

Advanced Robotics : Mobile Robotics

Lecture 13 : Sensor Fusion

SPONSORED BY AI FOR ALL

Agenda

- Odometry
- Uncertainty
- State Estimator

จุดประสงค์หลัก :

- เพื่อให้ผู้เรียนเข้าใจแนวคิดและหลักการของการประมาณสถานะของหุ่นยนต์

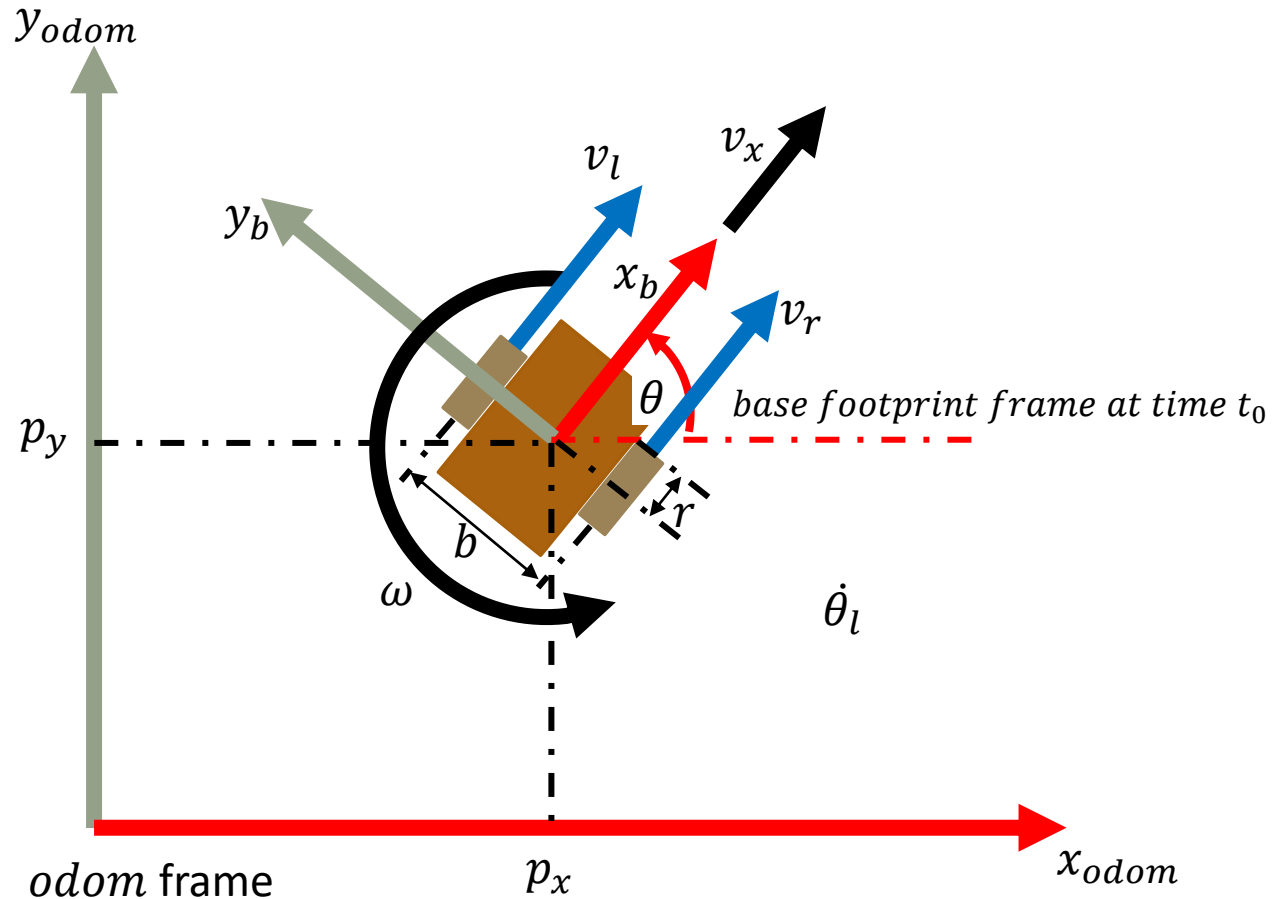
Odometry

- ใช้สำหรับประมาณความเร็วและตำแหน่งของหุ่นยนต์
- ประกอบด้วย
 - Parent Frame
 - Child Frame
 - PoseWithCovariance
 - TwistWithCovariance

