Gaussian process-assisted frontier exploration and indoor radio source localization for mobile robots

JAVIER RASINES SUÁREZ
Gaussian process-assisted frontier exploration and indoor radio source localization for mobile robots

JAVIER RASINES SUÁREZ
Abstract

Autonomous localization of a radio source is addressed, in the context of autonomous charging for drones in indoor environments. A radio beacon will be the only input used by the robot to navigate to an unknown charging station, at an unknown area. Previous proposed algorithms used frontier-based exploration and the measured RSS to compute the direction to the source. The use of Gaussian processes is studied to model the Radio Signal Strength (RSS) distribution and generate an estimation of the gradient. This gradient was also incorporated into a frontier exploration algorithm and was compared with the proposed algorithm. It was found that the usefulness of the Gaussian process model depended on the distribution of the RSS samples. If the robot had no prior samples of the RSS, then the gradient-assisted solution performed better. Instead, if the robot had some prior knowledge of the RSS distribution, then the Gaussian process model yields a better performance.
Sammanfattning

Contents

1 Introduction 1
   1.1 Problem Statement 3

2 Background 4
   2.1 RSS Model and Filtering 4
   2.2 Frontier Exploration 5
   2.3 Gaussian Process Models 5

3 Related Work 8

4 Methodology 10
   4.1 Initial Exploration 12
   4.2 Frontier Identification 12
   4.3 Gradient Estimation 12
      4.3.1 Gradient assisted exploration 12
      4.3.2 Gaussian-Process assisted exploration 13
   4.4 Frontier Ranking 16
   4.5 RSS Sampling 16
   4.6 Ending Condition 17

5 Experimental Setup 18
   5.1 Scenarios 18
   5.2 Algorithms and Metrics 19
   5.3 Parameter tuning 20

6 Results and Discussion 21
   6.1 Scenario 1 21
   6.2 Scenario 2 23
      6.2.1 Discussion on Scenario 2 24
   6.3 Scenario 3 25
6.3.1 Discussion on Scenario 3 . . . . . . . . . . . . . . . . . . . . 26

7 Conclusions and Future Work . . . . . . . . . . . . . . . . . . . . 28

A Sustainability, Ethics & Society . . . . . . . . . . . . . . . . . . . 34
Chapter 1

Introduction

During recent years, there is an increasing interest on the use of Unmanned Aerial Vehicles (UAV) for multiple applications [1]. These applications cover a wide range of fields, from military and defense, to recreational or photography. Recent improvements in electronics, materials and sensor technology, are helping to the development of better performing drones and their use for autonomous applications, in which human operators are not needed for the drone to complete a task.

Nowadays, one of the most limiting factors for autonomous applications of drones, is battery capacity. While microprocessor technology is improving at a doubling rate every 24 months (known as Moore’s law [2]), which increases the capacity of computations that a drone can execute, battery technology is developing in a more slow pace, by doubling every decade [3]. Therefore, battery life is the key to allow drones to perform longer autonomous tasks. This problem can be mitigated by the addition of extra battery modules, but this solution is limited, as there is an upper limit to which the weight of the extra battery increases the power consumption of the drones, more than the extra stored energy. Other solutions have been proposed to address this problem for example, by adding power consumption into the path planning [4], or integrating stations with wireless charging technology [5], so drones can go a station at a predefined location to charge their battery. Even though this solution is much slower than others involving human interaction, it gives drones an increase autonomy.

The use of radio signal strength (RSS) have been studied in the
field of robotics for multiple applications. The most common is the localization of a robot in an indoor environment \cite{6} \cite{7}, where the GPS solutions can not be used. This is known as simultaneous localization and mapping (SLAM) \cite{8}. Another range of applications involve the opposite problem, for the robot to find a radio source \cite{9}. To achieve this goal, a directional rotating antenna is the key component to estimate the direction of arrival of the signal. It is combined with the power received and the estimate of the source localization is computed \cite{10} \cite{11}. These types of solutions rely on specialized receptors with specific configurations, so their reliability is increased. A more generic and systematic approach is to integrate the RSS gradient as part of a frontier-based exploration \cite{12}, which also allows a robot to find a radio source, even if the environment is unknown a priori.

