

Application of Chaos Theory and Null Space-Based Behaviour Control for Swarm Robot Navigation

Le Thi Thuy NGA¹, Van Binh NGUYEN^{2*}

¹ Department of Cybernetics, University of Transport and Communications, Hanoi, Vietnam
lethuynga@utc.edu.vn

² International University, Vietnam National University, Ho Chi Minh City 70000, Vietnam
nvbinh@hcmiu.edu.vn (*Corresponding author)

Abstract: The practical applicability of controlling swarm robots so that they perform various tasks is gaining traction. The key aspects of this control problem include obstacle avoidance, inter-robot interaction, collision avoidance, and ensuring the orderly movement of robots towards a set goal. This paper proposes a hybrid approach combining Null Space-Based behaviour (NSB) control techniques with logistic map-based chaos theory (C-NSB) in order to enable multi-tasking for swarm robots. When swarm robots need to execute multiple tasks simultaneously, it's crucial to prioritize the respective tasks. Additionally, this article proposes using the Takagi-Sugeno fuzzy model for estimating the attraction/repulsion forces between the individual robots in a swarm in order to maintain the swarm's formation. A novel contribution of this paper lies in the introduction of a chaotic variable for facilitating a smoother and more agile robot navigation when a robot is in an area where there is a high risk of collision with an obstacle. The theoretical research results were validated with regard to their correctness and effectiveness through simulations carried out in the Matlab environment.

Keywords: Behaviour control, Chaos theory, Fuzzy logic, Null space-based behaviour, Swarm robot navigation.

1. Introduction

Swarm robotics draws its operational paradigm from the self-organized collective patterns observed in biological groups, such as bird flocks, fish schools, or ant colonies. By leveraging decentralized control architectures, high scalability, and minimal hardware complexity per agent, swarms offer high survivability solutions for demanding missions. These range from wilderness search-and-rescue and ecological monitoring to geographic mapping within highly volatile spaces (Sahin, 2005; Tan & Zheng, 2013). This section traces swarm navigation from the early bio-inspired models to the modern frameworks integrating chaos maps, fuzzy logic, and artificial intelligence.

Swarm engineering translates natural flocking dynamics into distributed rules. Reynolds (1987) established this by generating collective behavior through three local rules: cohesion, separation, and alignment without global control. Kelly & Keating (1996) physically validated these principles using a multi-layer control stack on a five-robot testbed, demonstrating decentralized obstacle circumvention, leader tracking, dynamic splitting, and regrouping.

Building upon these historical milestones, contemporary research focuses on steering efficiency on cluttered terrain. Hanada, Lee & Chong (2007) proposed an adaptive flocking framework based on local rules with two nearest neighbors, enabling a decentralized obstacle navigation and swarm formation recovery. To formalize swarm dynamics, Gazi & Passino

(2003) established Lyapunov stability proofs using attractive-repulsive potential fields. Furthermore, Wang & Fang (2010) extended these models to anisotropic configurations in non-uniform environments, while Xue et al. (2011) formulated a globally stable stochastic model for target tracking and area coverage.

Despite their mathematical rigor, purely potential-based models are often affected by local minima in unmapped dynamic environments. To overcome this, Nga & Thang (2017) utilized fuzzy logic with soft membership functions for repulsion and attraction behaviors, achieving a superior adaptability in comparison with rigid formulations. Similarly, Ni & Yang (2012) combined fuzzy inference with chaotic genetic algorithms for a resilient multi-robot foraging in blind zones, leveraging the flexible behavioral framework established by Wang (1997).

Local obstacle avoidance was categorized by Brooks (1986) into arbitration and command fusion, though both struggle with severe task conflicts. To resolve this, Antonelli, Arrichiello & Chiaverini (2008) and Marino et al. (2012) introduced the Null Space-Based behavior (NSB) control paradigm. By mathematically projecting lower-priority task vectors into the null space of higher-priority tasks, NSB establishes a strict hierarchy that prevents secondary objectives from compromising safety, proving highly effective in complex swarm operations.

On a separate scientific track, the late 20-th century was defined by three paradigm-shifting intellectual movements: relativity, quantum mechanics, and chaos theory. Chaos theory -alternatively classified as the non-linear dynamical systems theory - investigates aperiodic, unstable trajectories born within completely deterministic systems (Lorenz, 1963; Lorenz, 1993). The overarching hallmark of any chaotic system is its extreme sensitivity to initial conditions. Under this regime, even if a set of initial states x_0, x_1, x_2, x_3 are packed incredibly close together within the phase space, their long-term trajectories will diverge exponentially over time (Figure 1).

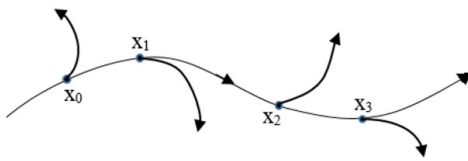


Figure 1. The trajectories which diverge over time, corresponding to different initial conditions

In practical applications, while Bjerknes et al. (1910) and Orrell (2002) focused on documenting chaos to explain the fundamental limits of long-range weather forecasting, and Slingo & Palmer (2011) explored its role in climate prediction uncertainty, roboticists soon weaponized this unpredictability. Nakamura & Sekiguchi (2001) were among the trailblazers to implement chaotic kinematics on a physical differential-drive robot. They proved that a mathematically chaotic trajectory covers an unknown bounded workspace significantly faster and more uniformly than classical pseudo-random walk-based heuristics.

Building on this, the recent studies are combining non-linear dynamics with metaheuristics. For instance, Singh et al. (2024) merged chaos theory with dragonfly optimization to maximize spatial search efficiency. Similarly, Mingle & Xiaoming (2016) used chaotic perturbations to prevent premature stagnation in ant colony systems, while Xusheng, Wenming & Jun (2017) combined genetic algorithms with chaotic local search to optimize robot trajectories with a lower computational time.

