University Institute of Lisbon

Department of Information Science and Technology

Motion-induced sound level control for socially-aware robot navigation

João Paulo Cordeiro Andrade

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master in Computer Engineering

Supervisor

Prof. Dr. Pedro Figueiredo Santana, Assistant Professor

ISCTE-IUL

Co-Supervisor

Prof. Dr. Alexandre Manuel de Castro Passos Almeida, Assistant Professor

ISCTE-IUL

October, 2018

Resumo

Com a crescente presença dos robôs no espaço ocupado pelos seres humanos, é necessário que a relação entre robôs e humans seja natural e não invasiva. Para alcançar este objectivo, é preciso que os robôs tenham formas de assegurar que o ruído causado pelos seus movimentos não incomodam as pessoas inseridas no meio envolvente. Esta dissertação propôe uma solução que permite que um robô aprenda a controlar a quantidade de ruído que produz, tendo em conta as suas caracteristicas e o meio ambiente onde é colocado. O robô concretiza as suas tarefas adaptando a sua velocidade de forma a produzir menos ruído que o presente no meio ambiente e, assim, não incomodar as pessoas. Para que o robô possa executar alguma tarefa num determinado ambiente, é necessário aprender a quantidade de ruído que faz ao movimentar-se a diferentes velocidades nesse ambiente. Para tal, um microfone foi instalado num robô com rodas para obter informação sobre a sua acústica. Para validar a solução proposta, foram realizados testes, com sucesso, em diferentes ambientes com diferentes ruídos de fundo. Posteriormente, foi instalado no robô um sensor PIR para analizar a capacidade do robô executar o controlador de velocidade quando alguém entra no campo de visão do sensor. Este segundo teste demonstrou a possibilidade de incluir a solução proposta em outros sistemas.

Palavras-chave: Robôs sociais · Som acústico · Controlo de movimento · Aprendizagem auto-supervisionada · ROS · Raspberry Pi · Controlo de ruído · Noção de ruído

Abstract

With the growing presence of robots in human populated environments, it becomes necessary to render the relation between robots and humans natural, rather than invasive. For that purpose, robots need to make sure the acoustic noise induced by their motion does not disturb people that are in the same environment. This dissertation proposes a method that allows a robot to learn how to control the amount of noise it produces, taking into account the environmental context and the robot's mechanical characteristics. The robot performs its task while adapting its speed, so it produces less acoustic noise than the environment's, and thus, not disturbing nearby people. Before the robot can execute a task in a given environment, it needs to learn how much noise it induces while moving at different speeds on that environment. For that, a microphone was installed on a wheeled robot to gather information about it's acoustic noise. To validate the proposed solution, tests in different environments with different background sounds were performed. After that, a PIR sensor was installed on the robot to verify the ability of the robot to use the motion controller when someone enters the sensor's field of vision. This second test demonstrated the possibility of using the proposed solution in another systems.

Keywords: Social Robots · Acoustic Noise · Motion Control · Self-Supervised Learning · ROS · Raspberry Pi · Noise control · Noise awareness

Acknowledgements

A special thank you to my supervisor, Professor Pedro Figueiredo Santana, and my co-supervisor, Professor Alexandre Manuel de Castro Passos Almeida, for their untiring support and help, and for pointing me to the right direction through every step of the dissertation.

I would also like to thank ISCTE-IUL and the professors i had contact with, for giving me the opportunity to do the master's degree at their facilities and helping me through the different subjects.

Finally, i would like to thank my parents, brother and friends for their constant motivation and support throughout this stage of life.

Contents

List of Figures

Abbreviations

ROS	**R**obotic **O**perating **S**ystem
RPI	**R**aspberry **PI**
PIR	**P**assive **I**nfra**r**ed sensor

Chapter 1

Introduction

1.1 Motivation

It is not from nowadays that humans dream of having robots serving them and doing their hard work. Examples of important roles that robots may play in the future include advising people on what to buy, accompanying and helping shoppers carrying their bags, executing tasks of high risk to humans, and doing house chores, freeing families to better enjoy their time. These dreams can be seen in all the different media. Books, movies, television content, and video games have been used to tell amazing stories where robots are a part of the civilization and sometimes play a major role on those stories. Today, the world and robots are not yet so advanced, but robots are becoming more present in our society. They are used in different situation. like military applications (Cook, 2007), in our houses with the autonomous vacuum cleaners and even on cars where it is already possible for them to drive without the human intervention(Paden, Čáp, Yong, Yershov, & Frazzoli, 2016).

For a robot to complete any task, it normally needs to process sensors' data properly. With sensors, like infra-red, ultra-sound and microphones, robots can gather data about an environment. These data can be used to avoid obstacles, map a place, follow or guide people. Microphones can be used to detect sound sources (Kallakuri, Even, Morales, Ishi, & Hagita, 2013b) and create an acoustic map (Kallakuri, Even, Morales, Ishi, & Hagita, 2013a) of an environment making the robot more or less noticeable by planning a path far or near those sound sources.

There has been a lot of advancements on the ability to process effectively information from visual sensors, but in comparison, techniques to reduce the robot's noise have not been as much researched. This dissertation aims to take a step forward regarding controlling the robot's noise and contribute to the research community with a new approach that allows a robot to be aware of its own acoustic noise and control it autonomously.

1.2 Context

While there has been a lot of research about analyzing data from visual sensors (Rios-Martinez, Spalanzani, & Laugier, 2015) (Kruse, Pandey, Alami, & Kirsch, 2013), the ability to distinguish different sounds in an environment with a robot and reducing the robot's noise when moving have not been as much researched. Yet, progress has been made, like the work of Martinson (Martinson, 2007) where a robot tries to hide its noise by moving near a sound source in the environment before moving towards the objective and, therefore, being less noticeable. Another way to reduce the impact of a robot's noise is through the creation of acoustic maps (Kallakuri et al., 2013a) (Martinson & Schultz, 2006) (Martinson & Schultz, 2007) that expose where the sound sources are in an environment and, that way, the robot can be as much or less noticeable as possible by planning a path towards an objective by moving far or near those sound sources. It is also possible to use microphones to detect obstacles that are not in the robot's field of vision, for example, when a robot is arriving at an intersection, where it cannot know what is on the other paths, it can listen for moving obstacles on the other paths (Kallakuri et al., 2013b).

The system proposed in this dissertation stands out, because the robot does not depend on sound sources that may be on the environment to be less noticeable. Regardless of having a lot or little noise in the environment, the robot is aware of how much noise is producing and can produce, enabling it to quickly adapt to high variations of the environment's noise to complete its task as quickly as possible while not disturbing humans nearby with its noise. To do so, the robot processes the environment's acoustic noise at any moment and, based on a previously learned information about its own noise in that environment, selects the highest speed it can move, that does not produce more noise than the already present in the environment. This solution discards the necessity for creating acoustic maps, because by knowing how much noise the robot produces at different speeds in an environment, those maps are not required if the objective is solely to not disturb humans nearby while conducting an activity. Nevertheless, the solution can also be used to complement other solutions and be integrated into a system more complex. For example, if this solution were integrated with the work of Martinson (Martinson, 2007), because the robot is aware of its own noise, the robot can hide significantly better its acoustic noise, and thus, making it harder to be detected. Another advantage is in a scenario where a sound source, that the robot is using to conceal its noise, stops emitting sound. In this situation, the robot would be easily detected because it would keep producing the same amount of noise and not adapt itself. By using the solution proposed in this dissertation, the robot knowing that a person is in the environment, can adapt the amount of noise it is producing by reducing its velocity as soon as the sound source stops producing sound, and with that, become harder to detect.

1.3 Research questions

This dissertation aims at answering a few questions, with the main one being:

- Is it possible to learn a robot's acoustic impact on a human populated environment, based on data gathered by one or more microphones placed on the robot?

In order to understand the effect of a robot's acoustic noise in an environment, it is necessary to measure the robot's acoustic noise while performing a task. By placing a microphone near a robot's sound source, like wheels motors, it is possible to measure the robot's noise when moving. One way of learning the robot's noise is by gathering its noise through the microphone when it is performing a task, at different speeds and when there is as low background noise as possible. This is to capture only the robot's acoustic noise. Having the information about the relation between the speed and the acoustic noise, allows to control the robot's noise, and thus, predicting the impact the robot's acoustic noise will have in an environment.

Other questions aimed at by this dissertation are:

- What is a good spacial distribution of a set of microphones in a robot that allows for a better measurement of the robot's acoustic noise, at different environments?
- Which movements should a robot perform during a learning phase, for a better measurement of the robot's acoustic noise, in different environment's?

Regarding the spacial distribution of the microphones, these should be placed near the robot's sound sources. If all sound sources are placed near each other, maybe one microphone could be enough, but if the robot's sound sources are far from each other, like a rather big robot, then multiple microphones may be required to measure the noise. For a learning phase, the robot should perform the same movements that it would make for the task it is going to do on the environment.

1.4 Objectives

Robot safe navigation in human-populated environments has been one of the most research topics regarding the robotic's field. Such advancements allowed for self driving cars becoming a reality (Paden et al., 2016). In order for robots to be accepted in different environments like factories, households and offices, it is necessary that they navigate in predictable and non-disturbing way. To solve this issue, a field, socially-aware robot navigation, gives the robot's navigation system explicit knowledge of human behavior, including cultural preferences (Kruse et al., 2013) (Rios-Martinez et al., 2015).

It is already possible to place a robot in an environment and analyze the environment's acoustic automatically (Kallakuri et al., 2013a) or controlled by a human (Martinson & Schultz, 2006), to create an acoustic map and perform tasks with that information. There has been effort to reduce the impact of the robot's noise, for example, by making it move near sound sources (Martinson, 2007).

Robot's motion planner should take into consideration personal spaces, to not invading them, and also subjective aspects related to the acoustic impact robots have over the environment. These considerations go along the current knowledge about the relevance of adequate noise prevention and mitigation strategies for public health (Basner et al., 2014). Therefore, the robot should be able to induce more or less acoustic noise depending on the environment the robot is in, e.g., a library or a restaurant.

The aim of the system proposed in this dissertation is the ability to control the amount of noise the robot does when moving. Regardless of the robot's activity, the amount of noise it produces can be unsettling and cause discomfort to nearby humans. Therefore, it is necessary to control the amount of noise a robot produces, as to become irrelevant for humans to have a robot nearby performing tasks if the robot does not disturb them because of their noise. To be able to do that, a robot is equipped with a microphone to listen to its own noise when moving. The position of the microphone is something that needs to be addressed, as it is necessary to place it in several positions on the robot, for example near the wheels or far from them, to understand where is the best position for the microphone that better captures the robot's noise and differentiates it better from the environment's noise. By listening to its own noise when moving at different velocities, it becomes aware of its impact on the environment and, in several situations, when it detects a person, it can automatically adapt itself to produce less noise, by moving slower. This knowledge can then be stored in a database as to avoid re-learning its noise levels every time a task is performed in an environment. This ability to adapt autonomously to the environment the robots are in as to not disturb humans is fundamental for them to be accepted as part of our daily lives. Figure 1.1 shows a representation of the proposed method's use case. Both top-left and bottom-left images represent the initial state where the robot is idle at an environment where there are some people having conversations. The difference is that at the top images, the robot does not use any noise controller, while in the bottom images, the robot uses the proposed noise controller. At the top-right image, the robot moved near humans while making considerable noise, causing discomfort and forcing people to speak louder to continue the conversations. At the bottom-right image, because the robot is moving slower, it does not make more noise than the people and executes its task while people continue to talk to each other normally. The squared object with two semi circle at the sides is the robot. Humans are represented by circle with a more flatten circle, somewhat similar to a plus signal. The dialogue balloons represent the conversations and the bigger the balloon and the letters, the higher the people are talking. The dotted lines represent the path of the robot and the semi circles nearby represent the noise the robot is producing, where the higher quantity of semi circles, the more noise.

1.5 Research method

To enable this dissertation to be an important contribution in the robotic field and to the coexistence of autonomous robots in our society, this project is organized into different sections until the objective is achieved, that is the ability of

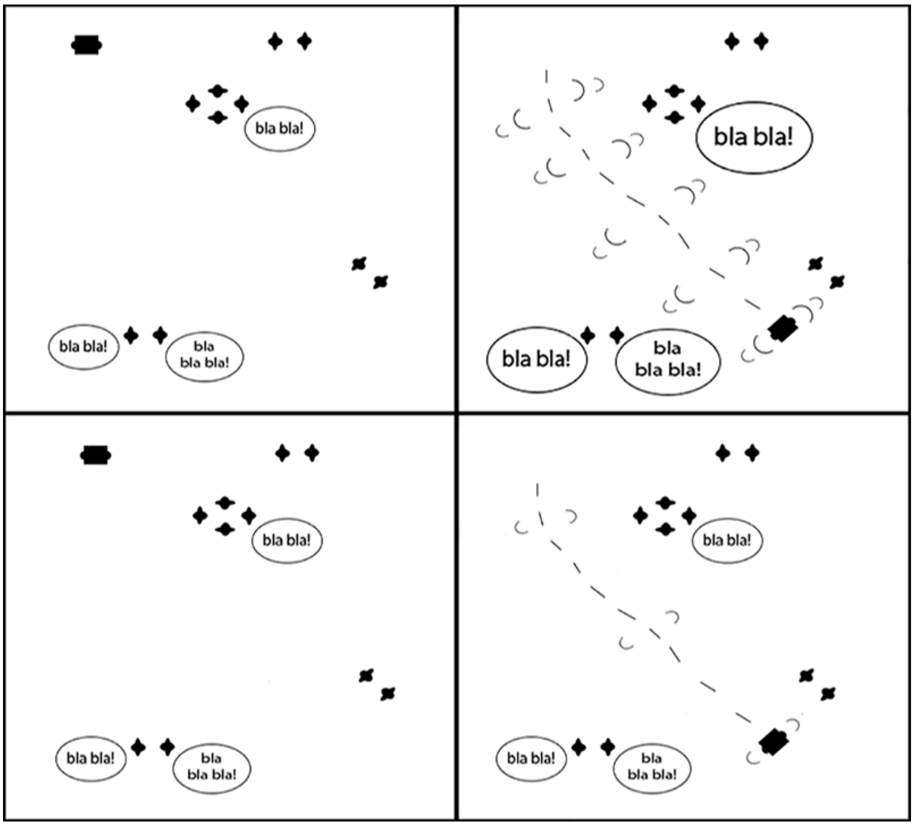


Figure 1.1: A cartoon representation of the proposed method's use case. By controlling the robot's (square object with two semi circles at the sides) noise through it's speed, nearby people are not upset by the robot's acoustic noise. This is visible by the dialogue balloons, where people do not speak in a higher volume when the robot is passing by at a controlled speed.

having autonomous robots perform a variety of tasks in any environment without disturbing humans nearby.

At a first stage, research was made about what is already possible and investigations related to the dissertation subject, to understand how this dissertation would contribute to the investigation community and to establish its objective.

With the objective and contribution established, the development stage, or second stage, would start. It was necessary to research what tools and hardware to use. A Turtlebot 2.0 robot, a laptop with ROS (robot operating system) where the main system runs, and a microphone connected to a Raspberry pi or Arduino to send data between the microphone and the laptop. The first step in this stage was to acquire a microphone and the Raspberry or Arduino and be able to transfer the microphone data to ROS. It was necessary to understand if it was possible to distinguish different contexts, or environments through the sound captured by the microphone and if it was possible to differentiate different sounds. These environments are an array of rooms and an outdoors space.

In a third phase, the microphone was placed on the robot in different position to understand what would the best position that captures and separates better the robot's noise from the environment's. The objective was to achieve that with only one microphone to avoid making the system more complex than necessary at an early phase of development. To make the system more robust, it can be extended, at a later development stage, to consider an array of microphones placed in different positions. After finding the ideal position, the robot was placed in different environments performing tasks, like moving, to understand the amount of noise the robot produces in each environment. The next step was to develop a system that makes the robot produce less noise. One way to do that is by reducing the velocity that uses to execute tasks. The ideal is to make the robot perform tasks without people nearby realizing the robot's presence through sound.

The fourth phase was the testing phase. The robot was placed at the same environments as the previous phase and test the algorithm. Different sounds were played at different volumes, and the system should be able to select a velocity that makes the robot produce less noise than present in the environments. This should ensure that the robot does not produce more noise than the already present, and thus, not cause discomfort to nearby humans because of its noise. To test the algorithm in a possible real situation, the robot was placed in the same environments equipped with a PIR sensor and ready to perform a task. When the PIR detects a person, the robot should be able to adapt its velocity, which indicates that the system could be integrated into a more complex system.

The fifth phase was the writing of this dissertation.

Chapter 2

Related work

2.1 State of the art

To have robots performing a variety of tasks near humans, they need to be able to navigate safely through an environment populated with humans. Fortunately, robots are being more socially-aware as this field is being improved over time. Socially-aware robot navigation has already been tested in different situations, for example, office locations, houses, museums (Kruse et al., 2013) (Rios-Martinez et al., 2015) and factories (Michalos et al., 2015) (Marques, Gonçalves, Barata, & Santana, 2017). Besides that, robots need to ensure that they respect the human space to not cause discomfort in human-robot interactions, according to the theory of proxemics for human-human interactions (Hall, 1969), which indicates that comfort is a function of the distance between the participating agents and their relationships, cultures and intentions. Contemporary socially-aware robot navigation has included these concepts to ensure more natural interactions between humans and robots (Takayama & Pantofaru, 2009) (Luber, Spinello, Silva, & Arras, 2012) (Rios-Martinez et al., 2015).

In this chapter a review over what the community has already accomplished and that is related to the objective of this dissertation is exposed. After that, the different technologies used on this project are exposed, starting by the main framework used and the backbone of the project, ROS that is a library designed for robotic systems. After an introduction to ROS, an overview of the different hardware used, namely, the robot Turtlebot, the microphone, the Arduino and Raspberry pi is given.

