Kalman Filters for Mapping and Localization
 Autonomous Aerial Navigation in Confined Indoor Environments

Shaojie Shen, Nathan Michael, Vijay Kumar

University of Pennsylvania
RGBD SLAM
with
ROS + Kinect
Sensors

If you can’t model the world, then sensors are the robot’s link to the external world (obsession with depth)
Sensors

Robots’ link to the external world...

Sensors, sensors, sensors!
and tracking what is sensed: world models

Gyroscope
(orientation)

Force/
Torque

Inertial measurement unit
(gyro + accelerometer)

GPS

Compass
Infrared sensors

“Noncontact bump sensor”

IR emitter/detector pair

(1) sensing is based on light intensity.

“object-sensing” IR

looks for changes at this distance

diffuse distance-sensing IR

(2) sensing is based on angle received.
Infrared Calibration

The response to white copy paper (a dull, reflective surface)

raw values (put into 4 bits)

15° increments

in the dark

fluorescent light

incandescent light
Infrared Calibration

energy vs. distance for various materials
( the incident angle is 0º, or head-on )
( with no ambient light )
Sonar sensing

single-transducer sonar timeline

0

75µs

a “chirp” is emitted into the environment
typically when reverberations from the initial chirp have stopped

the transducer goes into “receiving” mode and awaits a signal...

limiting range sensing

.5s

after a short time, the signal will be too weak to be detected

Polaroid sonar emitter/receivers

No lower range limit for paired sonars...
**Sonar effects**

(a) Sonar providing an accurate range measurement

(b-c) Lateral resolution is not very precise; the closest object in the beam’s cone provides the response

(d) Specular reflections cause walls to disappear

(e) Open corners produce a weak spherical wavefront

(f) Closed corners measure to the corner itself because of multiple reflections --> sonar ray tracing
Sonar modeling

- Initial time response
- Accumulated responses
- Blanking time
- Cone width
- Spatial response

Note: dB normalized to on-axis response
Laser Ranging

LIDAR/Laser range finder

LIDAR map
Souped-Up Robots
Recent, Cool...

The 3d “time-of-flight” camera

(1) “RF-modulated optical radiation field output”

(2) Reflects off the environment

(3) Time-interval input measurements provides data (not time-integrated!)

How can we get an image from this information?
More recent, Cooler...

Structured light:
Project a known dot pattern with an IR transmitter (invisible to humans)

Infer depth from deformation to that pattern
Infer depth from focus: Points far away are blurry
Infer depth from stereo: Closer points are shifted
The Latest, Coolest...

Light field camera (passive)

“Capture” the light going in every direction at every 3D point

http://excelmathmike.blogspot.com
The Problem

• Mapping: What is the world around me (geometry, landmarks)
  – sense from various positions
  – integrate measurements to produce map
  – assumes perfect knowledge of position

• Localization: Where am I in the world (position wrt landmarks)
  – sense
  – relate sensor readings to a world model
  – compute location relative to model
  – assumes a perfect world model

• Together, these are SLAM (Simultaneous Localization and Mapping)
  – How can you localize without a map?
  – How can you map without localization?

• All localization, mapping or SLAM methods are based on updating a state:
  – What makes a state? Localization? Map? Both?
  – How certain is the state?
Representations for Bayesian Robot Localization

Discrete approaches (’95)
- Topological representation (’95)
  - uncertainty handling (POMDPs)
  - occas. global localization, recovery
- Grid-based, metric representation (’96)
  - global localization, recovery

Particle filters (’99)
- sample-based representation
- global localization, recovery

Kalman filters (late-80s?)
- Gaussians
- approximately linear models
- position tracking

Multi-hypothesis (’00)
- multiple Kalman filters
- global localization, recovery

Robotics

AI
Gaussian (or Normal) Distribution

**Univariate**

\[ p(x) \sim N(\mu, \sigma^2) \]

\[ p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}} \]

