Distributed Cooperative Control System Algorithms – Simulations and Enhancements

Po Wu and Panos J. Antsaklis

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Email: {pwu1, antsaklis.1}@nd.edu

Abstract

In this paper, we describe and simulate a number of algorithms created to address problems in distributed control systems. Based on tools from matrix theory, algebraic graph theory and control theory, a brief introduction is provided on consensus, rendezvous, and flocking protocols, and Dubins vehicle model. Recent results are then shown by simulating and enhancing various multi-agent dynamic system algorithms.

Index Terms

Cooperative systems, distributed control, multi-agent system

I. INTRODUCTION

Distributed control systems refer to control systems in which the controller elements are not centralized but are distributed throughout the system with each component sub-system controlled by one or more controllers. The entire system of controllers is connected by networks for communication and monitoring. In such multi-agent systems, various control algorithms to achieve different purposes are considered, such as the attitude alignment in multiple spacecraft setting [3][28], formation control of unmanned air vehicles [1], and flocking [2]. See also [27].

Consensus problems under different information constraints have been addressed by many researchers [16]. Jadbabaie et al. [3] focus on coordination under undirected graphs. Ren et al. [4] extend the result to the directed graph case. Average consensus problem is solved over balanced directed networks in [5]. The speed of consensus can be increased by weight optimization [20], or by via random rewiring [6] since the algebraic connectivity of a regular network can be greatly enlarged. The robustness to changes in network topology due to link/node failures, time-delays, and performance guarantees is analyzed in [7]. Asynchronous consensus has been studied in [8][17][18][19].

Rendezvous problems have been introduced in [9], in which all agents are homogeneous and memoryless. It is shown in [10] that synchronous and asynchronous rendezvous is achieved with the initial graph connected. The circumcenter algorithm is designed [11] to set target points as circumcenters. A related algorithm, in which connectivity constraints are not imposed, is proposed in [12].

Flocking, which means convergence to a common velocity vector and stabilization of inter-agent distances, is guaranteed as long as the position and velocity graphs remain connected at all times [13]. [14] provides a stability result for the case where the topology of agent interconnections changes in a completely arbitrary manner. The split/rejoin maneuver and squeezing maneuver are performed with obstacle avoidance.

Taking into account inherent kinematic limitations of automobiles, Dubins vehicle is introduced in [21]. A model of Dubins vehicle is a controllable wheeled robot with a constraint on the turning angle along a given route in two dimensions. [22] gave the shortest paths joining two arbitrary configurations using Optimal Control Theory.

Rendezvous of Dubins vehicles is discussed in [25].

Simulation and enhancement of distributed control algorithms mentioned above provides illustrative examples of how the dynamic system evolves. The discrete event simulator is implemented in Java as a class with heading, position, velocity and other variables, and with some Java methods for neighborhood updating, moving and status displaying. Communication is accomplished via message exchanging mechanism taking into account noise. The behavior of every agent is illustrated as the 2-D position and trajectories.

The paper is organized as follows. In Section II, we describe recent results of consensus protocols. In Section III, rendezvous problem is shown. In Section IV, flocking problems are discussed. Section V contains the simulation results and Section IV contains concluding remarks and future directions.

II. Consensus protocols and stability theorems

A. *Definition and Notations*

To describe the relationships between multiple agents, we have a digraph G to model the interaction topology. If agent j can receive information from agent i, then graph nodes v_i and v_j correspond to agent i and j, and a directed edge e_{ij} represents a unidirectional information exchange link from v_i to v_j, that is, agent j can receive information from agent i. The interaction graph represents the communication pattern at certain discrete time.

Let $G = \{V, E, A\}$ be a weighted digraph (or direct graph) of order n with the set of nodes $V = \{v_1, v_2, ..., v_n\}$, set of edges $E \subseteq V \times V$, and a weighted adjacency matrix $A = [a_{ij}]$ with nonnegative adjacency elements a_{ij}. The node indices belong to a finite index set $I = \{1, 2, ..., n\}$. A directed edge of G is denoted by $e_{ij} = (v_i, v_j)$, where $e_{ij} \in E$ does not imply $e_{ji} \in E$. The adjacency elements corresponding to the edges of the graph are positive, i.e., $a_{ij} > 0$ if and only if $e_{ji} \in E$. Moreover, we assume $a_{ii} \neq 0$ for all $i \in I$. The set of neighbors of node v_i is the set of all nodes which communicate to v_i, denoted by $N_i = \{v_j \in V : (v_j, v_i) \in E\}$.

A graph G is called strongly connected if there is a directed path from v_i to v_j and v_j to v_i between any pair of distinct vertices v_i and v_j. Vertex v_i is said to be linked to vertex v_j across a time interval if there exists a directed path from v_i to v_j in the union of interaction graphs in that interval. A directed tree is a directed graph where every node except the root has exactly one parent. A spanning tree of a directed graph is a tree formed by graph edges that connect all the vertices of the graph.

Let **1** denote a $n\times 1$ column vector with all entries equal to 1. Let $M_n(R)$ represent the set of all $n\times n$ real matrices. A matrix $F\in M_n(R)$ is nonnegative, $F\geq 0$, if all its entries are nonnegative, and it is irreducible if and only if $(I+F)^{n-1}>0$. Furthermore, if all its row sums are +1, F is said to be a (row) stochastic, while doubly stochastic if it is both row stochastic and column stochastic.

The interaction graph is time-dependent since the information flow among agents may be dynamically changing. Let $\overline{G}=\{G_1,G_2,...,G_M\}$ denote the set of all possible interaction graphs defined for a group of agents. Note that the cardinality of $\overline{G}$ is finite. The union of a collection of graphs $\{G_{i_1},G_{i_2},...G_{i_m}\}$, each with vertex set V, is a graph G with vertex set V and edge set equal to the union of the edge sets of G_{i_j}, $j=1,2,...,m$.

B. Consensus protocols and general stability theorems

Consider the following synchronous discrete-time consensus protocol [4], [7]

$$x_i(t+1)=\frac{1}{\sum_{j=1}^{n}A_{ij}(t)}\sum_{j=1}^{n}A_{ij}(t)x_j(t) \qquad (1)$$

where $t\in\{0,1,2,...\}$ is the discrete-time index, $A_{ij}(t)>0$ if information flows from v_j to v_i at time t. The magnitude of $A_{ij}(t)$ represents possibly time-varying relative confidence of agent i in the information state of agent j at time t or the relative reliabilities of information exchange links between them.

