Robotica (2011) volume 29, pp. 515–525. © Cambridge University Press 2010
doi:10.1017/S0263574710000366

Optimization-based formation of autonomous mobile robots

Huan Zhang† and Pubudu N. Pathirana‡*

†*Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia*
E-mail: Hzhang5@unimelb.edu.au
‡*School of Engineering and Information Technology, Deakin University, Pigdons Road, Geelong, VIC 3217, Australia*

(Received in Final Form: July 16, 2010. First published online: August 5, 2010)

SUMMARY
The formation of autonomous mobile robots to an arbitrary geometric pattern in a distributed fashion is a fundamental problem in formation control. This paper presents a new asynchronous, memoryless (oblivious) algorithm to the formation problem via distributed optimization techniques. The optimization minimizes an appropriately defined difference function between the current robot distribution and the target geometric pattern. The optimization processes are performed independently by individual robots in their local coordinate systems. A movement strategy derived from the results of the distributed optimizations guarantees that every movement makes the current robot configuration approaches the target geometric pattern until the final pattern is reached.

KEYWORDS: Swarm robotics; Mobile robots; Multi-robot systems; Motion planning; Navigation.

1. Introduction

This paper studies the problem of coordinating a group of autonomous mobile robots to form a prescribed geometric pattern in a two-dimensional (2D) plane. The study of cooperative behavior among a group of autonomous agents has drawn increased attention of researchers in many disciplines, including biology, robotics, and control community.[7,9–13,17] Swarm behaviors in certain living beings have been observed for a long time. Examples are bird flocks, fish schools, and animal herds. The cooperative behavior is very important to the survival of these species since it has certain advantages such as avoiding predators and increasing the chance of finding food. Usually, swarm behaviors in nature are understood as a consequence of simultaneous mutual attraction and repulsion.[7,11] Similarly, cooperative behaviors are also important for many engineering applications such as multi-robot teams, autonomous air vehicles, and automated highway systems.[2,4,8] Formation problem formulation and solutions may be characterized by their sensing requirements, their method of behavior integration, and their commitment to replanning.[2] For example, Fax *et al.*[5,9,10,13] studied the conditions of geometric pattern formation using local information alone, i.e. positions of neighbors.

* Corresponding author. E-mail: Pubudu.Pathirana@deakin.edu.au

The formation problem we study in this paper can be described by an example.[16] Suppose a schoolteacher asks his or her students to form a particular shape, say a fixed size ellipse, in the ground to play a game. However, the teacher does not designate the position and orientation of the ellipse; hence the students don't have a consensus on where and what orientation the final ellipse should be. Furthermore, the students are not permitted to talk to each other and the only information they can use to form the ellipse is the observed relative positions of other students.

This class of autonomous agent formation problem was formulated and studied by Suzuki and Ramashita,[16] Flocchini *et al.*[3,6] The main feature of this form of formation problem can be described by the so-called hard task for "weak robots."[6] For this, it means that the robots are anonymous, memoryless, asynchronous with only local views, limited movement capabilities and without communication capabilities, albeit they are required to achieve a given target geometric pattern (TGP) from any given initial robots configuration. Heuristic algorithms[2,14,15] for the formation problem were proposed and the accuracy of these was tested either by simulations or experiments. Recent concerns on the computational complexity of the coordination problem have motivated researchers to approach the underlying problem from a computational standpoint.[1,3,16] For example, Cieliebak *et al.* proposed a computationally tractable approach that solved the gathering problem, i.e. forming to a point.

In this paper, we propose and formalize the following formation ideas. Let us consider the example of ellipse formation problem by the students. In determining a move strategy, every student first relocates and rotates the target ellipse (since they are not prefixed by the teacher) to compare with the form of current positions he or she observed in order to find the best matching location and orientation of the target ellipse. From this optimal target ellipse location and orientation, every student is able to obtain the next position he/she should move to. The process of finding the best matching position and orientation involves solving an optimization problem of minimizing an appropriate cost function defining the difference between the target and current geometric forms. Although the comparing processes are performed in a distributed fashion by the students independently in their local coordinate systems, we show that the results and the effects bear global consequence. That is, different students will come up with the same "best

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

matching" ellipse location and orientation. Consequently, every movement based on the same best matching target geometric form leads to a new better distribution for the students, which is closer to the final ellipse to be formed. Repeating the process finally achieves the target geometric form.

Compared with most of the existing heuristic solutions for similar problems, optimization-based formation ideas provide a general, yet mathematically rigorous and justified approaches for solving distributed formation problems. The mathematical framework also allows an analysis of the performance of the algorithm including the speed of convergence.

This paper is organized as follows. After carefully defining the formation problem in Section 2, we present a solution in Section 3 via distributed optimization ideas. Section 4 proposes both synchronous and asynchronous algorithms based on the ideas in Section 3. Section 5 presents simulations for the proposed algorithms and Section 6 concludes the paper.

2. Formation Problem Statement

Consider the problem of coordinating a group of autonomous, mobile robots to form a specific geometric pattern in a plane. A *geometric pattern* is a distribution of the robots in the 2D plane such as a uniform distribution on a circle. However, the location and orientation of the geometric pattern are not fixed in advance. The formation is performed by autonomous robots, which means there is no central control scheme available and each robot moves based on local measurements and computations.

The solution to this problem depends on a number of problem specifications. In particular, the degree of common knowledge, the capability of a single robot, and the geometric pattern the robots are required to form have a prominent impact on the structure and complexity of the solution. The amount of available common knowledge is vital to the solvability of the problem.[6] For example, if there is a common (global) coordinate system available to the group of robots, then the problem is almost trivial since every robot can be given the explicit target position in the geometric pattern in terms of this coordinate system. It becomes more difficult if all the robots can only view the world according to their own local coordinate system. The capabilities of a single robot such as the computational power, memory size, and movement ability also affect the nature of the solution. For example, it has been proved in ref. [16] that if the robots have a limited memory size and can only make a decision of movement based on the current view of the world (*oblivious*), then it is impossible for two robots to gather at one point in finite steps, while they can do it if the robots can remember and utilize the past observations (*nonoblivious*). The target geometric pattern is also important to the solution. While simple solutions exist for forming simple symmetric patterns like a circle, some geometric patterns may not even be achievable.[16]

In this section we carefully formulate our autonomous robots formation problem. That is, we clearly specify factors such as common knowledge, the robot capabilities, and properties of target geometric patterns. The aim is to formulate a formation problem in which a group of "*weak robots*" accomplish a "*hard task.*"

2.1. Weak robots

Let us study the behavior of a group of n robots in a plane. A robot is defined as a mobile computational unit equipped with sensors. It is viewed as a point in the plane, and hence the problem of collision does not arise as in refs. [3, 16]. We assume the robots are identical and *anonymous*. This implies that a robot does not have an identity to use in the computation. We also assume that there is no active communication between the robots. Here active communication means communicating via some channels with protocols. In our problem setting, all of the robots are silent and they only passively see the relative positions of other robots. Here "passively see" means the sensing robot acquires the position information without the involvement of the robot being sensed. The locations of other robots are expressed in terms of the observer's *local* (Cartesian) coordinate system $\mathcal{Z}_i = (\mathbf{o}_i, \mathbf{x}_i, \mathbf{u}_i)$. Here $\mathbf{o}_i$ is the location of the origin, $\mathbf{x}_i$ is the direction of the positive $\mathbf{x}$ axis, and $\mathbf{u}_i$ is the size of the unit distance. For simplicity, we assume that all robots have a common sense of distance, i.e. $\mathbf{u}_i = u, \forall i$. It is also assumed that all robots have a common sense of orientation so that the positive $\mathbf{y}_i$-direction is 90 degrees counterclockwise from the positive $\mathbf{x}_i$-direction as shown in Fig. 1.

