UNIVERSITY OF THESSALY

DEPARTMENT OF MECHANICAL ENGINEERING

Diploma thesis

COMPARISON OF THREE POSITION MEASUREMENT METHODS AND THEIR ACCURACY: MEMS accelerometers, Ultrasound-based transmitters-receivers and Optical motion capture.

By Alexandra Papachatzopoulou

Supervisor: *Prof. Alexandridis A.*

Submitted for the purpose of fulfilling the requirements

for a master's degree in Mechanical Engineering

2018

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Alexandros Alexandridis for his help throughout the course of this study, as well as the other members of the examination committee, Prof. Kostas Papadimitriou and Prof. Alexis Kermanidis for the time they dedicated to examine my thesis and provide valuable comments. I am also immensely grateful to my professors and colleagues at the Budapest University of Technology and Economics; to Prof. Rita M.Kiss, for providing me with an interesting and challenging topic for my thesis; to Mr. Gergerly Nagymáté and Mr. Kristóf Rácz for teaching me how to use the equipment and software at the motion capture laboratory of the Opticks an Mechatronics department and for helping me overcome technical obstacles; and above all, to Mr. Bálint Petró, for his invaluable help and mentorship over the course of this study and for providing me with knowledge, sources and tools vital for the completion of this master thesis. Finally, I would like to address a massive thank you to my friends and family, for their moral support, love and encouragement throughout the years and the last few months in particular. I would like to dedicate this master thesis to Mr. Bálint Petró.

Thesis Committee

Prof. Alexandridis A.

Prof. Kermanidis A.

Prof. Papadimitriou K.

Important terms

Rigid-body: a solid body with zero deformation
Anatomical landmarks: a morphologic feature of the anatomy that is readily recognisable and may be used as a reference point for other body features
DAQ: data acquisition
COP: center of pressure
Mocap: motion capture

Abstract

Introduction and objective: Position measurements are applied in a plethora of scientific areas such as the automotive industry, robotics, medical imaging technology, biomechanics and biomedical engineering. Some of the methods used for position measurements include *optical motion capture, ultrasonic transmitter-receiver systems* and *MEM senseors.* The aim of this thesis was to compare this methods based on their accuracy, user-friendliness, versatility and cost. *Materials and methods:* Three devices were tested for this study: The *Optitrack* mocap system, *Zebris CMS10* (ultrasound transmitters-receivers system) and an *MPU-9150* (MEMS accelerometer and gyroscope). Measurements conducted by all three devices calculated the 2D position (x-y level) of a Posturomed platfrom, performing free oscillation tests. Statistical tests were conducted with the aim of comparing the devices' measurement accuracies. *Results:* One-sample Wilcoxon signed rank tests were used. The p-statistic was calculated to be lower than 0.05 (5% significance level) in all conducted tests. *Discussion and conclusions:* Statistical tests showed that the measurement accuracies of the MPU-9150 accelerometer and the Zebris CMS10 were significantly different from that of the method of reference, Optitrack. Accuracy of the study's results could have been affected by the type of statistical tests (signrank tests) that were employed or the double integration and filtering of acceleration data.As far as user-friendliness is concerned, the Zebris CMS10 and the MPU accelerometer are both compact, postable, easy to set-up devices with intuitive softwares. For Optitrack, its online documentation provides users with valuable tutorials and instructions on installation, set-up and mocap tests. In terms of versatility, Optitrack and the MPU are the two devices employed in a plethora of applications and disciplines, while Zebris CMS10 is mostly limmited to human movement analysis studies. The MPU device is the most inexpensive one by a large margin.

Keywords

Position measurements
Motion analysis
MEMS accelerometer
Posturomed
Displacement data
Statistical tests
Accuracy

Contents

Index of figures

Index of tables

1. Introduction

1.1 Position measurement applications

Over the last few years, position measuremets have been proving to be extremely useful and practical in a vast area of applications, covering a variety of scientific fields and principles, including navigation, transportation, security, medicine and many more.

One of the most useful and well-known applications of position measurements is the *Global Positioning System*, otherwise known as *GPS*. The basic operational principle of the GPS is the calculation of the position of an object on or close to the surface of the earth based on its distance from a satellite. GPS systems are applied on a huge variety of areas, from civilian to military intelligence applications. To summarize them, the main categories that GPS applications fall in are 1) location determination, 2) timing, 3) nagivation, 4) mapping and 5) tracking [10].

In the *automotive industry*, position measurements are beginning to play a very important role in the development of new technologies, meant to transform driving into an easier and safer experience. Speed measurements, collision avoidance systems, obstacle detection, self-parking and self-driving cars are all cutting-edge applications whose operation is based on position measurement systems [3]. These systems are not only facilitating transportation and driving themselves, but also enabling engineers to contibute to accident prevention and, subsequently, to saving human lives.

On the matter of saving lives, position measuring systems have been one of the most highly effective tools used in *medical imaging* technlogies. *Medical ultrasounds* allows doctors to detect fetuses' positions in pregnant women, organ and tissue movement as well as internal abnormalities, such as tumors. *Biopsies* as well as *computer aided surgery* technologies are also heavily based on calculating the position of organs, vessels or tissues or the position of the medical probes used in these operations [11]. *Capsule endoscopes*, used for diagnosis of deseases of the digestive organs as well as a number of other clinical applications, can be monitored by measuring the capsule position inside the human body [12]. All these applications and technologies have been contributing to raising the effectiveness of clinical operations while simultaneously minimizing the risk and have provided doctors with more accurate diagnostic and patient monitoring tools.

In the field of *biomechanics*, position measurement systems have also been proving to be fantastic tools for diagnosis, patient monitoring and rehabilitation procedures. In *gait analysis*, using position measuring systems allows us to observe the dynamics of human walking and detect any malfunctions of abnormalities. *Joint position tracking* provides us with a new medium for monitoring the displacement of knee of hip joints after joint replacement surgery, facilitating the rehabilitation process. Body or body segment postition measurement throught *motion capture* makes motion tracking and 3D body representation in real-time possible, thus enable human motion analysis. The calculation of head rotation, body orientation or center of pressure (COP) are key elements to studying and monitoring human balance before or after prostheses.

The applications of position measurement by no means end here. Other fields and disciplines where they are used include security systems, robot monitoring and localisation, object dimentioning and many more; the list goes on and on. However, this master thesis is focused on

the field of biomechanics and, more specifically, on biomedical applications.

A variety of scientific methods have been used for position measurements in biomechanics. Naturally, differences exists between these methods as far as efficiency, accuracy, cost and user-friendliness are concerned. In the next section of this master thesis, some of the most relevant previous findings on the various methods for position tracking and measurement in biomechanical applications are presented, as well as their previously mentioned characteristics.

1.2 Previous findings

The current most used clinical solution for motion sensing and position determination is *optical motion capture (mocap).* In motion capture systems, body or object motion and position derives from multiple markers, attached to the body or object under observation. This method is highly accurate and effective, it is, however, obtrusive and rather expensive [7]. Vixon (Oxford metrics, UK) is an example of a high-end motion capture system used specifically for scientific purposes, while OptiTrack (NaturalPoint, Corvallis, OR, USA) is a newer system, which was first used for animation purposes but has recently started being widely used in motion labs thanks to its smaller cost, compared to Vixon, but equally high accuracy [1]. Despite their advantages, motion capture systems can cost thousands of dollars, which is the reason why their use is limitted to the financial abilities of each institution or laboratory. This necessitates the search for alternative, more economic but equally accurate and practical methods for position measurements.

A more low-cost but also highly efficient method involves the use of *ultrasonic receiver-transmitter systems.* There are a few different techniques for distance (or position) measurements using ultrasound systems: 1) the time-of-flight (TOF), 2) the binary frequency-shift-keying (BFSK), 3) the binary amplitude-shift-keyed (BASK) and 4) the pulse compression technique. The BFSK method is more accurate than the TOF method, but it's difficult for both of them to accurately measure the time the sound-wave signal takes to travel between the transmitter and the receiver. In order to reduce distance measurement errors, a phase-shift analysis of signal-frequency continuous-wave transmission is used in most scientific applications. The disadvantage of using ultrasonic transmitter-receiver systems is that, usually, a compromise has to be made between operating range and measurement accuracy [13]. There is a large number of companies that design and develop ultrasonic sensor systems for position measurement as well as other purposes. Keyence is an American company that produces high accuracy 3D displacement sensors for a range of applications, from automotive to pharmaceutical. Baumer, Pepperl+Fuchs and SparkFun Electronics are but a few more companies in the same category. In the field of biomechanics, Zebris is a Germany-based company that provides ultrasonic transmitter-receiver systems specifically designed for biomedical applications.

Micro-electro-mechanical-systems (MEMS) are an innovative tool that is becoming more and more popular in position measurements due to their high accuracy, reliability and flexible and variable functionalities [7]. The small scale of these motion sensing technologies has made them extemelly practical for human balance prostheses devices (calculation of head rotation or body orientation), as well as applications that include radiotherapy (calculation of respiratory waveform), sports medicine (heart rate analysis) and biomedical research [7]. They are portable, disposable and wearable devices, characteristics which contribute to their popularity for healthcare applications, such as physical activities' sensing, heart-rate, breath and blood-pressure monitoring as well as for diagnostic methods for specific medical conditions, such as neurophysical disorders [14]. Furthermore, MEMS systems are considerably more inexpensive compared to motion capture or

ultrasounds transmitter-receiver systems, which offers them an additional important advantage. *MEMS accelerometers* are proving to be a valuable tool in mechatronics and low-invasive human motion tracking, such as body segment motion tracking [27] as well as kinetic desease prognosis [28]. Attached to a moving rigid body, these accelerometers can provide accurate measurements of displacement and orientation without the need for any fixtures or physical reference bases [17]. Despite their many advantages, MEMS sensors are susceptible to measurement errors that derive from a variety of factors, such as noise, offset and drift. For this reason, the limitation or elimination of these error sources is vital for the effective use of MEMS systems [16]. Nevertheless, previous research findings have demonstrated that measurement errors in MEMS accelerometer applications can be as low as 1nm over a 4 second period [17].

1.3 Research topic and its importance

In the field of biomechanics and biomedical engineering, body position measurements are of crucial importance for motion analysis. *Center of gravity (COG) tracking* is a valuable tool for postural or dynamic balance and gait analysis, in determining the displacement of the body's center of gravity during static or dynamic motions and, as a result, detect possible abnormalities. In neurology, measurement of *head position and postural alignement* can enable doctors to identify, and therefore possibly treat, neurological disorders [20]. *Sensor-studded monitoring systems* use position data for fall or motion detection of patients or the elderly, both in hospitals and homes, contributing to an increase in their safety during rehabilitation periods or every-day life. *Hip or knee joint position tracking* allows doctors and physiotherapists to monitor the rehabilitation process of patients after hip or knee replacement surgeries.

Position measurements and their applications are an incredible tool for engineers and medical professionals that provides them with the chance to study human movement in depth. As was mentioned previously, the results they extract from these measurements enable them to diagnose abnormalities in the normal physiological function of patients, apply more efficient treatment methods and contribute to a safer and more controlled rehabilitation process. Adressing these issues requires great precision and caution as they have to do directly with human health and safety. Thus, there is a necessity for highly accurate methods for position data acquisition so as to minimize the measurements' errors and, consequently, be able to apply the results efficiently and without risking human health and well-being.

In section 1.2, the most important and widespread methods for position measurements and data acquisition were presented. Their efficiency and suitability depends on the type of biomedical application they are being used for, the available equipment of the experimental laboratory, the proficiency and scientific capabilities of the user etc. Optical motion capture and ultrasound transmitter-receiver systems are the two methods included in most of the studies and experimental procedures on human motion analysis, while MEMS accelerometers have only recently started getting popular in biomedical applications for position measurements. These three methods are quite different, not only in their accuracy but also in their cost, user-friendliness and versatility.

Unfortunately, there is limited literature on quantitative or qualitative comparisons of these three different methods. In the matter of position measurements, performing a study that would compare and contrast three of the most popular methods could provide the necessary information and tools to choose the one that would offer the best balance between accuracy, cost, and efficiency, according to each individual biomedical application and that would thus ensure the best

possible application of the extracted results on rehabilitation, training and diagnosis problems.

1.4 Description of measurement methods

For the purpose of this master thesis, three different types of position measurement methods are going to be used throughout the experimental procedures. Position measurements will take place: a) through optical motion capture using the **Optitrack** system, b) with the use of **Zebris CMS10**, an ultrasound-based transmitters-receivers system and c) with the use of a 6-axis MEMS accelerometer (and gyroscope), the **MPU-9150**. The Optitrack and Zebris systems are supported by accompanying factory software. For the accelerometer, custom software was built to enable acceleration data acquisition from the performed measurements.

What follows is a detailed description of the operational principles of each of the three different methods as well as a demonstration of how these methods can be used for position measurement purposes.

1.4.1 Optitrack

Motion capture systems have been widely used over the past few years in a variety of different fields such as biomechanics, sports sciences and entertainment [1]. One such system is Optitrack, developed by the company NaturalPoint (© 2017 NaturalPoint, Inc.), that designs and develops software and hardware for motion tracking purposes.

Optitrack's current applications include robotics, movement sciences, virtual reality as well as animation [2]. In the field of biomechanics, the use of motion capture systems such as Optitrack, enables engineers to study human motion and perform movement analysis, through the acquisition of high-precision motion capture data. More specifically, some examples for the use of Optitrack in biomedical or biomechanical applicatios include gait analysis, which is the analysis of human walking, EMGs, centre of pressure (COP) or center of gravity (COG) tracking, measurement of forces and torqes of joints during human motion, as well as, what our area of interest is in this master thesis, measurement of velocity or position during a human or rigid-body motion.

In the next section, the basic elements and principles of the Optitrack hardware and software will be presented.

Hardware

Motion detection is acheived through a set of synchronised infra-red cameras (Figure 1.1) combined with a biomechanic markerset. NaturalPoint offers a selection of several camera types with varying motion capture capabilities, as well as instructions on how to assemble one's own Optitrack set-up, depending on the field where motion tracking is used. Cameras are arranged in the periphery of the desired *capture volume* according to the requirements of the specific motion capture application one wishes to perform. The markerset consists of an array of small spheres with reflective surfaces (Figure 1.3), which are mounted on the object or subject one wishes to monitor on key points or, in case of human subjects, on *anatomical landmarks* (Figure 1.2).

Figure 1.1. Camera set-up for Optitrack

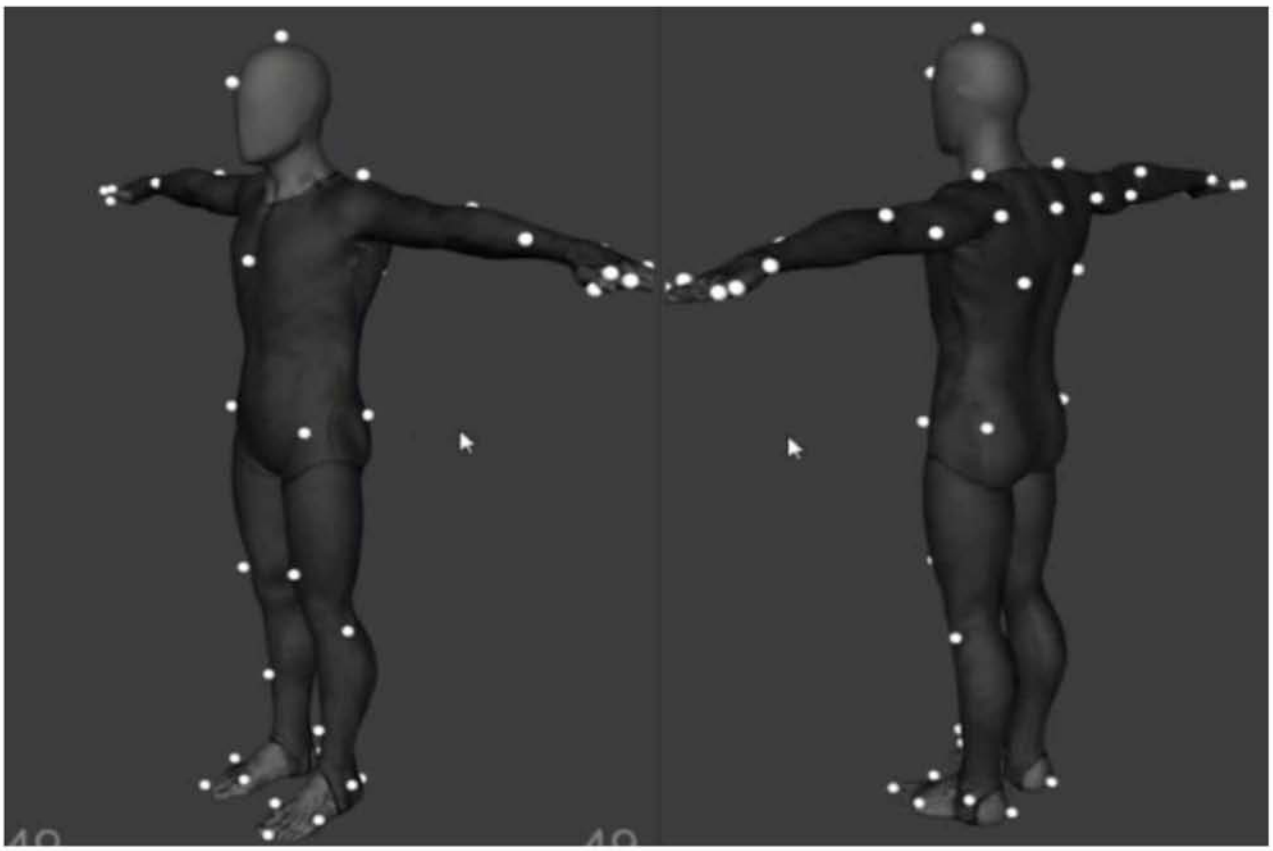

Figure 1.2. Marker placement on anatomical landmarks.

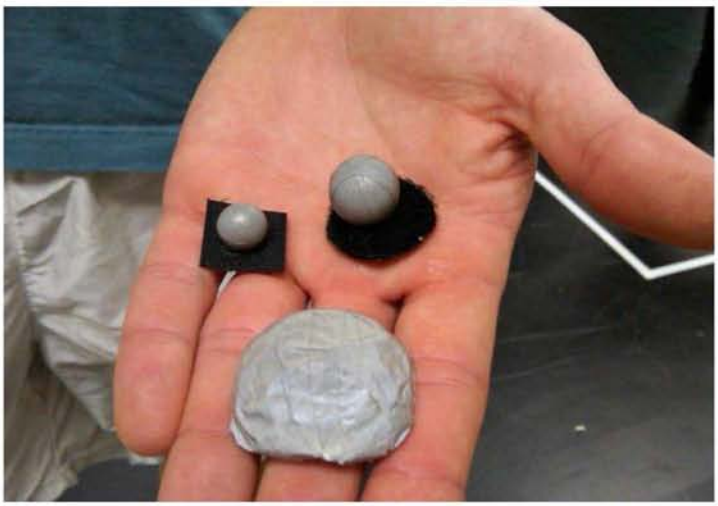

Figure 1.3. Reflective spheres (markers)

The basic principle under Optitrack's operation is the emission and reflection of infra-red light. Infra-red light emmitted by the cameras is reflected back to them by a number of markers mounted on the object or subject the movement of which is being detected. This enables the calculation of the distance between the cameras, the transmitter, and the moving object or subject, the receiver.

In order for Optitrack cameras to capture human or rigid-body motions, at least three reftective markers should be visible by the camera set. Every camera captures 2D images of the moving target, from which the 2D and 3D positions of the moving body can be calculated. The software is what allows users to visualise and record the 3D motion of the target in the capture volume, as well as extract motion data such as centre of gravity displacement, velocity etc.

Software

Motive is the software used for the collection and processing of motion capture data obtained by Optitrack's camera set. Through the 2D position data collected by the cameras, Motive can calculate 3D position and orientation data for moving skeletons or rigid bodies. Motive also enables livestreaming of the captured data to other computers or softwares, thus allowing position, velocity or acceleration data acqusition.

When light is reflected off the markers back to the camera set, Motive can calculate the position of the reflective markers that are inside the capture volume. Once a marker is detected, Motive creates a visual representation of it on the software's user interface. By selecting the desired markers on the screen, the user can create either a *skeleto*n (Figure 1.4) or a *rigid-body*, depending on whether they are tracking a human or an object.

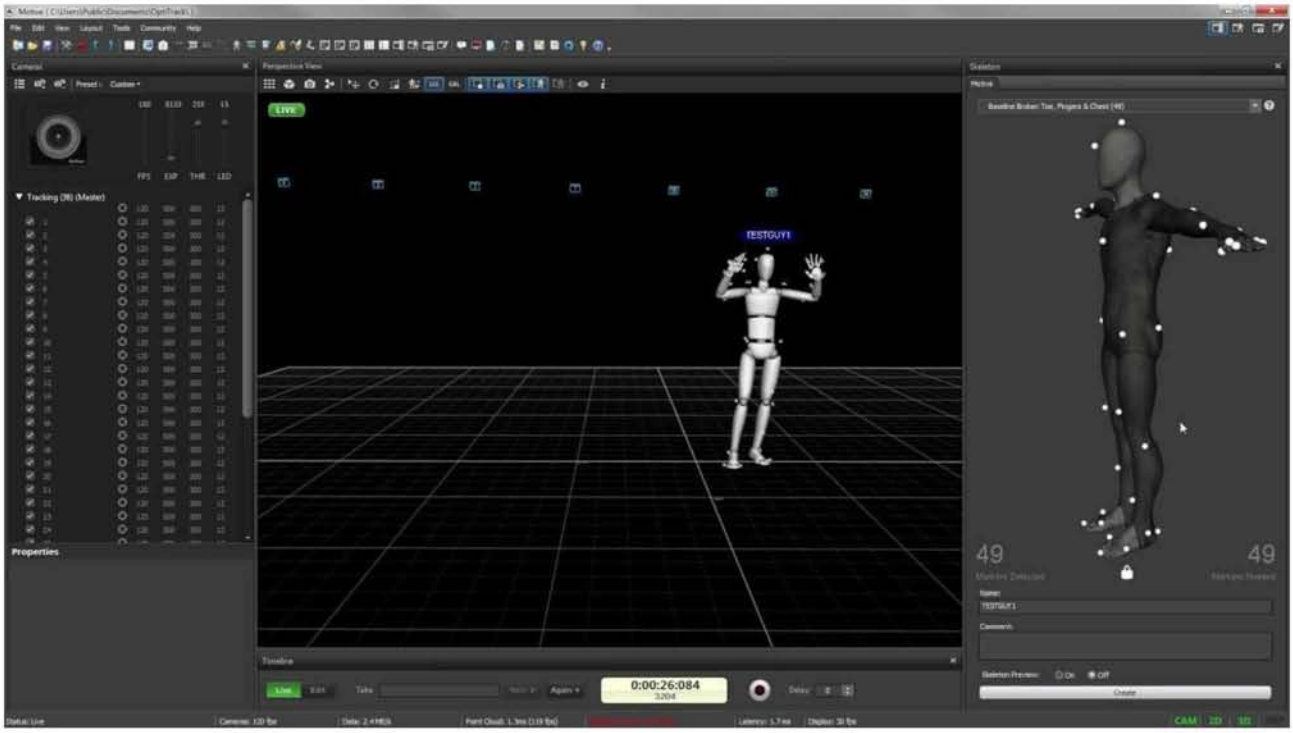

Figure 1.4. Motive skeleton for human mocap

In human motion tracking, when the reflective markers are placed on the correct anatomical landmarks, Motive performs a 3D representation of the person's skeletal system in the form of a virtual avatar, which allows the visualisation of the human motion on the computer screen. Similarly, one can track the movement of an object inside the capture volume by mounting a set of markers on it and, afterwards, creating a rigid-body representation of that object on Motive.

Position measurements

In order to measure the position of the moving target, motion data acqusition is necessary. Unfortunately, Motive can only record visual data, which can then be played back and edited, such as 2D and 3D object images and marker data, but not data such as displacement or velocity. It offers, however, the possibility to livestream motion data to other softwares designed for data acquisition that can record and save motion data in real-time. This feature allows the extraction of position, velocity or acceleration data from an object's or subjects' movement and, therefore, enabels further processing of these results for the desired purposes.

1.4.2 Ultrasound transmitters-receivers systems

Nowadays, ultrasonic sensors are used for a plethora of technological applications that include vehicle detection and parking, level measuring, medical diagnosis ultrasounds, as well as for non-contact velocity and distance measurements. By using unltrasonic sensors, position measurements can be achieved simply through the calculation of the distance between a transmitter and a receiver. The basic principle begind this method and technology is the measurement of the time of flight (TOF), which is, essentially, the time needed for an ultrasonic sound wave to travel from the transmitter to the receiver [3].

