Sound Source Localization and SLAM

Yan Wang, Dian Yuan, Kaustabh Paul
TAS Project 22WS Group6
Supervisor:
Dr.-Ing. habil. Dirk Wollherr
M.Sc. Salman Bari
M.Sc. Tommaso Benciolini

Chair of Automatic Control Engineering
Technical University of Munich

Introduction

- Locate and reach the sound sources in the unknown environment
- Build the map with SLAM algorithm

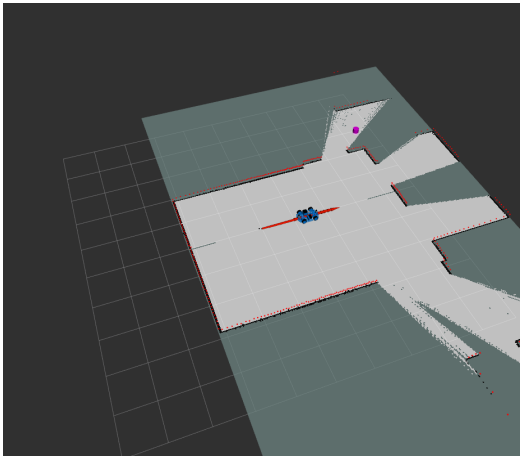


Figure: Environment in RVIZ

Project Structure

The following packages are mainly used in this project:

- **signal generator package**: generate the position of the signal source and publish the signal info.
- **planner**: locate the sound source and the behavior state machine of the car.
- **slam_tool_box**: localization and create the map.
- **nav2**: navigate the car to desired pose.
- **Lscan**: own developed collision detection and map exploring node.

Signal setup

contributor: Yan Wang

Our assumptions:

- Signal Strength attenuation curve is an inverse proportional function
- Won't be affected by the obstacles

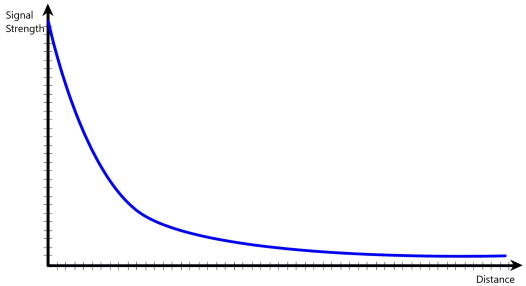


Figure: Signal strength attenuation curve

Sensor setup

contributor: Yan Wang

Our assumptions:

- Virtual sensors
- On both the side of TAS car
- Ability to detect sound strength and the time difference between sound arriving at different sensors

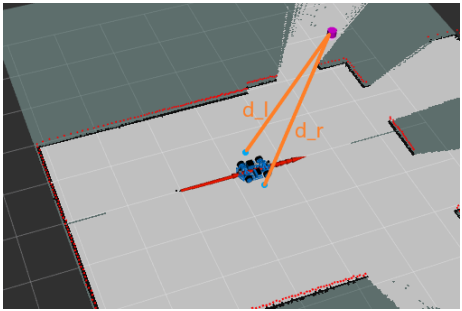
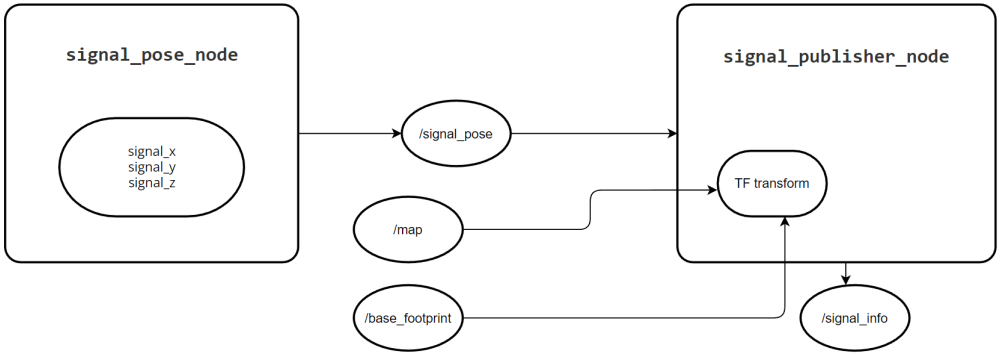


Figure: Virtual sensors

Package structure

contributor: Yan Wang



Random signal source

contributor: Yan Wang

- Valid positions should have a value of 254
- Problem: signal source could be outside the range of the current map

