Copyright F.L. Lewis
Updated: October 15, 2007

Control, Decision, and Trust Graphs

Control, Decision, and Trust Graphs 1
Control Graphs 2
Linear Integrator Dynamics 3
Switched Graphs and Changing Topology 3
Vicsek Models 4
More General Node Dynamics 4
Nonlinear Coupling 6
Potential Fields 6
Nonholonomic Vehicles 8
Nonlinear Formation Control Law 8
Kuramoto Oscillators. 9
Decision Graphs 9
Distributed Kalman Filtering on Graphs 9
Distributed Bayesian Belief Propagation on Networks 10
Distributed Dempster-Shafer Belief Propagation on Networks 11
Matrix Computations for Dempster-Shafer Theory 13
Trust Graphs: Propagation and Maintenance of Trust 14
Cooperative Control Based on Trust 16
References 18

Control, Decision, and Trust Graphs

Consider a graph *G= (V,E)* with *V* a set of *N* agents or nodes $V = \{v_1, \cdots, v_N\}$ and *E* the set of edges. Assume $(v_i, v_i) \notin E, \forall i$, i.e. no self loops. Edge $e_{ij} = (v_i, v_j)$ is represented as an arrow with tail at v_i and head at *vj*. Edge $e_{ij} = (v_i, v_j)$ is said to be outgoing with respect to node v_i and incoming wrt v_j. Node v_i is termed the parent and v_j the child. The in-degree of v_i is the number of edges having v_i as a head. The out-degree of a node v_i is the number of edges having v_i as a tail. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with edges incoming to v_i. The number of neighbors $|N_i|$ of node *i* is equal to its in-degree. If $(v_i, v_j) \in E \Rightarrow (v_j, v_i) \in E, \forall i, j$ the graph is said to be undirected, otherwise it is directed, termed a digraph. If the in-degree equals the out-degree for all nodes $v \in V$ the graph is said to be balanced.

Associate with each edge $v_i, v_j \in E$ a strictly positive weight w_{ji}. Define the (weighted) adjacency or connectivity matrix $W = [w_{ij}]$. Define the weighted degree matrix as a diagonal matrix $D = [d_{ij}]$ with diagonal element $d_{ii} = \sum_j w_{ij}$, the weighted in-degree. Define the weighted graph Laplacian matrix as $L = D - W$. Note that L has row sums equal to zero.

The communication cost $C(G)$ of digraph $G=(V,E)$ is defined as the total number of directed edges $C = |E|$. For the weighted graph $C = \sum_{i,j} \text{sgn}(w_{ij}) = \sum_i |N_i|$. The graph algebraic connectivity is defined as $\lambda_2(L)$ is the Fiedler eigenvalue of L, which is a measure of interconnection or communication complexity.

Control Graph. Given a set of N agent nodes $\{v_i\} \in V$, define three undirected graphs as follows. Define a control graph $G^c = (V, E^c)$, where edge $(v_i, v_j) \in E^c$ if node v_j can use information from v_i for control purposes. Define the control neighborhood of node v_i as $N^c{}_i = \{v_j : (v_j, v_i) \in E^c\}$, i.e. the set of nodes with edges incoming to v_i. Define the control weight matrix W^c and the control graph Laplacian L^c.

Decision Graph. Define a decision graph $G^d = (V, E^d)$, where edge $(v_i, v_j) \in E^d$ if node v_j can use information from v_i for decision or estimation purposes. Define the decision neighborhood of node v_i as $N^d{}_i = \{v_j : (v_j, v_i) \in E^d\}$, i.e. the set of nodes with edges incoming to v_i. Define the decision weight matrix W^d and the decision graph Laplacian L^d.

Trust Graph. Define a trust graph $G^t = (V, E^t)$, where edge $(v_i, v_j) \in E^t$ if node v_j obtains a direct trust evaluation about node v_i. Define the direct trust neighborhood of node v_i as $N^t{}_i = \{v_j : (v_j, v_i) \in E^t\}$, i.e. the set of nodes with edges incoming to v_i. Define the trust weight matrix W^t and the trust graph Laplacian L^t.

Traditional work in networked control systems has assumed that $G_c = G_d = G_t$, i.e. that the control links provide exact reliable information with no noise, and that each node bases its control, decisions, and trust on the same set of neighbors.

Control Graphs

Given a control graph $G^c = (V, E^c)$ let agent i have a state that evolves according to the dynamics $\dot{x}_i = f_i(x_i, u_i)$. The control given by $u_i = k_i(x_{i_1}, x_{i_2}, \cdots, x_{i_{m_i}})$ is said to be a protocol with topology G if $v_{i_j} \in v_i \cup N_i,\ j \in \{1, m_i\}$. The protocol is called distributed if $m_i < N, \forall i$. An important problem is the consensus problem. The nodes achieve consensus asymptotically if, for all initial conditions $x_i(0)$, $\lim_{t\to\infty} \|x_i(t) - x_j(t)\| = 0, \forall i, j$. Then, if the equilibrium is a constant value, the states all go to the SAME steady-state value. The importance is that the nodes achieve a global consensus result using only local protocol controls, i.e. local information.

Linear Integrator Dynamics

Much work has been done for the simplified scalar case with linear integrator control dynamics $\dot{x}_i = u_i$. In this case, properties of the networked system generally depend *only on the graph structure,* as captured in the graph Laplacian matrix. Select the gradient-based control protocol $u_i = \sum_{j \in N_i} w_{ij}(x_j - x_i)$. Then the closed-loop system is

$$\dot{x}_i = \sum_{j \in N_i} w_{ij}(x_j - x_i)$$

or $\dot{x}_i = -\sum_{j \in N_i} w_{ij} x_i + \sum_{j \in N_i} w_{ij} x_j = -d_{ii} x_i + \sum_{j \in N_i} w_{ij} x_j$ so that $\dot{x} = -(D^c - W^c)x = -L^c x$ with L^c the control graph Laplacian. Note also that $u = -L^c x$.

For linear integrator dynamics the stability and consensus properties depend only on the graph connectivity as reflected in the Laplacian L^c. The zero eigenvalue of L^c corresponds to the unobservability of the absolute motion (center of gravity) of the formation. The relative formation dynamics is said to be stable if the interconnected system is asymptotically stable for all eigenvalues of L^c other than the zero eigenvalue.

