RESEARCH ON ACOUSTIC TECHNOLOGIES AND PROCESSING CONCEPTS TO ACTIVATE AND CONTROL DEVICES BY GESTURE

DOCTORAL THESIS

Borja Saez Mingorance

Doctoral Programme in Information and Communication Technologies
Department of Electronic and Computer Tecnology
University of Granada

January 2022

Editor: Universidad de Granada. Tesis Doctorales
Autor: Borja Sáez Mingorance
ISBN: 978-84-1117-243-1
URI: http://hdl.handle.net/10481/73136

Este documento está preparado para ser impreso a doble cara.

RESEARCH ON ACOUSTIC TECHNOLOGIES AND PROCESSING CONCEPTS TO ACTIVATE AND CONTROL DEVICES BY GESTURE

Directed by:

Prof. Diego Pedro Morales Santos
Prof. Encarnación Castillo Morales

Developed at:

Infineon Technologies AG
BEX RDE RDF ISS
Munich, Germany

Doctoral Programme in Information and Communication Technologies
Department of Electronic and Computer Tecnology
University of Granada

January 2022

Acknowledgements

We must always find the time to thank
the people who change our lives
John F. Kennedy

As in quantum mechanics, the uncertainty principle also applies to person relationships. It is impossible to accurately know who the people who you choose during your life will help you to achieve your initial goals. But with persons, as your experiment conclude, we have the chance to look at those who really helped you and express your gratitude.

First, I want to Diego P. Morales and Encarnación Castillo, my thesis directors, for the incommensurately help offer during this period. Beyond the technical knowledge, both of you understood when put or release pressure on me, help when the results where not coming at the expected rate and set always ambitious objectives. I feel extremely honored to have you guiding me during this period. Thanks Diego for that call that unofficially was the starting point for this incredible journey. Thanks for being always close and sincere, and for those "person reading" sessions. Thanks Encarni for keeping always the feet on the ground, calling us back when was necessary. Thanks for the exhaustive and always right reviews. This is not a farewell, I promise!

I want to thank also IFAG RDE RDF for allowing me to have this opportunity. Thanks to all Infineon and Eesy's colleagues for every moment we could enjoy together (or via WebEx), not only the technical ones, but also coffee, lunch, "merienda" or waitting the S-Bahn, all the moment when the expertise and knowledge was transfer in a smooth but warm way, making the doctoral period also a very important person growing period. Do not want to forget any name (or maybe yes), but certainly I want to highly here three names: Antonio Escobar, Marta Verona and Javier Mendez. Thank for being there in the period when I most needed you.

Even though being far far away, I would like to thank those friends who from Spain make feel the distance short. Thanks "Verbeneros" and "Telecos Dispersos" for all the good moments at HOME, being a huge help for recharging batteries and helping me to go through this period abroad easily.

An it is time for the families, starting for the one that one choose, the

Ultimate family. Thank MINT for being my family in Munich. I cannot imagine how different could have been this years abroad without all you. Thanks for the laughs, tears, training, beers, soared-muscles, trips, skiing-weekends, long conversation. Since the first practice at the Englischer you made me feel at home, and this is something I will never forget. Of course I want to thank also Disckatus and Granayd, thanks for make me feels always part of the team, even allowing me to join tourneys, or joining me here to play tourneys. Ultimate Frisbee is a cult, but thanks to people like you I am proud to tell I do belong to this cult!

Por último, pero más importante, mi verdadera familia. Gracias por siempre confiar en mi, empujándome sin quererlo a superarme cada día. Gracias por no limitar mis ideas, por apoyar las "pollaicas"de Borja, y estar siempre ahí para todo lo necesario. Gracias a todos, cañoneros y lagartos, por recortar distancias y estar siempre disponibles para ayudar de cualquier forma. Gracias por hacerme sentir parte del crecimiento de los enanos, con esas fotos y videos que te sacan la sonrisa los días más difíciles. Por supuesto, gracias a mis padres, Antonio y Matilde, y a mi hermana, Helena, por sufrirme durante estos largos 28 años. Puedo decir orgulloso que todo lo que hoy soy se basa en los valores que he aprendido de vosotros, y que no cambiaría de un ápice de esos cimientos que construísteis a base de esfuerzo y trabajo. ¡GRACIAS!

I do not want, but certainly I do forget people. If I have meet you on the path, and you have left your impression on me, thank you too.

Resumen

Los modos de interacción con sistemas están evolucionando para mimetizar la comunicación humana. Esto implica que estos sistemas están pasando de basarse en comandos de entrada únicos (como pulsar teclas, comandos de voz, gestos) a una interpretación de varios comandos. En otras palabras, la interacción se convertirá en un sistema multimodal, capaz no sólo de reconocer lo que el usuario está dando como entrada, sino también de modificar el significado de esta entrada en función del contexto en el que se ha proporcionado. Esto implica la necesidad de introducir nuevas tecnologías en el campo del control de dispositivos para incrementar la información que se recoge del usuario.

Esta tesis doctoral es el resultado de la investigación de tecnologías acústicas, en concreto ultrasonidos, para la activación y control de dispositivos a través de gestos. Dichos resultados se presentan en forma de compendio de publicaciones, recopilando los artículos científicos publicados durante el periodo doctoral.

La investigación se ha realizado en paralelo al desarrollo de un innovador transductor ultrasónico. El objetivo ha sido estudiar la viabilidad del uso de dicho sensor como único método de entrada para sistemas de reconocimiento de gestos. Se ha evaluado y demostrado la posibilidad de utilizar sistemas con recursos limitados, como Edge devices, como dispositivos para la adquisición y procesado de la señal obtenida por el sensor ultrasónico. Posteriormente se han investigado algoritmos para el posicionamiento de objetos (por ejemplo, una mano) basados en el tiempo de vuelo (ToF por sus siglas en inglés), generalizándolos para más de un sensor. Por último, se ha desarrollado un sistema de escritura aérea, capaz de obtener y reconocer una serie de caracteres alfanuméricos realizados por el usuario en el aire. Dicho sistema puede ser generalizado a otro tipo de trayectorias o gestos.

La investigación se ha realizado bajo un contrato doctoral en las instalaciones de Infineon Technologies AG, en su sede principal de Múnich, Alemania.

Abstract

Human interaction systems (HSI) are evolving to mimic human communication. This implies that these systems are moving from being based on single input commands (such as key presses, voice commands, gestures) to a multi-command interpretation. In other words, interaction will become a multimodal system, capable not only of recognizing what the user is giving as input, but also of modifying the meaning of this input depending on the context. This implies the need to introduce new technologies in the field of device control to increase the information collected from the user.

This doctoral thesis is the result of the investigation of acoustic technologies, specifically ultrasound, for the activation and control of devices through gestures. These results are presented in the form of a compendium of publications, compiling the scientific articles published during the doctoral period.

The research has been carried out in parallel with the development of an innovative ultrasonic transducer. The objective has been to study the feasibility of using this sensor as a unique input method for gesture recognition systems. The possibility of using systems with limited resources, such as Edge devices, as devices for the acquisition and processing of the signal obtained by the ultrasonic sensor has been evaluated and demonstrated. Subsequently, algorithms for object positioning (e.g. a hand) based on time-of-flight (ToF) have been investigated and generalized to more than one sensor. Finally, an airborne writing system has been developed, capable of obtaining and recognizing a series of alphanumeric characters made by the user in the air. This system can be generalized to other types of trajectories or gestures.

The research has been carried out under a doctoral contract at the facilities of Infineon Technologies AG, at its headquarters in Munich, Germany.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

La mejor herencia que podemos dejaros
es vuestra educación
Matilde Mingorance

Charles Dickens wrote “Electric communication will never be a substitute for the face of someone who with their soul encourages another person to be brave and true” reflecting skepticism about the new technology called Telegraph. Nowadays, and more evidently after those strange 2020 and 2021, it is clear that this electric communication has evolved not to be a substitute for human communication, but to facilitate face-to-face conversation even when the distance makes it physically impossible. But this is just one more step, as the technology keeps evolving to not only connect us through video calls but “be a feeling of presence, like you are right there with another person or in another place”, as Mark Zuckerberg defines the new concept of the metaverse. An here is where a new sense of communication arises. Not only human to human communication is important, but human-system communication, or Human System Interaction (HSI) as it is known, needs to be further developed. Only then the conception of a new virtual space where the interaction with other users will mimic the interaction in real life can be achieved. Well-known virtual personal assistants such as Alexa or Siri, developed by Amazon and Apple respectively which allow communication with the system using only voice commands, allowing the user to interact with diverse house devices, ask for information, or make a phone call.

As Dickens reflected in his quote, human communication is based on a multi-modal system where not only words are important to transmit a message, but body language, voice tone or intensity, or facial expressions share an important role in the meaning of those words. HSI is evolving from well-known interfaces like keyboards, buttons, or touchscreens to other interfaces, as voice commands, to mimic the natural human communication process. For example, with the emerging of Virtual Reality (VR) technology, the actions of the user need to be introduced into the virtual environment. Cameras, motion sensors, or even radar sensors are used to translate the user movements (or gestures) into the virtual world. In other cases, those gestures may be needed to perform actions while non compromising the user focus on other more important aspects. Vehicles infotainment systems are integrating sensors to allow the user to perform gestures to interact with the car without needing to use the touchscreen or remove the sight from the road to look for a specific button.

1.1. Motivation

Parameters like power consumption, computational requirements, physical constraints, or privacy are a key factor in the development of new HSI systems. Ultrasonic technology offers a good trade-off between the parameters above and the complexity of gestures that can be detected. According to Cambridge Dictionary, a gesture is defined as "a movement of the body, hands, arms, or head to express an idea or feeling".

Ultrasound technology is not an emerging technology, has been used for decades in applications as medical imaging, or material fault detection. For those applications the sensor and system used are complex, and with an ultrasonic frequency not adequate to be used in air transmission. Ultrasonic transducers have also started to be used in airborne applications: bulk piezo-electric sensors allow distance ranging based on the time elapsed between the emission and reception of a pulse, known as Time of Flight (ToF). Those sensors present mainly one inconvenience, the volume of the sensor can be too big to fit in in everyday use devices. The development of new transducers based on microelectromechanical systems (MEMS) microphones decreases the space needed for the integration, opening new applications. Those transducers are based on commercial microphones that are currently on the market, with some changes that allow them to produce ultrasound, but without dropping their audio capabilities.

As a new ultrasonic application, there are also several systems that introduce gesture control, i.e. SoundWave [1], AudioGest [2], Dolphin [3], or UltraGesture [4]. All of them use very low-frequency ultrasound signals to recognize between 5 and 12 gestures, which are mostly based on the Doppler shift effect (frequency variation due to movement) while running the recognition algorithms on PC or Smartphones. The use of ultrasound technology with this purpose presents some key advantages:

- Low power consumption.
  Nowadays most gesture recognition systems are based on optical sensors (image recognition), which have a considerable power consumption. The advantages of ultrasound transducers are that the power

consumption is quite lower than the previous. Additionally, as it is shown by DasIvan et al [5], changing the optical sensor by an ultrasound sensor can produce an increase by a factor of 200 in the number of gestures that can be detected with the same device. These results are based on the specification of head-mounted displays.

- Easy integration.
  That means that the resulting system can be quickly implemented in devices that already have microphones, without the requirement of extra parts or dedicated hardware. This is something that could have a positive impact on the adoption of this new technology, because the option of reusing existing hardware, or not having to add a new one, could help to “break a significant barrier” [1].

- Privacy.
  The advantage of this technology is that it will only capture the frequencies above human hearing range, or in the case, it will be used in devices that have already a microphone, and it will reduce the time that other sensors, like a camera, are actively checking the user.

1.2. Objectives

This work aims to design, model, and develop signal processing and recognition algorithms for gesture recognition systems based on ultrasound technology. The target system uses only ultrasound technology as a sensing method, not supported by any other technologies (i.e. radio or image). The algorithms have been developed with the final goal of deploying the system in Edge Devices, allowing future users to develop standalone applications running in environments where no PC is available, or not desirable.

The results obtained during this research period, and by using the transducers further described in Chapter 2, set the starting line for developing multi-modal HSI with ultrasonic signals as basis technology.

1.3. Outline/Thesis Structure

The present document constitutes a thesis by a compendium of publications, which means it is formed by published papers as result of the research performed during the doctoral period. A brief description of each one of these contributions is provided below:

Publication I The aim of this work is to research the viability of ultrasounds as the technology used for gesture recognition. The main focus is to study whether the signal is possible to be processed in devices with limited resources, Edge devices, in particular, performing at least the signal acquisition and main parameters extraction. Based on the Time of Flight (ToF) signals obtained from two transducers, the work proves is possible to distinguish among 7 different two-dimensional gestures previously defined.

Publication II This work features a framework for trajectory-based data generation. For its implementation, a novel two-step algorithm has been defined and tested. The algorithm is based on Scaling-by-Majorizing-a-Complicated-Function (SMACOF) and the Limited-Memory-Broyden-Fletcher-Goldfarb-Shannon optimization (LM-BFGS) algorithms, which has been used before as data representation algorithms. This two-step model calculates the position of the transducers as well as the position of the obstacle, based on the pairwise distance among them. For the trajectory, the position estimated is recorded as a series of temporal positions, and different filters are applied to eliminate adverse effects as noise in the measurements or outliers. The work has been particularized for the ultrasonic data generation, but the framework can be generalized by modifying the parameters of the technology to use (as noise figure or distance range).

Publication III This work proposes a new air-writing system based only on ultrasonic signals. Based on the knowledge acquired in the two previous publications, this work uses an array of four transducers to locate and track the user hand-marker. Based on the pairwise distance among hand-marker and each transducer, and using the algorithm developed in Publication II, the system is able to obtain the character drawn by the user in the air. For the recognition, different Deep learning techniques have been tested, keeping the focus on the timing for use in real applications. For training and testing the algorithms, a database with four digits (“1”, “2”, “3” and “4”) and four characters(“A”, “B”, “C” and “D”) has been created.

This document is structured as follows: this introductory chapter gathers a brief review of the State of the Art and Motivation, as well as a description of each of the publications achieved. Afterward, Chapter 2 covers the research context and the methodology used in this work. Furthermore, Chapters 3,4,5 collect the full text of the publications introduced above. Finally, Chapter 6 summarizes the results achieved in this doctoral period and describes the future trends.

References

[1] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: using the doppler effect to sense gestures. *Proceedings of the SIG-*

CHI Conference on Human Factors in Computing Systems, pages 1911–1914, 2012.

[2] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pages 474–485, 2016.

[3] Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin, and Zhang Sanfeng. Dolphin: Ultrasonic-based gesture recognition on smartphone platform. *2014 IEEE 17th International Conference on Computational Science and Engineering*, pages 1461–1468, 2014.

[4] Kang Ling, Haipeng Dai, Yuntang Liu, and Alex X Liu. Ultragesture: Fine-grained gesture sensing and recognition. *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2018.

[5] Amit Das, Ivan Tashev, and Shoaib Mohammed. Ultrasound based gesture recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 406–410. IEEE, 2017.

Chapter 2

Materials and Methodology

En esta vida las luces las pone aquel que las tiene

Antonio Sáez

In this chapter, the main methodological aspects related to the development of the different results achieved in this thesis are presented. After defining the research context, the first part describes the transducer used as the main sensor during the research. The second part presents the tools and software needed for data acquisition and data processing.

The Ph.D. research has been done while being part of the RD Funding department at Infineon Technologies AG. The tasks of this contract consist not only to perform the technical work and demonstrators presented in this thesis, but also collaborating in the Proposal phase, Reports/Deliverables writing, and internal project management. The Ph.D. contract was attached to different EU and German-funded projects related to the research topic:

- SILENSE

SILENSE ((Ultra) Sound Interfaces and Low Energy iNtegrated Sensors)[1], co-financed by the European Union's HORIZON 2020 program and German Federal Minister of Education and Research (BMBF), researches both hardware and software blocks for the development of gesture control, data communication, and indoor positioning based on innovative acoustic technologies.

The main objective of this project is to lower the cost and power consumption of the transducers, by improving the transducers themselves, and developing an optimized package, and designing low-power integrated circuits. On the software level, the target is providing smart algorithms enabling communication and sensing using the transducers developed.

SILENSE started in May 2017, with a duration of 36-months. A consortium of 32 partners from 9 different countries worked together to achieve the goals described. The total investment for the project is M€ 29. The consortium is formed by private companies (i.e. NXP, Continental, Grupo Antolin) and research centers and universities (i.e. TU Delft).

- SEMULIN

Funded by the German Federal Ministry for Economic Affairs and Energy, Project SEMULIN (Self-Supporting Multimodal Interaction) [2] is a research project that will study and develop a self-supporting, natural human-machine interface for automated driving using multi-modal input and output modalities.

By merging psychological models with complex machine learning algorithms, the goal is to develop an HMI intuitive and natural, minimizing the errors. By using video, audio, and ultrasound technologies the system's target is to recognize the user's facial expressions, gestures, gaze, and speech.

SEMULIN started in November 2020, with a duration of 36-months. The total investment for the project is M€ 6.3. The consortium is formed by AudEERING, Blickshift, Eesy Innovation, Elektrobit Automotive, Fraunhofer IIS, Infineon and the University of Ulm.

- MARVEL

Financed by the European Union's HORIZON 2020, Marvel project [3] researches a disruptive Edge-Fog-Cloud computing framework for audio-visual scene recognition based on Artificial Intelligence and Big Data algorithms. The goal is event detection in a smart city environment.

Deliver AI-based multi-modal perception and intelligence for audio-visual scene recognition, event detection, and situational awareness in a smart city environment. The main challenge this project faces is the collection, stream, and analysis of audio and video without violating ethical and privacy limits, but fulfilling the goal established.

MARVEL started in January 2021, with a duration of 36-months. The MARVEL consortium consists of 17 partners based in 12 different countries. The total investment for the project is M€ 6.

2.1. Ultrasound Transducer

As has been briefly commented in Chapter 1, the reference sensor for the research in this period is a MEMS ultrasound transducer [4], developed by Infineon Technologies. The research started with the very first version of the sensor. Both research and sensor development has run in parallel, which derived from the necessity of modifications in the experimental designs or work previously done.

As of today, Infineon is the market leader in the microphone sector. Many of our everyday-use devices are equipped with those sensors. Based on the knowledge acquired during the process to position the microphones on the best sellers in the market, there have been other ideas to exploit this expertise. One of these ideas is the transducer used in this research. It is a capacitive ultrasonic transducer (CMUT), based on dual-backplate MEMS technology, depicted in Figure 2.1. It is compound by a flexible membrane which is deformed with the acoustic pressure waves, and two fixed backplates around. An application-specific integrated circuit (ASIC) measures the capacitance variance produced by the membrane deformation. For sending a signal, the process runs backward. An electric signal is applied to one of the backplates, producing a deformation (attracting or repelling) on the membrane. This movement produces a change in pressure.

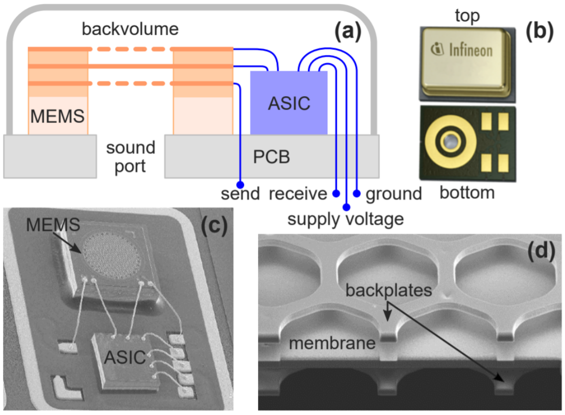


Figure 2.1: (a) Schematic of the transducer module consisting of a MEMS, ASIC, PCB and metallic lid, (b) ultrasonic transducer package, (c) SEM picture of an assembled prototype and (d) zoomed view of the membrane and backplates. [4]

2.2. Transducer readout

As the transducer described in the previous sensor provides an analog output signal, it is necessary to digitalize it before performing any processing. During the work performed in this research, two devices has been mainly used for this task:

- Analog Discovery 2

The Analog Discovery 2, presented in Figure 2.2, is a device produced by Digilent. It is a versatile device which provides an Oscilloscope, Waveform Generator, Logic Analyzer, Protocol Analyzer, Spectrum Analyzer, and Power Supplies functionalities, controlled by custom software (for PC or Raspberry Pi). The connection with the computer is via only USB. Digilent provides too a software development kit (SDK) which allows the connection with other software (as Matlab or Python), easing the information transference and control commands. In the analog reading specifications, it provides 2 differential channels, with 14-bit resolution each (absolute resolution up to 0.32 mV)

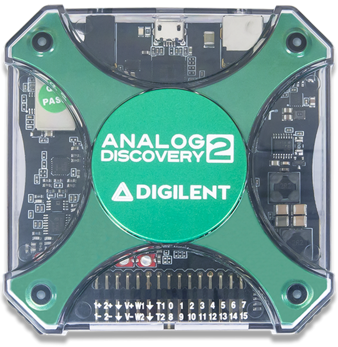


Figure 2.2: Analog Discovery 2 [5]

- XMC4700

  The XMC4700 is an Infineon microcontroller based on an ARM Cortex M4 processor core. Among the microcontrollers developed by Infineon for multipurpose applications, the XMC4000 family is the family with a higher performance featuring digital signal processing (DSP) and float point unit (FPU) capabilities. Besides the CPU core, this XMC includes, among other peripherals, four independent Analog-Digital-Converter (ADC) with 12-bit resolution (absolute resolution up to 1.2 mV). Infineon offers a development kit based on this microcontroller, the XMC4700 Relax Kit (Figure 2.3), facilitating the interconnection with the computer or other sensors (i.e. the ultrasound transducer).

As the signal provided by the transducer can be too small for being digitalized with the devices previously described, and it contains not only the ultrasonic part but also the audible sound spectrum, an Analog-Front-End (AFE), Figure 2.4, has been designed during the first stages of this research. It performs a bandpass filter, with X dB of amplification in the range from 20 kHz to 100 kHz, rejecting the signal out of this frequency. It

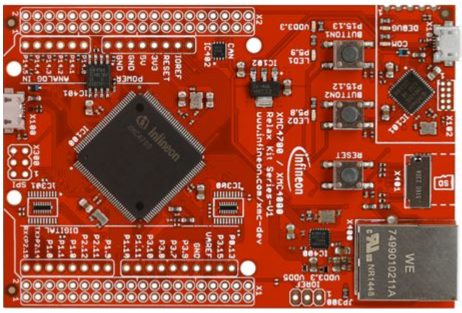


Figure 2.3: XMC4700 development board [6]

has 4 channels, and the layout has been designed to fit the XMC4700 Relax Kit layout, being possible to interconnect them like many other commercial shields.

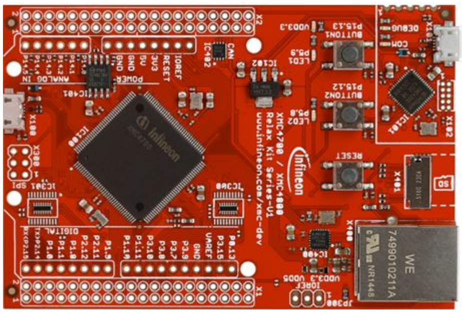


Figure 2.4: Analog-Front-End board

It integrates also DC/DC converter capabilities, for up to 4 channels, allowing the output signal voltage to increase by using an external DC source. The simulations have been made with Pspice and the PCB design with Eagle.

2.3. Signal processing

The signal acquired by the devices described in the previous section is the raw ultrasound signal. This signal needs to be further processed to extract information about gestures.For its processing, Matlab (version 2017b, 2019b, and 2021b) and Python (version 3) have been the tools selected in this research. Matlab eases the algorithms research and development, allowing quick implementations based on the libraries offered by Mathworks, and easily translated to other programming languages if needed. Particularly, this environment has been use on the ToF calculation, comparing methods (i.e. Threshold or Cross-correlation), and allowing the particularization for ultrasonic signals. It has been used as well on the filtering processing, denoising the acquired signal and removing possible outliers. Python has been used for the information extraction, developing algorithms for recognizing the gesture performed by the user based on Machine learning models. It has been used also for calculating the position of a target based on the previously calculated ToF values, using algorithms such as Multidimensional Scale (MDS). All those algorithms and models are further detailed in the following chapters.

References

[1] Silense. https://silense.eu/. Accessed: 2021-11-31.

[2] Semulin. https://www.semulin.de/. Accessed: 2021-11-31.

[3] Marvel project. https://www.marvel-project.eu/. Accessed: 2021-11-31.

[4] Daniel Lagler, Sebastian Anzinger, Eugen Pfann, Alessandra Fusco, Christian Bretthauer, and Mario Huemer. A single ultrasonic transducer fast and robust short-range distance measurement method. In *2019 IEEE International Ultrasonics Symposium (IUS)*, pages 2533–2536. IEEE, 2019.

[5] Digilent. Analog discovery 2. https://digilent.com/reference/test-and-measurement/analog-discovery-2/. Accessed: 2021-11-31.

[6] Infineon Technologies AG. Xmc4700. https://www.infineon.com/cms/en/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-cortex-m/32-bit-xmc4000-industrial-microcontroller-arm-cortex-m4/xmc4700/. Accessed: 2021-11-31.