2.1.1 Noise detection

2.1.1.1 Preliminaries on Acoustics

Sound is a vibration that can travel through a medium, like air, gas or liquid. This vibration can be described as a wave of pressure, which is perceived as sound when received by human hears. Humans can perceive sound when this comes within a range of frequencies between 20Hz and 20000Hz. Some factors about how the sound is perceived can be found on those waves: pitch, which is the perception of the fundamental frequencies of the sound, can be described as the ability to perceive the sound between bass and treble; duration relates to how long a sound is; loudness refers to the amplitude of the sound wave; timbre is the ability to distinguish different types of sound. The perception of the sound is influenced by the distance from the source and by the environment's characteristics, like a room's design, the furnishing or the surfaces. As the distance between a person and the sound source increases, the sound level perceived by the person decreases in a non-linear way.

The wave of pressure, also known as sound pressure, is the difference between the pressure originated by a sound wave and the ambient pressure, the wave is passing through. With a microphone, it is possible to measure the sound pressure level (SPL), which is the logarithmic measure of the effective pressure of a sound relative to a reference value. Normally, on air, the reference value is 20 μP, which is the threshold of the human earing. This value differs from other mediums, like on water, where the reference value is 1 μP Sound pressure level. Denoted L_p and measured in dB, is defined by

$$L_p = 20 \log \frac{p}{p_0} [dB], \tag{2.1}$$

where p is the root mean square sound pressure and p_0 is the reference sound pressure (normally the lowest threshold of human hearing, 20 μPa).

Since microphones have a transfer factor or sensitivity given by some value in mV/Pa, in a particular context or environment we can relate the sample amplitudes acquired by the microphone to the strength of the acoustic signal (pressure level). This can be represented by the simplified (un-weighted) linear sound pressure level (LSPL) given by:

$$LSPL = \frac{1}{N} \sum_{i=1}^{N} \mid x_i - \bar{x} \mid, \tag{2.2}$$

where N is the number of samples per second, x_i is the sampled amplitude, and $\bar{x}$ is the sample mean value.

2.1.1.2 Masking the robot's acoustic noise

An important aspect that causes discomfort to humans is the acoustic pollution caused by robots. One way to reduce this acoustic impact in people-populated environments is by choosing paths that dictate the robot to move near sound sources present in the environment, as tested by E. Martinson (Martinson, 2007). In his work, where the goal is to hide the robot's acoustic noise, the robot moves to a near sound source before moving to its objective. Two tests were performed, and in both, it was analyzed the robot's performance when moving directly to its target and using the proposed algorithm to hide its noise. On the first test (see Figure 2.1), the robot was placed in an open space and the sound source was far from the walls and it was pointing to the robot's final destination.

When the robot chose the shortest path, it was detected about 76 % of the time and when following the longer path, to mask its acoustic noise with the sound

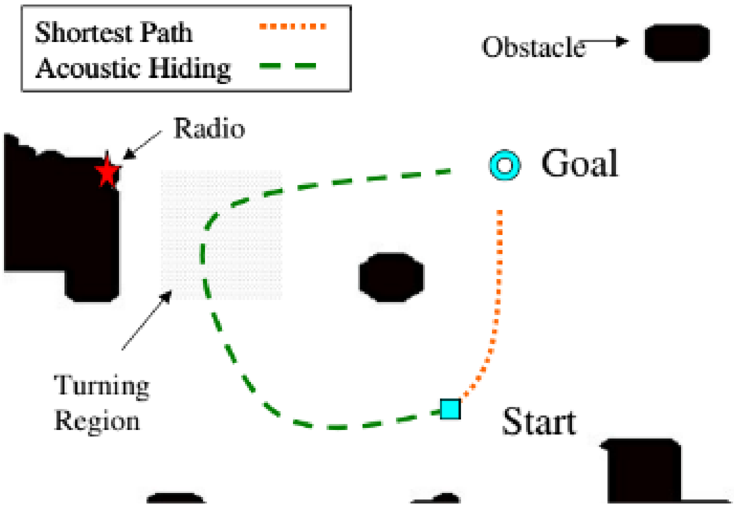


FIGURE 2.1: Martinson's work. First scenario. It is possible to compare the path the robot follows when it is using the proposed algorithm and when it is not. The green longer lines represent the path when using the algorithm. The red short lines represent the path directly to the objective.

source, it was detected about 57 % of the time. It is worth mentioning that most of the time that was detected, when following the longest path, was when the robot was aligning itself with the sound source before proceeding to the destination. A second experiment was conducted on this scenario, where the sound source was louder. This experiment had the purpose of showing that, when there is to much noise in the environment, there is no advantage in hiding its acoustic noise. The results showed that, when following the shorter path, in 99 % of the samples, the robot's noise surpassed the environment noise less than one decibel and when the robot followed the longest path, it is undetectable for about 96 % of the samples and only detected when aligning with the sound source.

The second test (see Figure 2.2) the robot was asked to move near walls. By being close to the wall, which partially reflects the sound, the robot was detectable more easily.

As expected, the results were much worse in the test. When the robot was trying to hide its noise, the robot was not detected only in 17 % of the samples. When moving directly to the objective, it was not detected only in 9% of the

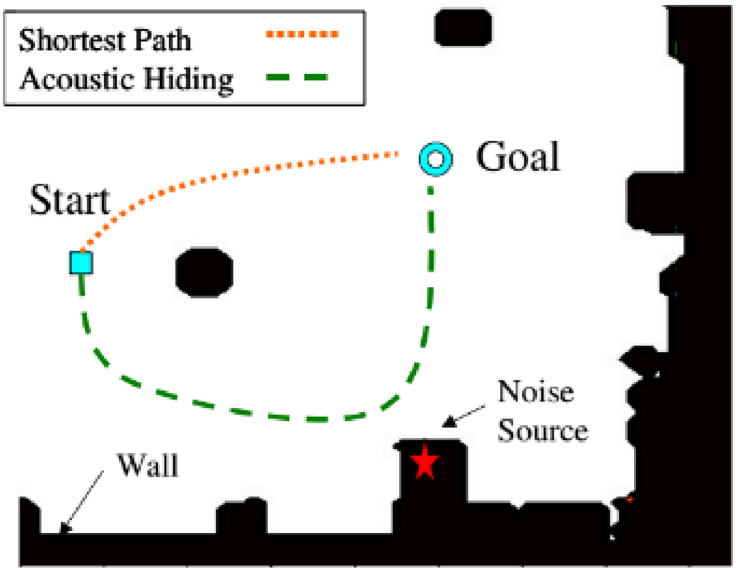


Figure 2.2: Martinson's work. Second scenario. It is possible to notice that the robot requires to move near walls, which difficult the ability to hide its acoustic noise. The green longer lines represent the path when using the proposed algorithm. The red short lines represent the path directly to the objective.

samples. The robot kept having a better performance when trying to hide its acoustic noise, but it was not as good as in the first test.

2.1.1.3 Detecting the environment's sound sources

To handle multiple sound sources, acoustic maps can be created and updated by the robot by actively moving in the environment. These maps reveal the location of the sound sources present on the environment, which can be used to hide the robot's acoustic signature. In the work of Kallakuri et al (Kallakuri et al., 2013a), the robot navigates autonomously through a division while gathers sound samples. When a sound source is present on the environment, the robot updates its acoustic map with the position of the sound source, which gets more accurate as more samples are gathered. In the work of Martinson and Schultz (Martinson & Schultz, 2006) the robot is teleoperated in a division. Two experiments were performed. One where the robot makes regular stops to gather sound samples and another where the robot gather sound samples without stopping. The goal of this experiments was to verify whether the robot is able to detect the sound sources regardless of its own acoustic noise. Results showed that, while the robot's acoustic noise introduces some error, the algorithm works when there are only one or two sound sources, regardless of the robot's acoustic noise. If there are more than two points of origin, the algorithm is not capable of detecting correctly the sound sources, even if the robot is idle collecting sound samples. In the work of Martinson and Schultz (Martinson & Schultz, 2007), the robot first navigates through the environment to detect possible sound sources. The robot is equipped with more than one microphone, so it can compare the arrival time of a sound on the microphones and estimate the angle where the sound is coming from. Since it does this while it is moving on the environment, it can, after some time, start estimating the position of each sound source in a division. The final step on this experiment takes the robot to each detected source to confirm that the source exists.

2.1.2 Self-supervised learning

It can be rather difficult designing manually a set of rules that allows the robot to control its acoustic signature as to not disturb people by its presence. The difficulty raises from, for example, the wide variety of robot mechanical structures and environments. An alternative to this process is to allow the robot to learn its acoustic noise in a self supervised way as a function of the environment and motion speed.

Recently, Self-supervised learning, specially in safe navigation domain, has been highlighted, which requires the robot to autonomously learn classifiers for terrain assessment from images and point clouds (Bajracharya, Howard, Matthies, Tang, & Turmon, 2009) (Heidarsson & Sukhatme, 2011) (Wurm, Kretzschmar, Kümmerle, Stachniss, & Burgard, 2012) (Pinto et al., 2014) (Baleia, Santana, & Barata, 2015) (Pombeiro et al., 2015) (van Hecke, de Croon, van der Maaten, Hennes, & Izzo, 2018). For instance, by manipulating an object, the robot is able to obtain ground-truth regarding how traversal that object is (Baleia et al., 2015). Then, the learned associative mapping can be used to predict future robot-terrain interactions, given sensory feedback. In this dissertation a similar problem is addressed: to learn the acoustic noise induced by the robot in a given environment by executing pre-defined motor actions to generate sufficient ground-truth labels for the learning process to take place.

Related to the concepts of active perception (Bajcsy, 1988) (Ballard, 1991), the self-supervised learning concept follows the affordance principle studied by Gibson for the animal kingdom (Gibson, 1977). The concept of affordances links the ability of a subject, through its actions, to the features of the environment and, so, to learn an affordance the agent needs to interact with the environment. Studies about this idea has been done in humans (Lacey, Hall, & Sathian, 2010) (Schwenkler, 2013) and robotics (Jamone et al., 2016) for safe navigation purposes (Uğur & Şahin, 2010). In this dissertation, the problem of learning what acoustic noise level is afforded by the environment, taking into account its and the robot's characteristics, is addressed.

2.2 Working concepts

2.2.1 ROS - Robot operating system

ROS is a collection of libraries and frameworks, based on Linux, designed specifically for robotic engineering. Its main advantage is the possibility of having different teams developing different isolated software pieces that enables a robot to perform a certain task, like creating a robust map of an environment or process some sensors information, and easily integrate these components into a robust system without wasting too much time and resources to use these systems together.

At the time of the writing of this dissertation, ROS already have multiple versions released, with "Melodic Morenia" being the latest version released. For the proposed system, the version used was "Kinetic Kame", as it was the latest version released at the start of development.

ROS is a system that allows the creation of nodes, that can implement algorithms, that communicate between each other through topics or services. The concepts of node, topic and service are explained next.

2.2.1.1 Nodes

A node is a process that executes a certain task. Normally a robot has several nodes running simultaneously, performing different tasks. For example, a node can be used to gather data from a range sensor, another controls the wheels motors, another has the objective of knowing the robot's position and another node, somehow, gathers and process those data and gives instructions to other nodes.

The ability to use nodes is a very useful feature as enables to have the different tasks and processes of the robot running independently of each other and avoid that the whole system collapses if there is some problem with one of the nodes.

2.2.1.2 Topic

A topic is a way of communication between nodes. It is based on publications and subscriptions of messages and is a way of sharing information. When a node wants to share some data on a certain topic, it publishes a message (with that data) on the topic. When a node wants to know what is published on a topic, it subscribes to the topic. This way of communication can be useful when there are several nodes that are interested in the information of one or more nodes that contains similar content. For example, in the robot used for experimental validation in this dissertation, there is a node that is responsible for gathering information from a microphone and there are one or more nodes that require that information. Instead of the node having a list of the nodes that require its information and sending to each one of them, it can just publish that information on a topic, and any other node that needs that information, can just subscribe to that topic. This enables the possibility of incrementing or decreasing the number of nodes that require a certain information from another node without changing the code or structure of other nodes.

There can be any number of nodes that publish and subscribe to a topic. This is a unidirectional way of communication where it is not defined who subscribes and publishes to a topic. Figure 2.3 exemplifies how this type of communication is organized.

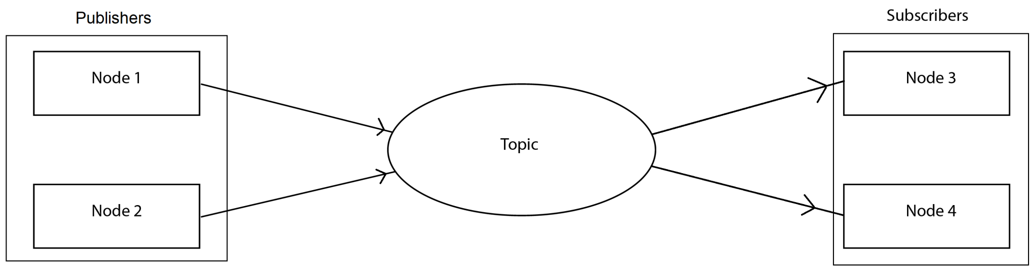


FIGURE 2.3: Figure showing how nodes communicate over topics. One or more nodes can publish data to the topic (left side). One or more nodes can subscribe to the topic to receive the published data (right side).

2.2.1.3 Service

A service is another way of communication between nodes. This is a method of communication only between two nodes. A node provides a service, that can be the data from a sensor, and any other node that is interested on that information sends a request to the given node providing the service and waits for its response. Only one node can provide a service, but several nodes can make requests to that service and it is possible to have different services running in different nodes. Figure 2.4 represents the communication over a service.

2.2.2 Hardware

In addition to the robot itself, there are three key hardware components necessary for the correct work flow of the proposed system: a laptop, where the system is run, a microphone to capture the noise from the robot and the environment, and

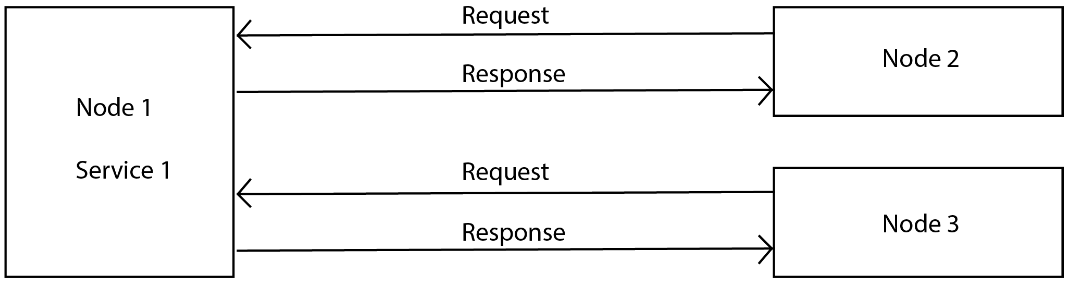


Figure 2.4: Figure showing how node communication works over services. A node provides a service, and other nodes make requests to the node responsible for the service and wait for a response.

an Arduino or RPI to receive the microphone data and send them to the laptop. Any laptop that is able to run ROS should be able to be used for the proposed system in this dissertation.

To receive data from the microphone and send them to the laptop, an additional component is required because the laptop cannot receive the data directly from the microphone. There are many solutions for this problem, but to keep the budget low, and yet, accessible to a large community, the decision lied between an Arduino Uno and a Raspberry pi 2 Model B+. Both these hardware pieces are not expensive, have a large community developing projects using these two components and are beginner friendly, meaning that anyone that is interested in electronics can start developing projects easily with these two boards.

2.2.2.1 Turtlebot

Turtlebot [1] is a robot kit, with an open source software, that was designed to perform tasks in indoor situations, including mapping a room to plan the robot's path when performing a task or following someone. On its shelfs, users can place other hardware components to enhance the capabilities of the system they are developing. It is designed to be easy to build and use to appeal people new to robotics. Because its core technology is SLAM (simultaneous localization and

[1] `https://www.turtlebot.com/`

mapping), the robot can be used to develop applications where the robot needs to navigate through one or more rooms in a building.

There are, so far, three versions of the Turtlebot. Turtlebot 1, 2 and 3. The Turtlebot 1 kit was the first robot kit released, and consisted of a base (iRobot create), a battery pack, a powerboard with a gyroscope, a kinect sensor a laptop and an hardware mounting kit. Turtlebot 2 has some extra features Turtlebot 1 does not have, such as a fast charger and a charging dock, and the base is changed (Yujin kobuki). Turtlebot 3 is designed to let users customize its shape. It is made up of modular plates, have three different types available (small size Burger, medium size Waffle and Waffle Pi) and has more features than the predecessor, like a LIDAR (Light detection and ranging), that enables the robot to measure distances to a target through laser light, a raspberry pi 3 to replace the laptop and a camera, while the kinect sensor was removed. Figure 2.5 depicts the turtlebot 2, the robot used for the experimental validations in this dissertation.

FIGURE 2.5: Turtlebot 2. The robot used on this project.

2.2.2.2 Microphone

The microphone is an important component needed for the development of the proposed system, as the system needs to read and process data from the environment in real time to determine the speed at which the robot should move. Although any microphone that can be connected to a RPI or arduino should be able to be used with the proposed system in this dissertation, the one used is a Sparkfun ADMP401. This microphone is not expensive, to keep the budget as low as possible. Figure 2.6 shows the microphone used for experimental validation.

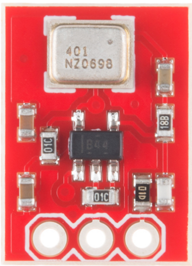


FIGURE 2.6: Figure showing the microphone used on this project.

