INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL.15,NO.1,MARCH 2010,9-17

Synchronization on a Segment Without Localization: Algorithm, Applications, and Robot Experiments

Hua WANG and Yi GUO

***Abstract*—We study the multi-robot synchronization on a segment. The goal is for each robot to move along a subsegment of equal length in equal time interval with potential impacts. To achieve the synchronization, we propose a decentralized algorithm by designing impact laws, which do not depend on the positions of the robots, but on the time information. Specifically, "the time interval between two consecutive impacts" is exchanged when the robots meet. We also show how to apply the synchronization algorithm to a planar patrolling problem. Simulations and robot experimental results show the feasibility and robustness of our algorithm.**

***Index Terms*—Multi-robot System, Synchronization, Collective Behavior, Robotic Patrolling**

1. INTRODUCTION

The problem of N-beads sliding freely on a curve with collisions has fascinated researchers for long. Sevryuk in [1] elucidates some fundamental results when the N beads collide elastically on a *line* (infinitely long), and proves that the total number of collisions between particles for all initial conditions is finite and upper bounded. By contrast, on a *ring*, most initial conditions lead to an infinite number of collisions, which make the problem of "N-beads sliding freely on a ring" more interesting. Numerous work studies the influence of the collision and friction on the impact dynamics. In [2], Glashow and Mittag observe that the problem of three beads sliding on a frictionless ring with elastic collisions is equivalent to a standard billiard flow, where the dynamic system is studied that consists of three hard rods sliding along a frictionless ring with potential elastic collisions, as well as the dynamic system that consists of one ball moving on a frictionless triangular table with elastic rails. Cooley and Newton in [3] study the elastic/inelastic impact dynamics of N-beads on a frictionless ring via matrix products. Each collision sequence is taken as a billiard trajectory in a right triangle with non-standard reflection rules, and the existence of the periodic orbits is proven.

Imitating the impact behavior of N-beads' collective motion, and designing an artificial impact law in robotics applications become very interesting. In [4], it is shown that the synchronization of beads on a ring can be achieved by modifying the impact law based on the knowledge of discrete-time consensus algorithms, in which each bead updates its logic state based on the distance from its current position to the "center of dominance region". A necessary condition for synchronization is that the number of beads in the collection is even, the initial velocity of half beads in the system is clockwise, and the other half is counterclockwise. The algorithm is proven to converge to a steady state locally. In [5], Kingston *et. al.* propose a decentralized solution to the cooperative perimeter-surveillance problem, which is robust to the insertion/deletion of team members and the perimeter expanding and contracting. The algorithm requires that each agent knows the length of the perimeter, the total number of the agents on the team, and its position in the team. The approach converges in finite time. Both works require only intermittent communication, which is very important in robotics patrolling and surveillance applications. Moreover, in [6], an algorithm is proposed to the patrolling problem by generating a circular patrol path for a team of mobile robots inside a designated target area to guarantee the maximum uniform frequency. The robots are distributed uniformly along the path, and terrain directionality and velocity constraints are also taken into consideration therein. William and Burdick in [7] study the problem of patrolling a multi-object boundary by a multi-robot system, where the complexity of the original problem is reduced based on a graph representation, and an algorithm is proposed to revise paths in cases that the team size or the environment changes.

In this paper, we consider a multi-robot system moving on a line segment with potential impacts. We design impact laws (i.e. control laws when robots meet each other) to achieve motion synchronization by each robot moving along an equal-length subsegment in equal time-span on a line segment. The algorithm assumes simple information exchange, namely, the time span since the last impact, and assumes no knowledge of the total number of robots, nor the total length of the line segment. While similar ideas appear in [4] and [5], some distance measurement to critical points and the priori knowledge such as the perimeter length, the positions of the robots, or the total robot count are required. We relax such assumptions, and use only the information of the robot interaction time and velocities in constructing the control laws. We also consider the scenario when multi-robot impact (more than two robots) at the same point, which is not studied in previous work. Our algorithm is decentralized, and robots only communicate to their adjacent neighbors when they meet each other. It is robust to robot failures, in the sense that a removal or an addition of robots does not affect the patrolling goal and eventually every point of the patrolling path is visited with a

Hua Wang and Yi Guo are with the Electrical and Computer Engineering Department, Stevens Institute of Technology, Hoboken, NJ 07030, USA hwang@stevens.edu; yi.guo@stevens.edu

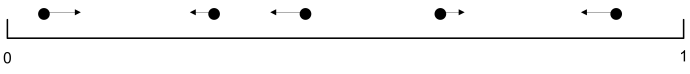


Fig. 1. A demonstration of a 5 robot system moving along the segment [0,1]

uniform frequency.

The rest of the paper is organized as follows. In Section 2, we present the system model, define the synchronization problem and desired system behavior. In Section 3, we propose an algorithm by designing the impact laws for different impact types, and analyze the convergence of this system. In Section 4, we apply the synchronization problem into a planar patrolling problem using the Hamiltonian path. We demonstrate the satisfactory simulation results in Section 5. We validate the feasibility of the system on an e-puck multi-robot platform in Section 6. Finally, we conclude in Section 7.

2. PROBLEM FORMULATION

Consider assigning an N homogeneous mobile robot system $\mathcal{S}$ to move efficiently along the segment $[0,1]$, with sporadic communications among robots when they meet. A demonstration is shown in Fig. 1, where 5 robots are moving along the segment with potential impacts (either between robots or between the robot and the boundary).

The vehicle model we study is a first-order point-mass model with state vector q_i as

$$q_i = [x_i \ v_i \ k_i \ t_{k_i-1}]^T \tag{1}$$

where $i \in [1, 2, \dots N]$, x_i is a continuous-time state denoting the robot's position; $v_i \in \mathbb{R}$ denotes the velocity, and $k_i \in \mathbb{Z}$ is a discrete state denoting the number of impacts that the i^{th} robot has encountered; and t_{k_i-1} is a discrete state timer denoting the time instant the last $(k_i-1)^{th}$ impact happens.

