Robot motion

Javier González Jiménez

Reference Books:

- Probabilistic Robotics. S. Thrun, W. Burgard, D. Fox. MIT Press. 2001
- Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods. Juan-Antonio Fernández-Madrigal and José Luis Blanco Claraco. IGI-Global. 2013.

Content

- Locomotion of wheeled robots
  - Differential drive
- Pose of the robot through composition of poses
  - Composition
  - Pose as a rigid transformation
    - Inverse of a pose
    - Concatenation of poses
- Probabilistic motion model
  - Velocity-based
  - Odometry-based

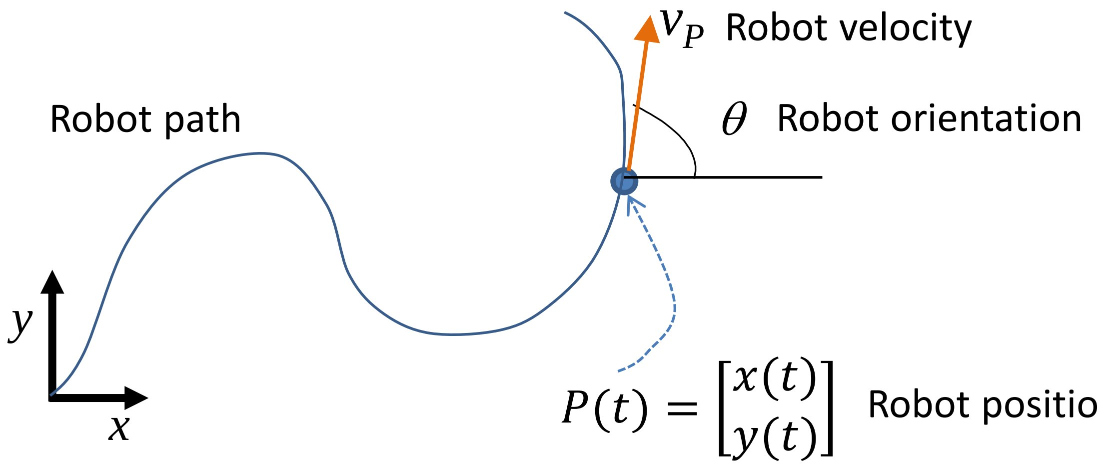
Locomotion of Wheeled Robots

Locomotion: The act of moving from one place to another

Robot pose: position and orientation

$$pose = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

P: position



$$P(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$ Robot position at time t

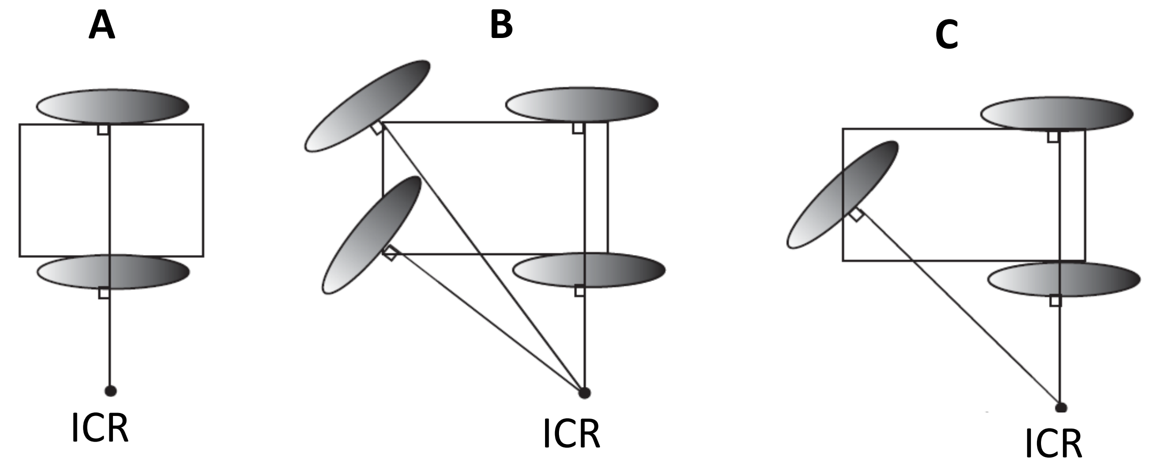
- A vehicle is holonomic if the number of its **local d**egrees-**o**f-**f**reedom (dof) of movement equals the number of **global** dof (3 for planar motion, 6 in 3D space)
- If the vehicle can not move in some direction (has some motion constraint) it becomes non-holonomic

Typically, a non-holonomic vehicle is that whose velocity vector $\boldsymbol{v_P}$ is restricted to be **tangent to the path** → moves on a circular trajectory and cannot move sideways

Locomotion of Wheeled Robots

Two main motion systems in mobile robotics:

- Differential drive (A) ⟵ We'll focus on this
- Ackermann steering: Car drive (B) Bicycle (C)



ICR: Instantaneous Center of Rotation

All these are Non-holonomic vehicles:

- 2 local dof: 2 actuators/motors
- 3 global dof: $[x, y, \theta]$

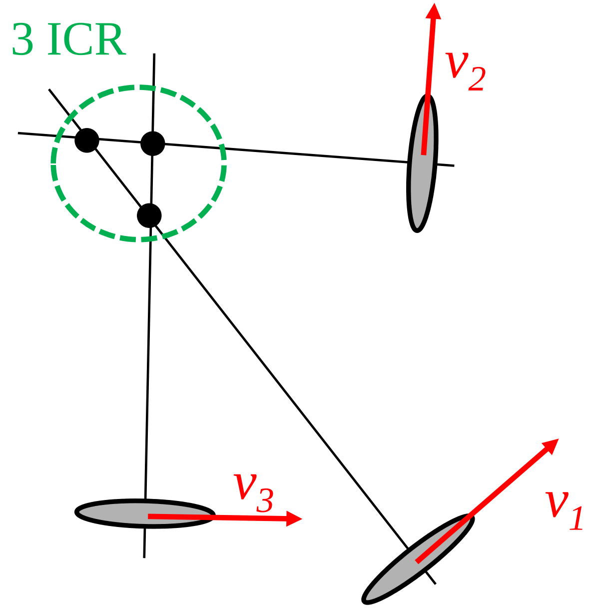
local dof (2) < global dof (3)

If not wheel slippage the motion is always a pure rotation about ICR:

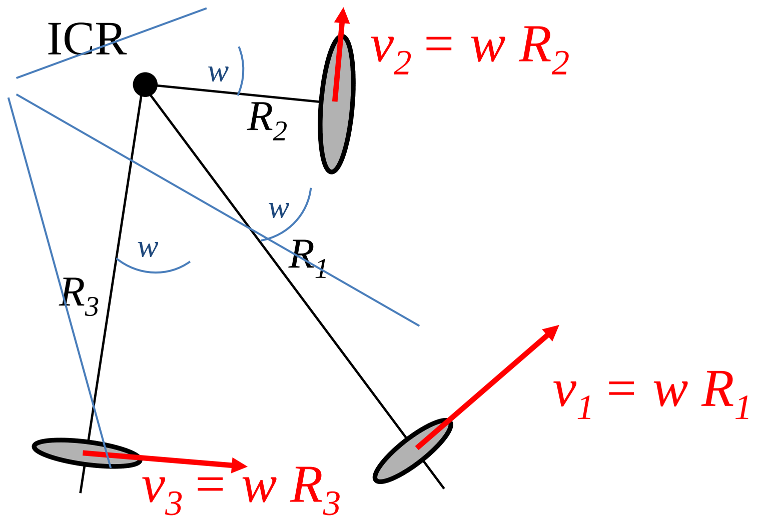
- the robot is always describing a circular motion
- the center of rotation (ICR) moves at each instant of time

Locomotion of Wheeled Robots

Rolling *vs.* slippage (vehicle with 3 wheels)



Wheels will slip: not single ICR

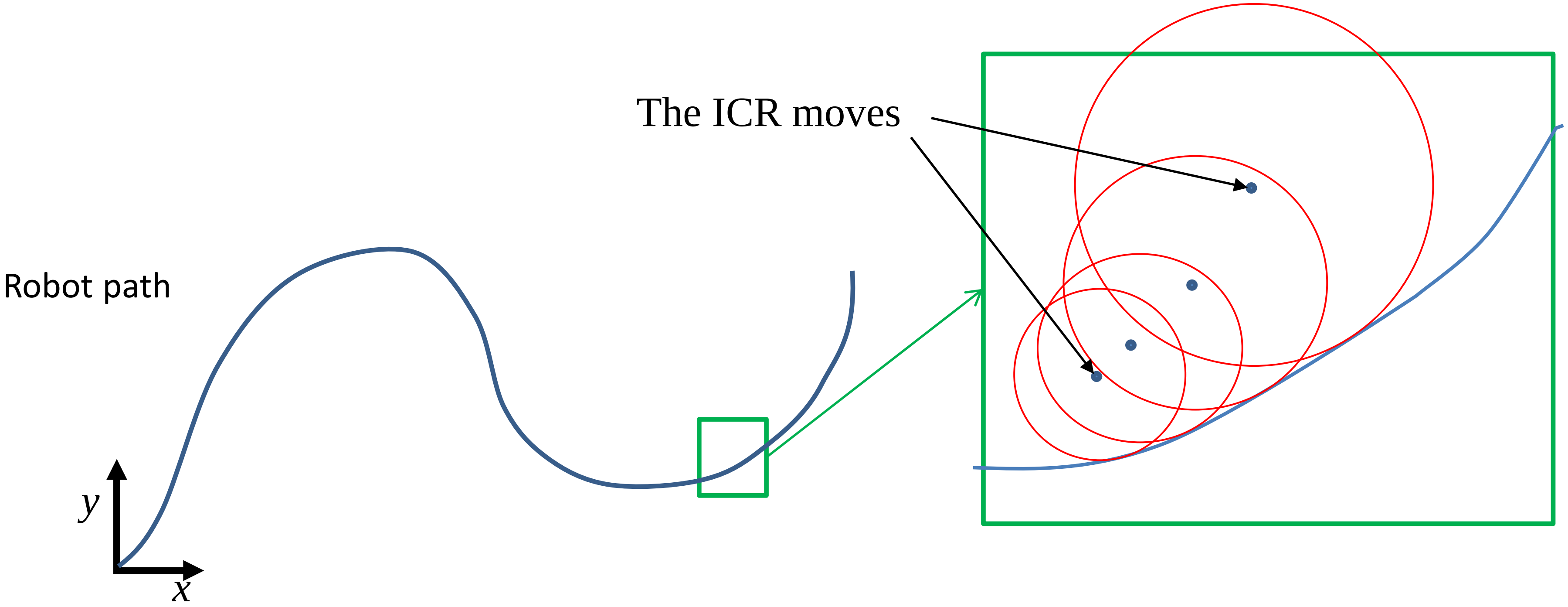


Wheels will be rolling (no slippage): Just one ICT

- For rolling motion to occur (no slippage), all wheels must share the same ICR and the same w: angular velocity wrt (with-respect-to) the ICR
- $v_i = w\, R_i \rightarrow$ same angular velocity w for each wheel, different turning radius R_i

