Carnegie Mellon University

Robotics Capstone Project

Concept Design

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. David Wettergreen

Version 2
November 9, 2016

Contents

List of Figures

RH: Need to fix references in system description and concept operation. Need to explain deployment in installation. Need to full in subsystem listing. Need fix each subsystem description. Need to add requirements fulfilled. Need to add artistic sketch + trade study. Need to add research for planning.

1 System Description

The goal of our project is to develop a multi-agent robotic painting system. Using~~Use~~(RH, V2) a team of homogenous robots, we can efficiently parallelize the dull and often dirty task of painting. Our system is motivated by the miles of paint needed~~needs~~(RH, V2) to maintain runways and sport areas as well as the widespread use of sidewalk chalk. Below, in Fig.1, we model our overall system graphically~~graphical~~(RH V2), with a more detailed description following. We then enumerate each subsystem in Sec. **??**.

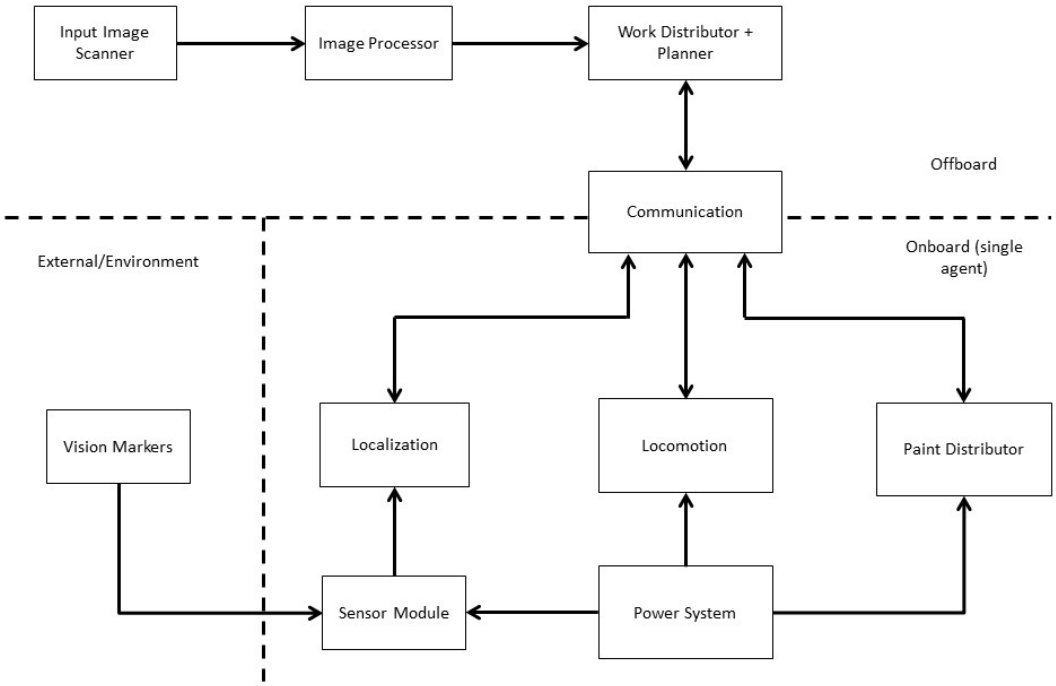

Figure 1: High Level Architecture Diagram

Our system encapsulates three components, shown in Fig.1 through dotted lines. First, there is a single offboard unit that hands off most~~hands most~~(RH, V2) of the coordination, planning and user interface. The initial drawing process begins in the offboard unit, where users will generate an input image ~~and process it through a scanner~~(RH, V2). Image processing will then occur, where the input is processed into a format compatible by the planner and work distribution subsystem Sec. **??**. Moving forward, the planner module will take in updates to current robot progress and use it to update commands it sends to robot agents.

Our second component is the robotic agents, marked in Fig.1 as "onboard". While the diagram shows one onboard agent, this schema is the same for each agent. Commands are sent~~sednt~~(RH, V2) to these agents. Communication protocols are able to send information both to and from the work planner to robot agents (Sec. **??**). From here, onboard systems are separated into three subsystems: Localization (Sec. **??**), Locomotion (Sec. **??**), and Paint Distribution (Sec. **??**). The paint distributor is involved with all commands regarding engaging and disengaging the writing implement to output paint onto the writing surface. Locomotion is the drive system, which engages wheels and motors used for movement. This subsystem also includes any motion-related safety mechanisms, such as quickly stopping. The Localization subsystem works to determine the position and orientation of an individual

robot agent. This subsystem takes input from the sensor module (Sec. **??**), which provides data about the surroundings for the localization system to process.