We assume that the robots have a limited memory capacity such that they can remember only the current location information of other robots. The robots always move in straight lines and the maximum distance a robot can move in one step is $\epsilon > 0$.

2.2. Hard task

The task for robots is to form a particular TGP from an arbitrary initial distribution. We assume the robots initially occupy different points in the plane. The TGP is passed to all robots at the beginning. Because there are no global coordinate systems present, the location and orientation of the final form cannot be specified. Instead, different robots view the TGP in their local coordinate systems and may have different understanding of the location and orientation of the TGP as shown in Fig. 1. However, we do fix the size of the TGP, which means it is not scalable as those in refs. [6, 16]. Fixing the size is possible because we have assumed a common unit of length for all the local coordinate systems. Furthermore, we assume that the target robot positions on the TGP are all distinct. The formation process must be performed in a fully *distributed* fashion, by which we mean each robot executes the same algorithm independently to determine its movement. There is no central scheme coordinating the collective behavior of the group of robots. This is the reason that the robots are termed as *autonomous*.

We assume that the formation process takes place in discrete time sequence $T = \{t_0, t_1, t_2, \ldots\}$. At any time instance t_k, a robot can be in the state of either *sleep* or *awake*. When a robot is in *sleep* state, it does nothing. While when it is *awake*, it goes through a cycle of three

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

actions: *LOOK*, *THINK,* and *MOVE*. The *LOOK* action returns the robot with the locations of all other robots at that moment via its sensors. This establishes the current robot configuration (CRC). In the *THINK* action, the robot executes the algorithm and makes a decision of the next movement based on the established CRC. In the *MOVE* action, the robot simply moves to the destination point along the straight line connecting the current and destination points. If the destination point is less than ϵ distance away, then it reaches it directly. Otherwise, it travels ϵ distance toward the destination along the line. In that case, the robot is not able to remember the current computational result. At a later time, it has to go through the cycle again and recalculate the new destination position. Assume that for robot $\mathcal{R}_i$, the *activation* times are given by the sequence $\tau_i = (t_1^i, t_2^i, \ldots)$, which is a sequence of time instants when the robots are awake. By *asynchronous* movement, we mean the activation sequences of different robots are independent, while we call the movement *synchronous* if every robot is active at every time instant, that is, $\tau_i = T, i = 1, 2, \ldots, N$. For asynchronous movement, we assume that every robot is activated infinitely many times, but no additional assumptions are made on the timing of activations. In the above notion of *asynchronous* movement, it does require the robots to have some common sense of time since all robots know the same time sequence T.

Since the robots have only a limited memory size, they remember neither any past observations nor any computational results. In calculating the next destination position, the robot uses only the measured current locations of all robots. An algorithm that makes use of only the current robots configuration to determine its movement is called *oblivious*. In the formation problem, we also require the robots to achieve the target geometric pattern in finite time steps. If a geometric pattern is reached asymptotically, then it is usually referred to as the *convergence* problem.

Now we summarize the formation design problem as follows: For a group of robots with limited memory size and limited mobility, we seek an oblivious, distributed algorithm such that each robot uses its observations to determine its movement to form a given target geometric pattern from any arbitrary initial configuration in the plane without the presence of a global coordination scheme and direct communication.

3. Optimization-Based Solutions

In this section, we present a systematic solution to the above formation problem. The general ideas are schematically shown in Figs. 1–4.

In step 1 (Fig. 1), the robots are assigned a task to form an ellipse in the plane. The location and orientation of the ellipse are not prefixed. Different robots view the current robot position distribution differently in their local coordinate systems and also have different understanding of the ellipse.

In step 2 (Fig. 2), any active robot (here robot R_1) performs optimization to find the best location of the ellipse. We will prove in Theorem 3.1 that, for the cost function defined below, the best location for the ellipse is at such a position that the center of gravity of the current robot configuration and the ellipse are at the same point. To help perform a rotation optimization in step 3, move the origin of the local coordinate system $\mathcal{Z}_1$ to the center of gravity.

In step 3 (Fig. 3), an optimization to find the best orientation of the ellipse with respect to the current configuration is performed. Rotating to the best orientation

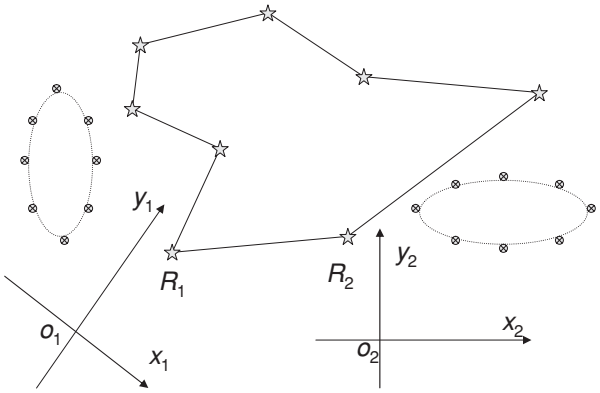


Fig. 1. Step 1: The target form (an ellipse) has been assigned. The robots have different understanding of the assignment in terms of the location and orientation of the ellipse in their own coordinate systems. The stars represent the current robot positions and dots represent the target positions of the robots.

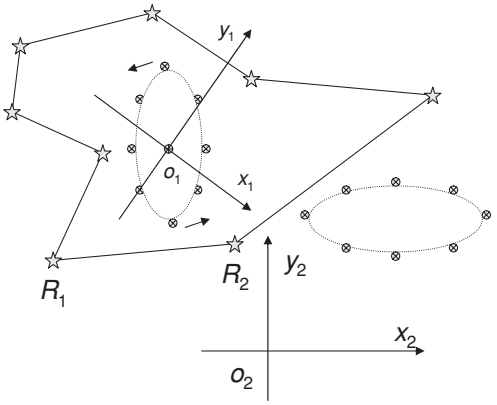


Fig. 2. Step 2: Robot $\mathbf{R}_1$ finds the optimal location for the ellipse with respect to the current configuration. It is proven that the best ellipse location is at such a location that the center of gravity of the TGP and the CRC is the same. Then do a translation to move the origin of the local coordinate system $\mathcal{Z}_1$ to the center of gravity.

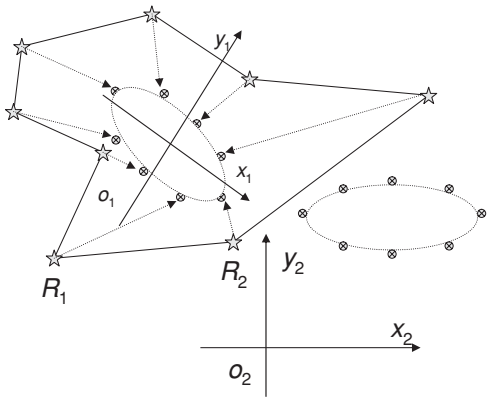


Fig. 3. Step 3: Do a rotation to find the best orientation of the TGP, and at the best location and orientation, it returns with the destinations for every robot.

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

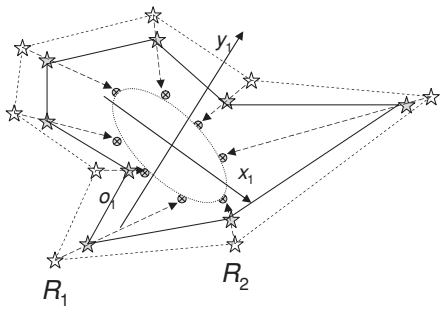


Fig. 4. Step 4: The computations for all robots will come up with the same destination points. The robots move a distance of ϵ along the straight lines connecting the current and destination points, which lead to a new form more similar to the TGP.

will produce robot R_1 the destination of itself as well as all other robots.