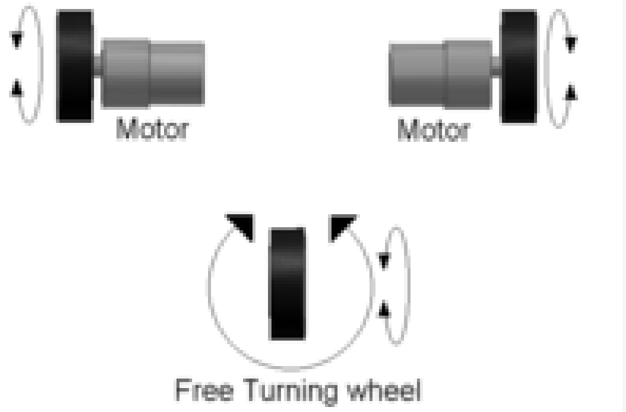
Differential Drive

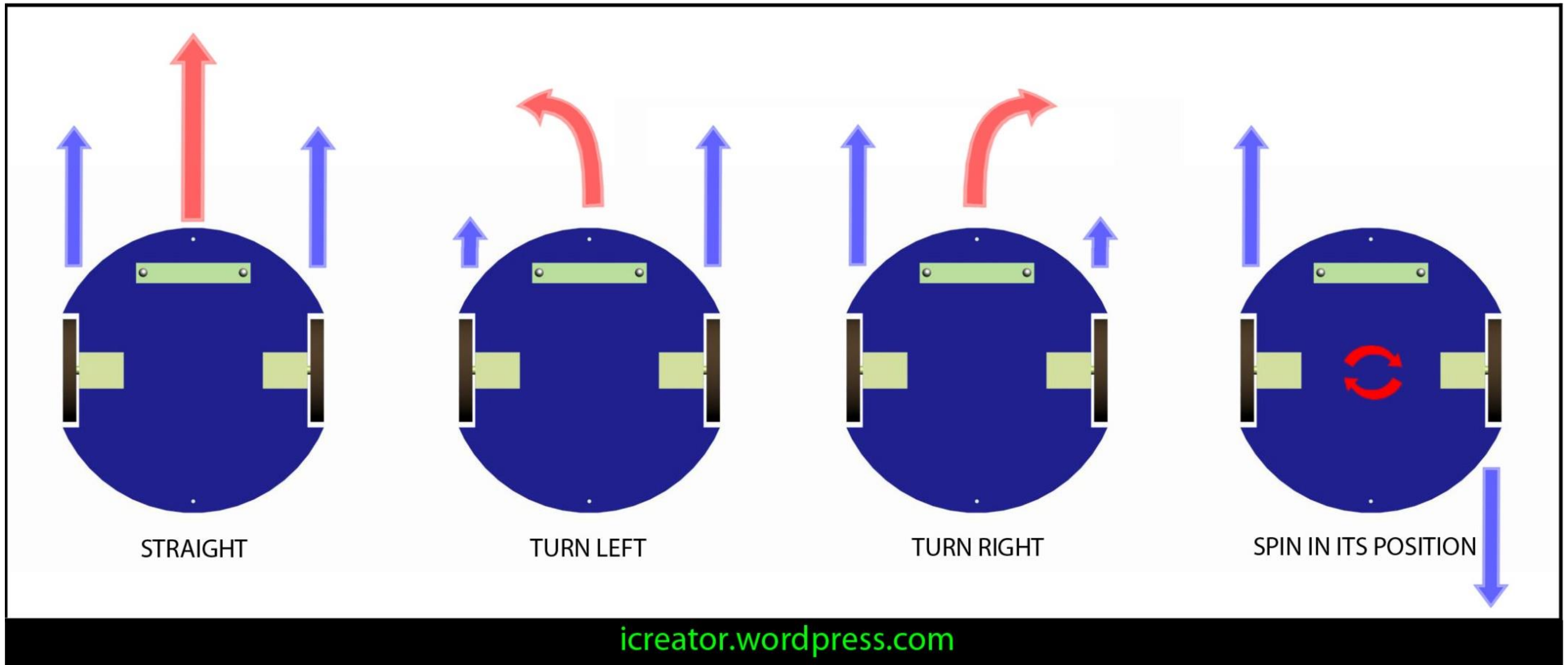
How is it possible that a robot moves along this path if the instantaneous motion is always circular?



The path is the composition of a sequence of circular motions with an IRC that moves

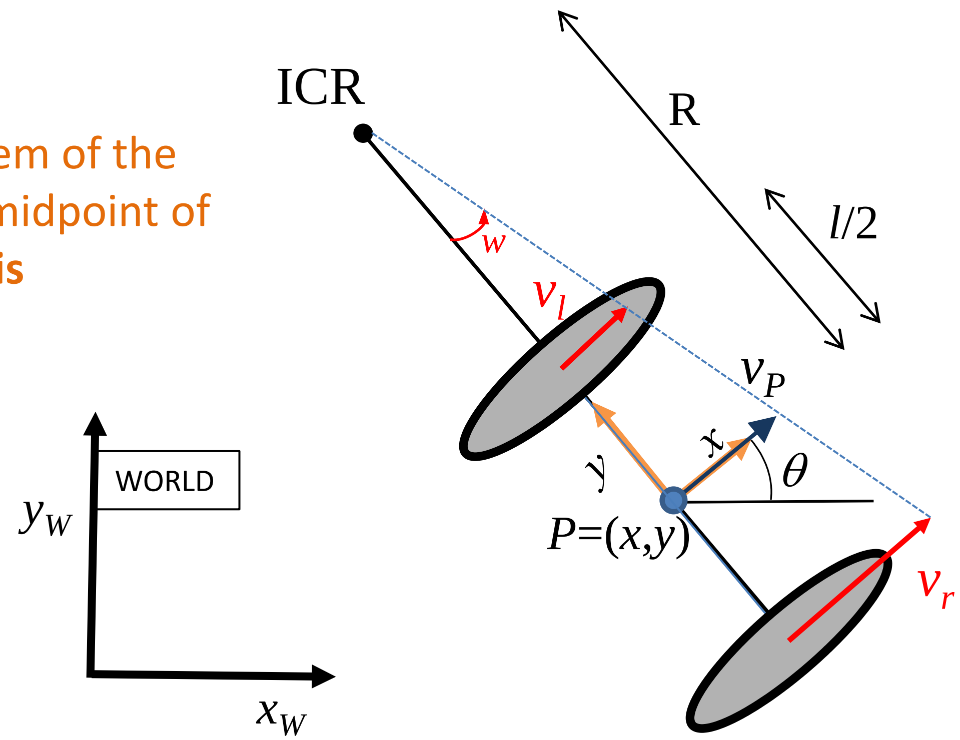
Differential Drive





Differential Drive

Reference system of the vehicle at the midpoint of the **wheels' axis**



Lineal velocity of the **wheels**:

$$v_r = w(R + l/2)$$
$$v_l = w(R - l/2)$$

$$\Rightarrow \quad R = \frac{l}{2}\frac{(v_l + v_r)}{(v_r - v_l)} \qquad w = \frac{v_r - v_l}{l}$$

Velocity of the **robot-axis midpoint**:

$$v_P = w \cdot R$$

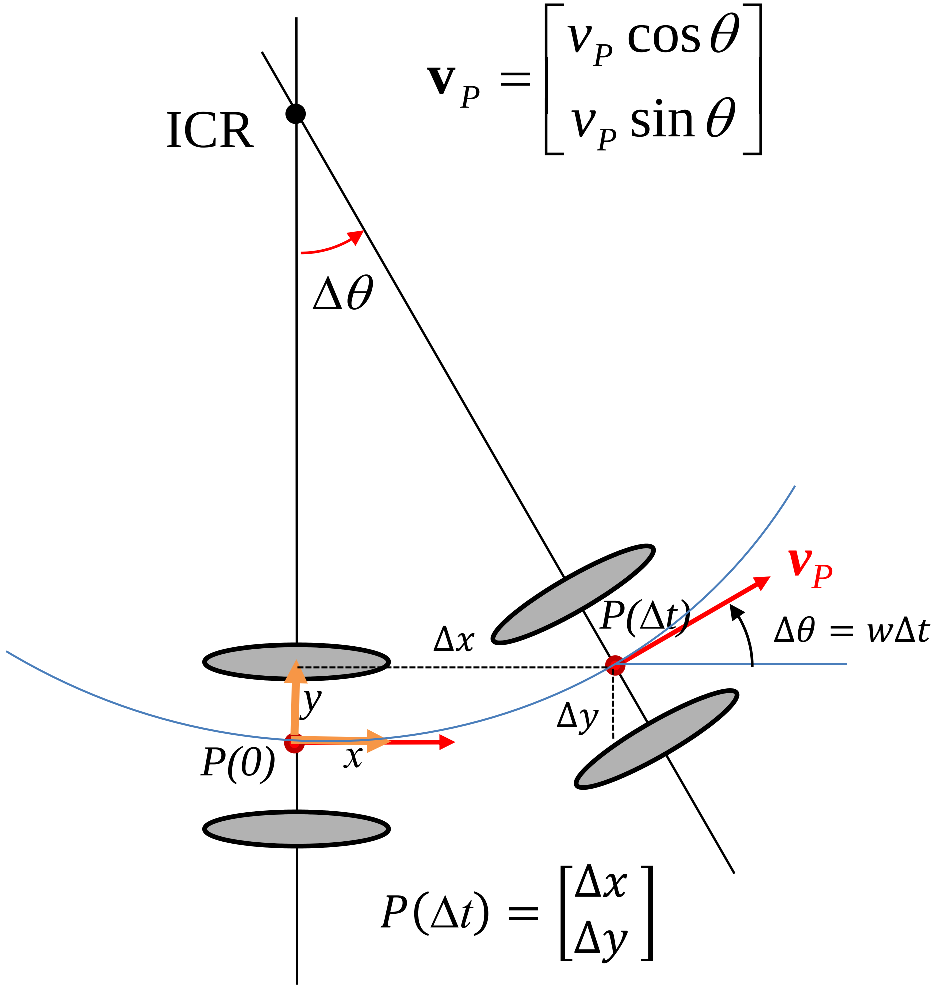
Vehicle velocity is always perpendicular to the robot wheel axis

Typically, the robot motion is given by any of these pairs of variables: $\langle v_l, v_r \rangle, \langle v_P, w \rangle$

$$v_r = w(R + l/2) = v_P + w(l/2)$$
$$v_l = w(R - l/2) = v_P - w(l/2)$$

Differential Drive

Incremental robot pose: How much has the robot moved in Δt



Assuming w y R constant ($v_P = wR$ constant) during the period Δt

$\Downarrow \Delta\theta = w\Delta t$

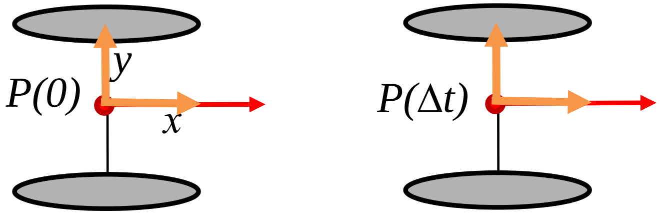
$$\Delta x = x(\Delta t) = \int_0^{\Delta t} v_P(t)\ \cos[\Delta\theta(t)]dt = \frac{v_P}{w}\sin(w\Delta t)$$

$$\Delta y = y(\Delta t) = \int_0^{\Delta t} v_P(t)\ \sin[\Delta\theta(t)]dt = \frac{v_P}{w}[1 - \cos(w\Delta t)]$$

$$\Delta\theta = \theta(\Delta t) = \int_0^{\Delta t} w(t)\ dt = w\Delta t$$

Special case: If the robot moves *straight forward* *(no rotation)*

$w=0 \rightarrow v_r = v_l = v_P$



$$x(\Delta t) = v_P\Delta t$$
$$y(\Delta t) = 0$$
$$\theta(\Delta t) = 0$$

ICR at infinite

Differential Drive

Summary:

$$w = \frac{v_r - v_l}{l}$$

$$v_P = \frac{v_l + v_r}{2}$$

$$u = [v_l, v_r]^T \longrightarrow \boxed{\text{Robot Kinematics}} \xrightarrow{\Delta p_t}$$

Alternatively

$$u = [w, v_P]^T$$

$$\Delta x = x(\Delta t) = \frac{v_P}{w} \sin(w\Delta t)$$

$$\Delta y = y(\Delta t) = \frac{v_P}{w} [1 - \cos(w\Delta t)]$$

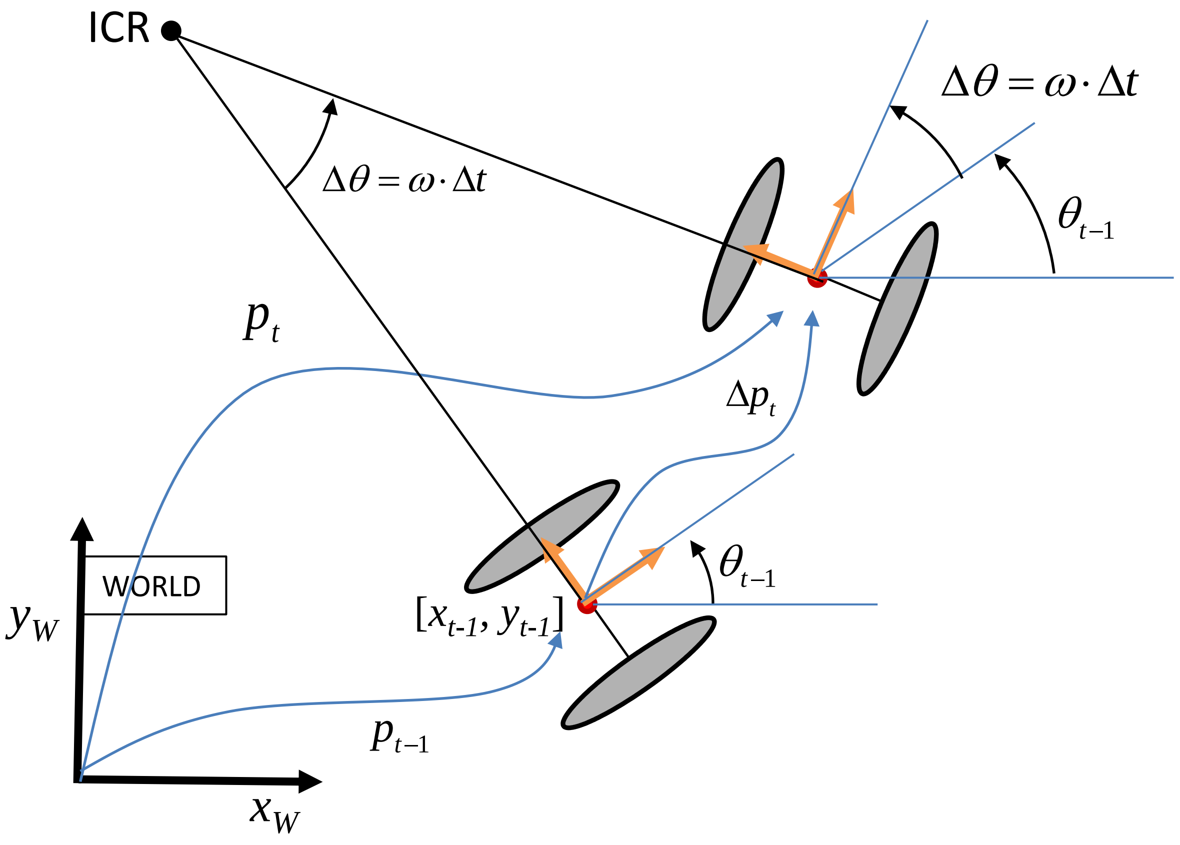
$$\Delta\theta = \theta(\Delta t) = w\Delta t$$

This pose increment is expressed in the first robot coordinate system!

