Proceedings of the 58th Israel Annual Conference
on Aerospace Sciences, Tel-Aviv & Haifa, Israel,
March 14-15, 2018

The Aggregating Consensus Protocol - a Case Study of Behavioral Multi-Agent Systems

Noam Leiter* and Daniel Zelazo†
Technion—Israel Institute of Technology, Haifa 32000, Israel

We suggest a behavioral multi-agent system where each behavioral primitive is identified with an operation on the underlying graph structure of the networked system. As an example of a behavioral multi-agent system, the aggregating consensus protocol is defined as a consensus model with the graph contraction operation as a behavioral primitive. Several case study simulations are provided to demonstrate the dynamic properties of this new model.

I. Introduction

In the seminal works by Brooks [1,2] on artificial intelligence, the notion of *behavioral primitives* were established. These primitives, Brooks argued, define a set of low-level basic tasks that any intelligent robot should be able to perform. For multi-agent systems, it is natural to ask if there are similar behavioral primitives that define a collective intelligence of the system. In fact, it may not be sufficient that the individual robots (or agents) in a multi-agent system behave intelligently. Rather, the intelligence of a multi-agent system should be based specifically on the tasks that can be performed by inter-agent interactions and coordination.

Multi-agent operations such as gathering, avoidance, and docking were suggested as behavioral primitives for obtaining intelligent satellite clusters [3–5]. However, in those studies the graph-basis of the system is not exploited. Graph grammars have been used for self-organizing of robots [6].

Here we suggest a behavioral multi-agent system where each behavioral primitive is identified with an operation on the underlying graph structure of the networked system. We can then fully exploit graph theoretical tools and algorithms for the design and analysis of such systems.

In a networked dynamic system a graph operation results in a change of the graph structure, and a corresponding change of the dynamical system, e.g., networked dynamic systems with switching connectivity [7]. In a more general behavioral multi-agent system, the translation of behavioral primitives to graph operations may be interpreted as a hybrid dynamical system [8] or an event-triggered multi-agent system [9].

The consensus protocol has been a benchmark problem in the study of multi-agent systems. In a consensus model, agents asymptotically converge to an agreement of a single value [7]. In practice, agents have a finite size and will eventually collide during

*Graduate Student, Faculty of Aerospace Engineering, noaml@technion.ac.il.
†Assistant Professor, Faculty of Aerospace Engineering, dzelazo@technion.ac.il.

the process. For some applications, as smart structures assembly, these collisions end in a physical link between agents which are required to form larger structures from agent level building blocks. In a previous study [10], we have presented model reduction of the consensus protocol based on graph contractions. In this study we use graph contractions as a single behavioral primitive representing the merging of nodes and the resulting reduction of the underlying graph structure and a dimension reduction of the consensus model.

The remaining sections of this paper are as follows. In Section II, we provide the mathematical description of the graph contraction operation which is the basis of the aggregating consensus protocol. In Section III, we formulate the aggregating consensus as a case study for a behavioral multi-agent system. In Section IV, two simulation case studies of the aggregating consensus are presented. Section V provides concluding remarks.

PRELIMINARIES A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ consists of a vertex set $\mathcal{V}(\mathcal{G})$, an edge set $\mathcal{E}(\mathcal{G}) = \{\epsilon_1, \ldots, \epsilon_{|\mathcal{E}|}\}$ with $\epsilon_k \in \mathcal{V}^2$, and a set of positive edge weights, $\mathcal{W}(\mathcal{G}) = \{w_i\}_{i=1}^{|\mathcal{E}|}$. The order of the graph is defined as the number of nodes, $n = |\mathcal{V}|$. Two nodes $u, v \in \mathcal{V}(\mathcal{G})$ are *adjacent* if $\{u, v\} \in \mathcal{E}(\mathcal{G})$. A *path* in $\mathcal{G}$ is a sequence of distinct vertices such that consecutive vertices are adjacent. If all distinct pairs of vertices of $\mathcal{G}$ are connected by a path in $\mathcal{G}$ then $\mathcal{G}$ is *connected*.

We assign an orientation to the edges using head and tail functions, $h_{\mathcal{E}}, t_{\mathcal{E}} : \mathcal{E} \to \mathcal{V}$ where $h_{\mathcal{E}}(\epsilon_k)$ and $t_{\mathcal{E}}(\epsilon_k)$ return, respectively, the *head* and *tail* nodes of edge e_k. If $\mathcal{G}$ is an *undirected graph* then the head and tail of each edge are arbitrary; if $\mathcal{G}$ is a *directed graph* (digraph) then the head and tail define the direction of the edge. A *self-loop* is an edge $e_k \in \mathcal{E}$ such that $h_{\mathcal{E}}(\epsilon_k) = t_{\mathcal{E}}(\epsilon_k)$. A *simple graph* does not include self-loops. The head and tail functions can be used to define the *incidence function* $f_E : \mathcal{V}(\mathcal{G}) \times \mathcal{E}(\mathcal{G}) \to \{\pm 1, 0\}$, with $f_E(v_i, \epsilon_j) = 1$ if $h_{\mathcal{E}}(\epsilon_j) = v_i$, $f_E(v_i, \epsilon_j) = -1$ if $t_{\mathcal{E}}(\epsilon_j) = v_i$, and 0 otherwise. The incidence function can be used to define the corresponding *incidence matrix*, $E(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, with entries $[E(\mathcal{G})]_{ij} = f_E(v_i, \epsilon_j)$. For a simple undirected graph, the Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is defined as $L(\mathcal{G}) = E(\mathcal{G}) W(\mathcal{G}) E(\mathcal{G})^T$ [11]. The Laplacian matrix eigenvalues are non-negative and can be ordered as $0 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The second smallest eigenvalue of $L(\mathcal{G})$, $\lambda_2(\mathcal{G})$ is also known as the Fiedler eigenvalue or the algebraic connectivity of $L(\mathcal{G})$.

