ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ &ΕΠΙΧΕΙΡΗΣΙΑΚΗΣ ΕΡΕΥΝΑΣ
ΜΟΝΑΔΑ ΕΡΓΟΝΟΜΙΑΣ

MARG System for Real Time Upper Body Movement Monitoring

Σύστημα MARG για Ενεργή Παρακολούθηση της Κίνησης Κορμού και Άνω Άκρων

Διπλωματική Εργασία

Γιαγκλίσης Ευάγγελος

Επιβλέπων: Δημήτριος Ναθαναήλ, Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2019

Ευχαριστίες

Με την ολοκλήρωση αυτής της εργασίας φαίνεται να κλείνει ένας μεγάλος κύκλος της ζωής μου. Σε όλο αυτό το χρονικό διάστημα η γνώση, οι εμπειρίες και όλα εκείνα τα μαθήματα που πήρα, θα μείνουν χαραγμένα πάντα μέσα μου. Όλα αυτά όμως θα ήταν τελείως διαφορετικά αν δεν υπήρχαν ορισμένα άτομα που μου προσέφεραν τα μέγιστα, μέσα από τη προσωπικότητα, το ενδιαφέρον και τις γνώσεις τους, σε αυτή την αξέχαστη πορεία. Θέλω και εγώ λοιπόν, με την σειρά μου, να αφιερώσω τις πρώτες λέξεις αυτού του συγγράμματος, για να πω ένα μεγάλο “ευχαριστώ” και να εκφράσω την ευγνωμοσύνη μου σε αυτούς τους ανθρώπους.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Δημήτριο Ναθαναήλ, που καθόλη την πορεία της διπλωματικής μου εργασίας, πίστεψε σε εμένα, με στήριξε και με καθοδήγησε. Η προσφορά του ήταν καθοριστική, μέσα από την ευρεία γνώση του πάνω στις θεματολογίες της εργασίας και την ουσιώδη υποστήριξη του καθ ’όλη την διάρκεια της εργασίας.

Ακόμα, θα ήθελα να ευχαριστήσω όλους εκείνους τους ανθρώπους που συνέβαλλαν στην ολοκλήρωση της εργασίας μου, προσφέροντας τον χρόνο, τις γνώσεις και την διάθεσή τους. Ιδιαίτερα, θα ήθελα να εκφράσω την ευγνωμοσύνη μου για την συμβολή των κυρίων Κ. Γκίκα, Λ. Ψαράκη, Α. Μουρελάτο και Π. Κοντραζή, μελών της μονάδας εργονομίας. Επίσης και τους Γ. Παπαζέτη, Μ. Αγγέλου, Γ. Καλαπόδη, Ι. Νταβλιάκο και Ν. Ρούβαλη για την συνεισφορά με τις υποδείξεις τους.

Επιπλέον θα ήθελα να ευχαριστήσω και τους φίλους μου που με στήριξαν και με βοήθησαν με μεγάλη προθυμία στην εκτέλεση πειραμάτων για την εξέλιξη του συστήματος.

Κλείνοντας, θα ήθελα να πω ένα τεράστιο ευχαριστώ, πρωτίστως στους γονείς μου για την αδιάκοπη υποστήριξή τους κατά την διάρκεια των μαθητικών μου χρόνων, σε δύσκολες στιγμές, σε αποτυχίες και επιτυχίες. Ήταν και είναι πάντα δίπλα μου και δυστυχώς δεν μπορώ να βρω αρκετές τέτοιες ευκαιρίες να τους εκφράσω την αγάπη και την ευγνωμοσύνη μου.

Abstract

MARG systems have become a viable, accurate and cost efficient solution for MoCap (Motion Capture) applications. Implementing a magnetomer in a common IMU module, could provide for more accurate and responsive measurements. Moreover, it could link the data received by the other two modules of the IMU (accelerometer and gyroscope), with Earth's magnetic field, creating a geodetic reference system.

In this thesis an IMU and an MARG system are evaluated. Comparing the behavior, the accuracy and the response of the two systems we extract the advantages and disadvantages of each method. Results revealed that the simple IMU method benefits in response, convenience and practicality, were as, the MARG method benefits in accuracy, referencing and reliability.

Thus a wearable, wireless MARG monitoring system for upper body movement and posture is developed, using nine Microprocessor Units (MPUs) placed on torso and arms. Its purpose is to later establish a testing system for various motion capture applications, such as educational virtual environments, work force training etc.

The development of the system was extended to mechanical, electrical and electronical rework. New robust cases for the sensors, a new method of interchanging data through and a more efficient algorithm were created. Moreover the system was designed in prospect to accommodate even more modules and in the future be a full body host, of a completely independent and wireless system, having any desired number of sensors connected with it.

In conclusion, further evaluation and improvements need to be made, so as to create a fully operational full body MARG monitoring system. Despite of that, the sensor coverage has increased and significant steps forward have been made to achieve modularity, robustness and accuracy with this new upper body MARG low-cost testing system.

Περίληψη

Τα συστήματα MARG έχουν γίνει μια βιώσιμη, ακριβής και αποδοτική λύση για εφαρμογές καταγραφής κίνησης (MoCap). Η εφαρμογή ενός μαγνητόμετρου σε συνεργασία με ένα σύστημα IMU θα μπορούσε να παρέχει πιο ακριβείς και καλύτερης ανταπόκρισης μετρήσεις. Επιπλέον, θα μπορούσαν να συνδιαστούν τα δεδομένα που λαμβάνουν τα άλλα δύο στοιχεία του IMU (επιταχυνσιόμετρο και γυροσκόπιο), με το μαγνητικό πεδίο της Γης, δημιουργώντας ένα γεωδαιτικό σύστημα αναφοράς.

Σε αυτή την εργασία αξιολογούνται ένα απλό IMU και ένα σύστημα MARG. Συγκρίνοντας τη συμπεριφορά, την ακρίβεια και την απόκριση των δύο συστημάτων εξάγουμε τα πλεονεκτήματα και τα μειονεκτήματα κάθε μεθόδου. Τα αποτελέσματα αποκάλυψαν ότι η απλή μέθοδος IMU επωφελείται από καλύτερη απόκριση, ευκολία και πρακτικότητα, ενώ αντίθετα, η μέθοδος MARG επωφελείται από ακρίβεια, καλύτερη αντιστοίχιση κινήσεων και αξιοπιστία.

Έτσι, αναπτύχθηκε ένα φορητό, ασύρματο σύστημα παρακολούθησης MARG για την κίνηση και τη στάση του άνω σώματος, χρησιμοποιώντας εννέα Μονάδες Μικροεπεξεργαστών (MPUs) τοποθετημένες σε κορμό και άνω άκρα. Σκοπός του είναι να καθιερώσει αργότερα ένα σύστημα δοκιμών για διάφορες εφαρμογές καταγραφής της κίνησης, όπως εκπαιδευτικά εικονικά περιβάλλοντα, κατάρτιση και εκπαίδευση εργατικού δυναμικού κλπ.

Η ανάπτυξη του συστήματος επεκτάθηκε τόσο σε μηχανικές, όσο και ηλεκτρικές και ηλεκτρονικές αλλαγές. Δημιουργήθηκαν νέες στιβαρές θήκες για τους αισθητήρες, μια νέα μέθοδος εναλλαγής δεδομένων και ένας πιο αποδοτικός αλγόριθμος. Επιπλέον, το σύστημα σχεδιάστηκε με την προοπτική να φιλοξενήσει ακόμα περισσότερες μονάδες MPU και στο μέλλον να είναι ένα πλήρες σώμα, αποτελούμενο από έναν εντελώς ανεξάρτητο και ασύρματο αριθμό επιθυμητών αισθητήρων που θα συνδέονται με αυτό.

Εν κατακλείδι, πρέπει να γίνουν περαιτέρω αξιολογήσεις και βελτιώσεις, ώστε να δημιουργηθεί ένα πλήρως λειτουργικό σύστημα παρακολούθησης MARG ολόκληρου του ανθρωπίνου σώματος. Παρ 'όλα αυτά, η κάλυψη του σώματος από αισθητήρες έχει αυξηθεί και έχουν γίνει σημαντικά βήματα προς τα εμπρός για να επιτευχθεί η ανεξαρτητοποίηση κάθε αισθητήρα, περισσότερη στιβαρότητα και ακρίβεια, με αυτό το νέο σύστημα ελέγχου MARG χαμηλού κόστους.

Table of Contents

i. Table of Figures

ii. Table of Charts

iii. Table of Tables

iv. Glossary

- CAD: Computer Assisted Design
- CGI: Computer Generated Imagery
- DAC: Digital to Analogue Converter
- DMP: Digital Motion Processor
- DoF: Degrees of Freedom
- EDR: Enhanced Data Rate
- EKF: Extended Kalman Filter
- EMG: Electromagnetic
- EEG: Electroencephalography
- EU: European Union
- FDM: Fused Deposition Modeling
- IMU: Inertial Measurement Unit
- I2C or I^2C: Inter-Integrated Circuit
- MARG: Magnetic, Angular Rate and Gravity
- MCM: Multi Chip Module
- MEMS: Microelectromechanical System
- MoCap: Motion Capture
- MPU: Microprocessor Unit
- NTUA: National Technical University of Athens
- OMC: Optical Motion Capture
- PE: Physical Education
- PWM: Pulse Width Modulation
- QFN: Quad Flat No-leads
- SPI: Serial Peripheral Interface
- SPP: Serial Port Protocol
- TTL: Transistor-Transistor Logic
- US: United States
- USART: Universal Synchronous/Asynchronous Receiver/Transmitter
- VE: Virtual Environment
- VR: Virtual Reality

v. Introduction

Human body motion analysis and tracking has been, over the last few years, an important issue of many scientific fields such as medicine, mechanics, biomedical engineering and ergonomics. Especially in topics of industrial and scientific research such as optics and surveillance, VR (Virtual Reality), worker training, PE (Physical Education), physiotherapy and rehabilitation, CGI in entertainment and video games, human motion tracking and movement measurement has been a major point of focus (Lee, Low, & Taher, 2010).

The idea of understanding human locomotion began centuries ago and thus the need of capturing and measuring human movement was created. Starting from the attempts of the Weber brothers (1836) in reporting quantitative studies of the temporal and distance parameters during human locomotion (Weber & Weber, 1836), E. Marey (Animal Mechanism: A Treatise on Terrestrial and Aerial Locomotion, 1874) and E. Muybridge (Animal locomotion, 1887) trying to quantify patterns of animal movement through photography, to W. Braune and O. Fisher (Determination of the moments of inertia of the human body and its limbs, 1988) calculating joint forces and energy expenditures using Newtonian mechanics, the research approaches have been many (Mündermann, Corazza, & Andriacchi, 2015).

Even today, many of those ideas are still developed and applied in several systems, creating a basis for many new platforms of human body and limbs motion analysis. Therefore the present thesis is going to be separated in five main parts:

A. The evaluation of the current technology, the uses and the benefits of the existing methods of human body motion and limbs analysis and the explanation of the selected choice for the task at hand.
B. A Mathematical, Physical, Anatomical and Engineering background lay-out of the current system.
C. The description of the new upper-half body setup, based on an innovative design of a fully developed VR environment along with a setup for measuring the angles of the wrist and the forearm created by previous researchers of the NTUA Lab of Cognitive Ergonomics.
D. A validation test of the current setup.
E. An assessment of the current status and ideas for further improvement.

A. Review of Motion Tracking Technology

A.1. Methods for Motion Tracking

Motion tracking methods can be separated into three main categories based on the media of measurement. Those are:

A.1.1 Optical Tracking

Also known as OMC (Optical Motion Capture), optical tracking is characterized by an image capturing device, usually a camera or IR camera. The setup might include one or several cameras supported by a method for feature recognition. There are two types of setups based on the method of feature recognition: Marker and Markerless.

A setup using markers usually incorporates some piece of clothing or cupping rig to attach IR reflecting marker or LEDs that are tracked by special software installed in the motion capture setup. These setups are often used in CGI capture studios, video-games and gait and movement rehabilitation centers.

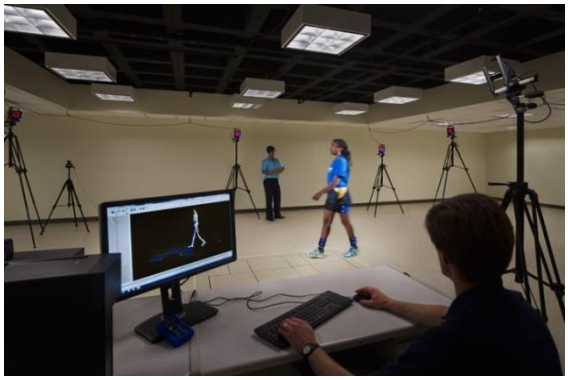


Figure 1: Marker-using setup in Gait Lab Drexel's College of Nursing and Health Professions **(Ewing Rachel, 2015)**

Markerless setups make use of smart matrix-masks which can be applied on the image matrix provided by a camera to recognize pre-saved human features such as body contours, or limb shapes, by adjusting their size, rotating and translating their features. Moreover, by integrating several cameras providing several images per second, they can calculate other features of human movement with relative accuracy, such as orientation, speed and acceleration (Bobick & Davis, 2001). Such setups are usually used for home entertainment systems, gaming, surveillance, product visualization and advertising (Schecter, 2014).

Figure 2: Commercial Visual Based MoCap Products: Microsoft X-BOX Kinect™ [1] (left), SONY PlayStation Camera™ [2] (right)

The cost of this method increases along with the desired accuracy and can reach really high standards using a multi-camera setup along with very sophisticated software. There some drawbacks of portability and space requirements but commercial systems have already solved some of the parameters by creating fully integrated gaming systems and really well backed-up software with a big community of developers creating new applications every day.

It is worth noting, that there is a unique method using optical fiber sensors usually integrated in glove or finger-type motion detectors (Kramer, Linderner, & George, 1991). It measures bend through a light source and a photodetector (Kortier, Sluiter, Roetenberg, & Veltink, 2014). The light travels through fiber optics passing from finger joints and is detected at certain photodetector points, determining the change of orientation of the link-joint system (Simone & Kamper, 2005). This method is, for now, deemed as costly, bulky and underdeveloped for many systems trying to incorporate, not only finger motion, but full body motion as well.

[1] Retrieved 10/16/2018, from Xbox Support: https://support.xbox.com/en-GB/xbox-360/accessories/setup-and-playspace-info

[2] Retrieved 10/16/2018, from Playstation: https://www.playstation.com/el-gr/explore/accessories/playstation-camera/

A.1.2 Mechanical Tracking

Mechanical tracking is usually base on devices called goniometers. These devices are attached to a human joint, providing one degree of rotational freedom per measurement. They usually consist of two main rigid splint-like parts that are strapped to the two body parts that usually are significant for the joint (e.g. elbow; arm and forearm) and a rotational joint mechanism (e.g. a bearing). This method is really old, but is still applied in fields of medicine, for research that does not require great precision, range of or fast movements.

Figure 3: Plastic Wrist Goniometer

Due to its simplicity portability and inveteracy, this method is really common and has a really low cost, but also has very small potency for innovation.

A.1.3 Inertial Tracking

Inertial Tracking is based on a complex device called IMU (Inertial Measurement Unit). The device has been considered as a novel Microelectromechanical system (MEMS) and is comprised of three main parts; an accelerometer, a gyroscope and a magnetometer.

Accelerometer

A device used to measure proper acceleration (the rate of change of velocity) of a body in its own instantaneous frame (Tinder, 2007). Common MEMS accelerometers use different methods of acceleration sensing such as: Liquid tilt sensors, Bulk Micro-machined Piezo Resistive, Bulk Micro-machined Capacitive, Piezoelectric or Surface Micro-machined Capacitive (Doscher, 2008). Each method has its own advantages and disadvantages that will not be analyzed in the present thesis. More importantly the breakout board MPUs (Microprocessor units) that are low cost, relatively accurate and available in the market, such as the MPU-9250 (InvenSense Inc., 2016), use Surface Micro-machined Capacitive sensors (Digi-Key's European Editors, 2013). A conventional IMU comprises three accelerometers (Schopp, Klingbeil, Peters, Buhmann, & Manoli, 2009).

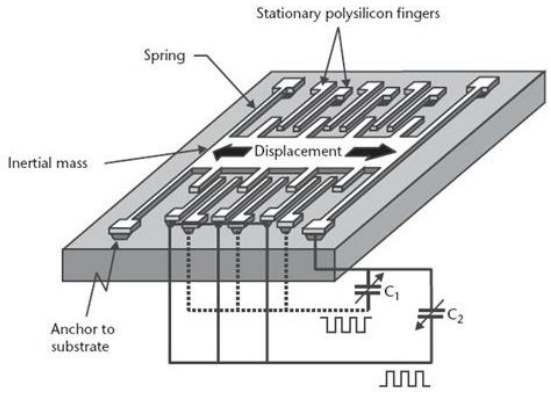


Figure 4: Surface Micro-machined Capacitive Accelerometer **(MEMS Class 6 Microsensors)**

Gyroscope

A device used to measure angular velocity. The usual gyroscope found IMUs is a micromachined MEMS based on the idea of the Foucault pendulum (von Bergmann & von Bergmann, 2007). Almost all reported micromachined gyroscopes use vibrating mechanical elements to sense rotation. All vibratory gyroscopes are based on the transfer of energy between two vibration modes of a structure caused by Coriolis acceleration (Yazdi, Ayazi, & Najafi, 1998). A conventional IMU comprises three gyroscopes (Schopp, Klingbeil, Peters, Buhmann, & Manoli, 2009).

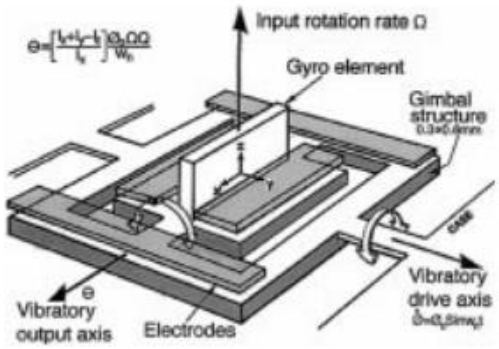


Figure 5: Draper's first silicon micromachined double-gimbal vibratory gyroscope **(Greiff, Boxenhorn, King, & Niles, 1991)**

Common MEMS gyroscopes can be responsible for error amplification due to high durability designs, restricting the movements and causing damping in certain inertial readings.

Magnetometer

A device used to measure the direction, strength or relative change of a magnetic field at a particular location. Common MEMS magnetometers are usually 3-Axis compass modules with an accuracy of 1^{o}-2^{o}. The compass comprises of Magneto-resistive type sensors that read the changes in magnetic field based on the Principle of Hall effect (Ramsden, 2001). Commercial Magneto-resistive sensors all suffer from certain problems:

- Low-frequency noise measurements due to sensitivity, bias voltage and required bias filed (Stutzke, Russek, & Pappas, 2005).
- Drifting phenomena due to Earth's magnetic field fluctuations (Anderson, Anghel, Yumoto, Ishitsuka, & Kudeki, 2002)
- Interferences due to high ferromagnetic material presence in the area (Sheinker, Frumkis, Ginzburg, Salomonski, & Kaplan, 2009).

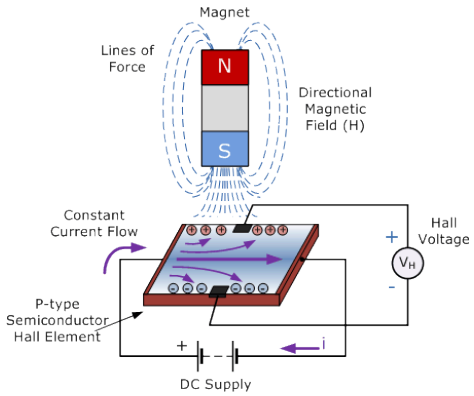


Figure 6: The Hall Effect sensor principle on a common magnetometer[3]

[3] Retrieved 11/27/2018, from https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html

Those three different sensors are merged together using different methods for sensor fusion. By incorporating the data from each main part, the IMU can provide a rather accurate and easily accessible set of coordinates. Its main advantage is its small size and low cost coming along with limited power consumption (Chen X. , 2013). The use of several sensors can enable accurate tracking of body segment motion through rotation and, if supported by a modular design, it can facilitate easy donning and doffing in a non-intrusive and fast way (Fitzgerald, et al., 2007). Due to their size and design, IMUs can also be sewn into clothing or modularly added to headgear and VR headsets. Thus the ease of use provides a huge advantage over more bulky and costly methods.

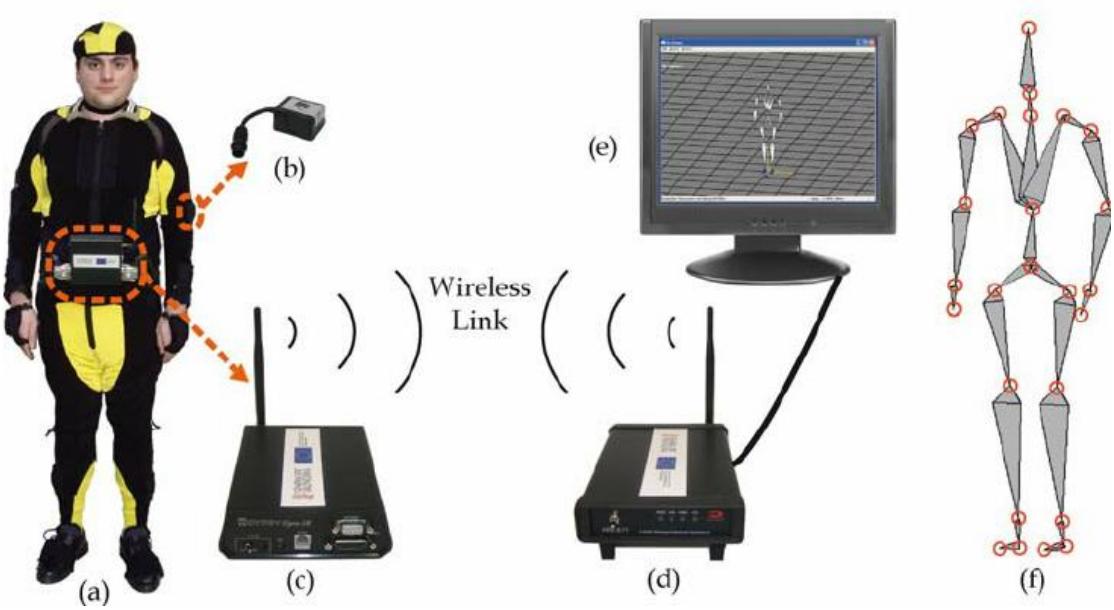


Figure 7: Components of the inertial motion capture system: (a) suit, (b) IMU, (c) MPU, (d) wireless modem, (e) controller PC and (f) skeleton structure. **(Corrales, Candelas, & Torres, 2010).**

A.1.4 Electrical and Magnetic Tracking

Electrical and Magnetic Tracking methods can vary in terms of quality and complexity depending on the tracking module that is used. The main principle though is the same; calculating movement depending on the changing currency of the module. The types of modules can be categorized as;

Electromagnets

Replacing regular magnets, electromagnetic motion tracking systems have been used to objectively measure surgical skill (Datta, Mackay, Mandalia, & Darzi, 2001), assess joint kinematics (Bottlang, Mady, Steyers, Marsh, & Brown, 2000) and support methods and metrics of general rehabilitation. Electromagnetic motion analysis systems are amongst most widely used systems for joint movements' analysis, because of their advantage as a non-invasive technique (Lin, et al., 2005). One great example is the Polhemus FASTRAK™ motion analysis system. It's characterized as a reliable and accurate 6 DoF EMG tracking system delivering real-time data with up to four sensors (Polhemus).

Figure 8: FASTRAK™ EMG tracking system by Polhemus

Magnetic field sensing can track the orientation and movement of a human joint accurately and occlusion-free (Raab, Blood, Steiner, & Jones, 1979) by generating a very high strength magnetic field on one end of the joint and measuring its change with multiple sensors on the other end.

"In order to avoid drifting issues with such an approach, the base station must be stationary, and to guarantee accuracy the three axes of fields must be time-synced. In addition, the user must stay within a narrow sensing range relative to the base station. These requirements make the system unsuited for portable use, especially for wearable applications" (Chen, Patel, & Keller, 2016). Moreover the issue of cumulative bulk and rising expenses through adding sensors makes the system inapproachable and therefore it is rarely used for non-scientific research tasks.

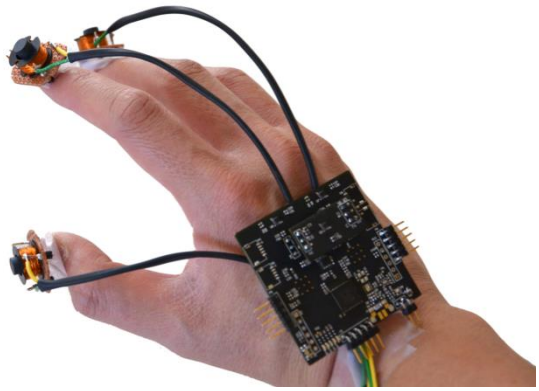


Figure 9: Finexus EMG tracking system **(University of Washington, Ubicomlab, Chen et al.)**

A radical approach was made by the aforesaid researchers (Chen, Patel, & Keller, 2016) by creating the Finexus, a less costly and bulky approach to an EMG tracking system. Still, there are a few problems to be confronted, such as noise and drift of the EMG readings due to various ferromagnetic objects, planetary magnetic field and EMG waves emitted from nearby electronic devices (Chen, Patel, & Keller, 2016).

Flex sensors

Flex sensors are of four different types, one of which, the optical-fiber flex sensor (pg. 16) will not be analyzed in this chapter to avoid repetitiveness. The three that will be analyzed are:

- **The conductive fabric/thread/polymer-based flex sensor**

  Its function is very similar to the piezoelectric module of the accelerometer. This sensor actually reacts (drops its resistance) to pressure and not specifically bend and usually uses easily accessible tactile-purposed fabrics such as Neoprene with a layer of resistive material (e.g. Velostat) (Sensor Wiki, 2011). These sensors are typically employed by hobbyists because of their low cost. However they suffer from poor accuracy, repeatability and hysteresis (Dunne, Smyth, & Caulfield, 2007).