Rewrite (1) in a compact form

$$x(t+1)=F(t)x(t) \qquad (2)$$

where $x=[x_1,...,x_n]$, $F=F_{ij}=\dfrac{A_{ij}(t)}{\sum_{j=1}^{n}A_{ij}(t)}$. An immediate observation is that the matrix F is a nonnegative stochastic matrix, which has an eigenvalue at 1 with the corresponding eigenvalue vector equal to **1**.

Lemma 1 ([7]): Let G be a digraph with n nodes and maximum degree. Then with parameter $\Delta = \max_i(\sum_{j\neq i} a_{ij})$ and $\varepsilon \in (0,1/\Delta]$, *F* satisfies the following properties:

i) *F* is a row stochastic nonnegative matrix with a trivial eigenvalue of 1;

ii) All eigenvalues of *F* are in a unit circle;

iii) If *G* is a balanced graph, then *F* is a doubly stochastic matrix;

iv) If *G* is strongly connected and $0 < \varepsilon < 1/\Delta$, then F is defined as a primitive matrix.

With the connection between the graph *G* and the matrix *F*, we have the stabilization condition of consensus, i.e. the convergence of all agents in fixed topology.

Theorem 1 ([7]): Consider a network of n agents with topology G applying the consensus algorithm (2). Suppose G is a strongly connected digraph, then

i) A consensus is asymptotically reached for all initial states;

ii) The consensus value is $\alpha = \sum_i w_i x_i(0)$ with $\sum_i w_i = 1$;

iii) If the digraph is balanced (or P is doubly stochastic), an average-consensus is asymptotically reached and $x_{ss} = (\sum_i x_i(0))/n$.

In switching topology, we have the following results.

Theorem 2 ([3]): Let $x(0)$ be fixed and let $\bar{G}$ be a switching signal for which there exists an infinite sequence of contiguous, nonempty, bounded, time-intervals, $[t_i, t_{i+1})$, starting at $t_0 = 0$, with the property that across each such interval, the *n* agents are linked together. Then

$$\lim_{t\to\infty} x(t) = x_{ss}1 \tag{3}$$

where x_{ss} is a number depending only on $x(0)$ and $\bar{G}$.

Theorem 3 ([4]): Let $\bar{G}$ be a switching interaction graph. The discrete update scheme (2) achieves consensus asymptotically if there exists an infinite sequence of uniformly bounded, nonoverlapping time intervals $[t_i, t_{i+1})$, $i = 1, 2,\ldots$, starting at $t_0 = 0$, with the property that each interval $[t_i, t_{i+1})$ is uniformly bounded and the union of the graphs across each interval $[t_i, t_{i+1})$ has a spanning tree. Furthermore, if the union of the graphs after some finite time does not have a spanning tree, then consensus cannot be achieved asymptotically.

The algorithms applied above have a relative slow convergence speed. It is shown in [6] that small-world network can make the algebraic connectivity more than 1000 times greater than a regular network, which means that small-world networks go through a spectral phase transition phenomenon achieving ultrafast and more robust consensus. We will show that in the simulation. Here, the algebraic connectivity, which determined the speed of convergence, is defined as:

$$\lambda_2(L) = \min_{1^T x=0, x\neq 0} \frac{x^T Lx}{\| x \|^2} \tag{4}$$

where L is the graph Laplacian defined as $L = D - A$, $D_{ij} = \sum_{j, j\neq i} a_{ij}$

C. Delay in communication

If communication delay exists, the asynchronous continuous time consensus protocol can be described as

$$\dot{x}_i(t) = \sum_{V_j \in N_i} a_{ij}[x_j(t-\tau_i(t)) - x_i(t-\tau_i(t))] \tag{5}$$

Assume that the original graph leads to consensus. Introducing delay into the protocol may affect the performance of the whole distributed control system or even undermine final consensus. If delay τ is bounded, global asymptotical consensus is still achievable. The following table shows delay results both in continuous time (CT) and discrete time (DT).

No.	Results	CT/DT	Value	Time
1	$\tau \leq \frac{\pi}{2\lambda_n}$ Th. 10 in [5]	CT	Uniform	Time-invariant
2	$\tau_i \leq \frac{\pi}{2\lambda_n}$ Th. 5 in[24]	CT	Non-uniform	Time-invariant
3	$\tau_i(t) \leq \frac{1}{\sum_{i,i' \in I} \|\Delta_i \Delta_{i'}\| \cdot \|\Delta^{-1}\|}$ Th. 6 in [24]	CT	Non-uniform	Time-variant
4	$\tau(t) < \frac{3}{2\lambda_n}$ In [25]	CT	Uniform	Time-variant
5	$0 \leq t_j^i(k) \leq B_1 - 1$ Proposition 1 in [26]	DT	Non-uniform	Time-variant

Table 1. A Categorization of exiting consensus delay results

D. Optimization of edge weights

Assume the topology of the graph is known, i.e. $G = \{V, E, A\}$ be a weighted digraph, then the weights of graph Laplacian can be optimized so that maximum consensus rate is achieved. In [20], the optimization problem becomes:

maximize $\lambda_2(L)$

subject to $A^T L(u_1,...,u_r)A - \lambda I \geq 0$, $I - A^T L(u_1,...u_r)A \geq 0$

But even when this LIM problem has a solution, the system is no longer distributed controlled. In order to design a distributed algorithm, we notice from simulations that generally link losses happen when neighbors are at the edge of detection range. Thus increasing weights corresponding to distance will prevent link dropping.

III. FLOCKING

Flocking is introduced in [2], with three flocking rules of Reynolds:

i) Flock Centering: attempt to stay close to nearby flockmates;

ii) Collision Avoidance: avoid collisions with nearby flockmates;

iii) Velocity Matching: attempt to match velocity with nearby flockmates.

As stated in [13], the protocol can be expressed as:

$$\dot{r}_i = v_i \tag{6a}$$

$$\dot{v}_i = u_i \tag{6b}$$

r_i and v_i are position and speed of agent i. Agent i is steered via its acceleration input u_i which consists of two components, $u_i = \alpha_i + a_i$, $i = 1, . . . ,N$ Component α_i aims at aligning the velocity vectors of all the agents, which is similar in consensus protocols. Component a_i is a vector in the direction of the negated gradient of an artificial potential function, V_i, and is used for collision avoidance and cohesion in the group.

