Journal of Southeast University (English Edition) Vol. 31, No. 1, pp. 74 − 79 Mar. 2015 ISSN 1003—7985

Consensus control for multi-agents in a non-rectangular bounded space: algorithm and experiments

Zhu Dezheng Tian Yuping

(School of Automation, Southeast University, Nanjing 210096, China)

Abstract: Aiming for the coordinated motion and cooperative control of multi-agents in a non-rectangular bounded space, a velocity consensus algorithm for the agents with double-integrator dynamics is presented. The traditional consensus algorithm for bounded space is only applicable to rectangular bouncing boundaries, not suitable for non-rectangular space. In order to extend the previous consensus algorithm to the non-rectangular space, the concept of mirrored velocity is introduced, which can convert the discontinuous real velocity to continuous mirrored velocity, and expand a bounded space into an infinite space. Using the consensus algorithm, it is found that the mirrored velocities of multi-agents asymptotically converge to the same values. Because each mirrored velocity points to a unique velocity in real space, it can be concluded that the real velocities of multi-agents also asymptotically converge. Finally, the effectiveness of the proposed consensus algorithm is examined by theoretical proof and numerical simulations. Moreover, an experiment is performed with the algorithm in a real multi-robot system successfully.

Key words: multi-agent system; consensus; non-rectangular bounded space; mirrored velocity

doi: 10.3969/j.issn.1003 −7985.2015.01.013

Over the past few years, the distributed cooperative control of multiple autonomous agents has found applications in many motion control problems, such as flocking[1-3], formation control[4] and rendezvous[5]. The consensus algorithm, as one of the most popular algorithms for cooperation control, has been studied in many circumstances for multi-agents with single-integrator dynamics[6] and double-integrator dynamics[7-8].

Many biological social flocking behaviors occur in bounded spaces, e.g., crowded individuals in escaping panics and public traffic systems. Also, many applications in actual engineering aspects occur in a bounded space. For example, when multiple service robots complete a task, such as sweeping the floor, they should maintain a formation in a bounded space. Through collaboration, multiple robots can work more efficiently than a single robot. So the research on consensus behaviors in a bounded space is of particular interest for engineering applications. Although some of the previous multi-agent control algorithms can be applied in a rectangular space[9-10], they do not automatically work in the irregular space. Therefore, there is a need to generalize the algorithm to work out the problems caused by the bouncing boundaries of a bounded space. In this paper, we introduce the concept of the mirrored velocity to make the algorithm work in such spaces. Moreover, the algorithm is successfully implemented in a group of real robots.

In this paper, we first give the system model and problem statements, and present the construction of the multi-agent consensus algorithm and the proof of control law. Then, we provide the simulation results and experimental validation to examine the effectiveness of the consensus algorithm.

1 Preliminaries and Problem Statements

Based on the algebraic graph theory, $G=(V, \varepsilon)$ denotes a weighted graph of order n. Here, $V=\{1, 2, \dots, n\}$ denotes the set of nodes, and $\varepsilon \in V\times V$ denotes the set of edges. Let $\boldsymbol{A}=(a_{ij})\in \mathbf{R}^{n\times n}(a_{ij}\geqslant 0)$ denote a weighted adjacency matrix. For an undirected graph, the adjacency matrix $\boldsymbol{A}$ is symmetric. If $(i, j)\in\varepsilon$, then $a_{ij}=1$; otherwise $a_{ij}=0$. In this paper, self edges are not allowed, i.e., $a_{ii}=0$.

We consider a group of agents moving in a regular triangle plane with a Cartesian coordinate system x-y. The origin is established in the left vertex of the bottom, and the x axis points to the right along the bottom side. Without loss of generality, we suppose the time sequence $0<\tau(1)<\tau(2)<\dots$ to be the instantaneous time when the agents touch the wall. Each agent has the following dynamic equation:

$$\left.\begin{aligned}\dot{\boldsymbol{p}}_i(t)&=\boldsymbol{v}_i(t)\\ \dot{\boldsymbol{v}}_i(t)&=\boldsymbol{u}_i(t)-2\boldsymbol{\Delta}_i(t)\langle \boldsymbol{v}_i(t-), \boldsymbol{n}\rangle \boldsymbol{n}\end{aligned}\right\} \tag{1}$$

where $i\in V$ and $\boldsymbol{p}_i=[p_i^x, p_i^y]^{\mathrm{T}}$, $\boldsymbol{v}_i=[v_i^x, v_i^y]^{\mathrm{T}}$, $\boldsymbol{u}_i=[u_i^x, u_i^y]^{\mathrm{T}}\in\mathbf{R}^2$ are the position, the velocity, and the control

Received 2014-08-10.

