Nonlinear Kalman Filtering

Introduction

RECALL OUR ASSUMPTIONS…

What do we assume?

Bayes' Filter

Prior: $p(x_0)$

Transition model: $f(x_t|x_{t-1}, u_t)$

Measurement model: $g(z_t|x_t)$

Prediction step:

$$p(\bar{x}_t|z_{1:t-1}, u_{1:t}) = \int f(x_t|x_{t-1}, u_t)\, p(x_{t-1}|z_{1:t-1}, u_{1:t-1}) dx_{t-1}$$

Update step:

$$p(x_t|z_{1:t}, u_{1:t}) = \frac{g(z_t|x_t) p(x_t|z_{1:t-1}, u_{1:t})}{\int g(z_t|\, \bar{x}_t)\, p(\, \bar{x}_t|z_{1:t-1}, u_{1:t}) d\bar{x}_t}$$

What do we assume?

Kalman Filter

Prior: $p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$

Transition model:

- $x_t = A_t x_{t-1} + B_t u_t + \eta_t$
- $\eta_t \sim \mathcal{N}(0, Q_t)$

Measurement model:

- $z_t = C_t x_t + v_t$
- $v_t \sim \mathcal{N}(0, R_t)$

Prior: μ_{t-1}, Σ_{t-1}

Prediction step:

- $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
- $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$

Update step

- $K_t = \bar{\Sigma}_t C^T (C_t \bar{\Sigma} C_t^T + R)^{-1}$
- $\mu_t = \bar{\mu}_t + K_t (z_t - C \bar{\mu}_t)$
- $\Sigma_t = \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t$

What do we assume?

The prior state is best represented by a normal distribution parameterized by a mean and covariance.

The process model and measurement model are linear with additive white (normally distributed) noise.

Problems:

- Dynamic systems (and most things in life) are very rarely completely linear.
- How do we account for nonlinear processes and measurements while preserving our linear normally distributed estimate?

Nonlinearities

WHERE DOES THE KALMAN FILTER START TO BREAK DOWN?

Nonlinearities

Recall our previous two-dimensional example: 2D, planer, 2 DOF.

- Able to use the Kalman Filter due to holonomic motion example and external position measurements
- In practice, this would require knowledge of orientation or a specific holonomic drive
- That model was more applicable to target tracking rather than localization/navigation

Nonlinearities

Now consider a localization or navigation problem in a two-dimensional planer space with *three* degrees of freedom (x, y, θ).

How can we derive a global position by only measuring from the body frame?

In the body frame, what can we measure?

- Body frame accelerations and velocities (IMU, odometry)
  - v, ω
- Heading (odometry, magnetic compass)
  - θ
- Global position (GPS)
  - p_x, p_y

Nonlinearities: Propagation

System propagation equation transforming body frame velocities into global frame position:

$$\begin{bmatrix} p_x \\ p_y \\ \theta \\ v \\ \omega \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 & t\cos\theta & 0 \\ 0 & 1 & 0 & t\sin\theta & 0 \\ 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ \theta \\ v \\ \omega \end{bmatrix}_{t-1} + B_t u_t + \eta_t$$

Non-linear propagation method is simple to resolve (basic 2D kinematics), simply update the matrix every iteration as a function of the current state: $F = F(\mu_t, t)$

GPS provides a convenient way to measure global position from the body frame: $z_t = \begin{bmatrix} p_{x,z} \\ p_{y,z} \end{bmatrix}_t$

Nonlinearities: Propagation

Full system (covariance update omitted)

- $\mu_t = \begin{bmatrix} 1 & 0 & 0 & t\cos\theta & 0 \\ 0 & 1 & 0 & t\sin\theta & 0 \\ 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ \theta \\ v \\ \omega \end{bmatrix}_{t-1} + B_t u_t + \eta_t + K_t \left(\begin{bmatrix} p_{x,z} \\ p_{y,z} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \bar{\mu}_t \right)$

First problem: nonlinear propagation

- Trigonometric functions in the propagation matrix introduce nonlinearities
- Remember affine transformation properties: a normal distribution put through a nonlinear function results in a non-normal distribution!

Nonlinearities: Measurements

What if a GPS is not available?

- How do we measure a global position $(p_{x,z}, p_{y,z})$ in the body frame?
- We can measure range and bearing to a known position $\left(z_x, z_y\right)$
- Can we produce a matrix that relates the current state to a range and bearing?
- $\begin{bmatrix} l \\ \phi \end{bmatrix} = [?]x_t + v_t$
- $[?] = \begin{bmatrix} \sqrt{(z_x - p_x)^2 + \left(z_y - p_y\right)^2} \\ \operatorname{atan}\frac{z_y - p_y}{z_x - p_x} - \theta \end{bmatrix}$
- This measurement matrix cannot be applied to the linear model above

Nonlinearities

The nonlinearities in this system introduce two significant problems

- Sending any normally distributed state variable through nonlinear function violates the normally distributed assumption
- If we cannot directly measure global position, the measurement update must be a function of the current state and violates the linear system model

Taylor series to the rescue:

- Approximate our nonlinearities via a first order expansion
- Turns functions of variables into simple coefficients
- This is the basis of the Extended Kalman Filter

Linearization

TAYLOR SERIES TO THE RESCUE!

Linearization

Let's relax the strict linear model assumption and go back to Bayes and a continuous system:

- Prior: $p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
- Transition model: $\dot{x} = f(x, u, \eta)$
- Measurement model: $z = g(x, v)$

Prediction step

- Process model is nonlinear, and we need to convert continuous time dynamics to discrete time
- We will consider a finite time interval $\tau = [t', t)$ where $\delta t = t \; - t', \bar{t} = \delta t \to 0$
- Options to solve:
  - Classical integration of $f(x, u, \eta)$ over τ: $x_{\bar{t}} = \Phi(\bar{t}, x_{t-1}, u, \eta)$, difficult to do
  - Numerical one-step Euler integration, easier to do, but approximate and requires linearization

Linearization: Prediction

Linearize the dynamics about $x = \mu_{t-1}, u = u_t, \eta = 0$

$$\dot{x} \approx f(\mu_{t-1}, u_t, 0) + \left.\frac{\delta f}{\delta x}\right|_{\mu_{t-1}, u_t, 0} (x - \mu_{t-1}) + \left.\frac{\delta f}{\delta u}\right|_{\mu_{t-1}, u_t, 0} (u - u_t) + \left.\frac{\delta f}{\delta \eta}\right|_{\mu_{t-1}, u_t, 0} (\eta - 0)$$

Some substitutions:

- $A_t = \left.\frac{\delta f}{\delta x}\right|_{\mu_{t-1}, u_t, 0}$
- $B_t = \left.\frac{\delta f}{\delta u}\right|_{\mu_{t-1}, u_t, 0}$
- $U_t = \left.\frac{\delta f}{\delta \eta}\right|_{\mu_{t-1}, u_t, 0}$