**Multivariate**

\[ p(x) \sim N(\mu, \Sigma) \]

\[ p(x) = \frac{1}{(2\pi)^{d/2} \left| \Sigma \right|^{1/2}} e^{-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu)} \]
Properties of Gaussians

\[ X \sim N(\mu, \sigma^2) \]
\[ Y = aX + b \] \implies Y \sim N(a\mu + b, a^2\sigma^2) \]

\[ X_1 \sim N(\mu_1, \sigma_1^2) \]
\[ X_2 \sim N(\mu_2, \sigma_2^2) \] \implies X_1 + X_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) \]

- We stay in the “Gaussian world” as long as we start with Gaussians and perform only linear transformations.
- Same holds for multivariate Gaussians
Dynamical Systems

• “System” that changes
• We will represent a system by its state \( x_t \) at time \( t \)
• We will also require that a system be observable

\[
\begin{align*}
&u_t \\
x_t \\
z_t \\
u_{t+1} \\
x_{t+1} \\
z_{t+1} \\
u_{t+2} \\
x_{t+2} \\
z_{t+2}
\end{align*}
\]

external command input

“process model” (state transition)

“observation model” (measurements of the states)
Kalman Filter

- Seminal paper published in 1960
- Great web page at http://www.cs.unc.edu/~welch/kalman/
- Recursive solution for discrete linear filtering problems
  - A state $x \in \mathbb{R}^n$
  - A measurement $z \in \mathbb{R}^m$
  - Discrete (i.e. for time $t = 1, 2, 3, \ldots$)
  - Recursive process (i.e. $x_t = f(x_{t-1})$)
  - Linear system (i.e. $x_t = A x_{t-1}$)
- The system is defined by:

1) *linear* process model
   \[
   x_t = A x_{t-1}
   \]

   state transition

2) *linear* measurement model
   \[
   z_t = H x_t.
   \]

   observation model

How a state transitions into another state       How a state relates to a measurement
Kalman Filter

1. **Prior estimate** $\hat{x}_t'$ at step $t$: Use the process model to predict what will be the next state of the robot

2. **Posterior estimate** $\hat{x}_t$ at step $t$: Use the observation model to correct the prediction by using sensor measurement

Compute *posterior* estimate as a linear combination of the *prior* estimate and difference between the actual measurement and expected measurement

$$\hat{x}_t = \hat{x}_t' + K_t (z_t - H\hat{x}_t')$$

Where the Kalman gain $K_t$ is a blending factor that adds a measurement innovation.
Kalman Filter

Define **prior error** between true state and prior estimate
\[ e'_t = x_t - \hat{x}'_t \]
and the **prior covariance** as
\[ \Sigma'_t = E(e'_t e'_t^T) \]

Define **posterior error** between true state and posterior estimate
\[ e_t = x_t - \hat{x}_t \]
and it’s **posterior covariance** as
\[ \Sigma_t = E(e_t e_t^T) \]

The gain \( K \) that **minimizes** the **posterior covariance** is defined by
\[ K_t = \Sigma'_t H^T (H\Sigma'_t H^T + R)^{-1} \]

Note that
- If \( R \to 0 \) then \( K_t = H^{-1} \) and \( \hat{x}_t = \hat{x}'_t + K_t(z_t - H\hat{x}'_t) \)
- If \( \Sigma'_t \to 0 \) then \( K_t = 0 \) and \( \hat{x}_t = \)
**Kalman Filter**

- **Recipe:**
  1. Start with an initial guess
     \[
     \hat{\mathbf{x}}_0, \mathbf{\Sigma}_0
     \]
  2. Compute prior (prediction)
     \[
     \hat{\mathbf{x}}'_t = A\hat{\mathbf{x}}_{t-1} + B\mathbf{u}_{t-1} \\
     \mathbf{\Sigma}'_t = A\mathbf{\Sigma}_{t-1}A^T + \mathbf{Q}
     \]
  3. Compute posterior (correction)
     \[
     K_t = \mathbf{\Sigma}'_tH^T (H\mathbf{\Sigma}'_tH^T + \mathbf{R})^{-1} \\
     \hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + K_t (\mathbf{z}_t - H\hat{\mathbf{x}}'_t) \\
     \mathbf{\Sigma}_t = (1 - K_t H)\mathbf{\Sigma}'_t
     \]

