Team I : Orbiters

Spring Project Test Plan

Team:

Jinaki Qiu
Gaurav Sethia
Roman Kaufman
Siddhatha Ghodasara
Robert Kim
Tanmay Agarwal

Submitted:

February 13, 2024

Table of Contents

Introduction

This document presents a comprehensive overview of the testing protocols and methodologies for the ORBOT project. The testing strategy is structured to commence with the evaluation of individual components and subsystems, progressively escalating to more complex integrations. This incremental approach is designed to ensure thorough verification at each stage, culminating in a fully functional robot capable of efficiently managing OR inventory and autonomously delivering medical items. The outlined schedule delineates the specific functionalities targeted in each test phase, detailing their contribution towards fulfilling the overarching system requirements. The culmination of these efforts will be showcased in our Spring Validation Demonstration, where the robot's advanced capabilities in navigating hospital environments, tracking inventory, and manipulating objects on supply shelves will be demonstrated, highlighting its potential to revolutionize OR logistics.

Logistics

Personnel

No additional personnel are required for testing.

Equipment

The team will utilize a Fetch mobile manipulator to demonstrate the developed software stack.

Location

Because the Fetch Robot is provided by AI Makerspace, all the testing will be performed at AI Makerspace. Below shows a region (colored in red) within AI Makerspace that will be used to test the functional requirements.

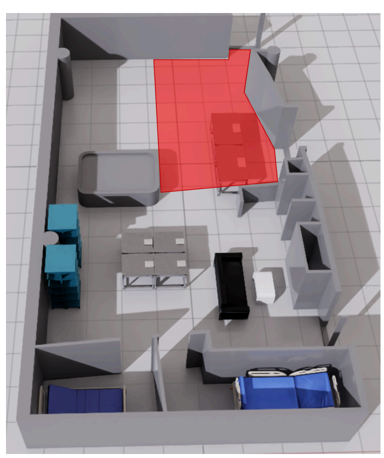

FIGURE 1: 3D rendering of AI Makerspace; Fetch will be tested only within the red region

Schedule

The following table outlines the key milestones and capabilities we aim to review and test on specific dates, ensuring our project aligns with the planned requirements.

Date	Identifier	Capabilities	Test	Requirements
02/15/2024	Progress Review 1	- Simulator is functional with ROS2 interface configured for virtual Fetch - Custom ROS2 nodes can publish/subscribe to simulator interface	1	M.N.4
02/29/2024	Progress Review 2	- Onboard computer can subscribe/publish to Fetch via ROS2 - Vending-shelf can receive commands from onboard computer and dispense item(s) accordingly	2, 3	M.F.6
03/21/2024	Progress Review 3	- Fetch can localize in virtual OR and testbed using built-in sensors within an error of 5cm - Fetch can plan and move to a target location using odometer and Lidar reading - Fetch can plan and execute manipulator trajectories to a target pose	4, 5	M.F.2
04/04/2024	Progress Review 4	- GUI can take target item to retrieve as an input - Fetch can plan and execute path to the target item upon receiving the GUI command - Fetch can plan and execute manipulator trajectory to a position on the vending-shelf upon arriving at the target location - Fetch can command the vending-shelf to dispense an item	6, 7	M.F.1, M.F.4, M.N.3
04/18/2024	Progress Review 5 (SVD)	- The integrated system can provide the total number of items stored in the inventory - The integrated system can plan a path/trajectory to its target position/pose within 5 seconds - The integrated system can collect objects in a bin attached to its end-effector with 80% success rate - GUI can alert the user of malfunctions/errors	8	M.F.3, M.F.5, M.F.6, M.F.7, M.N.1, M.N.2, M.N.5, M.N.6

Tests

Below are detailed tables for each test, specifying objectives, elements involved, locations, equipment, personnel, procedures to be followed, and verification criteria for the successful validation of our project's functionalities.

Test No. 1: Simulator with Fetch Robot + Virtual OR Environment	
Objective	Validate simulator functionality
Elements	Simulation Subsystem
Location	Isaac Sim (Virtual)
Equipment	PC, Monitor
Personnel	Gaurav Sethia
Procedure	
1. Run Isaac Sim with ROS2 interface enabled 2. Open virtual OR environment + Fetch Robot 3. Spawn example medical supply items within simulator at a desired location 4. Run Teleop Node (ROS2) to move Fetch within simulator 5. Acquire and visualize data from Fetch robot's 2D lidar and camera	
Verification Criteria	
1. Command from ROS2 is read by Isaac Sim 2. Simulator is set up with virtual OR environment with medical supplies and Fetch 3. Sensor data from simulated Fetch are read by ROS2 nodes	

Test No. 2: Jetson Xavier Integration Test with Robot	
Objective	Validate onboard compute functionality
Elements	Onboard Computer Subsystem
Location	AI MakerSpace
Equipment	Jetson Xavier, Fetch Robot, Monitor
Personnel	Tanmay Agarwal
Procedure	
1. Run Teleop Node from Xavier's ROS2 Docker	

2. Acquire and visualize Fetch sensor data using ROS2 nodes
Verification Criteria
1. Command from containerized ROS2 is read by Fetch (ROS1) 2. Containerized ROS2 can read sensor data from Fetch (ROS1)

