Autonomous Collaborative Ground Traversal of Discontinuous Terrain

Spring 2023 Project Test Plan

TEAM A

Dhanvi Sreenivasan | Hari Shankar Ravisankaran | Sandhya Vidyashankar | Sankalp Chopkar | Sudhansh Yelishetty

Sponsors: Prof. Zeynep Temel | Prof. Katia Sycara | Sha Yi

Date: February 12, 2023

Introduction

This document outlines the tests planned by Team A - Mind The Gap for the Spring 2023 semester. The goal of these tests is to verify that our subsystems are on track to meeting our mandatory requirements as laid out in the Conceptual Design Review created at the end of Fall 2022 (Appendix A). The results of these tests will be reported during Progress Reviews and the Spring Validation Demonstration. Each test is designed to validate specific critical functions and interfaces in our system. Furthermore, we present a chronological breakdown of our high-level milestones that we plan to achieve by each progress review.

Logistics

Location of Testing

For the most part, all of our tests are planned to take place in Newell-Simon Hall Room 1604. We choose this location because we will be using the Vycon system there to localize our agents for the SVD, as well as performing other experiments relying on localization. We plan to modify the testbed to demonstrate gap crossing behavior more effectively.

Personnel

For each test, we provide the key personnel required to perform said test. This is the minimum number of people as some tests might require recalibration of the VyCon which needs one more person. Each test is primarily run by that subsystem's leader and one more person.

Equipment

All of our work will be based on the Khepera IV platform. We have secured 3 agents for the first half of the semester to conduct tests and will receive a few more as we get closer to the SVD. The Vycon camera system is fixed to NSH 1604, and we will be using it exclusively for localization this semester. FInally, we run all communications through an MRSD Computer acting as our main ROS server

Schedule

Date	PR	Capability Milestones	Tests	Associated Milestones
Feb 15	1	Mechanism design finalized, comms with Khepera set up, initial planner written	1, 6, 8, 10, 13	MF1, MF2, MF4
Mar 2	2	Initial Mechanism Prototype tested, planner and task allocation in simulation, Localization and Map building	2, 5, 6, 9, 11, 15	MF1, MF2, MF3, MF4
Mar 23	3	Final Mechanism Prototype tested, Planner and Task allocation integrated to hardware, Full Navigation stack readiness	3, 7, 12, 14	MF1, MF2, MF3, MF4, MF6
Apr 6	4	Testing of all subsystems, integration	All tests	MF1, MF2, MF3, MF4, MF5, MF6

Tests

Test 1: Payload Test for Khepera IV

Test Name	
Khepera IV Payload Test	
Objective	
To determine the payload capabilities of Khepera IV for determining the permissible weight of the mechanism	
Equipment	Khepera IV, MRSD Desktop 4, Enclosure-shaped modular dead weight
Elements	Physical Coupling Mechanism
Personnel	Sankalp Chopkar
Location	NSH 1604
Procedure	
1. 3D print the enclosure and mount it on the top of Khepera 2. Add dead weights and drive the enclosure along the test path 3. Repeat step 2 until Khepera resists motion without any damage	
Validation / Results	
● Maximum payload value is obtained ● Driveability of Khepera under load is better understood	

Test 2: Pin and Hole Mechanism Alignment Tolerance

<table>
<tr><th colspan="2">Test Name</th></tr>
<tr><td colspan="2">Pin and Hole Mechanism Test on Individual Agents</td></tr>
<tr><th colspan="2">Objective</th></tr>
<tr><td colspan="2">To successfully validate the ‘work-in-motion’ of the pin and hole mechanism on individual agents</td></tr>
<tr><td>Equipment</td><td>Khepera IV, Mechanism Enclosure, PS4 Controller</td></tr>
<tr><td>Elements</td><td>Physical Coupling Mechanism</td></tr>
<tr><td>Personnel</td><td>Sankalp Chopkar</td></tr>
<tr><td>Location</td><td>NSH 1604</td></tr>
<tr><th colspan="2">Procedure</th></tr>
<tr><td colspan="2">1. Manufacture, procure and assemble the elements of the mechanism
2. Actuate the linear stage actuator to test the pin motion throughout stroke
3. Insert the test pin in the hole and actuate the solenoid lock</td></tr>
<tr><th colspan="2">Validation / Results</th></tr>
<tr><td colspan="2">● Linear to and fro motion of the pin should be seamless, this will validate the alignment of the components
● The pin should slide along the roller supports
● The Solenoid lock pin should actuate in the slot given in the pin</td></tr>
</table>