Dynamics are now linear with respect to the state:

$$\dot{x} \approx f(\mu_{t-1}, u_t, 0) + A_t(x - \mu_{t-1}) + B_t(u - u_t) + U_t(\eta - 0)$$

Linearization: Prediction

One-step integration:

$$
\begin{aligned}
\bar{x}_t &\approx x_{t-1} + f(x_{t-1}, u_t, \eta_t)\delta t \\
&\approx x_{t-1} + \delta t\,(f(\mu_{t-1}, u_t, 0) + A_t(x - \mu_{t-1}) + B_t(u_t - u_t) + U_t\eta_t) \\
&\approx (I + \delta t\, A_t)x_{t-1} + \delta t(f(\mu_{t-1}, u_t, 0) - A_t\mu_{t-1}) + \delta t\, U_t\eta_t \\
&\approx F_t x_{t-1} + \delta t\,(f(\mu_{t-1}, u_t, 0) - A_t\mu_{t-1}) + V_t\eta_t
\end{aligned}
$$

We now have some components that we can work with that look like the Kalman Filter.

- $\bar{\mu}_t = \mu_{t-1} + \delta t\, f(\mu_{t-1}, u_t, 0)$
- $\bar{\Sigma}_t = F_t\Sigma_{t-1}F_t^T + V_tQ_tV_t^T$
- $F_t = I + \delta t\, A_t$

Linearization: Update

Now we need to do the same for the observation model:

$$g(x, \nu) \approx g(\bar{\mu}_t, 0) + \left.\frac{\delta g}{\delta x}\right|_{\bar{\mu}_t, 0} (x - \bar{\mu}_t) + \left.\frac{\delta g}{\delta \nu}\right|_{\bar{\mu}_t, 0} (\nu - 0)$$

Again, substituting:

- $G_t = \left.\frac{\delta g}{\delta x}\right|_{\bar{\mu}_t, 0}$
- $W_t = \left.\frac{\delta g}{\delta \nu}\right|_{\bar{\mu}_t, 0}$

This again results in a linear model with respect to the state:

$$z_t = g(x_t, \nu_t) \approx g(\bar{\mu}_t, 0) + G_t(x_t - \bar{\mu}_t) + W_t \nu_t$$

Derivation from this point forward is the same as the Kalman Filter.

The Extended Kalman Filter

A KALMAN FILTER APPROXIMATION FOR NONLINEAR SYSTEMS

The Extended Kalman Filter

Very similar to the base linear Kalman Filter:

- Prediction:
  - $\bar{\mu}_t = \mu_{t-1} + \delta t \, f(\mu_{t-1}, u_t, 0)$
  - $\bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^T + V_t Q_t V_t^T$
- Measurement update:
  - $K = \bar{\Sigma}_t G_t^T (G_t \bar{\Sigma}_t G_t^T + W_t R_t W_t^T)^{-1}$
  - $\mu_t = \bar{\mu}_t + K_t\big(z_t - g(\bar{\mu}_t, 0)\big)$
  - $\Sigma_t = \bar{\Sigma}_t - K_t G_t \bar{\Sigma}_t$
- Note that the only real changes are on the noise terms (Q_t and R_t), and how the current state maps to the measurement

The Extended Kalman Filter

Matrices are calculated via Taylor series expansion of $f(x_{t-1}, u_t, \eta_t)$ and $g(x_t, v_t)$ at each time interval to calculate the F_t and V_t, and G_t and W_t matrices, respectively.

- $F_t = I + \delta t \left.\frac{\delta f}{\delta x_t}\right|_{\bar{\mu}_t, u_t, 0}$
- $V_t = \delta t \left.\frac{\delta f}{\delta \eta_t}\right|_{\bar{\mu}_t, u_t, 0}$
- $G_t = \left.\frac{\delta g}{\delta x_t}\right|_{\bar{\mu}_t, 0}$
- $W_t = \left.\frac{\delta g}{\delta v_t}\right|_{\bar{\mu}_t, 0}$

This describes a first order Taylor series approximation EKF

- Higher order EKFs use additional terms from the Taylor series to approximate F_t and G_t
- Only beneficial with lower measurement noise

Try not to get too hung up on specific notation, understand what each variable and function is saying

- Again, some authors frequently use A_t, and F_t interchangeably
- Measurement function and matrix sometimes use h and H_t

The EKF: An Example

Let's return to our previous example:

- 2D planer motion, 3 degrees of freedom: $x_t = [p_x, p_y, \theta, v, \omega]$
- Body frame range and bearing measurement to a known landmark: $z_t = [r, \phi]$

No control inputs, noise is only on the velocity terms.

$$\dot{x}_t = f(x_t, u_t, \eta_t) = \begin{bmatrix} 0 & 0 & 0 & t\cos\theta & 0 \\ 0 & 0 & 0 & t\sin\theta & 0 \\ 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ \theta \\ v \\ \omega \end{bmatrix}_{t-1} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathcal{N}(0, Q_t) = \begin{bmatrix} vt\cos\theta \\ vt\sin\theta \\ \omega t \\ \eta_v \\ \eta_\omega \end{bmatrix}$$

Measurement returns range and bearing to a known landmark with position (z_x, z_y)

$$g(\bar{\mu}_t, v_t) = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(z_x - \bar{p}_x)^2 + (z_y - \bar{p}_y)^2} + v_{t,r} \\ \operatorname{atan} \dfrac{z_y - \bar{p}_y}{z_x - \bar{p}_x} - \theta + v_{t,\phi} \end{bmatrix}$$

The EKF: An Example

Linearization:

- $\dot{x} \approx f(\mu_{t-1}, u_t, 0) + \left.\frac{\delta f}{\delta x}\right|_{\mu_{t-1},u_t,0} (x - \mu_{t-1}) + \left.\frac{\delta f}{\delta u}\right|_{\mu_{t-1},u_t,0} (u - u_t) + \left.\frac{\delta f}{\delta \eta}\right|_{\mu_{t-1},u_t,0} (\eta - 0)$

