Beginning Navigation

Robot Challenge 2011
Happylab

Simon Coupland
simonc@dmu.ac.uk

Overview

- Motivation
- Dead reckoning
- Odometry
- Mapping

Motivation

- Navigation is key problem in robotics
  - Where am I?
  - Where do I want to go?
  - How do I get there?

Motivation

- Navigation is a key problem in robotics
  - Where am I? (localization)
  - Where do I want to go? (reasoning about a goal)
  - How do I get there? (path planning)

  - These are all very challenging problems

Motivation

- Navigation is a key problem in robotics
  - Where am I? (localization)
  - Where do I want to go? (reasoning about a goal)
  - How do I get there? (path planning)

  - These are all very challenging problems
  - Uncertainties in:
    - Sensor readings
    - State measurements

Motivation

- Localization
  - Where am I?

  - This afternoon we'll answer:
    - Where am I given that I know where I've started?
    - How do I model the environment?

  - Happy to discuss more difficult questions

Dead Reckoning

- Traditional form of navigation
- Current position is based on previous position:

$$P_{t+1} = P_t + \ Movement$$

Dead Reckoning

- Traditional form of navigation
- Current position is based on previous position:

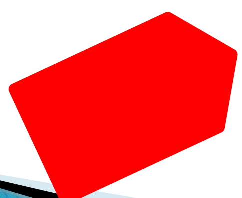
$$P_{t+1} = P_t + \ Movement$$

- For a robot:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} Movement\ in\ x \\ Movement\ in\ y \\ Rotation\ \theta \end{bmatrix}$$

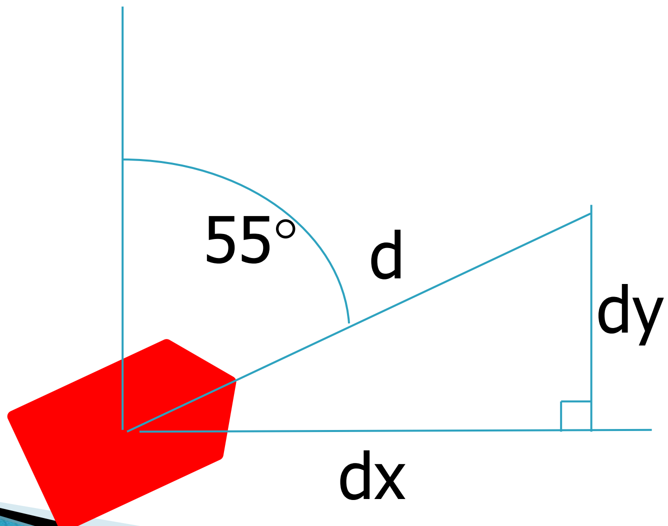
Dead Reckoning

- A boat is located at (45m, 22m)
- It's heading is 55 degrees
- It travels at 5m/s for 2 minutes
- Where is it now?



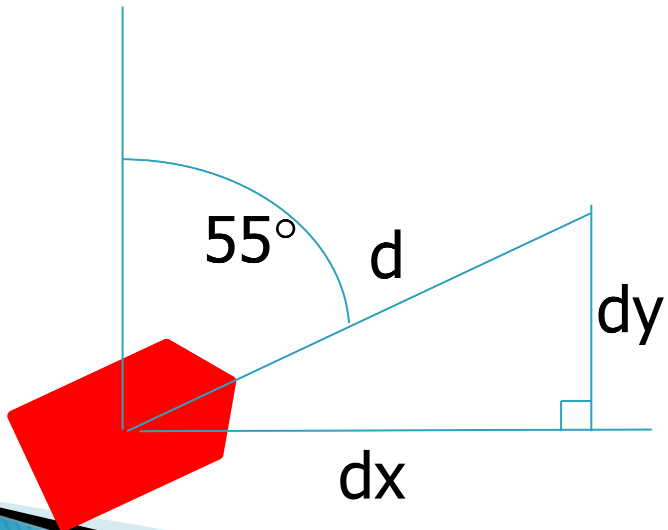
Dead Reckoning

- A boat is located at (45m, 22m)
- It's heading is 55 degrees
- It travels at 5m/s for 2 minutes
- Where is it now?



Dead Reckoning

- A boat is located at (45m, 22m)
- It's heading is 55 degrees
- It travels at 5m/s for 2 minutes
- Where is it now?



d = speed X time
d = 5 X 120
d = 600m

SOHCAHTOA

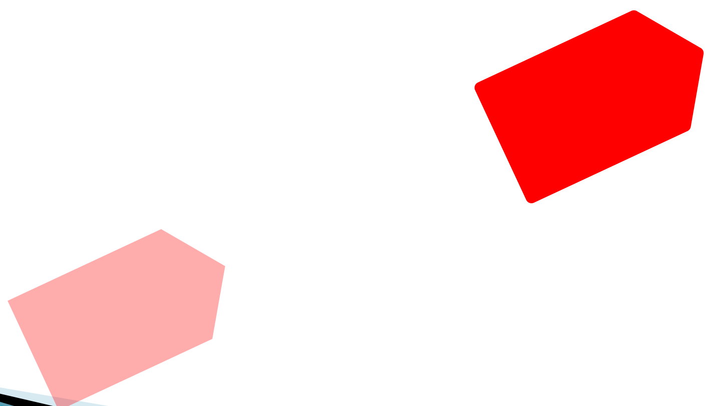
cos(35°) = dx / d
dx = cos(35°) X d
dx = 491.5m

sin(35°) = dy / d
dy = sin(35°) X d
dy = 344.1m

Dead Reckoning

- x‘ = x + dx
- x‘ = 45 + 491.5 = 536.5m

- y‘ = y + dy
- y‘ = 22 + 334.1 = 353.4m
- The boat is now located at (536.5m, 353.4m)



Dead Reckoning

- Exactly the same principle can be applied to a two wheeled robot
- The robot has a position and a heading:
  - (x,y,θ)

- Initially x, y and theta are zero – (0,0,0)

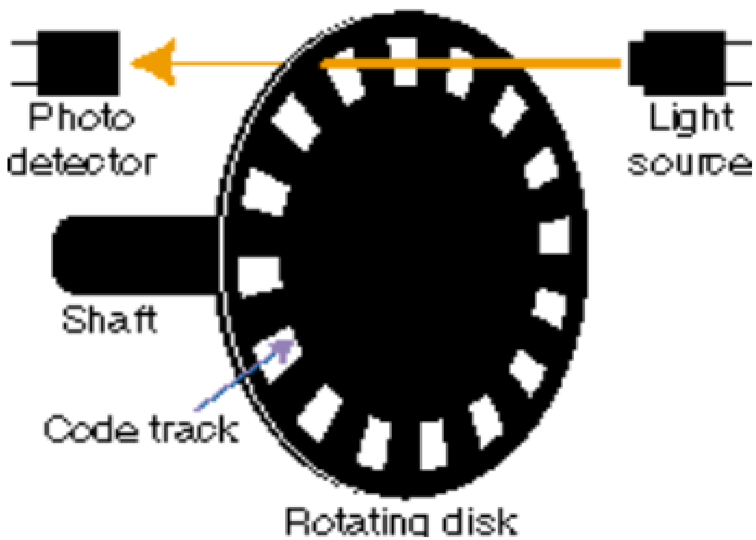
- Rather than speed we monitor the amount of rotation in each wheel

Rotation Encoding

- Required to measure the amount of rotation of each wheel
- Should be independent of motors
- Typically optical encoders are used

Encoder (optical)

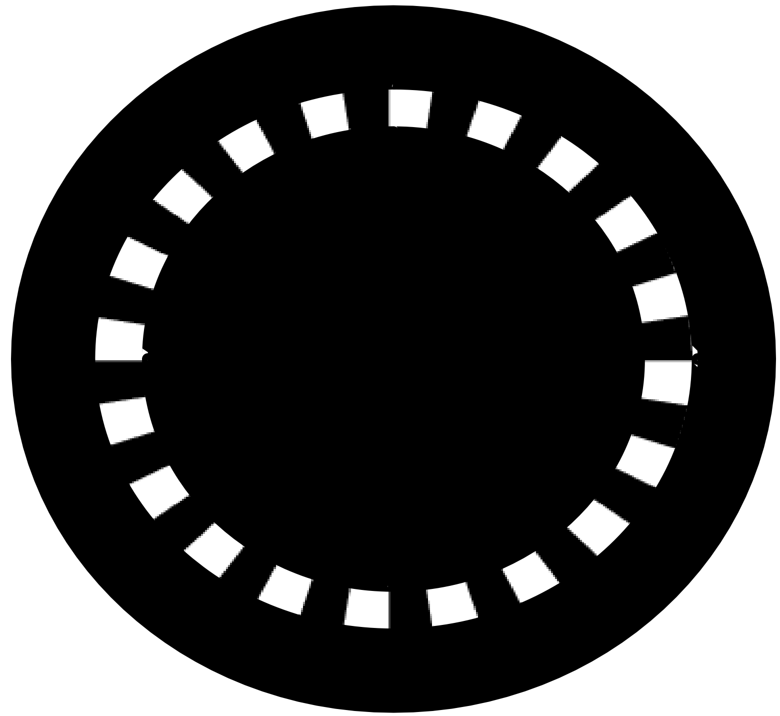
- Measure light through a disk:



- Each change of light level represents a set rotation of the shaft

Encoder (optical)

- 40 divisions
- Each change is light is 360/40 = 9 degrees
- Call these ticks or encoder ticks
- The number of ticks for one rotation is denoted t_r

