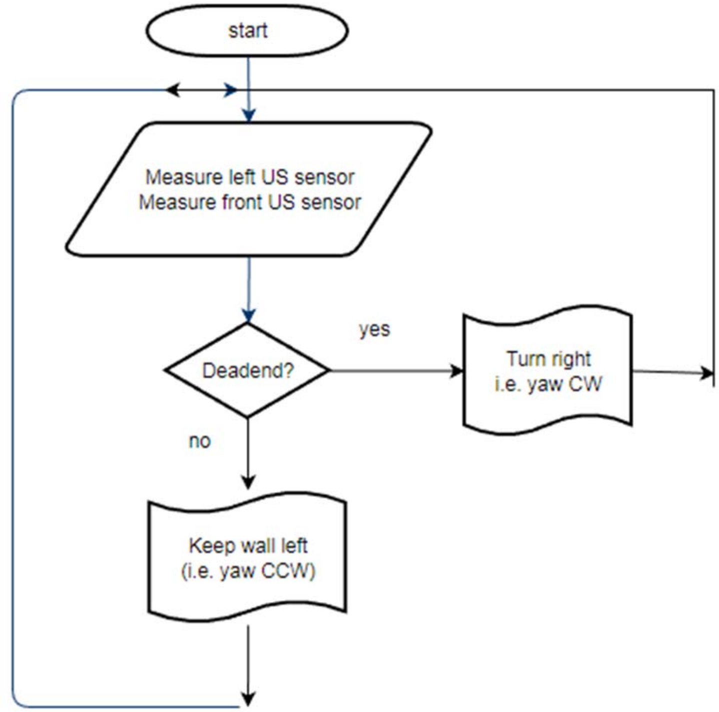
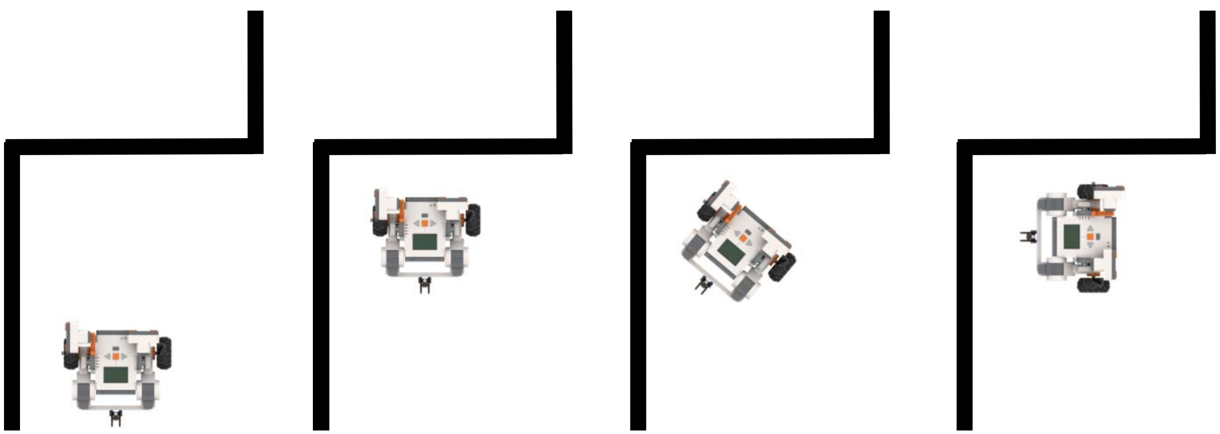
Lego NXT Domabot Wall-Following

Bang-Bang and Proportional Control

Motivation: Finding paths to complete a maze. Google Maps begins this way

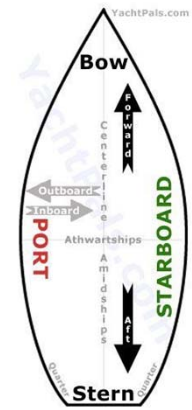


Solution: (1) Always hug the wall left of the robot; (2) Turn right if there's an obstacle

