VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ESTIMATING HUMAN MOVEMENT USING ACCELEROMETERS

VYHODNOCOVÁNÍ LIDSKÉHO POHYBU POMOCI AKCELEROMETRŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR‘S THESIS

AUTOR PRÁCE
AUTHOR

Tomáš Matula

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Roland Dobai, Ph.D.

BRNO 2016

Abstrakt

Tato práce je zaměřena na analýzu lidského pohybu, zejména měření úhlu kolena, co je důležité především při procesu rehabilitace u pacientů s protézy kolenního kloubu nebo po operaci. Pro měření IMU - inerciální měřící jednotky od firmy Xsens jsou použité, přičemž pouze data z 3-osého akcelerometru a gyroskopu jsou zahrnuty. Vhodné umístění jednotek je vybráno, jakož i metody pro kalibraci a následný výpočet úhlu ze zaznamenaných dat. Tyhle metody jsou implementovány a experimentálně ověřeny. Experimenty ukazují, že výsledky jsou docela přesné, a toto řešení je použitelné při analýze pacientů například provedením monitorování chůze.

Abstract

This work is aimed at analysis of human movement, especially measurement of knee angle, which is important data for monitoring in process of rehabilitation on patients with knee arthroplasty or after surgery. For measurement IMUs - Inertial Measurement Units from Xsens are used, while only data from 3-axis accelerometer and gyroscope are gathered. Appropriate unit position is chosen, as well as methods for calibration and angle calculation from stored measured data. These methods are implemented and experimentally validated. Experiments shows, that results are pretty accurate and this solution is useful in analysis of patients for example by doing gait analysis.

Klíčová slova

Měření úhlu, koleno, analýza chůze, IMU, inerciální měřící jednotka, akcelerometr, gyroskop, komplementární filtr

Keywords

Angle measurement, knee, gait analysis, IMU, Inertial measurement unit, accelerometer, gyroscope, complementary filter

Citation

Matula Tomáš: Estimating human movement using accelerometers, bachelor’s thesis, Brno, FIT BUT, 2016

Estimating human movement using accelerometers

Declaration

I declare that I made this thesis on my own, under supervision of Dr Roland Dobai from BUT in Czech Republic and Dr Hans Hallez from KU Leuven in Belgium, and in consultation with physiotherapist Henri De Vroey from KU Leuven in Belgium. All sources and literature that I have used are cited with reference to the corresponding source.

……………………
Tomáš Matula
May 17, 2016

Acknowledgements

In this section I would like to thank my supervisor Dr Hans Hallez for his lead and suggestions, Mr Henri De Vroey for physiotherapist background and for assistance with measurements, and Dr Roland Dobai for his remote support during my Erasmus study.

Contents

1 Introduction

Three-dimensional computer analysis of human motion and respective body segments is useful tool for clinical evaluation notably in the orthopaedic and rehabilitation fields because it makes easier the evaluation, storing or comparing measured results. Many tracking systems based on electro-mechanical, optical, magnetic or ultrasonic technologies are already available [1] however they require special laboratories, complicated setup or their high cost make them prohibitive for ordinary application. This work is focused on analysis of the knee joint angle to allow evaluation of gait analysis which is important to monitor in process of rehabilitation on patients with knee arthroplasty [2].

Inertial Measurement Units (IMUs) are used in order to increase accuracy because these days it is quite simple and cheap to get these precise units and use them in measurements. However, with use of this devices few fundamental problems arise for example sensor placement, measurement or evaluation of data. This work is aimed at addressing the issue of knee angle measurement by choosing appropriate methods for calibration and for angle calculation as well as implementation and experimental evaluation.

The rest of the thesis is organized as follows. Chapter 2 describes IMUs with its problems. Sensor placement, measurement problems and methods for calibration and calculation are described in Chapter 3. Comparison of methods, detailed specification of chosen one and description of implementation are done in Chapter 4. The experimental results are discussed in Chapter 5. Chapter 6 concludes the thesis and discuss possible future directions.

2 Inertial Measurement Unit

The IMU is an electronic device consisting of multiple 3-axis sensors to estimate sensor orientation and motion. Usually those are gyroscopes, accelerometers and magnetometers. Low cost, small weight and dimensions of this devices makes them useful in a various fields of use, such as in aircrafts, unmanned aerial vehicles (UAVs), self-balancing robots, navigation devices or even smartphones.

2.1 Orientation representation

The IMUs are able to estimate orientation in reference to Earth‘s fixed frame. This data can be represented in various forms, most common and easy to understand is Euler angles (Pitch, Roll and Yaw) as depicted in Figure 2.1.

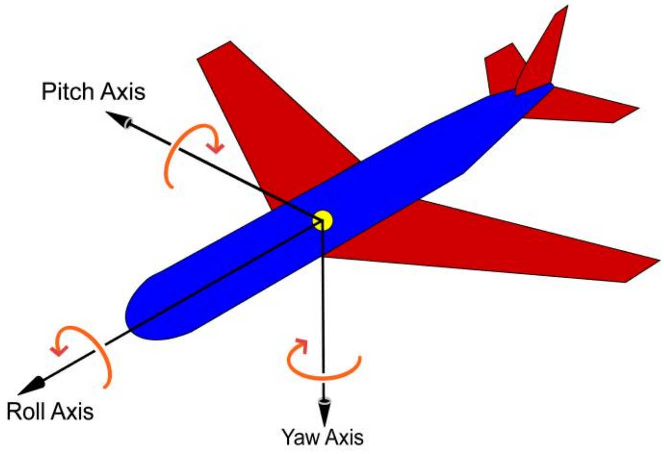

Figure 2.1 Euler angles orientation representation [3]

However, Euler angles suffer with gimbal lock problem which means losing one degree of freedom. For example, if plane from Figure 2.1 would be pointing up, changes in either Yaw or Roll axis would control the plane in same way.

That is why recently Quaternion-based orientation is starting to take place. It does not suffer from gimbal lock, it is easier to compose and implementations using this consume less computation power. They are however harder to understand from just simple look because they work on principle that any rotation in 3D space can be expressed as a combination of vector and scalar. Vector represents position and scalar orientation around vector.

2.2 Gyroscope

Digital gyroscope also known as angular velocity sensor is device that sense angular velocity, so rotation about its axis, which is usually expressed in degrees per second [4]. It has very precise readings, however, due to non-zero bias, white noise and following integration of those data, gyroscopes are subject of drifting which causes output to accumulate error in time.

Figure 2.2 illustrates gyroscope drift, represented as orientation data in degrees. For example, for Xsens MTw IMU gyroscopes it is approximately 2 degrees per minute but it is not constant and cannot be predetermined.

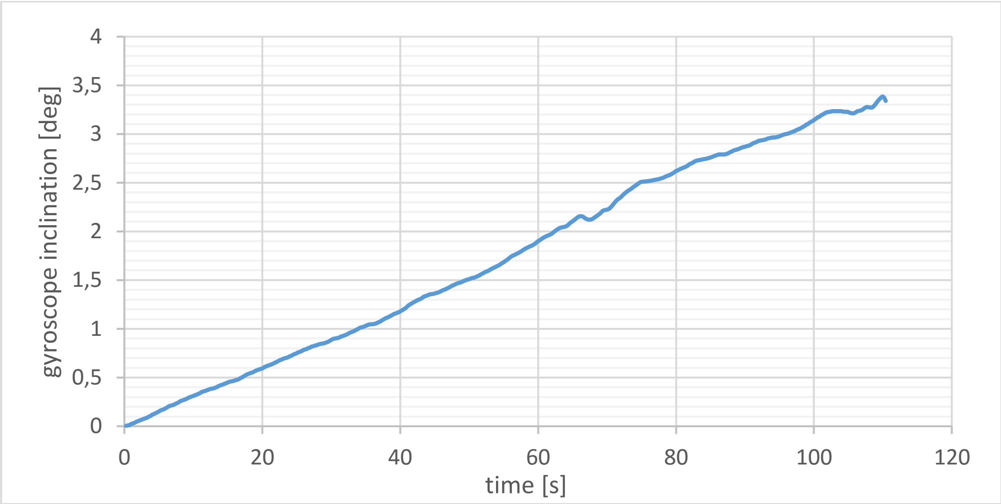

Figure 2.2 Gyroscope drift

2.3 Accelerometer

Accelerometers are electromechanical devices that measure acceleration, which is the rate of change of the velocity of an object, typically in meters per second squared [5]. They are sensing static forces as gravitation, and dynamic forces like movement or vibrations.

