Noname manuscript No.
(will be inserted by the editor)

Polygonal Obstacle Avoidance Method for Swarm Robots via Fluid Dynamics

Shotaro Shibahara · Kenji Sawada

Received: date / Accepted: date

Abstract Swarm robots in obstacle environments perform search tasks efficiently while avoiding obstacles. Obstacle avoidance methods are classified by the way obstacles are represented. Potential functions are typical for circular obstacles, while collision cones and polygonal regions are used for non-circular obstacles. The shape of the obstacle this paper considers is angular. When swarm robots try to avoid angular obstacles with potential functions, they require information such as the relative angle between the robot and the obstacle and the size of the obstacle. This paper focuses on the shape parameters of angular obstacles and proposes an easier way to avoid angular obstacles in column formation. The key idea is fluid dynamics. The proposed method is based on a Polygon-Wall expression of MPS (Moving Particle Semi-implicit) method, a fluid dynamics method. The Polygon-Wall expression allows robots to measure the distance from the obstacle wall and avoid the obstacle without using the relative angles and the obstacle sizes. Usual MPS methods have the challenge of requiring centralized computation, while typical swarm robots are based on distributed computation. Therefore, this paper addresses this challenge by introducing potential functions into MPS method. The contribution of this paper is to propose a distributed obstacle avoidance method for angular obstacles with a small number of parameter designs for swarm robots. The results of numerical experiments reflecting these ideas show the effectiveness of the proposed method.

Keywords Swarm Robots · Obstacle Avoidance · MPS method · Polygon-Wall · Formation Control

This paper is funded by JSPS KAKENHI Grant Number 22H00519, 22H01509, JP19H02158, JP19K04444, and JP19H02163.

Shotaro Shibahara
Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Tokyo, Japan
Tel.: +042-443-5891
E-mail: s-shibahara@uec.ac.jp

Kenji Sawada
Info-Powered Energy System Research Center, The University of Electro-Communications, Tokyo, Japan
Tel.: +042-443-5253
Fax: +042-443-5253
E-mail: knj.sawada@uec.ac.jp

1 Introduction

A multi-agent system (MAS) is a system in which multiple agents cooperate with each other to accomplish tasks [1–4]. Agents share information with each other through the network to accomplish tasks more efficiently than if they work alone. One example of tasks performed by swarm robots is a search task that they perform while staying formation [5].

In such a task, round or angular-shaped obstacles may be present, which the robot must avoid. Potential method is one of the typical methods used in path planning and involves the design of a function that generates attracting and repulsion forces with different positive and negative values at the obstacle and target positions [6–8]. Potential functions are often used for round obstacles because the functions become popular in a circular fashion. Although this method also approximates angular obstacles with round obstacles, it is not easy to represent an angular obstacle directly with a potential function. Then, a number of researches taking into account he obstacle shape have been studied [9–12]. [9] proposes the obstacle magnification method which magnifies but retains the detected obstacle and [10] proposes the collision cone approach which has its origin in aerospace domain. Also, other researches [11,12] are being conducted on robot path planning for polygonal environments. In these cases, a polygonal environment is prepared to deal with angular obstacles.

Unlike previous studies [9–12], this paper tries to propose an angular obstacle avoidance algorithm based on the potential-like method. Our research question is: "What is an avoidance method that can be implemented distributed to each robot like the potential method, but with fewer configuration parameters for angular obstacles than the potential method?" As one answer to this research question, we focus on an angular obstacle design method of fluid dynamics. The reasons for introducing fluid dynamics to swarm robots are two.

The first reason is the academic interest in incorporating the behavior of collections of active particles (active matter) into swarm robot control. Theory of active matter [13,14] analyzes the collections of active particles, including organic and inorganic matter and thus covers a wide range of subjects such as fluid dynamics and swarm robots. It is natural to apply control laws that reflect the active matter to swarm robots, and this leads to the integration of control engineering and fluid mechanics. This paper focuses on the analogy between swarms and fluids in obstacle avoidance.

The second reason is to propose a new obstacle avoidance method for swarm robots. Previous studies applying fluid properties to swarm robot control have been done in the literature [15,16]. The previous study [15] applies the MPS (Moving Particle Semi-implicit) method, a fluid analysis method for incompressible fluids, to swarm robotics control. Based on the discussion in Reference [15], Reference [16] verifies through numerical experiments whether the pressure calculation between particles of the MPS method is applicable for the collision avoidance for swarm robots. This paper tries to remove some assumptions made in [15,16] when applying the MPS method to swarm robots and then to realize an obstacle avoidance method that better reflects fluid characteristics.

The challenge for swarm robots to realize fluidic phenomenon is to perform pressure calculations in a distributed manner. In [15,16], the fluid pressure is centrally calculated at the same time as the swarm robots are centrally controlled. The behavior of active particles in fluid depends on the interaction between them, which is a distributed control aspect. Therefore, the decentralized control collision avoidance method reproducing the fluid behaviors is this paper's novelty and advantage compared to the centralized methods in [15, 16]. As a first step in our study, this paper implements an approximate distributed calculation of the particle pressures by the potential function.

Also, this paper introduces Polygon-Wall [17], an obstacle design method different from the previous studies [15,16], to swarm robots. The MPS method determines the pressure value of each particle and the pressure gradient prevents particles from colliding with each other. This concept of collision avoidance between particles is also used for avoidance against obstacles. Meanwhile, its obstacle boundaries are not smooth. To resolve the disadvantage, reference [17] proposes the concept of Polygon-Wall. Polygon-Wall has a smooth obstacle boundaries and calculates the repulsion force from the obstacle only using the distance between the particle and the obstacle.

Further, this paper compares the proposed method based on the Polygon wall with the potential method in terms of configuration parameters. Reference [8] proposes an obstacle avoidance method using a time-varying potential function. This method is helpful for avoiding angular obstacles, but, has a difficulty of parameter settings. Meanwhile, the proposed method only needs to distance sensors from an obstacle, regardless of the shape of the obstacle. This may make it easier to design angular obstacles than reference [8]. In this paper, the angular obstacle avoidance in [8] is compared with the proposed method. This paper deals with multitasking such as maintaining formation, reaching the target position, and obstacle avoidance. For multitasking, fewer design parameters for each control input are more effective. Using numerical experiments, we show that the proposed parameter configurations for angular obstacles become easier than the potential method.

