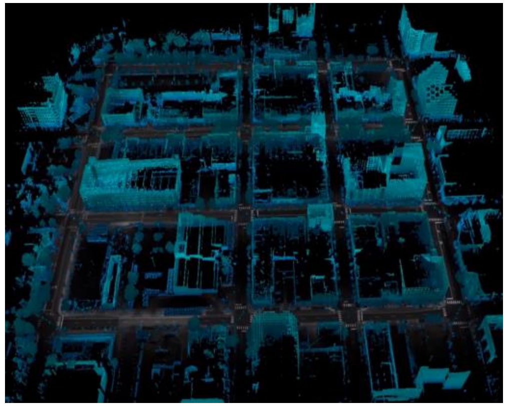
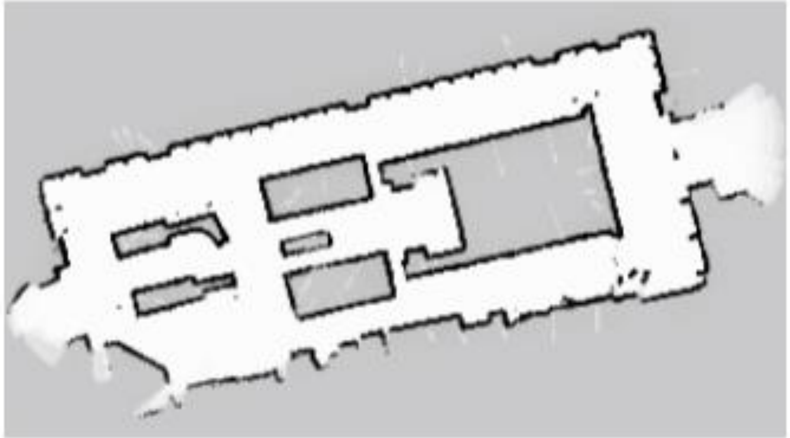
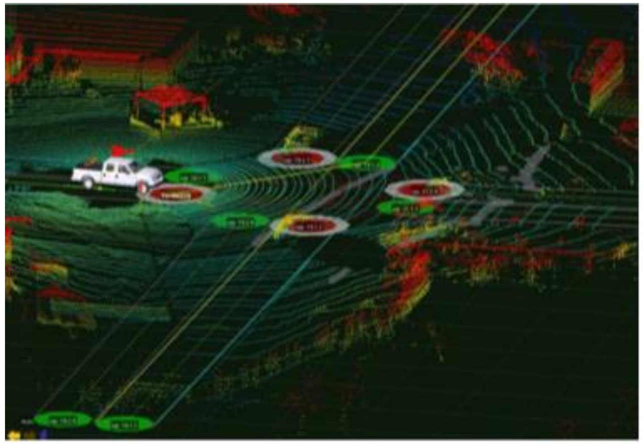
BYU Electrical & Computer Engineering





PROB. REVIEW - UNCERTAINTY PROPAGATION

ECEN 633: Robotic Localization and Mapping

Some slides courtesy of Ryan Eustice.

Agenda

- Uncertainty Propagation (Linear Case)
- Implementation Details
  - Characterizing Sensors
  - Sampling from Gaussians
- Uncertainty Propagation (Non-linear Case)

Uncertainty Propagation (Linear Case)

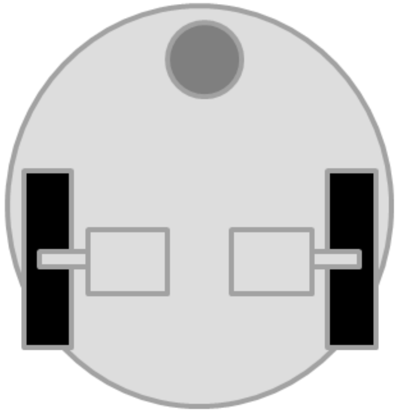
Uncertainty Projection/Propagation

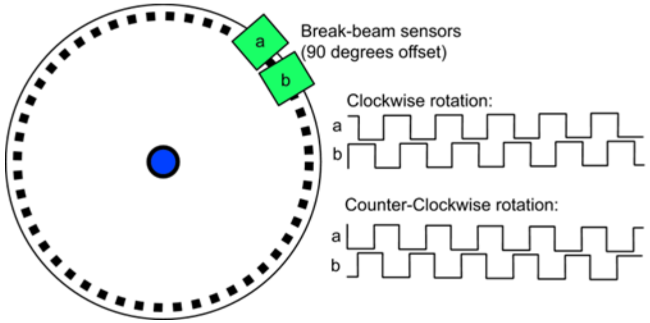
- Suppose I know $\mathbf{x} \sim \mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}$
- How do we handle $\mathbf{y} = \mathrm{A}\mathbf{x} + \mathbf{b}$ **???**

