Which behaviors did you implement?

- Wall Following: The neato locates the wall and then moves alongside it at a specific distance.
- Person Following: The neato locates the nearest person and follows their movement.
- Obstacle Avoidance: The neato moves forward, and avoids obstacles in its path.
- Wall Following and Person Following: The neato locates the nearest wall or person and either moves alongside the wall at a specific distance or follows the movement of the nearest person.

For each behavior, what strategy did you use to implement the behavior?

- Wall Following: We had two stages to our code. Initially, the neato would find the closest object and treat it as a wall. It would then use proportional control to alter the speed of the turn when orienting the laser on the neato to see the wall at 90 degrees (parallel to the wall). The proportional control would turn faster when significantly off and slower when close to 90 degrees to not overshoot. Once it was oriented to face a wall, we made the neato keep going forward. It would still try and orient itself so that 90 degrees was always pointing towards the largest object. This allowed us to be able to turn corners in some cases.
- Person Following: We had the neato find the closest object within a 150 degree window in front of the neato. The neato would then orient itself to follow that person, keeping about 0.5m away using proportional control. We used running averages to make our closest object measurement more accurate.
- Obstacle Avoidance: The strategy used was for the neato to move forward until it senses an object less than 1.5m away. The neato then turns right 90 degrees and moves forward until the object can not be seen within 4m in its 90 to 100 degree window. The neato then turns to its original orientation and continues moving forward.
- Finite State Machine (Wall Following and Person Following): The strategies for the two states are almost the same as stated above. In wall following mode, the neato did not try and turn to orient itself while going forward. It would just keep going straight after it had figured out which direction to move in. This means that the neato would not be able to turn corners to follow a new wall while in wall following mode. The details of switching modes are addressed in more detail in the next question.

For the finite state controller, what were the states? What did the robot do in each state? How did you combine and how did you detect when to transition between behaviors?

The two states were wall following and person following. The robot observes person following whenever there is a person close enough (within 1.5m). In this mode, the robot finds the closest object in front of it (at a 60 degree range) and attempts to follow it by 0.5m. If the robot doesn't observe anyone within 2.5m, it switches out of person following into wall following. In wall following mode, the robot first orients itself such that the laser sees the wall (or closest large object) at 90 degrees. After reaching this point, the robot moves forward parallel to the wall until it senses another object close in front of it, and switches back to person following mode.

How did you structure your code?

We created a robot class that stored the important information about the neato, including speeds, necessary flags, and laser scan range measurements. This information was written to the instance of the class by the callback functions we had from the topics we subscribed to. The main behavior in each script was run by a while loop, which called functions we wrote to control the robot based on the parameters in the robot object.

What if any challenges did you face along the way?

- Turning to follow a different wall for wall following - We were able to implement this by using slow speed and creating walls that gradually changed instead of sharp corners.
- Detecting feet - The neato could not see feet, so instead the person had to drag a poster board along as feet.
- Getting the angles right for obstacle avoidance - We kept running into the object we were trying to avoid because we were not forcing our neato to drive completely past the object. Instead of looking at angles 85 to 95, we started looking at angles 90 to 100.
- Differentiating between walls and people for the finite state machine - We said that if the neato saw anything to it’s left while it was wall following, the object was a wall. Anything in front of it was treated as a person.

What would you do to improve your project if you had more time?

- Fixed distance from wall for wall following - We could have implemented this using the laser scanner in the initial step where the neato orients itself to be parallel to the wall
- More smooth detection of people
    - Potentially differentiate moving and nonmoving objects (people versus walls) in some way. Knowing the approximate width of an object might also help with differentiating between different kinds of objects
    - Maybe use a better weighting algorithm to figure out where to follow instead of just using running averages
- Avoid obstacles to reach a goal location instead of just going straight
- Add obstacle avoidance or other features to the finite state machine
- Taking time averages of data might have also helped us acquire smoother measurement, in addition to using running averages over the laser ranges

Did you learn any interesting lessons for future robotic programming projects?

Keeping track of which angles you are dealing with and in which coordinate system is important. Also try and structure your code such that all your variables that are used to direct behavior get updated as quickly as possible. We created one variable that was cleared at the beginning of a function and not written to until the end of the function. This caused problems, because while that function was running other processes would try and access the variable leading to very strange errors. We now know to keep track of the various processes that are running and how they interact.

We also found that there is a lot of “noise” in the laser scanning the robot does. We’re interested to learn more about good averaging and filtering techniques to best understand laser scan data. Using running averages definitely helped, but more advanced techniques would improve our person following and obstacle avoidance capabilities.