

Dock-Bot

Final Report

Huo, Ke
2011/4/19



Instructors: Dr. A. Antonio Arroyo
Dr. Eric M. Schwartz

University of Florida
Department of Electrical and Computer Engineering
EEL 4665/5666
Intelligent Machines Design Laboratory

Content

Abstract	1
Executive Summary.....	2
Introduction.....	3
Integrated System.....	3
Platform.....	4
Actuation	5
Sensors	5
Hardware and operation	5
Software and algorithm.....	7
Communication	8
Behaviors	8
Experimental layout and results	9
Conclusions.....	10
Documentation	10
Appendices	11
Code for IR beacon	11
Code for image process	12
Code for xbee and serial communication.....	12

Abstract

My project involves two mobile robots is designed to simulate the docking process. The chaser robot Dock-Bot has the capability to detect the base robot, locate its position, and move over to the base and avoid the obstacles on the route. By using sonar and camera which is used to recognize characters set on the base, Dock-Bot will find the right dock position and move toward to it. There are also bump sensors help the Dock-Bot finish the dock. Once finishing the docking, Dock-Bot is planned to act as a universal wheel of base robot. Project involves communication between two robot and laptop, which will be implemented by x-Bee and BAM module.

Executive Summary

This project is designed to simulate the autonomous rendezvous and docking of space crafts. So there is no third part navigation system like several external cameras which can be used to calculate the coordinates and attitude of each robot. By using two sets of special sensors: IR beacons and wireless camera, a complete system includes approximate/large range approaching, recognition in a short distance and position/attitude discerning has been established.

IR beacons are easy to use. But since it is too sensitive, I have to find a proper algorithm to calculate how much my dock-bot should turn. The algorithm I used is composition of vector. Four receivers stand for four directions which are vertical to each other. Specifically, north, east, south and west are expressed as $[0, 1]$, $[1, 0]$, $[0, -1]$ and $[-1, 0]$ respectively. I Also because of its sensitivity to the environment, sampling and weighting are applied.

Three different methods are designed and tried in the target recognition and position/attitude discerning part. First, I want to use three points with different colors on a line. By calculating the slope of this line, I can tell which side dock-bot is on the base robot. Second, I want to use three painted ping-pang balls lies in one line. Both these two plans did not work very well. Then, the final plan placing three hemisphere objects as a triangle jumped into my mind. And it turned out to be the best one.

During working one the image process part, I found a useful library in OpenCV which helped me a lot to calculate filter blobs, calculate the area of blobs. I guess this is the most charming advantage of open source software, you always can find many surprising tools from internet for your own project.

As well, since I am using two robots and camera in my project, I need to handle communications between two robots and my laptop. I am using PVR2 board as the controller of Dock-Bot. Both iRobot Create and PVR2 will be programmed to communicate with my laptop, used as upper controller. For the wireless part, I am using Bluetooth module BAM on iRobot create, X-bee on PVR2. Wireless micro-cam RC410 +203 and receiver connected with laptop will be the vision part. The laptop processes the image and sends commands to both two robots. After all these, the hardest part in my project had been solved. Then I begin to integrate every sub system work together. In the first place, I want to use ask and answer pattern in the communication between my laptop and dock-bot, but the results were not perfect, it always took long time to receive command. So I change it to an easier way, which is sending and listening always.

Introduction

Four steps as following shows can briefly cover the project:

- Locate the base and approximate alignment.

Base robot will detect the direction of dock-bot by using IR beacon, and point to it.

- Move toward base, avoid obstacles and recognize base.

Dock-bot, guided by IR beacon, move toward base. In this movement, if any obstacle appears, dock-bot has the capability to tell whether this is the base, if it is not, avoid this obstacle and continue move toward the base; if it is, go into next step. Camera, sonar, IR beacon and also IR range sensors are used.

- Position attitude adjust and docking.

Once dock-bot recognize the base, by using camera, dock-bot can tell which side it is on the base and whether the center of it is pointing to center of base. After adjustment, dock-bot will complete the docking conducted by bumper sensors.

Act as one robot.

4 servos are going to be used on dock-bot. As well, sonar, IR beacon, IR range sensors, wireless camera and bump sensors contributes the sensing system of the robot.

