Carnegie Mellon University

Robotics Capstone Project

Performance Validation Demonstration

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. Cameron Riviere

Version 1.0
May 11, 2017

Contents

List of Figures

1 System Overview

This section details individual subsystems and how they function together to form the complete robot system. It is split into descriptions of the mechanical design, electrical components, and software subsystem architecture. System design is described for a single robot, with some software subsystems utilizing information for both robots at once.

1.1 Mechanical System

As shown below in Fig.1, our system contains two identical robots. Each robot is 23 cm in diameter, 19 cm tall, and weights 1.5lb with all electronics installed. Details for mechanical components are explained below. Besides the two robots, mechanical system also contains a camera jig, constructed using 8020 aluminum extrusions.

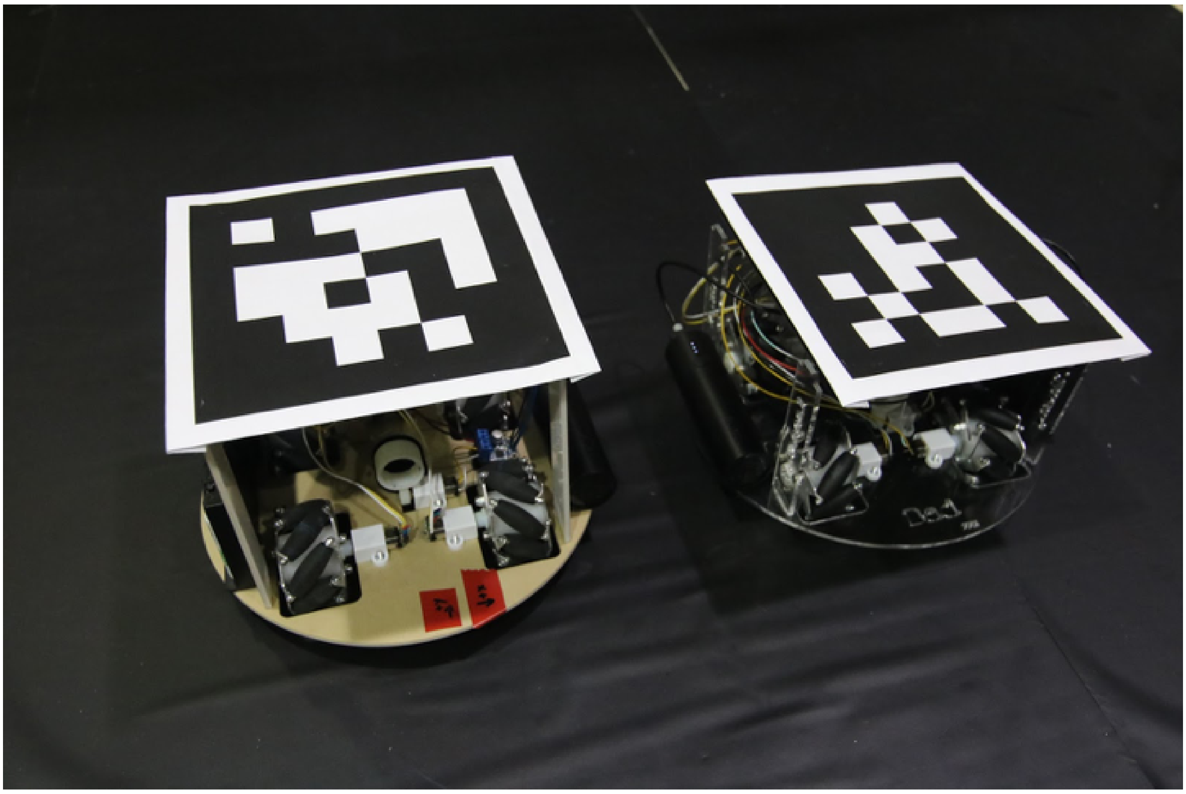

Figure 1: Robots Overview

1.1.1 Locomotion

For locomotion, each robot drives with four Mecanum wheels. Due to the linear nature of the drawing inputs, we expect our robots to make sharp turns quite often. Therefore, Mecanum wheels proved the best locomotion mechanism, for their high mobility and high controllability. As shown in the photo of wheel assembly (Fig.2 left), each Mecanum wheel is powered by a metal micro gearmotor, which is mounted to the chassis via an off-the-shelf motor bracket. Each Mecanum wheel is connected to the motor through a customized wheel adaptor, as shown in Fig.2 right. These wheel adaptors were fabricated using 3D printing so that we could optimize their design iteratively in a short period of time.

1.1.2 Writing Tool

The robot writes using a lever-like writing mechanism. The chalk marker is slipped into a customized marker holder (Fig.3 left) which has a tight D-shaft cutout for connecting the micro gearmotor. As the motor rotates, the marker holder raises or lowers the marker which enables or disables drawing lines. The assembled mechanism in shown below in Fig.3 middle. Similar to the motor adaptors, the customized marker holder is fabricated using 3D printing for rapid prototyping.

