Distributed local estimation in interconnected systems with application to localization

**Ruggero Carli**, A. Carron, L. Schenato, M. Todescato

*Department of Information Engineering, University of Padova*

Strasbourg, 24 June 2014
Robotic networks

What kind of systems?
Group of systems with *control, sensing, communication* and *computing*

Individual members
- **sense** its immediate environment
- **communicate** with others
- **process** the information gathered
- **take a local action** in response
Robotic networks

Embedded robotic systems and sensor networks for

- high-stress, rapid deployment – e.g., disaster recovery networks
- distributed environmental monitoring – e.g., portable chemical and biological sensor arrays detecting toxic pollutants
- autonomous sampling for biological applications – e.g., monitoring of species in risk, validation of climate and oceanographic models
- science imaging – e.g., multispaceship distributed interferometers flying in formation to enable imaging at microarcsecond resolution
Research challenges

How to coordinate individual agents into coherent whole?

**Objective:** systematic methodologies to design and analyze cooperative strategies to control multi-agent systems

**Coordination tasks:** exploration, map building, search and rescue, surveillance, distributed sensing, monitoring

**Network localization:** preliminary and fundamental problem

**Optimization problem**
Research challenges

**Localization**: to reconstruct / estimate the network structure

- **Relative distance measurements**
- **GPS measurements**
- **bearing measurements**

**TODAY**

- **Scalar positions**
- **Only relative distance measurements**
THE PROBLEM: To estimate a set of $N$ scalar variables $(x_1, \ldots, x_N)$
THE PROBLEM: To estimate a set of $N$ scalar variables $(x_1, \ldots, x_N)$

RELATIVE NOISY MEASUREMENTS

$z_{ij} = x_i - x_j + n_{ij}, \quad (i, j) \in \{1, \ldots, N\}$

$n_{ij}$: zero mean gaussian error

What kind of information?
The problem setup

THE PROBLEM: To estimate a set of $N$ scalar variables $(x_1, \ldots, x_N)$

What kind of information?

RELATIVE NOISY MEASUREMENTS

$z_{ij} = x_i - x_j + n_{ij}, \quad (i, j) \in \{1, \ldots, N\}$

$n_{ij}$: zero mean gaussian error

We assume the available measurements are $M > N$
The problem setup

**RELATIVE NOISY MEASUREMENTS**

\[ z_{ij} = x_i - x_j + n_{ij}, \quad (i,j) \in \{1, \ldots, N\} \]

\[ n_{ij} : \text{zero mean gaussian error} \]
The problem setup

**RELATIVE NOISY MEASUREMENTS**

\[ z_{ij} = x_i - x_j + n_{ij}, \quad (i, j) \in \{1, \ldots, N\} \]

\( n_{ij} \): zero mean gaussian error

**MEASUREMENT GRAPH**

\[ G = (V, E) \]
The problem setup

**RELATIVE NOISY MEASUREMENTS**

\[ z_{ij} = x_i - x_j + n_{ij}, \quad (i, j) \in \{1, \ldots, N\} \]

\[ n_{ij} : \text{zero mean gaussian error} \]

**MEASUREMENT GRAPH**

\[ G = (V, E) \]

\[ V = \{1, \ldots, N\} \]
The problem setup

RELATIVE NOISY MEASUREMENTS

\[ z_{ij} = x_i - x_j + n_{ij}, \quad (i, j) \in \{1, \ldots, N\} \]

\( n_{ij} : \) zero mean gaussian error

MEASUREMENT GRAPH

\[ G = (V, E) \]

\[ V = \{1, \ldots, N\} \]

\[ (i, j) \in E \iff \text{there exists } z_{ij} \]
The problem: least-squares approach

\[ \mathbb{E}[n_{ij}] = 0 \quad R_{ij} = \mathbb{E}[n_{ij}^2] \]

Functional cost: least-squares approach

\[ F = \frac{1}{2} \sum_{(i,j) \in E} (x_i - x_j - z_{ij})^2 / R_{ij} \]
The problem: least-squares approach

\[ \mathbb{E}[n_{ij}] = 0 \quad R_{ij} = \mathbb{E}[n_{ij}^2] \]

Functional cost: least-squares approach

\[ F = \frac{1}{2} \sum_{(i,j) \in E} \frac{(x_i - x_j - z_{ij})^2}{R_{ij}} \]

