Collective Decision-Making in Robot Swarms with Noisy Perception and Defecting Individuals

Nishan Srishankar (RBE)
Advisor: Professor Carlo Pinciroli (RBE/CS)

Abstract

Collective perception and decision-making is important for robots to maintain coherence in decentralized robotics systems. Existing algorithms assume every agent has perfect information and collaborates with neighbors to achieve consensus. However, in practical settings, agents have noisy information from sensor faults and sometimes have bugs or suffer deliberate attacks that turn agents into defectors.

In this work, we devise an application where robots are preloaded with a set of possible worlds and must collectively agree on which map corresponds to the current environment. Our approach uses Bayesian decision processes where robots combine both ground sensor information and neighbors' beliefs to optimize memory usage for learned representations.

Background & Motivation

- Robots are presented with spatially distributed environmental features that should be identified and agreed upon in a coordinated and decentralized fashion.
  - Decentralized: No single hub that transmits information or relays orders.
  - Scalable: Robust algorithm that works under varying swarm sizes or environmental sizes.
  - Redundant: The algorithm should be able to perform its task (sub)-optimally even after losing agents.
  - Anonymous: Should be able to perform tasks without robot identifiers.
- Real-world considerations: Sensor noise, adversarial robots, communication failures.
- Identifying features such as contaminant spill regions or areas under rubble in search-and-rescue operations from low-resolution images.

Difference from state-of-the-art

- Compared to other decision-making research problems in robotic swarms, in addition to communicating information with neighboring agents, robots take in **noisy ground truth information** and account for **defecting robots**.
- Compared to other network consensus problems, the agents/ nodes in our swarm don't maintain constant connectivity.
- Assuming that any implementable security methods (e.g. private/public keys, hash IDs, blockchains have already failed or been bypassed.

Defecting Robots

- Lying robots that masquerade as legitimate agents of the swarm and transmit incorrect information to reduce the effectiveness of the swarm.

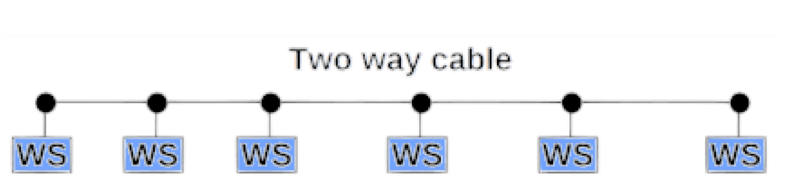

Figure 1- Carrier-sense multiple access with collision avoidance, a bus topology used in Kilobot communication

- Misinformation: Generate messages that can be passed off as genuine.
- Could be due to bugs in the system, are maliciously attacked etc. and don't follow the given algorithm.

Implementation

Assumptions

- Robots are:
  - Localized i.e. know positions w.r.t. global frame.