Furthermore, we assume:

1) Robot i moves on the segment $[0,1]$, with a constant speed v_i.
2) Robots communicate only when they meet.
3) Robots change velocity instantly right at the moment they meet.
4) Robot i is initialized with random position $x_i^{init} \in [0,1]$, random nonzero velocity $v_i^{init} \in [-1,1] \setminus 0$, $k_i^{init} = 0$, $t_i^{init}(-1) = 0$. Furthermore, robots initially are not aligned at the same position, i.e, $\forall i,j \in \mathcal{S}, x_i^{init} \neq x_j^{init}$.

Since the robot in the system moves at a constant speed, the continuous dynamic can be modelled as

$$f(q) = \begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = 0 \end{cases}, \quad \forall q \in C \tag{2}$$

where C is the flow set where the continuous dynamic is available.

However, when the impacts happen among robots, the velocity vector jumps in response to the impact, the dynamic system is discrete at this point, We describe such discrete dynamic model as

$$g(q) = \begin{cases} x_i^+ = 0 \\ v_i^+ = u_i \end{cases}, \quad \forall q \in D \tag{3}$$

where D is the jump set when the impact happens. Thus this dynamic system can be written in a hybrid system form

$$H := (f, C, g, D) \tag{4}$$

Remark 1: The assumptions imply that, robot $i \in 2 \dots N-1$ will only impact with robot $i-1$ and $i+1$; robot 1 only impacts with the boundary and robot 2; robot N only impacts with robot $N-1$ and the boundary.

Remark 2: The communication topology is connected over finite time because all the impacts happen in finite time.

We now define our control goal as achieving the synchronization on a segment, which is an efficient solution to the patrolling problem.

Definition 1: (Synchronization on a segment) Consider a collection of robots moving along the segment $[0,1]$, the system reaches *synchronization on a segment*, if
(1) All the robots move at the same nonzero speed v_{ss} which is pre-set among robots. That is

$$\forall i \in \mathcal{S}, |v_i| = v_{ss} \neq 0 \tag{5}$$

(2) For an impact occurring on the robot pair i and $i+1$, $i \in [1, 2 \dots N-1]$, it is the k_i^{th} impact on robot i, and the k_{i+1}^{th} impact on robot $i+1$, the time spans since their last impact are equal, that is

$$\Delta t(k_i - 1, k_i) = \Delta t(k_{i+1} - 1, k_{i+1}) \tag{6}$$

where

$$\begin{aligned} \Delta t(k_i - 1, k_i) &= t_{k_i} - t_{k_i-1} \\ \Delta t(k_{i+1} - 1, k_{i+1}) &= t_{k_{i+1}} - t_{k_{i+1}-1} \end{aligned} \tag{7}$$

Such distribution of a patrolling goal is effective because each robot patrols equal distance in the same time span t_{ss}. So the segment is visited by the multi-robot system with equal frequency. The frequency of the area being patrolled is $1/t_{ss}$ once the system synchronizes on the segment.

Based on the definition of synchronization on a segment, we define the problem under consideration.

Problem 1: Under assumptions 1 to 4, find a decentralized control algorithm for (4) such that robots achieve synchronization on the segment [0,1].

As the robots move along the segment $[0,1]$, impacts happen either between two robots, or one robot hits the boundary. Now we define the behavior of impacts in four categories.

Definition 2: (Impact type) "Impact" happens in four cases, which are head-head type, head-tail type, boundary-hit type, and multi-hit type respectively.

- Head-Head Type Impact: As shown in Fig. 2, the head-head type happens between robots i and $i+1$, when $v_i v_{i+1} < 0$.
- Head-Tail Type Impact: As shown in Fig. 3, the head-tail type happens between robots i and $i+1$, when $v_i v_{i+1} > 0$, and $v_i(|v_i| - |v_{i+1}|) > 0$.
- Hit-Boundary Type Impact: As shown in Fig. 4, the hit-boundary type impact happens between the robot 1 or N, when either $v_1 < 0$ or $v_N > 0$.

Fig. 2. An illustration of the head-head type impact between robots i and $i+1$

Fig. 3. An illustration of the head-tail type impact between robots i and $i+1$

- Multi-Hit Type Impact: As shown in Fig. 5, the multi-hit type happens when the robots $i, i+1, \ldots,$ and $i+k$, $k \geq 2$ bump into the same position. Given the velocity and position vector p and v at time t_0, under the condition

$$\begin{cases} v_i > v_{i+1} > \cdots > v_{i+k} \\ p_i < p_{i+1} < \cdots < p_{i+k} \end{cases}$$

there exists time period $t_\alpha > 0$ that agents $i, i+1 \ldots i+k$ bump into each other at $t = t_0 + t_\alpha$

$$p_i + t_\alpha v_i = p_{i+1} + t_\alpha v_{i+1} \ldots = p_{i+k} + t_\alpha v_{i+k}$$

3. Synchronization Algorithm

We describe our decentralized control law to achieve synchronization in this section. The basic idea is that each robot in the system under motion moves in a constant velocity until an impact happens (i.e., when they meet). Then, we define different updating laws when different types of impacts happen. "Constant velocity" means that the robot moves along a straight line without any changes of the magnitude nor the direction of its velocity.

The flow chart of the algorithm is shown as in Figure 6. We describe each block of the flow chart in the following.

3.1. Initialize state space

At time $t = 0$, according to the assumptions defined in Section 3, for robot $i \in \mathcal{S}$, its state vector $q_i^{init} = [x_i^{init}\ v_i^{init}\ k_i^{init}\ t_{k_i-1}^{init}]$ is initialized as:

1) $x_i^{init} \in [0,1], \forall j \in \mathcal{S}, j \neq i \Rightarrow x_i^{init} \neq x_j^{init}$;
2) $v_i^{init} \in [-1,1] \backslash 0$;

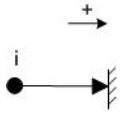


Fig. 4. An illustration of the hit-boundary type impact of robot i

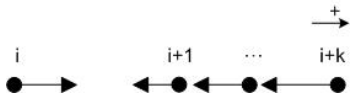


Fig. 5. An illustration of the multi-robot hit type impact of robot i, $i+1$ ... and $i+k$