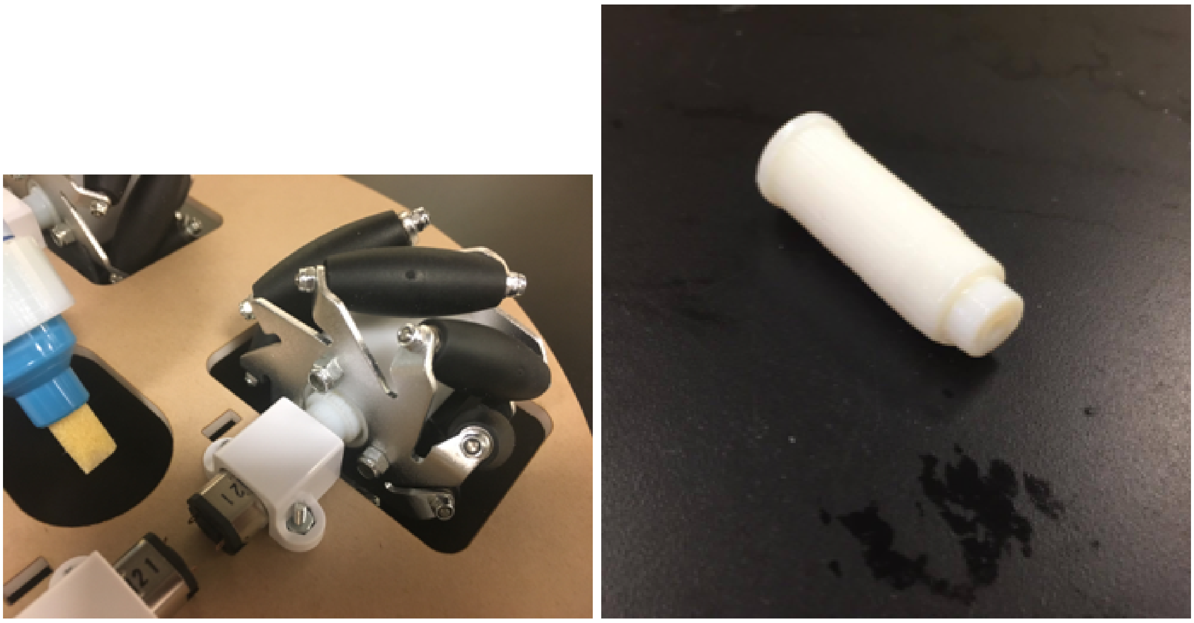

Figure 2: Wheel Assembly (left), Wheel Adaptor (right)

Compared to other drawing utilities like spray paint, wet paint, or regular markers, we used chalk based drawing tool because so we could easily reuse the drawing surface, and the mark has a unique look and texture. We choose a liquid chalk marker instead of regular chalk so we did not have to account for the size of the tool decreasing during consumption.

Fig.3 right shows a screenshot of drawing mechanism working in action.

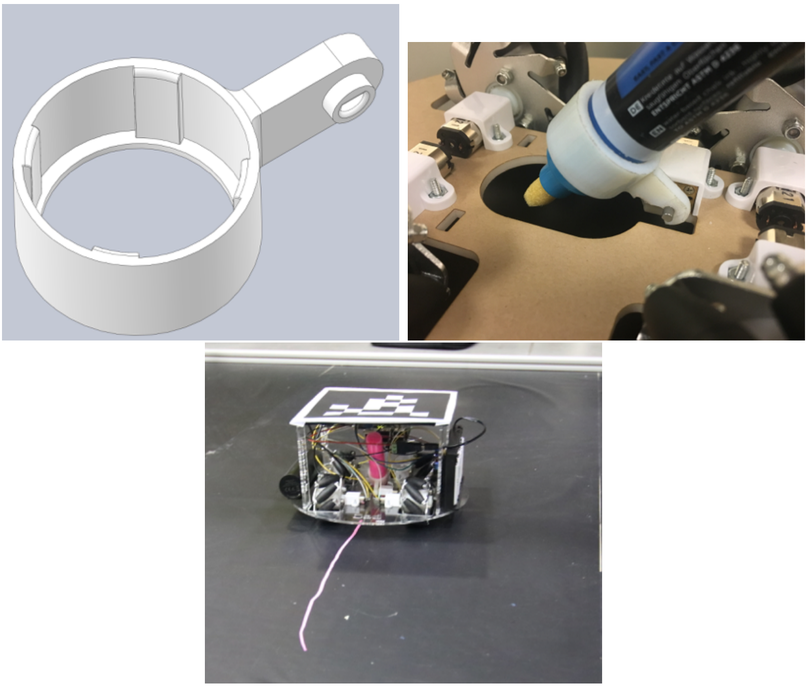

Figure 3: Marker Holder (left), Writing Assembly (middle), Robot Drawing (right)

1.1.3 Chassis

Since we mounted a significant amount of electronics and mechanical components onto the chassis, we choose laser cutting as the fabrication technique to produce the chassis (Fig.4 left). The chassis is laser cut using acrylic because the material has a good combination of durability and machinability.

Supporting structures for Raspberry Pi and AprilTag are also assembled using laser cut acrylic pieces. They are attached to the chassis through slot joints, as shown below in Fig.4 right. Batteries for Raspberry Pi and motors are attached to the chassis using Velcro tape so that we can easily swap them during testing.

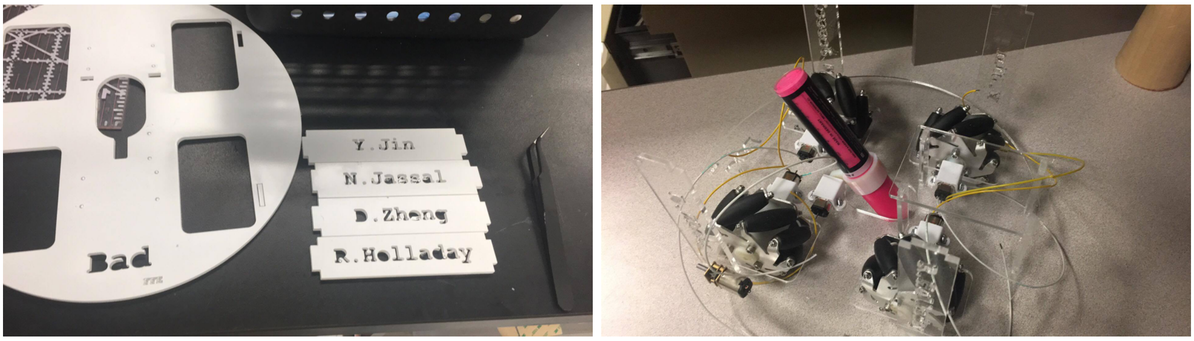

Figure 4: Laser Cut Chassis (left), Chassis Assembly (right)

1.1.4 Infrastructure

As shown in Fig.5, the camera jig is designed and machined using 8020 aluminum extrusion, chosen due to the frames' ease-of-use and durability. The jig has a footprint of 6ft by 6ft and a height of 5ft. 5ft was determined to be the optimal height for capturing all corners of the 6ft by 6ft work space with the overhead camera. The camera was mounted to the jig using Velcro tape. Due to 80/20 structure's easy construction, we were able to easily disassemble the jig into two pieces for transportation. Despite its large size, the camera jig is not restricted to any specific space.

