1. Differential Drive Mobile Robot Labs

1.1 Pose Compounding

Goal 1.1.1 This exercise explains how to define a robot pose in 3 Degree Of Freedom (DOF) as well as how to compose different poses using the direct and the inverse compounding operations.

Download the the *lab_compounding.ipynb* Jupiter Python Notebook and follow it.

1.2 Robot Simulation

Goal 1.2.1 In this lab we will learn how to simulate a 3 DOF Differential Drive Mobile Robot

1. ***SimulatedRobot* base class**:
   (a) Read its documentation in the *"prpy: Probabilistic Robot Localization Python Library"* document.
   (b) Check the provided code of the *RobotSimulation.py* file.
2. ***DifferentialDriveSimulatedRobot* class**:
   (a) Check the uncompleted code provided for this class in the *DifferentialDriveSimulatedRobot.py* file.
   (b) Complete the code of the class by implementing the methods labeled as *"# Todo: To be completed by the student"*
3. **Test the Robot simulation**:
   (a) **Circular Trajectory**: program the robot simulation to perform a circular shaped trajectory.
   (b) **Eight Shaped Trajectory**: program the robot simulation to perform a trajectory shaped as an "8".

1.3 Dead Reckoning

Goal 1.3.1 In this lab we will solve the Dead Reckoning Localization of a 3 DOF Differential Drive Mobile Robot

1. ***Localization* base class**:
   (a) Read its documentation in the *"prpy: Probabilistic Robot Localization Python Library"* document.
   (b) Check the provided code of the *Localization.py* file.
2. ***DR_3DOFDifferentialDrive* class**:
   (a) Check the uncompleted code provided for this class in the *DR_3DOFDifferentialDrive.py* file.
   (b) Complete the code of the class by implementing the methods labeled as *"# Todo: To be completed by the student"*
3. **Test the Robot Dead Reckoning Localization**:
   (a) **Circular Trajectory**: Check the localization results of the implemented Dead REckoning algorithm.

1.4 EKF Localization using a Displacement Motion Model

1.5 EKF Localization using a Constant Velocity Model