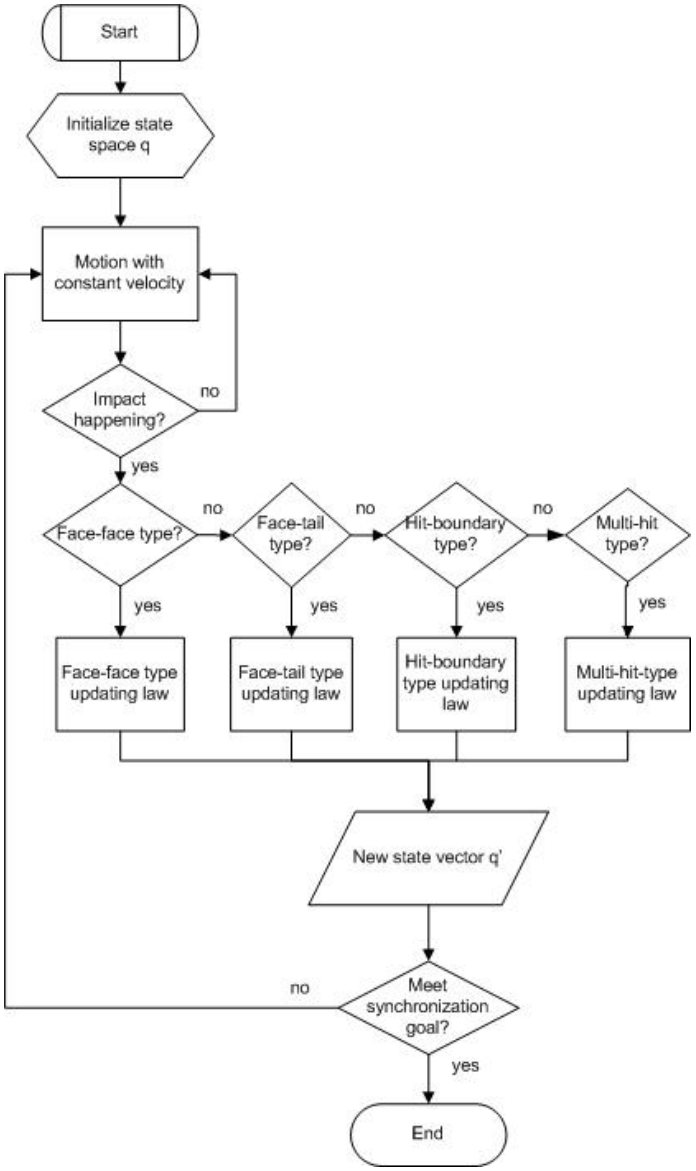


Fig. 6. The flowchart of the synchronization algorithm

3) $k_i^{init} = 0$;
4) $t_{k_i-1}^{init} = 0$.

Once initialized with positions and velocities, a robot is free to move on the segment at the constant velocity initialized, until the first impact happens. We then come up with the state-vector updating laws in different impacting cases. Again, as mentioned earlier, two robots communicate when impacting, in other words, the state vectors are updated at impact time.

3.2. Motion in constant velocity

At time t_p, any robot i in the robot system S moves at constant velocity. The equation of motion is modelled as:

$$f(q) \stackrel{\text{def}}{=} \begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = 0 \end{cases}, \quad q \in C \tag{8}$$
$$C = \{q : x_j \neq x_k, \forall i, j \text{ in } S\}$$

3.3. Face-face type updating law

At time t_p, a face-face type impact happens between robot i and $i+1$, $(i, i+1 \in \mathcal{S})$. At t_p^+, the positions of i and $i+1$ are intact. In the following, we denote $\Delta t(k_i - 1, k_i)$ and $\Delta t(k_{i+1} - 1, k_{i+1})$ as Δt_i and Δt_{i+1}. Next, the counters k_i and k_{i+1} are accumulated by one, since one more impact just happened. The velocities of robots i and $i+1$ are to be reversed, and updated according to the difference between the time span Δt_i and Δt_{i+1}, which are the time spans from t_p^+ traced back to the last impacting instant for robots i and $i+1$, respectively. Finally, the timers t_{k_i-1} and $t_{k_{i+1}-1}$ are replaced by the newest impact moment t_p. The other robot state vectors are intact at time t_p.

The velocity updating law is originated from the idea of the feedback control. Since the algorithm is decentralized, one is only able to communicate with its neighbors. We want the magnitude of the velocity reaches v_{ss}, as well as the impact time span reach consensus. Thus, v_{ss} dominates the magnitude of the velocity. The velocity is lowered, when the latest impact time span is shorter than the neighbor, and vice versa. Once the time span of two impacting robots are the same, the velocity is set as the ultimate velocity v_{ss}.

The mathematical representation of such impacting law is shown as follows. When the k^{th} face-to-face type impact happens between robots i and $i+1$ at time instant t_p, the state update law is:

$$g_1(q,t_p) \stackrel{\text{def}}{=} \begin{cases} v_i(t_p^+) = -sgn(v_i)(v_{ss} + \\ \quad a_1(\Delta t_i - \Delta t_{i+1})v_{ss}v_i) \\ v_{i+1}(t_p^+) = -sgn(v_{i+1})(v_{ss} + \\ \quad a_2(\Delta t_{i+1} - \Delta t_i)v_{ss}v_{i+1}) \end{cases}, q \in D_1 \quad (9)$$
$$D_1 = \{q : x_i = x_{i+1}, v_i v_{i+1} < 0\}$$

a_1 and a_2 are parameters that work as the feedback gain, $|a_1| < 1, |a_2| < 1$. $a_1 < 0, a_2 > 0$ if $\Delta t_{i+1} > \Delta t_i$, and vice versa. $k_i^+ \leftarrow k_i + 1$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_i-1}^+ \leftarrow t_p$, $t_{k_{i+1}-1}^+ \leftarrow t_p$. And sgn is the sign function that extracts the sign of a real number. So the $-sgn(v_i)$ plays the role that it reverses the velocity of the robot i after the impact at time t_p.

3.4. Face-tail type updating law

Considering the face-tail type impact happens at t_p between robots i and $i+1$, $(i, i+1 \in \mathcal{S})$. At time t_p^+, the positions and the velocity magnitude of i and $i+1$ remain the same; the direction of the one with larger velocity magnitude will be reversed; the impacting counters k_i and k_{i+1} is added by one, the timer t_{k_i-1} and $t_{k_{i+1}-1}$ are replaced by the newest impact time t_p. The other robot state vectors are intact at time t_p.