Bio-inspired heuristics has revolutionized multi-agent systems by replicating natural decentralized mechanics. Key paradigms include Ant Colony Optimization (Dorigo, Birattari & Stützle, 2006) for task allocation and path planning

(Athira, Divya Udayan & Subramaniam, 2024), and Particle Swarm Optimization (Kennedy & Eberhart, 1995), whose chaotic adaptive variants enable a smooth trajectory planning (Lian et al., 2020). Additionally, the Firefly Algorithm (Yang, 2010) optimizes dynamic swarm configurations, while ecological frameworks (Wang et al., 2022) allow an adaptive swarm formation recovery. Collectively, these paradigms significantly enhance operational scalability and adaptability (Sahin, 2005; Tan & Zheng, 2013).

Recently, integrating deep learning with swarm mechanics has enhanced operational performance. DRL-enabled agents execute complex missions in chaotic environments (Zhu, Zhang & Li, 2022), while embedding non-linear chaotic perturbations resolves the local-minima trap in classical artificial potential fields (APF) (Zhang, 2018; Fang & Xu, 2024). For an advanced non-linear control, fractional-order chaotic loops improve area coverage (Cui & Zheng, 2025), and chaos-driven neural networks aid in processing turbulent wind dynamics for drone stabilization (Barış, Yanarateş & Altan, 2024). These methods build upon foundational multi-agent path-tracking control (Miswanto et al., 2011).

Based on a synthesis of the inherent limitations of the previously presented methods, and with the aim of enhancing robot manoeuvrability, optimizing kinematic parameters, and ensuring feasibility when integrated robot resources are limited, this paper proposes a solution combining Null-space-based behavioural control for multi-tasking swarm robots. Simultaneously, this solution applies the Logistic map-based chaos model to guide robot movement in areas where there is a risk of collision between robots and obstacles. This allows individual robots within the swarm to avoid obstacles more smoothly, significantly increasing the practical applicability of robots.

The remainder of this paper is organized as follows. Section 2 describes the mathematical model of a swarm robot. Section 3 delves into the main contributions of combining chaos maps and null space in order to enhance the feasibility and efficiency of the robot's movement process. Further on, the simulation results obtained by applying the proposed method are presented and discussed in Section 4. Finally, Section 5 concludes this paper.

2. Mathematical Model of the Swarm Robot

When a swarm of robots needs to perform a complex task, the task can be divided into several basic tasks (behaviours), which are then combined appropriately to achieve the intended objective. This combination specifically involves prioritizing the basic tasks and projecting the lower-priority tasks into the null space of the higher-priority tasks.

Let $p^i = \begin{bmatrix} p_1^i \\ p_2^i \end{bmatrix} \in \mathbb{R}^{2 \times 1}$, $v^i = \begin{bmatrix} v_1^i \\ v_2^i \end{bmatrix} \in \mathbb{R}^{2 \times 1}$ be the position and velocity, respectively, of individual i in a 2D Euclidean space, then the mathematical model of individual i is described as follows:

$$\dot{p}^i = v^i \quad (1)$$

Let $u^i \in \mathbb{R}^{n_u \times 1}$ be the control input value for the individual i to achieve the task objective, then u^i will depend on p^i , which can be expressed as:

$$u^i = f(p^i) \quad (2)$$

The time derivative of equation (2) is:

$$\dot{u}^i = \frac{\partial f(p^i)}{\partial p^i} \dot{p}^i \quad (3)$$

Combining equations (1) and (3):

$$\dot{u}^i = J(p^i) v^i \quad (4)$$

where $J(p)$ is the Jacobian matrix, $J(p^i) \in \mathbb{R}^{n_u \times 2}$.

It follows that:

$$v^i = J^+ \dot{u}^i = J^+ (JJ^+)^{-1} \dot{u}^i \quad (5)$$

where J^+ is the pseudoinverse of $J(p^i)$, $J^+ \in \mathbb{R}^{2 \times n_u}$ is defined according to the four Moore-Penrose conditions (Antonelli, Arrichiello & Chiaverini, 2008):

$$\begin{aligned} JJ^+ J &= J; \\ J^+ JJ^+ &= J^+; \\ (JJ^+)^T &= JJ^+; \\ (J^+ J)^T &= J^+ J \end{aligned} \quad (6)$$

Let u_d^i be the desired value (the desired distance from the robot to the target), then equation (5) can be rewritten as follows:

$$v^i = \lambda J^+ (u^i - u_d^i) = k_v J^+ \tilde{d} \quad (7)$$

where k_v is a positive coefficient and $\tilde{d} = (u^i - u_d^i)$ is the deviation between the actual value and the desired value.

Since u_d^i is a constant, then:

$$\dot{\tilde{d}} = (\dot{u}^i - \dot{u}_d^i) = \dot{u}^i = Jv^i = \lambda \tilde{d} \quad (8)$$

The orthogonal projection matrix of J , denoted as $N_J \in \mathbb{R}^{2 \times 2}$, is defined as follows (in Figure 2):

$$N_J = I_n - J^+ J \quad (9)$$

where I_n is the identity matrix, $I_n \in \mathbb{R}^{2 \times 2}$.

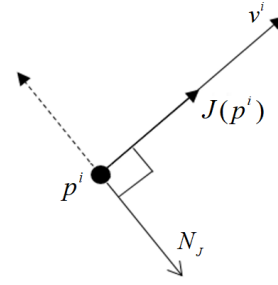


Figure 2. Diagram of the null space of N_J

N_J is referred to as the null space of the task to be completed. The matrix N_J is a symmetric matrix, and it can be easily shown that:

$$\begin{aligned} N_J (AN_J)^+ &= (AN_J)^+, \quad \forall A \in \mathbb{R}^{1 \times 2}; \\ N_J^+ &= N_J; \\ JN_J &= N_J J^+ = 0 \end{aligned} \quad (10)$$

When a robot swarm performs the task of moving to a destination, it must simultaneously execute various tasks to achieve the set objectives. Each individual robot in the swarm must perform three main tasks: 1) obstacle avoidance; 2) moving to the destination; and 3) maintaining the swarm to prevent collisions among individuals while avoiding group separation. Using the null space-based behaviour control technique, the robots need to prioritize these tasks. This study selects the priority order as follows: obstacle avoidance, moving to the destination, and finally, maintaining the swarm. Obstacle avoidance is assigned the highest priority to guarantee hardware safety, as a collision with an obstacle would destroy the robot and invalidate the subsequent tasks. Target tracking is of secondary priority when driving the robot swarm toward the goal, while swarm maintenance is projected into the null space of higher-priority tasks to preserve swarm formation topology without compromising robot safety.

or navigation. It is assumed that the obstacles are static and known in advance. In this case, the velocity vector of each individual robot is synthesized according to the diagram in Figure 3.