2.2.2.3 Arduino

Arduino is a single board microcontroller used by many people, including students, hobbyists or companies to develop a variety of projects, from projects for beginners, like turning on a LED or using some sensor, to more complex projects. There are different Arduino versions, some more towards beginner and others for more advanced users. The major differences between them are the performance and features. The beginner friendly are Arduino 101, Arduino Uno, Arduino Leonardo and Arduino Esplora. Table 2.1 show a comparison between these boards. Figure 2.7 shows a Arduino Uno, which is the board used during the development of this project.

Arduino Uno is the recommended version from the creators to everyone who wants to start doing experiments related to electronics. Arduino 101 is a board that is similar to Arduino Uno, but with low-power consuption in mind. Besides

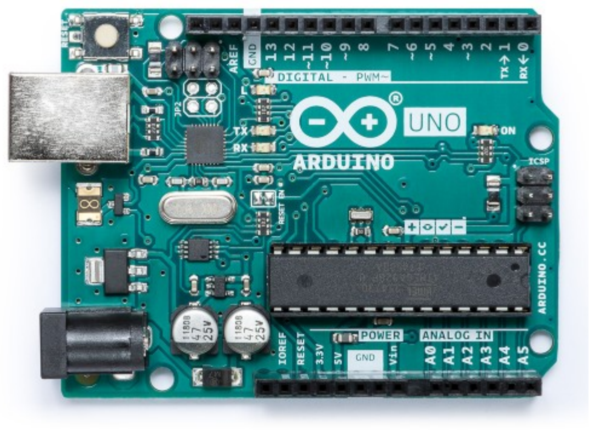


FIGURE 2.7: Arduino Uno. The same board as used during the development of this project.

TABLE 2.1: Comparison between the different Arduino boards designed for beginners.

Specification	Arduino 101	Arduino Uno	Arduino Leonardo	Arduino Esplora
CPU speed	32 MHz	16 MHz	16 MHz	16 MHz
Output voltage	3.3 V	5 V	5 V	5 V
Analog inputs	6	6	12	0
Digital IO	14	14	20	0
PWM	4	6	7	0
EEPROM	0 kb	1 kb	1 kb	1 kb
SRAM	24 kb	2 kb	2.5 kb	2.5 kb
Flash	196 kb	32 kb	32 kb	32 kb

that, Arduino 101 has better CPU speed, more SRAM and more flash memory than the other boards. Arduino Leonardo is another entry-level board similar to Arduino Uno. The main characteristic of this board is that it has a built-in USB communication, making it unnecessary to have a secondary processor. Arduino Esplora is a board based of Arduino Leonardo, and distinguishes itself by having multiple built-in sensors, making it accessible to people who want to work with Arduino but do not want to learn the electronic part yet.

Table 2.2: Comparison between the different version of the Raspberry pi. It is only compared the latest release of each version.

Version	RPI A+	RPI B+	RPI 2 B	RPI 3 B+	RPI Zero W
CPU	Single-core 700 MHz	Single-core 700 MHz	Quad-core 900 MHz	Quad-core 1.4 GHz	Single-core 1 GHz
Memory	512 MB	512 MB	1 GB	1 GB	512 MB
Network	None	Cable	Cable	Cable / Wireless	Wireless
Number of GPIOs	40	40	40	40	40
Output voltage	3.3/5 V	3.3/5 V	3.3/5 V	3.3/5 V	3.3/5 V

2.2.2.4 Raspberry Pi

RPI is a small not expensive computer used by a large community to create projects, including enabling the control of lights in a division or a house from a web application and controlling small cars or robots. RPI was created with the ambition and objective to allow kids, and anyone who wants, to learn basic coding skills. This computer is not expensive, which enables people in developing countries to acquire them to teach other persons who, generally, cannot afford computers that are more costly. Although this computer is inexpensive and small, it has more potential than just teach coding skills. Many people outside their target audience, including hobbyists, researchers and even companies, acquired them to develop their own projects. This allowed the RPI to became popular for a diversity of projects that range from home automation, to AI (artificial intelligence) research, to computer network applications or online servers.

Throughout the years, since the original RPI, there have been a few other versions released. There are three main models, until the time of the writing of this dissertation, Model A, Model B and Pi zero. The Model A was the first model released. Model B is the current main computer model with multiple versions released so far. Pi zero is an even cheaper computer that is mostly used for smaller projects that do not require the specifications available on the others raspberry pi models, as it is smaller and less powerful than model A and B.

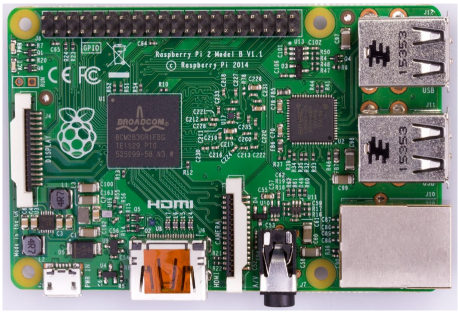


FIGURE 2.8: Raspberry pi 2 Model B. The same board as used during the development of this project.

Table 2.2 shows a comparison between the different models. To simplify the explanation of the different models, only the latest release of each version for each model will be compared. As it is possible to verify, the Raspberry Pi 3 Model B+ is the flagship model and with the better characteristics. RPI Zero W, while the cheaper version available is it already better at some specifications than the precursor versions. Figure 2.8 shows a Raspberry pi 2 Model B, which is the board used during the development of this project.

Chapter 3

Proposed Method

The acoustic noise a robot induces in the environment can be uncomfortable or annoying to people present in the same environment. Therefore, the amount of acoustic noise generated by the robot must be limited so it becomes unobtrusive. To do that, the robot used in this dissertation is equipped with a microphone to listen to itself while moving or executing any task it is required, allowing it to learn how much acoustic noise it produces on different environments. This poses the challenge of choosing the best position to put the microphone on the robot, so it best captures its own induced acoustic noise. Another challenge is to determine the amount of microphones required for a proper sampling of the induced acoustic noise. The robot must be able to understand the acoustic characteristics of the environment, for example, by quantifying the amount of acoustic noise it makes on the environment while moving with different speeds and the amount of background noise present. By being able to do that, the robot acquires sufficient knowledge to disturb nearby humans as little as possible. Nevertheless, in quiet environments, like a library, it may not be possible to avoid a certain degree of disturbance even if the robot moves at slow speeds.

3.1 Sensing

In order for the system to work properly, a set of rules are necessary to be established. A robot can, potentially, be placed in any environment under any conditions. Since every situation can be unique, because the sound can be propagated differently, the acoustic noise induced by the robot may be different for each context. For example, an indoor situation is different from an outdoor situation. For indoor situations, different floor types can cause a ground robot to produce different acoustic noises when moving at similar speeds. The size of the division the robot is placed in and different wall materials contribute for variations in the sound propagation. For outdoor situations, factors like wind or the place the robot is in, for example in a city, a park, a forest or a desert also impact the way the sound is propagated. Therefore, the robot must take into consideration the context in which is in to select the correct speed. The set of possible contexts the robot may operate in is defined as:

$$C = \{c_1, c_2, \ldots\}. \tag{3.1}$$

One way of directly controlling the amount of noise the robot is producing is by controlling the speed the robot executes its tasks. Normally, if the robot is moving faster, the motors of the wheels, or propellers if it is an air drone for example, rotate at a faster rate, which induce more acoustic noise. Depending on the robot's characteristics, the robot may produce different noise levels, which makes it necessary for the robot to learn the impact of each speed in each context. However, it is impractical and may not even be possible to test all speeds in all environments. So it is necessary to define a set of speeds that the robot is expected to use for each context. That way, it is only learned the amount of noise the robot produces for speeds that it will use and not a range of randomly selected speeds. Let us define the set of speeds that have been tried by the robot in a given context $c \in C$:

$$S^{[c]} = \{s_1^{[c]}, s_2^{[c]}, \ldots\}. \tag{3.2}$$

The noise the robot produces at each speed needs to be measured somehow. Each speed in each environment is tested multiple times (up to $n^{[c][s]}$), to make the system robust, by performing a set of fixed action patterns, where these patterns must be related to the robot's objective in the contexts. These actions produce noise, whose magnitude, x, is measured with the robot's microphone, resulting in a time-series associated to the context $c \in C$ and speed $s \in S^{[c]}$ in question:

$$X^{[c][s]} = \left\{x^{[c][s]}[0], x^{[c][s]}[1], \ldots\, x^{[c][s]}[n^{[c][s]}]\right\}. \tag{3.3}$$

3.2 Learning

The measurement of noise magnitudes are important, because that data can be used to know how much noise the robot usually produces while performing an action on a certain context. Learning occurs by storing in an associative memory the average noise level, $\mu^{[c][s]}$, and a conservative measure of the noise level variation, $\sigma_+^{[c][s]}$, observed while performing each assessed speed $s \in S^{[c]}$ in context $c \in C$:

$$M = \left\{\left(\mu^{[c][s]}, \sigma_+^{[c][s]}\right), \forall c \in C, \forall s \in S^{[c]}\right\}, \tag{3.4}$$

where the average noise level for speed $s \in S^{[c]}$ in context $c \in C$ is given by

$$\mu^{[c][s]} = \frac{1}{n^{[c][s]}} \sum_{i=0}^{n^{[c][s]}} x^{[c][s]}[i]. \tag{3.5}$$

To account for the uncertainty that can emerge from the sample size, instead of using only the standard deviation of the samples, the conservative noise level variation measure, which is the sum of the standard deviation and the standard error of the mean, is used:

$$\sigma_{+}^{[c][s]} = \sigma^{[c][s]} + \sigma_{-}^{[c][s]}, \tag{3.6}$$

where the standard deviation of the noise level for speed $s \in S^{[c]}$ in context $c \in C$ is:

$$\sigma^{[c][s]} = \sqrt{\frac{\sum_{i=0}^{n^{[c][s]}} \left(x^{[c][s]}[i] - \mu^{[c][s]}\right)^2}{n^{[c][s]} - 1}}. \tag{3.7}$$

The standard error of the mean of the noise level for speed $s \in S^{[c]}$ in context $c \in C$ is given by:

$$\sigma_{-}^{[c][s]} = \frac{\sigma^{[c][s]}}{\sqrt{n^{[c][s]}}}. \tag{3.8}$$

The algorithm can then access this memory to predict the amount of noise a robot should produce, depending on the speed and context.

3.3 Memory recall

The associative memory is useful to for the robot to know how much acoustic noise it should induce when moving at different speeds for different contexts. By using this information, the robot can adapt its speed so as to avoid producing noise whose magnitude is higher than the one of the environment's.

Typically, the robot is required to perform a certain action which needs the robot to move at a given desired speed s_r. The robot, first, needs to estimate if it will cause more acoustic noise, at speed s_r, than the noise present in the environment and, if it does, calculate the maximum speed it can move in order to not produce more noise than the environment. For that, the robot needs to access its memory.

Let us assume that the robot needs to calculate the expected conservative noise level variation when traveling at a desired speed, s_r, in a context c. If the robot has already moved at that speed, and stored the relation between the speed and the noise, then it can access directly to its memory. Otherwise, if the robot is unaware of the conservative noise level for that speed, then it can linearly interpolate from the two closest speeds stored in memory. The expected conservative noise level variation is:

$$\sigma_r(c, s_r) = \begin{cases} \sigma_+^{[c][s_r]} & \text{if } s_r \in S^{[c]} \\ \psi\left(s_r, s^-, \sigma^{[c][s^-]}, s^+, \sigma^{[c][s^+]}\right) & \text{if } s_r \notin S^{[c]} \end{cases} \tag{3.9}$$

where:

$$\psi(x, x_0, y_0, x_1, y_1) = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}, \tag{3.10}$$

where $s^{+[c]}$ and $s^{-[c]}$ are the closest higher and lower speeds stored in memory $M^{[c]}$ to s_r for the context $c \in C$. Figure 3.1 shows a representation of these values, which are given by:

$$s^{+[c]} = \arg \min_{s \in S^{[c]}, s > s_r} (s - s_r), \tag{3.11}$$

$$s^{-[c]} = \arg \min_{s \in S^{[c]}, s_r > s} (s_r - s). \tag{3.12}$$

In memory, the robot can have access to the expected average noise level for different speeds. Instead of consulting directly the memory, like for $\sigma_r(c, s_r)$, the robot uses a model learned from the values stored in memory. After some tests (see Section 5.1), it was verified that for low speeds, a second degree polynomial fits well the data, while for high speeds a linear model is enough. That way, the expected average noise level for speed s_r (see Figure 3.2) is:

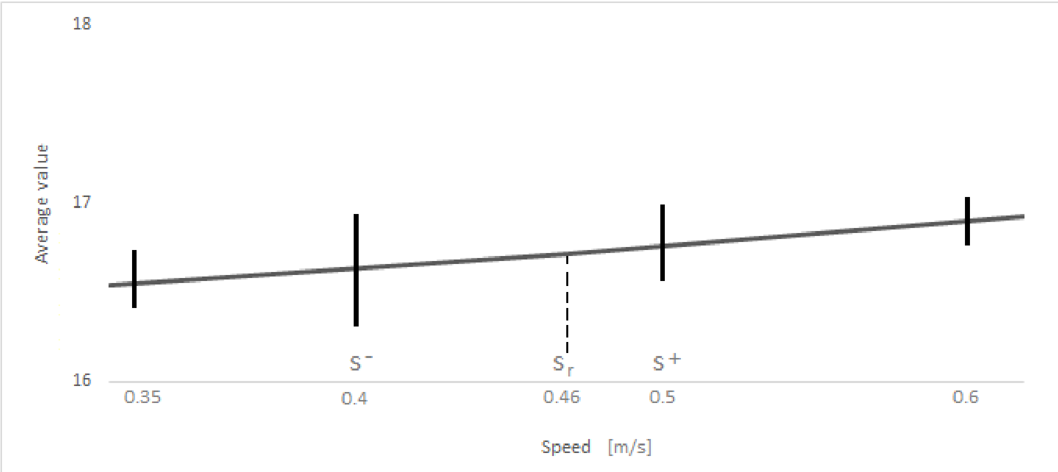


FIGURE 3.1: Visual representation of s^+ and s^- for a given s_r. Both s^+ and s^- are the closest speeds stored in memory M to s_r.

$$\mu_r(c, s_r) = \begin{cases} ax^2 + bx + c & \text{if } s_r <= s_d \\ dx + e & \text{if } s_r > s_d \end{cases} \tag{3.13}$$

where s_d is the threshold where the model (see Section 5.1) is best represented by a second degree polynomial or a linear model. $a, \ldots, e$ are parameters learned with regression based on a set of data points corresponding to tuples speed-noise, which can be gathered with a microphone (see Equations 3.2 and 3.5):

$$D^{[c]} = \left\{ \left(s, \mu^{[c][s]}\right), \forall s \in S^{[c]} \right\}. \tag{3.14}$$

3.4 Motion control system

In order for the robot to use the proposed system, it needs to have a controller that implements the process defined in this chapter. This controller can be responsible for, when prompted to do so, take control of the robot's movement until it finds the maximum speed the robot can move on that moment for that environment so it does not disturb any humans nearby with its acoustic noise.

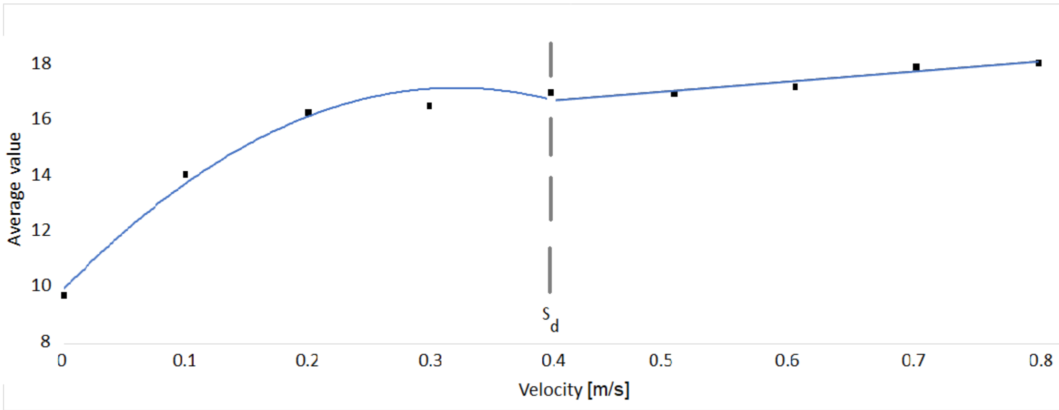


FIGURE 3.2: Visual representation of the selection process for μ_r given a threshold value s_d. If the desired speed, s_r, is lower than s_d, then μ_r is selected through a second degree polinomial equation. If s_r is higher than s_d, then μ_r is selected through a linear equation. The dots represents the real μ_r for each speed. The lines corresponds to the equations that represents the μ_r values.

With the knowledge acquired from the learning stage, the robot can then select, at each moment, the maximum velocity possible (up-to a reference desired speed) that induces less acoustic noise than the background's environment acoustic noise. It does this by stopping the robot, listening to the environments acoustic noise and then predicting, based on previous acquired data on that environment about its noise, the highest speed it can perform the task without producing too much acoustic noise. Figure 3.3 illustrates the basic principles of operations of the proposed system.