Diagram illustrating the kinematic model of a mobile robot in a 2D coordinate system. The robot is represented by a brown rectangle with a smaller brown rectangle inside it. The robot's position is (p_x, p_y) in the odom frame. The robot's orientation is θ . The robot's velocity vector is v_x . The robot's angular velocity is ω . The robot's linear velocity is v_l . The robot's right wheel velocity is v_r . The robot's left wheel velocity is v_l . The robot's base footprint frame at time t_0 is shown as a red dashed line. The robot's base footprint frame at time t_0 is labeled "base footprint frame at time t_0 ".

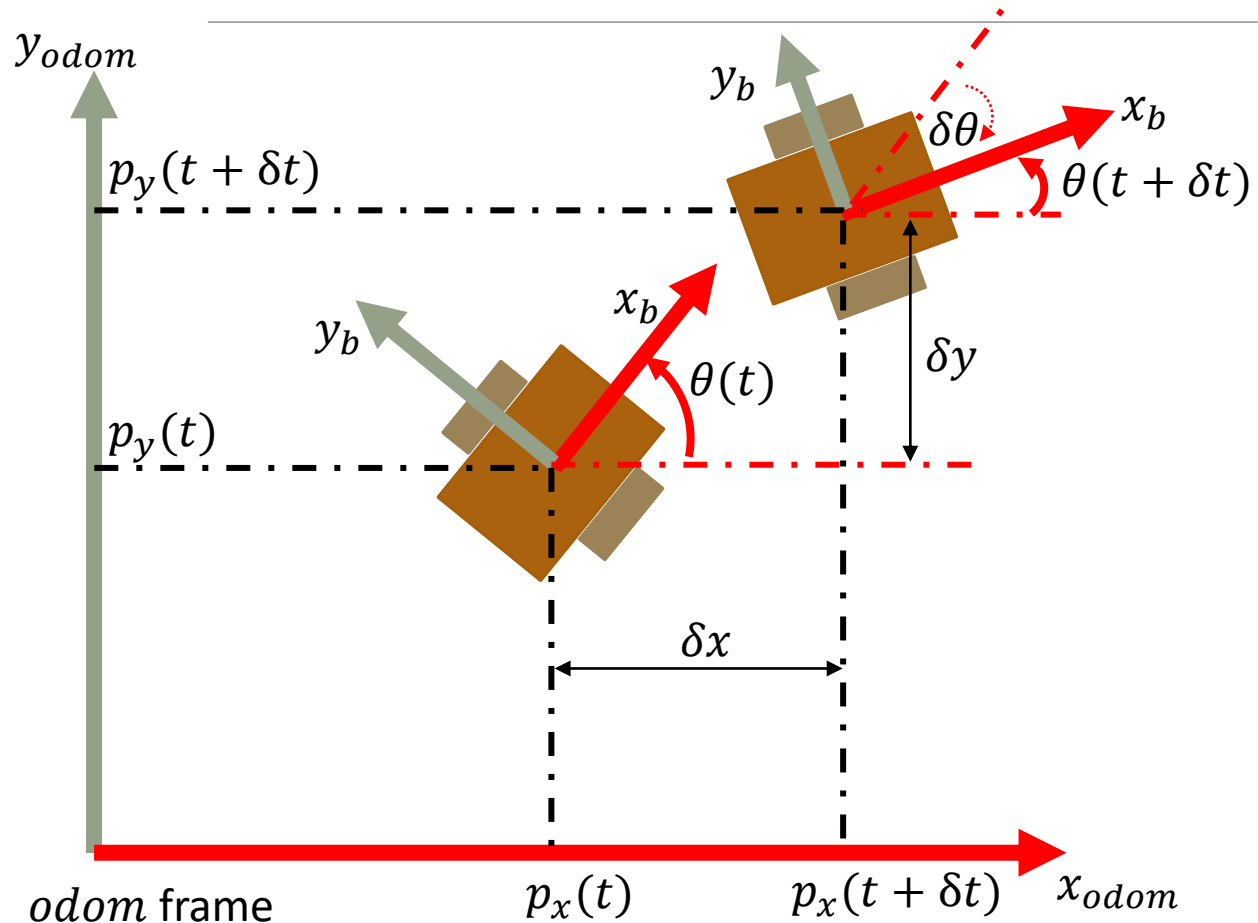
- หาความเร็วเชิงเส้นและเชิงมุมของหุ่นยนต์จาก
การเคลื่อนที่ของล้อ
- หาระยะทางในการเคลื่อนที่ ณ ขณะนั้น
- หาผลรวมระยะทางการเคลื่อนที่ทั้งหมด