An short ultrasonic sound wave (pulse) is sent out by the receiver and, simultaneously, an electrical signal is sent from the power source to the receiver that tells it to start the time count. The pulse is reflected by the detected surface and when it is sent back to the receiver, the time count stops. This counts time from the moment the sound wave left the receiver till the moment it returns to it (time of flight).

The ultrasonic pulse travels through air at a specific and approximately constant speed. By multiplying this speed by the time of flight, one can easily calculate the distance travelled by the pulse or, in other words, the distance between the transmitter and the receiver, which is the detected object.

1.4.2.1. Zebris CMS10

In the field of biomechanics, ultrasound-based transmitters-receivers systems are one of the technologies used for analyzing human or rigid-body motion. The Zebris CMS10 (Figure 1.5) is a compact system specifically designed for the purposes of 3D motion analysis, developed by the company Zebris (zebris Medical GmbH) [4]. Its measuring distance reaches up to 1.5 meters while its measuring rate up to 200 Hz.

Hardware

The Zebris hardware consists of a) a portable device (receiver) of adjustable height, equipped with a mobile floor stand with two joints that carries a sound signal reception panel with three built in microphones, and b) a set of specialized miniature markers (transmitters), mounted on the object or subject under observation.

Figure 1.5. Zebris CMS10

The system works under the same principle that other ultrasound transmitter-receiver systems work, as we described in the previous section. Short ultrasonic pulses are emitted by the markers and are received by the three microphones on the receiver's panel [6]. The time of flight is the time this signal takes to reach the receivers from the time it was emitted. Thus, the CMS10 is able

to calculate the distance between the moving object (or subject) and the receiver and provide us with a real-time 3D representation of the movement.

The system provides 6 marker channels, which means that up to 6 markers can be directly connected to the device [6]. Furthermore, the CMS10 has to be connected to a power source in order for it to operate, as well as to a computer so as to enable motion data acquisition. Connection to the computer and motion data transfer is done via the digital input channel, an integrated USB interface [6].

Software

Depending on the measurement or application one wishes to implement, there is a wide range of specialized software developed and provided by Zebris, such as WinArm, WinSpine and WinPosture. For the purpose of this master thesis, WinPosture is the software that was used during the experimental procedures. One can observe that the software names refer to specific human body parts or functions. The reason for this is that they were developed mainly for biomedical applications and measurements of human motion or, more specifically, 3D motion analysis of specific body parts or the entire human posture. However, these softwares can also be used to acquire motion data from rigid-body movements.

WinPosture is a static and dynamic stabilography software; using the time-of-flight method, the measurement software can calculate the spacial coordinates of the detected markers (transmitters). Posturographic data is acquired via the USB interface and displayed in real time on the computer screen as a set of graphs of position data on the X, Y and Z axis, as well as the XY level. For every individual measurement, motion data can be saved in the form of a text file for later processing and assessment.

1.4.3 MEMS MPU-9150 accelerometer

Micro-electro-mechanical systems (MEMS) combine mechanical and electrical components into small structures in the micrometer scale [8]. Every MEMS system is a combination of mechanical elements, a sensing mechanism, as well as a micro-controller. MEMS sensors are being used more and more for the design and development of a number of devices and applications, such as game consoles, smartphones and tablets, car crash airbag sensors etc.

Before explaining the operational principals of MEMS accelerometers, the difference between accelerometers and gyroscopes must be clarified. Accelerometers measure linear acceleration along one or more axis, whilst gyroscopes measure angular velocity [8]. By combining the use of an accelerometer and a gyroscope, we can obtain information both on the acceleration (or position) and the orientation of a moving object.

Capacitive accelerometers

The most common type of MEMS accelerometers are the capacitive type. The interior of an accelerometer is not a solid structure, but a multilayered one that consists of a moveable mass, that is connected to a mechanical spring system (frame) and is suspended between two fixed electrodes. In accelerometers with a capacitance sensing mechanism, a change in acceleration is related to a change in capacitance between the moving mass and the fixed electrodes. By arranging multiple such accelerometers in different angles, one can get a multiple-axis

accelerometer configuration.
An acceleration applied to the device causes a small displacement on the moveable mass. As a result, the distance between the mass and the static electrode plates changes, which subsequently causes a change in capacitance between the moving and the stable part of the device. It is proved that the distance of the moving mass from the two neighboring electrodes is a function of the produced capacitance upon acceleration [9].

Because the displacement of the mass is of a really small scale (micrometers) the change in capacitance it produces is also too small to be properly detected and calculated. This necessitates a configuration of multiple electrodes-mass layers connected parallel to each other so that the signal of change in capacitance is amplified and more effectively and accurately detected.

Upon acceleration, an analogue voltage of the moveable mass is produced which, after going through a series of processing steps (charge amplification, signal conditioning, demodulation, and lowpass filtering), gets converted into a digital voltage signal. With the use of the appropriate software, this digital voltage signal can be transformed to an acceleration signal, which allows for calculation of the acceleration of the object or subject on which the accelerometer is mounted.

Position measurements

MEMS sensing systems are being used more and more in body motion sensing and analysis applications thanks to their small cost, high accuracy and reliability [7]. Connecting an accelerometer configuration to a data acquisition device or software, allows for collection of not only acceleration, but also orientation, velocity and position data.

As was mentioned previously, the acceleration of a moving human or rigid body can be calculated by transforming the digital voltage signal to acceleration data. If the function of acceleration is known the function of velocity can be obtained by integrating once. Thus, one can obtain the function for position by single integration of the velocity function, or double integration of the acceleration function.

Hardware

For the conduction of the position measurements, an *MPU-9150 accelerometer* (Figure 1.6) and an *Arduino UNO* (Figure 1.7) board were used.

The MPU-9150 is a nine degrees of freedom (9DOF) inertial measurement unit (IMU) in a single package. It houses a 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer and a Digital Motion Processor™ (DMP™) hardware accelerator engine. The range of each sensor is configurable: the accelerometer's scale can be set to ±2g, ±4g, ±6g, ±8g, or ±16g, the gyroscope supports ±250, ±500, and ±2000 °/s, and the magnetometer has full-scale range of ±1200μT (±12 gauss).

Figure 1.6. MPU-9150

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. Connecting the board to a computer with a USB cable or to a battery with an AC-DC adapter is all it takes to support the microcontroller. Arduino boards are able to read inputs and turn them into outputs.The board can be programmed by sending a set of instructions to the microcontroller on the board. To do so, one has to use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

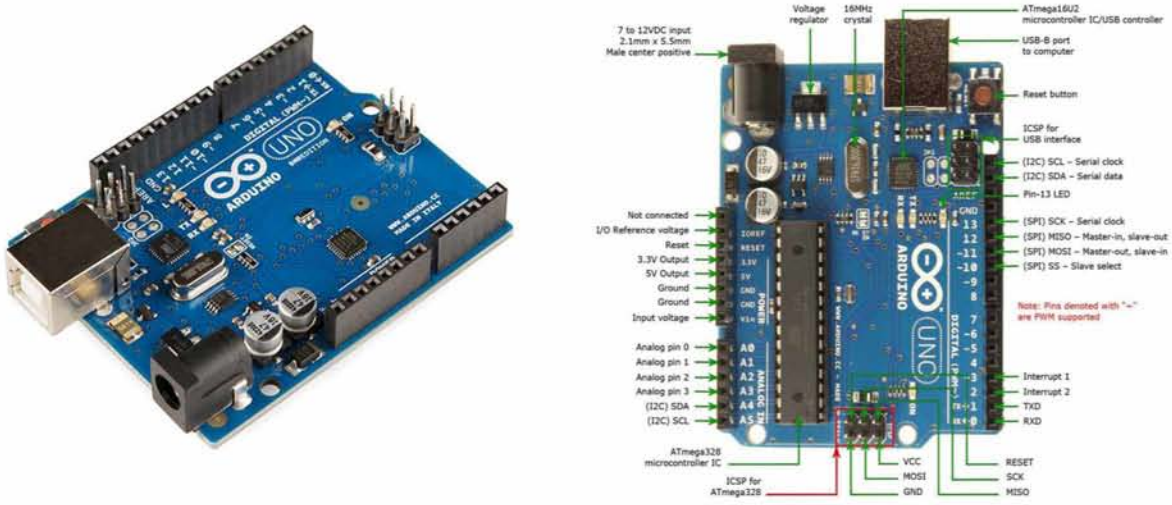

Figure 1.7 Arduino UNO

Software

The software used for acceleration data acquisition was IDE, the open-source Arduino software. This software allows the user to write codes (or *sketches*) and upload them to the board. Instructions on how to program using the Arduino software can be found on the official website. Furthermore, there is a wide range of tutorials on how to write sketches as well as a vast number of tutorials, user-made and tested codes and projects, all of which can be used as a base for constructing a suitable code for the desired application.

1.5 Aim of the thesis

The purpose of this master thesis is to compare position measurement methods and their accuracy as well as a number of other factors such as their versatility, user-friendliness and cost. More specifically, the objective is to investigate whether optical motion capture systems could be replaced by ultrasonic transmitter-receiver systems or MEMS accelerometers for the same position measurement applications, without affecting the measurements' accuracy. At the same time, devices' technical characteristics as well as advantages and disadvantages are offered as a catalogue to readers or researchers with the aim of enabling them to choose an optimal method for their particular position measurement applications or studies.

2. Materials and Methods

2.1 Experimental process

Position measurement tests were performed in the Motion Capture laboratory of the University of Technology and Economics of Budapest, in the MOGI department. For the experimental procedures, a free oscillating platform was used (Posturomed©), as well as a combination of three different position measurement methods: a motion capture system along, an ultrasound-based transmitter-receiver system, and a 6-axis MEMS accelerometer (and gyroscope). For the motion capture, the Optitrack system was used, along with its available factory software, Motive. The Zebris CMS10 system was used for the ultrasound type, with WinPosture, which is part of its factory software. Finally, an MPU-9150 (MEMS) accelerometer was also used, along with a user-made Arduino sketch on Arduino's software (IDE).

Before describing the experimental procedures, it is necessary to explain the functionality and operational principles of the Posturomed© (Haider Bioswing, Weiden, Germany). It is a free-oscillating platform that operates after sudden perturbation [18]. Essentially a neuro-orthopedic device, the Posturomed© (Figure 2.1) is mainly used as a tool in physiotherapy for therapeutic and rehabilitation purposes, as well as for the training of athletes with the aim of improving their balancing skills.

Figure 2.1. The Posturomed platform

The operational principles of the Posturomed© are quite simple; the platform is mounted on a set of steel springs, which enable it to perform a damped free oscillation along the horizontal plane. In order for the oscillation to take place, the platform is first locked outside its resting point by a fastening mechanism. When the fastening mechanism is released, a sudden perturbation of the

platform occurs causing its free oscillation until it finally returns to its initial stable state [18].
In patient rehabilitation or in athletes' training, subjects are placed on the moving platform and are instructed to try to maintain their postural balance upon the perturbation (Figure 2.2). These exercises are proven to contribute to increased sensomotoric skills, ankle, knee and spine stability as well as muscle strength or even lung capacity [19].

Figure 2.2. Posturomed perturbation tests

2.2 Perturbation tests

For the purpose of this master thesis, no human subjects were used in the experimental procedures. Instead, the motion of the Posturomed© platform was tracked during perturbation tests in order to acquire its position data and calculate its displacement from its original steady state.

The tests took place in a process similar to the one explained in the previous section. The Posturomed© was first equipped with a set on biomechanical markers on its upper surface (Figure 2.3), that enabled the platform's detection from the Optitrack camera set. Zebris CMS10 was connected to the platform through a set of wires and accompanying ultrasonic transmitters. A MEMS MPU-9150 accelerometer was also mounted on the platform to allow us to perform acceleration data acquisition (Figure 2.4).

Figure 2.3. Reflective tape

Figure 2.4. MPU-9150 taped on the Posturomed platform

The next step was to connect the Posturomed to all the necessary computers with data acquisition softwares for all three position measurement methods. Three different computers and softwares were used, one for each individual method. Finally, the platform was locked outside of its resting point using the fastening apparatus of the device. After motion data recordings was initiated on all three DAQ softwares, the platform was released from the fastening mechanism and performed a free damped oscillation. Once the oscillation had damped significantly, the recordings were terminated and data from each measurement methods was saved in individual files on every computer. Overall, around 240 tests were run, each one consisting of 3 data sets of equal size, one for each one of the methods used.

2.3 Equipment set-up

In order to start running the experimental tests, all the equipment (devices, hardware and software) had to be properly set. To begin with, the Posturomed device was brought into the capture volume of the Optitrack system, to ensure it was detectible by the infra-red camera set. Six infra-red markers were placed in a hexagonal shape on the upper surface of the Posturomed's platform (Figure 2.6) to to enable its detection by the motion capture system. For the Posturomed, a *four-spring condition* was selected, allowing only a unidirectional platform motion in the horizontal plane.

Secondly, the Zebris CMS10 system was connected to the Posturomed, as well as to a power source and a computer with the appropriate software (WinPosture). The CMS10 was connected to the Posturomed device through a set of wires, leading to the ultrasonic transmitters, located on the side of the platform. The ultrasound-based measuring panel was arranged at a 30 degree angle on the side of the platform (Figure 2.5), to enable correct detection of the pulses emitted by the transmitters.

Figure 2.5 Zebris CMS10 placement relative to the Posturomed

Figure 2.6. Posturomed platform marked with reflective tape

Finally, the MEMS accelerometer was mounted on the platform to enable the acquisition of acceleration data. Double-sided adhesive tape was used so that the bottom surface of the MPU chip was securely and in direct contact with the platform surface (Figure 2.7). Due to its shape, the accelerometer could only be mounted on the sides of the platform, or on any point on its surface providing with MPU was mounted directly onto the Arduino board.

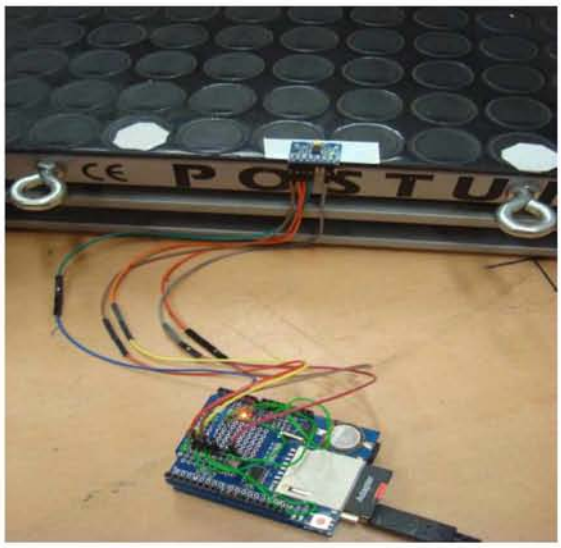

Figure 2.7. Connection of MPU-Arduino configuration to the Posturomed

2.3.1 Data acquisition

The next stage was to prepare the necessary software in order to acquire the measurement data from the platform's perturbation tests.

1.Optitrack

For the motion capture method, the software *Motive* was used for data acquisition. First, a new project and a *rigid body* were created on Motive. The rigid-body consisted of a visual representation of the oscillating platform on the software. Taking a look at Motive's user interface one can see, depicted on the screen, all the markers that are detected by the camera set-up at that specific time. In order for Motive to be able to monitor the motion of the Posturomed during oscillation, the visible markers from the platform's surface were selected on the screen (Figure 2.8) and a rigid-body representation was created out of those markers (Figure 2.9). Once that was done, recordings of the platform's motion caused by the perturbation tests were enabled.

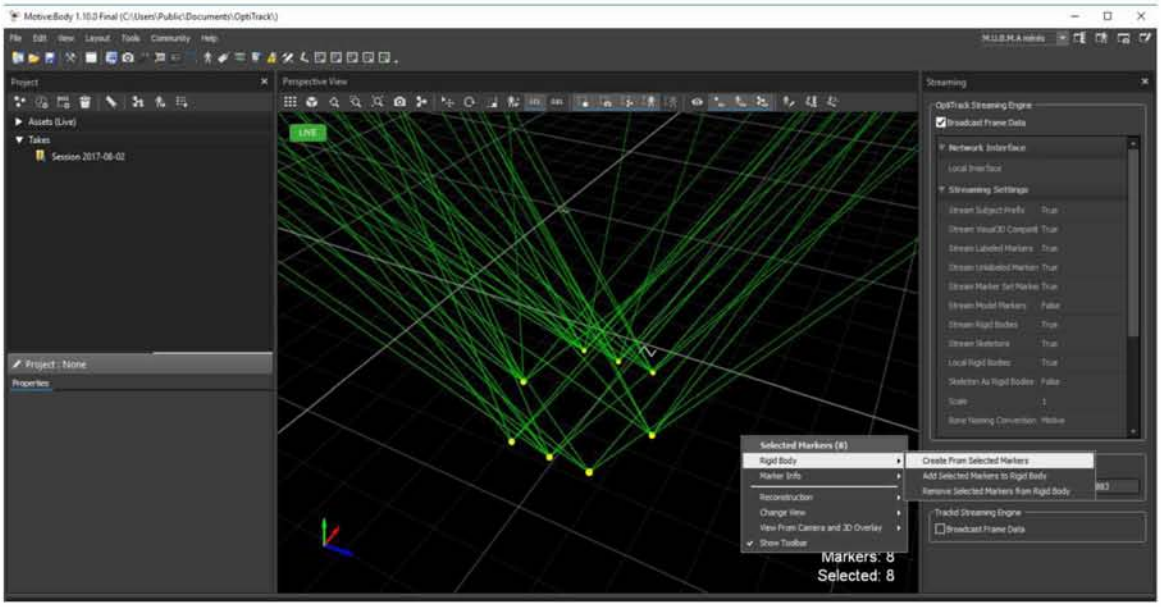

Figure 2.8. Visible marker seletion-Motive

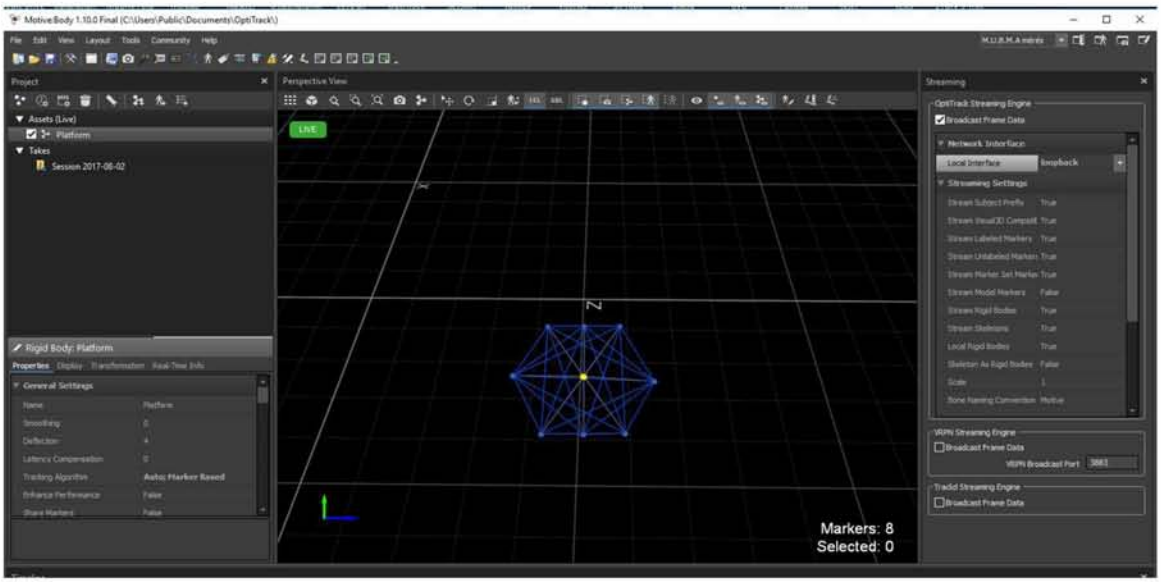

Figure 2.9. Rigid-body "Platform"-Motive

However, this configuration was not enough as Motive can only capture and record 2D and 3D images of the ridig-body's motion, but not motion data such as acceleration, velocity or position. As a result, a way to stream the data to a data acquisition software was required. Luckily, as mentioned previously, Motive allows live-streaming of motion data to other softwares that do allow data acquisition (DAQ). For this purpose, a custom-made DAQ software, called *MUMBA* (or *MOGI Universal Motion Analyzer*) was used. MUMBA allows users to record Posturomed perturbation tests and save the acquired platform motion data to an excel file.

To enable data streaming from Motive to MUMBA, the two pieces of software needed to be connected. This was simple enough to accomplish; first, on Motive's streaming panel, the appropriate computer IP address was selected on the "local interface" panel. Because the data streaming from Motive to MUMBA takes place on the same computer, the local interface IP address was set to "Loopback", which corresponds to the IP address of 127.0.0.1. Afterwards, this address was copied to the "server" and "local" panels of the MUMBA user interface, and connection was achieved by hitting the "connect" button.

One last step for the software configuration was to select the appropriate custom-made model for data acquisition on MUMBA. This was done by selecting the model "Posturomed" on MUMBA's "Options" panel for the available GaitModels (Figure 2.10). One important step to be taken with the purpose of achieving data streaming was to set the rigid-body's name on Motive to "Platform". This way, MUMBA would "recognize" the Posturomed and enable data acquisition from its motion.

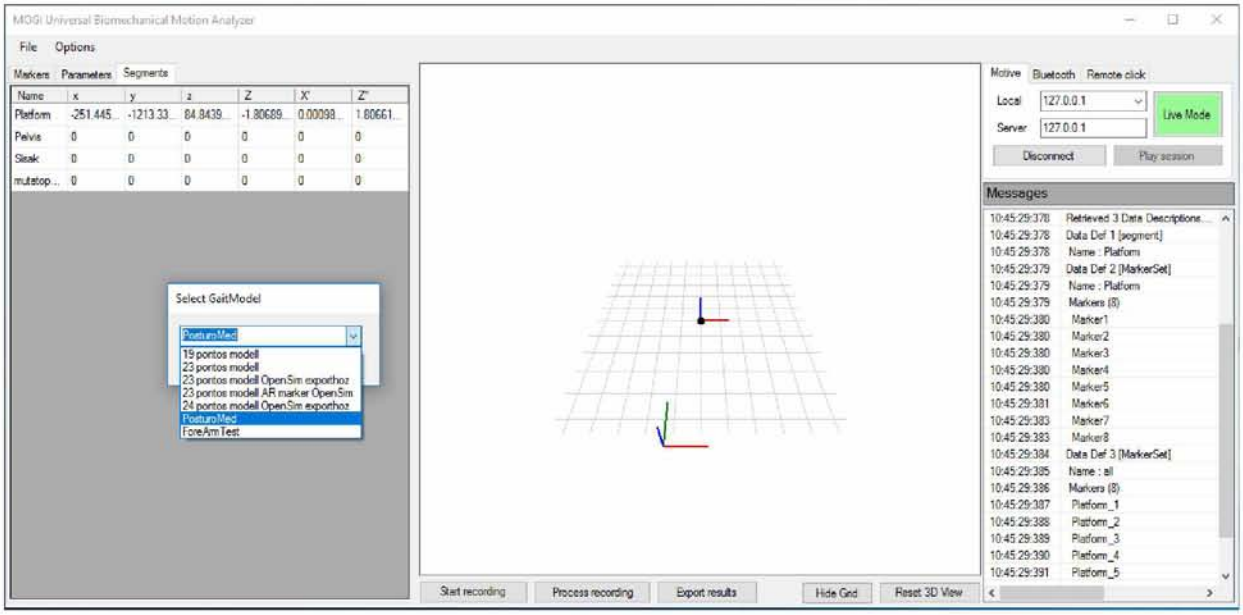

Figure 2.10. "Platform" motion data acquisition-MUMBA

After that step was completed, recording the perturbation tests could begin. Firstly, the platform was fastened outside its resting point. Recordings were initiated by hitting the "Start recording" button on the software's user interface. Immediately after that, the platform was released from the fastening mechanism and performed a free oscillation. After a few seconds, oscillation damped significantly and that was the moment when the recording was terminated by and the recorded data of each measurement was saved in a separate text file.