Integrated System

This project is designed to use two robots, the base robot, which is based on iRobot Create, and chaser robot, Dock-Bot. I am going to use PVR2 board as the controller of Dock-Bot. Both iRobot Create and PVR2 will be programmed to communicate with my laptop, used as upper controller. For the wireless part, I am going to use Bluetooth module BAM on iRobot create, X-bee on PVR2. Wireless micro-cam RC410 +203 and receiver connected with laptop will be the vision part. The laptop processes the image and sends commands to both two robots. A brief summary is showed in the Figure 1.

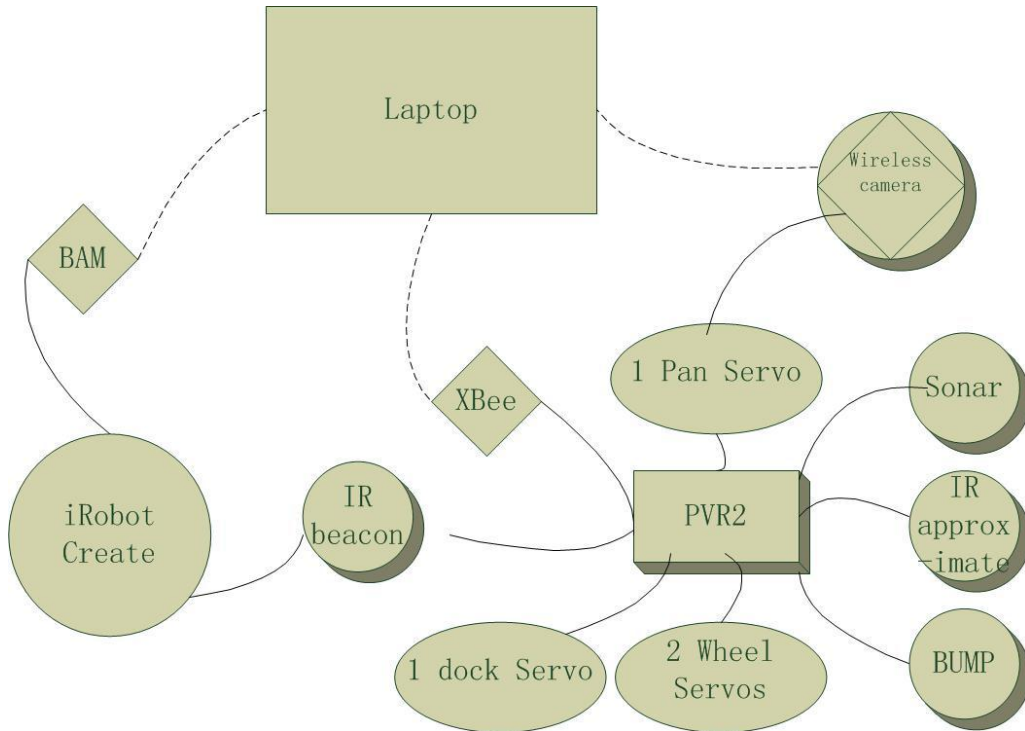


Figure 1: Integral system

Platform

iRobot Create (Figure 2) which owned by my lab will be the base robot's platform. Another three wheels platform (Figure 3) also from lab will be Dock-Bot's mobile platform.