Definition 1 (Potential function) Potential V_{ij} is a differentiable, nonnegative, function of the distance $\|r_{ij}\|$ between agents i and j, such that

1) $V_{ij}(\|r_{ij}\|) \rightarrow +\infty$ as $\|r_{ij}\| \rightarrow 0$,

2) V_{ij} attains its unique minimum when agents i and j are located at a desired distance, and

3) $\frac{d}{d\|r_{ij}\|}V_{ij} = 0$, if $\|r_{ij}\| > R$

Theorem 5 ([13]) Consider a system of N mobile agents with dynamics (8), each steered by control law with potential function. Let both the position and velocity graphs be time-varying, but always connected. Then all pairwise velocity differences converge asymptotically to zero, collisions between the agents are avoided, and the system approaches a local extremum of the sum of all agent potentials.

While [13] solves the rule of alignment and collision avoidance, it does not deal with fragment of agents and obstacle avoidance. In [14] these requirements are taken into account by adding a virtual leader and setting potential functions of obstacles. A virtual leader or moving rendezvous point represents a group objective. If the virtual leader moves along a straight line with a desired velocity, based on modified expression of $u_i = \alpha_i + a_i + f_i(q_i,p_i,q_r,p_r)$ A secondary objective of an agent is to track the virtual leader.

Despite the similarities between certain terms in these protocols, the collective behavior would change drastically and never lead to fragmentation.

IV. RENDEZVOUS ALGORITHMS

In rendezvous algorithms, multiple agents in the network also have the sensing, computation, communication, and motion control capabilities. Synchronous rendezvous can be performed as "stop-and-go" maneuvers [10]. A stop-and-go maneuver takes place within a time interval consisting of two consecutive sub-intervals. When "stopped", i.e. in sensing period, agents are stationary and calculating where to go. Then agents try to move from their current positions to their next "way-points" and again come to rest. Here we will not consider collision avoidance.

Proper way-points or target points for each agent are of interest. In fact, target points are not unique. Agent i's kth way-point is the point to which agent i is to move to at time t_k. Thus if $x_i(t)$ and $x_i(\bar{t})$ denotes the position of agent i after and before moving, then agent i's kth moving protocol is:

$$x_i(t_k) = x_i(t_{k-1}) + u_{m_i(\bar{t}_k)}(x_{i_1}(\bar{t}_k) - x_i(\bar{t}_k), x_{i_2}(\bar{t}_k) - x_i(\bar{t}_k), \ldots, x_{i_{m_i(\bar{t}_k)}}(\bar{t}_k) - x_i(\bar{t}_k)) \quad (7)$$

And the positions of agent i's relative neighbors are $z_j \triangleq x_{i_j}(\bar{t}_k) - x_i(\bar{t}_k),\ j \in \{1, 2, \ldots, m_i\}$.

u_m is defined as

$$u_m(z_1, z_2, \ldots, z_m) \in D_M \cap C(z_1, z_2, \ldots, z_m) \cap \langle 0, z_1, z_2, \ldots, z_m \rangle, \quad (8)$$

$\forall \{z_1, z_2, \ldots, z_m\} \in D^m$

and

$$u_m(z_1, z_2, ..., z_m) \neq \text{a corner of } \langle 0, z_1, z_2, ..., z_m \rangle \tag{9}$$

Where $C(z)$ denote the closed disk of diameter r centered at the point $z/2$, and $C(z_1,\ldots z_m)$ $= \cap C(z_j)$, $j=1,2,\ldots,m$. Then we have the protocol of rendezvous:

Theorem 4 ([10]) Let $u_0 = 0$ and for each $m \in \{1,2,\ldots,n-1\}$, let u_m be any continuous function satisfying (6) and (7). For each set of initial agent positions $x_1(0)$; $x_2(0)$;…; $x_n(0)$, each agent's position $x_i(t)$ converges to a unique point $pi \in R^2$ such that for each $i, j \in \{1,2,\ldots,n\}$, either $p_i = p_j$ or $\|p_i - p_j\| > r$. Moreover, if agents i and j are registered neighbors at any time t, then $p_i = p_j$.

A most common target point would be the centroid of all relative neighbors, which would be simulated later. Another target point would be the circumcenter [11]. Each agent performs the following tasks:

i) it detects its neighbors according to G;

ii) it computes the circumcenter of the point set comprised of its neighbors and of itself;

iii) it moves toward this circumcenter while maintaining connectivity with its neighbors, which is proved in [11] and will be demonstrated in the simulation.

Below, in section VI, we have simulations of all common target points mentioned above. We have also enhanced the performance of rendezvous by adding a relay protocol. It can be shown that information between agents will flow more quickly with relay, which is similar to enlarging the area of sensing for every agent.

V. DUBINS VEHICLES

In [22], the dynamic model of the vehicle can be described as

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{10}$$

where $M = (x, y)$ are the coordinates of the reference point of the vehicle with respect to a reference frame, θ is the heading angle with respect to the frame x-axis. The configuration of the vehicle is $X = (M, \theta) \in \mathbb{R}^2 \times S^1$

v and ω are the linear and angular velocities of the vehicle. Without loss of generality, up to a time axis rescaling, we assume that $v(t) = V$. V is a constant. The turning radius of the vehicle is lower bounded by a constant value R > 0, which results in an upper bound on the vehicle's angular velocity ω.

For this model we consider the problem of determining a path of minimal length for reaching tangentially a rectilinear route with a specified direction of motion. We denote by *T* a target rectilinear path in the plane, with a prescribed direction of motion determined by the angle α with respect to the x-axis (see Figure 2). We consider the optimal control problem:

$$\text{Minimize } J = \int_0^T \sqrt{\dot{x}^2 + \dot{y}^2} dt = VT, \tag{11}$$

It is subject to (1) and (2), with $X(0)= (M_0, \theta_0)$ and such that, at the unspecified terminal time T, $M(\mathrm{T}) \in T$ and $\theta(\mathrm{T}) = \alpha$. By applying Pontryagin's Maximum Principle, the optimization problem is solvable.

After putting constraints on turning rate of agents, the characteristics of Dubins Vehicle will affect all distributed control algorithms discussed above, especially rendezvous. Intuitively, kinematic constraints such as bounded turning rate will restrict the mobility of agents and decrease the performance of distributed control systems, which is approved by simulations.

