Distributed Leader Selection in Robotic Networks

Iman Shames, André M. H. Teixeira, Henrik Sandberg, Karl H. Johansson

Australian National University and NICTA, iman.shames@anu.edu.au ; KTH – Royal Institute of Technology, {andretei,hsan,kallej}@kth.se

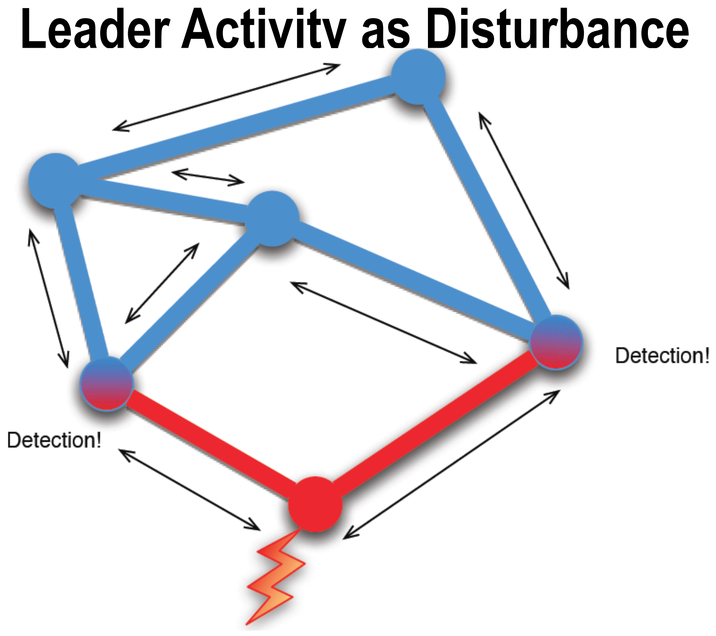
Contributions

Distributed Observer Scheme is used for:
- Leader Selection
- Bidding

In this work we use:

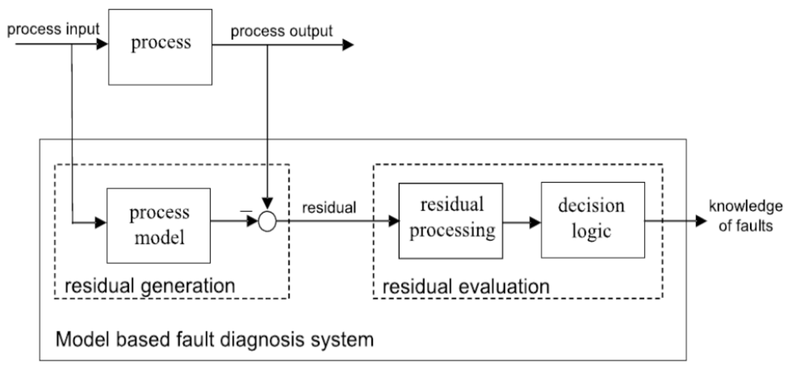
- Nonlinear observers to estimate the states of the agents in a robotic network.
- Linear Matrix Inequalities to prove the stability of the observers.
- Markov chains to prove the stability of leader selection algorithm.

Leader Activity as Disturbance



Agents assume no leader is present, so the network is "settled".

Introduction of the leader is similar to an external fault in the system, i.e. "unsettles" it.



Agents detect the leader by comparing the output of their "expected states" of the network calculated locally using their observers and the measured values.

To do so, they need to measure the states of their neighbours, and know the structure of the network.

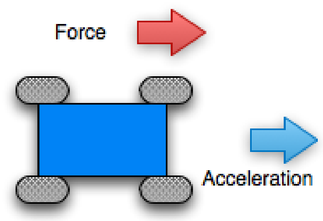
Leader Selection

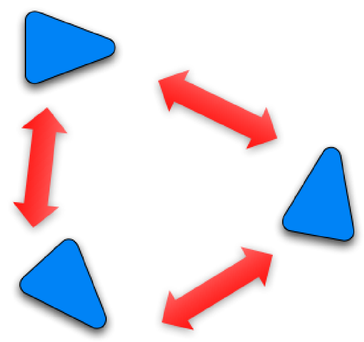
- Leadership and selection of a leader can be observed among social animals, e.g. human, and canis lupus.
- The presence of a leader is necessary to accomplish some of the tasks associated with robotic networks as well.

