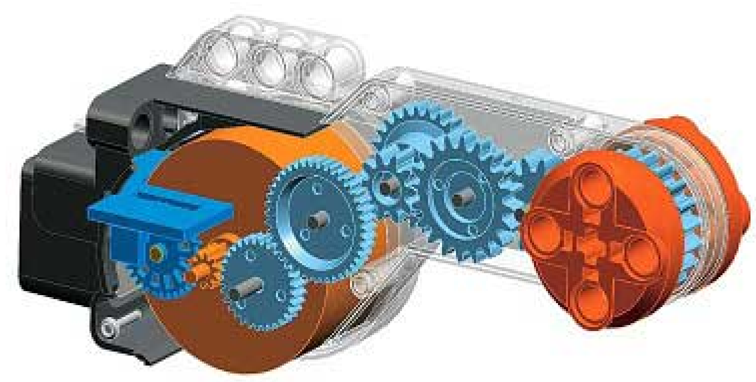
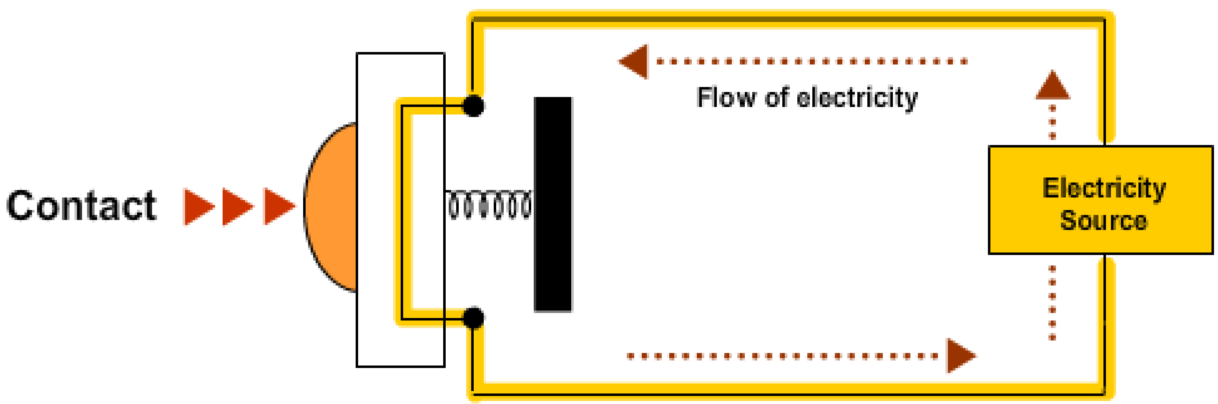
Robot Perception – The NXT sensors

Sensors and actuators

- Built in position sensor
- 360 degrees – Resolution ±1 degree



Touch sensor

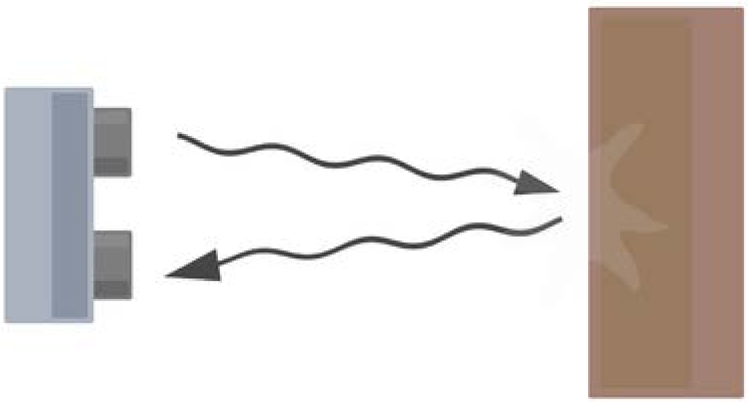


Light sensor

- Two modes of operation: active and non active LED.
- Color detection in active mode
- Light intensity detection in non active mode.

