

ROBOTIC SWARM SELF-ORGANISATION CONTROL

Zenon HENDZEL*, Jakub WIECH*

*Faculty of Mechanical Engineering and Aeronautics, Department of Applied Mechanics and Robotics, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

zenhen@prz.edu.pl, j.wiech@prz.edu.pl

received 26 June 2018, revised 25 June 2019, accepted 28 June 2019

Abstract: This article proposes a new swarm control method using distributed proportional-derivative (PD) control for self-organisation of swarm of nonholonomic robots. Kinematics control with distributed proportional-derivative (DPD) controller enables generation of desired robot trajectory achieving collective behaviour of a robotic swarm such as aggregation and pattern formation. Proposed method is a generalisation of virtual spring-damper control used in swarm self-organisation. The article includes the control algorithm synthesis using the Lyapunov control theory and numeric simulations results.

Keywords: swarm robotics self-organization, PD controller, nonholonomic robots

1. INTRODUCTION

The problem of swarm self-organisation is one of the several elementary collective behaviours observed in nature. The examples of swarming behaviours are line formations in ant colonies, bird flocks, fish schools and so on. These basic formations enable animals to enhance foraging success or give better protection from predators. In case of robotic swarms, formation control enables achieving collective goals in orderly manner, very often more efficiently than without behaviours of this type. Area coverage, oil plume removal, swarm transport and object manipulation are some of the examples in which swarm formation control is widely used (Brambilla et al., 2013).

The self-organisation process allows for pattern creation from randomly distributed robots. The robots can establish formation using only local information about their neighbouring robots or using a global reference if it is required by the control algorithm. By self-organisation, we mean the emergence of order in a robotic swarm as the result of local interactions amongst robots constituting the swarm. Existing methods of swarm formation control are either bio-inspired, based on physics phenomena, or derived from control theory. Bio-mimicking methods were the first to emerge, for example, Reynolds study of flocks, herds and schools (Reynolds, 1987) or starlings flocking analysis by Hildenbrandt et al. (2010). Rauch et al. (1995) studied class of models for pattern formation based on the behaviours of social insects. Similar works are presented in Christensen et al. (2009), Hsieh (2008) and Trianni (2008). Physics-inspired methods are also popular, thanks to the well-described mathematical equations that are the basis for developing a control algorithm. For example, by imagining the robots being connected to each other by virtual spring and damper, forcing them to move, the swarm will eventually aggregate to one point or, with careful adjustment of virtual parameters, the swarm will establish a formation (Balkacem and Foudil, 2016; Urcola et al., 2008; Shucker and Bennett, 2005; Spears et al.,

2004; Wiech et al., 2018). Another approach is to develop the control algorithm using stability and optimisation theory (Cheah et al., 2009). The works of Gazi and Passino (2003, 2004) and Gazi (2005) based on artificial potential functions allow for formation control, foraging, swarm tracking and so on. In the following work, a swarm self-organisation control algorithm was formulated, inspired by virtual spring damper method with stability proof using the Lyapunov techniques.

2. DESIRED ROBOT TRAJECTORY GENERATION

The kinematics control with distributed proportional-derivative (DPD) controller will be used for desired trajectory generation for each member of a swarm. In case of mentioned physics-based method (Fig. 1), the robots were connected by virtual spring dampers, which resulted in formation creation. The net virtual force exerted on a robot lead the robot to a point in space R_z , which can be described as a local stability point. The position of R_z is a function of time, and it is related to springs deformations between the robot and its neighbours. As generalisation of said method, we present an approach different from the virtual spring dampers.

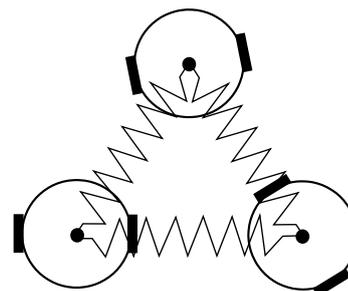


Fig. 1. Virtual physics based method

2.1. Geometric relations

The robot R and its neighbouring robots $R_{s1}, R_{s2}, \dots, R_{si}$ are minimising the distance differences e_1, e_2, \dots, e_i between current distances between robots l_1, l_2, \dots, l_i and the desired distance δ . For this purpose, the differences e_1, e_2, \dots, e_i were substituted with the distance error e_δ pictured in Figure 2.

