2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Cuts and Cycles in Relative Sensing and Control of Spatially Distributed Systems

Jasmine Sandhu, Mehran Mesbahi, and Takashi Tsukamaki

***Abstract*— We consider transformations that characterize equivalent classes of relative sensing and control topologies for spatially distributed systems. We employ tools from algebraic graph theory, and in particular, notions associated with cut and cycle spaces of a graph, to derive explicit formula for such characterizations. Simulation results, demonstrating the utility of the developed framework in the context of reconfigurable control, conclude our presentation.**

***Index Terms*— Distributed sensing; networked control; algebraic graph theory**

I. INTRODUCTION

Our goal in this work is to provide a deeper understanding of how the sensing geometry in a spatially distributed dynamic system influences the control system design. The general control configuration is shown in Figure 1 where the signal z captures the coordination states; signals x, w, y, and u, denote respectively, the system state comprised of states of the individual dynamic elements, exogenous signal, the measurement signal available to the controller (sensed or communicated), and finally, the control input. The control objective is assumed to be maintaining a particular coordination among the states of the various dynamic elements. In many such scenarios, for example when the distributed dynamic system corresponds to a multiple vehicle system, the coordinated states are the relative states among each pair of elements. In this case, the vector z in Figure 1 consists of vectors of the form $x_i - x_j$ $(i \neq j)$. Since the control objective is achieving a set of performance measures defined on the relative states, it is natural to assume that the information available to the controller also consists of a subset of these relative states (again either measured or communicated). The main question that we would like to address in this paper is as follows: suppose that a controller has been designed for a spatially distributed system in order to achieve a particular control objective. Furthermore, suppose that this controller was constructed based on a particular underlying information geometry. Are there transformations that allow for a seamless computation of an *equivalent* controller when the underlying information geometry changes? The solution to this problem turns out to not only provide a reconfiguration capability in the control law, but also provide a deeper insight into the problem

This research was supported by a grant from Phantom Works, The Boeing Company.

The first two authors are with the Department of Aeronautics & Astronautics at the University of Washington, Seattle, WA. Emails: $\{$*sandhu+mesbahi*$\}$ @*aa.washington.edu*. T. Takashi is with Boeing Phantom Works, Seattle, WA. Email: *takashi.tsukamaki@boeing.com*.

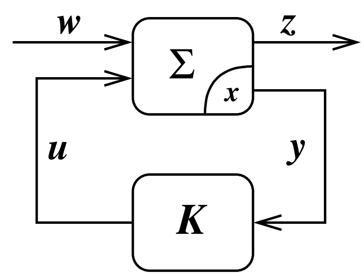

Fig. 1. The feedback configuration

of sensing and control over spatially distributed dynamic systems via algebraic graph theory.

A work that is particularly relevant to the present paper is that of Smith and Hadaegh [17] where the problem of identifying equivalent information topologies for formation control was considered.

A. Notation

A graph $G = (V, E)$ consists of a vertex set $V(G)$ and an edge set $E(G)$, whose elements (i.e., edges) connect pairs of vertices, making them adjacent to each other. The graphs that will be of interest to us will be simple; as such, multiple edges connecting the same pair of vertices and those starting and ending at the same vertex (i.e., loops) will not be allowed. Graphs that consist of edges with an "orientation," identifying their beginning (tails) and ending (heads), will be called directed graphs. A complete graph on n vertices is the graph that has all the potential

$$\binom{n}{2} := \frac{n!}{2!\,(n-2)!} \tag{1}$$

edges; we write $\binom{n}{m}$ when "2" in (1) is replaced by another nonnegative integer "m" not greater than n. If G_i is a subgraph of G_j with $V(G_i) \subseteq V(G_j)$ and $E(G_i) \subseteq E(G_j)$, then $G_{j/i}$ is a graph obtained by removing the edges of G_i from those of Gj. Recall that a walk in a graph is an alternating sequence of vertices and edges with the property that the consecutive vertices are the end-vertices of the edges between them. A walk that touches each vertex once is called a path. A connected graph is a graph where there is a path between every pair of distinct vertices. A connected graph that has the minimal number of edges is called a tree. Hence, if any edge of a tree is removed the resulting graph becomes disconnected. It is intuitive to realize that a tree can not contain a cycle- a subgraph where