Algorithm 1 describes the process to select which speed it is sent to actuators, while taking into account the induced noise, its memory and desired speed. The controller operates according to three input data and outputs the speed value at which is expected the robot to not produce more acoustic noise than the environment. Regarding the input data: desired speed s_r is the speed the robot should move if the noise level was not a concern, and is often task-oriented; Environment context c is the context where the robot is placed, and is assumed it is known, meaning, the robot's memory has entries related that environment; Speed search step α is the amount of speed that gets decreased every cycle of the controller until it finds the output speed s. Ideally, speed search step should be set to the minimum speed possible the robot can perform the task at hand, because if the

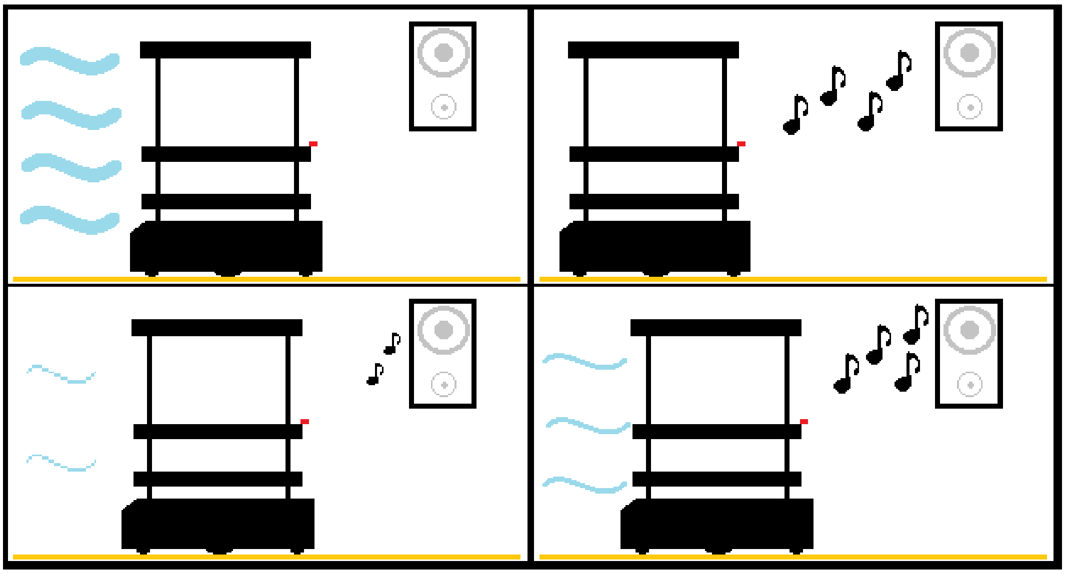


FIGURE 3.3: A cartoon representation of the proposed method's major steps. The red dot at the robot's middle compartment represents the microphone. The blue curved lines represent the robot's speed. The more and ticker the lines, the faster the robot is moving. The musical notes represent the environment's background noise. The more and bigger notes, the higher the volume. At the top-left image, the robot is not using the purposed method, and so, the robot is moving at any speed without consideration of the noise around it. When using the system, the robot needs to be idle and listen to the background noise, as depicted at the top-right image. Then, the lower the background noise (bottom-left image), the slower the robot moves. If there is a high background noise (bottom-right image), the robot is allowed to move faster.

robot cannot perform the task without producing too much acoustic noise at a reasonable speed, it should move at the lowest speed possible. The output s is the maximum speed the robot can move at that does not produce more acoustic noise than the environment, according to the input settings. If the desired speed s_r, leads the robot to produce less noise than the noise present in the environment, then s is set to s_r.

The first step (Step 1) of the algorithm orders the robot to stop moving. This is important, because that way, the robot's acoustic noise is not captured by the microphone and the algorithm does not need to have additional steps to separate the environment's noise from the robot's. After the robot is idle, the robot, through the microphone, gathers the environment's noise levels, E, for a certain amount of seconds δ (Step 2). Next (Step 3), the average background noise level, μ_e,

```
Data: desired speed, s_r (input)
Data: environment context, c (input)
Data: speed search step, α (input)
Result: Noise aware robot speed control
1  Set robot's speed to 0
2  Store environment noise level for δ seconds in E
3  Compute: μ_e = Σ_{e∈E} e/|E|
4  Initialise x: x ← μ_r(c, s_r) + σ_r(c, s_r)
5  Initialise selected speed: s ← s_r
6  while x > μ_e ∧ s > α do
7  |   Decrement selected speed: s ← s − α
8  |   Update x: x ← μ_r(c, s) + σ_r(c, s)
   end
10 Set robot's speed to s
```

Algorithm 1: Motion controller

is calculated with the the noise levels. Then, in a conservative way, the robot estimates the expected noise produced, x, at speed s (Step 4).

Initially, speed s is set to the desired speed s_r, because if the user selects a speed as the desired speed, it does not make sense to waste processing power and time on predicting if the robot is allowed to move at higher speeds, so the algorithm assumes the maximum speed possible is the desired speed s_r (Step 5). Next (Steps 6-8), a small cycle to find the best speed is executed. While the predicted robot's noise x is higher than the environment's background noise μ_e and the selected speed s is higher than the speed search step α, the selected speed is decremented and is predicted the robot's acoustic noise for that speed.

When the predicted robot's noise is lower than the environment's acoustic noise, the cycle breaks and the algorithm returns the selected speed s. If the selected speed s gets lower than the speed search step α, then it is assumed the robot cannot produce less noise than the environment's and it is returned the selected speed (Step 9). This can happen in situations where there is not much noise in the environment, like in a library.

Chapter 4

Implementation

4.1 Noise capture

To capture the robot's noise, a microphone needs to be installed on one of the robot's compartments. As a proof-of-concept, one microphone is enough. However, the system developed can be further expanded to consider several microphones, either on the robot or in the environment, for a better capture of both robot's noise and environment's noise. The microphone used is a Sparkfun ADMP401 (See Chapter 2.2.2). As the microphone cannot be connected directly to a laptop, additional hardware between the microphone and the laptop for the system to use the microphone's data is required. There are multiple hardware pieces capable of receiving input from the microphone and transport it to the laptop. Because this project aims for a low cost and be available to a large community right away, the decision lied between an Arduino Uno and an Raspberry pi 2 Model B+. To verify which hardware should be used, an experiment (See section 4.1.3) was conducted to compare both devices.

4.1.1 Connecting to a Arduino Uno

The connection between the Arduino and the microphone is simple because the Arduino can receive analog signals, the type of signal that the microphone outputs. A connection diagram between the Arduino and the microphone is exposed in Figure 4.1.

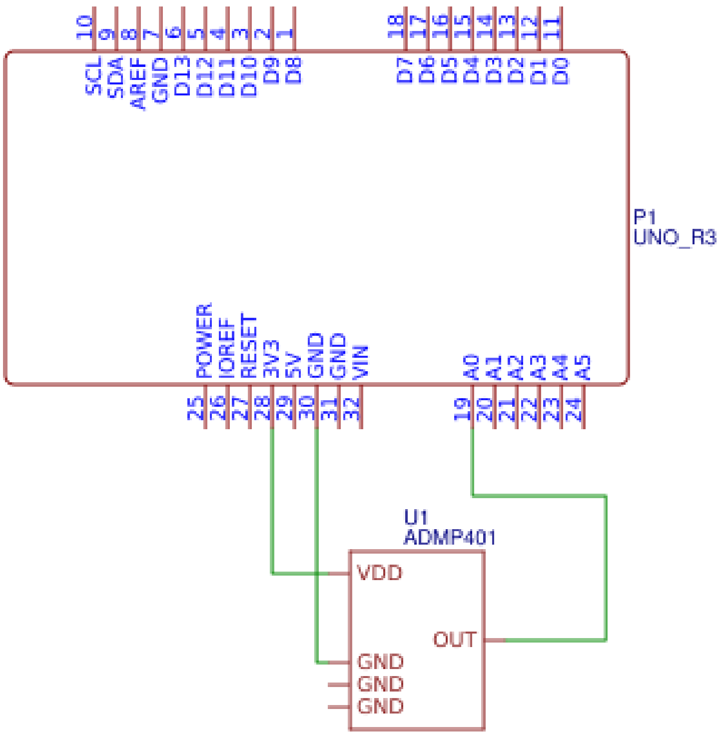


FIGURE 4.1: Diagram showing the connection between the Arduino Uno and the microphone.

4.1.2 Connecting to a raspberry pi - Model B+

The raspberry pi does not have an analog input to receive data directly from the microphone, so it was necessary to add an additional component to the circuit between the microphone and the raspberry pi that converts an analog signal to a digital signal. That component is an ADC (analog-digital converter) MCP3008. This is a 10-bit ADC, with 8 channels. The raspberry pi and the computer communicate through an ethernet cable. The complete circuit between the raspberry pi and the microphone is presented in Figure 4.2. The figure shows that there are a few extra components, like the resistors between the microphone and the raspberry pi. These resistors have the purpose of decreasing the voltage that goes to the reference voltage (Vref) pin on the ADC. The purpose of this is explained on Section4.1.4.

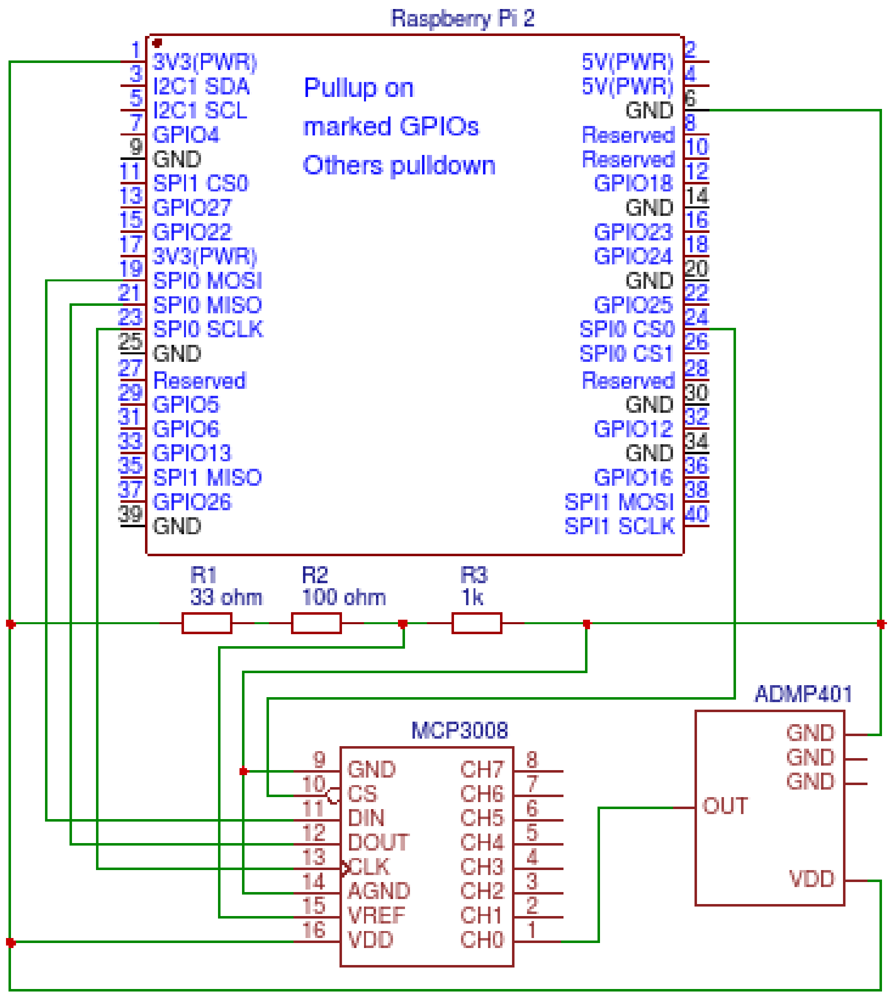


FIGURE 4.2: Diagram showing the connections between the microphone and the Raspberry pi 2. It is possible to see that the "Vref" (reference voltage) is different from "VDD" on the ADC in order to amplify the signal.

4.1.3 Arduino vs Raspberry pi

The Arduino is cheaper than the raspberry pi but in terms of computational performance, the raspberry pi is superior than the Arduino. The decision was made following a series of tests, instead of just evaluations based on both specification and price. The test consisted in connecting the microphone to the Arduino and the raspberry pi to send the data to the laptop while different sounds are being captured, which is enough to gather enough data to compare. The tests were done in a room when there are no background noises, outside of the ones the microphone is supposed to capture. All sounds were played under the same conditions for both devices, meaning the music starts at roughly the same place with the same volume and the human voice is roughly at the same volume and saying a constant sound, like a vowel. After having the tests completed, the results were compared. There were a total of 2 different sounds for the microphone to capture and one situation where there is no background noise, to understand the microphone levels when the ambient is silent. The sounds are: music track and human voice. These sounds are not selected randomly, as the robot can be placed in environments where there is music playing and/or humans talking. So, it makes sense to do every experiment as close to a real scenario as possible, and for this step, where it is tested the microphone with the Arduino and the raspberry pi, it is good to test in a quiet environment with some sounds it might capture when the system is implemented in a real situation. Another important detail is that both devices store as much data from the microphone as possible before sending to the laptop. This can be important, as delay can occur when sending data to the computer, and by gathering as much data as possible before sending them, the reality is better sampled with some missing samples occurring occasionally, instead of having missing samples between each sample sent.

Figure 4.3 shows the results of both the Arduino and the raspberry pi, respectively. To better visualize the data from the microphone when connected to the RPI, the charts only show the first 1000 samples gathered. For the data from the microphone when connected to the Arduino Uno, all samples are represented on the charts.

The first noticeable difference between both is the amount of data they can store. This is to be expected because of the amount of memory of each device, where the Arduino has available 2kb of SRAM, while the raspberry has 1GB of

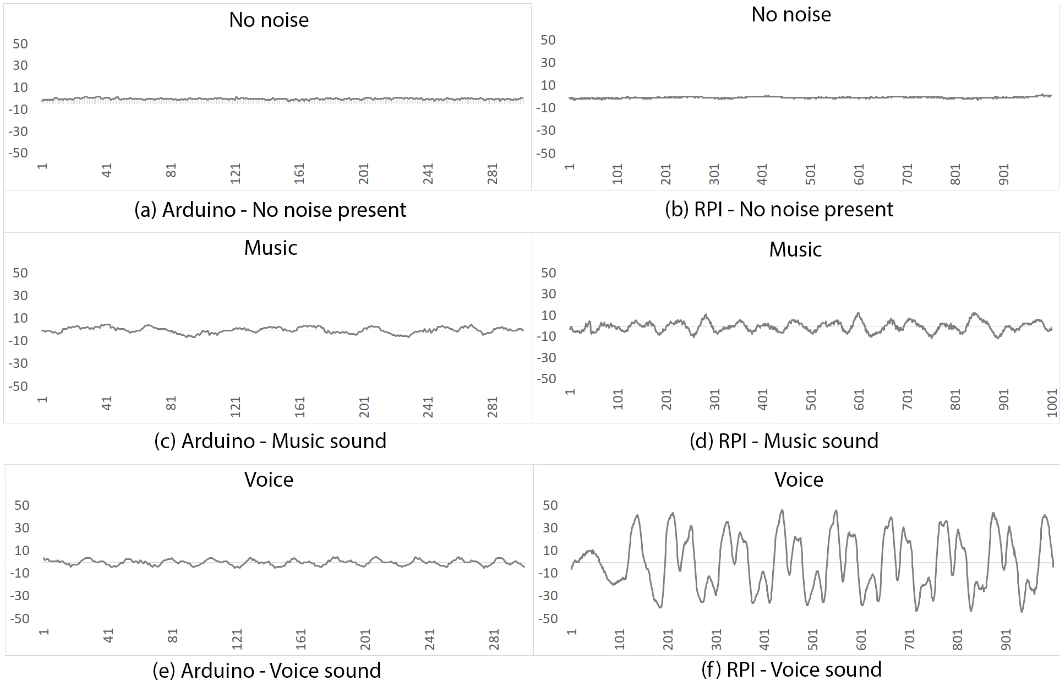


FIGURE 4.3: Tests made with the microphone connected to an Arduino Uno and a Raspbery Pi 2.

RAM. For the Arduino, it is possible to store about 300 samples every 33 ms, which can mean a possibility of 9000 samples per second, if the data is transferred instantaneously to the computer. With the raspberry pi, it was possible to store about 8000 samples in 110 ms, or about 2400 samples every 33 ms, which means it could send about 72000 samples per second, if the data is transferred instantaneously to the computer. This high difference in samples gathered between the devices is the reason why the sounds are better represented using a raspberry pi. Another difference is the range in intensity the sound is represented. With the Arduino, the sound varies in intensity between -10 and 10 (20 values), while with the raspberry py it varies between -50 and 50 (100 values).

Although the Arduino is cheaper than the raspberry pi, the latter was chosen, because the overall performance was better, the difference in number of samples is too much and the ability to gather a high number of samples is an important detail for this project.

4.1.4 Adjustments

With the hardware chosen to transmit data between the microphone and the computer, more improvements can be made to the overall setup so far, from the communication between the raspberry pi and the laptop to the circuit between the microphone and the raspberry pi.

The intensity levels of the noise captured from the vacuum cleaner do not vary much. With a difference between the highest and lowest intensity levels captured with the setup being about 100 values, it can render problematic in situations where the background noise is low and where the system should indicate the robot to move slowly, as it might not differentiate between that situation and an environment where there is no background noise. For that reason, it is necessary to understand how much variation it is possible to detect at each moment and a way to increase that variation.

To verify the maximum variation, it is enough to gather data when there are loud noises occurring on the environment. For that, a vacuum cleaner was turned on with the microphone close by. The values captured by the microphone might not represent the highest variation possible, but it should be a high enough noise to understand the current capabilities of the system. Figure 4.4 shows the resulting plot, where only 1000 samples are represented for the sake of clarity.