A Robotic Network Model

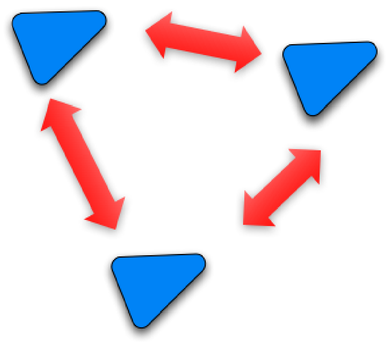
$\dot{\xi} = \zeta \quad \xi$: Position

$\dot{\zeta} = u \quad \zeta$: Velocity





Without a Leader

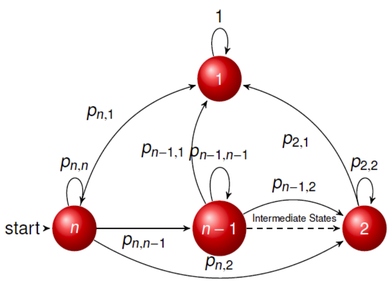


With a Leader

Leader Selection Algorithm

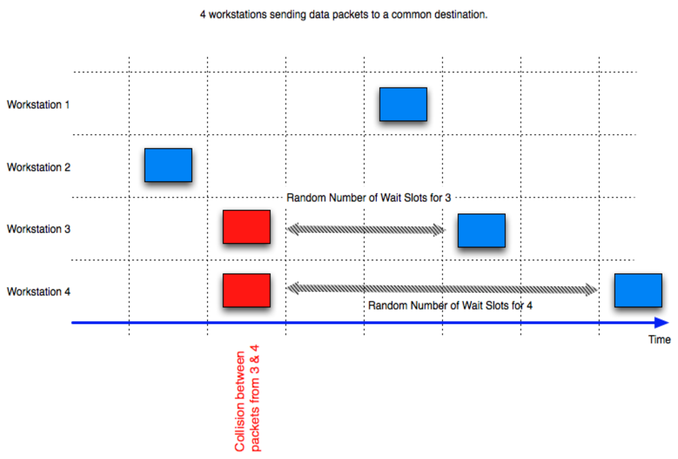
Algorithm (Distributed Leader Selection)

Wait for a random number of time slots $i \in \{1, \cdots, N\} \subset \mathbb{Z}$
i is not the leader;
do *Bid for Leadership* **loop**
if *It is the start of a time slot.* ***then***
  if *any agent has taken a leadership action before i* **and** *waiting time for i is up* ***then***
    agent i will not be able to bid for leadership; **return**;
  else
    agent i starts the leadership action;
  end if
  if *any other agent is taking a leadership action at the same time as i* ***then***
    Stop taking leadership action; Wait for a random number of time slots;
  end if
  if *It has been one time slot since the leadership action of i and no other agent has taken a leadership action.* ***then***
    i is the leader; **return**;
  end if
end if
end loop

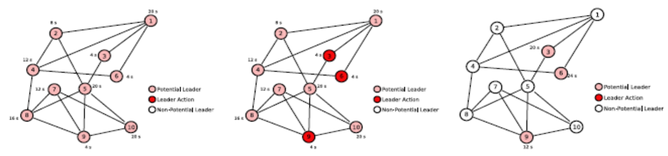


Example: The average number of steps for the leader selection algorithm to terminate for a formation of n=10 agents and N=10 is 25.

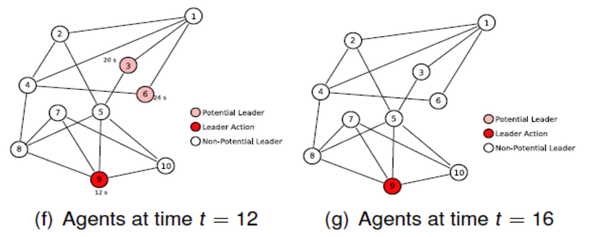
Comparison with Slotted ALOHA



Simulation Result



(c) Agents at time $t = 0$ (d) Agents at time $t = 4$ (e) Agents at time $t = 8$



(f) Agents at time $t = 12$ (g) Agents at time $t = 16$

Conclusion

- An algorithm for leader selection in multi-agent formations.
- Only local measurements and a knowledge of the structure of the underlying graph of the formation at each agent.
- The algorithm relies on existence of state observers at each agent.

Shames