Università degli Studi di Padova
Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Bioingegneria della Riabilitazione

Development of a mobile application for the automation of rehabilitation tasks using data collected from a wearable IMU device

Relatore: Giacomo Cappon
Correlatore: Zimi Sawacha, Elena Pegolo

Laureando:
Mattia Zanot
Matricola: 2026804

03 Aprile 2023
Anno Accademico 2022/2023

“Never apologize for trying hard, it’s an insult to your determination”

-Might Dai

INDEX

Summary

Lo studio condotto verte nella creazione di una applicazione Android/iOS che consenta la raccolta dati durante l'esecuzione di task riabilitative tramite un sensore posto nella zona lombo sacrale. Tale applicazione vuole sostituire l'attuale metodo di acquisizione che viene eseguito con l'utilizzo di un semplici approcci quali un cronometro tascabile.

In particolare verrà implementata una connessione di tipo BLE (Bluetooth Low Energy) e la trasmissione tra sensore-applicazione verrà gestita seguendo il protocollo di comunicazione proposto dall'azienda stessa che ha offerto il sensore.

Tale applicazione avrà a disposizione funzionalità di persistenza dati, analisi di task precedenti, comparazione di task ed in fine come output presenterà dei valori scelti per la descrizione della qualità di esecuzione del task integrando algoritmi di processing dedicati.

Abstract

This study focuses on creating an Android/iOS application that allows for data collection during the performance of rehabilitation tasks using a sensor placed in the lumbosacral area. This application aims to replace the current data acquisition method, which is performed using simple approaches such as a handheld stopwatch. In particular, a BLE (Bluetooth Low Energy) connection will be implemented and the transmission between the sensor and the application will be managed following the communication protocol proposed by the company that provided the sensor. This application will have data persistence, analysis of previous tasks, comparison of tasks, and finally it will present selected values for describing the quality of task performance, integrating dedicated processing algorithms.

1. Introduction

1.1 Rehabilitation Task

In rehabilitative medicine, the assessment of abilities is often considered more important than a detailed analysis of forces, joint angles, and muscle activation. [3] Mobility limitations are typically present as early manifestations of a disability process and can predict the progression of disability.[3] Various tests have been proposed to assess these limitations, including observation of an individual performing selected motor tasks and providing semi-quantitative measures based on predefined scales.[3]

In our daily life, we perform complex movements such as getting up from a chair, moving in bed or climbing stairs, automatically.[1] These postural transitions are delicate acts, especially given the problems that may arise during their execution, such as the risk of falling or potential harm that can occur in elderly patients.[1] This is particularly true for the Sit-to-Stand task, i.e. is a movement in which the subject must start sitting on a chair and, trying to use as little help as possible, put themselves in an upright position.

The execution of the transition from a seated to standing position requires the ability to maintain balance along with the production of sufficient muscle force to lift the body's center of mass (COM) from a seated to an upright position.[5] With aging, there is a deterioration of muscle strength, joint mobility, balance performance, and the ability to rise from a chair, which increases the risk of falls.[5]

The frequency with which this latter transaction occurs over the course of a day makes the task one of the main activities that can be used to measure the patient's physical independence. It is therefore essential to analyze and evaluate this movement with advancing age, illness, or disability.[1]

Movement analysis provides biofeedback to patients and healthcare providers, thus playing a role in patient empowerment and self-motivation.[2] Assessment of outcomes in clinical studies allows for the identification of safe or unsafe procedures and differentiation of therapeutic options to identify the best alternative.[2] Evidence-based medicine relies on routine, valid, and responsive assessment methods, including domains of function and activity as the main intervention goal of orthopedics, thus, knowledge, selection, and application of adequate outcome assessment methods are becoming increasingly relevant in clinical practice and research.[2]

Several studies have determined Sit-to-Stand performance and its correlation with falls in stroke patients, and a few studies also suggest an increased risk of falls in the elderly who have difficulty rising from a chair.[5]

Figure 1 In this figure, the 4 execution steps of the Sit-to-Stand are shown: an initial phase of rest, the twisting of the torso to start the movement, the phase prior to the final one, and the upright positioning.

In 2009, approximately 2.2 million nonfatal fall injuries were reported among the elderly and were treated in emergency departments, and more than 581,000 of these patients were hospitalized.[5] In addition, in 2007, over 18,000 elderly people died from fall-related injuries. The association between falls and difficulty in performing chair lift is not debatable, as both seem to indicate deficits in postural control ability.[5]

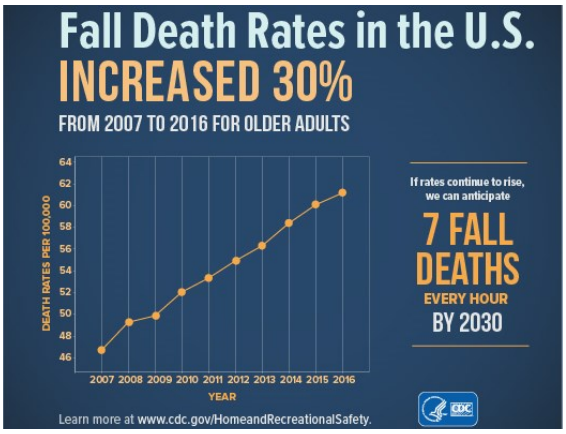


Figure 2 Graph that represented a prediction for elderly people who died due to a fatal fall in the US from a 2007-2016 study.

In accordance with the literature and studies undertaken, movement analysis using non-invasive methods is performed using multiple tools and systems, such as motion capture systems and/or wearable sensors applied directly to the patient. [2]

The evaluation and the possibility of continuous monitoring of the elderly subject could, for example, allow to act adequately in case of a fall, to detect which motor behaviors led the subject to fall and to apply monitoring techniques to prevent the event from recurring.[1] For this reason, measurement procedures based on simple and portable instrumentation assume great importance and appeal in both clinical and research contexts. [1]

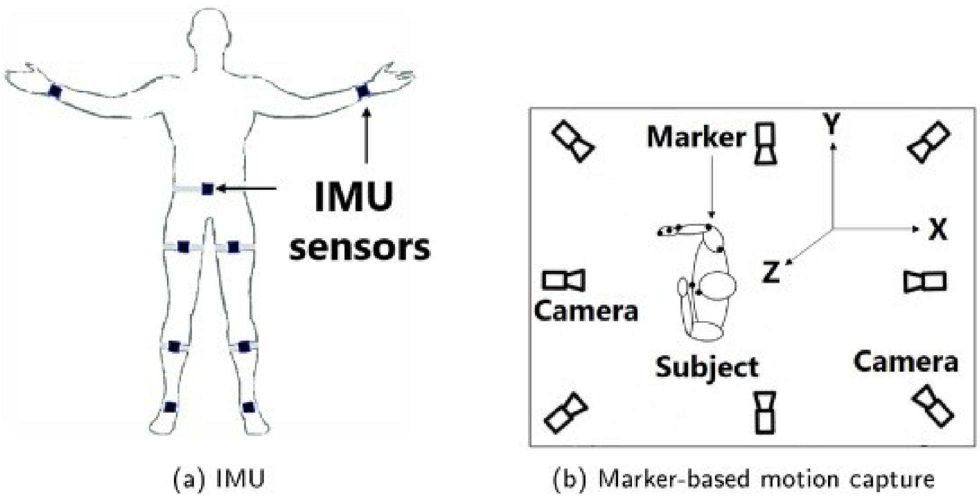


Figure 3 In this figure, the main differences between an analysis with IMU and an analysis with a motion capture system are shown: the presence of markers and the acquirable data.

The detection of movement and identification of daily life activities through inertial measurement units (IMUs) are useful for evaluating the quality of life and highlighting mobility issues, for example, in patients with Parkinson's disease or stroke.[4] Numerous studies show how accelerometers have been used to recognize activities.[4] Another important subject of study is sensor placement: chest, wrist, waist, thigh, calf, and according to Mathie et al., the most comfortable position to wear an IMU is the waist. [4]

Wearable sensors promise to acquire functional parameters identical or similar to those measured with more advanced laboratory techniques at a lower and more accessible cost.[2]

In the next paragraph, we will present in detail the sensors used in this thesis, namely Inertial Measurement Units (IMU) sensors, with a discussion on the technologies used to transform input data, i.e., the patient's movement, into usable output data, such as the acceleration of the movement.

1.2 IMU Sensor Technology

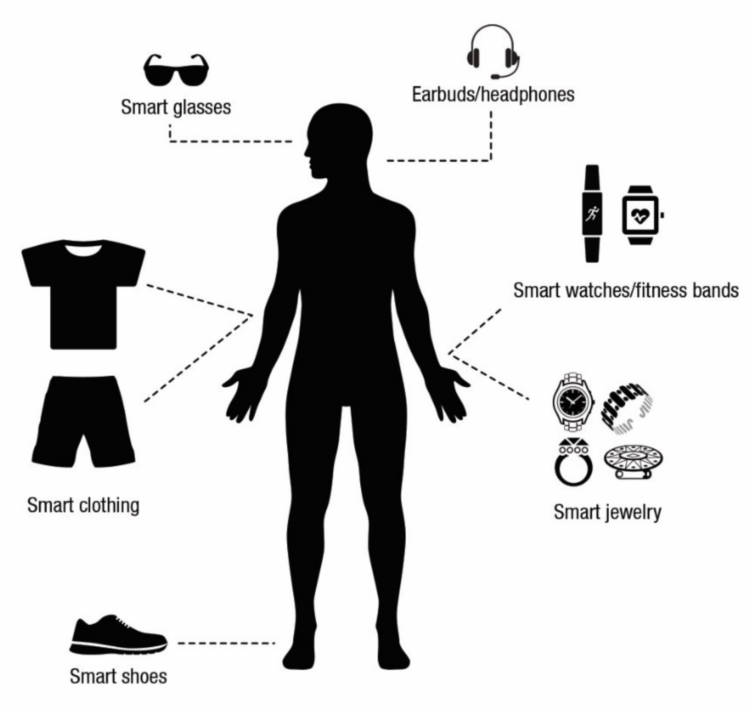


Figure 4 This figure shows how wearable devices are becoming increasingly common and are taking on more common forms for better biocompatibility and form factor.

Wearable sensors for measuring physiological parameters or signals are composed of a sensor that measures the physical quantity of interest and an embedded processing unit.

By sensor, we mean "a device that responds to physical inputs of interest with a functionally related output, typically electrical or optical" or "a sensor generally refers to a device that converts physical measurements into signals that are read by an observer or an instrument."[6]

The classification of sensors is based on their interactions with the measurement:

- Contact sensors: sensors that require physical contact and are further classified by the degree of invasiveness: invasive, minimally invasive, or non-invasive. Better is the contact, less artifacts will be introduced in the measurement.
- Non-contact sensors: sensors that do not require contact with the quantity of interest. They introduce minimal noise and are usually implemented in the collection of environmental data such as infrared thermometers.
- Sample-based sensors: this approach is invasive and collects samples that are analyzed using laboratory-based sensors or instrumentation. They are commonly used in healthcare, such as monitoring blood glucose concentration.[6]

Wearable sensors are a type of contact sensor, generally non-invasive or minimally-invasive, that can be worn by an individual as visible in *Figure 4*, and this means that they must have a comfortable form factor. Wearable sensors can be divided by their sensing technique such as electrical, optical or chemical sensor, but in this thesis only mechanical sensors will be used. Mechanical sensors are device that detect mechanical deformation of a material in response to an input and translate it into an electrical, optical, magnetic or thermal signal. Most used are electromechanical sensors that are divided by:

- Piezoresistive sensors, device subject by piezoresistive effect, that's when a conductive materials are subjected to mechanical deformation.
- Capacitive sensors, that are based on variations of capacitance of a material in response to mechanical stimuli;
- Iontronic sensors. It's a capacitive sensor that use ionic fluid as dielectric that have electrical capacitance thousand times larger compared to traditional parallel plate;
- Piezoelectric sensors are based on the piezoelectric effect of the materials that generate electrical chargers under external mechanical force, pressure or strain.
- MEMS or Micro Electro-Mechanical System are miniaturized chip that including mechanical and electrical structures with sensibility that can rang from millimeter to less than 1 micrometer.

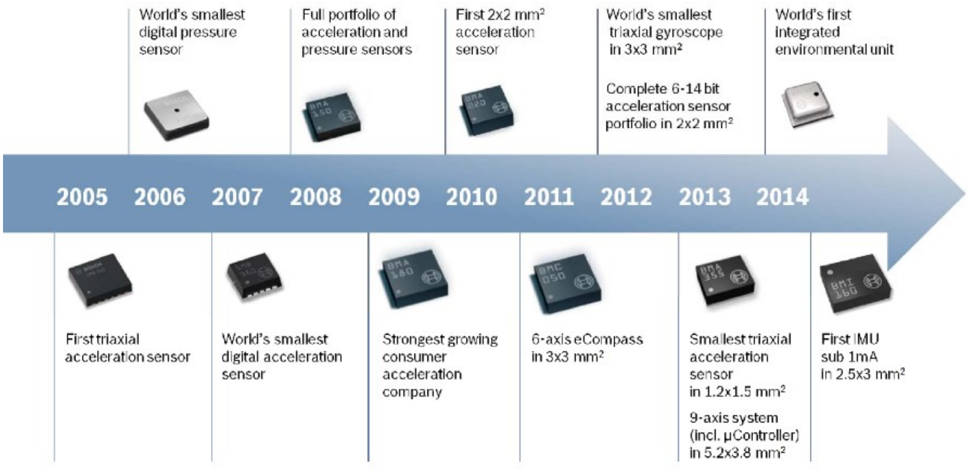


Figure 5 This figure represents MEMS first appearance 2005-2014

Most MEMS used as inertial sensor are:

- Accelerometers: measure proper acceleration, how much a mass is pressed on an elastic element by an external force. Conceptually, is a box with a damped mass on a spring and when the accelerometer experiences an acceleration, the mass is moved compare to the resting position.

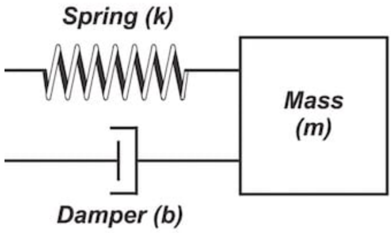


Figure 6 Mass-spring-damper system used in a capacitive MEMS accelerometer.

- Gyroscope: measure the angular rate of rotation around one or more axes. MEMS gyroscope exploit the Coriolis effect or when a mass m is moving in linear direction with velocity v and undergoes angular rotation with velocity Ω the mass experiences an apparent force F in a direction perpendicular to the other two size. Gyroscope include commonly two masses that are stimulated by an applied oscillating electric field which causes the to oscillate and move constantly in opposite directions.

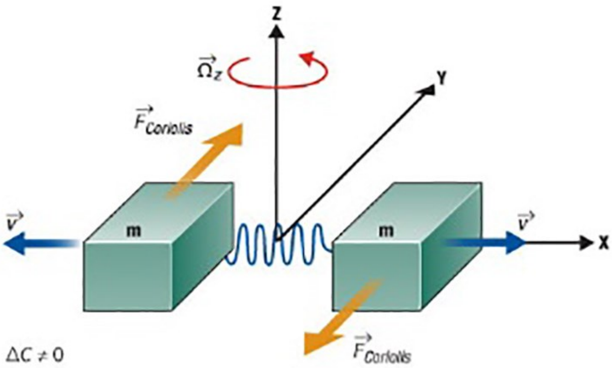


Figure 7 MEMS gyroscope

- Inertial Measurement Unit (IMU): is an electronic device that measures and reports a body's specific acceleration and angular rate using a combination of accelerometers and gyroscopes. We have 3-6-9 axis, the last ones are IMUs that includes magnetometer that measure strength and direction of a magnetic field; [6]

The combination of medical devices, the use of algorithms for data processing, and, after COVID-19, the convenience of telecommunication, have led to the exponential growth of the field of telemedicine and digital health.

1.3 Digital health & medical technologies for rehabilitation

Figure 8 In this figure, it is shown how the field of digital health is expanding to 360 degrees, allowing patients and doctors to speed up communication timelines. Not only that, but through applications, patients themselves can better understand their own health status.

Digital health encompasses various categories, such as mobile health (mHealth), health information technology (IT), wearable devices, telehealth and telemedicine, and personalized medicine.[A]

Digital health technologies employ computing platforms, connectivity, software, and sensors for healthcare and related purposes. They span a broad range of uses, from general wellness applications to those serving as medical devices. These technologies include those intended for use as a medical product, within a medical product, as companion diagnostics, or in conjunction with other medical products such as devices, drugs, and biologics and they may also be employed to develop or study medical products.[A]

Digital health tools provide a more comprehensive view of patient health by offering data access to providers and control over their health to patients. These technologies create real opportunities for enhancing medical outcomes and efficiency. Consumers can use these tools to make informed health decisions and improve prevention, early diagnosis, and chronic disease management outside of traditional health care settings.[A]

Figure 9 In this figure, it is shown how a well-structured single application can allow us to do multiple things, such as video calling the doctor or calling an ambulance, downloading the data from the latest tests done, or booking an appointment.

