Dynamic Resource Networks: Coordination and Control of Networks with Mobile Actuators and Sensors

Kevin L. Moore, Ph.D., P.E.¹
G.A. Dobelman Distinguished Chair and Professor of Engineering
Division of Engineering
Colorado School of Mines

¹Some work performed at the Johns Hopkins University Applied Physics Laboratory and the Utah State University Center for Self-Organizing and Intelligent Systems
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem
Networked Sensors

- Through history, many technologies have become ubiquitous:
  - Microprocessors, Motors
- Today a new technology has the same promise;
  - Networked sensors
- Due to advances in biology, electronics, nanotechnology, wireless communications, computing, networking, and robotics, we can now:
  - Design advanced sensors and sensor system
  - Use wireless communications, or telemetry, to more effectively communicate sensor data from a distance than ever before
  - Build networks of sensors, using wireless communications and computer networking technology, that can provide the capability to obtain spatially-distributed measurements from low-power sensors which communicate and relay information between each other
  - Develop reconfigurable, or adaptable, networks of distributed sensors by providing mobility or actuation to the individual sensors in the network
Technology Enables Paradigm Change

• Smaller, larger, more complex: “smaller becomes larger”
  – Nano-tech, MEMS, molecular computing devices
• Advances in sensors, communications, nanotechnology, and biology enable the concept

  “information about everything available everywhere”

• Information is no longer a simple sensor output, but becomes multi-modal/multi-media
• Result is more complex engineered systems, with implications for
  – Hardware: distributed (wireless) and embedded
  – Software: parallel, distributed, intelligent algorithms
  – Systems: distributed and embedded, issues include:
    ▪ Information management
    ▪ Information processing
    ▪ Decision making
• Systems and control thinking becomes key
Paradigm Change Brings Challenges

• Key challenge: the classical, lumped parameter ODE/PDE paradigm of systems and control is inadequate for future progress.

“… Essentially every thing done in the last [50] years of control theory rests on a common presumption of centrality [that all the information available about the system, and the calculations based on this information, take place at a single location]…” “Survey of decentralized control methods for large scale systems,” Sandell, Varaiya, Athans, Sarnov, IEEE Transactions on Automatic Control, April 1978.

• From an algorithmic perspective, we need to turn to spatial-temporal methodologies.
Paradigm Change Brings Opportunity

• One perspective on future challenges and opportunities for sensor networks can be summarized by:

Only “…through control will sensor networks achieve their value …,” Michael Bruns, VP A&D AS Process Automation, Siemens AG, DE, comments during Plenary Lecture, Wednesday, July 6, 2005 IFAC World Congress, Prague.

• From this perspective, I would like to comment on ideas related to what “Dynamic Resource Networks,” which includes the idea of “Mobile Actuator/Sensor Networks,” or MASNET
Dynamic Resource Networks

- Network of “entities”
  - Communication infrastructure
  - Entity-level functionality
  - Implied global functionality
  - Not necessarily homogeneous

- Resource:
  - Primarily considering entities that are sensors
  - “Bigger picture” includes actors as well

- Dynamic
  - Entities may be mobile
  - Communication topology might be time-varying
  - Data actively and deliberately shared among entities
  - Decision-making and learning
A Prototypical Problem

• Given:
  – Network of distributed, static sensors
    ▪ Each instantiated with a perfect “classifier” that allows successful target identification if given perfect measurements
    ▪ Each receives corrupted signatures due to sensor placement in the environment
    ▪ Each sensor has a limited detection range and a limited communication range and BW between sensors
    ▪ Each sensor is connected to some of the other sensors; assume at least one spanning tree exists in the graph; also assume that at least range between communicating sensors is known
    ▪ Each sensor can compute the bearing and possible classification of three dominant targets in its region of detection
  – Multiple moving targets
    ▪ Motion vector (velocity, attitude)
    ▪ Characteristic signature (e.g., tank, car, motorcycle, etc.) with spatially-limited range of influence

• Find:
  – Estimated motion vector and classification of all targets moving through the sensor field
A Dual to the Prototypical Problem

- **Given:**
  - **Network of distributed, static targets**
    - Each has a characteristic signature with spatially-limited range of influence, each related to the other in some (known) way (e.g., fixed network of weather stations for ecological monitoring in the wilderness)
    - Each target is connected to some of the other sensors; assume at least one spanning tree exists in the graph
    - Some targets have higher-level communications
  - **Single moving sensor**
    - Instantiated with a perfect “classifier” that allows successful classification of the phenomena represented by the targets if given perfect measurements
    - Receives corrupted signatures due to sensor placement in the environment