Forward Kinematics สำหรับการสร้าง Wheel Odometry



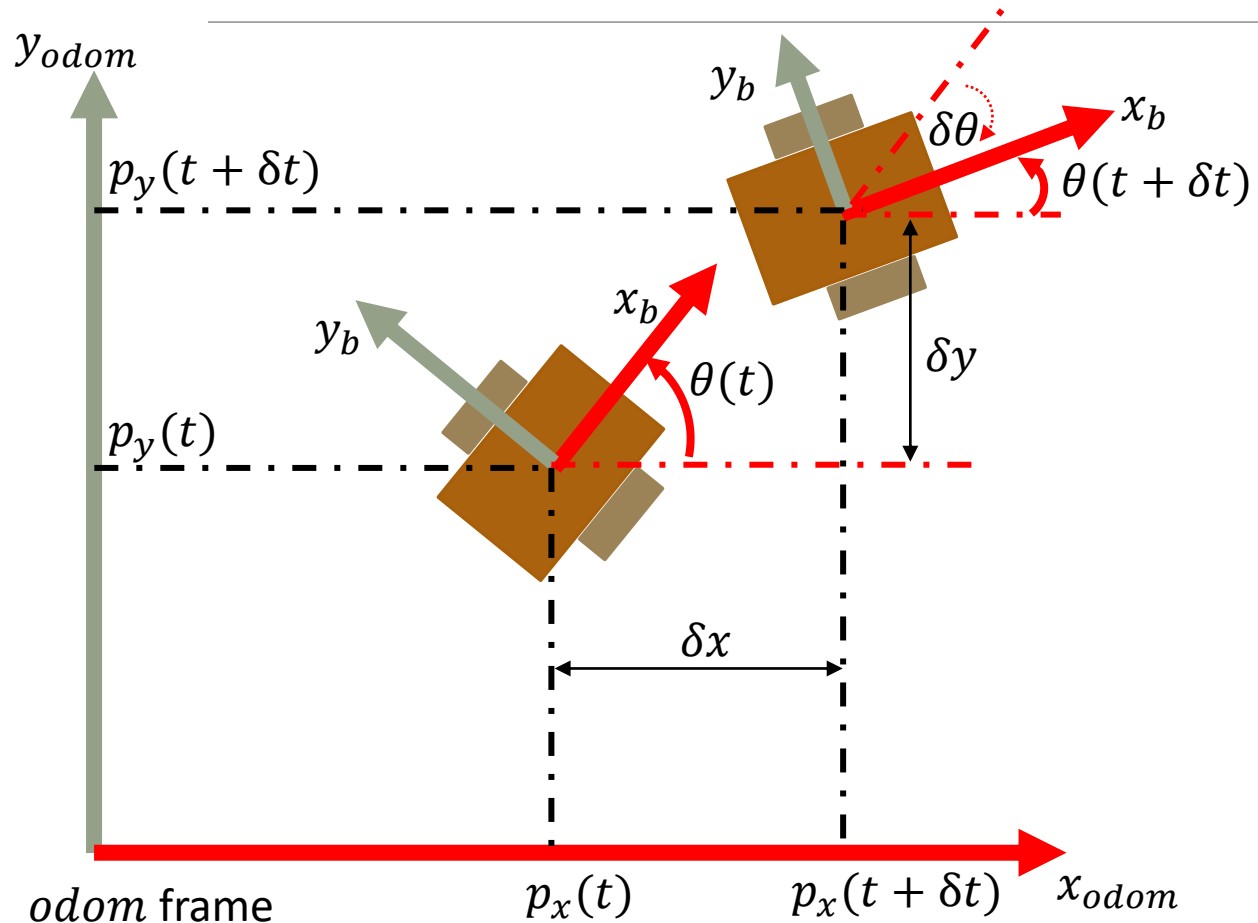
- หาความเร็วเชิงเส้นและเชิงมุมของหุ่นยนต์จากการเคลื่อนที่ของล้อ
- $$v_x = \frac{v_l + v_r}{2} = \frac{(\omega_l + \omega_r)r}{2}$$
- $$v_y = 0$$
- $$\omega = \frac{(\omega_r - \omega_l)r}{b}$$