Goal

To find a N-upla \((x_1, \cdots, x_N)\) minimizing the functional cost

\[ F = \frac{1}{2} \sum_{(i,j) \in E} \frac{(x_i - x_j - z_{ij})^2}{R_{ij}} \]
The problem: least-squares approach

\[
x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N \\
\]

\[
n, z \in \mathbb{R}^M \\
\]

\[
n = \begin{bmatrix} (n_{ij})_{(ij) \in E} \end{bmatrix} \\
\]

\[
z = \begin{bmatrix} (z_{ij})_{(ij) \in E} \end{bmatrix} \\
\]
The problem: least-squares approach

\[ x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N \]

\[ n, z \in \mathbb{R}^M \]

\[ n = \begin{bmatrix} (n_{ij})_{(ij)\in E} \end{bmatrix} \]

\[ z = \begin{bmatrix} (z_{ij})_{(ij)\in E} \end{bmatrix} \]

Functional cost: least-squares approach

\[ F(x, z) = \frac{1}{2} (z - Ax)^T R^{-1} (z - Ax) \]

\[ A : incidence \ matrix \ of \ the \ graph \ G \]

\[ R = \text{diag}\{R_{ij}\} \]
The problem: least-squares approach

\[
x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N \\
\]

\[
n, z \in \mathbb{R}^M \\
n = \begin{bmatrix} (n_{ij})_{(ij)\in E} \end{bmatrix} \\
z = \begin{bmatrix} (z_{ij})_{(ij)\in E} \end{bmatrix}
\]

Functional cost: least-squares approach

\[
F(x, z) = \frac{1}{2} (z - Ax)^T R^{-1} (z - Ax)
\]

\[A: \text{incidence matrix of the graph } G\]

\[R = \text{diag}\{R_{ij}\}\]

Optimal solution?

\[
x_{opt} = (A^T R^{-1} A)^\dagger A^T R^{-1} z
\]
The problem: least-squares approach

\[ x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N \quad n, z \in \mathbb{R}^M \]

\[ n = \begin{bmatrix} (n_{ij})_{(ij) \in E} \end{bmatrix} \]

\[ z = \begin{bmatrix} (z_{ij})_{(ij) \in E} \end{bmatrix} \]

Functional cost: least-squares approach

\[ F(x, z) = \frac{1}{2} (z - Ax)^T R^{-1} (z - Ax) \]

\[ A : \text{incidence matrix of the graph } G \]

\[ R = \text{diag}\{R_{ij}\} \]

Optimal solution?

\[ x_{opt}^* = (A^T R^{-1} A)^\dagger A^T R^{-1} z \]

\[ x_{opt} = (A^T R^{-1} A)^\dagger A^T R^{-1} z + \alpha \mathbf{1}, \quad \alpha \in \mathbb{R} \]
The problem: least-squares approach

\[ x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N \quad n, z \in \mathbb{R}^M \]
\[ n = \begin{bmatrix} (n_{ij})_{(ij) \in E} \end{bmatrix} \]
\[ z = \begin{bmatrix} (z_{ij})_{(ij) \in E} \end{bmatrix} \]

**Functional cost: least-squares approach**

\[ F(x, z) = \frac{1}{2} (z - Ax)^T R^{-1} (z - Ax) \]

\( A : \text{incidence matrix of the graph } G \)

\( R = \text{diag}\{R_{ij}\} \)

**Optimal solution?**

\[ x^*_\text{opt} = (A^T R^{-1} A)^\dagger A^T R^{-1} z \]
\[ x_{\text{opt}} = (A^T R^{-1} A)^\dagger A^T R^{-1} z + \alpha \mathbf{1}, \quad \alpha \in \mathbb{R} \]
No centralized solutions!

\[ z = \left[ (z_{ij})_{(i,j) \in E} \right] \]

\[ A, R \]

\[ x_{opt}^* = (A^T R^{-1} A)^\dagger A^T R^{-1} z \]
Distributed and Asynchronous

- **Distributed** as opposed to **centralized**

Each node has computational, communication, memory capabilities

Local knowledge of the topology
Distributed and Asynchronous

- **Distributed** as opposed to **centralized**

  Each node has computational, communication, memory capabilities

  Local knowledge of the topology

- **Asynchronous** as opposed to **synchronous**

Ruggero Carli
Robust distributed localization
Distributed and Asynchronous

- **Distributed** as opposed to centralized

  Each node has computational, communication, memory capabilities

  Local knowledge of the topology

- **Asynchronous** as opposed to synchronous

Ruggero Carli

Robust distributed localization
Agents’ Assumptions

\[ \text{Communication graph} \quad = \quad \text{Measurement graph} \]

\[ N_i \] denotes the set of neighbors of node \( i \) in the graph \( G \)

\[ N_i = \{ j \in V \quad \text{s. t.} \quad (i, j) \in E \} \]
Agents’ Assumptions

