Simone Del Favero

PhD Thesis dissertation

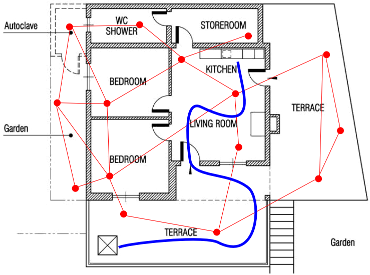
Title: Analysis and Development of Consensus–based Estimation Schemes.

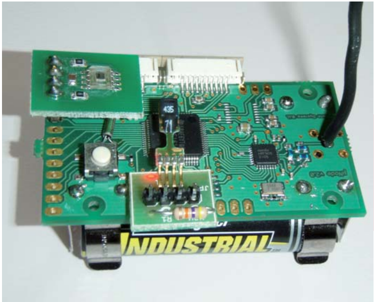
Advisor: Sandro Zampieri

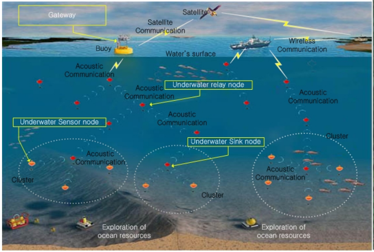
e-mail: simone.delfavero@dei.unipd.it

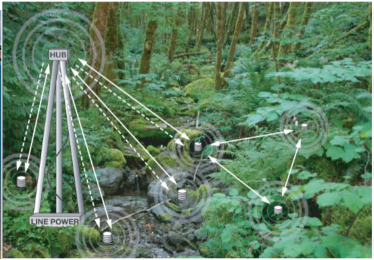
Outline of the talk

Sensor Networks







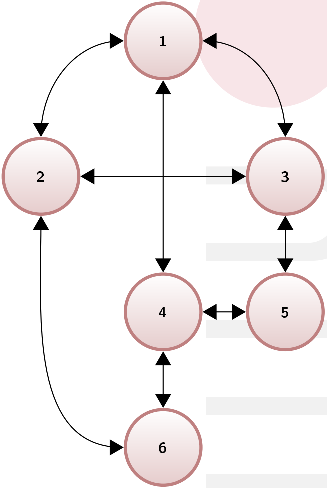


What is Consensus?

The solutions we consider are based on Consensus Algorithms.
What is Consensus?

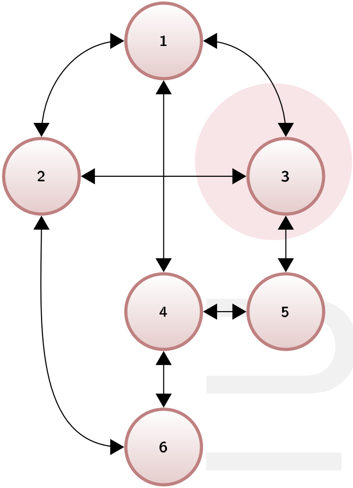
Network of

- N agents
- Communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
- i-th node neighbors: $\mathcal{N}(i)$
- Every node stores a variable:
  node i stores x_i.



Definition

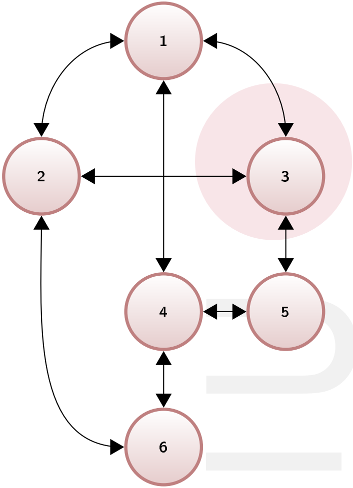
We call Recursive Distributed Algorithm adapted to the graph $\mathcal{G}$ any recursive algorithm where the i node's update law of depends only on the state of i and in its neighbors $j \in \mathcal{N}(i)$



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

Definition

We call **Recursive Distributed Algorithm adapted to the graph $\mathcal{G}$** any recursive algorithm where the i node's update law of depends only on the state of i and in its neighbors $j \in \mathcal{N}(i)$