VI. SIMULATION

The simulation is implemented in Java with the Jprowler tool from Vanderbilt University [15]. Jprowler is a probabilistic simulator for prototyping and analyzing communication protocols of ad-hoc wireless networks. The simulator supports pluggable radio models and multiple application modules.

The simulation has added agents the ability to move in two dimensions, under steady speed varying their headings. The mobility of the agents creates a dynamic topology of nodes connectivity graph. The behavior of agents could be expressed as a sequence of discrete events. 3-D distribution and movement is available, but Java is not effective in displaying 3-D objectives.

When characterizing agent movements, two events are crucial in describing the behavior of agents, the *move* event and *update* event. The *move* event calculates new positions of each agent by acquiring the original position, the velocity and the heading. The *update* event collects information from neighbors, and then applies different protocols to figure out new velocity and headings, preparing for the next move event.

A. Consensus

After setting the adjacent matrices with weights when initiating the connection status of all moving agents, the consensus of 6 agents under fixed digraph is shown in Fig. 1. With strongly connected balanced digraph or spanning tree, consensus is achieved under nearest neighbor rules. It is plotted that the agent in the dark side of the line is identified as a neighbor of the agent in the light side.

Randomly weighted consensus under nearest neighborhood rule is demonstrated in Fig. 2, where fragments are almost inevitable since the detect radius is less than 2 grids. The darkness of the lines between agents implies the weight of edges.

In Fig. 3, Small-world algorithm is also simulated. Even under a relative low rewiring rate (<0.01) and with strong noise (>20%), consensus can be achieved much faster than in nearest neighborhood rules, no matter the graph is weighted or not.

Simulations with delayed communication have similar performance as shown in Fig. 2, but the consensus value varies when we have random delays for each agents. A straightforward explanation of consensus variations can be deduced from [26], where the asynchronous system has an enlarged graph Laplacian. Although the original graph Laplacian is consistent, different delays of communication links will cause the enlarged matrix to vary. Consensus is still feasible if the delay is bounded, but the consensus value changes every time when we run the simulation.

One major reason of link dropping is that the consensus protocol does not ensure agents always move closer to each other. Consensus is a distributed feedback law which compensates in some extent the increased distance between agents. But when one agent is at the edge of detection range of another agent, very likely they will lose their communication links. Our adaptive weight optimization adds weights to longer links proportionally. This time varying optimization will change consensus value (if achieved) while maintaining network connectivity.

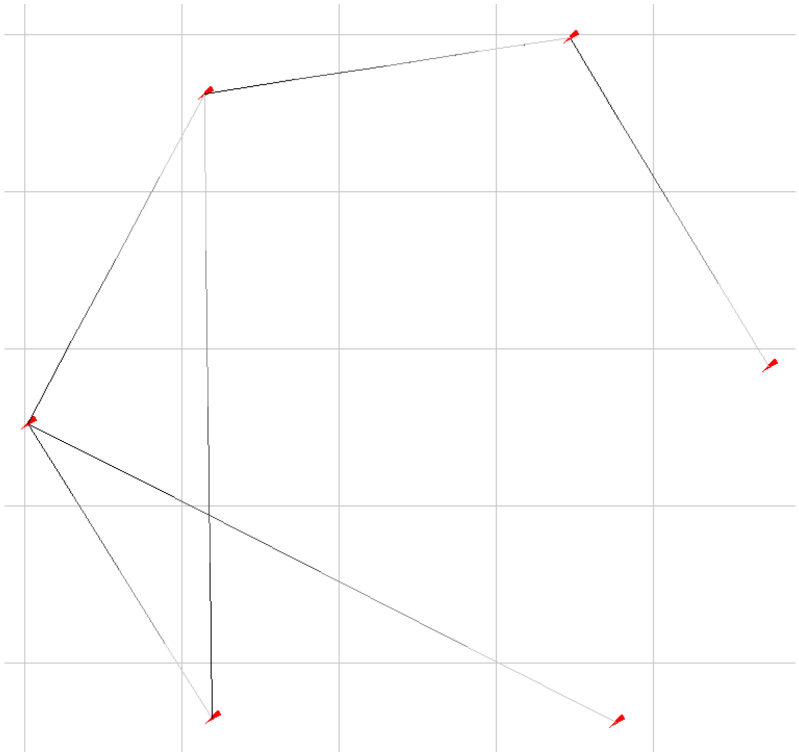

Fig. 1. Consensus under fixed topology (6 agents)

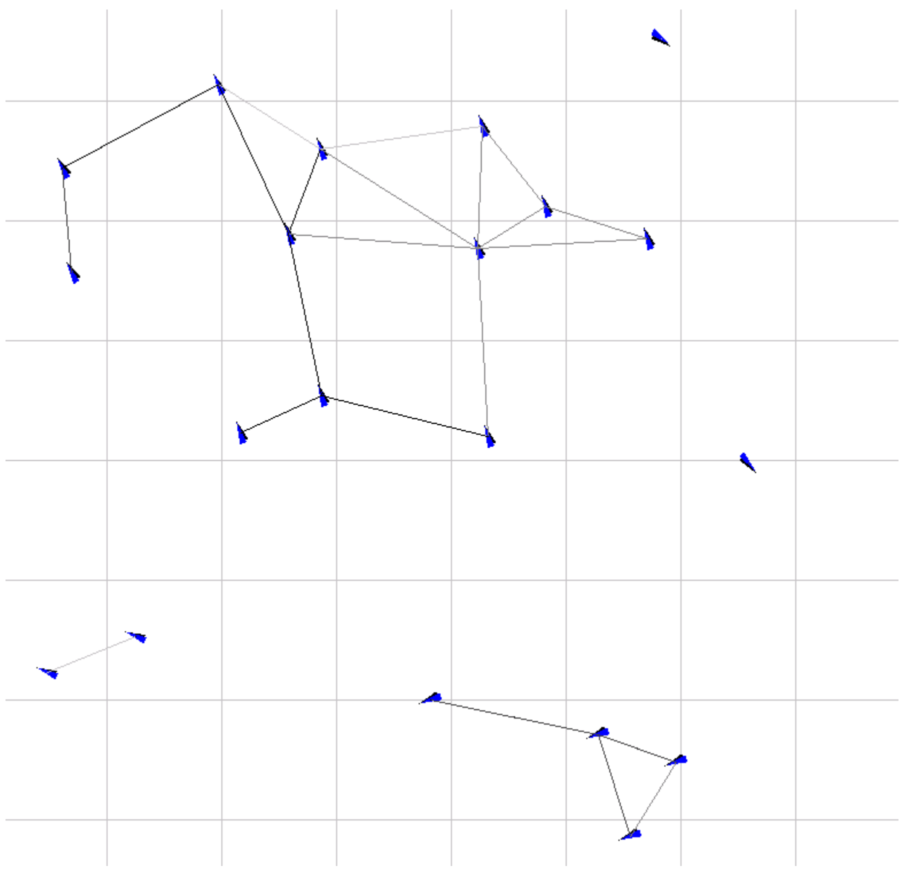

Fig. 2. Weighted consensus under nearest neighborhood rule
(20 agents, fragment)

