Transport and Communications Science Journal, Vol. 74, Issue 4 (05/2023), 519-529

Transport and Communications Science Journal

MULTI-TASK CONTROL OF SWARM ROBOT WITH DOUBLE INTEGRAL MODEL

Le Thi Thuy Nga[1*], Nguyen Thanh Hai[1], Nguyen Van Tung[2], Pham Xuan Tich[1]

[1]University of Transport and Communications, No 3 Cau Giay Street, Hanoi, Vietnam

[2]Hanoi University of Industry, No 298 Cau Dien Street, Hanoi, Vietnam

ARTICLE INFO

TYPE: Research Article
Received: 22/02/2023
Revised: 31/03/2023
Accepted: 12/04/2023
Published online: 15/05/2023
https://doi.org/10.1080/14680629.2017.1389072 (edited by editors)

[*] *Corresponding author*
Email: lethuynga@utc.edu.vn; Tel: +84987065998

Abstract. Biological individuals in the wild often have a definite size and mass, so simulation of the real biological swarm behavior should take these factors into account. The article focuses on building an algorithm to calculate the force acting on each individual robot in the swarm based on their mass through the "Double integral" model. When the robots are required to perform multi-tasks simultaneously, the priority order of tasks must be classified, and the task with lower priority will be projected into the Null space of the higher priority tasks. Each individual robot in the swarm has to fulfill following three tasks: avoid obstacles, move to the goal, maintain the swarm. In this study, the author chooses the priority level in the following order: avoiding obstacles, moving to the goal and finally maintaining the swarm. With an assumption that the obstacles are fixed and known in advance. Finally, the theoretical studies are simulated and verified by Matlab software.

Keywords: swarm robot, double integral model, multitask control, null space-based behavior.

1. INTRODUCTION

"Swarm robotics" is the use of a large group of relatively simple robots to perform tasks that a single robot cannot through the collaboration between individuals that mimics the behavior of species living in groups. The number of studies on swarm robots is quite a lot, and the number of studies on the stability of swarm robots is not less with certain successes. Swarm behavior is established based on the forces of interaction among individuals in the swarm and in the environment. This interaction force built by the authors [1-4] is usually explicit and precise mathematical functions without a convincing rationale. In addition, the environment of the swarm robots is relatively complicated and often changes, so it is difficult to satisfy an explicit mathematical model. Therefore, for natural logic, in [5-7], we propose a new swarm model, in which robot individuals are considered particles, and the interaction force between them is determined based on fuzzy logic. Fuzzy logic is flexible through a selection of input/output signals, blurring, fuzzy rules, etc., and is used to approximate attraction/repulsion in order to provide a more general reflection of explicit attraction/repulsion. The obtained result is that robots have converged at a region with a definite radius around the center of the swarms after a period of movement. In the study [8], the author used the NSB technique in combination with fuzzy logic to protect the swarm robots from obstacles and guide them to goals. However, it is just an assumption that the individual robots were considered the particles without taking into account the influence of the individual robot's mass.

For further development of the above studies, the article has built a dynamic model of the swarm taking into account the mass of each individual, and an algorithm to calculate the forces acting on robots when they perform multiple tasks simultaneously. Theoretical studies are simulated and tested for their correctness by Matlab software.

2. MATHEMATICAL MODEL OF THE SWARM ROBOTS

In [5-8], the authors have conducted studies on the movement of individual robots in the form of particle kinematics, which only describes the motion without taking into account the cause of motion. Particle kinematics is used to investigate the agents causing motion change. Through observations and studies on the motion of objects in nature, it can be seen that objects can only move or change their motion when subject to the impact of other objects. In terms of mechanics, we can divide forces into two types: the first includes the forces that appear when there is contact between the interacting objects (elastic force, friction force, etc.), and the second includes the forces that appear when there is no contact between the interacting objects (attraction, repulsion, etc.). In the field of swarm robots, the forces acting on each individual robot in the swarm include the following components: attraction/repulsion between individuals in the swarm, between robots and obstacles in the environment, and between robots and the goals. These forces are also a vector characterized by four factors: setpoint, direction, and magnitude. Dynamic models of swarm robots have been presented in studies [9-11]. Each motion of a robot is characterized by the following system of dynamic equations:

$$\begin{cases} \dot{p}^i = v^i \\ \dot{v}^i = \dfrac{F^i}{m^i} \end{cases} \tag{1}$$

Where: p^i, v^i and m^i are the position vectors, velocity vectors and mass of robot i in 2D space, respectively. Control input F^i of robot i is designed by the interaction forces between robot i and: the remaining robots in the swarm, obstacles, and goal. According to Newton's second law, the single integral of acceleration will give the velocity, and the double integral gives the position, so the swarm model like formula (1) can also be called the "Integral" model dual" of the swarm robots.