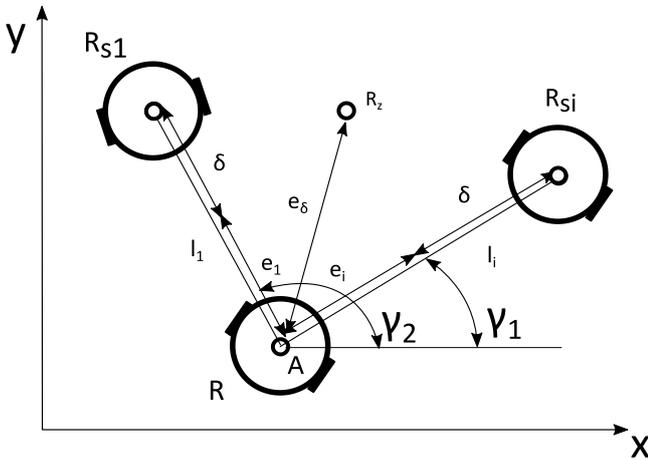


Fig. 2. Distances in the swarm

The value of the distance error e_δ is defined as follows:

$$e_\delta = \sqrt{\sum_{i=1}^n e_{yi}^2 + \sum_{i=1}^n e_{xi}^2}, \quad (1)$$

where $e_{xi} = e_i \cos \gamma_i$ and $e_{yi} = e_i \sin \gamma_i$

The distance difference e_i is described using equations:

$$e_i = l_i - \delta, \quad (2)$$

$$e_i = \sqrt{e_{yi}^2 + e_{xi}^2}. \quad (3)$$

All robots are equipped with distance sensors with limited range. Sensor range determines how many robots are detected. In order to detect only the nearest neighbours, the sensing range, SR, has to meet the inequality:

$$\delta\sqrt{3} > SR > \delta. \quad (4)$$

If we set the range $SR = 1.2\delta$, the swarm will form the equilateral triangle formation. Geometric interpretation of SR value is depicted in Figure 3.

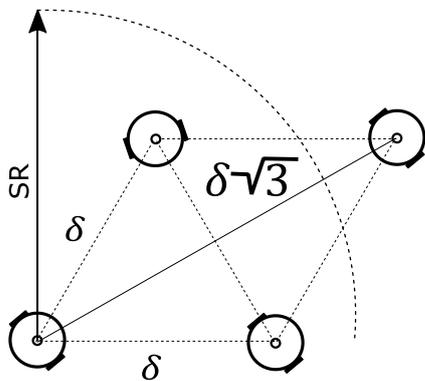


Fig. 3. Robot sensor range for sensing only nearest neighbours

2.2. Stability analysis of the algorithm generating the desired trajectory

Each member of the swarm calculates the distance error e_δ and an angle ψ between robot heading and e_δ based on the position of its neighbouring robots (Fig. 4). In the following equations, the index i was omitted.

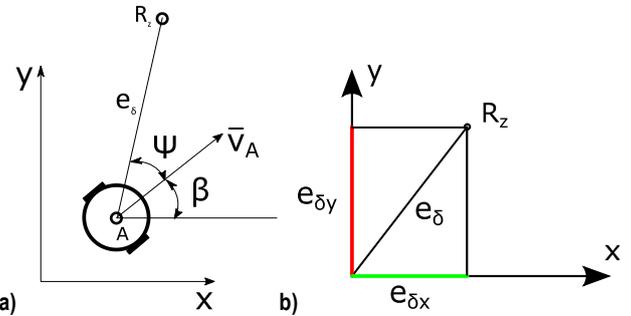


Fig. 4.(a) Robot R in relation to point R_z and (b) projections of e_δ

The value of e_δ is defined using equation (1), whereas ψ is calculated using the following formula:

$$\psi = \arctg\left(\frac{e_{\delta y}}{e_{\delta x}}\right) - \beta, \quad (5)$$

where $e_{\delta x}$ and $e_{\delta y}$ are the projections of e_δ on the axes x and y (Fig. 4b).

