Carnegie Mellon University

Robotics Capstone Project

Test Plan

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. David Wettergreen

Version 1.0
November 21, 2016

Contents

List of Figures

1 Design Verification

1.1 Writing Implement

1.1.1 Performance Tests

able to load/move/replace tool within req time limit (10s) metric = duration

test quality of mark made by writing implement metric TBD, could be thickness relative to person using tool(?)

1.1.2 Functional Tests

bool - able to load/move/replace tool metric = yes/no

bool - can motors be actuated to move up/down, tests the motors metric = yes/no

bool - can writing implement system be controlled, tests controllers metric = yes/no

bool - does writing implement make mark when pushed down metric = yes/no

bool - can locomote and draw with tool simultaneously need to make sure it doesn't get stuck speed threshold? aka how fast before poor drawing quality/tool breaks, etc. metric = yes/no

bool - can force sensor measure force being applied well enough to distinguish between states (states = optimal, underactuated, overactuated)

1.1.3 Failure Modes

NJ: Fail: out of ink NJ: Fail: mechanism failure - unable to raise/lower

1.2 Locomotion

1.2.1 Performance Tests

test of accuracy of positional/rotational accuracy acceptance - based on requirements spec of pos/rot accuracy metric - positional/rotational accuracy in distance units

1.2.2 Functional Tests

bool - can the robot achieve a desired speed

1.2.3 Failure Modes

NJ: Robot unable to move accurately (both position and orientation) NJ: Robot unable to move omnidirectionally

1.3 Localization

1.3.1 Performance Tests

positional accuracy of each robot acceptance - requirements spec for localization accuracy metric - accuracy distance

accuracy of detecting bounds of workspace acceptance - how close to actual drawing space the determined bounds are metric - accuracy distance

1.3.2 Functional Tests

bool - can localization find robot position

bool - can localization detect bounds/workspace

1.3.3 Failure Modes

NJ: Fail: camera failure (due insufficient power, bad mounting aka falls) NJ: Fail: unusable localization NJ: Fail: object/person on surface, obscures vision tags causes bad localization

1.4 Image Processing

1.4.1 Performance Tests

how closely does image processor output resemble original image metric - image distance metrics

1.4.2 Functional Tests

can the image processor return data usable by the planner (series of lines)
does the image processor reject improper input
does the image processor keep lines to be drawn within bounds

1.4.3 Failure Modes

NJ: Fail: unable to process user input image

1.5 Work Scheduling, Distribution and Planning

1.5.1 Performance Tests

how consistent is the planner at determining executable plans acceptance criteria: plans do not involve robot collision or out of bounds

parallelism test - how efficiently is execution time distributed. Execution time is the total time robots spend moving to complete the drawing metric: min[exec(r0), exec(r1)] / max[exec(r0), exec(r1)]

parallelism test - how efficiently is draw time distributed. Draw time is how long individual robots spend drawing on the ground vs. moving around metric: min[draw(r0), draw(r1)] / max[draw(r0), draw(r1)]

how efficient are the two robots at working together. metric = (time, 2 robots) / (time, 1 robot) acceptance criteria - optimal speedup of 2x from requirements spec

1.5.2 Functional Tests

are plans generated collision free

autonomy test - does the system require no user input beyond adding the image to be drawn (exception being error handling)

1.5.3 Failure Modes

NJ: Fail: unable to generate plan

1.6 Communication

1.6.1 Performance Test: Uptime

Test Question: What is the uptime on our ability to communicate data to between the robots and the offboard system?
Operational Procedure: Run the system for a significant period of time (several hours) and record any communication downtime or data loss during communication.
Metric: Time duration of down communication and packet loss.
Acceptance Criteria: Operational 95% of the time.
Requirement(s) Verified: **RH: Communication one**

1.6.2 Functional Test: Sending and Receiving Data

Test Question: Can the robot send and receive data from the off-board device and can the off-board device send and receive data to the robot?
Operational Procedure: Send data from the off-board device to the robot and verify the robot received it. Send data from the robot to the off-board device and verify the off-board device received it.
Metric: Four booleans on whether the data is successfully sent and recieved on both ends.
Acceptance Criteria: We must succeed on all four accounts.
Requirement(s) Verified: **RH: communication**

1.6.3 Functional Test: Data Parsing

Test Question: Can the data on each side (robot, off-board device) be parsed by each other?
Operational Procedure: Send data from the off-board device to the robot and verify the robot received it and can execute it. Send data from the robot to the off-board device and verify the off-board device received it and can respond to it.
Metric: Check whether the data was successfully parsed on all sides.
Acceptance Criteria: We require all data be parsable.
Requirement(s) Verified: **RH: communication**
is data sent using the same communication protocol/can be parsed by both ends of the system

1.6.4 Failure Mode: Loss of Connection

NJ: loss of connection - server side, robot/client side, no connection vs. intermittent are equivalent in handling