Test No. 3: Vending Machine Integration Test	
Objective	Validate automated vending-shelf functionality
Elements	Vending-Shelf Subsystem
Location	AI MakerSpace
Equipment	Custom-built vending machine module, Jetson Xavier
Personnel	Robert Kim
Procedure	
1. Send commands from Xavier to vending-shelf using ROS2 2. Vending-shelf dispenses one item per request via Wifi	
Verification Criteria	
1. Command from ROS2 is read by vending-shelf 2. Vending-shelf dispenses items upon receiving command	

Test No 4: Navigation Test PR 3	
Objective	Validate navigation functionality
Elements	Localization and Path-Planning Subsystems
Location	Isaac Sim (Virtual), AI Makerspace
Equipment	PC, Monitor, Fetch, Xavier
Personnel	Jinkai Qiu
Procedure	
1. Localization command is sent to Fetch to identify its starting position 2. Input the desired position via ROS2 3. Execute path-planning algorithm and monitor Fetch as it navigates	

Verification Criteria
1. Fetch localizes itself within a virtual and testbed environment 2. Fetch receives command on desired position 3. Fetch plans and executes a path from its original location to desired goal position

Test No. 5: Manipulation Test PR 4	
Objective	Validate manipulation functionality
Elements	Manipulation Subsystems
Location	Isaac Sim (Virtual), AI Makerspace
Equipment	PC, Monitor, Fetch, Xavier
Personnel	Siddharth Ghodasara
Procedure	
1. Input the desired manipulator pose to Fetch via ROS2 2. Plan and execute manipulator trajectory	
Verification Criteria	
1. Fetch receives command on desired manipulator pose 2. Fetch plans the manipulation trajectory from its location to desired position	

Test No. 6: Graphical User Interface Tests PR 5	
Objective	Validate user interface functionality
Elements	User Interface Subsystem
Location	Isaac Sim (Virtual), AI Makerspace
Equipment	PC, Monitor
Personnel	Roman Kaufman
Procedure	
1. Fetch status is sent to GUI via Wifi 2. GUI send navigation command to Fetch 3. GUI send manipulation command to Fetch	

Verification Criteria
1. GUI accurately displays the status of Fetch in real-time 2. Fetch executes navigation to a desired position specified by GUI command 3. Fetch executes manipulation to a desired pose specified by GUI command

Test No. 7: Subsystem Integration Tests PR 5	
Objective	Validate integration of subsystems
Elements	Manipulation, Navigation, Vending-Shelf Subsystems
Location	Isaac Sim (Virtual)
Equipment	PC, Monitor
Personnel	Roman Kaufman
Procedure	
1. Configure Isaac Sim with Fetch integrated with GUI, manipulation, and navigation subsystems 2. Command Fetch to navigate to shelf, identify an item, and perform manipulation 3. Monitor time it takes for Fetch to complete one cycle	
Verification Criteria	
1. Execute end-to-end tasks involving GUI, navigation, and manipulation subsystems.	

Test No. 8: Spring Validation Demo	
Objective	System Integration and Validation Test for SVD
Elements	Onboard Computer, Manipulation, Vending-shelf, Navigation, GUI Subsystems
Location	AI Makerspace
Equipment	Jetson Xavier, Fetch, 3 different medical items, PC, vending-shelf
Personnel	Team ORbiters
Procedure	
1. Connect Jetson Xavier to Fetch via Ethernet 2. Connect external PC to Jetson Xavier and set up GUI	

3. Set up communication with vending-shelf
4. Check GUI for inventory counts
5. Press a button within the GUI for delivery of an item
6. Execute navigation to where the item is located
7. Execute manipulation to move end-effector to target item’s vending module
8. Command and dispense item from vending-shelf to drop on end-effector
9. Execute manipulation to reset manipulator pose
10. Execute navigation to a designated drop-off station
11. Repeat 1~10 for two other items

Verification Criteria

1. The system provides inventory information
2. The system delivers a medical item within 5 minutes
3. The system delivers a different item after completing the previous task
4. The system alerts the user via GUI if it failed its task or malfunctions/errors occur

Appendix: Requirements

Table 1: Mandatory Functional and Performance Requirements

Functional Requirement	Performance Requirement
M.F.1 The system shall receive inputs from the users	M.P.1 The system will receive inputs from the user in less than 5 seconds
M.F.2 The system shall localize in the pre-mapped environment	M.P.2 The system will localize in the environment within an error of 5cm
M.F.3 The system shall collect medical supplies	M.P.3 The system will collect objects with success rate of more than 80% accuracy
M.F.4 The system shall plan and navigate to its destination	M.P.4 The system will plan a global path in 5 seconds and navigate to its destination within maximum speed of 0.8m/s
M.F.5 The system shall deliver medical supplies to operating room	M.P.5 The system will deliver supply to operating room within 5 minutes
M.F.6 The system shall inspect supply inventory to estimate the quantity of each medical supply	M.P.6 The system will inspect 10 supplies with accuracy of 2 item counts for LOW segment
M.F.7 The system shall update the inventory	M.P.7 Within a 15 minute sweep of the supply the system will have inventory knowledge in keeping with M.P.6 above

Table 2: Mandatory Non-Functional Requirements

Non-Functional Requirements
M.N.1 The system will perform rapid retrieval of desired objects
M.N.2 The system will adhere to all relevant standards pertaining to medical robotic systems
M.N.3 The system will have a modular design: for hardware, software, and all things in between
M.N.4 The system will be aesthetic
M.N.5 The system will detect malfunctions and errors so as to notify user
M.N.6 The system will remain available in times of need