Indor Mapping

Validation and Verification

Group 3

Abdel-Latif, Sari	0840264
Batth, Chanderdeep	0856000
Bishara, Marc	0858892
Elsaftawy, Mahmoud	0951912
Mansour, Ahmed	0857403
Wahid, Fahim	0965325

25/02/2014

Table of Contents

List of Figures:

Revision Table:

Version	Date	Author	Description
0.0	23/2/2014	C Batth	Template to be followed
0.1	24/2/2014	C Batth	Added the POI
0.2	25/2/2014	M Bishara	Added SIM, CAR and CM
0.3	25/2/2014	M Bishara	General formatting
0.4	25/2/2014	A Mansour	Added MMU
0.5	25/2/2014	M Saftawy	Added the planner module
0.6	25/2/2014	M Bishara	Added introduction and system description
0.7	25/2/2014	S Latif	Added the Visualizer and command center
0.8	25/2/2014	M Bishara	Completed CM and various modifications
0.9	25/2/2014	S Latif	Final formatting and edits

Introduction:

Purpose:

This document will contain the details of the validation and verification test cases and results conducted on the indoor mapping system implementation version 1. It will assess the validity of the results of the implementation and their compliance to the initial requirements and design specifications.

Scope:

The document does not include any of the top system's requirements or any of the module system requirements. However, reference documents are available. This document contains black box testing of the modules and the system integration to insure the proper functionality of the different modules. Details on how some of the tests were conducted were omitted from the document for the sake of keeping it brief and to the point. However, inputs, outputs and results of those tests are very clearly indicated to give feedback on the state of the various modules.

References:

System Requirements specifications:

Group_3_Deliverable_2_Draft_System_Requirements_v0.24

Goals:

Group_3_Capstone Project Goals V3.pdf

Note:

Variable Names are case insensitive. Variable names have been chosen to be as descriptive as possible however Upper and Lower case has no inherent meaning.
(i.e. M_Human_Ctrl is the same as m_human_ctrl)
.

System Overview:

Module breakdown diagram:

