**About Using Pioneer 3-DX Robot to Achieve Navigation**

Enwei XU, Zhongjian SUN, Zihang WANG, Xuan YU and Chaoyi HU

(xuen, sunzh5, wangzi3, xuan, huch)@udmercy.edu

University of Detroit Mercy

***Abstract*-This paper will show you our group’s achievement of how to make our Pioneer. When the first time we got the Pioneer, we started making the plan about making the Robots. We started planning the plan in two directions. One is the hardware part, the other is the software port.**

I. INTRODUCTION

We made the plan first and then we finished our work depending on our schedule. As the hardware part, we need a platform to set our sensor machines such as the Hokuyo, Kinect and our microprocessor. This platform is so important because it can promise us that we can set up all the hardware on the robot in the same time, making sure they will not interfere with each other. We also need to solder the board ourselves and connect the wires ourselves. We installed a transformer for our robot while solving the battery allocation problem. We also need to solder the board ourselves and connect the wires ourselves. We installed a transformer for our robot while solving the battery allocation problem. We make the sensors and the microprocessor connect with the robot and make sure they can work successfully. As the software port, our group finish the G-mapping first to solve the problem that it can’t draw the map. We use the G-mapping to make the Pioneer use the Hokuyo to scan the environment surrounding the robot and then remember them in the Microprocessor, such as the obstacles. Then we finish the robot’s navigation. when we use the Navigation, we can give the robot a starting point and the ending point. And the robot will go form the starting point and then arrive the ending, avoiding the obstacles. Depending on the schedule, we should do the Remote-control system next. We use the TeamViewer this software to finish controlling the Microprocessor. We can use our computer to control the robot’s computer and we can also control the movement of the robotics and get the feedback from (the map of the surrounding environment) the microcomputer on our own computer. So now our group’s achievement is controlling the robots by the TeamViewer to draw the map and then give it the direction to finish the avoiding obstacles. This report not only shows our achievement but also show my groupmates’ ability. Without our spirit of grouping together, we can’t finish this achievement.

*1. The goals*

The goals of our project are to achieve the auto-navigation, obstacles avoiding with Hokuyo Laser sensor and color detecting with Kinect sensor. In order to achieve these goals, we learned the knowledge about the GMapping and the navigation from two pdf files.

*2. Hardware part*

1. Computer: Intel NUC (NUC6i7kyk)

1: Processor: Sixth Generation Intel ® Core ™ i7-6770HQ Processor with Intel® RTX Pro Graphics 2.6 - 3.5 Ghz Turbo, Quad, 6 MB Cache, 45W TD

2:RAM: Dual Channel DDR4-2133 + SODIMM 1.2 / 1.35V, maximum 32 GB.

3: Graphics: Intel® Turquoise Pro Pro 580 1 HDMI \* 2.0 (4K 60 Hz) 1 Mini DisplayPort \* 1.2 (4K 60 Hz) 1 DisplayPort \* 1.2 (via Type-C)

4: Audio: Up to 7.1 multichannel digital audio over HDMI or DisplayPort signals 3.5 mm front headphone jack, 3.5 mm rear speaker r / TOSLINK combo jack

5: Peripheral connectivity: Thunderbolt ™ 3 (40 Gbps) or USB 3.1 Gen2 (10 Gbps) via USB Type C connector ,2 front USB 3.0 (one of them is charged),2 rear USB 3.0, 2 internal USB 3.0 and 2 internal USB 2.0 via connector,Consumer Infrared Sensor, (Front Panel)

6: storage: 2 M.2 22x42 / 80 (key M) slots (for SATA3 or PCIe \* X4 Gen3 NVMe or AHCI SSDs), Support UHS-I SDXC card slot

7: The internet: Intel® I219-LM 10/100/1000 Mbps Ethernet ,Intel® Dual Band Wireless-AC 8260 Soldered Antenna (IEEE 802.11ac 2x2, Bluetooth \* 4.2, Internal Antenna, Intel® Wireless Display Technology 6.0)

8: Chassis: Metal and plastic with replaceable cover.

Volume: 211 mm x 116 mm x 28 mm

9: power supply: 19 V, 120 W ac - dc power adapter

10：Included in the box：No extra skull cover VESA \* mount 19-volt 120-watt power supply Integration Guide Processor logo

2.Pioneer P-3Dx robot

Function:

The main thing to achieve our goals of the project. We use it to simulate the self-drive car which can achieve obstacles avoiding and auto-navigation. The platform to carry the Hokuyo, Kinect and NUC. At the same time, also as the power supply for Hokuyo, Kinect and NUC.