- **Find:**
  - Classification of the phenomena represented by the targets
A Set of Related Problems

- Sensors:
  - Passive (listen only) or Active (illuminates and listens)
  - Homogeneous or heterogeneous
  - Mobile or Fixed
  - Single or Multiple (Isolated or Networked)

- Target:
  - Passive (must be illuminated) or Active (generates signature)
  - Homogeneous or heterogeneous
  - Mobile or Fixed
  - Single or Multiple (Isolated or Networked)
Dynamic Resource Networks

• Define seven types of resources:
  - UGS: Unattended ground sensor
  - UGA: Unattended ground system with direct action capability (e.g., autonomous artillery)
  - UGS-n: Local network of multiple UGS
  - UAV-s: UAV used as a sensor
  - UAV-a: UAV with direct action capability (e.g., strike)
  - UGV-s: UGV used as sensor
  - UGV-a: UGV with direct action capability

• Each entity is characterized by
  - Type
  - Action Region
  - Sensor region of attraction (blue)
  - Actor region of influence (red)
  - Communication Envelope (yellow)
  - Mobility Vector (green)
Motivating Example 1

- Autonomous swarm for plume tracking

Sensor-carrying UAVs and UGVs assess and track the development of a hazardous plume resulting from a CBR terrorism act.
Motivating Example 2

- Autonomous confederation building, adaptive to changes in battlefield conditions
Motivating Example Three -1

- **Data exfiltration application**
  - Sensor clusters are deployed “by hand”
  - Not all clusters can communicate with each other
  - Exact cluster locations are not known

- **UAVs execute a cooperative search to find the clusters**
  - Search begins with a pre-planned raster scan
  - Search is refined based on cluster discovery results

- **After discovery, UAVs cooperate to collect data from all the clusters in some optimal way**
  - UAVs configure to provide maximum coverage of clusters or
  - An optimal path is planned to travel between clusters
Motivating Example Three -2

- Adaptation occurs in response to changes in the UAV resources or the sensor clusters:
  - Periodically, one of the UAVs returns to the base station to relay data from the sensor clusters; when this happens the remaining UAV automatically re-plans its operations.
  - If a sensor cluster is lost the UAVs cooperatively reconfigure.
Motivating Example Four

Landslide Detection/Prediction

- Network of communication sensor such as geophones
- Changes in relative locations used to detect onset of landslide
Motivating Example Five

Center Pivot Irrigator Control

- Assume can bury a sensor at some prescribed depth at regular intervals
- Adjust water application rate based on the sensor readings between cycles
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem

KL Moore, Dynamic Resource Networks, CSM MCS Colloquium, 12/2/05
Mote-Based Distributed Robots

Prototype plume-tracking testbed - 2004

$2000 2\textsuperscript{nd} Place Prize in 2005 Crossbow Smart-Dust Challenge
MAS-DIFF Simulator Developed at USU
(Prof. YangQuan Chen)
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem
An Algorithmic Approach for Cooperation

- Bigger picture is coordination and control of multiple, cooperating, heterogeneous entities:

- Our technical approach, a generalization of potential field approaches, is based on so-called consensus variables and has connections to problems in:
  - Coupled-oscillator synchronization
  - Neural networks
Consensus Variable Perspective

- **Assertion:**
  - Multi-agent coordination requires that *some* information must be shared

- **The idea:**
  - Identify the essential information, call it the *coordination or consensus variable*.
  - Encode this variable in a distributed dynamical system and come to consensus about its value

- **Examples:**
  - Heading angles
  - Phase of a periodic signal
  - Mission timings

- **In the following we build on work by Randy Beard, Wei Ren, *et al.*, at BYU, to use consensus variables to solve global problems in a distributed fashion**
Consensus Variables

- Suppose we have $N$ agents with a shared global consensus variable $\xi$
- Each agent has a local value of the variable given as $\dot{\xi}_i$
- Each agent updates their local value based on the values of the agents that they can communicate with

$$\dot{\xi}_i(t) = -\sum_{j=1}^{N} k_{ij}(t)G_{ij}(t)(\xi_i(t) - \xi_j(t))$$

where $k_{ij}$ are gains and $G_{ij}$ defines the communication topology graph of the system of agents
- Key result from literature: If the graph has a spanning tree then for all $i$ $\xi_i \to \xi^*$
Consensus Variables

