Module D: SLAM& Pure Pursuit

Part 1. Introduction to SLAM

Part 2: Cartographer

Part 3. Pure Pursuit

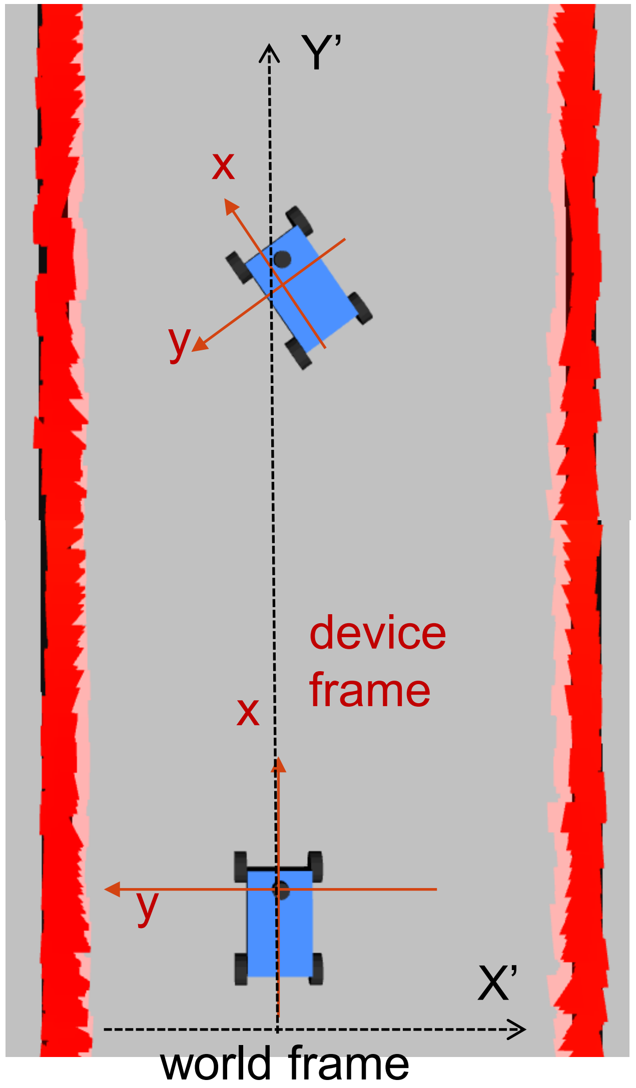
References:

https://f1tenth.org and UPenn ESE 680 Slides

https://google-cartographer.readthedocs.io/en/latest/

Reactive vs. Deliberative Paradigms

- Reactive Paradigm:
  - Sense-act all in the robot's own frame;
  - No need to have a global world representation;
  - Some reactive methods also need to know a global goal.
- Deliberative paradigm
  - Sense-plan-act → have a goal and make plans to achieve it
  - Need a world representation → robot driving needs a map
  - Needs to know where the robot is in the map → localization
  - Plans to get to the destination → navigation and path planning
- Localization, Mapping and Navigation methods
  - Dead reckoning: update location from speed and heading
  - Inertial navigation systems: magnet, gyro and accelerometer
  - GPS – Global Positioning System or Advanced GPS
  - SLAM –Simultaneous Localization and Mapping



