ECE276A: Sensing & Estimation in Robotics
Lecture 9: Motion and Observation Models

Lecturer:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Siwei Guo: s9guo@eng.ucsd.edu

Anwesan Pal: a2pal@eng.ucsd.edu

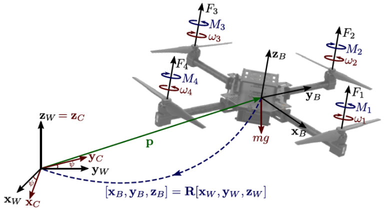
UC San Diego

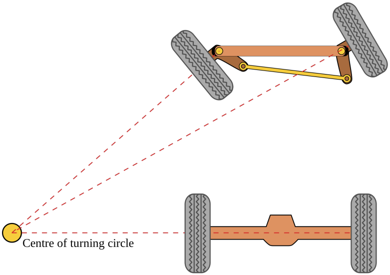
JACOBS SCHOOL OF ENGINEERING

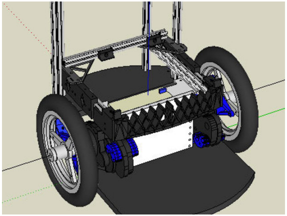
Electrical and Computer Engineering

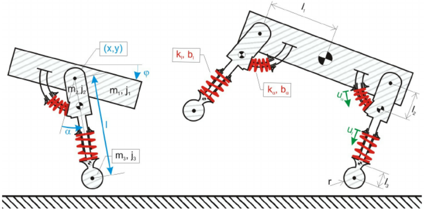
Motion Models

- **Goal**: model the likelihood $p_a(\cdot \mid x, u)$ for given state x and control u
- Based on robot kinematics or dynamics:
  - Differential drive model
  - Ackermann drive (bicycle) model
  - Quadrotor model
  - Legged locomotion model
- Based on **odometry**:
  - sensor data (e.g., wheel encoders, IMU, camera, laser) used to estimate ego motion
  - Only available in retrospect: useful for filtering but not for motion planning









Motion Models

- **Differential drive motion model**: position $p \in \mathbb{R}^2$, orientation $\theta \in (-\pi, \pi]$, linear velocity $v \in \mathbb{R}$, rotational veclotiy $\omega \in \mathbb{R}$:

  State: $x = (p, \theta)$

  Control: $u = (v, \omega)$

  Motion Model: $$\dot{x} = a(x, u) = \begin{cases} \dot{p} = v \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \\ \dot{\theta} = \omega \end{cases}$$

- **Quadrotor motion model**: position $p \in \mathbb{R}^3$, velocity $\dot{p} \in \mathbb{R}^3$, orientation $R \in SO(3)$, rotational velocity $\omega_B \in \mathbb{R}^3$, thrust $f \in \mathbb{R}$, moment $M_B \in \mathbb{R}^3$, mass $m \in \mathbb{R}_{>0}$, gravitational acceleration g, moment of inertial $J \in \mathbb{R}^{3\times 3}$, z-axis $e_3 \in \mathbb{R}^3$:

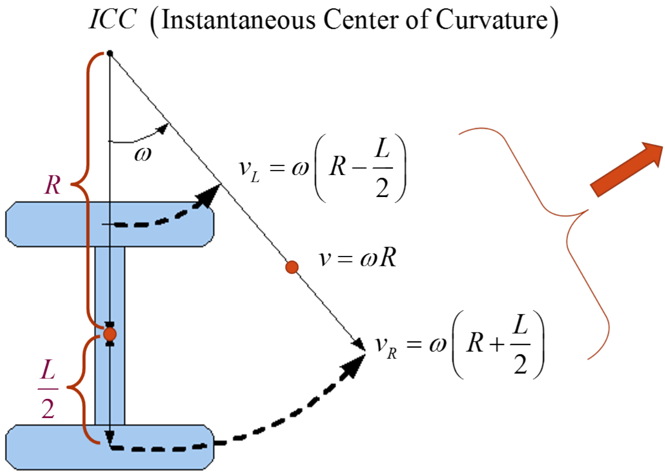
  State: $x = (p, \dot{p}, R, \omega_B)$