The work space is constructed by laying chalk board paper on sheets of plywood board. Using plywood ensures that the robots always operate on flat surface. Due to the size constraints of available plywood sheets, the work space is assembled using two 3ft by 6ft sheets. These two sheets are then folded for straightforward transportation.

1.2 Electrical System

The electrical systems for both robots are identical. A Raspberry Pi is used as the main controller, with individual motor controllers for each of the four wheels, and the writing implement motor. Power is provided to the Raspberry Pi via power bank, and the motors have separate power provided via a series of batteries. Use of battery power for the Raspberry Pi and motors allows for each robot to be completely autonomous, without the need for external power during operation.

1.3 Software System

The software system is mainly run offboard, where most system processing occurs. The onboard software systems process directional commands to direct the motors for movement and controlling the writing implement.

1.3.1 Controller

The controller takes information from all software subsystems, both onboard and offboard, and combines it to command the system as a whole. The offboard controller takes waypoint input from the planner, as described in Sec. 1.3.6, and combines it with localization data to determine each robot's next action. This includes collision response, and standard locomotion targets. This data is processed and combined into a message sent via the communication subsystem to the onboard system. The onboard system is significantly simpler: it waits for new messages from the offboard system, and upon receiving, processes the target into motor commands which are then directly sent to each motor. Motor commands include wheel locomotion and movement of the writing implement.

At a fixed interval of 20Hz, the offboard controller runs an update loop. This update loop contains all processing, which pulls data from the various subsystems as described above. All blocking subsystems

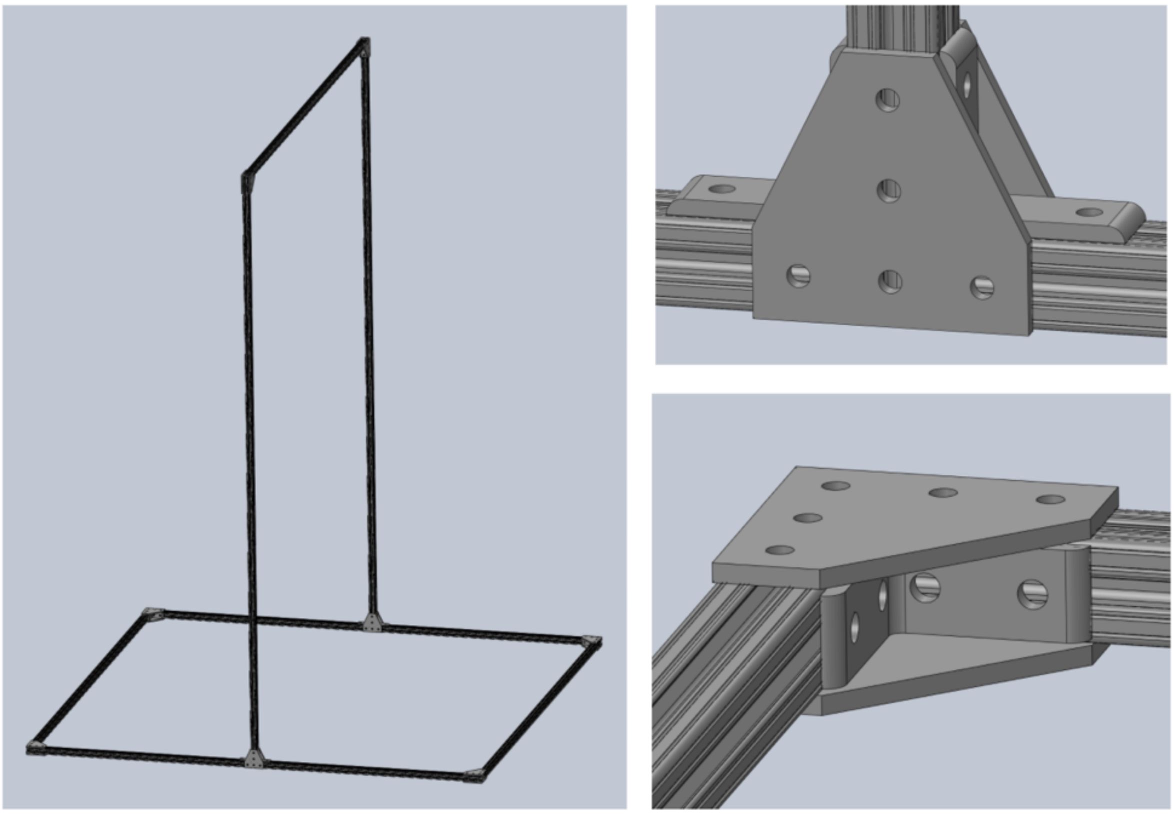

Figure 5: 8020 Camera Jig

and functions are processed in separate threads, to minimize latency between messages sent to the robots, and to maintain a constant update loop speed.

1.3.2 Communication

The communication subsystem has been static in design throughout the project. As input, it receives data consisting of locomotion and onboard processes, and sends them over TCP wireless connection. Locomotion data is simplified for each robot. It contains the robot's current transform (2D position and orientation), and target waypoint transform. It also includes a stop parameter, and the expected status of the writing implement. These parameters are packaged into the Protobuf message format, which is serialized and passed wirelessly. The onboard system takes advantage of the communication subsystem by polling for new messages, which provide updates to the robot's mechanical status. Processing of these messages are done by the onboard controller, and the locomotion subsystem (Sec. 1.3.3).

