Distributed Intelligent Systems – W13:
Distributed Sensing using Robotic Sensor Networks
Outline

• Motivation for robotic sensing assets

• Gas source localization as a sensing mission example

• Algorithms for gas source localization
  – Bio-inspired algorithms
  – Formation-based algorithms
Motivation
Spatially Dense Air Quality Monitoring

Air pollution in urban areas is a global concern
• affects quality of life and health
• urban population is increasing

Air pollution is highly location- and time-dependent
• traffic chokepoints and rush hours
• urban canyons and weather
• industrial installations and activities

Air pollution monitoring today
• Sparse, stationary and expensive stations
• Spatial interpolation with mesoscale models
Environmental Monitoring

Typical solution in environmental monitoring:
- sparse sensing
- expensive
- field estimation via models
- possible mobility

Distributed solution for augmentation:
- size, cost
- number
- networked
- mobile

Physical field:
- built or natural
- bounded or unbounded
- 2D or 3D
OpenSense Vision

**Measurement data**
Citizen-, consortium-, agency-operated sensors

**Explanatory Variables**
Land-use, meteorology, traffic

**Exposure information**
Personal recommendations, health studies, urban planning, crowdsensing

**High-resolution pollution maps**
Spatiotemporally flexible, modeling; emphasis on data-driven statistical modeling methods
Problems in Distributed Sensing

Possible missions:  Robotic solution for:
- monitoring  - fast deployability
- patrolling  - adaptive spatiality
- searching  - larger coverage
- mapping
- inspecting

Physical field:
- artificial or natural
- bounded or unbounded
- 2D or 3D
Gas Source Localization (GSL)
Possible Applications of GSL
Mobile Sensing Assets for GSL

Gas source

- e.g.: leaking gas pipe
- bomb / mine
- food

Wind flow
Mobile Sensing Assets for GSL

Gas source

Wind flow

e.g.:
leaking gas pipe
bomb / mine
food

Khepera III robot with chemical and anemometric sensing
Plumes: A Tricky Field

Steam Plume Visualization

Courtesy by L. Marques, simulated plume
Sub-Tasks for GSL

- Plume acquisition
- Plume tracking
- Source declaration
Algorithms for Plume Tracking

- Gradient-based Algorithms
  - Time-averaged plume model
  - + Computationally efficient, intuitive
  - - Need for several samples, slow

- Bio-inspired Algorithms
  - Reactive
  - Finite State Machine
  - + Computationally efficient, easy to implement
  - - Not reliable in realistic environments

- Metaheuristic Algorithms
  - Approximation and optimization methods
  - + Robust controller
  - - Learning process
Algorithms for Plume Tracking

- Formation-based Algorithms
  - Natively designed for multi-robot systems
  - Reactive
  - + Computationally efficient
  - - Requires inter-robot localization
  - - Possibly fragile in realistic environments

- Probabilistic Algorithms
  - Probability distribution of the source location
  - + Flexible to the type of sensing system
  - + Rich set of information as output
  - + Level of uncertainty
  - - Require a plume model
  - - Computationally expensive
Single Robot Algorithms
Systematic experiments in the wind tunnel

Arena: 18 x 4 m
Wind speed: 1 m/s (~laminar)
Chemical and Anemometric Sensing

• Khepera III robot
  – 13 cm diameter
  – Differential-drive
  – 400 MHz ARM CPU
  – No floating point unit (FPU)

• Chemical sensing board
  – VOC sensor, ppm level
  – Open sampling system
  – Actively sniffing (micro-pump)

• Wind direction sensor board
  – Thermistor-based array
  – Accuracy < 10 degrees
Bio-Inspired Algorithms

State machines, reactive

Combination of surging, casting, spiraling

Binary odor concentration (in plume, not in plume)

Wind direction
Surge-Spiral: Algorithm

Wind flow

Wind direction measurement

\( d_{\text{gap}} \) \( d_{\text{lost}} \)
Surge-Spiral: Real Robot Implementation
Surge-Spiral: Sample Trajectory

Upwind in the plume, spiraling for reacquisition
From Single Robot to Multi-Robot Algorithms
Distance overhead = $D_{sf}/D_{min}$ normalized for single robot
With Cooperation
unpublished
With Cooperation
unpublished

? Improve? ? Redesign?
Improve?

Crosswind formation

Redesigned solution!

