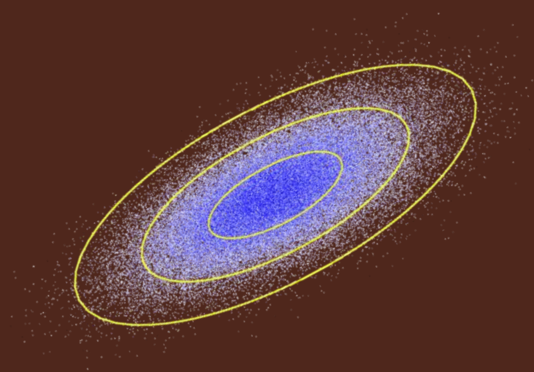
1



L12. Covariance Projection

EECS 498-6: Autonomous Robotics Laboratory

Covariance Projection

2

- Suppose we know something about random variable x:

$$x \sim N(\mu_x, \Sigma_x)$$

- And suppose I know a function y:

$$y = f(x)$$

- What is the distribution of y?
  - Let's derive μ_y, Σ_y

Mean Projection

3

- Let's start with the linear case:

$$y = f(x)$$
$$y = Ax + b$$

- What is E(y)?

$$\begin{aligned} \mu_y &= E(y) \\ &= E(Ax + b) \end{aligned}$$

- Simplify:

$$\mu_y = AE(x) + b$$

Covariance Projection

4

- Reminders:

$$\Sigma_y = E[(y - E[y])(y - E[y])^T]$$
$$\mu_y = AE(x) + b$$

$$\begin{aligned} \Sigma_y &= E[(Ax + b - A\mu_x - b)(Ax + b - A\mu_x - b)^T] \\ &= E[(Ax - A\mu_x)(Ax - A\mu_x)^T] \\ &= E[A(x - \mu_x)(x - \mu_x)^T A^T] \\ &= AE[(x - \mu_x)(x - \mu_x)]A^T \\ &= A\Sigma_x A^T \end{aligned}$$

Non-linear case

- Again, suppose:

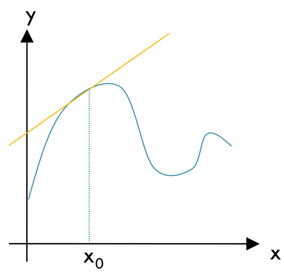
$$x \sim N(\mu_x, \Sigma_x)$$

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

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

Projecting covariances (non-linear case)

- First-order Taylor expansion
  - Let's 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:

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

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

"Jacobian"

$$\vec{y} \approx J|_{\vec{x_0}}(\vec{x} - \vec{x_0}) + f(\vec{x_0})$$

Projecting covariances (non-linear case)

$$y \approx J|_{x_0}(x - x_0) + f(x_0)$$

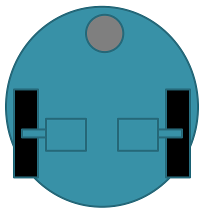
$$y \approx \underbrace{J|_{x_0}}_{A} x \underbrace{- J|_{x_0} x_0 + f(x_0)}_{b}$$

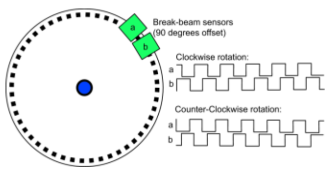
$$y = Ax + b$$
$$\Sigma_y = A \Sigma_x A^T$$

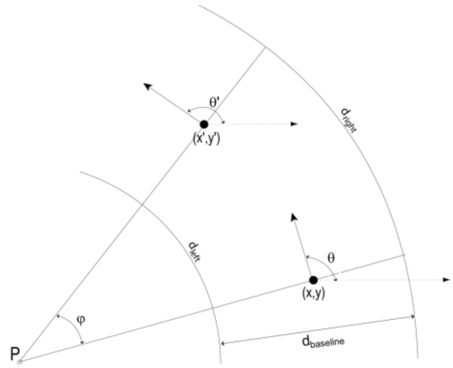
Non-linear case is reduced to linear case via first-order Taylor approximation.

What do we lose by dropping higher order terms?

Odometry Example





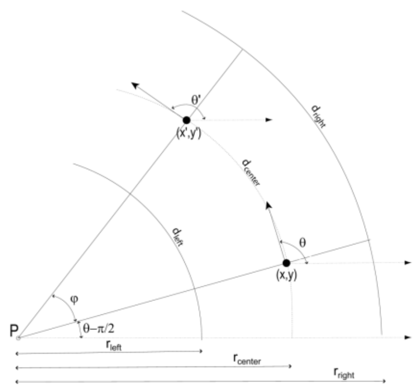


Odometry Example

- 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:

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

Odometry Example: Plan

13

- d_R, d_L are f(encoder counts)

$$\begin{aligned} \Delta x &= \frac{d_R + d_L}{2} \\ \Delta \theta &= \frac{d_R - d_L}{d_B} \end{aligned}$$

- $\Delta x, \Delta y$ are f(d_R, d_L)

- We'll project the covariances *twice*!

Odometry: First projection

- What's the uncertainty of d_R, d_L ?

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

$$d_R = \alpha c_R + w_1$$
$$d_L = \alpha c_L + w_2$$
$$\Delta x = \frac{d_R + d_L}{2}$$
$$\Delta \theta = \frac{d_R - d_L}{d_B}$$

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

But what's $\sum_w$???

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 $\sum_w$???

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

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

Odometry Example

- We are half-way there now!

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

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

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

Odometry: second projection

- 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 x in terms of d

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

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

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

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

Odometry Example

- We're done!

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

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

Sampling from Gaussians

20

- How do we generate random samples from a Gaussian distribution?

$$y \sim N(\mu_y, \Sigma_y)$$

- Idea: sample from a simpler Gaussian distribution, then project.
- We'll assume we can sample from $N(0, I)$

Sampling from Gaussians

21

- Well, suppose we know L such that:

$$y = Lw + \mu_y$$

- This would make $y \sim N(\mu_y, LL^T)$
- So, we just need to find an L such that $\Sigma_y = LL^T$
  - Cholesky decomposition

Sampling Algorithm

- Sample from Gaussian $y \sim N(\mu_y, \Sigma_y)$
  - Factor $\Sigma_y = LL^T$
  - Generate Gaussian noise w with $w \sim N(0, I)$
  - return $y = Lw + \mu_y$