Biographies: Zhu Dezheng (1989—), male, graduate; Tian Yuping (corresponding author), male, doctor, professor, yptian@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 61273110), the Specialized Fund for the Doctoral Program of Higher Education (No. 20130092130002).

Citation: Zhu Dezheng, Tian Yuping. Consensus control for multi-agents in a non-rectangular bounded space: algorithm and experiments [J]. Journal of Southeast University (English Edition), 2015, 31(1): 74 −79. [doi: 10.3969/j.issn.1003 −7985.2015.01.013]

input, respectively; $\boldsymbol{n}$ denotes the unit normal vector of the regular triangular side; $\boldsymbol{v}_i(t-)=\lim_{s\to t-}\boldsymbol{v}_i(s)$ and $\boldsymbol{v}_i(t+)=\lim_{s\to t+}\boldsymbol{v}_i(s)$ are the left and the right limits of $\boldsymbol{v}_i(s)$ at $s=t$; $\langle \boldsymbol{v}_i(t-),\boldsymbol{n}\rangle$ denotes the inner product; the $\boldsymbol{\Delta}_i(t)$ matrix is defined as

$$\boldsymbol{\Delta}_i(t)=\begin{bmatrix}\sum_{k=1}^{\infty}\delta(t-\tau(k)) & 0\\ 0 & \sum_{k=1}^{\infty}\delta(t-\tau(k))\end{bmatrix}$$

where δ is the Dirac function.

When $t=\tau(k)$,

$$\begin{aligned}\boldsymbol{v}_i(t+)&=\boldsymbol{v}_i(t-)+\int_{t-}^{t+}\boldsymbol{u}_i(s)\mathrm{d}s-2\int_{t-}^{t+}\boldsymbol{\Delta}_i(s)\mathrm{d}s\langle \boldsymbol{v}_i(t-),\boldsymbol{n}\rangle\boldsymbol{n}=\\ &\boldsymbol{v}_i(t-)-2\langle \boldsymbol{v}_i(t-),\boldsymbol{n}\rangle\boldsymbol{n}=\\ &\boldsymbol{v}_i(t-)-2\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{v}_i(t-)=\\ &(\boldsymbol{I}-2\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{v}_i(t-)\end{aligned}$$

It implies that when $t=\tau(k)$, the elastic collision occurs for the agent. $\boldsymbol{v}_i(t-)$ and $\boldsymbol{v}_i(t+)$ are the incident and reflection velocity, respectively. When $t\neq\tau(k)$, $\boldsymbol{v}_i(t+)=\boldsymbol{v}_i(t-)+\int_{t-}^{t+}\boldsymbol{u}_i(s)\mathrm{d}s=\boldsymbol{v}_i(t-)$, which means that the velocity is continuous if the agent does not touch the wall.

Let $\boldsymbol{w}$ be the unit normal vector which is perpendicular to $\boldsymbol{n}$. We use the convention such that $(\boldsymbol{n},\boldsymbol{w})$ forms a right-handed coordinate frame with $\boldsymbol{n}\times\boldsymbol{w}$ pointing to the reader. θ is the angle between vector $\boldsymbol{n}$ and vector $\boldsymbol{x}$, which is positive if $\boldsymbol{n}\times\boldsymbol{x}$ points to the reader, otherwise negative. $\boldsymbol{v}_i(t-)$ denotes the velocity before the agent i touches the wall, and $\boldsymbol{v}_i(t+)$ denotes the velocity after the collision. Defining symbolic variable $L_i(t)$, it is 1 before the collision and -1 after it.

Define the mirrored velocity matrix as

$$\boldsymbol{k}_i(t)=\begin{bmatrix}-b\sin\theta+aL_i(t)\cos\theta & a\sin\theta+bL_i(t)\cos\theta\\ -b\cos\theta-aL_i(t)\sin\theta & a\cos\theta-bL_i(t)\sin\theta\end{bmatrix}$$

where $\boldsymbol{n}=[a,b]^{\mathrm{T}}$, $\boldsymbol{w}=[-b,a]^{\mathrm{T}}$, and they meet $a^2+b^2=1$; $\boldsymbol{k}_i(0)=\boldsymbol{I}^{2\times2}$, $\boldsymbol{K}_i(t)=\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\boldsymbol{k}_i(\tau(2))\cdots\boldsymbol{k}_i(\tau(k))$, where $\boldsymbol{k}_i(\tau(1))$, $\boldsymbol{k}_i(\tau(2))$, $\cdots$, $\boldsymbol{k}_i(\tau(k))$ denote the value after the agent touches the wall, that is to say, $L_i(t)=-1$.

