ON-PLATE AUTONOMOMOUS EXPLORATION FOR AN INSPECTION ROBOT USING ULTRASONIC GUIDED WAVES

A Thesis
Presented to
The Academic Faculty

By

Ayoub Ridani

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
G. W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

May 2021

ON-PLATE AUTONOMOMOUS EXPLORATION FOR AN INSPECTION ROBOT USING ULTRASONIC GUIDED WAVES

Approved by:

Dr. Nico Declercq, Co-advisor
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Cedric Pradalier, Co-advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Ye Zhao
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: March 4, 2021

To my parents and my sisters.

Thank you for your endless love, sacrifices, prayers, support and advices.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisors, Dr. Cedric Pradalier and Dr. Nico Declercq, for giving me the opportunity to work on such an interesting project and to sharpen my skills on robotics and acoustics. I would like to thank them for constant support and guidance. I would also like to thank Dr. Ye Zhao for serving on my thesis committee.

Special thanks to my colleague Othmane Ouabi, who significantly helped me in the completion of my thesis; his assistance and contributions were essential to my success.

Finally, I would like to express my gratitude to my parents. Their love is always my motivation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Autonomous Robotic Exploration is a major research issue in robotics incorporating the aspect of how to make decisions for the next actions to maximize information gain and minimize costs. In this work, we elaborate an active-sensing strategy based on frontier-based exploration to enable the autonomous reconstruction of the geometry of a metal surface by a mobile robot relying on ultrasonic echoes. Such a strategy can be beneficial to the development of a fully autonomous robotic agent for the inspection of large metal structures such as storage tanks and ship hulls. Our exploration strategy relies on the occupancy grid generated by detecting the first echo of the signal referring to the closest edge to the sensor, and it employs a utility function that we define to balance travel cost and information gain using the plate's geometry estimation. Next, the sensor is directed to the next best location. In simulation, the method developed is evaluated and compared with multiple algorithms, essentially closest and random frontier point selection. Finally, an experiment using a mobile robot equipped with co-localized emitter/receiver pair of transducers is used to validate the viability of the proposed approach.

CHAPTER 1
INTRODUCTION

In many different fields of industry, monitoring the health of of structures is a major stake (if not critical). Ultrasound-based inspection methods have been identified as a promising solution for the long range inspection of materials. Structures under consideration are usually made of metal (*e.g.* pipes, rails...) or composite materials (*e.g.* plane outer shapes...).

On the one hand, most recent ultrasound-based inspection methods are designed to be used on static networks of sensors that are permanently attached to the structure, with applications in structural health monitoring (SHM). Such methods, however, can only be used to monitor a very small area. Ultrasound-based robotic inspection, on the other hand, has yet to be established, despite the fact that it could benefit from agent mobility in order to precisely assess the integrity of large structures such as ship hulls or storage tanks using acoustic tomography.

While precise localization of a mobile agent on a plate-based metal structure has been identified as a key requirement for long-range robotic inspection [1], UGWs have proven to be useful for both plate geometry reconstruction and on-plate robot localization, providing an innovative solution to the SLAM problem. Nonetheless, this solution was evaluated without the use of a real robotic platform and with pre-defined robot paths.

From a robotic perspective, plate geometry reconstruction and defect detection can be considered as a mapping problem, for which autonomous solutions are desirable [1]. Autonomous robotic exploration is a major research issue in robotics that involves the aspect of how to make decisions for the next actions in order to maximize information gain while minimizing costs. Frontier-based approaches are widely used to map indoor environments and yield common solutions to the exploration problem [2]. In our application, we extended such strategies as to determine appropriate robot trajectories with the goal of achieving fast and accurate reconstruction of the plate geometry, while also ensuring that the sensor stays on the plate during the reconstruction and does not cross any boundary.

In this work, we consider a mobile unit equipped with a pair of co-localized piezoelectric transducers for ultrasonic waves emission and reception. Contrasting with standard sensing technology (range-finder laser, ultra-wide band beacons, sonar...), the exploitation of guided waves measurements in a pulse-echo setup is more challenging due to their dispersive nature which causes wave-packet deformation when the propagation distance is large, and second, due to the overlapping of the multiple reflections of the incident wave on the plate boundaries which results in diffuse mixture data.

1.1 Contributions

In this work, we aim at pursuing the work which was initiated in [3] by presenting a novel approach for exploring and estimating the geometry of rectangular metallic structures using Lamb waves.

In summary the contributions of this thesis are the following:

1. A demonstration of the utility of using the occupancy grid map in combination with the geometry estimator to map a ”safe space” that ensures the sensor remains on the plate during robot motion preventing the plate’s geometry estimation from being lost.

2. Defining a utility function for the frontier exploration approach that balances information gain and travel cost.

3. A demonstration of the method’s validity on simulation and real-world scenarios using a mobile robot on a rectangular metallic plate.

1.2 Thesis overview

The remainder of the thesis is outlined as follows. In the second chapter, relevant studies on ultrasonic guided waves and frontier exploration are reviewed. In chapter 3, the proposed method for mapping and frontier exploration using ultrasonic guided waves is presented, evaluated on simulation and real-world experimentation with a mobile robot. Finally the method's results and limitations are discussed. In chapter 4, a mapping method that enables the correction aspect of the occupancy grid is briefly presented. Frontier exploration is tested under this mapping modality. Simulation and experiment results with a mobile robot are presented and discussed. The findings are summarized in Chapter 5 and future works are discussed.

CHAPTER 2

RELATED WORK

2.1 Guided Waves

On metal plates, Ultrasonic Guided Waves (UGWs) are generated by applying piezoelectric transducers in contact with the surface. These waves propagate radially around the emitter through the plate material, and potentially over large distances. When encountering the plate edges, these waves are reflected perpendicularly, and a receiver can sense the reflections.

Guided waves are successfully used by SHM systems for defect detection on structures such as pipes or rails [4, 5], where the sensors positions and the structure geometry are known accurately. Other works [6, 7] investigate methods to infer a metal structure geometry and structural features such as stiffeners, by relying on guided waves as well. Still, static networks of sensors are used in a pitch-catch setup.

In the robotics literature, the problem of room shape reconstruction from acoustic echo is studied [8, 9]. However, they rely on sound waves propagating in the air without dispersion. Furthermore, the integration of UGWs on a mobile unit for shape reconstruction is not thoroughly studied in the literature.

With the goal of demonstrating the applicability of UGWs for a robotic application, previous works [3, 10] establish that these waves can yield sufficient capabilities for simultaneous localization and mapping with a centimetric precision on rectangular metal plates. However, only pre-defined robot paths (based on the prior knowledge of the plate geometry) are taken into account. Furthermore, they emphasize that the robot trajectory in itself can have a significant impact on both the precision and time of the reconstruction process, with a greater impact for larger surfaces. As a result, exploration strategies for autonomous plate geometry reconstruction are required.

2.2 Frontier Exploration

Robotics exploration, which uses mobile robots to map unknown environments, has been studied for years. The main question in exploration is: based on what you already know about the world, where should you move to gain as much new information as possible? Among the various proposed methods, the frontier-based exploration is one of the classical approaches [11, 12]. The main idea is to move to the frontier which represents the boundary between unexplored space and known space in order to gain more information about the robot's environment. The mapped territory expands as the robot moves closer to frontiers, pushing back the boundary between the known and unknown. As a result, the robot's knowledge of the world increases. The key to effective exploration is the selection of target frontier points. In the original method [12], the closest frontier is selected as the next target. In most cases, exploration strategies select the next best frontier by evaluating candidate locations according to different criteria resulting in various extensions of the basic frontier based exploration strategy [2]. In [13] a utility function is defined balancing the travel cost with the information gain (in terms of the maximum unexplored area that could be viewed from it) of frontier cells. In [14], a more principled approach to aggregate criteria, based on multi-objective optimization, is proposed. Currently, there are two methods for calculating information gain: one uses direct measurements of undetected space size in the visible region of the target frontier point [14], and the other uses the information entropy method [15]. In both cases we use the current map to provide a more accurate estimate for the information gain. With the assumption that the robot has some nominal sensor range, the number of ”unknown” cells within that radius of the frontier point is counted. In our specific problem, given that lamb waves propagate over long distances in metallic plates and the maximum range exceeds the standard dimension of these plates, using the sensor range in this case is inappropriate.

On the other hand, [16] recently introduced Lamb wave-based frontier exploration strategy (LFE), demonstrating the potential contribution of Lamb wave-based sensing to the field of mobile robot exploration. The work considers a pair of transducers in a pitch-catch configuration on the surface of an isotropic metallic plate structure. The length of the path from the actuator to the sensor via the nearest edge point is calculated using time-of-flight (TOF) methods. The actuator and the sensor are the foci of the explored space, which is an ellipse. In other words, there are no reflection sources inside the ellipse, such as edges or defects. This sensing modality serves then as the foundation for mapping and frontier exploration. First, the obstacle cells are defined as frontier points that are believed to exist on or near the true plate's edges. Then, the minimal bounding box that adequately considers these estimated obstacle cells serves as an estimate of unknown environment. The greedy strategy is defined then as the subsequent transducer movements maximizing information gain(maximum covered area) based on the estimation of the environment. The method's major drawbacks, as mentioned by the authors, include the inability to map the entirety of a sharp corner of a structure. The transducers are limited in their placement to the edges to avoid the risk of falling off the structure; thus, there is always an edge closer to the transducers than the corner point. The implication is that mapping a plate structure with sharp corners using this lamb wave-based sensing modality cannot be guaranteed even under ideal conditions. Besides, a gridlock situations occur also when the bounding box used to estimate the environment is inaccurate, resulting in inaccurate estimations, and it is impossible to distinguish between complete mapping and a gridlock without knowing the upper limit of obtainable coverage. Transducers are manually positioned in the desired position, moving them instantaneously between two locations. However, in a real-world scenario, a mobile robot moves continuously to the desired position, gathering more information about its surroundings, and the optimality of the chosen goal doesn't necessarily persist.

[17] discusses the drawback of continuing to move towards the intermediate target with-

out considering the continuously updated frontier information. In order to reduce redundant exploratory motion, a repetitive rechecking approach is proposed in which the designated intermediate target is checked for being a valid frontier cell.

CHAPTER 3

FRONTIER EXPLORATION USING ULTRASONIC GUIDED WAVES

3.1 Method

We are considering a mobile robot equipped with a co-located transmitter/receiver pair of transducers and moving on a metal surface. The emitter sends a pulse $s(t)$ to excite guided waves in the plate material at each scanning phase, and the receiver collects the acoustic response $z(t)$ containing the ultrasonic echoes.

Frontier-based exploration is taken as the basic framework and we seek to find the optimum frontier points to take autonomous action to reconstruct the geometry of the plate as quickly as possible, and without getting too close to its boundaries.