For an undirected control graph G^c (e.g. flocks and swarms), one has that consensus is reached iff G^c is (strongly) connected. For digraphs [Saber and Murray 2004], consensus is reached if G^c is strongly connected. Then, the consensus value is given by $\chi(x(0)) = \dfrac{\sum_i \gamma_i x_i(0)}{\sum_i \gamma_i}$, where the left eigenvector of L^c for $\lambda = 0$ is $w_l = [\gamma_i]$. The average consensus problem $\chi(x(0)) = \frac{1}{N}\sum_i x_i(0)$ is solved iff the strongly connected control graph is balanced, for then $\gamma_i = 1, \forall i$. The case of strongly connected and balanced control digraphs covers flocks and swarms, but does not cover formations, which generally have a tree structure that is not balanced. It has been shown [Ren and Beard 2005] that the given control law achieves consensus asymptotically iff the control digraph has a spanning tree, a milder condition than balanced strong connectivity that does include formations. A graph has a spanning tree iff it has a node connected to all other nodes, e.g. a leader.

For a dynamic graph with discrete-time state $x_i(k+1) = x_i(k) + \varepsilon u_i(k)$, the gradient-based control above gives the closed-loop system

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} w_{ij} \left(x_j(k) - x_i(k) \right)$$

or $x(k+1) = P_\varepsilon x(k)$ with $P_\varepsilon = I - \varepsilon L$ the Perron matrix, which is a row stochastic matrix. Similar results have been shown.

Switched Graphs and Changing Topology

The graph connectivity or topology may change over time. Let $\bar{G} = \{G_1, G_2, \cdots, G_M\}$ be the set of all control digraphs defined for a given set of nodes V. Let the Laplacian for control graph G_i be denoted L_i. The union of any group of digraphs contained in $\bar{G}$ is a digraph with nodes V and

edge set given by the union of the edge sets of that group. A group of graphs is said to be jointly (strongly) connected if their union is (strongly) connected [Jadbabaie, Lin, Morse 2003].

With switching, the gradient-based control is $u(t) = -L(t_k)x(t), t \in [t_k, t_{k+1})$ *and* $\dot{x}(t) = -L(t_k)x(t), t \in [t_k, t_{k+1})$ with t_k the switching times. [Saber and Murray 2004] have shown that consensus is achieved asymptotically if there exists an infinite sequence of consecutive bounded nonoverlapping time intervals with the property that the union of digraphs over each interval is strongly connected. Then, if each union is also balanced, average consensus is solved.

This does not cover the formation case. [Ren and Beard 2005] showed that consensus is achieved asymptotically under the milder condition of existence of an infinite sequence of consecutive bounded nonoverlapping time intervals with the property that the union of digraphs over each interval has a spanning tree. Furthermore, if the union of graphs after some time does not have a spanning tree, consensus cannot be achieved asymptotically.

Vicsek Models

[Jadbabaie, Lin, Morse 2003] consider the Vicsek [1995] model of swarm heading alignment for the case of undirected graphs. Let $x(t+1) = x(t) + u(t)$ and $u(t) = -(I + D^c(t))^{-1} L^c(t)x(t)$, with the average heading error given by $e(t) = L(t)x(t)$. Then the closed-loop system is $x_i(t+1) = \frac{1}{1+n_i(t)}\left(x_i(t) + \sum_{j \in N_i(t)} x_j(t)\right)$, where $n_i(t) = |N_i(t)|$. Note that $x(t+1) = (I + D^c(t))^{-1}(I + W^c(t))x(t)$, where all nonzero weights w_{ij} are equal to one. This closed-loop matrix is a stochastic matrix.

It was shown that if the switched system has an infinite sequence of contiguous, nonempty, bounded time intervals with the property that, across each time interval the graphs are jointly connected. Then $\lim_{t \to \infty} x(t) = x_{ss}1$. where x_{ss} is a number depending only on *x(0)* and the switching signal, and **1** is a vector of all ones. That is, consensus is achieved.

It was shown that there may NOT EXIST a common Lyapunov function in such cases. Joint Lyapunov function is too strong a requirement. The analysis is done using the properties of the graph Laplacian.

The case of weighted digraphs was considered by Moreau [2005] who showed: Let the nonzero weights be bounded above and below. If there is a $T \geq 0$ such that for all $t_0 \in N$ there is a node connected to all other nodes across $[t_0, t_0 + T]$, then the N values $x_i(t)$ converge to a common value as $t \to \infty$.

More General Node Dynamics

If the nodes have more general dynamics, these dynamics must be taken into account along with the control graph structure. Then, depending on the dynamics, adding links can even destabilize the system.

[Lee and Spong 2007] considered inertial dynamics $m_i \ddot{x}_i = u_i$ with control $u_i = \sum_{j \in N_i} bw_{ij}(\dot{x}_j - \dot{x}_i) + kw_{ij}(x_j - x_i)$ which yields closed-loop Lagrangian (robot) dynamics

$m_i\ddot{x}_i + \sum_{j\in N_i} bw_{ij}(\dot{x}_j - \dot{x}_i) + kw_{ij}(x_j - x_i)$ or $M\ddot{x} + bL\dot{x} + kLx = 0$ with $M = diag\{m_i\}$ and L the control graph Laplacian. If the control graph is directed, *L* can be nonsymmetric. The system was decomposed into a locked (c.g.) system and a shape (relative) system. It was shown that these two systems are decoupled iff the graph is balanced. The centroid velocity is invariant if the graph is balanced. If the graph is balanced and strongly connected, and the damping gains *b* are selected large wrt the spring const gains *k*, then the agents velocities converge to the centroid velocity, and the positions converge to the desired shape. If all the inertias m_i are the same, the centroid velocity is the average velocity. The same results hold for switched topologies if the graphs are all balanced.

[Fax and Murray 2004] considered identical node (vehicle) dynamics $\dot{x}_i = Ax_i + Bu_i$, $x_i \in R^n$ with feedback $u_i = \frac{1}{n_i}\sum_{j\in N_i} K(x_i - x_j)$ and n_i the node in-degree. It was shown that if the control graph is strongly connected, then the only equilibrium point is the desired relative position of the vehicles.

If all the vehicle systems are single-input/single-output, it was shown that the formation is relatively stable iff the net encirclement of $-\lambda_i^{-1}$ by the Nyquist plot of *P(s)* is zero for all nonzero eigenvalues λ_i of L^c, with *P(s)* the closed-loop vehicle transfer functions.

