Hybrid Network Localization Using Distributed Contractors

O. Reynet

IT Department/Ocean Sensing and Mapping Team - ENSTA Bretagne - France

June 14, 2011
1. Distributed Localization Context

2. Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3. Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4. Simulations and Results
   - Demonstration
   - Results

5. Conclusion
Outline

1. Distributed Localization Context

2. Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3. Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4. Simulations and Results
   - Demonstration
   - Results

5. Conclusion
Outline

1. Distributed Localization Context

2. Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3. Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4. Simulations and Results
   - Demonstration
   - Results

5. Conclusion
Outline

1. Distributed Localization Context

2. Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3. Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4. Simulations and Results
   - Demonstration
   - Results

5. Conclusion
1. Distributed Localization Context

2. Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3. Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4. Simulations and Results
   - Demonstration
   - Results

5. Conclusion
1 Distributed Localization Context

2 Contractor-based Localization Algorithm
   - Centralized approach
   - Distributed approach

3 Localization Using Contractors Combining
   - Natural Hybrid Scheme
   - Range Contractor
   - Angle Contractor

4 Simulations and Results
   - Demonstration
   - Results

5 Conclusion
# Distributed Localization Context

Localization of a networked swarm

## Nature of the Swarm
- Mobile phones,
- Sensors networks,
- Robots,
- A multi-agent system.

## Constraints
- Distance between agents,
- Angle between agents,
- Direction: ”b is to the north” . . .
- Mobility models.

## Each agent $A_i$ of the network is able to:
- measure some neighbours: angle, range, doa, tdoa . . .
- build contractors from these constraints,
- compute contractors $C$ to estimate its own location,
- exchange its localization $[p_i]$ with its neighbours.
Distributed Localization Context
A multi-agent system with anchors

- Agents are represented by circles,
- Anchors by black crosses,
- Sensing-range $r$ is sketched by a dashed circle,
- The connectivity is the number of edges which end at the vertex $\mathcal{A}_i$, the number of neighbours.
Figure: A contractor graph $\mathcal{G} = (\mathcal{A}, \mathcal{E})$: each edge stands for a contractor $C^k_i$. Each agent $A_i$ is able to compute the contractors $C^k_i$. 
Distributed Localization Context

Goals

- Fully distributed algorithm,
- Efficiency.

We set some rules:

- no special anchor placement or gridding: only random configurations,
- no hypothesis on the noise, except it is bounded,
- no outliers,
- no bisections occur during the algorithm (to speed up the computation).
Definition

A contractor is a monotonic mapping $C$ from $\mathbb{R}^n$ to $\mathbb{R}^n$ such that:

1. $\forall [\mathbf{x}] \in \mathbb{R}^n, C([\mathbf{x}]) \subseteq [\mathbf{x}]$ (contractance)
2. $\forall [\mathbf{x}] \subset [\mathbf{y}] \Rightarrow C([\mathbf{x}]) \subset C([\mathbf{y}])$ (monotony)

An omniscient supervisor of the network

- has access to the set of all contractors of the agents: $\mathcal{L} = \{C_1, C_2, \ldots, C_m\}$,
- Build the $C_\infty$ contractor as follow:

$$C_\infty = C_1 \circ C_2 \circ \cdots \circ C_m \circ C_1 \circ C_2 \circ \ldots$$
Contractor-based Centralized Algorithm

$\mathcal{C}_\infty$, monotonicity and fairness of the strategy

Theorem

$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}_\infty([\mathbf{x}])$ converges to the largest box $[\mathbf{z}] \subset [\mathbf{x}]$ such that $[\mathbf{z}]$ is a fixed point, i.e. $\mathcal{C}_\infty([\mathbf{z}]) = [\mathbf{z}]$.

Proof.

The demonstration of this theorem is based on the facts that:

- $\mathcal{C}_\infty$ is a contractor (contractance and monotonicity),
- $\mathcal{C}_\infty$ is following a fair strategy, i.e. the way the contractors are combined is fair.

Definition

A combining strategy is said to be fair if, for any $k \geq 1$ and any contractor $\mathcal{C}$ of $\mathcal{L}$, there exists $j \geq k$ such that $\mathcal{C}$ is computed at rank $j$. 
Contractor-based Centralized Algorithm

\(C_\infty\) solves the localization problem in a centralized way.

### \(C_\infty\) as computed by the supervisor

Let \([p] = [p_1] \times [p_2] \times \cdots \times [p_n]\) and \(\mathcal{L} = \{C_1, C_2, \ldots, C_m\}\)

1. Build \(C := C_1 \circ C_2 \circ \ldots C_m\) from \(\mathcal{L}\)
2. repeat
3. \([p] := C([p])\)
4. until a fixed point is reached

### How to distribute \(C_\infty\)'s computation to agents?

Contractors' composition \(\circ\) in \(C_\infty\) should be replaced by:

- network exchange of the domains of \([p_i]\),
- intersection of the domains of \([p_i]\).