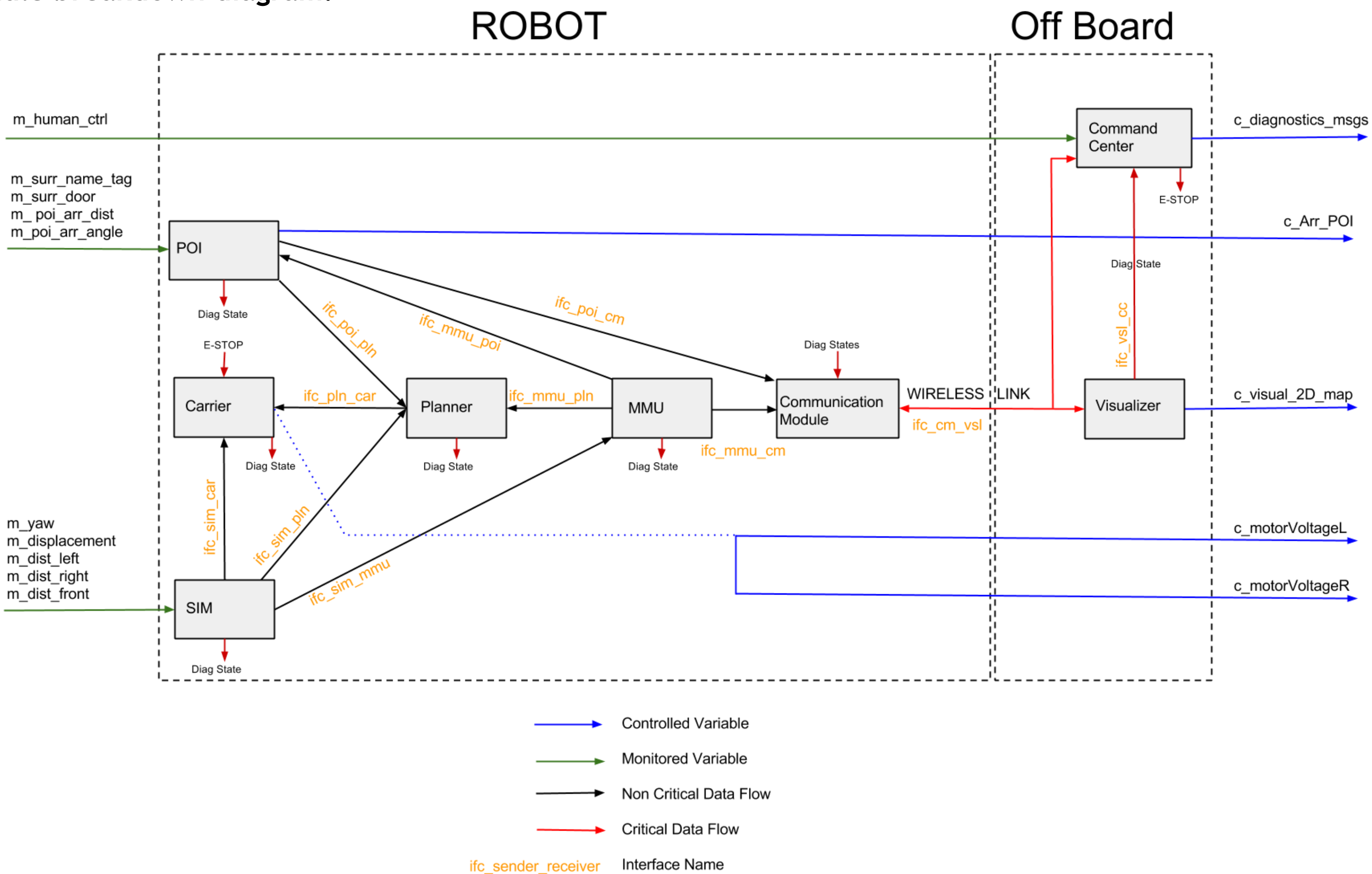

Figure 1 - System Modules breakdown showing interfaces, monitored and controlled variables

System Modules Description:

Module	Purpose	PC
SIM	Sensor Interface Module connects to all the sensors. It uses drivers to abstracts all the hardware from other modules.	Arduino
Carrier	Carrier module caries all the hardware and translates planner commands into actual movement. It includes the control system and feedback for insuring proper movement	
MMU	Modular Mapping Unit uses outputs from the SIM to create a map of the environment and localize the unit on that map	Beagle Black
Planner	Planner plans the exploration and movement. It monitors all the sensors and sends its commands to the carrier for execution	
POI	Point of Interest module monitors doors and nametags in the environment to locate them on the map	
Communication Module	The purpose of the communication module is to keep connect the robot with the off-board system. It handles all the buffering, sending and receiving of data for all the modules	
Visualizer	The Visualizer is off-board, receiving data from the MMU and POI it constructs the visual map for the user	Laptop
Command Center	The Command Center houses the GUI interface to the human allowing them to monitor the status of the various modules, the last known location of the Bot and an option to execute an Emergency Stop.	

System Hardware overview:

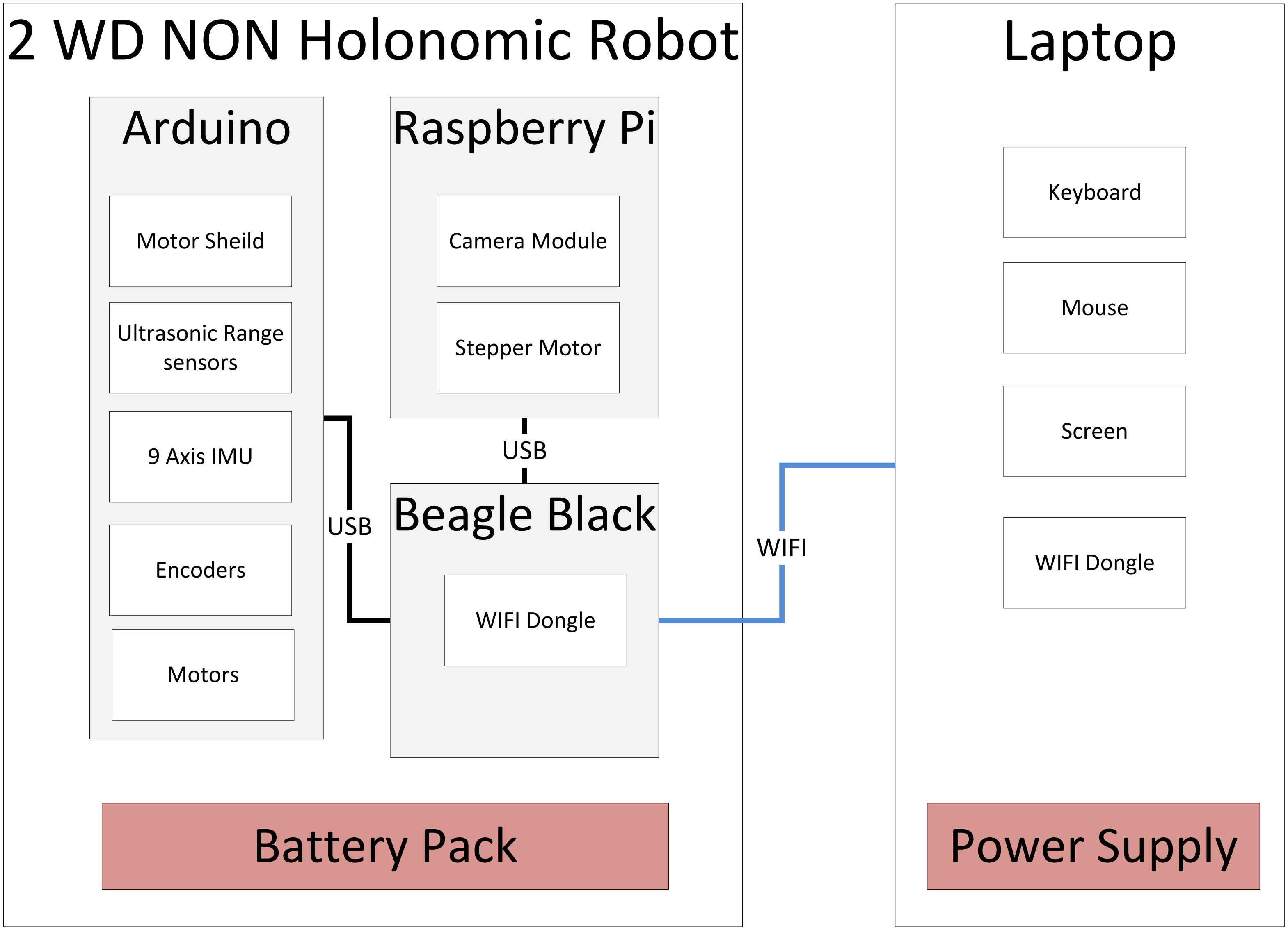

Figure 2 - Hardware overview

Definitions, Acronyms and abbreviations:

MMU	Modular Mapping Unit
POI	Point of Interest Module
SIM	Sensor Interface Module
CAR	Carrier Module
PLN	Planner Module
CM	Communication Module
CC	Command Center Module
VSL	Visualizer Module

System Monitored and Controlled Variables:

Variable	Units	Range		Accuracy
		Min	Max	
m_surr_name_tag	m^2	12 (Radius of 2 m around the bot)		0.5
m_surr_door	m^2	12 (Radius of 2 m around the bot)		0.5
m_poi_dist	m	0	2	0.2
m_poi_angle	m	0	180	5
m_yaw	Deg	0	360	2
m_displacement	m	All Real Numbers		5
m_dist_left	m	0	5	0.2
m_dist_right	m	0	5	0.2
m_dist_front	m	0	7.5	0.2
m_human_ctrl	NA	NA		100%
c_Arr_POI	NA	Real Positive Integers		80%
c_visual_2D_map	m^2	Real Positive Integers		5
c_left_motor_speed	m/s	0	0.3	0.01
c_right_motor_speed	m/s	0	0.3	0.01
c_diagnostics_msgs	NA	NA		100%

System Constants:

Variable	Units	Range		Accuracy
		Min	Max	
K_system_Speed	m/s	0	0.16	0.02

Sensor Interface Module

Summary

Sensor interface module (SIM) is meant to interface and abstract all the hardware on the robot. Testing has been done to insure that sensor behave as expected and that they integrate well when running along with other modules.

