PROBABILISTIC NAVIGATION TECHNIQUES FOR SWIMMING ROBOTS

Anwar Ahmad Quraishi
Supervisor: Yannick Morel
Introduction

Project Envirobot: Autonomous swimming robot for locating pollution sources in water bodies

Objective: Track regions of high concentration and possible sources

Goal of my project: Compare different probabilistic navigation techniques for the swimming robot
Outline

1. Envirobot project
2. Probabilistic navigation techniques
3. Numerical simulations
4. Impact of parameters
5. Performance metrics and tuning of parameters
6. Experimental tests
7. Conclusion
1. Envirobot project

- Measure concentration as the robot moves
- Assess local concentration trends
- Estimate and move in the direction of highest gradient

Trajectories of the robot navigating a diffusion plume, moving towards areas of greater concentration
2. Probabilistic navigation

- Noisy sensors and actuators, hence motion not deterministic
- Accurate estimate of trajectory important
- Shape of real and estimated trajectory must be similar
2. Bayes filter

- General algorithm to compute probability distributions over robot state
- Models noise in sensors and actuators
- Recursive algorithm consisting of two steps
  - Predict new state from the prior state
  - Correct prediction using sensor measurements

\[
p(s_k | Z_{k-1}, U_k) = \int p(s_k | s_{k-1}, u_k)p(s_{k-1} | Z_{k-1}, U_{k-1}) ds_{k-1}
\]

**Motion model**

\[
p(s_k | Z_k, U_k) = \eta p(z_k | s_k)p(s_k | Z_{k-1}, U_k)
\]

**Measurement model**
2. Extended Kalman filter

- Represent probability distributions by a Gaussian
- Linearize state transition and measurement functions
- Recursive algorithm, computes $\mu_k, \Sigma_k$ using $\mu_{k-1}, \Sigma_{k-1}$ at each time step $k$

\[
\begin{align*}
\bar{\mu}_k &= g(\mu_{k-1}, u_k) \\
\bar{\Sigma}_k &= G_k \Sigma_{k-1} G_k^T + R_k \\
K_k &= \bar{\Sigma}_k H_k^T (H_k \bar{\Sigma}_k H_k^T + Q_k)^{-1} \\
\mu_k &= \bar{\mu}_k + K_k (z_k - h(\bar{\mu}_k)) \\
\Sigma_t &= (I - K_k H_k) \bar{\Sigma}_k
\end{align*}
\]
2. Particle filter.

- Probability distribution represented by a set of $m$ particles
- Higher density where probability is higher
- Can model arbitrary distributions, not just linear Gaussian
- Recursive algorithm; computes the distribution at time step $k$

Predict

- Propagate each particle $s[k]^{m}$ forward using the motion model, $p(s_k|u_k, s_{k-1}^{[m]})$

Correct

- Compute importance of each particle, $w_k^{[m]} = p(z_k|s_k^{[m]})$
- Include particle $s[k]^{m}$ in the new set with probability $\propto w_k^{[m]}$
3. Extended Kalman filter

Mean GPS error: \(3.70 \pm 1.22\,\text{m}\)
Mean estimation error: \(3.29 \pm 1.70\,\text{m}\)
3. Particle filter

Mean GPS error: $3.70 \pm 1.22\text{m}$
Mean estimation error: $3.14 \pm 1.35\text{m}$
4. Tuning of parameters

- Noise in sensors and actuators is modeled by Gaussian
- Zero mean and finite variance
- Theoretically, chosen according to datasheet or by experimental tests
- Directly affects performance of navigation algorithm

- $\sigma_{GPS}$, GPS measurement error variance
- $\sigma_c$, Compass measurement error variance
- $\sigma_{xy}$, Position prediction error variance
- $\sigma_\psi$, Heading prediction error variance
4. Impact of parameters

High motion error variance
Low measurement error variance

Low motion error variance
High measurement error variance
5. Performance metrics

• Position estimation error
  • Difference between the actual and estimated position of the robot over its trajectory

\[
e_e = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (\hat{x}_k - x_k)^2 + (\hat{y}_k - y_k)^2}
\]

• Shape error
  • Measure of the deformation of the shape of the estimated trajectory without considering the offsets

\[
e_s = \frac{1}{n-w} \sum_{k=1}^{n-w} \sqrt{\frac{1}{w} \sum_{l=k}^{k+w} (\hat{x}_l - x_l + x_k)^2 + (\hat{y}_l - y_l + y_k)^2}
\]
5. Performance metrics

- Direction error
  - Measure of the effect of navigation on estimation of the direction of greatest concentration gradient

\[ e_\psi = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (\hat{\psi}_{sk} - \psi_{sk})^2} \]
5. Tuning of parameters

- Effect on position estimation error

![Graphs showing the effect of parameters on position error.](image)
5. Tuning of parameters

- **Effect on Shape error**
5. Tuning of parameters

- **Effect on direction estimation error**

**Direction error [rad]**

- **GPS error variance [m^2]**
- **Compass error variance [deg^2]**

**Direction error [rad]**

- **Position prediction error variance [m^2]**
- **Heading prediction error variance [deg^2]**
6. Experimental tests

- Tests performed in a pool of water
- Cameras detect position; additional error to emulate GPS
- Navigation algorithm running off-board
- Tested EKF; PF not tested due to computational constraints
6. Experimental tests

- Implementation simplified because of computational constraints

Original implementation
- State
  \[
  s_k \triangleq \begin{bmatrix}
  x_k & y_k & \psi_k & v_k & \dot{v}_k & \dot{\psi}_k
\end{bmatrix}^T.
\]
- 6 × 6 motion error covariance matrix
- 6 × 6 Jacobian matrix of motion model

Simplified implementation
- State
  \[
  s_k \triangleq \begin{bmatrix}
  x_k & y_k & \psi_k
\end{bmatrix}^T
\]
- 3 × 3 motion error covariance matrix
- 3 × 3 Jacobian matrix of motion model
- \(v_k, \dot{v}_k, \dot{\psi}_k\) computed but not part of EKF
- Matrix inverse computed in closed form
6. Experimental tests

- Position measurement without added noise
6. Experimental tests

- Position measurement with added noise
Conclusion

• Two probabilistic navigation techniques compared
  • EKF makes approximations, can represent only Gaussian distributions but computationally efficient
  • Particle filter overcomes these shortcomings but is computationally complex

• Effect of tunable parameters on trajectory estimation is studied

• Quantitative measures of performance are developed and parameters tuned to improve performance

• EKF tested experimentally
Thank you