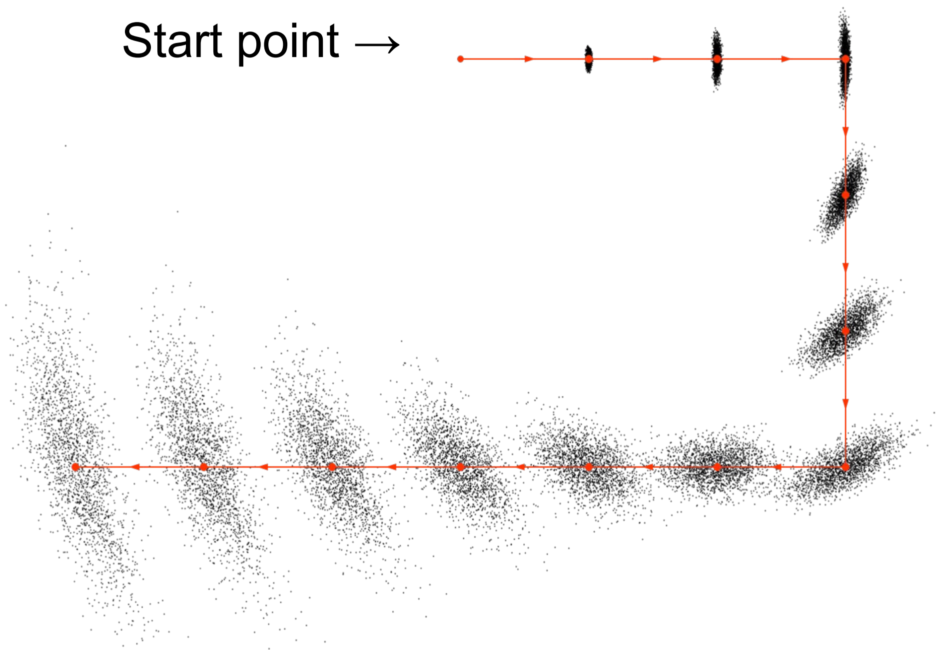
What is SLAM?

- SLAM– Simultaneous Localization and Mapping:
  - Definition by Durrant-Whyte and Bailey 2005: SLAM is a process by which a mobile robot can build a map of an environment and at the same time use this map to deduce its location.
  - Need a map to support path planning → deliberative paradigm and hybrid paradigm;
  - We have GPS and Google maps, why SLAM?
    – GPS denied area (Urban environment), indoor, underwater, underground (e.g. Tunnel).
  - Localization and Mapping -- A Chicken and Egg problem;
  - Solution: an iterative process with localization, scan matching, and loop closure.
- Available SLAM packages online (openslam-org.github.io)
  - Camera/Vision based: EKF-SLAM, RatSLAM, ORB-SLAM
  - Lidar based: GridSLAM (w/ particle filter), FastSLAM (T. Bailey), tinySLAM
  - Feature based: TreeMap, Google cartographer
  - Graph-based: FLIRTLib, G2O, HOG-MAN, vertigo

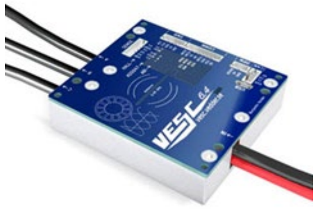
SLAM: Localization w/o GPS

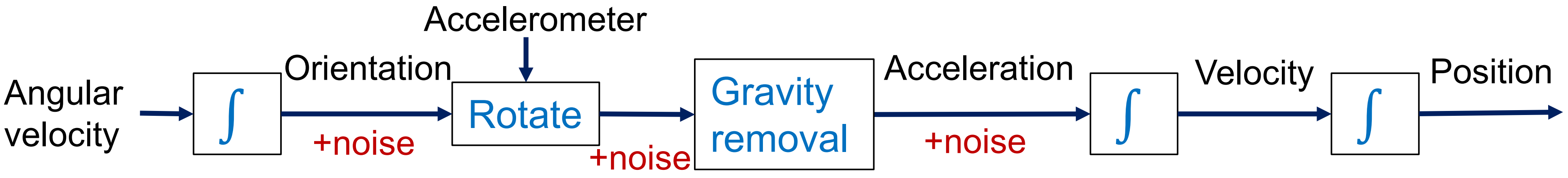
LEHIGH UNIVERSITY

- Localization definition:
  - Determine the state (position and orientation) of a robot with respect to the environment represented by a global world frame or a local map relative to robot's starting point.
- Conventional method using Odometry and IMU (Inertial Measurement Unit)
  - Start at a known pose and integrate control and motion measurements to estimate the current pose
  - Open loop estimation → error accumulation.