The radio signal propagation model tends to be very noisy, especially in an indoor environment. Gaussian processes (GP) have been successfully employed to map the RSS distribution in indoor \cite{13} and outdoor \cite{14} environments. The probabilistic nature of the Gaussian processes make them a powerful tool to integrate noisy measurements coming from the radio receiver.

Therefore, with the addition of a radio emitter to a charging station, a robot can use the received radio signal strength as a radio beacon. The idea is similar to the one used on avionics systems \cite{15}, to guide aircraft by using radio signals. By the combination of frontier-exploration and Gaussian processes, a robot will be able to discover and navigate to stations at unknown locations in unknown environments. The main limitation of the project is that, the radio receivers need to be present on consumer electronics commonly used in robotics. This means that antennas are restricted to be omnidirectional.
1.1 Problem Statement

The objective of this project is to implement an exploration algorithm that will navigate a robot to an unknown radio source, and to study how Gaussian process techniques can be used as a tool to improve existing algorithms based on the estimation of the RSS gradient.
Chapter 2

Background

In this section, the most important theoretical concepts applied in the methodology are presented to the reader.

2.1 RSS Model and Filtering

When a radio signal propagates from a source to a destination, its attenuation depends on environmental factors such as distance (path loss), objects in the environment (shadowing) and spatio-temporal dynamics (multipath fading) [16]. A frequently used model to represent the RSS is the log-distance path loss model [16]. The log-distance path loss model is a radio propagation model used in indoor environments or densely populated area, and it is used to predict the losses that a signal encounters:

\[
PL = PL_0 + 10 \gamma \log_{10}\left(\frac{d}{d_0}\right) + X_g
\]  

(2.1)

The model shows all the different effects that can happen to a radio signal when propagating in an indoor environment. \(PL_0\) represents the losses at a standard distance, usually 1 meter, within line of sight of the transmitter. The second term \(10 \gamma \log_{10}\left(\frac{d}{d_0}\right)\) reflects the losses due to the distance from the transmitter to the receiver. The term \(\gamma\) is called the path loss exponent. This is an empirical term used to adjust the attenuation to the frequency band of the radio signal. For a typical indoor scenario, within the IEEE 802.11 band, the typical value is 2.6. The last term \(X_g\) is a random variable that represents effects of con-
structive or destructive interferences of other signals. This is a spe-
cially important effect in the indoor scenarios considered in this project.

RSS measurements tend to be very noisy. To mitigate this effect
some filtering, an exponentially weighted moving average (EWMA)
filter \[11\] can be applied to the readings coming from the wireless re-
ceiver:

\[
S_t = \alpha Y_t + (1 - \alpha) S_{t-1}
\]  

(2.2)

This type of filter is an Infinite Impulse Response (IIR) filter. The
EWMA filter, applies a term \(\alpha\) (also known as discount, and its value
is between 0 and 1) to the new measurements. The higher the discount
value, the faster old data loses its importance on the output of the filter,
and more recent values get more importance.

2.2 Frontier Exploration

Frontier exploration is a technique first presented by \[17\] to allow robots
to gather information about unknown areas of a map. Frontier-based
exploration strategies usually operate on grid maps, where a frontiers
are defined as the cells between known and unknown areas. The idea
of frontier-based exploration strategies is to guide the robot to the cen-
troid of a frontier. The centroid chosen is based on an objective or
scoring function.

The main steps of a frontier exploration are:

1. Update the occupancy grid with the sensor information.
2. Each cell is classified as occupied, open or unknown.
3. All different frontiers are detected.
4. Next centroid is selected according to an objective function.

All these steps are repeated until the robot explores all the map.