Adding links can destabilize the formation if the new links make the graph more nearly *k*-periodic. An example where this may occur is the position/velocity dynamics with delay $\ddot{x}_i = e^{-s\tau}u_i$.

Chopra and Spong [2007] used Lyapunov methods to show that if the dynamics of each agent are passive, then again the stability properties depend only on the graph interconnections. Consider a dynamic graph with node dynamics $\dot{x}_i = f_i(x_i) + g_i(x_i)u_i$, $y_i = h_i(x_i)$ which are passive. The outputs are said to synchronize if $\lim_{t\to\infty}\left\|y_i(t) - y_j(t)\right\| = 0, \forall i, j$. Consider control $u_i = \sum_{j\in N_i} K(y_j - y_i)$. It was shown that if the control digraph is connected and balanced (i.e. strongly connected), then the system is globally stable and the outputs synchronize.

In the case of switched graphs, it was shown that if each control graph in the switching set is balanced, and there exists an infinite sequence of bounded, non-overlapping time intervals across which the graphs are jointly connected, then the system is globally stable and the outputs synchronize.

Time delays were handled as well. Consider constant bounded time delays. The outputs synchronize if $\lim_{t\to\infty}\left\|y_i(t - T_{ij}) - y_j(t)\right\| = 0, \forall i, j$, where T_{ij} is the sum of time delays along the path from v_i to v_j. Consider time-delayed control $u_i(t) = \sum_{j\in N_i} K(y_j(t - T_{ji}) - y_i)$. Let control graph be connected and balanced (i.e. strongly connected). Let there be a unique path between any two distinct nodes. Then the system is globally stable and the outputs synchronize.

Nonlinear Coupling

Chopra and Spong [2007] considered the nonlinear coupling control $u_i = \sum_{j \in N_i} \varphi(y_j - y_i)$ with $\varphi(.)$ a nonlinear function that is continuous, locally Lipschitz, and antisymmetric. He showed that, if the control graph is undirected and connected, then the system is globally stable and the trajectories converge to the largest set where the Lyapunov derivative is equal to zero. Also considered was the case of nonlinear coupling with time delay, using the scattering transformation.

One can steer the difference of the outputs to a desired set by selecting $\varphi(.)$ as the gradient of a potential function with selected local minima.

Potential Fields.

Gazi and Passino [2003, 2004] considered swarms of *N* agents with the integrator dynamics with control given by $\dot{x}_i = \sum_{j \neq i} g(x_i - x_j)$, $x_i \in R^n$. This amounts to a fully connected undirected graph. Function *g(.)* is the gradient of a potential function consisting of a repulsive part and an attractive part, with a unique value δ where $g(\delta) = 0$. The example used was $g(y) = -y\left(a - b\exp\left(-\frac{\|y\|^2}{c}\right)\right)$, which is equal to zero for $\delta = \sqrt{c\ln(b/a)}$, repulsive for $\|y\| < \delta$, and attractive for $\|y\| > \delta$.

The following results are shown. Define the center of the swarm as $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$. Then the center is stationary, $\dot{\bar{x}} = 0$.

Let an agent *i* satisfy $\|x_i - x_j\| > \delta, \forall j \neq i$, i.e. it experiences purely attractive forces. Then its motion is towards the center. Proof uses Lyapunov function $V_i = \frac{1}{2}(x_i - \bar{x})^T(x_i - \bar{x})$.

As time passes, the agents will converge to a hyperball $\|x - \bar{x}\| \leq \varepsilon$, with $\varepsilon = \frac{b}{a}\sqrt{\frac{c}{2}}\exp(-1/2)$. This is independent of the number of agents *N*, so density increases with *N*.

The motion converges with time to $\dot{x} = 0$, i.e. all agents come to relative rest. Proof is based on the Lyapunov potential field function $V(x) = \frac{1}{2}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}\left[a\|x_i - x_j\|^2 + bc\exp\left(-\frac{\|x_i - x_j\|^2}{c}\right)\right]$. Then the gradient gives the closed-loop system.

Define the local potential function $V_i(x) = \frac{1}{2}\sum_{\substack{j=1 \\ j \neq i}}^{M}\left[a\|x_i - x_j\|^2 + bc\exp\left(-\frac{\|x_i - x_j\|^2}{c}\right)\right]$. Then each agent minimizes its local potential function, and the global minimum is achieved.

If each agent has finite body size of radius η, then the swarm size is bounded below by $\varepsilon = \eta\sqrt[n]{N}$. Now the size depends on the number of agents. The density is now bounded above by $\rho \le 1/\pi\eta^2$.

Gazi [2005] extends the above integrator dynamics results to vehicles with Lagrangian dynamics $M_i\ddot{x}_i + f_i = u_i$ by defining the sliding manifold $s_i = \dot{x}_i + \nabla_{x_i} V(x) = 0$ with *V(x)* a potential function. On the sliding manifold one has $\dot{x}_i = -\nabla_{x_i} V(x)$ as above.

In [Leonard and Fiorelli 2001] potential field control of position/velocity dynamics $\dot{p}_i = v_i, \dot{v}_i = u_i$ is considered. Define a potential function $V(r_{ij})$ with r_{ij} the distance from node *j* to node *i*. Define the gradient $f_{ij}(r_{ij}) = \nabla_{r_{ij}} V(r_{ij})$. Control is taken as $u_i = -\sum_{j\neq i} \frac{f_{ij}}{|r_{ij}|} r_{ij} + f_{v_i}$ with the last term a damping, e.g. $f_{v_i} = -av_i$ or $f_{v_i} = -a\dot{r}_i$.

Define polar coordinates $r_i = \begin{bmatrix} r_i\cos\theta_i \\ r_i\sin\theta_i \end{bmatrix}$. Then the closed loop system (without damping) is

$$\ddot{r}_i = \dot{r}_i\dot{\theta}_i^2 - \sum_{j\neq i}\left(r_i - r_j\cos(\theta_i - \theta_j)\right)\frac{f_{ij}(r_{ij})}{|r_{ij}|}$$

$$r_i\ddot{\theta}_i = -2\dot{r}_i\dot{\theta}_i - \sum_{j\neq i} r_j\sin(\theta_i - \theta_j)\frac{f_{ij}(r_{ij})}{|r_{ij}|}$$

Compare the last equation to Kuramoto oscillators.

