Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following
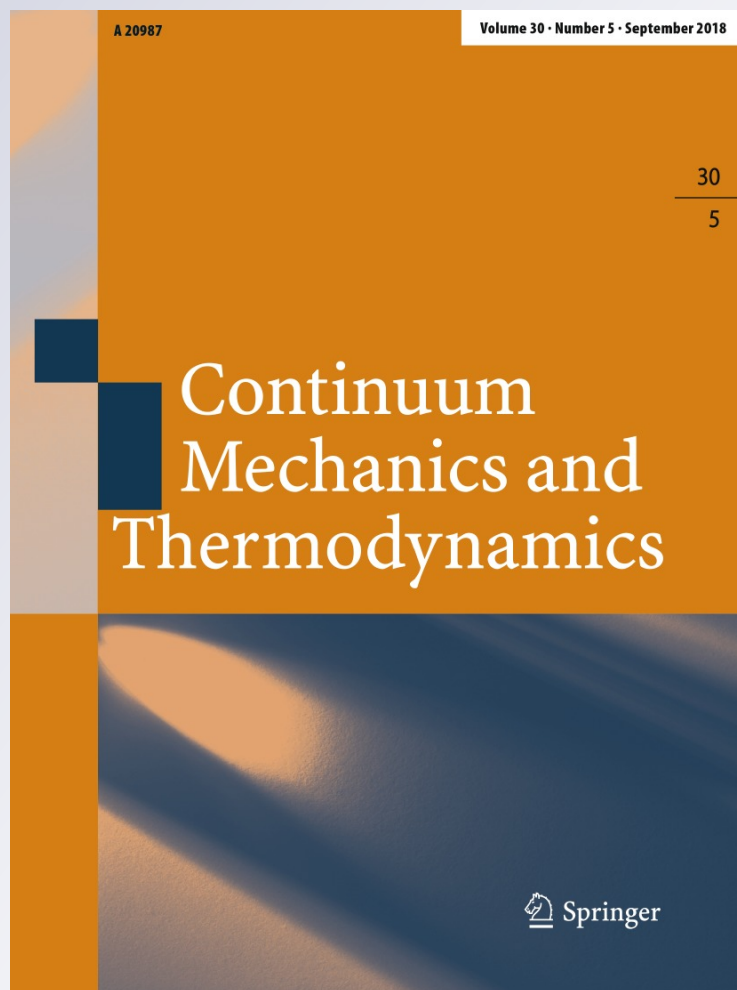
Jakub Wiech, Victor A. Eremeyev & Ivan Giorgio

A 20987

Volume 30 · Number 5 · September 2018

30
5

Continuum Mechanics and Thermodynamics

🦅 Springer

🦅 Springer

Springer

CrossMark

ORIGINAL ARTICLE

Jakub Wiech · Victor A. Eremeyev · Ivan Giorgio

Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following

© The Author(s) 2018

Abstract In this paper, we demonstrate a method for self-organization and leader following of nonholonomic robotic swarm based on spring damper mesh. By self-organization of swarm robots we mean the emergence of order in a swarm as the result of interactions among the single robots. In other words the self-organization of swarm robots mimics some natural behavior of social animals like ants among others. The dynamics of two-wheel robot is derived, and a relation between virtual forces and robot control inputs is defined in order to establish stable swarm formation. Two cases of swarm control are analyzed. In the first case the swarm cohesion is achieved by virtual spring damper mesh connecting nearest neighboring robots without designated leader. In the second case we introduce a swarm leader interacting with nearest and second neighbors allowing the swarm to follow the leader. The paper ends with numeric simulation for performance evaluation of the proposed control method.

1 Introduction

Recently, swarm robotics includes the both robots design and the developments in the controls of their behavior. Here we consider a swarm consisting of two-wheel robots and propose some algorithms for their self-organization. By self-organization of swarm robots we mean the emergence of order in a robotic system as the result of interactions among the single robots constituting the swarm. In other words this self-organization mimics that one which we often observe in biology, and in particular in animal societies, not limited to social insects like bees, ants or termites, see, e.g., [1–5]. For example, swarms of submarine robots which mimics some fish behavior were analyzed in [6,7], where in particular were described how the configuration of the swarm can be calculated from the individual element, in order to achieve the desired shape of the swarm. The objective of swarm control algorithm is to achieve desired swarm behavior by introducing the proper local interaction between swarm members. A collective behavior emerges from all combined decisions made by

Communicated by Francesco dell'Isola.

J. Wiech
Faculty of Mechanical Engineering and Aeronautics, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Rzeszów, Poland

V. A. Eremeyev (✉)
Faculty of Civil and Environmental Engineering, Gdańsk University of Technology, Gdańsk, Poland
E-mail: eremeyev.victor@gmail.com

I. Giorgio
Department of Mechanical and Aerospace Engineering, University of Rome La Sapienza, Rome, Italy

robots according to relatively simple rules. As a decision-making input, robots take into consideration their relative position with respect to the local neighbors and to the environment that they can sense and/or interact with. Besides, other requirements are swarm scalability and robustness that have to be ensured in designing swarm control algorithms. Available methods of swarm control that met all the listed requirements are either bioinspired, based on physics phenomena, or various optimization approaches. Similar works using virtual spring-based approach have been released; for example, Urcola et al. [8] present a formation control based on virtual structure composed of virtual spring dampers connecting following robots to the leader. Shucker and Bennett [9] propose a fully distributed robotic macrosensor in which control algorithm was influenced by spring damper mesh connectivity. Robotic swarm interconnected by virtual spring and damper connected in parallel was proposed by Chen et al. [10] where formation control of satellites under a gravitational potential field is investigated. Belkacem and Foudil [11] study a swarm aggregation model based on equilibrium emerging from robots interconnected with virtual springs and dampers. A slightly different approach can be found in the works [12,13] where a centroid-based interaction is analyzed; in particular, these works are focused on local and nonlocal interactions; in [14], a mathematical proof of the cohesion of a monodimensional swarm is given.

The study of discrete systems as those analyzed is particularly interesting not only to evaluate the behavior of robotic swarms, but can also be used for other purposes: In fact, considering the single constituent of the swarm as a material particle that can interact with neighboring particles with suitable potentials, the discrete approach can be seen as a possible numerical implementation tool for the study of deformable continua which may also show nonlocal interaction effects (see, e.g., for possible fields of application [15–22]). The presented approach to robotic swarm is rather similar to the modeling of a discrete system by a continuum and vice versa which has very long traditions in the continuum mechanics.