Specifications:

Power

Run Time: 8-10 hours w/3 batteries (with no accessories)

Charge Time: 12 hours (standard) or 2.4 hours (optional high-capacity charger)

Available Power Supplies: 5 V @ 1.5 A switched, or 12 V @ 2.5 A switched

Batteries

Supports up to 3 at a time

Voltage: 12 V

Capacity: 7.2 Ah (each) Chemistry: lead acid Hot-swappable Batteries: Yes

*3. Hokuyo UST-10LX Scanning Laser Rangefinder*

Function:

Scan the environment and detect the obstacles. Hokuyo provides the main information for us to make the map and do the navigation work.

Specifications:

Supply voltage: 12VDC/24VDC

Supply current: 150mA or less (during start up 450mA is necessary.)

Accuracy: ±40mm

Repeated accuracy: σ< 30mm

Scan angle: 270°

Angular resolution: 0.25°

Detection range: 0.06m to 10m (white Kent sheet), 0.06m to 4m (diffuse reflectance 10%)

Max. detection distance: 30m

*4. Desktop sharing*

Because our NUC is fixed on the platform of Pioneer robot, so we cannot use the screen when robot is moving. So, to solve this problem, we need to use the desktop sharing function. So, in order to achieve this goal, we use TeamViewer, a desktop sharing software, which can help us achieve this goal. We download TeamViewer on our NUC and our own computer. And open both of them, there is a user-number and password on the NUC. And we use this number and password in our own computer. After the connection between NUC and our own computer, we can use our own computer to control the NUC and the Pioneer robot.

*5. Gmapping*

slam\_gmapping（Simultaneous Localization and Mapping） contains a wrapper around gmapping which provides SLAM capabilities.

Robot Slam\_gmapping can solve: At the same time positioning and map construction. the robot moves from an unknown position in an unknown environment, locates itself according to the position estimation and the map during the moving process, and simultaneously increases and establishes a map based on the positioning to realize independent positioning and navigation. [2]

Install the slam\_gmapping from the website by the follow command. **$ git clone https://github.com/ros-perception/slam\_gmapping.** Then change some data in the launch files, which are suit for the users. In the laser\_lms1xx.launch. The launch file includes some information: call the minimal.launch to get the odom data by RosAria. Using the hokuyo to get the laser data. tf is a package that lets the users keep track of multiple coordinate frames over time. [1] tf maintains the relationship between coordinate frames in a tree structure buffered in time, and lets the users transform points, vectors, etc between any two coordinate frames at any desired point in time. People can write some nodes about tf in this launch file to create the relationship between two coordinates.

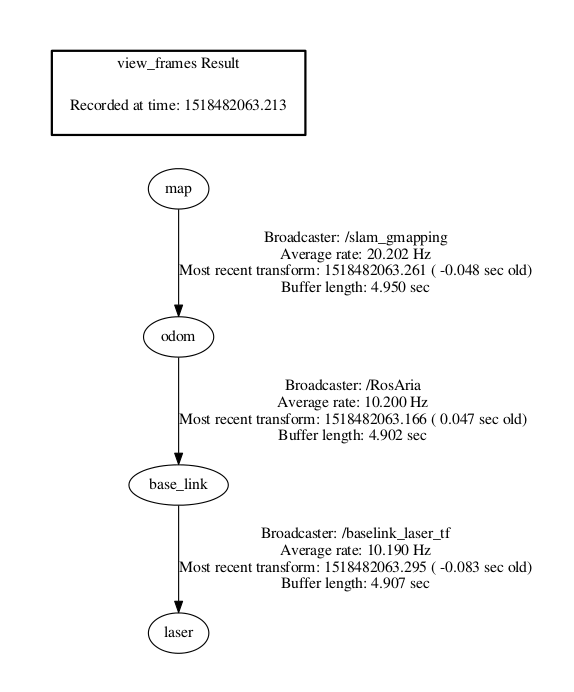


Fig.1. tf view

For example, the pioneer3DX-SH will create a local coordinate. The follow picture will show the coordinate.

