

Development of a device for neurogenic bladder control

Sara Pardilla Vidal



FEUP

Master Thesis in Biomedical Engineering

Supervisor: Dulce Alves de Oliveira

Co-supervisor: Rui Prado Costa

15th July 2024

Abstract

Neurogenic bladder (NB) is a medical condition where the nervous system does not communicate properly with the bladder. Patients suffering from NB have problems coordinating filling and emptying actions of the bladder. There are several causes of this disease, such as spinal cord injuries, neurological diseases, diabetes, infections, etc.

Today, the most commonly used treatment for this condition is catheterization, which is an invasive technique that effectively drains urine from the bladder, but can lead to urological complications. To reduce these complications, a professional establishes a drainage schedule for the patient, estimating the times at which the bladder will be full.

However, bladder filling is not always continuous, as it depends on various physiological and external factors. Thus, to enhance the accuracy of programming bladder emptying and significantly improve patients' quality of life, a monitoring device is proposed capable of real-time tracking of bladder filling levels.

Thus, the aim of this thesis is to design a non-invasive, portable and ergonomic device enabling accurate bladder volume readings, facilitating optimal self-catheterization for patients. This is a follow-up project since the device, called SafeBladder, had already begun to be developed. The initial stages involved selecting the technology for measuring urine volume, with infrared spectroscopy being the chosen method. Subsequently, the project advanced to the assembly of the prototype, marking the initial strides toward its realization.

During this project, the first stage in the creation of a device has been carried out: the prototype.

A functional prototype has been built and developed for a specific environment, using near infrared spectroscopy technology, which allows the detection of water volume variations thanks to the incorporation of Machine

Learning in the prototype's software. During the project, different tests have been carried out to improve and verify that the prototype's functions were being as expected. Although *in vivo* results showed limitations, the device's potential for improvement can be enhanced with additional calibrations and adjustments.

In addition, a mobile application has been designed so that patients can easily monitor the evolution of these data. This application will also send an alert message to the patient to notify them when to empty their bladder.

Acknowledgements

I would like to express my sincere thanks to all the people who have contributed directly or indirectly to the completion of this Master's thesis. First of all, I am deeply grateful to my tutor, Dulce Oliveira, for her invaluable help and understanding of my situation at all times.

I also thank my parents, for their unconditional love and unwavering support. Their confidence in me has given me the strength to overcome challenges and achieve my goals.

To my boyfriend, Daniel, thank you so much for your patience, support and encouragement in the most difficult moments. Your understanding and love have been a constant source of motivation and strength.

I would like to extend my deepest gratitude to Diogo Sêco da Rocha, whose contributions have been pivotal to the success of this project. I am incredibly thankful for the time he has invested in helping us overcome various challenges.

I am also grateful to Dr. Rui Prado for allowing us to conduct the in vivo experiments. His collaboration has enriched our study significantly.

Finally, I would like to express my deepest thanks to the University of Porto. Despite coming from another country, they have made me feel at home. Your hospitality and support have been essential to make my experience here so enriching and unforgettable.

To all of you, my sincere thanks.

Contents

Abstract	iii
Acknowledgements	v
List of Figures	ix
List of Tables	ix
List of Acronyms and Symbols	xvi
1 Introduction	1
1.1 Problem Statement	1
1.2 Objectives	2
2 State of Art	3
2.1 Neurogenic Bladder	3
2.1.1 Causes and Symptoms	4
2.1.2 Treatments	5
2.2 Bladder anatomy	6
2.3 Bladder physiology	7
2.4 Bladder filling monitorization techniques	8
2.4.1 Architecture of wearable bladder monitoring systems	9
2.4.2 Bladder monitoring products available on the market	10
2.4.3 Bladder monitoring prototype devices	13
2.5 Near-Infrared Spectroscopy (NIRS)	18
2.5.1 Operation	20
2.5.2 Modes of Operation	21

3	SafeBladder	23
3.1	Concept	23
3.2	Previous work	23
3.2.1	Technical description	24
3.3	Testing the functioning of the prototype	24
3.4	Actual Prototype	29
3.4.1	System Architecture	29
3.4.2	Hardware	29
3.4.3	Calibration	31
3.4.4	Moving Average Filter	35
3.5	Improved prototype accuracy	37
3.5.1	Support structure	38
3.6	Algorithm	40
3.6.1	Steps for the creation of the algorithm	40
3.6.2	Machine Learning Code	43
3.6.3	Algorithm code	47
3.6.4	Results of the algorithm	49
3.7	Experiments	51
3.7.1	<i>In vitro</i> experiment	52
3.7.2	<i>In vivo</i> Experiment	54
4	Mobile Application	59
4.1	Objectives of the Mobile Application	59
4.2	Preliminary Design	61
4.3	Integrated Development Environments	63
4.4	Framework	63
4.5	Emulator	64
4.6	Algorithm Modification for Mobile Application Integration	64
4.7	Connection	66
4.7.1	Connection Selection	66
4.7.2	Cable Connection Page	66
4.8	Results	67
5	Conclusions	71
5.1	Conclusions	71
5.2	Future Work	72

List of Figures

2.1	Bladder nervous system [6].	4
2.2	Catheterization treatment for urine drainage [10].	5
2.3	Bladder anatomy [12].	6
2.4	SENS-U monitoring device [18].	10
2.5	DFree monitoring device [20].	12
2.6	BladerScan BVI 9400 monitoring device [21].	13
2.7	Prototype of a wearable ultrasound device for bladder monitoring [23].	14
2.8	Prototype of a bioimpedance device for bladder monitoring [24].	15
2.9	Conceptual diagram of (a) the proposed sensor based on a resistor ladder network and (b) the discretized changes in the overall resistance of the sensor [25].	15
2.10	Pressure Sensor System CAD Design. Component 1 is the MS5534C miniature pressure sensor. Component 2 is the ATtiny43u microcontroller. Component 3 is the ZL70101 transmitter. Component 4 is a standard oscillator to be used for the miniature pressure sensor [26].	16
2.11	The wireless pressure sensor circuit diagram, (a) the size compared to a coin (b) [27].	17
2.12	Implantable wireless and real-time bladder strain-monitoring system [28].	17
2.13	NIRS sensor componentes [29].	18
2.14	Different paths that a given photon can travel. After several reflections by the bio-logical tissue, photon 1 is effectively detected at detector, photon 2 ends up getting captivated by tissue, while photon 3 escapes through the surface [33]. . .	20

3.1	Initial prototype [34].	25
3.2	Prototype of the first test.	26
3.3	Prototype of the third test.	27
3.4	Results of the third test with the prototype.	28
3.5	System architecture diagram.	29
3.6	Assembly of the prototype.	31
3.7	Plotted results of the photodiode reading at a distance (A) of 3 cm and at a distance (B) of 2 cm.	32
3.8	Graphical results of the photodiode reading using a resistance of (A) 1 M Ω and (B) 2 M Ω	34
3.9	Graphical results of the photodiode reading using a resistance 2 M Ω (A) without optical filter and (B) with optical filter.	35
3.10	Graphical results of the photodiode reading (A) without the moving average filter and (B) with the moving average filter.	36
3.11	Graphical results of the photodiode reading (A) without the moving average filter and (B) with the moving average filter as the tank fills with water.	37
3.12	Exemplification of the resulting readings with a prototype with one sensory module (left) and with three sensory modules (right).	38
3.13	SolidWorks design of the supporting structure.	39
3.14	Prototype of 3 modules together with a social structure.	40
3.15	Extract of the csv file with the data collection of the prototype to be used with Machine Learning.	41
3.16	Reading of the csv file with the stored readings of the prototype.	44
3.17	<i>train_test_split</i> function fragment code.	44
3.18	Fragment of the code with the definition of the models.	44
3.19	Hyperparameters used for Random Forest and SVR models.	45
3.20	Fragment of the Machine Learning code where the search for the best model is performed.	45
3.21	<i>VotingRegressor</i> model code.	46
3.22	Evaluation of the MSE of each model.	46
3.23	Code for the construction of the <i>linear_model_pipeline</i> model.	46
3.24	Load code of the trained models.	47
3.25	<i>make_predictions</i> function code.	47

3.26	<i>read_arduino_data</i> function code.	48
3.27	<i>get_input_features</i> function code.	48
3.28	Looping code for continuous water volume prediction. . .	49
3.29	Graphical result of water volume prediction using 1 and 3 sensory modules.	51
3.30	Setup of the <i>in vitro</i> experiment.	52
3.31	<i>In vitro</i> experiment with a 1.5 cm thick silicone.	54
3.32	<i>In vitro</i> experiment with a 3 cm thick silicone.	55
3.33	BladerScan device.	56
3.34	Subject using the SafeBladder prototype for testing.	57
4.1	Process cycle of the SafeBladder device.	60
4.2	Diagram of the connections between devices.	60
4.3	App main page.	61
4.4	App sections pages.	62
4.5	Structures of the elements of the Back-end.	64
4.6	Algorithm modification for Mobile APplication Integration. .	65
4.7	Scrolling of the main menu screen.	68
4.8	Percentage and chromophores content page.	68
4.9	Record content page.	69
4.10	Connection type configuration page.	69

List of Tables

2.1	Classification of bladder filling monitorization techniques.	8
3.1	Difference of the components incorporated in the pre-prototype and prototype.	31
3.2	Maximum and minimum values of the plotted results for a SD distance of 3 cm and 2 cm.	33
3.3	Responsivity and Bandwidth [36].	33
3.4	Performance of the Machine Learning Models for 1 module	50
3.5	Performance of the Machine Learning Models for 3 modules.	50
3.6	Prototype testing results for Setup 1 (Plastic Container) and Setup 2 (Silicone + Plastic Container)	54

List of Acronyms and Symbols

NB	Neurogenic Bladder
NIRS	Near-Infrared Spectroscopy
UTI	Urinary Tract Infections
DAT	Digital Assistance Technology
IMU	Inertial Measurement Unit
BIA	Bioelectrical Impedance Analysis
ASIC	Application-Specific Integrated Circuit
MEMS	Micro-electromechanical
RF	Radio Frequency
LED	Light-Emitting Diode
O2Hb	Oxygenated Hemoglobin
HHb	Deoxygenated Hemoglobin
SD	Separation Distance
AFE	Analog Front-End
ADC	Analog Digital Converter
MSE	Mean Square Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
SVR	Support Vector Regression

Chapter 1

Introduction

Neurogenic bladder (NB) is a pathological condition where nerve communication with the bladder does not occur properly due to lesions at any level of the nervous system, including the cerebral cortex, spinal cord or peripheral nervous system [1].

Around 60% of patients suffering from these lesions use the catheterization technique to urinate, which is usually accompanied by a monitoring of the urine filling level to avoid urological complications [2].

This Master's Thesis was developed in the Faculty of Engineering of the University of Porto and INEGI. The aim of this project is to design a device to monitor the bladder filling level in an ambulatory, ergonomic and simple way, improving the quality of life of patients suffering from neurogenic bladder.

1.1 Problem Statement

The number of people suffering from the neurogenic bladder is rising every year. Neurogenic bladder will affect 80% of the patients with new bladder lesions, 70% of patients with multiple sclerosis, 40% of patients with Parkinson's disease and 15% of patients who suffer a stroke [2].

The main treatment for patients with neurogenic bladder is catheterization. This treatment is an invasive technique, where the catheter is inserted through the urethra until it reaches the bladder to allow it to drain. Therefore, it is in the patient's best interest to perform this technique as few times as possible to avoid discomfort and pain, and to prevent infection and damage

[3].

Thus, it is appropriate to use a monitoring system to know the level of bladder filling, so that when the predetermined threshold is reached, bladder drainage is performed.

One of the drawbacks is that most monitoring devices on the market are not ambulatory or require a professional to understand the data received by the device, which greatly limits the quality of life of patients [4].

1.2 Objectives

The SafeBladder project was previously started at INEGI, where the initial design and construction of the pre-prototype device was carried out.

Thus, the first objective was to understand the Near Infrared Spectroscopy (NIRS) technology, which components had been used for the pre-prototype and their function, and how they were assembled in the circuit. The code used for the development of the device was also analysed to understand the functions it performs and parameters it uses.

The next objective was to perform proofs of concept and experiments with the pre-prototype created to see if it works correctly with the requirements set.

Then, since the prototype created so far is unfinished and performs basic functions, some improvements and changes must be made both in the assembly and in the code to obtain the final results of the volume, and with this, experimental tests will be carried out for validation.

Finally, a mobile application was designed to allow the patient to easily visualise the evolution of this volume in real time and to alert the patient when to empty their bladder.

Chapter 2

State of Art

2.1 Neurogenic Bladder

The process of urination is mainly coordinated by the activity of the detrusor muscle of the bladder and the urethral sphincter, controlled by the central nervous system, which in turn controls the activation of the sympathetic or parasympathetic nervous system to achieve normal micturition [3]. Figure 2.1 shows these nerve connections to the various anatomical components of the bladder.

The storage phase of urine is controlled by the sympathetic system, where the detrusor is relaxed while the internal and external sphincters contract to prevent the expulsion of urine. In the emptying process, controlled by the parasympathetic system, the opposite occurs, the detrusor contracts exerting pressure on the bladder and the urethral sphincters relax [5].

Patients suffering from this condition have no sensitivity to perceive how full their bladder is. Therefore, patients must empty their bladder by carrying out time protocols or by drinking manually, which will allow them to guess in an orientative way how full their bladder is. However, the bladder does not always fill at the same rate, as it depends on other physiological factors such as breathing or sweating, and other parameters such as time of day or physical condition. Therefore, this estimation of bladder emptying is very random [7].

It is therefore necessary for patients to empty the bladder at the correct times, as an overly full bladder can lead to involuntary urine leakage, as well as backflow of urine, resulting in kidney damage [7].

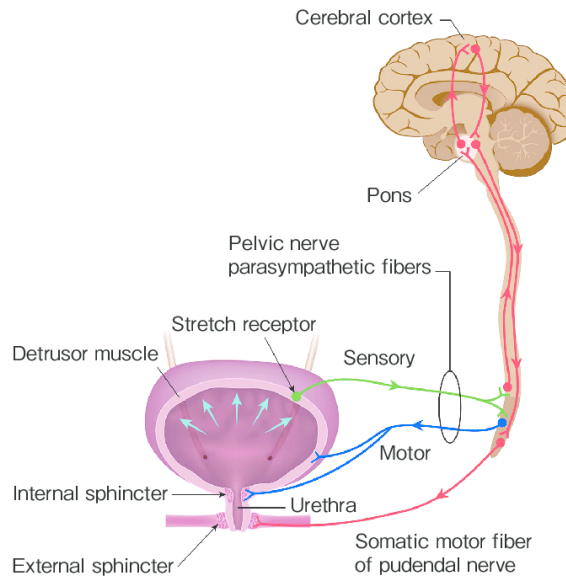


Figure 2.1: Bladder nervous system [6].

2.1.1 Causes and Symptoms

Physiological and mechanical limitations in the area of the urinary system can result in urination problems, leading to the sphincter failing to exert adequate pressure. Injury or disease that disrupts the central, peripheral or autonomic nervous systems can lead to neurologically based bladder dysfunction.

The main conditions that lead to the occurrence of NB are Parkinson's Disease and Multiple System Atrophy, Dementia Incontinence, Cerebrovascular Disease, Multiple Sclerosis, Spinal Cord Injury and Spina Bifida [8].

Patients with neurological bladder dysfunction face significant concerns, such as recurrent urinary tract infections (UTI) that may progress to sepsis, urinary retention that may lead to renal failure due to high bladder pressures. Both of these problems can increase the risk of mortality. Furthermore, these patients often experience additional complications, such as bladder wall changes and urinary incontinence, which are difficult to manage because of their limited mobility due to the underlying neurological condition [9].

The stress and discomfort associated with urinary incontinence can have a negative impact on the patient's self-esteem as well as on their social, sexual, work and educational life, further complicating rehabilitation from

their primary neurological condition [9].

2.1.2 Treatments

Treatment of neurogenic bladder has several goals, such as achieving or maintaining continence to avoid both social and physical repercussions associated with incontinence; to prevent the development of elevated detrusor muscle pressure; to minimise the risk of symptomatic urinary tract infections; and finally, to avoid excessive bladder distention [3].

There are two types of treatment: pharmacological treatments, which do not require invasive procedures. Among the most common are tricyclic antidepressants, anticholinergic drugs (known as antimuscarinics), cholinergic agonists, botulinum toxin injection and opioids [3].

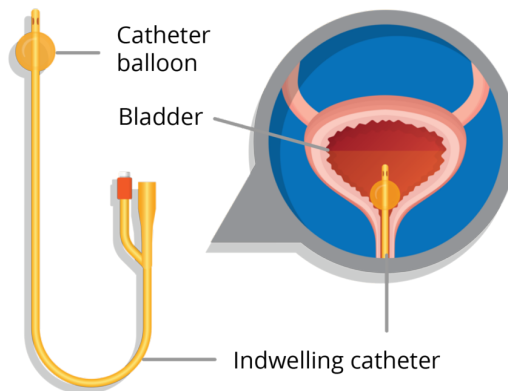


Figure 2.2: Catheterization treatment for urine drainage [10].

Then there are non-pharmacological treatments, where clean intermittent catheterization is the preferred method (Figure 2.2). Self-catheterization in cases of neurogenic bladders is a process in which a person inserts a sterile catheter through the urethra in order to empty the bladder when it cannot empty naturally due to neurological dysfunction [11].

With this method, a regular emptying of the bladder is achieved, reducing the intravesical pressure of the bladder, improving the blood circulation in the bladder wall, and thus avoiding the risk of damage to the upper urinary tract [3], [11].

When drug treatments and non-pharmacological interventions cannot manage neurogenic detrusor overactivity, surgical alternatives are considered appropriate, including neuromodulation as a treatment option [3].