Define mirrored velocity $\tilde{\boldsymbol{v}}_i(t)=\boldsymbol{K}_i(t)\boldsymbol{v}_i(t)$. As $t=\tau(k)$, one has

$$\begin{aligned}\tilde{\boldsymbol{v}}_i(t+)&=\boldsymbol{K}_i(t+)\boldsymbol{v}_i(t+)=\\ &\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\cdots\boldsymbol{k}_i(\tau(k-1))\cdot\\ &\begin{bmatrix}-b\sin\theta-a\cos\theta & a\sin\theta-b\cos\theta\\ -b\cos\theta+a\sin\theta & a\cos\theta+b\sin\theta\end{bmatrix}\cdot\\ &(\boldsymbol{I}-2\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{v}_i(t-)=\\ &\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\cdots\boldsymbol{k}_i(\tau(k-1))\cdot\\ &\begin{bmatrix}-b\sin\theta+a\cos\theta & a\sin\theta+b\cos\theta\\ -b\cos\theta-a\sin\theta & a\cos\theta-b\sin\theta\end{bmatrix}\boldsymbol{v}_i(t-)=\\ &\boldsymbol{K}_i(t-)\boldsymbol{v}_i(t-)=\tilde{\boldsymbol{v}}_i(t-)\end{aligned}$$

It means that at the instantaneous time $t=\tau(k)$, the real agent velocity $\boldsymbol{v}_i(t)$ is not continuous. However, the mirrored velocity $\tilde{\boldsymbol{v}}_i(t)$ is always continuous.

2 Main Results

Next, we present our control law as follows:

$$\boldsymbol{u}_i(t)=-\boldsymbol{K}_i^{-1}(t)\sum_{j=1}^{n}a_{ij}(\boldsymbol{K}_i(t)\boldsymbol{v}_i(t)-\boldsymbol{K}_j(t)\boldsymbol{v}_j(t))\tag{2}$$

Theorem 1 Consider a system including n agents, each with dynamic equation (1) under the control protocol (2). If the graph G is undirected connected, the velocities of all the agents asymptotically converge to the same values.

Proof First, we define the mirrored acceleration $\tilde{\boldsymbol{u}}_i(t)=\boldsymbol{K}_i(t)\boldsymbol{u}_i(t)$, and the mirrored position and velocity have been defined as before. $\tilde{\boldsymbol{p}}_i(t)=\boldsymbol{p}_i(0)+\int_0^t\tilde{\boldsymbol{v}}_i(s)\mathrm{d}s$, and $\tilde{\boldsymbol{v}}_i(t)=\boldsymbol{K}_i(t)\boldsymbol{v}_i(t)$. Obviously, $\dot{\tilde{\boldsymbol{p}}}_i(t)=\tilde{\boldsymbol{v}}_i(t)$.

Because of $\dot{L}_i(t)=-2L_i(t)\delta$, when $t=\tau(k)$, taking the derivative of $\tilde{\boldsymbol{v}}_i(t)=\boldsymbol{K}_i(t)\boldsymbol{v}_i(t)$, one has

$$\begin{aligned}\dot{\tilde{\boldsymbol{v}}}_i(t)&=\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\cdots\boldsymbol{k}_i(\tau(k-1))\cdot\\ &\left\{\begin{bmatrix}-b\sin\theta+aL_i(t)\cos\theta & a\sin\theta+bL_i(t)\cos\theta\\ -b\cos\theta-aL_i(t)\sin\theta & a\cos\theta-bL_i(t)\sin\theta\end{bmatrix}\dot{\boldsymbol{v}}_i(t)+\right.\\ &\left.\begin{bmatrix}a\dot{L}_i(t)\cos\theta & b\dot{L}_i(t)\cos\theta\\ -a\dot{L}_i(t)\sin\theta & -b\dot{L}_i(t)\sin\theta\end{bmatrix}\boldsymbol{v}_i(t+)\right\}=\\ &\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\cdots\boldsymbol{k}_i(\tau(k-1))\cdot\\ &\left\{\begin{bmatrix}-b\sin\theta+aL_i(t)\cos\theta & a\sin\theta+bL_i(t)\cos\theta\\ -b\cos\theta-aL_i(t)\sin\theta & a\cos\theta-bL_i(t)\sin\theta\end{bmatrix}\boldsymbol{u}_i(t)-\right.\\ &2\delta L_i(t)\boldsymbol{v}_i(t-)\cdot\\ &\left.\begin{bmatrix}a\cos\theta(a^2+b^2-1) & b\cos\theta(a^2+b^2-1)\\ -a\sin\theta(a^2+b^2-1) & -b\sin\theta(a^2+b^2-1)\end{bmatrix}\right\}\end{aligned}\tag{3}$$