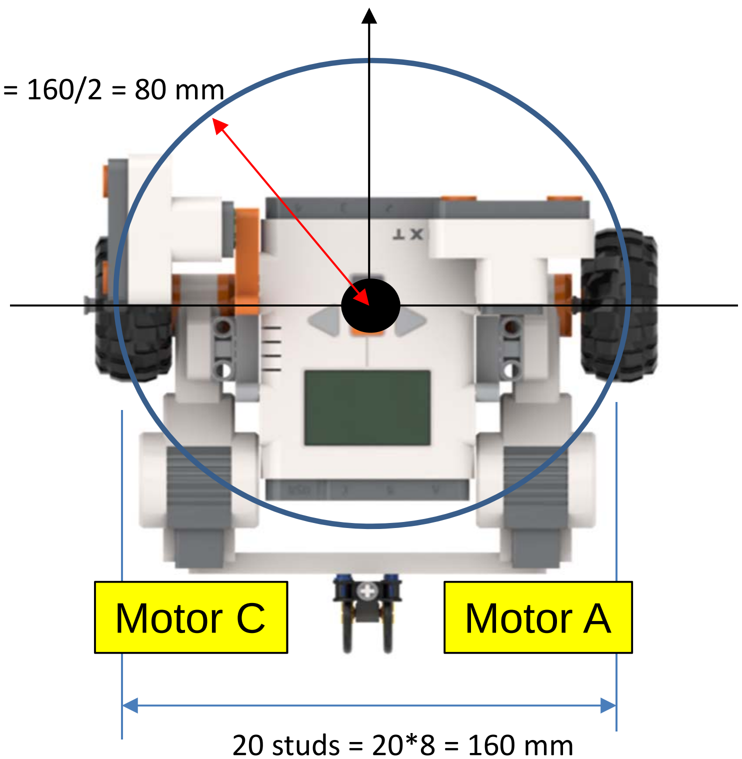


Implementation: Part 1 (wall-following)

First, we need an algorithm to keep the robot's portside a fixed distance from the wall. Like line-following, one uses closed-loop feedback e.g. bang-bang, Proportional, or PID



Q1: What portside-to-wall distance to use?



Tight CCW yaw:
- Rotate around body center (black circle)
- Motor A = speed and Motor C = `-speed`

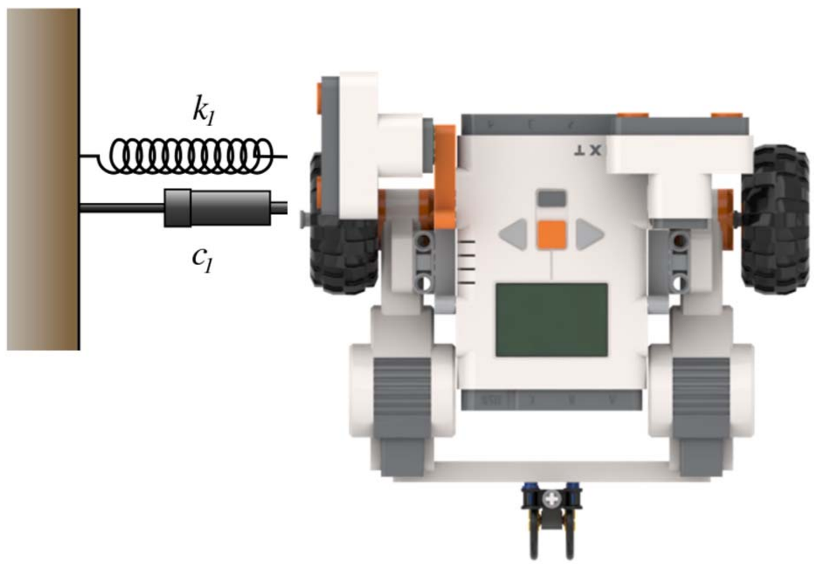
Wide CCW yaw:
- Rotate about wheel
- Motor A = speed and Motor C = `0`

Thus, a **10 cm** portside-to-wall distance is possible with tight yaws

Q2. Which closed-loop control to use?

Line-following experience contrasts bang-bang and proportional control. Both oscillate but the latter does so less. Why?

Conceptually, PID acts like a mass-spring-damper **on the error**



$$\text{Control Effort} = K_p e + K_I \int e dt + K_D \dot{e}$$

$$K_p E(s) + K_I \frac{E(s)}{s} + K_D \mathrm{sE(s)} \ \text{ or } \left\{K_P + \frac{K_I}{s} + K_D s\right\} E(s)$$

$$\text{or } \{s^2 K_D + s K_P + K_I\}$$

In other words, the form is a 2nd order system

Selection of PID gains can increase performance (i.e. reduce rise time) and/or stability (i.e. steady-state error, undershoot, overshoot)