The organization of this paper is as follows. Section 2 describes the basic theories used in this paper. Section 3 describes the problem setting. Section 4 describes the proposed method based on Section 2. In Section 5, numerical experiments are conducted using MATLAB and the conclusion of this paper is given in Section 6.

2 Preliminaries

This section describes the basic theories behind the MAS and MPS method used in the proposed method [17–19].

2.1 Graph Theory

The structure of network communication among agents is represented using graph $G = (V, E)$ composed of the vertex set $V = \{1, 2, \cdots, n\}$ and the edge set $E \subseteq V \times V$. For agent $i(\in V)$ and $j(\in V)$, $(i, j) \in E$ indicates that there is an information transfer path between the two agents. In this case, agent i and j are neighbors. A graph G in which all information transfer paths are bidirectional is called an undirected graph. An adjacent matrix $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } (j,i) \in E \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The degree matrix $\boldsymbol{D}$ is defined by the number d_i^{in} of edges entering vertex i as follows:

$$d_i^{\text{in}} = \sum_{j=1}^{n} a_{ij}, \quad i = 1, 2, \cdots, n,$$
$$\boldsymbol{D} = \text{diag}\left(d_1^{\text{in}}, d_2^{\text{in}}, \cdots, d_n^{\text{in}}\right).$$

From the adjacent matrix $\boldsymbol{A}$ and the degree matrix $\boldsymbol{D}$, the Graph Laplacian $\boldsymbol{L}$ and Perron matrix $\boldsymbol{P}$ are given by

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A},$$
$$\boldsymbol{P} = \boldsymbol{I} - \varepsilon \boldsymbol{L}$$

where $\boldsymbol{I}$ is the identity matrix and ε $(0 < \varepsilon < 1)$ is some positive number.

2.2 Consensus Control

The dynamics of agent $i\,(= 1, 2, \cdots, n)$ is defined by a discrete-time integral system as follows:

$$x_i[k+1] = x_i[k] + u_i[k]$$

where x_i is the state variable and u_i is the control input of agent i.

Let d be the offset, The control input u_i of the offset-aware consensus control is given by

$$\begin{aligned} u_i[k] &= -\varepsilon \sum_{j=1}^{n} a_{ij}\left(x_i[k] - x_j[k] - d\right) \\ &= -\varepsilon \sum_{j=1}^{n} a_{ij}\left(x_i[k] - x_j[k]\right) + b_i, \end{aligned} \tag{2}$$
$$b_i = \varepsilon d \sum_{j=1}^{n} a_{ij}.$$

In Eq. (2), b_i does not contribute to stability. Considering $b_i = 0$ and summing up the states of all agents, the state equation of consensus is represented by

$$\boldsymbol{x}[k+1] = \boldsymbol{P}\boldsymbol{x}[k]. \tag{3}$$

Since Perron matrix $\boldsymbol{P}$ always have eigenvalues of 1, the relationship is given by

$$\boldsymbol{P}\boldsymbol{1}_n = \boldsymbol{1}_n. \tag{4}$$

In Eqs. (3) and (4), when $k \to \infty$, we get

$$\lim_{k\to\infty} \boldsymbol{x}[k+1] = \frac{1}{n}\boldsymbol{1}_n\boldsymbol{1}_n^{\mathrm{T}}\boldsymbol{x}_0,$$
$$\lim_{k\to\infty} x_i[k+1] = \frac{1}{n}\sum_{j=1}^{n} x_{0j}.$$

Thus, the target system achieves the average consensus in the case that the undirected graph representing the inter-agent networks are connected.

2.3 Polygon-Wall in MPS method

MPS method is a typical fluid dynamics method. In this method, fluid is calculated as a collection of fluid particles.

As governing equations for incompressible fluids, Navier-Stokes equations and continuity equation are given by

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\boldsymbol{v} + \boldsymbol{g}, \tag{5}$$
$$\frac{D\rho}{Dt} = 0$$

where $\boldsymbol{v}$ is the fluid velocity, ρ is the density of the fluid, ν is the kinematic viscosity coefficient and $\boldsymbol{g}$ is the gravitational acceleration. The first term on the right-hand side of Eq. (5) is called the pressure term, the second is the viscosity term, and the third is the external force term. There are two calculation processes for MPS method: explicit and implicit. As an explicit, the method calculates the viscosity and external force terms. From the calculation results, the tentative position and velocity of the particles are determined. As an implicit, the method calculates the pressure term. From the calculation results, the particle information is updated as to its true position and velocity. In the pressure term, collision avoidance is based on the pressure gradient between particles by deriving the pressure value for each particle. In addition, the conventional MPS method represents obstacles as a collection of particles. However, this method has the problem that the obstacle wall boundaries are not smooth. Therefore, the paper [17] proposes a method to represent the boundary using the concept of Polygon-Wall. This subsection focuses on the calculation of the pressure term in Eq. (5).

MPS method uses a weight function as an indicator of the influence of particles on each other. The weight function w which depends on the distance between particles is given by

$$w(r_{ij}) = \begin{cases} \left(1 - \frac{r_{ij}}{r_e}\right)^2 & 0 \leq r_{ij} \leq r_e \\ 0 & r_{ij} > r_e \end{cases} \tag{6}$$

where r_{ij} is the distance between particles i and j, r_e is the effective radius.

In addition, the particle number density n_i is an indicator of the degree to which particles are densely concentrated around particle i. In Eq. (6), we get

$$n_i = \sum_{j\neq i} w\left(r_{ij}\right). \tag{7}$$

Before starting the calculation, we arrange particles in a grid. The particle number density of the particles in the center of the initial arrangement is used as n^0.

In Eq. (5), the discretized model for particle i is given by

$$\frac{D\boldsymbol{v}_i}{Dt} = -\frac{1}{\rho}\langle \nabla P\rangle_i + \nu \left\langle \nabla^2 \boldsymbol{v}\right\rangle_i + \boldsymbol{g}, \tag{8}$$

$$\langle \nabla P\rangle_i = \frac{d}{n^0}\sum_{j\neq i}\left[\frac{P_j - P_i}{r_{ij}^2}\boldsymbol{r}_{ij} w\left(r_{ij}\right)\right]$$

where $\boldsymbol{r}_{ij}$ is the relative vector from particle i to particle j, d is the number of dimensions and P_i, P_j are the pressure values of particles i and j.