What if we want the robot pose in an arbitrary coordinate system?

Pose of the robot through composition of poses

Robot poses in the World System



pose composition operator

This pose p_{t-1} in the t-1 frame

$$p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = p_{t-1} \oplus \Delta p_t = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} \oplus \begin{bmatrix} \Delta x_t \\ \Delta y_t \\ \Delta\theta_t \end{bmatrix}$$

$$= \begin{bmatrix} x_{t-1} + \Delta x_t \cos\theta_{t-1} - \Delta y_t \sin\theta_{t-1} \\ y_{t-1} + \Delta x_t \sin\theta_{t-1} + \Delta y_t \cos\theta_{t-1} \\ \theta_{t-1} + \Delta\theta_t \end{bmatrix}$$

We'll see next where this expression comes from

Wheel Odometry

Vehicle pose given by composing small incremental movements with *w* and *R* constant

Composition of poses: $p_t = p_{t-1} \oplus \Delta p_t = \begin{bmatrix} x_{t-1} + \Delta x_t \cos\theta_{t-1} - \Delta y_t \sin\theta_{t-1} \\ y_{t-1} + \Delta x_t \sin\theta_{t-1} + \Delta y_t \cos\theta_{t-1} \\ \theta_{t-1} + \Delta\theta_t \end{bmatrix}$

Assuming $w \neq 0$ and R (v_p) constant

$$\Delta x_t = \frac{v_P}{w}\sin(w\Delta t)$$
$$\Delta y_t = \frac{v_P}{w}[1 - \cos(w\Delta t)]$$
$$\Delta\theta_t = w\Delta t$$

For a differential drive vehicle commanded with $u = [v_P, w]^T$

If linear motion (w=0)

$$\Delta x_t = v_P\Delta t$$
$$\Delta y_t = 0$$
$$\Delta\theta_t = 0$$

For clarity, we omit the subscript t in v_p and w

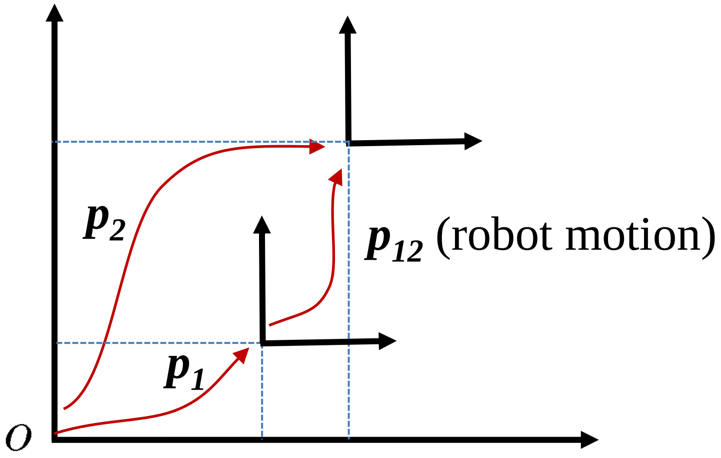
$$p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} -\frac{v_P}{w}\sin\theta_{t-1} + \frac{v_P}{w}\sin(\theta_{t-1} + w\Delta t) \\ \frac{v_P}{w}\cos\theta_{t-1} - \frac{v_P}{w}\cos(\theta_{t-1} + w\Delta t) \\ w\Delta t \end{bmatrix}$$

$$p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + v_P\Delta t \begin{bmatrix} \cos\theta_{t-1} \\ \sin\theta_{t-1} \\ 0 \end{bmatrix}$$

Odometry is computed by the wheel controller at a very high rate in order to guarantee that w and R remain constant

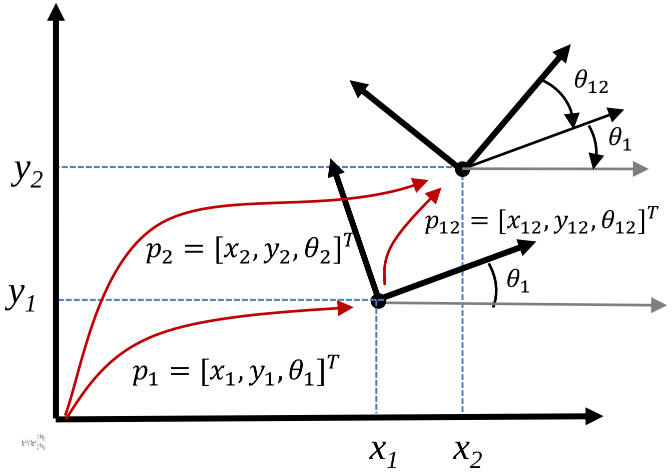
Pose of the robot through composition of poses

Just translation (i.e. the robot is a point, has not orientation)



$$\underbrace{p_2 = p_1 + p_{12}}_{\text{Behave like vectors!}} = \begin{bmatrix} x_1 + x_{12} \\ y_1 + y_{12} \end{bmatrix}$$

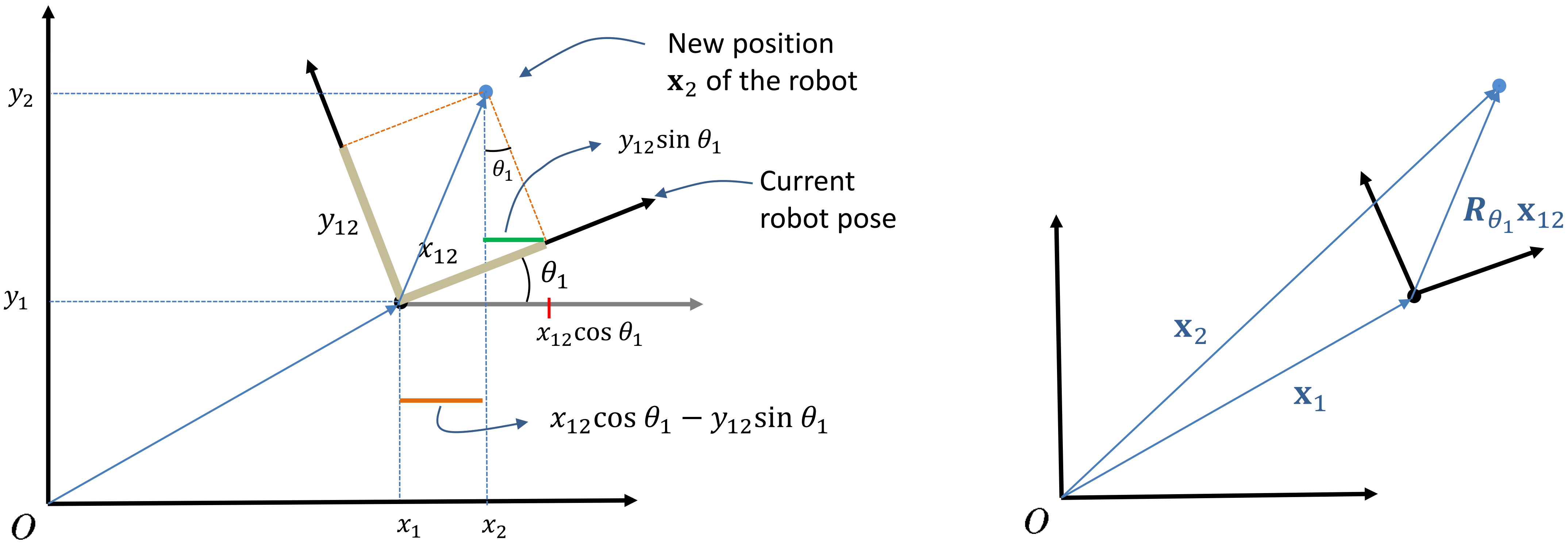
Translation + rotation: $p_2 \neq p_1 + p_{12} \Rightarrow \boxed{p_2 = p_1 \oplus p_{12}}$ **Pose composition**



$$p_2 = \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = p_1 \oplus p_{12} = \begin{bmatrix} x_1 + x_{12} \cos\theta_1 - y_{12} \sin\theta_1 \\ y_1 + x_{12} \sin\theta_1 + y_{12} \cos\theta_1 \\ \theta_1 + \theta_{12} \end{bmatrix}$$

Where does this expression come from?
NEXT

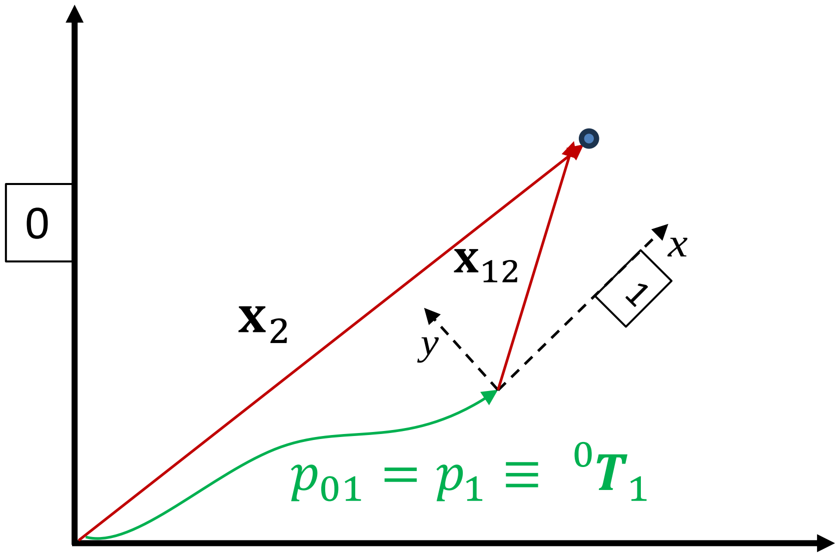
Rigid (Euclidean) transformation between coordinate systems



$$\mathbf{x}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_{12}\cos\theta_1 & -y_{12}\sin\theta_1 \\ x_{12}\sin\theta_1 & y_{12}\cos\theta_1 \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix}\begin{bmatrix} x_{12} \\ y_{12} \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \boldsymbol{R}_{\theta_1}\mathbf{x}_{12} + \boldsymbol{x}_1$$

If no rotation $\boldsymbol{R}_{\theta_1} = \boldsymbol{I}$ $\Rightarrow$ $\mathbf{x}_2 = \mathbf{x}_{12} + \boldsymbol{x}_1$

Pose as a rigid (Euclidean) transformation between coordinate systems



$$\mathbf{x}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \boldsymbol{R}_{\theta_1}\mathbf{x}_{12} + \mathbf{x}_1$$

Using homogeneous coordinates: $\tilde{\mathbf{x}}_2 = \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}$

$$\tilde{\mathbf{x}}_2 = \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\theta_1} & \mathbf{x}_1 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{12} \\ 1 \end{bmatrix} = {}^0\boldsymbol{T}_1 \tilde{\mathbf{x}}_{12}$$