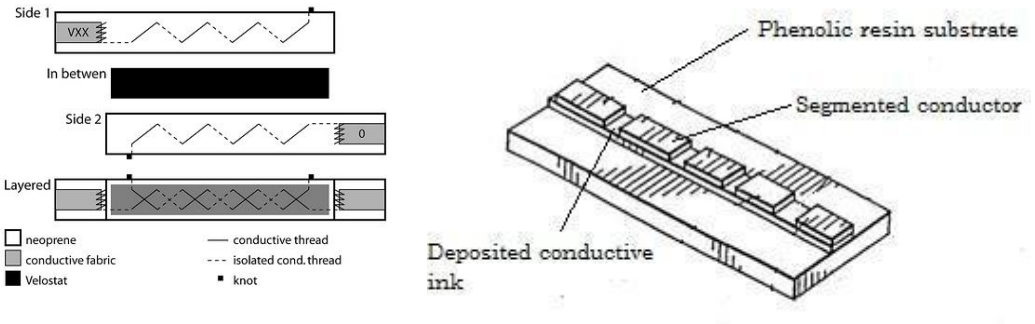


Figure 10: Neoprene Bend Sensor[4] (left), Conductive ink based flex sensor (right) **(Sreejan & Narayan, 2017)**

[4] Retrieved 10/30/2018, from https://www.kobakant.at/DIY/?p=20

- **The conductive ink based flex sensor**

  A passive resistive device build with a strip of conductive ink deposited in a pattern on a flexible resin preferably with a segmented conductor positioned on top, forms a flexible potentiometer in which the resistance consistently and predictably changes upon tensile or shear phenomena (Langford, 1990). When laid at rest and run by a known current the sensor provides certain Voltage and thus, certain resistance. This metric is its nominal resistance and can increase by a factor of ten at full deflection (Sensor Wiki, 2011). They are of relatively low cost and have negligible hysteresis and noise (Dunne, Smyth, & Caulfield, 2007).

- **The capacitive bend sensor**

  A dielectric material is disposed between the comb-patterned portions of two elements so as to bond them by one sliding relatively to other. Bend angle is measured by measuring the voltage that changes according to the alignment of the comb-patterned portions. Capacitive bend sensors are relatively accurate, robust modules and their cost and quality of measurements depends on the chosen materials. The simplest and cheapest of them are available on the market and can be integrated in systems for commercial purposes (Fifth Dimension Technologies 5DT). The ones that use more unique materials, for now, are only used for research (Cotton, Graz, & Lacour, 2009).

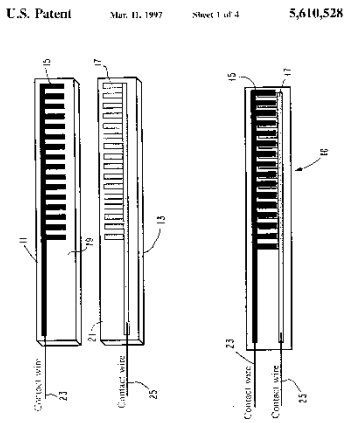


Figure 11: Capacitive bend sensor design **(Neely & Restle, 1997)**

A.1.5 Method Comparison

To assess the pros and cons of each method for each given task, a detailed comparison chart (Table 1) is created. Table 1 provides the characteristics of each method and some sub-methods that might differ. The evaluation is based on accuracy, resolution, repeatability, cost [5], portability, range of motion, modularity, automation, durability/robustness, simplicity of use and commercial availability. The methods chosen, based on their uniqueness and differences with other sub-methods are: optical markerless, optical with markers, mechanical, inertial, magnetic and flex sensor.

The matrix includes the range of cost and accuracy without making any correlation between them, as sometimes the lowest cost does not correspond to the lowest accuracy. Accuracy refers to the difference between actual specimen movements and recorded specimen movements mostly based on center of mass position errors. Also the term repeatability refers to single sensor measurement variance between identical testing. Lastly automation refers to the system automatically providing measurements after setup, without user adjustments.

Table 1: Motion Capture Method Comparison

	Optical (IR/LED)	Optical (Software Image Capture)	Mechanical (Goniometer)	IMU	Magnetic	Flex (Electric)
Cost	80$- >15000$	Commercial Unavailable/ Custom-made	4-1350	3-260	300-6350	1000-6000
Accuracy	1mm (<1°)	5mm-3cm	User Depended	1°-5°	1mm (<1°)	1°-3°
Resolution	Setup Depended	Setup Depended	1°	0.01°-0.1°	0.025°	<1°
Repeatability	<0.3mm	Setup Depended	User Depended	0.5°-2°	0.15°	3°
Range of motion	Setup Depended 3-9 DoF	Setup Depended 3-9 DoF	1DoF	9DoF (Full)	6DoF	2-6 DoF
Portability	1	5	5	4	3	3
Modularity	5	5	1	5	4	3
Automation	5	3	1	5	5	5
Robustness	4	5	4	3	3	3
Simplicity of use	3	2	5	3	3	4
Availability	3	1	5	4	2	3

[5] Cost is based on pricing through various manufacturers or resellers as of 11/28/2018

On the other hand, Table 2 presents the results of each method regarding common problems that are frequently observed through testing.

Table 2: Common MoCap problems and method correspondence

	Jittering	Ferromagnetic Interference	Drifting	Skin Movement Error	Depth of Field Loss	EM Signal interference	User Error	Placement Error
Optical (IR/LED)	✓	-	✕	✓	✕	✕	✕	✓
Optical (Software Image Capture)	✕	-	✕	✓	✓	✕	✓	-
Mechanical (Goniometer)	-	-	-	✕	-	-	✓	✓
IMU	✓	✓	✓	✓	-	✓	✕	✓
Magnetic	✓	✓	✓	✕	-	✓	✕	✓
Flex (Electric)	✓	✕	✕	✓	-	✕	✕	✓

✕= Does not face this problem

✓= Could face this problem

- = Irrelevant problem

Chart 1 also shows a comparison between some examples of available products with a price, accuracy, resolution, repeatability and other subjective ratings of the other features mentioned on Table 1. The chart ranks them from 0-10 depending on their relative rating of the features that were previously mentioned:

- "0" being the most costly, less accurate, most difficult to repeat the same experiment, with lowest resolution and least degrees of freedom, non-portable, non-modular, non-automated and not robust and simple to use.
- "10" being the most cost efficient, most accurate, least difficult to repeat the same experiment, with highest resolution and most degrees of freedom, most portable and modular, completely automated and most robust and simple to use.

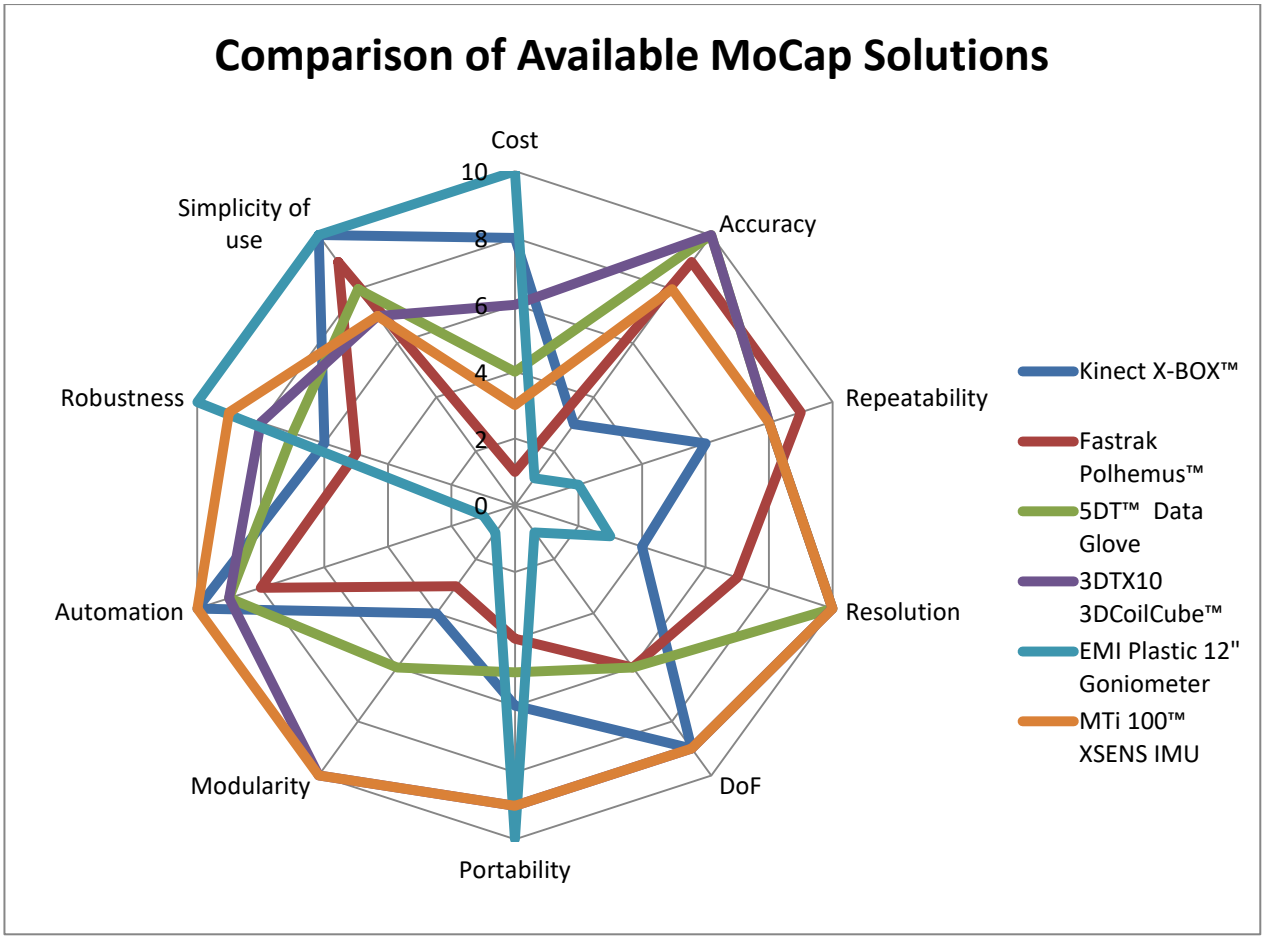


Chart 1: Qualitative comparison of available MoCap Solutions

Even though the comparison is qualitative, we can see that there are various methods to achieve the desired outcome in terms of accuracy, robustness and cost. The IMU system presented in Chart 1, seems like an all-around effective solution, but has the second most costly setup from the list. Reducing the cost of such a setup is one of the main topics on this thesis that will be analyzed later on.

A.2. Applications of Motion Tracking

A.2.1 CGI and Animation

Computer-generated imagery, also known as CGI, is referred to the use of computer graphics to create dynamic or static images that are later processed for creating animated objects, animals or characters in any form of animated visual scene. Contemporary motion capture for animation mostly employs optical tracking techniques, although several different methods have been used in the past (Gleicher, 1999).

The field became radically popular through movies, using animations and video-games, wanting to create more realistic characters. The technology behind the needs of the industry has advanced so much, that even face expressions and micro-gesture have been implemented, so as to portray each and every movement of the complex muscular groups of the human face. Common setups use a full body motion capture suit along with headgear, body-painted tracking markers, sound capture devices, green-screen backgrounds and props that blend with animated environment for more cinematic displays.

Figure 12: Face motion capture method used by KINETIC™ [6]

[6] Retrieved 10/26/2018, from http://kinectic.net/motion-capture-face/

A.2.2 Ergonomic Evaluation

What better field to employ human MoCap than ergonomics and the evaluation of industrial workstations and tools? Many companies dedicate large amounts of resources to improve the design, the usability and the customer experience for their products along with their employee's productivity. To make such improvements, research in ergonomics has been combined with motion capture systems, allowing a more accurate and detailed approach the observation and analysis of potential features of interest.

Some examples of such research involves assessing workstation ergonomics (Duffy, 2007), pose estimation and improvement (Moeslund & Granum, 2001) hand tool use (Karakikes, 2017), sensorimotor tasks (Mourelatos, 2018) and cooperation with robotic arms and use of dangerous machinery (Kontrazis, 2018).

The choice of MoCap method has been confined mainly in the most developed methods for motion capture; Optical Tracking and Inertial Tracking. The idea usually revolves around observing of an employee doing everyday tasks and then implementing improvements in terms of posture orthopaedics, tool handle design, methods of human-machine interactions etc. mentioned also as Reactive ergonomics. "Reactive ergonomics (production line ergonomics) is concerned with human anthropometric, physiological and biomechanical characteristics as they relate to physical activity. Relevant topics include working postures, materials handling, biomechanics, repetitive movements, takt time reduction, work efficiency, work-related musculoskeletal disorders, workplace optimization, safety and health" (Xsens).

Three of the most proven systems in the field are the Kester™ Optical Mocap system by Motion Analysis (Motion Analysis) the Movit G1™ and Overtraq™ IMU-based MoCap systems by Captiks (Captiks) and the Xsens MVN Analyse™ IMU-based MoCap system (Xsens).

Figure 13: The Xsens MVN Analyse™ IMU-based MoCap system used for workspace and ergonomic evaluation [7]

[7] Retrieved 11/9/2018, from https://www.xsens.com/tags/ergonomics-human-factors/

A.2.3 Gesture Recognition

The use of motion capture and analysis methods has been integrated, over the last few years, to commercial cameras and cellphones for the purpose of gesture recognition. Gesture recognition does not only refer to facial expressions and hand signs made by humans, but also movement of hand, body and iris, or a combination of all of them. The idea behind gesture recognition is to automate certain software features, by recognizing a certain gesture either by camera or device implemented IMU.

Figure 14: Brainwave app using Android phone camera for hand gesture recognition[8]

The research on the field has advanced to a point where, even full body gestures can by analyzed and interpreted into certain software functions (Hwang, Kim, & Lee, 2006). Moreover it has been used for sign language recognition (Chen & Koskela, 2013) and adaptive, personalized posture and gesture detection (Neverova, Wolf, Taylor, & Nebout, 2016).

The availability of inertial sensors embedded in mobile devices, along with the use of the front and back cameras has allowed for a plethora of commercial applications of different kinds of movements (Xian, Tarrío, Metola, Bernardos, & Casar, 2012). The most characteristic ones are hand and face gestures shown in Figure 14.

[8] Retrieved 11/19/2018, from https://www.dailymail.co.uk/sciencetech/article-2786803/Conduct-playlist-wave-hand-App-lets-pause-play-skip-songs-WITHOUT-touching-phone.html

A.2.4 Physical Education (PE) and Sports Science

One of the main fields in sports science is evaluating, improving and optimizing the most important parameters of sports performance (Bideau, Kulpa, Vignais, Brault, Multon, & Craig, 2010). To achieve the most out of the scientific research made in this field, several systems have been created. Some of them try to implement motion capture techniques to analyze the movement and the force parameters of the athlete's routine and then work on improving several aspects of it via physiological, biomechanical and psychological analyses.

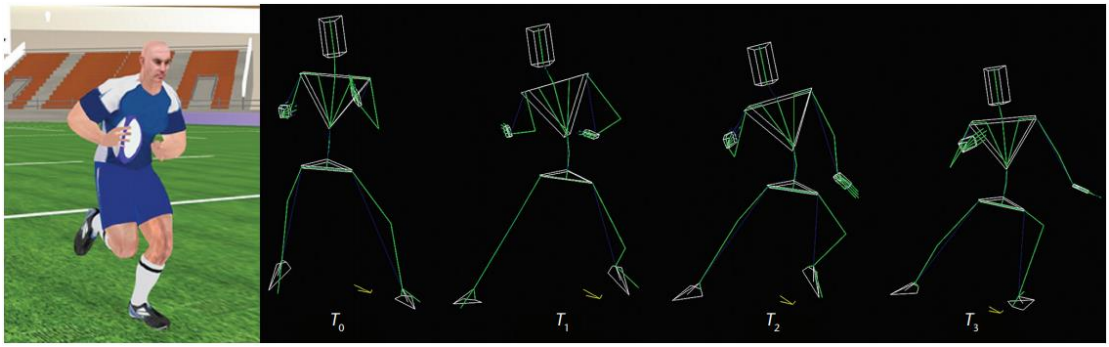


Figure 15: Using temporal occlusion in the rugby case study: (left) a virtual rugby player and (right) instances of this player at different cutoff times **(Bideau, Kulpa, Vignais, Brault, Multon, & Craig, 2010).**

Although many of the virtual-reality-based systems are too expensive for most average schools in the US and EU, with the new commercialized products for Physical Education can be paired in an accessible way, so that students can practice in otherwise dangerous and inaccessible physical activities, through a safe virtual environment. That implies that there should be methods for accurate motion tracking so that their movements can be adequately represented through the graphical imagery. Several systems have been developed in order to be introduced experimentally to schools and universities in the future.

These systems typically address basic PE national standards of the United States of America (Finkenberg & Mohsen, 2003), such as motor skill development (standard 1 of the US national standards PE) and participation in physical activity and fitness-related exercises (standards 3 and 4 of the US national standards for PE). They are also beneficial learning experiences providing a less hazardous and more forgiving environment by decreasing the complexity of social interactions especially for students with intellectual disability (Standen & Brown, 2006).

A.2.5 Physiotherapy and Rehabilitation

Some of the most beneficial applications of the existent technology on human body and limb tracking are based in studies of diseases that show symptoms of motion disorder. Such applications include gait tracking and symptom-disease correlation through statistical clustering and body movement rehabilitation through physiotherapy support using body tracking software.

Gait Analysis has been a major field of research in medicine over the last few years. According to recent studies, the gait patterns of humans that have been through, or suffering from certain diseases, such as a stroke (Mirelman, Patritti, Bonato, & Deutch, 2010) or diabetic neuropathy (Bacarin, Sacco, & Henning, 2009) are very characteristic and can be easily recognized using a combination of human movement monitoring systems (usually optical) and pressure plates. Therefore research is made on predicting probable future diseases that a subject might have, or has already symptoms of, through gait and stride analysis.

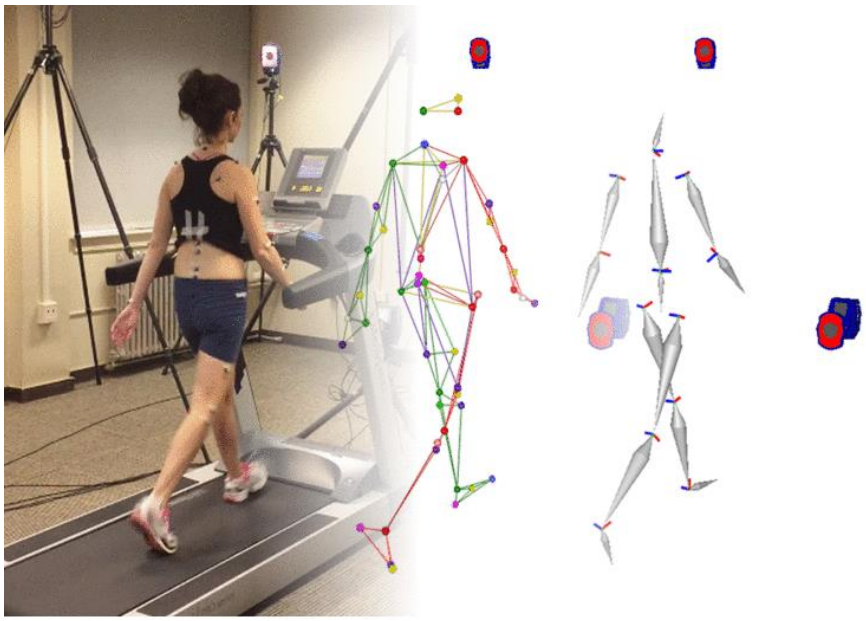


Figure 16: Dr. Robert Catena's Gait and Posture Biomechanics Lab **(Washington State University)**

Physiotherapy support on the other hand is based on muscle flection and extension through basic stretching and training exercises. Thus, optical methods are avoided. A very discrete example is Biofeedback, a technique that uses electrical sensors that help you receive information (feedback) about your body (bio). Biofeedback methods include brain wave, breathing, temperature, sweat glands, heart rate and muscle monitoring using either interactive computer or mobile device programs or wearable devices (Mayo Clinic, 2018). The aim of this method is to train your body through harmless electrical signals, by tracking your movement, physique and behavior throughout your day, or through several physiotherapy sessions. It is usually used on patients suffering from autism, Asperger syndrome, epilepsy, ADHD, mood and attention disorders (JuniorMed Centrum Medyczne).

Figure 17: EEG Biofeedback device used on child patient **(JuniorMed Centrum Medyczne)**

A.2.6 Surveillance and Security

"The capture and analysis of surveillance footage has been an indispensable tool for U.S. counterterrorism and law enforcement in the past decade" (Greenemeier, 2011). Combined with face recognition technology, human body motion capture methods have been used by law enforcement over the past two decades to identify possible threats, recognize suspects and dangerous individuals and provide predictive safety over a fully monitored area. Although the technology has also been used for rather inhumane military actions, its basic human motion recognition methods have, lately, been commercially available through security systems.

From motion activated infrared sensors to image processing cameras this field is mainly based on the optical recognition methods. The simpler systems use infrared technology, only to identify a change in a static system/area and report movement. More sophisticated complex and costly systems use cameras with a processing unit as simple as a Raspberry Pie™ (Rosebrock, 2015) to identify, not only movement in the area, but also human beings and body motion characteristics.

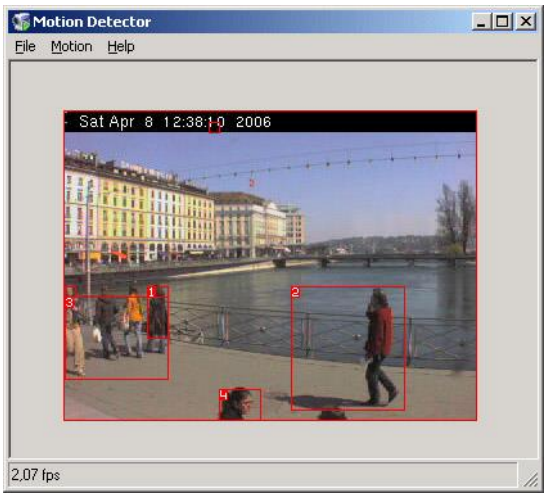


Figure 18: Simple DIY Human Motion Detecting Program **(Kirillov, 2007)**

Moreover, research studies made by New York University (NYU), on improving surveillance software, have made possible the extraction of data through body and face movement recognition for better software observation techniques of human behavior (Williams, Taylor, Smolskiy, & Bregler, 2010), 3D Skeletal reconstruction from low-resolution multi-view images (Rana, Taylor, Spiro, & Bregler, 2012) and even unnatural burden or outfit-dependent movement recognition for security purpose (Greenemeier, 2011).

A.2.7 Training and Education

A rather recent application of motion capture technology in employee/worker training has been a point of focus for many researchers in the field, aiming to prove that virtual reality (VR) training assisted by MoCap methods can efficiently and safely transfers technical skills as part of an education program (Seymour, et al., 2002).

Most studies focus on possibly hazardous or life threatening situations, such as use of heavy machinery (SANLAB, 2015), surgical simulations (Grantcharov, Kristiansen, Bendix, Bardram, Rosenberg, & Funch-Jensen, 2004) and worker-robot collaboration (Kontrazis, 2018). The idea lies behind monitoring the physical movement of the worker and translating them in a hazard-free, forgiving environment, so that he can get familiarized with the equipment without being harmed or causing any harm.

Figure 19: SANLAB SimPro3™ Heavy Machinery Simulator

The most common MoCap methods for these applications are usually through IMU sensors and cameras in VR headsets, along with certain handles, dials, tools, controllers and other hardware devices that are commonly used for the task. These devices usually are the same as the ones used in a real situation, for user experience correspondence, except for the sensors implemented in them, for electronic interpolation of the user's movements.

A.2.8 Video Games and Virtual Reality

With the blemish of the videogame industry over the last four decades and the increase in demand for more realistic videogame graphic design, environments and immersion, VR technology has become, at least since 2010, a commercial and very profitable market accessible, now, to the public. The idea of Virtual Reality through the use of VR Headsets has been a long-thought idea for over 60 years. During the 60's, trying to create a breakthrough in the movie industry M.Heilig (The Franklin Institute) was the first to patent a VR headset used for entertainment. Since then, through many drafts, ideas, patents and time-consuming research from renowned institutions such as the NASA Ames Research Center and Harvard University the collaboration between haptic and movement tracking systems, along with Virtual Reality has been achieved.

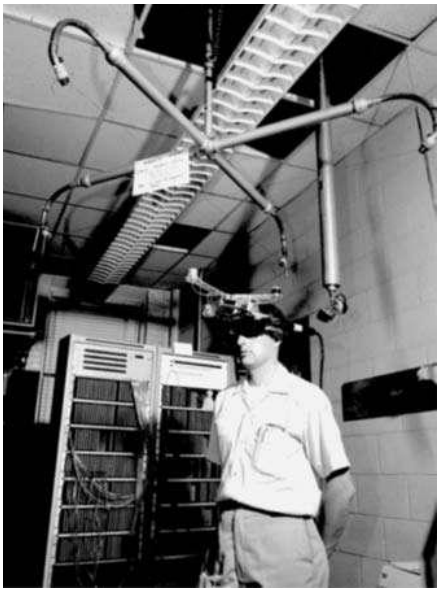


Figure 20: Early Development of a VR headset by Ivan Sutherland, University of Harvard, 1967, **(Lowood, 2018)**

Nowadays, modern VR systems have become quite sophisticated aiming to, not only provide visual imaging of a VR environment, but also to allow the user to interact naturally with objects through more accurate body movement monitoring. One example is TrackIR™ by NaturalPoint; a company specialized in eye tracking. TrackIR™, created back in 2002, was one of the first commercial products to be able to track head movement using a fully integrated optical camera tracking system connected to a PC, so as to provide greater VR Simulator Game immersion (Richardson, 2003). The system has, since then, been improved several times and has recently released its 5th variation TrackIR 5™.

Figure 21: NaturalPoint TrackIR 5™ [9]

We, now, have fully integrated VR headsets and suits that have come way further, but are still based on, Heilig's idea. Such examples are Rift™, Gear VR™ and Go™, from Oculus and the Vive™ from HTC, which use two completely different tracking methods and still can incorporate human body movement relatively accurately in a fully developed VR environment. Moreover the company X-Sens has created a full VR suit used, for the moment, mostly in research applications.

Figure 22: Fully Integrated VR Commercial Products: Oculus Rift™ [10] (left), Xsens MVN™ [11] (right)

Figure 23: HTC Vive™ [12]

[9] Retrieved 10/15/2018, from https://www.naturalpoint.com/trackir/
[10] Retrieved 10/15/2018, from https://www.oculus.com/rift/
[11] Retrieved 10/15/2018, from https://www.xsens.com/products/xsens-mvn-animate/
[12] Retrieved 10/15/2018, from https://www.vive.com/us/

B. Basic Theory and Background of the Joint Rotation Measurement System

B.1. Human upper-body anatomy

The positioning of the sensors for optimal movement monitoring is based on the skeletal and muscular anatomy of the upper human body. This sub-chapter focuses on explaining the choices of sensor placement through basic robotic theory and modeling of the body.

