Fault-Tolerant and Dependable Robotics

Anders Lyhne Christensen

IRIDIA, Oct. 26th 2005

Content

- What is dependability?
- Why do we need dependability?
- How can it be achieved?
- Examples of existing research and results.
- Swarm robotics and dependability.
- Discussion…

What is Dependability?

Dependability includes the following attributes of a *computing system*:

- **Availability**: Readiness for correct service;
- **Reliability**: Continuity of correct service;
- **Safety**: Absence of catastrophic consequences on the user(s) and the environment;
- **Security**: The concurrent existence of (a) availability for authorized users only, (b) confidentiality, and (c) integrity.

A. Avizienis, J.-C. Laprie and B. Randell: *Fundamental Concepts of Dependability*. Research Report No 1145, LAAS-CNRS, April 2001.

Why dependability in robotics?

- In manufacturing the motivation is primarily *financial*.
- In human robot interaction the motivation is primarily *safety*.
- In hazardous environments and during rescue missions *dependability is a must*.

Experiments with the Stanford Robotics Platforms

Dependability through fault-tolerance

Fault-tolerance:

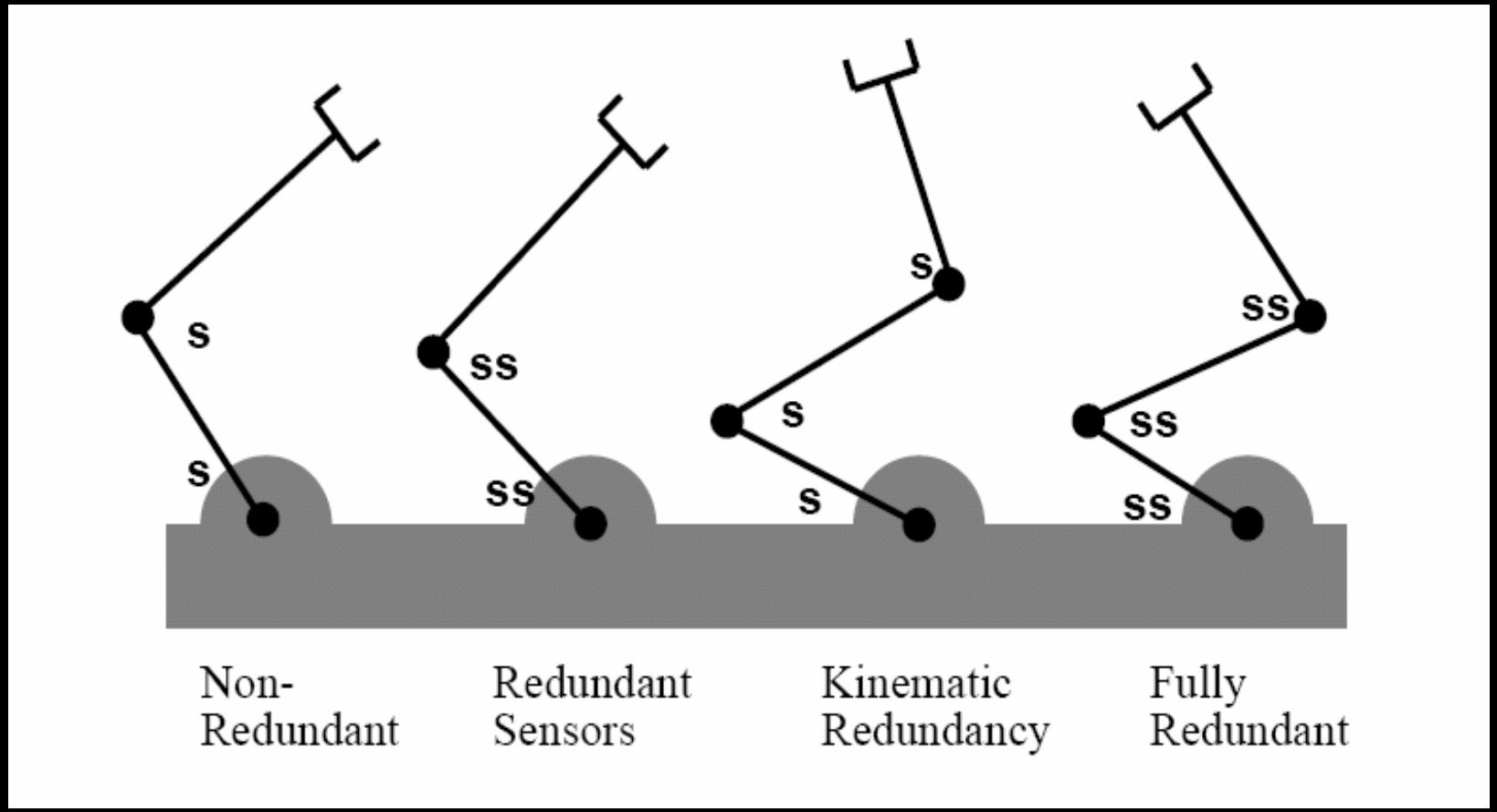
Fault-tolerance is the property of a system that continues to operate in the event of failure of some of its parts.