2.2 Bladder anatomy

The bladder, whose anatomy is shown in Figure 2.3, is a subperitoneal hollow muscular organ whose function is to store urine, which is excreted by the kidneys, and passes through the ureters and the intravesical junction to reach the bladder.

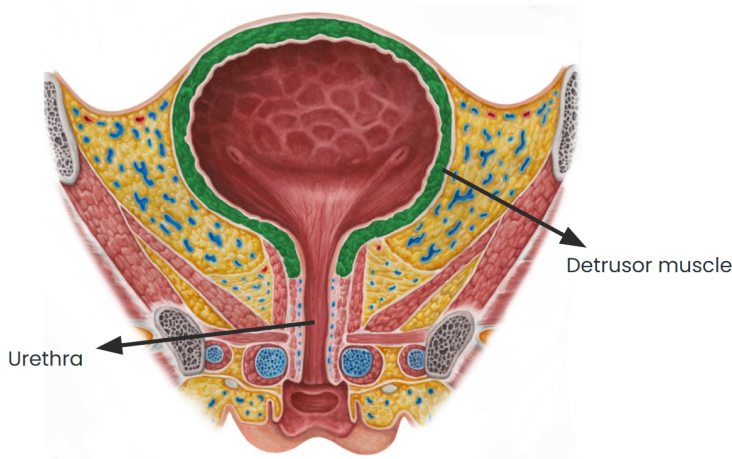


Figure 2.3: Bladder anatomy [12].

The position of the bladder varies depending on its state: when it has no urine in it, it is located in the lower pelvis, while when it is full, it extends into the abdominal cavity.

The lower urinary tract is composed of different muscles. The main muscle of the bladder is the detrusor muscle, which is a smooth muscle composed of transitional epithelium or urothelium. The cells that make up this type of musculature change depending on how full the bladder is: when the bladder is empty, these cells are round and large, while when it is full, these cells flatten out in order to increase the size of the bladder.

In addition, there is another muscle in the lower urinary tract: the urethral sphincter. The inner part of the urethral sphincter is composed of

smooth muscle and contains alpha-adrenergic receptors for sympathetic regulation. Activation of these receptors allows contraction of the sphincter to contract and de-inhibition allows relaxation of the sphincter to empty the bladder. The outer part of this sphincter is composed of striated skeletal muscle. This muscular area is composed of nicotinic receptors, which allow the sphincter to be contracted until the right moment for micturition.

The bladder is innervated by a network of parasympathetic, sympathetic and somatic nerve fibres. Parasympathetic nerves induce contraction of the detrusor muscle and relaxation of the internal urethral sphincter, thus facilitating the process of urination. Sympathetic stimulation, on the other hand, causes relaxation or expansion of the detrusor muscle, while contracting the internal sphincter.

Mechanoreceptors located in the bladder wall detect the stretching of the muscle as a result of distension. Sensory information from the upper bladder follows the sympathetic innervation pathway, while information from the lower bladder follows the parasympathetic nerve pathway.

The pudendal nerve (S2-S4), a derivation of the sacral plexus, provides voluntary somatic control of the striated muscles of the external urethral sphincter.

2.3 Bladder physiology

As previously mentioned, the function of the bladder is the storage of urine, which is produced by the kidneys, and reaches the bladder through the bilateral ureters [13].

The physiology involved in urination is quite complex. The innervation of the urinary tract is supplied by three sets of peripheral nerves: sacral parasympathetic nerves, thoracolumbar sympathetic nerves, and sacral somatic nerves (the pudendal nerve) [14].

The autonomic nervous system is responsible for indicating whether the bladder is to remain relaxed or contracted. Parasympathetic nerves stimulate bladder contraction, i.e. urination, while sympathetic nerves help relax the bladder for urine storage [13].

Preganglionic parasympathetic nerve fibres originate mainly in the spinal cord between levels S2 and S4. Postganglionic parasympathetic neurons release acetylcholine into the smooth muscle of the bladder, mainly

using M3 muscarinic receptors to induce detrusor contraction.

Sympathetic fibres originate in the thoracolumbar spine. Detrusor relaxation is mostly achieved by activation of the beta3 receptor.

The voiding response also involves somatic innervation. These fibres originate in the anterior horn of the sacral region of the spinal cord, and they fuse to form the pudendal nerve, which uses acetylcholine to induce contraction of the external urethral sphincter [13].

2.4 Bladder filling monitorization techniques

Real-time monitoring of the bladder filling status of patients with NB is essential to ensure that they empty their bladder at the right time and avoid physiological problems. This monitoring can be performed in an invasive and non-invasive manner, as shown in Table 2.1.

Table 2.1: Classification of bladder filling monitorization techniques.

	Non-Wearable	Wearable
Invasive	Catheterization	Implantable Sensors
Non-Invasive	Stationary and Portable Devices	External Sensor

Catheterization, besides being the treatment of choice for NB as explained above, is also a method of invasive monitoring. This technique consists of inserting a catheter, with an integrated pressure sensor, through the patient's urethra into the bladder. However, although this technique is highly accurate, it has major disadvantages such as the discomfort and pain caused by the insertion of this device, and of course, there is a high chance of contracting a UTI [4], [15].

There are other invasive monitoring systems, so-called minimally invasive methods, in which a sensor is inserted for implantation, usually in the inner walls of the bladder. These sensors collect the necessary data and send it to an external processor via wireless. Such systems are still under development, so clinical trials have not yet been conducted to see what negative effects they may have on the patient [16].

Within the non-invasive techniques, a subdivision can be made into three types of techniques most commonly used.

Firstly, there is the use of ultrasound, where a transducer sends ultrasonic waves into the patient's body. These waves reverberate off internal structures and are picked up again by the device and used to measure internal characteristics of the human body.

However, to carry out this technique, specialists are needed to understand and analyse the signals obtained. In addition, despite the existence of portable equipment, the majority of these devices are large in size, making them difficult to carry around [4], [15].

Bioimpedance is another relevant technique in this category, which consists of placing electrodes at different points on the patient's skin, injecting currents into these electrodes and measuring the voltage difference between these electrodes. Bioimpedance technology can be implemented in two different ways, but both have drawbacks. The single-channel method is simple and fast, but can only read information from a very small region. The multi-channel method obtains more information, but requires many electrodes and the patient to maintain the same posture during the reading process [4].

Finally, there is the NIRS technique, where the absorption properties of living tissues and water allow measurements of different physiological properties to be obtained [17]. NIRS technology is having a great relevance in urology, as it can be used both to analyse the haemodynamics of the detrusor muscle and to obtain the volume of urine in the bladder by reading the level of water absorption in the bladder [15].

This technique has been selected to carry out the project because, in addition to being non-invasive, it can be perfectly employed on an outpatient basis due to its portability and ease of use for the patient.

2.4.1 Architecture of wearable bladder monitoring systems

The architecture of wearable bladder volume monitoring systems does not follow a single or standardised pattern, but they all share some common parts.

A study of this type of monitoring system concluded that they all consist of hardware and software.

The hardware constitutes the physical components of the system. In the case of these wearable devices for the bladder, the hardware is made up of electrical components, such as power supply, storage unit, processing unit, communication unit and sensor unit.

This last component is the most important component of the system, as it determines how the measurement of the urine level in the bladder is performed. The sensor unit reads the preset parameter to be analysed, and this data is stored and sent by wireless communication to the patient's mobile phone (via Bluetooth or Wi-Fi) [16].

The software is responsible for obtaining the volume of urine in the bladder using different algorithms depending on the sensor.

2.4.2 Bladder monitoring products available on the market

SENS-U

SENS-U (Figure 2.4) is a compact, wearable ultrasound device, measuring $95 \times 55 \times 16$ mm. It operates wirelessly with batteries and connects to the lower abdomen by double bonding, using standard ultrasound gel.



Figure 2.4: SENS-U monitoring device [18].

The device includes four ultrasonic transducers that span a peripheral viewing angle of 30 degrees and emit ultrasonic waves towards the ab-

dominal walls. Bladder height and volume are calculated by measuring the average length of the bladder in the anterior-posterior direction using linear regression analysis.

The purpose of this device is to support children aged 6-12 years who are experiencing involuntary urine loss and are in the process of receiving medical treatment.

The only limitation of this system is that much of the accuracy of bladder filling level prediction is lost when used with obese children.

DFree

DFree is a dual-channel ultrasonic sensor comprising a controller and an ultrasonic probe employing digital assistance technology (DAT) to predict the need for voiding. This sensor is designed for patients suffering from bladder dysfunction of all ages [19].

The mode of operation of this sensor is to measure the degree of bladder dilatation as the bladder fills with urine, using ultrasound measurements. As it can be seen in Figure 2.5, the user must have an IOS mobile device or tablet, as the sensor works together with a mobile application, where the data received from the evolution of the bladder is sent wirelessly to track its status.

In addition, the mobile application alerts the user when to go to the toilet. This alert is sent when a threshold is exceeded, which can be determined by the user.

The size of the device is about 5 cm wide, 3.6 cm deep and 1.8 cm high, and its weight is 25,5 g. It has a rechargeable lithium-ion battery, which has an autonomy of 26 hours.

The sensor is placed above the pubic bone for more accurate detection. In addition, a gel and adhesives must be used to attach the sensor to the body.

However, this device has limitations in the accuracy of bladder status detection with patients with excess fat or sagging skin in the abdominal area, patients with scars, patients prone to skin rashes from the use of medical tape and ultrasound gel, pregnant women and minors, as their bladders are often too small for this device.



Figure 2.5: DFree monitoring device [20].

BladderScan BVI 9400

BladderScan BVI 9400 is a portable ultrasound device that allows a measurement of the urine volume in the bladder to assess urinary retention and post-void residual bladder volume.

This device has a transducer probe that can be seen in Figure 2.6, whose function is to send high-frequency waves into the bladder area and receive the waves reflected by the tissues in this part of the body. The ultrasound machine then receives this information and constructs a 2D image of the bladder.

In the UK, there are three BladderScan models available (BVI 9400, BVI 6100, and BVI 3000). These models claim to provide more precise measurements compared to traditional 2D ultrasound methods. They have evolved from earlier versions of the technology. The BVI 3000 is a portable console with an attached probe, while the BVI 6100 is a handheld probe with a built-in display. Both models are designed exclusively for use in adults. Utilizing VMODE software, they calculate bladder volume by transforming 2D slice images into a 3D image, which is not only shown on the screen but can also be printed. The BVI 9400 employs advanced patented software known as NeuralHarmonics for bladder volume measurements. This innovative technology transforms 2D slice images into a 3D representation and utilizes neural network algorithms to enhance the accuracy of segmentation for the targeted structures or organs [22].



Figure 2.6: BladerScan BVI 9400 monitoring device [21].

BladderScan BVI 9400 consists of a hand-held probe that sends and receives ultrasound waves, a colour LCD console, a lithium-ion battery, which can provide about 30 examinations in a 24-hour period.

2.4.3 Bladder monitoring prototype devices

Ultrasound Bladder monitoring devices

In 2021, Jo et al. [23] designed a prototype of a wearable device that uses ultrasound technology to obtain a real-time measurement of bladder fullness.

This system uses low-profile ultrasound transducers located in the lower abdominal zone and employing the forward-looking pulse-echo technique as sensing system.

The mode of operation is based on the reception of reflected waves from the anterior and posterior walls of the bladder. In this way, the depth of the bladder can be calculated as the difference between the amplitudes of the signals received from these walls.

Volume estimation is performed by interpolating the shape of the bladder based on the measured bladder wall locations.

The device is placed 2 cm above the pubic bone and attached to the abdominal area using straps as can be observed in Figure 2.7.

The trial showcased the viability of utilizing a 2-D ultrasound transducer array system for bladder monitoring when compared to commercially

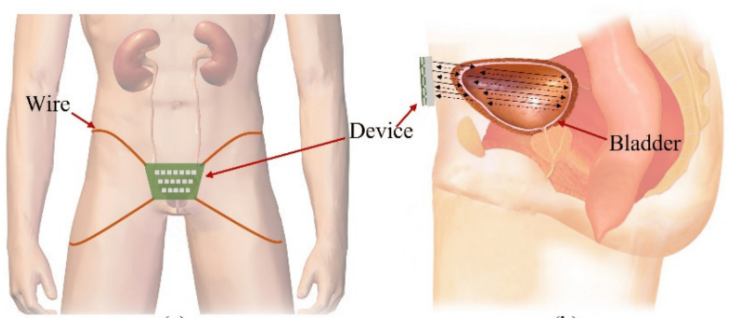


Figure 2.7: Prototype of a wearable ultrasound device for bladder monitoring [23].

available equipment. However, a limitation arises from their scanning configuration, leading to cost inefficiency.

Bioimpedance sensors

Reichmuth et al. [24] presents the design of wearable system for bioimpedance monitoring. This system uses electrodes attached to the patient's abdominal area as sensors to measure the electrical impedance of the bladder as it fills (Figure 2.8).

The system, in addition to the electrodes, consists of a Bluetooth module, which allows data to be sent wirelessly, a power management and other sensors such as a temperature sensor, an inertial measurement unit (IMU) and a bioelectrical impedance analysis (BIA) unit. All this components are powered with a Li-ion battery with a voltage of 3.3 V.

Another prototype of a bioimpedance-based bladder monitoring device is the one from Kim et al. [25]. The sensor comprises a biocompatible polypyrrole/agarose hydrogel composite organized in a multi-level resistor ladder. This design ensures the sensor output remains resistant to long-term drift, hysteresis, and degradation of the sensor material. The ladder legs have dimensions of $40 \times 5 \times 2$ mm (width $\times$ length $\times$ height).

The system is externally attached to the bladder, and as the volume changes, the number of arms in contact with the opposite rail varies, causing a discrete alteration in the total resistance of the sensor. This mode of operation can be seen in the Figure 2.9, where as the volume of the bladder increases the number of connected arms decreases.

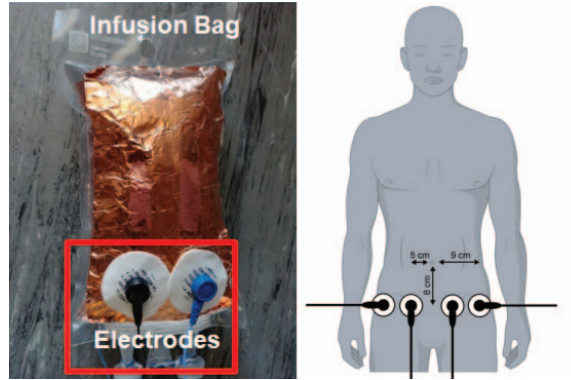


Figure 2.8: Prototype of a bioimpedance device for bladder monitoring [24].

The overall impedance is calculated based on the number of arms attached to both legs. In *ex vivo* experiments conducted on extracted pig bladders, the sensor successfully demonstrated changes in resistance corresponding to an increase in bladder volume. However, the measurement results indicate irregular fluctuations in resistance attributed to insecure attachment to the bladder.

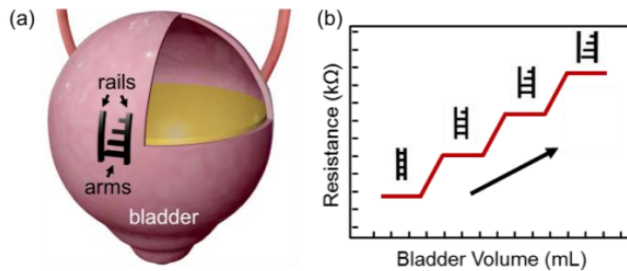


Figure 2.9: Conceptual diagram of (a) the proposed sensor based on a resistor ladder network and (b) the discretized changes in the overall resistance of the sensor [25].

Pressure sensors

Weaver et al. [26] features the design of a pressure sensor, which is attached to the inner wall of the bladder. To validate the design, a model of a bladder was created that emulates the different filling phases of the bladder. Through this simulation, a successful response of the pressure sensor was obtained.

This sensor has dimensions of $11.5 \times 17 \times 8$ mm (length $\times$ width $\times$ thickness), and is composed of different low current and low power components, such as a piezoresistive pressure sensor, a microcontroller, a wireless transmitter, and a battery, as shown in Figure 2.10.

The purpose of this device is for implantation via a cystoscope, with data transmission directed to a compact receiver affixed to a belt worn externally on the body. During a period of 12–24 hours, the user engages in their regular daily activities, allowing the accumulation of data. Subsequently, the patient is required to revisit the doctor's office for the downloading and analysis of the collected data.

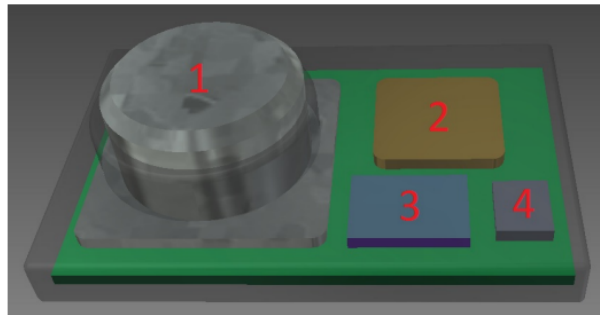


Figure 2.10: Pressure Sensor System CAD Design. Component 1 is the MS5534C miniature pressure sensor. Component 2 is the ATtiny43u microcontroller. Component 3 is the ZL70101 transmitter. Component 4 is a standard oscillator to be used for the miniature pressure sensor [26].

Majerus et al. [27] designed an implantable wireless and real-time bladder pressure-monitoring system. The complete implantable microsystem comprises an ultra-low-power application-specific integrated circuit (ASIC), a micro-electromechanical (MEMS) pressure sensor, radio frequency (RF) antennas, and a rechargeable battery within the device (Figure 2.11). To power the system, a wireless power transmitter is utilized, while the user rests on a cushion containing an embedded RF induction coil during sleep. Wireless real-time measurement of bladder pressure is achieved by the pressure sensor situated within the bladder wall. Subsequently, the pressure data is transmitted to an external receiver via a telemetry antenna.

