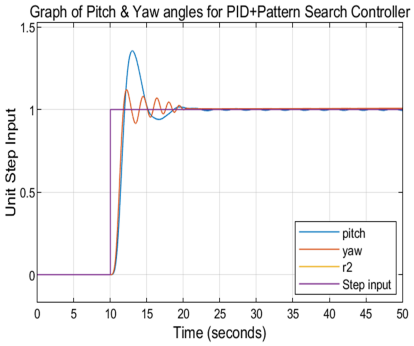
Figure 9 Set-point tracking responses of the seven methods: (a) Combined Simulink Model, (b) PID Controllers, (c) PID+GA Controllers, (d) PID+Pattern Search Controllers, (e) PID+Simulated Annealing Controllers, (f) LQG Controllers, (g) PID+PSO Controllers, (h) PID+JSOA-based Controllers



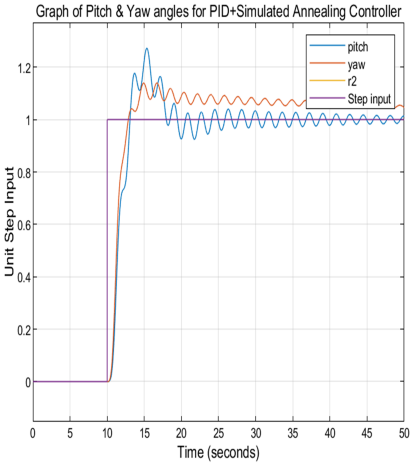
(a) Results for PID Controllers



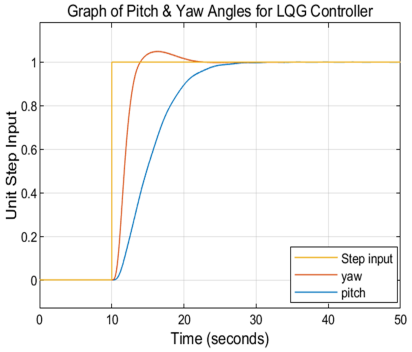
(b) Results for PID+GA Controllers



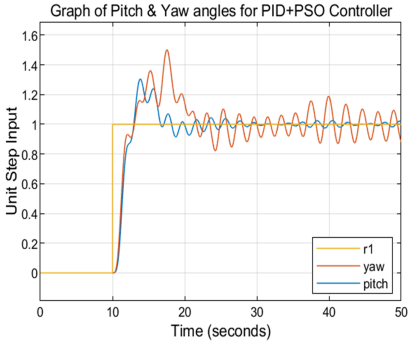
(c) Results for PID+Pattern Search Controllers



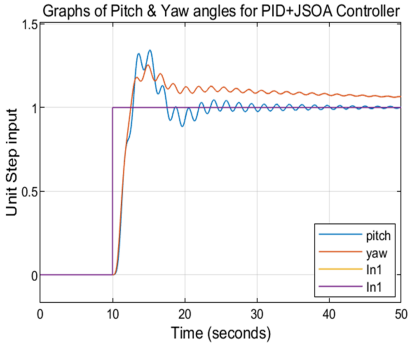
(d) Results for PID+Simulated Annealing Controllers