Plate estimation and occupancy grid update are kept running in the background, which continuously integrate robot pose and acoustic measurements. At each exploration step, the following steps are followed:

- Frontier points are extracted from the occupancy map generated
- A utility function is used to evaluate the potential destinations
- The candidate pose with the highest utility is selected as the next goal
- Navigate to the target position

The map is continuously updated as the robot moves toward the goal. If the goal is reached, no longer valid(discussed in more detail in section 3.1.6) or the local exploration step is attained, the exploration process is ran again. The exploration process is considered complete where there are no more valid frontier points. A graphical overview of the proposed approach is presented in Fig. 3.1.

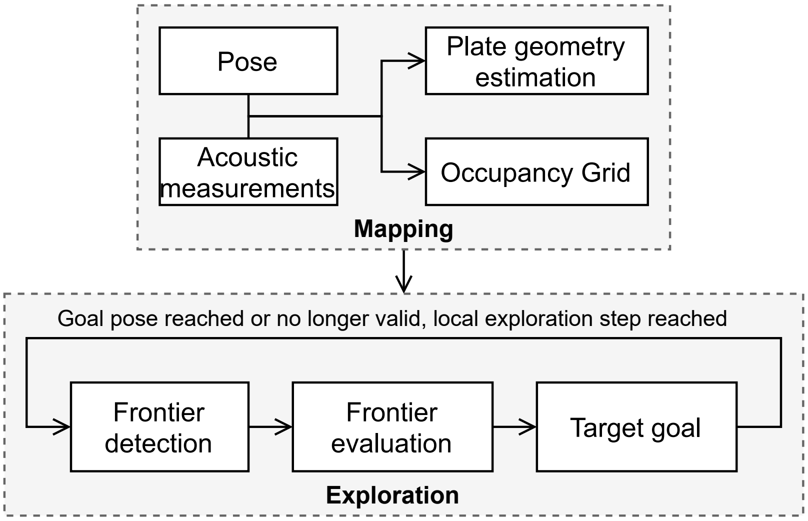


Figure 3.1: Block diagram of the proposed method

3.1.1 Plate geometry estimation

A model is adopted in which the plate is represented by a set of straight lines, each of which represents one of the plate's edges estimates. A line l can be represented in a unique way by a distance parameter r_l and an angular parameter θ_l with respect to a fixed coordinate system relative to the plate. Figure 3.2 illustrates this model.

As in [3], echo detection and plate geometry reconstruction are based on a propagation model which is used to construct $\hat{s}(r,t)$, the expected signal that would be received if the incident wave is reflected at a distance r to the transducers. Given measurement $z_i(t)$, the correlation signal:

$$z_i'(r) = \frac{\langle z_i(t), \hat{z}(r,t)\rangle}{\sqrt{\langle z_i(t), z_i(t)\rangle\langle \hat{z}(r,t), \hat{z}(r,t)\rangle}} \tag{3.1}$$

is computed and its envelop (that we will call $z_i(r)$ for simplicity, and which shall not be mistaken with the temporal signal $z_i(t)$):

$$z_i(r) = |z_i'(r) + j\mathcal{H}(z_i')(r)| \tag{3.2}$$

is retrieved where $\mathcal{H}$ stands for the Hilbert transform operator. The most likely first-order

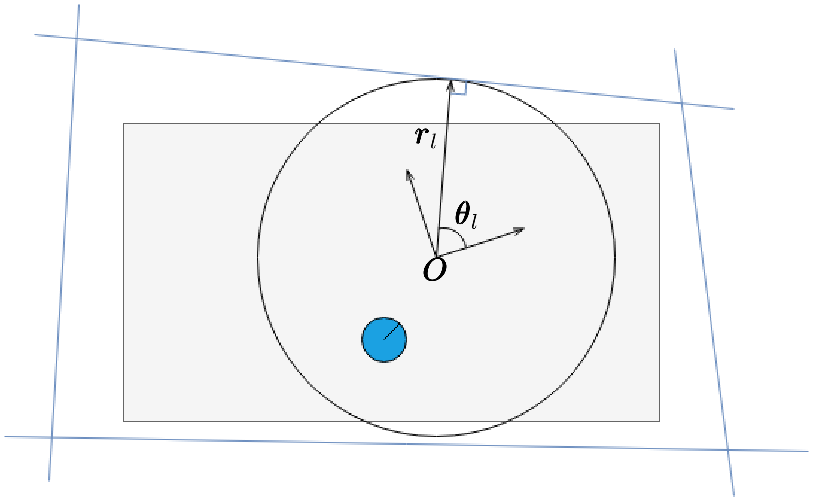


Figure 3.2: Estimating the plate's geometry with a set of lines. Each line is parameterized by a pair of variables (r_l, θ_l).

reflections can be identified by its local maxima.

The map is then represented by a set of lines $\mathbf{M} = \{r_l, \theta_l\}$ where the parameters $(r_l, \theta_l)_{l=1...4}$ define the line equation in the 2D plane as follows:

$$x \cdot \cos\theta_l + y \cdot \sin\theta_l - r_l = 0 \tag{3.3}$$

These boundaries are subsequently detected by constructing first the beamforming map given measurements $z_{1..T}$ obtained all along the robot trajectory $\mathbf{x}_{1..T}$:

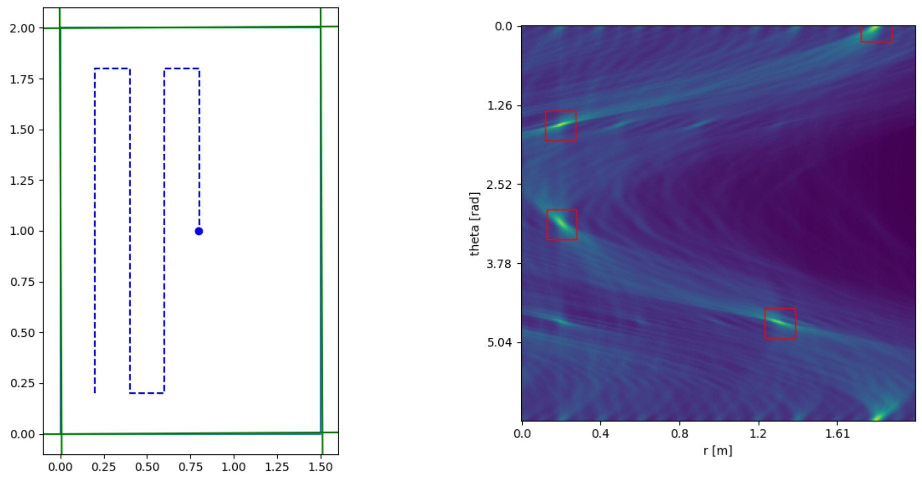
$$\mathcal{L}_T(r,\theta) = \sum_{i=1}^{T} z_i(|x_i \cdot \cos\theta + y_i \cdot \sin\theta - r|) \tag{3.4}$$

where $\mathbf{x}_i = (x_i, y_i)$ is the robot position during time-step i. By solving the optimization problem with the method described in [3]:

$$\hat{\mathbf{M}} = \arg\max_{\mathbf{M}} \mathcal{L}_T(\mathbf{M}) = \arg\max_{\mathbf{M}} \sum_{l=1}^{4} \mathcal{L}_T(r_l, \theta_l) \tag{3.5}$$

where the four lines are restricted to define a rectangle altogether.

The Figure 3.3 depicts an intermediate simulation step in which the sensor moves along a lawn-mower path. The signals are simulated using the measurement model based on the propagation model as explained in [3]. Using the method described, the approximate shape of the plate is recovered.



(a)

(b)

Figure 3.3: (a) The estimated map (green lines) of a $1.5 \times 2\,\mathrm{m}^2$ plate. The blue dot and blue dashed plot represents the position and path of the transmitter/receiver pair. (b) The correlation function's values. The red squares identify the four lines detected by the method.

3.1.2 Occupancy Grid update

We have chosen to use the occupancy map for three main reasons. There is no assumption about the structure of the surroundings of the robot and the uncertainty of the sensor is accounted for by the use of probability theory models. It is capable of maintaining full information on the explored and unexplored areas on the map [18, 19]. Using the first echo, referring to the nearest edge, to update the map, defines a ”safe area” in which the robot can move and estimate the geometry of the plate without surpassing the edges. The beamforming map also benefits from the information gathered in the occupancy grid filtering the lines with parameters (r, θ).

Let the occupancy grid $\mathbf{G}$ be decomposed into $n \times m$ evenly-spaced disjoint grid cells where the i-th grid cell c_i is assigned a static binary variable $o(c_i)$ for $i \in \{1, \ldots, n \times m\}$

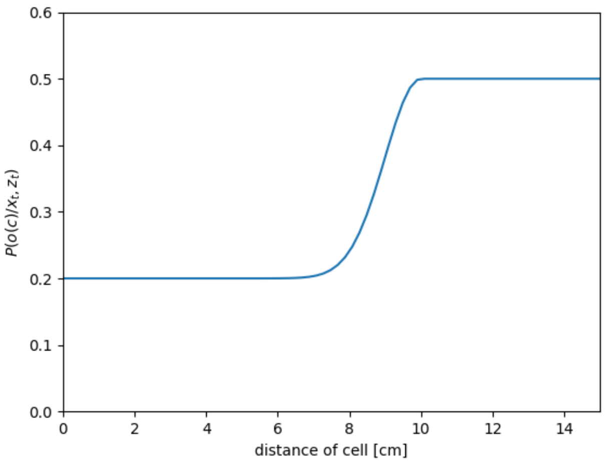


Figure 3.4: Profile of an inverse sensor model illustrating the occupancy probability given a reflection at a distance $d^* = 10cm$ and an uncertainty $\sigma = 1cm$.

that is defined as $o(c_i) = 1$ when occupied and $o(c_i) = 0$ when free. The location and size of each grid cell is assumed known.

To update the occupancy grid at each measurement step, first, we define an operator ϱ that estimates the closest edge reflection from the measured acoustic signal z_t as:

$$\varrho(z_t) = d^* \tag{3.6}$$

Assuming that the occupancy of different cells is conditionally independent with respect to x_t and z_t), we design an analytical form of the Inverse Sensor Model(ISM) (Fig. 3.4) $p(o(c)/x_t, z_t)$ referring to the occupancy probability of a cell c at a distance d from the sensor's position x_t given measurement z_t as follows:

$$p(o(c)/\mathbf{x}_t, z_t) = \begin{cases} \beta + (1-\beta)\lambda e^{-\frac{(d-d^*)^2}{2\sigma^2}}, & d \in [0, d^*] \\ 0.5, & d \in [d^*, d^*_{max}] \end{cases} \tag{3.7}$$

where λ is the scale factor for the Gaussian such that for $d = d^*$: $p(o(c)/\mathbf{x}_t, z_t) = 0.5$. β presents the probability at which the estimation fails completely to estimate the first echo and σ^2 the variance in the estimation reflecting the measurement's uncertainty. The use of 0.5 as a maximum value emphasizes the uniqueness of our problem: unlike most commonly used range sensors, we don't know the angle of incidence, so we can't assume a cell is occupied as the source of the reflection. The possible states remain then unoccupied and unknown during the exploration process.