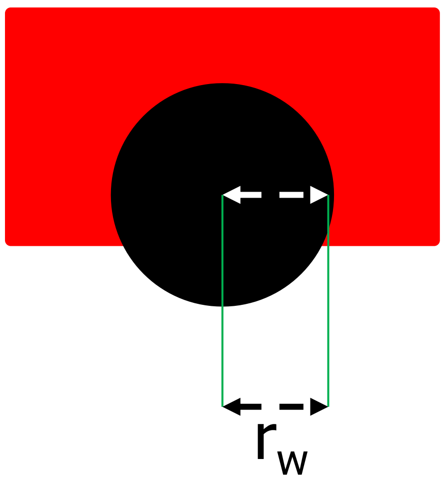


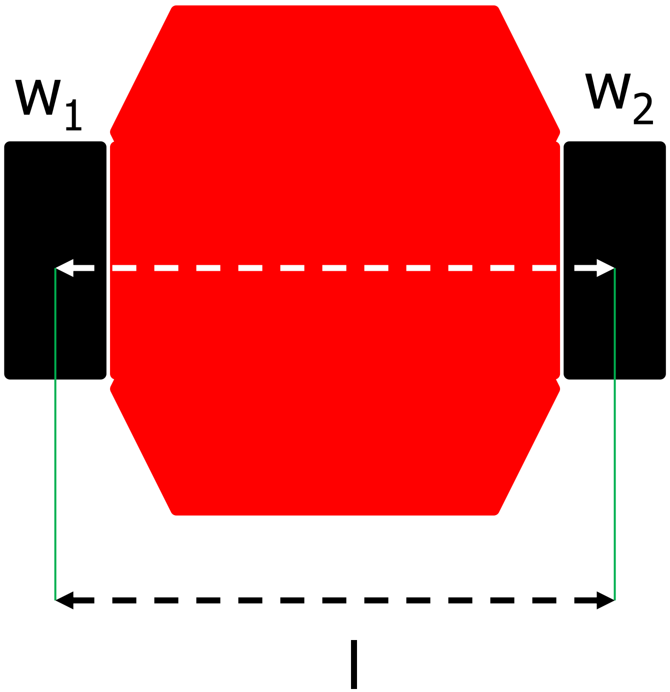
Our Robot

- What do we know?
  - Rotation of each wheel
  - Assume timing system

Our Robot

- What do we know?
  - Rotation of each wheel
  - Assume timing system
  - Physical dimensions:





Dead Reckoning

- Recall

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_\theta \end{bmatrix}$$

- Need to derive d_x, d_y, and d_θ

Dead Reckoning

- We know:
  - Radius and ∴ the circumference of the wheels ($2\pi r_w$)
  - The amount of rotation of each wheel (t_1 and t_2)
  - The number of ticks for one wheel rotation t_r

Dead Reckoning

- We know:
  - Radius and ∴ the circumference of the wheels ($2\pi r_w$)
  - The amount of rotation of each wheel (t_1 and t_2)
  - The number of ticks for one wheel rotation t_r
  - The distance travelled by each wheel is:

$$d_1 = 2\pi\, r_w \,\frac{t_1}{t_r}$$

$$d_2 = 2\pi\, r_w \,\frac{t_2}{t_r}$$

Dead Reckoning

- We know:
  - Radius and ∴ the circumference of the wheels ($2\pi r_w$)
  - The amount of rotation of each wheel (t_1 and t_2)
  - The number of ticks for one wheel rotation t_r
  - The distance travelled by each wheel is:

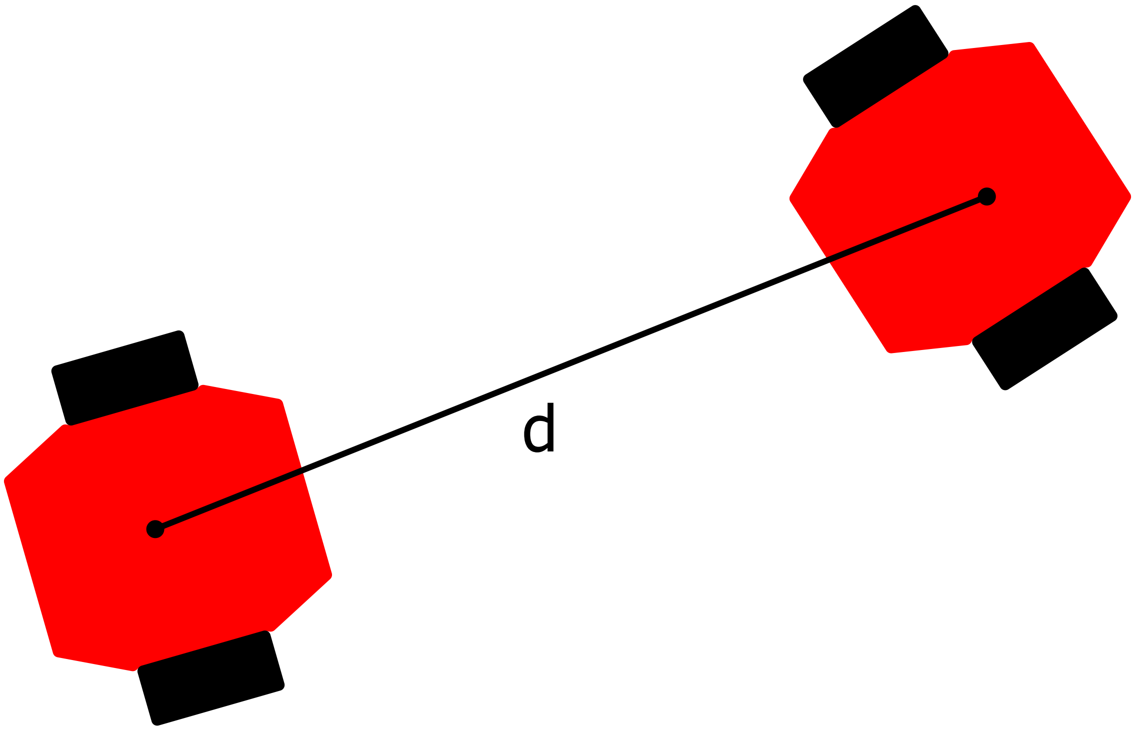
$$d_1 = 2\pi\, r_w \frac{t_1}{t_r}$$

$$d_2 = 2\pi\, r_w \frac{t_2}{t_r}$$

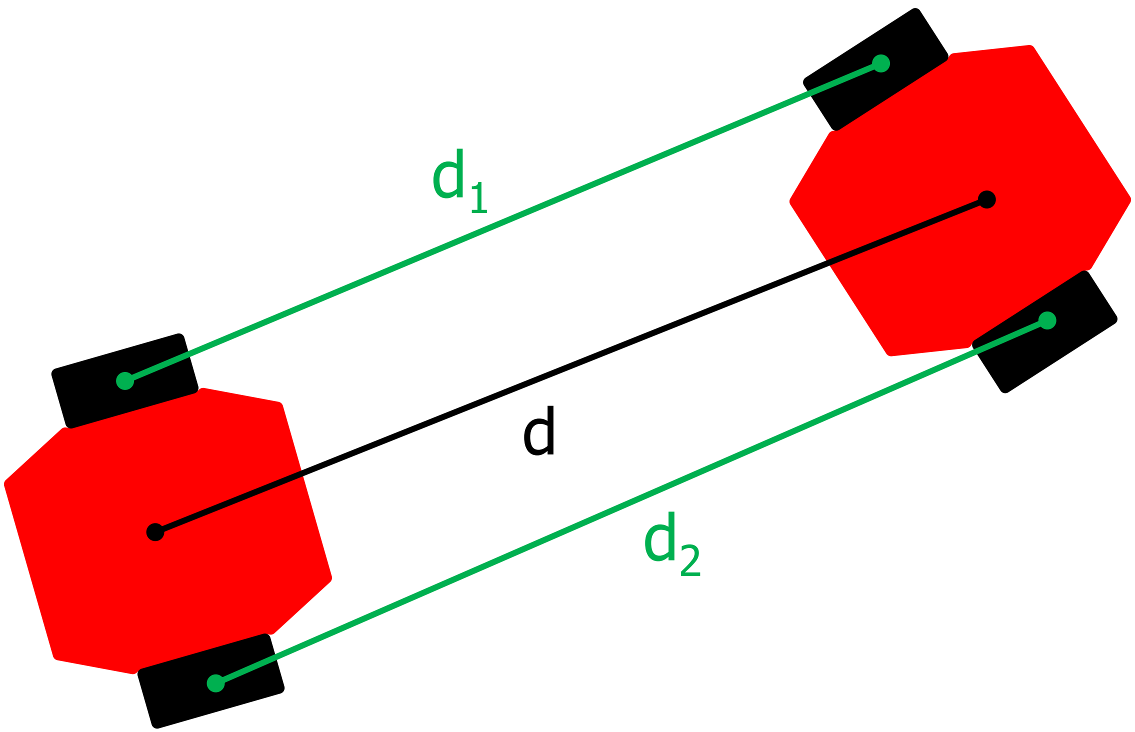
Wheel rotation as
A decimal

Dead Reckoning

- Distance (d) travelled by the robot:



Dead Reckoning



Given by the average distance travelled by the wheels

$$d = \frac{d_1 + d_2}{2}$$

Dead Reckoning

- Substitute in

$$d_1 = 2\pi\, r_w \frac{t_1}{t_r}$$

$$d = \frac{d_1 + d_2}{2}$$

$$d_2 = 2\pi\, r_w \frac{t_2}{t_r}$$

$$d = \frac{2\pi\, r_w \frac{t_1}{t_r} + 2\pi\, r_w \frac{t_2}{t_r}}{2}$$

Dead Reckoning

- Substitute in

$$d_1 = 2\pi\, r_w \frac{t_1}{t_r}$$

$$d = \frac{d_1 + d_2}{2}$$

$$d_2 = 2\pi\, r_w \frac{t_2}{t_r}$$

$$d = \frac{2\pi\, r_w\,(t_1 + t_2)}{2t_r}$$

Dead Reckoning

- Substitute in

$$d_1 = 2\pi\, r_w \frac{t_1}{t_r}$$

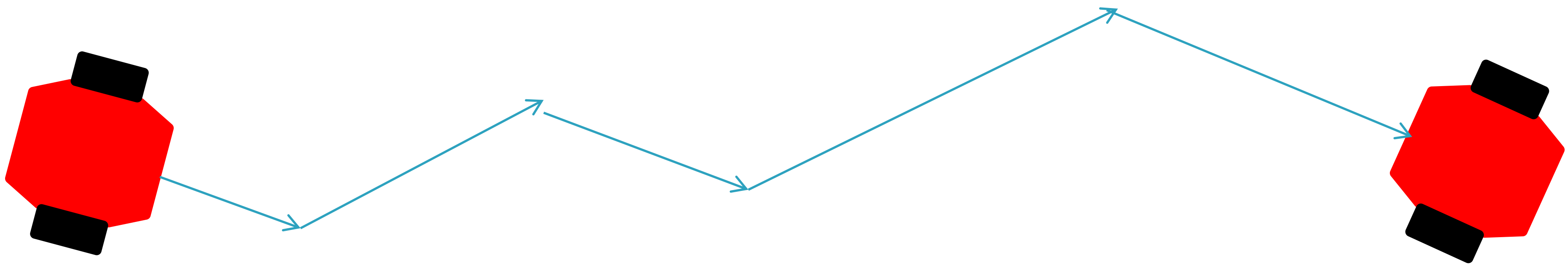
$$d = \frac{d_1 + d_2}{2}$$

$$d_2 = 2\pi\, r_w \frac{t_2}{t_r}$$

$$d = \frac{\pi\, r_w\, (t_1 + t_2)}{t_r}$$

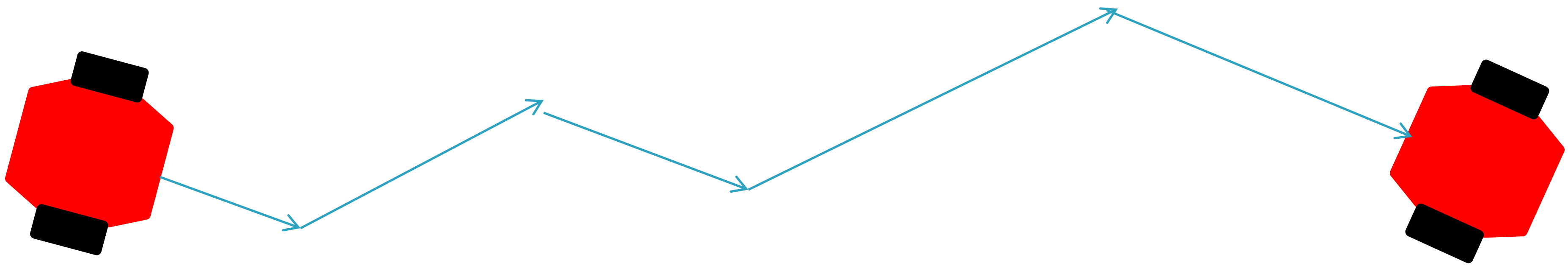
Dead Reckoning

- At each timestep we know how far we've travelled



Dead Reckoning

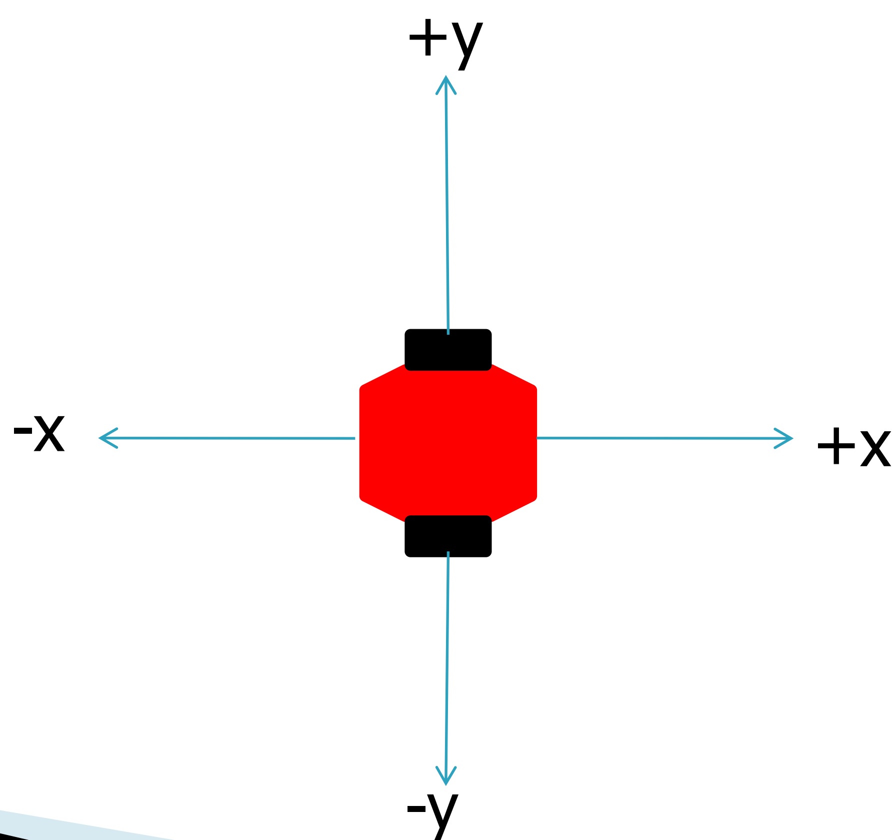
- At each timestep we know how far we've travelled



- But in which direction?