If another robot R_2 is active as well and does the same as R_1, then we will prove in Theorem 3.3 that R_2 will get the same best matching location and orientation for the ellipse. So R_2 will come to the same destination points for all robots as R_1 does. That is how the distributed computation leads to the common action strategy.

In step 4 (Fig. 4), all active robots move to their computed destinations in the straight lines connecting their current and destination points at a distance of ϵ.

Note that the location and orientation obtained are not necessarily the final ellipse location and orientation that the robots will eventually achieve. It may not be possible to reach all the destinations in one step due to the mobility limitations of the robots as well as the asynchronous nature of the formation. However, any movement of the robots will lead to a new robot configuration which is closer to the target ellipse. Also, because of the oblivious characteristic of the underlying formation, the robots need to go through step 1 to step 4 recursively to obtain a new best matching location and orientation of the ellipse with respect to the updated robot distribution. We will prove in Theorem 4.1 that the process can achieve the formation in finite steps for synchronous case and in Theorem 4.2 that it achieves the final form asymptotically in the asynchronous case.

3.1. Measure of difference between CRC and TGP

The essential point of the aforementioned approach is to guarantee that any movement of the robots puts the CRC more closer to the TGP. Mathematically, it can be realized as reducing some quantity indicating the difference between the CRC and the TGP. We now define such a function in an arbitrary local coordinate system.

Assume a robot $\mathcal{R}_i$ is active at some time instant $t \in \tau_i$. It observes and records all the current robot positions $\{z_j^{i,t} = [x_j, y_j]_i^{'t}, j = 1, 2, \ldots, n\}$, which establishes the CRC at time t. Here j represents the robot $\mathcal{R}_j$ while the subscript i indicates that the locations are expressed in the local coordinate system $\mathcal{Z}_i = (\mathbf{o}_i, \mathbf{x}_i, \mathbf{u}_i)$ of robot $\mathcal{R}_i$. When the local coordinate system and time instant are clear from the context, we save the notation and denote the positions as $\{z_j = [x_j, y_j]', j = 1, 2, \ldots, n\}$. Without loss of generality, we can assume that the origin $\mathbf{o}_i$ of $\mathcal{Z}_i$ is at the center of gravity (CoG) $\bar{z} = [\bar{x}, \bar{y}]'$ of the CRC according to

$$\bar{z} = \frac{1}{n}\sum_{j=1}^{n} z_j. \tag{1}$$

In such a coordinate system, the robot understands the TGP by marking the coordinates of the target robot positions $\{\gamma_j = [\alpha_j, \beta_j]', j = 1, 2, \ldots, n\}$. With no loss of generality, we assume that the CoG of the TGP $\bar{\gamma} = \frac{1}{n}\sum_{j=1}^{n}\gamma_j$ is at the origin $\mathbf{o}_i$, i.e. $\bar{\gamma} = [\bar{\alpha}, \bar{\beta}]' = [0, 0]'$.

Now a robot can order $[\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n]$ and $[P_1, P_2, \ldots, P_n]$ as the current and target robot positions according to the ascending order of the angles between the positive **x**-axis and the half line that starts from the origin and passes through the current (target) robot positions. If the angle is the same for two robots, then it is ordered according to the length of the segment between the origin and the robot position. With this ordering at hand, the coordinates of the current and target robot positions can be expressed in a more compact form as row vectors

$$Z = [z_1, z_2, \ldots, z_n], \Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]. \tag{2}$$

Under the ordering convention, we can also define a notion of *correspondence* between the robot positions of the CRC and the TGP. Denote $\delta = [\delta_1, \delta_2, \ldots, \delta_n], \delta_j \in \{1, 2, \ldots, n\}$ a permutation of $[1, 2, \ldots, n]$, and denote the set of all permutations $\Delta = \{\delta \text{ isapermutation}\}$. We define that a permutation $\delta = [\delta_1, \delta_2, \ldots, \delta_n]$ designates a *correspondence* between the current and target robot positions as

$$\mathcal{R}_j \leftrightarrow P_{\delta_j}, j = 1, 2, \ldots, n.$$

Designate $\Gamma_\delta = [\gamma_{\delta_1}, \gamma_{\delta_2}, \ldots, \gamma_{\delta_n}]$ the vector of target robot positions of ordering $\delta = [\delta_1, \delta_2, \ldots, \delta_n]$. All permutations Δ designates all possible one-to-one correspondences between the current and target robot positions under the CRC and the TGP in the current local coordinate system.

Given a correspondence $\delta = [\delta_1, \delta_2, \ldots, \delta_n]$ and the coordinates of the current and target robot positions in the form of the vectors Z and Γ, respectively, we define a difference function for the CRC and the TGP viewed by a robot $\mathcal{R}_i$ in its coordinate system $\mathcal{Z}_i$ as

$$\begin{aligned}\mathbf{J}^i(\delta) &= \|Z - \Gamma_\delta\|^2 \\ &\doteq \sum_{j=1}^{n}\|z_j - \gamma_{\delta_j}\|^2 \\ &= \sum_{j=1}^{n}[(x_j - \alpha_{\delta_j})^2 + (y_j - \beta_{\delta_j})^2].\end{aligned} \tag{3}$$

Here $\mathbf{J}^i(\delta)$ is the total Euclidean distances of all pairs $(\mathcal{R}_j \leftrightarrow P_{\delta_j})$, $j = 1, 2, \ldots, n$. It is obvious that $\mathbf{J}^i(\delta) \geq 0$ and $\mathbf{J}^i(\delta) = 0$ iff $x_j = \alpha_{\delta_j}, y_j = \beta_{\delta_j}, j = 1, 2, \ldots, n$, i.e. iff the robots achieve the TGP.

The robot $\mathcal{R}_i$ can rotate the TGP such that the TGP matches the CRC better, i.e. results in a smaller value of $\mathbf{J}^i$. A rotation operation of $\theta \in [-\pi, \pi)$ counterclockwise (a negative angle

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

$\theta \in (-\pi, 0)$ means a rotation of $-\theta$ clockwise) transforms the coordinates of the robot target positions Γ to

$$\Gamma(\theta) = \Pi(\theta)\Gamma,$$

where the rotation matrix $\Pi(\theta)$ is given by

$$\Pi(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ +\sin(\theta) & \cos(\theta) \end{bmatrix}.$$

Now the difference function after rotation is

$$\begin{aligned} \mathbf{J}^i(\delta, \theta) &= \|Z - \Gamma_\delta(\theta)\|^2 \\ &= \|Z - \Pi(\theta)\Gamma_\delta\|^2. \end{aligned} \tag{4}$$

Furthermore, the robot can translate the TGP to find a better matching between the TGP and the CRC. If the CoG of the TGP moves to $\psi = [\alpha, \beta]'$, the translation can be defined by the transform of the post-rotation target robot positions $\Gamma(\theta)$:

$$\Gamma(\theta, \psi) = \Gamma(\theta) + \Psi,$$

where $\Psi = [\psi, \psi, \ldots, \psi]$. After the rotation and translation, the difference function (3) becomes

$$\begin{aligned} \mathbf{J}^i(\delta, \theta, \psi) &= \|Z - \Gamma_\delta(\theta, \psi)\|^2 \\ &= \|Z - \Pi(\theta)\Gamma_\delta - \Psi\|^2. \end{aligned} \tag{5}$$

3.2. Finding the best matching TGP

In our algorithm, the most important computation for a robot to determine the next destination is to find the best matching TGP position and orientation with respect to the CRC. This involves solving an optimization problem over the difference function defined above.