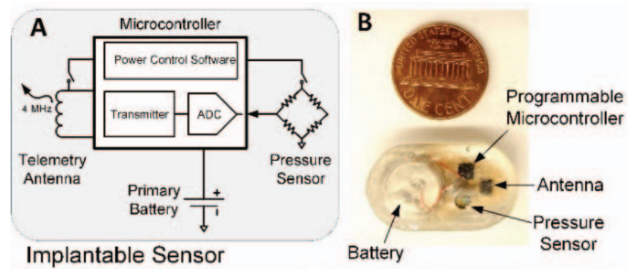


Figure 2.11: The wireless pressure sensor circuit diagram, (a) the size compared to a coin (b) [27].

Strain Sensors

Hannah et al. [28] has produced a soft, biocompatible and adaptable sensor that uses elastic electronics to monitor bladder stretch non-invasively. The device consists of layers of Cr/Au moulded polyurethane stretchable polyurethane, resulting in resistive strain sensors.

These sensors are attached to the bladder, and as the bladder fills, its walls stretch to increase its volume. The test performed with this sensor is shown in Figure 2.12, where as the size of the bladder changes, the resistance of the sensors will also change, allowing the bladder volume to be calculated.

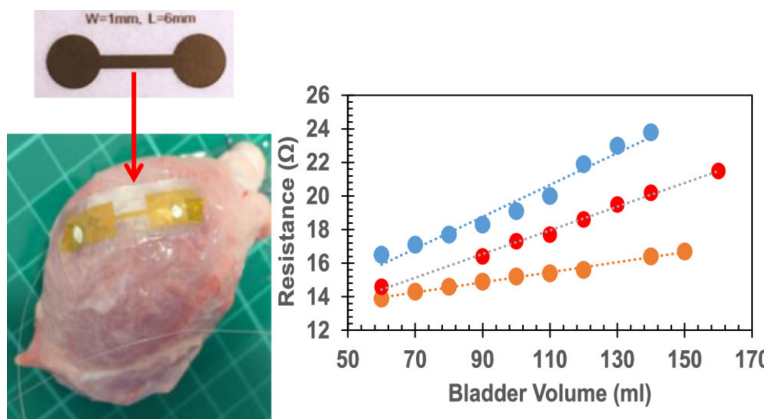


Figure 2.12: Implantable wireless and real-time bladder strain-monitoring system [28].

Near-Infrared Spectroscopy Systems

Molavi et al. [29] designed a wireless device using NIRS technology, which uses a 960 nm light-emitting diode (LED) as the emitter and a photodiode as the detector, with a separation of approximately 3 cm. The rest of the components that make up this device are shown on Figure 2.13

During use, the source LED emits light, and the detector captures the reflected light. The chromophores in the tissue absorb the emitted light, causing a reduction in light intensity. Through the application of machine learning algorithms, the degree of attenuation is analyzed to ascertain the bladder's fullness. Moreover, NIRS can be employed for various urological conditions beyond bladder monitoring. This includes but is not limited to conditions such as cryptorchidism, testicular torsion, erectile dysfunction, renal oxygenation, and muscle metabolism in renal dysfunction.

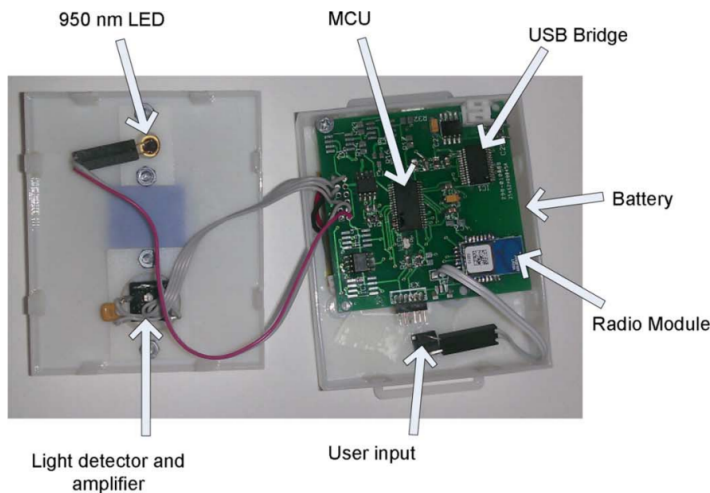


Figure 2.13: NIRS sensor componentes [29].

2.5 Near-Infrared Spectroscopy (NIRS)

Spectroscopy is the analysis of the interaction between matter and electromagnetic radiation. In other words, spectroscopy studies the amount of light absorbed or scattered by an object or matter.

From a technical perspective, NIRS is a non-invasive method that uses light in the near-infrared range (with wavelengths ranging from 650 to 1350 nm) to interact with biological material. Although this light is invisible to the human eye, it has the ability to penetrate deeper into tissue with less loss of intensity than visible light. The reflected signal is subjected to analysis and can provide information on sample characteristics such as shape, size, distribution and composition. This, in turn, allows real-time data to be obtained on physiological variables in any tissue [30], [31].

The principle of operation and analysis of NIRS is based on the absorption and reflection of near-infrared radiation. In this process, the absorption of NIRS rays of a specific wavelength varies according to the type of tissue, while all other wavelengths are reflected. The number of reflections at various wavelengths is quantified and then a microprocessor converts these measurements into analytical results [30].

This type of electromagnetic radiation is absorbed by specific compounds referred to as chromophores. Therefore, in urology, this technique is used to measure the concentration of oxygenated haemoglobin chromophores (O2Hb) and deoxygenated haemoglobin chromophores (HHb), which are related to the behaviour of the detrusor, and water, as it is the main component of urine [31].

Each chromophore possesses an absorption spectrum in which the specific extinction coefficient is represented as a function of wavelength [32].

Light attenuation can be quantified through the Beer-Lambert law and the associated equation.

$$A(\lambda) = \log \left(\frac{I_0}{I} \right) = \sum \epsilon_i(\lambda) c_i DL + G \quad (2.1)$$

Where $A(\lambda)$ is the light attenuation at wavelength λ , I_0 is the source intensity, I is detected light intensity, ϵ_i is the extinction coefficient chromophore i at wavelength λ , c_i is the chromophore concentration, B is the separation between the source and detector, L is the differential pathlength factor and G is an improver term to take into consideration fixed scattering losses [29].

2.5.1 Operation

The NIRS optode consists of two components. On the one hand we have the emitter, which is an infrared LED that will emit this electromagnetic radiation, and on the other hand we have the detector, which is a photodiode sensitive to this type of radiation.

Both the emitter and the detector are placed on top of the patient's skin, in the lower abdomen. In this way, the LED emits this infrared light, passing through the biological tissue, and depending on the tissue or components passed through, the amount of light received by the detector will vary.

The way light propagates through biological tissues is influenced by three fundamental phenomena: reflection, absorption and scattering. When photons penetrate the body through a light source, they undergo multiple reflections and can have three possible outcomes, as illustrated in the Figure 2.14.

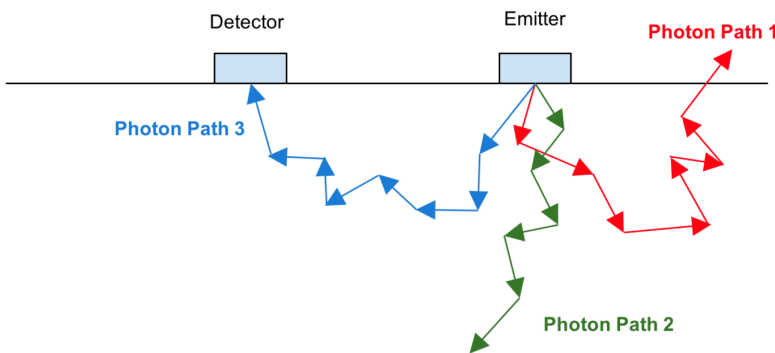


Figure 2.14: Different paths that a given photon can travel. After several reflections by the bio-logical tissue, photon 1 is effectively detected at detector, photon 2 ends up getting captivated by tissue, while photon 3 escapes through the surface [33].

The first photon, after multiple reflections, leaves the body undetected; the second photon, after multiple reflections, is absorbed by the tissue; while the third photon is finally successfully detected after its multiple interactions.

In relation to the bladder, this technique can be used to analyse changes in the volume of urine in the bladder. When the bladder is empty, i.e. there is no urine in the bladder, the attenuation is lower. However, as the bladder fills, the infrared light reaching the detector decreases, as the urine absorbs

some of this radiation.

2.5.2 Modes of Operation

NIRS operates mainly in two modes: the Transmission Mode and the Reflection Mode. In transmission mode, the emitter is placed at one end of the biological tissue, while the detector is placed at the opposite end. In this mode, the photons pass through the entire tissue, allowing comprehensive information to be obtained from the tissue environment. This transmission approach is usually most appropriate when the tissue thickness is less than 8 cm [33].

In reflection mode, both the emitter and the detector are placed on the same side of the biological tissue, which restricts the collection of information to a regional area, as the photons only travel a shallow depth. In this situation, the light will follow a curved path, and the maximum depth of that path is approximately half the distance between the source and the detector (SD).

Thus, if the separation between emitter and detector is too small, the light will not reach the bladder as its depth will be too shallow. However, if the distance is very large, there will be greater penetration of light into the bladder, but the luminous intensity that reaches the detector will decrease, since the path of the photons will be longer and a large part of them will be absorbed [32].

Chapter 3

SafeBladder

3.1 Concept

SafeBladder is a prototype wearable health device that was first developed in 2022 at the INEGI institute. This device acquires data related to the concentration of chromophores in the abdominal area and converts it into a urine volume that is displayed on a mobile application.

This device involves an optical sensor with near-infrared recognition, connected to a software system for data acquisition and analysis. It is composed by an emitter and a detector, placed next to each other, with a distance varying between 2 and 5.5 cm, and placed 2 cm above the symphysis pubis across the midline. The main function of the device is to warn the patient through a non-intrusive system. The limit value for when this notification is activated is previously defined by the physician with the aid of complementary examinations.

3.2 Previous work

The work over the past year has involved searching for commercial devices that perform this bladder monitoring function, finding and selecting the right components to use in the prototype for proper functionality, assembling the components, writing the code to make it work, and experimenting to see if the objectives have been met.

3.2.1 Technical description

Hardware

- **Source:** An infrared LED was used as the light source because of its small size, low weight and cost-effectiveness. Given that the absorption peak of water is 975 nm, the chosen LED should have a wavelength range that encompasses 975 nm. Nevertheless, commercially available LEDs with a range of 975 nm, are rare, therefore, an LED with a wavelength of 940 nm was selected for this work.

- **Detector:** A photodiode was used as a detector in previous experiments to gather the light generated by the LED. The selected photodiode must have: a large active area (to capture the largest number of photons), be sensitive to 940 nm (because the selected LEDs have a maximum wavelength of 940 nm), and a large viewing angle (to capture the largest number of photons escaping from the tissue).

- **Microcontroller:** A microcontroller is needed to manage and control the functions of the other components. The Arduino UNO was chosen as the microcontroller as it offers a very good price-quality ratio.

- **Analog-Digital Converter:** In order to allow the communication between the photodiode and the microcontroller, it's necessary a fully integrated front-end analog (AFE). The Analog-Digital Converter (ADC) chosen was the AFE4490, which filters and amplifies the signals obtained from the photodiode to obtain a more accurate signal.

- **Accelerometer:** accelerometers are sensing devices used to detect the movement of an object and to remove motion corrupting data segments. The accelerometer used for this previous prototype was the ADXL345.

3.3 Testing the functioning of the prototype

The first step was to check whether the prototype obtained (Figure 3.1) worked properly and met the objectives set.

For the first test, solid obstacles were used to see if they were detected by the infrared sensor. These objects were placed between the infrared LED and the photodiode to interrupt the infrared light reaching the photodiode. Large variations were observed when the path between these two components was free and when it was interrupted by the selected obstacle, resulting

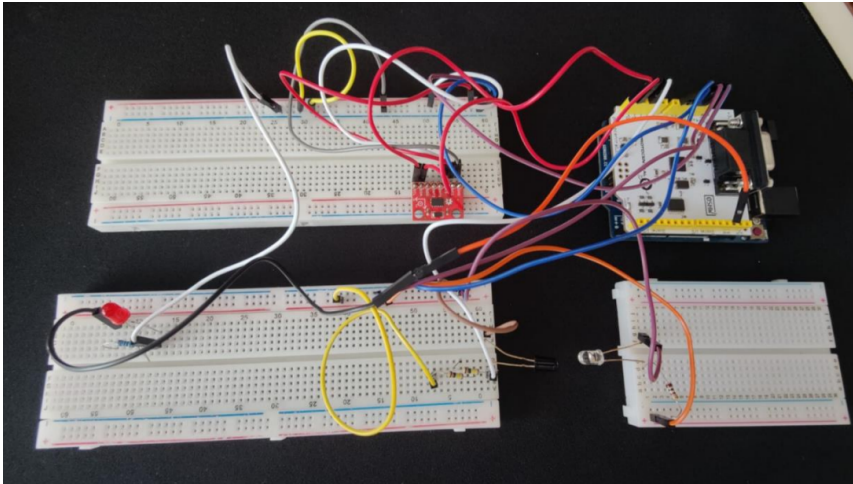


Figure 3.1: Initial prototype [34].

in a successful outcome of this first trial.

The aim of the second test was to check whether the prototype could detect the presence of water. To do this, a glass beaker was used and placed between the infrared LED and the photodiode. The sensor reading was displayed as the glass was filled with water. However, when the water exceeded the level of the sensor, there was not even a small change in the reading. Modifications to the alignment of these two components were tried without success.

So, as the problem was not entirely clear, it was decided to remove all the components of the prototype and start building it again step by step to try to solve the error.

First test

We started by using the Electro Fun IR sensor, which is a sensor that has a built-in emitter and an infrared receiver in parallel. This device was used because of its compatibility with Arduino, its availability in the INEGI laboratory and because it had been successfully tested beforehand.

This sensor is composed of an infrared LED and a photodiode in parallel, together with a potentiometer, which allows the sensitivity of the sensor to be changed. Unlike the other sensors, this component has a digital output, i.e. only a high or low signal (0 or 1) will be received.

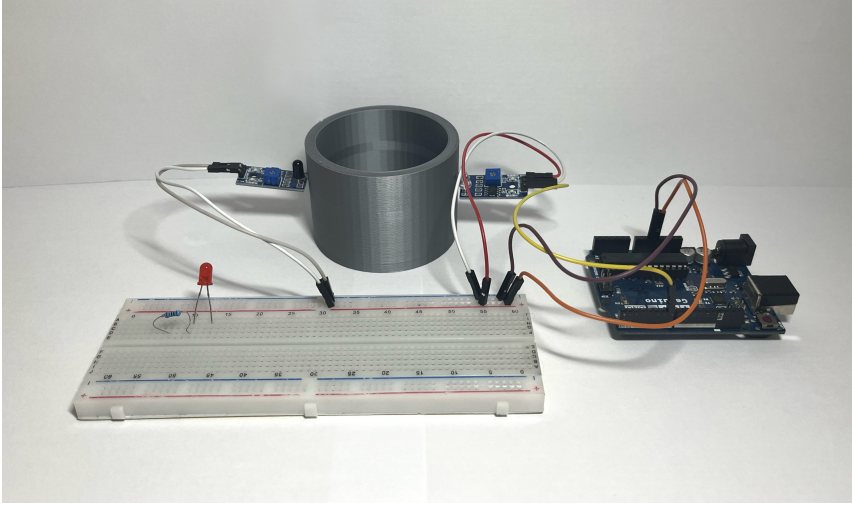


Figure 3.2: Prototype of the first test.

The assembly for this first test is shown in Figure 3.2. Two sensors powered by a 5V current were used, one placed in front of the other, and the water in the middle of the two sensors as an obstacle. To perform the test in a more comfortable way, a cylindrical support was designed and built, with two holes on the sides where the photodiode and infrared LED corresponding to each sensor are inserted so that they are fixed and aligned, and in the centre of this support the container with water will be placed.

The test procedure was to adjust the value of the potentiometer with the glass of water placed between the two sensors until the photodiode detected the presence of water. The result of this test was successful, obtaining a value equal to "0" when water was present, and equal to "1" when water was absent.

In this way, it was possible to confirm that there was a mistake with the previous prototype, as water detection was possible using this new sensor. However, with this model only digital values are obtained, and the aim of the project is to obtain an analogue value that shows the variations of the received signal.

Second test

For this second trial, it was decided to go back and use the original structure of the initial prototype, using a photodiode and individual infrared LEDs.

The same LED and photodiode were used as in the original prototype, and with the same configuration, however, the other components were removed in case they produced any interference.

The prototype was retested with these new changes without success. A check of the technical data sheet of these two components was carried out, and it was found that the value of the resistors used so far in the photodiode was too high ($2\text{ M}\Omega$). In this way, the photodiode was unable to detect the slightest variations produced by the presence of water, as the signal it received was very small.

The value of the resistor was changed to 100Ω to adjust the value of the voltage and current to be received by the photodiode. By making this change, successful data was obtained from the photodiode reading upon detection of the water presence.

Third test

For this test, it was decided to check whether the mechanism used previously would work with the arrangement with which the prototype was finally going to work, i.e. with the LED and photodiode in parallel, instead of facing each other, as both components should be on the same side of the patient's skin.