Messages are processed and sent asynchronously from the offboard controller. The subsystem holds partially completed messages as they are constructed in a single update loop, and then a message thread is activated upon receiving a message-ready status. Each robot has its own message thread, which will send the message while the controller continues on to its next update loop.

1.3.3 Locomotion

Locomotion subsystem processing is mainly done onboard each individual robot. At each onboard update loop, a message with the robot's current transform and target transform is computed. The vector between them is used to compute mecanum wheel motor powers. Translational motion uses the planar 2-dimensional vector, using standard mecanum wheel equations with a calibrated speed. Rotational motion is used as a correction factor, to help the robots maintain a forward heading throughout operation. When the robot's current orientation exceeds a threshold over the desired forward heading, a slow rotational factor is accumulated to the mecanum wheel powers to correct heading. This correction is purposefully slow, to prevent it from overshadowing translational motion, which could result in curved motions when straight lines are preferred.

1.3.4 Localization

Robot localization is done via the offboard system, which parses data from the overhead webcam to track Apriltags. Apriltag detection is done for the four corners of the drawing space, and for the robots, via a C++ Apriltag library. Data is passed up to the main Python control layer using Boost Python, and returned to the main controller. Apriltag data is received and oriented back from the webcam's pixel coordinate frame to a coordinate space relative to the corner Apriltags. This coordinate space transformation is necessary to ensure the robots can continually orient themselves with a forward heading.

Similar to other subsystems, localization is run in a separate thread from the main controller. When the controller calls for localization data, the most recently processed robot transform data is returned. Localization runs at a consistent 30Hz.

1.3.5 Writing Tool

Writing tool control is done via the onboard systems. The offboard controller passes an enable or disable status to the onboard system, which then controls the tool. To ensure the tool maintains a consistent force on the ground, it will send a pulsing signal to the motor when enabled. This ensures that the tool is always on the ground, even when small bumps or uneven terrain pushes the tool upwards during operation.

1.3.6 Scheduling, Distribution, and Planning (SDP)

The first step of our planner is to parse the inputted lines. Given our set of lines we want to efficiently divide the work between the two robots. The locomotion system takes as input a set of lines with a flag on whether or not we are drawing. Additionally, collision is handled on-line, so we only need to split the lines and perform no timing-based trajectory planning.

We distribute lines in a greedy manner. For each robot we maintain a cost, which corresponds to how much work the robot has done. This cost is initialized to zero, and start planning with either robot. For the this robot we pick the nearest line to its current location. We then assign that line to the robot and increment the cost to include the cost of driving to that line and the drawing that line. We update the drawing flags accordingly, and repeat until all lines have been assigned.

2 Risk Management

In this section, we revisit the risks defined by our previous document. The risk tables have been updated with the actions we've taken to address the risks.

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
1	Defective Parts	Don	No issues Persistent: Only ordering reliable parts

Description:
Parts that we ordered arrived defective or do not perform to specifications

Consequences:	Risk Type:
We need to reorder parts, expending extra time and budget	Parts
Risk Reduction Plan:	**Expected Outcome**
We will order only parts that have been extensively reviewed, or we have experience with, and order extra parts	We will be able to properly deal with any parts that break during the development process

	Consequence					
	1	2	3	4	5	
						5
						4
						3
		X				2
Likelihood						1

Figure 6: Risk 1: Defective Parts

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
2	Unavailable Group Member	All	4/27/17: Neil is sick and unable to work on the robot for the week, we reschedule work items and meeting times to stay on schedule Persistent: Keeping group members up to date

Description:	
A group member becomes unavailable for work due to travel, sickness, or other emergencies	

Consequences:	Risk Type:
Work that would have been distributed to that group member needs to be reassigned	Logistical

Risk Reduction Plan:	Expected Outcome
We will ensure that every group member is always on the same page about progress so we don't lose too much progress	If a member becomes unavailable, it will only be for a short time and can be easily dealt with

	Consequence					
	1	2	3	4	5	
						5
						4
				X		3
						2
Likelihood						1

Figure 7: Risk 2: Unavailable Member

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
3	Breaking parts	Eric	3/28/17: Motor shaft broken, reprinted 3/30/17: Motor encoder magnet flawed, needed realignment 4/12/17: Writing implement motor on Blue burned out, used replacement 4/25/17: Robot was ablaze, burned locomotive and writing motor controller on Bad 4/30/17: Writing implement holder broken, printed another Persistent: Careful handling of parts

Description:	
Parts unexpectedly break as a result of accidents or improper use	

Consequences:	Risk Type:
We need to reorder parts, expending extra time and budget	Parts

Risk Reduction Plan:	Expected Outcome
We will practice safe procedures when working with parts and order extras in case	Few parts will break, and even if they do we will have extras on hand

	Consequence					
	1	2	3	4	5	
						5
						4
						3
				X		2
Likelihood						1

Figure 8: Risk 3: Breaking Parts

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
4	Mecanum Drive Too Unstable	Eric	4/1/17: Tested mecanum wheels, no issues present 4/30/17: Difficulting writing and drawing on large bumps in writing surface, solved by lifting writing implement when approacing bumps Persistent: Built extra time into schedule

Description:
The drive mechanism for the robot proves too be too unstable or unreliable for our purposes

Consequences:	Risk Type:
We will need to redesign the drive mechanism, expending considerable time and effort	Design flaw
Risk Reduction Plan:	**Expected Outcome**
We will build enough time in our schedule to deal with it if necessary, and will use suspension	The instability resulting from the wheels will be manageable

	Consequence					
	1	2	3	4	5	
						5
						4
				X		3
						2
Likelihood						1

Figure 9: Risk 4: Mecanum Drive Too Unstable

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
5	Localization not precise enough	Neil	4/8/17: Preliminary test of localization, appears robust enough Persistent: None

Description:
Our localization system is not precise enough to ensure that the drawings are accurate representations of input