Uncertainty/variance increases w/ time





Rotation by Quaternion

LEHIGH UNIVERSITY

- What is Quaternion? (by William Rowan Hamilton in 1843)
  - Definition: $\boldsymbol{q} = q_0 + q_1 i + q_2 j + q_3 k$ Eq(1)
  - Rule of Production: $i \odot i = j \odot j = k \odot k = -1$

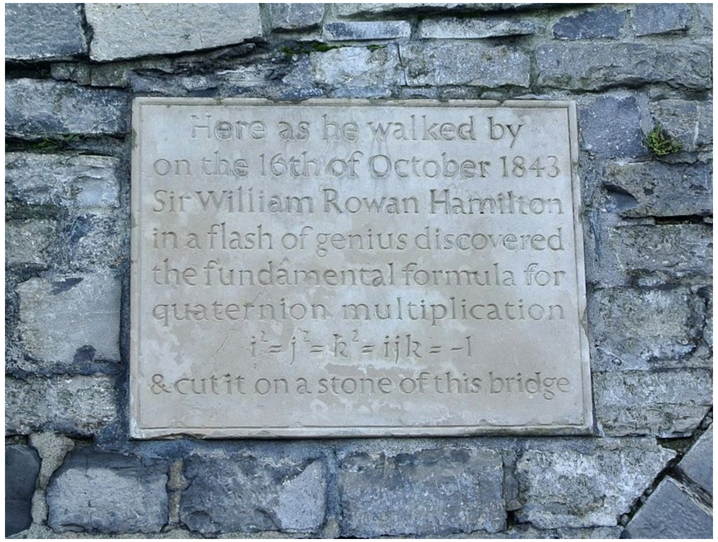
    $i \odot j = k, \quad j \odot k = i, \quad k \odot i = j$

    non-commutative $j \odot i = -k, k \odot j = -i, \quad i \odot k = -j$
  - Inverse: $\boldsymbol{q}^{-1} = \frac{q_0 - q_1 i - q_2 j - q_3 k}{q_0^2 + q_1^2 + q_2^2 + q_3^2}$ Eq(2)
- Rotation Quaternion
  - To rotate a vector $\boldsymbol{v}$ by an angle along another vector in the 3D space:

    $|\boldsymbol{\theta}|$ = the rotation angle, $\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|}$ = the normalized rotation vector

    $\boldsymbol{q}_{\boldsymbol{\theta}} = \cos\left(\frac{|\boldsymbol{\theta}|}{2}\right) + \sin\left(\frac{|\boldsymbol{\theta}|}{2}\right)\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|}$ Eq(1)

    $\boldsymbol{q}_{\boldsymbol{\theta}}^{-1} = \cos\left(-\frac{|\boldsymbol{\theta}|}{2}\right) + \sin\left(-\frac{|\boldsymbol{\theta}|}{2}\right)\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|}$ Eq(2)
  - Then the rotation is $\overline{\boldsymbol{v}}_{\boldsymbol{\theta}} = \boldsymbol{q}_{\boldsymbol{\theta}} \odot \overline{\boldsymbol{v}} \odot \boldsymbol{q}_{\boldsymbol{\theta}}^{-1}$, where $\overline{\boldsymbol{v}}$ = quaternion of $\boldsymbol{v}$



Quaternion plaque on Brougham (Broom) Bridge, Dublin, says:
Here as he walked by
on the 16th of October 1843
Sir William Rowan Hamilton
in a flash of genius discovered
the fundamental formula for
quaternion multiplication
$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$
& cut it on a stone of this bridge