1.6.5 Failure Mode: Incorrect Data

NJ: Send garbage/incorect data

1.7 User Interface

1.7.1 Performance Test: Emergency Stop Speed

Test Question: How fast does the emergency stop shut down the system?
Operational Procedure: While the system is in use, press the emergency stop button and time how long it takes for everything to completely shut down.
Metric: Elapsed time.
Acceptance Criteria: It is vital to safety that our emergency stop shuts everything down within a second.
Requirement(s) Verified: **RH: emergency stop one and safety**

1.7.2 Performance Test: Error Reporting Delay

Test Question: What is the delay between an error occurring and that error being reported to the user?
Operational Procedure: Given a list of known operational errors, intentionally trigger each error within the system and report the time between causing the error and it being reported to the user.
Metric: Averaged elapsed time across error reporting.
Acceptance Criteria: The average time to detect and report an error should be within 3 seconds.
Requirement(s) Verified: **RH: safety and error reporting**

1.7.3 Performance Test: Error Understandability

Test Question: How understandable and informative or are the user errors?
Operational Procedure: Given a list of known operational errors, intentionally trigger each error while a non-developer user is using the system (while masking the error cause) and evaluate how well the user can determine the error. For example, while the system is drawing the user could be in a different room with only the error reporting device, making the user unable to see what errors the robots are facing.
Metric: Determine if the user can determine the error and knows how to react to or correct the error.
Acceptance Criteria: The user should be able to determine and react effectively for 90% of the errors.
Requirement(s) Verified: **RH: error handling and ease of user use**

1.7.4 Functional Test: Emergency Stop

Test Question: Does the emergency stop fully stop the system?
Operational Procedure: While the system is in use, press the emergency stop button and check if all systems halt their operation.
Metric: Boolean on whether every subsystem stops or not.

Acceptance Criteria: It is only successful if the boolean metric is true.
Requirement(s) Verified: **RH: emergency stop one**

1.7.5 Functional Test: Error Reporting

Test Question: Is each operational error reported to the user?
Operational Procedure: Given a list of known operational errors, intentionally trigger each error within the system and report whether the error caused it reported to the user.
Metric: Each error must be reported correctly. Hence we can divide the number of correctly reported errors by the number of total errors caused to determine an error-reporting score.
Acceptance Criteria: Considering error handling is critical to performance, our system should have an error-reporting score of 90%.
Requirement(s) Verified: **RH: error handling one**

1.7.6 Failure Mode: UI Navigation

NJ: Fail: user unable to navigate UI

1.8 Power System

1.8.1 Performance Test: Battery Duration

Test Question: How long can an individual robot run for on a single battery charge?
Operational Procedure: Charge a robot fully. Given some example drawing inputs, continue to input drawings until the robot is fully drained of power. Time how long this takes.
Metric: The duration of operational time given one charge
Acceptance Criteria: We accept this if the operational time exceeds the necessary duration time of 90% of our test drawing inputs.
Requirement(s) Verified: **RH: battery life one**

1.8.2 Functional Test: Battery Life

Test Question: Can the robots complete a drawing from a single charge?
Operational Procedure: Given a set of example drawing inputs, we want to test the robots ability. For each input, fully charge each robot, send the input and keep track of whether the drawing is fully complete before the battery on either robot is fully drained.
Metric: The ratio of the number of completed drawings to the total number of drawings.
Acceptance Criteria: We want to be able to successfully draw 90% of the drawings in our example drawing input set.
Requirement(s) Verified: **RH: battery life one**

1.8.3 Failure Mode: Insufficient Battery

NJ: Failure: Insufficient battery - caused by low/no battery power

2 Full System Validation

NJ: Fail: out of bounds - uses localization, locomotion NJ: Fail: robot makes incorrect mark on ground - uses locomotion, localization, writing implement NJ: Fail: collision between robots - uses locomotion, localization NJ: Fail: nonuniform/flat surface - assumptions(?) causes slippage

Test: do the drive commands respect physical robot capabilities. This relates to whether they robot can drive/draw at the same time w/o marker getting stuck. Also, is there a speed threshold at which drawing quality degrades - involves writing implement, locomotion

Test: does an individual robot draw accurately relative to the commanded motion. This involves making sure line aren't jagged/squiggly, etc. come up with better words for this

Test: does the robot fulfill size requirements. CHECK REQ SPEC FOR VALUES

Test: does the robot fulfill weight requirements. CHECK REQ SPEC FOR VALUES

Test: is the robot safe, aka does it not have pointy (sharp) edges on the outside

Test/verification: Was robot system price within budget metric: dollars acceptance criteria: 2500 dollars

3 Risk Management

Battery explores Robots hit intruders Camera fall User's fingers somehow get stuck in rotating wheels when install and uninstall tools or get stuck in motors for lifting chalk

4 Requirements Traceability Matrix