Communication graph = Measurement graph

$N_i$ denotes the set of neighbors of node $i$ in the graph $G$

$N_i = \{ j \in V \text{ s. t. } (i,j) \in E \}$

Node $i$ keeps in memory

- $\hat{x}_i$ estimate of $x_i$
Agents’ Assumptions

Communication graph $= \text{Measurement graph}$

$N_i$ denotes the set of neighbors of node $i$ in the graph $G$

$$N_i = \{ j \in V \; \text{s. t.} \; (i, j) \in E \}$$

Node $i$ keeps in memory

- $\hat{x}_i$ estimate of $x_i$
- $\hat{x}^{(i)}_j$ estimate of $x_j$, for all $j \in N$
Agents’ Assumptions

Communication graph = Measurement graph

\( N_i \) denotes the set of neighbors of node \( i \) in the graph \( G \)

\[ N_i = \{ j \in V \text{ s. t. } (i, j) \in E \} \]

Node \( i \) keeps in memory

- \( \hat{x}_i \) estimate of \( x_i \)
- \( \hat{x}^{(i)}_j \) estimate of \( x_j \), for all \( j \in N \)
- \( z_{ij} \), for all \( j \in N \)
Gradient descent strategy

Gradient-based approach

\[ F(x, z) = \frac{1}{2} \sum_{j \in N_i} \frac{(x_i - x_j - z_{ij})^2}{R_{ij}} + \text{terms without } x_i \]
Gradient descent strategy

Gradient-based approach

\[
F(x, z) = \frac{1}{2} \sum_{j \in N_i} \left( x_i - x_j - z_{ij} \right)^2 / R_{ij} + \text{terms without } x_i
\]

\[
F_i \left( x_i, \{x_j, z_{ij}\}_{j \in N_i} \right)
\]
Gradient descent strategy

Gradient-based approach

\[ F(x, z) = \frac{1}{2} \sum_{j \in N_i} (x_i - x_j - z_{ij})^2 / R_{ij} \] + terms without \( x_i \)

\[ F_i \left( x_i, \{ x_j, z_{ij} \}_{j \in N_i} \right) \rightarrow F_i \left( \hat{x}_i, \{ \hat{x}_j^{(i)}, z_{ij} \}_{j \in N_i} \right) \]

Ruggero Carli

Robust distributed localization
Gradient descent strategy

Gradient-based approach

\[ F(x, z) = \frac{1}{2} \sum_{j \in N_i} \left( x_i - x_j - z_{ij} \right)^2 / R_{ij} \]

\[ + \] terms without \( x_i \)

\[ F_i \left( x_i, \{x_j, z_{ij}\}_{j \in N_i} \right) \rightarrow F_i \left( \hat{x}_i, \{\hat{x}^{(i)}_j, z_{ij}\}_{j \in N_i} \right) \]

Agent \( i \) will update \( \hat{x}_i \) moving along the gradient of this function
Asynchronous broadcast

At each iteration there is only one node transmitting information

$t_1, t_2, t_3 ..., : time\ instants\ when\ the\ algorithm\ is\ performed$
The algorithm: three steps

Assume node $i$ is activated at time instant $t_n$

$t_1$ $t_2$ $t_3$ $t_n$ $t_{n+1}$
The algorithm: three steps

Assume node $i$ is activated at time instant $t_h$

$t_1$ $t_2$ $t_3$ $t_h$ $t_{h+1}$

- **First step**: node $i$ updates its estimate $\hat{x}_i$ as

$$\hat{x}_i \leftarrow \hat{x}_i - \alpha_i \sum_{j \in N_i} \frac{\hat{x}_i - \hat{x}_j^{(i)} - z_{ij}}{R_{ij}}$$

Ruggero Carli
Robust distributed localization
The algorithm: three steps

Assume node $i$ is activated at time instant $t_h$

$t_1 \quad t_2 \quad t_3 \quad \quad t_h \quad \quad t_{h+1}$

- **First step**: node $i$ updates its estimate $\hat{x}_i$ as

$$\hat{x}_i \leftarrow \hat{x}_i - \alpha_i \sum_{j \in N_i} \frac{\hat{x}_i - \hat{x}_j^{(i)} - z_{ij}}{R_{ij}}$$

Gradient of...

$$F_i = \sum_{j \in N_i} \frac{(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij})^2}{R_{ij}}$$
The algorithm: three steps

