Lecture 8, Oct 3, 2023

Localization

- Localization is the process of determining where the robot is
  - Do we already have a map (i.e. landmarks) or do we need to build one?
  - How do we measure uncertainty arising from sensors and actuators?
  - How do we formulate the best estimate for localization from uncertain measurements?
- Any sensor measurement will invariably be corrupted by noise to some extent
  - Measurements are often distributed according to a Gaussian, due to the central limit theorem

Propagation of Error – Odometry Example

- How does uncertainty in measurements propagate?
- The covariance matrix generalizes variance to multiple dimensions
  - $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x} - \mathbb{E}(\boldsymbol{x}))(\boldsymbol{x} - \mathbb{E}(\boldsymbol{x}))^T]$
    $$= \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \cdots \\ \sigma_{x_2 x_1} & \sigma_{x_2}^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$
  - If we take the covariance between two different variables, it is known as the *cross-covariance*
  - $\operatorname{cov}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{y} - \boldsymbol{\mu}_y)^T]$
- Note some important properties of covariance:
  1. $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T$
  2. $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma} \geq 0$, i.e. the covariance matrix is semi-definite
  3. $\operatorname{cov}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{cov}(\boldsymbol{y}, \boldsymbol{x})^T$
  4. $\operatorname{cov}(\boldsymbol{x}_1 + \boldsymbol{x}_2, \boldsymbol{y}) = \operatorname{cov}(\boldsymbol{x}_1, \boldsymbol{y}) + \operatorname{cov}(\boldsymbol{x}_2, \boldsymbol{y})$, i.e. covariance is bilinear
  5. $\operatorname{cov}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{a}, \boldsymbol{B}\boldsymbol{y} + \boldsymbol{b}) = \boldsymbol{A} \operatorname{cov}(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{B}^T$
  6. $\operatorname{cov}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}$ if $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent (but a zero covariance does not mean no correlation)
- Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, then in general we can see how $\boldsymbol{\Sigma}_y$ relates to $\boldsymbol{\Sigma}_x$
  - By Taylor expansion $\boldsymbol{y} = \boldsymbol{y}_0 + (\vec{\nabla} \boldsymbol{f})_0 (\boldsymbol{x} - \boldsymbol{x}_0)$
    * Then $(\vec{\nabla} \boldsymbol{f})_0 \boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{y}_0 - (\vec{\nabla} \boldsymbol{f})_0 \boldsymbol{x}_0 = \boldsymbol{a}$
    * By property 5 above, $\boldsymbol{\Sigma}_y = (\vec{\nabla} \boldsymbol{f})_0 \boldsymbol{\Sigma}_x (\vec{\nabla} \boldsymbol{f})_0^T$
- Consider the problem of determining pose using only odometry, i.e. movement of the wheels $\Delta \boldsymbol{s} = \begin{bmatrix} \Delta s_l \\ \Delta s_r \end{bmatrix}$
  - $\Delta \boldsymbol{x} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \dfrac{\Delta s_r + \Delta s_l}{2} \cos\theta \\ \dfrac{\Delta s_r + \Delta s_l}{2} \sin\theta \\ \dfrac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} = \begin{bmatrix} \Delta s \cos\theta \\ \Delta s \sin\theta \\ \dfrac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$
- Our new position is given by $\Delta x' = \boldsymbol{f}(\boldsymbol{x} + \Delta \boldsymbol{s})$
  - Linearize: $\boldsymbol{x}' = \boldsymbol{x} + (\vec{\nabla}_{\Delta s} \boldsymbol{f})_0 \Delta \boldsymbol{s}$ where $\vec{\nabla}_{\Delta s} \boldsymbol{f}$ is the Jacobian $\begin{bmatrix} \frac{1}{2}\cos\theta & \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta & \frac{1}{2}\sin\theta \\ -b^{-1} & b^{-1} \end{bmatrix}$
  - Assume uncorrelated odometry error and $\boldsymbol{\Sigma}_\Delta = \begin{bmatrix} \sigma_{\Delta,l}^2 & 0 \\ 0 & \sigma_{\Delta,r}^2 \end{bmatrix}$
  - Error propagates as $\boldsymbol{\Sigma}_{x'} = \boldsymbol{\Sigma}_x + (\vec{\nabla}_{\Delta s} \boldsymbol{f}) \boldsymbol{\Sigma}_\Delta (\vec{\nabla}_{\Delta_s} \boldsymbol{f})^T$
  - Notice that the part we add is always positive due to the positive-semidefiniteness of the covariance, so the error always grows!
- This means using odometry alone, our estimate of where the robot is will get worse with time

