ME 312 - Project Proposal - Team 43

Srushti, Nikita, Tanmay, Divyanshu

February 1 2024

1 Introduction

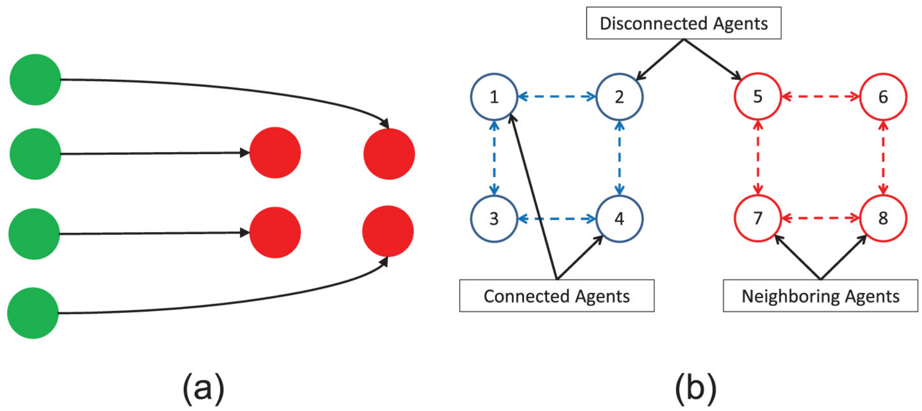
1.1 Overview

Motion planning and feedback control of multi-agent systems have been major areas of research over the past decades. Swarms of agents create interesting challenges in guidance and control due to the large number of agents, the small size of each individual agent, and the complicated dynamics. Specifically, the large number of vehicles moving in three dimensions makes collision avoidance a major challenge. Also, the limited computation and communication capabilities of each agent require the swarm reconfiguration algorithm to be very simple so that it can be run on board each small robot or vehicle in real time.

1.2 Challenges

The swarm reconfiguration problem consists of two parts: assignment and trajectory generation.
The **assignment problem** consists of finding the optimal mapping from a set of agents to a set of targets or tasks in order to minimize the total cost of interest.
The **optimal trajectory generation and collision avoidance** requires an algorithm that can solve the nonlinear optimization that minimizes the cost of the trajectory while satisfying collision avoidance and dynamic constraints.

2 Problem Statement and Approach



Visualization of problem statement and communication network
(a) Swarm reconfiguration (b) Visualizing connected and disconnected links
Image Reference: [Mor+16]

2.1 Objective

Our proposed algorithm addresses both optimal assignment and collision-free trajectory generation for robotic swarms in an integrated manner.

2.2 Input

The algorithm requires the desired shape of the swarm without pre-assigned terminal positions.

2.3 Optimal Assignment - challenge 1

We employ a distributed auction assignment approach that allows flexibility in the number of target positions in the assignment.

2.4 Trajectory Generation and optimization- challenge 2

Collision-free trajectories are generated using sequential convex programming in the reviewed method, we propose to test it with iterative Linear Quadratic Gaussian algorithm/ Approximate inference control algorithm.

2.5 Real-time Optimization - challenge 3

Model predictive control is utilized for real-time solving of assignment and trajectory generation with a receding horizon.

2.6 Features of algorithm

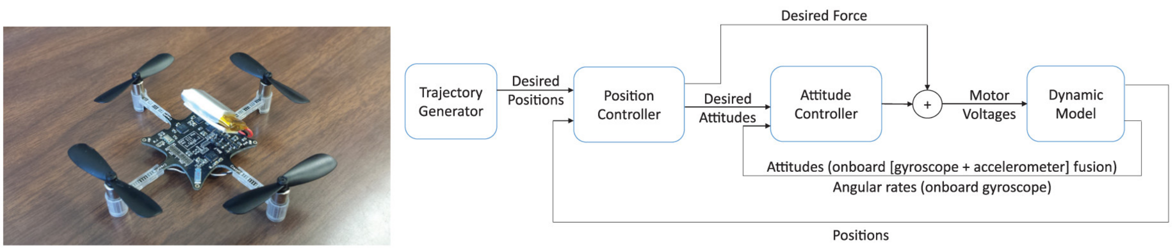
- **Algorithm Name**: The resulting algorithm is named Swarm Assignment and Trajectory Optimization (SATO).
- **Dynamic Adaptation**: The distributed auction algorithm and sequential convex programming are implemented using model predictive control, ensuring adaptability to dynamic swarm conditions.
- **Distributed Fashion**: SATO transfers a swarm of robots or vehicles to a desired shape in a distributed fashion. we propose to
- **Efficiency**: The algorithm determines each robot's position and optimal trajectory, ensuring fuel efficiency and collision-free movement.

2.7 Implementation Flow

1. Input desired swarm shape.
2. Utilize distributed auction for optimal assignment.
3. Generate collision-free trajectories through sequential convex programming/ iterative Linear Quadratic Gaussian algorithm/ Approximate inference control.
4. Implement real-time optimization using model predictive control with current state measurements.
5. Achieve distributed transfer of the swarm to the desired shape.