3. MULTI-TASK CONTROL OF SWARM ROBOT WITH DOUBLE INTEGRAL MODEL

3.1 Determination of force based on Null space

When the swarm robot performs the task of moving to the goal, it must dodge the obstacles on the way to not be damaged. So each individual robot in the swarm has to fulfill following three tasks:

- First task: avoid obstacles.
- Second task: move to the goal.
- Third task: maintain the swarm so that the individuals in the swarm do not hit each other provided that the groups are not separated.

In order to control the robots to perform the tasks mentioned above, the supervisor can choose the priority level of these tasks. In this study, the author chooses the priority level in the following order: avoiding obstacles, moving to the goal and finally maintaining the swarm. With an assumption that the obstacles are fixed and known in advance, together with the control technique based on Null spatial behavior, the force vectors acting on each robot by tasks are synthesized according to the diagrams of Figures 1 and 2.

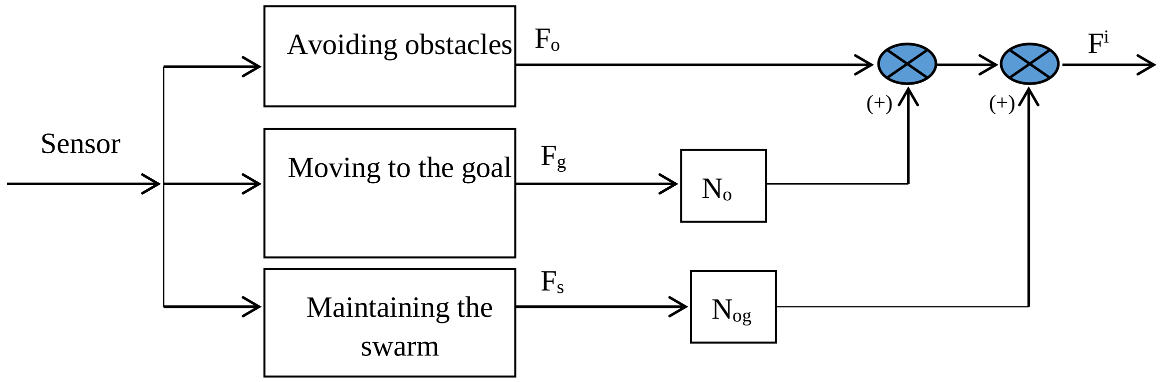

Figure 1. The block diagram of the force vector of the robot i.

The force acting on robot i is determined as follows:

$$F^i = F^o + N_o F^g + N_{og} F^s \tag{2}$$

where: F^o, F^g and F^s are the interaction force vectors, respectively, performing the tasks of avoiding obstacles, moving to the goal, and maintaining the swarm; N_o , N_{og} are orthogonal projection matrices, also known as Null matrices, defined based on the priority level of tasks.

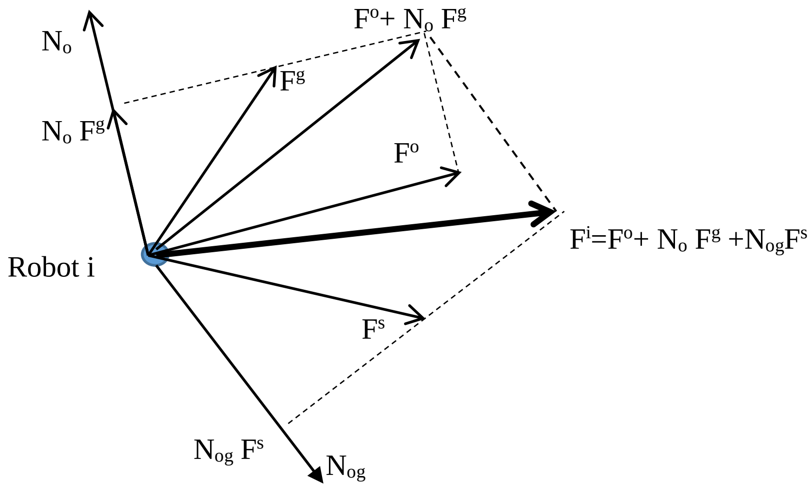

Figure 2. Diagram of forces by NSB method when robot i performs three tasks.