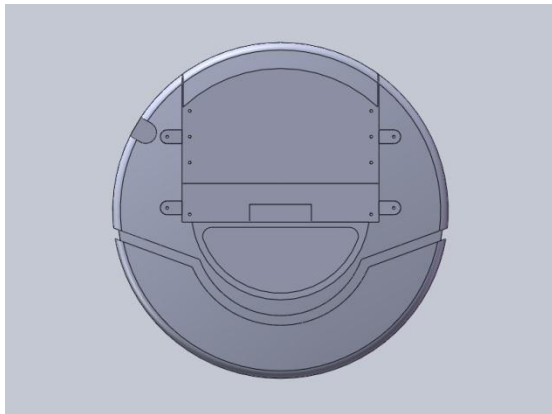


Figure 2: iRobot

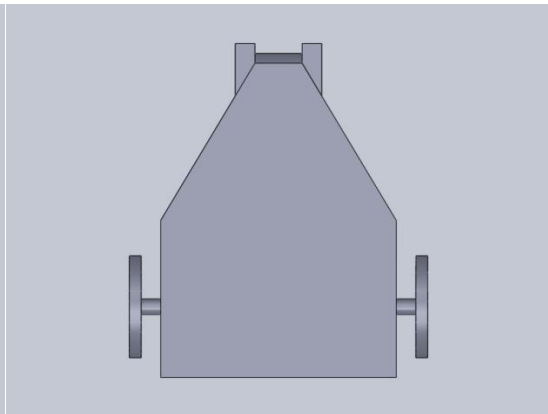


Figure 3: Dock-Bot's platform

Dock mechanism showed in figure 4 is basically a ramp mounted on iRobot.

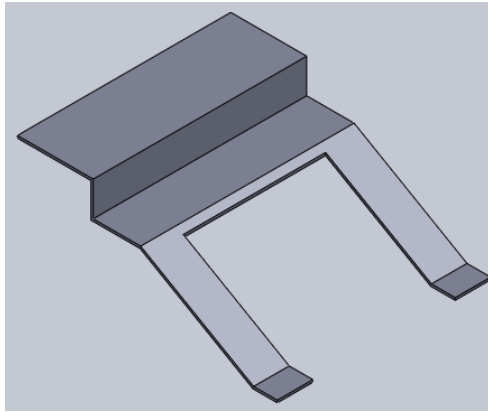


Figure 4: Ramp

Camera pan platform will be directly driven by a continuous servo.

Actuation

iRobot has its own independent and mature motion actuators and control methods. Actually there is even an open source MATLAB lib with kind interface for it.

Dock-bot's mobile platform is actuated by two continues servos from Parallax company.

Another HS-311 will be to pan the camera.

Buzzer and LCD screen will act as auxiliary reminders.

Sensors

Hardware and operation

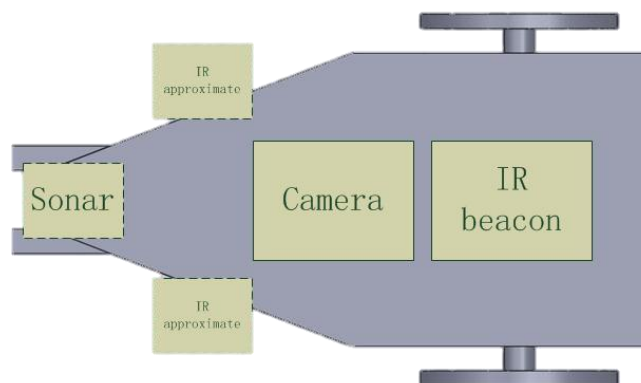


Figure 5: Sensors on dock-bot

IR beacon is a good choice for finding the base. The Pololu IR beacon is a compact board that is used in pairs to allow robots to locate each other. Each board has infrared emitters that shine in

all directions and four IR receivers for detecting the other beacon. The IR beacons have a range of about fifteen feet indoors.

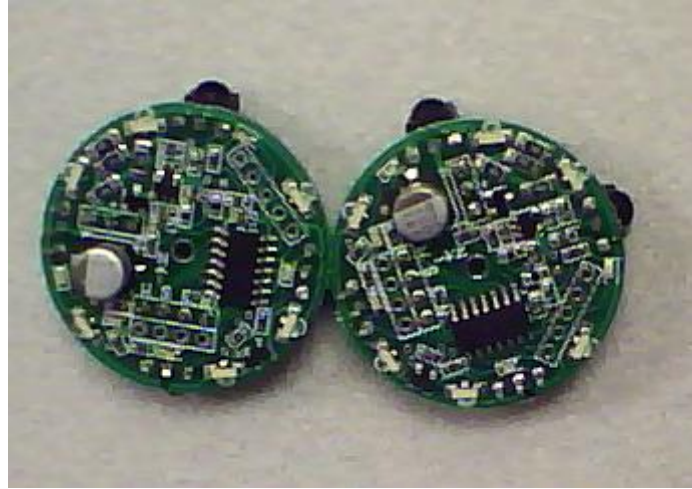


Figure 6: IR beacon pair

To avoid obstacles two IR approximate sensors SHARP GP2Df2 18 will be set under Dock-Bot's platform.

