# Cooperative Autonomous Robots
(in search and rescue applications)

Functional Description and Complete System Block Diagram

Students:
Adam Salisbury, Austin Scott, Bryan McKinney, and Rob Ruber

Advisors:
Dr. Huggins and Dr. Malinowski

November 30, 2006

## I. Introduction

The purpose of the senior design project is to design and implement cooperative autonomous robots for search and rescue applications. The robots will cooperatively search an unknown terrain for certain objects (“victims”) that are located in the environment. The task will be accomplished with each robot cooperatively mapping the area and communicating with the other robot. The robots will communicate to each other an updated map, their current position, position of victims, and the possible need for help with saving victims. The project focuses largely on optimizing cooperative function of the robots. The goal is to demonstrate that they will be more effective and efficient completing a search and rescue task cooperatively, rather than independently. The measurables for this are as follows:

- Number of victims recovered
- Amount of total area mapped
- Execution time

## II. System Block Diagram and Function

The overall system block diagram is shown in Figure 1 and depicts the inputs to and outputs from ‘N’ identical robots. For each robot the outputs include the action of wheel rotation, movement, and tool manipulation. Tool manipulation will be the action of grasping the victim with a gripper located on the front of the robot. The inputs are tool feedback, data collected from the camera, sensor inputs, and user inputs. The camera and sensor inputs will be discussed in more detail in the next section. The COM inputs and COM outputs are digital signals transmitted via a wireless link. Finally, the robots will have a display panel to give the user updates and retrieve user inputs when necessary.