We note $\hat{o}_t(c_i) = p(o(c_i)/\mathbf{x}_{1:t}, z_{1:t})$ the occupancy probability of grid c_i where $z_{1:t}$ the set of all measurements up to time t, and $x_{1:t}$ is the path of the robot defined through the sequence of all poses. The binary Bayes filter and log odds representation of occupancy are used then to update this probability for each grid cell[18] as follows:

$$l_{t,i} = \log \frac{\hat{o}_t(c_i)}{1 - \hat{o}_t(c_i)} = \log \frac{p(o(c_i)/\mathbf{x}_t, z_t)}{1 - p(o(c_i)/\mathbf{x}_t, z_t)} + \log \frac{\hat{o}_{t-1}(c_i)}{1 - \hat{o}_{t-1}(c_i)} \tag{3.8}$$

The probabilities are easily recovered from the log odds ratio following:

$$\hat{o}_t(c_i) = 1 - \frac{1}{1 + \exp l_{t,i}} \tag{3.9}$$

The Algorithm 1 describes how the occupancy grid update is implemented.

Algorithm 1: Occupancy Grid update

```
Input  : Occupancy Grid $\mathbf{G}_{t-1} = \{\hat{o}_{t-1}(c_i)\}$, log odds $\{l_{t-1,i}\}$, $x_t$, $z_t$
$d^* \leftarrow \varrho(z_t)$
$C_t(x_t, d^*) \leftarrow$ set of the cells in the perceptual field
for $c_i$ in $C_t$ do
    $l_{t,i} = l_{t-1,i} + \log \frac{p(o(c_i)/x_t,z_t)}{1-p(o(c_i)/x_t,z_t)}$
    $\hat{o}_t(c_i) = 1 - \frac{1}{1+\exp l_{t,i}}$
end
return $\{\hat{o}_t(c_i)\}$, $\{l_{t,i}\}$
```

3.1.3 Frontier generation and Beamforming map filtering

After each Occupancy Grid update, each grid cell has a state probability. Using a thresholding method, we assign discrete state to the cells (empty or unknown) resulting in a deterministic world model [19]. We choose a threshold δ based on the chosen ISM in which a grid cell is labeled as free for a state probability lower than δ and unknown otherwise. Frontier points then are generated based on an edge detection technique borrowed from computer vision [20].

The occupancy grid information can be used by the beamforming map to filter lines. Lines defined by (r, θ) parameters that contain a frontier point are filtered using the previously determined frontier points delimiting the free and unknown region.

Given a frontier point p_i with coordinates (x_i, y_i) in the Euclidean space we define the set of lines $l(p_i)$ that passes by the frontier points p_i by:

$$l(p_i) = \{(r, \theta), x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) = r\} \tag{3.10}$$

A mask is defined then to filter lines with:

$$F(r, \theta) = \begin{cases} 0, & if\ \exists p_i, \exists r^* \in [r, r_{max}] : (r^*, \theta) \in l(p_i) \\ 1, & otherwise \end{cases} \tag{3.11}$$

Given the convexity assumption of the geometry of the plate and of the free space defined by the generated occupancy grid, if a line (r_l, θ_l) passes by a frontier point all the lines in the same direction θ_l with a $r \leq r_l$ are filtered.

We reduce then the estimation process to:

$$\hat{\mathcal{L}}_T(r, \theta) = F(r, \theta)\mathcal{L}_T(r, \theta) \tag{3.12}$$

The filtered beamforming map $\hat{\mathcal{L}}_T(r, \theta)$ is then used the estimate the map $\hat{\mathrm{M}}$ as defined

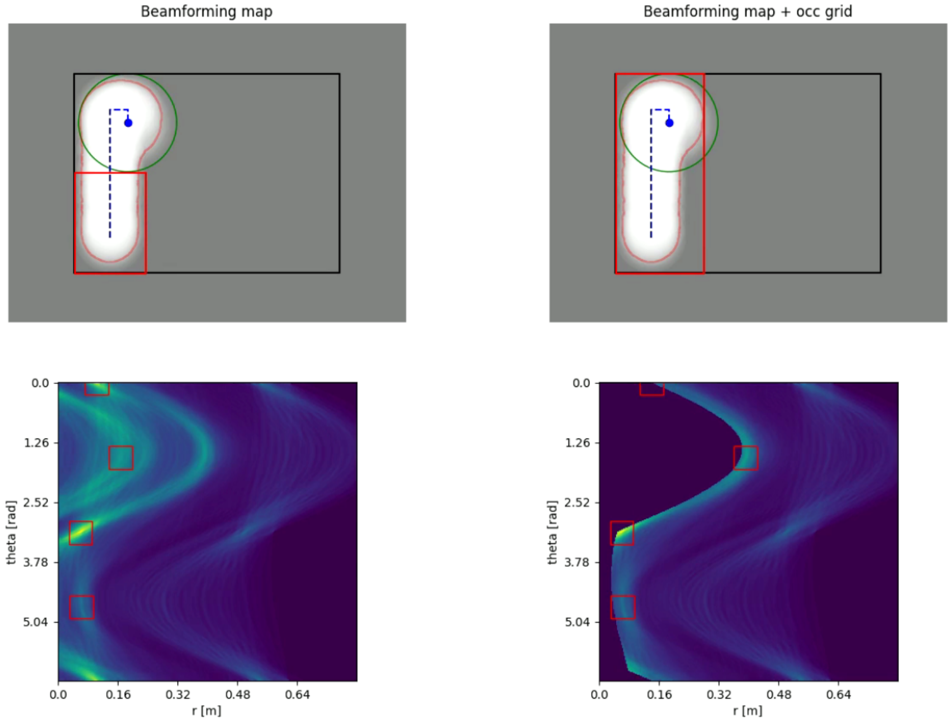


Figure 3.5: In simulation and for plate estimation, comparing the use of the beamforming map versus the use of the beamforming map plus the occupancy grid's information.

in a previous section.

Figure 3.5 illustrates the comparison between using the beamforming method alone versus using it in conjunction with the occupancy grid information. The sensor is following a lawn mower path, and we can see why such a mask would be useful in this intermediate step. The occupancy grid is depicted in the upper two figures, with the black rectangle referring to the real plate and the red rectangle referring to the estimate plate. The sensor's pose is represented by the blue dot, and its trajectory is represented by the dashed blue line. The closest edge echo is indicated by the green circle centered around the sensor with a radius equal to the first echo detected. The associated beamforming maps are shown in the lower figures with the selected line. As can be seen, the beamforming map takes advantage of the data gathered along the path to provide a more accurate estimate and eliminate any potential lines that belong to a free region.

3.1.4 Frontier evaluation

Frontier points evaluation function is the basis for the selection of frontier points. We evaluate the frontier points from the following two factors: information gain at the frontier point and the Euclidean distance to the robot's pose.

The information gain is defined as the area of unknown region expected to be explored for a given frontier point p_k and quantified using Shannon entropy:

$$I(p_k, \hat{\mathbf{M}}) = \sum_{c \in V_k} e(c) \tag{3.13}$$

where V_k is the set of grid cells contained in the expected area of measurement defined by a circle centered in the frontier point p_k with a radius equivalent to the distance to the closest edge given the plate's estimation $\hat{\mathbf{M}}$ (Fig. 3.6), and $e(c)$ the entropy of the grid cell using the occupancy probability $\hat{o}_t(c)$ such as:

$$e(c) = -\hat{o}_t(c) log(\hat{o}_t(c)) - (1 - \hat{o}_t(c)) log(1 - \hat{o}_t(c)) \tag{3.14}$$

We evaluate both unknown regions and regions where the map is still uncertain by taking into account the entropy of both observed and unobserved grid cells.

The second factor is the Euclidean distance $d(p_k)$ from sensor's pose to the frontier point p_k. Each factor of the utility function is subjected to a min-max normalization in order to map its values to a range between 0 and 1:

$$\hat{f}_k = \frac{f_k - min(\mathbf{f})}{max(\mathbf{f}) - min(\mathbf{f})} \tag{3.15}$$

where f_k is the factor associated to the frontier point p_k and $\mathbf{f}$ the set containing the factor associated to all frontier points. Based on both factors, the utility function is then defined as:

$$U(p_k, \hat{\mathbf{M}}) = \alpha(1 - \hat{d}(p_k)) + (1 - \alpha)\hat{I}(p_k, \hat{\mathbf{M}}) \tag{3.16}$$

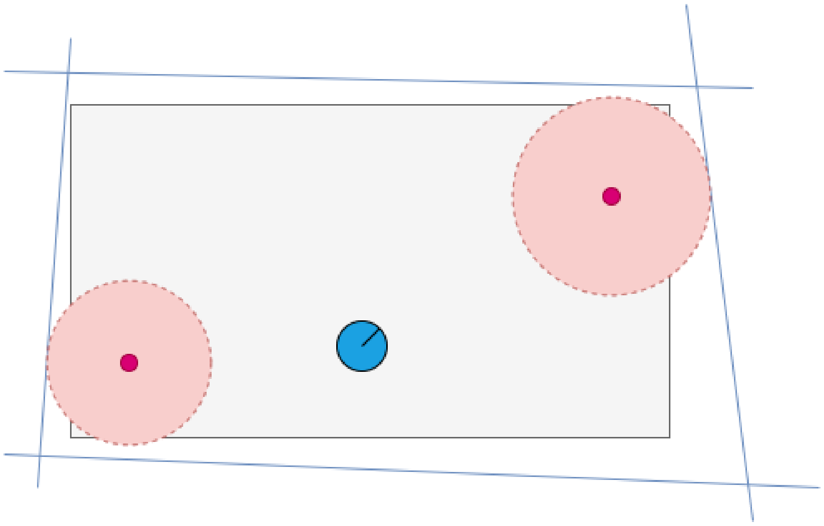


Figure 3.6: Graphic representation illustrating an example of the expected area (pink area) to explore for two frontier points (pink dots) given the plate geometry estimation.

where $\hat{d}(p_k)$ and $\hat{I}(p_k, \hat{\mathbf{M}})$ are the normalized factors and α is a weight parameter that varies between 0 and 1 to adjust the importance of each factor. Given N nominee points, the next goal is the point with the highest utility evaluation as:

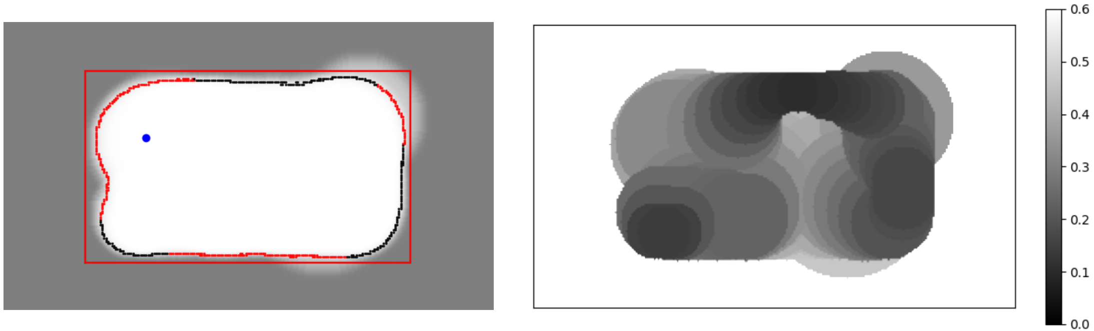
$$p_{opt} = \underset{p_k}{argmax}\{U(p_k, \hat{\mathbf{M}}),\ k \in [1, N]\} \tag{3.17}$$

3.1.5 Repetitive Re-checking and multi-step exploration

The map is updated continuously during navigation to the selected location. As a result, some new frontier points will be generated, some old frontiers will be no longer valid, and the selected point may no longer be the optimal target. It is therefore unnecessary to continue traveling to the chosen location.[17][21] We address this problem by using Repetitive Re-checking (RR) in which the robot checks after each measurement if the target goal is valid given (grid radius). We define also a local exploration path step size s_{exp}. Each time when the movement distance of the robot reaches the step size, the next optimal target is selected using the exploration strategy.