Consequences:	Risk Type:
We will need to redesign the localization system or redefine drawing requirements	Design flaw
Risk Reduction Plan:	**Expected Outcome**
We will test the localization system early on in order to catch any design flaws within the system	Localization will work well enough for our purposes

	Consequence					
	1	2	3	4	5	
						5
						4
						3
			X			2
Likelihood						1

Figure 10: Risk 5: Localization not precise enough

Risk ID:	Risk Title:	Risk Owner:	Actions taken:
6	Unexpected Budget Overruns	Rachel	No issues Persistent: We have a significant buffer in our budget

Description:
We unexpectedly run out of budget, because parts cost more than expected or other parties reduce our budget

Consequences:	Risk Type:
We need to scale down our project, or possibly even acquire funds through other means	Logicstical

Risk Reduction Plan:	Expected Outcome
We will leave a significant buffer in our budget in case unexpected situations occur	We will have a large enough buffer that essential components will be acquired

	Consequence					
	1	2	3	4	5	
						5
						4
						3
		X				2
Likelihood						1

Figure 11: Risk 6: Unexpected Budget Overruns

3 Future Work and Commercialization Considerations

We next look toward the future on what factors we would need to consider to develop our capstone into a full product and company.

3.1 Economic Factors

Having a sustainable business model is imperative to successful commercialization of our drawing agent system. The main economic considerations relate to selling the robots and accessories, production, and the main product to be used for profits. We would be looking at producing or selling paint and the holders, spare parts, and the robots themselves. Each of these must then take into account the need to produce them at scale for mass production. We could also consider selling robots of various sizes, for varying purposes.This could include airport runway lines, sports field lines, or more consumer-focused as a toy or tool for advertising on the ground. Our business models could include usage training, renting vs. buying individual units, and maintenance costs.

3.2 Reliability

To ensure our robot system is reliable for commercialization, we have considered a number of changes that would need to be made. Primarily, the system would need to be more robust - this includes the ability to work on surfaces that are not completely flat, robot weatherproofing, and updating durability to use something stronger than wood. Usage changes would allow the system to communicate consistently without latency, and take advantage of motor encoders or other sensors to significantly improve drawing accuracy. Finally, we would need a consistent and standardized methodology to verify the quality of finished drawings.

3.3 Safety

While the robots built for this project are small, updates in size or usability could potentially pose safety risks that must be accounted for. That in mind, a major change we would make to improve safety would be with regard to collision detection. Robot agents should be able to consistently avoid collisions with each other and react accordingly, as well as detect and halt operation in case of near-collision with other obstacles (such as bystanders). The writing tool would also need a more straightforward method for loading. Currently it involves reaching inside the robot, which could pose dangerous for inexperienced or young users. Finally, any use case in a public space would require coordination with local authorities. For airports, this would be air traffic control, for road painting, coordinating with the local transportation departments to work out road closings or detours.

3.4 Maintainability

Commercialization would also require the robot system to be more easily maintainable, as mass production makes it difficult for a small company or group to repair each robot if they constantly break. Simpler updates would be longer battery life, better resource management (this includes both drawing material and battery), and a more robust painting mechanism with easier assembly and setup. Aside from physical changes, maintenance plans could include service plans, geolocation as a feature for users, and straightforward documentation to assist users in system operation.

3.5 Sustainability

Sustainability is a common marketing tool, and this robot system lends itself fairly simply to being a more sustainable product. The drawing material could be updated to use an eco-friendly paint, and the materials and manufacturing process could easily be updated for environmental friendliness. Other considerations could allow the robot agents to use solar power, or have a charging system to avoid use of standard single-use batteries. Finally, systems could be provided to easily recycle the robots when and if they eventually do stop working.

3.6 Ethical Issues

Building a business around these robots comes with a couple of ethical issues, similar to any new product. Liability for accidents is a concern that would have to be dealt with, as well as constructing or designing theft-deterring mechanisms to help users feel more comfortable letting the robots run autonomously without supervision.

3.7 Marketability

Successful marketing can often make or break a new company. To successfully market our robot drawing system, we have a number of options. Advertising could show the system usage, in which a user can easily scan in plans for easy input, and let the robots run autonomously with little setup. At the same time, we could show usage of multiple robots, and provide information on the simple system scalability. The consumer product would most likely use a mobile app for providing input and status updates. For the commercial sector, the focus would be on showing drawing multiple types of lines for our various usages, such as airports, road lines, or sports fields. Finally, we believe that providing a development kit would help improve exposure of our system, allowing classrooms to leverage this robot in educational settings.

3.8 Technical Issues

Building consumer-ready robot agents would require a few major technical changes as well.The largest change would be collision detection - a complex and robust methodology would be needed to ensure safety and consistent operation. The localization system would also need improvement, likely via computer vision so the robots can run regardless of location. Finally, the ability to both specify and switch between colors when drawing would be necessary for a successful product.

Appendices

A Testing and Evaluation Plan

Below we present our verification and validation plan below.

A.1 Design Verification

This section details our test verification plan. This includes the questions and tests we intend to address, how we plan to measure and achieve success with respect to these tests and the result of our test on the final system.

A.2 Writing Implement

A.2.1 Performance Test: Loading

Test Question: Is a human operator able to reload the writing tool within the required time limit of 10s, how long does it take?
Operational Procedure: With a writing tool already installed in the writing assembly, a human test subject will perform 3 reload tests which are separately timed.
Metric: Average duration of the reload time.
Acceptance Criteria: The average reload time is under 10s.
Requirement(s) Verified: NFR14, FR5, FR6, FR7
Result: We were able to reload the writing tool in less than 10 seconds, usually in approximately 5 seconds, varying slightly based off the team member.