2.3 Gaussian Process Models

Gaussian process methods allow to incorporate noisy measurements
in a probabilistic way, from an unknown latent model. They also al-
low to make predictions on unknown states of the model. Gaussian
Process Latent Variable Models (GPLVM) have been used in multiple robotic applications. Some of these applications include sensor-centric robot localisation [18], active exploration using Gaussian Process Implicit Surfaces [19] and for implicit surfaces for shape estimation and grasping [20]. The RSS distribution can be modeled as a function $f : \mathbb{R}^2 \to \mathbb{R}$ that maps values on the $xy$ plane to single RSS measurements. These type of functions can be modeled in an efficient way by using GPLVM, due to they place a multivariate Gaussian distribution over the function-space, in which the mean function $\mu(x)$ and covariance function $k(x, x')$ are:

$$\mu(x) = \mathbb{E}[f(x)] \tag{2.3}$$

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))] \tag{2.4}$$

Thus, the Gaussian process is

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{2.5}$$

A set of N measurements $D = \{x_i, y_i\}_{i=1}^N$ is a training set where $x_i \in X$ is a training point on the $xy$-plane. And $y_i$ is the RSS reading from the wireless receiver. The subscript $*$ is used to represent the point $x_*$ at where a prediction is made by sampling the Gaussian process. A Gaussian noise with zero mean and $\sigma^2_n$ variance is added as part of the measurement model, $f(x) + \epsilon$. Thus, the covariance between measurements $y$ at points $X$:

$$\text{cov}(y) = K(X, X) + \sigma^2_n I \tag{2.6}$$

The joint Gaussian distribution on a test set $X_*$, gets the form

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N} \left( 0, \begin{bmatrix} K(X, X) + \sigma^2_n I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right)$$

As proven in [13], the squared-exponential kernel is the one that better represents the variance in RSS.

$$k(x_i, x_j) = \sigma^2_e \exp \left( -\frac{(x_i - x_j)^T(x_i - x_j)}{\sigma^2_w} \right) \tag{2.7}$$
The log marginal likelihood of the model can be computed with respect to the training data as:

\[
\log p(y|X) = -\frac{1}{2}y^T (K(X, X) + \sigma_n^2 I)^{-1}y \\
- \frac{1}{2} \log|K(X, X) + \sigma_n^2 I| - \frac{n}{2} \log(2\pi) \tag{2.8}
\]

The last step to complete the RSS model as a Gaussian process, is to define a prior mean function to improve the accuracy of the prediction. One possibility is to use the log-distance path loss model on section 2.1:

\[
m(x) = RSS_0 - 10 \eta \log_{10}(|x - x^s|) \tag{2.9}
\]

where \(x^s\) is the source localization. As mentioned by [14] the position is unbounded on the unexplored map. In addition to sparse and noisy data of a practical robotic application, optimizing the hyperparameters to get an accurate position will result on expensive computations and low accuracy on the results.

More complex priors can be used to introduce propagation effects, multi-path, fading, etc. But as can be seen on [13], a constant prior mean function provides a good accuracy while keeping the computational complexity relatively low, in comparison with other approaches.

\[
m(x) = C \tag{2.10}
\]

For running the Gaussian process online, the model is trained after each measurement and the hyperparameters are optimized using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.
Chapter 3

Related Work

The topic of radio signal strength and radio source tracking, has been addressed by several mobile robot researchers. One important topic in this field is to make robots aware of the connectivity, so data or a human operator can be in contact with the robot at all times. This problem was researched in [14]. In this paper, the robot was aware of the RSS distribution, and it was an input to the path planning algorithm, that choose paths which maximize the received power signal.

One important aspect is to estimate the RSS gradient and the radio source localization. The first techniques used, heavily depend on the local RSS gradient and the estimation of the radio source localization by using a propagation model. This technique showed a successful result to navigate a UAV to a radio source [21] but it was limited to an outdoor scenario with line of sight, from the transmitter to the receiver. Similar attempts in indoor environments were not that successful. Several other techniques were tried to estimate the source localization from RSS measurements, in an indoor scenario. As shown in [22] these other techniques are centroids, gradient and weighted centroids. In the centroids approach, the distance to the source is estimated from several measurements. Then, a circle with that radius is drawn for each measurement and in the intersection point is the position of the source. In the weighted version of the algorithm, every circle is weighted by the signal to noise ratio (SNR) of its measurement. In the gradient, for every measurement, instead of a circle, the local gradient is computed. In the following step, all gradients are weighted by the SNR. From the three solutions presented, the gradient one yield the best results. If the scenario was not too complex, the weighted centroids can...
obtain accurate enough results.