Sonar (model LV-MaxSonar-EZ4) will be used to detect the distance between the Dock-Bot and base robot, and to tell whether it is close enough for docking.

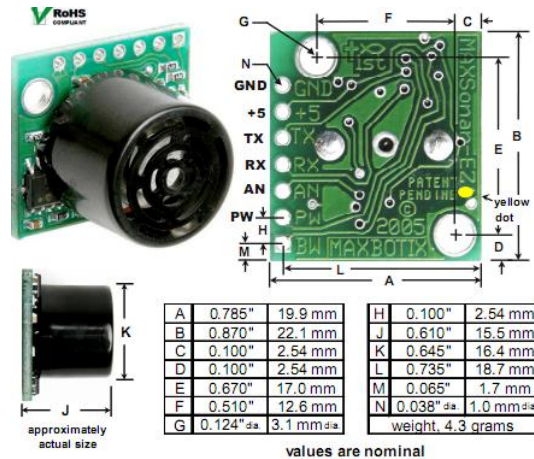


Figure 6: Sonar

Wireless Camera kit (model 203CWAS&RC410) cooperated with the characters on the base can help Dock- Bot to adjust its position and attitude. With Hauppauge usb live cable this kit can be connected to any PC.



Figure 6: Wireless camera kit

Precise closed-loop control of wheels in which encoder (Model WW01 from nubotics) will be used.

Figure 7: Hauppauge usb live



Figure 8: Encoder

Both two bump sensor will be set on the dock mechanism. One is used to tell when the dock begins and another tells when it is finished.

Software and algorithm

Details about software and algorithm of two special sensors IR beacon and wireless camera will be introduced in this part. Code will be attached in appendices.

➤ IR beacon

A schematic representation is showed in Figure 9. The algorithm I used is composition of vector. Four receivers stand for four directions which are vertical to each other. Specifically, north, east, south and west are expressed as $[0, 1]$, $[1, 0]$, $[0, -1]$ and $[-1, 0]$ respectively. I Also because of its sensitivity to the environment, sampling and weighting are applied.

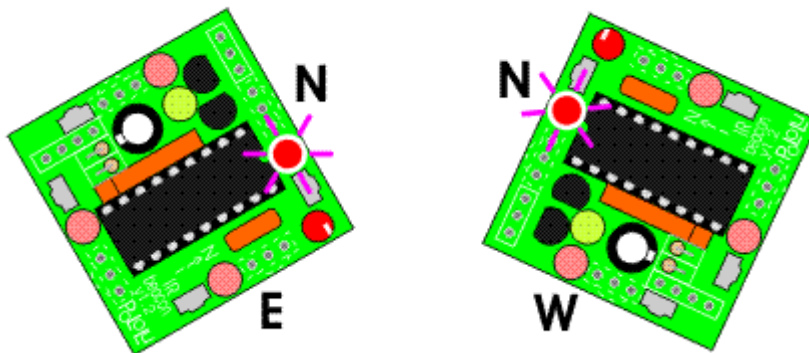


Figure 9: Schematic representation of IR beacon.

➤ Wireless camera and image process

By recognizing the characters which are set on the base robot, dock-bot can tell not only the relative position but also the relative attitude. Three hemisphere objects with different colors (orange, blue and green) are used as characters. By tell the area of the characters in the vision of camera, dock-bot get its ability to tell whether the object in front of it is the base. By comparing the areas of different colors, dock-bot can figure out which side it is on the base. OpenCV and a lib called cvblob help me to do image processing in laptop.



Figure 10: Characters in camera vision: colors from left to right, blue, orange and green; attitude from left to right, at the left side of base, in front of base and right of base.

Communication

Xbee kit includes two set of Xbee module and Xbee explorer.

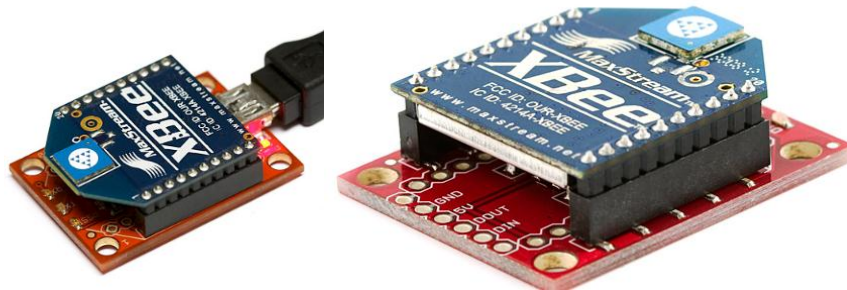


Figure 9: Xbee kit

BAM (short for Bluetooth Adapter Module) enables wireless control of the iRobot® Create™ robot from a Windows, Macintosh, or Linux PC. The BAM connects to the Create's cargo bay port - without any extra wires or cables.



