ISME

The Iranian Journal of Mechanical Engineering Transactions of ISME
Journal homepage: https://jmee.isme.ir/
Vol. 26, No. 2, 2025, pp. 93-110

Research Paper

DOI: https://doi.org/10.30506/jmee.2025.2038103.1355

Coordinated Landing Control of Multiple Vehicles using Rough Neural Network and Sliding Mode Methods

Saba Nikseresht[*]
Ph.D. Candidate

Mahdi Jafari Nadoushan[†]
Assistant Professor

This paper investigates and compares the coordinated landing of multiple vehicles using the Rough Mimetic Neural Controller (R-MNC) and Sliding Mode Controller. Coordinated landing scenarios, critical for advanced aerospace operations, require robust control strategies to handle nonlinear dynamics and ensure safe, precise landings. In the simulations, nonlinear dynamic equations of the agents are used, and control signals are allocated among system actuators based on inputs such as gamma angle, angle of attack, and altitude rate. The NSGA-II optimization algorithm tunes controller parameters to enhance performance and reduce control effort. Results demonstrate that both controllers effectively stabilize the vehicle and achieve desired outcomes, but R-MNC shows superior adaptability in dynamic environments, particularly under varying conditions. This study examines the trade-offs and complementary advantages of both methods, offering insights for designing reliable coordinated landing strategies in complex aerospace missions.

Keywords: Control allocation, Fuzzy system, Rough Neural Network, Sliding mode controller, Multi agent systems

1 Introduction

In advanced control systems for various vehicles such as airplanes, helicopters, ships, and others, enhancing safety, maneuverability, flexibility, system resilience to potential faults, and reducing control costs are of significant importance. This has led to the increasing use of control allocation algorithms across different industries. Control allocation refers to the distribution and management of control signals.

[*]Ph.D. Candidate, Faculty of Aerospace Engineering, K.N. Toosi University of Technology, Tehran, Iran, Sab.nikseresht@email.kntu.ac.ir
[†]Corresponding author, Assistant Professor, Faculty of Aerospace Engineering, K.N. Toosi University of Technology, Tehran, Iran, mjafari@kntu.ac.ir

Received: 2024/08/10, Revised: 2025/01/09, Accepted: 2025/01/15

Hence, the allocation and management of control signals produced by flight control systems among the system actuators fall under the responsibility of the control allocation section. The methods used in control allocation are divided into two categories: optimal and non-optimal methods. Optimal methods are those that can significantly reduce the desired control costs by finding the optimal solution. The challenge of these methods lies in their high computational complexity and, in some cases, achieving a solution with a certain level of error. On the other hand, non-optimal methods are generally capable of quickly finding a solution but may sometimes provide suboptimal results or the solution may be unattainable. In the present article, an optimal control allocation method is used to distribute control signals among vehicle actuators for performing coordinated landing. The use of control signal allocation methods greatly assists in improving vehicle performance during landing and takeoff processes. The main objective of this paper is to examine and compare the coordinated landing of multiple vehicle using two different control approaches based on the Rough mimetic neural controller and the sliding mode controller. The actuators involved in the coordinated landing process of the agents include the elevator and the thrust vector control system, where the optimal control signals generated by each of the aforementioned flight control systems are allocated between these two actuators. This enables the agents to achieve a coordinated landing with minimal control effort, high precision, short duration, and adequate and suitable stability. The inputs to both the Rough mimetic neural network -based flight control system and the Sliding Mode controller method include the gamma angle, angle of attack, and altitude changes of the vehicle, while their outputs include the variations in the angles of the elevator and thrust vector control actuators. To improve the performance of each flight control system, aiming to reduce the control effort of the actuators and increase the precision in coordinated landing, the NSGA-II optimization algorithm is used, and the parameters related to each of the controllers are optimized, with the parameter optimization process being detailed extensively. Subsequently, the research conducted in this field is briefly reviewed. Soriano et al. propose a novel method for collision avoidance in mobile robots using multi-agent systems to exchange information among agents, comprising phases like collision detection, obstacle identification, negotiation, and avoidance [1]. Wang et al. design a distributed control algorithm for affine formation control of multi-agent systems with prescribed convergence time using leader–follower strategies [2]. Du et al. solve the cooperative pursuit-evasion problem with a MARL-based approach, incorporating parameter sharing and curriculum learning to enhance vehicle performance in urban airspace [3]. Wang explores time-varying formation tracking for nonlinear multi-agent systems with an adaptive output-feedback controller in switching directed networks [4]. Elgohary and Moidel implement two adaptive controllers: an MPC for formation stability and an ESC for energy-efficient position optimization [5]. Wang et al. present a collision avoidance model inspired by the Local Pursuit strategy of foraging ants, combining formation control with velocity alignment for navigating obstacles [6]. Martinez-Ponce et al. study the effects of formation flight on energy efficiency and stability, highlighting its implications for long-duration and robust autonomous flight systems in applications like environmental monitoring [7]. Tabassum and Bai design an allocation agent for vehicle Detect and Avoid (DAA) systems, establishing a mixed-initiative framework to dynamically manage control authority in communication-limited environments [8]. Cao et al. propose a cloud-based predictive control protocol to address communication delays in networked multi-agent systems, ensuring formation control by proactively compensating for network delays. Their analysis establishes conditions for stability and consensus in closed-loop systems [9]. Kang et al. introduce a second-order differentiable virtual force metric for local trajectory planning, integrating artificial potential field forces to dynamically adjust formations. They develop a distributed trajectory optimization framework that combines obstacle avoidance with dynamic feasibility, providing a multi-agent consensus strategy that outperforms traditional methods [10]. The objective of the present research is to compare the efficiency and accuracy of two

flight control systems, namely the Rough Mimetic Neural Controller and the Sliding Mode Controller, in the process of automatic landing of multiple vehicle. The present study will address the following aspects:
1. The utilization of a fuzzy controller in the training of the Rough mimetic neural flight control system, and the training of the Rough neural network parameters based on the error between the fuzzy controller output and the neural network output.
2. The genetic algorithm will be used to optimize the parameters of the fuzzy controller and the Sliding Mode Controller. This will enhance control accuracy for proper stability, reduce the landing time of the agents, increase accuracy, and reduce the control effort of the actuators synchronized in the automatic landing process of the agents.
3. The selection of the gamma angle, angle of attack, and altitude variations of the vehicle as inputs for both the Rough mimetic neural network and Sliding Mode control systems is also crucial.
4. The aerodynamic lift actuator and the Thrust Vector Control system are selected as the participating actuators in the coordinated landing of the agents. This paper utilizes both types of actuators in a collaborative and simultaneous manner, which is a departure from many studies that use either aerodynamic actuators or thrust vector control actuators separately for vehicle control. Furthermore, the control signals are optimally allocated between them.
Subsequently, a comparative analysis of the results obtained from both the Rough mimetic neural and Sliding Mode longitudinal flight control systems is conducted to assess the performance and stability of each agent in the landing process. In the second section, the equations and mathematical relationships of the vehicle are presented. In the third section, the design of the longitudinal flight control systems is addressed, and in the fourth section, the simulation results are examined.

2 Mathematical model

2.1 Dynamics modeling

This section deals with extracting the equations of motion for agents. Our equations of motion consist of a set of nonlinear differential equations with six degrees of freedom for each vehicle. however, vehicle equations of motion are expressed as nonlinear equations with six degrees of freedom [11], divided into rotational and translational equations. It is assumed that changes in mass (m) and moments of inertia of the vehicle are negligible over the entire process. Rotational dynamics equations are calculated as follows:

$$\begin{aligned} &\dot{p}I_{xx} - I_{xz}(\dot{r} + pq) + \left(I_{zz} - I_{yy}\right)rq = L_A + L_T \\ &\dot{q}I_{yy} + pr(I_{xx} - I_{zz}) - (p^2 - r^2)I_{xz} = M_A + M_T \\ &\dot{r}I_{zz} - \dot{p}I_{xz} + pq\left(I_{yy} - I_{xx}\right) + rq\, I_{xz} = N_A + N_T \end{aligned} \tag{1}$$

