

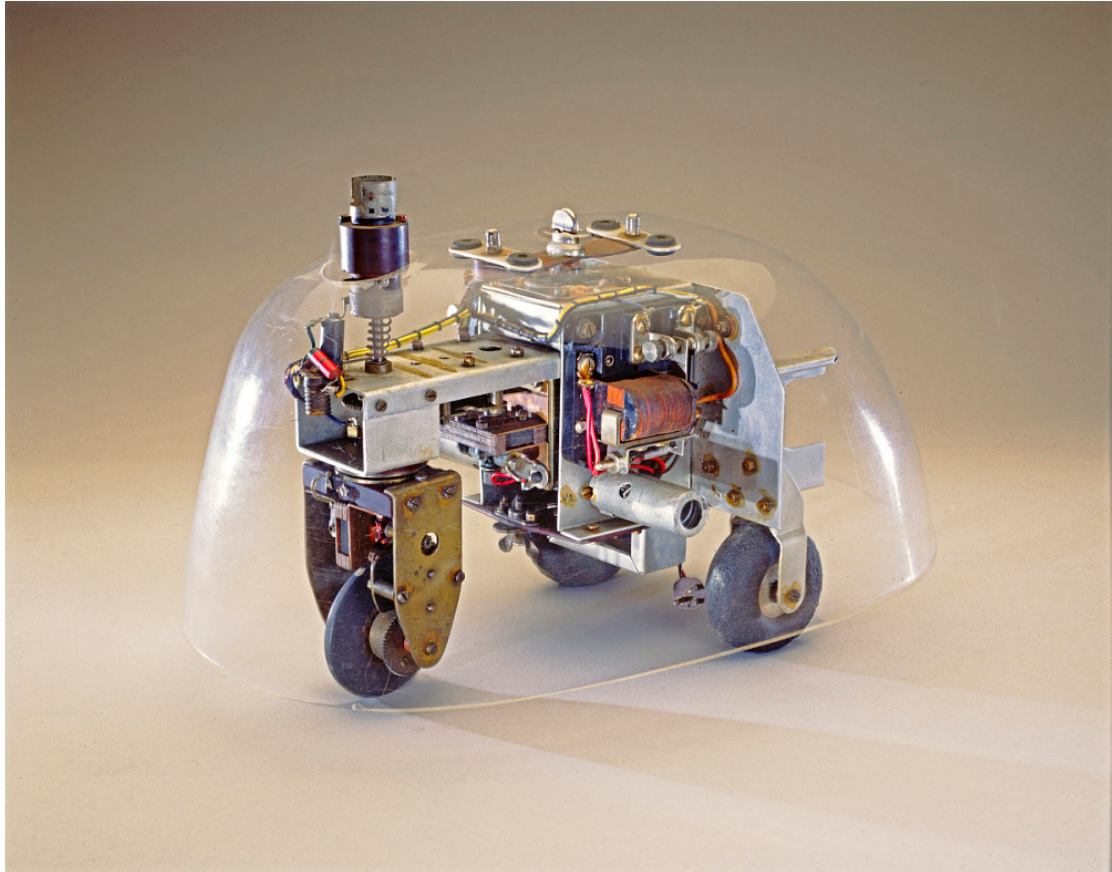
Turn and Push Robot

In this tutorial, you will learn a simple way to allow your robot to navigate its environment. Variations of what happens in this paper have been adapted for many robots, including space vehicles and the Roomba vacuum cleaning robot.

Grey Walter's Tortoises

"Tortoise" Mobile Robot (<https://www.youtube.com/watch?v=ILULRImXkKo>)

Grey Walter was a neuroscientist who developed a two robots called **tortoises** in order to demonstrate that even a small number of brain cells could give rise to complex behaviors. To demonstrate this, he created a robot with only two motors and a photo sensor on the top. One motor allowed the robot to go forward and backward, but defaulted to going forward. One motor controlled a steering column similar to the steering wheel in a car. This was attached to the photo sensor. The robot was tuned to make a spiraling motion until it saw light, and then lock onto that light signal using only a few transistors.



(Image from Alen Winfield's [blog](#).)