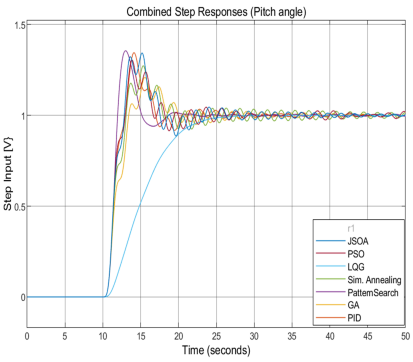
(e) Results for LQG Controllers



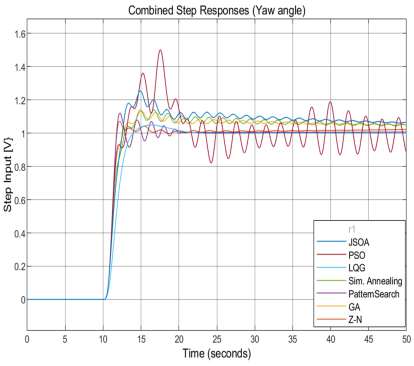
(f) Results for PID+PSO Controllers



(g) Results for PID+JSOA-based Controllers



(h) Results for Combined Pitch Angles



(i) Results for Combined Yaw Angles

Figure 10 (a) Results for PID Controllers, (b) Results for PID+GA Controllers, (c) Results for PID+Pattern Search Controllers, (d) Results for PID+Simulated Annealing Controllers, (e) Results for LQG Controllers, (f) Results for PID+PSO Controllers, (g) Results for PID+JSOA-based Controllers, (h) Results for Combined Pitch Angles, (i) Results for Combined Yaw Angles

Table 3 Summary of PID parameters for six methods

Method	Horizontal plane K_p	Horizontal plane K_i	Horizontal plane K_d	Vertical plane K_p	Vertical plane K_i	Vertical plane K_d
PID	7.5803	-0.2479	9.7221	1.7053	2.1643	3.0066
GA	4.9553	0.0693	5.2448	2.1593	1.5238	3.9310
LQG	–	–	–	–	–	–
Pattern Search	11.7969	-0.2656	19.2324	1.2267	2.5595	1.7598
Simulated Annealing	4.7778	0.0835	5.8484	2.6264	2.0038	4.5171
PSO	-8.2211	-2.6907	-19.9187	-12.8273	-12.0269	-19.9996
JSOA	-2.3897	-0.04852	-2.026	-1.2171	-1.2432	-2.3345

Table 4 Quantitative comparison of the proposed JSOA-PID controller and six other controllers design strategies with performance indices for pitch & yaw angles

Controller Method	Horizontal plane ϕ angle IAE	Horizontal plane ϕ angle ITAE	Horizontal plane ϕ angle ISE	Horizontal plane ϕ angle ITSE	Horizontal plane ϕ angle RMSE	Vertical plane θ angle IAE	Vertical plane θ angle ITAE	Vertical plane θ angle ISE	Vertical plane θ angle ITSE	Vertical plane θ angle RMSE
PID	1.787	31.32	0.997	10.72	0.02268	2.583	35.38	1.364	15.3	0.0009143
GA	3.78	87.21	1.193	15.58	0.04985	2.858	43.47	1.417	15.92	0.008896
LQG	2.068	24.49	1.384	15.02	0.000004	5.501	76.03	3.645	45.38	0.000003
Pattern Search	1.638	21.5	1.007	10.7	0.004712	2.29	30.04	1.327	14.63	0.003673
Simulated Annealing	3.946	90.81	1.243	16.47	0.04331	3.278	58.43	1.382	16.08	0.03074
PSO	4.109	76.61	1.551	20.9	0.002231	2.723	41.4	1.318	14.84	0.0008534
JSOA	2.008	14.6	0.904	0.7934	0.002157	3.138	28.47	1.194	2.041	0.003753

CONCLUSION

The score or performance of JSOA alongside other techniques used in this study are quantified in tables and graphs. It is seen that the optimum solution for the controller parameters K_p, K_i, and K_d values for the pitch are -1.2171, -1.2432, & -2.3345 while for the yaw are -2.3897, 0.0485, & -2.026 respectively. These global solutions converged after about 22 hrs of simulation time in MATLAB/Simulink software. For PID, Pattern Search and Simulated Annealing, the values obtained were: ZN: 7.5803, -0.2479, 9.7221, 1.7053, 2.1643, & 3.0066, Pattern Search: 11.7969, -0.2656, 19.2324, 1.2267, 2.5595, 1.7598; Simulated Annealing: 4.7778, 0.0835, 5.8484, 2.6264, 2.0038, 4.5171 respectively for the pitch and yaw. In terms of performance on a comparative scale, the fitness function for JSOA is 0.953288847 or 95.3%.

From the final plot (figure 9), a cursory look suggests that GA did not perform well (worst performing) while JSOA was the best performing among the optimization techniques in spite of the stiffness offered by the system, though the yawning action has an unresolved error that continued at the steady state, which is not surprising. This is because of the computational effort required (figure 4). The graphs confirm that methods like GA are unsuitable for optimizing nonlinear systems.

