Applications of Consensus Algorithms to Wireless Sensor Networks

Luca Schenato (Univ. Padova)
Joint work w/

Alessandro Chiuso
University of Padova

Sandro Zampieri
University of Padova

Ruggero Carli
UC Santa Barbara

Ph.D. Students:
Damiano Varagnolo
Simone Del Favero
Saverio Bolognani
University of Padova
Motivations and target applications
Overview of consensus algorithms
Application of consensus to WSN:
  Sensor calibration
  Least-square parameter identification
  Time-synchronization
Open problems
  Identification
  Estimation
  Control
Outline

- Motivations and target applications
- Overview of consensus algorithms
- Application of consensus to WSN:
  - Sensor calibration
  - Least-square parameter identification
  - Time-synchronization
- Open problems
  - Identification
  - Estimation
  - Control
Networked Control Systems

**Drive-by-wire systems**

**Wireless Sensor Networks**

**Swarms robotics**

**Smart structures:**
space telescope & satellites mesh

**Traffic Control:**
Internet and transportation

**Smart materials & MEMS:**
sheets of sensors and actuators

**NCSs:** physically distributed dynamical systems interconnected by a communication network
Wireless Sensor Actuator Networks (WSANs)

- Small devices
  - μController, Memory
  - Wireless radio
  - Sensors & Actuators
  - Batteries
- Inexpensive
- Multi-hop communication
- Programmable (micro-PC)
Applications: Smart Greenhouse

- Distributed estimation
- Distributed control
- Control under packet loss & random delay
- Sensor fusion
- Distributed time synchronization
Applications:
ThermoEfficiency Labeling

- Building thermodynamics model identification
- Sensor selection for identification
- Optimal sensor placement
Applications:
Distributed Localization & Tracking

- Indoor radio signal modeling
- Real-time localization
- Distributed tracking
- Coordination
NCSs: what’s new for control?

Classical architecture: Centralized structure

Diagram:
- Actuators
- Plant
- Sensors
- Controller
NCSs: what’s new for control?

NCSs: Large scale distributed structure

Plant

A
A
A
S
S
S

Connectivity
Limited capacity

Interference
COMMUNICATION
NETWORK
Congestion

Packet loss
Random delay
Quantization

C
C
C
C
C
C
Motivations and target applications

Overview of consensus algorithms

Application of consensus to WSN:
  - Sensor calibration
  - Least-square parameter identification
  - Time-synchronization

Open problems
  - Identification
  - Estimation
  - Control
The consensus problem

Main idea
- Having a set of agents to agree upon a certain value (usually global function) using only local information exchange (local interaction)

Also known as:
- Agreement problem (economics, signal processing, social networks)
- Gossip algorithms (CS & communications)
- Synchronization (statistical mechanics)
- Rendezvous (robotics)

Suitable for (noisy) sensor networks
Main features

- Distributed computation of general functions

\[ \theta = f(x_1, \ldots, x_N) = f \left( \frac{1}{N} \sum_{i=1}^{N} g_i(x_i) \right) \]

(ex. \( \theta = \frac{1}{N} \sum_{i=1}^{N} x_i \) for \( f = g = \text{ident} \))

- Computational efficient (linear & asynchronous)
- Independent of graph topology
- Incremental (i.e. anytime)
- Robust to failure
Some history (in control)

- Convergence of Markov Chains (60’s) and Parallel Computation Alg.(70’s)
- A. Jadbabaie, J. Lin, and A. S. Morse “Coordination of groups of mobile autonomous agents using nearest neighbor rules”, CDC’ 02 (Axelby Best Paper Award TAC)
- Time-varying topologies (worst-case)
- Randomized topologies
  - F. Fagnani, S. Zampieri, “Randomized consensus algorithms over large scale networks”, JSAC 08
- Applications:
  - Vehicle coordination: Jadbabaie, Francis’s group, Tanner, …
  - Kalman Filtering: Olfati Saber-Murray, Alighanbari-How, Carli-Chiuso-Schenato-Zampieri
  - Generalized means: Giarre’, Cortes
  - Time-synchronization: Solis-P.R. Kumar, Osvaldo-Spagnolini, Carli-Chiuso-Schenato-Zampieri
  - WSN sensor calibration and parameter identification: Bolognani-DelFavero-Schenato-Varagnolo
Consensus formulation