Generally, they internally contain fixed and spring attached capacitive plates so changes in capacity allows acceleration to be determined. Accelerometer-based on piezoelectric materials are also available. When tiny crystal is put under mechanical stress, e.g. force of mass from acceleration, it outputs electrical charge.

Spring attached plates or masses makes accelerometer readings subject to oscillation and as well sensitive to vibrations so while sensor is in motion, output could be noisy and less accurate [5].

Depicted in Figure 2.3 one can see comparison between slow (seconds 2 to 4) and fast (seconds 6 to 7) acceleration and deceleration of sensor. While in the first case the results are reasonably precise, in the second case there is a significant oscillation at the beginning and the end of the movement. Noise/vibration sensitivity can be seen as well.

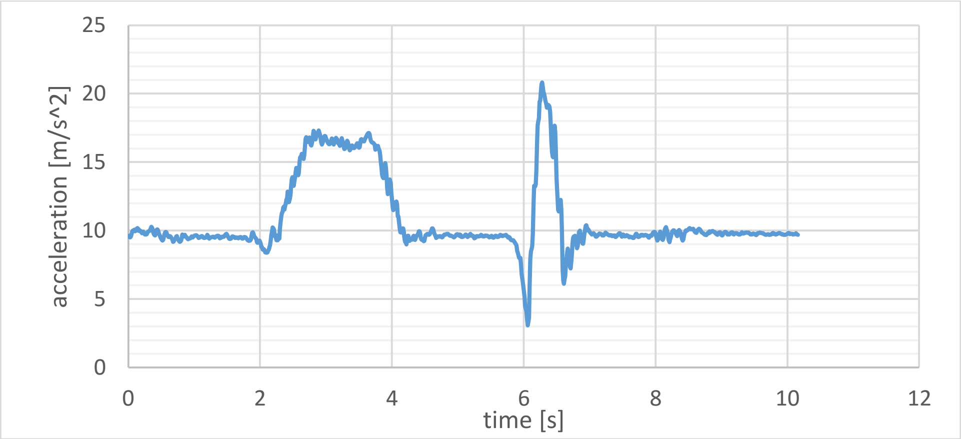

Figure 2.3 Accelerometer slow vs fast movement

2.4 Magnetometer

A magnetometer is a sensor which is used to measure the direction and strength of the magnetic field [6]. If Earth's magnetic field is the only magnetic field acting on the sensor, then it can be used as a compass to determine the direction in which the sensor is facing relative to the Earth's magnetic North pole. In terms of orientation estimation, it is precise device, however it can be easily interfered by ferromagnetic materials which makes it hard to use indoors or on moving subjects.

In Figure 2.4 Magnetic interferenceFigure 2.4 is possible to see the magnetic interference demonstrated by sensor passing near metal object. There is a significant peak 2 seconds after the beginning of the measurement.

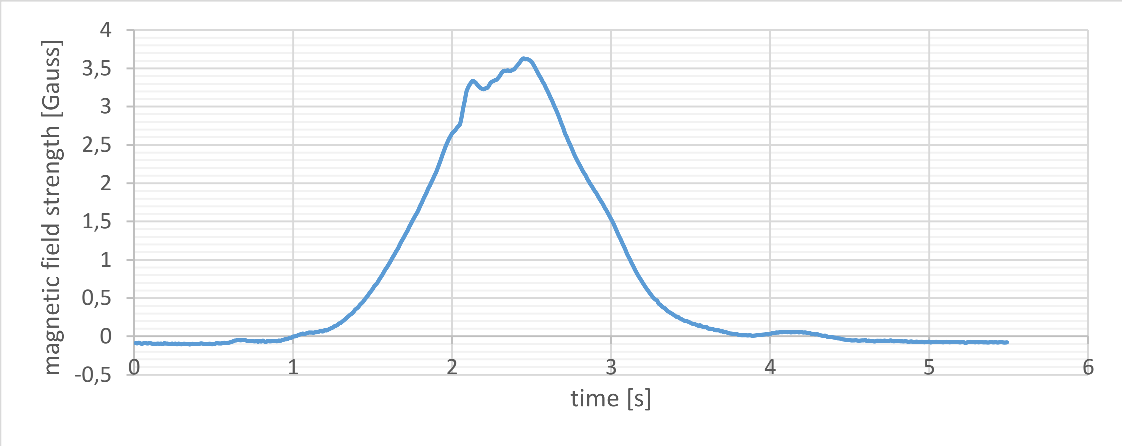

Figure 2.4 Magnetic interference

2.5 Fusion filters

First of all, sensors described earlier have different kind of problems making their use separately very limited. Also none of those sensors data could be used to estimate precise orientation data with reference to Earth's fixed frame. However, there are solutions to merge those data together and compensate for errors using fusion filter. Some of them are explained below.

2.5.1 Complementary filter

Complementary filter is a simple fusion mechanism merging different data using high-pass and low-pass filtering [7]. In terms of the IMU, accelerometer can give a good result while it is not moving, whereas gyroscope output could be precisely used in dynamic conditions like movement or rotation. So the idea behind complementary filter is to simply combine slow moving signals from accelerometer and fast moving signals from a gyroscope. This does not allow heading estimation so optionally magnetometer data can be fused as well [8].

2.5.2 Kalman filter

Kalman filter is an algorithm that consider previous measurements and predict next result which is then merged with current measured data. It can run in real-time, it is more precise but it is more complicated so it requires more computation power. Usually, computation is using accelerometer for inclination, gyroscope for fast rotations and magnetometers for heading estimation. In Euler representation, pitch and roll are estimated using accelerometers and gyroscopes data, while yaw is based on gyroscopes and magnetometers data [9].

2.5.3 Quaternion-based orientation filter by Sebastian Madgwick

Sebastian O.H. Madgwick in 2010 proposed filter which uses quaternion-based representation of orientation allowing accelerometer and magnetometer data to be used in an analytically derived and optimised gradient-descent algorithm [10] to compensate gyroscope. It is computationally inexpensive and effective even at low sampling rates as 10Hz.

2.6 Xsens IMU

Xsens Awinda MTw IMU are small, lightweight and precise wireless devices with simple possibilities of attachment to human body using Velcro straps. There is master docking station (Awinda Station) which allows to wirelessly communicate with up to 20 sensors simultaneously and charge up to 6 sensors at once [9]. Battery life of each sensor in continuous use is more than 5 hours. They contain 3-axis gyroscope, accelerometer and magnetometer but also pressure sensor. There is desktop application MT Manager for real-time monitoring of output, as well as possibilities to record and export data. Xsens also provide documented API for interfacing with sensors trough C, C++, C# or Matlab and also provide some basic examples.

In Figure **Chyba! Nenašiel sa žiaden zdroj odkazov.** Xsens MTw IMU is depicted with corresponding XYZ local frame in which internally all sensors are measuring.

