The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Synchronization on a Segment Without Localization: Algorithm and Applications

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 law, which does not depend on the position of the robot, 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. Simulation results show the feasibility and robustness of our algorithm.**

I. INTRODUCTION

The problem of N-beads sliding freely on a curve with collision fascinates researchers for long. Sevryuk in [1] elucidates some fundamental results when the N-bead colliding elastically on a *line*, and prove the total number 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 makes the problem of "N-beads slide 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 problem via matrix products. Each collision sequence is taken as a billiard trajectory in a right triangle with non-standard reflection rules, and it is proven the existence of the periodic orbits.

Imitating the impact behavior of N-beads' collective motion, and designing 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, 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 paths 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 the 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. Furthermore, a revision algorithm is proposed therein to revise paths in cases that the team size or the environment changes.

In this paper, we consider 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 a 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 total number of robots, nor the total length of the line segment be known by the robots. While similar ideas appeared in [4] and [5], some distance measurement to critical points or priori knowledge such as the perimeter length or the total robot number is required. We relax such assumptions, and use only the information of 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 ignored 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 uniform frequency.

The rest of the paper is organized as follows. In section II, we formulate the system model, define the synchronization problem and desired system behavior. In section III, we propose an algorithm by designing impact laws for different

Hua Wang and Yi Guo are with the Electrical and Computer Engineering Department, Stevens Institute of Technology, Hoboken, NJ 07030, USA hwang@stevens.com; yguo1@stevens.com

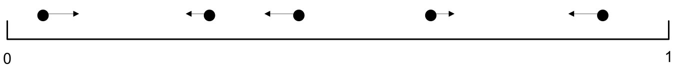

Fig. 1. A demonstration of a 5 robot system moving along the segment [0,1]

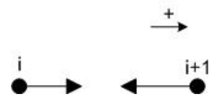

Fig. 2. An illustration of head-head type impact between robots i and $i+1$

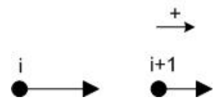

Fig. 3. An illustration of head-tail type impact between robots i and $i+1$

impact types, and analyze the stability of this system. In section IV, we apply the synchronization problem into a planar patrolling problem using the Hamiltonian path. We demonstrate the satisfactory performance using simulations in section V. Finally, we conclude in section VI.

II. 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 as in Figure 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 $i \in [1, 2, \ldots N]$, 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 switches 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 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 with a constant speed, the constant speed continuous dynamic can be model 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 constant dynamic is possible.

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 $H := (f, C, g, D)$.

Remark 1: The assumptions imply that, robot $i \in 2 \ldots N-1$ 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 a segment, 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

$$\forall i \in \mathcal{S}, |v_i| = v_{ss} \neq 0 \tag{4}$$

(2) For any robot i, the time span between it impacts robot $i+1$ and $i-1$ is a constant t_{ss}, that is

$$\Delta t_i(k_i - 1, k_i) = \Delta t_i(k_i, k_i + 1) = t_{ss} \tag{5}$$

where

$$\begin{aligned} \Delta t_i(k_i - 1, k_i) &= t_{k_i} - t_{k_i-1} \\ \Delta t_i(k_i, k_i + 1) &= t_{k_i+1} - t_{k_i} \end{aligned} \tag{6}$$

Such distribution of patrolling goal is effective because each robot patrols equal distance in the same time span t_{ss}. So each subarea is visited by a robot 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.

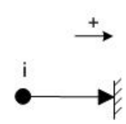

Fig. 4. An illustration of hit-boundary type impact of robot i

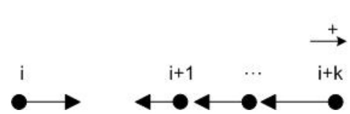

Fig. 5. An illustration of multi-robot hit type impact of robot i, $i+1$, ..., $i+k$

Problem 1: Under assumptions 1 to 4, find a decentralized control algorithm for (2) such that robots achieve synchronization on the segment [0,1].

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

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

- Head-Head Type Impact: As shown in figure 2, head-head type happens between robots i and $i + 1$, when $v_i v_{i+1} < 0$.
- Head-Tail Type Impact: As shown in figure 3, 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 figure 4, such impact happens either $v_1 < 0$ or $v_N > 0$.
- Multi-hit Type Impact: As shown in figure 5, multi-hit type happens when the robots $i, i+1, \dots, i+k$, $k \geq 2$ bump into the same position. Given the velocity and position vector p and v at time t_0, under the condition

$$\begin{cases} v_i > v_{i+1} > \cdots > v_{i+k} \\ p_i < p_{i+1} < \cdots < p_{i+k} \end{cases} \tag{7}$$

  there exists time $t^\diamond > 0$ that agents $i, i+1 \dots i+k$ bump into each other at $p^\diamond$.

$$p^\diamond = p_i + t^\diamond v_i = p_{i+1} + t^\diamond v_{i+1} = \cdots = p_{i+k} + t^\diamond v_{i+k}$$

III. 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 impact happens (i.e., when they meet). Then, we define different updating law when different type of impact happens. "Constant velocity" means that the robot moves along a straight line without any changes of the magnitude and 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.