Part II

Publications

Chapter 3

Gesture Recognition with Ultrasounds and Edge Computing

BORJA SAEZ[1], JAVIER MENDEZ[1], MIGUEL MOLINA[1], ENCARNACION CASTILLO[2], MANUEL PEGALAJAR[3], and DIEGO P. MORALES[2].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany

2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

3. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

IEEE Access

- Received January 2021, Accepted March 2021, Published March 2021
- DOI: 10.1109/ACCESS.2021.3064390
- Impact factor: 3.367
- JCR Rank: 94/273 in category Engineering, Electrical Electronic (Q2)

Gesture Recognition with Ultrasounds and Edge Computing

Abstract: The aim of this work is to prove that it is possible to develop a system able to detect gestures based only on ultrasonic signals and Edge devices. A set of 7 gestures plus idle has been defined, being possible to combine them to increase the recognized gestures. In order to recognize them, Ultrasound transceivers will be used to detect the 2 dimensional gestures. The Edge device approach implies that the whole data is processed in the device at the network edge rather than depending on external devices or services such as Cloud Computing. The system presented in this paper has been proven to be able to measure Time of Flight (ToF) signals that can be used to recognize multiple gestures by the integration of two transceivers, with an accuracy between 84.18 % and 98.4 %. Due to the optimization of the preprocessing correlation technique to extract the ToF from the echo signals and our specific firmware design to enable the parallelization of concurrent processes, the system can be implemented as an Edge Device.

Keywords: Edge computing, Gesture recognition, Human System Interaction (HSI), Ultrasound.

3.1. Introduction

The communication among humans is based on a multi-modal system, which includes not only verbal communication but also face and body expressions to intensify the meaning of the verbal content. The Human System Interaction (HSI) trend is evolving, leading to the research of emerging technologies that mimic this natural communication, minimizing the use of interfaces like touchscreens, buttons or sliders. Well known virtual personal assistants such as Alexa or Siri, developed by Amazon and Apple respectively which allow communication with the system using only voice commands. There are also several systems that introduce gesture control to the system, i.e. SoundWave [1], AudioGest [2], Dolphin [3], or UltraGesture [4]. All of them use low frequency ultrasound signals to recognize between 5 and 12 gestures, which are mostly based on Doppler shift effect (frequency variation due to movement) while running the recognition algorithms on PC or Smartphones.

The aim of this work is to prove the possibility to develop a system able to detect gestures based only on ultrasonic signals and to execute the signal processing in Edge devices, without using neither a PC nor a cloud environment. For testing, a set of 7 gestures plus idle has been defined, being possible to combine them to increase the recognized gestures. In order to recognize them, 2 transceivers will be used, since it is the minimum number of transceivers required to detect 2 dimensional gestures.

This device works as an active sonar system: it transmits ultrasonic waveforms, which are reflected back when they collide with any solid obstacle, to its environment. Then the transceivers receive these indirect echo signals in order to locate the echo produced by the obstacle. The transceivers are located on the same device. Thanks to this, it does not need an external synchronization signal to get the time-of-flight (ToF) value , which is the time between the transmitted signal emission and the echo signal reception. These measurements enable the system to have a great resolution in the depth dimension due to the direct relation between time-of-flight and the distance between the reflector object and the system. This is an advantage over 2D cameras or Electric Near Field sensors, which are more sensitive to noise and need to infer the distance from the strength of the received signals. However, it has low positioning accuracy when it comes to the lateral range. In spite of higher processing time, it could be solved by adding more devices to the system, getting a combination of time-of-flights estimations between them.

This article is structured as follows: Section 3.2 introduces the state of the art in Ultrasound technologies for gesture recognition and the advantages of use Edge Computing for this purpose. Section 5.2 explains in detail the system developed in this work, as well as the firmware developed for the signal acquisition and ToF calculation. Section 3.4 describes the gestures defined for the experiment and the algorithms studied for the recognition and classification. Section 5.5 summarizes the results obtained. Finally, Section 5.6 focuses on conclusions of this work.

3.2. Prior work / State of the art

3.2.1. Ultrasounds

Originally, ultrasound technology started to be used to increase the perception under the sea for navigation purposes, known as sonar devices [5]. However, ultrasounds were soon applied to medicine [6] and quickly found in more application fields, such as non-destructive testing methods [7].

Nowadays, ultrasounds are used for object recognition [8], which aim to reduce the power consumption, computation, and cost of current optical sensors. In [9], DasIvan et al. created an ultrasonic-based hand-gesture recognition device using a single piezoelectric transducer and an 8-element microphone array. Despite the fact that the accuracy was lower than in devices using optical sensors, it increased the number of gestures supported by a factor of 200 within the same energy budget. The developed system uses the Sound-Source Localization (SSL) algorithm.

However, other approaches have tried different techniques with the same goal. UltraGesture [4] uses the Channel Impulse Response (CIR) for finger motion perception and recognition, getting a resolution of 7 mm in the measurements. Soundwave [1], AudioGest [2], and Dolphin [3] measure the frequency variation of the hand in the incoming signal due to the movement of the user, known as Doppler effect. All three works use commercial speakers and microphones embedded in existing systems.

The difference among the previously commented systems are the developed algorithms for the gesture recognition. SoundWave [1] implements a threshold-based dynamic peak tracking technique to capture the Doppler shifts recorded by a laptop. Similarly, AudioGest [2] adds some of the signal contexts to the estimation of the hand in-air time, average waving speed as well as hand moving range. Smart mobile devices have also been used for a closer interaction with the user, using the same Doppler shift technique as the previous papers [3]. A further comparison of these studies will be shown in Section 5.5.

Apart from large-scale gestures as studied in our paper, ultrasound signals have also been used for multiple gesture types. An example of this is the classification of micro-gestures based on the micro-Doppler effect. Sang Y. et al. [10] and Zeng Q. et al. [11] proposed two different models for this purpose. The data to classify in these papers are seven and five finger-based gestures respectively. Both models are based on Recurrent Neuronal Networks (RNN) and Convolutional Neuronal Networks (CNN) to study the temporal evolution of the micro-Doppler images, achieving an accuracy over 90 % in both cases.

One of the reasons for the integration of ultrasound sensors when using these techniques rather than other technologies is its robust behaviour against the ambient light or visibility changes. At the same time, while cameras or microphones can easily differentiate not only the gestures or voice commands, but also who is doing it, they may incur privacy concerns. Ultrasounds only get relevant information of the movement and, consequently, capture fewer attributes from the users, which hardens user tracking and identification but improves the privacy of the user.

One of the goals of the proposed system in this paper is to integrate it into different multi-purpose large systems. Therefore, in order to reduce the complexity of the integration of the ultrasound module, an Edge approach has been researched. This implies that the whole data is preprocessed in the device at the network edge instead of depending on external devices or services such as Cloud Computing. At the same time, this approach would increase privacy since the raw data is not transmitted but only the final processed gesture classification is. The next subsection gives details about the advantages of this approach as well as a deeper description of Edge Computing.

3.2.2. Edge Computing

Edge Computing [12] is aimed at reducing Cloud workload to process device data. To do so, some preprocessing and/or computing tasks are executed at the network edge when possible. Thus, Edge Computing is suitable in scenarios where low latency is required for the user, or where the end device application has time critical constraints [13].

At the same time, this technique ensures integrity and confidentiality of the information [14]. As a result of not communicating the information with external devices, the energy consumption for the data transmission is reduced [15]. By preprocessing the data in the device, the confidential information which is not relevant for the final task can be masked/deleted before being shared with an external device. This process also can be used to standardize the format of the transmitted data in order to create a shared format that all the devices can understand even if initially the format of each device was different [16]. This is especially relevant when multiple devices are collaborating as it is in the Internet of Things environment.

3.3. Hardware description and signal acquisition

The proposed system uses two modules, as shown in Figure 5.1. The first one is used to control two transducers to generate the outgoing signal and acquire the incoming echo. This module also calculates the time elapsed between the emission and the reception of the signal for each transceiver. This time is known as Time of Flight (ToF). The first module also integrates the analog circuitry needed for the echo signals amplification. The second module receives the ToF values and, after filtering them, performs the recognition algorithm to determine the gesture realized by the user. If needed, this module can integrate an external Neuroshield board, to perform the recognition algorithm, and control an external device (such as a led strip) to display the detected gesture.

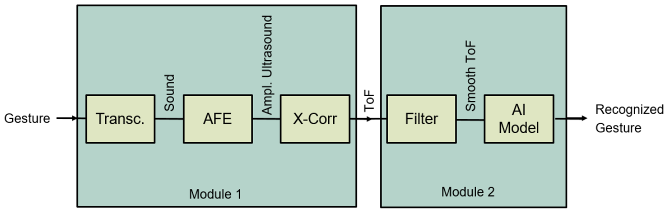


Figure 3.1: System diagram

Both modules are composed of a XMC4700 microcontroller performing the acquisition/recognition task, as well as a Bluetooth HC-05 device for the communication between them. This communication technology has been chosen to add a wireless channel between both modules to have flexibility on how to place them, but other technologies can be used as well.

The ultrasound transducers used in this work are based on a dual-backplate MEMS microphone technology allowing a combined use as an airborne ultrasonic transceiver and audio microphone. Those transducers need a low bias voltage and offer an audio performance of 68 dB(A) signal-to-noise ratio (SNR) and between 80 and 90 dB SNR in the ultrasonic frequency range. After the emission of the pulses, a free oscillation of the membrane (ringing) can override the incoming echo, producing a shadow zone that allows obstacle detection from 10 cm on [17].

3.3.1. Signal emission and reception

The signal emission and reception are performed by the module 1, whose block diagram is shown in Figure 3.2. The signal to transmit is a square signal generated by a Pulse Width Modulation (PWM) block integrated into the processor. This signal is later transformed into an acoustic wave by one of the transducers. As soon as the PWM block finishes the pulse generation, the microcontroller starts collecting samples using two Analog Digital Converter (ADC) in parallel, one for each transceiver, to minimize time skew between samples. The echo received by the transducer, as an analog signal, carry some noise from the environment (as could be the use of buttons from a computer's keyboard or mouse, that have been seen to be harmful to the device's operating frequency). A band-pass amplifier was developed for this task, which amplifies the lower ultrasonic band (20 kHz to 100 kHz) while filters out all other frequencies. After this filter, the signal must be digitalized by the microcontroller ADC module for further processing, as it is explained in the next subsection.

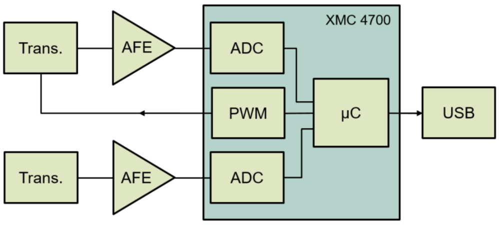


Figure 3.2: Transducer control and ToF calculation

3.3.2. Time Of Flight

After the signal is acquired it has to be processed to identify if there is an incoming echo, and the position of this if applicable. The ToF calculus has to be done while the following frame is being acquired, running both processes in parallel as shown in Figure 3.3.

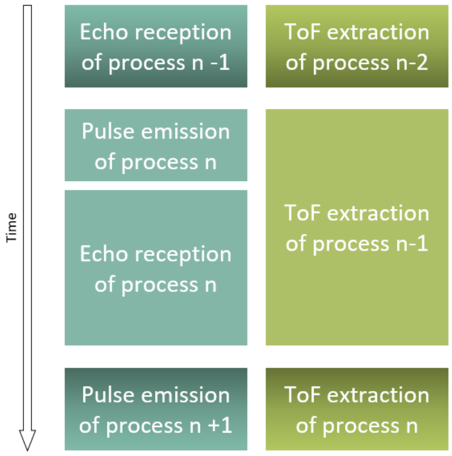

Time
Echo reception of process n -1
ToF extraction of process n-2
Pulse emission of process n
ToF extraction of process n-1
Echo reception of process n
Pulse emission of process n +1
ToF extraction of process n

Figure 3.3: Firmware task parallelization to minimize execution time.

The signal can be processed in different domains to calculate the ToF, finding in the literature several methods for each domain, as collected by J. C. Jackson et al. [18]. and summarized in table 3.1.

Domain	Methods
Time	- Threshold Detection - Cross-Correlation
Fourier Phase-Based	- Single-Frequency Signals - Chirps and the Cross-Spectrum
Hybrid Models	- Biologically Inspired

Table 3.1: Example of ToF calculus techniques.

Some methods try to imitate the nature systems to calculate the ToF. for example, D. Hayward et al. [19] developed the "Biologically Inspired Ranging Algorithm (BIRA)"based on the bats hearing system for echolocation.

Other models are based in the frequency domain, as for example K.-N. Huang et al. [20] use the phased difference of a single frequency signal to calculate the ToF. Also, signal with more than one frequency component has been studied to calculate the desired parameter, as for example D. M. Cowell et al. [21] used chirp-signals to increase the accuracy of the estimated ToF.

This approach also avoids multi-path problems and differentiates between several emitters.

Due to the low computation power required and good results, most works base this calculus on time domain methods, based i.e. on the amplitude of the incoming signal or in the cross-correlation of the echo with the sent (or expected) signal. The cross-correlation method reduces the high influence of noise in the amplitude method, since the cross-correlation, which acts as matched filtering, produces a time-domain signal with a maximum at the time when the echo was received [18] [22].

The ToF calculation proposed in this system can be divided in four steps as described in Figure 3.4. First, the acquired signal is cross-correlated with the template of the expecting echo, giving a maximum value where the expected and real echo overlap. Then, the envelope of the previous signal is obtained using a low pass filter. After that, the envelope is evaluated to extract the first cut with a dynamic threshold. This threshold represents the attenuation of the signal due to the distance traveled. It can be adjusted according to the ambient noise level of each specific scenario. Finally, the maximum of the cross-correlated signal is searched on a window with center in the threshold-envelope crossing value, giving the position of the ToF in number of samples. Once the number of ToF samples is determined, it can be easily converted to time knowing the ADC sampling frequency.

Using only one transceiver as emitter brings a non-desire effect in the ToF calculus. The distance from the obstacle to the transmitter is a direct relation with the ToF estimated, as shown in (1), but the ToF estimated in the signal of the second transceiver is a relation of the distance between the obstacle with both transceivers, as shown in (2). The solution to this effect will be further discussed in following sections.

$$ToF_1 = \frac{2d_1}{c_s} \tag{3.1}$$

$$ToF_2 = \frac{d_1 + d_2}{c_s} \tag{3.2}$$

Where ToF_n indicate the ToF for the transceiver n, d_n the distance between the target and the transceiver n and c_s the speed of the sound.

The proposed system is robust also to temperature changes. The speed of the sound in the air depends, among other environmental effects, on the air temperature [23]. This dependency is significant enough to allow the estimation of air temperature based on the difference between ToF measurement as shown by P.Annibale et al. [24]. Once more, the use of the relation between ToF of both transceivers provides the mitigation of this non desire effect.

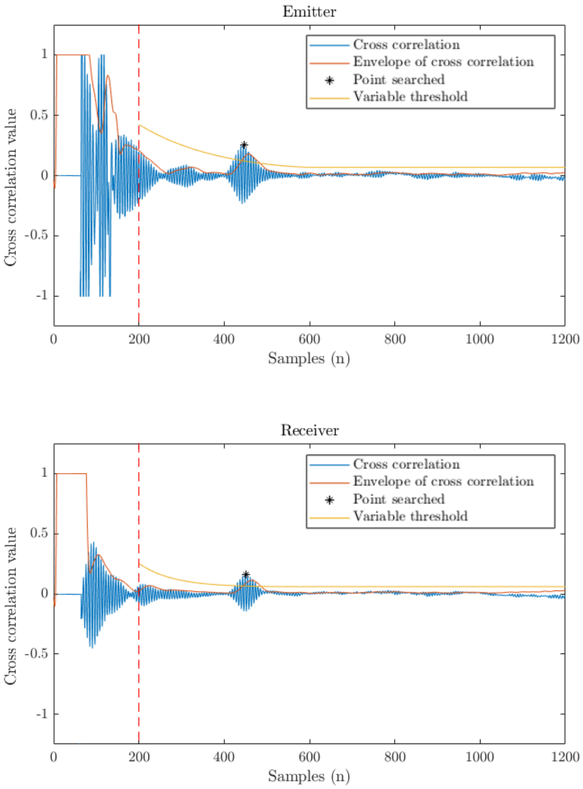


Figure 3.4: ToF calculus algorithm: cross-correlation signal (blue), cross-correlation signal envelope (red), threshold for echo detection (yellow) and peak value detected (black).

3.4. Gesture recognition methods

Seven gestures, and idle, have been selected for this experiment: front push, front pull, right push, right pull, left push, left pull, static position, and no gesture. These gestures are well defined arm or hand movements in two dimensions to minimize the gesture complexity and reduce to two the required transceivers. Therefore, all gestures must be contained in this plane and so they are assumed to be in the front part of the sensors as shown in Figure 3.5. Otherwise, the system won't be able to track the gesture, due to the transceiver's unidirectional sensitivity and radiation pattern. This is an effect of the package to protect the membranes and electronics, which is also used to increase the strength of the emitted signal.

These gestures are measured using both transceivers simultaneously. By extracting the ToF from each sensor in each moment, as explained in 3.4.1

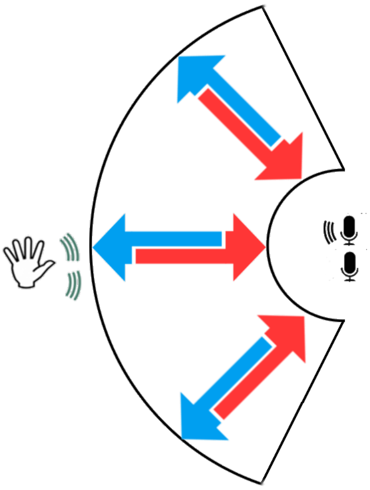


Figure 3.5: Gestures diagram: push(red) and pull(blue) direction in the three different regions (Top view).

by (1) and (2), it is possible to determine the movement direction and the region of the plane where the movement has been done.

Four individuals performed these gestures in different conditions within a distance of 15-50 cm from the device to collect data from different conditions. Each individual repeated each gesture 4 times per session during 20 sessions. These gestures have a variable length depending on the subject and the specific time, which helps to create a more diverse dataset. The average time length of these gestures was approximately 3 seconds after a review of average length on hand gestures. The frequency used for recording the ToF samples was 30 Hz. Nevertheless, the time length of the whole gesture is not a critical factor, since each gesture is classified multiple times during its performance. Therefore, even if a gesture is short, as far as it lasts for the required 7 ToF samples (250ms), it will be correctly classified. However, the speed of the gesture may affect on a larger scale since a lower hand speed will result in a smaller variation of the ToF. If this happens, the system may classify this gesture as idle due to its low variance of the position.

The final data-set created contains 3150 gesture samples where each gesture sample consists of a number ToF samples from each transceiver as shown in Figure 3.6. The specific number of ToF samples will be commented in Section 3.4.2. Out of all the gestures samples, 80 % were used during the training process and the remaining 20 % were used for testing the final system.

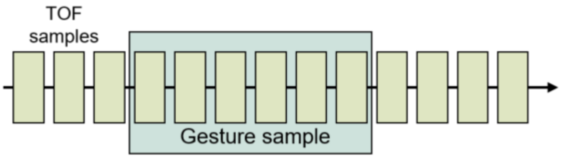


Figure 3.6: Gesture sample creation

3.4.1. Filtering the Raw ToF Data

After preprocessing the raw ToF data extracted from the transceivers, the data needs to be filtered in order to remove outlier points as well as reconstruct the ToF signal when possible.

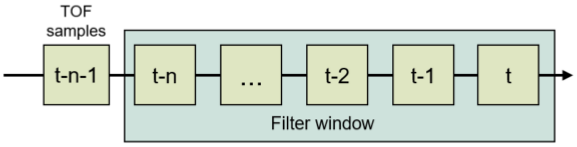


Figure 3.7: Filter window

While multiple filtering techniques may be applied in this scenario, the speed of the system when applying the filtering technique has to be taken as a constrain in order to avoid creating a bottleneck at this point. Therefore, a filtering technique where the ToF data is compared with the n previous ToF samples has been designed resulting in a smooth filter specific for this application. This filter has been designed to take into account the most frequent and relevant problems detected in the raw signal, such as missing information or measurements when the sensor is saturated. As a result of this, it is more suitable than a general purpose smooth filter.

The window approach used with the filtering technique described is shown in figure 3.7. The goal of this filter is to remove outlier points and recover lost ToF samples. The dimension of the window of data that will be used with this filter has been researched to determine the optimal size. The compared parameters for these filters are the execution time as well as the noise reduction. Table 3.2 shows all the compared dimensions.

The final size of the window is 11 ToF samples. This decision was based on the trade-off between the noise reduction and the execution time. Larger window filters lead to latency problems since its execution and the later classification would exceed the time limit of 33 us. At the same time, these filters only provide, as maximum, a 0.97 % improvement respect the chosen filter regarding noise reduction. The effect of applying this filter in the ToF data can be observed in figure 3.8.

This preprocessing has proven to increase the accuracy of the gesture

Window size	Execution time (us)	Noise reduction
8	7.42	74.86 %
10	9.87	84.70 %
11	10.92	85.30 %
15	14.1	85.92 %
20	18.3	86.27 %

Table 3.2: Comparison of multiple sizes for the window of the filter technique.

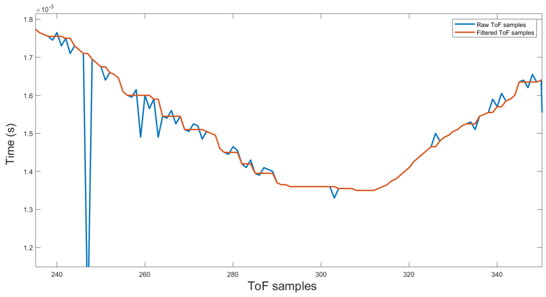


Figure 3.8: ToF data before and after applying filter.

classification, as shown in Section 5.5, where this fact will be further explained.

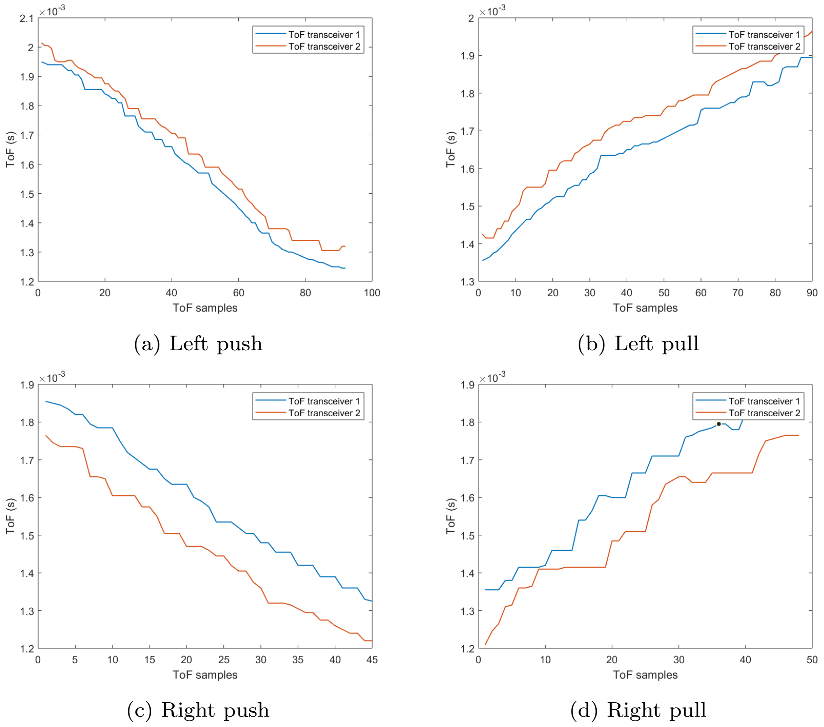
The filtered ToF samples of some of the studied gestures using the previous filtering technique are shown in figure 3.9 for a deeper understanding of the data used in this paper.

Besides the remaining noise in the signal after the filtering process, it is possible to obtain high classification accuracy thanks to the researched algorithms. During the training process, at the same time the AI models learn to classify the input data, they learn as well to adapt themselves to the noise of the signals. Further explanations of these algorithms are done in Section 3.4.2.

3.4.2. Algorithms

Multiple classification algorithms were applied to the gathered data aiming to compare the gesture recognition accuracy based on the collected data explained in the previous sections. The data used for the classification has been explained in Subsection 3.4, where Figure 3.6 shows how each gesture sample is created as a succession of ToF samples from both transceivers. This enables the system to learn the time evolution of the signal without using complex algorithms such as LSTM neural networks.

Each time a new ToF sample is received, the window slides creating a new gesture sample including the new ToF sample and removing the oldest one.



(a) Left push

(b) Left pull

(c) Right push

(d) Right pull

Figure 3.9: Filtered ToF samples of left and right push and pull gestures.

The sliding window enables the system to generate more gesture samples for the learning phase than dividing the whole data-set into sub-datasets of n ToF samples.

Since the algorithms used for the classification are based on a supervised learning approach, the ToF data does not have to be preprocessed to obtain the real distances with respect to each transceiver. At the same time, the algorithms learn to overcome the possible remaining noise in the data after the first filter explained in Subsection 3.4.1.

Finally, from each gesture sample, the slope of the gesture sample from each transceiver as well as the difference between their mean values were used as input features for the classification algorithms.

The relevant information of the gesture data for its classification is the evolution of the value of the ToF signals. Therefore, a study to decide the number of ToF samples contained in each gesture sample was carried out. As the gesture data will be preprocessed to extract the previously explained features, the number of inputs for the algorithms is independent from the number of ToF samples per gesture sample. The comparison was based on the final accuracy achieved in Multilayer Perceptron (MLP) [25] that will be commented in this section, as shown in Table 3.3.

As a result of this study, the number of ToF samples per gesture sample

Number of ToF samples	Final accuracy
4	84.78 %
6	92.63 %
7	92.87 %
8	92.87 %
10	90.15 %
12	90.12 %

Table 3.3: Comparison of multiple number of ToF samples per gesture sample.

was set to 7. The reason for this decision is its high accuracy in the MLP model as well as its reduced number of samples. The latest reason leads to an increase of the number of gestures samples created. This is beneficial during the training phase of the models. Its higher accuracy in comparison with the cases of a higher number of ToF samples is due to the fact this increase leads to problems during transitions between gestures.

