Network weight and time-varying potential function for obstacle avoidance of swarm robots in column formation

Shotaro Shibahara, Takuma Wakasa and Kenji Sawada
The University of Electro-Communications, Tokyo, Japan

ABSTRACT
This paper proposes network weight functions and time-varying potential functions for obstacle avoidance of swarm robots in column formation. We consider a potential function whose exponent is the distance between the robots and the obstacle for obstacle avoidance. When a robot tries to avoid an obstacle while staying in a formation, it behaves oscillatory due to the imbalance between the attraction of staying in the formation and the repulsion of avoiding the obstacle. In addition, swarm robots may become stagnant in front of large obstacles. To solve these problems, we propose two methods. The first method is to weaken the strength of the connection between robots according to their relative distance. The second method is to change the shape of the potential function depending on the relative angle between the robot and the obstacle. Numerical examples show that these methods reduce the oscillatory of the robot and alleviate the stagnation in front of the obstacle.

ARTICLE HISTORY
Received 26 October 2021
Revised 12 January 2022
Accepted 15 January 2022

KEYWORDS
Multi-agent system; network structure; formation control; obstacle avoidance; potential function

1. Introduction
A multi-agent system (MAS) allows multiple agents to share information with each other, so that the agents as a whole can efficiently accomplish tasks that would be difficult to accomplish independently [1, 2]. There has been a growing motivation to control large swarm robots of more than several thousand units [3, 4]. One of the important problems is obstacle avoidance while swarm robots stay in the desired formation [5]. Typical methods of obstacle avoidance include potential functions [6, 7] and model predictive control [8]. Potential function methods realize the collision avoidance by using the repulsion force that increases with the distance between the robot and the obstacle. In this case, the robot near the obstacle may show oscillatory behaviour. This is caused by the fact that the control inputs for staying formation and avoiding obstacles in opposite directions. In the previous study of the current authors [9], the imbalance between the two control inputs is alleviated by lowering the priority of the formation control. However, the number of robots assumed in [9] is much smaller than that in [1, 2]. This paper considers the case where the number of robots is significantly increased from the number set in [9].

In this paper, we propose a method of changing the potential function for obstacle avoidance in addition to the formation control for the column formation problem [10]. The objective of this paper is to reduce the oscillatory behaviour near obstacles and to recover and preserve the formation after the avoidance of obstacles.

Two issues are raised in this paper. The first issue is to alleviate the imbalance between the attractive force for staying the formation and the repulsive force for avoiding obstacles. The second issue is to eliminate the stagnation of the swarm robots in front of the large obstacle.

To address the first issue, this paper proposes a network weight function that changes the gravitational pull for staying the formation between robots based on the positions of the robots and obstacles. Swarm robots in formation are forced to break the desired formation to avoid obstacles. The larger the difference between the actual distance between robots and the desired distance, the larger the gravitational pull of the formation control. In this case, the robot near the obstacle may show oscillatory behaviour. This is caused by the fact that the control inputs for staying formation and avoiding obstacles in opposite directions. In the previous study of the current authors [9], the imbalance between the two control inputs is alleviated by lowering the priority of the formation control. However, the number of robots assumed in [9] is much smaller than that in [1, 2]. This paper considers the case where the number of robots is significantly increased from the number set in [9].

In this paper, we propose a method of changing the potential function for obstacle avoidance in addition to the formation control for the column formation problem [10]. The objective of this paper is to reduce the oscillatory behaviour near obstacles and to recover and preserve the formation after the avoidance of obstacles.

To address the second issue, we set a relative angle based on the positional relationship between the robot
and the obstacle and introduce a weight parameter to change the repulsion force of obstacle avoidance. The stagnation in front of the large obstacle causes a misalignment in the positional relationship between the robots and may make it impossible to stay in the desired formation after avoiding obstacles. This reason is thought to be that the potential function for obstacle avoidance is a circularly spreading function. Reference [11] adds weights depending on the relative angle between the robot and the obstacle to the circularly spreading potential function. The resultant range of the influence of the repulsion force changes from circularly spreading one to teardrop one and smooth obstacle avoidance is achieved. Motivated by this method, we propose a new weight that changes the shape of the potential function depending on the relative angle and construct a time-varying repulsive potential function. By changing the shape of the potential function to an ellipse instead of a circle, swarm robots do not stagnate in front of the large obstacle and can stay in the desired formation after passing through.

The conference version of this paper [9] only considers the design of the network weight function against the first issue. The function of this paper is a generalized form of [9]. Also, the second issue is not considered in [9]. The time varying repulsive force for swarm robots is the contribution of this paper compared with [11]. The number of robots to which the proposed method can be applied is comparable to those in [3, 4]. This paper suppresses the oscillatory behaviour of a large number of robots, which has not been solved in [9].