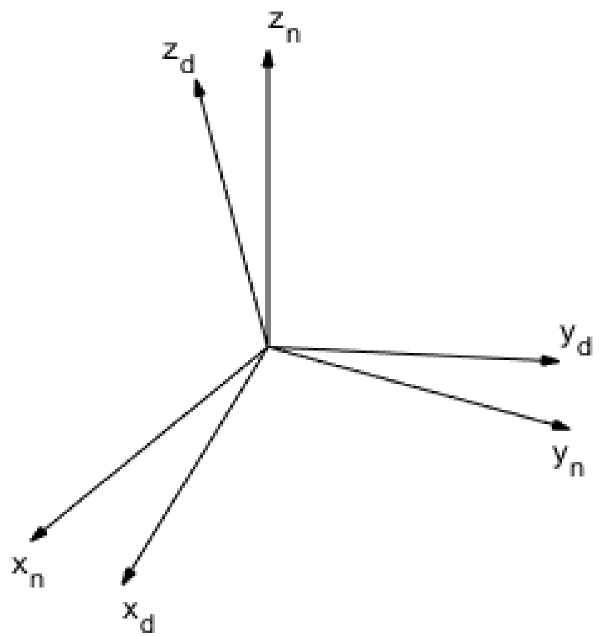
Quaternion Rotation Example

- Rotate a vector $\boldsymbol{v} = [1,2,3]$ by $120°$ along direction $[1,1,1]$ (provided by Xiyuan Zhu)
  - Vector $\boldsymbol{v}$ represented as a quaternion $\overline{\boldsymbol{v}} = 0 + 1i + 2j + 3k$
  - We have angle $|\boldsymbol{\theta}| = \frac{2\pi}{3}$, normalized rotation vector $\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|} = \frac{\sqrt{3}}{3}[1,1,1\,]$
  - Rotation quaternion

$$\boldsymbol{q}_{\boldsymbol{\theta}} = \cos\left(\frac{|\boldsymbol{\theta}|}{2}\right) + \sin\left(\frac{|\boldsymbol{\theta}|}{2}\right)\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|} = \cos\left(\frac{\pi}{3}\right) + \sin\left(\frac{\pi}{3}\right)\frac{\sqrt{3}}{3}(i + j + k)$$

$$\boldsymbol{q}_{\boldsymbol{\theta}}^{-1} = \cos\left(-\frac{|\boldsymbol{\theta}|}{2}\right) + \sin\left(-\frac{|\boldsymbol{\theta}|}{2}\right)\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|} = \cos\left(-\frac{\pi}{3}\right) + \sin\left(-\frac{\pi}{3}\right)\frac{\sqrt{3}}{3}(i + j + k)$$

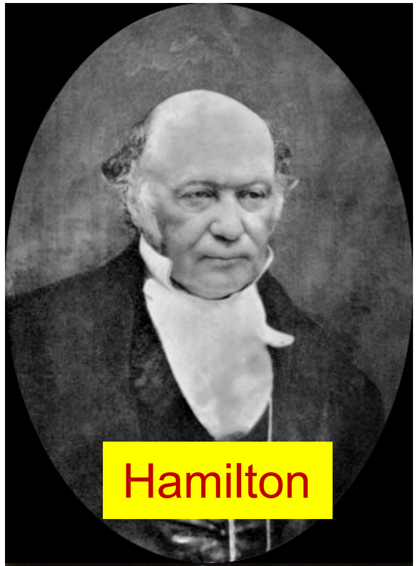
  - Rotation is done by $\overline{\boldsymbol{v}}_{\boldsymbol{\theta}} = \boldsymbol{q}_{\boldsymbol{\theta}} \odot \overline{\boldsymbol{v}} \odot \boldsymbol{q}_{\boldsymbol{\theta}}^{-1} = 0 + 3i + 1j + 2k$
  - How to transform the coordinates in device frame to global frame



- References on Quaternion:
  - https://en.wikipedia.org/wiki/Quaternion
  - Tutorial video: https://eater.net/quaternions/video/intro

Quaternion vs. Euler's Angle

- Euler's angle is defined as euler = [roll, pitch, yaw] in ROS
  - This is known as Intrinsic Euler, or Tait–Bryan angles: defined in device frame
  - also Extrinsic Euler or proper Euler angles: defined in global frame
  - Euler's angel and transform were published in 1776.
- Quaternion and Euler's angle conversion
  - Euler to Quaternion: localization from IMU (device frame) to map (global frame)
  - Quaternion to Euler: Used to generate control parameters and observation estimates.
  - Both are used heavily in SLAM
- Why not use Euler's transform directly?
  - Avoids the Gimbal lock in Euler's transform
  - Uses nonsingular representation: compared with Euler angles.
  - Better numerical properties, is faster and more compact than matrices;
  - Pairs of unit quaternions represent a rotation in 4D space.