  Control: $u = (f, M_B)$

  Motion Model: $$\dot{x} = a(x, u) = \begin{cases} m\ddot{p} = -mge_3 + fRe_3 \\ \dot{R} = R\hat{\omega}_B \\ J\dot{\omega}_B = -\omega_B \times J\omega_B + M_B \end{cases}$$

Differential Drive Motion Model

- Let $s_t := (x_t, y_t, \theta_t)^T \in SE(2)$ be the pose of a differential drive robot
- What is the state $s_{t+\tau}$ after τ seconds if we apply linear velocity v_t and angular velocity ω_t?



$$\omega = \frac{v_R - v_L}{L}$$

$$R = \frac{L}{2}\left(\frac{v_L + v_R}{v_R - v_L}\right) = \frac{v}{\omega}$$

$$v = \frac{v_R + v_L}{2}$$

$$\dot{x}(t) = v\cos\theta(t)$$

$$\dot{y}(t) = v\sin\theta(t)$$

$$\dot{\theta}(t) = \omega$$

Discrete-time Differential Drive Model

- To convert the continuous-time differential drive model to discrete time, we can solve the ordinary differential equations:

$$
\begin{aligned}
\dot{x}(t) &= v\cos\theta(t) \\
\dot{y}(t) &= v\sin\theta(t) \quad \Rightarrow \\
\dot{\theta}(t) &= \omega
\end{aligned}
$$

$$
\begin{aligned}
\theta(t) &= \theta(t_0) + \int_{t_0}^{t} \omega ds = \theta(t_0) + \omega(t - t_0) \\
x(t) &= x(t_0) + v\int_{t_0}^{t} \cos\theta(s)ds \\
&= x(t_0) + \frac{v}{\omega}\left(\sin\left(\omega(t-t_0)+\theta(t_0)\right) - \sin\theta(t_0)\right) \\
&= x(t_0) + v(t-t_0)\frac{\sin(\omega(t-t_0)/2)}{\omega(t-t_0)/2}\cos(\theta(t_0) + \omega(t-t_0)/2) \\
y(t) &= y(t_0) + v\int_{t_0}^{t} \sin\theta(s)ds \\
&= y(t_0) - \frac{v}{\omega}\left(\cos\theta(t_0) - \cos\left(\omega(t-t_0) + \theta(t_0)\right)\right) \\
&= y(t_0) + v(t-t_0)\frac{\sin(\omega(t-t_0)/2)}{\omega(t-t_0)/2}\sin(\theta(t_0) + \omega(t-t_0)/2)
\end{aligned}
$$

- Let $\tau := t - t_0$ be the time discretization

Discrete-time Differential Drive Model

- The discrete-time differential drive motion model with state $s_t := (x_t, y_t, \theta_t)^T \in SE(2)$, time discretization τ, and control input $u_t := (v_t, \omega_t)^T$ is:

$$s_{t+1} = a(s_t, u_t) := s_t + \tau \begin{pmatrix} v_t \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \cos\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ v_t \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \sin\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ \omega_t \end{pmatrix}$$

- EKF derivatives:

$$\frac{da}{ds}(s, u) = \begin{bmatrix} 1 & 0 & -v_t \tau \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \sin\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ 0 & 1 & v_t \tau \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \cos\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{da}{du}(s, u) = \tau \begin{bmatrix} \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \cos\left(\theta_t + \frac{\omega_t \tau}{2}\right) & v_t g_1\left(\frac{\omega_t \tau}{2}\right) \\ \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \sin\left(\theta_t + \frac{\omega_t \tau}{2}\right) & v_t g_2\left(\frac{\omega_t \tau}{2}\right) \\ 0 & 1 \end{bmatrix}$$

where $g_1(z) := -\mathbf{sinc}(z) \sin(\theta_t + z) + \cos(\theta_t + z) \frac{z \cos(z) - \sin(z)}{z^2}$ and

$$g_2(z) := \mathbf{sinc}(z) \cos(\theta_t + z) + \sin(\theta_t + z) \frac{z \cos(z) - \sin(z)}{z^2}$$

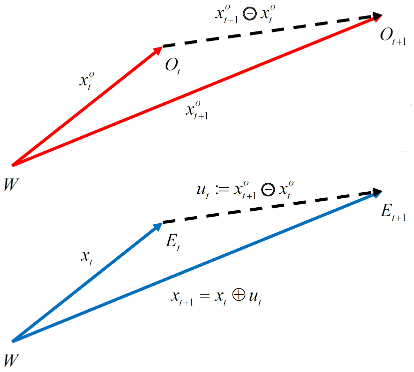
Odometry-based Motion Model

- A "drifting" estimate of the robot trajectory $x^o_{0:t} \subset SE(3)$ is provided by the motion sensors
- How can the odometry estimate be used to design a motion model for a filtering prediction step?