The mathematical velocity model of the individual robot when hierarchically assigning tasks is determined by:

$$v^i = v_o + N_o v_g + N_{og} v_s \quad (11)$$

where v_o, v_g, v_s are the velocity vectors corresponding to the tasks of obstacle avoidance, moving to the destination, and maintaining the swarm, respectively, while N_o, N_{og} are the null matrices calculated based on the priority order of the tasks. These values are specifically calculated based on the different tasks of the robot.

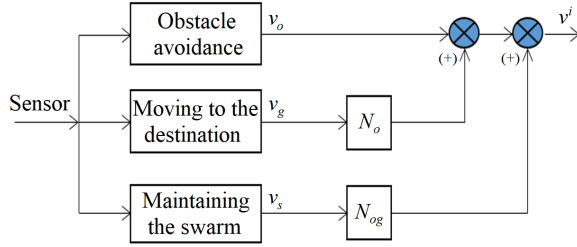


Figure 3. Block diagram for synthesizing the velocity vector of the i^{th} individual robot

3. Multi-task Control Using C-NSB

3.1 Controlling a Swarm Robot for Obstacle Avoidance

When using different chaos maps, the behaviour of the robot varies accordingly. Chaos map models demonstrate that the future state depends on the degree of chaos and the state at the previous time step. Denoting x_c, x_{c+1} as the values of the chaotic variable at the previous and subsequent time steps, respectively, and λ as the chaos parameter, the following chaos map models are commonly used:

- Sine map:

$$x_{c+1} = \frac{\lambda}{4} \sin(\pi x_c) \quad (12)$$

where $0 < \lambda \leq 4$.

- Singer map:

$$x_{c+1} = \lambda(7.86x_c - 23.31x_c^2 + 28.75x_c^3 - 13.3x_c^4) \quad (13)$$

where $0.9 < \lambda \leq 1.08$.

- Logistic map:

$$x_{c+1} = \lambda x_c (1 - x_c) \quad (14)$$

where $3.1 \leq \lambda \leq 3.99$.

- Kent map:

$$x_{c+1} = \lambda x_c (1 - x_c) \quad (15)$$

where $0 < \lambda < 4.5$.

During movement, individual robots must avoid the obstacles on their path.

Assumption: It is supposed that $p^{om} = \begin{bmatrix} p_1^{om} \\ p_2^{om} \end{bmatrix} \in \mathbb{R}^{2 \times 1}$

is the position of the m^{th} obstacle ($m = 1$ to M) in 2D Euclidean space. Then, the actual distance between the i^{th} individual robot and the m^{th} obstacle is determined by:

$$d_o = \|p^{om} - p^i\| = \sqrt{(p_1^{om} - p_1^i)^2 + (p_2^{om} - p_2^i)^2} \quad (16)$$

The goal is to control a robot in order to avoid obstacles: if an obstacle lies on the robot's path to the destination, the robot must maintain a safe distance d_o^* from the obstacle; if the obstacle is outside the robot's movement region, it does not affect the robot's movement velocity. To avoid obstacles, in a previous study of Nga & Thang (2017), the NSB (Null Space-based Behavior) control technique was employed. The results showed that robots successfully avoided obstacles, but the robot's turning angle changed abruptly, and the robot had to change direction multiple times, which posed challenges for the robot's actuation mechanisms. Therefore, this paper proposes a solution that combines chaos theory and the NSB technique to enable the robot swarm to navigate appropriately in environments with multiple obstacles, allowing for a smoother and more flexible obstacle avoidance. Specifically, the collision-risk operating mode is activated when the distance between the i^{th} robot and the m^{th} obstacle falls below the safe distance threshold $d_{o_{im}} < d_o^*$. Under this condition, the obstacle-avoidance navigation is executed by constructing the modified Jacobian matrix J_o incorporating chaotic perturbation as shown in equation (17). Conversely, if $d_{o_{im}} \geq d_o^*$, the obstacle does not affect the robot's motion. This paper utilizes the logistic chaos map model (equation 14) to move the robot in areas where there is a high-risk of collision with an obstacle, selecting $\lambda = 3.99$ as the initial value of the chaotic variable x_c : $0 \leq x_{c=0} \leq 1$. The logistic map is specifically selected due to

its computational simplicity combined with a rich nonlinear dynamics, deterministic pseudo-randomness, and an extreme sensitivity to initial conditions. Within the proposed C-NSB framework, these nonlinear properties generate dynamic directional perturbations that effectively break spatial symmetry and prevent local deadlocks, thereby improving robot navigation efficiency and enabling a smooth, highly adaptive collision mitigation. Using the chaotic variable x_c to represent the robot's turning angle in a chaotic environment, the Jacobian matrix $J_o \in \mathbb{R}^{M \times 2}$, which represents the robot's obstacle avoidance velocity, is determined as follows:

$$J_o = \hat{p}_{io}^T = \begin{bmatrix} \left[\frac{p^{\phi_1} - p^i}{\|p^{\phi_1} - p^i\|} \cos(\text{actan}(x_c)) \quad \frac{p^{\phi_1} - p^i}{\|p^{\phi_1} - p^i\|} \sin(\text{actan}(x_c)) \right]^T \\ \left[\frac{p^{\phi_2} - p^i}{\|p^{\phi_2} - p^i\|} \cos(\text{actan}(x_c)) \quad \frac{p^{\phi_2} - p^i}{\|p^{\phi_2} - p^i\|} \sin(\text{actan}(x_c)) \right]^T \\ \vdots \\ \left[\frac{p^{\phi_M} - p^i}{\|p^{\phi_M} - p^i\|} \cos(\text{actan}(x_c)) \quad \frac{p^{\phi_M} - p^i}{\|p^{\phi_M} - p^i\|} \sin(\text{actan}(x_c)) \right]^T \end{bmatrix} \quad (17)$$

