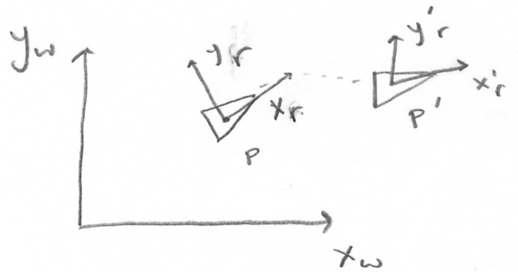
Odometry



Suppose we have a pose in our reference coordinate frame

$$p = X^w_r = \begin{bmatrix} x_w \\ y_w \\ \theta_w \end{bmatrix}$$

Odometry measures relative motion frome pose to p', i.e. $X^r_{r'}$

$$\Delta = x^r_{r'} = [\Delta x, \Delta y, \Delta\theta]^T$$

$$p' = p \oplus \Delta$$

$\oplus$ is the compound, or "head-to-tail" operator to add our special vectors, taking into account the rotation

$$x^w_{r'} = x^w_r \oplus x^r_{r'}$$

$$= \begin{bmatrix} x_r + \Delta x \cos\theta - \Delta y \sin\theta \\ y_r + \Delta x \sin\theta + \Delta y \cos\theta \\ \theta + \Delta\theta \end{bmatrix}$$

We measure Δ with the wheel encoders and (potentially) other sensors (gyro). we can measure the distance each wheel travels in a timestep by measuring the encoder ticks in that time step

$\pi D \rightarrow$ linear distance traveled by wheel per revolution

$$\frac{48 \text{ ticks}}{\text{motor revolution}} \times \frac{20.4 \text{ motor revolutions}}{\text{wheel rev}} \times \frac{1 \text{ wheel rev}}{\pi D \text{ meters}} = 3246.76 \frac{\text{ticks}}{\text{meter}}$$

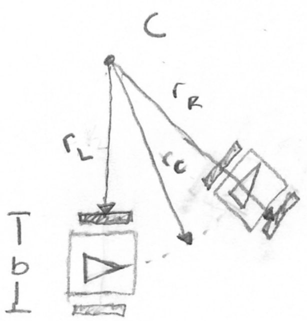
$$D = 0.08 \text{ m}$$

Differential Drive Kinematics

Given d_L & d_R over some Δt, where did robot go?

→ Assume $P_0 = [0, 0, 0]^T$ i.e. the initial pose serves as the world reference frame. (if using MoCap, then get inital pose from the data packets

→ Assume robot motion is an arc of constant curvature (reasonable for $\Delta t \to 0$)



b → baseline

Consider arc radius for each wheel, angle swept by d_L & d_R must be the same.

$$\Delta\theta = \frac{d_L}{r_L} = \frac{d_R}{r_R} = \frac{d_R}{r_L + b}$$

$$(r_L + b)\, d_L = d_R r_L$$

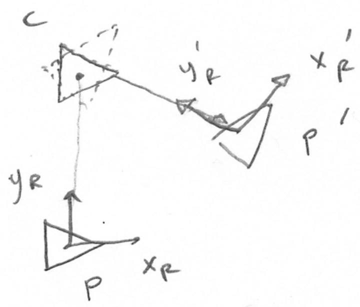
$$(d_R - d_L)\, r_L = b\, d_L$$

$$r_L = \frac{b\, d_L}{d_R - d_L} \quad \text{thus} \quad \boxed{\Delta\theta = \frac{d_L}{r_L} = \frac{d_R - d_L}{b}}$$

$$r_c = \frac{r_L + r_R}{2} \qquad r_L = \frac{d_L}{\Delta\theta}, \quad r_R = \frac{d_R}{\Delta\theta}$$

so $$r_c = \frac{d_L + d_R}{2} \cdot \frac{1}{\Delta\theta}$$

think of geometry as kinematic chain



In the robot frame we have

$$T_y R_\theta T_{y'}$$

for position:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & r_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta & 0 \\ \sin\Delta\theta & \cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -r_c \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta & r_c \sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta & -r_c\cos\Delta\theta + r_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} r_c \sin\Delta\theta \\ -r_c \cos\Delta\theta + r_c \\ 1 \end{bmatrix}$$

we also have a change in orientation $\Delta\theta$
so our pose vector becomes

$$x^r_{r'} = \begin{bmatrix} 0 \\ r_c \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 0 \\ \Delta\theta \end{bmatrix} \oplus \begin{bmatrix} 0 \\ -r_c \\ 0 \end{bmatrix} = \begin{bmatrix} r_c \sin\Delta\theta \\ -r_c\cos\Delta\theta + r_c \\ \Delta\theta \end{bmatrix}$$

using small angle approximation $\sin\Delta\theta \approx \Delta\theta$ $\cos\Delta\theta \approx 1$

$$x^r_{r'} = \begin{bmatrix} r_c \Delta\theta \\ 0 \\ \Delta\theta \end{bmatrix} = \begin{array}{c} \begin{bmatrix} (d_R + d_L)/2 \\ 0 \\ (d_R - d_L)/b \end{bmatrix} \end{array} \begin{array}{l} \leftarrow \Delta x \\ \leftarrow \Delta y \\ \leftarrow \Delta\theta \end{array}$$

odometry model

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ -\frac{1}{b} & \frac{1}{b} & 0 \end{bmatrix} \begin{bmatrix} d_L \\ d_R \\ d_S \end{bmatrix} \leftarrow$$ models non-ideal motion, side slip can set to $\underline{\underline{0}}$

Sources of error

Systematic Errors

→ unequal wheel diameters

→ average of both wheels differs from nominal

→ misalignment of wheels

→ uncertainty in wheelbase

→ limited encoder resolution

Non-systematic errors

→ travel on uneven floor

→ hitting unexpected obsticles

→ wheel slippage

- slippery floors
- over-acceleration
- fast turning (skidding)
- external forces
- non-point contact with floor

See Borenstein Paper on Google Drive for more info on calibrating out systematic errors.

Gyrodometry

for a time interval Δt define:

$\Delta_{odo,i}$ ⇒ change in robot heading based on encoders only

$\Delta_{gyro,i}$ ⇒ chang in heading based on IMU

Gyrodometry uses simple logic to account for time steps where encoder data is <u>bad</u> i.e. hitting a bump in the floor and slipping.

gyro data will typically be less accurate than encoder data <u>except</u> when encoder data is <u>bad</u>

Algorithm:

define $\Delta_{G\cdot O,i} = \Delta_{gyro,i} - \Delta_{odo,i}$

if $|\Delta_{G\cdot O,i}| > \Delta\Theta_{thresh}$ ← user defined

then $\Delta\Theta_i = \Delta\Theta_{gyro,i}$

else $\Delta\Theta_i = \Delta\Theta_{odo,i}$

See Borenstein paper #2 on google drive for more info.