A Look at Dependability and Synchrony in Distributed Mobile Robotic Systems: Adding Some Pragmatism to Theory

Xavier DÉFAGO

1) School of Information Science, Japan Adv. Inst. of Science & Tech. (JAIST)
2) PRESTO, Japan Science & Tech. Agency (JST)
Context / Motivation

- **Context**
  - Given: Multiple robots
  - Obtain: Single system

- **Requirements**
  - Deterministic solutions
  - Use little / no infrastructure
  - Weak / weaker / weakest assumptions

- **Characteristic**
  - Start from theory, move toward practice
**Outline**

- **Part I: Gathering problem**
  - Limited visibility
  - No communication
  - Compasses

- **Part II: Collision avoidance**
  - No visibility
  - Limited communication

- **Conclusion**
PART I

with:

• Samia Souissi (JAIST)
• Masafumi Yamashita (Kyushu Univ.)
Motivation

• **Context**
  • Autonomous robots
  • No infrastructure
  • No common knowledge (e.g., coordinate system)

• **Question**
  • What are the fundamental limits to robust coordination?
Gathering Problem

- **Gathering**
  - Set of robots located arbitrarily
    - ➔ All robots gathered at same location

- **Provides**
  - Agreement on common origin

- **Desired property**
  - Self-stabilization
System Model

- **Environment**
  - Euclidean plane
  - No landmarks
  - No obstacles
  - No boundaries

- **Robots**
  - Location: single point
  - Collocation possible; “no collisions”
  - Own coordinate system:
    - origin, directions, unit distance
System Model

- **Interactions**
  - **Vision**: get position of robots
  - No explicit communication
  - No global coordinates

- **Activations**
  - Cycle: *Look - Compute - Move - ...Sleep...*
  - Deterministic algorithm
  - **Oblivious** (stateless)
    => self-stabilizing
System Model

- Two Variants
  - Semi-Synchronous
  - Asynchronous
Semi-Synchronous


- **Activation Schedule**
  - **Atomic cycle:** *(Look - Compute - Move)*
  - **Parallel:** see same thing
  - **Sequential:** one see preceding movement
Gathering Problem

- **Goal**
  - From any configuration,
  - Eventually, all robots gather at single location

- **Difficulty: simple illustration**
  - Two robots A, B in semi-synchronous model
Asynchronous


- **No synchronization**
- Can be seen while moving
- Cannot anticipate others’ moves
Convergence vs. Formation

- **Convergence**
  - Solve problem asymptotically
  - Trivial to achieve (e.g., barycenter)

- **Formation**
  - Solve problem deterministically
  - Difficult to achieve
  - Need to break symmetry
Gathering w/Limited Visibility

- **Assumption**
  - Visibility graph connected initially

- **Safety**
  - Keep visibility graph connected

- **Why?**
  - Partition unrecoverable
    - Oblivious robots
    - => forget about each other’s existence
    - => gathering at more than one point (invalid)
Gathering Possible

- **...in Semi-synchronous model**
  - proved in [SY99]
  - **Non-oblivious** robots
  - **Full** visibility

- **...in Asynchronous model**
  - proved in [FPSW05]
  - **Oblivious** robots
  - **Limited** visibility
  - **with compass** (i.e., shared orientation)
Gathering w/Compass

[FPSW05]

- **Role**
  - Break symmetry

- **Idea (2 robots)**
  - Other robot to NW±90°:
    - Wait
    - Otherwise
    - Move to other robot
Question

[SDY06]

What if compasses are unreliable?

Especially

- Transient failures
- Interference
- Stabilization
Unreliable Compasses

- **Compass**
  - Function of (time, robot) -> direction

- **Perfect compasses**
  - Consistency: all compasses point to the same direction
  - Stability: compasses never change

- **Eventually-consistent Compasses**
  - There is a time (unknown) after which compasses are: consistent & stable
Unreliable Compasses

- Chaotic period
  - Safety: OK; Progress: *maybe*
- Good period
  - Safety: OK; Progress: OK
Unreliable Compasses

- Practically speaking...
  - Need good periods “long & frequent enough”
  - Algorithm tolerates unbounded number of transient failures
## Gathering w/Unreliable Comp.

<table>
<thead>
<tr>
<th></th>
<th>Perfect Compass</th>
<th>Eventual Compasses</th>
<th>No Compass</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Asynch. Model</strong></td>
<td><img src="https://example.com" alt="Content" /></td>
<td><img src="https://example.com" alt="Content" /></td>
<td>Impossible</td>
</tr>
<tr>
<td></td>
<td>$n \geq 4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$n = 2, 3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Semi-synchronous Model</strong></td>
<td><img src="https://example.com" alt="Content" /></td>
<td><img src="https://example.com" alt="Content" /></td>
<td>[SY99]</td>
</tr>
<tr>
<td></td>
<td>$n \geq 3$</td>
<td></td>
<td>Impossible</td>
</tr>
<tr>
<td></td>
<td>$n = 2$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**References:**
- [FPSW05]
- [SDY06]
Gathering w/Unreliable Comp.