The local gradient technique was used in [23], where a mobile robot was able to navigate to a particular location by using the RSS received from the Wireless Sensor Network (WSN). The WSN was previously deployed to identify a threat and then to guide the robot from one sensor to the next one, by the radio signal emitted from the network. In this approach, the robot already knows how the environment looks like, thanks to the data provided from the WSN.

In the paper [12], a robot is able to navigate to a radio source signal, in a previously unknown environment. The algorithm proposed in this paper merges the local gradient of the RSS with a frontier-based exploration. Every frontier is ranked not only by its distance to the robot, but also taking into account if the frontier is aligned with the RSS gradient.

One main problem with all these methods, is that they do not take into account the probabilistic nature of the radio signal propagation. The state of the art to model RSS distributions are GPLVMs [24]. They have been successfully used in robotics applications to model complex signals and latent variables in statistical models. They were first introduced as a radio mapping tool by Jonathan Fink in the paper [13]. They were also used in [14] with a successful prediction of the RSS, and once the robot had visited multiple locations on the map, the distribution was accurately approximated. Also, in the later paper, the Gaussian process was used online. This helped the robot, not only to create an accurate signal distribution, but also to adapt to changes on the environment. This was achieved by retraining the Gaussian process after each of the measurements.

This degree project combines the two ideas of radio frontier-based exploration and online GPLVM.
Chapter 4

Methodology

In this section, the exploration algorithm is presented, alongside the two variations of it: Gradient assisted and Gaussian-process assisted. The gradient assisted is based on the current techniques for exploration and radio source localization. The Gaussian-process algorithm includes the improvements to be analyzed in this project.

Both algorithms follow a similar high-level logic:

1. An initial exploration around the robot starting position. This step helps to get a better first estimate of the gradient.

2. Frontiers are identified in the exploration map.

3. A local RSS gradient is estimated.

4. Frontiers are ranked according to the distance to the robot and if they are on the way of the gradient.

5. The robot moves to the best frontier, while additional samples of the RSS are taken.

6. If the RSS measured is above a pre-establish threshold the exploration is finished. If not, the robot goes back to step 2.

These steps are detailed in the Figure 4.1.
Figure 4.1: Flowchart of the implemented algorithms
4.1 Initial Exploration

This is the first step of the algorithm. The goal here is to gather a small dataset of RSS measurements, so the first estimate of the RSS gradient is improved. To achieve it, the robot takes several samples at random places close to its original position.

4.2 Frontier Identification

As presented in section 2.2, a frontier is a set of cells on an occupancy grid, that are between known and unknown regions. To identify them several steps take place:

- All frontier cells are identified.
- A k-means algorithm is used to cluster the frontier cells.
- The centroid of each frontier is calculated. If two centroids are too close to each other, they are merged together.

![Figure 4.2: Example of frontier identification from the experiments](image)

4.3 Gradient Estimation

This step is the one that makes the difference between both algorithms.

4.3.1 Gradient assisted exploration

As shown in section 2.1, the RSS gradient is computed using RSS samples. Only samples close to the robot are used. The reason is that in
an indoor environment, the characteristic of a radio signal can change drastically with distance. The \(i\)-th RSS measurement has the following form \((X_{i,1}, X_{i,2}, RSS_i)\). The steps to estimate the RSS gradient are the following:

1. The location of the measurements is centered:
\[
\sum_i^n X_{i,1} = \sum_i^n X_{i,2} = 0 \quad (4.1)
\]

2. A two dimensional least-squares estimation is performed, in which a plane is fit to the RSS measurements:
\[
X \beta = Z \quad (4.2)
\]
\[
\beta = (X^T X)^{-1} X^T Z \quad (4.3)
\]

3. The estimated direction and magnitude of the gradient are:
\[
\Theta = \tan^{-1}\left(\frac{\beta_2}{\beta_1}\right) \quad (4.4)
\]
\[
|\beta| = \sqrt{\beta_1^2 + \beta_2^2} \quad (4.5)
\]

**4.3.2 Gaussian-Process assisted exploration**

In this algorithm, the data used to compute the RSS gradient does not come from the sampled measurements. Instead, the samples are used to train a GP with a RBF kernel. The GP is used to generate a model of the RSS distribution. The mathematical formulation is the same as presented in Section 2.3, with a constant function as a prior of the mean (this constant function is an hyperparameter for the optimizer).