3.1.6 Validity of frontier points and stopping condition

We create another grid layer naming it Radius Grid $\mathbf{R}$ (Fig. 3.7b) with the same dimensions as the occupancy grid $\mathbf{R}$ (ie $n \times m$ evenly-spaced grid cells). $\mathbf{R}$ is updated simultaneously as the Occupancy Grid using the same data (i.e. x_t and z_t). The goal of this grid is to define the validity of a frontier points in terms of its estimated distance from the true edge in order to avoid crossing any plate boundaries. For each grid cell g_i is assigned the minimum measured echo that passed by this cell. The validity of a frontier points p_i is defined by its associated Radius Grid value; if this value is less than a defined threshold ρ, p_i is then labeled as invalid candidate point. This method, which uses the first echo, is a heuristic that gives information if the transducers are too close to a real edge. It also allows us to set a stopping condition for the exploration process: there are no more valid candidate points. Figure 3.7 illustrates the Occupancy Grid as well as the parallel Radius Grid layer. Frontier points p_i in Fig. 3.7a associated with grid cells g_k which values less than the defined threshold ρ are marked as black dots, whereas valid candidate points are represented as red dots.



(a)

(b)

Figure 3.7: (a) The estimated map (red rectangle) of a $1.5 \times 2\,\mathrm{m}^2$ plate. The sensor position is represented by the blue dot. The red (resp. black) dots illustrate valid (resp. invalid) frontier points; (b) The Radius Grid $\mathbf{R}$.

3.1.7 Algorithm description

At each time-step, the robot with a pose x_t has a computed Occupancy Grid $\mathbf{G}$, a Radius Grid $\mathbf{R}$ and a map estimation $\hat{\mathbf{M}}$. First of all, the set frontier points Γ is retrieved from the occupancy grid. Valid candidate points are then defined based on the associated Radius Grid given a threshold ρ. Next, for each candidate point, its utility is computed as defined in section 3.1.4. Finally, the optimal frontier point is retrieved from the set of $\left\{U(p_k, \hat{\mathbf{M}}),\ k \in [1, N]\right\}$. The robot then navigates to the target position. If the goal is reached, no longer valid or the local exploration step s_{exp} is attained, the exploration process is ran again.

```
Algorithm 2: Exploration
  Input  : Occupancy Grid $\mathbf{G}$, Radius Grid $\mathbf{R}$, Robot Pose $x_t$, Plate Estimation $\hat{\mathbf{M}}$.
  Output: Goal pose p$_{opt}$.
  $\Gamma = \{fp_1, fp_2, ..., fp_n\} = GetFrontier(\mathbf{G})$ ;
  $\{p_1, p_2, ..., p_N\} = ValidPoints(\Gamma, \mathbf{R}, \rho)$ ;
  for $k = 1$ to $N$ do
      $d(p_k) = EucilidienDistance(p_k, x_t)$ ;
      $V_k = EstimatedView(p_k, \hat{\mathbf{M}})$ ;
      $I_k = \sum_{c \in V_k} e(c)$ ;
      $U(p_k, \hat{\mathbf{M}}) = \alpha(1 - \hat{d}_i) + \hat{I}_k$ ;
  end
  $p_{opt} = \underset{p_k}{argmax} \left\{U(p_k, \hat{\mathbf{M}}),\ k \in [1, N]\right\}$ ;
```

3.2 Experimental evaluation

The exploration algorithm has been evaluated in both simulated and real world experiments. To evaluate our approach , we present several test conducted in simulation and with real robots. We measure, as exploration time progresses, the distance travelled by the robot and

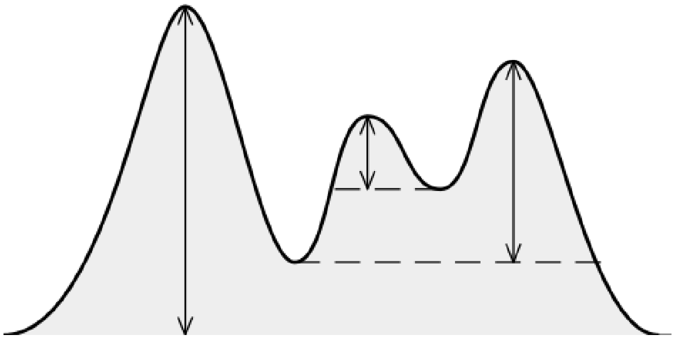


Figure 3.8: Graphic representation of the prominence definition. Vertical arrows depict the topographic prominence of three peaks. Dashed lines represent the lowest contours which do not encircle higher peaks.

the percentage of covered area (as done, e.g., in [13, 14]), namely the percentage of free area relative to the real area of the plate.

3.2.1 Echo detection

First, we illustrate the echo-detection principle. The emitted signal is a 2-cycle burst at 100kHz. We show, in Fig. 3.9, the measured acoustic signal form the plate structure for a sensor position corresponding to a position of coordinates (28, 50) centimeters relative to the plate's corner. On Fig. 3.9.b, we show the resulting correlation signal $z_i'(r)$ computed using eq. 3.1 and it's envelope $z_i(r)$ calculated with eq. 3.2 as explained in [3] yielding the signal which is fed to the algorithm that estimates to plate's geometry.

We use this correlation signal already computed to retrieve to first echo referring to the closest edge to the transducers. The method relies on the detection of the peaks (local maxima) in the correlation based on peak properties. In this case, the prominence is used as the main property to distinguish the echo from noisy peaks given that the higher the prominence is, the more important the peak is. The prominence of a peak is defined as the shortest drop in altitude required to get from the summit to any higher terrain, and this principle is represented in the Fig. 3.8.

To automate the process of the peak detection, first, we retrieve all the peaks of the correlation and calculate the prominence of each peak. Next, we calculate the k^{th} percentile

p^k of these values. Then we recalculate the peaks with a required prominence higher than the percentile value found in the first step. The Algorithm 3 summarizes the described steps.

```
Algorithm 3: Echo detection
  Input  : Correlation signal $z_i(r)$, percentile $p^k$.
  Output: Detected first echo $d^*$
  $peaks = \{e_1, e_3, ..., e_n\} = FindPeaks(z_i(r))$ ;
  $prominences = \emptyset$ ;
  for $k = 1$ to $n$ do
  |   $prominences = prominences \cup PeakProminence(z_t(r), e_k)$ ;
  end
  $p = Percentile(prominences, p^k)$ ;
  $d^* = Findpeaks(z_i(r), p)$ ;
```

Fig. 3.9 illustrates the result of the method for a prominence value $p^k = 85\%$. Fig. 3.9.a shows the acoustic signal detected for transducers placed on a position of $(28, 40)cm$ relative to the plate's corner. In Fig. 3.9.b, the correlation signal calculated given the measured signal $z_i(t)$ and the propagation model is represented. The red line presents the result of the peak detector at $29cm$ and the green line presents the echo expected given the ground truth pose (x, y) relative to the plate's corner which should be defined by $min(x, y, w-x, h-y)$ where (w, h) are the width and height of the plate. The error between the ground truth and peak detected in Fig.3.9 is $1.4\,\mathrm{cm}$.

The parameters used for both the simulated and real-world experiments can be found in Table 3.1.

Table 3.1: Parameters values used in experimental evaluation both in simulation and real-word.

Parameter	β	α	δ	s_{exp} $[cm]$	ρ $[cm]$	σ $[cm]$	$grid\ size$ $[mm^2]$	p^k $[\%]$
Simulation	0.2	0.7	0.2	20	15	2	10×10	80
Real-word	0.2	0.7	0.1	20	15	4	10×10	90

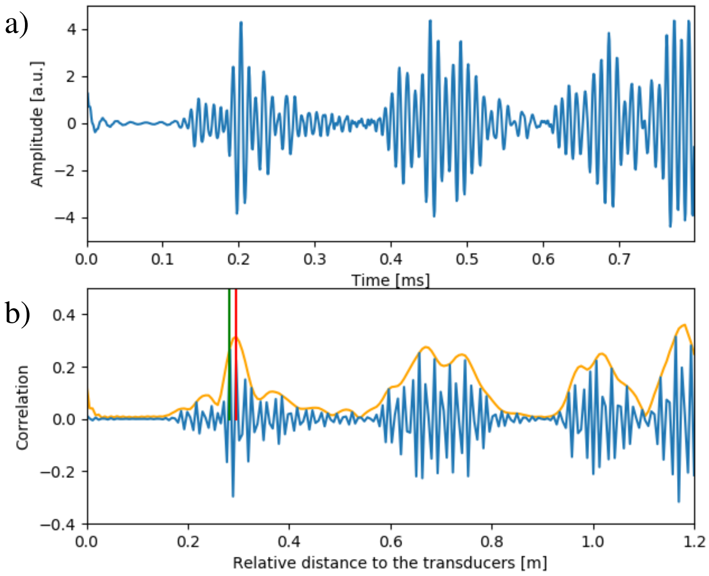
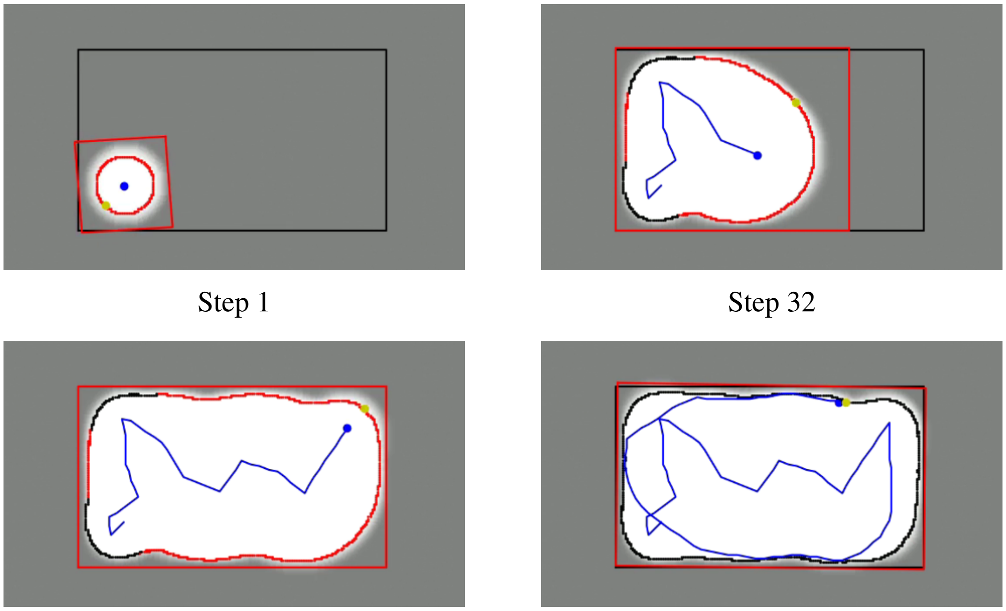