The pseudo-inverse matrix of J_o is expressed as:

$$J_o^+ = \hat{p}_{io}, J_o^+ \in \mathbb{R}^{2 \times M} \quad (18)$$

The orthogonal projection matrix of J_o is expressed as:

$$N_o = I_n - \hat{p}_{io} \hat{p}_{io}^T, N_o \in \mathbb{R}^{2 \times 2} \quad (19)$$

where I_n is the identity matrix.

Based on equation (7), the obstacle avoidance velocity of the robot is determined as follows:

$$v_o = -k_{vo} J_o^+ (d_o - d_o^*) = -k_{vo} J_o^+ \tilde{d}_o \quad (20)$$

where k_{vo} is a negative coefficient, and $\tilde{d}_o = d_o - d_o^*$ is the deviation between the actual distance and the safe distance from the robot to the obstacle.

3.2 Controlling a Swarm Robot for Moving Toward a Target

Let $p^g = \begin{bmatrix} p_1^g \\ p_2^g \end{bmatrix} \in \mathbb{R}^{2 \times 1}$ be the position of the target in 2D Euclidean space. The actual distance between the i^{th} robot and the target, d_g is calculated as such:

$$d_g = \|p^g - p^i\| = \sqrt{(p_1^g - p_1^i)^2 + (p_2^g - p_2^i)^2} \quad (21)$$

The goal of controlling the robot toward the destination is for the robot to reach the target, $d_g^* = 0$.

The Jacobian matrix $J_g \in \mathbb{R}^{1 \times 2}$ is calculated as follows:

$$J_g = \left[\frac{p^g - p^i}{\|p^g - p^i\|} \right]^T = \hat{p}_{ig}^T \quad (22)$$

The pseudo-inverse matrix of J_g is expressed as:

$$J_g^+ = \hat{p}_{ig}, J_g^+ \in \mathbb{R}^{2 \times 1} \quad (23)$$

The orthogonal projection matrix of J_g is expressed as:

$$N_g = I_n - \hat{p}_{ig} \hat{p}_{ig}^T, N_g \in \mathbb{R}^{2 \times 2} \quad (24)$$

Based on equation (7), the velocity of the i^{th} robot moving toward the target is rewritten as follows:

$$v_g = k_{vg} J_g^+ (d_g - d_g^*) = k_{vg} J_g^+ \tilde{d}_g \quad (25)$$

where k_{vg} is a positive coefficient, and $\tilde{d}_g$ is the deviation between the actual distance and the desired distance from the robot to the target, which can be expressed as:

$$\tilde{d}_g = d_g - d_g^* = d_g \quad (26)$$

3.3 Controlling a Robot Swarm in Order to Maintain Swarm Formation

Among the tasks of a robot swarm, maintaining the swarm is one of the most critical ones. In (Nga & Thang, 2017), the authors analysed the convergence behaviour of the swarm based on attraction/repulsion forces using fuzzy logic. This attraction/repulsion force is developed based on the Mamdani fuzzy model. In many cases, when the number of input variables increases or when integration with machine learning techniques is required, the Mamdani model struggles to meet the demands. Therefore, this article proposes using the Takagi-Sugeno fuzzy model to estimate the attraction/repulsion forces between robots in the swarm to ensure the swarm's formation is maintained. Unlike the Mamdani system, the Takagi-Sugeno inference mechanism generates crisp functional outputs without requiring a computationally heavy defuzzification. This enables a real-time, smooth adjustment of the dynamic attraction-repulsion boundaries, allowing the swarm to rapidly adapt to swift spatial configuration changes while maintaining formation stability.

The actual distance between the i^{th} and j^{th} robots ($1, \dots, N, j \neq i$) is d_s . The goal of the control is to maintain the distance between two individual

robots at a constant $d_s = d_s^* \in \mathbb{R}$. The deviation between the actual distance and the desired distance is $\tilde{d}_s = d_s - d_s^*$.

If f denotes the interaction force between the pair of robots (i, j) , then:

$$f = f(\|p^j - p^i\|) \quad (27)$$

where $\|p^j - p^i\|$ is the distance between the two individuals i and j , and d_s is the actual distance between the robots in the pair (i, j) , then:

$$d_s = \|p^j - p^i\| = \sqrt{(p_1^j - p_1^i)^2 + (p_2^j - p_2^i)^2} \quad (28)$$

Let

$$g(\|p^j - p^i\|) = \frac{f(\|p^j - p^i\|)}{\|p^j - p^i\|} = \frac{f(d_s)}{d_s} = g_a(d_s) - g_r(d_s) \quad (29)$$

where $g(\cdot)$ denotes the attraction/repulsion function related to the pair of individuals (i, j) , $g_a(\cdot)$ is the repulsion force, and $g_r(\cdot)$ is the attraction force.

Let $d_s^* \in \mathbb{R}$ be the distance between the two individuals i and j at which the attraction and repulsion forces between them are balanced:

$$f(d_s) \begin{cases} = 0 & \text{if } d_s = d_s^* \\ < 0 & \text{if } 0 < d_s < d_s^* \\ > 0 & \text{if } d_s^* < d_s \end{cases} \quad (30)$$

The interaction force between the individuals in the swarm $f(d_s)$ is a nonlinear function dependent on the distance between the pairs of individuals (i, j) . Therefore, this function can be constructed by using the Tasaki-Sugeno fuzzy logic system with a SISO structure as follows:

Step 1.