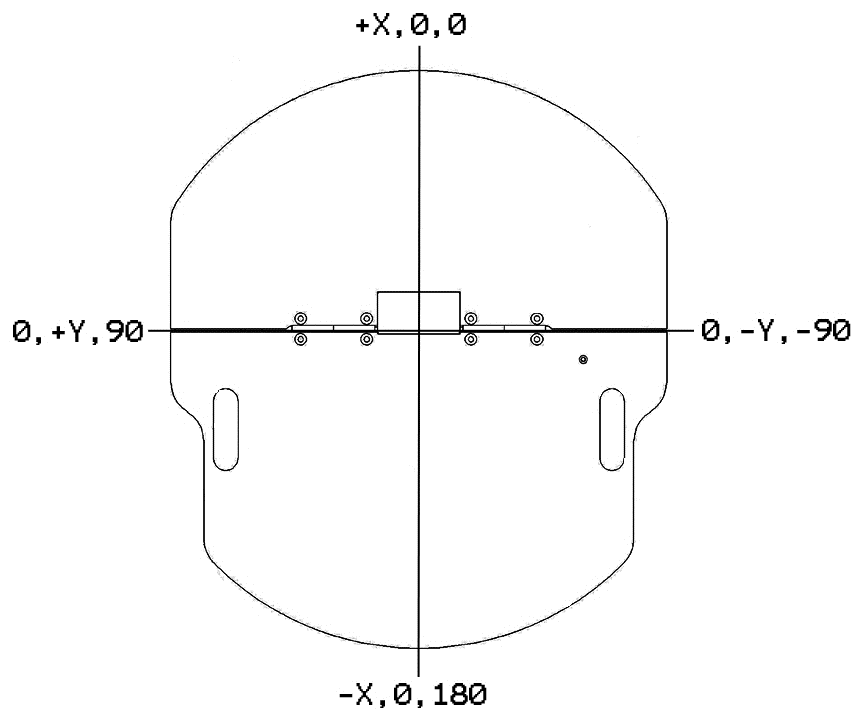


Fig.2. Internal coordinate system

The hokuyo will also send a coordinate. Connect these two coordinates by using the tf node and change the tf node according to the position of the hokuyo on the pioneer.

slam\_gmapping\_pr2.launch, this launch file can run the gmapping and have some parameters. Those parameters are used to change update frequency of the data and others also useful. People should experiment repeatedly to find the suitable parameters when use gmapping.

After installing the slam\_gmapping successful, run the follow commands in the terminals. First, open a new terminal. **$ roslaunch pioneer\_bringup laser\_lms1xx.launch.** This command is for connecting the pioneer3DX-SH to the computer.

Secondly, write the follow command in a new terminal. **$ roslaunch gmapping slam\_gmapping\_pr2.launch.** if the user run the gmapping successful, the terminal will be shown like the follow.

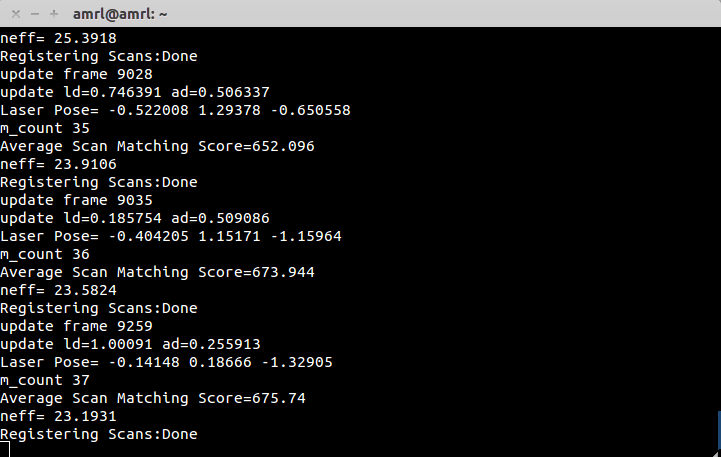


Fig.3. Receive the data by slam\_gmapping

Then open **rviz**. The follow is the map in the rviz. Create visualization by add map. If people wants to see the coordinates, add tf. The fixed frame chooses map in the global options.

