2012-06

Uncooperative Multi-agent Communication Network Control, Hybrid LQ Approach

Cosby, J. Alan; Shtessel, Yuri B.; Bordetsky, Alexander

http://hdl.handle.net/10945/43687

2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Uncooperative Multi-agent Communication Network Control, Hybrid LQ Approach

J Alan Cosby, *Member, IEEE*, Yuri B Shtessel, *Member, IEEE,* Alexander Bordetsky

Abstract—**Retaining connectivity of a mesh network of non-cooperative multiple mobile agents is considered. The new connectability theory is applied to preserving network connectivity of non-cooperative mobile wireless networks. The newly introduced connectability matrix allows detecting a loss of network connectivity, as well as determining which nodes are isolated. The hybrid system techniques are used to describe the network dynamics, and LQR is used to control the motion of the relay agents in the communication network with changing dimension. A numerical example demonstrates the efficacy of the proposed techniques.**

I. Introduction

Mobile ad hoc wireless networks (MANETs) are used in military and civilian public safety communications applications [10]. These networks are self-forming and self-healing if multiple communications links are present in the network. Tactical communications provided by MANETs support transmission of text, voice, and video data generated by Primary Mission Agents (PAs) that perform tasks such as surveillance and reconnaissance without regard to maintaining communications. In this sense, the PAs are non-cooperative in maintaining network connectivity.

Usually cooperative multi-agent networks are considered and controlled using consensus-based techniques [8], [9]. The concept of consensus is not applicable in non-cooperative multi-agent networks for maintaining connectivity. Non-cooperative multi-agent networks [1] require the use of Relay Agents (RAs) to preserve network connectivity. The RAs are placed in the PA network in order to preserve network connections as the PAs perform their primary mission. Predicting PA node isolation, determining RA waypoint locations and robustly controlling RAs, while driving them to the waypoints, is the scope of this paper. A novel concept of graph connectability is proposed in this paper and applied to the task of preserving network connectivity. The use of the novel connectability matrix allows identifying not only the isolated nodes but the number of RAs needed to connect the isolated nodes. The hybrid system techniques are used to describe the network dynamics, and LQR is used to control the motion of RAs in the network with changing dimension.

The paper is organized as follows. The problem is presented in Section II. Section III defines the newly developed connectability theory and traditional graph analysis techniques are compared to the connectability theory. Application of connectability theory to mobile ad hoc wireless communication networks is presented in Section IV. Control strategy is presented in Section V, while Section VI contains simulation examples of network connectivity preservation. Section VII concludes the paper.

II. Communication Network Math Model

A. Problem Formulation

The task of preserving connectivity in a MANET involves three main issues that are addressed in a concert:

- **Prediction of the loss of connectivity** for a dynamic graph on a given time window via the proposed approach based on newly introduced connectability matrix.
- **Computation of the waypoints** may be as simple as computing the midpoint between an isolated node and its nearest neighbor, if only one node is predicted to be isolated and there is at least on available RA to dispatch to that point. If more nodes are predicted to be isolated than the number of available RAs, then all RAs need to be utilized in order to minimize the distance between the isolated nodes and RAs, and between the RAs themselves.
- **Control and observation** must drive the RAs to the waypoints before the predicted isolation occurs. The control algorithm may change between time windows depending on the conditions of the network on each time window. This issue can be viewed as a hybrid dynamical system where the system state may switch at distinct time window transitions. Depending on the predicted network connection loss, the dimension of the system state may increase or decrease at time window transitions.

B. An Illustrating Example

As shown in Fig. 1 the network initially consists of four PAs. As the PAs move, node 1 moves from $q_1(t_0)$ to $q_1(t_1)$ resulting in node 1 predicted to be isolated. To prevent the isolation of node 1, an RA is moved into place between node 1 and node 2. The addition of the RA at time t_1 adds one vertice to the system graph.