**Time update (predict)**

**Measurement update (correct)**

**REPEAT**
Initial
\( \hat{x}_0, \Sigma_0 \)

Predict \( \hat{x}_1 \) from \( \hat{x}_0 \) and \( u_0 \)
\[
\hat{x}_1' = A\hat{x}_0 + B u_0 \\
\Sigma_1' = A\Sigma_0 A^T + Q
\]

Correction using \( z_1 \)
\[
K_1 = \Sigma_1'H^T (H\Sigma_1'H^T + R)^{-1} \\
\hat{x}_1 = \hat{x}_1' + K_1(z_1 - H\hat{x}_1') \\
\Sigma_1 = (1 - K_1H)\Sigma_1'
\]

Predict \( \hat{x}_2 \) from \( \hat{x}_1 \) and \( u_1 \)
\[
\hat{x}_2' = A\hat{x}_1 + B u_1 \\
\Sigma_2' = A\Sigma_1 A^T + Q
\]
Observability

\[ z_t = H x_t \]

- If \( H \) does not provide a one-to-one mapping between the state and the measurement, then the system is \textit{unobservable}

\[ z_t = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x_t \]

In this case \( H \) is singular, such that many states can generate the same observation

\[ K_t = \Sigma_t' H^T (H \Sigma_t' H^T + R)^{-1} \]

\[ \text{rank}(AA^T) = \text{rank}(A) \]
\[ \text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B) \]
Fully Observable vs Partially Observable
Kalman Filter Limitations

• Assumptions:
  – Linear process model
    \[ x_{t+1} = A x_t + B u_t + w_t \]
  – Linear observation model
    \[ z_t = H x_t + v_t \]
  – White Gaussian noise
    \[ N(0, \Sigma) \]

• What can we do if system is not linear?
  – Non-linear state dynamics
    \[ x_{t+1} = f(x_t, u_t, w_t) \]
  – Non-linear observations
    \[ z_t = h(x_t, v_t) \]

\[ \begin{align*}
    x_{t+1} & \approx \tilde{x}_{t+1} + A(x_t - \tilde{x}_t) + W w_t \\
    \tilde{x}_{t+1} & = f(x_t, u_t, 0) \\
    z_t & \approx \tilde{z}_t + H(x_t - \tilde{x}_t) + V v_t \\
    \tilde{z}_t & = h(\tilde{x}_t, 0)
\end{align*} \]
Extended Kalman Filter

- Where $A$, $H$, $W$ and $V$ are Jacobians defined by

$$A(x_t) = \begin{bmatrix}
\frac{\partial f_1(\bar{x}_t, u_t, 0)}{\partial x_1} & \cdots & \frac{\partial f_1(\bar{x}_t, u_t, 0)}{\partial x_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_M(\bar{x}_t, u_t, 0)}{\partial x_1} & \cdots & \frac{\partial f_M(\bar{x}_t, u_t, 0)}{\partial x_N}
\end{bmatrix}$$

$$W(x_t) = \begin{bmatrix}
\frac{\partial f_1(\bar{x}_t, u_t, 0)}{\partial w_1} & \cdots & \frac{\partial f_1(\bar{x}_t, u_t, 0)}{\partial w_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_M(\bar{x}_t, u_t, 0)}{\partial w_1} & \cdots & \frac{\partial f_M(\bar{x}_t, u_t, 0)}{\partial w_N}
\end{bmatrix}$$

$$H(x_t) = \begin{bmatrix}
\frac{\partial h_1(\bar{x}_t, 0)}{\partial x_1} & \cdots & \frac{\partial h_1(\bar{x}_t, 0)}{\partial x_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_M(\bar{x}_t, 0)}{\partial x_1} & \cdots & \frac{\partial h_M(\bar{x}_t, 0)}{\partial x_N}
\end{bmatrix}$$

$$V(x_t) = \begin{bmatrix}
\frac{\partial h_1(\bar{x}_t, 0)}{\partial v_1} & \cdots & \frac{\partial h_1(\bar{x}_t, 0)}{\partial v_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_M(\bar{x}_t, 0)}{\partial v_1} & \cdots & \frac{\partial h_M(\bar{x}_t, 0)}{\partial v_N}
\end{bmatrix}$$
EKF for Range-Bearing Localization