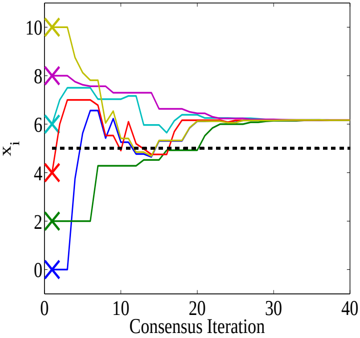


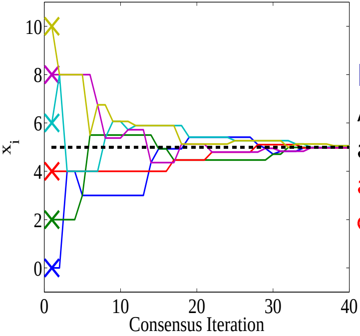
$$x(t+1) = P(t)x(t) \qquad P_{i,j} \neq 0 \Rightarrow (j,i) \in \mathcal{E}$$

Definition

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

$$x_i(t) \longrightarrow \alpha \qquad \forall i \in \mathcal{N}$$





Definition

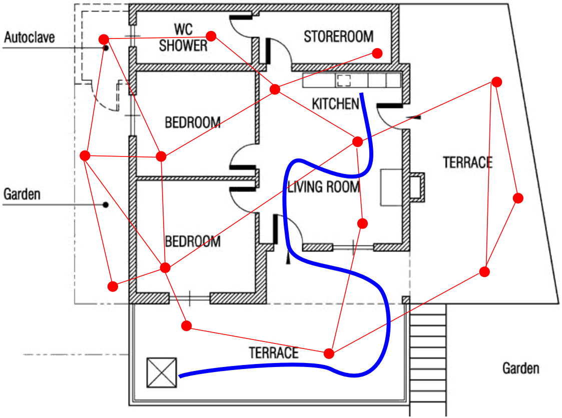
A Recursive Distributed Algorithm adapted to the graph $\mathcal{G}$ is said to asymptotically achieve *average* consensus if

$$x_i(t) \longrightarrow \frac{1}{N} \sum_{i \in \mathcal{N}} x_i(0) \qquad \forall i \in \mathcal{N}$$

An important application

Aim of the work:

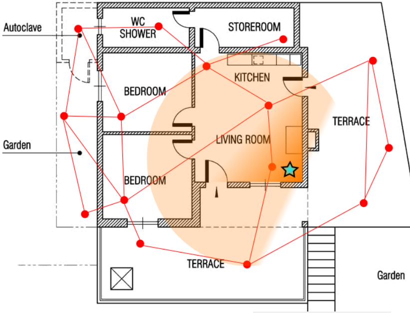
To address some modeling and algorithmic issues of localization and target tracking in wireless sensors networks

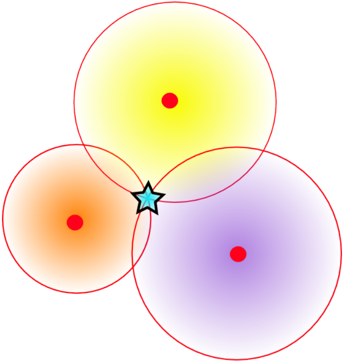


First, evaluate the distances nodes-the moving object. Then, reconstruct absolute moving object position

Map Based

Most likely location that matches with pre-learned maps.



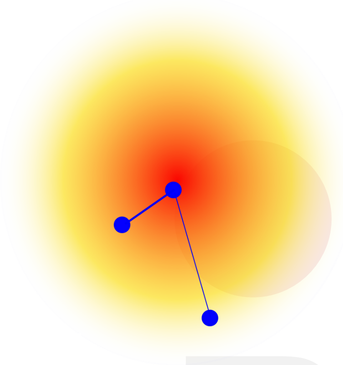


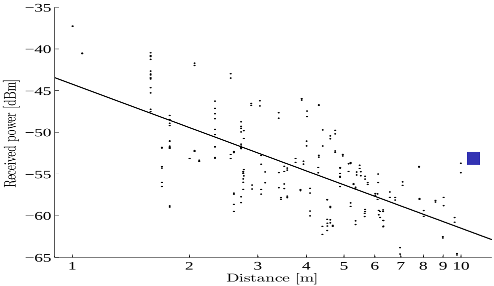
Range based

Triangulation (similarly to GPS)

Localization and tracking, the idea:

- Nodes measure the radio signal strength of the received packet
- It depends on the distance tx-rx.





