



FRA532 : Mobile Robot

Lecture 4

State Estimator

Kitti Thamrongaphichartkul

Institute of Field Robotics
King Mongkut's University of Technology Thonburi
Bangkok, Thailand

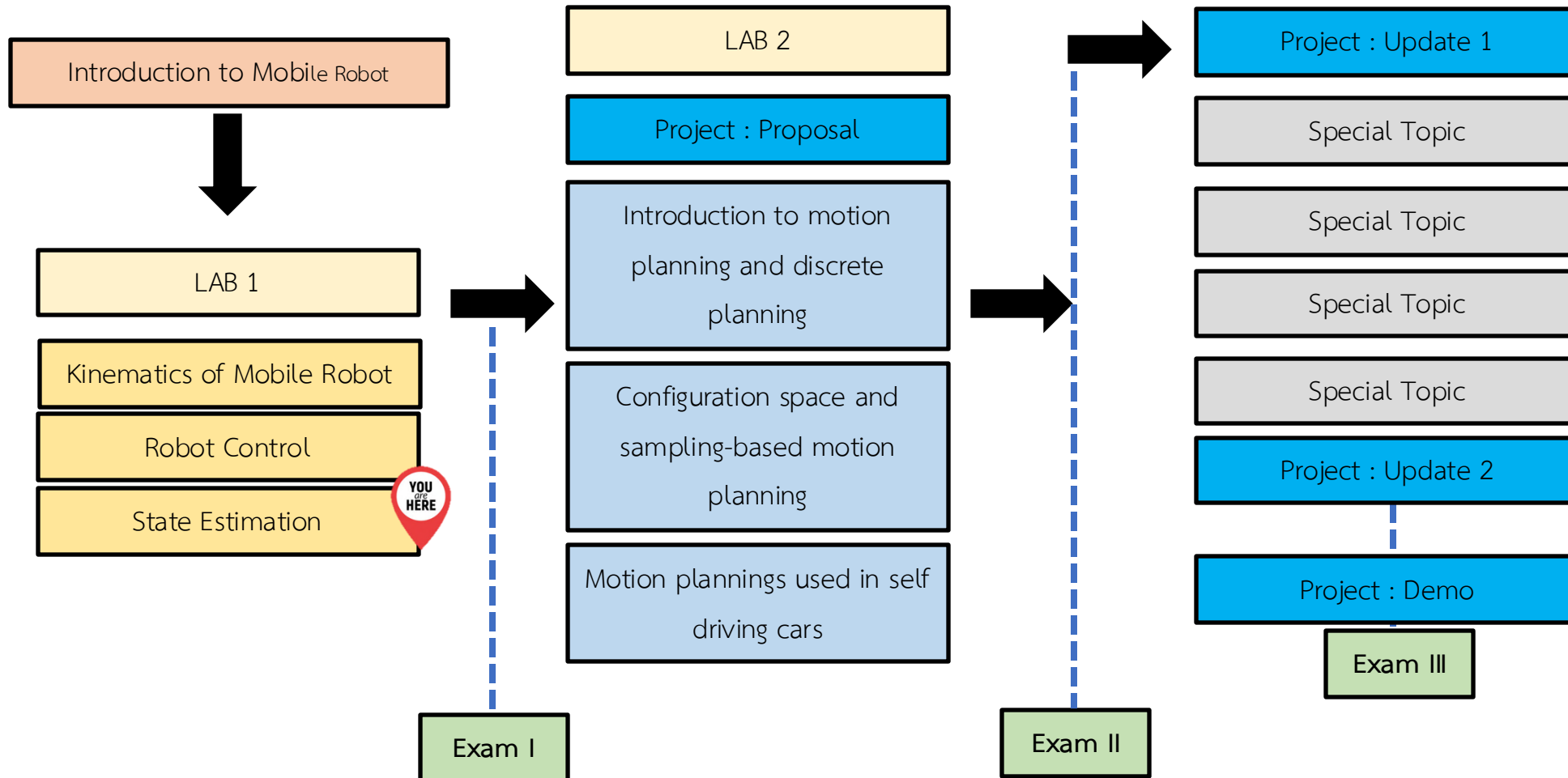


แผนการสอน

Week	Date	Lecture	Topic	Module	LAB / HW		Instructor	หมายเหตุ
					Assign	Due		
1	16-Jan-2025	1	Introduction to Mobile Robot (Motivation)				Aj.Nook	
2	23-Jan-2025	2	Kinematics of Mobile Robot		LAB 1		Aj.Nook	
3	30-Jan-2025	3	Mobile Robot Control				Aj.Nook	
4	6-Feb-2025	4	30 ปี ฟิโน				Aj.Nook	
5	13-Feb-2025	5	State Estimator				Aj.Nook	
6	20-Feb-2025		EXAM 1					
7	27-Feb-2025	6	MAP (Slam, Localization)		LAB 2	LAB 1	Aj.Nook	
8	4-March-2025	7	EXAM 1 / Hackathon Exam (24 Hour)				Aj.Nook	Project : Proposal
9	13-March-2025	8	Introduction to motion planning and discrete planning				Aj.Tee	