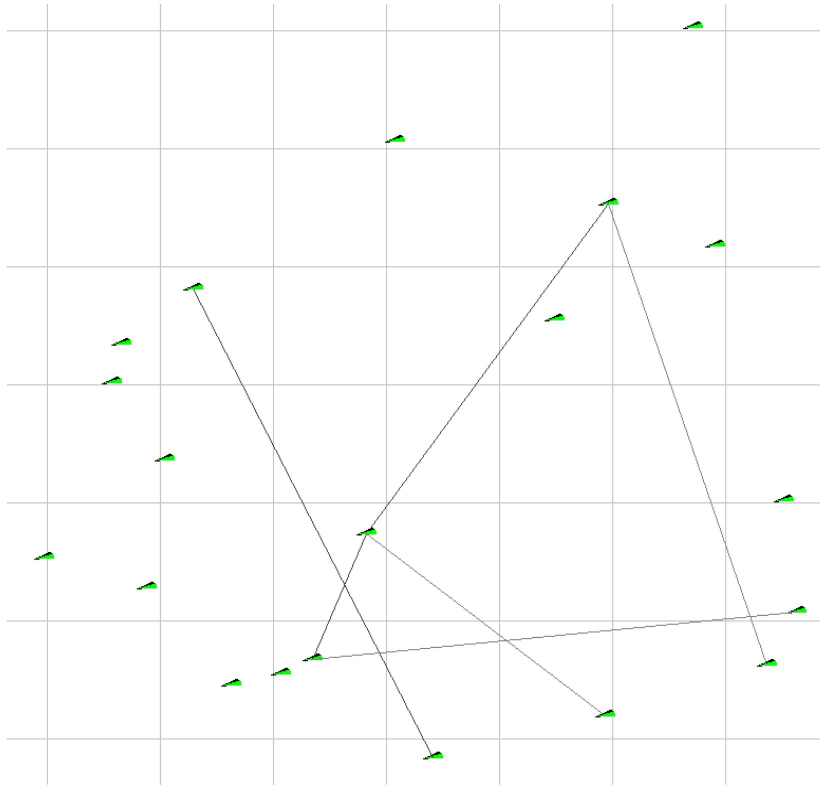

Fig. 3. Consensus with small world algorithm (20 agents)

B. Rendezvous

Trajectories of all mobile agents are plotted to illustrate the process of rendezvous. The discrete synchronous rendezvous problem has been simulated, but the implementation is slightly different from the "stop and go" maneuver mentioned in [8]. Agents will not wait until all the agents are ready for the next move. For certain initial conditions, this protocol will cause connectivity loss. The result in Fig. 4 shows that rendezvous is straight-forward if the position graph is connected, but link failure caused by limited sensing radius of agents may render decomposition.

Fig. 5 shows by applying the circumcenter algorithm in [11], the movement of agents is no longer smooth, but connectivity is well maintained. Computational complexity of the circumcenter algorithm is definitely larger than rendezvous to neighborhood centroid, thus circumcenter algorithm is more computationally costly after scaling.

By using the relay protocol, Fig. 6 is similar to the figure of rendezvous to centroid when sensing radius is doubled. It shows performance of rendezvous can be enhanced by adding communication cost.

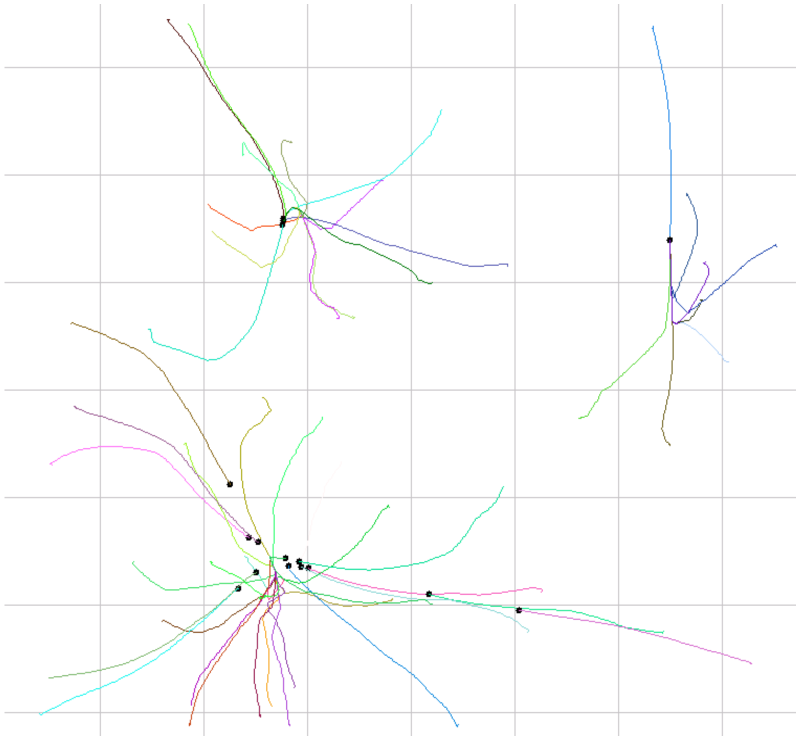

Fig. 4. Rendezvous to the centroid of neighbors (50 agents)