Substituting $a^2+b^2=1$ into Eq. (3) yields

$$\begin{aligned}\dot{\tilde{\boldsymbol{v}}}_i(t)&=\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\cdots\boldsymbol{k}_i(\tau(k-1))\cdot\\ &\begin{bmatrix}-b\sin\theta+aL_i(t)\cos\theta & a\sin\theta+bL_i(t)\cos\theta\\ -b\cos\theta-aL_i(t)\sin\theta & a\cos\theta-bL_i(t)\sin\theta\end{bmatrix}\boldsymbol{u}_i(t)=\\ &\boldsymbol{K}_i(t)\boldsymbol{u}_i(t)\end{aligned}$$

Therefore,

$$\left.\begin{aligned}\dot{\tilde{\boldsymbol{p}}}_i(t)&=\tilde{\boldsymbol{v}}_i(t)\\ \dot{\tilde{\boldsymbol{v}}}_i(t)&=\tilde{\boldsymbol{u}}_i(t)\end{aligned}\right\}$$

Calculating the determinant of $\boldsymbol{k}_i(t)$, one has

$$|\boldsymbol{k}_i(t)|=\begin{vmatrix}-b\sin\theta+aL_i(t)\cos\theta & a\sin\theta+bL_i(t)\cos\theta\\ -b\cos\theta-aL_i(t)\sin\theta & a\cos\theta-bL_i(t)\sin\theta\end{vmatrix}=$$

$$L_i(t)(a^2+b^2)=L_i(t)\neq 0$$

Therefore, $\boldsymbol{k}_i(t)$ is reversible, and $\boldsymbol{K}_i(t)=\boldsymbol{k}_i(0)\boldsymbol{k}_i(\tau(1))\boldsymbol{k}_i(\tau(2))\cdots$. Hence, $\boldsymbol{K}_i(t)$ is also reversible.

According to $\tilde{\boldsymbol{u}}_i(t)=\boldsymbol{K}_i(t)\boldsymbol{u}_i(t)$, the mirrored acceleration is $\tilde{\boldsymbol{u}}_i(t)=-\sum_{j=1}^{n}a_{ij}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))$.

We construct a Lyapunov function candidate: $H(t)=\frac{1}{2}\sum_{i=1}^{n}\tilde{\boldsymbol{v}}_i^{\mathrm{T}}(t)\tilde{\boldsymbol{v}}_i(t)$. Taking the derivative of $H(t)$ gives

$$\begin{aligned}\dot{H}(t)&=\sum_{i=1}^{n}\tilde{\boldsymbol{v}}_i^{\mathrm{T}}(t)\tilde{\boldsymbol{u}}_i(t)=\\&-\sum_{i=1}^{n}\tilde{\boldsymbol{v}}_i^{\mathrm{T}}(t)\sum_{j=1}^{n}a_{ij}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))=\\&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))^{\mathrm{T}}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))\leqslant 0\end{aligned}$$

Defining the average mirrored velocity $\bar{\boldsymbol{v}}(t)=\frac{1}{n}\sum_{i=1}^{n}\tilde{\boldsymbol{v}}_i(t)$, then we have

$$\begin{aligned}\frac{\mathrm{d}\bar{\boldsymbol{v}}(t)}{\mathrm{d}t}&=\frac{1}{n}\sum_{i=1}^{n}\tilde{\boldsymbol{u}}_i(t)=\\&-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))=0\end{aligned}$$

It means that $\bar{\boldsymbol{v}}(t)$ is a constant.

When the undirected graph is connected, by LaSalle's invariance principle, we have $\lim_{t\to\infty}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))^{\mathrm{T}}(\tilde{\boldsymbol{v}}_i(t)-\tilde{\boldsymbol{v}}_j(t))=0$, and when $t\to\infty$, $\tilde{\boldsymbol{v}}_i(t)\to\tilde{\boldsymbol{v}}_j(t)$.