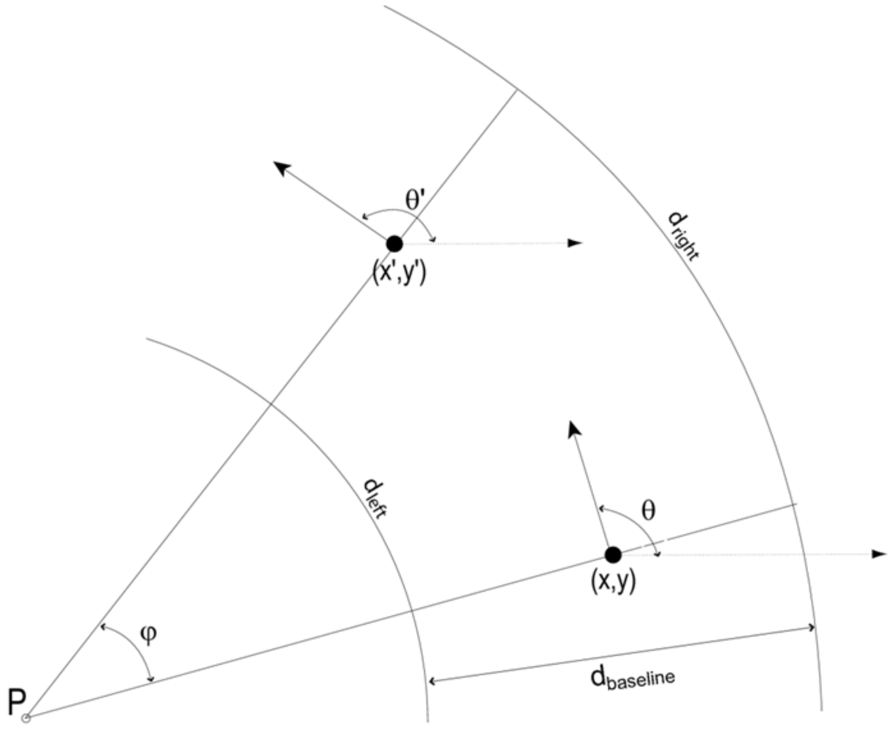
$$\Sigma_{\mathbf{yy}} = E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^\top]$$

- (Algebra) ➔ $\Sigma_{\mathbf{yy}} = \mathrm{A}\Sigma_{\mathbf{xx}}\mathrm{A}^\top$

Odometry Example





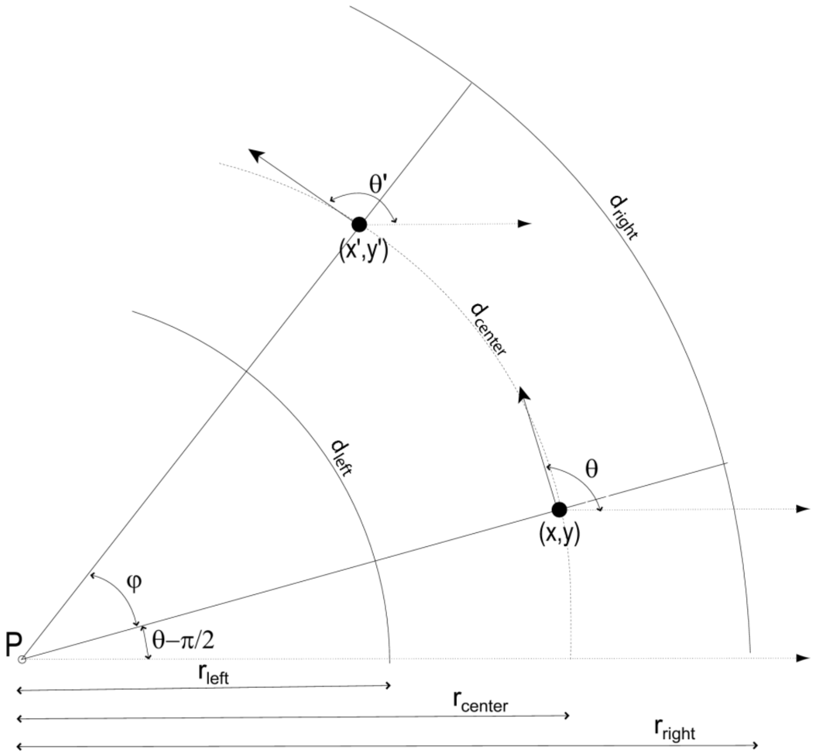


Odometry Example

- How to convert left/right ticks to a change in position?

$$\Delta x = \frac{d_R + d_L}{2}$$

$$\Delta\theta = \frac{d_R - d_L}{d_B}$$



Odometry Example

- Sensors observe:
  - Counts on left and right wheels

- No "noise" in those counts, however, there's slippage. Model distance as:

$$d_R = \alpha c_R + w_1$$

$$d_L = \alpha c_L + w_2$$

- Noise w_1, w_2 are iid Gaussian:

$$w_1, w_2 \sim N(0, \sigma^2)$$

Independent Identically Distributed

Odometry Example

- What is the uncertainty of $\Delta x,\ \Delta\theta$?
  - First, what's the uncertainty of d_R, d_L

$$\underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\boldsymbol{d}} = \underbrace{\begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}}_{\mathbf{w}}$$

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

$$\Sigma_d = A \Sigma_w A^T$$

But what's Σ_w ???