0-7803-9098-9/05/$25.00 ©2005 AACC

every vertex has exactly two neighbors. A spanning tree of a graph G is a tree on $V(G)$. The set $\{1, 2, \ldots, n\}$ will be denoted by $[n]$. The cardinality of a finite set will be denoted as in $|\,[n]\,| = n$. The matrix I is used for the identity matrix of appropriate dimensions; $\mathbf{1}$ is the vector with all entries equal to one and **span** $\{x\}$ for the vector x denotes the span of the vector x. For a subset X of an inner product space, $X^{\perp}$ denotes the subspace whose elements are orthogonal to those of X. We denote the direct sum of two subspaces $\mathcal{A}$ and $\mathcal{B}$ by $\mathcal{A} \oplus \mathcal{B}$. $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote, respectively, the range and the null spaces of matrix A. Finally, the composition of operators will be designated as in $f \circ g$, i.e., for all x, $(f \circ g)(x) = f(g(x))$. In this paper, we use the term "information geometry" or "information topology" when the underlying information graph contains at least one spanning tree.

The organization of the paper is as follows. We formally introduce the main problem considered in the paper in §II. The notions of cut and cycle spaces of a graph are introduced in §II.B. §III characterizes the sought transformations among the equivalent sensing or control topologies. Some of the ramifications of these characterizations are also explored in §III. Simulation results conclude our presentation in §IV.

II. PROBLEM SETUP

We consider a distributed dynamic system that has collectively been represented as

$$\Sigma: \quad \dot{x}(t) = f(x(t), u(t), w(t)) \tag{2}$$
$$y(t) = Cx(t) \tag{3}$$
$$z(t) = g(x(t), w(t), u(t)), \tag{4}$$

where as in §I, x represents the state of the system Σ, y is the information vector (measured or communicated) available to the controller, and z is the set of variables that are to be controlled. The dynamic system is connected to the controller in the feedback configuration as shown in Figure 1. In this paper we will assume the absence of noise in measurements available to the controller. The main assumption that we make at this early stage is that the information geometry represented by matrix C in (3) is associated with a *relative* state information structure. Therefore the vector y is juxtaposition of vectors of the form

$$x_{ij}(t) := x_j(t) - x_i(t)$$

for some distinct indices $i, j \in [n]$; we note that $x_{ji} = -x_{ij}$. This information geometry can naturally be represented in terms of a directed graph. For example, the graph in Figure 2 corresponds to the situation where the information vector is

$$y(t) = [\, x_{12}(t)\; x_{13}(t)\; x_{14}(t)\; x_{23}(t)\; x_{24}(t)\; x_{34}(t)\,]^T$$

that is available to the controller. Let us assume that a control law has been designed for a particular information geometry represented by oriented graph G_i in order to satisfy a given stability or performance criteria (e.g., H_2/H_∞). Denote this control law by K_i. Note that we are not making any assumption on the linearity of plant nor of the controller at this stage. Now consider a scenario where the information geometry represented by G_i is changed to one that is represented by G_j. One of the main objectives of this paper is the parametrization of the transformation T_{ij} such that

$$K_j = K_i \circ T_{ij};$$

see Figures 3 and 4.

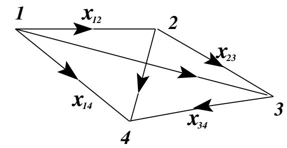

Fig. 2. Formation on $n = 4$ nodes

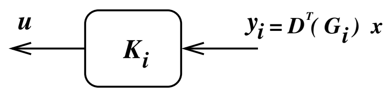

Fig. 3. Controller K_i is designed for information geometry G_i

A. Incidence Matrix

For the oriented information graph G, the incident matrix $D(G)$ is defined as the $|V(G)| \times |E(G)|$ such that:

- $[D(G)]_{k,l} = 1$ if v_k is the head of e_l
- $[D(G)]_{k,l} = -1$ if v_k is the tail of e_l
- $[D(G)]_{k,l} = 0$ if edge e_l is not incident on vertex v_k .