3 Simulation Results

We verify the proposed algorithm through numerical simulations in a regular triangle. In this paper, we choose $n=10$, and the initial position and velocity of each agent are chosen randomly. For the regular triangle, the side length is $L=100$. The vertex coordinates are $(0,0)$, $\left(\frac{L}{2},\frac{\sqrt{3}L}{2}\right)$, $(L,0)$. We can obtain $\theta=\left\{-\frac{\pi}{2},\frac{\pi}{6},\frac{5\pi}{6}\right\}$, the normal vector $\boldsymbol{n}=\left\{(0,1),\left(\frac{\sqrt{3}}{2},-\frac{1}{2}\right),\left(-\frac{\sqrt{3}}{2},-\frac{1}{2}\right)\right\}$, and the tangent vector $\boldsymbol{w}=\left\{(-1,0),\left(\frac{1}{2},\frac{\sqrt{3}}{2}\right),\left(\frac{1}{2},-\frac{\sqrt{3}}{2}\right)\right\}$. We choose adjacent matrix $\boldsymbol{A}=\begin{bmatrix}0&1&&\\1&0&\ddots&\\&\ddots&\ddots&1\\&&1&0\end{bmatrix}$, so the undirected graph is connected.

The simulation results are shown in Figs. 1 to 4. The initial distribution is given in Fig. 1, and the velocity of each agent is represented by the length and direction of the line. The trajectories are shown in Fig. 2. Fig. 3 gives the consistent state of all the agents at 75 s. We can con-

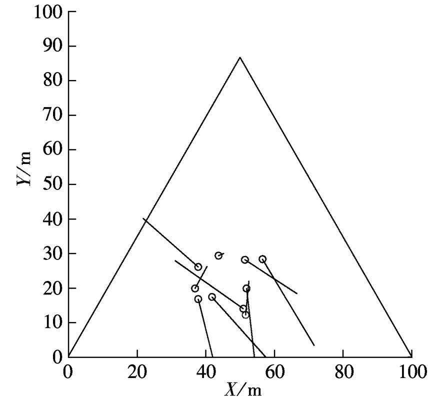

Fig. 1 The initial positions and velocities of 10 agents

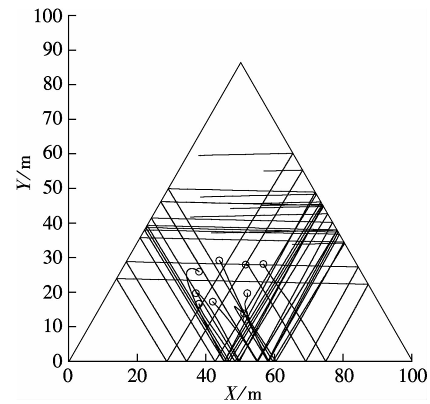

Fig. 2 The trajectories of 10 agents

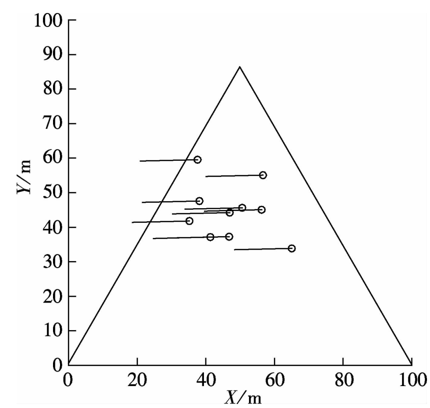

Fig. 3 The positions and velocities of 10 agents at $t=75$ s

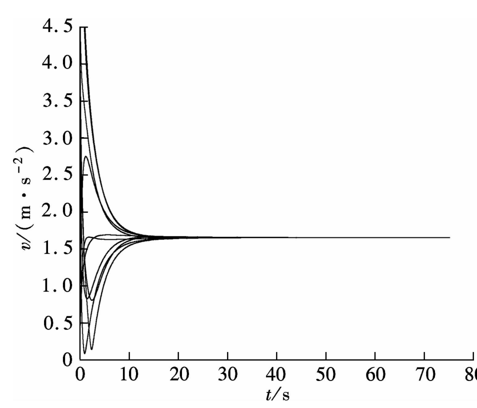

Fig. 4 The velocities of 10 agents

clude from Fig. 4 that the velocities asymptotically converge to the same values in about 20 s, and by the calculation, we find that the final velocity is the average of initial velocities.

4 Experimental Validation

We validate the proposed algorithm on a multi-robot platform. In the experiment, we choose the Amigobots which are used to obtain the results.