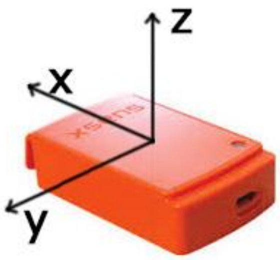

Figure 2.5 Xsens MTw IMU with local XYZ frame

2.6.1 Xsens Extended Kalman filter

Xsens IMU's already contains Extended Kalman filter implementation by Xsens [9]. Sensors are internally sampling at high frequency (1800 Hz) process this data and transmit them at lower frequency (20-120Hz) to output as orientation data, for example in form of Euler angles or Quaternions with respect to Earth fixed frame.

3 Knee angle measurement

There are different types of joints on the human body. Those are hinge, spheroidal, pivot, condyloid and saddle joints [11]. Knee is type of hinge joint (see Figure 3.1) which could be considered as hinge on the doors, so it allows only flexion/extension movement or in other words movement around axis of joint (*j*). Since the human knee is not perfect there could be small lateral or side to side movements but it is usually neglected.

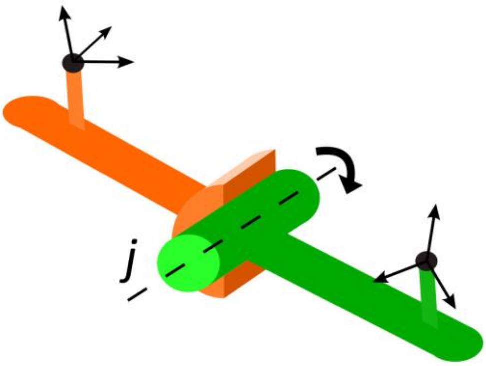

Figure 3.1 hinge joint [12]

3.1 Gait analysis

Gait in terms of humans is pattern of movement achieved by movement of limbs. For walking it is cycle described by series of steps shown in Figure 3.2.

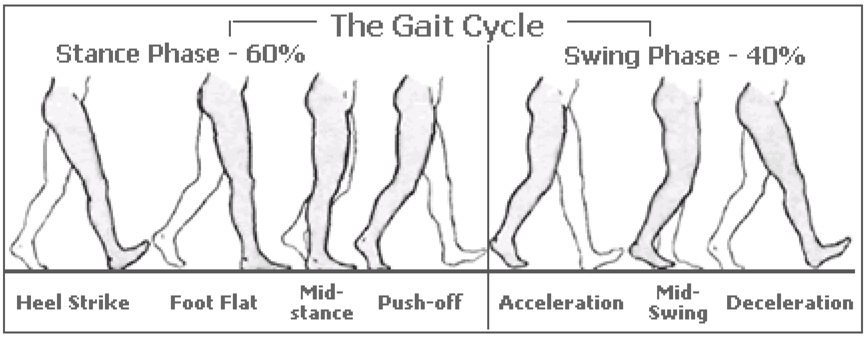

Figure 3.2 Gait cycle [13]

Gait analysis is process of monitoring performance of different tests like sit-to-stand test, stair climb test, six-minute walk test, etc. [14]. This is important to analyse mostly on people with knee problems or knee arthroplasty, since patterns in those movements could differ from those people who does not have any troubles [15], as depicted in Figure 3.3. As one can see there is significant difference between healthy patients (blue line) and pre-operation patients (red line). By performing rehabilitation after operation there is visible improvement. To be able to evaluate and compare these data, one of important parts to monitor is knee angle.

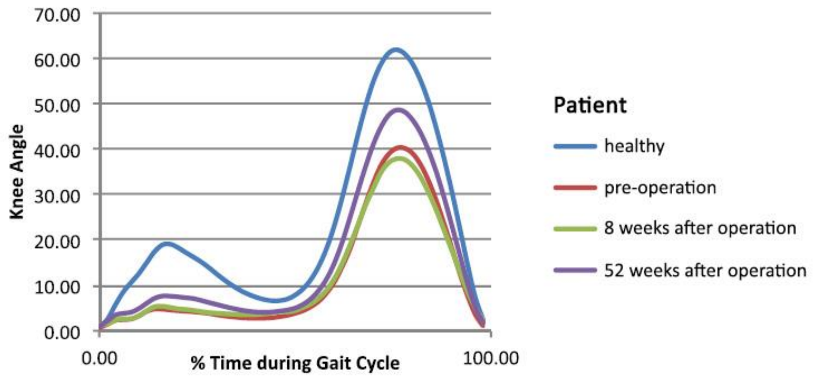

Figure 3.3 Knee angle gait cycle comparison [15]

3.2 Sensor placement

In Figure 3.4 the sensor placement to tight and shank is shown. This places contains small amount of fat or muscles and as well the skin movement is not significant during exercises or walking there, so compensation for any unexpected sensor deflection from its original position is not needed. Position for sensors facing upwards was chosen to avoid any possible calculation problems in future.

Figure 3.4 sensor placement

But still there is another problem and that sensors output is not completely corresponding with frame of bone (see Figure 3.5) so measuring angle α directly between sensors would distort results. Considering that some sort of calibration is required.

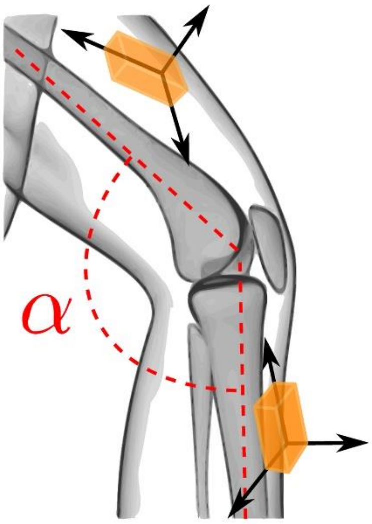

Figure 3.5 leg frame [12]

3.3 Methods

There are different solutions for sensor calibration and angle measurement using IMU. This subchapter will provide basic description of the four most suitable methods, as well as their advantages and disadvantages.

3.3.1 Simple quaternion-based estimation by YEI Technology

It uses quaternions output from sensors to calculate downward and forward vectors [11] and require standard T-pose (arms straight out of the side of the body and the palms facing forward) at start to calibrate. Angle is calculated from orientation data with respect to calibration.

3.3.2 Method proposed by Kun Liu

Analog inertial sensors are used to measure angular velocity and angular acceleration. It uses virtual sensors placed at centre of rotation of knee to calculate angle [16]. To be able to estimate virtual sensors position of joint needs to be known. Algorithm for virtual sensor based angle calculation was proposed. However as declared in [16], there were some sort of problems with precision.

3.3.3 Method proposed by J. Favre

This solution uses only accelerometer and gyroscope data and new method is proposed to determine each IMU’s orientation relative to common reference frame [17]. Then those orientation data are recalculated to align with tight and shank. To achieve this, it uses calibration procedure based on predefined standing pose and precise calibration movements like depicted in Figure 3.6A. Furthermore, angle is simply calculated from orientation data of tight and shank. Measurement is valid only for period of few minutes. After that, calibration procedure needs to be repeated.

3.3.4 Method proposed by T. Seel

Only accelerometers and gyroscopes are used, however no precise calibration motions are required, just random arbitrary motions, see Figure 3.6B. It does not align IMU’s coordinate systems it just identifies joint axis coordinates and joint position in local sensor coordinates [18]. It uses the fact that the knee joint behaves approximately like a mechanical hinge joint. The kinematic constraints that result from this fact are exploited to obtain the position vector and the direction vector of the knee flexion/extension axis in the local coordinates of both sensors.

Figure 3.6 calibration procedure using predefined (A) vs arbitrary (B) movements [18]

3.4 Goals

Goals of this work are implementation of algorithm to provide easy way to analyse knee angle measurement, especially for purposes of gait analysis. This incorporates choosing correct appropriate sensor position (which was already described), finding ways to gather, store and analyse data, choosing and implementing appropriate calibration method and finally calculating angle itself. Output should be displayed in form easy to understand, like charts. Last goal is to evaluate and compare results to some already available system.

