Distributed Estimation and Control for Networked Systems

Minyue Fu

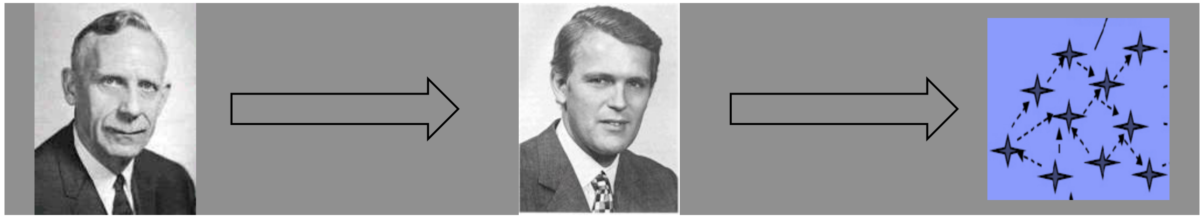
Prepared for Chinese Control Conference 2013

Main Collaborators: Zhiyun Lin, Damian Marelli
Ph.D. Students: Xin Tai, Eduardo Rohr, Hao Xing, Yuting Mu
at
Zhejiang University University of Newcastle

Outline

- **Motivations and Examples**
- **Distributed Approaches**
- **Distributed Solutions**
- **Conclusions**

Evolution of Our Field



- Control Science and Engineering is becoming a mature field.
- We are penetrating into neighbouring disciplines.
- Many applications for *Networked Dynamic Systems*.
- Abundant research opportunities, yet new theories and tools are seriously lacking.

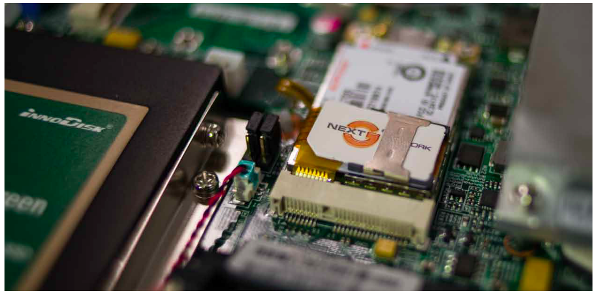
Opportunities

- Manufacturing: automation; robotics; high precision control; nano-technology; …
- Energy: smart power networks; renewable energy; smart buildings; carbon capture; emission reduction; …
- Water: modelling of water resources; usage efficiency; water pollution; desalination; …
- Transportation: road traffic control; air traffic control; high speed trains; hybrid vehicles; …
- Health: system biology; automated health systems; ageing problems and assisting systems for the aged; …
- Food: farming; land management; food processing; …

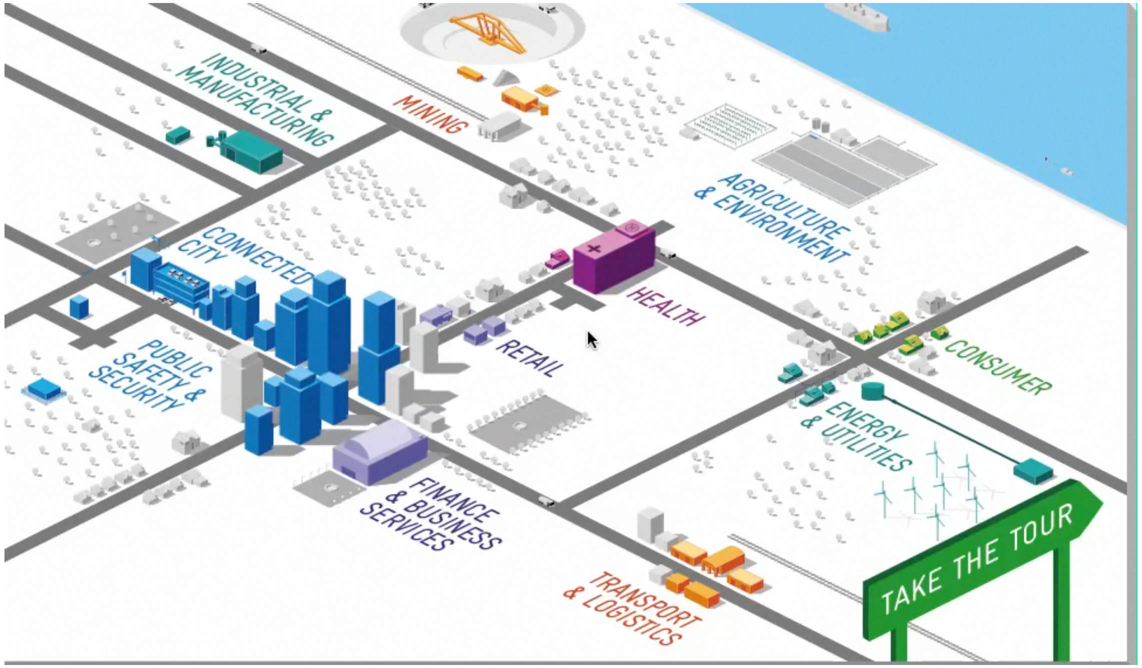
The list goes on …

Example 1: Machine-to-Machine (M2M) Communication

Mobile device:



Mobile network:

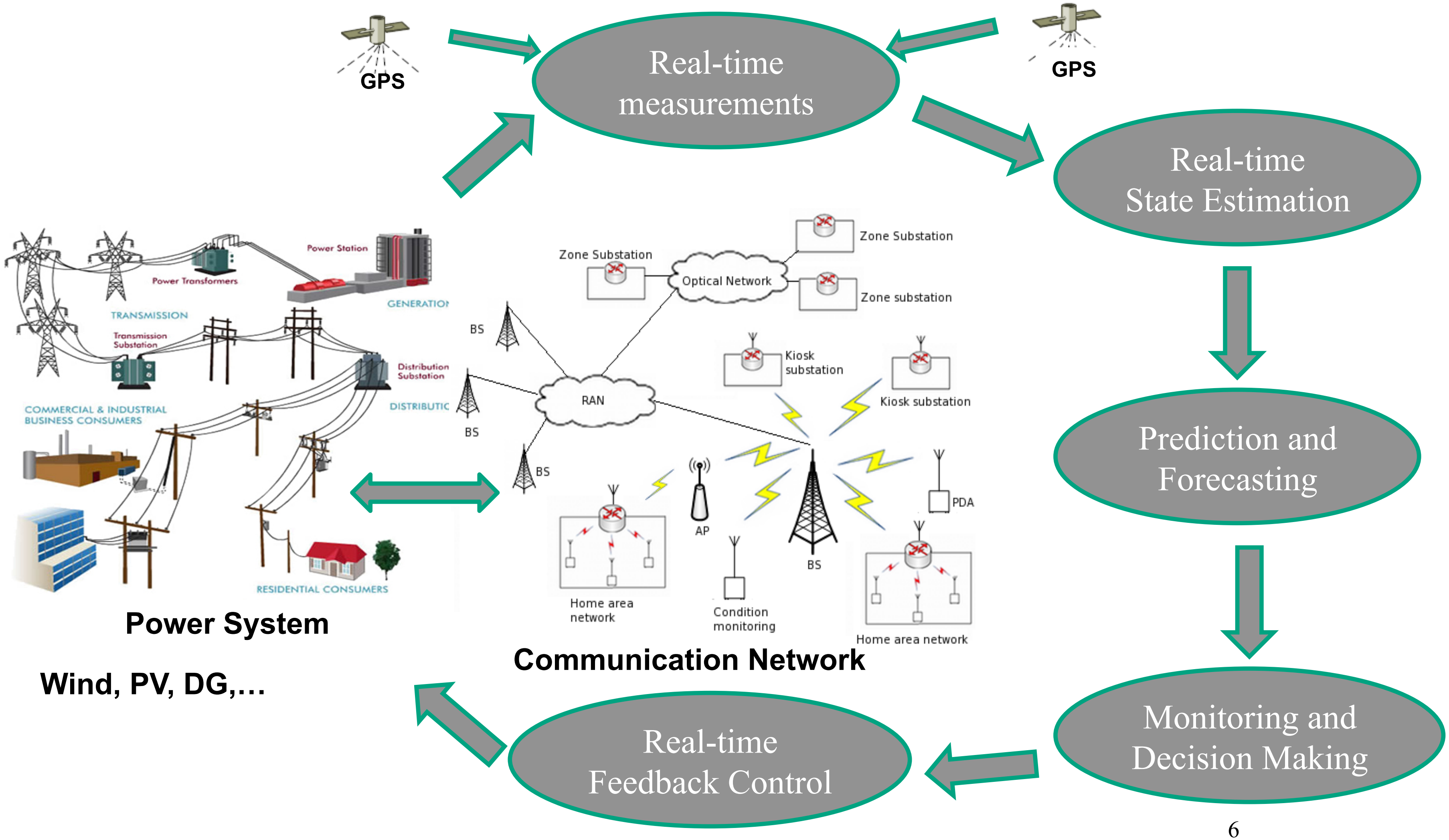


Applications:

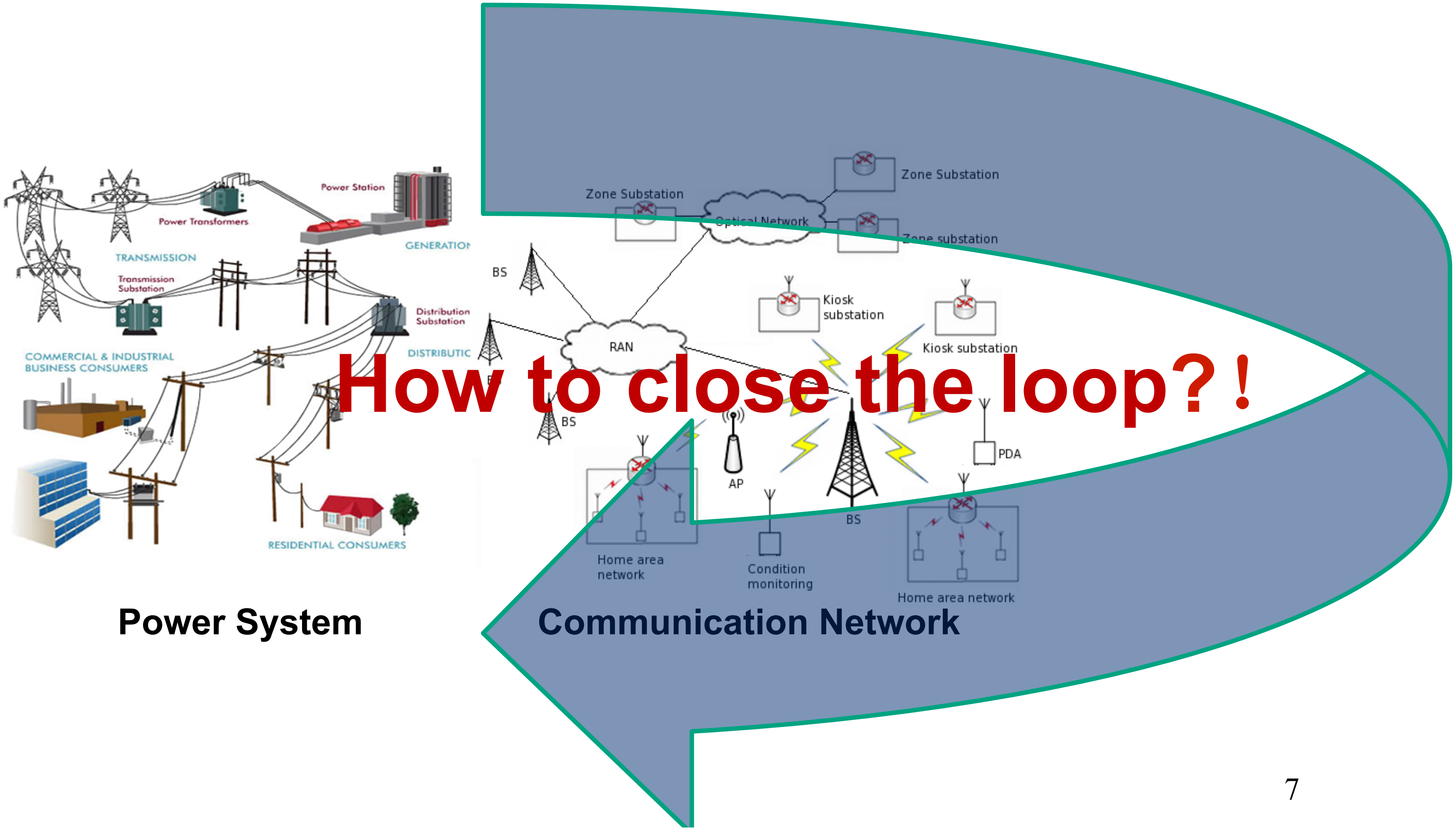
- Survellience
- Traffic control
- Smart grid
- Smart buildings
- Transportation
- Logistics
- Disaster management

(Image from Telstra, Australia)

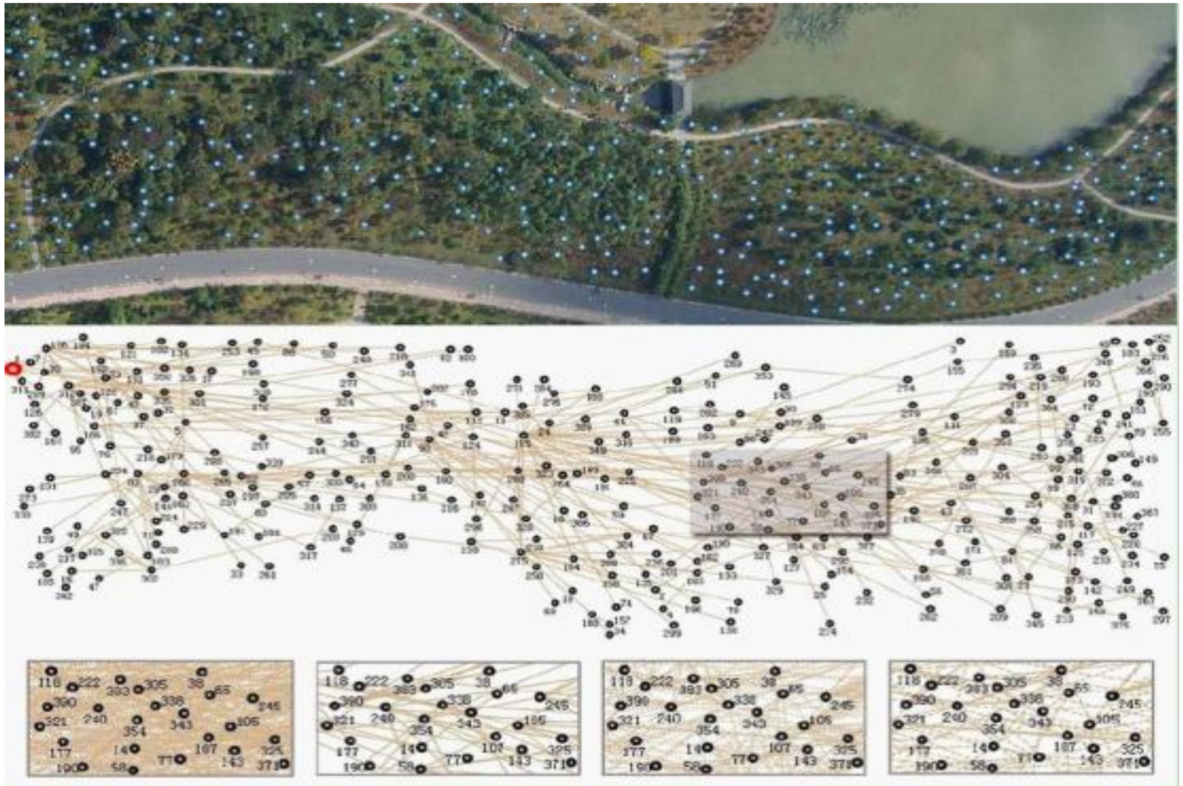
Example 2: Smart Electricity Network



For the first time, we have the technologies to do real-time control for large-scale power systems!