- State $s_t = [x_t \ y_t \ \theta_t]^T$ 2D position and orientation
- Input $u_t = [v_t \ \omega_t]^T$ linear and angular velocity
- Process model

$$f(s_t, u_t, w_t) = \begin{bmatrix} x_{t-1} + (\Delta t)v_{t-1}\cos(\theta_{t-1}) \\ y_{t-1} + (\Delta t)v_{t-1}\sin(\theta_{t-1}) \\ \theta_{t-1} + (\Delta t)\omega_{t-1} \end{bmatrix} + \begin{bmatrix} w_{x_t} \\ w_y \\ w_{\theta_t} \end{bmatrix}$$

- Given a map, the robot sees $N$ landmarks with coordinates

$$l_1 = [x_{l_1} \ y_{l_1}]^T, \ldots, l_N = [x_{l_N} \ y_{l_N}]^T$$

The observation model is

$$z_t = \begin{bmatrix} h_1(s_t, v_1) \\ \vdots \\ h_N(s_t, v_N) \end{bmatrix} h_i(s_t, v_t) = \begin{bmatrix} \sqrt{(x_t - x_{l_i})^2 + (y_t - y_{l_i})^2} \\ \tan^{-1}(\frac{y_t-y_{l_i}}{x_t-x_{l_i}}) - \theta_t \end{bmatrix} + \begin{bmatrix} v_r \\ v_b \end{bmatrix}$$
Linearize Process Model

\[ f(s_t, u_t, w_t) = \begin{bmatrix} x_{t-1} + (\Delta t)v_{t-1}\cos(\theta_{t-1}) \\ y_{t-1} + (\Delta t)v_{t-1}\sin(\theta_{t-1}) \\ \theta_{t-1} + (\Delta t)\omega_{t-1} \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_{\theta_t} \end{bmatrix} \]

\[ A(x_t) = \begin{bmatrix} \frac{\partial f_1(x_t, u_t, 0)}{\partial x_1} & \ldots & \frac{\partial f_1(x_t, u_t, 0)}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M(x_t, u_t, 0)}{\partial x_1} & \ldots & \frac{\partial f_M(x_t, u_t, 0)}{\partial x_N} \end{bmatrix} \]

\[ A = \begin{bmatrix} 1 & 0 & -\Delta t v_{t-1}\sin(\theta_{t-1}) \\ 0 & 1 & \Delta t v_{t-1}\cos(\theta_{t-1}) \\ 0 & 0 & 1 \end{bmatrix} \]
Linearize Observation Model

\[ z_t = \begin{bmatrix} h_1(s_t, v_1) \\ \vdots \\ h_N(s_t, v_N) \end{bmatrix} \]

\[ h_i(s_t, v_t) = \begin{bmatrix} \sqrt{(x_t - x_{l_i})^2 + (y_t - y_{l_i})^2} \\ \tan^{-1}\left(\frac{y_t - y_{l_i}}{x_t - x_{l_i}}\right) - \theta_t \end{bmatrix} + [v_r] \]

\[ H(s_t) = \begin{bmatrix} \frac{\partial h_1(s_t, 0)}{\partial x_t} & \frac{\partial h_1(s_t, 0)}{\partial y_t} & \frac{\partial h_1(s_t, 0)}{\partial \theta_t} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_N(s_t, 0)}{\partial x_t} & \frac{\partial h_N(s_t, 0)}{\partial y_t} & \frac{\partial h_N(s_t, 0)}{\partial \theta_t} \end{bmatrix} \]

\[ H_i(k + 1, j) = \begin{bmatrix} \frac{(\hat{x}_r(k+1|k) - x_{\ell_j})}{\sqrt((\hat{x}_r(k+1|k) - x_{\ell_j})^2 + (\hat{y}_r(k+1|k) - y_{\ell_j})^2)} \\ \frac{(\hat{y}_r(k+1|k) - y_{\ell_j})}{\sqrt((\hat{x}_r(k+1|k) - x_{\ell_j})^2 + (\hat{y}_r(k+1|k) - y_{\ell_j})^2)} \\ -\frac{(\hat{y}_r(k+1|k) - y_{\ell_j})}{\sqrt((\hat{x}_r(k+1|k) - x_{\ell_j})^2 + (\hat{y}_r(k+1|k) - y_{\ell_j})^2)} \\ \frac{1}{1 + \left(\frac{(\hat{y}_r(k+1|k) - y_{\ell_j})}{\hat{x}_r(k+1|k) - x_{\ell_j}}\right)^2 (\hat{x}_r(k+1|k) - x_{\ell_j})} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \]
Extended Kalman Filter