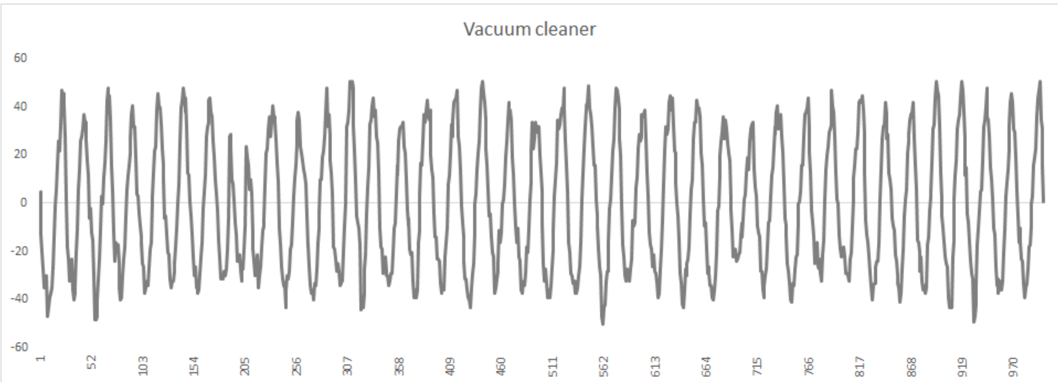


FIGURE 4.4: Noise values captured from a Vacuum cleaner with a microphone connected to a Raspberry pi 2.

A quick look at the plots indicate that the intensity levels did not vary as much as expected. In fact, it has roughly the same variation as the previous tests, even though the sound was much louder. The plot shows that the system is sensible to a variation of about 80 - 100 values. To increase this variation, it is enough to change the value of the VRef of the ADC. Originally the value of the VRef was 3.3 V, and by changing that value to 2.3 V, the signal is amplified. Figure 4.5 shows a plot representing the noise captured from a crowd sound, with the VRef set to 2.3v. Also, like the previous tests with the Raspberry pi, only 1000 samples are represented on the chart.

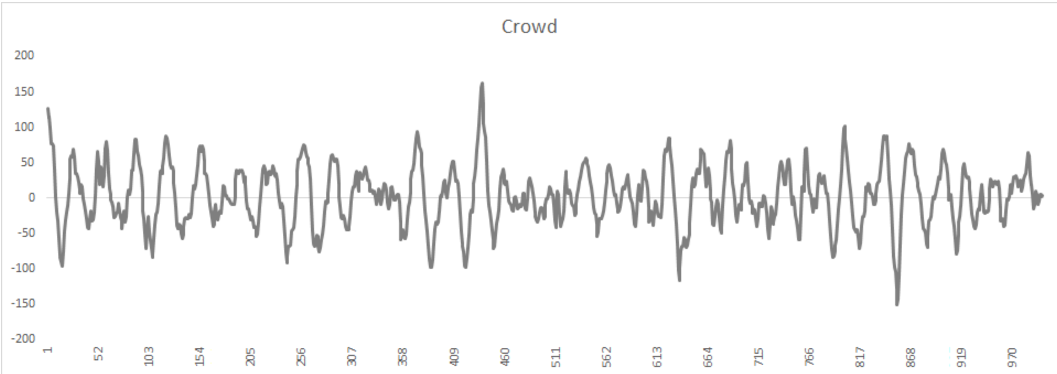


FIGURE 4.5: Noise values captured from a Crowd sound when Vref is changed to 2.3v instead of 3.3v.

Changing the value of VRef allowed for a higher variation on the noise captured by the microphone. This is more useful for low background and robot noises as these situations have low noise intensity variations, which amplified, allows for a higher difference between the background and the robot's noise. As it is possible to verify in Figure 4.5, the intensity vary about 150-200 values.

4.2 Microphone position on robot

Now that it is possible to capture data from the microphone to be used by the system, the next step is to place it on the robot. There are multiple possible positions for the microphone, as the robot is big enough to make the microphone's distance to the bottom of the robot, where the robot produces the most noise,

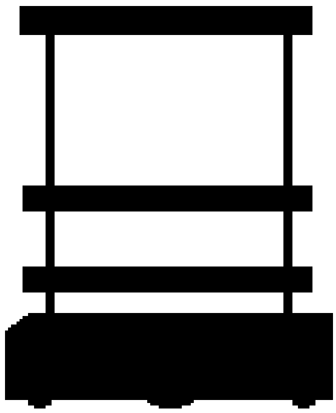


FIGURE 4.6: Robot with its top, middle, and bottom compartments.

impact the ability to capture that noise better or worse. The robot has three different compartments in which the microphone can be placed (top compartment, middle compartment and bottom compartment), Besides that, it is necessary to determine the position on the compartment and if it should be placed on top or below the compartment. To simplify the amount of options available, the decision lies between the front and back of the three compartments, making a total of six different possibilities, as depicted in figure 4.6. When the system is more robust, more positions can be studied to determine the best position as well as adding more microphones.

To decide the best position, a set of tests were performed. The microphone was placed on top of each compartment at each possible position facing downwards, since the robot's wheels and motors acoustic noise are at the bottom. The microphone captured the robot's acoustic noise, while the robot moved forward with a velocity of 0.2 m/s. To evaluate the different positions as thoroughly as possible, a wide range of speeds should be tested. Unless the robot is placed on a context where there are no people nearby or there are a lot of loud noises, the robot may move at speeds close to 0.2 m/s. Figure 4.8 shows a set of plots of the performed tests. The robot was placed on a living room with ceramic tiles. Note that for every test, the microphone gathers samples when the robot was stopped, so the first few samples represent the environment noise. Then, when the robot initiated its motion, it caused a significant amount of acoustic noise before stabilizing when it moved at the constant speed of 0.2 m/s. When the robot passed from a tile to another tile, it caused some additional noise because the floor is not flat between the tiles.

By analyzing the plots, several conclusions can be achieved. By comparing the front and back position of the microphone on the different compartments, the results are fairly similar, although when placed at the back of the robot, it captured higher intensity noise levels. Because the wheel's motors and the robot's base are placed at the bottom, when placed at the top compartment, the microphone exhibits the worst performance. It is still possible to distinguish the robot's acoustic noise from the environment's, but it is not as good as when placed at other compartments. Between the bottom and the middle compartment, both are good for the purpose of this project, although it is clear that when the microphone is placed at the bottom, it captures better the robot's acoustic noise. So, according to this plots alone, the microphone should be placed at the bottom compartment at the back of the robot, because it is where it performs better.

The microphone's performance is better when it is placed at the bottom compartment when the robot is moving at low speeds, but it is not possible to assume that it also has better when the robot is moving at higher speeds. A situation where this might occur is when the robot is placed on a small party or lounge where there are people talking and maybe some background music, allowing the robot to move at a higher speed. To check if the bottom compartment is still the best choice in those situations, another test was performed. On this second test, the robot moved at a speed of about 0.5 m/s on a garage with concrete floor. This speed is high enough to test the microphone's performance when the robot is moving fast, as 0.5 m/s is a relatively fast speed for a robot to move near humans, in an organization event in a closed space. Figure 4.7 presents two plots representing the robot's acoustic noise when moving at 0.5 m/s on a concrete floor. Like the previous test, the first samples correspond to the environment's noise. Then the robot starts moving, causing some higher mechanical noise. Finally, it stabilizes at the desired speed.

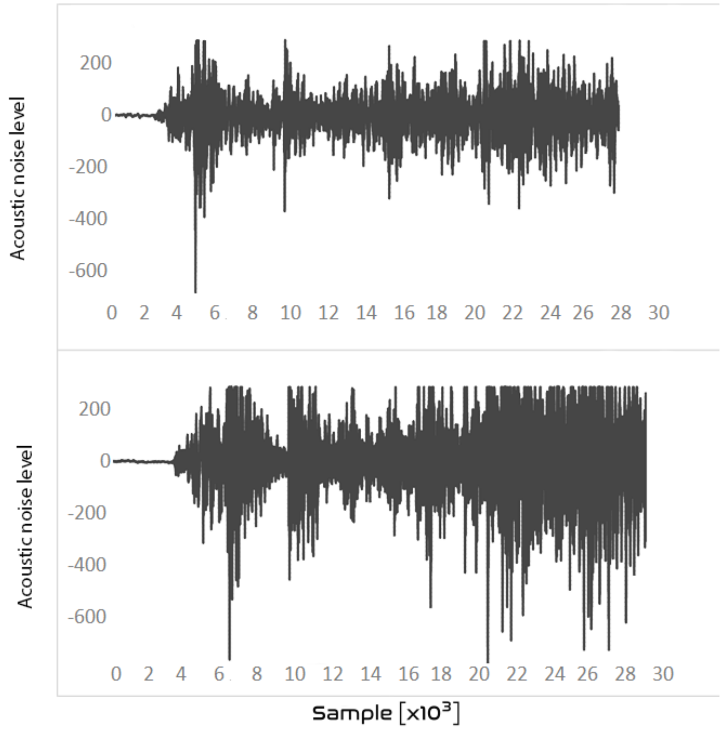


FIGURE 4.7: Acoustic noise level induced by the robot while moving at 0.5 m/s on top of rough concrete floor, with the microphone on different compartments located at the back of the robot. Like on fig. 4, the vertical axis represents the acoustic noise level, and the horizontal axis represents the different samples from the microphone. The top image plots the samples acquired with the microphone on the middle compartment and the bottom image plots the samples acquired with the microphone on the bottom compartment

Figure 4.7 shows the difference between the compartments as the intensity levels are much higher when the microphone is placed at the bottom compartment. It is also visible that the noise level gets more saturated when the microphone is at the bottom. Saturation occurs due to closeness to the motors and due to the high mechanical impact induced by the rough terrain on the robot's structure. Due to this saturation, the middle compartment was selected.

Every test of the motion controller and learning phase were done with the microphone placed at the middle compartment, at the back of the robot and facing down.

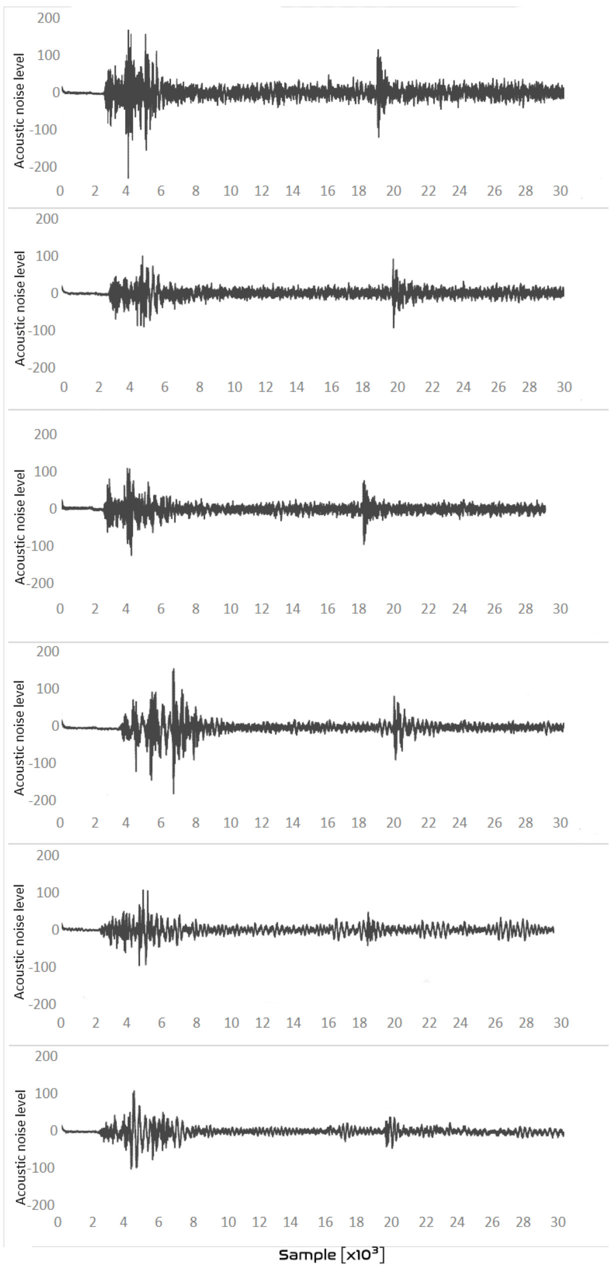


Figure 4.8: Acoustic noise level induced by the robot while moving at 0.2 m/s on top of ceramic tiles, with the microphone on different compartments and placed at the front and back of the robot. It is possible to check the environment's noise right at the beginning. The spike that follows represent the robot's motion onset. Then the noise stabilizes because the robot is constantly moving at the target speed. The spike that occurs at the middle represent the moment when the robot passes from one ceramic tile to another, which causes some mechanical impact that results in a higher acoustic noise. The order of plots, sequentially, from top to bottom are: 1 - bottom compartment, back position; 2 - bottom compartment, front position; 3 - middle compartment, back position; 4 - middle compartment, front position; 5 - higher compartment, back position; 6 - higher compartment, front position. The vertical axis represents the acoustic noise level, and the horizontal axis represents the different samples from the microphone.

4.3 Learning phase

With the microphone position determined, the precondition to develop the motion controller is fulfilled, i.e., capability to gather the robot's acoustic noise through a microphone for further use. The robot needs to have the associative memory M accessible at anytime, where it can consult the amount of noise it produces when moving at different speeds in a certain context. This is done through a learning phase. For every context the motion controller may be used, either M is available already, or the learning phase needs to be performed. This is an important step because the motion controller relies on that information to output which speed the robot may move to not disturb anyone nearby. It is also important that this phase is done when there is as less environment's acoustic noise as possible to avoid corrupting the data with noise not originated from the robot.

For the purpose of this dissertation, a set of five different contexts was selected. These contexts are all indoor situations, except for one. All of these contexts are controlled, meaning the amount of environment's acoustic noise present at each moment is predetermined and customizable, although the outdoor context has factors that can not be completely controlled, like the wind sound or some animal chirping. Because the robot moves on ground with wheels, the main acoustic characteristic of each context is the floor type. The outdoor context has the same floor type as one of the indoor contexts. This is useful to compare the performance of the system on different contexts that may have similar floor types. The set of tested contexts is:

$$C = \{tiles, wood, carpet, concrete, outdoor\_concrete\} \tag{4.1}$$

These contexts, depicted in Figure 4.9, are disparate enough to properly validate the motion controller's performance, as all of them are possibilities of where the robot may be placed in a real application. Both the tiles and carpet contexts are in the same space, a living room. The wood context occurs in another living room. The concrete context is in a garage and the outdoor concrete context is not

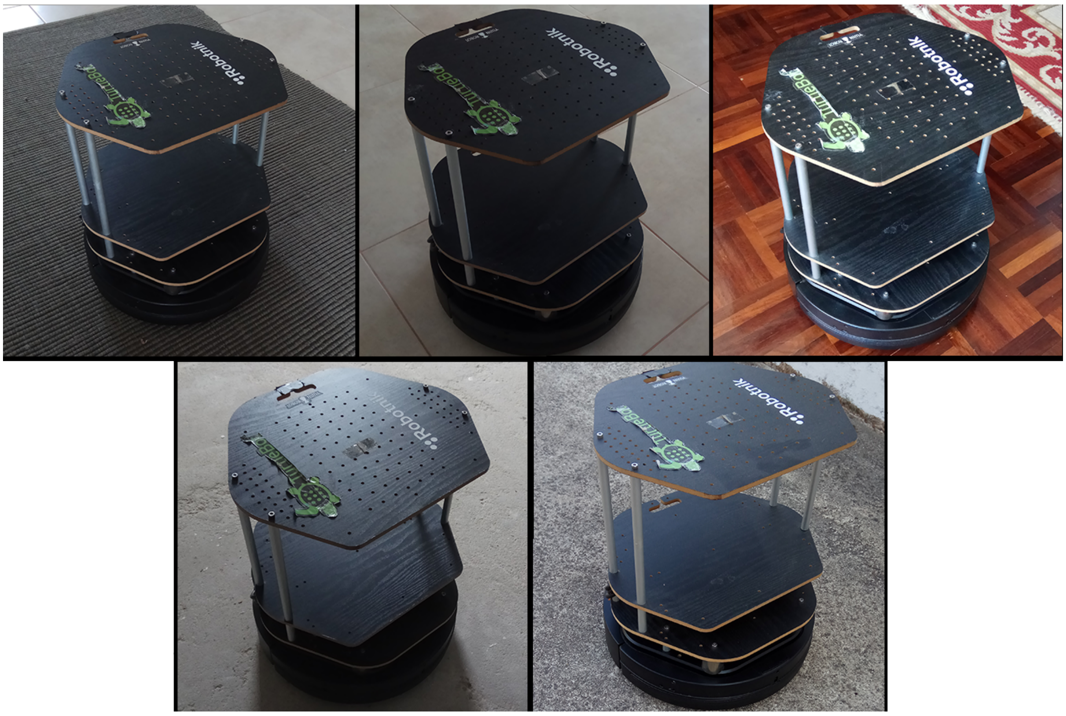


Figure 4.9: Different contexts tested. Top left: Carpet floor; Top middle: Ceramic tiles floor; Top right: Wood floor; Bottom left: concrete floor; Bottom right: Outdoor concrete floor.

in a confined space. The tiled context is a space with ceramic tiles on the floor and it represents an environment with a hard surface with small gaps every few centimeters, like a situation where the robot is placed in an environment where it is required to move between corridors. The wood context is a space with wood tiles on the floor. Contrary to the tiles context, it does not have gaps in between the tiles and is representative of environments like bedrooms. The carpet context is the robot moving on top of a carpet. This context simulates a softer ground, which is expected to make the robot produce less acoustic noise. The concrete contexts represent a rough floor with unpredictable irregularities, which should be the contexts that makes the robot produce the most acoustic noise.

For robots to coexist in the same context as humans, the robots cannot be uncomfortable to people nearby. Besides the acoustic noise the robot produces, the speed at which moves can also be uncomfortable when moving nearby humans. For example, it would be rather inconvenient if someone was walking on the street and a robot passes by them at a high speed, even if making no noise at all.