A. Initialize state vector

At time $t = 0$, according to the assumptions defined in Section III, 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) $v_i^{init} \in [-1, 1] \backslash 0$;
3) $k_i^{init} = 0$;
4) $t_{k_i-1}^{init} = 0$.

Furthermore, $k_{z_0}^{init}$, $t_{k_{z_0}-1}^{init} = 0$, $k_{z_1}^{init} = 0$, $t_{k_{z_1}-1}^{init} = 0$.

Once initialized with position and velocity, robots are free to move on the segment at the uniform velocity, until the first impact happens. We then come up with three 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.

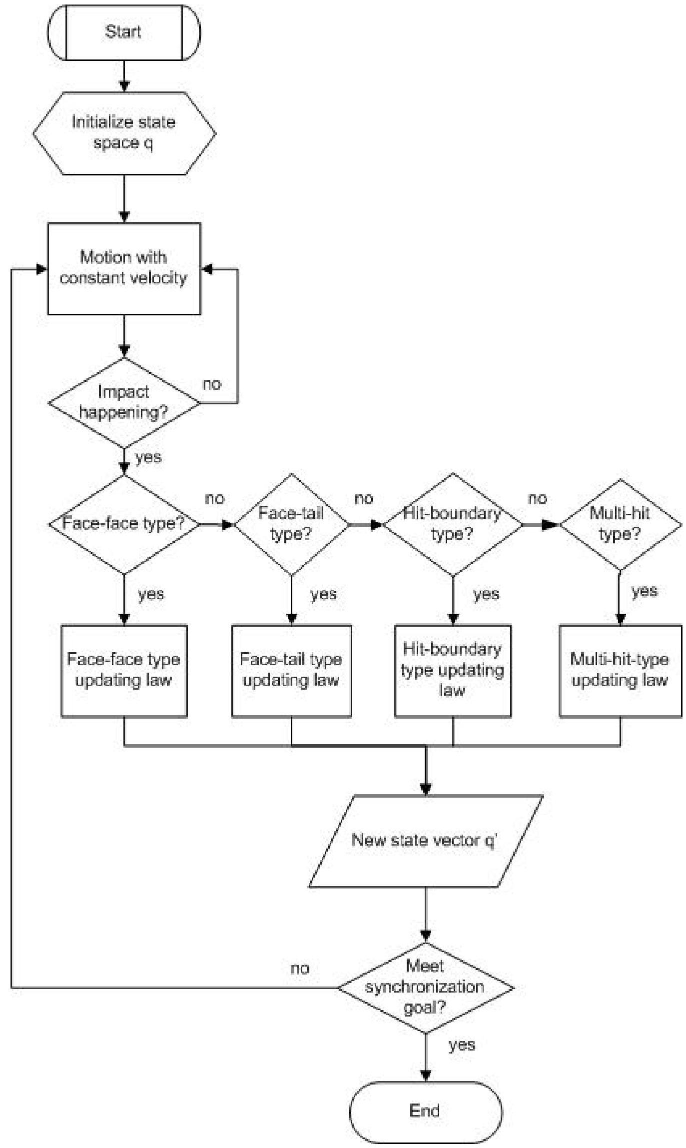

Fig. 6. A flowchart describes the process of synchronization algorithm

B. Motion in constant velocity

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

$$f(q) \overset{\text{def}}{=} \begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = 0 \end{cases}, \quad q \in C \tag{8}$$

where $C = \{q : x_j \neq x_k, \forall i, j \text{ in } \ S\}$.

C. Face-face type updating law

At t_p, an 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; the velocity of i and $i+1$ are to be reversed, and are updated according to the difference of the time span $\Delta t_i(k_i - 1, k_i)$ and $\Delta t_{i+1}(k_{i+1} - 1, k_{i+1})$, which are the time spans from t_p^+ traced back to the last impacting instant t_{k_i-1} and $t_{k_{i+1}-1}$ for robot i and $i + 1$, respectively. In the following, we denote $\Delta t_i(k_i - 1, k_i)$ and $\Delta t_{i+1}(k_{i+1} - 1, k_{i+1})$ as Δt_i and Δt_{i+1}. Next, the impacting time counter k_i and k_{i+1} is accumulated by one, since one more impact just happened. Finally, the recall timer 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 instant t_p.