Figure 3.9: Illustration of the echo detection principle for a real signal based on correlation with propagation model in a $1700 \times 1000 \times 6\,\mathrm{mm}^3$ metallic plate. a) the acoustic measurement. b) represents the correlation signal (blue) and its envelope (orange), the retrieved echo (red line) and ground true echo (green line) based on exterior localization.

3.2.2 Simulation

In Python environment, simulations are performed to evaluate the proposed method. The approach is applied to a $1700 \times 1000 \times 6\,\mathrm{mm}^3$ metallic plate. We simulate the two co-located transducers as a particle with a position referring to the central position between the two. The signals are simulated using the measurement model based on the propagation model as explained in [3] and correlation is retrieved as shown in Fig. 3.9.

In Fig. 3.10, we show the occupancy grid and the path followed by the particle to generate it. We also represent the geometry estimated, the valid and invalid frontier points and the selected candidate point based on the utility function defined. In step 1, the estimated plate is incorrectly estimated since only one measurement was integrated. Because all of the points are within the same distance of the sensor position, only the expected area to explore is used to differentiate the utility of the points. The next goal is then randomly chosen from one of the four points pointing to the edges. As in step 32, the three closest edges and the orientation are well estimated. We can see that the sensor first went to the upper right corner until there were no valid points in there, then went to the center of the plate to acquire the maximum of area to explore. In step 55, 78% of the area has been covered and the shape has been approximately fully recovered. Leading to the final step 129, the sensor followed the remaining frontier points until there were no valid frontier points left indicating the end of the frontier exploration process.



Step 1

Step 32

Step 55

Step 129

Figure 3.10: Occupancy grid and path generated by the exploration algorithm. The estimated plate is the red rectangle. The true outline of the plate and the true sensor positions correspond to the black rectangle and blue particle respectively. The valid (resp. invalid) frontier points correspond to the red (resp. black) dots. The yellow particle refers to the selected candidate point.

The following methods are compared to our approach: the classic frontier method (closest frontier point) [12], picking random points from frontier points, our method using the true plate's geometry instead of the estimated map. In the case of closest and random frontier point selection, the sensor moves until it reaches the selected point before moving on to the next location. Otherwise for the third method, it uses the same steps as our proposed exploration approach, except it uses the true plate geometry rather than the estimated map

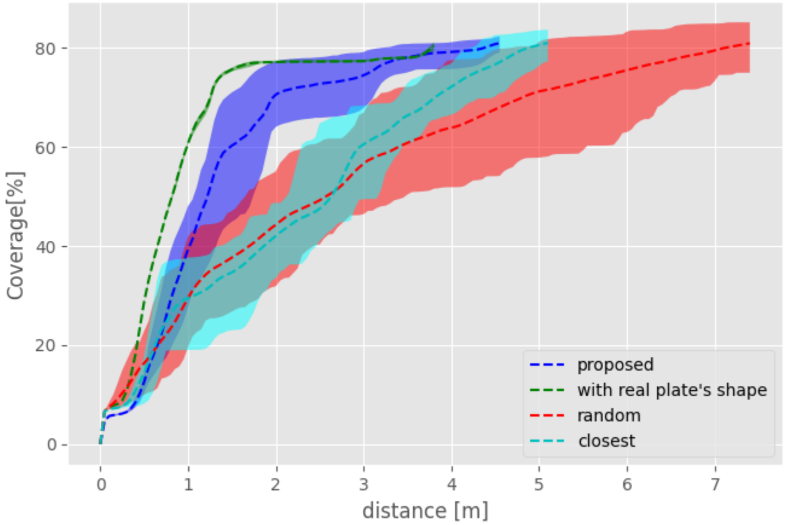


Figure 3.11: Comparison of the proposed method, classic and random frontier method, proposed exploration using the true plate geometry.

$\hat{M}$. To have fair comparison, we present results up to 80% of coverage relative to the true plate's shape, because some runs end without reaching full coverage. We show in Fig. 3.11 the average coverage increase calculated over 50 runs for each algorithm. The same starting position is used for each run corresponding to $20\,\mathrm{cm}$ to the plate's corner. We also represent the 10% and 90% quantiles in order to assess the repeatability of each approach.

The method using the true plate's geometry, shown in green, produces an approximate deterministic result with a low standard deviation, demonstrating the benefit of having a known map. The curve rises quickly, providing more than 60% coverage after $1\,\mathrm{m}$ of displacement. Following that, the rate of coverage gain appears to decrease as it approaches a certain maximum value. The closest selection method grows in a roughly linear fashion. Because all of the points are within equal distance of the sensor, the first goal is chosen at random. The sensor trajectory then takes a path similar to that of an edge following procedure.

On the other hand, though a random approach will eventually produce a map, random choices may not provide an efficient sequence for mapping an environment. Specifically, a random search lacks a mechanism for utilizing navigation cost and information gain to define a suitable goal. As expected, the result underperforms when compared to other algorithms, and the high standard deviation is a direct result of the randomness of the selection process.

Our method, on average, outperforms random and closest frontier exploration as evident by the mean result curves of Fig 3.11. Table 4.1 also is used to confirm this result displaying the length of the traveled path at coverage levels of 70% and 80%.

Table 3.2: Summary of (mean) simulation results using the different methods.

Method	c_{70} $[m]$	c_{80} $[m]$
Proposed	2.0 ± 0.6	4.4 ± 1.1
With Prior	1.3 ± 0.05	3.8 ± 0.05
Closest	3.85 ± 0.6	4.9 ± 0.5
Random	4.85 ± 1.25	7.15 ± 2.2

3.2.3 Real world experiment

A real world experiment was performed in order to validate the feasibility and viability of running the proposed exploration algorithm on real mobile robot. The experiments are conducted on a rectangular metallic plate with dimensions $1000 \times 1700 \times 6\,\mathrm{mm}^3$. The origin of the xy-coordinate system is at the bottom left corner of the plate s shown in Fig. 3.14.b. As show in Fig.3.12, we use two co-located pair of transducers fixed on a spring mechanism mounted on the TurtleBot. This mechanism is used to maintain a constant contact conditions with the surface. A layer of water is added on the plate's surface as a coupling medium for the transducers used in the acoustic measurements. Considering the projection on the plate of both the rotation center of the robot and the middle point between the two transducers, the distance between these two points is $32\,\mathrm{cm}$.

Figure 3.13 illustrates the setup for the acoustic data acquisition. The multifunction I/O data acquisition device NI USB-6341 is used to convert and transfert the acoustic data via

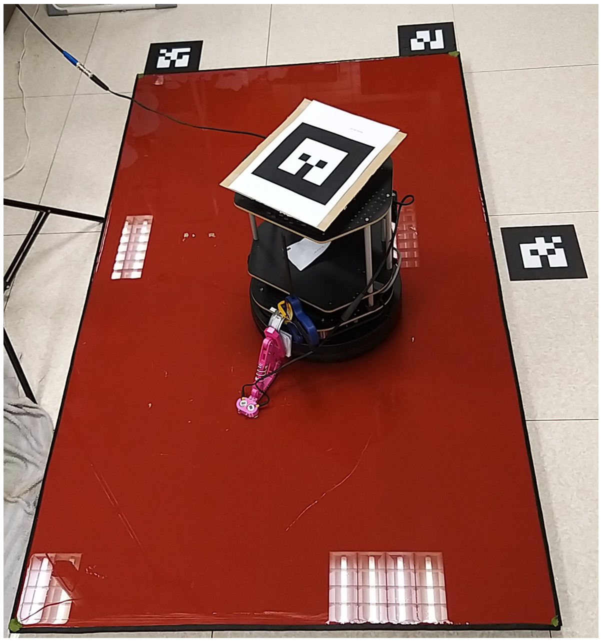


Figure 3.12: Experimental setup. Two co-located pair of transducers fixed on a spring mechanism mounted on the TurtleBot. AR-tags to localize robot's pose relative to the plate's corner.

USB cable.The data for the excitation signal is sent to the NI device via Matlab, and then the signal is generated using the first acoustic transducers. Signals propagates on the plate, therefore the second transducer depicts these signals that are amplified and communicated back to the Matlab interface by the acquisition instrument for processing. We use two tone bursts of a sinusoidal wave at 100 kHz as the excitation and a sampling frequency of 1,25 MHz. Moreover, the direct incident signal is smoothly removed from the data as it does not correspond to a reflection on an edge. This filtering limits the distance at which the signal containing the information about the closest edge can be detected to about $15\,\mathrm{cm}$. This filtering limits the sensor's positioning from the edge. The closest edge can be detected up to about 15 cm. For values less than that, the first echo is misdetected, resulting in an occupancy grid that crosses the real plate's boundaries.

Throughout all the experiments, we use a Realsense Depth Camera D435 with AR tags tracking a Robot Operating System (ROS) package [22] as a external localization system. Four AR tags are used as shown in Fig. 3.14b as ”bundle” representing a single unit.