Example 3: Sensor Networks



- GreenOrbs (2,000 sensor nodes)
- Developed by HKUST, Tsinghua University, etc.
- Located in Wuxi, China
- Environmental surveillance (forest, ocean, CO2 pollution…)

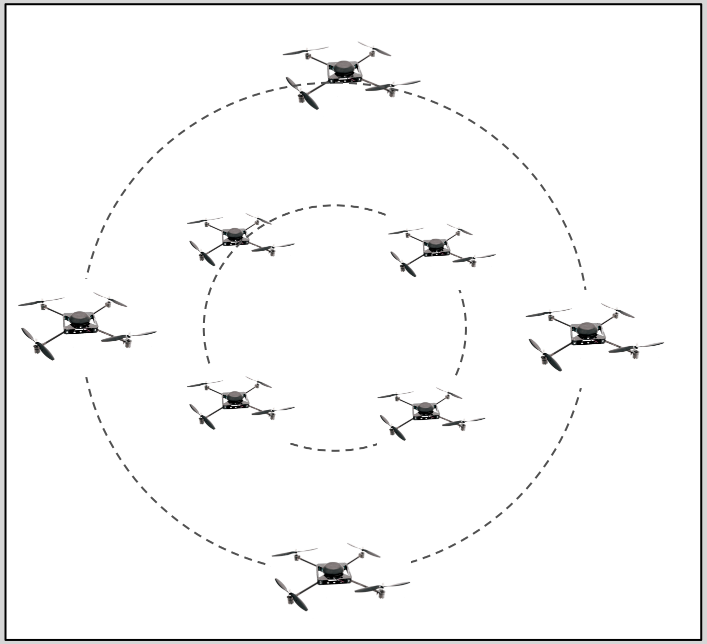
Large number of nodes;
Widely deployed;
Low power support

Distributed processing is a necessity

Key Research problems:

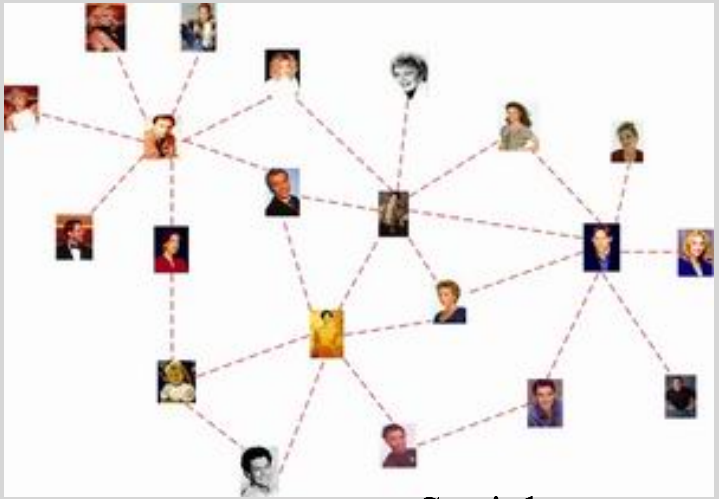
Clock synchronization
Localization
Data fusion
Target tracking
Mobile sensing …

Example 4: Multi-agent Systems



Engineered system

Natural system



Social system

Key Research problems:

- Consensus/synchronization
- Self organization/steering
- Cooperative motion/behavior
- Formation/flocking/swarming…

Recommended ref: Nagy, Akos, Biro, Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, 2010.

Outline

- Motivations and Examples
- Distributed Approaches
- Distributed Solutions
- Conclusions

Common features of these applications

Networked Systems

A networked System consists of overlay of two networks

- Physical network
  - ❑ power distribution/transmission network;
  - ❑ traffic network (roads and vehicles);
  - ❑ biological cells;
  - ❑ multi-agent system;
  - ❑ Internet of Things; …
- Information/communication network

New Research Challenges

- **New modelling methods:** physical networks; Communication networks; much faster dynamics (multi-time scales);
- **New control and estimation techniques:** theory and algorithms;
- **New simulation tools:** physical networks; communication networks; scenario simulators; (super) real-time simulators
- **Network design and planning**

Need *distributed* solutions!

Types of Distributed Solutions

Typical characteristics of a large network:

- Many many nodes (n)
- Each node (i) has only a few neighbors (k_i)
- Sparsely connected graph

- It is difficult to define "distributed solution"
- Characteristics of *fully* distributed solution:
  - Computational complexity per node: $\sim k_i$
  - Communication load per node: $\sim k_i$
  - Data storage size per node: $\sim k_i$
  - No global information (e.g., network size, topology etc) needed
  - No leader node is required/assumed
  - Certain global goal/performance is guaranteed
- Most available distributed solutions are only *partially* distributed, or global performances are not guaranteed.

Types of Distributed Solutions

Typical characteristics of a large network:

- Many many nodes (n)
- Each node (i) has only a few neighbors (k_i)
- Sparsely connected graph

- It is difficult to define "distributed solution"
- Characteristics of *fully* distributed solution:
  - Computational complexity per node: ~ k_i
  - Communication load per node: ~ k_i
  - Data storage size per node: ~ k_i
  - No global inf
  - No leader node is required/assumed
  - Certain global goal/performance is guaranteed
- Most available distributed solutions are only *partially* distributed, or global performances are not guaranteed.

Key Feature: Think globally, act locally!

Examples of Partially Distributed Solutions:

- Distributed control with a central node (leader) which collects information from each node and commands their actions
- Distributed state estimation where each node gathers data from neighbors and estimates the state of the entire network
- Average consensus algorithm for discrete-time systems:

$$x_i(k+1) = x_i(k) + d_i \sum_{j \in N(i)} w_{ij}(x_j(k) - x_i(k))$$

where the design of d_i requires global information. In this case, execution of algorithm is fully distributed, but design of the algorithm is not fully distributed.

Easy Problems vs. Hard Problems

Consensus:

- Every consensus: Easy
- Max consensus: Easy
- Sum consensus: Harder (?)

Network Topology:

- Without loops: Easy
  Many problems solved using ***belief propagation***
- With loops: mostly hard
  - ***Belief propagation*** provides good approximation
  - Some problems solvable (e.g., average consensus)

Challenge: Classification of easy/hard problems

Global design vs. local design

- Many distributed solutions are easy to implement, but the design requires global information.
- Question: what global information is acceptable?

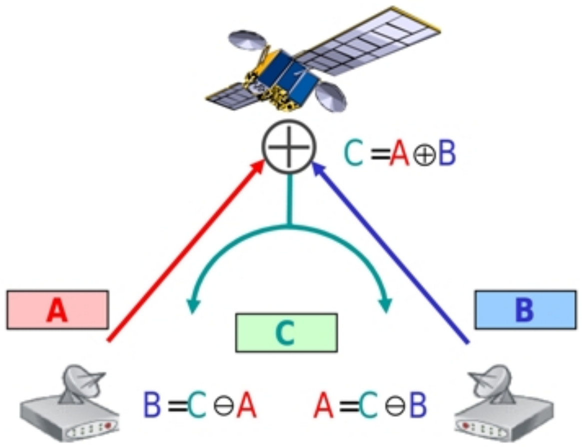
Optimal vs. sub-optimal (approximation)

- In some cases, distributed algorithms give globally optimal solutions, but in many cases, only approximations are available.
- Difficulty: How to quantify the approximation quality

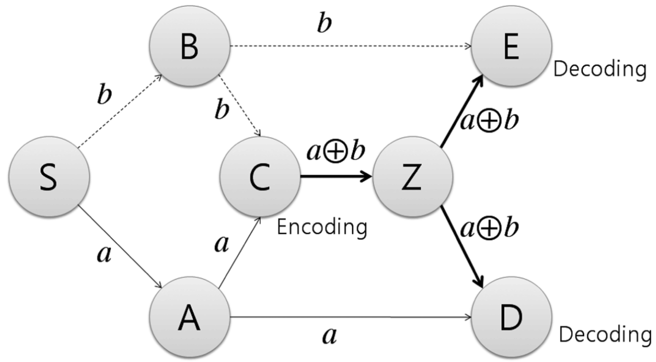
The need for distributed solutions is not unique to our field. We should learn from our neighboring disciplines …

Network Coding

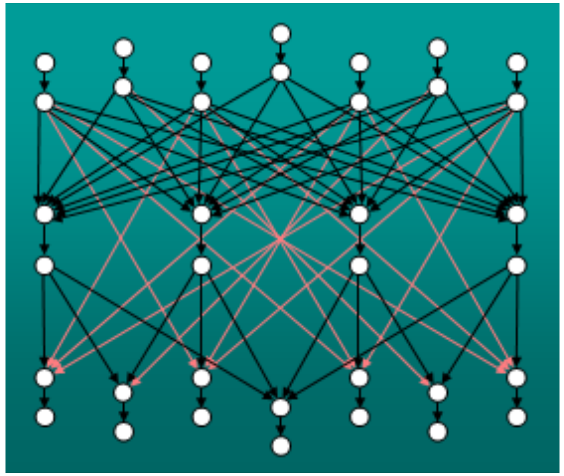
Major new development in Network comunications!



Relay system



Broadcast system



General system

Key Features:

- Much higher channel capacity than Shannon's
- Much more robust transmission
- Much higher security

Research problems:

- Determining network channel capacity
- Network encoding & decoding
- Reducing computational complexity
- Reducing time delay
- **Network coding based control?**

Recommended ref: R. Ahlswede et. al., "Network information flow," IEEE Trans. Info. Theory, 2001.

Brief Intro to Belief Propagation (BP)

- Well-known technique for distributed algorithms in artificial intelligence and statistics, Proposed by Pearl in 1982 (Also known as Pearl's BP).
- Very powerful for computing maximum *a posterior* (MAP) solutions, but powerful for many optimization problems, including marginal distributions in Bayes nets, Markovian random fields (MRF) and graphs (factor graphs in particular).
- Provide *exact* (*optimal*) solutions in for tree graphs, but approximate solutions for general graphs.
- Many known distributed algorithms can be interpreted as BP.
- Little known to the control community, but it is changing…

Recommended Ref: "Factor Graphs and the Sum-Product Algorithm" by F. R. Kschischang, B. J. Frey, and H-A Loeliger, IEEE Transactions on Info Theory, Feb. 2001

Problem: Given a function (think of joint probability density)

$$P(x_1, x_2, ..., x_N)$$

find the marginals (think of marginal distribution)

$$P_i(x_i) = \sum_{x_1 \cdots x_{i-1} x_{i+1} \cdots x_N} P(x_1, x_2, ..., x_N) \quad \text{for all } i = 1, 2, ..., N$$

Combinatoric explosion: If each x_i takes M possible states, brute force computation takes $O(M^N)$ calculations.

In general, the problem is known to be NP-hard.

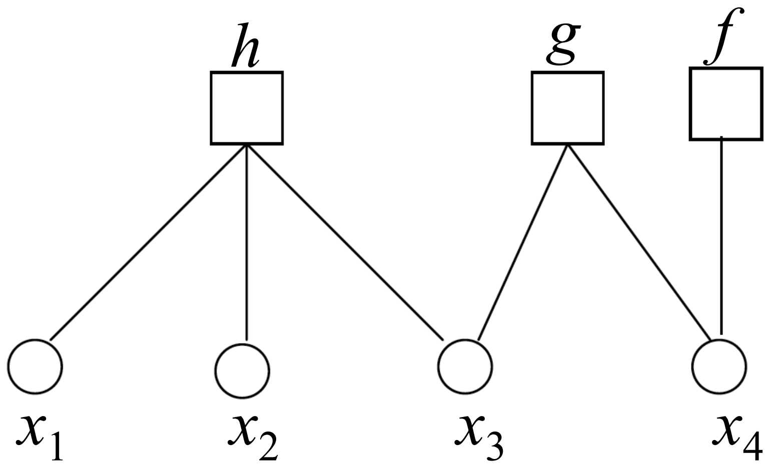
Efficient algorithm explores some underlying structure.

Factors: BP explores the structure of the function

$$P(x_1, x_2, \ldots, x_4) = h(x_1, x_2, x_3)\, g(x_3, x_4)\, f(x_4)$$

factors

Factor graph:

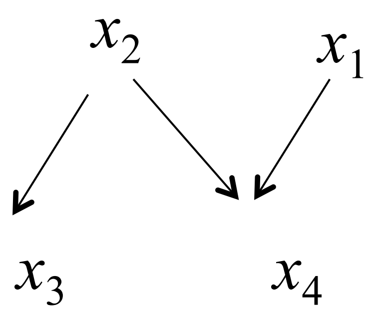


Example: Consider the joint probability density function

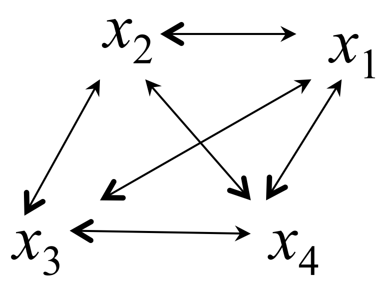
$$P(x_1, x_2, ..., x_4) = P(x_1)P(x_2)P(x_3 | x_2)P(x_4 | x_1, x_2)$$

instead of the general expression by Bayes rule:

$$P(x_1, x_2, ..., x_4) = P(x_1)P(x_2 | x_1)P(x_3 | x_1, x_2)P(x_4 | x_1, x_2, x_3)$$



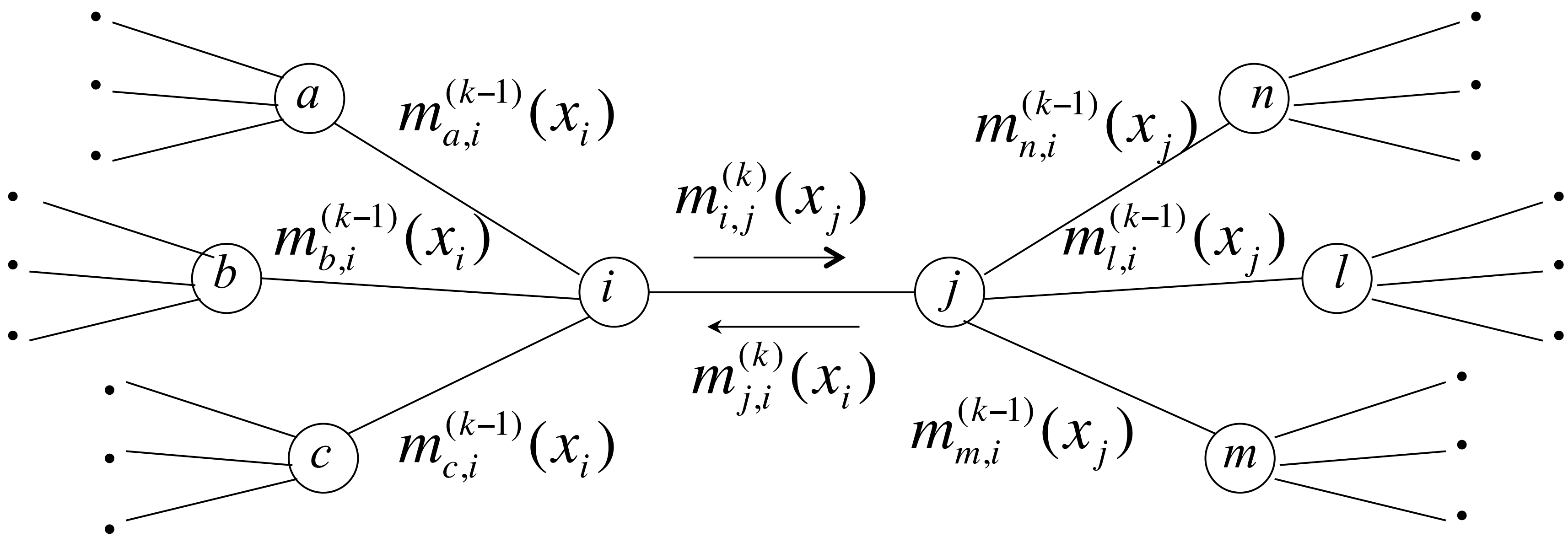
Graph for this example: tree



Graph for general function

Assumption: The graph does not have loops, i.e., it is a *tree*.