2. Zebris CMS10

The selected Zebris software, WinPosture, allows motion data acquisition as well as multiple visualizations of the stabilometric signal during the perturbation tests. In order for WinPosture to be able to collect the motion data, the software required information on the desired test type. To do so, the option "Provocation test" was selected from the software's control Panel, as shown in Figure 2.11:

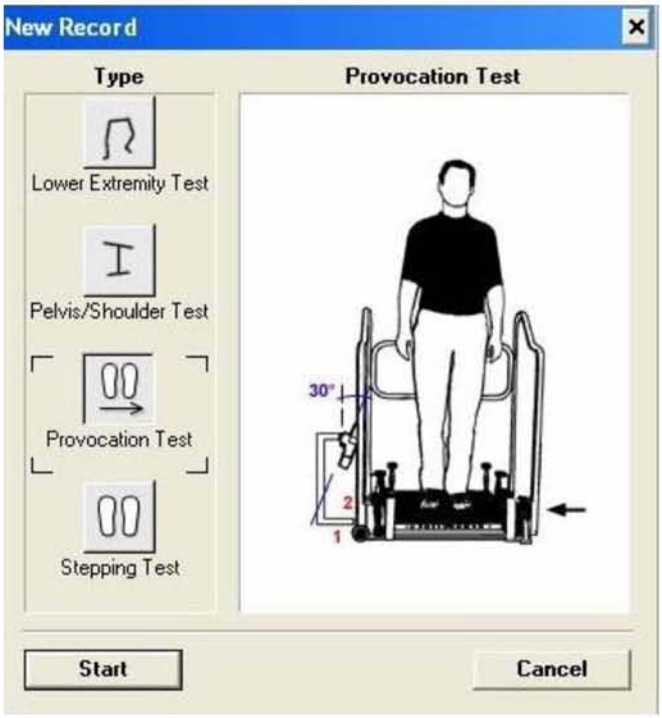

Figure 2.11. Test-type selection-WinPosture

By choosing the test type, WinPosture instantly provides clear and precise instructions on how to connect the software with the hardware used for the provocation test which, the Posturomed device. By following the directions, the Posturomed was connected to the CMS10 simply by using a set of cables attached to the Zebris hardware. In addition, the CMS10 had to be connected with the computer than contains the WinPosture software, so as to enable posturographic data acquisition.

Once connection was established an a new project was created, 4 graphs appeared on the software's user interface. These graphs corresponded to the platform's position measurements on the X, Y and Z axis, as well as on the XY level, as detected by the ultrasound transmitter-receiver system of the CMS10. When the platform was still stable, all graphs consisted of a straight line parallel to the horizontal axis of each graph. In a separate panel underneath the graph set, the software provides users with directions on the necessary steps needed for data recordings.

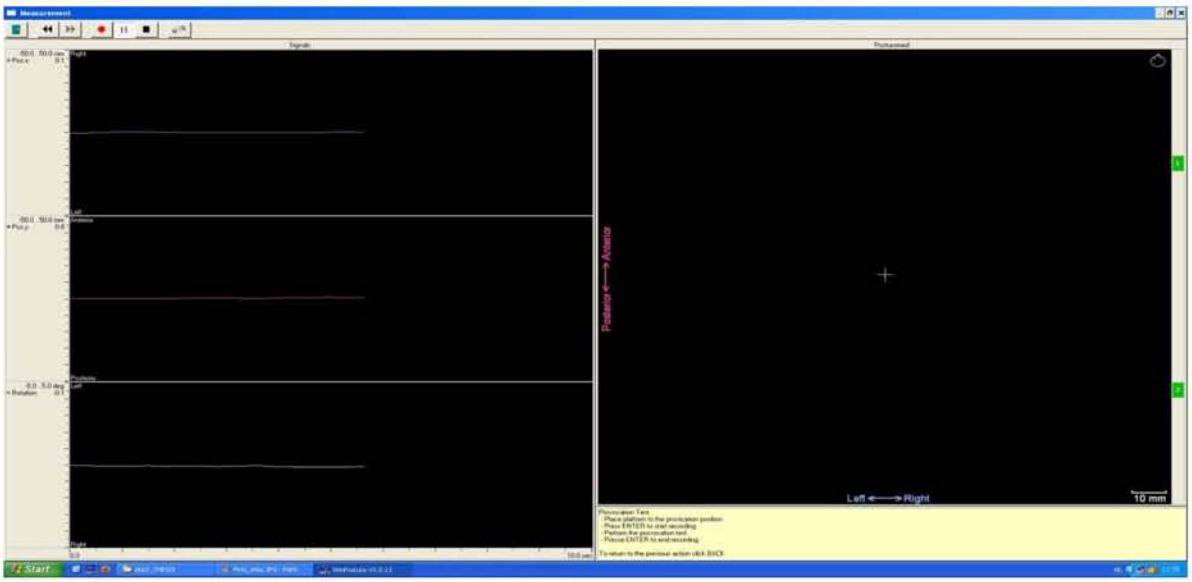

Figure 2.12. Zero measurement-WinPosture

First, with the platform in its resting point position, the user is instructed to press “Enter”. By doing so, WinPosture takes a zero measurement (Figure 2.12), which is essentially the calculation of the platform’s resting point position. The calculation of the coordinates of this position is what later enables the software to measure the platform's displacement from it during the perturbation tests. After this step was completed, the platform was locked outside its resting point using the fastening mechanism. Once the platform was secured, it was ready for the perturbation tests and motion recordings. By hitting “Enter” once again, position recordings were initiated. Shortly after this, the Posturomed platform was released from the fastening apparatus and performed a damping oscillation. Displacement from the resting point on the X,Y,Z axis as well as the motion trajectory of the platform on the XY level were demonstrated on the graphs on the user interface (Figure 2.13). When ending the recordings, position data was saved in a text file.

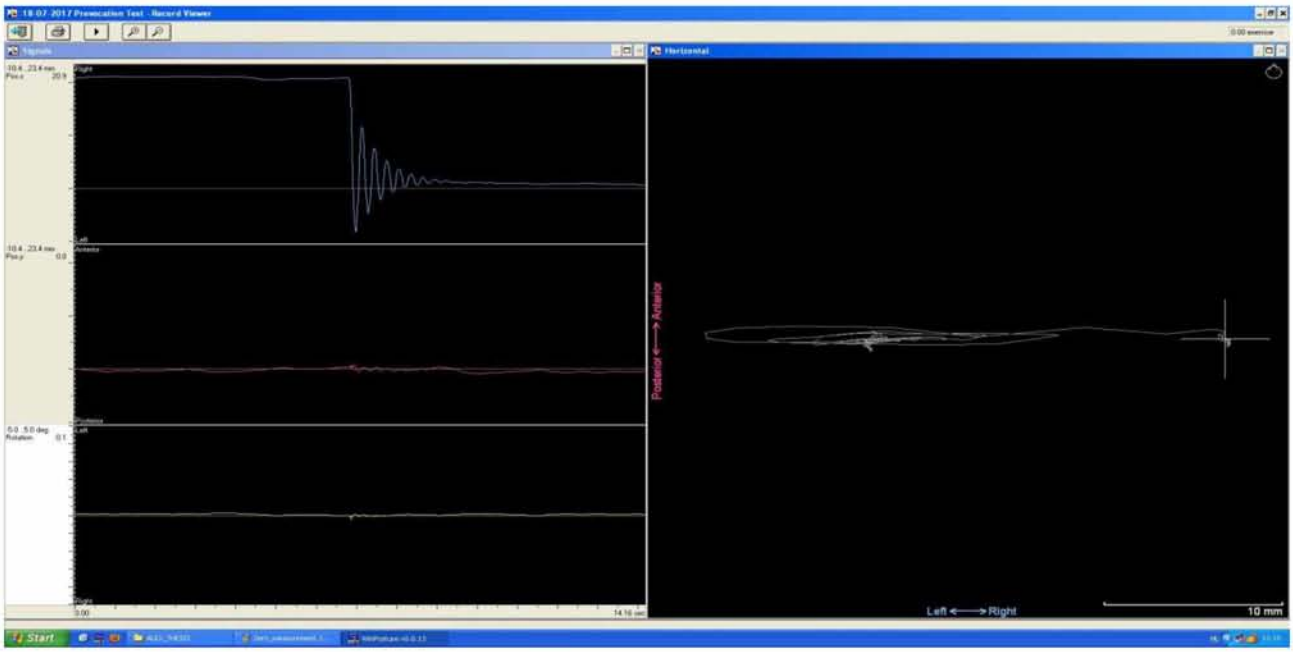

Figure 2.13. Displacement vs. Time graphs for the x,y,z axis and the x-y level-WinPosture

3. MEMS accelerometer

Acceleration data obtained by the position measurements were saved as individual text files on an SD card, connected to the Arduino UNO board. In order for the data acquisition to take place, a connection had to be established between the board and the accelerometer as well. That was achieved by connecting the appropriate pins of the MPU-9150 to the corresponding pin sockets of the Arduino board, as demonstrated in Figure 2.14. The MPU chip has a set of 8 pins in total, out of which 5 were used:

1) VCC: provides voltage to the chip
2) GND: ground (0 voltage) supply pin
3) SCL: I^2C serial clock pin
4) SDA: I^2C serial data pin
5) AD0: I^2C slave address LSB pin

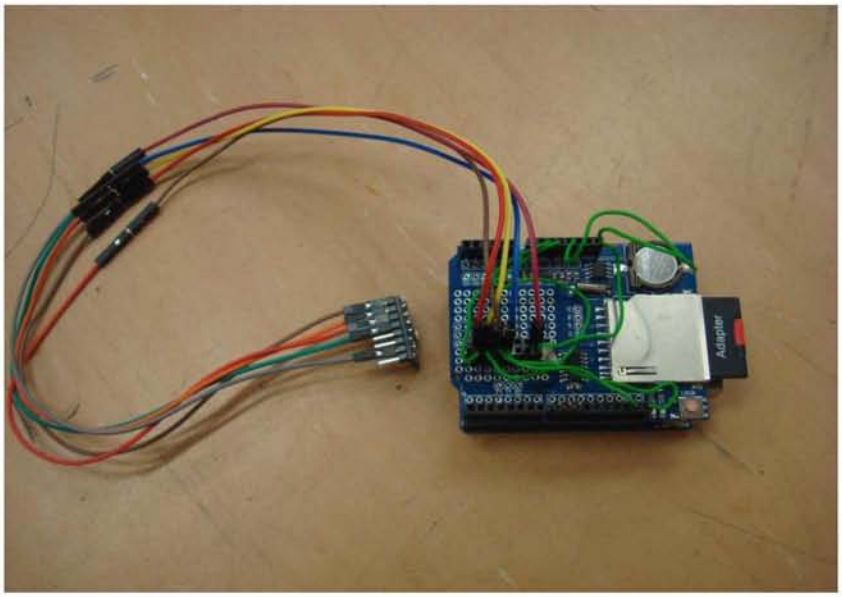

Figure 2.14. MPU-9150 and Arduino UNO connection

These input pins were connected to the respective output pins of the UNO board and enabled the data acquisition from the SD card.

Once the Arduino board was successfully connected with the MPU-9150, the UNO board had to be provided with a suitable code (sketch) which would enable acceleration data acquisition from the MPU chip's motion. For this purpose, an already existing and tested sketch from a previous project was used, with a few minor alterations in order to customize it to the requirements of the current experimental procedure.

To upload the sketch onto the UNO board, the user needs to connect the board to the computer with the USB cable, and then hit the command "Upload" on the IDE user interface. A pop-up message notifies you when the upload has been completed successfully. The sketch only needs to be uploaded on the board once and it automatically runS once the board is connected to a power

source. The code can be overwritten simply by uploading a new sketch on the board.

Figure 2.15. Laptop used as power source for the UNO board

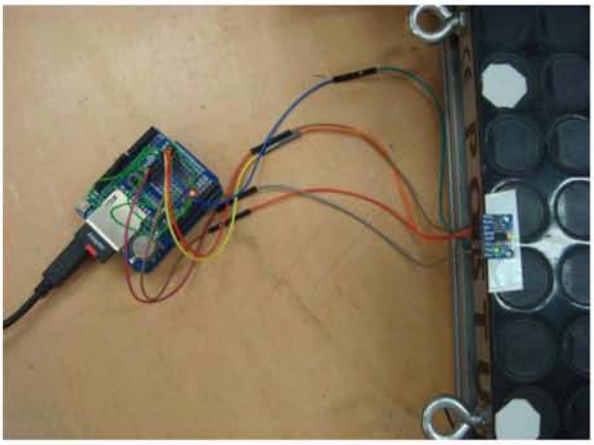

Figure 2.16. Connection of MPU-Arduino configuration to the Posturomed

The next step was to connect the Arduino board with a power source, as demonstrated in Figure 2.15. For this, the USB cable was used to provide power supply from a computer in the vicinity of the Posturomed platform and the accelerometer mounted on it (Figure 2.16). To commence the data acquisition process, the reset button on the UNO board was pressed. The time limit for acceleration data acquisition was set to 7 seconds. That means that every 7 seconds, a new text file was created on the SD card which included data that corresponded to that particular time period. To avoid the creation of multiple text files for each position measurement experiment, the Arduino board was disconnected from its power source once the Posturomed platform's oscillation had damped significantly.

A very important point that attention must be drawn to is the fact that *position measurements with all three different methods were performed simultaneously.* This means that, during each perturbation test, motion data recordings were initiated, performed and terminated from MUMBA, WinPosture and Arduino at the same time. Consequently, position measurements with all of the methods were extracted from the exact same platform oscillation test. The simultaneous acquisition of data using all the methods is what later allows performing a statistical analysis of the position measurement results and see if they are significantly (statistically) different.

3. Data processing methods

3.1 Construction of displacement matrices and graphs

For the processing of the acquired platform motion data, a custom-made Matlab script was contstructed. The script consisted of a number of sections, each responsible for a specific part of the data processing and result extraction.

In order to be able to perform a comparison between the position measurements of the three methods, three individual matrices had to be constructed, each including the position data acquired from the Optitrack, the Zebris and the MPU device respectively.

Before continuing with the description of the data processing methods, a couple of important notes must be made; first of all, the sampling frequencies of the three methods are *not* the same. For the MPU-9150 and Zebris, the sampling frequency is set to *50 Hz*, while for Optitrack it is set to *120 Hz.* That means that the sampling times of each method differed and thus, the constructed time and position matrices were not identical for all three methods.

In adition, the comparison of the three methods was based on calculations made concerning the *total platform displacement on the X-Y level,* calculated using the following formula:

$$D = \sqrt{pos_x^2 + pos_y^2} \tag{3.1}$$

where *pos_x* and *pos_y* are the position matrices on the X and Y axis. By including the additional information on the z-axis position of the platform, one could calculate the total displacement of the platform in the three-dimensional space by using the extended version of the formula for *D*:

$$D = \sqrt{pos_x^2 + pos_y^2 + pos_z^2} \tag{3.2}$$

Using this formula and extending this study to the three-dimensional space would necessitate a significantly more complex data filtering and analysis method, which would subsequenlty require further time and effort spent on researching and testing methods for optimizing the matlab script. Of course, extending the study's scope is certainly a possibility and would enable comparison of the accuracy of the three methods when studying the motion of an object in three instead of two dimensions. This approach would provide very valuable results for human motion and biomedical applications, where the moving object is a human body and its motion is ,of course, three-dimensional. For the time being, however, the current study was limmited to measuring the position of the Posturomed platform on the x-y level.

3.2 Optitrack position data processing

For Optitrack, the process was fairly simple. As mentioned in section 1.4, Optirack's software, Motive, recorded the Posturomed platform's position data in the X, Y and Z axis during the platform's oscillation and saved the measured data from each measurement/test on an Excel file. These files consisted of columns which corresponded to the sampling time, the position data on the X,Y and Z axis respectively as well as the Posturomed yaw.

The first step was to form a matrix including the position data acquired from the measurements

with Motive. This was done simply by reading the excel file of each measurement and inserting its data to a matrix created on matlab. For the purpose of this study, only the data included in the first 3 columns of the position matrix was used. The first column conisted of the time data , while the second and third column conisted of the platform's position data on the x and y axis respectively. The remaining two columns included the z-axis position data as well as the Posturomed yaw data. As was mentioned previously, the study was limited to the x-y level and therefore, there was no need to use the z-axis position data.

By using formula (3.1), the total platform discplacement was calculated by substituing pos_x with the 3rd column of the previously construcyed matrix and pos_y with the 2nd column's data.

While performing the position measurement experiments, data recordings had to begin on all the three different devices and softwares approximately at the same time, before setting the platform in motion. Starting all three recordings simultaneously was not an option so, as a result, the recordings of the platform's oscillation start on different points in time depending on the method and the respective measurement or test. On all tests, recordings on WinPosture were initiated first, followed by Motive and the accelerometer. For Motive for example, the initial form of the position graphs that we constructed was the following:

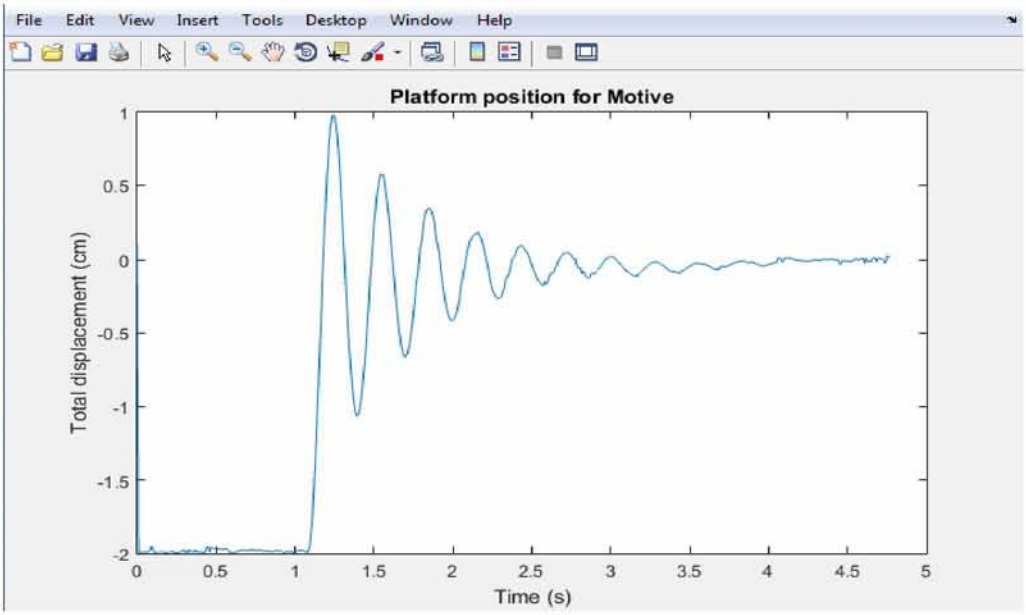

Figure 3.1. Displacement vs.Time graph prior to perturbation start cut - Motive

From the graph in Figure 3.1, one can observe that, during the first second of the recording, the position of the platform is stable at -2 cm, which means that during that time, the platform was still locked outsite its resting position. It can be seen that the oscillation began at around 1.1 seconds and lasts for about 2- 2.5 more seconds. As explained above, it is obvious that the ammount of time before the oscillation started was inevitably going to be different for each method. This means that time was relevant to the respective method and that, for example, t=0

sec was not necessarily the time when the oscillation started, but rather the time when the recording process started on each individual device.

In order to be able to later compare the position graphs for all three methods, their corresponding time-position matrices and graphs had to start at the same point in time, which, to make things easier, was set at 0 seconds. In order to achieve this, the time and position data corresponding to the time before the start of the perturbation had to be "cutt off", essentially shifting the whole graph to the left. This technique yielded a time-position graph in which the oscillation of the platform started at *t=0* seconds. The final form of the graphs is presented later on in this section.

As mentioned in section 3.1, the sampling frequency of Optirack is 120 Hz. In order to be able to, later on, compare the position measurement data between the three methods, they needed to correspond to the same point in time, for each data set. This particularity necessitated downsampling Motive's data, in order for its sampling rate to match that of the other two methods, which was 50 Hz. With the frequency of 120 Hz, Motive provided position data for approximately every 0.00833 seconds of the perturbation's duration. By downsampling the data to 50Hz, only data corresponding to approximately every 0.02 seconds of the total duration of the perturbation test was saved, which was essentially the sampling time for the two other methods, Zebris and the accelerometer.

It is important to note that, due to a certain level of inaccuracy in the downsampling method that was used, the new calculated sampling time of Motive was not exactly 0.02 s, but closer to *0.017 s*. This deviation may have affected the accuracy of the study's final results, which will be explained futher in following chapters.

Upon downsampling the data, position-time graphs for this method could be constructed. The result of this step were graphs with a general form shown in Figure 3.2:

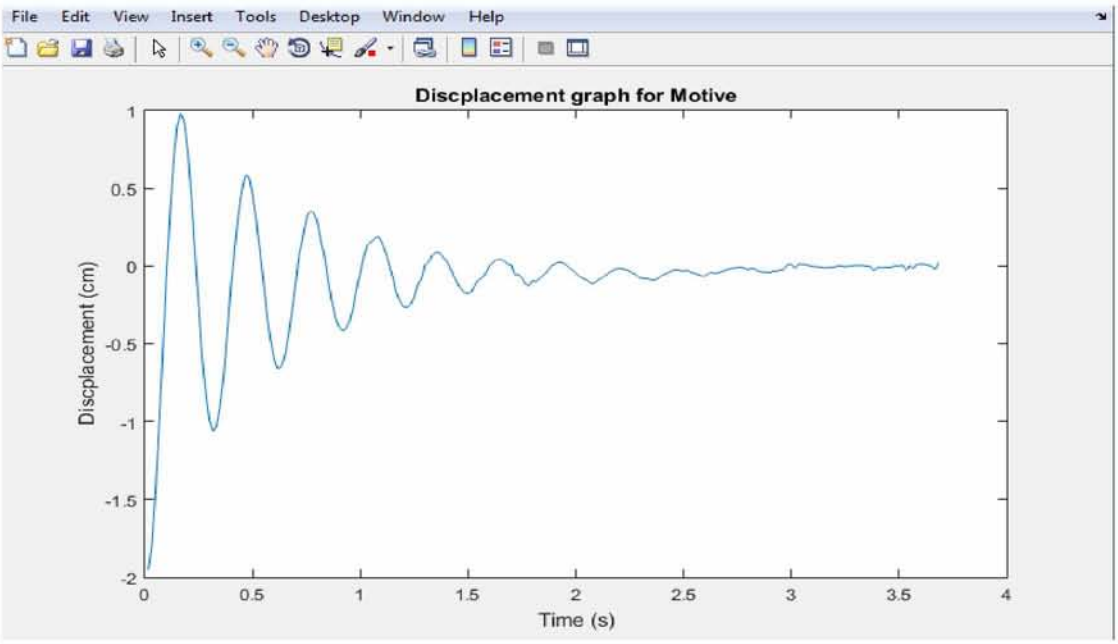

Figure 3.2. Displacement vs.Time graph - Motive

From the graph one may observe that the initial position of the platform was around -2 cm, which means that when the perturbation started, the platform was locked 2cm away from its resting position by the fastenning mechanism of the Posturomed device. It can also be noted that the platform underwent a *damped oscillation* with an amplitude of around 1 cm, in this particular measurement. According to the graph, the *damping time*, or the time it took for the platform to stop oscillating significantly, was around 2-2.5 seconds. Of course, the damping time as well as the amplitude of the oscillation can vary slightly between different measurements due to certain external factors, such as a more abrupt oscillation start.

3.3 Zebris position data processing

A similar process was carried out in order to acquire position matrices and graphs for the data ascuired using the Zebris device and software (WinPosture).

First, motion data was read from the excel file of each measurement and imported in a matlab matrix. Similar to Optitrack's case, the first column of each matrix consisted of the time data, while the second and third column consisted of the platform position on the y and x axis respectively. The fourth column included the z-axis position data, while the fifth and final column included the platform's rotation data. Again, for the purpose of this study, only the x, y position data as well as the time data was used.

Next, data prior to the perturbation start was cut off, which shifted the whole perturbation graph to the left, so that the oscillation began at 0 seconds. The final form of the position graphs for Zebris can be seen in Figure 3.3:

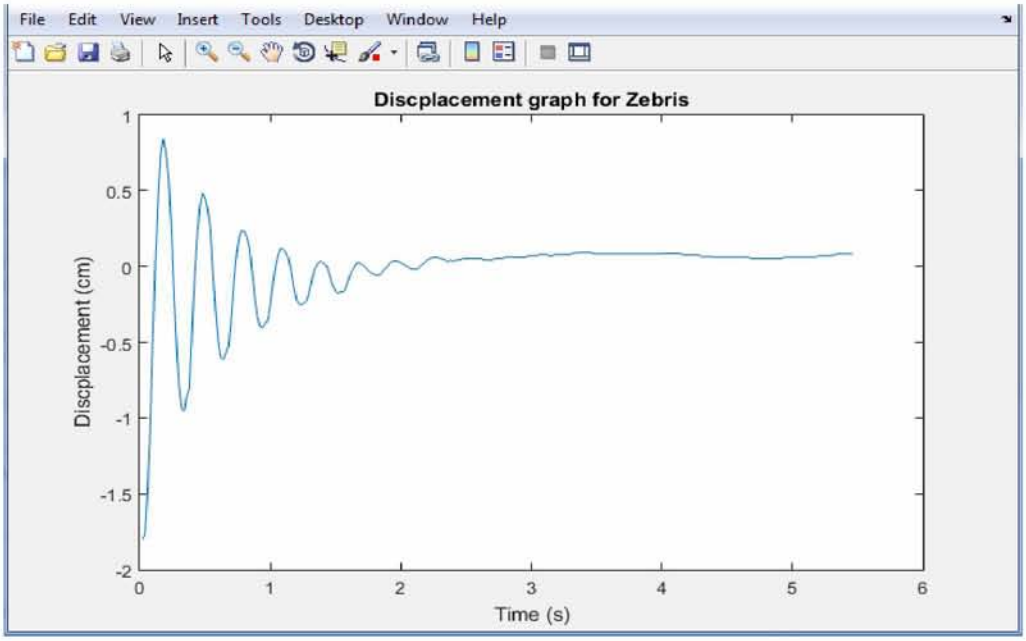

Figure 3.3. Displacement vs.Time graph - Zebris

3.3.1 Errors in the Zebris method

While contstructing the platform's displacement graphs for this particular method, a few problems were encountered; in some of the data sets, there was a small discontinuation in the data series, which meant that a few of the cells of those particular position matrices were registered as "NAN" ("Not A Number") instead of actual numbers, or position values. This error took place during the platform's motion recordings and data ascuisition and was a result of a fault in the software's DAQ process itself.