The incidence matrix proves to be a convenient way to represent the information geometry as

$$y_G(t) = (D(G)^T \otimes I_{n_i})\, x(t), \tag{5}$$

where $D(G)$ is the incidence matrix associated with a given oriented information graph on n dynamic elements with $x_i \in \mathbf{R}^{n_i}$, I_{n_i} is the $n_i \times n_i$ identity matrix, and "$\otimes$" denotes the Kronecker product [8]. To simplify our notation, we will use $D(G)$ to denote both the incidence matrix as well as its inflated version $D(G) \otimes I$. For example, the incidence matrix for Figure 2 is

$$D(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccccc} e_{1,2} & e_{1,3} & e_{1,4} & e_{2,3} & e_{2,4} & e_{3,4} \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

Consider now two arbitrary information geometries G_i and G_j that are related by T_{ij} via $y_i = T_{ij} y_j$. This implies that

$$D_i^T x(t) = T_{ij}\, D_j^T x(t)$$

for all x; we will adopt the convention of denoting $D(G_i)$ as D_i. Thus the desired transformation T_{ij} satisfies the matrix equation

$$T_{ij}\, D_j^T = D_i^T. \tag{6}$$

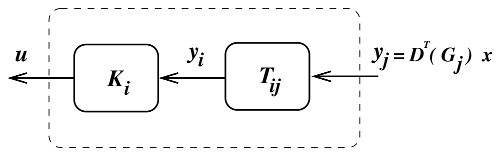

Fig. 4. Transform K_i for use with information geometry G_j

The existence and characterization of solutions to (6) is addressed in §III.

B. Cut and cycles spaces of a graph

Let us denote by $\mathcal{T}(G)$ and $\mathcal{C}(G)$ the cut and cycle spaces of graph G. These subspaces can be defined via the incidence matrix as follows:

$$\mathcal{T}(G) := \mathcal{R}(D^T), \quad \mathcal{C}(G) := \mathcal{N}(D),$$
$$\mathcal{T}(G)^{\perp} = \mathcal{C}(G), \quad \text{and} \quad \mathcal{C}(G)^{\perp} = \mathcal{T}(G).$$

Moreover, $\mathcal{C}(G) \oplus \mathcal{T}(G) = \mathbf{R}^{|E_G|}$. The cycle space is also referred to as the flow space. For a connected graph, the rank of the incidence matrix is $n-1$. Hence, the dimension of the cutspace is $n-1$. Analogously, the dimension of the cycle space is $m-(n-1)$, where m is the number of edges in G. Each row of $D(G)$ is called a cut of the graph G.

III. T-TRANSFORMATIONS

In this section we characterize the transformation T_{ij} shown in Figure 4. We proceed by considering the following two cases in sequence: (a) the initial graph is a spanning tree and the final graph is any connected graph, (b) the initial and the final graphs are two arbitrary connected graphs.

A. From a spanning tree to any connected graph

Recall that the transformation T_{ji} satisfies

$$T_{ji} D_i^T = D_j^T,$$

where in this section, D_i corresponds to a spanning tree on n nodes. The target graph D_j, on the other hand, represents a different information topology on these same nodes with m edges. A moment reflection on this transformation reveals that it essentially uses $\binom{n-1}{2}$ linearly independent cycles to rewrite the remaining unknown relative states; this is shown by an example in Figure 5 for a four nodes case. The sought matrix T_{ji}, transforming a measurement topology associated with a spanning tree to another connected topology, is characterized by the following proposition.