Forward Kinematics สำหรับการสร้าง Wheel Odometry



- หาระยะทางในการเคลื่อนที่ ณ ขณะนั้น
- $\delta x = v_x \cos(\theta) \delta t$
- $\delta y = v_x \sin(\theta) \delta t$
- $\delta \theta = \omega \delta t$

Forward Kinematics สำหรับการสร้าง Wheel Odometry



- หาผลรวมระยะทางการเคลื่อนที่ทั้งหมด
- $p_x(t + \delta t) = p_x(t) + \delta x$
- $p_y(t + \delta t) = p_y(t) + \delta y$
- $\theta(t + \delta t) = \theta(t) + \delta \theta$

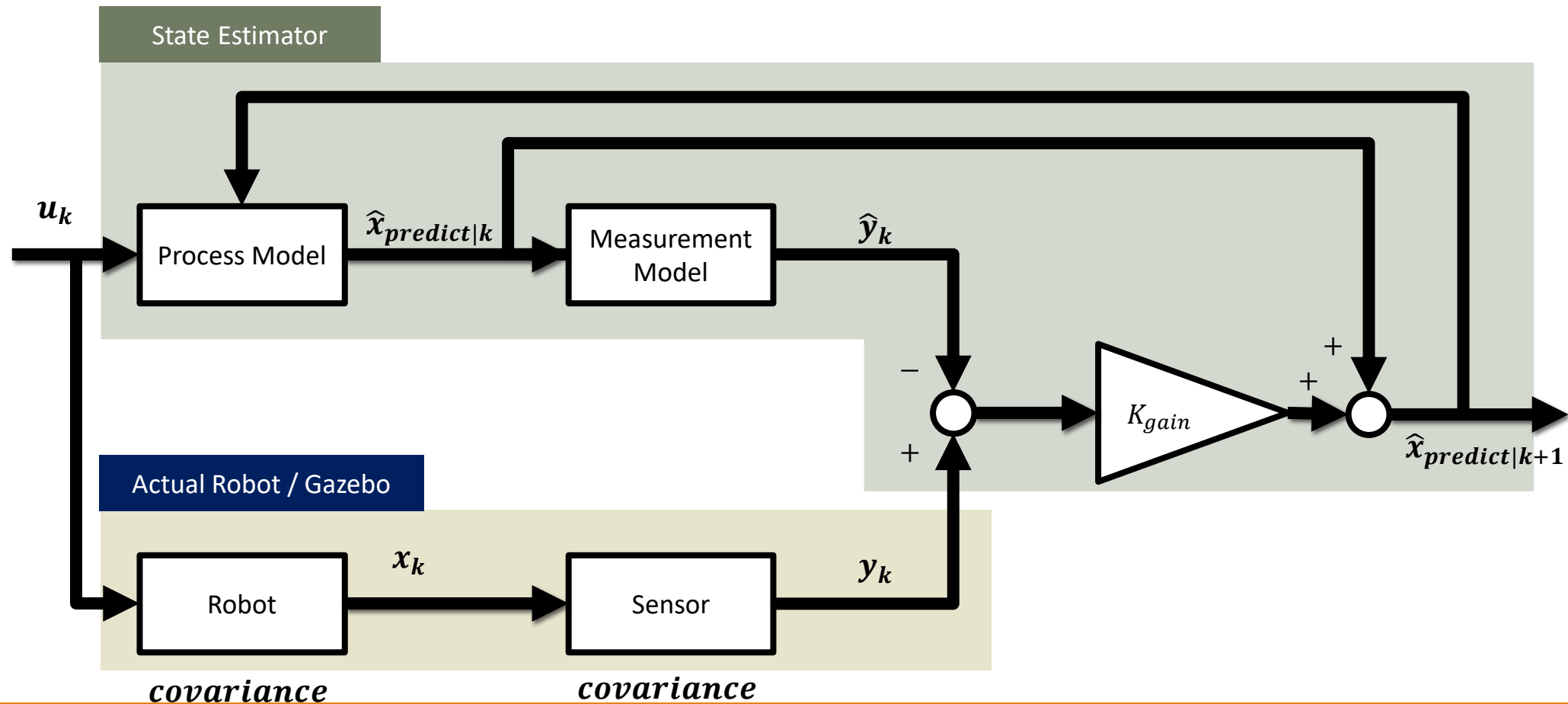
แบบฝึกหัดที่ 1 สร้าง Node สำหรับสร้าง Wheel Odometry