The result of this error was a "gap" or a discontinuation in the displacement graphs (Figure 3.4) constructed from the aforementioned data sets, which later on would not allow for a comparison between the position matrices of Zebris and the other two methods.

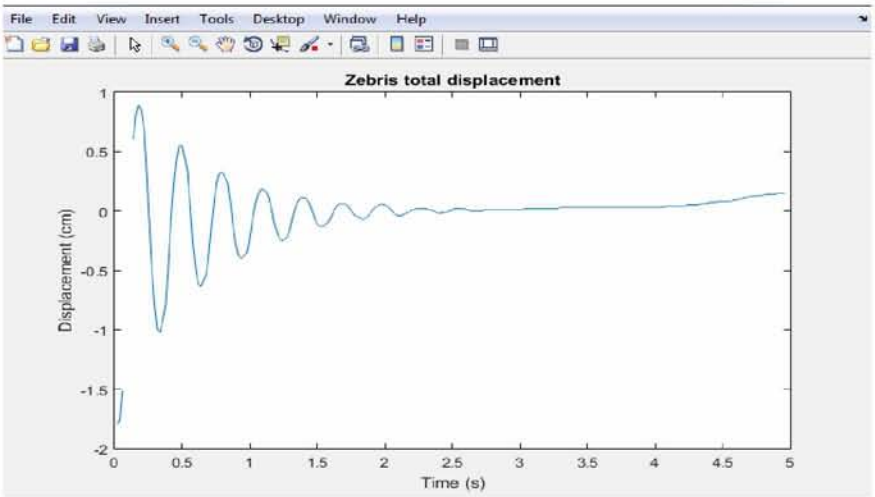

Figure 3.4. "Gap" in displacement data - Zebris

In order to overcome this obstacle in the experimental process, the missing part of the problematic position graphs had to be reconstructed. To achieve this, Matlab's *curve fitting tool* was used (Figure 3.5). The purpose of curve fitting is essentially to find a curve which matches the data set imported in the tool in the best possible way.

The first step was to import the correct data; in this case, the x-axis data was the time vector, while the y-axis data was the vector of the total displacement of the platform, as calculated by Zebris' software. Next, a curve fitting method needed to be chosen; there are a number of available methods provided by the tool, such as exponential, gaussian, interpolant or polynominal, to name but a few. A *linear interpolant* curve fitting was chosen fort he purpose of this study.

The result of this process was a continuous curve exactly matching the given data points while, at the same time, completing the missing NAN values. In addition, the curve fitting tool provides the option to generate the code behind each curve fitting process. After importing all the data and specifying the necessary parameters, such as the prefered curve fitting method, clicking the "generate code" option offered access to the code corresponding to the followed curve fitting process. By including it in the Matlab script, the fitted curve could be calculated for all problematic position data sets, simply by running the curve fitting code for each different position data vector,

without needing to re-set the parameters in the tool itself.

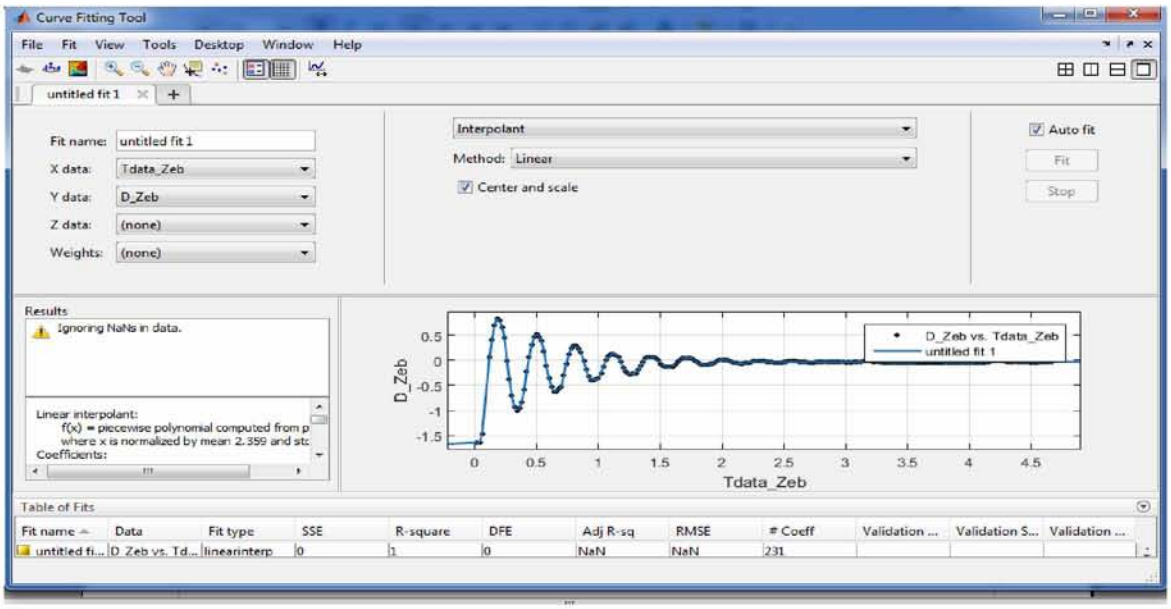

Figure 3.5. Curve fitting on Zebris displacement data

After performing the curve fitting, the matrix corresponding to the generated curve could easily be extracted and used instead of the initial discontinuous position data acquired by Zebris.

Another problem that was faced during processing of the Zebris position data was that a few of the data sets displayed unusually high position values at the start of the platform's oscillation, often up to 10 cm. Upon beginning its oscillation, the platform was locked only 2 cm away from its resting position, so it would be impossible for the amplitude to reach values higher than 2cm at any point in time, as the platform underwent a *damped* oscillation. For this reason, position values higher than 2 cm were incorrect and probably a result of an external factor which caused an error in the DAQ process. This factor could either be a malfunction of the WinPosture software's DAQ mechanisms or a sudden jolt of the platform caused by an abrupt start of the perturbation.

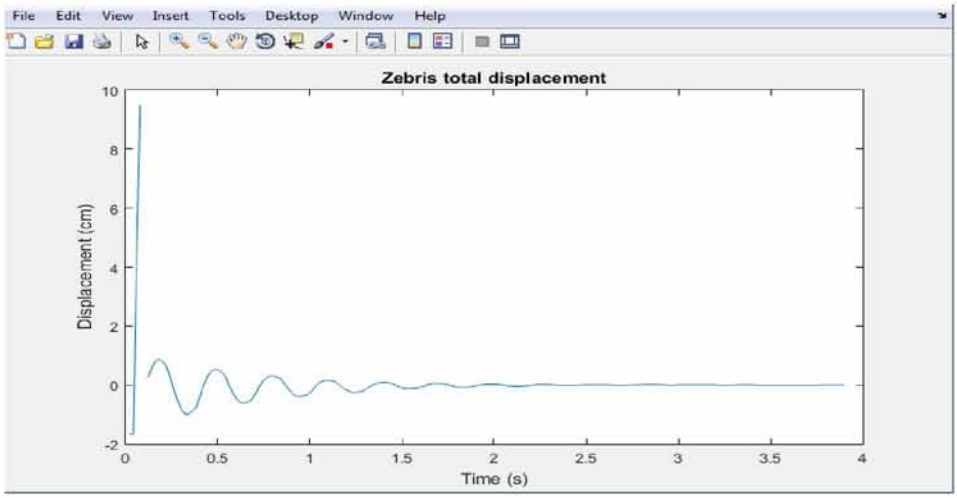

Figure 3.6. Initial overshoot in Zebris displacement

Including such data sets in this study would signify the existence of unusally high and incorrect position measurements in the followed process and could affect the quality and accuracy of the resutls. Fortunately, similar errors in the data sets did not come up very often, so in order to save time those problematic measurements were omitted from the epxerimental process.

3.4 MPU accelerometer position data processing

3.4.1 Obtaining position from acceleration data

Processing of the data collected by the MPU accelerometer was definetely the most challenging part of the present thesis.

The current study was focused on a comparison of methods for the acquistition of *position* measurements. Naturally, an accelerometer is designed to calculate the *acceleration* of the object on which it's mounted. As a result, it was necessary to find a means to obtain position data from the acquired acceleration data.

In physics, *acceleration* is the rate of change of velocity of an object with respect to time. In terms of calculus, *instantaneous acceleration* is the derivative of the velocity vector with respect to time:

$$a = dv/dt \tag{3.3}$$

It can be seen that the integral of the acceleration function *a(t)* is the velocity function, *v(t)*; the area under the curve of an acceleration versus time graph corresponds to velocity:

$$v = \int a dt \tag{3.4}$$

As acceleration is defined as the derivative of velocity, *v*, with respect to time *t* and velocity is defined as the derivative of position, *x*, with respect to time, acceleration can be thought of as the second derivative of position with respect to time:

$$a = dv/dt = d^2 x/dt^2 \tag{3.5}$$

Therefore, it's only logical that position *x* can also be seen as the integral of velocity *v* , or as the double integral of the acceleration *a*:

$$x = \int v dt = \int \int a dt \tag{3.6}$$

Acceleration has the dimensions of velocity (m/s) devided by time (s). In SI, the unit of acceleration is the meter per second squared (m/s^2).

With all the above being highlighted, it makes sense that in order to obtain position from acceleration, one has to perform a *double integration* on acceleration values in order to extract position values. For the purpose of this master thesis, *numerical integration* was used to obtain position from acceleration data.

Numerical integration essentially constitutes of a broad family of algorithms, the aim of which is to caclulate the numerical value of a defined integral. It is especially useful in cases where the

formula of the function one wishes to integrate, the integrand, is only known at specific points, such as obtained by sampling, often the case with embedded systems. In other words, numerical integration is usually used in cases of *discrete experimental data*.

During experiments carried out for this thesis, the data obtained from the MPU was the result of a sampling process, where the Arduino script we used was programmed to register the values of the platform's acceleration, as captured by the accelerometer, at specific points in time. The results of this process were registered in excel files which contained acceleration values during a time period of approximately 6 seconds, with intervals of 0.02 seconds.

Two different approaches were followed; for the first one, where the formula of the integrand was unknown, numerical integration was performed directly on the acceleration data in order to obtain velocity. Those results were integrated once again in order to yield position data. The second approach was an attempt to calculate the function of the platform's acceleration with relation to time, *a(t)* , by using Matlab's curve fitting tool. A double numerical integration was carried out on a(t) in order to obtain the position's equasion, *x(t)*. Both approaches will be throroughly analysed in the next section.

3.4.2 Numerical integration of acceleration data

3.4.2.1 The trapezoidal integration rule

Matlab is equipped with a variety of different functions which enable users to perform direct numerical integration on a specific data set, appliying one of the available integration methods. For the purpose of this master thesis, *cumulative trapezoidal numerical integration* was conducted on the acceleration data.

In mathematics, the function of an integral is to calculate the area which lies under the curve of a specific function. Given a function, the trapezoidal rule inlays trapezoids onto the function (Figure 3.7) and then computes the sum total of all the areas of the trapezoids covering the area under the curve (green area). If the integration is performed on an interval such as [a,b], the trapezoid rule partitions the integration interval, inlays a trapezoid to each individual section, and finally sums the results in order to approximate the entire region below the curve corresponding to the given function. The more trapezoids inlayed onto the function, the more accurate the computation is, as the area which is unaccounted for (red area) is minimized.

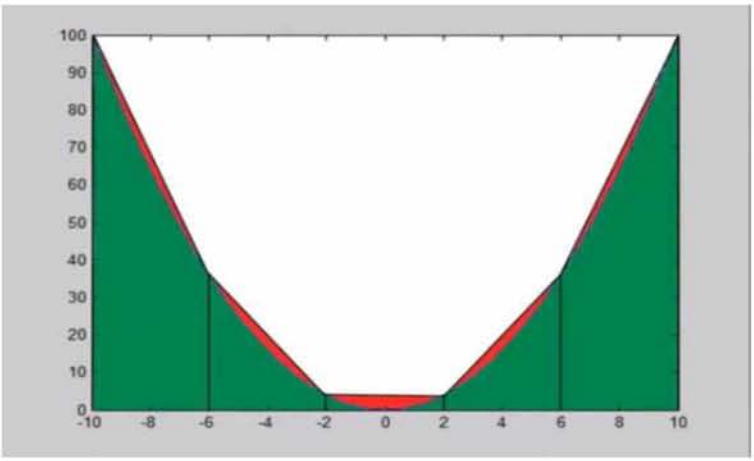

Figure 3.7. Graphical representation of the trapezoidal rule

Before proceding to analyse how Matlab performs numerical integration, it is important to mention a few necessary steps that needed to be taken prior to actually integrating the acceletion data to position data.

3.4.2.2 Acceleration signal

First, acceleration data was imported onto Matlab from the excel files. Every data set corresponded to one specific measurement of the Posturomed platform's motion. Three vectros were constructed; the first one included the time data, while the other two included the acceleration values of the x and y axis respectively. Each value of acceleration corresponded to a certain point in time, thus, each cell of the acceleration vectrors corresponded to a cell in the time vector.

The acceleration data as acquired by the MPU accelerometer is in gravitational acceleration, or g, units, so before continuing with out computations, we needed to tranform it to m/s^2, which was achieved by multiplying the two acceleration vectrors with 9.81, the value of g.

The original form of the acceleration values over time, for the x and y axis, is shown in the graphs presented in Figure 3.8:

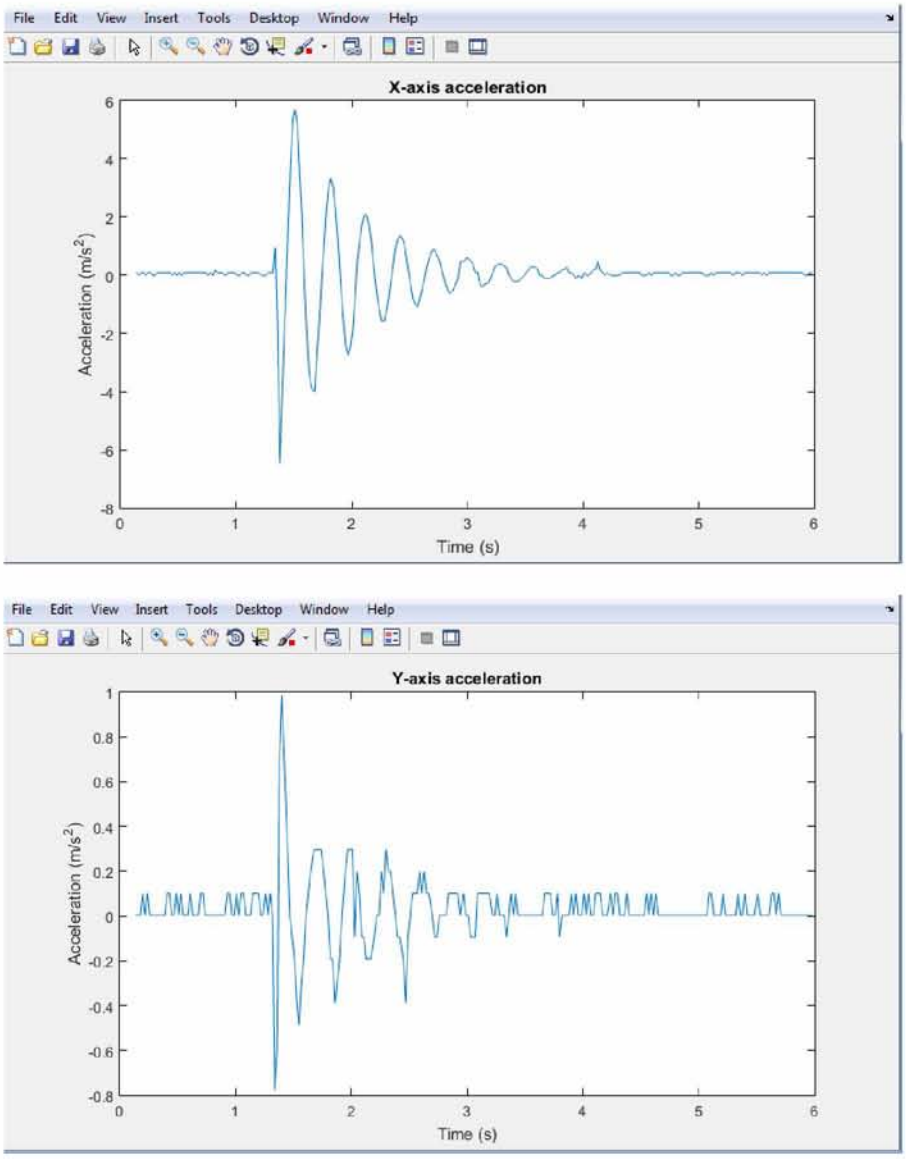

Figure 3.8. Original x and y-axis acceleration signals

It can be seen that, when plotted in Matlab, the acceleration signal has the form of a *damped oscillation wave*. In physics, the position of an object undergoing a damped oscillation is described by the following mathematical formula:

$$x(t) = A_0 e^{(-\gamma t)} \cos(\omega t + phi) \tag{3.7}$$

where *x(t)* is the function of position with respect to time. A_0 is the maximum amplitude of the signal (m), *γ* is the damping coefficient, *phi* is the phase (rad) and *ω* is the signal's frequency (rad/s). The term $A_0 e^{(-\gamma t)}$ refers to the *exponential decay* of the signal's amplitude.

Given that acceleration can be seen as the second derivative of position, it only makes sense that, since the position signal has a sinusoidal form, then its first and second derivative will also have a sinusoidal form. By computing the second derivative of x(t) as described by the formula 3.7, the result is the following equation:

$$a(t) = A\gamma e^{(-\gamma t)} \cdot (2\gamma\omega \sin(\omega t - \alpha) + (\gamma^2 - \omega^2)\cos(\omega t - \alpha)) \tag{3.8}$$

This is all to explain that, the signal of acceleration will have the same general form with the signal of position which is that of a damped oscillation's wave curve.

A few important observations should be made; to begin with, looking at the graphs in Figure 3.8, one can see that the damped oscillation curve does not start at point time t=0, but rather around t=1 sec., both for the x and y axis, as was the case with Optitrack and Zebris. As was previously mentioned, the reason behind this is that the platform was set in motion a couple of moments *after* the recordings had started for all three methods. In order to eliminate this part of our plots, where no platform motion took place, the first part of the acceleration plots for both axis was cut off, as it included no information on the platform's acceleration in that time period. Furthermore, including this part of the data in the computations could cause potential errors with filtering and numerical integration and eventually affect the quality of the end results.

It also be observed that, at the beginning of the oscillation on both plots, there was a sudden "drop" in the acceleration data (Figure 3.8). This initial peak, or "drop" of the curve, was cause by the sudden jolt of the platform upon its release from the fastening mechanism. This abrupt start of the platform's motion affected the smoothness of the wave form of the acceleration signal and created a sharp, jagged edge on the graph at that specific point in time, which later on affected the quality of the integration and filtering process. For this reason, this part of the waveform was omitted and, consequently, the respective acceleration data was not included in the computations and further signal processing which followed.

3.4.2.3 Digital signal filtering

By taking a close look at the acceleration graphs for the x and y-axis, one can easily notice that the acceleration waveforms are not perfectly smooth. This is more obvious for y-axis data, where the acceleration vs. time plot consisted of a number of jagged peaks (Figure 3.8). It is still obvious that the curve for the y-axis acceleration data corresponds to a damped oscillation, but the curve is by no means smooth and certainly not as smooth as that of the x-axis data.

It is crucial to remember that the data collected throughout the Posturomed's perturbation tests derived from a *digital signal*, which essentially represents data as a sequence of discrete time and

acceleration values. Every digital signal carries with it some level of noise. *Digital noise* is a term used to describe any unwanted modifications that a signal may suffer during capture, storage, transmission, processing or conversion. In signal processing, a *digital filter* is a system which, by performing mathematical operations on a sampled, discrete time-signal, serves to recover the original signal from the noise-corrupted one.

Transfer function

A filter is characterized by its *transfer function.* The transfer function of a filter is a mathematical function which gives the corresponding output value for each possible input value. It is most often defined in the domain of complex frequencies. The passage to and from this domain is operated by the *Laplace transform* and its inverse. The transfer function *H(s)* of a filter is the ratio of the output signal *Y(s)* to the input signal *X(s)* as a function of the complex frequency *s*:

$$H(s) = \frac{Y(s)}{X(s)} \tag{3.9}$$

where $s=\sigma + j\omega$. For signal processing applications, by defining σ=0, the Laplace transform which includes complex arguments, is reduced to a Fourrier transform, with a real argument ω.

The frequency response of a signal can be classified into a number of different *bandforms*, describing which frequency bands the filter passes *(the passband)* and which it *rejects (the stopband).* There are different filter types depending on which frequencies pass and which get rejected:

-In a low-pass filter, low frequencies will be passed, while high frequencies will be attenuated,
-In a high-pass filter, high frequencies will be passed, while low frequencies will be attenuated,
-In a band-pass filter, only frequencies in a certain frequency band are passed etc.

Butterworth filters

A *Butterworth filter* is a type of signal processing filter designed to have as flat a frequency response in the passband area. The frequency response of a system is characterized by the *magnitude* of the system's response, typically measured in decibels (dB), and the *phase*, measured in radians, versus frequency, in radians/sec or Hertz (Hz) (Figure 3.9). The frequency response of a Butterworth filter is maximally flat in the passband region and rolls off towards zero in the stopband. When viewed on a logarithmic Bode plot, the response slopes off linearly towards negative infinity:

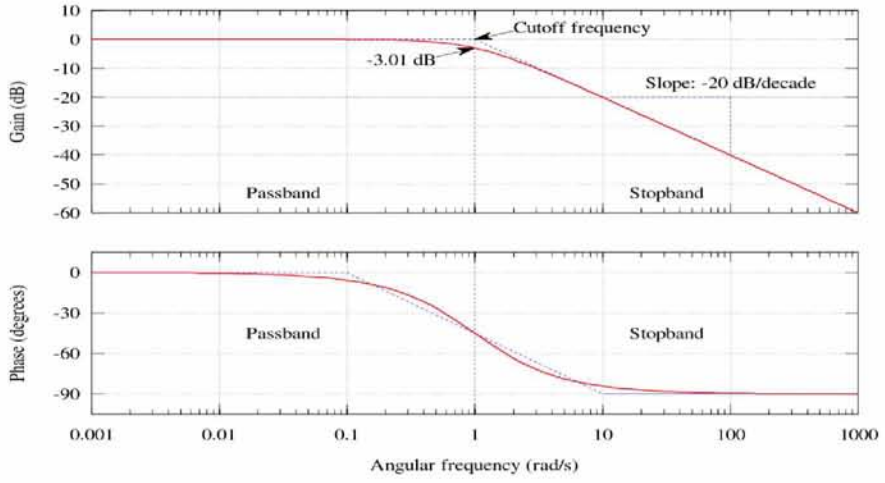

Figure 3.9 Butterworth filter frequency response

Like all filters, the typical prototype is the low-pass filter, which can be modified into a high-pass, band-pass or band-stop filter, or higher order versions of these. The gain *G(ω)* of a *nth-order* Butterworth filter is given in terms of a transfer function *H(s)* as:

$$G(\omega) = H(s) = H(j\omega) = \sqrt{\frac{G_o^2}{1+(\frac{j\omega}{j\omega_c})^{2n}}} \tag{3.10}$$

where G_o is the gain at zero frequency and ω_c is the *cut-off frequency,* which describes the frequency beyond which the filter will not pass signals. *n* is the order of the filter. As *n* approaches infinity, the gain becomes a rectangle function and frequencies below ω_c will be passed, in case of a low-pass filter.