The sensor module gets data from external vision markers (Sec. **??**) for processing by localization. It is important to note that the sensor module, along with the locomotion and paint distribution subsystems are all powered by an individual subsystem Sec. **??**.

Finally our third component comes from environmental setup, which is mainly the situational landmark tools need to localize the robot.

2 Concept Operation

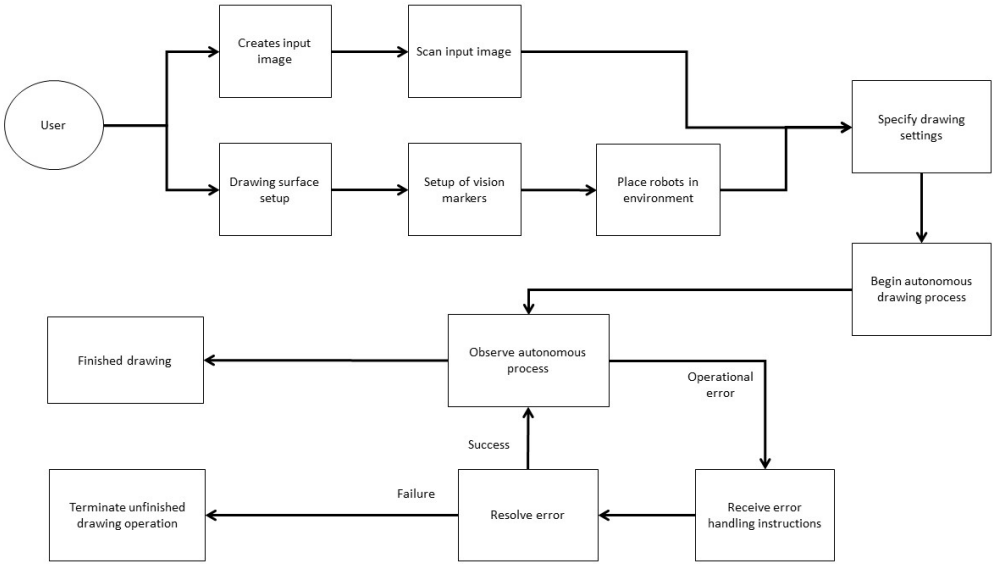

Figure 2: State Machine of Concept of Operations

Above, in Fig.2 we outline the user experience with the system. This is detailed below and our user interface is fully explained in Sec. **??**.

The initial setup contains two operations that can be performed simultaneously: adding the image to be drawn to the system, and setup of the drawing surface. Adding the input image involves generating the image, and then scanning it into the system. This satisfies requirements for input the drawing plan (Requirements Specification 5.1, FR9), and a user interface (Requirements Specification 5.1, FR12). Setup of the system involves placing the drawing surface, placing and calibrating vision markers, and finally placing the robot agents within the bounds of the drawing surface. Once both steps are done, the user can enter any required settings for their use, and begin operation. Processing of the input image is done automatically by the system and is invisible to the user (Sec. **??**).

Once the autonomous drawing process begins, the user simply observes the robots until they(RH V2) complete the task. However, in the case of problems, error(s) will be reported to the user, who then has the option of fixing the issue to continue operation, or terminating the drawing process. The autonomous process satisfies functional requirement FR2 (Requirements Specification 5.1, FR2).

3 Subsystem Listing

Rest of our document is split up by subsections. We describe these subsections.

4 Writing Implement

The writing implement actuator deposits writing material onto the working surface. It manages reloading of the writing implement and ensures that the implement is properly secured. Additionally, it can extend writing material closer to the ground as the material is shortened through use, and it can rotate the writing material for variation in stroke. In the event of writing material depletion, its sensors will detect the occurrence and alert the communication module (Sec. **??**). This subsystem satisfies writing tool related requirements (Requirements Specification, 5.1, FR6, FR8). **RH: How does it fulfill these requirements?**

The writing implement system is modular, in that various writing tools (such as chalk, markers, pens, etc.) can be easily inserted by users. This involves a fixed motor mechanism on the robot, with mounts for each tool that can be locked into place on the robot. The mounts can optionally connect to two motors: One for linear motion to raise and lower the writing implement, and another to rotate the writing

material as described above.