Message Passing: At each iteration k, each variable node i computes an estimate of the marginal $m_{i,j}^{(k)}(x_j)$ for each neighboring node j and pass it on to node j.



The celebrated sum-product rule:

$$m_{i,j}^{(k)}(x_j) \propto \sum_{x_i} P(x_j \mid x_i) P^a(x_i) \prod_{s \in \mathrm{N}(i) \backslash j} m_{s,i}^{(k-1)}(x_i)$$

Prior estimate of node *i*

Neighborhood (without node *j*)

Independent estimates of node *i*

Key Result: After a finite number (max path length) of iterations, $m_{i,j}^{(k)}(x_j)$ will converge, and the *exact* marginals are

$$P_i(x_i) \propto P^a(x_i) \prod_{j \in N(i)} m_{j,i}^{(k)}(x_i)$$

Further Comments on BP

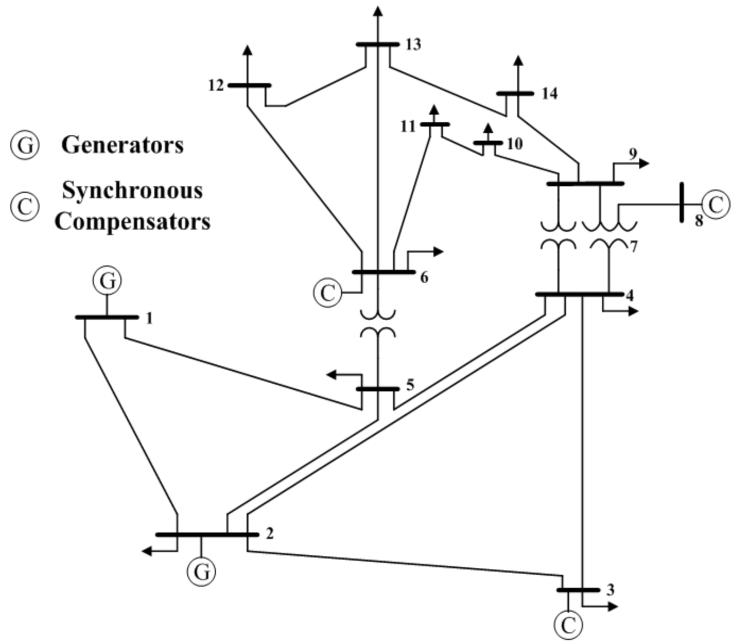
- Very general algorithm, easy to implement, fully distributed
- Kalman filtering algorithm is known to be BP
- Excellent approximations in many cases: Turbo codes, low-density parity check codes,…
- A variant of sum-product rule, max-product, is used to compute $\arg \max_{x_1, x_2, \ldots, x_N} P(x_1, x_2, \ldots, x_N)$
- Many other variants available for solving more complicated problems
- Many “tricks” available to deal with loops

How to connect control problems to BP?

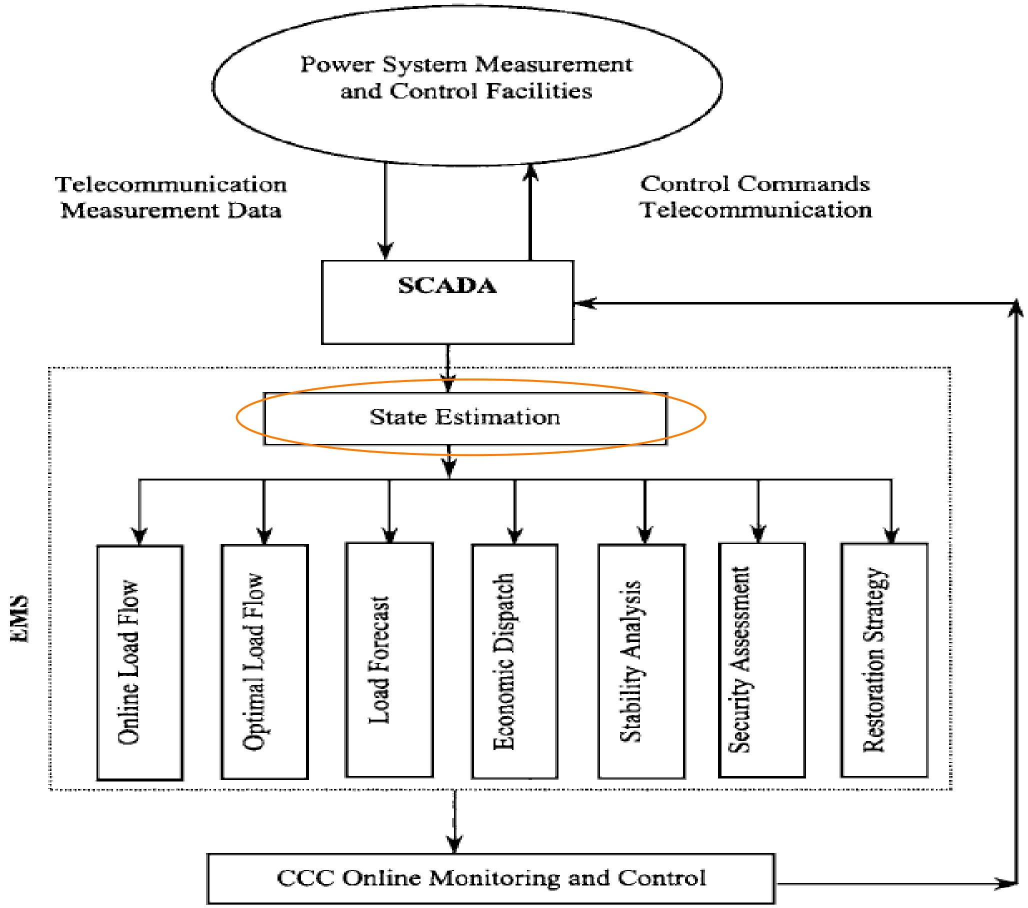
Outline

- **Motivations and Examples**
- **Distributed Approaches**
- **Distributed Solutions**
  - Distributed state estimation for power networks
  - Distributed localization for sensor networks
  - Distributed consensus with quantized information
  - Distributed control of multi-agent systems
- **Conclusions**

A New Distributed State Estimation Technique for Power Networks



Power network



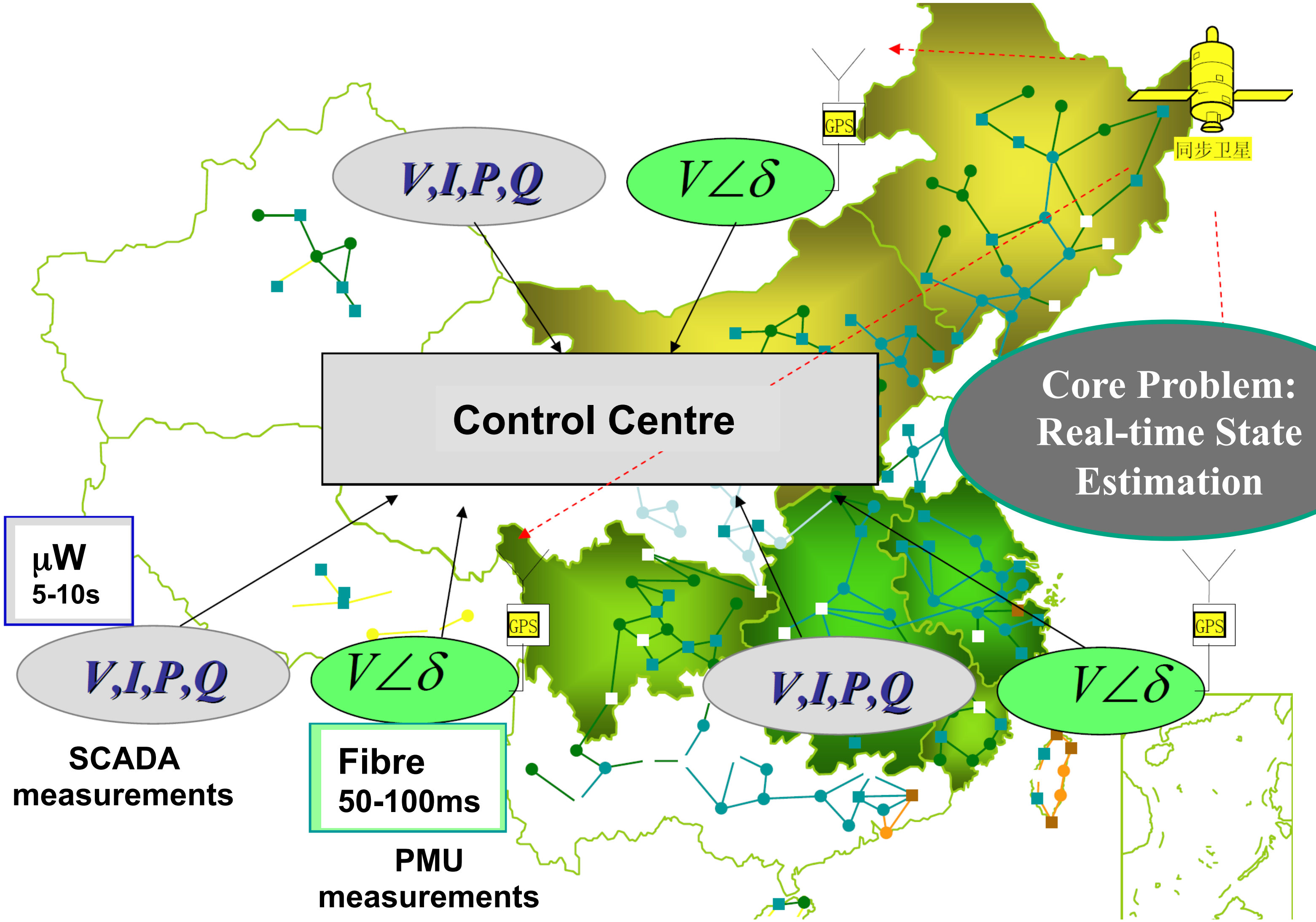
Energy management system (EMS)

X.Tai, Z. Lin, M. Fu, Y Sun, ACC 2013; D. Marelli and M. Fu, CDC 2013 (to appear)

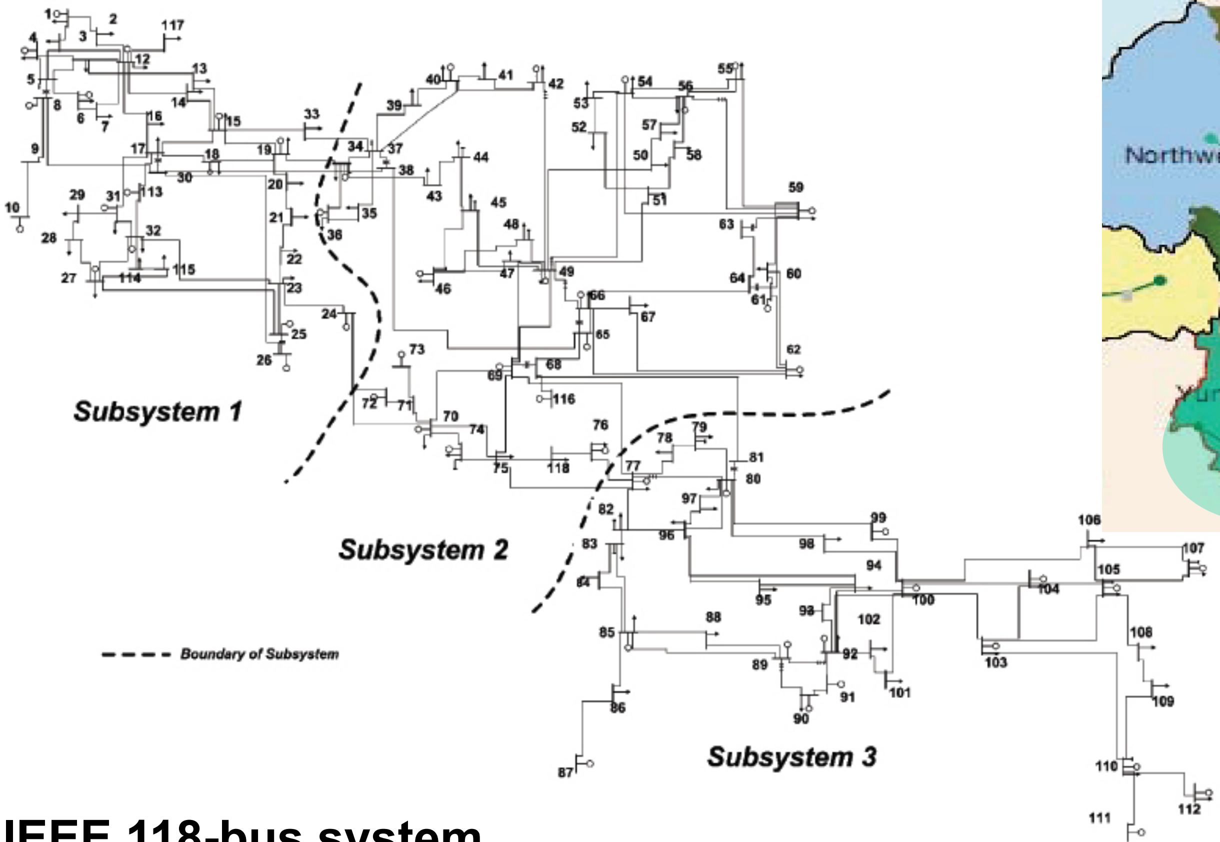
Wide Area Measurement System (WAMS):

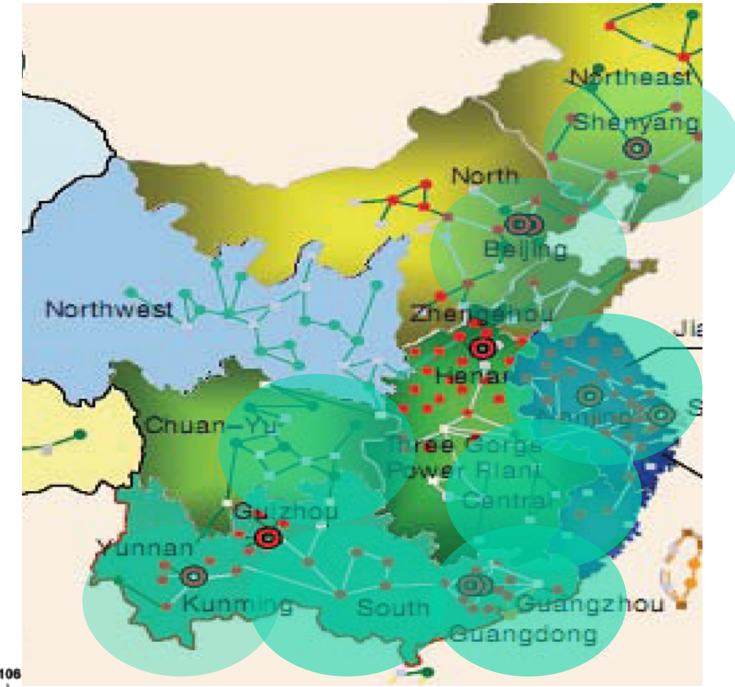
PMU = phasor measurement unit

Typical Communication Network for WAMS:



Distributed State Estimation Problem





IEEE 118-bus system

System Model

Measurement equation

$$m = Hx + e.$$

where m is the PMU measurement which contains cooresponding measured voltage phasors and current phasors. $x = \{x_1, x_2, \cdots, x_n\}$ is the state indicating the voltage phasors of all the buses. e is the measurement noise yields to Gaussian distribution with zero mean and covariance T.