B.1.1 Upper-body skeletal anatomy and measure point positioning

Firstly, we should define the three planes of motion of the human body. These are the median sagittal, coronal and transverse planes (Kahle, Leonhardt, Platzer, Palmer, & Platzer, 2004), which correspond to the side, frontal and axial plane respectably. There is more terminology associated with the planes of the human body which can be seen in Figures 24, 25).

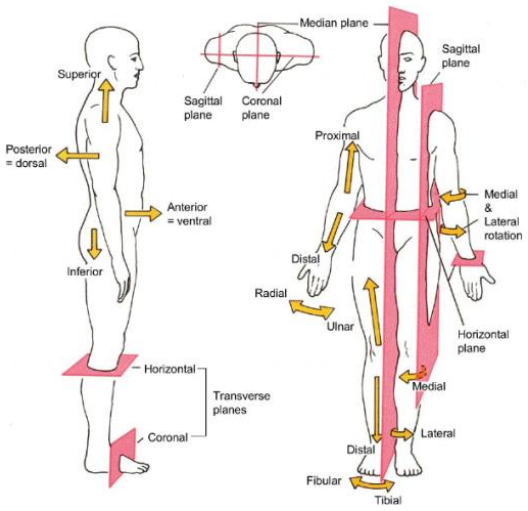


Figure 24: The Planes of the Human Body[13]

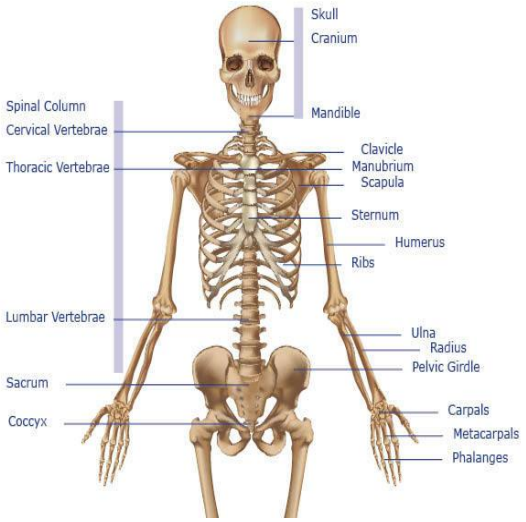


Figure 25: Basic Skeletal Terminology of the Upper Human Body[14]

[13] Retrieved 12/30/2018 from http://www.mccc.edu/~behrensb/documents/BasicBiombjb2011.pdf
[14] Retrieved 12/30/2018 from https://www.healthpages.org/anatomy-function/musculoskeletal-system-bones-joints-cartilage-ligaments/

By defining each bone, or group of bones of the upper human body as a link, its corresponding muscles as motion limited motors and each human joint as a robotic joint we can create an accurate model of the human body (Figure 26).

Due to our body's complexity and the plethora of joints, especially in the spinal column, the accurate monitoring of every joint is rather redundant, very costly, and nearly impossible. Therefore some simplifications to the original model of Figure 26 need to be made, so as to have a reasonable amount of sensors and data to process.

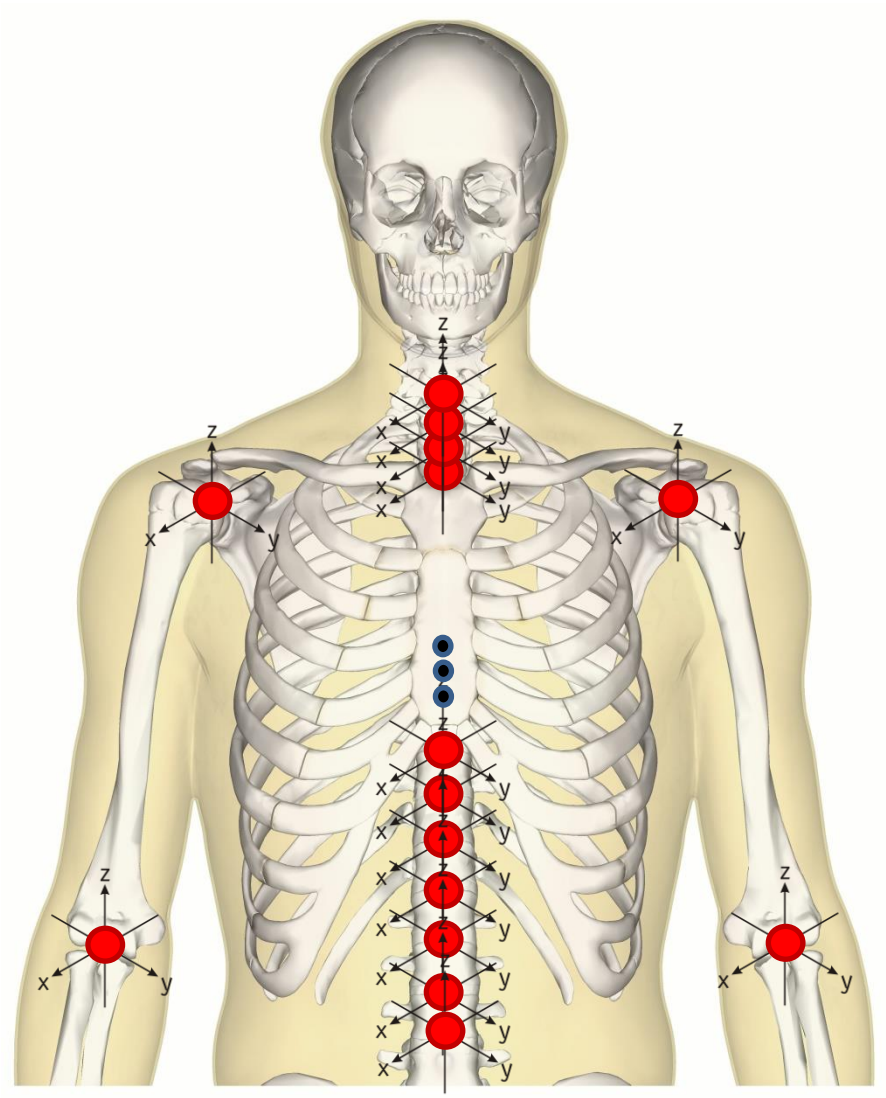


Figure 26: Model of the Joints of the Upper Human Body

There is no limitation to the amount of sensors one can use for monitoring the body, but there is a fine line based on efficiency and processing unit data handling. Based on a low-cost model, we chose to make some key assumptions to simplify our model and limit the sensor number to nine:

- First and foremost, due to our previous build's collaboration with unity and the Oculus Rift™ we chose to neglect the movement of all the vertebrae above the

parallel transverse plane defined by the clavicle, such as the cervical vertebrae and the first two to four thoracic (Figure 27) based on the subject wearing the rig.
Moreover the spinal column and the rest of the vertebral joints are monitored by three sensors creating a triangle shaped monitoring area; One located in the Lumbar or lower Thoracic area and two axisymmetric locations located in the top Thoracic area (Figure 27).

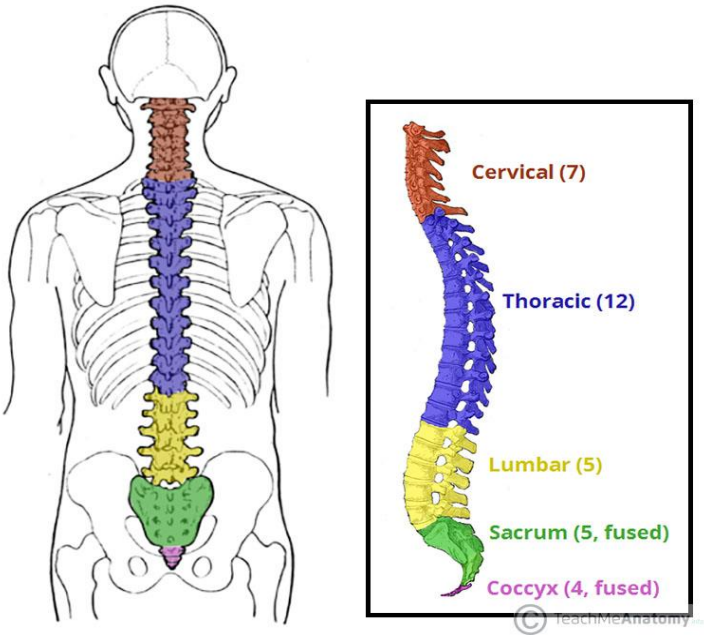


Figure 27: Vertebral Column[15]

- Secondly due to lack of finger based sensors and availability of products focused entirely on human finger motion, tracking of finger movement is not monitored and is left to be added in collaboration with already existing finger tracking market products such as the Manus glove. Therefore the palm and fingers are considered as one unified link.

- Last but not least, each joint except for the central triangle, defining the spinal joints, is monitored by two sensors, one located before the joint and one after (Figure 28). The global comparison of the data of the sensors can accurately measure the movement of the links and thus translate it, to rotations of the joints. Therefore measure points 2 through 9 (Figure 29) create unique pairs of joint defining data as described in Table 3.

[15]Retrieved 12/31/2018 from https://teachmeanatomy.info/back/bones/vertebral-column/

Table 3: Measure Points Pairs and Body Joints Correlation

Pair	Description	Planes of Motion
2-4 3-7	**Right Shoulder** **Left Shoulder**	**3**
4-5 7-8	**Right Elbow** **Left Elbow**	1
5-6 8-9	**Right Wrist** **Left Wrist**	2

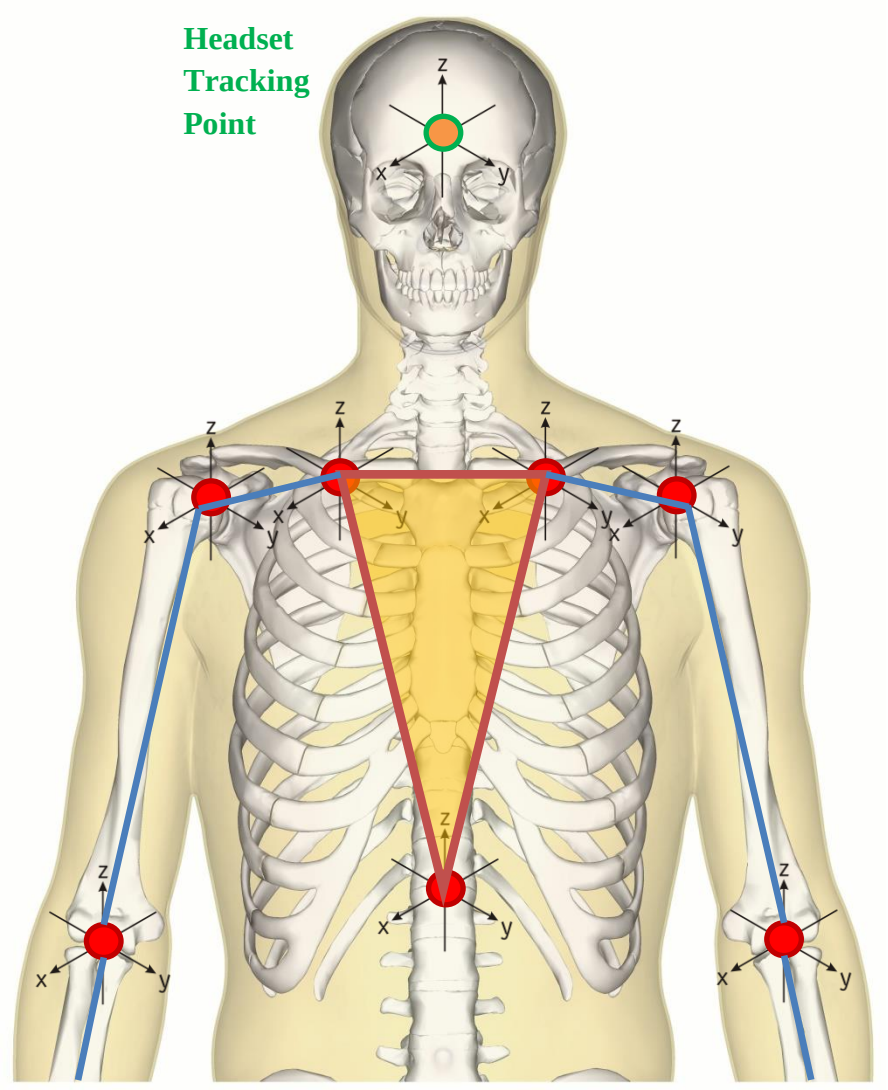


Figure 28: Simplified Model of the Joints of the Upper Human Body

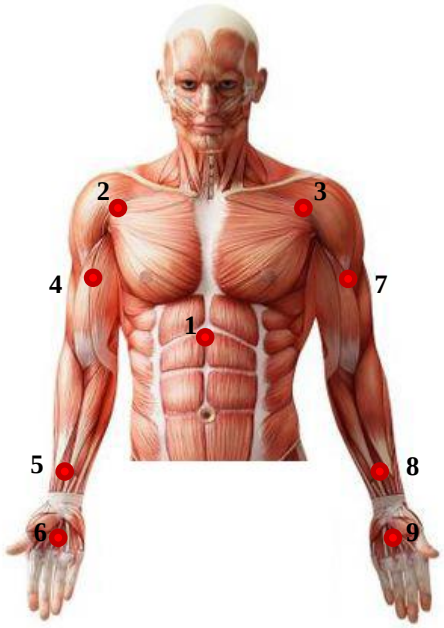


Figure 29: Chosen Measuring Points[16]

[16]Retrieved 1/7/2018 from http://www.jimmybluff.com/bluff-technique-classes/upper-body

B.1.2 Upper-body muscles, ligaments and motion capabilities

Based on the muscles and ligaments that engulf it, each type of body joint can be treated as a unique robotic joint with certain motion restrictions.

The shoulder joint is comprised by five smaller joints that can cooperatively reproduce the movement of a spherical joint (Desroches, Aissaoui, & Bourbonnais, 2006), in forms of flexion/extension, circumduction and a single type abduction/adduction (Wu, et al., 2005) (Figure 30).

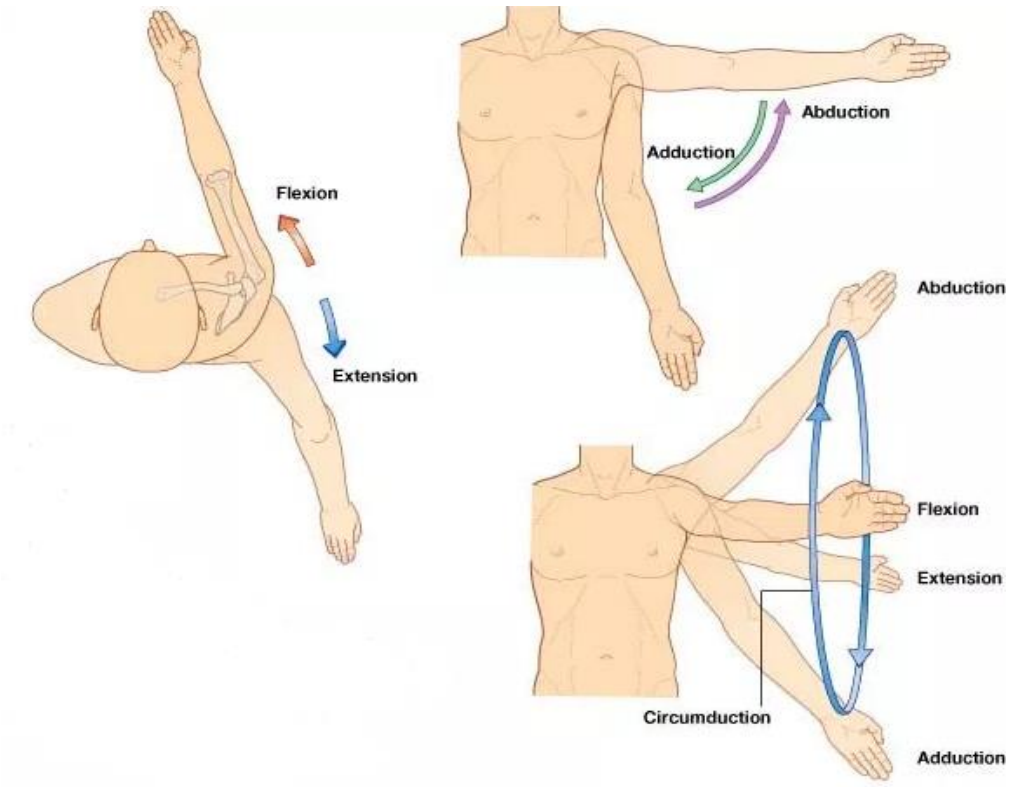


Figure 30: Shoulder Movement Types[17]

Moreover using the muscles around the clavicle, the sternum and the scapula, the shoulder can recreate a radial movement called protraction/retraction in the sagittal plane and elevation/depression in the coronal plane (Figure 31). This movement, due to the high ratio between radius and circular path length, could be considered nominally as the movement of a linear joint. Therefore the shoulder can normally reproduce motion in 3 different planes at any starting position.

[17]Retrieved 12/31/2018 from http://bestperformancegroup.com/?page_id=966

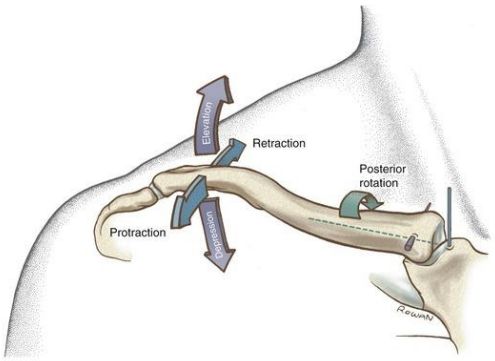


Figure 31: Clavicle Based Shoulder Movement Types[18]

The elbow joints are treated as tilted rotational joints and can reproduce rotational movement on a single plane that diverges approximately at an angle of 10^{o}-15^{o} from the transverse plane (Figure 32a). The vector of that plane is called distal or cubitus valgus/varus axis and is defined by the cubitus valgus/varus angle, which differs on each subject tested (Figure 32b). Moreover the elbow can rotate axially on the distal arm axis due to ligament and muscle elasticity. This extra degree of freedom can easily be monitored and is called the supination/pronation movement.

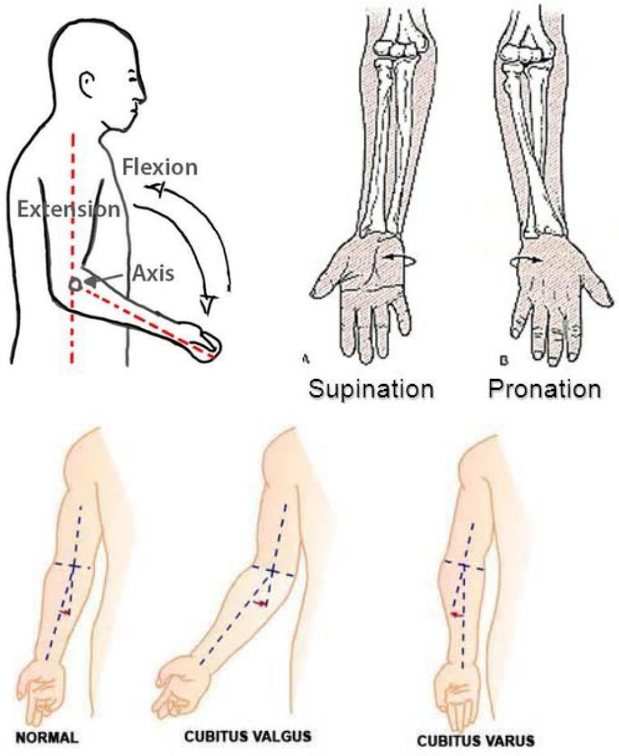


Figure 32: a. Movements of a Shoulder Joint (up)[19] ***b. The Cubitus Axis (down)***

[18]Retrieved 12/31/2018 from https://clinicalgate.com/shoulder-complex/ and 1/10/2019 from "Elbow Injuries Critical link in kinetic chain of upper extremity." Presentation by Khalil-Heckler.

[19]Retrieved 1/3/2019 from https://www.researchgate.net/figure/Movements-of-shoulder-and-elbow_fig6_288154412

Lastly the wrist joints are also treated as spherical joints with certain limitations due to surrounding bones, muscles and ligaments (Rainbow, Wolff, Crisco, & Wolfe, 2016). They can recreate movement in two different planes in the form of radial/ulnar deviation (or adduction/abduction) and flexion/extension (Figure 33). In contrary they are limited in their axial rotation, along the distal axis and cannot reproduce significant motion, limiting the wrist joints to just 2 planes of motion.

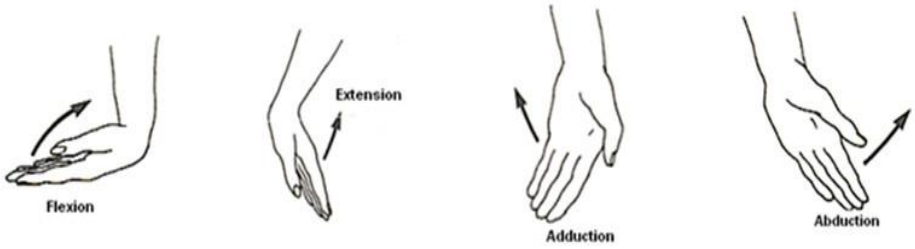


Figure 33: Movements of a Wrist Joint[20]

Due to the existence of muscles, skin and ligaments around the joints, all types of movements that were mentioned earlier pose some restrictions. Movements in each joint are characterized hereon as prime or secondary movements, where prime are those who reproduce substantial range of motion that is significant for the joint and secondary, those that are reproduced due to the elasticity of the ligaments or movement of adjacent bones.

[20]Retrieved 1/3/2019 from https://www.braceaccess.com/wrist-anatomy-carpal-tunnel-syndrome/

B.1.3 Upper-body motion limitations

The restrictions of the prime movements differ from subject to subject in terms of maximum and minimum angle and end positions, but are all defined by the same conditions.

For the shoulder, the adduction/abduction movement can cover a range of around 180° or -30° to +180° depending on the subject, but is interlocked with the rotary and scapular movement for the person to be able to reach the maximum range (Panjabi, Oxland, Yamamoto, & Crisco, 1994). In the normal/anatomical position as seen in Figure 29, the shoulder can achieve a rotational movement that has a range of 50° due to the humerus bone colliding with the glenoid cavity (Figure 34), whereas if, the shoulder is already in a +90° abductive position the range is increased to 120° or -70° to +50° differing from subject to subject and reaching up to -90°/+60° (Pruthviraj, 2012).

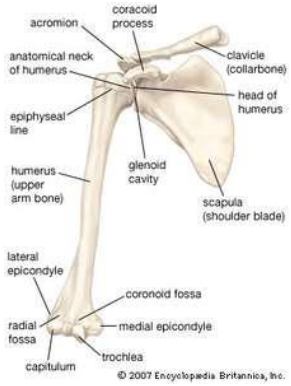


Figure 34: Skeletal Anatomy of the Shoulder[21]

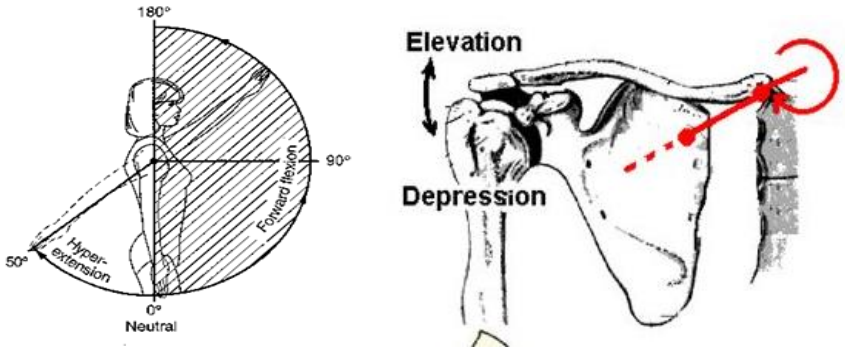


Figure 35: Shoulder Movement Limitations* (Cayson)*: Flexion/Extension (left), Elevation/Depression (right)

For flexion and extension of the shoulder, the human body can achieve a range from -50° to +180° (Figure 35) restricted again by parts of the humerus colliding with the glenoid cavity (Figure 34). Lastly the range of motion for the clavicle-assisted movement is limited within ±15° for protraction/retraction and -5° to +45° for elevation/depression. Circumdaction is a movement combining all those previous movement to achieve a full range of 360° (Hislop, Avers, & Brown, 2013).

[21] Retrieved 1/11/2019 from https://www.britannica.com/science/humerus

Table 4: Shoulder Movement Range

Type of Movement	Healthy Lower Limit of Rotation	Healthy Upper Limit of Rotation
Abduction/Adduction	-30°	+150°
Inner/Outer Rotation	-70°	+50°
Flexion/Extension	-50°	+180°
Protraction/Retraction	-15°	+15°
Elevation/Depression	-5°	+45°
Combined Circumduction	-360°	+360°

The elbow has a single rotary flexion/extension movement that ranges from +0° to +150° and with passive help can be extended from -10° to +150° due to hyperextension. Also the supination/pronation movement ranges from -80° to +80° and can reach -90° to +90° (Figure 36) depending on the subject (Range of Joint Motion Evaluation Chart, 2014).

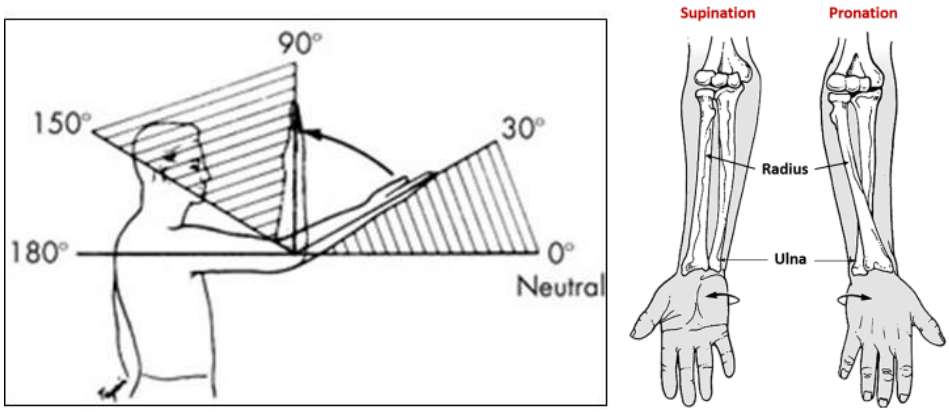


Figure 36: Elbow Movement Limitations: Flexion/Extension (Sullivan) (left), Supination/Pronation[22] (right)

Table 5: Elbow Movement Range

Type of Movement	Healthy Lower Limit of Rotation	Healthy Upper Limit of Rotation
Flexion/Extension	-10°	+150°
Supination/Pronation	-90°	+90°

[22] Retrieved 1/11/2019 from http://www.anatomyqa.com/anatomy/upperlimb/radioulnar-joints-supination-and-pronation/

As for the wrist radial/ulnar deviation reaches from -20° to +30° as seen in Figure 37 and the flexion/extension from -50° to +60° and can reach up to ±60° due to hyperextension.