A diagram depicting the position, heading, forward and angular velocities of the robot is shown in Fig. 5. We have the following dynamic equation,

$$\left.\begin{aligned}\dot{\boldsymbol{\xi}}_i(t) &= \begin{bmatrix}\cos\varphi_i(t)\\ \sin\varphi_i(t)\end{bmatrix} v_i(t)\\ \dot{\varphi}_i(t) &= \omega_i(t)\end{aligned}\right\} \tag{4}$$

where $v_i(t) \in \mathbf{R}$, $\omega_i(t) \in \mathbf{R}$ are the forward and angular velocity inputs, respectively; $\boldsymbol{\xi}_i(t) \in \mathbf{R}^2$ is the position of the center C of the robot, and $\varphi_i(t)$ is the orientation angle.

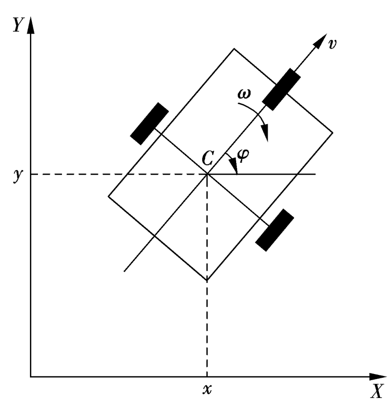

Fig. 5 The model of the wheeled mobile robot

To avoid the nonlinearities in Eq. (4), we introduce a new coordinate[11], $\boldsymbol{p}_i = \boldsymbol{\xi}_i + d[\cos\varphi_i \quad \sin\varphi_i]^{\mathrm{T}}$ for a positive number d. Here, d is the distance deviating from the center of mass. Letting

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos\varphi_i & \sin\varphi_i \\ -\frac{1}{d}\sin\varphi_i & \frac{1}{d}\cos\varphi_i \end{bmatrix} \boldsymbol{v}_i \tag{5}$$

we have

$$\begin{aligned}\dot{\boldsymbol{p}}_i &= \dot{\boldsymbol{\xi}}_i + d\begin{bmatrix}-\sin\varphi_i\\ \cos\varphi_i\end{bmatrix}\omega_i = \\ &\begin{bmatrix}\cos\varphi_i\\ \sin\varphi_i\end{bmatrix} v_i + d\begin{bmatrix}-\sin\varphi_i\\ \cos\varphi_i\end{bmatrix}\begin{bmatrix}-\frac{1}{d}\sin\varphi_i & \frac{1}{d}\cos\varphi_i\end{bmatrix}\boldsymbol{v}_i = \\ &\begin{bmatrix}\cos^2\varphi_i & \sin\varphi_i\cos\varphi_i\\ \sin\varphi_i\cos\varphi_i & \sin^2\varphi_i\end{bmatrix}\boldsymbol{v}_i + \\ &\begin{bmatrix}\sin^2\varphi_i & -\sin\varphi_i\cos\varphi_i\\ -\sin\varphi_i\cos\varphi_i & \cos^2\varphi_i\end{bmatrix}\boldsymbol{v}_i = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\boldsymbol{v}_i = \boldsymbol{v}_i\end{aligned}$$

According to the aforementioned results, we have $\boldsymbol{v}_i(t+) = (\boldsymbol{I} - 2\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}})\boldsymbol{v}_i(t-)$. When $t = \tau(k)$, for the bottom edge, we can conclude that $\varphi_i(t+) = -\varphi_i(t-)$, and $\boldsymbol{n} = [0 \quad 1]^{\mathrm{T}}$. Then

$$\boldsymbol{v}_i(t+) = \left(\boldsymbol{I} - 2\begin{bmatrix}0\\1\end{bmatrix}[0 \quad 1]\right)\boldsymbol{v}_i(t-) = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\boldsymbol{v}_i(t-)$$

As a result,

$$\begin{aligned}v_i(t+) &= [\cos\varphi_i(t+) \quad \sin\varphi_i(t+)]\boldsymbol{v}_i(t+) = \\ &[\cos(-\varphi_i(t-)) \quad \sin(-\varphi_i(t-))]\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\boldsymbol{v}_i(t-) = \\ &[\cos(\varphi_i(t-)) \quad \sin(\varphi_i(t-))]\boldsymbol{v}_i(t-) = v_i(t-)\end{aligned}$$

and

$$\begin{aligned}\omega_i(t+) &= \begin{bmatrix}-\frac{1}{d}\sin\varphi_i(t+) & \frac{1}{d}\cos\varphi_i(t+)\end{bmatrix}\boldsymbol{v}_i(t+) = \\ &\begin{bmatrix}-\frac{1}{d}\sin(-\varphi_i(t-)) & \frac{1}{d}\cos(-\varphi_i(t-))\end{bmatrix}\cdot \\ &\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\boldsymbol{v}_i(t-) = \\ &\begin{bmatrix}\frac{1}{d}\sin(\varphi_i(t-)) & -\frac{1}{d}\cos(\varphi_i(t-))\end{bmatrix}\cdot \\ &\boldsymbol{v}_i(t-) = -\omega_i(t-)\end{aligned}$$