The velocity updating law is originated from the idea of feedback control. Since the algorithm is thoroughly decentralized, one is only able to communicate with its neighbors. We want the magnitude of the velocity to consensus to v_{ss}. Thus, for agent i, when its latest impact time-span is shorter than its neighbors, its velocity should be lowered, and vice versa. Once the time span of two impacting robots are the same, it should be set as our 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:

$$g_1(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = -sgn(v_i)(v_{ss} + a_1(\Delta t_i - \Delta t_{i+1})v_{ss}v_i) \\ v_{i+1}^+ = \\ \quad -sgn(v_{i+1})(v_{ss} + a_2(\Delta t_{i+1} - \Delta t_i)v_{ss}v_{i+1}) \\ \qquad\qquad\qquad q \in D_1 \end{cases} \tag{9}$$

where $D_1 = \{q : x_i = x_j, v_iv_j < 0\}$, a_1, a_2 are parameters that work as the feedback gain, $|a_1| < 1, |a_2| < 1$, and $a_1 < 0, a_2 > 0$ if $\Delta t_{i+1} > \Delta t_i$), vice versa. And also, $k_i^+ \leftarrow k_i + 1$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_i-1}^+ \leftarrow t_p$, $t_{k_{i+1}-1}^+ \leftarrow t_p$. And sgn is the sign function that extracts the sign of a real number. So the $-sgn(v_i[k])$ plays the role that reverse the velocity of i after the impact at time t_p.

D. Face-tail type updating law

Considering the face-tail type impact happens at t_p between robot i and $i+1$, $(i, i+1 \in \mathcal{S})$. At t_p^+, the positions of i and $i+1$ remain the same; the velocity magnitude v_i will be swapped with v_{i+1}, the direction of v_{i+1} will be reversed; the impacting time counter k_i and k_{i+1} is added by one, the recall timer 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 instant t_p.

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

$$g_2(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = v_{i+1} \\ v_{i+1}^+ = -v_i \end{cases}, \quad q \in D_2 \tag{10}$$

where $D_2 = \{q : x_i = x_j, v_iv_j > 0, v_i(|v_i| - |v_{i+1}|) > 0\}$. And the timer and counter update: $k_i^+ \leftarrow k_i + 1$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_i-1}^+ \leftarrow t_p$, and $t_{k_{i+1}-1}^+ \leftarrow t_p$.

E. Hit-boundary type updating law

At t_p, suppose robot $i, (i \in \mathcal{S})$ hits boundary z_0 (or z_1). Taken as a virtual static robot, the boundary has constant zero velocity and fixed position. At t_p^+, The positions of i is intact; the velocity v_i will be reversed, and the magnitude will be tuned at the ultimate velocity v_{ss}. The impacting time counters k_i and k_{z_0} (or k_{z_1}) are added by one, the recall timers t_{k_i-1} and $t_{k_{z_0}-1}$(or $t_{k_{z_1}-1}$) are replaced by the newest impact moment t_p. The other robot state vectors are intact at instant t_p.

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

$$g_3(q,t_p) \overset{\text{def}}{=} v_i^+ = -sgn(v_i)v_{ss}, \quad q \in D_3 \tag{11}$$

where $D_3 = \{q : \{x_1 = 0, v_1 < 0\} \bigcup \{x_N = 1, v_N > 0\}\}$. And timer and counter update as: $k_i^+ \leftarrow k_i + 1$, and $t_{k_i-1}^+ \leftarrow t_p$.

F. Multi-hit type updating law

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

$$g_4(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = -v_i \\ v_{i+1}^+ = -v_{i+1} \\ \dots \\ v_{i+k}^+ = -v_{i+k} \end{cases}, \quad q \in D_4 \tag{12}$$

where $D_4 = \{q : x_i = x_{i+1} \dots = x_{i+k}, v_i < v_{i+1} \dots < v_{i+k}\}$. And again, counter and timer update:$k_i^+ \leftarrow k_i + 1$, $t_{k_i-1}^+ \leftarrow t_p$, $k_{i+1}^+ \leftarrow k_{i+1} + 1$, $t_{k_{i+1}-1}^+ \leftarrow t_p$, $\dots$, $k_{i+k}^+ \leftarrow k_{i+k} + 1$, $t_{k_{i+k}-1}^+ \leftarrow t_p$.