Network of

- N agents
- Communication graph $G = (\mathcal{N}, \mathcal{E})$
- i-th node neighbors: $\mathcal{N}(i)$
- Every node stores a variable: node $i$ stores $x_i$. 
Definition (Recursive Distributed Algorithm adapted to the graph $G$)

Any recursive algorithm where the $i$ node’s update law of depends only on the state of $i$ and in its neighbors $j \in \mathcal{N}(i)$

\[ x_i(t + 1) = f(x_i(t), x_{j_1}(t), \ldots x_{j_{N_i}}(t)) \]

with $j_1, \ldots, j_{N_i} \in \mathcal{N}(i)$
Definition

A Recursive Distributed Algorithm adapted to the graph $\mathcal{G}$ is said to asymptotically achieve consensus if

$$x_i(t) \rightarrow \alpha \quad \forall i \in \mathcal{N}$$

Definition

A Recursive Distributed Algorithm adapted to the graph $\mathcal{G}$ is said to asymptotically achieve average consensus if

$$x_i(t) \rightarrow \frac{1}{N} \sum_{i \in \mathcal{N}} x_i(0) \quad \forall i \in \mathcal{N}$$
Linear consensus

\[ x_i(t+1) = p_{ii}x_i(t) + \sum_{j \in \mathcal{N}(i)} p_{ij}x_j(t) \]

\[ x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix} \quad x(t+1) = P(t)x(t) \]

Say \( G_P \) Graph associated to \( P, P_{i,j} \neq 0 \iff (i,j) \in \mathcal{E}_P, \)

\[ G_P \subseteq G \quad (\mathcal{N} \equiv \mathcal{N}_P, \quad \mathcal{E} \subseteq \mathcal{E}_P) \]
A robotics example: the rendezvous problem

\[ x_i(t + 1) = x_i(t) + u_i(t) \]
\[ x_i(t + 1) = p_{ii}x_i(t) + \sum_{j \in N(i)} p_{ij}x_j \]

Convex hull always shrinks.
If communication graph sufficiently connected, then shrinks to a point.
Stochastic matrix

Definition (Stochastic Matrix)
If $P_{i,j} \geq 0$ and $\sum_j P_{i,j} = 1 \forall i$, than $P$ is said to be stochastic

$$P1 = 1$$

$$1 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Remark
If $P$ is stochastic the linear algorithm can be written in both forms:

$$x_i(t + 1) = p_{ii}x_i(t) + \sum_{j \in \mathcal{N}(i)} p_{ij}x_j(t)$$

$$x_i(t + 1) = x_i(t) + \sum_{j \in \mathcal{N}(i)} p_{ij}(x_j(t) - x_i(t))$$
Constant matrix $P$

**Synchronous Communication:**
At each time all nodes communicate according to the communication graph

\[ P(t) = P: \]

\[
\begin{bmatrix}
3/6 & 1/6 & 1/6 & 1/6 & 0 & 0 \\
1/6 & 3/6 & 1/6 & 0 & 0 & 1/6 \\
1/6 & 1/6 & 3/6 & 0 & 1/6 & 0 \\
1/6 & 0 & 0 & 3/6 & 1/6 & 1/6 \\
0 & 0 & 1/6 & 1/6 & 4/6 & 0 \\
0 & 1/6 & 0 & 1/6 & 0 & 4/6
\end{bmatrix}
\]
Convergence results

Theorem

\[ P(t) = P \text{ stochastic.} \]