Three algorithms have been researched in this paper:

- **Deep Learning model.** Different structures of Deep Neural Networks (DNN) were researched, such as MLP [25], Long Short Term memory (LST) DNN [26] and Convolutional Neural Network [27]. Since the features used for the classification do not require a time evolution study or a further feature extraction, we concluded the MLP was the structure that fits in this application among the DNN structure researched. This decision was based on the time required to re-train the DNN in case new gestures are added to the system as well as its speed to compute the result. In case any of the other DNN structure were implemented, the latency of the system would increase leading to bottleneck problems in the classification step of the pipeline.

  The proposed MLP model was designed keeping in mind the number of layers as well as artificial neurons while achieving high accuracy results. The chosen structure is an MLP of 4 layers as Figure 3.10 shows. The input layer includes 3 artificial neurons, which represent the number of features that will be fed into this DNN. Following the input layer, there are two hidden layers with 6 and 9 neurons respectively. The output layer contains 8 neurons to match the number of gestures (including idle) studied in this paper. In the structure, batch normalization layers have been added between each layer to increase the stability of the DNN.

  As a result, this model could be implemented in an Edge Device for the inference process due to its low memory requirements as well as the speed to process the input data.

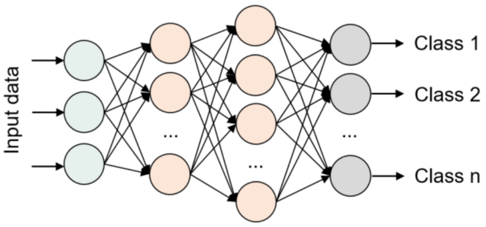


Figure 3.10: MLP network structure.

- **Deep Learning model based on Neuroshield device.** Another approach researched in this paper was the implementation of the classification task in the Neuroshield device [28]. This device includes 576 artificial neurons programmed with a radial basis activation function [29] rather than the previously commented DNN. This activation function computes the distance, in the feature representation plane, of the established center of each neuron with the input data as shown in Figure 3.4.2. After calculating all the distances, it calculates which neuron is the closest to the input data and, in case the distance is smaller than the activation distance, the input activates the corresponding artificial neuron.

  This optimized algorithm, apart from moving the inference stage to the network edge due to its reduced latency, enables the execution of the training of the AI model at the network edge. The limitations of this model fall on the fact the DNN designed for this device must be trained using the same technique, radial basis activation function.

  [h!][width=0.55]Imagenes/Grusec/neuroshieldDNN.PNG Neuroshield activation function structure.

- **Decision Tree model.** This model is based on a set of rules which are defined during the training stage in order to classify the gesture by comparing the input data with a list of conditional clauses where the data is divided into different decisions according to a certain parameter [30] leading to a final decision based on the results of these conditional clauses. This model is less computing-power demanding due to its simplicity to classify a new data sample. At the same time, this simplicity makes it difficult to maintain its accuracy when the complexity of the data increases.

The features fed into the classification techniques were the same: the slope of the ToF signal measured from the first transceiver and the average value of the last seven ToF samples as well as the difference of the mean values of the ToF signals measured with both transceivers. The same postprocessing technique has been applied to all the previous algorithms in order to further improve their accuracy while still being able to compare them. The postprocessing technique applied is a sliding window to extract the most frequent classification results in the last 5 classification results. Therefore, outlier classification results are filtered, maintaining a slow and continuous change between gestures. The improvement of the accuracy when applying this technique can be observed in Section 5.5.

3.5. Results

The results obtained with the previously explained techniques are presented in this section using the same data to ensure a correct comparison of the algorithms.

Due to the fact that all these techniques accomplish with the time restriction of the system, the compared parameter in this section is the accuracy, which is measured in this experiment as correct classifications over all the classifications.

The Table 3.4 shows the accuracy achieved using each classification approach. At the same time, this table compares the accuracy results obtained when using the raw signal (first column), the filtered ToF data (second column), and using all the previously explained preprocessing techniques as well as the window to filter the output classification results.

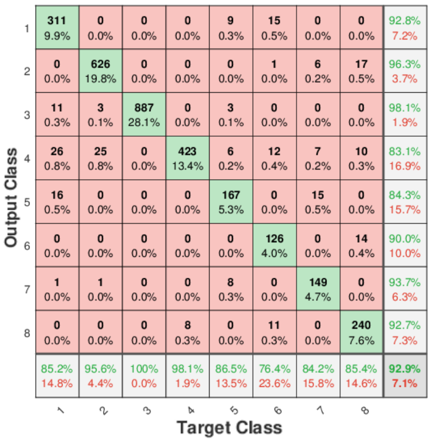
Classification technique	Acc. 1	Acc. 2	Acc. 3
MLP	84.18 %	91.16 %	92.87 %
Neuroshield	95.69 %	97.75 %	98.4 %
Decision Tree	91.8 %	92.15 %	96.94 %

Table 3.4: Accuracy results without any filter or window (acc. 1), without the window (acc. 2) and using all the filtering techniques (acc. 3).

The results obtained with the Neuroshield device achieved the highest accuracy among the researched techniques, both scenarios of not applying or applying the postprocessing technique. However, this system lacks the flexibility the other two techniques can provide due to the fact that this device can only execute one kind of DNN and it can not be transferred to another device different from a Neuroshield device.

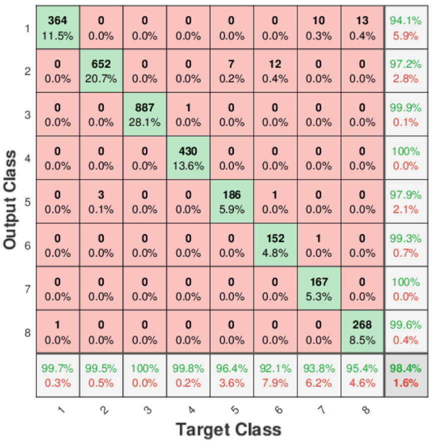
The Decision Tree algorithm achieved a final accuracy of 5.6 % and 1.46 % lower than the Neuroshield device, without the postprocessing and including it respectively. Nevertheless, this technique is the less power requiring due to its simplicity in comparison with the DNN structures presented in the paper.

The MLP classificator achieved a final accuracy of 6.59 % and 5.53 % lower than the Neuroshield devices, without the postprocessing and including it respectively. In spite of achieving the lowest accuracy among these techniques, this one provides the highest flexibility since the structure of the DNN and the activation function can be modified easily as well as transferred to other devices.


Output Class
Target Class

	1	2	3	4	5	6	7	8	
1	311 9.9%	0 0.0%	0 0.0%	0 0.0%	9 0.3%	15 0.5%	0 0.0%	0 0.0%	92.8% 7.2%
2	0 0.0%	626 19.8%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	6 0.2%	17 0.5%	96.3% 3.7%
3	11 0.3%	3 0.1%	887 28.1%	0 0.0%	3 0.1%	0 0.0%	0 0.0%	0 0.0%	98.1% 1.9%
4	26 0.8%	25 0.8%	0 0.0%	423 13.4%	6 0.2%	12 0.4%	7 0.2%	10 0.3%	83.1% 16.9%
5	16 0.5%	0 0.0%	0 0.0%	0 0.0%	167 5.3%	0 0.0%	15 0.5%	0 0.0%	84.3% 15.7%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	126 4.0%	0 0.0%	14 0.4%	90.0% 10.0%
7	1 0.0%	1 0.0%	0 0.0%	0 0.0%	8 0.3%	0 0.0%	149 4.7%	0 0.0%	93.7% 6.3%
8	0 0.0%	0 0.0%	0 0.0%	8 0.3%	0 0.0%	11 0.3%	0 0.0%	240 7.6%	92.7% 7.3%
	85.2% 14.8%	95.6% 4.4%	100% 0.0%	98.1% 1.9%	86.5% 13.5%	76.4% 23.6%	84.2% 15.8%	85.4% 14.6%	92.9% 7.1%

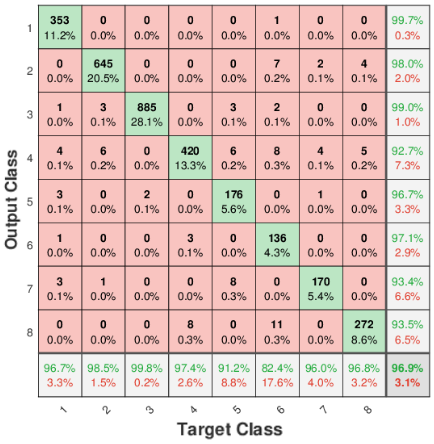
Figure 3.11: MLP confusion matrix.


Output Class
Target Class

	1	2	3	4	5	6	7	8	
1	364 11.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 0.3%	13 0.4%	94.1% 5.9%
2	0 0.0%	652 20.7%	0 0.0%	0 0.0%	7 0.2%	12 0.4%	0 0.0%	0 0.0%	97.2% 2.8%
3	0 0.0%	0 0.0%	887 28.1%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.9% 0.1%
4	0 0.0%	0 0.0%	0 0.0%	430 13.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	3 0.1%	0 0.0%	0 0.0%	186 5.9%	1 0.0%	0 0.0%	0 0.0%	97.9% 2.1%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	152 4.8%	1 0.0%	0 0.0%	99.3% 0.7%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	167 5.3%	0 0.0%	100% 0.0%
8	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	268 8.5%	99.6% 0.4%
	99.7% 0.3%	99.5% 0.5%	100% 0.0%	99.8% 0.2%	96.4% 3.6%	92.1% 7.9%	93.8% 6.2%	95.4% 4.6%	98.4% 1.6%

Figure 3.12: Neuroshield algorithm confusion matrix.

For a deeper comparison of the accuracy achieved for each gesture, Figures 3.11, 3.12 and 3.13 show the confusion matrix of the final algorithms (including all the filtering techniques). It is possible to observe how all the researched algorithms achieve high accuracy for all the gestures, being the lowest one the accuracy achieved for the gesture 5 (left push), 83.1 %, when



	1	2	3	4	5	6	7	8	
1	353 11.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
2	0 0.0%	645 20.5%	0 0.0%	0 0.0%	0 0.0%	7 0.2%	2 0.1%	4 0.1%	98.0% 2.0%
3	1 0.0%	3 0.1%	885 28.1%	0 0.0%	3 0.1%	2 0.1%	0 0.0%	0 0.0%	99.0% 1.0%
4	4 0.1%	6 0.2%	0 0.0%	420 13.3%	6 0.2%	8 0.3%	4 0.1%	5 0.2%	92.7% 7.3%
5	3 0.1%	0 0.0%	2 0.1%	0 0.0%	176 5.6%	0 0.0%	1 0.0%	0 0.0%	96.7% 3.3%
6	1 0.0%	0 0.0%	0 0.0%	3 0.1%	0 0.0%	136 4.3%	0 0.0%	0 0.0%	97.1% 2.9%
7	3 0.1%	1 0.0%	0 0.0%	0 0.0%	8 0.3%	0 0.0%	170 5.4%	0 0.0%	93.4% 6.6%
8	0 0.0%	0 0.0%	0 0.0%	8 0.3%	0 0.0%	11 0.3%	0 0.0%	272 8.6%	93.5% 6.5%
	96.7% 3.3%	98.5% 1.5%	99.8% 0.2%	97.4% 2.6%	91.2% 8.8%	82.4% 17.6%	96.0% 4.0%	96.8% 3.2%	96.9% 3.1%

Figure 3.13: Decision tree confusion matrix.

using the MLP algorithm. Therefore, we can conclude all these models can generalize the data properly. As previously commented, these tables also show how the MLP model achieves the lowest accuracy results for all the gestures among the researched algorithms. The main difference we can observe from these confusion matrices is the error distribution. While the errors in the MLP and decision tree models are distributed across all the gestures, the errors of the Neuroshield model are concentrated in the last 4 gestures.

Another relevant factor to compare among the researched algorithms is the memory consumption of the different models since this is one of the restrictive parameters in Edge Devices. Table 3.5 shows this comparison, where it is possible to observe how the MLP model, even when its accuracy is approximately 5 % lower than the best model of the Neuroshield device, leads to a memory consumption reduction for the model of an 83.1 %.

Classification technique	Model size
MLP	23KB
Neuroshield	136KB
Decision Tree	273 KB

Table 3.5: Comparison of the size of the researched algorithms.

The latency of these models has not been compared since all of them satisfied the restriction of the 33ms established by the hardware providing a classification result for any new data before receiving the next one.

A comparison of the studies described in Section 3.2 is presented in Table 3.6. Even though it is not possible to compare the performance of the algorithms due to the lack of a common public dataset as well as the diffe-

Studies	No. Gestures	Accuracy	Method	Hardware
SoundWave	5	86.7 – 100 %	Doppler shift	1 microphone 1 speaker
AudioGest	6	95.1 %	Doppler shift	1 microphone 1 speaker
Dolphin	24	93 %	Doppler shift	1 microphone 1 speaker Gravity sensor
UltraGesture	12	91.4 – 98.6 %	Channel Impulse Response (CIR)	4 microphones 1 speaker
Microsoft	5	64.5 – 96.9 %	CNN-LSTM	8 microphones 1 transceiver
Proposed system	8	84.2 – 98.4 %	AI models	2 transceiver

Table 3.6: Comparison of state-of-the-art techniques for gesture recognition with ultrasounds.

rence in the data structure each technique requires, significant parameters of each system can be compared. The future development of gesture recognition systems based on ultrasound technology could benefit from a common data framework, thus allowing the cooperative development of algorithms with much more data and from different sources and conditions.

One of the features that we can compare is the devices integrated into these systems. It is possible to observe how the majority of the researchers are basing the systems on a multi-sensor approach where a separated microphone and speaker are integrated. On the other hand, our proposed system tries to reduce the number of devices integrating transceivers.

3.6. Conclusion

The system presented in this paper has been proven to be able to measure ToF signals that can be later used to recognize multiple gestures by the integration of two transceivers. Due to the optimization of the preprocessing correlation technique to extract the ToF from the echo signals and the specific design of the firmware to enable the parallelization of concurrent processes, the system can be implemented as an Edge Device. This system does not require any external device or cloud server to preprocess the information.

At the same time, by using the Neuroshield device, which enables the implementation of an AI classificator at the network edge, or the MLP implemented in an Edge Device, it is also possible to execute the full process from data gathering to extract the classification at the network edge while maintaining high accuracy results. It has been shown how the researched algorithms provided high accuracy, where the best result is extracted from the Neuroshield with a 98.4 % accuracy.

The memory sizes of the models are also a relevant feature to compare since it is one of the main constrains in Edge Devices. Because of this, this feature has been taken into account during the optimization of the models. As a result of this, the size of all the proposed models has been reduced, i.e. the proposed MLP, whose size is 23 KB while it stills achieves an accuracy of 92.87 % in our dataset.

References

[1] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: using the doppler effect to sense gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1911–1914, 2012.

[2] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pages 474–485, 2016.

[3] Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin, and Zhang Sanfeng. Dolphin: Ultrasonic-based gesture recognition on smartphone platform. *2014 IEEE 17th International Conference on Computational Science and Engineering*, pages 1461–1468, 2014.

[4] Kang Ling, Haipeng Dai, Yuntang Liu, and Alex X Liu. Ultragesture: Fine-grained gesture sensing and recognition. *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2018.

[5] Jacques Lewiner. Paul langevin and the birth of ultrasonics. *Japanese Journal of Applied Physics*, 30(S1):5, jan 1991.

[6] Aladin Carovac, Fahrudin Smajlovic, and Dzelaludin Junuzovic. Application of ultrasound in medicine. *Acta Informatica Medica*, 19(3):168, 2011.

[7] Maurice G Silk. *Ultrasonic transducers for nondestructive testing*. Adam Hilger Ltd., Accord, MA, 1984.

[8] Y. Gao, M. A. Maraci, and J. A. Noble. Describing ultrasound video content using deep convolutional neural networks. *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 787–790, 2016.

[9] Amit Das, Ivan Tashev, and Shoaib Mohammed. Ultrasound based gesture recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 406–410, 2017.

[10] Yu Sang, Laixi Shi, and Yimin Liu. Micro hand gesture recognition system using ultrasonic active sensing. *IEEE Access*, 6:49339–49347, 2018.

[11] Qinglin Zeng, Zheng Kuang, Shuaibing Wu, and Jun Yang. A method of ultrasonic finger gesture recognition based on the micro-doppler effect. *Applied Sciences*, 9(11):2314, 2019.

[12] Dinesh Dash Partha Pratim Ray and Debashis. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. *Network and Computer Applications*, 140:1–22, 2019.

[13] M. Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[14] Sara Casado-Vara Roberto Sittón-Candanedo Inés Alonso Ricardo Corchado Rodríguez, Juan Rodríguez. A Review of Edge Computing Reference Architectures and a new Global Edge Proposal. *Future Generation Computer Systems*, 2019.

[15] J. Cao Q. Zhang Y. Li W. Shi and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[16] W. Yu et al. A Survey on the Edge Computing for the Internet of Things. *IEEE Access*, 6:6900–6919, 2018.

[17] Sebastian Anzinger, Christian Bretthauer, Johannes Manz, Ulrich Krumbein, and Alfons Dehé. Broadband acoustical mems transceivers for simultaneous range finding and microphone applications. *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, pages 865–868, 2019.

[18] J. C. Jackson, R. Summan, G. I. Dobie, S. M. Whiteley, S. G. Pierce, and G. Hayward. Time-of-flight measurement techniques for airborne ultrasonic ranging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 60(2):343–355, 2013.

[19] G Hayward, F Devaud, and JJ Soraghan. P1g-3 evaluation of a bio-inspired range finding algorithm (bira). *2006 IEEE Ultrasonics Symposium*, pages 1381–1384, 2006.

[20] Ke-Nung Huang and Yu-Pei Huang. Multiple-frequency ultrasonic distance measurement using direct digital frequency synthesizers. *Sensors and Actuators A: Physical*, 149(1):42–50, 2009.

[21] David MJ Cowell and Steven Freear. Separation of overlapping linear frequency modulated (lfm) signals using the fractional fourier transform. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 57(10):2324–2333, 2010.

[22] Jian Chen, Fan Yu, Jianxin Yu, and Lin Lin. A three-dimensional pen-like ultrasonic positioning system based on quasi-spherical pvdf ultrasonic transmitter. *IEEE Sensors Journal*, 2020.

[23] Dennis A Bohn. Environmental effects on the speed of sound. *Journal of the audio engineering society, Audio Engineering Society Convention 83*, 1987.

[24] Paolo Annibale, Jason Filos, Patrick A Naylor, and Rudolf Rabenstein. Tdoa-based speed of sound estimation for air temperature and room geometry inference. *IEEE transactions on audio, speech, and language processing*, 21(2):234–246, 2012.

[25] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. *Multilayer Perceptron: Architecture Optimization and Training with Mixed Activation Functions*. BDCA'17. Association for Computing Machinery, New York, NY, USA, 2017.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[27] T. A. Mohammed S. Albawi and S. Al-Zawi. Understanding of a convolutional neural network. *International Conference on Engineering and Technology (ICET), Antalya, 2017*, pages 1–6, 2017.

[28] General Vision. Neuroshield. (accessed: 20.04.2020).

[29] S. Elanayar V.T. and Y. C. Shin. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transactions on Neural Networks*, 5(4):594–603, 1994.

[30] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

Chapter 4

Object Positioning Algorithm Based on Multidimensional Scaling and Optimization for Synthetic Gesture Data Generation

Borja Saez-Mingorance [1,2,], Antonio Escobar-Molero [1], Javier Mendez-Gomez [1,2], Encarnacion Castillo-Morales [2] and Diego P. Morales-Santos [2,3].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany.
2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain.
3. RedNodeLabs UG, 80469 Munich, Germany.

MDPI Sensors

- Received July 2021, Accepted August 2021, Published September 2021
- DOI: 10.3390/s21175923
- Impact factor: 3.576
- JCR Rank: 14/64 in category Instrument and Instrumentation (Q1)

Object Positioning Algorithm Based on Multidimensional Scaling and Optimization for Synthetic Gesture Data Generation

Abstract: This work studies the feasibility of a novel two-step algorithm for infrastructure and object positioning using pairwise distances. The proposal is based on the optimization algorithms, Scaling-by-Majorizing-a-Complicated-Function and the Limited-Memory-Broyden-Fletcher-Goldfarb-Shannon. A qualitative evaluation of these algorithms is performed for 3D positioning. As the final stage, smoothing filtering techniques are applied to estimate the trajectory, from the previously obtained positions. This approach can also be used as a synthetic gesture data generator framework. This framework is independent from the hardware and can be used to simulate the estimation of trajectories, from noisy distances gathered with a large range of sensors by modifying the noise properties of the initial distances. The framework has been validated using a system of ultrasound transceivers. The results show this framework to be an efficient and simple positioning and filtering approach, accurately reconstructing the real path followed by the mobile object while maintaining a low latency. Furthermore, these capabilities can be exploited by using the proposed algorithms for synthetic data generation, as demonstrated in this work, where synthetic ultrasound gesture data has been generated.

Keywords: Infrastructure Positioning, Object Positioning, Multidimensional Scaling, Trajectory Optimization, Ultrasound, Synthetic Data Generation.

4.1. Introduction

Ultrasound technology has been widely used for object positioning. Applications such as robot navigation [1], indoor navigation [2], human-device interface systems [3], body-tracking [4] or medical-probes tracking [5] are just a few examples of the many potential applications based on ultrasonic waves.

Positioning systems usually have a set of fixed anchor nodes that defines the infrastructure for the location. To locate the target, there are mainly two different approaches:

1. Locating an active object, one able to emit and/or receive ultrasonic signals [6] [7].

2. Locating a passive object, one which just reflects the incoming ultrasonic wave emitted by the anchors [8].

Within the active-object alternative, one option is to use the anchors as receivers and the mobile node as signal emitter. Based on the Time of Flight (ToF) or on the Received Signal Strength Indicator (RSSI), the anchors calculate their distance to the object [9]. Another alternative is using the Angle of Arrival (AoA) [10], where the position is obtained from the direction of arrival of the signal to the receiver. This work focuses on mechanisms based on ToF measurements, which are generally more robust and accurate by relying on the predictable velocity of the ultrasonic wave in the air. If a ToF-based mechanism is used, all the anchors require an additional synchronization mechanism to have a common clock reference. Similarly, the roles can be inverted and the anchors can synchronously transmit beacons, being the mobile node the receiver, which computes the distances locally. To achieve the time synchronisation between the nodes, a combination of different technologies may be used in the same system, such as ultra wide band (UWB) and ultrasounds [11]. Another popular approach to avoid the requirement of having a tight synchronization mechanism between the anchors is to use two-way ranging mechanisms [12], in which either the mobile node or the anchors reply with another signal after a fixed amount of time and the one-to-one distances are computed based on the individual round-trip times.

There are works that use the active-object approach, based on ultrasound technology, for positioning and tracking: Chen H. et al. [9] proposed a system where the positioning is based in a fixed receiver array performing the localization of a transmitter array attached to the hand of the user. Chen J. et al. [13] describes using ultrasonic signal and radio signal together to develop a transmitting 3D-pen, and the algorithm to position the pen based on a set of receiving nodes covering the writing plane.

In the passive-object alternative, just the echo, or reflected wave, is detected back by the anchors. This is typically feasible for very short-range applications, like gesture recognition [14], in which the surface to locate is interference-free and has a reflective surface large enough to be easily recognised. The same distance measuring techniques used in the active alternative can be used with the passive alternative [15], taking into account the characteristics of the passive approach.

Ultimately, the accuracy and robustness of the system rely on the dependability of the distance measurements. It is critical to recognise the incoming signal (either reflected or actively transmitted by another node) over the ultrasonic background noise. Several methods are proposed in the literature based on different criteria (time, frequency, phase) [16], being the most popular the cross correlation of the received and expected signal. It requires low computational power, introduces low delay and offers higher robustness against noise when detecting an echo [16] [?].

This technique enables the emission of different signals (i.e. in the case where the anchors play the emitter role) to differentiate between incoming pulses, like Direct Sequence Code-Division Multiple Access (DS-CDMA) [17] [18].

Once the distances to the anchors are obtained, the location of the object can be determined using a positioning algorithm, one based on the trilateration concept [19] [20]. Knowing the position of three anchors $A_1(x_1, y_1, z_1)$, $A_2(x_2, y_2, z_2)$, $A_3(x_3, y_3, z_3)$ and the pairwise distances (d_1, d_2 and d_3), the coordinates of the object, $O(x, y, z)$ can be calculated solving the following system of equations:

$$\begin{cases} (x-x_1)^2+(y-y_1)^2+(z-z_1)^2=d_1^2 \\ (x-x_2)^2+(y-y_2)^2+(z-z_2)^2=d_2^2 \\ (x-x_3)^2+(y-y_3)^2+(z-z_3)^2=d_3^2 \end{cases} \tag{4.1}$$

This algebraic solution corresponds to the cross points of the three spheres with center A_1, A_2 and A_3, and radius d_1, d_2 and d_3, respectively.

Three anchor nodes, and the distance from all three anchors to the object are needed as a minimum requirement to obtain the 3D-location of the object. If there are less than three distances to the anchor nodes (i.e. there is no direct acoustic channel between the object and one anchor), it is not possible to determine the location.