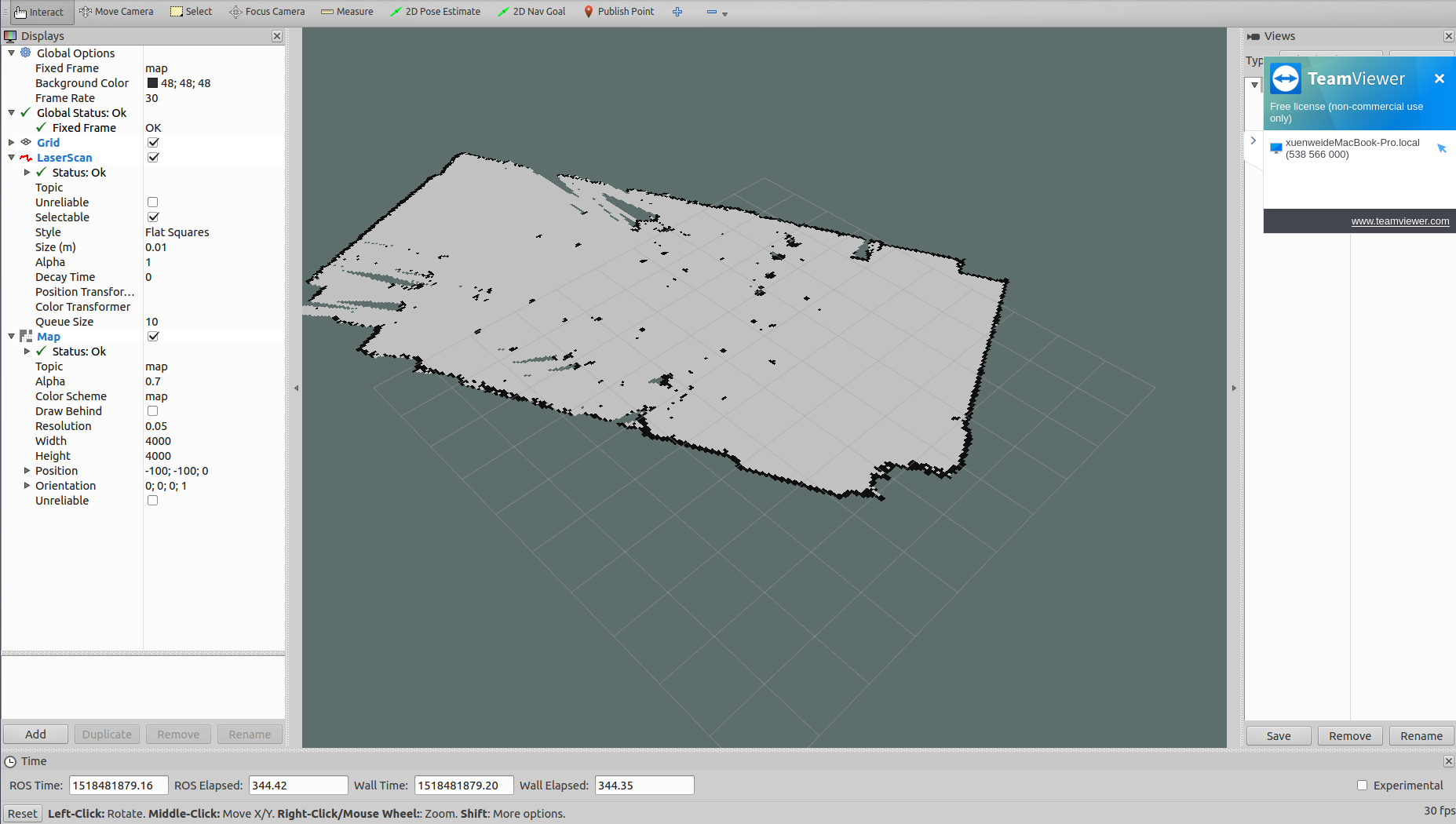


Fig.4. Rviz

After those steps, there is a part of the map in the rviz. The pioneer3DX-SH need to run to make a full map, so use the follow command in a new terminal to let the pioneer run.