Achieving fault-tolerance

- Diversity
- Replication
- Redundancy

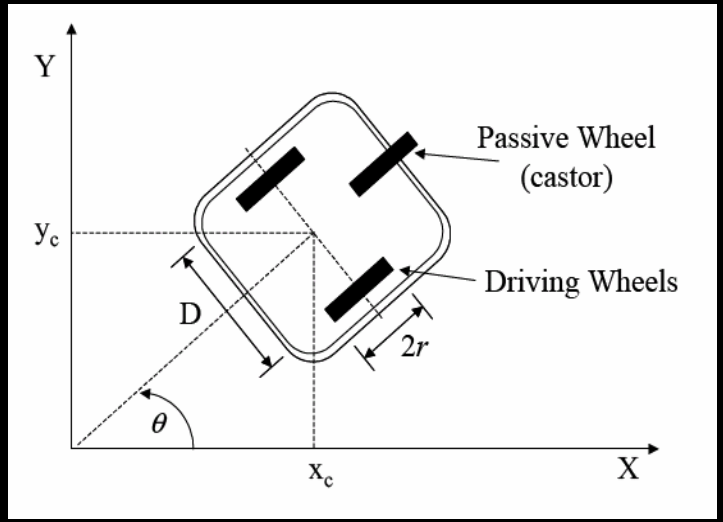
Fault detection is a major issue in robotics.

Example


S
S
SS
SS
S
S
SS
SS
Non-
Redundant
Redundant
Sensors
Kinematic
Redundancy
Fully
Redundant

Example – Detection

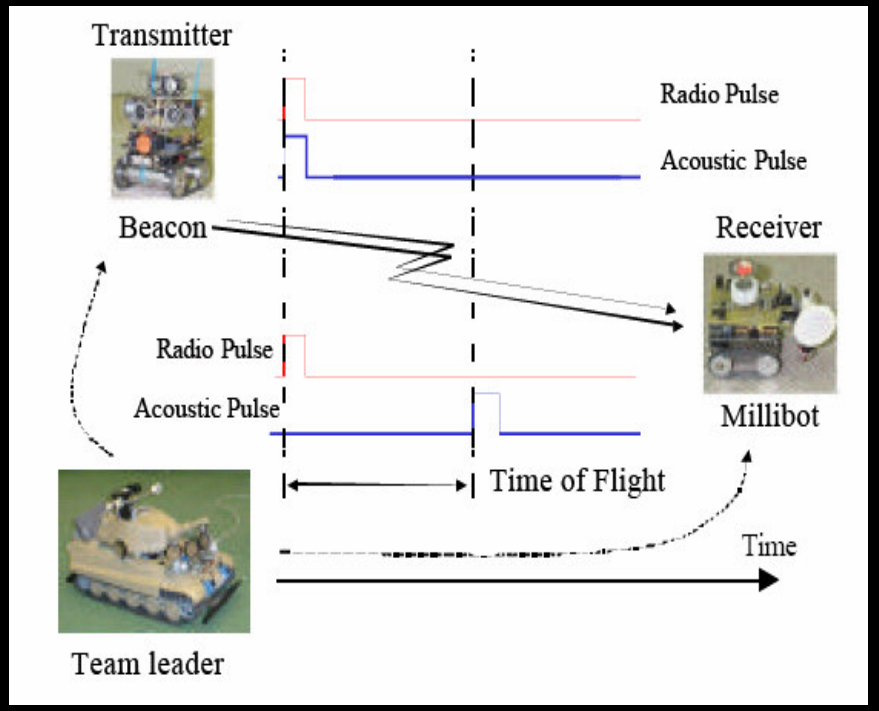
- A kinematic and a dynamic model is used and their predictions are compared with the sensor readings.
- Faults detected: Broken and deformed wheels, despite parametric uncertainty.