The kinematics equations of the mobile robot in relation to the goal position R_z in polar coordinate system are

$$\begin{aligned} \dot{e}_\delta &= -v_A \cos \psi, \\ \dot{\psi} &= v_A \frac{\sin \psi}{e_\delta} - \dot{\beta}. \end{aligned} \quad (6)$$

Error vector \mathbf{q}_p in polar coordinates is defined as

$$\mathbf{q}_p = [e_\delta, \psi]. \quad (7)$$

Differentiating the error vector yields the following equations:

$$\begin{aligned} \dot{e}_\delta &= -u_v \cos \psi, \\ \dot{\psi} &= u_v \frac{\sin \psi}{e_\delta} - u_\beta, \end{aligned} \quad (8)$$

where u_v and u_β are the generated velocities v_A and $\dot{\beta}$, respectively.

To determine the control law $\mathbf{u}_B = [u_v, u_\beta]^T$ allowing the calculation of robot motion parameters v_A and $\dot{\beta}$, the Lyapunov stability theory was used. We consider the following positive definite function:

$$V = V_1 + V_2 = \frac{1}{2} e_\delta^2 + \frac{1}{2} \psi^2. \quad (9)$$

By differentiating the function V , we obtain

$$\dot{V}_1 = e_\delta \dot{e}_\delta = -e_\delta u_v \cos \psi, \quad (10)$$

$$\dot{V}_2 = \psi \dot{\psi} = \psi \left(\frac{u_v \sin \psi}{e_\delta} - u_\beta \right). \quad (11)$$

Using the DPD controller in the form

$$u_{DPD} = k_{DP} e_\delta + k_{DD} \dot{e}_\delta, \quad (12)$$

where k_{DP} and k_{DD} are the proportional and derivative distributed controller gain coefficients, respectively.

The linear velocity control u_v is defined as

$$u_v = u_{DPD} \cos \psi. \quad (13)$$

From equations (8), (12) and (13), we have

$$u_v = \frac{k_{DPE_\delta} \cos \psi}{1+k_{DD} \cos^2 \psi}. \quad (14)$$

Whilst the angular velocity control u_β is calculated from the equation

$$u_\beta = u_{DPD} \frac{\cos \psi \sin \psi}{e_\delta} + \lambda \psi, \quad (15)$$

where λ is the angular proportional gain coefficient.

With equations (8), (12) and (15), u_β is described as

$$u_\beta = \frac{k_{DPE_\delta} \cos \psi \sin \psi}{1+k_{DD} \cos^2 \psi} + \lambda \psi. \quad (16)$$

Using equations (10),(11), (15) and (16), we can determine the sign of \dot{V} :

$$\dot{V}_1 = -e_\delta u_v \cos \psi = -\frac{k_{DPE_\delta^2} \cos^2 \psi}{1+k_{DD} \cos^2 \psi}, \quad (17)$$

$$\begin{aligned} \dot{V}_2 &= \psi \left(\frac{u_v \sin \psi}{e_\delta} - u_\beta \right) = \\ \psi \left(\frac{k_{DPE_\delta} \cos \psi \sin \psi}{1+k_{DD} \cos^2 \psi} - \frac{k_{DPE_\delta} \cos \psi \sin \psi}{1+k_{DD} \cos^2 \psi} - \lambda \psi \right) &= -\lambda \psi^2, \end{aligned} \quad (18)$$

$$\begin{aligned} \dot{V} = \dot{V}_1 + \dot{V}_2 &= -\frac{k_{DPE_\delta^2} \cos^2 \psi}{1+k_{DD} \cos^2 \psi} - \lambda \psi^2 < 0 \quad \forall k_{DD} > \\ 0, k_{DP} > 0, \lambda > 0. \end{aligned} \quad (19)$$

Thus the derivative of the proposed function (9) satisfies the inequality $\dot{V} < 0$, which means that the function V is the Lyapunov function, and we can conclude the asymptotic stability of the equilibrium point $q_p = [0,0]$.