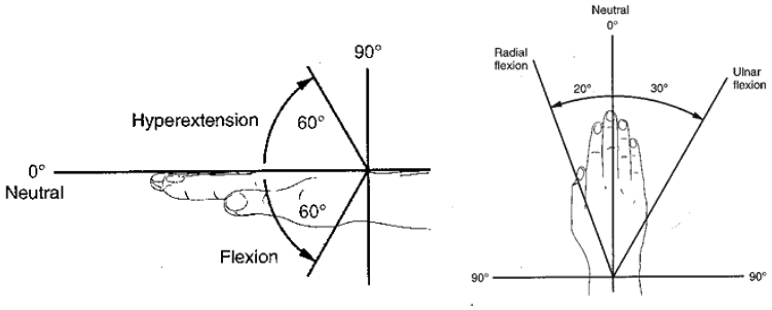


Figure 37: Wrist Movement Limitations[23]: Flexion/Extension (left), Radial/Ulnar deviation (right)

Table 6: Wrist Movement Range

Type of Movement	Healthy Lower Limit of Rotation	Healthy Upper Limit of Rotation
Flexion/Extension	-60°	+60°
Radial/Ulnar Deviation	-20°	+30°

[23] Retrieved 1/10/2019 from http://www.wellnesswithinclinic.com/docs/rom_lab.html?fbclid=IwAR1O38dIyV7BQrLo49y7KbmlpYgC46856C6NfEGCl68Rw1ySGyaczabPPIw

B.2. Mathematical and Physics Background

B.2.1 Rotation and Position

To describe the position of a point in three-dimensional spatial geometry, a vector composed of three base vectors i, j and k is used. Vectors are written as a triplet of real number and therefore as multiplication of the orthonormal base vectors:

i = (1 , 0 , 0) j = (0 , 1 , 0) k = (0 , 0 , 1)

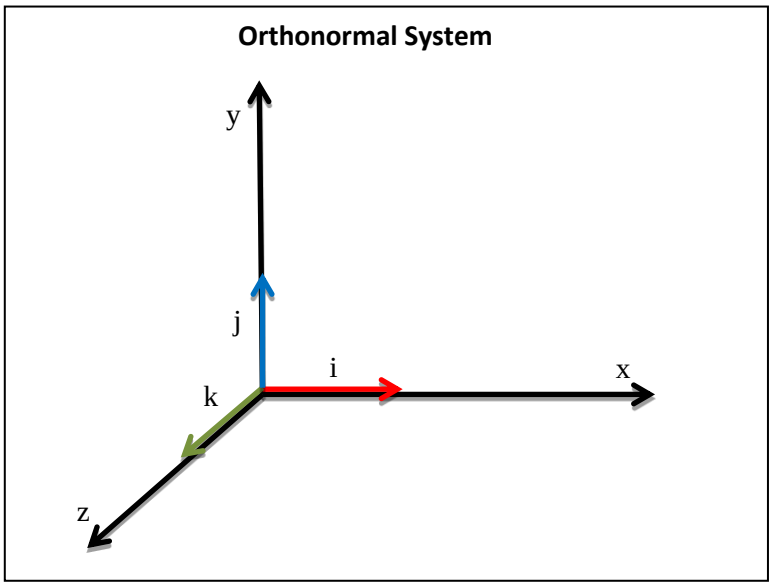


Figure 38: Orthonormal 3D System

To make the transition from defining just a single point, to defining the orientation and position of a full body object we need to add to the already existing tri-axial x_0, y_0, z_0, system. There are many methods to achieve an accurate description, the most basic of which is a 9x9 matrix based on the object's frame/coordinate system. Therefore we have three new object oriented axes x_1,y_1 and z_1 which can be described in the orthonormal base system by three vectors called column vectors ${}^0x_1$, ${}^0y_1$, ${}^0z_1$:

$$ {}^0x_1 = \begin{bmatrix} \cos(x_1, x_0) \\ \cos(x_1, y_0) \\ \cos(x_1, z_0) \end{bmatrix} = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix}, {}^0y_1 = \begin{bmatrix} \cos(y_1, x_0) \\ \cos(y_1, y_0) \\ \cos(y_1, z_0) \end{bmatrix} = \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix}, {}^0z_1 = \begin{bmatrix} \cos(z_1, x_0) \\ \cos(z_1, y_0) \\ \cos(z_1, z_0) \end{bmatrix} = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} $$

$$ {}^0R_1 = \begin{bmatrix} {}^0x_1, & {}^0y_1, & {}^0z_1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

(Siciliano, Sciavicco, Villani, & Oriolo, 2010)

B.2.2 The Unit Quaternion

Based on the previous facts, we can understand that only a simple vector cannot describe efficiently the orientation and position of a solid object in 3D-space as it lacks appropriate information. As a solution to that many different methods have been proposed, except for the 3x3 matrix that was mentioned. This method requires 9 different variables to be stored and processed and therefore is inefficient. Other more efficient methods are the Euler rotational angles, the unit Quaternion, the Tait-Bryan angles and an axis-angle (or vector-angle) pair (Bishop, 2008).

For the purpose of this thesis we will explain and analyze only the Unit Quaternion method, since it was the method used to describe the position and orientation of the parts of the human body.

A Quaternion (Hamilton, 1866), as the name suggests, is another method using a quadruplet of numbers to describe the position and orientation of the object and is written as:

$q = (q_0, q_1, q_2, q_3)$ where q_0, q_1, q_2, q_3 are real numbers or scalars.

(Kuipers, 1999)

If the rotation matrix R that was mentioned in the previous chapter is given, for a rotation of θ, then we can calculate the quaternion using a vector $\vec{q}$ and one parameter q_4, were:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1,$$

$$q_4 = \frac{1}{2}\sqrt{(1 + r_{11} + r_{22} + r_{33}}, \text{ for } 0 \leq \theta \leq \pi$$

$$\vec{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{4q_4}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}, \text{ with } q_4 \neq 0$$

And if $q_4 = 0$ then $\vec{q} = k$, with k being the axis of the rotation.

On the other hand, if the quaternion is given then we can calculate the rotation matrix R for a rotation of θ degrees, around the axis k as:

$$\vec{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{k}\sin(\frac{\theta}{2}), \; q_4 = \cos(\frac{\theta}{2})$$

$$R_\theta = (2{q_4}^2 - 1)I_3 + 2q_4\vec{q}^x + 2\vec{q}\vec{q}^T$$

(Papadopoulos, 2017)

B.3. Previous Builds

B.3.1 Development and Evaluation of a Wearable Motion Tracking System, to Support Hand-Tool Design

The origins of the project start with the idea of M. Karakikes former diploma student in the Ergonomics Unit and his thesis on “Development and Evaluation of a Wearable Motion Tracking System, to Support Hand-Tool Design”. The idea lied on the consensus that various professionals that are occupied with everyday tasks using hand-operated tools, such as industrial workers, surgeons and craftsmen, could, with long term improper use of the tool, risk upper-body musculoskeletal disorders.

An abundance of studies, publications and ergonomic principles of design Indicate that maintaining extreme positions, in any type of movement, of any joint, for even short-term periodical tasks, can be directly associated with such problems. Therefore, a goal of creating an inexpensive, wearable system to track wrist and forearm joint motion was set.

Based on similar builds, the system was based on data collected by IMUs, recording wrist and forearm posture and assisting on evaluating tool design and use. The system designed, included only 3 processing units and a main Arduino processing platform. The IMUs were used for tracking forearm and wrist movements and considered the elbow joint as fixed in position but still operational in all of its movements.

The IMUs chosen were from the MPU-6250 variant providing 6 DoFs and no magnetometer data since they did not include one.

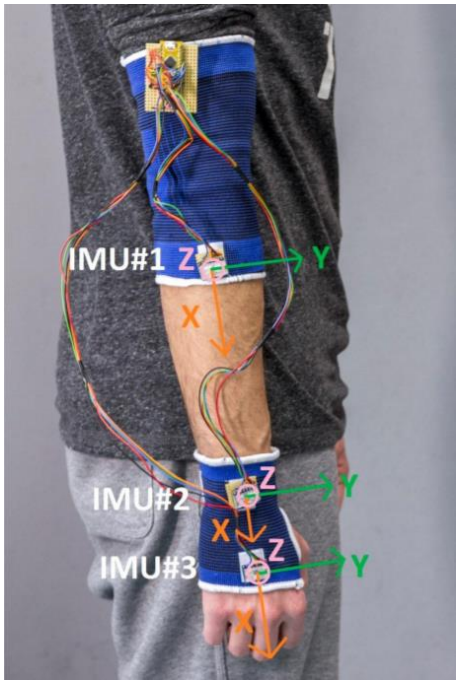


Figure 39: M. Karakikes' Wearable Motion Tracking System (Karakikes, 2017)

Following on from the assembly and coding of the system, a methodology for comparing alternative design solutions, using this wearable system, was proposed, through a study on monitoring three different shaving razor handles methods of use, so as to determine the existence of a correlation between joint movement and handle design and recognize the effect of design features.

The results verified the concept of the previous publication although the joints never achieved extreme positions. Therefore this thesis was the initiation of a long term study and development of a fully integrated, protected, 9DoF, measuring system of the arm.

(Karakikes, 2017)

B.3.2 Development and evaluation of a wearable motion tracking system for sensorimotor tasks in VR environments

The next step for the advancement of the project was the thesis of another diploma student of the lab A. Mourelatos on the "Development and evaluation of a wearable motion tracking system for sensorimotor tasks in VR environments". The main idea was focused on integrating the previous system in a 3D environment through Unity game engine, while making basic coding improvements. The system was used in conjunction with an Oculus Rift DK2 and an environment designed in the Unity Game Engine, to create a Virtual Environment (VE) "shooting target practice" task. The original game was adapted from a rigid shooting arm to a jointed arm so as to simulate better movement control.

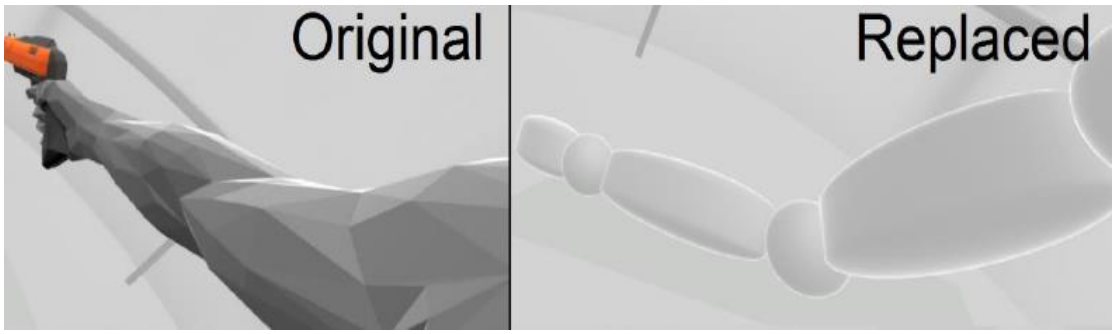


Figure 40: Comparison of the original vs the adapted for joint motion control game hand design

(Mourelatos, 2018)

In this thesis a series of experiments were designed to determine whether the visibility of one's limb movements in real time in a VE, improves the effectiveness in the execution of sensorimotor tasks within that environment. Moreover the speed and ease of the assimilation of a virtual representation of an arm with the subject's body image and the incorporation of this representation into their body schemas, was evaluated.

The results of the experiments showed that visibility did not appear to have a significant effect on performance, but opened the way for further investigation of training and performance controlling tasks.

(Mourelatos, 2018)

C. New Joint Rotation Measurement System

C.1. Mechanical Design Schematics

C.1.1 MPU Casing Design and Manufacture

The idea of creating a modular MPU system requires a robust design for the protection of the MPU that will allow it to be easily fitted and securely connected with cables, in the proper body position. Therefore a casing needed to be developed. The main characteristics of the case that were required were:

- Functional design able to secure the MPU in place so that minimal skin and clothing artefact errors are calculated.
- Enough space to withhold the MPU, connection, a 6 strand cable and possible extra modules that might be included when further development is made (e.g. An Rx/Tx module, Chapters C.3 and E.1).
- A robust material to protect the MPU from misuse errors.
- An available, easy, flexible and fast manufacturing method.
- An access point to the inner part of the enclosure, so that maintenance and repairs can be easily made, without permanently damaging the case.

After assessing the available resources and trying to restrain any excess expenses, so that the system could remain, a low cost experimental setup, we concluded in using the NTUA Manufacturing Technology Laboratory's Cube X tm 3D Printer (3D Systems, 2013) to create a custom-made casing design.

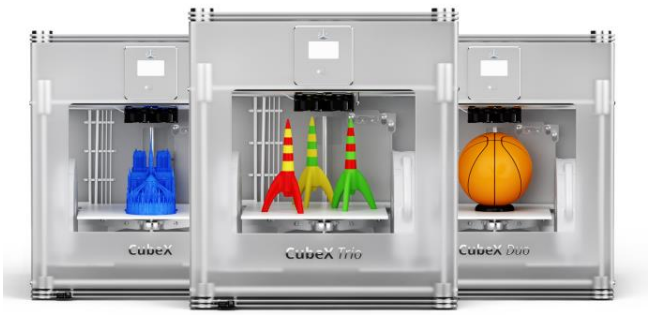


Figure 41: CubeX 3D Printer by 3D Systems

The casing was designed using the Dassault Solidworks CAD software and was manufactured with special issued PLA plastic for Fused Deposition Modeling (FDM). To solve the previous problems the casing design included:

- A socket designed to withhold the MPU-9250 securely in place, and protect it from any outer contacts, forces or collisions, that might harm the breakout board and its electronics.
- A slide-fit enclosure/top that can hold a 6-strand cable and protect the top part of the MPU. The top is also designed for future installation of a normal sized Rx/Tx breakout board.
- Supportive wing-like extrudes that slightly envelop the body part that the MPU is situated on.
- Dimensional adjustments to predict the PLAs shrink factor.
- Openings on the extrudes so that flat braided straps can be mounted, to hold the MPU in place

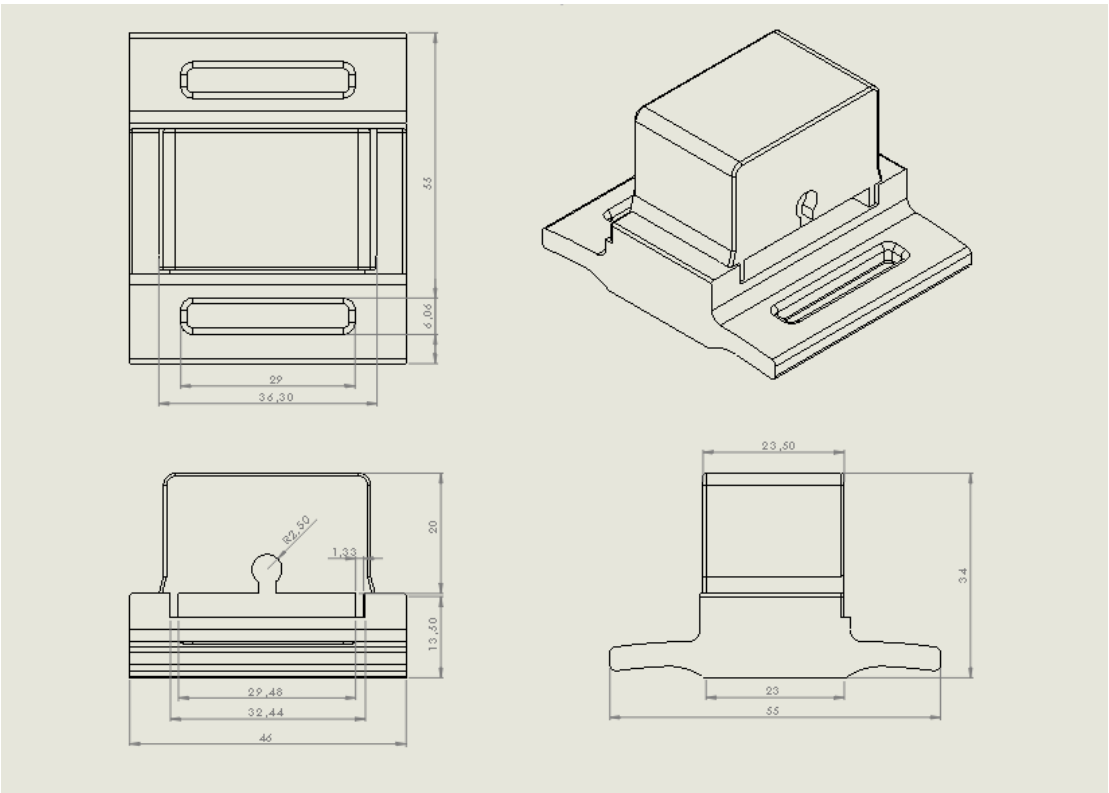


Figure 42: Mechanical Design of MPU Casing

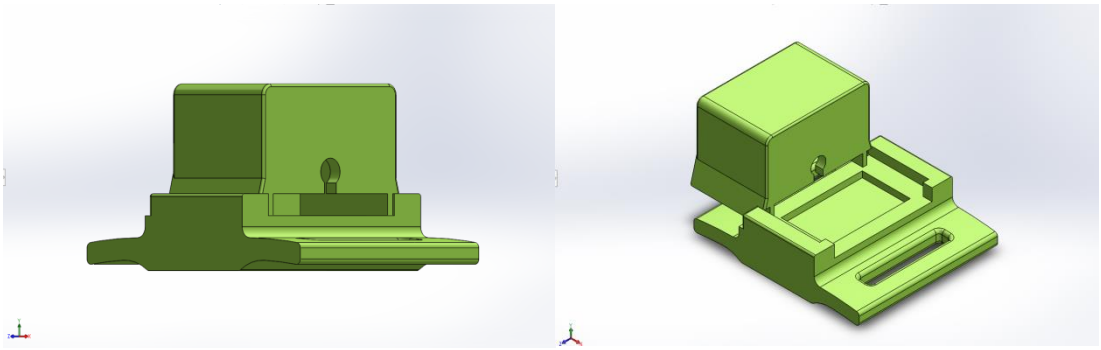


Figure 43: Assembly of MPU Casing with Slide-fit Protective Top

C.2 Electrical Design Schematics

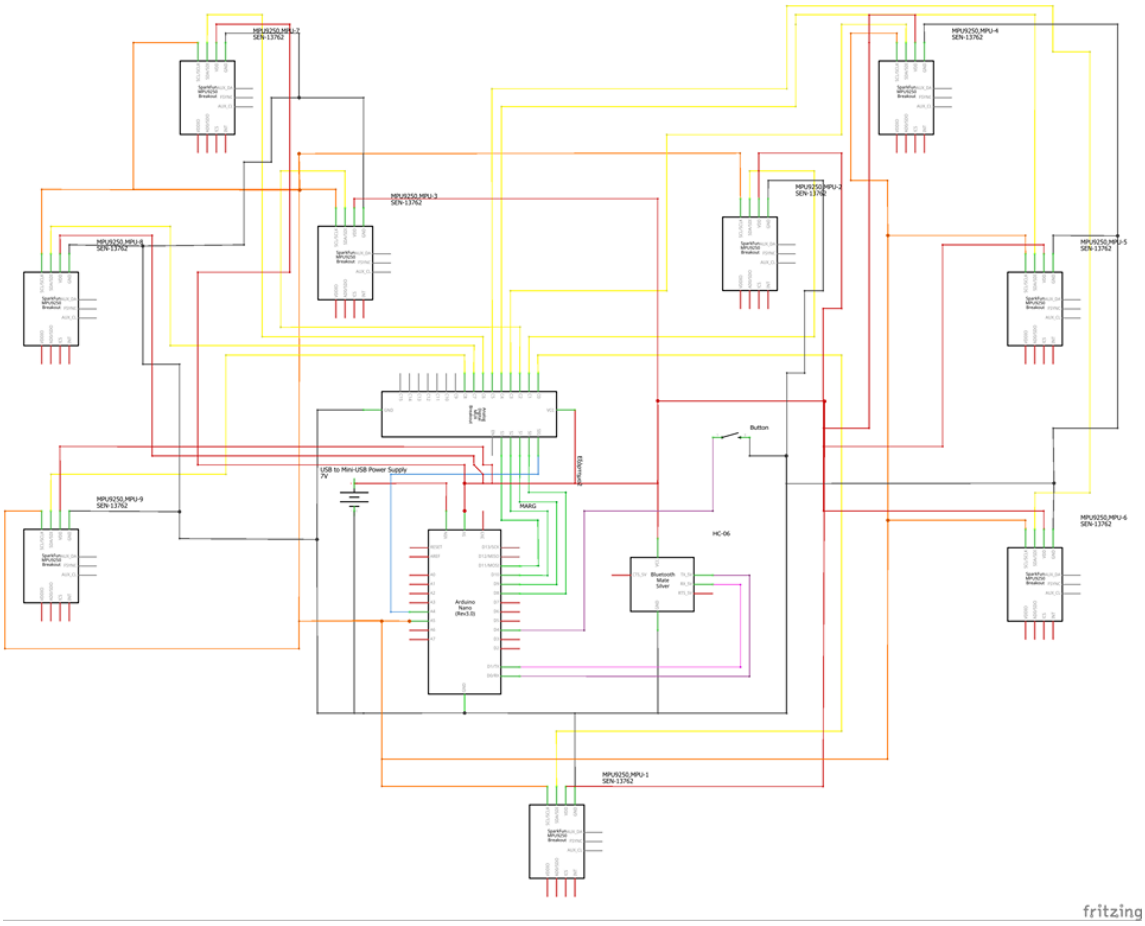


Figure 44: Electrical Schematics of the new Upper Body MARG System

C.3 Prototype Design and Concept

Some pictures taken from the prototype design of the casing and the setup can be seen in this part. Note that in Figure 46, a possible mount for an Rf module is displayed (Figure 46 top right).

Figure 45: Photo of the 3D-Printed Prototype Casing

Figure 46: Fitting the Various Components in the Casing

C.4 Algorithm

C.4.1 Madgwick Filter

In order to measure movement in three dimensions with a MARG type sensor a fusion algorithm of all three types of measurements (Magnetic, Angular Rate, and Gravity), are needed. The difference between a MARG sensor, such as the MPU-9250, and an IMU is that the MARG incorporates a tri-axis magnetometer. Therefore a simple IMU can only measure an attitude relative to the direction of gravity (Madgwick, 2010).

The Kalman Filter (Kalman, 1960), and the Extended Kalman Filter (EKF) is the main orientation filter that was used for commercial IMUs until recently, due to its accuracy. Despite that, the filter had some disadvantages such as complicate implementation, high sampling rate demand and large state vectors to linearize certain problems. Sebastian Madgwick, on the other hand, created a novel orientation filter applicable to both IMUs and MARGs to address the issues of computational loads and parameter tuning associated with the Kalman-based approaches (Madgwick, 2010).

The Madgwick Filter was based on quaternion theory and was derived as follows:

- A tri-axis gyroscope will measure the angular rate about the x, y and z axes of the senor frame, termed ω_x, ω_y and ω_z respectively, with a Δt sampling period of the gyroscope.

- By creating an initial guess of orientation ${}^{S}_{E}\hat{q}_0$ and a step-size μ, the filter uses a gradient descent algorithm, as one of the simplest and easiest to implement and compute optimization methods for a quaternion system (Xu, Xia, & Mandic, 2016).

- If we assume that the direction of gravity defines our vertical axis (z), the normalized accelerometer measurements are good approaches to the direction of the field and that the earth's magnetic field has components in one horizontal axis and the vertical axis with a local declination (Declination in NTUA Lab of Cognitive Ergonomics is 4° 29' East with an annual change of 5.6 '/y East in 1/22/2019 (Natural Resources Canada, 2017)) we can create a simplified version of the gradient decent algorithm expression.

- These previous basics are applied to most common fusion algorithms. Stephan Madgwick's approach was perceived to decrease the computation load of optimization methods using multiple iterations and adjusting step sizes μ to an optimal sample step value of μ_t, so as to ensure the convergence rate of the quaternion estimation. It also assists by applying certain weights γ to each orientation calculation, with an optimal weight of γ_t.

- Then we proceed in finding an optimal γ_t and a parameter, β, to ensure the weighted divergence of one of the orientation calculations, is equal to the weighted convergence of the other.

- Lastly, some extra steps are taken to compensate for magnetic distortions and gyroscope bias drift.

- The filter gain β represents all mean zero gyroscope measurement errors, expressed as the magnitude of a quaternion derivative. The sources of error include: sensor noise, signal aliasing, quantisation errors, calibration errors, frequency response characteristics and sensor miss-alignment and axis nonorthogonality.

- The filter gain ζ represents the rate of convergence to remove gyroscope measurement errors which are not mean zero, also expressed as the magnitude of a quaternion derivative.

