L05: Motion and sensor model

Gonzalo Ferrer

9 February 2021

1 Introduction to motion models

$$x_t = g(x_{t-1}, u_t, \varepsilon_t) \qquad \text{in general Non-linear.} \tag{1}$$

As designers we should find the right Transition function g(·) that better describes our system.

Ex:

$$\left.\begin{array}{l}\text{free point 2D, 3D}\\ \text{wheeled robot}\end{array}\right\} \text{ Introduction for today.}$$

car,

airplane,

quadrotor

serial manipulator,···

2 Free point 2D: Kinematics

$$x_t = \underset{\text{position}}{\begin{bmatrix} x \\ y \end{bmatrix}} \quad u_t = \underset{\text{velocity}}{\begin{bmatrix} v_x \\ v_y \end{bmatrix}} \tag{2}$$

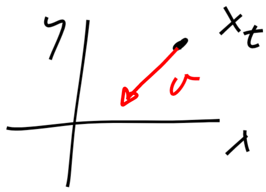

Figure 1: Free point 2D: Kinematics

This corresponds to a free moving point, for now deterministic model (no noise). There are two approaches:

Discrete-time model	$x_t = g(x_{t-1}, u_t)$	
Continuous-time model	$\dot{x}_t = f(x, u)$	(Transition eq.)

$$\dot{x} = u = \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} \quad \text{cont-time velocities} \tag{3}$$

$$x(t) = x(0) + \int_0^t f(x, u) dt' \quad \text{continuous-time trajectory} \tag{4}$$

2.1 Numerical method: Euler method, 1st order method

$$\dot{x} = \frac{dx}{dt} \simeq \frac{x(\Delta t) - x(0)}{\Delta t} \tag{5}$$

$$x(\Delta t) = x(0) + \Delta t \cdot \dot{x} = x(0) + \Delta t \cdot f(x(0), u(0)) \tag{6}$$

An alternative to the Euler method are Runge-Kutta methods: which are higher order integration methods.

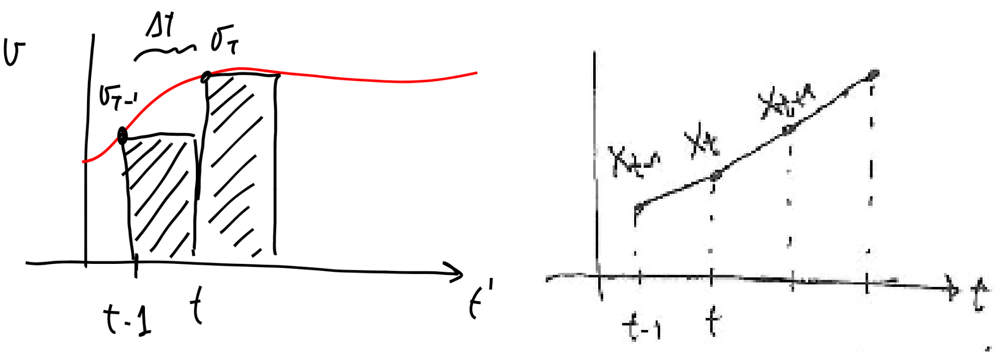

Figure 2: Euler method. On the left, piece-wise constant integration of velocity. On the right, piece-wise linear trajectory.

As a result, the 2D Kinematic point:

$$x_t = x_{t-1} + \Delta t \cdot f(x,u) = x_{t-1} + \Delta t \overbrace{\begin{bmatrix} v_x \\ v_y \end{bmatrix}}^{u} \quad \text{(Transition function)} \tag{7}$$

$$A = I_{2\times 2} \qquad B = \Delta t \cdot I_{2\times 2}$$

2.2 Probabilistic model 2D Kinematic point

$$x_t = x_{t-1} + \Delta t \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \eta \qquad \backslash\, \eta \sim \mathcal{N}(0, R) \tag{8}$$

Noise is added to the state space (ProbRob p.127)

$$R = \begin{bmatrix} \alpha_1 v_x^2 + \alpha_2 v_y^2 & 0 \\ 0 & \alpha_3 v_x^2 + \alpha_4 v_y^2 \end{bmatrix} \tag{9}$$

Sometimes, the noise variable is more convenient to be added in the action space:

$$x_t = I \cdot x_{t-1} + \Delta t \cdot I \begin{bmatrix} v_x + \eta_x^{'} \\ v_y + \eta_y^{'} \end{bmatrix} = x_{t-1} + \Delta t \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \underbrace{\Delta t \begin{bmatrix} \eta_x^{'} \\ \eta_y^{'} \end{bmatrix}}_{\eta} \tag{10}$$

where $\eta' \sim \mathcal{N}(0, M)$ can also be expressed in the action space if we transform the r.v. such as $\eta \sim \mathcal{N}(0, BMB^T)$. It will be crucial to check in which space the noise is defined.

