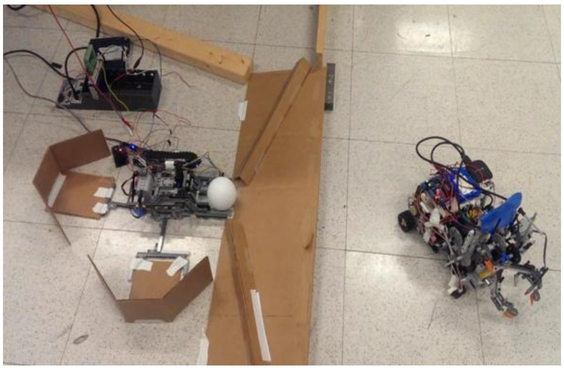
Final Report of EEC 255

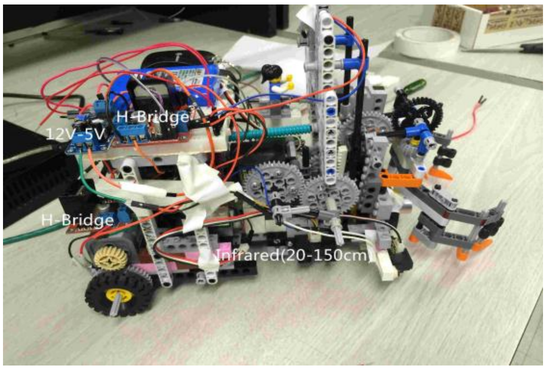
Leader: Zhiwei Zhao
Group Member: Nathan Wong, Haolun Yan, Mingjie Zhang
Project: Robotic System
Function: Collecting and Sorting Trashes

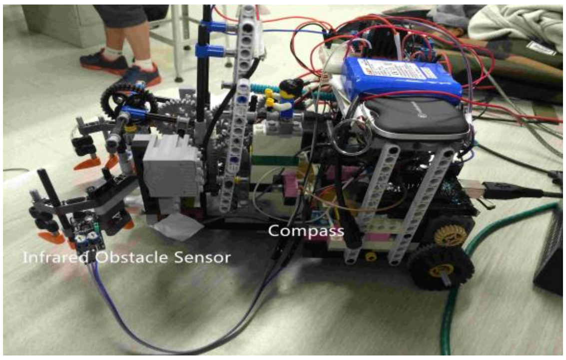
Brief introduction: In this project, as the requirement, we divide this system into two parts, stationary robot and mobile robot. We define a particular workspace. The trashes, which is designed as the color balls, are placed in this space randomly. The mobile robot start searching in the workspace, then get the color balls back. Finally, we use the stationary robot to sort the balls based on different colors. We separate the project into four parts, and each group member is responsible for one specific part. We also discuss and cooperate with each other to finish the project.

Components: Arduino UNO *2, Futaba Servo *2, Lego motor *4, L298N H-Bridge *3, 12V-5V transformer *1, Sharp Infrared Sensor *2, Ultrasonic HC-SR04 Ranging Detector *1, HMC5883L Electronic Compass *1, ZITRADES Infrared Obstacle Avoidance Sensor *1, TCS230 Color Sensor *1, 12V Battery *1, Lego Blocks, Wires.

Photos for the System:







Individual work: Zhiwei Zhao is responsible for the programming algorithm and majority part of the code. Nathan Wong is responsible for the mechanical design. Haolun Yan is responsible for the sensors, circuits, and minority part of the code. Mingjie Zhang is responsible for the LabVIEW part of the stationary robot.

1. Programming Algorithm

A) Calibration

For the compass, we set the car to rotate more than one circle, get data from compass every 10ms. So we will get 2000 data for all, and pick out the maximum and minimal ones to calculate and store as the offset. The angle of our car is calculated using the function: angle = atan2(scaled.YAxis - Y_offset, scaled.XAxis - X_offset).

B) Distance Detecting

We have three sensors with different ranges. When reading the values, we guarantee that they are all in range. The distance detecting function can get rid of the out range values, and do an average on the valid values. If there are several times out of range, then the function will set a big value to tell the microprocessor that the object is out of range.

C) Vertical Surface Getting

Since the workspace is surrounded by walls, we can get accurate 90 degree in our axises. In this part, the function reads the distance to walls and finds the smallest value. When the distance is larger than last one, it will inverse the car's turning direction. When it gets two inversions, we can ensure that this is the vertical angle. Here we design a filtering function to filter out the objects. Since the change of the distance to the walls should be continuously. If there is a jump value, then it could be ignored.