Odometry Example

$$\begin{bmatrix} d_R \\ d_L \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}$$

$$\Sigma_d = A \Sigma_w A^T$$

But what's Σ_w ???

Remember, we said c_R, c_L were "error-free", and (iid) $w_1, w_2 \sim N(0, \sigma^2)$

$$\Sigma_w = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}$$

Odometry Example

- We are half-way there now!

$$
\begin{aligned}
\Sigma_d &= \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix}^T \\
&= \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}
\end{aligned}
$$

- Does this make intuitive sense?
  - Answer is 2x2?
  - No alphas?

Odometry Example

- Where are we going again?
  - Trying to compute uncertainty of odometry measurements $\Delta x,\ \Delta\theta$
  - We know these in terms of : d_R, d_L
- We've gone from Σ_w to Σ_d
- Now, we need to go from Σ_d to Σ_x

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

Odometry Example

- Write $\boldsymbol{x}$ in terms of $\boldsymbol{d}$

$$\underbrace{\begin{bmatrix} \Delta x \\ \Delta\theta \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} d_R \\ d_L \end{bmatrix}}_{\mathbf{d}}$$

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$

$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta\theta = \frac{d_R - d_L}{d_B}$$

$$\Sigma_x = B\Sigma_d B^T$$

- We're done!

$$
\begin{aligned}
\Sigma_x &= \begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix}^\top \\
&= \begin{bmatrix} \sigma^2/2 & 0 \\ 0 & 2\sigma^2/d_B^2 \end{bmatrix}
\end{aligned}
$$

- Cross-correlations happen to cancel out
  - This does *not* happen in general!

Could do all this in one step

$$\begin{bmatrix} d_R \\ d_L \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 1 & 0 \\ 0 & \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} c_R \\ c_L \\ w_1 \\ w_2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/d_B & -1/d_B \end{bmatrix} \begin{bmatrix} d_R \\ d_L \end{bmatrix}$$

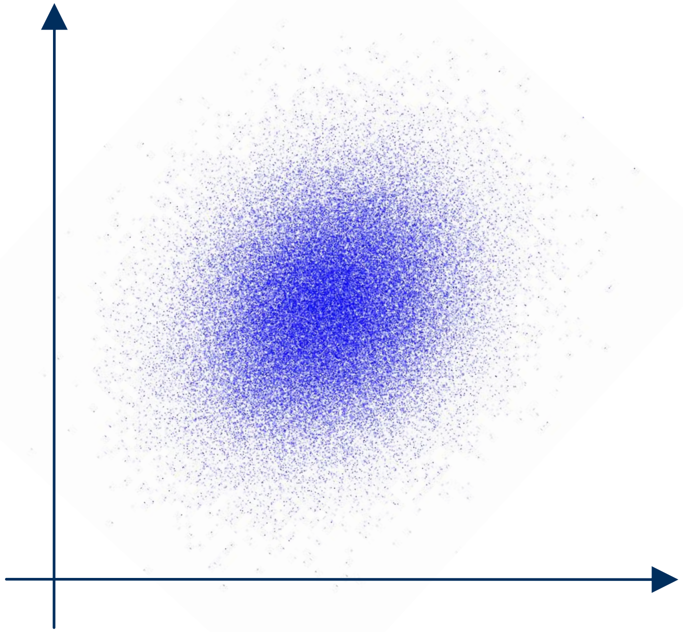
$$\mathbf{x} = BA\mathbf{w}$$

$$\Sigma_x = BA\Sigma_w (BA)^\top = BA\Sigma_w A^\top B^\top$$

Implementation Details

Estimating Meas. Uncertainty in Practice

- Where do uncertainty estimates come from?
  - Empirically measure uncertainty
  - Manufacturer data sheets
  - Educated guesses
    - Validate with χ^2 error