These technologies are used to reduce inefficiencies, improve access, lower costs, increase quality, and personalize medicine for patients. Patients and consumers can use technologies such as smartphones, social networks, and internet applications to manage and track health and wellness.[A] These advancements are creating innovative ways to monitor health and provide greater access to information, leading to a convergence of people, information, technology, and connectivity to improve health care and health outcomes.[A]

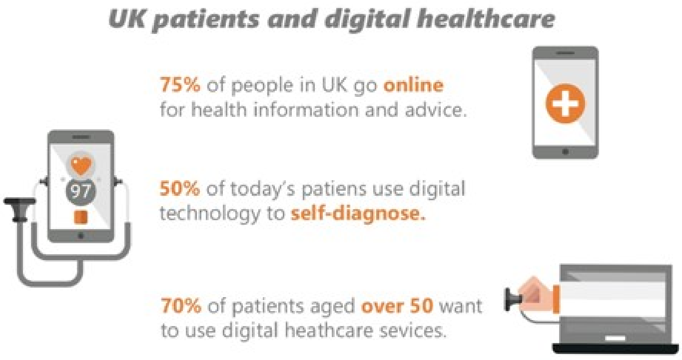


Figure 10 This figure show the growing prevalence of healthcare mobile Apps in the UK, that highlight the importance of implementing more application in digital health

From mobile medical apps and software that support clinical decisions to artificial intelligence and machine learning, digital technology has sparked a revolution in healthcare. [A] The potential of digital health tools is immense, including the ability to enhance disease diagnosis and treatment accuracy, and improve healthcare delivery for individuals.[A]

Algorithms are being used in various areas of healthcare, including public health, healthcare provision processes, medical research, prevention, prediction/risk profiling, diagnostics, therapeutic procedures, prognosis, rehabilitation, and nursing care.[7] Current research fields related to the use of algorithms in healthcare include diagnostic procedures, prediction and risk profiling and therapeutic procedures. Algorithm-based products already on the market include diagnostic software and mobile apps that use algorithms to interpret medical data.[7]

Mobile health apps have been suggested for a variety of healthcare settings, such as disease management and monitoring, diagnosis and triage, and medication reminders.[8] However, there is limited evidence for their effectiveness, and medical app governance is challenging. In response, the UK National Institute for Health and Care Excellence (NICE) developed an evidence standards framework for evaluating digital health technologies (DHTs), which categorizes digital technologies into tiers according to their intended function and provides recommendations for the associated standard of evidence required for each tier. [8]

Digital health interventions, such as telemedicine, mobile apps, and monitoring sensors, offer potential benefits in supporting independent living, self-management, and reducing healthcare costs.[9] Machine learning algorithms have the potential to revolutionize healthcare delivery. While machine learning has been applied in various healthcare domains, there is limited empirical evidence of its effectiveness in clinical practice.[9]

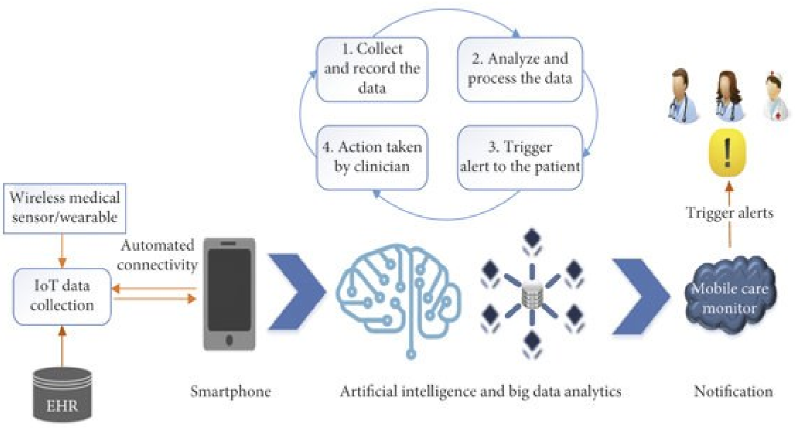


Figure 11 Example of how AI works on mHealth with the use of an application that permit patients to knows their health status, for example a glucose monitoring application with the relative sensor permit to a diabetic patient to be more controlled thanks to a prediction algorithms.

Large datasets and accurate labels provided by expert clinicians are necessary for the development of computerized decision support tools, which can empower patients to monitor their health status and assist health professionals in making decisions.[9] The effectiveness of machine learning applications in digital health interventions needs to be tested in real-life studies to achieve their full potential. [9]

Ekeland et al conducts a study where they describe the effectiveness of telemedicine.[10]This article highlights how the use of telemedicine has multiple benefits, such as efficacy, including therapeutic effects, improved efficiency in health services, and technical usability.[10] From an economic perspective, several reviews have suggested that telemedicine appears to be cost-effective, although few have drawn firm conclusions.[10] One review found that 91% of the studies showed tele-homecare to be cost-effective, as it reduced hospital usage and improved patient compliance, satisfaction, and quality of life.[10] This study also shows the limitations of telemedicine, given that it is still a relatively new field and subject to constant new publications on the topic. [10]

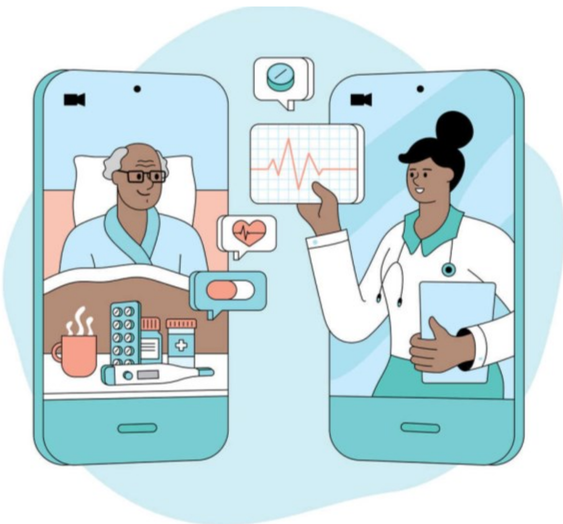


Figure 12 This figure shows us how an elderly patient, probably sick, still has the possibility to interface with their own doctor through platforms that allow it using a simple smart device such as their own smartphone.

Nowadays, the use of wearable devices combined with applications downloaded on our smartphones is becoming increasingly common. We move from using smartwatches during the workday to using sport-watches or sport-bands for analyzing and monitoring heart rates and calorie burn during exercise. This combination is also widely used in diabetic patients, where the sensor communicate with the application allow for constant monitoring of blood glucose levels.

The creation of rehabilitation applications allows us to break down important barriers such as the possibility for an elderly or immobile patient to have access to medical consultation. These types of applications, easily downloadable and user-friendly, are becoming increasingly popular in the community because they are easy to use and also very powerful. These applications not only concern patients, but also allow doctors to have a better perspective on the patient's clinical life through centralized systems that summarize and highlight the necessary features for the doctor to make a better diagnosis and proceed with the best possible treatments.

As we will explain in the next paragraph, our thesis will focus on the creation of a rehabilitation application for doctors to analyze and save patient data during the execution of the Sit-to-Stand task.

1.4 Aim of the thesis

Currently, the protocol used for data acquisition in the rehabilitation task of Sit-to-Stand is carried out using a stopwatch. This protocol is extremely poor because, in addition to the data acquisition time, no information related to movement is provided, such as the inclination of the torso, accelerations, or other important details for a detailed analysis of the patient. Moreover, the doctor will still have to save these data somewhere.

The aim of this thesis is to create an Android/iOS application that can be used on a smartphone to process data offline and collect it using an IMU sensor connected via BLE (Bluetooth Low Energy) protocol, with the goal of automating the current data collection and generating a report on the execution of the Sit to Stand task.

In the following chapters, the materials and methods used will be described, formally outlining the execution of the task of interest and providing a brief introduction to the platform used for creating the application. Subsequently, the created application will be presented in detail and finally, the results obtained will be discussed, including the population, protocol used, and subjects analyzed. The last chapter will report the conclusions drawn during the thesis and future developments.

In the upcoming chapter, I will discuss about the sensor used in the thesis and the manufacturer that build it."

2. MUSE: an open source IMU platform by 221e©

In this chapter, we will talk about the 221e© company and the sensor they built called Muse, which is used in the application to collect data for the task.

2.1 About 221e Company

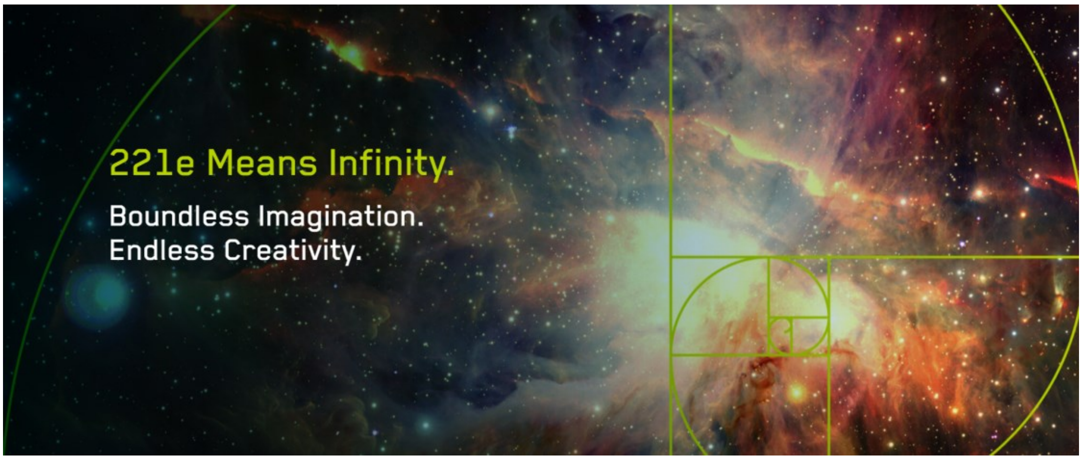


Figure 13 221e official site presentation

"*221e means infinity, because endless is the boundary of imagination and so of innovation*"

With this motto 221e describe itself. This company was founded in 2012 by Marco Signorelli: from his passion about motorsports security and his desire to start his own business, create revolutionary sensors with high sensibility and their quality is the hallmark of 221e. One of their first customers were indeed people in the motorsports industry.
Their technologies were immediately utilized in various applications thanks to the presence of real-time data collection and sensor fusion algorithms and their mission is the research, development and production of electronic systems embedded with proprietary algorithms to collect, elaborate and return clear data ready to be used.[B]
221e believes that wearable devices can provide numerous benefits in a variety of industries, including healthcare, sports, lifestyle, safety, robotics, industrial automation, and gaming.
As tomorrow's challenges will require greater data accuracy, extreme miniaturization, and low energy consumption, the company's plug-and-play products are designed to monitor

characteristics such as force, pressure, direction, speed, temperature, as well as biopotentials like respiration and heart rate, to obtain precise information and ensure the highest standards. 221e's solutions aim to simplify the work of data scientists, researchers, manufacturers, engineers, and pioneers worldwide who are leading the way in developing new wearable sensors. [B]

221e developed proprietary algorithms that serve as the backbone of embedded real-time applications. These algorithms process sensor data online to detect specific conditions and enable the execution of appropriate actions like an intelligent systems. Their hardware-software mix allows us to optimize data management through the use of distributed AI within nodes, edges and clouds, based on a fully modular architecture.[B]

Figure 14 NeuraSense AI presentation taken from the 221e official sites

The company collaborates with innovative brands on a unique journey in history, where opportunities for completely new technologies have arisen. Their working environment is capable of meeting the most demanding project requirements, providing full-service solutions that rely on reliable processes and methods to ensure quality, performance and timely delivery to market.[B]

Figure 15 Some partners of 221e that are presented in the sites

2.2 About MUSE

In this thesis and in the following chapters, we will describe the use of a sensor for data acquisition. In particular, the device used is a sensor created by the company just described. The product we used is called Muse™, a miniaturized multi-sensor imu that permit us to acquire motion and environmental data in real time.

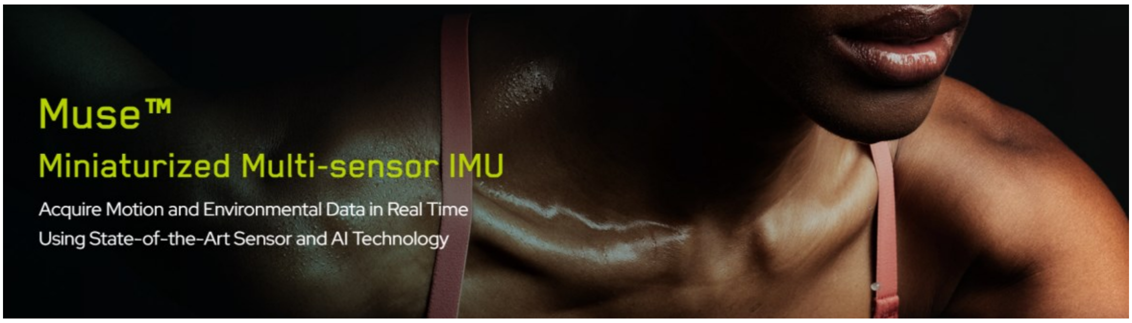


Figure 16 Official page MUSE presentation

MUSE is a wireless multi-sensor logger that is low power, miniaturized, and easy-to-use and incorporates state-of-the-art sensing technology into a compact, robust, and customizable solution.

With inertial and environmental sensors, on-board flash storage, wireless connectivity, automated on/off functions, and regulated rechargeable power, MUSE provides a versatile system for data acquisition. It is fully certified, safe, and suitable for both wearable and connected industrial use cases, in fact has been used in a variety of wearable and IoT devices by global enterprises, innovative startups, and world-leading researchers.

This sensor can measure acceleration, angular rates, 3D rotation, magnetic fields, temperature, humidity, proximity, ambient light, ambient pressure, and signals intensity and, additionally, the platform can run 221e's proprietary algorithms and embedded AI software libraries, which take intelligent precision sensing to higher levels.[C]

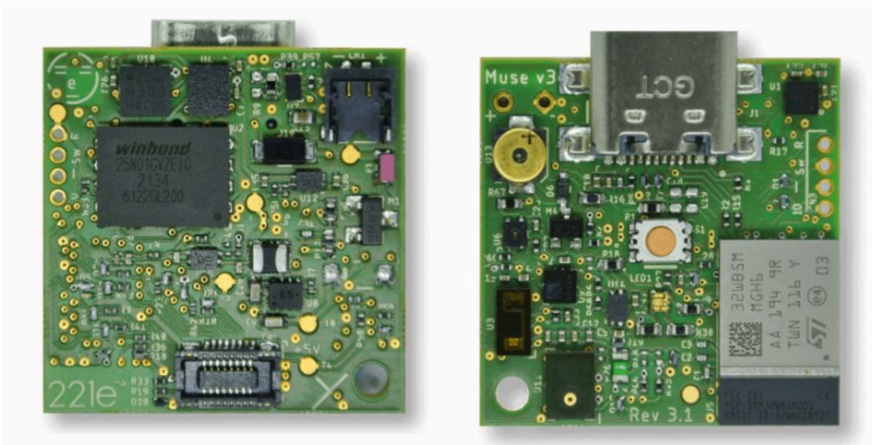


Figure 17 this figure show the architecture of the MUSE sensor

By utilizing its onboard flash storage, BLE, embedded processors, and array of sensors, Muse provides a versatile platform for implementing various data acquisition and processing tasks. Whether we want to transform ours product into an innovative smart device or use it as a standalone IMU, Muse offers the flexibility and capability to suit our needs.[C]
Our application will use MUSE for the kinematic analysis of a rehabilitation task and, to do this, 221e provided us the hardware description of the IMU sensors we will use:

- Accelerometer detect changes in velocity over time, which is also referred to as inertial acceleration. They detect linear acceleration of the object they are attached to in the form of movement, shock, or vibration, and their readings are usually measured in units of g, where one g is approximately 9.81 m/s.

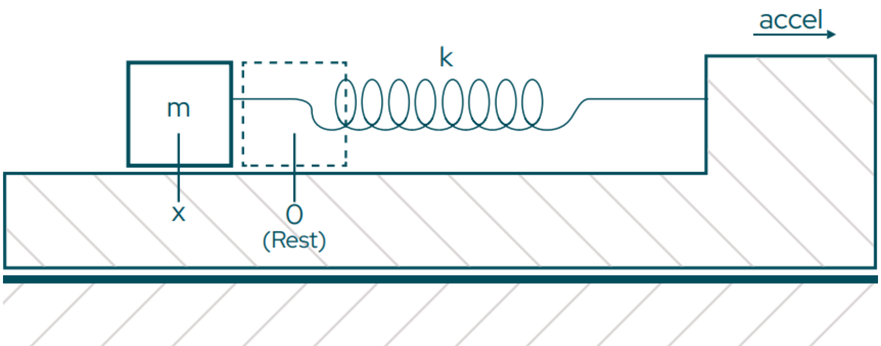


Figure 18 F=kx=ma, a mass-spring system representation

There are various types of accelerometers, including mechanical, quartz, and MEMS, but they all operate on the same principle of a spring-mass system. In this system, the mass is capable of moving along a fixed direction, which is known as the sensitivity axis. When the mass (m) experiences linear acceleration along the sensitivity axis, it moves from its resting position. According to Hooke's law of elasticity, the displacement (x) is proportional to the pulling force (F). The spring constant (k) indicates the relationship between the displacement (x) and the force (F), and ultimately, we get F=kx. Newton's second law of motion, F=ma, is also relevant in this context.[J]

- Gyroscope uses the Coriolis effect to detect the angular velocity of an object in units of degrees per second (dps) or revolutions per second (rps). It comprises a rotor that is mounted on a spinning axis within a larger wheel, known as a pivot that enables the rotor to rotate on a specific axis, which is referred to as a gimbal. As the rotor spins, the gyroscope maintains a constant direction.[J]

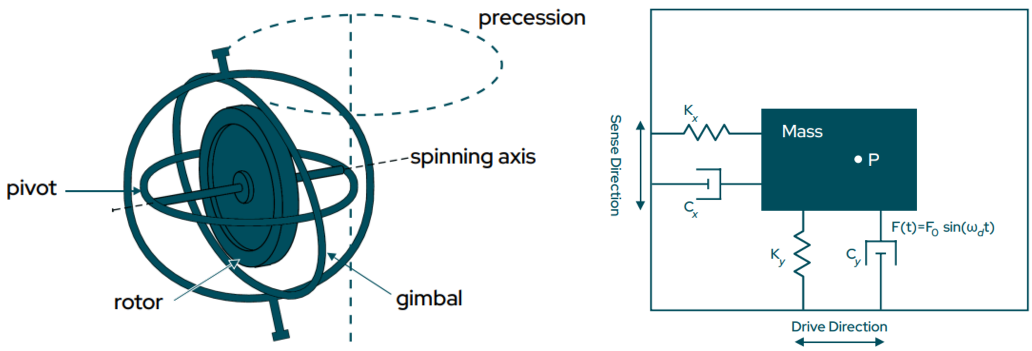


Figure 19 Gyroscope: mechanical(left), vibratory (right), how it works

There are various types of gyroscopes available in the sensor market, including mechanical, fiber-optic, ring laser, quartz, and MEMS gyroscopes, each with different levels of performance. MEMS gyroscopes use a vibrating electro-mechanical element instead of the rotating parts used in other types, allowing for greater miniaturization. These gyroscopes operate based on the energy transfer between two orthogonal vibration modes, the drive mode and the sense mode. The dynamics of the Coriolis force, which is proportional to the angular velocity, couple the two modes.
A MEMS gyroscope can be represented as a harmonic oscillator with two degrees of freedom, namely the drive mode and the sense mode. The displacement of the sense mode is a direct indicator of the applied angular rate. To enhance the gyroscope's bandwidth and dynamic range, it often functions as a closed-loop system.[J]

In the next chapter, the materials and methods used in the thesis will be described, with a focus on the rehabilitative task that will be carried out in the study, as well as a description of the processing of the raw signal collected and transformed into the final clean signal.