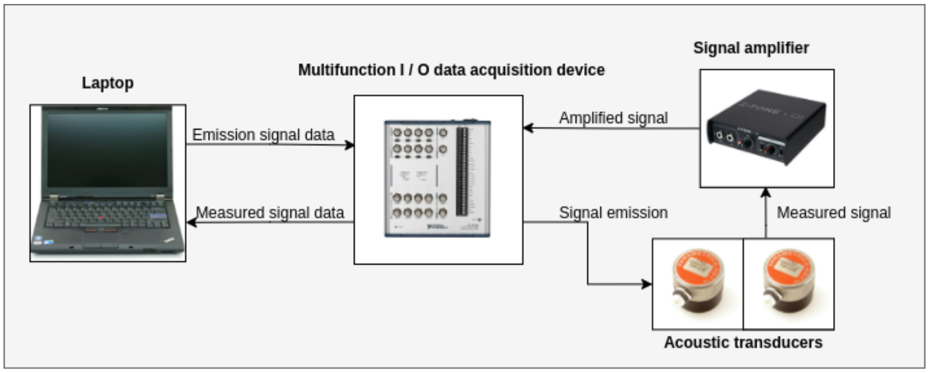
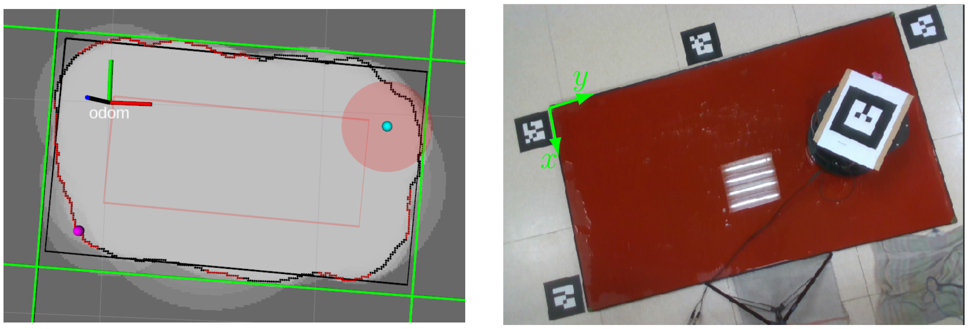


Figure 3.13: Setup for experimental data acquisition of the acoustic signals.

They're put near the plate so that they are visible by the camera. Their positions are measured relative to a "master" tag and fed to the ROS package, estimating then the transform between the two references: robot reference and master tag reference. Given the position of the master tag to the plate's corner, the robot and sensor pose are easily computed. Furthermore, to evaluate the system independently of the quality of the controller, the movement between waypoints is implemented with the joystick. The goal of the project is to apply the method to a real inspection robot equipped with transducers and magnetic wheels. The current robot does not have exactly the same kinematics as the real robot, and in particular it has a much stronger tendency to slide.



(a) (b)

Figure 3.14: Representative intermediate result for the proposed method in a real world scenario. (a) Computed occupancy grid. The blue dot refers to the sensors position whereas the purple dot is the goal position. The estimated plate (green lines) and valid (resp. invalid) frontier points are the red(resp. black) dots. (b) Represents the associated configuration on the lab experiment.

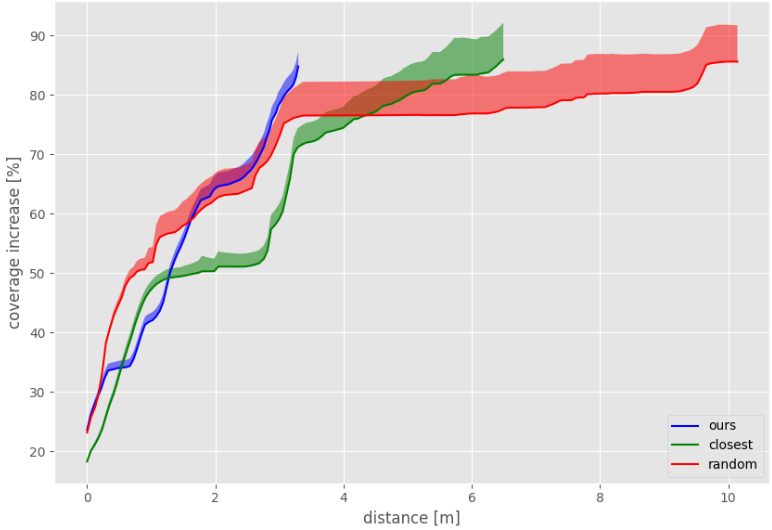


Figure 3.15: Comparing algorithms in real world experiment

The two methods, closest and random frontier selection, are compared to our approach as in simulation. As it may be seen in Fig. 3.14.a, the occupancy grid exceeds the plates real geometry. For that we calculate also the grid cells that were labelled as free outside of the plate. We run each algorithm five times each with the same starting position corresponding to $(25cm, 45cm)$. To have fair comparison and similar to what we did in simulation, we present results up to 86% of the mean coverage, since some runs end without reaching full coverage. As Fig. 3.15 illustrates, the coverage in this case is the full coverage computed during the exploration process minus the error coverage (ie. covered area outside the plate). In the proposed method, the coverage percentage increases fast reaching 86% with a displacement of $2.95m$. We remark that the random approach outperforms our method at the beginning but plateaus around the 75% coverage. The main reason is that random may choose far points which leads to fast coverage increase but fail to get all the details at the end of the day. A summary of the expected sensor's path length is provided in Table 4.2 for a coverage of 70% and 86%.

Table 3.3: Summary of (mean) experiment results using the different methods.

Method	c_{70} $[m]$	c_{86} $[m]$
Proposed	2.75	3.34
Closest	3.28	6.55
Random	2.93	10.47

3.3 Limitations and Discussion

The mapping algorithm performs well in simulation under near-perfect conditions for detecting the first echo. Given the dispersive and multi-modal nature of Lamb waves, as well as the waves' superposition, the automated first echo detection task is more difficult in practice. The ISM proposal aimed to produce decent results even when outliers were present. Otherwise, in real-word experiment, when multiple misdetected signals for first echo are present, the occupancy maps exceed the plate's geometry, filtering out the correct lines and making accurate geometry estimation impossible. Due to the absence of a correction term in the mapping method, the occupancy grid cannot recover from such an error. Figure 3.16 shows an example of how the occupancy map is generated incorrectly by labeling a large area as free while it is outside the plate, which has a negative impact on map estimation, falsely filtering lines in the beamforming map and also leading the sensor to cross the plate's boundaries loosing by that the map estimation. Several factors could cause the echo detection error, including changes in the propagation model due to the use of water as an intermediate medium between the transducers and the plate, no enough water between the transducer and the plate, human error while positioning the ar_tags to locate the robot on the plate and while positioning the spring mechanism on the TurtleBot. Furthermore, due to the removal of direct incident signals, a distance limitation to the edges exists, limiting sensor positioning. When the sensor is too close to the edge, a false detection occurs.

Another limitation in our approach is that even with using the Radius Grid the sensor in some cases gets too close to the edges going bellow the desired threshold ρ.

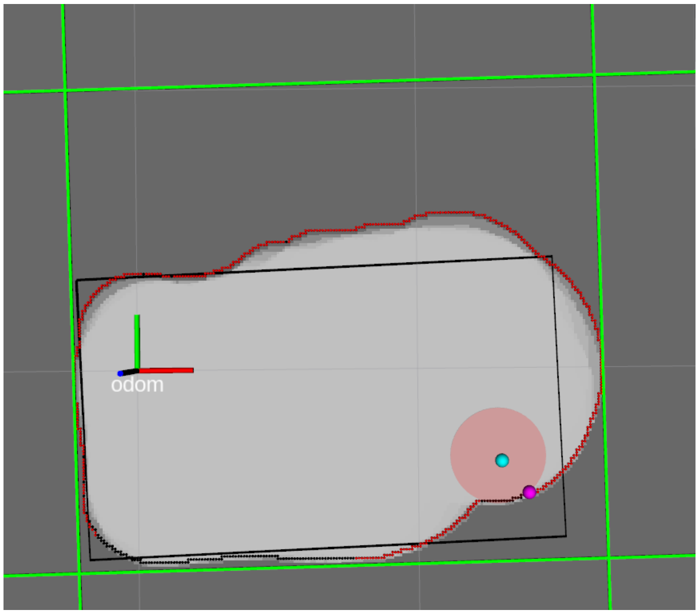


Figure 3.16: Mapping sequence in which the computed occupancy grid significantly exceededs the true plate's edges due to poor echo detection.

Finally, it is assumed the state estimation is perfect. We test our method in simulation and with ground truth localization, but we don't account for pose drifts. However, state estimation error is common and should not be overlooked.

CHAPTER 4

FRONTIER EXPLORATION WITH INTERIOR POINTS GRID

The previous method as discussed suffers from the absence of a correction term in the ISM given a first echo measurement. The mapping method described in this chapter was developed to address this problem. It was proposed by a PhD colleague and will be briefly presented; however, it will remain a work that will be fully presented on a future paper.

4.1 Method

On the one hand, the beamforming map can benefit from the use of a grid that presents cells belonging to the plate in order to filter interference (the propagation is multi-modal). Such a grid is defined by:

$$O(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} = [x, y] \text{ is a point from the plate} \\ 1, & \text{otherwise} \end{cases} \tag{4.1}$$

This grid, that will be referred to as interior points grid, is created using the measurement of the first echo, which will allow the interior points to be determined. On the other hand, the key idea is that the beamforming map can be used to determine the direction of arrival of this same first echo. Knowing this direction in addition to the distance, we can estimate the points outside the plate for that specific direction. A method based simply on the estimation of interior points does not allow such a correction, which is critical for filtering the beamforming map.

From the measurements z_{1t} obtained at positions x_{1t}, we build the beamforming map:

$$\hat{\mathcal{L}}_T^{intf}(r, \theta) = \sum_{i=1}^{T} z_i(|x_i \cdot \cos\theta + y_i \cdot \sin\theta - r|) \tag{4.2}$$

Let us simply denote by $\mathcal{L}_T(r,\theta)$ the same beamforming map which would contain no interference. Assuming that these are mainly observed due to close echoes, we want to estimate it using a mask:

$$Q(r,\theta) = \begin{cases} 0, & \text{if there's a point } \mathbf{x} \text{ from the line}(r,\theta) \text{ such as } O(\mathbf{x}) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{4.3}$$

where $O(\mathbf{x})$ corresponds to the belonging grid to the ”real” plate. Given this grid, we reduce the estimation process to simply applying a binary filter:

$$\hat{\mathcal{L}}_T(r,\theta) = Q(r,\theta)\hat{\mathcal{L}}_T^{intf}(r,\theta) \tag{4.4}$$

$O(r,\theta)$ and $Q(r,\theta)$ is of course not known in advance, it must be estimated at each instant in a similar manner to the Occupancy Grid, and we will construct $\hat{O}_t(\mathbf{x})$ and $\hat{Q}_t(r,\theta)$ such as:

$$\hat{\mathcal{L}}_T(r,\theta) = \hat{Q}_t(r,\theta)\hat{\mathcal{L}}_T^{intf}(r,\theta) \tag{4.5}$$

In our approach, $\hat{O}_t(\mathbf{x})$ and $\hat{Q}_t(r,\theta)$ can have continuous values ranging from 0 to 1 (soft masking). Knowing one should ideally allow to find the other, and vice versa.