By the definition (3.9) of the transfer function, it can be deduced that, knowing the output signal X(s) as well as the transfer function H(s), the input signal Y(s) can be calculated by multiplying X(s) with H(s):

$$Y(s) = H(s) \cdot X(s) \tag{3.11}$$

If X(s) is the noise-corrupted output acceleration signal, then by multiplying it with the transfer function, the original uncorrupted signal can be recovred. As a result, the key to filtering a signal is to compute the correct form of the Butterworth filter's transfer function. Matlab can design a Butterworth filter with the following command:

$$[b, a] = butter(n, w_n, ftype) \tag{3.12}$$

which returns the transfer function coefficients *a* and *b* of an nth-order Butterworth filter.
n is the order of the filter and w_n is the *normalized* cut-off frequency:

$$w_n = \frac{w_c}{\frac{w_s}{2}} \tag{3.13}$$

where w_s is the sampling frequency of the signal. Depending on the value of *ftype*, the function can design a high-pass, low-pass or band-pass filter. The transfer function coefficients *a* and *b* define the form of the filter's transfer function. The function which connects the input and output signals through the filter's transfer function is the following:

$$input = filtfilt(b, a, output) \tag{3.14}$$

Input and *output* are discrete-time digital signals. In this particular study, when *output* is defined as the noise-corrupted acceleration signal, *input* will be the original uncorrupted acceleration signal. “Bandpass” was chosen as the filter type, with a frequency band of [1 10] Hz.

The filtered x and y-axis acceleration signals can be observed in Figure 3.10:

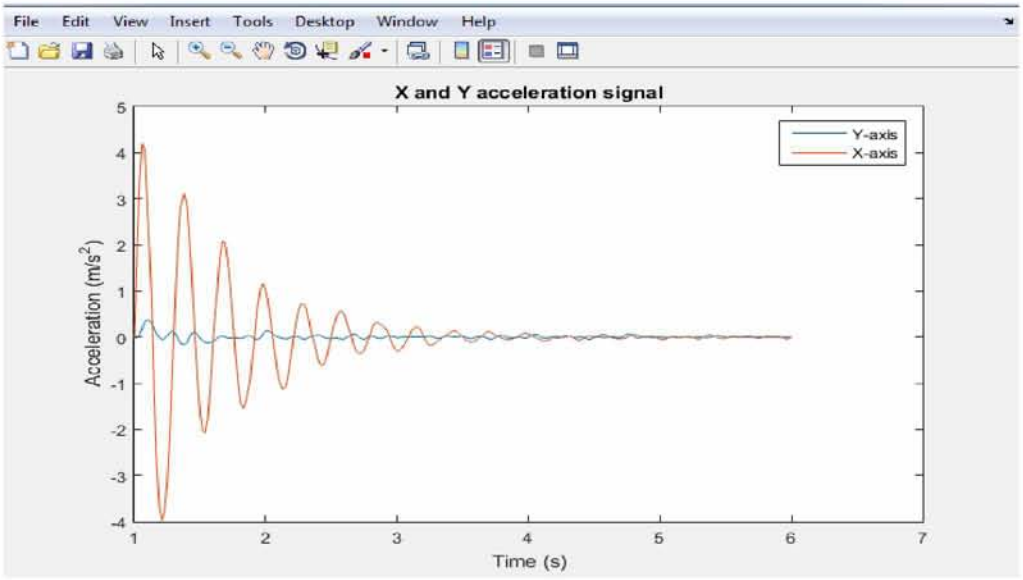

Figure 3.10. Filtered x and y-axis acceleration signals vs. Time

The now filtered acceleration data curve has a smoother form, with less sharp edges compared to its form prior to filtering. The filtered data was imported in two Matlab matrices, for the x-axis and the y-axis acceleration data respectively. The data contained in these matrices underwent two subsequent integrations in order to yield position data for the two axis, which was later combined into one formula for the total platform displacement on the x-y level.

In the next section, the integration process that was followed through the course of this study as well as its results will be explained and analyzed in detail.

3.4.2.4 Integration

Matlab provides the integrated function *cumtrapz,* which performs *cumulative trapezoidal numerical integration* either on a specific function f(x), or on discrete experimental data. For this approach, discrete acceleration and time data was used. More specifically, *cumtrapz (X,Y)* computes the cumulative integral of Y with respect to X using trapezoidal integratiom. X and Y must be vectors of the same length.

In section 3.4.1 it was explained how performing a double integration on acceleration data leads to obtaining position data. Instead of conducting one double integration, two single integrations were conducted, one on the acceleration signal to obtain velocity which was then integrated again in order to obtain position data for the platform's motion. Each integration process took place on the two distinct data sets for the x and y-axis data respectively.

The cumtrapz function was first applied on the *filtered* acceleration signal, for the x and y axis. For the x-axis data , in Matlab's command *cumtrapz(X,Y)* ,the time vector *T* was set as X and the filtered x-axis acceleration vector *Accel_X_filt* as Y. Similarly, for the y-axis data, Y was replaced with the filtered y-axis acceleration vector *Accel_Y_filt*, while X remained the *T* vector. Plotting the results of these integrations yielded the velocity vs time plots for both axis, as shown in Figure 3.11:

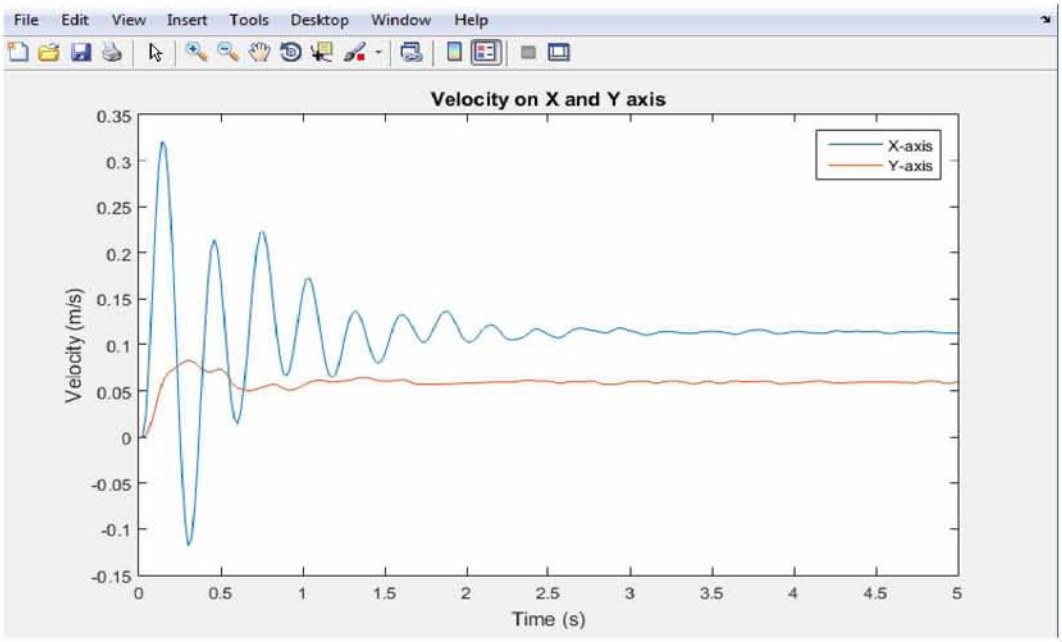

Figure 3.11. Unfiltered x and y-axis velocity signals vs. Time

While both curves seem to have sinusoidal forms, they show a slight offset, around 0.13 m/s for the x-axis velocity and 0.05 for the y-axis velocity. To remove this offset, a second series of filtering needed tob e applied using the butterworth filter, this time on the velocity data:

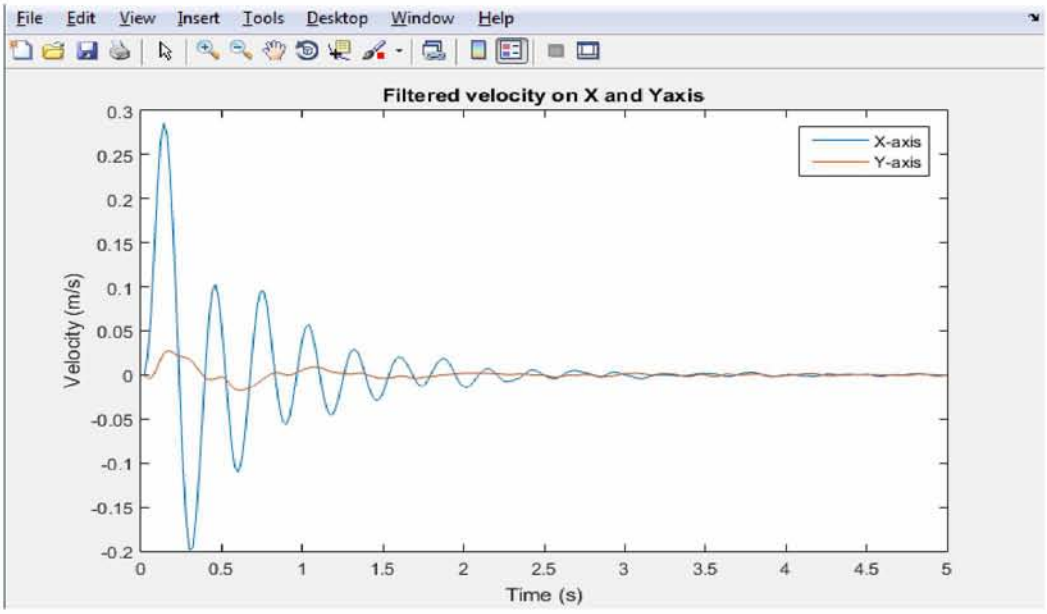

Figure 3.12. Filtered x and y-axis velocity vs. Time

The filtering process removed the offset from the velocity data (Figure 3.12). Nevertheless, some abnormanities can already by noted in the data's waveforms. First of all, although the y-axis velocity curve had a sinusoidal form, it was not perfectly smooth. In addition, the x-axis velocity

curve showed very high damping from the first to the second peak. In damped sinusoidal waves, the amplitute decreases exponentially over time, based on the formula (3.15), which is the amplitude decay term of function (3.7):

$$A = A_0 e^{(-\gamma t)} \tag{3.15}$$

If the exponential decay curve was drawn to fit the filtered velocity wave, for the x-axis, it would not fit all the peaks of the damped signal. It can therefore be deduced that there was still some residual offset in the velocity data, despite the fitering it underwent. It is highly possible that this offset is caused by the integration process. Numerical integration is a method for *approximating* integrals. More specifically, the cumulative trapezoidal rule is a technique which approximates the region under a curve as a number of trapezoids and calculates their area. Consequently, there is a certain ammount of error involved in every approximation and thus in every numerical integration process. Filtering the product of the integral does help with minimizing this error, it is not, however, able to eradicate it. This error becomes even more obvious after the next integration of velocity to position.

For that process, when using *cumtrapz(X,Y), X* was replace by the time vector while *Y* was replaced by the filtered velocity signal *velocity_X_filt* or *velocity_Y_filt*, depending on which axis the integration refered to. The results were the position data *position_X* and *position_Y.* The velocity data were in m/s units, so the acquired position data are in meters (m). In order to have measurements in cm, the *position_X* and *position_Y* matrices were multipliued with 100. Once the platform's position was calculated for both axis, the total displacement of the platform on the x-y level was computed, using the formula:

$$D_a = \sqrt{position_X^2 + position_Y^2} \tag{3.16}$$

where the index *"a"* in D stands for *"accelerometer",* as D_a is the total displacement of the platform as calculated by data obtained by using the MPU accelerometer. When plotted with respect to time, the resulting displacement data curve had a form like the one shown in Figure 3.13, with variations depending on the measurement it corresponded to:

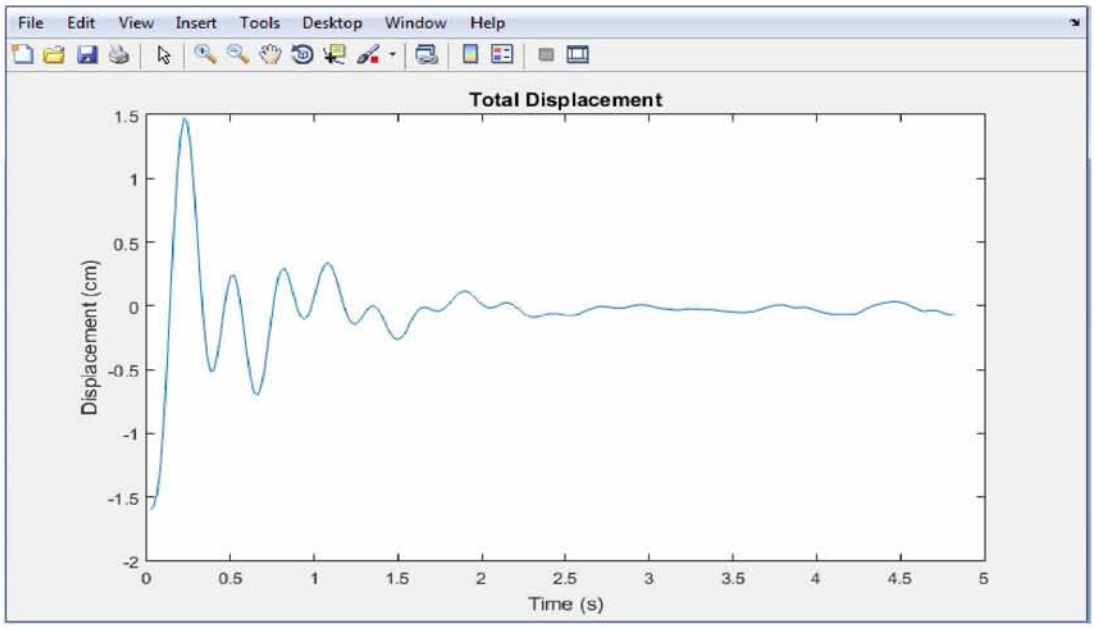

Figure 3.13. D_a vs. Time

It is obvious that the signal's amplitude decay was not exaclty exponential as, once again, its curve did not fit all the peaks of the displacement's waveform. This residual error without a doubt affected the quality of the results on the platform's total displacement, as the data curve did not exactly correspond to a damped oscillation signal, as with the previous two methods. There was, consequently, a relevant difference between the displacement data from the accelerometer and from Optritrack and Zebris, due to the error deriving from the double integration of the acceleration data. The significance of this error will become more apparent later in this study, when a comparison of the accuracy of the three methods will take place.

Despite the existing error, some positive remarks can be made on the form of our final results for this method; looking at the total displacement plot, one may observe that the platform's displacement at time t=0 seconds seems to be at -1.6-1.7 cm, which is a similar result to the one observed in the other two methods (-2 cm). Furtermore, the maximum amplitude of the signal was approximately 1.5 cm ,which is somewhere in between the maximum amplitudes for the other two methods, 2 cm for Optitrack and around 0.8 to 1 cm for Zebris. Finally, the oscillation of the displacement's signal seems to have damped significantly after the first 2 first seconds, similar to the other two methods. It can therefore be claimed that there are certain important similarities between the displacement results obtained from the accelerometer and the results from Optitrack and Zebris.

Section 3.4.2 was focused on the first approach, which consisted of performing numerical integration directly onto the discrete acceleration and time data obtain from our measurements with the MPU. In section 3.4.3, the process that was followed for the second approach will be analytically explained.

3.4.3 Curve fitting of acceleration data

During this approach, it was attempted to extract a function, *a(t),* that corresponded to the acceleration data with respect to time . The aim was to perform double intgration on a(t), so as to obtain the function of the platform's position with respect to time, *x(t)*. By relacing *t* with each time data point contained in the constructed time vector, the position (or displacement) of the platform for every sampled time *t* could be calculated.

The first part of this process was identical to the one followed for the previous approach; exactly the same steps were followed up until the filtering of the acceleration data, where the filtered acceleration signals for the x and y-axis were computed.

To calculate the function a(t), corresponding to the total acceleration of the platform, Matlab's *curve fitting tool* was used. As was briefly explained in section 3.3.1, the function of the curve fitting tool is to construct a curve which best fits the data points imported in it. When data points correspond to an acceleration versus time signal, constructing a curve matching those points and computing its mathematical formula would yield the acceleration function with respect to time that corresponds to the given data.

It is important to note that, since the aim was to calculate the function of the total acceleration of the platform, the data imported in the curve fitting tool should correspond to the total acceleration signal as well. For this purpose, the total acceleration data matrix was calculated, using the same formula that was used used for the total displacement (3.1), only this time substituing the position on the x and y axis with acceleration on the same axis:

$$Acc = \sqrt{Acc_X^2 + Acc_Y^2} \tag{3.17}$$

where *Acc* is the total platform acceleration matrix and Acc_x and Acc_y are the *filtered* acceleration signals for the x and y-axis respectively. A very significant observation need to be made at this point; using this formula, *Acc* is set as equivalent to a square root value. This means that no matter the values of the elements under the root, *Acc* is always going to be a *positive* number or, in this case, a matrix where all the cells' elements have positive values.

If one observes Figure 3.10, it's obvious that for both axis, acceleration not only take positive, but also negative values. Considering that the total acceleration signal is esentially a merging of the signals for the x and y axis, it would be irrational for it to only have positive values. In order to avoid this mistak a very simple process was followed; prior to calculating *Acc*, both acceleration curves were shifted a few units above the x axis, so that they both resided above zero.

With this alteration, Acc_x and Acc_y were now the filtered acceleration signals for both axis, shifted 5 units above. Then the total acceleration, *Acc*, was calculated using the formula (3.17). The result can be shown in Figure 3.14:

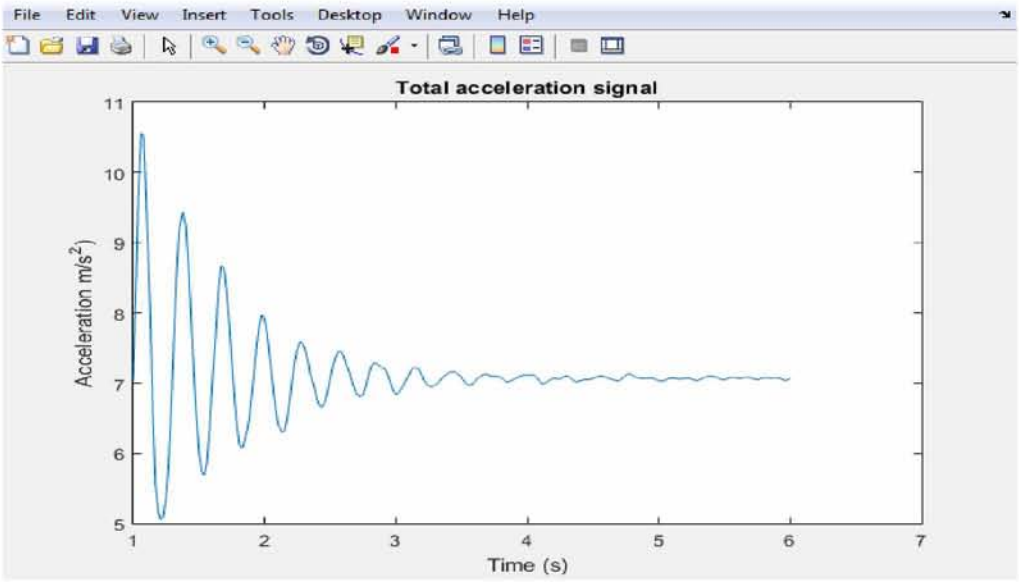

Figure 3.14. Acc_a vs. Time (7 units offset)

Naturally, all data points of the curve were values greater than zero, which made sense mathematically, but not logically. In order to remove this offset the entire curve needed tob e shifted again, this time a few units downwards, so that the total acceleration waveform oscillated around zero. The result is shown in Figure 3.15:

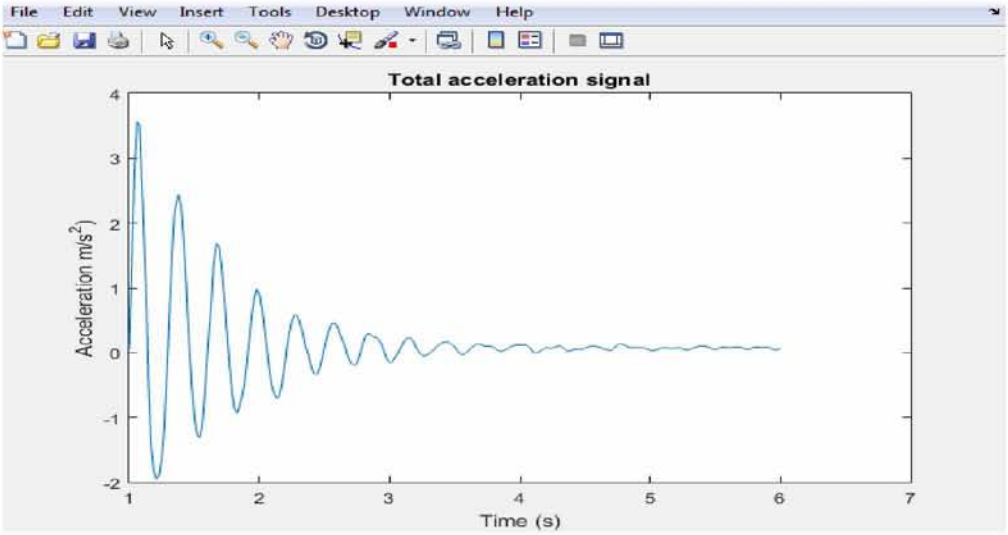

Figure 3.15 Total acceleration Acc_a vs. Time

The total acceleration data was stored in the vector *Acc_a* , which was later imported in the curve fitting tool.

A process similar to the one for the case of Zebris was followed; first, the correct vectors were imported to the tool. X data was replaced by the time vector, *Tdata*, and Y data was replaced by the vector *Acc_a*. The next step was to choose an appropriate curve fitting method from the ones available. By choosing a curve fitting method, the tool defines a function *f(x)*, which fits the data in the best possible way. *Acc_a* corresponds to a damped oscillation signal, which means that the function, *a(t)*, describing it should have the form of function (3.7).

When choosing a fitting method, the tool provides the user with the option *"custom equation"*, which allows them to set a custom function, *y=f(x),* as the curve fitting method. Knowing that the data curve corresponds to a damped oscillation signal, described by a(t), it is only rational to set *a(t)* as the fitting equation, *y=f(x),* so as to make sure that the most accurate fitting to our data points is achieved. Setting the custom equation as f(x) or f(t) doesn't make a difference as x and t are simply different names for the dependent variable of the equation. Therefore, *y* was set as:

$$y = a(x) \quad \text{or} \quad y = a(t) \tag{3.18}$$

where *a(x)* or *a(t)* was the acceleration function described in the formula (3.7).

Upon setting the desired fitting function, or custom equation, a blue line appeared in the data plot, where the data points were marked by black dots (Figure 3.16). The blue curve corresponds to the custom equation and its fitting ability to the data points is relevant to its coefficients' values. For example, if *ω* was set equal to *21* on the custom equation, it can be observed that the waveform of the equation achieved a very close match with the given data points. The goodness of the fit can be assessed by a number of statistical indicators, like the *RMSE* (root mean square error*), SSE* (sum of squared errors) or R-square (the determination coefficient), which are displayed in the "Results" box, on the left side of the curve fitting tool's window.

When no specific value was attributed to any of the remaining coefficients, *A, a* and *phi*, the curve fitting tool proceeded to automatically calculate values for the coefficients, such that the curve fitted the data the best way possible. When all coefficients had a value, then the equation (3.7)

represented the function that best corresponded to the acceleration data. As a result, integrating *a(t)* twice yielded *x(t)*, the function describing the platform's position at any given time of the perturbation. The acceleration vector the curve fitting was performed on refered to the total acceleration of the platform. Thus, *a(t)* was the total acceleration function and, as a result, *x(t)* corresponded to the total displacement of the platform. By substituting *t* with the elements of the time vector, *Tdata*, in *x(t)*, calculation of the total displacement vector for the accelerometer, *D_tot*, was achieved.

It was observed that, by setting *ω=21* and allowing the curve fitting tool to calculate the rest of the coefficients, the tool yielded the best match to the acceleration data. Of course, with every different measurement of the platform's acceleration, the matrices *Acc_a* and *Tdata* contained different values and thus, the resulting acceleration function, as calculated by the curve fitting tool, was different each time. In order to perform numerical integration of the acceleration function to obtain the displacement (or position) function, the full form of *a(t)* was required , which meanst that values needed to be attributed to all coefficients. The integration took place using Matlab's command "*int(f, x)*", where f is independent and x is the dependent variable of the acceleration equation (3.7), with *f(x)* instead of *a(t)* and *x* instead of *t*.

