Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

An exploration of Simultaneous Localisation and Mapping using Lidar

Paul Mahoney – C00227807

Technical Manual

April 2021

Contents

Introduction

In this document, I will be highlighting importing files and scripts in my project. Additionally, I will outline any dependencies and install instructions.

Dependencies and technologies

Robotic operating system (ROS): ROS is a middleware that enabled me to develop this system via an array of packages. These packages enabled me to create a complex system without complex knowledge of how the hardware works. This was most prevalent with the Lidar mapping package Google cartographer.

Mattplotlib: This is quite a useful tool that allowed me to create an animation to showcase how the lidar works.

Tkinter: is a technology that allowed me to create a GUI application around my project.

OS: This is a library that allowed me to execute bash terminal commands within my python program.

Numpy: is a technology that I utilized in the calculations for my Matplotlib animation.

Doxygen: is a documentation tool that generates complex documents from comments created within the code.

Files and Scripts

Application.py

Obstacle avoidance: This function is used for obstacle avoidance, It uses Skoltechs RPlidar driver to implement the wall follower algorithm discussed in my documentation. A for loop takes in an iterator being four measurements, new scan, quality, angle, and distance. I ensure only good data is being read in with an if statement. If quality and distance are greater than 0 no issues exist. Null distance values are read as 0. Following this, a series of conditional statements enact the wall follower algorithm. Constantly directly ahead, left and right of the lidar is checked for obstacles. If something arises a text box is populated with the relevant navigational instruction such as "Pivot left", "Pivot right" etc. This is how obstacle avoidance was achieved.

resetLidar: This function is used to allow the user to reset the lidar. After Obstacle avoidance and SLAM mapping are performed the lidar tends to keep spinning. I implemented this function to allow the user to manually turn it off. This is done quite simply by calling a series of lidar stop commands included in the rplidar driver for python. I also populate the text window with "Resetting!" to notify the user of the command's success.

Animate: This function creates a matplotlib animation to provide the user with a visual representation of what is happening when the lidar "scans". I had initially intended for this function to be embedded in the canvas, but the graph was quite buggy and would not update so I opted to leave it as a separate window. This is an example provided in Skoltech documentation and was adapted to fit into my application.

updateLine: This function is the "animation" function for my animation. It updates the line on the graph with new measurements.

```
#!/usr/bin/env python
# ------ Start of imports
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import numpy as np
import tkinter as tk
import threading
import os
from rplidar import RPLidar
from time import sleep
from tkinter import *
from matplotlib.figure import Figure
# ------ End of imports
"""@package docstring
This file is the
"""

# -- Getting permission for the port
PORT_NAME = "/dev/ttyUSB0"
os.system("sudo chmod 666 /dev/ttyUSB0")

# -- Setting variables for the visualization
DMAX = 4000
IMIN = 0
IMAX = 50


def Obstacle():
    lidar = RPLidar(PORT_NAME)

    print("Recording elements")

    try:
        right = range(85, 95)
        forward = range(0,10)
        scan = []
        iterator = lidar.iter_measurments()
        for new_scan, quality, angle, distance in iterator:
            if quality > 0 and distance > 0:
                if angle >= 0 and angle <= 10:
                    if distance < 300:
                        text.delete(1.0, END)
                        text.insert(END, "Object ahead! Turn right!")
                        window.after(100, window.update_idletasks())
                elif angle >= 85 and angle <= 95:
                    if distance < 300:
                        text.delete(1.0, END)
                        text.insert(END, "Pivot Left!")
                        window.after(100, window.update_idletasks())
                elif angle >= 265 and angle <= 275:
                    if distance < 300:
                        text.delete(1.0, END)
                        text.insert(END, "Pivot Right!")
                        window.after(100, window.update_idletasks())

    except KeyboardInterrupt:
        text.delete(1.0, END)
        text.insert(END, "Stopping!")
        window.after(100, window.update_idletasks())
        lidar.stop()
        lidar.stop_motor()
        lidar.disconnect()

def SLAM():
    os.system("gnome-terminal -x ./term1.sh")