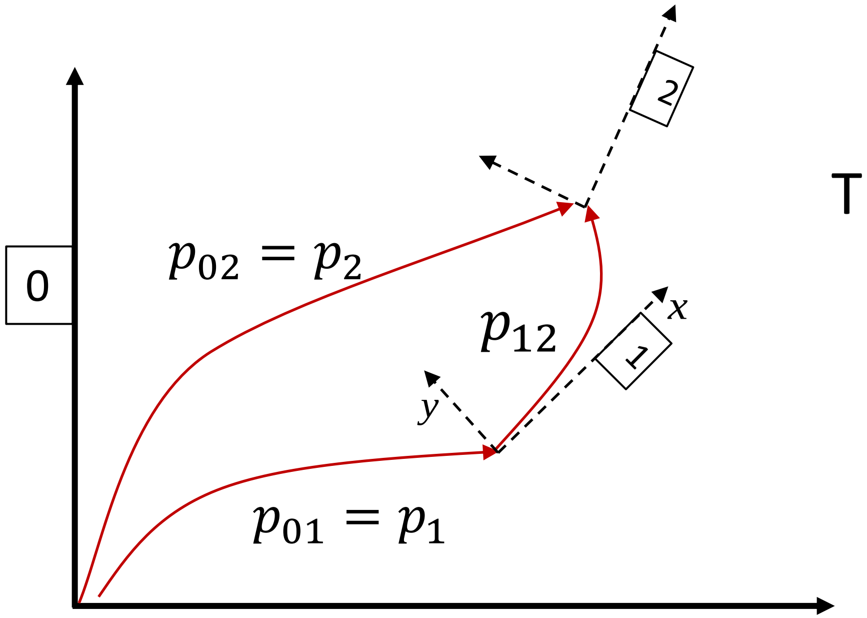
${}^0\boldsymbol{T}_1$ expresses point coordinates given in the **system 1** (e.g. $\mathbf{x}_{12}$) as coordinates of **system 0**

Equivalence between transformation and pose (${}^0\boldsymbol{T}_1 \equiv p_{01}$):

$$p_{01} = p_1 = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \theta_1 \end{bmatrix} \rightarrow {}^0\boldsymbol{T}_1 = \begin{bmatrix} \boldsymbol{R}_{\theta_1} & \mathbf{x}_1 \\ 0^T & 1 \end{bmatrix}$$

Typically, 0 is dropped for short

Pose of the robot through composition of poses



The robot, at pose p_1, moves p_{12} reaching a new pose p_2

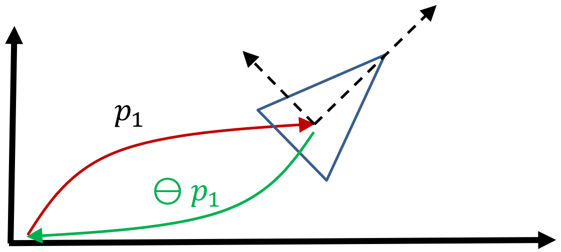
$$p_2 = p_1 \oplus p_{12}$$

$$p_2 = \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = p_1 \oplus p_{12} = \begin{bmatrix} x_1 + x_{12}\cos\theta_1 - y_{12}\sin\theta_1 \\ y_1 + x_{12}\sin\theta_1 + y_{12}\cos\theta_1 \\ \theta_1 + \theta_{12} \end{bmatrix} = \begin{bmatrix} {}^0\boldsymbol{T}_1\tilde{\mathbf{x}}_{12}(1) \\ {}^0\boldsymbol{T}_1\tilde{\mathbf{x}}_{12}(2) \\ \theta_1 + \theta_{12} \end{bmatrix}$$

First two elements of the 3x1 vector ${}^0\boldsymbol{T}_1\tilde{\mathbf{x}}_{12}$

Why is interesting to see poses as transformations?

1. **Inverse of a pose** (inverse of transformation)



$$p_1 \equiv {}^{0}\boldsymbol{T}_1 \Rightarrow \ominus p_1 \equiv {}^{0}\boldsymbol{T}_1^{\;-1} = {}^{1}\boldsymbol{T}_0$$

RECALL: $${}^{1}\boldsymbol{T}_0 = {}^{0}\boldsymbol{T}_1^{\;-1} = \begin{bmatrix} \boldsymbol{R}_{\theta_1} & \mathbf{x}_1 \\ 0^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{R}_{\theta_1}^{-1} & -\boldsymbol{R}_{\theta_1}^{-1}\mathbf{x}_1 \\ 0^T & 1 \end{bmatrix}$$

This transformation takes any point from 0 to 1, e.g. $O_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$:

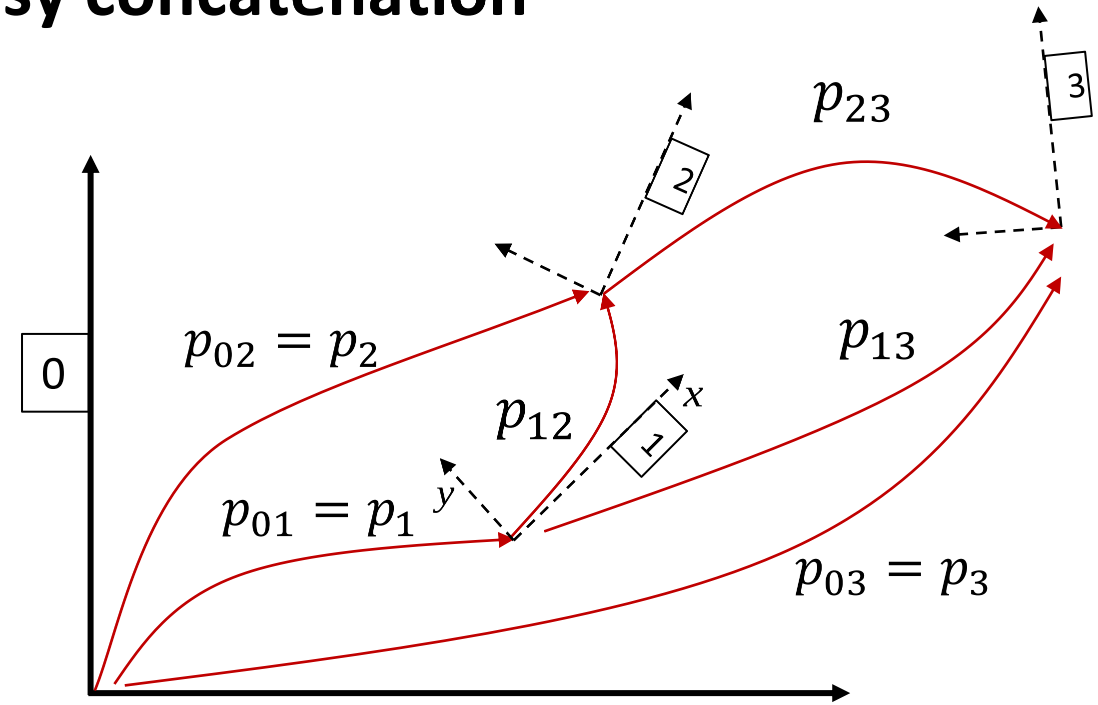
$$\widetilde{O_1} = \boldsymbol{T}_{10}\widetilde{O_0} = \begin{bmatrix} \boldsymbol{R}_{\theta_1}^{-1} & -\boldsymbol{R}_{\theta_1}^{-1}\mathbf{x}_1 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} c\theta_1 & s\theta_1 \\ -s\theta_1 & c\theta_1 \end{bmatrix} & -\begin{bmatrix} x_1 c\theta_1 + y_1 s\theta_1 \\ -x_1 s\theta_1 + y_1 c\theta_1 \end{bmatrix} \\ [0 \quad 0 \quad 0] & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1^- \\ y_1^- \\ 1 \end{bmatrix}$$

Question:

What is the pose of the robot A *wrt* to robot B if we know that pose of B *wrt* to A is $p_{01} = [2\ 3\ 45º]^{\mathrm{T}}$ **Answer:** $p_{10} = \ominus p_1 = \begin{bmatrix} \frac{-5}{2}\sqrt{2} & \frac{-\sqrt{2}}{2} & -45º \end{bmatrix}^T$

Why is interesting to see poses as transformations?

2. Easy concatenation



Sequence of transformation: $p_{02} = p_{01} \oplus p_{12}$ $\longrightarrow$ ${}^{0}\boldsymbol{T}_2 = {}^{0}\boldsymbol{T}_1\,{}^{1}\boldsymbol{T}_2$

Solve for any pose: $\ominus\, p_{12} = p_{23} \ominus p_{13}$ $\longrightarrow$ ${}^{1}\boldsymbol{T}_2^{-1} = {}^{2}\boldsymbol{T}_1 = {}^{2}\boldsymbol{T}_3\,{}^{3}\boldsymbol{T}_1 = {}^{2}\boldsymbol{T}_3\,{}^{1}\boldsymbol{T}_3^{-1}$

$p_2 = p_1 \oplus p_{13} \ominus p_{23}$ $\longrightarrow$ ${}^{0}\boldsymbol{T}_2 = {}^{0}\boldsymbol{T}_1\,{}^{1}\boldsymbol{T}_3\,{}^{3}\boldsymbol{T}_2$

Properties of poses

The set of 2D poses equipped with the composition operator $\oplus$ form an additive group called $SE(2)$ group (Special Euclidean). $SE(3)$ for 3D poses (6 dof)

Closure: $p_A \oplus p_B = p_C$ The composition of poses gives us a pose

Associativity: $(p_A \oplus p_B) \oplus p_C = p_A \oplus (p_B \oplus p_C)$

Identity element: $p_A \oplus 0 = 0 \oplus p_A = p_A$

Inverse: $\ominus p_{AB} = p_{BA}$

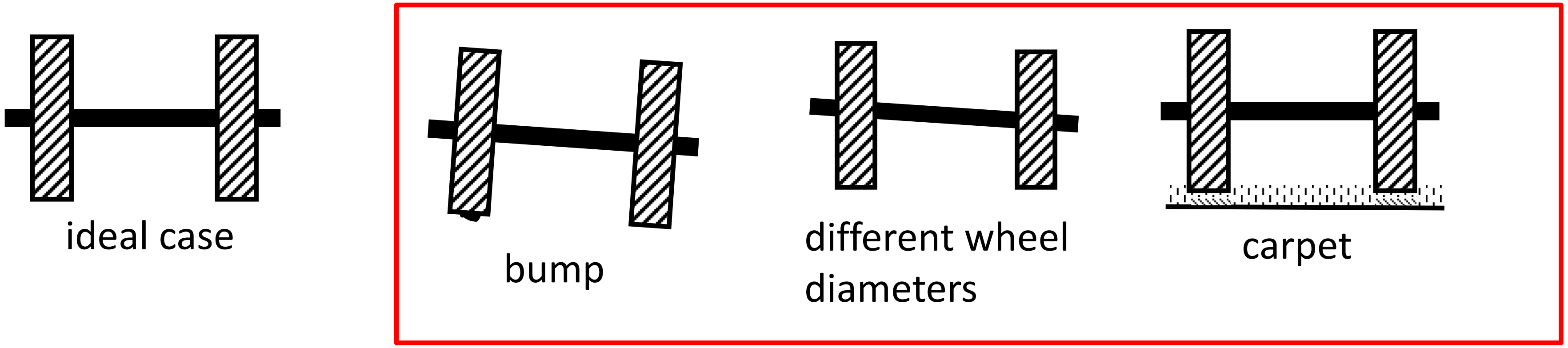
$$(\ominus p_A) \oplus p_A = p_A \oplus (\ominus p_A) = 0$$
$$p_A \ominus p_A = 0, p_A \ominus 0 = p_A$$

Watch out: Not commutative $p_A \oplus p_B \neq p_B \oplus p_A$

Probabilistic motion model

Motion error sources:

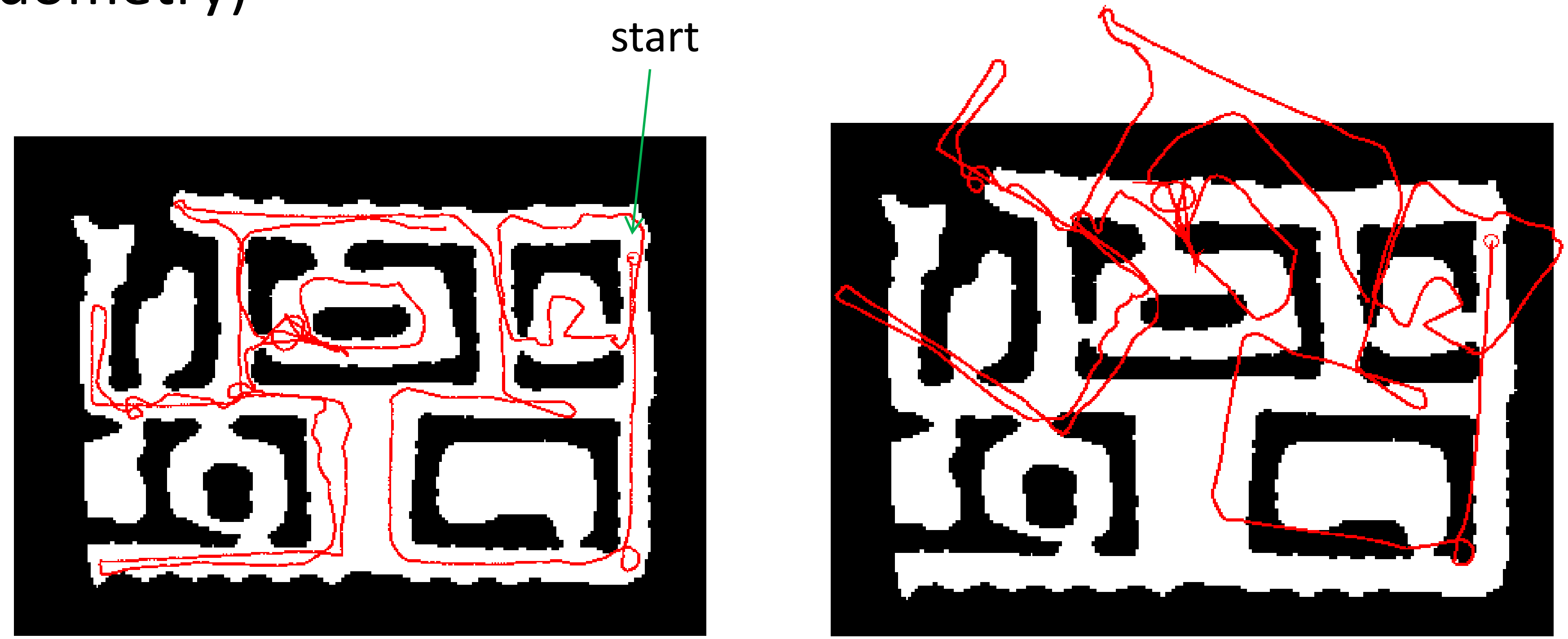
- Wheel slippage
- Inaccurate calibration
- Limited resolution during integration (time increments, measurement resolution)
- Unequal floor