- Kalman Filter Recipe:
  - Given \( \hat{x}_0, \Sigma_0 \)
  - Prediction
    \[
    \hat{x}'_t = A\hat{x}_{t-1} + Bu_{t-1} \\
    \Sigma'_t = A\Sigma_{t-1}A^T + Q
    \]
  - Measurement correction
    \[
    K_t = \Sigma'_tH^T(\Sigma'_tH^T + R)^{-1} \\
    \hat{x}_t = \hat{x}'_t + K(z_t - H\hat{x}'_t) \\
    \Sigma_t = (I - K_tH)\Sigma'_t
    \]

- Extended Kalman Filter Recipe:
  - Given \( \hat{x}_0, \Sigma_0 \)
  - Prediction
    \[
    \hat{x}'_t = f(\hat{x}_{t-1}, u_{t-1}, 0) \\
    \Sigma'_t = A_t\Sigma_{t-1}A_t^T + W_tQW_t^T
    \]
  - Measurement correction
    \[
    K_t = \Sigma'_tH_t^T(\Sigma'_tH_t^T + V_tR^{-1}V_t^T)^{-1} \\
    \hat{x}_t = \hat{x}'_t + K_t(z_t - h(\hat{x}'_t, 0)) \\
    \Sigma_t = (I - K_tH_t)\Sigma'_t
    \]
Kalman Filter for Dead Reckoning

- Robot moves along a straight line with state $x = [p, v]^T$
  - $p$: position
  - $v$: velocity
- $u$ is the input force applied to the robot

Newton's 2nd law: $\dot{v} = \frac{u}{m}$

First order finite difference:

$$\frac{v_{t+1} - v_t}{\Delta t} = \frac{u_t}{m}$$

Robot's state evolution:

$$x_{t+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ \frac{\Delta t}{m} \end{bmatrix} u_t$$

$$x_{t+1} = \begin{bmatrix} p_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} p_t + v_t \Delta t \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{u_t}{m} \Delta t \end{bmatrix}$$

- Robot has velocity sensor

The robot position is not measured

The measured velocity depends on the robot velocity (duh!)
Example

- Let plug some numbers

\[ m = 1 \]
\[ \Delta t = 0.5 \]
From Localization to Mapping

• For us, the landmarks have been a known quantity (we have a map with the coordinates of the landmarks), but landmarks are not part of the state

• Two choices:
  – Make the state the location of the landmarks relative to the robot (I also know exactly where I am …)
    • - No notion of location relative to past history
    • - No fixed reference for landmarks
  – Make the state the robot location now (relative to where we started) plus landmark locations
    • + Landmarks now have fixed location
    • - Knowledge of my location slowly degrades (but this is inevitable …)
EKF for Range-Bearing Localization

- State $s_t = [x_t \ y_t \ \theta_t]^T$ 2D position and orientation
- Input $u_t = [v_t \ \omega_t]^T$ linear and angular velocity
- Process model

$$f(s_t, u_t, w_t) = \begin{bmatrix} x_{t-1} + (\Delta t)v_{t-1}\cos(\theta_{t-1}) \\ y_{t-1} + (\Delta t)v_{t-1}\sin(\theta_{t-1}) \\ \theta_{t-1} + (\Delta t)\omega_{t-1} \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_{\theta} \end{bmatrix}$$

- Given a map, the robot sees $N$ landmarks with coordinates $l_1 = [x_{l1} \ y_{l1}]^T, \ldots, l_N = [x_{lN} \ y_{lN}]^T$