D) Searching

The searching part is divided into two parts. The first part is searching object in long distance, while the second is searching in near field, and finally getting the ball.

In the first part of searching, the robot sweep from left to right with firstly turning to left without scanning any values. The function reads the angle from compass before scanning and stores it. Then it scans and records the distance values of the front infrared sensor (it will be introduced in the Sensors and Circuits part). Only if the values jump from bigger to smaller then to bigger, it ensures that there is an object. After all the scanning processes, the function will find the one with smallest distance and its angle. If the distance is larger than 75cm, the robot will turn to the original angle after scanning; if it is smaller than 75cm but larger than 30cm, the robot will turn to this angle. In these two conditions, the robot will move forward a little after turning to the angle. Then if it is smaller than 30cm, the first searching part will be broken. Also the robot will read the distance in the beginning of the loop before scanning. If it is smaller than 60cm of ultrasonic sensor but larger than 30cm, and the value of the front infrared sensor is smaller than ultrasonic's, the robot will move forward.

In the second part, we first make the robot thinks it is in the 30cm range of the object, but it may out of the angle of ultrasonic sensor. It will scan and adjust to the position, in which the value of infrared sensor is smaller than that of ultrasonic. Then move forward. After the ultrasonic sensor finds our object is inside 10cm range, we break the process and go to grab the ball.

E) Dropping Detecting

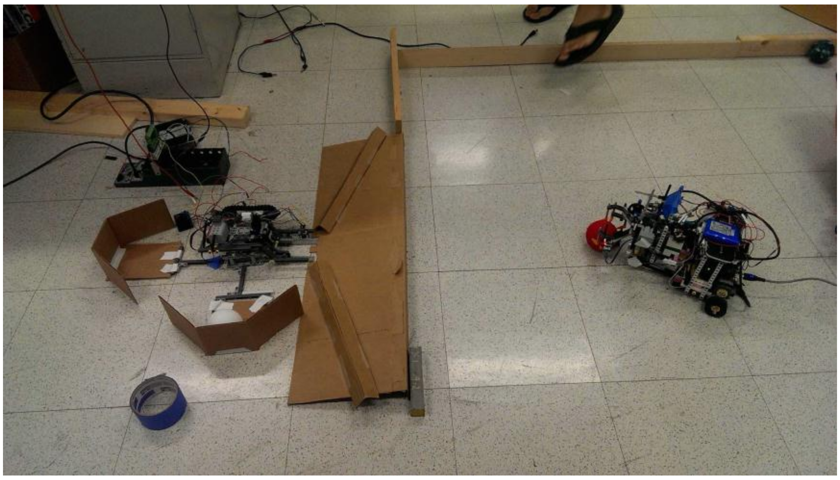
By using ZITRADES Infrared Obstacle Avoidance Sensor on the grabber, the function can read whether there is an object in the grabber. If not, move back and start searching.

F) Navigation

We firstly consider using the compass to find the direction of the stationary robot. However, the compass will be influenced by the magnetic field. Since we know the position of the stationary one, here the function uses the ultrasonic sensor and another infrared sensor on the right side of the car to detect 4 vertical surfaces of the workspace. Then the position of the car can be determined. In this way, the direction to the station can be computed. Since the error of the compass, after moving toward the determined direction for some time. It will get the vertical surface again to calibrate the direction. Finally the ultrasonic will sense the station and drop the ball.

2. Mechanical Design

For the mechanical design of both the mobile robot and stationary robot, Legos were used to create most of the structural components. Legos are a great do-it-yourself learning tool for building hobby-type projects, especially for robots, as Lego offers a robotics kit in the Lego Mindstorms product line.