Critical components to this system include(RH, V2) the writing implement holder, actuation mechanism, writing material levels sensor and reloading mechanism. We discuss these mechanical components in further detail in our trade study.(RH, V2)

4.1 Use Cases

4.1.1 Load/Switch Writing Implement

Description: The robot has the ability to allow quick swapping of writing implements. This may be necessary when the current implement has been depleted, or a new implement with different writing properties is desired. The writing implement actuator itself is modular and detachable, so different kinds of writing implements (markers, chalk, etc.) can be used in the same robot.(DZ, V2)

4.1.2 Actuate Writing Implement

Description: The robot can use its writing implement to make markings on the working surface. It can vary properties such as stroke size and thickness to a small degree by rotating or adjusting the pressure on the implement.(DZ, V2)

4.2 Trade Study

RH: Need to commit to chalk/liquid chalk. RH: Going to be prototyped.

4.3 Artistic Sketch

RH: insert eric drawing

4.4 Requirements Fulfilled

5 Locomotion

Description: The robot's locomotion system propels each individual robot across the working surface. If the robots use a holonomic locomotion systems, they are able to move in any direction, ensuring their ability to execute intricate designs. They also provide enough clearance for the writing implement to work effectively. In the case of a tread-based locomotion system, individual robots can rotate in place to move in all directions. A driving system that allows for multidirectional movement satisfies motion-related requirements (Requirements Specification, 5.1, FR7). Additionally, the locomotion system contains encoders whose data is used by the localization module (Sec. **??**).
Critical Components of this subsystem include(RH, V2) wheels or treads, motors and encoders. We discuss the choice of wheels in our trade study.(RH, V2)

RH: Comment from DW: "This description of the locomotion hardware is insufficient. How does it work? There is not enough information and analysis to move into a detailed design of how it would be built." Will the trade study address this?

5.1 Use Cases

5.1.1 Locomote

Description: The robot can use its array of motors and wheels to propel itself across the flat working surface. It can make arbitrarily sharp turns and acute curves in order to allow input drawings that incorporate such components. (DZ, V2)

5.2 Trade Study

NJ: All Trade studies should be subsubsection RH: Locomotion

One of our functional requirements with the highest priority (FR1) is to be able to move in specified directions with a high degree of accuracy. Additionally, a medium high priority functional requirement (FR 9) is to have a drive control system that enables this movement. Therefore, it is clear from our requirements that locomotion is critical to our robot's performance. When considering locomotion options there are three large categories: aerial, legged and wheeled.

Our drawing occurs on the two dimensional plane. Therefore, any flight based system would have to constrain one translational degree of freedom and two rotational degrees of freedom. Doing so would require precise control algorithm and large power input. Practically speaking, this seems to over-complicate the problem with little additional gain.

Therefore, we are narrowed down to a legged or wheeled system. While a legged system would have the ability to traverse uneven terrain, such capability is not necessary for our project due to the assumption (A3) that the drawing surface is flat and homogenous. Compared to wheeled systems, legged systems usually involve more complicated control algorithms and more complex electrical and mechanical components. Furthermore, we are inspired by the wide spread success of wheeled robotic systems and especially appreciate their stability, a critical concern when precisely drawing images.

Therefore, we have concluded that our agent's drive system should be wheeled. In our next analysis, we will further investigate different types of drive system.

RH: Drive System In continuing to prioritize our functional requirement (FR1) of being able to accurately move in specified directions, we further investigate possible drive systems.

Three different drive systems are evaluated in this section: differential drive, ackerman steering, and four-wheel steering. Differential drive on a robot typically consists of two independently driven wheels and a non-driven wheel. The non-driven wheel often creates unwanted resistence when turning which compromises the drawing accuracy. Occasionally, the non-driven wheel would create singularity and ruin the drawing mission. Moving up in complexity and taking after some cars is ackerman steering, where the back pair of wheels is fixed in orientation but the front pair can pivot. Such drive system does not suffer from singularities. However, turning with ackerman steering often requires large turning radius which means drawing shape corners extremely difficult. Another option is four-wheel steering, which is similar to ackerman steering but allows the back wheels to pivot. Even though such drive system can achieve in-place turn, doing so often requires dragging four wheels in their lateral directions which can potentially demage drawing surfaces.