A.2.2 Performance Test: Writing Quality

Test Question: Is the drawing produced by the robot of acceptable quality?
Operational Procedure: Using a fully loaded writing tool, the robot attempts to draw along a route with at least 4 ft. of travel distance and 3 turns exceeding 50 degrees. Verify that the resulting drawing is of acceptable quality.
Metric: Percent thickness of the route at its narrowest point compared to the maximum thickness of a line created with the writing tool. Boolean on whether or not there are complete breaks in the line.
Acceptance Criteria: The percent thickness is at least 70%, and there are no complete breaks in the line.
Requirement(s) Verified: NFR6
Result: When our assumptions of flat ground and perfect communication were met, we were produced drawings of acceptable quality.

A.2.3 Functional Test: Simultaneous Driving and Writing

Test Question: Can the writing tool make a mark while driving?
Operational Procedure: With a fully loaded writing implement, the robot will mark a 1 ft. line on the writing surface.
Metric: Whether or not any discernible mark is made and the full distance is covered.
Acceptance Criteria: A discernible mark must be made and the full distance must be travelled without the robot becoming stuck or breaking.
Requirement(s) Verified: NFR6
Result: We were able to drive and draw.

A.2.4 Functional Test: Marking

Test Question: Does the writing tool make a mark when pushed down?
Operational Procedure: The robot will press down on a writing surface with a fully loaded writing implement. Robot must mark with pressure necessary to mark the surface without damaging the surface or writing tool.
Metric: Whether or not a any discernible mark is made.
Acceptance Criteria: A discernible mark must be made.

Requirement(s) Verified: NFR6
Result: A discernible mark is made when the tool is activated.

A.3 Locomotion

A.3.1 Performance Test: Accuracy

Test Question: Is the robot able to drive with positional and rotational accuracy?
Operational Procedure: The robot drives along a predetermined testing route consisting of driving forward at least 3 feet and driving to its left at least 1 feet.
Metric: The difference of the robot's final position and orientation from the intended position and orientation.
Acceptance Criteria: During the forward driving test, the robot's position must be less than 1 inch away from the intended position. During the left driving test, its orientation must be less 10 degrees from the commended perpendicular line. This result must be achieved 90 percent of the time.
Requirement(s) Verified: NFR12, NFR13
Result: When our assumption of perfect communication was met, we met the standard of positional and rotational accuracy.

A.3.2 Functional Test: Speed

Test Question: Can the robot reach a desired speed?
Operational Procedure: The robot will drive along a straight line for 5 ft. during a timed trial.
Metric: The time required for the robot to reach the end of the line.
Acceptance Criteria: The robot must reach the end of the 5 ft. testing course in 20 seconds. This test must be repeatable 90 percent of the time.
Requirement(s) Verified: NFR5
Result: We could achieve the desired speed, when we maintained our assumptions of communication and flat surface. During operation, there were times when these two assumptions were not met, leading to a longer transport time.

A.3.3 Functional Test: Omnidirectional

Test Question: Can the robot drive omnidirectionally?
Operational Procedure: The robot will drive along a rectangular path that is 10 inches by 10 inches.
Metric: Whether or not the robot can complete the course with only linear motion.
Acceptance Criteria: The robot must be able to complete the course with linear motion only. **Requirement(s) Verified:** FR1, FR9
Result: The robots drove omnidirectionally.

A.4 Localization

A.4.1 Performance Test: Robot Position Accuracy

Test Question: Is the localization system able to accurately determine the position of each robot?
Operational Procedure: Both robots sit stationary within the working bounds. The localization system then attempts to determine their locations.
Metric: The difference of the robot's actual position from the position reported by the localization system.
Acceptance Criteria: The reported position must be within 1/10 in. of the actual position.
Requirement(s) Verified: NF6, FR4
Result: We were able to accurate detect each robot.

A.4.2 Performance Test: Bounds Accuracy

Test Question: Is the localization system able to accurately determine the boundaries of the workspace?
Operational Procedure: The localization system attempts to determine the bounds of the workspace.
Metric: The total difference in distance between the reported corners of the workspace and the distance between the actual corners.
Acceptance Criteria: The total difference must not exceed 1 in.

Requirement(s) Verified: FR4
Result: We were able to accurately detect the bounds.

A.4.3 Functional Test: Robot Position

Test Question: Can the localization system find the robot?
Operational Procedure: With a single robot within the working bounds, the localization system attempts to determine the robot's location. Robot operation out of bounds is also considered out of scope, and is undefined behavior.
Metric: Whether or not a location is returned by the localization system.
Acceptance Criteria: The system must return a location for the robot.
Requirement(s) Verified: FR4, NFR6, FR3, NFR13
Result: We were able to localize each robot.

A.4.4 Functional Test: Bounds

Test Question: Can the localization system find the working bounds?
Operational Procedure: The localization system attempts to find all four corners of the working bounds while they are all in its field of view.
Metric: Whether or not locations are returned for all four corners.
Acceptance Criteria: The system must return locations for all four corners of the working bounds.
Requirement(s) Verified: FR4, NFR6, FR3
Result: We received all four corners.

A.5 Work Scheduling, Distribution and Planning

A.5.1 Performance Test: Executable Plans

Test Question: How consistent is the planner at generating executable plans, ie those that avoid collision and stay within bounds?
Operational Procedure: Using the example input set (Appendix C), run each input and check the plan for potential robot-robot collisions and out of bounds driving.
Metric: Ratio of number of unacceptable plans, those that would involve collision or driving out of bounds, over the total number of plans.
Acceptance Criteria: Almost all, 99% of plans would not involve collision or out-of-bounds if executed.
Requirement(s) Verified: FR4, NFR11
Result: All plans respected bounds and collisions were checked on-line.

A.5.2 Performance Test: Execution Distribution

Test Question: How efficiently is execution time, i.e. the total time robots spend moving, distributed?
Operational Procedure: Using the example input set (Appendix C), run each input and record the total time each robot spends moving.
Metric: We define execution efficiency as $\frac{\min(execution(R_0),execution(R_1))}{\max(execution(R_0),execution(R_1))}$ where execution(R_0) refers to the execution time of robot 0 and execution(R_0) refers to the execution time of robot 1
Acceptance Criteria: Execution efficiency of 0.75.
Requirement(s) Verified: FR8, NFR5
Result: We failed to perform objective time tests. However, experience with this system shows success for this test.