4 Design and implementation of knee angle measurement

In this chapter we describe chosen sensors, analyse best suitable solution for calibration and angle measurement, specify detailed explanation of selected method and finally describe implementation. Implementation was done using Matlab, because it provides productive environment for mathematical computation, storing or reading large data in way which is easy to understand, for example in tables or charts.

4.1 IMU

In our work IMUs (Xsens Awinda MTw) and not just accelerometers are used because IMUs have several advantages over them and were already at our disposal. Xsens also provides documented API for interfacing their sensors or reading stored data in Matlab.

However, it was discovered that after measurements of orientation data inside of the school building Yaw is subject to drift from those Xsens units. Reason for that is magnetic interference since Yaw is strongly depended on magnetometer readings hence in combination with drifting gyroscope it makes the output inaccurate. As experimentally tested, using outdoors problem disappeared but that is not universal nor practical solution. Without magnetometer there is no heading reference hence it is not possible to estimate unit orientation relative to an Earth’s fixed frame defined by the gravity and magnetic field.

4.2 Method comparison

The method with simple quaternion-based estimation incorporates magnetometer as described in previous subchapter so this does not allow easy usage indoors. Methods proposed by Kun Liu, J. Favre and T. Seel avoid this. One proposed by J. Favre [17] uses recommendations from International Society of Biomechanics (ISB) and bone-embedded anatomical frames (BAF) which is higher standard in physiotherapeutic industry. It is also easier to determine angles on different type of joints, like spheroidal, pivot or condyloid. However, precision of measurement relies on accurate calibration procedure which could be on some patients after injuries difficult or even impossible. In contradiction, T. Seel’s method require only arbitrary movements for some period of time which is much easier for those people. This way it is able to estimate joint axis orientation and position. In solution by Kun Liu distance of joint axis from sensors need to be measured. This is naturally biggest source of error there.

Based on this, method proposed by Thomas Seel was chosen for this thesis. It does not incorporate magnetometer, it does not require precise calibration and it is not limited for runtime length. From calibration process it determines joint axis orientation and position but in local coordinate systems of each sensor respectively. Those are then used in calculations of slowly drifting angle from gyroscope and noisy angle from accelerometer. Angles are then merged using fusion filter resulting in stable output.

4.3 Data gathering

For recording and storing data application from Xsens MT Manager (see Figure 4.1). Using provided graphical user interface (GUI), it is easy to select connected sensors, monitor their battery status and functionality but mainly it allows start of recording with simple click of the button. After measurement, data are stored to **.mtb* file in desired location. Always at least 2 measurements are required. First one for calibration and others as actual trials.

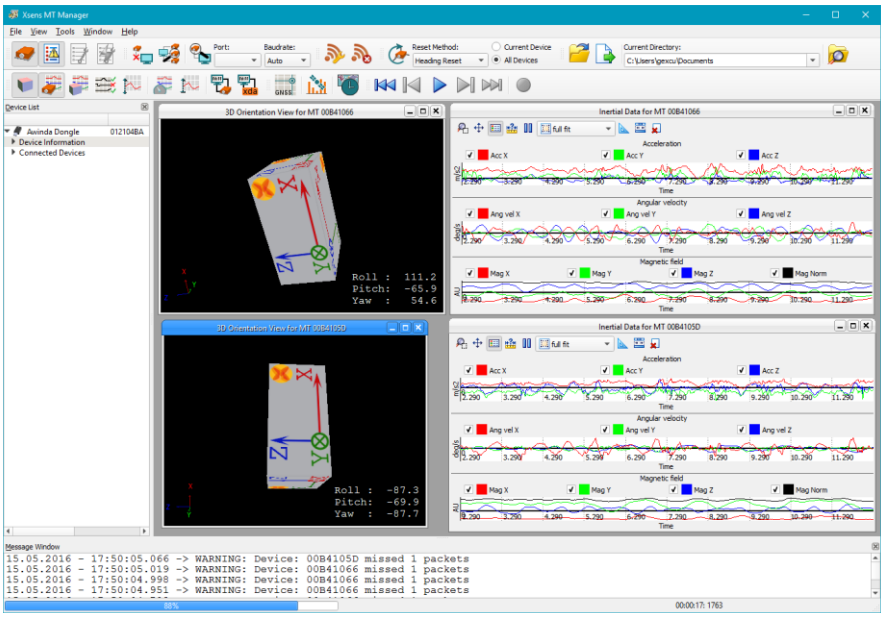

Figure 4.1 Xsens MT Manager

4.4 Data analysis

For importing recorded data to Matlab, two possible ways are introduced and implemented in form of the scripts `importXsensData` and `importXsensMTB`. Output from both functions are compatible.

The first one requires a *.*txt* file for each sensor in specific format which could be exported from *.*mtb* using MT Manager. Many files can by scanned at the same time using just one call of this function and its output is array of structs, where every structure represents one sensor. Each of them contains the frequency at which data were recorded, accelerometer data stored in table of three columns and gyroscope data stored same way as well. However, because it uses just simple `textscan` method, predefined formatting needs to be respected at export from MT Manager.

More convenient method is using function `importXsensMTB`. Using Xsens API it can directly read *.*mtb* file and store data from all sensors to array of structs. Downside of this method is that it requires Xsens libraries. Since we are using wireless Xsens units, first wireless master device is found and then it is looped trough all child devices and inertial data are stored. Sometimes it happens that sensors report incorrect value like Not a Number (NaN) or infinity (inf) which is incorrect. Simple compensation is made just by setting that value to zero. It can cause inaccuracies but since measurements are done in high frequencies (from 40Hz) and since this occur rarely, it makes actually no difference in result.

4.5 Joint axis coordinates and position estimation

In the inertial measurement data from motion of the subject are hidden key information allowing joint axis and position estimation. It is because angular rates from gyroscopes and accelerations from accelerometers must fulfil kinematic constrains. Those incorporate joint axis vector which is captured in local sensor coordinates. Joint axis orientation and position according to sensors is constant as depicted in Figure 4.2.

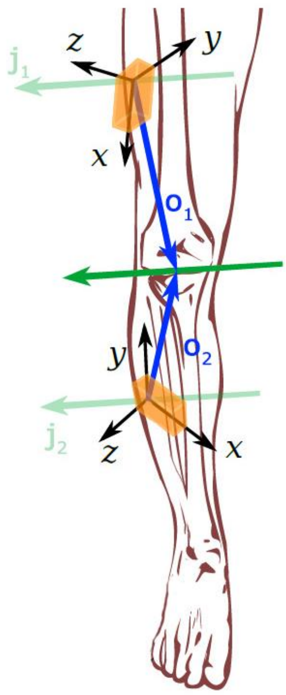

Figure 4.2 joint axis vectors [19]

4.5.1 Exploiting kinematic constraints of hinge joint

Now considering that tight and shank segments are free to move and rotate in any desired direction but are connected by a hinge joint. Each of this segments have IMU attached on location described earlier. This location can be random, however muscle and skin effect on different places could cause more inaccuracies.

Let's mark joint axis vector with respect to the local coordinate system of gyroscope on first segment j_1 and angular velocity of this gyroscope $g_1(t)$. Vice versa j_2, for vector coordinates in second segment gyroscope local coordinates and $g_2(t)$ for angular velocity of this gyroscope. Measurement is hold at some sample period Δt.

See Figure 4.2 for joint axis vector (green) and j_1, j_2 vectors. In other words, j_1 and j_2 are same vectors representing joint axis vector but both are expressed in different coordinates - coordinates of belonging IMU.