Waypoint_logger.py

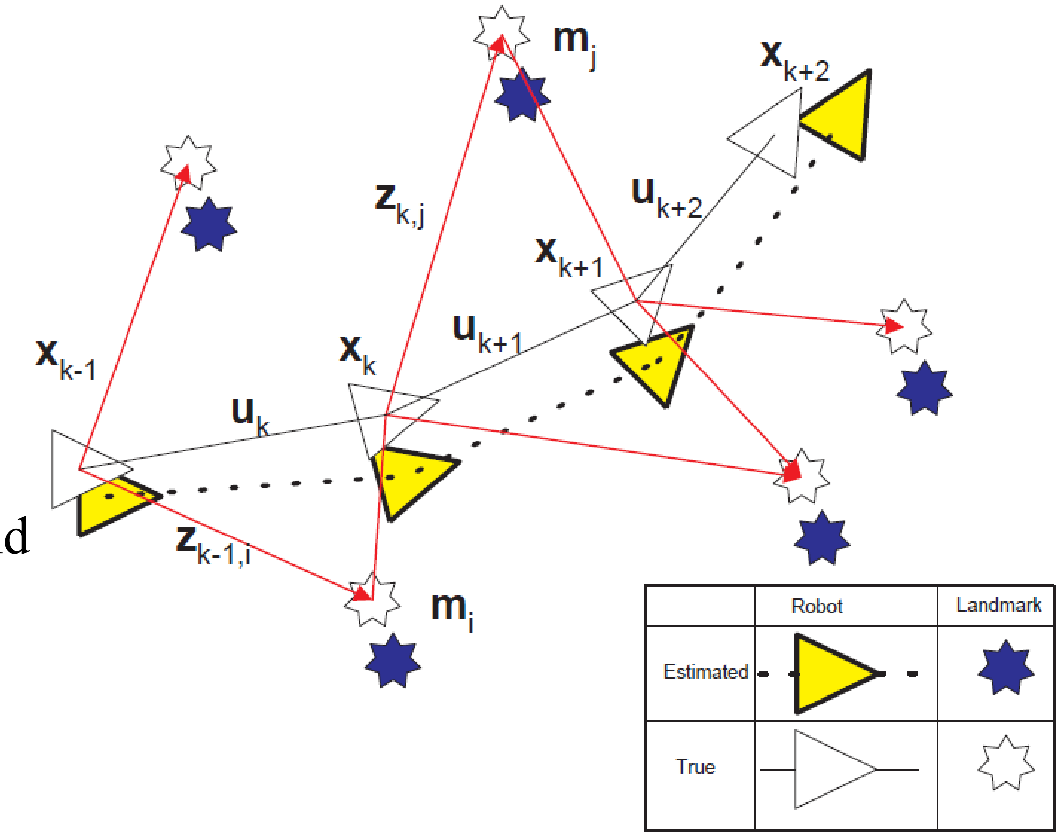
```
import tf
from os.path import expanduser
from time import gmtime, strftime
from numpy import linalg as LA
from tf.transformations import euler_from_quaternion
from nav_msgs.msg import Odometry
home = expanduser('~')
file = open(strftime(home+'/rcws/logs/wp-%Y-%m-%d-%H-%M-%S',gmtime())+'.csv', 'w')

def save_waypoint(data):
    quaternion = np.array([data.pose.pose.orientation.x,
                           data.pose.pose.orientation.y,
                           data.pose.pose.orientation.z,
                           data.pose.pose.orientation.w])
    euler = tf.transformations.euler_from_quaternion(quaternion)
    speed = LA.norm(np.array([data.twist.twist.linear.x, data.twist.twist.linear.y, data.twist.twist.linear.z]),2)

def listener():
    rospy.init_node('waypoints_logger', anonymous=True)
    rospy.Subscriber('pf/pose/odom', Odometry, save_waypoint)
    rospy.spin()
```