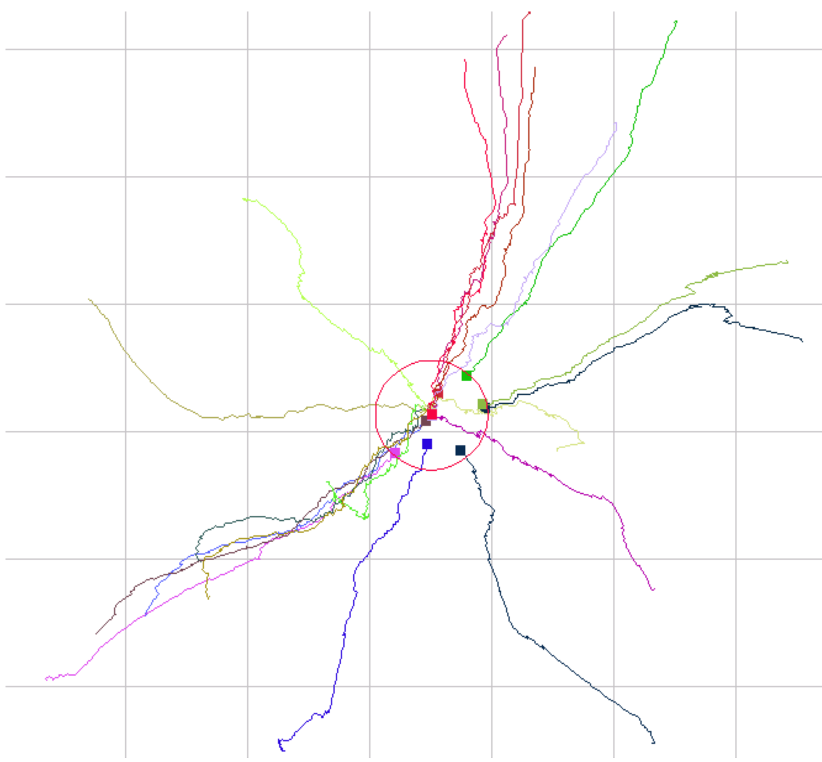

Fig. 5. Rendezvous to the circumcenter (20 agents)

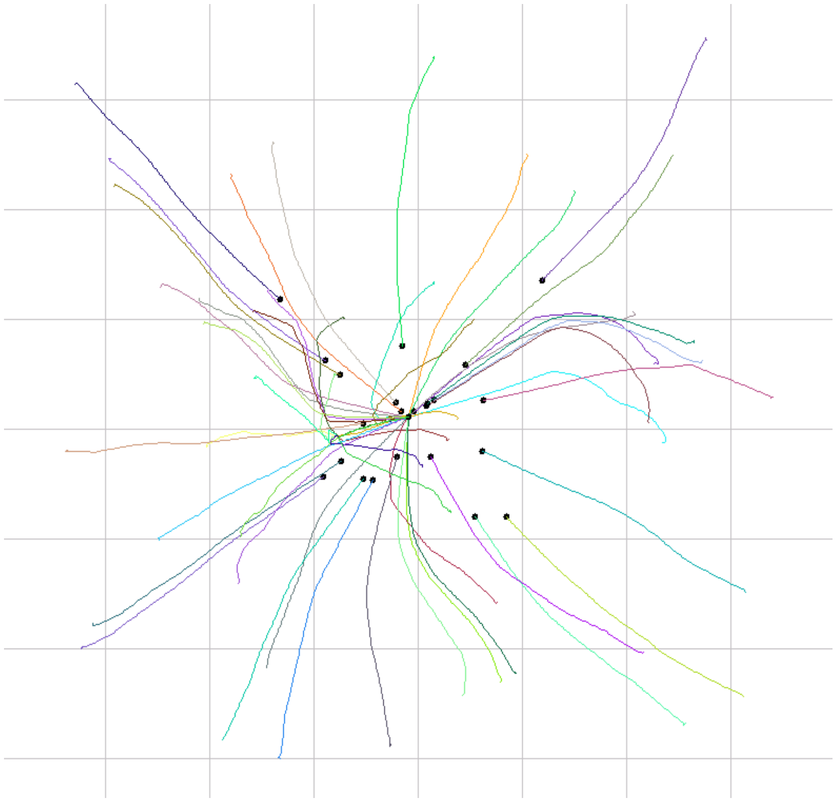

Fig. 6. Rendezvous to the centroid with relay protocol (50 agents)

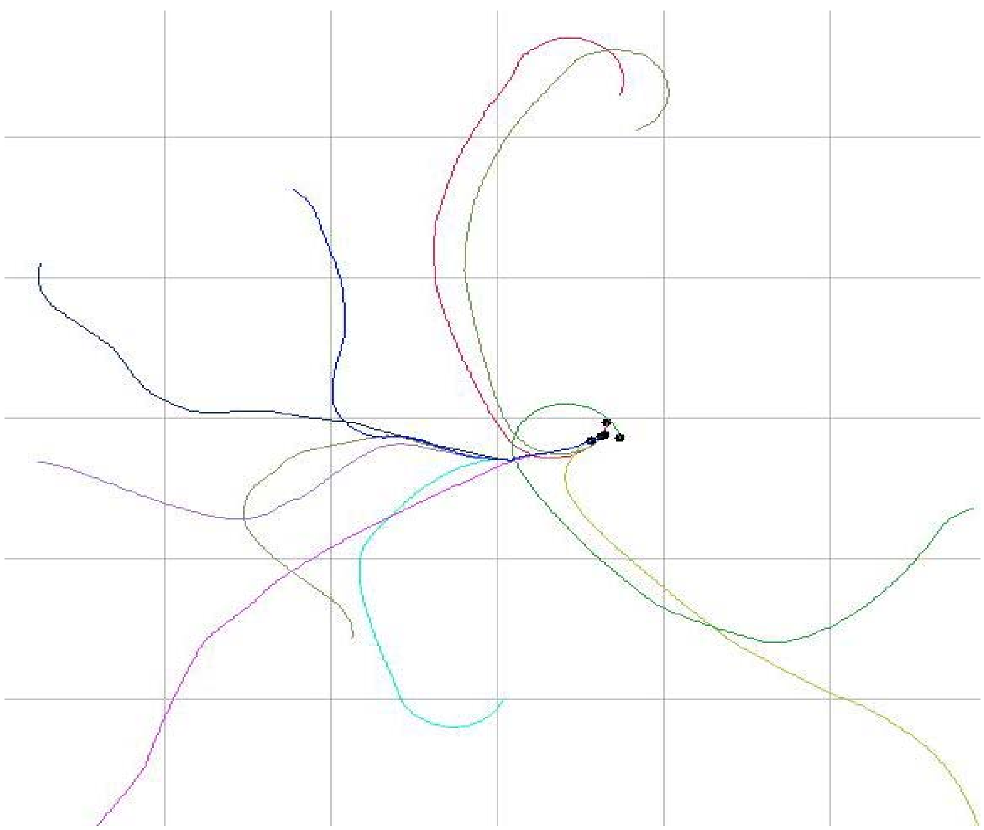

Figure 7. Rendezvous of Dubins' vehicles (10 agents)

Fig.7 is the simulations of rendezvous with the constraints of Dubins vehicle. When turning rate is bounded, no sudden turning is allowed. Thus rendezvous is more restricted and the trajectories of agents are smoother in the figure.