[Lochmatter et al, DARS 2010]
## Performance Comparison

<table>
<thead>
<tr>
<th></th>
<th>bio-inspired</th>
<th>probabilistic</th>
<th>formations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distance overhead</strong></td>
<td>Casting</td>
<td>Surge-spiral</td>
<td>Surge-cast</td>
</tr>
<tr>
<td><strong>Success rate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Single-robot</strong></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Multi-robot</strong></td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>state machine (bio. behavior)</td>
<td>discrete decisions (Bayes inference)</td>
<td>smooth adaptation (control theory)</td>
</tr>
<tr>
<td><strong>CPU requirements (MIPS)</strong></td>
<td>0.001</td>
<td>10'000</td>
<td>300</td>
</tr>
<tr>
<td><strong>Memory requirements</strong></td>
<td>few bytes</td>
<td>~ GB</td>
<td>&lt; 50 kB</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td>straightforward</td>
<td>hard</td>
<td>medium</td>
</tr>
<tr>
<td><strong>Plume model</strong></td>
<td>implicit</td>
<td>explicit</td>
<td>implicit</td>
</tr>
</tbody>
</table>

Laplacian Formation-Based Distributed Gas Source Localization
Approach and Robotic Node

- Gas concentration
- Wind direction
- Range and bearing
- Radio communication

No global positioning
No external coordination
Behavioral Approach

Formation control
Stay in formation

Upwind movement
Go towards the source

Plume centering
Stay in the plume
Control Algorithm

\[
\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad \text{Velocity vector in robot frame}
\]

\[ k_v, \ k_\omega \quad \text{constant gains} \]

\[ v = k_v \mathbf{u}_x \quad \text{Translational speed} \quad 0 \leq v \leq v_{max} \]

\[ \omega = k_\omega \mathbf{u}_y \quad \text{Rotational speed} \quad -\omega_{max} \leq \omega \leq \omega_{max} \]

\[
\mathbf{u} = k_w \mathbf{u}_w + k_c \mathbf{u}_c + k_f \mathbf{u}_f
\]

\[ \mathbf{u}_w = \text{upwind movement} \]

\[ \mathbf{u}_c = \text{plume centering} \]

\[ \mathbf{u}_f = \text{formation control} \]

\[ k_w = k_c = k_f = 1 \quad \text{weights} \]
Control Algorithm

Formation control with plume-based bias ($u_f$)

$$
\mathbf{u} = \left[ \sum_{j=0}^{N} \mathcal{L}_j \left( x_j - \beta_j^x \right) \right] - \left[ \sum_{j=0}^{N} \mathcal{L}_j \left( y_j - \beta_j^y \right) \right] + R(\theta) \left[ \begin{array}{c} u_{c_{\text{max}}} \\ 1 + e^{-\left(c_l-c_r\right)/k_l} \end{array} \right]
$$

$$
\beta_j = R(\theta) \begin{bmatrix} s_{uw} & 0 \\ 0 & s_{cw} \end{bmatrix} \left[ \mathbf{p}_j - \mathbf{p}_i \right]
$$

$$
s_{cw} = k_{cw} \frac{c_l + c_r}{1 + c_c}
$$

$\Theta$ = estimated wind direction (KF of wheel encoders and wind sensor board)

$R(\Theta)$ = rotational matrix (transformation from upwind-crosswind wind frame to x-y robot frame)

$s_{uw}$ = scaling factor upwind direction; $s_{cw}$ = scaling factor crosswind direction

$c_c$ = center concentration; $c_l$ = left concentration; $c_r$ = right concentration

$k_x$ = parameters (proportional gains)
High-Fidelity Simulation

Formation control
Stay in formation

Upwind movement
Move towards the source (while formation scaling)

Plume centering
Stay in the plume

Webots + plume plug-in (filament model, [Farrell, 2002])
Trajectories in Simulation
Algorithm Modifications for Wind Tunnel Experiments

#1: SCALING FACTOR

\[ \dot{s}_{cw} = k_{cw} ((c_l + c_r) - c_c) \]

#2: COLLISION AVOIDANCE

#3: RANGE AVERAGING

0.4 m

0.6 m
Trajectories in the Wind Tunnel
Conclusion
Take Home Messages

• Target localization (e.g., victims, chemical sources, anti-personnel mines, sound sources), coverage, mapping, inspection, and environmental monitoring are concrete applications for robotic sensor networks
• Sensing payload and robotic platform are typically very specific to the application
• Node control is data-aware: often close-loop control of mobility based on field data gathered
• Gas source localization as an example of sensing mission
• Multi-robot systems can perform distributed sensing of a chemical field (i.e., plume) and leverage a spatially adaptive array of sensors to carry out efficiently the mission
• Search and localization of a target can be sped up by the parallelism of a multi-robot system as long appropriate inter-robot coordination schemes are adopted
Papers