Discretizing Eq. (8) by the forward Euler method based on the time axis, we get

$$\boldsymbol{v}_i[k+1] = \boldsymbol{v}_i[k] + \boldsymbol{u}_i^{fluid}[k], \tag{9}$$

$$\boldsymbol{u}_i^{fluid}[k] = \Delta t\left[-\frac{1}{\rho}\langle \nabla P\rangle_i^{k+1} + v\left\langle \nabla^2 \boldsymbol{v}\right\rangle_i^k + \boldsymbol{g}\right]. \tag{10}$$

We describe how to obtain the pressure values in MPS method. To find the pressure values, the Poisson equation is given by

$$-\frac{1}{\rho}\langle \nabla^2 P\rangle_i = \frac{1}{\Delta t^2}\frac{n^* - n^0}{n^0} \tag{11}$$

$$\langle \nabla^2 P\rangle_i = \frac{2d}{\lambda n^0}\sum_{j\neq i}\left(P_j - P_i\right) w\left(r_{ij}^*\right)$$

$$\lambda = \frac{\sum_{j\neq i'} {r_{i'j}^0}^2 w\left(r_{i'j}^0\right)}{\sum_{j\neq i'} w\left(r_{i'j}^0\right)}$$

where n^* is the particle number density with respect to the tentative position.

From Eq. (11), we obtain

$$-\frac{1}{\rho}\frac{2d}{\lambda n^0}\sum_{j\neq i}\left(P_j - P_i\right) w\left(r_{ij}^*\right) = \frac{1}{\Delta t^2}\frac{n^* - n^0}{n^0}. \tag{12}$$

We extract the coefficients for each pressure in Eq. (12), we get

$$a_{i1}^p P_1 + a_{i2}^p P_2 + \cdots + a_{ii}^p P_i + \cdots + a_{iN}^p P_N = b_i,$$

$$a_{ij}^p = \begin{cases} -\frac{1}{\rho}\frac{2d}{\lambda n^0} w\left(r_{ij}^*\right) & (j\neq i) \\ \frac{1}{\rho}\frac{2d}{\lambda n^0}\sum_{j'\neq i} w\left(r_{ij}^*\right) & (j = i)\end{cases},$$

$$b_i = \frac{1}{\Delta t^2}\frac{n^* - n^0}{n^0}.$$

Let $\boldsymbol{A}^p = [a_{ij}^p] \in \mathbb{R}^{N\times N}$, $\boldsymbol{x} = [P_i] \in \mathbb{R}^N$ and $\boldsymbol{b} = [b_i] \in \mathbb{R}^N$, simultaneous linear equations is given by

$$\boldsymbol{A}^p\boldsymbol{x} = \boldsymbol{b}. \tag{13}$$

The pressure values for all particles are obtained by solving Eq. (13) using the conjugate gradient method or other methods.

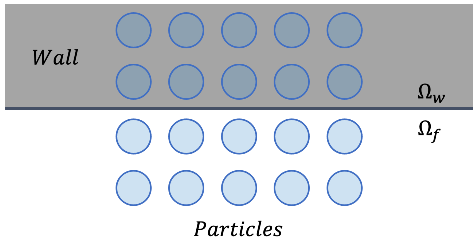


Fig. 1: Classification of regions where particles are present

We explain obstacle avoidance in MPS method. By classifying the region of particle presence into fluid region Ω_f and wall region Ω_w in Fig. 1, Eq. (7) can be rewritten by

$$n_i = \sum_{j\neq i, j\in\Omega_f} w\left(r_{ij}\right) + \sum_{j\neq i, j\in\Omega_w} w\left(r_{ij}\right), \tag{14}$$

$$Z(r_{iw}) = \sum_{j\neq i, j\in\Omega_w} w\left(r_{iw}\right) \tag{15}$$

where r_{iw} is the distance from particle i to the wall. The first term on the right side of Eq. (14) considers collision avoidance between particles and the second term considers collision avoidance with walls. Eq. (15) can be calculated using only the distance from the particle to the wall by pre-computing different values for the distance.

We decompose the pressure term in Eq. (8) in each region, we obtain

$$\langle \nabla P\rangle_i = \langle \nabla P\rangle_{i,\Omega_f} + \langle \nabla P\rangle_{i,\Omega_w},$$

$$\langle \nabla P\rangle_{i,\Omega_f} = \frac{d}{n^0}\frac{P_j - P_i}{r_{ij}^2}\boldsymbol{r}_{ij} w\left(r_{ij}\right), \tag{16}$$

$$\langle \nabla P\rangle_{i,\Omega_w} = \frac{d}{n^0}\frac{P_{wall} - P_i}{r_{iw}^2}\boldsymbol{r}_{iw} Z\left(r_{iw}\right) \tag{17}$$

where P_{wall} is the pressure from the wall and $\boldsymbol{r}_{iw}$ is the vertical vector from the wall to particle i.

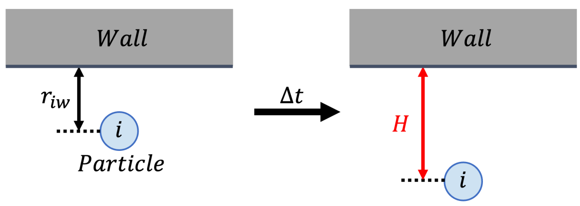


Fig. 2: Calculating the pressure from the wall to particle i

The following is a method for calculating P_{wall} in Eq. (17). In Fig. 2, particle i moves a certain distance

H during time $\varDelta t$ due to the influence from the wall. In this case, we obtain

$$\begin{aligned} d\boldsymbol{r}_i &= -\frac{\varDelta t^2}{\rho}\langle \nabla P\rangle_{i,\Omega_w} \\ &= -\frac{\varDelta t^2}{\rho}\frac{d}{n^0}\frac{P_{wall}-P_i}{r_{iw}^2}\boldsymbol{r}_{iw}Z\left(r_{iw}\right) \\ &= \frac{\boldsymbol{r}_{iw}}{r_{iw}}\left(H-r_{iw}\right). \end{aligned}$$

Solving for P_{wall}, we get

$$P_{wall} = P_i - \frac{\rho}{\varDelta t^2}\frac{n^0}{d}\frac{r_{iw}}{Z\left(r_{iw}\right)}\left(H-r_{iw}\right). \tag{18}$$

We substitute Eq. (18) into Eq. (17), we obtain

$$\langle \nabla P\rangle_{i,\Omega_w} = -\frac{\rho}{\varDelta t^2}\frac{\boldsymbol{r}_{iw}}{r_{iw}}\left(H-r_{iw}\right). \tag{19}$$

Eq. (19) represents the repulsive force exerted on the particle by the Polygon-Wall.