- $A_t = \left.\frac{\delta f}{\delta x}\right|_{\mu_{t-1},u_t,0} = \begin{bmatrix} \frac{\delta p_x}{\delta p_x} & \frac{\delta p_x}{\delta p_y} & \frac{\delta p_x}{\delta \theta} & \frac{\delta p_x}{\delta v} & \frac{\delta p_x}{\delta \omega} \\ \frac{\delta p_y}{\delta p_x} & \frac{\delta p_y}{\delta p_y} & \frac{\delta p_y}{\delta \theta} & \frac{\delta p_y}{\delta v} & \frac{\delta p_y}{\delta \omega} \\ \frac{\delta \theta}{\delta p_x} & \frac{\delta \theta}{\delta p_y} & \frac{\delta \theta}{\delta \theta} & \frac{\delta \theta}{\delta v} & \frac{\delta \theta}{\delta \omega} \\ \frac{\delta v}{\delta p_x} & \frac{\delta v}{\delta p_y} & \frac{\delta v}{\delta \theta} & \frac{\delta v}{\delta v} & \frac{\delta v}{\delta \omega} \\ \frac{\delta \omega}{\delta p_x} & \frac{\delta \omega}{\delta p_y} & \frac{\delta \omega}{\delta \theta} & \frac{\delta \omega}{\delta v} & \frac{\delta \omega}{\delta \omega} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_x \sin\theta & \cos\theta & 0 \\ 0 & 0 & v_y \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

The EKF: An Example

Linearization continued:

- $B_t = \left.\frac{\delta f}{\delta u}\right|_{\mu_{t-1}, u_t, 0} = [0]$
- $U_t = \left.\frac{\delta f}{\delta \eta}\right|_{\mu_{t-1}, u_t, 0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

The EKF: An Example

One-step Euler integration:

- $$F_t = I + \delta t \left.\frac{\delta f}{\delta x_t}\right|_{\bar{\mu}_t,\, u_t, 0} = \begin{bmatrix} 1 & 0 & -v\,\delta t \sin\theta & \cos\theta & 0 \\ 0 & 1 & v\,\delta t \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 & 0 & \delta t^2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- $$V_t = \delta t \left.\frac{\delta f}{\delta \eta_t}\right|_{\bar{\mu}_t,\, u_t, 0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & \delta t \end{bmatrix}$$

Note the change on the F_{25} term: continuous time is in t discrete time is in δt.

$$F_{25} = \delta t \ * t = \delta t^2$$

The EKF: An Example

Now the measurement matrices:

$$g(\bar{\mu}_t, \nu_t) = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(z_x - \bar{p}_x)^2 + \left(z_y - \bar{p}_y\right)^2} + \nu_{t,r} \\ \operatorname{atan} \dfrac{z_y - \bar{p}_y}{z_x - \bar{p}_x} - \theta + \nu_{t,\phi} \end{bmatrix}$$

$$g(x, \nu) \approx g(\bar{\mu}_t, 0) + \left.\frac{\delta g}{\delta x}\right|_{\bar{\mu}_t, 0} (x - \bar{\mu}_t) + \left.\frac{\delta g}{\delta \nu}\right|_{\bar{\mu}_t, 0} (\nu - 0)$$

The EKF: An Example

$$G_t = \frac{\delta g}{\delta x_t}\bigg|_{\bar{\mu}_t} = \begin{bmatrix} \frac{\delta r}{\delta p_x} & \frac{\delta r}{\delta p_y} & \frac{\delta r}{\delta \theta} & \frac{\delta r}{\delta v} & \frac{\delta r}{\delta \omega} \\ \frac{\delta \phi}{\delta p_x} & \frac{\delta \phi}{\delta p_y} & \frac{\delta \phi}{\delta \theta} & \frac{\delta \phi}{\delta v} & \frac{\delta \phi}{\delta \omega} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{z_x - p_x}{\sqrt{(z_x-p_x)^2 + (z_y+p_y)^2}} & -\frac{z_y - p_y}{\sqrt{(z_x-p_x)^2 + (z_y+p_y)^2}} & 0 & 0 & 0 \\ \frac{z_y - p_y}{(z_x-p_x)^2 + (z_y+p_y)^2} & \frac{p_x - z_x}{(z_x-p_x)^2 + (z_y+p_y)^2} & -1 & 0 & 0 \end{bmatrix}$$

$$W_t = \frac{\delta g}{\delta \nu}\bigg|_{\bar{\mu}_t,0} = \begin{bmatrix} \frac{\delta r}{\delta \nu_{t,r}} & \frac{\delta r}{\delta \nu_{t,\phi}} \\ \frac{\delta \phi}{\delta \nu_{t,r}} & \frac{\delta \phi}{\delta \nu_{t,\phi}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The EKF: An Example

Summary of matrices:

$$F_t = \begin{bmatrix} 1 & 0 & -v\,\delta t \sin\theta & \cos\theta & 0 \\ 0 & 1 & v\,\delta t\cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_t = [0]$$

$$U_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \eta_v & 0 \\ 0 & 0 & 0 & 0 & \eta_\omega \end{bmatrix}$$

$$V_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & \delta t \end{bmatrix}, \qquad \mathrm{R_t} = \begin{bmatrix} \nu_{t,r} & 0 \\ 0 & \nu_{t,\phi} \end{bmatrix}$$

$$G_t = \begin{bmatrix} -\dfrac{z_x - p_x}{\sqrt{(z_x-p_x)^2 + \left(z_y+p_y\right)^2}} & -\dfrac{z_y - p_y}{\sqrt{(z_x-p_x)^2 + \left(z_y+p_y\right)^2}} & 0 & 0 & 0 \\ \dfrac{z_y - p_y}{(z_x-p_x)^2 + \left(z_y+p_y\right)^2} & \dfrac{p_x - z_x}{(z_x-p_x)^2 + \left(z_y+p_y\right)^2} & -1 & 0 & 0 \end{bmatrix}$$

$$W_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

F_t and G_t need to be calculated at each iteration. If your system's noise parameters are constant, then U_t, Q_t, W_t, and R_t are constants.

The Extended Kalman Filter: Summary

Assumptions
- Normal distribution assumption is relaxed, but still relies on it being a reasonably accurate approximation
- Differentiability of the system model and measurement model (at least first order)
- Small perturbations

Strengths
- Simple, has the strengths of the Kalman Filter with added flexibility
- Works with generic models

Limitations and weaknesses
- Limited to systems with lower-order nonlinearities
- Requires repeated calculation of Jacobian (partial derivatives) matrices, more problematic the higher dimensions you have
- Still limited to unimodal distributions

The Extended Kalman Filter: Summary

So, we can now do Kalman Filtering on non-linear systems, but we added to the computational complexity and still can’t handle higher order nonlinearities very well.

That kind of stinks…

The Unscented Kalman Filter

BECAUSE THE EKF KIND OF STINKS

Assumptions

We start from the same point as the EKF.