```

```python
def resetLidar():
    lidar = RPLidar(PORT_NAME)
    text.delete(1.0, END)
    text.insert(END, "Resetting!")
    window.after(500, window.update_idletasks())
    print("Stoping.")
    lidar.stop_motor()
    lidar.stop()
    lidar.disconnect()
    lidar.reset()


def update_line(num, iterator, line):
    scan = next(iterator)
    offsets = np.array([(np.radians(meas[1]), meas[2]) for meas in scan])
    line.set_offsets(offsets)
    intens = np.array([meas[0] for meas in scan])
    line.set_array(intens)

    return (line,)


def animate():
    ##Place plot to canvas
    lidar = RPLidar(PORT_NAME)
    fig = plt.figure("Lidar Scan", figsize=(5, 5))
    ax = plt.subplot(111, projection="polar")
    line = ax.scatter([0, 0], [0, 0], s=5, c=[IMIN, IMAX], cmap=plt.cm.Greys_r, lw=0)

    ax.set_rmax(DMAX)
    ax.grid(True)

    iterator = lidar.iter_scans()
    ani = animation.FuncAnimation(fig, update_line, fargs=(iterator, line), interval=50)

    plt.show()
# ------ End of functions


# ------ Start of Tkinter
# -- Create the Tkinter Window and set its size and title
window = tk.Tk()
window.geometry("1920x1080")
window.title("Lidar Application")
window.resizable()
window.columnconfigure(0, weight=2)
window.columnconfigure(1, weight=1)

# -- Create the frame for the buttons etc
btns_frame = Frame(window, width=312, height=272.5)
btns_frame.grid(sticky=NE, row=0, column=1, padx=20, pady=20)

# -- Create a textbox
text = tk.Text(btns_frame, height=4, width=50)
text.pack()
text.insert(END, "Navigation Instructions will appear here")

# -- Create the navigation button
obstacleAvoidance = Button(
    btns_frame,
    text="Start obstacle avoidance",
    fg="black",
    width=47,
    height=4,
    command=lambda: Obstacle(),
)
```

```
obstacleAvoidance.pack()

# -- Create the button to launch SLAM
buttonSLAM = Button(
    btns_frame,
    text="Launch SLAM(Launches RVIZ)",
    fg="black",
    width=47,
    height=4,
    command=SLAM,
)
buttonSLAM.pack()

# -- Upload MAP
uploadMap = Button(
    btns_frame, text="Start animation", fg="black", width=47, height=4, command=animate
)
uploadMap.pack()

# -- Reset LIDAR
resetLidar = Button(
    btns_frame, text="Reset Lidar", fg="black", width=47, height=4, command=resetLidar
)
resetLidar.pack()

window.mainloop()
# ------ End of tkinter
```

Term1.sh

This function is used to launch my SLAM algorithm for mapping. It simply works by using the os.system command to create a new terminal and run a bash script. This bash script contains a series of terminal commands that opens ROS. It gives the new terminal access to the serial port where the lidar is located. It then changes the directory where the catkin workspace is located. It then sources the terminal to Catkin compiler and launches ROS.

```
#!/bin/bash
chmod 666 /dev/ttyUSB0
cd catkin_ws
source devel/setup.bash
roslaunch mapping mapping-l.launch
```