3.Materials and Methods

In this chapter, the rehabilitation task of interest, Sit-to-Stand, will be formally described. Various studies performed on the same task and related methods will be briefly reported, up to the description of the acquisition process that will be carried out for the thesis. Subsequently, the steps taken and the choices made for cleaning the raw signal collected from the sensor described in the previous chapter, will be explained. However, first we will introduce the platform used to create the application.

3.1 About Flutter Dart: the software framework used to build an app

As written by Antonio Tedeschi in 2019, developers have been compelled to extensively scrutinize and comprehend the mechanisms and principles of major operating systems, such as iOS and Android, in order to cater to a wider range of users through various markets, due to the continuous evolution of mobile devices.
Despite the advancements made in these OS, including the implementation of new languages and techniques that have facilitated and accelerated the development of native applications, creating and sustaining a native app for multiple platforms over an extended period of time remains challenging for individual developers. This is mainly due to the lack of a common codebase among applications for different mobile platforms.
Although significant simplifications have been implemented, these frameworks still fall short in fully utilizing the distinctive traits of various mobile platforms, and are hindered by slower performance and limited access to local resources.
In 2018, a new framework emerged in the cross-platform app development arena, developed by Google: Flutter, which is the topic of discussion of this paragraph. [F]

Figure 20 Dart and Flutter Logos

With a beta version released in January 2018, in less than a year, Flutter reached version 1.0 (which denotes its first stable version).

Flutter is a completely free and open-source project that enables the creation of high-quality native apps for iOS and Android quickly, with support for native interfaces and it is utilized by numerous developers and large enterprises, including Alibaba, Google itself, and many others. The objective of Flutter is to create new apps through a fast development phase with features like hot reload, which does not require recompiling the code; Expressive and flexible user interfaces with a set of composable widgets, animation libraries, and a layered and extensible architecture; Performance that is very close to native and a single codebase for Android and iOS applications. [F]

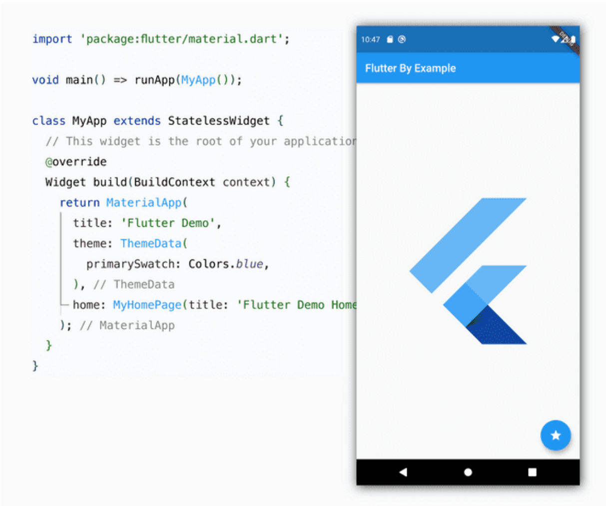


Figure 21 Example of Flutter and Dart code and visualization on a smartphone

To achieve this, Flutter comprises two macro layers: a layer written in C/C++ and a layer written in Dart, a new and modern object-oriented language that defines most of its system (gestures, animations, framework, widgets, etc.) and offers developers extensive control over the system itself.

Furthermore, Flutter differs from most hybrid app development frameworks in that it does not use either WebView or the OEM widgets (original equipment manufacturer) available on the device. In fact, when a native app interacts with the platform to create a widget, the widget is selected from the available OEM widgets, displayed within a canvas, and events are passed to the widgets, which will, in turn, communicate with the native application.[F]

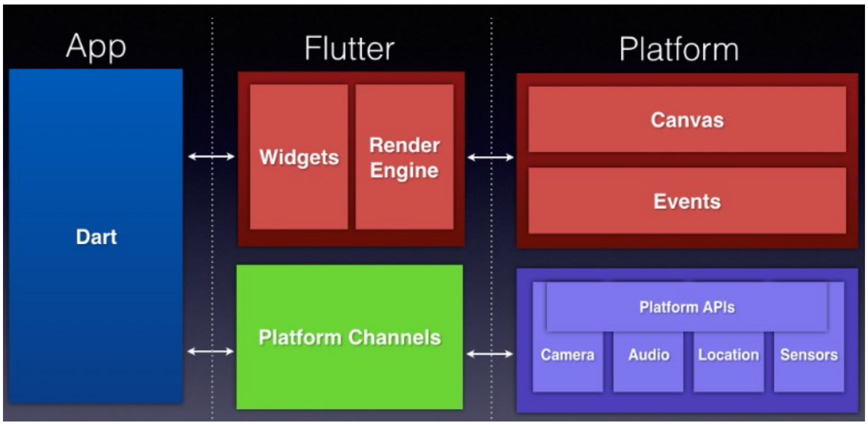


Figure 22 This figure show how an application communicate with flutter and the platform.

As shown in the figure above, and remembering what was said a little while ago, flutter use Dart: in its ecosystem, packages are utilized to manage shared software, including libraries and tools. To acquire Dart packages, the pub package manager is used, for example it's possible to create ours own package and upload on pub.dev sites and import it on ours application.

Figure 23 Pub.dev official sites logo

In the Flutter framework, the central class hierarchy is made up of Widgets. A widget represents an immutable description of a section of a user interface and can be transformed into elements, which manage the underlying render tree. Widgets don't have any mutable state; all their fields must be final. [G]

A widget can be included in the tree zero or more times, including multiple times and when is placed in the tree, it is inflated into an element: a widget that is incorporated into the tree multiple times will be inflated multiple times. The key property governs how one widget replaces another widget in the tree[G].

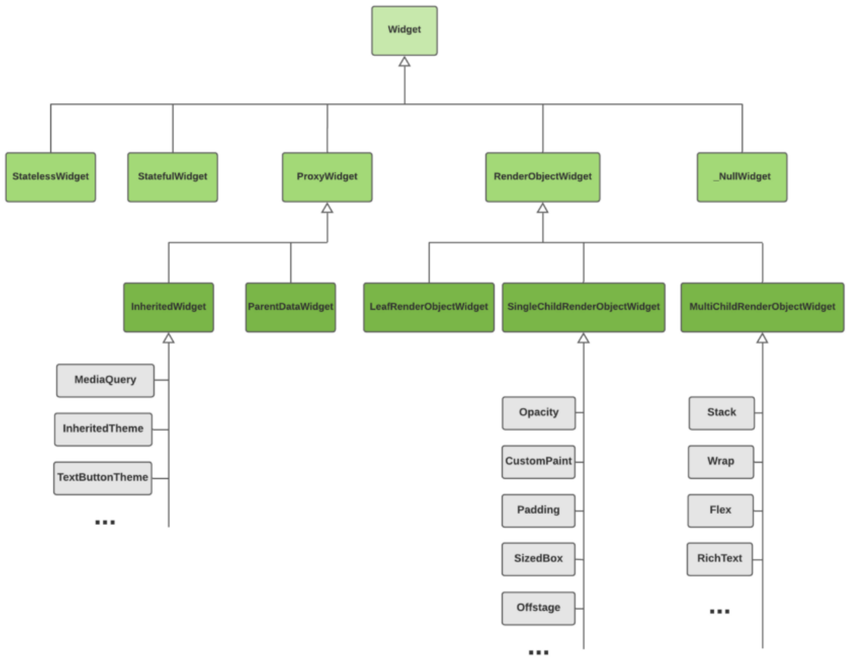


Figure 24 This figure show the different types of widget and their relative subcategories

As highlighted above there is different type of widget that can be used in an application. For example if we wish to associate mutable state with a widget, consider using a StatefulWidget, which creates a State object or StatelessWidget for widgets that always build the same way given a particular configuration and ambient state.[G]. There is also a difference between visible and invisible widget like a button or a column, the first is visible to the user but the second is an invisible widget.

The combination of Flutter, the Dart language, the use of packages and widgets will give rise to the application that will be explained step by step in chapter 4.

3.2 Formal description of the task

All studies conducted on Sit-to-Stand present differences in terms of task execution and sensor placement for better data collection.

In accordance with the literature, every studies require different placements of one or more imu sensors on different parts of the body. For instance, Cinzia amici et al. [1] placed the sensor on the proximal edge of the manubrium sterni or similar Najafi et al. placed on the chest.[11] Justine Hellec et al.[13] placed the sensor on glasses and similarly M. Aloqah et al [12] in their classification of human postures placed the sensor on a headband.
Uriel Martinez-Hernandez et al [14] placed the sensor on the thigh for probabilistic identification of the task and D. Rodríguez-Martína placed a single inertial sensor on the waist[4].
There is no better position than others a priori, but the position is chosen based on the nature of the movement to be analyzed, the type of data collected, and the research question to which we want to give an answer.
As reported below, according to Grimm et al.[2], for the sensor that we want to use and the task we want to analyze, we have the following optimal options:

Functional test	***Sensor type***	***Sensor Location***	***Motion Parameters***	***Clinical Application***
Sit to Stand	Inertial sensor	Sternum	Angular velocity	Frailty
		Lower back, thigh and tibia	Acceleration	Fall detection
Sit to Stand	Inertial sensor	Trunk: S1-L4	Phase detection	Frailty
			Trunk range of motion	Osteoarthritis
			Vertical Acceleration	Parkinson
Sit to Stand	Inertial sensor	Sewn into trouser: waist, thigh, knee	Temporal event detection	Osteoarthritis

Table 1 Overview of available methods to assess physical function of the lower extremity with wearable motion sensors [2]

In Sit to Stand, the key points to analyze are:

1. Starting position;
2. Patient's positioning;
3. Sensor positioning.

The subject's starting position varies depending on the task. For instance the position of the hands is thoroughly studied in [1], where Amici et al. suggest that the hands should be kept on the thighs or leg position study was performed in parallel with the study of the chair's height and the presence of armrests [15].
In this particular review, Wim et al. describe precisely all chair-related determinants, for example they wrote that lowering the height of a chair during a sit-to-stand movement can increase the difficulty of the movement or even result in failure: A lower seat height leads to increased angular velocity of the hip and more foot repositioning, which is also known as the "stabilization strategy" and can also increase trunk, knee, and ankle angular displacement, affecting the maximum moment needed at the hip and knee. Seat height changes can affect biomechanical demands, such as the need to move the body's center of mass over a larger distance or alter the strategy.[15]
Wim et al also report the utility of the Armrest: its use can impact the moments exerted and positioning of hands. They wrote that using armrests results in lower moments at the knee and hip, with a reduction of about 50% of the extension moment needed for the Sit-to-Stand movement.[15]
In this thesis, the chair used in the task will have the following characteristics:

a) The height will be adjustable to comply with the studies by Wim et al;
b) There will be no armrests in contrast to the effective aid described by Wim et al.
c) The chair will not have a backrest in accordance with Wheeler et al, who suggested a negative influence of seat posterior slant because of tilting the body's center of mass farther backward.

Figure 25 Chair used for the Sit-to-Stand experiment that follow the characteristics written above

The second fundamental point investigated in this thesis is the patient's positioning: usually, the subject performs multiple tasks, as shown in [16] where Mazzà et al perform the Sit-to-Stand task with two different velocity and analyzed it. In according with the literature, our patients will start from a position similar to that maintained in a chair with a backrest, and their arms will be crossed over their chest without influencing the movement or being used to perform the task, as in the study conducted by Hellec et al.[13]

The Sit-to-Stand movement will be divided into 3 steps, where initially there will be a resting period during which the application will already start collecting data, followed, when the patient prefers, by the change of position from sitting to standing. The movement will allow the patient a minimum forward torso twist. Finally, the patient will have to maintain the upright position as still as possible for a couple of seconds.

Last fundamental point investigated was the sensor positioning. As we see before, Grimm et al [2] already give us three different possibility.

In this thesis we want to analyze the Sit-to-Stand task using principally the Accelerometer data and in according to the subject positioning hypotheses described above, the most recommended choice is L4-S1. This positioning allows us to have data collection less affected by angular accelerations caused by torso torsion, which would occur if the sensor were placed, for example, at chest level. Therefore, we can say that most of the information regarding the movement will be described by the data collected with respect to a single selected axis. The analysis of the movement will then be a uniaxial type of analysis.

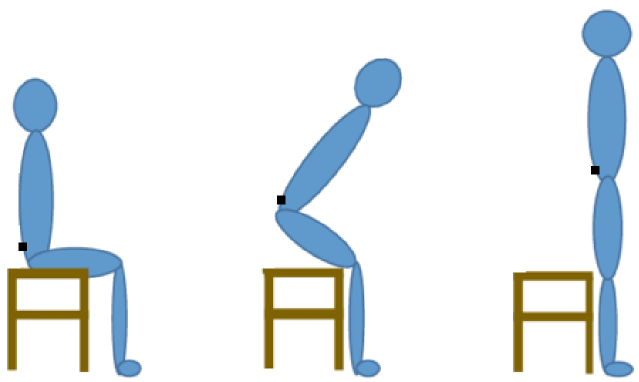


Figure 26 In this image, the sensor L4-S1 placement during the Sit-to-Stand task is shown, highlighting how the movement can be summarized in a uniaxial acceleration analysis.

Once the data collection of the task is carried out, we will have a signal that describes the patient's acceleration throughout the movement.

3.2 Processing algorithm pipeline

Once the application stops its collection, offline algorithm starts the processing of the raw signal that appears as a signal affected by noise, therefore, in order to make the signal more significant, it is necessary to perform some processing steps.

Figure 27 This image was taken from our application and shows a raw signal trend with outliners collected from the sensor.

Initially, a pre-filtering of the signal is performed in order to remove some outliers that are formed by the sensor.
Taking *Figure 27* as an example of a acceleration vs time chart, this figure shows a uniaxial data collection in which the sensor is simply placed on a flat surface so all the gravity component falls on the reference axis (in this case, the X axis). By observing the y-axis values of this graph, we immediately notice the presence of gravity, and we also note, as said before, the presence of some outliers that are formed only when the sensor is affected only by the force of gravity, in fact, successive movement tests, that we will see after in this paragraph, have shown that during the movement phase, these spikes are not present.
Every accelerometer on the market, considering also the description given in the previous chapter, is affected by the gravity component and this is the main reason why the signal requires pre-filtering and a processing pipeline.
To eliminate the gravity vector, we applied simple trigonometry: starting from the initial assumption that the reference axis of the movement will be the X axis and a small component

of this force will rotate on the Z axis during the movement, we will find ourselves in the following condition.