In our approach, the swarm control algorithm is similar in the behavior of point mass particles interconnected with spring dampers. Such swarm mimics the behavior of viscoelastic fluid which falls into the category of physics-based method. For movement control of nonholonomic robot, we define the relation between virtual net forces acting on the robots and their linear and angular velocity. The paper is organized as follows. In Sect. 2 we introduce the ordinary differential equations describing the motion of the two-wheel robots. In Sect. 3 we propose new swarm control methods based on spring damper mesh. The results of the model are discussed in Sect. 4 where we considered the motion of 21 two-wheel robots with and without swarm leader. It is shown that the technique based on the control of one robot called swarm leader with properly introduced inter-robot forces may lead to the formation of relative stable ordered structure of the swarm. Moreover controlling the behavior of the swarm leader we can almost get a priory desired configuration of the whole swarm. In addition we also consider the swarm leader technique in the case of avoiding of obstacles.

2 Dynamics of robot swarms

The most commonly used robots in swarm research are nonholonomic two-wheel robots. The robot is shown in Fig. 1.

To describe the dynamics of a two-wheel robot, we could use Euler–Lagrange's equations with multipliers. Unfortunately, one of the drawbacks is the necessity of decoupling the multipliers from robot torques. To avoid having to decouple multipliers we will use different formalism using Maggie's equations [23,24], derived from the principle of virtual work. The equations are given by

$$\sum_{j=1}^{n} C_{ij} \left[\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial E}{\partial \dot{q}_i} \right) - \frac{\partial E}{\partial q_i} \right] = \Theta_i, \quad i = 1, \ldots, s, \quad j = 1, \ldots, n, \tag{1}$$

where s is the number of independent parameters in the generalized coordinates q_j equal to the numbers of the system degrees of freedom. The generalized velocities and the right-hand sides of Eq. (1) can be expressed explicitly by

$$\dot{q} = \sum_{i=1}^{s} \mathrm{C}_{ij} \dot{e}_i + G_j, \tag{2}$$

$$\sum_{i=1}^{s} \Theta_i \delta e_i = \sum_{i=1}^{s} \delta e_i \sum_{j=1}^{n} \mathrm{C}_{ij} Q_j. \tag{3}$$

where $\dot{e}_i$, $i = 1, \ldots s$, are called kinetic parameters and in this case they are equal to $\dot{\alpha}_1, \dot{\alpha}_2$, i.e., the rotation angles of the two wheels.
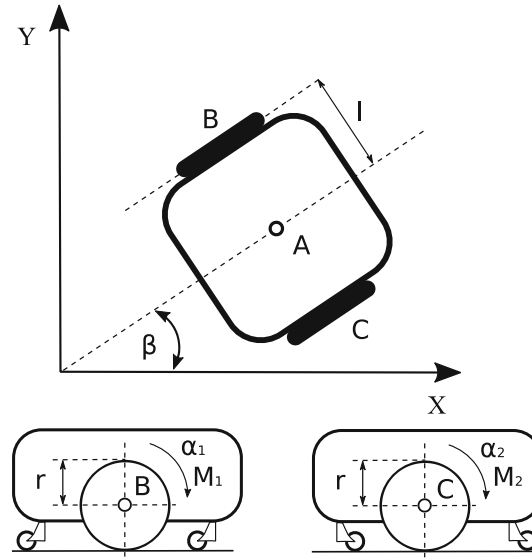
Fig. 1 Two-wheel robot schematic

The generalized coordinates and generalized velocities for two-wheel robot are

$$q = [x_A, y_A, \beta, \alpha_1, \alpha_2]^T,$$
$$\dot{q} = \left[\dot{x}_A, \dot{y}_A, \dot{\beta}, \dot{\alpha}_1, \dot{\alpha}_2\right]^T \tag{4}$$

where β is the angle between the line perpendicular to the segment joining the wheel centers and the x-axis.

For the described robot, we have two degrees of freedom and the independent generalized coordinates are the wheels rotation angles α_1, α_2. Using kinematics analysis, we determine the velocities relationships as follows

$$\dot{\beta} = \frac{r}{2l}\left(\dot{\alpha}_1 - \dot{\alpha}_2\right), \ \dot{x}_A = \frac{r}{2}\left(\dot{\alpha}_1 + \dot{\alpha}_2\right)\cos\beta, \ \dot{y}_A = \frac{r}{2}\left(\dot{\alpha}_1 + \dot{\alpha}_2\right)\sin\beta \tag{5}$$

where r is the radius of the wheels and l half the distance between the wheel centers.

The total kinetic energy E is equal to the sum of the kinetic energies of both wheels and robot chassis:

$$E = E_{w1} + E_{w2} + E_r \tag{6}$$

Explicitly, we have:

$$E_r = \frac{1}{2}m_r v_A^2 + \frac{1}{2}I_{zr}\dot{\beta}^2,$$
$$E_{w1} = \frac{1}{2}m_w v_B^2 + \frac{1}{2}I_{zw}\dot{\alpha}_1^2 + \frac{1}{2}I_{xw}\dot{\beta}^2,$$
$$E_{w2} = \frac{1}{2}m_w v_C^2 + \frac{1}{2}I_{zw}\dot{\alpha}_2^2 + \frac{1}{2}I_{xw}\dot{\beta}^2, \tag{7}$$

where $m_w, m_r, I_{zw}, I_{xw}, I_{zr}$ are, respectively, the masses and the moments of inertia of wheels and robot chassis with respect to the x- and z-axes.

Using Eqs. (2) and (5) we can calculate the coefficients C_{ij}, G_j, and with Eqs. (1) and (3) we determine the generalized forces Q_j in the following form

$$
\begin{array}{llll}
C_{11} = \frac{r}{2}\cos\beta, & C_{21} = \frac{r}{2}\cos\beta, & G_1 = 0, & Q_1 = (2m_w + m_r)\,\ddot{x}_A, \\
C_{12} = \frac{r}{2}\sin\beta, & C_{22} = \frac{r}{2}\sin\beta, & G_2 = 0, & Q_2 = (2m_w + m_r)\,\ddot{y}_A, \\
C_{13} = \frac{r}{2l}, & C_{23} = -\frac{r}{2l}, & G_3 = 0, & Q_3 = 2m_w l^2\ddot{\beta} + 2I_{zw}\left(\frac{l}{r}\right)^2\ddot{\beta} + 2I_{xw}\ddot{\beta} + I_{zr}\ddot{\beta}, \\
C_{14} = 1, & C_{24} = 0, & G_4 = 0, & Q_4 = I_{zw}\ddot{\alpha}_1 \\
C_{15} = 0, & C_{25} = 1, & G_5 = 0, & Q_5 = I_{zw}\ddot{\alpha}_2.
\end{array} \tag{8}
$$