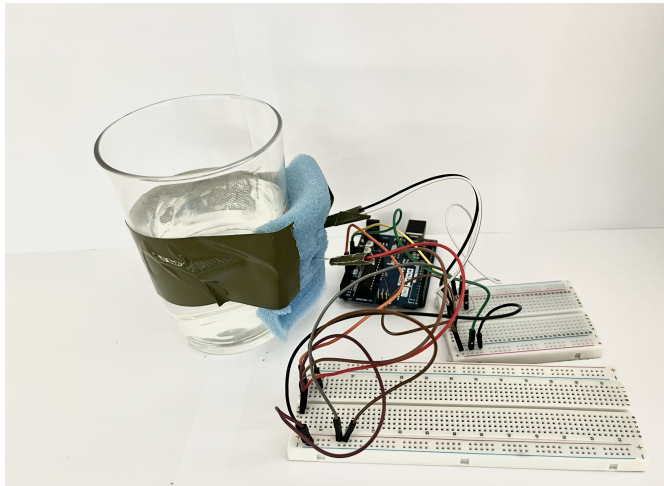


Figure 3.3: Prototype of the third test.

For this purpose, the connection of the components was the same as in the previous test, but this time 20 cm male-female wires were used in order

to be able to place these two components in the desired arrangement more freely.

In order to carry out this test, homemade elements were used due to their availability and the simplicity required to verify the correct functioning of the device. Figure 3.3 illustrates the specific layout of the components used. A sponge was chosen as a support for the photodiode and LED, as its flexible nature allowed it to adapt perfectly to the curved shape of the glass beaker. In order to keep both components in a parallel and stable position, two small holes were made in the sponge, spaced at a precise distance of 3 cm, to insert each of the elements.

Once the photodiode and LED were placed on the sponge, the sponge was firmly attached to the beaker using adhesive tape, ensuring that the components remained in place during the tests. This improvised method proved to be effective in maintaining the necessary stability while the measurements were being made.

During the test phase, the cup was repeatedly filled and emptied with water to simulate real-life conditions. As can be seen in detail in Figure 3.4, the results were highly satisfactory: clear peaks were obtained that accurately indicated the points where the vessel was full and empty. This clear and effective visualisation of the data validated the functionality of the device under test, confirming that the settings and layout of the components were suitable for their specific purpose.

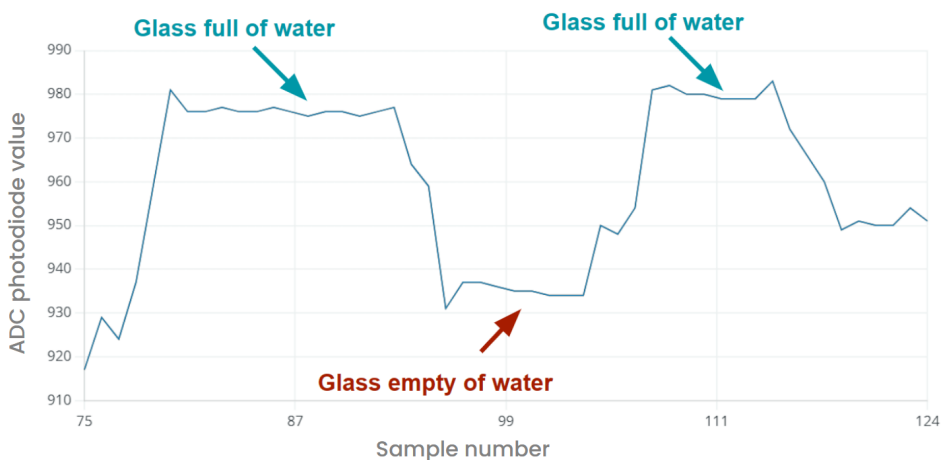


Figure 3.4: Results of the third test with the prototype.

3.4 Actual Prototype

Since the results of the experiments carried out in the previous sections had a positive result, showing that infrared light can indeed detect water, it was decided to make improvements to the prototype to make it more accurate and improve its performance.

3.4.1 System Architecture

The diagram in Figure 3.5 shows the operating architecture of the SafeBladder device, which combines hardware and software components for motion detection and volume prediction. The device uses an accelerometer (ADXL 345) to detect motion and send position data to a microcontroller (Arduino UNO). This microcontroller controls an LED to emit NIR (Near-Infrared) light and receives signals from a photodiode that detects this light. The data collected by the microcontroller is processed by machine learning algorithms, and the results are sent to a mobile application that notifies the user about the status of the predicted volume.

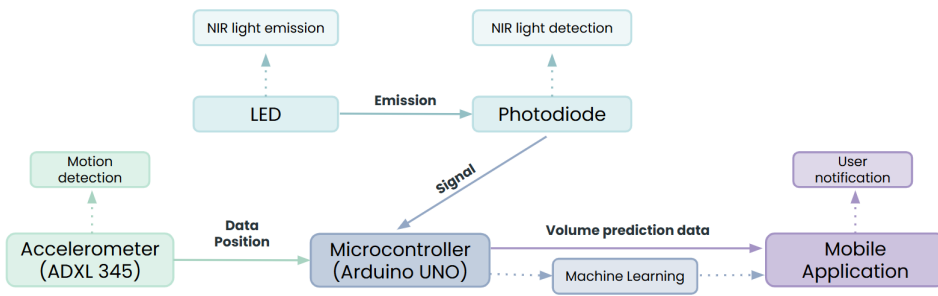


Figure 3.5: System architecture diagram.

3.4.2 Hardware

- **Source**: The same LED as in the previous prototypes was used as the light source, with a peak wavelength of 940 nm, as it is the most commercially available infrared LED and is close to the peak wavelength of water (975 nm).

- **Detector:** For this prototype, the detector was replaced by the CJMCU101, which is a sensor that has a photodiode and a transimpedance amplifier integrated on the same chip. The CJMCU101 photodiode has a size of 2.29 mm x 2.29 mm, is very small in size making it easy to handle, and has a high responsivity of 0.45 A/W. This sensor has several advantages compared to the previously used photodiode such as higher sensitivity (being able to detect infrared light better), lower noise (obtaining a cleaner signal) and easier to use, as it does not require the application of an external amplifier, making the circuit simpler.

- **Microcontroller:** Arduino UNO continued to be used as the microcontroller because of its ease of use (providing a simple programming environment), flexibility (being able to customise the prototype to fit the project's purpose) and low cost.

- **Optic Filter:** The CJMCU101 sensor, although having its peak responsivity in the infrared light range, detects more types of light (from the 300 nm to 1100 nm range), making the reading influenced by other types of light. Thus, to prevent the photodiode from detecting other wavelength ranges, it was decided to implement an optical filter, which will allow only light of the desired wavelength to be transmitted. The central wavelength for this optical filter is 940 nm $\pm$ 5nm, with an average bandwidth of 30 nm and a transmittance of $< 90\%$.

- **Accelerometer:** The detection of the patient's position is important for the correct functioning of the device, since, depending on this position, the liquid retained in the bladder will be found in different places in the bladder, varying the reading of the photodiode. The ADXL345 accelerometer was used, which has a low power consumption, a high resolution and can measure up to ± 16 g.

Table 3.1 shows the differences between the pre-prototype and the updated prototype, where it is possible to observe the improvement of components (as in the case of the photodiode) and the incorporation of new ones (as in the case of the optical filter).

Figure 3.6 shows the assembly of the new prototype, which is now using a transparent water tank that simulates the bladder. This allows a clearer and more accurate visualisation of the device's operation, making it easier to evaluate and adjust its performance.

Table 3.1: Difference of the components incorporated in the pre-prototype and prototype.

Components	Pre-Prototype	Prototype
Source	LED 940 nm	LED 940 nm
Detector	Photodiode 940 nm	CJMCU101
Microcontroller	Arduino UNO	Arduino UNO
Optic filter	-	940 nm
Accelerometer	ADXL345	ADXL345

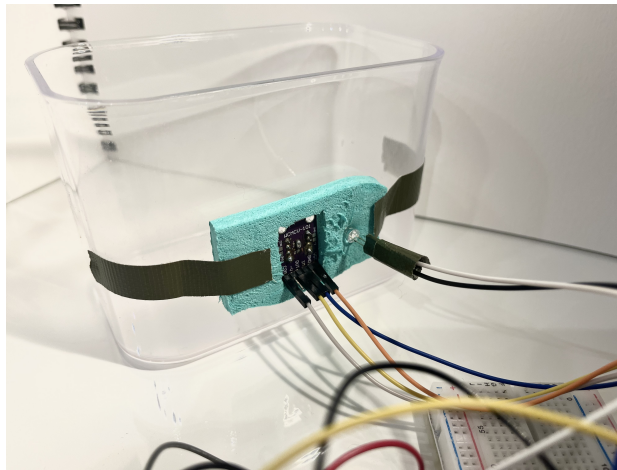


Figure 3.6: Assembly of the prototype.

3.4.3 Calibration

Calibration of the separation distance (SD)

The distance between the emitter and the detector is a critical factor affecting the optimisation of the data readout, as an incorrect distance can affect the intensity of infrared light received by the photodiode.

A literature search of the generally used distance between photodiode and LED with NIRS devices concluded that the SD varies between 2 and 3 cm [29], [33], [35].

Two tests were carried out using the prototype in the same environment

and under the same conditions, but varying the optode separation distance.

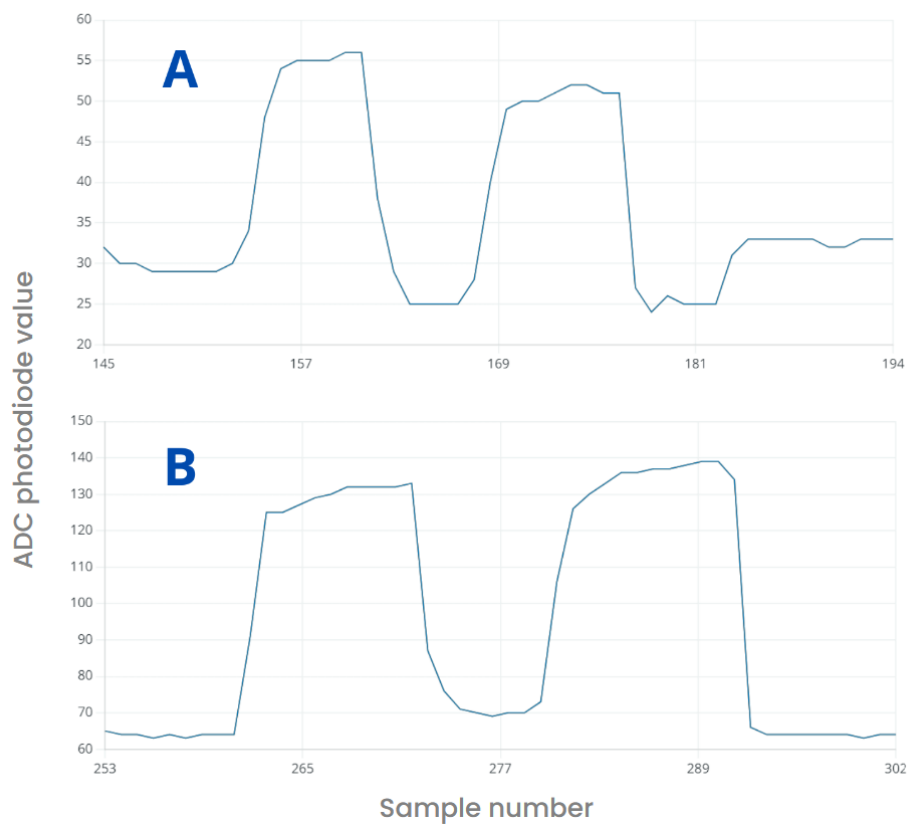


Figure 3.7: Plotted results of the photodiode reading at a distance (A) of 3 cm and at a distance (B) of 2 cm.

Figure 3.7 shows the graphs of the results at a distance of 3 cm and 2 cm respectively.

When analysing the graphs, the maximum and minimum values read by the photodiode (absence of water and presence of water respectively) in both tests were used to obtain the result of their difference, and thus, to know with which SD distance a higher sensitivity of water detection is obtained.

As shown in Table 3.2, the greatest difference between the peak values of the graphs occurs at a distance of 2 cm. Consequently, due to the larger range and amplitude between these values, more variations will be observed in the water volume readings, indicating that the prototype is more sensitive.

Table 3.2: Maximum and minimum values of the plotted results for a SD distance of 3 cm and 2 cm.

	3 cm	2 cm
Max	54	135
Min	25	63
Dif	29	72

Calibration of the CJMCU101 gain

The CJMCU101 module includes the OPT101 sensor, which is a high-speed transimpedance photodiode that converts light into an electrical signal. Its gain determines the ratio between light intensity and output voltage, a crucial factor for accurate measurements.

There are two different settings to change the voltage responsivity of the sensor. The first is to use the sensor’s internal resistance of 1 MΩ together with a modifiable external resistor in series with this internal resistance. The second configuration would be to simply replace the internal resistor with an external resistor. For this project it was decided to use the second configuration, as similar gains to the first configuration could be achieved using a circuit simpler [36].

Table 3.3 shows the variations produced by changing the value of the external resistor.

Table 3.3: Responsivity and Bandwidth [36].

R _{EXT} (MΩ)	C _{EXT} (pF)	DC Gain (× 10 ⁶ V/A)	Bandwidth (kHz)
0.05 ⁽¹⁾	56	0.05	58
0.1 ⁽¹⁾	33	0.1	44
1	—	1	23
2	—	2	9.4
5	—	5	3.6
10	—	10	1.8
50	—	50	0.34

The graphs in Figure 3.8 show the response of the CJMCU101 in the same environment by varying the external resistance from 1 MΩ to 2 MΩ. A comparative analysis shows that with the 2 MΩ resistor a higher gain is obtained due to the higher total resistance and a lower light intensity is

required to obtain the same reading. In addition, with this configuration, the response is more sensitive to light variations.

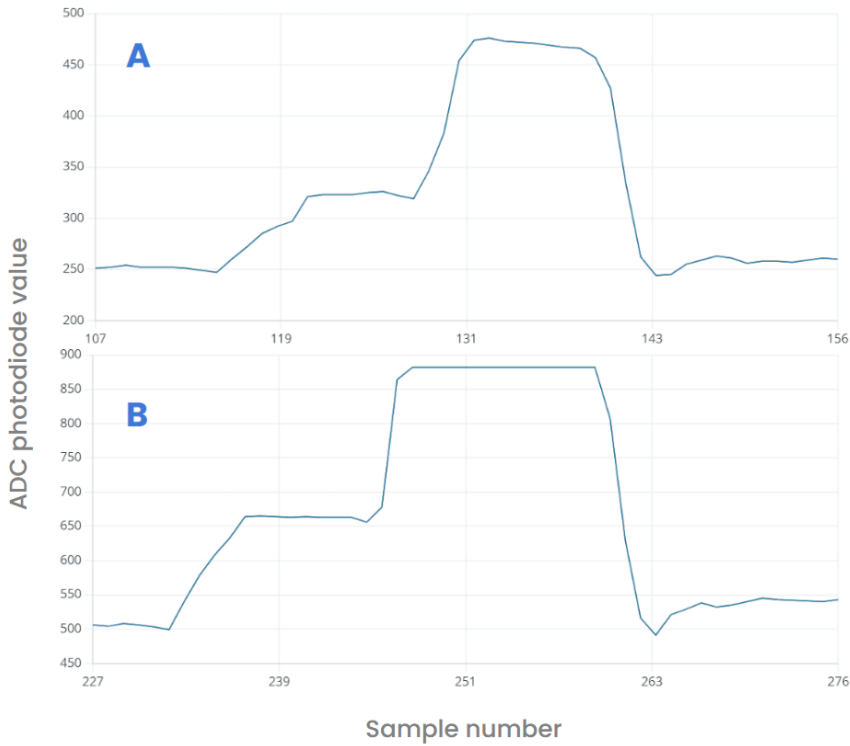


Figure 3.8: Graphical results of the photodiode reading using a resistance of (A) 1 MΩ and (B) 2 MΩ.

Optical filter application

The next and last step for the complete calibration of the prototype was to implement the optical filter to see if it works correctly.

Because this project is focused on the development of the SafeBladder prototype, and not the final device, an inexpensive bandpass filter with a central wavelength of 940 nm and a bandwidth of 10 nm was used to block the passage of light outside this range to the photodiode.

After placing this filter in front of the sensitive area of the photodiode, a graph was obtained with the results of its reading and a comparison of this reading was made in the same environment but without the filter, as shown

in Figure 3.9.

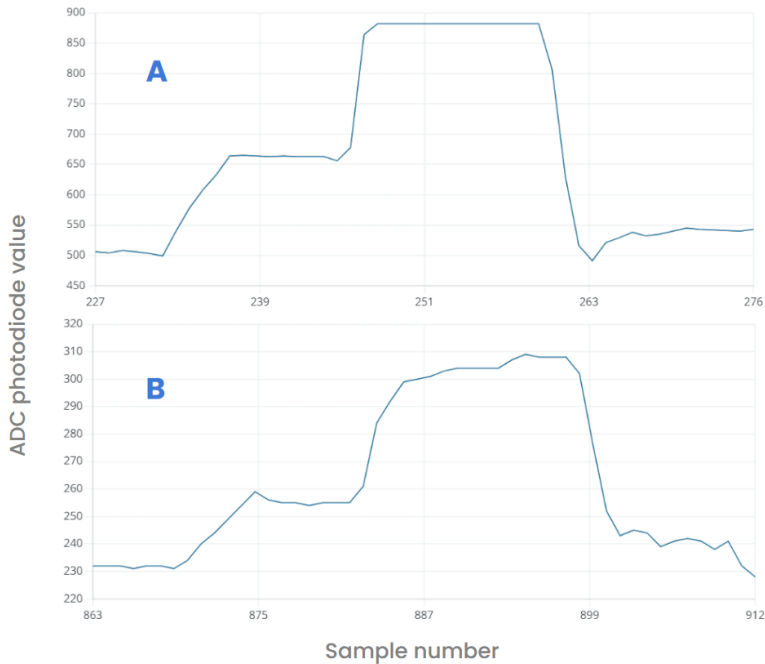


Figure 3.9: Graphical results of the photodiode reading using a resistance 2 M Ω (A) without optical filter and (B) with optical filter.

Analysing the results, it can be seen that in the lower graph, the analogue value of the photodiode is reduced on a large scale, meaning that the visible light from the environment is blocked by this filter, and also that the filter fulfils its function.