When there are more than 3 anchors involved in the location, we have an overdetermined system, and the method is called multilateration. Its advantage is a potentially increased robustness against inaccurate or missing distances. With N anchors, it is required to solve a system with N equations, making necessary the use of recursive algorithms to obtain an optimal solution [21]:

$$\begin{cases} (x-x_1)^2+(y-y_1)^2+(z-z_1)^2=d_1^2 \\ (x-x_2)^2+(y-y_2)^2+(z-z_2)^2=d_2^2 \\ \qquad\vdots \\ (x-x_N)^2+(y-y_N)^2+(z-z_N)^2=d_N^2 \end{cases} \tag{4.2}$$

To compute the coordinates of the object, and even to previously or simultaneously position the anchor infrastructure, fast and robust algorithms are required. They should be able to easily adapt to a varying number of noisy distances, and therefore not totally reliable. Furthermore, if trajectories are to be obtained, a further processing step is useful to smooth out the path of the object and improve the accuracy of the estimated track.

In this paper several approaches to achieve the object location and tracking are proposed using Multidimensional Scaling (MDS) and optimization algorithms. A qualitative evaluation of these algorithms is performed in this work. In addition, the integration of the algorithms in a synthetic data generation framework is discussed. This use case shows how the dataset creation task, i.e. ultrasound gesture dataset, could benefit from these algorithms due to the high flexibility to configure the desired output with different noise levels and gesture options. At the same time, since the desired data is configured by the user, this framework would generate simultaneously data and labels. By applying this framework, the possibility of incurring in human error is reduced, and also the required time to generate synthetic labeled datasets.

MDS localization techniques have been previously researched, mostly for technologies such as Wireless Sensor Networks (WSN), Radio or 5G [22]. However, to the best of our knowledge, these techniques have not been evaluated in emerging techniques such as ultrasound for airborne applications. Because of this, the aim of this work will be the usage of this algorithm for ultrasound data for target localization.

This work is structured as follows: Section 4.2 presents the objectives to cover in this work . Section 4.3 explains the proposed new algorithms to perform both the infrastructure and target positioning. Section 5.2.1.3 explains the filter techniques studied in this work for smoothing out the trajectory and Section 4.4 describes the simulation performed. Section 4.5 summarizes the results obtained, focusing on different parameters of each algorithm, and methods to improve the results via filtering or changing the infrastructure layout. Finally, Section 5.6 presents the conclusions of this work.

4.2. Envisioned System

The goal of the present work is to analyze the feasibility and performance of a synthetic data generation framework based on the researched algorithms due to its capabilities to accurately generate numerical samples. The required input for the data generation is an initial selection of the followed path (equation or time series of the desired movement). At the same time, this framework would enable the user to generate a more varied dataset since the noise level can be controlled as well as different modifications of the initial data (including rotating, scaling and translating the samples) in 3D axis, which can later be converted to different formats to fit the specific application, i.e. images or voxels.

This framework could ease data gathering tasks as real sensors are not required for this process and it can generate numerous relevant samples that emulate different scenarios/technologies based on the configuration selected by the user such as the anchor distribution and noise levels.

This system would be beneficial for tasks such as gesture recognition based on multiple technologies, which numerous authors are researching. Most of the researches in this field are focused on radar [15, 23], wifi [24] and ultrasound sensors [25, 9, 13]. In this paper the framework will be evaluated for the generation of ultrasound data for gesture recognition. This technology has been selected due to the emerging techniques with ultrasound sensors which could be implemented directly on simple microcontroller-based devices, as the proposed in [25].

The system that is going to be emulated with the proposed framework is assumed to perform the following tasks (Fig. 4.1):

1. Distance estimation. The devices use ultrasound transceiver(s) to locally compute their distances to an object, e.g. the user's hand, typically using ToF-based measurements. The pairwise distances between the anchors are also computed (with a lower frequency) to self-locate the anchor infrastructure.

2. Positioning algorithms. Using the pairwise distances between the anchors obtained in the previous point, the position of each anchor is computed. Then, using these positions and the distances between the user's hand and all the anchors, the current position of the object is computed.

3. Tracking algorithms. The position of the object is periodically updated, effectively obtaining an estimation of its trajectory. This trajectory is filtered to improve its accuracy.

4. Recognition. The estimated trajectory is used as input for a gesture recognition stage, e.g. implemented with a neural network.

The current work focuses on the second and third steps, in which we transform from a temporal series of distances to the 3D trajectory of the object and the 3D positions of the anchors. It is important to say that, even when in this paper the localization algorithms are tested with synthetic data, the proposed algorithms could also be deployed in a real scenario for target positioning.

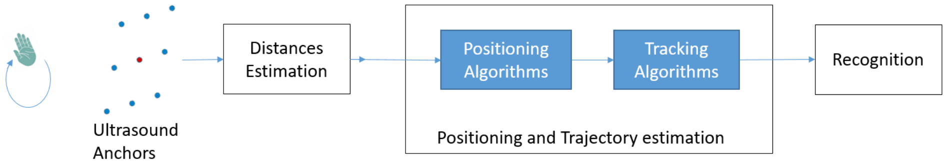


Figure 4.1: Ultrasound positioning system used as a reference for the simulations.

To evaluate the applicability of the proposed algorithms, the following criteria are used:

1. The computational requirements of the positioning and tracking algorithms must be low enough to be executed in real time on low-power devices. Furthermore, they must be flexible enough to adapt to time-varying and noisy conditions, with a potentially variable number of anchors in range.

2. Analyze the accuracy of both the estimated object's trajectory and the anchor's position. The precision of the measured data will directly affect the results when using a classification algorithm to study the data. Because of this, it is important to ensure the high performance of the localization algorithms as well as the proposed filtering techniques. To evaluate this, noise —as typically encountered in ultrasounds systems in this case— is added to the raw distances. The positioning and tracking algorithms must provide optimal estimations and a robust behaviour in the presence of noise, missing distances and outliers.

4.3. Infrastructure and Object Positioning Using Pairwise Distances

As presented in Section 4.1, classical trilateration techniques calculate the position of an object based on the measurement of its distance from different reference points, or anchors, which define the location infrastructure. The algorithms used for infrastructure and object positioning using pairwise distances will be described in this section, presenting the novel two-step approach proposed in this work.

4.3.1. Infrastructure Positioning Using Multidimensional Scaling

For many applications, in which the infrastructure may be portable and flexible, or a quick and seamless deployment is desired, the positions of the anchors may not be known beforehand.

A self-positioning infrastructure, in which relative coordinates of the anchors are obtained from their pairwise distances, can be achieved by using metric multidimensional scaling (mMDS) techniques. All the pairwise Euclidean distances among the anchors shape the dissimilarity matrix, which is then used to calculate the relative coordinates of the anchors by minimizing a stress function based on iterative metric-preserving techniques [26]. The scaling by majorizing a complicated function (SMACOF) algorithm is proposed for a computationally-efficient resolution of the problem [27]. Distances, or equivalently dissimilarities, are considered noisy and some pairs may be missing. To account for different degrees of confidence in the dissimilarities, they can be weighted differently, e.g. from zero (distance is considered missing and ignored during the stress computation) to one. Due to the noisy nature of real-world distance measurements, an analytical exact solution is usually not available, and iterative techniques, like SMACOF, are more suitable.

Different approaches based on MDS are proposed in the literature to improve the computation of the coordinates [28], like matrix completion, in which missing distances are estimated (e.g. with the Dijkstra's shortest path algorithm) instead of being given zero weight. Another is *out-of-sample* MDS [29] [30], in which the position of a subset of anchors (landmarks) can be fixed and only the remaining positions are computed. Furthermore, mixing different steps of non-metric [31] and metric MDS computations can be beneficial, particularly if the dissimilarities are not directly proportional to the Euclidean distances (e.g. they are based on RSSI measurements [32]). The initialization of the SMACOF algorithm may also impact the accuracy of the solution, and it is usual to run it with multiple random initializations and keep the solution with lower stress [33].

The result of the MDS algorithm is a cloud of points, one for every anchor. There are a set of transformations (translation, rotation and reflection) that can be applied to these points, without modifying the stress, which results in an infinite amount of equally valid solutions. The last step is to use physical constraints or general knowledge about how and where the anchors are deployed to apply these transformations and then fix the coordinate system and to go from relative to absolute positions [34].

4.3.2. Object Positioning Using Multidimensional Scaling

Once the coordinates of the anchors are computed, the second step is the addition of the pairwise distances of the moving objects to the dissimilarity matrix, ideally using the *out-of-sample* variation of the MDS algorithm, in order to keep the anchor positions fixed [29]. Nevertheless, MDS is computationally expensive, and while it might be the optimal solution for low-frequency infrastructure positioning, it could be too slow for high-frequency positioning of moving objects, particularly in edge computing and low power environments.

As an alternative, the position of the fixed anchors and the moving objects can be simultaneously computed. The main disadvantage of this approach is that normally the distance measurements between the fixed anchors are more reliable than the measurements between the anchors and the moving objects, e.g. distances between the anchors can be heavily averaged for noise reduction. Introducing noisier dissimilarities in the matrix affects the accuracy of the overall positioning, including that of the infrastructure, resulting in worse results than with the two-step (first MDS without mobile objects, then *out-of-sample* MDS) approach. Furthermore, the fixed infrastructure does not need to be re-positioned as fast as the objects, so it is useful to decouple both computations.

4.3.3. Object Positioning Using Optimization Algorithms

A more efficient approach for object positioning is to compute only the coordinates of the moving objects at a faster rate using a classical optimization algorithm, based on the anchor coordinates previously obtained with MDS. In our case, we choose Limited-Memory Broyden Fletcher Goldfarb Shanno (LM-BFGS) algorithm [35] [36] with the mean squared error as the objective function to be minimized. The processing and memory requirements of LM-BFGS are low-enough to be run in real-time in low-power edge devices with a typical number of distances to the anchors (fewer than a dozen) [37].

The accuracy of this approach is comparable to the *out-of-sample* MDS one, since the position of the anchors are considered fixed during the optimization iterations, but it is faster for independently obtaining the coordinates of individual objects. Furthermore, the error is typically lower than when using the *one-step* MDS approach, in which the coordinates of the anchors and mobile objects are computed at the same time, since the pairwise distances involving the mobile objects are normally noisier.

4.3.4. Trajectory Optimization Using Smoothing Techniques

The coordinates of the moving object create a (typically noisy) trajectory that benefits from proper filtering in order to provide a more accurate estimation for the final application, which could allow real-time localization or hand-gesture recognition. Different techniques are widely used for low-pass filtering and outlier detection. In our approach, we compare simple moving-average and moving-median filters [38] with a fixed window length determined heuristically. They provide optimal results, while keeping both the computational cost and low complexity.

To obtain the trajectory of a moving object, we compute its position periodically, building a discrete time-series of successive equally spaced points in time. Since additional information about the expected path is usually known, like the maximum velocity of the object, we can exploit this to further filter the trajectory, smoothing it out, restoring missing points by interpolation, and decreasing or removing the effect of outliers in the position time-series. We explore two filtering alternatives:

1. Moving-average filter. It is a low-pass filter that provides effective noise reduction, particularly in applications where the focus is on time-response (instead of frequency-response) analysis. It smooths the sig-

nal, but the predicted trajectory may fail to respond to quick movements.

2. Moving-median filter. The median filter is a non-linear filter that replaces the values in data with the moving median of the filtered and neighboring points. It is very robust against outliers and in suppressing spiky noise, but as with the moving-average filter filtering, it may lead to an underestimation of the path, particularly in sharp corners.

4.4. Simulation Setup

After presenting the different alternatives for object and infrastructure positioning based on pairwise distance measurements, we estimated the performance of the proposed algorithms in terms of accuracy and execution speed. We compare the two different approaches depicted in Fig. 4.2:

1. One-step approach (Fig. 4.2a). The SMACOF MDS algorithm is used to simultaneously obtain the positions of the anchors and the moving objects. It is expected to be slower and less accurate if noisy dissimilarities (like those between the moving objects) are introduced in the computation, but all the anchors and object positions are computed simultaneously.

2. Two-step approach (Fig. 4.2b). The SMACOF MDS algorithm is used once to obtain the positions of the anchors. Then, the LM-BFGS optimization algorithm is used to compute only the coordinates of the moving object, and repeated periodically to update its position. This approach is faster, but relies on an accurate initial estimation of the anchor positions.

By using these two algorithms, it is possible to design the proposed framework for synthetic data generation. It executes the SMACOF MDS algorithm periodically to ensure the position of the anchors is correct while locating simultaneously the target. Between these anchors check, the LM-BFGS algorithm is used due to its low latency and high accuracy when the position of the anchors is known.This approach is faster and results in a smaller error, as we will discuss in Sections 4.5.2 and 4.5.4. The SMACOF MDS and LM-BFGS optimization computation steps can be done in a central processing node, to which all the anchors report, or it can be done locally in the anchors or the mobile object, if they have access to all the distances. The particular communication scheme to disseminate the distances and the positions is out of the scope of this work.Finally, if we want to estimate a path, and not only single positions, a smoothing filter is used to compute the trajectory of the object.

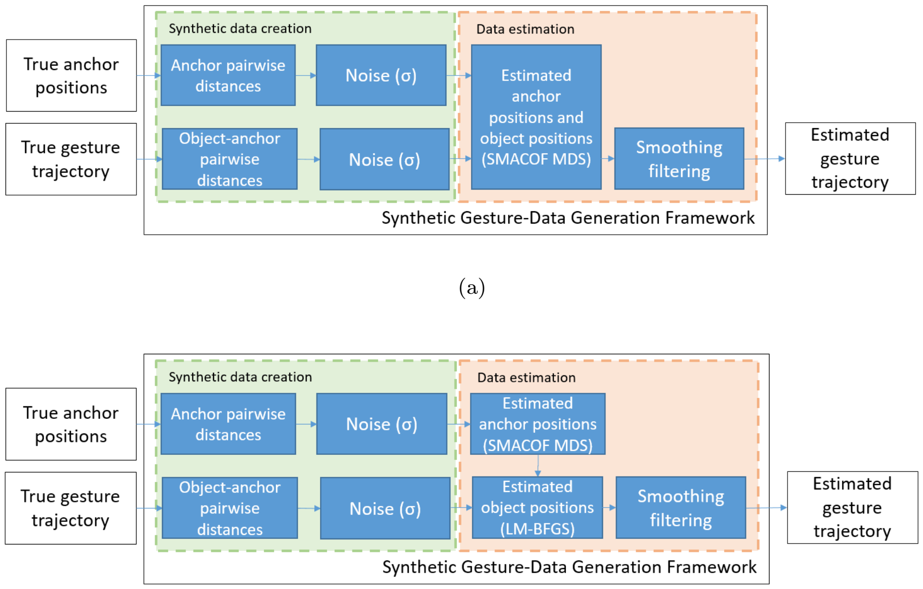

True anchor positions
True gesture trajectory
Synthetic data creation
Anchor pairwise distances
Noise (σ)
Object-anchor pairwise distances
Noise (σ)
Data estimation
Estimated anchor positions and object positions (SMACOF MDS)
Smoothing filtering
Estimated gesture trajectory
Synthetic Gesture-Data Generation Framework
(a)
True anchor positions
True gesture trajectory
Synthetic data creation
Anchor pairwise distances
Noise (σ)
Object-anchor pairwise distances
Noise (σ)
Data estimation
Estimated anchor positions (SMACOF MDS)
Estimated object positions (LM-BFGS)
Smoothing filtering
Estimated gesture trajectory
Synthetic Gesture-Data Generation Framework
(b)

Figure 4.2: Proposed simulation framework for the generation of synthetic gesture data.

Consequently, this framework can be used to generate synthetic trajectories for an arbitrary number of anchor configurations and gestures to fit multiple scenarios and applications, as shown in the Synthetic data creation block in Fig. 4.2. At the same time, data augmentation for a single gesture and anchors setup is possible by varying the random initialization seeds of the noise for the SMACOF MDS and the LM-BFGS optimization algorithms, as shown in the Data estimation block in Fig. 4.2. Consequently, this framework can efficiently generate numerous samples of the desired data to contemplate all the possible results of measurements with real devices. Furthermore, different noise models and strengths can be injected to the raw distances, emulating different disturbances and inaccuracies that the ultrasound distance gathering system can experience in a real deployment.

4.4.1. System Modeling

In this work, the framework emulates a system of nine ultrasound anchors sending a sinusoidal pulse (reference pulse). The echoes are then sampled and the ToF is obtained with classical cross-correlation techniques using a reference signal. Every pairwise distance is computed at 20 Hz, i.e., a new target position is computed every 50 ms. A two-step approach to compute the anchor and target positions as depicted in Fig. 4.2b

The infrastructure of the anchors for the proposed framework is shaped as a 2D array of nine anchors (ultrasound transceivers) located in the same surface (in the XY plane with $z = 0$), which represents a plausible configuration for future applications. Specifically, the anchors are located as seen in Fig. 4.3. The positioning is limited to the space in front of said surface ($z > 0$), since the sign of the z coordinate can not be defined when all the anchors are in the same plane. Furthermore, the ultrasound transceivers sensors used as an experimental support for the simulations have a detection range limited to 180 degrees in the Z axis.

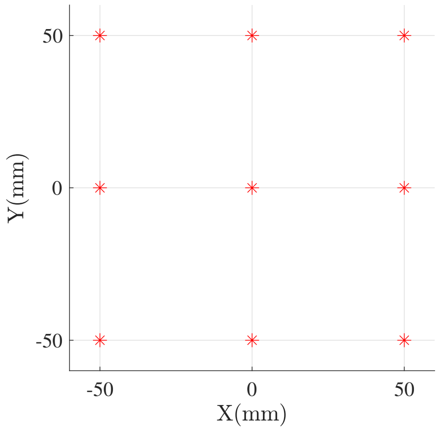


Figure 4.3: Position of the ultrasound anchors

The anchor array is able to transmit and receive ultrasonic signals and to locate passive objects based on ToF measurements. It has two operating modes:

1. To calculate the pairwise distances between the anchors they actively exchange ultrasonic signals (two-way ranging).

2. To calculate the pairwise distances between the anchors and the mobile object, they actively transmit and then sense the reflected echo. Anchors can be synchronized, in which case only one of the transceivers needs to transmit and they all can receive the echo and timestamp it based on a common clock. Otherwise, they can all transmit and sense only the echo coming from their own transmission, in such case, time synchronization is not required.

4.4.2. Noise Modeling

The noise in the distances obtained with an ultrasound-based measurement system depends on the accuracy of the ToF samples. There are different factors that impact the performance, such as the bandwidth of the transmitted pulse and the sampling rate of the acquisition stage.

Based on our experimental measurements using the system of Fig. 4.1, the noise, N, in the computed distances, $d(t)$, can be modeled as unbiased (zero average) additive white Gaussian noise (AWGN), with a given standard deviation, σ, and a probability density function, $pdf(N)$, given by:

$$d(t) = d_{real}(t) + N(t) \tag{4.3}$$

$$pdf(N) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \tag{4.4}$$

In a representative x-y-z point, $\mathbf{P} = (\mathbf{0\ cm},\ \mathbf{0\ cm},\ \mathbf{50\ cm})$, the measured equivalent noise in the euclidean distance, after acquiring 50,000 samples, can be fitted with a $\sigma = 3{,}2\ mm$, as seen in Fig. 4.4. This provides a good estimation of the scale of the expected noise in a real system, and it is used as a reference to model the noise in the simulations.

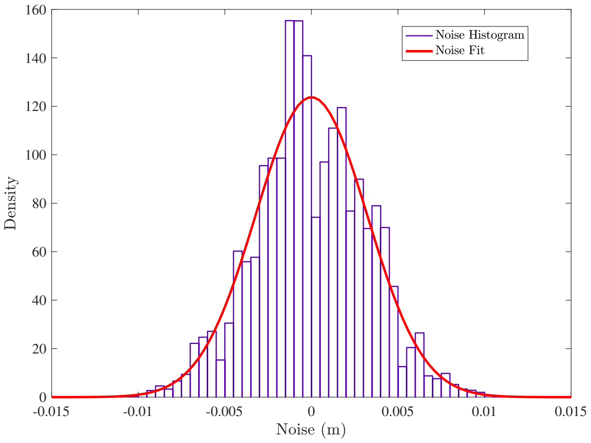


Figure 4.4: Histogram of the measured noise in experimentally-computed distances and normal distribution fit ($\sigma = 3{,}2\ mm$)

As summary, in this section the two-steps algorithm proposed and evaluated in this work has been described, as well as the steps performed to particularize the framework for validate the use of ultrasonic system as technology for that algorithm

4.5. Performance Results

In this section, the performance of the proposed algorithms for the object position and anchors localization is analyzed. At the same time, the utility of the proposed system for data generation tasks is discussed.

4.5.1. Infrastructure-Positioning Accuracy

First, we characterize the performance of SMACOF MDS locating the infrastructure by building the dissimilarity matrix, D, with the metric distances measured between the anchors, adding different realistic levels of AWGN noise, according to the experimental results obtained in subsection 4.4.2.

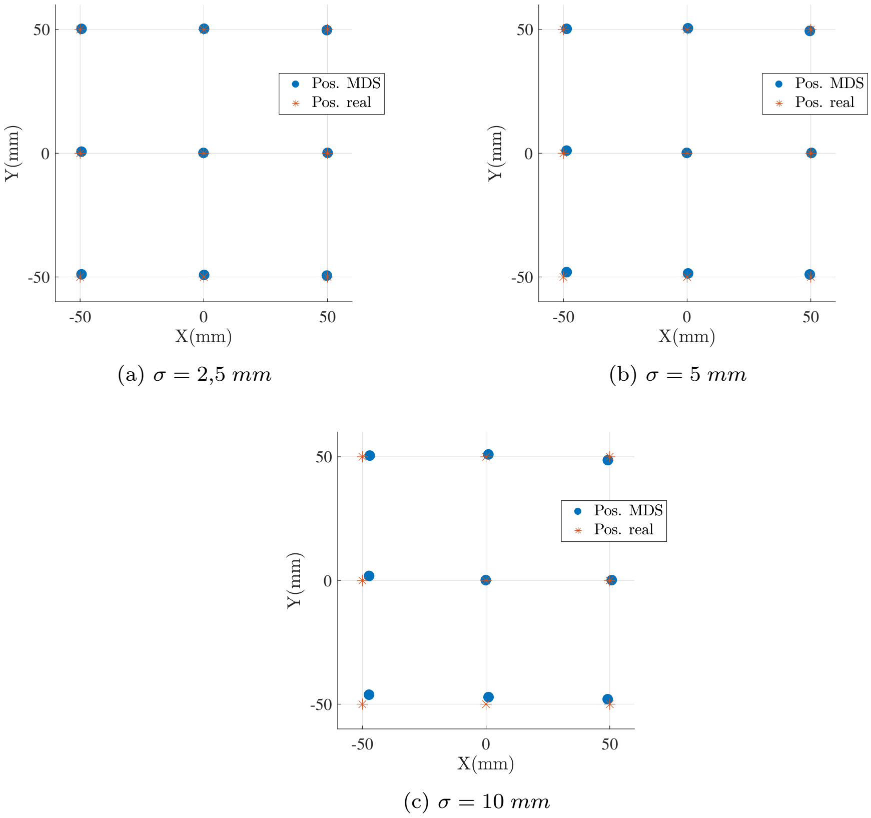
Since the positions computed by SMACOF MDS are equally valid if they are translated, rotated or mirrored; we need additional constraints to fix the coordinate system. We use an approach to agree on a common absolute reference that requires one translation and three rotations:

1. We translate the points so that the first anchor, m_0, fixes the origin of the coordinates system, $\mathbf{P_{m_0}} = (\mathbf{0}, \mathbf{0}, \mathbf{0})$.
2. We rotate the points around the X axis at an angle such that the second anchor, m_1, is in $z = 0$, $\mathbf{P_{m_1}} = (\mathbf{a}, \mathbf{b}, \mathbf{0})$.
3. We rotate the points around the X axis at an angle such that the second anchor, m_1, is in $y = 0$ and $c > 0$, $\mathbf{P_{m_1}} = (\mathbf{c}, \mathbf{0}, \mathbf{0})$.
4. We rotate the points around the X axis at an angle such that the third anchor, m_2, is in $z = 0$ and $e > 0$, $\mathbf{P_{m_2}} = (\mathbf{d}, \mathbf{e}, \mathbf{0})$.
5. In our configuration, all the anchors are in the same surface ($z = 0$), so no additional condition is required. In general, it is required to fix the positive direction of the X axis. If the anchors constitute a three-dimensional shape, another anchor (e.g. m_3), located in a surface different to the previous three (m_0 to m_2) is used to define the positive X direction.

Following these steps, we obtain a consistent reference system every time, removing any ambiguity in the positions. The results locating the anchor infrastructure depending on the strength, σ, of the AWGN noise in the distances are shown in Fig. 4.5. The error on the X and Y axes between the real and calculated positions can be observed. Table 4.1 provides the quantitative comparison for this error in the X, Y and Z axes, and also the total displacement. We obtain an optimal accuracy, with an expected error comparable to the deviation introduced by the noise strength.