The generalized forces and coefficients C_{ij}, G_j in (8) and Eq. (1) yield the following the Maggie equations

$$(2m_w + m_r)\left(\frac{r}{2}\right)^2 (\ddot{\alpha}_1 + \ddot{\alpha}_2) + (2m_w l^2 + 2I_{xw} + I_{zr})\frac{r}{2l}(\ddot{\alpha}_1 - \ddot{\alpha}_2) + I_{zw}\ddot{\alpha}_1 = M_1 - N_1,$$

$$(2m_w + m_r)\left(\frac{r}{2}\right)^2 (\ddot{\alpha}_1 + \ddot{\alpha}_2) - (2m_w l^2 + 2I_{xw} + I_{zr})\frac{r}{2l}(\ddot{\alpha}_1 - \ddot{\alpha}_2) + I_{zw}\ddot{\alpha}_2 = M_2 - N_2. \tag{9}$$

If we write Eq. (9) in matrix form, we get

$$M\ddot{q} + F(\dot{q}) = U, \tag{10}$$

where M is the inertia matrix, $F(\dot{q})$ is the friction function, and U is the wheels torque vector.

$$M = \begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix},$$

$$F(\dot{q}) = \begin{bmatrix} a_4 \mathrm{sgn}\dot{\alpha}_1 \\ a_5 \mathrm{sgn}\dot{\alpha}_2 \end{bmatrix},$$

$$U = \begin{bmatrix} M_1, M_2 \end{bmatrix}^{\mathrm{T}} \tag{11}$$

The parameters $a_1, \ldots, a_5$ are calculated using the relations

$$a_1 = (2m_w + m_r)\left(\frac{r}{2}\right)^2, a_2 = (2m_w l^2 + 2I_{xw} + I_{zr})\frac{r}{2l}, a_3 = I_{zw}, a_4 = N_1 f_1, a_5 = N_2 f_2, \tag{12}$$

where N_1, N_2 are the loads applied to wheels 1 and 2, while f_1, f_2 are the coefficients of rolling friction of both wheels.

Robot control inputs are wheels angular acceleration $\ddot{q} = [\ddot{a}_1, \ddot{a}_2]^{\mathrm{T}}$ and wheels angular velocities $\dot{q} = [\dot{a}_1, \dot{a}_2]^{\mathrm{T}}$ given by the equations

$$\dot{\alpha}_1 = \frac{v_A}{r} + \dot{\beta}\frac{l}{r}, \quad \dot{\alpha}_2 = \frac{v_A}{r} - \dot{\beta}\frac{l}{r} \tag{13}$$

3 Swarm control method

Swarm members are linked together with springs and dampers connected in parallel, see Fig. 2. We assume the robots have mass m and are connected via springs with stiffness k and dampers with damping coefficient c. The length of the springs at rest is equal to d_0.

Virtual springs and dampers are connected in the centers of mass of each robot, depicted in what follows as point A.

Every link between the robots exerts a force F_{ij} equal to the sum of elastic spring and viscous damper forces such as discussed in [8–11]. We assume that the spring and damper are linear and their forces are proportional, respectively, to the displacement and relative velocity between spring ends.

$$F_{ij} = F_{Sij} + F_{Dij} = k(d_{ij} - d_0) + c\frac{\mathrm{d}}{\mathrm{d}t}(d_{ij} - d_0), \tag{14}$$
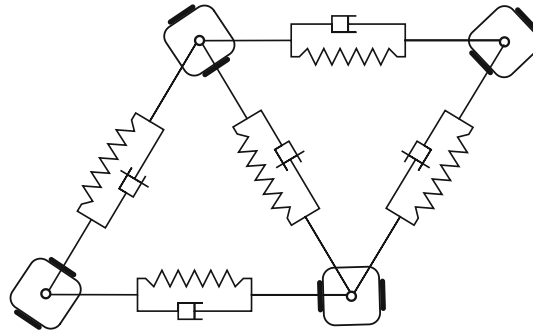

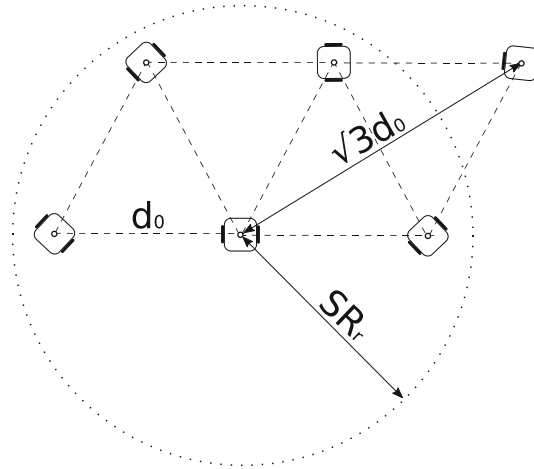Fig. 2 Spring dampers mesh connecting mobile robots

Fig. 3 Sensing range in nearest neighbors link connection

where k stands for the spring stiffness coefficient, c is the damping coefficient, and d_{ij} is the current length of the spring connecting the robots i and j.

Each robot has limited sensing range SR_r which determines how many spring damper links are present; the number of links is, in this way, equal to the number of visible neighboring robots. To establish links with only the nearest neighbors, the sensing range has to meet the inequality

$$d_0\sqrt{3} > SR_r > d_0. \tag{15}$$

For example, setting the sensing range at the value $1.2d_0$ results in creating links between only nearest neighbors forming the swarm in the equilateral triangle formation shown in Fig. 3. Let us note that the sensing range is quite similar to the notion of horizon used in the peridynamics, see [25], or the interaction range in the lattice dynamics [26]. The nearest neighbors do not remain the same all the time, in general.

The total force exerted on a robot i is equal to the sum of all link forces between robot i and its visible neighbors N_r

$$\overline{F_i} = \sum_{j \in N_r} \overline{F_{ij}} = \sum_{j \in N_r} \overline{F_{Sij}} + \overline{F_{Dij}}. \tag{16}$$

From Eq. (13) we conclude that we require values of linear velocity v_A and angular velocity $\dot{\beta}$, to specify robot control inputs, having the net force $\overline{F_i}$ as known value. To determine the values of v_A, $\dot{\beta}$ we first have to find the corresponding forces causing the robot to move forward or backward (force $F_{y'}$) and rotate around the point A (force $F_{x'}$). These forces are shown in Fig. 4.