When the k^{th} face-to-tail type impact happens between robots i and $i+1$ at time t_p, the state update control can be represented in the form:

$$g_2(q,t_p) \stackrel{\text{def}}{=} \begin{cases} v_i(t_p^+) = -sgn(|v_i| - |v_{i+1}|)v_i \\ v_{i+1}(t_p^+) = -sgn(|v_{i+1}| - |v_i|)v_{i+1} \end{cases}, q \in D_2 \quad (10)$$
$$D_2 = \{q : x_i = x_{i+1}, v_i v_{i+1} > 0, v_i(|v_i| - |v_{i+1}|) > 0\}$$

The timer and counter update as: $k_i^+ \leftarrow k_i + 1$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_i-1}^+ \leftarrow t_p$, and $t_{k_{i+1}-1}^+ \leftarrow t_p$.

3.5. Multi-hit type updating law

At time t_p, suppose that the robots $i, i+1, \ldots$ and $i+k$, meet each other at the same position. At t_p^+, the position vector stays the same, the velocities of the robots $i, i+1, \ldots i+k$ will be reversed; each of time counters $k_i, k_{i+1} \ldots$ and k_{i+k} is added by one, and the recall timer is updated by the current time. And we show the law as follows.

$$g_3(q,t_p) \stackrel{\text{def}}{=} \begin{cases} v_i(t_p^+) = -v_i \\ v_{i+1}(t_p^+) = -v_{i+1} \\ \cdots \\ v_{i+k}(t_p^+) = -v_{i+k} \end{cases}, q \in D_3 \quad (11)$$
$$D_3 = \{q : x_i = x_{i+1} \cdots = x_{i+k}, v_i < v_{i+1} \cdots < v_{i+k}\}$$

And again, counter and timer update as: $k_i^+ \leftarrow k_i + 1$, $t_{k_i-1}^+ \leftarrow t_p$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_{i+1}-1}^+ \leftarrow t_p$, $\ldots$, $k_{i+k}^+ \leftarrow k_{i+k} + 1$, and $t_{k_{i+k}-1}^+ \leftarrow t_p$.

3.6. Hit-boundary type updating law

At time t_p, suppose robot $i, (i \in \mathcal{S})$ hits boundary. The positions of i is intact. the velocity v_i is to be reversed, and the magnitude is to be tuned at the ultimate velocity v_{ss}. The impacting counter k_i is added by one, the timer t_{k_i-1} is replaced by the newest impact time t_p. The other robot state vectors are intact at time t_p.

When the k^{th} boundary-hit type impact happens between robot i and boundary, we set the state update law as:

$$g_4(q,t_p) \stackrel{\text{def}}{=} v_i(t_p^+) = -sgn(v_i)v_{ss}, q \in D_4 \quad (12)$$
$$D_4 = \{\ q : \{x_1 = 0, v_1 < 0\} \bigcup \{x_N = 1, v_N > 0\}\ \}$$

And timer and counter update as: $k_i^+ \leftarrow k_i + 1$, and $t_{k_i-1}^+ \leftarrow t_p$.

3.7. Convergence analysis

Next, we discuss the synchronization of the hybrid dynamic system H by looking into the consensus of the velocity magnitude $|v_i|$ and the elapsed time Δt_i, respectively.

A) Consensus of the velocity magnitude We first address the consensus of the velocity magnitude $|v_i|$. For agents $i \in \{2 \ldots N-1\}$ (that is excluding the leftmost and rightmost robots which applies the hit-boundary control law (12)) with control laws (8), (9), (10) and (11), in order to study the velocity magnitude instead of the velocity itself, we define

$$v_i^* = |v_i| - v_{ss} \quad (13)$$

Then we have

$$\begin{cases} \dot{v}_i^* = 0, \quad q \in C \\ v_i^{*+} = (a_1(\Delta t_i - \Delta t_{i+1})v_{ss} - 1)v_i^*, \quad q \in D_1 \\ v_i^{*+} = v_i^*, \quad q \in D_2 \\ v_i^{*+} = v_i^*, \quad q \in D_3 \end{cases} \quad (14)$$

Note that the flow set C, the continuous dynamic defined in (8) does not change the magnitude of the velocity, since the differential of velocity $\dot{v}$ is 0. On the other hand, in the jump set D, we look at the corresponding impact case individually. Obviously, the face-tail type and multi-hit type of impacting

laws in (10) and (11) do not change the magnitude of velocity, only the direction of the velocity reverses. So we have

$$\begin{cases} v_i^{*+} = a_1(\Delta t_i - \Delta t_{i+1}) v_{ss} v_i^*, & q \in D_1, \\ \Delta v_i^* = 0, & q \in C, D_2, \text{and} D_3 \end{cases} \tag{15}$$

Next, take a closer look at $\{v_i^{*+} : q \in D_1\}$. Since the robot moves in a bounded constant velocity between the impacts on a bounded segment within the length 1, the impact span Δt_i is upper and lower bounded as

$$0 < \frac{1}{max(|v_i|)} \leq \Delta t_i \leq \frac{1}{min(|v_i|)} \tag{16}$$

which implies

$$0 \leq |\Delta t_i - \Delta t_{i-1}| \leq \frac{1}{min(|v_i|)} \tag{17}$$

Since the velocity v_i is upper and lower bounded, from (9), it is seen that v_i jumps around v_{ss} with a small offset, which corresponds to the time inconsistence between two impacting robots. Due to the boundedness and the offset around v_{ss}, there exists $0 < e_1 < 1, e_2 > 1$ so that $e_1 v_{ss} < |v_i| < e_2 v_{ss}$ holds, we have $min(|v_i|) = e_1 v_{ss}$, $max(|v_i|) = e_2 v_{ss}$. Next consider the common ratio r_k defined as

$$r_k = \left| \frac{v^{*+}_i}{v^*_i} \right| \tag{18}$$

when the kth face-face type impact happens,

$$\begin{aligned} r_k &= \left| \frac{v^{*+}_i}{v^*_i} \right| = |a_1(\Delta t_i - \Delta t_{i+1}) v_{ss}| \\ &\leq \left| a_1 \frac{v_{ss}}{min(v_i)} \right| = \frac{|a_1|}{e_1} \end{aligned} \tag{19}$$