G. Convergence analysis of the dynamic system

Next, we discuss the convergence of the velocity magnitude $|v_i|$ and the elapsed time Δt_i between impacts in the hybrid dynamic system H.

We first take a look at the velocity magnitude. In the flow set C, the continuous dynamic defined in (8) does not change the magnitude 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 (12) do not change the magnitude of velocity, instead, it reverses the direction of the velocity. In the boundary-hit type, we can see every time it hits the boundary, its velocity magnitude will be set to the ultimate velocity v_{ss}. Next, we analyze the face-face type impacting law defined in (9). v_{ss} is a common term, without loss of generality, we write v_i^*, v_j^* as $v_i - sgn(v_i)v_{ss}$, $v_{i+1} - sgn(v_{i+1})v_{ss}$ respectively. where

$$\begin{cases} {v^*}_i^+ = -sgn(v_i^*)a_1(\Delta t_i - \Delta t_{i+1})v_{ss}v_i^* \\ {v^*}_{i+1}^+ = -sgn(v_{i+1}^*)a_2(\Delta t_{i+1} - \Delta t_i)v_{ss}v_{i+1}^* \end{cases} \tag{13}$$

Since we only discuss the velocity magnitude $|v_i|$, we ignore the sgn function, which only changes the direction. Obviously, v_i^* and v_{i+1}^* are decoupled, thus, we will analyze v_i^* next. Since the robot moves in a constant velocity between the impacts on a segment with length 1, the Δt_i is upper and lower bounded as

$$\frac{1}{max(v_i)} \leq \Delta t_i \leq \frac{1}{min(v_i)} \tag{14}$$

which implies

$$0 \leq |\Delta t_i - \Delta t_{i-1}| \leq \frac{1}{min(v_i)} \tag{15}$$

Since the velocity v_i is upper and lower bounded, we can write $e_1 v_{ss} < |v_i| < e_2 v_{ss}$, where $0 < e_1 < 1, e_2 > 1$. Actually, by choosing proper parameter a_1 and a_2, v_i can be guaranteed to be bounded for any given e_1 , e_2. Thus $min(|v_i|) = e_1 v_{ss}$. And we consider the ratio $r_k = |\frac{v^{*+}_i}{v^*_i}|$ when the kth face-face type impact happens,

$$r_k = |\frac{v^{*+}_i}{v^*_i}| = |a_1(\Delta t_i - \Delta t_{i+1})v_{ss}| \tag{16}$$

$$\leq |a_1 \frac{v_{ss}}{min(v_i)}| = \frac{|a_1|}{e_1} < 1 \tag{17}$$

for any given $|a_1| < e_1$. It can be seen that the velocity magnitude $|v_i^*|$ is 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 well. Then, the magnitude of v_i converges to v_{ss} as $t \to \infty$. Moreover, $\Delta t_i = \Delta t_{i+1}$ for any pair, the elapsed time between impacts for robot i and j reaches consensus. Thus, the system reaches synchronization on the segment.

IV. 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 figure 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 figure 1(a) and 1(b). Another Hamiltonian path over an area with obstacles(shaded area) is shown in figure 1(c).

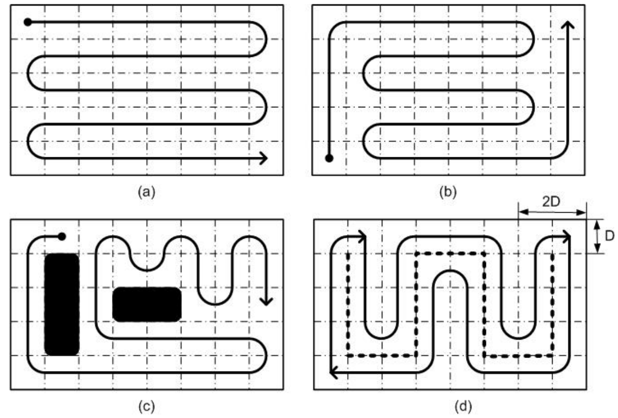

Fig. 7. A series of illustrations of Hamiltonian path that covers the whole area. a) A Hamiltonian path illustration. b) Another Hamiltonian path illustration. c) A Hamiltonian path in the environment with obstacles. d) An illustration of Hamiltonian cycle generated by STC method.

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 figure 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 use the synchronization algorithm in Section III to solve the multi-robot patrolling problem in a 2D area.