II. Graph Contractions

Graphs and their algebraic representations are the main modeling tools for networked systems. In order to model behavioral multi-agent systems we must translate network behavioral primitives to operations on the underlying graph structure. An important example of a behavioral primitive in a networked system is the physical connection between a subset of agents due to docking or collision. The corresponding graph operation is then the merging of the subsets of agent-representing nodes to a single node. If as a result of such physical connections in the system the communication network is preserved, then each merged node will maintain all edges connecting adjacent nodes prior to the merge. The graph operation modeling such changes in the network is a graph contraction. Graph contractions have been widely used in graph theory, e.g. for graph clustering [12] and in the study of network communities [13]. The following section describes the mathematical formulation of graph contractions.

A graph contraction is a reduction of a graph based on a partition of the vertex set. We define an *r-partition* π of a vertex set $\mathcal{V}$ as a collection of r subsets or cells $C_i \subseteq \mathcal{V}$ such that $C_i \cap C_j = \emptyset$ and $\cup_{i=1}^{r} C_i = \mathcal{V}$. We denote the set of all r-partitions of $\mathcal{V}$ by $\mathbb{P}_r(\mathcal{V})$. The graph contraction of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over a given r-partition $\pi \in \mathbb{P}_r(\mathcal{V}(\mathcal{G}))$ is an operation performed by merging all nodes inside each partition cell while maintaining the connectivity between adjacent nodes in different cells. Thus, each cell C_i is contracted to a single node, and cells C_i and C_j are connected by an edge in the reduced graph if there exists a node $u \in C_i$ and a node $v \in C_j$ such that $\{u, v\} \in \mathcal{E}(\mathcal{G})$. In the event of multiple edges connecting two cells, redundant edges are removed from the contracted graph. The contracted graph of $\mathcal{G}$ over π is denoted as $\mathcal{G} /\!\!/ \pi$. For example consider the graph contraction in Figure 1 over the partition $\pi = \{\{v_1\}, \{v_2, v_3\}, \{v_4, v_5\}\} \in \mathbb{P}_3(\mathcal{V}(\mathcal{G}))$.

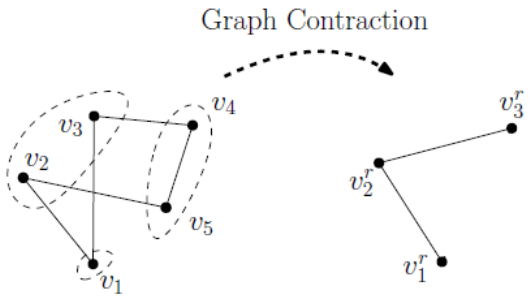


Figure 1: Graph contraction illustration

Theorem 1 *If $\mathcal{G}$ is connected then $\mathcal{G} /\!\!/ \pi$ is connected.*

Proof 1 *This can be shown based on the observation that any path in $\mathcal{G}$ is transformed to a walk in $\mathcal{G} /\!\!/ \pi$, where a walk is a sequence of nodes without restricting to distinct nodes [10].*

Definition 1 *We define the* partition characteristic matrix $P_\pi \in \mathbb{R}^{|\mathcal{V}| \times r}$ *for an r-partition $\pi \in \mathbb{P}_r(\mathcal{V}(\mathcal{G}))$ as $[P_\pi]_{ij} = 1$ if $v_i \in C_j$, and 0 otherwise.*

An important observation is that the graph contraction operation can be equivalently expressed as a linear operation on the incidence matrix of the graph. In our previous work [10], we have shown that the incidence matrix of the reduced graph $\mathcal{G}_r = \mathcal{G} /\!\!/ \pi$ can be expressed by the transformation

$$E(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_\pi, \tag{1}$$

where U_π is some appropriately chosen matrix. Recalling that $L(\mathcal{G}) = E(\mathcal{G}) W(\mathcal{G}) E(\mathcal{G})^T$ we obtain the corresponding contracted graph Laplacian matrix,

$$L(\mathcal{G}_r) = P_\pi^T E(\mathcal{G}) U_\pi W(\mathcal{G}_r) U_\pi^T E(\mathcal{G}) P_\pi, \tag{2}$$

with $W(\mathcal{G}_r)$ a chosen set of contracted graph edge weights.

III. The Aggregating Consensus Protocol

The consensus protocol can be viewed as a benchmark problem in the study of multi-agent systems, and therefore it is a natural case study of a behavioral multi-agent system. In a physical system running the consensus protocol, agents may have a finite physical dimension and once they are in close proximity they will collide. From another perspective, we can imagine agents that approach each other can *merge* together, forming a new 'super' agent that may continue running the consensus protocol with the remaining adjacent agents - the consensus protocol combined with this merging behavioral primitive is what we term an *aggregating consensus protocol*. This aggregating process can continue until all nodes have contracted. We first describe the linear consensus protocol and then combine it with the graph contraction operation to model the aggregating process.

We consider the linear consensus protocol over $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, a simple connected undirected graph of order n, with each node representing an agent. Each node $v_i \in \mathcal{V}(\mathcal{G})$ is associated with a state variable $x_i \in \mathbb{R}$ which evolves according to a continuous time dynamical system

$$\dot{x}_i = -\sum_{j \sim i} (x_i - x_j), \tag{3}$$

where $j \sim i$ denotes that node v_j is adjacent to node v_i. Taking $x = \mathrm{col}(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ as the multi-agent system state, the overall state-space representation of the linear consensus protocol is

$$\dot{x} = -L(\mathcal{G})\, x, \tag{4}$$

where $L(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is the Laplacian matrix [14]. For a multi-dimensional case $x_i \in \mathbb{R}^d$ and $x \in \mathbb{R}^{dn}$, and the consensus model is extended with the Kronecker product

$$\dot{x} = -\left[I_{d \times d} \otimes L(\mathcal{G})\right] x. \tag{5}$$

Some important properties of the linear consensus protocol (4) are as follows [14]:

Theorem 2 *A graph $\mathcal{G}$ is connected if and only if $\lambda_2(\mathcal{G}) > 0$.*

Theorem 3 *The linear agreement protocol converges to the agreement set $\mathcal{A}$ from any initial condition if and only if $\lambda_2(\mathcal{G}) > 0$, where the agreement set is the span of the all ones vector, $\mathcal{A} = span\{\mathbb{1}\}$. Furthermore, $\lambda_2(\mathcal{G})$ dictates the rate of convergence.*

Theorem 4 *If $\mathcal{G}$ is a connected graph of order n then all agents will asymptotically converge to a point which is the average of their initial condition,*

$$\lim_{t \to \infty} x(t) = \frac{1}{n} \mathbb{1}\mathbb{1}^T x(0) \in \mathcal{A}. \tag{6}$$

In the linear consensus model agents are represented by zero dimension points. In order to allow merging during the consensus protocol we define the aggregating threshold $r_c > 0$ such that two agents x_i and x_j will merge when $|x_i - x_j| \leq r_c$. We observe that when agents are associated to the nodes in a graph, this merge is equivalent to the graph contraction operation described in Section II. In order to fully describe the aggregating process with the mathematical abstraction of graph contractions, we define the contraction graph $\mathcal{G}_c(\mathcal{G}, x(t)) = (\mathcal{V}(\mathcal{G}), \mathcal{E}_c(x))$ where the edge set $\mathcal{E}_c$ captures all collisions,

$$\mathcal{E}_c(x(t)) = \left\{\{i,j\} \in \mathcal{V}^2 \,|\, \|x_i - x_j\| \leq r_c\right\}. \tag{7}$$

Definition 2 *The connected components partition of a graph $\mathcal{G}$ is the partition $\pi_{cc}(\mathcal{G}) = \{C_i\}_{i=1}^{n_{cc}}$ such that any distinct pair of nodes in the same cell of π_{cc} are connected by a path in $\mathcal{G}$, and any pair of nodes in distinct cells of π_{cc} are not connected by any path in $\mathcal{G}$.*

We construct the contraction partition $\pi_c(\mathcal{G}, x(t))$ as the connected components partition of $\mathcal{G}_c(\mathcal{G}, x(t))$

$$\pi_c(\mathcal{G}, x(t)) = \pi_{cc}(\mathcal{G}_c(\mathcal{G}, x(t))), \tag{8}$$

and the corresponding graph contraction as

$$\mathcal{G}_r = \mathcal{G} /\!\!/ \pi_c(\mathcal{G}, x(t)). \tag{9}$$

The contraction graph $\mathcal{G}_c$ is a dynamic graph and changes as a function of the state $x(t)$. As long as all agents are within a distance greater than r_c then $\mathcal{E}_c$ is an empty set and π_c is a trivial partition consisting only of singlet cells such that $\mathcal{G} /\!\!/ \pi_c(\mathcal{G}, x(t)) = \mathcal{G}$. With no change in the graph, the agents follow the linear consensus. If a subset of agents are within pairwise distances less than r_c then π_c will contain a non-singlet cell. At that point a graph contraction event occurs and the graph changes according to the graph contraction operation,

$$\mathcal{G}_{(k)} = \mathcal{G}_{(k-1)} /\!\!/ \pi_{(k)}, \tag{10}$$

where we define $\pi_{(k)} \triangleq \pi_c\left(\mathcal{G}_{(k-1)}, x(t_k)\right)$ and $\mathcal{G}_{(k-1)}$ is the constant graph between event times t_{k-1} and t_k. As described in Section II, we can perform an Algebraic transformation of the Laplacian matrix of the graph $\mathcal{G}_{(k)}$ to obtain the Laplacian of the contracted graph $\mathcal{G}_{(k+1)}$

$$L(\mathcal{G}_{(k+1)}) = P_{(k+1)}^T E\left(\mathcal{G}_{(k)}\right) U_\pi W\left(\mathcal{G}_{(k+1)}\right) U_\pi^T E\left(\mathcal{G}_{(k)}\right) P_{(k+1)}, \tag{11}$$

where $P_{(k)}$ is the partition characteristic matrix corresponding to the partition $\pi_{(k)}$. The contracted Laplacian is then a realization of a new consensus protocol over the contracted graph. Since the number of agents decreased we must reinitialize the state. There is some freedom in choosing the initial state of each node in the contracted graph $\mathcal{G}_{(k+1)}$. Here we choose the contracted state as the average of the states of the merged nodes in each cell $C_j \in \pi_{(k+1)}$

$$x_j^{(k+1)}(t_{k+1}) = \frac{1}{|C_j|} \sum_{i \in C_j} x_i^{(k)}(t_{k+1}). \tag{12}$$

This choice is consistent with Theorem 4 such that if all agents are within an initial distance less than r_c they will merge to a single position at the average of their initial state.

Proposition 1 *The reinitialized state after the k'th contraction of Eq.(12) takes the form,*

$$x^{(k)}(t_k) = D_{(k)}^{-1} P_{(k)}^T x^{(k-1)}(t_k), \tag{13}$$

and the extended model reinitialized state takes the form

$$x^{(k)}(t_k) = \left[I_{d\times d} \otimes \left(D_{(k)}^{-1} P_{(k)}^T\right)\right] x^{(k-1)}(t_k), \tag{14}$$

where $D_{(k)} \triangleq P_{(k)}^T P_{(k)}$ is a diagonal matrix with $\left[D_{(k)}\right]_{jj} = |C_j|$.