The organization of this paper is as follows. Section 2 explains consensus control of MAS. Section 3 describes the problem setting of the obstacle avoidance for swarm robots in column formation. Section 4 describes two proposed methods: a method for setting the network weight function and a method for designing a time-varying potential function. In Section 5, numerical experiments are conducted using MATLAB, and the summary of this paper is given in Section 6.

2. Preliminaries

This section explains the distributed optimization of MAS [12–15] which is the basis of the proposed method.

2.1. Graph theory

The network structure for information transfer between agents is represented by graph $G = (V, E)$ composed of the vertex set $V = \{1, 2, \ldots, n\}$ and the edge set $E \subseteq V \times V$. The edge $(i, j) \in E$ indicates an information transfer path from agent $i$ to agent $j$, that is, agent $i \in V$ is adjacent to agent $j \in V$. If all the paths are bi-directional, then $G$ is called an undirected graph. An adjacent matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ is defined as follows:

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

The neighbourhood of agent $i$ is the set of agent $j(\neq i)$ to which agent $i$ is adjacent. The number of edges entering vertex $i$ is called the input degree $d_i^\text{in}$ and given by

$$
d_i^\text{in} = \sum_{j=1}^{n} a_{ij}, \quad i = 1, 2, \ldots, n.
$$

The degree matrix $D$ of graph $G$ with the diagonal element of $d_1^\text{in}, d_2^\text{in}, \ldots, d_n^\text{in}$ is defined as follows:

$$
D = \text{diag}(d_1^\text{in}, d_2^\text{in}, \ldots, d_n^\text{in}).
$$

From $D$ and $A$, the Graph Laplacian is given by

$$
L = D - A.
$$

Also, Perron matrix with some positive number $\epsilon(0 < \epsilon < 1)$ is defined as follows:

$$
P = I - \epsilon L
$$

where $I$ is the identity matrix.

A weighted graph is a graph in which each edge has a positive real number (weight). The weight graph is represented by $(G, w)$ where $\omega: E \rightarrow \mathbb{R}_+ \setminus \{0\}$ is a weight function. Also, the adjacency matrix $A$ of the weighted graph $(G, w)$ is defined as follows:

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

2.2. Consensus control

Consider the case that agent $i(=1, 2, \ldots, n)$ is given by an integral system in discrete-time domain as follows:

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

The control input of agent $i$ considering the offset $d$ is given by

$$u_i[k] = -\epsilon \sum_{j=1}^{n} a_{ij} (x_j[k] - x_i[k] - d). \quad (2)
$$

We transform Equation (2) as follows:

$$u_i[k] = -\epsilon \sum_{j=1}^{n} a_{ij} (x_i[k] - x_j[k] - d)
$$

$$= -\epsilon \sum_{j=1}^{n} a_{ij} (x_i[k] - x_j[k]) + b_i \quad (3)
$$

where $b_i = \epsilon d \sum_{j=1}^{n} a_{ij}$. Equation (3) is a consensus control with bias, so $b_i$ has no effect on the stability analysis [12]. In Equation (3), we set $b_i = 0$. Summarizing
the agents and the inputs, the state-space equation of consensus control is represented by

$$x[k + 1] = Px[k]. \quad (4)$$

Since the Perron matrix $P$ has eigenvalue 1, we get

$$P1_n = 1_n.$$  

Let $k \to \infty$ in Equation (4), we get

$$\lim_{k \to \infty} x[k + 1] = \frac{1}{n} 1_n^T x_0,$$

$$\lim_{k \to \infty} x_i[k + 1] = \frac{1}{n} \sum_{j=1}^{n} x_j.$$  

This system proves that all agents achieve the average consensus.

### 2.3. Distributed optimization

Consider the unconstrained optimization problem:

$$\min_{\xi \in \mathbb{R}^N} \sum_{i \in V} J_i(\xi) \quad (5)$$

where the cost function $J_i(\xi)$ is a real-valued function defined in $\mathbb{R}^N$ and convex. The sub-gradient of differentiable $J_i(\xi)$ is as follows:

$$d_i(\xi) = \frac{\partial J_i}{\partial \xi}.$$

By using the consensus control of Section 2.2, the system equation for solving Equation (5) is defined as follows:

$$x_i[k + 1] = \sum_{j \in V} p_{ij} x_j[k] - s[k] d_i(x_i[k])$$

where $p_{ij}$ is an element of the Perron matrix $P$.