These systems all suffer from the fundamental issues of being nonholonomic. Therefore, we want to use a holonomic drive system to easiliy realize omnidirectional motion. This can be achieved with omniwheels, perhaps more commonly, with mecanum wheels. To eliminate singularities and large turning radius, we plan to use four-wheel steering system with mecanum wheels attached. To investigate potential drawbacks with this drive system, we plan to prototype a simple chassis with four mecanum wheels installed and test its mobility and motion accuracy. **NJ: Need conclusion that selects mecanum wheels**

5.3 Artistic Sketch

NJ: Description of sketch and how it relates to trade studies above

5.4 Requirements Fulfilled

NJ: Lists which requirements this subsystem fulfills

done

6 Localization

The localization system uses a hybrid infrared and ultrasonic marker-based method (Sec. 6.2) to determine a robot's position and orientation. By using the delay between the infrared and ultrasonic pulses detected by the sensor module (Sec. **??**) it computes the its distance to each marker. After performing some trigonometry, it can then accurately determine position. Orientation is determined by dead reckoning through the wheel encoders, and is constantly corrected by measuring changes in position. It then communicates the positions and orientations of the robots to the scheduling module (Sec. **??**). This

module directly satisfies local and global localization requirements, as well as indirectly allows for safe motion from the robots (Requirements Specification, 5.1, FR3, FR4).

Critical Components: Localization algorithm, sensor module, vision markers.

The environmental markers must be set up by the user before the system can begin drawing. The continuously output pings using both an infared light and ultrasonic transmitter. In doing so, they provide the data needed by the localization subsystem (Sec. **??**) to determine the robot's position and orientation in the workspace. They will be mounted high enough as to be visible by every robot in the workspace at all times.

The sensor module gathers data from environmental markers (Sec. **??**). This data is used by the localization module (Sec. **??**) to determine an individual robot's position and orientation in the workspace.

6.1 Use Cases

6.1.1 Localize

Description: The robot uses a combination of robot-mounted and static environmental markers, as well as a camera mounted above the working surface to determine the locations of each of the robots. This information is used to determine the motion plan of the robot.(DZ, V2)

6.2 Trade Study

In multi-agent planning, it is important to accurately localize robots' positions and orientations. Keeping in mind limitations in price and ease of use, we come up with two major methods for localization: vision based and marker based. They are described below.

Vision-based localization involves using cameras or other visual sensors to directly obtain information of the environment and localize the robots based on found landmarks in that environment. One example of this is SLAM (Simultaneous Localization and Mapping), often used by autonomous vehicles to simultaneously build maps and localize [1]. With this approach, robots could build small maps of their surroundings and match their locations to features they find in the environment. Benefits of this method include being location agnostic and requiring no additional parts or external setup. However, pure vision systems are difficult to calibrate and localization accuracy can depend heavily on static surroundings, which is not something this system can guarantee.

The other choice of methodology is marker based. Using markers placed around the drawing surface, robot agents can quickly locate these markers and their positions relative to each marker, and consequently triangulate their positions and orientations. While requiring additional setup and more parts to calibrate than vision based localization, existing technology makes it convenient and cheap to get marker based localization working. One example of a marker-based localization system is AprilTags [2], which can be described as 2D barcodes placed in a scene. Marker-based localization can be further classified into two subcategories: passive and active. Passive markers do not output any information and exist for the robot agents to observe and triangulate accordingly. AprilTags is an example of the passive marker system. On the other hand, active markers will look at robot agents to determine where the agents are, rather than the robots searching for markers. While less common, active systems behave well in conditions when the markers may not always be easily visible to robot agents [3].

Given the convenience and ease of use of marker-based localization, it is clearly the better choice. However, it is more difficult to determine whether using active or passive systems will be more effective. A prototype for each system should be made for further evaluation.

RH: Need to commit to overhead camera and description of it.

6.3 Requirements Fulfilled

7 Image Processing

The image scanner takes a user-provided image or generates one, and transfers it to the image processing module (Sec. **??**). The image scanner, being a module requiring user interaction, will also incorporate a front-end user interface (Sec. **??**). This subsystem satisfies requirements for adding a drawing plan

(Requirements Specification, 5.1, FR9).

The image processor takes input from the image scanner (Sec. **??**) and produces information that is recognizable by the planning module (Sec. **??**).