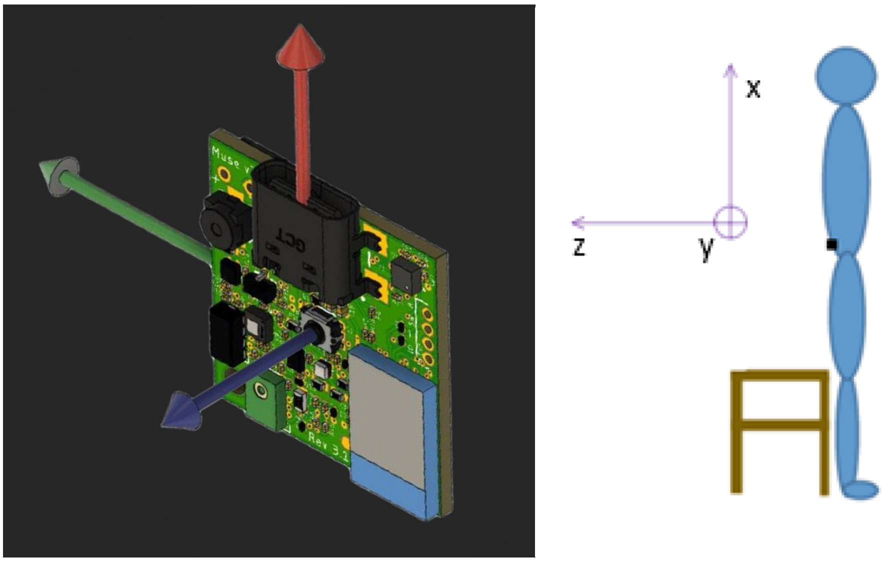


Figure 28 MUSE Axis representation(a) and how is placed on subject(b)

As we can see in the figure above, the main information about the movement will be contained on X axis, so, the algorithm to eliminate the gravitational component from the reference axis will follow these steps:

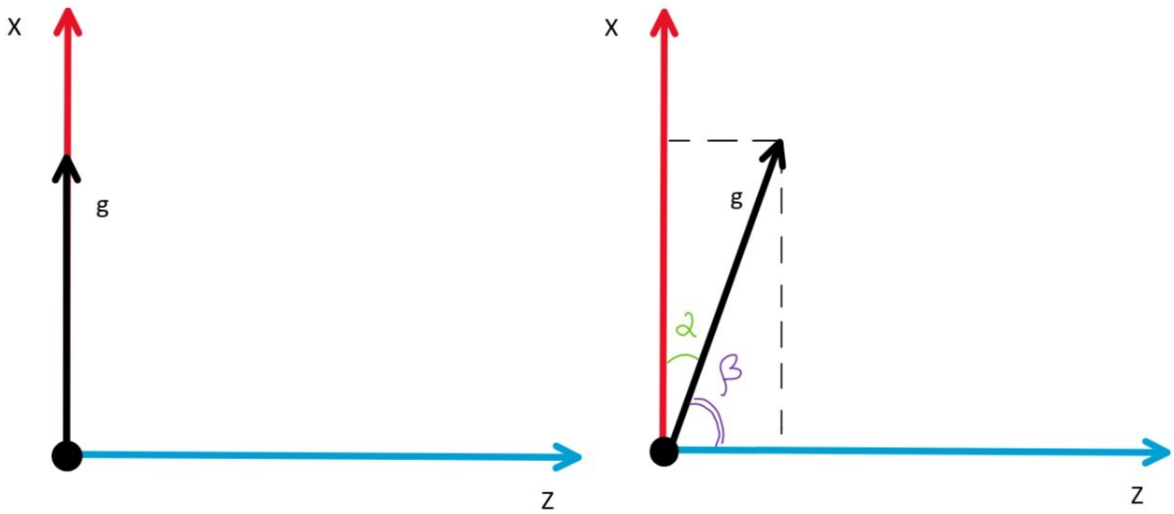


Figure 29 Representation of the gravity component in a 2D graph in which (a) shows how positioning the sensor with the x-axis as previously mentioned contains the entire component. In figure (b), the angles present during the movement are shown, which will be used to calculate the gravity components in the axes.

In order to perform a pre-processing of this signal, a calibration is therefore carried out in order to obtain the actual value of the force of gravity that the sensor perceives. Let's consider the figures above, in the left image we can see that if the sensor is positioned with the USB port

facing upward like *Figure 29*, we can say that the X-axis will be the only axis affected by the force of gravity (g component will be taken positive as shown in the *Figure 29*). Therefore, during the calibration function, data is collected by placing MUSE on a flat surface following the aforementioned indications. In the right image, we had to image that the sensor is placed on the subject, so the force of gravity will follow the physiological curve of the lumbar region where the sensor is placed. Consequently, the hypothetical starting position of the sensor is graphically represented as the figure above. After the calibration where the true value of gravity has been found, we start collecting the value about X and Z acceleration.
Thinking about the movement, the α angle will be always smaller than β for all the Sit-to-Stand movement, so for calculating the α angle we applied the Pythagoras theorem for right triangle as follow:

$$\alpha_i = \arctan\left(\frac{g_{iz}}{g_{ix}}\right)$$

With α_i the α angle at time *i* and g_{iz}, g_{ix} are the component sensed by MUSE for *i-th* x and *i-th* z acceleration. Once found the angle we can eliminate the component from the X axis with a simple assumption: when the movement start the acceleration that MUSE sense is a component given by the gravity plus the true acceleration, so for calculating the true value of acceleration we perform a simple difference as follow:

$$true\ a\_x_i = \frac{a\_x_i\ sensed}{cos(\alpha_i)} - g$$

Now our signal related to the X-axis is no longer affected by the gravity component, so we can work on a signal with greater physical significance. Next step will be processing this new signal with filtering and smoothing technique.
Essentially, filtering is a neighborhood function, which is why this operator is called a local operator. The filter acts on a window of chosen length (in our case will be 5), usually much smaller than the total duration of the signal.
There are various types of filters applicable to our signal:

- Low-pass filter attenuates signals with values higher than a certain cut-off value, which is defined based on the capacitance values.
- High-pass filter attenuates signals with values lower than a certain cut-off value. It’s also define the opposite of Low-pass filter.
- Band-pass filter is a combination of a high-pass filter and a low-pass filter.

- Band-stop filter, also known as notch filter, is a combination of filters that strongly attenuate the signals that are within a narrow band, while leaving the signals outside of that band unchanged. [D]

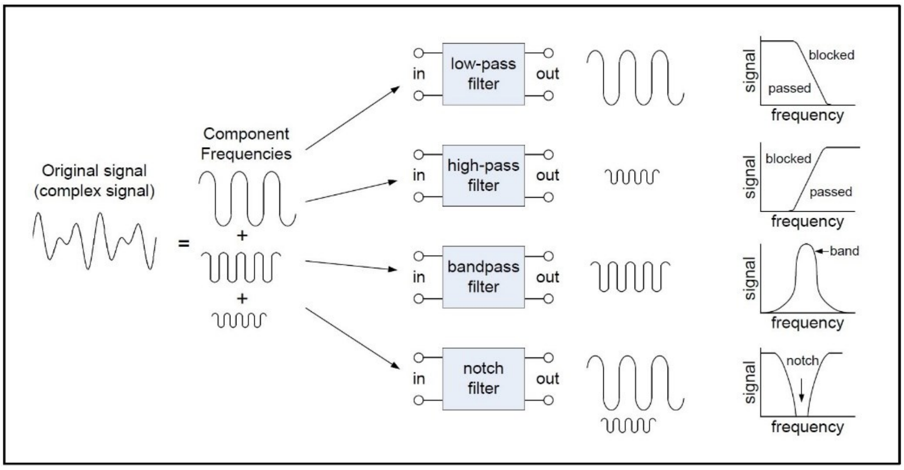


Figure 30 This figure represent how the different filtering technique can transform an original signal into its components

By observing the various types of filters in the figure above, the one that best suits our needs is a low-pass filter.

As can be seen in the previous figures, the gravitational component has been removed, but the outliers are still present. That's why the next step is to eliminate these outliers. For doing that we only replace the outliners with the value of gravity that we calculate before during the calibration, like a low pass filter. The presence of outliers occurs only in the case of a stationary sensor affected only by the gravitational component and this does not imply a loss of significant data for the purpose of analyzing the task.

Once the signal has been cleaned from outliers, our raw signal will present a trend like this:

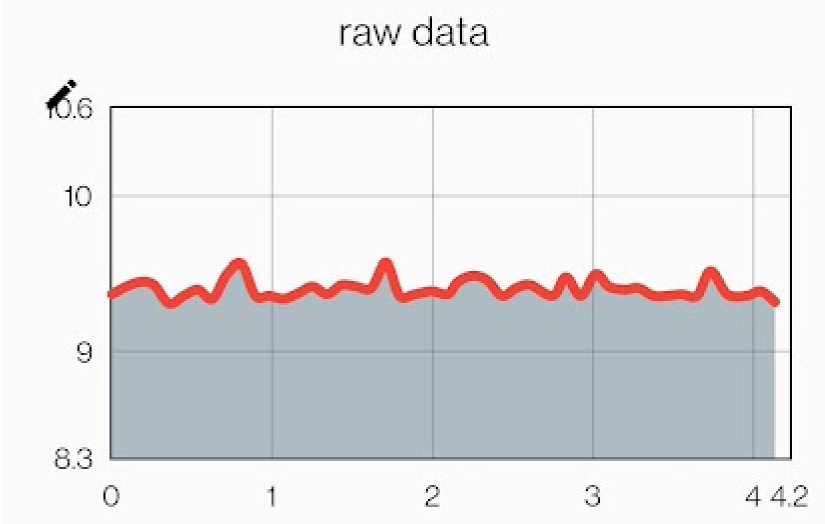


Figure 31 This figure represent the same data of Figure 29 but with the filter function all the outliners are eliminated. The collection is still too raw to use as significant value

Once this aspect of the signal has been corrected, we can start considering how the signal will be represented when a patient performs the task. From theory, we know that data related to an accelerometer can be interpreted in this way:

- Value 0: our sensor does not undergo any acceleration variations;
- Increasing values: our sensor is affected by an acceleration increase;
- Decreasing values: our sensor is affected by an acceleration decrease.

Now let's consider the Sit-to-Stand steps described earlier with the interpretation of the accelerometer data just listed.

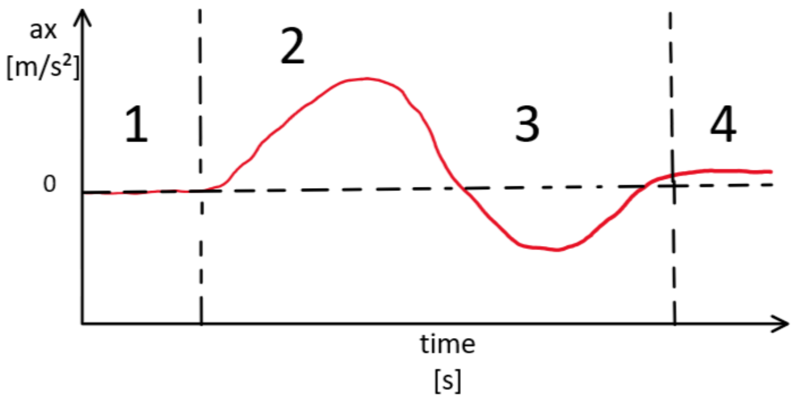


Figure 32 In this figure, it is shown from a logical point of view how a Sit-to-Stand task should appear in terms of a uniaxial acceleration analysis of the movement.

During the sitting rest (phase 1), our subject will be still without significant acceleration variations, which means that our acceleration in this phase should appear, after pre-filtering, as

a zero interval. The same reasoning is also applied to the standing rest phase, in which the subject will be identically immobile.
The significant phase of the task that we want to highlight is the transition from the sitting to the standing position (phase 2-3). Objectively analyzing the movement, in order to execute this change of state, we need to activate our muscles and exert a force greater than our weight to start the movement, thus obtaining a positive acceleration of the movement. Once the force generated by our body allows the movement, we automatically start to slow down to reach the final position, which is the standing position.
Thinking about this movement, the resulting signal should appear as an initial and final phase almost at zero, like in *Figure 32,* and an intermediate phase of acceleration-deceleration with a sinusoidal shape.
As said before our sensor will be affected by outliners, so our movement before being processed will be like this:

Figure 33 In this figure, a collection of raw data is shown before being processed by our Sit-to-Stand task pipeline.

As shown in *Figure 33*, we can say that the collected signal has physical meaning, but it is still too raw to be used. Once we have obtained this new raw signal, we can start processing the data: the ultimate goal is to eliminate non-significant data, highlight the values related to the Sit-to-Stand task, and finally calculate some significant values of this task.
In order to improve the quality of a digital signal, a technique of primary importance is filtering, with which certain features can be emphasized or others removed.[18] The work we now have to carry out is called smoothing: in statistics and digital image processing, smoothing or, better, equalization of a set consists of applying a filter function whose purpose is to highlight significant patterns, attenuating noise generated by environmental, electrical, electronic,

computer, or physiological artifacts or other disturbance phenomena related to very small scale factors or high-speed phenomena.[18]

Given the nature of our signal, we have decided to use a low pass filter: moving average filter. This particular filter allows smoothing of the signal through a window of selected length that is shorter than the total length of the signal (usually of odd length), generating series of averages that replace the value under consideration.

From a mathematical point of view, this particular filter is executed in this manner:

Let S be a vector containing the observed values for times t=1,2,...,T. We choose the length of our window, which we will call k, two intervals i_1, i_2 such that k= i_1+i_2+1. Our two intervals will be considered as a group of numbers before and after the observed data point. Finally, we define a weight w_i.

We define the calculation of the value at time t as follows:

$$S'(t) = \frac{1}{k} \sum_{i=-i_1}^{i_2} (w_i S_{t+1})$$

If we start from the first value, we can use two methods: add the same value of the first element for all the length of the left interval or starting from the middle position keeping the first i_1 value unchanged.

There are different types of moving averages, which differ from each other simply in the calculation formula, thus generating signals that are more or less sensitive to data variations.

From the knowledge gained in [18] and by definition given by [E], we know that there are fundamentally three types of moving averages:

- The Simple Moving Average (SMA) is the most commonly used and easiest to calculate moving average. The data for a specific period is considered and the average is calculated by summing them together and dividing by the total number of values.
- Weighted Moving Average, (WMA) was designed to overcome the problem of simple moving averages regarding the weight to be assigned to the values considered.
- Exponential Moving Average (EMA): This moving average is generated by a much more complex calculation system that always tries to eliminate the shortcomings of the simple moving average

Given the data type, the sporadic presence of outliers, and the difference between correct values and spikes, the best implementation choice is to use a weighted average, giving more weight to correct values than to spikes. However, pre-filtering allowed us to eliminate outliners easily, so we applied the SMA filter because after pre-filtering our values have the same importance.

The next step is to highlight the significant data of the task by trying to eliminate the two rest phases that are not of interest in our case. The rationale behind the next algorithm is to use peak detection techniques to highlight the interval representing the change of state, i.e., as mentioned earlier, the sinusoid seen in *Figure 33*.

Peak detection is a procedure performed with automated algorithms capable of recognizing significant events in the signal, namely peaks as we can see in the example below.

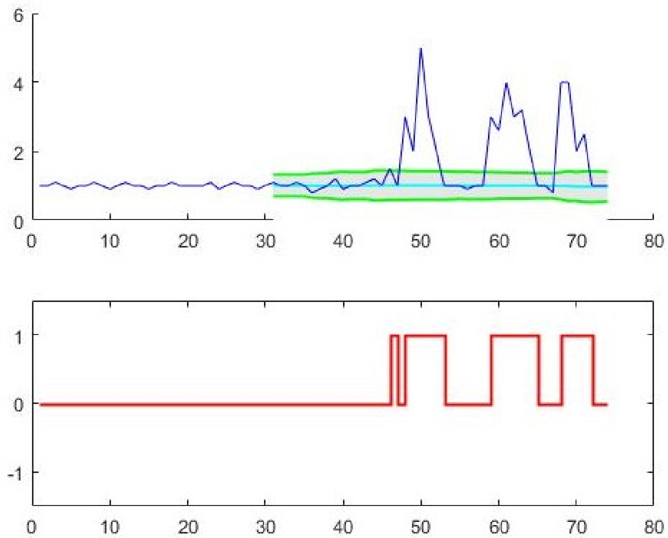


Figure 34 In this figure, we have two graphs: in the upper one, we have a series of example data that simulate peaks highlighting the variability bands of the data. In the lower graph, instead, it is shown how a peak detection algorithm is able to represent the peaks of the above graph.

The algorithm operates on the principle of dispersion, which involves signaling when a new datapoint deviates from a moving mean by a certain number of standard deviations (also known as a z-score). The algorithm is highly resilient because it calculates a separate moving mean and deviation, thereby preventing previous signals from influencing the threshold. As a result, future signals can be detected with consistent accuracy, regardless of the number of previous signals. This algorithm take all the signal and processed it putting all the values over a threshold equals to ±1 and the other values equals to 0, this threshold is calculated a z-scored algorithm that use mean and standard deviation.
This algorithm will return a signal composed by 0, +1 and -1 where the ones are the peak that the algorithm find during signal analysis as in the *Figure 35*.

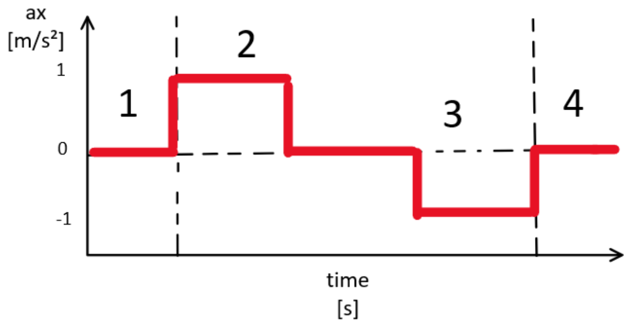


Figure 35 From a logical point of view, if we combine figures 34 and 36, our hypothetical Sit-to-Stand graph will be represented as shown in this figure.

After the peak detection algorithm our movement will be highlighted like the figure above, but now we want to cut the signal to show only the interval that describes the task, i.e., only phase 2 and 3.

The thing that is immediately noticeable, even at a logical level, is that the principal Sit-to-Stand phase is identified by the algorithm as a sequence of +1, followed by a sequence of 0, and finally a sequence of -1, caused by the acceleration-deceleration movement.

The algorithm that we implemented in our application will cut our signal, to show only the interval that describes the task. To found this sequence of numbers we use the signal given by the Z-score algorithm composed only by 0 and ±1, looking for the sequence of numbers that reflects the hypothesis just described to recognize the movement and take the position of starting and stopping Sit-to-Stand phase. Once found the position we cut the signal immediately before and after and cut the smoothed signal, obtaining a signal derived from the original one, but which shows only the principal phase.