Centralized estimate

$$\begin{aligned} \hat{x} &= \arg\min_x (m - Hx)^T T^{-1} (m - Hx) \\ &= \left(H^T T^{-1} H\right)^{-1} H^T T^{-1} m \end{aligned}$$

Weighted least squares (WLS)

Estimation error covariance

$$\begin{aligned} \Sigma_{\tilde{x}} &= \mathscr{E}\left\{(x - \hat{x})(x - \hat{x})^T\right\} \\ &= \left(H^T T^{-1} H\right)^{-1}. \end{aligned}$$

(max likelihood)

Distributed processing

Key Question: Can the global estimate be computed in a distributed manner?

System Partition

Suppose the whole system is partitioned into I subsystems.

- Internal local measurements in subsystem i:

$$y_i = A_i x_i + v_i$$

- Boundary measurements between subsystem i and its neighbour j:

$$z_{i,j} = D_{i,j} x_i + C_{i,j} x_j + w_{i,j}$$

Comparing to the global system model, we can get

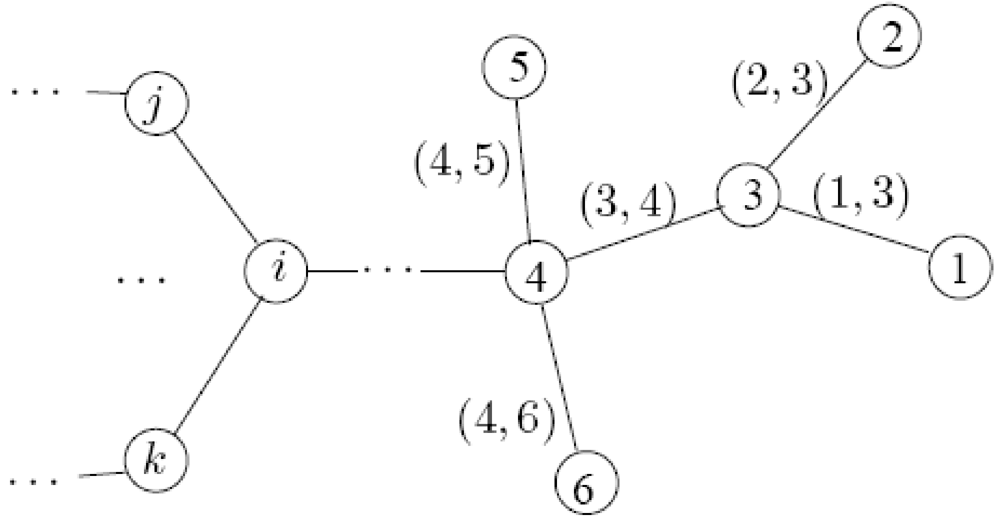
$$x = \left[x_i^T : i \in \mathscr{V} \right]^T, e = \left[v_i^T, w_{i,j}^T : i \in \mathscr{V}, j \in \mathscr{N}_i \right]^T,$$

$$m = \left[y_i^T, z_{i,j}^T : i \in \mathscr{V}, j \in \mathscr{N}_i \right]^T.$$

Distributed WLS Algorithm

Assumptions:

- The topological structure of the graph constituted by subsystems is acyclic.
- The measurement matrix of the whole system H is of full column rank.
- The connectivities among different subsystems are low.



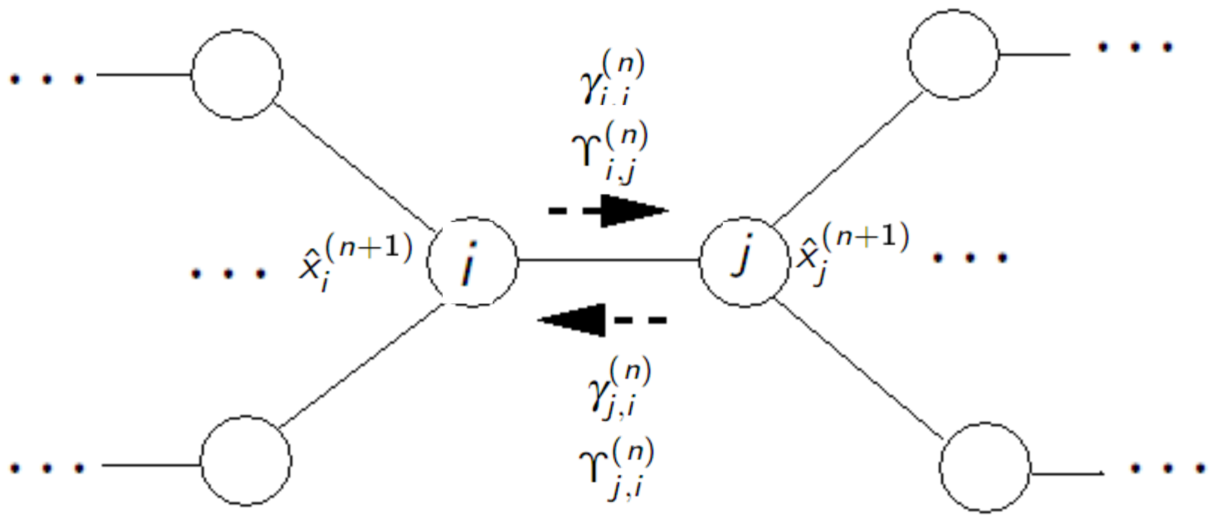
$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

$$i \in \mathcal{V} = \{1, 2, \cdots, n\}$$

An interconnected system represented by an acyclic graph.

Distributed WLS algorithm

Algorithm illustration:



Message passing

Distributed WLS algorithm

Initialization:

1) Compute the local estimate and its associated estimation error covariance

$$\begin{aligned}\check{x}_i^{(0)} &= \check{\Sigma}_i^{(0)}\check{\alpha}_i,\\ \check{\Sigma}_i^{(0)} &= \check{\Psi}_i^{-1},\end{aligned}$$

with

$$\begin{aligned}\check{\alpha}_i^0 &= A_i^T R_i^{-1} y_i + \sum_{j\in\mathcal{N}_i} D_{i,j}^T S_{i,j}^{-1} z_{i,j},\\ \check{\Psi}_i^0 &= A_i^T R_i^{-1} A_i + \sum_{j\in\mathcal{N}_i} D_{i,j}^T S_{i,\mathcal{N}_i(k)}^{-1} D_{i,j}.\end{aligned}$$

2) Transmit to every node $j \in \mathcal{N}_i$ the correction factor

$$\begin{aligned}\gamma_{j,i}^{(0)} &= D_{i,j}\check{x}_j^{(0)},\\ \Upsilon_{j,i}^{(0)} &= D_{i,j}\check{\Sigma}_j^{(0)} D_{i,j}^T.\end{aligned}$$

Distributed WLS algorithm

Main loop: For $t = 0, 1, \cdots$, let $\gamma_{i,j}^{(t)}$, $\Upsilon_{i,j}^{(t)}$ be the correction factor received by node i from node j.

1) Update the local estimate and its associated estimation error covariance based on the received correction factor:

$$\hat{x}_i^{(t+1)} = \Sigma_i^{(t+1)} \left(\check{\alpha}_i^0 - \sum_{j \in \mathcal{N}_i} \beta_{i,j}^{(t)} \right), \quad (20)$$

$$\Sigma_i^{(t+1)} = \left(\check{\Psi}_i^0 - \sum_{j \in \mathcal{N}_i} \Phi_{i,j}^{(t)} \right)^{-1}, \quad (21)$$

where

$$\beta_{i,j}^{(t)} = D_{i,j}^T S_{i,j}^{-1} \gamma_{i,j}^{(t)}, \quad (22)$$

$$\Phi_{i,j}^{(t)} = D_{i,j}^T S_{i,j}^{-1} \Upsilon_{i,j}^{(t)} S_{i,j}^{-1} D_{i,j}. \quad (23)$$

Distributed WLS algorithm

2) Compute

$$
\begin{aligned}
\check{x}_{j,i}^{(t+1)} &= \check{\Sigma}_{j,i}^{(t+1)} \left(\check{\alpha}_i^0 - \sum_{k \in \mathcal{N}_i / \{j\}} \beta_{i,k}^{(t)} \right), \\
\check{\Sigma}_{j,i}^{(t+1)} &= \left(\check{\Psi}_i^0 - \sum_{k \in \mathcal{N}_i / \{j\}} \Phi_{i,k}^{(t)} \right)^{-1}
\end{aligned}
$$

for every $j \in \mathcal{N}_i$, and then transmit to node j the correction factor

$$
\begin{aligned}
\gamma_{j,i}^{(t+1)} &= C_{j,i} \check{x}_{j,i}^{(t+1)}, \\
\Upsilon_{j,i}^{(t+1)} &= C_{j,i} \check{\Sigma}_{j,i}^{(t+1)} C_{j,i}^T.
\end{aligned}
$$

Properties

Theorem:

Consider the global system, with H having full column rank and T being invertible. If the above algorithm is used, then, for each $i \in \mathscr{V}$,

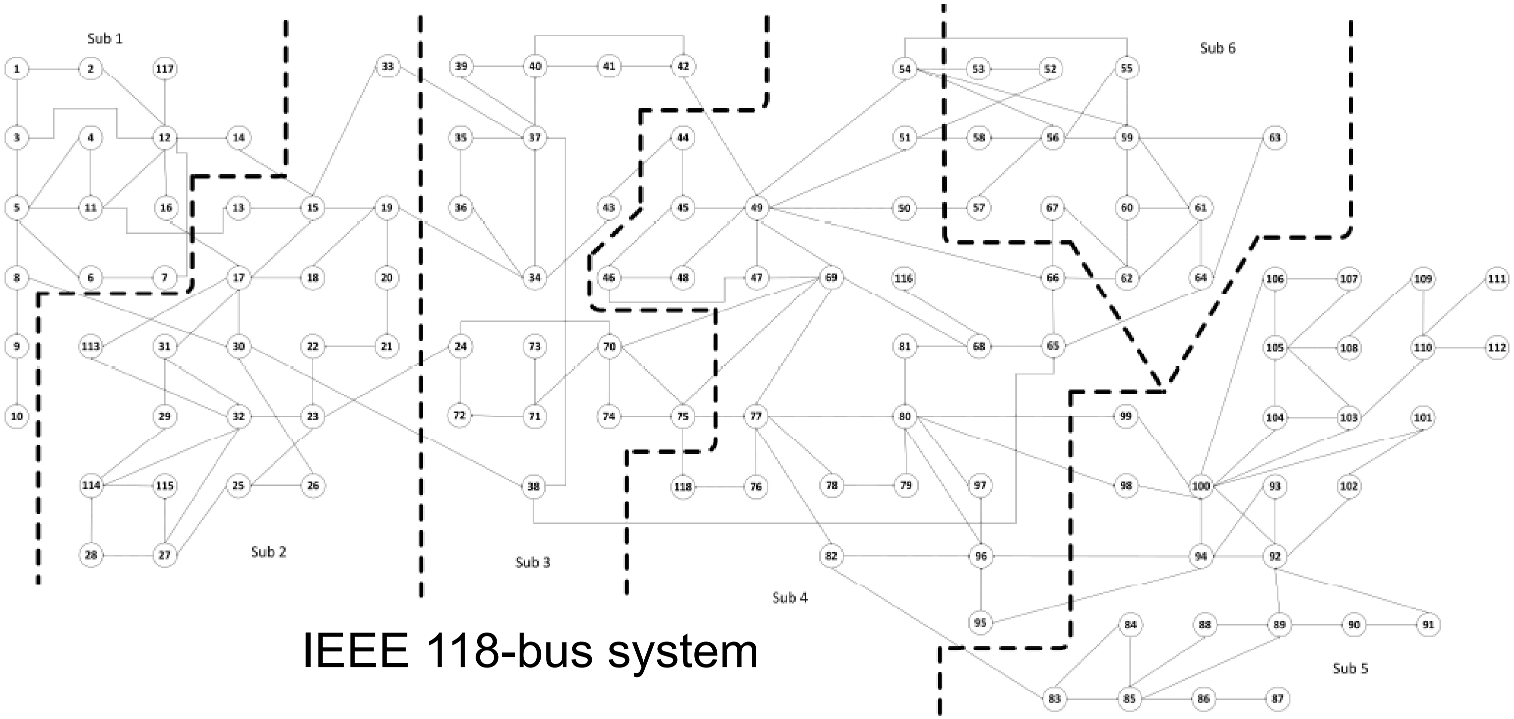
$$\hat{x}_i^{(N)} = \hat{x}_i.$$

N = max path length of the graph

Advantages:

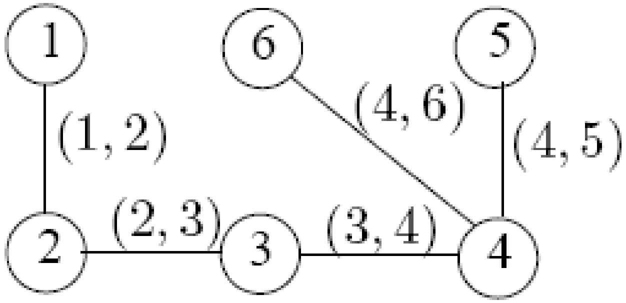
- Only use distributed computation
- Low communication burden.
- The same estimation performance as the centralized estimation.
- Good robustness to measurement delay and packet loss.