Motion error sources

Probabilistic motion model

Example of robot path from composition of incremental motion (odometry)



Real path

Path reconstructed from odometry

Error produces a drift in the path that accumulates overtime

Probabilistic motion model

We need to characterize the robot motion in probabilistic terms

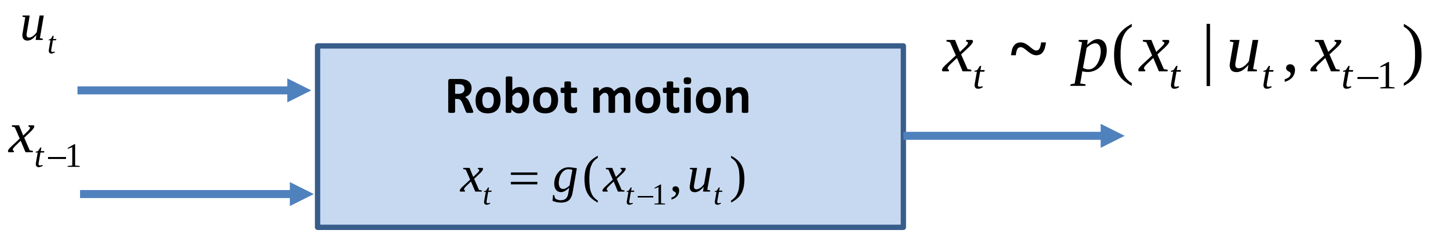
$$p(x_t \mid u_t, x_{t-1})$$

pose at time t

pose at time t-1

motion command at t-1

Distribution over poses when executing the motion command u_t and its pose is x_{t-1}

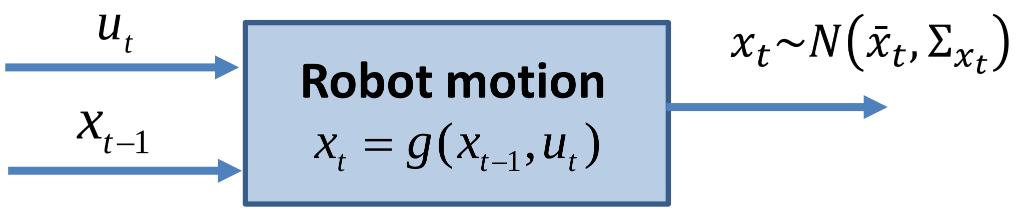


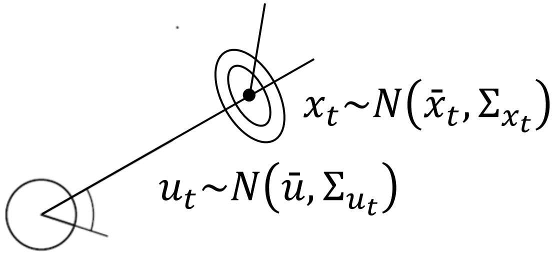
Probabilistic motion model

We may need this model **in two forms**:

- Analytic form, i.e. *pdf* of the x_t distribution: used in **Extended Kalman Filter**

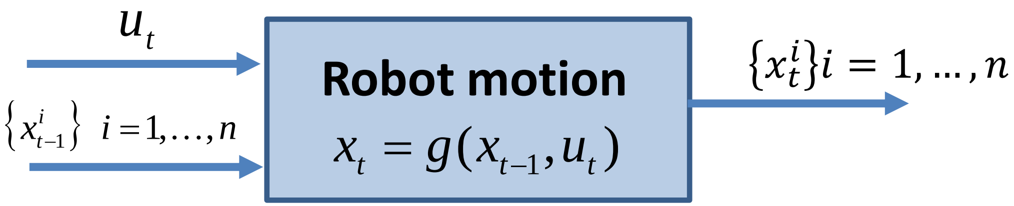
Typically: $u_t \sim N(\bar{u}, \Sigma_{u_t})$ $\quad x_{t-1} \sim N(\bar{x}_{t-1}, \Sigma_{x_{t-1}})$

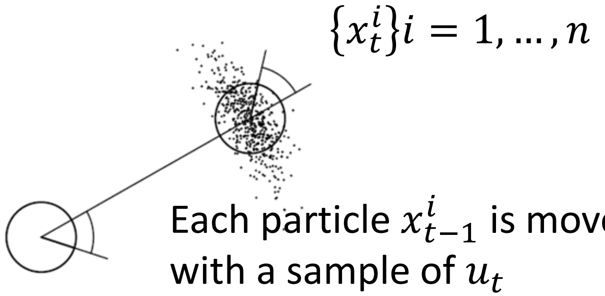




- Sample form: used in **Particle Filter (Sequential Montecarlo)**

Typically: $u_t \sim N(\bar{u}, \Sigma_{u_t})$ $\{x_{t-1}^i\} i = 1, \dots, n$ (samples)





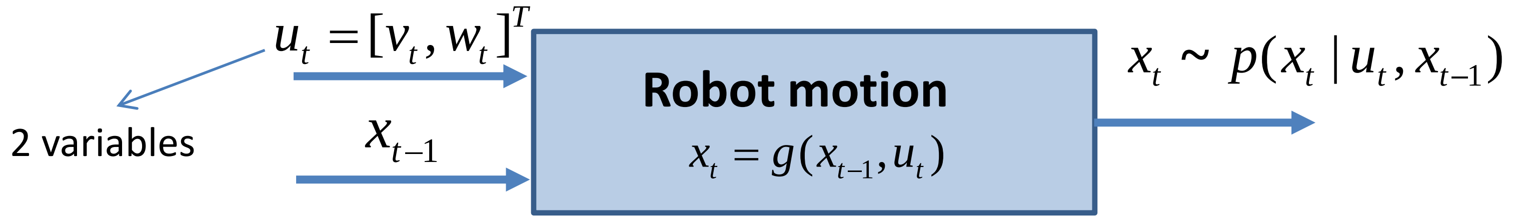
Each particle x_{t-1}^i is moved with a sample of u_t

Probabilistic motion model

In practice, two types of motion models $x_t = g(x_{t-1}, u_t)$:

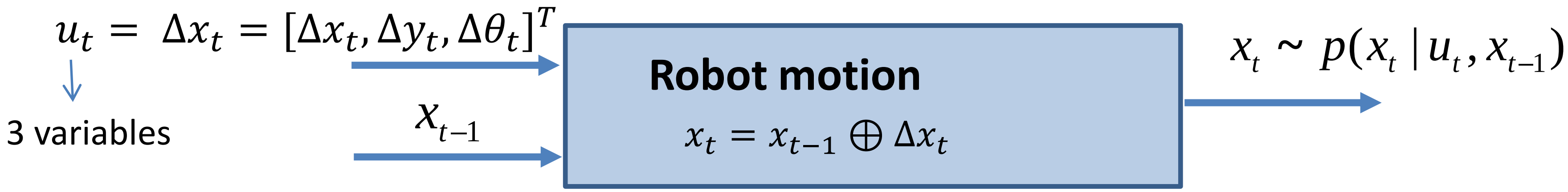
Velocity-based

- robot is controlled through linear and angular velocities <v, w>
- applied when no wheel encoders are given



Odometry-based

- robot is controlled through odometry pose increments $\Delta x_t = [\Delta x_t, \Delta y_t, \Delta\theta_t]^T$
- used when robot is equipped with wheel encoders



Velocity Motion Model

Assume $u_t \sim N(\bar{u}, \Sigma_{u_t})$ and $x_{t-1} \sim N(\bar{x}_{t-1}, \Sigma_{x_{t-1}})$ $\quad \Sigma_{u_t} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix}$

Equivalent to $u_t = [v_l, v_r]^T$ ← $u_t = [v_t, w_t]^T$

x_{t-1}

Robot motion

$x_t = x_{t-1} \oplus \Delta x_t(u_t)$

$x_t \sim p(x_t \mid u_t, x_{t-1}) \approx N(\bar{x}_t, \Sigma_{x_t})$

w and v_p are assumed to be constant

$$\Delta x = \frac{v_P}{w}\sin(w\Delta t)$$
$$\Delta y = \frac{v_P}{w}[1 - \cos(w\Delta t)]$$
$$\Delta\theta = w\Delta t$$

Since $x_t = x_{t-1} \oplus \Delta x_t$ is not linear → x_t is not Gaussian, but can be approximated

$$\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} -\frac{v_t}{w_t}\sin\theta_{t-1} + \frac{v_t}{w_t}\sin(\theta_{t-1} + w_t\Delta t) \\ \frac{v_t}{w_t}\cos\theta_{t-1} - \frac{v_t}{w_t}\cos(\theta_{t-1} + w_t\Delta t) \\ w_t\Delta t \end{bmatrix}}_{x_{t-1} \oplus \Delta x_t}$$

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + v_P\Delta t \begin{bmatrix} \cos\theta_{t-1} \\ \sin\theta_{t-1} \\ 0 \end{bmatrix}$$

If $w=0$ (linear motion)

Needs to be computed at a high rate in order to guarantee w and v constant

Velocity Motion Model

[See equations of Jacobians in appendix]

$u_t = [v_t, w_t]^T \rightarrow$ **Robot motion** $\leftarrow x_{t-1}$

$$x_t = g(x_{t-1}, u_t) = x_{t-1} \oplus \Delta x_t(u_t)$$

$$x_t \sim p(x_t | u_t, x_{t-1}) \approx N(\bar{x}_t, \Sigma_{x_t})$$

RECALL:

if $Z=f(X,Y)$

$$\Sigma_Z = \frac{\partial f}{\partial X} \Sigma_X \left(\frac{\partial f}{\partial X}\right)^T + \frac{\partial f}{\partial Y} \Sigma_Y \left(\frac{\partial f}{\partial Y}\right)^T$$

Mean: $\bar{x}_t = g(\bar{x}_{t-1}, \bar{u}_t) = \bar{x}_{t-1} \oplus \overline{\Delta x_t}$