Now this is how our signal in *Figure 33* pass from the raw signal to the processed signal that enhance the sit to stand movement:

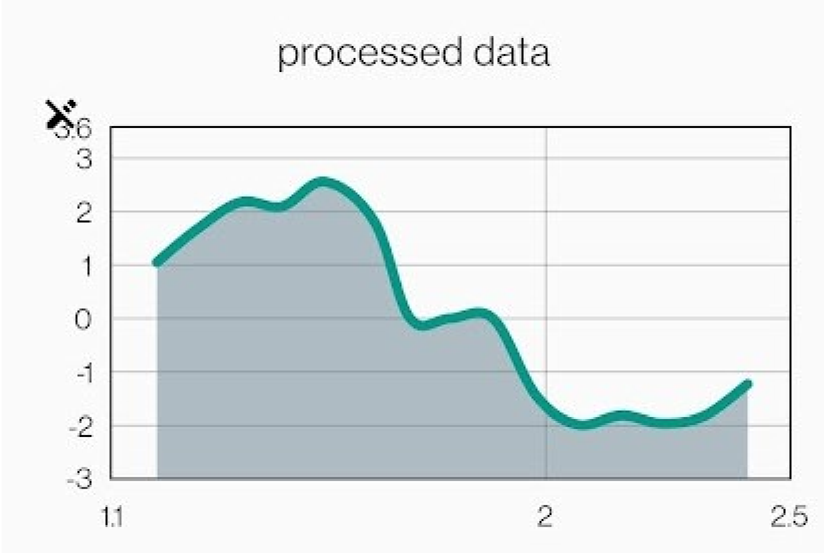


Figure 36 This image shows the final product of our pipeline, in which the raw signal from Figure 35 is processed and the portion of interest representing the execution of the Sit-to-Stand task is cropped.

In accordance with the literature, the fundamental data to highlight, in addition to the graph, that allow an optimal description of our task are the maximum acceleration and the execution time. For the first value, a simple research of the maximum absolute value was performed on the signal cleaned from the outliers and it will show the maximum acceleration reached during the task. For the task execution time, instead, we used the signal cutting algorithm that allowed us to obtain the significant positions that describe the boundaries of the task, and therefore this time is calculated as a simple difference between the end time and the start time of the task.

The next step we can take with the application is to save the data in our database so in the next chapter, we will describe in detail the application we have created to acquire and process the data.

4. Automatized Sit to Stand Application

In this chapter will be described the technology used and the application step by step starting from a general point of view and subsequently, we show more in detail how the application work.

4.1 Technology used

In this thesis one of the main objectives is to create an Android/iOS application that, through the Bluetooth Low Energy protocol, connects to the 221e sensor described in chapter 2.
First, we need to differentiate the BLE protocol from the classic Bluetooth protocol.

Figure 37 BLE vs classic Bluetooth

The term "Low Energy" in the BLE acronym implies its purpose: while Bluetooth Classic is designed for transmitting continuous streams of data like music playback, BLE is optimized for power efficiency. A BLE device can operate on a small battery for weeks, months, or even years, making it ideal for sensor-based or Internet of Things (IoT) applications.[H]
Another key difference between Bluetooth Classic and BLE is that BLE is more developer-friendly. By allowing developers to define various custom profiles for different use cases, BLE opens up a world of endless possibilities, while Bluetooth Classic mainly supports the Serial Port Profile (SPP) for sending custom data. [H]
The Android Bluetooth API is not very straightforward for Bluetooth Classic use cases due to several reasons:

- The Bluetooth Classic scanning API uses a messy, multicast Broadcast Receiver that is typically avoided in modern Android development.
- The Android SDK requires Bluetooth Classic devices to be paired with Android before establishing an RFCOMM (Radio frequency communication) connection, unlike the BLE use case, which does not have this restriction.
- The Android SDK provides implementation for only a limited number of Bluetooth Classic profiles out of the box.
- Once paired and connected, we need to manage a dedicated Android Thread object to communicate with a Bluetooth Classic device via the BluetoothSocket object, which is flexible but also error-prone.[H]

	Classic Bluetooth	Bluetooth Low Energy (BLE)
Rate of Data Transfer	2-3 Mbps	200 Kbps
Time Consumption	Approx. 100ms	Approx. 3ms
Power Consumption	~30mA	>15mA
Applications	Used in a continuous stream of data like headphones.	Does not require a continuous stream of data like proximity marketing campaigns.

Figure 38 This table represent the main difference between the classic Bluetooth and the Bluetooth Low Energy protocol

For the reasons written above and for the difference highlighted in *Figure 38*, our application will implemented the BLE protocol for managing the communication between application and MUSE.

A device acting as a *central* (a device that scans for and connects to BLE peripherals in order to perform some operation.) can establish connections with multiple peripherals simultaneously, but a BLE device acting as a *peripheral* (a device that advertises its presence and is connected to by a central in order to accomplish some task) can typically only interact with one central at a time. When a BLE device is connected to a central, it usually stops advertising as a peripheral because it's no longer available for connections.

The server (peripheral) holds a GATT, Generic ATTribute Profile, database that provides information accessed by the client (central) via BLE.

The essence of BLE communication can be summarized into three common operations:

- Write: The client (app/software) writes some bytes to a characteristic or descriptor on the server (BLE device). The server's firmware processes the write and performs some server-side operation in response to it.
- Read: The client (app) reads the value of a characteristic or descriptor on the server (BLE device) and interprets it based on a previously established protocol.
- Notify/Indicate: The client subscribes to a characteristic for notifications or indications, and the server notifies the client when the value of the characteristic changes. [H]

There are many APIs created to allow the implementation of the BLE protocol in Android/iOS applications, in our case we used the FlutterBluePlus package.[I]

As written on pub.dev, FlutterBluePlus is a Bluetooth plugin for Flutter, a new app SDK to help developers build modern multi-platform apps and in this API are implemented the basic function used for creating and managing the communication between central and peripherals.

As *pub.dev* [I] and *punchthrough* [H] said, the main function that we need to implement for managing a BLE connection are:

- Device scanning function permit the application to discover the Bluetooth devices around.
- Device connection function allow to connect the application to the device.
- Device discovery services, this function permit to analyze all the usable services of the device returning a list of all the services supported by the BLE device.
- Read and write characteristics, permit us to communicate with the device.
- Set Notify and listen to change, permit to control the listen function of the communication with services and characteristics.

Before introducing the application created we have to understand also how our MUSE sensor works, in particular now we will describe the MUSE V2 communication protocol used for managing the communication between app and sensor and for understanding the value listened by the app.

4.2 MUSE V2 Protocol

As mentioned in Chapter 2, the device used, MUSE, was provided by the company 221e with its own communication protocol [19] used within the application.

The project framework incorporates systems that adhere to a bidirectional communication protocol using Bluetooth Low Energy with a 20 Byte Type Length Value (TLV) format.

Type	Length	Value		
8 bit unsigned int	8 bit unsigned int	Byte1	....	ByteN

Figure 39 Representation of a 20 Byte Type Length Value (TLV) format and how is divided

Each packet includes a byte indicating the type, such as a command, a byte indicating the length of the value, such as the length of a string, and the value itself, such as the string. An ACK response, which can be positive (indicating success) or negative (indicating an error), follows each TLV packet and carries information regarding the outcome in the same TLV format. As written in this protocol, the ACK response includes the ACK_CODE, indicating that the message is an ACK, the length of the value, and the value itself. The value length is at least 2 bytes and contains information about the ACK's reference message and error code. If the previous command requires information in response, the bytes following the error code byte will contain the actual information.[19]

Type	Length	Value		
ACK_CODE	8 bit unsigned int	COMMAND	ERROR CODE	...

Figure 40 Architecture of an ACK message with the relative field of Type, Length of the payload and the Payload

All the commands present in the protocol are differentiated into READ and WRITE code, and further into MASTER TRANSMIT and MASTER RECEIVE message. Every read and write command are differentiated by a factor of 0x80: For example if we want to set a state (write) the code is 0x02 and if we want to know the device's status the code is 0x82. To use a specific command, the user must enter the exact sequence of bytes as written in the "MASTER TRANSMIT" line. The sequence of bytes displayed in the "MASTER RECEIVE" line serves as a guide for the system's response, which may differ from that particular sequence but should adhere to the same underlying logic.

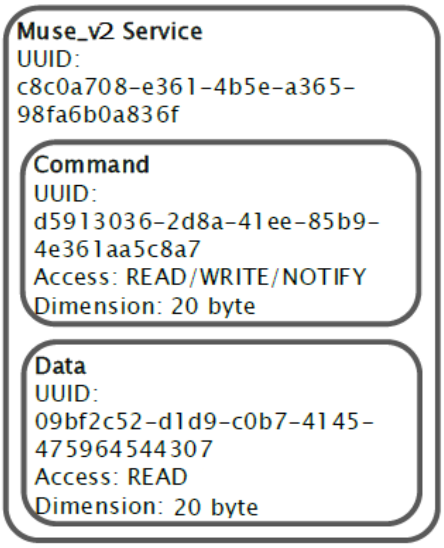


Figure 41 This image highlights how the custom service used to communicate with the sensor contains the two characteristics where the various commands will be written and read.

All the command that we will transmit to the sensor will be sent to the custom service with uuid (Universally Unique Identifier) “c8c0a708-e361-4b5e-a365-98fa6b0a836f”, more precisely to the characteristic with uuid “d5913036-2d8a-41ee-85b9-4e361aa5c8a7” and for the response we have to read this characteristic for all the one shot command.
An Example of how the protocol works can be the request of the battery charge: if we want to know the percentage of charge of the MUSE sensor we can use the relative command to access.

As written in this protocol, the master (app) had to transmit the master command with code 0x87 followed by the length of the payload and the relatively payload like that:

CMD	LEN	PAYLOAD
87	00	0000….00

Table 2 This table represents how the MASTER TRANSMIT command is structured with the TLV format

As we can see the length of the payload is zero, so when the master ask for battery charge, it can transmit only CMD and LEN, because the system understand that is no a following payload. Once the master transmit the command 0x8700, the device respond, as said before with a 20 Byte length response as follow:

ACK	LEN	CMD	ERROR	PAYLOAD
00	03	87	00	5F00….00

Table 3 This table represents how the MASTER RECIVE command is structured with the TLV format

In table 3 we can see that the sensor understand the request and response with a 2 Byte ACK 0x00 followed by the length, the command, the error and the payload. As suggested by the ACK, the command sent was successful, and this can be understood from the received CMD which corresponds to the one sent and the absence of negative bytes in the ERROR section. In the payload we have the value of 0x5F that is the hexadecimal of 95 decimal so this response want to communicate to us that the sensor have 95% of battery.

Essentially the command in this protocol are divided in two types:

- One shot command: this type of commands are, like battery charge, the easiest way to communicate with the sensor with a ping pong of request-response. Another example can be: Device status (0x82), device voltage (0x88) or get time (0x8B).
- Streaming command, this type of command permit the device to transmit continuously value changing sensed by the sensor until another one shot command stop it.

The main difference of this two type is that for the first one we will use only the first characteristic for communicating with the sensor, while the second one use the first characteristic as a start and stop and the second characteristic for reading continuously the data transmitted by the sensor.

The principal commands used in this application are:

- Device status read 0x82 and write 0x02. With this command we can understand the information of the status of the sensor, for example if is set as IDLE or in STREAM mode.

SYS_STARTUP: payload length 1; "02 01 01".
SYS_IDLE: payload length 1; "02 01 02".
SYS_STANDBY: payload length 1; "02 01 03".
SYS_LOG: payload length 3 (code of the state, log mode, freq mode).
STS_TX: payload length 3 (code of the state, stream mode, freqe mode).

Figure 42 This figure represent some example of how will be the master transmit command 0x02 and the difference in term of format between log and streaming state and idle state (the first ones have a bigger payloads)

As shown in the figure above we can set different type of status, in particular, in our application, we will use the last one: STS_TX that is the acronymous of System Streaming Transmit. STS_TX need a payload where will be present the code of state, modality and the frequencies for a correct implementation, if this code is not present in the payload of the master transmit, at the time of sending, the device will respond in master receive command with message containing the error code in the ERROR section. The status and streaming possibility are:

Application Mode							Application and Bootloader modes	Bootloader Mode		
0x01	0x02	0x03	0x04	0x05	0x06	0x07	0xFF	0xF0	0xF1	0xF2
SYS_STARTUP	SYS_IDLE	SYS_STANDBY	SYS_LOG	SYS_READOUT	SYS_TX	SYS_CALIB	SYS_ERROR	SYS_BOOT_STARTUP	SYS_BOOT_IDLE	SYS_BOOT_WRITE

0x00	0x01	0x02	0x03	0x04	0x05	0x08	0x09
NONE	IMU	MAGN	9DOF (IMU / MAGN)	HDR	IMU / HDR	ORIENTATION	IMU / ORIENTATION

0x00	0x01	0x02	0x04	0x08
NONE	25 Hz	50 Hz	100 Hz	200 Hz

Figure 43 This image represent all the possible combination in terms of STATE (up), MODE (middle), FREQUENCY (down) that we can set ours sensor

For example, if we want to set(0x02) our device in streaming state(0x06), IMU(0x01) modality with 50Hz(0x02) of frequencies we will use the command:

CMD	LEN	PAYLOAD
02	03	0601020…0

Figure 44 this table represent the TLV format for setting the sensor into a 50Hz streaming IMU status

As written before, the system recognize a payload length of 3 Byte so we can transmit only the significant information without the necessary zeros to complete the payload length.

- Full scale read (0xC0) and write(0x40) that permit us to read and set the full scale of the sensors.

MASTER TRANSMIT

CMD	LEN	PAYLOAD																	
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

MASTER RECEIVE

ACK	LEN	CMD	ERROR	PAYLOAD															
00	04	C0	00	20	18	60	30	00	00	00	00	00	00	00	00	00	00	00	00
				axl FS	gyro FS	mag FS	axl HDR FS												

Figure 45 This figure show how are structured the master transmit and master receive of a full-scale command 0xC0

With the acknowledge of the full scale setting we can use the relatively sensibility value to convert the payload into a significant value.

All the hex data transmitted by the sensors are unsigned int values with little endian architecture. Once we get the payload and the full scale value we can convert the hexadecimal string into an unsigned int and after we have to multiply it by the full scale value for have the real value sense by MUSE.

AXL FS [g]	2	4	6	8	12
CODE	0x00	0x08	0x10	0x18	0x20
RESOLUTION	0.061	0.122	0.183	0.244	0.732
GYRO FS [DPS]	500	1000	2000	4000	
CODE	0x00	0x08	0x10	0x18	
RESOLUTION	0.0655	0.0328	0.0164	0.0082	
MAG FS [G]	2	4	8	12	
CODE	0x00	0x20	0x40	0x60	
RESOLUTION	0.08	0.16	0.32	0.479	
AXL HDR FS[g]	100	200	400		
CODE	0x00	0x10	0x30		
RESOLUTION	49	98	195		

Table 4 This table describe all the sensor full scale with relative resolution and code

The sensor has multiple settings available that can be used, but for the data collection performed by the application, we will use the sensor in IMU mode with a sampling frequency of 25Hz.

For the streaming communication we have to write the state in the first custom characteristics and continuously reading in the second custom characteristic with uuid “09bf2c52-d1d9-c0b7-4145-475964544307” until another set state will be written in the first custom characteristics.

In the next paragraph, we will therefore combine Bluetooth technology with the communication protocol of 221e in our application.

4.3 Introduction of the app

As described in the previous chapters, this application aims to replace/automatized the actual data acquisition process for the Sit to Stand task.

Figure 46 This figure shows how the Sit-to-Stand task analysis is currently performed, which only involves the use of a handheld stopwatch by the doctor.

At medical and practical level, the task is studied very quickly and approximately, without analyzing the movement: the physiotherapist or whoever analyzes the task, equips themselves with a stopwatch which they use to calculate the execution time of the patient's movement. With this previous mode of acquisition, therefore, there is only a view of the time taken by the patient, healthy or pathological, to fully execute the task.

The analysis is therefore lacking in multiple pieces of information that could be acquired with a simple inertial sensor placed on the subject: our application comes to allow those who study the patient's movement to acquire more significant data and enable algorithms to analyze the movement in its entirety with a low cost and quick preparation for data acquisition.

The strength of the current data acquisition method is the quick preparation for the task and the extremely low cost: we only need a chair and a smartphone, in fact, the stopwatch inside the doctor's smartphone is often used directly.

Assuming that MUSE is a small-sized pocket sensor, approximately 1.5x1.5cm, and Bluetooth is a protocol implemented in every smartphone on the market for several years, we can say that the introduction of the new connection step between the sensor and our app, as well as the positioning on the patient, is of negligible speed, and therefore the speed factor can be considered stable.

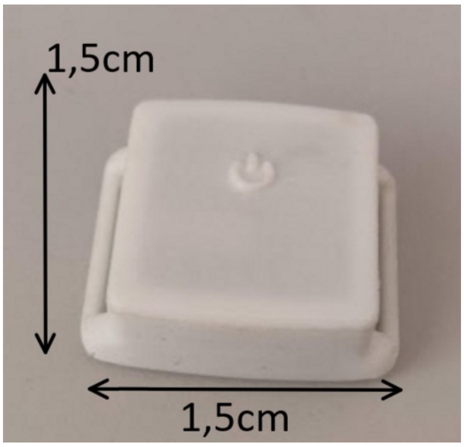


Figure 47 Dimension of MUSE sensor used in this thesis

Subsequently, the multiple advantages that the use of the application offers compared to the current data acquisition method will be explained, such as the ability to save information, calculate new values that describe the task, and much more. The name chosen for the application is ASiStApp that is the acronymous of Automatized Sit-to-Stand Application.

Figure 48 Application Logo

As written in the previous chapter, flutter applications use a tree structure for the pages organization. Below is shown the structure of our application, highlighting the dependencies and links between the pages of our application.