Test Cases

Validation

1. That the sonars can see the walls at the specified accuracies
2. That the sensor interface module can communicate with the command center and carrier modules
3. SIM can interface and run as expected with the other modules

Table 1: Test Cases for “Module Name”

Test ID	Predicted Value	Test Value	Result
1	+- 5 cm of actual wall distance	+- 5 cm	Pass
2	SIM can write to Serial bus and read from Serial bus	Successful simultaneous read and write	Pass
3	Sensor values are consistent when other modules are running	Sensor values are NOT consistent when Carrier module is running due to Motor magnetic noise	Fail

Verification

1. Observing the output frequency of the sensor data from the SIM and insuring that it is within the specified range stated in the requirement SIM_MR_01
2. The SIM is able to successfully measure the vehicle’s displacement within the specifications of SIM_MR_02
3. The SIM is able to measure the vehicle’s speed with the specifications of SIM_MR_03
4. The SIM is able to monitor the vehicle’s yaw within the specifications of SIM_MR_04

Table 2: Test Cases for “Module Name”

Test ID	Requirement ID	Predicted Value	Test Value	Result
1	SIM_MR_01	10 HZ	10 > X > 25	Pass
2	SIM_MR_02	Within 5% of Real displacement	13% of real displacement	Fail
3	SIM_MR_03	Within 5% of Real vehicle speed	Within 1% of vehicle speed	Pass
4	SIM_MR_04	Within 5% of Real vehicle orientation	Within 3% of Real vehicle orientation	Pass

Analysis

Sensor interface module is up to specifications in meeting the requirements. However failure to integrate with the carrier is causing major setbacks. The errors originate from the motors on the carrier module and as such will have to be fixed by considering a re-design in the carrier module.

Carrier

Summary

The purpose of the carrier is to translate the planner commands into actual translations in space; the testing has been completed to ensure that it meets the requirements specifications. As well, testing was completed to ensure that the level of functionality achieved will allow the carrier to be able to navigate through the test arena.

Test Cases

Validation

1. The carrier is able to go in strait line down the whole hallway compensating for any wheel slippage
2. Carrier is able to adjust for floor imperfections to keep going in the requested orientation

Table 3: Test Cases for "Module Name"

Test ID	Predicted Value	Test Value	Result
1	The carrier driving down a complete 50 m hallway in a straight line	As predicted	Pass
2	Carrier being able to adjust and when one wheel is help up or slipping	As predicted	Pass

Verification

1. Carrier is able to achieve the speed and orientation requested by the planner within the specifications of CAR_MR_02 & CAR_MR_03
2. Carrier shall have a fail-safe if it detects objects closer than a certain distance as to satisfy CAR_MR_04
3. Carrier shall be able to carry out an emergency stop if requested to do so as specified by CAR_MR_05

Table 4: Test Cases for "Module Name"

Test ID	Requirement	Predicted Value	Test Value	Result
1	CAR_MR_02 & CAR_MR_03	Carrier being able to go straight and turn in all angles from 0-360	As predicted	Pass
2	CAR_MR_04	Carrier stopping as it comes dangerously close to obstacles	Carrier crashes into obstacles	Fail
3	CAR_MR_05	Carrier stopping when receiving emergency stop signal	Functionality not implemented	Fail

Analysis

The carrier is able to reliably move in a straight line down the hallway using a closed loop system. As well it reliably turns 45, 90 and 180 degrees allowing it to navigate in the area of the test (ITB second floor). However the carrier is not measuring the displacement accurately enough which means that the hardware will be revised to achieve the requirements.

Planner

Summary

The main purpose of the Planner module is to be able to guide the Carrier around a hallway following the right-hand rule algorithm. Three sonar sensors located on the left, right and front of the Carrier detect the distance from the wall to the Carrier. As the Carrier drives forward, depending on the sensor readings, a suitable speed and direction is sent to the Carrier to control its route. Black box testing of the right-hand rule algorithm is crucial to ensure the Carrier follows the right path and without missing any hallways. This purely software module is also responsible for keeping track of the displacement covered by the Carrier and determining whether the mapping process is complete, i.e. the Carrier reached the starting point.

Test Cases

Validation

To ensure that the system or module is performing tasks that the user originally intended for, we need to check for the overall validity of the module. The following test case is used to determine if the system is valid. The test performed is used to validate the module as a whole rather than portions of the module.