4.5.2. Object-Positioning Accuracy

Once the anchor infrastructure is positioned, either with the MDS-based self-locating mechanisms explained in subsection 4.5.1, or because the positions of the anchors are previously known, we can either apply LM-BFGS optimization to locate the mobile objects or reapply SMACOF MDS adding the distance of the mobile object to the dissimilarity matrix. For the



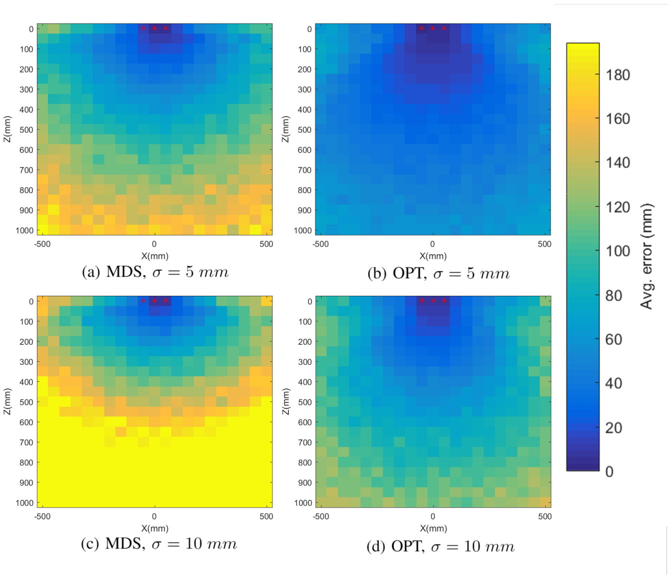
(a) $\sigma = 2{,}5\ mm$

(b) $\sigma = 5\ mm$

(c) $\sigma = 10\ mm$

Figure 4.5: Simulated infrastructure-positioning results for different noise levels using SMACOF MDS (MDS)

simulations, we use the anchor infrastructure of Fig. 4.3, and repeat the positioning of 250,000 objects randomly distributed in the surface given by $0 < z < 1000\ mm$; $-500\ mm < x < 500\ mm$ and $y = 0$. As shown in Fig. 4.6 and Table 4.2, since the anchor infrastructure is symmetrical with respect to the $z = 0$ surface, the averaged error in the X and Y axes is similar, while it is generally much larger for the Z axis. The chosen anchor infrastructure is flat, with no variability in the X axis. Such a configuration works well for locating in the X and Y axes, but it struggles in the Z axis. The error also increases as we move away from the anchors, being minimal in the region close to them. LM-BFGS optimization, besides being faster, has fewer error, since it only tries to optimize the position of the mobile object, without trying to minimize the error by also relocating the fixed anchors. The results further validate our simple two-step approach of using SMACOF MDS just for positioning the infrastructure and then employing LM-BFGS optimization to update the position of the mobile object.



(a) MDS, $\sigma = 5\ mm$

(b) OPT, $\sigma = 5\ mm$

(c) MDS, $\sigma = 10\ mm$

(d) OPT, $\sigma = 10\ mm$

Figure 4.6: Average object-positioning error (Euclidean distance between the real and the computed point) in different regions and for different noise levels, with SMACOF MDS (MDS) and LM-BFGS optimization (OPT)

4.5.3. Trajectory-Positioning Accuracy

As a relevant application of proposed system for synthetic data generation, we simulated the data acquisition process based on ultrasound transceivers for gesture recognition. This framework was evaluated by creating three different gestures based on some initial time series of the 3D coordinates of the trajectory.

The chosen gestures are presented in Fig. 4.7. They simulate the imperfections of real gestures. The first one (Fig. 4.7a, $circle_{XY}$) is a circular shape in a surface with low X axis variability ($100 < z < 120$), and mostly parallel to the anchor infrastructure, to simulate a gesture in the optimal recognising conditions. The second one (Fig. 4.7b, $loop_{XY}$) is a gesture with sharp corners, in the same surface, to evaluate the performance of the smoothing

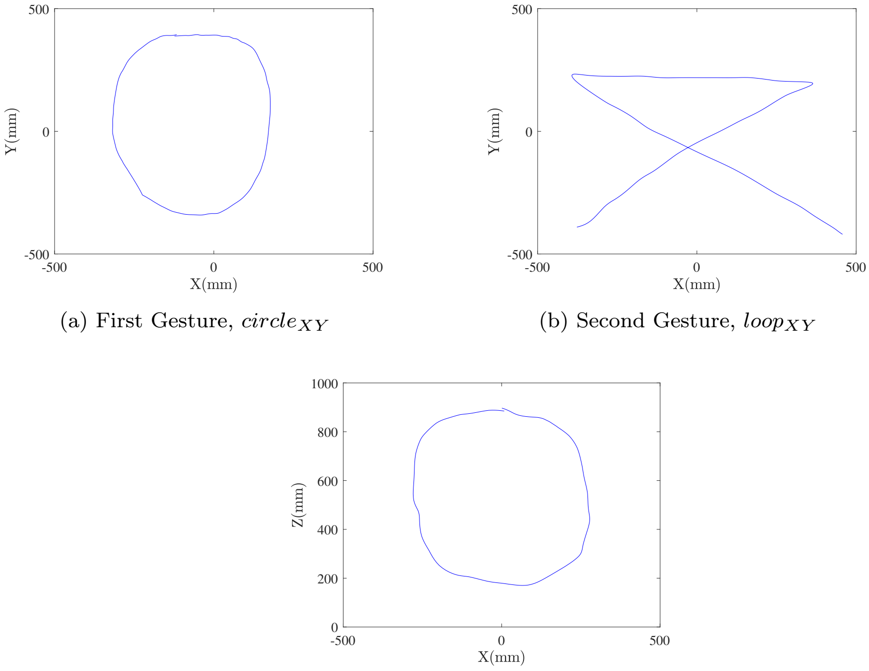
		Average Positioning Error (mm)			
	Algorithm	σ=0mm	σ=2.5mm	σ=5mm	σ=10mm
X Ax.	SMACOF MDS	0	0.02	0.08	0.11
Y Ax.	SMACOF MDS	0	0.03	0.08	0.10
Z Ax.	SMACOF MDS	0	0.01	0.01	0.03
Euc. Dis.	SMACOF MDS	0	0.03	0.08	0.14

Table 4.1: Simulated infrastructure-positioning average error (of the nine anchors) for different noise levels with SMACOF MDS, per-axis and Euclidean distance

		Average Positioning Error (mm)			
	Algorithm	σ=0mm	σ=2.5mm	σ=5mm	σ=10mm
X Ax.	SMACOF MDS	0.03	16.35	28.39	48.21
	LM-BFGS	0.01	7.64	15.21	28.69
Y Ax.	SMACOF MDS	0.03	15.67	26.98	45.54
	LM-BFGS	0.01	7.53	14.81	28.88
Z Ax.	SMACOF MDS	0.17	36.94	56.91	79.65
	LM-BFGS	0.04	23.25	42.05	67.50
Euc. Dis.	SMACOF MDS	0.19	47.44	75.85	114.79
	LM-BFGS	0.05	27.68	50.97	85.91

Table 4.2: Per-axis object-positioning error and Euclidean distance between the real and the computed point for different noise levels, with SMACOF MDS and LM-BFGS optimization, averaged for the $0 < z < 1000\ mm$; $-500\ mm < x < 500\ mm$ and $y = 0$ surface

filters in gestures with sharp edges. The third one (Fig. 4.7c, $circle_{XZ}$) is another circular shape, this time located in a surface mostly perpendicular to the anchor infrastructure ($100 < y < 120$), to check the worst-case performance (strong z variability). The first gesture has 400 samples, while the second and third have 250 samples each. They are all sampled at 20 Hz.



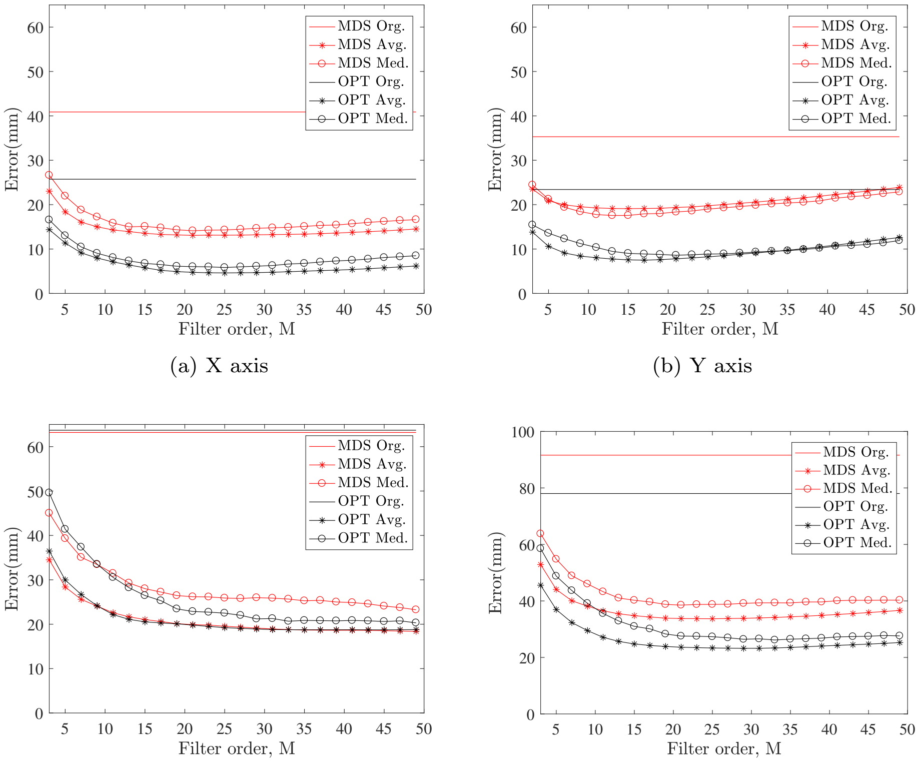
(a) First Gesture, $circle_{XY}$

(b) Second Gesture, $loop_{XY}$

(c) Third Gesture, $circle_{XZ}$

Figure 4.7: Gestures used in the simulations

First of all, the frame tests different filter orders, M, to heuristically choose a value in which the error is properly minimized. In Fig. 4.8 the error dependency with the filter order is shown, for both filters described (moving-median and moving-average) and for both positioning algorithms (SMACOF MDS and LM-BFGS optimization). The error without filtering has been added too, as a reference. Based on those results, $M = 11$ has been selected as a good trade-off between added delay and error suppression. With the considered AWGN noise model, the moving-average filtering performs consistently better than the moving-median filter, which may change if spiky noise and outliers are introduced in the model.



(a) X axis

(b) Y axis

(c) Z axis

(d) Euclidean Distance

Figure 4.8: Comparison of the average trajectory error for the $circle_{xy}$ gesture, with a fixed noise level ($\sigma = 10\ mm$), with moving-median(Med.), moving-average(Avg.) smoothing filter and without (Org.) smoothing, for SMACOF MDS (MDS) and LM-BFGS optimization (OPT) algorithms

Without added noise ($\sigma = 0\ mm$), we can see in Table 4.3 that the smoothing filters actually deteriorate the performance, since the edges are underestimated. In a realistic noisy scenario, the smoothing filter greatly reduces the error, as shown in Tables 4.4, 4.5 and 4.6, where the average error (in each axis and in total) is compared for each algorithm and each filtering technique previously described. As seen in Fig. 4.9, as expected, the noisiest gesture in the surface of interest is the $circle_{XZ}$, due to it being

			Average Positioning Error (mm)		
			$circle_{XY}$	$loop_{XY}$	$circle_{XZ}$
X Ax.	SMACOF MDS	Original	0.03	0.02	0.04
		Average	0.69	2.71	1.67
		Median	0.19	1.02	0.14
	LM-BFGS	Original	0.01	0.01	0.01
		Average	0.69	2.72	1.67
		Median	0.17	1.01	0.11
Y Ax.	SMACOF MDS	Original	0.03	0.03	0.04
		Average	1.4	1.79	0.29
		Median	0.07	0.67	0.19
	LM-BFGS	Original	0.01	0.01	0.01
		Average	1.4	1.79	0.29
		Median	0.05	0.66	0.18
Z Ax.	SMACOF MDS	Original	0.08	0.09	0.02
		Average	0.34	0.36	2.17
		Median	0.22	0.28	0.38
	LM-BFGS	Original	0.02	0.02	0.01
		Average	0.34	0.34	2.17
		Median	0.19	0.26	0.37
Euc. Dis.	SMACOF MDS	Original	0.09	0.1	0.07
		Average	1.76	3.55	3.07
		Median	0.43	1.49	0.62
	LM-BFGS	Original	0.03	0.03	0.02
		Average	1.76	3.54	3.07
		Median	0.39	1.46	0.6

Table 4.3: Average trajectory error for the three different gestures without noise ($\sigma = 0\,mm$), with no smoothing (Original) and with 11th order ($M = 11$) moving-average (Average) and moving-median (Median) smoothing filters, for SMACOF MDS and LM-BFGS optimization algorithms

performed in a plane normal to the anchor surface. The smoothing filtering also struggles in the sharp edges of the $loop_{XY}$ gesture, while it estimates the path with a very good accuracy for the inherently smooth circular gestures.

Nevertheless, the achieved results show how the proposed algorithms are able to accurately generate the desired gesture/trajectory data with different noise levels. This would enable the framework to simulate a large range of possible scenarios and sensors. The noise level can be defined by the user to adapt the specific application and scenario that the framework is emulating.

4.5.4. Execution Time

We use the implementation of the SMACOF MDS and LM-BFGS optimization algorithms included in the Python library for machine learning *scikit-learn 0.23*, and run them in a typical Windows laptop (Intel Core i5-8350U@1.70GHz., 8GB RAM). Since absolute values depend on the processing capabilities of the particular machine, only relative time differences are evaluated. In this case the evaluation has been performed over the gesture represented in Figure 4.7a. As seen in Table 4.7, the SMACOF MDS algorithm is about ten times slower, which might be relevant for real-time applications, particularly if the position inference needs to be done directly in an embedded edge device, which typically has constrained resources. These constrains could lead to a larger difference between these algorithms due to the computing power required to execute the SMACOF MDS algorithm. In

			Average Positioning Error (mm)		
			σ=2.5mm	σ=5mm	σ=10mm
X Ax.	SMACOF MDS	Original	14.56	23.68	40.88
		Average	5.05	8.35	14.30
		Median	6.26	9.84	15.84
	LM-BFGS	Original	6.68	13.24	25.73
		Average	2.19	4.02	7.11
		Median	3.36	5.45	8.08
Y Ax.	SMACOF MDS	Original	13.42	21.72	35.29
		Average	4.84	9.81	19.26
		Median	6.23	9.50	17.76
	LM-BFGS	Original	6.43	12.82	23.41
		Average	2.54	4.22	8.04
		Median	3.17	5.44	10.36
Z Ax.	SMACOF MDS	Original	33.33	48.24	63.19
		Average	10.46	15.52	22.56
		Median	12.65	19.12	31.46
	LM-BFGS	Original	18.61	38.22	63.7
		Average	7.54	15.73	22.16
		Median	7.21	13.72	30.55
Euc. Dis.	SMACOF MDS	Original	41.93	62.72	91.61
		Average	13.72	22.02	36.5
		Median	17.01	25.79	43.27
	LM-BFGS	Original	22.28	45.34	78.02
		Average	9.02	18.02	27.12
		Median	9.68	17.39	35.56

Table 4.4: Average trajectory error for the $circle_{XY}$ gesture, with no smoothing (Original) and with 11th order ($M = 11$) moving-average (Average) and moving-median (Median) smoothing filters, for SMACOF MDS and LM-BFGS optimization algorithms

			Average Positioning Error (mm)		
			σ=2.5mm	σ=5mm	σ=10mm
X Ax.	SMACOF MDS	Original	11.44	19.33	33.62
		Average	6.28	8.23	18.59
		Median	8.95	12.01	19.29
	LM-BFGS	Original	5.30	10.18	21.5
		Average	3.26	4.55	11.02
		Median	5.10	7.88	13.77
Y Ax.	SMACOF MDS	Original	11.11	18.32	34.4
		Average	4.98	7.29	18.63
		Median	6.64	8.59	21.52
	LM-BFGS	Original	5.75	10.64	22.07
		Average	2.45	3.95	8.75
		Median	3.48	6.12	11.34
Z Ax.	SMACOF MDS	Original	29.62	42.04	62.45
		Average	11.73	13.75	37.2
		Median	15.14	16.37	42.33
	LM-BFGS	Original	18.9	32.85	53.58
		Average	6.71	13.07	17.92
		Median	8.80	13.14	24.65
Euc. Dis.	SMACOF MDS	Original	36.09	53.85	85.89
		Average	15.48	19.33	48.43
		Median	20.77	24.9	55.18
	LM-BFGS	Original	21.92	38.82	66.6
		Average	8.96	15.65	25.77
		Median	12.19	18.99	33.85

Table 4.5: Average trajectory error for the $loop_{XY}$ gesture, with no smoothing (Original) and with 11th order ($M = 11$) moving-average (Average) and moving-median (Median) smoothing filters, for SMACOF MDS and LM-BFGS optimization algorithms

			Average Positioning Error (mm)		
			σ=2.5mm	σ=5mm	σ=10mm
X Ax.	SMACOF MDS	Original	32.46	65.27	1.5
		Average	8.98	19.12	32.04
		Median	12.25	25.04	35.4
	LM-BFGS	Original	11.74	25.61	47.97
		Average	3.97	7.80	16.2
		Median	5.31	9.45	17.85
Y Ax.	SMACOF MDS	Original	34.46	63.76	88.26
		Average	12.16	19.25	33.57
		Median	13.76	23.6	34.62
	LM-BFGS	Original	11.01	23.14	49.88
		Average	3.94	8.76	15.84
		Median	4.84	11.55	22.7
Z Ax.	SMACOF MDS	Original	12.85	26.46	47.18
		Average	5.59	14.14	33.14
		Median	7.36	12.96	32.24
	LM-BFGS	Original	4.71	9.93	19.13
		Average	2.61	4.07	9.21
		Median	3.91	5.80	9.80
Euc. Dis.	SMACOF MDS	Original	53.97	105.00	158.170
		Average	18.53	33.72	62.72
		Median	23.26	41.27	66.34
	LM-BFGS	Original	18.62	40.13	80.59
		Average	7.09	14.33	27.79
		Median	9.6	18.47	34.19

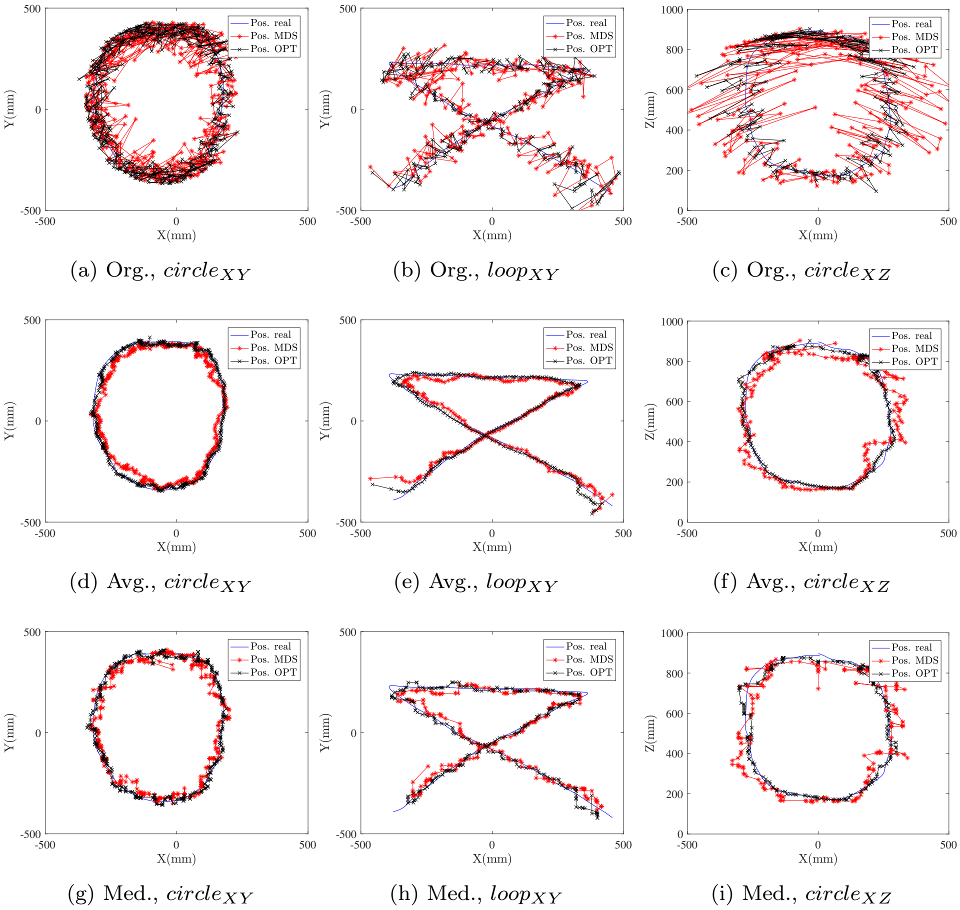
Table 4.6: Average trajectory error for the $circle_{XZ}$ gesture, with no smoothing (Original) and with 11th order ($M = 11$) moving-average (Average) and moving-median (Median) smoothing filters, for SMACOF MDS and LM-BFGS optimization algorithms

the context of continuously positioning a slowly (relative to the positioning sampling rate) moving object initializing the algorithm with the previous position greatly reduces (it takes half the time) the execution time of SMACOF MDS, while for LM-BFGS optimization it barely changes.

		Time (ms)
SMACOF MDS	Random initialization	162
	Previous-point initialization	77
LM-BFGS	Random initialization	16
	Previous-point initialization	15

Table 4.7: Average computation time of individual positions after 1,000 executions of the gesture of Fig. 4.7a using SMACOF MDS and LM-BFGS optimization algorithms

Although SMACOF MDS is slower, it needs to be stressed that its output includes the position of all the anchors, while LM-BFGS optimization only computes the position of a single moving object. Therefore, SMACOF MDS is relevant for self-calibrating anchor infrastructures or by simultaneously locating multiple moving objects. Once the infrastructure is properly positioned during the initial set-up phase, LM-BFGS optimization can be used for quick updates of the object position.



(a) Org., $circle_{XY}$ (b) Org., $loop_{XY}$ (c) Org., $circle_{XZ}$

(d) Avg., $circle_{XY}$ (e) Avg., $loop_{XY}$ (f) Avg., $circle_{XZ}$

(g) Med., $circle_{XY}$ (h) Med., $loop_{XY}$ (i) Med., $circle_{XZ}$

Figure 4.9: Comparison between the real and estimated trajectories for SMACOF MDS (MDS) and LM-BFGS optimization (OPT) algorithms for the three different gestures, with a fixed noise level ($\sigma = 10\ mm$), without smoothing (sub-figures a,b,c), with moving-average (Avg.) smoothing (sub-figures d,e,f) and with moving-median (Med.) smoothing (sub-figures g,h,i)

4.6. Conclusion

This work presents a novel two-step technique to perform general infrastructure and moving-object positioning based on measured pairwise distances. In the first step MDS is used to obtain the coordinates of the anchors, repeated with a low frequency, e.g. to correct minor and infrequent potential displacements of the anchors. We use the SMACOF variant of the mMDS family of algorithms. With the coordinates of the computed anchors computed, a fast optimization algorithm is used to obtain the unknown coordinates of the objects. This step is repeated with a high frequency. The LM-BFGS optimization algorithm has been used for this step. Its performance is thoroughly analyzed with simulations, particularized to the use case of a system with ultrasound transceivers. The distribution and shape of the anchor infrastructure, the size of the region in which the positioning takes place and the strength of the noise are realistically modeled after such a system.

This two-step approach described in the work would be optimal in scenarios where the position of the anchors do not change frequently through time. Therefore, the one-step approach described in section 4.4, in which all the positions are computed at the same time, is reserved for special situations, e.g. when there are no anchors (all the objects are considered mobile) or when we want to simultaneously obtain the position of several (more than a dozen) mobile objects. For the rest of the scenarios, our approach performs the localization with low computational time, making it suitable for its use in real-time systems and even in constrained edge devices.

Efficient and simple filtering techniques significantly reduce the error and improve the reconstruction of the real path followed by the mobile object. This feature can be exploited when using the proposed algorithms for synthetic data generation. The current dataset creation step for applications, such as AI models, are time consuming due to the complexity of the recording and labeling tasks, which could be reduced by using the proposed system as a synthetic data generation framework. This framework is independent from hardware and it could simulate trajectories/movement from a large range of sensors. The parameters of this framework (noise, gesture and anchors number and position) are defined by the user through the initial configurations.

The use of ultrasonic signals for target positioning has been widely researched, but to the best of our knowledge, our two-step approach inspired by wireless sensor network's positioning algorithms has not been used or described. The proposed technique enables using an arbitrary number of ultrasound transceivers and removes the constrain of knowing the position of the anchors beforehand, while providing an optimal AWGN rejection. This could drive the adoption of ultrasound technology in the positioning field and foster the research of novel applications and electronic components based on non-audible acoustic waves.

References

[1] John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. volume 7, pages 376–382, 1991.

[2] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the cricket location system. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 190–202, 2004.

[3] Chris Hand. A survey of 3d interaction techniques. In *Computer graphics forum*, volume 16, pages 269–281. Wiley Online Library, 1997.

[4] Christopher R. Wren, Ali Azarbayejani, Trevor Darrell, and Alex P. Pentland. Pfinder: Real-time tracking of the human body. volume 19, pages 780–785. IEEE, 1997.

[5] David G. Gobbi, Roch M. Comeau, and Terry M. Peters. Ultrasound probe tracking for real-time ultrasound/mri overlay and visualization of brain shift. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 920–927. Springer, 1999.

[6] Jun Qi and Guo-Ping Liu. A robust high-accuracy ultrasound indoor positioning system based on a wireless sensor network. *Sensors*, 17(11):2554, 2017.

[7] Carlos Medina, José Carlos Segura, and Angel De la Torre. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, 13(3):3501–3526, 2013.

[8] Antonin Povolny, Hiroshige Kikura, and Tomonori Ihara. Ultrasound pulse-echo coupled with a tracking technique for simultaneous measurement of multiple bubbles. *Sensors*, 18(5):1327, 2018.

[9] Hui Chen, Tarig Ballal, Ali Hussein Muqaibel, Xiangliang Zhang, and Tareq Y Al-Naffouri. Air writing via receiver array-based ultrasonic source localization. *IEEE Transactions on Instrumentation and Measurement*, 69(10):8088–8101, 2020.