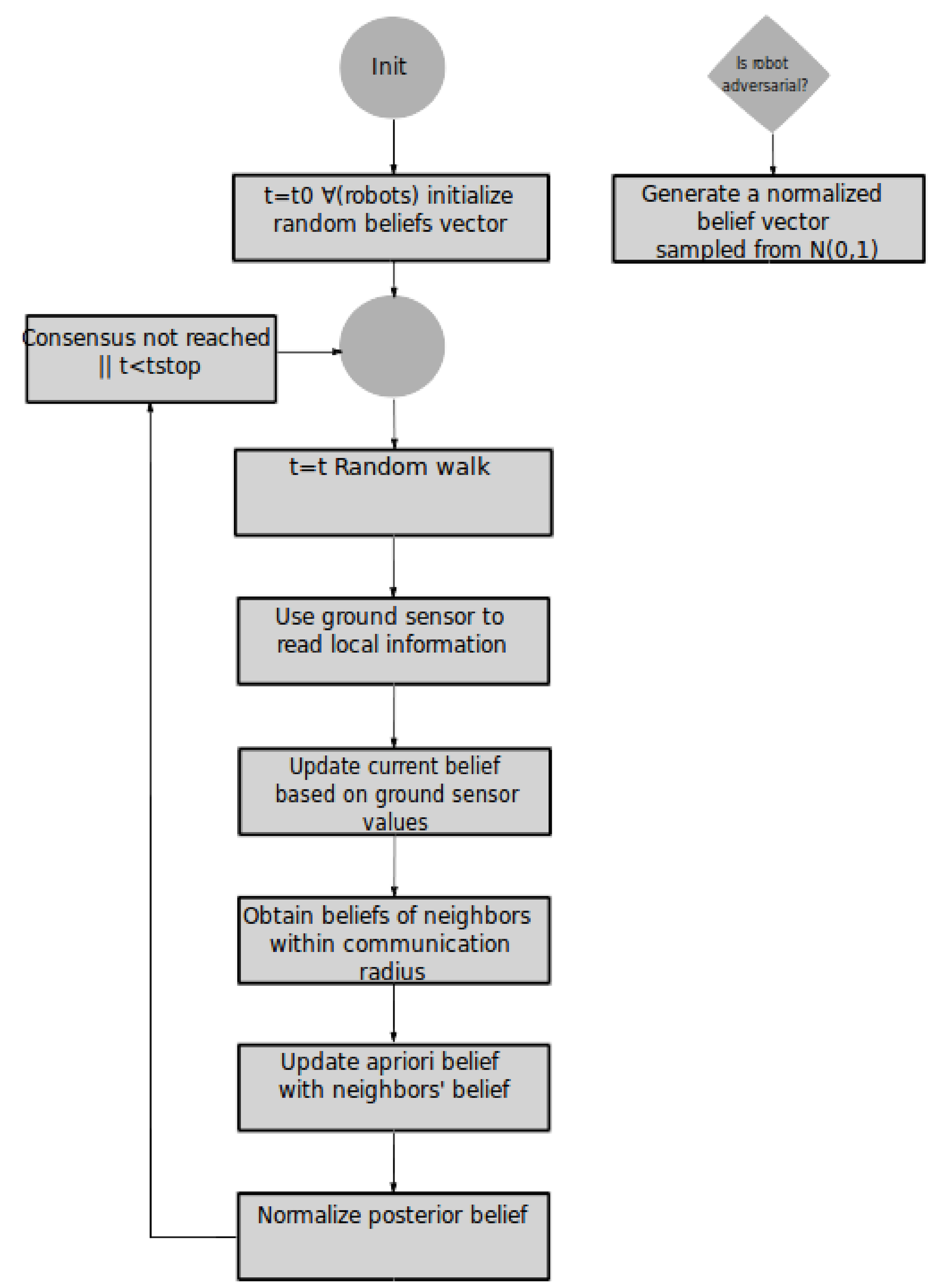

Figure 2- Exploration, collective perception, decentralized decision-making, and malicious attack algorithm

Implementation- In Depth

- Robots are preloaded with multiple representations that need to be compared between each other, and with the actual environment.
- At any instance in time, ground sensors can fail and provide wrong readings.
- Neighbors communicate a vector of beliefs as well as the current position.
- Main task is to avoid being influenced by defecting robot messages:
  - Rank received messages using a similarity measure.
  - Keep a time-varying threshold to eliminate unsuitable message candidates.
  - Generate a new motion vector to move away from robots that an agent disagrees with.