A.5.3 Performance Test: Drawing Distribution

Test Question: How efficiently is drawing time, i.e. the total time robots spend drawing, distributed?
Operational Procedure: Using the example input set (Appendix C), run each input and record the total time each robot spends drawing.
Metric: We define drawing efficiency as $\frac{\min(draw(R_0),draw(R_1))}{\max(draw(R_0),draw(R_1))}$ where draw(R_0) refers to the drawing time of robot 0 and draw(R_0) refers to the drawing time of robot 1
Acceptance Criteria: Drawing efficiency of 0.75.
Requirement(s) Verified: FR8, NFR5

Result: We failed to perform objective time tests. However, experience with this system shows success for this test.

A.5.4 Performance Test: Speedup

Test Question: What speedup is achieved by using two robots instead of one?
Operational Procedure: Using the example input set (Appendix C), run each input first with one robot and then with two. Time the execution time of each variant.
Metric: The comparison of duration, i.e. $\frac{executiontimewith2robots}{executiontimewith1robot}$.
Acceptance Criteria: According to our requirements we expect a speedup of 2x.
Requirement(s) Verified: NFR5
Result: We achieved close to the desired speed-up.

A.5.5 Functional Test: Collision Free

Test Question: Does the planner and executor generate collision free plans?
Operational Procedure: Using the example input set (Appendix C), run each input and check for any robot-robot collisions during execution.
Metric: Boolean across each plans on whether a collision occurred.
Acceptance Criteria: We only accept if collisions were avoided on 95% of our test cases.
Requirement(s) Verified: NFR 11
Result: We check for collisions on-line, therefore this is not the responsibility of the planner.

A.5.6 Functional Test: Autonomy

Test Question: Does the system require no user input beyond adding the image to be drawn (except for error handling)?
Operational Procedure: After having input a plan, press "Run" on the system and observe if the system requires user input to finish the drawing.
Metric: Boolean on whether user input was required, excluding input relating to errors.
Acceptance Criteria: Accept only if no input was required.
Requirement(s) Verified: FR 2
Result: Our system is autonomous.

A.6 Communication

A.6.1 Performance Test: Uptime

Test Question: What is the uptime on our ability to communicate data 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: FR12, NFR8, NFR9
Result: We experienced an unexpected amount of latency in our communication, however we could not trace the issue to anything within our system considering the issue occurred seemingly non-deterministically.

A.6.2 Functional Test: Sending and Receiving Data

Test Question: Can the robot send and receive data to and from the offboard device and can the offboard device send and receive data to and from 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: NFR9
Result: We were able to send and recieve on both ends.

A.6.3 Functional Test: Data Parsing

Test Question: Can the data on each side (robot, offboard device) be parsed by each system?
Operational Procedure: Send data from the offboard device to the robot and verify the robot received it and can execute and process the appropriate information. Send data from the robot to the offboard system and verify the offboard device received 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: NFR9
Result: All data was parsable.

A.7 User Interface

A.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: FR13, NFR11
Result: The emergency stop shuts down the system immediately.

A.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: NFR2, NFR11
Result: Errors are reported within our time bound.

A.7.3 Performance Test: Error Understandability

Test Question: How understandable and informative are error messages?
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: NFR1, NFR2, FR14
Result: We did not test with non-developers, however our developers were able to interpret the errors.

A.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: FR13
Result: The emergency stop shut down the system.

A.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: NFR1, NFR2, FR14
Result: We reported almost all errors.

A.8 Power System

A.8.1 Performance Test: Battery Life

Test Question: How long can an individual robot run for on a single battery charge?
Operational Procedure: Charge a robot fully. Command the robot to complete an input drawing repeatedly 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: NFR7
Result: We were able to complete all of the inputs we tested without battery concerns.

A.9 Full System Validation

B Full System Validation

B.1 Performance Test: Painting Accuracy

Test Question: How closely does the drawn image resemble the original input?
Operational Procedure: Using example input sets for the system to complete. After completion, overlap the input with the image of final drawing captured from overhead camera. Rescale the two images so that they are in the same size. Evaluate the coherence of the two images.
Metric: The percentage of drawn lines that were within 3 pixels of difference compared to those of the original image.
Acceptance Criteria: The system must successfully and accurately draw 95% of the lines in the original input.
Requirement(s) Verified: NFR6
Result: When our assumptions were met we visually confirmed that our produced image matched the input. We did not perform a pixel based test.

B.2 Performance Test: Reliability

Test Question: How reliable is the system in terms of successfully complete a series of drawing tasks?
Operational Procedure: Command the system to finish a series of drawing tasks. Measure the number of consecutive successful completion. Successful completion is defined as the system autonomously finishes painting and the painting process is free of errors including but not limited to localization breakdown, motor breakdown, or painting mechanism breakdown. Calling human interference with switching battery and drawing utility does not count as unsuccessful run.
Metric: Number of consecutive painting completion.
Acceptance Criteria: The minimum acceptable number of consecutive completion is 5.
Requirement(s) Verified: NFR8
Result: Our system was reliable when our assumptions were met. We were able to place additional mechanisms in our system to insure some level of reliability when these assumptions were broken.

B.3 Functional Test: Size

Test Question: Is the robot agent too big to be portable, i.e. carry the robot through a standard door?
Operational Procedure: Measure the physical dimensions of the robot in terms of width, length, and height or in terms of diameter and height.
Metric: Numeric value of each length measurement; robot footprint; robot volume.
Acceptance Criteria: Must be less than 80 in. x 36 in. x 36 in.
Requirement(s) Verified: NFR4
Result: We fit within our size requirement.

B.4 Functional Test: Weight

Test Question: Is the robot agent too heavy to be portable, i.e. able to be lifted by a normal person?
Operational Procedure: Measure the mass of the robot.
Metric: Numeric value of robot mass.
Acceptance Criteria: Must be less than 50 pounds.
Requirement(s) Verified: NFR3
Result: We fit within our weight requirement.