The method used to construct the two masks is the binary Bayesian filter in its logarithmic form and with the inverse models as in section 3.1.2:

$$\log\frac{\hat{O}_t(\mathbf{x})}{1-\hat{O}_t(\mathbf{x})} = \log\frac{\hat{O}_{t-1}(\mathbf{x})}{1-\hat{O}_{t-1}(\mathbf{x})} + \log\frac{p(O(\mathbf{x})/x_t, z_t, \hat{\mathcal{L}}_{t-1})}{1-p(O(\mathbf{x})/x_t, z_t, \hat{\mathcal{L}}_{t-1})} \tag{4.6}$$

The same equation is applied for $\hat{Q}_t(r,\theta)$ calculation. The main problem, then, is to specify appropriate models for $p(O(\mathbf{x})/x_t, z_t, \hat{\mathcal{L}}_{t-1})$ and $p(Q(r,\theta)/x_t, z_t, \hat{\mathcal{L}}_{t-1})$. The key ideas are, first, the introduction of a probabilistic model for the imperfect estimation of the first echo, and second, using the edges' estimation to determine the angle of incidence i.e.

suppose that the observation of the first echo is due to an edge of parameters $(\hat{r}, \hat{\theta})$.

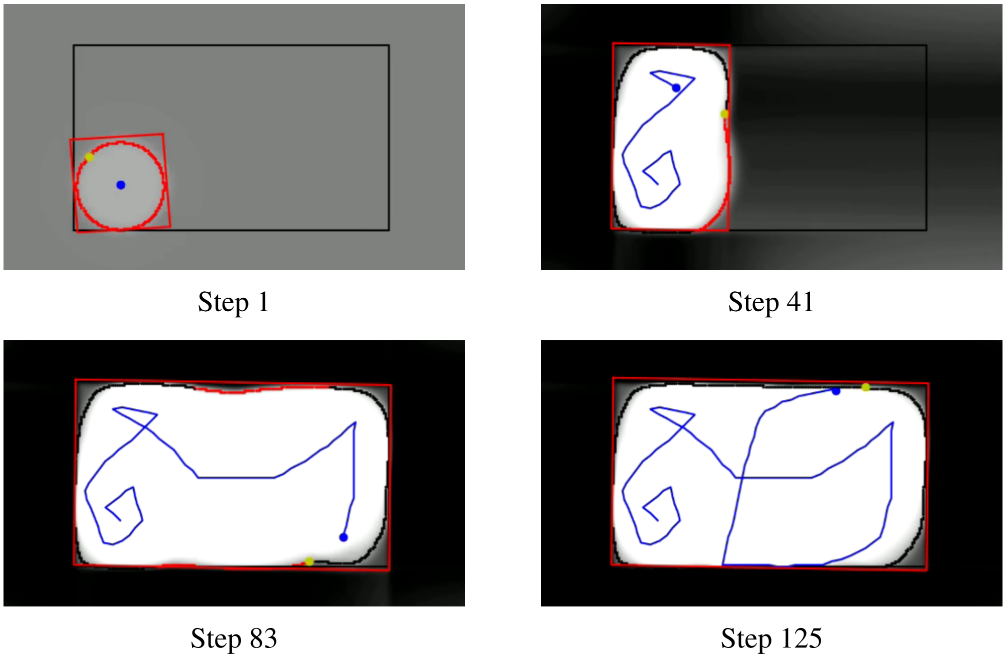
4.2 Experimental evaluation

The exploration algorithm has been evaluated in both simulated and real world experiments using the map generated by the new proposed mapping method. To evaluate the approach combined with the frontier exploration, we present several test conducted in simulation and with the real robot. We measure, as exploration time progresses, the distance travelled by the robot and the percentage of covered area, namely the percentage of free area relative to the real area of the plate.

The simulation and experiments are conducted in a similar setup to the one described in section 3.2.

4.2.1 Simulation

In Fig. 4.1, we show the interior points grid and the path followed by the particle to generate it. We also represent the geometry estimated, the valid and invalid frontier points and the selected candidate point based on the utility function defined. Overall, compared to the method described in chapter3, grid value range from 0 to 1. 0 referring the points that belong to the plate, 0.5 for unknown region and 1 for points exterior to the point. Cells with a value below 0.3 are labelled as interior points. In step 1, the estimated plate is incorrectly estimated as expected since only one measurement was integrated. Because all of the points are within the same distance of the sensor position, only the expected area to explore is used to differentiate the utility of the points. The next goal is then randomly chosen from one of the four points pointing to the edges. As in step 41, the three closest edges and the orientation are well estimated. We can see that the sensor moved closely to the left part of the plate until no valid candidate points remained, then went to the center of the plate to acquire the maximum of area to explore. In step 83, the shape has been approximately fully recovered and 83% of the plate has been covered with the interior points grid.. Leading to the final step 125, the sensor followed the remaining frontier points until there were no valid frontier points left indicating the end of the frontier exploration process.


Step 1

Step 41

Step 83

Step 125

Figure 4.1: Interior points grid and path generated by the exploration algorithm. The estimated plate is the red rectangle. The true outline of the plate and the true sensor positions correspond to the black rectangle and blue particle respectively. The valid (resp. invalid) frontier points correspond to the red (resp. black) dots. The yellow particle refers to the selected candidate point.

As in section 3.2 the following methods are compared to our approach: the closest and random selection of frontier points, our exploration method using the true plate's geometry instead of the estimated map. We present results up to 85% of coverage relative to the true plate's shape. We show in Fig. 4.2 the average coverage increase calculated over 50 runs for each algorithm with the same starting position corresponding to $20\,\mathrm{cm}$ to the edges in a corner of the plate. We also represent the 10% and 90% quantiles in order to assess the repeatability of each approach.

The method using the true plate's geometry, shown in green, produces an approximate deterministic result with a low standard deviation up to 63% of coverage, demonstrating the benefit of having a known map. The curve rises quickly, providing in average 70% coverage after $1\,\mathrm{m}$ of displacement. The closest selection method grows in a roughly linear

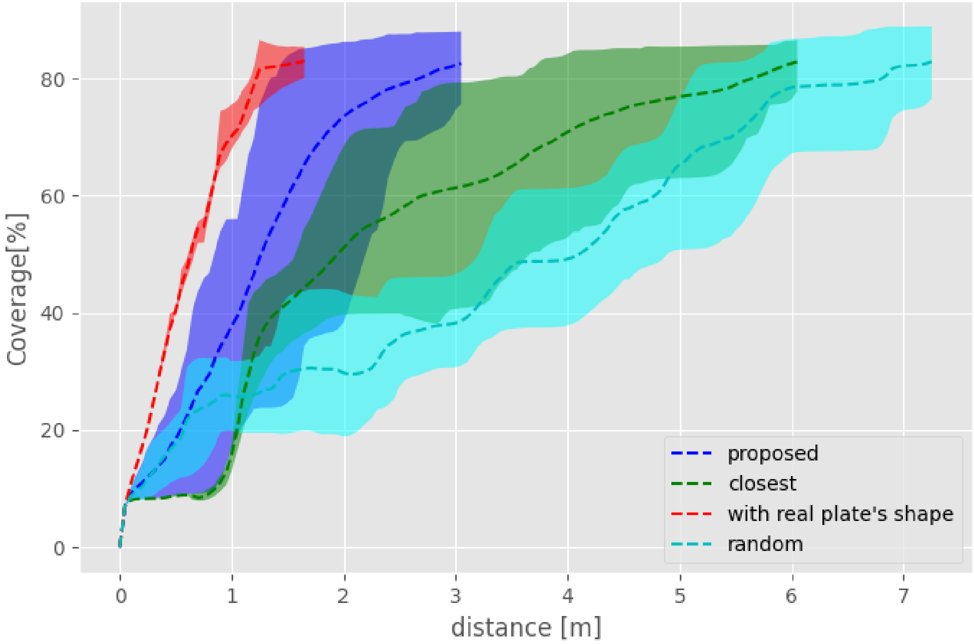


Figure 4.2: Comparison of the proposed method, classic and random frontier method, proposed exploration using the true plate geometry.

fashion with an average coverage rate of $0.15\%/cm$. And in this case presents a behaviour approximately equal to the random approach with a higher standard deviation.

On the other hand, though a random approach will eventually produce a map, random choices may not provide an efficient sequence for mapping an environment. Specifically, a random search lacks a mechanism for utilizing navigation cost and information gain to define a suitable goal. As expected, the result underperforms when compared to other algorithms.

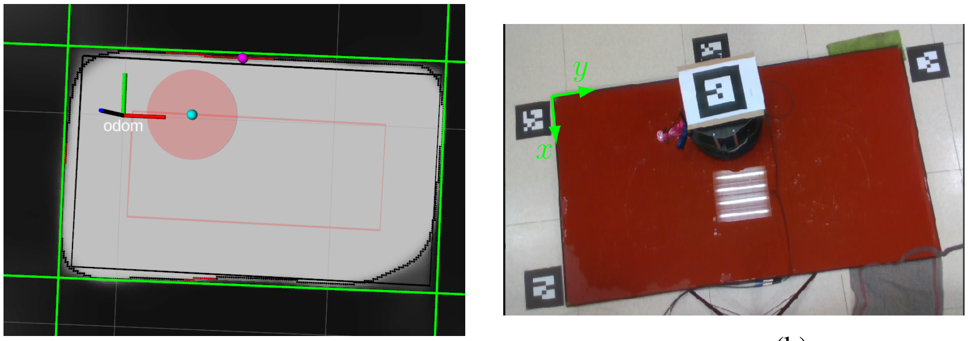
Our method, on average, outperforms random frontier exploration as evident by the mean result curves of Fig 4.2 as it takes into account information gain and navigation cost into account. Table 4.1 also is used to confirm this result displaying the length of the traveled path at coverage levels of 70% and 83%.

Table 4.1: Summary of (mean) simulation results using the different methods.

Method	c_{70} $[m]$	c_{83} $[m]$
Proposed	1.85 ± 0.6	3.1 ± 1.55
With Prior	1.0 ± 0.1	1.7 ± 0.45
Closest	3.95 ± 1.5	6.1 ± 1.9
Random	5.5 ± 0.7	7.3 ± 2.1

4.2.2 Real world experiment

A real world experiment was performed in order to test the mapping algorithm and validate the proposed exploration algorithm on real mobile robot. The same setup described in section 3.2.3 Fig. 3.12 is used. We use two co-localized pair of transducers fixed on a spring mechanism mounted on the TurtleBot. And the experiments are conducted a rectangular metallic plate with dimensions $1 \times 1.7\,\mathrm{m}^2$. Several factors, as mentioned in section 3.3, could contribute to the echo detection error. Additional efforts should be made to reduce various errors.