The chaotic order in which contractors are applied has no impact on the convergence property nor on the fixed point, as long as the order is fair.
Contractor-based Distributed Algorithm

CDA as computed by agent $A_i$

1: Inputs $[p_i], N_i$
2: for all $A_k \in N_i$ do
3:   $[p_k] := ]-\infty, +\infty[$
4: end for
5: SendMsg$(N_i, [p_i])$
6: while $(A_k, [p_r]) \leftarrow$ getMsg() do
7:   if $r == i$ then
8:     $[p_i] := [p_i] \cap [p_r]$
9:   else if $r == k$ then
10:    $[p_k] := [p_k] \cap [p_r]$
11: end if
12: $( [p_i], [p_k] ) := C_i^k([p_i], [p_k])$
13: if $[p_k]$ has been contracted then
14:   sendMsg$(A_k, [p_k])$
15: end if
16: end while
17: if $[p_i]$ has been contracted then
18:   sendMsg$(N_i, [p_i])$
19: end if
Contractor-based Distributed Algorithm
Forward/backward propagation over the contractors’ graph

Propagation starting from $A_1$

Figure: Forward propagation: $A_1$ and $A_2$ domains are exchanged and contracted.
Contractor-based Distributed Algorithm
Forward/backward propagation

Propagation over the contractors’ graph

Figure: Forward propagation: $A_2$, $A_3$, $A_5$, $A_6$ and $A_7$ domains are exchanged and contracted.
Contractor-based Distributed Algorithm
Forward/backward propagation

Propagation over the contractors’ graph

**Figure:** Forward propagation: \( A_3, A_4, A_6 \) and \( A_7 \) domains are exchanged and contracted.
Contractor-based Distributed Algorithm
Forward/backward propagation

Propogation over the contractors’ graph

Figure: Backward propagation: first step
Contractor-based Distributed Algorithm
Forward/backward propagation

Propagation over the contractors’ graph

Figure: Backward propagation: second step
Contractor-based Distributed Algorithm
Forward/backward propagation

Propagation over the contractors’ graph

Mostly, only 5 forward/backward passes are necessary to reach a fixed point.

Figure: Backward propagation: third step
### Independent Measurement Methods

- Only domains are exchanged and intersected,
- Intersections and exchanges do not depend on the used contractor,
- Therefore, each agent may use its own measurements and its own contractors.

### Simulated Methods

- Range-only,
- Angle-only,
- Hybrid range-angle.
Distance Equation

\[ d_{ab} = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \]  \hspace{1cm} (4)

- \( d_{ab} \) is the noisy measured distance between agent \( a \) and agent \( b \),
- \((x_a, y_a)\) and \((x_b, y_b)\) are the positions of agents \( a \) and \( b \),
- In our simulation framework, all these data are bounded intervals.

C++ IBEX library is used to automatically build the range-only contractor from this equation.
Distributed Localization

Angle Contractor

Given a box and an angle, it returns:
- the smallest box inside the angle
- or the smallest angle including the box.

Remarks:
- Using tangent or sine functions leads to contractors with a poor efficiency,
- Moreover, we are looking for a minimal contractor $\Rightarrow$ we built it in IBEX!
Distributed Localization
Angle Contractor

Definition

A contractor $C$ is minimal if $\forall [x] \in \mathbb{R}^n, C([x]) = [y]$ and $[y]$ is a fixed point, i.e. $C([y]) = [y]$.

Implementation

- All cases are detailed...
- Much quicker,
- but not easy to code...
Simulations
Demonstration

Three simulations:
- Range-only,
- Angle-only,
- Hybrid Range-Angle.

Parameters:
- Anchor ratio,
- Connectivity,
- Number of agents.
Results Analysis

Localization error vs. anchor ratio (100 random simulations)

1000 agents, mean connectivity 10

O. Reynet — Hybrid Network Localization Using Distributed Contractors
Results Analysis

Localization error vs. connectivity (100 random simulations)

1000 agents, anchor ratio 10%
Results Analysis
Computing Time vs. Number of Agents (100 random simulations)

mean connectivity 10, anchor ratio 10%
Conclusion
A new horizon for distributed localization

Advantages
- Contractors are easy to implement using IBEX,
- No linearization, no approximation,
- No gridding or anchor placement,
- No bisections,
- Accurate and very fast.

Further work
- Dealing with outliers (q-relaxed intersection),

Thank you for your attention
Some questions?