$I_{xx}, I_{yy}, and\ I_{zz}$ are the moments of inertia about the three axes of the body-fixed coordinate system, and $I_{xy}, I_{yz}, and\ I_{xz}$ are the products of inertia. Additionally, p, q, and r are the angular velocities about the body-fixed coordinate axes, and L, M, and N represent the torque acting on the vehicle along the X-axis, Y-axis, and Z-axis, respectively. Index A refers to aerodynamic torques, and index T represents torques resulting from thrust forces. The force relationships in the body-fixed coordinate system are calculated as follows [11]:

$$\begin{aligned} &m(\dot{u} - vr + wq) = -mgsin\theta + F_{A_x} + F_{T_x} \\ &m(\dot{v} + ur - wp) = mgsin\phi.cos\theta + F_{A_y} + F_{T_y} \\ &m(\dot{w} - uq + vp) = mgcos\phi.cos\theta + F_{A_z} + F_{T_z} \end{aligned} \tag{2}$$

u, v, w represent linear velocities in the body-fixed coordinate system, and F_x, F_y, and F_z represent the forces acting on the vehicle along the *X*-axis, *Y*-axis, and *Z*-axis, respectively. Index *A* refers to aerodynamic forces, and index *T* represents thrust force. Ttranslational dynamics equations are calculated as follows [11]:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}=\begin{bmatrix}cos\psi & -sin\psi & 0\\ sin\psi & cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}cos\theta & 0 & sin\theta\\ 0 & 1 & 0\\ -sin\theta & 0 & cos\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & cos\,\phi & -sin\phi\\ 0 & sin\phi & cos\phi\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{3}$$

x, y, and z, are the vehicle 's position, The Euler angle equations in matrix form are calculated as follows [11]:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & sin\phi\ tan\theta & cos\phi\ tan\theta\\ 0 & cos\,\phi & -sin\phi\\ 0 & sin\phi\ sec\theta & cos\phi\ sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{4}$$

Euler angles, denoted by ϕ (roll), θ (pitch), and ψ (yaw), describe the rotation of a vehicle around its principal axes x, y, and z, respectively.

2.2 Aerodynamic model

The aerodynamic model is predicated on a set of equations and parameters that represent the effects of airflow on the structure of the vehicle and the forces acting upon it. The aerodynamic model determines aerodynamic coefficients for various components of the vehicle, including the wings, tail, and fuselage. This model facilitates the calculation of aerodynamic forces and moments, which are critical inputs in various analyses and simulations related to flight dynamics, control design, and system optimization. The aerodynamic model can be described as follows [12]:

$$\begin{aligned} C_D &= C_{D_0}+C_{D_{\alpha^2}}\alpha^2+C_{D_\alpha}\alpha+C_{D_q}q\\ C_L &= C_{L_0}+C_{L_\alpha}\alpha+C_{L_q}q+C_{L_{\delta_{ele}}}\delta_{ele}\\ C_Y &= C_{Y_0}+C_{Y_\beta}\beta+C_{Y_p}p+C_{Y_r}r+\left(C_{Y_{\delta_{ail_\alpha}}}\alpha+C_{Y_{\delta_{ail}}}\right)\delta_{ail}+\left(C_{Y_{\delta_{ail_\alpha}}}\alpha+C_{Y_{\delta_{rud}}}\right)\delta_{rud} \end{aligned} \tag{5}$$

$$\begin{aligned} C_m &= C_{m_0}+C_{m_\alpha}\alpha+C_{m_q}q+C_{m_{\delta_{ele}}}\delta_{ele}\\ C_l &= C_{l_\alpha}\alpha+C_{l_\beta}\beta+C_{l_p}p+C_{l_r}r+\left(C_{l_{\delta_{ail_\alpha}}}\alpha+C_{l_{\delta_{ail}}}\right)\delta_{ail}+\left(C_{l_{\delta_{rud_\alpha}}}\alpha+C_{l_{\delta_{rud}}}\right)\delta_{rud}\\ C_n &= C_{n_\alpha}\alpha+C_{n_\beta}\beta+C_{n_p}p+C_{n_r}r+\left(C_{n_{\delta_{ail_\alpha}}}\alpha+C_{n_{\delta_{ail}}}\right)\delta_{ail}+\left(C_{n_{\delta_{rud_\alpha}}}\alpha+C_{n_{\delta_{rud}}}\right)\delta_{rud} \end{aligned} \tag{6}$$

Where C_D, C_L and C_Y are the aerodynamic force coefficients, C_m, C_l and $C_n$are the aerodynamic moment coefficients, and $\delta_{rud}, \delta_{ele}, \delta_{ail}$ are rudder angle, elevator angle, and aileron angle, respectively. The angle of attack is denoted by α, and the sideslip angle of the vehicle is represented by β.

3 Longitudinal flight control system design

The longitudinal flight control system is a foundational component of a vehicle's control system, responsible for managing flight movements along the longitudinal axis.

This system, by regulating forces and moments associated with longitudinal motions such as pitch, altitude, and speed, plays a crucial role in ensuring the stability and flight control of the vehicle. Ensuring the proper functioning of the longitudinal flight control system is paramount to maintaining the vehicle in a stable state and regulating changes in altitude and speed appropriately. This system, through continuous monitoring of the vehicle's status and the use of sensors, is able to make corrective actions to ensure that the vehicle remains within the desired flight parameters.

3.1 Fuzzy controller

A fuzzy controller is a type of artificial intelligence controller that operates on the basis of fuzzy logic. In this controller, inputs and outputs are defined as fuzzy values rather than precise values. However, it is important to note that the input and output values of the actual controlled systems are defined as precise values. Consequently, the fuzzification method is employed to transform the accurate values into fuzzy values. The transformation process can be delineated into two distinct phases: initial phase—the determination of the fuzzy domain; secondary phase—the actual fuzzification process. The operation of the controller is facilitated by a set of fuzzy rules, which are either manually defined by humans or automatically generated. The employment of fuzzy logic in this controller facilitates its adaptation to dynamic and nonlinear input environments, enhancing its performance in uncertain and noisy conditions. The employment of fuzzy controllers is particularly advantageous in scenarios where a precise model of the system is not available or when the system is complex and nonlinear. In the context of designing a fuzzy flight control system, it is imperative to ascertain the input variables of the controlled system (i.e., the vehicle) along with the control variables. Given that the fuzzy controller governs the longitudinal vehicle control during the landing process with two control operators (elevator and thrust vector control), it is essential to specify appropriate variables as inputs to the fuzzy controller. For this purpose, variables α, γ, and $\dot{h}$, representing angle of attack, pitch angle, and rate of altitude change at each moment, respectively, are chosen as the inputs to the controller, while δ_T and δ_{ele} will be the outputs of the fuzzy controller (where δ_{ele} represents elevator angle changes and δ_T represents thrust vector angle changes). Seven membership functions are used to describe each of the inputs α, γ, and $\dot{h}$, defined as Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). The membership functions (μ) of input variables and outputs are selected to be Gaussian. Gaussian membership functions are one of the common functions in fuzzy systems for representing fuzzy sets. These functions are defined by their center and their width (variance or standard deviation). The Gaussian function is defined as follows:

$$\mu = exp(-\frac{(x-c)^2}{2\sigma^2}) \tag{7}$$

In the above relation, x is the input variable, c is the center of the Gaussian membership function, and σ is the standard deviation that determines the width of the function. Gaussian membership functions are commonly used in fuzzy systems due to their noise tolerance and simplicity in relationships. Because of the similarity between the input and output membership functions, it is sufficient to illustrate just one example. In Figure (1), the membership function corresponding to the gamma variable (γ) is depicted. The fuzzy controller's rule base is presented in Table (1). Accordingly, forty-nine rules are defined for the Thrust Vector Control actuator, of which only ten rules are mentioned here. The rule base for the horizontal stabilizer actuator is similar to the rules in Table (1), and thus, it is omitted.

