

CSC 144V: Autonomous Vehicles

Lecture 9.1

SLAM Fundamentals Exercises Solution

Department of Computer Science,
Hofstra University

Exercise: 1-D EKF-SLAM update

1-D world. Robot x and landmark m on a line; measurement $z = m - x + \text{noise}$ (variance 0.09); motion $x \leftarrow x + u + \text{noise}$. At $x_0 = 0$ (known) the robot measures $z_0 = 4.8$ and initializes $m = 4.8$ (variance 0.09). Then it moves $u = 3.0$ (variance 0.16) and measures $z_1 = 1.6$.

- (a) Fill in the table (state x , m and covariance entries) after the initial observation, the prediction, and the update.
(b) Compute the innovation y , $S = HPH^T + R$ ($H = [-1, 1]$) and the Kalman gain K . (c) Why is $P_{xm} \neq 0$ after the update?

Step	x	m	P_{xx}	P_{mm}	P_{xm}
after z_0					
predict ($u = 3$)					
update ($z_1 = 1.6$)					

Exercise Solution: 1-D EKF-SLAM update

- (b) Predicted $\hat{z} = 4.8 - 3.0 = 1.8$, so $y = -0.2$. $S = 0.16 + 0.09 + 0.09 = 0.34$. $K = PH^T/S = (-0.16, 0.09)/0.34 = (-0.471, 0.265)$.
- (c) The landmark was initialized from the robot pose, and the measurement depends on **both**: the update correlates them ($\rho = 0.0424/\sqrt{0.0847 \cdot 0.0662} = 0.57$). Learning about one now improves the other; P_{mm} fell from 0.09 to 0.066.

Step	x	m	P_{xx}	P_{mm}	P_{xm}
after z_0	0	4.8	0	0.09	0
predict ($u = 3$)	3.0	4.8	0.16	0.09	0
update ($z_1 = 1.6$)	3.094	4.747	0.0847	0.0662	0.0424

Exercise: 1-D pose graph

1-D pose graph with $x_0 = 0$ fixed. Odometry edges: $x_1 - x_0 = 2.0$ and $x_2 - x_1 = 2.0$, each with information $\Omega = 50$. Loop-closure edge: $x_2 - x_0 = 3.7$ with $\Omega = 100$. Initial guess from odometry: $x = (2, 4)$.

- (a) Compute the three residuals e and the cost $F = \sum \Omega e^2$ at the initial guess.
- (b) Build H and b for the unknowns (x_1, x_2) .
- (c) Solve $H \Delta x = -b$ and give the new x and the new cost.
- (d) What is the standard deviation of x_2 , from H^{-1} ?

Exercise Solution: 1-D pose graph

- (a) $e_{01} = 0$, $e_{12} = 0$, $e_{02} = 4 - 3.7 = 0.3$; $F = 100 \cdot 0.3^2 = 9.0$.
- (b) $H = [[50 + 50, -50], [-50, 50 + 100]] = [[100, -50], [-50, 150]]$; $b = \sum \Omega e J^T = (0, 30)$ (only the loop edge has a residual).
- (c) $\Delta x = -H^{-1}b = (-0.12, -0.24) \rightarrow x = (1.88, 3.76)$. Residuals: $-0.12, -0.12, +0.06$; $F = 50 \cdot 0.0144 + 50 \cdot 0.0144 + 100 \cdot 0.0036 = 1.8$. One step is exact because the problem is linear.
- (d) $H^{-1} = [[0.012, 0.004], [0.004, 0.008]] \rightarrow \sigma(x_2) = \sqrt{0.008} \approx 0.089 \text{ m}$. The loop closure spread the 0.3 m inconsistency over all three edges, weighted by Ω .

Exercise: Gating and data association

A predicted landmark has innovation covariance $S = \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}$. Three candidate measurements have innovations $y_A = (0.8, -0.4)$, $y_B = (1.5, 0.2)$, $y_C = (1.6, 0.9)$.

(a) Compute $d^2 = y^T S^{-1} y$ for each candidate and test it against the χ^2 gates for 2 DoF (95 %: 5.99, 99 %: 9.21).

(b) Which candidate does nearest-neighbor association choose? What can go wrong?

