Test Document

Human-autonomous teamwork of ground and air vehicles

Team Members
Yavanni Ensley, yensley2022@my.fit.edu
Younghoon Cho, ycho2021@my.fit.edu
Jaylin Ollivierre, jollivierre2022@my.fit.edu

Faculty Advisor
Thomas Eskridge, teskridge@fit.edu

1. Introduction

1.1. Purpose

The test document will list all tests that will be needed to successfully run the multi-agent human autonomous teamwork system with ground and air vehicles.

1.2. Objective

Specifically, the performed tests aim to:

a. Ensure basic control and communication of robots function reliably.
b. Ensure predator-prey and search & rescue algorithms operate as intended
c. While performing objective (b), ensure humans can take control of the robot, while algorithm makes change of remaining robots movements.
d. Ensure system performance in terms of latency, robustness, and reliability.
e. Ensure the human operator can effectively take over robot control without weakening team performance.

2. Basic Control Test & Setup

2.1. General Movement Commands

1. Check device has decent build quality.
    a. Check for broken/glued parts.

2. Test basic movements (forward, backward, angle).
    a. Input each commands with ROS2
    b. Ensure consistency for devices
        i. (check the moves are within acceptable difference: ±5%).
    c. Verify the robot immediately stops when commanded (avoid collision)

3. Tests should be done for all ground vehicles.

2.2. ROS2 Communication

1. Confirm each device correctly receives and sends each topic as a subscriber and publisher.
    a. Run each Node and check with RQT graph
    b. Verify message frequency is within excepted range (possibly 10Hz = 0.1sec)

2.3. Sensor Output

1. Verify each device's sensor works (Lidar, Camera) without significant delay.
    a. Check with rviz.

3. Interface (GUI) Test

3.1. GUI Consistency / Button Functionality

1. Ensure for each button, intended action is triggered. (Latency Threshold: 1ms)
2. Ensure consistency between real environment and GUI visualization.

3.2. Robot Switching

1. Ensure the operator(human) can switch between robots without glitches.
2. Verify previously controlled robot(human → algorithm) continue tasks.
3. (Latency Threshold: 500ms)

3.3. GUI Visualization

1. Ensure operator(user) clearly understands and operates the GUI without performance decrease.
    a. GUI layout, labels, controls should be easy to understand
2. Operator(user) should be able to view the total map if needed)

4. Multi-Agent Collaboration Tests

4.1. Information Sharing

1. Ensure robots share precise data
    a. Location of each other (200ms).
    b. Location of target when found (200ms).

4.2. Map Generation

1. Ensure the accurate Map is generated with
   Multi-device SLAM(Simultaneous Localization and Mapping) for ground vehicles.
2. Ensure the generated map is sent to the GUI within acceptable time(500ms)

4.3. Efficiency Test

1. Assure the robots can generate the map efficiently for each robot combination
2. Measure time of Map generation time for each combination.
    a. Single robot (Agile Limo pro),
       multiple robot (Agile Limo pro),
       mixed combinations (Agile Limo pro + Unitree Go2)
3. Measure time of capture similar to (2)

5. Predator-Prey / Search & rescue algorithm test

5.1. Selection of search algorithm

1. Explore and test different algorithms (Running Time , implement difficulty)
    a. Q learning (light deep Q learning)
    b. Multi-Agent Reinforcement Learning
    c. Other Algorithms
2. Initially, the algorithm will work for Stationary targets.
3. After the first algorithm will aim to track escaping(non-Stationary) targets.

5.2. Human override during human control

1. Ensure smooth transition between algorithmic control and human operator intervention.
2. After humans take over control or have given control back,
   the algorithm will recalculate the search algorithm.

5.3. Target recognition algorithm

1. When input(Target image) is given, use Convolutional Neural Networks and YOLO to track the target.
2. Check the runtime and memory usage of the algorithm to avoid overload. After, decide the frequency of the algorithm.
3. Initially, start without this method (use geographical target[y,x] instead)

6. Performance and reliability test

6.1. Latency Test

Ensure control(commands) are executed quickly

- GUI button press → robot action (robot respond 300ms)

6.2. Disconnection Test

The system continues to work without freezing even if
one(or more) robot disconnects due to low battery, poor internet connection.

6.3. Emergency Stop

The system should halt within 300ms when the GUI button is pressed.
(needed to avoid unexpected collision)