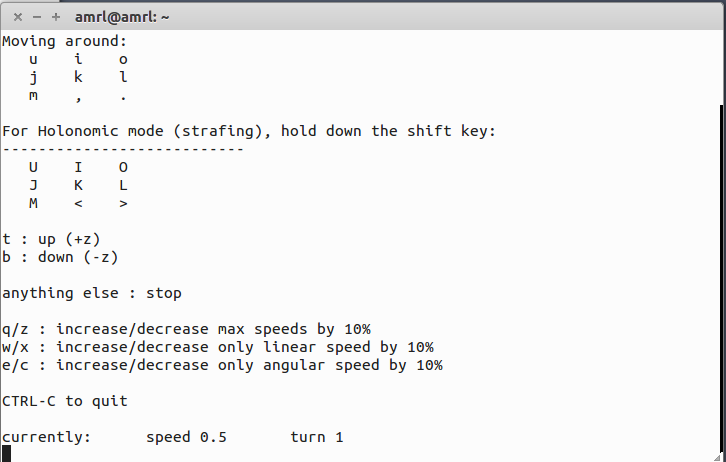
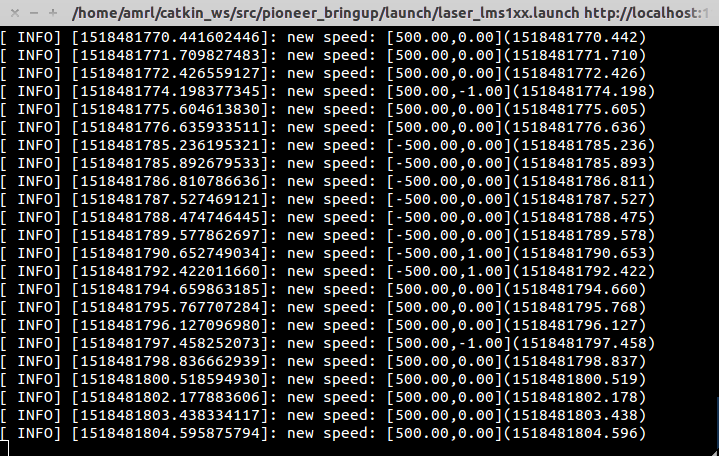
**$ rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py.** this command lets users use the keyboard to control the pioneer.

Fig.5. The running command in terminal

  
Fig.6 . Send the speed to the pioneer3DX-SH

After scanning the full map, use the follow command to save the map. **$ rosrun map\_server map\_saver -f <map\_name>**

The gmapping is finish.

*6. Navigation*

Navigation package is used to calculate the velocity for the pioneer3DX to go to the destination. Input the odometry, lidar data, transfer information and map which is scanned in advance. Then the pioneer3DX can get the cmd\_vel velocity by using navigation package.

the pioneer3DX can get the map of the environment which was created. First, the pioneer3DX should get the workspace which was created and initialized named “catkin\_ws”, then clone the pioneer3DX model to source directory by executing the below command.

**$ git clone https://github.com/LCAD-UFES/PioneerModel.git**

Once the clone is completed, proceed with the catkin\_make by executing the below commands. Then it is about the installation of navigation package for our robot. Then git clone the pioneer navigation package to the source directory by executing the below command.

**$ git clone https://github.com/LCAD-UFES/p3dx\_navigation.git**

Once the clone is completed, proceed with the catkin\_make by executing the below commands. Then git clone the pioneer navigation package to the source directory by executing the below command.

**$ git clone https://github.com/amor-ros-pkg/rosaria.git**

The next step for pioneer3DX is to update some of the launch files, with the details of the hardware. For updating the launch file contents, open the file named “pioneer.launch”, which is under /home/amrl/catkin\_ws/src/p3dx\_navigtion/launch. Inside this file, the below listed modification should be performed. The port address should be updated the ttyACM0 as well as the LIDAR details (LMS 1xx) in the file.

In the “move\_base\_rosaria.launch”, update the path of the map. Then this launch file can use the map, which was made by gmapping.

It can be seen that the “.yaml” files under /home/amrl/catkin\_ws/src/p3dx\_navigation/config/p3dx\_rosaria. There are some parameters in those files, some need to be changed depending on environment.

When the navigation package is used, the follow three commands need to be executed in three different terminals.

**$ roslaunch p3dx\_navigation pioneer.launch**

This command connects the pioneer3-DX and LIDAR to the computer.

**$ roslaunch p3dx\_navigation rviz\_p3dx.launch**

This command will open a rviz.

**$ roslaunch p3dx\_navigation move\_base\_rosaria.launch**

This command is used for uploading the map in the rviz.

II. PROBLEM AND SOLUTION

*Prlblem1*

Steps of set up the microchips on the NUC.