For any given $|a_1| < e_1$, it holds that

$$r_k < 1 \tag{20}$$

Therefore, the velocity magnitude $|v_i^*|$ is always upper bounded by a decaying geometric series with the common ratio ρ, where $r_k < \rho < 1$ for $k \in \{1, 2 \ldots, \infty\}$. Thus, v_i^* decays to zero as the number of face-face type impacts approaches infinity. Thus, for agent $i \in \{2 \ldots N-1\}$, the velocity magnitude $|v_i|$ approaches v_{ss} at $t \to \infty$. For agent $i = 1$, and N, as in (12), it jumps to v_{ss} after each boundary-hit update. In another word, the velocity magnitude $|v_i|$ jumps to v_{ss} on the jump set D_4.

Since the velocity magnitude converges to v_{ss} on the flow set C and the jump sets D_1, D_2, D_3, D_4. Therefore, the velocity magnitude of any agent i in the system reaches consensus at v_{ss} as time goes to infinity. Note that once for any given $i \in \{1 \ldots N\}$, its velocity magnitude $|v_i|$ reaches v_{ss}, the sets D_2, D_3 do not hold any more, in another word, the face-tail type, multi-hit type impacts are exclusive once the velocity magnitude reaches consensus.

B) Consensus of the time span Next, we analyze the time span between impacts for agent i and $i+1$, for $i \in \{1, 2 \ldots N-1\}$, as it is shown above that the magnitude of v_i converges to v_{ss} as $t \to \infty$. From (9), as $|v_i|$ converges to v_{ss}, Δt_i tends to Δt_{i+1} for any impacting pair i and $i+1$, for $i = \{1, 2 \ldots N-1\}$. Thus, for any agent i in the system, the elapsed time between two consecutive impacts reaches consensus as $t_{ss} = 1/(N \cdot v_{ss})$, which completes the convergence analysis of the dynamic system.

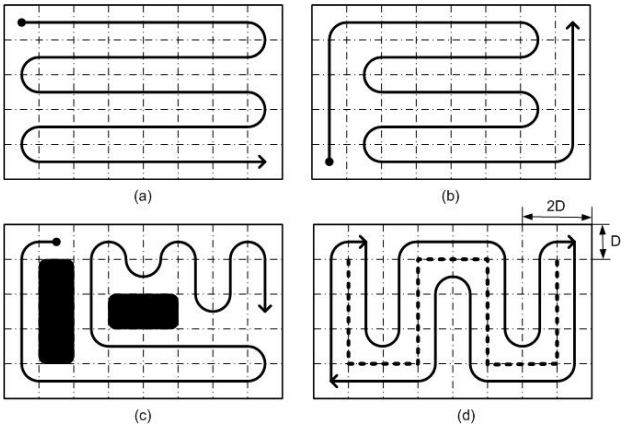


Fig. 7. Illustrations of Hamiltonian path that covers the whole area. (a) A Hamiltonian path. (b) Another Hamiltonian path. (c) A Hamiltonian path in the environment with obstacles. (d) A Hamiltonian cycle generated by the STC method.

4. An application of synchronization motion: Area Patrolling

In this section, we apply the segment synchronization into a multi-robot area patrolling problem.

Consider assigning an N homogeneous mobile robot system $\mathcal{S}$ to patrol a given 2D area, which has its patrolling interest uniformly distributed. We will first partition the planar area into grids, and by finding a Hamiltonian path, we can simplify the 2D patrolling problem into a 1D patrolling case.

In the mathematical field of graph theory, Hamilton graph is well associated with the salesman's problem.

Definition 3: [9](Hamiltonian path) Given a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, and two vertices $s, t \in \mathcal{V}$, a Hamiltonian path (traceable path) is a path in $\mathcal{G}$ from s to t that goes through every vertex $\mathcal{V}$ exactly once.

Definition 4: [9](Hamiltonian cycle) A Hamiltonian cycle (Hamiltonian circuit, vertex tour or graph cycle) is a cycle that visits each vertex exactly once (except the vertex which is both the start and end, and so is visited twice). A graph that contains a Hamiltonian cycle is called a Hamiltonian graph.

Patrolling in a 2D area can be converted to the problem of finding a Hamiltonian path. When a robot moves along the path, its sensor or effector covers the area eventually. Assume the robot sensor covers a rectangular area, we can use the regular grid-based decomposition to partition the area. In Fig. 7, we show several demonstrations of the Hamiltonian path (arrowed path) covering a partitioned area according to grid-based-partition method (dot-dash line). Two different Hamiltonian paths are shown in Fig. 7(a) and 7(b). Another Hamiltonian path over an area with obstacles (shaded area) is shown in Fig. 7(c).

Also, in [10], Gabriely and Rimon introduce a Spanning Tree Coverage (STC) method. The authors assume that a single robot is with a sensing range of D, then partition the area into cells that each cell has the size of $2D \times 2D$. Then, by building

a spanning tree according to the cell size, a Hamiltonian cycle visits all cells of the domain by following the tree around. An illustration of STC method is shown in Fig. 7(d), in which the dotted line is the spanning tree, the arrowed path is a Hamiltonian cycle around the spanning tree. Note that a Hamiltonian path can be generated from the Hamiltonian cycle by breaking the circle at any point.

Other works on "finding a minimal path that covers the whole area" have been solved by different researchers. Other methods on generating 1D path can be found in [6][14], for example.

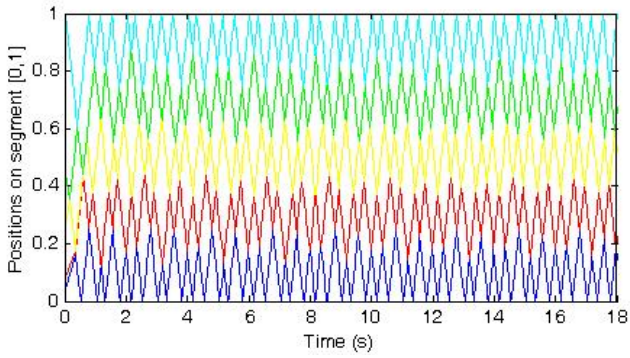
Remark 3: We have transformed the original 2D area patrolling problem into a 1D motion synchronization problem by applying the Hamiltonian path. Then, we can simply apply the synchronization algorithm in Section 3 to solve the multi-robot patrolling problem in a 2D area.

