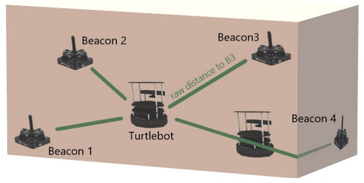
Ultrasound Localization for Cooperative Turtlebot Control

University of California San Diego
Multi-Agent Robotics Laboratory
Natalia Edith Mejia Bautista and Anthony Hirales Ahuatzin

UC San Diego
JACOBS SCHOOL OF ENGINEERING

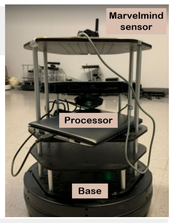
INTRODUCTION

The project aims to design and implement autonomous motion plans to coordinate turtlebot movements. The robots positions are estimated using noisy range measurements from fixed ultrasound emitting beacons. Particle filtering is employed to estimate robot positions and PID control is implemented to track given trajectory.



Turtlebot

A two wheeled research robot, that has a differential drive and works with assistance of a laptop using ROS and Linux, via Python or C++ codes.

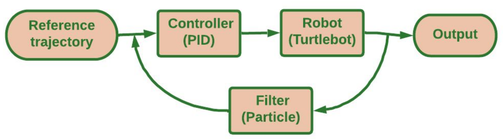


Marvelmind

The indoor positioning system obtains real-time distances between the static beacons in the laboratory and mobile ones on the robots.

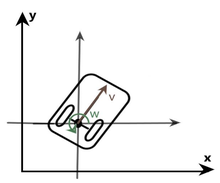
Feedback System Model

The following diagram shows the relationships between the robot control and the particle filter, that it needs to follow a given trajectory.



Plant Model Dynamics

Turtlebot movement is modelled with unicycle dynamics. It is a nonholonomic system. Robot configuration in a 2-D plane consists of x-y positions and heading (x1,x2,θ).



Equations of motion:

$$\dot{x}_1 = v\cos\theta$$
$$\dot{x}_2 = v\sin\theta$$
$$\dot{\theta} = \omega$$

The variables v and ω are the control inputs: linear and angular velocity, respectively.

Path Planning

The robot's trajectory can be selected to optimize several criteria like, path length, energy, time. Trajectory was divided into waypoints, and filtering and controls are used to reach them.

METHODOLOGY

Filtering

Filtering estimates the robot states using the noisy range data computed from Ultrasound waves between static and mobile beacons. Linear filtering methods such as Kalman and Extended Kalman Filter apply to systems that are close to linear. Since the range data and robot system are nonlinear, we work with another bayesian filtering method, called **Particle Filter**:

1. Create **particles** to randomly locate the robot.
2. Filter
   a. **Predict** robot state by applying control inputs to unicycle motion model

$$\mu_{t+1|t}^{(k)} \sim p_f\left(\cdot \mid \bar{\mu}_{t|t}^{(k)}, u_t\right) \qquad p_{t+1|t}(\mathbf{x}) \approx \sum_{k=1}^{N} \alpha_{t+1|t}^{(k)} \delta\left(\mathbf{x} - \mu_{t+1|t}^{(k)}\right)$$

   b. **Update** state estimate using measurement model p_h (observation model).

$$p_{t+1|t+1}(\mathbf{x}) = \sum_{k=1}^{N}\left[\frac{\alpha_{t+1|t}^{(k)} p_h\left(\mathbf{z}_{t+1} \mid \mu_{t+1|t}^{(k)}\right)}{\sum_{j=1}^{N_{t+1|t}} \alpha_{t+1|t}^{(j)} p_h\left(\mathbf{z}_{t+1} \mid \mu_{t+1|t}^{(j)}\right)}\right] \delta\left(\mathbf{x} - \mu_{t+1|t}^{(k)}\right)$$

where μ is the probability density that represents th $p_h\left(\mathbf{z}_{t+1} \mid \mu_{t+1|t}^{(k)}\right)$ robot, p_f is the probabilistic dynamics model and is the observation model.

   c. **Stratified resampling**. Sample particles from the updated pdf estimate which improves computational complexity in this step.
3. Obtain an x_t **estimated state**.

$$\mathbf{x}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim p_{t|t}(\mathbf{x}_t) := \sum_{k=1}^{N} \alpha_{t|t}^{(k)} \delta\left(\mathbf{x}_t; \mu_{t|t}^{(k)}\right)$$

Control

Point to point motion. PID Control

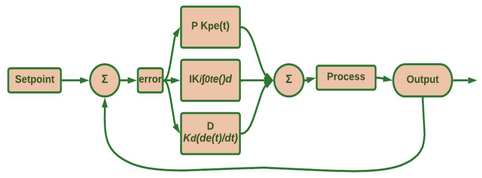
$$u_t = K_P \cdot e(t) + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t) + K_I \int e(\tau) d\tau$$

Where

- Error e(t) = goal(x_ref)-current estimated position(x_hat)
- K_P is the proportional gain
- K_I is the integral gain
- K_D is the derivative gain

The proportional control is commensurate to the error. Derivative control uses rate of change in error to dampen the control input as the turtlebot approaches the goal. This helps the bot from shooting the desired reference points. Integral control reduces the sum of error over large time duration.