Example



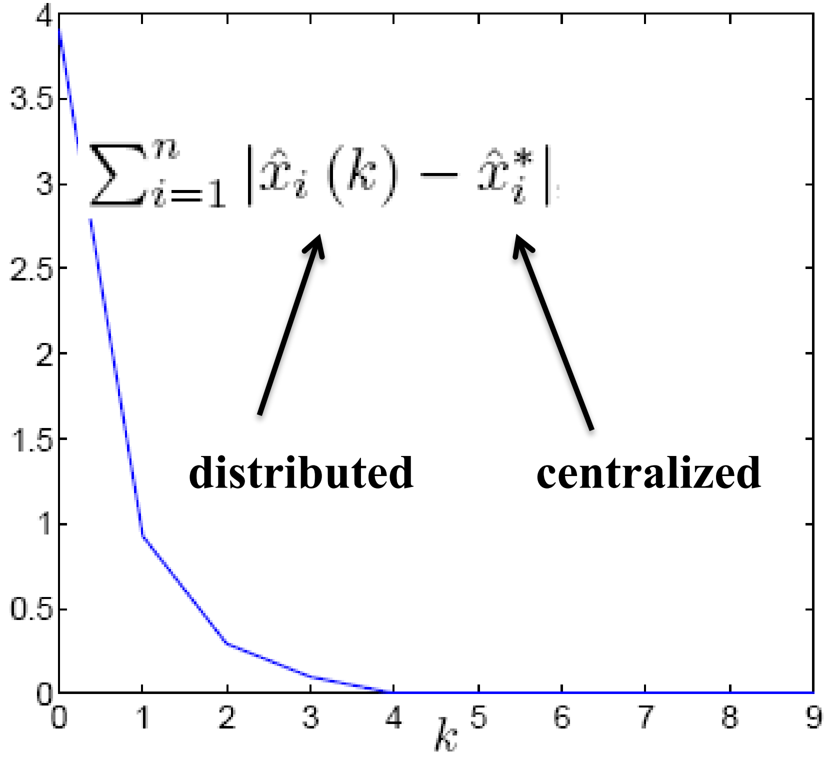
IEEE 118-bus system

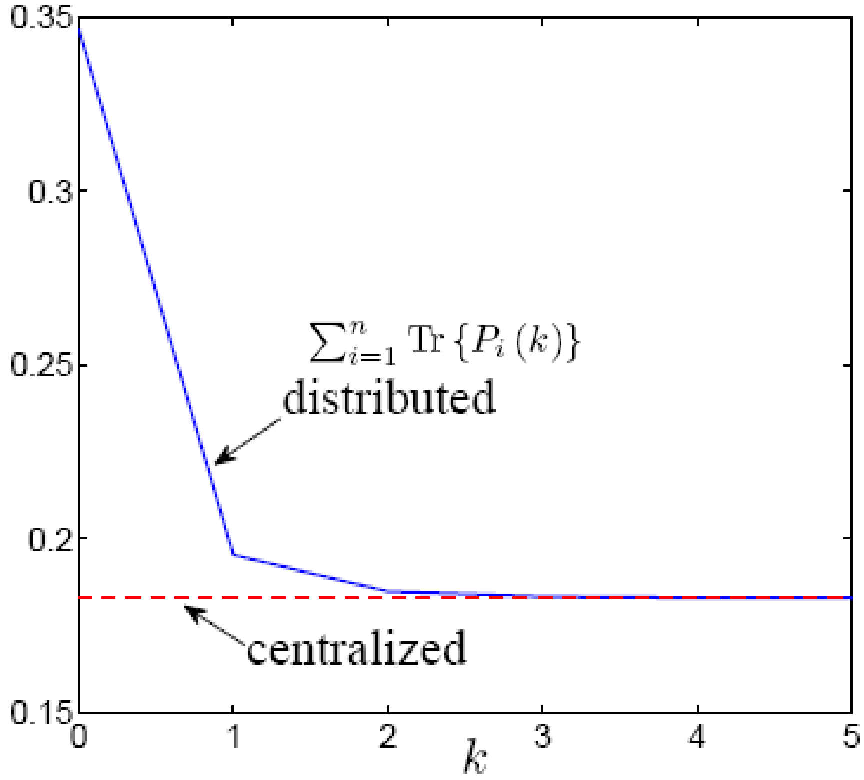
Partition graph of the system:



Graph Diameter N = 4

Distributed estimates vs. centralized estimates:





Observation: Convergence after 4 iterations

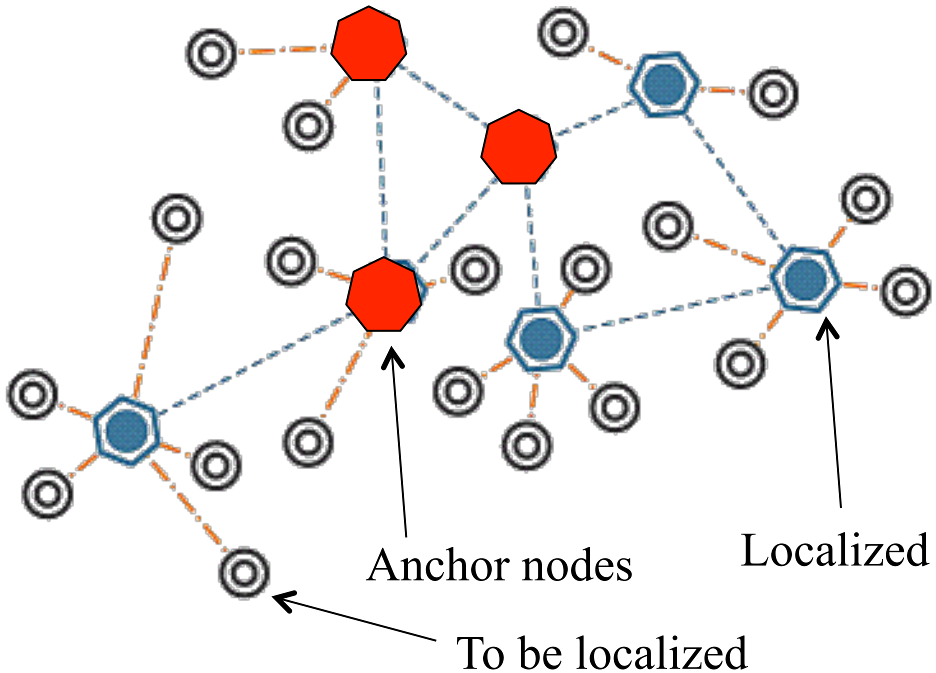
A new distributed localization method for sensor networks

2-D Localization Problem:

Given: A set of anchor nodes (with known positions) and relative distance measurements between neighboring nodes

Questions: Which nodes' positions can be uniquely determined and how to compute their positions?

Distributed algorithms are required.

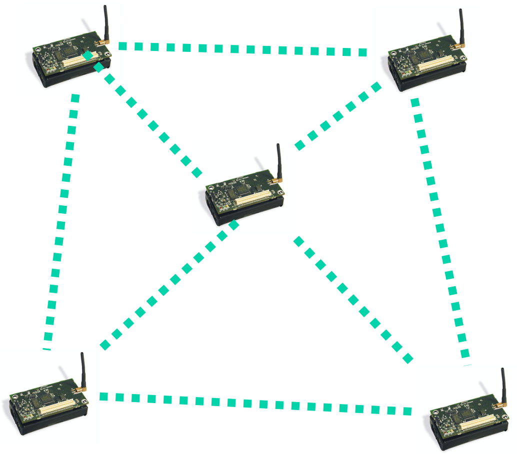


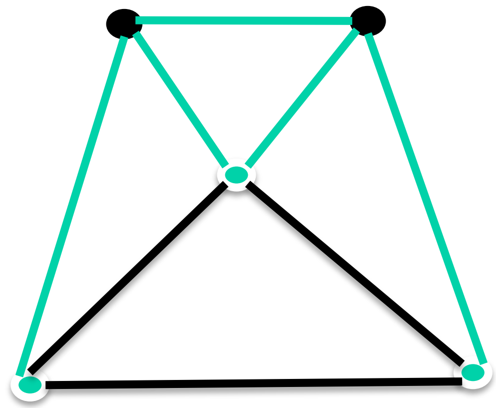
Distance Measurements:
Received Radio Strength (RSS)
Time of Arrival (TOA)
Time Difference of Arrival (TDOA)
Angel of Arrival (AOA)

Y. Diao, M. Fu, H. Zhang, Z. Lin, IEEE Trans. Networking (to appear); CDC 2012; ASCC 2013; LSS 2013

Localizability Conditions

Localizability of an entire sensor network ⟺ Rigidity of the corresponding distance graph



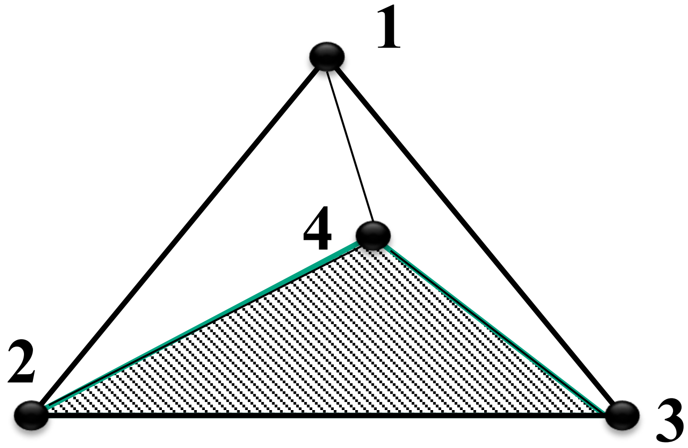


Network localizable ⟺ Globally Rigid + Three Anchor Nodes

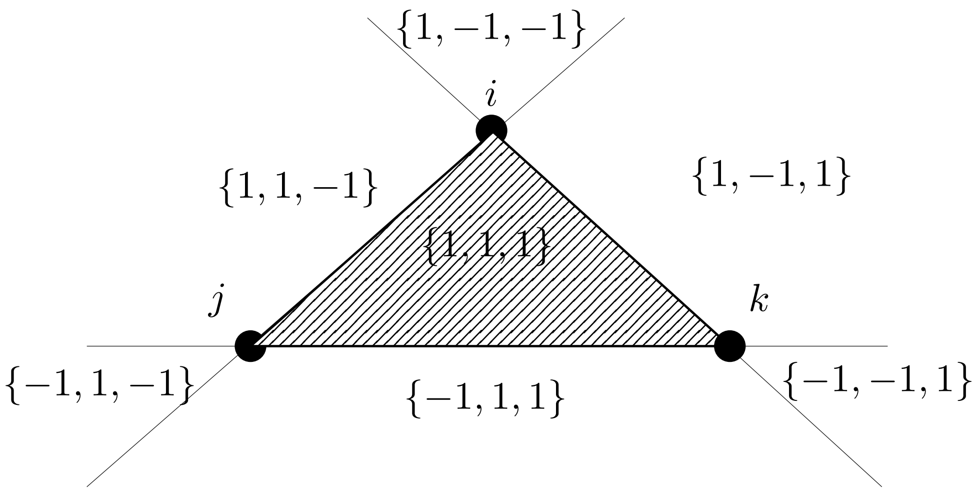
Eren et. al. , 2004

Very nonlinear problem! Also need distributed algorithm.

Barycentric Coordinate



Nodes 1,2,3 = anchor nodes
Node 4: To be localized



Pseudo linear representation:

$$p_4 = a_{41}p_1 + a_{42}p_2 + a_{43}p_3$$

where a_{ij} depend only on the distance measurements, with

$$1 = \pm a_{41} \pm a_{42} \pm a_{43}$$

Resulting localization problem:

$$\mathbf{p} = A\mathbf{p},$$

$$A = \begin{bmatrix} I_3 & 0 \\ B & C \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \mathbf{p}_a \\ \mathbf{p}_s \end{bmatrix}$$

Anchor nodes ($\mathbf{p}_a$)

Sensor nodes ($\mathbf{p}_s$)

Depending only on distance measurements (B, C)

The above can be rewritten as

$$\mathbf{p}_s = C\mathbf{p}_s + B\mathbf{p}_a$$

Iterative solution:

$$\mathbf{z}_s(t+1) = C\mathbf{z}_s(t) + B\mathbf{p}_a(t)$$

Challenge: How to guarantee convergence & how to implement?

$$\mathbf{p}_s = C\mathbf{p}_s + B\mathbf{p}_a$$

$$K\mathbf{p}_s = KC\mathbf{p}_s + KB\mathbf{p}_a.$$

K: diagonal stabilizer (pre-conditioner)

$$\mathbf{z}_s(t+1) = (I - K(I-C))\mathbf{z}_s(t) + KB\mathbf{p}_a,$$

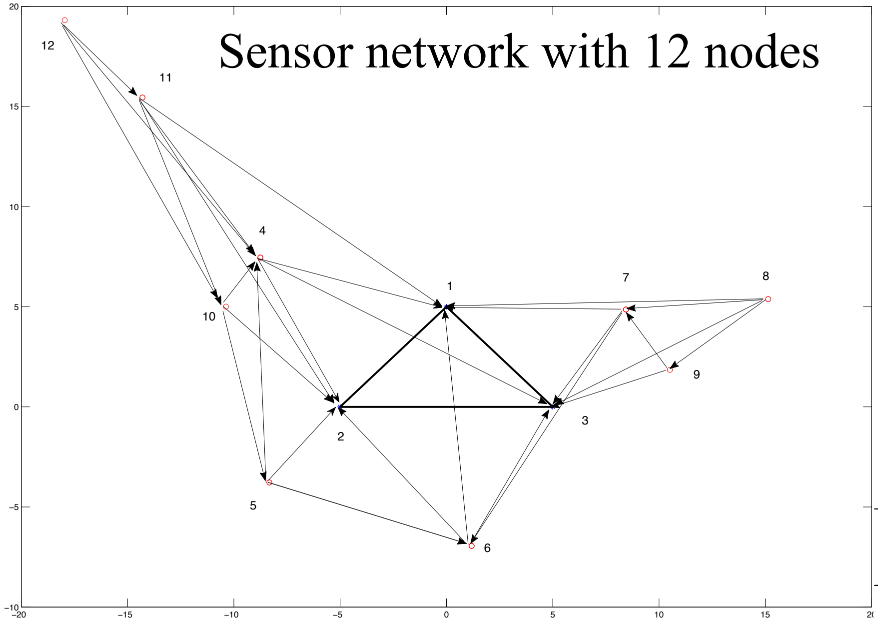
Distributed Implementation:

$$z_i(t+1) = z_i(t) - k_i\left(z_i(t) - \sum_{j\in\mathcal{N}_i} a_{ij} z_j(t)\right)$$

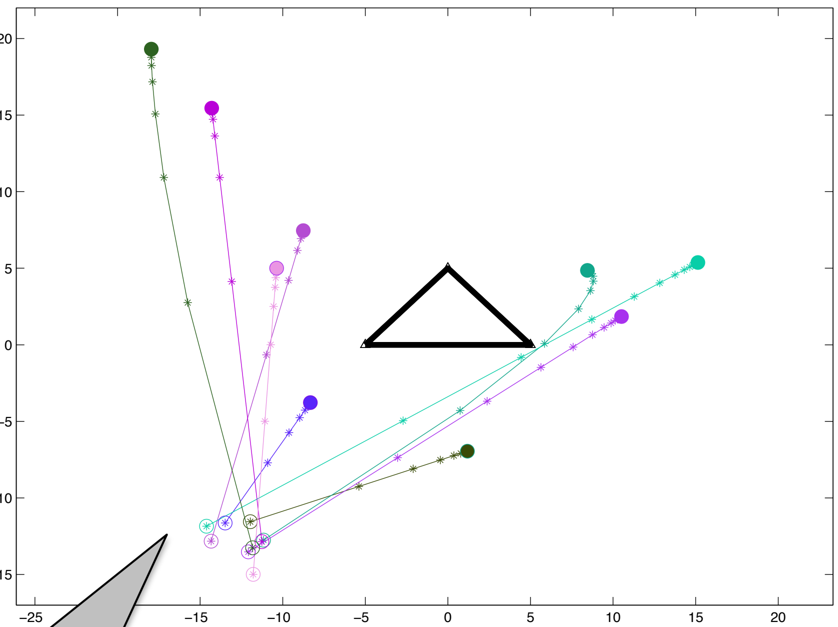
Only local information is required

Centralized and distributed methods available for computing K.

Simulation



(a) Original network topology.



(b) Trajectories of the coordinate estimates.