Covariance:

$$\Sigma_{x_t} \approx \frac{\partial g}{\partial x_{t-1}} \Sigma_{x_{t-1}} \frac{\partial g}{\partial x_{t-1}}^T + \frac{\partial g}{\partial u_t} \Sigma_{u_t} \frac{\partial g}{\partial u_t}^T = \frac{\partial g}{\partial x_{t-1}} \Sigma_{x_{t-1}} \frac{\partial g}{\partial x_{t-1}}^T + \frac{\partial g}{\partial \Delta x_t} \boxed{\Sigma_{\Delta x_t}} \frac{\partial g}{\partial \Delta x_t}^T$$

evaluated at $\bar{x}_{t-1}$ evaluated at $\bar{u}_t$

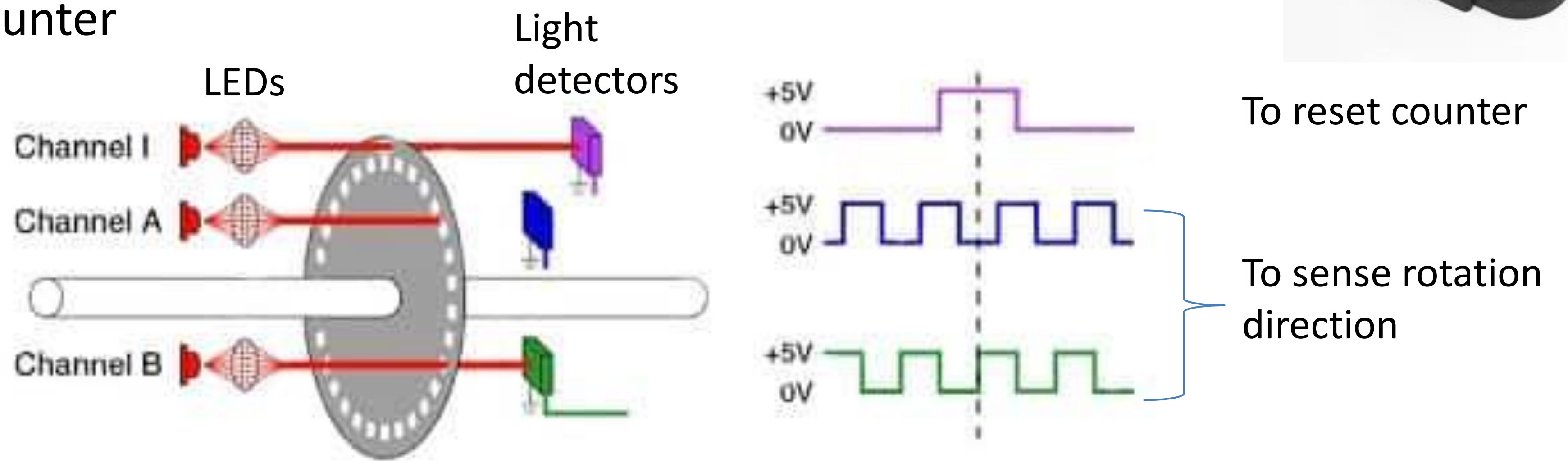
$$\frac{\partial g}{\partial \Delta x_t} \boxed{\frac{\partial \Delta x_t}{\partial u_t} \Sigma_{u_t} \frac{\partial \Delta x_t}{\partial u_t}^T} \frac{\partial g}{\partial \Delta x_t}^T$$

Equivalently:

$$\Sigma_{x_t} \approx \frac{dg}{d\{x_{t-1}, u_t\}} \begin{bmatrix} \Sigma_{x_{t-1}} & 0_{3x2} \\ 0_{2x3} & \Sigma_{u_t} \end{bmatrix} \frac{dg}{d\{x_{t-1}, u_t\}}^T = \begin{bmatrix} \frac{\partial g}{\partial x_{t-1}} & \frac{\partial g}{\partial u_t} \end{bmatrix} \begin{bmatrix} \Sigma_{x_{t-1}} & 0_{3x2} \\ 0_{2x3} & \Sigma_{u_t} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial x_{t-1}} \\ \frac{\partial g}{\partial u_t} \end{bmatrix}$$

Odometry Motion Model

- Technically, it is a measurement rather than a control, but usually treated as control to simplify the modeling
- Odometry: sums wheel encoder pulses to compute robot pose
  - Pulses are seen by light detector diodes
  - Several channels are used to sense rotation direction and to reset counter

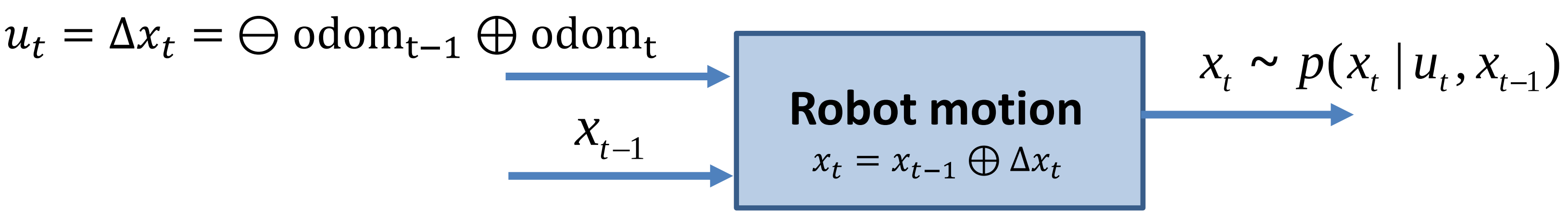


Encoders require +5V and GND to power them, and provide a 0 to 5V output:

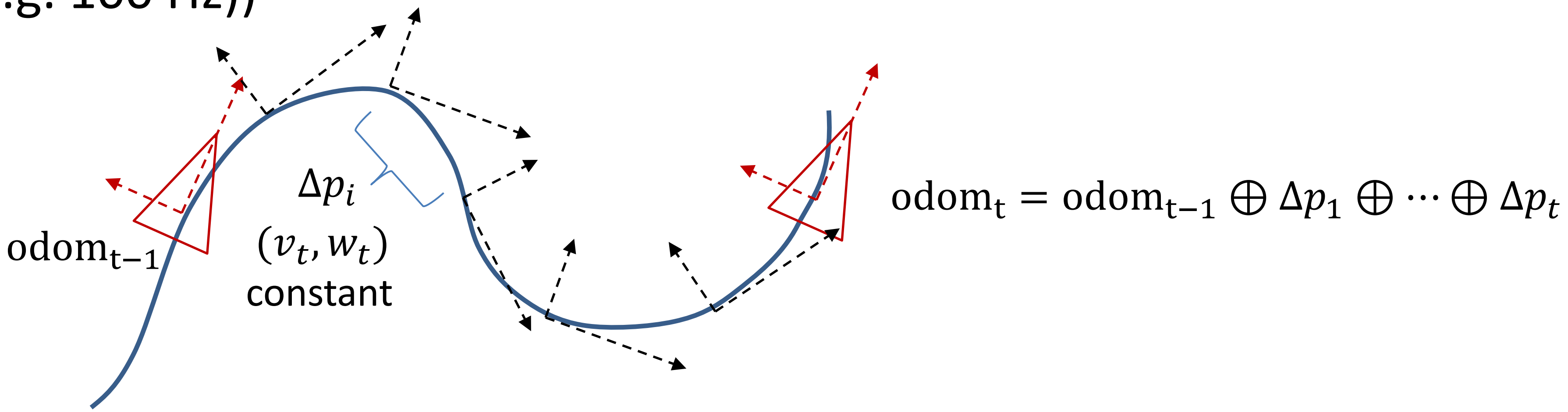
5V when they "see" white, and

0V output when they "see" black.

Odometry Motion Model



The wheel odometry is implemented by the firmware of the robotic platform by sequentially composing increment of poses Δp_i with $(\boldsymbol{v_t}, \boldsymbol{w_t})$ **constant** (then, at a very high rate (e.g. 100 Hz))



The odometry pose $\mathrm{odom}_t = \hat{p}_t = [\widehat{xt}\ \ , \hat{y}_t, \hat{\theta}_t]$ is published to the robot at lower rate (e.g. 10 Hz)

Odometry Motion Model

Analytic form:

$$x_t = g(x_{t-1}, u_t = \Delta x_t) = x_{t-1} \oplus \Delta x_t \sim N\left(\bar{x}_t, \Sigma_{x_t}\right)$$

Since $g(x_{t-1}, \Delta x_t))$ is not linear → x_t is not Gaussian, but can be approximated

Mean: $\bar{x}_t = g(\bar{x}_{t-1}, \Delta\bar{x}_t) = \bar{x}_{t-1} \oplus \Delta\bar{x}_t$

Covariance: $\Sigma_{x_t} \approx \frac{\partial g}{\partial x_{t-1}} \Sigma_{x_{t-1}} \frac{\partial g}{\partial x_{t-1}}^T + \frac{\partial g}{\partial \Delta x_t} \Sigma_{\Delta x_t} \frac{\partial g}{\partial \Delta x_t}^T$

Covariance at x_{t-1}

No correlation assumed

$$\Sigma_{\Delta x_t} = \begin{bmatrix} \sigma_{\Delta x}^2 & 0 & 0 \\ 0 & \sigma_{\Delta y}^2 & 0 \\ 0 & 0 & \sigma_{\Delta\theta}^2 \end{bmatrix}$$

Jacobians:

$$\frac{\partial g}{\partial x_{t-1}} = \begin{bmatrix} 1 & 0 & -\Delta x_k \sin\theta_{k-1} - \Delta y_k \cos\theta_{k-1} \\ 0 & 1 & \Delta x_k \cos\theta_{k-1} - \Delta y_k \sin\theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial g}{\partial \Delta x_t} = \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

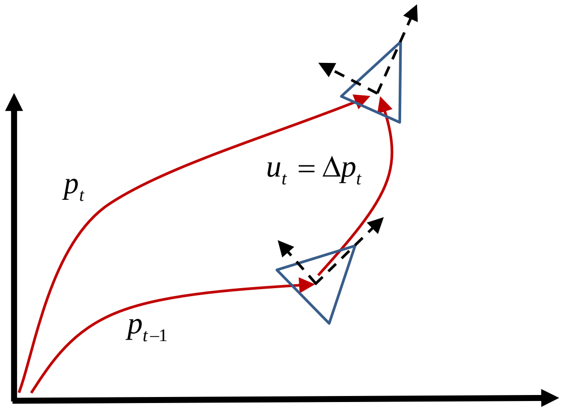
Odometry Motion Model

Analytic form:

We have the **odometry poses:**

$$\hat{p}_t = [\hat{x}_t, \hat{y}_t, \hat{\theta}_t]$$

$$\hat{p}_{t-1} = [\hat{x}_{t-1}, \hat{y}_{t-1}, \hat{\theta}_{t-1}]$$



Do not memorize, this will be used in practical sessions

Mean: $$\Delta\bar{x}_t = \ominus\,\hat{p}_{t-1} \oplus \hat{p}_t = \hat{p}_t \ominus \hat{p}_{t-1} = \begin{bmatrix} (\hat{x}_t - \hat{x}_{t-1})\cos\hat{\theta}_{t-1} + (\hat{y}_t - \hat{y}_{t-1})\sin\hat{\theta}_{t-1} \\ -(\hat{x}_t - \hat{x}_{t-1})\sin\hat{\theta}_{t-1} + (\hat{y}_t - \hat{y}_{t-1})\cos\hat{\theta}_{t-1} \\ \hat{\theta}_t - \hat{\theta}_{t-1} \end{bmatrix}$$

Covariance:

$$\Sigma_{u_t} = \begin{bmatrix} \sigma^2_{\Delta x} & 0 & 0 \\ 0 & \sigma^2_{\Delta y} & 0 \\ 0 & 0 & \sigma^2_{\Delta\theta} \end{bmatrix}$$

Assuming variance grows with the traversed distance $\sqrt{\Delta x^2 + \Delta y^2}$ and the increment in rotation $|\Delta\theta|$

$$\sigma^2_{\Delta x} = \zeta^2_{\Delta x} + \alpha_1\sqrt{\Delta x_t^2 + \Delta y_t^2} + \alpha_2|\Delta\theta_t|$$

$$\sigma^2_{\Delta y} = \sigma^2_{\Delta x}$$

$$\sigma^2_{\Delta\theta} = \zeta^2_{\Delta\theta} + \alpha_3\sqrt{\Delta x_t^2 + \Delta y_t^2} + \alpha_4|\Delta\theta_t|$$

constants

Odometry Motion Model

Sample form:

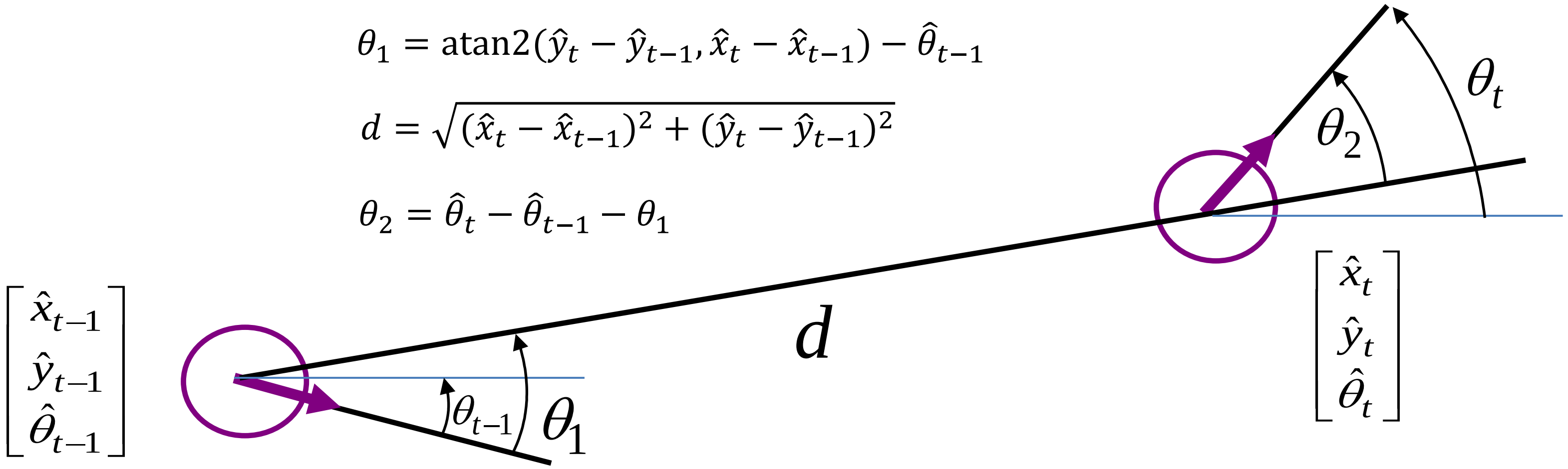
Again, we have the odometry poses $[\hat{x}_t, \hat{y}_t, \hat{\theta}_t]$ and $[\hat{x}_{t-1}, \hat{y}_{t-1}, \hat{\theta}_{t-1}]$, and compute the motion $u_t = [\theta_1, d, \theta_2]^T$ though:

$$\theta_1 = \text{atan2}(\hat{y}_t - \hat{y}_{t-1}, \hat{x}_t - \hat{x}_{t-1}) - \hat{\theta}_{t-1}$$