3.4.4 Moving Average Filter

Despite replacing the initial photodiode with the CJMCU101 together with the optical filter, the signal read by the sensor still generated a lot of noise. Therefore, it was decided to use the moving average filter. This filter is a signal processing technique whose function is to reduce the random noise that occurs in the sensor readings, smoothing the signal.

Its operation is based on calculating the average of a predefined number of samples and displaying this average value as the filtered value.

The number of samples taken to make this average was set to 10,

thus, in the code, a circular array is being created where the last 10 data are continuously stored in the array.

To analyse whether the reading is satisfactory, a test was first carried out with the same volume of water by obtaining the reading of the sensor with and without the moving average filter (Figure 3.10).

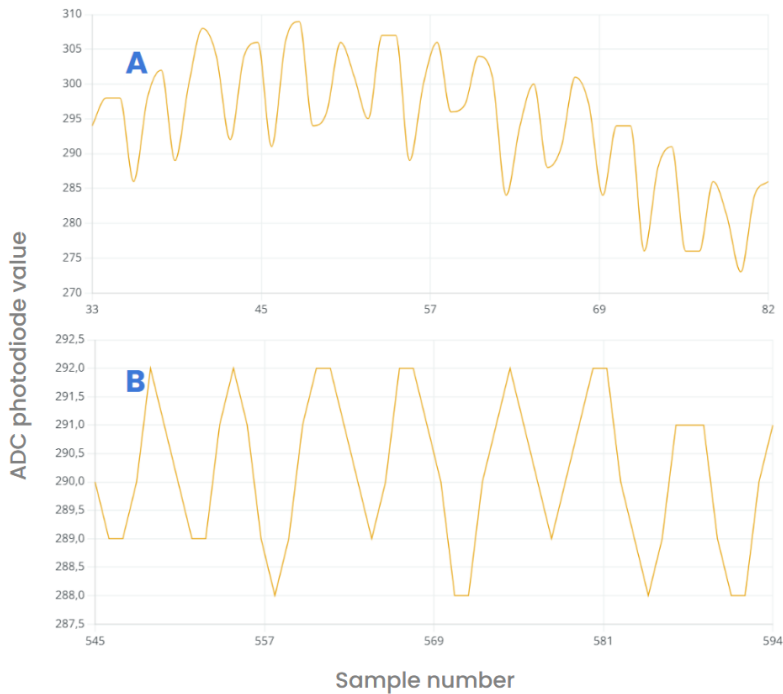


Figure 3.10: Graphical results of the photodiode reading (A) without the moving average filter and (B) with the moving average filter.

Analysing the peaks of the graphs in Figure 3.10, it can be seen that in the first graph the maximum and minimum values are 302 and 287 digital units respectively. In the second graph (using the moving average filter) it can be seen that these values are 292 and 289 digital units respectively. Thus the difference in the values for the first case is 15 and in the second case is 3, concluding that the use of the moving average filter helps to smooth the signal obtained by the photodiode.

In addition, for a fully satisfactory corfirmation, another sensor reading (Figure 3.11) was taken as the tank was filled with water, to see the differences that occurred when the moving average filter was applied.

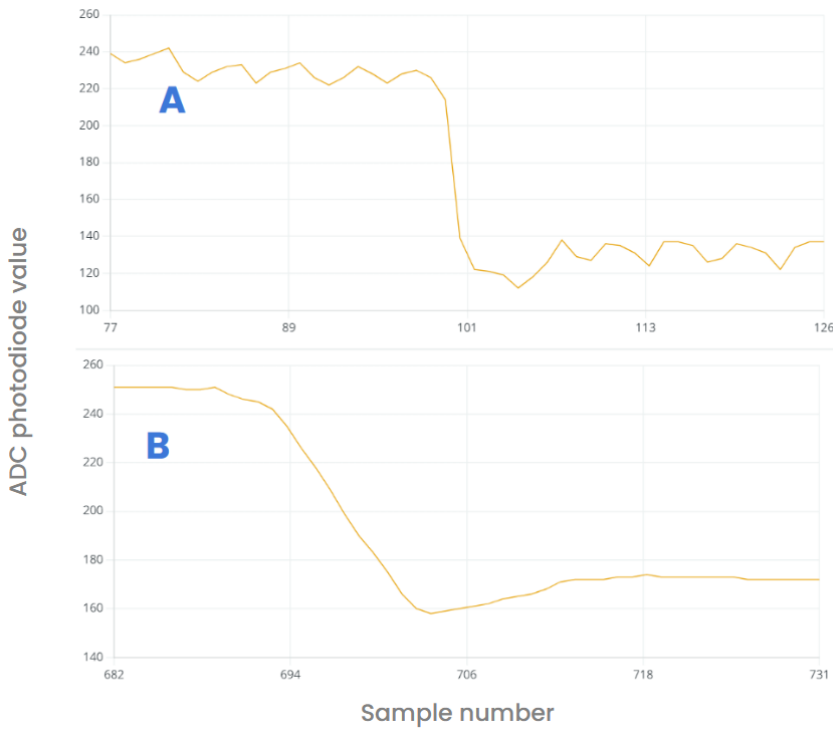


Figure 3.11: Graphical results of the photodiode reading (A) without the moving average filter and (B) with the moving average filter as the tank fills with water.

3.5 Improved prototype accuracy

Sensory modules

The initial prototype consists of an infrared LED and a photodiode (a single sensor module). With this prototype it has been possible to prove that water detection using NIRS technology is possible.

However, the prediction of water volume with this prototype is of low accuracy. The mode of operation of this prototype is to detect the absence and presence of water at a given level, thus being able to calculate an approximate water volume. In this way, as can be seen in Figure 3.12, if there is a single sensor module at a single level, the prototype will detect a change in volume at a single level.

Thus, as a solution to improve the accuracy of water volume prediction,

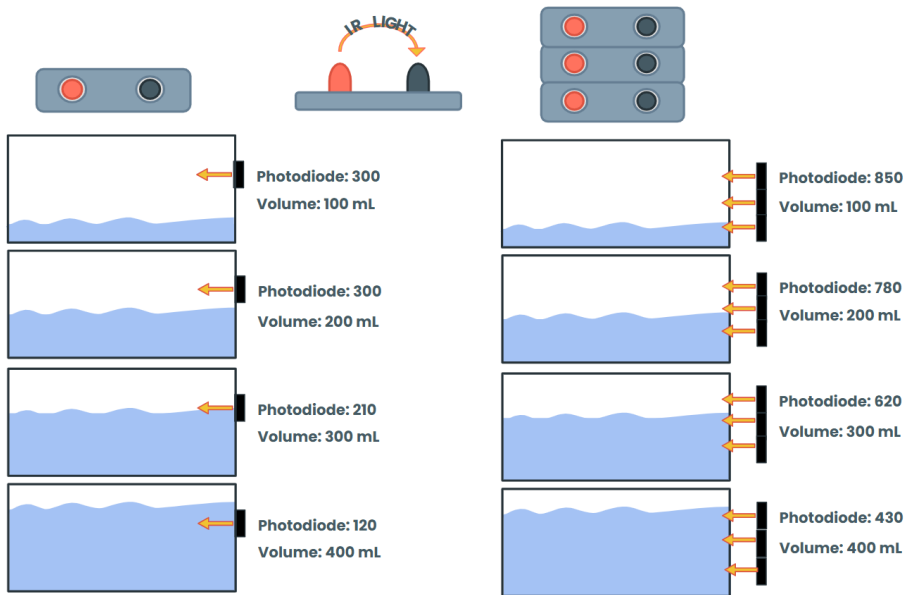


Figure 3.12: Exemplification of the resulting readings with a prototype with one sensory module (left) and with three sensory modules (right).

it was decided to increase the number of sensors to detect different water levels.

For this prototype, a total of 3 modules will be used. Each one will be vertically separated by 2 cm from the previous one. By using 3 modules, the operation mode of the software has been modified: instead of taking a single reading of the sensor as it was done with a single module, it will now take the sum of the readings of each photodiode. In this way, every time the water level reaches a photodiode, the total reading will be decreased by the presence of water.

These new sensors will have the same characteristics as the previous prototype: a distance of 2 cm between optodes and a gain of 2 from the photodiode amplifier.

3.5.1 Support structure

Until now, foam had been used to fix the sensor modules, as it was flexible and allowed easy movement of these elements during the calibration stage.

Once calibrated, it was decided to design a more rigid support structure

that would allow a more stable fixation of these elements in the position and distance between optodes and a fixation without movement with the water tank.

The first step was to design it in SolidWorks. To do this, measurements were made of the dimensions of the LEDs, photodiodes and optical filters.

The distance between optodes (horizontal direction) is 2 cm, as concluded in the previous calibration sections. For the distance between sensor modules (vertical direction), after a literature research, this used to vary between 1.5 and 4 cm. As the tank used is of a small size, a distance of 2 cm was finally used as well.

In addition, as can be seen in Figure 3.13, two holes have been added to the sides of the support structure with the idea of placing rubber bands or straps around the patient's waist to help fix it to the abdomen for future testing of the prototype.

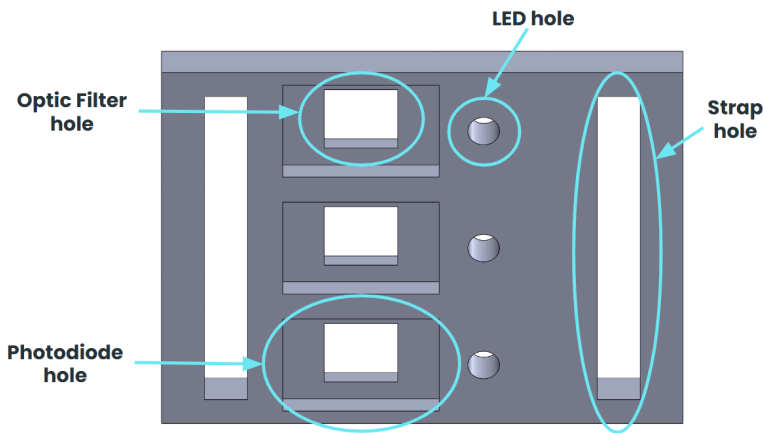


Figure 3.13: SolidWorks design of the supporting structure.

Once the design of the supporting structure was completed, it was 3D printed. The material used was PETG, as it is easy to use, durable, and above all it allows some flexibility, which was interesting if in the future the straps are used to fix the prototype to the patient's abdomen, as when tension is applied to both ends of the support structure, it may tend to flex.

Figure 3.14 shows the new prototype with the 3 sensory modules attached to the new support structure made with 3D printing.

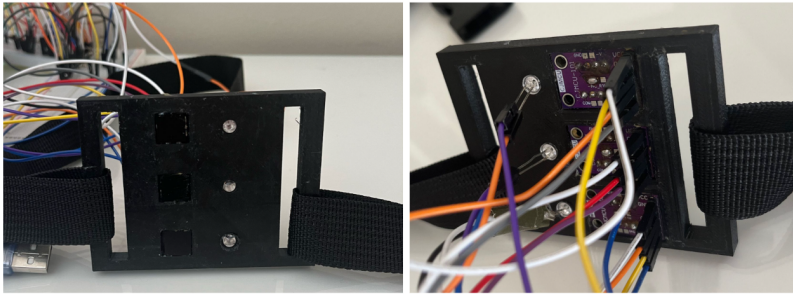


Figure 3.14: Prototype of 3 modules together with a social structure.

3.6 Algorithm

The ultimate goal of this prototype is to obtain the volume of urine contained in the bladder. For this purpose, as explained throughout this project, NIRS is used, where the interaction of infrared light with the different biocomponents is studied. In this way, the function of SafeBladder is to read the light detected by the photodiode and study its attenuation in order to estimate the volume of urine.

The attenuation of this light is due to the fact that a large part of the photons that pass through the tissue are absorbed by the water molecules (chromophores). However, there is another very important factor to take into account: scattering. When infrared light passes through human tissue, part of these photons do not follow the "expected pattern", but are scattered, and may not reach the receptive area of the sensor, leading to errors in obtaining the urine volume.

Initially, the idea was to use the Beer-Lambert law to relate the attenuation of the light to the concentration of the water molecules, i.e. only the absorption of the photons would be taken into account for the calculation of the volume. After analysing the existence of the scattering factor, it was decided to use Machine Learning to obtain this volume.

3.6.1 Steps for the creation of the algorithm

Data collection and preparation

Several readings were taken with different volumes of water (volumes from 0 to 700 mL with 50 mL variations), collecting the ADC values of the

CJMCU101 photodiode, the separation distance of the optodes, the gain of the amplifier, the accelerometer readings.

Approximately 140 readings were taken for each water volume level (i.e. approximately 2000 readings in total for the 14 different water volumes). The data was stored in a .csv format, shown in Figure 3.15.

```
Volume,SD,Gain,Photodiode,Acc (x),Acc (y),Acc (z)
0.0,2.0,2.0,234.0,0.24,-9.96,0.00
0.0,2.0,2.0,236.0,0.24,-10.04,0.00
0.0,2.0,2.0,238.0,0.24,-10.00,0.00
0.0,2.0,2.0,241.0,0.27,-10.00,0.00
0.0,2.0,2.0,240.0,0.27,-9.96,0.00
```

Figure 3.15: Extract of the csv file with the data collection of the prototype to be used with Machine Learning.

Machine learning modelling

Different machine learning models (Linear Regression, Random Forest and Support Vector Regression (SVR)) were used to be trained using the collected data to estimate the volume of water.

- **Linear Regression:** Linear regression is an easy-to-understand model that assumes a straight-line relationship between the input features (NIRS signals) and the output variable (bladder volume). The model finds the best weights for each feature to minimize the difference between the predicted and actual bladder volumes. Linear regression acts as a baseline model to evaluate the potential of using machine learning for estimating bladder volume.
- **Random Forest:** Random Forest is an ensemble learning technique that aggregates multiple decision trees to generate predictions. Each decision tree is trained on a random subset of the input features and data samples, which helps to mitigate overfitting and enhance generalization. The final prediction is made by combining the outputs of all the individual trees. Random Forest is renowned for its capability to manage non-linear relationships and identify intricate patterns within the data.
- **Support Vector Regression (SVR):** SVR is a variant of the Support Vector Machine (SVM) algorithm tailored for regression tasks. Its

objective is to find an optimal hyperplane that fits the training data while minimizing prediction errors. SVR includes a margin of tolerance (epsilon) for prediction errors, which helps in handling noise and outliers within the dataset. Utilizing kernel functions, SVR can model non-linear relationships by transforming input features into higher-dimensional spaces

An Ensemble model has also been implemented to combine the forecasts of the individual models in order to improve the accuracy of the forecasts.

Adjustment and Optimisation

A fine tuning of the models was performed for model optimisation, which is a technique to optimise the hyperparameters of a model and improve its performance. The tuning techniques used were Grid Search and Cross-Validation.

- Grid Search is a technique that explores a set of values for the hyperparameters and selects the most optimal of them. It is a robust but computationally expensive technique.
- Cross-validation is a technique that evaluates model performance by analysing different subsets of data. It is an efficient but less accurate technique.

The combination of Grid Search and Cross-Validation enables accurate optimisation and evaluation of Machine Learning models. Grid Search, although computationally expensive, ensures that all possible combinations of hyperparameters are explored to find the most optimal one. Cross-Validation, by evaluating model performance on different subsets of data, provides a robust measure of the model's ability to generalise to unseen data. Together, these techniques significantly improve model generalisation and help prevent overfitting, ensuring that the model does not overfit the training data and maintains good performance on new data.

Model Evaluation

Each trained model was evaluated for performance using the mean square error (MSE), the mean absolute error (MAE) and Root Mean Squared Error

(RMSE)

- **MSE**: Measures the average squared difference between the predicted and actual bladder volumes. It penalizes larger errors more heavily than smaller errors, making it sensitive to outliers. A lower MSE indicates a better fit of the model to the data, while a higher MSE indicates a poorer fit (mL^2).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **MAE**: Measures the average absolute difference between the predicted and actual bladder volumes. It provides an intuitive understanding of the model's prediction error in the original unit of measurement (mL).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **(RMSE)**: The square root of the MSE. It provides an estimate of the standard deviation of the prediction errors and is often preferred over MSE due to its interpretability in the original unit of measurement (mL).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

3.6.2 Machine Learning Code

For the creation of the Machine Learning code, the programming language Python has been used. The CSV file *general.csv*, where the data measured by the prototype has been stored, is read and loaded into a **DataFrame** called *data*. In this **DataFrame**, the variable **Volume** is separated from the rest of

```
data = pd.read_csv('general.csv')
X = data.drop('Volume', axis=1)
y = data['Volume']
```

Figure 3.16: Reading of the csv file with the stored readings of the prototype.

the variables. The latter variables are used to train the algorithm to try to predict the *Volume* variable.

The code snippet uses the *train_test_split* function to split the data set into training and test sets. The variables *X_train* and *y_train* contain 80% of the original data, which will be used to train the model, while *X_test* and *y_test* contain the remaining 20%, which will be used to evaluate the model's performance. The *random_state=42* seed ensures the reproducibility of the random split.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 3.17: *train_test_split* function fragment code.

The dictionary *models* defines three machine learning models (Figure 3.18):

- **Linear Regression:** Utilizes a basic linear regression model.
- **Random Forest:** Employs a random forest regression model wrapped in a *Pipeline* that includes feature standardization.
- **SVR (Support Vector Regressor):** Utilizes a support vector regressor model also wrapped in a *Pipeline* with feature standardization.

Each model is configured with its own preprocessing to ensure consistency in data treatment before modeling.

```
models = {
    'Linear Regression': LinearRegression(),
    'Random Forest': Pipeline([
        ('scaler', StandardScaler()),
        ('model', RandomForestRegressor(random_state=42))
    ]),
    'SVR': Pipeline([
        ('scaler', StandardScaler()),
        ('model', SVR())
    ])
}
```

Figure 3.18: Fragment of the code with the definition of the models.