The origin of the vectors $\overline{F_{y'}}$, $\overline{F_{x'}}$ is in the point D. If we know the values of link forces and their orientation, we can find the x and y components of the total force $\overline{F_i}$ given by

$$\begin{aligned} F_x &= \sum_{j \in N_r} F_{ij} \cos \alpha_{ij}, \\ F_y &= \sum_{j \in N_r} F_{ij} \sin \alpha_{ij}. \end{aligned} \tag{17}$$

The vectors $\overline{F_{y'}}$, $\overline{F_{x'}}$ are the components of the vector F_{ij} in a rotating frame of reference around the point A by an angle $\varphi_i = \beta_i - \frac{\pi}{2}$. The addition of $-\frac{\pi}{2}$ brings $\overline{F_{y'}}$ (instead of vector $\overline{F_{x'}}$) in the direction of the velocity vector $\overline{v_A}$. The values $\overline{F_{y'}}$, $F_{x'}$ are calculated using the following equation

$$\begin{aligned} F'_x &= -F_x \sin \varphi_i + F_y \cos \varphi_i, \\ F'_y &= F_x \cos \varphi_i + F_y \sin \varphi_i. \end{aligned} \tag{18}$$

Using the above equations the values of the linear velocity v_A and the angular velocity $\dot{\beta}$ are determined by

$$\begin{aligned} v_A &= \int \frac{F'_y}{2m_w + m_r} \mathrm{d}t, \\ \dot{\beta} &= -\lambda F'_x, \end{aligned} \tag{19}$$
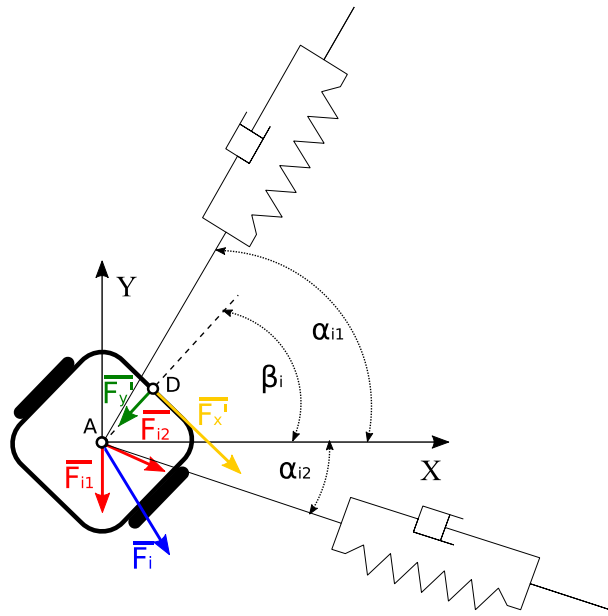
Fig. 4 Robot i driving forces

Table 1 Simulation parameters

m_w (kg)	m_r (kg)	I_{zr} (kgm^2)	I_{xw} (kgm^2)	I_{zw} (kgm^2)	N (N)	l (m)	r (m)	f (m)	k (N/m)	c (Ns/m)	λ (−)	d_0 (m)
0.05	0.4	0.08	0.01	0.03	22	0.058	0.032	0.004	7	1	4	1
0.05	0.4	0.08	0.01	0.03	22	0.058	0.032	0.004	15	1.5	4	1

where λ is the gain constant.

The gain constant λ allows one to control the robots' rotation speed around the point A, aligning the robot frame with the direction of the total force $\overline{F_i}$ acting on the robot. Moreover, the damping coefficient c and stiffness coefficient k can be set by the designer so that the values of v_A and $\dot{\beta}$ are adequate for the task at hand.

4 Simulation experiments

In order to evaluate the performance of the presented control method, two simulations were performed, that is without and with a leader. The robots were initially randomly distributed with random headings. In both simulations 21 robots were used. The parameters and the control algorithm constants for both simulations are listed in Table 1.

4.1 Self-organization without swarm leader

In the first case the swarm self-organizes with no leader, forming equilateral triangle formation, the results are shown in Fig. 5.

In Fig. 6 it is seen some oscillations during first seconds in the linear and angular velocities of the robots which vanish in finite time due to the presence of friction and limited time response of robot actuators. In the first simulation, the formation reached by the swarm is imperfect (disordered), and this may suggest a change of control gains to different values.

Forces from virtual springs and dampers acting on the robots also oscillated around zero or in some cases reached zero. For the second simulation in case of robot no. 5 marked in Fig. 5 the velocities and forces acting on the robot oscillate around zero, whereas heading has nonzero value (Fig. 6).
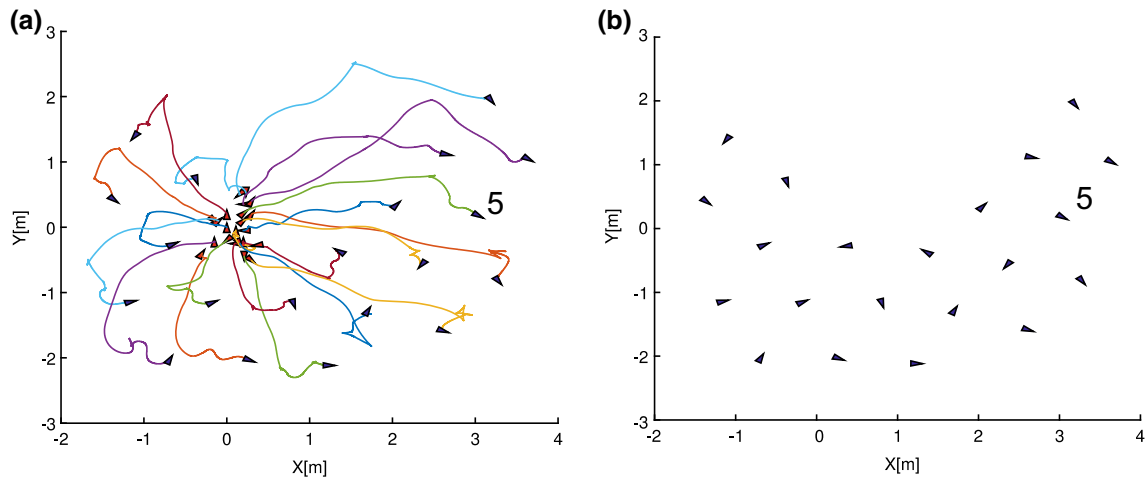
Fig. 5 Swarm self-organization for first set of parameters **a** robots trajectories, **b** final robot positions

Oscillations at the first 2 s are present due to the termination of links with robots going out of range. This process creates sudden change of acting force value as well as its heading, which results in change of angular and linear velocity of the robot. With correctly chosen values of k, c, λ the swarm reaches the desired formation. As we mention before there are some similarities between swarm robots and other discrete models, such as lattices [26]. Let us also underline the similarities with dynamic fracture of discrete lattices, where the breakage of a link leads to the local oscillations or a wave propagation for infinite systems, see, e.g., Slepyan et al. [27–30].