3 Problem Setting

This section sets up the situation shown in Fig. 3. We consider a diamond-shaped obstacle as an angular obstacle on a two-dimensional plane. Swarm robots in column formation move straight ahead in the positive direction of the x-axis in such an environment. Column formation means that $N(= m \times n)$ robots are aligned vertically in m rows and horizontally in n rows. Each robot has sensors that acquire information about its surroundings and measure the distance from the robot to wall obstacles.

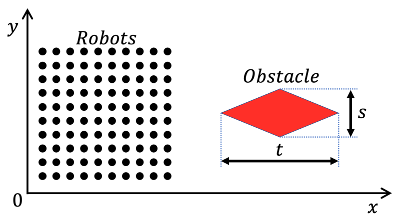


Fig. 3: Problem Environment

Let the set of robots be $V = \{1, 2, \cdots, N\}$, the dynamics of robot $i \in V$ at discrete-time integral systems $k \in \mathbb{N}$ is given by

$$\boldsymbol{p}_i[k+1] = \boldsymbol{p}_i[k] + \boldsymbol{u}_i[k], \tag{20}$$

$$\boldsymbol{p}_i[k] = \left[\, p_{ix}[k]\; p_{iy}[k] \,\right]^{\mathrm{T}},$$

$$\boldsymbol{u}_i[k] = \boldsymbol{u}_i^f[k] + \boldsymbol{u}_i^g[k] + \boldsymbol{u}_i^c[k] + \boldsymbol{u}_i^o[k] \tag{21}$$

where $p_{ix}[k], p_{iy}[k]$ are x, y coordinates of robot i, $\boldsymbol{u}_i^f[k] \in \mathbb{R}^2$ is the control low for staying formation, $\boldsymbol{u}_i^g[k] \in \mathbb{R}^2$ is the control low for reaching the target, $\boldsymbol{u}_i^c[k] \in \mathbb{R}^2$ is the control low for avoiding collision between robots and $\boldsymbol{u}_i^o[k] \in \mathbb{R}^2$ is the control low for avoiding obstacle.

In Eqs. (9), (10), (20) and (21), there is the analogy between swarm robots and fluid dynamics. The inputs $\boldsymbol{u}_i^c[k]$ and $\boldsymbol{u}_i^o[k]$ of (21) correspond to the pressure term of (10), which relizes collision avoidance with particles and obstacles based on the pressure gradient. The input $\boldsymbol{u}_i^f[k]$ of (21) corresponds to the viscosity term of (10), which adjusts velocities between particles. The position tracking input $\boldsymbol{u}_i^g[k]$ of (21) corresponds to the external force term of (10), which is the effects of gravity. On the other hand, there is a gap between the centralized pressure calculation of particles in (13) and decentralized collision avoidance within robots in $\boldsymbol{u}_i^c[k]$. The academic interest in this study is a new collision avoidance method using this gap.

In Eqs. (20) and (21), the states of all robots vertically are given by

$$\boldsymbol{p}[k+1] = \boldsymbol{p}[k] + \boldsymbol{u}^f[k] + \boldsymbol{u}^g[k] + \boldsymbol{u}^c[k] + \boldsymbol{u}^o[k].$$

3.1 Formation Control

Information transfer path of swarm robots is undirected and connected graph as shown Fig. 4. For the direction of movement, let $\mathcal{N}_i = \{R_i, L_i, F_i, B_i\}$ be the set of robots located in each direction. Also, the adjacency matrix $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ representing the connection state to the robot included in $\mathcal{N}_i$ is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if} \quad p_{ij} \leq S \\ 0 & \text{otherwise} \end{cases}, \quad j \in \mathcal{N}_i$$

where p_{ij} is the distance between robot i and robot j, $S(>0)$ is the sensor's range of a robot. The reason for introducing parameter S is to compare this paper's method with the formation control method proposed in the previous study [8]. The fixed graph that does not depend on S is also acceptable.

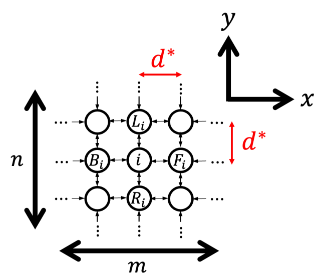


Fig. 4: Network structure between robots

Each robot maintains the desired distance d^* from its neighbors. A set of vectors representing the desired distances for the robots included in $\mathcal{N}_i$ is defined as follows:

$$\begin{aligned} \boldsymbol{d}_{R_i} &= \begin{bmatrix} 0 \\ d^* \end{bmatrix}, \quad \boldsymbol{d}_{L_i} = \begin{bmatrix} 0 \\ -d^* \end{bmatrix}, \\ \boldsymbol{d}_{F_i} &= \begin{bmatrix} -d^* \\ 0 \end{bmatrix}, \quad \boldsymbol{d}_{B_i} = \begin{bmatrix} d^* \\ 0 \end{bmatrix}. \end{aligned} \tag{22}$$

The following equation must be satisfied for swarm robots to maintain column formation.

$$\lim_{k\to\infty} \|\hat{\boldsymbol{p}}_i[k] - \hat{\boldsymbol{p}}_j[k]\| = 0, \quad j \in \mathcal{N}_i \tag{23}$$

where $\hat{\boldsymbol{p}}_i[k] \; (= \boldsymbol{p}_i[k] - \boldsymbol{d}_j/2)$, $\hat{\boldsymbol{p}}_j[k] \; (= \boldsymbol{p}_j[k] + \boldsymbol{d}_j/2)$ are the target relative vectors.

According to Section 2.2, Eqs. (22) and (23), the control law for staying the formation is given by

$$\boldsymbol{u}_i^f[k] = -\varepsilon \sum_{j\in\mathcal{N}_i} a_{ij}[k] \left(\boldsymbol{p}_i[k] - \boldsymbol{p}_j[k] - \boldsymbol{d}_j\right) \tag{24}$$

where $\boldsymbol{d}_j$ ($j \in \{R_i, L_i, F_i, B_i\}$) is given by (22) [20,21].