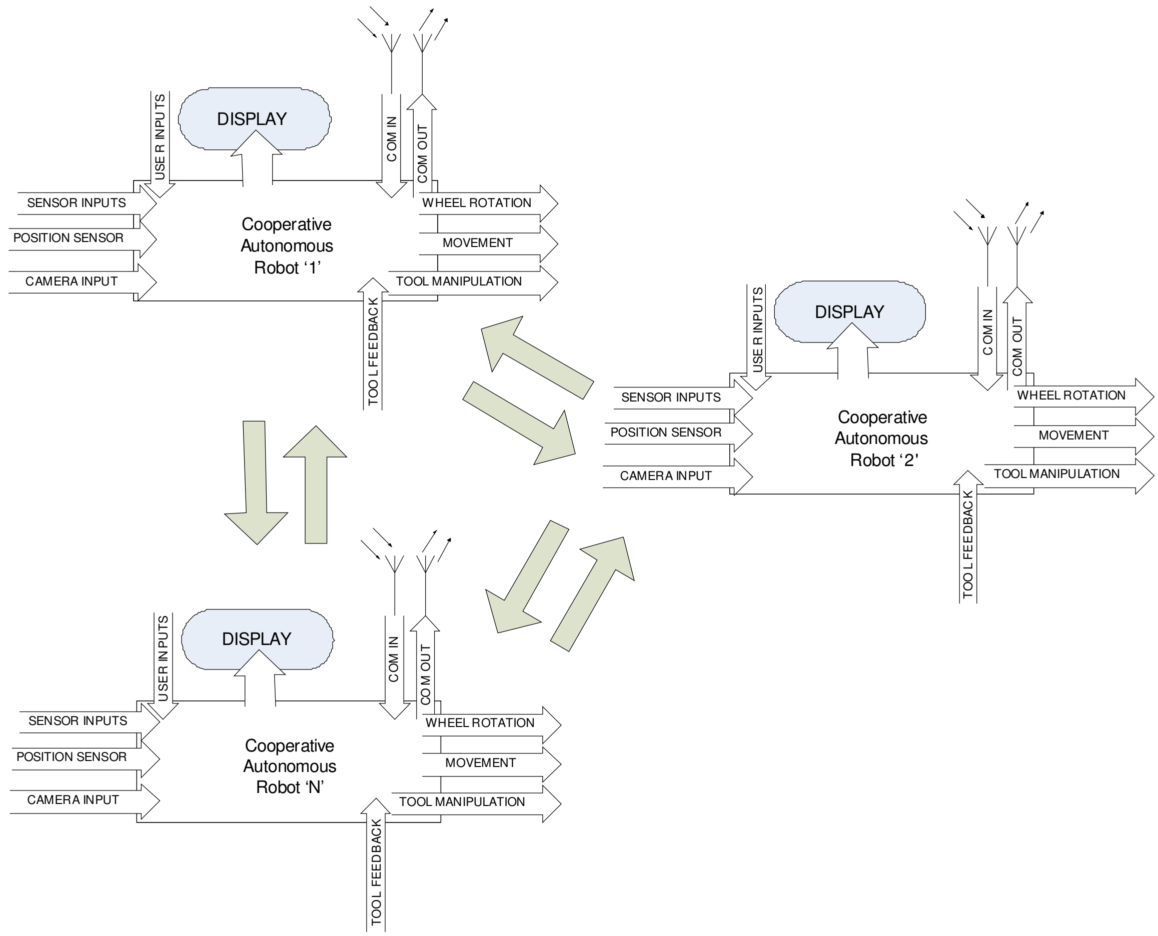


*Figure 1 – Overall System Block Diagram*

## III. Detailed Block Diagram and Function for a Single Robot

A detailed block diagram for a single robot is shown in Figure 2 in which a microcontroller (denoted as CPU) will be utilized as the main component of the robot. It will process and control all inputs and outputs. Wireless COM links will be used for the communication between the robots. The following table shows the main inputs to and outputs from the CPU.

*Table 4-1*
*"CPU Inputs and Outputs"*

| **Technology** | **Input / Output** | **Function** |
|---|---|---|
| Power | Input | Powers all equipment on the robot |
| Camera | Input | Used to identify victim |
| Distance Sensors | Input | Ultrasonic sensors for realizing distances from walls and obstacles for mapping purposes |
| Hazard Sensors | Input | Sensors to identify hazards |
| Other Robot's Current Status / Objectives | Input and Output | Used by robot to cooperate effectively by assisting in victim rescue or searching unknown map locations based on the other robot's current status and objectives. |
| Tool Sensors | Input | Used to prevent excess force being exerted by the grippers and to verify victim has been grasped |
| Updated Map | Input and Output | Robot sends and receives an updated map to or from the other robot(s) for merging |
| Power Electronics | Output | Includes wheel motors and tool control |
| Camera Control | Output | Control of a pan-mount fixture which allows 180° rotation of the camera |

Using the inputs described above, the CPU will process all the information, as described in flowcharts in later sections, and in turn develop an updated map, update its status and high level objectives, and send all necessary information to the other robot(s). The high level objectives a particular robot may have include the following:

- Navigate the area
    - Search for victims
    - Map newly discovered boundaries/objects/obstacles
- Rescue victim
- Communicate
    - Ask for assistance from other robot(s) in rescuing victims
    - Inform other robot(s) of current status including failure (described later)
    - Send/Receive updated map