4.2 Self-organization with swarm leader

In the second case one of the robots is designated to be the swarm leader, interacting with nearest and second neighbors. Swarm cohesion is achieved due to two types of interactions. Inter-agent interaction as in previous example and additional virtual forces acted by the leader on its first and second neighbors. In calculating leader–follower forces the same values of k and c are used as given in Table 1. The length of the unstressed spring between leader and its nearest neighbors is equal to d_0, while the one between the leader and its second neighbors is equal to $1.5d_0$.

Followers position themselves around the leader and are shown in Fig. 7. The diameter of the swarm is between $2d_0$ and d_0. In this case it is equal to 1.57 m.

With robots organized around the leader the swarm will follow the leader trajectory. Example of leader following is shown in Fig. 8.

During the swarm movement, the swarm maintains its cohesion. The swarm moves in the same heading as the leader. The individual robots are rotating around the leader to the left, which is the same direction as the first turn of the swarm. The followers continue to rotate with constant speed until the next turn of the swarm to the right which slows the rotation of the robots. If friction is present, the system of robots would eventually stop rotating around its leader.

The swarm trajectory can be represented as trajectory of the swarm center of mass. The swarm trajectory is calculated only for the followers and compared to the trajectory of the leader.

The difference between position of the swarm leader and center of mass is smaller than 4 [cm] in X and Y directions. The calculated error could be used as a design parameter for choosing correct values of k, c, λ and d_0 for swarm leader following.

4.3 Obstacle and collision avoidance

Obstacle and collision avoidance can be achieved using the same approach as for maintaining the cohesion of the swarm. If a robot moves near its neighbor closer than allowed minimal distance d_{min}, a spring damper link is established between the robots creating a repulsive force, moving the robots apart. The same happens in
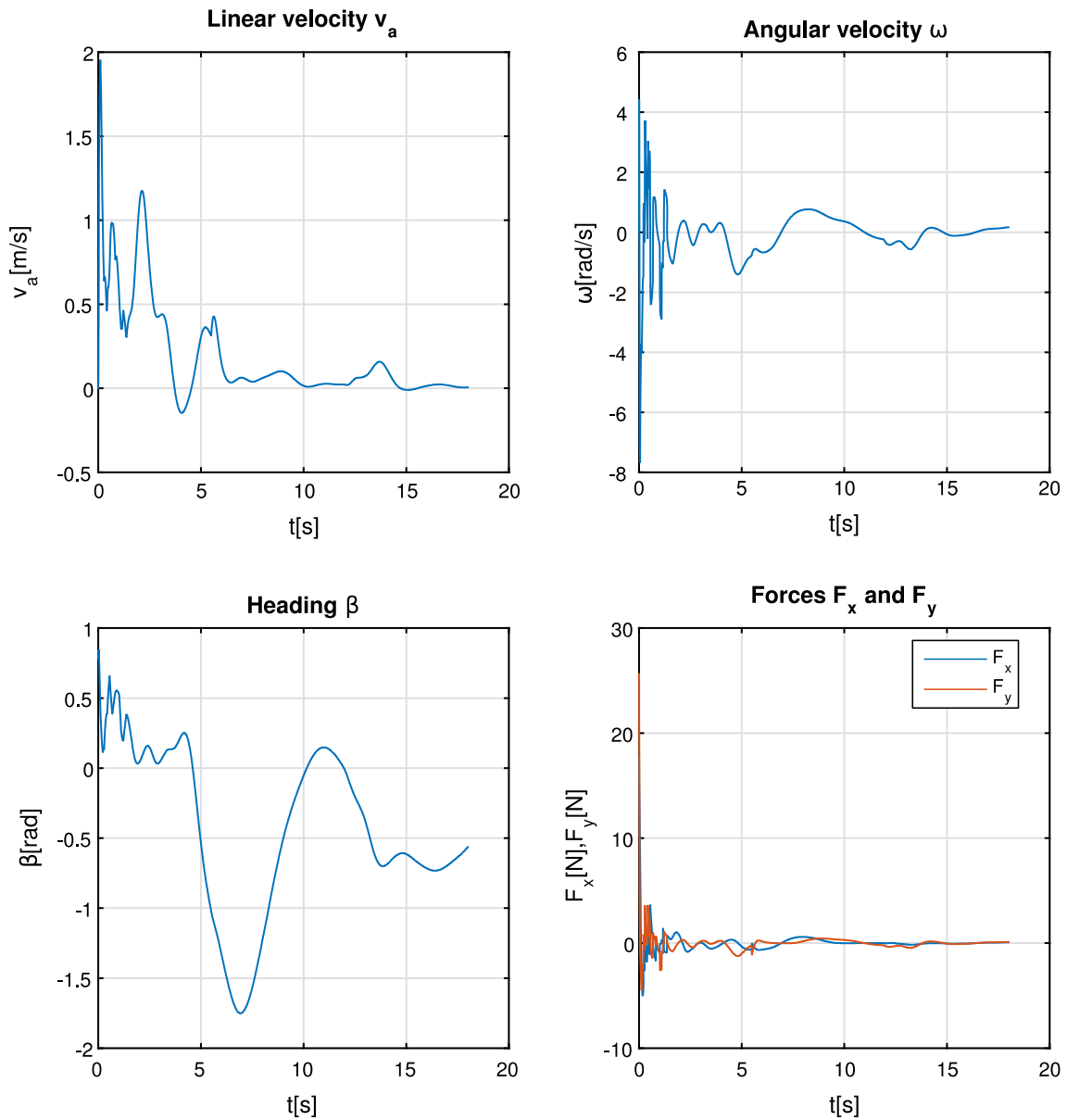
Fig. 6 Simulation results for robot no. 5

case robots driving near obstacles. One of the simplest forms of such forces for i-th robot the force is given by

$$F_i = \begin{cases} k\,(d_{\min} - d)\,, & 0 < d < d_{\min}, \\ 0, & d \geq d_{\min} \end{cases},$$

where $k = k_{\mathrm{ob}}$, $d_{\min} = d_{\mathrm{minob}}$ and for obstacles and $k = k_{co}$, $d_{\min} = d_{\mathrm{minco}}$ to prevent collisions, and d is the distance between i-th robot and the nearest neighbor or the obstacle. The link robot–obstacle is perpendicular to the obstacle and created between the robot and the closest point on the obstacle. Simulations for swarm obstacle avoidance were performed using the parameters given in Table 2.

