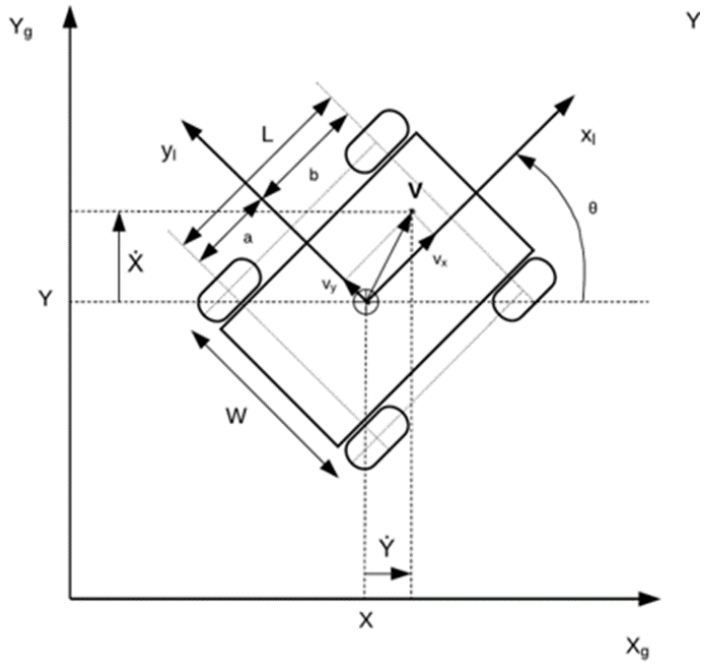
Objective:

Understand and implement different state estimation methods for mobile robots. Students will apply Bayes filters, Kalman Filter (KF), Extended KF (EKF), Unscented KF (UKF) in simulation, integrating ROS2 and Gazebo.

Theorical

Motion model

Derive motion model of 4 wheeled robot (skid_steering). consider each side has the same speed for wheels.



parameters

vx	wz	r	w	x	y	θ	Δt
Linear velocity	Angular velocity	Wheel radius	Wheel seperation	state	state	state	time

Measurement Model

Consider the robot has an rgbd camera with imu. We have developed an algorithm that drives visual odometry ($\Delta x, \Delta y$) and imu orientation (θ). Derive measurement model from these two sources.

Extended kalman filter

Based on previous questions, design an EKF based on motion model and measurement model. Write prediction update formulas and measurement updates formulas.

Implementation

a. Clone package of ta session github. Add 2 motors instead of diff_drive plugin in the urdf. The robot should be controllable with giving rpm to each motor.
b. Add noise to imu in urdf.
c. Write controller node in c++ for converting cmd_vel to motor commands in same package
d. Create a folder in src called robot_autonomy. In that folder create a new package called robot_local_localization. Create a prediction_node.py file and implement prediction updates in python. Add parameters to the file like wheel_seperation, wheel_radius, cmd_vel_topic and any other parameters needed.
e. Create a measurement_node.py. In that file it should subscribe to imu and visual odometry topics (An open-source algorithm will be provided for you to use) and Compute combined measurement model formulas and publish them.
f. Create an ekf_node.py. In that file it subscribes to motion_model and measurement model and implements ekf. Output should be odometry message type. Also add a listener so you can see the result in rviz.
g. Implement a test_node.py which subscribes to ekf_odometry and gives robot commands to follow a rectangular path. Then plot real_path, measurement path, motion model path, ekf path in rviz. Use a Path message to do that.
h. Use rosbag + foxglove or lichtblick to plot part g. (bonus)