A Dynamic Network Perspective on Resilient Control

Kevin L. Moore
Dean and Professor
College of Engineering & Computational Sciences
Colorado School of Mines

Plenary Lecture

Resilience Week 2014
Denver, Colorado
20 August 2014

Colorado School of Mines

Located in Golden, Colorado, USA
10 miles West of Denver

CSM sits in the foothills of the Rocky Mountains

CSM has ~300 faculty and ~5600 students
(~4200 undergrad and ~1400 grad)

CSM is a public research institution devoted to engineering and applied science, especially:

- Discovery and recovery of **resources**
- Conversion of resources to **materials and energy**
- Utilization in advanced **processes and products**
- Economic and social systems necessary to ensure **prudent and provident use of resources in a sustainable global society**

Sustainable Global Society

CSM
PRO

- W
- Sc
- Al)
- Po tion res
- NS
- No ator)
- En
- Bi l
- Coursework in sustainable design
- Humanitarian Engineering minor; Engineering by Doing program

- **Comment:** Resilience, especially as related to infrastructure, is essential to a sustainable global society

- **Today:** Explore resilience from the perspective of controlling dynamic networks

Synopsis -1

- Many important systems can be modeled as
  - Collection or network of **integrating agents or subsystems**
  - **Exchanging energy, material, or information**
  - According to some **protocol or physical laws**
  - Subject to an **interconnection topology**

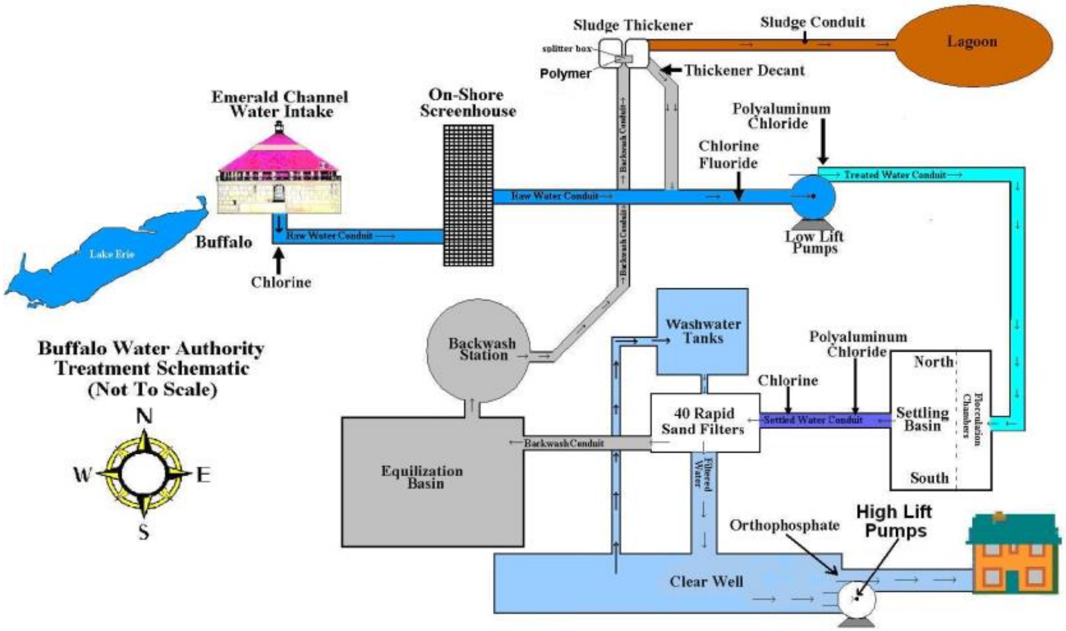


Figure: from http://www.buffalowater.org/files/Schematic-2011.JPG. Used without permission.

Example: Natural Gas System

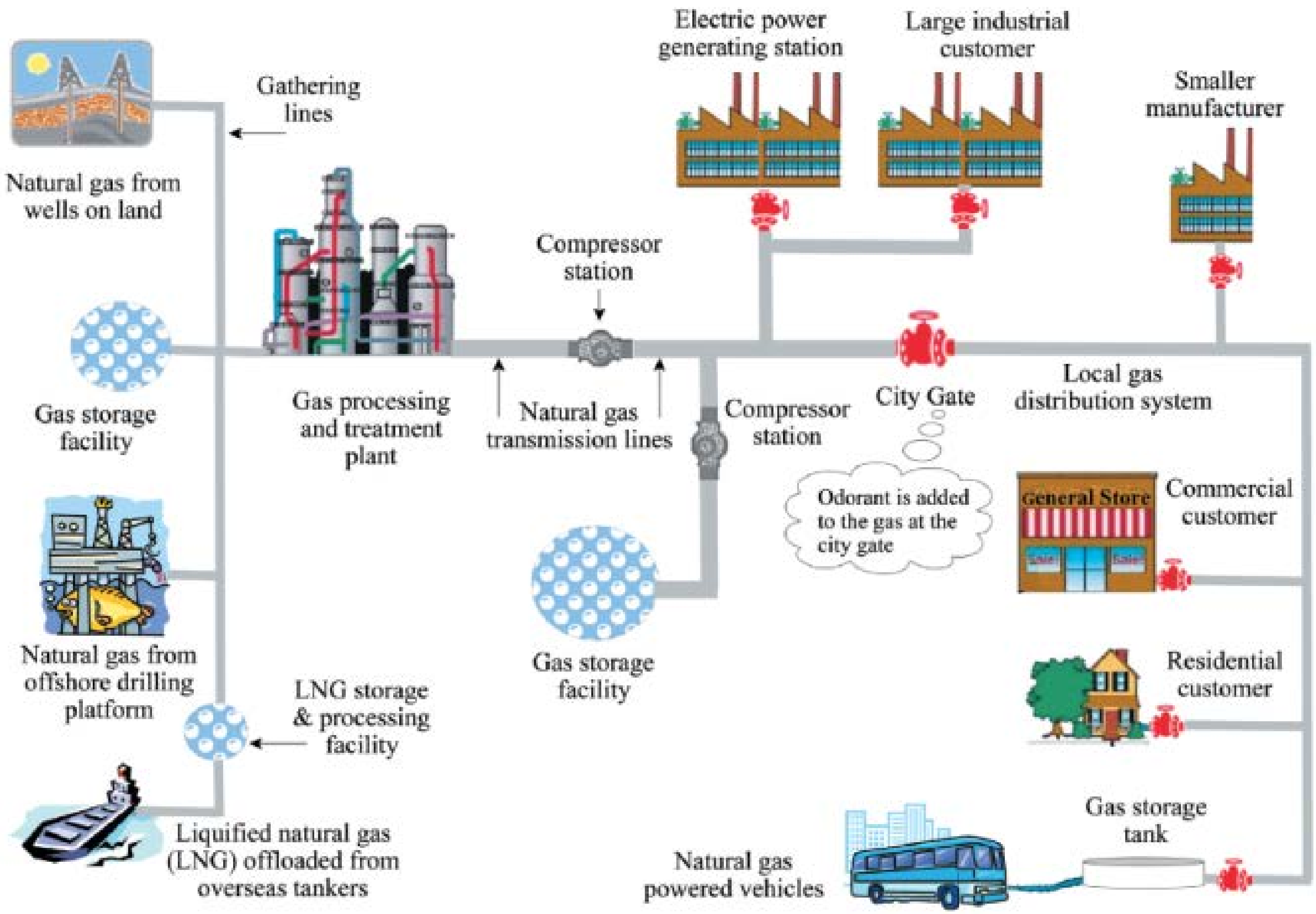
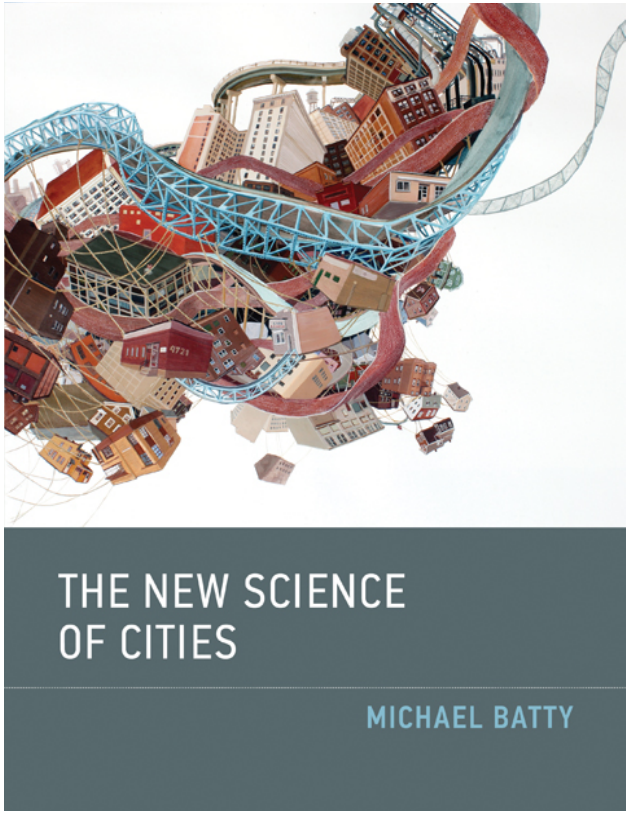


Figure: from Figure 256 in “From Reservoir to Burner Tip: A Primer,” Curtis and Schwochwow, in *Potential Supply of Natural Gas*, 2008. Used without permission.

Synopsis -2

- When such systems are governed by differential equations we call them a **dynamic network**
- Also called a **cyber-physical system** when there is
  - Tight integration of
    - Physical system dynamics
    - Sensors and actuators
    - Computing infrastructure
  - Multiple time and spatial scales
  - Multiple behavioral modalities
  - Context dependent interactions
- **Example**: Intelligent vehicle-highway system; cities



Example: Global Supply Chain

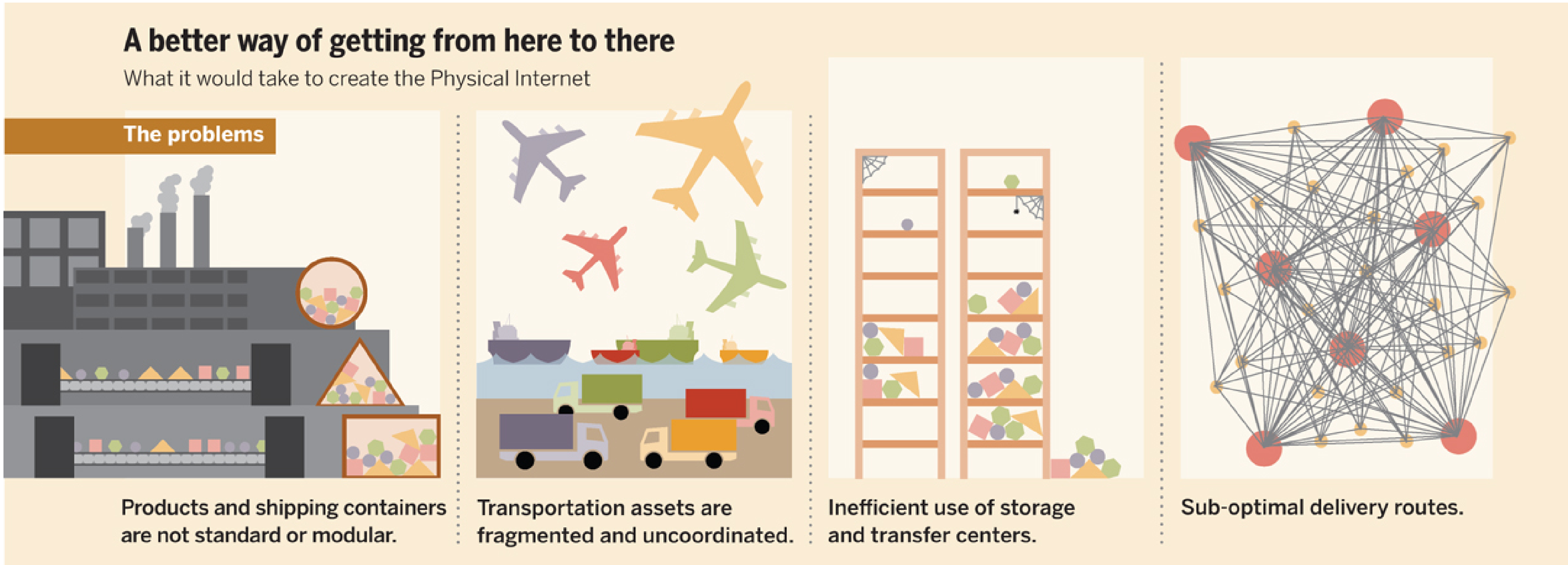
B Wible et al. Science 2014;344:1100-1103