- Determination of the force for the robot to avoid obstacles:

It is an assumption that, there are M obstacles in the environment of the swarm robot, call $p^{o_m} = \begin{bmatrix} p_x^{o_m} \\ p_y^{o_m} \end{bmatrix} \in R^{2\times1}$: is the position of the m^th obstacle (m=1÷M) in 2D Eclied space, $d_o \in R$: so the actual distance between robot i and obstacle m is:

$$d_o = \left\| p^{o_m} - p^i \right\| = \sqrt{\left(p_x^{o_m} - p_x^i\right)^2 + \left(p_y^{o_m} - p_y^i\right)^2}$$

The desire of controlling the robot to avoid obstacles: if the obstacle is on the way the robot moves to the goal, the robot must keep a safe distance away from the obstacle. d_o^*(also called the desired distance) $d_o = d_o^*$; If the obstacle is outside the moving range of the robot, the obstacle will not affect its moving velocity, meaning that the moving force of the robot depends on the distance between the robot and the obstacle:

Jacobi matrix $J_o \in R^{M\times 2}$ represent the force for the robot to avoid obstacles:

$$J_o = \begin{bmatrix} \left[\dfrac{p^{o_1} - p^i}{\left\| p^{o_1} - p^i \right\|}\right]^T \\ \left[\dfrac{p^{o_2} - p^i}{\left\| p^{o_2} - p^i \right\|}\right]^T \\ \vdots \\ \left[\dfrac{p^{o_M} - p^i}{\left\| p^{o_M} - p^i \right\|}\right]^T \end{bmatrix} = \left(\hat{F}^{io}\right)^T \tag{3}$$

Pseudo-inverse matrix of J_o:

$$J_o^+ = \hat{F}^{io}, J_o^+ \in R^{2\times M}$$

Orthogonal projection matrix of J_o:

$$N_o = I_n - \hat{F}^{io}\left(\hat{F}^{io}\right)^T, N_o \in R^{2\times 2} \tag{4}$$

where: I_n is the unit matrix.

The force for robot to avoid obstacles is determined as follows:

$$F^o = -k_o J_o^+ \left(d_o - d_o^*\right) \tag{5}$$

where: k_o is a negative coefficient.

- Determination of the force for the robot to move to the goal:

Call: $p^g = \begin{bmatrix} p_x^g \\ p_y^g \end{bmatrix} \in R^{2\times 1}$ the location of the goal, $d_g \in R$ the actual distance between robot i to the goal, so d_g is calculated according to the following formula:

$$d_g = \left\| p^g - p^i \right\| = \sqrt{\left(p_x^g - p_x^i\right)^2 + \left(p_y^g - p_y^i\right)^2}$$

The desire of controlling the robot to move to the goal is that the robot touches the goal, i.e. the desired distance d_g^* is equal to 0.

Jacobi matrix $J_g \in R^{1\times 2}$:

$$J_g = \left[\frac{p^g - p^i}{\left\| p^g - p^i \right\|} \right]^T = \left(\hat{F}^{ig} \right)^T \tag{6}$$

Pseudo-inverse matrix of J_g: $J_g^+ = \hat{F}^{ig}, J_g^+ \in R^{2\times1}$

Orthogonal projection matrix of J_g:

$$N_g = I_n - \hat{F}^{ig} \left(\hat{F}^{ig} \right)^T, N_g \in R^{2\times2} \tag{7}$$

The force for robot i to move to the goal is rewritten as:

$$F^g = k_g J_g^+ \left(d_g - d_g^* \right) \tag{8}$$

where: k_g is a positive coefficient.