Ultrasonic sensor

- Ultrasonic transmitter and receiver
- Calculates distance depending on time of travel

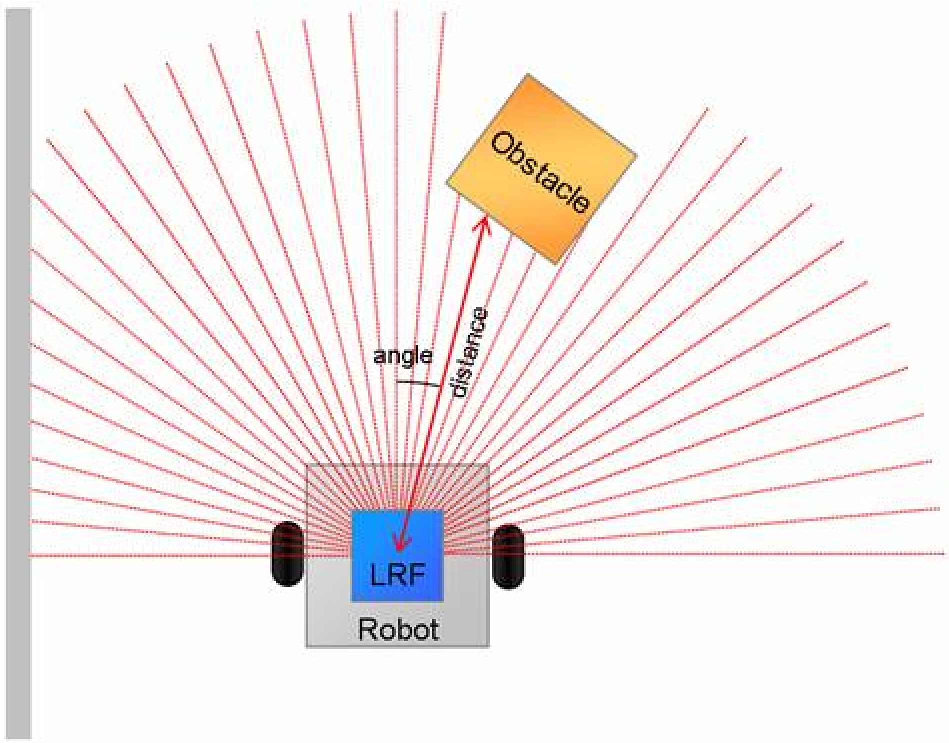


Sound sensor

- Simple microphone

Sound Sensor Readings	4-5%	5-10%	10-30%	30-100%
Comparable Sounds	Silent Room	Person Talking Far Away	Person Talking Near By	Shouting or Playing Loud Music

Mobile robots and useful sensors



How to read from Distance Sensor?

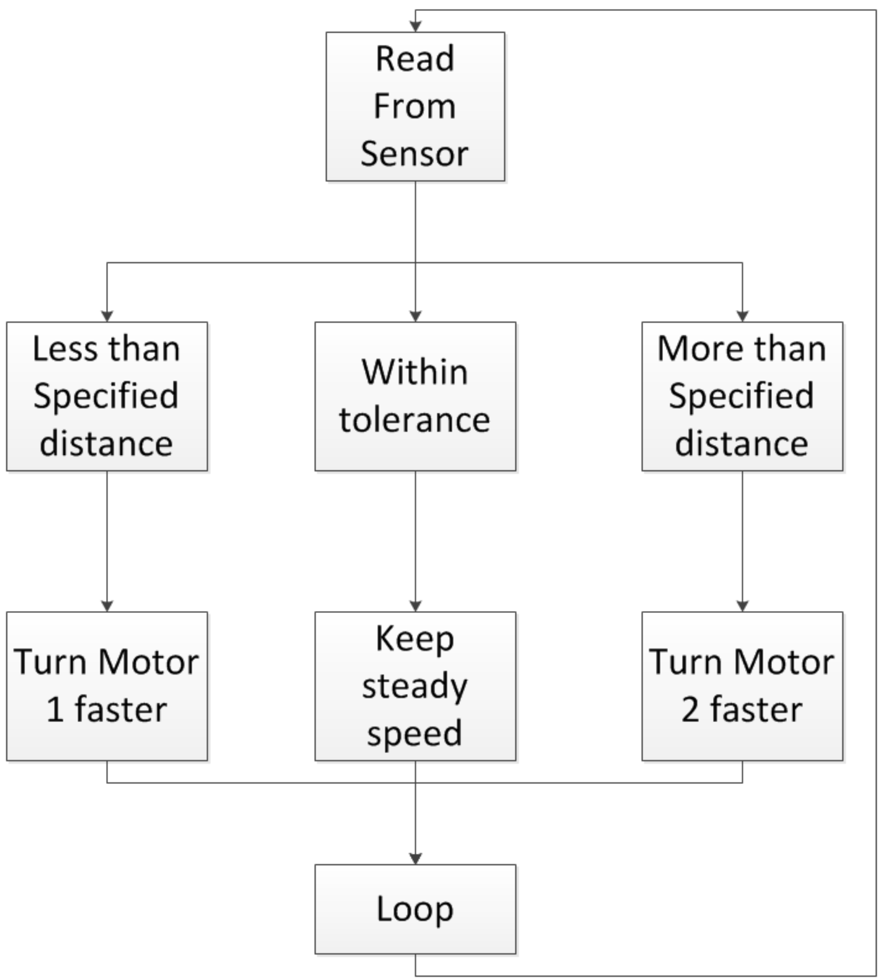
```
%%Variables
SENSOR_PORT = 3; %Sensor Port

%% Initialize sensor(Once only)
  OpenUltrasonic(SENSOR_PORT);

%% Getting readings(do this in your loop)
  distance = GetUltrasonic(SENSOR_PORT);

%% Close sensor(Before ending script)
```