Flocking is considered. Also schooling, where $f_{v_i} = a(v_d - v_i)$ with v_d a desired velocity.

[Saber and Murray 2003] consider the dynamics $\dot{p}_i = v_i, \dot{v}_i = u_i$ and graph theory is used. A flock is defined as a weakly connected digraph, or a connected undirected graph. Define the c.g. as $\bar{p} = ave(p) = \sum_i p_i$ and the relative position as $\tilde{p}_i = p_i - \bar{p}$. Then the translational dynamics are $\dot{\bar{p}}_i = \bar{v}_i, \dot{\bar{v}}_i = \bar{u}_i$ and the relative dynamics are $\dot{\tilde{p}}_i = \tilde{v}_i, \dot{\tilde{v}}_i = \tilde{u}_i$. A separation principle is shown. (The zero eigenvalue of the Laplacian corresponds to the absolute motion of the c.g.)

The flocking protocol used is $u_i = \sum_{j\in N_i} \frac{f_{ij}\left(\|p_j - p_i\| - d_{ij}\right)}{\|p_j - p_i\|}(p_j - p_i) + c_d(v_j - v_i)$ or $u_i = \sum_{j\in N_i} s_{ij}(p_j - p_i) + c_d(v_j - v_i)$ with s_{ij} the edge stress weights, d_{ij} the desired separation, and c_d the damping. The force f_{ij} is the gradient of a potential function.

Dang and Lewis [2006] used potential field methods for dynamic localization in wireless sensor networks. They defined a Lyapunov function in terms of an 'error potential,' and took the

gradient-based control to update estimates in an observer for estimation of the positions of nodes in a network.

Nonholonomic Vehicles

In [Lin, Francis, Maggiore 2005] unicycle agents were considered

$$\dot{x}_i = v_i \cos\theta_i$$
$$\dot{y}_i = v_i \sin\theta_i$$
$$\dot{\theta}_i = \omega_i$$

or

$$\dot{\bar{z}}_i = v_i e^{j\theta_i}$$
$$\dot{\theta}_i = \omega_i$$

with $\bar{z}_i = x_i + jy_i$. The control is taken as

$$v_i = k \sum_{j \in N_i} (z_j - z_i) \bullet r_i$$
$$\omega_i = \cos t$$

in terms of the dot product, with $z_i = [x_i \; y_i]^T$ and r_i the Frenet-Serret vector tangent to the trajectory. Then the closed-loop dynamics is $\dot{z} = -kH(\theta(t))\left(L \otimes I_2\right) z$ with $H(\theta(t)) = diag\{r_i r_i^T\}$ and $\otimes$ the Kronecker matrix product.

It was shown that the formation stabilizes to a point (i.e. all vehicles converge to the same value of z_i) for all initial (x_i, y_i, θ_i) iff the graph has a reachable node. [They define graph edge arows 'backwards' so this corresponds to existence of a node connected to all other nodes, i.e. a spanning tree.]

It was shown that the formation stabilizes to a line iff there are at most two disjoint closed sets of nodes in the control digraph.

Nonlinear Formation Control Law.

In [Justh and Krishnaprasad 2002] are considered nonholonomic vehicle dynamics in kinematic Frenet-Serret form

$$\dot{r}_k = x_k$$
$$\dot{x}_k = y_k u_k$$
$$\dot{y}_k = -x_k u_k$$

with *r* the position and *x,y* orthogonal vectors specifying the orientation. The control law is

$$u_k = \sum_{j \neq k} \left[-\eta(|r_{jk}|) \left(\frac{r_{jk}}{|r_{jk}|} . x_k \right) \left(\frac{r_{jk}}{|r_{jk}|} . y_k \right) - f(|r_{jk}|) \left(\frac{r_{jk}}{|r_{jk}|} . y_k \right) + \mu(|r_{jk}|) x_j . y_k \right]$$

with $r_{kj} = r_j - r_k$ and using the dot product. It is shown using Lyapunov methods that the system converges to the equilibrium points.

In the control, the last term forces a common orientation. Using this term only yields $u_k = \mu \sum_{j \neq k}^{N} x_j . y_k$. Define $x_k = \exp(i\theta_k)$. Then $y_k = \left(i \exp(i\theta_k)\right)^*$, and this becomes $\dot{\theta}_k = u_k = \mu \sum_{j \neq k} \sin(\theta_j - \theta_k)$. This is the Kuramoto oscillator, and is the gradient system wrt the energy function $V_{orient} = -\frac{\mu}{2} \sum_{j,k} \cos(\theta_k - \theta_j)$.

The function *f(.)* allows one to achieve a desired separation, for it is shown that equilibrium points have $f(\rho) - \frac{1}{\rho} = 0$.

Kuramoto Oscillators.

N Kuramoto oscillators have dynamics $\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_j \sin(\theta_j - \theta_i)$ with oscillation frequency ω_i and coupling gain *K* [Strogatz 2000]. The oscillators are said to synchronize if $\dot{\theta}_i(t) - \dot{\theta}_j(t) \to 0 \; as \; t \to \infty \; \forall i, j$.

Kuramoto took the oscillation frequencies as distributed according to a probability density function. He showed that there is a coupling gain K_c below which the oscillators remain incoherent, i.e. do not synchronize. Above this gain the incoherent state becomes unstable, and the oscillators split into two groups- those with oscillation frequencies close to the mean synchronize to a mean frequency $\bar{\theta}$, while the others drift relative to this group.

In [Chopra and Spong CDC 2005], synchronization is studied using the Lyapunov function $V = \frac{1}{2} \dot{\theta}^T \dot{\theta}$, with $\theta = [\theta_1 \cdots \theta_N]^T$. It is shown that $\dot{V} = -\frac{K}{N} \sum_{i,j} \cos(\theta_i - \theta_j)(\dot{\theta}_i - \dot{\theta}_j)^2$. It is shown that above some coupling gain, and if the initial phase differences are in a certain compact set, the oscillators converge to the mean frequency $\bar{\theta} = \frac{1}{N} \sum_i \omega_i$.

Decision Graphs

The above work on control graphs assumes that full exact information is available to node *i* from its neighbors. Consider a decision graph $G^d = (V, E^d)$. Define the edge weights $w_{ij}(t)$ to be equal to 1 if there is a link (v_j, v_i) in $G^d = (V, E^d)$ at time *t*, and 0 if not.