Assume the coordinates of the current and target robot positions are given in terms of the coordinate system with the origin located at the CoG of the CRC, we define the best matching TGP with respect to the CRC in the triplet $(\delta^*, \theta^*, \psi^*)$ minimizing the difference function $\mathbf{J}^i(\delta, \theta, \psi)$ in (5):

$$\mathbf{J}^i(\delta^*, \theta^*, \psi^*) = \min_{\delta \in \Delta, \theta \in [-\pi, \pi), \psi \in \mathbf{R}^2} \mathbf{J}^i(\delta, \theta, \psi). \tag{6}$$

The optimization problem (6) can be broken in two steps:

Step 1: First fix a particular $\delta \in \Delta$ and minimize for the orientation and location to find the best (θ^*, ψ^*)

$$\begin{aligned} \mathbf{V}^i(\delta) &= \|Z - \Pi(\theta^*)\Gamma_\delta - \Psi^*\|^2 \\ &= \min_{\theta \in [-\pi, \pi), \psi \in \mathbf{R}^2} \mathbf{J}^{i,\delta}(\theta, \psi) \\ &= \min_{\theta \in [-\pi, \pi), \psi \in \mathbf{R}^2} \|Z - \Pi(\theta)\Gamma_\delta - \Psi\|^2. \end{aligned} \tag{7}$$

Step 1 optimization can be solved explicitly as shown in the following theorem.

Theorem 3.1: Given the coordinates Z and Γ of the CRC and TGP in the local coordinate system of robot $\mathcal{R}_i$ and a given correspondence $\delta \in \Delta$, the minimum of the difference function $\mathbf{J}^{i,\delta}$ is

$$(\theta^*(\delta), \psi^*(\delta)) = \left(-\arctan\left(\frac{B^\delta}{A^\delta}\right), [0, 0]' \right), \tag{8}$$

where

$$\begin{aligned} A^\delta &= \sum_{j=1}^{n} (x_j \alpha_{\delta_j} + y_j \beta_{\delta_j}), \\ B^\delta &= \sum_{j=1}^{n} (x_j \beta_{\delta_j} - y_j \alpha_{\delta_j}). \end{aligned}$$

Proof: We find the minimum point for the difference function

$$\begin{aligned} \mathbf{J}^{i,\delta}(\theta, \psi) &= \|Z - \Pi(\theta)\Gamma_\delta - \Psi\|^2 \\ &= \sum_{j=1}^{n} \{[(x_j - (\cos(\theta)\alpha_{\delta_j} - \sin(\theta)\beta_{\delta_j}) - \alpha]^2 \\ &\quad + [y_j - (\sin(\theta)\alpha_{\delta_j} + \cos(\theta)\beta_{\delta_j}) - \beta]^2\} \end{aligned}$$

by solving the following equations:

$$\begin{cases} \dfrac{\partial \mathbf{J}^{i,\delta}}{\partial \alpha} = -2\displaystyle\sum_{j=1}^{n} (x_j - \cos(\theta)\alpha_{\delta_j} + \sin(\theta)\beta_{\delta_j} - \alpha) = 0 \\ \dfrac{\partial \mathbf{J}^{i,\delta}}{\partial \beta} = -2\displaystyle\sum_{j=1}^{n} (y_j - \sin(\theta)\alpha_{\delta_j} - \cos(\theta)\beta_{\delta_j} - \beta) = 0 \\ \dfrac{\partial \mathbf{J}^{i,\delta}}{\partial \theta} = 2\displaystyle\sum_{j=1}^{n} [-\alpha_{\delta_j}\sin(\theta) - \beta_{\delta_j}\cos(\theta)]\alpha \\ \qquad + [-\beta_{\delta_j}\sin(\theta) + \alpha_{\delta_j}\cos(\theta)]\beta + (x_j\alpha_{\delta_j} \\ \qquad + y_j\beta_{\delta_j})\sin(\theta) + (x_j\beta_{\delta_j} - y_j\alpha_{\delta_j})\cos(\theta) = 0 \end{cases} \tag{9}$$

From the first two equations, we obtain ψ in terms of θ

$$\begin{cases} \alpha = \dfrac{1}{n}\displaystyle\sum_{j=1}^{n} \beta_{\delta_j}\sin(\theta) - \frac{1}{n}\sum_{j=1}^{n} \alpha_{\delta_j}\cos(\theta) + \frac{1}{n}\sum_{j=1}^{n} x_j = 0 \\ \beta = -\dfrac{1}{n}\displaystyle\sum_{j=1}^{n} \alpha_{\delta_j}\sin(\theta) - \frac{1}{n}\sum_{j=1}^{n} \beta_{\delta_j}\cos(\theta) + \frac{1}{n}\sum_{j=1}^{n} y_j = 0 \end{cases} \tag{10}$$

It shows that the origin is the unique critical point of the quadratic function $\mathbf{J}^{i,\delta}(\theta, \cdot)$ for given δ and θ. Hence the minimum is $\psi^*(\delta) = [0, 0]'$.

From the third equation of (9), we obtain the equation solving the best rotation angle θ^* to be

$$A^\delta \sin(\theta) + B^\delta \cos(\theta) = 0. \tag{11}$$

Therefore, the unique solution in $(-\pi, \pi)$ is

$$\theta^*(\delta) = -\arctan\left(\frac{B^\delta}{A^\delta}\right). \tag{12}$$

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

Step 2: Find the optimal correspondence δ^* by minimizing over $\delta \in \Delta$

$$\begin{aligned} \mathbf{V}^i &= \min_{\delta\in\Delta} \mathbf{V}^i(\delta) \\ &= \min_{\delta\in\Delta} \|Z - \Pi(\theta^*(\delta))\Gamma_\delta\|. \end{aligned} \tag{13}$$

Since Δ is a finite set, the optimal δ^* can be found by a simple compare and search method. The minimum is δ^*. It can be seen that the minimum point of the two-step optimization $(\delta^*, \theta^*(\delta^*), \psi^*(\delta^*))$ is the minimum of the difference function $\mathbf{J}^i$ in (5).

Remark 3.2: Theorem 3.1 implies that the best location for the TGP is that the CoG of the TGP should be located at the CoG of the CRC. Hence from now on, we assume, without loss of generality, that the CoG of the TGP is always at the CoG of the CRC. The only computation for the robot is to rotate the TGP at this convenient location according to $\theta^*(\delta)$ to find the best TGP orientation. This is equivalent to optimizing the difference function $\mathbf{J}^i(\delta, \theta)$ in (4). We denote the minimum to be $(\delta^*, \theta^*(\delta^*))$.

3.3. Determining the move strategy

The solution to the optimization problem provides the optimizing robot with the *best matching* TGP given in terms of the post-rotation target robot positions and orientation $\hat{\Gamma} = \Pi(\theta^*(\delta^*))\Gamma$ and the best correspondence δ^* between the current and target robot positions. Here $\hat{\Gamma}$ and δ^* determine a *move strategy*

$$Z_j \to \hat{\gamma}_{\delta_j^*}, \; j = 1, 2, \ldots, n, \tag{14}$$

that is, the current robot $\mathcal{R}_j$ should move toward the point of the target robot position $\hat{\gamma}_{\delta_j^*}$. This move strategy is based on the local computation of the robot $\mathcal{R}_i$ and is independent of the computation of all other active robots. The key aspect that makes the optimizations useful to our robot formation problem is that the simultaneous computations of all active robots will be the same if expressed in a global coordinate system. That means all the move strategies as (14) for individual robots based on their local computations are the same if looked from a global coordinate system. This idea of distributed computation resulting in a global consequence is fundamental to our autonomous robot formation solution as stated in the following theorem.