J. Alan Cosby is with the Department of Electrical and Computer Engineering, The University of Alabama Huntsville, Huntsville, AL 35899 (email: jac0014@uah.edu).
Yuri B. Shtessel is with the Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, Huntsville, AL 35899 USA (phone: 256-824-6164; fax: 256-824-6803; e-mail: shtessel@eng.uah.edu).
Alexander Bordetsky is with the Department of Information Technology, Naval Postgraduate School, Monterey, CA, (email: abordets@nps.edu)

978-1-4577-1096-4/12/$26.00 ©2012 AACC

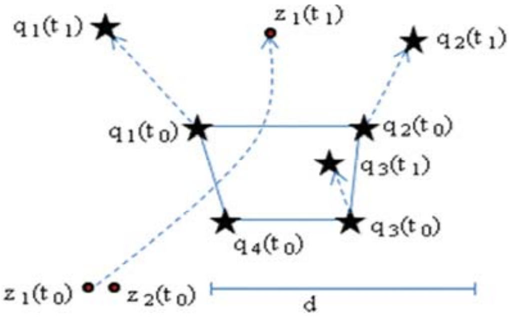

Fig. 1 Sample network graph of four PAs and two RAs

With each additional RA entry to the network the state dimension increases and the system is termed a hybrid system as shown in Figure 2. These jumps in the dimension of the state occur at distinct, but not known a priori, t_j times based on the inter-node distances. In Figure 2, the Γ_{ij} are the guard conditions that cause the state transitions to a higher order. The R_{ij} are the reset conditions that reduce the dimension of the system state. If more disconnections are predicted than the number of RAs in the system, the communications disruption may be prevented by engaging all RAs to minimize the distance between the PAs and between the other RAs.

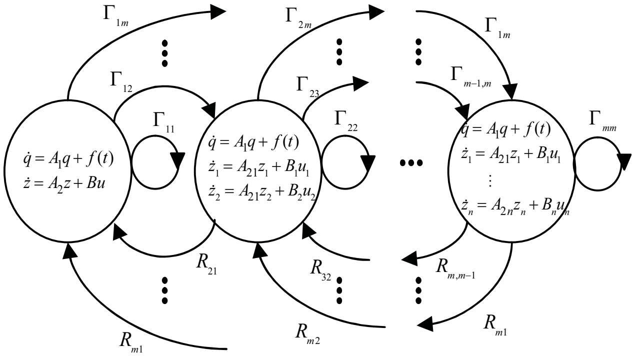

Fig. 2 The state transitions of the hybrid dynamical system.

Each mission time window is considered to be an interval [0, T]. For each time interval

$$\left\{(t_j, t_j+\Delta T)\in[0,T] \,\middle|\, \Delta T << T, j=1,2,\dots,\frac{T}{\Delta T}\right\}$$

The future positions of the PAs are predicted based on their positions and their estimated velocities at $t = t_j$. The velocities are estimated using HOSM observers [6]-[7]. When a loss of communication is predicted at $\tilde t \in (t_j, t_j + \Delta T)$ the network is analyzed to determine if already dispatched RAs can be utilized to prevent the network loss. If already dispatched RAs cannot prevent the disconnection, and there is a RA available, the commanded position (waypoint) $z(\tilde t)$ is "dropped" on the midpoint of the predicted cut edge. When more disconnects are predicted than available RAs, the waypoints are "dropped" to minimize the disconnections. The minimization technique is discussed in Section IV. The RAs are guided to the commanded position using hybrid LQ control. When the RAs arrive at their commanded positions, the time next t_j is set to $\tilde t$, and the process is repeated until $t_j + \Delta T = T$.

C. Communication Network Math Model

The position of each PA is $q_i \in R^2$ and the velocity of an agent is p_i. The unknown PA control commands are $f_i(t) \in R^2$ that is a smooth, bounded vector field. The position of each RA is $z_k \in R^2$ and the velocity is denoted g_k. The system dynamics is represented as:

$$\begin{cases}\dot q = A_1 q + f(t)\\ \dot z = A_2 z + Bu\end{cases} \quad where \quad \begin{cases}\dot q_i = p_i\\ \dot p_i = f_i(t)\end{cases} \quad \begin{cases}\dot z_k = g_k\\ \dot g_k = u_k\end{cases} \qquad (2.1)$$