- From the distance it can be estimated the position of the nodes

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:

$o_i - \hat{o}_i = \alpha \qquad \alpha \cong 0 \qquad$ equal for all nodes

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

Sensor Calibration as a Consensus Problem

Asking

$$o_i - \hat{o}_i = \alpha \text{ equal for all nodes}$$

means to cast our sensor calibration problem into a consensus problem

Consider the consensus algorithm

$$(o_i - \hat{o}_i)^+ = (o_i - \hat{o}_i) + \sum_{j \in \mathcal{V}(i)} p_{ij}((o_j - \hat{o}_j) - (o_i - \hat{o}_i))$$

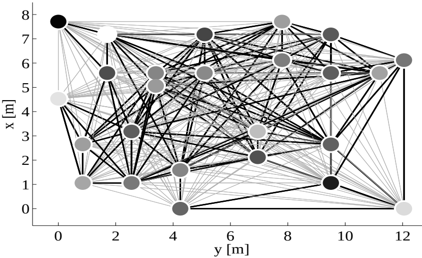
That leads to

$$\hat{o}_i^+ = \hat{o}_i - \sum_{j \in \mathcal{V}(i)} p_{ij}(\bar{P}_{rx}^{ji} - \bar{P}_{rx}^{ij} + \hat{o}_j - \hat{o}_i)$$

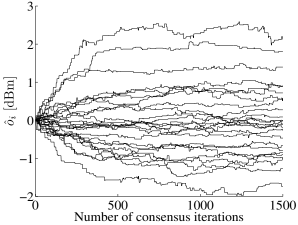
Experimental Results

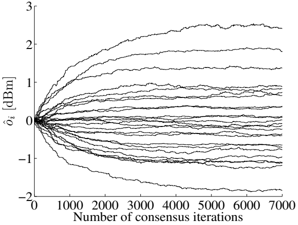
Links divided in 2 categories:

- Training links (black)
- Validation links (gray)



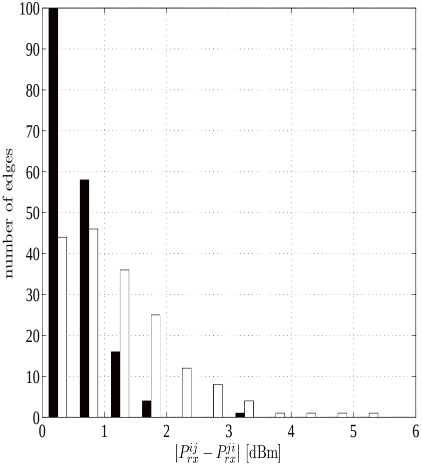
Estimate time evolution





Experimental Results:
Asymmetric error before and after offset correction

$$\Delta \bar{P}^{ij} = \bar{P}^{ij} - \bar{P}^{ji} = o_i - o_j$$



	Before	After
<1	50%	88 %
>2dB	35%	0.6 %

Effects of systematic errors when estimating distances

$1dB \longmapsto \cong 2m \pm 0.28m$.
$6dB \longmapsto$ uncertainty for $0.9m$ to $4.4m$ for an actual distance of $2m$.

Kalman Filter:

Carli et al.

R. Carli, A. Chiuso, L. Schenato and S. Zampieri (2008). *Distributed Kalman filtering using consensus strategies* IEEE Journal on Selected Areas in Communications, 26(4), pp. 622-633.

Standard

Let us consider a simple case: scalar random walk

$$\begin{cases} x(t+1) & = x(t) + w(t) \\ y_i(t) & = x(t) + v_i(t) \end{cases}$$

The local Kalman Filter looks like

$$\hat{x}_i(t+1) = (1-\ell)\hat{x}_i(t) + \ell y_i(t+1)$$

A distributed algorithm

2 phases algorithm

$$\hat{x}_i^{\text{loc}}(t+1) = (1-\ell)\hat{x}_i(t) + \ell y_i(t+1)$$

$$\hat{x}_i(t) = \alpha_{ii}(t)\hat{x}_i^{\text{loc}}(t) + \sum_{j\in\mathcal{N}(i)} \alpha_{ij}(t)\hat{x}_{\text{loc}_j}(t)$$

Kalman Filter:

Carli et al.

R. Carli, A. Chiuso, L. Schenato and S. Zampieri (2008). *Distributed Kalman filtering using consensus strategies* IEEE Journal on Selected Areas in Communications, 26(4), pp. 622-633.