Proof 2 *For an r-partition $\pi \in \mathbb{P}_r(\mathcal{V}(\mathcal{G}))$ the partition characteristic matrix is defined as $[P_\pi]_{ij} = 1$ if $v_i \in C_j$, and 0 otherwise (Def. 1). Let d_i denote the number of nodes in the cell $C_i \in \pi$, $d_i = |C_i|$. Without loss of generality we can order the vertices such that*

$$P_\pi = \begin{bmatrix} \mathbb{1}_{d_1} & & & \\ & \mathbb{1}_{d_2} & & \\ & & \vdots & \\ & & & \mathbb{1}_{d_r} \end{bmatrix} \tag{15}$$

where $\mathbb{1}_{d_i}$ is the all ones vector in $\mathbb{R}^{d_i}$. We first obtain that

$$D_\pi = P_\pi^T P_\pi = Diag(d_i, d_2, \ldots, d_r).$$

We also observe that for $x \in \mathbb{R}^n$ we get $y = P_\pi^T x \in \mathbb{R}^r$ where

$$\begin{aligned} y_j &= \mathbb{1}_{d_j}^T \begin{bmatrix} 0 & x_{j_1} & x_{j_2} & \cdots & x_{j_{d_j}} & 0 \end{bmatrix}^T \\ &= \sum_{i=1}^{d_j} x_{j_i}, \end{aligned}$$

with $j_i = \sum_{k=1}^{j-1} d_k + i$. Since we ordered the nodes such that $v_{j_i} \in C_j$ we obtain $y_j = \sum_{i=1}^{d_j} x_{j_i} = \sum_{i \in C_j} x_i$, and normalizing by D_π^{-1} we get $\hat{y} = D_\pi^{-1} P_\pi^T x$ and

$$\hat{y}_j = \frac{1}{d_j} \sum_{i \in C_j} x_i = \frac{1}{|C_j|} \sum_{i \in C_j} x_i.$$

For $x \in \mathbb{R}^{dn}$ we can reshape the vector to a matrix $X = \left[x^{[1]}, x^{[2]}, \ldots, x^{[d]}\right] \in \mathbb{R}^{n \times d}$ where each column corresponds to a dimension. We then get $\hat{Y} = D_\pi^{-1} P_\pi^T X$, and column stacking Y and using the Kronecker product we get the required equation

$$\hat{y} = \left[I_{d \times d} \otimes \left(D_\pi^{-1} P_\pi^T\right)\right] x. \tag{16}$$

With the contraction event we can fully describe the aggregating consensus protocol as an event triggered system given in Algorithm 1.

Theorem 5 *For any connected graph $\mathcal{G}_{(0)}$ of any order $n_0 < \infty$ and any $x^{(0)}(t_0) \in \mathbb{R}^{dn_0}$, $\exists N < \infty$ and $t_N < \infty$ such that the aggregating consensus protocol (Algorithm 1) terminates in $\mathcal{G}_{(N)}(t_N)$ of order $n_N = 1$.*

Proof 3 *Without loss of generality we can assume that at time t_0 all agents are within pairwise distances greater than r_c and that the agents follow the linear consensus over $\mathcal{G}_{(0)}$ (Eq. (17)). Since $\mathcal{G}_{(0)}$ is connected then from Theorem 2 we obtain that $\lambda_2\left(\mathcal{G}_{(0)}\right) > 0$, and therefore, the system will converge towards agreement from any initial $x^{(0)}(t_0)$ (Theorem 3). Therefore, after a finite time $\delta_1 = t_1 - t_0$ a pair of agents will be within a distance r_c, such that $\pi_{(1)} = \pi_c\left(\mathcal{G}_{(0)}, x^{(0)}(t_1)\right)$ will contain a non-singlet cell. At that point a graph contraction event is triggered resulting in the reduced graph $\mathcal{G}_{(1)} = \mathcal{G}_{(0)} /\!\!/ \pi_{(1)}$ of order*

Algorithm 1 Aggregating consensus protocol

Initialize: $x^{(0)}(t_0) \in \mathbb{R}^{dn_0}$, $\mathcal{G}_{(0)}$ of order n_0

1. Integrate the system dynamics from time t_k according to the linear consensus protocol (4) over the graph $\mathcal{G}_{(k)}$ of order n_k

$$\dot{x}^{(k)}(t) = -\left[I_{d\times d} \otimes L\left(\mathcal{G}_{(k)}\right)\right] x^{(k)}(t). \tag{17}$$

2. Continuously calculate the contraction partition at time t, $\pi_c\left(\mathcal{G}_{(k)}, x^{(k)}(t)\right) = \pi_{cc}\left(\mathcal{G}_c\left(\mathcal{G}_{(k)}, x^{(k)}(t)\right)\right)$.

3. If $\pi_c\left(\mathcal{G}_{(k)}, x^{(k)}(t)\right)$ is not trivial,

   (a) perform a graph contraction $\mathcal{G}_{(k+1)} = \mathcal{G}_{(k)} /\!\!/ \pi_{(k+1)}$,

   (b) switch the dynamics to a consensus model over the contracted graph $\mathcal{G}_{(k+1)}$ of order $n_{k+1} < n_k$,

   (c) reinitialize the consensus system state as according to $x^{(k)}(t_k) = \left[I_{d\times d} \otimes \left(D^{-1}_{(k)} P^T_{(k)}\right)\right] x^{(k-1)}(t_k)$.

$n_1 < n_0$. *From theorem 1 we obtain that* $\mathcal{G}_{(1)}$ *is connected, and therefore,* $\lambda_2\left(\mathcal{G}_{(1)}\right) > 0$ *and the agents running the linear consensus must again converge to agreement from any initial condition, such that after a second finite time interval* $\delta_2 = t_2 - t_1$ *the system will contract again to a graph of order* $n_2 < n_1$. *Since the graph order monotonically decreases, i.e.,* $n_N < n_{N-1} < \ldots < n_0$, *then* $\exists N < \infty$ *such that after* N *contraction events the graph order must be* $n_N = 1$ *and the last contraction time is* $t_N = t_0 + \Delta_N$ *where* $\Delta_N = \sum_{i=1}^{N} \delta_i$. *All intervals are finite therefore* $\Delta_N < \infty$ *and* $t_N < \infty$.