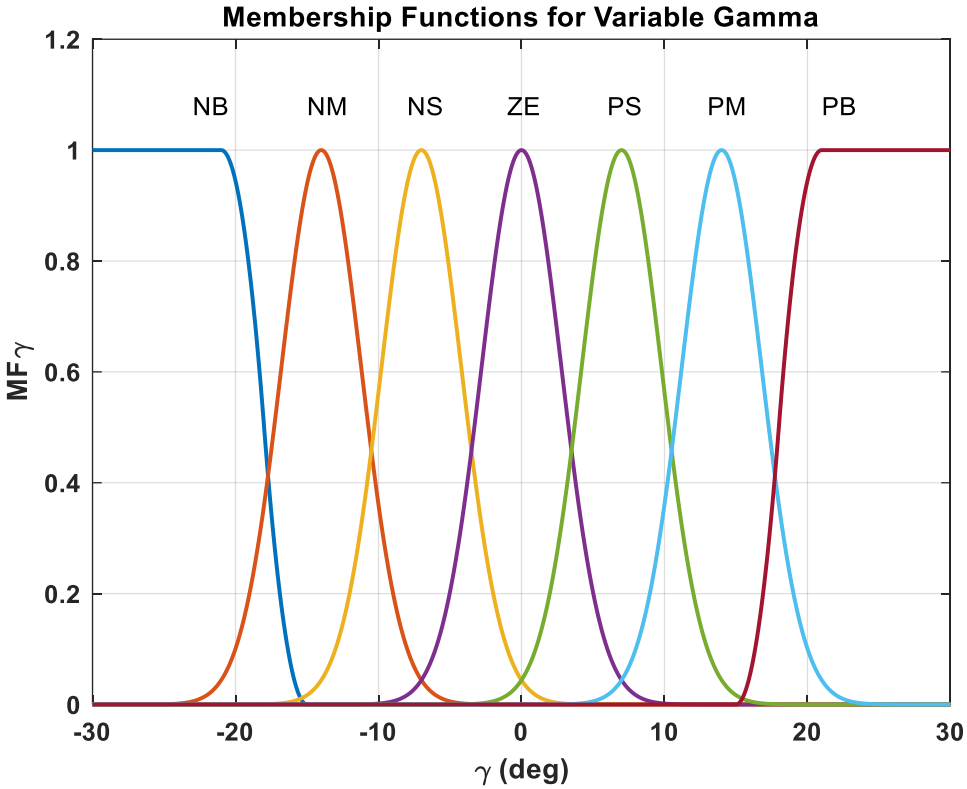


Figure 1 Membership function for variable Gamma (γ)

Table 1 Rule base related to the thrust vector control

Rul. no	$\dot{h}$	γ	U_{TVC}
1	PB	ZE	PL
2	ZE	NB	NL
3	NB	ZE	NL
4	ZE	PB	PL
5	PM	ZE	PM
6	ZE	NM	NM
7	NM	ZE	NM
8	ZE	PM	PL
9	PS	ZE	PS
10	NS	NB	NXL

3.2 Rough mimetic neural controller

The term Mimetic Neural Controller refers to a controller designed to learn and replicate the functionality of another controller within a system. As depicted in Figure (2), the Mimetic neural controller operates in parallel with another controller, specifically a fuzzy controller in this context, to emulate its behavior similarly to an identifier. In Figure (2), the variables are defined as follows: u_{nn} represents the neural controller's output, u_f denotes the fuzzy controller's output, u_e is the error signal, y_p is the system output, and y_d is the desired output. The fuzzy controller acts as an expert system, with the neural network attempting to approximate its internal nonlinear functions. By mimicking the fuzzy controller's behavior, the neural network

can eventually control the system independently and potentially replace the fuzzy controller after adequate training.

The error between the neural networks and the fuzzy controller's outputs is used to adjust the neural network's parameters, meaning the input data does not directly influence the learning process. However, since the neural controller works in parallel with the fuzzy controller, it ultimately approximates the fuzzy controller's performance upon completing the learning process. In practical systems, data is often subject to uncertainty, where input and output data are mixed with noise and disturbances. Additionally, most systems are nonlinear, making it challenging for conventional neural networks to provide accurate approximations. To address these challenges, Rough neural networks are introduced, offering resistance to uncertainty and noise. The primary difference between these networks and traditional multi-layer perceptron neural networks lies in the number of neurons and the nature of connection weights. Instead of using fixed weights to handle uncertainty, Rough neural networks utilize interval weights, and Rough neurons replace definite neurons. This approach is applied to all network layers.

Figure (3) illustrates Rough neurons and interval weight connections, where U represents the upper part of the neuron and L the lower part. Inputs enter both the upper and lower bound neurons through interval weights. The outputs of these neurons, denoted as O_U and O_L , are combined using weight coefficients or averaging. The initial weights are chosen such that the upper bound weight exceeds the lower bound weight. In this article, the Rough mimetic neural controller is used to learn the behavior of the fuzzy controller. The forward-backward algorithm is presented according to the following relationships. The structure of the Rough mimetic neural network consists of two layers, including the middle layer and the output layer. The neurons in the middle layer are of the Rough type. The number of neurons in each layer varies based on the type of time series and the activation function. The relationships for the forward part are as follows [13]:

$$\begin{aligned} net_{Uj}^{1}(k) &= \sum_{i=1}^{n0} w_{U}^{1}(k)\, x(k) + w_{Ubj}^{1}(k) \\ net_{Lj}^{1}(k) &= \sum_{i=1}^{n3} w_{L}^{1}(k)\, x(k) + w_{Lbj}^{1}(k) \\ net_{j}^{2}(k) &= \sum_{i=1}^{n1} w_{ji}^{2}(k)\, o_{i}^{1}(k) + w_{bj}^{2}(k) \end{aligned} \tag{8}$$

$w^1{}_{Uj}$ and $w^1{}_{Lj}$ are the upper and lower bound weights related to the first layer, w^2 are the weights related to the second layer. Also, $w^1{}_{bUj}$ and $w^1{}_{bLj}$ are the upper and lower bound biases of the first layer, $w^2{}_b$ are the biases related to the second layer.

$$\begin{aligned} o_{Uj}^{1}(k) &= max(f_{j}^{1}(net_{Lj}^{1}), f_{j}^{1}(net_{Uj}^{1})) \\ o_{Lj}^{1}(k) &= min(f_{j}^{1}(net_{Lj}^{1}), f_{j}^{1}(net_{Uj}^{1})) \\ o_{j}^{1}(k) &= \frac{o_{Uj}^{1} + o_{Lj}^{1}}{2} \\ o_{j}^{2}(k) &= f^{2}\left[net_{j}^{2}(k)\right] = net_{j}^{2}(k) \end{aligned} \tag{9}$$

In which $o_j^1(k)$ corresponds to the output of the *j*th neuron in the first layer, $o_j^2(k)$ corresponds to the output of the *j*th neuron in the second layer. $f_j^1\left(net_{Lj}^1\right)$, $f_j^1(net_{Uj}^1)$ are the internal functions of the upper bound and lower bound neurons of the neuron in the first layer,

$f^2\left[net_j^2(k)\right]$ is the internal function of the jth neuron in the second layer. To train the parameters of the Rough neural network, the Levenberg-Marquardt algorithm is used, according to the following relation.

$$w(k+1) = w(k) - \eta(J^T(k)\,J(k) + \mu(k)I)^{-1}\,J^T(k)e(k) \tag{10}$$

In Eq. (10), J^T Jacobian matrix, η learning rate in neural network and e is the network error. The network error is equal to the difference between the control signal of the fuzzy controller u_f and the output of the Rough neural network o_3 (or the neural controller signal u_{nn}). The results of training the rough neural network using input and output data from the fuzzy controller for the thrust vector control operator are illustrated in Figures (4) and (5), respectively.

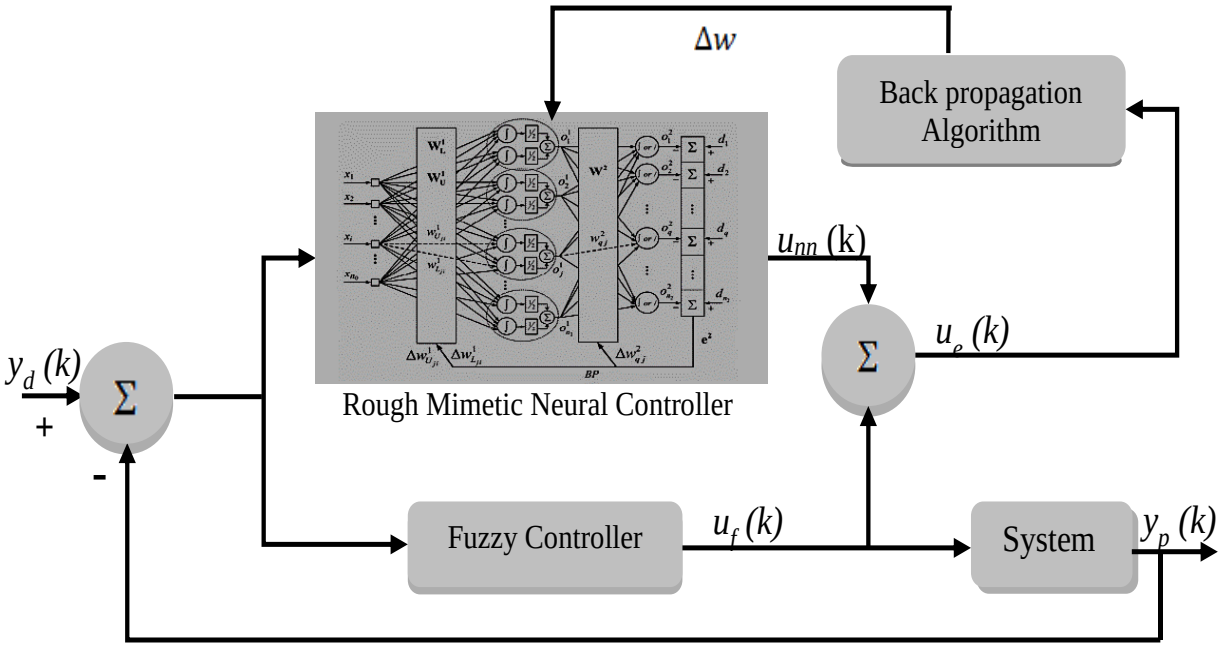


Figure 2 Rough mimetic neural controller [13]