The code fragment in Figure 3.19 defines the hyperparameters for fitting Random Forest and Support Vector Regression (SVR) models by hyperparameter search.

```
param_grids = {
    'Random Forest': {
        'model__n_estimators': [100, 200, 500, 1000], # Increased number of trees
        'model__max_depth': [None, 10, 20, 30, 50], # Deeper trees
        'model__min_samples_split': [2, 5, 10, 15], # Varied splits
        'model__min_samples_leaf': [1, 2, 4, 6], # Different minimum samples per leaf
        'model__max_features': ['auto', 'sqrt', 'log2'], # Number of features to consider at every split
    },
    'SVR': {
        'model__C': [0.1, 1, 10, 100, 1000], # Higher penalty values
        'model__gamma': ['scale', 'auto', 0.01, 0.1, 1], # Different gamma values
        'model__kernel': ['rbf', 'linear', 'poly', 'sigmoid'], # Including sigmoid kernel
        'model__degree': [2, 3, 4], # Degree for the polynomial kernel
        'model__epsilon': [0.1, 0.2, 0.5, 0.8], # Epsilon in the epsilon-SVR model
    }
}
```

Figure 3.19: Hyperparameters used for Random Forest and SVR models.

In the code snippet in Figure 3.20, it first goes through each model in the models list. If the model has hyperparameters defined in *param_grids*, it performs an exhaustive search for the best combination of those hyperparameters using cross-validation. Once the best model is found, it stores it in the models dictionary. If there are no hyperparameters for the model (as in Linear Regression), it simply trains the model with the training data.

```
for name, model in tqdm(models.items(), desc="Training models"):
    if name in param_grids:
        grid_search = GridSearchCV(model, param_grids[name], cv=5, scoring='neg_mean_squared_error', n_jobs=-1, verbose=2)
        grid_search.fit(X_train, y_train)
        models[name] = grid_search.best_estimator_
    else:
        model.fit(X_train, y_train) # Linear Regression does not need scaling or grid search
```

Figure 3.20: Fragment of the Machine Learning code where the search for the best model is performed.

The code fragment in Figure 3.21 is created an ensemble model and trained using sklearn's *VotingRegressor* to combine multiple regression models. The *VotingRegressor* is initialised with a list of estimators, each consisting of a name and a regression model from the models dictionary. It is then trained with the scaled training data (*X_train_scaled*) and labels (*y_train*). This process allows the *VotingRegressor* to combine the predictions of several models, improving the accuracy and robustness of the final result.

The code fragment in Figure 3.22 evaluates the performance of the models trained on the test set using the mean square error (MSE) metric.

```
ensemble = VotingRegressor(estimators=[(name, model) for name, model in models.items()])
ensemble.fit(X_train_scaled, y_train)
```

Figure 3.21: *VotingRegressor* model code.

```
for name, model in tqdm(models.items(), desc="Evaluating models"):
    if name in ['Random Forest', 'SVR']: # Models trained on scaled data
        predictions = model.predict(X_test)
    else: # Linear Regression does not use scaled data
        predictions = model.predict(X_test)
    mse = mean_squared_error(y_test, predictions)
    print(f"{name} Test MSE: {mse}")
```

Figure 3.22: Evaluation of the MSE of each model.

The proposed pipeline shown in Figure 3.23 integrates three fundamental stages for the construction of an advanced linear regression model:

- **Polynomial Features:** this stage introduces polynomial features, allowing to capture non-linear relationships between predictor variables. This inclusion is essential for modelling data that exhibit more complex behaviour than a simple linear relationship.
- **Standard Scaler:** Subsequently, standard scaling is applied to the features, ensuring that they are all on the same scale. This procedure is crucial to avoid variables with different magnitudes distorting the model estimation.
- **Ridge CV:** Finally, Ridge regression with cross-validation is used to optimise the regularisation of the model. This approach allows finding an optimal balance between the fit to the training data and the generalisation to new data, mitigating the risk of overfitting.

```
linear_model_pipeline = Pipeline([
    ('poly_features', PolynomialFeatures(degree=2, include_bias=False)),
    ('scaler', StandardScaler()),
    ('ridge', RidgeCV(alphas=[0.1, 1.0, 10.0], cv=5))
])
```

Figure 3.23: Code for the construction of the *linear_model_pipeline* model.

3.6.3 Algorithm code

The function of the algorithm created is based on connecting to the Arduino (the SafeBladder prototype), using several pre-trained machine learning models to make predictions based on this data, and then displaying the predictions of each model on the console.

The *load_models* function (Figure 3.24) is responsible for loading various machine learning models previously trained from joblib files.

```
def load_models():
    linear_model = load('linear_model.joblib')
    rf_model = load('rf_model.joblib')
    svr_model = load('svr_model.joblib')
    ensemble = load('voting_regressor.joblib')
    enhanced_linear_model = load('enhanced_linear_model.joblib')
    return linear_model, rf_model, svr_model, ensemble, enhanced_linear_model
```

Figure 3.24: Load code of the trained models..

The *make_predictions* function (Figure 3.25) takes as input a set of machine learning models and a set of input features. It then makes predictions using each of the models and returns a dictionary containing the model names as keys and the corresponding predictions as values.

```
def make_predictions(models, input_features):
    linear_model, rf_model, svr_model, ensemble, enhanced_linear_model = models
    predictions = {
        'Linear Regression': linear_model.predict([input_features]),
        'Random Forest': rf_model.predict([input_features]),
        'SVR': svr_model.predict([input_features]),
        'Ensemble': ensemble.predict([input_features]),
        'Enhanced Linear Model': enhanced_linear_model.predict([input_features])
    }
    return predictions
```

Figure 3.25: *make_predictions* function code.

The function *read_arduino_data* (Figure 3.26) is responsible for establishing a serial connection to the Arduino. It then reads each line of data from the Arduino and decodes it into a string using UTF-8. It then splits the data string into a list of values, assuming the data is in CSV format and separated by commas. Finally it converts all the values in the list to floating point numbers and stores them in a new list called data and returns a list containing all the numerical values read from the Arduino.

In addition, it was decided to print the photodiode readings to the console to see if the predictions correlated correctly.

```
def read_arduino_data():
    # Replace "COM7" with the actual port your Arduino is connected to
    with serial.Serial("COM7", baudrate=9600) as ser:
        data_string = ser.readline().decode("utf-8").strip()
        # Split the data string based on commas (assuming CSV format)
        data_list = data_string.split(",")

        # Extract photodiode value
        photodiode_value = float(data_list[2])

        # Print photodiode value
        print(f"Photodiode Value: {photodiode_value}")

        # Convert remaining data to floats (assuming remaining elements are numeric)
        data = [float(value) for value in data_list[3:]]

    return data
```

Figure 3.26: *read_arduino_data* function code.

The *get_input_features* function (Figure 3.27) is designed to acquire input features. In particular, it uses the *read_arduino_data* function to obtain data from the Arduino device.

```
def get_input_features():
    feature_names = ['SD', 'Gain', 'Photodiode', 'Acc (x)', 'Acc (y)', 'Acc (z)']
    input_features = read_arduino_data() # Use the read_arduino_data function here
    return input_features
```

Figure 3.27: *get_input_features* function code.

The presented code fragment in Figure 3.28 sets up an infinite loop that performs the following operations:

- Gets real-time data from the Arduino using the *get_input_features* function.
- Uses the obtained data to make predictions using a set of previously loaded models, using the *make_predictions* function.
- Prints the resulting predictions for each model.

- Waits for a 5 second interval before continuing with the next data reading.

This cycle allows continuous monitoring and prediction of the data acquired from the Arduino.

```
while True:
    # Get real-time data from Arduino every 10 seconds
    input_features = get_input_features()

    predictions = make_predictions(models, input_features)

    # Print predictions
    for model_name, prediction in predictions.items():
        print(f"{model_name} predicted Volume: {prediction[0]}")

    # Wait for 5 seconds before reading again
    time.sleep(5)
```

Figure 3.28: Looping code for continuous water volume prediction.

3.6.4 Results of the algorithm

The above steps were carried out for the training of two different algorithms: one for a single sensory module (the old prototype) and one for 3 sensory modules (the current prototype), in order to observe whether the algorithm has improved its performance.

Algorithm training for 1 module

The results of the machine learning models generated with a single sensory module are shown in Table 3.4:

Algorithm training for 3 modules

The results of the machine learning models generated with three sensory modules are shown in Table 3.5:

Table 3.4: Performance of the Machine Learning Models for 1 module

Model	MAE (mL)	MSE (mL ²)	RMSE (mL)
Linear Regression	100.5	16927.9	130.1
Random Forest	41.9	4356.1	66
SVR	46.2	5462.4	72.9
Ensemble	53.9	5959.1	77.2
Enhanced Lineal Model	73.7	9453.1	97.2

Table 3.5: Performance of the Machine Learning Models for 3 modules.

Model	MAE (mL)	MSE (mL ²)	RMSE (mL)
Linear Regression	45.2	3266.2	57.2
Random Forest	28.6	834.6	28.9
SVR	35.8	1576.3	39.7
Ensemble	29.4	1308.8	36.2
Enhanced Lineal Model	35.9	2072.6	45.5

Evaluation

These results indicate that the Random Forest model obtained the lowest error in the test set, suggesting a better predictive performance compared to the other models evaluated.

By comparing the results of the 1-module and 3-module Random Forest, we can see a significant drop in MSE from 4356 to 834. Even so, although this last MSE may seem quite high, it is important to consider the context and complexity of the problem in question. Given that we are dealing with the prediction of urine volumes in the bladder using NIRS, some level of error is to be expected due to factors such as light scattering and individual variations.

Thus, to see the results of volume prediction with the Random Forest model, readings were taken with the prototype from 50 mL volumes up to 700 mL. These readings were carried out with both the old prototype of one sensory module and the current prototype of three sensory modules, to observe the improvements produced.

These results are plotted in Figure 3.29, where the standard deviation

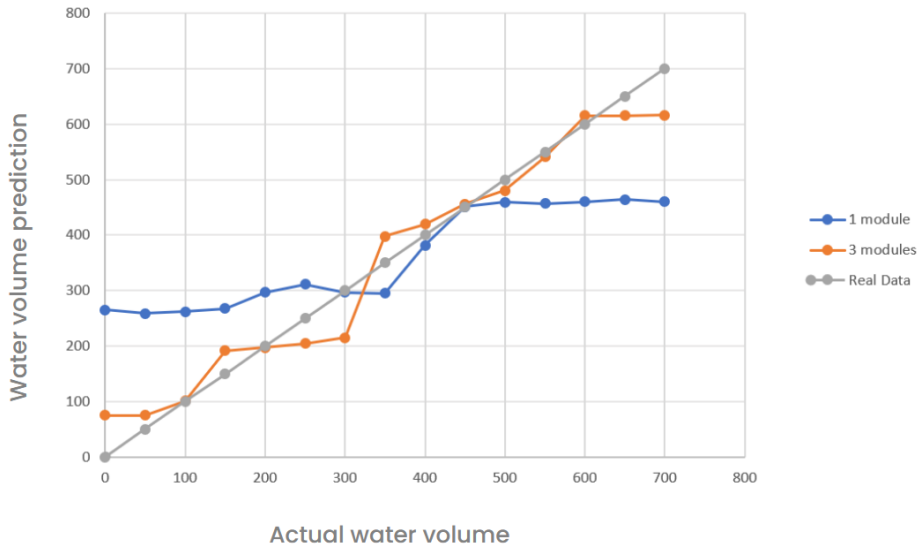


Figure 3.29: Graphical result of water volume prediction using 1 and 3 sensory modules.

for the 1 sensory module prototype is 139.39, and for the 3 sensory module prototype it is 43.85. Thus, we can conclude that the prediction is quite high and improves considerably as the number of sensors applied to the prototype increases, making it more sensitive to variations in water volume.

Although the accuracy is improved, there are some readings where the prediction has a larger error range, which can be solved in the future by using a larger number of sensors placed at a smaller distance from each other.

3.7 Experiments

This section details the experiments conducted to evaluate the efficacy and functionality of the designed bladder monitoring device. Two types of experiments were conducted: one using an *in vitro* model with silicone to simulate the properties of human skin, and the other *in vivo* with the participation of a human volunteer.

3.7.1 *In vitro* experiment

The first experiment was designed to evaluate the ability of the bladder monitoring device to function properly when a barrier simulating human skin is attached. For this purpose, silicone, known for its skin-like properties, was used as the test material.

Setup

The experiment was conducted in a controlled environment using a water tank to simulate a full bladder. Two silicone samples with different thicknesses were prepared: one of 1.5 cm and one of 3 cm. These samples were selected to evaluate the influence of the thickness of the "skin" on the performance of the device (Figure 3.30).

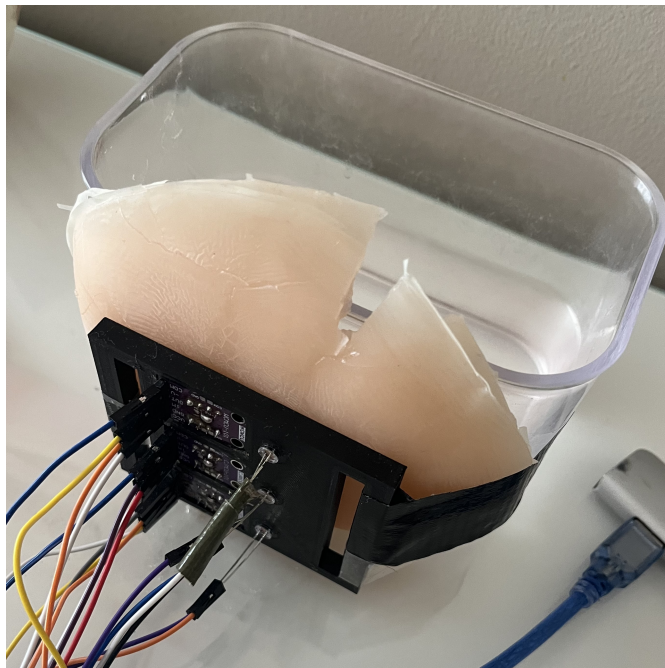


Figure 3.30: Setup of the *in vitro* experiment.

The silicone samples were placed on one side of the water tank so that the silicone acted as a barrier between the sensing modules of the device and the water inside the tank. This arrangement was designed to mimic the location of the device in the abdominal area, with the silicone representing the skin and underlying tissues of the abdomen.

The water tank was gradually filled, while continuous readings were taken from the device's integrated photodiode. The photodiode was used to detect variations in the received light intensity, which were expected to change as a function of the volume of water in the tank.

Results

The photodiode readings showed significant variations as the water tank filled, indicating that the device was able to detect changes in water volume through the silicone. However, a difference in accuracy was observed between the two silicone samples used.

With the 1.5 cm thick silicone sample (Figure 3.31), the photodiode readings accurately reflected the progressive filling of the water tank. In contrast, with the 3 cm thick silicone sample (Figure 3.32), a slight decrease in the accuracy of the photodiode readings was observed, indicating a decrease in the accuracy of detection of changes in liquid volume.

This decrease in accuracy is logical, as the prototype has been calibrated to operate in a certain environment (no silicon), so when introducing this medium, the signal is more distorted. However, as the prototype still detects variations in the volume of water in the presence of silicon, the prototype should be recalibrated for this new environment to make it more accurate.

The Table 3.6 shows the results of the in vitro experiment performed with the silicone (setup 2) and the results of the water volume prediction only with the water tank without the silicone (setup 1).

This table includes the mean of the predicted water volume together with its deviations. Also shown is the percentage error between the SafeBladder estimate and the actual water volume.

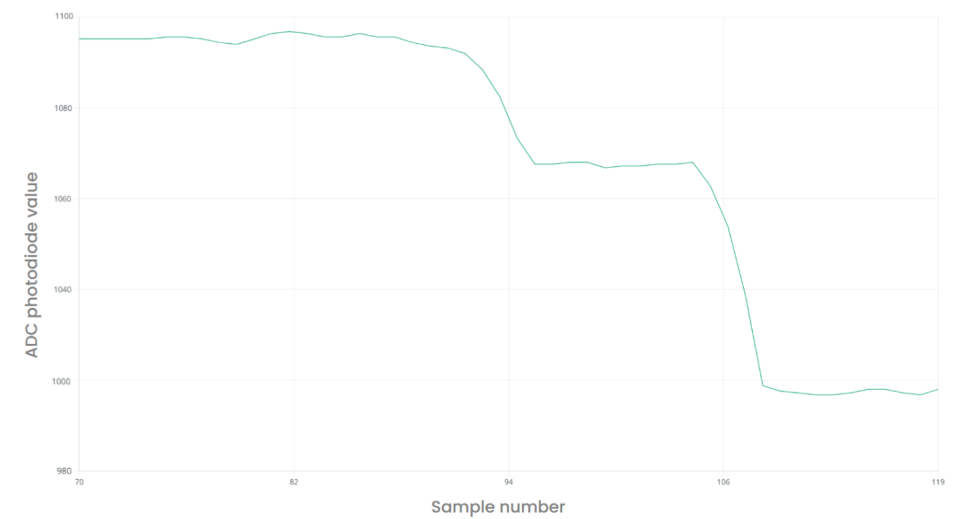


Figure 3.31: *In vitro* experiment with a 1.5 cm thick silicone.

Table 3.6: Prototype testing results for Setup 1 (Plastic Container) and Setup 2 (Silicone + Plastic Container)

Actual Volume (mL)	Setup 1		Setup 2	
	Predicted Volume (mL)	Error (%)	Predicted Volume (mL)	Error (%)
0	5 ± 2	-	8 ± 3	-
100	90 ± 10	10	85 ± 12	15
200	180 ± 15	10	170 ± 18	15
300	270 ± 20	10	255 ± 22	15
400	360 ± 25	10	340 ± 28	15
500	425 ± 30	15	400 ± 35	20
600	480 ± 35	20	450 ± 40	25

3.7.2 *In vivo* Experiment

The second experiment was conducted to evaluate the capability of the prototype bladder monitoring device under real-world conditions, using human subjects. This trial was aimed at comparing the bladder volume readings obtained with our prototype.