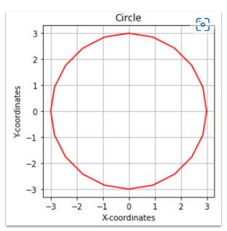
The gains were obtained by trial and error in the simulations, and further tuning is needed to control the turtlebots in a real scenario.



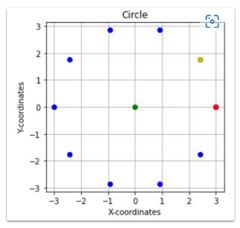
Simulation

Parameters used in this simulation are:

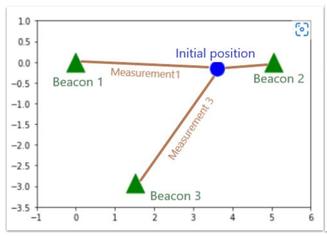
- Timestep dt=0.01 seconds
- Maximum of loops per goal K=300
- Number of particles M=100
- Number of beacons NB=3
- Number of waypoints NG=10



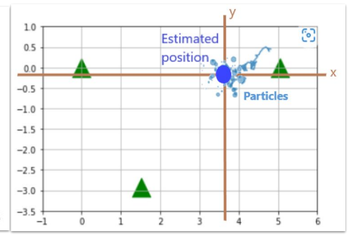
Determine trajectory



Dividing the trajectory into various goals



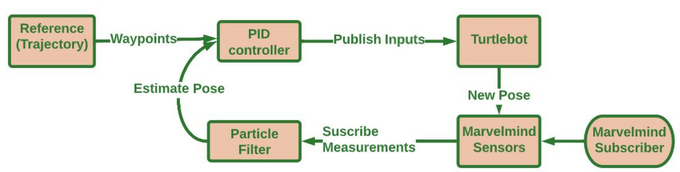
Get the initial measurement of the start point



Estimate position with particle filter

Implementation

Initialize Marvelmind subscribers
Initialize particle filter
Initialize turtlebot control publisher
While (Robot not near the goal){
  Retrieve data from Marvelmind beacons
  Get raw distances
  Process raw distances via particle filter
  Input estimated state to the PID controller
  Publish data to the turtlebot
}



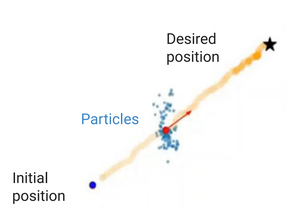
Results

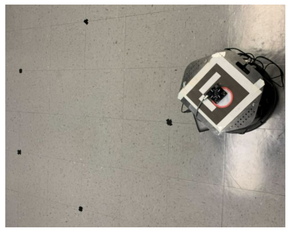
Simulate Turtlebot with unicycle model and range measurements.

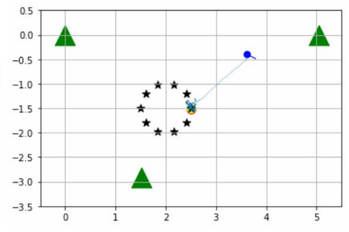
The feedback-loop sequence is as follows:

A. Estimate robot location with particle filter.
B. Generate inputs with PID control using estimated location and goal.
C. Publish movement commands to the robot.
D. Measure beacon ranges to estimate the robot state.

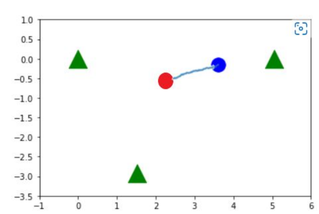
Repeat (A-D) until reaching the goal and continue to other waypoints.







Matching simulation to hardware: The control inputs and measurements from a turtlebot run are matched to the unicycle simulation.



Simulation

True position

Conclusions

- Tested extended Kalman filter for linearized unicycle model.
- Implemented PID controller using the particle estimates for the turtlebot.
- Implemented a robust particle filter handling non-linearities in unicycle model and range measurements.
- Future work: Coordinating the motion of several turtlebots and drones.

Bibliography

- Wescott, Tim (2006). *Applied Control Theory for Embedded Systems*. Elsevier/Newnes. Section 1.2 (Anatomy of a Control System). ISBN 978-0-7506-7839-1 – via Google Books.
- L. Sciavicco and B. Siciliano, Modelling and Control of Robot Manipulators, Springer, 2000.

Acknowledgements

The authors want to thank Parth, Scott and Brandon and Pamica for helping and mentoring us on this project. We are thankful to Prof. Jorge Cortes, Prof. Sonia Martinez, Prof. Olivia Graeve and the ENLACE Program by providing us with this wonderful opportunity.