2.3. Robot control algorithm

We consider a swarm of N two-wheel mobile robots (Fig.4a) whose dynamics of the i th robot can be described as (Giergiel and Żylski, 2005)

$$M\ddot{q} + C(\dot{q})\dot{q} + F(q) = \tau, \quad (20)$$

where $q = [\alpha_1, \alpha_2]^T$ is the vector of generalised coordinates (rotation angles of both wheels), M is the inertia matrix, $C(\dot{q})\dot{q}$ is the vector of Coriolis and centripetal forces, $F(q)$ is the wheels friction vector, $\tau = [\tau_1, \tau_2]^T$ denotes the control inputs and τ_1, τ_2 are driving torques.

The desired angular kinematic parameters $\dot{q}_d = [\dot{\alpha}_{1d}, \dot{\alpha}_{2d}]^T$ of a robot are calculated from the following equations (Giergiel and Żylski, 2005):

$$\dot{\alpha}_{1d} = \frac{u_v}{r} + u_\beta \frac{l}{r}, \quad \dot{\alpha}_{2d} = \frac{u_v}{r} - u_\beta \frac{l}{r}, \quad (21)$$

where $\dot{\alpha}_{1d}$ and $\dot{\alpha}_{2d}$ are the desired wheels angular velocities, respectively; r is the wheel radius; l is the half of the length of the drive axle (Fig. 5); u_v is the linear velocity of the point A; and u_β is the angular velocity of the robot frame.

Desired trajectory generation schematics is shown in Figure 6.

For tracking the desired trajectory generated by the DPD controller by a point A_i of the i th robot, the following proportional-derivative (PD) controller is used [9,12]:

$$u = k_p e_t + k_D \dot{e}_t, \quad (22)$$

where k_p is the proportional tracking controller gain, k_D is the derivative tracking controller gain, $e_t = q_d - q$ is the trajectory tracking error and $\dot{e}_t = -\dot{q}$ is the derivative of the trajectory tracking error.

Control schematics of trajectory tracking is shown in Figure 7.

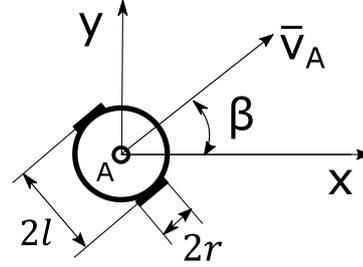


Fig. 5. Scheme of a two-wheeled robot

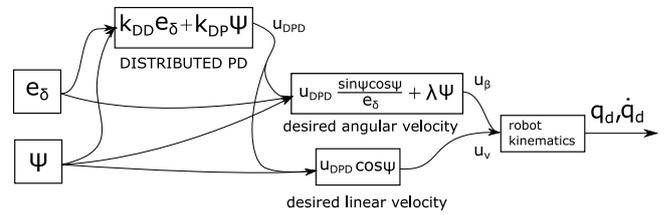


Fig. 6. Scheme of desired trajectory generation

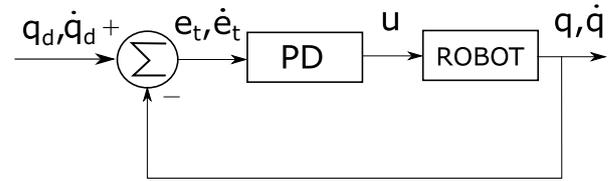


Fig. 7. Scheme of desired trajectory tracking

2.4. Stability analysis of robot trajectory tracking algorithm

To prove the stability of trajectory tracking, the introduction of servomotor equation is necessary. The servomotor dynamics is described by the equation (Spong and Vidyasagar, 1989)

$$J_k \ddot{\theta}_k + (B_k + K_D K_M / R) \dot{\theta}_k = K_M / R V_k - r_k \tau_k, \quad (23)$$

where $k = 1, 2$, J_k is the sum of the actuator and gear inertias, θ_k is the rotor position [rad], B_k is the viscous friction coefficient, K_D is the back emf constant, K_M is the torque constant, R is the armature resistance, V_k is the armature voltage and r_k is the gear ratio.