Remark*. The function $f_i(t)$ is the external commands from the ground control station of the i^{th} PA and are unknown by the controller that drives the RAs to maintain network communications.*

As RAs are added to the system at distinct instances t_j when the loss of connectivity is predicted, the dimension of the state vector increases and system equations of (2.1) become

$$\dot q = A_1 q + F(t)$$

$$\dot z = \begin{cases}\dot z_1 = A_{21} z_1 + B_1 u_1\\ \dot z_2 = A_{22} z_2 + B_2 u_2\\ \vdots\\ \dot z_k = A_{2k} z_k + B_k u_k\\ \vdots\\ \dot z_m = A_{2m} z_m + B_m u_m\end{cases} \qquad (2.2)$$

The waypoint $z_{k+1}(t_j)$ for $(k+1)^{th}$ RA at $t = t_j$ is computed based on the prediction of discontinuity as described in Section IV.

D. Estimation of PA Velocity

The q_i position of each PA is assumed to be known. The p_i velocity of each PA is estimated by Higher Order Sliding Mode (HOSM) observers [6] as:

$$\begin{aligned}&\dot\zeta_0 = v_0,\ v_0 = -3L^{1/3}|\zeta_0 - q_i|^{2/3} sign(\zeta_0 - q_i) + \zeta_1\\ &\dot\zeta_1 = v_1,\ v_1 = -1.5L^{1/2}|\zeta_1 - v_0|^{1/2} sign(\zeta_1 - v_0) + \zeta_2\\ &\dot\zeta_2 = -1.1Lsign(\zeta_2 - v_1)\end{aligned} \qquad (2.3)$$

where L is a Lipchitz constant such that $L > \max\|f_i(t)\|$.

The HOSM observer in equation (2.3) converges to the estimated PA velocity $\zeta_1 \to p_i$ and acceleration $\zeta_2 \to f_i(t)$ in finite time. The estimated velocity and reconstructed acceleration is used in equation (4.1) to calculate the position q_i on each interval (t_j, t_{j+1}).

III. Connectability Theory and Comparison to Traditional Techniques

A. Traditional Graph Theory

The scenario depicted in Figure 1 is described using traditional graph theory terminology. The agents are considered to move in the $\square^2$ plane. The interconnection of adjacent agents is modeled as an undirected graph $G_n : (V_n, \Omega_n)$ where $V_n : \{1,2,...,n\}$ is a finite set of vertices consisting of n agents and $\Omega_n \subseteq \{(i,j) : i,j \subset V_n, i \neq j\}$ is the edges of G which are the communication links between agents. When an edge exists between agents, the agents are said to be adjacent. The set of vertices adjacent to a given vertex is called the neighborhood, $N_i \subseteq V : \{j \in V \mid ij \in \Omega\}$ of the vertex. The graph is considered undirected because the existence of an edge between agents means that bidirectional communications are insured. The distance between two adjacent agents i and j is given by

$$\left\| q_i - q_j \right\| = r_{ij} = r_{ji} > 0 \tag{3.1}$$

Definition 1* [1].** An undirected graph $G_n : (V_n, \Omega_n)$ is ***d*-connected if $r_{ij} \le d^*, \forall i,j : (i,j) \in \Omega_n, d^* > 0$. The network is ***d*-connected*** if the graph is d*-connected. The graph is ***d*-disconnected*** if $r_{ij} > d^*$ for at least one edge $(i,j) \in \Omega_n$; the network is ***d*-disconnected***, and at least one agent is ***isolated***. The edge of the graph for which $r_{ij} > d^*$ is said to be ***cut*** [5].

***Definition 2* [1].** A **path** in the graph is given by a sequence of adjacent vertices. A path consisting of n vertices includes $n-1$ edges. If the vertices of a **path** are distinct except for the end vertices, the path is called a cycle. The degree of vertex i in an undirected graph, d_i, is the number of vertices that are adjacent to that vertex and in G, that is the degree is the cardinality of the neighborhood N_i of vertex i.

For instance, nodes 1 through 4 of Figure 3 form a path.