Every time the curve fitting tool was run for each data set, the custom functions' coefficients' values were displayed under "Results" of the tool's user interface. Theoretically, these values could be copied from the "Results" window and replaced with *A, a* and *phi* in *f(x)*. This approach, however, would be highly impractical as it would require running the tool for each individual data set. This would require to importing the data matrices and re-setting the custom equation for each and every measurement, a process which would be extremely time-consuming. To avoid this, the same process as with Zebris was followed; a Matlab code was generated from the tool, which performed a curve fit on the acceleration data of each measurement and calculated the coefficients' values. The latter was achieved by using the command *coeffs()* with the number corresponding to each coefficient in the brackets. The coefficients were numbered with the order in which they appear in the custom equation, so *A=coeffs(1), a=coeffs(2)* and *phi=coeffs(3)*. These values were then added in the acceleration formula set at *f_x* in the Matlab script, which was then integrated twice using the command *int(f,x)*, in order to yield the velocity and finally the position function, *x(t)*.

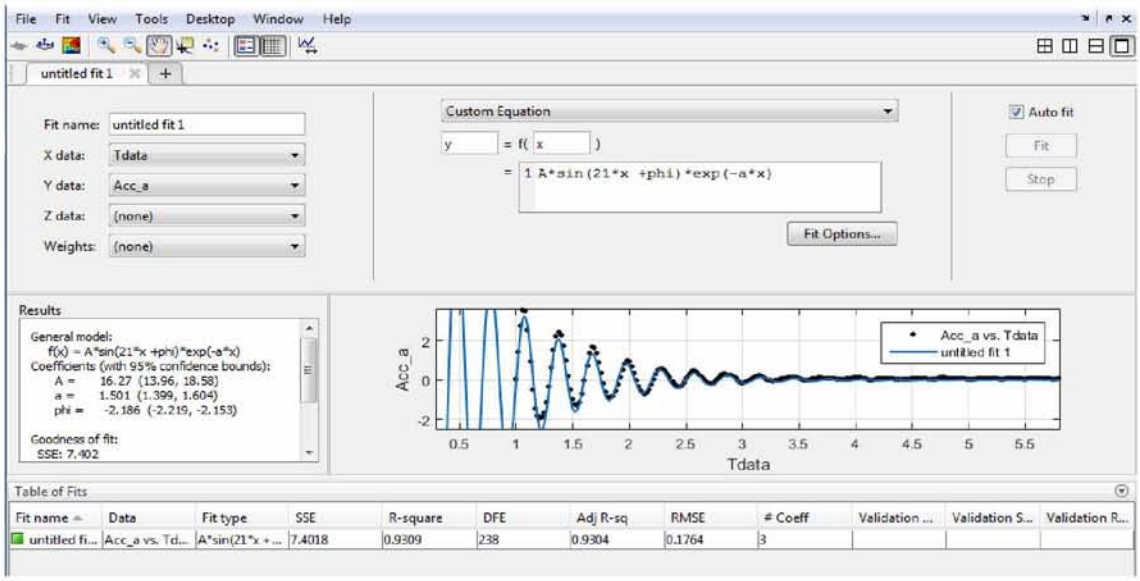

Figure 3.16. Custom equation curve fitting on acceleration data

The final step was to construct the total displacement matrix. This was achieved by using a loop command, where the variable *x* of the position function, was replaced by an element of the time matrix, *Tdata,* on each run, so x_i=*Tdata(i)*, where *i* was the cell number of the time vector. For each value of x_i the value of the position function was calculated and stored it in the vector *D_tot*, which contained the values for the total displacement (or position) of the platform.

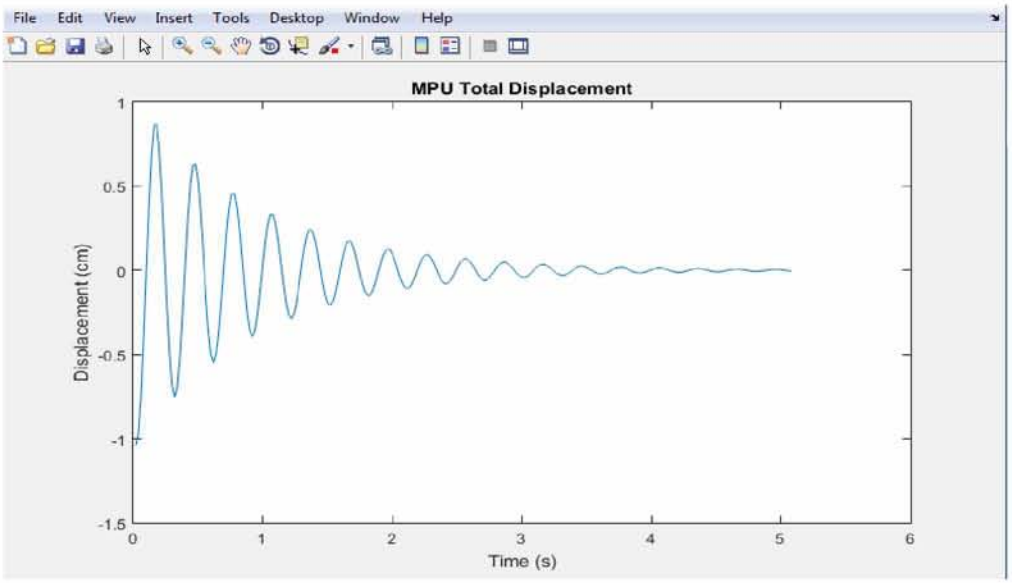

Figure 3.17. Total displacement D_{tot} vs.Time – MPU-9150

This step concluded the motion data processing with the purpose of acquring data for the Posturomed platform's displacement. The next section is devoted to the strategy followed for the comparison of the three methods, Optitrack, Zebris and the MPU accelerometer, based on a number of factors such as their accuracy, user-friendliness, cost, effectiveness etc.

4. Comparison of methods-Accuracy

4.1 Comparison of accuracy

The term *"accuracy"* refers to the relative difference between the measured position of an object, when using an electronic device, to its actual position in space, given a certain point of reference. The smaller this difference is, the higher the measurement accuracy. Naturally, since there is a certain ammount of error or inaccuracy involved in every DAQ mechanism it has currenly been impossible to achieve 100% accuracy. Inaccuracies of electronic motion capture devices derive from a number of factors such as programming imperfections, digital signal noise, signal processing errors etc.

Optitrack is the most widely used method for motion capture and position measurement applications. It offers measurements of excellent quality as it can achieve an accurcay of sub-millimeter scale, with optimal capture volume size and camera configurations. As was mentioned in chapter 1, however, one of the drawbacks of this particular method is its high cost, raising to thousands of dollars. Being able to identify a method for position measurements, offering similar accuracy to Optitrack, for a fraction of the cost, would have immense benefits.

Over the course of this thesis, it has been attempted to compare the accuracy of the Zebris CMS10 device and the MPU accelerometer to that of Optitrack, with the aim of investigating whether these two methods could be used for position measurement applications as effectively as Optitrack, with respect to the level of accuracy they provide. For this purpose, Optitrack was used as the reference method, which means that the position results extracted from Zebris and the MPU where compared to the ones extracted from Optitrack in order to identify whether their difference was statistically significant.

Two different strategies were followed in order to examine the significance of the statistical difference of the methods' results. For both approaches, statistical comparison of the methods was conducted based of the results extacted from relevant statistical tests. Before explaining the approaches, the next section will be devoted to explaining the significance and function of certain types of different statistical tests for the purpose of statistical comparison of data.

4.2 Statistical tests for data comparison.

4.2.1 T-tests

A *t-test*, is a comparison of two populations' means values, through the use of statistical examination; it assesses whether the means of two groups of data are statistically different and how significant that difference is. A *statistically significant* t-test result is one in which a difference in means (averages) between two groups is unlikely to have occured because the sample happened to be atypical; in other words, it is unlikely to have occured by chance.

Statistical significance is determined by the value of the *t-statistic (or t-value)* and its distribution (Figure 4.1). The *t-statistic* evaluates the difference between the groups' means relative to the spread of the variability in the two groups:

$$t - statistic = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

where $\bar{x_1}$ and $\bar{x_2}$ are the means of the two populations (or samples), s_1 and s_2 are the standard deviations of the samples and n_1 and n_2 are the sample sizes. The probability that the t-statistic assumes a certain value can be presented in the normal t-distribution graph:

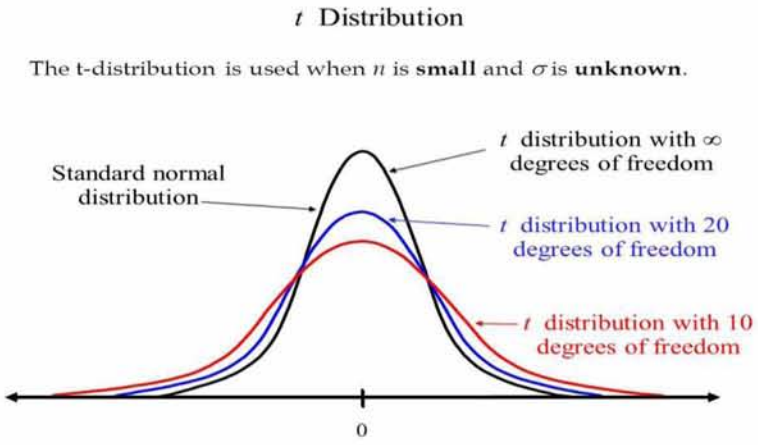

Figure 4.1. T-distribution with different degrees of freedom

where the x-axis corresponds to the t-statistic's values and the y-axis corresponds to the possibility that the t-statistic has each relevant value.

Whenever one wishes to apply any type of statistical test on a certain set of experimental data, there is the need to pose the question they seek the answer for in a statistical form. That means that they need to formulate a *hypothesis* about their data, which will be either confirmed or rejected, depending on the results of the test. Any data set, or set of measurements, represents a sample from the population of all possible results. It's only natural that, for every set of replicate measurements, there will always be a certain variation in the mean value of each set. During a statistical test, we need to establish whether or not the data sets under comparison are drawn from the same or different populations. This can be be expressed by the following hypothesis, *H*:

$$H: \mu_1 = \mu_2 \tag{4.1}$$

which suggests that two sets of data with means μ_1 and μ_2 are both part of the same population and, consequently, their means are equal.

One very important concept in statistical tests is that of *the null hypothesis, H_0*. Interpretations of the term may vary, depending on the test type, but the main idea is that, the measurements' values belonging to the sample are the result of pure chance when taking the sample measurements from the population of all possible values. Upon formulating the null hypothesis, one also has to formulate the alternative hypothesis H_a. If, for example, the chosen null hypothesis states that there is no difference between the populations' means, then the two hypothesis are often stated in the following form:

$$H_0: \mu_1 = \mu_2 \text{ and } H_a: \mu_1 \neq \mu_2 \tag{4.2}$$

These two hypothesis must be mutually exclusive; if the null hypothesis is confirmed then the alternative hypothesis must be rejected, and vice versa.

Whether or not the null hypothesis is confirmed or rejected depends on the value of the t-statistic of the samples and, more specifically on the value of the probability, *p*, that the t-statistic resides in a specific region of the t-distribution graph. Let us assume that, for two samples with means μ_1 and μ_2 the t-statistic was calculated to be equal to 2. On the t-distribution graph, the x-axis represents the values of the t-statistic (Figure 4.2). In a *two-tailed* t-test, the t-statistic defines two regions, one on the positive and one on the negative direction of the graph. For t=2, in a two-tailed test, those regions will consist of the area under the curve for which *t<=-2* as well as the area for which *t>=2*:

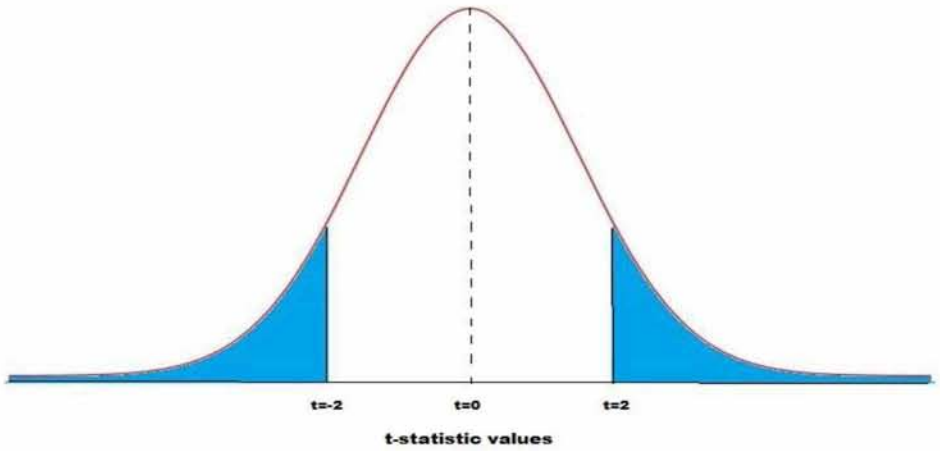

Figure 4.2. T-distribution graph

The *p-value* is defined as the probability of the t-value residing in one of those regions or, in other words, the t-value belonging to the following range of values:

$$t \in (-oo, 2) \cup (2, +oo) \tag{4.3}$$

and thus:

$$p = P(t \leq -2) + P(t \geq 2) \tag{4.4}$$

where *P* defines the probability of t possessing a certain value.

The null hypothesis will be rejected if *p* is lower than a critical value, while it will be confirmed if p is higher than said value. The critical value of *p* depends on the *confidence interval, a.* When a series of measurements follows a normal distribution, most measurements will reside around the mean value, with fewer measurements residing away from the mean. This phenomenon can tell us that if a certain value is away from the mean, then the chance of getting this value by chance in a sampled measurement is small. As a result, if this value appears in a number of repetitive measurements then it's possible that we are measuring a system with a different mean value.

Let us for example suppose that we have a system with mean μ, standard deviation σ and size n. Now let us suppose that one of the experimental values, b, only occurs with a probability of less

than 0.05 or 5% of the time. In other words, if we took repetitive measurements of a system with the same characteristics (μ,σ and n) there would be a 5% chance that we would get a result near b. If many values close to b appeared in the same tests, this would mean that for some reason, the system we measured has changed while we were working and so that means that, ultimately, we measured a different system to the one we started with. In that case, we would be 95% confident that the value of b belongs to a different system. This is called a *95% confidence level or interval*.

In the t-distribution graph of a 95% confidence level, the shaded area under the curve contains all the values of t within 95% of the central mean value:

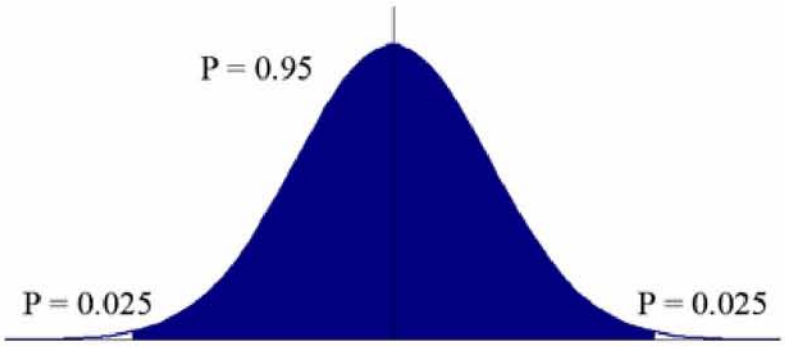

Figure 4.3. Probability regions in a 95% confidence interval

This area represents a 95% probability that the values of t will reside within a range of values on either side of the mean, up to the limits of the shaded area. We can note that there is a small probability of 0.025 (or 2.5%) that t will exceed the upper limit and the same probability (2.5%) that is will be lower than the lower limit. Therefore we can say that there is a 5% probability in total that t will not reside within the limit on either side of the central mean value. Remember that the value of *p* gives us the probability of t residing in a specific area under the curve of the normal distribution. We can deduce that, if *p* is smaller than 0.05 or 5%, then this means that *t* does not reside in the area defined by the limits of -0.025 and 0.025. Consequently, there is a 95% chance that *t* belongs to a system with a different mean and therefore the null hypothesis ($\mu_1=\mu_2$) is rejected. The conclusion then is that there is a significant statistical difference between the means of the two sample groups.

There is a number of types of t-tests which can be performed on a group or a number of groups. A ***one-sample*** *t-test* is a tests where we seek to examine whether the mean of a sample is significantly different to a certain value (often zero). A ***two-sample*** *t-test* examines whether or not there is a significant statistical difference between the means of two different groups.

The t-test type also depends on the relationship between the data contained in the samples or groups. In ***paired*** *t-tests*, the data from the groups is linked in some way, whilst in ***unpaired*** *t-tests*, the data from the groups is not linked. For example, if the samples are taken from the same population then the data in all groups of samples are obviously connected in some way as they derive from the same origin. In that case, if one wishes to examine whether the samples are significantly different, a *paired* t-test is the appropriate method to follow.

4.2.2 Kolmogorov-Smirnov test

As was mentioned in the previous section, one of the requirements of a t-test is that the data group(s) participating in it follow a normal distibution. A standard t-test assumes that the sample is normally distributed around a mean value of zero. It is important to remembed that, in this particular study, data derived from sampling of a damped oscillation signal. That is to say that the displacement data is not random, but rather refer to an object's motion described by a specific mathematical equation, that of a damped oscillation. It would, therefore, be rather arbitrary to assume that the data follows a normal distribution without previously testing the validity of this hypothesis.

A *one-sample Kolmogorov-Smirnov test* returns a decision fort he null hypothesis that the data in a vector *x* comes from a standard normal distribution. Matlab's function:

$$[h, p] = kstest(x) \tag{4.5}$$

provides users with a tool to test whether or not the data contained in vector *x* follows a normal distribution. For a standard 95% confidence interval, the null hypothesis is confirmed *(h=0)* when p<0.05 or 5%, and rejected *(h=1)* when p<0.05. A confirmation of the null hypothesis would signify that the x-vector data follows a normal distubution and, as a result, it would be scientifically correct to conduct a t-test on the same vector.

4.2.3 Wilcoxon signed rank test

Rejection of a ks-test's null hypothesis would signify that the t-test requirement for normal distribution of data is not fullfilled. In this case, statistical comparison of data should be carried out with a different type of statistical test which does not have the requirement for normal distribution of data, such as the *Wilcoxon signed rank test.* In Matlab, the function:

$$[p, h] = signrank(x) \tag{4.6}$$

returns a decision for the null hypothesis that the data in vector *x* comes from a distribution with a median equal to zero, at a 95% confidence interval. The test assumes that the data in *x* come from a continuous, but not necessarily normal, distribution symmetric about its median.

4.3 Statistical comparison of methods

As was explained at the end of section 4.1, the levels of accuracy of position measurements obtained from the Zebris CMS10 device as well as the MPU accelerometer were explored, compared to the data extracted from Optitrack, which was set as the point (or method) of reference. For this purpose, statistical tests were employed in order to examine the significance of the statistical difference between the three methods. Zebris CMS10 and the MPU were individually compared with Optitrack, the method of reference. Thus, two separate tests were performed; in the first one, statistical comparison was performed with data extracted from Optitrack with respect to data extracted from Zebris while for the second one, the comparison was carried out with data extracted from Optitrack with respect to the accelerometer's data instead.

In order to compare the methods' accuracies to each other, two different approaches were followed; for the first approach, the significance of the the *root mean squared error*, or *RMSE*,

values between the position data of the three methods was examined. For the second approach, the statistical significance of the calculated difference between the displacement values acquired by the three devices was tested, using the Optitrack data as reference. In sections 4.3.1 and 4.3.2, the two approaches are analyzed in detail.

4.3.1 First approach: Statistical tests on RMSE values

In statistics, the ***root-mean-squared error (RMSE)*** is a frequently used measure of the *predicted* values of a sample (or population) and the values which are actually *observed*. In other words, it estimates the *sample standard deviation* of the differences between the observed and the predicted values, the prediction errors. When talking about estimators or prediction models, *RMSE* is a measure of accuracy to compare forecasting errors of different models.

RMSE is calculated as the square root of the average of the squared error, where the error can be described as the difference between two samples of forecasted values, f_j, and observed values, o_j :

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(f_j - o_j)^2} \tag{4.7}$$

where n is the size of the samples f_j and o_j. The smaller the value of RMSE, the smaller the difference between the predicted and observed values and thus, the smaller the prediction error.

RMSE can be used for the purpose of computing the deviation of one set of data (i.e the predicted valued) from another (i.e the observed values), or the difference between the values of the two data sets. It has become obvious by now that no prediction models have been used to calculate position data for any of the three methods. What has been attempted was to calculate the position data acquired by three devices and evaluate how different these results turned out to be. Essentially, the aim was to calculate is the deviation of one position data set from another, which can be expressed in the form of an RMSE where, instead of the forecasted and observed values, we have the difference between the displacement data of two methods. Therefore, using the displacement vectors created in the Matlab code, two RMSE values were calculated:

$$RMSE_1 = \sqrt{\overline{((D_m - D_a)^2)}} \quad (4.8a), \quad RMSE_2 = \sqrt{\overline{((D_m - D_{Zeb})^2)}} \quad (4.8b) \qquad (4.8)$$

where *D_m, D_a* and *D_Zeb* were the displacement vectors for the Optitrack, the MPU and the Zebris device respectively. It is important that the vectors in each formula have the same size. When looking at the position graphs constructed from the displacement vectors, $RMSE_1$ is practically a measure of the distance between the curves of Optritrack and Zebris data, while $RMSE_2$ is a measure of the distance between the curves corresponding to the data of Optitrack and the accelerometer.

It is worth mentioning that for both approaches, displacement data acquired from the accelerometer contained in the vector *D_a* was used, instead of *D_tot*. *D_a* is the displacement vector constructed through the *numerical integration* process analyzed in section 3.4.2.4. There are a couple of reasons behind this differentiation; first of all, the numerical integration process with the aim of acquiring position data, was much less tedious than the curve fitting method and thus using *D_a* saved us a lot of processing time. In addition, the position graphs constructed from both *D_a* and *D_tot* showed that *D_tot* achieved much smoother results than *D_a.* As a result, the

curve corresponding to the *D_tot* data acquired by the accelerometer had a very similar form to the data curves for the other two methods, whilst the curve deriving from *D_a* data showed some fluctuations and anomalies, caused by the successive filtering processes.

