Robotics
Homework Assignment #1

Sang Woo Lee

Assignment 1:
When *Roomba* has completed its cleaning cycle or when the battery is running low, it automatically returns to its Home Base (figure 1) to recharge. The goal of this assignment is write a program for the *Create* robot to find and dock with its charging station without human intervention.
You may assume that the Home Base is located directly next to a wall, its immediate vicinity is unobstructed by obstacles, and all walls within the environment are connected, with no gaps. You should not assume that the Home Base will always be in range of the infrared sensor on *Create*, so you will need to decide on a strategy to guide the robot towards its goal without that information until a signal is detected.

My solution:
I solved this problem with two stages. First stage, Roomba Create finds wall and follow the wall. Second stage, Roomba docks to home base.

First stage can be accomplished by simple planning. When the wall is left side of Roomba, we can follow wall by keeping wall on the left side. This is true if wall shape is connected concave. This can be done by this simple strategy.

Strategy for following wall for connected concave shape:
1. Move straight until bumped (find wall)
2. Move backward slightly and turn the wall to be located on left side of Roomba
    Both bumps: 90 CCW turn
    Only left bump: 45 CCW turn
    Only right bump: 135 CCW turn
3. Drive CCW turning at radius of 200mm until bump
4. Repeat 2 when bumped

I think about using wall signal intensity to maintain some fixed range distance, but it seems that the signal depends on the wall's color, so I used this simple strategy and it works well.

Second stage is quite tricky, because we can only rely on IR beam to locate docking station. Because IR sensor is not quite sensitive and precise, so there are always errors. Open Interface for Roomba create only provide information for what IR beam regions, so we should guess Roomba's directions by detecting changes of IR beam regions. However, IR signal is very unstable, so it changes quite rapidly if Roomba is far from the Home base.
At first time, I tried to follow IR beam when I first met the IR signal, but there are frequent errors and changes of IR receivers when it has some distance from Home base. Therefore, I changed my plan to follow the wall first, and Roomba start docking when it found IR signal during following the wall. Because home base should be directly next to the wall, Roomba is very close to home base when it detects IR signal, so it is guided well by strong IR signal.
We can only guess where Roomba is currently headed by detecting last region and current region of Roomba. Therefore, I used simple guidance policy based on recent region changes. It works well because it can detect correct regions of IR signal with strong (close) signal from home base.

Strategy for docking based on IR beam (R: Red Buoy, G: Green Buoy, F: Force Field, N: No IR beam signal)
RGF->RGF: straight
RG->RGF: straight
R->RGF: CW turning at small radius

G->RGF: CCW turning at small radius
RF->RGF: CW turning at small radius
GF->RGF: CCW turning at small radius
N>RGF: straight
F->RGF: straight
RGF->RG: CW turning at small radius
RG->RG: straight
R->RG: CW turning at small radius
G->RG: CCW turning at small radius
RF->RG: CW turning at small radius
GF->RG: CCW turning at small radius
N->RG: straight
F->RG: straight
RGF->R: CW turning at small radius
RG->R: CCW turning at small radius
R->R: straight
G->R: CCW turning at small radius
RF->R: CW turning at small radius
GF->R: CW turning at small radius
N->R: straight
F->R: straight
RGF->G: CCW turning at small radius
RG->G: CW turning at small radius
R->G: CW turning at small radius
G->G: straight
RF->G: CCW turning at small radius
GF->G: CCW turning at small radius
N->G: straight
F->G: straight
RGF->RF: CCW turning at small radius
RG->RF: CCW turning at small radius
R->RF: CCW turning at small radius
G->RF: CCW turning at small radius
RF->RF: CCW turning at small radius
GF->RF: CCW turning at small radius
N->RF: CW turning at small radius
F->RF: CW turning at small radius
RGF->GF: CW turning at small radius
RG->GF: CW turning at small radius
R->GF: CW turning at small radius
G->GF: CW turning at small radius
RF->GF: CW turning at small radius
GF->GF: CW turning at small radius
N->GF: CCW turning at small radius
F->GF: CCW turning at small radius
(Any RGF)->N: turning 180 degree to go to the IR region

I elongated the distance for backing when bump is detected, because if it is too short then Roomba cannot escape the home base's two charging column. I also used virtual wall signal flag, but it seems to be more tricky for making proper backing, so I just used simple backing.
Another concern is proper speed for docking. If it is too slow, sometimes wheels are hanged on the air. If it is too fast, bump detected faster than charge available, so Roomba keep docking and backing by bumping.

I used Command module to control Roomba. The sources code is based on drive.c, which is one of the Roomba Command module example codes.

Result:
It works well for lots of trials. Because the Roomba find the wall first and follow it while the wall is on left side of Roomba, it takes longer time to find home base if the Roomba's first contacted the wall with right side of home base. However, it works well because when it entered IR beam region, Roomba is very close to home base, so IR region data is more relevant than far from home base case. Sometimes IR signals became lost unexpectedly, but Roomba backs to IR regions with bump backing because it is close to wall.

Considerations:
Bluetooth testing takes quite long time for me. Bluetooth control can suggest much more flexibility, but I couldn't use it because it seems that incoming data from Roomba is not correct. I wonder it is caused by too Bluetooth connection for windows. I don't have much time to check this. I hope next time I can use Bluetooth connection, because it is much easier to see the status.