- **If** \( P \) **such that** \( G_P \subseteq G \) **is rooted** **then the algorithm achieves consensus**
- **If also** \( P^T \) **is stochastic** (\( P \) **doubly stochastic**), **then average consensus is achieved**

\[
P = \begin{bmatrix}
 p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} & 0 & 0 \\
 p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 & p_{2,6} \\
 p_{3,1} & p_{3,2} & p_{3,3} & 0 & p_{3,5} & 0 \\
 p_{4,1} & 0 & 0 & p_{4,4} & p_{4,5} & p_{4,6} \\
 0 & 0 & p_{5,4} & p_{5,3} & p_{5,5} & 0 \\
 0 & p_{6,2} & 0 & p_{6,4} & 0 & p_{6,6}
\end{bmatrix}
\]

\[ p^t \xrightarrow{\lambda^t} 1 \rho^T, \rho \text{ is left eigenvector of } 1 \]
\[ \sum_i \rho_i = 1, \rho_i \geq 0, \quad (\rho_i > 0 \text{ if strong. conn.}) \]
\[ \rho = \frac{1}{N} 1 \text{ if } P \text{ doubly stochastic} \]
Time varying $P(t)$: broadcast

Broadcast:
At each time one node randomly wakes up and broadcasts its information to all its neighbors.

$$P(t) = \begin{bmatrix} \frac{3}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{3}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{3}{4} \end{bmatrix}$$
Symmetric Gossip:

At each time one node randomly wakes up and chose randomly a its neighbor. Those two nodes exchange information

\[
P(t) = \begin{bmatrix}
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]
Consensus strategies for WSN

Broadcast
- 1 message broadcasted, $|\mathcal{N}(i)|$ estimate updated
- Does not guarantee average consensus

Symmetric Gossip
- At least 3 messages exchanged, 2 estimate updated
- Guarantee average consensus
Convergence results: 
P=P(t) deterministic

Theorem

Suppose that $P_{ii}(t) > 0, \forall i, \forall t$ and that there exists $K$ such that 
$G_\ell = G_{P[(\ell+1)K]} \cup \ldots \cup G_{P(\ell K)}$ is rooted at some node $j$ for all $\ell$ then

- the sequence $\{P(t)\}$ achieves consensus
- if also $P^T(t)$ are stochastic for all $t$, then the sequence $\{P(t)\}$ achieves average consensus

Remark:

Estimates of rate of convergence are very conservative (worst case)

Convergence results: \( P = P(t) \) randomized

**Theorem**

Suppose \( \{P(t)\} \) is a sequence of i.i.d. stochastic random matrices. Suppose moreover \( \mathcal{G}_{P(t)} \subseteq \mathcal{G} \ \forall t \) and call \( \bar{P} = \mathbb{E}[P] \).

- If \( \mathcal{G}_{\bar{P}} \) is rooted that consensus is achieved \( \text{w.p.1} \)
- If also \( P(t)^T \) is stochastic for every \( t \), then average consensus is achieved \( \text{w.p.1} \)

**Remark:**

It is **not** sufficient \( \bar{P} \) doubly stochastic to guarantee average consensus

\[
x(t + 1) = P(t)x(t) = P(t)P(t - 1) \cdots P(0)x(0) = Q(t)x(0) \quad (Q(t) = P^t \text{ if } P(t) = P)
\]

\[
Q(t) \to 1 \rho^T, \quad \mathbb{E}[\rho] = \frac{1}{N} 1, \quad \text{Var}(\rho) \sim \frac{1}{N}
\]

---

Generalized mean

\[ x_i(t + 1) = p_{ii}x_i(t) + \sum_{j \neq i} p_{ij}x_j(t) \quad x_i(t) \rightarrow \frac{1}{N} \sum_i x_i(0) \]

\[ \theta = f(a_1, \ldots, a_N) = f \left( \frac{1}{N} \sum_{i=1}^N g_i(a_i) \right) \]