Because this algorithm uses a model, it involves the extra difficulty of ensuring that the model will adjust properly to the real world phenomena, and how the gradient is computed from the predicted model. These two challenges are discussed in the following sub-sections.
Convergence of the model

When using GPLVM to estimate the RSS distribution, it is important to know how many measurements are needed, for the model to converge to the real distribution. In previous works [13] [14], it is shown how a Gaussian process can estimate the RSS distribution after most of the map has been sampled. In this project, a GPLVM is used as part of an exploration algorithm, so the Gaussian process will start without any samples, and it will keep converging during the execution of the algorithm.

In Figure 4.3 can be seen how the predicted RSS model evolves depending on the number of samples. It can be seen that not only the

![10 samples](image1)

![20 samples](image2)

![30 samples](image3)

![40 samples](image4)

Figure 4.3: RSS model (red) for different number of samples (green). On blue is the real radio source signal value.
number of measurements is important, but also their spatial distribution. From the case of 10 to 20 samples, even though the number of samples is doubled, the estimation is similar. The reason behind is that the new samples are in the same line as in the previous ones. But from 20 to 30 samples, the relative increment of sample is less (50%) but the model converges more to the real distribution. This effect is cause by the spatial diversity, because the new points are no longer in the same line as the previous ones. To mitigate this effect when executing the exploration algorithm, a small deviation from the path is included when the robot is going to move. The deviation follows a uniform distribution between -0.2 to 0.2 m, which will help to add some spatial diversity to the measurements, as the will not follow a perfect straight line and they will be more distributed.

A new sample is added every time the robot moves. This means that the estimation of the RSS can change drastically. This is specially true in the beginning of the algorithm, due to the low number of samples, a new addition can affect significantly. For this reason, it was decided that after a new sample was taken, the frontiers will be identified and ranked and the RSS model computed again. This will make the algorithm computationally more expensive, but the gradient estimation (described in the next subsection) will be better, as it is generated with the latest information.

**Gradient calculation**

As seen in the previous section, the Gaussian process model yields an accurate prediction of an area, if it has samples spatially well distributed. For this reason, the model will be only use to estimate the RSS distribution on a small area around the robot. As long as the area is not too large, the prediction converges to a good estimation of the real distribution, even if on regions far away is too different from the real one.

After the estimation of the RSS is completed, the gradient is calculated in the way as in [4.3.1](#) by solving a two dimensional least-squares estimation.
4.4 Frontier Ranking

Once the frontiers have been identified, and the gradient computed, the next step is to select to which frontier to move. This is done with the following scoring function:

$$\text{Frontier Score} = \alpha \cdot \frac{1}{\text{angle}} + \beta \cdot \frac{1}{\text{distance}}$$ (4.6)

The score of a frontier is equal to the inverse of its distance with the robot and their alignment with the RSS gradient. The alignment is computed as the inverse of the angle between the frontier to the robot and the gradient. The former term, makes frontiers that are on the way of the RSS gradient to have a higher score, because they have a higher chance to be on the path to the radio source. The latter term, is used to prioritize closer frontiers, so the movements of the robot are more efficient, and more unknown cells are explored with the distance traveled.

The parameters $\alpha$ and $\beta$ are used to tune the performance of the algorithm.

4.5 RSS Sampling

After every move taken by the robot, the RSS is sampled. To remove the noise, several samples are taken and then passed through an exponentially weighted moving average filter (as seen in Section 2.1). This
step can increase the overall computational time of the algorithm, because of the number and delay between measurements. This can be relevant if the scenario has a high level of noise.