- An explanatory block diagram of the procedures taken in Madgwick's algorithm can be seen in Figure 47 and all of the pre-mentioned equations are analytically written in his own works: (An efficient orientation filter for inertial and inertial/magnetic sensor arrays, 2010)

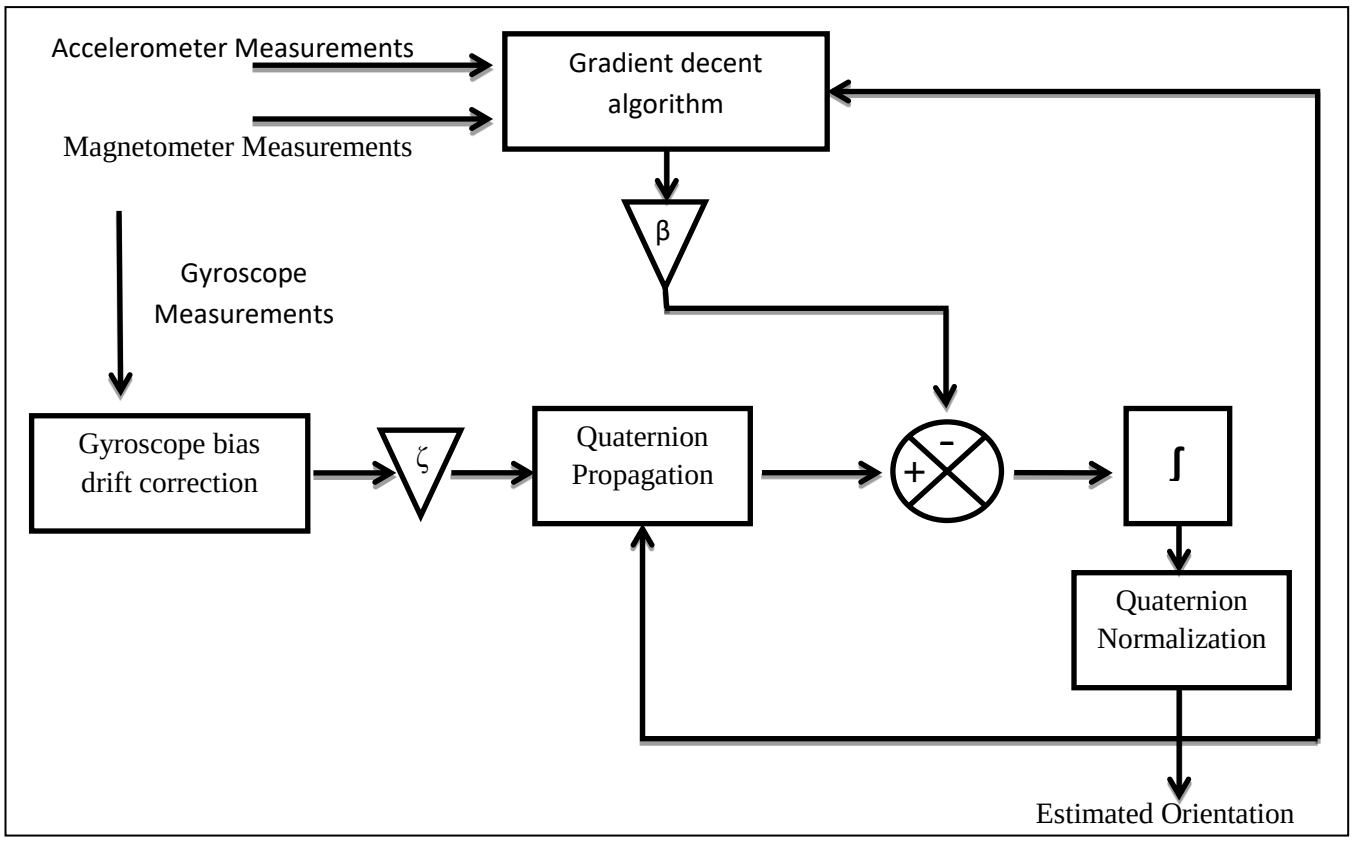


Figure 47: Madgwick's Filter Block Diagram

C.4.2 Mahony Filter

The non-linear complementary filter, proposed by R. Mahony, is another method for bypassing the complex computational requirements of the EKF. This filter was designed for small, low-cost, embedded IMU systems and uses two non-linear observers, and operates as follows:

- A special orthogonal group with its own associated Lie-algebra is assumed and the orientation estimation problem is posed directly to that orthogonal group, as a deterministic observation (Cavallo, et al., 2014).

- Two observers called the direct complementary filter and the passive complementary filter are defined to create, through reformulation, a third formulation called the explicit complementary filter.

- The explicit complementary filter, a reformulation of the passive filter in terms of error measurements, is derived directly through the previous filters, while remaining well defined, even when the data provided is insufficient.

- This observer does not require online algebraic reconstruction of attitude and is ideally suited for implementation on embedded hardware platforms owing to its low complexity (Cirillo, Cirillo, De Maria, Natale, & Pirozzi, 2016).

- The algorithm suffers from possible discontinuities in the bias correction signal, resulting in systematic errors in the reconstructed attitude, when the equivalent rotation angle of the estimated quaternion approaches $\pm\pi$ rad

- A correction step using a Proportional-Integral (PI) compensator is used to correct the measured angular velocity and, exactly as the Madgwik filter, after the quaternion normalization, the quaternion propagation is integrated to obtain an estimate of the orientation.

- The output result depends on well-chose (PI) compensator gains called k_p and k_i.

- The analytic mathematic equations that the non-linear complementary filter was based on can be found on R. Mahony's publication: (Mahony, Hamel, & Pflimlin, 2008)

- An explanatory block diagram of the filter's algorithm can be seen in Figure 48.

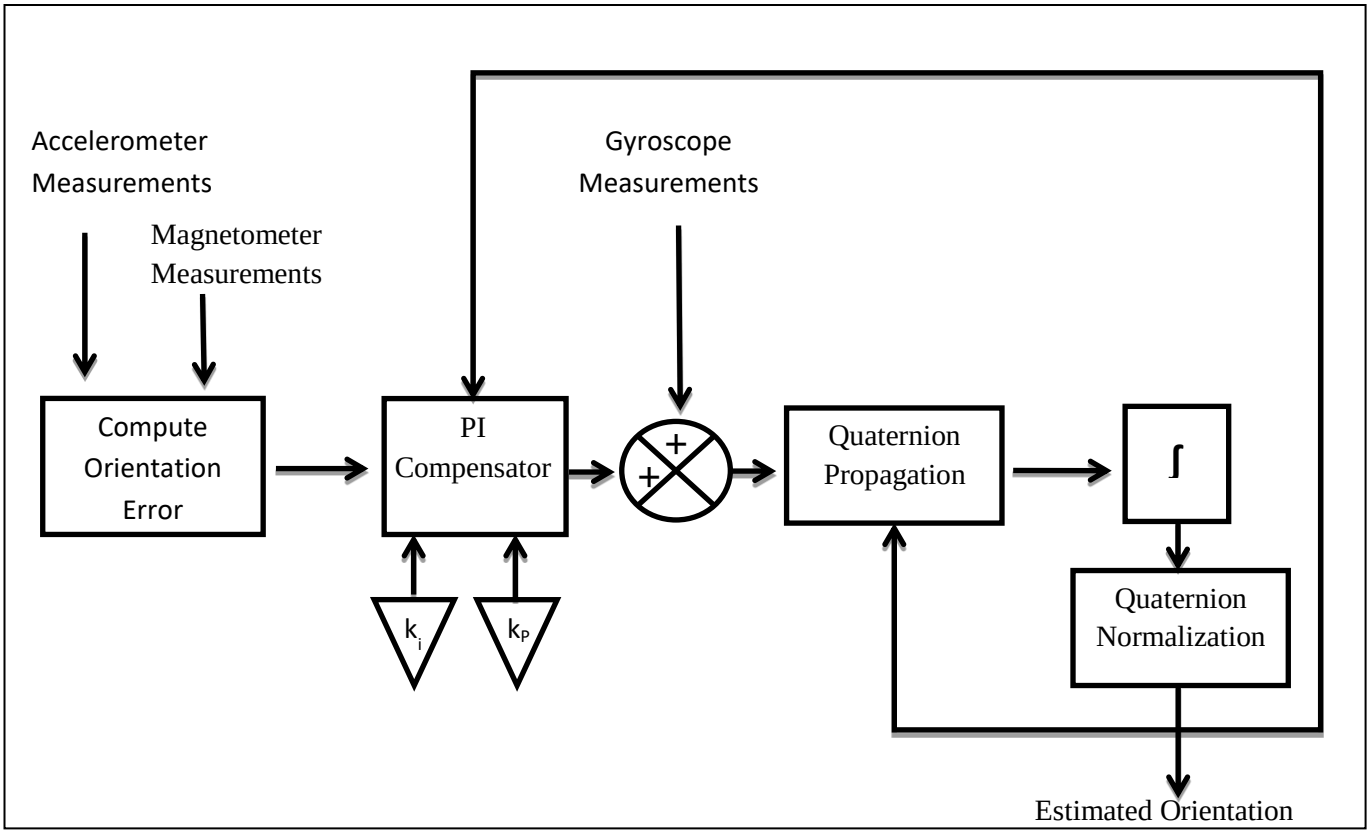


Figure 48: Mahony's Filter Block Diagram

C.4.3 Mahony vs Madgwick

Although both algorithms are widely spread as primary choices for many different applications involving IMUs, there is still a question as to which algorithm is the most fitting for our case. Before mentioning the benefits of each one, it is worth noting that both algorithms were used as better substitutes to the complex, in terms of computation and understanding, EKF. The benefits and drawbacks of each algorithm can be seen in Table 7 as described in experimental test made in (Cavallo, et al., 2014), (Cirillo, Cirillo, De Maria, Natale, & Pirozzi, 2016) & (Jouybari, Ardalan, & Rezvani, 2017).

Table 7: Mahony and Madgwick Filters Comparison

Non-linear Complementary Filter	Madgwick Filter
Specifically designed for integrated low-cost MARGs	**Applicable to both IMUs and MARG sensor arrays**
Smallest computational burden per iteration	**Much smaller computational burden than the EKF**
Partially lower mean of attitude estimation error, in slow single Euler angle rotation tests	**Overall lower standard deviation of attitude estimation error, in slow single Euler angle rotation tests**
Overall lower mean of attitude estimation error, in fast single Euler angle rotation tests	**Partially lower standard deviation of attitude estimation error, in fast single Euler angle rotation tests**
Fine-tuning of k_p and k_i parameters is needed	**Correct choice of β and ζ parameters is needed**

The main question derived from the previous Table is, why the Mahony filter was chosen in place of the Madgwick filter for the new Arduino Nano algorithm. The explanation is based on our system parameters and attempts made to integrate the Madgwick filter into a single Arduino Nano, controlling 9 MPUs. Various methods were used and many computation optimizations were proposed, but using the Madgwick algorithm the Arduino Nano was not able to compute orientations for more than 6 MPUs, fast enough, so that Unity could produce 50-60 fps and a smooth graphic environment for acceptable user immersion.

C.4.4 Magnetometer Calibration Method

Magnetometers, as mentioned previously, are essential for achieving true 9 DoF measurements in any IMU application. The main problem with these instruments is the number of factors including biases, scale factors and non-orthogonality corrections that if not set correctly can create non-ideal response surfaces, due to external magnetic field interference (Real-time attitude-independent three-axis magnetometer calibration).

The main idea behind a magnetometer calibration is adjusting the values of six different factors; three scalars and three biases. When receiving the magnetic data for each axis (usually in mG) m_x, m_y and m_z we can plot them alternately as $M_{xy}(m_x,m_y)$, $M_{xz}(m_x,m_z)$, and $M_{yz}(m_y,m_z)$. We can form these three plots, by recording the readings of several different spatial movements to achieve various 3D space orientations. If the magnetometer is calibrated correctly and each one of the six factors is set, every point in each of these plots should be confined within a circle named C_{xy}, C_{xz}, and C_{yz} respectively, with the extreme points of orientation belonging to the circle and each circle having a center point on the origin point of the plane (O_{xy}, O_{xz}, and O_{yz}) .

Therefore we can assume that every point m_{xy}, m_{xz}, and m_{yz} the functions M_{xy}, M_{xz}, and M_{yz} are defined as:

$$m_{xy} \leq C_{xy},\ m_{xz} \leq C_{xz}, \text{ and } m_{yz} \leq C_{yz}$$

Moreover if the calibration is done correctly these three circles should be approximately the same:

$$C_{xy} \approx C_{xz} \approx C_{yz}$$

Thereby when visualized in 3D space, since they represent 3 different planar shapes, they should form an almost perfect sphere with a center point of O= (0, 0, 0) the origin point of our 3D reference system.

There are two types of interference; hard-iron and soft-iron interference. Hard-iron distortions are produced by materials, usually ferromagnetic, that exhibit a constant, additive field to the earth's magnetic field. Therefore they generate a bias to the output of each magnetometer axis, visually represented with the circles having an offset from the origin point towards the bias axis. Soft-iron distortions on the other hand, are a result of the influence and distortions of several materials that do not necessarily generate a magnetic field and are therefore non-additive. Soft-iron distortions are even dependent on the orientation of the material relative to the sensor. Soft-iron distortions can be visually represented with the circles turning into ellipsoids (Konvalin, 2009).

Hard-iron distortions can be compensated by adding or subtracting up to three "bias constants" for each axis, so as to transverse the points towards forming an origin circle. Soft-iron corrections are difficult to compute in real-time apps and for the purposes of this

project we try to compensate by using three scalar factors to multiply the data so as to re-scale the axial response and make it more spherical.

To achieve manual calibration we can adjust the three scales and three biases with trial and error. For example we can see the measurements of an uncalibrated MPU-9250 in Chart 2.

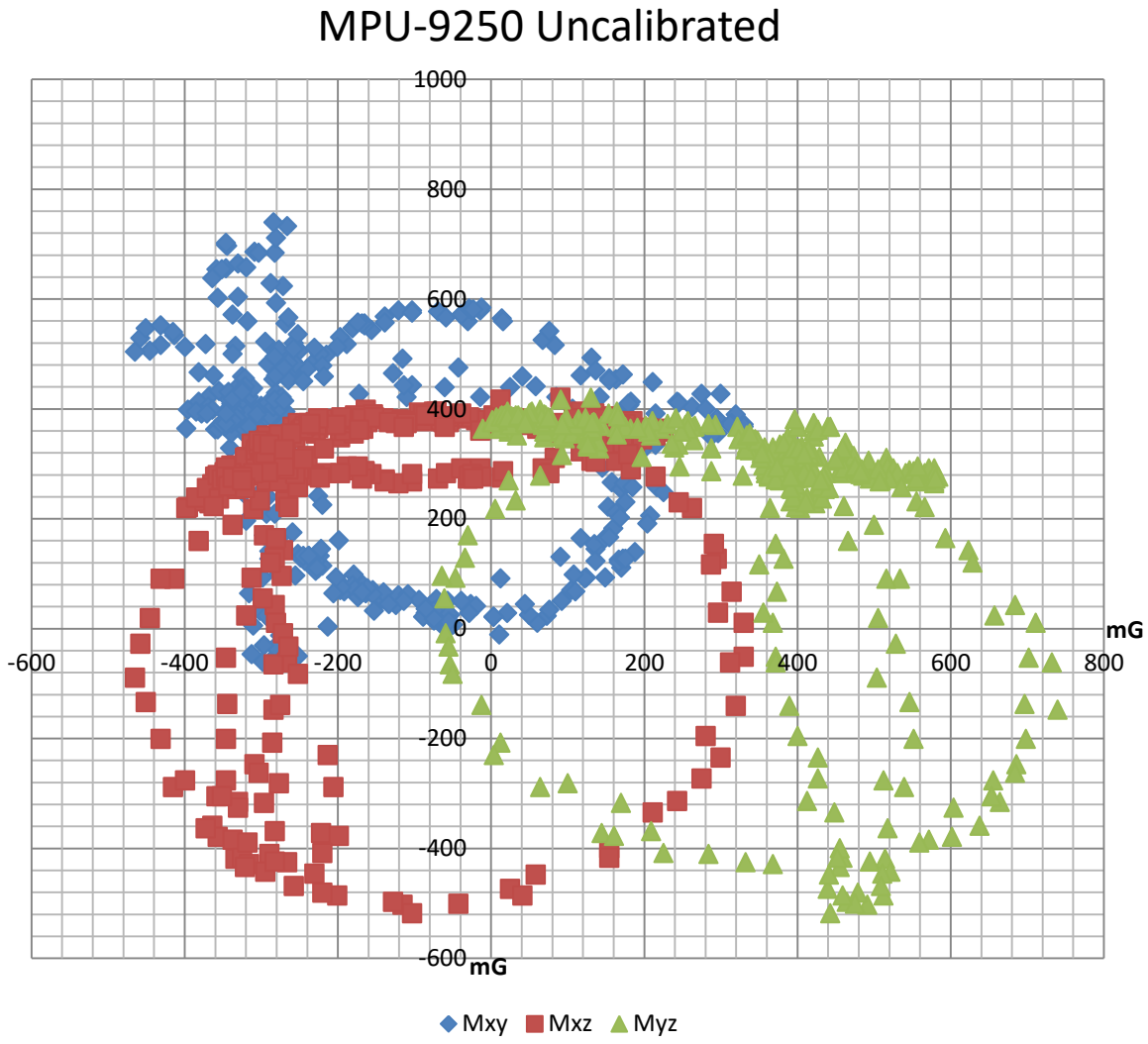


Chart 2: MPU-9250 Uncalibrated

As we can clearly observe the points of each measurement (red, green and blue) are scattered in ellipsoids and do not have the origin point as their center. These ellipsoids can be approximately defined in Chart 3 where we can observe that the ellipsoids are of different size and orientation too.

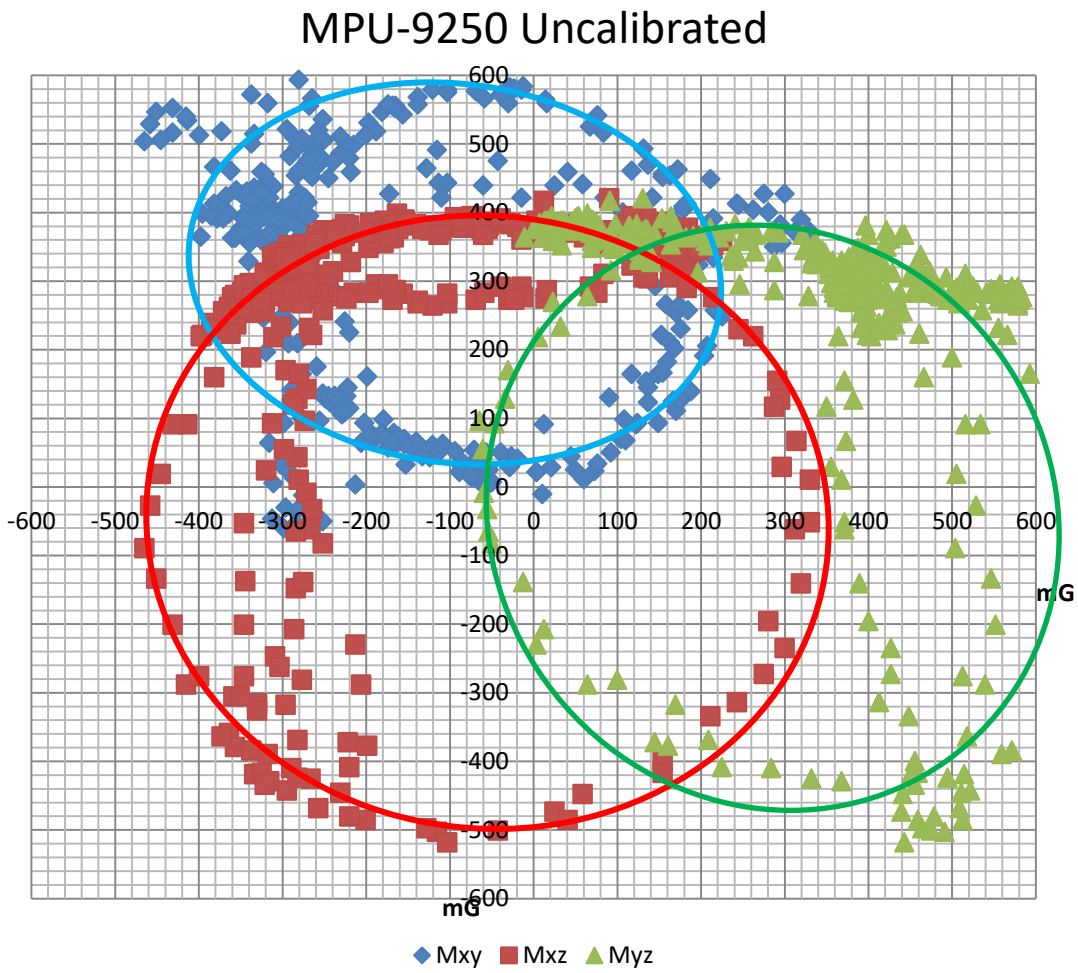


Chart 3: MPU-9250 Uncalibrated Ellipsoid Approximation

Biases are easy to compute by finding the minimum min_{axis} and maximum max_{axis} measurements for each axis and then shifting them accordingly by a bias $magbias_{axis}$ so that:

$$|min_{axis}+magbias_{axis}| = |max_{axis}+magbias_{axis}|, \text{ where } magbias_{axis}$$

After various trials for a manual adjustment, an adequate calibration can be seen in Chart 4 for the uncalibrated MPU-9250 with applied biases:

$$magbias_x = +120mG,\ magbias_y = -375mG,\ magbias_z = +30mG$$

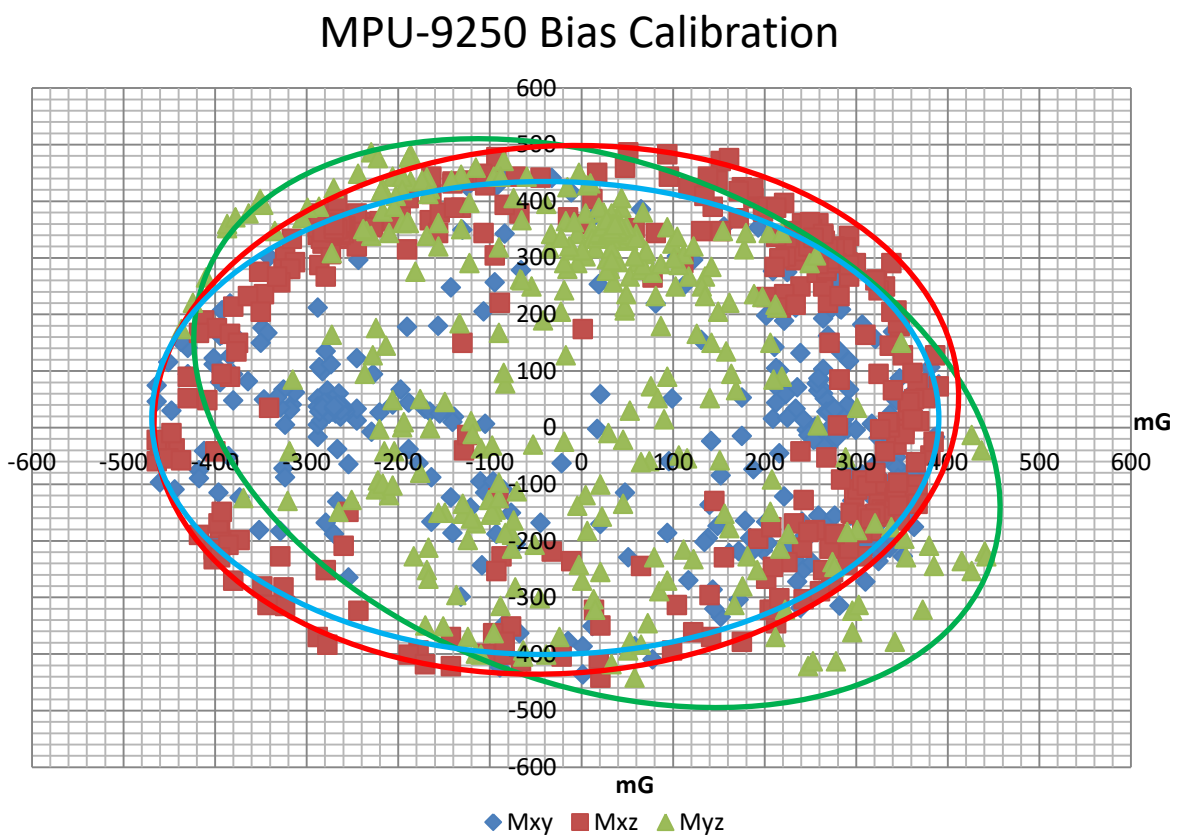


Chart 4: MPU-9250 Bias Calibration

Now that the ellipsoids have been centered we need to adjust them into circles using the scalar parameters: $scale_x$, $scale_y$, $scale_z$.

A scale factor is calculated as the ratio of the average max - min along each axis and the average of all three axes. This means that an axis where the max - min is large has its magnetic field reduced and an axis that under-measures the field with respect to the other axes has its magnetic field values increased. This is just a simple orthogonal rescaling, equivalent to a diagonalized 3 x 3 calibration matrix but it allows some additional correction for scale bias.

