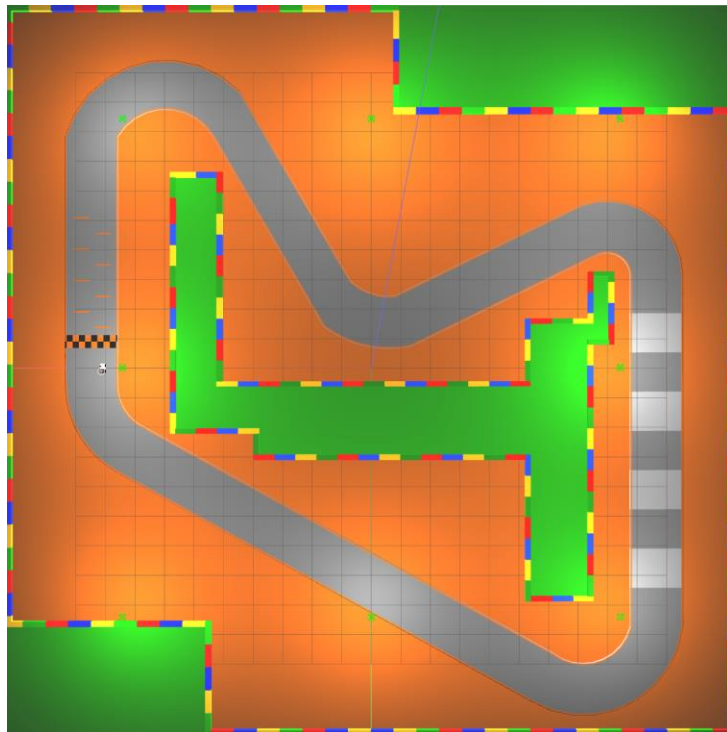


## Lab 1: ยำรวมมิตร (กลุ่มละ 2 คน)

### Overview

- Lab 1.1 Kinematics of Mobile Robot
  - คำนวณ wheel odometry จากการ teleoperation หุ่นยนต์
- Lab 1.2 Mobile Robot Control
  - Navigation หุ่นยนต์รอบ Map โดยใช้ Controller ต่าง ๆ (PID, Pure Pursuit, MPC)
- Lab 1.3 State Estimator
  - เพื่อทำ Sensor Fusion ในการหาตำแหน่งของหุ่นยนต์

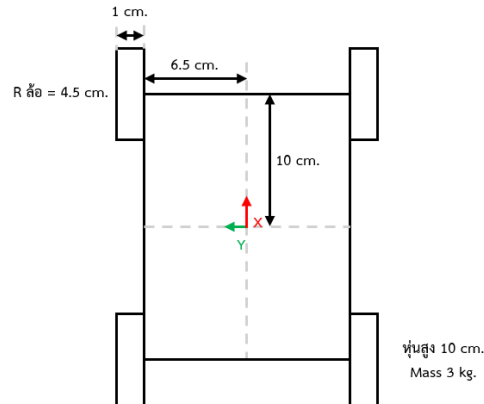
### Map



## Lab 1.1 Kinematics of Mobile Robot

### 1. Model Ackermann Steering

- สร้าง ROS2 Package สำหรับ robot\_description เพื่อเขียน URDF ของหุ่นยนต์
- ในไฟล์ URDF กำหนดขนาดของหุ่นยนต์ดังนี้



- หลังจากสร้างไฟล์ URDF เสร็จแล้วให้ import หุ่นไปยัง GAZEBO



### 2. Inverse Kinematics

- เขียน ROS2 Node สำหรับการคำนวณหา Inverse Kinematics โดยรับค่า cmd\_vel และแปลงเป็นความเร็วล้อของหุ่นยนต์ (wheel speed)
  - ใช้ Basic Model (steer angle ของทั้ง 2 ล้อเท่ากัน) [Link สำหรับศึกษาเพิ่มเติม: <https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Control/BicycleModel.html>]
  - กำหนดเงื่อนไขแบบ No Slip condition constraints [Link สำหรับศึกษาเพิ่มเติม: <https://www.mathworks.com/help/vdynblks/ref/kinematicsteering.html>]

### 3. Forward Kinematics

- เขียน ROS2 Node สำหรับการคำนวณหา Forward Kinematics เพื่อแปลงค่า ความเร็วล้อของหุ่นยนต์ (wheel speed) เป็นค่า odometry โดยใช้ 3 Model ในการคำนวณคือ
  - Yaw rate
  - Single-track model
  - Double-track model

[Link สำหรับศึกษาเพิ่มเติม: <https://ieeexplore.ieee.org/document/8574906>]

(ถ้าเจอที่อ่านง่ายกว่านี้บอกอ. ด้วย)

#### 4. Validation

- ทำการ Validate ค่า odometry ที่คำนวณได้ โดยการเปรียบเทียบกับค่า pose ใน GAZEBO
- เปรียบเทียบผลลัพธ์ของ Model ทั้ง 3 แบบ