Test 3: Final Coupling Mechanism Test

<table>
<tr><th colspan="2">Test Name</th></tr>
<tr><td colspan="2">Physical Coupling Mechanism Test</td></tr>
<tr><th colspan="2">Objective</th></tr>
<tr><td colspan="2">To successfully validate the coupling between two agents</td></tr>
<tr><td>Equipment</td><td>Khepera IV, Mechanism Enclosure, PS4 Controller</td></tr>
<tr><td>Elements</td><td>Physical Coupling Mechanism</td></tr>
<tr><td>Personnel</td><td>Sankalp Chopkar</td></tr>
<tr><td>Location</td><td>NSH 1604</td></tr>
<tr><th colspan="2">Procedure</th></tr>
<tr><td colspan="2">1. Move the agents to the desired positions to initiate the coupling sequence (preferably, edge-to-edge contact)
2. Actuate the pin
3. Actuate the solenoid lock following the complete actuation of the pin</td></tr>
<tr><th colspan="2">Validation / Results</th></tr>
<tr><td colspan="2">● The pin should self-align with the hole in case of minor misalignments
● The coupling sequence should be completed without any lateral displacement of the agent</td></tr>
</table>

Test 4: Coupled Motion Test

Test Name	
Agent's Coupled Motion Test	
Objective	
To successfully validate the coupled motion of multiple agents and cross the gap	
Equipment	Khepera IV, Mechanism Enclosure, PS4 Controller
Elements	Physical Coupling Mechanism
Personnel	Sankalp Chopkar
Location	NSH 1604
Procedure	
1. Navigate and Localize the agents in the coupled state 2. Traverse three agents across the gap in coupled state	
Validation / Results	
● Coupled agents should locomote as one body (No motion w.r.t each other) ● One agent should be able to bear the weight of other agent while crossing the gap	

Test 5: SLAM Toolbox Unit Test

Test Name	
SLAM Toolbox Test	
Objective	
Get SLAM toolbox working on a single agent in a structured environment	
Equipment	Khepera IV robot with LRF module, WiFi router for test network, MRSD Desktop #4
Elements	Localization pipeline: Khepera 4 low-level hardware drivers, agent-server communication, server data unpacking and broadcasting, ROS localization package
Personnel	Sudhansh
Location	NSH 1604
Procedure	
1. Place powered-on agent with LRF module in the test area. The agent should have code to read and transmit all sensors of interest 2. Set up test area to have visually recognizable landmarks and obstacles 3. Use WiFi router to get laptop and agent on the test network 4. Establish server-agent communication: agent should now stream data to the server and be able to receive commands 5. Run data unpacker. Sensor data is packaged on the agent-side and must be unpacked server-side. Sensor data should be broadcast as topics on ROS network 6. Use RVIZ to check sensor data a. Visualize odometry data. When the robot is manually moved to follow a square track with a length of 30 cm, odometry data should also show the robot moving in a square b. Visualize LIDAR data. The resulting scan should show parts of recognizable landmarks and obstacles near robot 7. Configure SLAM Toolbox to use odometry data and LIDAR data for localization and mapping. It should output a robot pose and map of test area 8. Use teleoperation to command the agent to move around the test area. This should fill out more of the map, and the robot pose should update	
Validation / Results	
● Odometry visualization: resulting odometry data is roughly in the shape of a square with nearly right angles. The length of the side of the square is within 10% of the expected length ● LIDAR visualization: laser scan data should show obstacles and landmarks placed in the test area, and be easily recognizable. No obstacles should be missed. ● By inspection, the estimated robot pose should be within 30 cm of the actual pose.	