1. แก้ไขไฟล์ `wheel_odom.py` ใน `xxx_control/scripts`
2. แก้ไขไฟล์ `display_gazebo.launch.py`
3. ทดลองเคลื่อนที่หุ่นยนต์ใน Gazebo

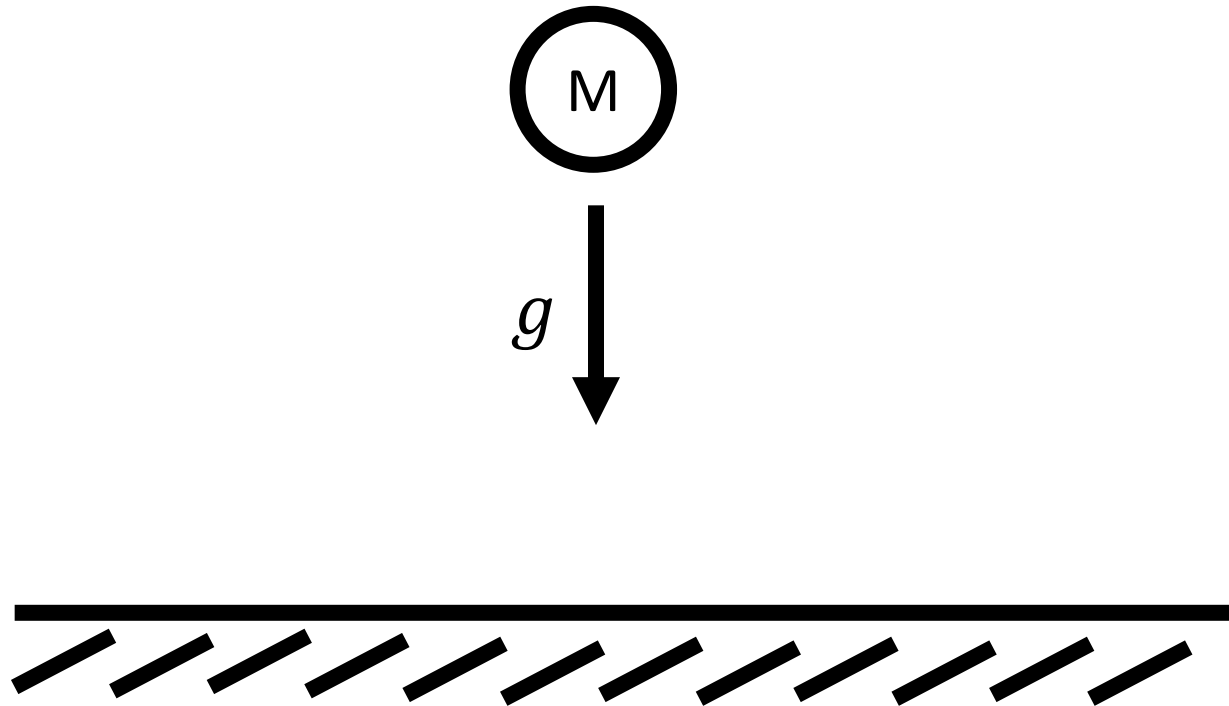
ความไม่แน่นอนที่เกิดขึ้นกับการเคลื่อนที่ของหุ่นยนต์