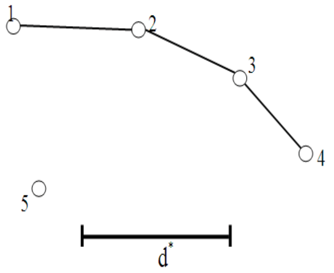

Fig. 3 A d*-connected graph (nodes 1 through 4).

Definition 3 [5].The ***degree matrix***, $\Delta(G)$, of graph G is a diagonal matrix $\Delta(G) = diag\{d_i\}$

Definition 4 [5]. The ***adjacency matrix***, $A(G)$, of graph G is a symmetric matrix containing the adjacency relationships of each vertex in graph G. $A(G)$ is computed as:

$$A_{ij}(G) = \begin{cases} 1 & if \quad r_{ij} \le d^*, i \neq j \\ 0 & if \quad r_{ij} > d^*, or, i = j \end{cases} \tag{3.2}$$

Definition 5 [5]. The ***graph Laplacian matrix***, of G is the difference of the degree of G and the adjacency of G:

$$L(G) = \Delta(G) - A(G) \tag{3.3}$$

The graph Laplacian is also known as the Kirchhoff matrix of the graph. By Kirchhoff's matrix tree theorem, the value of any cofactor of the Laplacian matrix is equal to the number of spanning trees of the graph [4]. If a zero cofactor exists then one or more edges are cut and the graph is d-disconnected. *Another way of determining the loss of connectivity* is to find the eigenvalues of the graph Laplacian. The eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n, \lambda_1 = 0$ for $i > 1$ will be greater than zero if the graph is d*-connected. If any eigenvalues are equal zero other than the first eigenvalue, then the graph is d*-disconnected [5]. *Note that* $\lambda_i = 0, i \ge 2$ *only indicates that an isolated node exists, it does not indicate which node is isolated or the distance from an isolated node and its nearest neighbor.*

B. Connectability Theory

Connectability theory builds on traditional graph theory so that a single matrix, the **connectability matrix**, indicates not only that an isolated node exists but also which node is isolated and the distance between an isolated node and its nearest neighbor. First some terms are defined.

Definition 1. The isolated i^th node in a d*-disconnected graph is called **k-connectable** if the minimal distance between the node and its nearest neighbor in the graph is less than a distance $\overline{d}$, such that $(k-1) \le d^* < \overline{d} \le k < d^*$ where k is an integer. The number of additional nodes required to make a k-connectable node d*-connected is k-1. Figure 4 contains a **k-connectable** node.

For instance Node 1 in Figure 4 is a distance of $2 \cdot d^*$ from its nearest neighbor, node 2.

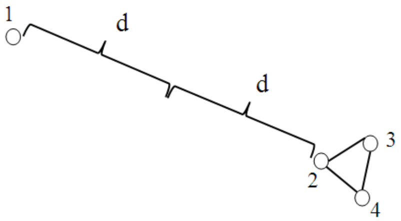

Fig. 4 A k-connectable node (Node 1)

Corollary. If an isolated node in an undirected graph **is k-connectable** then a **k-path** that consists of a chain of d*-connected nodes exists between this node and its nearest neighbor. In Fig. 4, the path from 1st to 2nd is a k-path.

Definition 2. An undirected graph is called k-connectable if all nodes in the graph are connected by at least one k-path.
Corollary. The **d*-connected** graph is **1-connectable**.

Definition 3. The symmetric matrix

$$\left[C\right]_{ij} = \begin{cases} k = ceil(\left[D\right]_{ij} / d^*) & if \quad i \neq j \\ 0 & if \quad i = j \end{cases} \tag{3.4}$$

is called **a connectability matrix** for a graph with at least one isolate node**,** where

$$D = \left[D\right]_{ij} \quad i, j \in V_n$$

is a distance matrix, whose elements represent distances between ith and jth node, and V$_n$ is the finite set of the graph's vertices.