The new manual adjustment can be seen in Chart 5 for the uncalibrated MPU-9250 with applied scalars:

$$scale_x = 1.02,\ scale_y = 0.87\ ,\ scale_z = 0.93$$

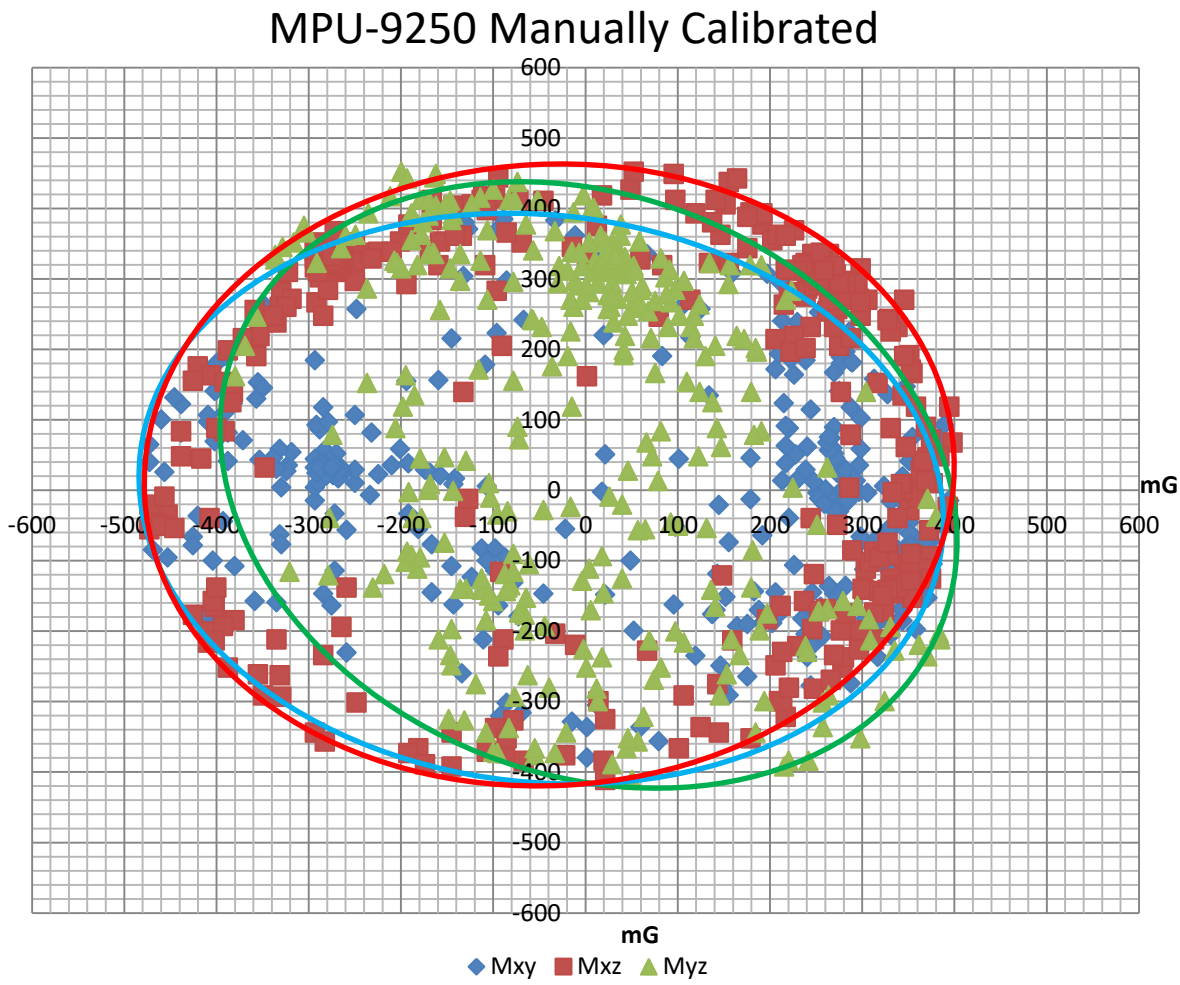
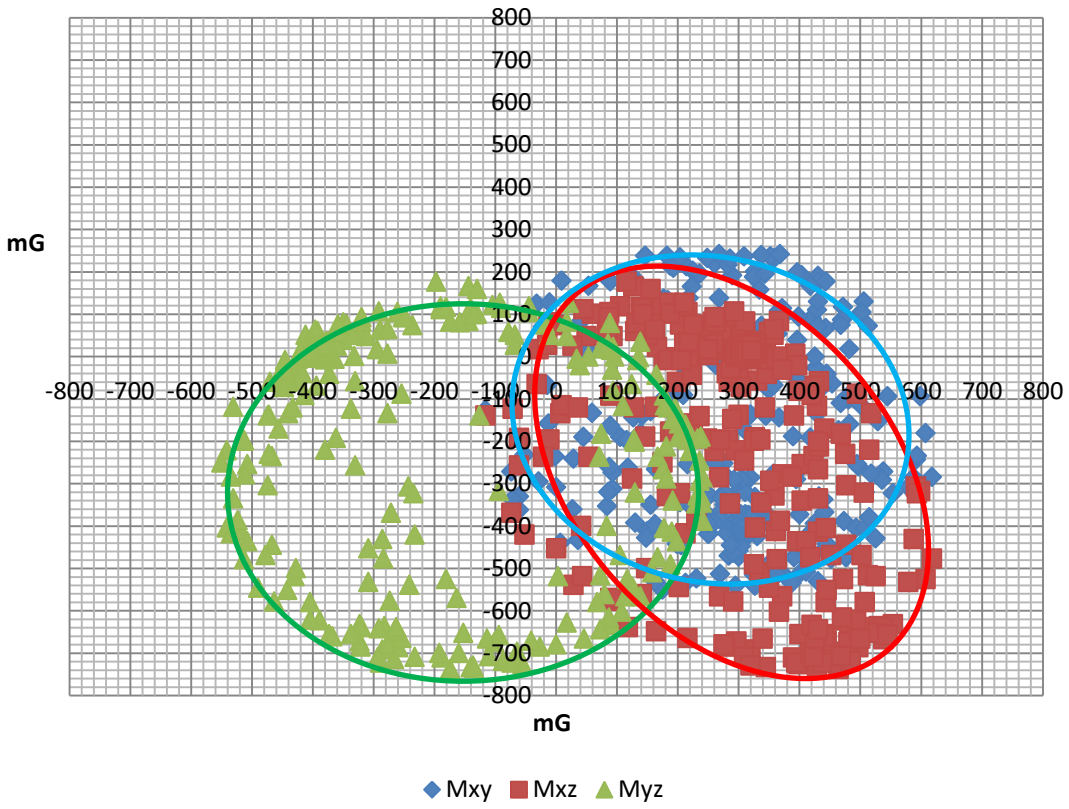


Chart 5: MPU-9250 Manually Calibrated

Because the procedure of the calibration is nowhere near perfect and very time-consuming, as it has to be done for each sensor, each time the experimenting environment changes, there is a function in our code called "*magcalMPU9250*" which can be applied at the start of every measurement session for each sensor to automatically calibrate it. The function has efficient algorithms to acquire enough data within 15 seconds of figure eight spatial waving motions, to calculate the biases and scales for each sensor and provide them on screen. Then the data should be stored in the appropriate matrix positions in the program to proceed with the experiment. It is important to note that the figure eight movements are not random and should create various planar 8-like shapes in each plane of motion (sagittal, coronal transverse). We can see the results of an automatic calibration in Chart 6. It is worth pointing out the difference in the maximum and minimum values of the horizontal and vertical scale between the two diagrams of Chart 6, as well as the difference of the automatically calibrated magnetometer response in Chart 6, with the manually calibrated magnetometer response in Chart 5.

(Winer, 2017)



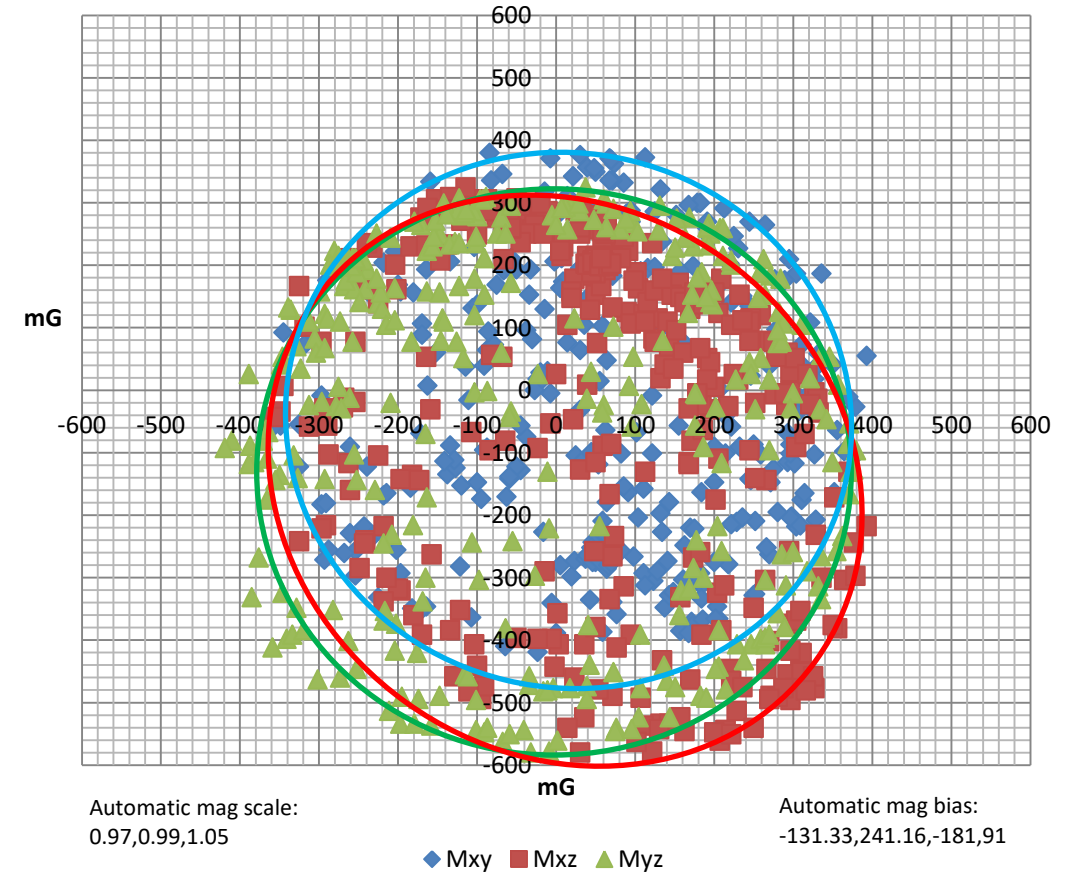


Chart 6: Magnetometer Response of an uncalibrated MPU (up) and an automatically calibrated MPU (down).

C.4.5 Main Algorithm

The main algorithm is a heavily converted version of Kris Winer's sketches (Winer, Arduino sketches for MPU9250 9DoF with AHRS sensor fusion) of a Teensy 3.1 and several other components assisting in the creation a MPU-9250, 9 DoF sensor fusion AHRS. A side-function "Plex" is used, so that the multiplexer can control the flow of data and provide a serial feed from each sensor. In the following sub chapters the commands of the algorithm will be explained step by step.

Main Libraries

The main libraries, created by Kris Winer, are declared. The first library is used to create function that calculates the Mahony or Madgwick results with a single call of the function. The second library is used to automatically set the register map for each sensor and translate the analog readings into data with physical meaning.

```
#include "quaternionFilters.h"
#include "MPU9250.h"
```

Definitions for Debugging and Result Type

We can choose to perform either a single-sensor Serial Debug to ensure that the sensor is working correctly, or a simplified extraction of the AHRS data, if we do not need the Arduino to perform the complex calculations of the filters and their corrections. Moreover we setup the pins, so as to expect certain types of data from each connection. Lastly we choose the library with the appropriate functions to establish serial communication and we set the communications speed and address.

```
#define AHRS    true        // Set to false for basic data read.
#define SerialDebug   false  // Set to true to get Serial output for debugging.

// Pin definitions.
int intPin = 12;  // These can be changed, 2 and 3 are the Arduinos ext int pins.
int myLed  = 13;  // Set up pin 13 led for toggling.
```

```
// Multiplexer Pin definitions.

#define s0 8  // bit 7 of PORTB Multiplexer's s0.
#define s1 9  // bit 6 of PORTB Multiplexer's s1.
#define s2 10 // bit 5 of PORTB Multiplexer's s2.
#define s3 11 // bit 4 of PORTB Multiplexer's s3.

// Counters and Button ON/OFF variables.
int j = 1; // Counter for the multiplexer.
int buttonState;

// I2C Communication Presets and choice of Library (Wire.h in our case).

#define I2Cclock 400000
#define I2Cport Wire

// Use either the next line or one after that to select which I2C address your device is using.
#define MPU9250_ADDRESS MPU9250_ADDRESS_AD0
//#define MPU9250_ADDRESS MPU9250_ADDRESS_AD1

MPU9250 myIMU(MPU9250_ADDRESS, I2Cport, I2Cclock);
```

Preset Values to Bypass Calibration

In this part we can preset certain values to bypass the time-consuming part of waving each sensor for 15 seconds to achieve a good calibration. As it will be described in a forthcoming chapter, certain measurements have to be done repeatedly so that a correct preset for the magnetometer scale and bias values is chosen. The other presets (Factory Magnetometer Calibrations, Accelerometer Resolution, Gyroscope Resolution and Magnetometer Resolution) are unique for each sensor and do not change through an alteration of testing environment or though time. Therefore, they are already set in the code for our 9 MPUs.

```
float magBiasx [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float magBiasy [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float magBiasz [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float magScalex [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float magScaley [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float magScalez [] = {0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0};

float factoryMagCalibrationx [] = {1.19 , 1.18 , 1.19 , 1.2 , 1.2 , 1.18 , 1.19 , 1.2 , 1.18};

float factoryMagCalibrationy [] = {1.2 , 1.19 , 1.19 , 1.2 , 1.2 , 1.19 , 1.19 , 1.21 , 1.19};

float factoryMagCalibrationz [] = {1.15 , 1.14 , 1.15 , 1.16 , 1.16 , 1.14 , 1.15 , 1.16 , 1.4};

float aRes = 0.00;

float gRes = 0.01;

float mRes = 1.50;
```

Parameter Setup

In this part the communication baud-rate is chosen. The “WHO AM I” register is checked so that it is assured that the connections are done correctly and the reset pin and led of the Arduino are configured. There is also a loop to ensure that serial communication is maintained.

```
void setup()

{

  Wire.begin();

  // TWBR = 12;

 // 400 kbit/sec I2C speed.
```

```
Serial.begin(38400);  // We initialize serial communication with a baud-rate of 38400.
while(!Serial){};  // This loop is for ensuring that we maintain serial communication.


// Set up the interrupt pin, its set as active high, push-pull.
pinMode(intPin, INPUT);
digitalWrite(intPin, LOW);
pinMode(myLed, OUTPUT);
digitalWrite(myLed, HIGH);
// Set up the control pins for the multiplexer as outputs.
pinMode(s0,   OUTPUT);
pinMode(s1,   OUTPUT);
pinMode(s2,   OUTPUT);
pinMode(s3,   OUTPUT);
```

MPU Initialization Functions

It should be noted that the rest of the setup and calibration does not proceed if the registers are not correct, assuming that something is connected or configured correctly. In this section there is enough code to proceed with single MPU initialization, without knowing the presets. Since we have pre-calibrated our sensors, though, this part is commented out (greyed out) and not used during our tests.

```
for (int l=1; l <= 9; l++) {
  Plex(l); // Multiplexer function, analyzed at the end of the algorithm.
  // Read the WHO_AM_I register, this is a good test of communication.
  byte c = myIMU.readByte(MPU9250_ADDRESS, WHO_AM_I_MPU9250);
  Serial.print(F("MPU9250 I AM 0x"));
  Serial.print(c, HEX);
  Serial.print(F(" I should be 0x"));
```

```
Serial.println(0x71, HEX); // Or (0x73, HEX) depending on the manufacturer of the MPU.

Serial.print ("MPU # ");
Serial.print (j);
Serial.println (" ");


if ((c == 0x71)|| (c == 0x73)) // WHO_AM_I should always be 0x71 or 0x73
{
 Serial.println(F("MPU9250 is online..."));

 // For a single MPU with no presets start by performing self-test and reporting values.
 /*myIMU.MPU9250SelfTest(myIMU.selfTest);
 Serial.print(F("x-axis self test: acceleration trim within : "));
 Serial.print(myIMU.selfTest[0],1); Serial.println("% of factory value");
 Serial.print(F("y-axis self test: acceleration trim within : "));
 Serial.print(myIMU.selfTest[1],1); Serial.println("% of factory value");
 Serial.print(F("z-axis self test: acceleration trim within : "));
 Serial.print(myIMU.selfTest[2],1); Serial.println("% of factory value");
 Serial.print(F("x-axis self test: gyration trim within : "));
 Serial.print(myIMU.selfTest[3],1); Serial.println("% of factory value");
 Serial.print(F("y-axis self test: gyration trim within : "));
 Serial.print(myIMU.selfTest[4],1); Serial.println("% of factory value");
 Serial.print(F("z-axis self test: gyration trim within : "));
 Serial.print(myIMU.selfTest[5],1); Serial.println("% of factory value");
 */
 // Calibrate gyro and accelerometers, load biases in bias registers for all MPUs.
 myIMU.calibrateMPU9250(myIMU.gyroBias, myIMU.accelBias);
```

```
myIMU.initMPU9250();
// Initialize each device for active mode read of acclerometer, gyroscope, and temperature.
Serial.print("MPU9250 #");
Serial.print(j);
Serial.println ("initialized for active data mode....");

// Read the WHO_AM_I register of the magnetometer, this is a good test of communication.
byte d = myIMU.readByte(AK8963_ADDRESS, WHO_AM_I_AK8963);
Serial.print("AK8963 ");
Serial.print("I AM 0x");
Serial.print(d, HEX);
Serial.print(" I should be 0x");
Serial.println(0x48, HEX);

// If the WHO_AM_I register of the magnetometer is not correct the interrupt the program.

if (d != 0x48)
{
  // Communication failed, stop here.
  Serial.println(F("Communication failed, abort!"));
  Serial.flush();
  abort();
}

// Get single MPU non-preset magnetometer calibration from AK8963 ROM.
// We use our own presets, but this function is needed to initialize the magnetometers.
```

```
myIMU.initAK8963(myIMU.factoryMagCalibration);
// Initialize devices for active mode read of magnetometer.
Serial.println("AK8963 initialized for active data mode....");
```

MPU Calibration Functions

Same as before, this part of the code focuses on single MPU calibration functions, which are greyed-out because they are preset. Moreover the main function for calibrating the MPU with spatial figure-8 movements for 15 seconds is called. “myIMU.magCalMPU9250” is the function that delays the experiments the most, asking for a total of 19-24 seconds of calibration and is the main reason the presets were created.

```
// This part is used to inform us about single MPU non-preset magnetometer calibration.
/*
if (SerialDebug)
{
  //  Serial.println("Calibration values: ");
  Serial.print("X-Axis factory sensitivity adjustment value ");
  Serial.println(myIMU.factoryMagCalibration[0], 2);
  Serial.print("Y-Axis factory sensitivity adjustment value ");
  Serial.println(myIMU.factoryMagCalibration[1], 2);
  Serial.print("Z-Axis factory sensitivity adjustment value ");
  Serial.println(myIMU.factoryMagCalibration[2], 2);
}
*/
// This part is used to get sensor resolutions. We only need to do this once.
/*
myIMU.getAres();
myIMU.getGres();
myIMU.getMres();
```

```
Serial.println (myIMU.aRes);
Serial.println (myIMU.gRes);
Serial.println (myIMU.mRes);
*/

// The next call delays for 4 seconds, and then records about 15 seconds of
// data to calculate bias and scale.
// myIMU.magCalMPU9250(myIMU.magBias, myIMU.magScale);

// We could also set presets for single MPU use with the following functions:
/*
myIMU.magBias[0]= ;
myIMU.magBias[1]=  ;
myIMU.magBias[2]=  ;
Serial.println("AK8963 mag biases (mG)");
Serial.println(myIMU.magBias[0]);
Serial.println(myIMU.magBias[1]);
Serial.println(myIMU.magBias[2]);
myIMU.magScale[0]= ;
myIMU.magScale[1]= ;
myIMU.magScale[2]= ;
Serial.println("AK8963 mag scale (mG)");
Serial.println(myIMU.magScale[0]);
Serial.println(myIMU.magScale[1]);
Serial.println(myIMU.magScale[2]);

if(SerialDebug)
{
```

```
    Serial.println("Magnetometer:");
    Serial.print("X-Axis sensitivity adjustment value ");
    Serial.println(myIMU.factoryMagCalibration[0], 2);
    Serial.print("Y-Axis sensitivity adjustment value ");
    Serial.println(myIMU.factoryMagCalibration[1], 2);
    Serial.print("Z-Axis sensitivity adjustment value ");
    Serial.println(myIMU.factoryMagCalibration[2], 2);
  }
  */
   delay(2000); // Add delay to see results before serial spew of data
 } // if (c == 0x71) concludes here.
```

Wrong Address Response

If the register of "WHO AM I" is not the one that was expected, the communication is terminated for prevention of displaying false results.

```
 else
 {
   Serial.print("Could not connect to MPU9250: 0x");
   Serial.println(c, HEX);

   // Communication failed, stop here.
   Serial.println(F("Communication failed, abort!"));
   Serial.flush();
   abort();
 }
}
```

Main Loop

```
void loop()
{
  // If intPin goes high, all data registers have new data.
  // On interrupt, check if data ready interrupt.
  if (myIMU.readByte(MPU9250_ADDRESS, INT_STATUS) & 0x01)
  {

    myIMU.readAccelData(myIMU.accelCount);  // Read the x/y/z adc values.

    // Now we'll calculate the accleration value into actual g's.
    // This depends on scale being set.
    myIMU.ax = (float)myIMU.accelCount[0] * aRes[j-1];
// myIMU.aRes - myIMU.accelBias[0]; For single MPU calculations.
    myIMU.ay = (float)myIMU.accelCount[1] * aRes[j-1];
// myIMU.aRes - myIMU.accelBias[1]; For single MPU calculations.
    myIMU.az = (float)myIMU.accelCount[2] * aRes[j-1];
 // myIMU.aRes - myIMU.accelBias[2]; For single MPU calculations.

    myIMU.readGyroData(myIMU.gyroCount);  // Read the x/y/z adc values.

    // Calculate the gyro value into actual degrees per second
    // This depends on scale being set
    myIMU.gx = (float)myIMU.gyroCount[0] * gRes[j-1];
// myIMU.gRes; For single MPU calculations.
    myIMU.gy = (float)myIMU.gyroCount[1] * gRes[j-1];
// myIMU.gRes; For single MPU calculations.
```

```
    myIMU.gz = (float)myIMU.gyroCount[2] * gRes[j-1];
// myIMU.gRes; For single MPU calculations.

    myIMU.readMagData(myIMU.magCount);  // Read the x/y/z adc values

    // Calculate the magnetometer values in milliGauss
    // Include factory calibration per data sheet and user environmental
    // corrections
    // Get actual magnetometer value, this depends on scale being set
myIMU.mx = (float)myIMU.magCount[0] * mRes[j-1]
* factoryMagCalibrationx[j-1] - magBiasx[j-1];
// myIMU.mRes * myIMU.factoryMagCalibration[0] - myIMU.magBias[0]
// For single MPU calculations.
    myIMU.my = (float)myIMU.magCount[1] * mRes[j-1]
            * factoryMagCalibrationy[j-1] - magBiasy[j-1];
// myIMU.mRes * myIMU.factoryMagCalibration[1] - myIMU.magBias[1];
// For single MPU calculations.
    myIMU.mz = (float)myIMU.magCount[2] * mRes[j-1]
            * factoryMagCalibrationz[j-1] - magBiasz[j-1];
// myIMU.mRes * myIMU.factoryMagCalibration[2] - myIMU.magBias[2];
// For single MPU calculations.

  } // if (readByte(MPU9250_ADDRESS, INT_STATUS) & 0x01) ends here.
  // Must be called before updating quaternions!
  myIMU.updateTime();
```

Choosing Filter

We can choose which filter to use, as there is code written for both Mahony and Madgwick filter. It is very important to feed the data to each function with the global reference of NED (North, East, Down). This means that each axis of each module should be sent with the correct order and sign. For example, we chose to send the axis of each data in the Mahony filter in the following order: ax, ay, -az, gx, gy, gz, my, mx, mz. This is done to achieve NED orientation and because the magnetomer's x axis is parallel to the accelerometers y axis.

```
// MadgwickQuaternionUpdate(-myIMU.ax, -myIMU.ay, myIMU.az, myIMU.gx *
// DEG_TO_RAD, myIMU.gy * DEG_TO_RAD, -myIMU.gz * DEG_TO_RAD,
// myIMU.my, myIMU.mx, myIMU.mz, myIMU.deltat);
```

```
MahonyQuaternionUpdate(myIMU.ax, myIMU.ay, -myIMU.az, myIMU.gx*PI/180.0f,
myIMU.gy*PI/180.0f, myIMU.gz*PI/180.0f,  myIMU.my,  myIMU.mx, myIMU.mz,
myIMU.deltat);
```

Result Display Method

As mentioned previously, in the "Definitions for Debugging and Result Type" part, we can choose the type of data that will be displayed to us by changing the "AHRS" parameter. If the "AHRS" parameter is set to *false*, then we will receive messages regarding calculated values for each module, with their individual measurement sizes. On the other hand, if we set the parameter to *true*, we will receive the Quaternion iteration depending on the filter that we chose previously. We can also receive the inner temperature measurements from the respective MPU module, if there is one.

```
if (!AHRS)
{
  myIMU.delt_t = micros() - myIMU.count;
  if (myIMU.delt_t > 500)
  {
    if(SerialDebug)
```

```
{
 // Print acceleration values in mg!
 Serial.print("X-acceleration: "); Serial.print(1000 * myIMU.ax);
 Serial.print(" mg ");
 Serial.print("Y-acceleration: "); Serial.print(1000 * myIMU.ay);
 Serial.print(" mg ");
 Serial.print("Z-acceleration: "); Serial.print(1000 * myIMU.az);
 Serial.println(" mg ");

 // Print gyro values in deg/sec
 Serial.print("X-gyro rate: "); Serial.print(myIMU.gx, 3);
 Serial.print(" degrees/sec ");
 Serial.print("Y-gyro rate: "); Serial.print(myIMU.gy, 3);
 Serial.print(" degrees/sec ");
 Serial.print("Z-gyro rate: "); Serial.print(myIMU.gz, 3);
 Serial.println(" degrees/sec");

 // Print mag values in deg/sec
 Serial.print("X-mag field: "); Serial.print(myIMU.mx);
 Serial.print(" mG ");
 Serial.print("Y-mag field: "); Serial.print(myIMU.my);
 Serial.print(" mG ");
 Serial.print("Z-mag field: "); Serial.print(myIMU.mz);
 Serial.println(" mG");

 myIMU.tempCount = myIMU.readTempData();
 // Read the adc values Temperature in degrees Centigrade
 myIMU.temperature = ((float) myIMU.tempCount) / 333.87 + 21.0;
```

```
      // Print temperature in degrees Centigrade
      Serial.print("Temperature is ");  Serial.print(myIMU.temperature, 1);
      Serial.println(" degrees C");
    }
    myIMU.count = micros();
    digitalWrite(myLed, !digitalRead(myLed));  // Toggle led.
  } // if (myIMU.delt_t > 500) ends here.
} // if (!AHRS) ends here.

else
{
  myIMU.delt_t = micros() - myIMU.count;
  if (myIMU.delt_t > 500)
  {
    if(SerialDebug)
    {
      // We export data to unity as follows: (MPU#, q0, q1, q2, q3, Button On/Off).
      Serial.print(j);
      Serial.print(",");
      Serial.print(*getQ());
      Serial.print(",");
      Serial.print(*(getQ() + 1));
      Serial.print(",");
      Serial.print(*(getQ() + 2));
      Serial.print(",");
      Serial.print(*(getQ() + 3));
      Serial.print(",");
      Serial.println(buttonState);
```

```
      j++; // Move to the next MPU.
      // Renew the multiplexer when it finishes with the 9th MPU.
      if (j==10)
      {
       j=1;
      }
     Plex(j); // Configure the Multiplexer for the next MPU.
}
     myIMU.count = micros();
     myIMU.sumCount = 0;
     myIMU.sum = 0;
   } // if (myIMU.delt_t > 500) ends here.
  } // if (AHRS) ends here.
}// Main loop ends here.
```

Multiplexer Function

This function is used to automatically configure the pin of the multiplexer that should be active at the given time.

```
void Plex(int port)
{
 if (port==1)
 {
   digitalWrite(s0, LOW);
   digitalWrite(s1, LOW);
   digitalWrite(s2, LOW);
   digitalWrite(s3, LOW);
 }
 else if (port==2)
```

```
{
  digitalWrite(s0, HIGH);
  digitalWrite(s1, LOW);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
}
else if (port==3)
{
  digitalWrite(s0, LOW);
  digitalWrite(s1, HIGH);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
}
else if (port==4)
{
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
}
else if (port==5)
{
  digitalWrite(s0, LOW);
  digitalWrite(s1, LOW);
  digitalWrite(s2, HIGH);
  digitalWrite(s3, LOW);
}
else if (port==6)
{
  digitalWrite(s0, HIGH);
  digitalWrite(s1, LOW);
```

```
  digitalWrite(s2, HIGH);
  digitalWrite(s3, LOW);
 }
 else if (port==7)
 {
  digitalWrite(s0, LOW);
  digitalWrite(s1, HIGH);
  digitalWrite(s2, HIGH);
  digitalWrite(s3, LOW);
 }
 else if (port==8)
 {
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
  digitalWrite(s2, HIGH);
  digitalWrite(s3, LOW);
 }
 else if (port==9)
 {
  digitalWrite(s0, LOW);
  digitalWrite(s1, LOW);
  digitalWrite(s2, LOW);
  digitalWrite(s3, HIGH);
 }
 }// Plex ends here.
// Program ends here.
```

C.5 Individual Components

C.5.1 MPU-9250

The MPU-9250 is a multi-chip module (MCM) consisting of two dies integrated into a single QFN package. One die houses the 3-Axis gyroscope and the 3-Axis accelerometer and the other die houses the AK8963 3-Axis magnetometer from Asahi Kasei Microdevices Corporation™. Hence, the MPU-9250 is a 9-axis motion tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer and a Digital Motion Processor™ (DMP) all in a small and affordable package. The MPU-9250 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I2C port.