Published by AAAS

Port of Hong Kong

Science
AAAS

Logistics Today



Non-standard, non-modular shipping containers

Fragmented, un-coordinated transportation assets

Inefficient storage & transfer centers

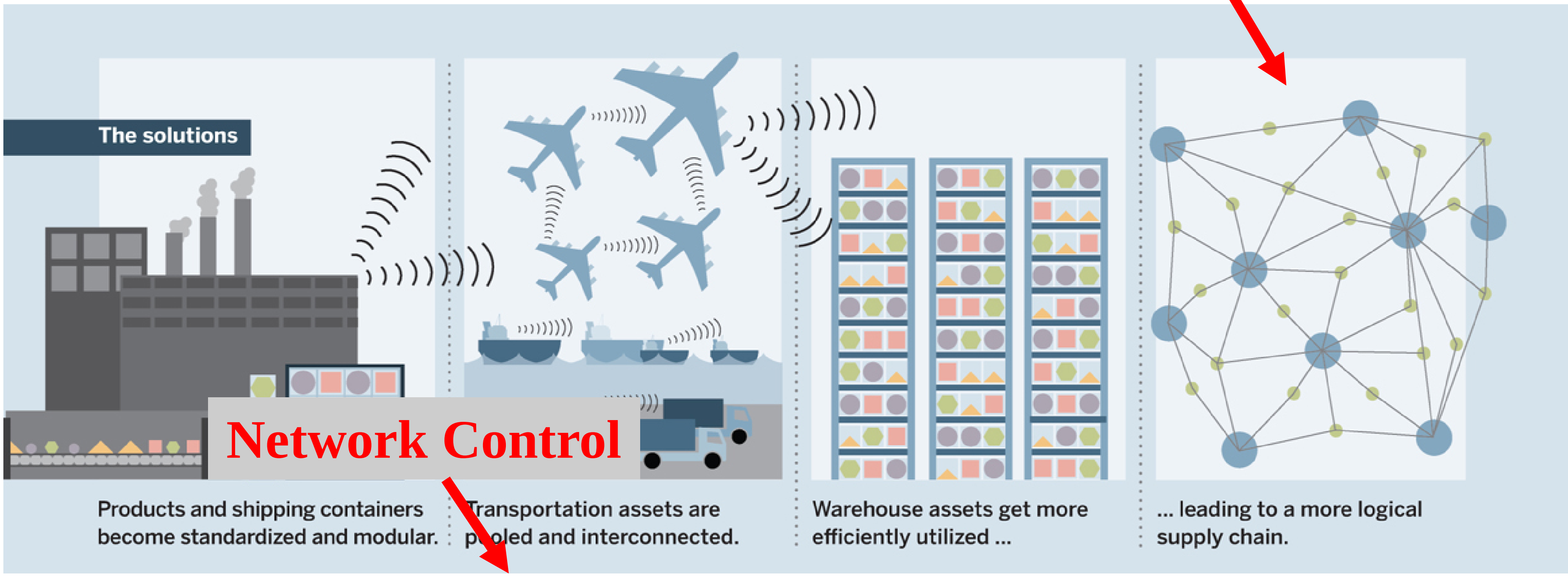
Sub-optimal delivery routes

Figure from J Mervis Science 2014;344:1104-1107

Published by AAAS

Logistics Tomorrow

Resilient Dynamic Network



Network Control

Standard, modular shipping containers

Pooled, interconnected transportation assets

Organized, coordinated warehouse assets

A more logical supply chain

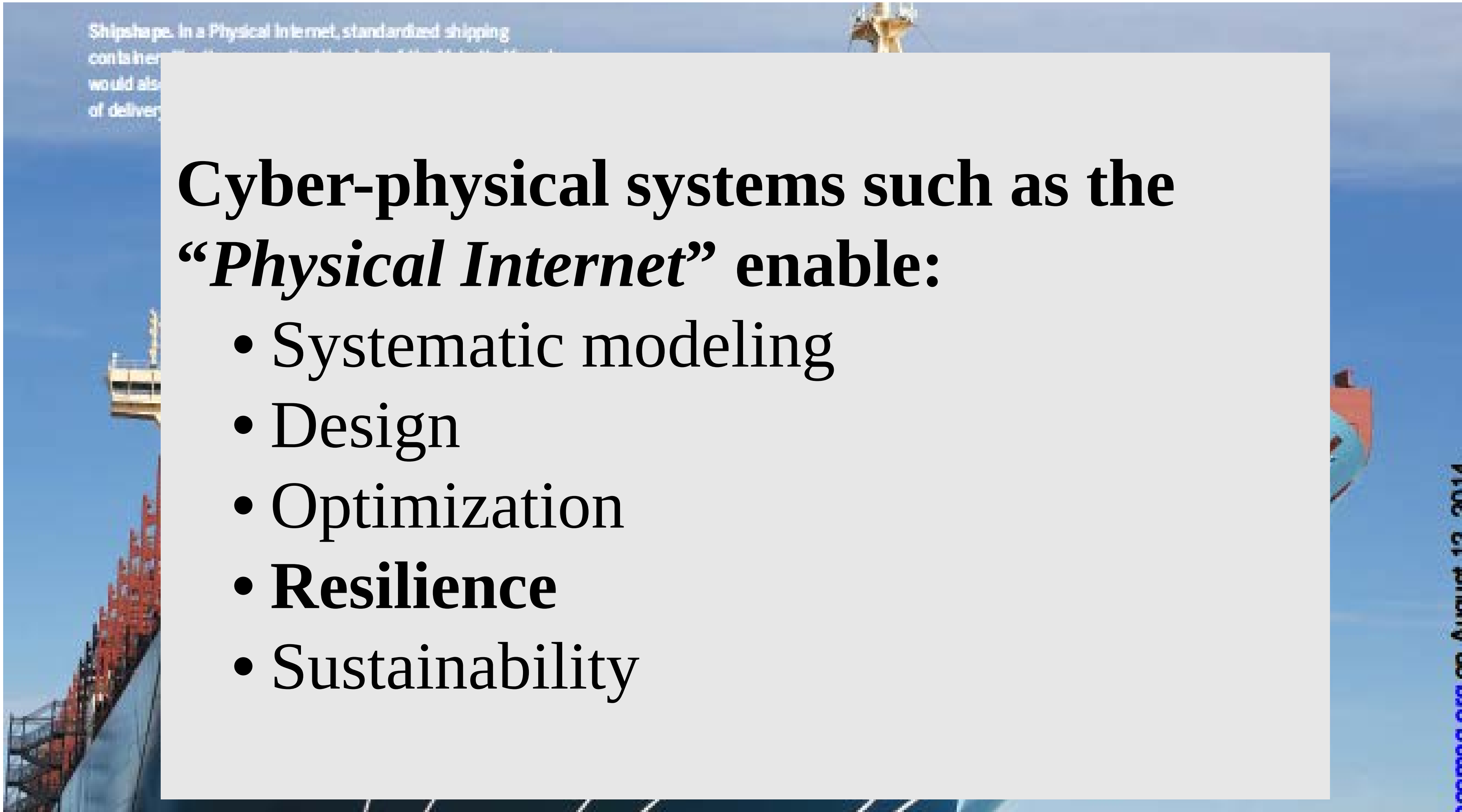
Network Design

Figure from J Mervis Science 2014;344:1104-1107

Published by AAAS

Science AAAS

The Physical Internet



Cyber-physical systems such as the "*Physical Internet*" enable:

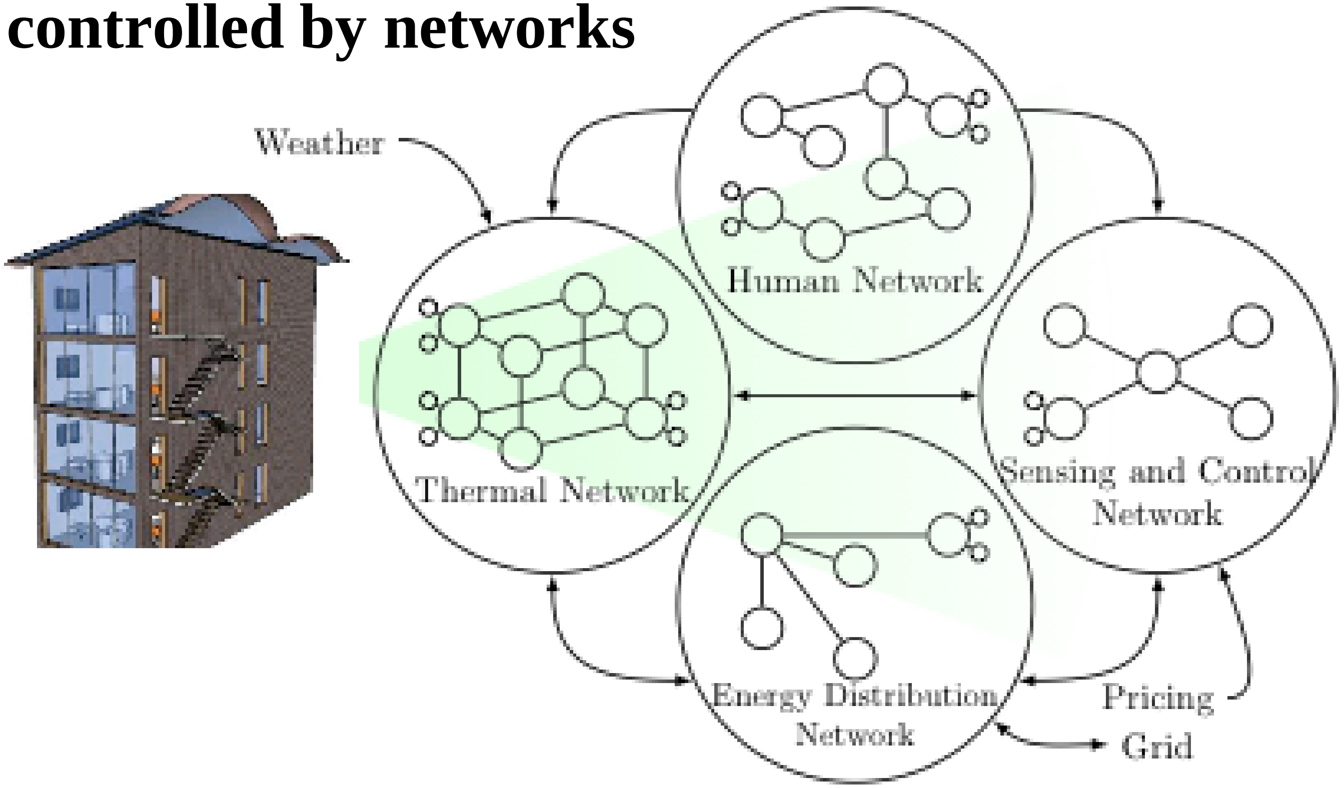
- Systematic modeling
- Design
- Optimization
- **Resilience**
- Sustainability

The information highway gets physical

The Physical Internet would move goods the way its namesake moves data

Synopsis -3

- Physical processes interacting with other sources of energy, material, and information suggests the interpretation:
  - **Complex infrastructure systems are networks controlled by networks**



Synopsis -4

- **In this talk**:
  - Apply “networks controlled by networks” idea to resilience
- **Approach**:
  - Model resilient control problem as disturbance or noise attenuation in dynamics consensus networks
- **Observation** will be:
  - Network topology matters
- **Comment** will be: future research needs to explore relationships between
  - Control-theoretic concepts
  - Graph-theoretic properties of networks

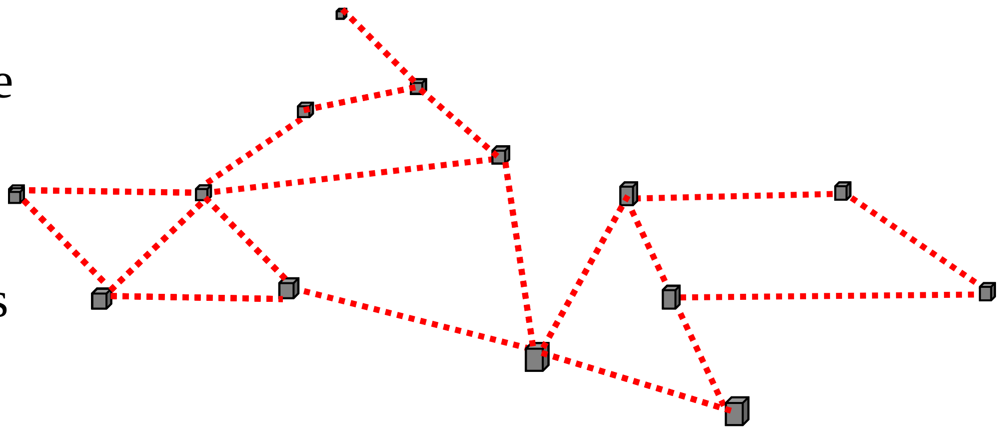
Outline

Introduction
- Systems as networks

Consensus Paradigm
- **Dynamic Networks**
- **Concepts and extensions**
- **Consensus and resilience**
- **Examples**

Resilient Dynamic Networks though Disturbance Attenuation
- Designing network weights
- Designing network controllers

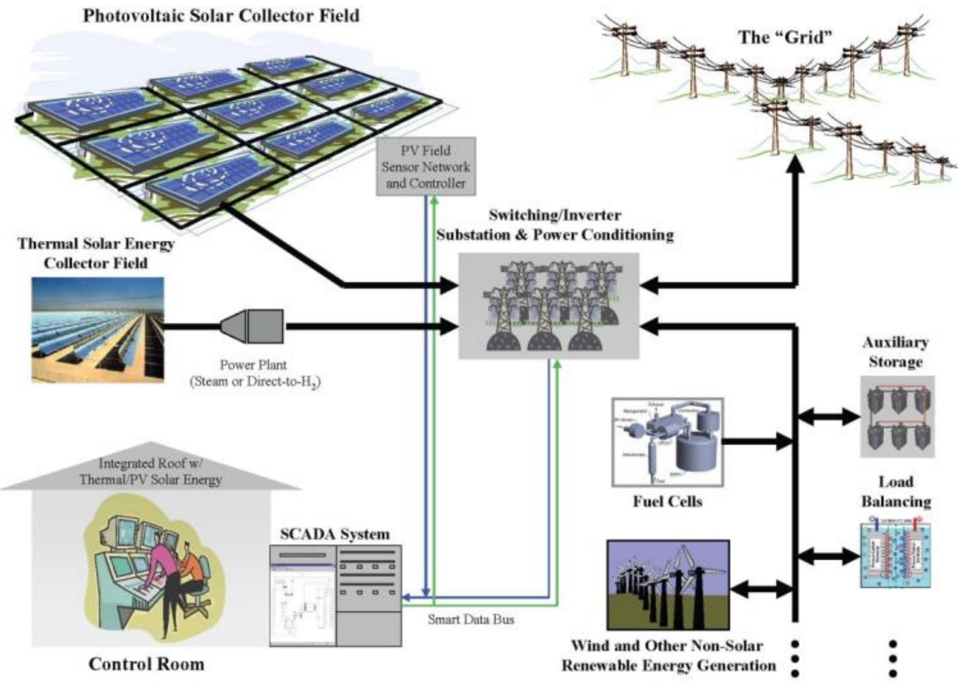
Dynamic Networks



- **Network of "entities"**
  - Communication infrastructure
  - Entity-level functionality
  - Implied global functionality
  - Not necessarily homogeneous
- **Nodes:**
  - Entities could be sensors
  - Entities could be actors (actuators)
  - Entities could be people
- **Dynamic**
  - Entities may or may not be mobile
  - Communication topology might be time-varying
  - Data actively and deliberately shared among entities
  - Decision-making and learning
  - Links between entities might be dynamic systems

Dynamic Networks as Models for …

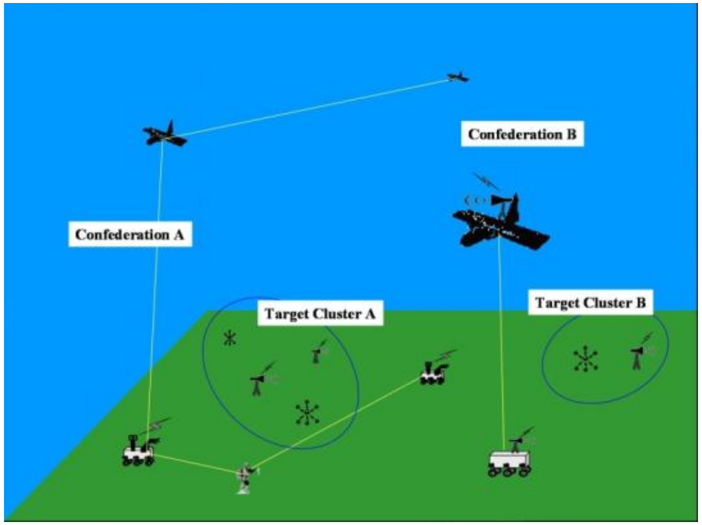
- **Many systems of interest:**
  - Cooperating robots
  - Buildings, cities
  - Power systems
  - Water distribution
  - Information networks
  - Socio-economic systems
  - …. many more ….