In this paper, we assume that the step width is a constant $s[k] = s(> 0)$. For any $i \in V$, the following equation holds:

$$\lim \sup_{k \to \infty} J(\hat{x}_i[k]) \leq \bar{J} + \frac{s C_1^2 C_2}{2},$$

$$C_1 = \max_{\xi} ||d_i(\xi)||, \quad C_2 = n \left\{ 1 + 8 \left( 2 + \frac{n \beta}{1 - \beta} \right) \right\},$$

$$\beta = 1 - \frac{\eta}{4n^2},$$

$$\hat{x}_i[k] = \frac{1}{k - 1} \sum_{h=0}^{k-1} x_i[h] \quad (k = 2, 3, \ldots)$$

where $\eta$ is the minimum value of non-zero elements of the Perron matrix $P$[14, 15]. Summarizing all agents, the overall system is given by

$$x[k + 1] = Px[k] - s[k] d_i(x[k]). \quad (6)$$

### 3. Problem setting

This paper considers a column formation problem in which $N(= m \times n)$ swarm robots move straight ahead and avoid collisions on the $x$-$y$ plane with $c$ obstacles. The network structure between the robots is an undirected and connected graph as shown Figure 1. The shapes of each robot and obstacle are circulars of radius $r$ and $R$, respectively, and their position coordinates are their centres.

Let $\mathcal{N}_i$ be the neighbourhood set of robot $i$. When the direction of the swarm robots is the positive direction of the $x$-axis, the other robots networked to the right, left, front and back of robot $i$ are defined as $R_i, L_i, F_i, B_i \in \mathcal{N}_i$ ($\mathcal{N}_i = \{R_i, L_i, F_i, B_i\}$) as shown in Figure 2.

We assume a leaderless formation. In addition, each robot obtains information about its surrounding environment from sensors whose detection range is radius $\rho(> 0)$. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ representing the inter-robot connections is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } p_{ij} \leq \rho \\ 0 & \text{otherwise} \end{cases}, \quad j \in \mathcal{N}_i \quad (7)$$

where $p_{ij}$ is the distance between robot $i$ and robot $j$.

This paper aims to make each robot follow the target coordinates $x_g \in \mathbb{R}$ while avoiding obstacles under the formation movement problem. For straight-line movement of the column formation, $x_g$ is set far enough away. In the $y$-direction, each robot only stays the desired distance $d \in \mathbb{R}$ from its neighbouring robots. Let the set of robots be $V = \{1, 2, \ldots, n\}$. The dynamics of robot $i \in V$ at discrete time $k \in \mathbb{N}$ is given by

$$p_i[k + 1] = p_i[k] + u_i[k],$$

$$p_i[k] = [p_{i,1}[k], p_{i,2}[k]]^T \quad (8)$$

where $p_i[k] \in \mathbb{R}^2$ is the $x$-$y$ coordinate of robot $i$ position, and $u_i[k] \in \mathbb{R}^2$ is the control input. $u_i[k]$ is...
composed of the consensus control law $\mathbf{u}_i^c[k] \in \mathbb{R}^2$ for preserving formation, the control law $\mathbf{u}_i^f[k] \in \mathbb{R}^2$ for reaching the target position, $\mathbf{u}_i^c[k] \in \mathbb{R}^2$ for avoiding robot collision and $\mathbf{u}_i^o[k] \in \mathbb{R}^2$ for avoiding obstacle collision as follows:

$$\mathbf{u}_i[k] = \mathbf{u}_i^f[k] + \mathbf{u}_i^c[k] + \mathbf{u}_i^r[k] + \mathbf{u}_i^o[k].$$  \hspace{1cm} (9)

Substituting Equation (9) into Equation (8) and arranging vertically for $i = 1, 2, \ldots, n$, we obtain

$$\mathbf{p}[k + 1] = \mathbf{p}[k] + \mathbf{u}_i^f[k] + \mathbf{u}_i^c[k] + \mathbf{u}_i^r[k] + \mathbf{u}_i^o[k].$$  \hspace{1cm} (10)

Consider the inputs $\mathbf{u}_i^f[k], \mathbf{u}_i^c[k]$ and $\mathbf{u}_i^o[k]$. The formation control preserving the desired distance between the inter-connected robots is based on consensus control. The desired vectors representing the distance between robot $i$ and robot $j \in \{R_i, L_i, F_i, B_i\} \subseteq \mathcal{N}_i$ are given as follows:

$$d_R_i = \begin{bmatrix} 0 \\ -d \end{bmatrix}, \quad d_L_i = \begin{bmatrix} 0 \\ d \end{bmatrix}, \quad d_F_i = \begin{bmatrix} -d \\ 0 \end{bmatrix},$$

$$d_B_i = \begin{bmatrix} d \\ 0 \end{bmatrix}. \hspace{1cm} (11)$$

For the swarm robots to form the desired shape, the following conditions need to be satisfied.