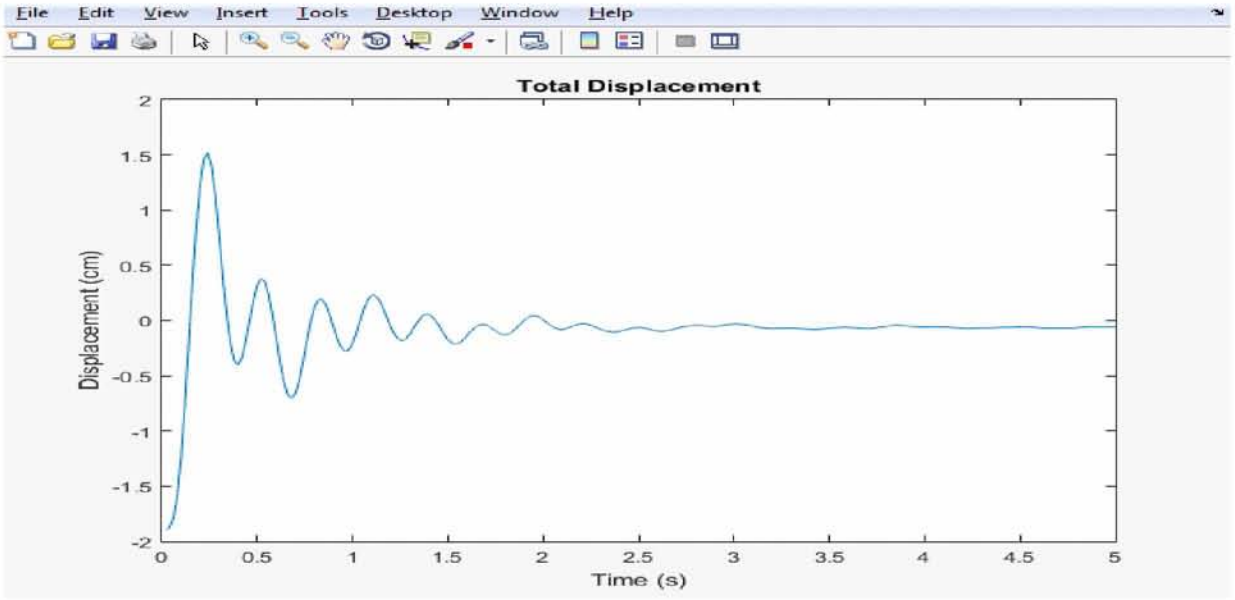

Figure 4.4.a. D_a vs. Time graph

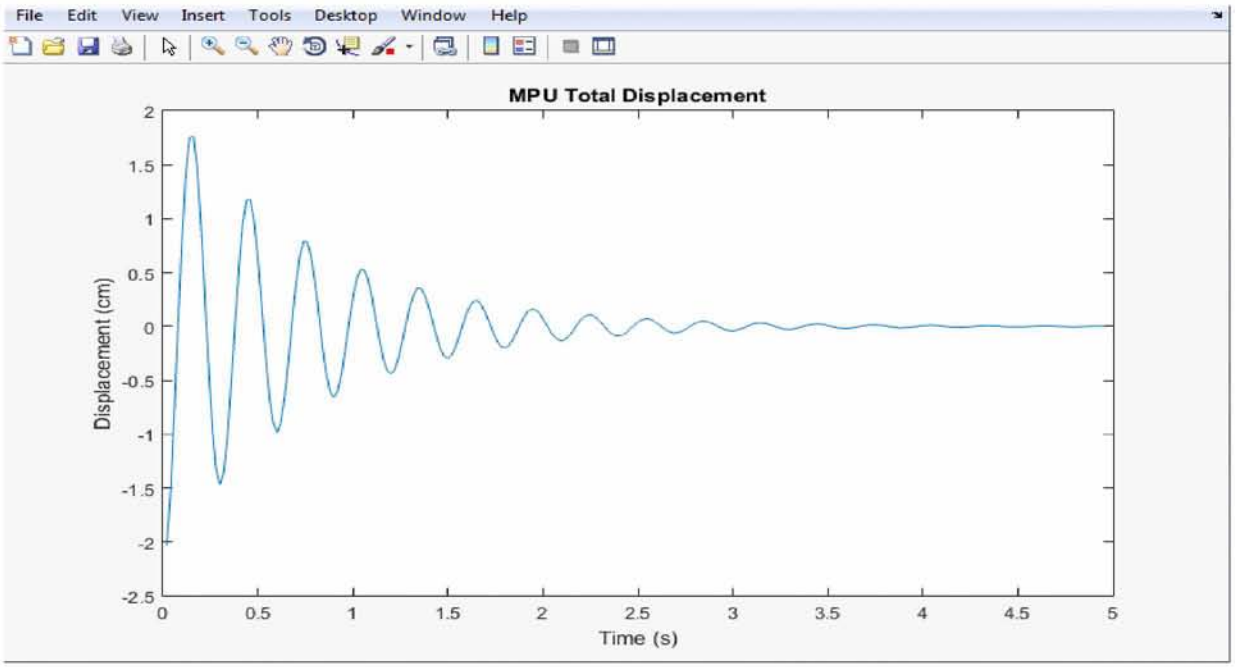

Figure 4.4.b. D_tot vs.Time graph

Figure 4.4. Displacement vs.Time graphs – MPU-9150

Consequently, we based our decision on the hypothesis that, if the t-tests yielded satisfactory results when comparing *D_a* with *D_m*, one could conclude that it would also yield similar or even better results when using *D_tot* instead, since *D_a* data seemed to be of poorer quality that that of *D_tot*.

For each measurement of the platform's motion, each of the three methods sampled a discrete-time series of position data using its DAQ mechanism. These sampled data were processed by the Matlab script and yielded three position (or displacement) vectors, one for each method, for each measurement of the platform's oscillation. Using the formulas 4.8a and 4.8b, $RMSE_1$ and $RMSE_2$ were calculated, using Optitrack's data vector as a common point of reference. These two values were saved in an excel file in two separate columns, so that every row of the file corresponded to the $RMSE_1$ and $RMSE_2$ values of a different measurement of the platform's motion. When the code for all measurements, each column of the final excel file represented a distinct data set (or vector) of RMSE values.

The two data columns $rmse_1$ and $rmse_2$ contained the values of $RMSE_1$ and $RMSE_2$ which described the difference between the position data of Optitrack and the MPU ($RMSE_1$) and the difference between the position data of Optitrack and Zebris ($RMSE_2$) respectively. In other words, $RMSE_1$ and $RMSE_2$ show the deviation of the Zebris and MPU position data from that acquired by Optitrack, which is the method of reference.

Calculating the RMSE between the two pairs of methods provided a measure of the difference between the sampled positions during measurements of the Posturomed platform's motion. It did not, however, provide information on how significant this difference is. Of course, the smaller the RMSE is ,the smaller is the deviation between the data sets included in the RMSE formula (4.8). Despite how small its value is, there is still no information included in the RMSE itself on whether or not this value shows a significant difference between the two sets of data.

In order to investigate the significance of the RMSE values, a series of statistical tests was performed; the first step was to check the normality of the distributions of the $RMSE_1$ and $RMSE_2$ vectors. This was achieved by carrying our two *one-sample ks-tests,* using the formula (4.5), where *x* was replaced by each of the $RMSE_1$ and $RMSE_2$ vectors respectively. *normh* was the test's decision for the null hypothesis for normality, while *normp* was the corresponding p-value. A confirmation of the null hypothesis *(normh=0)* would signify that the data contained in *x* followed a normal distribution around a mean value of zero, which would be the case for *normp*>0.05 or 5%. In this senario, with the requirement for a normal distribution fullfilled, one could proceed with carrying out one -sample t-tests on our RMSE values, to determine their statistical significance. Two t-tests could be carried out, where the x-vector in the formula (4.5) would be replaced by the vectors $RMSE_1$ and $RMSE_2$ respectively. Confirmation of the t-test's null hypothesis *(h=0)* would mean that the data contained in the participating vector *x* is not significantly different from the mean value of 0. In other words, it would signify that the respective RMSE value is insignificantly different from zero.

In case, however, that the data did not follow normal distributions *(normh=1),* it would be incorrect to proceed with t-tests. Instead, *one-sample signrank tests* on the RMSE vectors would be more appropriate. Once again, confirmation of the *Wilcoxon signed rank test's* null hypothesis *(h=0)* would mean that the data contained in the participating vector *x* is not significantly different from the mean value of 0. Rejection if the null hypothesis on the other hand (h=1) would lead us to the opposite results, where the *x*-vector data, or the RMSE values, is significantly different frm zero.

4.3.2 Second approach: Statistical tests on displacement differences

In section 4.3.1, the process followed in order to investigate the statistical significance of the RMSE

values calculated for the 3 method (with Optitrack used as a method of reference) was thoroughly described. While those statistical tests yileded results on the relationship of the RMSE values of the three displacement data sets, they provided this information neglecting the fact that all the acquired displacement data was *time-dependent*.

In a standard statistical test, the sequence of the samples in the participating data groups is *not* taken into account. However, this is an important omission when dealing with a time-dependent signal, such as the signal of a damped oscillation. To make up for this omission a statistical test which would not exclude time from its parameters should be used. This could be achieved by making sure that the data groups participating in each test included position data referring to the same moment of the oscillation or in other words, to the same point in time.

In the experiments carried out for thie master thesis, the Zebris CMS10 device as well as the MPU accelerometer had a sampling rate (or frequency) of 50Hz, so samples were acquired every 0.02 seconds in both methods. By processing the signals acquired from these two devices the platform's displacement values were extracted and saved in two separate excel files, in the form of two vectors*, D_a* and *D_Zeb*, where *a* refered to the accelerometer and *Zeb* to Zebris. Both vectors refered to the same points in time, as the samples contained in them were acquired with the same frequency (50Hz). This meant that, the element of the first cell in both files corresponded to time t_0= 0.02 s, while the element of the second cell corresponded to time t_1=0.04 s, and so on. However, this was not the case for Motive's displacement data, as its sampling frequency was not 50, but 120 Hz. The displacement data acquired by Motive, contained in the D_m vector did not correspond to the same points in time as the data in the D_Zeb and D_tot vectors.

In order to ensure that all data points for all three methods were sampled with the same frequency, the position data acquired from Motive needed to be *downsampled*, so as to only save data occuring with a frequency of 50 instead of 120 Hz.

Downsampling the data was achieved using the Matlab function :

$$downsample(x, n, phase) \tag{4.9}$$

which decreases the sampling rate of *x* by keeping every *nth* sample starting with the first one. *Phase* is an integer from *1* to *n-1* specifying the number of samples by which the function offsets the downsampled frequency. In out Matlab script, downsampling was performed twice, with the *x* vector being replaced by the time vector *Tdata_m* and the displacement vector *D_m* respectively.

As this method uses approximation, the resulting frequency was not exactly 50 Hz but a slightly higher value, which consequently created a small offset in the resulting displacement values, as they corresponded to a slightly higher frequency than the desired one. Naturally, this offset caused a certain degradation of the statistical tests' accuracies, which is important to mention when optimization of the statistical comparison of the methods is the goal. Once the downsampling process was complete, the displacement vectors for all three methods corresponded to (approximately) the same time points.

For the second approach, *44* sets of displacement measurements were used, each in a form of an excel file containing a *nx3* matrix with *n* ranging from 160 to 260. Each column corresponded to the displacement values acquired by one of the three devices, the accelerometer, Optitrack and Zebris, following this exact order. Each row corresponded to one particular moment in time of the

platform's oscillation, so the values of the three cells in each row referred to the platform's displacement from its resting point in that particular sampling time, as calculated by the three different devices. The sampling frequency was *50Hz* for all 44 sets of measurements.

The idea behind this approach was rather simple; if the difference between the methods' displacement results is insignificant, then all three methods can offer similar or practically equivalent accuracies of measurements in position measurement applications. Since Optitrack was used as the ground method (or method of reference) it made sense to compare its displacement results with those of the other two methods and identify the significance of their differences.

In order to calculate these differences of displacement values, the displacement values of the MPU accelerometer and the Zebris CMS10 device were subtracted from Motive's displacement data, for every row of each measurement file or, in other words, for each sampled time-point. In addition, to provide a supplementary comparison, the differences in displacement values between Zebris and the accelerometer were also calculated, following the same process only this time subtracting MPU's from Zebris' displacement values for each sampled time-point. These differences were calculated using a simple Matlab code, where each measurement was imported in a *nx3* matrix, *data(i,j)*. The 3 differences ,*dif*, were then computed, by performing subtractions between the 3 columns of every file. The first column corresponded to the data acquired by the accelerometer, the second to data acquired by Optitrack and the third by Zebris:

Motive and MPU: $dif_1 = data(i,2) - data(i,1)$ (4.10a)

Motive and Zebris: $dif_2 = data(i,2) - data(i,3)$ (4.10b)

Zebris and MPU: $dif_3 = data(i,3) - data(i,1)$ (4.10c)

where *i* is the row number.

dif_1, dif_2 and dif_3 were then saved in a separate excel file, which eventually contained all the calculated differences for all measurements, resulting in an excel file with a dimension of *4809x3,* each column containing the values of dif_1, dif_2 and dif_3 respectively. These values were either positive or negative, fluctuating around zero. Proving that dif_1, dif_2 and dif_3 were insignificantly different from zero would prove that the difference between the displacement values participating in the formulas of dif_1, dif_2 and dif_3 were not significant themselves.

In order to test for the normality of the samples dif_1, dif_2 and dif_3 we needed to conduct a *Kolmogorov-Smirnov test*, or *ks-test*. A one-sample Kolmogorov-Smirnov test returns a decision fort he null hypothesis that the data in a vector *x* comes from a standard normal distribution. Using function (4.5) and replacing x with each of the three vectors dif_1, dif_2 and dif_3 respectively, the test's decisions for the null hypothesis were acquired for each of the three one-sample ks-tests. As the standard value of 95% for the confidence interval was used, the null hypothesis was confirmed *(h=0)* when p>0.05 or 5%, and rejected otherwise.

A confirmation of the null hypothesis would signify that the data contained in the *x*-vector followed a standard normal distribution, which would therefore allow proceeding to conducting a t-test for the aforementioned vector. A rejection of the null hypothesis *(h=1)*, on the other hand, would not permit carrying out a t-test, as its requirement for samples of normal distribution would not be fullfilled. In that case, a t-test would not be a suitable method for exploring the statistical

significance of the deviation of the values in the *dif* vectors from zero. Instead, a type of statistical test which does not require samples to follow a normal distribution should be used, such as the *Wilcoxon signed rank test*. The signrank-test could then be carried out using the function (4.6) and replacing x with each of the three vectors dif_1, dif_2 and dif_3 respectively.

5. Results

5.1 Statistical comparison of position data

The comparison of accuracy of the three position measurement methods (or devices), Optitrack, Zebris and MPU accelerometer, aimed at investigating whether or not all three methods could be used for position measurement applications, without affecting the accuracy of the acquired results. More specifically, since Optitrack is the motion capture method which is more widely used because of its high accuracy, the goal was to determine whether this device could be substituted by either Zebris CMS10 or the MPU accelerometer for the same applications, with an insignificant difference in accuracy.

For this purpose, statistical tests were conducted with the aim of investigating the significance of the difference between the displacement results (or data) acquired by the three devices, during the Posturomed platform's perturbation tests.

5.2 Results: first approach

Two *one-sample ks-tests* were performed on the calculated $RMSE_1$ and $RMSE_2$ vectors, in order to examine the normality of their distributions. The results of the ks-tests are demonstrated on Table 5.1:

Vectors	*normh*	*normp*
$RMSE_1$	1	1.715125479251664e-51
$RMSE_2$	1	1.880664754450035e-50

Table 5.1. ks-test results for $RMSE_1$ and $RMSE_2$

normh was the test's decision for the null hypothesis for normality, while *normp* was the corresponding p-value. It can be observed that both p-values are smaller than 0.05, which is the standard significance level. Thus, the null hypothesis for normality of the data samples is *rejected (normh=1)* for both data vectors.

The next objective was to determine the statistical significance of the deviation of the two separate RMSE vectors from zero. In other words, the statistical hypothesis was that the data in vectors $RMSE_1$ and $RMSE_2$ respectively, derived from distributions with mean equal to zero (and unknown variance). For this purpose, two *one-sample signrank-tests* were used. Performing t-tests was not an option since both data vectors did not follow normal distributions. The signrank-tests' results are presented in Table 5.2:

	h	*p*
$RMSE_1$	1	1.682934727529787e-28
$RMSE_2$	1	1.682827635694589e-28

Table 5.2. Signrank-test results for $RMSE_1$ and $RMSE_2$

Once again, one can observe that both p-values were smaller than 0.05; this lead to the *rejection (h=1)* of the null hypothesis, for both data samples ($RMSE_1$ and $RMSE_2$). Consequently, the ks-tests' decision were that both samples derived from distributions with mean values significantly different from zero. In other words, the RMSE values contained in the two vectors deviate from zero significantly themselves.

5.3 Results: second approach

The calculated differences in displacement values, dif_1, dif_2 and dif_3, were saved in three nx1 vectors. In order to test the normality of the vectors' data distributions, three *one-sample ks-tests* were run, the results of which may be observed in Table 5.3:

Vectors	*normh*	*normp*
dif_1	1	0
dif_2	1	0
dif_3	1	0

Table 5.3. ks-test results for dif1, dif2 and dif3

All the calculated values of the p-statistic (normp) were calculated to be zero, which is smaller than the significance level of 0.05. Therefore, the null hypothesis for normality of the distributions was rejected *(normh=0)* in all three cases.

Since the t-test's requirement for samples following normal distributions was not met, three subsequent *one-sample signrank-tests* were conducted for each vector (Table 5.4):

Vectors	*h*	*p*
dif_1	1	1.461594939875593e-04
dif_2	1	8.579103177522811e-15
dif_3	1	1.429007243216107e-19

Table 5.4. signrank-test results for dif1, dif2 and dif3

The signrank-tests' p-values were calculated to be smaller than 0.05 for all three cases (or data sets), resulting in the rejection *(h=0)* of the null hypothesis for all vectors (dif_1, dif_2 and dif_3). These results lead to the deduction that the data in vectors dif_1, dif_2 and dif_3 is derived from distributions with mean values significantly different from zero, which subsequently means that the calculated differences of displacement values are significantly different from zero themselves.

A standard 95% confidence interval was used for all statistical tests. The tests' results seemed tob e unaffected when using lower confidence intervals.

6. Discussion

For the purpose of this master thesis, three different methods were tested on position measurements: optical motion capture (Optitrack), ultrasonic transmitter-receiver systems (Zebris CMS10) and MEMS accelerometers (MPU-9150). In this section, remarks are made on the methods' accuracies based on the results extracted by the statistical comparison of the displacement data acquired by the three devices, which was thoroughly explained over section 5. Furthermore, the advantages, disadvantages and characteristics of each device are analyzed, taking into account their accuracy, user-friendliness, versatility and cost.

Overall, the comments stated in this section, will be based on observations made throughout the course of the followed experimental process as well as on information collected upon research on the characteristics of each device.

6.1 Accuracy

The current master thesis was centered around the comparison of three devices for the purpose of position measurements, particularly in the field of biomechanics and biomedical applications. One of the main goals of this study was to identify whether or not the measurement accuracies of these three devices were comparable and to what extent.

Optitrack is a 3D motion capture system, which employs infra-red cameras to detect and track motion. It is often used in the field of 3D human motion studies, such as gait and movement analysis as well as in a number of other areas such as robotics and animation. Zebris CMS10, a system designed for the analysis of 3D motion patterns, is yet another powerful device which has been present for over 20 years in the field of biomechanical studies. The MPU-9150 accelerometer is a MEMS sensor, a portable device which has gotten more and more popular in recent years in applications revolving around mechatronics and low-invasive human motion tracking. Despite their practicality and advantages, MEMS sensors are especially susceptible to measurement errors (noise, offset, drift etc.) which need to be eliminated for the effective use of such devices for position measurements.

6.1.1 Statistical test results

The online research conducted on currently available motion capture methods showed Optitrack as the most widely used one, with a level of accuracy at sub-millimeter scale. Therefore, for this master thesis, it was accepted as the method of reference and the accuracy of the other two devices, the MPU and the Zebris CMS10, was assessed with respect to Optitrack.

The comparison of the method's accuracies was based on the results extracted from a series of statistical tests, conducted according to the two different approaches analyzed in chapter 4. More specifically, it was investigated whether or not the difference between the displacement results

acquired by the MPU and Zebris was statistically significant, with respect to results yielded by Opitrack.

For the first approach, two one-sample ks-tests (Table 5.1) and two one-sample signrank-tests (Table 5.2) were run on the $RMSE_1$ and $RMSE_2$ values calculated between the Optitrack and MPU displacement data and the Optitrack and Zebris data respectively, as described by formulas (4.8). The results of the signrank-tests, which can be seen on Table 5.2, showed that both $RMSE_1$ and $RMSE_2$ data sets contained RMSE values significantly different from zero.

Looking at formulas (4.8a) and (4.8b), it is obvious that both RMSE values are equal to the square root of a squared, absolute difference between two vectors, *D_m* and *D_a* for $RMSE_1$ and *D_m* and *D_Zeb* for $RMSE_2$. Since $RMSE_1$ and $RMSE_2$ are significantly different from zero (based on the signrank-tests' decisions), then the respective square roots, and therefore the differences *D_m-D_a* and *D_m-D_Zeb*, are also significantly different from zero. Consequently, this means that there is a (statistically) significant difference between the displacement vecotrs D_m and D_a and D_m and D_Zeb respectively. These results can be translated into deduction that the displacement data acquired by both the MPU accelerometer and Zebris deviate from the data acquired by Optitrack by a statistically large margin. Thus, the measurement accuracies of the MPU and Zebris differ significantly from the accuracy of Optitrack, which was used as the method (or device) of reference.

It is important to remind the reader that, for the first approach, the relationship of displacement and time was *not* taken into consideration, which is a factor that could have affected the accuracy of the performed statistical tests and therefore, the accuracy of the decuctions deriving from their results.

For the second approach, three one-sample ks-tests (Table 5.3) and three one-sample signrank-tests (Table 5.4) were conducted on the data vectors dif_1, dif_2 and dif_3, calculated according to the formulas (4.10a), (4.10b) and (4.10c). The signrank-test results, as seen on Table 5.4, showed that all three vectors contained data sets deviating significantly from zero. As explained in section 4.3.2 of chapter 4 and by the formulas (4.10), the vectors dif_1, dif_2 and dif_3 corresponded to differences between simultaneous displacement data acquired by pairs of the 3 available methods/devices. A statistically important deviation of dif_1, dif_2 and dif_3 from zero can be logically assessed as a statistically important difference between the data vectors contained in of the formulas for dif_1, dif_2 and dif_3, (4.10a), (4.10b) and (4.10c). Therefore, one can deduce that the difference between the displacement values participating in the aforementioned formulas are statistically significant. In other words, three deductions can be made:

(1) the displacement data acquired by the MPU accelerometer differ significantly from the data acquired by Optitrack,
(2) the displacement data acquired by Zebris differ significantly from the data acquired by Optitrack, and

(3) the displacement data acquired by the MPU accelerometer differ significantly from the data acquired by Zebris.

Optitrack was once again used as the reference method. Throughout this approach, the aim was for all the calculated differences between displacement data to refer to same time-points throughout the Posturomed platform's oscillation. However, as mentioned previously, due to the downsampling process and its approximative nature, the displacement data of Optitrack did not refer to *exactly* the same points in time as the data from the MPU or Zebris. This deviation could have affected the accuracy of the statistical test results, as well as that of the the comparison between the methods' measurement accurasies. There is a certain possibility that the performed statistical tests could have yielded different results if all the measurements refered to exactly the same points in time.

6.1.2 Factors affecting methods' accuracies

In the case of Zebris, certain "gaps" were noticed in the position graphs acquired by WinPosture and therefore in the displacement data itself. This obviously affected the accuracy of the acquired data and needed to be corrected on a separate software, like Matlab, using a data fitting method (curve fitting). Errors of this type not only degraded the quality of the measurements' results but also necessitated a rather time-consuming process for the purpose of their eradication.

It is also important to stress out that the accuracy of the position data acquired by the accelerometer was affected by the process followed to extract them from the original acceleration data of the platform's oscillation. Both numerical integration and curve fitting, the methods used for this purpose, are based on approximation algorithms, which inevitably introduce a certain level of error in the process and therefore, the final position data. Especially numerical inegration as well as successive filtering of the data, were the cause of corruption of the original digital signal, which consequently affected its final quality. It can be easily deduced that optimizing these procedures would contibute to a smoother signal processing and, as a result, would provide position measurements of higher quality and accuracy.

6.2 User-friendliness

6.2.1 Optitrack

In chapter 1, information was provided on the basic principles of Optitrack's operation. The Optitrack system consists of a combination of appropriate hardware and software; a set of infra-red cameras detects the motion patterns of reflective markers places on the object or subject under study, while the corresponding software, Motive, creates a visual, 3D representation of the rigid-body as defined by the selected group of markers.

Set-up of the device, as well as its accompanying software, is an essential process to be followed by the user. In motion capture systems such as Optitrack, where multiple cameras and other components are involved, the quality of the set-up has a significant impact on the quality of the capture volume and thus the motion data produced by the system. An optimized capture volume can ensure high system performance and accuracy, while preventing marker occlusions and mislabelling as well as facilitating and enhancing data post-processing. There is a number of steps involved in this process; setting up the hardware includes preparing the set-up area, correctly

placing the camera mount structures and the cameras themselves, cabling and wiring as well aiming and focusing the cameras on the desired tracking volume. In addition, Motive needs to be installed and activated in order to allow the user to acquire and process motion data. Finally, in order to enable Motive to construct a 3D capture volume, cameras need to be calibrated upon installation as well as periodically throughout Optiracks life circle, as calibration accuracy naturally deteriorates over time due to ambient factors, such as fluctuations in temperature.

Set-up can be a tedious and time-consuming process which needs to be carried out meticulously in order to ensure high quality motion tracking. However, it can be greatly simplified by following the instructions provided by Optitrack's official website, as well as its own documentation wiki page, providing tutorials and manuals on hardware and software set-up, plugins, marker placement and tracking and additional tools, to mention but a few. Furthermore, a list of tutorial videos are provided to enable user not only to install and activate Motive, but also to learn how to operate the system, including information on calibration, marker labeling, creation of rigid-bodies and skeleton assets, file management and a number of other tools and configurations.

Motive itself provides a very modern and clean user interface, where a variety of toolbars, dropdowns, panes and layouts enable the user to customize and configure mocap parameters such as the form of the ridid body or skeleton, acquisition of 2D or 3D data views, recordings of mocap takes, data streaming and post-processing. In addition, Optitrack allows for a calculation of rigid-body coordinates in the capture volume at an outstanding speed, providing users with a visual 3D representation of the body's motion, thus enabling the observation of motion tracking and accelerating implementation of any user-induced changes on the rigid-body itself, at almost real time.

One major drawback of the Optitrack device, which is very relevant to the topic of this master thesis, is Motive's inability to record position data. Motive allows us to track and record and post-process 2D or 3D motion patterns of a rigid-body, it does not however include an integrated DAQ mechanism for the purpose of position data acquisition. MUMBA is a software created by a researcher at the University of Technology and Economics of Budapest, with the purpose of sampling position data during the recordings of motion capture experiments at the Mocap laboratory, allowed by Motive's data live-streaming option. This means that there is a need for separate development and installation of an individual DAQ software for application where position data acqusition is demanded.