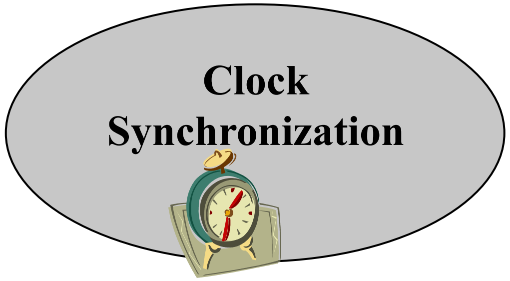
Started from randomly chosen initial positions

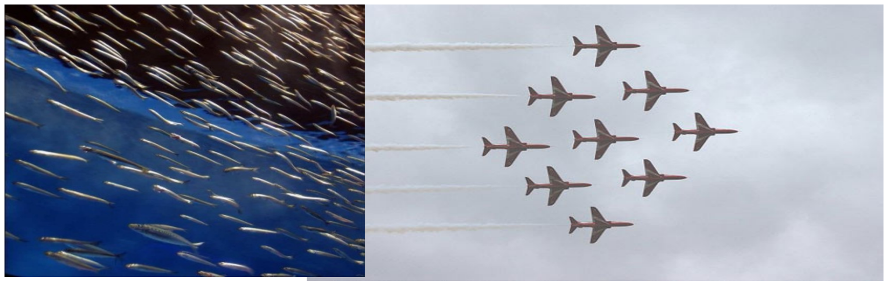
Distributed Average Consensus with Quantized Information

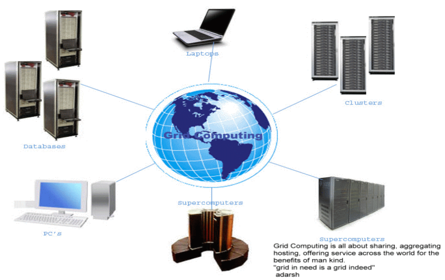
T. Li, M. Fu, L. Xie, J. Zhang,
ASCC 2009, *IEEE-TAC* 2010.

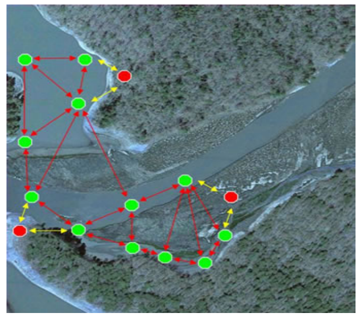
- Formation control
- Distributed estimation
- Multi-sensor data fusion
- Distributed computing

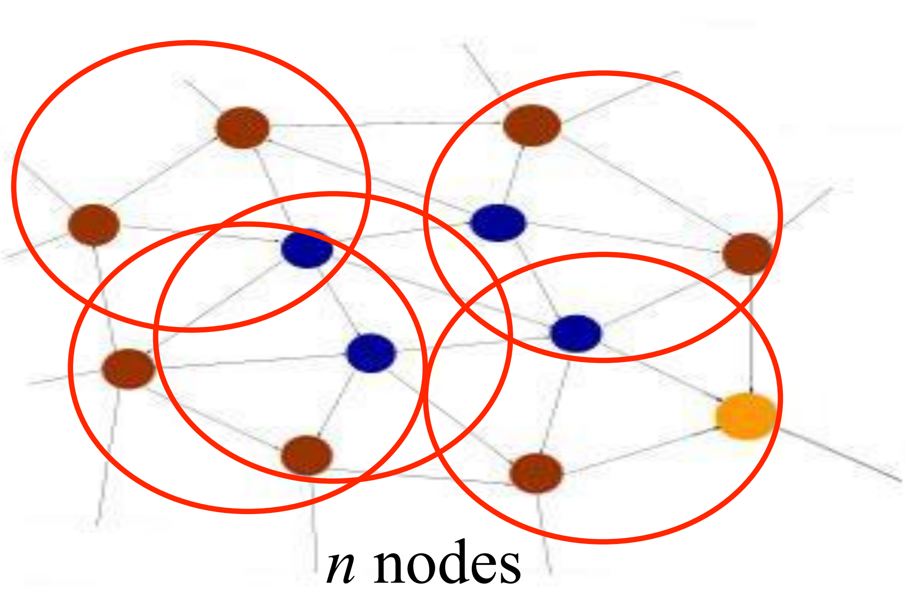
Distributed average consensus control











$$G = \{V, E, A\}$$

$$x_i(t+1) = x_i(t) + hu_i(t),$$
$$t = 0, 1, \ldots, \quad i = 1, 2 \ldots, N$$

Distributed consensus: to achieve agreement by distributed information exchange

$$x_i(t) \longrightarrow \frac{1}{N}\sum_{j=1}^{N} x_j(0), \qquad t \longrightarrow \infty$$

average consensus

With limited data rate between neighbors！

Motivations

- Pioneering work: Olfati-Saber & Murray (2004), Ren &Beard (2005), Xiao & Boyd (2004), …
- But requiring exact information exchange.
- Quantized information exchange: Kashyap et al. (2007), Carli et al. (2007,2008), Kar & Moura (2007),…
- But requiring infinite-level quantization or having non-zero steady-state error.

Our objectives:

Achieve exact average consensus with exponential convergence rate using finite data rate.

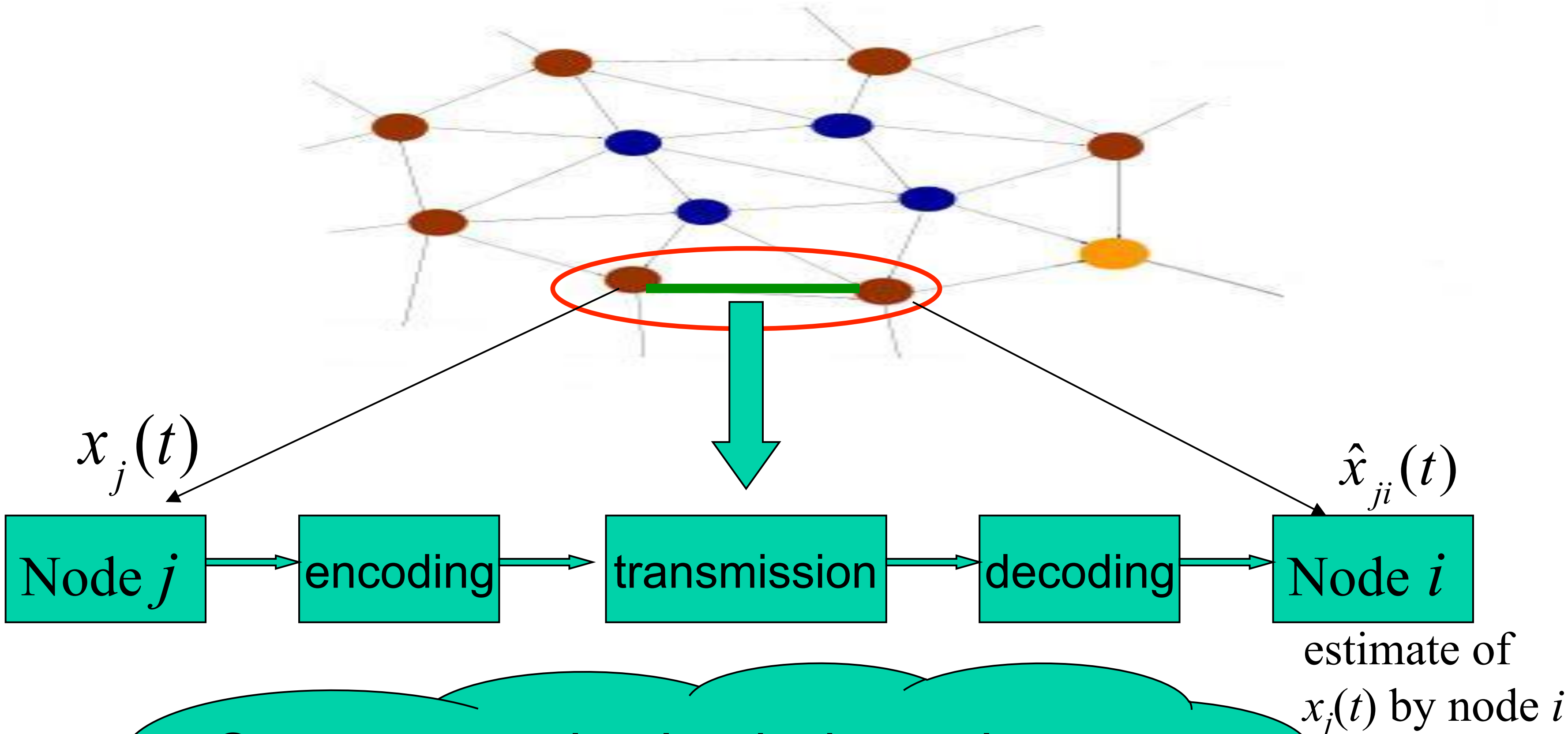
Minimize the number of bits transmitted.

Energy to transmit 1 bit	≈	Energy for 1000-3000 operations	Shnayder et al. 2004

Technical Problems:

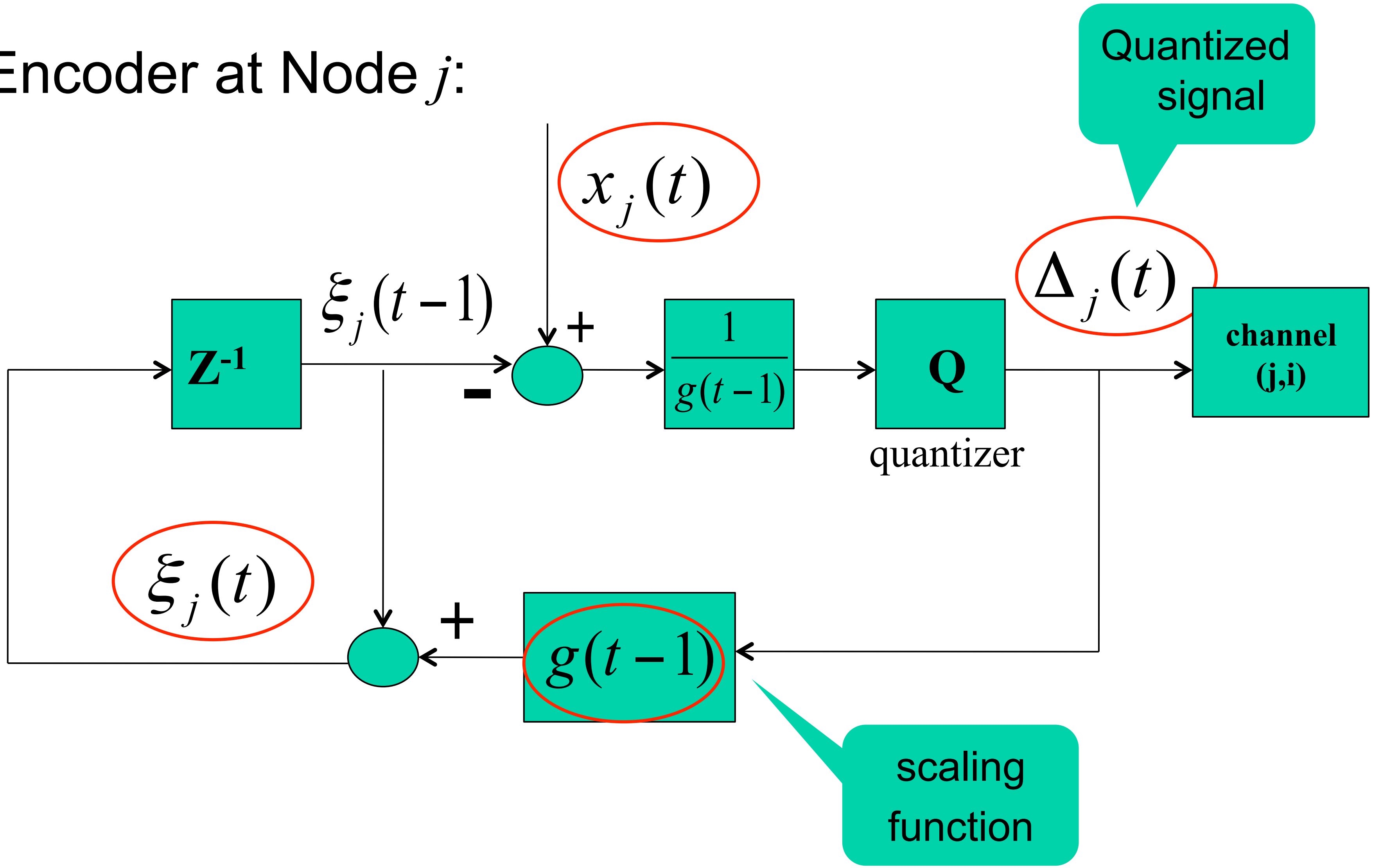
1. **How many bits does each pair of neighbors need to exchange at each time step to achieve consensus of the whole network ?**
2. **What is the relationship between the consensus convergence rate and the number of quantization levels ?**

Distributed protocol

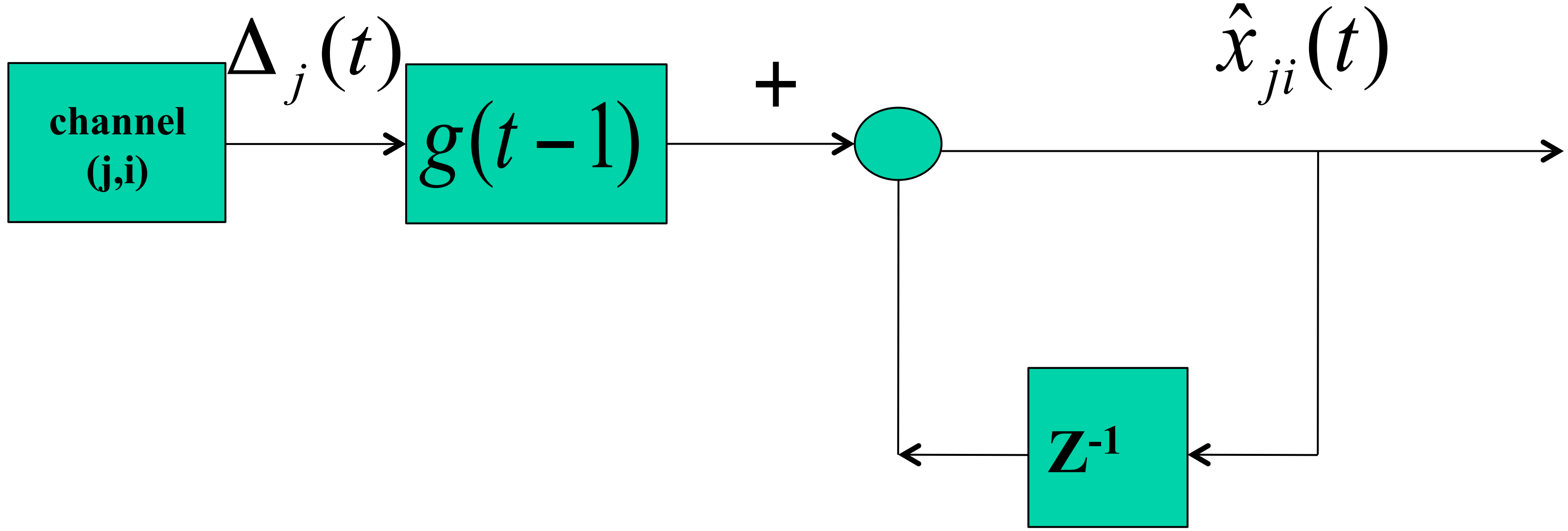


States are real-valued, but only finite bits of information are transmitted at each time-step

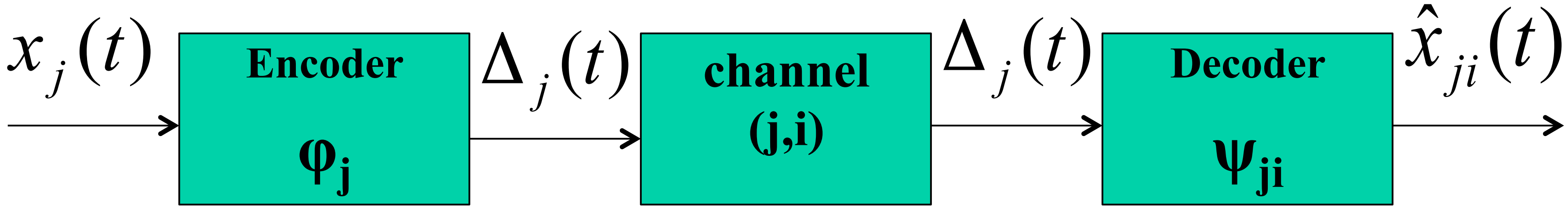
Encoder at Node j:



Decoder at Node *i*:



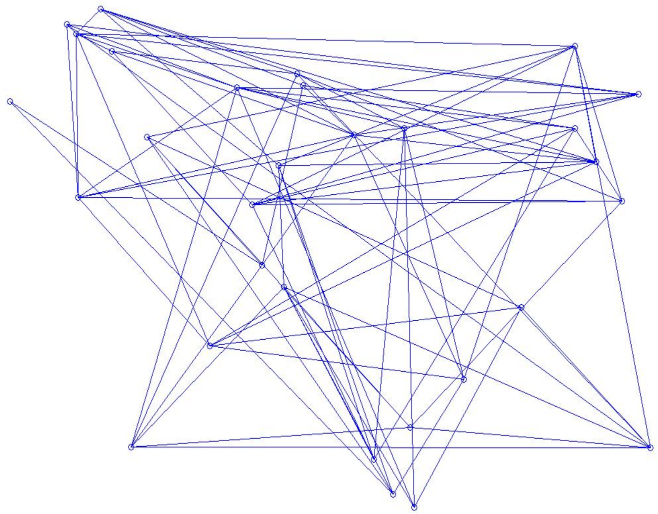
Combined Scheme:



For a connected network, average-consensus can be achieved with exponential convergence rate base on a single-bit exchange between each pair of neighbors at each time step.