Theorem 3.3: Assume that two robots $\mathcal{R}_i, i = a, b$ are active and perform the above optimization processes independently with unique minimum in their own local coordinate systems $\mathcal{Z}_i, i = a, b$. Assume under an ordering convention $[\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n]$ and $[P_1, P_2, \ldots, P_n]$, the optimal correspondences are $\delta_i^*, i = a, b$ and under a third global coordinate system $\mathcal{Z}$, the post-rotation target robot positions are $\hat{\Gamma}^i = [\hat{\gamma}_1^i, \hat{\gamma}_2^i, \ldots, \hat{\gamma}_n^i], i = a, b$, then it holds

$$\delta_a^* = \delta_b^* \tag{15}$$

and

$$\hat{\Gamma}^a = \hat{\Gamma}^b. \tag{16}$$

Proof: From Remark 3.2, we can assume the origins of the two local coordinate systems to be the same at the CoG of the CRC and the TGP. We can also assume without loss of generality $\mathcal{Z}_b$ to be the third global coordinate system $\mathcal{Z}$. We need to show that the coordinates of post-rotation target robot positions expressed in $\mathcal{Z}_b$ are the same.

Under the ordering system stated in Theorem 3.3, assume that the coordinates of current robot positions are vectors $Z^i, i = a, b$. Since they are describing the same positions, it must hold

$$Z^b = \Pi(\phi)Z^a \tag{17}$$

for some ϕ which designates the orientation difference between $\mathcal{Z}^a$ and $\mathcal{Z}^b$. The pre-rotation coordinates of the target robots positions in the TGP are designated by $\Gamma^i, i = a, b$. Since these are coordinates of the same TGP with different orientation, there exists a φ such that

$$\Gamma^b = \Pi(\varphi)\Gamma^a. \tag{18}$$

Let us denote $\check{Z}^a$ and $\check{\Gamma}^a$ the coordinates of the current and pre-rotation target positions, respectively, of the first robot $\mathcal{R}^a$ expressed in $\mathcal{Z}_b$. Since it only involves a rotation from $\mathcal{Z}_a$ to $\mathcal{Z}_b$, we have from (17)

$$\check{Z}^a = \Pi(\phi)Z^a \tag{19}$$

and

$$\check{\Gamma}^a = \Pi(\phi)\Gamma^a. \tag{20}$$

We obtain from (17)–(20)

$$\check{Z}^a = Z^b \tag{21}$$

and

$$\check{\Gamma}^a = \Pi(\phi - \varphi)\Gamma^b. \tag{22}$$

We assume $(\delta_i^*, \theta_i^*(\delta_i^*)), i = a, b$ are the unique minimum of the optimization performed in $\mathcal{Z}_i, i = a, b$. The post-rotation coordinates of the target robot positions are given by

$$\Gamma_i^* = \Pi(\theta_i^*(\delta_i^*))\Gamma^i, i = a, b \tag{23}$$

in their own local coordinate systems. From (19) Γ_a^* is expressed in $\mathcal{Z}_b$ as

$$\check{\Gamma}_a^* = \Pi(\phi)\Gamma_a^*.$$

To prove (16), we need to show

$$\check{\Gamma}_a^* = \Gamma_b^* \tag{24}$$

in our context of $\mathcal{Z} = \mathcal{Z}_b$.

Define an optimization problem in $\mathcal{Z}_b$ with the cost

$$\check{\mathbf{J}}^b(\check{\delta}, \check{\theta}) = \|\check{Z} - \Pi(\check{\theta})\check{\Gamma}_{\check{\delta}}\|^2$$

with initial configuration $\check{Z}^a$ and $\check{\Gamma}^a$. Let us denote the unique minimum to be $(\check{\delta}_a^*, \check{\theta}_a^*(\check{\delta}_a^*))$. For a given $\check{\delta}$, we get from (21)

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

and (22)

$$\begin{aligned}\check{\theta}^*(\check{\delta}) &= -\arctan\left(\frac{\sum_{j=1}^n \left(\check{Z}_j^a\right)' \Xi \check{\gamma}_{\check{\delta}_j}^a}{\sum_{j=1}^n \left(\check{Z}_j^a\right)' \check{\gamma}_{\check{\delta}_j}^a}\right)\\ &= -\arctan\left(\frac{\sum_{j=1}^n \left(Z_j^b\right)' \Xi \Pi(\phi-\varphi)\gamma_{\check{\delta}_j}^b}{\sum_{j=1}^n \left(Z_j^b\right)' \Pi(\phi-\varphi)\gamma_{\check{\delta}_j}^b}\right)\\ &= -\arctan\left(-\frac{\sum_{j=1}^n \left(Z_j^b\right)' \Gamma_{\check{\delta}_j}^b \sin(\varphi-\phi) - \sum_{j=1}^n \left(Z_j^b\right)' \Xi \Gamma_{\check{\delta}_j}^b \cos(\varphi-\phi)}{\sum_{j=1}^n \left(Z_j^b\right)' \Gamma_{\check{\delta}_j}^b \cos(\varphi-\phi) + \sum_{j=1}^n \left(Z_j^b\right)' \Xi \Gamma_{\check{\delta}_j}^b \sin(\varphi-\phi)}\right)\\ &= -\arctan\left(-\frac{\tan\left(\theta_b^*(\check{\delta})\right) + \tan(\varphi-\phi)}{1-\tan\left(\theta_b^*(\check{\delta})\right)\tan(\varphi-\phi)}\right)\end{aligned}$$

where

$$\Xi = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Hence we obtain

$$\tan(\check{\theta}^*(\check{\delta})) = \tan(\theta_b^*(\check{\delta}) + (\varphi - \phi)).$$

In addition,

$$\check{\delta}^* = \delta_b^*, \check{\theta}^*(\check{\delta}^*) = \theta_b^*(\delta_b^*) + (\varphi - \phi). \tag{25}$$

On the other hand, from (19) and (20), we have

$$\begin{aligned}\check{\theta}^*(\check{\delta}) &= -\arctan\left(\frac{\sum_{j=1}^n \left(\check{Z}_j^a\right)' \Xi \check{\gamma}_{\check{\delta}_j}^a}{\sum_{j=1}^n \left(\check{Z}_j^a\right)' \check{\gamma}_{\check{\delta}_j}^a}\right)\\ &= -\arctan\left(\frac{\sum_{j=1}^n \left(Z_j^a\right)' \Pi'(\phi) \Xi \Pi(\phi) \gamma_{\check{\delta}_j}^a}{\sum_{j=1}^n \left(Z_j^a\right)' \Pi'(\phi) \Pi(\phi) \gamma_{\check{\delta}_j}^a}\right)\\ &= -\arctan\left(\frac{\sum_{j=1}^n \left(Z_j^a\right)' \Xi \gamma_{\check{\delta}_j}^a}{\sum_{j=1}^n \left(Z_j^a\right)' \gamma_{\check{\delta}_j}^a}\right)\\ &= -\arctan(-\tan(\theta_a^*(\check{\delta})))\\ &= \theta_a^*(\check{\delta}).\end{aligned}$$

Hence, it must hold

$$\check{\delta}^* = \delta_a^*, \check{\theta}^*(\delta^*) = \theta_a^*(\delta_a^*). \tag{26}$$

From (25) and (26), we get

$$\delta_a^* = \delta_b^*,$$

which proves (15). Furthermore, we have

$$\begin{aligned}\check{\Gamma}_a^* &= \Pi(\phi)\Gamma_a^*\\ &= \Pi(\phi)\Pi(\theta_a^*(\delta_a^*))\Gamma_a\\ &= \Pi(\phi)\Pi(\theta_b^*(\delta_b^*) + (\varphi - \phi))\Pi^{-1}(\varphi)\Gamma^b\\ &= \Pi(\theta_b^*(\delta_b^*))\Gamma^b\\ &= \Gamma_b^*\end{aligned}$$

which is (24) and completes the proof.