Standard

Let us consider a simple case: scalar random walk

$$\begin{cases} x(t+1) & = x(t) + w(t) \\ y_i(t) & = x(t) + v_i(t) \end{cases}$$

The local Kalman Filter looks like

$$\hat{x}_i(t+1) = (1-\ell)\hat{x}_i(t) + \ell y_i(t+1)$$

A distributed algorithm

2 phases algorithm (rewritten using matrices)

$$\hat{x}^{\mathrm{loc}}(t+1) = (1-\ell)\hat{x}(t) + \ell y(t+1)$$

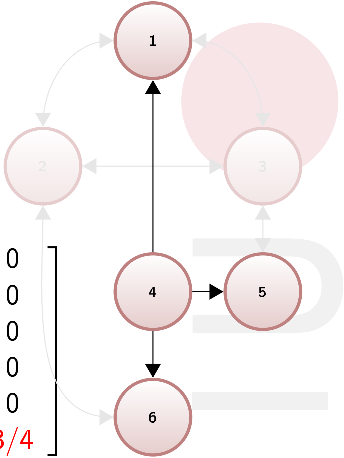
$$\hat{x}(t) = P(t)\hat{x}^{\mathrm{loc}}(t)$$

Results on Convergence, P random matrix

Broadcast:

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

$$P(t) = \begin{bmatrix} 3/4 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 0 & 1/4 & 0 & 3/4 \end{bmatrix}$$



Algorithm Analysis

i-th Node Estimation Error : $\tilde{x}_i(t) = x(t) - \hat{x}_i(t)$.

Estimation Error Variance:

$$\Sigma(t+1) = (1-\ell)^2 \mathbb{E}\left[P(t)\Sigma(t)P^T(t)\right] + \ell^2 r \mathbb{E}\left[P(t)P^T(t)\right] + q\mathbb{1}\mathbb{1}^T.$$

The quantity we are interested to

$$\lim_{t\to\infty} \Sigma(t)$$

No close form expression for such a quantity.
We propose an upperbound:

$$\frac{1}{N}\sum_{j=0}^{N-1} l^2 r \frac{\lambda_j\left(\mathbb{E}\left[P(t)P^T(t)\right]\right)}{1-(1-l)^2\lambda_j\left(\mathbb{E}\left[P(t)P^T(t)\right]\right)} + \frac{q}{1-(1-l)^2}$$

Algorithm Analysis

i-th Node Estimation Error : $\tilde{x}_i(t) = x(t) - \hat{x}_i(t)$.

Estimation Error Variance:

$$\Sigma(t+1) = (1-\ell)^2 \mathbb{E}\left[P(t)\Sigma(t)P^T(t)\right] + \ell^2 r \mathbb{E}\left[P(t)P^T(t)\right] + q\mathbb{1}\mathbb{1}^T.$$

The quantity we are interested to

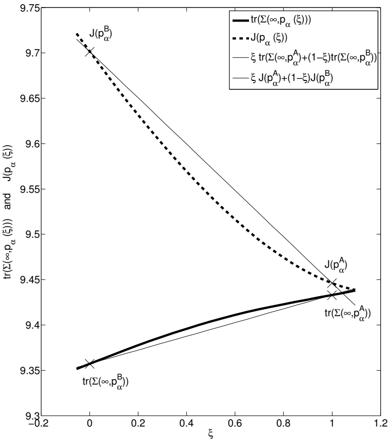
$$\frac{1}{N}\lim_{t\to\infty} \mathrm{tr}\Sigma(t)$$

No close form expression for such a quantity.
We propose an upperbound:

$$\frac{1}{N}\sum_{j=0}^{N-1} l^2 r \frac{\lambda_j\left(\mathbb{E}\left[P(t)P^T(t)\right]\right)}{1-(1-l)^2\lambda_j\left(\mathbb{E}\left[P(t)P^T(t)\right]\right)} + \frac{q}{1-(1-l)^2}$$

Design

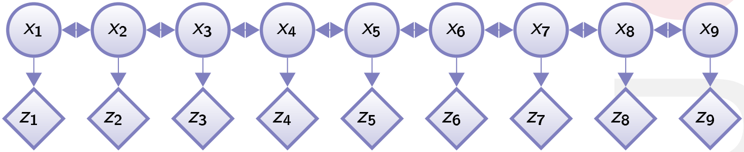
Minimizing this quantity with respect to the link selection probability is a convex optimization problem.
On the contrary, the original cost function is not convex with respect to the link selection probability.



