Simultaneous Localization & Mapping

Paril Jain
Previous Week

IMU and LIDAR

Localization

PID Control
Limitations : Basic Path Planning

- High Level Path Assignments

- 2nd right, 2nd right, 1st right, 1st left, 1st right
Race Lines

ACCELERATION

LATE APEX

BALANCED THROTTLE OR TRAIL BRAKING

TURN IN

GEAR CHANGE

BRAKING

PEDAL TRANSITION

FULL THROTTLE

drivingfast.net
Limitations: No Future Information

- Tracing the correct line into "S shape" corner.
- Early possibility to apply power
- Correct apex and correct line
- Maximizing speed out of corners and minimize time of vehicle's load transfer
- Incorrect apex and line
- Slower speed out of corner caused by late roll of vehicle and late power application
- Late possibility to apply power
System Overview

Mapping → Localization → Path Planning → Control
SLAM: A Chicken-Egg problem
Overview of SLAM

1. Laser scan at time = t₀
2. Register the scan as initial Map
3. Change in time/position
4. Laser Scan at next time instant
5. Estimate pose change from transformation of new laser scan
6. Align Laser scan to new Pose estimate
7. Map Update
Video provided separately

- Car running in corridor
- Map being generated
- Video will be used for explaining the overview
Occupancy Grid Mapping

Measurement Model

Actual Map

Created Grid Map

Occupied Cell
Free Cell
Un-Explored Cell
Occupancy Grid Mapping

Measurement Model

- Measurement:
  \( m_{x,y} = 1 \) LiDAR hit
  \( m_{x,y} = 0 \) No occlusion
Occupancy Grid Mapping

Measurement Model

- Measurement:
  \[ m_{x,y} = 1 \quad \text{LiDAR hit} \]
  \[ m_{x,y} = 0 \quad \text{No occlusion} \]

- Map Cell:
  \[ Z = 1 \quad \text{Occupied} \]
  \[ Z = 0 \quad \text{UnExplored} \]
  \[ Z = -1 \quad \text{Free} \]
Occupancy Grid Mapping

Measurement Model

• Measurement:
  \[ m_{x,y} = 1 \quad \text{LiDAR hit} \]
  \[ m_{x,y} = 0 \quad \text{No occlusion} \]

• Map Cell:
  \[ Z = 1 \quad \text{Occupied} \]
  \[ Z = 0 \quad \text{UnExplored} \]
  \[ Z = -1 \quad \text{Free} \]

• Measurement Model:
  \[ p(z|m_{x,y}) \]
Occupancy Grid Mapping

\[
\log \text{odd}_{\text{occ}} := \log \frac{p(z = 1|m_{x,y} = 1)}{p(z = 1|m_{x,y} = 0)}
\]

Log Probability for occupied cells

\[
\log \text{odd}_{\text{free}} := \log \frac{p(z = -1|m_{x,y} = 0)}{p(z = -1|m_{x,y} = 1)}
\]

Log Probability for free cells
Occupancy Grid Mapping

Map Update:
- Cells with \( z = 1 \):
  - \( \log \text{odd} = \log \text{odd} + \log \text{odd\_occ} \)
- Cells with \( z = -1 \):
  - \( \log \text{odd} = \log \text{odd} - \log \text{odd\_free} \)
Occupancy Grid Mapping

Map Update:

- Cells with $z = 1$:
  - $\log\ odd = \log\ odd + \log\ odd_{occ}$

- Cells with $z = -1$:
  - $\log\ odd = \log\ odd - \log\ odd_{free}$

- Threshold the cell values by upper/lower limit to avoid being completely certain
Occupancy Grid Mapping

Map Update:

- Cells with $z = 1$:
  - $\log \text{odd} = \log \text{odd} + \log \text{odd}_\text{occ}$
- Cells with $z = -1$:
  - $\log \text{odd} = \log \text{odd} - \log \text{odd}_\text{free}$

- Threshold the cell values by upper /lower limit to avoid being completely certain

$p(z|m_{x,y})$

log odds probability
Occupancy Grid Mapping

Map Update:

- Cells with $z = 1$:
  - $\log \text{odd} = \log \text{odd} + \log \text{odd}\_occ$

- Cells with $z = -1$:
  - $\log \text{odd} = \log \text{odd} - \log \text{odd}\_free$

- Threshold the cell values by upper/lower limit to avoid being completely certain

![Graph showing log odds vs. probability with saturation limit at 0.5 and 1.](image)
Registering the first Scan
Registering the first Scan
Scan Matching