$$\lim_{k \to \infty} \left\| \mathbf{p}_i[k] - \mathbf{p}_j[k] - d_i \right\| = 0, \quad j \in \mathcal{N}_i$$

$$\lim_{k \to \infty} \left\| \tilde{\mathbf{p}}_i[k] - \tilde{\mathbf{p}}_j[k] \right\| = 0, \quad j \in \mathcal{N}_i$$  \hspace{1cm} (12)

where $\tilde{\mathbf{p}}_i[k] = \mathbf{p}_i[k] - d_i/2$, $\tilde{\mathbf{p}}_j[k] = \mathbf{p}_j[k] + d_j/2$ are the target relative vectors. From this, formation control can be expressed by the consensus control law. The consensus control law for preserving formation is given by

$$\mathbf{u}_i^f[k] = -\varepsilon \sum_{j \in \mathcal{N}_i} a_{ij} (\mathbf{p}_i[k] - \mathbf{p}_j[k] - d_i)$$  \hspace{1cm} (13)

where $d_i(j \in \{R_i, L_i, F_i, B_i\})$ is given by (11). The gravitational potential function $V_\mathbf{r}^f(r)$ for robot $i$ to reach the target position is given by

$$V_\mathbf{r}^f(r) = k_s \left\| x - x_0 \right\|^2$$  \hspace{1cm} (14)

where $k_s \in \mathbb{R}_+$ is the adjustment parameter, $r$ denotes the vector of $r = [x \ y]^T$. The target value $x_0$ is given to the first column of the formation as the target value $x_0$, and $x_n - n'd$ is given to the $n$th column from the first column as the target value $x_n$. The input $x_i[k]$ is the gradient of the potential function in Equation (14)

$$\mathbf{u}_i^f[k] = -\alpha \left( \frac{\partial V_\mathbf{r}^f(r)}{\partial r} \right)^T \mathbf{p}_i[k]$$  \hspace{1cm} (15)

where $\alpha$ is the step size.

For simplicity, regarding the convergence of the algorithm proposed in this paper, we do not consider the transient state with collision avoidance, but discuss the steady state. This is because the potential functions do not always converge to certain values except in steady-state conditions where there are no collisions. That is, if $\mathbf{u}_i^c[k] = \mathbf{u}_i^r[k] = \mathbf{0}$ in Equation (9), then

$$\mathbf{p}_i[k + 1] = \mathbf{p}_i[k] + \mathbf{u}_i^f[k] + \mathbf{u}_i^o[k]$$

$\mathbf{p}_i[k] + \mathbf{u}_i^f[k]$ is attributed to the consensus control with bias in Section 2.2. Also, $\mathbf{u}_i^o[k]$ satisfies the convergence condition for distributed optimization in Section 2.3 because $V_\mathbf{r}^o(r)$ is a convex function from Equations (14) and (15). The existing results [12–15] guarantee that the algorithm proposed in this paper converges in the steady state. Instead of considering the convergence condition with collision avoidance, we aim to propose a control algorithm that takes into account stability as much as possible in transient states where obstacle avoidance is considered. The collision avoidance of this paper is so-called ‘soft constraint’ such that the repulsion force does not diverge to infinity when a group of robots maintaining a formation avoids an obstacle. Since the exponential function is differentiable, the force imbalance problem can be analysed at the pole of a linear approximation system.

Also, to prevent robots from colliding with each other, we consider the repulsive potential function $V_\mathbf{r}^c(r)$ given by

$$V_\mathbf{r}^c(r) = \sum_{j \in \mathcal{N}_i} k_e \left( \frac{1}{\left\| \mathbf{p}_i[k] - \mathbf{p}_j[k] \right\|} \right)^\sigma$$  \hspace{1cm} (16)

where $k_e, \sigma \in \mathbb{R}_+$ are the adjustment parameters. The set $\mathcal{L}_i[k] = \{ j \in \mathcal{V} || \mathbf{p}_j - \mathbf{p}_i \| \leq \rho \}$ is the set of other robots that the robot detects with its sensors. $V_\mathbf{r}^c(r)$ is a circularly spreading potential function. $\mathbf{u}_i^o[k]$ are given by the gradient of the corresponding potential function:

$$\mathbf{u}_i^o[k] = -\alpha \left( \frac{\partial V_\mathbf{r}^c(r)}{\partial r} \right)^T \mathbf{p}_i[k]$$  \hspace{1cm} (17)

where $\alpha > 0$ is the step size.

The problem this paper considers is to reduce the robot oscillatory behaviour near obstacles and to recover the formation after the avoidance of obstacles. Two issues are raised against the column formation (10)–(17). The first issue is to alleviate the imbalance between the attractive force for preserving the formation and the repulsive force for avoiding obstacles. In addition, there is a force imbalance between the robots in the formation control. The second issue is to eliminate the stagnation of the swarm robots in front of the large obstacle. A solution to the first issue is to design a new network weight $a_{ij}$ instead of (7). We design the network weight $a_{ij}$ in (13) depending on the
relative distance between the robots. Also, as a solution to the second issue, we design the input $u^r[k]$ in (10) that changes the shape of the time varying potential function depending on the relative angle between the robot and the obstacle.