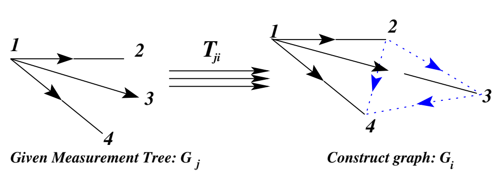

Fig. 5. Transforming a spanning tree to the complete graph

Proposition 3.1:

$$T_{ji} = \left\{ \left[(D_i^T D_i)^{-1} D_i^T \right] D_j \right\}^T. \tag{7}$$

The relation (7) is obtained by taking the appropriate pseudo-inverses in solving the matrix equation

$$D_i T_{ji}^T = D_j. \tag{8}$$

B. Existence of transformation T_{ji}

To show that such a transformation T_{ji} exists and is correctly characterized by proposition (7), it suffices to show that the following two properties hold: (1) $D_i^T D_i$ is positive definite. This holds since the graph G_i is assumed to be a tree, **rank** $D_i = n-1$, and **size** $D_i^T D_i = (n-1) \times (n-1)$. (2) $D_j \in \mathcal{R}(D_i)$; let us provide both a linear algebraic as well as a graphical justification for this relation. Since both graphs G_i and G_j are connected, **rank** $D_i^T =$ **rank** $D_j^T = (n-1)$, **size** $D_i^T = (n-1) \times n$, and **size** $D_j^T = m \times n$. This implies that **dim** $\mathcal{N}(D_i^T) =$ **dim** $\mathcal{N}(D_j^T) = 1$. In the meantime, for any connected graph one has $\mathbf{1} \in \mathcal{N}(D^T)$. Thus each column of D_j is an element of the range space of D_i. Figure 5 illustrates this algebraic proof via a graphical construction. Given any spanning tree G_i on *n* nodes, the graph G_j is obtained by completing the subsequent cycles.

Remark 3.2: As a consequence of above result it is natural to identify a spanning tree with a *basis* for the set of information graphs on *n* nodes.

C. From a connected graph to any other connected graph

Consider now the general transformation T_{ji} satisfying

$$T_{ji} D_i^T = D_j^T,$$

where D_i and D_j are incidence matrices corresponding to arbitrary information graphs on n nodes. Note that the justification for Proposition 3.1 is no longer valid in this case as the matrix product $D_i^T D_i$, although positive semidefinite, is not necessary positive definite.

Theorem 3.3: Any $n-1$ cuts of a connected graph are linearly independent and span the cut space.

Proof: Let D_i be the incidence matrix associated with G_i. Denote by v_i the cut at vertex i, i.e., v_k is the k-th row of D_i. As it was pointed out in §III.B,

$$D_i^T \mathbf{1} = 0 \quad \text{and thus} \quad v_n = -v_1 - v_2 - \cdots - v_{n-1}.$$

Pick arbitrary $n-1$ cuts of G_i (i.e., rows of D_i), say $\{v_1, \ldots, v_{n-1}\}$. We show that whenever $\alpha_1 v_1 + \cdots + \alpha_{n-1} v_{n-1} = 0$, one can conclude that $\alpha_1 = \cdots = \alpha_{n-1} = 0$. Note that the cut v_k at vertex k assigns $\{+1, -1\}$ to incoming/outgoing edges. Since G_i is connected, for some $k \in \{1, \ldots, n-1\}$, there exists an edge between vertex k and n. Moreover, there exists a cut v_k, $k \in \{1, \ldots, n-1\}$, such that for some $l \in 1, \ldots, m$, $|v_{kl}| = 1$, where $m = |E(G_i)|$. In addition, for all $k = 1, \ldots, n-1$, $v_{kl} = 0$. This is true since the l-th column of D_i defines an edge connecting vertex k and n. Thereby among the indices appearing in

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_{n-1} v_{n-1} = 0,$$

there exists v_k such that $\alpha_k = \pm\alpha_k$. This implies that $\alpha_k = 0$. Applying the above procedure recursively results in the conclusion that $\alpha_1 = \ldots = \alpha_{n-1} = 0$. ■

Proposition 3.4:

$$T_{ji} = \left\{ \left[\hat{D}_i^T (\hat{D}_i \hat{D}_i^T)^{-1} \right] \hat{D}_j \right\}^T, \tag{9}$$

where $\hat{D}_i$ and $\hat{D}_j$ correspond to any $n-1$ linearly independent cuts.