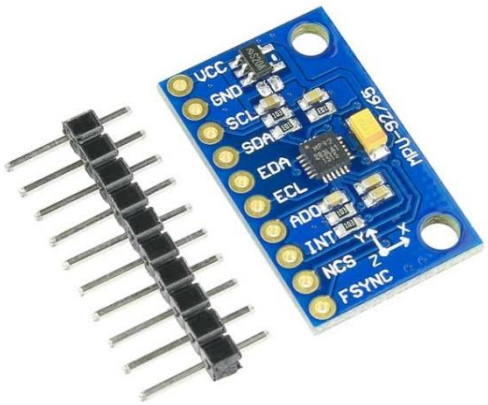


Figure 49: The MPU-9250/65[24]

Moreover the MPU-9250 features 9-axis integration on-chip Motion Fusion™ and run time calibration firmware with three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs, three for digitizing the accelerometer outputs, and three for digitizing the magnetometer outputs. Communication with all registers of the device is performed using either I2C at 400 kHz or SPI at 1 MHz. For applications requiring faster communication, the sensor and interrupt registers may be read using SPI at 20MHz.

Its magnetometer is a typical 3-axis monolithic Hall-effect magnetic sensor with magnetic concentrator. Moreover its MEMS structure is hermetically sealed and bonded at wafer level. Its most important features on all three parts (accelerometer, gyroscope and magnetometer) and its general capabilities are listed in Table 8 below.

[24] Retrieved 1/21/2019 from: https://www.addicore.com/mpu-9250-p/ad280.htm

Table 8: MPU-9250 Main Characteristics

Operating Voltage (V_{DD})	2.4 V - 3.6 V
Temperature Range	-40^0 C to 85^0 C
Operating Current	3.5 mA (when all 9 motion sensing axes and the DMP are enabled)
Size	3 mm x 3 mm x 1 mm
I^2C communication speed	100 kHz on Standard Mode 400 kHz on Fast Mode
Shock tolerance	10000 g
SPI serial interface for communicating with all registers	1 MHz
SPI serial interface for reading sensor and interrupt registers	20MHz

Table 9: MPU-9250 Individual Module Characteristics

Feature	Gyroscope	Accelerometer	Magnetometer
Full-scale Range	±250, ±500, ±1000, and ±2000°/sec	±2g, ±4g, ±8g and ±16g	±4800μT
Normal operating current	3.2mA	450μA	280 μA at 8Hz
Self-test	Yes	Yes	Yes

(MPU-9250 Product Specification Revision 1.0, 2014)

We are using an OEM MPU-9265 version and not the original InvenSense MPU-9250 breakout-board, so it should be mentioned that the figures of Table were not considered to be exact and during design and prototyping, were handled with various safety factors.

C.5.2 Arduino Nano™

The Arduino Nano is an open-source, compact, bread-board friendly processing unit board, that uses the ATmega328 (Arduino Nano 3.x) processor. It was chosen for its small size, low cost, easy programming and overall adequate technical specifications for the task at hand. It is designed to control and communicate with up to 8 analog and 22 digital pins, 6 of which are square wave (PWM) generating pins. Moreover some of the digital pins are designed to serve as specially purposed communication diodes, such as a clock related gateway, an integrated led controlling pin and serial communication (TX/RX) gateways.

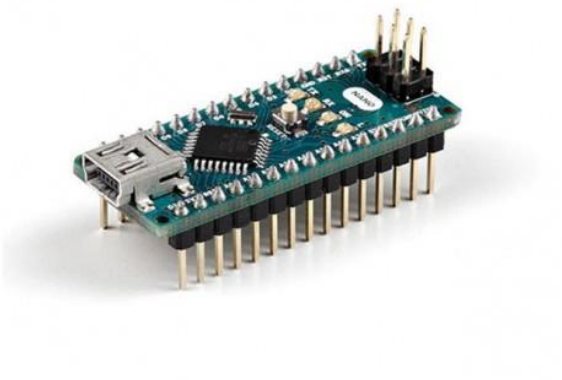


Figure 50: The Arduino Nano[25]

Its technical specifications and mechanical design are seen on Table 10 and Figure 51 respectively.

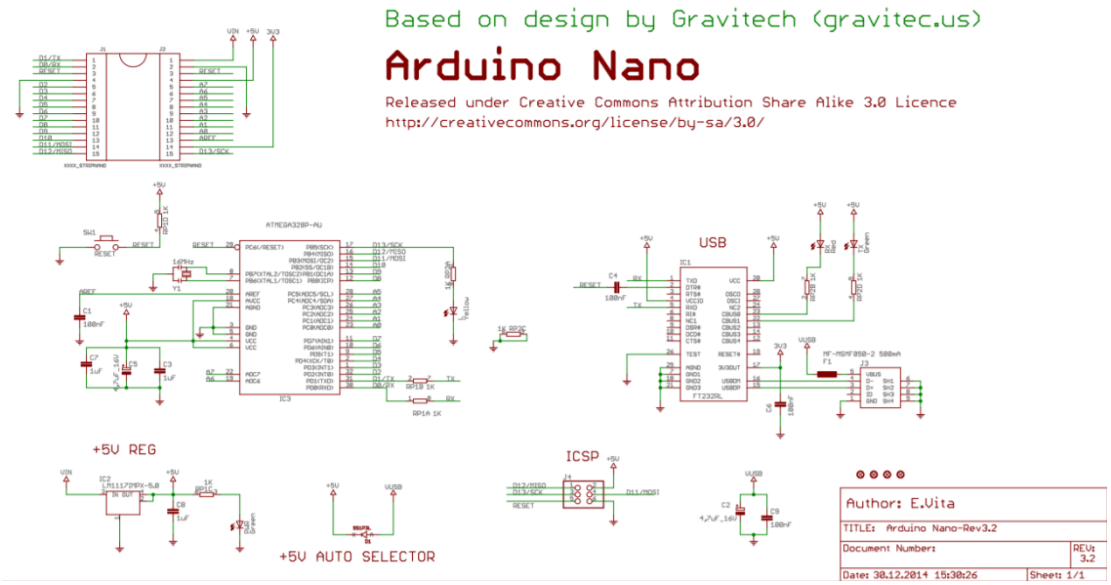


Figure 51: Arduino Nano-Rev3.2 Schematics **(Vita, 2014)**

[25] Retrieved 1/20/2019 from: https://store.arduino.cc/arduino-nano

Table 10: Arduino Nano Basic Characteristics

Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12V
Power Consumption	19 mA
PCB Size	18mm x 45mm
Weight	7 g

(Arduino Nano Tech Specs)

C.5.3 HC-05 Bluetooth Module and CSR 4.0 USB Receiver

The HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for wireless serial communication, providing an easy communication interface with controller or PC at a small size and low cost.

The device is used to achieve wireless communication between a PC and the Arduino Nano. An extra USB dongle device is needed if the PC does not have a Bluetooth receiver already installed. Its specifications can be seen in Table 11.

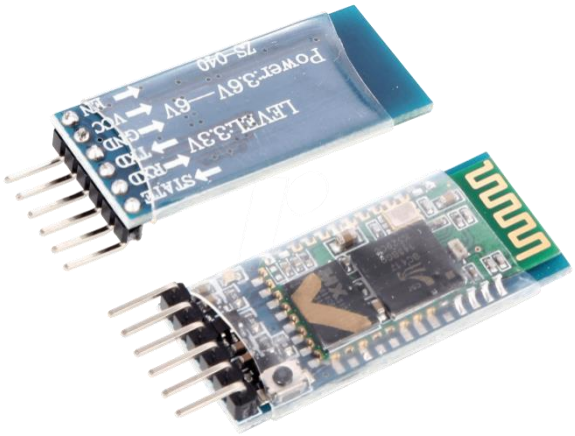


Figure 52: The HC-05[26]

Table 11: HC-05 Product Specifications

Operating Voltage	4V-6V (Typically +5V)
Operating Current	30mA
Range	<100m
Serial communication	USART and TTL compatible
Transmission method	Frequency-Hopping Spread spectrum (FHSS)
Operation Type	Master, Slave or Master/Slave mode
Supported baud rates	9600/19200/38400/57600/115200/230400/460800

(HC-05 Bluetooth Module User's Manual V1.0)

The CSR 4.0 or BlueCore CSR8510 A10 WLSCP is a single-chip radio and baseband IC for Bluetooth 2.4 GHz systems including EDR to 3 Mbps. It is capable of maintaining up to 3 dual-mode Bluetooth low energy connections and has all the basic specifications (Table 12) needed to cooperate with the HC-05 Bluetooth module. It has an integrated Stack Software, CSR's Bluetooth Protocol Stack, which provides master/slave operation, encryption, a Security Manger firmware and other useful protocol software.

[26] Retrieved 1/19/2019 from: http://rosetaelectronics.com/shop/bluetooth-hc-05/

Figure 53: The CSR V4.0[27]

(Bluetooth v4.0 Specification CSR8510A10)

Table 12: CSR V4.0 Technical Characteristics

RF Transmit Power	9.5 dBm
Dynamic Range of 6-bit DAC	>30 db
Receiver Sensitivity	-91 dBm (basic rate), -86dBm@.01%BER
USB 2.0 Interface Full-speed	12 Mbps
Integrated balun	50Ω impedance in TX and RX modes
Frequency Band	2.402G-2.480G
Receiving/sending range	20 m
Supply Voltage	5V DC(MAX5.75V)
Working Current	22mA with typical profiles
Compliance	V4.0 Class2, Bluetooth 2.1+EDR
OS supported	Windows '98,2000, 7, Vista, XP
Temperature	-20 ^{O}C to +70 ^{O}C

D. Validation of the Current Setup

D.1 Presets

As described in the Main Algorithm Chapter C.4.5, to accurately calibrate each MPU we need to execute a range of planar and spatial figure 8 motions for a period of 15 seconds. This procedure is time-consuming and can increase the testing time for each subject massively. Therefore, a calibration matrix has been stored in the Algorithm. This matrix serves the role of a preset of bias and scale measurements done for each sensor in NTUA Lab of Cognitive Ergonomics. The values represent the averages of each calibration variable, calculated through a series of 10 correct calibrations that were executed in the Lab. In Table 13 we can see that the results for MPU differ, in all variables, and should be incorporated individually in the Mahony filter. It is worth noting that after a certain period of time, where

[27] Retrived 1/19/2018 from: https://www.kuongshun-ks.com/raspberry-pi/raspberry-pi-3-2-model-b-accessories/bluetooth-4-0-usb-dongle-adapter-csr-4-0.html

the Earth's magnetic field will have changed significantly, these values should be recalculated.

Table 13: Calibration Preset Measurements for each of the 9 MPUs

MPU#1	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	309.04	321.55	319.76	314.4	316.19	325.12	317.98	316.19	330.48	303.68	317.439
Mag Bias y	698.97	677.47	668.51	686.43	695.39	681.05	725.86	697.18	668.51	718.69	691.806
Mag Bias z	39.6	74.04	98.15	65.43	37.88	79.21	-5.17	29.27	72.32	60.27	55.1
Mag Scale x	0.89	0.83	0.82	0.88	0.9	0.85	0.96	0.88	0.88	0.92	0.881
Mag Scale y	1.01	0.98	0.97	0.98	0.97	1	0.98	1	0.93	0.91	0.973
Mag Scale z	1.12	1.28	1.34	1.19	1.17	1.23	1.06	1.15	1.27	1.24	1.205
MPU#2	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	136.65	141.97	126	131.33	134.87	126	136.65	120.68	111.8	117.13	128.308
Mag Bias y	543.06	518.05	539.49	550.21	519.84	509.12	541.27	555.56	551.99	553.78	538.237
Mag Bias z	-49.77	-37.75	-20.59	-27.46	17.16	10.3	-41.19	-51.48	-42.9	-73.79	-31.747
Mag Scale x	0.92	0.87	0.91	0.89	0.9	0.83	0.95	0.93	0.94	0.92	0.906
Mag Scale y	1.01	1.06	0.94	0.96	0.99	0.99	0.98	1.04	0.98	1.03	0.998
Mag Scale z	1.09	1.1	1.19	1.19	1.14	1.27	1.08	1.05	1.09	1.07	1.127
MPU#3	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	41.09	62.52	41.09	51.81	44.66	35.73	44.66	53.59	33.94	21.44	43.053
Mag Bias y	702.05	735.99	744.92	677.04	723.48	694.9	753.85	689.54	700.26	741.35	716.338
Mag Bias z	-122.67	-136.5	-158.96	-126.13	-176.24	-153.78	-136.5	-98.49	-126.13	-195.24	-143.064
Mag Scale x	0.85	0.86	0.9	0.81	0.91	0.88	0.88	0.82	0.87	0.95	0.873
Mag Scale y	0.98	0.94	0.96	1.03	0.97	1.01	0.95	0.95	0.93	0.99	0.971
Mag Scale z	1.24	1.29	1.18	1.27	1.15	1.16	1.24	1.38	1.29	1.07	1.227

MPU#4	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	-14.38	12.59	21.58	-5.39	0	-7.19	8.99	0	7.19	-16.18	0.721
Mag Bias y	151.04	133.06	131.26	152.84	151.04	156.43	174.42	158.23	122.27	136.66	146.725
Mag Bias z	-466.36	-481.96	-481.96	-478.49	-492.36	-499.3	-514.9	-488.89	-495.83	-499.3	-489.935
Mag Scale x	0.96	0.99	0.96	1	1.02	1.03	1.02	1.03	1.01	1.01	1.003
Mag Scale y	0.87	0.87	0.93	0.86	0.91	0.9	0.92	0.91	0.92	0.9	0.899
Mag Scale z	1.24	1.19	1.13	1.19	1.08	1.09	1.07	1.08	1.09	1.12	1.128
MPU#5	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	21.58	-17.98	-34.16	-17.98	-14.38	-5.39	0	8.99	0	0	-5.932
Mag Bias y	301.1	268.84	277.8	304.68	276	274.21	308.27	310.06	322.6	319.02	296.258
Mag Bias z	-261.78	-213.24	-312.06	-319	-256.58	-338.07	-296.46	-263.52	-273.92	-263.52	-279.815
Mag Scale x	1	0.93	1.01	1.06	1.01	1.03	1.04	1.01	1.03	1.01	1.013
Mag Scale y	0.93	0.87	0.95	0.92	0.88	0.93	0.89	0.89	0.9	0.92	0.908
Mag Scale z	1.07	1.29	1.05	1.03	1.15	1.05	1.09	1.13	1.08	1.08	1.102
MPU#6	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	-124.23	-127.78	-118.9	-120.68	-122.45	-117.13	-115.35	-118.9	-110.03	-113.58	-118.903
Mag Bias y	439.45	423.37	441.24	444.81	448.38	484.11	434.09	426.95	430.52	466.25	443.917
Mag Bias z	-106.4	-114.98	-101.25	-58.35	-73.79	-142.44	-109.83	-94.39	-101.25	-94.39	-99.707
Mag Scale x	1	1.02	1.06	0.97	0.94	0.98	1.02	0.99	0.98	1	0.996
Mag Scale y	0.92	0.94	0.91	0.89	0.92	1.05	0.94	0.91	0.91	0.94	0.933
Mag Scale z	1.09	1.05	1.04	1.19	1.17	0.98	1.05	1.12	1.13	1.07	1.089

MPU#7	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	56.98	49.85	39.17	33.83	42.73	51.64	46.29	74.78	58.76	64.1	51.813
Mag Bias y	665.92	683.72	671.26	690.84	680.16	673.04	649.89	706.87	657.01	633.87	671.258
Mag Bias z	-1.72	-31	-27.55	-36.16	-20.66	-46.49	-17.22	-20.66	-22.39	-20.66	-24.451
Mag Scale x	0.95	0.99	0.99	0.99	0.99	0.98	0.95	0.95	0.95	0.89	0.963
Mag Scale y	0.85	0.85	0.85	0.82	0.86	0.88	0.86	0.87	0.88	0.92	0.864
Mag Scale z	1.29	1.24	1.24	1.29	1.21	1.2	1.28	1.25	1.23	1.26	1.249
MPU#8	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	-55.92	-68.55	-77.57	-73.96	-81.18	-70.35	-50.51	-50.51	-64.94	-48.71	-64.22
Mag Bias y	195.46	200.89	224.42	193.65	229.85	211.75	213.56	228.04	226.23	217.18	214.103
Mag Bias z	-250.49	-213.96	-238.32	-224.4	-219.18	-247.01	-153.08	-116.55	-189.61	-193.09	-204.569
Mag Scale x	0.88	0.92	0.9	0.92	0.87	0.89	0.95	1.02	0.97	0.92	0.924
Mag Scale y	0.95	0.99	1	1.05	0.95	0.99	1.05	1.06	1	0.99	1.003
Mag Scale z	1.23	1.1	1.13	1.04	1.26	1.15	1	0.93	1.03	1.11	1.098
MPU#9	1	2	3	4	5	6	7	8	9	10	Average
Mag Bias x	-278.62	-282.17	-282.17	-276.85	-287.5	-278.62	-278.62	-282.17	-291.05	-285.72	-282.349
Mag Bias y	92.89	87.53	94.68	89.32	82.17	91.11	80.39	91.11	78.6	100.04	88.784
Mag Bias z	233.39	226.52	259.13	209.36	274.58	228.24	235.11	252.27	284.87	221.38	242.485
Mag Scale x	0.95	0.97	0.97	0.98	0.94	1	0.99	0.97	0.95	1.03	0.975
Mag Scale y	0.97	0.98	0.94	0.98	0.96	0.98	0.94	0.96	0.92	0.98	0.961

Mag Scale z	1.08	1.05	1.1	1.04	1.11	1.02	1.08	1.09	1.15	1	1.072

D.2 Method Comparison Test

For the first part of the validation tests we focused on comparing the new algorithm of the MPU using the magnetometer with the old algorithm which was not based on Kris Winer's code but on Jeff Rowberg's (Rowberg) DMP method. This method was not designed for implementing the magnetometer in the algorithm. And thus the main question arises; how do we achieve a comparison in equal grounds.

An idea would be to test the different systems by engaging multiple test subjects that have used both of the setups through a simulation, with a questionnaire. Assuming that the subject pool was big enough, we could extract very important results based on user experience.

There a two main problems though. Firstly, we would receive many different subjective opinions, depending on what the subject prioritizes as an important feature. For example, an experienced subject that has used MoCap technology before (e.g in PlayStation VR™) could prioritize gaming-based features such as display delay, fast responses etc. On the other hand a subject that has no experience with this technology might focus on the best matching of its movements with the simulation display. Therefore, for each questionnaire we should assess the subject's background, or try to attract subjects with a relatively similar background, both of which are very time-consuming processes. The second problem is derived from the placing or matching of the sensor with a body part. Due to the differences in range of motion for each joint, the two systems, when only using a single sensor, could be evaluated differently and thus produce contradicting results.

To achieve a just testing method, we chose to mount a sensor on a controlled 6-Volt DC motor and visually compare through various rotations, the response of the two algorithms. Moreover we simulated different motion patterns, by moving the MPU in various directions and compared the behavior of each method.

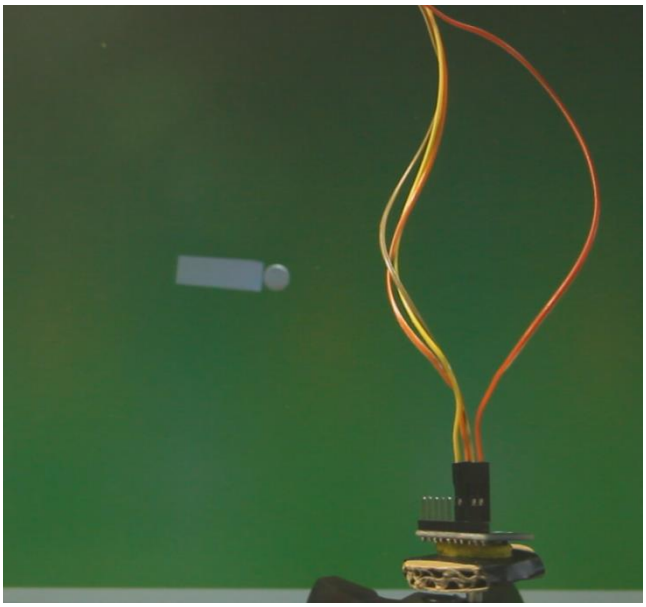


Figure 54: MPU with Rowberg's Algorithm Tested with DC Motor

For each session of DC motor testing the same setup and starting point was used between the two methods. After each method's filming was finished, we returned the MPU to its original starting point and only changed the code in the Arduino that was connected with it. Various rotations were tested, with the most accurate for the setup being the Yaw-type rotation displayed in Figure 55. As for the rotation speed, we used a pattern were we would reach a maximum speed to allow the motor to start its rotation, by surpassing its initial stall torque and after 0,2 seconds lower the speed by 60% and maintain it at that point providing close to 7,5 rpm.

For the spatial movement evaluation we performed a series of various movements which later assessed though video. The movements were both focusing on a single type of rotation and random possible rotations that could occur in a real simulation.

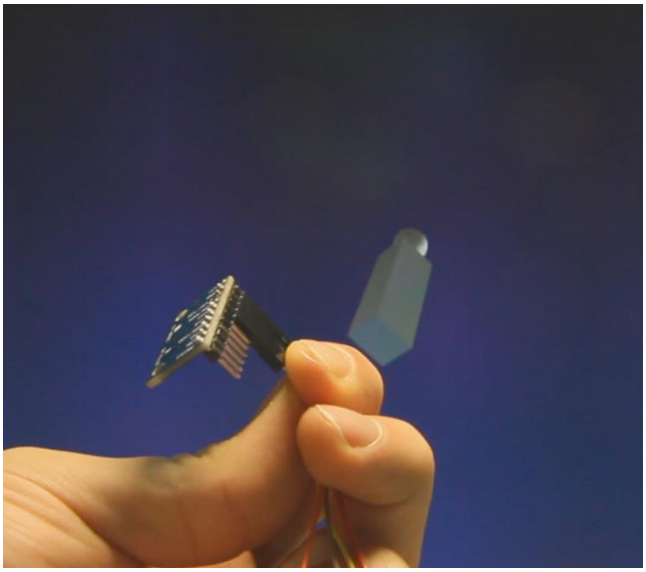


Figure 55: Recreating Various Movement Patters Using the New MPU Code

D.3 Result Evaluation

Through the previous tests many differences between the two systems were revealed. These differences are a mean of evaluation, when choosing which algorithm should be used and when focusing on improving each setup. The positive and negative characteristics of single MPU testing for each setup are described in the following points:

Rowberg's Algorithm:

+ The response of the MPU to the algorithm had no perceivable delay.
+ There was no need for sensor calibrating, hence the algorithm could run correctly on any MPU with no previous changes in preset values.
+ The algorithm displayed no jitter in the single MPU tests.

− The MPU was displaying a constant Yaw-drift that became significant with time (Figure 56). Especially during Yaw DC motor rotations with the same direction with drift components, for each full Yaw rotation of the MPU, up to 90^{o} of extra rotation were added through drift. Note that in Figure 56 the mpu and the display are almost vertical, after a full rotation.

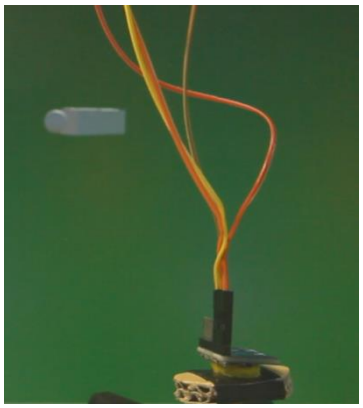


Figure 56: Rowberg's Algorithm Drift

− The drift was also affecting other single rotation movements and seemed to be always adding to Yaw component. Therefore when performing a full Roll rotation the drift was turning the MPU clockwise in the z axis from 0^{0} to 180^{0} and anticlockwise from 180^{0} to 360^{0}, resulting in a vacillating movement.
− The algorithm also seemed to crash in two different ways. From time to time, data feeding stopped and the Arduino needed to be reset, this was probably an effect of the interrupt pin function of on Rowberg's code. On the other hand, sometimes the MPU displayed random movements and completely lost its orientation, like reaching a singularity point.
− Lastly the MPU was not Earth-bound, meaning that it would begin its spew of data with a random starting orientation in Yaw axis and therefore needed an adjustment in the display, so that its movements can be matched with the Unity object's movements.

The Current Algorithm:

+ The response of the MPU was depended on the orientation of the module, always calculating a starting position that was relevant with Earth's magnetic North.
+ There was matching of movement done with the module and display data.
+ There were no drift effects
+ There was no need for sensor calibrating, but the calibration presets of the sensor need to be added in the code beforehand.

– The output of each sensor was very depended on setting correct calibration presets. If not chosen correctly, wrong presets could result in restrictions of certain rotations or confusion between rotations (Figure 57).

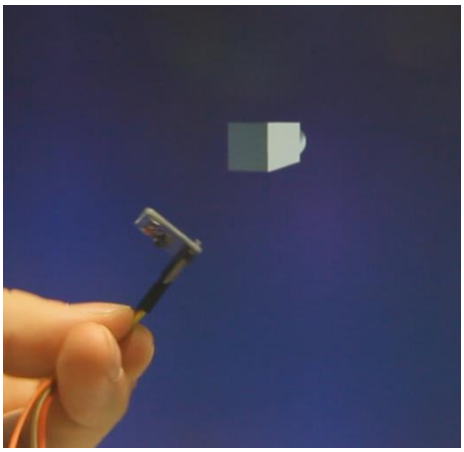


Figure 57: MPU with Incorrect Presets Confusing Rotations

– The sensor had noticeably more delay, but not as much, as to not be able to correlate its movements with the display.
– The response of the sensor was degrading after several minutes of operation, resulting in static errors.

D.4 Upper Body Simulation

In this part, the suit and its use and fitting procedure will be described, along with a display of photos taken in a trial operation session. Also we would like to thank L. Psarakis, a member of the ergonomics unit for agreeing to participate in this session and allowing the display of the following images.