By engineering the robot to turn its steering column towards a light source, the tortoise was said to "like" light. With getting close to the light as its goal, the robot would turn its steering column.

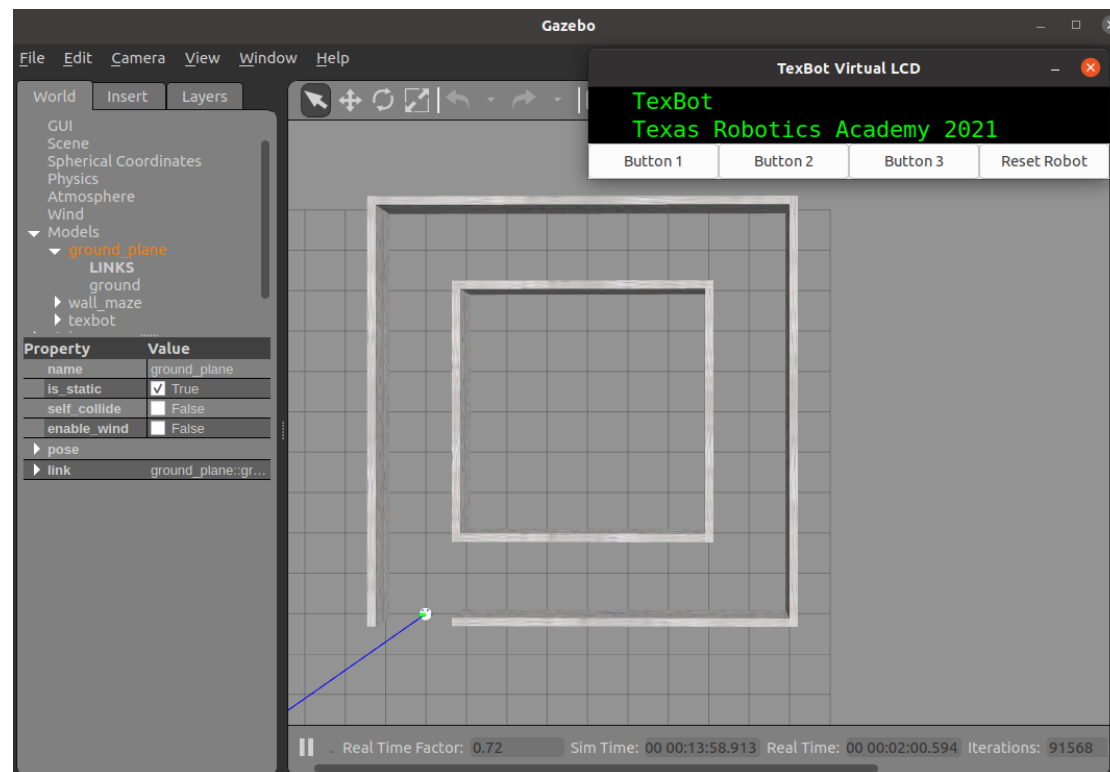
However, if an obstacle blocked the robot's path, with only the light sensor to guide it, the robot would become stuck, and never reach the light.

On the outside of the tortoise was a plastic shell. This was attached to a bump sensor. When bumped, the robot would do a very simple behavior. It would turn its steering column slightly to the left, and then back up. It did this on a timer. This behavior is called "turn and push."

By turning and pushing, the robot would eventually make its way around the obstacle, and to the light.

The Roomba does something similar as it vacuums. When its bump sensor is hit, it turns and begins its wandering behavior again.

Mid Term Project: Exercise 5



Following light is not part of this challenge, but we will be having the TexBot emulate the tortoise's **turn and push** behavior using our obstacle sensor.

- Start with the "empty" program.
- Make the robot move forward.
- When the obstacle sensor is triggered, rather than stopping the robot, make it go in reverse and to one side (you pick) for a moment. Use the `usleep` function to regulate how long it does this.

Test your program by running it in a new launch file we provided, `maze.launch`.

If your program is successful, when encountering a wall, it should eventually turn so as to avoid hitting the wall, rather than stopping! If the robot can get to the other side, you have passed the Turn & Push Challenge, and receive all of the bragging rights associated therewith.

Reference:

- <https://github.com/Texas-Robotics-Academy/markdown.git>