Using the relation between the rotor position and wheel position $\theta_k = \alpha_k / r_k$, the equation (23) will become

$$\frac{1}{r_k^2} J_k \ddot{\alpha}_k + \frac{1}{r_k^2} (B_k + K_D K_M / R) \dot{\alpha}_k = \frac{K_M}{r_k R} V_k - \tau_k. \quad (24)$$

Substituting $B_k + \frac{K_D K_M}{R} = B$ and $\frac{1}{r_k^2} J_k = J$, we have

$$J \ddot{\alpha}_k + B \dot{\alpha}_k = \frac{K_M}{r_k R} V_k - \tau_k. \quad (25)$$

Combining equations (23) and (20), the robot dynamics with servomotor dynamics in matrix notation will be of the form

$$[M + J]\ddot{q} + [C(\dot{q}) + B]\dot{q} + F(\dot{q}) = u, \quad (26)$$

where $u = [u_1, u_2]^T$ and $u_k = \frac{K_M}{r_{kR}} V_k$ for $k=1,2,$

For control signal u , we propose PD controller described by equation (22) that can be rewritten as

$$u = k_P e_t - k_D \dot{q}. \quad (27)$$

We consider the following positive definite function:

$$V = \frac{1}{2} \dot{q}^T [M + J] \dot{q} + \frac{1}{2} e_t^T k_P e_t. \quad (28)$$

The derivative of function V is

$$\dot{V} = \dot{q}^T [M + J] \ddot{q} - \dot{q}^T k_P e_t. \quad (29)$$

From equations (26) and (29), we have

$$\dot{V} = \dot{q}^T [u - B\dot{q} - F(\dot{q}) - k_P e_t] \dot{q} - \dot{q}^T C(\dot{q}) \dot{q}. \quad (30)$$

From robot dynamics, we know that the matrix $C(\dot{q})$ is a skew symmetric. Substituting equation (27) into equation (30), we will have

$$\dot{V} = -\dot{q}^T [k_D + B + F(\dot{q})] \dot{q}. \quad (31)$$

To prove stability, we use LaSalle's theorem [18]. Suppose $\dot{V} \equiv 0$. Then equation (31) implies that $\dot{q} \equiv 0$ and, hence, $\ddot{q} \equiv 0$. From equations (26) and (27), we have

$$[M + J]\ddot{q} + [C(\dot{q}) + B]\dot{q} + F(\dot{q}) = k_P e_t - k_D \dot{q}. \quad (32)$$

Substituting $\dot{q} \equiv 0$ and $\ddot{q} \equiv 0$ and knowing that $F(0) = 0$, we obtain

$$0 = -k_P e_t, \quad (33)$$

which implies that $e_t = 0$. LaSalle's theorem then implies that the equilibrium is asymptotically stable.

3. SIMULATIONS

As per the performance evaluation, two simulations were performed with swarm having 21 constituting robots. The robots were initially randomly distributed with random headings (triangles in swarm centre). The final positions and heading of robots are depicted by equally distributed triangles with $\delta = 1.5m$. The control algorithm parameters were set experimentally as follows: $k_{DP} = 2.4, k_{DD} = 1.3, \lambda = 10, k_P = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, k_D = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$.

3.1. Swarm formation

The desired distance δ between robots was set to 1.5m, which leads to dispersion of robots and establishing a formation. The formation is shown in Figure 8; the change in velocity and the distance of robot no. 5 is depicted in Figure 9.

The simulation shows that the swarm self-organise into formation consisting mostly of equilateral triangles of length equal 1.5m. In Figure 8, we can see that some of the distances between robots on the outer edge of the formation are smaller than 1.5m, which could be avoided by setting different values for k_{DP} and k_{DD} .

The maximum velocity v_A the robot can reach is 0.3 m/s. The velocity change at the robot acceleration faze was approximated by a smooth ramp seen in Figure 9a. The distance difference e_δ

reaches 0 after 12s, whereas heading difference ψ reaches 0 in finite time.