The input signal is $u = \tilde{d}_s$. It is assumed that u has a domain of values in the interval $[\alpha_b, \beta_b] \in \mathbb{R}$, and u is divided into $2K+1$ intervals B^k as shown in Figure 4.

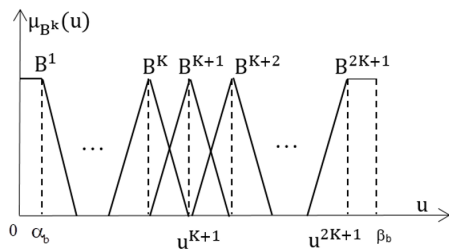


Figure 4. Fuzzification of the input signal of the fuzzy controller

The output signal is $A = f(u)$, with a domain of values in the interval $[\alpha_a, \beta_a]$, divided into $2K+1$ intervals A^k with $k=1, 2, \dots, 2K+1$ as shown in Figure 5. The centroid a^k of the fuzzy interval A^k is:

$$a^k \begin{cases} < 0 & \text{if } k = 1, 2, \dots, K \\ = 0 & \text{if } k = K + 1 \\ > 0 & \text{if } k = K + 2, \dots, 2K + 1 \end{cases} \quad (31)$$

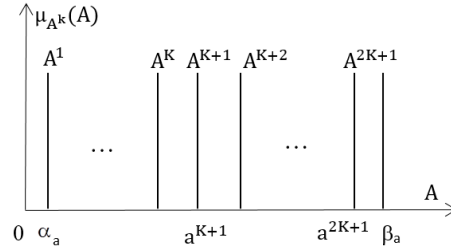


Figure 5. Fuzzification of the output signal of the fuzzy controller

Step 2.

Establishing $2N+1$ IF...THEN... rules of the form: IF $u = B^k$ THEN $a = A^k$ with A^k being a constant.

Step 3.

Selecting the composition rule and defuzzifying by using the weighted average method. According to Wang (1997), the control law is obtained as:

$$f(u) = \frac{\sum_{k=1}^{2K+1} a^k \mu_{B^k}(u)}{\sum_{k=1}^{2K+1} \mu_{B^k}(u)} \quad (32)$$

With the fuzzy controller designed through the three steps above, the relationship between the input signal (distance) and the output signal (interaction force between the individuals in the pair (i, j)) is obtained with the following properties:

$$f(u) \begin{cases} > 0 & \text{if } d_s > d_s^* \\ < 0 & \text{if } 0 < d_s < d_s^* \\ = 0 & \text{if } d_s = d_s^* \end{cases} \quad (33)$$

The fuzzy function $f(u)$ is a continuous function satisfying the following conditions:

- Upper and lower bounds: $-A_{\min} \leq f(u) \leq A_{\max}$, where $A_{\min} = -a^1$, $A_{\max} = a^{2N+1}$.
- Piecewise linearization equation:

$$f(\tilde{d}_s) = f(u) = \frac{(a^{k+1} - a^k)u + a^k u^{k+1} - a^{k+1} u^k}{u^{k+1} - u^k} \quad (34)$$

where $u \in [u^k, u^{k+1}]$, with $k \in \{1, 2, \dots, 2K\}$.

The Jacobian matrix representing the swarm maintenance process is expressed as follows:

$$J_s = \hat{p}_s^T = \begin{bmatrix} J_{s1} \\ J_{s2} \\ \vdots \\ J_{sN} \end{bmatrix} = \begin{bmatrix} \hat{p}_{s1}^T \\ \hat{p}_{s2}^T \\ \vdots \\ \hat{p}_{sN}^T \end{bmatrix} = \begin{bmatrix} \left(\frac{p^1 - p^i}{\|p^1 - p^i\|} \right)^T \\ \left(\frac{p^2 - p^i}{\|p^2 - p^i\|} \right)^T \\ \vdots \\ \left(\frac{p^N - p^i}{\|p^N - p^i\|} \right)^T \end{bmatrix} \in \mathbb{R}^{N \times 2} \quad (35)$$

The pseudo-inverse matrix of J_s is expressed as:

$$J_s^+ = \hat{p}_s = \begin{bmatrix} J_{s1} \\ J_{s2} \\ \vdots \\ J_{sN} \end{bmatrix}^T = \begin{bmatrix} \hat{p}_{s1}^T \\ \hat{p}_{s2}^T \\ \vdots \\ \hat{p}_{sN}^T \end{bmatrix}^T = \begin{bmatrix} \left(\frac{p^1 - p^i}{\|p^1 - p^i\|} \right)^T \\ \left(\frac{p^2 - p^i}{\|p^2 - p^i\|} \right)^T \\ \vdots \\ \left(\frac{p^N - p^i}{\|p^N - p^i\|} \right)^T \end{bmatrix}^T \in \mathbb{R}^{2 \times N} \quad (36)$$

The orthogonal projection matrix of J_s is expressed as:

$$N_s = I_N - \hat{p}_s \hat{p}_s^T, N_s \in \mathbb{R}^{2 \times 2} \quad (37)$$

Based on (7), the velocity of the i^{th} individual robot performing the swarm maintenance task is determined as follows:

$$v_s = J_s^+ f(\tilde{d}_s) \quad (38)$$

Specifically, the Takagi–Sugeno fuzzy controller evaluates the inter-robot attraction/repulsion forces to determine the desired swarm maintenance velocity v_s . Within the NSB architecture, v_s is then projected onto the null space of the higher-priority tasks (obstacle avoidance and navigation towards the target), thereby preserving swarm formation cohesion without interfering with the higher-priority control objectives.