- **Idea**: add only a local transformation provided by the odometry to the current robot pose $x_t \in SE(3)$

- Odometry-based motion model with input $u_t := x^o_{t+1} \ominus x^o_t \in SE(3)$ and noise $w_t \in SE(3)$

$$\boxed{x_{t+1} = x_t \oplus w_t \oplus u_t}$$



Smart Plus Derivative

- Parameterize the pose $x_t \in SE(3)$ via position s_t and rotation vector r_t
- Parameterize the noise $w_t \in SE(3)$ with position noise $\eta_t \sim \mathcal{N}(0, W_{3\times 3})$ and rotation vector noise $\xi_t \sim \mathcal{N}(0, R_{3\times 3})$
- Parameterize the odometry $u_t \in SE(3)$ via translation p_t and rotation R_t
- The derivative of the exponential map, for $a, b \in \mathbb{R}^3$, is:

$$\frac{d}{da}\exp(\hat{a})b = \exp(\hat{a})\frac{d}{da}\hat{a}b = \exp(\hat{a})\frac{d}{da}\left(-\hat{b}a\right) = -\exp(\hat{a})\hat{b}$$

- The odometry motion model is:

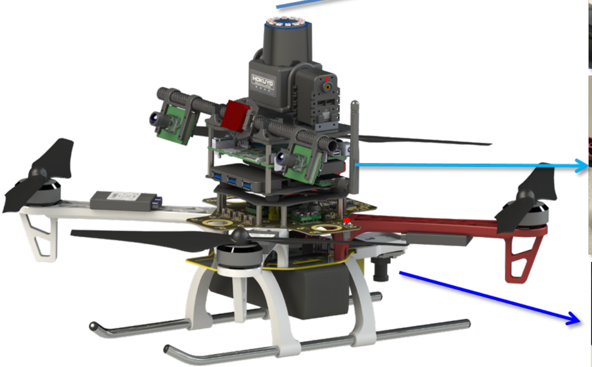
$$a(x_t, u_t, w_t) := \begin{bmatrix} e^{\hat{r}_t} & s_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{\hat{\xi}_t} & \eta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_t & p_t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{\hat{r}_t} & s_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} D & d \\ 0 & 1 \end{bmatrix}$$

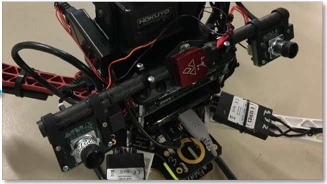
- The derivatives with respect to s_t and r_t are:

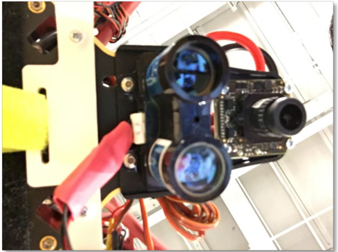
$$\frac{da}{ds_t} := \left[\frac{d}{ds_t} e^{\hat{r}_t} D \quad \frac{d}{ds_t}\left(e^{\hat{r}_t} d + s_t\right)\right] = \left[\mathbf{0}_{3\times 9} \quad I_{3\times 3}\right]$$

$$\frac{da}{dr_t} := -e^{\hat{r}_t}\left[\hat{d}_1 \quad \hat{d}_2 \quad \hat{d}_3 \quad \hat{d}\right]_{3\times 12}, \qquad \text{where } D =: \left[d_1 \quad d_2 \quad d_3\right]$$