mapping-l.launch

This is the launch file for google cartographer. It essentially ties the functionalites of RPlidar and google cartographer together. It also launches a RVIZ terminal which the graphical interface of the mapping.

```
<launch>
  <param name="robot_description" textfile="$(find mapping)/urdf/head_2d.urdf" />

  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />

  <node name="rplidarNode" pkg="rplidar_ros" type="rplidarNode" output="screen">
    <param name="serial_port" type="string" value="/dev/ttyUSB0"/>
    <param name="serial_baudrate" type="int" value="115200"/>
    <param name="frame_id" type="string" value="laser"/>
    <param name="inverted" type="bool" value="false"/>
    <param name="angle_compensate" type="bool" value="true"/>
  </node>
```

```

    <node name="cartographer_node" pkg="cartographer_ros" type="cartographer_node" args="
        -configuration_directory
          $(find mapping)/configuration_files
        -configuration_basename mapping.lua" output="screen">
    </node>

    <node name="cartographer_occupancy_grid_node" pkg="cartographer_ros" type="cartographer_occupancy_grid_node" args="-resolution 0.05" />

  <node pkg="rviz" type="rviz" name="show_rviz" args="-d $(find mapping)/rviz/demo.rviz"/>
</launch>
```

Mapping.lua

This is the configuration file for Cartographer.

```
include "map_builder.lua"
include "trajectory_builder.lua"

options = {
  map_builder = MAP_BUILDER,
  trajectory_builder = TRAJECTORY_BUILDER,
  map_frame = "map",
  tracking_frame = "base_link",
  published_frame = "base_link",
  odom_frame = "odom",
  provide_odom_frame = true,
  publish_frame_projected_to_2d = true,
  use_odometry = false,
  use_nav_sat = false,
  use_landmarks = false,
  num_laser_scans = 1,
  num_multi_echo_laser_scans = 0,
  num_subdivisions_per_laser_scan = 1,
  num_point_clouds = 0,
  lookup_transform_timeout_sec = 0.2,
  submap_publish_period_sec = 0.3,
  pose_publish_period_sec = 5e-3,
  trajectory_publish_period_sec = 30e-3,
  rangefinder_sampling_ratio = 1.,
  odometry_sampling_ratio = 1.,
  fixed_frame_pose_sampling_ratio = 1.,
  imu_sampling_ratio = 1.,
  landmarks_sampling_ratio = 1.,
}

MAP_BUILDER.use_trajectory_builder_2d = true

TRAJECTORY_BUILDER_2D.min_range = 0.5
TRAJECTORY_BUILDER_2D.max_range = 8.
TRAJECTORY_BUILDER_2D.missing_data_ray_length = 8.5
TRAJECTORY_BUILDER_2D.use_imu_data = false
TRAJECTORY_BUILDER_2D.use_online_correlative_scan_matching = true
TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.linear_search_window = 0.1
TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.translation_delta_cost_weight = 10.
TRAJECTORY_BUILDER_2D.real_time_correlative_scan_matcher.rotation_delta_cost_weight = 1e-1
TRAJECTORY_BUILDER_2D.motion_filter.max_angle_radians = math.rad(0.2)
TRAJECTORY_BUILDER_2D.num_accumulated_range_data = 1

```

```
POSE_GRAPH.constraint_builder.min_score = 0.65
POSE_GRAPH.constraint_builder.global_localization_min_score = 0.65
POSE_GRAPH.optimization_problem.huber_scale = 1e2
POSE_GRAPH.optimize_every_n_nodes = 35

return options
```

head_2d.lua

This file defines the physical dimensions of our “robot”.

```
<robot name="head_2d">

  <material name="orange">
    <color rgba="1.0 0.5 0.2 1" />
  </material>
  <material name="gray">
    <color rgba="0.2 0.2 0.2 1" />
  </material>

  <link name="laser">
    <visual>
      <origin xyz="0 0 0" />
      <geometry>
        <cylinder length="0.03" radius="0.03" />
      </geometry>
      <material name="gray" />
    </visual>
  </link>

  <link name="base_link">
    <visual>
      <origin xyz="0.01 0 0.015" />
      <geometry>
        <box size="0.11 0.065 0.052" />
      </geometry>
      <material name="orange" />
    </visual>
  </link>

  <joint name="laser_joint" type="fixed">
    <parent link="base_link" />
    <child link="laser" />
    <origin rpy="0 0 3.1415926" xyz="0 0 0.05" />
  </joint>

</robot>
```