Distributed Kalman Filtering on Graphs

If the channels are noisy, then the information from node *j* available to node *i* is $z_{ij} = x_j + v_{ij}$, with $v_{ij} \sim (0, R_{ij})$ the noise on edge (v_j, v_i). Define the unknown consensus value as x^*. Assume that $\dot{x}^* = w$, with $w \sim (0, Q)$ a process noise. Define error covariance

$P_i = E\{(x_i - x^*)(x_i - x^*)^T\}$. Then the estimator dynamics are given by [Ren, Beard, Kingston 2005]

$$\dot{x}_i(t) = \sum_{j \in N_i} w_{ij}(t) K_{ij}(t) \left(z_{ij}(t) - x_i(t) \right)$$

with Kalman gain given by

$$\dot{P}_i = -P_i \left[\sum_{j \in N_i} w_{ij}(t) (P_j + R_{ij})^{-1} \right] P_i + Q$$

$$K_{ij} = P_i (P_j + R_{ij})^{-1}$$

In the discrete case, with assumed consensus dynamics $x^*(k+1) = x^*(k) + w(k)$, one has the decision dynamics

$$x_i(k+1) = x_i(k) + \sum_{j \in N_i} w_{ij}(k) K_{ij}(k+1) \left(z_{ij}(k) - x_i(k) \right)$$

with Kalman gain given by

$$P_i(k+1) = \left[\left(P_i(k) + Q(k) \right)^{-1} + \sum_{j \in N_i} w_{ij}(k) \left(P_j(k) + R_{ij}(k) \right)^{-1} \right]^{-1}$$

$$K_{ij}(k+1) = P_i(k+1) \left(P_j(k) + R_{ij}(k) \right)^{-1}$$

In [Ren, Beard, Kingston 2005] it is shown that consensus is achieved asymptotically (i.e. the node Kalman filters converge to the same estimates) if there exist an infinite sequence of consecutive bounded nonoverlapping time intervals with the property that the union of the decision digraphs over each interval has a spanning tree.

Distributed Bayesian Belief Propagation on Networks

Given a set of *N* agent nodes $\{v_i\} \in V$, each of which has sensors to measure quantities $y_i \in Y$, with *Y* a vector space of sensor measurements. Define an acyclic decision graph $G^d = (V, E^d)$, where edge $(v_i, v_j) \in E^d$ if node v_i communicates with node v_j, i.e. v_j can use information from v_i for decision or estimation purposes. Define the decision neighborhood of node v_i as $N^d{}_i = \{v_j : (v_j, v_i) \in E^d\}$, i.e. the set of nodes with edges incoming to v_i. Edge $e_{ij} = (v_i, v_j)$ is represented as an arrow with tail at v_i and head at v_j. Denote the source and destination of the edge resp. as $s(e_{ij}) = v_i$, $d(e_{ij}) = v_j$.

Let the distribution of the measured vector $Y_v \in Y^v$ have the form

$$P(Y_v = y_v : v \in V\} \propto \prod_{v \in V} \varphi_v(y_v) \prod_{e \in E^d} \psi_e(y_{s(e)}, y_{d(e)})$$

This form arises, e.g. in Bayesian graphs having the Markov property where node potential $\varphi_v(.)$ is a prior distribution (associated with a node) and edge potential $\psi_e(y_{s(e)}, y_{d(e)}) \approx p(y_{d(e)} / y_{s(e)})$ is a conditional probability (associated with an edge).

It is desired to compute the marginal distributions at each node using only local computations. Let the *k*th message sent by node $v_j \in N_i^d$ to node v_i be the vector $m_k^{ji}(y)$. Construct the messages using the *sum-product message passing algorithm* [Pearl 1988]

$$m_0^{ji}(y) = 1$$
$$m_{k+1}^{ji}(y) = \sum_{y_j \in Y} \psi_{(v_j, v_i)}(y_j, y) \varphi_{v_j}(y_j) \prod_{\ell \in N_j - \{i\}} m_k^{\ell j}(y_j)$$

After receiving the *k*th message, each node constructs an estimate of its local marginal pdf using

$$\hat{p}_k^i(y) \propto \varphi_{v_i}(y) \prod_{j \in N_i} m_k^{ji}(y)$$

This algorithm converges to the local marginals if the graph is undirected and is a tree.

In the special case $\psi_{(v_j, v_i)}(y_j, y) = 1$, *iff* $y_j = y$, then one has

$$m_0^{ji}(y) = 1$$
$$m_{k+1}^{ji}(y) = \varphi_{v_j}(y) \prod_{\ell \in N_j - \{i\}} m_k^{\ell j}(y)$$

Take now logarithms and define

$$x_k^{ji}(y) = \log(m_k^{ji}(y)), \qquad u^i(y) = \log(\varphi_{v_i}(y))$$

Then

$$x_{k+1}^{ji}(y) = u^j(y) + \sum_{\ell \in N_j - \{i\}} x_k^{\ell j}(y_j)$$

This compares in an intriguing manner to the networked dynamical systems seen in the controls literature. However, here *the state is associated with each edge*, not with each node.

This special case was used in [Saligrama 2006] for distributed Bayes hypothesis testing. It was shown that it yields correct decisions in a digraph that is connected and acyclic. The correct unique MAP estimate is reached regardless of initial conditions if the graph is balanced.

Distributed Dempster-Shafer Belief Propagation on Networks

Let θ be a frame of discernment and $m(A)$, $A \subset \theta$ a basic probability assignment bpa. $A \subset \theta$ is a focal element of θ if $m(A)>0$, and the set of all focal elements is called the core. The belief of A is defined as

$$Bel(A) = \sum_{B \subset A} m(B)$$

and represents the degree to which event A is certain. The plausibility of A is defined as

$$Pl(A) = \sum_{B \cap A \neq 0} m(B)$$

and represents the degree to which event A may possibly have occurred.

Bayes probability theory is a special case of Dempster-Shafer theory when the bpas are assigned only to each singleton element of θ, and then $Bel(A) = Pl(A) = probability(A), \forall A \in \theta$. DS theory is more general than Bayes theory and can capture situations intractable for Bayes. In Bayes theory, one uses Bayes rule

$$P(A/B) = \frac{P(A,B)}{P(B)} = \frac{P(B/A)P(A)}{P(B)}$$

which relies on knowing a priori probability density functions *P(A)* and *P(B).* Such a priori knowledge is not required in DS theory.