The observation model is

$$z_t = \begin{bmatrix} h_1(s_t, v_1) \\ \vdots \\ h_N(s_t, v_N) \end{bmatrix} h_i(s_t, v_t) = \begin{bmatrix} \sqrt{(x_t-x_{li})^2 + (y_t-y_{li})^2} \\ \tan^{-1}\left(\frac{y_t-y_{li}}{x_t-x_{li}}\right) - \theta_t \end{bmatrix} + \begin{bmatrix} v_r \\ v_b \end{bmatrix}$$
Data Association

• From observation model, we have an expected

\[
\mathbf{z}_t = \begin{bmatrix}
  h_1(s_t, v_1) \\
  \vdots \\
  h_N(s_t, v_N)
\end{bmatrix}
\]

Observation of landmark #1

Observation of landmark #N

• So if we have N landmarks \( l_1, \ldots, l_N \) and we are given a scan \( \mathbf{z}_t \), how do associate each landmark to a scan observation?

\[
\begin{bmatrix}
  \hat{l}_{i_t} \\
  \vdots
\end{bmatrix} = \begin{bmatrix}
  \hat{l}_{i_t}' \\
  \vdots
\end{bmatrix} + K_t \left( \begin{bmatrix}
  \mathbf{z}_{i_t} \\
  \vdots
\end{bmatrix} - h_i(\mathbf{x}_t', 0) \right)
\]

Observation of landmark #\( i \) must be in the \( i \)th row!
Natural Visual Landmarks

- Visual landmarks: SIFT, SURF, corners,
Artificial Landmarks

Lippsett
EKF for Range-Bearing Localization

- **State** \( s_t = [x_t \ y_t \ \theta_t]^T \) 2D position and orientation
- **Input** \( u_t = [v_t \ \omega_t]^T \) linear and angular velocity
- **Process model**
  \[
  f(s_t, u_t, w_t) = \begin{bmatrix}
  x_{t-1} + (\Delta t)v_{t-1}\cos(\theta_{t-1}) \\
  y_{t-1} + (\Delta t)v_{t-1}\sin(\theta_{t-1}) \\
  \theta_{t-1} + (\Delta t)\omega_{t-1}
  \end{bmatrix} + \begin{bmatrix}
  w_{x_t} \\
  w_y \\
  w_{\theta_t}
  \end{bmatrix}
  \]

- Given a map, the robot sees N landmarks with coordinates
  \( l_1 = [x_{l_1} \ y_{l_1}]^T, \ldots, l_N = [x_{l_N} \ y_{l_N}]^T \)

The observation model is

\[
  z_t = \begin{bmatrix}
  h_1(s_t, v_1) \\
  \vdots \\
  h_N(s_t, v_N)
  \end{bmatrix} \\
  h_i(s_t, v_t) = \begin{bmatrix}
  \sqrt{(x_t - x_{l_i})^2 + (y_t - y_{l_i})^2} \\
  \tan^{-1}\left(\frac{y_t-y_{l_i}}{x_t-x_{l_i}}\right) - \theta_t \\
  v_r \\
  v_b
  \end{bmatrix}
  \]
Kalman Filters and SLAM

• Localization: state is the location of the robot
• Mapping: state is the location of 2D landmarks
• SLAM: state combines both
• If the state is \( \mathbf{s}_t = [x_t \ y_t \ \theta_t \ l_{1t}^T \ \ldots \ l_{Nt}^T]^T \)
then we can write a linear observation system
   – note that if we don’t have some fixed landmarks, our system is unobservable
     (we can’t fully determine all unknown quantities)
• Covariance \( \Sigma \) is represented by

   http://ais.informatik.uni-freiburg.de
EKF Range Bearing SLAM
Prior State Estimation

• State \( s_t = [x_t \ y_t \ \theta_t \ l_1^T \ \cdots \ l_N^T] \) position/orientation of robot and landmarks coordinates