- Specifically, let $\xi = (\xi_1, \xi_1, \ldots, \xi_N)^T$ so that $\dot{\xi} = C\xi$
- If C satisfies $c_{ii} < 0$
  $c_{ij} > 0, i \neq j$
  $\sum_j c_{ij} = 0, \forall i$

Then:

1. C has one eigenvalue at zero and the others are stable (occurs iff the associated graph has a spanning tree)
2. $e^{Ct}$ is a stochastic matrix (positive, with row sum equal one)
3. $e^{Ct} \rightarrow \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \nu_1 & \ldots & \nu_N \end{bmatrix}, \sum_i \nu_i = 1$
Example: Single Consensus Variable

$$\begin{pmatrix}
\xi_1 \\
\xi_2 \\
\xi_3 \\
\xi_4 \\
\xi_5 \\
\xi_6 \\
\xi_7 \\
\xi_8 \\
\xi_9
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -k_{43} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -k_{54} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -k_{56} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -k_{67} - k_{68} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{71} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{87}
\end{pmatrix}
\begin{pmatrix}
\xi_1 \\
\xi_2 \\
\xi_3 \\
\xi_4 \\
\xi_5 \\
\xi_6 \\
\xi_7 \\
\xi_8 \\
\xi_9
\end{pmatrix}$$
Extension 1 - Forced Consensus

- Forced Consensus
  - Sometimes we may like to force all the nodes to follow a hard constraint
  - This can be done by injecting an input into a node as follows
    \[ \dot{\xi}_i(t) = - \sum_{j=1}^{N} k_{ij}(t)G_{ij}(t)(\xi_i(t) - \xi_j(t)) + b_i u_i \]
  - Then we use a feedback controller as given in the following

- **Theorem** Let \( A \) be a set of agents with \( b_k = 1, b_i = 0, \forall i \neq k \), and
  \[ u_k(t) = k_p(\xi^{sp} - \xi_k) \]
  where \( \xi^{sp} \) is a constant setpoint and \( k_p > 0 \) is a constant gain. Then the consensus strategy achieves global asymptotic consensus for \( A \), with
  \[ \lim_{t \to \infty} \xi_i(t) = \xi^{sp} \quad \forall i \]
  if and only if node \( k \) is a spanning node for the communication graph \( G \).
Example – Forced Consensus
Extension 2 – Multiple, Constrained Consensus

- Often we will have multiple consensus variables in a given problem.

- It can be useful to enforce constraints between these variables, specifically, to have \( \xi_i = \xi_j + \Delta_{ij} \).

- Again we can give a feedback control strategy to achieve this type of constrained consensus between groups of agents.
**Theorem**  Let \( A^a \) and \( A^b \) be two set of agents, each negotiating locally about consensus variables \( \xi^a \) and \( \xi^b \), respectively, and each with communication graphs \( G^a \) and \( G^b \) defined by communication topologies \( G^a_{ij} \) and \( G^b_{ij} \), respectively. Suppose

1. Each agent set updates the local values of their consensus variable by
   
   \[
   \dot{\xi}^a_i(t) = -\sum_{j=1}^{n^a} k^a_{ij}(t)G^a_{ij}(t)(\xi^a_i(t) - \xi^a_j(t)) + b^a_i u^a_i
   \]
   
   \[
   \dot{\xi}^b_i(t) = -\sum_{j=1}^{n^b} k^b_{ij}(t)G^b_{ij}(t)(\xi^b_i(t) - \xi^b_j(t)) + b^b_i u^b_i
   \]

   where \( b^a_{k^a} = 1, b^a_i = 0, \forall i \neq k^a \) and \( b^b_{k^b} = 1, b^b_i = 0, \forall i \neq k^b \)

2. The two agent sets communicate to each other via the nodes \( k^a \) and \( k^b \) using the following agent-to-agent consensus update law:
   
   \[
   u^a_{k^a} = -(\Delta_{ab} - (\xi^b_{k^b} - \xi^a_{k^a}))
   \]
   
   \[
   u^b_{k^b} = \Delta_{ab} - (\xi^b_{k^b} - \xi^a_{k^a})
   \]