IV. Case Studies

A. First Case Study - Aggregating Protocol with 8 Node Graph

As a first example, consider the aggregating consensus protocol (Algorithm 1) with initial simple graph $\mathcal{G}_{(0)}$ of order 8 with unit edge weights given in Fig.2. The initial state is $x^{(0)}(t_0) = \mathrm{col}\left(x_1(t_0), \ldots, x_8(t_0)\right)$ with $x_i(t_0) \in \mathbb{R}^2$ according to the positions in Fig.2.

As a reference the consensus model ($r_c = 0$) over $\mathcal{G}_{(0)}$ is simulated for 10 sec and the results are shown in Fig.3.

The aggregating protocol is then performed for $r_c = 0.1$ and all reduced graphs are assigned unit edge weights. The simulation results are shown in Fig.4.

For the same aggregating model we also examine the number of agent cells (Fig.5), the graph dynamics sequence $\left\{\mathcal{G}_{(k)}\right\}_{k=0}^{n_c}$ (Fig.6) and the variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$ (Fig.7), where n_c is the number of contractions performed from $n = n_0$ nodes to $n = 1$.

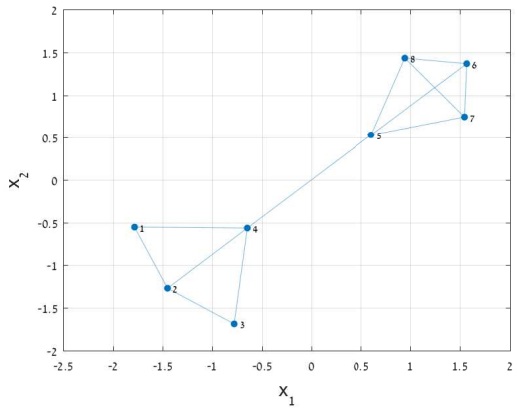
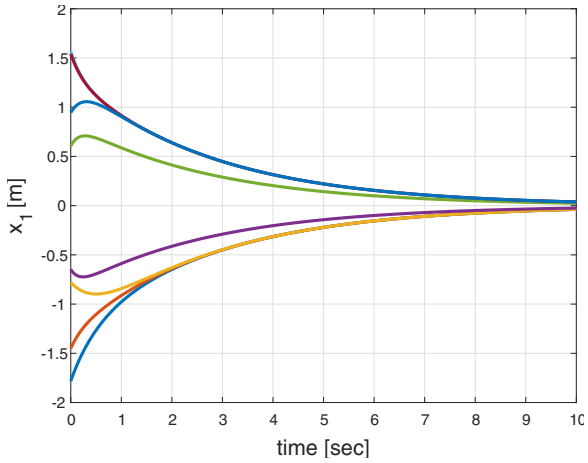
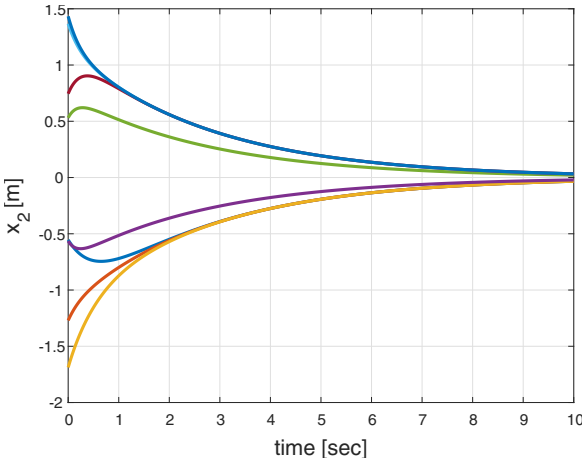