- Determination of the force vector to maintain the swarm:

The actual distance between the robots i and j (j=1÷N, j≠i) is d_s:

$$d_s = \left\| p^j - p^i \right\| = \sqrt{\left(p_x^j - p_x^i \right)^2 + \left(p_y^j - p_y^i \right)^2}$$

The control is to keep the distance between the two robots at a constant value d_s^* in order to maintain the swarm. This is one of the very important tasks in controlling the swarm robots. In the study [5-7], the author designed a formula to calculate attraction/repulsion between robots based on fuzzy logic provided that it satisfies the following requirements: the farther the distance between the robots is, the greater the attraction between the robots is, and the closer distance between the robots is, the greater the repulsion between the robots is. Specifically, as follows:

- Step 1:

➢ The input signal is $u = d_s - d_s^*$, assuming that u has the range of $[\alpha_b, \beta_b] \in R$, dividing u into $2N_f+1$ intervals of B^k as shown in figure 3.

➢ The output signal is $A = f(u)$ with a domain within $[\alpha_a, \beta_a]$, dividing A into $2N_f+1$ intervals of A^k with k = 1, 2, …, $2N_f+1$ (figure 4) and the centroid a^k of A^k the fuzzy interval is as so:

$$a^k \begin{cases} < 0, k = 1, 2, ...N_f \\ = 0, k = N_f + 1 \\ > 0, k = N_f + 2, N_f + 3, ...2N_f + 1 \end{cases} \tag{9}$$

- Step 2: Establish $2N_f+1$ luật IF… THEN… rules with the following form: IF $u=B^k$ THEN $a=A^k$.

- Step 3: Choose the association rule, defuzzify by the weighted average method, we get the control rule (10):

$$f(u) = \frac{\sum_{k=1}^{2N_f+1} a^k \mu_{B^k}(u)}{\sum_{k=1}^{2N_f+1} \mu_{B^k}(u)} \tag{10}$$

With the solution for the establishing of fuzzy function as shown above, we have yielded the result which is the relationship between input signal being the distance and the output signal being the interactive force between individuals (i,j) with the following properties (11):

$$f(u)\begin{cases} >0, d_s > d_s^* \\ =0, d_s = d_s^* \\ <0, d_s < d_s^* \end{cases} \tag{11}$$

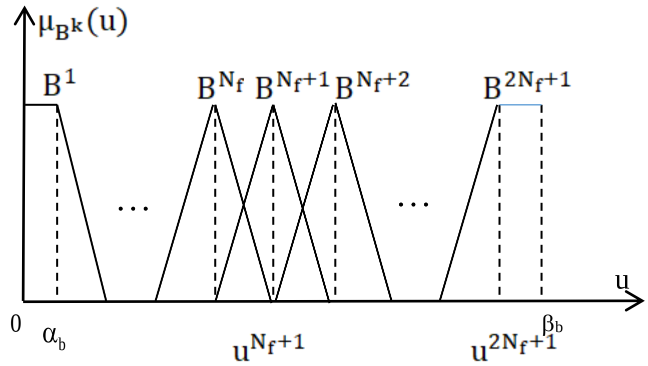

Figure 3. Fuzzyfying the input signal of the fuzzy control function.

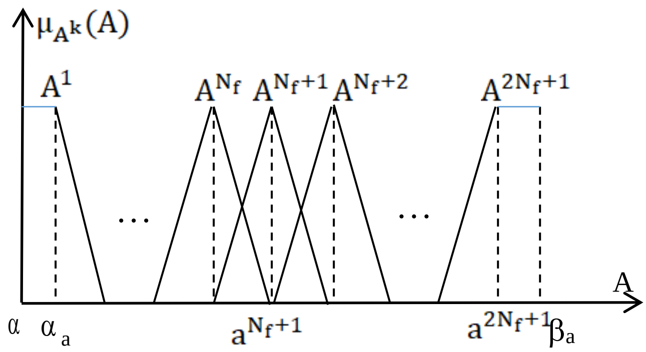

Figure 4. Fuzzyfying the output signal of the fuzzy control function.