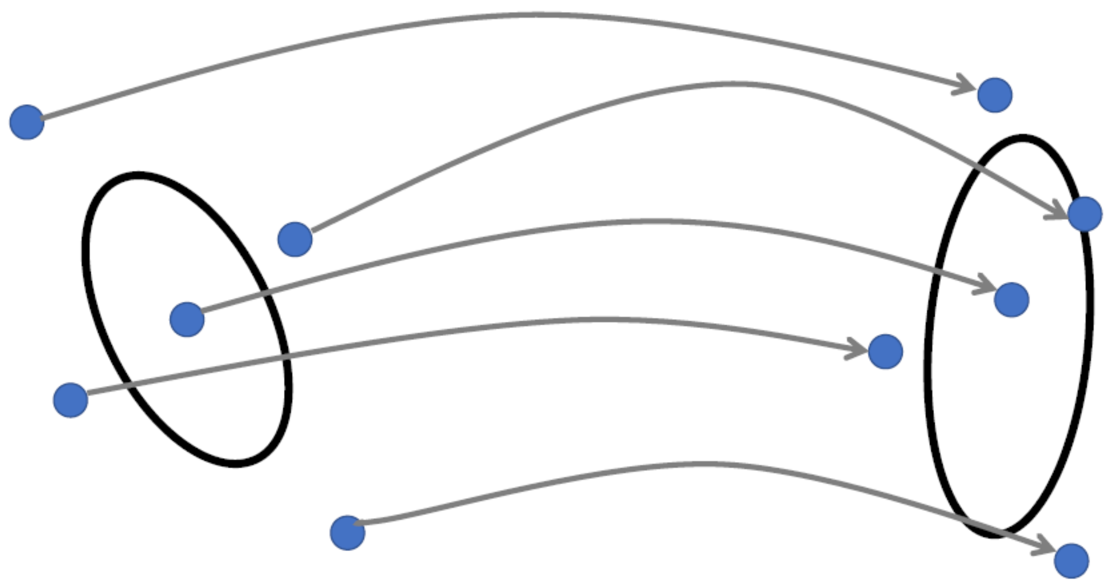
- Prior: $p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$
- Continuous time generic (nonlinear) process model with normal white noise (η): $\dot{x} = f(x, u, \eta)$
- Observation model is a generic (nonlinear) function with normal white noise (ν): $z = g(x, \nu)$

We now include something called the unscented transformation.

- This is an alternative linearization approach to the Taylor series used in the EKF
- Performs a stochastic linearization using a weighted statistical linear regression
- Essentially, picks a representative number of points, propagates them, and attempts to reconstruct a normal distribution after propagation

Named ‘unscented’ because the authors thought the EKF stunk.

The unscented transform



The unscented transform

INPUT

Normal distribution parameters

- Mean
- Covariance
- Dimension of the system (d)

OUTPUT

Sigma points

- Discrete selections of probability distribution
- Number of points: $2d + 1$
- Each point is weighted which corresponds to how they are used to reconstruct the distribution

The unscented transform

The sigma points are a column-wise array corresponding to:

$$\mathcal{X}_t = \left[\mu_t, \mu_t \pm \sqrt{(d+\lambda)\Sigma_t}\right]$$

Alternatively, as a list:

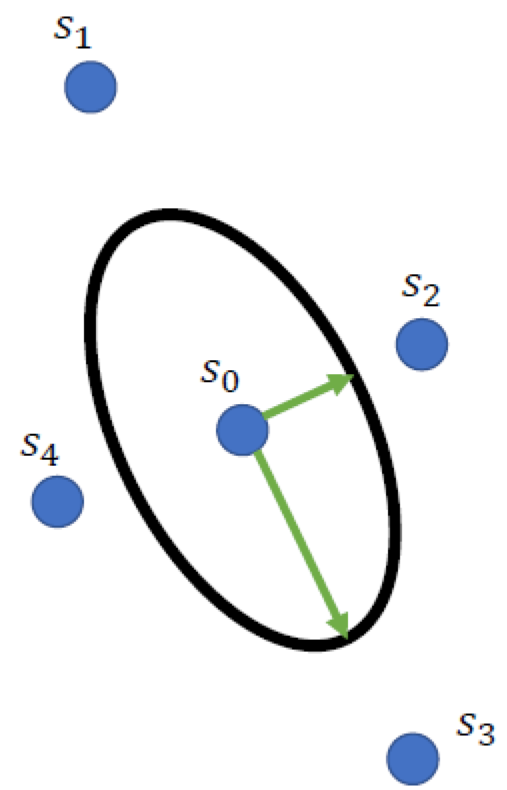
$$\mathcal{X}_t^{[0]} = \mu_t$$

$$\mathcal{X}_t^{[i]} = \mu + \sqrt{(d+\lambda)\Sigma_\mathrm{t}} \quad \text{for } i = 1, \dots, d$$

$$\mathcal{X}_t^{[i]} = \mu - \sqrt{(d+\lambda)\Sigma_\mathrm{t}} \quad \text{for } i = d+1, \dots, 2d$$

Where λ is a scaling parameter corresponding to the scaling parameters α and κ:

$$\lambda = \alpha^2(d+\kappa) - d$$



The unscented transform

Each sigma point has a corresponding weight.

- w_c is used to reconstruct the covariance
- w_m is used to calculate the mean

$$w_m^{[0]} = \frac{\lambda}{d + \lambda}$$

$$w_c^{[0]} = w_m^{[0]} + 1 - \alpha^2 + \beta$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(d + \lambda)} \quad \text{for } i = 1, \ldots, 2d$$

The unscented transform

Sigma point parameters

- α and κ
  - Control the spread of the points
  - Values are dependent on the scenario being modeled
  - Typical choice is $\alpha = 10^{-3}$ and $\kappa = 1$
- β
  - Factor relating to the distribution
  - If the distribution is truly normal, $\beta = 2$ is optimal
  - Reinforces the weight of the sigma point corresponding to the mean

The Unscented Kalman Filter

With the unscented transform used to linearize, the system the becomes relatively simple despite the somewhat complex notation.

Prediction step

- Calculate sigma points from prior estimate $(\mu_{t-1}, \Sigma_{t-1})$
- Propagate mean and covariance using a weighted sum $(\bar{\mu}_t, \bar{\Sigma}_t)$

Update step

- Calculate propagated sigma points from propagated mean and covariance
- Transform the sigma points via the measurement model and calculate the anticipated measurement
- Calculate the Kalman gain from the measurement covariance and cross-covariance
- Compute the updated mean and covariance

It ends up still being the same five lines as the EKF

- Take $\mathcal{X}_t(\dots)$ to be the function that calculates the sigma points

The Unscented Kalman Filter

Prediction:

- $\bar{\mathcal{X}}_t = f(u_t, \mathcal{X}(\mu_{t-1}, \Sigma_{t-1}), 0)$
- $\bar{\mu}_t = \sum_{i=0}^{2d} w_m^{[i]} \bar{\mathcal{X}}_t$
- $\bar{\Sigma}_t = \sum_{i=0}^{2d} w_c^{[i]} \left(\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t\right)\left(\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t\right)^T + Q_t$

Update:

- $\bar{\mathcal{Z}}_t = g\left(\mathcal{X}(\bar{\mu}_t, \bar{\Sigma}_t)\right)$
- $\hat{z}_t = \sum_{i=0}^{2d} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$
- $S_t = \sum_{i=0}^{2d} w_c^{[i]} \left(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t\right)\left(\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t\right)^T + R_t$
- $\hat{\Sigma}_t = \sum_{i=0}^{2d} w_c^{[i]} \left(\mathcal{X}_t^{[i]} - \bar{\mu}_t\right)\left(\mathcal{Z}_t^{[i]} - \hat{z}_t\right)^T$
- $K_t = \hat{\Sigma}_t S_t^{-1}$
- $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$
- $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$

The Unscented Kalman Filter: Summary

Assumptions

- Normal distribution assumption is relaxed, but still relies on it being a reasonably accurate approximation

Strengths

- Simple, has the strengths of the Kalman Filter with added flexibility
- Works with generic models
- Very computationally efficient, sums can be vectorized, and works well even for higher dimensional systems

Limitations and weaknesses

- Limited to systems with lower-order nonlinearities, but is more flexible than the EKF
- Still limited to unimodal distributions

The EKF vs. UKF

The EKF is more frequently used for nonlinear systems.

- The Kalman Filter is old (original research published from 1959-1961) and the EKF (1970’s and 1980’s) is a straightforward step forward from there using old and well-established linearization techniques (Taylor Series: 1715).
- More frequently encountered in industry and the literature simply due to the fact that it is well understood and people have practice with it
- Nonetheless it has notable shortcomings, is difficult to implement and tune, and reliable only in certain well constrained systems (Julier, S.J.; Uhlmann, J.K. (2004). "Unscented filtering and nonlinear estimation")

The UKF is relatively new (1997-2004) and not widely adopted

- In my opinion, while the notation is a bit intimidating, the UKF ends up being easier to implement
- Has been found to be more accurate than the EKF, but EKFs that use higher-order series-expansions have been shown to recover that accuracy.

Maps and Sensor Models

HOW WE MODEL THE REAL WORLD

A word on models in general

In a probabilistic sense of the world, we need to develop models that create a *likely* sense of the world.

The nonlinear techniques typically don't care what the model is, so long as it transforms (returns) the appropriate data.

Frequently, but not always, we make our models with zero-mean (white) Gaussian noise, but that is not always the correct probability distribution.

Measurement models

Until now we haven’t really discussed how measurements are formed

- Assuming we can get some measurement and relate it back to our state
- No analysis of measurement models as a probability distribution

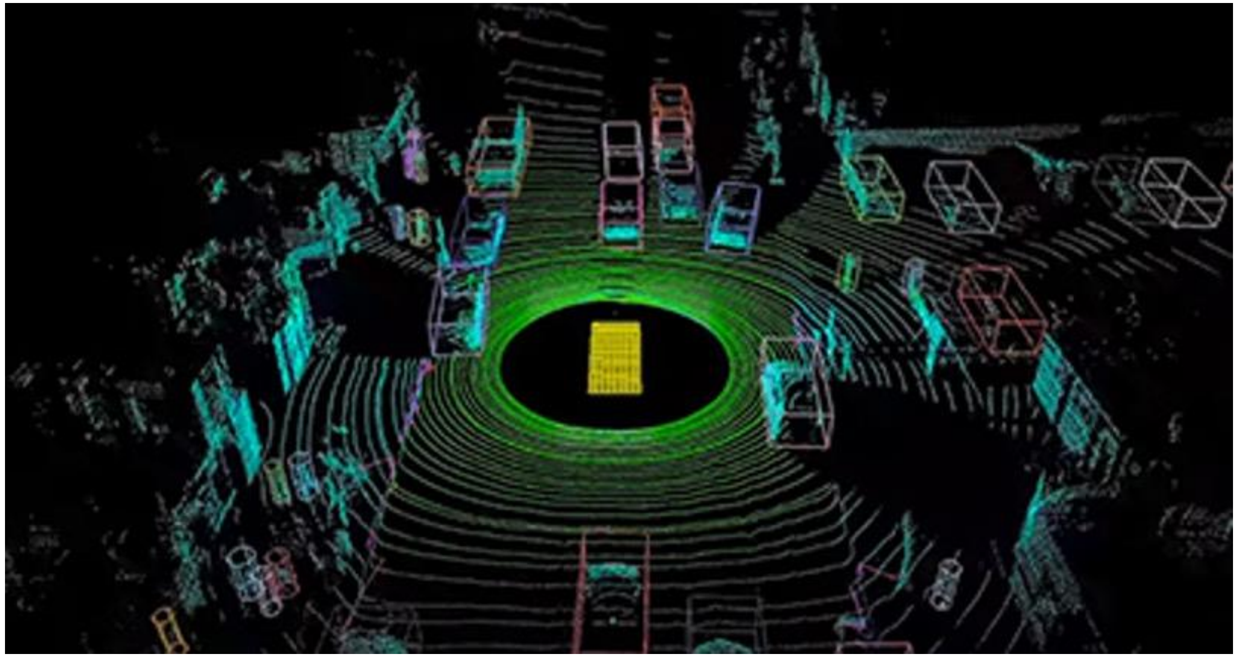
Measurements often have multiple components

- Range and bearing
- Multiple laser scans
- Need to understand how our sensors work in order to accurately model them

Measurement models: Lidar, an example

Lidar (light detecting and ranging) is an extremely common and very useful tool in robotics.

- Consists of one or more lasers, a rotating mirror, and receiver
- Typically generates a point cloud of data



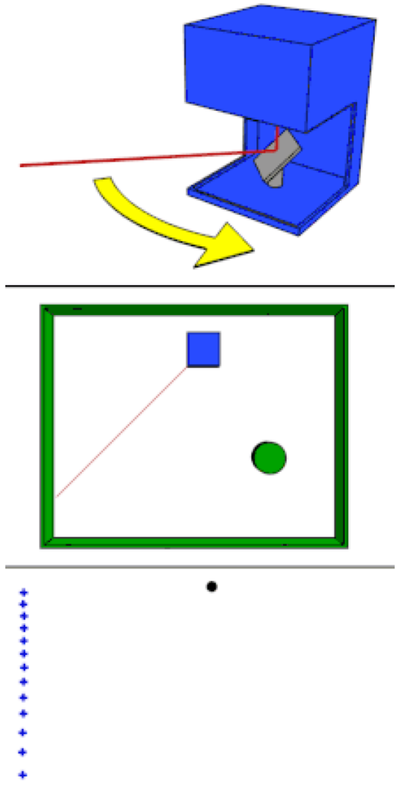
2D Lidar

Let’s simplify the problem to two dimensions

- Lidar has k beams: $z_t = \{z_t^1, \ldots, z_t^k\}$ that return a distance value
- Measurements have conditional independence