4. Main results

To consider the two issues, we introduce the variable $\hat{a}^{\ell}_i$ that represents the detection/nondetection of obstacle $\ell(=1,2,\ldots,c)$ by robot $i$ as follows [16]:

$$\hat{a}^{\ell}_i = \begin{cases} 1 & \text{if } \|\mathbf{p}_i - \mathbf{p}_{\ell}^o\| - r - R^\ell \leq D \\ 0 & \text{otherwise} \end{cases}$$

(18)

where $D > 0$ is a threshold of detection/nondetection, $\mathbf{p}_{\ell}^o$ is the coordinate of obstacle $\ell$, $r$ and $R^\ell$ are radius of robot $i$ and obstacle $\ell$, respectively. This variable $\hat{a}^{\ell}_i$ is used in the design of the network weight function and the time-varying potential function.

4.1. Design of network weight function

This section considers the first issue related to the network weight $a_{ij}$ in (13). We apply a network weight function depending on the distance between robots to preserve the formation of swarm robots after the collision avoidance.

The previous study [9] proposes two types of network weight functions. The first one takes into account the distance between robot $i$ and robot $j$ as follows:

$$a_{ij} = \frac{d}{\|\mathbf{p}_i - \mathbf{p}_j\|} \quad \text{if } j \in \mathcal{N}_i.$$  

(19)

In Figure 3(a), the orange line shows the shape of $a_{ij}$. The weight $a_{ij}$ corresponds to the strength of the connection between robot $i$ and robot $j$. The distance between the robots decreases, the corresponding strength increases. The second one takes into account the disconnection of the network:

$$a_{ij} = \begin{cases} \frac{d}{\|\mathbf{p}_i - \mathbf{p}_j\|} & \text{if } \|\mathbf{p}_i - \mathbf{p}_j\| \leq \rho \\ 0 & \text{if } \hat{a}^{\ell}_i = 1 \quad \text{and} \quad \frac{d}{\|\mathbf{p}_i - \mathbf{p}_j\|} < \sigma \end{cases}$$

(20)

where $\sigma > 0$ is a threshold. As shown in the blue line of Figure 3(a), this weight becomes 0 when the strength of the connection between the robots is less than $\sigma$. The weight $a_{ij}$ in (20) includes the weight $a_{ij}$ in (19) as a special case $\sigma \to 0$. This disconnection eliminates the imbalance between the attractive force for preserving the formation and the repulsive force when the robots diverge each other due to the collision avoidance. On the other hand, there still remains the imbalance force when the distance between the robots becomes close. This phenomenon is demonstrated in Section 5.

To solve the above open problem of [9], this paper proposes a new network weight function that changes its value depending on the displacement between the robot relative distance $\|\mathbf{p}_i - \mathbf{p}_j\|$ and the desired distance $d$ as shown in Figure 3(b). That is, the proposed network weight function is defined as follows:

$$a_{ij} = \exp\left(-\frac{d}{\|\mathbf{p}_i - \mathbf{p}_j\| - d}\right) \quad \text{if } j \in \mathcal{N}_i.$$  

(21)

This works to weaken the formation control when the distance between the networked robots is greater or less than the desired distance $d$ as shown in Figure 3(b). (21) may alleviate the force imbalance when the desired formation cannot be maintained due to the obstacle avoidance. The effectiveness is demonstrated in Section 5.

4.2. Time-varying potential function

This section considers the second issue related to the input $u^r[k]$ in (10). To avoid the obstacles, the previous work of the current authors [9] uses a circularly spreading potential function. The large number of robots causes stagnation around obstacles. This situation will be demonstrated in Section 5. To address this issue, this paper considers a time-varying potential function. The basic idea is the potential function considering the relative angle between the robot and the obstacle [11].

First, we explain the potential function proposed in [11]. $\mathbf{r}$ denotes the vector of $\mathbf{r} = [x \ y]^\top$. The repulsive potential function $V^\ell_{ij}(\mathbf{r})$ for robot $i$ to avoid obstacle $\ell$

$$a_{ij}$$

$\begin{array}{c}
\text{(a)} \\
\text{Figure 3. Network weight function: (a) previous method [9] and (b) proposed method.}
\end{array}$
is given by

$$V_i^o(\mathbf{r}) = \sum_{\ell=1}^{c} k_\ell^o w^\ell(\theta_i^{\text{go}}) \dot{a}_i^\ell e^{-\frac{|r-p_\ell^i|}{\sigma_\ell^i}}$$  \hspace{1cm} (22)$$