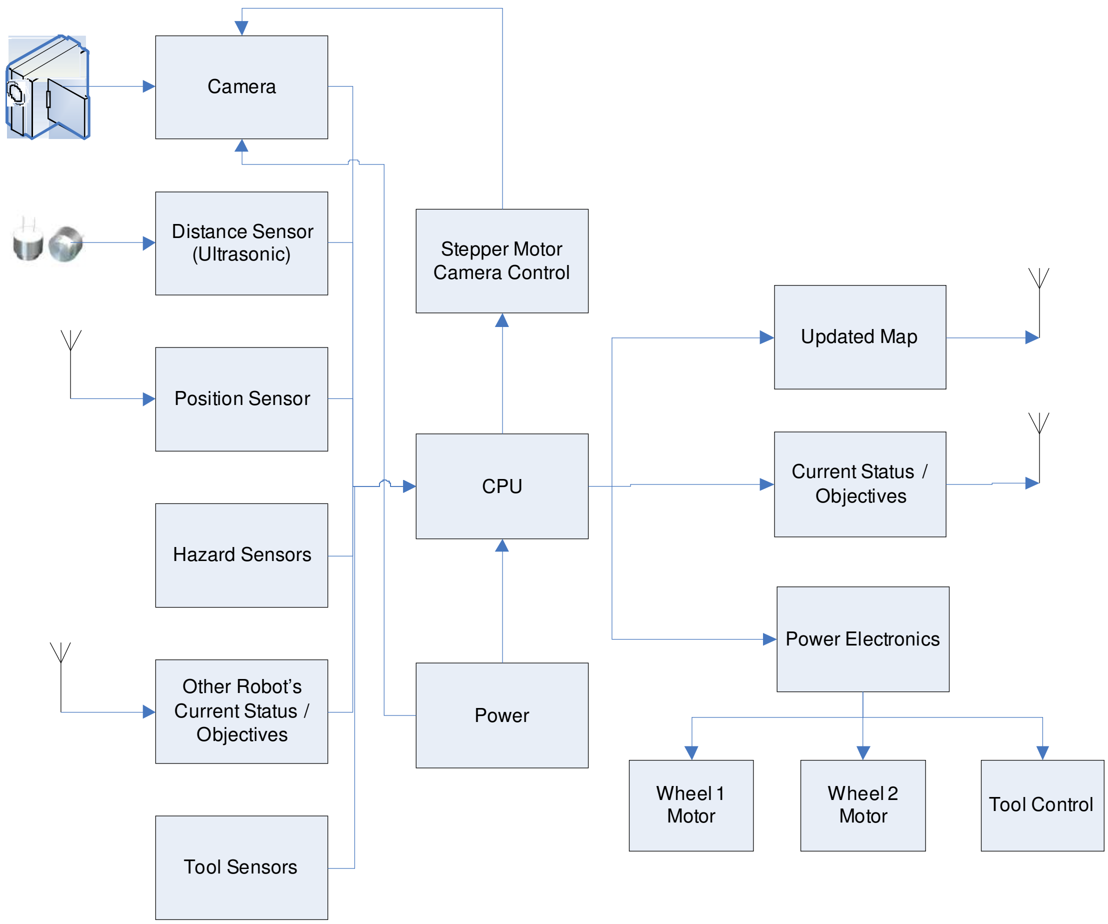


*Figure 2 – Block Diagram of a Single Robot*

## IV. Robot Modes

The high-level software flowchart is shown in Figure 3. Each robot will be identically programmed to operate in the following five modes: search, map, communicate, rescue, and failure. These modes are described in more detail in the following section. While the robots are working cooperatively, they will physically operate independent of each other and may therefore be in different modes at any given time. At first, the robots will be in search mode where they will search for the target(s) by panning a camera mount 180° and take pictures at 45° intervals. They will then continue forward following a predetermined path in map mode. The algorithm used to determine this path is yet to be determined. While moving ahead, the robot will simultaneously map the boundaries and obstacles that they encounter. After moving 1.5 meters, the robot will stop and return to search mode. The maps generated will be sent to the other robot(s) so each robot can generate a composite map. This way, both robots use the same combined map. Upon finding a victim, the robot will enter the rescue mode and will retrieve the victim by utilizing the onboard tool. Finally, failure occurs when a robot loses the ability to communicate or utilize any function that it is programmed to do. The robot will carry out a specific function based on the type of failure that occurs.