Then the consensus strategy achieves global asymptotic global consensus for each set \( A^a \) and \( A^b \), with

\[
\xi^a_i \to \xi^a^* \\
\xi^b_i \to \xi^b^* \\
\xi^b^* = \xi^a^* + \Delta_{ab}
\]

if and only if nodes \( k^a \) and \( k^b \) are spanning nodes for the graphs \( G^a \) and \( G^b \), respectively.
Example – Multiple, Constrained Consensus
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem
Outline

- Introductory Comments
- Dynamic Resource Networks (DSN)
  - General Ideas
  - Motivating Examples
- A Framework for Diffusion Problems
- A Coordinated Optimization Approach
  - Consensus Variables and Extensions
    - Example – Decentralized, cooperative adaptive scheduling
  - Higher-order Consensus
    - Example – Flocking-like behavior in coordinated motion
  - Distributed Computation
  - Global Optimization via Coordination
    - Example – Data exfiltration problem
Extension 3 – Higher-Order Consensus

• Can generalize to higher-order as follows. Let

\[ \xi^{(1)}_i = \xi^0_i \]
\[ \xi^{(2)}_i = \xi^1_i \]
\[ \vdots \]

\[ \xi^l_i (t) = -\sum_{j=1}^{N} k_{ij} (t) G_{ij} (t) \left[ \sum_{k=0}^{l-1} \gamma_k (\xi^{(k)}_i - \xi^{(k)}_j) \right] \]

• Then, if there is a spanning tree (and system is stable) you will get

\[ \xi^l_i \rightarrow \xi^l_i^* \text{ for all } i \]
\[ \xi^{l-1}_i \rightarrow t \xi^{l-1}_i^* + \gamma \text{ for all } i \]
\[ \xi^{l-2}_i \rightarrow t^2 \xi^{l-2}_i^* + t \gamma + \beta \text{ for all } i \]

• Stability depends on the gains \( \gamma_k \)
• In more detail, we can show that the global dynamics are given by

\[
\begin{bmatrix}
\dot{\xi}^{(0)} \\
\dot{\xi}^{(1)} \\
\vdots \\
\dot{\xi}^{(\ell-1)}
\end{bmatrix} = (\Gamma \otimes I_m)
\begin{bmatrix}
\xi^{(0)} \\
\xi^{(1)} \\
\vdots \\
\xi^{(\ell-1)}
\end{bmatrix}
\]

where

\[
\Gamma = 
\begin{bmatrix}
0_n & I_n & 0_n & \cdots & 0_n \\
0_n & 0_n & I_n & \cdots & 0_n \\
0_n & \cdots & \cdots & \cdots & \cdots \\
0_n & 0_n & \cdots & \cdots & I_n \\
-\gamma_0 L & -\gamma_1 L & -\gamma_2 L & \cdots & -\gamma_{\ell-1} L
\end{bmatrix}
\]

and \( L \) is the communication topology matrix.

• Turns out \( \Gamma \) has \( \ell \) zero eigenvalues (each of geometricity 1) iff \( L \) has one one zero eigenvalue (iff there exists a spanning node).
Extension 3a – Model-Reference Consensus

As a final point, suppose we are given a reference model, defined by

\[ \dot{\xi}_r^{(0)} = \dot{\xi}_r^{(1)} \]
\[ \vdots \]
\[ \dot{\xi}_r^{(\ell-2)} = \dot{\xi}_r^{(\ell-1)} \]
\[ \dot{\xi}_r^{(\ell-1)} = u_r \]

Let the consensus protocol be given as

\[ \dot{\xi}_i^{(1)} = \xi_i^0 \]
\[ \dot{\xi}_i^{(2)} = \xi_i^1 \]
\[ \vdots \]
\[ \dot{\xi}_i^l(t) = -\sum_{j=1}^{N} k_{ij}(t)G_{ij}(t)\left[\sum_{k=0}^{l-1} \gamma_k (\xi_i^{(k)} - \xi_j^{(k)})\right] - \eta \sum_{k=0}^{l-1} \gamma_k (\xi_i^{(k)} - \xi_r^{(k)}) + u_r \]

Then it is possible to give conditions for which \( \xi_i^{(k)} \to \xi_r^{(k)} \) for all \( i, k \)
Extension 3 – Higher-Order Consensus

- For example, consider a third-order consensus problem, applied to a formation control problem with five vehicles, with one vehicle having an acceleration setpoint input.