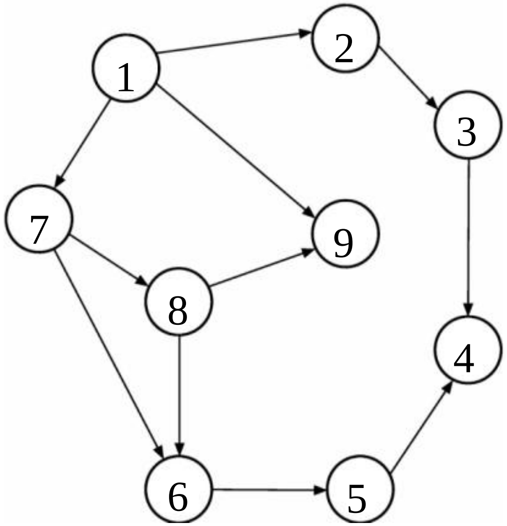


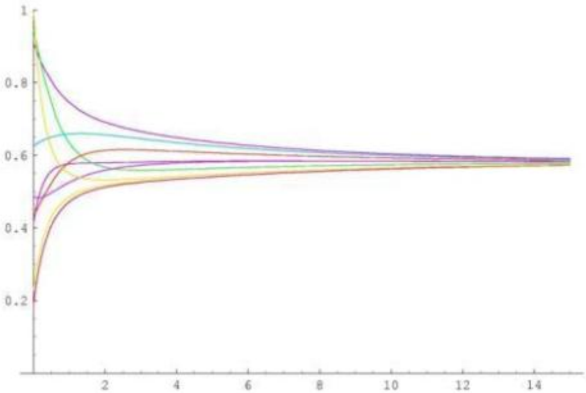
- Need a **framework for analysis and design** of these networks
  - One useful paradigm is the **consensus variable** approach

Consensus: an Algorithmic Approach to Coordination and Control in Networks

- The **consensus variable paradigm** is a generalization of potential field approaches and has connections to problems in:
  - Coupled-oscillator synchronization
  - Neural networks
- Also called agreement protocol
- Related to gossip algorithms
- Articulated in context of team theory in 1960s







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:**
  - Planning date and time and place of a meeting
  - Frequency control in power grid
  - Adaptive scheduling of mission timings

Consensus Variables

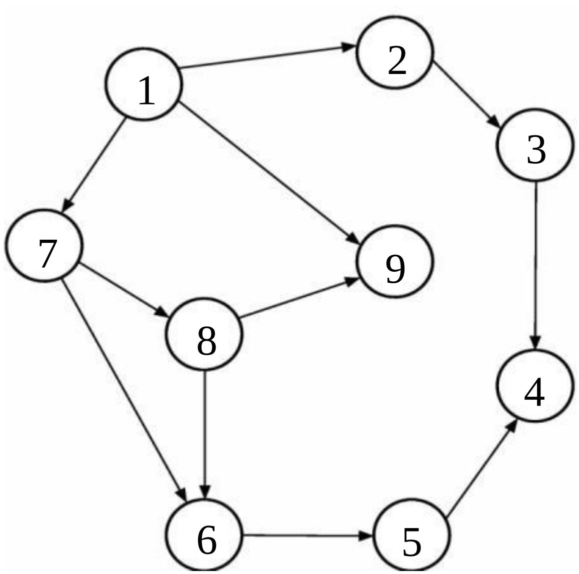
- Suppose we have *N* agents with a shared *global* consensus variable ξ
- Each agent has a *local* value of the variable given as ξ_i
- Ea **Change in value** eir local value based on the values of the agents that they can communicate wi **Difference with neighbors**

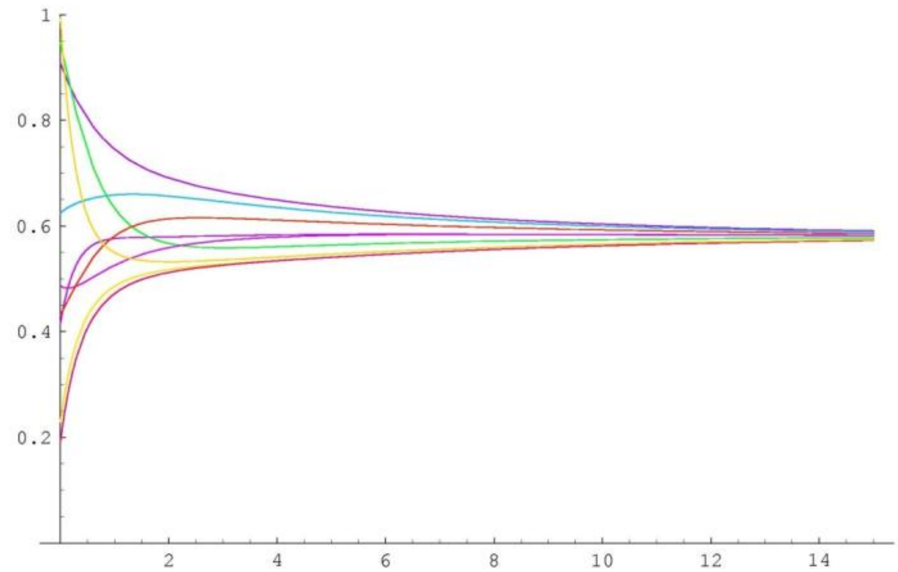
$$\dot{\xi}_i(t) = -\sum_{j=1}^{N} k_{ij}(t) G_{ij}(t) (\xi_i(t) - \xi_j(t))$$

where k_{ij} are gains and G_{ij} defines the communication topology graph of the system of agents

- **Key result** from literature: If the corresponding graph has a spanning tree then $\xi_i \longrightarrow \xi^*$ for all *i*

Example: Single Consensus Variable





$$
\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \\ \dot{\xi}_5 \\ \dot{\xi}_6 \\ \dot{\xi}_7 \\ \dot{\xi}_8 \\ \dot{\xi}_9 \end{pmatrix}
=
\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
k_{21} & -k_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & k_{32} & -k_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & k_{43} & -k_{43}-k_{45} & k_{54} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -k_{56} & k_{56} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -k_{67}-k_{68} & k_{67} & k_{68} & 0 \\
k_{71} & 0 & 0 & 0 & 0 & 0 & -k_{71} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & k_{87} & -k_{87} & 0 \\
k_{91} & 0 & 0 & 0 & 0 & 0 & 0 & k_{98} & -k_{91}-k_{98}
\end{bmatrix}}_{\text{Laplacian Matrix}}
\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \\ \xi_9 \end{pmatrix}
$$

Extension 1 - Forced Consensus

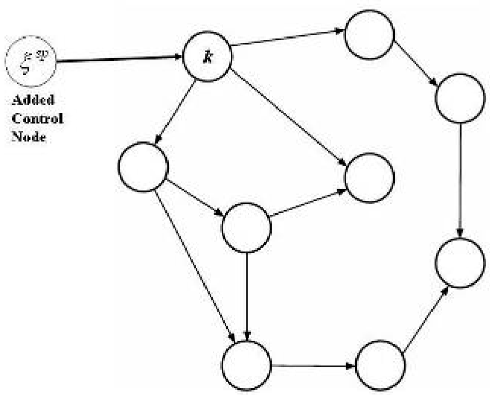
- Forced Consensus
  - Injecting an input into a node:

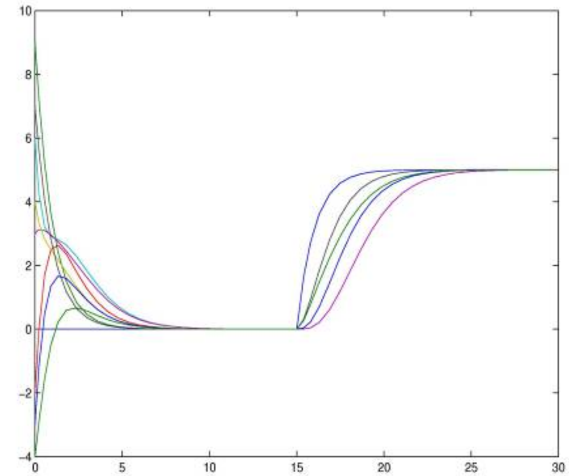
$$\dot{\xi}_i(t) = -\sum_{j=1}^{N} k_{ij}(t) G_{ij}(t)(\xi_i(t) - \xi_j(t)) + b_i u_i$$

  - Then we use a feedback controller:

$$u_i(t) = k_p(\xi^{sp} - \xi_i)$$

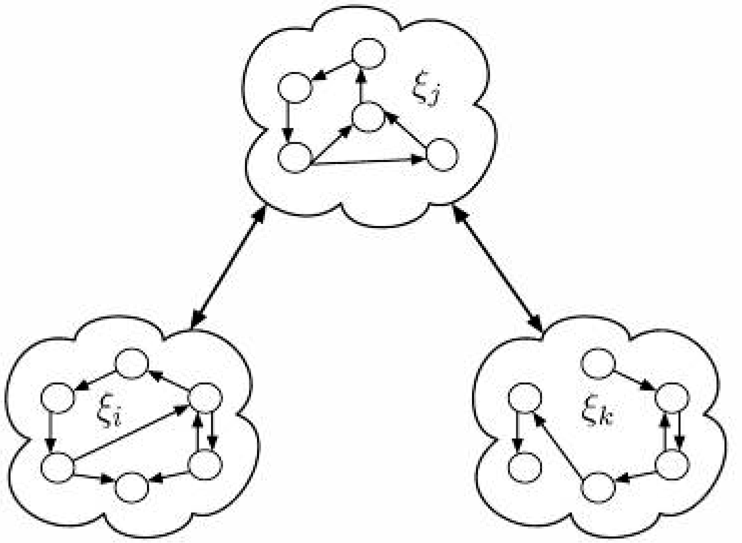
- Example:





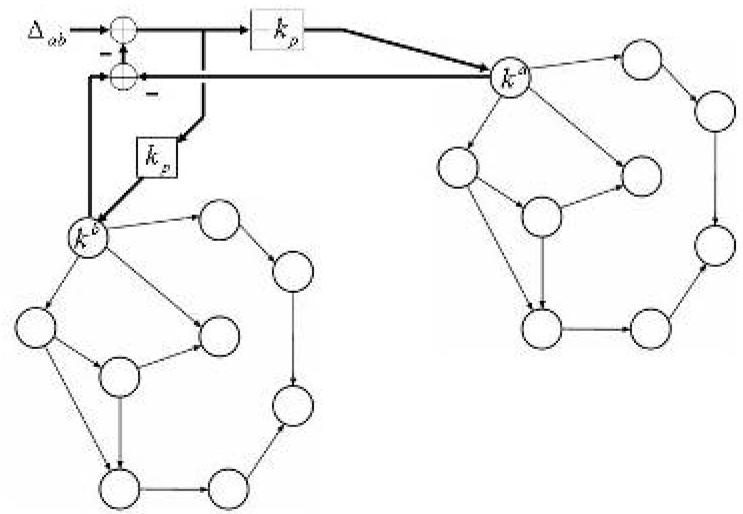
Extension 2 – Multiple, Constrained Consensus

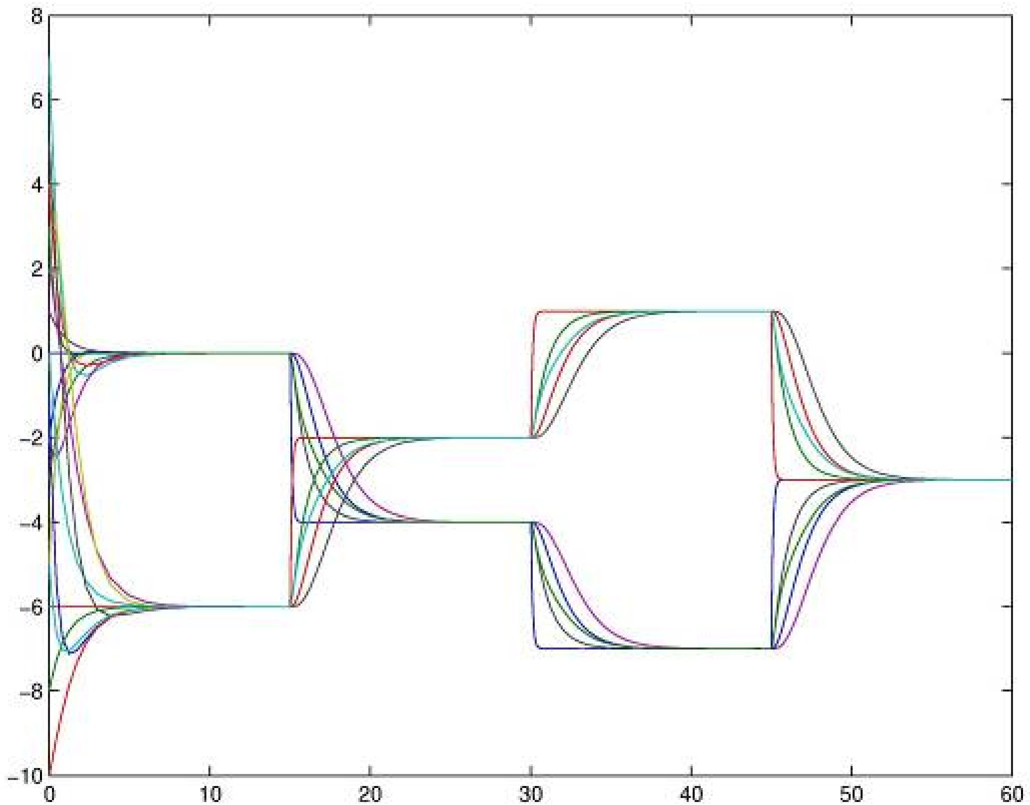
- Often we will have multiple consensus variables in a given problem