5. SIMULATION RESULTS

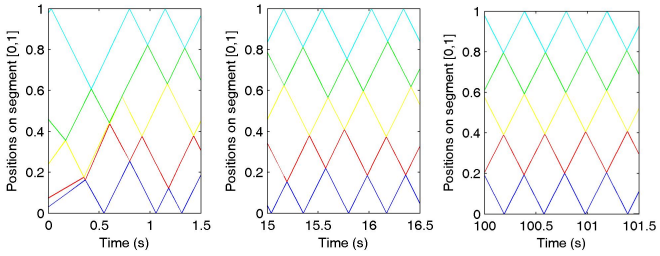
As shown in Fig. 8, a 5-robot system reaches synchronization on the segment $[0, 1]$. At time $t = 0$, the position vector and velocity vector of the system are [0.0299 0.0737 0.2393 0.4585 0.9791] and [0.3643 0.2898 0.6868 -0.6052 0.7126] respectively. We choose the parameter in (9) as $a_1 = 0.43$, and $a_2 = -0.31$. The system tends to reach synchronization by its trajectory uniformly distributing along the segment. The subsegments are [0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1]. Each robot moves along an equal-length subsegment, back and forth at the same speed $v_{ss} = 1$, which can be seen in Fig. 8(d) as the slope of each single short line is '1' at time $t = 100$ s.

In Fig. 9, we simulate the scenario that at time $t = 122.7$s, a robot is suddenly taken out, which is illustrated as a vertical line from 0.5 to 0 at 122.7 sec. The other three robots will adapt to such dynamic change and reach a new synchronization configuration by uniformly distributing along the segment, and the equal length subsegments are [0, 0.333], [0.333, 0.667], [0.667, 1].

In Fig. 10, we demonstrate the case that 2 robots are added into the system at time point 202.4s, at $x_1 = 0.35$ and $x_2 = 0.6$ with the velocity $v_1 = 0.342$ and $v_2 = -0.874$. It shows that the system reaches a new synchronization configuration in about 15 seconds.



(a)



(b) t=0s (c) t=15s (d) t=100s

Fig. 8. Simulation results of 5-robot system synchronization on the segment [0,1]. (a) The overall system trajectory. (b) The zoom-in snapshot at t = 0s. (c) The zoom-in snapshot at t = 15s. (d) The zoom-in snapshot at t = 100s.

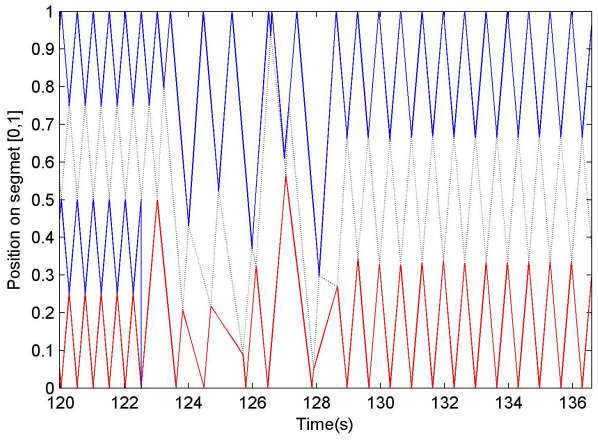


Fig. 9. The system response when a robot is taken out at time 122.7 sec

6. EXPERIMENTAL RESULTS

We validate the synchronization algorithm on an e-puck multi-robot testbed. The e-puck mobile robot is a commercial robot developed for educational purpose by Swiss Federal Institute of Technology in Lausanne. It is equipped with two step motors running at the maximal speed of 15 cm/s, a 16 bit microprocessor with around 15 MIPS by Microchip, 8 infrared (IR) sensor modules, a bluetooth communication module, a VGA camera, a 3D accelerometer, an on-board speaker, and 3 omni-directional microphones. Due to the sporadic communication scheme in our algorithm, we employ the IR sensors functioning at either the proximity detection or the communication mode at different time divisions. *The proximity detection mode* functions to detect obstacles as each IR module sends out a beam of IR light, then measures the characteristics of the reflected signal. *The communication mode* functions as each IR sensor transmits and receives frequency modulated signals within the IR frequency range. It is based on the *libIrcom* library package in [19] for the local range communication among e-pucks, which allows up to 25 cm, at a data rate of 30 Byte/sec.

Our experimental testbed is as shown in Fig. 11(a). The track being patrolled is about 105 cm long, which is built on 52 Lego bricks. The robots P_1 and P_2 in the middle of the track are called *patrolling robots*, which are enabled with 2 step motors, 8 infrared sensors and a 32-bit timer. In order to identify the boundaries of the track, we assign two robots B_0

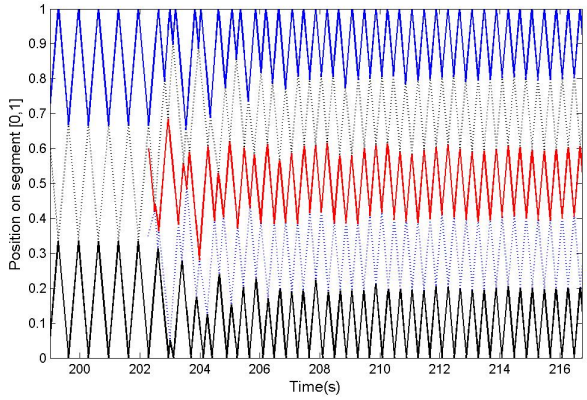
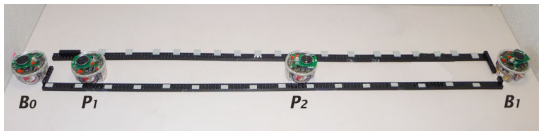


Fig. 10. The system response when another two robots are added into the system at time 202.4 sec