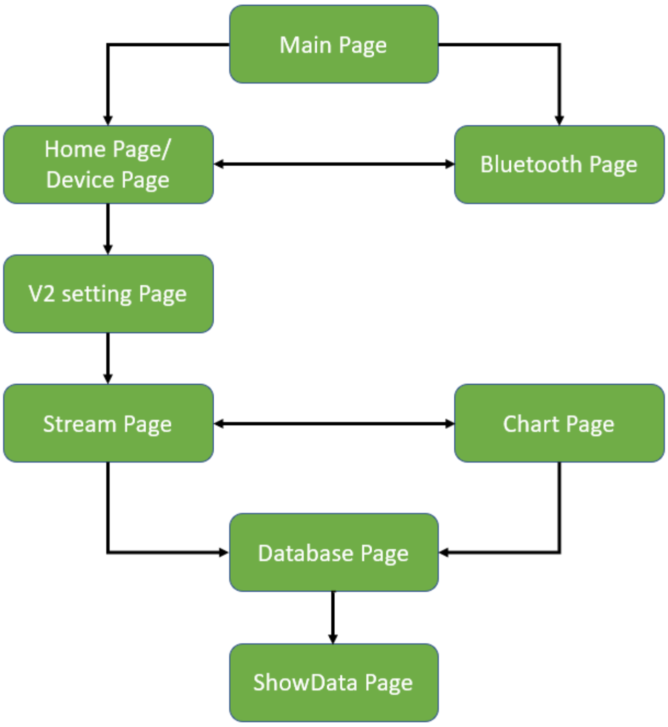


Figure 49 This image represent the organization of the pages inside our apllication

ASiStApp is an application composed by this page:

- MainPage: the main function of the application is simply used for initializing the database connection and starting the application. An essential requirement for the correct use of the application is to have Bluetooth and location services active, therefore, before redirecting to the HomePage, there is a check of these two functions. If they are not active, the user will be redirected to the BluetoothPage.
- BluetoothPage: This page will notify that Bluetooth is turned off and will ask to enable the connection and location services.
- HomePage: In this page we will see two different list of device. First one is a list of all the device already connected with ours smartphone and the second one is a list of object that represent all the device that our smartphone find via Bluetooth scanning. When we find MUSE we can connect this device through DevicePage touching the button ‘Connect’.

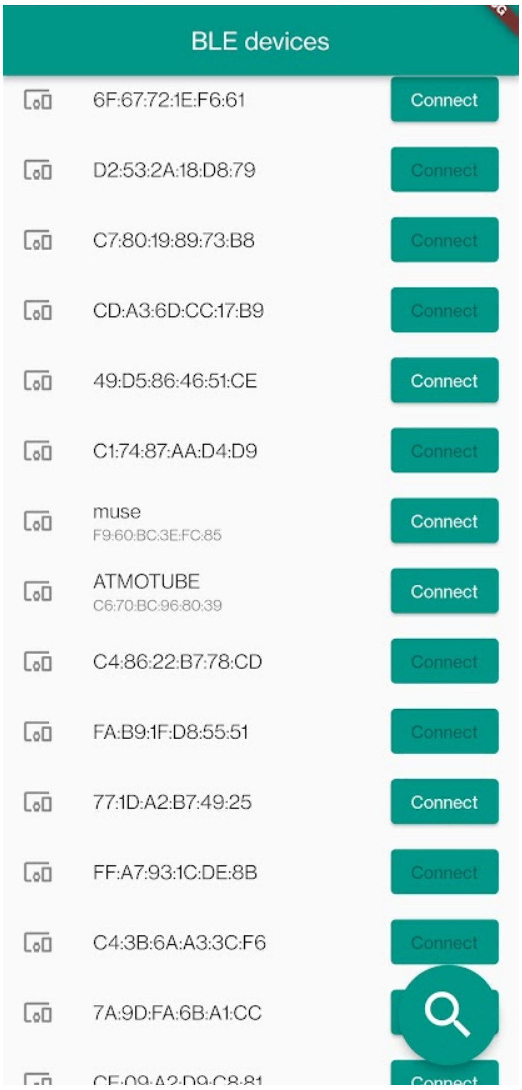


Figure 50 This figure is a screenshot of our homepage that show in the middle of the list of devices the sensor used in the thesis: muse

- DevicePage: This page is only use to stabilize a connection between our smartphone and MUSE and for finding the custom service of the device. Once the connection is stabilized we will go be sent to V2SettingPage.

Figure 51 This is a screenshot of the message that our application will show after the connection between app and device successful.

- V2settingPage: Once the custom services have been found, we want to set up our device, that is, choose which type of sensor to use, the sampling frequency, and the full-scale values of the respective sensors. If nothing is changed, IMU at 25Hz will be set with full-scale values given by the default settings.

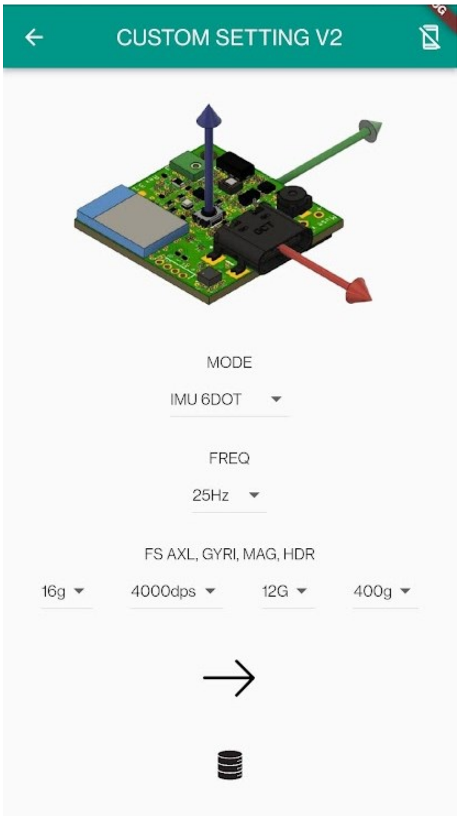


Figure 52 This screenshot of the V2SettingPage show that in this page the user can easily change modality, frequencies and full scale of the sensor.

- StreamPage: On this page, the user will calibrate the sensor following the instructions provided by the application and start data acquisition by clicking the play button. To stop the acquisition, the pause button must be clicked, and once clicked, the button to switch to the ChartPage will appear, where the graph of the obtained data will be displayed.

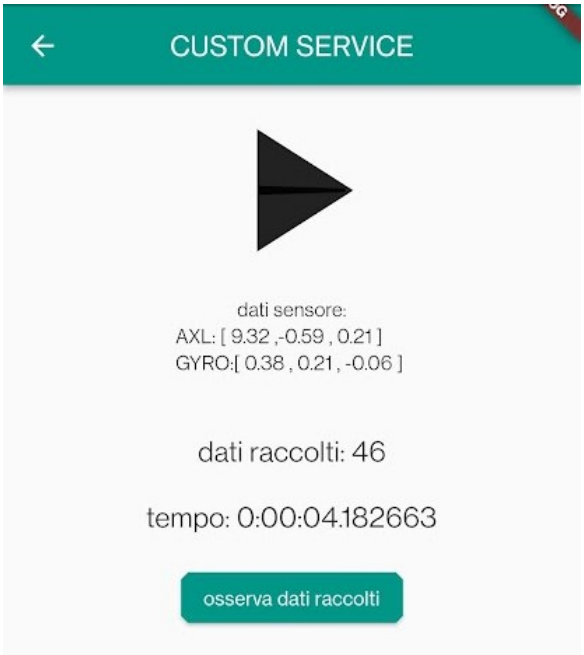


Figure 53 In this screenshot are represented the StreamPage structure where there is a play-pause button for starting/stopping the collection, below are the real-time representations of the changing values of the accelerometer and gyroscope. Below that, the total execution time of the task is shown. Finally, at the bottom, there is a button that will appear only once the data collection is finished, which will allow us to move to the ChartPage

- ChartPage: Here the user can see the data acquisition obtained in the StreamPage, and in the graph, it is possible to dynamically switch back and forth between the raw signal and the processed signal, in which only the data related to the task are highlighted. If we want, we can save this data by entering the name and surname of the subject in the dedicated space and then proceeding with the data saving in our database or we can directly go to the database with the icon on the high-right part of the screen. If we touch the database icon we will ben sent on DatabasePage.

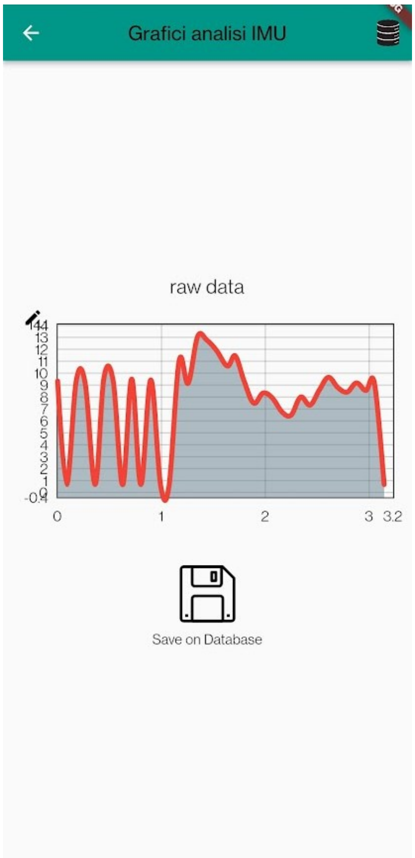
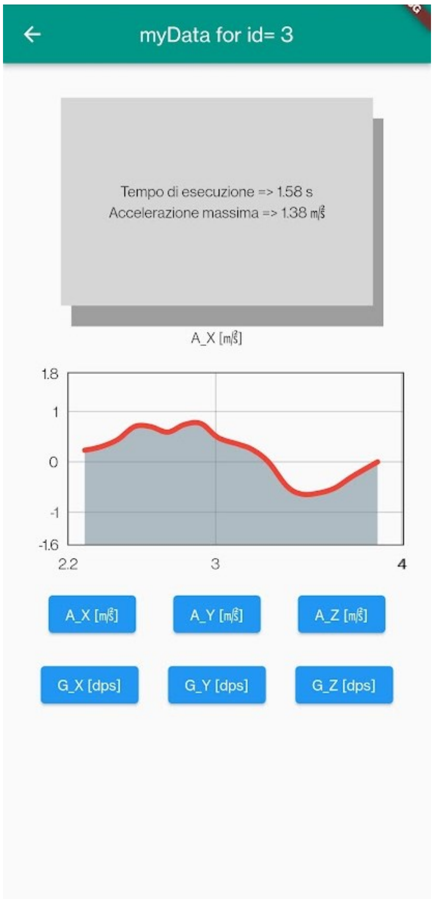


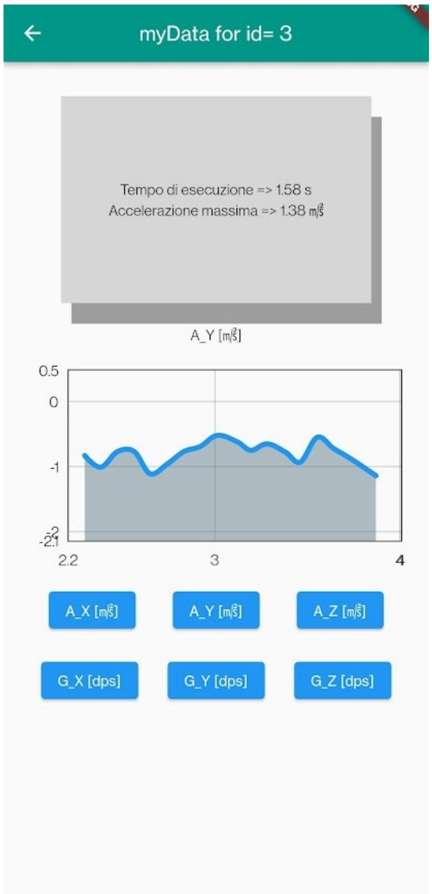
Figure 54 In this page, the raw representation of the signal collected from our sensor is shown. By clicking on the pencil icon on the top left of the graph, the post-processing graph will be displayed.

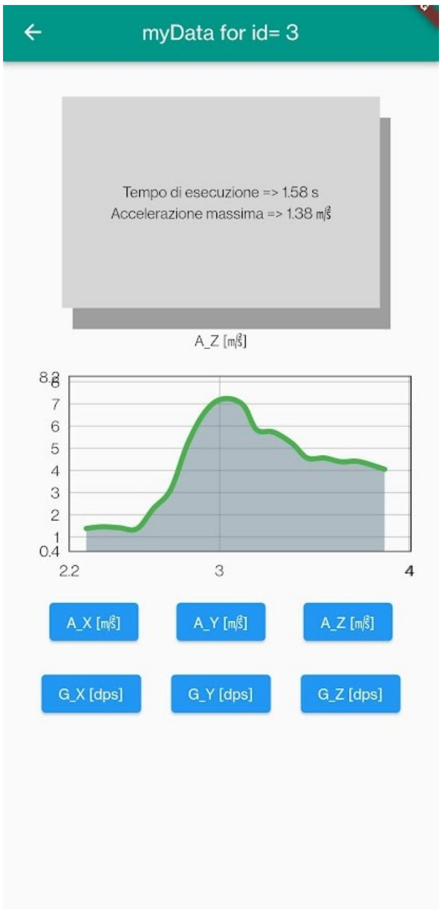
- DatabasePage: In this page we can see a list of all the task performed by all the subjects. If we can’t find a precisely task we can use the search icon and by entering name of the subject we will see a restricted list of task performed by the subject that which we searched for. If we press the back button where once was the search button we will see again all the task on our database.
  Considering a task, if we click on it, we will be directed to the ShowDataPage, where we can see all the data collected for that task. If, on the other hand, we want to delete a task, we can simply swipe to the side, and it will be deleted from the database. At the bottom of the screen, there is a download button that, when clicked, will save a text file of all the data in our database, sorted, to our device's Download folder.

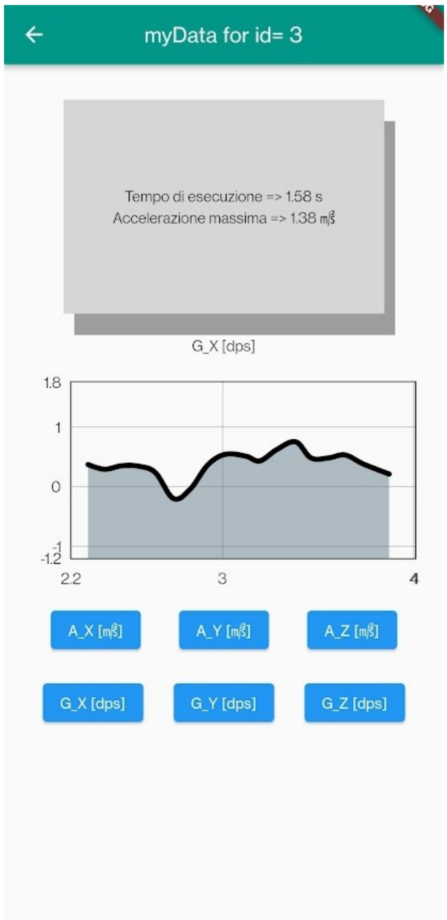
Figure 55 This screenshot represents the structural view of our database, which is a list of sessions that have been saved on this device. In the top right, there is the option to search by name, and by clicking on any session, we will be directed to a more comprehensive view of the saved data. e

- ShowDataPage: This page shows all the data related to the chosen task in the DatabasePage. In particular, we will find data related to the execution time of the task, the maximum acceleration during the task, and a dynamic graph that changes based on the button clicked, in particular, saved data related to triaxial acceleration and triaxial gyroscope will be shown.


myData for id= 3
Tempo di esecuzione => 1.58 s
Accelerazione massima => 1.38 m/s^2
A_X [m/s^2]
A_X [m/s^2]
A_Y [m/s^2]
A_Z [m/s^2]
G_X [dps]
G_Y [dps]
G_Z [dps]


myData for id= 3
Tempo di esecuzione => 1.58 s
Accelerazione massima => 1.38 m/s^2
A_Y [m/s^2]
A_X [m/s^2]
A_Y [m/s^2]
A_Z [m/s^2]
G_X [dps]
G_Y [dps]
G_Z [dps]


myData for id= 3
Tempo di esecuzione => 1.58 s
Accelerazione massima => 1.38 m/s^2
A_Z [m/s^2]
A_X [m/s^2]
A_Y [m/s^2]
A_Z [m/s^2]
G_X [dps]
G_Y [dps]
G_Z [dps]


myData for id= 3
Tempo di esecuzione => 1.58 s
Accelerazione massima => 1.38 m/s^2
G_X [dps]
A_X [m/s^2]
A_Y [m/s^2]
A_Z [m/s^2]
G_X [dps]
G_Y [dps]
G_Z [dps]

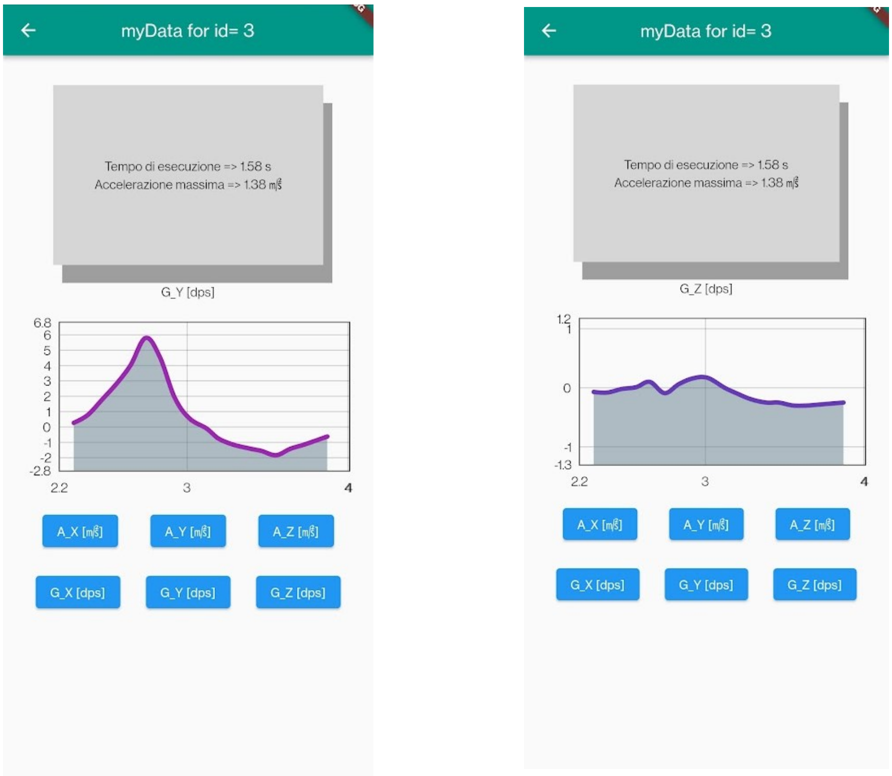


Figure 56 These screenshot represent all data saved for patient with ID=3

On the next paragraphs we will describe precisely our function and how the application work trough the code of the 3 main page of the application: StreamPage, ChartPage and DatabasePage.