Table 1: Validation Test cases for Planner

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	i_displacement = 0 45 < i_yaw <=45	o_speed = 0 o_angle = 0	N/A	Fail

This test requires the Planner module to be connected to and running with the SIM module (which sends displacement data to the Planner). However, this level of integration in the project has not been reached yet and is therefore incomplete.

Verification

To check if different units of the Planner module are functioning correctly and fulfilling the requirements, we need to perform black box and white-box testing on those units.

Table 2: Verification Test cases for Planner

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	0 < i_dist_left < 700 0 < i_dist_right < 700 i_dist_front >= 700	o_speed = 10 o_angle = 0	o_speed = 10 o_angle = 0	Pass
2	i_dist_left >= 700 0 < i_dist_right < 700 i_dist_front >= 700	o_speed = 10 o_angle = 0	o_speed = 10 o_angle = 0	Pass
3	0 < i_dist_left < 700 0 < i_dist_right < 700 0 < i_dist_front < 700	o_speed = 10 o_angle = 90	o_speed = 10 o_angle = 90	Pass
4	i_dist_left >= 700 0 < i_dist_right < 700 0 < i_dist_front < 700	o_speed = 10 o_angle = 90	o_speed = 10 o_angle = 90	Pass

5	i_dist_left >= 700 i_dist_right >= 700 0 < i_dist_front < 700	o_speed = 10 o_angle = -90	o_speed = 10 o_angle = -90	Pass
6	0 < i_dist_left < 700 i_dist_right >= 700 0 < i_dist_front < 700	o_speed = 10 o_angle = -90	o_speed = 10 o_angle = -90	Pass
7	i_dist_left >= 700 i_dist_right >= 700 i_dist_front >= 700	o_speed=10 o_angle =-90	o_speed=10 o_angle =-90	Pass
8	0 < i_dist_left < 700 i_dist_right >= 700 i_dist_front >= 700	o_speed=10 o_angle =-90	o_speed=10 o_angle =-90	Pass

Analysis

The input variables i_dist_left, i_dist_right, and i_dist_front carry sensor readings (in cm). The Carrier is handheld and moved towards and away from the walls as it moved through the hallway. This ensures the full range of the sensor is tested given the width of the hallway. Several tests were performed such that the range of these sensors varied between 0-700 meaning the Carrier sees a wall and values greater than or equal to 700 which means the Carrier does not see a wall, i.e. a hallway exists in the direction where the sensor is pointing. For the output, o_speed, a value of 10 means moving forward, -10 moving backward, and 0 stopping. For the output, o_angle, 90 means turning 90 degrees leftward and -90 means turning 90 degrees rightward.

The test shows correct functionality on the Planner's side when it comes to controlling the Carrier. Note that these eight test cases are the only cases necessary for the right-hand rule algorithm to function properly.

POI

Summary

The role of POI module is to detect doors and provide the Communication Center module with information about the doors, such as room number and other tags, added with the location of that door. That information is further by the Visualizer to create points of interests on the 2D map. Since this module is a combination of hardware and software components, we will be performing both black box and white-box tests to verify and validate the design.

Test Cases

Validation

To ensure that the system or module is performing tasks that the user originally intended for, we need to check for the overall validity of the module. The following test case is used to determine if the system is valid. The test performed is used to validate the module as a whole rather than portions of the module.

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	Camera Feed	The module should detect doors, successfully execute OCR and send OCR data & current location to Communication Center via Ethernet	The module is able to detect the door and save the image. However, the module does not produce the right result using OCR.	Fail

All the separate units and functions that the module uses, they all execute successfully if tested separately. But at this early stage, integration of separate functionalities is bound to create problems and bugs. To improve the result, more testing and debugging is required to ensure that the module is valid.

Verification

To check if different units of the POI module are functioning correctly and fulfilling the requirements, we need to perform black box and white-box testing on those units.