C. Flocking

In flocking algorithms, a potential function is applied to avoid collision. In simulation to the algorithm in [13], taking into account collision avoidance only, fragment is almost inevitable in Fig. 8.

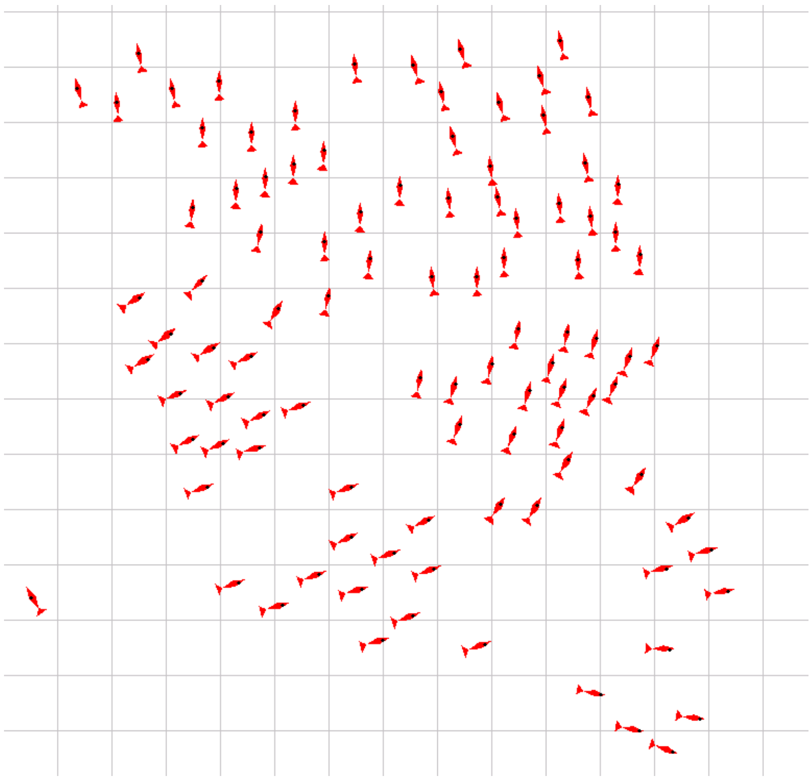

Fig. 8. Flocking with collision avoidance (100 agents)

Fig. 9 shows algorithm 2 in [14]. A virtual leader is moving around the gray obstacle. All agents are attempting to follow it, thus gathering into the same direction. The potential field and the radius of sensing needs to be carefully adjusted.

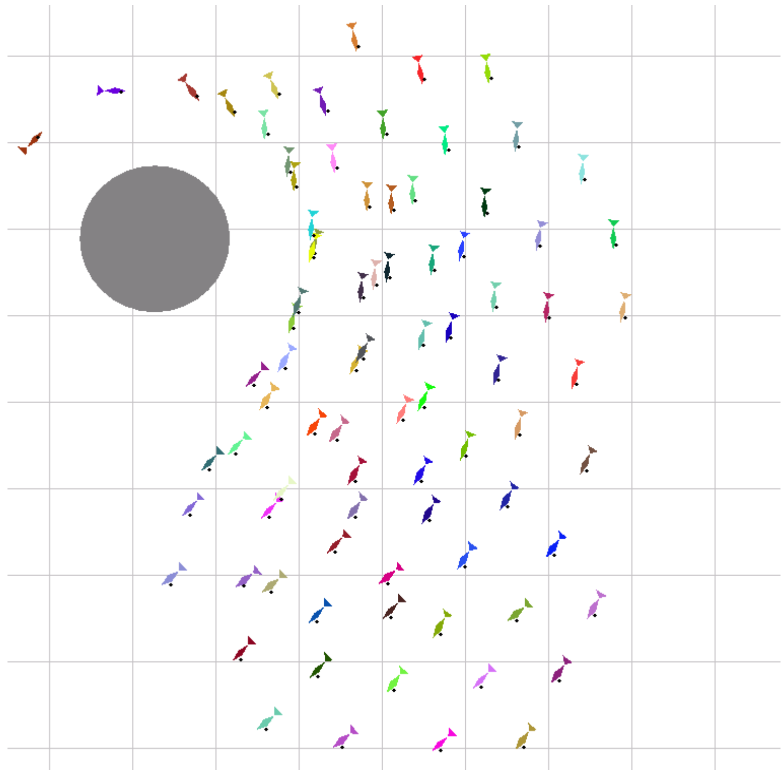

Fig. 9. Flocking with virtual leader (100 agents)

VII. Conclusion

In this paper, a number of recent results in distributed control algorithms were summarized. We demonstrated some examples of multi-agent dynamic systems. Simulations of consensus, rendezvous and flocking are instrumental in studying various protocols under different information constraints. Dubins vehicle is a more realistic model to simulate agent kinematics.

Future work will be focused on further understanding asynchronous protocols. For example, with more constraints such as communication delay and limited agent mobility, whether there are any necessary or sufficient conditions to make these protocols still feasible.

Simulation software implementing all above algorithms is available from the first author (pwu1@nd.edu).