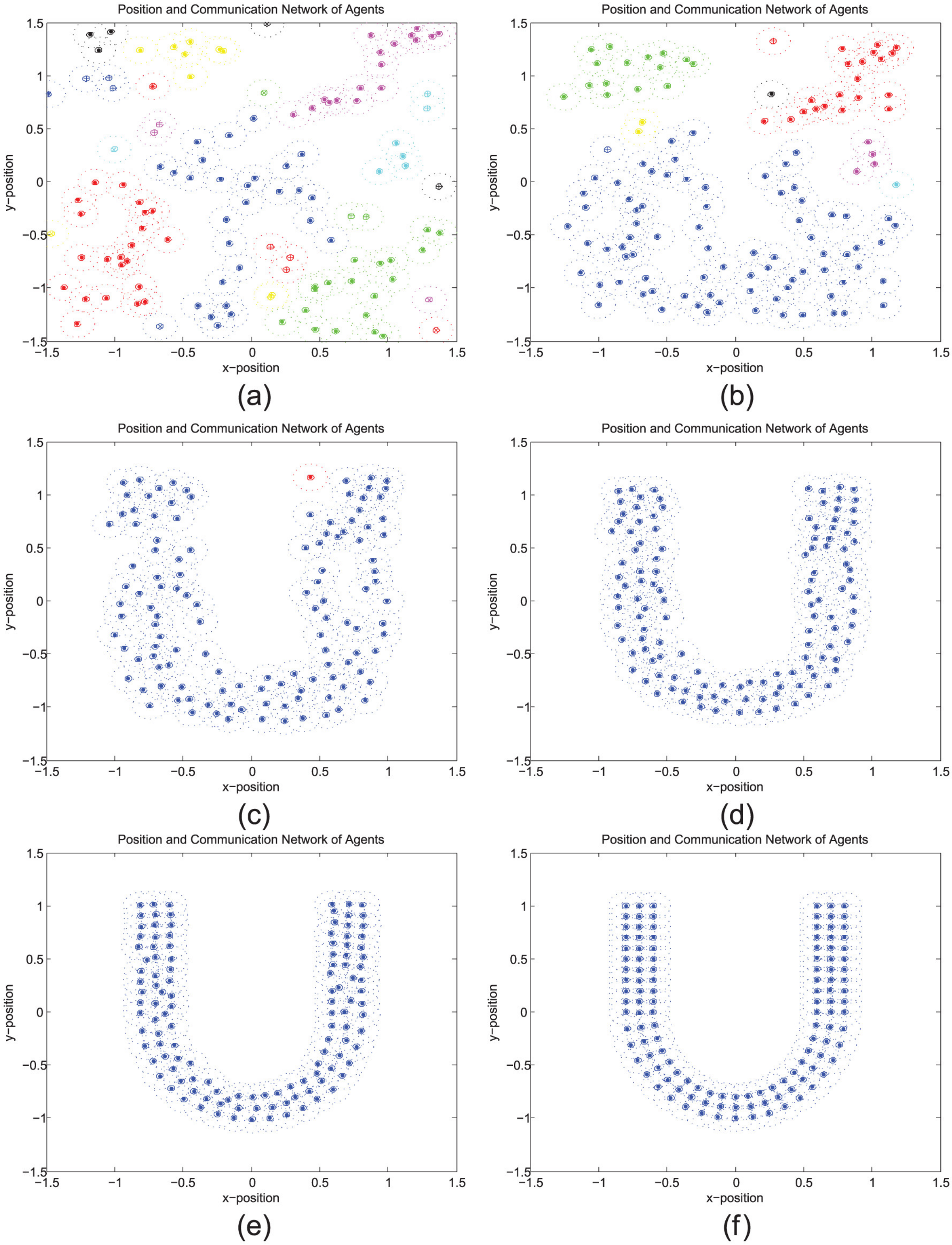
3 Expected Results

In the 2-D double integrator simulation, a swarm of 123 agents is randomly initialized and undergoes reconfigurations forming the letters 'U'.The connected agents are shown with the same color and shape while the dotted circle represents half of the communication radius. It shows the evolution of the communication network as the swarm converges to its desired shape. Figure (a) shows the initial swarm with many disconnected communication networks. In Figure (b), over half of the swarm is connected (blue) and the agents are roughly starting to form a 'U'. In Figure (c), all but one of the agents are connected so the assignment is nearly complete and the agents are moving towards their targets. Figures (d)–(f) show the movement of the swarm once the agents are all connected and the assignment is complete. These figures show the trajectory optimization part of SATO as the agents reach their desired positions in Figure (f). We aim to use a Quadrotor, ie. a 6 Degree of Freedom System as our Double integrator system for simulations. The control architecture of the Quadrotor is shown below.



(a) (b)

The position controller takes the current position and desired positions to generate the total desired force and the attitude commands which are tracked by the nonlinear attitude controller.
Image Reference: [Mor+16]



Various time instances in the reconfiguration simulation from a random swarm to a ‘U’ shape.
(a) k = 0. (b) k = 7. (c) k = 14. (d) k = 21. (e) k = 28. (f) k = 34.
Image Reference: [Mor+16]

References

[Tou09] Marc Toussaint. “Robot trajectory optimization using approximate inference”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1049–1056.

[MK11] Daniel Mellinger and Vijay Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.

[Mor+16] Daniel Morgan et al. “Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming”. In: *The International Journal of Robotics Research* 35.10 (2016), pp. 1261–1285.