Table 2: Verification Test cases for POI

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	Door in camera FOV	Convert camera feed into binary image (door appears as white)	Door appears as binary	Pass
2	Binary feed	Convert binary feed to contour feed	Converts to contour feed	Pass
3	Contour feed	Save contour image if contourArea>= 5000	Saves contour image when contourArea>= 5000	Pass
4	Saved Contour image	Execute OCR & return text file result	Executes OCR, not able to read result file	Fail

5	Location data from MMU	Save data from MMU to a stack	No interface between MMU and POI at this point	Fail
6	OCR and MMU data	Store both MMU and OCR data in a structure	Able to group dummy data. No real data to work with.	Fail
7	Communication Center hostname & port number	Initiate connection with Communication Center	Able to initiate connection	Pass
8	OCR and MMU data	Send data to Communication Center	Able to send dummy data. No real data to work with	Fail

Analysis

As we can see from the test cases above, most of the tests are failing only due to the fact that there is no real data to work with. Using dummy data to decide if a module is completely functional is an appropriate way to determine if the module or system is verified and valid. As progress is made, it will allow us to work with real data and be able to determine then if the system and module is verified.

MMU

Summary

The role of the MMU is to provide a map of the target environment while simultaneously estimating the pose of the robot in the target environment.

Test Cases

Validation

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	Estimate pose	The MMU should output the pose of the robot within the world, the pose should update as the robot traverses the target environment	The MMU is able to estimate the pose and it is able to update the robot traverses the target environment	Pass

Verification

Test ID	Test Case	Predicted Value	Test Value	Result (Pass/Fail)
1	Receive data from SIM	The MMU should receive data from the communication module correctly for all SIM data	The MMU correctly receives all SIM data through the communication module	Pass
2	Send data to Visualizer	The MMU should send data to the Visualizer through the communication module	The MMU sends data through to the Visualizer	Pass
3	Send data to POI	The POI should receive the pose at k_pose_update_freq as per SRS	The POI does not receive the pose data because the functionality is not yet implemented in the communication module	Fail
4	Send data to Planner	The Planner should provide the map to the Planner from any given mapped point	The Planner does not receive map data from the MMU because this functionality is not currently implemented in the MMU	Fail

Analysis

The MMU will be tested further once the communication module is fully implemented and the accuracy requirements of the map will be tested during that phase.

Communication Module

Summary

The communication module insures that different modules are connected to each other and that the can all communicate together through this central node. It also keeps the off-board server updated with all the required information that will be displayed to the end user.

Test Cases

Validation

1. To be able to communicate with the Arduino, this has both the Sensor interface module and the carrier, over the serial bus.
2. The commutation module is able to communicate with the offline modules over Wi-Fi. Offline modules are the command center and the visualizer.
3. The communication module is successfully communicating with both the MMU and the planner using shared memory.

Table 5: Test Cases for Communication Module

Test ID	Predicted Value	Test Value	Result
1	Being able to read and write on the serial bus	As predicted	Pass
2	Being able to witch between wifi networks and re-connect and reach main server and send communication data over wifi	As predicted	Pass
3	Being able to read and write to shared memory	As Predicted	Pass

Verification

1. The command center can buffer messages when communicating with the various modules as described in CM_MR_01
2. The command center is able to verify that any of the modules has gone offline.

Table 6: Test Cases for Communication Module

Test ID	Requirement	Predicted Value	Test Value	Result
1	CM_MR_01	When communication drops and reconnects the messages are not lost	As predicted	Pass
2	CM_MR_02	When a module is offline because of communication issues the module is able to identify the state	Functionality not implemented	Fail

Analysis

The implementation of the communication module is at 50% from the final state. The watchdog timer is the most safety critical part of the implementation that has not been completed on the count that we are still in the testing phase. Further development is expected to bring the implementation to its final revision to meet the project requirements within the expected timeline. No changes will need to be done to the design or requirements at of the module following the first round of testing.

Visualizer Module

Summary

The main purpose of the Visualizer module is to construct the visual map and to display the POI points on the map.
It will be getting data from the MMU and POI modules wirelessly while the vehicle is moving and mapping the area. The MMU data will consist of X and Y coordinates of the current position of the vehicle as well of the surrounding walls. The POI data will be tags describing the point of interest plus the X and Y coordinates of it.

Test Cases

Validation

1. To be able to communicate with the Vehicle and receive data feeds from MMU
2. To be able to communicate with the Vehicle and receive data feeds from POI.
3. Process X and Y data and generate visual representation of each point.
4. Being able to scale and pan generated map to facilitate large area mapping.