APPENDIX

	Algorithms	Authors
Consensus	Consensus achieved in DT / CT (dwell time needed) under nearest neighbor rules, undirected and jointly connected graphs in every period required	Jadbabaie, A. Jie Lin and Morse, A.S. [2], 2003
	Average consensus achieved for fixed and variant topology: strongly connected balanced digraph, also a sufficient condition of time-delays for convergence	Olfati-Saber, R.; Murray, R.M. [3], 2004
	Consensus achieved in DT / CT (dwell time needed) under variant graph: jointly spanning tree in every period	Wei Ren; Beard, R.W. [4], 2005
	Asynchronous consensus achieved, but direction unspecified	Lei Fang; Panos J. Antsaklis, Anastasis Tzimas [19], 2005
	Small world algorithm (random rewiring networks) The algebraic connectivity is more than 1000 times greater than regular networks.	Olfati-Saber, R [5], 2005
Rendezvous	Rendezvous achieved under a synchronous “stop and go” maneuver if the initial graph is connected. Specified target points needed, such as the centroid of neighbors, or the center of the smallest circle containing convex hull	J. Lin, A. S. Morse [8], 2003
	Asynchronous rendezvous achieved if the directed graph characterizing registered neighbors is strongly connected.	J. Lin, A. S. Morse [8], 2003
	Circumcenter algorithm in arbitrary dimension, robust to link failures given strongly connected graph	J. Corte´s, S. Martı´nez, and F. Bullo [9], 2004
	Rendezvous of Dubins Vehicles	Amit Bhatia, Emilio Frazzoli [23], 2008
Flocking/ swarm	Flocking achieved under time-varying but always connected position and velocity graphs, collision avoidance ensured, no dwell time needed	H Tanner, A Jadbabaie, GJ Pappas [6], 2005
	Flocking satisfying Reynolds’ three heuristic rules achieved, taking account obstacle avoidance. Group objective is necessary by setting moving rendezvous points to prevent fragments.	Olfati-Saber, R. [7], 2006

Table 2. A Categorization of some recent distributed control algorithms

REFERENCES

[1] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Trans Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[2] C.W. Reynolds, “Flocks, herds, and schools: a distributed behavioral model,” in *Computer Graphics (ACM SIGGRAPH ’87 Conf. Proc.)*, vol. 21, July 1987, pp. 25-34

[3] A. Jadbabaie, L. Jie, A.S. Morse, “Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules”, *Automatic Control, IEEE Transactions* on Vol 48, No 6, June 2003 Page(s):988 - 1001

[4] W. Ren, R.W. Beard, “Consensus seeking in multi agent systems under dynamically changing interaction topologies”, *Automatic Control, IEEE Transactions* on Volume 50, No 5, May 2005 Page(s):655 - 661

[5] R. Olfati-Saber, and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays”, *Automatic Control, IEEE Transactions* on Vol 49, No 9, Sept. 2004 Page(s):1520 – 1533

[6] R. Olfati-Saber, “Ultrafast consensus in small-world networks”, *American Control Conference*, 2005. Proceedings of the 2005

[7] R. Olfati-Saber, J.A. Fax, and R.M. Murray, “Consensus and Cooperation in Networked Multi-Agent Systems”, *Proceedings of the IEEE*, Volume 95, No 1, Jan. 2007 Page(s):215 – 233

[8] L. Fang; P.J. Antsaklis, "Information consensus of asynchronous discrete-time multi-agent systems," *American Control Conference*, 2005. Proceedings of the 2005 , vol., no., pp. 1883-1888 vol. 3, 8-10 June 2005

[9] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, “Distributed memoryless point convergence algorithm for mobile robots with limited visibility,” *IEEE Trans. Robot. Automat*. vol. 15, no. 5, pp. 818–828, Oct. 1999.

[10] J. Lin, A. S. Morse, and B. D.O. Anderson, “The Multi-agent Rendezvous Problem”, in *Proc. 42nd IEEE Conf. Decision and Control*, Dec. 2003

[11] J. Corte´s, S. Martı´nez, and F. Bullo, “Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions”, *IEEE Trans. Automatic Control*, vol. 51, no. 8, pp. 1289–1298, Aug. 2004.

[12]Z. Lin, M. Broucke, and B. Francis, “Local control strategies for groups of mobile autonomous agents,” *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 622–629, Apr. 2004.

[13]H.Tanner, A.Jadbabaie, G.J.Pappas, “Flocking in fixed and switching networks”, - *IEEE Conference on Decision and Control*, 2005

[14]R. Olfati-Saber, “Flocking for Multi-Agent Dynamic Systems Algorithms and Theory”, *Automatic Control, IEEE Transactions* on vol.51, no.3, March 2006 pp.401 – 420

[15]http://w3.isis.vanderbilt.edu/projects/nest/JProwler/index.html

[16]R.Olfati-Saber, J.A.Fax, R.M.Murray, "Consensus and Cooperation in Networked Multi-Agent Systems", *Proc. of the IEEE*, vol.95, no.1, pp.215-233, Jan. 2007

[17]L. Fang, P. J. Antsaklis, "Asynchronous Consensus Protocols Using Nonlinear Paracontractions Theory," IEEE Trans on Auto. Control, 2008

[18]L. Fang, P. J. Antsaklis, "On Communication Requirements for Multi-agent Consensus Seeking," Networked Embedded Sensing and Control, *Proceedings of Workshop NESC*’05

[19]L. Fang, P. J. Antsaklis, A. Tzimas, "Asynchronous Consensus Protocols: Preliminary Results, Simulations and Open Questions," *Proc. IEEE CDC*, 2005

[20]P. Yang, R. A. Freeman, K. M. Lynch “Optimal Information Propagation in Sensor Networks” *IEEE International Conference on Robotics and Automation*, 2006

[21]L. E. Dubins. “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents.” *American Journal of Mathematics*, 79(3):497 – 516, 1957.

[22]P. Soueres, A. Balluchi, A. Bicchi, “Optimal feedback control for route tracking with a bouded-curvature vehicle”, *Proc. of the IEEE Int. Conference on Robotics and Automation*, San-Fransisco, California, USA, May 2000.

[23]Amit Bhatia, Emilio Frazzoli. “Decentralized algorithm for minimum-time rendezvous of Dubins vehicles.” *2008 American Control Conference*. Seattle, USA June 2008, Page(s):1343 – 1349

[24]P. Bliman, G. Ferrari-Trecate, “Average consensus problems in networks of agents with delayed communications”, Technical Report, arXiv:math/0503009v2, 2005

[25]J.A.Yorke, “Asymptotic stability for one dimensional differential-delay equations” *Journal of Differential equations*, 7:189–202, 1970.

[26]A. Nedich and A. Ozdaglar, "Convergence Rate for Consensus with Delays," LIDS Technical Report 2774, MIT, Lab. for Information and Decision Systems, 2007

[27]vol. 95, No. 1, *Proceedings of the IEEE*

[28]Bertsekas, D. P.; Tsitsiklis, J. N.; “Comments on ‘Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules’”, *Automatic Control, IEEE Transactions* on Volume 52, Issue 5, May 2007 Page(s):968 - 969