Covariance Projection

- 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 $\mu_y, \Sigma_y$
Mean Projection

- Let’s start with the linear case:
  \[ y = f(x) \]
  \[ y = Ax + b \]

- What is \( E(y) \)?
  \[ \mu_y = E(y) \]
  \[ = E(Ax + b) \]

- Simplify:
  \[ \mu_y = AE(x) + b \]

Covariance Projection

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

\[ \Sigma_y = E[(Ax + b - A\mu_x - b)(Ax + b - A\mu_x - b)^T] \]
\[ = E[(Ax - A\mu_x)(Ax - A\mu_x)^T] \]
\[ = E[A(x - \mu_x)(x - \mu_x)^T A^T] \]
\[ = AE[(x - \mu_x)(x - \mu_x)]A^T \]
\[ = A\Sigma_x A^T \]
Non-linear case

- Again, suppose:
  \[ x \sim N(\mu_x, \Sigma_x) \]
  \[ y = x + b \quad y = f(x) \]

- 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:
  \[ y = \begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots
\end{bmatrix} = \begin{bmatrix}
  f_1(x_1, x_2, \ldots) \\
  f_2(x_1, x_2, \ldots) \\
  \vdots
\end{bmatrix} \]

  \[ y \approx \begin{bmatrix}
  \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \ldots \\
  \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \ldots \\
  \vdots & \vdots & \vdots
\end{bmatrix} \begin{bmatrix}
  x_1 - x_{10} \\
  x_2 - x_{20} \\
  \vdots
\end{bmatrix} + \begin{bmatrix}
  f_1(x_{10}, x_{20}) \\
  f_2(x_{10}, x_{20}) \\
  \vdots
\end{bmatrix} \]

  "Jacobian"

  \[ \bar{y} \approx J|_{\bar{x}_0} (\bar{x} - \bar{x}_0) + f(\bar{x}_0) \]

Projecting covariances (non-linear case)

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

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

\[ y = Ax + b \]

\[ \Sigma y = A \Sigma x A^T \]

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

What do we lose by dropping higher order terms?
Odometry Example

- 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

- \( d_R, d_L \) are \( f(\text{encoder counts}) \)

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

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

- We’ll project the covariances \textit{twice}!
Odometry: First projection

- What’s the uncertainty of $d_R, d_L$?

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

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

But what’s $\Sigma_w$??

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 $\Sigma_w$??

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

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

- We are half-way there now!

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

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

Odometry: second projection

- We’ve gone from \( \Sigma_w \) to \( \Sigma_d \)
- Now, we need to go from \( \Sigma_d \) to \( \Sigma_x \)

\[
\begin{align*}
d_R &= \alpha c_R + w_1 \\
d_L &= \alpha c_L + w_2 \\
\Delta x &= \frac{d_R + d_L}{2} \\
\Delta \theta &= \frac{d_R - d_L}{d_H}
\end{align*}
\]
Odometry Example

- Write $x$ in terms of $d$

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

\[
\Delta x = \frac{d_R + d_L}{2}
\]

\[
\Delta \theta = \frac{d_R - d_L}{d_B}
\]

\[
\Sigma_x = B \Sigma_d B^T
\]

Odometry Example

- We’re done!

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

\[
= \begin{bmatrix}
1/2\sigma^2 & 0 \\
0 & 2\sigma^2/d_B^2
\end{bmatrix}
\]

- Cross-correlations happen to cancel out
  - This does not happen in general!
Sampling from Gaussians

- 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

- 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 $y = Lw + \mu_y$