10	20-March-2025	9	Configuration space and sampling-based motion planning			LAB 2	Aj.Tee	
11	27-March-2025	10	Motion plannings used in self driving cars				Aj.Tee	
12	3-April-2025		EXAM 2					
13	10-April-2025	11	Project : Update 1				Aj.Nook	
14	18-April 2025	12	Special Topic I				Aj.Nook / Dummy	
15	24-April 2025	13	EXAM 2 / CBS + Nav2				Aj.Nook / Dummy	
16	1 May 2025	14	Special Topic III				Aj.Nook / Dummy	
17	8 May 2025	15	Project : Update 2				Aj.Nook	
18	15 May 2025	-	-					
19	22 May 2025	-	-					
20	29 May 2025	16	Project : Demo				Aj.Nook	

เนื้อหา

Link : <https://github.com/kittinook/MobileRobotics2025/tree/main>

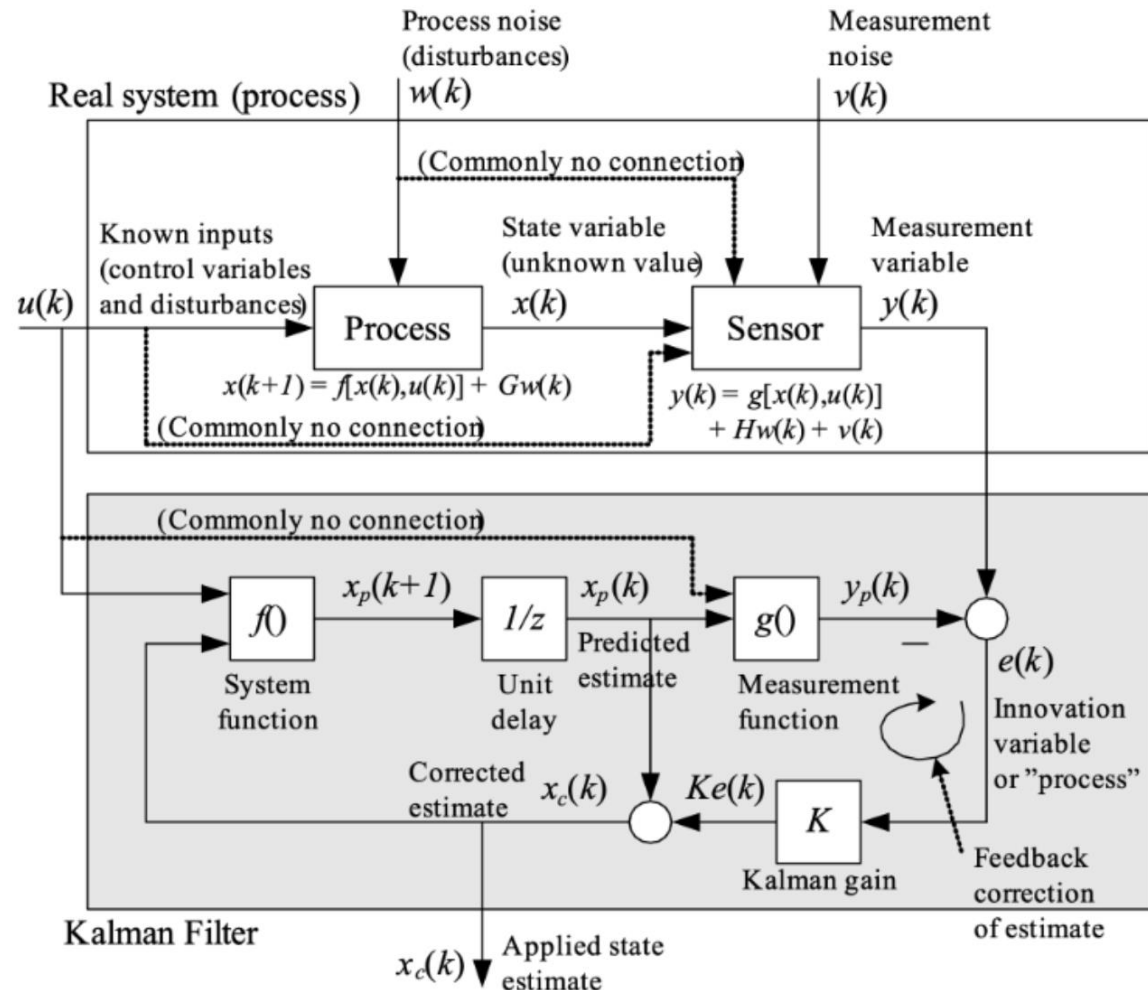


Agenda

- Review Kalman Filter (FRA233)
- Investigation of EKF in “robot_localization”
- Code “ekf_v4.ipynb”
- robot_localization package