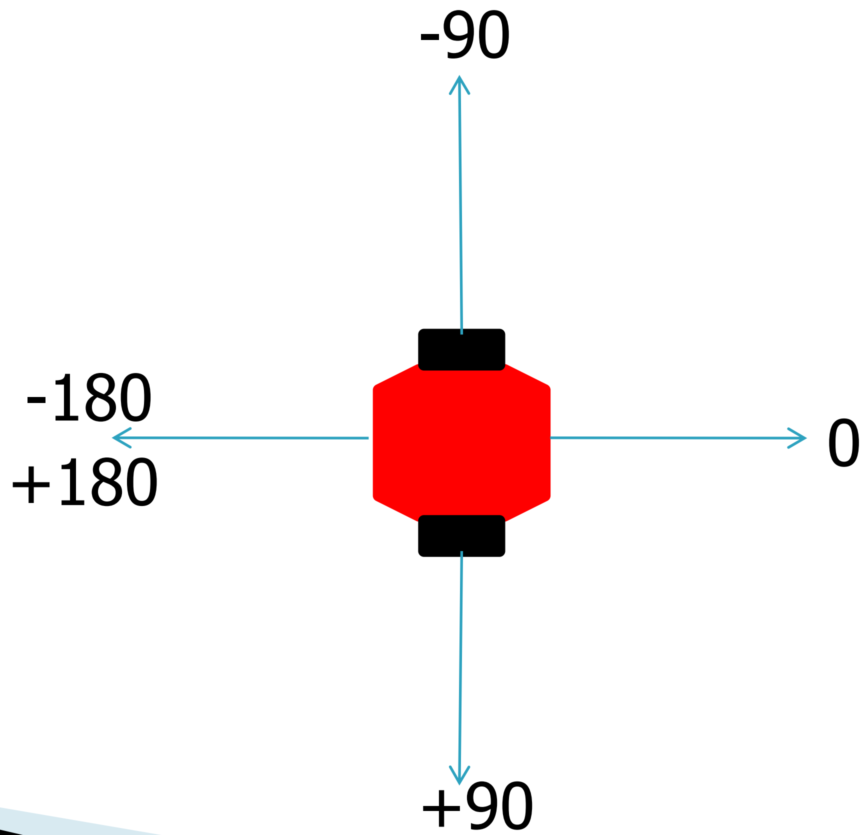
Robot Co-ord system

- Let's work in a 2D Cartesian space



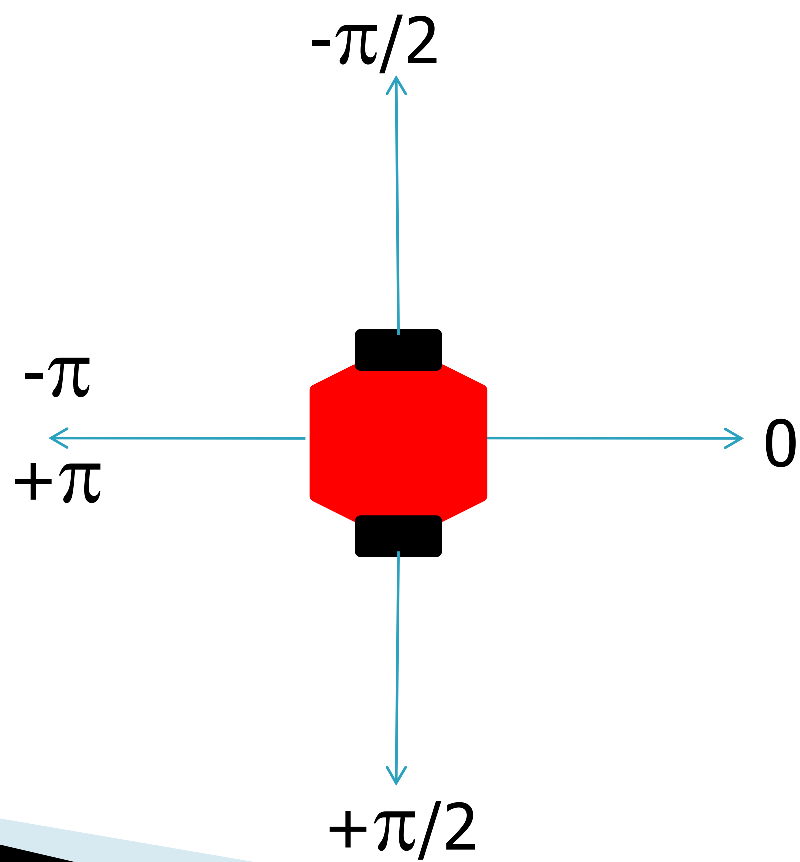
Robot Co-ord system

- Let's work in a 2D Cartesian space



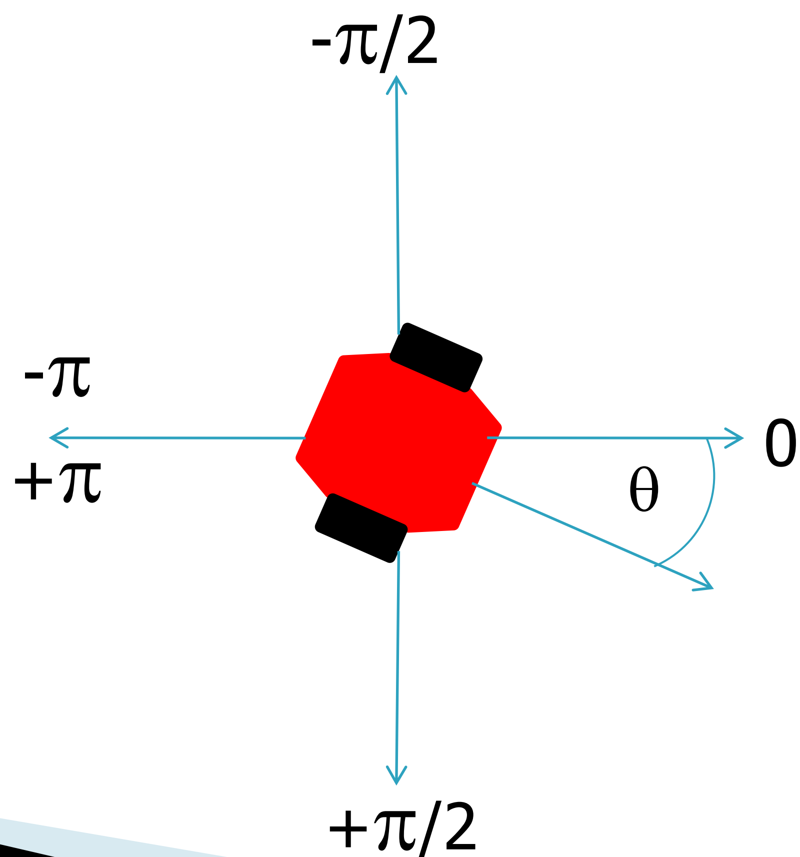
Robot Co-ord system

- Let’s work in a 2D Cartesian space



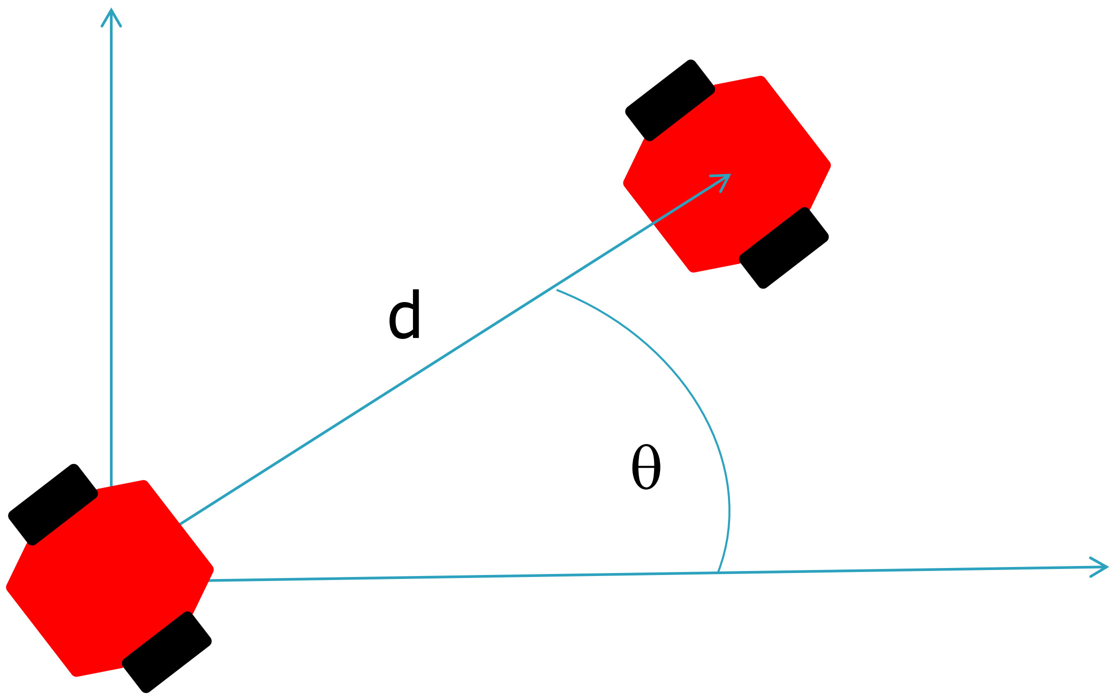
Robot Co-ord system

- Let's work in a 2D Cartesian space



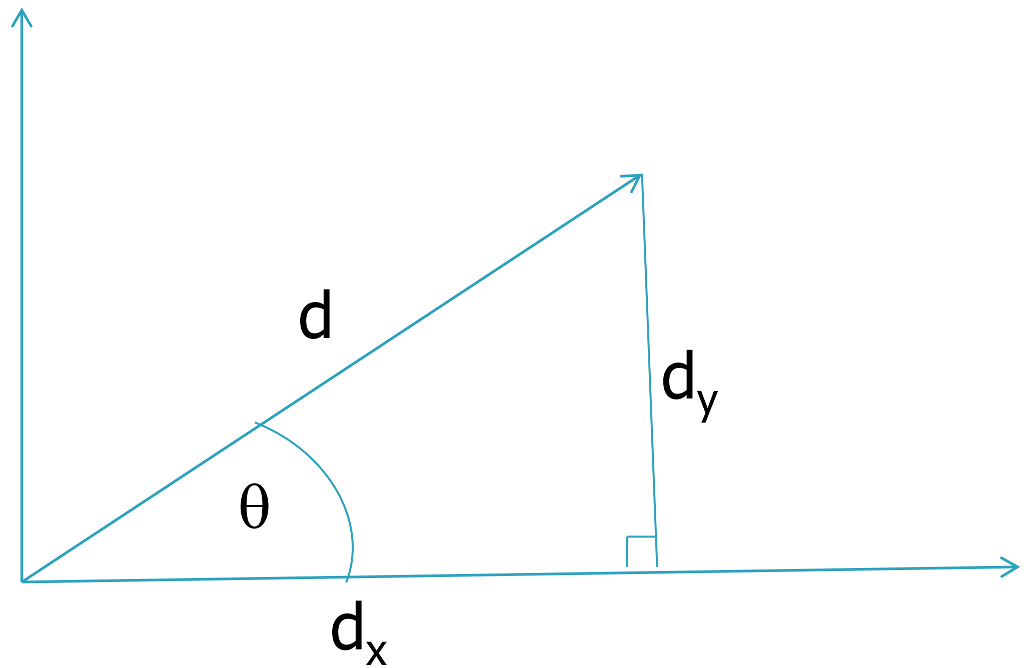
Dead Reckoning

- Simple trig to find x and y components of d



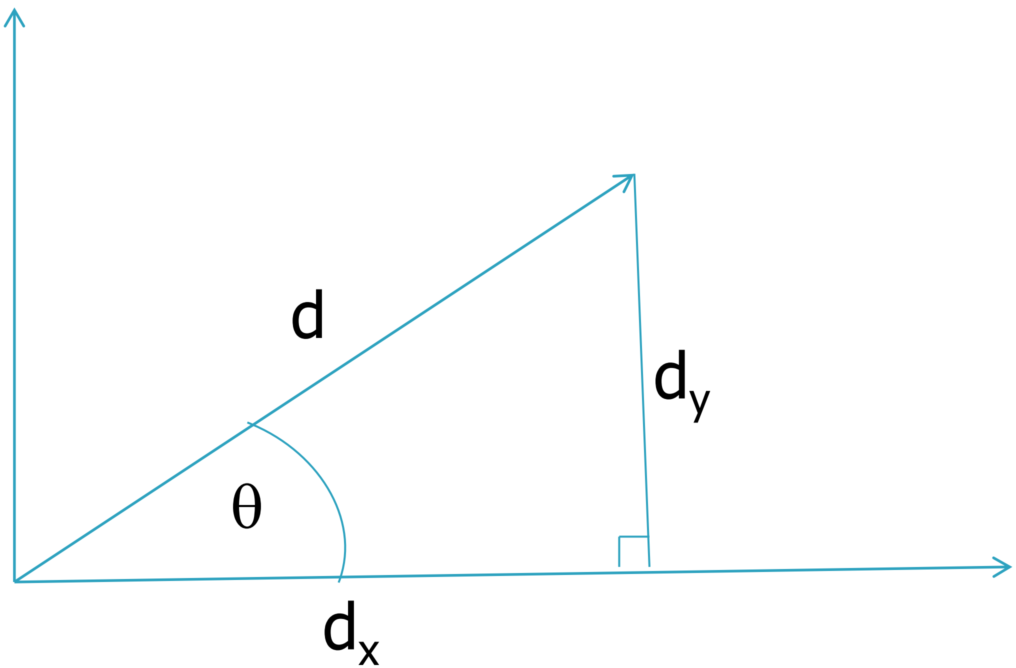
Dead Reckoning

- Simple trig to find x and y components of d



Dead Reckoning

- Simple trig to find x and y components of d



$$d_x = \cos(\theta) d$$

$$d_y = \sin(\theta) d$$

Dead Reckoning

- Simple trig to find x and y components of d

$$d_x = \cos(\theta)\,\frac{\pi\, r_w\,(t_1 + t_2)}{t_r}$$

$$d_y = \sin(\theta)\,\frac{\pi\, r_w\,(t_1 + t_2)}{t_r}$$

Dead Reckoning

- Simple trig to find x and y components of d