Table 7: Test Cases for Visualizer Module

Test ID	Predicted Value	Test Value	Result
1	receive a data feed from MMU	As predicted	Pass
2	receive a data feed from POI	No data received (POI not integrated yet)	Fail
3	Identifying the X and Y,and generate an 2D visual element.	As predicted	Pass
4	Generated Representation can accommodate any size of data	As Predicted	Pass

Verification

1. Visualizer shall be wirelessly connected to the vehicle via WiFi as in VSL_MR_01
2. Visualizer shall be off-board and run on a computer device (laptop) independent from the Vehicle as in VSL_MR_02
3. Visualizer shall convert point array sent from MMU to basic (x,y) coordinates to build 2D map and the base point on map shall be the lower left corner as in VSL_MR_03
4. Visualizer shall construct a visual 2D map using the points sent from the MMU.
   - o 2D map shall be (1cm : 1m) scale
   - o 2D map shall have indication (astrik) for poi identified.
   - o 2D map shall have clickable poi points to lead to further data on poi (name,data,image)

   As in VSL_MR_04
5. Visualizer shall report its status to the Command Centre module as in VSL_MR_05

Table 8: Test Cases for Visualizer Module

Test ID	Requirement	Predicted Value	Test Value	Result
1	VSL_MR_01	Vehicle can send MMU data wirelessly to Visualizer	As predicted	Pass
2	VSL_MR_02	Visualizer is run independent of the vehicle.	As predicted	Pass
3	VSL_MR_03	Process MMU data feed,extract X and Y coord and create points.	As predicted	Pass
4	VSL_MR_04	Process POI data and add an overlay over the generated map.	Functionality not implemented	Fail
5	VSL_MR_05	Visualizer sends debug data to Command Center periodically	Functionality not implemented	Fail

Analysis

The implementation of the Visualizer module is at 50% from the final state. Once POI is integrated further testing is possible on that aspect, further development is expected to bring the implementation to its final revision to meet the project requirements within the expected timeline. No changes will need to be done to the design or requirements at of the module following the first round of testing.

Command Center Module

Summary

The Command Center houses the GUI interface, in our implementation this represents the web front site. The Web front provides the ability to monitor the status of the various modules, the last known location of the vehicle, the visual map generated by the Visualizer module and an option to execute an Emergency Stop.

Test Cases

Validation

1. To be wirelessly connected to the vehicle and display vehicle status(offline/online).
2. Received data feeds from the various modules and display them on the screen.
3. Pushing the STOP button shuts down the vehicle.

Table 9: Test Cases for Command Center Module

Test ID	Predicted Value	Test Value	Result
1	Command Center shows that Vehicle is connected	As predicted	Pass
2	Display data from the vehicle modules	As predicted	Pass
3	Vehicle stops all functionality and shuts down	Functionality not implemented	Fail

Verification

1. Command center will be wirelessly connected to the vehicle via Wifi as in CC_MR_01
2. Command Center will be off-board, run from a computer device (laptop) independent from the indoor mapping vehicle as in CC_MR_02
3. Command center will have to display diagnostic information from the various modules, such as:
    - Current module state
    - Error Code signals.
    - Memory Usage

   As in CC_MR_03
4. Command Center will be able to send a KILL signal to the vehicle remotely, to stop its modules and movement at any point in time as in CC_MR_04

Table 10: Test Cases for Command Center Module

Test ID	Requirement	Predicted Value	Test Value	Result
1	CC_MR_01	Command Center shows that Vehicle is connected	As predicted	Pass
2	CC_MR_02	Command Center is run independent of the vehicle	As predicted	Pass
3	CC_MR_03	Display each module's diagnostic data on the webpage	Functionality not implemented	Fail
4	CC_MR_04	Vehicle shuts down upon pressing the STOP button	Functionality not implemented	Fail

Analysis

The implementation of the communication module is at 50% from the final state. Many parts of the Command Center are dependent on the other modules being at a mature enough level to send proper diagnostic data, as it is the infrastructure to receive the feeds is in place all that's left is configure the modules to send their data. The STOP signal feature is dependent on the Communication module completion. Further development is expected to bring the implementation to its final revision to meet the project requirements within the expected timeline. No changes will need to be done to the design or requirements at of the module following the first round of testing.