3.4 The Movement Velocity of the Swarm Robot

By combining equations (11), (20), (25) and (38), the synthesis of the velocity vectors for each individual robot in the swarm when performing the three tasks using the C-NSB method is as follows:

$$\begin{aligned} v^i &= v_o + N_o v_g + N_{og} v_s \\ &= -k_{vo} J_o^+ \tilde{d}_o + k_{vg} N_o J_g^+ \tilde{d}_g + N_{og} J_s^+ f(\tilde{d}_s) \end{aligned} \quad (39)$$

where

$$v^i \in \mathbb{R}^{2 \times 1}$$

$$J_{og} = \begin{bmatrix} J_o \\ J_g \end{bmatrix}, J_{og} \in \mathbb{R}^{(M+1) \times 2} \quad (40)$$

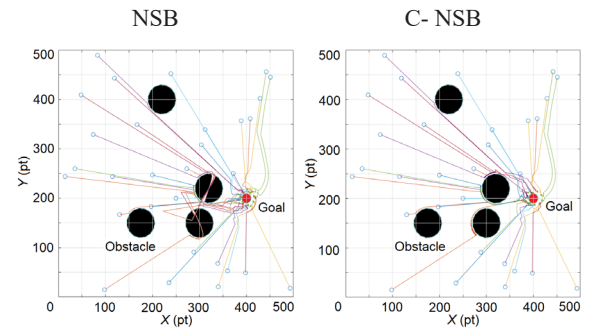
$$N_{og} = I_n - J_{og}^+ J_{og}, N_{og} \in \mathbb{R}^{2 \times 2}$$

Based on these integrated techniques, it can be observed that the proposed C-NSB methodology achieves a seamless multi-task coordination by synthesizing the Takagi-Sugeno fuzzy model, chaotic perturbation within the obstacle avoidance Jacobian matrix J_o , and null-space projection matrices (N_o, N_{og}). This hierarchical task decomposition mathematically guarantees that chaotic dynamic steering operates strictly within the safety-critical layer without introducing control conflicts among the secondary behavioral objectives.

4. Simulation Results

The movement process for the robots was simulated in a 2D space with dimensions of 500x500 steps. The simulations were conducted with varying numbers of robots in the swarm, different initial positions of the robots and obstacles, and different k_{vo} and k_{vg} coefficients.

Figure 6 illustrates the simulation results for the coefficients $k_{vo} = -1.5$ and $k_{vg} = 0.0005$. With the NSB method, some robots failed to avoid obstacles, whereas with the C-NSB method (Figure 6, right side), all robots successfully avoided the obstacles and converged to the target.



Parameters: $N = 30$; $M = 4$; $k_{vo} = -1.5$; $k_{vg} = 0.0005$

Figure 6. Comparison of the movement process for the robot swarm when using NSB and C-NSB

Figure 7 illustrates the movement process for the robots from their initial positions to the target position, when the C-NSB method was employed, the robots were able to avoid obstacles more smoothly in comparison with the case when the

NSB method was used (Figure 7(a)). As regards to the robot trajectory, the movement using C-NSB is less jagged (right side of Figure 7(b)). The robot moves more optimally in terms of distance, resulting in overall energy consumption savings.

As the number of robots increases, the positive impact of the C-NSB algorithm becomes more evident. The paths of the robots are smoother, especially when approaching obstacles and nearing the destination (Figure 8(a)). Also, the distance from the robot to the destination over time (Figure 8(b)) is more uniform with the C-NSB method. This indicates that the robot moves in a more orderly manner and that the swarm operates more harmoniously as a group.

Empirical evidence gathered from the numerical simulations confirms that the proposed C-NSB framework systematically outclasses the traditional NSB formulation across every assessed operational index, a trend consistently observed for both the 16-agent and 30-agent swarm configurations (Table 1). Quantitative tracking indicates that the C-NSB control loop fundamentally optimises the spatial navigation profiles. This structural refinement yields a substantial 14.2% to 15.3% contraction in the cumulative travel distances, alongside a

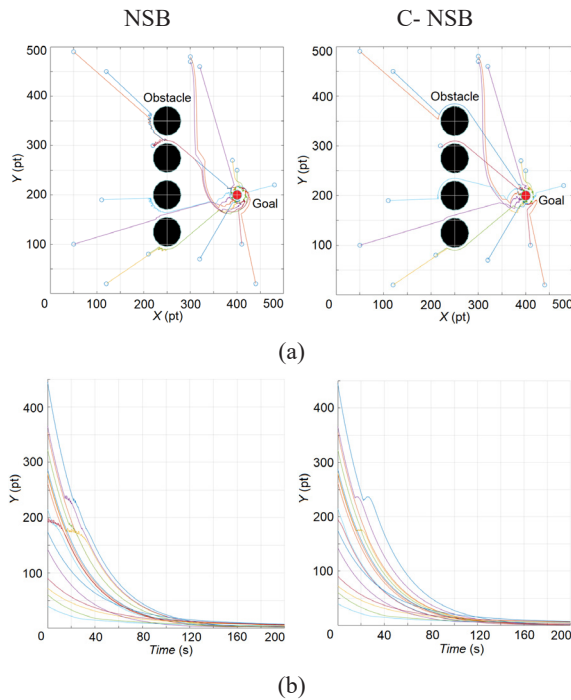
corresponding 16.1% to 17.5% reduction in the total convergence time.