7.1 Use Cases

7.1.1 Input User Image

Description: The system can take in an image provided by the user, and from it determine what the workspace should look like upon completion of the task. The user also inputs parameters such as scale, resolution, and possibly color.(DZ, V2)

7.2 Software Architecture

RH: Explain why?

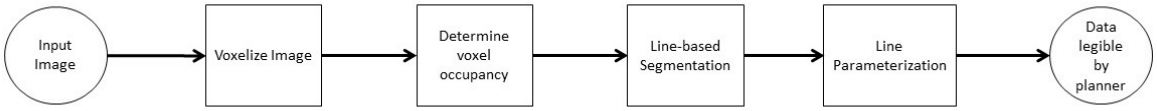

Figure 3: Image Processing Software Subsystem

The image processing pipeline (Fig.3) takes the image scanned by the user, and parses it into a form readable by the work planner. The goal of this software system is to create a series of lines that describes the image. To do this, the system first creates an occupancy grid of the input via voxelization. For a black and white image, the occupancy grid determines black or white. In the case of color, voxels are assigned color based on the image values inside of the voxel.

Once voxelization is complete, lines are formed from the voxels through a nearest-neighbor search. These lines are simply a series of voxel squares that pass through the image. In order to preserve curvature of lines from the input, the grid-delineated lines are reparamterized with splines into paths. These paths are then sent to the work planning module.

7.3 Requirements Fulfilled

8 Work Scheduling, Distribution and Planning

Given the output from the image processor (Sec. **??**), and parameters such as the number of robots and the size of the workspace, the module determines the work that will be distributed to each robot. See software architecture (Sec. **??**) for more information. Planning between robots allows for coordination and efficiency, satisfying requirements for both (Requirements Specification, 5.2, NFR2, NFR7).

8.1 Use Cases

8.1.1 Decompose Plan

Description: The system generates an efficient distribution of work to each of the individual robots. The work distribution takes into account the amount of writing material needed, thereby limiting the need for reloading. It is also based on distance that the robots must cover and the speed at which they must travel to ensure that robots sit idle for as little time as possible.(DZ, V2)

8.1.2 Schedule Lines

Description: The system can divide each robot's specific tasks into subtasks, and determines the order in which they will be completed. This order ensures that the robot spends little time traversing unnecessary distance and completes its tasks in a short amount of time.(DZ, V2)

8.1.3 Plan Movement

Description: Each robot is capable of determining how to actuate its motors to get from its current location to a desired location. By using its motion model based on the properties of its locomotion system, it can queue motor values and adjust for noise and error along the way.(DZ, V2)

8.1.4 Detect Collisions

Description: The system can use a combination of localization sensors and motor encoders to determine if any robot has encountered an obstacle.(DZ, V2)