• Input \( u_t = [v_t \ \omega_t]^T \) forward and angular velocity

• The process model for localization is

\[
\begin{bmatrix}
   x'_{t-1} \\
   y'_{t-1} \\
   \theta'_{t-1} \\
   l'_{1,t-1} \\
   \vdots \\
   l'_{N,t-1}
\end{bmatrix} = \begin{bmatrix}
x_{t-1} \\
y_{t-1} \\
\theta_{t-1} \\
l_{1,t-1} \\
\vdots \\
l_{N,t-1}
\end{bmatrix} + \begin{bmatrix}
1 & 0 & 0 & \Delta t v_{t-1} \cos(\theta_{t-1}) \\
0 & 1 & 0 & \Delta t v_{t-1} \sin(\theta_{t-1}) \\
0 & 0 & 1 & \Delta t \omega_{t-1}
\end{bmatrix}
\]

This model is augmented for 2N+3 dimensions to accommodate landmarks. This results in the process equation

\[
\begin{bmatrix}
x'_{t-1} \\
y'_{t-1} \\
\theta'_{t-1} \\
l'_{1,t-1} \\
\vdots \\
l'_{N,t-1}
\end{bmatrix} = \begin{bmatrix}
x_{t-1} \\
y_{t-1} \\
\theta_{t-1} \\
l_{1,t-1} \\
\vdots \\
l_{N,t-1}
\end{bmatrix} + \begin{bmatrix}
1 & 0 & 0 & \Delta t v_{t-1} \cos(\theta_{t-1}) \\
0 & 1 & 0 & \Delta t v_{t-1} \sin(\theta_{t-1}) \\
0 & 0 & 1 & \Delta t \omega_{t-1}
\end{bmatrix}
\]

Landmarks don’t depend on external input
EKF Range Bearing SLAM
Prior Covariance Update

We assume static landmarks. Therefore, the function $f(s,u,w)$ only affects the robot’s location and not the landmarks.

$$\Sigma'_t = A_t \Sigma_{t-1} A_t^T + W_t Q W_t^T$$

Jacobian of the robot motion

$$A = \begin{bmatrix}
\frac{\partial x}{\partial x} & \frac{\partial x}{\partial y} & \frac{\partial x}{\partial \theta} \\
\frac{\partial y}{\partial x} & \frac{\partial y}{\partial y} & \frac{\partial y}{\partial \theta} \\
\frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \theta} \\
\frac{\partial x}{\partial x} & \frac{\partial y}{\partial y} & \frac{\partial \theta}{\partial \theta}
\end{bmatrix} + \begin{bmatrix} 0 \\
0 \\
0 \\
1
\end{bmatrix}$$

The motion of the robot does not affect the coordinates of the landmarks.

Jacobian of the process model

2Nx2N identity
EKF Range Bearing SLAM
Kalman Gain

\[
K_t = \Sigma_t^T (H_t\Sigma_t^T H_t^T + V_t R V_t^T)^{-1}
\]

\[
h_i(s_t, v_t) = \begin{bmatrix}
\sqrt{(x_t - x_{li})^2 + (y_t - y_{li})^2} \\
\tan^{-1} \frac{y_t - y_{li}}{x_t - x_{li}} - \theta_t
\end{bmatrix} + [v_r, v_b]
\]

Compute the Jacobian \( H_i \) of each \( h_i \) and then stack them into one big matrix \( H \). Note that \( h_i \) only depends on 5 variables: \( x_r, y_r, \theta_t, x_{li}, y_{li} \)

\[
H_i = \begin{bmatrix}
\frac{\partial y_i}{\partial x_r} \\
\frac{\partial y_i}{\partial y_r} \\
\frac{\partial y_i}{\partial x_{li}} \\
\frac{\partial y_i}{\partial y_{li}} \\
\frac{\partial y_i}{\partial \rho_i^2}
\end{bmatrix} = \\
\begin{bmatrix}
-x_{li}(k) + x_r(k) & -y_{li}(k) + y_r(k) & 0 & \cdots & \frac{x_{li}(k) - x_r(k)}{\rho_i^2} & \frac{y_{li}(k) - y_r(k)}{\rho_i^2} & \cdots \\
-x_{li}(k) + y_r(k) & -y_{li}(k) + x_r(k) & 0 & \cdots & \frac{-y_{li}(k) + y_r(k)}{\rho_i^2} & \frac{x_{li}(k) - x_r(k)}{\rho_i^2} & \cdots \\
-\frac{x_{li}(k) - y_{li}(k)}{\rho_i^2} & -\frac{x_{li}(k) - y_{li}(k)}{\rho_i^2} & -1 & \cdots & -\frac{-y_{li}(k) + y_r(k)}{\rho_i^2} & -\frac{x_{li}(k) - x_r(k)}{\rho_i^2} & \cdots
\end{bmatrix}
\]