Review Kalman Filter (FRA233)

Kalman Filter VS Extended Kalman Filter



linear Kalman filter

EKF

$$\bar{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{F} = \left. \frac{\partial f(\mathbf{x}_t, \mathbf{u}_t)}{\partial \mathbf{x}} \right|_{\mathbf{x}_t, \mathbf{u}_t}$$

$$\bar{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^T + \mathbf{Q}$$

$$\bar{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

$$\bar{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^T + \mathbf{Q}$$

$$\mathbf{y} = \mathbf{z} - \mathbf{H}\bar{\mathbf{x}}$$

$$\mathbf{H} = \left. \frac{\partial h(\bar{\mathbf{x}}_t)}{\partial \bar{\mathbf{x}}} \right|_{\bar{\mathbf{x}}_t}$$

$$\mathbf{y} = \mathbf{z} - h(\bar{\mathbf{x}})$$

$$\mathbf{K} = \bar{\mathbf{P}}\mathbf{H}^T(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R})^{-1}$$

$$\mathbf{K} = \bar{\mathbf{P}}\mathbf{H}^T(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R})^{-1}$$

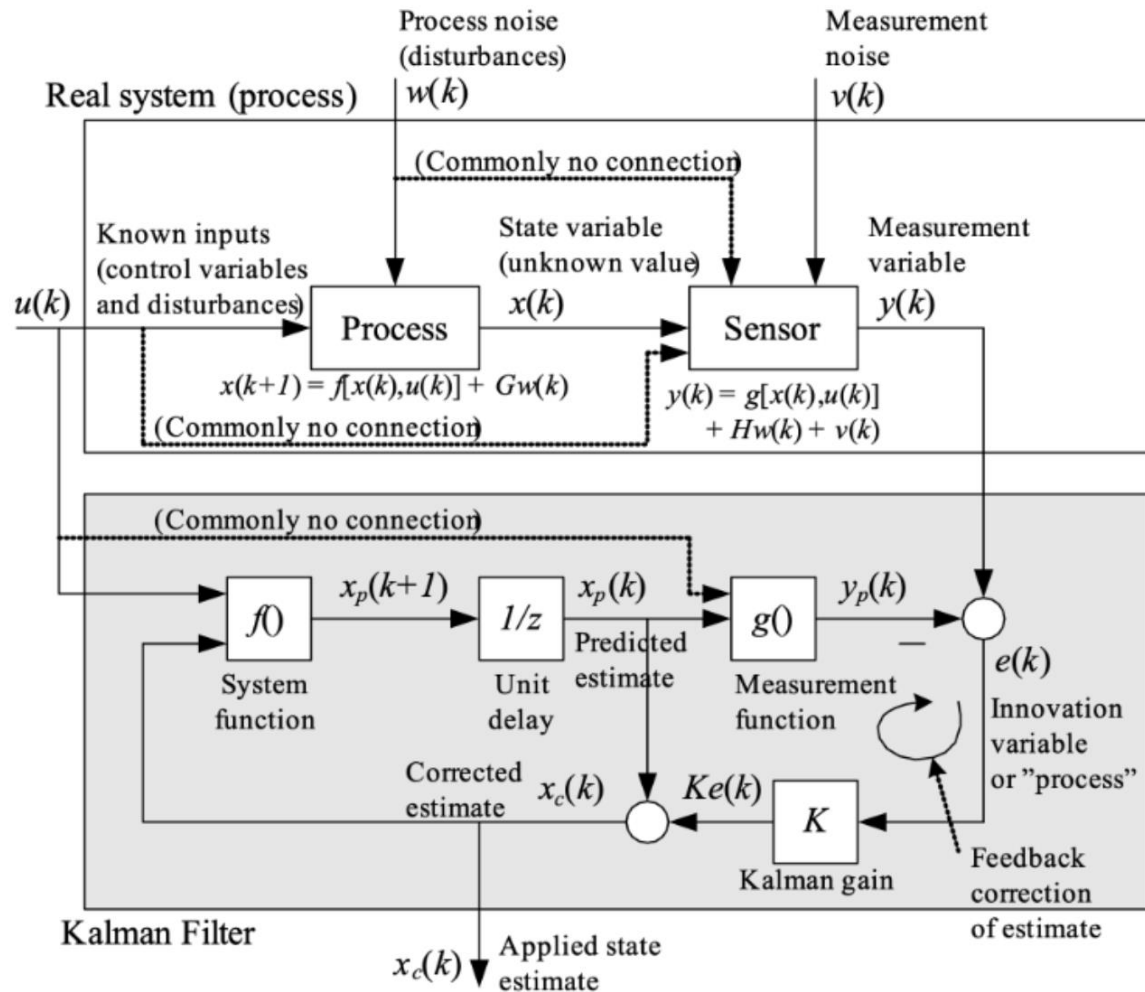
$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{K}\mathbf{y}$$