\[
\begin{align*}
\dot{x}_i &= v_i \\
\dot{v}_i &= a_i \\
\dot{a}_i &= -\sum_{j=1}^{n} g_{ij} k_{ij} \{\gamma_0 [(x_i - \delta_i) - (x_j - \delta_j)] \\
&\quad + \gamma_1 (v_i - v_j) + \gamma_2 (a_i - a_j)\} - \alpha (a_i - a_i^*)
\end{align*}
\]

Enables formation control

Acceleration Input
Extension 3 – Higher-Order Consensus

- Suppose the “leader node” sees the following acceleration input profile:

![x-axis acceleration](image1)

![y-axis acceleration](image2)
Extension 3 – Higher-Order Consensus

- The resulting paths look like:
Extension 3 – Higher-Order Consensus

- The resulting x-axis trajectory shows the effect of the higher-order consensus algorithm

![Graphs showing x-axis trajectory](image)
Outline

- Introductory Comments
- Dynamic Resource Networks (DSN)
  - General Ideas
  - Motivating Examples
- A Framework for Diffusion Problems
- A Coordinated Optimization Approach
  - Consensus Variables and Extensions
    - Example – Decentralized, cooperative adaptive scheduling
  - Higher-order Consensus
    - Example – Flocking-like behavior in coordinated motion
  - Distributed Computation
  - Global Optimization via Coordination
    - Example – Data exfiltration problem
Extension 4 – Distributed Computation

• Suppose we want to solve

\[
\begin{pmatrix}
\dot{v}_1 \\
\vdots \\
\dot{v}_N
\end{pmatrix}
= 
\begin{pmatrix}
f_1(v) \\
\vdots \\
f_N(v)
\end{pmatrix}
\quad \text{for} \quad v=(v_1, \ldots, v_N)^T \in \mathbb{R}^N
\]

where we think of \( v_i \) as a local variable on node \( i \)

• Assumptions are that
  – we do not have global knowledge of the complete vector \( v \) on every node
  – nodes are connected with a graph topology whereby every node is a spanning node
Extension 4 – Distributed Computation

- Approach uses a multivariable extension of the forced consensus ideas presented above.
- Let \( \nu^* \) be the solution of \( \dot{\nu} = f(\nu) \)
- Introduce a variable \( x_i \) at each node (a local estimate of \( \nu_i \))
- Set up the following system on each node

\[
\begin{align*}
\dot{x}_i &= f_i(\xi_{i1}, \ldots, \xi_{iN}) \\
\begin{pmatrix}
\dot{\xi}_{i1} \\
\vdots \\
\dot{\xi}_{iN}
\end{pmatrix} &= -\sum_{j=1}^{N} k_{ij}(t) G_{ij}(t)(\xi_i(t) - \xi_j(t)) + b_i(x_i - \xi_{ii})
\end{align*}
\]

- Then \( x_i \to \nu_i^* \) for all \( i \) if every node is a spanning node (conjecture)

\( b_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T \)

Inserts a setpoint into the \( i-th \) component of node \( i \)
Extension 4 – Distributed Computation

\[ \dot{x}_7 = f_7(\xi_7, \ldots, \xi_{79}) \]

\[
\begin{pmatrix}
\ddot{\xi}_{71} \\
\vdots \\
\ddot{\xi}_{79}
\end{pmatrix} = -k_{71}(t)(\xi_7(t) - \xi_1(t)) - k_{78}(t)(\xi_7(t) - \xi_8(t)) + b_7(x_7 - \xi_{77})
\]

\[ \dot{x}_4 = f_4(\xi_{41}, \ldots, \xi_{49}) \]

\[
\begin{pmatrix}
\dot{\xi}_{41} \\
\vdots \\
\dot{\xi}_{49}
\end{pmatrix} = -k_{43}(t)(\xi_4(t) - \xi_3(t)) + b_4(x_4 - \xi_{44})
\]

\[ b_4 = (0,0,0,0,0,0,0,0,0)^T \]

\[ b_7 = (0,0,0,0,0,0,1,0,0)^T \]
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem
Outline

• Introductory Comments
• Dynamic Resource Networks (DSN)
  – General Ideas
  – Motivating Examples
• A Framework for Diffusion Problems
• A Coordinated Optimization Approach
  – Consensus Variables and Extensions
    ▪ Example – Decentralized, cooperative adaptive scheduling
  – Higher-order Consensus
    ▪ Example – Flocking-like behavior in coordinated motion
  – Distributed Computation
  – Global Optimization via Coordination
    ▪ Example – Data exfiltration problem