Because of the connection within segments of knee it is just matter of geometry that $g_1(t)$ and $g_2(t)$ are changing simultaneously with difference only in joint angle velocity and rotation matrix. For example, if person rotates to the left, both gyroscopes captures angular velocity caused by rotation which is same for both segments. However, values of this angular velocities are not equal, because each sensor is not perfectly aligned with the bone and within each other. If person's leg is bending during this movement as well, the velocity caused by joint angle change also create angular velocity change in gyroscope.

Now as explained in [12], projections of $g_1(t)$ and $g_2(t)$ into the joint plane (plane to which joint axis is normal vector) in each moment of time regardless sensor attachment, have same lengths. This leads to Equation 4.1 where $\| \ \|_2$ is Euclidian norm.

$$\|g_1(t) \times j_1\|_2 - \|g_2(t) \times j_2\|_2 = 0 \qquad \forall t$$

Equation 4.1

Equation has solution for every measured data, therefore we can identify j_1 and j_2 vectors in a least squares sense. We define sum of squared errors in Equation 4.2.

$$\psi(\phi_1, \phi_2, \theta_1, \theta_2) := \sum_{i=1}^{N} e_i^2, \qquad e_i = \|g_1(t_i) \times j_1\|_2 - \|g_2(t_i) \times j_2\|_2$$

Equation 4.2

And express j_1 and j_2 accordingly like described in [18], with Equation 4.3.

$$j_1 = (\cos(\phi_1)\ cos(\theta_1), \cos(\phi_1)\ sin(\theta_1), \sin(\phi_1))^T$$
$$j_2 = (\cos(\phi_2)\ cos(\theta_2), \cos(\phi_2)\ sin(\theta_2), \sin(\phi_2))^T$$

Equation 4.3

Since signs of gyroscopes output depend on sensor orientation it is necessary to match signs of j_1 and j_2 as well, according to orientation. This is necessary since both vectors need to point in the same direction, like in Figure 4.2. It can be derived from measured data itself as explained in [12] or [19] but also by simple observation in case sensors are pointing to the same direction.

Let's analyse another constrain according to [12] and [18]. Each sensor acceleration can be considered as sum of joint centre's acceleration and the acceleration due to the rotation of this sensor around the joint centre. Then naturally, the acceleration of the joint centre is same in both local frames. So let us mark accelerometer readings with $a_1(t)$ and $a_2(t)$, and o_1, o_2 vectors from the joint centre to the origin of the sensor frame for each segment respectively, just as before. Those vectors are constant and do not change during measurement. Mathematically it claims to Equation 4.4.

$$\left\|a_1(t) - \Gamma_{g_1(t)}(o_1)\right\|_2 - \left\|a_2(t) - \Gamma_{g_2(t)}(o_2)\right\|_2 = 0\ \forall t$$
$$\Gamma_{g_i(t)}(o_i) = g_i(t) \times (g_i(t) \times o_i) + \dot{g}_\iota(t) \times o_\iota, \qquad i = 1, 2$$

Equation 4.4

Where $\Gamma_{g_i(t)}(o_i)$ is the radial and tangential acceleration by cause of rotation around the joint centre and $\dot{g}_\iota(t)$ is time derivate of angular rate or gyroscope calculated via the third order approximation.

This can be solved in least squares sense as well since it is also valid for every measured dataset. For this we define sum of squared errors in Equation 4.5

$$\tilde{\psi}(o_1, o_2) \coloneqq \sum_{i=1}^{N} e_i^2\,, \qquad e_i = \left\|a_1(t) - \Gamma_{g_1(t)}(o_1)\right\|_2 - \left\|a_2(t) - \Gamma_{g_2(t)}(o_2)\right\|_2$$

Equation 4.5

Using optimization method we minimize it, however as the result we get $\hat{o}_1$ and $\hat{o}_2$ which refers to a random point along the joint axis. Therefore, we shift it to local frames using Equation 4.6 as explained in [18].

$$o_1 = \hat{o}_1 - j_1 \frac{\hat{o}_1.j_1 + \hat{o}_2.j_2}{2}$$
$$o_2 = \hat{o}_2 - j_2 \frac{\hat{o}_1.j_1 + \hat{o}_2.j_2}{2}$$

Equation 4.6

4.5.2 Implementation

Based on constrains and equations above it is possible to calculate vectors for joint axis coordinates and joint axis position using considerably small amount of data. There is no specific calibration process but calibration itself is needed anyway because motion of segments should be rich enough for angular rates and accelerations in any direction which could not happen if only walk trial would be performed for example.

This was implemented into the script `computeSeel` which require as input array of sensors. Only first 2 sensors are used in case more of them were connected. In calculations time derivation of angular velocity is required which incorporates function `calculateDerivativeGyroscope`. This is done by third order approximation, according to Equation 4.7.

$$\dot{g}_i(t) \approx \frac{g_i(t-2\Delta t) - 8g_i(t-\Delta t) + 8g_i(t+\Delta t) - g_i(t+2\Delta t)}{12\Delta t}, \qquad i = 1, 2$$

Equation 4.7

Naturally, this require to look 2 time steps forward and 2 backward so after calculation this data samples are stripped away from gyroscope and accelerometer. It makes no difference since lowest sampling frequency is 40Hz and even on this it is still fraction of second and calibration takes naturally more time. Thanks to MATLAB’s data representation and handling it is not necessary to compute this in loop for every instance of time since it can be done at once for whole table of data. As experimentally tested this dramatically reduce runtime.

Using technique of anonymous function and build-in function `lsqnonlin` for least squares solving Equation 4.2 and Equation 4.3 are estimated which results in vectors j_1, j_2. As the initial value zero vector is used. Since IMU’s point to the same direction but have opposite rotation in vertical axis, because of their attachment chosen earlier (as depicted in Figure 3.4A), vectors j_1, j_2 needs to be corrected for signs. It was experimentally found that it can be done by just simply multiplying j_2 with value -1.

Then o_1, o_2 are estimated by Equation 4.4, Equation 4.5 and Equation 4.6 and using anonymous function and MATLAB's `lsqnonlin`, with zero vector as initial value. At this point we successfully have joint axis coordinates and joint axis position which is returned as struct of vectors from this script.

4.6 Calculation of the Angle

Vectors identified before are used to calculate angle of knee flexion/extension. Slowly drifting gyroscope-based estimation of angle is merged using fusion filter with accelerometer-based angle estimation.

4.6.1 Gyroscope-based angle calculation

Calculation of angle using gyroscope is pretty easy just by doing integration of difference of angular velocities around joint axis over time, see Equation 4.8.

$$\alpha_{gyr}(t) = \int_0^t (g_1(\tau).j_1 - g_2(\tau).j_2)d\tau$$

Equation 4.8

Angle α_{gyr} is however subject to drift.

4.6.2 Accelerometer-based angle calculation

For extracting angle from acceleration data it is necessary to do some steps first. As introduced in [16] and extended in [18], it is needed to shift accelerations $a_1(t)$ and $a_2(t)$ onto the joint axis by applying Equation 4.9

$$\tilde{a}_i(t) = a_i(t) - \Gamma_{g_i(t)}(o_i), \qquad i = 1, 2$$

Equation 4.9

Those shifted accelerations are the same in terms of quantity with difference in rotation within each other around joint axis which allows angle measurement. But since they are in different local frames it is necessary to project them to one plane. For this, joint plane is defined in Equation 4.10 for each local coordinate system using coordinates x_i, y_i.

$$x_i = j_i \times c, \qquad x_i = j_i \times x_i, \qquad c \nparallel j_i, \qquad i = 1, 2$$

Equation 4.10

Where c could be random vector not parallel with joint axis. Now, we can make calculation of joint angle from accelerometer readings. Since it is projected to joint plane it is just 2D angle between 2 vectors. This is expressed by Equation 4.11.

$$\alpha_{acc}(t) = \left(\begin{bmatrix} \tilde{a}_1(t).x_1 \\ \tilde{a}_1(t).y_1 \end{bmatrix}, \begin{bmatrix} \tilde{a}_2(t).x_2 \\ \tilde{a}_2(t).y_2 \end{bmatrix}\right)$$

Equation 4.11

In case shifted accelerations are almost collinear with joint axis, significant errors could occur but in practice this could only happen if knee axis is in vertical position which means that patient would have to be lying on side. That is not real situation for measurement.