Test 6: Localization Unit Test

<table>
<tr><th colspan="2">Test Name</th></tr>
<tr><td colspan="2">Localization Accuracy Test</td></tr>
<tr><th colspan="2">Objective</th></tr>
<tr><td colspan="2">Quantify accuracy of localization algorithm in test area with known map</td></tr>
<tr><td>Equipment</td><td>Khepera IV robot with LRF module, WiFi router for test network, MRSD Desktop #4, VICON cameras, computer connected to VICON system</td></tr>
<tr><td>Elements</td><td>Localization pipeline: Khepera 4 low-level hardware drivers, agent-server communication, server data unpacking and broadcasting, ROS localization package</td></tr>
<tr><td>Personnel</td><td>Sudhansh</td></tr>
<tr><td>Location</td><td>NSH 1604</td></tr>
<tr><th colspan="2">Procedure</th></tr>
<tr><td colspan="2">
1. Place powered on agent with LRF module in test area. Agent should have code to read and transmit all sensors of interest
2. Setup test area to have visually recognizable landmarks and obstacles
3. Use teleoperation to move the agent around in the test environment. Collect the following data during this step:
4. Use the localization system to produce a map of the test area and an estimate of robot pose. These shall act as experimental data
5. Utilize VICON system to produce a map of the test area and an estimate of robot pose. These shall act as ground truth
6. Continue to move agent around in the test setup until a comprehensive map has been produced
7. Using data acquired from the test:
8. Compare the accuracy of the experimental map with the ground truth map. Produce metric of accuracy of map
9. Compare the accuracy of the experimental pose with the ground truth pose. Produce metric of accuracy of pose estimate
</td></tr>
<tr><th colspan="2">Validation / Results</th></tr>
<tr><td colspan="2">
● Map accuracy: experimental map has no error or deviation greater than 30 cm
● Pose accuracy: RMS pose error is less than 30 cm
</td></tr>
</table>

Test 7: Navigation Subsystem Stack Test

Test Name	
ROS Navigation Stack Test	
Objective	
Verify trajectory generation and tracking, localization within built map, ability to successfully visit specified waypoints	
Equipment	Khepera IV robot with LRF module, WiFi router for test network, MRSD Desktop #4, VICON cameras, computer connected to VICON system
Elements	Localization, mapping, navigation
Personnel	Sudhansh
Location	NSH 1604
Procedure	
1. Set up test area to have visually recognizable landmarks and obstacles. 2. Place powered-on Khepera with LRF module in Vicon area. 3. Run the agent bringup node to set up communications with the robot. 4. Run navigation stack, which includes localization, robot controller, trajectory generation. 5. Send a series of waypoints to the robot. 6. Observe the robot as it navigates to the waypoint. 7. When the waypoint is reached, the robot should navigate to the next one. 8. Compare ground truth robot trajectory with generated trajectory.	
Validation / Results	
● Evaluate RMS error in robot pose at waypoints ● Percent of waypoints robot is able to successfully reach ● Observe consistent trajectory tracking	

Task 8: Agent-Server Communication Unit Test

<table>
<tr><th colspan="2">Test Name</th></tr>
<tr><td colspan="2">Agent-Server Communication test</td></tr>
<tr><th colspan="2">Objective</th></tr>
<tr><td colspan="2">Ensure lossless communication between agents and server</td></tr>
<tr><td>Equipment</td><td>Server system, ROS packages on the server, communication bus, UDP and WiFi, Khepera</td></tr>
<tr><td>Elements</td><td>Fleet Management System</td></tr>
<tr><td>Personnel</td><td>1 Person</td></tr>
<tr><td>Location</td><td>NSH 1604</td></tr>
<tr><th colspan="2">Procedure</th></tr>
<tr><td colspan="2">Procedure:
1. Connect Khepera robots and Server to the same network
2. Connect 1 agent to the server
3. Plot the data from the agent and check for missing/inconsistent data.
4. Save data from server to a file in agent and check for inconsistency
5. Repeat steps 2 and 3 to rectify communication problems
6. Connect multiple agents and repeat steps 2,3, 4 and 5.

Performance metrics:
a. Number of agents at a time
b. Feedback speed - Sensor data from agent to server
c. Control speed - Velocity commands from server to agent</td></tr>
<tr><th colspan="2">Validation / Results</th></tr>
<tr><td colspan="2">1. Lossless data transfer from agents to server
2. Lossless data transfer from server to agents</td></tr>
</table>

Task 9: Pure Pursuit Controller Test

Test Name	
Pure pursuit controller test	
Objective	
Ensure that the robot is able to track the desired trajectory	
Equipment	Server, Khepera Robots, WiFi
Elements	Tracking controller
Personnel	1 Person
Location	NSH 1604
Procedure	
Procedure: 1. Prepare a trajectory that has to be followed. 2. Create the path on the test area with a tape 3. Place the robot at the start location 4. Give the command to start the motion 5. Observe the robot as it follows the trajectory 6. Plot the errors for multiple robots and tune the hyperparameters for pure pursuit controller	
Validation / Results	
The robot should lie within the tape limits for the entire duration for 3 repetitions	