V. Simulation results

As shown in Fig. 8, an 6-robot system reaches synchronization on segment $[0, 1]$. At time $t = 0$, the position vector and velocity vector are [0.0960 0.2843 0.3708 0.5275 0.5456 0.9811] and [-0.8706 0.0896 0.6728 -0.7094 -0.6570 -0.8639] respectively. We choose the parameter in (9) as $a_1 = 0.92$, and $a_2 = -0.84$. The system tends to reach synchronization by its trajectory uniformly distributing along the segment. Each subsegment is [0, 0.167], [0.167, 0.333], [0.333, 0.5], [0.5, 0.667], [0.667, 0.833], [0.833, 1], each robot moves along an equal length subsegment, back and forth at the same speed $v_{ss} = 1$, which can be seen in the figure as the slope of each single short line is all the same at time $t = 6$.

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

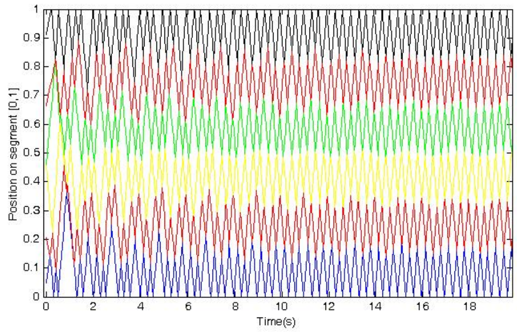

Fig. 8. Simulation result of 8-robot system synchronization on the segment [0,1]

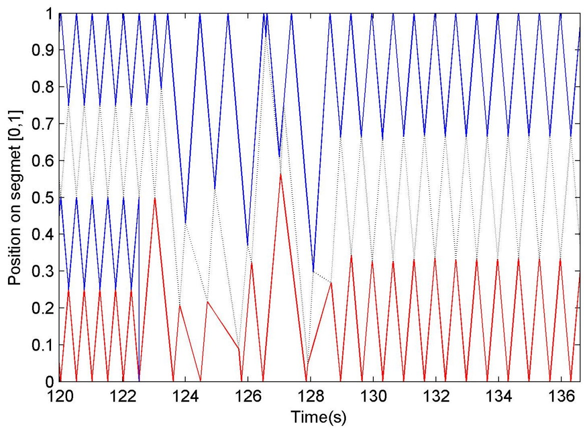

Fig. 9. The system response when a robot is taken out at time 122.7

three robots will adapt to such dynamic change and reaches 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 the system reaches a new synchronization configuration in about 15 seconds.

VI. Conclusion and future work

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

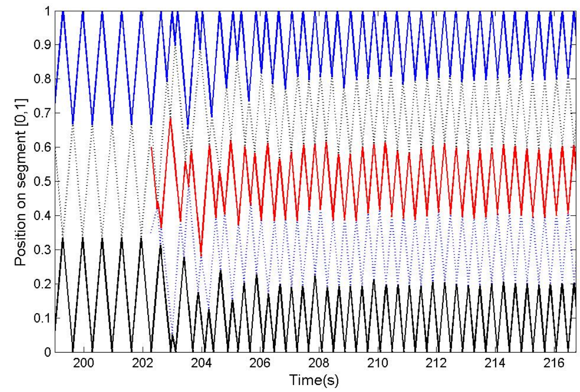

Fig. 10. The system response when two other robots are added into the system at time 202.4 sec

Acknowledgement

The work was supported in part by US Army contract W15QKN-05-D-0011.

References

[1] M. B. Sevryuk, Estimate of the number of collisions of N elactic particles on a line, *Theoretical and Mathematical Physics*, 96(2003), pp 64-78.

[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, 1997, pp. 937-941.

[3] B. Cooley and P. K. Newton, Iterated Impact Dynamics of N-Beads on a Ring, *SIAM Review*, Vol. 47 , Issue 2(2005), pp. 273-300.

[4] S. Susca and F. Bullo, Synchronization of beads on a ring, Decision and Control, 2007 46th IEEE Conference on , vol., no., pp.4845-4850, 12-14 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, *IEEE ICRA 2007*, Roma, Italy, 2007, pp. 385.

[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, May 2006, 1716-1723.

[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, ISBN 0486247759, 9780486247755, 294 pages.

[10] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77-98, 2001.

[11] N. Agmon, N. Hazon and G. A. Kaminka, Constructing spanning trees for efficient multi-robot coverage, *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, vol., no., pp. 1698-1703, May 15-19, 2006

[12] D. B. Kingston, Decentralized control of multiple UAVs for perimeter and target surveillance, *Doctor of Philosophy thesis*, Brigham Young University, December, 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*, vol.2, no., pp. 1293-1303 vol.2, 30 March-3 April 2003.

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