Proof: *Existence*: As shown in previous section $\mathcal{R}(D_i) = \mathcal{R}(D_j)$ spans the cut space. By Proposition 3.4, any $n-1$ cuts associated with D_i and D_j also span the cut space.

Correctness: The transformation T_{ji} that satisfies the matrix equation $T_{ji}\hat{D}_i^T = \hat{D}_j^T$ also satisfies $T_{ji}D_i^T = D_j^T$. This is due to the structure of the incidence matrix. Define the $n-1$ cuts for each graph as follows:

$$D_i^T = [v_1, v_2, \ldots, v_{n-1}, v_n],$$
$$D_j^T = [u_1, u_2, \ldots, u_{n-1}, u_n].$$

Given that $T_{ji}\hat{D}_i^T = \hat{D}_j^T$, and from the incidence relation, $v_n = -v_1 - \cdots - v_{n-1}$ and $u_n = -u_1 - \cdots - u_{n-1}$. Thus

$$T_{ji}v_n = T_{ji}[-v_1 - \cdots - v_{n-1}] = -u_1 - \cdots - u_{n-1} = u_n,$$

as required. ■

D. Controller projection

Now consider the situation where the controller is linear as in Figure 1, i.e.,

$$u(t) = K_i D_i^T x(t),$$

and K_i has been designed for the information graph G_i. In order to keep the same control input at all times, even when the information topology changes to G_j, we can use the transformations of the previous sections to write

$$u(t) = K_i\, T_{ij}\, D_j^T x;$$

thus $K_j := K_i T_{ij}$. Viewing T_{ij} as *left transformation* on the controller K_i, we arrive at a equivalent way of representing our result in terms of a transformation for the *controller reconfiguration*. Hence $K_i \rightsquigarrow K_j$ via the transformation T_{ij}. Figure 6 illustrates the T_{ij}-transformations in action. These transformations can be applied to either the relative information graph incident matrix D_j (from the left-hand side) or to the controller K_i (from the right-hand side). The transformation T_{ij} effectively captures the transformation on the controller spaces making them *robust* with respect to variations in the information topology.

E. Robustness

Assume that the linear control law $K_i(s)$ has been designed for the relative sensing geometry G_i for a linear system

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{10}$$
$$y(t) = D_i^T x(t). \tag{11}$$

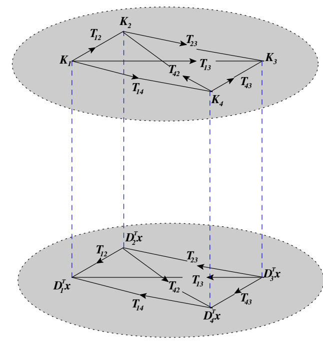

Fig. 6. T-transformation in action

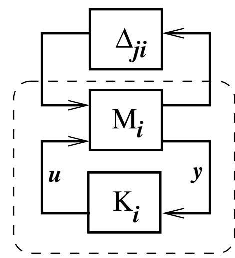

Fig. 7. Controller robustness for uncertain sensing geometries

Denote by $P_i(s)$ the transfer matrix from u to y, assuming the form $P_i(s) := D_i^T(sI - A)^{-1}B$. Now let G_j denote an uncertain relative sensing graph having the same number of edges as G_i. Furthermore, let $\Delta_{ji} = I - T_{ji}$ where $T_{ji}D_j^T = D_i^T$; T_{ji} is the corresponding T-transformation between G_i and G_j.