Sampling from Gaussians

- Sample from Gaussian y where $y \sim N(\mu_y, \sigma_y^2)$
  - Generate Gaussian noise w with $w \sim N(0,1)$
  - return
    $$y = \sigma_y w + \mu_y$$
- Sample from Gaussian $y \sim N(\mu_y, \Sigma_y)$
  - Factor $\Sigma_y = LL^T$
    - If PD, Cholesky gives a unique lower triangular L
    - If PSD, Eigen-decomposition gives a (non-unique) factorization $L=VD^{1/2}$
  - Generate Gaussian noise w with $w \sim N(0, I)$
  - return
    $$y = Lw + \mu_y$$

Uncertainty Propagation (Non-Linear Case)

Projecting Covariances (Non-linear Case)

- Again, suppose $x \sim \mu_x, \Sigma_x$

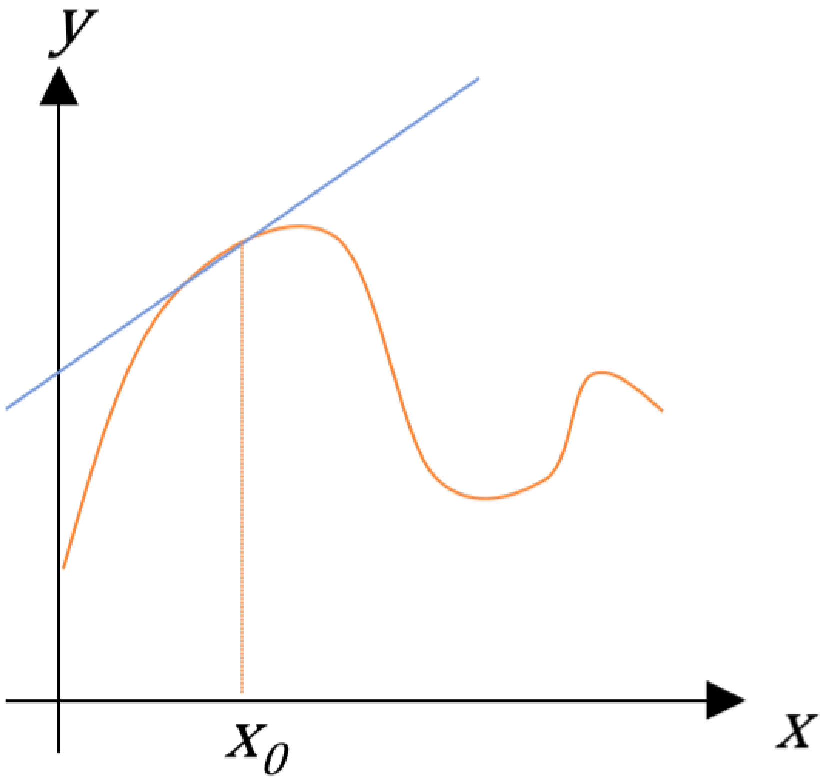
$$y = \cancel{A}x + b \qquad\qquad y = f(x)$$

- Approach: approximate f(x) with Taylor expansion
  - What point should we approximate f(x) around?

Projecting Covariances (Non-linear Case)

- First-order Taylor expansion
  - Lets review 1D case

$$y \approx \left.\frac{df}{dx}\right|_{x_0} (x - x_0) + f(x_0)$$



Projecting Covariances (Non-linear Case)

- Generalized case:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots) \\ f_2(x_1, x_2, \dots) \\ \dots \end{bmatrix}$$

$$\mathbf{y} \approx \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots \\ \dots & \dots & \dots \end{bmatrix}}_{\text{“Jacobian”}} \begin{bmatrix} x_1 - x_{1_0} \\ x_2 - x_{2_0} \\ \dots \end{bmatrix} + \begin{bmatrix} f_1(x_{1_0}, x_{2_0}) \\ f_2(x_{1_0}, x_{2_0}) \\ \dots \end{bmatrix}$$

$$\mathbf{y} \approx J|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \mathbf{f}(\mathbf{x}_0)$$

Projecting Covariances (non-linear case)

$$\mathbf{y} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{y} \approx J|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \mathbf{f}(\mathbf{x}_0)$$

$$\mathbf{y} \approx \underbrace{J|_{\mathbf{x}_0}}_{\mathbf{A}}\mathbf{x} \underbrace{- J|_{\mathbf{x}_0}\mathbf{x}_0 + \mathbf{f}(\mathbf{x}_0)}_{\mathbf{b}}$$

$$\boxed{\begin{aligned} y &= Ax + b \\ \Sigma_y &= A\Sigma_x A^T \end{aligned}}$$

Non-linear case is reduced to linear case via first-order Taylor approximation. Expansion point $\mathbf{x}_0$ is typically taken as the mean $\mu_{\mathbf{x}}$.

What do we lose by dropping higher order terms?

Projecting Covariances (Non-linear case)

- Summary:
  - In non-linear case, the projected covariance depends only on the Jacobian and the covariance of the input variables.
  - Will be computing lots of Jacobians:
    - Can do manually
    - Can do numerically