- `CloseSensor(SENSOR_PORT);`

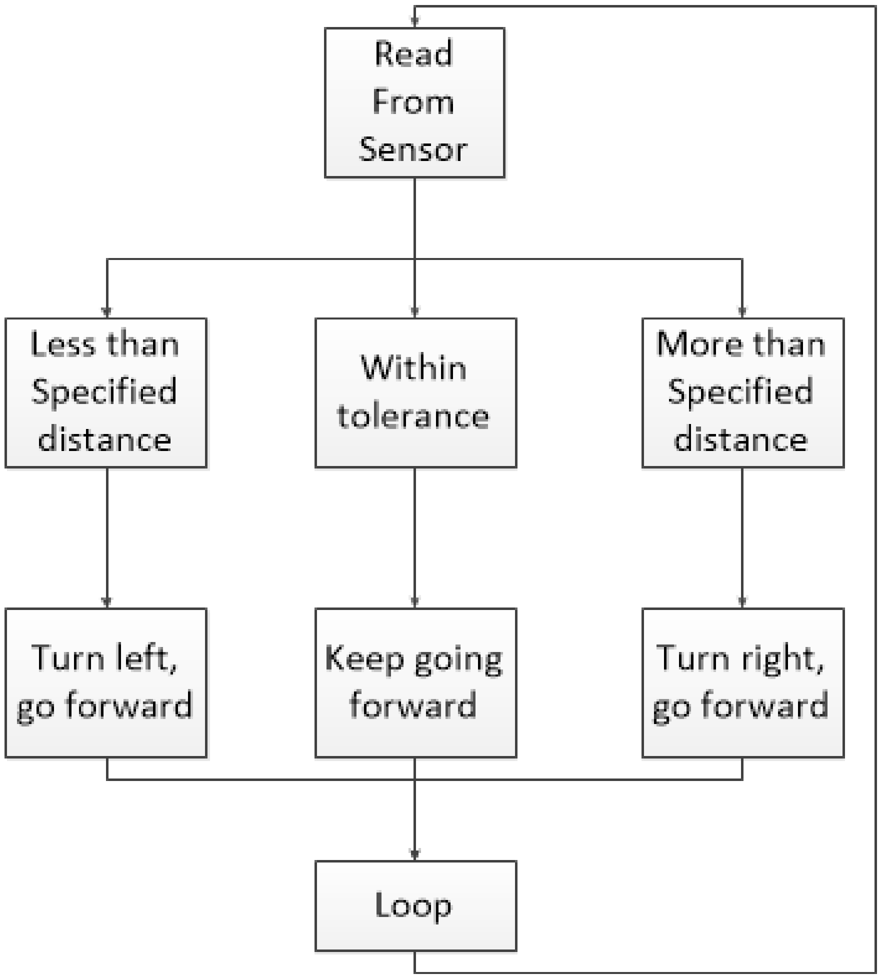
Challenge: Wall following robot

- Install your ultrasonic sensor pointing towards one side of your robot
- Point the sensor to a wall
- Make the robot cruise at a distance of 40±4 cm from the wall.
- Plot the distance from the wall real time versus time.