4.6.3 Fusion

We are already able to calculate angles but since gyroscope-based calculation is subject to drift and accelerometer-based calculation is very sensitive to fast movements, noise or vibrations it is advantageous to combine them using sensor fusion. Diagram of this used method can be seen in Figure 4.3.

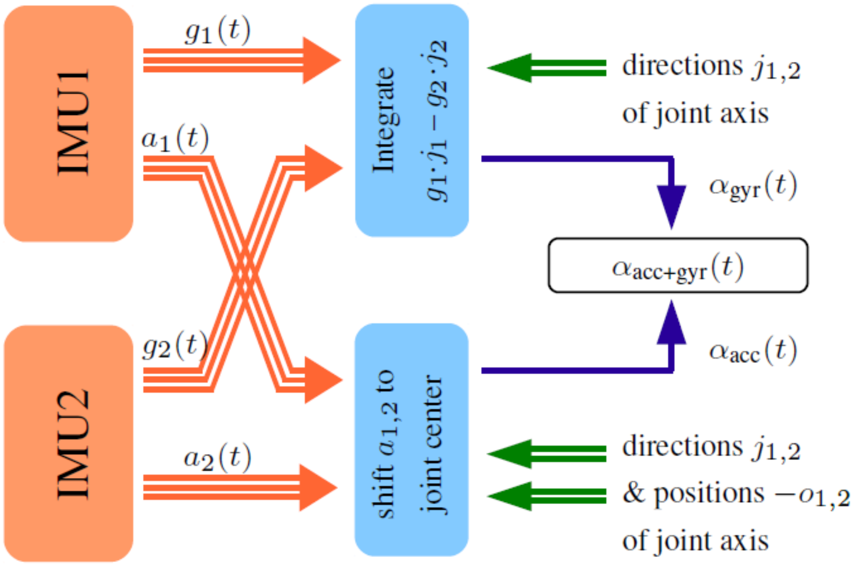

Figure 4.3 Diagram of method proposed by Thomas Seel [18]

It can be done by any fusion method like complementary filter, Kalman filter or other. For angle fusion in this work complementary filter is used because it is easy to implement and as it will be shown in Chapter 5, results are good enough. Complementary filter can be expressed by Equation 4.12 [18].

$$\alpha(t) = \lambda\, \alpha_{acc}(t) + (1-\lambda)\left(\alpha(t-\Delta t) + \alpha_{gyr}(t) - \alpha_{gyr}(t-\Delta t)\right), \quad \lambda \epsilon\ (0;1)$$

Equation 4.12

With adjusting of parameter λ there could be changed weight between accelerometer reading and old angle measurement plus gyroscope angle difference in one step of time Δt.

4.6.4 Implementation

Implementation of actual angle calculation is made in script `calcJointAngles` with two input parameters. First one is struct of vectors, so output from `computeSeel` script. Second parameter is struct of measured data like walk cycle, sit-stand procedure, or similar. At first derivates are calculated by `calculateDerivativeGyroscope` explained before, since they are needed in accelerometer-based calculations. Then, using built-in matlab function `cumtrapz` integration of angular velocities difference is evaluated over time, like explained in 4.6.1. This results directly in gyroscope-based angle for each sample in time and those data are stored in array. Since calculation is done by integration, measurement angle from gyroscopes always start at 0 degrees. This require measurement to start with straightened leg.

Accelerometer data are then shifted and joint plane is calculated as suggested in 4.6.2. As a *c* vector, unit vector [1 0 0] was chosen since it is impossible for it to be parallel with joint axis. This could happen only if sensor would be placed to face its X axis with joint axis, which is unlikely and in our chosen mounting position it cannot happen, because their construction does not allow that (Velcro strap on the bottom of sensor) and they are pointing upwards. Equation 4.11 for angle calculation is implemented according to Equation 4.13.

$$\alpha_{acc}(t) = \operatorname{acos}\left(\frac{A.B}{\|A\|\|B\|}\right)$$

Equation 4.13

For noise correction data from accelerometer are shifted to 0 degrees so start point match with gyroscope.

Now both angles need to be fused. For this, complementary filter from Equation 4.12 was implemented and parameter λ was experimentally evaluated to 0.01, as can be seen later in Chapter 5. This means there is 99% weight for gyroscope reading and 1% for accelerometer reading but it is enough to eliminate drift even in long term operations (few minutes) as will be demonstrated in Chapter 5. Also there is no accelerometer noise or any spikes.

Results are plotted using built-in function since it is easy to understand and there are tools for zooming, moving, or reading values from plots already.

4.7 Usage simplification

In order to speed process of work with those scripts, one main script `main` was created, which takes two parameters, calibration file and data file, both in **.mtb* file. As output, it returns struct of joint axis vectors – result from calibration. For this reason, calibration file can be exchanged by this struct, so in case of multiple measurements on one patient, calibration does not have to be always computed all over again, since it takes some time (depends on calibration length). Then all necessary scripts are started and result is plotted.

5 Experimental results

Different experiments were made for example to compare angle results from gyroscope, accelerometer and fused angle, to find best lambda value for complementary filter, or comparison between optotrack system. For all measurements wireless Xsens MTw inertial measurement units with recording frequency 100Hz were used.

First tests were made with IMU attached by tape to arms of plastic goniometer as depicted in Figure 5.1.

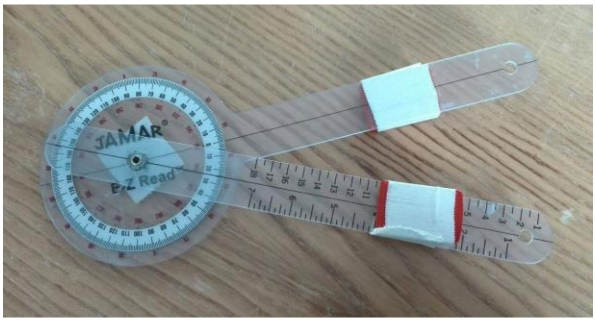

Figure 5.1 plastic goniometer with IMUs attached

We measured series of bending to different angle going from 0 to 110 degrees, from 0 to 90 degrees, etc., up to 10 degrees and back to zero. Results displaying measured angles but also comparison between gyroscope, accelerometer and fused angle are shown in Figure 5.2.