Also, Eq. (24) is proportional to the magnitude of the displacement between the distance between the robots and the desired distance. In other words, the displacement changes the attraction of the formation control. In reference [8], to mitigate this change in input $\boldsymbol{u}_i^f[k]$, the adjacency matrix $\boldsymbol{A}[k] = [a_{ij}[k]] \in \mathbb{R}^{N\times N}$ is given by

$$a_{ij}[k] = \begin{cases} \exp\left(-\left|p_{ij}[k] - d\right|\right) & \text{if} \quad p_{ij}[k] \le S \\ 0 & \text{otherwise} \end{cases}, \quad j \in \mathcal{N}_i.$$

3.2 Reaching the Target Position

This paper uses the potential method to get the robot to the target position. In this case, gravitational potential function $V_i^g(\boldsymbol{r})$ and control low $\boldsymbol{u}_i^g[k]$ for robot i is given by

$$\begin{aligned} V_i^g(\boldsymbol{r}) &= k_g \left\|x - x_g^i\right\|^2 \\ \boldsymbol{u}_i^g[k] &= -\alpha \frac{\partial V_i^g(\boldsymbol{r})}{\partial \boldsymbol{r}}\bigg|_{\boldsymbol{p}_i[k]}^{\mathrm{T}} \end{aligned}$$

where $\boldsymbol{r} = [x\ y]^{\mathrm{T}}$, $k_g \in \mathbb{R}_+$ is the adjustment parameter, x_g^i is the target x-coordinate and α is the step size. Let x_g be the criterion of the target positions, the target x-coordinate of the n' th from the first column is $x_g^i = x_g - (n'-1)d^*$ [8].

4 Main Result

This section describes the solutions to issues addressed in this study. The first is a design method for obstacles with angular shape, and the second is to incorporate fluid dynamics methods for distributed control. For the first issue, the concept of Polygon-Wall in MPS method described in Section 2.3 is applied to the control of swarm robots. The second issue is addressed by substituting the centralized computation process of MPS method in other way.

4.1 Obstacle Avoidance

This subsection describes two design methods of diamond-shaped obstacle. Section 4.1.1 is an approximate design method using the time-varying potential function of [8]. Section 4.1.2 describes the Polygon-Wall based method proposed in this paper. Also, this subsection compares the parameter properties of the two design methods.

4.1.1 Conventional Method

In reference [8], obstacle avoidance is performed by assigning weights to exponential potential function. Relative angle θ_i^{go} between the direction of movement and the obstacle with respect to robot i is used to design the weight. Thus, θ_i^{go} needs the obstacle's position. Using this method, a diamond-shaped obstacle is represented by varying the major and minor axes with respect to an ellipse. Referring to design methods in reference [8], time-varying potential function $V_i^o(\boldsymbol{r})$ and control law $\boldsymbol{u}_i^o[k]$ are given by

$$\begin{aligned} V_i^o(\boldsymbol{r}) &= k_o w(\theta_i^{go}) \\ &\quad \exp\left[-\left\{\left(\frac{x - p_{ox}}{\sigma_o + \sigma}\right)^2 + \left(\frac{y - p_{oy}}{\sigma_o - \sigma}\right)^2\right\}\right], \\ \sigma &= \frac{k_\sigma R}{r}\left|\theta_s - \theta_i^{go}\right|, \\ w(\theta_i^{go}) &= \exp\left(k_\theta \left|\theta_s - \theta_i^{go}\right|\right), \\ \boldsymbol{u}_i^o[k] &= -\alpha \frac{\partial V_i^o(\boldsymbol{r})}{\partial \boldsymbol{r}}\bigg|_{\boldsymbol{p}_i[k]}^{\mathrm{T}} \end{aligned} \tag{25}$$

where $k_o, \sigma_o, k_\sigma, k_\theta \in \mathbb{R}_+$ are the adjustment parameters, $\boldsymbol{p}_o = [p_{ox}\ p_{oy}]^{\mathrm{T}}$ is the position of obstacle and r, R are the sizes of the robot and obstacle, respectively. In Eq. (25), the obstacle size R determines the magnitude of σ. Peculiar shaped obstacles make it difficult to measure the position $\boldsymbol{p}_o$ and the size R of the obstacle.

4.1.2 Proposed Method

In paper study, we introduce the concept of Polygon-Wall in fluid dynamics to swarm robots as an avoidance method for angular obstacles. Using the theory in Section 2.3, we can obstacles design more easily than with traditional potential function. We use Eq. (19) that represents the repulsive force against the Polygon-Wall.

Repulsive force $\langle \nabla P \rangle_{i,\Omega_w}$ in Eq. (19) allows us to design even diamond-shaped obstacle and to determine the strength of the weights independently of the size of the obstacle. In Fig. 5, control law $\boldsymbol{u}_i^o[k]$ for obstacle avoidance using the concept of Polygon-Wall is given by

$$\boldsymbol{u}_i^o[k] = \begin{cases} -k_w \frac{\boldsymbol{p}_{iw}[k]}{p_{iw}[k]} (H - p_{iw}[k]) & \text{if } p_{iw}[k] < H \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

where $k_w, H \in \mathbb{R}_+$ are the adjustment parameters, $\boldsymbol{p}_{iw}[k]$ is the vertical vector from the wall to robot i and $p_{iw}[k]$ is the distance between robot i and the wall. $\frac{\rho}{\Delta t^2}$, $\boldsymbol{r}_{iw}$ and r_{iw} of (19) are replaced by k_w, $\boldsymbol{p}_{iw}[k]$, and $p_{iw}[k]$, respectively. This proposed method requires a sensor acquiring p_{iw} even if the obstacles are peculiar shaped.

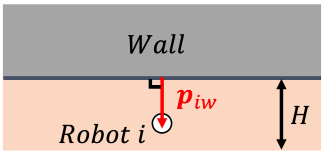


Fig. 5: Generation of repulsion by Polygon-Wall

4.2 Collision Avoidance between Robots

This subsection describes collision avoidance between robots. In this paper, the fluid dynamics method is introduced for swarm robots. In the existing studies [15, 16], MPS method is applied directly to the control of swarm robots. However, as explained in Section 2.3, MPS method is a centralized control that aggregates information from all robots in the process of pressure calculation. By substituting other methods for the process of pressure calculation, distributed control may be maintained when MPS method is applied to control of swarm robots. Since the pressure gradient between the particles prevents the particles from colliding with each other, this paper replaces the computation of pressure gradients with the computation of potential functions.