Here $k_r, k_l, k_{\mathrm{ob}}, k_{co}$ are the spring constants, respectively, for neighboring robots, leader, obstacles and robots collision avoidance, d_0 is the resting spring length for springs between robots, and $d_{\mathrm{minob}}, d_{\mathrm{minco}}$ are the minimal allowed distances from obstacles and between robots.
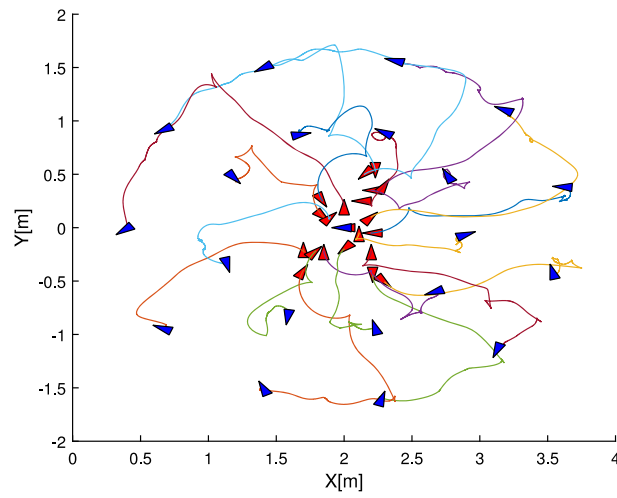
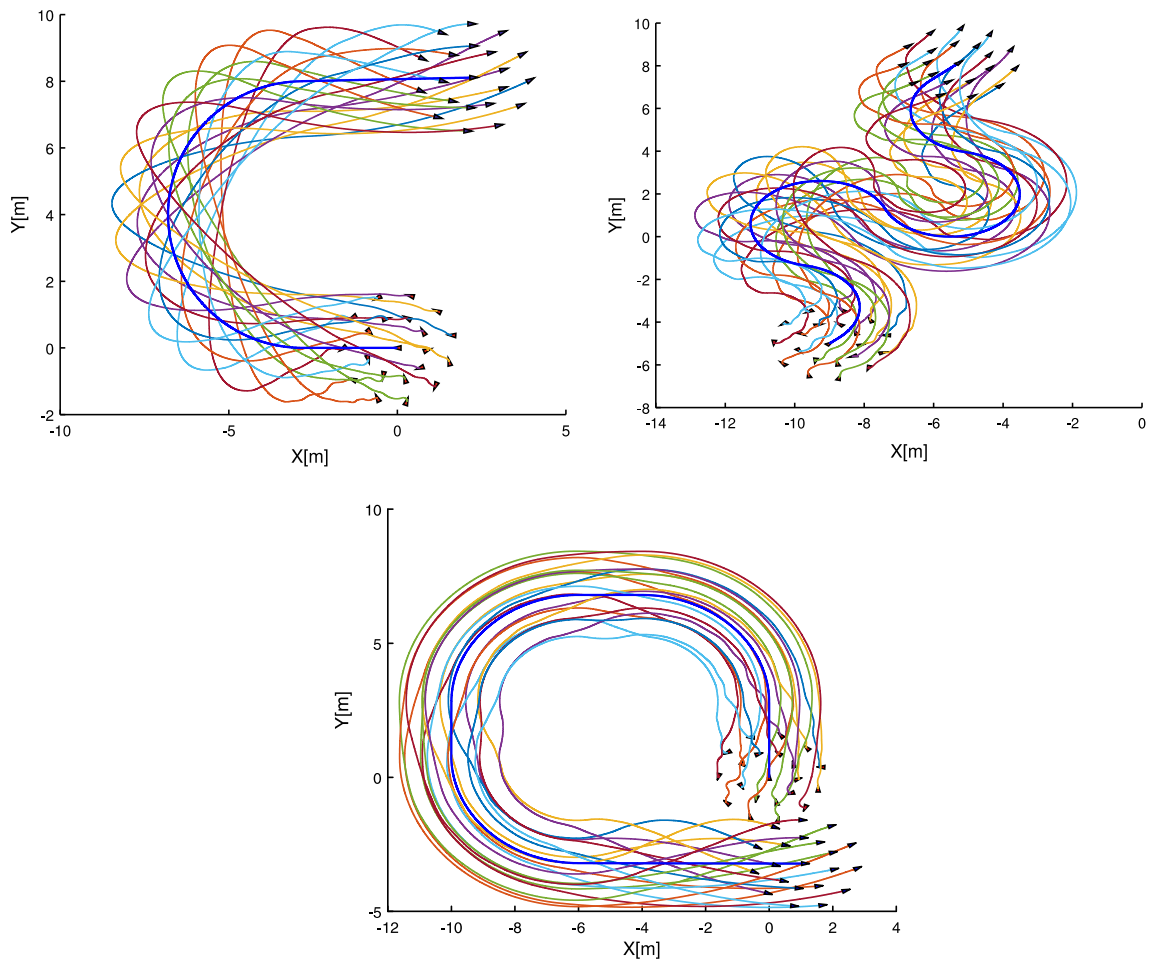**Fig. 7** Self-organization with swarm leader
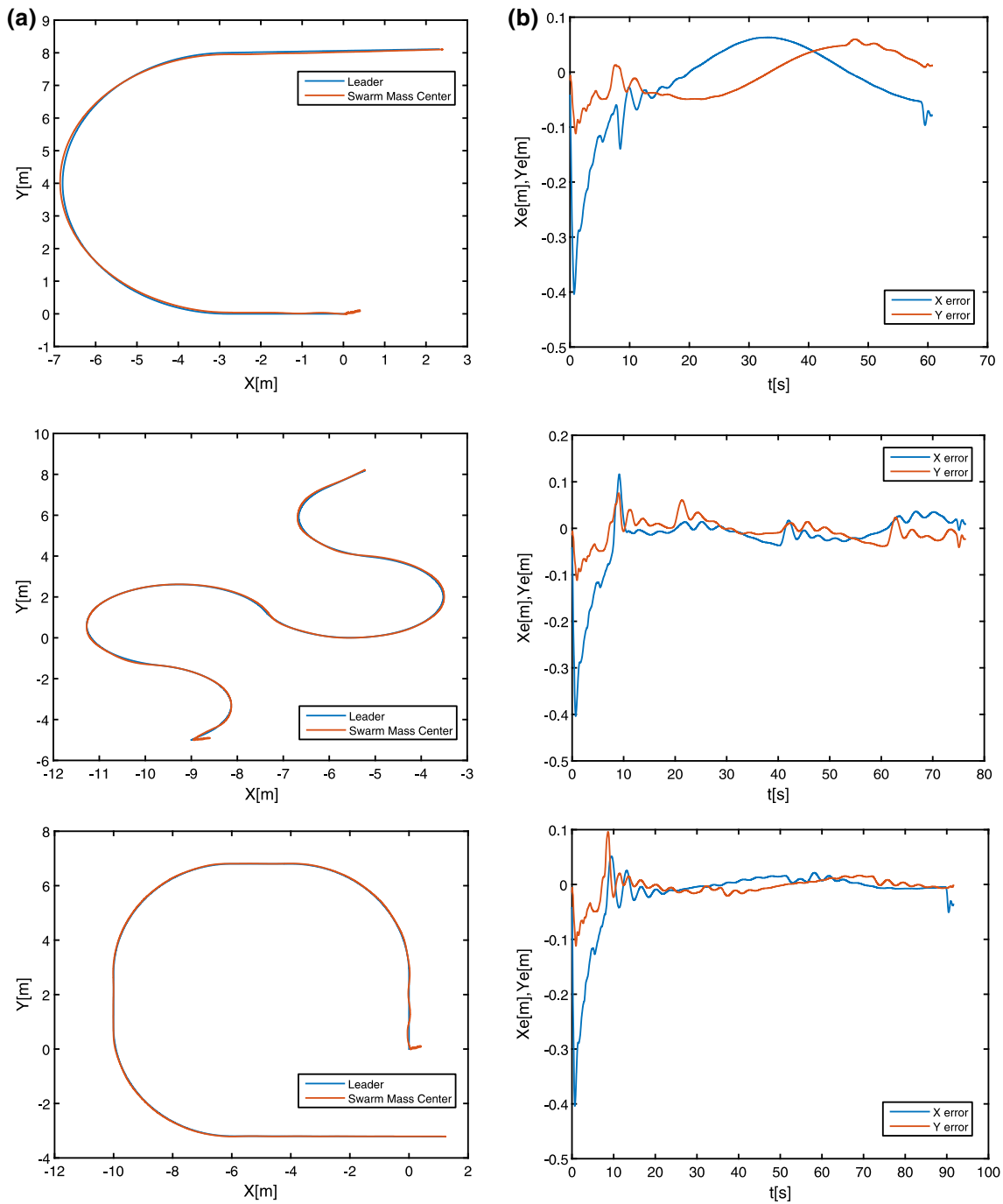