Assume node $i$ is activated at time instant $t_h$

$\begin{align*}
\quad & t_1 & t_2 & t_3 & t_h & t_{h+1} \\
\end{align*}$

- **First step**: node $i$ updates its estimate $\hat{x}_i$ as

$$\hat{x}_i \leftarrow \hat{x}_i - \alpha_i \sum_{j \in N_i} \frac{\hat{x}_i - \hat{x}_j^{(i)} - z_{ij}}{R_{ij}}$$

$\alpha_i$: *updating step size*

$F_i = \sum_{j \in N_i} \frac{(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij})^2}{R_{ij}}$

Ruggero Carli
Robust distributed localization
The algorithm: three steps

- **Second step**: node $i$ transmits the updated value $\hat{x}_i$ to all its neighbors.
The algorithm: three steps

- **Second step**: node \( i \) transmits the updated value \( \hat{x}_i \) to all its neighbors.

- **Third step**: node \( j, j \in N_i \), updates \( \hat{x}_i^{(j)} \) as
  
  \[
  \hat{x}_i^{(j)} \leftarrow \hat{x}_i
  \]
Definitions: how frequently a node communicates?

Randomly persistent communicating network

There exists N-upla $\beta_1, ..., \beta_N$ with $\beta_i > 0$, $\sum \beta_i = 1$, such that

$$\mathbb{P}[\text{node } i \text{ is the node performing the } h-th \text{ iteration}] = \beta_i$$
Definitions: how frequently a node communicates?

Randomly persistent communicating network

There exists N-upla $\beta_1, ..., \beta_N$ with $\beta_i > 0$, $\sum \beta_i = 1$, such that

$$\mathbb{P}[\text{node } i \text{ is the node performing the } h-th \text{ iteration}] = \beta_i$$

Uniformly persistent communicating network

There exists positive integer number $\tau$ such that

$$\forall h, \text{ each node performs at least one iteration within the interval } (h, h + \tau)$$
Results

Let \( \hat{x}(t) = [\hat{x}_1(t), \ldots, \hat{x}_N(t)]^T \)

Theorem (Uniformly persistent communicating network)

Assume the weights \( \alpha_i \) satisfy

\[
0 < \alpha_i \leq \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}
\]

- the evolution \( h \to \hat{x}(h) \) asymptotically converges to an optimal solution
- the evolution \( h \to \hat{x}(h) \) is exponentially convergent
Results

Let \( \hat{x}(t) = [\hat{x}_1(t), ..., \hat{x}_N(t)]^T \)

Theorem (Randomly persistent communicating network)

Assume the weights \( \alpha_i \) satisfy

\[
0 < \alpha_i \leq \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}
\]

- the evolution \( h \rightarrow \hat{x}(h) \) converges almost surely to an optimal solution
- the evolution \( h \rightarrow \hat{x}(h) \) is exponentially convergent in mean – square sense
Remarks

- If \( \alpha_i = \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1} \) then

\[
\hat{x}_i \leftarrow \text{argmin} \frac{1}{2} \sum_{j \in N_i} \left( \hat{x}_i - \hat{x}^{(i)}_j - z_{ij} \right)^2 \frac{1}{R_{ij}}
\]

At each iteration the optimal gradient step is performed!
At each iteration the optimal gradient step is performed!

\[
\hat{x}_i \leftarrow \text{argmin} \frac{1}{2} \sum_{j \in N_i} \left( \hat{x}_i - \hat{x}_j^{(i)} - z_{ij} \right)^2 \frac{R_{ij}}{R_{ij}}
\]

\[
\hat{x}_i \leftarrow p_{ii} \hat{x}_i + \sum_{j \in N_i} p_{ij} \hat{x}_j^{(i)} + \alpha_i \sum_{j \in N_i} \frac{z_{ij}}{R_{ij}}
\]
Remarks

- If $\alpha_i = \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$ then

$$\hat{x}_i \leftarrow \text{argmin} \frac{1}{2} \sum_{j \in N_i} \frac{(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij})^2}{R_{ij}}$$

At each iteration the optimal gradient step is performed!

- $\hat{x}_i \leftarrow p_{ii} \hat{x}_i + \sum_{j \in N_i} p_{ij} \hat{x}_j^{(i)} + \alpha_i \sum_{j \in N_i} \frac{z_{ij}}{R_{ij}} \quad \sum p_{ij} = 1$
Remarks

- If $\alpha_i = \left(\sum_{j \in N_i} \frac{1}{R_{ij}}\right)^{-1}$ then

$$\hat{x}_i \leftarrow \arg\min \frac{1}{2} \sum_{j \in N_i} \frac{(\hat{x}_i - \hat{x}_j^{(i)} - z_{ij})^2}{R_{ij}}$$

At each iteration the optimal gradient step is performed!