$$d_x = \cos(\theta)\frac{\pi r_w (t_1 + t_2)}{t_r}$$

Constants

$$d_y = \sin(\theta)\frac{\pi r_w (t_1 + t_2)}{t_r}$$

Dead Reckoning

- Simple trig to find x and y components of d

$$d_x = \cos(\theta)(t_1 + t_2)\frac{\pi r_w}{t_r}$$

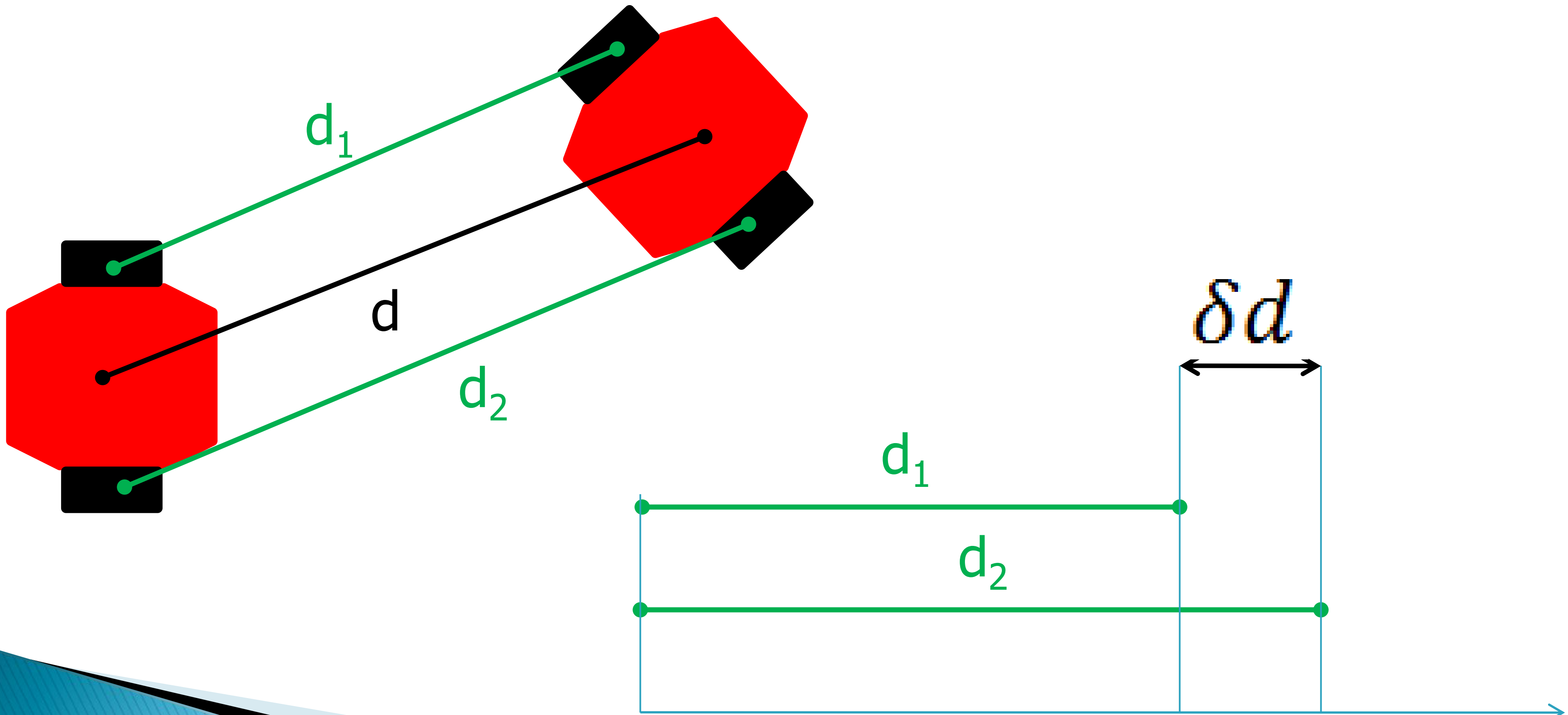
$$d_y = \sin(\theta)(t_1 + t_2)\frac{\pi r_w}{t_r}$$

Dead Reckoning

- So that just leaves rotation…

Dead Reckoning

- We need to look at the difference in distance travelled between the wheels



Dead Reckoning

- We need to look at the difference in distance travelled between the wheels

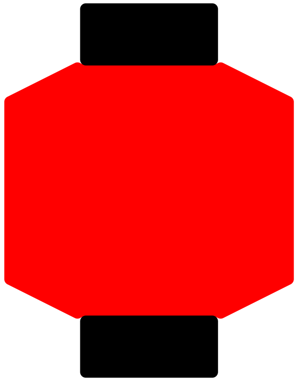
$$\delta d = d_2 - d_1$$

$$\delta d = 2\pi\, r_w \frac{t_2}{t_r} - 2\pi\, r_w \frac{t_1}{t_r}$$

$$\delta d = 2\pi\, r_w \frac{(t_2 - \ t_1)}{t_r}$$

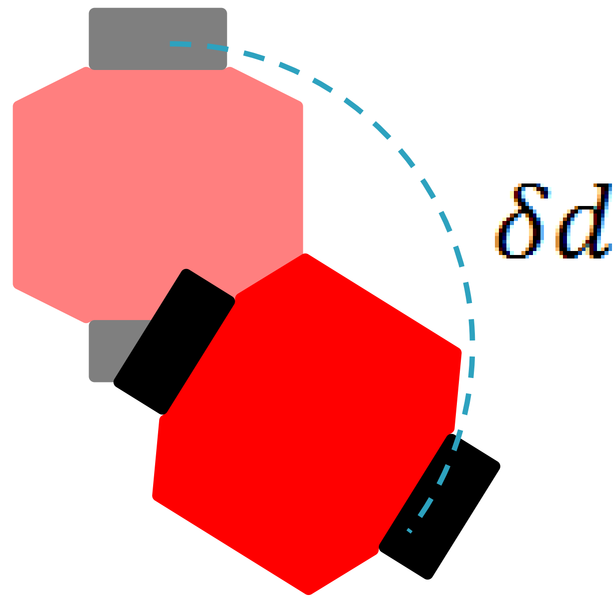
Dead Reckoning

- So what does this distance represent?



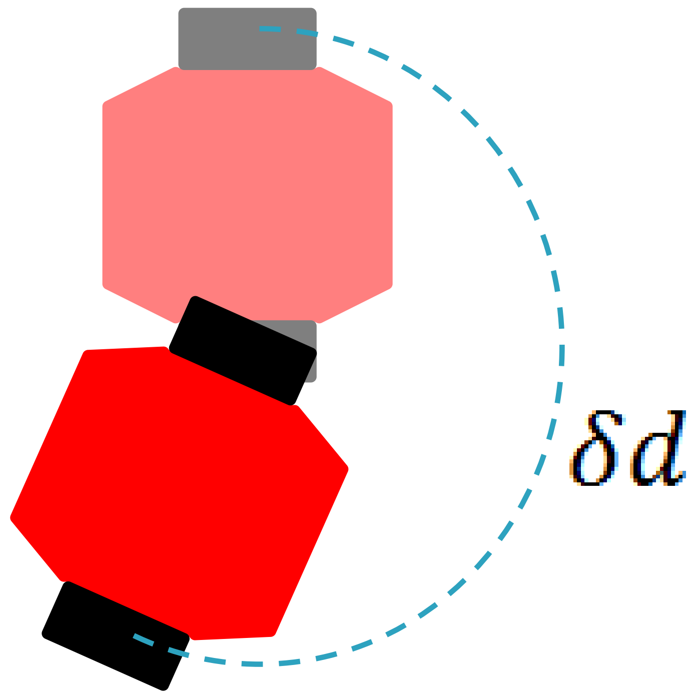
Dead Reckoning

- So what does this distance represent?