Table 1. Performance metrics values for the proposed C-NSB method in comparison with the NSB method

Comparison parameters	Improvement (%)	
Number of robots (N)	16	30
Number of obstacles (M)	4	4
Average distance (%)	15.3	14.2
Average convergence time (%)	17.5	16.1
Movement smoothness	9.7	11.3

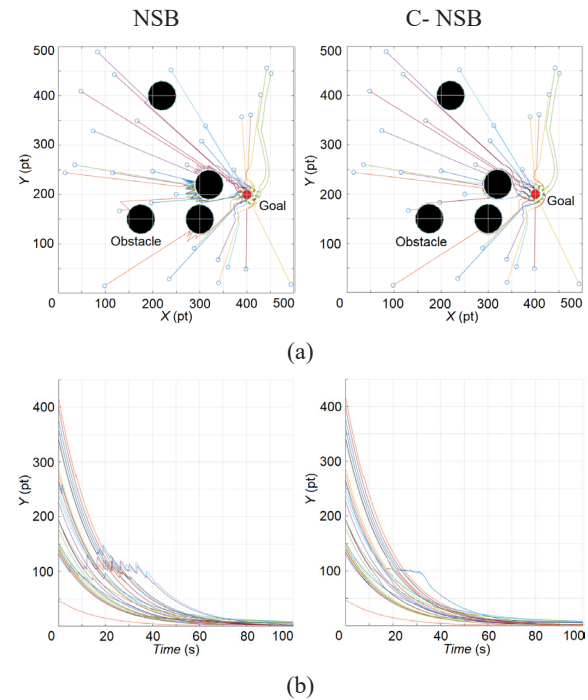
Crucially, the kinematic smoothness index exhibits an increase of 9.7% to 11.3%. This particular improvement is vital because it demonstrates that the chaotic injection effectively dampens the severe trajectory oscillations that typically affect the standard potential field or behaviour-switching methods during sharp obstacle-circumvention manoeuvres.

Underlying these empirical results, controller gains directly govern the convergence speed toward the desired target, where higher gains reduce the total travel time but may introduce excessive jerk and degrade trajectory smoothness. Meanwhile, properly tuned chaos parameters generate bounded pseudo-random perturbations that effectively break local deadlocks without sacrificing navigation



Parameters: $N = 16$; $M = 4$; $k_{vo} = -10$; $k_{vg} = 0,0005$

Figure 7. Performance comparison between the NSB and C-NSB methods for a 16-robot swarm: (a) movement trajectories, and (b) distance to the target over time.



Parameters: $N = 30$; $M = 4$; $k_{vo} = -15.5$; $k_{vg} = 0,0005$

Figure 8. Performance comparison between the NSB and C-NSB methods for a 30-robot swarm: (a) movement trajectories, and (b) distance to the target over time.

stability. Together, an optimal selection of these parameters achieves a balanced trade-off between rapid convergence, motion smoothness, and the overall swarm stability.

Building upon this parametric stability, the robustness of the system becomes particularly evident under stress testing. For the scenario involving 30 robots - a threshold that drastically amplifies localized physical congestion and introduces highly non-linear inter-agent interaction vectors - the C-NSB controller preserves its operational edge. In this dense environment, the framework maintains efficiency gains exceeding 11% for path smoothness and over 14% for both travel distances and convergence time. These metrics rigorously validate the high scalability, mathematical resilience, and superior multi-objective coordination enabled by the chaos theory-driven null-space-based architecture.

5. Conclusion

This paper proposed a solution that combines the Takagi-Sugeno fuzzy model with the null

space-based behaviour control technique and a logistic chaos map for navigating a robot swarm performing tasks in an environment with multiple obstacles. The results demonstrate the effectiveness of the proposed method in comparison with previously employed methods. After a period of movement, the robots successfully reached the target and avoided obstacles smoothly, as their actuation mechanisms could operate more flexibly. This enables the robot swarm to be more widely applied in practice, given that the variability, uncertainty, and randomness of the obstacles, the surrounding objects, and operational targets in real-life environments vary greatly. Moreover, the results provide further impetus for applying new technologies to the field of robotics, which has garnered significant attention in recent years.

Acknowledgements

This research is funded by the University of Transport and Communications (UTC) under grant number T2025-DT-003TĐ.

REFERENCES

- Antonelli, G., Arrichiello, F. & Chiaverini, S. (2008) The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*. 1(1), 27-39. <https://doi.org/10.1007/s11370-007-0002-3>.
- Athira, K. A., Divya Udayan, J. & Subramaniam, U. (2024) A systematic literature review on multi-robot task allocation. *ACM Computing Surveys*. 57(3), Art. No. 68. <https://doi.org/10.1145/3700591>.
- Bariş, C., Yanarates, C. & Altan, A. (2024) A robust chaos-inspired artificial intelligence model for dealing with nonlinear dynamics in wind speed forecasting. *PeerJ Computer Science*, 10, Art. No. e2393. <https://doi.org/10.7717/peerj-cs.2393>.
- Bjerknes, V., Sandström, J. W. & Devik, O. M. (1910) *Dynamic meteorology and hydrography*. Washington, D.C. Carnegie Institution of Washington.
- Brooks, R. (1986) A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*. 2(1), 14-23. <https://doi.org/10.1109/JRA.1986.1087032>.
- Cui, Y. & Zheng, Z. (2025) Novel fractional-order chaotic system applied to mobile robot path planning and chaotic path synchronization. *Symmetry*. 17(3), Art. No. 350. <https://doi.org/10.3390/sym17030350>.
- Dorigo, M., Birattari, M. & Stützle, T. (2006) Ant colony optimization. *IEEE Computational Intelligence Magazine*. 1(4), 28-39. <https://doi.org/10.1109/CI-M.2006.248054>.
- Fang, J. & Xu, S. (2024) An improved hybrid path planning algorithm in indoor environment. *International Journal of Advanced Network, Monitoring and Controls*. 9(2), 67-81. <https://doi.org/10.2478/ijanmc-2024-0017>.
- Gazi, V. & Passino, K.M. (2003) Stability analysis of swarms. *IEEE Transactions on Automatic Control*. 48(4), 692-697. <https://doi.org/10.1109/TAC.2003.809765>.
- Hanada, Y., Lee, G. & Chong, N.Y. (2007) Adaptive flocking of a swarm of robots based on local interactions. In: *2007 IEEE Swarm Intelligence Symposium, 1-5 April 2007, Honolulu, HI, USA*. New York, USA, IEEE. pp. 340-347. <https://doi.org/10.1109/SIS.2007.367957>.
- Kelly, I.D. & Keating, D.A. (1996) Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots. In: *Proceedings of the 3rd International Conference on Mechatronics and Machine Vision in Practice, Guimaraes, Portugal*. pp. 1-4.