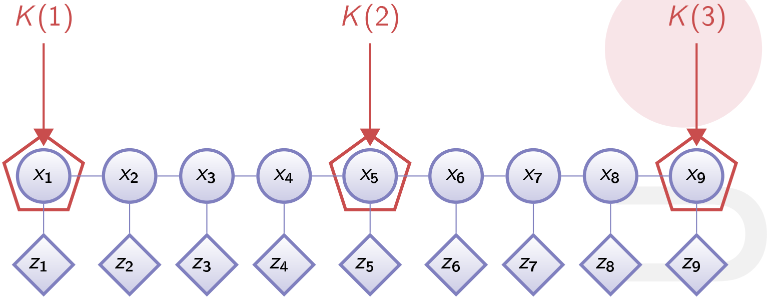
Aim:

Estimation of different but correlated quantities.

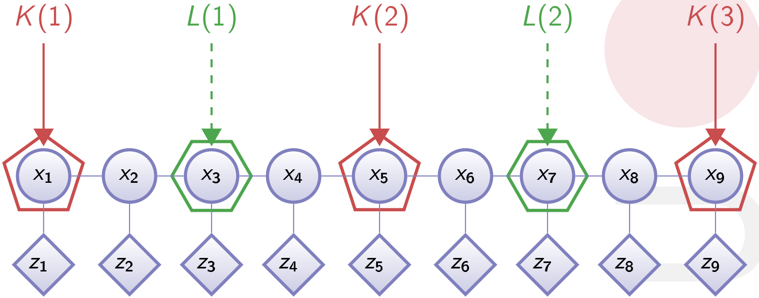
Kalman smoother on a Gauss Markov Random Field



N nodes, p groups of J elements. $N = pJ + 1$.



N nodes, p groups of J elements. $N = pJ + 1$.



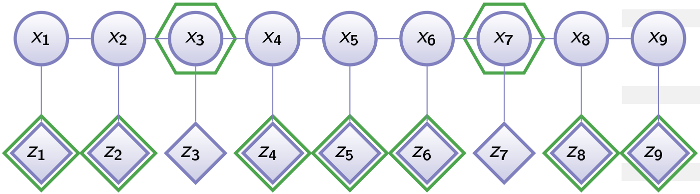
Algorithm

Step 1: Initialization

Second partition is active.

Node $K(j)$, the middle node of the group, computes an estimate of its state based only the measurements collected by the nodes of its group $Z_{L(j-1,j)}$.

$$x^0_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)}]$$



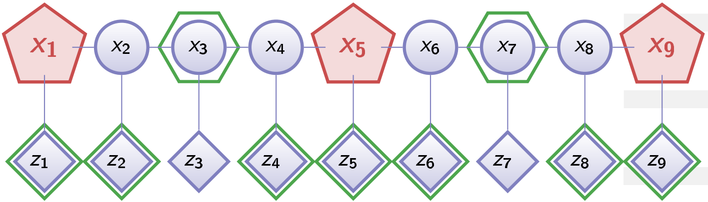
Algorithm

Step 1: Initialization

Second partition is active.

Node $K(j)$, the middle node of the group, computes an estimate of its state based only the measurements collected by the nodes of its group $Z_{L(j-1,j)}$.

$$x^0_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)}]$$



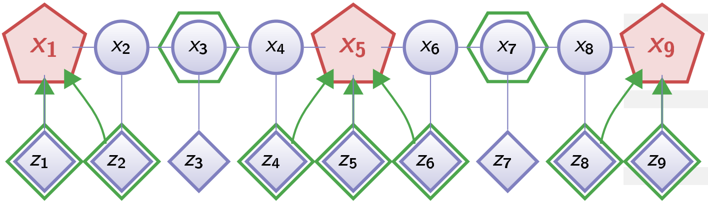
Algorithm

Step 1: Initialization

Second partition is active.

Node $K(j)$, the middle node of the group, computes an estimate of its state based only the measurements collected by the nodes of its group $Z_{L(j-1,j)}$.

$$x^0_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)}]$$



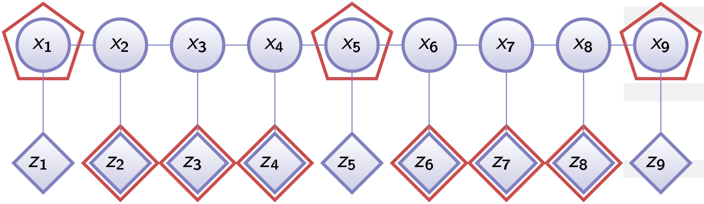
Algorithm:

Step 2:

First partition is active.