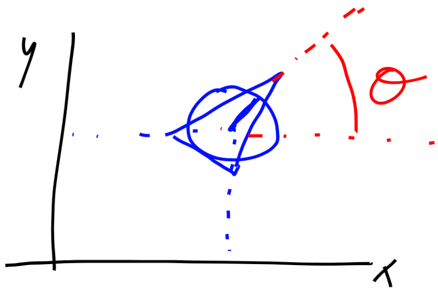

Figure 3: 2D pose. Position and orientation (or heading)

3 2D pose: Position and orientation (or heading)

$$x = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad \text{is the state variable.} \qquad \dot{x} = \underbrace{\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}}_{u} \text{is the control variable.}$$

$$x_t = x_{t-1} + \Delta t \cdot u = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \Delta t \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_t \tag{11}$$

Q: are the heading and position variables decoupled?
There are 3 control variables for 3 state variables $\Rightarrow \forall x \in X$ is reachable.
Linear approximation $\rightarrow$ build probabilities model as before.

Wheeled robots in 2D

$$x_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad \text{State is a pose: position and orientation}$$

Why not 3D? Wheeled robots (normally) stay on the ground, which we locally approximate as a plane (2D)

4 Kinematic Unicycle

We have already presented the c-t transition equation $\dot{x} = f(x, u)$. This is a deterministic function.
Wheels add constraints: heading and velocity are related, see Fig. 4.

$$\frac{dy}{dx} = \tan(\theta) \tag{12}$$

$$\frac{\dot{y}}{\dot{x}} = \frac{\sin\theta}{\cos\theta} \quad \Rightarrow \quad (-\dot{x}\sin\theta + \dot{y}\cos\theta) \cdot v = 0 \tag{13}$$

$$\Rightarrow \quad \begin{matrix} \dot{x} = v \cdot \cos\theta \\ \dot{y} = v \cdot \sin\theta \end{matrix} \quad \text{proof: } -v\underbrace{\cos\theta}_{\dot{x}}\sin\theta + v\underbrace{\sin\theta}_{\dot{y}}\cos\theta = 0 \tag{14}$$

$$\dot{x} = f(x, u) = \begin{bmatrix} v \cdot cos\theta \\ v \cdot sin\theta \\ \omega \end{bmatrix} \qquad u = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

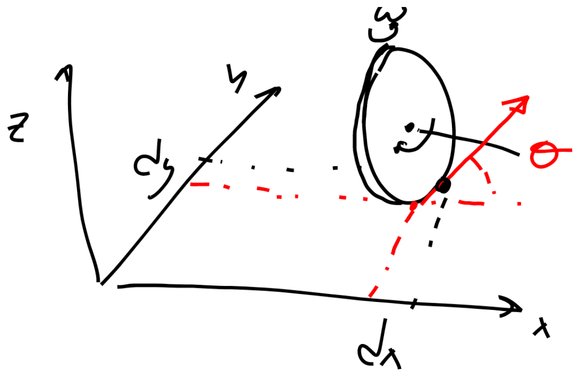

Figure 4: Kinematic Unicycle. Imagine a large disk, it can only move on the theta direction for infinitesimally small increments. Therefore, the constraints to the system follow from this fact.

Still ∀ x ∈ X is reachable. (Proof in Lavalle book).

$$x_t = \underbrace{\begin{bmatrix} \chi_{t-1} + \Delta t \cdot v_t \cos\theta_{t-1} \\ y_{t-1} + \Delta t \cdot v_t \sin\theta_{t-1} \\ \theta_{t-1} + \Delta t \cdot \omega_t \end{bmatrix}}_{g(x_{t-1},u_t)} = I \cdot x_{t-1} + \underbrace{\begin{bmatrix} \Delta t \cdot \cos\theta_{t-1} & 0 \\ \Delta t \cdot \sin\theta_{t-1} & 0 \\ 0 & \Delta t \end{bmatrix}}_{B(x) \text{ - Non-linear}} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{15}$$

4.1 Unicycle: probabilistic model

$$x_t = g(x_{t-1}, u_t) \simeq g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1}) + V_t(u_t - \overline{u}_t), \tag{16}$$

where the mean value $u_t = \overline{u}_t + \delta u_t$.