SLAM: The Basic Idea

- Quaternion is used to update robot's location at each time instant
- SLAM: How to utilize sensing data to improve localization and pose estimation?
- An iterative process:
  - Localization: updates robot pose/location using Odometry & IMU
  - Scan matching: use Lidar/camera data to find scenes and landmarks, build local submaps
  - Use submaps to correct pose estimation
  - Register submaps to build global maps – loop closure
  - Use global map to correct pose/location estimation



Figures from: Durrant-Whyte and Bailey 2005, Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms

Math Formulation of SLAM

- Definitions:
  - $\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} = \{\mathbf{X}_{0:k-1}, \mathbf{x}_k\}$: The history of vehicle locations.
  - $\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$: The history of control inputs.
  - $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n\}$ The set of all landmarks.
  - $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$: The set of all landmark observations.
- Probabilistic Models
  - Observation model $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$.
  - Motion model $P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k)$
  - Time update: $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1}$
  - Measurement update: $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \dfrac{P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})}$
- SLAM solves for joint probability $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$

Correlation of Landmarks and Robot

Definitions

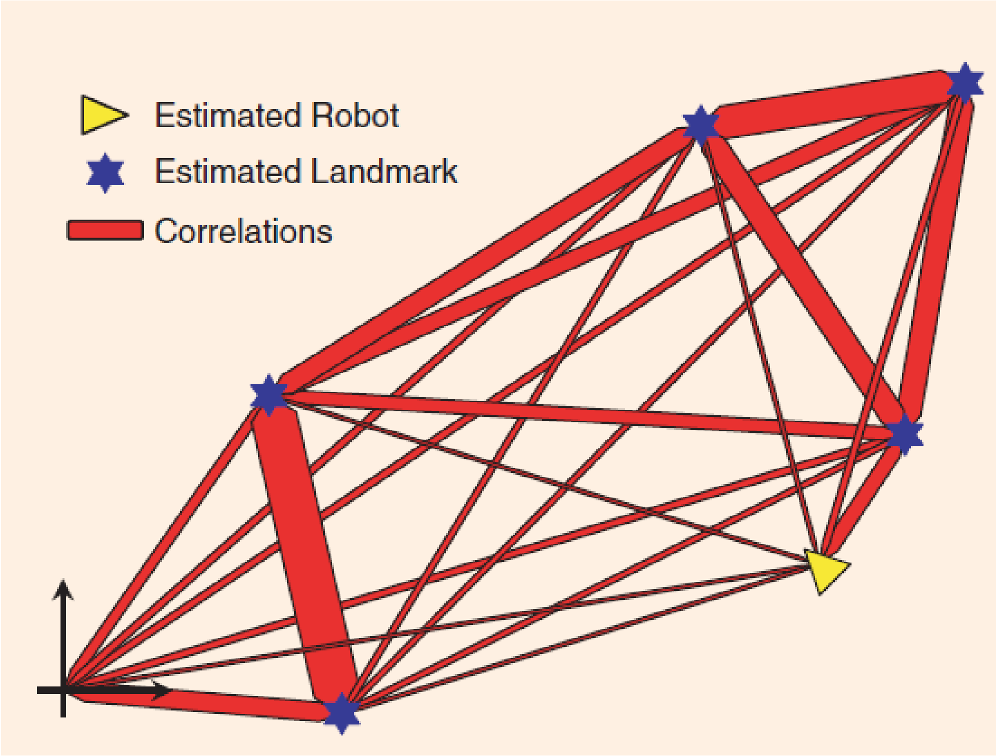
- Conditional Mean (ensemble average)

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \mathrm{E}\begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} \mid \mathbf{Z}_{0:k} \end{bmatrix}$$

- Covariance matrix:

$$\begin{aligned} \mathbf{P}_{k|k} &= \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{xm}^T & \mathbf{P}_{mm} \end{bmatrix}_{k|k} \\ &= \mathrm{E}\left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix}\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix}^T \mid \mathbf{Z}_{0:k}\right] \end{aligned}$$

Recall the nav_msgs has two layers of pose and twist. What is in the first layer of pose or twist?