8.2 Research

8.3 Software Architecture

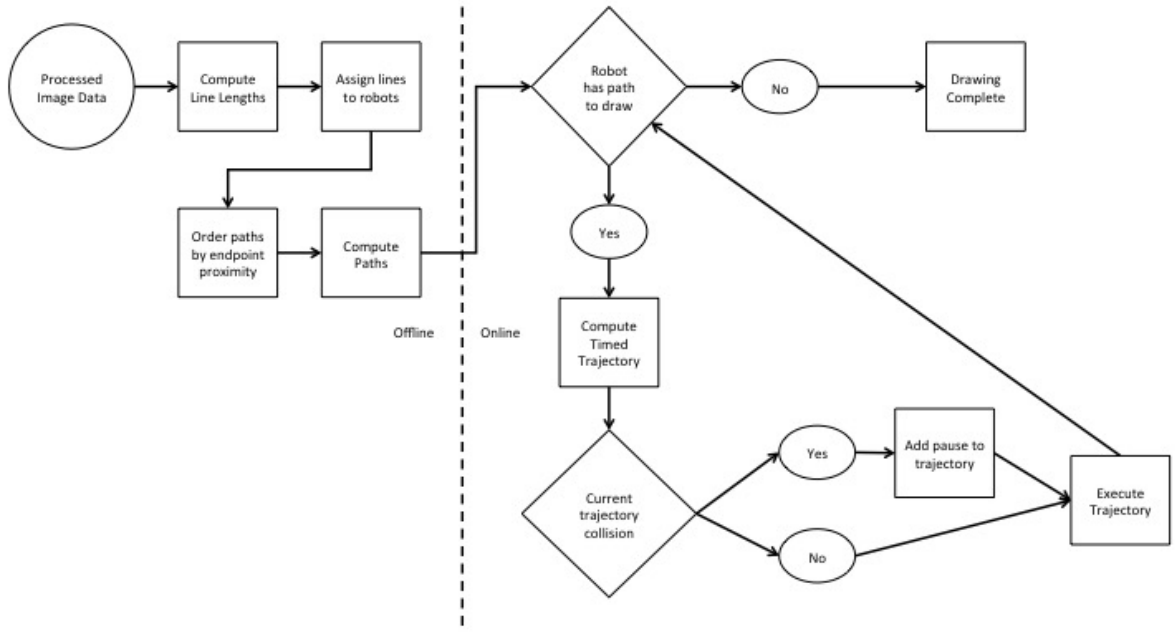

Figure 4: Planning and Scheduling Software Model

A key aspect of our software system is distributing and planning the work to the robot agents, modeled in Fig.4. Two of our nonfunctional requirements are to be efficient and have the robots coordinate with each other (NFR4, NFR9). These two objectives inform our planning pipeline. We split the planning and coordination into two separate problems, allowing us to frame coordination as a scheduling problem [4]. Hence our work distribution and planning can be handled offline while our coordinate occurs online.

From our image processing Sec. 7.2, we recieve processed image data. Using this data, we compute the length of every individual line. Guided by the assumption that line length correlates to amount of work done and the time to perform that work, we then use those lengths to load balance when assigning which lines should be drawn by each robot. Once each line has been assigned to each robot, each robot has a complete picture of their work and assign an ordering to their lines. To limit wasteful movement, the robots order lines greedily: having completed one lines, they pick their next lines (from their assigned set) based on which line has the closet endpoint to the lines they have just finished. Having ordered lines, we now can determine each robot's set of paths.

Next, we briefly describe a sketch of our coordination mechanism. Each robot has a queue of paths, based on the set ordering, to execute. Given that a robot has not finished all of it's assigned paths, it removes a path from the queue. The path is then uniformly timed and converted into a trajectory. Given timing information, the system can check for path collision between any trajectories currently being executed. If the trajectory is not at risk of collision, then it can be executed - and is sent to the robot via the protocol mentioned in Sec. 9.2.

If the trajectory is in collision with a currently executing trajectory, then the trajectory being processed defers to the one being executed. The timing of the processing trajectory is adjusted by adding a pause to avoid collision. Given this modified timing we can then execute it.

Once all paths have been drawn, the drawing is completed.

8.4 Requirements Fulfilled

9 Communication

The communication module is the link between the offboard system and the individual robots. To facilitate real-time changes in the working schedule, communication will be speedy and reliable. Potential communication protocols include WiFi and Bluetooth. Communication between the offboard system will allow individual robots to know their progress relative to the entire drawing (Requirements Specification, 5.1, FR10).

Critical Components: Antennae, wireless communication protocol.

9.1 Use Cases

9.1.1 Communicate and Parse Data

Description: Individual robots report sensor readings to the offboard central computer, and the central computer sends updated localization information and schedules back to the robots.(DZ, V2)