Setup

Two human subjects were selected to participate in the experiment. Each subject was tested using the BladderScan and our prototype bladder monitoring

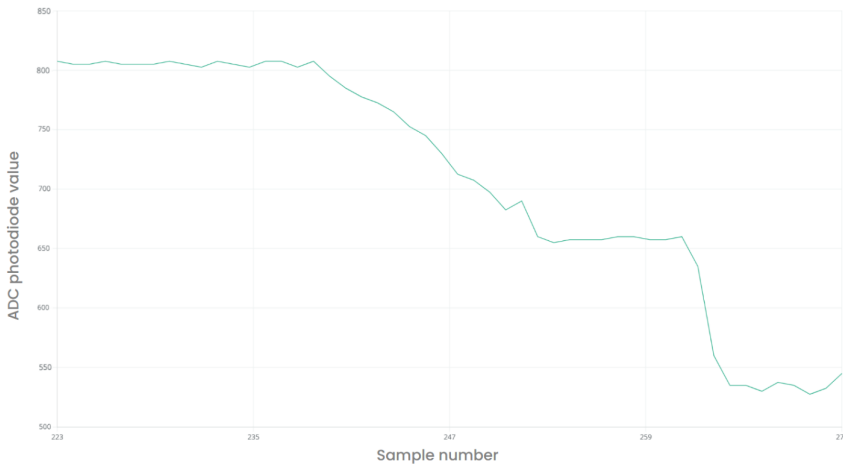


Figure 3.32: *In vitro* experiment with a 3 cm thick silicone.

device. The BladderScan (Figure 3.33) is a medical device commonly used in clinics and hospitals to non-invasively measure bladder volume. It works by emitting ultrasound that bounces off the walls of the bladder, allowing the volume of urine present in the bladder to be calculated with high accuracy.

First, each subject's bladder volume was measured using the BladderScan to establish an accurate baseline. Next, our prototype was used (Figure 3.34) to measure the same bladder volume. The readings obtained with our prototype were compared with those of the BladderScan.

Results

The bladder volume measurements obtained with our prototype were found to be incorrect when compared to the accurate BladderScan measurements. This discrepancy is attributed to the fact that our prototype is specifically calibrated to work with a transparent tank of water, rather than a human body, which affects the accuracy of the readings due to differences in light refraction and the optical properties of human tissues.

However, during the trial, an interesting behaviour was observed that indicated that the prototype was still able to detect certain changes. Each subject was asked to partially empty their bladder during the experiment. After the subjects went to the toilet and reduced the volume of urine in their bladder, confirmed by BladderScan, the prototype's photodiode readings showed a significant increase. This suggests that, although the exact volume



Figure 3.33: BladerScan device.

measurements were incorrect, the prototype could detect changes in the amount of fluid present in the bladder.

This trial was repeated with a second subject, with similar results: the absolute volume measurements were not accurate, but the prototype consistently detected the change in bladder volume when the bladder was emptied.

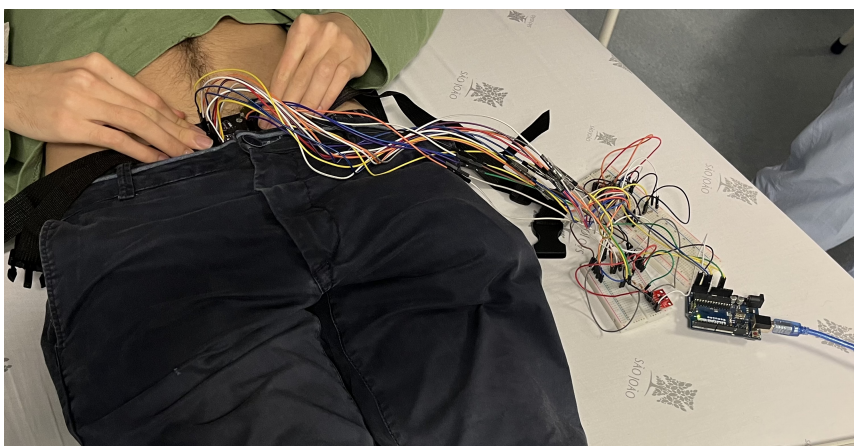


Figure 3.34: Subject using the SafeBladder prototype for testing.

Chapter 4

Mobile Application

The original concept behind the SafeBladder device was to develop a comprehensive mobile app to help users with Neurogenic Bladder (NB). This app would provide real-time information on bladder status, including bladder fullness monitoring and proactive alerts to indicate when they need to urinate. The aim was to provide users with greater awareness and control over their condition, thus improving their quality of life by offering an innovative and practical technological solution in the field of health.

4.1 Objectives of the Mobile Application

The objective of this application is to show in an efficient and simple way to the patient the filling status of his/her bladder. In turn, this application must send a notification to the patient when the urine volume level threshold is reached with sufficient time to indicate that he/she must perform his/her emptying.

The NIRS technology of the SafeBladder device allows to obtain a reading of the water absorption peaks and the concentration of the chromophores of interest, thus obtaining the volume of urine in the patient's bladder.

To ensure that there have been no reading errors and improve the efficiency of the device, the alarm will be displayed when this threshold is exceeded 10 consecutive times.

Figure 4.1 shows in a more simplified way the process cycle of the SafeBladder device operation together with the App.

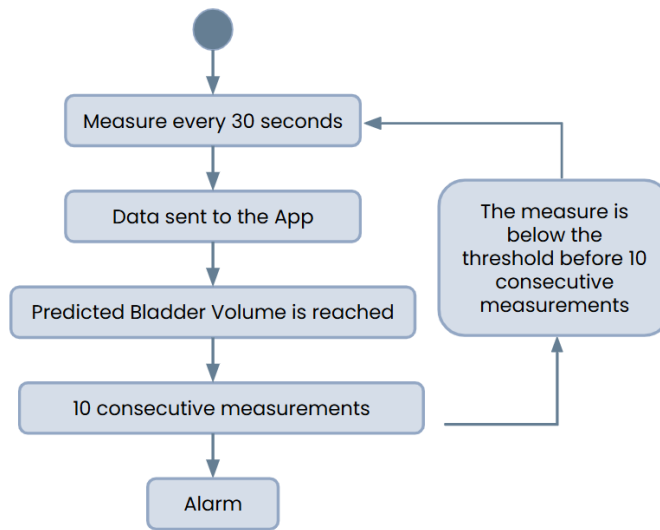


Figure 4.1: Process cycle of the SafeBladder device.

Figure 4.2 shows a diagram of how the connections are made for data transmission between the different devices, including the mobile application.

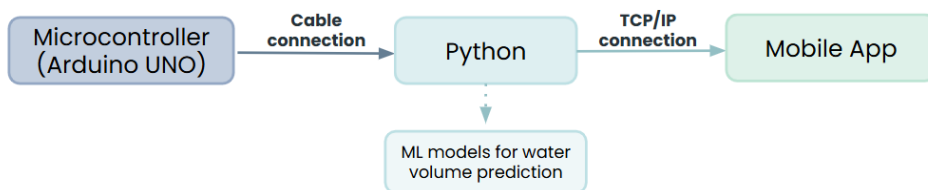


Figure 4.2: Diagram of the connections between devices.

- **Arduino sends data to the Python code by cable:** The Arduino is physically connected to the computer where the Python code is executed.
- **Processing and calculations in the Python code:** Once the Python code receives data from the Arduino, it uses pre-loaded machine learning models to perform calculations and predictions on this data.
- **Sending results via TCP/IP to the application:** After performing the calculations and obtaining predictions, the Python code sends the results (predictions) via a TCP/IP connection to the mobile application.

4.2 Preliminary Design

The following figures are sketches showing the preliminary idea of the application.

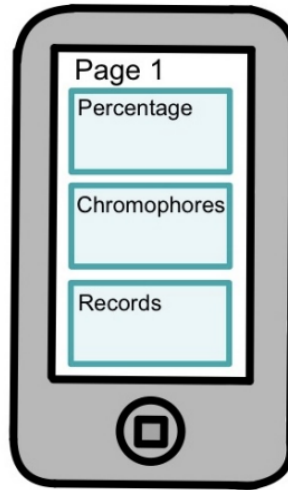


Figure 4.3: App main page.

Figure 4.3 shows is the main menu, in which the user will be able to see and access the three types of functions that the application performs.

- The first function or section will show the level of bladder filling as a percentage.
- The second section will show the concentration of chromophores updated every 30 seconds. This reading pulse time was established because the bladder, after liquid administration, has a filling rate of 4 mL/min, with 30 seconds being sufficient time to detect variations in bladder filling.
- The third section has been a major idea that was considered useful. In this section, the hours at which the bladder was emptied will be recorded, allowing the doctor or user to access this data easily whenever required.

Figure 4.4 shows in more detail the different content pages with the sections mentioned above that make up this application.

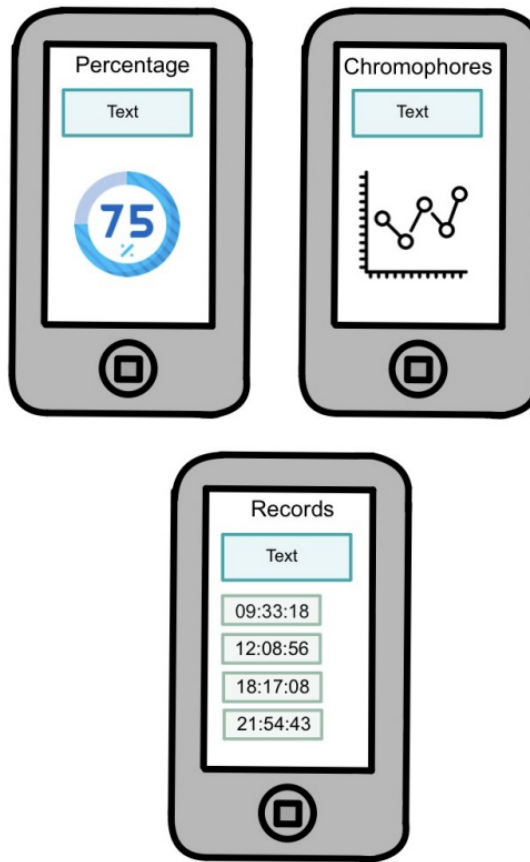


Figure 4.4: App sections pages.

- On the page related to the percentage of fill, a chart will be displayed in the form of a Donut that is updated every 30 seconds. In this way, the patient can efficiently understand how much he/she has left to perform bladder emptying.
- On the page related to the concentration of chromophores, a line graph will be shown that is updated every 30 seconds to be able to observe the variation of this concentration.
- The last page will show a table with the hours at which the voiding has been performed.

4.3 Integrated Development Environments

For the creation and design of the application, a search for integrated development environments (IDE) was carried out, where Android Studio and Visual Studio stand out.

Finally, it was decided to use Visual Studio for its advantages [37].

- Visual Studio allows the design of mobile applications for multiple platforms (Windows, iOS and Android). In contrast, Android Studio is specialized for developing applications exclusively for the Android platform, making it less versatile.
- Visual Studio offers a wide variety of supported programming languages.

4.4 Framework

The framework chosen for the development of this mobile application was Xamarin.Forms which has the following features:

- It allows the development of applications for both Android and iOS using a single code base.
- The working method is based on the use of a Front-end and a Back-end, which are parts of the application with different functionalities.

The Front-end is related to the visual part, what the user sees, and uses a XAML markup language. The Back-end focuses on providing functionality to the application and involves working with the C# programming language. C# is recognized for its relative ease of learning and utilization [38].

It is important to understand the hierarchical structure of the elements of the Back-end in order to further define its design (Figure 4.5) [39].

- Objects: in programming, objects are instances that combine data (properties) and actions (methods).
- Properties: modify the fields of a class.
- Methods: these are the operations or actions that are carried out.
- Classes: are object constructors that define the structure and behaviour of an object.

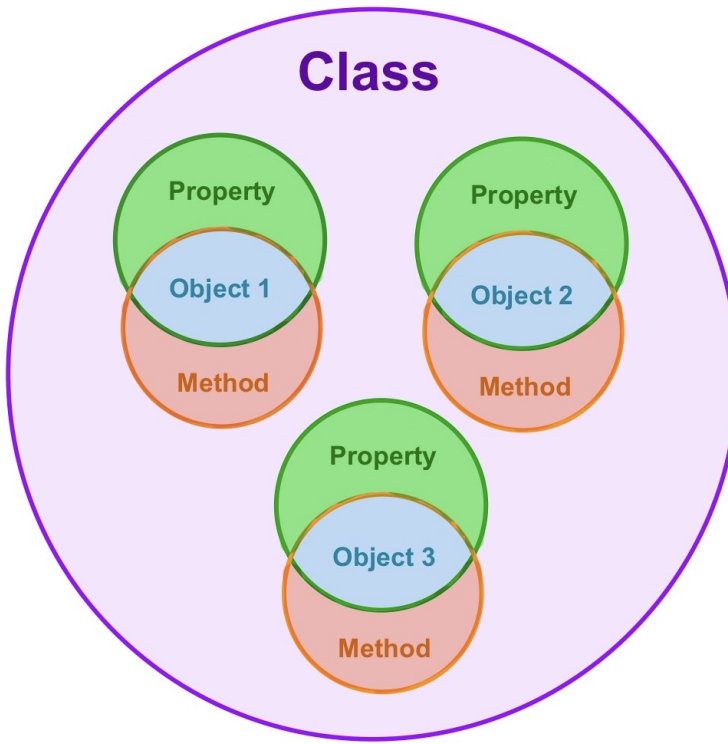


Figure 4.5: Structures of the elements of the Back-end.

4.5 Emulator

To visualize the outcomes of code generated through Visual Studio, acquiring an emulator is essential. This software facilitates the execution of the created application by simulating a mobile phone environment.

After looking for different PC emulators, the MEmu Play emulator was the one chosen for this project, because of its ease of use with a very intuitive interface, its high performance and because it is a very light software [40].

4.6 Algorithm Modification for Mobile Application Integration

A modification was made to the data acquisition and processing system of the Arduino, with the aim of facilitating its integration and visualisation in

the mobile application.

Initially, the system was configured to acquire data from a sensor connected to the Arduino using a direct data connection. While this method is effective for local applications, it has limitations when data needs to be accessed from remote locations or viewed on mobile devices. To overcome these limitations and allow for a more flexible integration of the system, it was decided to implement a solution based on TCP/IP communication.

TCP/IP communication is a fundamental standard for data transmission over networks, such as the internet. Thus, a server was set up on the Arduino embedded system. This server enables external applications, such as a mobile app, to connect and retrieve real-time data efficiently and securely.

```
# Setup TCP server
host = '0.0.0.0' # Listen on all network interfaces
port = 12345     # Port number for the server
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((host, port))
server_socket.listen(1)
print(f"Listening on {host}:{port}")

# Accept a client connection
client_socket, address = server_socket.accept()
print(f"Connection from {address}")

try:
    while True:
        input_features = get_input_features()
        if input_features is not None:
            prediction = make_predictions(models, input_features)
            client_socket.sendall(str(prediction).encode('utf-8'))
        else:
            print("Error reading from Arduino. No data sent.")
            time.sleep(5)
except (KeyboardInterrupt, ConnectionResetError, serial.SerialException) as e:
    print(f"Connection closed or Serial error: {e}")
finally:
    client_socket.close()
    server_socket.close()
```

Figure 4.6: Algorithm modification for Mobile Application Integration.

The code in Figure 4.6 implements a TCP server in Python that acts as an intermediary between an Arduino device and a client application, such as a mobile app. The server is configured to listen for incoming connections on all available network interfaces (host = '0.0.0.0') on the specific port 12345 (port = 12345).

When a client connects to the server, the connection is accepted via *server_socket.accept()*, which creates a new socket (*client_socket*) for communication with that client. A message is displayed indicating the established connection and the client address.

Within the loop (while True), the server calls the function *get_input_features()* to retrieve data from the Arduino. If the data is valid, it uses pre-loaded machine learning models to make predictions via *make_predictions(models, input_features)*. Subsequently, it sends these predictions to the client using *client_socket.sendall(str(prediction).encode('utf-8'))*.

In the final part of the code, it is ensured that the communication channels of both the client and the server are closed correctly. This is crucial to free up resources and ensure that the programme terminates in an orderly manner and without leaving open connections.

4.7 Connection

In order for the application to display the prototype data to the user, it is necessary to make a connection between app-arduino to send the data.

The main idea was that the connection would be made wirelessly via Bluetooth, however, due to the limited time for the development of the prototype, it has not been possible to incorporate the Bluetooth module to the SafeBladder prototype.

Thus, the only connection option was a physical cable connection instead of wireless.

4.7.1 Connection Selection

A new content page was designed in the application where the different connection options (Bluetooth, Wi-Fi, Cable) are displayed. The intention was to incorporate the wireless options so that in the future the application and the prototype can be developed using them, but for the moment, the only valid option for connection is by cable.

4.7.2 Cable Connection Page

The "Cable Connection" page is intended to establish a physical connection to the Arduino device.

Within the *StackLayout*, there are three *Entry* controls that allow users to enter information. One is used to enter the IP address of the server and the other for the port number to connect to.

In the context of this project, where Python acts as an intermediary server to facilitate communication between the mobile application and the Arduino device, the inclusion of the IP address in the user interface of the application is essential for the following reasons:

- **Server ID:** The IP address provides the unique location of the Python server on the local network. Including the IP address in the user interface tells the mobile application where to direct its connection requests.
- **Connection establishment:** In order for the mobile application to communicate with the Python server, it needs to establish a network connection to it.
- **Communication coordination:** Once the connection is established, the Python server acts as an intermediary to coordinate the communication between the mobile application and the Arduino device.