The fitting procedure of the upper body system is as follows:

- A chest vest for supporting the main unit on the subjects back is worn.
- The main unit including the Arduino Nano, the multiplexer, the Bluetooth module and the first MPU sensor is installed on the vest using Velcro fabric.
- Each sensor is sequentially strapped on the correct position, along with its cables that are connected to main unit.
- After all sensors are placed one by one, adjustments are made so that the sensors are aligned as seen on Figure 58.

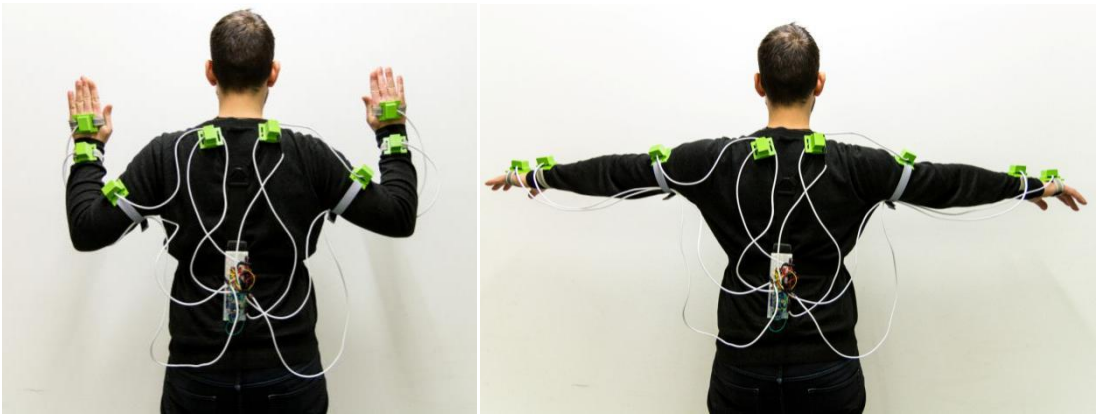


Figure 58: The Upper Body MARG System Trial Fit

It is worth noting that the straps and the vest can be fitted over clothing as seen on Figure 59, but it is suggested that when testing, at least for the arm sensors no sleeves should be worn to avoid unwanted sensor slipping and testing delay.

Figure 59: Upper Body MARG Fitted Over Clothing

E. Conclusion

E.1 Future System Improvements

With the conclusion of this thesis significant steps forward have been made from the previous systems. The improvements were both in terms of design, modularity and number of sensors used and in terms of coding efficiency and accuracy. Of course the setup now resembles more of a full-body training suit, but is not yet as versatile and as robust as a commercial issued suit. There are many improvements to be made, although it is worth noting that, the system was deliberatively designed and engineered as a low-cost, in-house and easy-to-manufacture modular motion capture setup.

Firstly, to achieve full modularity we could incorporate Rx/Tx transmitters for each case and each MPU, having multiple low cost slave modules sending data to a single master module connected with the PC-Unity setup. Of course that translates to an increased cost of construction, a more sophisticated receiver/master and a more complex design in terms of modular MPU electronics board and case. On the other hand, one could incorporate as many MPUs as needed, for each test, as long as there is enough computation power and receiver capabilities, since no cables would need to be connected.

Secondly, the design of the hub for each MPU could be modified, so that it can be mass produced, through a different manufacturing method and not the slow and costly method of 3D-printing. Plastic injection molding manufacturing, could greatly lower the cost for mass production, although the design of the case should change so that there is proper flow of the material, ventilation and as well as a gate and a runner imprint caused from the mold design Plastic Part Design for Injection Molding. Moreover, the casing enclosure type should be altered from a sliding fit, two-part enclosure to a snapping half mold design. An example of basic modifications on the design can be seen in Figure 60.

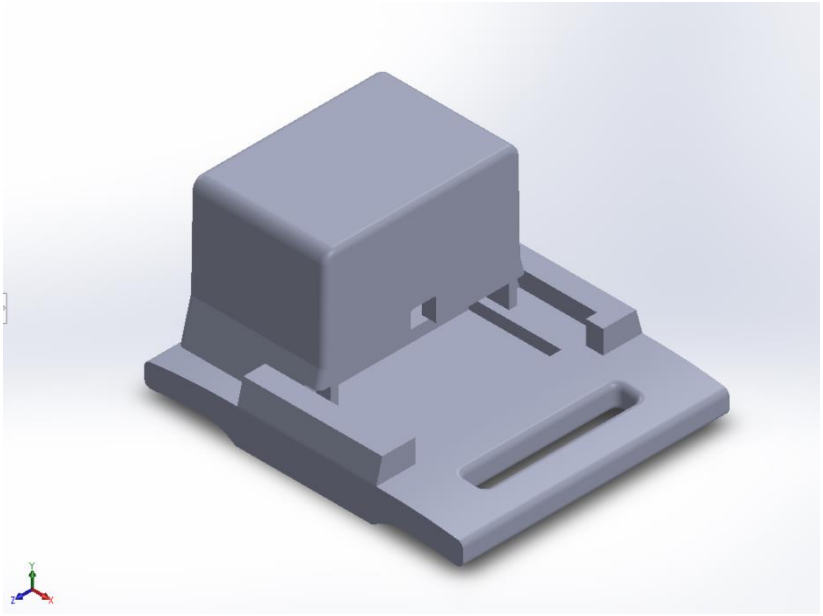


Figure 60: Simplified Casing for Injection Mold Manufacturing

The main changes that needed to be made in the design focus around having a uniformed wall thickness and simplified geometry so that the mold can be easily manufactured. The fitting method and the cable hole could also be changed so that the design of the mold is not costly, requiring adjustment to the many crevices and inner angles of the slide-fit closing method.

It has to be pointed out, though, that 3D-printing, as a manufacturing method, was chosen for its flexibility, availability and due to a small amount of cases needed, since the system is used for experimental training purposes. Therefore a manufacturing method for mass production is not yet required.

Lastly another idea for improvement revolves around creating a reference system by either fusing Oculus data or metrics from an outer calibrated MPU array, to achieve simulation of not only rotation of joints but movement of the whole body through the laboratory's testing space. Such idea has been previously attempted (Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM) using a single inertial sensor. Thus, further research and modification to the main algorithm should be made so that we can achieve multiple MPU and vision fusing, into an immersive graphic environment.

E.2 Final Evaluation and Future Research

The results from both the algorithm comparison test and the full body simulation were encouraging and showed new possibilities on creating an accurate testing system. The new algorithm seems to be more reliable but still needs some fine-tuning to surpass Rowberg's algorithm in every aspect. A choice has to be made on the use of an appropriate algorithm and research on improving the efficiency and performance of the code. Moreover, as mentioned in the previous chapter, there are many technical improvements and additions that could be implemented in the system. Of course, further development and validation is also welcome, so as to fail-proof and adjust the system to the possible difficulties and problems that might be revealed throughout a vigorous testing session. Therefore, before completing this thesis we would like to mention some possible future uses of the system assisting in various types of research:

A. Work Posture Monitoring

The idea of work posture monitoring usually requires very lengthy monitoring sessions and human observers or camera recording. That usually results in tampering with the daily working tasks of the subject. With the use of wearable wireless setup we could minimize the intrusive nature of such a system, allowing for more representative data gathering from the subject's usual working posture.

B. Heavy Machinery Control Simplification

By communicating with a controller for heavy machinery, we could translate simple human gestures and upper body motions to commands or even functions. An easier method of learning the use of a complicated machine could be very beneficial for industrial operations.

C. Upper Body Rehabilitation

In physiotherapy treatments, such a system could be very helpful for the expert to monitor patient movement range and response to certain techniques. This idea though, needs further adjustment of data display, so as to translate the quaternions or Unity's visual representation, to meaningful and useful information for the physiotherapist.

Bibliography

3D Systems. (2013). CubeX 3D printer User guide. Rock Hill, South Carolina, United States of America: 3D Systems, Inc.

Anderson, D., Anghel, A., Yumoto, K., Ishitsuka, M., & Kudeki, E. (2002). Estimating daytime vertical ExB drift velocities in the equatorial F-region using ground-based magnetometer observations. *Geophysical Research Letters, 29(12)*, 37-1-37-4.

Arduino. (n.d.). *Arduino Nano Tech Specs.* Retrieved 1 21, 2019, from Arduino Store: https://store.arduino.cc/arduino-nano

Bacarin, T. A., Sacco, I. C., & Henning, E. M. (2009). Plantar pressure distribution patterns during gait in diabetic neuropathy patients with a history of foot ulcers. *Clinics 64(2)*, 113-120.

Bideau, B., Kulpa, R., Vignais, N., Brault, S., Multon, F., & Craig, C. (2010). Using Virtual Reality to Analyze Sports Performance. *IEEE Computer Graphics and Applications 30(2)*, (pp. 14-21).

Bishop, R. H. (2008). *Mechatronic Systems, Sensors, and Actuators: Fundamentals and Modeling, The mechatronics handbook.* Boca Raton: FL: CRC Press Inc.

Bluetooth v4.0 Specification CSR8510A10. (n.d.).

Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3), 257-267.

Bottlang, M., Mady, S. M., Steyers, C. M., Marsh, J. L., & Brown, T. D. (2000). Assessment of elbow joint kinematics in passive motion by electromagnetic motion tracking. *Journal of Orthopaedic Research, 18(2)*, 195-202.

Braune, W., & Fisher, O. (1988). *Determination of the moments of inertia of the human body and its limbs.* Berlin: Springer-Verlag.

Captiks. (n.d.). *Captiks Capturing Kinematics*. Retrieved 11 9, 2018, from Captiks: http://www.captiks.com/

Cavallo, A., Cirillo, A., Cirilli, P., De Maria, G., Falco, P., Natale, C., et al. (2014). Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation. *19th IFAC World Congress*, (pp. 7585-7591). Cape Town, South Africa.

Cayson, G. (n.d.). *Chapter 4: Shoulder Girdle*. Retrieved 1 11, 2019, from Shoulder Anatomy Part II: https://slideplayer.com/slide/3319759/

Chen, K. Y., Patel, S. N., & Keller, S. (2016). Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, (pp. 1504-1514).

Chen, X. (2013). *Human motion analysis with wearable inertialsensors. Ph.D. dissertation.* Tennessee: University of Tennessee.

Chen, X., & Koskela, M. (2013). Online RGB-D gesture recognition with extreme learning machines. *15th ACM on International conference on multimodal interaction* , (pp. 467-474).

Cirillo, A., Cirillo, P., De Maria, G., Natale, C., & Pirozzi, S. (2016). A comparison of multisensor attitude estimation algorithms. In H. Fourati, & D. Belkhiat, *Multisensor Attitude Estimation: Fundamental Concepts and Applications* (pp. Chapter: 29, 529-539). CRC Press.

Corrales, J. A., Candelas, F. A., & Torres, F. (2010). Kalman Filtering for Sensor Fusion in a Human Tracking System. *Intech*.

Cotton, D. P., Graz, I. M., & Lacour, S. P. (2009). A multifunctional capacitive sensor for stretchable electronic skins. *IEEE Sensors Journal , 9(12)*, 2008-2009.

Datta, V., Mackay, S., Mandalia, M., & Darzi, A. (2001). The use of electromagnetic motion tracking analysis to objectively measure open surgical skill in the laboratory-based model1. *Journal of the American College of Surgeons 193.5*, 479-485.

Desroches, G., Aissaoui, R., & Bourbonnais, D. (2006). Effect of system tilt and seat-to-backrest angles on load sustained by shoulder during wheelchair propulsion. *Journal of Rehabilitation Research and Development, 43(7)*, 871.

Digi-Key's European Editors. (2013, 5 9). *Sensor Technologies for Accelerometers*. Retrieved 10 25, 2018, from Digi-Key Electronics: https://www.digikey.ca/en/articles/techzone/2013/may/sensor-technologies-for-accelerometers

Doscher, J. (2008, 12 13). *Accelerometer Design and Applications*. Retrieved from Analog Devices: https://www.analog.com/en/index.html

Duffy, V. G. (2007). Duffy, V. G. (2007). A methodology for assessing industrial workstations using optical motion capture integrated with digital human models. *Occupational Ergonomics, 7(1)*, 11-25.

Dunne, L. E., Smyth, B., & Caulfield, B. (2007). A Comparative Evaluation of Bend Sensors for Wearable Applications. *11th IEEE International Symposium on Wearable Computers* (pp. 1-2). Boston, MA, USA: University College Dublin.

Ewing Rachel. (2015). Runners Get a Step Ahead at Drexel. *DrexelNOW*.

Fifth Dimension Technologies 5DT. (n.d.). *Data Gloves*. Retrieved 10 30, 2018, from Fifth Dimension Technologies 5DT.

Finkenberg, M. E., & Mohsen, B. (2003). Virtual Reality Applications in Physical Education. *Journal of Physical Education, Recreation & Dance Volume 74 - Issue 9*, 13-15.

Fitzgerald, D., Foody, J., Kelly, D., Ward, T., Markham, C., McDonalad, J., et al. (2007). Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation exercises. *29th Annual International Conference of the IEEE EMBS*, (pp. 4870-4874).

Gleicher, M. (1999). Animation from observation: Motion capture and motion editing. *ACM SIGGRAPH Computer Graphics 33.4*, 51-54.

Grantcharov, T. P., Kristiansen, V. B., Bendix, J., Bardram, L., Rosenberg, J., & Funch-Jensen, P. (2004). Randomized clinical trial of virtual reality simulation for laparoscopic skills training. *British Journal of Surgery, 91(2)*, 146-150.

Greenemeier, L. (2011, 9 27). *Something in the Way You Move: Cameras May Soon Recognize Criminals by Their Gait [Video]*. Retrieved 10 29, 2018, from SCIENTIFIC AMERICAN: https://www.scientificamerican.com/article/motion-capture-surveillance/

Greiff, P., Boxenhorn, B., King, T., & Niles, L. (1991). Silicon monolithic micromechanical gyroscope. In Solid-State Sensors and Actuators. *1991 International Conference on. IEEE*, (pp. 966-968).

Hamilton, W. R. (1866). *Elements of quaternions.* Longmans, Green, & Company.

HC-05 Bluetooth Module User's Manual V1.0. (n.d.).

Hislop, H., Avers, D., & Brown, M. (2013). *Daniels and Worthingham's muscle Testing-E-Book: Techniques of manual examination and performance testing.* Elsevier Health Sciences.

Hwang, B. W., Kim, S., & Lee, S. W. (2006). A full-body gesture database for automatic gesture recognition. *IEE*, (pp. 243-248).

InvenSense Inc. (2016). *MPU-9250 Product Specification Revision 1.1.* San Jose: InvenSense Inc.

Jouybari, A., Ardalan, A. A., & Rezvani, M.-H. (2017). Experimental Comparison between Mahony and Complementary Sensor Fusion Algorithm for Attitude Determination by Raw Sensor Data of XSens IMU on Buoy. *Tehran's Joint ISPRS Conferences of GI Research, SMPR and EOEC 2017* (pp. 497-502). Tehran, Iran: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

JuniorMed Centrum Medyczne. (n.d.). *EEG Biofeedback*. Retrieved 10 22, 2018, from JuniorMed Centrum Medyczne: http://www.poradniajuniormed.pl/class/eeg-biofeedback/

Kahle, W., Leonhardt, H., Platzer, W., Palmer, E., & Platzer, W. (2004). *Color atlas and textbook of human anatomy. Vol. 1, Locomotor system.* Thieme.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45.

Kanumuri, P., Ganai, S., Wohaibi, E., Bush, R. W., Grow, D. R., & Seymour, N. E. (2008). Virtual reality and computer-enhanced training devices equally improve laparoscopic surgical skill in novices. *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, 12(3), 219.

Karakikes, M. (2017). *Development and Evaluation of a Wearable Motion Tracking System, to Support Hand-Tool Design.* Athens: Εκδόσεις Εθνικού Μετσόβιου Πολυτεχνείου.

Kilani, M. (n.d.). *MEMS Class 6 Microsensors*. Retrieved 10 25, 2018, from SlidePlayer: https://slideplayer.com/slide/6940332/

Kirillov, A. (2007, 3 27). *Motion Detection Algorithms*. Retrieved 10 29, 2018, from Code Project: https://www.codeproject.com/Articles/10248/Motion-Detection-Algorithms

Kontrazis, P. (2018). *Creating a virtual environment and scenarios for studying human robot collaboration in virtual reality. Diploma Thesis.* Athens: National Technical University of Athens.

Konvalin, C. (2009, 12 1). *Compensating for Tilt, Hard-Iron, and Soft-Iron Effects*. Retrieved 1 23, 2019, from Sensors Online: https://www.sensorsmag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects

Kortier, H. G., Sluiter, V. L., Roetenberg, D., & Veltink, P. H. (2014). Assessment of hand kinematics using inertial and magnetic sensors. *ournal of NeuroEngineering and Rehabilitation*, 11,70.

Kramer, J. P., Linderner, P., & George, W. R. (1991). *Patent No. US5047952.* United States of America.

Kuipers, J. B. (1999). Quaternions and Rotation Sequences. *Geometry, Intefrability and Quantization*, 127-143.

Langford, G. B. (1990). *Patent No. US07552575.* United States of America.

Lee, G. X., Low, K. S., & Taher, T. (2010). Unrestrained Measurement of Arm Motion Based on Wearable Wireless Sensor Network. *IEEE Transactions on Instrumentation and Measurement*, 59(5), 1309-1317.

Lin, J. J., Hanten, W. P., Olson, S. L., Roddey, T. S., Sotoquijano, D. A., Lim, H. K., et al. (2005). Functional activity characteristics of individuals with shoulder dysfunctions. *ournal of Electromyography and Kinesiology, 15(6)*, 576-586.

Lowood, H. E. (2018, 9 24). *Virtual reality*. Retrieved 10 15, 2018, from Encyclopedia Britannica: https://www.britannica.com/technology/virtual-reality

Madgwick, S. O. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25: 113-118.

Mahony, R., Hamel, T., & Pflimlin, J.-M. (2008). Nonlinear Complementary Filters on the Special. *IEEE Transactions on Automatic Control 53 (5)* (pp. 1203-1217). Institute of Electrical and Electronics Engineers.

Marey, E. (1874). *Animal Mechanism: A Treatise on Terrestrial and Aerial Locomotion.* London: Henry S. King & Co.

Mayo Clinic. (2018, 1 3). *Biofeedback*. Retrieved 10 22, 2018, from Mayo Clinic: https://www.mayoclinic.org/tests-procedures/biofeedback/about/pac-20384664

Mirelman, A., Patritti, B. L., Bonato, P., & Deutch, J. E. (2010). Effects of virtual reality training on gait biomechanics of individuals post-stroke. *Gait & posture*, 31(4), 433-437.

Moeslund, T. B., & Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer vision and image understanding, 81(3)*, 231-268.

Motion Analysis. (n.d.). *About Motion Analysis*. Retrieved 11 9, 2018, from Motion Analysis: https://motionanalysis.com/contact/about-motion-analysis/

Mourelatos, A. (2018). *Development and evaluation of a wearable motion tracking system for sensorimotor tasks in VR environments.* Athens: Εκδόσεις Εθνικού Μετσόβιου Πολυτεχνείου.

MPU-9250 Product Specification Revision 1.0. (2014, 1 17). Retrieved 1 21, 2019, from Inven Sense Inc.

Mündermann, L., Corazza, S., & Andriacchi, T. P. (2015). The evolution of methods for the capture of human movement. *Journal of NeuroEngineering and*, 1-2.

Muybridge, E. (1887). *Animal locomotion.* Philadelphia: J.B. Lippincott Company.

Natural Resources Canada. (2017, 2 24). *Magnetic declination calculator*. Retrieved 1 22, 2019, from Natural Resources Canada: http://www.geomag.nrcan.gc.ca/calc/calc-en.php

Neely, J. S., & Restle, P. J. (1997). *Patent No. US5610528A.* United States of America.

Neverova, N., Wolf, C., Taylor, G., & Nebout, F. (2016). Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(8)*, (pp. 1692-1706).

Panjabi, M. M., Oxland, T. R., Yamamoto, I., & Crisco, J. J. (1994). Mechanical behavior of the human lumbar and lumbosacral spine as shown by three-dimensional load-displacement curves. *JBJS, 76(3)*, 413-424.

Papadopoulos, E. (2017). Intelligent Control Systems & Robotics, Lecture Notes. *Κεφάλαιο 2: Θέση και Προσανατολισμός*. Athens, Greece.

Polhemus. (n.d.). *FASTRAK®*. Retrieved 10 29, 2018, from Polhemus: https://polhemus.com/motion-tracking/all-trackers/fastrak/

Pruthviraj, N. (2012, 8 23). Presentation of Shoulder Examination.

Raab, F., Blood, E., Steiner, T., & Jones, H. (1979). Magnetic Position and Orientation. *IEEE Transactions on Aerospace*, (pp. 709–718).

Rainbow, M. J., Wolff, A. L., Crisco, J. J., & Wolfe, S. W. (2016). Functional kinematics of the wrist. *Journal of Hand Surgery (European Volume), 41(1)*, 7-21.

Ramsden, E. (2001). *Hall Effect Sensors: Theory & Application.* Advanstar Communications Incorporated.

Rana, M., Taylor, G., Spiro, I., & Bregler, C. (2012). Skeletal reconstruction from low-resolution multi-view images. In Computer Vision and Pattern Recognition Workshops (CVPRW). *IEEE Computer Society Conference*, 58-63.

Range of Joint Motion Evaluation Chart. (2014, 3). Washington, United States of America: Department of Socail & Health Services.

Richardson, J. (2003, 1 6). Interview with Jim Richardson, NaturalPoint President. (J. Toll SimHQ, Interviewer)

Rosebrock, A. (2015, 5 25). *Basic motion detection and tracking with Python and OpenCV*. Retrieved 10 29, 2018, from pyimagesearch: https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/

Rowberg, J. (n.d.). *i2cdevlib*. Retrieved 2 13, 2019, from Github: https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050

SANLAB. (2015). *HEAVY EQUIPMENT SIMULATOR*. Retrieved 11 20, 2018, from SANLAB: http://www.sanlab.net/eng/?p=276/

Schecter, S. (2014, 5 9). *What is markerless Augmented Reality? | AR Bites*. Retrieved 10 18, 2018, from Marxent Labs: https://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/

Schopp, P., Klingbeil, L., Peters, C., Buhmann, A., & Manoli, Y. (2009). Sensor fusion algorithm and calibration for a gyroscope-free IMU. *Procedia Chemistry, 1(1)*, 1323-1326.

Sensor Wiki. (2011, 11 15). *Flexion*. Retrieved 10 30, 2018, from Sensor Wiki: http://www.sensorwiki.org/doku.php/sensors/flexion#fn__6

Seymour, N. E., Gallagher, A. G., Roman, S. A., O'brien, M. K., Bansal, V. K., Andersen, D. K., et al. (2002). Virtual Reality Training Improves Operating. *Annals of surgery*, 236(4), 458.

Sheinker, A., Frumkis, L., Ginzburg, B., Salomonski, N., & Kaplan, B. Z. (2009). Magnetic anomaly detection using a three-axis magnetomete. *IEEE Transactions on Magnetics, 45(1)*, 160-167.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: modelling, planning and control.* Springer Science & Business Media.

Simone, L. K., & Kamper, D. G. (2005). Design considerations for a wearable monitor to measure finger posturE. *Journal of NeuroEngineering and Rehabilitation*, 2,5.

Sreejan, A., & Narayan, Y. S. (2017). Development of Flex Sensor Array to Identify Damage on Sheet Metal. *International Journal of Engineering Science and Technology. 09*, 967-974.

Standen, P. J., & Brown, D. J. (2006). Virtual reality and its role in removing the barriers that turn cognitive impairments into intellectual disability. *Virtual Reality, 10 (3-4)*, 241-252.

Stutzke, N. A., Russek, S. E., & Pappas, D. P. (2005). Low-frequency noise measurements on commercial magnetoresistive magnetic field sensors. *Journal of Applied Physics 97*, 10Q107.

Sullivan, E. (n.d.). Chapter 6: The Elbow and Radioulnar Joints 6-1.

The Franklin Institute. (n.d.). *HISTORY OF VIRTUAL REALITY*. Retrieved 10 15, 2018, from The Franklin Institute: https://www.fi.edu/virtual-reality/history-of-virtual-reality

Tinder, R. F. (2007). *Relativistic Flight Mechanics and.* Morgan & Claypool.

University of Washington, Ubicomlab, Chen et al. (n.d.). *Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing*. Retrieved from ubicomplab: https://ubicomplab.cs.washington.edu/publications/finexus/

Vita, E. (2014). Schematic of Arduino Nano-Rev3.2. *Arduino Nano*. Gravitech.

von Bergmann, J., & von Bergmann, H. (2007). Foucault pendulum through basic geometry. *American Journal of Physics, 75(10)*, 888-892.

Washington State University. (n.d.). *DR. ROBERT CATENA'S Gait and Posture Biomechanics Lab*. Retrieved 10 22, 2018, from Washington State University: https://labs.wsu.edu/biomechanics/

Weber, W., & Weber, E. (1836). *Mechanik der menschlichen Gehwerkzeuge.* Dieterich.

Williams, G., Taylor, G., Smolskiy, K., & Bregler, C. (2010). Body Motion Analysis for Multi-Modal Identity Verification. *2010 International Conference on Pattern Recognition* (pp. 2198-2201). Dept. of Computer Science, Courant Institute, New York University.

Winer, K. (2017, 8 15). *Simple and Effective Magnetometer Calibration*. Retrieved 1 24, 2019, from Github: https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration

Winer, K. (n.d.). *Arduino sketches for MPU9250 9DoF with AHRS sensor fusion*. Retrieved 2 1, 2019, from Github: https://github.com/kriswiner/MPU9250

Wu, G., Van der Helm, F. C., Veeger, H. D., Makhsous, M., Van Roy, P., Anglin, C., et al. (2005). ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—Part II: shoulder, elbow, wrist and hand. *Journal of biomechanics, 38(5)*, 981-992.

Xian, W., Tarrío, P., Metola, E., Bernardos, A. M., & Casar, J. R. (2012). Gesture Recognition Using Mobile Phone's Inertial Sensors. In S. Omatu, S. J. De Paz, S. González, J. Molina, A. Bernardos, & J. Rodríguez, *Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing, vol 151.* (pp. 173-184). Heidelberg: Springer, Berlin.

Xsens. (n.d.). *Ergonomics and Human factors*. Retrieved 11 9, 2018, from Xsens: https://www.xsens.com/tags/ergonomics-human-factors/

Xu, D., Xia, Y., & Mandic, D. P. (2016). Optimization in quaternion dynamic systems: gradient, Hessian, and learning algorithms. *IEEE transactions on neural networks and learning systems, 27(2)*, (pp. 249-261).

Yazdi, N., Ayazi, F., & Najafi, K. (1998). Micromachined inertial sensors. *IEEE, 86(8)*, (pp. 1640-1659).