The highest asymptotic convergence rate increases as the number of quantization levels and the synchronizability increase, and decreases as the network expands.

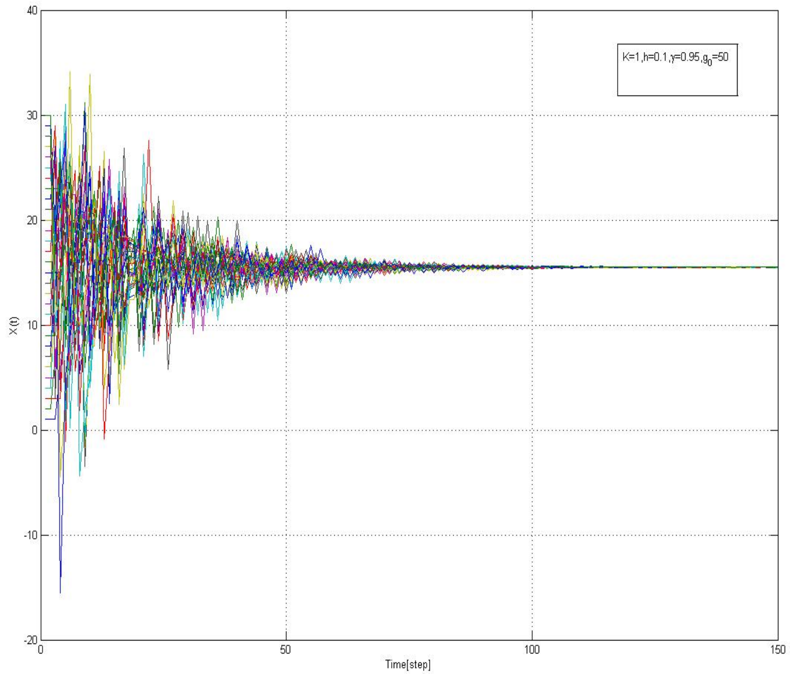
Simulation Example:



A network with 30 nodes

$$P\{(i,j)\in E\} = 0.2$$

$$x_i(0) = i,\ \ i = 1, 2, \ldots, 30$$



1 bit quantizer

Generalizations to continuous-time systems and sampled-data systems

S. Liu, T. Li, L. Xie, M. Fu, J. Zhang, to appear in *Automatica*.

Dynamics at each node:

$$\dot{x}_i(t) = u_i(t),\ i = 1, 2, \ldots, N$$

Quantizer at each node (logarithmic):

$$y_{j,i}(t) = q(x_j(t)),\ j \in \mathcal{N}_i$$

Distributed control law:

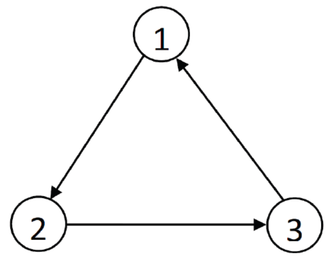
$$u_i(t) = \sum_{j \in \mathcal{N}_i} [y_{j,i}(t) - y_{i,i}(t)]$$

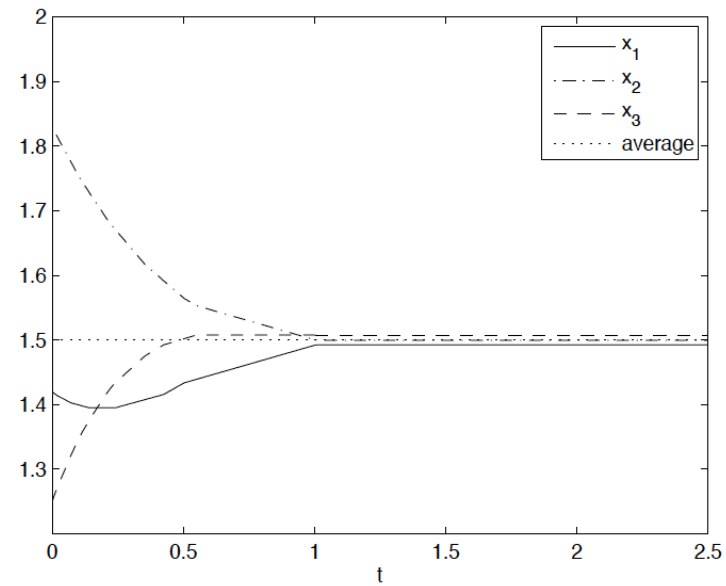
Sampled –data implementation:

$$u_i(t) = \sum_{j \in \mathcal{N}_i} [q(x_j(kh)) - q(x_i(kh))]$$

$$t \in [kh, (k+1)h)$$

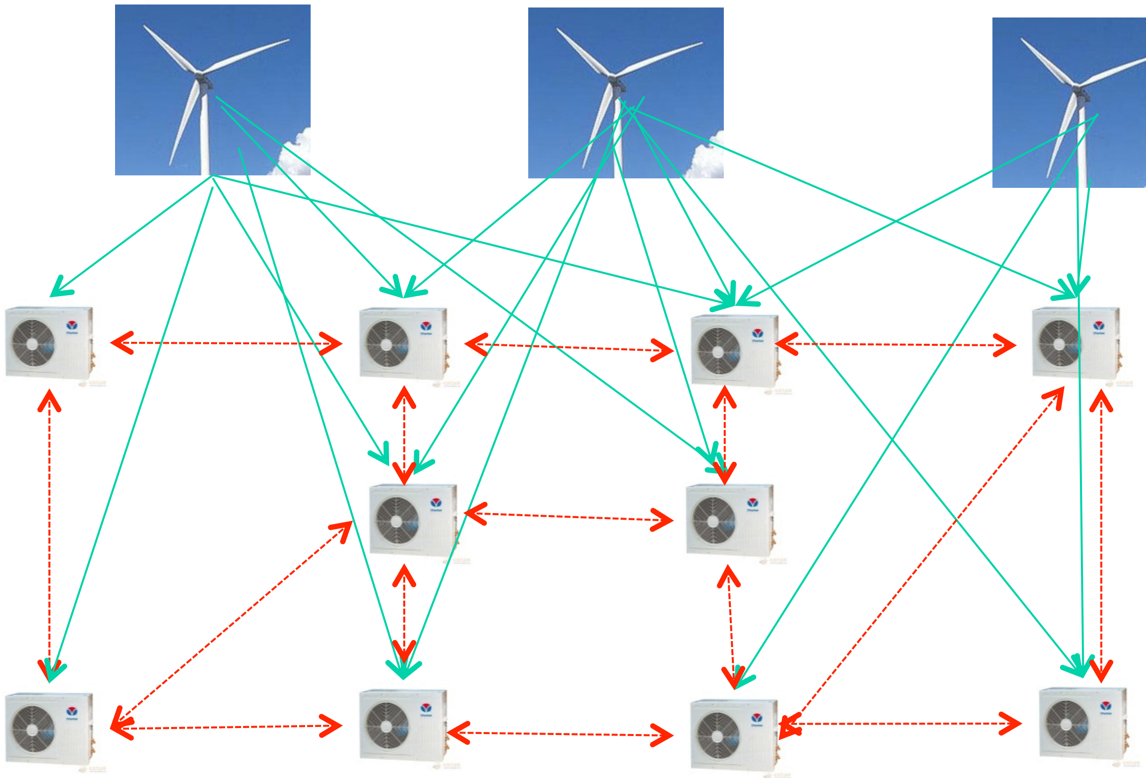
Example:





Wind Power Regulation using Air Conditioner Network

by H. Xing, Y. T. Mu,
M. Fu



- Total power supply = baseline power + fluctuating wind power;
- *m* wind power generators supplying *n* variable-freq. air conditioners (VFAC)
- **Goal**: Adjust the VFAC's load to balance the wind power fluctuation while keeping the room temperatures within a prescribed threshold.

Problem Statement

Thermal Model for VFAC:

$$T(t+1) = aT(t) + (1-a)(T_a - \eta R P(t)) + w(t)$$

temperature   time constant   ambient temp   input power   noise

Steady State Model:

$$T^{SS} = T_a - \eta R P^{SS}$$

Power consumption by the *i*-th VFAC (control variables):

$$P_i = P_{i1} + P_{i2} + \cdots + P_{im}$$

Control Objective for VFACs:

$$\frac{T_1^{SS} - T_1^{SP}}{\Delta T_1} = \frac{T_2^{SS} - T_2^{SP}}{\Delta T_2} = \ldots = \frac{T_n^{SS} - T_n^{SP}}{\Delta T_n}$$

set point

comfort zone

Power consumption constraint for the *j*-th wind power plant:

$$G_j = P_{1j} + P_{2j} + \cdots + P_{nj}, \quad j = 1, 2, \ldots, m$$

forecast power by generator *j*

Power Dispatch using Consensus Algorithm

Proposed Consensus Control Algorithm:

$$P_{ij}(k+1) = P_{ij}(k) - W_{ii}^{j} \frac{P_i(k) - P_i^{SP}}{\Delta P_i} - \sum_{l \in N_i} W_{il}^{j} \frac{P_l(k) - P_l^{SP}}{\Delta P_l}$$

for all i and j

$$P_{1j}(0) + P_{2j}(0) + \cdots + P_{nj}(0) = G_j$$

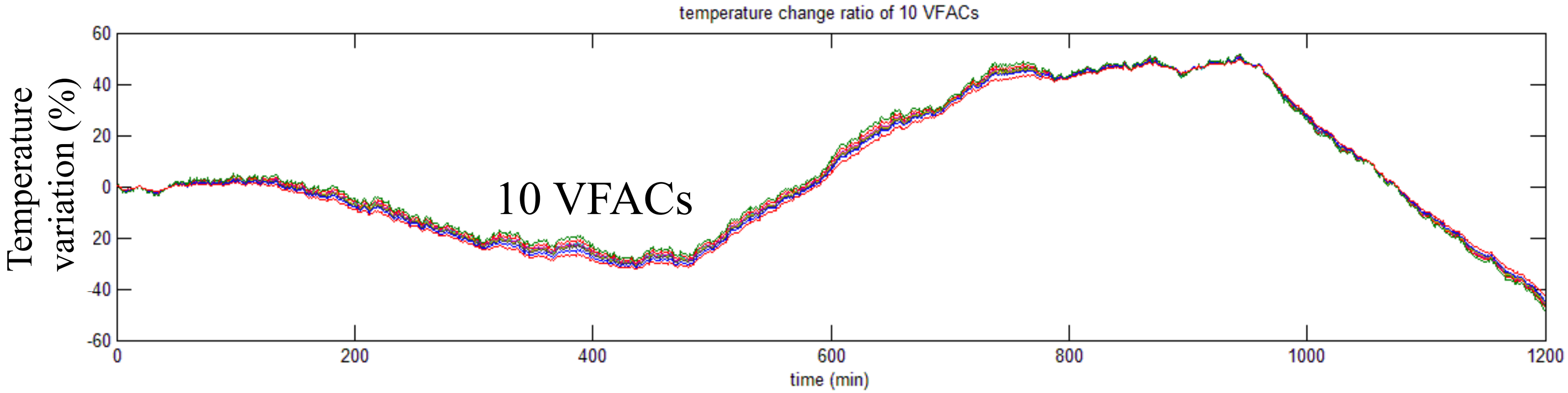
We can select the weighting parameters (in a distributed fashion) to guarantee the convergence of the algorithm, which in turn guarantees

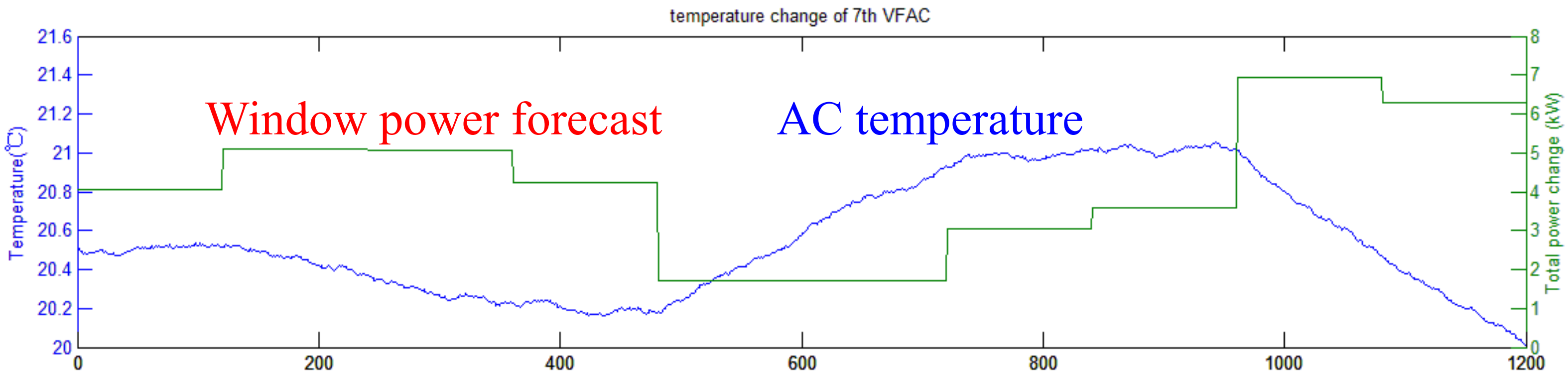
$$\frac{T_1(\infty) - T_1^{SP}}{\Delta T_1} = \frac{T_2(\infty) - T_2^{SP}}{\Delta T_2} = \ldots = \frac{T_n(\infty) - T_n^{SP}}{\Delta T_n}$$

$$P_{1j}(k) + P_{2j}(k) + \cdots + P_{nj}(k) = G_j, \text{ for all } j \text{ and } k$$