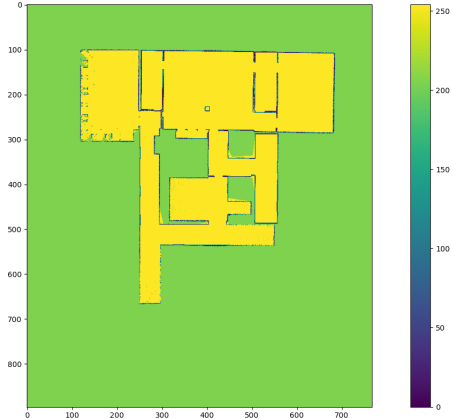
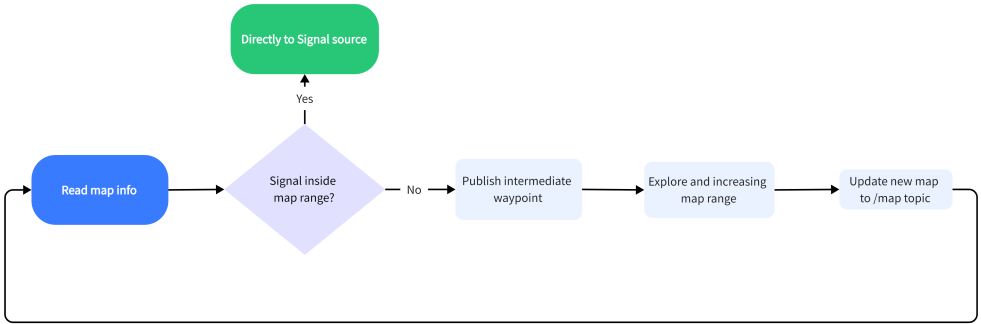


Figure: Pgm file of the basement

Exploration

contributor: Yan Wang



Evaluation

contributor: Yan Wang

- Problem 1: The signal generator could generate some random signal sources that are very near to obstacle.

  Possible solution: generate the random signal source based on the cost map.

- Problem 2: The exploration algorithm is not efficient enough.

  Possible solution: Frontier based exploration algorithm.

Source Localization

contributor: Dian Yuan

TDoA(Time Difference of Arrival)[a]

- From the time difference of the sound arrival derives the potential hyperbolic trajectory of the source
- Two potential sound locations are calculated from the sound strength
- Determine the unique signal source location based on intensity variations at different positions

[a]Weiwei Cui et al. *Dual-Microphone Source Location Method in 2D Space*

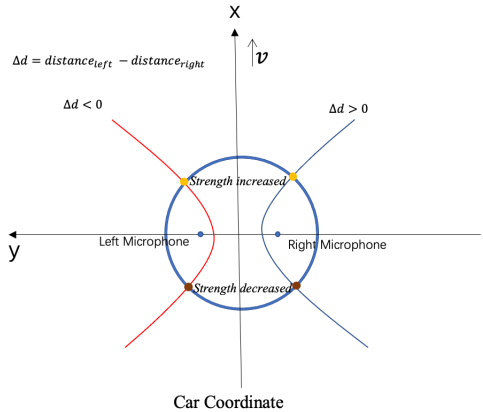
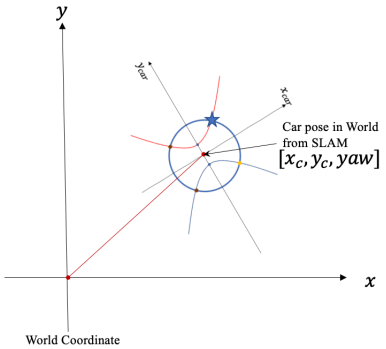


Figure: Principle of sound source localization

Source Localization

contributor: Dian Yuan



- distance between two microphones: d
- estimated sound source distance: r
- sound speed: c

Example:

- time difference of arriving microphones $\Delta t = t_{left} - t_{right} < 0$
- sound strength are increasing

i.e. select the result of the 1st quadrant under the car coordinate as ★ in the figure.

$$ {}_cx_s = \frac{1}{4d}\sqrt{(4r^2 - (c\Delta t)^2)(4d^2 - (c\Delta t)^2)} $$

$$ {}_cy_s = \sqrt{r^2 - {}_cx_s{}^2} $$

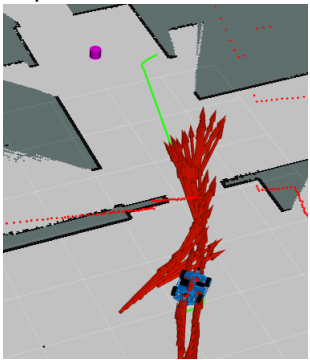
$$ x_s = x_c +_c x_s \cos yaw -_c y_s \sin yaw $$

$$ y_s = y_c +_c x_s \sin yaw +_c y_s \cos yaw $$

Evaluation

contributor: Dian Yuan

The accuracy of the sound source depends heavily on the degree of accuracy of localization, especially when the signal source is far away, orientation plays a very important role. Some time the sound localization may not accurate enough.