Geometric mean: \[ \theta = \sqrt[\pi]{\prod_i a_i} = \exp \left( \frac{1}{N} \sum_i \log(a_i) \right) \]

\[ x_i(0) = \log(a_i), \quad \hat{\theta}_i(t) = \exp(x(t)) \rightarrow \theta \]

Armonic mean: \[ \theta = \left( \frac{1}{N} \sum_i \frac{1}{a_i} \right)^{-1} \quad x_i(0) = \frac{1}{a_i}, \quad \hat{\theta}_i(t) = \frac{1}{x_i(t)} \rightarrow \theta \]

Quadratic mean: \[ \theta = \sqrt{\frac{1}{N} \sum_i a_i^2} \quad x_i(0) = a_i^2, \quad \hat{\theta}_i(t) = \sqrt{x_i(t)} \rightarrow \theta \]


J. Cortés, Distributed algorithms for reaching consensus on general functions, Automatica 44 (3) (2008), 726-737
Node counting

\[ x_i(t + 1) = p_{ii}x_i(t) + \sum_{N_j} p_{ij}x_j(t) \quad x_i(t) \to \frac{1}{N} \sum_i x_i(0) \]

\[ x_1(0) = 1, \quad x_i(0) = 0, i = 2, \ldots, N \quad \Rightarrow \quad \frac{1}{N} \sum_i x_i(0) = \frac{1}{N} \]

\[ \hat{\theta}_i(t) = \frac{1}{x_i(t)} \to N \]
Outline

- Motivations and target applications
- Overview of consensus algorithms
- Application of consensus to WSN:
  - Sensor calibration
  - Least-square parameter identification
  - Time-synchronization
- Open problems
  - Identification
  - Estimation
  - Control
Localization with WSN
Localization with WSN

Each node can measure the Radio Signal Strength Indicator, RSSI, i.e. the received signal power $P_{rx}$ in dBm.

$$P_{ij} = P_{tx}^i + \beta - 10\gamma \log_{10}(||x_i - x_j||) + f_{sf}(x_i, x_j) + v(t) + o_i$$

- **Map Based**
- **Transmi pow**
- **Most likely location that matches with pre-learned maps.**

- **Range based**
- **Triangulation (similarly to GPS)**

$f_{sf}(x_i, x_j) = f_{sf}(x_j, x_i)$
Offset effect

Reception offset is particularly harmful for localization applications, Experiment inside a basketball court.\[S\ 07]\textsuperscript{2}

WSN sensor calibration

Ideally:

- Estimate \( o_i \): \( \hat{o}_i \)
- Use \( \hat{o}_i \) to compensate the offset: \( o_i - \hat{o}_i = 0 \)

What we propose is:

\[
\alpha \approx 0 \quad \text{equal for all nodes}
\]

All nodes overestimate or underestimate the distance similarly. The errors, in the triangulation process, cancel out partially.
Calibration as consensus problem

\[ x_i(t + 1) = p_{ii}x_i(t) + \sum_{j \in N(i)} p_{ij}x_j(t) \]
\[ x_i(t + 1) = x_i(t) + \sum_{j \in N(i)} p_{ij}(x_j(t) - x_i(t)) \quad o_i - \hat{o}_i(t) = x(t) \]

Remark
If \( P \) is stochastic the linear algorithm can be written in both forms:

\[ o_i - \hat{o}_i(t + 1) = o_i(t) - \hat{o}_i(t) + \sum_{j \in N_i} p_{ij} ((o_i - \hat{o}_i(t)) - (o_j - \hat{o}_j(t))) \]

\[ \hat{o}_i(t + 1) = \hat{o}_i(t) - \sum_{j \in N_i} p_{ij} (P^{ij} - P^{ji} - \hat{o}_i(t) + \hat{o}_j(t)) \]

\[ \hat{o}_i(t) \rightarrow o_i - \frac{1}{N} \sum_i o_i = o_i - \alpha \approx o_i \]
25 TMote-Sky nodes with Chipcon CC2420 RF Transceiver randomly placed inside a single conference room:

Network topology and nodes displacement:

Kept just the links that safely carried the 75% of the sent messages over them
Experimental results

Links divided in 2 categories:
- Training links (black)
- Validation links (gray)

Estimate time evolution
\[ \Delta \bar{P}^{ij} = \bar{P}^{ij} - \bar{P}^{ji} = o_i - o_j \]

<table>
<thead>
<tr>
<th></th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0.5 dB</td>
<td>24%</td>
<td>56%</td>
</tr>
<tr>
<td>&lt;1</td>
<td>50%</td>
<td>88%</td>
</tr>
<tr>
<td>&gt;2dB</td>
<td>35%</td>
<td>0.6%</td>
</tr>
<tr>
<td>Max</td>
<td>&lt;6dB</td>
<td>&lt;3.5dB</td>
</tr>
</tbody>
</table>

Effects of systematic errors when estimating distances

1dB \[\implies \cong 2m \pm 0.28m.\]

6dB \[\implies \text{uncertainty for 0.9m to 4.4m for an actual distance of 2m}.\]

1dB \[\implies \cong 10m \pm 1.4m.\]
Another important problem:
Accurately identify the wireless channel parameters $\beta$ and $\gamma$.

In fact:
- Parameters extremely environment dependent
- $\gamma \in [1, 6]$
- Environment change hourly or daily

$$P_{rx}^{ij} = P_{tx}^j + \beta - 10\gamma \log_{10}(||x_i - x_j||) + f_{sf}(x_i, x_j) + v(t) + o_i$$
Modeling

Recall the Wireless Channel Model

\[ \tilde{P}_{rx}^{ij} + \hat{o}_i = P_{tx} - \beta - \gamma 10 \log_{10}(d_{ij}) + f_{sf}(x_i, x_j) + (o_i + \hat{o}_i) + w_i \]

\[ \tilde{P}_{rx}^{ij} + \hat{o}_i = \beta - \gamma 10 \log_{10}(d_{ij}) + w_i \]

For each link:

\[ \underbrace{\tilde{P}_{rx}^{ij} + \hat{o}_i}_{b_{ij}} = \underbrace{[1 - 10 \log_{10}(d_{ij})]}_{a_{ij}^T} \begin{bmatrix} \beta \\ \gamma \\ \theta \end{bmatrix} + w_i \]
Modeling (cont’d)

Each node

- knows its distance with its neighbor
  \[ d_{ij} \rightarrow a_{ij} \]
- measures the strength of the message received from its neighbors
  \[ P_{ij} \rightarrow b_{ij} \]

Globally, the network collected \( M \) couples measure-regressors:
\((a_1, b_1), \ldots (a_M, b_M)\)

For ease of notation, assume that
Each node stores one couple measure-regressor.
Globally, the sensor network collected $M$ couples measure-regressors: $(a_1, b_1), \ldots, (a_M, b_M)$.

Let us call

$$A = [a_1, \ldots, a_M]^T \quad \text{and} \quad b = [b_1, \ldots, b_M] .$$

$$b = A\theta + w$$

The least square estimate of $\theta$, given the measurements $b$ is

$$\hat{\theta} = \arg\min_{\theta} \| A\theta - b \| = (A^T A)^{-1} A^T b$$
\[ \hat{\theta} = \arg \min_{\theta} \| A\theta - b \| = (A^T A)^{-1} A^T b = \left( \frac{1}{N} \sum_{i \in N} a_i a_i^T \right)^{-1} \left( \frac{1}{N} \sum_{i \in N} a_i b_i \right) \]
Communication schemes