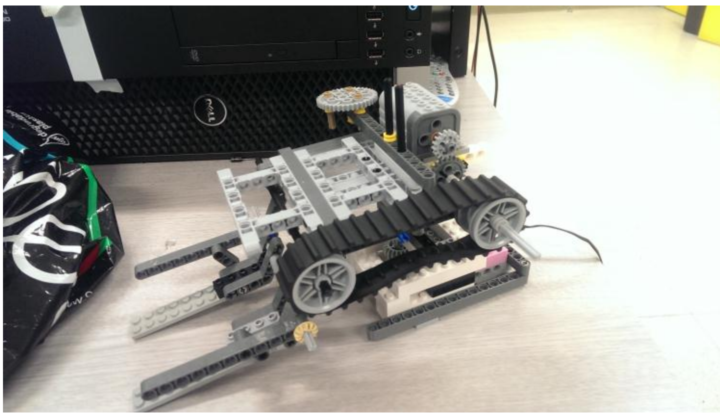


Mobile Robot and Stationary Robot

Unfortunately, the Lego kits provided by the lab for the EEC 255 course were incomplete and shared among the whole class. This led to an inadequate amount of parts in the beginning of the quarter and made it difficult to go through a formal design process. We were limited to the parts available and under deadline to make the first prototypes of an end effector. Furthermore, we were unsure when new parts (lab ordered some more Lego kits for the sake of the class) would be available to build the rest of the bots. For these reasons, we did not design and build with a fully tested final robot design in mind and ran into some mechanical problems later in the project, but we were unable to remake the entire robot due to time constraints. These problems were dealt with, but obviously it would have been helpful had we designed our robot appropriately to avoid them in the first place. These mechanical design obstacles are discussed in further detail below.

Stationary Robot

The mechanical design of the stationary robot was quite simple. We used a Futaba servomotor to rotate a stage on which had a Lego Mindstorms DC motor used to lift our objects. Once the object was placed in the lift, the DC motor was able to lift the arms and

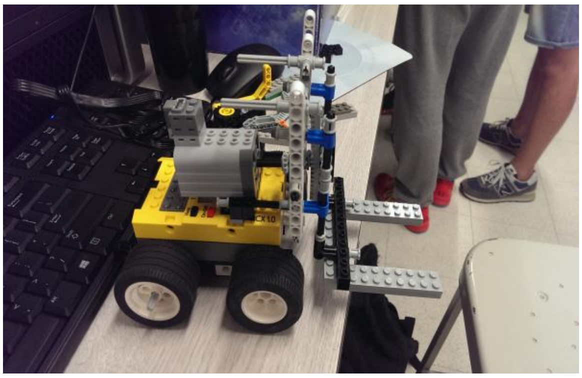


Early Stages of Stationary Robot

object, and the servomotor could rotate the stage to allow the object to be dropped into different containers. The containers were boxes made of cardboard. The motors and stand structure were all mounted using Lego pieces from the
Mindstorms Kits. The most difficult part of building the stationary robot was figuring out how to mount the servomotor, which was not part of a Lego set. In the end, we were able to use bolts and nuts to secure the servomotor to a Lego beam and attach it to the Lego structure. The rotating shaft of the servomotor was attached to the stage for rotation by using Velcro. Velcro is a super useful mechanism for attachment and detachment for hobbyist robotics projects like this. The stand was then able to rotate 360 degrees to drop the objects in any location around it, and the DC motor was blocked by physical stops to hold the arms in the lifted position while the stand could rotate and transport the objects.

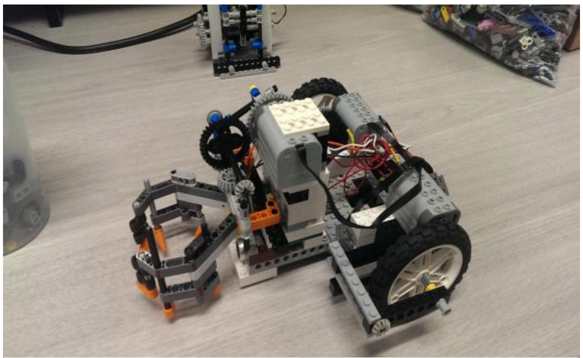
Mobile Robot

The mobile robot was a more difficult and time consuming build. We started the mobile robot design by building two prototype end effectors as required by our project deadline. One end effector was a claw with two opposing sides to grab objects, and the other end effector was designed to be a forklift. At this point, we still were having trouble figuring out what materials and equipment were available for us to use.