The physical significance of these results is that with a PID controller, the derivative gain (D) can have a different sign from the integral gain (I), in order to return a stable controller even if one or more gains are negative. This is in order for phase adjustment in the loop, if the plant is non-minimum phase. The TRMS is clearly a non-minimum phase system, having a pole or zero at the origin and so can be made asymptotically stable. This means the system

Table 5 Objective function values against Controller design strategy for the seven methods

Controller Method	Yaw plane (ϕ angle) ISE	Pitch plane (θ angle) ISE	Objective function
PID	0.997	1.364	1.1805
GA	1.193	1.417	1.305
LQG	1.384	3.645	2.5145
Pattern Search	1.007	1.327	1.167
Simulated Annealing	1.243	1.382	1.3125
PSO	1.551	1.318	1.2309
JSOA	0.904	1.194	1.049

Table 6 Fitness function (or score) against Controller Design Strategy (Method) for the seven methods

Controller Method	Yaw plane (ϕ angle) ISE	Pitch plane (θ angle) ISE	Fitness score
PID	0.997	1.364	0.847098687
GA	1.193	1.417	0.766283525
LQG	1.384	3.645	0.397693378
Pattern Search	1.007	1.327	0.856898029
Simulated Annealing	1.243	1.382	0.761904762
PSO	1.551	1.318	0.697107006
JSOA	0.904	1.194	0.953288847

veers to the opposite direction first before following the prescribed trajectory.

In this study, white box modelling approach was used, where rotational kinematics equations developed by Newton and Euler were used to accurately describe the rotational forces and moments/momentums of the TRMS in flight. A PID controller was then subsequently developed for the decoupled and linearized plant, being a nonlinear, complex, system with dynamic couplings. To obtain the optimum values for the PID controller, JSOA was employed which was able to evolve a global solution readily without having to search through the entire solution search space, and time to process increases exponentially as the size of the search agents or population size. A maximum generation with population sizes of 100 and 30 respectively were used in this study. The fitness score of 0.953288847 or about 96% showed highly fit individuals as the global solution evolved, which is excellent for any meaningful control design performance in terms of reference tracking, trajectory tracking and servo and regulatory control.

This should be 'Controlling the TRMS using conventional means is a huge challenge, particularly tuning of the gains especially when the number of parameters to be tuned is large. For a stiff system, determining the gains by trial and error, may be impractically impossible, hence the need and use of intelligent methods like GA among others. Among the many intelligent techniques, JSOA has proven to be unique, in that it is capable of evolving the global solution with relative ease, including systems and processes with very fast changing dynamics. Ultimately the main idea about using optimization techniques in scientific and engineering studies of this kind is to help in tuning some goals (parameters/variables), so as to achieve the best possible or optimum values for our overall design. This is true in all fields of human endeavours.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Agrawal, A. K., 2013 Optimal Controller Design for Twin Rotor MIMO System .

Al-Mahturi, A. and H. Wahid, 2017 Optimal Tuning of Linear Quadratic Regulator Controller Using a Particle Swarm Optimization for Two-Rotor Aerodynamical System. World Academy of Science, Engineering and Technology-International Journal of Electronics and Communication Engineering **11**: 196–203.

Basturk, H. I., 2006 Quasi-LPV Modelling and Control of Twin Rotor Multiple-Input Multiple-Output System .

Butt, S. S. and H. Aschemann, 2015 Multi-variable Integral Sliding Mode Control of a Two Degrees of Freedom Helicopter. 8th Vienna International Conference on Mathematical Modelling - MATHMOD 2015 .

Chaudhary, C. and A. Kumar, 2019a Control of Twin Rotor MIMO System using PID and LQR Controller. Journal of Aircraft and Spacecraft Technology **3**: 211–220.

Chaudhary, C. and A. Kumar, 2019b Control of Twin Rotor MIMO System using PID and LQR Controller. Journal of Aircraft and Spacecraft Technology **3**: 211–220.

Choudhary, S. K., 2017 H Optimal Feedback Control of a Twin Rotor MIMO System. International Journal of Modelling and Simulation **37**: 46–53.