Theorem 3.5: The linear control law $K_i(s)$ robustly stabilizes the system (10) for an uncertain sensing graph G_j as long as

$$\|\Delta_{ji}\| < \frac{1}{\|\underline{S}(M_i, K_i)\|}$$

where

$$M_i(s) := \begin{bmatrix} 0 & P_i(s) \\ I & P_i(s) \end{bmatrix},$$

$\underline{S}(M, K)$ denotes the lower linear fractional transformation of $M(s)$ and $K(s)$, and the norm for a transfer matrix is its maximum singular value across all frequencies (i.e., its H_∞ norm).

Proof: This follows from the small gain theorem [5]. See Figure 7. ■

F. Controller transformation at each node

Recall that the control input to the distributed system has the form $u(t) = Kz(t)$. Denote each row of the K matrix by a bracketed superscript. Thus the i-th row of K, $K^{(i)}$, defines the control input $u_i(t) = K^{(i)}z(t)$ for node i. Figure 8 shows a block diagram, including the controller reconfiguration capability on each node in the absence of an external reference signal.

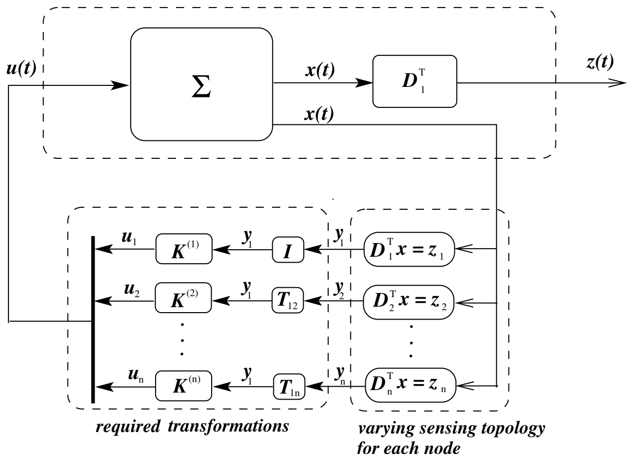

Fig. 8. Controller implementation on each node.

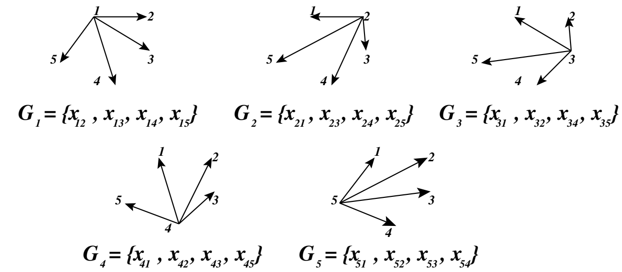

Fig. 9. Sensed graphs $G_1, \ldots, G_5$ in the simulation example

G. Controllability and spanning trees

We next consider the special case where each node of our spatially distributed system is described by a point mass model. Thus we have $u_i(t) = f_i(t)/m_i$, where m_i is the mass of each node. Denote by x_i the inertial position of node i and let $z(t) = D^T x(t)$. In this case, a reduced system model describing the dynamic evolution of the relative state z assumes the form

$$\ddot{z}(t) = D^T \ddot{x}(t) = D^T u(t). \tag{12}$$

We note that the system (12) is not controllable unless D^T corresponds to a spanning tree.

H. Minimal realization and controller implementation

When the objective of the feedback system is assumed to be controlling the relative states $\{z(t), \dot{z}(t)\}$ in (12), a controller can be designed based on a variety of methods. However as most control synthesis techniques require that the underlying system is stabilizable, we note that the relative state equation should correspond to a spanning tree.

Theorem 3.6: Suppose that Σ is a dynamical system that describes the time evolution of relative states in a network of n elements. Then the minimal system associated with Σ corresponds to a relative state geometry that is defined by a spanning tree.

IV. Simulation results

Let K_1 denote the optimal LQ control gain assuming an information tree G_1 in Figure 9, i.e., $u(t) = K_1 D_1^T x(t)$. In other words, the controller on each node expects as input the sensing graph given by G_1. In this simulation two systems are propagated. For system 1, the information tree is what the controller is designed for. For system 2, each node senses the relative state with respect to its neighbors. Thus the sensing graph for each node j is G_j for $j = 1, \ldots, 5$; see Figure 9. The control for system 2 with varying sensing geometries is labeled as $\tilde{u}$.