Theorem 3.3 shows that the move destinations obtained from local computations are the same in a global coordinate system as claimed. We have the following result which says that the optimal cost is the same for all robots obtained from local computations.

Corollary 3.4: If the minimum for the optimizations performed in local coordinate systems $\mathcal{Z}_i, i = a, b$ for any two robots $\mathcal{R}_i, i = a, b$ at time t is $\mathbf{V}^i, i = a, b$, then

$$\mathbf{V}^a = \mathbf{V}^b. \tag{27}$$

Proof: From Theorem 3.3, we know that the optimal correspondence $\delta_i^*, i = a, b$ and the post-rotation TGP positions $\check{\Gamma}_i, i = a, b$ are the same under a common coordinate system. It implies that the relative positions between the CRC and the post-rotation TGP are the same for robots $\mathcal{R}_i, i = a, b$. According to the definition of the difference function $\mathbf{J}^i$, we know that $\mathbf{V}^a = \mathbf{V}^b$. From now on, we can omit the superscripts of $\mathbf{V}^i$ for different $\mathcal{R}_i$ and write them as common $\mathbf{V}$. We now show that any movement according to the move strategy leads to a robot configuration closer to the TGP.

Theorem 3.5: Assume the active robots move according to the move strategy (14) obtained from the distributed optimizations. If $\mathbf{V}_k, k = 1, 2, \ldots$ are the minimum of the difference function (5) calculated in the local coordinate system of any active robot at time k, then

$$\mathbf{V}_k \geq \mathbf{V}_{k+1}, \forall k \geq 0. \tag{28}$$

Proof: Let us take $\mathcal{V}_k$ as the value of the difference function immediately after the move according to the move strategy (14) at time k. Note that immediately after the move, the local coordinate system, the correspondence, the CoG and orientation of the TGP are the same as before the move. It must hold

$$\mathbf{V}_k \geq \mathcal{V}_k.$$

Now $\mathbf{V}_{k+1}$ is the minimum value of the difference function (5) minimized over all possible correspondences, CoG and orientation of the TGP with respect to the CRC at time $k+1$ which is the post-move robot configuration at time k. Hence, it must hold

$$\mathcal{V}_k \geq \mathbf{V}_{k+1}.$$

Hence, we get

$$\mathbf{V}_k \geq \mathbf{V}_{k+1}, k = 1, 2, \ldots,$$

which completes the proofs.

4. Algorithms

We suggest two algorithms for the TGP formation problem based on the above move strategy. The first algorithm achieves the TGP in finite steps for synchronous robots. The other one is a convergence algorithm for the asynchronous robots.

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

4.1. Formation algorithm for synchronous robots

Algorithm: *FORM-TGP–BY-SYNCHRONOUS-ROBOTS*

At every step $t = t_0, t_1, t_2, \ldots,$ each robot $\mathcal{R}_i$ *Do*

(1) *LOOK AND RECORD*: Establish the CRC by recording the current robot positions $Z = [Z_1, Z_2, \ldots, Z_n]$ from the sensor readings in the local coordinate system $\mathcal{Z}_i$. At the same time, establish an understanding of the TGP by denoting the coordinates for the target robot positions $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]$.

(2) *THINK to obtain the best move strategy*:
- Do translations such that the origin of the local coordinate system is at the CoG of the CRC and the TGP.
- Optimize the cost function (4) and obtain the minimum $(\delta^*, \theta^*(\delta^*))$ and the optimum cost $\mathbf{J}^i(\delta^*, \theta^*(\delta^*))$. Rotate the TGP to the optimal orientation with coordinates $\Gamma^* = \Pi(\theta^*(\delta^*))\Gamma$ according to $\theta^*(\delta^*)$. The best action for a robot $\mathcal{R}_i$ is to move from the current position Z_i to $\gamma^*_{\delta^*_i} = \Pi(\delta^*)\gamma_{\delta^*_i}$.

(3) *If* $\mathbf{J}^i(\delta^*, \theta^*(\delta^*)) = 0$, then *STOP*, return with TGP achieved and output the coordinates of the robot positions of the CRC and the TGP. *Else* make a *MOVE* to $\gamma^*_{\delta^*_i}$ along the line connecting Z_i and $\gamma^*_{\delta^*_i}$ and go back to step 1.

Theorem 4.1: For given initial robot configuration described by the position vector Z and the TGP described by the target robot position vector Γ in any local coordinate systems, the Algorithm *FORM-TGP–BY-SYNCHRONOUS-ROBOTS* achieves the TGP in finite steps N; furthermore

$$N \le \frac{\mathbf{V}_0}{\epsilon} + 1. \tag{29}$$

Proof: From Theorem 3.3, we know the minimum value $\mathbf{V}_0$ of the difference function (4) for the initial configuration and TGP is independent of the local coordinate system. Let us denote

$$d^k(Z_i, \gamma^*_{\delta^*_i}), i = 1, 2, \ldots, n$$

the distances between the current robot positions Z^k_i and their destination positions $\gamma^*_{\delta^*_i}$ at time k. Now if $\max_{i\in\{1,2,\cdots,n\}} d^k(Z_i, \gamma^*_{\delta^*_i}) \le \epsilon$, then the robots achieve the TGP at time $k+1$ by directly moving to the destination positions. Hence, only $\max_{i\in\{1,2,\ldots,n\}} d^{N-1}(Z_i, \gamma^*_{\delta^*_i}) \le \epsilon$. While for all $0 \le k \le N-1$, it holds $\max_{i\in\{1,2,\cdots,n\}} d^k(Z_i, \gamma^*_{\delta^*_i}) > \epsilon$, i.e. there is at least one robot that moves a distance at least ϵ. From Theorem 3.5 we know

$$\mathbf{V}_k \ge \mathbf{V}_{k+1} + \epsilon, 0 \le k \le N-1.$$

We obtain

$$\mathbf{V}_0 - \epsilon * (N-1) \ge \mathbf{V}_{N-1} \ge 0,$$

which is

$$N \le \frac{\mathbf{V}_0}{\epsilon} + 1.$$

4.2. Convergence algorithm for asynchronous robots

From Theorem 3.5, we know that if all the active robots move according to the move strategy (14), then the sequence $\{\mathbf{V}_0, \mathbf{V}_1, \ldots\}$ is nonincreasing. Hence, there exists $\mathbf{V}_\infty \ge 0$ such that

$$\mathbf{V}_t \to \mathbf{V}_\infty, t \to \infty.$$

If $\mathbf{V}_\infty = 0$, then the current configuration sequence converges to the TGP. Unfortunately, a simple algorithm based on the move strategy (14) like *FORM-TGP–BY-SYNCHRONOUS-ROBOTS* does not necessary guarantee $\mathbf{V}_\infty = 0$ for asynchronous autonomous robots. The reason is that as time goes on, the decrease of values $\mathbf{V}_t$ may become too small to reduce it to 0. However, we show that we are able to revise the algorithm *FORM-TGP–BY-SYNCHRONOUS-ROBOTS* to obtain a convergence algorithm for asynchronous autonomous robots. The key idea is that if the amount of decrease is too small, we hold it until the robot which has a larger move distance becomes active.