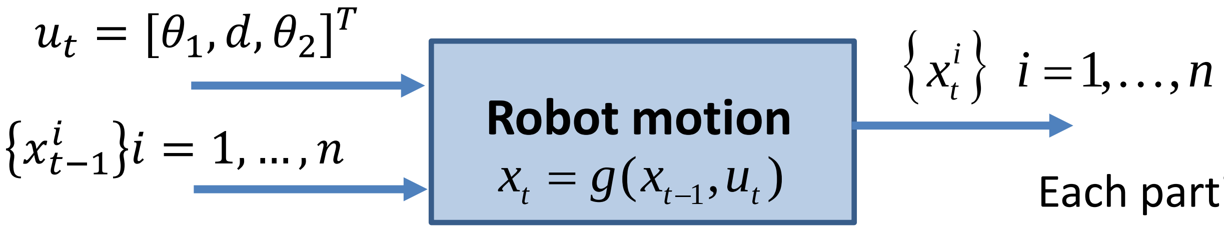
$$d = \sqrt{(\hat{x}_t - \hat{x}_{t-1})^2 + (\hat{y}_t - \hat{y}_{t-1})^2}$$

$$\theta_2 = \hat{\theta}_t - \hat{\theta}_{t-1} - \theta_1$$

Do not memorize, this will be used in practical sessions



The robot rotates θ_1, then moves straight *d*, and then rotates θ_2



Each particle x_{t-1}^i is moved with ramdon $u_t = [\theta_1, d, \theta_2]^T$

Odometry Motion Model

Algorithm **sample_motion_model**(u, x_{t-1}):

$$u = \left(\hat{\theta}_1, \hat{d}, \hat{\theta}_1\right), \quad x_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$$

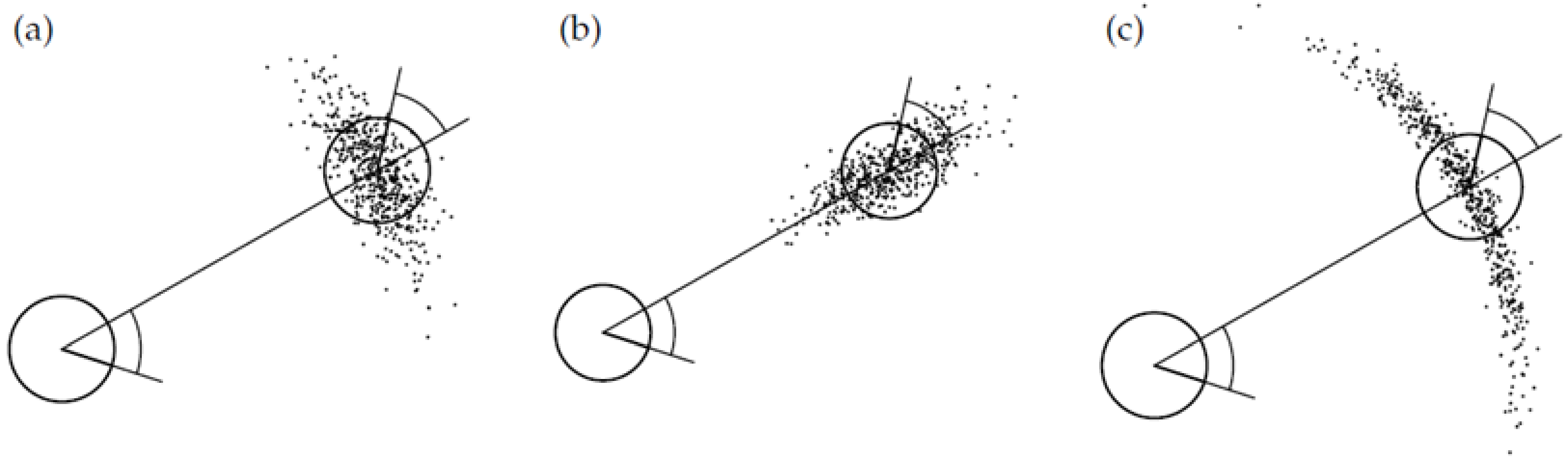
1. $\theta_1 = \hat{\theta}_1 + \text{sample}(\alpha_1\, \hat{\theta}_1^2 + \alpha_2\, \hat{d}^{\,2})$
2. $d = \hat{d} + \text{sample}(\alpha_3\, \hat{d}^{\,2} + \alpha_4\, (\hat{\theta}_1^2 + \hat{\theta}_2^2))$
3. $\theta_2 = \hat{\theta}_2 + \text{sample}(\alpha_1\, \hat{\theta}_2^2 + \alpha_2\, \hat{d}^{\,2})$

$\hat{u}$

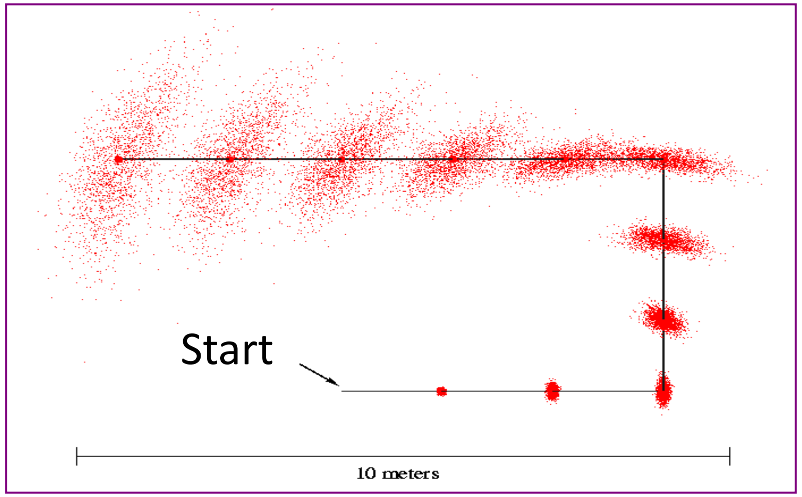
4. $x_t = x_{t-1} + d\cos(\theta_{t-1} + \theta_1)$
5. $y_t = y_{t-1} + d\sin(\theta_{t-1} + \theta_1)$
6. $\theta_t = \theta_{t-1} + \theta_1 + \theta_2$
7. Return (x_t, y_t, θ_t)

sample($\boldsymbol{b}$) draws a gaussian random sample from $N(0, \sigma^2 = \boldsymbol{b})$

Odometry Motion Model



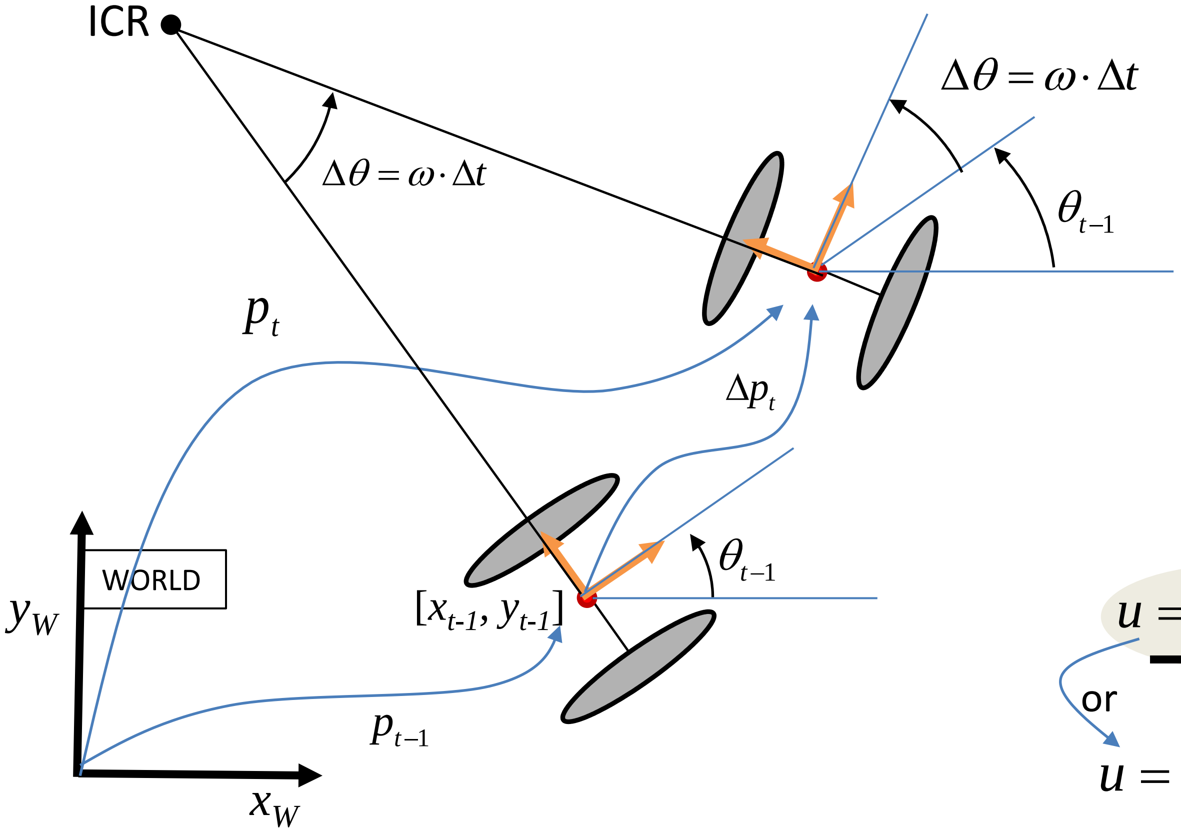
Running the above algorithm with different set of parameters α



Concatenating a sequence on motions

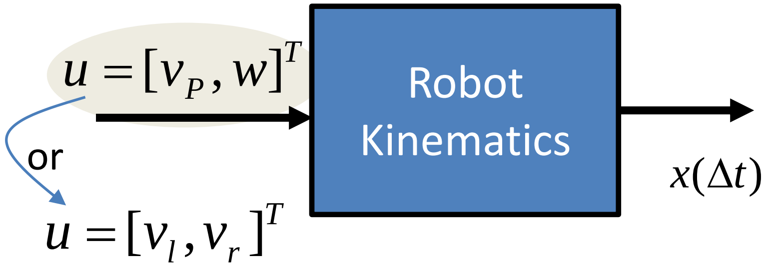
Summary

Vehicle kinematics (deterministic perspective): $x_t = g(x_{t-1}, u_t)$



pose composition

$$p_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \Delta x_t \cos\theta_{t-1} - \Delta y_t \sin\theta_{t-1} \\ y_{t-1} + \Delta x_t \sin\theta_{t-1} + \Delta y_t \cos\theta_{t-1} \\ \theta_{t-1} + \Delta\theta_t \end{bmatrix} = p_{t-1} \oplus \Delta p_t$$



$$x(\Delta t) = \frac{v_P}{w} \sin(w\Delta t)$$

$$y(\Delta t) = \frac{v_P}{w} [1 - \cos(w\Delta t)]$$

$$\theta(\Delta t) = w\Delta t$$

$$x_t = g(x_{t-1}, u_t) = x_{t-1} \oplus u_t$$

Summary

Probabilistic perspective (includes the vehicle kinematics):

$$p(x_t \mid u_t, x_{t-1})$$

pose at time *t* ← x_t

u_t → motion command at *t-1*

x_{t-1} → pose at time *t-1*

u_t

$x_{t-1} \sim N\left(\bar{x}_{t-1}, \Sigma_{x_{t-1}}\right)$

$or\ \{x_{t-1}^i\} i = 1, \dots, n$

Robot motion

$x_t = x_{t-1} \oplus u_t$

Analytic: $x_t = x_{t-1} \oplus \Delta x_t \sim N\left(\bar{x}_t, \Sigma_{x_t}\right)$