Figure 10: BAM

Behaviors

All the functions can be summarized as four steps:

Step 1: Dock-bot finds the base by IR beacon, and moves closer to it. Obstacles are designed to be avoided in this movement.

Step 2: By using sonar and camera, control Dock-bot goes around base and locate the dock mechanism's position on base robot, then goes towards to it.

Step 3: By using sonar and camera (which is designed to be able to rotate) find the center of the dock mechanism on base. Two bump sensors will conduct the dock mechanism's behavior.

Step 4: Act as one complete robot.

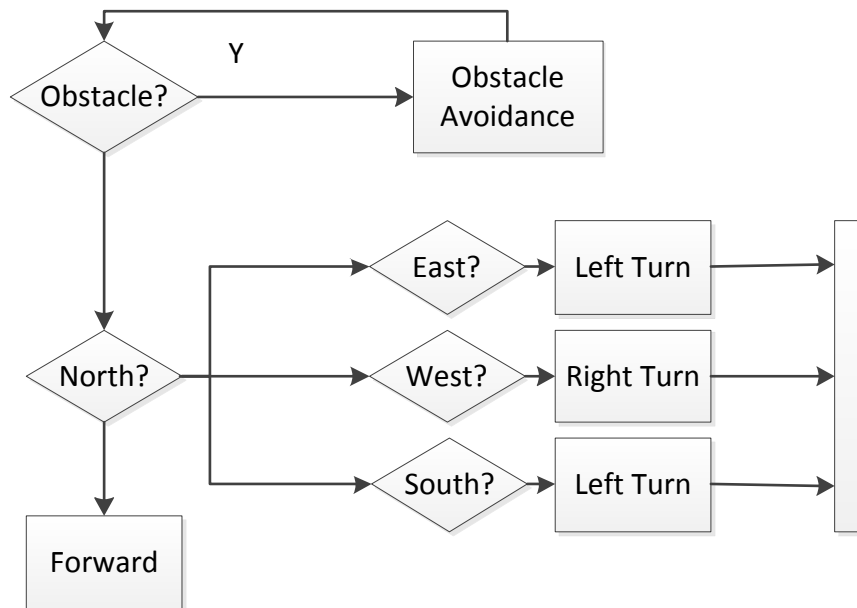


Figure 11: Move toward base flow chart

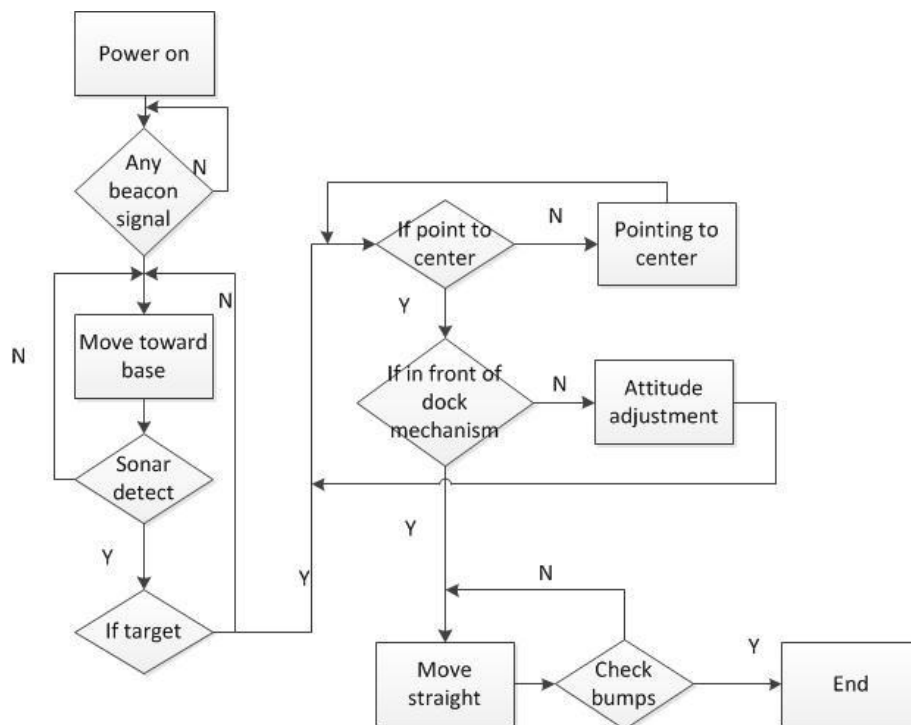


Figure 12: Complete system flow chart

Experimental layout and results

Trial and error for the attitude adjustment algorithm took me a long time. After numerous

modifications, I find out one effective way to adjust the position and attitude.

To find out a good way to recognize the relative position and attitude of the two robots, I tried three different ways. First, I want to use three points with different colors on a line. By calculating the slope of this line, I can tell which side dock-bot is on the base robot. Second, I want to use three painted ping-pang balls lies in one line. Both these two plans did not work very well. Then, the final plan placing three hemisphere objects as a triangle jumped into my mind. And it turned out to be the best one.

Conclusions

In the mid-term, I thought I cannot finish this project because all the serial communication and vision process are almost totally new for me, a typical mechanical engineering student. However, my enthusiasm for robot supports me so much that I persist and keep work hard. Also Dr. Arroyo and Dr. Schwartz, all the kind TAs help me so much. And I appreciate the sharing spirit of all the students who fronted the similar problems to mine. After all these, I finally build my robot, and after the successful demonstration on media day, I am so excited.