4.3.1 Main pages of application

As we said before now we will describe the code behind the application, starting from the StreamPage. The onerous tasks of this page are first to calibrate the gravity force.

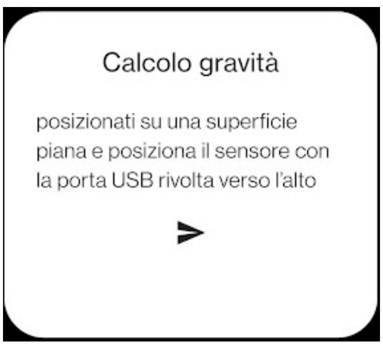


Figure 57 This alert dialog will be shown before starting the collection, that permit to calculate a better estimation of the gravity force sensed by the sensor in that moment

The calibration is a simple function where for 4 seconds we listen our sensor accelerometer and calculate the gravity with an average function where we don't consider the outliners.

The second task of this page is to write the mode and frequency chosen, on the first characteristic as set in the previous page, V2SettingPage.

Once the mode and selected frequency are set, this page saves the converted data transmitted by the sensor to the application. The conversion of the transmitted payload is done in accordance with the communication protocol guidelines described earlier. This page has the responsibility of constantly converting the values and showing real-time variations of the accelerometer and gyroscope.

In the following figure we will see how our application convert the data transmitted by MUSE into significant values for the user.

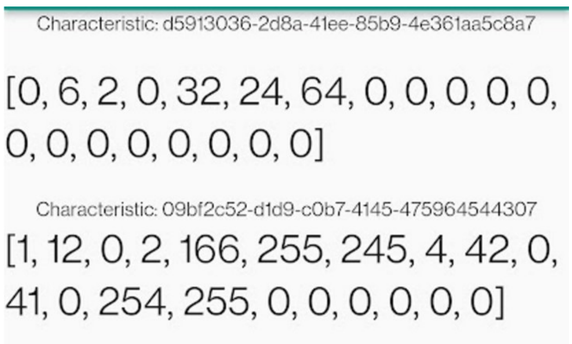


Figure 58 This screenshot represent how our application receive data from the sensor with two 20 Bytes TLV format. The vector above is fixed and represents the response to the streaming command with its corresponding full scale values, while the vector below is continuously changing and contains the values of the respective sensors according to the format mentioned earlier.

As can be seen from the figure above, the sensitivity values related to the scale setting are extracted from the first characteristic as follow:

ACK	LEN	CMD	ERROR	PAYLOAD
00	06	02	00	20184000

Table 5 TLV format for master receive example during streaming transmission

In the table above the payload is already converted into hexadecimal values (i.e 32_{10}->20_{16}). Taking the payload of *Table 5* with the code of *Table 4*, by comparing these two tables, it is possible to obtain the corresponding sensitivity values of the sensors, which in this case will be:

AXL	GYRO	MAGN	HDR
0.732	0.0082	0.32	49

Table 6 According to the table seen before this table represent the sensitivity value from the full scale receive

Once the sensitivity values are obtained, we can analyze the command received in the second characteristic. In our case, the sensor is set in IMU 6 DOF mode, so our second characteristic will be read as follows (we will explain the conversion only for x axis accelerometer, same steps will be done for the remaining values in the application):

MODE	LEN	PAYLOAD						
		AX	AX	AY	AY	AZ	AZ	GX,GY,GZ
1	12	00	02	A6	FF	F5	04	2A002900FEFF

Table 7 This table represent how is structure the TLV format during a streaming communication through the characteristics value.

The values in the table above are related to the positioning with the Z axis upward during a movement along the Y axis, so, we expect a null X-axis values, a gravity component on Z, and an acceleration on Y.

As mentioned earlier, the conversion of axis values is done by taking various bytes and processing them with a conversion to a 16-bit unsigned int in little-endian architecture. Once this value is obtained, it will be multiplied by its sensitivity, which in our case is 0.732, and we will obtain our value in mg. To obtain the final value in m/s^2, we need to convert mg to g and then g to m/s^2 by dividing by 1000 (mg to g) and 9.80665 (g to m/s^2), respectively:

Accelerometer value	Conversion(mg -> m/s^2)
(A_X) 0x0002	0.23
(A_Y) 0xA60F	12,85
(A_Z) 0xF504	9,10

Table 8 In this table is show the conversion from hex to m/s^2 of the value sensed by our sensor

As can be seen in the table above, the obtained values reflect the previously described movement. This conversion can be performed on all sensors present in MUSE by using their sensitivity values and properly converting the units of measure described in the communication protocol. [19]

Once the user stops the data collection with the icon show in *Figure 56*, a button will be displayed and if we click the button we'll pass on ChartPage.
As we can see in the *Figure 54*, the first things that we'll see is our raw data with outliners and some spikes, but if we click on the pencil icon on the top left of the chart, we will immediately see the data processed as follow:

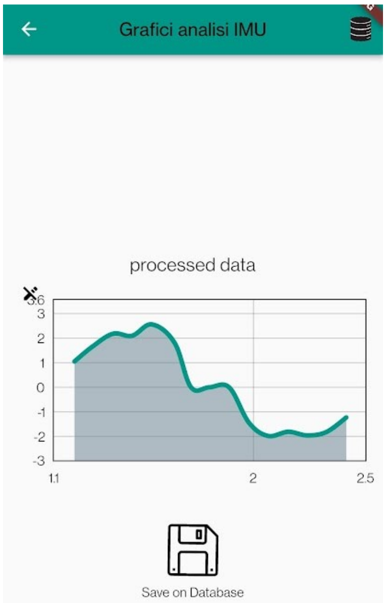


Figure 59 In this figure is shown what will happens if we click the pen in Figure 57, this page will show the processed data

In this particular page we will see the data processed offline with a clear vision of the data collection before and after the algorithms already explained in the previous chapter.

Last main page is DatabasePage, but first we have to understand how is composed our database. As we know, to store and organize data collected by wearable sensors, a database is required. There are several advantages of using a database in an application:

- Data storage: Databases provide a structured and organized way to store and manage large amounts of data, making it easier to access and retrieve information.
- Data integrity: Databases ensure the integrity of the data by enforcing rules and constraints, preventing inconsistencies and errors that can occur with manual data management.
- Data scalability: Databases can easily scale to accommodate large amounts of data, making it easy to add new data as the application grows.
- Data analysis: Databases provide powerful tools for data analysis and reporting, allowing developers to extract meaningful insights and trends from the data.

In this thesis, we want to store the data related to our collections. In particular, for each session, we want to save an ID, the subject's name and surname, and the date of the experiment. For each session, we want to save the data related to the goals calculated by the application through the algorithms: the execution time of the movement and the maximum acceleration. Furthermore, for each session, we want to save all the data related to the accelerometer, gyroscope, and their respective timestamps.

Given the problem just described, the related diagrams are shown below.

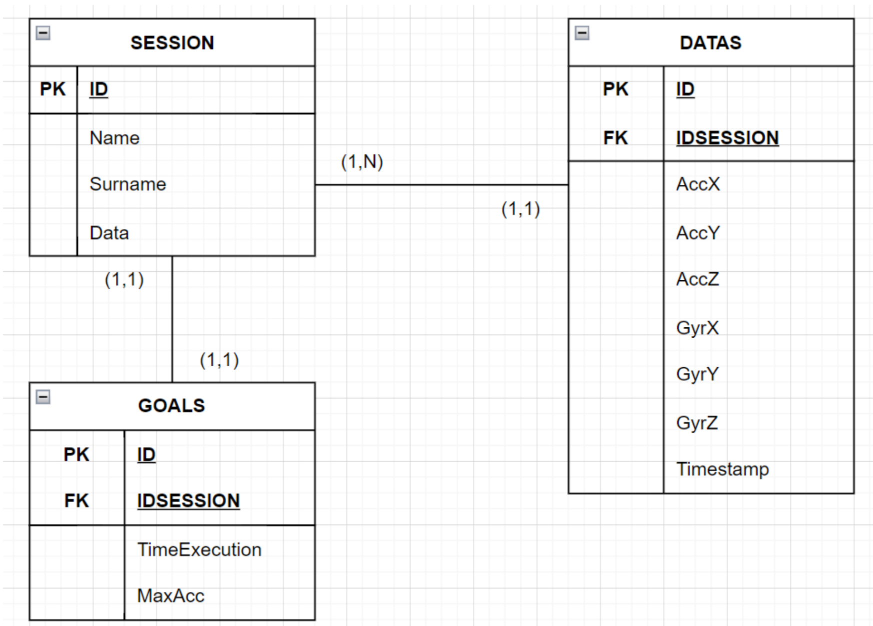


Figure 60 Entity-Relationship model (up) and logical schema (down) of our database implemented in the application

SESSION (Id, Name, Surname, Data)
DATAS(Id, idSession, AccX, AccY, AccZ, GyrX, GyrY, GyrZ, timestamp)
GOALS (Id, idSession, timeExecution, maxAcc)

In flutter, for storing data, are present various package that permit the creation and the manage of database. In our application we used Floor package: Floor is a lightweight and SQL-centric database abstraction library for Flutter applications, inspired by the Room persistence library. It provides automatic mapping between in-memory objects and database rows, while still allowing full control of the database through SQL. To fully utilize Floor's capabilities, understanding of SQL and SQLite is necessary. Additionally, Floor is null-safe and type-safe, making it reliable and robust. It is also reactive, allowing for quick updates to the UI based on database changes and compatible with iOS, Android, Linux, macOS, and Windows operating systems.[L]

In our application, there are some queries that allow us to easily analyze and manage our database. For example, when we are in the DatabasePage and we want to know the saved data related to a specific patient ID, we will use the following queries:

```
SELECT *
FROM Goals
WHERE idSession= (ID)

SELECT *
FROM Data
WHERE idSession= (ID)
```

Figure 61 how is structured in SQL two of the queries written inside the code of the application

By using these two queries, we can retrieve the execution time and maximum acceleration of the patient with ID from the Goals table, and the sensor data with their corresponding timestamps from the Datas table obtaining the representation showed in *Figure* above.

Once the data are stored we can save the database with the download button at the bottom. When we click the button our database will be saved as .txt in our download folder. Once saved on our smartphone we can import it on excel or another application to other implementation or analysis.

In the next chapter, we will analyze the data collected from the application and compare it with values found in the literature. Additionally, we will discuss the population examined and the results obtained.

5. Results

In this chapter, we will discuss the results obtained from our experiment. Firstly, we will talk about the populations examined in the literature and our experiment. Then we will discuss the data collection on individual patients and finally, we will compare our results with the literature.

5.1 Population description and data acquisition

According to literature, the fundamental parameters to analyze in a population for a Sit to Stand task may include:

- Time: the time taken to complete the movement from sitting to standing position. It is a fundamental parameter as it reflects the effectiveness of the movement and the patient's ability to perform it correctly.
- Acceleration: the maximum acceleration reached during the movement, which can be calculated through the analysis of sensor data. This parameter is important as it reflects the strength and power required to perform the movement and can be used to assess muscle functionality.
- Anthropometric variables: these include patient's height, weight, age, and gender. These variables may affect the patient's ability to perform the movement correctly and can be used to differentiate performances between different subgroups of the population.

In fact, Cinzia et al [1] and Hellect et al. [13] used variables such as sex, height, and weight to analyze the variability of data. Many others have used these parameters as a foundation for analysis, which could lead to more significant hypotheses, such as the difference in sex could highlight a slight difference in acceleration of movement, or the difference in height would provide more sense to the correlation between time and calculated acceleration. Weight would also influence the analysis of movement, as patients with greater mass may require more force and therefore more acceleration during movement.

Several studies have been conducted on the assessment of postural control using wearable sensors in healthy individuals and patients with neurological disorders.
From the literature, we can say that the populations analyzed vary considerably depending also on the type of task, research question, and target population. For example, Amici et al. used 33 patients, including 20 healthy subjects between 20 and 26 years old [1], Veltnik et al. used 10 male healthy patients between 23 and 42 years old [20], Mazzà et al. used 22 healthy subjects

between 19 and 25 years old [3], and in another paper Mazzà et al. used 29 patients between 57 and 72 years old. [16] Najafi et al. analyze 11 subject between 73 and 85. [11]
The majority of literature analyzed presents research on fall prevention in elderly patients and pathological patients such as those with Parkinson's disease or post stroke.
Considering the aim of the thesis, the literature and population available, our experiment will analyze 16 healthy students, 9 females and 7 males, between the ages of 20 and 30 that are more similar to the control group of Amici et al. [1].

To collect data on individual patients, we used the MUSE sensor to measure acceleration movement during the Sit to Stand tasks and a strap that allows us to keep the sensor still in position. The data collected were then processed and analyzed using our application, which allowed us to obtain relevant parameters such as execution time and maximum acceleration.
To each patient, as described in chapter 3, are asked to sit in the chair with their arms crossed over their chest to prevent them from being used to aid in movement, and once the data acquisition is started, the patient is told that they can perform the task at their natural speed whenever they want.

This is the collected data:

Name	**Surname**	**Date**	**Time[s]**	**Acceleration[m/s^2]**
Paziente	0	08/03/2023	1,61	1,65
Paziente	1	08/03/2023	1,58	1,38
Paziente	2	08/03/2023	1,58	0,17
Paziente	3	08/03/2023	0,99	0,67
Paziente	4	08/03/2023	1,00	1,30
Paziente	5	08/03/2023	1,35	6,99
Paziente	6	08/03/2023	1,40	1,03
Paziente	7	08/03/2023	1,35	2,35
Paziente	8	08/03/2023	1,17	0,48
Paziente	9	08/03/2023	1,76	1,38
Paziente	10	08/03/2023	1,57	0,96
Paziente	11	08/03/2023	1,49	0,38
Paziente	12	08/03/2023	1,26	0,47
Paziente	13	08/03/2023	1,84	0,81
Paziente	14	08/03/2023	1,17	0,84
Paziente	15	08/03/2023	1,02	2,61
MEAN±STD			1,38±0,22	1,47±0,97

Table 9 Data saved from our database in ASiStApp

The most important value that our application aimed to highlight was the task completion time during the Sit to Stand phase only and as can be seen, the mean of 1.38 m/s^2 with a standard deviation of 0.22 m/s^2 can be plausible values.

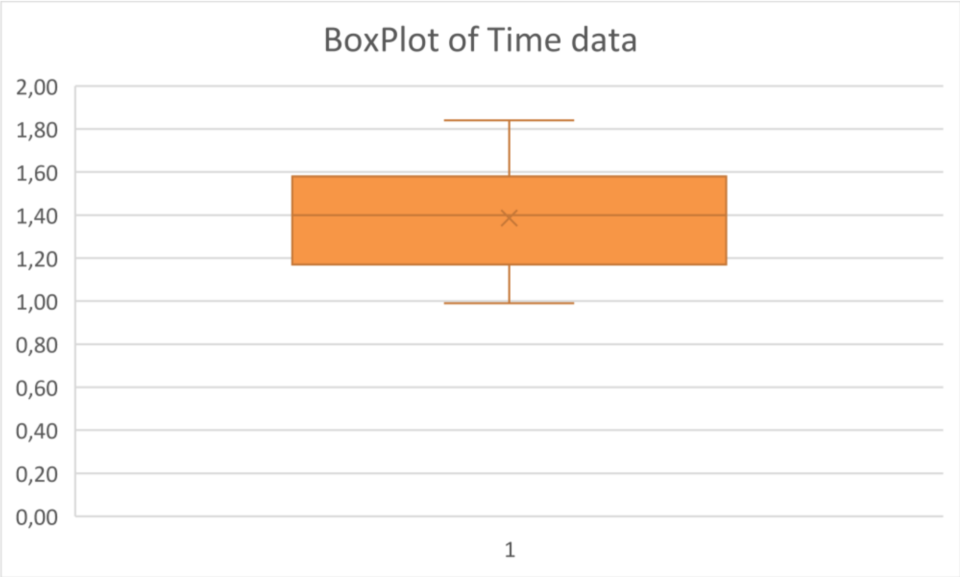


Figure 62

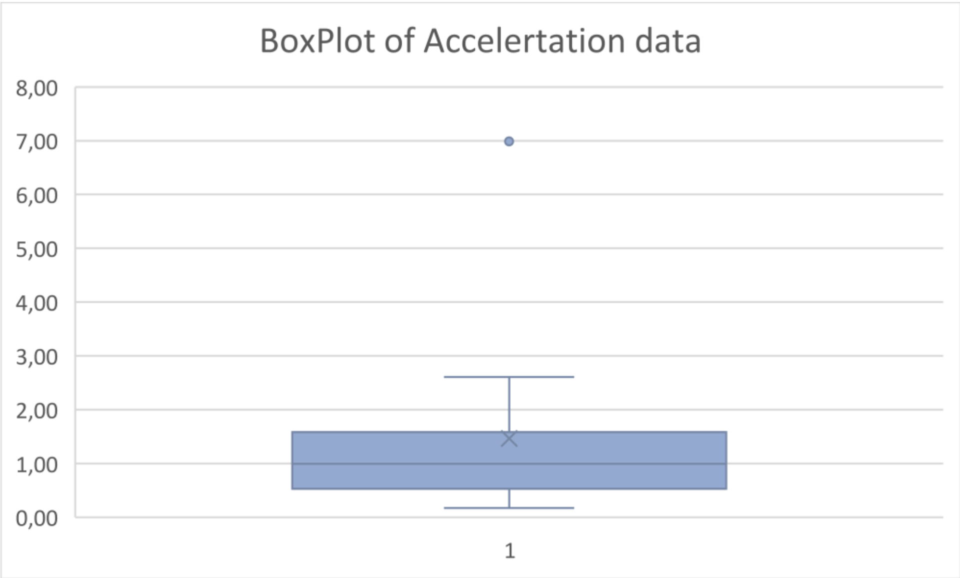


Figure 63

Considering again the table above, unlike the calculation of time which seems to present excellent values that we will discuss below, some issues arise during the calculation of the patient's maximum acceleration.