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

- Upper and lower limit:

$$-A_{min} \le f(u) \le A_{max} \tag{12}$$

in which: $A_{min}=-a^1$, $A_{max}=a^{2N_f+1}$

- Piecewise linearization equation:

$$f(u)=\frac{\left(a^{k+1}-a^k\right)u+a^k u^{k+1}-a^{k+1}u^k}{u^{k+1}-u^k} \tag{13}$$

in which: $u\in\left[u^k,u^{k+1}\right]$, với $k\in\{1,2,...2N_f\}$

The Jacobi matrix represents the process of maintaining the swarm:

$$J_s = \left(\hat{F}^s\right)^T = \begin{bmatrix} J_{s1} \\ J_{s2} \\ \vdots \\ J_{sN} \end{bmatrix} = \begin{bmatrix} \left(\hat{F}^{s1}\right)^T \\ \left(\hat{F}^{s2}\right)^T \\ \vdots \\ \left(\hat{F}^{sN}\right)^T \end{bmatrix} = \begin{bmatrix} \left[\dfrac{p^1 - p^i}{\left\|p^1 - p^i\right\|}\right]^T \\ \left[\dfrac{p^2 - p^i}{\left\|p^2 - p^i\right\|}\right]^T \\ \vdots \\ \left[\dfrac{p^N - p^i}{\left\|p^N - p^i\right\|}\right]^T \end{bmatrix} \in R^{N\times 2} \tag{14}$$

Pseudo-inverse matrix of J_s:

$$J_s^+ = \hat{F}^s = \begin{bmatrix} J_{s1} \\ J_{s2} \\ \vdots \\ J_{sN} \end{bmatrix}^T = \begin{bmatrix} \left(\hat{F}^{s1}\right)^T \\ \left(\hat{F}^{s2}\right)^T \\ \vdots \\ \left(\hat{F}^{sN}\right)^T \end{bmatrix}^T = \begin{bmatrix} \left[\dfrac{p^1 - p^i}{\left\|p^1 - p^i\right\|}\right]^T \\ \left[\dfrac{p^2 - p^i}{\left\|p^2 - p^i\right\|}\right]^T \\ \vdots \\ \left[\dfrac{p^N - p^i}{\left\|p^N - p^i\right\|}\right]^T \end{bmatrix}^T \in R^{2\times N} \tag{15}$$

Orthogonal projection matrix of J_s:

$$N_s = I_n - \hat{F}^s \left(\hat{F}^s\right)^T, N_s \in R^{2\times 2} \tag{16}$$

The force for robot i to maintain the swarm is determined as follows:

$$F^s = J_s^+ f\left(d_s - d_s^*\right) \in R^{2\times 1} \tag{17}$$

• Synthesize force vectors by 3 tasks to be performed by each individual robot in the swarm according to NSB method as shown in Figure 2:

$$F^i = F^o + N_o F^g + N_{og} F^s = -k_o J_o^+\left(d_o - d_o^*\right) + k_g N_o J_g^+\left(d_g - d_g^*\right) + N_{og} J_s^+ f\left(d_s - d_s^*\right) \tag{18}$$

where: $F^i \in R^{2\times 1}$ $J_{og} = \begin{bmatrix} J_o \\ J_g \end{bmatrix} \in R^{(M+1)\times 2}$, $N_{og} = I_n - J_{og}^+ J_{og} \in R^{2\times 2}$,

3.2. Algorithm of controlling swarm robots for multi-tasks

In order to control the swarm robots to perform three tasks including avoiding obstacles, moving to the goal and maintaining the swarm, the following steps must be taken::

- Step 1:
- Enter the number of robots in the swarm: N.
- Enter the number of obstacles in the motion space of the swarm robots: M.

- Set initial positions for robots in 2D space: $p^1 = \begin{bmatrix} p_x^1 \\ p_y^1 \end{bmatrix}, \ldots, \; p^N = \begin{bmatrix} p_x^N \\ p_y^N \end{bmatrix}$
- Set the location of M obstacle and the destination g in 2D space: $p^{o_1} = \begin{bmatrix} p_x^{o_1} \\ p_y^{o_1} \end{bmatrix}, \ldots,$ $p^{o_M} = \begin{bmatrix} p_x^{o_M} \\ p_y^{o_M} \end{bmatrix}, \; p^g = \begin{bmatrix} p_x^g \\ p_y^g \end{bmatrix}$
- Enter the mass of each individual robot: m^i
- Enter a safe distance between individual robots and obstacles d_o^*, and the distance between individual robots with each other d_s^*.
- Enter the coefficients k_o and k_g.
- Enter the number of steps K.

➢ Step 2:

- Calculate the distance between robot i (i=1÷N) with each obstacle d_o, between robot i and the goal d_g and between robots i and j (j=1÷N, j≠i) d_s.
- Calculate the attraction/repulsion f according to (13).