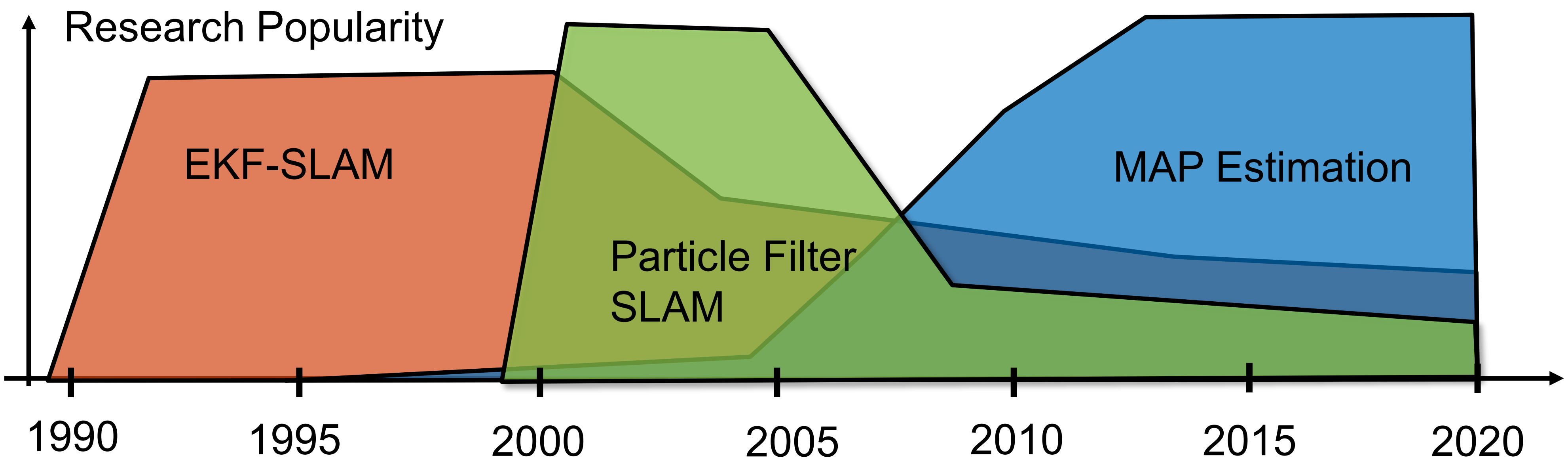
Correlations among robot states and landmarks are very strong → good SLAM solutions possible

Figure from: G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csobra. A solution to the simultaneous localisation and mapping (SLAM) problem. IEEE Trans. Robotics and Automation,17(3):229{241, 2001.

History of SLAM Algorithms

LEHIGH UNIVERSITY

- Historical Development (1986-2004): Probabilistic Foundations
  - Extended Kalman Filter (EKF) SLAM: still used in visual sensing + inertial odometry systems.
  - Particle Filter SLAM: very efficient localization – will be used in Lab 4.
- Modern Era (2004-Now): Algorithmic Improvements
  - MAP Estimation, or factor graph optimization, graph-SLAM,
  - Smoothing and mapping (SAM), bundle adjustment, Machine Learning.



Important References for SLAM

LEHIGH UNIVERSITY

- Extended Kalman Filter (EKF) SLAM:
  - Randall C. Smith and Peter Cheeseman, “On the Representation and Estimation of Spatial Uncertainty,” SAGE J., Volume: 5 issue: 4, page(s): 56-68, 1986.
  - J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," in IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 376-382, June 1991.
- Particle Filter SLAM:
  - Arnaud Doucet, Nando de Freitas, Kevin Murphy, Stuart Russell, “Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks,” UAI 2000.
  - M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," 2003 IEEE ICRA, Taipei, Taiwan, 2003, vol.2. pp. 1985-1991.
  - Sebastian Thrun, Wolfram Burgard, and Dieter Fox, “Probabilistic Robotics” MIT Press, 2005;
- MAP Estimation:
  - M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," in IEEE Transactions on Robotics, vol. 24, no. 6, pp. 1365-1378, Dec. 2008.
  - C. Cadena et al, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Transactions on Robotics, Vol. 32, pp. 1309—1332, 2016.