Node $L(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{K(j,j+1)}$ and assuming that $x_{K(j)}$ and $x_{K(j+1)}$ were exactly $X^{\ell}_{K(j)}$ and $X^{\ell}_{K(j+1)}$.

$$X^{\ell+1}_{L(j)} = \mathbb{E}[x_{L(j)} \mid Z_{K(j,j+1)} \, , X_{K(j,j+1)} = X^{\ell}_{K(j,j+1)} \,]$$



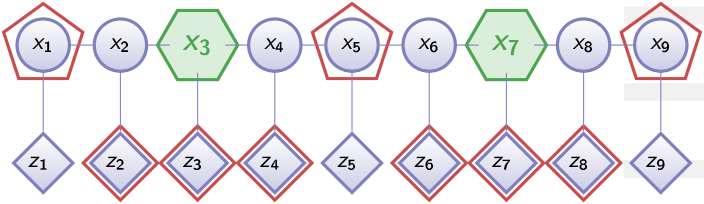
Algorithm:

Step 2:

First partition is active.

Node $L(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{K(j,j+1)}$ and assuming that $x_{K(j)}$ and $x_{K(j+1)}$ were exactly $X^{\ell}_{K(j)}$ and $X^{\ell}_{K(j+1)}$.

$$X^{\ell+1}_{L(j)} = \mathbb{E}[x_{L(j)} \mid Z_{K(j,j+1)}\, , X_{K(j,j+1)} = X^{\ell}_{K(j,j+1)}\,]$$



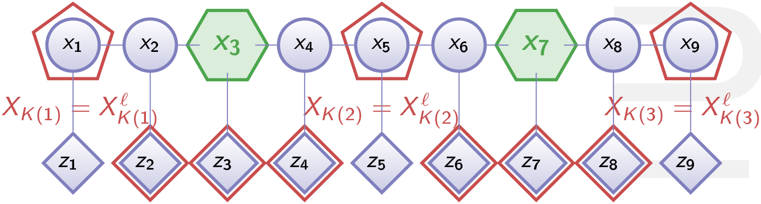
Algorithm:

Step 2:

First partition is active.

Node $L(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{K(j,j+1)}$ and assuming that $x_{K(j)}$ and $x_{K(j+1)}$ were exactly $X^{\ell}_{K(j)}$ and $X^{\ell}_{K(j+1)}$.

$$X^{\ell+1}_{L(j)} = \mathbb{E}[x_{L(j)} \mid Z_{K(j,j+1)} \, , X_{K(j,j+1)} = X^{\ell}_{K(j,j+1)} \,]$$



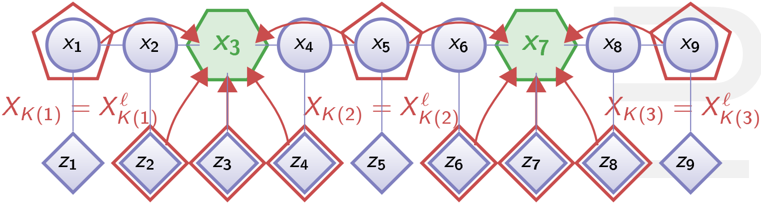
Algorithm:

Step 2:

First partition is active.

Node $L(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{K(j,j+1)}$ and assuming that $x_{K(j)}$ and $x_{K(j+1)}$ were exactly $X^{\ell}_{K(j)}$ and $X^{\ell}_{K(j+1)}$.

$$X^{\ell+1}_{L(j)} = \mathbb{E}[x_{L(j)} \mid Z_{K(j,j+1)} \, , X_{K(j,j+1)} = X^{\ell}_{K(j,j+1)} \,]$$



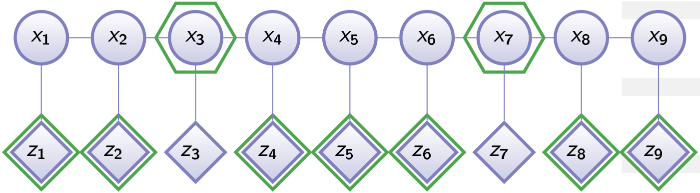
Algorithm:

Step 3:

Second partition is active.