[10] Maximo Cobos, Fabio Antonacci, Anastasios Alexandridis, Athanasios Mouchtaris, and Bowon Lee. A survey of sound source localization methods in wireless acoustic sensor networks. volume 2017. Hindawi, 2017.

[11] Veronika Putz, Julia Mayer, Harald Fenzl, Richard Schmidt, Markus Pichler-Scheder, and Christian Kastl. Cyber–Physical Mobile Arm Gesture Recognition using Ultrasound and Motion Data. In *Proc. of the 3rd IEEE International Conference on Industrial Cyber–Physical Systems*, 2020.

[12] Taavi Laadung, Sander Ulp, Muhammad M. Alam, and Yannick Le Moullec. Active-passive two-way ranging using uwb. In *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–5, 2020.

[13] Jian Chen, Fan Yu, Jiaxin Yu, and Lin Lin. A three-dimensional pen-like ultrasonic positioning system based on quasi-spherical pvdf ultrasonic transmitter. *IEEE Sensors Journal*, 21(2):1756–1763, 2020.

[14] Costas Yiallourides and Pablo P. Parada. Low power ultrasonic gesture recognition for mobile handsets. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2697–2701, 2019.

[15] Muhammad Arsalan and Avik Santra. Character recognition in air-writing based on network of radars for human-machine interface. *IEEE Sensors Journal*, 19(19):8855–8864, 2019.

[16] Joseph C. Jackson, Rahul Summan, Gordon I. Dobie, Simon M. Whiteley, S. Gareth Pierce, and Gordon Hayward. Time-of-flight measurement techniques for airborne ultrasonic ranging. volume 60, pages 343–355, 2013.

[17] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112, 2012.

[18] Alejandro Lindo, Enrique Garcia, Jesus Urena, Maria del Carmen, and Alvaro Hernandez. Multiband waveform design for an ultrasonic indoor positioning system. volume 15, pages 7190–7199. IEEE, 2015.

[19] J. M. Martın, A. R. Jiménez, F. Seco, L. Calderón, Jose L. Pons, and R. Ceres. Estimating the 3d-position from time delay data of us-waves: experimental analysis and a new processing algorithm. volume 101, pages 311–321. Elsevier, 2002.

[20] Daniel Ruiz, Jesús Ureña, Juan C. García, Carmen Pérez, José M. Villadangos, and Enrique García. Efficient trilateration algorithm using time differences of arrival. volume 193, pages 220–232. Elsevier, 2013.

[21] Abdelmoumen Norrdine. An algebraic solution to the multilateration problem. In *Proceedings of the 15th international conference on indoor positioning and indoor navigation, Sydney, Australia*, volume 1315, 2012.

[22] Nasir Saeed, Haewoon Nam, Tareq Y Al-Naffouri, and Mohamed-Slim Alouini. A state-of-the-art survey on multidimensional scaling-based localization techniques. *IEEE Communications Surveys & Tutorials*, 21(4):3565–3583, 2019.

[23] Seong Kyu Leem, Faheem Khan, and Sung Ho Cho. Detecting mid-air gestures for digit writing with radio sensors and a cnn. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1066–1081, 2019.

[24] Yinyin Fang, Yong Xu, Heju Li, Xin He, and Longlong Kang. Writing in the air: Recognize letters using deep learning through wifi signals. In

2020 6th International Conference on Big Data Computing and Communications (BIGCOM), pages 8–14, 2020.

[25] Borja Saez, Javier Mendez, Miguel Molina, Encarnación Castillo, Manuel Pegalajar, and Diego P Morales. Gesture recognition with ultrasounds and edge computing. *IEEE Access*, 9:38999–39008, 2021.

[26] Hervé Abdi. Metric multidimensional scaling (mds): analyzing distance matrices. pages 1–13. Sage Thousand Oaks, CA, 2007.

[27] Jan De Leeuw and Patrick Mair. Multidimensional scaling using majorization: Smacof in r. 2011.

[28] Nasir Saeed, Haewoon Nam, Mian I. U.l Haq, and Dost B. Muhammad Saqib. A survey on multidimensional scaling. volume 51, pages 1–25. ACM New York, NY, USA, 2018.

[29] Michael W. Trosset and Carey E. Priebe. The out-of-sample problem for classical multidimensional scaling. volume 52, pages 4635–4642. Elsevier, 2008.

[30] Vin De Silva and Joshua B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University, 2004.

[31] Ling Xiao, Renfa Li, and Juan Luo. Sensor localization based on nonmetric multidimensional scaling. volume 2. Citeseer, 2006.

[32] Shuxia Wang. Wireless network indoor positioning method using nonmetric multidimensional scaling and rssi in the internet of things environment. volume 2020. Hindawi, 2020.

[33] Marco Patanè, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Wireless sensor networks localization with outliers and structured missing data. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7. IEEE, 2017.

[34] Xinrong Li. Collaborative localization with received-signal strength in wireless sensor networks. volume 56, pages 3807–3817. IEEE, 2007.

[35] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. volume 45, pages 503–528. Springer, 1989.

[36] José L. Morales. A numerical study of limited memory bfgs methods. volume 15, pages 481–487. Elsevier, 2002.

[37] Aryan Mokhtari and Alejandro Ribeiro. Global convergence of online limited memory bfgs. volume 16, pages 3151–3181. JMLR. org, 2015.

[38] Deepti Singhal and Rama M. Garimella. Simple median based information fusion in wireless sensor network. In *2012 International Conference on Computer Communication and Informatics*, pages 1–7. IEEE, 2012.

Chapter 5

Air-Writing Character Recognition with Ultrasonic Transceivers

Borja Saez-Mingorance [1,2,], Javier Mendez-Gomez [1,2], Gianfranco Mauro [1,2], Encarnacion Castillo-Morales [2],Manuel Pegalajar-Cuellar [3], and Diego P. Morales-Santos [2]

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany.

2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain.

3. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

MDPI Sensors

- Received September 2021, Accepted September 2021, Published October 2021
- DOI: 10.3390/s21206700
- Impact factor: 3.576
- JCR Rank: 14/64 in category Instrument and Instrumentation (Q1)

Air-Writing Character Recognition with Ultrasonic Transceivers

Abstract: The interfaces between users and systems are evolving into a more natural communication, including user gestures as part of the interaction, where Air-Writing is an emerging application for this purpose. The aim of this work is to propose a new air-writing system based on only one array of ultrasonic transceivers. This track will be obtained based on the pairwise distance of the hand marker with each transceiver. After acquiring the track, different Deep Learning algorithms, such as Long-Short Term Memory (LSTM), Convolutional Neural Networks (CNN), Convolutional Autoencoder (ConvAutoencoder), and Convolutional LSTM have been evaluated for the character recognition. It has been shown how these algorithms provide high accuracy, where the best result is extracted from the ConvLSTM, with a 99.51 % accuracy and 71.01 milliseconds of latency. Real data has been used in this work to evaluate the proposed system in a real scenario to demonstrate its high performance regarding data acquisition and classification.

Keywords: Ultrasound, Air-writing, Gesture Recognition, Deep Learning

5.1. Introduction

Air-Writing is a particular case of gesture recognition. The user draws in the air the character or word to recognize, and the system performs the tracking of the movement and matches the drawn with the actual character or word [1, 2]. Air-writing systems present several challenges, as the lack of a physical writing plane (gestures performed in an imaginary plane) and the detection of starting and ending points of the drawn character. In addition, these systems present a lack of visual feedback when a sequence of tracks is performed, thus increasing the recognition task complexity.

Air-writing systems are described in the literature using different technologies. There are systems based on video [3], infrared (IR) sensors [4, 5], radar [6, 7, 8], WiFi signal [9], RFID [10], or combination of those technologies (i.e IR sensors and video [11, 12]). There are also works based on ultrasound technology such as Chen H. et al. [13], who proposed a system where the recognition is based in a fixed receiver array performing the localization of a transmitter array attached to the user's hand. Similarly, Chen J. et al. [14] describe the use of ultrasonic signal and radio signal together to develop a transmitter 3D-pen, and the algorithm to positioning it based in a set of receiver nodes.

The aim of this work is to propose a new air-writing system based on only one array of ultrasonic transceivers, which will perform both emitter and receiver roles. This array will remove the necessity of any active part used by the person performing the character track. This track will be calculated based on the pairwise distance of the hand marker with each transceiver.

The development of the air-writing system can be depicted as the study of two individual tasks, as shown in Fig. 5.1. The first task consists in the estimation of the character drawn by the user. After the track is estimated, in the second task, the recognition of the character will be performed. To do so, the recognition algorithm may execute necessary transformations to the track.

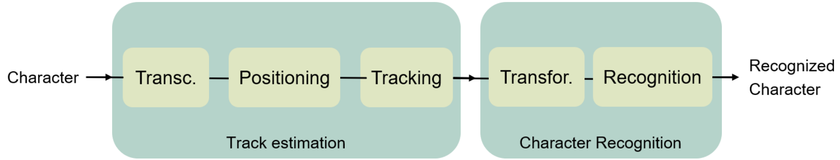


Figure 5.1: General block diagram for air-writing systems

The first task starts with signal acquisition and analysis. While this is highly technology-dependent, the following steps may use common approaches. The hand marker location can be based on method such a Time of Flight (ToF)[15], Time Difference of Arrival (TDoA)[16], Direction of Arrival (DoA)[17], Angle of Arrival(AoA) [18], etc. After the parameter calculation, the actual position has to be computed. The preferred method in the literature to determine the position is Trilateration (or its generalization for multiple nodes, Multilateration) [19]. The system described in this paper is based on the ToF method to acquire the hand marker position. The 3D position is determined by using multidimensional scaling (MDS) and optimization algorithms, as Limited-Memory Broyden Fletcher Goldfarb Shanno (LM-BFGS) algorithm [20].

The second task is to perform the characters recognition. The track estimated in the previous steps may need to be transformed to fit the characteristic of the recognition algorithms. There is a huge variety of algorithms in the literature for this purpose. As examples, Arsalan et al. [6] transform the three-dimensional estimated track and in a two-dimensional track, which will be fed into a Neural Network to perform the recognition; Leem et al. [8] converts the point-based track obtained from the radar signal to an actual image, using image processing techniques to obtain the written character.

In this work, multiple classification algorithms have been tested on ultrasound-based gesture recognition to determine their suitability. To be precise, these Deep Learning (DL) algorithms are Convolutional Neural Networks (CNN), Long-Short Term Memory Neural Networks (LSTM NN), Autoencoders, and variations of these algorithms. These algorithms have been selected according to the high-accuracy results achieved by other authors for gesture recognition tasks [6, 8, 9].

The use of ultrasound technology mitigates the disadvantages that other technologies present. Image-based systems (cameras or IR sensors) are affected by changes in the ambient light conditions, as well as they may lead to privacy issues related to the identification of users. Radio-based systems (as WiFi or radar), thanks to intrinsic technology characteristics, could overcome these problems. These systems anyway could suffer from signal interference due to the increasing number of applications based on this technology. Finally, the use of active wearable-based systems (as Ultrasound transmitters) can lead to a more complex solution, consequently less intuitive for users.

This work is structured as follows: Section 5.1 introduces the state of the art. Section 5.2 and Section 5.3 explain in detail the algorithms studied in this work for the track acquisition and classification, respectively. Section 5.4 presents the dataset used in this work and the specific parameters used for each of the classification algorithms. Section 5.5 summarizes the results obtained. Finally, Section 5.6 focuses on conclusions of this work.

5.2. System description

This section covers the detection of the user movements and the translation into a temporal series of positions. It is divided into three tasks, covering the movement sensing, individual position calculation and the complete track estimation, following the steps shown in Fig. 5.2.

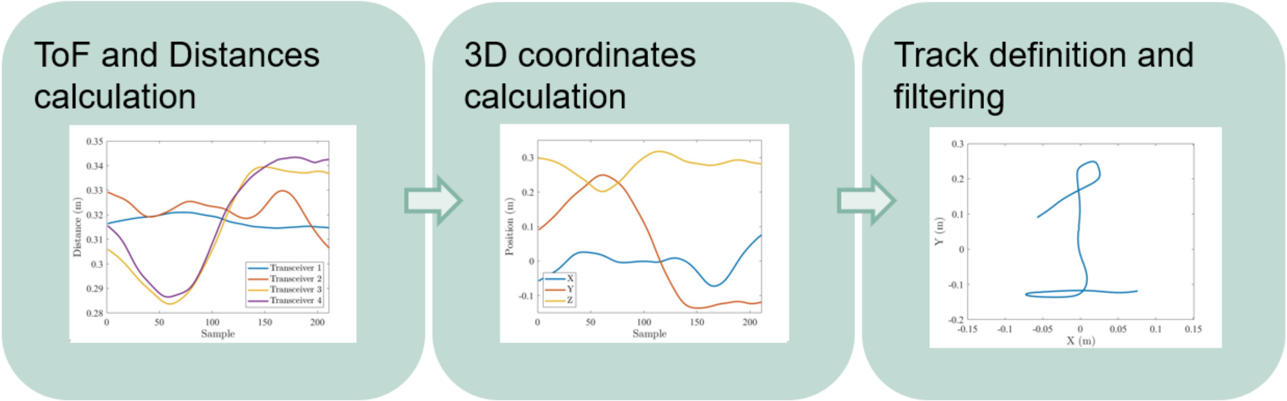


Figure 5.2: Track generation pipeline.

Parameters Setting	
Actuation pulse frequency (Fc)	30kHz
Number of pulses	6
Pulse repetition interval	20ms
Sampling frequency (Fs)	200kHz

Table 5.1: Parameters setting used for the transceivers actuation and data acquisition.

5.2.1. Hardware

For user detection, this work uses four dual-backplate MEMS microphone-based ultrasonic transceivers [21], in a squared shape matrix shown in Fig. 5.3. The transceivers need low bias voltage and support the use of both audio microphones (with a 68 dB(A) signal-to-noise (SNR) performance) and air-borne ultrasonic transceiver (with between 80 and 90 dB SNR). The use of the transceiver to emit an ultrasonic pulse produces a shadow zone of about 10 cm, due to the free oscillation of the membrane (ringing).

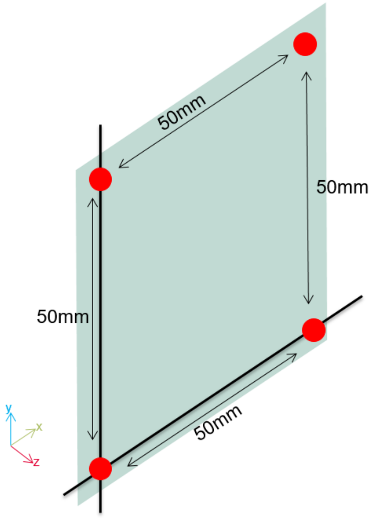


Figure 5.3: Position of the ultrasonic transceivers.

For the transceivers actuation and read-out, the Analog Discovery 2 (AD) has been used [22]. As each AD has two analog input channels, it is necessary to use two devices to acquire the output signal of the four transceivers. The AD waveform generator allows the creation of arbitrary signals, used for the transceiver actuation. The parameters used for the transceivers actuation and data acquisition are listed in Table 5.1.

5.2.1.1. Signal model/Target detection

As stated in Section 5.1, the system is based on the time elapsed between the signal emission and the echo reception, known as ToF. In the literature, numerous methods can be found based on techniques such as biologically inspired algorithms [23], based on phase difference [24] or frequency difference [25]. Most works use a method based on cross-correlation and threshold power because of the low computational power required and the noise influence removal effect of the cross-correlation, which acts as matched filtering. Using a template of the expected echo and performing the correlation with the acquired signal, this method produces a time-domain signal with a peak value when the actual echo is received [14, 26].

For this work, the ToF will be obtained through a cross-correlation algorithm and a dynamic threshold method. The target distance is then obtained from the ToF. The process can be divided into four steps [27]:

1. Cross-correlation. The acquired signal is cross-correlated with a template containing the expected echo. This method will give a maximum value in the sample where the template and the acquired signal match.

2. Dynamic Threshold. In order to distinguish whether there is an echo or not, the value of the cross-correlated signal need to be greater than a threshold level. The dynamic threshold used in this step decrease the value with the time, to match the attenuation of the signal with the distance traveled [28]. This parameter can be increased or decreased to fit certain conditions, i.e. ambient noise. The cross-correlated signal obtained in the previous step is filtered to extract the envelope, and this envelope is then evaluated to check if and where it cross the threshold level.

3. ToF calculation. All the previous calculations are done over the sample number. When the crossing point between the cross-correlation envelope and threshold is calculated, the sample can be converted to time using the ADC sampling frequency parameter.

4. Distance calculation. Once the ToF is calculated, it can be converted to distance using the following equation:

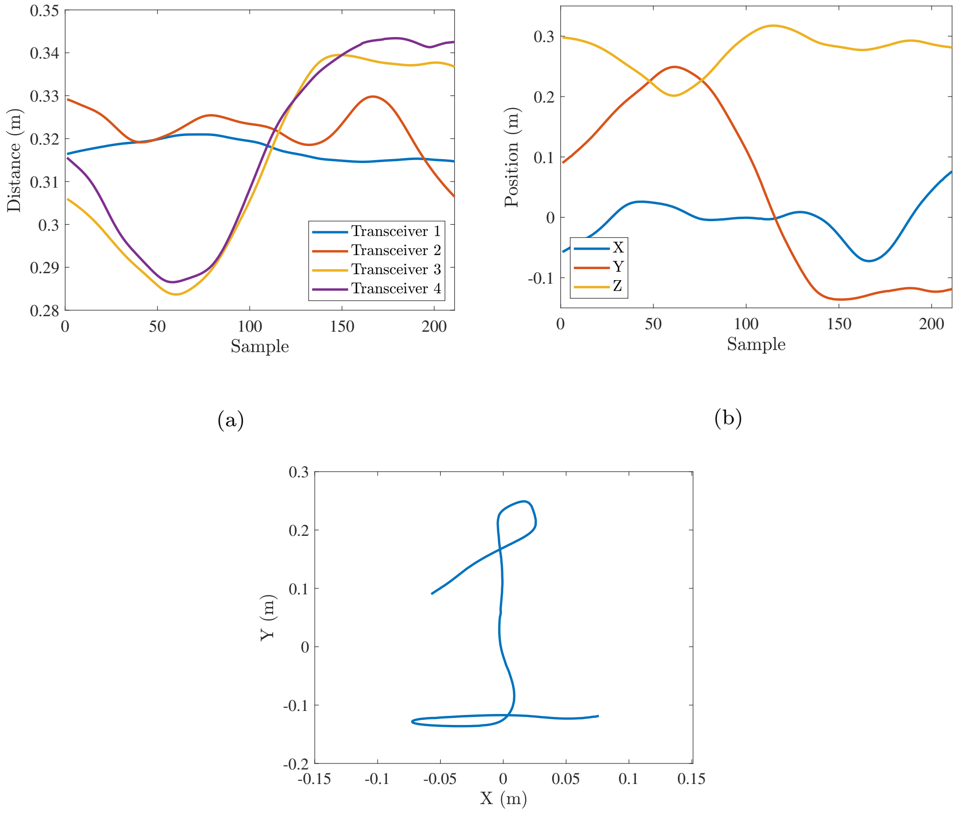
   $$d = \frac{ToF\, c_s}{2} \tag{5.1}$$

   Where d is the distance between the hand marker and the transceiver, ToF indicates the ToF calculated, and c_s the speed of sound.

5.2.1.2. Object Positioning

Once the pairwise distance between the hand marker and each anchor has been obtained as explained in the previous section, those values can be feed to the algorithm to determine the 3D space position as shown in Fig. 5.4a.

As described in Section 5.1, in this work a novel algorithm [20] will be used, instead of Multilateration. This method proposes a two-steps algorithm to obtain the hand marker location. The first step performs the anchor location, and the second step calculates the hand marker location based on the previously calculated anchors position and the pairwise distances. As the distances among the anchors are known, the first step can be repeated with a low frequency to check whether the previously calculated anchor positions are still valid or not. The hand marker position must be calculated with every new sample (each 20ms as explained in Table 5.1). The 3D position will be calculated using the LM-BFGS algorithm, minimizing the mean squared error as the objective function.



(a)

(b)

(c)

Figure 5.4: Different steps in the track estimation: Pairwise distances among transceivers and hand marker (a), 3D position time series (b), 2D projection (c)

5.2.1.3. Track definition and filtering

Based on the algorithms previously described, and performing the position estimation periodically, the trajectory of the hand marker is built as a discrete time-series of successive points, as shown in Fig. 5.4b. The knowledge of parameters like the maximum speed of the movement and the time between samples makes possible the use of filters. These filters can be used to smooth the trajectory, remove the effect of outliers and restore missing points by interpolation. In this work, the smoothing will be based on the Moving-average filter algorithm (EQ), which has been proved to reconstruct the original gesture [20].

$$y[n] = \frac{1}{M} \sum_{j=\frac{-(M-1)}{2}}^{\frac{(M-1)}{2}} x[n+j] \tag{5.2}$$

Where X is the input signal, Y is the output, and M is the window size –being M odd –.This filter behaves as a low-pass filter but is focused on time-based response instead of frequency-based. The bigger the window size (greater M), the stronger the noise reduction but the greater the delay introduced by the filter as well. Therefore, the value M has to be a trade-off to remove the noise but also to be able to respond to faster movements.[29]

5.2.1.4. Track transformations

Before using the gathered data for gesture recognition, the data has been adapted to the required format for each of the researched classification algorithms. The first step in the preprocessing pipeline was the projection of the 3D gestures into a 2D plane, as depicted in Fig. 5.4c, to generate 2D images that can be fed into the CNN, ConvLSTM and the convolutional autoencoder. The selected 2D plane was the XY plane due to the fact that the variance of the gesture respect the Z axis is much smaller than the variance in the other axis. These images have been later normalized and converted to gray-scale to reduce their dimensions while maintaining relevant features. Due to the nature of the data studied in this research, data augmentation techniques have been applied to the image as well as 3D coordinate data as described in Section 5.4.1.

5.3. Character Recognition Algorithms

Multiple Deep Neural Networks (DNN) are examined for character recognition based on the previously generated trajectory data. The trajectory can be represented as an image or as the 3D numerical coordinates depending on the data type required for each algorithm.

Using these two datasets (images and 3D coordinates), a large range of DNN models can be trained for the classification of the gestures. The most relevant DNN structures in gesture recognition have been selected to classify our data due to previous high-performance results [3, 6, 8, 9, 30, 31, 13, 32, 4, 7]. Different DNN approaches have been included in this work to also compare the effect in the classification of the two previously depicted data types. The compared algorithms are:

- **Convolutional Neural Network**. This DNN model is based on a set of convolutional filters that are applied sequentially to the input data to generate feature maps. The bias and kernel values of these filters are calculated during the training phase of the model. The features extracted with these convolutional filters are later used by fully connected layers for classification or prediction tasks as a traditional Multilayer Perceptron would do. In this work, the CNN will be used to classify the input data as one of the possible studied characters. The input data fed into the CNN are the final 2D-images where the whole characters are represented. Consequently, this DNN is trained to classify each input data individually without taking into account the time length of each character or the time distribution of the positions.

  This DNN structure has been selected according to the high accuracy results achieved in the literature for gesture recognition [3, 6, 8, 9]. These works focus on gesture data recorded with multiple sensors such as radar or WiFi. Because of this, it is desired to research if similar results can be achieved when using ultrasound data.

- **Convolutional Autoencoder**. The convolutional autoencoder can be employed to extract features from data in an unsupervised fashion. This DNN consists of two main parts: an encoder, which maps the images into an embedded representation called code, and a decoder that reconstructs the original image from the code. Therefore, the encoder and decoder can be trained by using the same data as input data and expected output. The Autoencoder can also be combined with convolutional filters for efficient data coding when a more complex feature extraction is required. As for CNN, the encoder can be the input of fully connected layers for the classification of features in different categories. The use of encoders in classification tasks can bring several benefits such as dimensionality reduction and performance improvements in supervision [30, 31, 13, 32]. As with CNN, the inputs fed into the convolutional autoencoder are the final 2D images where the whole characters are represented. Therefore, no time information is considered.

- **Long-short Term Memory (LSTM) DNN**. This DNN structure focuses on studying temporal features of the input data by studying its evolution during a selected period of time following a window approach as shown Fig. 5.5. The main characteristic of this structure lies in the fact that the output of a hidden layer is transferred, as part of the input, to the hidden layer of the next time step to preserve previous information. After temporal features are extracted, the data is transmitted to fully connected layers to perform the classification or prediction as with the previous explained models.

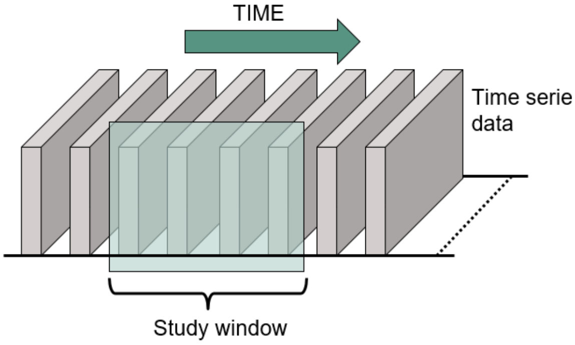


Figure 5.5: Sliding window used when studying time series with LSTM.

To extract these temporal features, the model maps the input data to a sequence of hidden parameters of the network. This leads to an output series of activation by implementing (3):

$$h(k) = \sigma(W_{hx}x(k) + h(k-1)W_{hh} + b_h) \tag{5.3}$$

Where σ is the used non-linear activation function for the DNN, $h(k)$ represents the hidden parameters of the network, $x(k)$ is the input data, b_h is the bias vector of the hidden layer and W represents the weights of the kernels. These weights can be divided into two sets of weights: input layer W_hx and hidden layers W_hh.