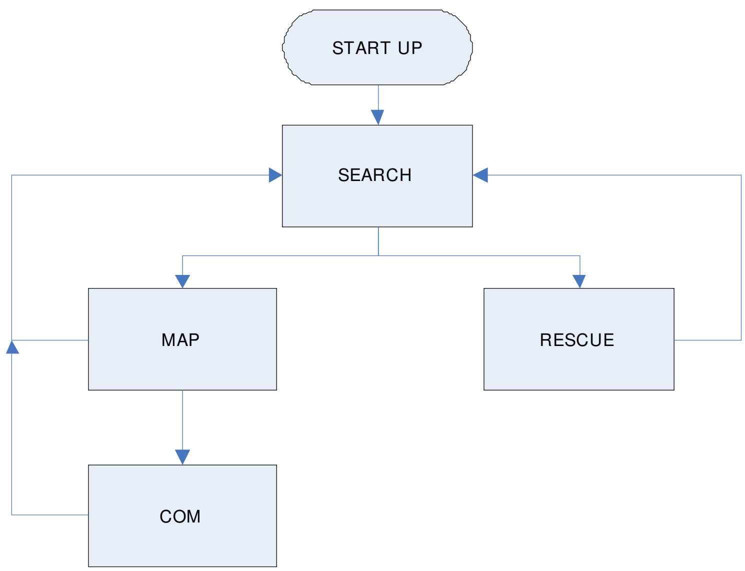


*Figure 3 – High Level Software Flowchart*

## V. Robot Function

The flowchart for the search mode is shown in Figure 4. In this mode the robot will remain stationary. A camera attached to a pan-mount fixture will capture an image and the CPU will use an image processing algorithm to determine if a victim is present. For this project, a victim will be a geometrical object of a given shape and color. If a victim is found the robot will enter rescue mode. Otherwise, the pan-mount fixture will rotate 45° and repeat the search process until a victim is found or the fixture has rotated a full 180°. Upon completion of searching for a victim, the robot will then enter map mode.

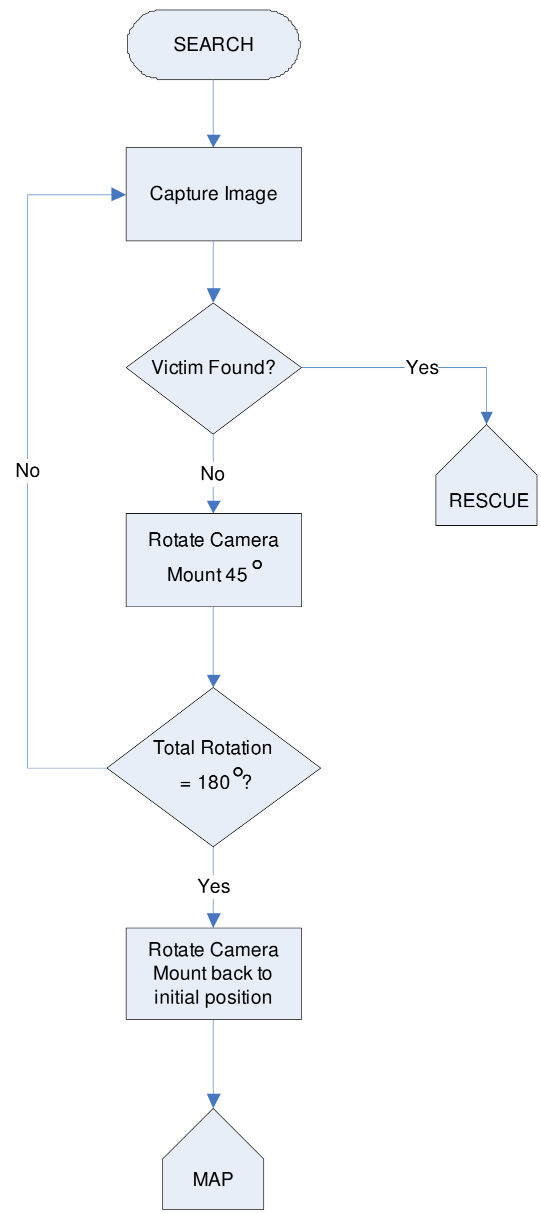


*Figure 4 – Search Mode Flowchart*

The flowchart for the map mode is shown in Figure 5. In this mode, sonar data is retrieved from the ultrasonic sensors located on the outside of the robot, as shown in Figure 6, and is used to generate a map of the terrain (Sullivan, 2005). The algorithm used will continuously check to verify that the current planned path is a viable route based on the sonar data. If the current path is feasible, then the robot will continue this process until it has moved a previously specified distance. Otherwise, it will change direction and return to search mode. Once the robot has traveled the total distance, it will enter communicate mode to share the additional map data obtained.

