INTER IIT
TECH MEET 13.0

MID
PREP

BHARAT FORGE | KALYANI

CENTRALIZED INTELLIGENCE FOR DYNAMIC SWARM NAVIGATION

About BharatForge

Bharat Forge Ltd., part of the Kalyani Group, is one of India's largest and most diversified engineering companies. Founded in 1961 and headquartered in Pune, Maharashtra, Bharat Forge initially focused on metal forging and evolved into a major supplier of high quality, precision-engineered components for the automotive and industrial sectors. Over the years, the company has expanded its footprint globally, acquiring companies and establishing centers in various countries, including Germany, Sweden, and the United States.

Bharat Forge today is known for its advanced manufacturing capabilities and is a major global supplier to industries ranging from automotive and aerospace to energy, rail, and defense. Commitment to innovation is evident in their extensive investments in research and development (R&D), focusing on new materials, product design, and manufacturing techniques. The company continues to adopt advanced artificial intelligence technologies to enhance production efficiency and maintain quality control.

Introduction & Motivation

Unlike external environments, robots placed in indoor workplaces cannot rely on GPS navigation. Currently, companies rely on ArUco markers for both navigation and object pose estimation purposes.

However, the goal of development in this field is to achieve complete autonomy in robot operations, without the need for human or environmental intervention. The time and labour demand for setting up markers in workplaces are issues that cease to exist if one develops an independent navigation system.

This problem statement requires you to develop and train such an algorithm (using ML, DL, RL, or hybrid approaches) for AMRs (autonomous mobile robots), allowing movement in dynamic environments where certain obstacles move regularly in unpredictable directions.

At Bharat Forge, they aim to deploy this software on their ground vehicles, including quadruped and mobile robots, for the purpose of optimized autonomous maintenance and inspection of machinery. They have multiple warehouses and assembly lines, each with unique floor plans and obstacles. Thus, they require a software that is environment-independent and is able to understand where it needs to navigate through continuous learning as it observes its surroundings.

Problem Statement

The problem focuses on designing a **Singular Brain for a Robot Swarm** tasked with performing optimized path planning in highly dynamic environments. You are required to design and train a multiagent system that can navigate in an environment where:

- **No GPS** is available.
- **Frequent and unpredictable changes** occur (e.g. moving obstacles, rearranged objects).
- The robot swarm must **collaboratively plan** the optimal path to achieve a set of tasks, while avoiding collisions and delays.
- Efficiency of the robot swarm is of greatest importance, so integration of **dynamic ranking** of navigation tasks based on traveltime.
- The solution must integrate **memory persistence**, where each robot can label and store dynamic changes in the environment within a shared, continuously updating database.

The ideal solution would be a software that maintains the same efficiency even when deployed in a new environment with a larger swarm of robots. Hence, scalability is our priority.

The robot swarm should be able to explore any given environment and create an adaptable database/map of object locations **without any manual intervention.**

Specifications

1. Development platform: ROS-compatible software (Isaac Sim, Gazebo,etc)

2. Environment selection: Any workspace (warehouse, factory, office,etc)

3. Minimum environment size: 10x10m

4. Minimum robot swarm size: 4

5. Robot type: Any mobile robot (ground vehicles, differential drive, quadrupeds, bipeds, 4WD)

6. Minimum number of unique objects in environment: 10

7. Dynamic obstacle type: e.g. humans, moving machinery

8. Minimum number of dynamic obstacles in environment: 3

Given that our focus is on scalability, there are no maximum limits

Bonus

An additional implementation of a chatbot interface embedded with an LLM for ease of assigning navigation tasks and receiving feedback of which robot is performing which task. An example of its functioning:

1. User types “Find me the nearest fire extinguisher.”
2. System understands object of importance is ‘fire extinguisher’ and searches the database.
3. Locations of robots and all fire extinguishers in the environment are compared and minimum distance is considered for optimum navigation task.
4. Nearest robot to fire extinguisher moves towards the fire extinguisher if unoccupied.
5. GUI output shows ‘I have assigned the task to robot x’.
6. Upon reaching the fire extinguisher, GUI output shows ‘Robot x has reached the fire extinguisher successfully.’

Note: Only if all deliverables are met will the bonus be reviewed and rewarded during scoring

Solution Deliverables

- The complete software should be put on GitHub. Your repository should contain a readme file with complete instructions for installation and execution of your software. You may also add videos or GIFs of how to setup and run your software.

- Simulations demonstrating how the swarm responds to dynamically changing environments, avoids obstacles, and completes navigation tasks. You must upload these simulations as videos in a separate folder.

There must be at least 4 simulations that properly demonstrate:

1. support for scalability with a larger environment
2. support for scalability with a larger swarm of robots
3. support for dynamic changes in object positions within the environment
4. the ability to navigate around dynamic obstacles

- A detailed technical report explaining the AI/ML models used, including how they were trained and how they handle the challenges mentioned (dynamic obstacle avoidance, goal setting, etc.)

- Final Presentation

Evaluation Parameters

1. **Effectiveness (40%):**
   - Obstacle Avoidance Accuracy: Measures success in avoiding collisions with both static and dynamic obstacles (5)
   - Autonomy in Task Completion: Evaluates the solution's independence in task execution without human intervention (5)
   - Real-Time Responsiveness: How well the system adjusts paths with environmental changes and change in object positions (5)
   - Formula: (0.35 * Obstacle Avoidance Score + 0.35 * Task Autonomy Score + 0.3 * RealTime Responsiveness Score)

2. **Scalability (30%):**
   - Environment Adaptability: Ability to operate in varying environments, from small to complex layouts. (5)
   - Swarm Size Flexibility: Measures performance consistency when scaled to larger robot swarms. (5)
   - Resource Management: Assesses memory and computation usage across multiple scales. (5)
   - Formula: (0.4 * Environment Adaptability Score + 0.4 * Swarm Size Flexibility Score + 0.2 * Resource Management Score)

3. **Innovation and Technical Depth (30%):**
   - Algorithm Complexity (5)
   - Problem-Specific Optimizations (5)
   - Formula: (0.5 * Algorithm Complexity + 0.5 * Problem-Specific Optimizations)

4. **Bonus (extra 10%):**
   - Chatbot Interface with LLM: Assesses functionality, accuracy, and integration of an LLM-based chatbot for task assignment. (3)
   - Task Feedback: Evaluates how well the system communicates robot status and task completion. (3)
   - Formula: (0.5 * Chatbot Functionality Score + 0.5 * Task Feedback Score)