Dixon, W. E., I. D. Walker, and D. M. Dawson, "Fault Detection for Wheeled Mobile Robots with Parametric Uncertainty," *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Como, Italy, July 2001, pp. 1245-1250.

Example – Detection and tolerance

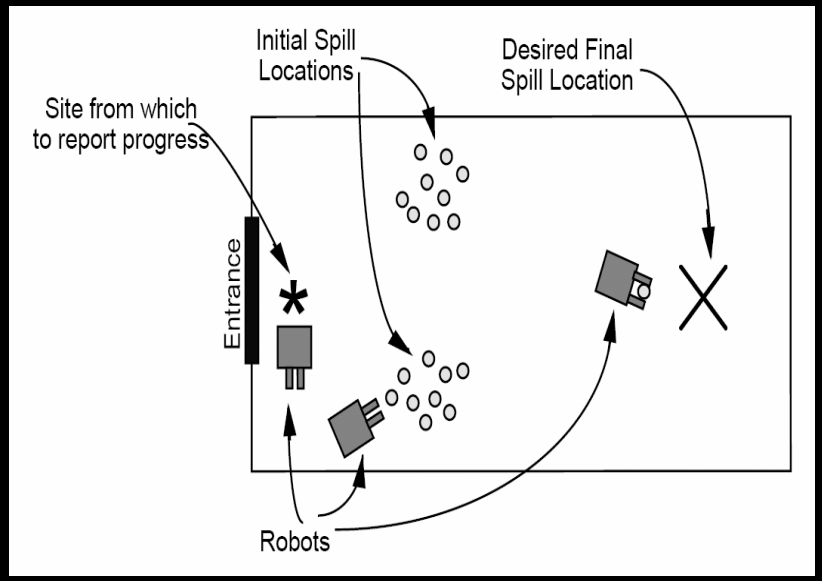
- Several “Milibots” use collective sensing and should therefore be able to locate each other.
- Radio and ultrasonic signals are used.
- “Faults” handled: No signal, multi-path signals, obstacles.
- Measurements are compared to “dead reckoning” information.
- Fault-tolerance: Detect and isolate/adjust the information used for localization.



R. Tins, L.E. Navarro-Serment, and C.J.J. Paredis, "Fault tolerant localization of teams of distributed robots," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2001, vol. 2, pp. 1061-1066

Example - Multi-robot Dependability

- Alliance:
  - Impatience
  - Acquiescence
- One robot, one task.
- Each robot learns its abilities and the abilities of others.
- Broadcast communication is used.
- Tested on real robots.



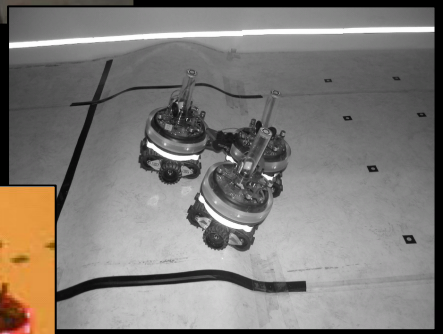
Lynne E. Parker, Oak Ridge National Lab. and MIT, several publications

Summary

- A large body of work related to mechanical failures has been performed.
- Fault detection plays a major role.
- Research and results are specific to tasks and platforms.
- Only few studies regarding dependability in cooperative robotics have been done.

Fault-tolerance in Swarm Robotics Systems

The hypothesis is that swarming robots are fault-tolerant because of *massive redundancy* – but no one has demonstrated or even tested if it is true. Is it?



Fault-tolerance in Swarm Robotics Systems

- What happens when a robot fails?
  - Detection $\rightarrow$ Isolation (+possible reconfiguration).
- How does a robot detect that it has some failure(s)?
  - Test if sensor-readings correspond to what is predicted (requires an internal world model)?
  - Train one or more neural networks?
- How should the swarm detect a broken member?
- Use more redundancy in tasks like path formation?

Where I will go from here…

IDEA:

Try to handle faults in a swarm-bot, where one or more robots fail (stop working completely) and the swarm-bot reconfigures to exclude the failed s-bots.

Both a behavior-based approach and via artificial evolution.

Discussion

How can dependability be incorporated in:

- Coordinated motion?
- Collective transport?
- Path formation?

Is there some cheap, general way to improve the fault-tolerance of a swarm robotics system?