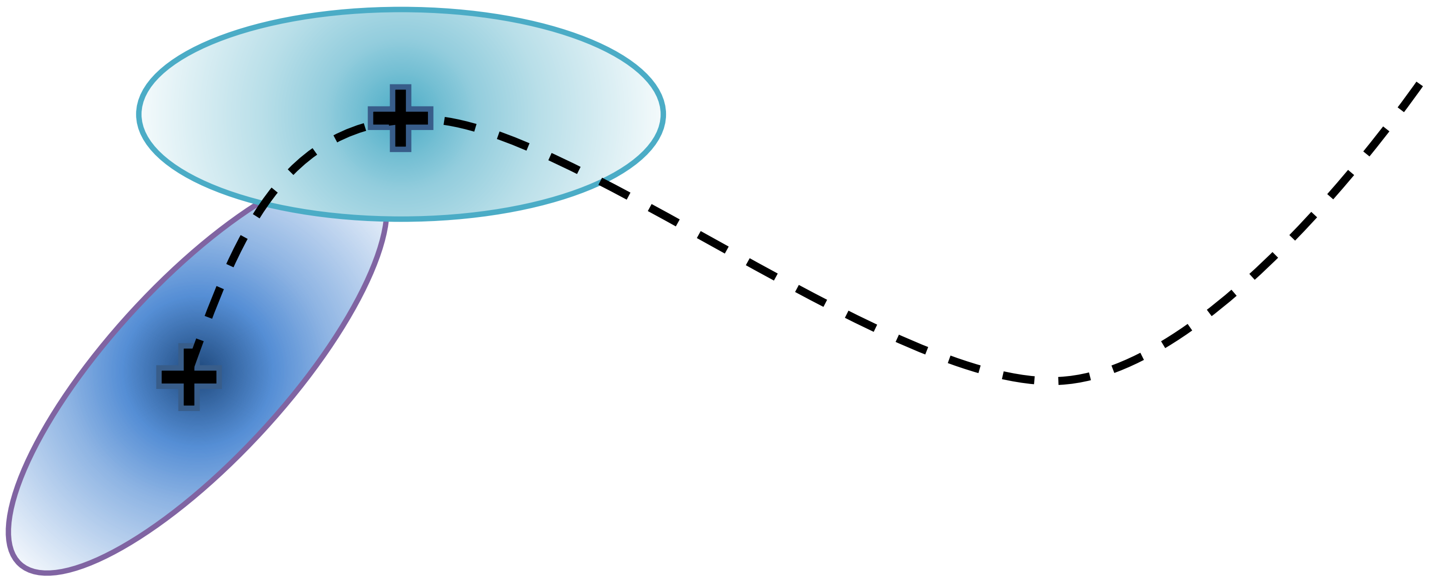
An Extended Kalman Filter for Camera/IMU Fusion

AR Track, Week 5

EKF

- Review the Kalman Filter and Extended Kalman Filter from the Estimation and Learning Course.
- For the basic EKF, we will only be using the velocity commands from your controller and the gyroscope on the IMU



Robot Model

Our state and control vectors are:

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}, \qquad u_t = \begin{pmatrix} v + n_v \\ \omega + n_\omega \end{pmatrix}$$

The unicycle model is:

$$\begin{aligned} \mu_t &= \mu_{t-1} + \delta t \begin{pmatrix} (v + n_v)\cos\theta \\ (v + n_v)\sin\theta \\ (\omega + n_\omega) \end{pmatrix} \\ &= \mu_{t-1} + \delta t \begin{pmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{pmatrix} + \delta t \begin{pmatrix} n_v\cos\theta \\ n_v\sin\theta \\ n_\omega \end{pmatrix} \end{aligned}$$

Our measurement model is:

$$z_t = x_t + n_m$$

The General EKF

We will be using slightly different notation compared to the Estimation and Learning Course:

- Prediction Step (assuming $\frac{\partial f}{\partial n} = I$):

$$\hat{\mu}_t = f(x_{t-1}, u_{t-1}, 0)$$

$$\hat{\Sigma}_t = \frac{\partial f}{\partial x} \Sigma_{t-1} \frac{\partial f}{\partial x}^T + \frac{\partial f}{\partial n} Q_t \frac{\partial f}{\partial n}^T$$

- Kalman Gain:

$$K_{t+1} = \hat{\Sigma}_{t+1} \frac{\partial h}{\partial x}^T \left(\frac{\partial h}{\partial x} \hat{\Sigma}_{t+1} \frac{\partial h}{\partial x}^T + R_t\right)^{-1}$$

- Update Step (assuming that we can measure our full state):

$$\mu_{t+1} = \hat{\mu}_{t+1} + K_{t+1}(z_t - \hat{\mu}_{t+1})$$

$$\Sigma_{t+1} = \hat{\Sigma}_{t+1} - K_{t+1} \frac{\partial h}{\partial x} \hat{\Sigma}_{t+1}$$

Robot Model

Our state and control vectors are:

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}, \qquad u_t = \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Our prediction model is:

$$f(x_{t-1}, u_{t-1}, n_{t-1}) = \mu_{t-1} + \delta t \begin{pmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{pmatrix} + \delta t \begin{pmatrix} n_v \cos\theta \\ n_v \sin\theta \\ n_\omega \end{pmatrix}$$

$$\left.\frac{\partial f}{\partial x}\right|_{(x_{t-1}, u_{t-1}, 0)} = I + \delta t \begin{pmatrix} 0 & 0 & -v\sin\theta \\ 0 & 0 & v\cos\theta \\ 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial f}{\partial n} = \delta t \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix}$$

Our measurement model is:

$$h(x_t, v_t) = x_t + v_t$$

$$\frac{\partial h}{\partial x} = I$$

Pipeline

- Start at an arbitrary location with very high uncertainty
- Every time step_filter is called, check if there are any IMU or AprilTag measurements
- Given linear velocity and IMU measurements, run the prediction step to propagate the state
- Given AprilTag measurements, run the update step to update the state
- Use the simulator to check the accuracy of your filter
  get_gt_pose()
  set_est_pose(pose)

TODO: Image of simulator

Extensions

- As an extension, you can incorporate the accelerometer measurements into the prediction step, but you will need to estimate gravity and the accelerometer bias