Measurements have some degree of uncertainty due to various sources

- General noise
- Unexpected measurements
- Failure
- Randomness



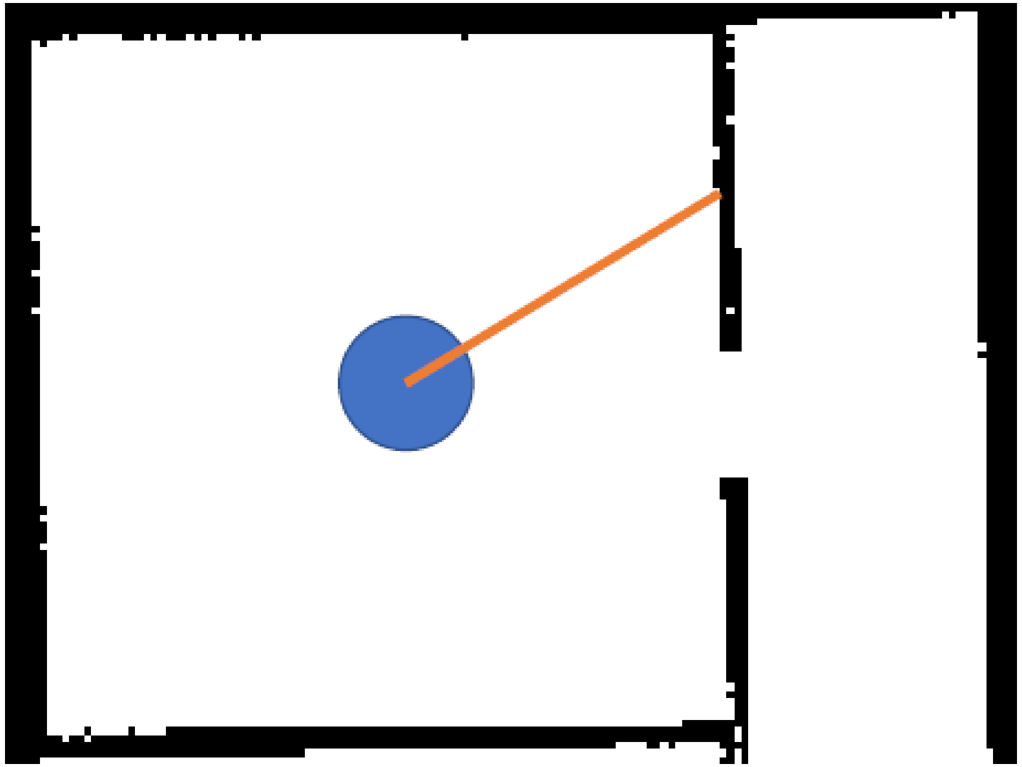
Measurement Noise

The sensor detects the correct object, but not precisely the correct distance

- We can calculate the expected value $z_t^{k^*}$ using the current state and the sensor's noise parameters (σ_{hit}^2) and the map m

$$p_{hit}(z_t^k | x_t, m) = \begin{cases} \gamma \mathcal{N}(z_t^k | z_t^{k^*}, \sigma_{hit}^2) & \text{if } 0 \le z \le z_{max} \\ 0 & \text{otherwise} \end{cases}$$

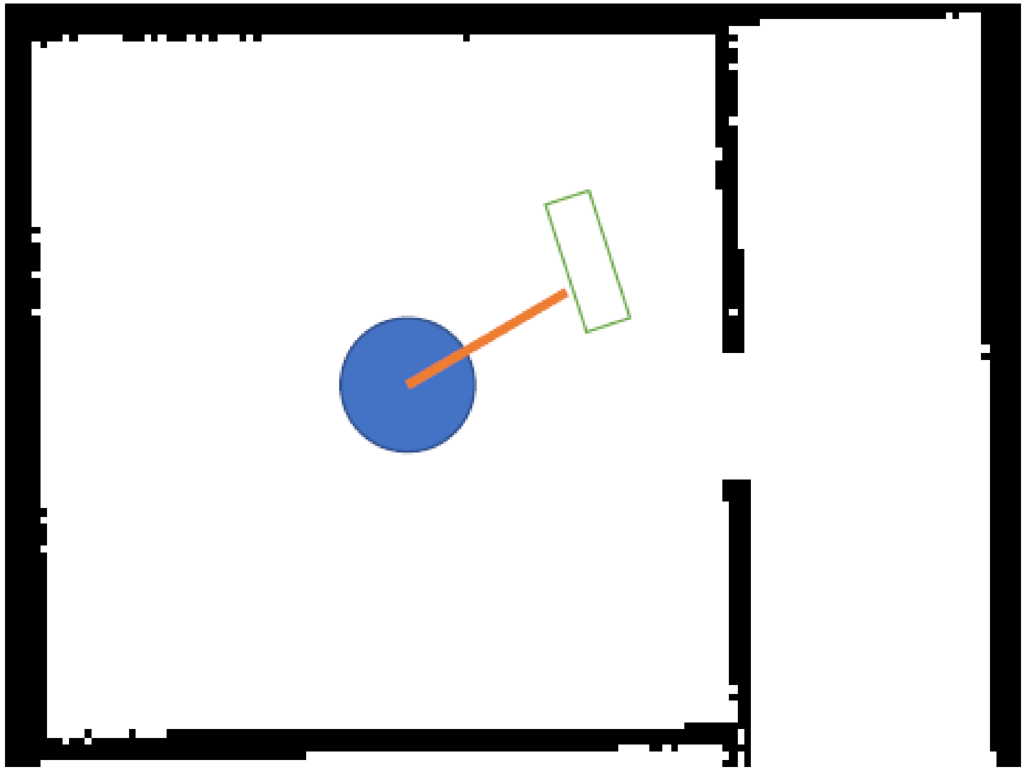
- σ_{hit}^2 would be determined as part of the manufacturing and quality assurance process



Unexpected Objects

A true positive detection is made of an unmapped object between the robot and an expected object