### 4.6 Ending Condition

At the end of every iteration, the maximum value of RSS measured is compared with a threshold and if it is greater, then the exploration is finished. The value of the threshold depends on the radio source characteristics. Its value should be such, that when it is reached, the robot is close to the source (around 1 meter).
Chapter 5

Experimental Setup

The experiments were run on a laptop with an Intel Core i7 8th gen and 16 GB of memory. As a software framework, ROS (Robot Operating System) was used and Gazebo was the simulation environment. It is relevant to mention, that the wireless transmitters and receivers were implemented using the sensor plugin from Gazebo’s library. The version of ROS was kinetic and version 7 for Gazebo.

A ROS package with the implementation of the exploration was developed using Python 2.7. For the gaussian process regressor, the library GPy [25] from the machine learning group from the university of Sheffield was used. Hector Slam [26] was used to provide a reliable simultaneous localization and mapping of the environment.

5.1 Scenarios

For evaluating both algorithms, three scenarios were chosen. These scenarios can be seen in the figure below.
These scenarios have been chosen to reflect a different set of challenges that the algorithms can face. In the first one, the challenges are that the robot has to move on opposite direction to the gradient to overcome the obstacle and then continue to the radio source. In the second scenario, the challenge is that the robot should avoid going into the two rooms on the side of the corridor. The second reason for choosing this scenario is because it is similar to the one presented on [12], so the comparison with previous work can be done more easily. On the final scenario, the robot starts between two corridors. In this setup, some frontiers will be detected on the wrong corridor, and the robot will have to choose the correct one with few RSS samples.

5.2 Algorithms and Metrics

For every scenario, the two algorithms presented in Chapter 4 are simulated. As it was discussed on Section 4.3.2 the convergence of the Gaussian process to the real distribution is an important factor. The two metrics that were recorded during the simulations were the distance travel by the robot, and the time that it took to the robot to complete that distance. These two metrics were chosen to be the more relevant to evaluate for the goal of this project, when the robot is low in battery and it has to reach a charging station. To evaluate how the amount of samples used in the Gaussian process affects the final result, a third case of a Gaussian-process assisted algorithm with an initial dataset is studied. This initial dataset includes samples around the starting position of the robot.
For every case, the total distance and time were average over 5 simulations.

5.3 Parameter tuning

In section 4 the parameters of the algorithms were presented. Before starting the experiments, these parameters were tuned to ensure the algorithms had a correct behavior. The values of the tuning process are presented on table 5.1. After the tuning process was completed, these parameters were kept constant during all the experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster k-means</td>
<td>12</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\beta$</td>
<td>2.0</td>
</tr>
<tr>
<td>Samples initial dataset</td>
<td>40</td>
</tr>
<tr>
<td>RSS stopping threshold [dbm]</td>
<td>0</td>
</tr>
</tbody>
</table>
Chapter 6

Results and Discussion

In this section, the results of the experiments are presented. The results of each scenario are discussed in a subsection after the results.

6.1 Scenario 1

In table 6.1 the averaged results of the simulations of scenario 1 are presented. GP-assisted algorithm with initial dataset shows the lowest mean distance, while Gradient-assisted algorithm shows the lowest average time. In figure 6.1 paths for each algorithm in scenario 1 are shown. Gradient-assisted followed the straightest path, while both GP-assisted paths had some drift to the top of the map. Figure 6.2 is presented to show the final RSS distribution that the GP had modeled, after the radio source has been reached. While the maximum does not coincide with the radio source, the gradient during the path points towards the source.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distance [m]</th>
<th>Time[s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>16.88 ± 7.18</td>
<td>403.44 ± 82.02</td>
</tr>
<tr>
<td>GP</td>
<td>23.43 ± 2.41</td>
<td>616.05 ± 123.9</td>
</tr>
<tr>
<td>GP-dataset</td>
<td>14.22 ± 0.21</td>
<td>588.32 ± 10.73</td>
</tr>
</tbody>
</table>