- Kennedy, J. & Eberhart, R. (1995) Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks. 27 November-1 December 1995, Perth, WA, Australia*. New York, USA, IEEE. pp. 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>.
- Lian, J., Yu, W., Xiao, K. et al. (2020) Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm. *Mathematical Problems in Engineering*. 2020(3), Art. No. 1849240. <https://doi.org/10.1155/2020/1849240>.
- Lorenz, E.N. (1963) Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*. 20(2), 130-141. [https://doi.org/10.1175/1520-0469\(1963\)020<0130:DNF>2.0.CO;2](https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2).
- Lorenz, E.N. (1993) The essence of chaos. Seattle, USA, University of Washington Press.
- Marino, A., Parker, L., Antonelli, G. et al. (2012) A decentralized architecture for multi-robot systems based on the null-space behavioral control with application to multi-robot border patrolling. *Journal of Intelligent & Robotic Systems*. 71(3-4), 423-444. <https://doi.org/10.1007/s10846-012-9783-5>.
- Mingle, X. & Xiaoming, Y. (2016) Chaotic ANT system optimization for path planning of the mobile robots. *Computer Science & Engineering: An International Journal*. 6(2-3), 1-8. <https://doi.org/10.5121/cseij.2016.6301>.
- Miswanto, M. U., Pranoto, I., Muhammad, H. et al. (2011) The collective behaviour of multi-agents system for tracking a desired path. *International Journal of Basic & Applied Sciences*. 11(6), 52-54.
- Nakamura, Y. & Sekiguchi, A. (2001) The chaotic mobile robot. *IEEE Transactions on Robotics and Automation*. 17(6), 898-904. <https://doi.org/10.1109/70.976022>.
- Nga, L.T.T. & Thang, N.T. (2017) The multitasking system of swarm robot based on null-space-behavioral control combined with fuzzy logic. *Micromachines*. 8(12), Art. No. 357. <https://doi.org/10.3390/mi8120357>.
- Ni, J. & Yang, S.X. (2012) A fuzzy-logic based chaos GA for cooperative foraging of multi-robots in unknown environments. *International Journal of Robotics and Automation*. 27(1), 15-30. <https://doi.org/10.2316/Journal.206.2012.1.206-3553>.
- Orrell, D. (2002) Role of the metric in forecast error growth: How chaotic is the weather? *Tellus A: Dynamic Meteorology and Oceanography*. 54(4), 350-362. <https://doi.org/10.3402/tellusa.v54i4.12159>.
- Reynolds, C.W. (1987) Flocks, herds, and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*. 21(4), 25-34. <https://doi.org/10.1145/37402.37406>.
- Sahin, E. (2005) Swarm robotics: From sources of inspiration to domains of application. In: *Sahin, E. & Spears, W.M. (eds.) Swarm Robotics. SR 2004*. (part of the series *Lecture Notes in Computer Science*, vol 3342). Berlin, Heidelberg, Springer, pp. 10-20. https://doi.org/10.1007/978-3-540-30552-1_2.
- Singh, A.P., Kumar, G., Dhillon, G.S. et al. (2024) Hybridization of chaos theory and dragonfly algorithm to maximize spatial area coverage of swarm robots. *Evolutionary Intelligence*. 17(3), 1327-1340. <https://doi.org/10.1007/s12065-023-00823-5>.
- Slingo, J. & Palmer, T. (2011) Uncertainty in weather and climate prediction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 369(1956), 4751-4767. <https://doi.org/10.1098/rsta.2011.0161>.
- Tan, Y. & Zheng, Z.-Y. (2013) Research advance in swarm robotics. *Defence Technology*. 9(1), 18-39. <https://doi.org/10.1016/j.dt.2013.03.001>.
- Wang, C., Wang, D., Gu, M. et al. (2022) Bioinspired environment exploration algorithm in swarm based on Lévy flight and improved artificial potential field. *Drones*. 6(5), Art. No. 122. <https://doi.org/10.3390/drones6050122>.
- Wang, L. & Fang, H. (2010) Stability analysis of practical anisotropic swarms. In: *11th International Conference on Control Automation Robotics & Vision, 07-10 December 2010, Singapore*. New York, USA, IEEE. pp. 768-772. <https://doi.org/10.1109/ICARCV.2010.5707423>.
- Wang, L.X. (1997) *A course in fuzzy systems and control*. Upper Saddle River, NJ, USA, Prentice-Hall International, Inc.
- Xue, Z., Zeng, J., Feng, C. et al. (2011) Swarm target tracking collective behavior control with formation coverage search agents & globally asymptotically stable analysis of stochastic swarm. *Journal of Computers*. 6(8), 1772-1780.
- Xusheng, G., Wenming, G. & Jun, H. (2017) Robot path planning based on genetic and chaotic optimization algorithm. In: *5th International Conference on Computer, Automation and Power Electronics, 13-15 October, Dalian, China*. pp. 114-119. <https://doi.org/10.25236/cape.2017.023>.
- Yang, X.-S. (2010) Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-inspired Computation*. 2(2), 78-84. <https://doi.org/10.1504/IJBIC.2010.032124>.
- Zhang, C. (2018) Path planning for robot based on chaotic artificial potential field method. *IOP Conference Series: Materials Science and Engineering (4th International Conference on Advanced Engineering and Technology (4th ICAET) 15-17 December 2017, Shanghai, China)*. Vol. 317. Bristol, UK, IOP Publishing Ltd.. <https://doi.org/10.1088/1757-899X/317/1/012056>.
- Zhu, X., Zhang, F. & Li, H. (2022) Swarm deep reinforcement learning for robotic manipulation. *Procedia Computer Science*. 198, 472-479. <https://doi.org/10.1016/j.procs.2021.12.272>.



This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.