CSCE 274 Section 1 Fall 2016 – Project 04 – Boyd Compton, Jose Tadeo, Timothy Senn

Project Description:

Our main program for project four uses a behavior based control scheme. Each behavior is responsible for submitting a potential action to the main controller. The main controller would then select one of the submitted actions based on the behavior's priority. This controller also features action preemption so that the higher priority actions have precedence over lower priority behaviors at all times. The controller is comprised of the following behavior modules ordered by priority:

- Docking:

  The docking module currently has the highest priority and can preempt any other behavior. This behavior will remain inactive until a docking station is close enough to derive an action. Currently, this translates to the robot needing to face the front of the docking station before the behavior activates.

  Once the behavior activates, the dock's infrared characters are sampled a number of times before an action is determined. The sample amount was decided based on two criteria:

  1. The effective time it takes to successfully dock
  2. The reliability of the sample

  During sampling, each incoming infrared character was decomposed into its respective beam's flag via bit wise operations and then fused with the current sample.

  After enough samples were collected, the next action to approach the dock was decided according to the following decision table located below. The selected action would be the first action to meet at least one condition. The order that the actions were checked is presented in the action column from top to bottom.

Docking Decision Table	
Action	**Condition** *(The * specifies conditions that do not require sampled data)*
Terminate Program	▪ The robot is in any charging state*.
Drive Backwards	▪ The left or right bump is pressed*. ▪ The robot is too close to the dock with the wrong rotation*.
Drive Forward	▪ The green and red buoys are detected on fusion of the left and right infrared character. ▪ The robot has the correct rotation and is close to the dock*. ▪ No action can be determined and the previous action was *Drive Forward*.

Rotate Clockwise	▪ Only a green buoy is detected on the right.
Rotate Counter-Clockwise	▪ Only a red buoy is detected on the left.
Repeat Previous Action	▪ The dock can still be detected, the center of the dock is already lost, and no action can be determined.
Randomly Rotate	▪ The dock can still be detected, the center of the dock was just lost, no action can be determined, and the previous action was *Await Further Instruction*.
Reverse Previous Action	▪ The dock can still be detected, the center of the dock was just lost, and no action can be determined.
Await Further Instruction	▪ The dock's infrared sensor was lost. ▪ No other action's condition is met.

- *Obstacle Avoidance*:

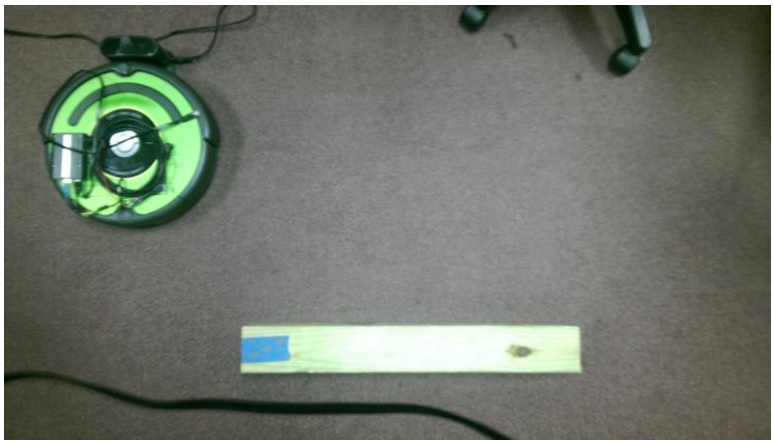
  This behavior is responsible for collision response and the response to unsafe situations. Collisions are detected by interrupting the physical bump and cliff sensors' value while an unsafe situation are determined by checking the wheel drop sensors. The amount of rotation on a collision was derived from the kinematics equation for in-place rotation. The formula was rearranged so that given an angle the amount of time could be determined. The amount of rotation upon collision is a random time in between a maximum and minimum bound. The bounds were tuned with the stability of the wall following behavior in mind. Currently, this module can preempt only the wall following behavior.

- *Wall Following:*

  The wall following behavior is capable of following walls with left and right turns to an acceptable degree. The degree of stability is dependent on the angles of the turns and the type of wall material. This module was tested with wooden wall of standard household wall angles (~45° - ~135°). Additionally, this can only follow walls in a clockwise manner due to the left light bump sensor placement.

  The standard wall following behavior and right turns are handled with a PID controller while the left turns are handled by a PD controller. The PID controller uses a right light bump sensor as input while the PD controller uses the center light bump sensors as input. If this behavior cannot find a wall after a couple of seconds, it will then begin to seek out a new wall to follow by driving forward.

Project Evaluation:



The image above details a situation where all behavior will work reasonably well and the charging dock will be detected by the docking behavior. The shape of the wall does not matter; only the relative position of the dock to the wall.

Overall, the obstacle avoidance and wall following behaviors are the most reliable and will work reasonably well if the environment is within the parameters expressed above. However, when an environment does not meet the expressed parameters, the performance of each of the behaviors will degrade. Usually the wall following behavior could oscillate more around the set point or the set point could move closer or further away from the wall. Although, when the wall angles are acute, the performance of both behaviors can vary significantly due to the potential of continually oscillating between rotating clockwise and rotating counter-clockwise. An amount of randomness is built into the behaviors to help reduce the chances of this happening, but the effectiveness of the added random nature degrades as the angle approaches 0°. In total, these two behaviors do reasonable work under specific environmental constraints.

Overall, the docking behavior will often successfully dock the robot given the following conditions (visual depiction above):

1. The robot will roughly face the dock at some point during its motion.
2. The dock is about tangent to the sides of the robot while docking.
3. No obstacles are between the dock and robot during the docking process.
4. The robot has enough free space in front of the dock to seek the dock.

In the ideal environment, the docking behavior will always approach the charging station. When the forward approach is lined up properly, the robot will successfully dock. However, if the robot missed the contact points, it will back up to try again. If the robot accidently touches both contact points while moving backwards, the robot will be switched to passive mode, and then continue decelerating. This sometimes leads to the robot undocking itself before the charging state can be detected and accidently, engaging built-in behaviors since the robot is now in passive mode. This

results in the program losing actuator control and the iRobot Create 2 begins to vacuum. With the docking behavior relying on the automatic change in state from safe to passive to enable the charging detection, there was no decent way to avoid this series of events. In total, the docking behavior will successfully dock given the robot remains in a charging state long enough for detection, and the conditions presented above are met.

Allocation of Effort:

Boyd Compton:

- Augmented the interfaces to include the infrared character sensors
- Wrote the underlying code for the behavior based controller
- Planned out a useful manner to check for a specific buoy
- Helped retune the PID controller for the wall following behavior
- Helped plan the basic docking behavior
- Wrote the project write up

Jose Tadeo:

- Helped retune the PID controller for the wall following behavior
- Adjusted the new gains, constants, and set points for the new time scale
- Aided in establishing the detection area of the infrared character sensors.
- Helped analyze and improve the docking behavior

Timothy Senn:

- Helped retune the PID controller for the wall following behavior
- Helped analyze and improve the docking behavior
- Aided in maintaining a clear and understandable code structure
- Wrote the README.txt