Lab4: iRobot Navigation in C

Sam Mansfield and Toan Vuong
TA: Hoekun Kim
EECS149

October 2nd, 2013

1 Introduction

2 Analysis

We ended up adding two new states, BACK and TURNRIGHT. When the iRobot is unpaused and senses an obstacle or cliff in the DRIVE state, it goes into obstacle avoidance mode and goes into the BACK state. The BACK state travels 50 mm backwards and goes into the TURNRIGHT state and the TURNRIGHT state, after making a right turn goes back into the DRIVE state. Figure 1 displays our final FSM.

At the beginning of the lab we tried to create a more elaborate FSM and realized that we were actually making things more complicated to debug and therefore implement. By simplifying our design we created a better FSM. If I had to change one thing I would take out the square pattern, it makes the program more complicated to think about since you only have to correct within your square, which is a little odd considering that you will eventually be going in the corrected orientation after an obstacle eventually.

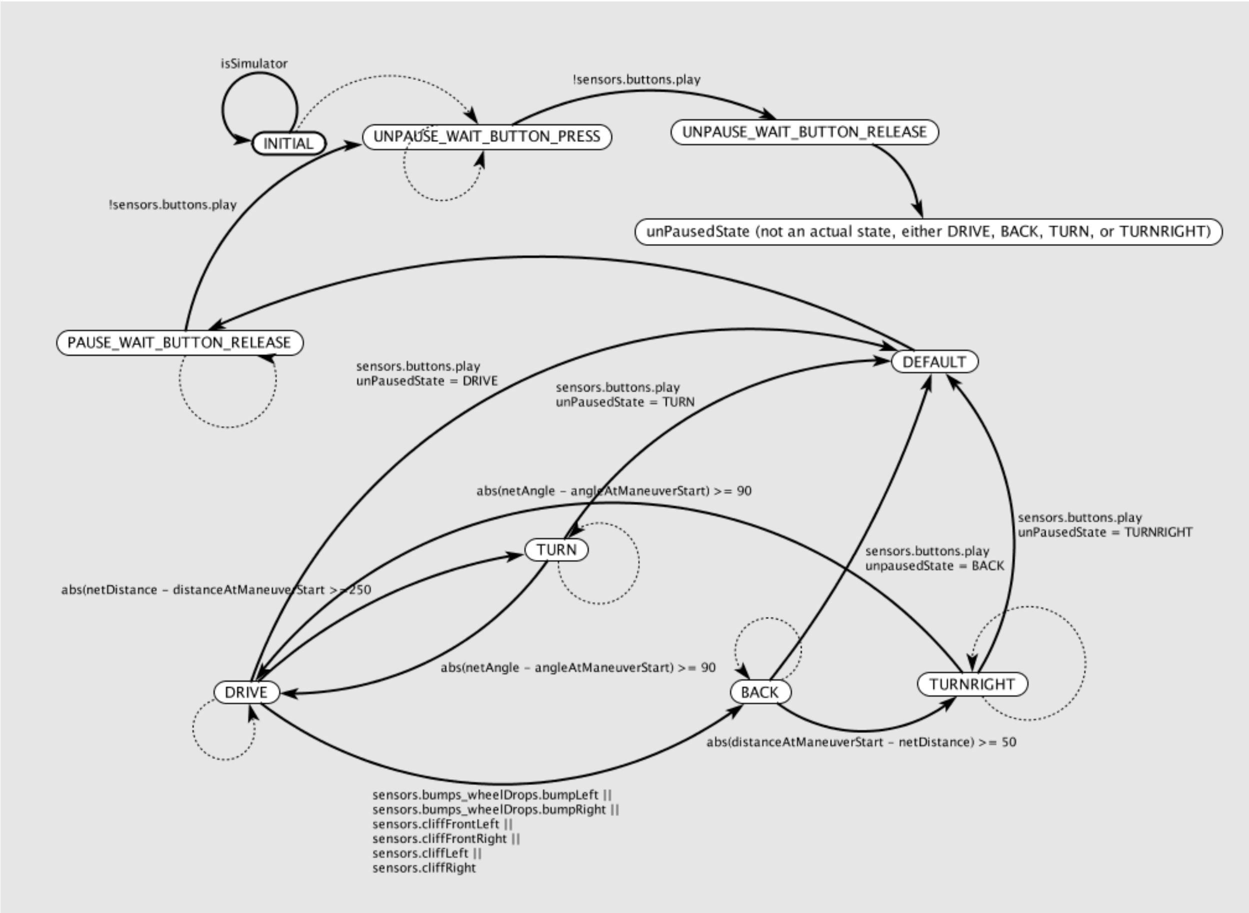
isSimulator
INITIAL
UNPAUSE_WAIT_BUTTON_PRESS
!sensors.buttons.play
UNPAUSE_WAIT_BUTTON_RELEASE
!sensors.buttons.play
unPausedState (not an actual state, either DRIVE, BACK, TURN, or TURNRIGHT)
PAUSE_WAIT_BUTTON_RELEASE
DEFAULT
sensors.buttons.play
unPausedState = DRIVE
sensors.buttons.play
unPausedState = TURN
abs(netAngle – angleAtManeuverStart) >= 90
TURN
sensors.buttons.play
unPausedState = TURNRIGHT
sensors.buttons.play
unpausedState = BACK
abs(netDistance – distanceAtManeuverStart >= 250
abs(netAngle – angleAtManeuverStart) >= 90
DRIVE
BACK
TURNRIGHT
abs(distanceAtManeuverStart – netDistance) >= 50
sensors.bumps_wheelDrops.bumpLeft ||
sensors.bumps_wheelDrops.bumpRight ||
sensors.cliffFrontLeft ||
sensors.cliffFrontRight ||
sensors.cliffLeft ||
sensors.cliffRight

Figure 1: Obstacle/Cliff avoidance FSM implemented in Lab 4

3 Conclusion

Designing an obstacle/cliff avoidance algorithm was quite fun and rewarding when it actually worked. This was a very applicable problem for state machines and we can apply this technique to future problems that are similar in nature. This lab also helped us think about solving very simple human problems (avoiding an obstacle) step by step so that it can be applied to a robot.