Table 6.1: Results scenario 1
Discussion on Scenario 1

First, Gradient and GP based algorithms are going to be discussed, as GP-assisted is an improvement over the later. In this scenario, the gradient assisted algorithm has a better performance than the GP version in terms of distance and time. The main reason for this result is the initial bad estimation made by the GP due to a low number of samples. This can be seen in figure 6.1 when the GP-assisted algorithm tried to go around the corner. It can be seen also at the end of the path, in which the GP algorithm drifted to the top of the map. In both cases, the Gradient-assisted algorithm followed a shorter path.
This problem is fixed when a dataset is included to the Gaussian process. Not only the paths tend to be slightly shorter, but most important is paths became more consistent. This can be seen as how the variance is reduced to $\pm 0.21$ [m], caused by a more accurate model of the RSS distribution. This allowed the robot to make better estimations and therefore, to follow straighter paths. Another effect is that the path did not present oscillations, as can be seen in the other two algorithms. In both cases of the GP algorithm, the final RSS model presented in figure 6.2 shows the maximum of the distribution to be in the top right corner. This corresponds to the lack of measurements in that area, and the tendency of all measurements to increase in that direction. Nevertheless, the accuracy of this model is sufficient so the robot can successfully navigate to the radio source.

The time on the three cases is similar. Between gradient-assisted and GP-assisted with initial dataset, there is a trade-off between distance and time. The gradient case is lighter to compute but generates less optimal paths, when in the GP with dataset the computation time is longer due to the higher amount of data points but the path is shorter.

### 6.2 Scenario 2

In table 6.2 the averaged results of the simulations of scenario 2 are presented. GP-assisted algorithm with initial dataset shows the lowest mean distance and the lowest average time. In figure 6.3, paths for each algorithm in scenario 2 are shown. The paths of GP and Gradient-assisted algorithms show noisy spikes, which are due to the complexity of the scenario and the wrong directions taken by the algorithms because of it. Figure 6.4 is presented to show the final RSS distribution that the GP had modeled, after the radio source has been reached. While the maximum does not coincide with the radio source, the gradient during the path points towards the source.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distance [m]</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>14.10 ± 3.27</td>
<td>581.55 ± 096.1</td>
</tr>
<tr>
<td>GP</td>
<td>16.07 ± 6.48</td>
<td>634.42 ± 326.7</td>
</tr>
<tr>
<td>GP-dataset</td>
<td>07.36 ± 1.15</td>
<td>385.99 ± 08.37</td>
</tr>
</tbody>
</table>
Figure 6.3: Results on scenario 2

6.2.1 Discussion on Scenario 2

In this scenario, the relative performance between the algorithms show the same evidences as in the first scenario. The gradient based algorithm performed better than the GP without initial dataset. This can be seen in figure 6.3 as how the GP-assisted path (blue line) has some bias towards the bottom of the scenario, before correcting its trajectory at the entrance of the third bottom room.

The performance of the GP with initial dataset is significantly better than the other two. It has less variance, as in the previous scenario, and the total distance is half of the other two cases. An important
reason for this improvement of performance, is due to the maximum of the model is close to the real position of the radio source. The radio source is located at \( x = 10, \ y = 5 \) and the maximum of the model is at \( x = 12, \ y = 4 \).

The increase in performance of the case of GP with initial dataset also translates to the total time. As seen in the previous scenario, the extra computation of the Gaussian process can cancel the gain obtained from a shorter path. But in this case, the path obtained is efficient enough to compensate the additional computation and to obtain a smaller final time.

In comparison with the scenario presented on [12], the limitations of the simulator used in this project did not allow to see the expected oscillations between the top and bottom middle rooms.

### 6.3 Scenario 3

In table 6.3, the averaged results of the simulations of scenario 3 are presented. GP-assisted with initial dataset shows the lowest distance and time to complete the task. In figure 6.5, paths for each algorithm in scenario 3 are shown. Figure 6.6 is presented to show the final RSS distribution that the GP had modeled, after the radio source has been reached.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Distance [m]</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>14.58 ± 5.04</td>
<td>602.25 ± 28.69</td>
</tr>
<tr>
<td>GP</td>
<td>15.58 ± 3.87</td>
<td>640.17 ± 203.3</td>
</tr>
<tr>
<td>GP-dataset</td>
<td>11.18 ± 1.80</td>
<td>568.56 ± 24.97</td>
</tr>
</tbody>
</table>

Table 6.3: Results scenario 3
6.3.1 Discussion on Scenario 3

In this scenario, all algorithms chose the correct corridor. This shows that the frontier-based exploration with the RSS gradient is robust enough to correctly solve scenarios like this one.