For the right hypotenuse, $\varphi_i(t+) = -120° - \varphi_i(t-)$ and $\boldsymbol{n} = \begin{bmatrix}-\frac{\sqrt{3}}{2} & -\frac{1}{2}\end{bmatrix}$, and then $\boldsymbol{v}_i(t+) = \begin{bmatrix}-\sin 30° & -\cos 30°\\ -\cos 30° & \sin 30°\end{bmatrix}\boldsymbol{v}_i(t-)$. Therefore,

$$\begin{aligned}v_i(t+) &= [\cos\varphi_i(t+) \quad \sin\varphi_i(t+)]\boldsymbol{v}_i(t+) = \\ &[\cos(-120° - \varphi_i(t-)) \quad \sin(-120° - \varphi_i(t-))]\cdot \\ &\begin{bmatrix}-\sin 30° & -\cos 30°\\ -\cos 30° & \sin 30°\end{bmatrix}\boldsymbol{v}_i(t-) = \\ &[\cos(\varphi_i(t-)) \quad \sin(\varphi_i(t-))]\boldsymbol{v}_i(t-) = v_i(t-)\end{aligned}$$

and

$$\begin{aligned}\omega_i(t+) &= \begin{bmatrix}-\frac{1}{d}\sin\varphi_i(t+) & \frac{1}{d}\cos\varphi_i(t+)\end{bmatrix}\boldsymbol{v}_i(t+) = \\ &\begin{bmatrix}-\frac{1}{d}\sin(-120° - \varphi_i(t-)) & \frac{1}{d}\cos(-120° - \varphi_i(t-))\end{bmatrix}\cdot \\ &\begin{bmatrix}-\sin 30° & -\cos 30°\\ -\cos 30° & \sin 30°\end{bmatrix}\boldsymbol{v}_i(t-) = \\ &\begin{bmatrix}\frac{1}{d}\sin(\varphi_i(t-)) & -\frac{1}{d}\cos(\varphi_i(t-))\end{bmatrix}\boldsymbol{v}_i(t-) = \\ &-\omega_i(t-)\end{aligned}$$

In the same way, when $t = \tau(k)$, for the left hypotenuse, we have the same conclusion, namely $v_i(t+) = v_i(t-)$ and $\omega_i(t+) = -\omega_i(t-)$. When $t \neq \tau(k)$, the controller is Eq. (5) with $\dot{\boldsymbol{v}}_i(t) = \boldsymbol{u}_i(t)$ and $\boldsymbol{u}_i(t)$ in Eq. (2).

Next, an experiment is conducted for the velocity consensus algorithm in the simulation software MobileSim with three Amigobot wheeled mobile robots to demonstrate its effectiveness. The control parameters are set as follows: $n=3$, $d=0.02$ m, the sampling time $T=0.015$ s, and the initial conditions of the group are chosen as

$$\boldsymbol{p}_1(0) = [5 \quad 2.8]^{\mathrm{T}}\ \mathrm{m}$$

$$\boldsymbol{p}_2(0) = [5 \quad 1.8]^{\mathrm{T}}\ \mathrm{m}$$

$$\boldsymbol{p}_3(0) = [5 \quad 0.8]^{\mathrm{T}}\ \mathrm{m}$$

$$v_1(0) = 120\ \mathrm{mm/s}$$

$$v_2(0) = 80\ \mathrm{mm/s}$$

$$v_3(0) = 40\ \mathrm{mm/s}$$

$$\varphi_1(0) = \frac{2\pi}{3}, \quad \varphi_2(0) = \frac{5\pi}{6}, \quad \varphi_3(0) = \frac{8\pi}{9}$$

First, let the three robots run for a while, and then each robot will acquire some velocity. The complete moving trajectories of three robots are plotted in Fig. 6 according to the distributed algorithm. At the beginning, the velocities of three robots are different, but due to several incidences of bounces on the boundaries, the velocities of the robots will reach a consensus eventually. We find that the robot crashes into the wall, just as a beam of light falls onto the mirror and creates a regular reflection. Meanwhile, when we enlarge the parameter d, the robot will make a slow turn. On the contrary, if we decrease the parameter, the robot will make a sharp turn and become more unstable. In other words, the robustness decreases. Although sometimes the actual velocities do not parallel each other, the mirror velocities always do. However, due to several incidences of bounces, the actual velocities also parallel each other. We can find that eventually the robots synchronize to a common moving direction with a common speed. It is clear that the experimental results are well consistent with the corresponding theoretical results.