- 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

Example – Multiple, Constrained Consensus





Extension 3 – Higher-Order Consensus

- **Example: Flocking and Formation Flight**
- Consider a third-order consensus problem, applied to a formation control problem with five vehicles
- One vehicle has acceleration setpoint input and is the leader

$$\dot{x}_i = v_i$$

$$\dot{v}_i = a_i$$

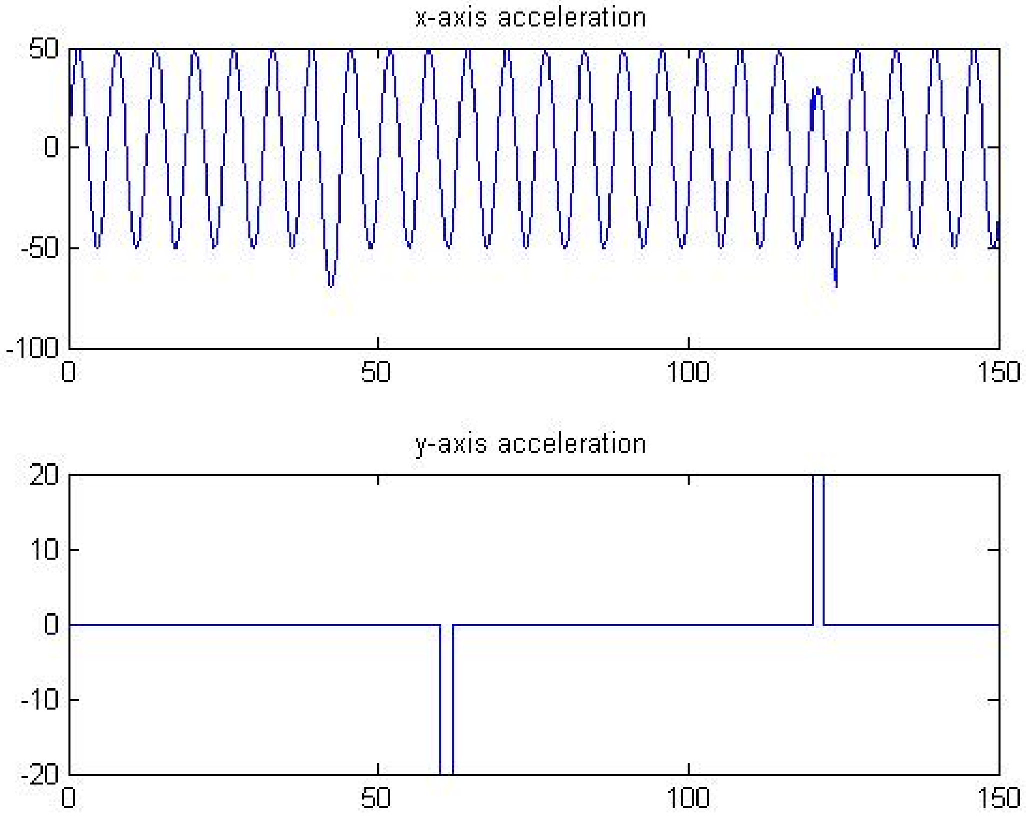
$$\dot{a}_i = -\sum_{j=1}^{n} g_{ij} k_{ij} \{ \gamma_0 [(x_i - \delta_i) - (x_j - \delta_j)] + \gamma_1 (v_i - v_j) + \gamma_2 (a_i - a_j) \} - \alpha (a_i - a_i^*)$$

Enables formation control

Acceleration Input

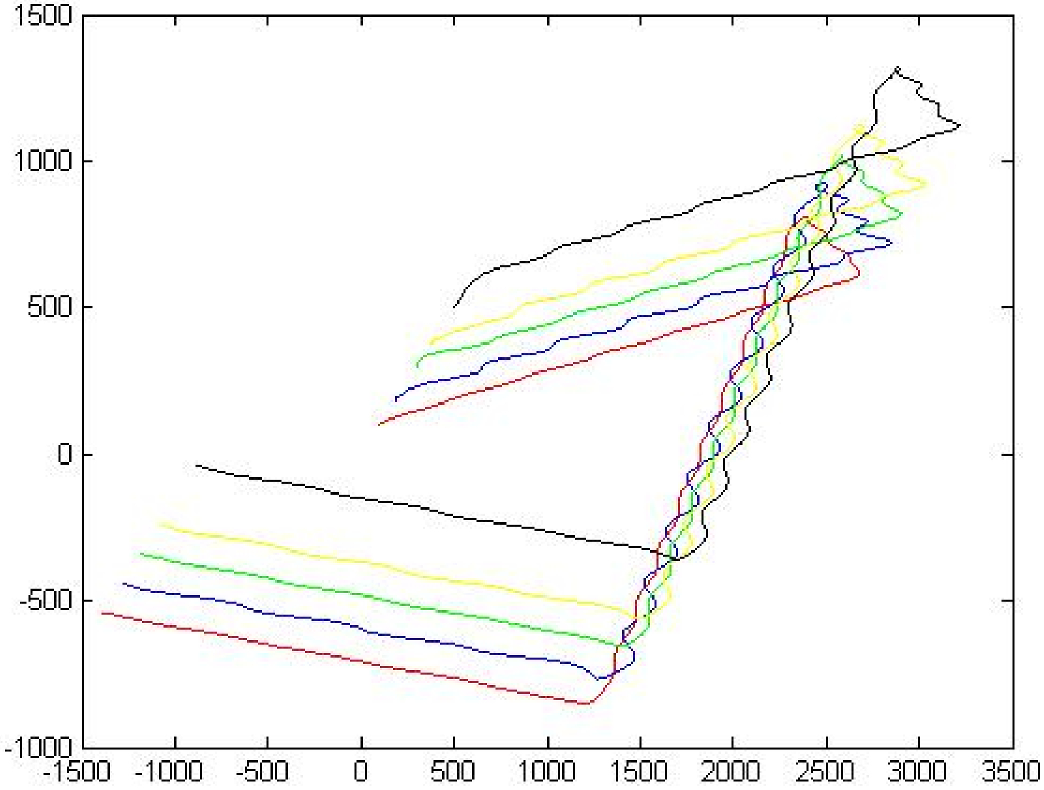
Extension 3 – Higher-Order Consensus

- The "leader node" sees the following acceleration input profile:



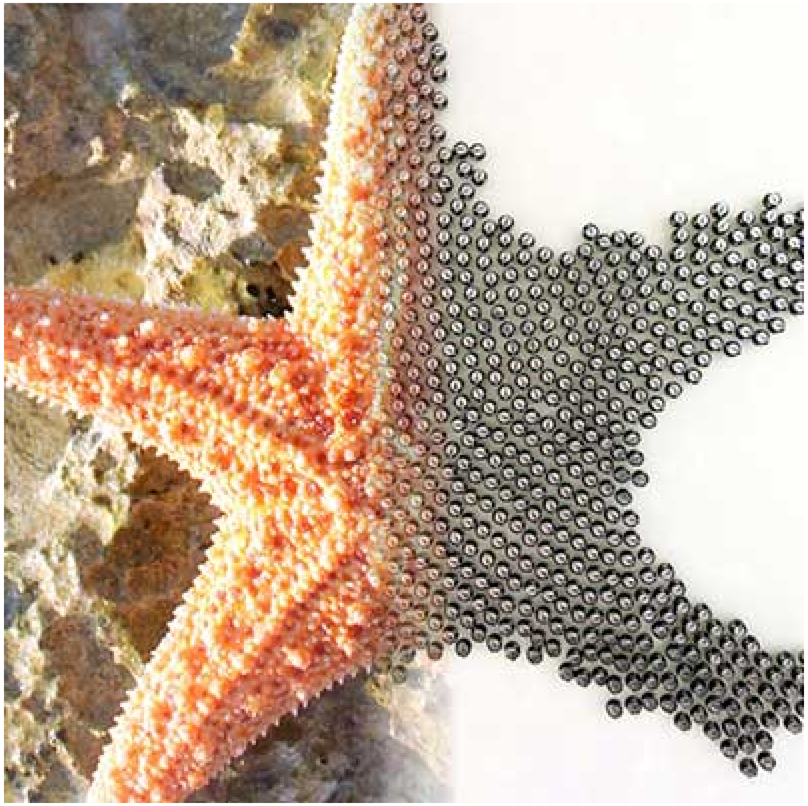
Extension 3 – Higher-Order Consensus

- The resulting paths look like:

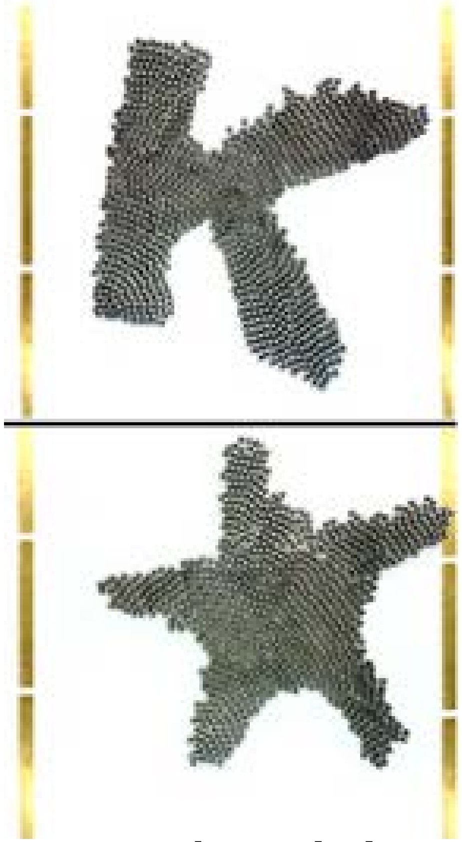


Consensus for Real -1: Harvard 1000-Robot Swarm

- *Science* 15 August 2014: Vol. 345 no. 6198 pp. 795-799



http://www.seas.harvard.edu/sites
/default/files/images/news/Image1_sq_0_op.jpg

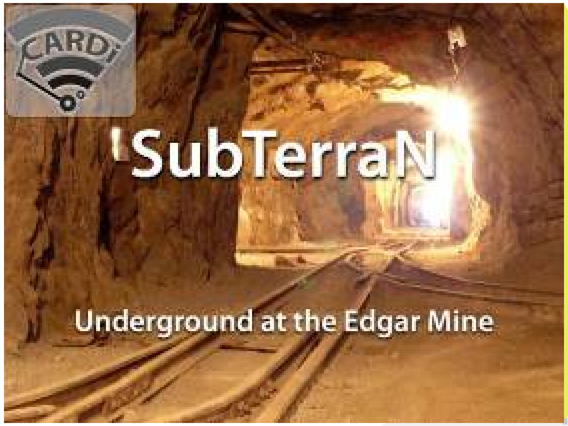


http://www.seas.harvard.edu/sites/default/fil
es/images/news/Image2_650.jpg

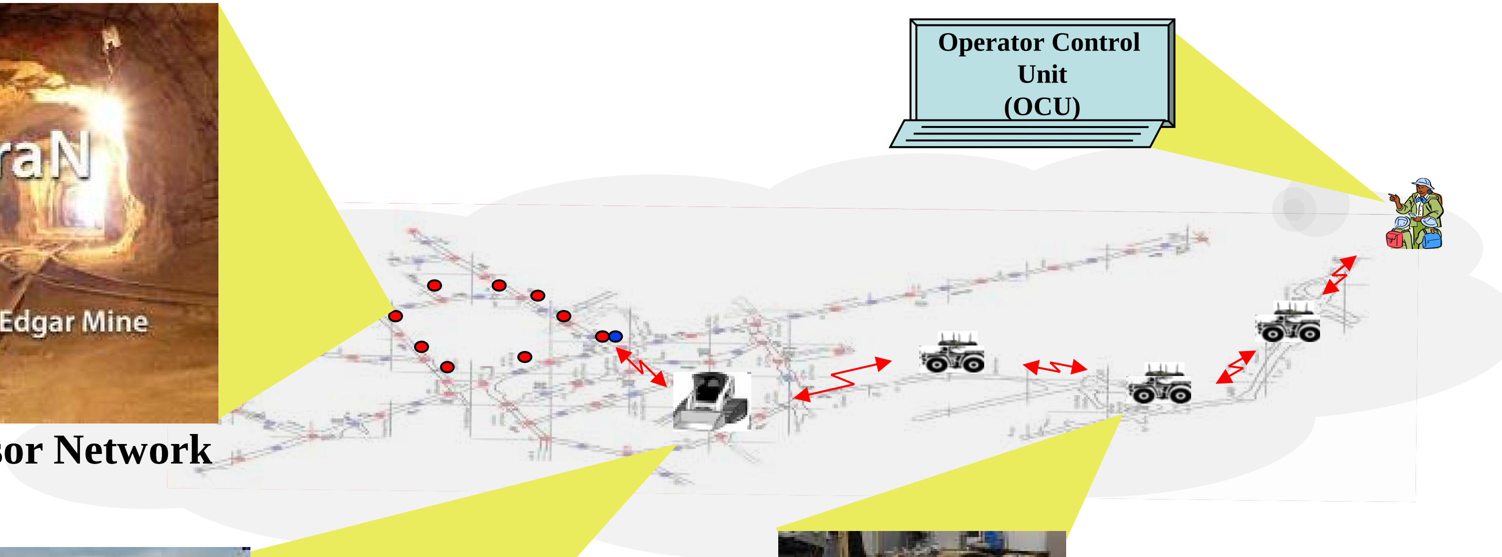
Consensus for Real -2: Radio Tethering in Subterranean Environments