where $k_\ell^o, \sigma_\ell^o \in \mathbb{R}_+$ are the adjustment parameters. $w^\ell(\theta_i^{\text{go}})$ is the weight function depending on the relative angle $\theta_i^{\text{go}}$ between the robot and the obstacle. If $w^\ell(\theta_i^{\text{go}})$ is constant, (22) is a circularly spreading function whose exponent is the distance between robot $i$ and obstacles $\ell (= 1, 2, \ldots, c)$. The previous study [11] achieves smooth collision avoidance by introducing the weight function $w^\ell(\theta_i^{\text{go}})$. The target position of [11] is a single point, while the target position of this paper is not a single point for the straight-line movement. For simplicity of explanation, we use the relative angle that is in line with the problem setting of this paper as shown in Figure 4. The relative angle $\theta_i^{\text{go}}$ between the direction of robot $i$ and obstacle $\ell$ is given by

$$\theta_i^{\text{go}} = \cos^{-1} \left( \frac{\mathbf{e}_x \cdot (\mathbf{p}_i - \mathbf{p}_\ell^i)}{\|\mathbf{p}_i - \mathbf{p}_\ell^i\|} \right)$$  \hspace{1cm} (23)$$

where $\mathbf{e}_x$ is the unit vector in the $x$-axis direction. The weight function $w^\ell(\theta_i^{\text{go}})$ proposed in [11] enables us to change the range of influence of the repulsive force, for example, from circularly spreading to teardrop spreading. The weight is given by

$$w^\ell(\theta_i^{\text{go}}) = \begin{cases} c_w e^{-c_{\text{rad}}(\theta_i^{\text{go}})} + c_0 & \text{if } 0 < \theta_i^{\text{go}} < \theta_s \\ 0 & \text{otherwise} \end{cases}$$

$$c_w = \frac{1}{(1 - e^{-c_{\text{rad}}(\theta_s)})}, \quad c_0 = -e^{-c_{\text{rad}}(\theta_s)}$$  \hspace{1cm} (24)$$

where $c_w, \theta_s \in \mathbb{R}_+$ are the adjustment parameters, $\text{rad}(\cdot)$ is the radian of the angle unit. Varying the repulsion potential function based on the relative angle can avoid an unnecessarily large turn due to the collision avoidance.

Second, based on the above idea, we consider the mitigation of the stagnation phenomenon when swarm robots avoid the large obstacle. The stagnation of swarm robots causes a misalignment of the robot positions. This makes it difficult for the robots to preserve the desired formation after avoiding obstacles. To avoid this problem, we consider changing the shape of the potential function from a circle to an ellipse. In addition, we adjust the lengths of the major and minor axes based on the positional relationship between the robot and the obstacle. The resulting time-varying repulsive potential function makes the swarm robots stay in the desired formation after avoiding obstacles. Specifically, the repulsive potential function $V_i^o(\mathbf{r})$ for robot $i$ to avoid the obstacle is given by

$$V_i^o(\mathbf{r}) = \sum_{\ell=1}^{c} k_\ell^o w^\ell(\theta_i^{\text{go}}) \dot{a}_i^\ell e^{-\frac{|r-p_\ell^i|}{\sigma_\ell^i}}$$

$$\exp \left[ \frac{\left( x - p_{\alpha}^\ell \right)^2}{\left( \sigma_{\alpha}^\ell + \sigma_s \right)^2} + \frac{\left( y - p_{\beta}^\ell \right)^2}{\left( \sigma_{\beta}^\ell + \sigma_s \right)^2} \right]$$

$$\sigma_x = \frac{k_x R^\ell}{r} \left( \theta_s - \theta_i^{\text{go}} \right) \quad \text{if } 0 < \theta_i^{\text{go}} < \theta_s$$

$$\sigma_y = \frac{k_y R^\ell}{r} \left( \theta_s - \theta_i^{\text{go}} \right) \quad \text{if } 0 < \theta_i^{\text{go}} < \theta_s$$

$$\text{otherwise}$$

$$\text{otherwise}$$  \hspace{1cm} (25)$$

where $k_x, k_y \in \mathbb{R}_+$ are the adjustment parameters, $p_{\alpha}^\ell, p_{\beta}^\ell$ are the $x, y$ coordinates of the obstacle $\ell$. The initial shape of the repulsive potential function is elongated in the direction of $x$-axis when the relative angle $\theta_i^{\text{go}}$ is almost all 0. The shape shrinks in the direction of $x$-axis and goes to a circular one as the robot approaches the obstacle in the direction of $x$-axis and crosses the obstacle. The situation is shown in Figure 5.

Finally, the input $u_i^o[k]$ is given by the gradient of $V_i^o(\mathbf{r})$ is given by

$$u_i^o[k] = -\alpha \frac{\partial V_i^o(\mathbf{r})}{\partial \mathbf{r}} \bigg|_{p[k]}^\mathsf{T}$$

where $\alpha$ is the step size. Applying the time-varying repulsive potential function, we can generate a trajectory (solid red line) for the robot that smoothly avoids obstacles as shown in Figure 6.