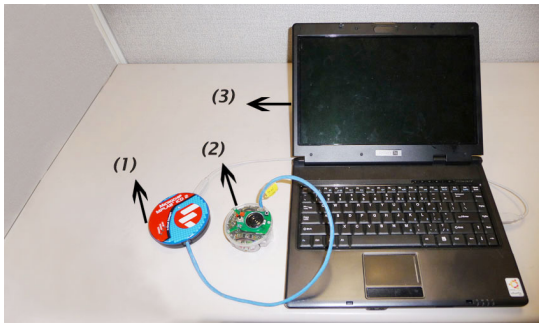
and B_1 sitting at two ends of the track, called *boundary robots*, which are enabled with 8 IR sensors only. The boundary robots are introduced to overcome the limits on the robot sensing and communication capabilities. Alternatively, it can be replaced by 2 static IR sensors mounted on the boundaries. Fig. 11(b) shows our environment setup, which includes a Microchip in-circuit debugger(ICD), a laptop with the MPLAB Integrated Development Environment (IDE) loaded, and a 6-pin RJ-25/Micro-MaTch connection cable.

The multi-robot system functions as follows. Initially, the boundary robots are set at the IR proximity detection mode. It switches to IR communication mode instantly, once another patrolling robot is approaching. It sends out binary codes with all '1's, indicating its identity as a boundary. After a 200 ms delay, it switches back to the proximity detection mode. Each patrolling robot moves at a constant speed on the track at the proximity detection mode. Once it detects any approaching robot, two step motors stop, the timer stops, as well as the IR modules switch to the communication mode. It sends out binary codes consisting of the velocity, the direction, the timer and parity check information, and captures the transmitted signal from neighbors. If the receiving signal is with all '1's, a boundary robot is approaching, and the boundary-hit law applies. Otherwise, either face-face/face-tail type control law applies upon velocity comparison.

We show the system performance in Fig. 12. Initially, as in Fig. 12(a), $P1$ and $P2$ are placed at 4 cm, 46 cm respectively from the left end of the track, with initial velocity 3.30 cm/sec and -2.85 cm/sec (the initial motor speed is 220 and -190, while the step motor speed ranges from -1000 to 1000). In Fig. 12(b) they meet at the first time. In Fig. 12(c), $P1$ hits the boundary for the first time, while $P2$ keeps moving. In Fig. 12(d), they meet around the center of the track. In Fig. 12(e), they hit boundaries at the same time. In Fig. 12(f), they meet at the center of the track. In Fig. 12(g), they approach the boundaries at the same pace. From Fig. 12(e), 12(f), 12(g) the system motion partitions the segment into equal length, and each patrolling robot visit the subsegment in the same time span. Thus, we conclude the system reaches synchronization. Fig. 13 shows that the system reaches synchronization at 152 s as two patrolling robots travel equal length (half of the track) within equal amount of time, which takes around 39 s between two consecutive impacts. It is a successful proof of principles, on a platform that is not equipped with localization sensors.



(a)



(b)

Fig. 11. The experimental platform overview. (a) The snapshot of the experimental testbed, in which B_0, B_1 are boundary-sitting robots, and P_1 and P_2 are patrolling robots. (b) The snapshot of the debugging and programming toolset, in which (1) is the MPLAB in-circuit debugger, (2) is an e-puck mobile robot, and (3) is a laptop with the MPLAB Integrated Development Environment (IDE) installed onboard

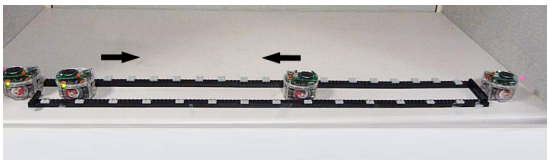
We experience some critical communication delays when conducting this experiment. These delays are caused by unidentified IR readings, which could be either proximity signals or communication signals. In order to reduce such delays, a series of the signal samplings are carried out to test the consistency of the IR signal. If a dubious signal is consistent, it is a proximity signal; otherwise, the signal is probably a disturbing communication signal from a distant robot.

7. Conclusion

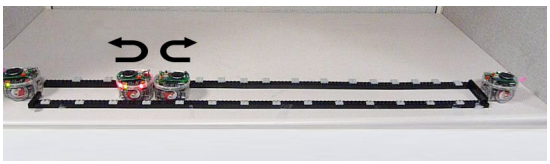
In this paper, we present a solution to multi-robot synchronization on a line segment with sporadic communications. In our algorithm, it does not require any localization of the robots. Instead, each robot updates its velocity based on the time span between two consecutive impacts. We then apply the synchronization to a planar patrolling problem through the Hamiltonian path. Our solution guarantees that each point in the area is visited with a uniform frequency. Simulations and experimental results validate our algorithm, and show the efficiency and feasibility of the method.

ACKNOWLEDGMENT

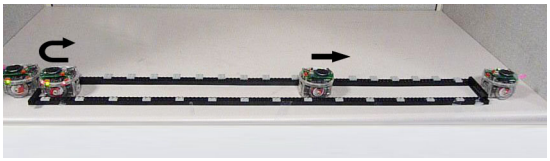
The work was supported in part by US Army contract W15QKN-05-D-0011, and by the National Science Foundation under Grants CMMI 0825613 and DUE 0837584.



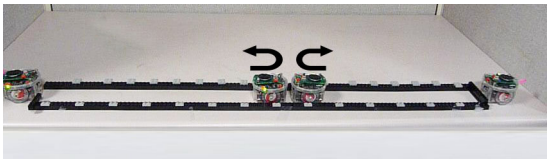
(a) At t = 0



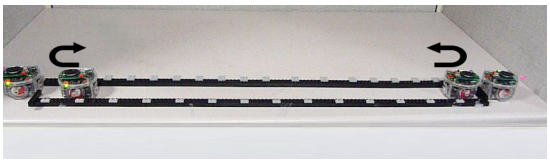
(b) At t = 19s



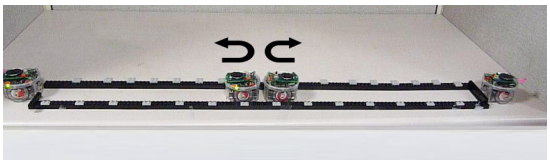
(c) At t = 41s



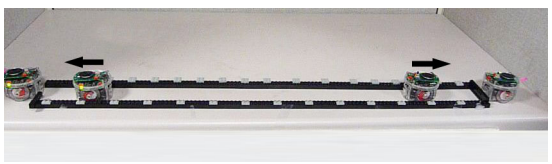
(d) At t = 82s



(e) At t = 113s



(f) At t = 152s



(g) At t = 176s

Fig. 12. Snapshots taken from the experiment platform at intervals. (a) The initial positions of the system. (b) The first time meet-up (c) The robot to the left touches the boundary for the first time (d) The second meet-up (e) The patrolling robots hit the boundaries simultaneously (f) The third meet-up at the center of the track (g) The patrolling robots approaching the boundaries at the same pace.