Define the 'conditional bpa' of *B* given *A* as [Xu and Smets 1994]

$$m(B/A) = \begin{cases} \sum_{X \subset \bar{A}} m(B \cup X), & if\ B \subset A \\ 0, & otherwise \end{cases}.$$

The belief function constructed from this is called the conditional belief of *B* given *A* and denoted *Bel(B/A).*

Evidence from two witnesses 1 and 2 may be combined in several ways, e.g., using Dempster's rule of combination

$$m_{1,2}(A) = \frac{\sum_{B \cap C \subset A} m_1(A) m_2(B)}{1-\kappa} \qquad (1)$$

where the conflict is defined as

$$\kappa = \sum_{B \cap C = \phi} m_1(B) m_2(C).$$

We use here the un-normalized combination rule preferred in [Xu and Smets 1994]

$$m_{1,2}(A) = \sum_{B \cap C \subset A} m_1(A) m_2(B).$$

Define the combined belief function based on this combined bpa as the 'orthogonal sum'

$$Bel_{12} = Bel_1 \oplus Bel_2.$$

Propagating DS Beliefs in Networks.

Let $Y = \{y\} \subset 2^\theta$ be the core of a belief function over θ. Given a set of *N* agent nodes $\{v_i\} \in V$, each of which can formulate opinions about $y \in Y_i$, given in terms of the local (a priori) belief functions $Bel_i(y)$. Note that the cores at different nodes can be different, but the frame of discernment is the same. Let the nodes be connected in an acyclic decision graph $G^d = (V, E^d)$, where edge $(v_i, v_j) \in E^d$ if node v_i communicates with node v_j, i.e. v_j can use information from v_i for decision or estimation purposes.

It is desired to compute the marginal beliefs at each node using only local computations. Let the *k*th message sent by node $v_j \in N_i^d$ to node v_i be the belief vector $Bel_k^{ji}(y)$. Construct the messages using the *Belief message passing algorithm* [Xu and Smets 1994]

$$Bel_0^{ji}(y) = 1$$

$$Bel_{k+1}^{ji}(y) = \sum_{y_j \in Y_j} m_{i/j}(y / y_j) \left[Bel_j(y_j) \oplus \left(\bigoplus_{\ell \in N_j - \{i\}} Bel_k^{\ell j}(y_j) \right) \right]$$

where $m_{i/j}(y / y_j)$ is the conditional bpa induced at node *i* given information at node *j*, and $Bel_k^{\ell j}$ is the belief at node *j* conditioned on information at node ℓ at time *k*. After receiving the *k*th message, each node constructs an estimate of its local marginal belief using

$$\widehat{Bel}_k^i(y) = Bel_i(y) \oplus \left(\bigoplus_{j \in N_i} Bel_k^{ji}(y) \right)$$

This algorithm converges to the local marginal beliefs if the graph is undirected and is a tree [Xu and Smets 1994].

Matrix Computations for Dempster-Shafer Theory

This distributed algorithm is complicated to implement due to the numerous summations over set inclusions. It is conveniently implemented in terms of our new matrix formulation for DS computations [Ballal and Lewis 2007]. The beliefs and conditional bpas at each node are stored as vectors and operations are carried out in terms of matrices. Details follow.

Given a frame of discernment θ with focal elements $\{S_1, S_2, \cdots, S_q\}$, define a $q \times q$ containment matrix *C* as a matrix with element *(i,j)=1* if $S_i \supset S_j$, and *(i,j)=0* otherwise. Define a $q \times q$ intersection matrix *E* having element *(i, j)=1* if $S_i \cap S_j \neq \phi$, and *(i,j)=0* otherwise. Define the conflict matrix as $K = \overline{E}$ with overbar denoting logical negation. Define a *q*-vector of bpas indexed by the focal elements of θ as $\underline{m} = [m(S_1)\, m(S_2) \cdots m(S_q)]^T$. Define a commensurate belief *q*-vector $\underline{Bel} = [Bel(S_1) \cdots Bel(S_q)]^T$ and similarly a plausibility *q*-vector $\underline{Pl}$.

Now one has the easy matrix equations to compute belief and plausibility

$$\underline{Bel} = C\underline{m}$$

$$\underline{Pl} = E\underline{m}$$

The Dempster rule of combination is computed efficiently as follows. Two witnesses may have different information about different events. Thus, given two bodies of evidence, the two bpa vectors $\underline{m}_1, \underline{m}_2$ will not generally be commensurate since the two cores are different, e.g. the two vectors are indexed by different focal element sets. Therefore, merge the two cores to obtain the joint core, consisting of all the focal elements of the two bodies of evidence, each occurring once. Expand the two bpa vectors $\underline{m}_1, \underline{m}_2$, indexing each by the same joint core set and adding zeros as necessary to the vectors. Define the combination matrix

$$S_{12} = \underline{m}_2 \underline{m}_1^T$$

Then, the conflict is

$$K_{12} = \underline{m}_2^T K \underline{m}_1$$

and the combined bpa vector with elements given by (1) is

$$\underline{m}_{12} = diag\{SC + C^T S - S\}/(1 - K_{12})$$

where *diag{.}* means take the diagonal of a matrix as a vector. These equations are easily implemented, e.g., in MATLAB as matrix operations.

Trust Graphs: Propagation and Maintenance of Trust

Given a network of N agents or nodes $V = \{v_1, \cdots, v_N\}$ who are to engage in cooperative trust evaluation. Define a trust graph $G^t = (V, E^t)$, where edge $(v_i, v_j) \in E^t$ if node v_j obtains a direct trust evaluation about node v_i. Define the direct trust neighborhood of node v_i as $N_i^t = \{v_j : (v_j, v_i) \in E^t\}$, i.e. the set of nodes with edges incoming to v_i. The graph is directed since if node *j* can obtain a direct evaluation of trust about node *i,* the reverse may not be true. With each edge $(v_i, v_j) \in E^t$ associate a weight w_{ji}. Define the trust weight matrix $W^t = [w_{ij}]$, the weighted in-degree matrix as $D^t = diag\{n_i\}, n_i = |N_i^t|$, and the trust graph Laplacian L^t. If there is a directed path, e.g. a sequence of nodes $v_0, v_1, \cdots, v_r$ such that $(v_i, v_{i+1}) \in E, i \in \{0,1,\cdots, r-1\}$, then, node v_r should be able to form an indirect trust opinion about node v_0 based on the opinions of the agents along the path. Likewise, if two paths converge at an agent v_r, each of which contains agent v_0, then v_r has a basis to form a more confident opinion about the trustworthiness of agent v_0 than if there were only a single path.