Lessons I learned from this whole semester include always making a clear plan and trying to catch up with the plan if there is one.

Documentation

LV-EZ4 datasheet:

<http://www.maxbotix.com/uploads/LV-MaxSonar-EZ4-Datasheet.pdf>

IR range:

<http://www.acroname.com/robotics/parts/R48-IR12.html>

Xmega128a1:

http://www.atmel.com/dyn/resources/prod_documents/doc8077.pdf

Opencv:

<http://www.opencv.org.cn/index.php/%E9%A6%96%E9%A1%B5>

<http://code.google.com/p/cvblob/>

Usart:

<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=45341>

<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=48188>

Appendices

Code for IR beacon

```
void getDirectionVector(void)
{
    direction_vector[0]=0;
    direction_vector[1]=0;
    n!=(PORTJ_IN&0x01);
    e!=(PORTJ_IN&0x02);
    s!=(PORTJ_IN&0x04);
    w!=(PORTJ_IN&0x08);
    if (n==1)
    {
        direction_vector[1]=direction_vector[1]+1;
    }
    if (e==1)
    {
        direction_vector[0]=direction_vector[0]+1;
    }
    if (s==1)
    {
        direction_vector[1]=direction_vector[1]-1;
    }
    if (w==1)
    {
        direction_vector[0]=direction_vector[0]-1;
    }
    //lcdInt(sum_direction);
    //delay_ms(50);
    //lcdData(0x01);
}

void setDirection(void)
{
    int j=1;
    sum_dir[0]=0;
    sum_dir[1]=0;
    while (j<=10)
    {
```

```
    getDirectionVector());

    sum_dir[0]=sum_dir[0]+direction_vector[0];
    sum_dir[1]=sum_dir[1]+direction_vector[1];

    j++;
    delay_ms(5);
}
//lcdInt((int)sum_dir);
//delay_ms(50);
//lcdData(0x01);
aver_direction_vector[0]=sum_dir[0]/10;
aver_direction_vector[1]=sum_dir[1]/10;
}

void getProduct()
{
    setDirection();
    vector_product=aver_direction_vector[1]*(-1);
    cos_angle=vector_product/sqrt(aver_direction_vector[0]*aver_direction_vector[0]+aver_dir
ection_vector[1]*aver_direction_vector[1]);
}
}
```

Code for image process

Please check my project website:

http://i-robot-e.comze.com/1_21_Code.html

Code for xbee and serial communication

Please check my project website:

http://i-robot-e.comze.com/1_21_Code.html