



### Idea

Basically I calculated the distance between the robot and the target. If the robot is getting close to the target I decrease the speed.linear.x from a fixed value to another fixed value.

### How to stop

I have a hard time making it arrive at exact coordinates, so I create a radius and make the target coordinates as the center. Once the robot reaches into the circle, I set both speed.angular.z and speed.linear.x to 0 to make it stop.

### Things to Improved

At the very first beginning, I thought too far ahead. I thought that I was asked to create the code to make the robot dynamically calibrate the speed.linear.x and speed.angular.z. So based on the atan2, I think I can adjust the speed.angular.z since it is proportional to the return value from atan2. Also, there might be a way to calibrate the speed based on the distance.

I spent too much time on thinking the details and ending up not enough time to achieve every task (such as 6 and 7). I still wish to update this if I can! I have learned so much from this lab! I enjoy it!