A. Case 1:

Consider a scenario where the transformation T_{1j} is not applied for nodes 2, 3, 4, and 5, to make G_j consistent with the sensing graph G_1 for which the controller had been designed. Figure 10 (a) depicts the control u applied by each node in this case. Moreover, Figure 10 (b) shows the time history of the difference $\|\tilde{u}(t) - u(t)\|$ in the controls input for the two systems. Note that since the second system has a varying sensing geometry and the transformation T_{1j} is not applied to each node, the controls time history for the two systems are different. It is interesting to note that even though the required T_{1j} is not applied, the control $\tilde{u}(t)$ is nevertheless stabilizing. The control performance can also be compared in the inertial motion of each node. Figure 10 (a) shows that both systems achieve the desired relative states. However, since the applied control forces are different, the inertial states of the two systems have a difference trajectory; see Figure 11. Figure 11 depicts the motion of the nodes in the xy-plane. In this figure, the asterisk represents the initial condition and the circle represents the final state.

B. Case 2:

For case 2, the control input for both systems is the same and the appropriate T_{1j} transformations are applied for nodes 2, 3, 4 and 5. Hence, the control for each node, given different information graphs, is equivalent to the control for the system with G_1 as its information graph. Figure 13 (b) shows the error time history $\|\tilde{u}(t) - u(t)\|$ where $\tilde{u}(t) = K_1 T_{1j} D_j^T x(t)$ and $u(t) = K_1 D_1^T x(t)$. Similarly, Figure 14 shows the inertial positions of all five nodes. As expected, the trajectories of the nodes are the same for the case when $G_1(t) \neq G_j(t)$, since the appropriate transformation are correctly applied. These transformations make the varying sensing geometry consistent with the geometry for which the controller was designed for. Equivalently, these transformations effectively *reconfigure* the corresponding controller, making it consistent with the updated information geometry.

References

[1] B. Bamieh, F. Paganini, and M. Dahleh. Distributed control of spatially-invariant systems, *IEEE Transactions on Automatic Control*, 47 (7): 1091-1107, July 2002.

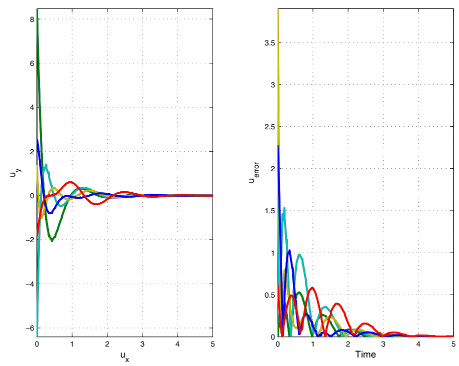

Fig. 10. (a) Control input $u(t)$ on each node (b) $\|\tilde{u}(t) - u(t)\|$

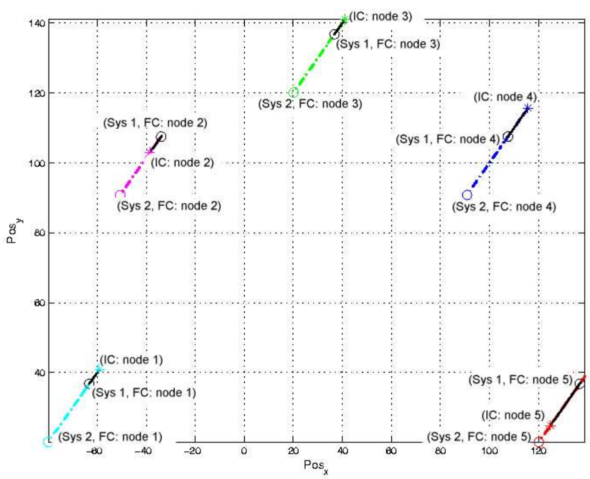

Fig. 11. Inertial translational motion of each node

[2] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[3] R. Diestel. *Graph Theory*, Springer, 2000.