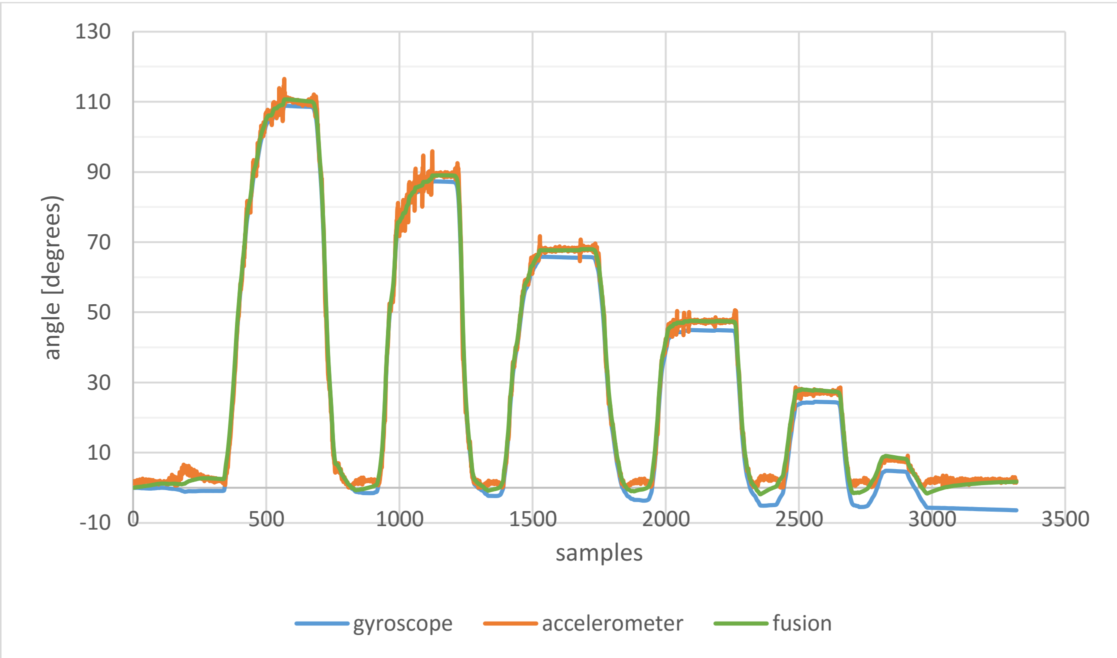

Figure 5.2 Comparison between accelerometer-based, gyroscope-based and fused angle measurement

This clearly shows drifting gyroscope (blue line), especially at end of test. Accelerometer noise (orange line) is also visible, mostly when changes in angle were made. Angle fused by complementary filter (green line) shows no affect to drift and small sensitivity to accelerometer inaccuracies. Since measurement was made only with goniometer moved by hands there is no guarantee of setting it completely correctly to desired angle. But anyway, if we look at the results for fused data the biggest error is around 2 degrees.

As stated in 4.6.4, fused result is depended on parameter λ. In Figure 5.3 value 0.01 was compared with values 0.2 and 0.001 to show the difference. Comparison was made on same dataset as previous one.

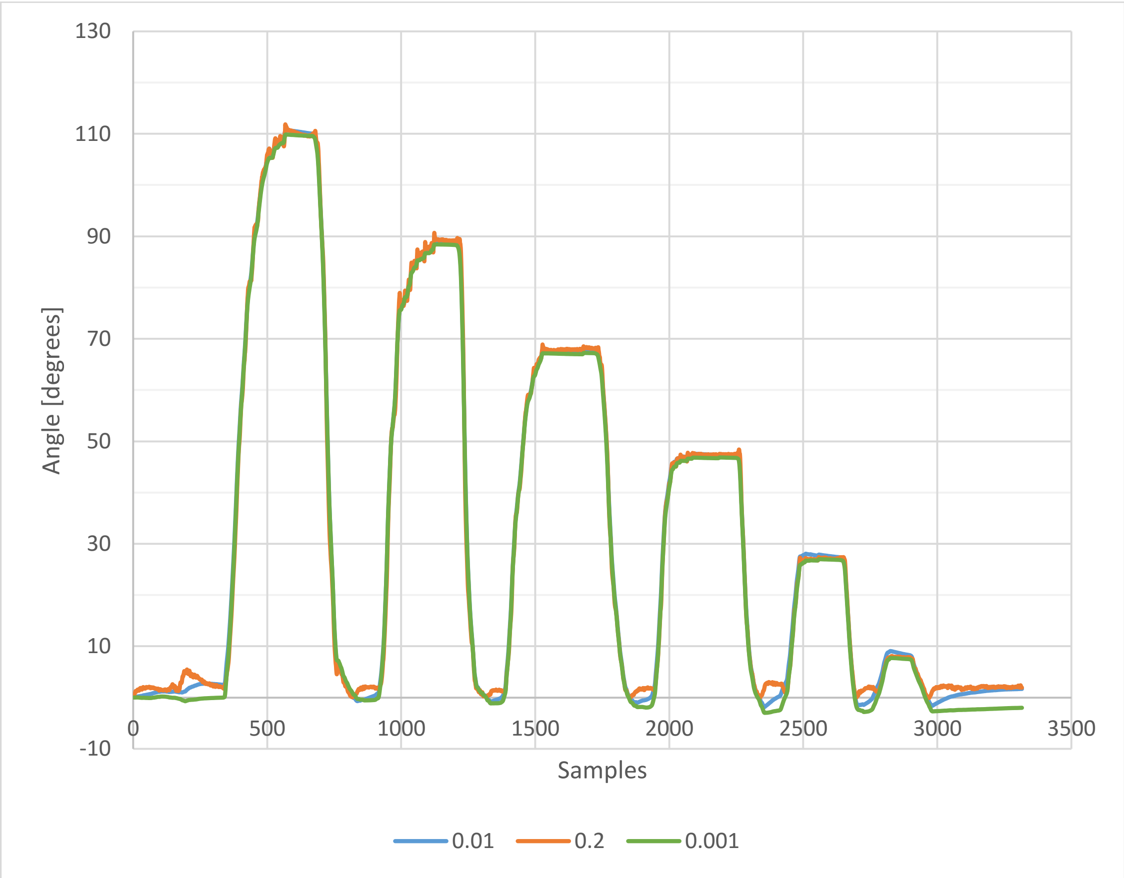

Figure 5.3 comparison between different settings for complementary filter

As expected, value 0.2 for λ is more sensitive to accelerometer noise since it put less weight for gyroscope data and on the other side value 0.001 is not dealing with drift enough.

Another test for checking reliability of complementary filter was made to see if it is affected by drift in long term operations. This is illustrated in Figure 5.4. Measurement started at 0, then it was bended to around 55 degrees and left for more than 6 minutes.

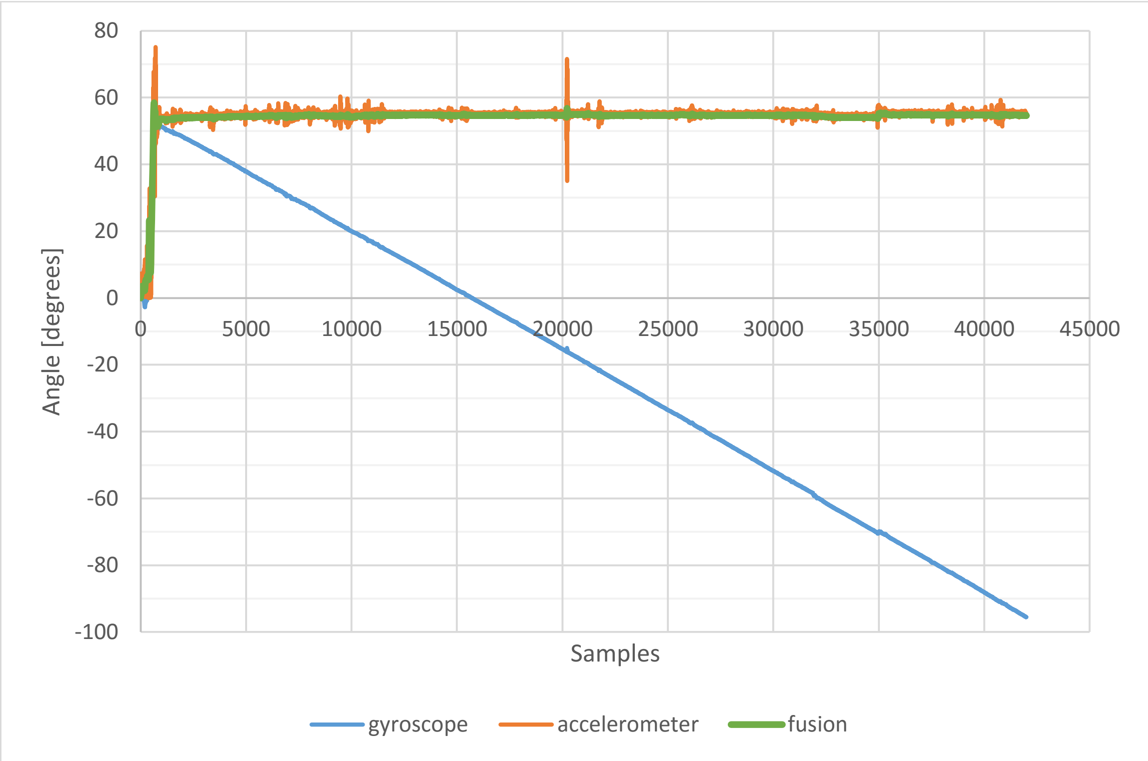

Figure 5.4 Long time measurement

As one can see drift of gyroscope is significant but there is no effect on fused data whatsoever, so this proves λ to be at best possible value as well.