Forklift Prototype

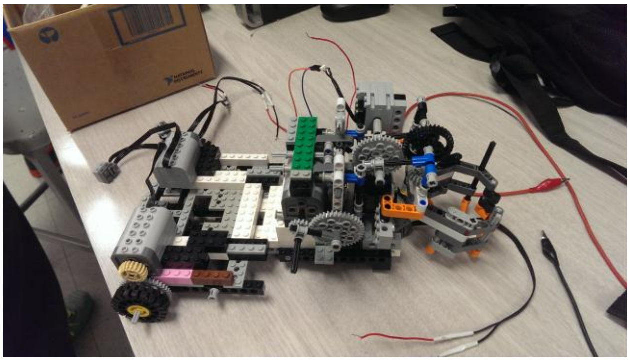
After creating two end effectors, we decided to use the gripping claw as our primary end effector to pick up spherical objects. The forklift prototype did not go to waste though. We ended up using the forklift prong concept as the arm for our stationary robot (shown above) and used the lift concept of the forklift on our mobile robot. Next we built our mobile robot chassis to place our gripper on the front. We decided on a two-wheel robot, with a castor for support. We bought a Lego designed castor for easy attachment to our Lego parts. The chassis was made out of Lego blocks, for easy attachment and build of Lego wheels driven by two Lego DC motors. Design with two DC motors allowed for differential drive to pivot the robot for tight turns and best maneuverability.



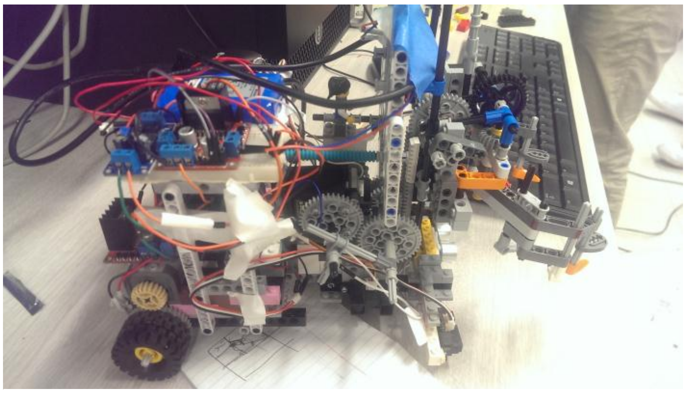
Two Wheel Simple Mobile Robot for Mobile Robot Demo

At this stage, we were able to attach an ultrasonic sensor to the front (by enclosing it in more Lego bricks) so that the robot could find, move towards, and pick up objects. This was our robot used for our mobile robot demonstration. We started to find that the robot has difficult driving smoothly and tried different Lego wheels until finally settling on some small solid rubber wheels that seemed to get the best traction. This proved to still be a problem in our final design as the rubber wheels easily attracted dust from the dirty lab floor and often slipped instead of driving. The final robot design could have been better at driving and this would have helped our final demonstration run better. Improvements could have been made by using better wheels or switching to treads.

After our first mobile robot demonstration, we realized that a requirement of the robot project was to lift the object. With our current design, we only had a gripper at ground level that could grab objects but did not have a vertical lift feature. We took the forklift design and made modifications to add support to be able to hold and lift the gripper and gripper actuator. This also required changing to a smaller motor for the gripper actuator. For driving the lift, we used another servomotor, which again proved difficult to mount within Lego parts. We were able to use bolts to attach the servomotor to smaller Lego attachments and use Lego shafts connected to those attachments to hold the motor in place. Alas, this structure did not fit perfectly and often needed manual adjustment before finding the right gear placement. In the end though, the servomotor drove the lift function of the gripper pretty well without any problems. The servomotor traveled 180 degrees to set two gripper positions, the up position and the down position.


Mobile Robot with Enlarged Chassis

Lastly, we used Lego beams to create a double decker feature on the back of the robot chassis so that we could place our battery and other electronic boards. We also had to continue tinkering with the robot to find best ways to attach sensors and electronic equipment. We mostly used small bolts and double-sided foam tape to attach our sensors. We used electric tape to organize the wires and keep them out of the way of moving parts.


Final Mobile Robot Design

Accessories

In order to complete the full robot demonstration, we needed to create/build some other features. First we made walls to define our mobile robot search field. The walls were wooden 2-by-4 beams and planks placed in a 5-foot by 7-foot rectangle to create the boundaries of the mobile robot workspace. We also built a ramp for the mobile robot to lift and place the sphere objects (colored foam balls) onto, which would funnel the balls into the carrier of the stationary robot. The ramp was made of cardboard with cardboard walls to direct the balls. It was designed to account for any error in the mobile robot's ability to find the stationary robot's position and place the ball in the exact location of the stationary robot's carrier.