Node $K(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{L(j-1,j)}$ and assuming that $x_{L(j-1)}$ and $x_{L(j)}$ were exactly $X^{\ell}_{L(j-1)}$ and $X^{\ell}_{L(j)}$.

$$X^{\ell+2}_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)} \,,\, X_{L(j-1,j)} = X^{\ell}_{L(j-1,j)} \,]$$



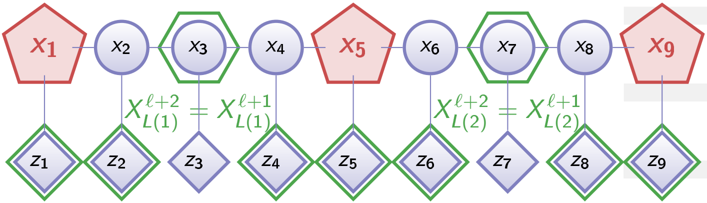
Algorithm:

Step 3:

Second partition is active.

Node $K(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{L(j-1,j)}$ and assuming that $x_{L(j-1)}$ and $x_{L(j)}$ were exactly $X^{\ell}_{L(j-1)}$ and $X^{\ell}_{L(j)}$.

$$X^{\ell+2}_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)}\ ,\ X_{L(j-1,j)} = X^{\ell}_{L(j-1,j)}\ ]$$



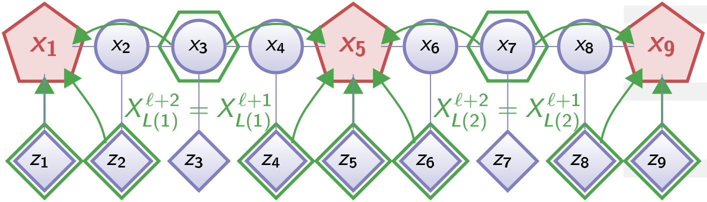
Algorithm:

Step 3:

Second partition is active.

Node $K(j)$ (middle node) estimate of its state based only measurements collected in its group $Z_{L(j-1,j)}$ and assuming that $x_{L(j-1)}$ and $x_{L(j)}$ were exactly $X^{\ell}_{L(j-1)}$ and $X^{\ell}_{L(j)}$.

$$X^{\ell+2}_{K(j)} = \mathbb{E}[x_{K(j)} \mid Z_{L(j-1,j)} \, , X_{L(j-1,j)} = X^{\ell}_{L(j-1,j)} \,]$$



Algorithm:

Step 4:

if there has been a significant improvement in the estimate

$$|x_{K(j)}^{\ell+2} - x_{K(j)}^{\ell}| \geq \varepsilon$$

then set $\ell = \ell + 2$ and go to step 2.
else the optimal estimate is achieved $X_{K(j)}^{\ell+2} = \mathbb{E}[x_{K(j)} \mid Z]$.
Move to Step 5.

Step 5:

For any node i of the j-th group(Markov Property).

$$\mathbb{E}[x_i \mid Z] = \mathbb{E}[x_i \mid Z_{K(j,j+1)}, X_{K(j,j+1)}].$$

Compute all the other estimates of the j-th group as

$$\mathbb{E}[x_i \mid Z_{K(j,j+1)}\ , X_{K(j,j+1)} = X_{K(j,j+1)}^{\ell+2}].$$

Algorithm:

Step 4:

if there has been a significant improvement in the estimate

$$|x_{K(j)}^{\ell+2} - x_{K(j)}^{\ell}| \geq \varepsilon$$

then set $\ell = \ell + 2$ and go to step 2.
else the optimal estimate is achieved $X_{K(j)}^{\ell+2} = \mathbb{E}[x_{K(j)} \mid Z]$.
Move to Step 5.