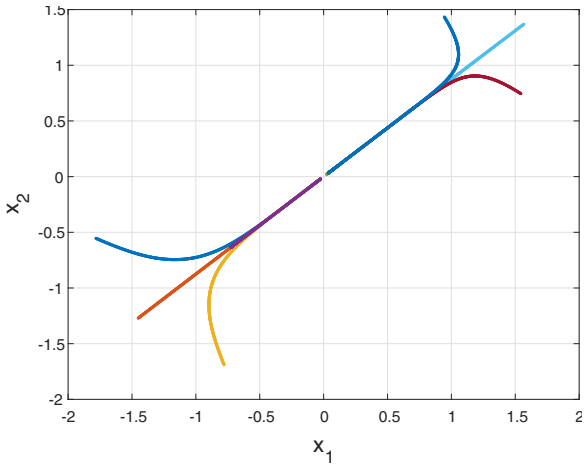
Figure 2: Simple graph of order 8



(a) x_1 coordinate dynamics

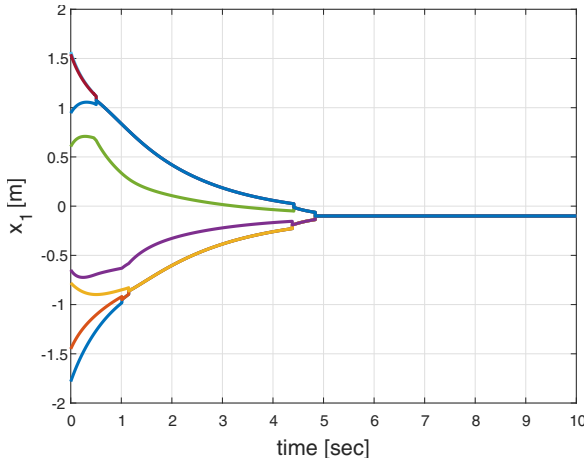


(b) x_2 coordinate dynamics

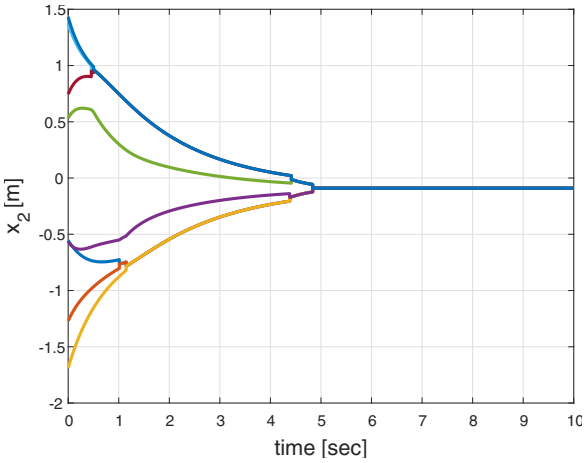


(c) x_1x_2 plane dynamics

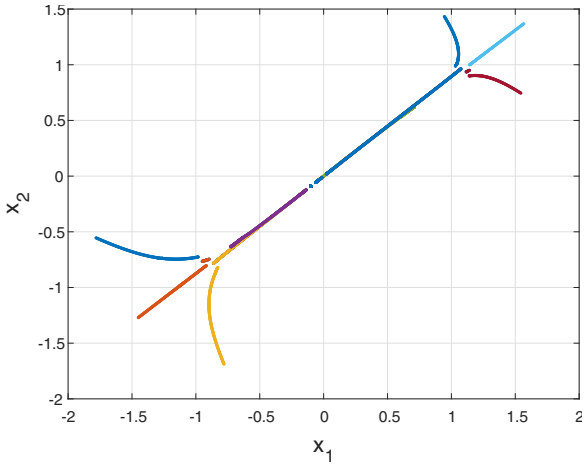
Figure 3: Linear consensus model over 8 node graph



(a) x_1 coordinate dynamics



(b) x_2 coordinate dynamics



(c) x_1x_2 plane dynamics

Figure 4: Aggregating protocol with initial 8 node graph

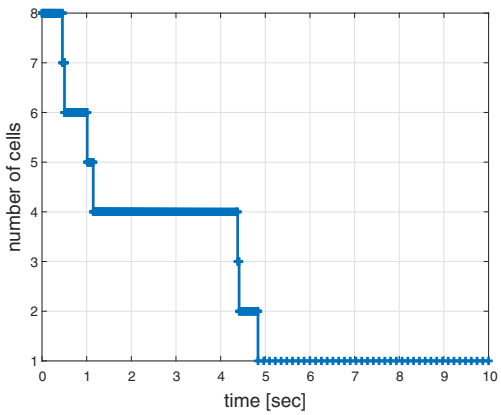


Figure 5: Number of agent cells during the aggregating protocol

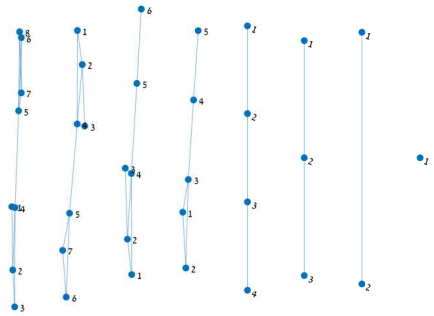


Figure 6: Graph contraction sequence during the aggregating protocol

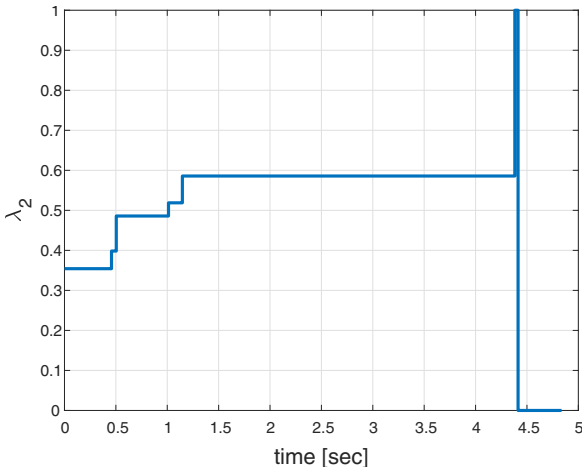


Figure 7: Variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$

We examine the effect of varying r_c on the aggregating dynamics.

For $r_c = 0, 0.01, 0.02, 0.05, 0.1, 0.2$ we obtain the number of agent cells (Fig.8), the variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$ (Fig.9) and the standard deviation of agent positions along both coordinates (Fig.10).