Simulation Results

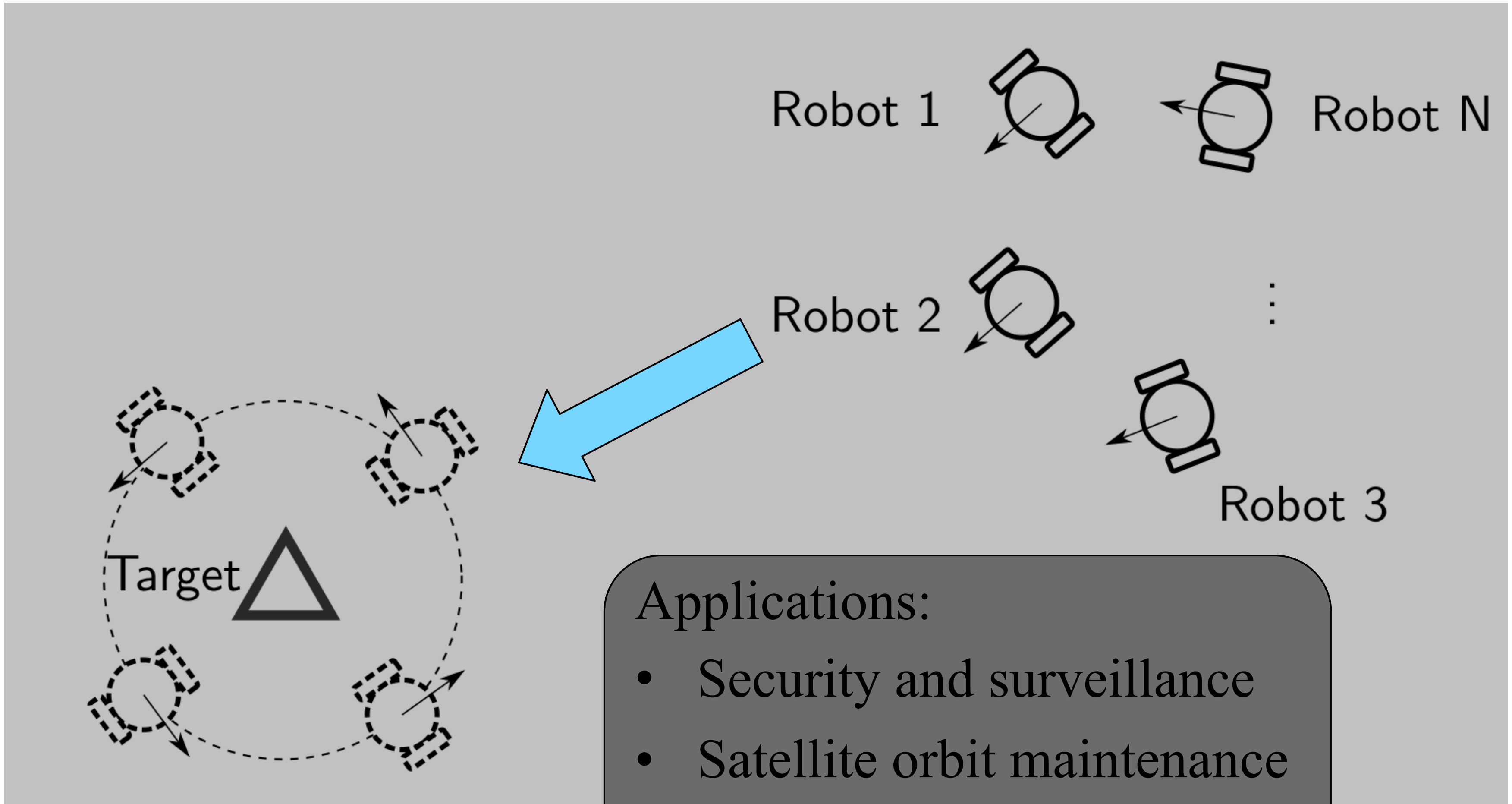
- 10 VFACs, 3 wind power plants.
- Temperature model sampling time: 1min
- Wind forecast interval: 1hour





Distributed Circumnavigation by Unicycles

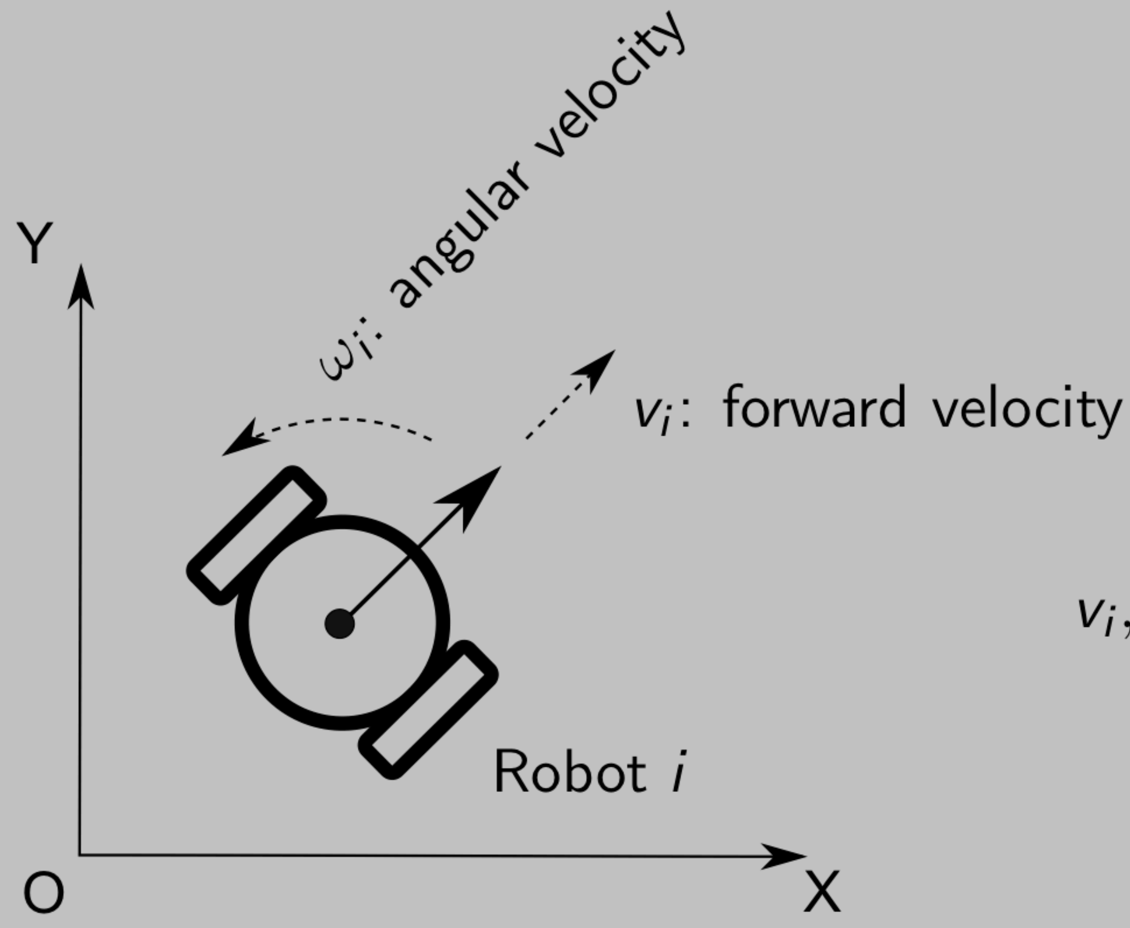
R. Zheng, Z. Lin, M. Fu and D. Sun, *ASCC* 2013



Applications:

- Security and surveillance
- Satellite orbit maintenance
- Source seeking …

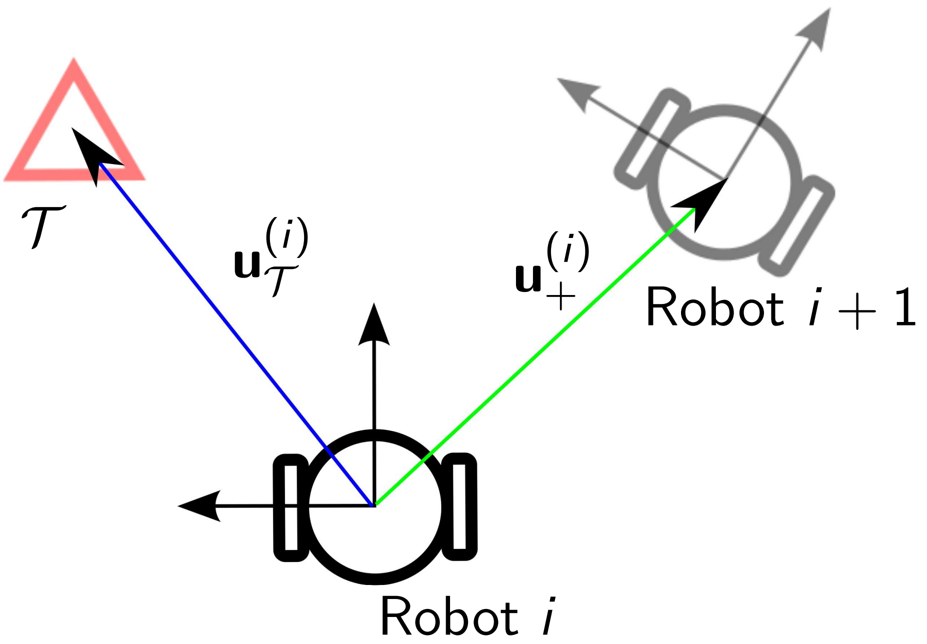
Unicycle-like vehicles are considered :



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

v_i, ω_i: control inputs

Measurements: Each robot i measures the relative positions of the target and another agent $i+1$ modulo N.



$$\mathbf{u}_{\mathcal{T}}^{(i)} = R(\theta_i)\begin{bmatrix} x_{\mathcal{T}} - x_i \\ y_{\mathcal{T}} - y_i \end{bmatrix}$$

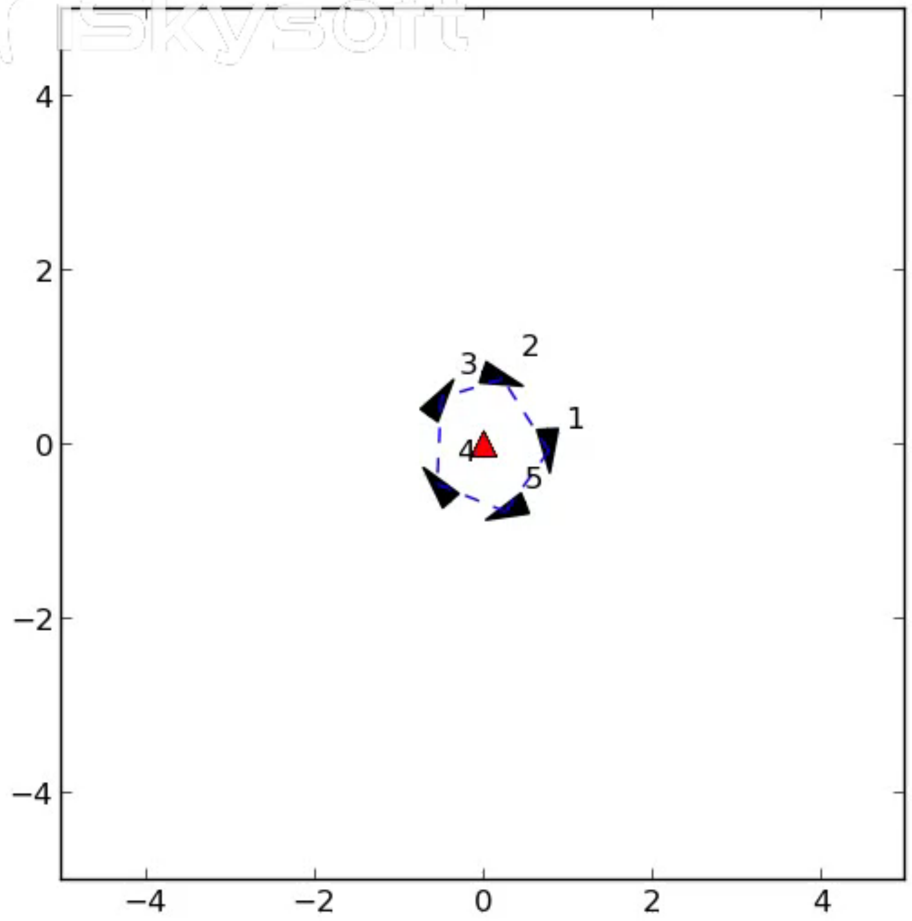
$$\mathbf{u}_{+}^{(i)} = R(\theta_i)\begin{bmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{bmatrix}$$

$$R(\theta_i) = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$$

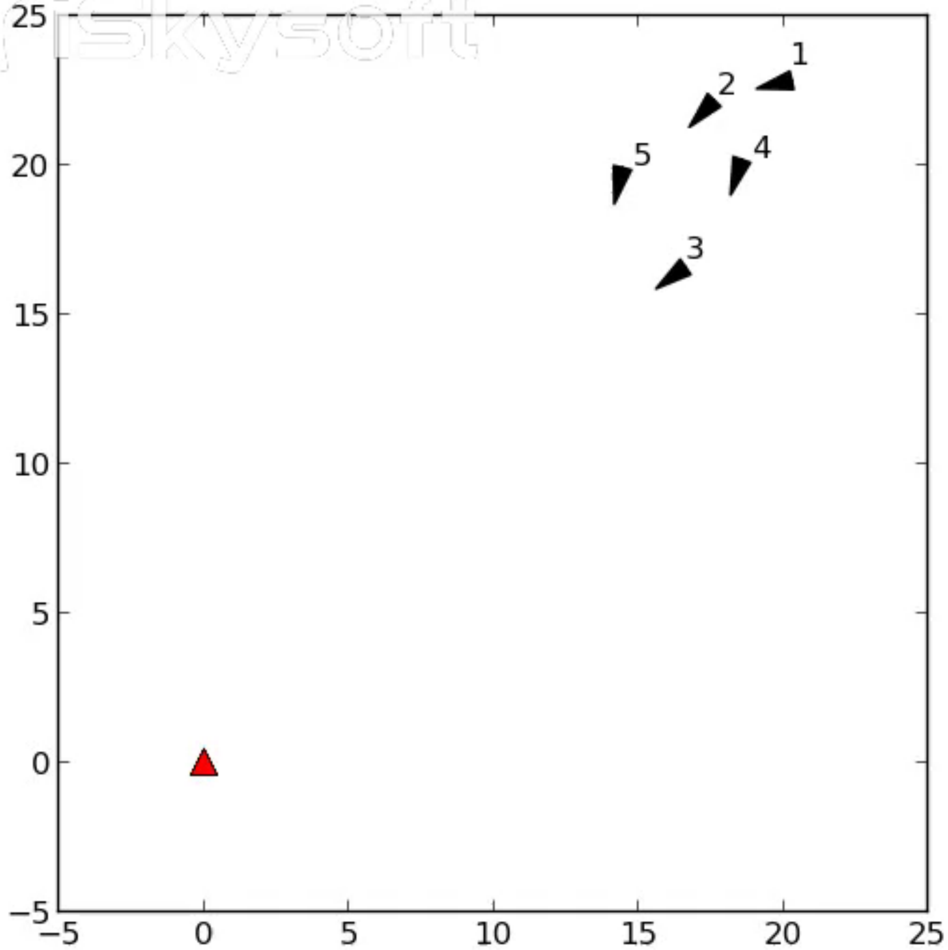
Distributed Control: $$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} k_v & 0 \\ 0 & k_\omega \end{bmatrix}\left[\underbrace{(1-a)\mathbf{u}_{+}^{(i)}}_{\text{repulsion}} + \underbrace{a\mathbf{u}_{\mathcal{T}}^{(i)}}_{\text{attraction}}\right] \quad (a > 1)$$

k_v, k_ω: control gains (> 0) a: control parameter

Simulation Examples



Simulation Examples



Concluding Remarks

- Great opportunities for new control theory and applications
- *Many* exciting and challenging research problems
- Urgency about real, relevant and applicable research
- Multidisciplinary research:
  - Communication network design
  - Network-based control paradigms and algorithms
  - Distributed sensing, sensor fusion and estimation
  - What to learn from other disciplines?
    - Network coding?
    - Belief Propagation?
    - Sparse optimization? …