Pose of the Car at $t = t_1$

Laser Scans w.r.t car at Time $t = t_1$
Scan Matching

Pose of the Car at $t = t_1$

Laser Scans w.r.t. car at Time $t = t_1$

Laser Scans w.r.t. car at Time $t = t_2$
Scan Matching

Pose of the Car at $t = t_1$

Pose of the Car at $t = t_2$

Laser Scans w.r.t car at Time $t = t_1$

Laser Scans w.r.t car at Time $t = t_2$
Scan Matching

Iterative Closest Point

Source: Mathworks – File Exchange: Iterative Closest Point Package
Scan Matching

Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds

Source: Mathworks – File Exchange: Iterative Closest Point Package
Scan Matching

Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds

- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

\[
[R, T] = \arg\min_{R,T} \sqrt{\sum_{i=1}^{n} \text{dist}(R \cdot \text{Red}_i + T, \text{Blue}_i)^2}
\]
Scan Matching

Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds

- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

\[
[R, T] = \arg\min_{R,T} \sum_{i=1}^{n} \text{dist}(R \cdot \text{Red}_i + T, \text{Blue}_i)^2
\]
Scan Matching

Iterative Closest Point

- Minimize Root Mean Squared Distance between Point Clouds

- Find R & T matrix for transformed Point Cloud w.r.t original cloud.

\[
[R, T] = \arg\min_{R,T} \sqrt{\sum_{i=1}^{n} dist(R * \text{Red}_i + T, \text{Blue}_i)^2}
\]
Scan matching: Hector Slam
Scan matching: Hector Slam

Robot Pose \( \xi = (p_x, p_y, \psi)^T \)
Scan matching: Hector Slam

Robot Pose

\[ \xi = (p_x, p_y, \psi)^T \]

Impact coordinates of \(i^{\text{th}}\) scan in world frame

Total of \(n\) scans

\[ \xi^* = \arg\min_{\xi} \sum_{i=1}^{n} \left[1 - M(S_i(\xi))\right]^2 \]
Scan matching: Hector Slam

Robot Pose \( \xi = (p_x, p_y, \psi)^T \)

Impact coordinates of \( i^{\text{th}} \) scan in world frame

Total of \( n \) scans

Map Value at coordinates given by \( S_i \)

\[ \xi^* = \arg\min_{\xi} \sum_{i=1}^{n} [1 - M(S_i(\xi))]^2 \]
Scan matching: Hector Slam

\[
\sum_{i=1}^{n} [1 - M(S_i(\xi + \Delta \xi))]^2 \to 0.
\]
Scan matching: Hector Slam

\[ \sum_{i=1}^{n} [1 - M(S_i(\xi + \Delta \xi))]^2 \rightarrow 0. \]

Taylor Expansion of Function M
Scan matching: Hector Slam

Taylor Expansion of Function M

\[
\sum_{i=1}^{n} [1 - M(S_i(\xi + \Delta \xi))]^2 \rightarrow 0.
\]

\[
\sum_{i=1}^{n} \left[1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta \xi \right]^2 \rightarrow 0.
\]
Scan matching: Hector Slam

Taylor Expansion of Function $M$

$$\sum_{i=1}^{n} [1 - M(S_i(\xi + \Delta \xi))]^2 \rightarrow 0.$$ 

Solving for $\Delta \xi$ yields Gauss-Newton Equation

$$\sum_{i=1}^{n} \left[ 1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta \xi \right]^2 \rightarrow 0.$$ 

Evaluation of Gauss-Newton equation gives a step $\Delta \xi$ that minimizes the objective function.
Raw LiDAR Scans
Scans after transforming by $\Delta \xi$ at each stage
Map Update
Multi-Resolution Map Representation

- 20 cm Grid Cell
- 10 cm Grid Cell
- 5 cm Grid Cell
Saving the map
Saving the map

• ROS Package called **MAP Server**

• Allows saving a map currently being published over /map topic

• Save the map:
  rosrun map_server map_saver [-f mapname]

• Load the map:
  rosrun map_server map_server <name.yaml>
Odometry Using Hector Mapping
Odometry Using Hector Mapping

- Using Hector Slam for measuring $\Delta \xi$, while discarding the map
Odometry Using Hector Mapping

- Using Hector Slam for measuring $\Delta \xi$, while discarding the map

- Optional Approach: CSM (Canonical Scan Matcher) by Andrea Censi
  - Scan matching between 2 scans
System Tf tree

- Map Frame
- Odom Frame
  - Tf Provided by Hector dometry
- Base Frame
  - Tf required by Hector package
- Laser Frame
Parameters for Hector SLAM: ROS

- **map resolution** - Grid resolution
- **map_update_distance_thresh** - minimum distance to be travelled before having a map update
- **map_update_angle_thresh** - minimum angle to be travelled before a map update
- **laser_max_dist** - Laser sensor specification
- **update_factor_free** - Log odds probability for occupied cells
- **update_factor_occupied** - Log odds probability for free cells
Next Lecture

• Using the map generated today

• Localizing using Adaptive Monte Carlo localization (AMCL)

• Integrating hector odometry and AMCL