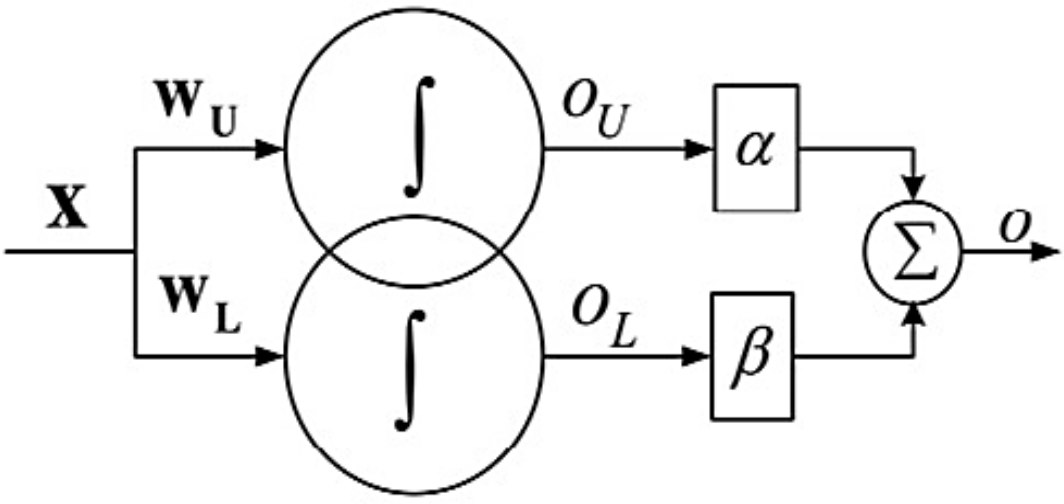


Figure 3 Rough neurons [13]