Fig. 8 Swarm following the leader

Fig. 9 Swarm center of mass position compared to the leaders position **a** X and Y error, **b** swarm center of mass and leader trajectory

Table 2 Simulation parameters

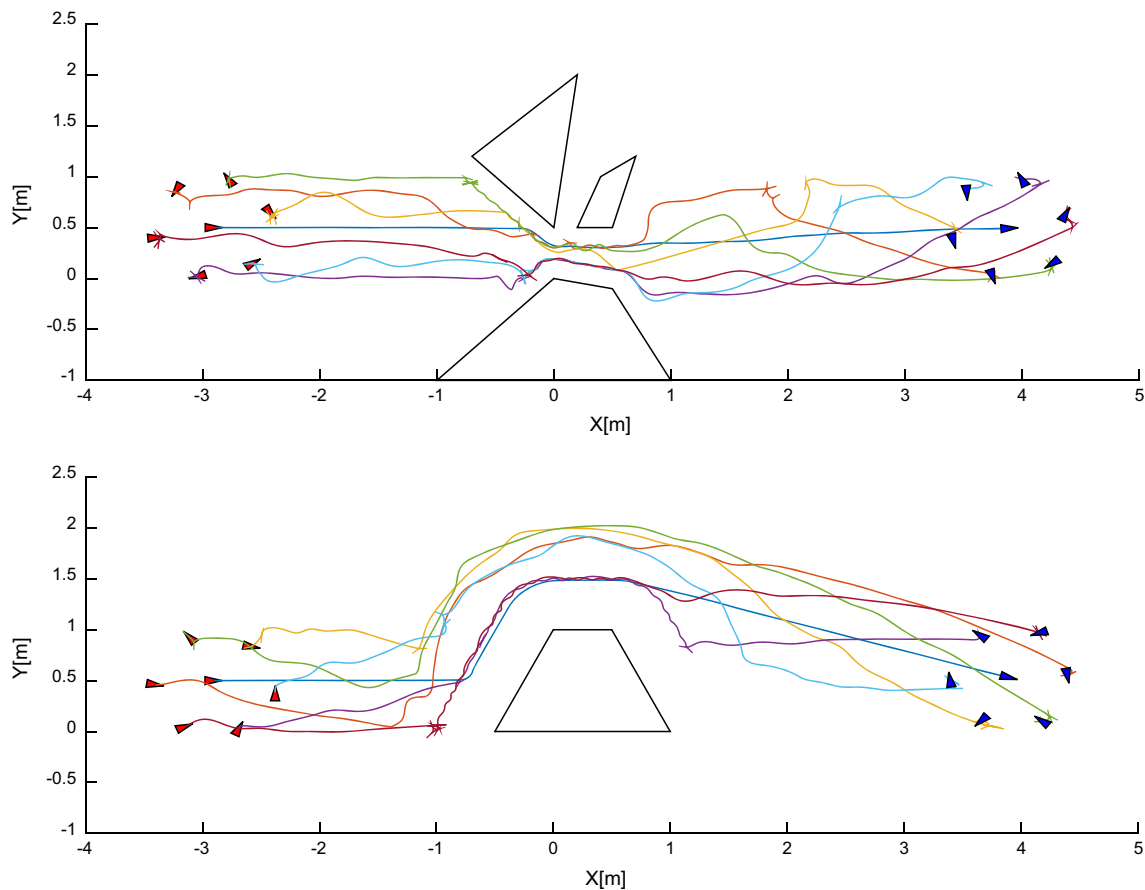| k_r (N/m)	k_l (N/m)	k_{ob} (N/m)	k_{co} (N/m)	c (Ns/m)	d_0 (m)	d_{minob} (m)	d_{minco} (m)
22.5	45	975	135	1	0.5	0.2	0.1

Fig. 10 Examples of swarm obstacles avoidance

5 Conclusions and future work

In this paper, we have presented a method for self-organization of nonholonomic robotic swarm using virtual spring damper mesh. Choosing appropriate values of spring stiffness, damping coefficient and angular velocity constant, the swarm is able to achieve equilateral triangle formation. Moreover, if we introduce a swarm leader, with correct design parameters, the swarm follows the leader almost keeping the established formation. Numerical simulations show that for the detailed nonholonomic vehicle dynamics it is possible to design a self-organizing swarm control algorithm based on restoring force of spring with damping. It is of interest in future works to address to problem of selecting the optimal values of control gains to achieve better algorithm performance including the determination of the level of nonlocality that is determination of the sensing range. Moreover, future works should include the development of an extended version of the presented method with obstacle avoidance and swarm shape formation control.