Define a trust opinion as a pair (t_{ij}, c_{ij}) [Theodorakopoulos and Baras 2006], where t_{ij} is the trust that agent *i* has about agent *j*, and c_{ij} is the confidence agent *i* has in that trust value. Suppose for discussion here that $(t_{ij}, c_{ij}) \in [0,1] \times [0, \infty) \equiv S$, with S the opinion space, so that $t_{ij} = 0$ indicates minimum trust, $t_{ij} = 1$ indicates maximum trust, and $c_{ij} = 0$ indicates minimum confidence. Define two operators $\oplus, \otimes$ where $\oplus$ denotes the combination of opinions between two independent agents (i.e. at an intersection of two paths in the trust graph), while $\otimes$ denotes the combination of opinions along a single path in the trust graph. Clearly, along a single path, trust and confidence should decrease as one gets further from direct evidence about a given node. When trust opinions of two paths are combined, on the other hand, the confidence should increase and trust should be some sort of weighted average.

Useful choices are $(t_1, c_1) \otimes (t_2, c_2) = \left(t_1 t_2, (c_1^{-1} + c_2^{-1})^{-1}\right)$, so that along a path the trust and confidence decrease with confidence combining like the parallel combination of two resistors, and $(t_1, c_1) \oplus (t_2, c_2) = \left(\dfrac{c_1 t_1 + c_2 t_2}{c_1 + c_2}, c_1 + c_2 \right)$. It is interesting to note that the confidence is similar to the inverse of the error covariance in filtering theory [Lewis 1986, 2007], and plays a role akin to that of the Fisher information matrix. Then $\oplus$ is akin to the measurement update in the Kalman filter. It is important to select the operators $\oplus, \otimes$ so that $(S, \oplus, \otimes)$ forms an algebraic semiring [Theodorakopoulos and Baras 2006], with $\otimes$ distributing over $\oplus$. Then operations $t_{ij} = \bigoplus_k (t_{ik} \otimes t_{kj})$ are efficiently written in terms of matrices as $T = T_1 T_2$ with matrix multiply

defined as $\otimes$ and matrix addition defined as $\oplus$. This is easily written as a short program in, e,g., Matlab. Lewis [1997] has used the operations *or,and* as $\oplus, \otimes$ to allow discrete event rule-based controllers (e.g. Petri nets) to be written and efficiently computed in terms of matrices.

In [Jiang and Baras 2006, Theodorakopoulos and Baras 2006] the trust opinions are encoded in the *edge weights* of a graph. Then, edge weights become vector quantities, and more and more weights are added as trust relations propagate through the network.

By contrast, we encode the trust opinions an agent *i* has about other agents in the network as a trust vector ξ_i associated with each *node*, with elements indexed by all the nodes about which node *i* has an opinion. That is $\xi_i = [t_{ii_1}\ t_{ii_2}\ \cdots]^T$ where t_{ij} is the trust node *i* has for node *j*. The new values of ξ_i after an opinion update are given by

$$\xi_i(k+1) = \xi_i(k) + \mu_i(k)$$

where $\mu_i(k)$ is the new information obtained at time *k* that will result in a trust vector update. Note that, as node *i* gets information about more and more nodes, the trust vector ξ_i will grow in size (up to a max length of N, the number of nodes), but the structure of the trust graph does not change (in contrast to [Theodorakopoulos and Baras 2006]). Regardless, the local voting rule only depends on the opinions of each node and its neighbors and is easy to implement.

One possibility for trust updating is the local voting rule $\mu_i(k) = \nu_i(k) + \frac{1}{1+n_i}\sum_{j\in N_i} w_{ij}(\xi_j - \xi_i)$, with $n_i = \sum_{j\in N_i} w_{ij}$ the weighted in-degree. The auxiliary input $\nu_i(k)$ represents trust value updates at time *k* arising from new evidence observed at node *i*. This voting rule is based on differences of opinion between node *i* and its neighbors, and yields

$$\xi_i(k+1) = \frac{1}{1+n_i}\left(\xi_i(k) + \sum_{j\in N_i} w_{ij}\xi_j\right) + \nu_i(k).$$

Note that the operations of addition and multiplication throughout his discussion are in fact given by the selected trust update operations $\oplus, \otimes$.

Define the overall state vector $\xi = [\xi_1^T\ \xi_2^T \cdots \xi_N^T]^T \in R^{N^2}$ and the corresponding overall trust input vector $\mu \in R^{N^2}$. Then, if $\nu_i(k) = 0$ the overall system trust dynamics becomes

$$\mu(k) = \left((I_N + D^t)\odot I_N\right)^{-1}\left(L^t \odot I_N\right)\xi(k), \qquad \xi(k+1) = \xi(k) - \left((I_N + D^t)\odot I_N\right)^{-1}(L^t \odot I_N)\xi(k)$$

$$\xi(k+1) = \left((I_N + D^t)\odot I_N\right)^{-1}\left((I_N + W^t)\odot I_N\right)\xi(k) \equiv F\xi(k)$$

where W^t is the trust graph weighted connectivity matrix, D^t the diagonal in-degree matrix, and $L^t = D^t - W^t$ the graph Laplacian matrix. The $N\times N$ identity matrix is I_N and $\odot$ is the Kronecker product. This is a vector version of the Vicsek system [Jadbabaie et al. 2003] studied in flocking behavior and particle phase transitions. Note that *F* is a stochastic matrix.

The voting rule given above is *local* in that each node only uses information from itself and its neighbors. We would like to know *when all the nodes in the network reach the same conclusion about the mutual trustworthiness of the nodes*. This is a *global consensus problem*

where the trust vector ξ_i of each node should converge to the same consensus value ξ^*, i.e. $\lim_{k\to\infty} \xi_i(k) = \xi^*, \forall i$. The next result is not difficult to establish using properties of the Kronecker product and the graph Laplacian.

Our First Lemma. Consensus of Trust. Let the new evidence $\nu_i(k) = 0$. Then the local trust update rule given results in a global consensus if the trust graph is strongly connected, and if and only if it has a spanning tree. Then the consensus trust vector is

$$\xi^*(\xi(0)) = \frac{\sum_i \gamma_i \xi_i(0)}{\sum_i \gamma_i}$$

with $\xi_i(0)$ the initial trust vector of node *i*, and $w_l = [\gamma_1\ \gamma_2 \cdots \gamma_N]^T$ the left eigenvector of L^t for eigenvalue $\lambda=1$.