The following properties of the connectability matrix (3.4) can be observed:
Property 1. The connectability matrix of an undirected graph is symmetric.
Property 2. The diagonal elements of the connectability matrix are zero, and the off diagonal elements are the multiples of d* between the ith and jth nodes.
Property 3. If C]$_{ij}$=1, the nodes are connected, otherwise, the value of [C]$_{ij}$-1>0 (i.e. k-1) is the number of additional nodes necessary to connect nodes i and j.

Theorem 1. The graph with a **k-connectable** isolated node can be made **d*-connected** by inserting k-1 additional nodes between the isolated node and already d*-connected nodes.

Proof. The matrix constructed using equation (3.4) has off-diagonal elements equal to the k-connectability of the edge between nodes i and j. As stated in Property 3, [C]$_{ij}$=1, the nodes are 1-connected. If $\left[C\right]_{ij} = k > 1$, then $\left[D\right]_{ij} = k \cdot d^*$ and from definition 2 in Section III-A, the d*-connected path between nodes i and j includes k-1 nodes and the theorem is proven.

Theorem 2. The ith node is k-connectable to the jth node with k=[C]$_{ij}$.

Proof. By examining the connectability matrix, isolated nodes may be identified. Using a metric (note this corresponds to the diagonal elements of the Laplacian matrix)

$$m_i = \sum_{j=1,n} \left(\left[C\right]_{ij} = 1 \right) \tag{3.5}$$

gives the number of adjacent connected nodes of node i. Therefore, if m$_i$ in equation (3.5) is zero, then node i is isolated, and the off-diagonal elements indicate the k-connectability of node i to its nearest neighbors. This means that min$\left(\left[C\right]_{ij} - 1 \right)$ is the least number of additional nodes required to connect nodes i and j (if $m_i = 1$ node i is 1-connected). The theorem is proven.

C. Connectability Theory Compared to Traditional Graph Theory

Traditional graph theory involving the construction and evaluation of the Laplacian matrix is presented above. The graph is connected if $eig_1(L) = 0$ and $eig_i(L) > 0,\ i = 2,3,...$ whereas the graph has at least one isolated node if $eig_2(L) = 0$. Unfortunately, $eig_2(L) = 0$ does not indicate which node is isolated. In the previous work [1], the isolated nodes of a dynamic mobile network are determined by comparing the adjacency matrix before and after $eig_2(L) = 0$.

In contrast, the connectability matrix constructed as in equation (3.4) indicates the existence and location of an isolated node. Additionally, for row i, if the metric m$_i$=0 then the minimum element in row i is the k-connectability of node i and k-1 nodes must be added between node i and j to connect i and j.

IV. Network Disconnect Detection

A. Maintaining Network Connectivity

In [1], the algorithm used to predict cut edges and the time at which d*-connectivity will be lost was based on traditional graph theory. The eigenvalues of the graph Laplacian were computed to indicate an isolated node, and the adjacency matrix before and after the disconnected network prediction were compared to determine which node was isolated. The algorithm in this paper based on connectability theory is presented below.

The *current* time instant $t = t_j$ is fixed. Virtual time increases from t_j to t_{j+1} by small simulation steps $\Delta T << T$. The following algorithm steps are repeated in sequence on each ΔT simulation step. If the entire virtual time frame $[t_j, t_{j+1}]$ is processed and no loss of connectivity is predicted, the real time is allowed reach $t = t_{j+1}$, and the algorithm is repeated. If a disconnection is predicted at $\tilde{t} \in [t_j, t_{j+1}]$ and no other RAs have been dispatched, a RA is dispatched (as described in the following subsection), its trajectory is defined, the graph is updated, and the algorithm is re-initialized once the real time reaches $t = \tilde{t}$.

The algorithm is accomplished in the following sequence of steps.
Step 1. The distance matrix, $D(G)$ of the graph is constructed.
Step 2. The connectability matrix, $C(G)$ of the graph is constructed as in equation(3.4).
Step 3. The connectability metric, m$_i$, is computed as shown in equation(3.5), and if $m_i = 0$ for any $i = 1,2,...,n$ then node i is predicted to be isolated and the graph is predicted to be d*-disconnected.
Steps 4 and 5 below are executed only if loss of network connectivity is predicted in Step 3.
Step 4. Since d*-connectivity of the graph is checked in Step 3 at each simulation step, the instant $\tilde{t} \in [t_j, t_{j+1}]$ of its loss

can be identified. The minimum element of row i in $[C]_{ij}$=k and k-1=the number of RAs needed to connect node I to node j.