[4] R. D'Andrea and G. E. Dullerud, Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, (48) 9: 1478-1495, 2003.

[5] G. Dullerud and F. Paganini. *A Course in Robust Control Theory*, Springer, 2000.

[6] A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations, *IEEE Transactions on Automatic Control*, (49) 9: 1465- 1476, 2004.

[7] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer, 2001.

[8] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*, Cambridge, 1991.

[9] A. Jadbabaie, J. Lin and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control*, (48) 6: 988–1001, 2003.

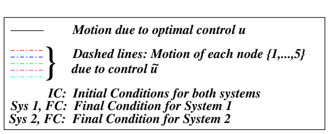

Fig. 12. Key for Figure 11 and 14

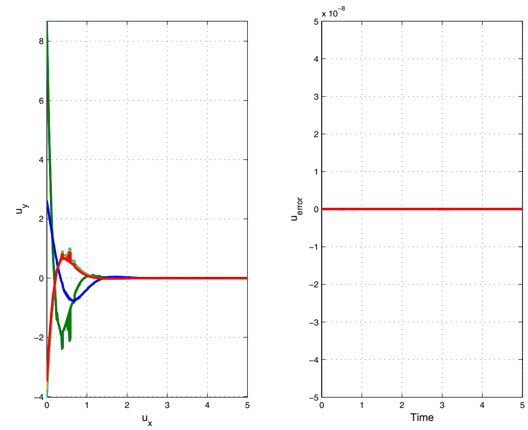

Fig. 13. (a) Control input $u(t)$ on each node (b) $\|\tilde{u} - u\|$

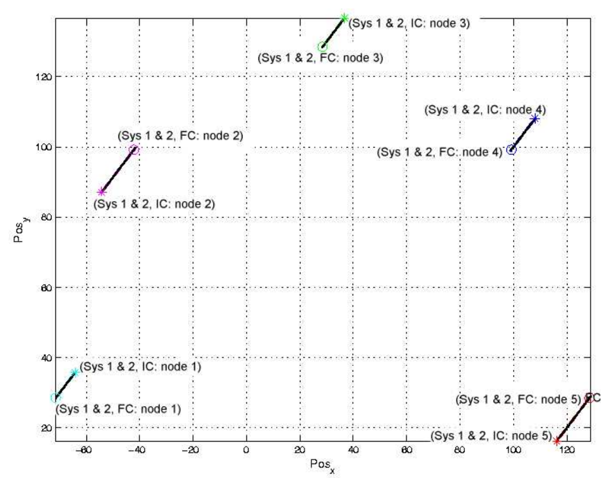

Fig. 14. Inertial translational motion of each node

[10] W. Kang, A. Sparks, and S. Banda. Coordinated control of multiscale systems, *AIAA Journal of Guidance, Control, and Dynamics*, 24 (2): 360–368, 2001.

[11] U-L. Ly. *Lecture Notes on Linear Systems*. Department of Aeronautics and Astronautics, University of Washington, Autumn 2003.

[12] M. Mesbahi and F. Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching, *AIAA Journal of Guidance, Control, and Dynamics*, (24) 2: 369–377, 2001.

[13] M. Mesbahi. State-dependent dynamic graphs and their controllability properties, *IEEE Transactions on Automatic Control* (50) 3: 387-392, 2005.

[14] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control*, (49) 9: 1520- 1533, 2004.

[15] P. K. C. Wang and F. Y. Hadaegh. Coordination and control of multiple microspacecraft moving in formation, *Journal of the Astronautical Sciences*, 44: 315–355, 1996.

[16] *http://planetquest.jpl.nasa.gov/TPF/*

[17] R. S. Smith and F. Y. Hadaegh. Control topologies for deep space formation flying spacecraft, *American Control Conference*, 2002.