$$bel_t^l = \eta p(z_t|y^l, x_t) \prod_i^k p(y^l|m_t^i) bel_{t-1}^l$$

Equation 1- Bayesian belief update

Results

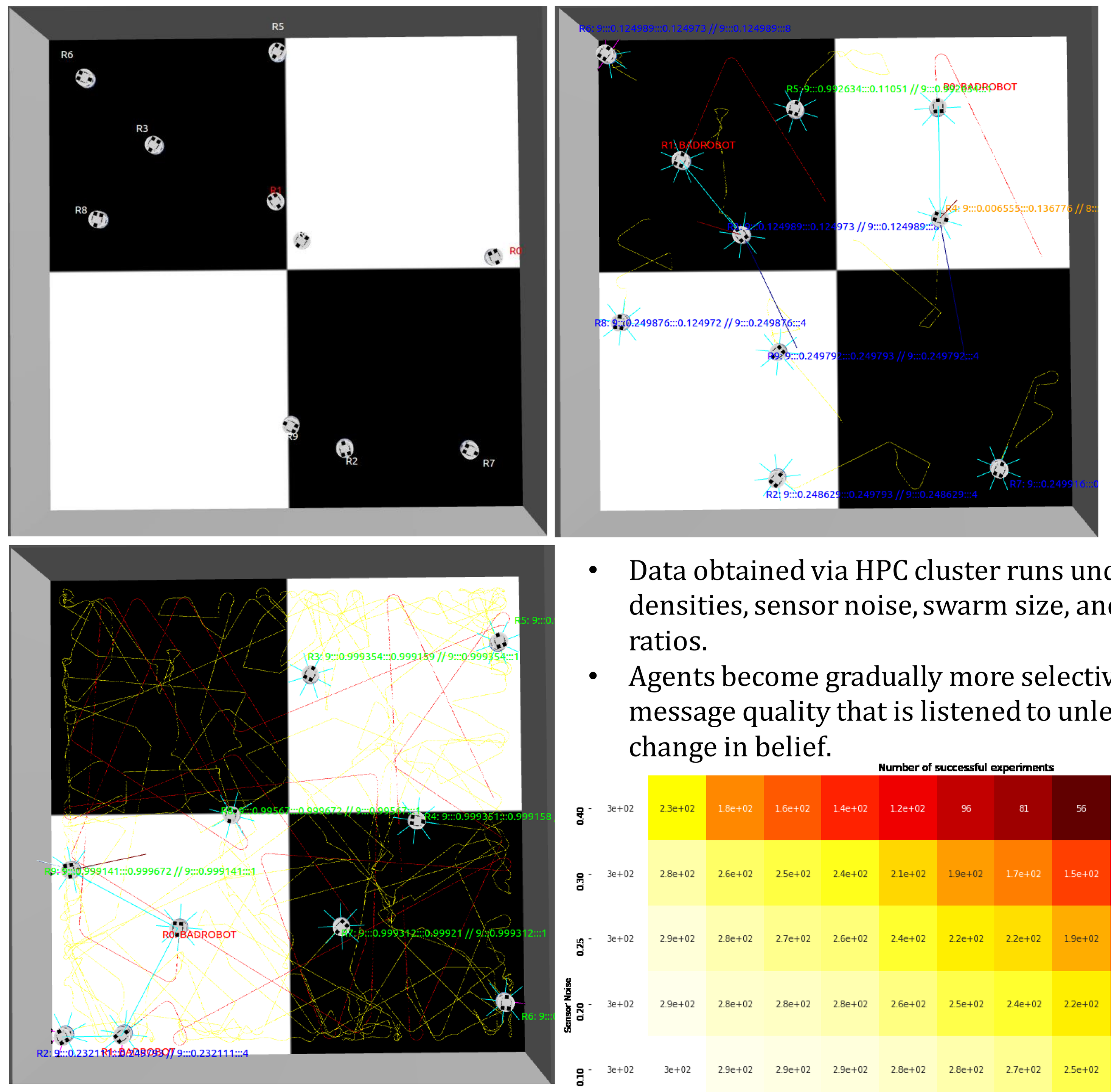

Figure 3- Algorithm at various stages of completion

- Data obtained via HPC cluster runs under varying densities, sensor noise, swarm size, and defecting robot ratios.
- Agents become gradually more selective about the message quality that is listened to unless there is a change in belief.

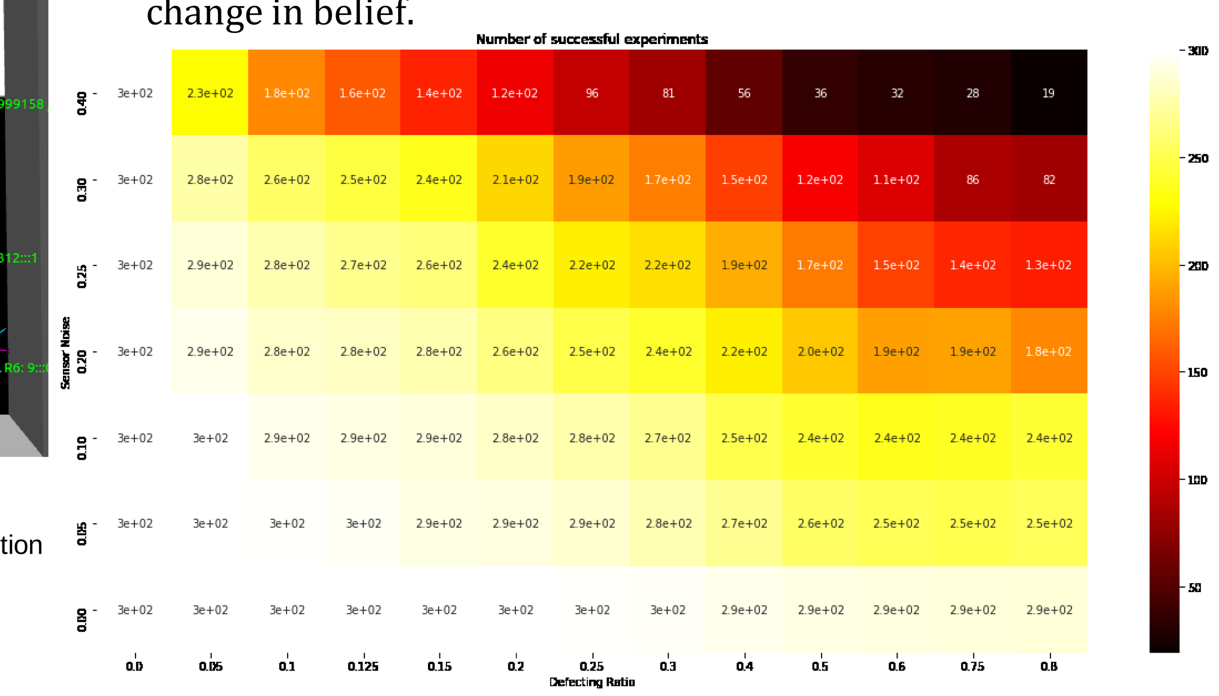

Figure 4- Heatmap of number of successful experiments for sensor noise vs. Defecting robot ratio