Candidate	y	d^2	95 % gate	99 % gate
A	(0.8, -0.4)			
B	(1.5, 0.2)			
C	(1.6, 0.9)			

Exercise Solution: Gating and data association

- (b) Nearest neighbor (smallest d^2) picks **A**. B is also inside the gate, so the association is **ambiguous**: a wrong choice moves the landmark and the pose the wrong way, and EKF-SLAM cannot undo it (the past is marginalized). Remedies: joint compatibility, multiple hypotheses (FastSLAM), or a verified graph edge.
- C is a clear outlier at 95 % but would pass a 99 % gate: a tighter gate rejects more outliers but also more true matches.

Candidate	y	d^2	95 % gate	99 % gate
A	(0.8, -0.4)	2.40	inside	inside
B	(1.5, 0.2)	4.54	inside	inside
C	(1.6, 0.9)	6.32	outside	inside

Exercise: EKF vs. FastSLAM cost

EKF-SLAM keeps the pose (3) and N landmarks (2 each) with a full covariance; one update costs about N^2 operations. FastSLAM with $M = 100$ particles costs about $M \log_2 N$ per update. Doubles take 8 bytes.

(a) Fill in the table for $N = 1,000$ and $N = 100,000$ landmarks.

(b) At which map size does a full-covariance EKF become impractical, and what do graph SLAM and FastSLAM do differently?

N	State dim	Covariance entries	Memory	EKF ops $\approx N^2$	FastSLAM $\approx M \log_2 N$
1,000					
100,000					

Exercise Solution: EKF vs. FastSLAM cost

- State dimension $3 + 2N$; covariance has $(3 + 2N)^2$ entries. For $N = 10^5$ the covariance alone needs ≈ 320 GB, and each update touches all of it.
- (b) EKF is fine up to a few hundred landmarks, hard beyond a few thousand. **FastSLAM** factors the posterior into N tiny 2×2 EKFs per particle; **graph SLAM** stores a **sparse** H (each constraint links two nodes) instead of a dense covariance.

N	State dim	Covariance entries	Memory	EKF ops $\approx N^2$	FastSLAM $\approx M \log_2 N$
1,000	2,003	4.0×10^6	32 MB	10^6	$\approx 1,000$
100,000	200,003	4.0×10^{10}	320 GB	10^{10}	$\approx 1,700$

Exercise: ATE and RPE

Ground-truth and estimated 2-D positions of five poses are given (already aligned). Truth: (0,0), (1,0), (2,0), (3,0), (4,0). Estimate: (0,0), (1.1,0.1), (2.0,0.3), (3.2,0.2), (4.1,0.5).

(a) Compute the position error of each pose and the ATE (RMSE). (b) Compute the RPE of each step (error of the relative motion) and its RMSE.

(c) The estimate looks good locally but bends away: which metric shows it? (d) A false loop closure is added: which would increase most, ATE or RPE?

Pose	Truth	Estimate	Error (m)
1	(0, 0)	(0, 0)	
2	(1, 0)	(1.1, 0.1)	
3	(2, 0)	(2.0, 0.3)	
4	(3, 0)	(3.2, 0.2)	
5	(4, 0)	(4.1, 0.5)	

Exercise Solution: ATE and RPE

- (a) errors 0, 0.141, 0.300, 0.283, 0.510 $\rightarrow$ ATE = $\sqrt{((0 + 0.02 + 0.09 + 0.08 + 0.26)/5)} = \mathbf{0.30\ m}$ (max 0.51 m).
- (b) relative motions (1.1, 0.1), (0.9, 0.2), (1.2, -0.1), (0.9, 0.3) vs. (1, 0): errors 0.141, 0.224, 0.224, 0.316 $\rightarrow$ RPE RMSE = **0.23 m per step**.
- (c) **ATE** measures global consistency (accumulated drift); RPE measures local accuracy. (d) A false loop edge bends the whole trajectory: **ATE** grows a lot, RPE only near the bad edge. This is why SLAM is evaluated with both, and with several runs.

Pose	Truth	Estimate	Error (m)
1	(0, 0)	(0, 0)	0
2	(1, 0)	(1.1, 0.1)	0.141
3	(2, 0)	(2.0, 0.3)	0.300
4	(3, 0)	(3.2, 0.2)	0.283
5	(4, 0)	(4.1, 0.5)	0.510