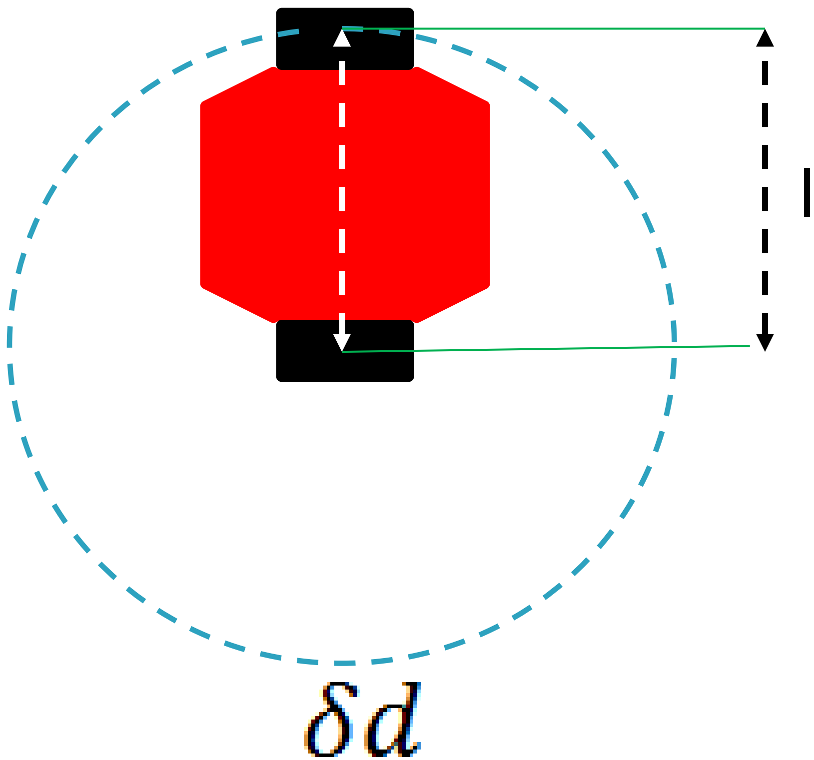
Dead Reckoning

- So what does this distance represent?



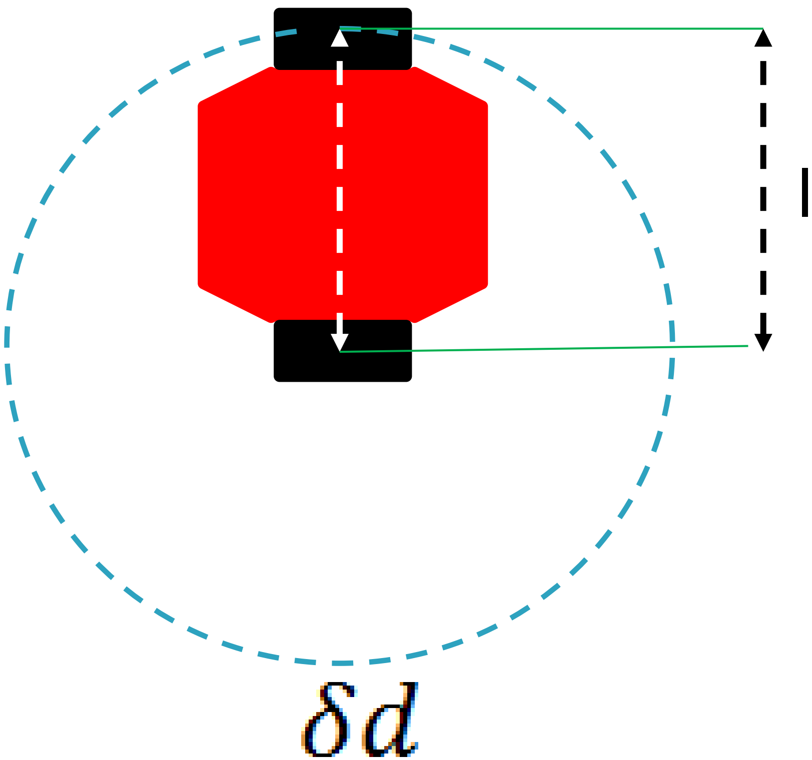
Dead Reckoning

- So what does this distance represent?



Dead Reckoning

- So what does this distance represent?



For one rotation:

$$\delta d = 2\pi l$$

$$2\pi l = 2\pi\, r_w \frac{(t_2 - t_1)}{t_r}$$

Dead Reckoning

- Divide both sides by $2\pi l$ to get a decimal description of amount the rotation ([0,1])

$$t = \frac{2\pi\, r_w \dfrac{(t_2 - t_1)}{t_r}}{2\pi l}$$

$$t = \frac{2\pi\, r_w (t_2 - t_1)}{2\pi l\, t_r}$$

$$t = \frac{r_w (t_2 - t_1)}{l\, t_r}$$

Dead Reckoning

- Multiply by 2π to get rotation of the robot in radians

$$t = \frac{r_w\,(t_2 - t_1)}{l\,t_r}$$

$$d\theta = 2\pi\,\frac{r_w\,(t_2 - t_1)}{l\,t_r}$$

$$d\theta = (t_2 - t_1)\,\frac{2\pi\,r_w}{l\,t_r}$$

Odometry System

- So

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_\theta \end{bmatrix}$$

- Where

$$d_x = \cos(\theta)(t_1 + t_2)\frac{\pi\, r_w}{t_r}$$

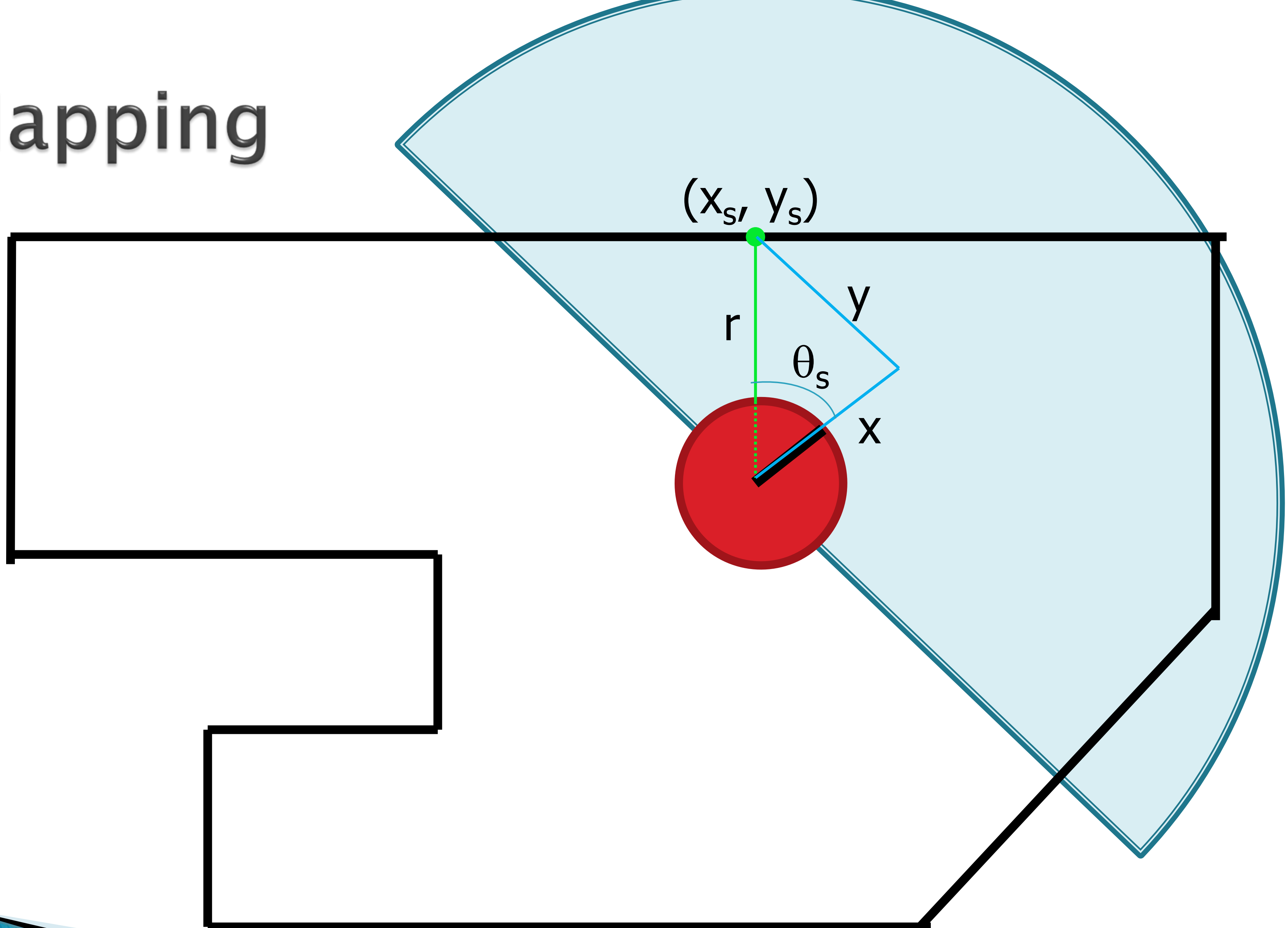
$$d_y = \sin(\theta)(t_1 + t_2)\frac{\pi\, r_w}{t_r}$$

$$d\theta = (t_2 - t_1)\,\frac{2\pi\, r_w}{l\, t_r}$$

Simple Mapping

- We now explain the construction of simple maps
- Let's assume we have an ultrasonic sensor (sonar)
  - Distance to object, denoted r
  - Angle to object, denoted θ_s
    - Either from
      - Sonar array calculation
      - Physical position of sonar sensor