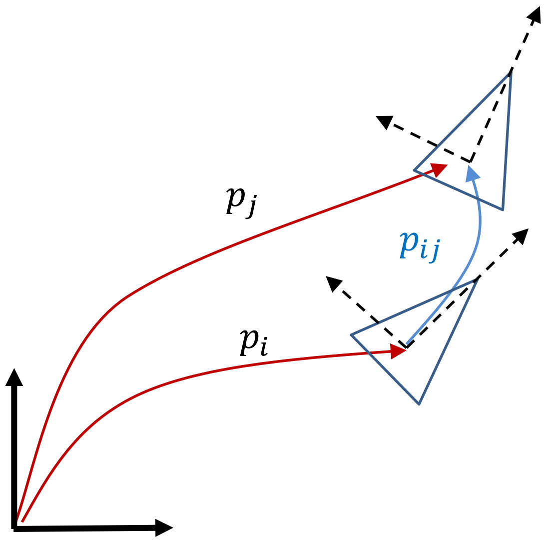
Samples: $\{x_t^i\} i = 1, \dots, n$

x_{t-1}^i is moved with ramdon $u_t = [\theta_1, d, \theta_2]^T$

	Velocity-based	Odometry-based
Analytic form $x_t \sim N\left(\bar{x}_t, \Sigma_{x_t}\right)$	$u_t = [v_t, w_t]^T$	$u_t = [\Delta x_t, \Delta y_t, \Delta\theta_t]^T$
Sample form $\{x_t^i\}\ i = 1, \dots, n$	Not used	$u_t = [\theta_1, d, \theta_2]^T$

Appendix

Composition of two poses:



$$p_j = p_i \oplus p_{ij} = f(p_i, p_{ij}) = \begin{bmatrix} x_i + x_{ij}\cos\theta_i - y_{ij}\sin\theta_i \\ y_i + x_{ij}\sin\theta_i + y_{ij}\cos\theta_i \\ \theta_i + \theta_{ij} \end{bmatrix}$$

Jacobians:

Derivatives evaluated at p_i and p_{ij}!!

$$\frac{\partial p_j}{\partial p_i} = \frac{\partial f(p_i, p_{ij})}{\partial\{x_i, y_i, \theta_i\}} = \begin{bmatrix} 1 & 0 & -x_{ij}\sin\theta_i - y_{ij}\cos\theta_i \\ 0 & 1 & x_{ij}\cos\theta_i - y_{ij}\sin\theta_i \\ 0 & 0 & 1 \end{bmatrix}$$

Jacob1

$$\frac{\partial p_j}{\partial p_{ij}} = \frac{\partial f(p_i, p_{ij})}{\partial\{x_{ij}, y_{ij}, \theta_{ij}\}} = \begin{bmatrix} \cos\theta_j & -\sin\theta & 0 \\ \sin\theta_j & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Jacob2

```
function Jacob1  = J1(x1,x2)
%x1=pi; x2=pij
s1 = sin(x1(3)); c1 = cos(x1(3));
Jacob1  = [1 0 -x2(1)*s1-x2(2)*c1;
           0 1 x2(1)*c1-x2(2)*s1;
           0 0 1];
```

```
function Jacob2  = J2(x1,x2)
%x1=pi; x2=pij
s1 = sin(x1(3));c1 = cos(x1(3));
Jacob2  = [c1 -s1 0;
           s1 c1 0;
           0 0 1];
```

Appendix

Jacobians of the Velocity Motion Model:

$w \neq 0$

$$\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} x_{t-1} - \frac{v_t}{w_t}\sin\theta_{t-1} + \frac{v_t}{w_t}\sin(\theta_{t-1} + w_t\Delta t) \\ y_{t-1} + \frac{v_t}{w_t}\cos\theta_{t-1} - \frac{v_t}{w_t}\cos(\theta_{t-1} + w_t\Delta t) \\ \theta_{t-1} + w_t\Delta t \end{bmatrix}}_{g(x_{t-1}, u_t)} = \begin{bmatrix} x_{t-1} - R\sin\theta_{t-1} + R\sin(\theta_{t-1} + \Delta\theta_t) \\ y_{t-1} + R\cos\theta_{t-1} - R\cos(\theta_{t-1} + \Delta\theta_t) \\ \theta_{t-1} + \Delta\theta_t \end{bmatrix}$$

$$\boxed{R = \frac{v_t}{w_t} \; ; \; \Delta\theta_t = w_t\Delta t}$$

$$\frac{\partial g}{\partial x_{t-1}} = \begin{bmatrix} \frac{\partial x_t}{\partial x_{t-1}} & \frac{\partial y_t}{\partial y_{t-1}} & \frac{\partial \theta_t}{\partial \theta_{t-1}} \\ \frac{\partial y_t}{\partial x_{t-1}} & \frac{\partial y_t}{\partial y_{t-1}} & \frac{\partial \theta_t}{\partial \theta_{t-1}} \\ \frac{\partial \theta_t}{\partial x_{t-1}} & \frac{\partial \theta_t}{\partial y_{t-1}} & \frac{\partial \theta_t}{\partial \theta_{t-1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & R(-s_{t-1}s_\Delta - c_{t-1}(1 - c_\Delta)) \\ 0 & 1 & R(c_{t-1}s_\Delta - s_{t-1}(1 - c_\Delta)) \\ 0 & 0 & 1 \end{bmatrix}$$

$$c_\Delta = \cos\Delta\theta_t$$
$$s_\Delta = \sin\Delta\theta_t$$
$$c_{t-1} = \cos\theta_{t-1}$$
$$s_{t-1} = \sin\theta_{t-1}$$

$$\frac{\partial g}{\partial u_t} = \frac{\partial g}{\partial\{R, \Delta\theta_t\}} \frac{\partial\{R, \Delta\theta_t\}}{\partial\{v_t, w_t\}} = \begin{bmatrix} c_{t-1}s_\Delta - s_{t-1}(1 - c_\Delta) & R(c_{t-1}c_\Delta - s_{t-1}s_\Delta) \\ s_{t-1}s_\Delta + c_{t-1}(1 - c_\Delta) & R(s_{t-1}c_\Delta - c_{t-1}s_\Delta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{w} & -\frac{v}{w^2} \\ 0 & \Delta t \end{bmatrix}$$

$u_t = [v_t, w_t]^T$

Appendix

Jacobians of the Velocity Motion Model:

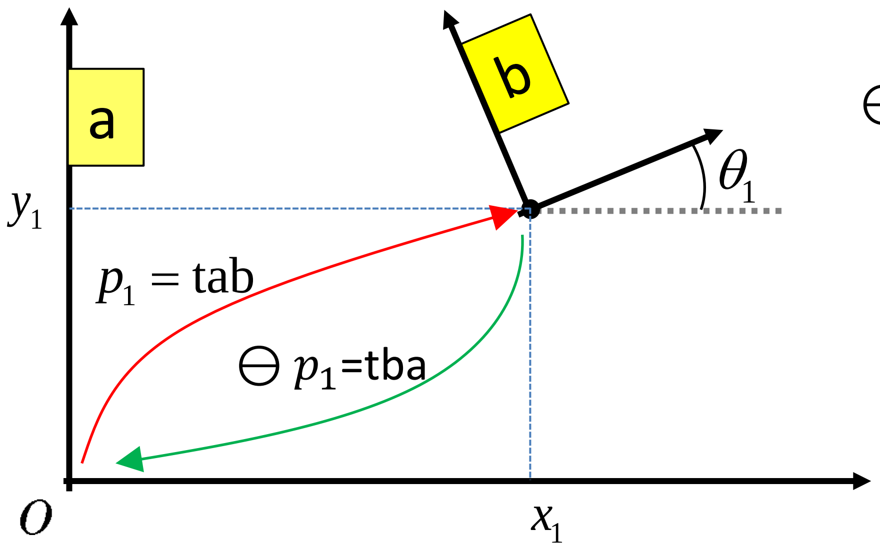
$$w = 0$$

$$\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\boldsymbol{x}_t} = \underbrace{\begin{bmatrix} x_{t-1} + v_t \Delta t \cos\theta_{t-1} \\ y_{t-1} + v_t \Delta t \sin\theta_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{g(x_{t-1}, u_t)}$$

$$\frac{\partial g}{\partial x_{t-1}} = \begin{bmatrix} 1 & 0 & -v_t \Delta t \sin\theta_{t-1} \\ 0 & 1 & v_t \Delta t \cos\theta_{t-1} \\ 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial g}{\partial u_k} = \begin{bmatrix} \frac{\partial x_t}{\partial v_t} & \frac{\partial x_t}{\partial w_t} \\ \frac{\partial y_t}{\partial v_t} & \frac{\partial y_t}{\partial w_t} \\ \frac{\partial \theta_t}{\partial v_t} & \frac{\partial \theta_t}{\partial w_t} \end{bmatrix} = \begin{bmatrix} \Delta t \cos\theta_{t-1} & 0 \\ \Delta t \sin\theta_{t-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Appendix

Inverse of a pose



$$p_1 = [x_1 \;\; y_1 \;\; \theta_1]^T$$

$$\ominus p_1 = f(p_1) = \begin{bmatrix} -x_1 \cos\theta_1 - y_1 \sin\theta_1 \\ x_1 \sin\theta_1 - y_1 \cos\theta_1 \\ -\theta_1 \end{bmatrix} = \begin{bmatrix} x_1^- \\ y_1^- \\ \theta_1^- \end{bmatrix}$$

```
function tba=tinv(tab)
s = sin(tab(3));
c = cos(tab(3));
tba = [-tab(1)*c - tab(2)*s;
        tab(1)*s - tab(2)*c;
       -tab(3)];
```

$$\frac{\partial \ominus p_1}{\partial p_1} = \frac{\partial f(p_1)}{\partial p_1} = \frac{\partial\{x_1^-, y_1^-, \theta_1^-\}}{\partial\{x_1, y_1, \theta_1\}} = \begin{bmatrix} -\cos\theta_1 & -\sin\theta_1 & x_1 \sin\theta_1 - y_1 \cos\theta_1 \\ \sin\theta_1 & -\cos\theta_1 & x_1 \cos\theta_1 + y_1 \sin\theta_1 \\ 0 & 0 & -1 \end{bmatrix}$$

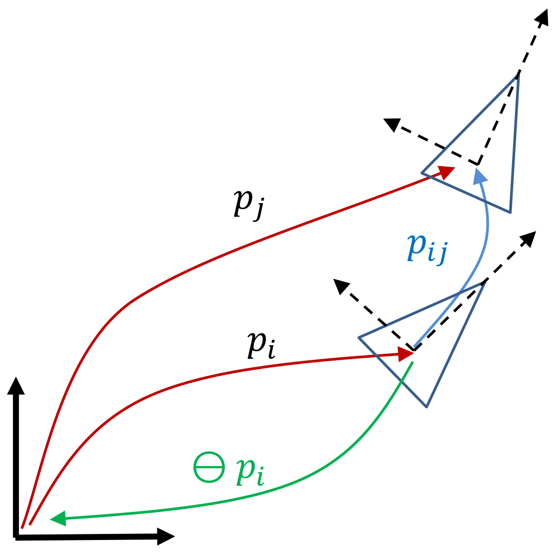
```
function Jac=Jba(tab)
s = sin(tab(3));
c = cos(tab(3));
c = cos(tab(3));
Jac = [-c -s  tab(1)*s-tab(2)*c
       s  -c tab(1)*c+tab(2)*s
        0   0   -1];
```

The inverse: $\dfrac{\partial p_1}{\partial \ominus p_1} = \left(\dfrac{\partial \ominus p_1}{\partial p_1}\right)^{-1}$

Covarianze: $\Sigma_{\ominus p_1} = \dfrac{\partial \ominus p_1}{\partial p_1} \Sigma_{p_1} \left(\dfrac{\partial \ominus p_1}{\partial p_1}\right)^T$

Appendix

Inverse composition:



$$p_{ij} = \ominus p_i \oplus p_j = p_j \ominus p_i = f(p_j, p_i) =$$

$$= \begin{bmatrix} (x_j - x_i)\cos\theta_i + (y_j - y_i)\sin\theta_i \\ -(x_j - x_i)\sin\theta_i + (y_j - y_i)\cos\theta_i \\ \theta_j - \theta_i \end{bmatrix}$$

Jacobians:

$$\frac{\partial p_{ij}}{\partial p_j} = \frac{\partial f(p_j, p_i)}{\partial\{x_j, y_j, \theta_j\}} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial p_{ij}}{\partial p_i} = \frac{\partial f(p_j, p_i)}{\partial\{x_i, y_i, \theta_i\}} = \begin{bmatrix} -\cos\theta_i & -\sin\theta_i & -(x_j - x_i)\sin\theta_i + (y_j - y_i)\cos\theta_i \\ \sin\theta_i & -\cos\theta_i & -(x_j - x_i)\cos\theta_i - (y_j - y_i)\sin\theta_i \\ 0 & 0 & -1 \end{bmatrix}$$