Another situation can be at a convention where it may be a lot of people gathered in the same space. If there were robots passing through them fast, they might get alarmed, not comfortable in that space and always aware if there was a robot nearby. For that reason, it is established, for this project, that the speeds range from 0.0 m/s to 0.8 m/s and in 0.1 m/s steps for every context. A speed of 0.8 m/s is already a relatively high for the robot to use, but since it is the learning phase, it is useful for the robot to have knowledge about speeds a little bit higher than the ones it is supposed to move at, even if it is just to have a bigger data-set to use. The set of tested speeds in each environment $c \in C$ is:

$$S^{[c]} = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\} \tag{4.2}$$

The learning phase consists of having the robot performing an action several times for a pre-determined time and speed. For the sake of this project, the robot moves only forward. The robot moves at each speed for each context for 2 seconds. To have a sufficient number of samples, the motion was repeated 15 times. If a more robust system is desired, the amount of repetitions can be higher. Every repetition for every context must have the same conditions to maintain consistency. The starting position is always the same, meaning the robot starts and ends each repetition at the same place. This is to make the each repetition as similar as possible. To make sure the only noise captured is the robot's, there can not be any addition noise that may pollute the during each repetition. The ideal situation would be to not have any kind of noise other than the robot's, but that is impossible for the contexts tested. Therefore, before each run, it is made sure there was as little external noise as possible. Every time some additional noise during each run that was not coming from the robot is noticed, the data gathered from that run is discarded and the a new repetition was performed until all 15 similar repetitions are completed.

Not all the data gathered from the microphone is necessary or useful to determine the robot's noise. For example, the first few samples are always from the environment's noise, because the robot is idle. Then, The robot starts its motion. Only after a some samples, it is acceptable to assume the data from the microphone corresponds to the robot's acoustic noise. For that reason, only 5000 samples (0.33 seconds) are used from each repetition. In a range of approximately 30000 samples (two seconds) for each run, the ones used to determine the robot's noise are from 15000 to 20000 (between 1 and 1.33 seconds). Because these data are nothing more than a discrete-time signal of the robot's noise, it is possible to calculate and use the signal energy of the data of each repetition. The signal energy, is calculated with the Equation 4.3, where $x(n)$ is is the discrete-time signal:

$$E_s = \sum_{n=15000}^{20000} |x(n)|^2 \tag{4.3}$$

Using Equation 4.3 may result in very high values to work with. A workaround to this is calculating the logarithm (ln) of the signal energy. This has the advantage of scaling the data to reasonable numbers, which makes it more amenable and easy to work with.

These data are essential to compose the set of noise levels $X^{[c][s]}$ (see Equation 3.3), which are used to fill the set of tuples M (see Equation 3.4) that contain the noise level average and the conservative noise level variation and understand the regression equations for each context $\mu_r(c, s_r)$ (see Equation 3.13).

4.3.1 Floor type's impact on the robot's acoustic noise

Learning and storing the amount of acoustic noise the robot produces is crucial for the proper operation of the system. A factor that can also be used to predict the amount of noise a ground robot produces in a certain context is understanding the amount of noise the robot produces when it is not touching the ground. This can be done by suspending the robot mid-air while trying to move. It can be important to understand and differentiate the impact of the robot's motors from the floor type in the robot's acoustic noise. Because it is gathered the context's acoustic noise during the learning phase (by listening to the acoustic noise when the robot is stopped), it is possible to determine, the relation between a floor type and a robot's noise. It is not taken into account other characteristics, like how open the space is or wall types, but a future improvement might be associating, comparing, understanding how different characteristics of a context affect the propagation of the robot's noise and how humans perceive that noise.

To understand the floor type's impact on the robot's acoustic noise, a similar experience to the learning phase was performed, except the robot was suspended mid air (wheels not touching anything). The robot performs the same 15 runs moving the motors that enables the robot to move at each speed for about 2 seconds. Like the learning phase, the environment's noise is as low as possible and each time a noise that is not from the robot is detected, the run is repeated.

With this information, the robot can, potentially, be placed in a new context, and based on the context's characteristics and previous experience on other contexts, predict the amount of noise it will produce without needing to have a learning phase.

4.4 Motion controller

The motion controller is the tool used by the robot, to determine at which speed it should move in order to not produce more acoustic noise than the already present in the environment. It uses the data from the learning phase and listens to the environment's acoustic noise at any given time to calculate the maximum speed the robot can move and not produce a higher acoustic noise.

Based on the algorithm depicted on chapter 3, a flowchart explaining how the implemented motion controller works is depicted in Figure 4.10. Before the user or the robot use the developed system, the robot must be idle to not pollute the data acquired by the microphone when listening to the environment's noise. This way, it is not needed an additional step to filter the robot's noise from the environment's. The first step needed is to define the initial variables: desired speed s_r, which is the desired speed for the robot to perform a task; the context c where the robot is going to be placed in; the speed search step α, which represents the amount of speed the algorithm reduces each cycle. By having a low α and setting a reasonable minimum speed for the robot, it is possible to verify possible flaws with the algorithm as the low α guarantees that the algorithm chooses a speed that makes the robot produce acoustic noise levels close to the environment's. A bigger α makes the robot to produce lower noise levels, as it would probably indicate lower speeds. The controller starts by listening to the environment's noise for about one second, which then uses that information to calculate the average noise level present in the environment (μ_e), which is considered the maximum noise the robot is allowed to produce. The current speed s is set to the desired speed s_r. Because that is the desired speed, it is not necessary to waste processing power and time verifying faster speeds. With data from the learning phase, the conservative noise level x for the robot for the speed s in context c is predicted, which is the sum of the average noise level μ_r and the conservative noise level variation σ_+. If the robot's conservative noise level x is higher than the environment's average noise and the selected speed is still higher than the minimum defined, the selected speed is decreased by the value of α and the cycle is repeated. If the robot's conservative noise level gets lower than the environment's average noise ($x > \mu_e$), then the cycle stops and the selected speed is returned and sent to the robot's actuators. If the selected speed gets lower than the minimum established, then the motion controller assumes the robot cannot perform its task without disturbing humans nearby.

4.4.1 Integration with another application

One of the important aspects of the motion controller developed in this dissertation is the possibility to integrate it into a bigger system. To test this, a simple application where the moves until a PIR sensor detects something was developed,

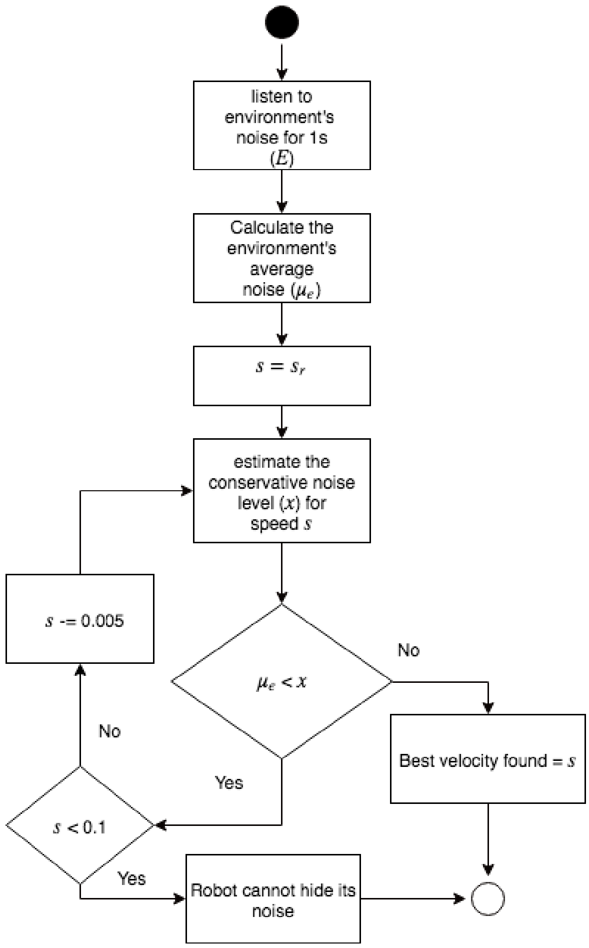


Figure 4.10: Motion controller flowchart. It is described the speed selection process, where s_r is the desired speed and s is the speed selected.

which makes the robot stop. This is to simulate an environment in which the robot performs its activities normally until a person is detected in the vicinity.

The PIR sensor is a Motion Sensor Module IM1206280099, which has a range of 7 m and a field of view of 110 degrees, and was installed at the front on the top compartment of the robot. The PIR sensor is connected to the same RPI as the microphone, as the RPI is powerful enough to handle both devices at the same time. Figure 4.11 shows the connection between the RPI and the PIR.

With the motion controller implemented into this application, the robot does not need to stop executing its tasks just because a person entered the room. It just needs to adjust it's speed to a value in which it does not produce more acoustic noise than the already present in the environment. To do this, when the PIR sensor detects someone, the robot determines the ideal speed using the controller's algorithm, avoiding to bother whoever is in the room.

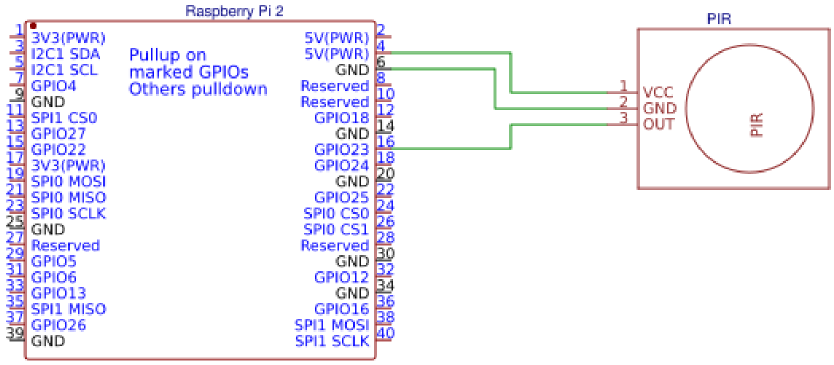


FIGURE 4.11: Connection between the Raspberry pi 2 and the PIR sensor.

Chapter 5

Experimental results

To validate the developed system, a set of tests were performed. The way found to do that, was to play a set of specific sounds in the environment and having the motion controller to predict which speed the robot should use to produce less noise than the environment's. This tests the algorithm's performance, but since this system is intended to be a complement to a bigger and robust system, it is necessary to test the motion controller in an application. To test the cooperation between two features, a second test was performed. In this second test, the robot, equipped with a PIR sensor, now performs the simple task of indefinitely moving forward until the PIR sensor detects something (a human appearing in front of the robot). Then, the robot stops, executes the motion controller and starts moving at the speed indicated.

All the tests for this dissertation were executed under controlled contexts, meaning that there were no unpredictable situations, like people passing by at random moments or sounds that are not a part of the tests. Also, they are done when there is very low environment's noise, except for the sounds played specially for specific tests. When a sound that was not coming from the robot or was not a sound played for testing purposes was detected, either that test or that part of the test was repeated.

5.1 Learning phase

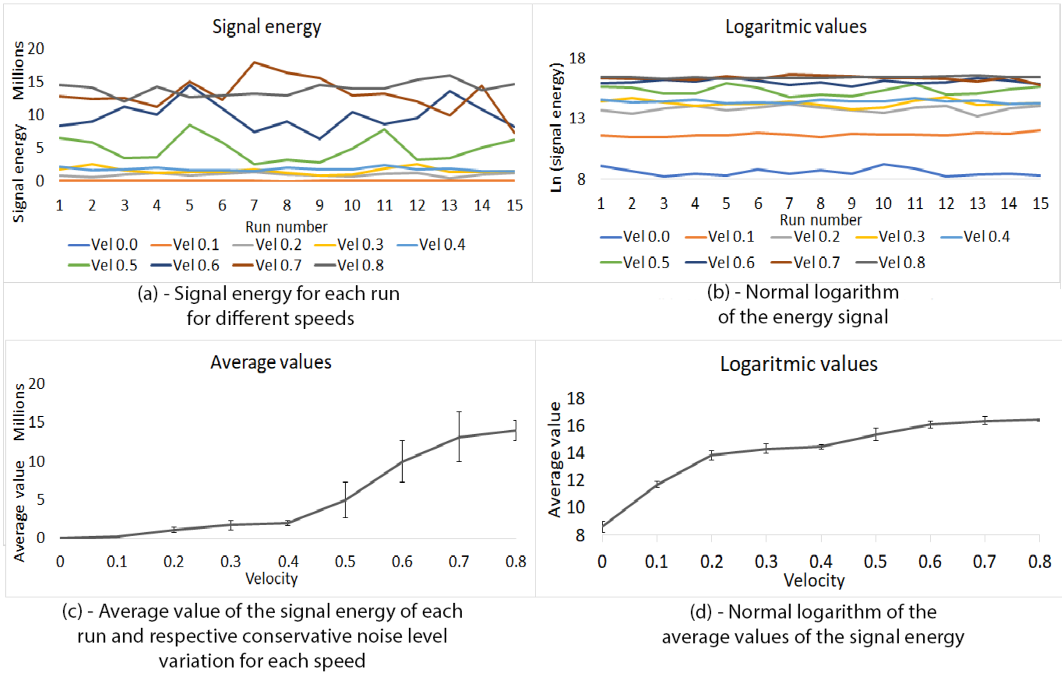
The learning phase is used to store information related to the robot's acoustic noise in a given context, to be used by the motion controller. In this phase, the robot moves forward at different speeds (from 0.0 to 0.8 m/s, in 0.1 m/s steps) for about 2 seconds while it gathers the amount of noise the robot produces. This is repeated 15 times for each speed and each context. For the remaining of this chapter, each repetition is refereed to as a *run*.

5.1.1 Signal energy

As previously discussed in chapter 4, only a portion of the data gathered by the microphone is actually used. The signal energy of that data and then the natural logarithm of that energy is calculated. To better understand the need to calculate the ln of the energy, a set of figures (Figures 5.1, 5.2, 5.3, 5.4 and 5.5) is presented, one for each context, which shows a comparison between the data with and without calculating the ln. For all of them, the top plots represent the signal energy for each run for the different speeds. The bottom plots represent the average value of the signal energy of each run and the respective standard deviation for each speed.

As it is possible to verify, by calculating the normal logarithm of the energy signal, the data gets much more linear and easier to understand. By analyzing the top charts, related to the energy signal of each run, it is possible to see how much normalized the values are, with all of the contexts having values compressed between 0 and 20.

In section 3.2, it was stated that it is necessary to store the information about the average noise levels and the conservative noise level variation. The bottom plots represent these values, and again, the data gets normalized to lower numbers. On Section 3.3, it was mentioned that the motion controller calculates the average noise values through an equation system. By looking at the plots at the bottom of each figure, that are not the normal logarithm values, the average values could be



(a) - Signal energy for each run for different speeds

(b) - Normal logarithm of the energy signal

(c) - Average value of the signal energy of each run and respective conservative noise level variation for each speed

(d) - Normal logarithm of the average values of the signal energy

FIGURE 5.1: Signal energy when using its normal logarithm values and not, for the tiles context.

represented through an exponential equation or a high polynomial equation. But with the normal logarithm values, a simple linear equation is enough, although a better discrimination can be achieved. For the first speed values, there is a high variation on the values and can be described with second degree polynomial equation, since it can be arguable these values form a curve. The rest of the speed values resemble a line, which can be represented through a linear equation. A second degree polynomial and a linear equation requires less computational power and, by extension, less time to be processed than a exponential equation and it also makes sure that better fits the real data. Figure 5.6 show the plots with the representations of the equations to calculate the average value for each context.

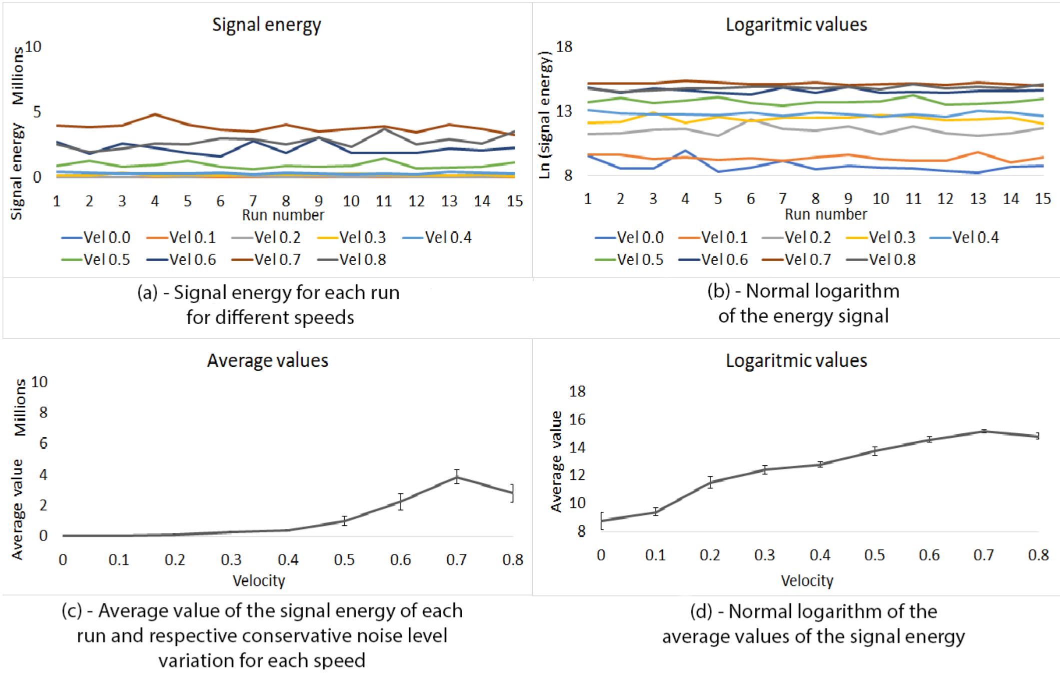


FIGURE 5.2: Signal energy when using its normal logarithm values and not, for the carpet context.