Mapping



Simple Mapping

- Assume the robot is at (0,0,0)
- (x_s, y_s) can be found with simple trig

$$x_s = \cos(\theta_s)\, r$$

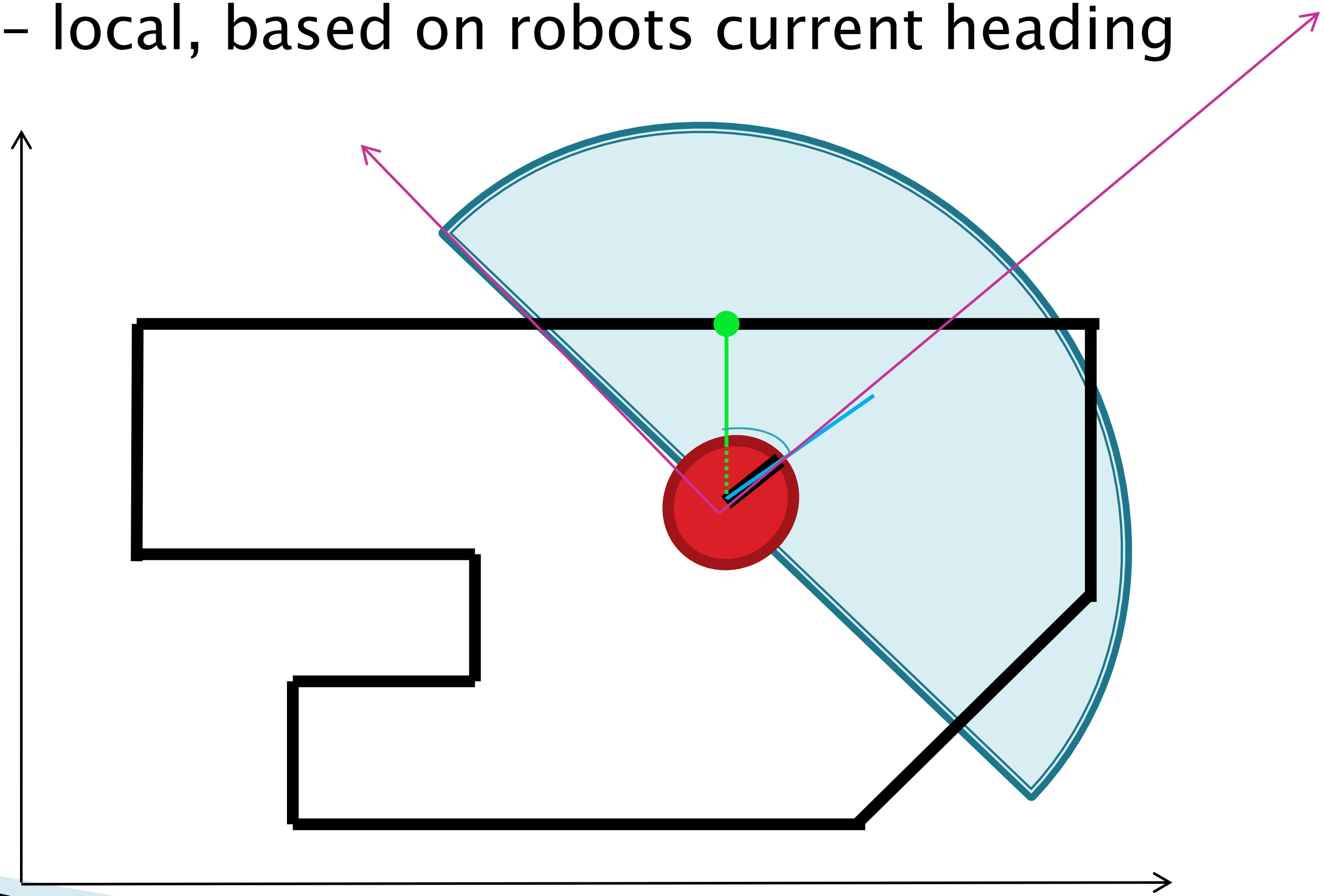
$$y_s = \sin(\theta_s)\, r$$

Simple Mapping

- Of course the robot is not at (0,0,0)
- We need to project the sonar reading from the robots real position (x_r, y_r, θ_r)

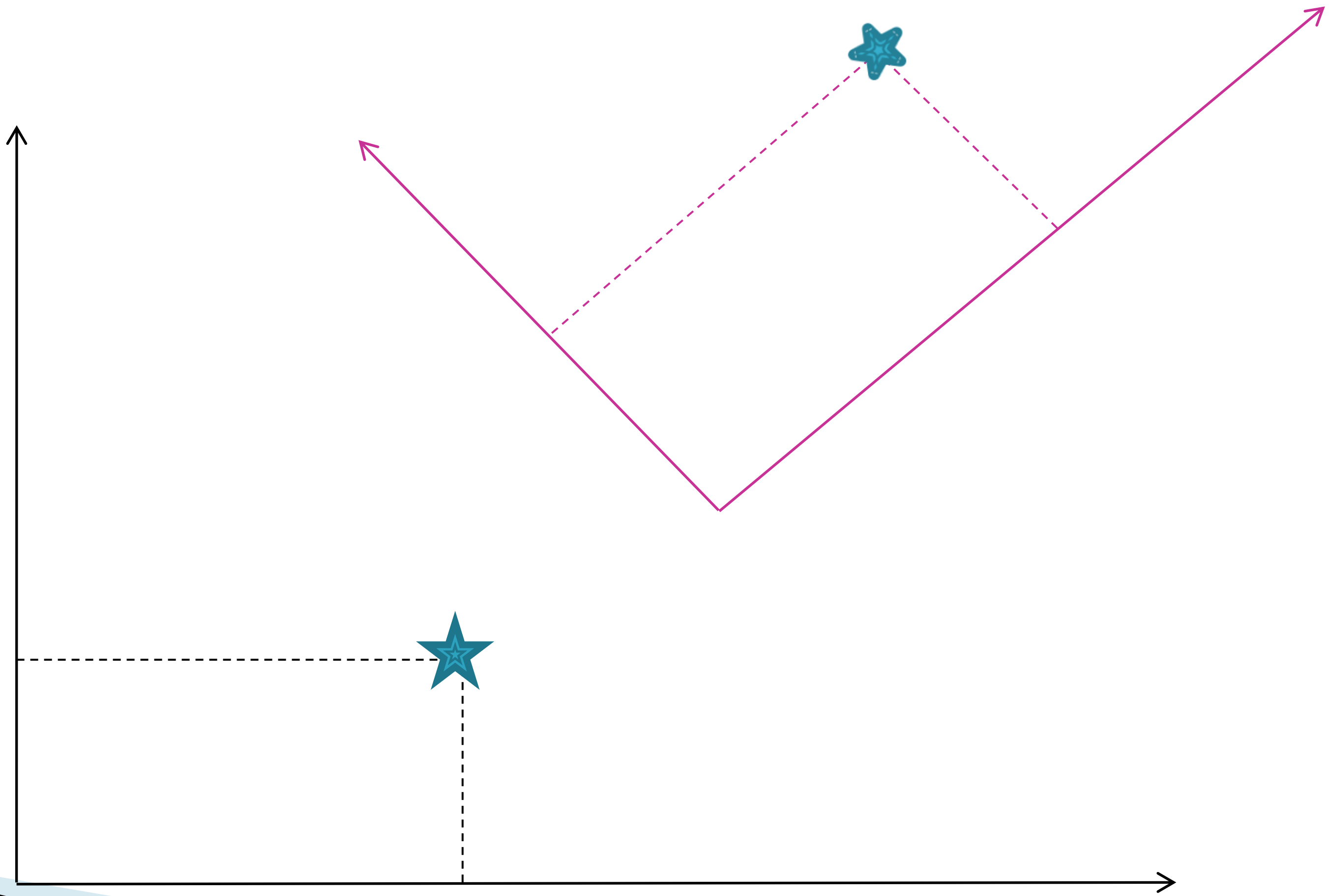
Simple Mapping

- Working in two co-ordinate systems:
  - Map – global
  - Robot – local, based on robots current heading