Solution: repositioning of the sound source in the following cases

- the navigation task ends but the sound source is not found
- the received sound strength is continuously decreasing
- over a period of time

State Machine

contributor: Dian Yuan

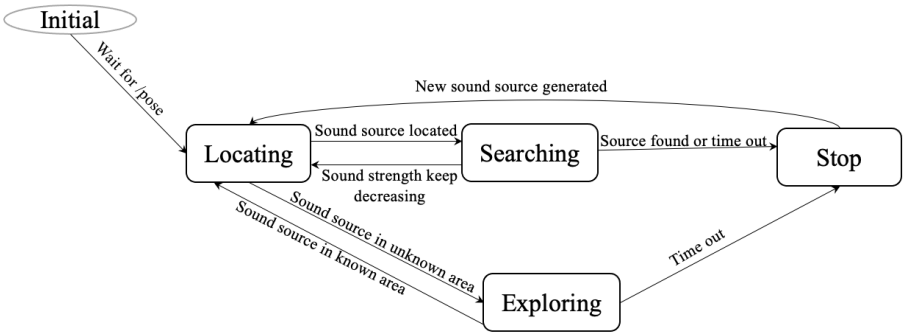


Figure: Behavior state machine of TAS car

SLAM

contributor: Kaustabh Paul

SLAM Package: Slam toolbox
Method: Graph based SLAM
Builds pose graph while trying to find a node configuration that minimizes error introduced through constraints

- Node in graph corresponds to pose during mapping
- New scans used to generate edges ⇒ correspond to spatial constraint between nodes

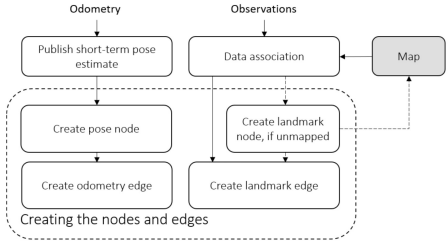


Figure: Pose graph building process

SLAM

contributor: Kaustabh Paul

Three available mapping modes:

- Synchronous mode (offline)
  - Estimates all poses $\rightarrow$ slow
- Synchronous mode (online)
  - All scan measurements processed
- Asynchronous mode (online)
  - Processes new measurement after current completed

$\Rightarrow$ asynchronous mode used

SLAM

contributor: Kaustabh Paul

original map (left) compared to SLAM generated map (right)

Nav2

contributor: Kaustabh Paul

Goal: reach points in unknown environment without colliding

- Nav2 used without nav2-amcl since SLAM toolbox localizes already
- Regualted Pure Pursuit controller used
- Reversing enabled since stationary rotations cause collisions $\Rightarrow$ lidar range changed to 360° to build map when moving backwards

Map Exploration

contributor: Kaustabh Paul

- Subscribes to scan topic and extracts distances from multiple sides of car
- Publishes twist to cmd-vel topic

Algorithm:

- Car moves forward until obstacle detected closer than treshold distance
- Rotates to avoid obstacle
- Constantly conducts collision checks in case car get too close to obstacle

Evaluation

contributor: Kaustabh Paul

After some time the amount of errors in the map increase drastically
- Localization error needs to be reduced

Collisions sometimes occur during navigation
- Alternative collision monitor needed

Map exploration can only create partial map
- Could be improved by following walls or visiting unknown areas

Summary and Conclusion

What we achieved

- Generate virtual sound distribution model and microphone sensor model on TAS car.
- Locate the sound source in the environment with TDoA algorithm and navigate the car to the source.
- Grid-based SLAM to create the map while navigating in the environment.
- Basic algorithm to autonomously explore the map.

Summary and Conclusion

To be improved

- A more complex and realistic signal distribution and generation model.
- A real-time sound source localization algorithm.
- A more efficient map exploring algorithm.
- An alternative collision monitor.
- Try other SLAM algorithms.