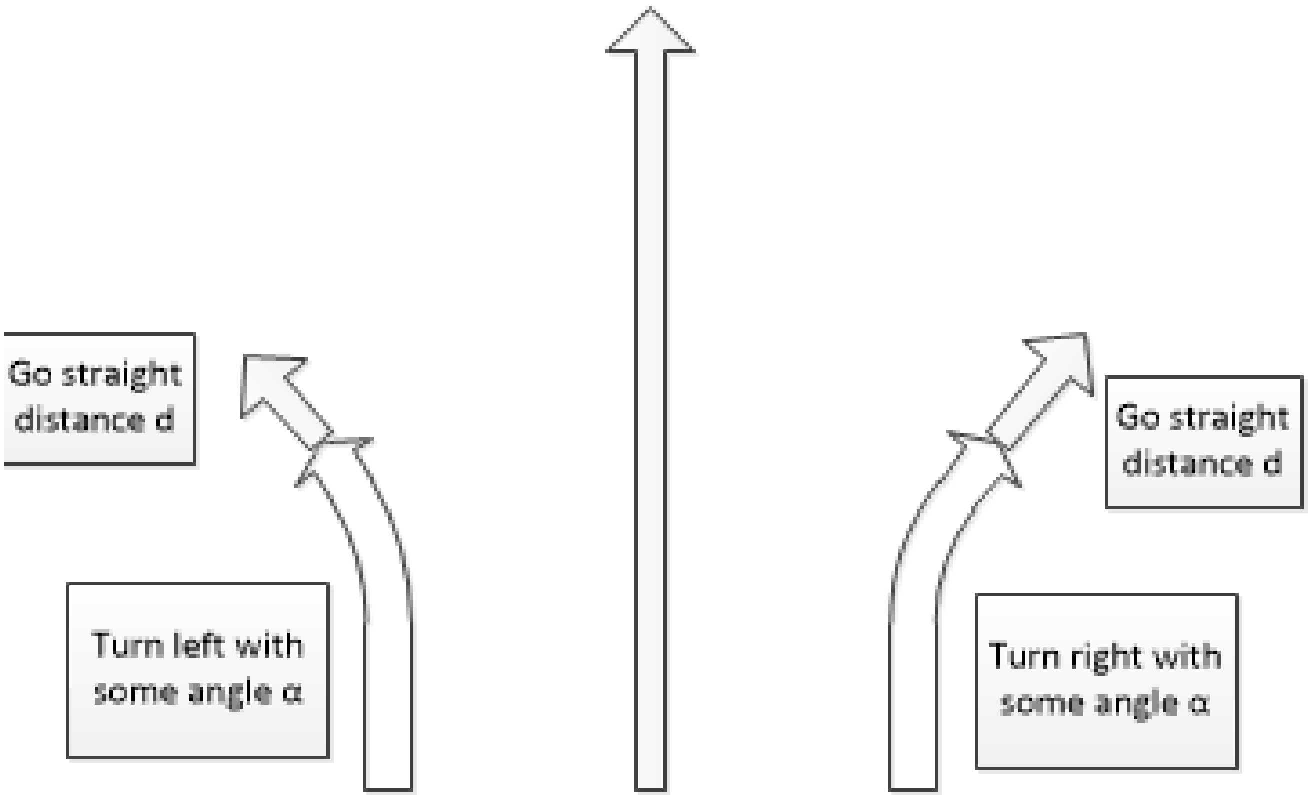
Wall following robot: Logic 1



Wall following robot: Logic 2



Wall following robot: Logic 2



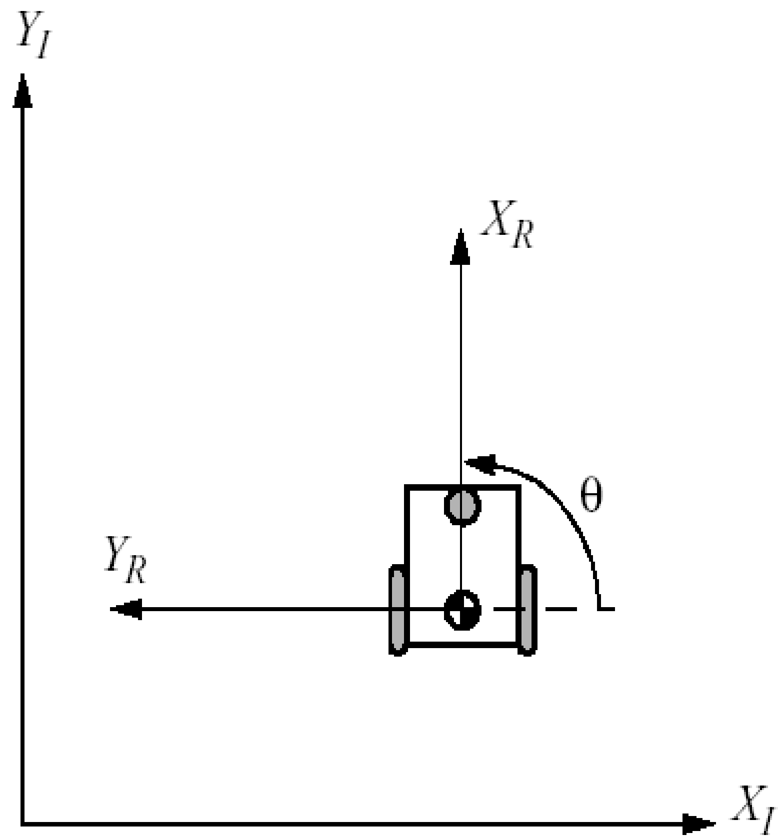
Motion model: Translate

Rotation around the center of the robot requires having $\dot{x} \neq 0, \dot{y} = 0, \dot{\theta} = 0$