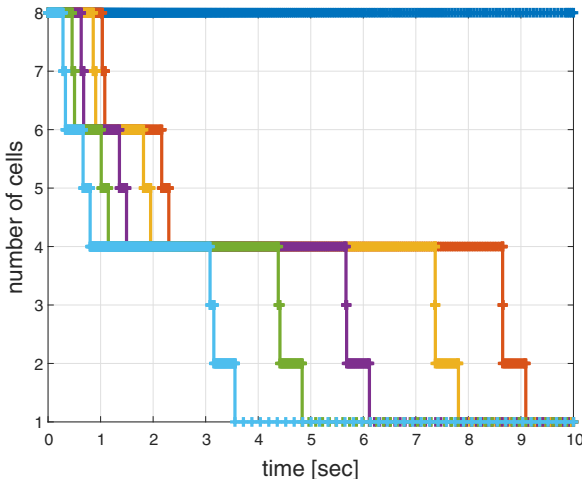


Figure 8: Number of agent cells with varying r_c

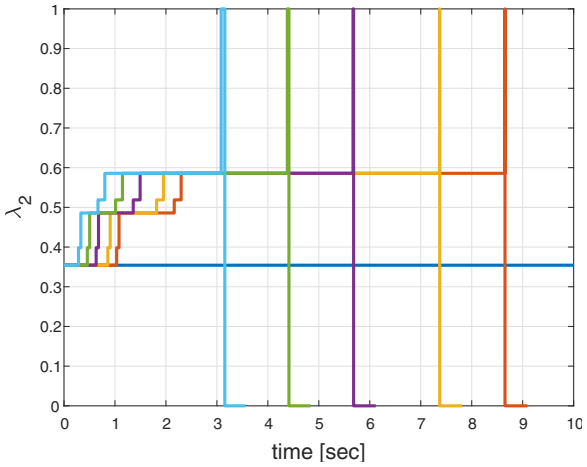
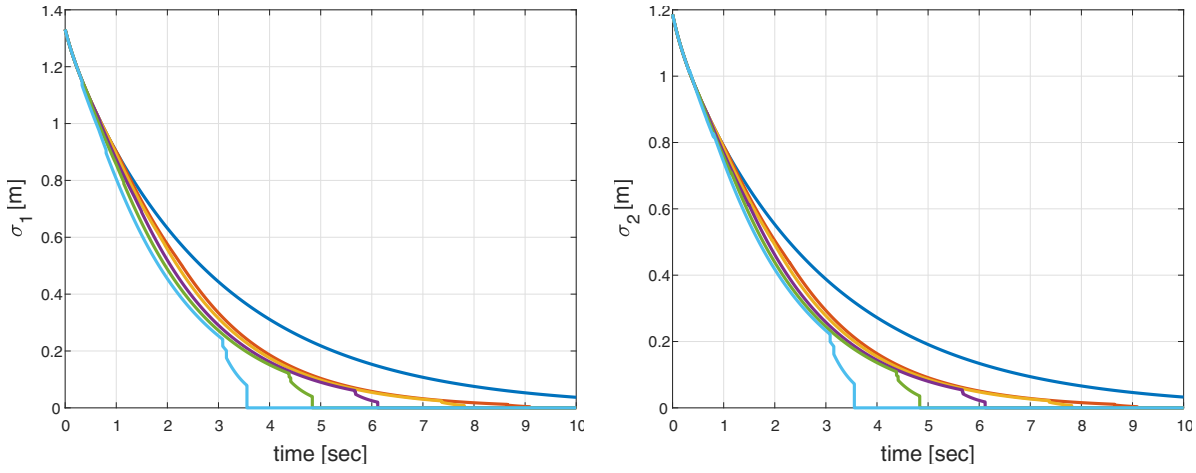


Figure 9: Variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$ with varying r_c



(a) σ_1 (b) σ_2

Figure 10: Standard deviation with varying r_c

B. Second Case Study - Aggregating Protocol with the Bucky Graph

Next we apply the aggregating protocol on a Buckminster Fuller ("Bucky") graph with unit edge weights (Fig.11). The Bucky graph is of order 60 with 90 edges (Fig.11). The initial state is $x^{(0)}(t_0) = \mathrm{col}(x_1(t_0), \ldots, x_{60}(t_0))$ with $x_i(t_0) \in \mathbb{R}^2$ according to the positions in Fig.11. The aggregating protocol is then performed for $r_c = 0.1$ and all reduced graphs are assigned unit edge weights. The simulation results are shown in Fig.12.

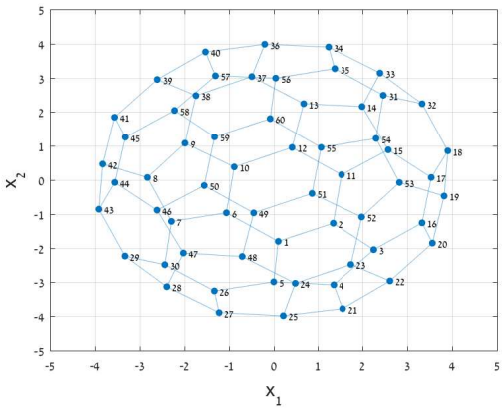


Figure 11: The Bucky graph

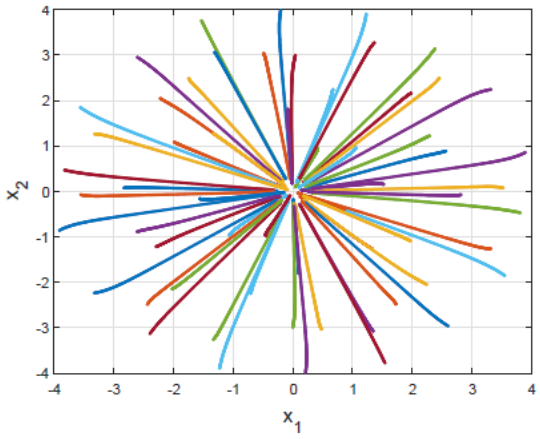


Figure 12: Aggregating protocol with Bucky graph x_1x_2 plane dynamics

Again we examine the effect of varying r_c on the aggregating dynamics. For $r_c = 0, 0.01, 0.02, 0.05, 0.1, 0.2$ we obtain the number of agent cells (Fig.13), the variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$ (Fig.14) and the standard deviation of agent positions along both coordinates (Fig.15).