Therefore, in this paper, the conventional potential function is used for collision avoidance between robots. Repulsive potential function $V_i^c(\boldsymbol{r})$ and control law $\boldsymbol{u}_i^c[k]$ for collision avoidance between robots are

$$V_i^c(\boldsymbol{r}) = \sum_{j \in \widetilde{\mathcal{N}}_i[k]} k_c e^{-\frac{\|\boldsymbol{r}-\boldsymbol{p}_j\|^2}{\sigma_c{}^2}},$$

$$\boldsymbol{u}_i^c[k] = -\alpha \frac{\partial V_i^c(\boldsymbol{r})}{\partial \boldsymbol{r}}\bigg|_{\boldsymbol{p}_i[k]}^{\mathrm{T}}$$

where $k_c, \sigma_c \in \mathbb{R}_+$ are the adjustment parameters [22].

5 Numerical Experiment

In paper study, MATLAB is used as the analysis software for simulation. Two types of control laws $\boldsymbol{u}_i^o[k]$ (Conventional:(25), Proposed:(26)) for obstacle avoidance are used and compared in the problem setup in Section 3. In addition, this paper conducts on the case with three differently shaped obstacles to demonstrate the effectiveness of a proposed method (26).

5.1 One Diamond-shaped Obstacle

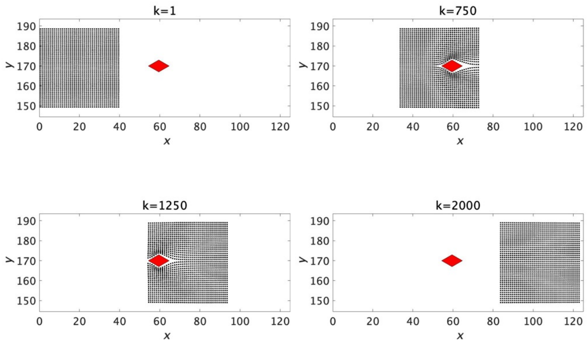
The parameters used in each numerical experiment are shown in Table 1. Parameter values are determined by trial and error. Fig. 6 shows snapshots of swarm robots in each step. The trajectory of swarm robots is shown in the upper parts of Fig. 7 and it means the solid black line connecting the plot points at each step of the robot. Since the main topic of this paper is collision avoidance, we calculate relative distances to obstacles and other robots. Focusing on a specific robot (ID: 59), the lower parts of Fig. 7 shows the distances between it and its neighbors $\mathcal{N}_{59} = \{19, 58, 60, 99\}$, and between it and the obstacle wall.

Table 1: Parameters

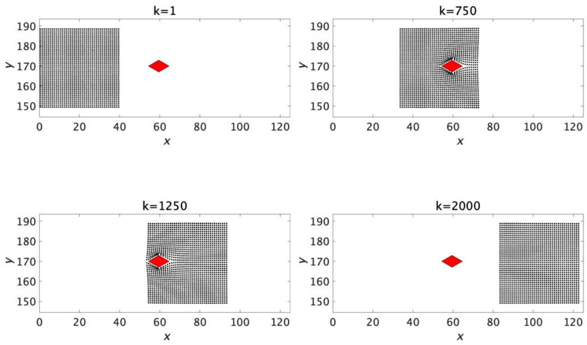
parameter	value
N, m, n	$1600, 40, 40$
d^*	1
ϵ, α, S	$0.2, 0.005, 50$
s, t	$6, 6\sqrt{3}$
x_g	500
$\boldsymbol{p}_o$	$[59.5\ 170]^{\mathrm{T}}$
k_g, k_c, σ_c	$0.01, 1, 0.5$
$k_o, \sigma_o, k_\sigma, k_\theta$	$800, 2, 0.025, 0.001$
r, R	$0.1, (s/2 + t/2)/2$
k_w, H	$0.4, 1$

Figs. 6 and 7 show that the obstacle avoidance method using Polygon-Wall proposed in this paper avoids obstacles in the same way as the conventional potential method. Compared to the conventional method [8], the proposed method requires fewer parameters. Also, no information on the relative angle and the size to the obstacle is required and the swarm robots can avoid even diamond-shaped obstacles. In addition, we can apply the Polygon-Wall-based obstacle avoidance method to swarm robots in a distributed manner.

Fig. 7 shows the trajectory of Robot 59 (solid blue line) and the relative distances to $\mathcal{N}_{59}$ and the obstacle. Since the radius of the circular robot is $r = 0.1$ in



(a) Conventional [8]



(b) Proposed

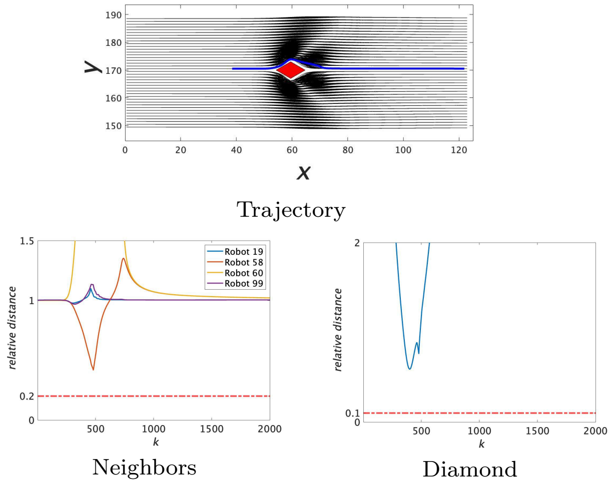
Fig. 6: Snapshots of swarm robots at each step

Table 1, the lower parts of Fig. 7b show that the proposed method (26) does not collide with other robots and obstacles.

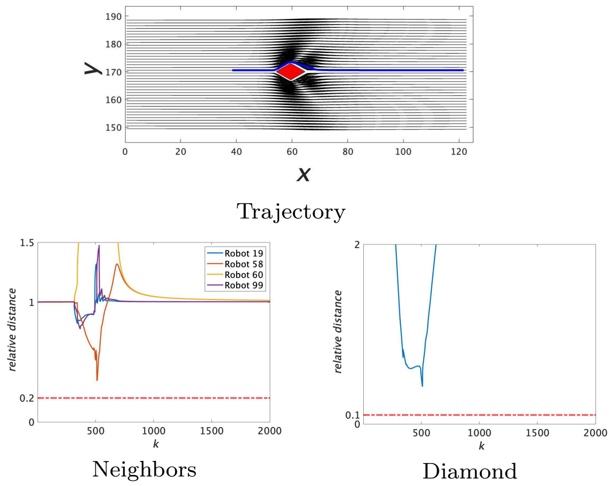
These numerical experiments with one diamond-shaped obstacle show that the proposed method (26) can achieve obstacle avoidance and parameter reduction comparable to the conventional method (25) without collisions.