Quadrotor Sensors





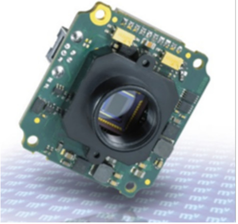


Observation Models

- **Goal**: model the measurement likelihood $p_h(z \mid x, m)$ for a given sensor pose x and environment representation m
- **Position model**: direct position measurements, e.g., GPS, RGBD camera, laser scanner
- **Bearing model**: angular measurements to points in 3-D, e.g., compass, RGB camera
- **Range model**: distance measurements to points in 3-D, e.g., radio received signal strength (RSS) or time-of-flight
- **Inertial measurement unit**: magnetometer, gyroscope, accelerometer



Cameras



Global shutter

Stereo (+ IMU)

RGBD

Event-based

Lasers

Single-beam Garmin Lidar

2-D Hokuyo Lidar

3-D Velodyne Lidar

Observation Models

- **Position sensor**: state $x = (p, \theta)$, position $p \in \mathbb{R}^2$, orientation $\theta \in (-\pi, \pi]$, observed point $y \in \mathbb{R}^2$, measurement $z \in \mathbb{R}^2$:

$$z = h(x, y) = R(\theta)^T (y - p)$$

- **Range sensor**: state $x = p$, position $p \in \mathbb{R}^n$, observed point $y \in \mathbb{R}^n$, distance $z \in \mathbb{R}$:

$$z = h(x, y) = \|p - y\|_2$$

- **Bearing sensor**: state $x = (p, \theta)$, position $p \in \mathbb{R}^2$, orientation $\theta \in (-\pi, \pi]$, observed point $y \in \mathbb{R}^2$, bearing $z \in (-\pi, \pi]$:

$$z = h(x, y) = \arctan\left(\frac{y_2 - p_2}{y_1 - p_1}\right) - \theta$$

- **Camera sensor**: position $p \in \mathbb{R}^3$, orientation $R \in SO(3)$, intrinsic camera matrix $K \in \mathbb{R}^{2\times 3}$, observed point $y \in \mathbb{R}^3$, pixel $z \in \mathbb{N}^2$:

State:	$x = (p, R)$
Projection:	$\pi(p) := \frac{1}{p_3} p$
Observation Model:	$z = h(x, y) = K\pi(R^T(y - p))$

Observation Model Linearization

- **Position sensor**: $z = h(x, y) = R(\theta)^T(y - p)$ for $x = (p, \theta)$

$$R(\theta) = \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix} \quad \frac{dR}{d\theta}(\theta) = \begin{bmatrix}-\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta\end{bmatrix} = JR(\theta) \quad J := \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}$$

$$\frac{dh}{dx}(x, y) = \begin{bmatrix}-R^T(\theta) & R^T(\theta)J^T(y - p)\end{bmatrix} \qquad \frac{dh}{dy}(x, y) = R^T(\theta)$$

- **Range sensor**: $z = h(x, y) = \|p - y\|_2$ for $x = p$

$$\frac{dh}{dx}(x, y) = -\frac{(y - x)^T}{\|y - x\|_2} \qquad \frac{dh}{dy}(x, y) = \frac{(y - x)^T}{\|y - x\|_2}$$

- **Bearing sensor**: $z = h(x, y) = \arctan\left(\frac{y_2 - p_2}{y_1 - p_1}\right) - \theta$ for $x = (p, \theta)$

$$\frac{dh}{dx}(x, y) = \begin{bmatrix}\frac{(y-p)^T J}{\|y-p\|_2} & -1\end{bmatrix} \qquad \frac{dh}{dy}(x, y) = \frac{(y - p)^T J}{\|y - p\|_2} \quad J := \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}$$

Observation Model Linearization

- **Camera sensor**: $z = h(x, y) = K\pi(R^T(y - p))$ for $x = (p, R = e^{\hat{a}})$

$$\pi(x) = \frac{1}{x_3}x = \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{pmatrix} \qquad \frac{d\pi}{dx}(x) = \frac{1}{x_3^2}\begin{bmatrix} x_3 & 0 & -x_1 \\ 0 & x_3 & -x_2 \\ 0 & 0 & 0 \end{bmatrix}$$