Algorithm: *CONVERGE-TGP–BY-ASYNCHRONOUS-ROBOTS*

Set a convergence sequence $\{\xi_l > 0, l = 1, 2\cdots\}$ and $\xi_l \downarrow 0$. When $t = t_0$, set $n_t = 1$. At every step $t = t_0, t_1, t_2, \ldots,$ *If* a robot $\mathcal{R}_i$ is active *Do* the following steps:

(1) *LOOK AND RECORD*: Establish the CRC by recording the current robot positions $Z = [Z_1, Z_2, \ldots, Z_n]$ from the sensor readings in the local coordinate system $\mathcal{Z}_i$. At the same time, establish an understanding of the TGP (target geometric pattern) by denoting the coordinates for the target robot positions $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]$.

(2) *THINK to obtain the best move strategy*:
- Do translations such that the origin of $\mathcal{Z}_i$ is at the CoG of the CRC and the TGP.
- Optimize the cost function (4) and obtain the minimum $(\delta^*, \theta^*(\delta^*))$ and the optimum cost $\mathbf{V}_t = \mathbf{J}^i(\delta^*, \theta^*(\delta^*))$. Rotate the TGP to the optimal orientation with coordinates $\Gamma^* = \Pi(\theta^*(\delta^*))\Gamma$ according to $\theta^*(\delta^*)$. Calculate the distances between the current robot positions and destination robot positions

$$(Z_j(t), \gamma^*_{\delta^*_j}(t)) = (x_j(t) - \alpha^*_{\delta^*_j}(t))^2 + (y_j(t) - \beta^*_{\delta^*_j}(t))^2,$$
$$j = 1, 2, \ldots, n.$$

(3) *MOVE*: *If* $\mathbf{V}_t = 0$, then *Stop*, return with *TGP achieved* and output the coordinates of the robot positions of the CRC and the TGP.

Else
If $\max_{j\in\{1,2,\cdots,n\}} d(Z_j(t), \gamma^*_{\delta^*_j}(t)) < \xi_{n_t}$, *Then Don't Move, Go to 1 for* $t+1$ *and Set* $n_{t+1} = n_t + 1$.
Else if $d(Z_i(t), \gamma^*_{\delta^*_i}(t)) < \xi_{n_t}$, *Then Don't Move, Go to 1 for* $t+1$ *and* $n_{t+1} = n_t$.
Else make a *MOVE* to $\gamma^*_{\delta^*_i}$ with distance $\lambda_t = \min\{\epsilon, d(Z_j(t), \gamma^*_{\delta^*_j}(t))\}$ along the line connecting Z_i and $\gamma^*_{\delta^*_i}$ and Go to 1 for $t+1$ with $n_{t+1} = n_t$.

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

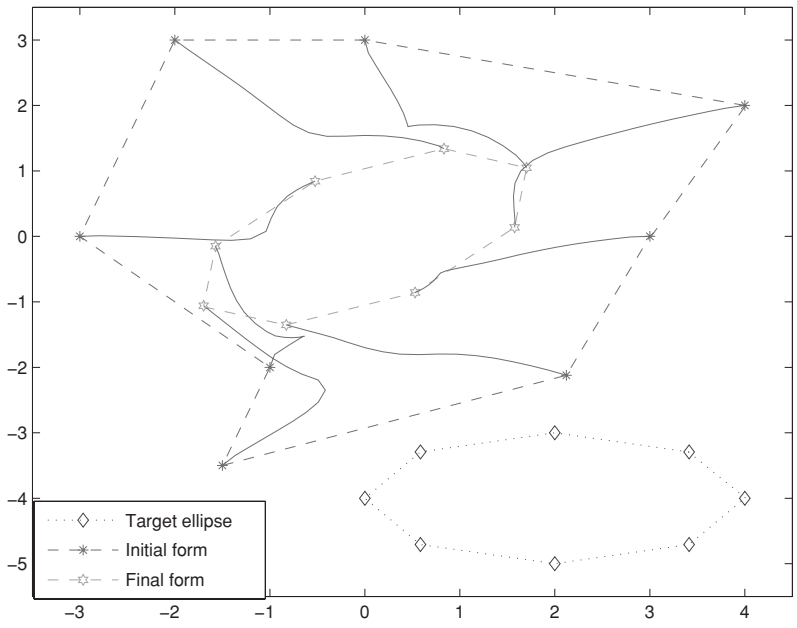


Fig. 5. Synchronous formation of an ellipse.

Theorem 4.2: The algorithm *CONVERGE-TGP–BY-ASYNCHRONOUS-ROBOTS* is a convergence algorithm for a group of n asynchronous autonomous robots.

Proof: Assume at time t, there are m_t active robots. If $m_t = 0$, the algorithm does nothing, we can assume without loss of generality $m_t > 0$. We show if the robots execute the algorithm, there exists some finite number $p_t \geq 0$ such that

$$\max_{j\in\{1,2,\cdots,n\}} d(Z_j(t+p_t), \gamma^*_{\delta^*_j}(t+p_t)) \leq \xi_{n_{(t+p_t)}}. \tag{30}$$

If at time t, it holds $\max_{j\in\{1,2,\ldots,n\}} d(Z_j(t), \gamma^*_{\delta^*_j}(t)) < \xi_{n_t}$, then obviously it can be taken as $p_t = 0$. Otherwise there must be a robot $\mathcal{R}_i$ such that $d(Z_i(t), \gamma^*_{\delta^*_i}(t)) \geq \xi_{n_t}$. We show it must hold

$$\mathbf{V}_{t+q_t} \leq \mathbf{V}_t - \min\{\epsilon, \xi_{n_t}\}$$

for some $q_t \geq 1$. If $\mathcal{R}_i$ is active at time t, it moves a distance of

$$\lambda_t = \min\{\epsilon, d(Z_i(t), \gamma^*_{\delta^*_i}(t))\} \geq \min\{\epsilon, \xi_{n_t}\},$$

hence take $q_t = 1$ and

$$\mathbf{V}_{t+1} \leq \mathbf{V}_t - \min\{\epsilon, \xi_{n_t}\}.$$

Otherwise it implies $d(Z_j(t), \gamma^*_{\delta^*_j}(t)) < \xi_{n_t}$ for any active robots $\mathcal{R}_j$. According to the algorithm, no active robot moves at time t.

From the assumption of the autonomous movement of the robots, every robot must be activated infinite times. There must exist $q_t - 1 > 0$ such that at least one robot $\mathcal{R}_k$ is activated at time $q_t - 1$ and $d(Z_k(t+q_t-1), \gamma^*_{\delta^*_k}(t+q_t-1)) \geq \xi_{n_{(t+q_t-1)}} = \xi_{n_t}$. Now it moves a distance of

$$\begin{aligned}\lambda_{t+q_t-1} &= \min\{\epsilon, d(Z_k(t+q_t-1), \gamma^*_{\delta^*_k}(t+q_t-1))\}\\ &\geq \min\{\epsilon, \xi_{n_{(t+q_t-1)}}\}\\ &= \min\{\epsilon, \xi_{n_t}\}.\end{aligned}$$

Hence,

$$\mathbf{V}_{t+q_t} \leq \mathbf{V}_{t+q_t-1} - \min\{\epsilon, \xi_{n_{(t+q_t-1)}}\} = \mathbf{V}_t - \min\{\epsilon, \xi_{n_t}\}.$$

From the algorithm, we know that before (30) is satisfied, n_t does not change. $\mathbf{V}_t$ will decrease by $\min\{\epsilon, \xi_{n_t}\}$ for q_t steps until (30) holds.

Now the above argument shows that by executing the algorithm, there exists a subsequence $\{t_l, l = 1, 2, \ldots\}$ from the sequence $t = 1, 2, \ldots$ such that

$$\max_{j\in\{1,2,\cdots,n\}} d(Z_j(t_l), \gamma^*_{\delta^*_j}(t_l)) \leq \xi_{n_{t_l}}.$$

Then

$$\mathbf{V}_{t_l} \leq n * \xi_{n_{t_l}} \downarrow 0, \text{ as } l \uparrow \infty.$$

Obviously it holds $\mathbf{V}_t \downarrow 0$ which proves that the TGP is achieved asymptotically.