(a)

(b)

Figure 4.3: Representative intermediate result for the proposed method in a real world scenario. (a) Computed occupancy grid. The blue dot refers to the sensors position whereas the purple dot is the goal position. The estimated plate (green lines) and valid (resp. invalid) frontier points are the red(resp. black) dots. (b) Represents the associated configuration on the lab experiment.

The two methods, closest and random frontier selection, are compared to our approach as in simulation. As it may be seen in Fig. 4.3.a, the occupancy grid exceeds the plates real

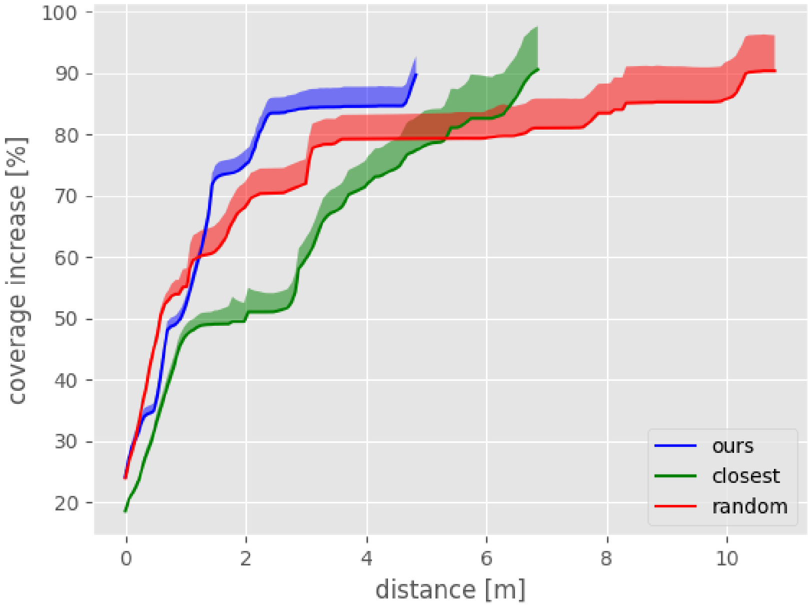


Figure 4.4: Comparing algorithms in real world experiment

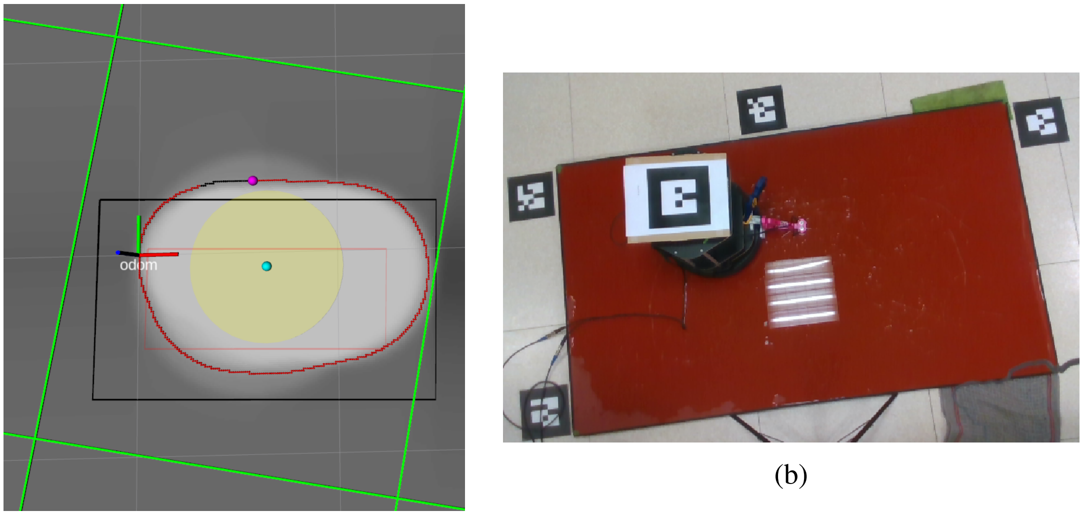
geometry. For that we calculate also the grid cells that were labelled as free outside of the plate. We run each algorithm five times each with the same starting position corresponding to $(25cm, 45cm)$. To have fair comparison and similar to what we did on simulation, we present results up to 91% of the mean coverage. As Fig. 3.15 illustrates, the coverage in this case is the full coverage computed during the exploration process minus the error coverage (ie. covered area outside the plate).

In the proposed method, the coverage percentage increases fast reaching 83% with a displacement of $2.38m$. We remark that the random approach outperforms our method at the beginning as remarked with the previous set of experiments (section 3.2.3) but plateaus around the 80% coverage. The main reason is that random may choose distant points, resulting in rapid coverage increase but fails to obtain all details at the end. The proposed exploration algorithm overperform the two other methods balancing between travelling cost and information gain. A summary of the expected path length is provided in Table 4.2 for a coverage of 80% and 91%.

Table 4.2: Summary of (mean) experiment results using the different methods.

Method	c_{80} $[m]$	c_{91} $[m]$
Proposed	2.22	4.88
Closest	5.41	6.91
Random	6.64	11.18

4.3 Limitations and Discussion



(a)

(b)

Figure 4.5: Representative intermediate result failing to map the plate.

Even tough, the mapping method was developed to have a correction term it still fails to recover given misdetected first echoes and while the plate is not properly estimated. Figure 4.5 shows an example in which the generation of the occupancy map fails as explained. Furthermore, due to the filtering of the signals a limitation relative to the distance to the edges exists. In our case it's approximately $20\,\mathrm{cm}$. If the robot gets any closer, the first echo detection is inaccurate as it's data has been filtered. Several factors, as mentioned in section 3.3, could contribute to the echo detection error. Additional efforts should be made to reduce these various errors.

CHAPTER 5
CONCLUSION AND FUTURE WORK

UGWs have proven to be useful for plate geometry reconstruction, leading to an innovative solution to the mapping problem. Nonetheless, this solution was evaluated without using a real robotic platform and with pre-defined robot paths. In this thesis, I demonstrated the utility of using the occupancy grid in conjunction with the beamforming map to achieve a fast and accurate reconstraction of the plate using a real mobile robot. The main constraint that the sensor must remain on it without crossing the edges. In order to gain the most information from the environment, I proposed an algorithm, based on the frontier exploration method, that takes into account the estimated geometry of the plate balancing the information gain and the travel cost. On the one hand, my proposed mapping solution performs well for good first-echo-detection; however, the lack of a correction term on the ISM limits its applicability in the real world when the data contains significant amplitude noise. The detection error is caused by a variety of factors, and additional efforts should be made to reduce these various errors. The proposed exploration framework, on the other hand, has been evaluated using quantitative metrics as well as in comparison to alternative mapping approaches, and it has been demonstrated that it outperforms both closest and random frontier point selection on simulation and real-life experiments.

When using a more sophisticated mapping method that combines plate estimation and interior point grid estimation, the map benefits from a correction term that, in some cases, corrects mislabeled grid cells, increasing the chances of successfully exploring and mapping the rectangular plate.

5.1 Future Works

In future works, this method can be tested and adapted in more realistic scenarios. In reality, due to factors such as poor surface quality, anti-fouling coating on the plates, or wave scattering caused by the welds that connect the various plates together, more complex and noisy signals are expected on a large metal structure in an outdoor environment. The automated first-echo-detection can be further investigated to have a more robust method. Furthermore, the exploration method can be integrated with a SLAM framework to account for the state estimation error. Finally, Robot dynamics can be considered when selecting frontier points in order to minimize velocity changes and maintain a consistent high speed for fast reconstruction and exploration of the plate.

REFERENCES

[1] "Bugwright2, autonomous robotic inspection and maintenance on ship hulls and storage tanks, description of the innovative action," 2019.

[2] M. Juliá, A. Gil, and Ó. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, pp. 427–444, 2012.

[3] O. L. Ouabi, P. Pomarede, M. Geist, N. F. Declercq, and C. Pradalier, "A fastslam approach integrating beamforming maps for ultrasound-based robotic inspection of metal structures," *IEEE Robotics and Automation Letters*, pp. 1–1, 2021.

[4] Z. Su and L. Ye, *Identification of Damage Using Lamb Waves: From Fundamentals to Applications*. Jan. 2009, vol. 48, ISBN: 978-1-84882-783-7.

[5] P. Cawley and D. Alleyne, "The use of lamb waves for the long range inspection of large structures," *Ultrasonics*, 1996.

[6] E. Hong and C. Schaal, "Reverse engineering stiffened plates using guided wave-based nondestructive testing methods," in *Health Monitoring of Structural and Biological Systems XII*, T. Kundu, Ed., International Society for Optics and Photonics, SPIE, 2018.

[7] C. Schaal, M. Brown, and K. Schulz, "Experimental investigation of lamb wave-based edge detection methods," Apr. 2019, p. 71.

[8] M. Kreković, I. Dokmanić, and M. Vetterli, "Echoslam: Simultaneous localization and mapping with acoustic echoes," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[9] F. Peng, T. Wang, and B. Chen, "Room shape reconstruction with a single mobile acoustic sensor," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2015.

[10] C. Pradalier, O.-L. Ouabi, P. Pomarede, and J. Steckel, "On-plate localization and mapping for an inspection robot using ultrasonic guided waves: A proof of concept," in *IROS*, IEEE, 2020.

[11] C. I. Connolly, "The determination of next best views," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 432–435, 1985.

[12] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. ’Towards New Computational Principles for Robotics and Automation’*, 1997, pp. 146–151.

[13] H. H. González-Baños and J.-C. Latombe, “Navigation strategies for exploring indoor environments,” *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[14] F. Amigoni and A. Gallo, “A multi-objective exploration strategy for mobile robots,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 3850–3855.

[15] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, “Information based adaptive robotic exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, 540–545 vol.1.

[16] A. Miranda, J. V. Hook, and C. Schaal, “Lamb wave-based mapping of plate structures via frontier exploration,” *Ultrasonics*, vol. 110, p. 106 282, 2021.

[17] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “A comparative evaluation of exploration strategies and heuristics to improve them,” in *ECMR*, 2011.

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005, ISBN: 0262201623.

[19] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 116–121.

[20] L. Roberts, *Machine Perception of Three-Dimensional Solids*. Jan. 1963, ISBN: 0-8240-4427-4.

[21] B. Fang, J. Ding, and Z. Wang, “Autonomous robotic exploration based on frontier point optimization and multistep path planning,” *IEEE Access*, vol. 7, pp. 46 104–46 113, 2019.

[22] S. Niekum, *Ros package: Ar_track_alvar*. [Online]. Available: `http://wiki.ros.org/ar_track_alvar`.