The input data for this model is a time serie that includes temporal features, which in our case are the 3D coordinates values. These values can be studied to extract the evolution of the movement (direction in 3 axes as well as the speed). This DNN structure has been analyzed in the literature for gesture recognition as well as trajectory prediction, due to its capabilities to extract features from movements [4, 6, 7].

- **Convolutional LSTM**. This model, often called ConvLSTM, is a variation of the previous LSTM model that includes convolutional layers. These initial convolutional layers are used to extract non-temporal features in a previous step. To do so, this structure uses convolutions to study the input data executing convolutions at each gate in the LSTM structure rather than using matrix multiplications typical of the dense layer approach in the traditional LSTM structure. Because of this, apart from time series, image series can be studied with this algorithm to extract information about the time evolution of the images. However, non-image data type inputs can also be used in case this feature extraction step is desired as in this work, where 3D coordinate data will be used as input for this model. However, since the ConvLSTM studies the time evolution of the trajectory, to ensure the length of the characters is always the same, a number of 0s have been included at the end of some samples to achieve the desired length.

This DNN structure, as the previously mentioned ones, has been selected due to the high accuracy results achieve din the literature for gesture recognition and trajectory prediction tasks [6, 33]. One of the possible reasons for its high performance results is the fact that this model can extract high level features such as movement direction before studying its time evolution, leading to a more logical feature study pipeline.

The configuration of each of these models for the specific application researched in this work, as well as a deeper description of the gathered data for this task, are presented in the next section.

5.4. Experiment definition

5.4.1. Dataset

To the best of our knowledge, there is no public dataset available. So far only Chen H. et al.[13] performed an air-writing system based on ultrasound technology, using their own dataset. That work was based on an active ultrasound array location, instead of the passive approach described in this work, making not possible the use of this database as input in our system for comparison purposes. Consequently, a new ultrasound data dataset was recorded for this experiment.

This dataset, recorded to test the proposed system, is composed by series of 3D coordinates where each of these series represents one sample of the studied gestures. These gestures are the digits “1”, “2”, “3” and “4” as well as the characters “A”, “B”, “C” and “D” as shown in Fig. 5.6. Due to the characteristics of the studied gestures, the length of the 3D coordinates series varies. The length of each gesture is in the range 5-8 seconds except for the gesture “C” that, due to the simplicity to perform it, takes between 3 and 7 seconds.

Using the previously explained time-series dataset, a 2D image dataset

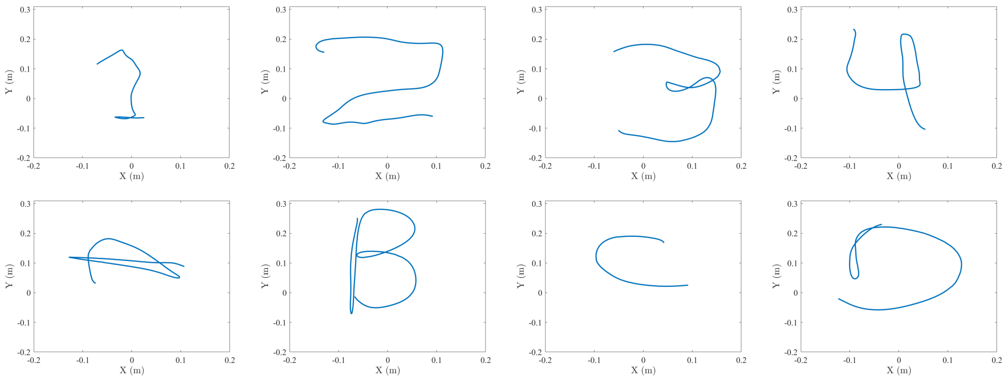


Figure 5.6: Samples of series of 3D coordinates from the recorded gestures.

has been generated. These images, of dimensions 100x100 pixels, are the projection of the gestures in the XY plane as shown in Fig. 5.7.

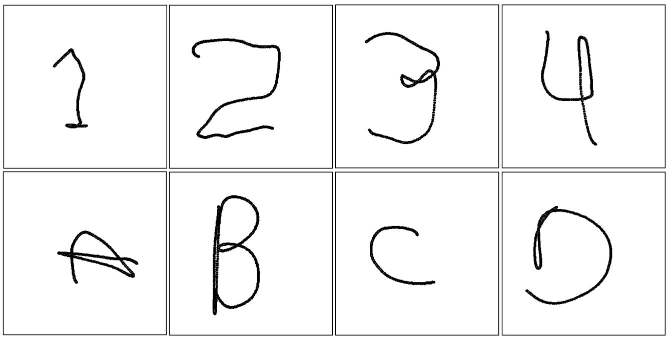


Figure 5.7: Samples of images generated from the studied gestures.

Initially, 40 samples were recorded of each possible gesture and then, multiple data augmentation techniques have been applied to generate synthetic data in order to have a large enough dataset to train the DL models. An example of the resulting images after applying this technique is shown in Fig. 5.8. The used data augmentation techniques are:

- Gesture translation(Fig. 5.8b): The center positions of the initial gestures were not constant but they were always near the center of the image. To include more positions in the dataset, all the gesture were translated so their centers are located in the center position in the XY plane. After this step, a random translation is performed in the X and Y axis or only in one of them.
- Gesture scaling (Fig. 5.8c): The gestures have been scaled within a random percentage in the interval 20 % - 50 % to generate a more variate dataset. Since the data is represented in 2 dimensions, each

time that scaling was applied, a random variable controls if the scaling was performed in one of the axis or in both as well as the scaling factor for each axis. Consequently, uniform scaled images as well as anisotropic scaled images are included in the dataset.

- Gesture rotations (Fig. 5.8d): The images have been rotated at a random angle in the interval 1°-359° to generate positions different from the original. As a result of this, all writing directions are included in the generated dataset.

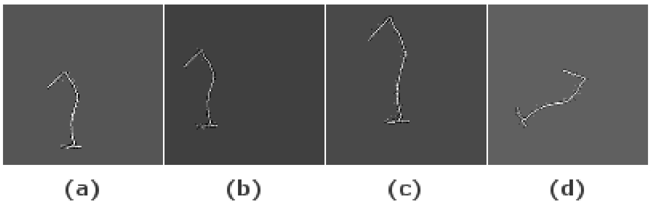


Figure 5.8: Images of original “1” image (a), translation (b), translation-scaling (c) and rotation (d).

The dataset has been augmented to 27670 samples where 5539 (including 20 % of the original samples) were used for testing purposes and 22131 for training to correctly measure the accuracy results in the test data. These test and train sub-datasets have been equally split between the image and 3D coordinate data to ensure a correct comparison of the accuracy results among the studied algorithms.

5.4.2. Deep Neural Networks configuration

Each of the researched algorithms for the classification task studied in this paper has been tuned and trained to fit the application. Therefore, their final structure and characteristics are further commented in each of the following subsections.

5.4.2.1. Convolutional Neural Network

The final structure of this model is shown in Fig. 5.9, where it is possible to observe that it has 3 convolutional layers to extract relevant features from the input data. The first layer has 32 filter of dimensions 5×5, the second layer has 64 filters of dimensions 5×5 and the final convolutional layer has 64 filters of dimensions 3×3. After these convolutional layers, a flatten layer and 3 fully connected layers (64, 32 and 8 neurons respectively) are included in the network structure to classify the features into the 8 possible gestures. Between all the layers, batch normalization layers have been included to ensure the normalization of the data is not lost during the data study. All the layers included in this network use the ReLU activation function except for the last fully connected layer which uses the softmax activation function for the final classification.

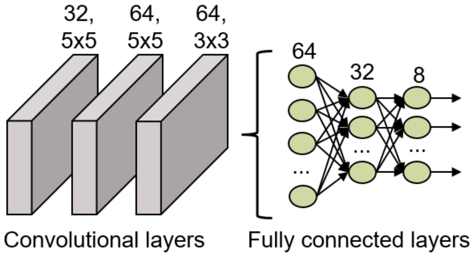


Figure 5.9: CNN structure implemented for the gesture recognition.

5.4.2.2. Convolutional Autoencoder

The training of this model takes place in two steps:

- The two main components of the autoencoder are trained to reconstruct as output, the images provided as input. The structure of this model consists of three 2D-convolutional layers for the encoder and another three identical and mirrored layers for the decoder. In the encoder, the first layer has 128 filters of dimensions 5×5, the second layer has 64 filters of dimensions 3×3 and the final convolutional layer has 32 filters of dimensions 3×3.

  The internal code layer, which provides the embedding, consists of a max 2D pooling applied on the 32 filters. All these convolutional layers use the ReLU activation function, except for the last layer of the decoder, which employs a sigmoid activation function for the nonlinear image reconstruction. The used loss function is the binary crossentropy while the optimizer is Adam. The application of the autoencoder enables the reduction of dimensionality from 10,000 (100×100) corresponding to an image, to only 1,296 (6×6×36) values, which represent the embedding space dimension.

- After the autoencoder training, the encoder part is extracted and kept frozen for training, so that it can be used as a feature extractor without further parameters tuning. A flatten layer and two dense layers consisting of 32 and 8 neurons respectively are then connected to the model. The parameters of the fully connected layer are trained so as to associate the information extracted from the encoder with the respective labels of the drawn characters. The first dense layer uses the ReLU activation function while the second one uses the softmax activation function for the categorization purpose.

The model in its ensemble (Convolutional Autoencoder and Fully Connected) is shown in Fig. 5.10.

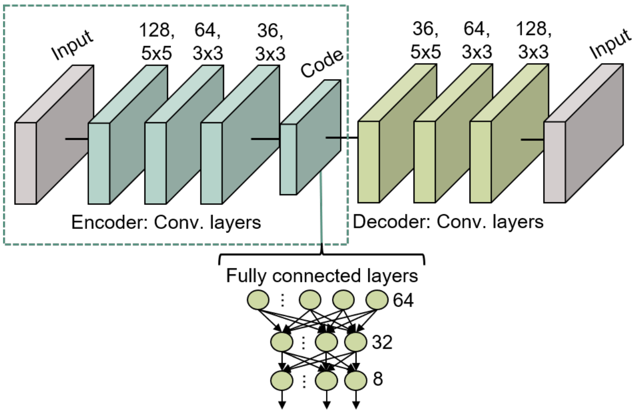


Figure 5.10: Convolutional Autoencoder structure for gesture recognition.

5.4.2.3. Long Short Term Memory Neural Network

The structure of this network is an LSTM layer with 100 units, with a time-step of size 10, followed by 3 fully connected layers (100, 40 and 8 neurons in each layer respectively) with batch normalization layers between the fully connected layers. All these fully connected layers use the ReLU activation function except for the last layer which uses the traditional softmax activation function for classification tasks. The structure can also be visualized in Fig. 5.11.

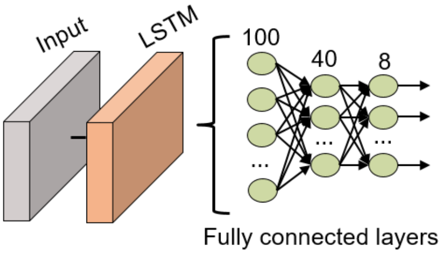


Figure 5.11: LSTM NN structure for gesture recognition

5.4.2.4. 1D-Convolutional LSTM neural network

The network structure is similar to the previously commented LSTM DNN but it includes a 1D-convolutional layer at the beginning of the network to extract relevant features that later can be studied over time as shown in Fig. 5.12. This convolutional filter has a dimension of 128×3. Following the convolutional layer, an LSTM layer with 100 units is in charge of studying the time evolution of the data and 3 fully connected layers (40, 40 and 8 neurons respectively) to classify the data. All these fully connected layers use the ReLU activation function except for the last layer which uses the traditional softmax activation function for classification tasks. Between all layers, batch normalization layers have been included to ensure the normalization of the data is not lost during the model training.

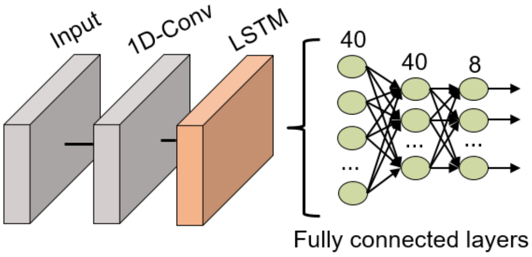


Figure 5.12: 1D-ConvLSTM NN structure for gesture recognition

5.5. Results

The results of the studied classification algorithms when using the test dataset are commented in Table 5.2. The compared parameters are the accuracy, number of parameters and the latency of the models to provide information regarding their suitability for the tasks as well as the complexity and size of the models.

Algorithm	Number of Parameters	Latency (ms)	Accuracy
CNN	1,730,472	63.43	97.39 %
ConvAutoencoder	184,396	45.50	98.28 %
LSTM	56,868	71.90	83.25 %
ConvLSTM	99,960	71.01	99.51 %

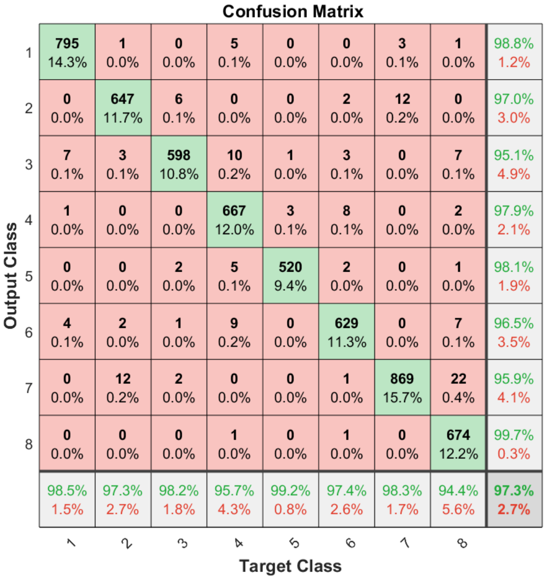
Table 5.2: Comparison of the studied classification algorithms for the gesture classification.

It is important to remark that all the latency measurements have been executed in the same device (Intel core i5 CPU) in order to be able to compare the latency results. Similarly, test datasets (images and 3D coordinates) contain the same samples.

Table 5.2 shows how the accuracy achieved by the CNN, ConvAutoencoder and ConvLSTM classification algorithms are quite similar since they all achieve an accuracy of above 97.39 % while the LSTM algorithm achieves an 83.25 %. The model that achieved the best accuracy results is the ConvLSTM model with a 99.51 %. However the CNN and ConvAutoencoder achieved similar results with a difference of a 2.02 % and 1.23 % lower accuracy respectively. The LSTM model achieved the lowest accuracy among the studied algorithms with a different respect the ConvLSTM model of a 16.26 % in the accuracy results. This may be the result of the complexity of the dataset when study as individual numerical values in a time serie in comparison with an image where the information of the gesture is easier to extract. Regarding the number of parameters and latency, the CNN model has the larger number of parameters (1,730,472 parameters), which is one or two orders of magnitude higher than the rest of the models. The LSTM model only has 56,868 parameters. This shows how there is a trade-off between the size of the model and the accuracy, apart from the network structure. However, even when the CNN model has the larger number of parameters, it still achieves latency results comparable with the rest of the models and even faster than the ConvLSTM and LSTM models. Nevertheless, the fastest model among the studied ones is the ConvAutoencoder that only requires 45.5 milliseconds to generate an output from an input data. The rest of the models require a similar time between 63.43 and 71.90 milliseconds.

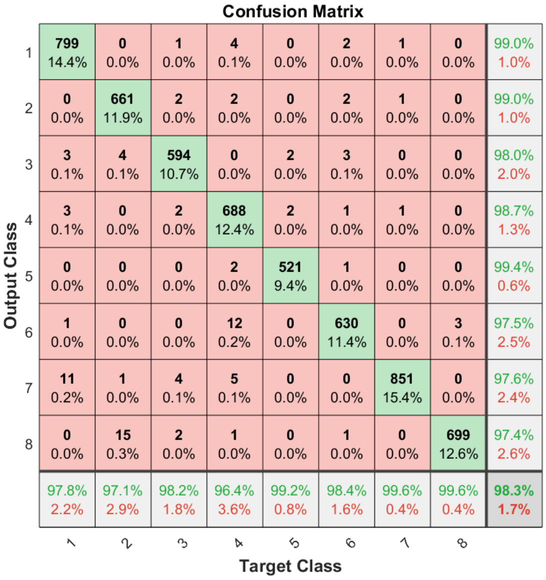
Apart from the general accuracy achieved by the studied models, the accuracy for each of the individual classes is shown in Figures 5.13, 5.14, 5.15 and 5.16. The labels in these tables have been codified where gestures “1” to “4” are represented with the labels 1 to 4 and the characters “A” to “D” with the labels 5 to 8. These figures show how the accuracy of the classes are well balanced in the studied algorithms except for LSTM, where the gesture “1” and “C” achieved an accuracy under the average. At the same time, in Fig. 5.13 we can see how most of the CNN misclassification error are located in the gestures “2”, “C” and “D”. This may be because these three gestures have similar curves that can lead the algorithm to a misclassification in some cases. In the rest of the studied algorithms, the error distribution among classes does not indicate a clear misclassification between specific classes since errors are equally distributed among all of them.

It can be observed how the ConvLSTM model has the best performance taking into account the trade-off among its latency, the number of parameters and classification accuracy (99.51 %), especially when comparing it with the LSTM model that also studies the temporal evolution of the data. As regards models that require the whole image as input, the ConvAutoencoder


Confusion Matrix
Output Class
Target Class

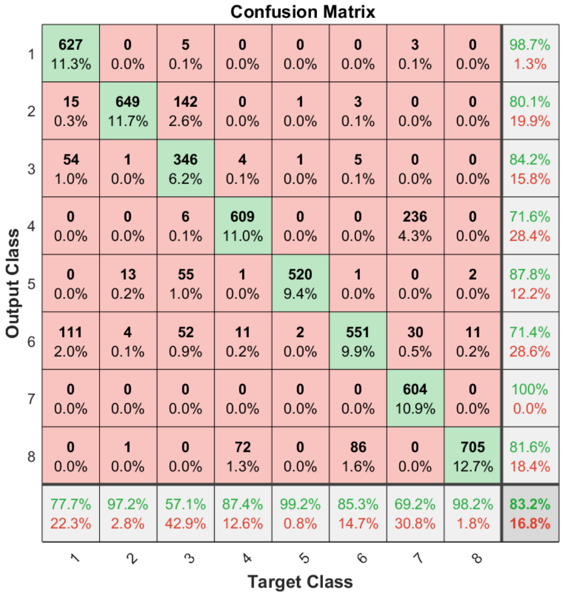
	1	2	3	4	5	6	7	8	
1	795 14.3%	1 0.0%	0 0.0%	5 0.1%	0 0.0%	0 0.0%	3 0.1%	1 0.0%	98.8% 1.2%
2	0 0.0%	647 11.7%	6 0.1%	0 0.0%	0 0.0%	2 0.0%	12 0.2%	0 0.0%	97.0% 3.0%
3	7 0.1%	3 0.1%	598 10.8%	10 0.2%	1 0.0%	3 0.1%	0 0.0%	7 0.1%	95.1% 4.9%
4	1 0.0%	0 0.0%	0 0.0%	667 12.0%	3 0.1%	8 0.1%	0 0.0%	2 0.0%	97.9% 2.1%
5	0 0.0%	0 0.0%	2 0.0%	5 0.1%	520 9.4%	2 0.0%	0 0.0%	1 0.0%	98.1% 1.9%
6	4 0.1%	2 0.0%	1 0.0%	9 0.2%	0 0.0%	629 11.3%	0 0.0%	7 0.1%	96.5% 3.5%
7	0 0.0%	12 0.2%	2 0.0%	0 0.0%	0 0.0%	1 0.0%	869 15.7%	22 0.4%	95.9% 4.1%
8	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	674 12.2%	99.7% 0.3%
	98.5% 1.5%	97.3% 2.7%	98.2% 1.8%	95.7% 4.3%	99.2% 0.8%	97.4% 2.6%	98.3% 1.7%	94.4% 5.6%	97.3% 2.7%

Figure 5.13: Confusion matrix generated using the CNN algorithm


Confusion Matrix
Output Class
Target Class

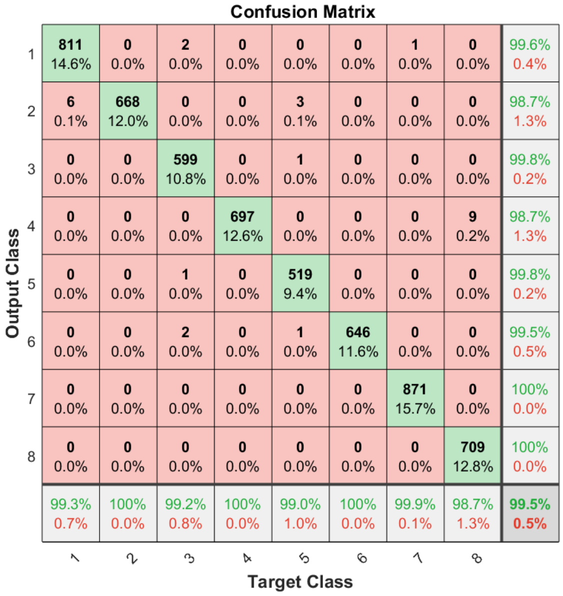
	1	2	3	4	5	6	7	8	
1	799 14.4%	0 0.0%	1 0.0%	4 0.1%	0 0.0%	2 0.0%	1 0.0%	0 0.0%	99.0% 1.0%
2	0 0.0%	661 11.9%	2 0.0%	2 0.0%	0 0.0%	2 0.0%	1 0.0%	0 0.0%	99.0% 1.0%
3	3 0.1%	4 0.1%	594 10.7%	0 0.0%	2 0.0%	3 0.1%	0 0.0%	0 0.0%	98.0% 2.0%
4	3 0.1%	0 0.0%	2 0.0%	688 12.4%	2 0.0%	1 0.0%	1 0.0%	0 0.0%	98.7% 1.3%
5	0 0.0%	0 0.0%	0 0.0%	2 0.0%	521 9.4%	1 0.0%	0 0.0%	0 0.0%	99.4% 0.6%
6	1 0.0%	0 0.0%	0 0.0%	12 0.2%	0 0.0%	630 11.4%	0 0.0%	3 0.1%	97.5% 2.5%
7	11 0.2%	1 0.0%	4 0.1%	5 0.1%	0 0.0%	0 0.0%	851 15.4%	0 0.0%	97.6% 2.4%
8	0 0.0%	15 0.3%	2 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	699 12.6%	97.4% 2.6%
	97.8% 2.2%	97.1% 2.9%	98.2% 1.8%	96.4% 3.6%	99.2% 0.8%	98.4% 1.6%	99.6% 0.4%	99.6% 0.4%	98.3% 1.7%

Figure 5.14: Confusion matrix generated using the ConvAutoencoder algorithm



	1	2	3	4	5	6	7	8	
1	627 11.3%	0 0.0%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	3 0.1%	0 0.0%	98.7% 1.3%
2	15 0.3%	649 11.7%	142 2.6%	0 0.0%	1 0.0%	3 0.1%	0 0.0%	0 0.0%	80.1% 19.9%
3	54 1.0%	1 0.0%	346 6.2%	4 0.1%	1 0.0%	5 0.1%	0 0.0%	0 0.0%	84.2% 15.8%
4	0 0.0%	0 0.0%	6 0.1%	609 11.0%	0 0.0%	0 0.0%	236 4.3%	0 0.0%	71.6% 28.4%
5	0 0.0%	13 0.2%	55 1.0%	1 0.0%	520 9.4%	1 0.0%	0 0.0%	2 0.0%	87.8% 12.2%
6	111 2.0%	4 0.1%	52 0.9%	11 0.2%	2 0.0%	551 9.9%	30 0.5%	11 0.2%	71.4% 28.6%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	604 10.9%	0 0.0%	100% 0.0%
8	0 0.0%	1 0.0%	0 0.0%	72 1.3%	0 0.0%	86 1.6%	0 0.0%	705 12.7%	81.6% 18.4%
	77.7% 22.3%	97.2% 2.8%	57.1% 42.9%	87.4% 12.6%	99.2% 0.8%	85.3% 14.7%	69.2% 30.8%	98.2% 1.8%	83.2% 16.8%

Figure 5.15: Confusion matrix generated using the LSTM algorithm



	1	2	3	4	5	6	7	8	
1	811 14.6%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	99.6% 0.4%
2	6 0.1%	668 12.0%	0 0.0%	0 0.0%	3 0.1%	0 0.0%	0 0.0%	0 0.0%	98.7% 1.3%
3	0 0.0%	0 0.0%	599 10.8%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
4	0 0.0%	0 0.0%	0 0.0%	697 12.6%	0 0.0%	0 0.0%	0 0.0%	9 0.2%	98.7% 1.3%
5	0 0.0%	0 0.0%	1 0.0%	0 0.0%	519 9.4%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
6	0 0.0%	0 0.0%	2 0.0%	0 0.0%	1 0.0%	646 11.6%	0 0.0%	0 0.0%	99.5% 0.5%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	871 15.7%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	709 12.8%	100% 0.0%
	99.3% 0.7%	100% 0.0%	99.2% 0.8%	100% 0.0%	99.0% 1.0%	100% 0.0%	99.9% 0.1%	98.7% 1.3%	99.5% 0.5%

Figure 5.16: Confusion matrix generated using the ConvLSTM algorithm

provides better results than the CNN for what regards latency, number of parameters and accuracy. This might be due to the efficient compression of relevant features performed by the ConvAutoencoder that leads to a better classification accuracy.