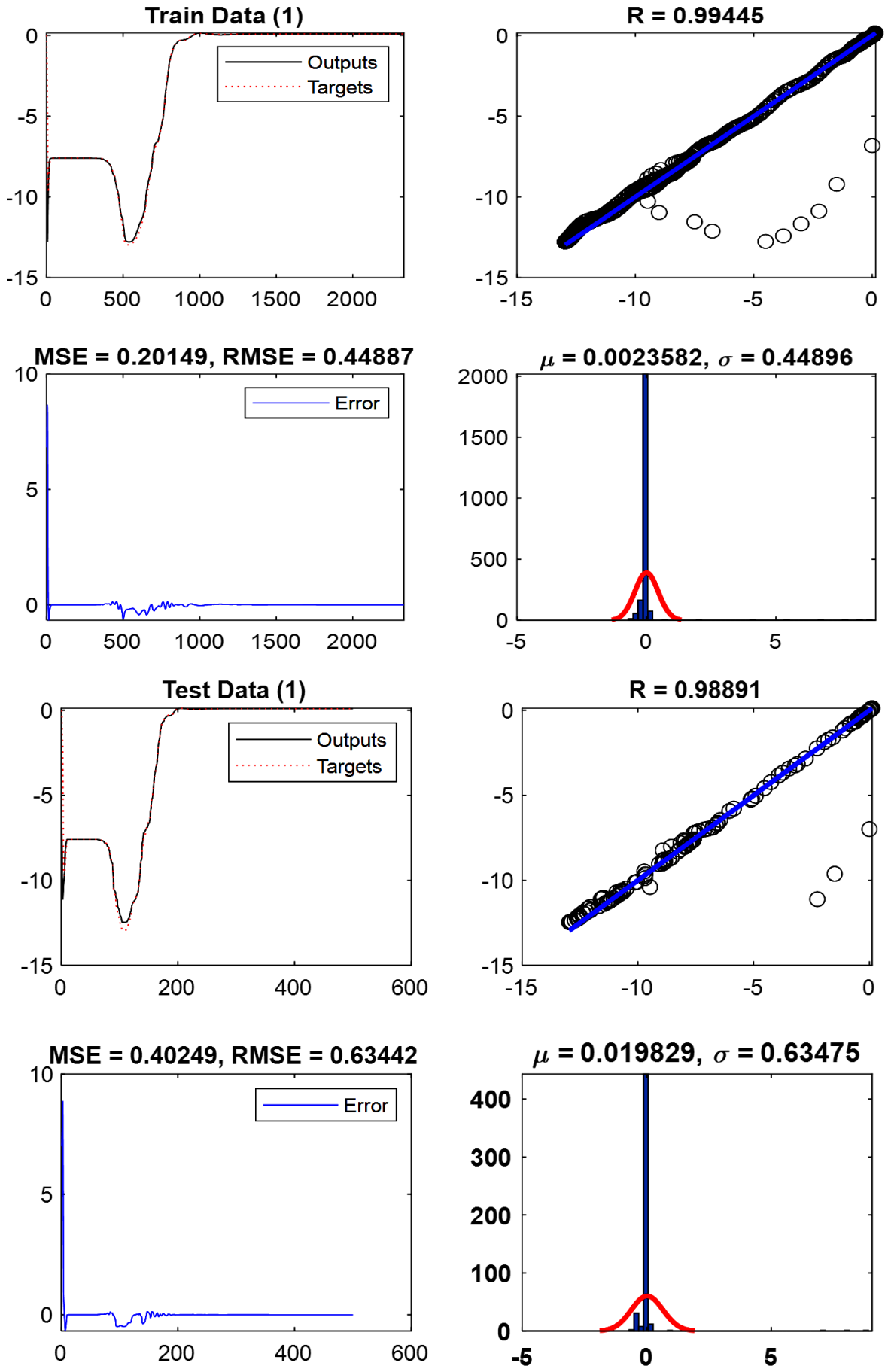


Figure 4 The results of training the Rough neural network

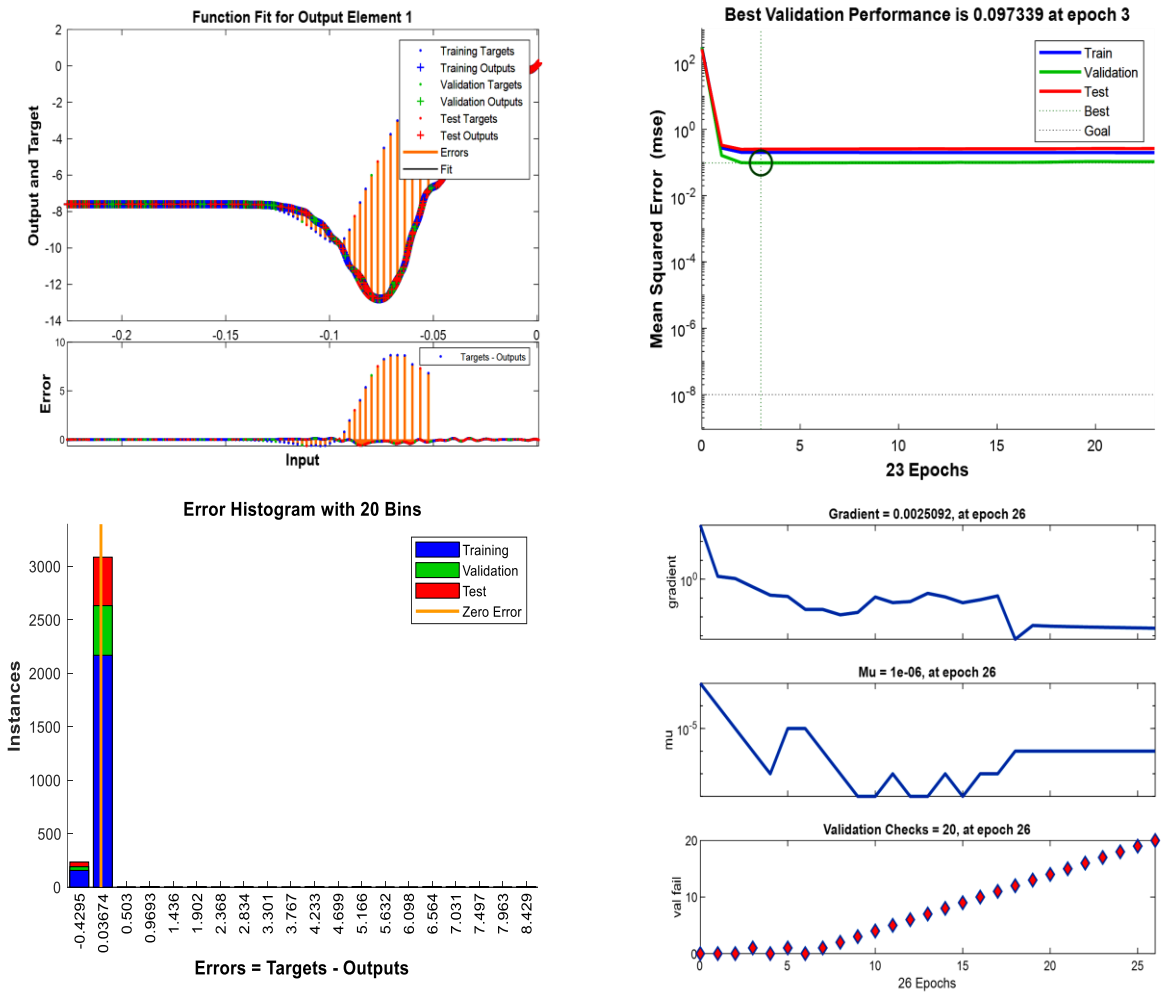


Figure 5 The results of training the Rough neural network

3.3 Sliding mode controller

Sliding mode control constitutes a specialized approach within the broader framework of variable structure control. This methodology is frequently employed to manage nonlinear dynamic systems, with the objective of ensuring their operation aligns with a predetermined reference value. This technique involves defining the reference value as a solution to a set of equations known as sliding or switching functions. To ensure effective tracking of the reference value, it is imperative that the number of variables to be monitored aligns with the number of control inputs available. The selection of an appropriate reaching law or dynamics facilitates the formulation of a stable control law. This control law determines the inputs necessary to drive the sliding functions toward their target values, thereby enabling the system (e.g., a vehicle) to adhere to the reference trajectory. It is noteworthy that, with suitable gain settings and reaching laws, the controller can be designed to be robust against unknown but bounded model uncertainties and external disturbances. In the context of vehicle landing, the problem is framed as a reference value tracking challenge, with the underlying equations often expressed in a general autonomous nonlinear form.

$$\dot{x} = g(x, u) \tag{11}$$

In this scenario, x represents the state vector and u denotes the control input vector. For the selected vehicle model, the motion equations are nonlinear but affine with respect to the control inputs. These equations can be expressed as follows [14]:

$$\dot{x} = A(x) + B(x)u \tag{12}$$

The initial step in developing a sliding mode controller involves defining a set of sliding functions, where the number of these functions matches the number of available control inputs [14]:

$$s = [s_1, s_2, \dots, s_m]^T \tag{13}$$

Typically, the sliding functions are selected as differential equations that describe the error dynamics, taking the form:

$$s_i = (\frac{d}{dt} + \lambda_i)^{k-1} e_i \quad , \qquad i = 1,2, \dots, m \tag{14}$$

Here, e_i represents the error, λ_i is a positive constant, and k is the smallest integer such that the matrix $\frac{\partial s}{\partial x}$B(x) is non-singular. For a vehicle model, the state and control input vectors are defined as follows:

$$x = [\alpha, \gamma, \theta, h, x, q]^T \tag{15}$$

$$u = [F, M]^T \tag{16}$$

By selecting the flight path angle (γ), angle of attack (α), and rate of altitude ($\dot{h}$) as the variables of interest, the error vector can be defined as:

$$e = \begin{bmatrix} \gamma - \gamma_d \\ \alpha - \alpha_d \\ \dot{h} - \dot{h}_d \end{bmatrix} \tag{17}$$

In this context, the subscript *d* indicates the desired state variables. Given the chosen error vector, *k = 2* in Eq. (7). By expanding the error dynamics equations, the sliding functions can be derived as follows:

$$s = \begin{bmatrix} s_\gamma \\ s_\alpha \\ s_{\dot{h}} \end{bmatrix} = \begin{bmatrix} (\dot{\gamma} - \dot{\gamma}_d) + \lambda_\gamma(\gamma - \gamma_d) \\ (\dot{\alpha} - \dot{\alpha}_d) + \lambda_\alpha(\alpha - \alpha_d) \\ (\ddot{h} - \ddot{h}_d) + \lambda_{\dot{h}}(\dot{h} - \dot{h}_d) \end{bmatrix} \tag{18}$$

where λ_γ, λ_α, $\lambda_{\dot{h}}$ and are positive constants. γ_d, α_d, $\dot{h}_d$, $\dot{\gamma}_d$, $\dot{\alpha}_d$, $\ddot{h}_d$ are also reference values for each of the respective variables. The control law is obtained according to the following relationship:

$$s\dot{s} = -\varepsilon s * \tanh(s - \delta) \tag{19}$$

3.4 NSGA-II optimization method

To enhance the performance of the fuzzy controller for better training of the Rough mimetic neural controller and the sliding mode controller, the parameters of both flight control systems are optimized using the genetic algorithm. The objectives of this optimization are to achieve

optimal responses in such a way that control effort and costs are minimized while reducing undesired vehicle deviations. Ultimately, all three vehicle achieve coordinated landings with suitable stability and sufficient accuracy. For this purpose, the centers of the membership functions ($\bar{x}_0^l$ ،$\bar{y}_0^l$) and deviation from the standard deviation σ of the fuzzy system inputs and outputs are optimized. The stages of optimizing the Sliding Mode controller parameters are as follows. To determine a set of optimal solutions that minimize both cost and control effort by fine-tuning the Sliding Mode controller parameters (λ_γ, λ_α, $\lambda_{\dot{h}}$, δ_γ, δ_α, $\delta_{\dot{h}}$, ε_γ, ε_α, $\varepsilon_{\dot{h}}$), the system is expected to perform effectively in maintaining (q، α، γ، h and $\dot{h}$) at their desired values. This optimization aims to reduce undesirable deviations of the vehicle and ensure acceptable stability during descent. To optimize the parameters of both the fuzzy flight control system and the sliding mode system, the following cost functions are defined:

$$J_1 = \int (\delta_e)^2 \, dt$$
$$J_2 = \int (\delta_T)^2 \, dt \tag{20}$$

The multi-objective optimization problem, taking constraints into account, is formulated as follows [12]:

$$S = Min\,\{J_1, J_2\} \tag{21}$$

$$S.t. \begin{cases} |x_i| < X_i \\ |\delta_{ele}| < \delta_{ELE} \\ |\delta_T| < \delta_{TB} \\ |sse_i| < SSE_i \\ po_i < PO_i \\ tr_i < TR_i \end{cases} \tag{22}$$

in which S represents a set of optimal solutions. x_i is the deviation of state variables i around the equilibrium point, and X_i is its maximum allowable value. δ_{ELE} and δ_T also indicate the maximum deviation of the angles of the elevator and thrust vector, respectively. tr_i and TR_i represent the rise time and its maximum value, po_i and PO_i express the percentage of maximum overshoot from the desired value and its upper limit, and sse_i and SSE_i denote the steady-state error and its maximum allowable value for each state variable i. The optimization problem constraints are presented according to Table (2). The Genetic algorithm settings for optimizing the parameters of the controller are provided in Table (3).