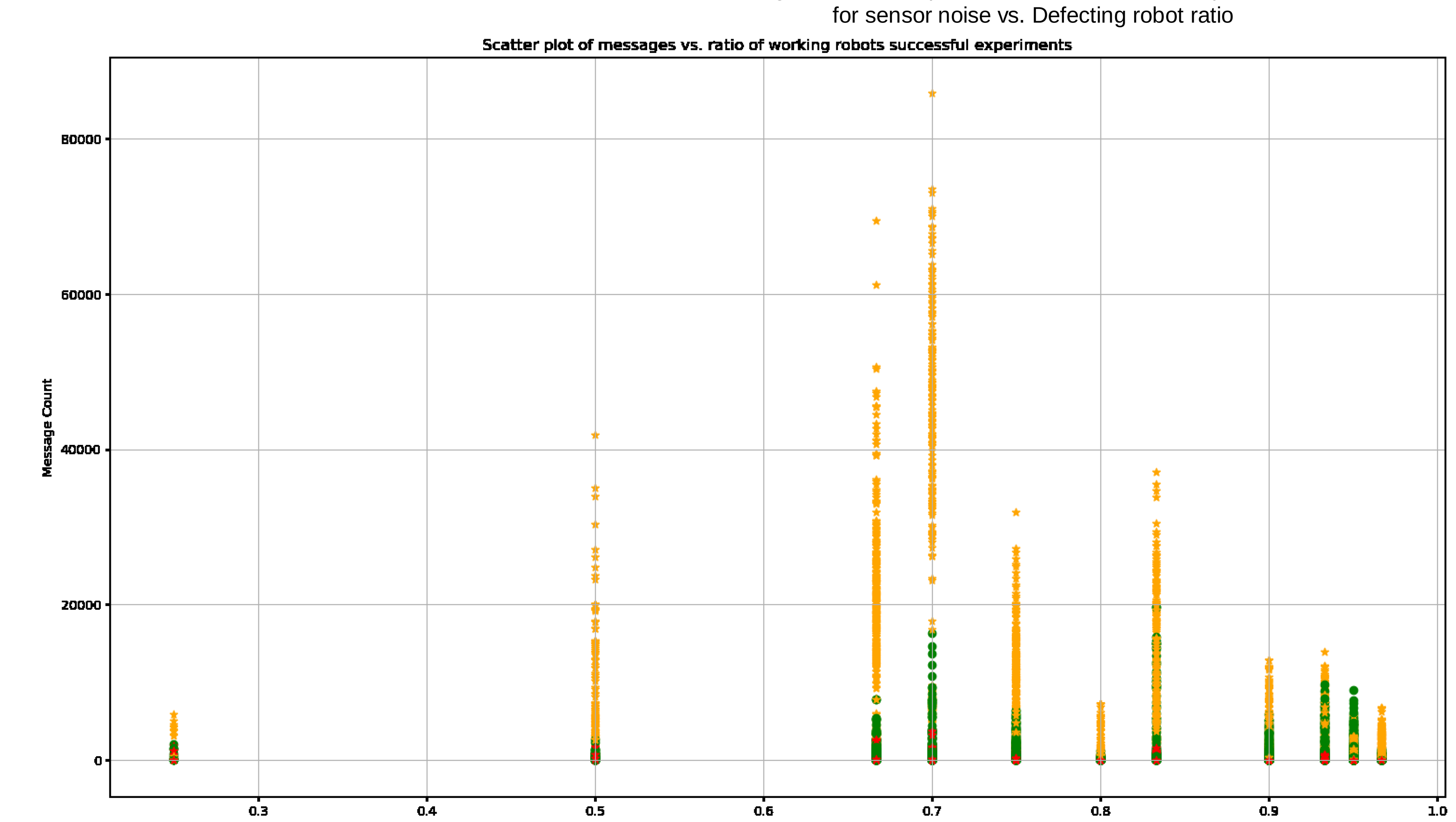

Figure 4- Message counts for successful experiments
[yellow= defecting robot messages, green=messages used for belief update, red=false messages used for belief update]

- Future work involves implementing and analyzing motion algorithms for defecting robots to be able to "infect" working robots with a higher probability than by random walk.