In the case of the robot having an objective of reaching a known targeted zone (known victim location or base), the robot will initially calculate the most efficient path to the targeted zone. The robot will then follow the process described above until it has reached its destination where it will either search for or drop off the victim. Since the robots' objective is to reach the targeted zone, stops every 1.5 meters to search are not implemented.

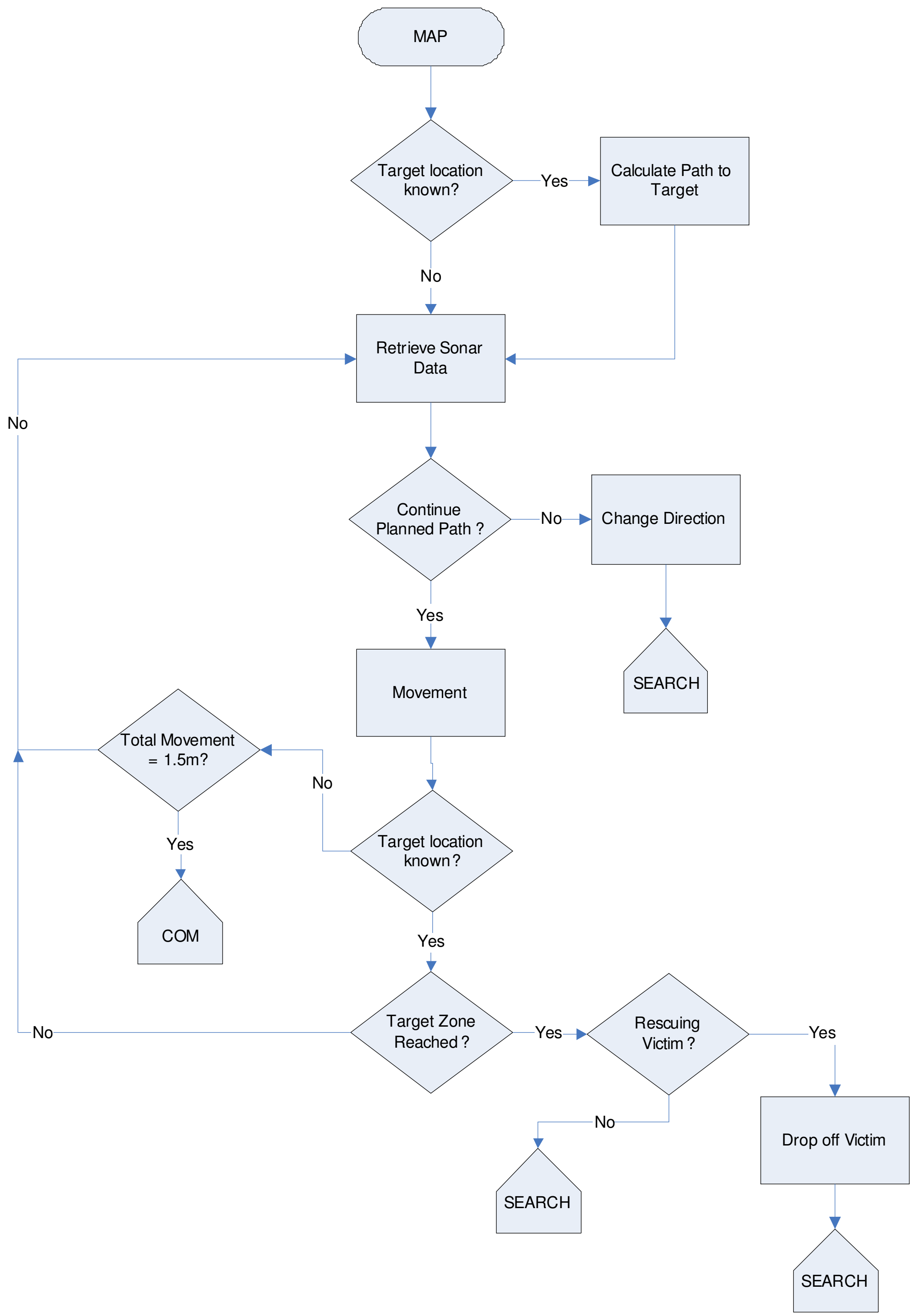


*Figure 5 – Mapping Mode Flowchart*

*Figure 6 – Ultrasonic Sensors on the Robot*

The flowchart for the communication mode is shown in Figure 7. Here the robot will either send or retrieve map information, or inquire about help with rescuing victims. When a robot is asked to help another robot in retrieving victims, a decision