EKF-SLAM

- EKF-SLAM formulation:
  - Motion model: $P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \Longleftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k$
  - Observation model: $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \Longleftrightarrow \mathbf{z}(k) = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k,$
- EKF-SLAM solution:
  - Time update:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\ \mathbf{P}_{xx,k|k-1} &= \nabla\mathbf{f}\ \mathbf{P}_{xx,k-1|k-1}\nabla\mathbf{f}^T + \mathbf{Q}_k\end{aligned}$$

  - Observation update:

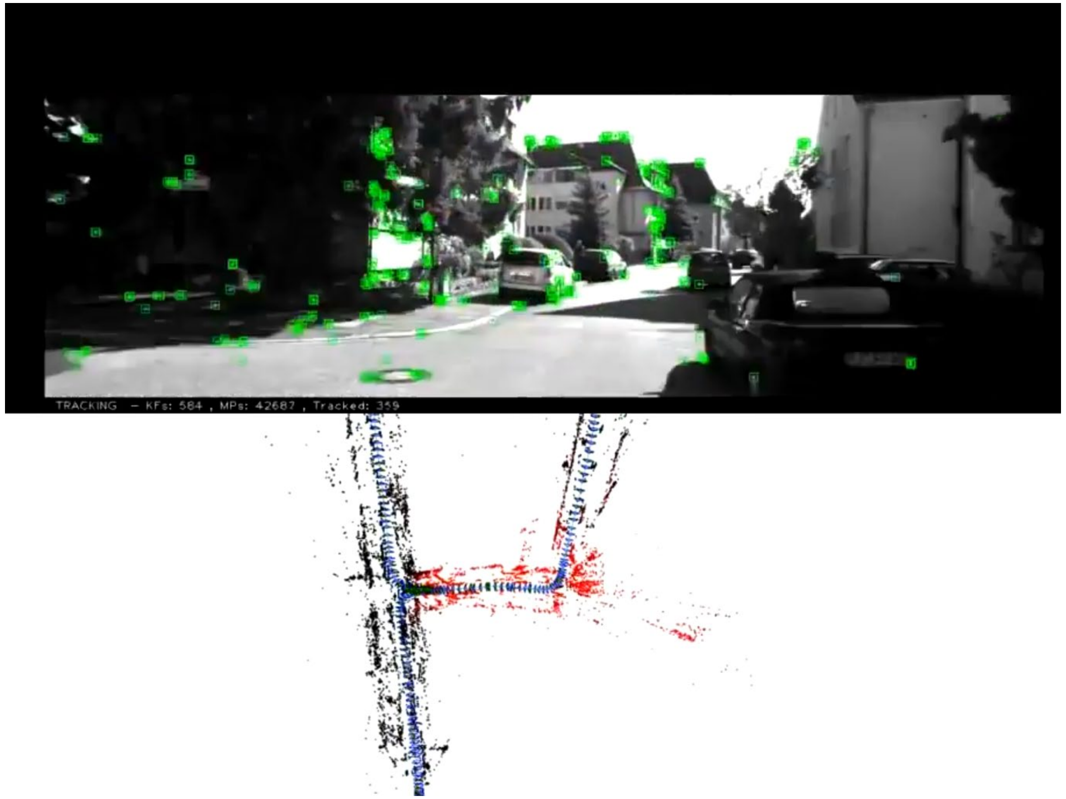
$$\begin{bmatrix}\hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{m}}_{k-1}\end{bmatrix} + \mathbf{W}_k\left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{m}}_{k-1})\right]$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}_k\mathbf{S}_k\mathbf{W}_k^T$$

where

$$\mathbf{S}_k = \nabla\mathbf{h}\mathbf{P}_{k|k-1}\nabla\mathbf{h}^T + \mathbf{R}_k$$

$$\mathbf{W}_k = \mathbf{P}_{k|k-1}\nabla\mathbf{h}^T\mathbf{S}_k^{-1}$$



[EKF-SLAM Example: https://www.youtube.com/watch?v=7eMUQomI36Q](https://www.youtube.com/watch?v=7eMUQomI36Q)