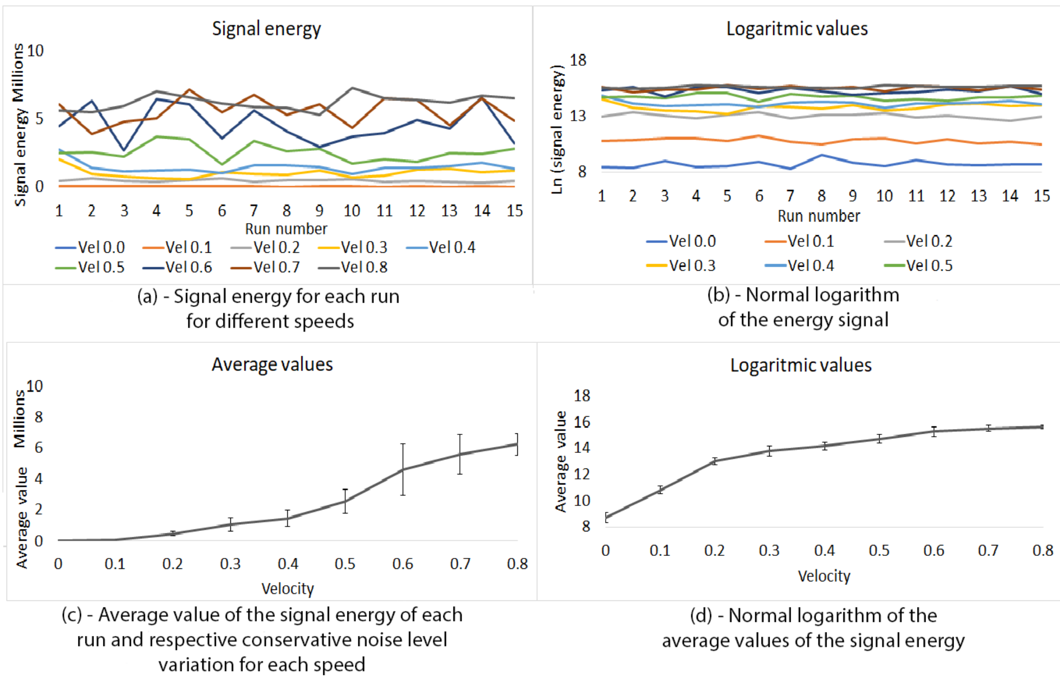
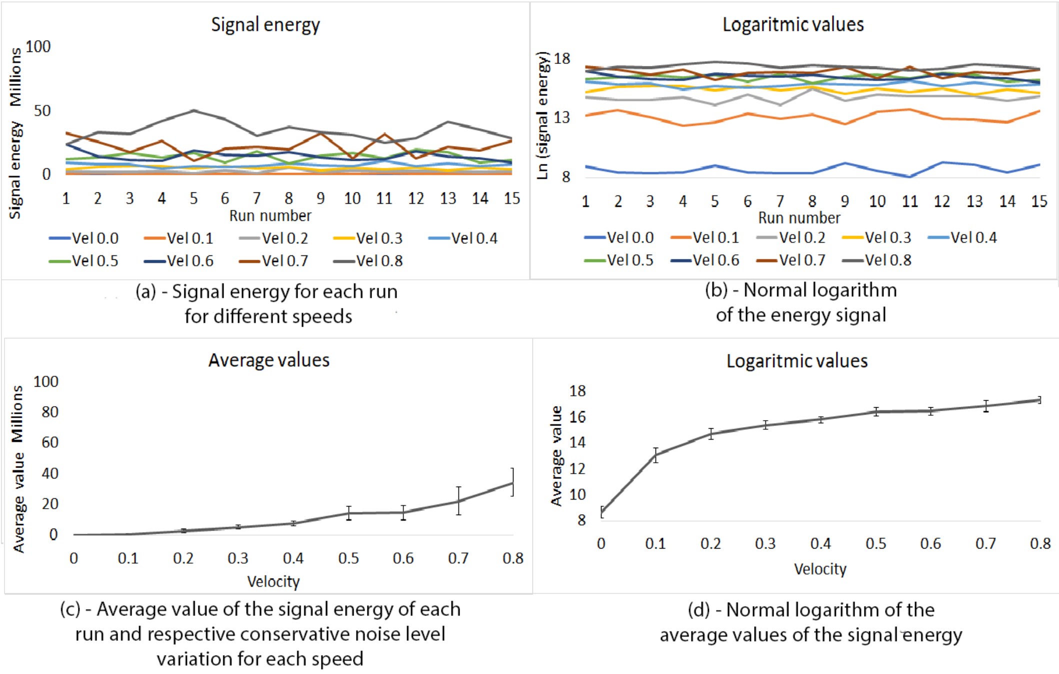


FIGURE 5.3: Signal energy when using its normal logarithm values and not, for the wood context.



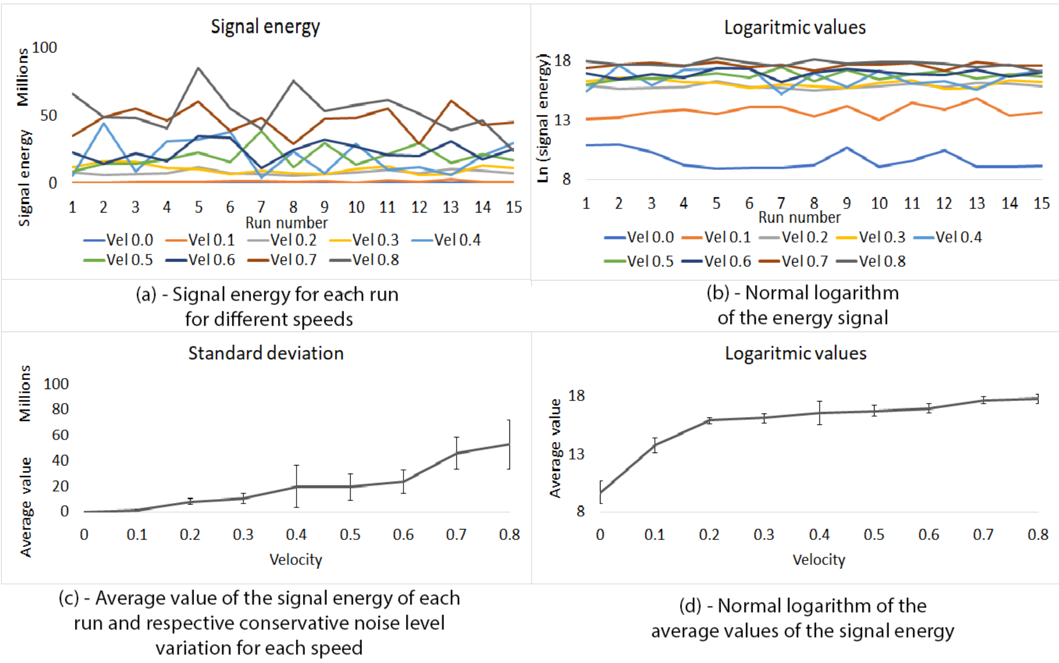
(a) - Signal energy for each run for different speeds

(b) - Normal logarithm of the energy signal

(c) - Average value of the signal energy of each run and respective conservative noise level variation for each speed

(d) - Normal logarithm of the average values of the signal energy

FIGURE 5.4: Signal energy when using its normal logarithm values and not, for the indoor concrete context.



(a) - Signal energy for each run for different speeds

(b) - Normal logarithm of the energy signal

(c) - Average value of the signal energy of each run and respective conservative noise level variation for each speed

(d) - Normal logarithm of the average values of the signal energy

FIGURE 5.5: Signal energy when using its normal logarithm values and not, for the outdoor concrete context.

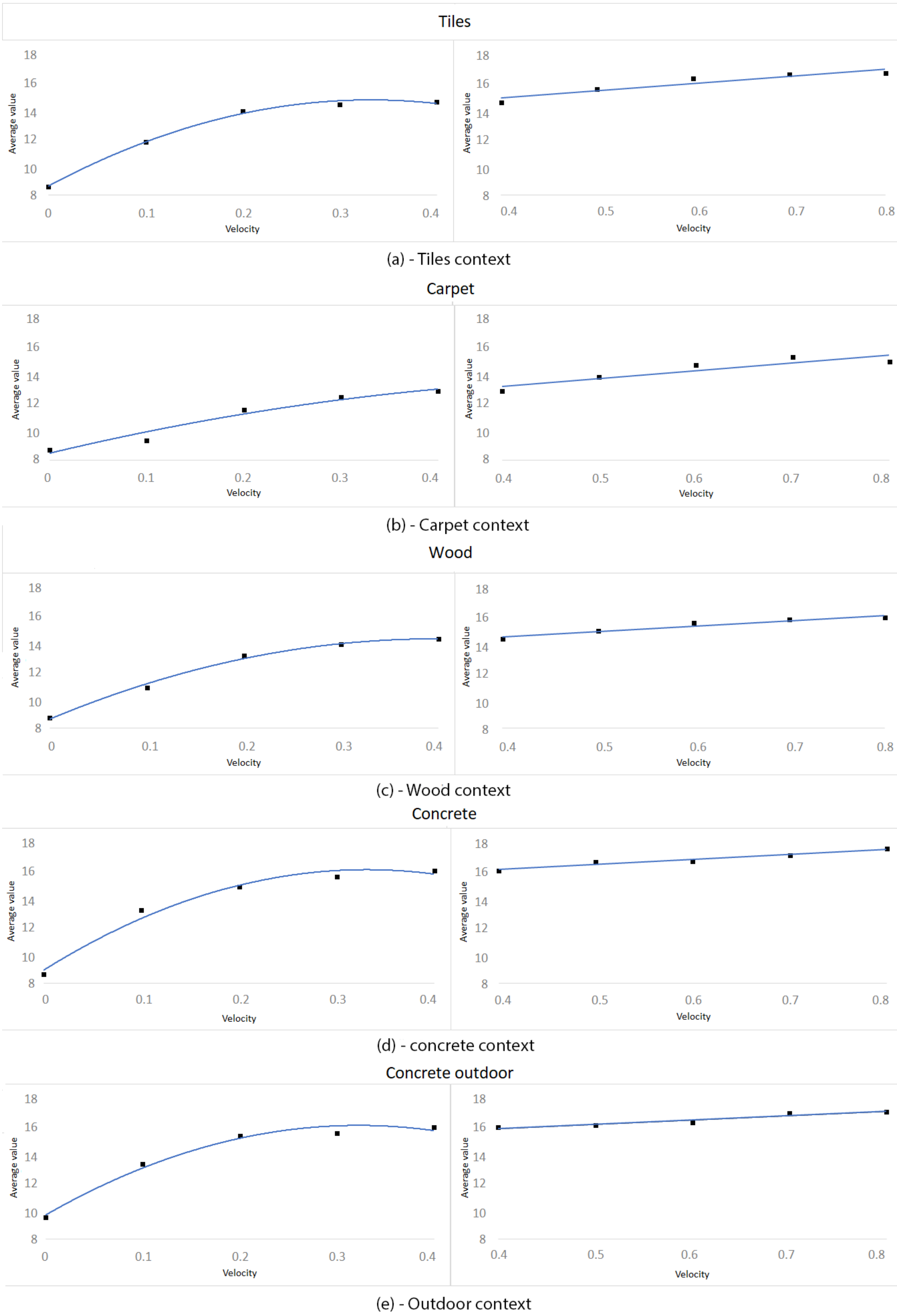


FIGURE 5.6: Charts with the average value of the signal energy and the representation of the equation that best represent the data. The dots represent the average value for each speed and the line represent the equation.

5.1.2 Difference between contexts

Through the learning phase it is also possible to compare the different contexts by analyzing the amount of noise the robot produces at each context. Figure 5.7 shows a plot in which the average value and the respective conservative noise level variation of the signal energy for each speed and context is represented.

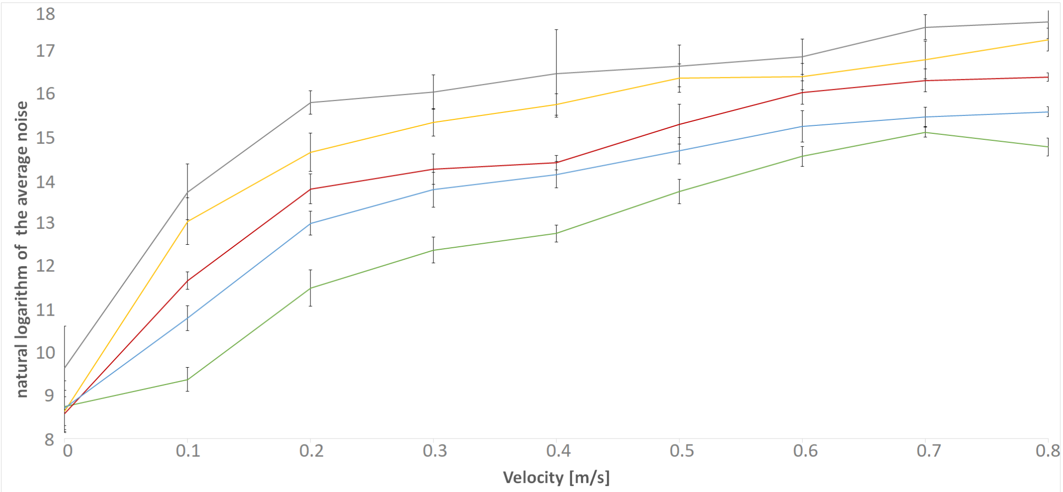


FIGURE 5.7: Learning phase results showing the robot's average noise level and conservative noise level variation for the different contexts. Contexts from top to bottom: concrete_outdoor, concrete, tiles, wood and carpet. The horizontal axis represents the speed of the robot in m/s, and the vertical axis represents the average noise of the robot. The vertical lines at each velocity represents the conservative noise level variation.

The plot shows that in environments with concrete floor the robot produces the highest level of acoustic noise, since the floor is the hardest of all tested, with the outdoor situation producing more acoustic noise than the indoor one. This may be due to the floor composition. Although both are concrete floors, they can be composed of different portions of the materials used to make the floor. Also, below the floor on the outdoor context, there is a small compartment, while on the indoor context, there is the ground below. By having a hollow space below the floor, the robot's movement can originate different vibrations on the floor, which makes it produce more noise. Also predictably, the carpet context makes the robot produce the least acoustic noise, because a carpet is a softer floor type than any of the other contexts. It is also possible to verify a clear distinction between the contexts. Although a higher difference between noise values of the different contexts can be preferable, it is enough, in a future improvement, to create a system that, based on the robot's acoustic noise for a given speed, predicts at which context the robot is in, without needing to have that information given by the user.

5.1.3 Floor type's impact on robot's noise

To understand how the floor type can influence the amount of noise the robot produces, an additional experiment was performed. By placing the robot suspended in the air (wheels not touching the ground) and by making it try to move the robot at different speeds, it is possible gather the information of how much noise the robot does when not moving on top of anything and how much the floor type impact the amount of noise generated. Also, with the information of the environment's natural noise (when there is nothing extraordinary producing noise), it is possible to analyze the impact of having a specific floor type in a given environment. Figure 5.8 shows a set of plots representing the amount of acoustic noise generated by the robot, for all contexts, when it is moving on the floor and when it is suspended mid air.

As it is possible to verify from the plots, by making the robot move on top of a floor, makes it produce higher volumes of noise than when not in contact with the floor. When the robot is not touching the ground, the amount of noise it produces is similar across the contexts, with the exception of the outdoor situation. Because of that, it is also clear that the softer the floor, the less difference there is between the robot moving on the ground and not.

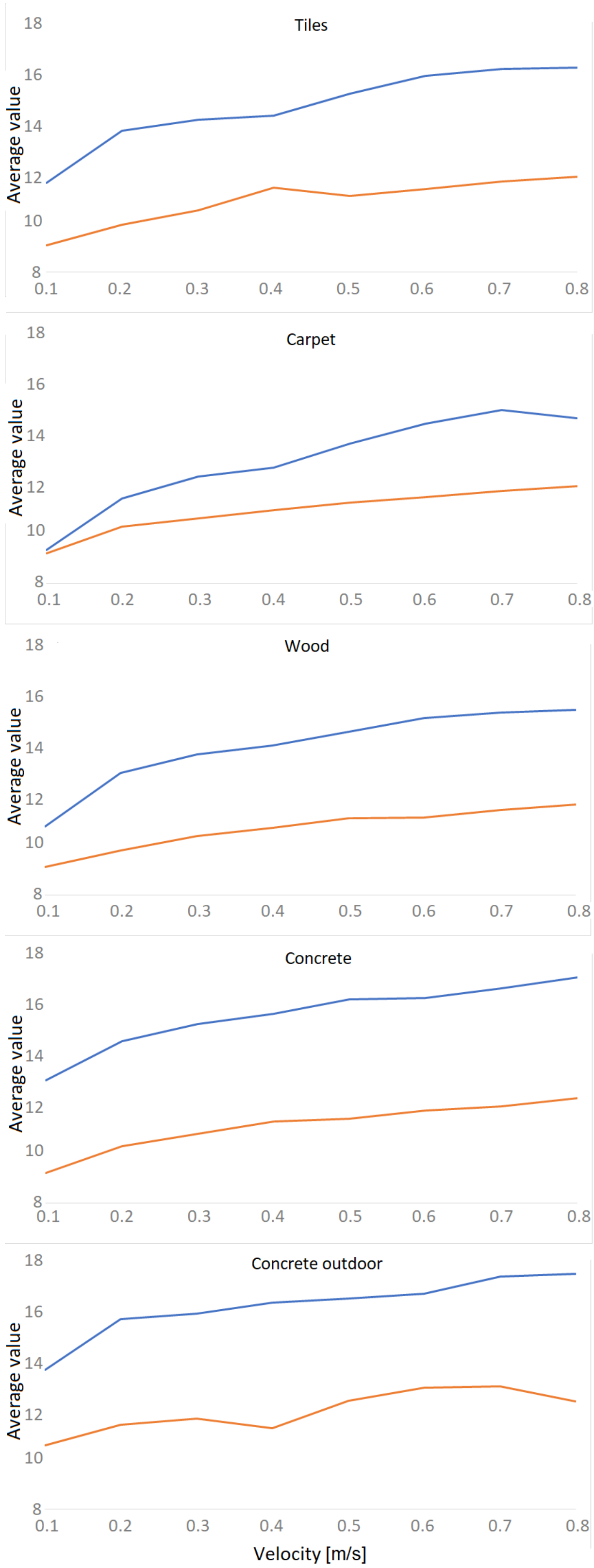


FIGURE 5.8: Difference between the robot's acoustic noise when the wheels are in contact with the floor and not. The top line on all charts represent the robot moving on the floor. The bottom line for all charts represent the robot suspended mid air. Contexts from top to bottom: Tiles, carpet, wood, concrete and concrete in an outdoor situation.