- Subways, mines, caves, underground buildings
- Limited
  - Entrances/exits
  - Navigation
  - Limited ventilation
  - Communications
- Challenging emergency management environment
  - Assume no infrastructure
  - Radio relays may be necessary at/near junctions
  - Rescue workers must carry their own comms

MineSENTRY - Autonomous Mobile Radio Relays



Underground Sensor Network



Teleoperated Bobcat

Mesh Radio System
(Rajant Breadcrumb™)

Autonomous Radio Node (AMR)

- Provides Communication Tether
- Uses CSM-developed UGV Autopilot

Theoretical Approach

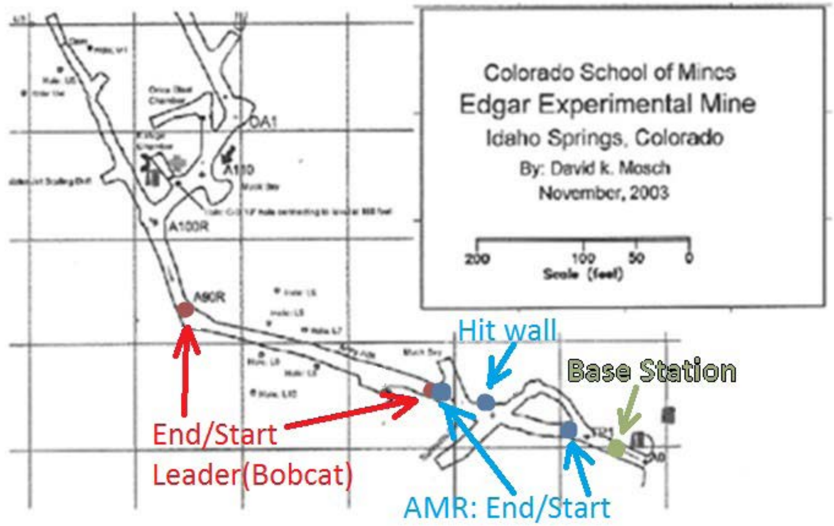
- Wireless 1-D tethering
  - Not in physical coordinates
  - Rather, in radio signal strength (RSS) space
- Goal is to maintain equal RSS between ARMs while the leader moves forward in the mine
- Our approach uses the **internal model principle to develop a higher-order (2nd) consensus algorithm**:

$$\ddot{x}_i = -k_i^p RSS^{dB}{}_{i+1,i} + k_i^p RSS^{dB}{}_{i,i-1} - k_i^d \frac{d}{dt}(RSS^{dB}{}_{i+1,i}) + k_i^d \frac{d}{dt}(RSS^{dB}{}_{i,i-1})$$

Experimental Results

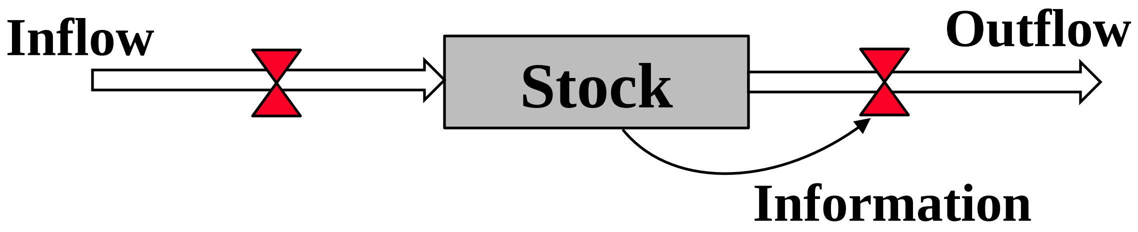
MineSENTRY

Mine Safety and Rescue through Sensing Networks and Robotics Technology



Consensus Networks and Resilience -1

- Dynamic consensus networks give a reasonable paradigm for modeling systems that have
  - Storage and computation at nodes
  - Flows between edges and along edges
- One possible way to see this is to consider the ideas from Jay Forrester's (MIT) System Dynamics paradigm
  - Basically a "poor man's control theory"
  - Envisioned all systems as having "stocks" and "flows" that are interconnected through positive and negative feedback



Consensus Networks and Resilience -2

- Stocks and flows and some other components can be assembled to build up complex systems models
- These models can be simulated using tools such as STELLA

Example: Public Policy Process

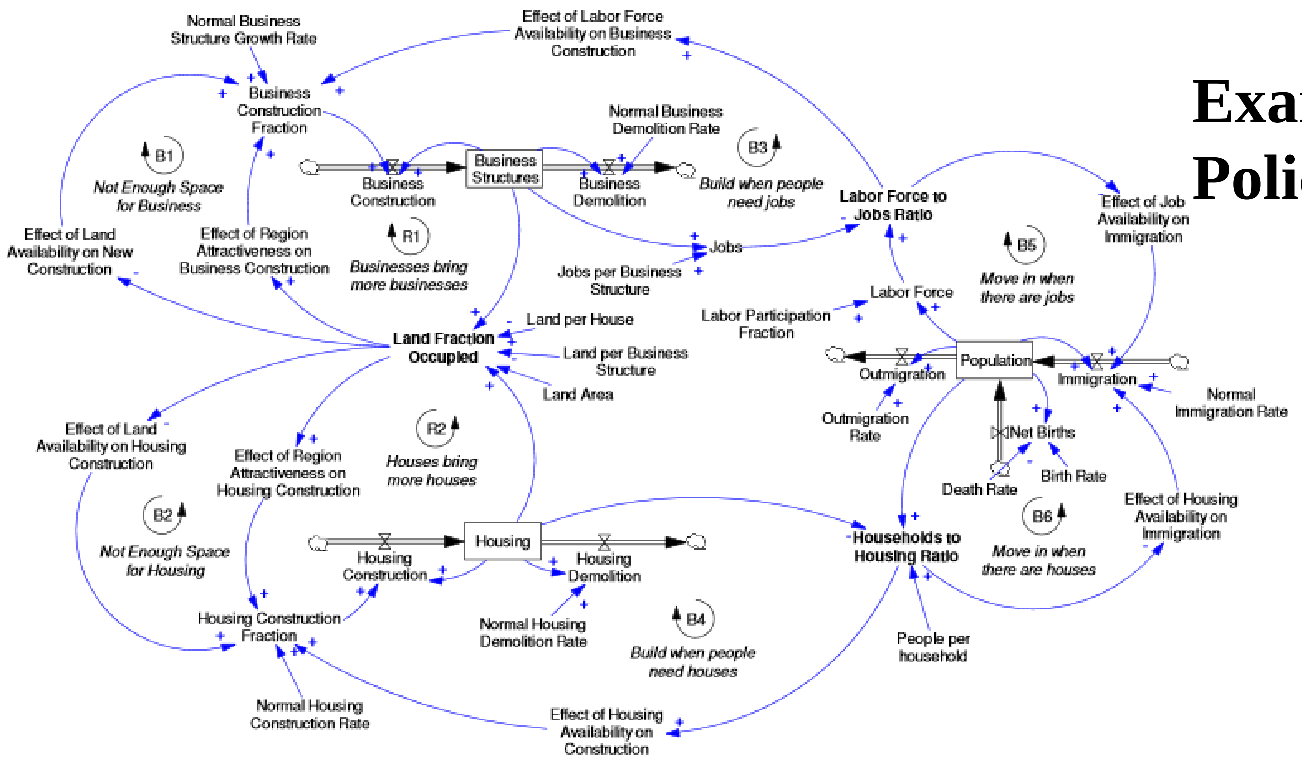
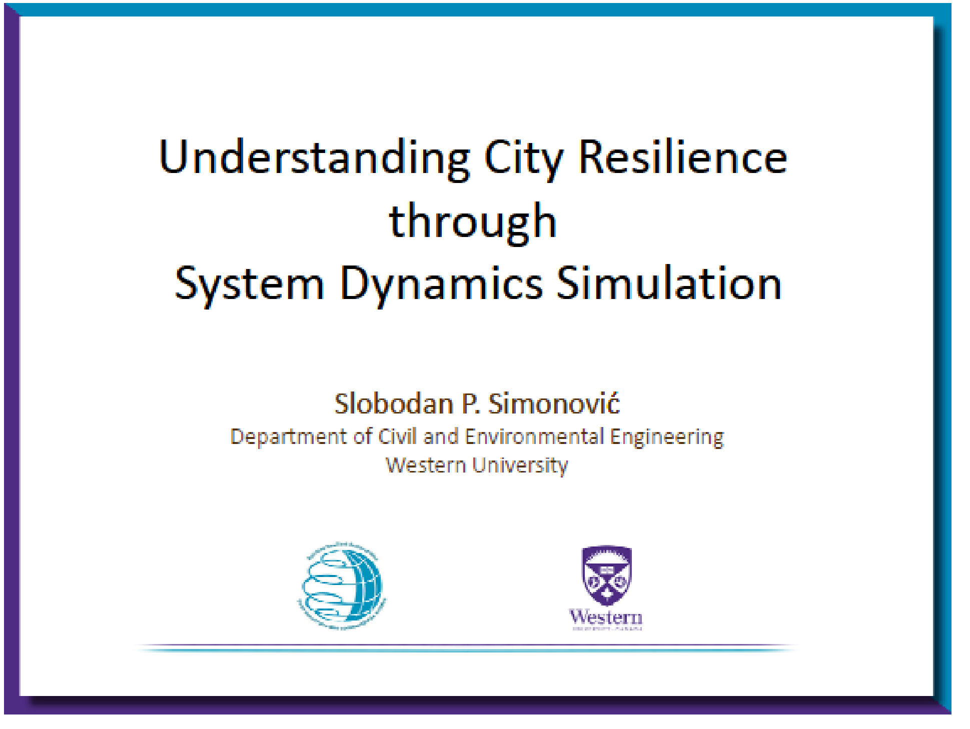


Figure: From "How Small System Dynamics Models Can Help the Public Policy Process," http://www.albany.edu/~gpr/SmallModels.pdf. Used without permission.

Consensus Networks and Resilience -3

- People have used these ideas to study resilience and sustainability. For example:



2012 Advanced Institute, Taipei
Slobodan P. Simonović

Consensus Networks and Resilience -4

- Consensus paradigm provides an analytical tool for analysis of systems modeled using Forrester's System Dynamics

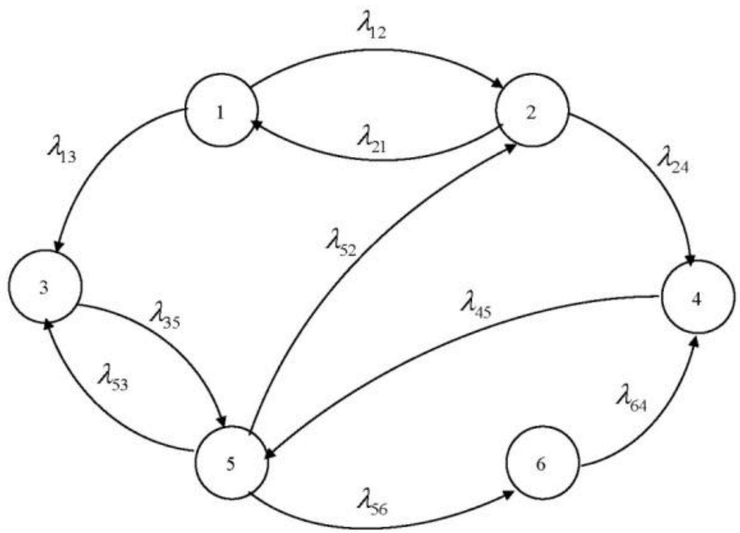
- Key idea is that it works for systems where

$$\textit{change in storage} \propto \sum \; (\textit{multipliers}) \; x \; (\textit{flows in} - \textit{flows out})$$

- Below we will illustrate this for several systems:
  - Cooperating robots (discussed above)
  - Thermal systems (e.g., buildings)
  - Electric circuits (e.g., power systems)

Static Graphs and Laplacian Matrix

- The neighbors of node n_i are $\mathcal{N}_i = \{j : (n_i, n_j) \in \mathcal{E}\}$
- The Laplacian matrix $L = [l_{ij}]$ is defined by:



$$l_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} \lambda_{ik} & i = j \\ -\lambda_{ij} & i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

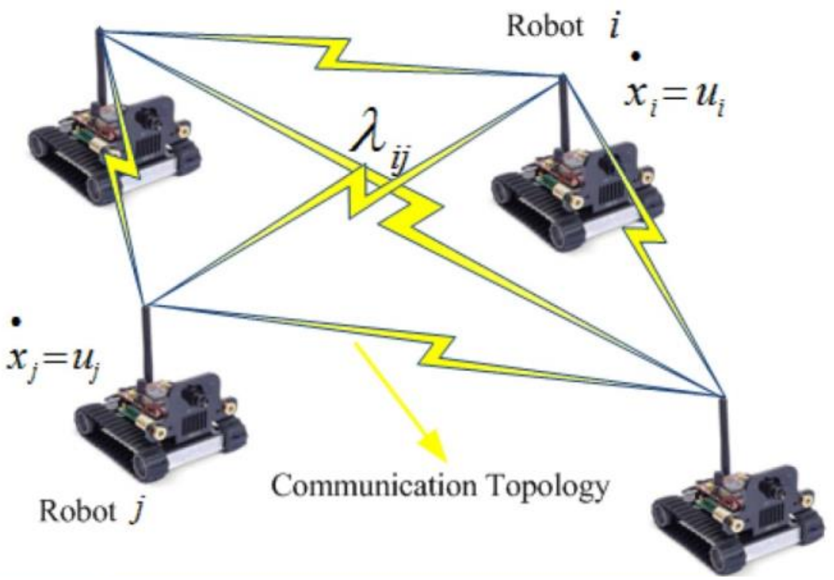
$$L = \begin{bmatrix} \lambda_{12} + \lambda_{13} & -\lambda_{12} & -\lambda_{13} & 0 & 0 & 0 \\ -\lambda_{21} & \lambda_{21} + \lambda_{24} & 0 & -\lambda_{24} & 0 & 0 \\ 0 & 0 & \lambda_{35} & 0 & -\lambda_{35} & 0 \\ 0 & 0 & 0 & \lambda_{45} & -\lambda_{45} & 0 \\ 0 & -\lambda_{52} & -\lambda_{53} & 0 & \lambda_{52} + \lambda_{53} + \lambda_{56} & -\lambda_{56} \\ 0 & 0 & 0 & -\lambda_{64} & 0 & \lambda_{64} \end{bmatrix}$$