After the comparison of the performance of the studied classification algorithms, a comparison of our results with other technologies and authors is shown in Table 5.3. This table shows how multiple technologies are being tested for air writing tasks. Even when different technologies are used, such as radar, ultra-wide-band and ultrasound, the target detection can be calculated based on similar techniques. This is since these technologies calculate the position of the target by measuring the time difference between the transmission of a signal and the reception of its echo. Consequently, the classification algorithms are compared in Table 5.3 rather than the technique for the data gathering. At the same time, it is important to remark that, since the technologies, gesture number and platform are different, this comparison should be understood as a general observation to get a deeper understanding regarding the state of the art rather than a direct comparison among techniques.

Studies	**No. of Characters**	**Accuracy**	**Latency (ms)**	**Method**	**Hardware**
ORM Ultrasound [13]	26	96.31 %	1.8	Order-restricted matching (ORM) classifier	2 ultrasound arrays
Radar DNN [6]	15	98.33 %	–	ConvLSTM-CTC	3 radars
Radio CNN [8]	10	99.7 %	52.2	CNN	3 radars
This work	8	99.51 %	71.01	ConvLSTM	1 ultrasound array
This work	8	98.28 %	45.5	ConvAutoencoder	1 ultrasound array

Table 5.3: Comparison of state-of-the-art techniques for air writing.

Among the compared techniques, the most popular are the DL techniques such as CNN or LSTM due to their high-performance results for classification tasks. Only one of the compared techniques is based on a different approach, the Order-Restricted Matching (ORM) classification algorithm. This algorithm, differently from the rest of the algorithms, is not trained in advance but features are extracted and later compared directly with a feature template for each of the possible characters. The sequence with the minimum accumulated distance between the features and the template feature is selected as the classification result.

However, even if different technologies and approaches are used for this task, similar accuracy results are achieved. All the compared techniques have an accuracy between 96.31 % and 99.7 %. Among these techniques, the one that provides the highest accuracy results is the Radio CNN (99.7 %). Nevertheless, if the latency of the system is taken into account, the ORM Ultrasound technique may provide a best performance since the accuracy difference compared to the Radio CNN is 3.39 % but it achieves a latency reduction by a factor of 29 times. Our studied ConvAutoencoder could be considered as a middle point between these two extreme cases since it achieves an accuracy of 98.28 %, higher than the ORM technique, and a latency of 45.5 ms, 7.7 ms faster than the Radar DNN.

Another feature that can compared is the devices integrated into these systems. The techniques based on radar sensors require at least 3 sensors in order to locate the target in 3 dimensions. Each of these radar sensors includes a different number of transmitter and receiver antennas, i.e. each of the radars integrated in the system in [6] uses 1 receiver and 1 transmitter antenna. In the case of ORM Ultrasound, 2 arrays of ultrasound sensors are required. The sensors of the first array are used exclusively for transmitting while the ones from the second array are used to receive the echo signals. On the other hand, the system presented in this work only requires 1 array where 1 ultrasound transceiver is used to transmit and receive while the other 3 transceivers are only used for receiving. As a result of this, a reduce number of devices are required in comparison with the rest of the compared techniques while maintaining similar high performance results.

5.6. Conclusion

An Air-Writing system, based on one ultrasonic array that includes 4 transceivers has been presented in this work. The system determines the point-to-point distance to the target from the ToF. Those distances calculated are fed to the positioning algorithm to extract the 3D position of the target, and determine the trajectory as a successive series of points equally spaced in time.

To test this system, a dataset containing 8 gestures (4 letters and 4 numbers) has been recorded. This raw data has later been filtered and preprocessed to generate a dataset for gesture classification. Multiple algorithms have been researched in this paper to study this dataset. Since the original data was a time-series of 3D coordinates, 2 different approaches have been studied to analyze the data: time evolution algorithms (LSTM and ConvLSTM) and image classification algorithms (CNN and ConvAutoencoder).

It has been shown how these algorithms provided high accuracy, where the best result is extracted from the ConvLSTM, with a 99.51 % accuracy and 71.01 milliseconds of latency, when studying time-series of 3D coordinates. Among the algorithms based on images, the ConvAutoencoder provided the best results with a latency of 45.50 milliseconds and an accuracy of

98.28 %. Consequently, we can conclude that the proposed system could be implemented in multiple ways so the recognition algorithm can fit the desire platform/scenario.

References

[1] Mingyu Chen, Ghassan AlRegib, and Biing-Hwang Juang. Air-writing recognition—part i: Modeling and recognition of characters, words, and connecting motions. *IEEE Transactions on Human-Machine Systems*, 46(3):403–413, 2015.

[2] Mingyu Chen, Ghassan AlRegib, and Biing-Hwang Juang. Air-writing recognition—part ii: Detection and recognition of writing activity in continuous stream of motion data. *IEEE Transactions on Human-Machine Systems*, 46(3):436–444, 2015.

[3] Sohom Mukherjee, Sk Arif Ahmed, Debi Prosad Dogra, Samarjit Kar, and Partha Pratim Roy. Fingertip detection and tracking for recognition of air-writing in videos. *Expert Systems with Applications*, 136:217–229, 2019.

[4] Pradeep Kumar, Rajkumar Saini, Santosh Kumar Behera, Debi Prosad Dogra, and Partha Pratim Roy. Real-time recognition of sign language gestures and air-writing using leap motion. In *2017 Fifteenth IAPR international conference on machine vision applications (MVA)*, pages 157–160. IEEE, 2017.

[5] Najeed Ahmed Khan, Shariq Mahmood Khan, Maria Abdullah, Sana Jamaluddin Kanji, and Urooj Iltifat. Use hand gesture to write in air recognize with computer vision. *IJCSNS*, 17(5):51, 2017.

[6] Muhammad Arsalan and Avik Santra. Character recognition in air-writing based on network of radars for human-machine interface. *IEEE Sensors Journal*, 19(19):8855–8864, 2019.

[7] Pengcheng Wang, Junyang Lin, Fuyue Wang, Jianping Xiu, Yue Lin, Na Yan, and Hongtao Xu. A gesture air-writing tracking method that uses 24 ghz simo radar soc. *IEEE Access*, 8:152728–152741, 2020.

[8] Seong Kyu Leem, Faheem Khan, and Sung Ho Cho. Detecting mid-air gestures for digit writing with radio sensors and a cnn. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1066–1081, 2019.

[9] Yinyin Fang, Yong Xu, Heju Li, Xin He, and Longlong Kang. Writing in the air: Recognize letters using deep learning through wifi signals. In

2020 6th International Conference on Big Data Computing and Communications (BIGCOM), pages 8–14, 2020.

[10] Haoyu Wang and Wei Gong. Rf-pen: Practical real-time rfid tracking in the air. *IEEE Transactions on Mobile Computing*, pages 1–1, 2020.

[11] Xin Zhang, Zhichao Ye, Lianwen Jin, Ziyong Feng, and Shaojie Xu. A new writing experience: Finger writing in the air using a kinect sensor. *IEEE MultiMedia*, 20(4):85–93, 2013.

[12] Ziyong Feng, Shaojie Xu, Xin Zhang, Lianwen Jin, Zhichao Ye, and Weixin Yang. Real-time fingertip tracking and detection using kinect depth sensor for a new writing-in-the air system. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service*, pages 70–74, 2012.

[13] Hui Chen, Tarig Ballal, Ali Hussein Muqaibel, Xiangliang Zhang, and Tareq Y Al-Naffouri. Air writing via receiver array-based ultrasonic source localization. *IEEE Transactions on Instrumentation and Measurement*, 69(10):8088–8101, 2020.

[14] Jian Chen, Fan Yu, Jiaxin Yu, and Lin Lin. A three-dimensional pen-like ultrasonic positioning system based on quasi-spherical pvdf ultrasonic transmitter. *IEEE Sensors Journal*, 21(2):1756–1763, 2020.

[15] Sverre Holm. Ultrasound positioning based on time-of-flight and signal strength. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2012.

[16] Amit Kumar and James McNames. Wideband acoustic positioning with precision calibration and joint parameter estimation. *IEEE Transactions on Instrumentation and Measurement*, 66(8):1946–1953, 2017.

[17] Mohamed M Saad, Chris J Bleakley, Tarig Ballal, and Simon Dobson. High-accuracy reference-free ultrasonic location estimation. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1561–1570, 2012.

[18] Tarig Ballal Khidir Ahmed. *Angle-of-Arrival Based Ultrasonic 3-D Location for Ubiquitous Computing.* PhD thesis, University College Dublin, 2010.

[19] Daniel Ruiz, Jesús Ureña, Juan C. García, Carmen Pérez, José M. Villadangos, and Enrique García. Efficient trilateration algorithm using time differences of arrival. volume 193, pages 220–232. Elsevier, 2013.

[20] Borja Saez-Mingorance, Antonio Escobar-Molero, Javier Mendez-Gomez, Encarnacion Castillo-Morales, and Diego P. Morales-Santos.

Object positioning algorithm based on multidimensional scaling and optimization for synthetic gesture data generation. *Sensors*, 21(17), 2021.

[21] Sebastian Anzinger, Christian Bretthauer, Johannes Manz, Ulrich Krumbein, and Alfons Dehé. Broadband acoustical mems transceivers for simultaneous range finding and microphone applications. *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, pages 865–868, 2019.

[22] Digilent. Analog discovery 2. `https://digilent.com/reference/test-and-measurement/analog-discovery-2/`. Accessed: 2021-08-31.

[23] G Hayward, F Devaud, and JJ Soraghan. P1g-3 evaluation of a bio-inspired range finding algorithm (bira). *2006 IEEE Ultrasonics Symposium*, pages 1381–1384, 2006.

[24] Ke-Nung Huang and Yu-Pei Huang. Multiple-frequency ultrasonic distance measurement using direct digital frequency synthesizers. *Sensors and Actuators A: Physical*, 149(1):42–50, 2009.

[25] David MJ Cowell and Steven Freear. Separation of overlapping linear frequency modulated (lfm) signals using the fractional fourier transform. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 57(10):2324–2333, 2010.

[26] J. C. Jackson, R. Summan, G. I. Dobie, S. M. Whiteley, S. G. Pierce, and G. Hayward. Time-of-flight measurement techniques for airborne ultrasonic ranging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 60(2):343–355, 2013.

[27] Borja Saez, Javier Mendez, Miguel Molina, Encarnación Castillo, Manuel Pegalajar, and Diego P Morales. Gesture recognition with ultrasounds and edge computing. *IEEE Access*, 9:38999–39008, 2021.

[28] Henry E Bass, Louis C Sutherland, Allen J Zuckerwar, David T Blackstock, and DM Hester. Atmospheric absorption of sound: Further developments. *The Journal of the Acoustical Society of America*, 97(1):680–683, 1995.

[29] José Luis Guiñón, Emma Ortega, José García-Antón, and Valentín Pérez-Herranz. Moving average and savitzki-golay smoothing filters using mathcad. volume 2007, 2007.

[30] Yasi Wang, Hongxun Yao, Sicheng Zhao, and Ying Zheng. Dimensionality reduction strategy based on auto-encoder. In *Proceedings of the*

7th International Conference on Internet Multimedia Computing and Service, pages 1–4, 2015.

[31] Dimitris Perdios, Adrien Besson, Marcel Arditi, and Jean-Philippe Thiran. A deep learning approach to ultrasound image recovery. In *2017 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. Ieee, 2017.

[32] Onur Karaoğlu, Hasan Şakir Bilge, and İhsan Uluer. Reducing speckle noise from ultrasound images using an autoencoder network. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2020.

[33] Ue-Hwan Kim, Yewon Hwang, Sun-Kyung Lee, and Jong-Hwan Kim. Writing in the air: Unconstrained text recognition from finger movement using spatio-temporal convolution. *arXiv preprint arXiv:2104.09021*, 2021.

Part III

Conclusions

Chapter 6

Conclusions

Those are my principles, and if you don't like them... well, I have others.

Groucho Marx

This research has focused on the design and optimization of algorithms to perform gesture recognition based on ultrasound technology. The aim and focusing during the research is to use edge devices as system developing devices. The main conclusions obtained on this research are listed below:

- The use of Edge Devices to perform the signal processing and recognition implies the necessity of optimum resource usage and timing. It is essential to be able to increase the amount of data acquired, minimizing the loss of information that can be relevant for recognition. By using optimized mathematical functions, and removing the Analog-Digital Converter (ADC) control from the main core, it has been possible to perform the ToF calculation with the XMC with a rate of 30 positions per second with up to 2 transceivers.
- When increasing the number of transceivers, therefore the number of ToF measurements, the complexity of positioning the obstacle increases as well. Traditional positioning algorithms can be slow and require a heavy amount of resources in that case. Furthermore, knowing the exact position of the transceivers is fundamental for those algorithms. Therefore, new algorithms are needed for allowing a higher amount of transceivers. With the use of multidimensional scaling algorithms(MDS) presented in Chapter 4, both problems are solved. This algorithm presents a high speed for obtaining the position of the obstacle, and can also define the position of the transceiver.
- An important problem while tracking the movement of a target is erroneous positions due to mistakes either on ToF calculation or by

the positioning algorithms. For its mitigation, in Chapter 4 different techniques have been tested and customized for the ultrasound scenario. Based on low-pass filtering techniques, those approaches decrease the error in the final track estimation, removing as well the outliers'positions which could affect the track.

- As the transceivers used in this work were in an early development stage, it was difficult to get access to real data. To avoid this bottleneck and be able to keep the research, the framework explained in Chapter 4 was necessary. It allows generating gesture data using the same algorithms that will be used afterward to generate the real data. That way it is possible to keep researching further steps, as to how the transceivers'position affects the recognition, filtering techniques, or recognition algorithms.

- Applying the knowledge acquired during the research, it has been possible to design and test an air-writing system based only on Ultrasonic signals, not using any supplementary sensor technology. Air-writing is a particular case of gesture recognition. As it can be found in the literature based on many other sensing technologies, it is a good application to perform a comparison among those technologies. To the best of our knowledge, the publication presented in Chapter 5 is the first publication developing such a system based only on ultrasonic waves.

- This work shows also the versatility of algorithms. Algorithms well known in some fields can be exploited in fields a priory quite different. In Chapter 4 algorithms are used for data representation and data clustering. In Chapter 5 above the positioning algorithms, the recognition has been performed with techniques used in the image recognition field. That means adding a pre-processing layer (i.e. calculation of ToF or convert position vector into an image) allows the use of algorithms developed for other objectives, being able to obtain results without needing to develop new algorithms. This idea expedites the comparison between different technologies.

6.1. Future trends

To conclude, based on the experience gather during this research, we foresee three main trends in the HSI and Ultrasound topic:

- The HSI will keep evolving to mimetic human communication. This implies the input systems will change from single input commands (as key pressing, voice commands, single gestures) to an interpretation of several commands. In other words, HSI will become a multi-modal system, being able not only to recognize what the user is giving as input but also to modify the meaning of this input based on the context where it has been provided.

- Derived from the previous point, the HSI will be based on several technologies, instead of the single technology system in use nowadays. Topics like sensor fusion will become more important in the future in the HSI definition and development. On the other hand, this will imply hard work on privacy and security to break the acceptance barrier of such devices.

- Beyond HSI and medical devices, ultrasound technology can and will be used for way more applications. As the interest for data and measurements is increasing due to the possibilities brought by Artificial Intelligence techniques, ultrasonic waves can be used for new applications in automotive, machinery early maintenance, or material testing fields.

References

[1] Hervé Abdi. Metric multidimensional scaling (mds): analyzing distance matrices. *Encyclopedia of measurement and statistics*, pages 1–13, 2007.

[2] Paolo Annibale, Jason Filos, Patrick A Naylor, and Rudolf Rabenstein. Tdoa-based speed of sound estimation for air temperature and room geometry inference. *IEEE transactions on audio, speech, and language processing*, 21(2):234–246, 2012.

[3] Sebastian Anzinger, Christian Bretthauer, Johannes Manz, Ulrich Krumbein, and Alfons Dehé. Broadband acoustical mems transceivers for simultaneous range finding and microphone applications. *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII (TRANSDUCERS & EUROSENSORS XXXIII)*, pages 865–868, 2019.

[4] Dennis A Bohn. Environmental effects on the speed of sound. *Journal of the audio engineering society, Audio Engineering Society Convention 83*, 1987.

[5] Aladin Carovac, Fahrudin Smajlovic, and Dzelaludin Junuzovic. Application of ultrasound in medicine. *Acta Informatica Medica*, 19(3):168, 2011.

[6] Jian Chen, Fan Yu, Jianxin Yu, and Lin Lin. A three-dimensional pen-like ultrasonic positioning system based on quasi-spherical pvdf ultrasonic transmitter. *IEEE Sensors Journal*, 2020.

[7] Maximo Cobos, Fabio Antonacci, Anastasios Alexandridis, Athanasios Mouchtaris, and Bowon Lee. A survey of sound source localization methods in wireless acoustic sensor networks. *Wireless Communications and Mobile Computing*, 2017, 2017.

[8] David MJ Cowell and Steven Freear. Separation of overlapping linear frequency modulated (lfm) signals using the fractional fourier transform. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 57(10):2324–2333, 2010.

[9] Tobias Dahl, Joao L Ealo, Hans J Bang, Sverre Holm, and Pierre Khuri-Yakub. Applications of airborne ultrasound in human–computer interaction. *Ultrasonics*, 54(7):1912–1921, 2014.

[10] Amit Das, Ivan Tashev, and Shoaib Mohammed. Ultrasound based gesture recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 406–410, 2017.

[11] Jan De Leeuw and Patrick Mair. Multidimensional scaling using majorization: Smacof in r. 2011.

[12] Vin De Silva and Joshua B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University, 2004.

[13] W. Yu et al. A Survey on the Edge Computing for the Internet of Things. *IEEE Access*, 6:6900–6919, 2018.

[14] Y. Gao, M. A. Maraci, and J. A. Noble. Describing ultrasound video content using deep convolutional neural networks. *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 787–790, 2016. doi: 10.1109/ISBI.2016.7493384.

[15] David G. Gobbi, Roch M. Comeau, and Terry M. Peters. Ultrasound probe tracking for real-time ultrasound/mri overlay and visualization of brain shift. pages 920–927, 1999.

[16] José Luis Guiñón, Emma Ortega, José García-Antón, and Valentín Pérez-Herranz. Moving average and savitzki-golay smoothing filters using mathcad. *Papers ICEE*, 2007, 2007.

[17] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: using the doppler effect to sense gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1911–1914, 2012.

[18] Chris Hand. A survey of 3d interaction techniques. 16(5):269–281, 1997.

[19] G Hayward, F Devaud, and JJ Soraghan. P1g-3 evaluation of a bio-inspired range finding algorithm (bira). *2006 IEEE Ultrasonics Symposium*, pages 1381–1384, 2006.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[21] Ke-Nung Huang and Yu-Pei Huang. Multiple-frequency ultrasonic distance measurement using direct digital frequency synthesizers. *Sensors and Actuators A: Physical*, 149(1):42–50, 2009.

[22] J. C. Jackson, R. Summan, G. I. Dobie, S. M. Whiteley, S. G. Pierce, and G. Hayward. Time-of-flight measurement techniques for airborne ultrasonic ranging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 60(2):343–355, 2013.

[23] Young Jeon, Taehong Kim, and Taejoon Kim. Fast and robust time synchronization with median kalman filtering for mobile ad-hoc networks. *Sensors*, 21(2):590, 2021.

[24] Taavi Laadung, Sander Ulp, Muhammad M. Alam, and Yannick Le Moullec. Active-passive two-way ranging using uwb. pages 1–5, 2020. doi: 10.1109/ICSPCS50536.2020.9309999.

[25] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. pages 99–112, 2012.

[26] John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on robotics and Automation*, 7(3):376–382, 1991.

[27] Jacques Lewiner. Paul langevin and the birth of ultrasonics. *Japanese Journal of Applied Physics*, 30(S1):5, jan 1991. doi: 10.7567/jjaps.30s1.5. URL `https://doi.org/10.7567%2Fjjaps.30s1.5`.

[28] Xinrong Li. Collaborative localization with received-signal strength in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 56(6):3807–3817, 2007.

[29] Alejandro Lindo, Enrique Garcia, Jesus Urena, Maria del Carmen, and Alvaro Hernandez. Multiband waveform design for an ultrasonic indoor positioning system. *IEEE Sensors Journal*, 15(12):7190–7199, 2015.

[30] Kang Ling, Haipeng Dai, Yuntang Liu, and Alex X Liu. Ultragesture: Fine-grained gesture sensing and recognition. *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2018.

[31] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[32] J. M. Martın, A. R. Jiménez, F. Seco, L. Calderón, Jose L. Pons, and R. Ceres. Estimating the 3d-position from time delay data of us-waves: experimental analysis and a new processing algorithm. *Sensors and Actuators A: Physical*, 101(3):311–321, 2002.

[33] Carlos Medina, José Carlos Segura, and Angel De la Torre. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, 13(3):3501–3526, 2013.

[34] Aryan Mokhtari and Alejandro Ribeiro. Global convergence of online limited memory bfgs. *The Journal of Machine Learning Research*, 16 (1):3151–3181, 2015.

[35] José L. Morales. A numerical study of limited memory bfgs methods. *Applied Mathematics Letters*, 15(4):481–487, 2002.

[36] Abdelmoumen Norrdine. An algebraic solution to the multilateration problem. 1315, 2012.

[37] Dinesh Dash Partha Pratim Ray and Debashis. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. *Network and Computer Applications*, 140:1–22, 2019.

[38] Matti Pastell, Lilli Frondelius, Mikko Järvinen, and Juha Backman. Filtering methods to improve the accuracy of indoor positioning data for dairy cows. *Biosystems Engineering*, 169:22–31, 2018.

[39] Marco Patanè, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Wireless sensor networks localization with outliers and structured missing data. pages 1–7, 2017.

[40] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. *Multilayer Perceptron: Architecture Optimization and Training with Mixed Activation Functions*. BDCA'17. Association for Computing Machinery, New York, NY, USA, 2017. ISBN 9781450348522. doi: 10.1145/3090354.3090427. URL `https://doi.org/10.1145/3090354.3090427`.

[41] Antonin Povolny, Hiroshige Kikura, and Tomonori Ihara. Ultrasound pulse-echo coupled with a tracking technique for simultaneous measurement of multiple bubbles. *Sensors*, 18(5):1327, 2018.

[42] Veronika Putz, Julia Mayer, Harald Fenzl, Richard Schmidt, Markus Pichler-Scheder, and Christian Kastl. Cyber–Physical Mobile Arm Gesture Recognition using Ultrasound and Motion Data. 2020.

[43] J. Cao Q. Zhang Y. Li W. Shi and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[44] Jun Qi and Guo-Ping Liu. A robust high-accuracy ultrasound indoor positioning system based on a wireless sensor network. *Sensors*, 17(11): 2554, 2017.

[45] Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin, and Zhang Sanfeng. Dolphin: Ultrasonic-based gesture recognition on smartphone platform. *2014 IEEE 17th International Conference on Computational Science and Engineering*, pages 1461–1468, 2014.

[46] Sara Casado-Vara Roberto Sittón-Candanedo Inés Alonso Ricardo Corchado Rodríguez, Juan Rodríguez. A Review of Edge Computing Reference Architectures and a new Global Edge Proposal. *Future Generation Computer Systems*, 2019.

[47] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pages 474–485, 2016.

[48] Daniel Ruiz, Jesús Ureña, Juan C. García, Carmen Pérez, José M. Villadangos, and Enrique García. Efficient trilateration algorithm using time differences of arrival. *Sensors and Actuators A: Physical*, 193:220–232, 2013.

[49] T. A. Mohammed S. Albawi and S. Al-Zawi. Understanding of a convolutional neural network. *International Conference on Engineering and Technology (ICET), Antalya, 2017*, pages 1–6, 2017.

[50] Nasir Saeed, Haewoon Nam, Mian I. U.l Haq, and Dost B. Muhammad Saqib. A survey on multidimensional scaling. *ACM Computing Surveys (CSUR)*, 51(3):1–25, 2018.

[51] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[52] Yu Sang, Laixi Shi, and Yimin Liu. Micro hand gesture recognition system using ultrasonic active sensing. *IEEE Access*, 6:49339–49347, 2018.

[53] M. Satyanarayanan. The emergence of edge computing. *Computer*, 50 (1):30–39, 2017.

[54] Maurice G Silk. *Ultrasonic transducers for nondestructive testing.* Adam Hilger Ltd., Accord, MA, 1984.

[55] Deepti Singhal and Rama M. Garimella. Simple median based information fusion in wireless sensor network. pages 1–7, 2012.

[56] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the cricket location system. pages 190–202, 2004.

[57] Michael W. Trosset and Carey E. Priebe. The out-of-sample problem for classical multidimensional scaling. *Computational statistics & data analysis*, 52(10):4635–4642, 2008.

[58] General Vision. Neuroshield. (accessed: 20.04.2020). URL `https://www.general-vision.com/hardware/neuroshield/`.

[59] S. Elanayar V.T. and Y. C. Shin. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transactions on Neural Networks*, 5(4):594–603, 1994.

[60] Shuxia Wang. Wireless network indoor positioning method using non-metric multidimensional scaling and rssi in the internet of things environment. *Mathematical Problems in Engineering*, 2020, 2020.

[61] Christopher R. Wren, Ali Azarbayejani, Trevor Darrell, and Alex P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785, 1997.

[62] Ling Xiao, Renfa Li, and Juan Luo. Sensor localization based on non-metric multidimensional scaling. *STRESS*, 2(1), 2006.

[63] Costas Yiallourides and Pablo P. Parada. Low power ultrasonic gesture recognition for mobile handsets. pages 2697–2701, 2019. doi: 10.1109/ICASSP.2019.8683781.

[64] Qinglin Zeng, Zheng Kuang, Shuaibing Wu, and Jun Yang. A method of ultrasonic finger gesture recognition based on the micro-doppler effect. *Applied Sciences*, 9(11):2314, 2019.