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{K}\mathbf{y}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\mathbf{P}}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\mathbf{P}}$$

How to Design Extended Kalman Filter



- Design State Transition Function
- Design the Process Noise Matrix
- Design the Control Function
- Design the Measurement Function
- Design the Measurement Noise Matrix
- Initial Conditions

Investigation of EKF in “robot_localization”

Code “ekf_v4.ipynb”

Implementation

1. กำหนดค่าความไม่แน่นอน (Noise) ให้กับระบบ

กำหนด Q สำหรับความไม่แน่นอนของการคำนวณ (process noise)

กำหนด R สำหรับเซ็นเซอร์ odometry และ IMU (measurement noise)

2. สร้างฟังก์ชันช่วยคำนวณ

ฟังก์ชันคำนวณ **Rotation Matrix** จากค่า roll, pitch, yaw

ฟังก์ชันคำนวณ **Jacobian Matrix** สำหรับแปลงความเร็วเชิงมุมให้เป็นการเปลี่ยนแปลงของมุม

ฟังก์ชันคำนวณอนุพันธ์ของ Rotation Matrix และ Jacobian (สำหรับการหา linearization)

3. สร้างแบบจำลองการเคลื่อนที่ (Dynamic Model)

รับค่า state ปัจจุบันที่ประกอบด้วย ตำแหน่ง, ทิศทาง, ความเร็ว, ความเร็วเชิงมุม และความเร่ง

คำนวณตำแหน่งใหม่โดยใช้การหมุน (Rotation Matrix) บวกกับระยะทางที่เดินทาง (จากความเร็วและความเร่ง)

คำนวณทิศทางใหม่โดยใช้ Jacobian คูณกับความเร็วเชิงมุม

อัปเดต

Implementation

4. คำนวณ Jacobian ของแบบจำลอง (F Matrix)

หาอนุพันธ์ของแบบจำลอง เพื่อประมาณค่าการเปลี่ยนแปลง state
ใช้อนุพันธ์ของ Rotation Matrix และ Jacobian ที่ได้จากขั้นตอนที่ 2

5. ขั้นตอน Prediction (คาดการณ์ state ใหม่)

ใช้แบบจำลองการเคลื่อนที่ (dynamic model) ในการคำนวณ state ที่คาดว่าจะเกิดขึ้น (xPred)
คำนวณ covariance ใหม่ (PPred) โดยใช้ Jacobian (F) และเพิ่ม process noise Q

6. ขั้นตอน Update (ปรับปรุง state ด้วยข้อมูลเซ็นเซอร์)

สำหรับ odometry:

รับข้อมูลตำแหน่งและความเร็วจากเซ็นเซอร์
คำนวณความคลาดเคลื่อนระหว่างข้อมูลวัดกับค่าที่คาดการณ์ได้
คำนวณ Kalman Gain แล้วปรับปรุงค่า state และ covariance

สำหรับ IMU:

รับข้อมูลมุม (roll, pitch, yaw), ความเร็วเชิงมุม และความเร่ง
คำนวณความคลาดเคลื่อน (innovation) และปรับมุมให้ถูกต้อง (normalize)
คำนวณ Kalman Gain แล้วปรับปรุงค่า state และ covariance

Implementation

7. ROS Node

สร้าง Node:

- ตั้งค่า state เริ่มต้นและ covariance
- สร้าง subscriber รับข้อมูล odometry และ IMU
- ตั้ง timer ให้ทำงานเป็นช่วง ๆ (เช่น ทุก 0.1 วินาที)

สร้าง Timer Callback:

- คำนวณเวลาที่ผ่านไป (dt)
- ทำขั้นตอน prediction ด้วย EKF
- ตรวจสอบว่ามีข้อมูลใหม่จากเซ็นเซอร์หรือไม่ ถ้ามีก็ทำ update
- ส่งออก (publish) ค่าประมาณตำแหน่งและทิศทาง (PoseStamped) ไปยัง topic ที่กำหนด

robot_localization package

https://github.com/cra-ros-pkg/robot_localization



Q & A

A Cradle of Future Leaders in Robotics