$$G_t = \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}}\bigg|_{\mu_{t-1}} = \begin{array}{c} \begin{bmatrix} 1 & 0 & -\sin\theta_{t-1} \cdot \Delta t v_t \\ 0 & 1 & \cos\theta_{t-1} \cdot \Delta t v_t \\ 0 & 0 & 1 \end{bmatrix} \\ \partial/\partial x \quad\quad \partial/\partial y \quad\quad \partial/\partial\theta \end{array} \tag{17}$$

$$V_t = \frac{\partial g(x_{t-1}, u_t)}{\partial u_t}\bigg|_{\overline{u}_t} = \begin{bmatrix} cos\theta_{t-1} \cdot \Delta t & 0 \\ sin\theta_{t-1} \cdot \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{18}$$

Add noise

$$x_t = g(x_{t-1}, u_t, \varepsilon_t) = g(x_{t-1}, u_t) + \varepsilon_t \tag{19}$$

$$\begin{aligned} &x_{t-1} \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}) \quad , \quad \varepsilon_t \sim \mathcal{N}(0, R) \qquad \text{(state space noise)} \\ &\Rightarrow x_t \sim N(g(\mu_{t-1}, u_t), G_t \Sigma G_t^T + R) \end{aligned} \tag{20}$$

Q: Show this at home.

Example: Circular path give the controls $\theta = 1, \omega = 1$.

Can we do better than Euler Integration? ProbRob Ch5.3 uses a sequence of arcs to approximate the unicycle:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \begin{bmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t + \underbrace{\gamma\Delta t}_{\text{Extra term: final orientation}} \end{bmatrix}_t = g(x_{t-1}, u_t) \tag{21}$$

Figure 5: Circular path. In the first order approximation we use linear segments, which has associated some error.

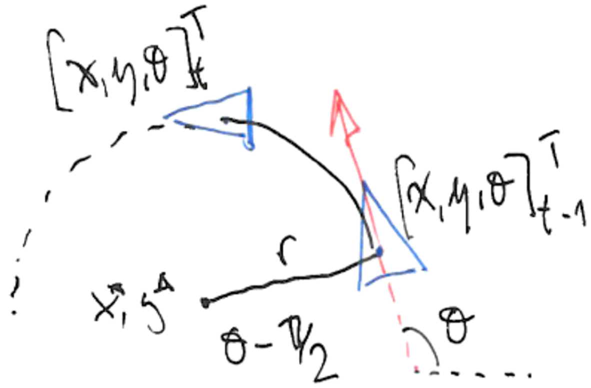

Figure 6: Approximation of the unicycle. In this approach there is a center of rotation (x^*, y^*), from which the robot describes an arc.

4.2 Probabilistic Arc Kinematic motion model

$$\begin{bmatrix} \hat{v} \\ \hat{\omega} \end{bmatrix} = \begin{bmatrix} v + \varepsilon_v \\ \omega + \varepsilon_\omega \end{bmatrix} \quad , \quad \varepsilon \sim \mathcal{N}(0, M) \qquad \text{(Jacobians in ProbRob p.127)}$$

$$x_t \sim N\big(g(\mu_{t-1}, u_t), \quad G_t \Sigma_{t-1} G_t^T + V_t M V_t^T\big). \tag{22}$$

5 Odometry model

Counts of increments on spinning wheels (observation) more accurate than velocity models.

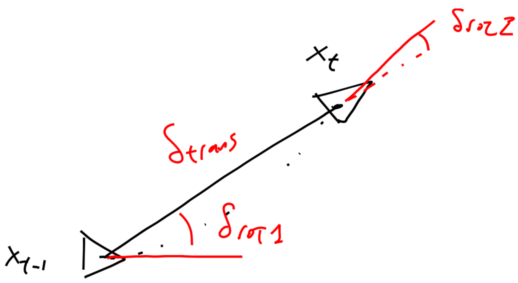

Figure 7: Odometry model. The robot first rotates, then translates in the heading direction and translates a second time to reach any possible end configuration.

$$u = \begin{bmatrix} \delta_{rot1} \\ \delta_{trans} \\ \delta_{rot2} \end{bmatrix} = \begin{bmatrix} atan2(y_t - y_{t-1}, x_t - x_{t-1}) \\ \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \\ \theta_t - \theta_{t-1} - \delta_{rot1} \end{bmatrix} \tag{23}$$

Discrete-time transition function:

$$x_t = g(x_{t-1}, u_y) = \begin{bmatrix} x_{t-1} + \delta_{trans}\cos(\theta + \delta_{rot1}) \\ y_{t-1} + \delta_{trans}\sin(\theta + \delta_{rot1}) \\ \theta_{t-1} + \delta_{rot1} + \delta_{rot2} \end{bmatrix} \tag{24}$$