- สภาพแวดล้อมภายในและภายนอกของหุ่นยนต์
 - แรงเสียดทาน
 - น้ำหนัก
 - พื้นไม่เรียบ
- คุณภาพและความน่าเชื่อถือของ **sensor**
- ระดับแรงดันในแบตเตอรี่
- อื่นๆ

การประมาณสถานะของหุ่นยนต์



Kalman Filter (Free Fall Motion)

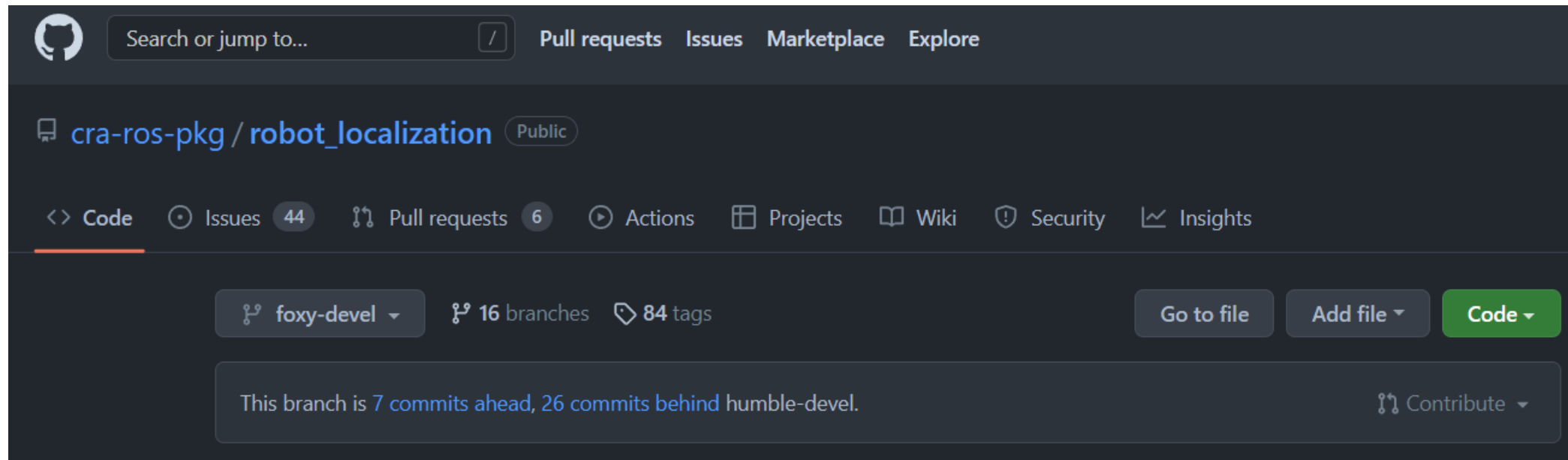


สามารถศึกษาเพิ่มเติม : การประมาณค่าสถานะ

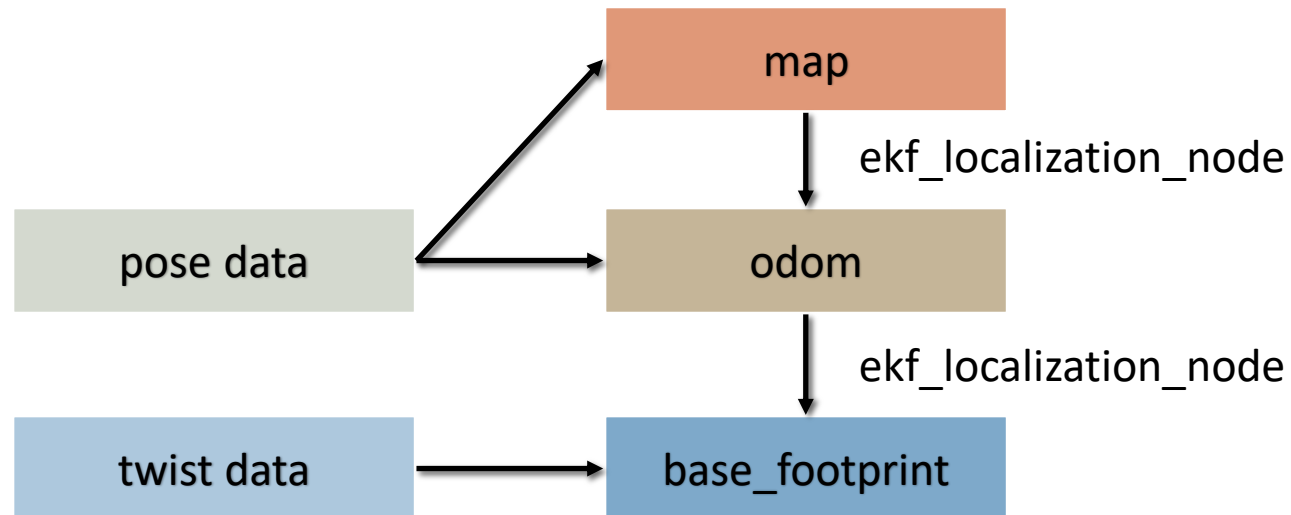
<https://www.youtube.com/watch?v=sELYOzbYpUg&list=PLHnZZE5j6C67lhKC6X6JRGNGhQCtTtHu&index=1>

robot_localization

```
sudo apt-get install ros-foxy-robot-localization
```



Preparing Your Sensor Data: Transforms



*https://roscon.ros.org/2015/presentations/robot_localization.pdf

Configuring kf_localization_node

Coordinate frame specification

```
map_frame: map           # Defaults to "map" if unspecified
odom_frame: odom         # Defaults to "odom" if unspecified
base_link_frame: base_footprint # Defaults to "base_link" if unspecified
world_frame: odom        # Defaults to the value of odom_frame if unspecified
```

Input specification

```
odom0: wheel/odom
imu0: imu
```

Basic input configuration

```
odom0_config: [false, false, false,   # x y z
               false, false, false,   # roll pitch yaw
               true,  true,  false,    # vx vy vz
               false, false, true,     # vroll vpitch vyaw
               false, false, false]    # ax ay az
```

*https://roscon.ros.org/2015/presentations/robot_localization.pdf

Configuring kf_localization_node

Covariance specification

```
process_noise_covariance: [0.05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
initial_estimate_covariance 0.0, 0.05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.06, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.06, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.025, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.025, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.015]
```

x, y, z, roll, pitch, yaw, vx, vy, vz, vroll, vpitch, vyaw, ax, ay, az

*https://roscon.ros.org/2015/presentations/robot_localization.pdf

แบบฝึกหัดที่ 2 สร้าง Node สำหรับสร้าง Wheel Odometry

1. แก้ไขไฟล์ `robot_ekf.launch.py` ใน `xxx_gazebo/launch`
2. แก้ไขไฟล์ `ekf.yaml` ใน `xxx_description/config`
3. ทดลองเคลื่อนที่หุ่นยนต์ใน Gazebo

Wheel Odometry VS EKF (robot_localization)