Table 2 Optimization problem constraints

Parameter	Value	Parameter	Value
X_θ	[-0.5, 0.5] deg	δ_E	[-20, 20] deg
X_γ	[-0.5, 0.5] deg	δ_T	[-20, 20] deg
X_{n_z}	[-1, 1]	X_q	[-0.5,0.5] deg/s
PO	10%	$X_{\dot{h}}$	[-0.5, 0.5] m
SSE	≤ 0.02	X_h	[-0.5, 0.5] m
TR	≤ 30 s	X_α	[-5, 5] deg

Table 3 Genetic algorithm parameter settings

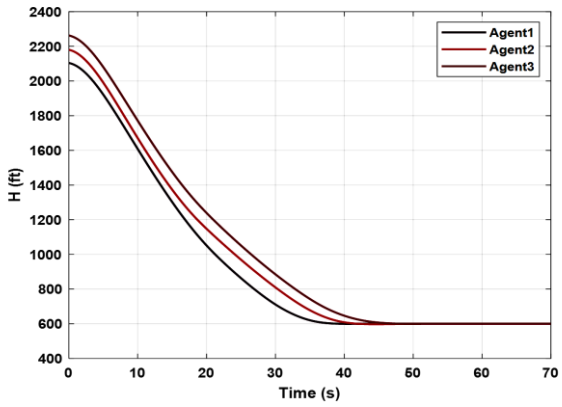
Parameter	Value
N_{pop}	70
I_{max}	1000
M_p	0.06
P_m	0.10
η_c	15
η_m	15

4 Results and discussion

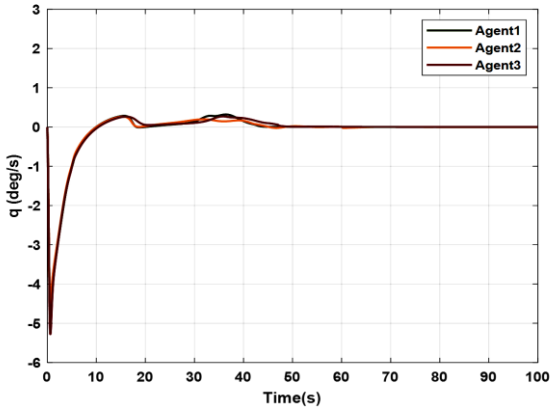
In this section, the simulation results are presented to evaluate the effectiveness and capability of the rough mimetic neural controller and the sliding mode controller in the coordinated landing process. Figure (6) presents the diagrams associated with the Rough mimetic neural controller, while Figure (7) presents the diagrams associated with the sliding mode controller. Figure (8) presents the overall schematic of the longitudinal flight control system for the coordinated landing process. Figure (9) compares the performance of the two flight control systems.

As illustrated in Figure (6), the coordinated landing of agents utilizing the Rough mimetic neural flight control system is examined. As illustrated in panel (a), the three agents initiate the automated landing procedure from varying initial altitudes. The Rough mimetic neural flight control system then stabilizes the agents and inputs control signals to achieve positional consensus with minimal error and without oscillations. The agents reach their predetermined positions at the designated altitude with the appropriate stability and in a relatively brief timeframe, landing in a coordinated manner. The subsequent panel (b) of Figure (6) illustrates the pitch rate of the agents throughout the mission. The desired pitch rate range is approximately zero, and the application of the control signal causes each agent's pitch rate to converge to zero and remain stable. Panel (c) of Figure (6) displays the velocities of each agent. As depicted, the agents have successfully achieved a consensus on their velocity. Panel (d) of Figure (6) illustrates each agent's flight path angle. It is imperative to note that the flight path angle is a critical flight variable for achieving a coordinated landing. Any deviation in the final value of this variable can potentially disrupt the coordinated landing process. Consequently, the controller must effectively manage this flight variable. The desired value for the flight path angle is approximately zero. As previously mentioned, the Rough mimetic neural controller successfully converges each agent's flight path angle to the desired value of zero, thereby placing the agents on the optimal horizontal path. As illustrated in panel (a), the thrust vector control operator angle during the mission for each agent is displayed. It is noteworthy that the thrust vector control actuator angle remains within the prescribed range of -15 to 15 degrees throughout the mission, in accordance with the defined allowable operator angle range. As illustrated in Panel (b) of Figure (6), the elevator operator angle for the agents is maintained within the defined allowable range, consistent with Panel (a). Figure (7) examines the coordinated landing of agents using the Sliding Mode Flight Control System. As illustrated in panel (a), the three agents initiate the automated landing process from varying initial altitudes. The Sliding Mode Flight Control System stabilizes the agents and inputs control signals to achieve positional consensus with minimal error. The agents reach the agreed positions at the specified altitude and land in a coordinated manner. However, the presence of the chattering phenomenon inherent in this controller results in small oscillations around the agreed altitude,

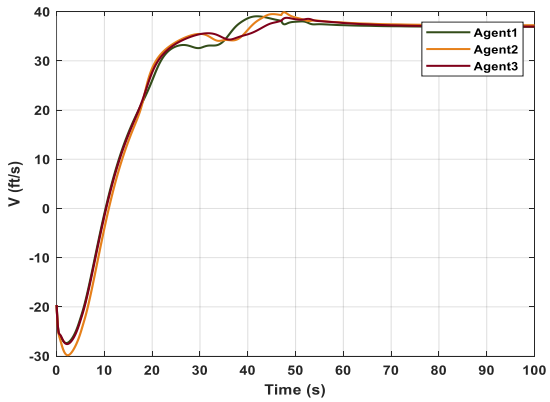
which were not present in the previous case with the Rough mimetic neural controller. The subsequent panel (b) of Figure (7) illustrates the pitch rate of the agents throughout the mission. The desired pitch rate range is approximately zero, and the application of the control signal causes each agent's pitch rate to converge to zero. However, the chattering phenomenon persists over time and in the steady state for this variable. The subsequent panel, C, provides a comprehensive representation of the altitude variations associated with each factor. The flight path angle of each agent is illustrated in Panel (d) of Figure (7). As previously mentioned, the Sliding Mode Controller successfully converges each agent's flight path angle to the desired value of zero, thereby ensuring that the agents are positioned along the optimal horizontal path. A notable distinction between the two figures is that the gamma angle values throughout the mission are higher in the former case compared to the latter case. The Sliding Mode Controller has been successful in achieving a coordinated landing of the agents with a reduced flight path angle. The longitudinal flight control system for actuator landing, as depicted in Figure (8), has been designed to facilitate the automatic landing of actuators. As illustrated in Figure (9), the performance of two distinct flight control systems, namely the Rough Mimetic neural flight control system and the Sliding Mode Control system, is examined. To facilitate a more robust comparison, a sample actuator is analyzed at an altitude of 2400 feet. As illustrated in panel (a) of Figure (9), the load factor applied to the selected actuator is depicted for both flight control systems. It is evident from the data presented in Panel (a) that the load factor when the actuator is controlled by the Rough mimetic neural controller is lower compared to the Sliding Mode controller. This results in less stress on the actuators. Panel (b) illustrates the thrust vector control operator's control effort for the selected actuator in both flight control systems. As illustrated in the figure, the control effort required by the Rough mimetic neural controller is higher compared to the Sliding Mode controller.