Dimassi, H., S. H. Said, A. Loria, and F. M. M'Sahli, 2019 An Adaptive Observer for a Class of Nonlinear Systems with a High-Gain Approach. Application to the Twin-Rotor System. International Journal of Control **3**.

Ezekiel, D. M., S. Ravi, and O. Matsebe, 2020a Modelling of the Twin Rotor MIMO System (TRMS) Using the First Principles Approach. IEEE: 2020 International Conference on Computer, Communication, and Informatics (ICCCI) pp. 726–732.

Ezekiel, D. M., S. Ravi, and O. Matsebe, 2020b Pitch and Yaw Angular Motions (Rotations) Control of the 1-DOF and 2-DOF TRMS: A Survey. Archives of Computational Methods in Engineering .

Ghellab, M. Z., S. Zeghlache, and A. Bouguerra, 2018 Real-Time Implementation of Fuzzy Gain-Scheduled PID Controller for Twin Rotor MIMO System (TRMS). Advances in Modelling Analysis **73**: 137–149.

Instruments, F., 2000 Twin Rotor MIMO System Advanced Teaching Manual 33-007-4M5 .

Instruments, F., 2006 Twin Rotor MIMO System Control Experiments Manual: 33949S .

Juang, J.-G., R.-W. Lin, and W.-K. Liu, 2008 Comparison of Classical Control and Intelligent Control for a MIMO System. Applied Mathematics and Computation **205**: 778–791.

Klee, H. and R. Allen, 2018 Simulation of Dynamic Systems with MATLAB and Simulink .

Mondal, S., 2012a Adaptive Second Order Sliding Mode Control Strategies For Uncertain Systems .

Mondal, S., 2012b Adaptive Second Order Sliding Mode Control Strategies For Uncertain Systems .

Mondal, S. and C. Mahanta, 2012 Adaptive Second-Order Sliding-Mode Controller for a Twin Rotor Multi-Input–Multi-Output System. IET Control Theory and Applications **6**: 2157–2167.

Mones, M. and T. Diaa-Eldeen, 2017 Experimental Nonlinear Identification of a Lab-Scale Helicopter System using MLP Neural Network. 13th International Computer Engineering Conference (ICENCO) .

Peraza-Vázquez, H., A. Peña-Delgado, P. Ranjan, C. Barde, A. Choubey, *et al.*, 2022 A Bio-Inspired Method for Mathematical Optimization Inspired by Arachnida Salticidae. Mathematics **10**: 102.

Raghavan, R. and S. Thomas, 2017 Practically Implementable Model Predictive Controller for a Twin Rotor Multi-Input Multi-Output System. Journal of Control, Automation, and Electrical Systems **28**: 358–370.

Sodhi, P. and I. Kar, 2014 Adaptive Backstepping Control for A Twin Rotor MIMO System .

Tao, C. W., J. S. Taur, and Y. C. Chen, 2010 Design of a Parallel Distributed Fuzzy LQR Controller for the Twin Rotor Multi-Input Multi-Output System. Fuzzy Sets and Systems **161**: 2081–2103.

Toha, S. F. and M. O. Tokhi, 2010 Augmented feedforward and feedback control of a twin rotor system using real-coded moga. In *IEEE Congress on Evolutionary Computation*, Barcelona, Spain.

Toha, S. F. and M. O. Tokhi, 2011 PID and Inverse-Model-Based Control of a Twin Rotor System. Robotica **29**: 929–938.

Wen, P. and T.-W. Lu, 2008 Decoupling Control of a Twin Rotor MIMO System using Robust Deadbeat Control Technique. IET Control Theory & Applications **2**: 999–1007.

Yang, X., J. Cui, D. Lao, and J. Chen, 2016 Input Shaping Enhanced Active Disturbance Rejection Control for a Twin Rotor Multi-Input Multi-Output System (TRMS). ISA Transactions **62**: 287–298.

How to cite this article: Ezekiel, D. M., Samikannu, R., and Matsebe, O. Bio-Inspired Jumping Spider Optimization for Controller Tuning/Parameter Estimation of an Uncertain Aerodynamic MIMO System. *Chaos Theory and Applications*, 6(3), 205-217, 2024.