References

1. Trianni, V.: Evolutionary Swarm Robotics: Evolving Self-organising Behaviours in Groups of Autonomous Robots. Studies in Computational Intelligence, Vol. 108. Springer, Berlin (2008)

2. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
3. Sahin, E., Spears, W. M. (Eds).: Swarm Robots. Lecture Notes in Computer Science book series (LNCS, vol. 3342). Springer, Berlin (2005)
4. Moriconi, C. dell'Erb, R.: Social Dependability: a proposed evolution for future Robotics, Sixth IARP-IEEE/RAS–EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments May 17–18, (2008), Pasadena, California
5. Bossi, S., Cipollini, A., dell'Erba, R., Moriconi, C.: Robotics in Italy. Education, Research, Innovation and Economics outcomes. Enea, Rome, (2014)
6. dell'Erba, R., Moriconi, C.: HARNESS: a robotic swarm for environmental surveillance. In 6th IARP Workshop on Risky Interventions and Environmental Surveillance (RISE). Warsaw, Poland, (2012)
7. dell'Erba, R.: Determination of spatial configuration of an underwater swarm with minimum data. Int. J. Adv. Robotic Syst. **12**(7), 97–114 (2015)
8. Urcola, P., Riazuelo, L., Lazaro, M., Montano, L.: Cooperative navigation using environment compliant robot formations. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 2789–2794, IEEE (2008)
9. Shucker, B., Bennett, J.K.: Virtual spring mesh algorithms for control of distributed robotic macrosensors. University of Colorado at Bulder, Technical Report CU-CS-996-05 (2005)
10. Chen, Q., Veres, S.M., Wang, Y., Meng, Y.: Virtual spring, -damper mesh-based formation control for spacecraft swarms in potential fields. J. Guid. Control Dyn. **38**(3), 539–546 (2015)
11. Balkacem, K., Foudil, C.: A virtual viscoelastic based aggregation model for self-organization of swarm robots system. TAROS 2016: Towards Autonomous Robotic Systems, pp. 202–213, Springer (2016)
12. Della Corte, A., Battista, A., dell'Isola, F.: Referential description of the evolution of a 2D swarm of robots interacting with the closer neighbors: perspectives of continuum modeling via higher gradient continua. Int. J. Non-Linear Mech. **80**, 209–220 (2016)
13. Battista, A. et al.: Numerical investigation of a particle system compared with first and second gradient continua: Deformation and fracture phenomena. Math. Mech. Solids, https://doi.org/10.1177/1081286516657889 (2016)
14. Della Corte, A., Battista, A., dell'Isola, F., Giorgio, I.: Modeling deformable bodies using discrete systems with centroid-based propagating interaction: fracture and crack evolution. In: Mathematical Modelling in Solid Mechanics, pp. 59–88. Springer Singapore, (2017)
15. Pideri, C., Seppecher, P.: A second gradient material resulting from the homogenization of an heterogeneous linear elastic medium. Contin. Mech. Thermodyn. **9**(5), 241–257 (1997)
16. Samuel, F., Sab, K.: Finite-deformation second-order micromorphic theory and its relations to strain and stress gradient models. Math. Mech. Solids. (2017). https://doi.org/10.1177/1081286517720844
17. Steigmann, D.J.: Theory of elastic solids reinforced with fibers resistant to extension, flexure and twist. Int. J. Non-Lin. Mech. **47**, 742–743 (2012)
18. Turco, E., dell'Isola, F., Cazzani, A., Rizzi, N.L.: Hencky-type discrete model for pantographic structures: numerical comparison with second gradient continuum models. ZAMP **67**(4), 1–28 (2016)
19. Alibert, J.-J., Seppecher, P., dell'Isola, F.: Truss modular beams with deformation energy depending on higher displacement gradients. Math. Mech. Solids **8**(1), 51–73 (2003)
20. Buttà, P., De Masi, A., Rosatelli, E.: Slow motion and metastability for a nonlocal evolution equation. J. Stat. Phys. **112**(3–4), 709–764 (2003)
21. Cuomo, M., dell'Isola, F., Greco, L., Rizzi, N.L.: First versus second gradient energies for planar sheets with two families of inextensible fibres: investigation on deformation boundary layers, discontinuities and geometrical instabilities. Compos. Part B Eng. **115**, 423–448 (2017)
22. dell'Isola, F., Cuomo, M., Greco, L., Della Corte, A.: Bias extension test for pantographic sheets: numerical simulations based on second gradient shear energies. J. Eng. Math. **103**(1), 127–157 (2017)
23. Giergiel, J., Żylski, W.: Description of motion of a mobile robot by Maggie's equations. J. Theor. Appl. Mech. **43**(3), 511–521 (2005)
24. Gutowski R.: Mechanika Analityczna, 1971, PWN, Warszawa
25. Madenci, E., Oterkus, E.: Peridynamic Theory and Its Applications. Springer, New York (2014)
26. Born, M., Huang, K.: Dynamical Theory of Crystal Lattices. Clarendon Press, Oxford (1988)
27. Slepyan, L.I.: Models and Phenomena in Fracture Mechanics. Springer, Berlin (2002)
28. Mishuris, G.S., Movchan, A.B., Slepyan, L.I.: Waves and fracture in an inhomogeneous lattice structure. Waves Random Complex Media **17**, 409–428 (2007)
29. Mishuris, G.S., Movchan, A.B., Slepyan, L.I.: Dynamics of a bridged crack in a discrete lattice. Q. J. Mech. Appl. Math. **61**, 151–160 (2008)
30. Slepyan, L.I.: Wave radiation in lattice fracture. Acoust. Phys. **56**(6), 962–971 (2010)