Step 5. Equation (4.1) is solved using the current (measured) position $q_i(t_j)$, current (estimated) velocity $p_i(t_j)$, and acceleration $f_i(t_j)$ (assuming the acceleration is constant on the time interval $[t_j, t_{j+1}]$) for all agents belonging to the predicted cut edge(s):

$$\begin{aligned} q_i(t) &= q_i(t_j) + p_i(t_j)t + \frac{f_i(t_j)}{2}t^2 \\ p_i(t) &= p_i(t_j) + f_i(t_j)t \end{aligned} \tag{4.1}$$

Equation (4.1) is evaluated at each $t + \Delta T \in [t_j, t_{j+1}]$to estimate the positions and velocities of the agents. If more edge cuts are predicted than the number of RAs, equation (4.1) is used to calculate the predicted positions of the PAs at $t = \tilde{t}$, and the new waypoints of the RAs is given by (4.2). If RAs have already been dispatched when node isolation is predicted, the already dispatched RAs are evaluated using equation (4.4) to verify if they can be used to prevent the loss of communications. The possible new waypoints are dropped on a "phantom" graph and the connectability matrix and metric of the phantom graph are calculated. If $m_i > 0 \quad \forall \quad i = 1,2,...,n$ the phantom graph becomes the actual graph and the already dispatched RAs are driven to the computed waypoints.

Remark [1]. *It is assumed that if the disconnection of the network is predicted on the time interval* $[t_j, t_{j+1}]$*, then this event will definitely happen during this time interval*

B. Generation of Waypoints for Controlled Agents

When a disconnection is predicted then:

1) The waypoints for the already dispatched RAs are recalculated in order to minimize the distances between the waypoints and the predicted positions of the PAs. In particular equation 3.11 is minimized for z_k, k=1,2,..m.

2) The connectivity of the new phantom graph is assessed using the process described in the previous section. If $m_i > 0$ then the waypoints computed in step 1 are sufficient and the phantom graph becomes the new graph at time t_{j+1}, and the RAs are driven to those waypoints.

3) If $m_i = 0$in step 2, and there is another RA that has not been dispatched, the waypoint for that RA is set at the midpoint between the two PAs where the new discontinuity is predicted. If there are no more RAs available, then the waypoints computed in step 1 minimize the loss of communications and the RAs are driven to those points.

In step 3, when a loss connectivity ($r_{ij} > d^*$) is predicted, then a new RA is dispatched. The RA is driven to arrive at time $t = \tilde{t}$ to the midpoint of the cut edges(i, j), $z_k(\tilde{t}) = \frac{q_i + q_j}{2}$.

For the step 1, the inter-agent distance is minimized using

$$\begin{aligned} \min_{Z_k} \Bigg[&\sum_{i=1}^{n}\sum_{k=1}^{m}\left(\rho_{ik}\left\|z_k(\tilde{t}) - x_i(\tilde{t})\right\|^2\right) \\ &+ \sum_{j=1}^{m}\sum_{k=1}^{m}\left(\rho_{jk}\left\|z_k(\tilde{t}) - z_j(\tilde{t})\right\|^2\right)\Bigg] \end{aligned} \tag{4.2}$$

To find the new waypoints, $z_k(\tilde{t})$of the RAs, equation (4.2) must be differentiated as

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{k=1}^{m}\frac{d}{dz_k}\left(\rho_{ik}\left\|z_k(\tilde{t}) - x_i(\tilde{t})\right\|^2\right) \\ &+ \sum_{j=1}^{m}\sum_{k=1}^{m}\frac{d}{dz_k}\left(\rho_{jk}\left\|z_k(\tilde{t}) - z_j(\tilde{t})\right\|^2\right) \end{aligned} \tag{4.3}$$

So each $z_k(\tilde{t})$is given by

$$z_k(\tilde{t}) = \frac{\sum_{i=1}^{n}\rho_{ik}x_i(\tilde{t}) + \sum_{j=1}^{m}\rho_{jk}z_j(\tilde{t})}{\sum_{j=1}^{m}\rho_{jk}} \tag{4.4}$$

As shown in Figure 1, the position of a RA is denoted $z_k \in \Re^2$, and the initial position of the RAs is at the origin. In practice, the RAs would be loitering at a specified "safe" location but as close as possible to the operations of the PAs.