3. Sensors and Circuits

A) Sensors for the Stationary Robot

For the stationary robot, we use the TCS230 color sensor to detect the color of each ball, and send the signals to the LabVIEW part. Since we find that the sampling frequency of the LabVIEW cannot meet the requirement for our color sensor, here we choose to use another Arduino UNO board. We firstly build a function to configure the color sensor, since it is necessary to set a white balance process to make the sensor distinguish the colors more accurately. Then the function can get the color values from TCS230. After that, the microprocessor will send the signals to the LabVIEW part. Here we have to use an external interrupt to control the frequency, so we import an additional library TimerOne to our environment.

B) Sensors for the Mobile Robot

Here we divide the sensors based on the phases of our robot's motions.

1) Calibration
When running the calibration function, we choose to use the HMC5883L electronic compass. For this sensor, we need to import another HMC5883L library into the environment. This sensor can detect the degrees from 0 to 360, but it cannot be accurate unless finishing the calibration and set the offset. Unfortunately, the magnetic field of the lab cannot change in a well-distributed way. So we can only guarantee the compass detect the accurate angles within three blocks of the lab's floor, and the relatively accurate angles in a same position.

2) Searching and Grabbing
We use one Sharp infrared sensor with 10-80cm detect range, and one HC-SR04 ultrasonic sensor with 2-500cm detect range. We combine them to increase the accuracy of searching process, based on three-point locating method. These two sensors can send the distance signals out without any additional libraries. So we should only supply the power and read the values. For the infrared one, we find that after computing the distance based on the formula 4800/(sensor_value-20), the value can be accurate enough. So the long-distance searching should be based on this sensor. But for the short-distance, we have to use the ultrasonic one, which can detect 2cm distance. That is reason for dividing the searching phase to two parts, as the algorithm above.

3) Navigation
When navigating to station, we use another Sharp infrared sensor with 20-150cm detect range with the ultrasonic. Based on the vertical surface getting algorithm, we read the values from the two sensors at the same time. After four times of vertical surface getting, we will get 8 values. In that way, the function can guarantee the accurate values. Then compute the direction to the station.

4) Searching the Next Object
After dropping the object into the stationary robot, the mobile robot will turn 180 degree, and start the next searching and grabbing function. Here the 180 degree is based on compass. As we mentioned above, after calibration, the compass can be accurate within the 3 blocks of the lab's floor. The stationary robot can be in that range, since we run the calibration just near that.

5) Dropping Detection
ZITRADES Infrared Obstacle Avoidance Sensor can send the infrared light, and receive the return light to sense the object, but it cannot detect the distance. We should set the valid distance manually. If the object is in the valid distance, the out pin will send a LOW signal out, otherwise a HIGH signal.

C) Circuits

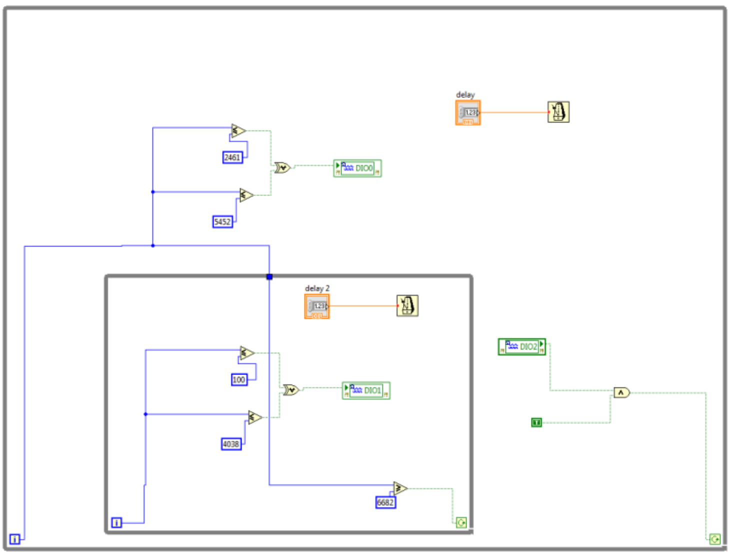
1) Power Supply for Motors
The Lego DC motors are placed for the wheels, and the grabbers of both mobile and stationary ones.

They need 12 volts power supply, so we just use the 12V battery to supply them directly. Here we use three H-Bridge Driver to get the signals from the microprocessor or FPGA. H-Bridge part can control the direction of the motors based on the import logics. The whole power supply system guarantees the moving of the mobile robot and the grabber’s putting up and down of the stationary robot.