9.2 Software Architecture

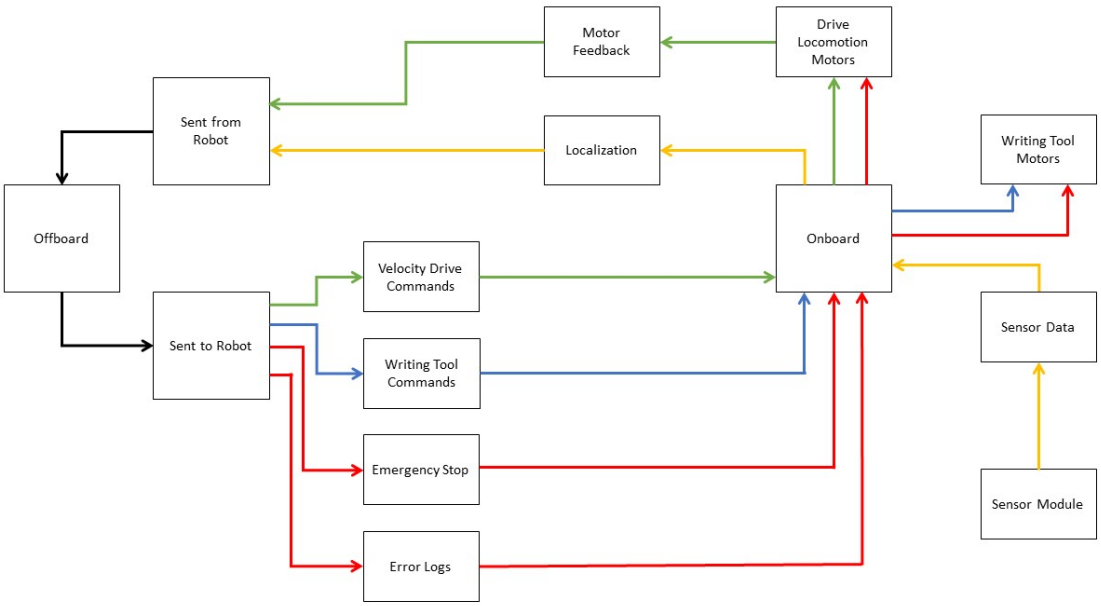

Figure 5: Communication Software Diagram. Green are the motion commands, Blue are the writing tool commands, Red is the error handling system, and Yellow is the localization system

The communication diagram (Fig.5) describes how an individual robot communicates with the off-board path planning system. The system can be described by communication in four categories: motion commands (green)(RH, V2), writing tool commands (blue)(RH, V2), error handling (red)(RH, V2), and localization (yellow)(RH, V2).

The offboard system will use the planning algorithm to determine motion plans for the robot systems. It determines and then sends velocity commands to the onboard system. The onboard system passes these commands to the drive motors for motion.

To command the writing tool, the offboard system also decides how to move the writing implement based on localization of the robot systems. Similar to velocity commands, the onboard system receives commands for the writing tool, and commands the system accordingly.

Localization receives input from the sensor module, and computes position and orientation. This data is then sent back to the central offboard module to update planning and scheduling.

Error logs are generated by the offboard system based on localization and planning, and sent to each robot's onboard system for appropriate reaction. This includes the emergency stop, which will immediately shut down writing tool and drive motors.

9.3 Requirements Fulfilled

10 User Interface

Description: The user interface provides a unified system for user input. This is the system with which the user specifies the input image and monitors the robots' progress. The system will also display any error messages or anomalies detected by the system, and how the user should address them. There will also be a system-wide kill switch in case of emergencies to satisfy requirement FR11 (Requirements Specification, 5.1, FR11).
Critical Components: Screen, input device.

10.1 Use Cases

10.1.1 Display Information

Description: The system is able to show a human user information pertinent to system operation, including but not limited to the location of the robots, their battery level, amount of task completed, estimated time of completion, and any existing obstructions or anomalies.(DZ, V2)

10.2 Requirements Fulfilled

11 Power System

The power system supplies power to the rest of the system. Each robot has an onboard battery that is small and light enough to satisfy the size and weight requirement, but also provides enough uptime to last a entire drawing session. The power system will satisfy battery life and contribute to portability requirements (Requirements Specification, 5.2, NFR1, NFR6).
Critical Components: Battery, voltage regulator modules.

11.1 Requirements Fulfilled

12 Installation

The system's setup requires a drawing surface and vision markers under specific conditions. Conditions for setup must be in line with the scope and assumptions made for successful operation of the system. Maintaining scope of the project, as described in the requirements specification (Requirements Specification, 2.3), the drawing surface must be placed indoors and in an obstacle-free area. For compliance with surface requirements (Requirements Specification, 2.4, A1), the surface chosen must be a flat and homogenous surface.

Once the drawing surface has been placed indoors on a flat surface, vision markers must be set up and calibrated. As per Sec. 6.2, localization markers must be placed around the edges of the drawing surface. These markers must then be calibrated by the user, who will input marker locations into the system. The localization markers serve as calibrated base points from which the robot agents will determine their position and orientation.

NJ: Need to explain how system gets deployed

13 Requirements Table

Requirement	Section
1	6
2	7
3	545
4	545
5	88

References

[1] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *ICRA*, pp. 3400–3407, IEEE, 2011.

[3] R. Cassinis, F. Tampalini, and R. Fedrigotti, "Active markers for outdoor and indoor robot localization," *Proceedings of TAROS*, pp. 27–34, 2005.

[4] P. A. O'Donnell and T. Lozano-Pérez, "Deadlock-free and collision-free coordination of two robot manipulators," in *ICRA*, IEEE, 1989.