➢ Step 3:

✓ Compare the actual distance and safe distance from robot i to obstacle m (m=1÷M). If:

- $d_o \geq d_o^*$: robot i does not need to avoid obstacle o, i.e. $J_o = [0]$.
- $d_o < d_o^*$: robot i needs to avoid obstacle o, now we need to calculate J_o according to formula (5).

Calculate: J_o^+, N_o, F^o.

✓ Compare the actual distance and the desired distance from robot i to the goal. If:

- $d_g = 0$: robot i has reached its the goal g, $J_g = [0]$.
- $d_g > 0$: robot i does not reach the goal g, calculate J_g according to formula (8).

Calculate: J_g^+, N_g, F^g.

Calculate: J_{og}, J_{og}^+, N_{og}.

✓ Compare the actual distance and the desired distance from robots i to robot j. If:

- $d_s > d_s^*$: Robots i and j move towards each other by attraction function $f > 0$.
- $d_s < d_s^*$: Robots i and j move away from each other by repulsion function $f < 0$.
- $d_s = d_s^*$: Robots i remain the same route of travel $f = 0$.

Calculate: J_s, J_s^+, F^s.

➢ Step 4:

- Force to move individual i at step k (k=0÷K-1) is determined by the formula (18):

$$F^i[k] = F^o[k] + N_o[k]F^g[k] + N_{og}[k]F^s[k]$$

- Movement acceleration of robot i:

$$\Delta v^i\left[k-1\right]=\frac{F^i\left[k-1\right]}{m^i}\Delta t$$

- Movement velocity of robot i:

$$v^i\left[k\right]=v^i\left[k-1\right]+\Delta v^i\left[k-1\right]\Delta t$$

- The distance that robot i can move:

$$S^i\left[k+1\right]=S^i\left[k\right]+v^i\left[k\right]\Delta t$$

- New coordinates of individual i after (k+1) moves:

$$p^i\left[k+1\right]=p^i\left[k\right]+S^i\left[k+1\right]$$

The loop from step 2 to step 4 is executed until the end of K step.

4. SIMULATION RESULT

For each simulation, the search space is set up on a two-dimensional coordinate system [500, 500]. With an assumption that N=15, M=4, safe distance between robot and obstacle: $d_o^*=5$, safe distance among robots in the swarm: $d_s^*=30$. The simulation results of the movement process when performing the tasks of the swarm robots with different simulation intervals, the mass of each individual robot in each simulation is different as shown in Figures 5 and 6.

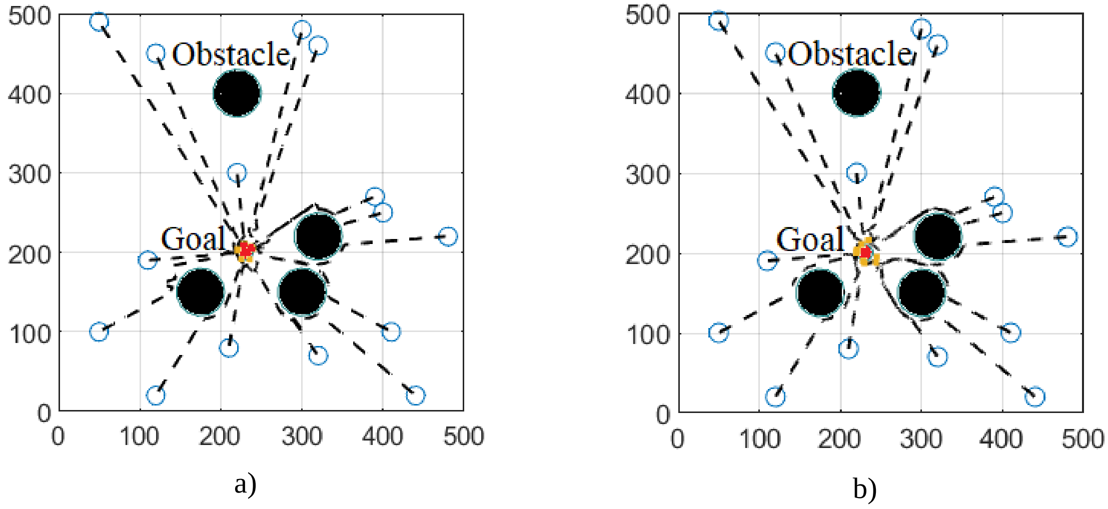

Figure 5. The movement of swarm robots in the interval t= 30s when the mass changes.

a) m=[1,5 1,1 1,8 1,4 1,0 1,2 1,1 1,7 1,1 1,8 1,7 1,9 1,5 1,1 1,8].

b) m=[6,9 6,4 5,6 5,2 13,6 13,0 9,8 13,0 5,1 6,0 8,3 7,5 12,0 11,3 13,4].