Step 5:

For any node i of the j-th group(Markov Property).

$$\mathbb{E}[\, x_i \mid Z] = \mathbb{E}[x_i \mid Z_{K(j,j+1)}, X_{K(j,j+1)}].$$

Compute all the other estimates of the j-th group as

$$\mathbb{E}[x_i \mid Z_{K(j,j+1)}\ , X_{K(j,j+1)} = X_{K(j,j+1)}^{\ell+2}].$$

Convergence Results:

$$\delta_K^\ell = X_K^\ell - \mathbb{E}[X_K | Z]$$

Theorem:

1. the error dynamics of the distributed smoothing algorithm at the nodes $\{K(j)\}$ are regulated by the equation
$$\delta_K^{\ell+2} = R\delta_K^\ell$$
2. R is asymptotically stable, i.e. all its eigenvalues are inside the open complex unit circle

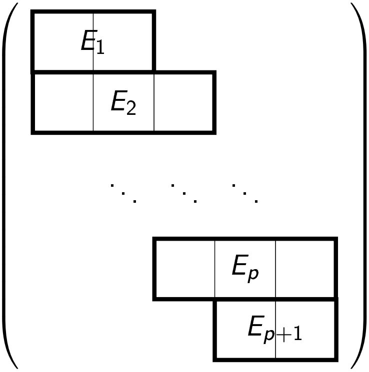
Convergence Results:

$$\Xi_j = \mathbb{V}\left(x_{K(j)}, X_{L(j-1,j)} \mid Z_{L(j-1,j)}\right) \mathbb{V}\left(X_{L(j-1,j)} \mid Z_{L(j-1,j)}\right)^{-1}$$

$$\Pi_j = \mathbb{V}\left(x_{L(j)}, X_{K(j,j+1)} \mid Z_{K(j,j+1)}\right) \mathbb{V}\left(X_{K(j,j+1)} \mid Z_{K(j,j+1)}\right)^{-1}$$

$$E_j = \Xi_j \begin{pmatrix} \Pi_j & 0_{n\times n} \\ 0_{n\times n} & \Pi_{i+1} \end{pmatrix} \qquad E_1 = \Xi_1 \Pi_1 \qquad E_{p+1} = \Xi_{p+1}\Pi_p,$$

$R =$



Specialization to the random walk case:

Scalar Random Walk

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ z_k &= x_k + v_k \end{aligned} \qquad k = 1, \dots N$$

w_k and v_k mutually independet white Gaussian noises

$$\mathbb{V} w_k = \lambda \qquad \mathbb{V} v_k = \sigma$$

Theorem

All the eigenvalues of R have the same asymptotic behavior,

$$\lambda(R(J)) \sim \text{const} \cdot \nu^{-J}, \qquad \text{as } J \to +\infty$$

where

$$\nu = 1 + \frac{\lambda}{2\sigma} + \frac{1}{2}\sqrt{\frac{4\lambda}{\sigma} + \frac{\lambda^2}{\sigma^2}} \geq 1$$

Summing Up

1. Sensor Networks and Consensus
2. An application: Localization and Tracking
3. Distributed Sensors Calibration
4. Analysis of a Randomized Kalman Filter
5. Distributed Kalman Smoother

Publications

Journal papers:

Appeared — **Consensus-based distributed sensor calibration and least-square parameter identification in WSNs.**
Saverio Bolognani,Simone Del Favero, Luca Schenato, Damiano Varagnolo. *International Journal of Robust and Nonlinear Control*

Submetted — **A majorization inequality and its application to distributed Kalman filtering.**
S. Del Favero. and S. Zampieri. *Submetted to Automatica*

Submetted — **Unconstrained and inequality constrained distributed kalman smoother.**
G. Pillonetto, S. Del Favero and B. Bell *Submetted to Automatica*

Conference papers:

Accepted — **Distributed Inequality Constrained Kalman Smoother.**
S. Del Favero, G. Pillonetto and B. Bell. 19th MTNS, Budapest, Hungary, 5-9 July 2010

Appeared — **Distributed estimation through randomized gossip Kalman filter.**
S. Del Favero and S. Zampieri. *Combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai 2009 December 16-18.

Appeared — **A distributed solution to estimation problems in wireless sensor networks leveraging broadcast communication..**
Simone Del Favero, Federico Librino, Michele Zorzi, Francesco Zorzi and Albert Harris III. *WONS 2009, The Sixth International Conference on Wireless On-demand Network Systems and Services, February 2-4, 2009. Snowbird, Utah, USA*

Appeared — **Distributed Sensor Calibration and Least-Square Parameter Identification in WSNs Using Consensus Algorithms.**
Saverio Bolognani, Simone Del Favero, Luca Schenato , Damiano Varagnolo. *Forty-Sixth Annual Allerton Conference on Communication, Control, and Computing September 23 - September 26, 2008 Allerton Retreat Center, Monticello, Illinois*