Broadcast

- 1 message broadcasted, $|\mathcal{N}(i)|$ estimate updated
- Does not guarantee average consensus

Symmetric Gossip

- At least 3 messages exchanged, 2 estimate updated
- Guarantee average consensus
Experimental results

Residual: $\frac{1}{M} \| A\hat{\theta} - b \|^2$
Tracking results
Time synchronization in sensor networks

Diagram showing a sensor network with a BASE STATION and sensor nodes. The diagram illustrates the concept of transmission between Node i and Node j with ON and OFF states.
Clock characteristics & standard clock pair synch

- **Offset**: instantaneous time difference
- **Skew**: clock speed
- **Drift**: derivative of clock speed

\[ \tau_i = a_i t + b_i \]

- **Offset synch**: periodically remove offset with respect to reference clock
- **Skew compensation**: estimate relative speed with respect to reference clock
State-of-the-art

Tree-based sync

Cluster-based sync

Distributed

Root

Single-hop clusters

Gateways

Nodes

Communication links

Nodes
MODEL: \( N \) clocks as discrete time integrators

\[
x_i(t + 1) = x_i(t) + d_i
\]

\( d_i : \text{skew (clock speed)} \)

\( x_i(0) = \beta_i : \text{initial offset} \)

CONTROL: Assume that it is possible to control each clock by a local input \( u_i(t) \):

\[
x_i(t + 1) = x_i(t) + d_i + u_i(t)
\]

GOAL: Clocks Synchronization

\[
\lim_{t \to \infty} x_i(t) - x_j(t) = 0
\]

\[
\lim_{t \to \infty} y(t) = \lim_{t \to \infty} \left( I - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) x(t) = 0
\]

CONTROL: Proportional controller

\[
u_i(t) = - \sum_{j \in \mathcal{N}(i)} k_{ij} (x_j(t) - x_i(t))
\]

\[
u(t) = -Kx(t)
\]
$$x(t + 1) = x(t) + d + u(t)$$

$$u(t) = -Kx(t)$$

$$\lim_{t \to \infty} y(t) = \lim_{t \to \infty} \left( I - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right)x(t) = 0$$

**If $K$ symmetric:**

Eigenvalues of $K \in \mathbb{R}$

$$h = 1$$

$$h = 2, \ldots, N$$
\[
x(t + 1) = x(t) + d + u(t)
\]
\[
w(t + 1) = w(t) - \alpha K x(t)
\]
\[
u(t) = w(t) - K x(t)
\]
\[
\lim_{t \to \infty} y(t) = \lim_{t \to \infty} \left( I - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) x(t)
\]

If \( K \) symmetric:

\[
h = 1
\]

\[
h = 2, \ldots, N
\]
\[ x(t + 1) = x(t) + d + u(t) \]

\[ u(t) = C(z)Kx(t) \]

\[ \lim_{t \to \infty} y(t) = \lim_{t \to \infty} \left( I - \frac{1}{N}11^T \right) x(t) \]

If \( K \) is symmetric:

\[ \text{eigenvalues of } K \in \mathbb{R} \]

\[ h = 1 \]

\[ h = 2, \ldots, N \]
Parameter design (undirected graphs)

GOAL: fastest rate of convergence

\[
\begin{bmatrix}
    x(t+1) \\
    w(t+1)
\end{bmatrix} =
\begin{bmatrix}
    I - K & I \\
    -\alpha K & I
\end{bmatrix}
\begin{bmatrix}
    x(t) \\
    w(t)
\end{bmatrix} +
\begin{bmatrix}
    d \\
    0
\end{bmatrix}
\]

- Suboptimal design (no topology needed):

\[
k_{i,j} = -\frac{1}{\max(d_i, d_j)+1} \quad i \neq j, \quad \alpha = \frac{1}{2}, \quad \text{where } d_i \text{ is } \# \text{ of neighbors of node } i.
\]