First, you should use the screwdriver to screw off the nails on the back of the NUC. When you finish that you can put the first two chips in to the computer. Be careful the way of setting up the chips. You should let the chips down first, then the chips are connecting with the connecter in the computer then use force to push the chips in to the connector. The second one is the same way.

Then we should set up the two SSD. We have two of them. One is the M2\_B(80mm), the other is the M2\_A(42mm). when we set up the M2\_B, we need to screw off the nails on the setting board and then push the chips inside to the connector .and we put the nails back to the board to fix the M2\_B. the M2\_A is the same way to set up. But be careful the positions of them.

After we finish all the chips setting up, we can just put the computer’s back cover back.

Then the computer setting up is finish. You can just put the plug and the cable connect to the computer, connecting the screen on the computer. the you can just be setting the computer.

*Problem2*

Wi-Fi problem

Problem2: At first, we connected our Hokuyo to the NUC, but we found that we cannot use Wi-Fi if we connect the Hokuyo. Because we do not set that we can use both Wi-Fi and Ethernet at the same time.

Solution: So, firstly search for Network Connections in the unity dash. Then, under the Ethernet section, click Wired connection 1. Under general section, we change ‘Automatically connect to this network when it is available’ to ‘All users may connect to this network’. Then go to the IPv4/IPv6 settings and then click on the Routes button. Check the ‘Use this connection only for resources on its network’ option. Click Save. After doing all these steps, we can use both the LAN and Wi-Fi simultaneously. But after we connect Hokuyo to the NUC, we need to click the ‘Wired connection 1’to make it work.

*Problem3*

using slam\_gmapping\_pr2.launch to run the gmapping. But the data updating was not corrected. Data updating was continuous when the pioneer stopped.

Solution: add <rosparam> in this launch file.

*Problem4*

Using RosAria to let the pioneer3DX-SH run. But the RosAria was used in the laser\_lms1xx.launch, so this time cannot use RosAria to let the pioneer3DX-SH run.

Solution: Using new commands. **$ sudo apt-get install ros-indigo-teleop-twist-keyboard.** This command is used for installing the software, which is able to use the keyboard to make the pioneer3DX-SH run. **$ rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py.** Runningthis command allows people to use the keyboard to let the pioneer run.

*Problem5*

During creating the map, the odom coordinate usually jumps in the map. The pioneer3DX-SH didn’t know where it was, so the map was wrong.

Solution: because there are two odom data outputs. Deleted one odom node in the laser\_lms1xx.launch. The problem was solved.

*Problem6*

When we did the navigation, after cd catkin\_ws folder, we should run the “catkin make” command, however the system could not find a package configuration file provided by “controller\_manager”.

Solution: This problem proved that our system did not have the controller\_manager package installed. As ros\_control is not yet release for Jade, we should clone the ros\_control repository with “git clone https://github.com/ros-controls/ros\_control.git -b jade\_devel”.

III. CONCLUSION

In the past, our group learned a lot and we had done some works about our project. First, with the help of the GTA, we successfully assembled our minicomputer. After our minicomputer can be used, the next thing we had done was that we installed Ubuntu system in our computer. And then, we installed ROS, Matlab and Gazebo based on Ubuntu system. We connected the Pioneer 3 robot with our minicomputer and we can control the Pioneer 3 robot preliminarily by using the keyboard. Second, in order to design the holder of our computer and some sensors, we measured the size of Pioneer 3. At the same time, one of our group members finished the work of welding the Hokuyo sensor. Third, we finished how a ROS-based system is used with a Pioneer 3-DX robot for indoor mapping (G Mapping), which is depending mainly on LIDAR and odometer for acquiring data from environment for constructing the map. Fourthly, we finished setup and navigate Pioneer P3Dx robot using the navigation package. Now we are able to setup on our pioneer robot will make the robot capable of publishing coordinate frame information, receiving messages from all sensors that are to be used with the navigation stack, and publishing odometry information, while also taking in velocity commands to send to the base.

We still have a lot of works to do for our project in the future. For example, we need to make our robot identify the color. We will continue to work hard!

REFERENCES

[1] No author. “Setting up your robot using tf” wiki.ros.org/navigation/Tutorials/RobotSetup/TF.

[2] Rehatbil J N., “SLAM algorithm applied to robotics assistance for navigation in unknown environments”,