Task 10: Task Allocation Unit Test

Test Name	
Regions of Interest Allocator Test	
Objective	
Given multiple regions of interest, the task allocator must assign each agent a feasible set of regions to explore, while minimizing the total distance traveled by the multi-agent system.	
Equipment	MRSD Desktop 2, Personal laptops
Elements	Task Allocation
Personnel	Sandhya Vidyashankar
Location	NSH 1604
Procedure	
Procedure: 1. Take in a grid map of the environment with regions of interest and initial robot poses. 2. Introduce obstacles to the map. 3. Assign costs for each robot to reach each ROI. 4. Assign robots to ROIs using the task allocator, minimizing over the cost.	
Validation / Results	
1. Verify quantitatively and qualitatively that 100% of the given ROIs are covered. 2. If number of ROIs >= number of robots, ensure all robots have tasks. 3. Minimize idle time for each agent	

Task 11: Gap Crossing Task Allocator Test

<table>
<tr><th colspan="2">Test Name</th></tr>
<tr><td colspan="2">Gap Crossing Task Allocator Test</td></tr>
<tr><th colspan="2">Objective</th></tr>
<tr><td colspan="2">The task allocator needs to be able to assign agents for gap crossing. In this case, we assume that the robot cannot go around the gap. We need to find the optimal point of crossing based on agent poses after Test 1. We need to assign the optimal subset of agents for gap crossing such that idle time is minimized.</td></tr>
<tr><td>Equipment</td><td>MRSD Desktop 2, Personal laptops</td></tr>
<tr><td>Elements</td><td>Task Allocation</td></tr>
<tr><td>Personnel</td><td>Sandhya Vidyashankar</td></tr>
<tr><td>Location</td><td>NSH 1604</td></tr>
<tr><th colspan="2">Procedure</th></tr>
<tr><td colspan="2">Procedure:
4. Take in map with obstacles, gap location, and task set from Test 1
5. Using predicted end locations of each agent, search along the gap to find the closest possible point of crossing.
6. Assign a subset of 3 agents to the closest point of crossing.</td></tr>
<tr><th colspan="2">Validation / Results</th></tr>
<tr><td colspan="2">1. Minimize idle time for each agent
2. Minimize total distance for gap crossing (ensure optimality of crossing point, to the extent possible)
3. Test for different numbers of agents (2-5)</td></tr>
</table>

Task 12: Task Allocation Test with Kheperas

Test Name	
Task Allocation Hardware Test	
Objective	
The Task Allocator requires an occupancy grid with obstacle coordinates and robot poses. This test ensures that these data reach the system in a readable format. It also ensures that the task allocator is able to handle any real-world edge cases that may arise.	
Equipment	MRSD Computer System, VyCon, Khepera
Elements	Task Allocation
Personnel	Sandhya Vidyashankar, Sudhansh Yelishetty
Location	NSH 1604
Procedure	
Procedure: 1. Create a static map and occupancy grid files using VyCon 2. Compute centroid of random Regions of Interest to assign POIs 3. Verify localization data and assign robots to goal locations. 4. Run the planning stack on the system	
Validation / Results	
1. Task Allocation is able to assign all POIs 2. All POIs fall within ‘safe’ regions (not in gap/ obstacles) 3. Obstacles are taken into account.	

Task 13: CBS Planner Unit Test

Test Name	
CBS Planner Simulation test	
Objective	
For the multi-agent path finding problem, we are writing a Conflict Based Search (CBS) program from scratch. We need to test if the solver works in simulation by providing discrete grids with user-defined obstacles and checking completeness and speed of the solver	
Equipment	Personal laptops, MRSD Computer
Elements	Planning, Simulation
Personnel	Dhanvi, Harishankar
Location	NSH 1604
Procedure	
Procedure: 1. Create static 8-connected graph with randomly placed obstacles for test cases 2. Randomize start and goal locations for increasing number of agents (from 2 to 5) 3. Write the planned paths to a text file and visualize results using Stage 4. Run this test at least 10 times with random initialization to catch edge behavior	
Validation / Results	
1. Able to produce either collision free paths or return infeasibility 100% of the time 2. Data is read correctly by Stage and visualization is as expected 3. No non-terminating conditions during the test	