based auction is used. The auction is a process where the robots are "bidding" to do a certain task. The robots perform the appropriate algorithm, which is a function of their current status and objectives, to determine which objective each robot will be assigned (Sariel, 2005).

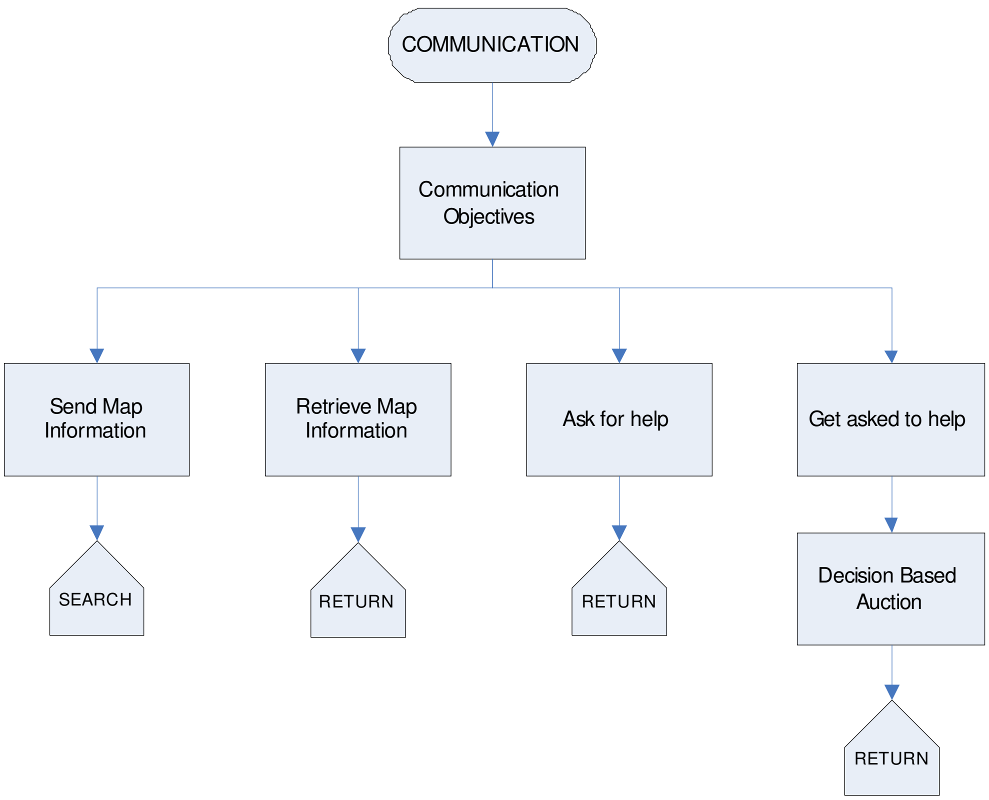


*Figure 7 – Communication Mode Flowchart*

The rescue mode is shown in Figure 8. During this mode the robot will properly position itself relative to the victim, pick it up, and then enter map mode. The camera and tool manipulation feedback will be the inputs to the system. The outputs in this mode are the tool manipulation and motion for correct tool positioning. This gripping device will be more accurately controlled by feedback which will provide data about the responsiveness of the gripper.