Figure 5 shows that with the same conditions, the robot's mass change will affect the movement of that individual.

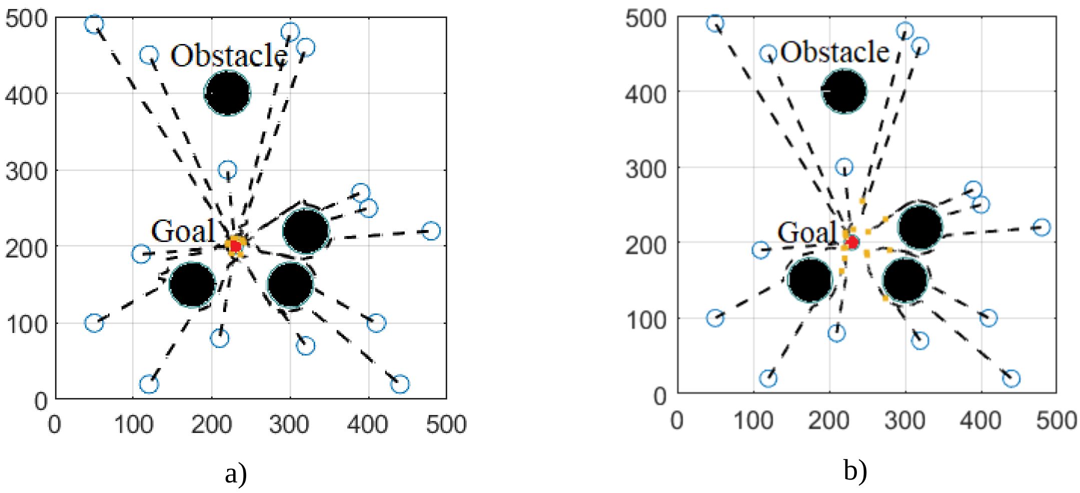

Figure 6. The movement of swarm robots in the interval t=10s when the mass changes.

a) m=[1,5 1,1 1,8 1,4 1,0 1,2 1,1 1,7 1,1 1,8 1,7 1,9 1,5 1,1 1,8].

b) m=[6,9 6,4 5,6 5,2 13,6 13,0 9,8 13,0 5,1 6,0 8,3 7,5 12,0 11,3 13,4].

Figure 6 shows that the larger the mass of robot is, the higher the inertia is, and the robot's lower movement speed results in a longer interval for the whole swarm to converge at the goal.

With the same initial coordinates, each robot in the swarm has constant mass: when the coefficient $k_o<0$ và $k_g>0$ then the individual robots found the goal and avoided the obstacles (figure 7a), when $k_o>0$ and $k_g>0$ then the individual robots found the goal but cannot avoid the obstacles (figure 7b), when $k_o>0$ and $k_g<0$ then the individual robots did not find the goal (figure 7c).