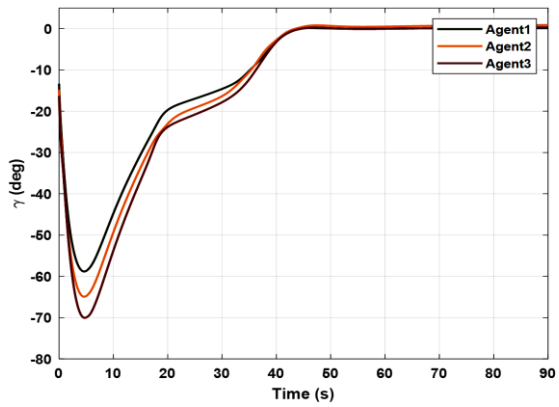
(**a**)



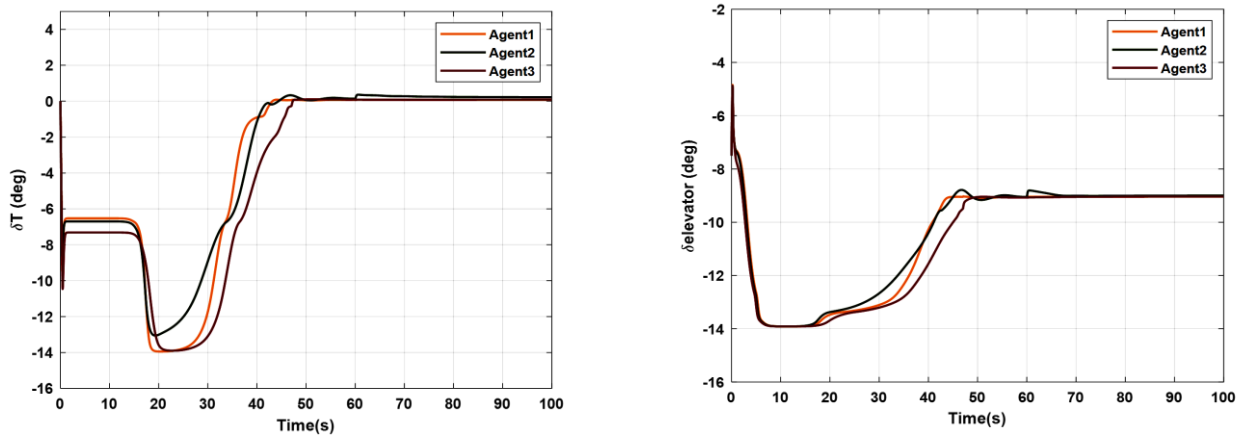
(**b**)



(**c**)

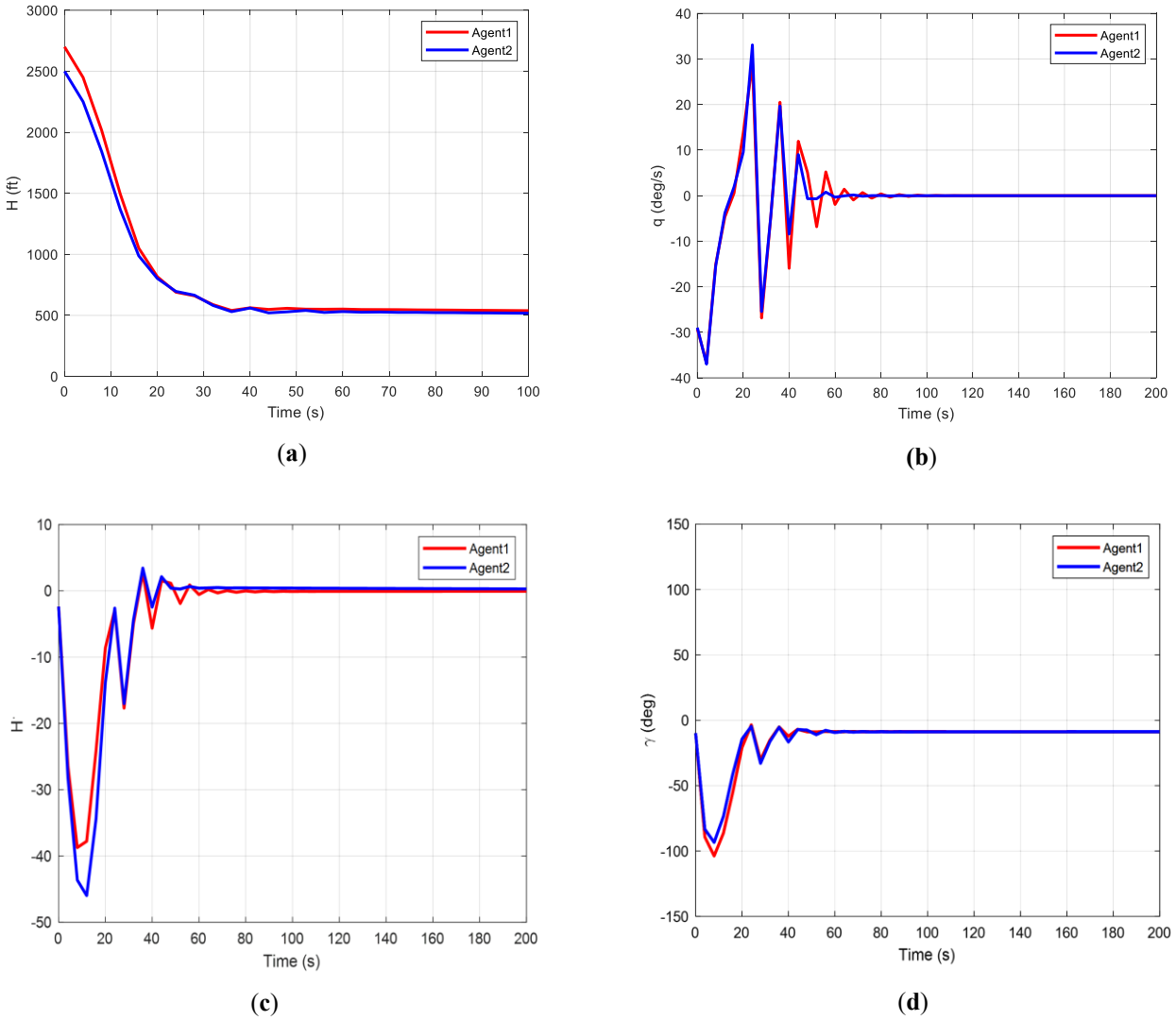


(**d**)



(**e**) (**f**)

Figure 6 State variables for each agent (Rough mimetic neural controller); (**a**) Altitude of agents(H), (**b**) Pitch Rate (q), (**c**) Speed agreement among agents (V), (**d**) Flight Path Angle (γ), (**e**) Elevator Angle ($\delta_{elevator}$), (**f**) Thrust Vector Angle (δ_T)



(**a**) (**b**)

(**c**) (**d**)

Figure 7 State variables for each agent (Sliding mode controller); (**a**) Altitude of agents (H), (**b**) Pitch Rate (q), (**c**) Altitude changes of agents ($\dot{h}$), (**d**) Flight path angle (γ)