After the RAs arrive in their designated positions, the graph becomes $G_{n+k} : (V_{n+k}, \Omega_{n+k})$which contains additional vertices and the cut edges are replaced by 2 new edges for each dispatched RA. The new graph G_{n+k} is d*-connected. If there were not enough RAs to compensate for the number of cut edges, the new graph$G_{n+k} : (V_{n+k}, \Omega_{n+k})$is evaluated to determine if the addition of the available RAs has resulted in a d*-connected graph.

C. Collision Avoidance

The PAs collision avoidance is assumed to be performed by each PAs ground station, but these ground stations do not receive the locations of the RAs. Therefore, the RA's control must provide collision avoidance between RAs or with the PAs. The trajectory of the RA must not intersect the trajectories of PAs or other RAs.

Collision avoidance algorithms are used in game applications as described here. A small radius ρ is defined around each agent, called a security zone. A collision occurs if the security zones of any agents intersect.

In accordance with [1], collisions are predicted using an algorithm similar to the connectivity algorithm described above. The *current* time $t = t_j$ is fixed. Virtual time increases from t_j to t_{j+1}, the predicted disruption time, by incrementing small simulation steps$\Delta T << T$. The following algorithm is repeated at each ΔT simulation step. If the virtual time interval $[t_j, t_{j+1}]$ is processed without a predicted collision, the real time is allowed reach $t = t_{j+1}$, and the algorithm is repeated. If a collision is predicted at some $\tilde{t} \in [t_j, t_{j+1}]$, a loiter or via point [4] is introduced and a new trajectory is calculated from the current position $z_k(t_j)$ to the via point, and from the via point to the destination $z_k(t_{j+1})$.

V. Control Strategy

A high level supervisory application in the Network Operations Center performs the algorithms described in Section IV. The low level supervisory algorithm computes the local control laws in order to drive the RAs to the waypoints. Use of high order sliding mode (HOSM) control for this task has been demonstrated [1]. In this sequel, a hybrid LQ strategy is performed.

It has been shown that a hybrid LQ strategy for driving RAs to waypoints can be used to control a system with incremental increases in the state dimension [2].

The control u_k of equation (2.2) is calculated using hybrid linear quadratic (LQ) control in order to minimize the function

$$\begin{aligned}\min_{u_k} J_k &= \frac{1}{2}(e_k(t_{j+1}))^T S_k(e_k(t_{j+1})) \\ &+ \frac{1}{2}\int_{t_j}^{t_{j+1}} (e_k(t))^T Q_k(e_k(t)) + (u_k^T R_k u_k) d\tau, \;\; e_k(t) = z(t) - \overline{z}(t_{j+1})\end{aligned} \quad (5.1)$$

From the hybrid Maximum Principle [2], the Hamiltonians associated with the j and $j+1$ states are

$$\begin{aligned}&H_k(t,z,u,\psi) = \left\langle \psi_k, A_k(t) z_k + B_k(t) u_k \right\rangle \\ &-\frac{1}{2}\left(z_k^T Q_k(t) z_k + u_k^T R_k(t) u_k\right)\end{aligned} \quad (5.2)$$

In this case, P_k satisfies the differential Riccati equation,

$$\begin{aligned}&\dot{P}_k(t) = -P_k(t) A_k(t) - A_{k1}^T(t) P_k(t) \\ &+P_k(t) B_k(t) R_k^{-1}(t) B_k^T(t) P_k(t) - S_k(t), \quad P_k(t_{j+1}) = S_k\end{aligned} \quad (5.3)$$

Solving equation of (5.3) for P_k on each jump of system dimension allows the computation of LQ type optimal control as $u_k(t) = -R_k^{-1}(t) B_k^T(t) P_k(t) z_k(t), t \in [t_j, t_{j+1}]$ [2].