5.2 Motion controller

The motion controller is the part of the system that selects the highest speed possible that the robot can use and not make more acoustic noise than the environment in a given moment, as explained in Section 4.4.

To test the motion controller, the robot was placed idle at each context, except the outdoor concrete (concrete, tiles, cement, wood and carpet). The outdoor concrete was excluded from this test because it was not possible to meet the requirements (reproduce loud sound clips) in the context. Then, one of three determined sounds was played. These sounds are a clip from a crowded area, a clip from a jazz music and a vacuum cleaner. Only the vacuum cleaner was not a recorded sound clip, as a real vacuum cleaner was turned on for the tests. For each sound, except the vacuum cleaner, it was tested when the sound is being played in a high volume and a low volume making a total of five tests for each context. By having two different intensities (low and high volumes), the motion controller was tested against volumes where the ideal speed is on both sides of the defined threshold where the robot's noise is calculated through an equation system (see Section 5.1.1). These sounds, except for the vacuum cleaner, were played by a speaker placed in the environment and each of the three sounds had a specific purpose: the sound clip from a crowded area simulates an environment where there are people inducing background noise; the vacuum cleaner simulates an environment where there is a constant background noise, like an air ventilation system; the jazz song simulates an environment where the background acoustic noise is dynamic, oscillating its volume through time. While these sounds were played, the motion controller acquired data from a microphone about the environment to determine the amount of acoustic noise present and then, it predicted the highest speed possible for the robot. The search speed step α was set to 0.05, to have speeds that makes the robot produce similar acoustic noise as the present in the environment, and the desired speed s_r was set to 2.0 m/s to force the algorithm to search for the best velocity. Table 5.1 shows the results of the experiment.

Table 5.1: Results from the motion controller implementation. The headers are, from left to right: Ambient noise, floor type, environment back noise, predicted robot induced noise, noise difference and noise ratio.

Ambient sound	Floor type	Env. back noise (δ)	Pred. rob. ind. noise (μ_r)	Noise difference ($\mu_r - \delta$)	noise ration [%] ($1 - \mu_r/\delta$)
Jazz High volume	concrete	16.61	16.59	-0.02	0.10%
	tiles	16.63	16.62	-0.01	0.03%
	wood	16.53	16.51	-0.02	0.15%
	carpet	16.03	16.01	-0.02	0.10%
Jazz Low volume	concrete	14.24	14.20	-0.04	0.30%
	tiles	14.28	14.27	-0.01	0.05%
	wood	13.81	13.76	-0.05	0.34%
	carpet	12.81	12.79	-0.02	0.14%
Crowd High volume	concrete	16.61	16.59	-0.02	0.11%
	tiles	16.40	16.38	-0.02	0.14%
	wood	16.04	16.02	-0.02	0.12%
	carpet	15.94	15.93	-0.01	0.08%
Crowd Low volume	concrete	15.64	15.59	-0.05	0.35%
	tiles	13.91	13.84	-0.07	0.51%
	wood	12.99	12.95	-0.04	0.28%
	carpet	11.80	11.79	-0.01	0.09%
Vacuum cleaner	concrete	15.29	15.25	-0.04	0.27%
	tiles	16.15	16.13	-0.02	0.13%
	wood	15.51	15.50	-0.01	0.04%
	carpet	14.75	14.74	-0.01	0.07%

By looking at the results present in table 5.1, it is clear that the speed indicated by the motion controller should make the robot produce similar amounts of acoustic noise as the present in the environment. This may be due to the use of such a small search step α. By defining α with a small number, such results are expected. Higher value would result in a higher gap between the environment's noise and the robot's noise, with the robot's noise being lower.

This test shows that the system works as expected, because in no context, sound or sound volume, the robot should make more acoustic noise than the environment's, which means that people nearby should not be distracted or troubled by the robot's noise. A more conservative system would have a higher α, to make the robot's noise even less noticeable. The environment's and robot's noise are measured by a microphone placed on the robot, so the farther away from the robot a person may be, the less impact the robot's noise has on the person.

Table 5.2: Results from the motion controller implementation on the outdoor concrete context. The headers are, from left to right: Ambient noise, floor type, environment back noise, predicted robot induced noise, noise difference and noise ratio.

Ambient sound	Floor type	Env. back noise (δ)	Pred. rob. ind. noise (μ_r)	Noise difference ($\mu_r - \delta$)	noise ration [%] ($1 - \mu_r/\delta$)
High volume	concrete	18.00	17.98	-0.02	0.1%
Low volume	concrete	14.50	14.44	-0.06	0.4%

5.2.1 outdoor concrete context

The outdoor concrete context was excluded from the previous test because it was not possible to reproduce the determined sound clips at high enough sound levels to perform the test under the same conditions as for the other contexts. Although it would be interesting to also execute the test in this context, since it was the only one outdoor, only a simpler test could be executed to test the motion controller on this context.

Instead of having the microphone gathering the environment's acoustic noise for the motion controller to execute its algorithm based on that value, the environment's acoustic noise was predetermined. Although the full work flow of the system is not tested, under this setup the motion controller can still be tested for its ability to find the ideal speed for a hypothetical environment's noise value, and thus, give a good enough overview of the motion controller's performance on the context. Table 5.2 presents the results of this test. As it is possible to see on the table, the results are also good, with the algorithm indicating a speed for which the robot produces less acoustic noise than the environment's.

5.3 People detector

To test the ability of the proposed system to be integrated in an application, a second experiment has been conducted. The robot, now equipped with a PIR sensor to detect the presence of a person, moves forward while waiting for the PIR sensor to detect something. When the PIR sensor detects someone, the motion

TABLE 5.3: Results from the second experiment, where the robot stops performing a task with the presence of a person and adapts its velocity to not disturb the person. The headers are, from left to right: Ambient sound, predicted robot induced noise, environment back noise, robot selected speed, noise difference and noise ratio.

ambient sound	pred. rob. ind. noise (μ_r)	env. back noise (δ)	robot selected speed (s)	noise difference ($\mu_r - \delta$)	noise ratio[%] ($1 - \mu_r/\delta$)
Vac. cleaner	14.57	14.89	0.395	-0.32	2.12%
Crowd	14.94	14.96	0.405	-0.02	0.13%
Jazz	14.57	14.79	0.395	-0.22	1.49%

controller is executed. The robot stops, listens to the environment and, when the motion controller indicates a speed that produce less acoustic noise than the environment, moves forward at that speed. The test was performed only in the tiled floor context and the sound clips where the same as for the previous test (crowd sound, vacuum cleaner and a music), except now, all the sounds produced similar noise levels without different volume variations.

For this test, the desired speed was set to 0.5 m/s for a more realistic application. The 2.0 m/s of the previous test is too fast for real life applications. The person appeared in front of the robot approximately at 1 m away for every run. This test showed that the system could work, for example, for an autonomous vacuum cleaner, where the vacuum cleaner could clean a house while there are people inside, as it would produce lower acoustic noise when a person was in the same division, making people comfortable with the robot doing its task near them.

Table 5.3 shows the results for this experiment were positive. The motion controller was successfully integrated into a bigger system and performed well. Similar to the previous test, the robot does not produce more acoustic noise than the environment's. The robot's selected speed, which is the speed indicated by the motion controller, is close to the threshold from which the algorithm chooses between a linear equation or a 2^{o} degree polynomial equation to predict the robot's induced noise. This is a good situation, because it shows that the algorithm also works for sensible values such as near the threshold. It is also possible to see that the algorithm chose the closest speed possible above the threshold for one of the sounds and the closest speed below the threshold for the other two sounds.

Chapter 6

Conclusion and future work

6.1 Conclusion

Despite the robot's task, the acoustic noise it produces might be distressing to people nearby. To avoid that, the amount of acoustic noise the robot produces needs to be regulated so as to not be uncomfortable.

One way of doing that is by controlling the speed at which the robot performs a certain task. For that, the robot needs to learn the amount of acoustic noise it induces while moving at any speed in any context. For this dissertation, a motion controller was developed that should allow different robots to adapt their motion speed, when a human is nearby, to a value that allows the robot to produce less acoustic noise than the present in the environment, avoiding this way distracting people due to its noise.

For the robot to have a notion of how much acoustic noise it generates, a learning phase needs to be performed. This learning phase can be done in any context, but, most importantly, it should be done in contexts where the robot will be performing its tasks. The result of the learning process is a relation between the different speeds and the corresponding average and conservative variation of the noise level for the robot for each context. This phase is the only prerequisite before the robot can use the motion controller in a given context.

The proposed method was validated with two test sets. In the first one, four different floor contexts and three different sound clips were used. The types of floors chosen were cement, tiles, wood and carpet. The sound clips were a jazz song, crowd noise and vacuum cleaner. The jazz song allowed to represent an ambient were the environment's noise is not constant, with many variations. The crowd noise represented an environment with many people. The vacuum cleaner represented a place were the background noise is constant. The robot, placed at each context, runned the motion controller's algorithm and predicted the ideal speed at which it could move without producing more noise than the environment's. It is worth noting that the noise values are obtained from the robot's perspective, and the people's perception might vary according to their distance to the robot.

In order to complement the first test, a second experiment to test whether the motion controller could be implemented into a more complex system was performed. The robot was equipped with a PIR sensor in order to find out if the robot could select an adequate speed each time a person got in the sensor's field of view. This test was performed in one of the studied contexts and showed that the robot is capable of controlling its speed in a way that the noise it produces is lower than the environment's.

The main research question addressed in this dissertation was if it is possible to learn a robot's acoustic impact on a human populated environment, based on data gathered by a set of microphones placed on the robot. Based on the experimental results, the answer is yes. By having a learning phase, where it is measured how much acoustic noise the robot produces at different speeds in different contexts, the robot can execute its tasks at those contexts without inducing more noise than the environment's. The microphone was placed at the middle compartment of the robot, where it could capture the robot's noise without having the noise values saturated. The microphone was facing down, because the robot's motors were at the bottom of the robot.

6.2 Future work

While the obtained results showed that the proposed method works for a variety of contexts, there are still situations where it is nearly impossible for a robot to execute a task without disturbing humans nearby. For instance, in places where there is almost no background noise, like a library, the robot may not be able to perform a given task while maintaining a reasonable speed. In a situation like that, the option could be either execute the task at the lowest speed possible or delay the execution of the task until there is no humans nearby.

Even though the overall results were very positive, there is still room for improvements that can be made in the future. For instance, in the tests carried out, the robot was limited only to forward direction movements. In subsequent work, a more diverse set of movements needs to be considered. The majority of the contexts tested were located indoors and all of them were in flat terrains. Future work should take into consideration different types of contexts, like different outdoor environments and rougher terrains. Another interesting feature, would be the implementation of the motion controller into different types of robots, like an aerial drone. The system itself can be improved, either in overall performance or regarding the work flow. The learning phase could be skipped if the robot could isolate its acoustic noise from the environment's. For example, the robot could capture the environment's noise for a while, and then, while actively listening to the noise, change its speed until the noise present in the environment was not higher than the previously captured. Another possibility could be start moving and periodically test the current speed. For instance, if the robot started at the lowest possible speed, and gradually increased the speed, the environment's noise captured would have similar values until the robot reached a certain speed, at which point the environment's noise would increase as the speed increases. At that point, the robot would know the latest speed tested that did not make the environment's noise to rise. If a microphone were to be installed on the environment, it could increase the system's performance, allowing the robot to have a better knowledge about it's own acoustic noise and the environment's. The two microphones would capture the noise from both the robot and a nearby person's perspective. Another useful scenario could be the installation of more microphones on the robot. The top compartment could have a microphone that would listen to the environment and the bottom compartment could have a microphone to listen to the robot. Further, a microphone could be placed nearby and directed to each robot's sound source, like the wheel motors, and so increasing the understanding of where the noises come from in the robot.

Another possibility would be to consider active noise cancellation where one or more speakers would produce anti-phase audio signals in order to cancel those produced by the driving motors of the robot. This would increase the computing power necessary to address such strategy.

Bibliography

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, *76*(8), 966–1005.

Bajracharya, M., Howard, A., Matthies, L. H., Tang, B., & Turmon, M. (2009). Autonomous off-road navigation with end-to-end learning for the lagr program. *Journal of Field Robotics*, *26*(1), 3–25.

Baleia, J., Santana, P., & Barata, J. (2015). On exploiting haptic cues for self-supervised learning of depth-based robot navigation affordances. *Journal of Intelligent & Robotic Systems*, *80*(3-4), 455–474.

Ballard, D. H. (1991). Animate vision. *Artificial intelligence*, *48*(1), 57–86.

Basner, M., Babisch, W., Davis, A., Brink, M., Clark, C., Janssen, S., & Stansfeld, S. (2014). Auditory and non-auditory effects of noise on health. *The Lancet*, *383*(9925), 1325–1332.

Cook, K. L. B. (2007). The silent force multiplier: The history and role of uavs in warfare. *IEEE Aerospace Conference*, 1–7.

Gibson, J. (1977). The concept of affordances. In *Perceiving, acting, and knowing* (pp. 67–82). Lawrence Erlbaum.

Hall, E. (1969). *The Hidden Dimension : man's use of space in public and in private* (B. Head, Ed.). Bodley Head.

Heidarsson, H., & Sukhatme, G. (2011). Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 3160–3165).

Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., & Santos-Victor, J. (2016). Affordances in psychology, neuroscience and robotics: a survey. *IEEE Transactions on Cognitive and Developmental Systems*, 4–25.

Kallakuri, N., Even, J., Morales, Y., Ishi, C., & Hagita, N. (2013a). Probabilistic approach for building auditory maps with a mobile microphone array. In *Proceedings of the ieee international conference on robotics and automation (icra)* (pp. 2270–2275).

Kallakuri, N., Even, J., Morales, Y., Ishi, C., & Hagita, N. (2013b). Using sound reflections to detect moving entities out of the field of view. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 5201–5206).

Kruse, T., Pandey, A., Alami, R., & Kirsch, A. (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, *61*(12), 1726–1743.

Lacey, S., Hall, J., & Sathian, K. (2010). Are surface properties integrated into visuohaptic object representations? *European Journal of Neuroscience*, *31*(10), 1882–1888.

Luber, M., Spinello, L., Silva, J., & Arras, K. O. (2012). Socially-aware robot navigation: A learning approach. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 902–907).

Marques, F., Gonçalves, D., Barata, J., & Santana, P. (2017). Human-aware navigation for autonomous mobile robots for intra-factory logistics. In *Proceedings of the 6th international workshop on symbiotic interaction* (pp. 79–85).

Martinson, E. (2007). Hiding the acoustic signature of a mobile robot. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems* (pp. 985–990).

Martinson, E., & Schultz, A. (2006). Auditory evidence grids. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems* (pp. 1139–1144).

Martinson, E., & Schultz, A. (2007). Robotic discovery of the auditory scene. In *Proceedings of the ieee international conference on robotics and automation* (pp. 435–1144).

Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovrakis, D., & Chryssolouris, G. (2015). Design considerations for safe human-robot collaborative workplaces. *Procedia CIrP*, *37*, 248–253.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, *1*(1), 33–55.

Pinto, E., Marques, F., Mendonça, R., Lourenço, A., Santana, P., & Barata, J. (2014). An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception. In *Proceedings of the ieee international conference on robotics and biomimetics (robio)* (pp. 443–450).

Pombeiro, R., Mendonça, R., Rodrigues, P., Marques, F., Lourenço, A., Pinto, E., … Barata, J. (2015). Water detection from downwash-induced optical flow for a multirotor uav. In *Proceedings of the oceans'15 mts/ieee washington* (pp. 1–6).

Rios-Martinez, J., Spalanzani, A., & Laugier, C. (2015). From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics*, *7*(2), 137–153.

Schwenkler, J. (2013). Do things look the way they feel? *Analysis*, *73*(1), 86–96.

Takayama, L., & Pantofaru, C. (2009). Influences on proxemic behaviors in human-robot interaction. In *Proceedings of the ieee/rsj international conference on intelligent robots and systems, iros* (pp. 5495–5502).

Uğur, E., & Şahin, E. (2010). Traversability: A case study for learning and perceiving affordances in robots. *Adaptive Behavior*, *18*(3-4), 258–284.

van Hecke, K., de Croon, G., van der Maaten, L., Hennes, D., & Izzo, D. (2018). Persistent self-supervised learning: From stereo to monocular vision for obstacle avoidance. *International Journal of Micro Air Vehicles*, *10*(2), 186–206.

Wurm, K. M., Kretzschmar, H., Kümmerle, R., Stachniss, C., & Burgard, W. (2012). Identifying vegetation from laser data in structured outdoor environments. *Robotics and Autonomous Systems*, *62*(5), 675–684.