\( \rho_i \) is the range of the landmark

Need to be in the correct columns of \( H \)
EKF Range Bearing SLAM Measurement

\[ z_t = \begin{bmatrix} h_1(s_t, v_1) \\ \vdots \\ h_N(s_t, v_N) \end{bmatrix} \iff h_i(s_t, v_t) = \begin{bmatrix} \sqrt{(x_t - x_{i_t})^2 + (y_t - y_{i_t})^2} \\ \tan^{-1} \frac{y_t - y_{i_t}}{x_t - x_{i_t}} - \theta_t \end{bmatrix} + [v_r] \]

- Observe N landmarks \( z_{i_t} = [r_{i_t} \ \phi_{i_t}]^T \)
- Must have data association
  - Which measured landmark corresponds to \( h_i \)?
  - If \( s_t \) contains the coordinates of \( N \) landmarks in the map, \( h_i \) predicts the measurement of each landmark
- Must figure which measured landmark corresponds to \( h_i \)
  \[ \hat{s}_t = \hat{s}_t' + K_t (z_t - h(\hat{s}_t', 0)) \]
  Make sure that each landmark observation \( z_{i_t} \) appears in the correct rows of \( z_t \)

\[
\begin{bmatrix} \hat{l}_{i_t} \\ \vdots \end{bmatrix} = \begin{bmatrix} \hat{l}'_{i_t} \\ \vdots \end{bmatrix} + K_t \left( \begin{bmatrix} \vdots \\ z_{i_t} \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ h_i(\hat{s}_t', 0) \\ \vdots \end{bmatrix} \right)
\]
EKF Range Bearing SLAM
Posterior Update

- From $K_t$ and $H$ update the posterior state estimate
  \[
  \hat{s}_t = \hat{s}_t' + K_t (z_t - h(\hat{s}_t', 0))
  \]
  \[
  \Sigma_t = (I - K_t H_t) \Sigma_t'
  \]

  Tada! And we are done!
Bearing-Only SLAM

Let's assume one landmark for now

\[ s = \begin{bmatrix} x & y & \theta & l_x & l_y \end{bmatrix}^T \]

\[ h(s) = \tan^{-1} \left( \frac{l_y - y}{x_l - x} \right) - \theta \]

\[ z = h(s) \]

Tully 2008

Often use omni-directional sensor
Why Bearing-Only SLAM is Challenging

- We cannot estimate the landmark location with one measurement
- We must guess the range and initialize with a large covariance due to the lack of range information
- The location is very uncertain and difficult to resolve with low parallax measurements
- The measurement model is very nonlinear, which breaks conventional filtering techniques

Tully 2008
Bearing-Only SLAM with EKF

- EKF uses the standard Kalman update

- The Kalman gain is computed through a linearization about the current estimate

- The result diverges

- Very dependent on the initialization “guess” of landmarks

http://www.pracsyslab.org
Mono SLAM

- A visual landmark with a single camera does not provide range
- Data association is given by tracking or matching visual descriptors/patches
Submaps

- As landmarks are added, the state vector grows
- For large maps only a few landmarks can be visible at any given time
- Use submaps to reduce the number of landmarks to manageable numbers
Navigation: RMS Titanic
Leonard & Eustice

- EKF-based system
- 866 images
- 3494 camera constraints
- Path length 3.1km 2D / 3.4km 3D
- Convex hull > 3100m²
- 344 min. data / 39 min. ESDF*

*excludes image registration time
Search of Flight 370

Inertial Navigation System (INS):
A glorified IMU with Kalman filter
Summary

• Basic system modeling ideas

• Kalman filter as an estimation method from a system model

• Linearization as a way of attacking a wider variety of problems

• Mapping localization and mapping into EKF

• Extensions for managing landmark matching and not-well-constrained systems.