- $\hat{x}_i \leftarrow p_{ii} \hat{x}_i + \sum_{j \in N_i} p_{ij} \hat{x}_j^{(i)} + \alpha_i \sum_{j \in N_i} \frac{z_{ij}}{R_{ij}}$ \hspace{1cm} $\sum p_{ij} = 1$

Consensus step
Robustness….

How robust is the algorithm?

- Unreliable communications (packet losses)
- Communication delays
Assumptions: bounded packet losses

Assumption (bounded packet losses)

There exists a positive integer \( L \) such that

\[ \text{the number of consecutive communication failures between neighboring nodes is smaller than } L \]

Loosely speaking there can be no more than \( L \) consecutive packet losses between any pair of nodes \( i, j \)
Assumptions: bounded delay

Assumption (bounded delay)

Assume that
- node $i$ transmits its information at $h$-th iteration;
- communication link $(i,j)$ does not fail;

Then node $j$ receives $\hat{x}_i$ not later than iteration $h+D$. 

Ruggero Carli  
Robust distributed localization
Results

Theorem (Uniformly persistent communicating network + bounded packet losses + bounded delays)

Assume the weights $\alpha_i$ satisfy

$$0 < \alpha_i < \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$$

- the evolution $h \to \hat{x}(h)$ asymptotically converges to an optimal solution
- the evolution $h \to \hat{x}(h)$ is exponentially convergent
Results

Theorem (Uniformly persistent communicating network + bounded packet losses + bounded delays)

Assume the weights $\alpha_i$ satisfy

$$0 < \alpha_i < \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$$

- the evolution $h \rightarrow \hat{x}(h)$ asymptotically converges to an optimal solution
- the evolution $h \rightarrow \hat{x}(h)$ is exponentially convergent

If $\alpha_i = \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$ for some $i \Rightarrow$ COUNTEREXAMPLE of no Convergence
Theorem (Uniformly persistent communicating network + bounded packet losses + bounded delays)

Assume the weights $\alpha_i$ satisfy

$$0 < \alpha_i < \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$$

- the evolution $h \rightarrow \hat{x}(h)$ asymptotically converges to an optimal solution
- the evolution $h \rightarrow \hat{x}(h)$ is exponentially convergent

If $\alpha_i = \left( \sum_{j \in N_i} \frac{1}{R_{ij}} \right)^{-1}$ for some $i$ \Rightarrow COUNTEREXAMPLE of no Convergence
Numerical results

\[ J = \log(\| \hat{x}(k) - x^* \|) \]
Numerical results

\[ J = \log(||\hat{x}(k) - x^*||) \]
Research challenges

Localization: to reconstruct / estimate the network structure

- Relative distance measurements
- GPS measurements
- Bearing measurements

Nonlinear functional cost
## Conclusions

### Summary

1. distributed and asynchronous localization algorithm
2. convergence to optimal solution
3. robustness to bounded delays
4. robustness to bounded packet losses

### On going work

1. 3D scenario
2. nonlinear functional cost
3. similar ideas to **state estimation** in smart grid
4. rate of convergence
The space of partitions

Definition (space of $N$-partitions)

$V_N^{(L)}$ is collections of $N$ subsets of $Q$ \( v = \{v_1, \ldots, v_N\} \) s. t.

1. \( v_i \neq \emptyset \) and \( v_1 \cup \cdots \cup v_N = Q \)
2. \( \text{interior}(v_i) \cap \text{interior}(v_j) \neq \emptyset \) if \( i \neq j \), and
3. \( \exists L \) such that \( v_i = \bigcup_{i=1}^{L} S_i \), \( S_i \) CONVEX and CLOSED
The space of partitions

**Definition (space of $N$-partitions)**

$V_N^{(L)}$ is collections of $N$ subsets of $Q$  
$v = \{v_1, \ldots, v_N\}$ s. t. 
1. $v_i \neq \emptyset$ and $v_1 \cup \cdots \cup v_N = Q$
2. interior($v_i$) $\cap$ interior($v_j$) $\neq \emptyset$ if $i \neq j$, and
3. $\exists L$ such that $v_i = \bigcup_{i=1}^{L} S_i$, $S_i$ CONVEX and CLOSED

- How can we measure distances in the space of partitions?
- Meaning of convergence?