Demo.rviz

This file launches RVIZ, my graphical interface for Simultaneous Localization and Mapping (SLAM). The parameters in this file were gained from online tutorials and templates.

```
Panels:
  - Class: rviz/Displays
    Help Height: 78
    Name: Displays
    Property Tree Widget:
      Expanded:
        - /TF1/Frames1
```

```
      Splitter Ratio: 0.6006709933280945
    Tree Height: 719
  - Class: rviz/Selection
    Name: Selection
  - Class: rviz/Tool Properties
    Expanded:
      - /2D Pose Estimate1
      - /2D Nav Goal1
      - /Publish Point1
    Name: Tool Properties
    Splitter Ratio: 0.5886790156364441
  - Class: rviz/Views
    Expanded:
      - /Current View1
    Name: Views
    Splitter Ratio: 0.5
  - Class: rviz/Time
    Experimental: false
    Name: Time
    SyncMode: 0
    SyncSource: PointCloud2
Preferences:
  PromptSaveOnExit: true
Toolbars:
  toolButtonStyle: 2
Visualization Manager:
  Class: ""
  Displays:
    - Alpha: 0.5
      Cell Size: 1
      Class: rviz/Grid
      Color: 160; 160; 164
      Enabled: true
      Line Style:
        Line Width: 0.029999999329447746
        Value: Lines
      Name: Grid
      Normal Cell Count: 0
      Offset:
        X: 0
        Y: 0
        Z: 0
      Plane: XY
      Plane Cell Count: 100
      Reference Frame: <Fixed Frame>
      Value: true
    - Class: rviz/TF
      Enabled: true
      Frame Timeout: 15
      Frames:
        All Enabled: false
        base_link:
          Value: true
        laser:
          Value: false
        map:
          Value: true
        odom:
          Value: true
      Marker Alpha: 1
      Marker Scale: 1
      Name: TF
      Show Arrows: true
      Show Axes: true
      Show Names: true
      Tree:
```

```yaml
    map:
      odom:
        base_link:
          laser:
            {}
  Update Interval: 0
  Value: true
- Class: Submaps
  Enabled: true
  Fade-out distance: 1
  High Resolution: true
  Low Resolution: false
  Name: Submaps
  Queue Size: 10
  Submap query service: /submap_query
  Submaps:
    All: true
    All Submap Pose Markers: true
    Trajectory 0:
      0.180: true
      1.123: true
      2.33: true
      Submap Pose Markers: true
      Value: true
  Topic: /submap_list
  Tracking frame: base_link
  Unreliable: false
  Value: true
- Alpha: 1
  Autocompute Intensity Bounds: true
  Autocompute Value Bounds:
    Max Value: 10
    Min Value: -10
    Value: true
  Axis: Z
  Channel Name: intensity
  Class: rviz/PointCloud2
  Color: 0; 255; 0
  Color Transformer: FlatColor
  Decay Time: 0
  Enabled: true
  Invert Rainbow: false
  Max Color: 255; 255; 255
  Min Color: 0; 0; 0
  Name: PointCloud2
  Position Transformer: XYZ
  Queue Size: 10
  Selectable: true
  Size (Pixels): 3
  Size (m): 0.05000000074505806
  Style: Flat Squares
  Topic: /scan_matched_points2
  Unreliable: false
  Use Fixed Frame: true
  Use rainbow: true
  Value: true
- Class: rviz/MarkerArray
  Enabled: true
  Marker Topic: /trajectory_node_list
  Name: MarkerArray
  Namespaces:
    Trajectory 0: true
  Queue Size: 100
  Value: true
- Alpha: 0.699999988079071
  Class: rviz/Map
```

```
    Color Scheme: map
    Draw Behind: false
    Enabled: false
    Name: Map
    Topic: /map
    Unreliable: false
    Use Timestamp: false
    Value: false
  - Alpha: 1
    Class: rviz/RobotModel
    Collision Enabled: false