The reasons can be :

- During the data acquisition, patients 2 and 11 started from a more kyphotic position and they had the need to tilt their torso more compared to the other patients. This could have influenced the calculation of the maximum acceleration as the information regarding the speed of the movement that may have shifted to the other axis (axis Z).
- As described in previous chapters, our sensor is extremely sensitive to vibrations and small variations. This could explain why even patients 5 performed the movement with an extremely high acceleration and subject 2 have such a low acceleration compared to the others.

The main difference between these outlier patients is that the 5th subject did not present any issues during data processing, while the 2nd and 11th were retested until, noticing a straighter starting position, the application processed more suitable data for the task compared to the other and to the literature.

5.2 Comparison with literature results

Mazza et al. [3] state that the time required to perform a sit to stand at a natural speed with feet apart is approximately 1.6 seconds. Najafi et al. [11] reported an average of 2.95 seconds in elderly patients at low risk of falling, while Arash et al. [21] found a time of 2.10 for patients aged between 25 and 35, but the movement considered was a turn and sit that is more complex. In the end Lukas et al. [22] found in young people 1.7s as mean time for a Sit-to-Stand task.
When comparing our results with those found in the literature, we observed similar trends and patterns, more similarity between our data and Mazza and Lukas. [3][22]

DATASET	GROUP 1	GROUP 2
LENGTH	22	16
AVERAGE	1,60	1,38
DEVIATION STANDARD	0,20	0,22
T-STUDENT	3,2104	
P	0,028	

Figure 64 test t-Students between our data and Mazza control group

As you can see in the figure above, we compute a statistic test with our data and the dataset of Mazza[3] and the result is that the difference of the observed means are significant for $p<0,0028$.

In the last chapter, we will discuss about our conclusions regarding the aim of this thesis, evaluating our collected and analyzed data in comparison with those found in the literature. Additionally, we will present future developments of this thesis.

6. Conclusions and future developments.

In this chapter, we will draw conclusions on the study performed and discuss future developments for this application and for this study.

6.1 Summary of the thesis

The aim of this thesis was to create an Android/iOS application that can be used on a smartphone to process data offline and collect it using an IMU sensor connected via BLE protocol, with the goal of automating the current data collection process carried out using a handheld stopwatch, and generating a report on the execution of the Sit to Stand task.
Overall, our results suggest that our application and the MUSE sensor can provide valuable information on time extraction in the analysis of the task. The acceleration data collected can be used to assess postural analysis and to track changes over time, which may be useful in rehabilitation and in the prevention of falls.
In conclusion, we can say that the application created is very simple to use and useful as it allows to find similar timing to those found in literature and permit to the user to visualize a better view of the task.

6.2 Future developments

Future developments of this experiment and application will include:

- For the experiment, multiple sensors can be set up together to perform, for example, a trunk analysis to derive further components on the axis of movement.
- For the application, a Flutter package is already in production that will allow for easier connection and use of the Muse sensor without having to analyze the entire communication protocol of the 221e, simply by implementing functions that will replace these steps.
- Test our application in a population of elderly people or pathological subject for the prevention of falling.

Bibliography

1. C. Amici et al., “Quantitative Kinematic Assessment of the Sit-to-Stand Transition using an IMU Sensor.”, 2021 24th International Conference on Mechatronics Technology (ICMT)| DOI: 10.1109/ICMT53429.2021.9687250 , 2021
2. B.Grimm et al., “Evaluating physical function and activity in the elderly patient using wearable motion sensors”, EOR | volume 1 | DOI: 10.1302/2058-5241.1.160022 , 2016
3. C. Mazza et al., “Sequencing sit-to-stand and upright posture for mobility limitation assessment: determination of the timing of the task phases from force platform data”, www.elsevier.com/locate/gaitpost, 2004
4. D. Rodriguez-Martina et al., “Identification of Sit-to-Stand and Stand-to-Sit Transitions Using a Single Inertial Sensor”, https://www.researchgate.net/publication/230783929, DOI: 10.3233/978-1-61499-069-7-113, 2012
5. R. Soangral et al., “A comparative study for performance evaluation of sit-to-stand task with body worn sensor and existing laboratory methods.”, Biomed Sci Instrum, 2012
6. M. Vettoretti et al., ”Biomedical Wearable Device for Healthcare and Wellbeing”, Academic Materials, A.Y 2021-2022
7. M. Jannes et al., “Algorithms in Digital Healthcare An Interdisciplinary Analysis.” https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/BSt_Algorithms_Healthcare.pdf, 2019
8. K Nwe et al. “Medical Mobile App Classification Using the National Institute for Health and Care Excellence Evidence Standards Framework for Digital Health Technologies: Interrater Reliability Study”. PubMed, DOI: 10.2196/17457, 2020
9. A. Triantafyllidis et al. “Applications of Machine Learning in Real-Life Digital Health Interventions: Review of the Literature” J Med Internet Res 2019;21(4):e12286, URL: https://www.jmir.org/2019/4/e12286, DOI: 10.2196/12286
10. AG. Ekeland et al. “Effectiveness of telemedicine: a systematic review of reviews”. Int J Med Inform. 2010 Nov;79(11):736-71. doi: 10.1016/j.ijmedinf.2010.08.006. PMID: 20884286.
11. Najafi et al, “Measurement of stand–sit and sit-stand transitions using a miniature gyroscope.”, IEEE transactions on biomedical engineering, vol. 49, 2002
12. M. Aloqah et al., A Headband for classifying Human postures, 32nd Annual International Conference of the IEEE EMBS (2010)

13. J. Hellec et al. « Sit-To-Stand Movement Evaluated Using an Inertial Measurement Unit Embedded in Smart Glasses—A Validation Study.", Sensor 2020, vol 20. https://doi.org/10.3390/s20185019, 2020
14. U. Martinez-Hernadez et al, "Probabilistic identification of sit-to-stand and stand-to-sit with a wearable sensor", Elsevier, 2018
15. W. G. M. Janssen et al, "Determinants of the sit-to-stand movement: A review," Phys. Therapy, vol. 82, no. 9, 2002
16. C. Mazzà et al, "Biomechanic modeling of sit-to-stand to upright posture for mobility assessment of persons with chronic stroke." Arch Phys Med Rehabil, 2006
17. J. Wheeler et al. "Rising from a chair: influence of age and chair design." Phys Ther. 1985
18. F. Scarpa, "Bioimmagini", Academic material, A.Y 2020-2021, and G. Sparacino "Analisi dei dati biologici", Academic material, A.Y 2021-2022.
19. 221e, "Muse V2 communication protocol" v1.2.0, 2021
20. P. H. Veltink et al, "Detection of static and dynamic activities using uniaxial accelerometers," IEEE transactions on rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society, vol. 4, 1996.
21. A. Salarian et al, "iTUG, a sensitive and reliable measure of mobility," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 18, 2010.
22. L. Adamowicz et al. "Assessment of Sit-to-Stand Transfers during Daily Life Using an Accelerometer on the Lower Back." Sensors 2020, 20, 6618. https://doi.org/10.3390/s2022661, 2020

Sitography

A. FDA, https://www.fda.gov/medical-devices , U.S. Food and Drug Administration, 25-03-2023
B. 221e, https://www.221e.com/about-us, 221e website, 25-02-2023
C. 221e, https://www.221e.com/muse-miniaturized-multi-sensor-imu, 221e website, 25-02-2023
D. Moodle calvino , https://moodle.calvino.ge.it/mod/resource/view.php?id=3682#:~:text=Il%20filtro%20passa%2Dbasso%20attenua,e%20resistenza%20presenti%20nel%20circuito., 25-02-2023
E. Wikipedia, https://it.wikipedia.org/wiki/Media_mobile, 25-03-2023

F. A. Tedeschi, https://www.html.it/pag/367891/flutter-nuovo-framework-per-lo-sviluppo-cross-platform/. 25-02-2023
G. Pubdev, https://dart.dev/ & https://pub.dev/, 25-02-2023
H. Punchthrough, https://punchthrough.com/android-ble-guide/, 25-02-2023
I. Pubdev, https://pub.dev/packages/flutter_blue_plus, 25-02-2023
J. 221e, https://www.221e.com/wp-content/uploads/2022/11/221e-Whitepaper-An-Introduction-to-Inertial-Sensors.pdf., 25-02-2023
K. Stack Overflow, https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/66700221#66700221, 25-02-2023
L. Pinch , https://pinchbv.github.io/floor/#welcome-to-floor, 25-02-2023

Figure Bibliography

Figure 1 : https://www.researchgate.net/figure/30-Second-Sit-Stand-phases_fig2_282888196.
Figure 2 : https://www.cdc.gov/falls/facts.html.
Figure 3 : https://www.sciencedirect.com/topics/computer-science/optical-motion-capture.
Figure 4 : https://www.designworldonline.com/wearables-dressed-for-success/.
Figure 5 : https://www.semanticscholar.org/paper/The-future-of-MEMS-sensors-in-our-connected-world-Lammel/1047c832359d3fb590234f87843e13ec419d2e3a.
Figure 6: https://www.allaboutcircuits.com/technical-articles/deriving-the-transfer-function-of-a-capacitive-sensing-accelerometer/.
Figure 7 : https://www.analogictips.com/gyroscopes-part-2-optical-and-mems-implementations-faq/.
Figure 8 : https://www.insuranceup.it/it/scenari/digital-health-cos-e-vantaggi-startup-assicurazioni/.
Figure 9: https://auradirect.be/category/mobile-medical-applications/mhealth/.
Figure 10: https://www.redbytes.co.uk/mobile-health-app-usage-statistics/.
Figure 11: https://www.researchgate.net/figure/Smartphone-based-m-health-model-with-AI-and-big-data-analytics_fig4_344057654.
Figure 12: https://www.npr.org/sections/health-shots/2021/10/18/1044358309/patients-say-telehealth-ok-but-doctor-visits-in-person-better.
Figure 13,14,15,16: https://www.221e.com/
Figure 20: https://medium.com/flutter-community/execute-only-dart-code-after-installing-flutter-e05ea9b46627
Figure 21: https://flutterbyexample.com/

Figure 22: https://medium.com/@hrcovertx/understanding-how-flutter-works-fdd54e3ca0d1.

Figure 23 : https://pub.dev/

Figure 24: https://itnext.io/flutter-types-of-widgets-8d457281ad03

Figure 26 : https://www.researchgate.net/figure/Phases-of-the-Sit-to-Stand-movement_fig1_347234572

Figure 30: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/

Figure 34:[K].

Figure 37: https://elainnovation.com/en/bluetooth-vs-bluetooth-low-energy-whats-the-difference/

Figure 38:

https://shop18102.thuocxoang.net/category?name=bluetooth%20low%20energy%20vs%20bluetooth

Figure 41, 42, 43, 44, 45, 46, 47: [19]

Figure 46: https://www.nejm.org/doi/full/10.1056/NEJMvcm2009406

RINGRAZIAMENTI

Per questi ringraziamenti immaginatevi un sottofondo che colpisca, come il cambio di stile nostalgico delle sigle iniziali più conosciute, tipo un “big bang theory” o un “friends”, perché quest’avventura si è conclusa, con un finale come fosse una bellissima serie tv.
Eccovi i titoli di coda:

In Primis voglio ringraziare i miei genitori che durante tutto il percorso della laurea mi hanno sopportato e supportato in tutto e per tutto, grazie di cuore, senza voi non sarebbe stato possibile.

Voglio ringraziare mia sorella Giada che nonostante i grandi cambiamenti che ha fatto nell’ultimo periodo, sia lavorativi che nella vita stessa, è sempre stata presente ed è stata il mio modello di riferimento per tutta la laurea. Grazie.

Un ringraziamento ai miei nonni Gianni e Daniela che mi hanno sempre spronato a seguire i miei sogni e soprattutto a godermi ciò che la vita mi sta offrendo ora, in uno dei periodi forse più belli della mia vita. Grazie.

Sempre presenti, voglio ringraziare la family Pellegrini per essere come una seconda famiglia. In particolare il fratellone Andrea che come sempre può passare una vita tra un ritrovo e l’altro ma è come se ci vedessimo ogni giorno. Vi voglio bene!

Voglio ringraziare l’ingegner Sergio che è stato presente da prima dell’università e come amico e collega. Mi hai sempre mostrato cosa vuol dire forza di volontà e determinazione. Grazie di tutto! Meriti quei 50cent di basilico come in una catena di montaggio.

Voglio ringraziare Sofia per esser stata presente in questi 5 anni di università, sei stata il miglior acquisto di questa avventura! Grazie di tutto. Ti voglio bene Ubuske.

Voglio ringraziare i miei coinquilini: Lorenzia, Nic, Sere, Frah e Prix. Grazie a voi ho concluso nel migliore dei modi il capitolo “Ingegneria UniPd” della mia vita. Ognuno di voi è stato importante a modo suo. Un grazie a Lorenzo per le sue cappellate che ricorderò per sempre. Grazie a Nic, sei una persona meravigliosa e ti auguro il meglio, sei stato di fondamentale importanza, ti devo davvero molto, grazie davvero di cuore, best compagni di stanza ever. Un grazie a Serena, abbiamo legato tantissimo e anche te meriti un sacco di ringraziamenti per tutta la pazienza che hai portato, ascoltandomi e consigliandomi sempre, ti voglio tanto tanto bene. Un grazie a Francesca, la prima persona con cui ho legato in casa, sei di una solarità e simpatia incredibile, grazie di tutto fraaah! In fine Priscila, thanks for everything, you taught me to see many things from a different perspective, you showed me how people can move forward and face everything in life thanks to their experiences. I wish you lots of luck and happiness. Obrigado!

Un ringraziamento speciale va anche ai miei coinquilini della triennale: Marta, Chiara e Tommaso. Siete stati una famiglia per me e ogni volta che vi vedo realizzare un nuovo obiettivo nelle vostre vite mi rende tanto felice. Sono davvero contento e fortunato ad avervi incontrato, vi voglio davvero tanto tanto bene. Buona fortuna per tutto.

Un grazie va al mio carissimo amico Isack, altra colonna portante del mio ultimo anno e non solo. Ormai sono più di 10 anni che ci conosciamo e posso dire di essere davvero felice di averti avuto così vicino, soprattutto quest'ultimo anno, letteralmente a 200m di distanza. Grazie di tutto.

Un ringraziamento ai compagni di corso Alessandro, Marco, Luca e tutta la compagnia di Go-Go-Gadgets per aver reso le lezioni e gli esami più facili! Magari un giorno lavoreremo insieme, ma fino ad allora buona fortuna a tutti!

Voglio ringraziare anche i nuovi acquisti di quest'ultimo anno a Padova, come Alessio e Vincenzo detto Cumpà, mi sono divertito tantissimo con voi e ho un sacco di bellissimi ricordi che porterò con me come il FluoParty. Vi voglio bene!

Un grazie va anche a due altri bellissimi acquisti di quest'anno: Virginia ed Antonella. Virgi sei una forza della natura, eclettica per definizione, metti una passione in ogni cosa che fai. Mi hai sempre tirato su il morale anche quando non c'era bisogno, sei fantastica! Anto, detta anche squala, grazie per tutte le uscite e le chiacchierate fatte, tu e Mirko siete una coppia meravigliosa e capace di andare avanti nonostante tutto e tutti. Mi avete mostrato che c'è ancora del buono nelle persone. Grazie di cuore, auguro a tutti le migliori cose.

Un ringraziamento va ad Alen e alla pizzeria “A tutta Pizza” che mi ha permesso di aiutare economicamente la famiglia. Grazie

Un ringraziamento va anche alla compagnia dei colleghi di “A tutta Pizza” come Paolo, Marcello, Alissa, Eleonora, Melanie, Pi(e)ro, Patrik, Francesco ecc. è stato bello lavorare con voi!

Infine voglio ringraziare il mio relatore Giacomo Cappon e la dottoranda Elena Pegolo per l’incredibile pazienza portata durante tutto il periodo della tesi. Grazie.