5.1 Probabilistic odometry model

$$x_t = g(x_{t-1}, u_t, \varepsilon_t) \qquad \overset{\text{(noise in action space)}}{\varepsilon_t \sim \mathcal{N}(0, M)} \qquad \text{(ProbRob p.139)}$$

$$\varepsilon_t = \begin{bmatrix} \varepsilon_{\delta_{rot1}} \\ \varepsilon_{\delta_{trans}} \\ \varepsilon_{\delta_{rot2}} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \alpha_1\delta_{rot1}^2 + \alpha_2\delta_{trans}^2 & 0 & 0 \\ 0 & \alpha_3\delta_{trans}^2 + \alpha_4(\delta_{rot1}^2 + \delta_{rot2}^2) & 0 \\ 0 & 0 & \alpha_1\delta_{rot2}^2 + \alpha_2\delta_{trans}^2 \end{bmatrix}\right) \tag{25}$$

5.2 Jacobian for odometry model

$$G_t = \left.\frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}}\right|_{\mu_{t-1}} = \begin{array}{c} \begin{bmatrix} 1 & 0 & -\delta_{trans}\sin(\theta + \delta_{rot1}) \\ 0 & 1 & \delta_{trans}\cos(\theta + \delta_{rot1}) \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{matrix} \partial/\partial x & \partial/\partial y & \partial/\partial\theta \end{matrix} \end{array} \tag{26}$$

$$V_t = \left.\frac{\partial g([x_{t-1}, u_t)}{\partial u_t}\right|_{\overline{u}_t} = \begin{array}{c} \begin{bmatrix} -\delta_{trans}\cdot\sin(\theta + \delta_{rot1}) & \cos(\theta + \delta_{rot1}) & 0 \\ \delta_{trans}\cdot\cos(\theta + \delta_{rot1}) & \sin(\theta + \delta_{rot1}) & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ \begin{matrix} \partial/\partial\delta_{rot1} & \partial/\partial\delta_{trans} & \partial/\partial\delta_{rot2} \end{matrix} \end{array} \tag{27}$$

6 2D-Rigid Body Transformations

2D poses can be interpreted as a transformation between coordinate frames → XYT parameterization. In the example below, between the global frame g and the robot frame r.

$${}^g x_r = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

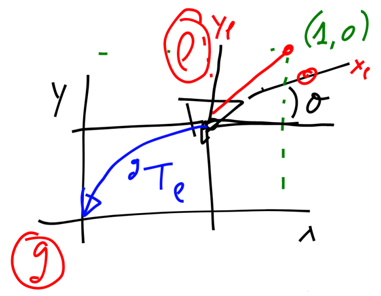

Figure 8: 2D-Rigid Body Transformation. From the local frame to the world frame. This transformation is expressed as a 2D pose.

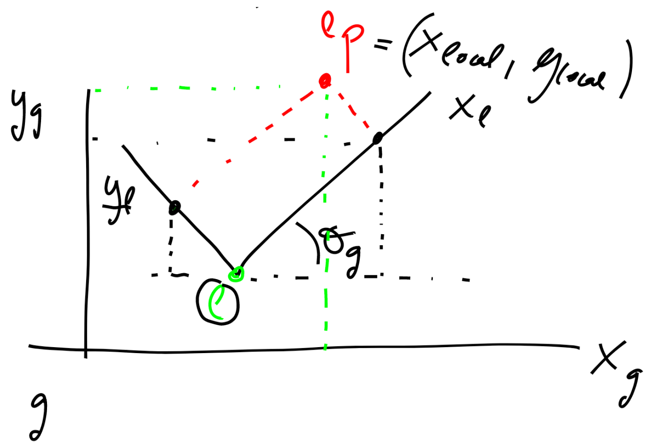

Figure 9: 2D-Rigid Body Transformations 2

We need to project point $^l p$ in the local frame (or robot frame) to the global frame g.

$$^g p = \begin{bmatrix} x_{local} \cdot \cos\theta_g - y_{local} \sin\theta_g + t_x \\ x_{local} \cdot \sin\theta_g - y_{local} \cos\theta_g + t_y \end{bmatrix} = \underbrace{\begin{bmatrix} cos\theta_g & -sin\theta_g & t_x \\ sin\theta_g & cos\theta_g & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{local} \\ y_{local} \\ 1 \end{bmatrix}}_{\text{(homogeneous coordinates)}} \tag{28}$$

We can express this transformation more compactly as a linear transformation of homogeneous coordinates:

$$^g p = \begin{bmatrix} ^g R_l(\theta_r) & t \\ 0_{1\times 2} & 1 \end{bmatrix} \cdot {}^l p = {}^g T_l \cdot {}^l p = {}^g p. \tag{29}$$

Note that the superscript with the point coordinates, has to match the subcript of the transformation, and as a result the new frame of the point is the global frame.