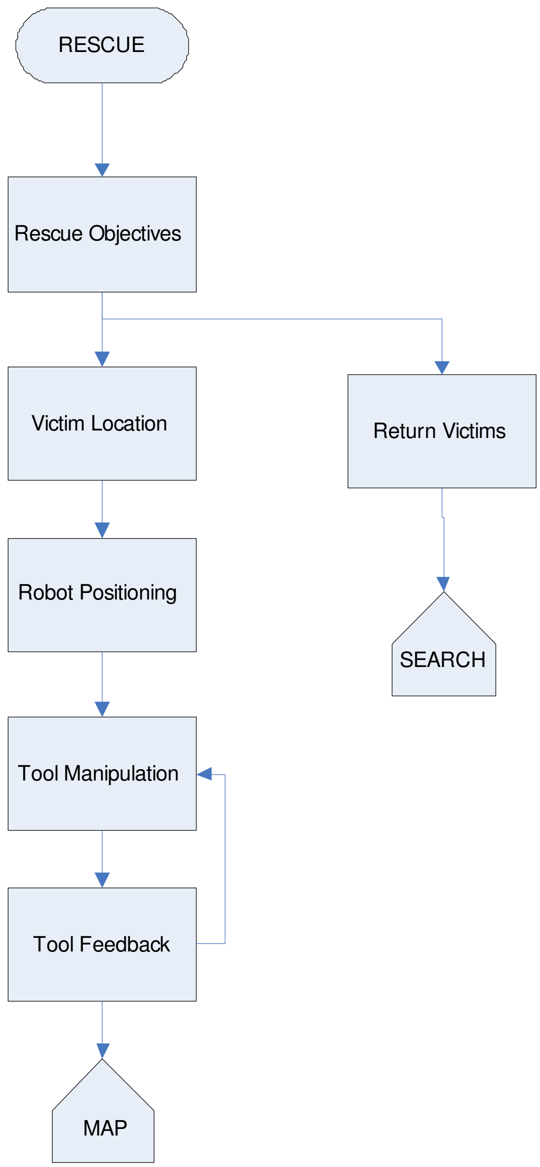


*Figure 8 – Rescue Mode Flow Chart*

The failure mode flowchart is shown in Figure 9. This mode will be triggered if there is a map error, the robot becomes immobile, there are communication issues, or if an unknown error occurs. A mapping error occurs if there is a discrepancy between the two robots' maps. In this case, the robot will remove the map data, retrieve the correct map form the other robot, and then attempt to return to the base. If the robot becomes immobile it will inform the other robot of its error status. If the communication system fails, then the robot will retrace its route back to the initial entrance. If an unknown error occurs, it will return back to the base and tell the other robot that something is wrong. If this is the case, then the other robots will attempt to complete the task. A message will be displayed on the screen detailing any error that has occurred. During this mode the outputs mainly consist of communication, display, and movement.