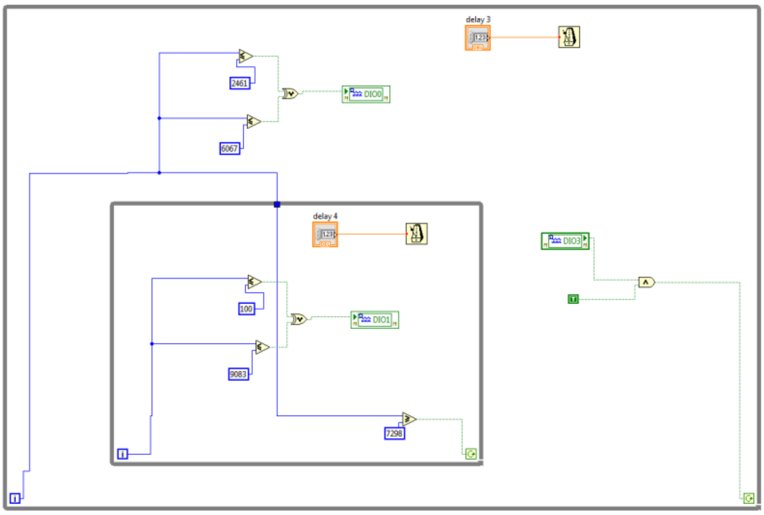
2) Power Supply for Servos
For the servo of the stationary robot, the FPGA can output 5 volts and a PWM signal to drive and control the servo accurately. For the servo of the mobile robot, Arduino cannot supply too big currency, since we have so many sensors. To guarantee the performance of the servo, we choose to use a 12V-5V transformer, which linked with the battery. In that way, the sensors and servo can get enough power supply for sure.

3) Power Supply for Sensors
All the sensors except ultrasonic sensor can perform well with the 5 volts from Arduino. However, we find that the ultrasonic need a stable 5 volts power to guarantee its performance. So we choose to do the same method as above by using a 12V-5V transformer.

4. LabVIEW for Stationary Robot
LabVIEW software is ideal for any measurement or control system, and the heart of the NI design platform. Honestly speaking, LabVIEW is new for our group. It takes us several weeks to learn some basic knowledge of LabVIEW. LabVIEW is used in the stationary part. If the color sensor senses the white ball, the port DI02 receives a high signal, then it runs the project below. The stationary part lifts up the white ball and rotates. PWM is created to control the rotation. When rotating it clockwise by particular degree, the white ball is dropped. Then the stationary part goes back to the original point.



Also, if the color sensor senses the red ball, the port DI03 receives a high signal, then it runs the project below. The stationary part lifts up the red ball and rotates. When rotating it clockwise by another degree, the red ball is dropped. Then the stationary part goes back to the original point.



For some dark balls, which we choose to reject them, the color sensor still send a signal. But this time, the stationary robot will do no any motions. If it is necessary, we can also make the grabber lift and down without rotation, to show the rejection behavior.

The sorting is done successfully.

5. The Complete Process, Advantages and Limitations

A) Complete Process

After mobile and stationary robots are ready, mobile one will first calibrate for the compass. Then it finds a vertical surface to set a standard. Mobile robot turn left for 90 degree, then start to searching. It will scan the whole workspace to find a nearest object, then go and grab it. Based on vertical surface getting, the mobile robot will get the four vertical surface, sense the distances to determine the location of itself. Furthermore, it computes the direction to the station, and uses the compass to go. Finally, it will get the vertical surface of the station, and go and drop the ball. Color sensor can sense that ball with a RGB value. The LabVIEW part will receive that and do the reaction. The mobile robot then go to search and get the other ball.

B) Advantages and Limitations

Based on our system, our mobile robot can search and navigate for the workspace as large as 2meters x 5meters, the limitation is from the detect range of the sensors. The navigation part can guarantee the accurate location of the mobile robot, and the accurate direction from the mobile to the station. That means we do not need to remember the action of the mobile, or set a mark, or turn 180 degree directly to get the position of the station. But the robot can adjust to get the direction based on the environment.

The limitations of our system are first the sensors have the range to detect the object, and second the mobile robot can be heavy, which lead the speed cannot be fast enough.

6. Conclusion

We finished the project, and meet the requirement. The mobile robot and stationary robot can cooperate well. Everyone in the group did a good job for their parts. And also we make efforts and help with each other. That is the reason for accomplishing of the project. All of us have done their best, and we cooperate with each other with assisting others' parts. For all I am satisfied with our group's work.