7 RBT composition

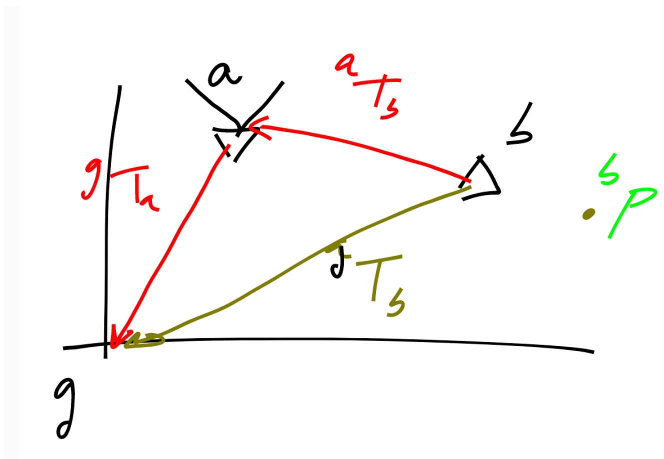

Figure 10: RBT composition. Two solution to transform the point $^b p$.

Two solutions to transform the same point, one directly from $b \to g$ and the second $b \to a \to g$:

$$
\begin{aligned}
{}^g p &= {}^g T_b \cdot {}^b p \\
{}^a p &= {}^a T_b \cdot {}^b p \\
{}^g p &= {}^g T_a \cdot {}^a p = {}^g T_a \cdot {}^a T_b \cdot {}^b p
\end{aligned}
$$

8 Sensor models

$$
\begin{aligned}
&p(z_t|x_t) && \text{Observation distribution (Lecture 4)} \\
&z_t = h(x_t, \delta_t) && \text{Observation function}
\end{aligned}
$$

Example: Accelerometer, sensor, radar, Lidar, camera, ...
We will propose a sensor model, but any function relating state and observation is a valid sensor model.

8.1 Feature-based measurements model

The idea is to extract features f from observations $f(z_t) = \{f_1, f_2, \cdots f_n\}$.
Example: lines, corners, point description, objects, etc...

Landmark definition: feature which corresponds to physical objects.

$$
(2D) \quad f_i = \begin{bmatrix} \text{'range'} \\ \text{'bearing'} \\ \text{'signature'} \end{bmatrix} = \begin{bmatrix} r \\ \phi \\ s \end{bmatrix}
$$

s is optional to alleviate data association problem. (color, size, $\mathbb{R}^{60}$ embedding, etc ...)

each feature corresponds to a location $[m_{i,x}, m_{i,y}]^T$

$$
\begin{bmatrix} r_t^j \\ \phi_t^j \\ s_t^j \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2} \\ atan2(m_{i,y} - y, m_{i,x} - x) - \theta \\ s_i \end{bmatrix} + \begin{bmatrix} \delta_{\sigma_r^2} \\ \delta_{\sigma_\phi^2} \\ \delta_{\sigma_s^2} \end{bmatrix} \quad , \quad \delta \sim \mathcal{N}(0, \Sigma_\delta) \tag{30}
$$

The data association problem: the i^{th} map location m_i corresponds to the j^{th} feature.

$$
f^j = h(x_t, m_i) + \delta_t \tag{31}
$$

$$
f \sim \mathcal{N}(f(z_T); h(x_t, m_i), \Sigma_\delta) \qquad \text{Probabilistic model} \tag{32}
$$

Sampling x_t from m_i in ProbRob 280.

9 Summary

- Transition function: $x_t = g(x_{t-1}, u_t)$, discrete-time model obtained by integrating $\dot{x} = f(x, u)$ (NL)
- Probabilistic model:
  - Add noise to the state/action space
  - Linearize $g(\cdot)$
  - Covariance transformation: $x_t \sim N(g(\mu_{t-1}, u_t), G_t \Sigma G_t^T + R))$
- RBT 2D: $p = {}^g T_l \cdot {}^l p = {}^g T_a \cdot {}^a T_l \cdot {}^l p$ – chain of transformations

- Observation function → Landmarks:

$$m_i = \begin{bmatrix} m_{i,x} & m_{i,y} \end{bmatrix}^T$$

$$\text{(features from sensor data:) } z = h(x, m_i) = \begin{bmatrix} r & \phi & s \end{bmatrix}^T$$

  – range, bearing, appearance

$$p(z|x) \sim N(z; h(\mu_x, m), \Sigma_z)$$