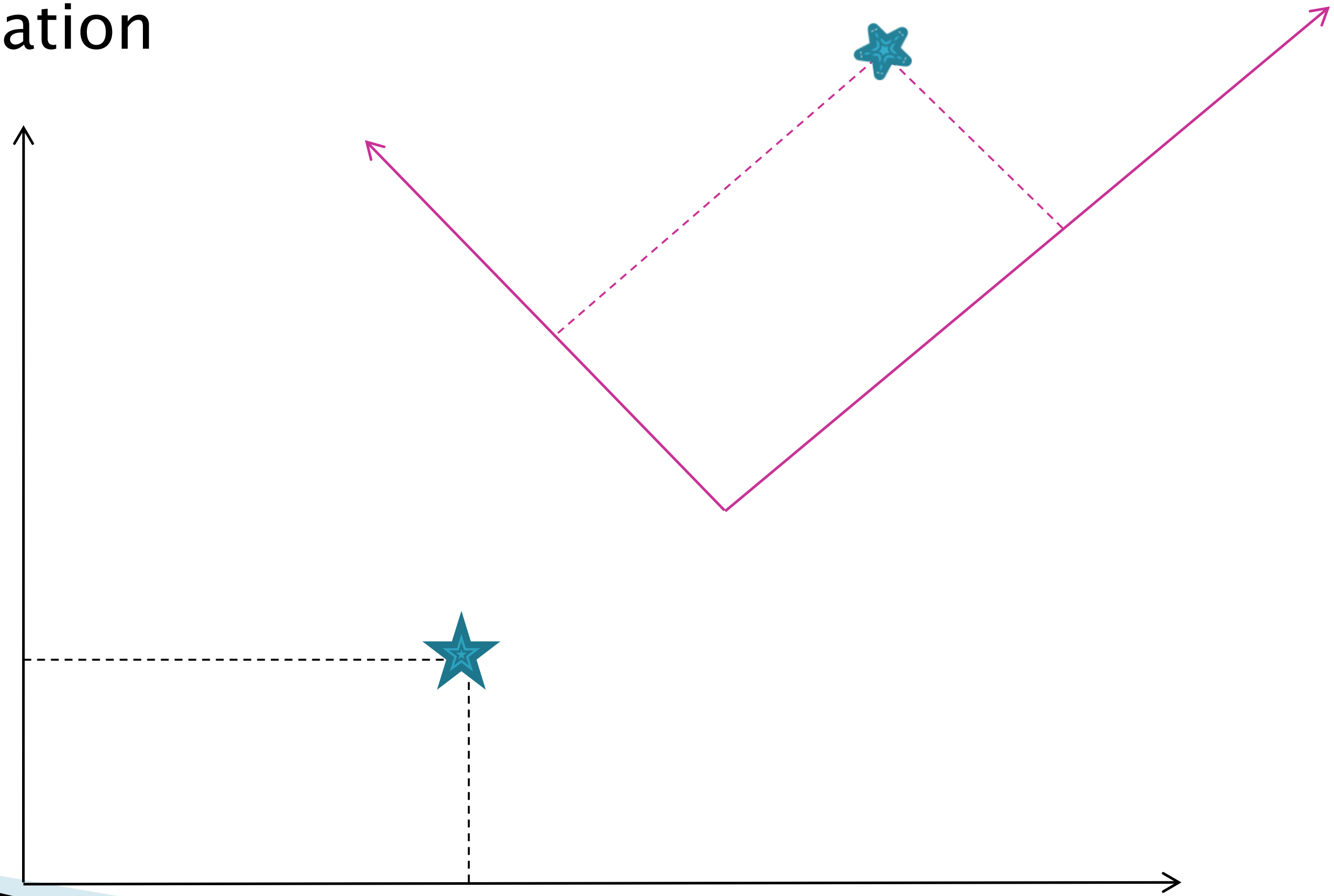
Simple Mapping

- Working in two co-ordinate systems:
  - Position (6, 3) in both systems



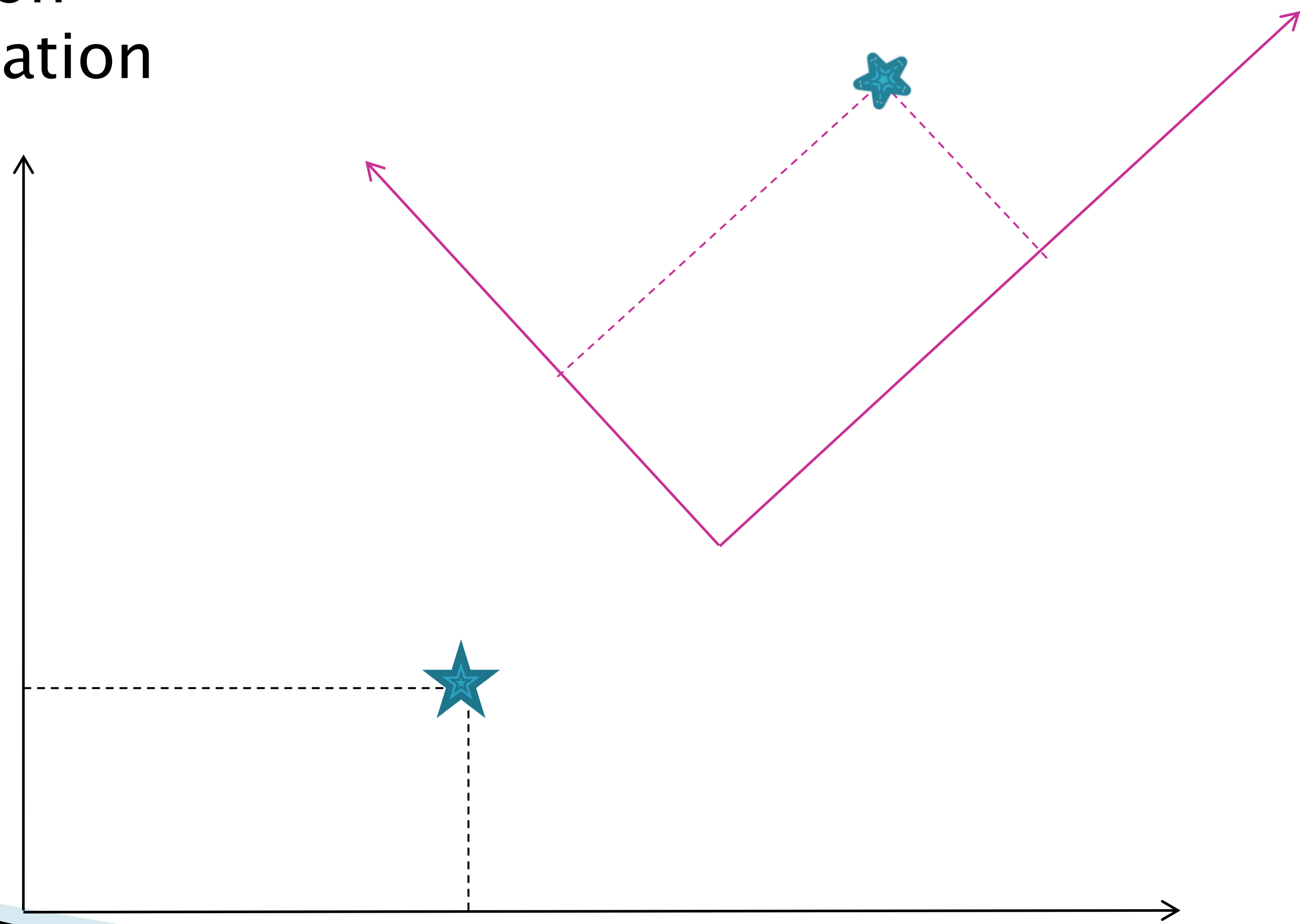
Mapping

- To move between we need:
  - Rotation
  - Translation



Simple Mapping

- To move between we need:
  - Rotation
  - Translation



Simple Mapping

- This gives us an algorithm:
  - Find the position of each sonar reading in the robots co-ordinate system
  - Rotate all calculate positions by the robots heading
  - Translate all rotated co-ordinates by the robots position
- All readings are now in the global co-ordinate system

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- $r = 1.1$
- $\theta_s = 1.35$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- $r = 1.1$, robot radius $= 0.6$
- $\theta_s = 1.35$

- Step one – find x and y
- $x = \cos(\theta_s) \times (r + \text{robot radius})$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- r = 1.1, robot radius = 0.6
- $\theta_s = 1.35$

- Step one – find x and y
- x = $\cos(\theta_s)$ X (r + robot radius)
- x = cos(1.35) X (1.1 + 0.6)
- x = 0.22 X 1.7
- x = 0.37

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- $r = 1.1$, robot radius $= 0.6$
- $\theta_s = 1.35$

- Step one – find x and y
- $y = \sin(\theta_s) \text{ X } (r + \text{robot radius})$
- $y = \sin(1.35) \text{ X } (1.1 + 0.6)$
- $y = 0.98 \text{ X } 1.7$
- $y = 1.66$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- r = 1.1, robot radius = 0.6
- $\theta_s = 1.35$
- (x,y) = (0.37, 1.66)

- Step two – rotate (x,y) to global co-ord system

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix}$$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- r = 1.1, robot radius = 0.6
- $\theta_s = 1.35$
- (x,y) = (0.37, 1.66)

- Step two – rotate (x,y) to global co-ord system

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.37 \\ 1.66 \end{bmatrix} \begin{bmatrix} 0.98 & 0.2 \\ -0.2 & 0.98 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.37 \times 0.98 + 0.37 \times 0.2 \\ 1.66 \times -0.2 + 1.66 \times 0.98 \end{bmatrix}$$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- $r = 1.1$, robot radius $= 0.6$
- $\theta_s = 1.35$
- $(x,y) = (0.37, 1.66)$

- Step two – rotate (x,y) to global co-ord system

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.37 \\ 1.66 \end{bmatrix} \begin{bmatrix} 0.98 & 0.2 \\ -0.2 & 0.98 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.37 \times 0.98 + 1.66 \times 0.2 \\ 0.37 \times -0.2 + 1.66 \times 0.98 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0.70 \\ 1.55 \end{bmatrix}$$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- r = 1.1, robot radius = 0.6
- $\theta_s = 1.35$
- (x,y) = (0.37, 1.66), (x',y') = (0.70,1.55)

- Step three – translate (x',y') to global co-ord system

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' + x_r \\ y' + y_r \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 0.70 + 12.6 \\ 1.55 + 4.3 \end{bmatrix}$$

Worked Example

- $(x_r, y_r, \theta_r) = (12.6, 4.3, -0.2)$
- r = 1.1, robot radius = 0.6
- $\theta_s = 1.35$
- (x,y) = (0.37, 1.66), (x',y') = (0.70,1.55)

- Step three – translate (x',y') to global co-ord system

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' + x_r \\ y' + y_r \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 13.3 \\ 5.85 \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 0.70 + 12.6 \\ 1.55 + 4.3 \end{bmatrix}$$

Demo

- Simulated pioneer 3dx
- Aria library

Summary

- Robot position:
  - Previous position + change
- Mapped obstacles
  - Project sonar readings from calculate robot position
- Assumes a lack of uncertainties