5.2 Three Differently-Shaped Obstacles

This paper conducts an additional numerical experiment to verify that the proposed method (26) avoids swarm robots along the walls of the obstacles. Fig. 8 shows the case with three differently-shaped obstacles. For diamond, circular and isosceles right triangle-shaped obstacles, the positions are $\boldsymbol{p}_{o1}, \boldsymbol{p}_{o2}, \boldsymbol{p}_{o3}$, and the sizes are R_1, R_2, R_3, respectively. Table 2 shows the parameters, the upper parts of Figs. 9 and 10 show the trajectories of swarm robots. Also, we calculate the relative distances as Section 5.1. Focusing on specific robots



(a) Conventional [8]



(b) Proposed

Fig. 7: Trajectory and Relative Distance (ID:59)

(ID: 49, 73), the lower parts of Figs. 9 and 10 show the distances between them and their neighbors $\mathcal{N}_{49} = \{9, 48, 50, 89\}, \mathcal{N}_{73} = \{33, 72, 74, 113\}$, and between them and the obstacle wall.

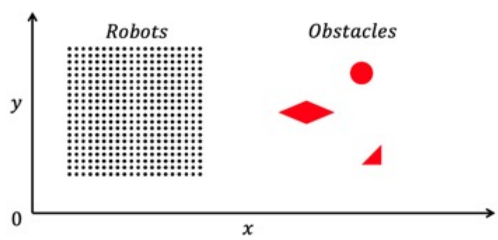
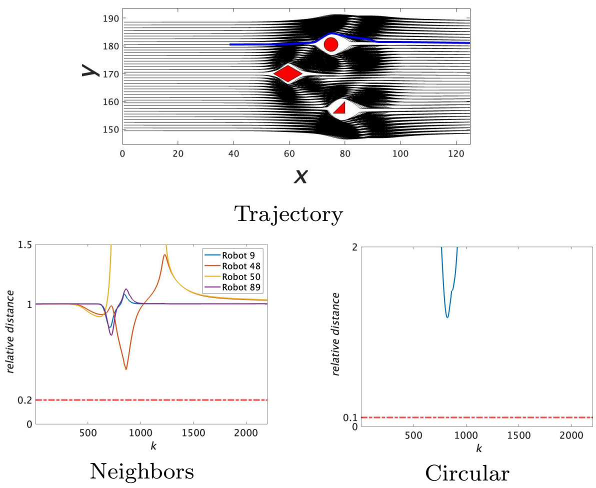


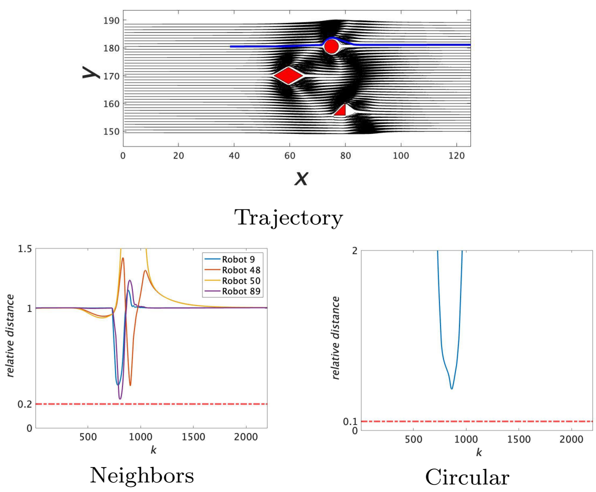
Fig. 8: Problem Environment

Table 2: Parameters

parameter	value
N, m, n	$1600, 40, 40$
d^*	1
ϵ, α, S	$0.2, 0.005, 50$
x_g	500
$\boldsymbol{p}_{o1}, \boldsymbol{p}_{o2}, \boldsymbol{p}_{o3}$	$[59.5\ 170]^{\mathrm{T}}, [75\ 180.5]^{\mathrm{T}}, [78.6\ 157.2]^{\mathrm{T}}$
k_g, k_c, σ_c	$0.01, 1, 0.5$
$k_o, \sigma_o, k_\sigma, k_\theta$	$800, 2, 0.025, 0.001$
r, R_1, R_2, R_3	$0.1, 4.1, 2.5, 1.4$
k_w, H	$0.4, 1$

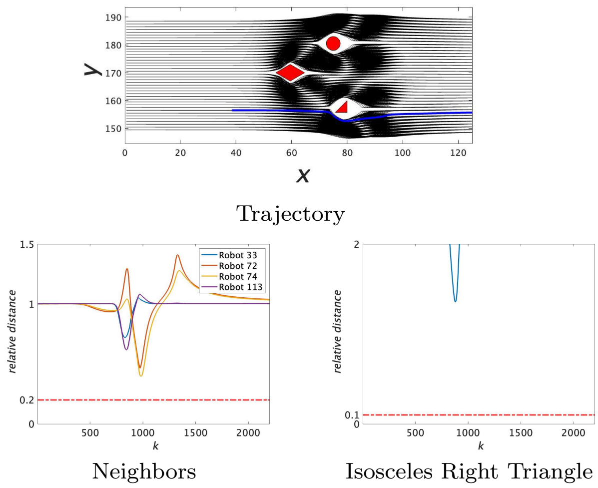


(a) Conventional [8]

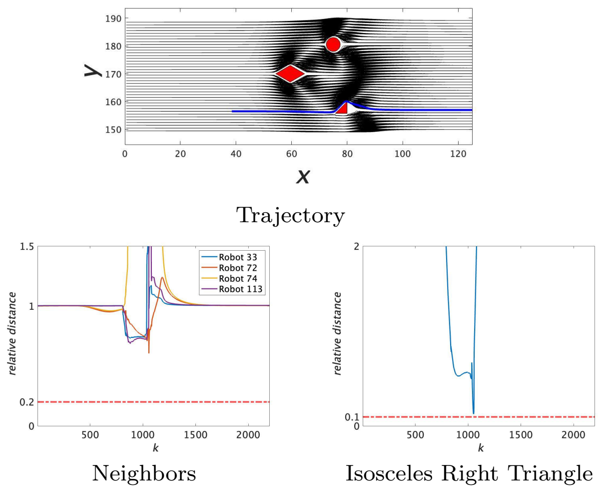


(b) Proposed

Fig. 9: Trajectory and Relative Distance (ID:49)



(a) Conventional [8]



(b) Proposed

Fig. 10: Trajectory and Relative Distance (ID:73)

The upper parts of Figs. 9 and 10 show that the trajectories of Robot 49, 73 (solid blue line). These show that the conventional method (25) avoids obstacles with a large turn, but the proposed method (26) avoids them along the walls of the obstacles. The lower parts of Figs. 9, 10 show the relative distances to $\mathcal{N}_{49}, \mathcal{N}_{73}$ and the obstacle. The lower parts of Figs. 9b and 10b show that the proposed method (26) does not collide with other robots and obstacles.