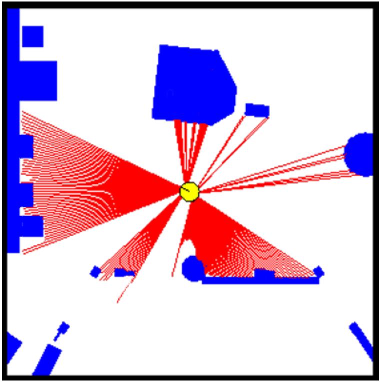
$$\frac{dh}{dp}(x, y) = -K\frac{d\pi}{dx}\left(R^T(y - p)\right)R^T$$

$$\frac{dh}{da}(x, y) = K\frac{d\pi}{dx}\left(R^T(y - p)\right)R^T(\widehat{y - p}) \qquad \text{for } R = e^{\hat{a}}$$

$$\frac{dh}{dy}(x, y) = K\frac{d\pi}{dx}\left(R^T(y - p)\right)R^T$$

Laser Scanner

- **Lidar**: LIght Detection And Ranging
- illuminates a target with pulsed laser light and measures the reflected pulses (return times and wavelengths)
- Example: Hokuyo URG-04LX; detectable range: 0.02 to 4m; 240° field of view with 0.36° angular resolution (666 beams); 100 ms/scan



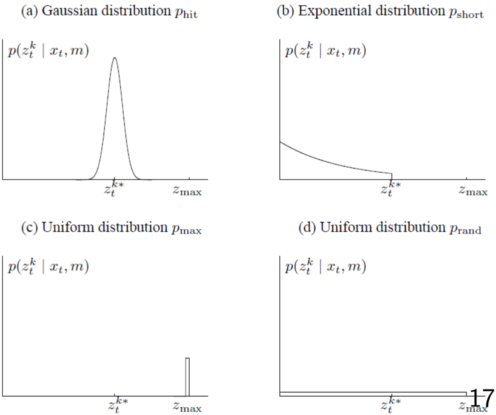
Laser Beam Model

- Let z_t^k be the k-th laser beam obtained from sensor pose x_t in map m
- Let z_t^{k*} be the expected range measurement from x_t in m and let z_{max} be the max laser range
- The laser sensor model assumes that the **beams are independent**:

$$p_h(z_t \mid x_t, m) = \prod_k p(z_t^k \mid x_t, m)$$

Four types of measurement noise:

1. Small measurement noise: p_{hit}, Gaussian
2. Unexpected object: p_{short}, Exponential
3. Unexplained noise: p_{rand}, Uniform
4. No objects hit: p_{max}, Uniform



(a) Gaussian distribution p_{hit}
(b) Exponential distribution p_{short}
(c) Uniform distribution p_{max}
(d) Uniform distribution p_{rand}

Laser Beam Model

- Independent beam assumption: $p_h(z_t \mid x_t, m) = \prod_k p(z_t^k \mid x_t, m)$
- Four types of noise:

$$p(z_t^k \mid x_t, m) = \alpha_1 p_{hit}(z_t^k \mid x_t, m) + \alpha_2 p_{short}(z_t^k \mid x_t, m) + \alpha_3 p_{rand}(z_t^k \mid x_t, m) + \alpha_4 p_{max}(z_t^k \mid x_t, m)$$

$$p_{hit}(z_t^k \mid x, m) = \begin{cases} \frac{\phi(z_t^k; z_t^{k*}, \sigma^2)}{\int_0^{z_{max}} \phi(s; z_t^{k*}, \sigma^2) ds} & \text{if } 0 \leq z_t^k \leq z_{max} \\ 0 & \text{else} \end{cases}$$

$$p_{short}(z_t^k \mid x, m) = \begin{cases} \frac{\lambda_s e^{-\lambda_s z_t^{k*}}}{1 - e^{-\lambda_s z_t^{k*}}} & \text{if } 0 \leq z_t^k \leq z_t^{k*} \\ 0 & \text{else} \end{cases}$$

$$p_{rand}(z_t^k \mid x, m) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \leq z_t^k < z_{max} \\ 0 & \text{else} \end{cases}$$

$$p_{max}(z_t^k \mid x, m) = \delta(z_t^k; z_{max}) := \begin{cases} 1 & \text{if } z_t^k = z_{max} \\ 0 & \text{else} \end{cases}$$