- Optimal design: almost convex problem (SDP + 1D non-convex search)
Model w/ noise

\[ u_i(t) = - \sum_{j \in \mathcal{N}(i)} k_{ij} (x_j(t) - x_i(t)) \]

white measurement noise white process noise

\[
\begin{bmatrix}
    x(t+1) \\
    w(t+1)
\end{bmatrix} = \begin{bmatrix}
    I - K & I \\
    -\alpha K & I
\end{bmatrix} \begin{bmatrix}
    x(t) \\
    w(t)
\end{bmatrix} + \begin{bmatrix}
    -K \\
    -\alpha K
\end{bmatrix} v(t) + \begin{bmatrix}
    d \\
    0
\end{bmatrix} + \begin{bmatrix}
    I \\
    0
\end{bmatrix} n(t)
\]

GOAL: smallest steady state mean square error: \[ J(K, \alpha) = \frac{1}{N} E[||y(\infty)||^2] \]

- Suboptimal design still OK

\[ k_{ij} = \frac{1}{\max(d_i, d_j)+1}, \quad \alpha = \frac{1}{2}, \quad \text{where } d_i \text{ is } \# \text{ of neighbors of node } i. \]

- Optimal design: almost convex problem (Semidefinite programming in K+ 1D non-convex search in \( \mathcal{F} \))
Simulations

Model parameters based on experimental data from real WSN and pseudo-synchronous implementation

\[ w_i(t) \]

\[ y_i(t) \]

\[ w_i(\infty) + d_i = w_j(\infty) + d_j \]

\[ y_i(\infty) = 0 \]
Outline

- Motivations and target applications
- Overview of consensus algorithms
- Application of consensus to WSN:
  - Sensor calibration
  - Least-square parameter identification
  - Time-synchronization
- Open problems
  - Identification
  - Estimation
  - Control
Which problem can be casted as a consensus problem?

- Kalman filtering
- unbiased broadcast communication
- 

Consensus algorithms == optimization tool
Identification: large scale structured systems

\[ x \sim \mathcal{N}(0, \Sigma), \quad \Sigma^{-1} \text{ is sparse (graph model)} \]

- \( \Sigma \) only partially known and noisy \( \Rightarrow \Sigma^{-1} \) is full.
- communication graph \( \neq \) correlation graph
- weak correlation, i.e. \( \Sigma^{-1} \) full w/ some small entries \( \Rightarrow \) Graph identifiability
- what if dynamics also, i.e. \( x_{t+1} = Ax_t + w_t \) ?
- if a node dies, i.e. remove row-column from \( \Sigma \), how to compute \( \Sigma^{-1} \) ?
- how to do model reduction preserving graph structure ?
- is consensus relevant ?

Carlos Carvalho "Structure and Sparsity in High-Dimensional Multivariate Analysis", Ph.D. Theis, Duke Univ., 2007
Identification/Estimation of infinite dimensional space \( f : R^n \rightarrow R \).

**Centralized** learning: \( \hat{f}(\cdot) = \sum_{n=1}^{N} \alpha_n \Phi(x_n, \cdot) \)

**Totally decentralized** learning: \( \hat{f}_i(\cdot) = \sum_{n=1}^{N_i} \alpha_n^i \Phi(x_n, \cdot), \quad N_i \ll N \)

What to exchange?
- all \((x_i, f(x_i))\) of neighbors?
- most informative \((x_i, f(x_i))\) of neighbors?
- smoothed observation of neighbors \((x_i, \hat{f}_i(x_i))\)
- virtual observations \(\hat{f}(\hat{x}_i, \hat{f}_i(\hat{x}_i))\)

Time synchronization example:

$P_{dist}$ symmetric: slow convergence but robust

$P_{hier}$ asymmetric: fast convergence but fragile to node failure

$P_{soft} = \alpha P_{dist} + (1 - \alpha) P_{hier}$, optimal $\alpha$ depends on failure rate
THANK YOU