Task 14: Planner Test on Kheperas and Central Server

Test Name	
Planner hardware test for single and multiple agents	
Objective	
To test our CBS based planner, we need to create a ROS package, interface it with nav_msgs and OccupancyGrid and write commands to the Khepera. This test validates whether the node gets information and correctly writes commands to the FMS	
Equipment	Khepera, MRSD Computer as Server, Vycon
Elements	Planning, Communications
Personnel	Dhanvi, Harishankar, Sudhansh
Location	NSH 1604
Procedure	
Procedure: 1. Create static map and occupancy grid files for the testbed using VyCon 2. Choose 5 random goal locations on one side and plan paths for a single agent 3. Send commands to the Khepera through the FMS 4. Introduce a second agent and set up 5 pairs of goal locations 5. Create and send non-conflicting paths using CBS node and execute on Khepera	
Validation / Results	
1. Obtained OccupancyGrid is accurate and represents current map state 2. Planner is able to correctly solve for paths 100% of the time with zero collisions 3. Planner commands are executed within 5 cms by the Kheperas at least 75% of the time	

Task 15: Integrating Planner with Task Allocation

Test Name	
Integrating CBS Node and Task Allocation in Simulation	
Objective	
In the final system, the task allocator will dynamically call upon the planning block in intervals and reassign agents depending on priorities of tasks. We need to integrate the two and optimize to allow for concurrent movement, replanning and adapting.	
Equipment	MRSD Computer as Server
Elements	Planning, Communications
Personnel	Dhanvi, Sandhya, Harishankar
Location	NSH 1604
Procedure	
Procedure: 1. Map test environment using LIDAR beforehand and process to get .pgm and yaml files 2. Test two cases - excess agents and excess PoIs. For excess agents, use 4 agents and 2 PoIs, while for the latter use 4 agents and 8 PoIs 3. Start the system and wait for agents to finish tasks at PoI (1 min wait) and assemble at determined point of coupling	
Validation / Results	
1. Optimally assign multiple goal locations to a single agent > 75% of the time 2. Necessary number of agents reach the point of coupling under 15 mins 3. Zero collisions between agents during the task phase	

Test 16: Final Spring Validation Demonstration

Test 3	
Spring Validation Demo	
Objective	
Demonstrate the working of the entire system on only one side of a gap	
Equipment	Khepera IV robot with LRF module, WiFi router for test network, MRSD Desktop #4, VICON setup
Elements	Localization, mapping, task allocation, planning, mechanism
Personnel	The entire team
Location	NSH 1604
Procedure	
The final demo will happen in 2 phases: 1. Phase 1: Coverage of all PoIs in the region a. Place all the powered-on Khepera IV robots with the LRF. b. Connect all of them to the server. c. Input a map of the region and mark points of interest (PoIs). These PoIs will all lie on one side of the gap only. d. Allocate these PoIs to each agent and plan collision-free paths till a determined region of coupling. e. Observe the system in motion. 2. Phase 2: Coupling Test a. Manually align the robots in desirable poses for coupling. b. Initiate the coupling mechanism c. Drive coupled system as a unit across a gap	
Validation / Results	
• 100% of all PoIs are covered (Phase 1) • Time of coverage should be < 15 minutes (Phase 1) • The three robots should move together across the gap (Phase 2)	

Appendix

Our system-level requirements were elicited from the use case, discussions with our sponsors, and internal team meetings based on the time and budget constraints to the project timeline. All the requirements are shown in Table 1 to Table 3.

Functional Requirements

Table 1. Functional Requirements

	Requirement
MF1	The system shall localize agents in a given map.
MF2	The system shall route agents and avoid collisions.
MF3	The system shall determine feasible gaps.
MF4	The system shall determine and achieve coupled configurations.
MF5	The system shall cross gaps.
MF6	The system shall reach given regions of interest.

Performance Requirements

Table 2. Performance Requirements

	Requirement
MP1	The system will cross gaps up to 1.5 agent lengths.
MP2	The system will have 0 unplanned collisions between agents.
MP3	The system will achieve formations with at least 3 robots.
MP4	The system will cross feasible gaps 75% of the time.
MP5	The system will reach all POIs 75% of the time.
MP6	The coupling mechanism will bear the weight of one agent.
MP7	The coupling mechanism will self-align against heading errors of 5 degrees and position error of 10 mm.

Non-Functional Requirements

Table 3. Non-Functional Requirements

	Requirement
MN1	The weight of an agent shall be minimal.
MN2	The coupling mechanism shall consume a low amount of energy.
MN3	The system shall be scalable.
MN4	The system shall be easily maintainable.
MN5	The team shall maximize learning and fun throughout the project through flexible work plans across subsystems.