    Enabled: true
    Links:
      All Links Enabled: true
      Expand Joint Details: false
      Expand Link Details: false
      Expand Tree: false
      Link Tree Style: Links in Alphabetic Order
      base_link:
        Alpha: 1
        Show Axes: false
        Show Trail: false
        Value: true
      laser:
        Alpha: 1
        Show Axes: false
        Show Trail: false
        Value: true
    Name: RobotModel
    Robot Description: robot_description
    TF Prefix: ""
    Update Interval: 0
    Value: true
    Visual Enabled: true
Enabled: true
Global Options:
  Background Color: 100; 100; 100
  Default Light: true
  Fixed Frame: map
  Frame Rate: 30
Name: root
Tools:
  - Class: rviz/Interact
    Hide Inactive Objects: true
  - Class: rviz/MoveCamera
  - Class: rviz/Select
  - Class: rviz/FocusCamera
  - Class: rviz/Measure
  - Class: rviz/SetInitialPose
    Theta std deviation: 0.2617993950843811
    Topic: /initialpose
    X std deviation: 0.5
    Y std deviation: 0.5
  - Class: rviz/SetGoal
    Topic: /move_base_simple/goal
  - Class: rviz/PublishPoint
    Single click: true
    Topic: /clicked_point
Value: true
Views:
  Current:
    Angle: 0
    Class: rviz/TopDownOrtho
    Enable Stereo Rendering:
      Stereo Eye Separation: 0.05999999865889549
      Stereo Focal Distance: 1
```

```
        Swap Stereo Eyes: false
        Value: false
      Invert Z Axis: false
      Name: Current View
      Near Clip Distance: 0.009999999776482582
      Scale: 101.94859313964844
      Target Frame: <Fixed Frame>
      X: 0
      Y: 0
    Saved: ~
Window Geometry:
  Displays:
    collapsed: false
  Height: 1016
  Hide Left Dock: false
  Hide Right Dock: false
  QMainWindow State:
000000ff00000000fd0000000400000000000001560000035afc0200000008fb0000001200530065006c0065006300740069006
f006e00000001e10000009b0000005c00fffffffb0000001e0054006f006f006c002000500072006f00700065007200740069006 5
007302000001ed000001df00000185000000a3fb000000120056006900650077007300200054006f006f02000001df000002110
000018500000122fb000000200054006f006f006c002000500072006f0070006500720074006900650073003203000002880000
011d000002210000017afb000000100044006900730070006c006100790073010000003d0000035a000000c900fffffffb000000
2000730065006c0065006300740069006f006e00200062007500660066006500720200000138000000aa0000023a00000294fb
00000014005700690064006500530074006500720065006f02000000e6000000d2000003ee0000030bfb0000000c004b006900
6e0065006300740200000186000001060000030c00000261000000010000010f0000035afc0200000003fb0000001e0054006f0
06f006c002000500072006f00700065007200740069006500730100000041000000780000000000000000fb0000000a0056006
900650077007301000000 3d0000035a000000a400fffffffb0000001200530065006c0065006300740069006f006e010000025a0
00000b200000000000000000000000200000490000000a9fc0100000001fb0000000a00560069006500770073030000004e000
00080000002e10000019700000003000007800000003efc0100000002fb0000000800540069006d0065010000000000000007800
000026c00fffffffb0000000800540069006d006501000000000000045000000000000000000000051e0000035a0000000400000
004000000080000008fc0000000100000002000000010000000a0054006f006f006c00730100000000ffffffff0000000000000000
00
  Selection:
    collapsed: false
  Time:
    collapsed: false
  Tool Properties:
    collapsed: false
  Views:
    collapsed: false
  Width: 1920
  X: 0
  Y: 27
```