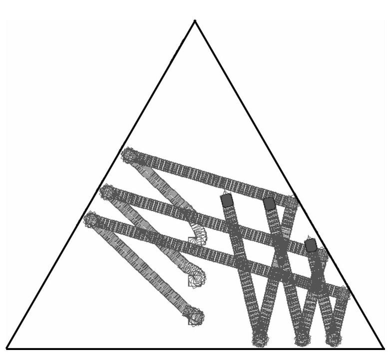

Fig. 6 The trajectories of 3 Amigobots

5 Conclusion

In this paper, we introduce the concept of mirrored speed and extend the double-integrator control algorithm from an infinite space to a bounded space. With the help of a mirrored matrix, the discontinuous real speed is converted into a continuous mirrored speed, and the mirrored matrix is also suitable for non-rectangular regions, so it is more general. The connectivity of the undirected graph ensures the consensus of velocity. Furthermore, the consensus algorithm is implemented in the control of the wheeled multi-robot systems. Results of the simulations and experiments show that the proposed algorithm can be effectively applied to multiple wheeled mobile robots. The future focus is to study how to maintain a formation in a bounded space, so that it makes more sense to conduct cooperative exploration in a bounded space.

References

[1] Zavlanos M M, Jadbabaie A, Pappas G J. Flocking while preserving network connectivity [C]//*Proceedings of the 46th IEEE Conference on Decision and Control*. New Orleans, LA, USA, 2007: 2919－2924.

[2] Wang W K, Peng H X. Flocking control with communication noise based on second-order distributed consensus algorithm [C]//*Power Engineering and Automation Conference*. Wuhan, China, 2012: 1－4.

[3] Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory [J]. *IEEE Transactions on Automatic Control*, 2006, **51**(3): 401－420.

[4] Cao H, Chen J, Mao Y T, et al. Formation control based on flocking algorithm in multi-agent system [C]//*Proceedings of the 8th World Congress on Intelligent Control and Automation*. Jinan, China, 2010: 2289－2294.

[5] Dong Y, Huang J. A leader-following rendezvous problem of double integrator multi-agent systems [J]. *Automatica*, 2013, **49**(5): 1386－1391.

[6] Peng K, Su H S, Yang Y P. Coordinated control of multi-agent systems with a varying-velocity leader and input saturation [J]. *Communications in Theoretical Physics*, 2009, **52**(3): 449－456.

[7] Su H, Wang X, Chen G. Rendezvous of multiple mobile agents with preserved network connectivity [J]. *Systems & Control Letters*, 2010, **59**(5): 313－322.

[8] Abdessameud A, Tayebi A. On consensus algorithms design for double integrator dynamics [J]. *Automatica*, 2013, **49**(1): 253－260.

[9] Chen Z Y, Zhang H T, Fan M C, et al. Algorithms and experiments on flocking of multiagents in a bounded space [J]. *IEEE Transactions on Control Systems Technology*, 2014, **22**(4): 1544－1549.

[10] Zhang H T, Zhai C, Chen Z Y. A general alignment repulsion algorithm for flocking of multi-agent systems [J]. *IEEE Transactions on Automatic Control*, 2011, **56**(2): 430－435.

[11] Cao Y C, Stuart D, Ren W, et al. Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: algorithms and experiments [J]. *IEEE Transactions on Control Systems Technology*, 2011, **19**(4): 929－938.

多智能体在非矩形有界空间中的一致性控制算法和实验

朱德政　田玉平

（东南大学自动化学院，南京 210096）

摘要：针对多智能体在非矩形有界空间的运动，提出了二阶动态系统的速度一致性算法．传统的有界空间一致性算法只适合矩形有界空间，对于非矩形有界空间不再适用．为了将已有的一致性算法扩展到非矩形空间，引入镜像速度矩阵的概念，它不仅可将不连续的实际速度转化成连续的镜像速度，而且可将有界空间扩展成无限大虚拟空间．运用此算法，发现多智能体在虚拟空间中镜像速度渐近一致．由于每个镜像速度对应唯一的实际空间速度，多智能体实际速度也达到渐近一致．最后，通过理论证明和数值仿真验证了算法的可行性，并且成功地将算法运用到一组实际多机器人系统上．

关键词：多智能体系统；一致性；非矩形有界空间；镜像速度

中图分类号：TP242.6