On the other hand, the port is needed to specify the specific connection point on the Python server where the mobile application should send its data or requests.

4.8 Results

Screenshots obtained from the MEmu play emulator of the result of the designed application are shown in the following figures.

In Figure 4.7 it is shown the main menu of this mobile application. If the user scrolls down over this menu, he/she will be able to see the different sections that make up this application and access them by clicking on the frame.

In Figure 4.8 it is shown the "Percentage" and "Chromophores" sections. Both contain a short description of their functionality so that the user can understand what each section is about.

The "Percentage" section contains a chart in the form of a Donut. This chart is updated every 30 seconds with the filling level of the bladder. When

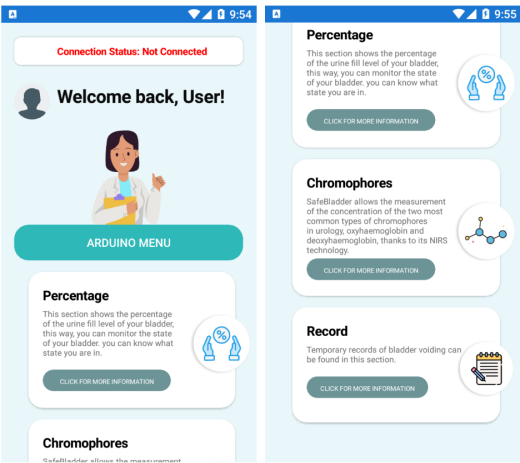


Figure 4.7: Scrolling of the main menu screen.

the chart reaches a threshold percentage, the application displays an alert message to the user to warn them to empty their bladder.

The section "Chromophores" shows two line diagrams with the evolution of the other two chromophores of interest; O₂Hb and HHb.

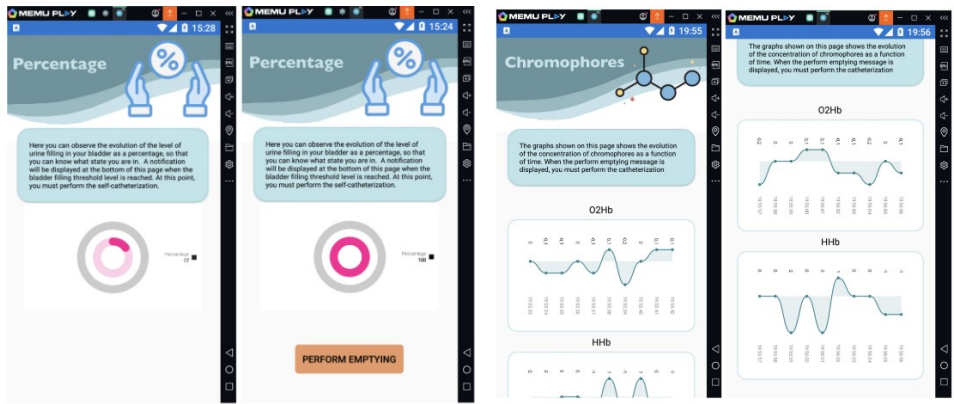


Figure 4.8: Percentage and chromophores content page.

Figure 4.9 shows a screenshot of the "Record" section, where as in the previous sections, its function is summarised in a frame for a more understandable use by the user, and in the lower part, the frames with the times at which the emptying was performed are shown.

Figure 4.10, shows the screenshot of the configuration option selection

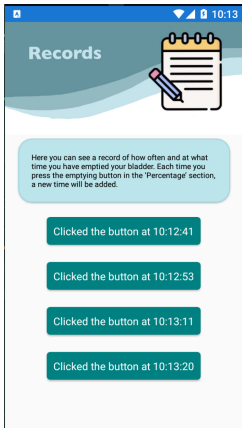


Figure 4.9: Record content page.

page and the cable connection configuration page respectively. By entering the correct data, the data transmission between the Arduino and the application was successful, allowing the percentage of water in the tank to be updated.

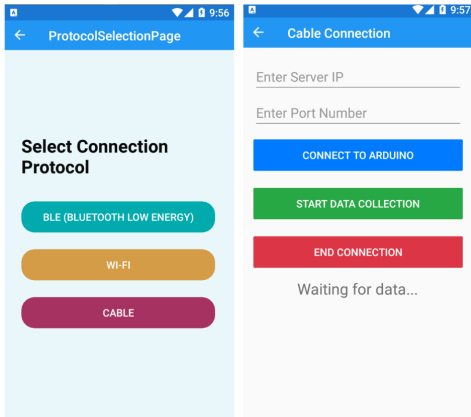


Figure 4.10: Connection type configuration page.

Chapter 5

Conclusions

5.1 Conclusions

In this Master Thesis, a prototype for bladder monitoring has been developed. The process started with the careful selection and assembly of components, resulting in a device composed of three sensing modules employing near-infrared spectroscopy (NIRS) technology. This approach enabled the creation of a dedicated sensing system for bladder volume monitoring.

In parallel, a machine learning algorithm designed to predict the volume of water in the bladder was developed. This algorithm was trained with data obtained from preliminary experiments, thus improving its accuracy and predictive capacity. The integration of advanced machine learning techniques represents a crucial step towards automation and accuracy in bladder monitoring.

In addition, a functional mobile application was created that allows users to monitor the status of their bladder and visualise the estimated volume in real time. This app effectively connects to the prototype, providing an intuitive and accessible interface for the end user.

Experiments to evaluate the prototype included *in vitro* and *in vivo* tests. In the *in vitro* tests, using silicone models of different thicknesses, the device demonstrated its ability to detect changes in water volume, although with lower accuracy at higher thicknesses. In *in vivo* tests with human subjects, the readings of the prototype were compared with those of the BladderScan device. Although the absolute volume measurements were inaccurate due to the initial calibration for a tank of water, the prototype

was able to detect changes in bladder volume upon emptying, indicating its potential with additional adjustments to improve accuracy in human settings.

In summary, the development of this prototype has proven to be a promising approach for non-invasive bladder volume monitoring. Although *in vivo* results showed limitations in accuracy, the device has demonstrated its ability to detect changes in fluid volume. These findings suggest that, with additional calibrations and adjustments, it is possible to improve the accuracy and clinical utility of the device. Furthermore, compared to other existing commercial monitoring systems, the significant advantage of this prototype is its cost-effectiveness, thanks to the simple technology it employs.

5.2 Future Work

In this project, an initial prototype of SafeBladder has been successfully developed, but it is important to recognise the limitations and areas of improvement needed for future iterations. As this was the first phase of the device's creation, inexpensive components were used in its construction. For example, the optical filters used do not effectively block ambient light due to their low cost, which affects the accuracy of the measurements. To improve the performance of the device, it will be necessary to use higher quality optical filters that effectively eliminate interference from ambient light.

Also, the LEDs used in this initial phase are inexpensive and not powerful enough to penetrate abdominal tissue effectively and reach the bladder. For future versions of the prototype, it will be crucial to integrate more powerful LEDs that allow better light penetration, thus ensuring more accurate detection of the bladder volume.

The environment in which the prototype has been tested was a tank of water, which does not accurately replicate the conditions of a human body. In the future, it will be essential to recalibrate the device specifically for human anatomy. This will include collecting data from multiple human subjects to train the machine learning algorithm, thereby improving the accuracy and reliability of bladder volume predictions in clinical settings.

In addition, the device cannot currently be considered wearable due to the amount of cables required for its operation and the wired connection to the mobile application. In order to make SafeBladder a wearable and

user-friendly device, it will be necessary to develop a more compact design and eliminate wires by integrating a Bluetooth module. This will enable wireless connectivity with the mobile application, significantly improving usability and user convenience.

These improvements and future refinements are essential to take SafeBladder from the initial prototype to a clinically viable and commercially successful device. The implementation of higher quality components, recalibration with human data and transformation into a wearable device are crucial steps to achieve accurate and non-invasive bladder volume monitoring, providing a significant advance in the field of medical monitoring.

References

- [1] C. Verpoorten and G. M. Buyse, “Neurogenic bladder: medical treatment,” *Pediatric Nephrology (Berlin, Germany)*, vol. 23, no. 5, pp. 717–725, 2008, doi: 10.1007/s00467-007-0691-z.
- [2] N. Kinnear, D. Barnett, M. O’Callaghan, K. Horsell, J. Gani, and D. Hennessey, “The impact of catheter-based bladder drainage method on the risk of urinary tract infection in spinal cord injury and neurogenic bladder: a systematic review,” *Neuourology and Urodynamics*, vol. 39, no. 2, pp. 854–862, 2020, doi: 10.1002/nau.24253.
- [3] P. T. Dorsher and P. M. McIntosh, “Neurogenic bladder,” *Advances in Urology*, pp. 1–16, 2012, doi: 10.1155/2012/816274.
- [4] S.-C. Shin, J. Moon, S. Kye, K. Lee, Y. S. Lee, and H.-G. Kang, “Continuous bladder volume monitoring system for portable applications,” *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, doi: 10.1109/EMBC.2017.8037840.
- [5] P. Ochodnický, B. Uvelius, K.-E. Andersson, and M. C. Michel, “Autonomic nervous control of the urinary bladder,” *Acta Physiologica (Oxford, England)*, vol. 207, no. 1, pp. 16–33, 2013, doi: 10.1111/apha.12010.
- [6] K. Lee and H. Kobayashi, “Autonomic dysfunction in spinal cord injury,” *Juntendo Medical Journal*, vol. 62, no. 5, pp. 370–376, 2016, doi: 10.14789/jmj.62.370.
- [7] P. Fechner, J. Lockl, N. Ruhland, T. Zurl, and T. Zwede, “A model to predict the amount of urine in the bladder based on tracking data

- generated by the application,” *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2952–2954, 2020, doi: 10.1109/BIBM49941.2020.9313381.
- [8] J. N. Panicker, “Neurogenic bladder: epidemiology, diagnosis and treatment,” *Seminars in Neurology*, vol. 40, no. 5, pp. 569–579, 2020, doi: 10.1055/s-0040-1713876.
- [9] A. Manack, S. P. Motsko, C. Haag-Molkenteller, R. R. Dmochowski, E. L. G. B.-A. Nguyen-Khoa, and J. K. Jones, “Epidemiology and health care utilisation of patients with neurogenic bladder in a us claims database,” *Neurourology and Urodynamics*, vol. 30, no. 3, pp. 395–401, 2011, doi: 10.1002/nau.21003.
- [10] Roswell Park Comprehensive Cancer Center, “What is a foley catheter?” <https://www.roswellpark.org/cancertalk/202101/whats-foley-catheter>, s/f, online, Accessed: 3 October.
- [11] E. Lamin and D. K. Newman, “Clean intermittent catheterisation review,” *International Urology and Nephrology*, vol. 48, no. 6, pp. 931–939, 2016, doi: 10.1007/s11255-016-1236-9.
- [12] Kenhub, “Urinary bladder,” <https://www.kenhub.com/es/library/anatomia-es/vejiga-urinaria>, online, Accessed: 12 October.
- [13] N. J. Lanzotti, M. A. Tariq, and S. R. Bolla, “Bladder physiology,” *StatPearl*, 2022, pMID: 30860768.
- [14] C. William, “Physiology of the urinary bladder and urethra,” *Annals of Internal Medicine*, vol. 312, 1980, doi: 10.7326/0003-4819-92-2-312.
- [15] K. Holmes-Martin, Kieran, M. Zhu, and S. X. y Faezeh Arab Hassani, “Avances en soluciones de dispositivos electrónicos de asistencia para urología,” *Micromáquinas*, vol. 13, no. 4, p. 551, 2022, doi: 10.3390/mi13040551.
- [16] M. Z. Nasrabadi, H. Tabibi, M. Salmani, M. Torkashvand, and E. Zarepour, “A comprehensive survey on non-invasive wearable bladder volume monitoring systems,” *Med Biol Eng Comput*, vol. 59, pp. 1373–1402, 2021, doi: 10.1007/s11517-021-02395-x.

- [17] S. J. Matcher, M. Cope, and D. T. Delpy, "Use of water absorption spectra to quantify changes in the concentration of tissue chromophores in near infrared spectroscopy," *Physics in medicine and biology*, vol. 39, no. 1, pp. 177–196, 1994, doi: 10.1088/0031-9155/39/1/011.
- [18] Novioscan, "How it works - sens-u kids bladder sensor," <https://novioscan.com/how-it-works/>, online, Accessed: 29 November.
- [19] S. Hofstetter, M. Zilezinski, A. Wolf, D. Behr, D. Paulicke, D. Stoevesandt, K. Schwarz, S. Schönbürg, and P. Jahn, "Dfree ultrasonic sensor in supporting quality of life and patient satisfaction with bladder dysfunction," *International Journal of Urological Nursing*, vol. 17, no. 1, pp. 62–69, 2023, doi: 10.1111/ijun.12334.
- [20] S. Star, "Ces 2019: Technological advances for health," <https://sonorastar.com/2019/07/09/ces-2019-avances-tecnologicos-para-la-salud/>, online, Accessed: 30 November.
- [21] Verathon, "Verathon medical equipment - refurbished medical equipment," <https://www.somatechnology.com/spanish/fabricante/equipo-medico-verathon/>, online, Accessed: 30 November.
- [22] NICE, "Bladderscan bvi 9400 3d portable ultrasound scanner for measuring bladder volume," <https://www.nice.org.uk/advice/mib50>, 2016, online, Accessed: 18 November.
- [23] H. G. Jo, B. H. Park, D. Y. Joung, J.-K. H. Jung Ki Jo, W. Y. Choi, and K. K. Park, "Forward-looking ultrasound wearable scanner system for estimation of urinary bladder volume," *Sensors*, vol. 21, no. 16, p. 5445, 2021, doi: 10.3390/s21165445.
- [24] M. Reichmuth, S. Schürle, and M. Magno, "A non-invasive wearable bioimpedance system to wirelessly monitor bladder filling," *2020 Design, Automation and Test in Europe Conference and Exhibition*, 2020, doi: 10.23919/DATE48585.2020.9116378.
- [25] M. K. Kim, H. Kim, Y. S. Jung, K. M. A. Adem, S. S. Bawazir, C. Stefanini, and H. J. Lee., "Implantable bladder volume sensor

- based on resistor ladder network composed of conductive hydrogel composite,” *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2017, doi: 10.1109/EMBC.2017.8037177.
- [26] J. N. Weaver, J. C. Alspaugh, and B. Behkam, “Toward a minimally invasive bladder pressure monitoring system: Model bladder for in vitro testing,” *International Conference on Biomedical Robotics and Biomechatronics*, 2010, doi: 10.1109/BIOROB.2010.5625981.
- [27] S. Majerus, A. S. Basu, I. Makovey, H. Z. Peng Wang, C. Zorman, W. Ko, and M. S. Damaser, “Wireless bladder pressure monitor for closed-loop bladder neuromodulation,” *IEEE SENSORS*, 2016, doi: 10.1109/ICSENS.2016.7808967.
- [28] S. Hannah, P. Brige, A. Ravichandran, and M. Ramuz, “Conformable, stretchable sensor to record bladder wall stretch,” *ACS Omega*, no. 1, 2019, doi: 10.1021/acsomega.8b02609.
- [29] B. Molavi, B. Shadgan, A. J. Macnab, and G. A. Dumont, “Noninvasive optical monitoring of bladder filling to capacity using a wireless near infrared spectroscopy device,” *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 3, pp. 325–333, 2014, doi: 10.1109/TB-CAS.2013.2272013.
- [30] R. Garcia-Sotero, P. Carranza-Castro, J. Gutierrez-Salina, L. Garcia-Ortiz, and S. Hernandez-Rodriguez, “Aplicacion en medicina de la espectroscopia de infrarrojo cercano,” *Medicina interna de Mexico*, vol. 28, no. 4, pp. 365–370, 2012.
- [31] A. Koven and S. Herschorn, “Nirs: Past, present, and future in functional urology,” *Current Bladder Dysfunction Reports*, vol. 17, no. 4, pp. 241–249, 2022, doi: 10.1007/s11884-022-00665-4.
- [32] A. Pellicer and M. D. C. Bravo, “Near-infrared spectroscopy: A methodology-focused review,” *Seminars in Fetal and Neonatal Medicine*, vol. 16, no. 1, pp. 42–49, 2011, doi: 10.1016/j.siny.2010.05.003.
- [33] P. Gupta, “Nirs based bladder volume sensing for patients suffering with neurogenic bladder dysfunction,” <http://arxiv.org/abs/1807.03145>, 2018, online, Accessed: 26 September.

-
- [34] P. M. O. Valinhas, “Safebladder - device for neurogenic bladder control- Experimental and computational design report, degree in Mechanical Engineering, Faculty of Engineering, University of Porto,” 2023.
- [35] A. J. Macnab, “Applications of near infrared spectroscopy in urology,” *Nova Science Publishers Inc.*, p. 127–168, 2014, doi: 10.1021/acso-mega.8b02609.
- [36] T. Instruments, “Opt101 datasheet, product information and support,” <https://www.ti.com/product/OPT101>, online, Accessed: 28 January.
- [37] (s/f), “Visual studio development features,” <https://visualstudio.microsoft.com/es/vs/features/?tab=solutions>, Accessed on 27 February 2023.
- [38] Microsoft, “What is xamarin.forms?” <https://learn.microsoft.com/es-es/xamarin/get-started/what-is-xamarin-forms>, Accessed on 28 February 2023.
- [39] A. Narayanaswamy, “Description of classes, methods and properties in c#,” <https://www.developer.com/microsoft/c-sharp/understanding-classes-methods-and-properties-in-c/#>, Accessed on 27 September 2022.
- [40] MEmu, “How to use android applications on pc,” <https://www.memuplay.com/blog/es/usar-aplicaciones-android-en-pc.html>, Accessed on 9 January 2022.