6.2.2 Zebris CMS10

The Zebris CMS10 system is a compact, portable device, used in research as a biometric tool. With the use of several specialized markers, the device allows for real-time evaluation and display of measurement data, enabling users to calculate the full range of motion from several different body parts, specific to the demands of each measurement.

Unlike Optitrack, no particularly complex set-up is necessary for Zebris CMS10; in order for it to operate, the device needs to be connected to the measurement device, a power source and a computer, using a simple USB cable. Appart from the hardware and depending on the type of applications or research the user wishes to focus on, an independently developed software, such as WinPosture, also needs to be installed on the computer where the device is connected, so as to enable motion data acquisition. WinPosture is a static and dynamic stabilography software. Posturographic data is acquired via the USB interface and displayed in real time on the computer

screen as a set of graphs of position data on the X, Y and Z axis, as well as the XY level. WinPosture offers a simple and rather intuitive environment, where the user is guided step by step through the process of selecting the type of measurement they wish to conduct, inserting subject information (in biometric applications) as well as record and save motion data in the form of excel files.

Though the user interface is relatively self-explanatory and the device itself rather simple to use, it is worth mentioning that conducting a position measurement requires completing an extensive and rather time-consuming sequence of steps; first of all, Zebris CMS10 needs to be connected properly to the power source, the computer and measurement device upon starting a motion tracking and analysis. Following that, a series of steps needs to be completed on WinPosture (choosing test type, filling in subject information etc) while the device itself has to be properly placed in front of the measurement area, so that it covers the capture volume optimally. Furthermore, not being wireless, the device limits the measurement in a control volume within proximity of either a computer or a power source, potentially confining the user to less options for a position measurement test parameters.

6.2.3 MPU-9150

The *MPU-9150* is a MEMS sensor device which, when connected to a data acquisition device or software, allows users to collect not only acceleration, but also orientation, velocity and position data. In order to conduct such measurements, one would simply need the MPU device itself as well as a micro-controller to connect it to. For the purpose of the current thesis, an *Arduino UNO* board was used as a base for the accelerometer, which contains an integrated micro-controller.

No particularly complex hardware set up was necessary; all that is required for the accelerometer-Arduino configuration in order to operate is a power source, which can be either a battery or a computer, combined with a set of instructions sent to the micro-controller on the UNO board. Instructions are uploaded onto the board's micro-controller in the form of a simple programming code (or sketch) through a standard USD cable, connected to a computer with the open-source Arduino software, IDE. Information on how to use Arduino boards as well as its programming language can be freely accessed by any user by simply visiting Arduino's official website, where there is a wide range of tutorials on how to write sketches as well as user-made and tested codes and projects, all of which can be used as a base for constructing a suitable code for the desired application. As any necessary information can be accessed online, no specific previous knowledge on programming or micro-controllers is particularly needed from the user. This offers the possibility for execution of wireless sensor (or other) applications to practical anyone possessing technical knowledge or not.

Saving acceleration data from MPU's signal can be easily achieved by uploading a code on the Arduino board, instructing it to sample acceleration data with a specific frequency and store it in a simple SD card connected to the board itself.

The MPU-Arduino configuration poses great advantages relating to its small size, simple programming and lack of need for complex set-up. There are, nevertheless, a few areas which could potentially prove to be problematic in motion tracking applications. In the experimental process of this study, one of the major difficulties was finding an effective way to mount the MPU accelerometer onto the Posturomed platform. Despite its small size, because of its particular shape, it was necessary to place the MPU on one of the sides of the platform instead of any place on its surface:

Figure 6.1. MPU-9150 taped on Posturomed platform

Furthermore, in order to ensure a good quality of the sampled acceleration data during the platform's oscillation, it was crucial to securely mount the MPU onto the platform itself, so as to prevent any excess "noise" in the acceleration signal due to possible shifts of the MPU's position during motion. There are several mounting methods, such as taping (Figure 6.1),waxing, or studding, to name-but a few, and it us up to the user to test them and choose the most appropriate one for the application they wish to carry out. Finally, maintaining the MPU in an as much as possible horizontal position was also rather important in order to prevent the appearance of offsets in the acceleration signal, the removal of which would cost time in the post-processing of the motion data.

6.3 Versatility

6.3.1 Optitrack

As an advanced, high-accuracy and quality motion capture system, Optitrack can and has been used for a variety of applications, including human motion analysis, virtual reality, robotics and animation. Because of its versatility, the device is suitable for research purposes in a plethora of fields, scientific and non-scientific, such as entertainment, sports and mechatronics or biomedical engineering.

The Optitrack system is not confined to the study of human motion alone; Motive allows tracking the motion of human subjects as well as objects by creating visual representations of them in the form of skeletons of rigid bodies. The reflective markers placed on the subject or object under examination can be individually selected on Motive's user interface and categorised into groups, each representing an individual rigid-body, allowing to observe the motion of multiple rigid-bodies simultaneously. Furthermore, the options provided by the software not only allow users to record motion experiments, but also modify mocap recording through post-processing and extract different types of motion tracking data (2D or 3D representation).

Optitrack's operational principles are based on an infra-red transmitter-receiver system, using a set of infra-red cameras and reflective markers. As mentioned in section 6.2.1, the cameras need to be properly set-up, wired and calibrated. Upon completing this process, and the installation of the camera mounting sructure and the cameras themselves, it is practically impossible to move the

entire structure as that would require the whole aformentioned process to be repeated. Consequently, mocap applications with Optitrack have to be confined in the environment set by the camera structure. This fact is quite limitting in the sense that it only allows for mocap experiments in said particular environment, within the set capture volume. Furthermore, mocap measurements can only be carrried out indoors, in the absence of natural light or reflective surfaces, which could cause marker occlusion.

6.3.2 Zebris CMS10

The Zebris CMS10 device is used for various applications of 3D movement analysis allowing the simultaneous calculation of the full range of motion from several different body parts. It is compatible with a range of different accessory products, which makes adjustable to users' needs and requirements. The available attachment sets can be positioned on various different body parts, for example by using cervical or lumbar markers, along with the specialized analytics software (WinSpine, WinArm,WinPosture). The software itself gives users the option to choose the type of motion analysis test they wish to perform, offering a selection of many different test types, such as stepping or provocation tests. In addition, its adjustable floor stand and holding device give the CMS10 flexibility which, combined with its portable nature, provide a certain freedom to the user and increase the number of possible applications that can be performed using the device. Furthermore, requiring only connection to a computer and a power source to operate, the Zebris CMS10 is not confined to any particular controlled environment, providing there is no sunlight or reflecting surfaces at the time of mocap experiments which could affect the measurement accuracy. This is an important trait as it makes the device highly suitable for mobile biomechanical measuring stations.

Nevertheless, while the CMS10 certainly holds a number of positive characteristics, we need to mention that its ability to perform motion tracking is highly limited by the range of its capture volume; markers can only be detected within a distance of 1.5 m away from the receiver's pannel, which doesn't allow for wide range motion tracking. Also, prior to every measurement, the aformentioned pannel needs to be properly targeted onto the object or subject of observation so as to effectively track and capture its motion. Finally, when using a device such as the Posturomed for a motion capture experiment, the CMS10 needs to be wired to said device in order to permit motion data acqusition by the accompanying software. Thus, while certainly a protable device, the CMS10 is limited within a close region of the studied moving target.

6.3.3 MPU-9150

Micro-electro-mechanical systems (MEMS) sensors are motion tracking devices being used more and more for the design and development of a number of devices and applications, such as game consoles, smartphones and tablets, car crash airbag sensors etc. Their main operational principle lies in the calculation of 3D acceleration, though post-processing of acceleration data can lead to the acquisition of velocity or even position values.

Their particularly small size makes MEMS sensors applicable to a variety of mechanical applications where large measurements are needed. In addition, when more than one device is needed to deploy a certain process, they can all be integrated in a MEMS chip by using microelectronics, which allows for data reception, filtering, storage and other processes to be carried out on a single chip.
In order to operate, the MPU-9150 breakout board only requires connection with an Arduino

board, through a set of jumper wires, and a power source. Once a code is uploaded onto the microprocessor of the Arduino board, connection with a computer is no longer required, as long as an alternative power source is provided. Thus, an MPU chip can be mounted on an object without the need to be within reach or permanently connected to a computer, which gives users the opportunity to perform measurements in a broad range of environment, even outdoor spaces, instead of being confined to a restricted station or laboratoty. Furthermore, it is possible to change the data acqusition parameters of the device, such as the sampling freqeuncy or the file type for saved data, by simply altering the script imported onto the Arduino board.

Despite its small size, the MPU-9150 device requires secure mounting on a level surface, which could potentially be problematic in certain applications, particularly when tracking the motion of a iregularly shaped object or a human body. Additionally, contrary to Optitrack, the MPU does not allow observation or recording of the visual respresentation of the objects' or subjects' motion, as was the case with Optitrack's rigid-bodies. Furthermore, unlike Zebris, since the MPU device isn't connected to any motion tracking software, there is not the option of obtaining acceleration or position graphs in real time from the motion measurements. Instead, the acquired and saved acceleration data needs to be post-processed in order to extract motion graphs (or patterns) or other parameters like velocity or position.

6.4 Cost

The Optitrack motion capture system is undoubtedly the most expensive out of all three devices used for this master thesis. A set of 8 cameras is the minimun requirement in order for the system to be able to operate. For movement sciences applications, a set of 8 cameras of the type Flex 3, the most inexpensive one, costs **9,076 $** when purchased from the official website of Optitrack, including connection cables, calibration tools and the lisence for Motive. Choosing a more advanced camera type, like the prime 41, or a more extensive ammount of cameras (24) can elevate the cost to a maximum of approximately **152,000 $**!

Online research has show that "HaB direct" is he main supplier for the Zebris CMS10 device, which unfortunately is not available for online purchase. Information on the device's price was neither available on Hab Direct's or Zebris' own online site. However, upon inquisitions made at the University of Technology and Economics' motion capture laboratory's personel, an esstimation of the price of Zebris CMS10 was drawn, raising to approximately **2,000 $**. Of course, further research on the item's price is encouraged for any individual wishing to purchase the CMS10.

The cost of the MPU-9150 and Arduino Uno configuration is considerably smaller than that of the other two motion capture devices. The Uno board, along with the breakaway headers and the jumper cables can be purchased with even less than 20 $, while a simple SD card for data storage and a micro adapter for data transfer to a computer do not exceed the ammount of 10 $. The MPU-9150 device is currently discontinued and replaced by a newer version, the *MPU-9250*, which costs approximately 15 $. In total, putting together an MPU-Arduino assembly does not cost more than **50 $**, a price which can easily be reduced by purchasing these items online from inexpensive sources.

7. Conclusions

Upon completing the experimental process and analyzing the advantages and disadvantages of each method certain conclusions were reached revolving around the comparison of accuracy of the three methods, as well as their suitability for specific types of applications.

7.1 Comparison of methods' accuracies

The results of the statistical comparison of the methods' measurements were presented and discussed in chapters 5 and 6. There it was demonstrated how the statistical tests showed that the measurement accuracies of the MPU-9150 accelerometer and the Zebris CMS10 were *significantly different* from that of the method of reference, Optitrack. Therefore, according to the results of the curtent study, replacing Optitrack with an MPU (MEMS) accelerometer or Zebris CMS10 for position measurement applications, would require carrying out processes in order to enhance the measurement accuracy of the last two devices.

Nonetheless, it is important to make a few relevant comments; though the results of the current research showed that it cannot replace Optitrack effectively, Zebris CMS10 has indeed been the device of choice for 3D motion analysis. It was in fact designed as a biometric tool for biomechanical applications and has been the device of choice for a number of past human analysis research purposes, such as gait analysis and posturography. 3D motion analysis of the elbow joint position in children has been studied using Zebris CMS10 [31], while the device's reliability has been tested on althletes having undergone cruciate ligament reconstruction [30]. Similarly, online research conducted for this master thesis has shown that MEMS accelerometers have been used for biomedical and biomechanical research, especially for the development of wearables. In the field of motion analysis for sports, MEMS accelerometers have been used as an alternative to videographical mocap methods to calculate the position of different body segments of athletes such as swimmers and golfers [27]. Additionally, MEMS systems have been applied for the prognosis of Parkinson's desease, to measure patients' impairement degree [28], as well as for human motion recognition applications [29].

Measurement accuracy requirements can vary depending on the type of each biomedical or biomechanical application. It is up to each researcher to choose a measurement method taking into consideration a number of criteria relevant to their particular study, aside from accuracy, such as effectiveness, under-friendliness and cost. In cases where optical motion capture systems such as Optitrack cannot be employed (for example due to their high cost or to the lack or an appropriate environment for their installation), researchers could choose to sacrifice some of the measurement accuracy by using an alternative, more cost-effective or user-friendly method.

Finally, it is vital to note once again that, throughout the statistical comparison of the methods, measured data did not refer to *exactly* the same points in time during the platform's oscillation (due to Motive's data downsampling). As was highlighted in section 6.1 of chapter 6, this detail could have affected the accuracy of the statistical tests and, consequently, of the following conclusions themselves. Using displacement data referring to exactly the same time-points could have lead to different results which could have potentially shown that the MPU's or Zebris' measurement accuracy could be indeed comparable to that of Optitrack. Therefore, there is still some room for doubt as far as the accuracy of the followed statistical comparison process, as well as of the results themselves, is concerned.

7.2 Comparison of methods on user-friendliness, versatility and cost

7.2.1 User-friendliness

As far as user-friendliness is concerned, both Optitrack and the MPU-Arduino configuration hold a great advantage, thanks to the great amount of relevant available online resources. Opitrack's official website contains step by step information on hardware set-up and software (Motive) installation as well as tutorials on conducting mocap measurements with Motive. Though very thorough and detailed, the online library for Optitrack seems to be orientated towards individuals with at least some level of technical knowledge or understanding of optical motion capture technologies. The online library for Arduino, on the other hand is ideal for a user just getting started with micro-prosessing and wireless transmitter-receivers projects. The *Arduino playground,* a wiki where all Ardunio users can contribute and benefit from their collective research, is a place to share codes, diagrams tutorials and instructions available for anyone to access and edit. The Zebris CMS10 device is quite simple to set-up, while the software used for the current study, WinPosture, was also rather self-explanatory by providing step-by-step instructions for every perturbation test.

In the experimental process of this study, conducting position measurements with Zebris was considerably more time consuming than with the other two devices, as for each measurement the name and location of the storage folder as well as the form in which the position data would be saved had to be individually specified. For position data recordings with Motive, all that needed to be specified upon completion of the measurement was the new file name, while for the MPU, the IDE script imported on the UNO board enabled a fully automated data storage process.

7.2.2 Versatility

When it comes to motion capture, Opitrack certainly offers the widest variety of possible applications out of all three methods; it is not only a pioneering device for biomedical studies, but also a very valuable mocap tool in a variety of fields, such as sports, entertainment and robotics. Available plugins and developer tools for Motive expand the system's capabilities even further by providing additional tools for data streaming and recording or allowing for client applications to run seamlessly on the same system as the tracking software. Furthermore, thanks to its large capture volume and its ability to discern between specified marker sets, Optitrack can track and record the motion of multiply objects (or subjects) simultaneously.

In a similar sense, Zebris CMS10 is also a rather versatile device as it is highly effective for 3D motion tracking of several different body parts, using specialized analytic software depending on which body part the study is focused on. Though not tested throughout this particular experimental process, it is possible that, due to the narrow capture volume of the CMS10 device (1.5 m distance from the receiver's pannel) marker occlusion could arise if one tried to conduct position measurements on more than one software at the same time (for example WinArm and WinSpine). Therefore, simultaneous motion tracking of various body parts could be potentially problematic when using this device.

For the purpose of this master thesis, an MPU accelerometer was used to acquire and post-process acceleration data with the aim of extracting position data for a moving platform (Posturomed). The portable and possibly wireless nature of the MPU-Arduino configuration offers users the freedom

to conduct motion tracking experiments in a variety of environments and conditions, which is not the case for Optitrack and Zebris which can only operate on controlled conditions. Additionally, due to its small size, it is the only device out of all three which can be applied for the development and use of wearables for the purpose of human motion analysis.

7.2.3 Cost

Based on section 6.4, one can easily see that the MPU-Ardunio configuration is by far the most inexpensive out of the three devices tested for this thesis. In reality, the set of MPU-9150 and Arduino Uno board, along with the necessary accompaniments, only has around 0.5-1% of Optitrack's cost, and approximately 2.2% of Zebris CMS10's cost.

8. Limitations to the study

For the purpose of this master thesis, the motion of the Posturomed platform during perturbation test was evaluated on the level defined by the x and y axis alone. Therefore, the displacement results we extracted from the platform's oscillation, as well as the conclusions we reached to on the comparison of accuracy of the three methods were based on the assumption of a *two-dimensional* problem. Biomedical and biomechanical applications are based on the study of the motion of the human body, which takes place in *three-dimensional* space. Inclusion of z-axis motion data in the current thesis would expand the range of current study and the possibilities of its practical application to 3D problems, such as human movement analysis, like gait analysis or posturography.

It is safe to assume that a lot of offsets in the final form of the total displacement data and curves were possibly caused by errors that took place during the post-processing of the acceleration signal acquired by the MPU device. Such offsets could have negatively affected the comparison of accuracy between the MPU and Optitrack data to a significant degree. Thus, it would be worth mentioning that a more advanced filtering process of the acceleration signal and its integrals (velocity and position), thoroughly analysed in section 3.4.2, could optimize the extraction of displacement data from the MPU acceleration data. The same could be assumed for the curve-fitting method on the acceleration signal, with the purpose of extracting a total displacement function, explained in section 3.4.3. Higher accuracy of the final displacement data would signify a smaller deviation from the actual displacement values and could consequently contribute to a more realistic comparison of accuracy between the three methods (or motion capture devices).

It is also important for the reader to keep in mind that, for the current thesis, only non-parametric statistical tests (Wilcoxon signed rank tests) were employed to investigate the significance of the difference between the methods' measured position data. Especially in small data groups or samples, non-parametric methods are less powerful at detecting an effect where there is one than their equivalent parametric methods [32]. This, of course, is true providing that requirements fort he respective parametric methods are met (e.g sample following normal distributions for t-tests). The lack of parameters in non-parametric tests makes it diffucult to reach to quantitative deductions on the actual difference between data groups, or populations. For example, in one-sample non-parametric tests, the data values of a sample may be significantly different from a certain value, but no information is provided on how big that difference actually is [33]. The Wilcoxom signed rank test yields a p-value only, while it does not provide an actual estimate of the magnitude of any effect. In addition, non-parametric tests tend to discard information; the order and signs of the data (whether they lie below, "-", or above, "+", the median) are taken into consideration, while the actual data values are not. It is therefore possible that, due to their weakneses, the types of statistical tests employed for this master thesis have affected the accuracy of the final conclusions on the comparison of the three methods' accuracies.

References

1. Gergely Nagymáté, Rita M. Kiss, 2016, *Application of OptiTrack motion capture systems in human movement analysis - A review.*

2.© 2017 NaturalPoint, Inc. DBA OptiTrack , http://optitrack.com/applications/ (last accessed: 30.05.2017)

3. Michal Kelemen, Ivan Virgala, Tatiana Kelemenová, Ľubica Miková, Peter Frankovský, Tomáš Lipták, Milan Lörinc, 2015, *Distance Measurement via Using of Ultrasonic Sensor*, Journal of automation and control, Vol.3 No. 3, pp 71-74. doi: 10.12691/automation-3-3-6

4. Habdirect, https://www.habdirect.co.uk/zebris-cms10-compact-system-for-3d-motion-analysis (last accessed: 30.05.2017)

5. Portal site for ultrasound sensors, Ultrasonic technology, http://www.ndk.com/en/sensor/ultrasonic/aboutl.html (last accessed: 30.05.2017)

6. prophysics, http://www.prophysics.ch/propduct-database/cms10-en-GB/ (last accessed: 31.05.2017)

7. Hansong Zeng and Yi Zhao, 2011, *Sensing Movement: Microsensors for Body Motion Measurement*, Sensors 11, 638-660;

8. maxim integrated, https://www.maximintegrated.com/en/app-notes/index.mvp/id/5830 (last accessed: 31.05.2017)

9. Instrumentation Today, http://www.instrumentationtoday.com/mems-accelerometer/2011/08/ (last accessed: 31.05.2017)

10. Infohost, http://infohost.nmt.edu/~mreece/gps/applications.html (last accessed: 07/06/17)

11. L. Klimek, R. Mösges, G. Schlöndorff & W. Mann , 2010, *Development of Computer-Aided Surgery for Otorhinolaryngology,* Computer Aided Surgery, Vol. 3, 1998, Issue 4, doi: 10.3109/10929089809148145

12. T. Nagaoka, A. Uchiyama , 2004, *Development of a small wireless position sensor for medical capsule devices*, Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE, doi: 10.1109/IEMBS.2004.1403626

13. Aldawi, Fouad, Longstaff, Andrew P., Fletcher, Simon, Mather, Peter, Myers, Alan and Briggs, Jack, 2008, *A MULTIFREQUENCY FM-BASED ULTRASONIC SYSTEM FOR ACCURACY 3D MEASUREMENT*, Proceedings of Computing and Engineering Annual Researchers' Conference 2008: CEARC'08. University of Huddersfield, Huddersfield, pp. 9-14. ISBN 978-1-86218-067-3

14. Gastone Ciuti, Leonardo Ricotti, Arianna Menciassi, Paolo Dario, 2015, *MEMS Sensor Technologies for Human Centred Applications in Healthcare, Physical Activities, Safety and Environmental Sensing: A Review on Research Activities in Italy,* doi: 10.3390/s150306441

15. KEYENCE, http://www.keyence.com/usa.jsp *,(last accessed: 08/06/2017)*

16. Jiaying Du, 2016, *SIGNAL PROCESSING FOR MEMS SENSOR BASED MOTION ANALYSIS SYSTEM, Mälardalen University Press Licentiate Theses No. 228*

17. Wesley J. E. Teskey, Swavik A. Spiewak, Associate Professor, 2005, *Precision 6-DOF Position Measurement with Inertial MEMS Sensors, University of Calgary, 2500 University Dr. NW, Calgary, AB, Canada T2N 1N4.*

18. Bálint Petró, Dr Ákos Pethes, Prof. Rita M. Kiss, 2017, *Characterising different strategies in a dynamic balance recovery task*

19. Bioswing Canada/ Posturomed, http://bioswing.ca/Bioswing_Canada/Posturomed.html#2 (last accessed: 12/06/2017)

20. P. Kutilek, J. Charfreitag, J. Hozman, 2010, *Comparison of Methods of Measurement of Head Position in Neurological Practice* ,Czech Technical University in Prague, Faculty of Biomedical Engineerig, doi: 10.1007/978-3-642-13039-7_114

21. ARDUINO, https://www.arduino.cc/en/Guide/Introduction (last accessed: 26/09/17)
22. Wikipedia-Numerical Integration, https://en.wikipedia.org/wiki/Numerical_integration
23. Using the trapezoidal rule for numerical integration in MATLAB, https://www.youtube.com/watch?v=QvCYQHRlsO4&t=69s
24. Investopedia, *T-test,* https://www.investopedia.com/terms/t/t-test.asp (last accessed 22/12/17)
25. Statwing Documentation, *Statistical significance (T-test)*, http://docs.statwing.com/examples-and-definitions/t-test/statistical-significance/ (last accessed: 22/12/17)
26. http://www.chem.utoronto.ca/coursenotes/analsci/stats/ttest.html (last accessed: 22/12/17)
27. Yugi OHGI, 2006, *MEMS sensor application for the motion analysis in sports science*, Graduate School of Media and Governance, Keio University.
28. MUHAMMAD ATIF, SERKAN SERDAROĞLU, 2012, *A Measurement System for Human Movement Analysis*, CHALMERS UNIVERSITY OF TECHNOLOGY, Department of Signals and System, Division of Biomedical Engineering.
29. Guangyi Shi, Yuexian Zou , Yufeng Jin, 2009, *Towards HMM based human motion recognition using MEMS inertial sensors*, doi: 10.1109/ROBIO.2009.4913268.
30. Hoda Niknam et al., 2017. *Reliability of Zebris Motion Analysis System in Healthy Athletes and Athletes with Anterior Cruciate Ligament Reconstruction*, Asian J Sports Med, doi: 10.5812/asjsm.42040.
31. So-Young Hong et al., 2016, *Three-dimensional motion analysis in the elbow joint position sense in children*, J Phys Ther Sci, doi: 10.1589/jpts.28.3313.
32. Elise Whitley, Jonathan Ball , 2002, *Statistics review 6: Nonparametric methods,* BioMed Central Ltd.
33. Gerard E. Dallal, Ph.D., 2000, *Nonparametric Statistics.*