Static Consensus Protocols

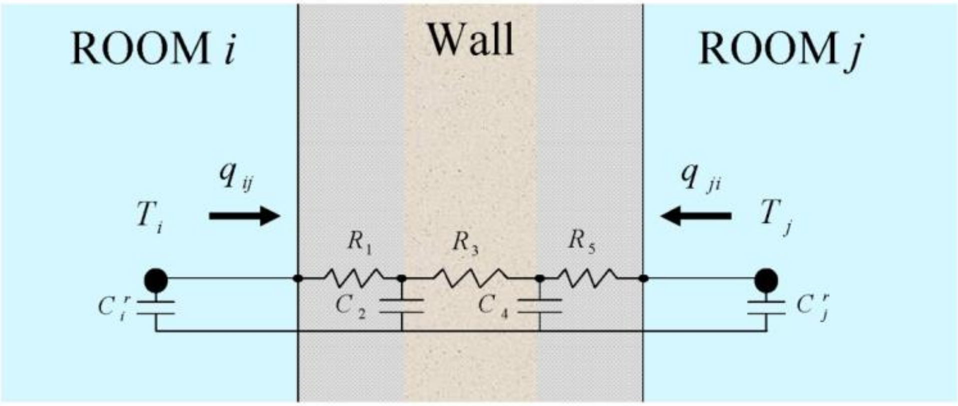
- Consider 4 robots with the velocity $\dot{x}_i = u_i$ and interconnected using the following static consensus protocol:

$$\dot{x}_i = u_i = -\sum_{j \in \mathcal{N}_i} [\lambda_{ij}(x_i - x_j)].$$

Then $$\dot{X} = -LX,$$
where L is the static Laplacian associated with the communication topology of the 4 robots graph.



Two Rooms Modeled as Two Interconnected Nodes

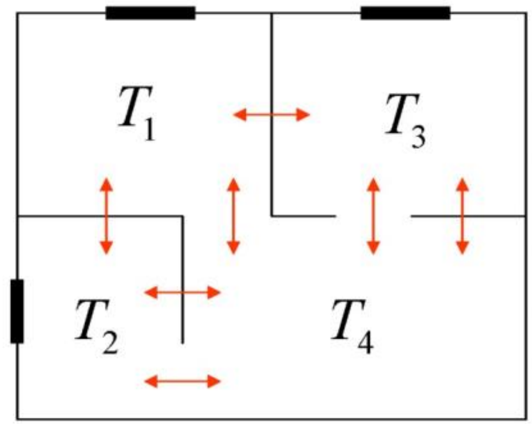


- The node equation can be written as: $C_i^r \frac{dT_i}{dt} = q_i^{in} - q_{ij}(t)$
- The heat flows can be written as:

$$\begin{bmatrix} q_{ij} \\ q_{ji} \end{bmatrix} = \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix}$$

- $A_{ij}, A_{ji}, B_{ij}, D_{ij}$ are transfer functions (poylnomials in s), representing physical dynamics (ultimately described by a differential equation)

Hypothetical Four-Room Example



- For each room use

$$C_i^r \frac{dT_i}{dt} = q_i^{in}(t) - \sum_{j \in \mathcal{N}_i} q_{ij}(t)$$

or

$$sT_i(s) = \frac{1}{C_i^r} Q_i^{in}(s) - \sum_{j \in \mathcal{N}_i} [\lambda_{ij}^S(s) T_i(s) - \lambda_{ij}^C(s) T_j(s)]$$

- Define the vectors

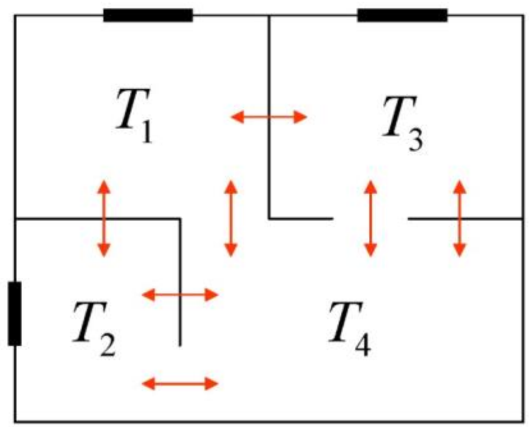
$$T(s) = \begin{bmatrix} T_1(s) & T_2(s) & T_3(s) & T_4(s) \end{bmatrix}^T$$

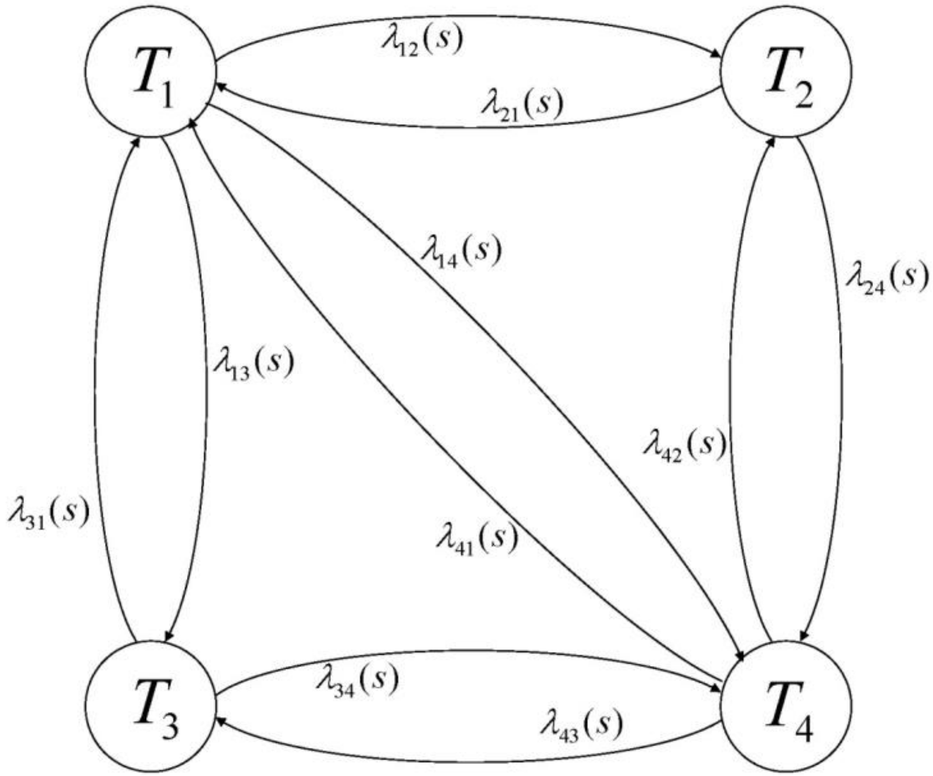
$$Q^{in}(s) = \begin{bmatrix} Q_1^{in}(s) & Q_2^{in}(s) & Q_3^{in}(s) & Q_4^{in}(s) \end{bmatrix}^T$$

- Then we can write $T(s) = \frac{1}{s}[Q^{in}(s) - L(s)T(s)]$, where

$$L(s) \quad = \quad \underbrace{\begin{bmatrix} \sum\limits_{j=2,3} \lambda_{1j}^S(s) & -\lambda_{12}^C(s) & -\lambda_{13}^C(s) & 0 & 0 & 0 \\ -\lambda_{21}^C(s) & \sum\limits_{j=1,4} \lambda_{2j}^S(s) & 0 & -\lambda_{24}^C(s) & 0 & 0 \\ 0 & 0 & \lambda_{35}^S(s) & 0 & -\lambda_{35}^C(s) & 0 \\ 0 & 0 & 0 & \lambda_{45}^S(s) & -\lambda_{45}^C(s) & 0 \\ 0 & -\lambda_{52}^C(s) & -\lambda_{53}^C(s) & 0 & \sum\limits_{j=2,3,6} \lambda_{5j}^S(s) & -\lambda_{56}^C(s) \\ 0 & 0 & 0 & -\lambda_{64}^C(s) & 0 & \lambda_{64}^S(s) \end{bmatrix}}_{\text{Dynamic Laplacian Matrix}}$$

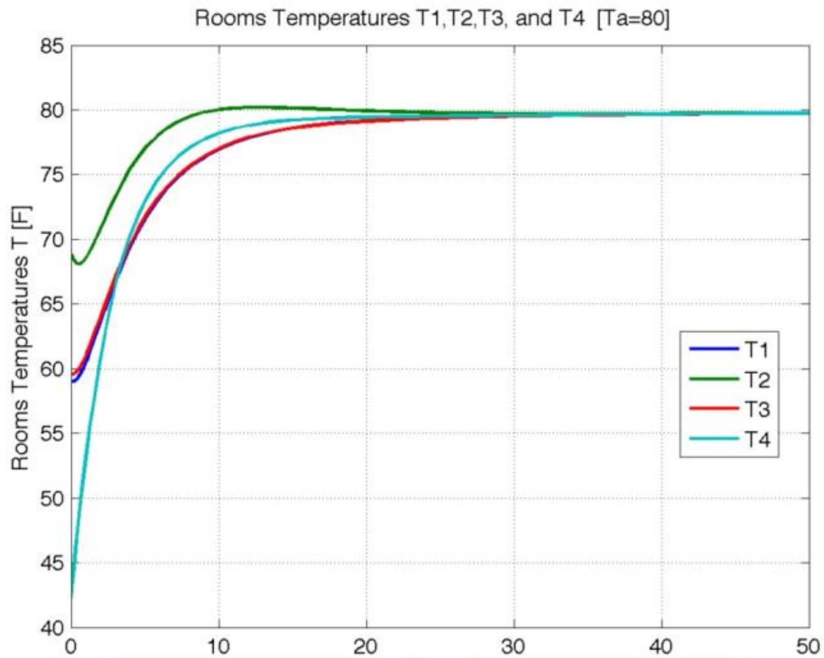
Hypothetical Four-Room Example as a Graph





Simulation: Hypothetical Four-Room Example

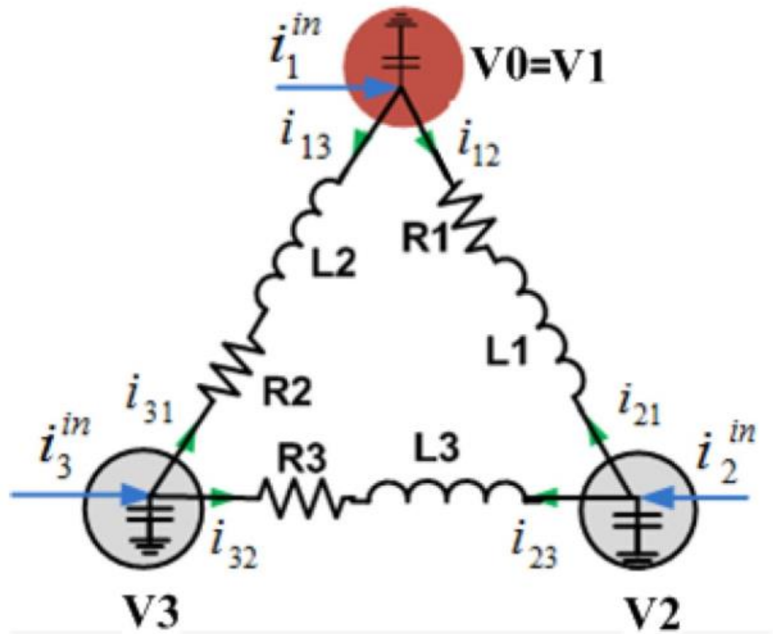
- Consider case with: (1) an outside node representing ambient conditions (dynamic consensus with a leader), with $T_a = 80$; (2) no other input energy; and (3)rooms initially set to arbitrary temperatures (less than ambient)
- As expected, all temperatures converge to the ambient temperature



Electrical Network as an Undirected Dynamic Consensus Network

- The dynamic model of each node is:

$$sV_i(s) = I_i^{in}(s) - \sum_{j \in \mathcal{N}_i} [Y_{ij}(s)(V_i(s) - V_j(s))]$$



- The dynamic consensus protocol:

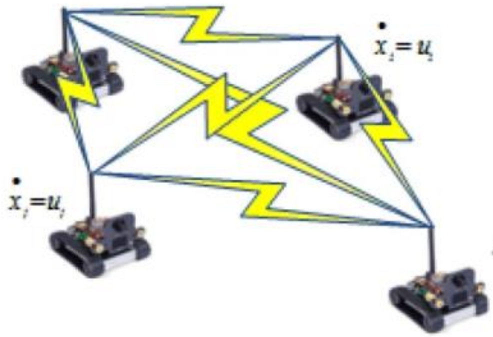
$$V_i(s) = -\frac{1}{s} \sum_{j \in \mathcal{N}_i} [Y_{ij}(s)(V_i(s) - V_j(s))]$$

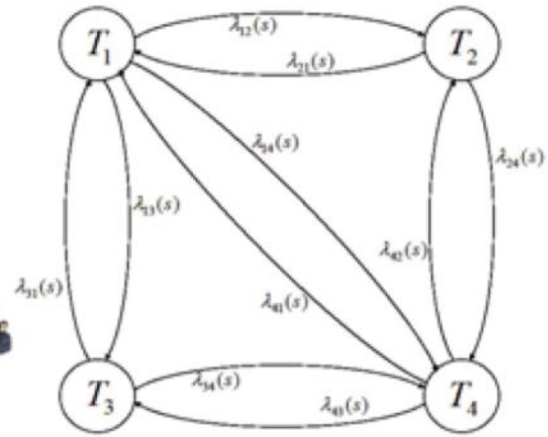
- The overall system:

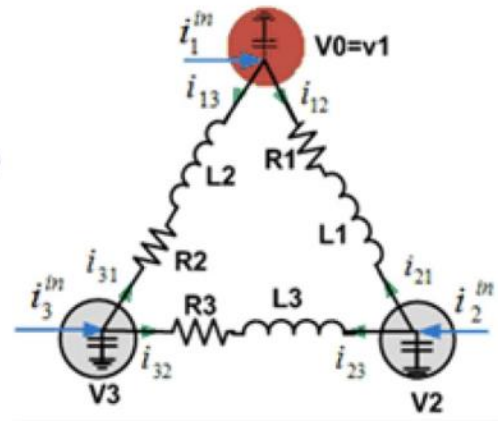
$$sV(s) = -L(s)V(s)$$

$$L(s) = \begin{bmatrix} Y_{12}(s) + Y_{13}(s) & -Y_{12}(s) & -Y_{13}(s) \\ -Y_{12}(s) & Y_{12}(s) + Y_{23}(s) & -Y_{23}(s) \\ -Y_{13}(s) & -Y_{23}(s) & Y_{13}(s) + Y_{23}(s) \end{bmatrix}$$

Comparing These Examples







Robot network: $$sX(s) = -LX(s)$$

Thermal network: $$sT(s) = -L(s)T(s)$$

Electrical network: $$sV(s) = -L(s)V(s)$$

Static consensus protocol:

$$x_i(s) = -\frac{1}{s}\sum_{j\in\mathcal{N}_i}[\lambda_{ij}(x_i(s) - x_j(s))]$$

Dynamic consensus protocol:

$$x_i(s) = -\frac{1}{s}\sum_{j\in\mathcal{N}_i}[\lambda_{ij}(s)(x_i(s) - x_j(s))]$$

Consensus conditions: Under some conditions on L and $L(s)$,

$$x_i(t) \to x^* \qquad T_i(t) \to T^* \qquad v_i(t) \to v^*$$

Outline

Introduction

- Systems as networks

Consensus Paradigm

- Dynamic Networks
- Concepts and extensions
- Consensus and Resilience
- Examples

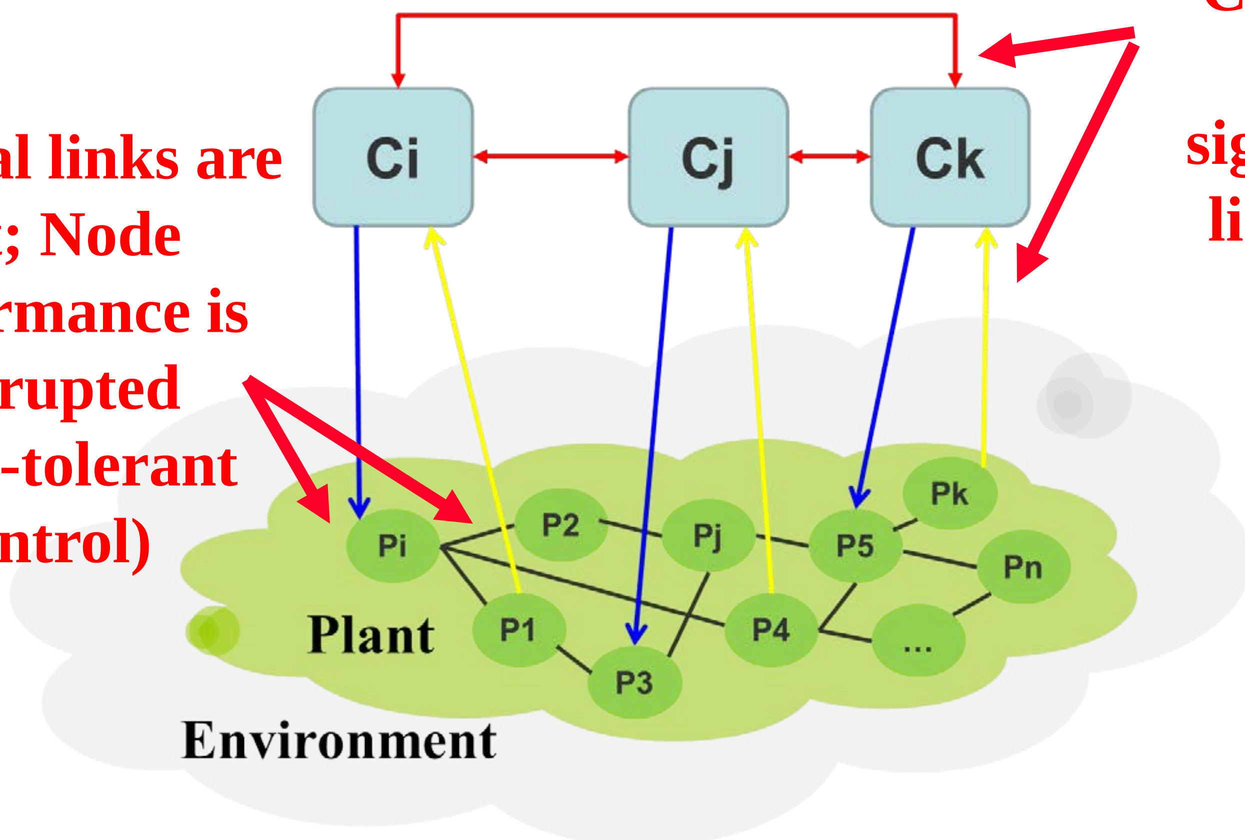
Resilient Dynamic Networks though Disturbance Attenuation

- **Designing network weights**
- **Designing network controllers**

Resilient Control as Disturbance Attenuation -1

- Consider a consensus network that models some critical system
- What could go wrong?

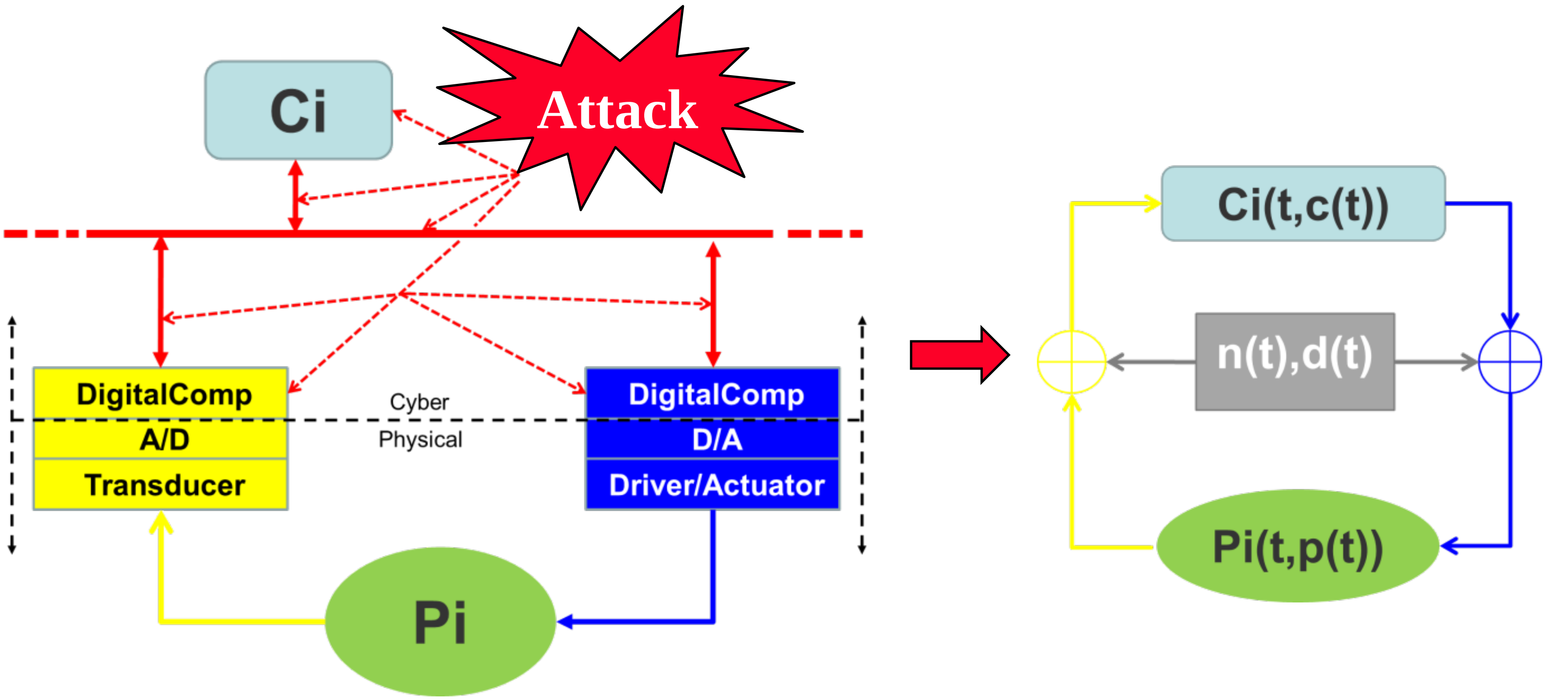
Physical links are lost; Node performance is corrupted (fault-tolerant control)



Cyber-attack corrupts signals; Cyber links are lost (robust control)

Resilient Control as Disturbance Attenuation -2

- The cyber-attack problem can be viewed as follows:



Design for Resilient Dynamic Networks

- An aspect of resilience is disturbance attenuation: keeping network as close to consensus as possible
- Will consider two disturbance attenuation problems in consensus networks:
  1. Designing network weights
  2. Designing decentralized and distributed controllers
- Will consider $\mathcal{L}_2$ disturbances with finite energy
- Also have results addressing bounded (but infinite energy) disturbances in $\mathcal{L}_\infty$

A Quick Reminder: H_∞ Norm

- For a system with an **input** $d(t)$ and an **output** $z(t)$, the H_∞ norm of the transfer function matrix from d to z ($T_{zd}(j\omega)$) is defined by:

$$\|T_{zd}(j\omega)\|_\infty = \max_{d(t)\neq 0} \frac{\|z(t)\|_2}{\|d(t)\|_2} = \max_{\omega} \sigma(T_{zd}(j\omega))$$

- We use the H_∞ norm of the $T_{zd}(j\omega)$ because it is an upper bound on the amplification of the energy in d to z:

$$\|z(t)\|_2 \leq \|T_{zd}(j\omega)\|_\infty \|d(t)\|_2 \, , \, d(t) \in \mathcal{L}_2(t)$$

1. Graph Design for Disturbance Attenuation

- Suppose our network of identical nodes is defined as

$$\begin{aligned}\dot{x}_i &= Ax_i + F\sum_{j\in\mathcal{N}_i} w_{ij}(x_j - x_i) + Ed_i, \\ z_i &= x_i - \frac{1}{N}\sum_{j=1}^{N} x_j,\end{aligned}$$

- $z_i = [z_1, z_2, \ldots, z_N]^T$ is called the "*disagreement vector*"
- $d_i = [d_1, d_2, \ldots, d_N]^T$ is the disturbance vector
- **Problem**: Pick the weights w_{ij} so that
  1. Consensus is achieved when $d_i(t) \equiv 0$
  2. $\|T_{zd}(j\omega)\|_\infty$ is as small as possible when $d_i(t) \neq 0$

Solution

Theorem

Let $\gamma > 0$ be given. The network reaches consensus when $d \equiv 0$ and $\|T_{zd}(s)\|_\infty \le \gamma$ if $\exists\ P = P^T \succ 0$ so

$$\begin{bmatrix} \Omega_2 & PE \\ E^T P & -\gamma^2 I_n \end{bmatrix} \preceq 0,$$
$$F^T P + PF \succeq 0,$$

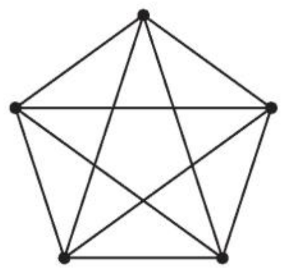
where $\Omega_2 = (A - \lambda_2 F)^T P + P(A - \lambda_2 F) + I_n$.

- λ_2 is the second smallest eigenvalue of L
- Design the weights by solving (convex optimization):

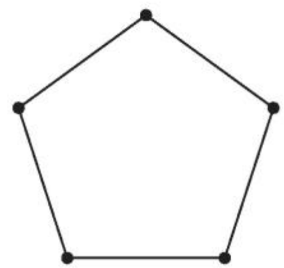
$$\begin{aligned} &\text{maximize } \lambda_2(w) \\ &\text{subject to } c^T w \le b_c \end{aligned}$$

Graph Design Example

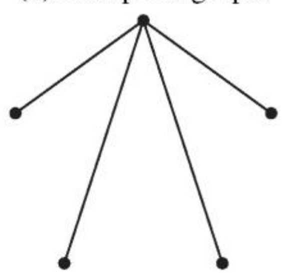
- Consider 5 different networks:



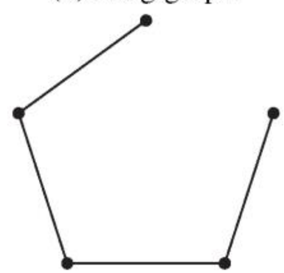
(a) Complete graph.



(b) Ring graph.



(c) Star graph.



(d) Tree graph.

- Question: what value of weights give best disturbance rejection?