5. Numerical Simulations

We develop simulations based on the suggested synchronous and asynchronous formation control algorithms. The simulation results are shown in Figs. 5–7.

The aim is to ask a group of eight autonomous robots to form an ellipse somewhere in the plane as shown in Figs. 5 and 6. It is assumed that the maximum distance a robot can travel at one step is 0.2. Figure 5 shows the formation by the synchronous algorithm. Figure 6 is the simulation result

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

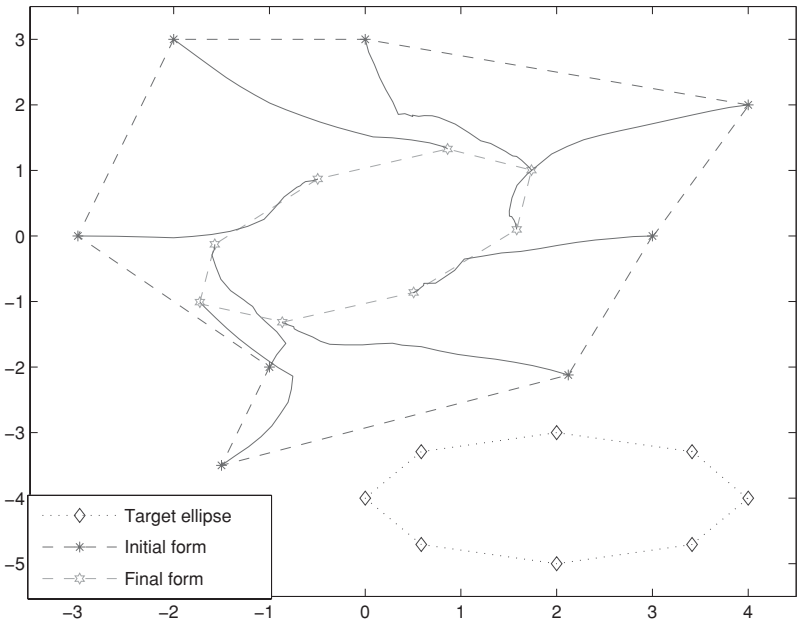


Fig. 6. Asynchronous formation of an ellipse.

using asynchronous algorithm from the same initial robot configuration. In the simulation of asynchronous formation, we set the probability to be 0.6 for any robot to be active at any step. Although the formation trajectories and the final ellipses are quite similar for synchronous and asynchronous cases, a closer scrutiny reveals their difference as expected.

Figure 7 shows the optimal costs $(\mathbf{V}_1, \mathbf{V}_2, \ldots,)$ history recorded in the formation process. As expected, both of them decrease and the synchronous formation decreases faster. In our simulation, the optimal cost decreases to 0 in 17 steps for the synchronous formation, which implies that the formation is achieved in 17 steps. However, for the asynchronous formation, the optimal cost decreases to 0.2238 in 31 steps and keeps decreasing.

6. Concluding Remarks

We have developed a new distributed solution to the formation problem for autonomous robots. The key to the solution is that every robot solves an optimization problem independently at every step to find the best matching orientation and correspondence between the CRC and the TGP. Every movement based on the strategy from the optimization results guarantees the CRC more similar with the TGP until it is completely achieved.

We have divided the optimization process into two steps. The first step is to find the best orientation of the TGP relative to the CRC which can be solved explicitly. However, for the second step, which is to find the best correspondence, we have to search a finite set of n elements. Practically, it is only possible for small n. How to narrow down the search range

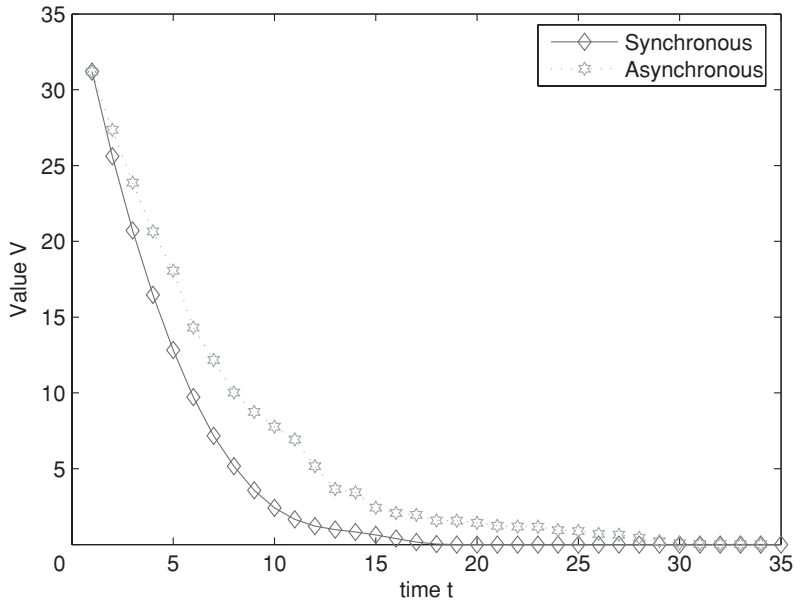


Fig. 7. Optimal costs for synchronous and asynchronous formations.

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press

to save computation is important and is an interesting topic of future research.

Acknowledgment
This research was supported by Australian Research Council.

References

1. H. Ando, Y. Oasa, I. Suzuki and M. Yamashita, "A distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Trans. Robot. Autom.* **15**(5), 818–828 (1999).
2. T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.* **14**(6), 926–939 (1998).
3. M. Cieliebak, P. Flocchini, G. Prencipe and N. Santoro, "Solving the Robots Gathering Problem," In *LNCS* 2719 (2003) pp. 1181–1196.
4. S. Darbha and K. R. Rajagopal, "Intelligent cruise control systems and traffic flow stability," *Transp. Res.* Part C **7**, 329–352 (1999).
5. J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control* **49**(9), 1465–1476 (2004).
6. P. Flocchini, G. Prencipe, N. Santoro and P. Widmayer, "Hard Task for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots," *ISAAC'99* (1999) LNCS, vol. 1741, pp. 93–102.
7. V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Autom. Control.* **48**(4), 692–697 (2003).
8. F. Giulietti, L. Pollini and M. Innocenti, "Autonomous formation flight," *IEEE Control Syst. Mag.* **20**(6), 34–44 (2000).
9. A. Jadbaie, J. Lin and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control* **48**(6), 988–1001 (2003).
10. Z. Lin, B. Francis and M. Maggiore, "Necessary and sufficient graphical conditions of formation control of unicycles," *IEEE Trans. Autom. Control* **50**(1), 121–127 (2005).
11. Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Trans. Autom. Control* **49**(1), 30–44 (2004).
12. S. Martinez, J. Cortes and F. Bullo, "Motion coordination with distributed information," *IEEE Control Syst. Mag.* **27**(4), 75–88 (Aug. 2007).
13. A. Savkin, "Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model," *IEEE Trans. Autom. Control* **49**(6), 981–983 (2004).
14. K. Sugihara and I. Suzuki, "Distributed Motion Coordiantion of Multiple Mobile Robots," *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, Philadelphia, PA (1990) pp. 138–143.
15. K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *J. Robot. Syst.* **13**, 127–139 (1996).
16. I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric pattern," *SIAM J. Comput.* **28**(4), 1347–1363 (1999).
17. K. Warburton and J. Lazarus, "Tendency-distance models of social cohesion in animal groups," *J. Theor. Bio.* **150**, 473–488 (1991).

https://doi.org/10.1017/S0263574710000366 Published online by Cambridge University Press