Project Proposal: Project SwiftScout

Harsh Senjaliya and Dhairya Shah

November 14, 2024

Category: Multi-Robot/Swarm Actions

Audience: Acme Chief Technology Officer

Overview and Objective

Project SwiftScout aims to develop a multi-robot system using four TurtleBot robots within a simulated Gazebo environment to collaboratively search for and identify a human or object target. This system will demonstrate the application of swarm intelligence, coordinated exploration, and distributed decision-making. The primary goal is to showcase how decentralized multi-agent systems can achieve complex search objectives efficiently through cooperation and real-time data exchange.

The SwiftScout project has practical applications, such as enhancing search-and-rescue missions, surveillance, or industrial monitoring tasks, by leveraging multiple autonomous robots working in synergy.

System Design and Architecture

The system architecture for Project SwiftScout involves multiple interconnected subsystems:

- **Communication Subsystem:** All robots will communicate using ROS2's built-in Data Distribution Service (DDS), enabling real-time decentralized communication of key data (e.g., location, detected objects, task progress).
- **Perception Subsystem:** LiDAR and RGB-D cameras provide each TurtleBot with the ability to detect obstacles, identify targets, and map the environment. OpenCV-based object detection models will be employed for identifying the human or object target.
- **Navigation Subsystem:** The TurtleBots use ROS2 Navigation Stack for autonomous path planning, incorporating SLAM algorithms for mapping and exploring unknown environments. Decentralized navigation ensures coverage while minimizing redundancy and collisions.
- **Swarm Coordination Subsystem:** Algorithms for swarm behaviors, such as distributed task allocation, leader-follower dynamics, and adaptive area division, will guide the collective behavior of the robots.

Prototype Product and Usage Context

The SwiftScout system simulates real-world scenarios where a group of autonomous robots collaborates to search for a hidden human or object in an unknown environment filled with obstacles. This application is relevant to disaster response operations, environmental monitoring, and automated surveillance. The system will demonstrate robust swarm behavior, including cooperative search, real-time data exchange, and obstacle navigation.

Development Process and Quality Assurance

The development process will follow Agile principles, structured into three-week iterative sprints, with daily stand-up meetings for continuous progress evaluation. Quality assurance measures include:

- **Version Control:** Using GitHub repositories to manage all changes and ensure collaboration through code reviews.
- **Continuous Integration (CI):** Automated testing pipelines will validate each code change via unit, integration, and simulation tests.
- **Code Review and Documentation:** Comprehensive documentation for each ROS node, including descriptions of inputs, outputs, and interactions, ensuring transparency and replicability.

Technologies and Tools

- **Platform:** TurtleBot 3 simulation in Gazebo
- **Programming Language:** Python/C++
- **ROS Version:** ROS2 Humble Hawksbill
- **Libraries/Packages:** ROS2 Navigation Stack, OpenCV, DDS-based inter-robot communication
- **Licensing Information:** Reused libraries/packages will comply with their licenses (e.g., BSD, MIT). The project code will adopt a BSD license for compatibility with open-source standards.

Applicable ROS REPs

- REP 103 (Standard Units of Measure and Coordinate Conventions)
- REP 117 (Robot Naming Conventions)

Algorithms and Techniques

- **Swarm Coordination Algorithms:** Distributed control models for leader-follower dynamics, adaptive area division, and task allocation.
- **Object Detection Techniques:** OpenCV-based image recognition models for target identification.
- **Path Planning and Navigation Algorithms:** SLAM for environment mapping, RRT* (Rapidly-exploring Random Trees) for navigation optimization, and decentralized exploration strategies to prevent overlap.

Technical and Project Risks and Mitigation Strategies

- **Scalability Issues:** Initial focus on a four-robot system to ensure stability, followed by stress-testing for expanded swarm sizes.
- **Communication Latency:** DDS optimizations and lightweight data structures will minimize network congestion and delays.
- **Navigation Conflicts:** Fine-tuning SLAM and collision avoidance algorithms to prevent navigation errors in complex environments.

Team Contributions and Pair Programming

Harsh and Dhairya will work collaboratively, alternating between driver (active coding) and observer (reviewing/testing) roles. Pair programming sessions will be logged with details on tasks completed, issues faced, and solutions implemented. Daily stand-ups will provide a forum to synchronize progress and address challenges.

Final Deliverables

- **Source Code and Documentation:** Professionally documented codebase, meeting ROS2 best practices, with explanations for each component and its interactions.
- **Functional Demo:** A Gazebo simulation demonstrating the collaborative search behavior, with clear visualizations of real-time coordination among TurtleBots.
- **Technical Presentation:** A recorded 5-minute video covering the system's architecture, implemented algorithms, and a live demonstration showcasing the robots' coordinated search for the target.

System Architecture Diagram