1D Kalman Filtering

- If we have n measurements for a static variable x, how do we obtain the best estimate $\hat{x}$?
  - We can try to minimize $e = \sum_{k=1}^{n} w_k(\hat{x} - x_k)^2$, i.e. weighted least squares
  - The weight can be $w_k = \frac{1}{\sigma_k^2}$, so that measurements with higher variance (uncertainty) are weighted less
  - The solution is given by $\hat{x} = \frac{\sum_k \sigma_k^{-2} x_k}{\sum_k \sigma_k^{-2}}$
    * This is a weighted average of all the x_k with weights $\frac{1}{\sigma_k^2}$
- Consider the case where we have only 2 measurements, then $\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2$
  - Then the variance of $\hat{x}$ is $\operatorname{var} \hat{x} = \left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2 \sigma_1^2 + \left(\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2 \sigma_2^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$
  - But note, this is less than both σ_1^2 and σ_2^2!
- Note also $\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} x_2 = x_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(x_2 - x_1)$
  - This is a much more convenient form for us, since we've turned it from batch form (needing all measurements at once) into a recursive form (where we can continuously update)
- If we have $\hat{x}_k, \hat{\sigma}_k$ as the previous estimate at the current timestep, and we get a new measurement x_{k+1} with variance σ_{k+1}^2 then we can update:
  - $\hat{x}_{k+1} = \hat{x}_k + \frac{\hat{\sigma}_k^2}{\hat{\sigma}_k^2 + \sigma_{k+1}^2}(x_{k+1} - \hat{x}) = \hat{x}_k + W_{k+1}(x_{k+1} - \hat{x}_k)$
  - $\hat{\sigma}_{k+1} = \frac{\hat{\sigma}_k^2 \sigma_{k+1}^2}{\hat{\sigma}_k^2 + \sigma_{k+1}^2} = \hat{\sigma}_k^2 - W_{k+1}\hat{\sigma}_k^2$
  - W is known as the *Kalman gain*
  - We can see that this is similar to a feedback control law – the correction to the state is the gain multiplied by the "error"
- Kalman filtering is a special case of Bayesian filtering, where the distribution is a Gaussian
- But we still haven't accounted for the fact that x may be dynamic, i.e. it can evolve over time; to account for this, we will predict what the new state should be based on the old estimate, and then compute the error from the measurement of the new state
  - Consider the 1D state update equation $x_{k+1} = x_k + u_k + v_k$ where u_k is the control input and v_k is some noise
    * $v_k \sim \mathcal{N}(0, \varsigma_k^2)$, i.e. normally distributed, zero-mean with variance ς_k^2
    * Assume that u_k can be accurately delivered, i.e. there is no noise
  - Let $\hat{x}_{k|k}$ be the estimate of x at step k, given measurements $\{ x_0, x_1, \dots, x_k \}$
  - Let $\hat{x}_{k+1|k}$ be the prediction of x at step $k+1$, given measurements $\{ x_0, x_1, \dots, x_k \}$
    * $\hat{x}_{k+1|k} = \hat{x}_{k|k} + u_k$ (note since the noise is zero-mean, we can disregard it)
  - Now to get $\hat{x}_{k+1|k+1}$, we can use the same Kalman update formula as above
    * $\hat{\sigma}_{k+1|k}^2 = \hat{\sigma}_{k|k}^2 + \varsigma_{k+1}^2$
    * $W_{k+1} = \frac{\sigma_{k+1|k}^2}{\sigma_{k+1|k}^2 + \sigma_{k+1}^2}$
    * $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + W_{k+1}(x_{k+1} - \hat{x}_{k+1|k})$
    * $\hat{\sigma}_{k+1|k+1}^2 = \hat{\sigma}_{k+1|k}^2 - W_{k+1}\hat{\sigma}_{k+1|k}^2$
- Intuitively, Kalman filters combine an estimate and a new measurement, both of which have some uncertainty, and finds the most likely new state according to the distributions of error in both

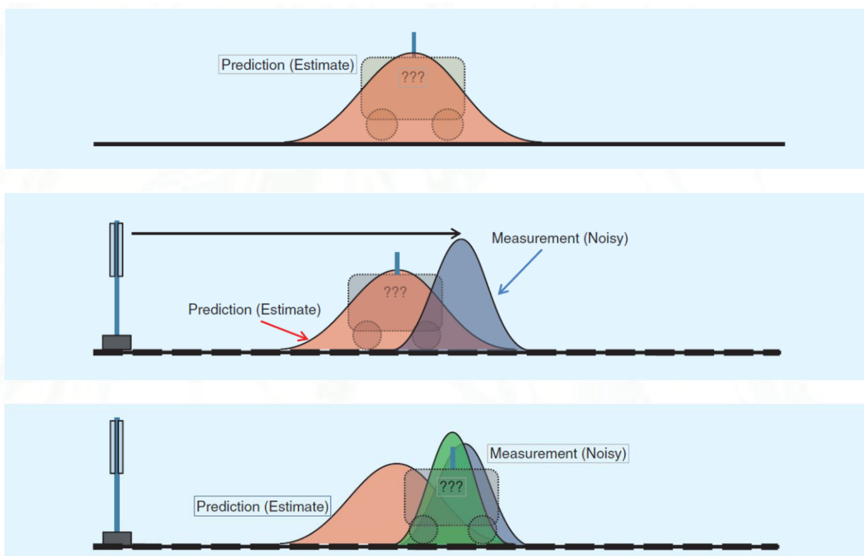

Figure 1: Diagram of Kalman filtering.