These numerical experiments with three differently-shaped obstacle show that the proposed method (26) can reduce the parameters while maintaining the avoidance performance comparable to the conventional method (25). Also, it is possible to realize avoidance with less detours along the shape of obstacles than the conventional method (25).

From the above results, Polygon-Wall is an effective solution to the research question of this paper. Meanwhile, replacing the pressure calculations by the calculations of potential functions remains a challenge. To adequately reflect fluid characteristic in swarm robots, we need to solve Eq. (13) in a distributed manner. This is a work for future research.

6 Conclusion

This paper has applied the concept used in fluid dynamics to swarm robots. The obstacle of Polygon-Wall allows angular obstacles to be designed more easily than with traditional potential methods. The partial incorporation of a fluid dynamics method into the control of swarm robots also helps to maintain distribution.Our future work is to calculate pressures in a distributed manner to better reflect fluid characteristics.

References

1. Qirong Tang, Fangchao Yu, Zhipeng Xu and Peter Eberhard, Swarm Robots Search for Multiple Targets, IEEE Access, Vol.8, pp.92814-92826 (2020)
2. Muhammad S.A Khan, Tarem Ahmed and Mohammad Faisal Uddin, Multi-robot Search Algorithm using Timed Random Switching of Exploration Approaches, IEEE Region 10 Symposium (TENSYMP), pp.868-871 (2020)
3. Matin Macktoobian, Denis Gillet and Jean-Paul Kneib, Astrobotics: Swarm Robotics for Astrophysical Studies, IEEE Robotics & Automation Magazine, Vol.28, Issue.3, pp.92-101 (2021)
4. Shinsaku Izumi, Yuto Shiomoto and Xin Xin, Mass Game Simulator: An Entertainment Application of Multiagent Control, IEEE Access, Vol.9, pp.4129-4140 (2020)
5. Song Ping, Li Kejie, Han Xiaobing and Qi Guangping, Formation and Obstacle-Avoidance Control for Mobile Swarm Robots Based on Artificial Potential field, IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2273-2277 (2009)
6. Shun Yang, Tieshan Li, Quan Shi, Weiwei Bai and Yue Wu, Artificial Potential-Based formation Control with Collision and Obstacle Avoidance for Second-order Multi-Agent Systems, International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), pp. 58-63 (2020)
7. Xug Lanejin and Wenbiao Xu, The null-space-based behavioral control for a swarm of robots tracking a target region in obstacle environments, Chinese Control Conference (CCC), pp. 5574-5578 (2019)
8. Shotaro Shibahara, Takuma Wakasa and Kenji Sawada, Network weight and time-varying potential function for obstacle avoidance of swarm robots in column formation, SICE Journal of Control, Measurement, and System Integration, Vol.15, Issue.1, pp. 24-35 (2022)
9. Jeonggeun Lim, Sangjin Pyo, Namyun Kim, Jehong Lee and Jongho Lee, Obstacle Magnification for 2-D Collision and Occlusion Avoidance of Autonomous Multirotor Aerial Vehicles, IEEE/ASME Transactions on Mechatronics, Vol.25, Issue.5, pp.2428-2436 (2020)
10. Vishwamithra Sunkara and Animesh Chakravarthy, Collision Avoidance Laws for Objects with Arbitrary Shapes, IEEE 55th Conference on Decision and Control (CDC), pp.5158-5164 (2016)
11. Xiao Liang, Honglun Wang and Haitao Luo, Collaborative Pursuit-Evasion Strategy of UAV/UGV Heterogeneous System in Complex Three-Dimensional Polygonal Environment, Complexity, Vol.2020, pp.1-13 (2020)
12. Tomasz Gawron and Maciej Marcin Michalek, VFO feedback control using positively-invariant funnels for mobile robots travelling in polygonal worlds with bounded curvature of motion, IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp.124-129 (2017)
13. Sriram Ramaswamy, The Mechanics and Statistics of Active Matter, Annual Review of Condensed Matter Physics, Vol.1 pp.323-345 (2010)
14. Alexey Dmitriev *et al*, Statistical Correlations in Active Matter Based on Robotic Swarms, International Conference Engineering and Telecommunication (En&T), pp.1-3 (2021)
15. Joseph Aldrin Chua *et al*, Moving Particle Semi-Implicit Method for Control of Swarm Robotic Systems, IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), pp. 1-5 (2019)
16. Joseph Aldrin Chua *et al*, Swarm Collision Avoidance using Moving Particle Semi-Implicit Method, IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), pp. 1-6 (2021)
17. Naoto Mitsume, Shinobu Yoshimura, Kohei Murotani and Tomonori Yamada, Explicitly represented polygon wall boundary model for the explicit MPS method, Comp. Part. Mech, Vol.2, pp.73-89 (2015)
18. Masahiro Kondo and Seiichi Koshizuka, Improvement of stability in moving particle semi-implicit method, International Journal for Numerical Methods in Fluids, Vol.65, Issue.6, pp. 638-654 (2011)
19. Angelia Nedić and Asuman Ozdaglar, Cooperative distributed multi-agent optimization, In Convex Optimization in Signal Processing and Communications, pp.340-386 (2009)
20. Yuan Zhou, Yongfang Liu and Yu Zhao, Prescribed-time Bipartite Consensus Formation Control for General Linear Multi-agent Systems, IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, pp.3562-3567 (2020)
21. Suleman Khan, Irshad Hussain and Muhammad Irfan Khattak, Consensus Based Formation Control of Multiple UAVs, jictra, Vol.11, No.1 pp.31-37 (2020)
22. Dong Hun Kim, Hyun-Woo Lee, Seiichi Shin and T.Suzuki, Local Path Planning Based on New Repulsive Potential Function with Angle Distributions, International Conference on Information Technology and Applications (ICITA'05) (2005)