VI. Case Study

A two dimensional Simulink simulation was constructed with four PAs and two RAs. The initial position of the RAs was the origin and the PAs were initially positioned in a quadrilateral as shown in Figure 5. The maximum communications range was set at d=200. Two of the PAs were commanded to move so that their inter-agent distance would exceed 200. The edge r_{12} exceeds 200 at 0.73 seconds at which time RA$_1$ is in position between the two PAs. Edge r_{23} exceeds 200 at 3.334 seconds, and RA$_2$ is in position between PA$_2$ and PA$_3$ at that time.

The plots contained in Figs. 5 and 6 demonstrate the algorithms presented in sections III and IV. In Fig. 5, the movement of two PAs results in the predicted isolation of the PAs. The two available RAs are commanded to the positions between the two isolated PAs and one of the other connected PAs in order to maintain the communications network. Fig. 6 shows the minimization technique where movement of the PA would result in all four PAs being isolated. The two available RAs are commanded so as to minimize the communications loss. In this case, the communications links to all four PAs remains intact.

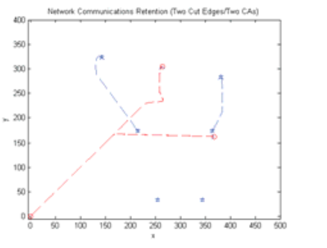

Fig. 5 Two isolated PAs

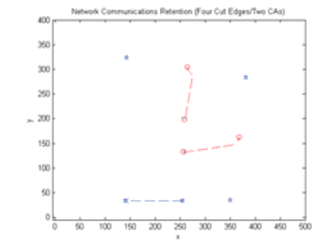

Fig. 6 Four isolated PAs.

VII. Conclusions

The algorithms for minimizing the communication loss and driving the RAs to their waypoints were found to successfully address the task of maintaining communication connectivity of a mobile communication network of uncontrolled agents maneuvering to perform their own tasks. The network of controlled agents was able to preserve communications among the PAs, and the hybrid system created by the introduction of new RAs was controlled using LQR techniques.

References

[1] S. Holleran, S. Baev, Y. Shtessel, "Preventing Disruption of a Mobile Communication Network using Higher Order Sliding Mode Control," Proceedings of the Conference on Decision and Control, December, 2010.

[2] R. Galvan-Guerra, V. Azhmyakov, and M. Egerstedt, "Optimization of Multiagent Systems with Increasing State Dimensions: Hybrid LQ Approach," 2011 American Control Conference.

[3] V. Azhmyakov, R. Galvan-Guerra, A. Panyak, "On the hybrid LQ-based control design for linear networked systems," Journal of the Franklin Institute., vol. 347, pp. 1214-1226, 2010.

[4] F. Harary, "Graphs and matrices," *SIAM Review*, vol. 9, no. 1, pp. 83-90, Jan. 19

[5] Mehran Mesbahi and Mangus Egerstedt, "Graph Theoretic Methods in Multiagent Networks", Princeton Univ Press, Princeton, NJ, 2010.

[6] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," Int. J. Control, vol. 76, # 9/10, pp. 924-941, 2003.

[7] A. Levant, "Quasi-continuous high-order sliding-mode controllers," IEEE Trans. Autom. Control, vol. 50, no. 11, pp. 1812-1815, Nov. 2005.

[8] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," IEEE Control Syst. Mag., vol. 27, issue 2, pp. 71-82, Apr. 2007.

[9] G. Schöner, M. Dose, and C. Engels, "Dynamics of behavior: theory and applications for autonomous robot architectures," Robotics and Autonomous Systems, vol. 16, pp. 213-245, 1995.

[10] Bordetsky A., and Bourakov, E., "Network on Target: Remotely Configured Adaptive Tactical Network," *In Proceedings of the 10th Command and Control Research & Technology Symposium*, San Diego, CA, 2006.