5. Numerical experiment

In this study, numerical experiments are conducted using MATLAB. The five experimental patterns prepare a combination of the network weight $q_{ij}$ and the potential function $V_i^o(\mathbf{r})$ used in (26) against the column formation (10)–(15). This paper verifies the trajectories and analyses the stability of the swarm robots. Table 1 shows the parameters for each experiment. In this paper, gain tuning is a trial-and-error process. The larger the size of the obstacle, the larger the gain. The sensing range $\rho$ should be larger than $D$, since the
First of all, we focus on convergence in steady state without collision. The convergence with respect to the objective function is shown in Figure 7 where only $V^g$ is considered because $V^s$ is for tracking. The target $x$-coordinate of the first row ($i = 1, 2, \ldots, 100$) is set to $x_g$ and the target position of the robots in the $n'$th row from the top is set to $x_g - n'd$. Figure 7 shows that all patterns converge to the target position after enough time has passed. This result indicates that the all potential functions do not adversely affect the objective function $V^g$ for tracking in a collision-free steady state.

In the following sections, the target position is set to a common value for all robots for simplicity (all $x_i^g$ are the same) and we discuss the oscillatory behaviour and stability of the algorithms during collision avoidance.
5.1. Trajectories

The results of numerical experiments are shown in Figures 8 and 9. The solid black line represents the trajectory of the swarm robots, and the gray circles represent the obstacles. Pattern (a) is the power imbalance between maintaining formation and avoiding obstacles, Patterns (b)–(c) are the force imbalance between robots in formation control, Pattern (d) is the non-recovery of the formation after avoiding obstacles, Pattern (e) is recover formation after avoiding obstacles. In pattern (a), oscillatory behaviours are observed near the obstacles. Patterns (b) and (c), to which the method of the previous study [9] is applied, also show oscillatory behaviours. This is because the repulsion force becomes too large due to the network weight functions (19) and (20) when the robots are too close to each other. As a result of increasing the total number of robots, the oscillation could not be reduced by the method of the previous study [9]. In pattern (d), we use the proposed network weight function (21) and a potential function with weights that adjust the repulsion depending on the angle [11]. Pattern (d) of Figures 8 and 9 shows that the proposed network weight function (21) reduces the oscillatory behaviours near obstacles. However, the swarm robots are not able to recover to the desired formation after avoiding obstacles. The cause of this is thought to be the stagnation of the swarm robots in front of the large obstacle. In pattern (e), which uses the proposed potential function (25), the swarm robots are able to recover to the desired formation without stagnating in front of the obstacle.

Figure 10 shows the \( x \)-coordinate of the robots \((i = 1, 2, \ldots, 100)\) at the head of the formation as the ordinate and the number of steps \(k\) as the abscissa. These figures indicate the degree of convergence to the target \(x\)-coordinate \(x_g\). Since the variation of Pattern (e) is small against Patterns (a)–(c), it is considered...
that the imbalance between the attraction of the formation control and the repulsion of obstacle avoidance has been alleviated by the proposed method. Also, comparing Pattern (e) with Pattern (d), we see that the time-varying repulsive potential function eliminates the stagnation in front of obstacles during the formation movement.

5.2. Stability analysis
To analyse the proposed method, we approximate the entire robots behaviour to the form $p[k + 1] = A[k]p[k]$, and perform eigenvalue analysis from the coefficient matrix $A[k]$. Specifically, for each step, we obtain a linear approximation system and its eigenvalues. When the maximum eigenvalue exceeds 1, it means that the robots leave the desired position and diverge due to obstacle avoidance. For Patterns (a), (b), (d), and (e), the eigenvalues are obtained as real numbers because the coefficient matrices are symmetric. In Pattern (c), the coefficient matrix is sometimes not symmetric because of the network disconnections. In Figure 11, we compare the maximum eigenvalues for each pattern. This figure shows that the maximum eigenvalues of Patterns (a)–(c) are oscillatory and the robots divergence is large. In Patterns (d) and (e), the maximum eigenvalues are not oscillatory and the robots divergence is small. Pattern (c) has the highest robots divergence among all patterns, and Pattern (e) has the lowest. The reason of the worst result due to Pattern (c) is that we increased the total number of robots compared to the problem setting in the previous study [9]. In the previous study [9], the oscillatory behaviour is not completely suppressed, and it is believed that the increase in the total number of robots caused the oscillation to propagate. The proposed method in this study can be applied even when the total number of robots is increased, and this method is considered to be

Figure 9. Trajectories (Enlarge): (a) conventional; (b) previous 1; (c) previous 2; (d) proposed 1; (e) proposed 1 + proposed 2.
effective. Also, the maximum eigenvalues for different values of gain and sensing range for obstacle avoidance are summarized in Table 2. From this, pattern (e) is the smallest even when the gain is changed. In other words, the two proposed methods have the lowest degree of instability regardless of the gain value.