- **Algorithm**
  - Use perfect compasses
  - or
  - Use semi-synchrony

- **Impossible case**
  - Asynchronous, n >= 4
  - Algorithm [SDY06] => breaks visibility
  - Best possible: Admits deadlock situations
Some Future Work

- **Unreliable compasses**
  - with bounded imprecisions

- **Faulty robots**
  - Byzantine robots

- **Dynamic problems**
  - Flocking, etc...
PART II

with:

• Julien Cartigny (Univ. Limoges, France)
• Nak Young Chong (JAIST)
• Rami Yared (JAIST)
• Matthias Wiesmann (JAIST)
Motivating Context

**Equipment**
- 4 Pioneer-3 robots
- Laptop
- Wireless (WiFi; bluetooth)
- Sonar (180°, 6-7m)

**Objective**
- Group movement without collisions

**Desired Properties**
- Decentralized solution
- Fault-tolerance
- Fail-safe behavior
Path Reservation

- **Robot knows**
  - own destination / path
  - own location
  - information in messages

- **Does NOT know**
  - others’ destinations
  - others’ location
  - others’ velocity
  - communication delays
• **Collision-free protocol**
  - Ensure no-collision
  - Fail-safe behavior

• **Local subsystem**
  - Individ. robot movements
  - Detect inert obstacles
  - Use sonars
Assumptions

- **Real-time**
  - Local subsystem
  - Sensors (sonars)
  - Motor control code

- Bounded Errors
  - Positioning system
  - Robot movement
  - Collision-free protocol
  - Motion planning
  - Network

- RT guarantees
System Model

- **Robots**
  - Have footprint
  - No vision
  - Ad hoc wireless communication

- **Positioning System**
  - Global $x$-$y$ referential
  - Robot can query asynchronously
  - Robots get own position

- **Communication**
  - Time-free
  - Two models: full, ad hoc
Path Reservation

- **Idea**
  - Similar to database locking

- **Primitives**
  - *acquire*: request lock on a zone
  - *release*: release zone

- **Benefits**
  - Prevent collisions
  - Provides some information on location
Path Reservation

- Desired properties
  - Mutual exclusion between conflicting requests
  - No starvation of requests (unless deadlock)
Anatomy of a Zone (errors)

• Errors
  • $\varepsilon_{\text{gps}}$: pos. system
  • $\varepsilon_{\text{tr}}$: translational movement
  • $\varepsilon_{\theta}$: rotational movement (incl. sensors)
Model 1: Fully connected

● **Model**
  ● All robots “know” each other
  ● All robots can communicate
  ● Communication is reliable asynchronous

● **Purpose**
  ● Few robots
  ● Limited area

● **Benefit**
  ● Simple, fault-tolerant solution
Model 1: Reservation

- **Idea**
  - Use **Total Order Broadcast** protocol

**Total Order Broadcast**

- Broadcast primitive
- Hosts deliver same sequence of messages

**Ordinary Broadcast**

```
p1
p2
p3
m1
m2
```

**Atomic Broadcast**

```
p1
p2
p3
m1
m2
```
Model 1: Reservation

- **Idea**
  - Use of Total Order Broadcast

- **Advantages**
  - Well-known requirements
  - Many algorithms (see survey [DSU04])
  - Fault-tolerant solutions
    (e.g., with unreliable FDs & maj. correct hosts)

- **Synchrony assumption**
  - E.g., unreliable failure detectors
  - FAIL => liveness violation
Model 1: Protocol

- **To move**
  - Get own position
  - Compute zone Z
  - TO-bcast(Request, Z)

- **When TO-deliver (Request, Z)**
  - If conflict => put Z in pending requests
  - Else lock(Z)

- **When lock(Z)**
  - Move along Z
  - Wait until destination reached
  - Get own position; compute new pre-zone
  - R-bcast(Release, Z - new pre-zone)
Model 1: Drawbacks

- **Limited scalability**
  - Protocol involves all robots
  - ... regardless of actual location

- **Limited flexibility**
  - Requires initial knowledge of all robots

- **Energy consumption**
  - Requires coverage of whole area
    (or supplemented by routing)
Model 2: Ad hoc

[YCDW06]

- **Limited communication range**
  - Known to all robots: $D$
    (NB: $D$ can be minimum of all ranges)

- **Asynchronous**
  - No bounds on message delays
    (e.g., because of retransmission).

- **Neighborhood discovery primitive**
  - Get set of direct neighbors
  - Query based primitive
  - Stronger guarantees
Model 2: Neighborhood Discovery

- **Primitive**
  - Query by robot $R$ at time $t$
  - Return set $\text{Neighbors}(r,t)$

- **Query period**
  - From *query* to *return*
  - For each robot $s'$; during query period:
    - $s'$ in range $\Rightarrow$ $s'$ in $\text{Neighbors}(r)$
    - $s'$ not in range $\Rightarrow$ $s'$ not in $\text{Neighbors}(r)$
    - $s'$ partially in range $\Rightarrow$ undetermined.
Model 2: Restrictions

- **Restriction**
  - Reservations within $D/2$ (- errors)

- **Ensures**
  - Cannot conflict without being “introduced”
Model 2: Discussion

- **Advantage**
  - Dynamic groups
  - Locality-preserving
  - Scalable

- **Drawback**
  - No fault-tolerant protocol yet

- **Synchrony assumption**
  - Neighborhood discovery
  - FAIL => safety violation
Future Directions

- **Protocol extension**
  - pipelining / interleaving

- **Parameter dimensioning**
  - robots density
  - robot speed / acceleration / braking distance
  - communication range
  - communication delays
  - errors
Conclusion

- **Part I**
  - Study limits of coordination
  - Tolerate faulty compasses

- **Part II**
  - Use communication & location
  - Reservation system
  - Full connected: simple, FT
  - Ad hoc: scalable, flexible

- **Theory: still a long way to go...**
References

- **Models**

- **Gathering**
Unreliable compasses


Other formation problems

Collisions


Path reservation