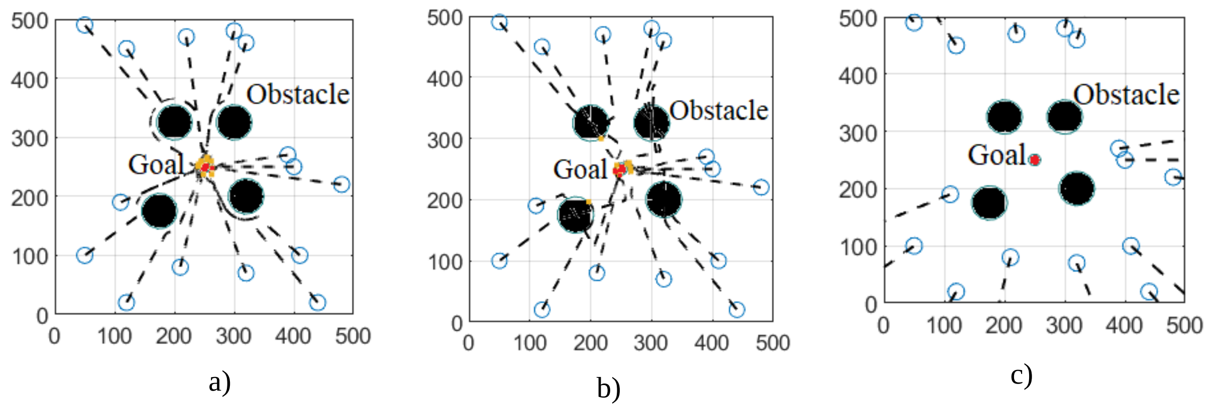

Figure 7. Movement of the swarm robots while the coefficients k_o and k_g change.

a) $k_o<0$, $k_g>0$.

b) $k_o>0$, $k_g>0$.

c) $k_o>0$, $k_g<0$

5. CONCLUSION

This article proposes a method to control the swarm robots to perform multi-tasks by using Null space-based behavioral control techniques and combining fuzzy logic taking into account the robot's mass. Matlab simulation results have confirmed the correctness of the proposed algorithm which is suitable for the control of swarm robots to avoid obstacles, search for the goal and maintain swarms in environments with many obstacles.

ACKNOWLEDGMENT

This research is funded by University of Transport and Communications (UTC) under grant number T2022-ĐĐT-014TĐ.

REFERENCES

[1]. V. Gazi, M. Passino, Stability Analysis of Swarms, IEEE Transactions on Automatic Control, 48 (2003) 692-697. https://doi.org/10.1109/TAC.2003.809765

[2]. L. Wang, H. Fang, Stability Analysis of Practical Anisotropic Swarms, International Conference on Control Automation Robotics & Vision, 2011, IEEE, Singapore, pp.768-772, https://doi.org/10.1109/ICARCV.2010.5707423

[3]. Z. Xue, J. Zeng, C. Feng, Z. Liu, Swarm Target Tracking Collective Behavior Control with Formation Coverage Search Agents & Globally Asymptotically Stable Analysis of Stochastic Swarm, Journal of Computers, 8 (2011) 1772-1780. https://doi.org/10.4304/jcp.6.8.1772-1780

[4]. Miswanto, I. Pranoto, H. Muhammad, D. Mahayana, The Collective Behaviour of Multi-Agents System for Tracking a Desired Path, International Journal of Basic & Applied Sciences, 1 (2011) 81-86.

[5]. Le Thi Thuy Nga, Le Hung Lan, Analysis of stability swarm determination of the swarm robot using the function fuzzy suck/push, Transport and Communications Science Journal, 10 (2013) 88-93.

[6]. Le Thi Thuy Nga, Le Hung Lan, Control Swarm Robots Foraging and Obstacle Avoidance Using Fuzzy Logic, Transport and Communications Science Journal, 3 (2014) 15-20.

[7]. Le Thi Thuy Nga, Nguyen Van Binh, Le Hung Lan, Stabilizing the Flexible Convergence of Swarm Robot by Fuzzy Attraction and Repulsion, Control Engineering and Applied Informatics, 4 (2022) 58-66.

[8]. Nga Le Thi Thuy, Thang Nguyen Trong, The Multitasking System of Swarm Robot based on Null-Space-Behavioral Control Combined with Fuzzy Logic, Micromachines, 8 (2017) 1-16, https://doi.org/10.3390/mi8120357

[9]. Anh Duc Dang, Joachim Horn, Formation Control of Leader Following UAVs to Track a Moving Target in a Dynamic Environment, Journal of Automation and Control Engineering, 1 (2015) 1-8. https://doi.org/10.12720/joace.3.1.1-8

[10].H. G. Tanner, A. Jadbabaie, G. J. Pappas, Stable Flocking of Mobile Agents, Part I: Fixed Topology, Proceedings of the 42nd IEEE Conference on Decision and Control, 2003, IEEE, USA, pp. 2010-2015. https://doi.org/10.1109/CDC.2003.1272910

[11].Y. Liu, K. M. Passino, Stability Analysis of Swarms in a Noisy Environment, Proceedings of the 42nd IEEE International Conference on Decision and Control, 2003, pp.3573-3578. https://doi.org/10.1109/CDC.2003.1271702