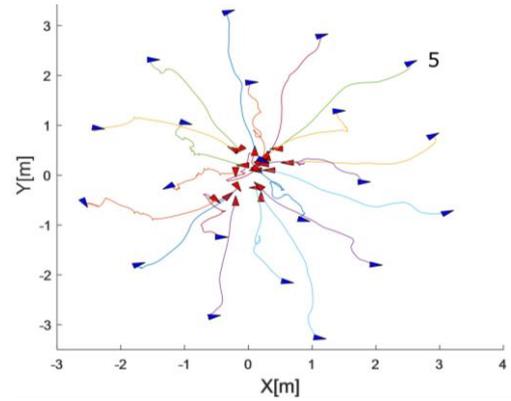


Fig. 8. Self-organisation of 21 robot swarm: trajectories and final formation: robots initial positions depicted in red and final positions in blue

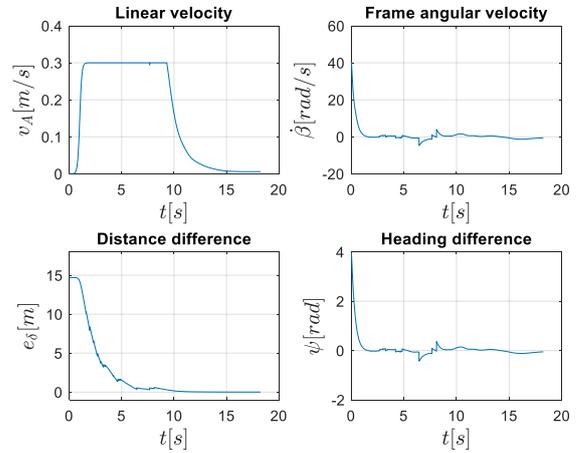


Fig. 9. Simulation results for robot no. 5: (a) linear velocity v_A , (b) frame angular velocity β , (c) distance difference e_δ and (d) heading difference ψ

3.2. Swarm aggregation

Swarm aggregation can be achieved by setting the desired distance δ to 0. All robots would converge to single point. Real robots are constrained by their size; to avoid collisions, the desired distance δ was set to 0.15m, which is equal to the robot diameter (Fig. 10). The velocities and distance differences are shown in Figure 11.

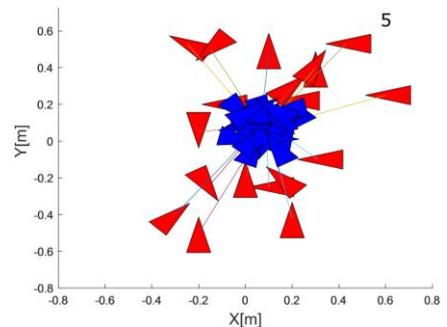


Fig. 10. Swarm aggregation: robots initial positions depicted in red and the final positions in blue

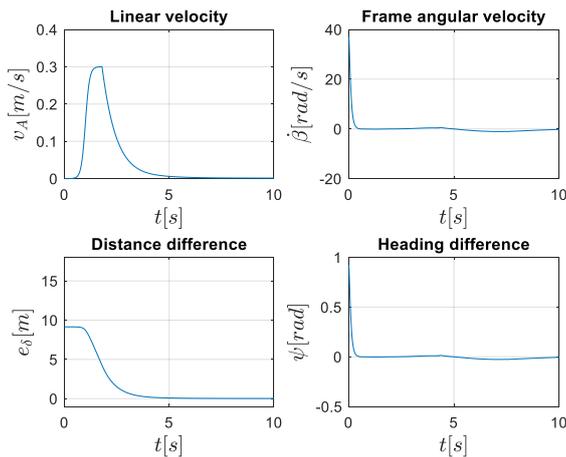


Fig. 11. Simulation results for robot no. 5 in swarm aggregation process: (a) linear velocity v_A , (b) frame angular velocity β , (c) distance difference e_δ and (d) heading difference ψ

4. SUMMARY

In this article, the method of self-organisation of nonholonomic robotic swarm using DPD controller was presented. Numerical simulations show that using this method, a robotic swarm is able to achieve equilateral triangle formation as well as reach the desired neighbourhood of the aggregation point.