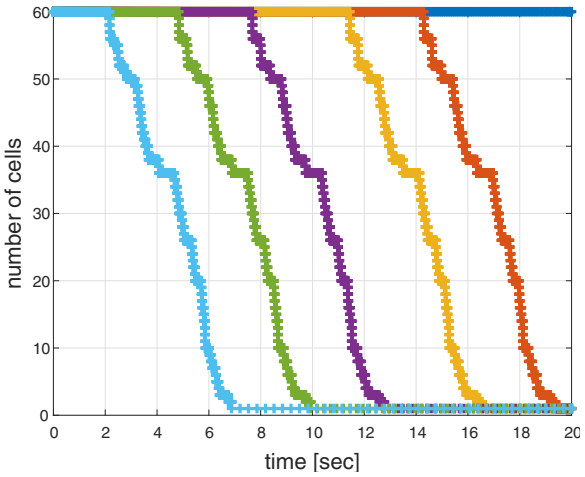


Figure 13: Number of agent cells with varying r_c

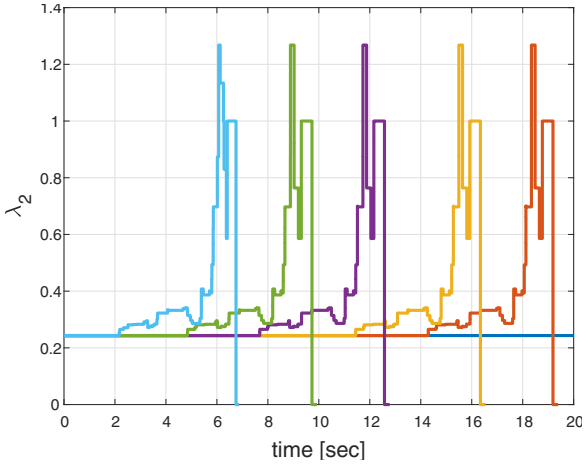
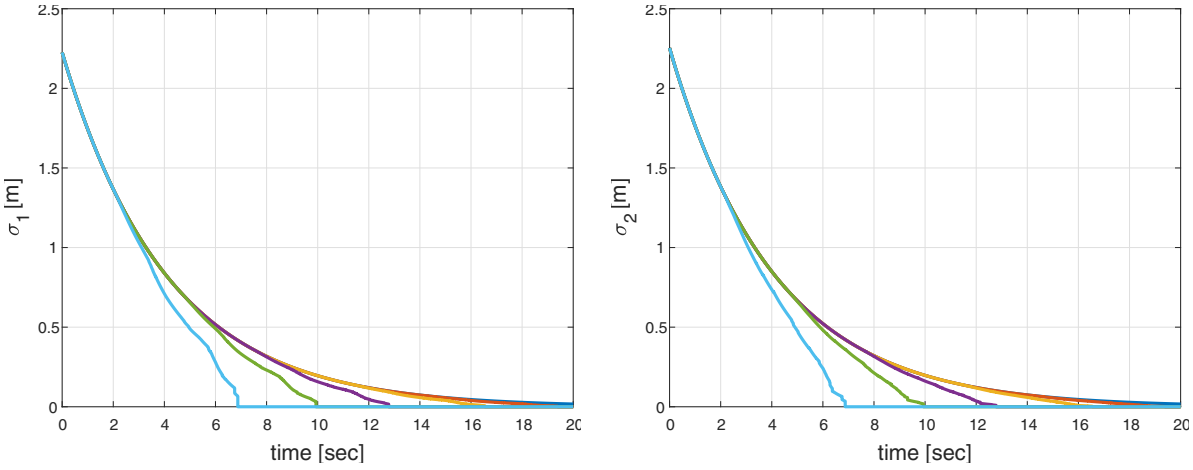


Figure 14: Variation in $\lambda_2\left(\mathcal{G}_{(k)}\right)$ with varying r_c



(a) σ_1 (b) σ_2

Figure 15: Standard deviation with varying r_c

C. Discussion

From the two case studies we observe, as expected, that the change in the contraction radius r_c determines the time for first contraction event, i.e., the larger r_c the sooner the first contraction will occur. A less obvious result is that after the first contraction event the system undergoes similar dynamical changes in the number of cells, in λ_2 and in the standard deviations, i.e., the rate of convergence, the number of merged cell as well as the change in λ_2 are only shifted with the change of the contraction radius r_c. We also observe that the change of magnitude of λ_2 is not preserved and not monotonically increasing or decreasing.

V. Conclusions

The aggregating consensus model has been presented as an example for a behavioral multi-agent system, and graph contractions were used as a model for merging agents. An event-triggered algorithm has been derived for the aggregating protocol and a proof of convergence of the aggregating process was provided.

From the two case studies we observe that when changing the contraction radius r_c the system preserved similar dynamical properties. This similarity is a matter of further research and investigation. As the rate of convergence is dictated by λ_2, the time for first contraction can be predicted from the initial dispersion and initial λ_2. We can also further exploit the adaptation of edge weights to maintain a constant λ_2.

References

1. Brooks, R. A. et al., "Intelligence without reason," *Artificial intelligence: critical concepts*, Vol. 3, 1991, pp. 107–63.

2. Brooks, R. A., "Intelligence without representation," *Artificial intelligence*, Vol. 47, No. 1-3, 1991, pp. 139–159.

3. Ayre, M., Izzo, D., and Pettazzi, L., "Self assembly in space using behaviour based intelligent components," *TAROS, Towards Autonomous Robotic Systems*, 2005.

4. Izzo, D. and Pettazzi, L., "Autonomous and distributed motion planning for satellite swarm," *Journal of Guidance Control and Dynamics*, Vol. 30, No. 2, 2007, pp. 449–459.

5. Nag, S. and Summerer, L., "Behaviour based, autonomous and distributed scatter manoeuvres for satellite swarms," *Acta Astronautica*, Vol. 82, No. 1, 2013, pp. 95–109.

6. Klavins, E., Ghrist, R., and Lipsky, D., "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 6, 2006, pp. 949–962.

7. Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, Vol. 49, No. 9, 2004, pp. 1520–1533.

8. Van Der Schaft, A. J. and Schumacher, J. M., *An introduction to hybrid dynamical systems*, Vol. 251, Springer London, 2000.

9. Dimarogonas, D. V., Frazzoli, E., and Johansson, K. H., "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, Vol. 57, No. 5, 2012, pp. 1291–1297.

10. Leiter, N. and Zelazo, D., "Graph-Based Model Reduction of the Controlled Consensus Protocol," *IFAC World Congress*, Toulouse, France, Sept. 2017.

11. Godsil, C. and Royle, G., *Algebraic Graph Theory*, Springer, 2001.

12. Schaeffer, S. E., "Graph clustering," *Computer science review*, Vol. 1, No. 1, 2007, pp. 27–64.

13. Newman, M. E. and Girvan, M., "Finding and evaluating community structure in networks," *Physical review E*, Vol. 69, No. 2, 2004, pp. 026113.

14. Mesbahi, M. and Egerstedt, M., *Graph theoretic methods in multiagent networks*, Princeton University Press, 2010.