Find the angle θ which defines the case where the frame considered is o the robot($X_R = X_I$ and $Y_R = Y_I$)

Find the relation between $\dot{x}\ and\ (\dot{\varphi}_1, \dot{\varphi}_2)$ and integrate to get the relation between total distance $x\ and\ (\varphi_1, \varphi_2)$

$$\begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r\dot{\varphi}_1 \\ r\dot{\varphi}_2 \\ 0 \end{bmatrix}$$



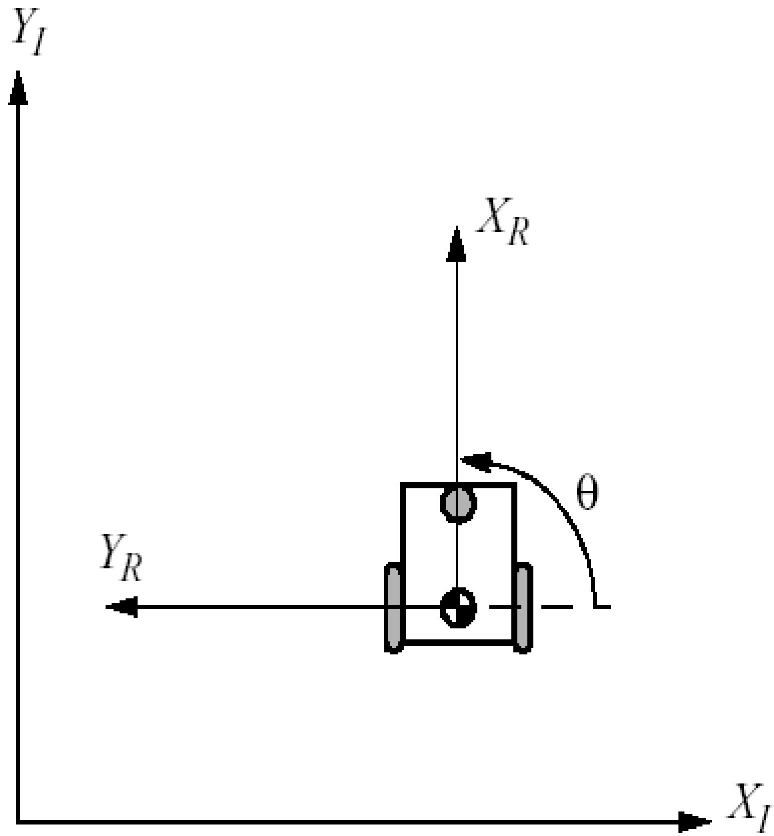
Motion model: Rotate

Rotation around the center of the robot requires having $\dot{x} = 0, \dot{y} = 0, \dot{\theta} \neq 0$

Find the angle θ which defines the case where the frame considered is o the robot($X_R = X_I$ and $Y_R = Y_I$)

Find the relation between $\dot{\theta}\ and\ (\dot{\varphi}_1, \dot{\varphi}_2)$ and integrate to get the relation between total rotation $\theta\ and\ (\varphi_1, \varphi_2)$

$$\begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r\dot{\varphi}_1 \\ r\dot{\varphi}_2 \\ 0 \end{bmatrix}$$



The challenge

- Write a function that drives the robot in a straight line for a distance d. It should take the radius *r* of the wheel and the distance to travel. It should calculate the angles $\varphi_1\ and\ \varphi_2$ needed in order to drive the robot the specified distance. Make sure you use the parameters tacholimit and use Waitfor to make sure the motors reached the goal.
- Write a function that rotates the robot around its center by an angle θ. It should take the radius *r* of the wheel, the half wheel to wheel distance *l* and rotation angle θ. It should calculate the angles $\varphi_1\ and\ \varphi_2$ needed in order to drive the robot the specified distance. Make sure you use the parameters tacholimit and use Waitfor to make sure the motors reached the goal.

Tips

- You can create motor objects INSIDE the functions, they will not be affecting the motor objects you have in your script.
- Your script should drive the motor forward until the distance from the wall is out of the allowed tolerance.
- When this condition is met, the motors should stop and the robot should rotate around its center and go forward for a certain distance(20cm is a good guess). The sign of the angle of rotation is determined by the sign of the error of distance you calculate(Setpoint – MeasuredDistance).