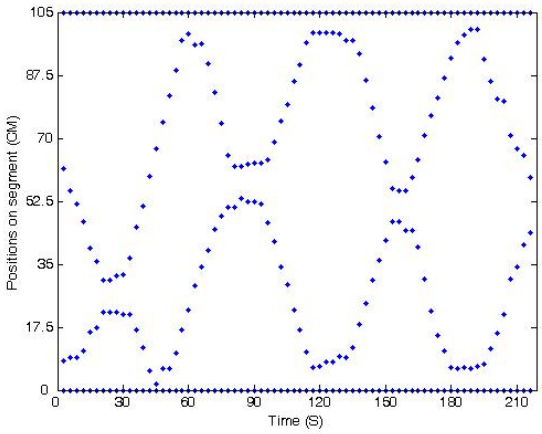


Fig. 13. Experimental result showing the system synchronization in around 152 s.

REFERENCES

[1] M. B. Sevryuk, "Estimate of the number of collisions of N elactic particles on a line," *Theoretical and Mathematical Physics*, Vol. 96, Issue 1, pp. 64-78, 1993.

[2] S. L. Glashow and L. Mittag, "Three rods on a ring and the triangular billiard," *Journal of Statistical Physics*, Vol. 87, No. 3-4, pp. 937-941, 1997.

[3] B. Cooley and P. K. Newton, "Iterated impact dynamics of N-beads on a ring," *SIAM Review*, Vol. 47, Issue 2, pp. 273-300, 2005.

[4] S. Susca and F. Bullo, "Synchronization of beads on a ring," *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, pp. 4845-4850, Dec. 2007.

[5] D B. Kingston, R W. Beard, and R Holt, "Decentralized perimeter surveillance using a team of UAVs," *IEEE Transactions on Robotics*, Vol. 24, No. 6, pp. 1394-1405, 2008.

[6] Y. Elmaliach, N.Agmon, and G. A. Kaminka, "Multi-robot area potrol under frequency constraints," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, pp. 385-391, 2007.

[7] K. Williams and J. Burdick, "Multi-robot boundary coverage with plan revision," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, pp. 1716-1723, May 2006.

[8] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. Mclain, S.-M. Li, and R. Mehra, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Sciences*, Vol. 37, No. 6, pp. 351-360, 2006.

[9] G. Chartrand, *Introductory Graph Theory*, Courier Dover Publications, 1985.

[10] Y. Gabriely and E. Rimon. "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and Artificial Intelligence*, Vol. 31, No. 1-4, pp. 77-98, 2001.

[11] N. Agmon, N. Hazon, and G. A. Kaminka, "Constructing spanning trees for efficient multi-robot coverage," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, pp. 1698-1703, May, 2006.

[12] D. B. Kingston, "Decentralized control of multiple UAVs for perimeter and target surveillance," Ph.D. Diss., Brigham Young University, Dec., 2007.

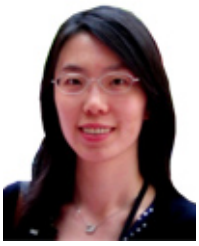
[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," *INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, San Franciso, CA, pp. 1293-1303, 2003.

[14] Y. Guo and M. Balakrishnan, "Complete coverage control for nonholonomic mobile robots in dynamic environments," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, pp. 1704-1709, May 2006.

[15] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli, "On synchronous robotic networksłpart I: models, tasks, and complexity," *IEEE Transactions on Automatic Control*, Vol.52, No.12, pp. 2199-2213, Dec. 2007.

[16] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli, "On synchronous robotic networksłpart II: time complexity of rendezvous and deployment

algorithms," *IEEE Transactions on Automatic Control*, Vol. 52, No. 12, pp. 2214-2226, Dec. 2007.
[17] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, Vol. 50, No. 2, pp. 169-182, 2005.
[18] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a Robot Designed for Education in Engineering," *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Castelo Branco, Portugal, pp. 59-65, 2009.
[19] The e-puck education robot - homepage, http://www.e-puck.org/, EPFL, Mar. 2010.
[20] X. Long, J. Jiang, and K. Xiang, "Towards multirobot communication," *IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, pp. 307-312, Aug. 2004.
[21] T. Rupp, T. Cord, R. Lohnert, and D.E. Lazic, "Positioning and communication system for autonomous guided vehicles in indoor environment," *The 9th Mediterranean Electrotechnical Conference (MELECON'98)*, Vol.1, No., pp. 187-191, May 1998.
[22] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. Mclain, and M. Goodrich, "Autonomous vehicle technologiesfor small fixed wing UAVs," *AIAA J. Aerosp. Comput. Inf. Commun.*, Vol. 2, No. 1, pp. 92-108, Jan. 2005.



Hua Wang is currently a Ph.D candidate in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. She received her B.S. and M.S. degrees from Harbin Institute of Technology, China in 2004 and 2006 respectively. She is currently interested in autonomous mobile robotic system, topology control and complex network systems.

Yi Guo is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. She obtained her Ph.D. degree in Electrical and Information Engineering from University of Sydney, Australia, in 1999. She was a postdoctoral research fellow at Oak Ridge National Laboratory from 2000 to 2002, and a visiting Assistant Professor at University of Central Florida from 2002 to 2005. Her research interests are mainly in nonlinear systems and control, autonomous mobile robotics, and control of multi-scale complex systems. Dr. Guo is a Senior Member of IEEE.