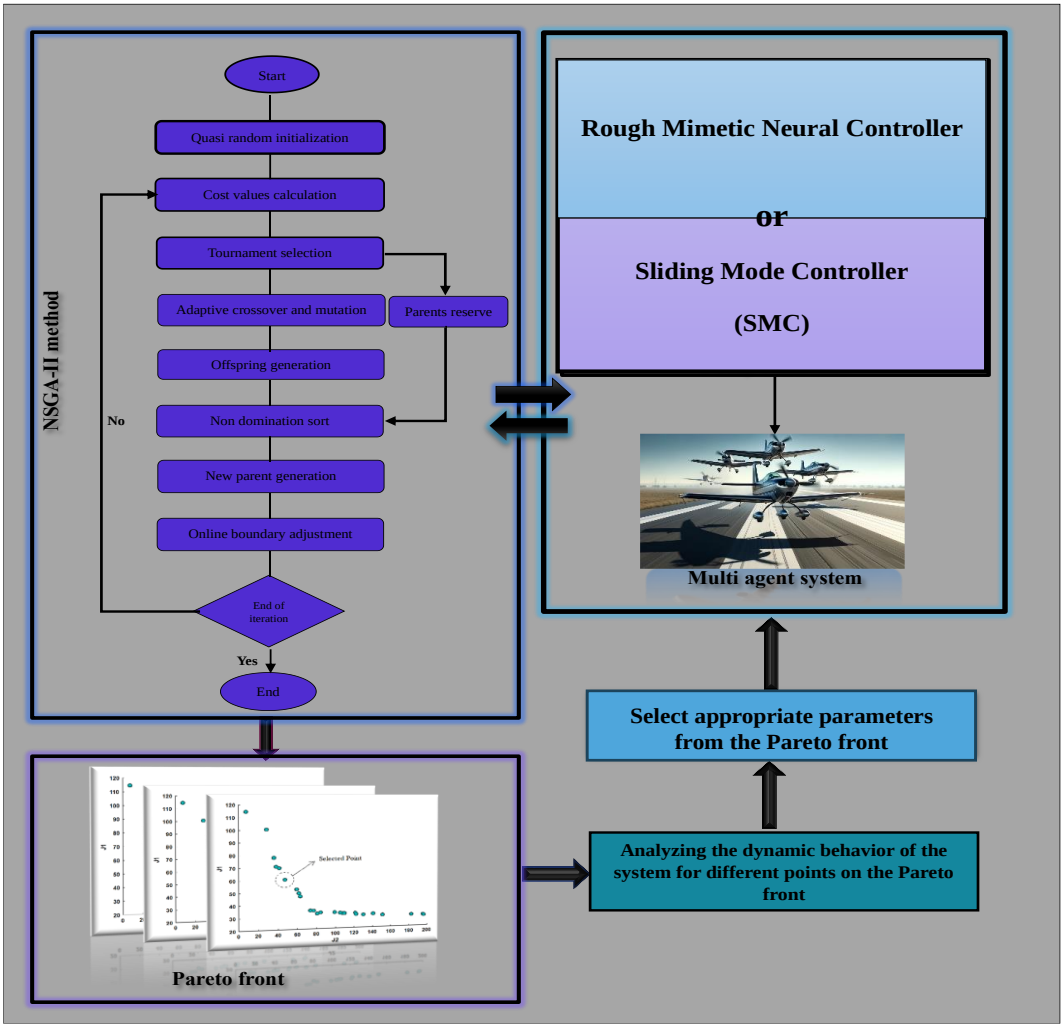
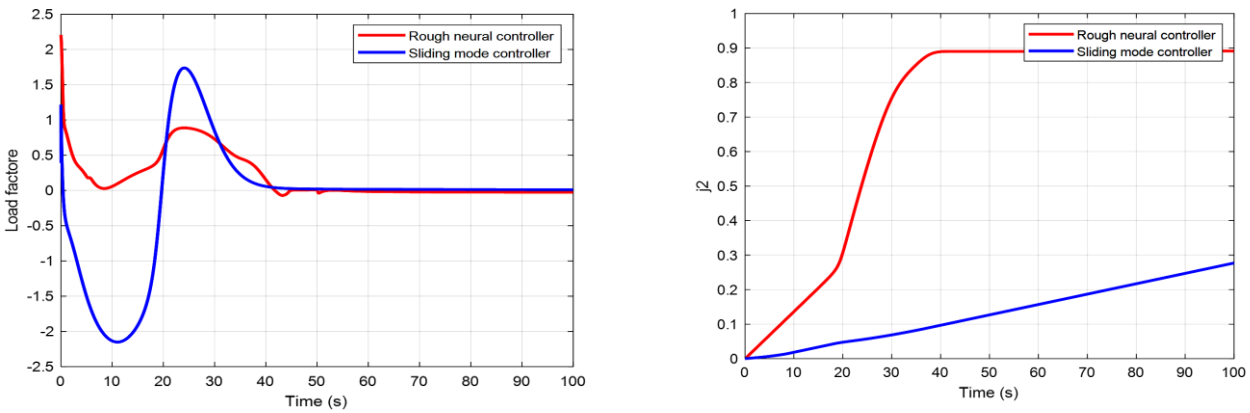


Figure 8 The overall schematic of the longitudinal flight control system



(**a**) (**b**)

Figure 9 (**a**) Load factore, (**b**) Control effort of thrust vector control operator (J_2)

5 Conclusion

This paper investigates and compares the coordinated landing of multiple vehicle using Rough Mimetic Neural Controller and Sliding Mode Controller (SMC). Coordinated landing scenarios, which are critical for advanced aerospace operations, demand robust and precise

control strategies to handle nonlinear dynamics and ensure safe and accurate landings. The simulations utilize nonlinear dynamic equations for each agent, incorporating cooperative actuators such as elevators and thrust vector control systems. The allocation of control signals among these actuators is determined by inputs such as the gamma angle, angle of attack, and altitude rate, with outputs considering changes in elevator angle and thrust vector control. The NSGA-II optimization algorithm is employed to tune the controller parameters, enhancing performance and reducing control effort. The simulation results demonstrate the effectiveness of both controllers in stabilizing the vehicle and achieving desired outputs with minimal control effort and sufficient accuracy. However, the Rough mimetic neural controller has been shown to exhibit superior adaptability and performance in dynamic environments, particularly in reducing response time and maintaining stability under varying conditions. The significance of this research lies in its potential to enhance the safety and efficiency of aerospace operations by addressing the challenges of coordinated landings in nonlinear and dynamic environments. The proposed methods provide a robust framework for achieving precise and reliable landings, paving the way for advancements in automated and cooperative aerospace systems. This study underscores the complementary advantages and trade-offs between the two methods, offering valuable insights for designing coordinated landing strategies in complex aerospace missions.

References

[1] Á. Soriano, E. J. Bernabeu, Á. Valera, and M. Vallés, "Multi-agent Systems for Evasive Maneuvers of Mobile Robots through Agreements," in *International Conference on Informatics in Control, Automation and Robotics*, 2013, Vol. 2: Scitepress, pp. 140-147, https://doi.org/10.5220/0004430101400147.

[2] J. Wang, X. Ding, C. Wang, L. Liang, and H. Hu, "Affine Formation Control for Multi-agent Systems with Prescribed Convergence Time," *Journal of the Franklin Institute*, Vol. 358, No. 14, pp. 7055-7072, 2021, https://doi.org/10.1016/j.jfranklin.2021.07.019.

[3] W. Du, T. Guo, J. Chen, B. Li, G. Zhu, and X. Cao, "Cooperative Pursuit of Unauthorized UAVs in Urban Airspace via Multi-agent Reinforcement Learning," *Transportation Research Part C: Emerging Technologies*, Vol. 128, pp. 103122, 2021, https://doi.org/10.1016/j.trc.2021.103122.

[4] R. Wang, "Adaptive Output-feedback Time-varying Formation Tracking Control for Multi-agent Systems with Switching Directed Networks," *Journal of the Franklin Institute*, Vol. 357, No. 1, pp. 551-568, 2020, https://doi.org/10.1016/j.jfranklin.2019.11.077.

[5] A. A. Elgohary, and B. Moidel, "A Novel Use of Model Predictive Control with Extremum Seeking in Formation Flight," in *AIAA SCITECH 2024 Forum*, 2024, pp. 2750, https://doi.org/10.2514/6.2024-2750.

[6] Y. Wang, N. Li, B. Wang, X. He, Y. Zhu, and M. Zhou, "Local Pursuit Strategy-inspired Cooperative Formation Flight and Collision Avoidance for UAV Cluster," in *ASME International Mechanical Engineering Congress and Exposition*, 2023, Vol. 87639: American Society of Mechanical Engineers, pp. V006T07A083, https://doi.org/10.1115/IMECE2023-113399.

[7] J. Martinez-Ponce, B. Herkenhoff, A. Aboelezz, and M. Hassanalian, "Load Distribution on “V” and Echelon Formation Flight of Flapping-wings," in *AIAA SCITECH 2024 Forum*, 2024, pp. 2337, https://doi.org/10.2514/6.2024-2337.

[8] A. Tabassum and H. Bai, "Dynamic Control Allocation between Onboard and Delayed Remote Control for Unmanned Aircraft System Detect-and-Avoid," *Aerospace Science and Technology*, Vol. 121, pp. 107323, 2022, https://doi.org/10.1016/j.ast.2021.107323.

[9] L. Cao, G.-P. Liu, and D.-W. Zhang, "Cloud-based Predictive Formation Control of Networked Multi-agent System and Its Application to Air Bearing Spacecraft Simulators," *ISA Transactions*, Vol. 138, pp. 696-704, 2023, https://doi.org/10.1016/j.isatra.2023.03.013.

[10] C. Kang, J. Xu, and Y. Bian, "Affine Formation Maneuver Control for Multi-agent Based on Optimal Flight System," *Applied Sciences*, Vol. 14, No. 6, p. 2292, 2024, https://doi.org/10.3390/app14062292.

[11] R.C. Nelson, "*Flight Stability and Automatic Control*", New York: WCB/McGraw Hill, Vol. 2, 1998.

[12] Q. Bian, B. Nener, and X. Wang, "An Improved NSGA-II Based Control Allocation Optimisation for Aircraft Longitudinal Automatic Landing System," *International Journal of Control*, Vol. 92, No. 4, pp. 705-716, 2019, https://doi.org/10.1080/00207179.2018.1473643.

[13] M. Teshnehlab, and K. Watanabe, "*Flexible Neural Networks. In Intelligent Control Based on Flexible Neural Networks*", Dordrecht: Springer Netherlands, pp. 61-84. 1999.

[14] D. V. Rao and T. H. Go, "Automatic Landing System Design using Sliding Mode Control," *Aerospace Science and Technology*, Vol. 32, No. 1, pp. 180-187, 2014, https://doi.org/10.1016/j.ast.2013.10.001.