Owing to high amplitudes of robots acceleration, it was necessary to introduce a smooth velocity ramp and a proximity zone in which the robot will stop, if it is reached. The proximity zone is a zone of small radius from the equilibrium point, introduced because of the requirement $e_\delta \neq 0$ from equation (6).

The shape of the formation depends on the correctly set DPD parameters as well as the parameters of PD controller used for trajectory tracking. The lower the tracking error is, the faster and more accurate is the swarm self-organisation. It could be beneficial to substitute the PD controller with more precise control method such as adaptive neural control or neural dynamic programming.

It is of interest in future works to address the problem of generating the optimal desired robot trajectories and finding the optimal values of control parameters. Moreover, future works will address swarm leader following and obstacle avoidance.

REFERENCES

1. Balkacem K., Foudil, C. (2016), A virtual viscoelastic based aggregation model for self-organization of swarm robots system, TAROS 2016: Towards Autonomous Robotic Systems, 202–213, .
2. Brambilla M. Ferrante E., Birattari M., Dorigo M. (2013), Swarm robotics: a review from the swarm engineering perspective, *Swarm Intell.*, 7(1), 1-41.
3. Cheah C.C., Hou S.P., Slotine J.J. (2009), Region-based shape control for a swarm of robots, *Automatica*, 45(10), 2406-2411.
4. Christensen A.L., O'Grady R., Dorigo M. (2009), From fireflies to fault-tolerant swarms of robots, *IEEE Transactions on Evolutionary Computation*, 13(4), 754-766.
5. Gazi V. (2005), Swarm aggregations using artificial potentials and sliding-mode control, *IEEE Transactions on Robotics*, 21(6), 1208-1214.
6. Gazi V., Passino K.M. (2003), Stability analysis of swarms, *IEEE Transactions on Automatic Control*, 48(4), 692-697.
7. Gazi V., Passino K.M. (2004), A class of attractions/repulsion functions for stable swarm aggregations, *International Journal of Control*, 77(18), 1567-1579.
8. Giergiel J., Żylski, W. (2005), Description of motion of a mobile robot by Maggie's equations, *J. Theor. Appl. Mech.*, 43(3), 511-521.
9. Hendzel Z. (2007), An adaptive critic neural Network for motion control of Wheeler mobile robot, *Nonlinear Dynamics*, 50, 849-855.
10. Hildenbrandt H., Carere C., Hemelrijk C.K. (2010) Self-organized aerial displays of thousands of starlings: a model, *Behavioral Ecology*, 21(6), 1349-1359,.
11. Hsieh M.A., Halasz A., Bergman S., Kumar V. (2008), Biologically inspired redistribution of a swarm of robots among multiple sites, *Swarm Intelligence*, 2(2-4), 121-141.
12. Lewis F.L., Jagannathan S., Yesildirek A. (1999), *Control of Robot Manipulators and Nonlinear Systems*, Tylor & Frnacjis, London.
13. Rauch E., Millonas M.M., Chialvo D.R. (1995), Pattern formation and functionality in swarm models, *Physics Letters A*, 207(3-4), 185-193.
14. Reynolds C.W. (1987) Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH Computer Graphics*, 21(4), 25-34, New York.
15. Shucker B., Bennett J.K. (2005), *Virtual spring mesh algorithms for control of distributed robotic macrosensors*, University of Colorado at Bulder, Technical Report CU-CS-996-05.
16. Spears W.M., Spears D.F., Hamann J.C., Heil R. (2004), Distributed, physics-based control of swarms of vehicles, *Autonomous Robots*, 17(2-3), 137-162.
17. Spong M.W., Vidyasagar M. (1989), *Robot dynamics and control*, John Wiley & Sons.
18. Trianni V. (2008), Evolutionary Swarm Robotics: Evolving Self-organising Behaviours in Groups of Autonomous Robots, *Studies in Computational Intelligence*, 108. Springer, Berlin.
19. Urcola P., Riazuelo L., Lazaro M.T., Montano L. (2008), Cooperative navigation using environment compliant robot formations, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2789-2794.
20. Wiech J., Eremeyev V.A., Giorgio I. (2018), Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following, *Continuum Mechanics and Thermodynamics*, 1-12, Springer, 2018