Results

- Let the plant be defined by

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \; E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

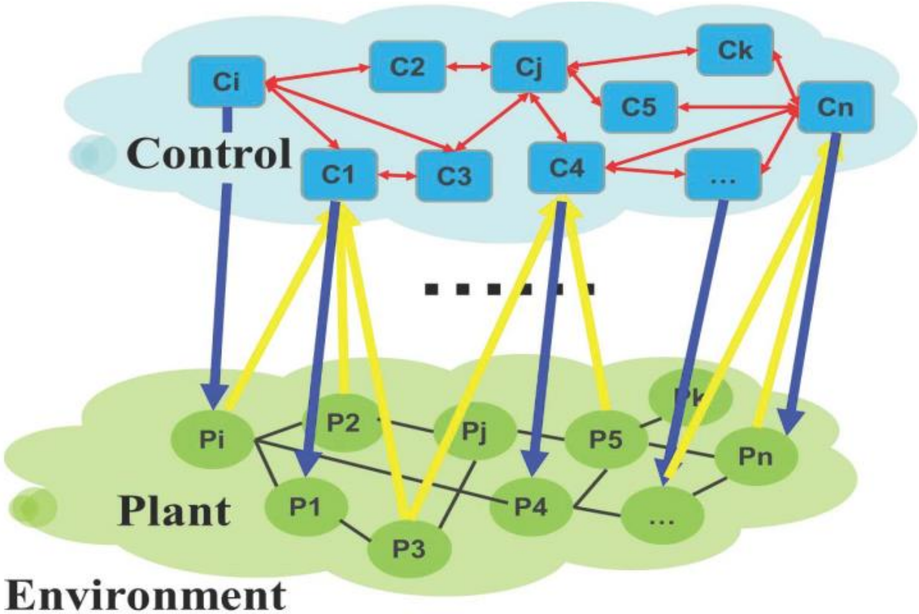
  and let the constraint be $\sum w_{ij} \leq 50$
- This produces the following

	Complete graph	Ring graph	Star graph	Tree graph
$w^\star$	$5 \times 1_{10}$	$10 \times 1_5$	$12.5 \times 1_4$	$[10\ 15\ 15\ 10]^T$
$\lambda_2^\star$	25.0	13.82	12.5	5.0
γ_{min}	0.0565	0.1021	0.1128	0.2774
$\lVert T_{zd}(s) \rVert_\infty$	0.0565	0.1021	0.1128	0.2774

- Perhaps not surprising, the complete graph is best

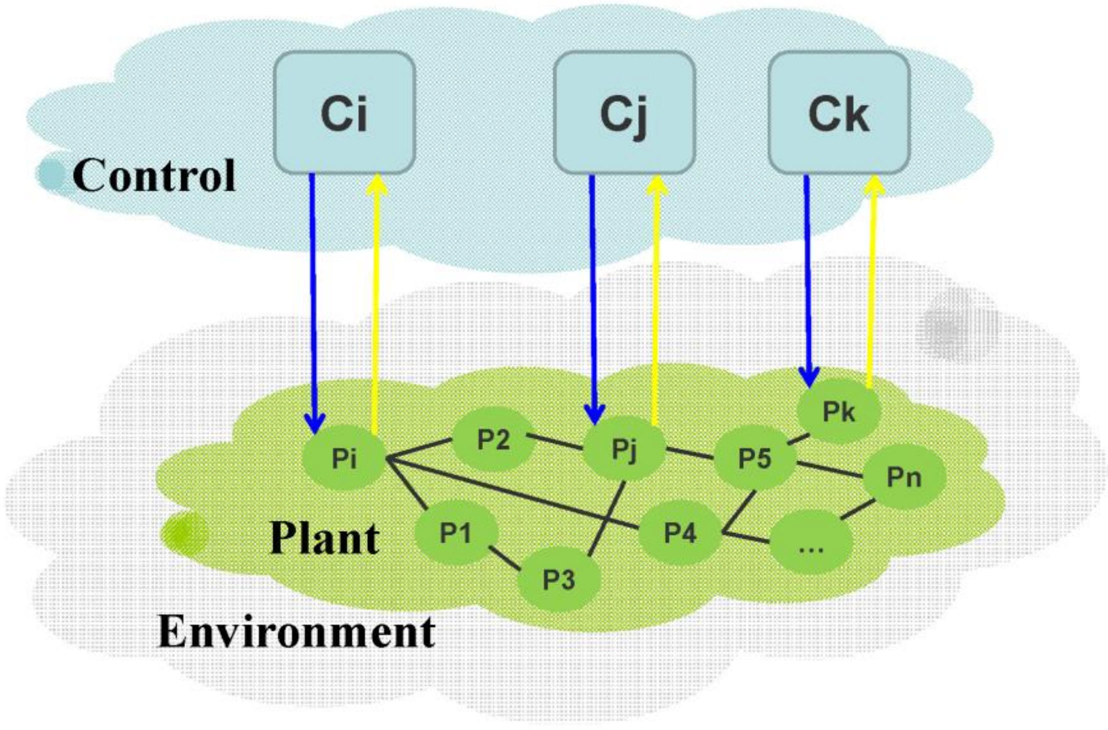
Networks Controlling Networks

- Let each node in the plant have inputs and be subject to disturbances; let the state be available as an output
- Introduce a controller network; three variants:
  - Decentralized
  - Distributed
  - Fully-interconnected (shown)



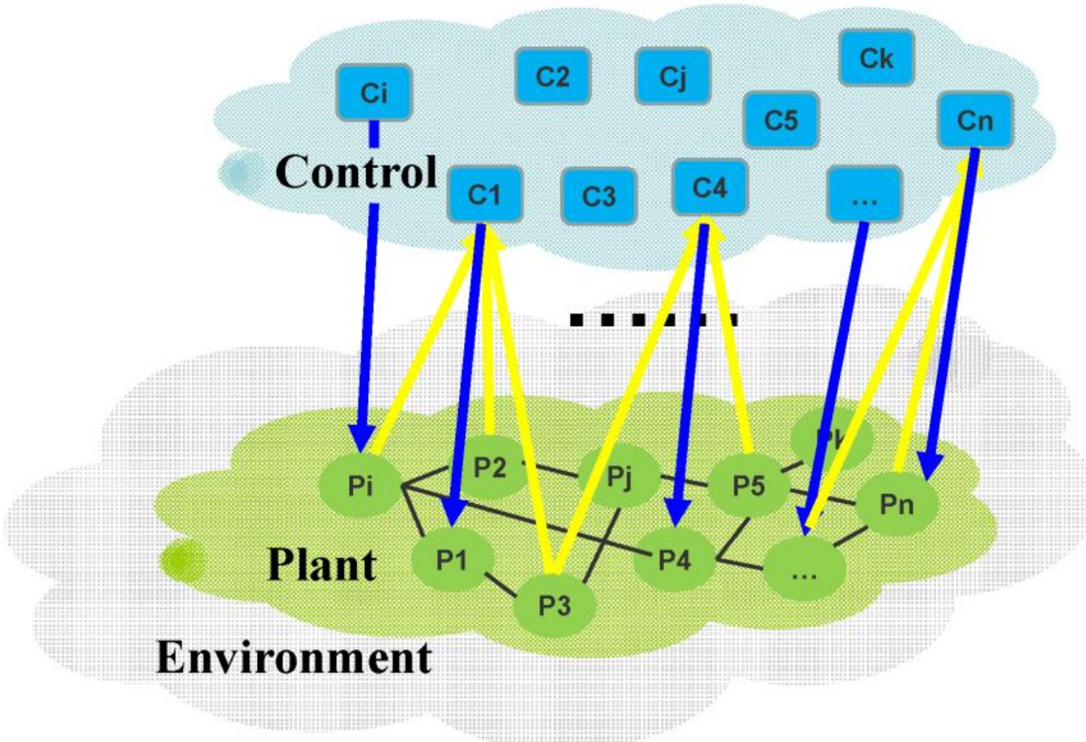
Decentralized Control

- Each node has its own controller
- Each controller uses only its own node's state



Distributed Control

- Each node has its own controller
- Each controller has a state feedback neighborhood
- Controller neighborhoods do not have to match plant neighborhoods



2. Controller Design for Disturbance Attenuation

- Consider again the same plant network, but with an added input

$$\begin{aligned} \dot{x}_i &= Ax_i + Bu_i + F \sum_{j \in \mathcal{N}_i} w_{ij}(x_j - x_i) + Ed_i, \\ z_i &= x_i - \frac{1}{N} \sum_{j=1}^{N} x_j, \end{aligned}$$

- Two cases
  1. Case 1: Decentralized control: $u_i = -Kx_i$
  2. Case 2: Distributed control: $u_i = K \sum_{j \in N_{c,i}} w_{c,ij}(x_j - x_i)$
- **Problem**: Pick the gain K so that
  1. Consensus is achieved when $d_i(t) \equiv 0$
  2. $\|T_{zd}(j\omega)\|_\infty$ is as small as possible when $d_i(t) \neq 0$
- Can give similar Theorems and LMIs as above

Controller Design Example

- Consider same plant as above, with inputs added to each node
- Plant network is a ring graph with unity weights
- Results become
  1. Decentralized controller: $\|T_{zd}\|_{\infty} = 0.0389$
  2. Distributed controller with different controller topologies:

	Complete graph	Ring graph	Star graph	Tree graph
w_c	1_{10}	1_5	1_4	1_4
$\lambda_{c,2}$	5	1.382	1	0.382
K	$\begin{bmatrix} 0.1908 & 34.5966 \\ 34.7841 & -0.1740 \end{bmatrix}$	$\begin{bmatrix} 0.6425 & 34.4697 \\ 35.0695 & -0.5626 \end{bmatrix}$	$\begin{bmatrix} 0.9244 & 34.4434 \\ 35.3193 & -0.8597 \end{bmatrix}$	$\begin{bmatrix} 2.7760 & 33.8264 \\ 36.2309 & -2.5066 \end{bmatrix}$
γ_{min}	0.0439	0.0835	0.0982	0.1592
$\|T_{zd}(s)\|_{\infty}$	0.0095	0.0286	0.0440	0.0862

- **Comment**: Decentralized controller does better than some distributed topologies

Summary

Introduction

- Systems as networks

Consensus Paradigm

- Dynamic Networks
- Concepts and extensions
- Consensus and Resilience
- Examples

Resilient Dynamic Networks though Disturbance Attenuation

- Designing network weights
- Designing network controllers

What's next? -1

- Apply ideas from control theory in a network context
  - Controllability/observability/fault-tolerance
  - Note: New journal:

IEEE Transactions on Control of Network Systems

… systems with interconnected components ...

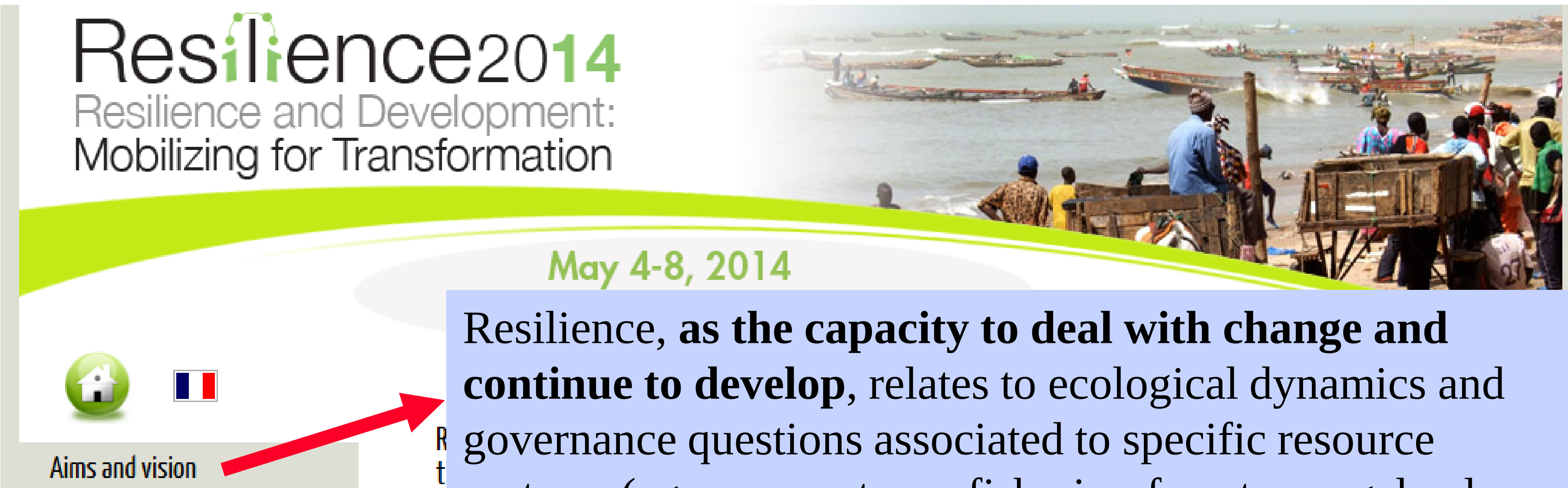
The IEEE Transactions on Control of Network Systems publishes high-quality papers on systems with interconnected components. The journal is primarily interested in problems related to the control

What's next? -2

- Apply ideas from graph theory and network science
  - Degree distributions, clustering, centrality, betweenness, communicability, …
  - E.g., betweenness centrality: the number of shortest paths from all vertices to all others that pass through that node
    - These two graphs will have different vulnerabilities

What’s next? -3

- Apply ideas from ecology (http://www.resilience2014.org)



Resilience, **as the capacity to deal with change and continue to develop**, relates to ecological dynamics and governance questions associated to specific resource systems (agro-ecosystems, fisheries, forests, rangelands, marine and freshwater ecosystems), and to global issues such as biodiversity conservation, urban growth, economic development, human security and well- being.

Acknowledgments

Professor D. Subbaram Naidu, Idaho State

– Measurement and Control Engineering Research Center

Professor Nicholas Flann, Utah State

– Center for Self-Organizing and Intelligent Sy

Professor YangQuan Chen, UC-Merced

– The MESA Lab

David Watson, David Schied

Dr. I-Jeng Wang, Dr.

– Johns Hopki (APL)

Professors T Steele, Mines

Dr. Manoja W erospace

– Center for Ro tics, Automation and Distributed Intelligence

Professor Hyo-Sung Ahn; Kwang-Kyo Oh

– Gangju Institute of Technology, Korea

Professor Deyuan Meng

– Beihang University, Beijing, Chinea

Dr. Craig Rieger, Idaho National Laboratory

and many students!

Thanks for your attention!

Questions?