B.5 Functional Test: Budget

Test Question: Does the cost for developing this robotic system exceed our budget?
Operational Procedure: Document total amount of money spent for designing and constructing this robot system. This includes machining expense, part cost, and etc.
Metric: Total amount of money spent.
Acceptance Criteria: Total developing expense has to be less than $2500.
Requirement(s) Verified: NFR10
Result: We fit within our budget.

B.6 Functional Test: Safety

Test Question: Is the robot safe during operation? Specifically, when collision happens, will the robot harm other robots, external environment, or human?
Operational Procedure: Count the number of sharp edges on the exterior of the robot. Also, measure the time it takes from the overhead camera detects collision to robot agent stops moving motors. Intermediate steps involved are: camera sends collision signal to system controller and system controller commands involved robot agent to stop its current action.
Metric: Number of sharp edges (angles less than 90 degrees); amount of time takes from detection to action.
Acceptance Criteria: Values for these two metrics need to be as small as possible. Zero sharp edges can be on the exterior - any edges from, for example, a rectangular chassis, should be rounded. The maximum amount of time is 1.5 seconds.
Requirement(s) Verified: NFR11
Result: We had zero dangerous sharp edges.

B.7 Functional Test: Documentation

Test Question: Is the documentation for the developing process comprehensive and replicable?
Operational Procedure: Give the full documentation to another design group or stakeholder and inquiry if they can duplicate the project with those documents.
Metric: Boolean on whether or not reviewers can replicate the system development process.
Acceptance Criteria: Reviewers are confident to replicate system development process based on the documentation.
Requirement(s) Verified: NFR1
Result: We produced an acceptable amount of documentation for project reproducibility.

C Planner Inputs

The following are a set of sample drawing inputs that were used to test the system. Some test inputs have been randomly generated while others were designed to stress test a particular feature.

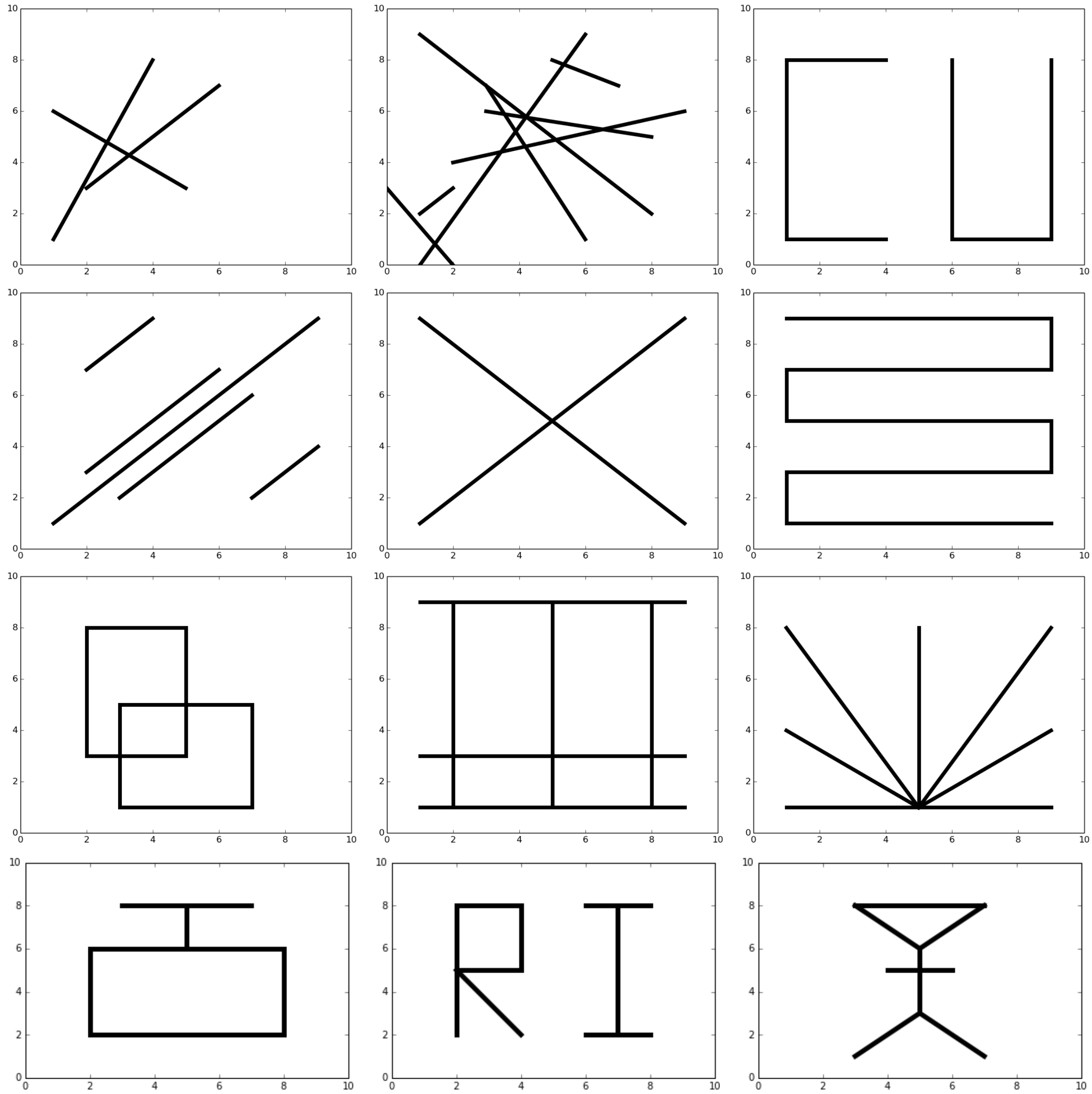

Figure 12: Planner Test Cases.