The results of Table 6.3 follow the same lines of the previous discussions. The best performance is given by the Gaussian process with initial dataset.

An important remark of the results of this scenario is the final RSS model. As in the scenario 2, the maximum of the distribution is close to the real position of the radio source. But on the top left there is a
local maximum, corresponding to the unexplored corridor. In this type of cases, the term depending on the distance of the scoring function, on equation 4.6, had an important role. In the beginning the robot followed the higher gradient, but later on the path, the distance term helped the robot to go to closer frontiers and prevented to go back to the unexplored corridor. In the case the robot would have gone into the wrong corridor, the score corresponding to the RSS would become large enough to overcome the score of the close frontiers.
Chapter 7

Conclusions and Future Work

After the implementations and experiments have been completed, there are several conclusions that can be made. The first and most important is the convergence of the Gaussian process model. As shown in Chapter 6, the GP-assisted algorithm obtained a worse performance than the gradient-assisted, due to the lack of samples, that make the RSS model to be inaccurate. But when an initial dataset was provided to the GP-assisted algorithm, the performance is similar to the gradient-assisted. In complex scenarios, like scenario 2, the performance with the dataset is significantly better than the gradient-assisted.

The research question asked whether the use of a Gaussian process will improve the gradient-assisted algorithm. The answer to this question can be divided in two, depending of the goal of the robot. If the goal of the robot only includes discovering and navigating to radio sources, in a completely unknown scenario, then the use of the gradient-assisted algorithm will obtain better results. In the case the primary goal of the robot is not just localize radio sources, but to perform another task, this will provide an initial dataset with which the Gaussian-process assisted algorithm will yield better performance, once we want to localize the radio source.

This second context is closer to the scenario considered in this degree project, where autonomous drones will be performing tasks and they will just navigate to the radio source only when they need to charge their battery.

In the future lines of this project, there is the study of the algorithms on a real scenario. This will allow a more in depth analysis of the propagation effects that are not modeled in the simulator, and how
these affect both algorithms and their performance.

In this project only the case of one drone has been studied. I believe it will be very interesting to study how the algorithm can be improved by sharing the RSS measurements between drones in a peer-to-peer (P2P) fashion. This presents the challenge of data fusion from all different drones data, but once is solved, the problem with spatial distribution of the samples can be efficiently mitigated.

Other machine learning techniques, such as neural networks, will be interesting to study, instead of using GPLVM. It could be specially interesting if the neural network will not only take as an input the RSS measurements, but also the obstacles on the occupancy grid.
Bibliography


Appendix A

Sustainability, Ethics & Society

This degree project does not present any direct ethical issues, neither in the methodology nor the results presented. But on a broader view, the field of robotics presents deep and complicate ethical issues. With the recent progress in automation, the number of jobs that are going to evolve, or disappear, in the near future is increasing.

Since the beginning of the human race, people have to perform multiple tasks in order to ensure their survival. With the rise of written communication and more complex societies, specialized jobs started to appear and the quality of life increased significantly. Since the industrial revolution, the trend has been to automate and remove the human factor in order to obtain better and cheaper products and services.

But with the advent of automated factories, self-driven cars and package-deliver drones a future with less humans required to work seems more probable than ever before. This will lead humanity to a debate that not only involves jobs, but also how societies are organized. Without going deeper into the topic, it is important to remark that if this transition is done poorly, it can cause a big human problem. Proposals and debates are needed to reach a consensus for solutions to solve this problem and build a sustainable future in which robots and humans coexist.

One well-known proposed solution is to introduce a *universal basic income*. It consist on governments providing a small amount of money to their citizens, that combined with the low price of products thanks to a high amount of automation, it will help to make products accessible to everyone. This idea has a lot of people on favor and against it, because it really changes how our society have been working since the
beginning of time.

In this context, even if this degree project does not directly affects any current jobs, it lays down another stone on the path to this fully automated future. The reader is encourage to read and form an opinion on this topic, as it will be gaining relevance in the years to come.