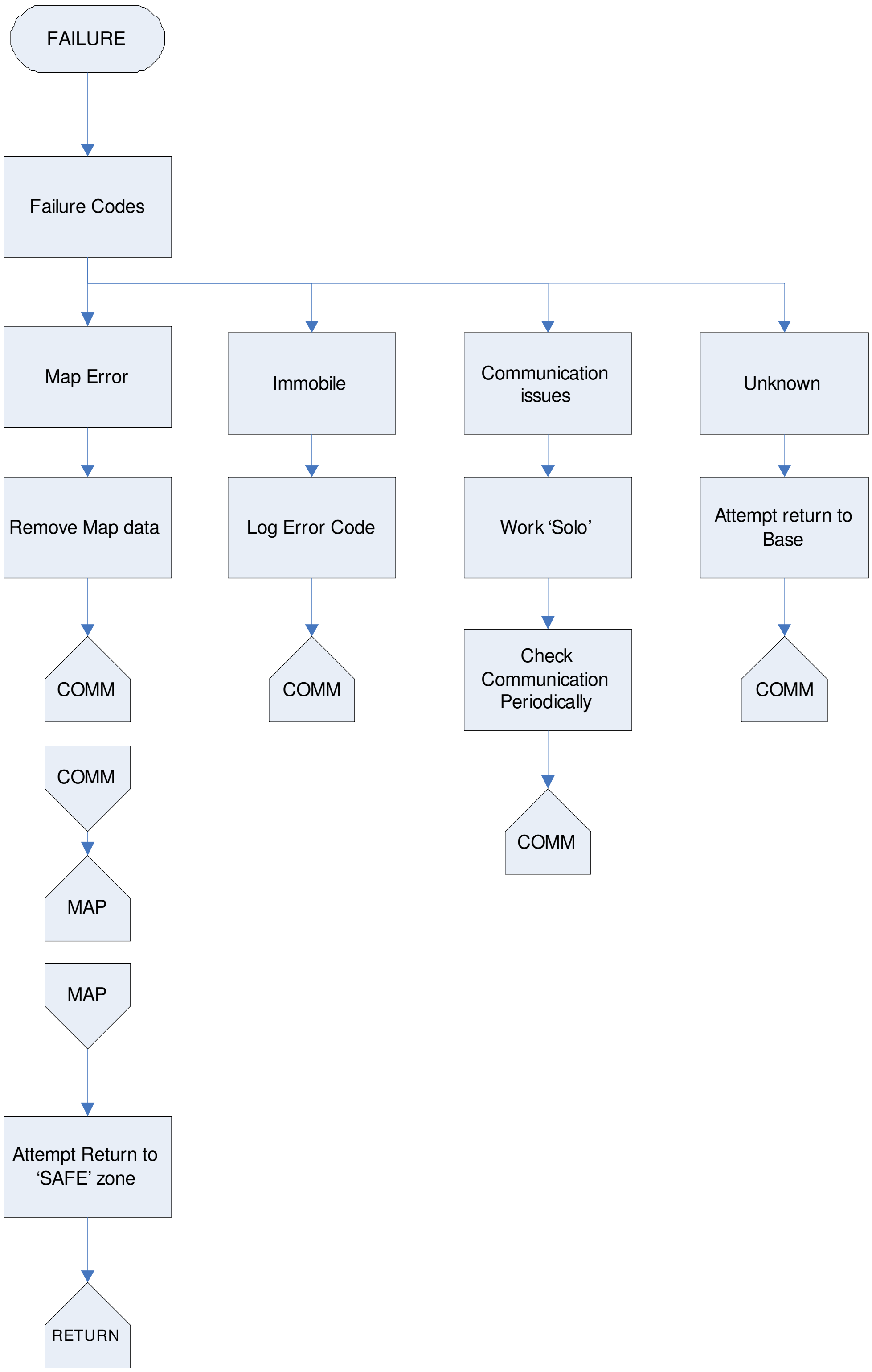


*Figure 9 – Failure Mode Flow Chart*

## VI. Conclusion

Cooperative autonomous robots will be used to search an unknown terrain, map the environment, and locate and save victims. Using wireless communication, the robots will be able to maximize results by cooperating with each other. Due to this cooperation, areas will not be covered twice and robots will have the ability to assist each other in recovering victims. Upon implementation of the final design of this project, the goal is to demonstrate that having cooperative autonomous robots perform a search and rescue task is more effective than robots operating independent of each other. Benefits include more victims saved, more area covered, and a faster execution time. In the case of search and rescue applications, the benefits previously mentioned are all keys to a successful operation.

## References


Sariel, S and Balch, T, 2005. Real Time Auction Based Allocation of Tasks for Multi-Robot Exploration Problem in Dynamic Environments. http://www.cc.gatech.edu/~sariel/publications/AAAI_workshop_2005_sariel.pdf

Sullivan, S, 2005. MapViewer – A Mapping Tool for Mobile Robots.

http://mapviewer.skynet.ie/