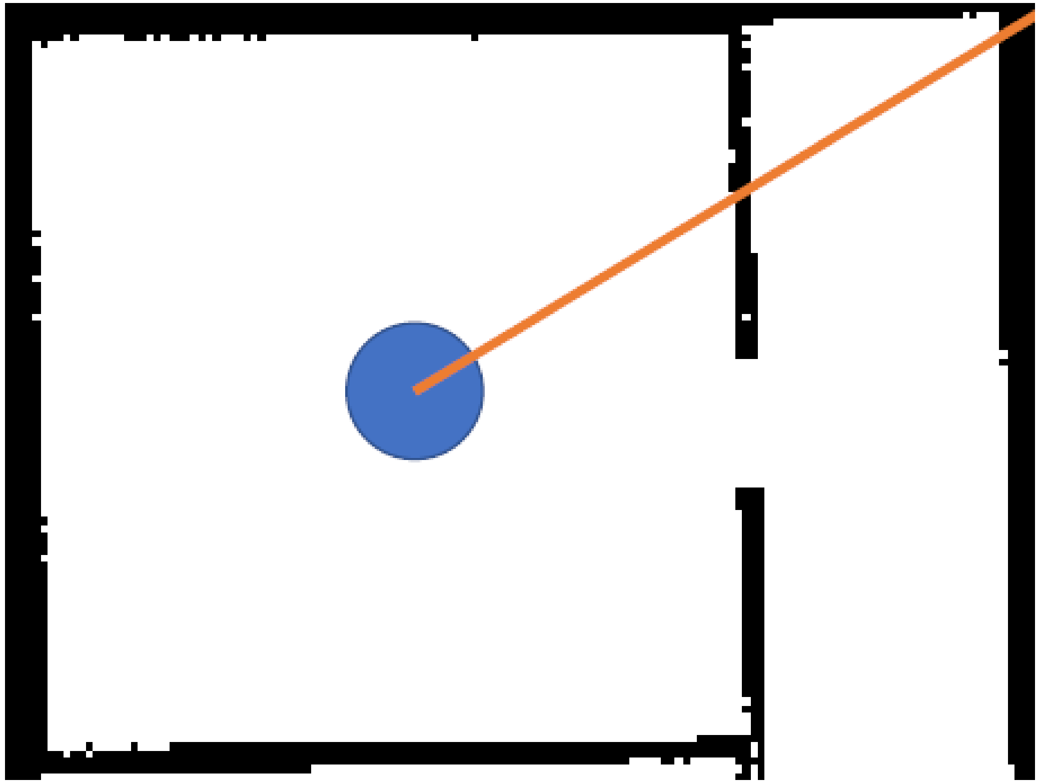
$$p_{short}(z_t^k | x_t, m) = \begin{cases} \gamma\lambda e^{-\lambda z_t^k} & \text{if } 0 \leq z_t^k \leq z_t^{k^*} \\ 0 & \text{otherwise} \end{cases}$$



Sensor failures

Sensor does not get a return signal or returns a max value.

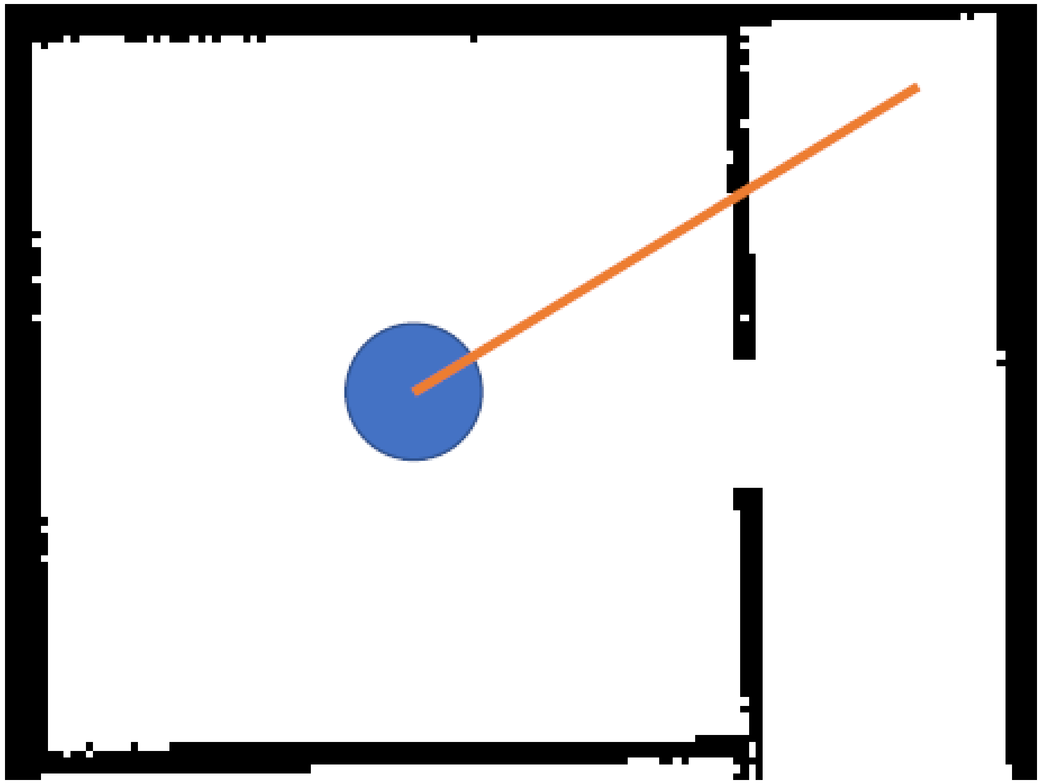
$$p_{fail}(z_t^k | x_t, m) = \begin{cases} 1 & \text{if } z_t^k = z_{max} \\ 0 & \text{otherwise} \end{cases}$$



Random Measurements

Occasionally, measurements simply cannot be explained.

$$p_{random}(z_t^k | x_t, m) = \begin{cases} \dfrac{1}{z_{max}} & if\ 0 \leq z_t^k < z_{max} \\ 0 & otherwise \end{cases}$$



Lidar Sensor Model: Beam Model

We can construct a comprehensive sensor model for a lidar sensor by taking the weighted sum of the previous four components

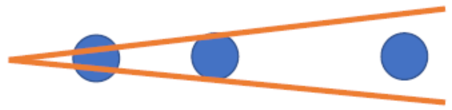
$$p(z_t^k|x_t, m) = w_{hit}p_{hit}(z_t^k|x_t) + w_{short}p_{short}(z_t^k|x_t)$$
$$+w_{fail}p_{fail}(z_t^k|x_t) + w_{random}p_{random}(z_t^k|x_t)$$

Advantages:

- Based on the underlying physics of the sensors
- Realistic representation, if results don't correspond you likely have a faulty sensor

Drawbacks

- Computational complexity, combination of four different conditional probability distributions
- Sensitive and susceptible to geometry



Likelihood Field Model

Changes the probability distribution for a hit to a normal distribution with the failure and random measurements staying the same

$$p_{hit}\left(z_t^k \middle| x_t, m\right) = \mathcal{N}\left(d^2, 0, \sigma_{hit}^2\right)$$

Where d is the distance from the beam return to the nearest object and p_{short} is discarded.