6. Conclusion
This paper has proposed a network weight function and a time-varying repulsion potential function for obstacle avoidance of swarm robots in column formation. The proposed network weight function is a method to reduce the force of the formation control according
to the distance between robots. The repulsion potential function is a method to change the shape according to the angle between the direction of motion and the obstacle. From the results of the numerical experiments, the oscillatory behaviour of the swarm robots during obstacle avoidance is reduced compared to the conventional method, and the desired formation can be preserved after passing through the obstacle. Future work is a generalization of the parameter selection and consideration of the limited sensor range.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This paper is funded by JSPS KAKENHI Grant Number JP20K20314, JP19K04444, JP19H02163, and JP17H06293.

Notes on contributors

Shotaro Shibahara received the B.E. degree in mechanical engineering from the University of Electro-Communications, Tokyo, Japan, in 2021. He is currently pursuing the M.E. degree at the University of Electro-Communications. His research interests include the control theory of multi-agent systems.

Takuma Wakasa received the B.E. and M.E. degrees in mechanical engineering from the University of Electro-Communications, Tokyo, Japan, in 2019 and 2021. His research interests include the control theory of multi-agent systems.

Kenji Sawada received his Ph.D. degree in engineering in 2009 from Osaka University. He is an Associate Professor in Info-Powered Energy System Research Center, The University of Electro-Communications, Japan. He is also an advisor of Control System Security Center since 2016. He received Outstanding Paper Awards from FA Foundation (2015 and 2019), Fluid Power Technology Promotion Foundation (2018), and JSME (2018). His research interests include the control theory of cyber-physical systems and control system security. He is a member of IEICE, ISCIE, IEEJ, JSME, IEEE.

References

[1] Tang Q, Yu F, Xu Z, et al. Swarm robots search for multiple targets. IEEE Access. 2020;8:92814–92826.
[2] Khan MSA, Ahmed T, Uddin MF. Multi-robot search algorithm using timed random switching of exploration approaches. 2020 IEEE Region 10 Symposium (TENSYMP); 2020. p. 868–871.
[3] Macktoobian M, Gillet D, Kneib J-P. Astrobotics: swarm robotics for astrophysical studies. IEEE Robot Automat Mag. 2021;28(3):92–101.
[4] Izumi S, Shiomoto Y, Xin X. Mass game simulator: an entertainment application of multiagent control. IEEE Access. 2020;9:4129–4140.
[5] Ping S, Kejie L, Xiaobing H, et al. Formation and obstacle-avoidance control for mobile swarm robots based on artificial potential field. IEEE International Conference on Robotics and Biomimetics (ROBIO); 2009. p. 2273–2277.
[6] Yang S, Li T, Shi Q, et al. 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); 2020. p. 58–63.
[7] Lanejin X, Xu W. The null-space-based behavioral control for a swarm of robots tracking a target region.
in obstacle environments. Chinese Control Conference (CCC); 2019. p. 5574–5578.

[8] Huang D, Yuan Q, Li X. Decentralized flocking of multi-agent system based on MPC with obstacle/collision avoidance. Chinese Control Conference (CCC); 2019. p. 5587–5592.

[9] Shibahara S, Wakasa T, Sawada K. Network weight function for obstacle avoidance of swarm robots in column formation. SICE Annual Conference 2021; 2021. p. 259–262.

[10] Henry J, Shuma HPH, Komura T. Interactive formation control in complex environments. IEEE Trans Vis Comput Graph. 2014;20(2):211–222.

[11] Kim DH, Lee H-W, Shin S, et al. Local path planning based on new repulsive potential function with angle distributions. International Conference on Information Technology and Applications (ICITA’05); 2005.

[12] Olfati-Saber R, Alex Fax J, Murray RM. Consensus and cooperation in networked multi-Agent systems. Proc IEEE. 2007;95(1):215–233.

[13] Nedić A, Ozdaglar A. Distributed subgradient methods for multi-Agent optimization. IEEE Trans Automat Contr. 2009;59(1):48–61.

[14] Nedić A, Ozdaglar A. Cooperative distributed multi-agent optimization. Convex Optimization in Signal Processing and Communications. 2009. p. 340–386.

[15] Azuma S-i., Nagahara M, Ishii H, et al. Control of multi-Agent systems. Japan: Corona Publishing Co. Ltd.; 2015.

[16] Dang AD, Horm J. Formation control of autonomous robots following desired formation during tracking a moving target. International Conference on Cybernetics (CYBCONF); 2015. p. 160–165.