This is not a nice result since the consensus trust vector depends on the network connection structure through the left eigenvectors of the graph Laplacian. The next result shows when it does not.

Our Second Lemma. Consensus of Trust to Average of Initial Trusts. Let the trust digraph be strongly connected. Then the consensus trust vector is

$$\xi^*(\xi(0)) = \tfrac{1}{N} \sum_i \xi_i(0)$$

e.g., the average of the initial trust vectors $\xi_i(0)$ of all the nodes, if and only if the trust graph is balanced.

Recall that a digraph is balanced if the in-degree of each node is equal to its out-degree.

The weights w_{ij} for edge e_{ji} in the voting rule can be selected to depend on the confidence c_{ij} that node *i* has in node *j*. The confidence values can be updated using a variant of the Kalman error covariance updates given in [Ren, Beard, Kingston 2005]. Such updates need to be further investigated. We will also investigate other local trust update protocols.

The topology of the trust graph may change as agents move. This can be handled using techniques such as those in [Jadbabaie et al. 2003, Saber and Murray, Ren and Beard 2005].

The Lemmas given also hold if some trust values t_{ij} are negative, e.g. node *i* distrusts node *j*. However, more work is needed in selecting the trust operations $\oplus, \otimes$ for negative trust values, since the enemy of one's enemy is not always one's friend.

Cooperative Control Based on Trust

Consider a dynamic graph with node dynamics $\dot{x}_i = f_i(x_i) + g_i(x_i)u_i$, $y_i = h_i(x_i)$. The trust values as updated above may be incorporated into a networked control law for formations, flocking, etc., as

$$u_i = \sum_{j\in N_i} t_{ij} K_{ij} (y_j - y_i).$$

For example, if node *i* has no trust basis for node *j*, then t_{ij} is equal to zero, and node *i* would not respond to changes in the state of node *j*. Again, if node *i* distrusts node *j*, then t_{ij} is negative, and this would, e.g., cause the motion of node *i* to be away from any approaching distrusted node *j*. More work needs to be done to understand all of this.

References

T. Balch and R.C. Arkin, “Behavior-based formation control for multirobot teams,” IEEE Trans. Robotics and Automation, vol. 14, no. 6, pp. 926-939, Dec. 1998.

J. Baillieul, “Remarks on a simple control law for point robot formations with exponential complexity,” Proc. IEEE Conf. Decision and Control, pp. 3357-3362, Dec. 2006.

N. Chopra and M.W. Spong, “Passivity-based control of multi-agent systems,” in “Advances in Robot Control: From Everyday Physics to Human-Like Movements,” ed. S. Kawamura and M. Svinin, pp. 107-134, Springer-Verlag, Berlin, 2006.

P. Dang, F.L. Lewis, and D.O. Popa, “Dynamic Localization of Air-Ground Wireless Sensor Networks,” Proc. Mediterranean Conf. Control and Automation paper WM1-1, Ancona, June 2006.

W.B. Dunbar and R.M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” Automatica, vol. 42, pp. 549-558, 2006.

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, Sept. 2004.

V. Gazi and K.M. Passino, “Stability analysis of swarms,” IEEE Trans. Automatic Control, vol. 48, no. 4, pp. 692-697, April 2003.

V. Gazi and K.M. Passino, “A class of attractions/repulsion functions for stable swarm aggregations,” Int. J. Control, vol. 77, no. 18, pp. 1567-1579, 2004.

F. Giulietti, L. Pollini, and M. Innocenti, “Autonomous formation flight,” IEEE Control Systems Magazine, pp. 34-44, Dec. 2000.

A. Jadbabaie, J. Lin, and A.S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Trans. Automatic Control, vol. 48, no. 6, pp. 988-1001, June 2003.

T. Jiang and J.S. Baras, “Trust evaluation in anarchy: a case study on autonomous networks,” Proc. Infocom, Barcelona, 2006.

H. Lakshmanan and D.P. de Farias, “Decentralized approximate dynamic programming for dynamic networks of agents,” Proc. Amer. Control Conf., pp. 1648-1653, June 2006.

D. Lee and M.W. Spong, “Stable flocking of multiple inertial agents on balanced graphs,” preprint, 2007.

N.E. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials, and coordinated control of groups,” Proc. IEEE Conf. Decision and Control, pp. 2968-2973, Dec. 2001.

F.L. Lewis, D.Tacconi, H.-H. Huang, A. Gurel, and O. Pastravanu, “Manufacturing controller design and deadlock avoidance using a matrix model for discrete event systems,” in Methods and Applications of Intelligent Control, ed. S. Tzafestas, chapter 17, 1997.

F.L. Lewis, “Optimal Estimation,” John Wiley, New York, 1982.

F.L. Lewis, L. Xie, and D. Popa, Optimal and Robust Estimation, second edition, CRC Press, Boca Raton, 2007.

R. Olfati-Saber and R.M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Trans. Automatic Control, vol. 49, no. 9, pp. 1520-1533, Sept. 2004.

J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,” Morgan-Kauffman, 1988.

W. Ren and R.W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” IEEE Trans. Automatic Control, vol. 50, no. 5, pp. 655-661, May 2005.

W. Ren, R.W. Beard, and D.B. Kingston, “Multi-agent Kalman consensus with relative uncertainty,” Proc. Amer. Control Conf., pp. 1865-1870, Portland, 2005.

V. Saligrama, M. Alanyali, and O. Savas, “Distributed detection in sensor networks with packet losses and finite capacity links,” IEEE Trans. Signal Proc., vol. 54, no. 11, pp. 4118-4132, Nov. 2006.

S.H. Strogatz, “From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators,” Physica D, vol. 143, pp. 1-20, 2000.

G. Theodorakopoulos and J.S. Baras, “On trust models and trust evaluation metrics for ad hoc networks,” IEEE J. Selected Areas in Communications,” vol. 24, no. 2, pp. 318-328, Feb. 2006.

H. Xu and P. Smets, “Reasoning in evidential networks with conditional belief functions,” Int. J. Approximate Reasoning,” vol. 11, no. 1, pp. 1-31, 1994.