Advantages

- Smoothness – small changes in pose result in small changes in likelihood
- Efficiency – can precompute distances to objects

Disadvantages

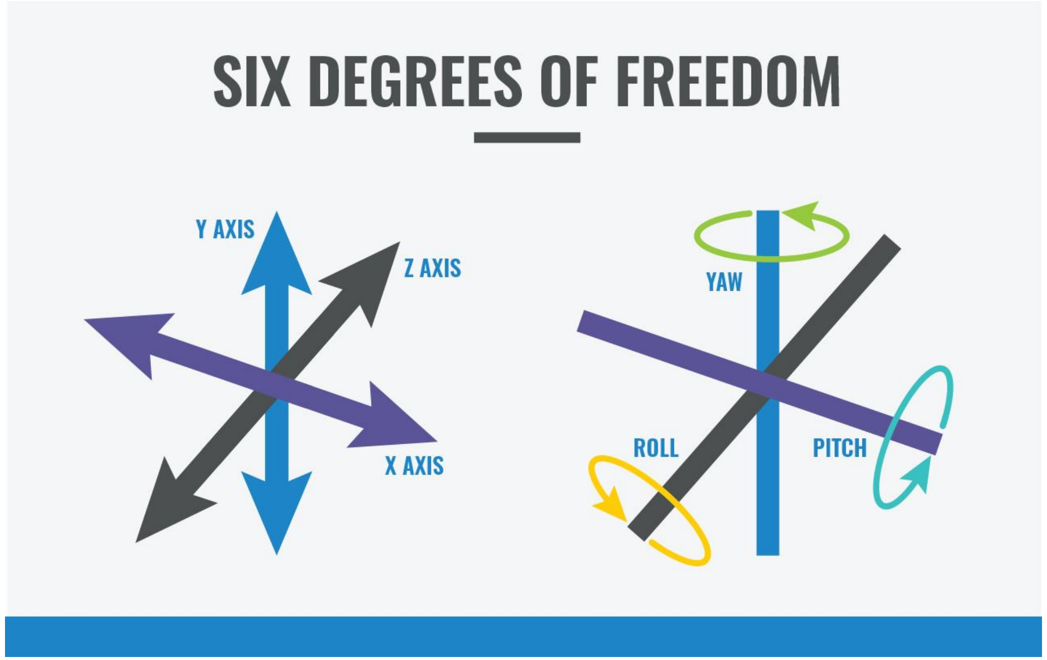
- Not based on the physics

Inertial measurement units

Alternatively, we also have sensors that attempt to measure and compute internal states.

- Inertial measurement units (IMU) are a common sensor that does this
- Measures specific forces (accelerations) with accelerometers and angular rate (angular velocity) with gyroscopes
- Depending on the specific configuration, might report angular and linear velocities

Notoriously noisy and error prone sensors and a large body of work for how to integrate these sensors into navigation systems (we'll get to that later in the course!)



Gyroscope model

Gyroscopes measures angular velocity ($\widehat{\omega}$) which is used to update the orientation (θ).

$$\theta_t = \theta_{t-1} + \widehat{\omega} t$$

$\widehat{\omega}$ is the output from the IMU where the true angular velocity is ω and is the sum of the true angular velocity, a bias term, and white noise.

$$\widehat{\omega} = \omega + b + \eta$$

Accelerometer model

Accelerometers measure specific force – not actual accelerations – and we need orientations to calculate the relative body-frame accelerations.

- Gravity is a constant acceleration on the body frame
- Orientation determines along which axis
- In small-scale settings we can typically initially align the z-axis of the vehicle to be parallel to the gravity vector
- Orientation representation allows us to transform the gravity vector and remove it from the specific forces
- Second half of this course covers what we do when we can't assume this

Accelerometers have a bias like gyroscopes but also can be misaligned and have scalar errors.

$$\hat{a} = (I + M)a + b + \eta$$

- M matrix is the scalar and cross-coupling matrix that accounts for scalar errors and misalignments
- Diagonal elements are scalar; off-diagonal are the cross-coupling errors
- Cross-coupling is typically and ideally zero: the IMU is properly aligned

Bias values tend to dominate in both accelerometers and gyroscopes.

Maps

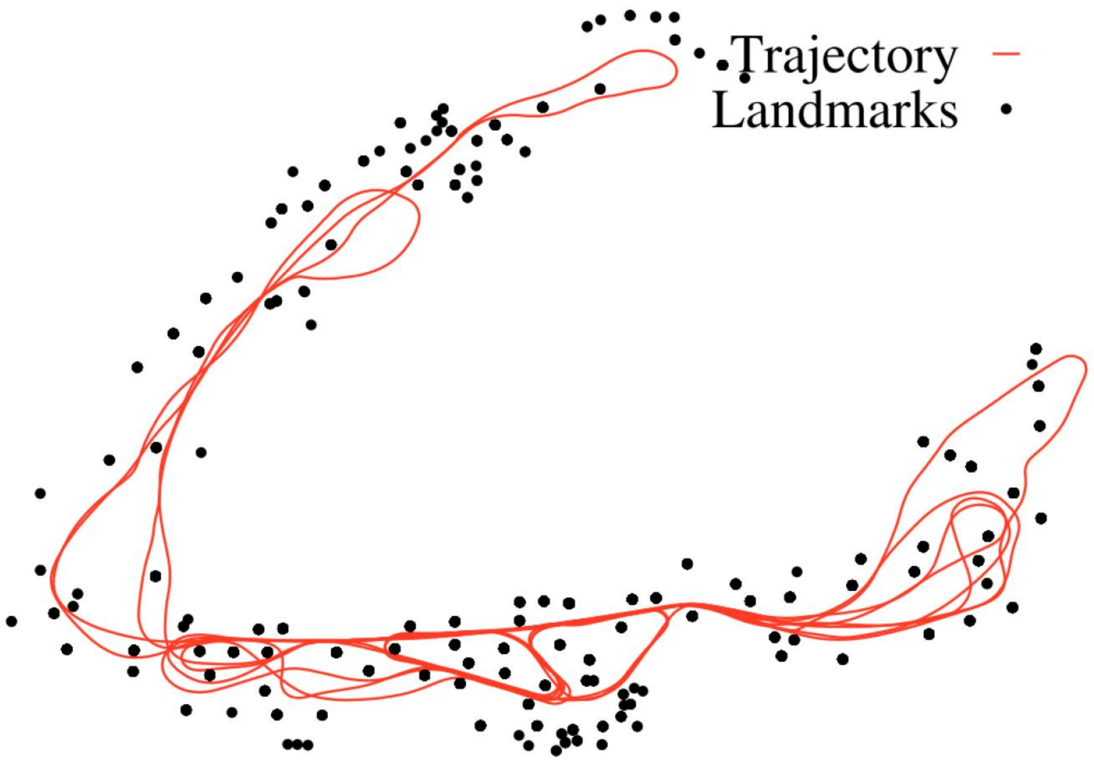
A map is a list of objects with properties as it relates to physical space and can be feature-based or location based

Feature based maps have elements m_i that represent an object in the map that have a set of properties

- Location
- Classification/Label
- Scalar quantity

Location based maps have elements m_i that correspond to a location in space with typically a single value.

Feature-based map



Location-based Maps

OCCUPANCY GRID

TERRAIN RELIEF / CONTOUR