Comparison between Motive Optitrack measurement system for simple knee bend was done. For Optitrack system only 6 cameras were used and only lower body part reconstruction was done from simple 16 reflective markers, see Figure 5.5. Result for simple knee bend is shown in Figure 5.6. Blue line is IMU-based measurement, red line is Optitrack result.

Figure 5.5 Optitrack reflective markers placement

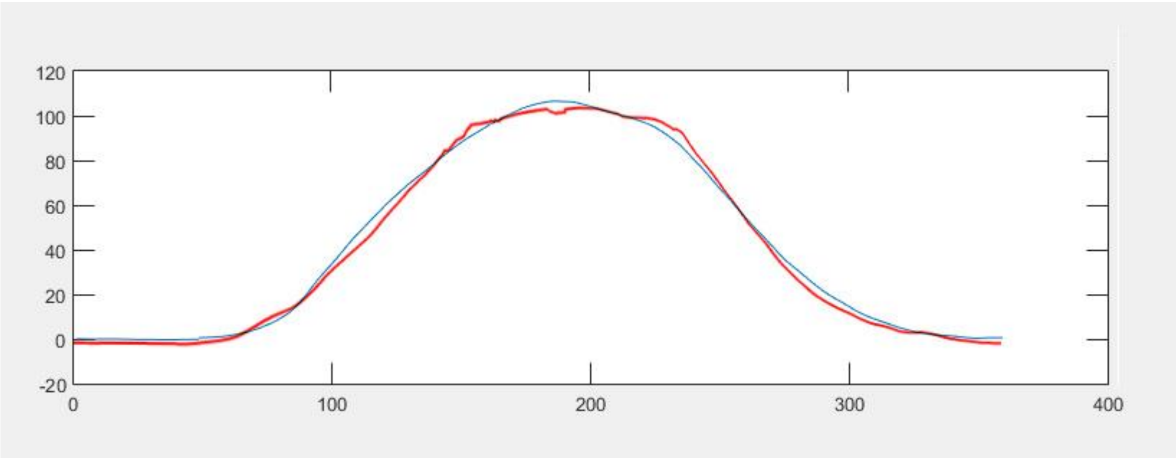

Figure 5.6 IMUs vs Optitrack angle measurement comparison

Since Optitrack measurement was limited it results in noisy output. However, comparison shows that results are still very similar, actually IMU-based measurement is more smooth and looks realistic.

Finally, gait analysis was done and result of four walk cycles is shown in Figure 5.7.

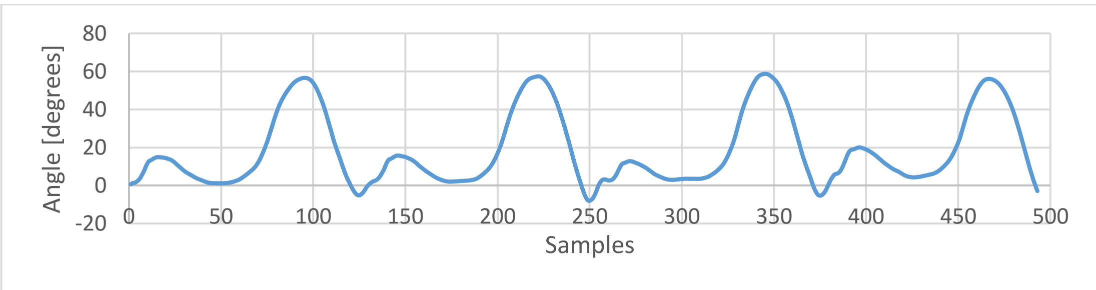

Figure 5.7 Gait analysis

5.1 Problems and weaknesses

After some experiments, some weaknesses of this method came up as well. First one happen, when there is insufficient calibration. Even if it does not require any precise movements or postures, data should be rich enough for movement in different axis. This could make problem if the calibration is too short, or if the calibration didn't contained enough movement. As depicted in Figure 5.8, both gyroscope-based (blue line) and accelerometer-based (orange line) angle estimations are incorrect, which leads to incorrect result (green line) compared to correct one (yellow line) with correct calibration.

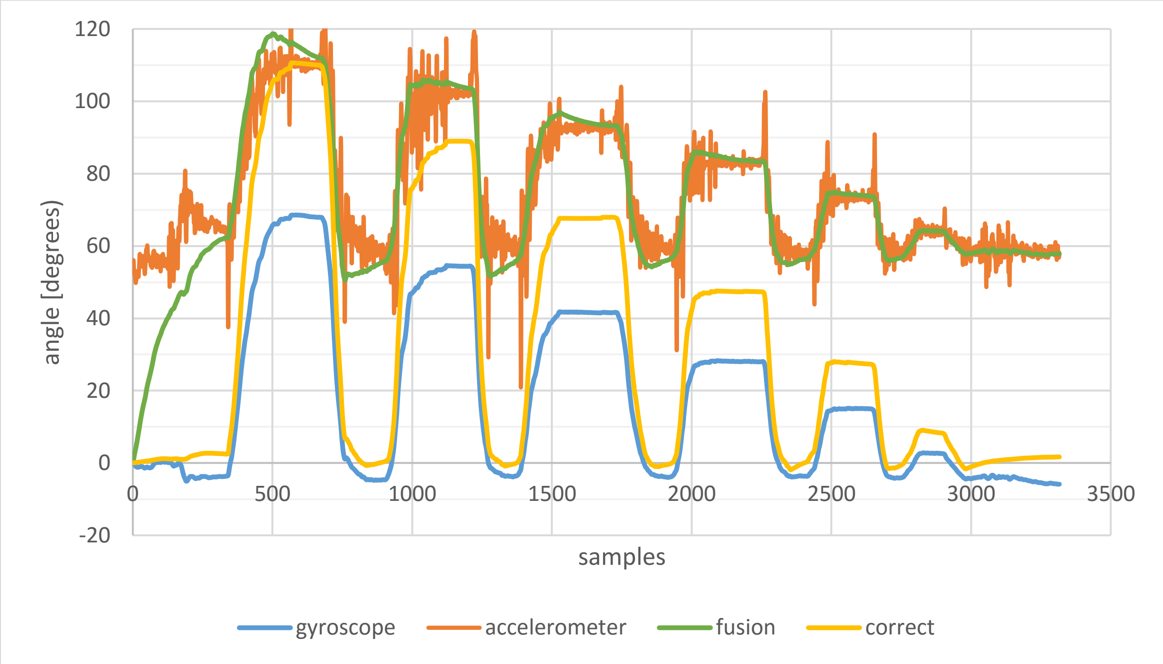

Figure 5.8 Insufficient calibration leads to wrong results

In case sensors are attached differently as supposed, for example upside down, it will lead to wrong estimation of joint axis vectors j_1 and j_2 which could make partial results of angle calculation upside down, so fusion will not work correctly, as shown in Figure 5.9. However, when vectors j_1 and j_2 are not matching, partial results could be affected more than by just simple inversion.

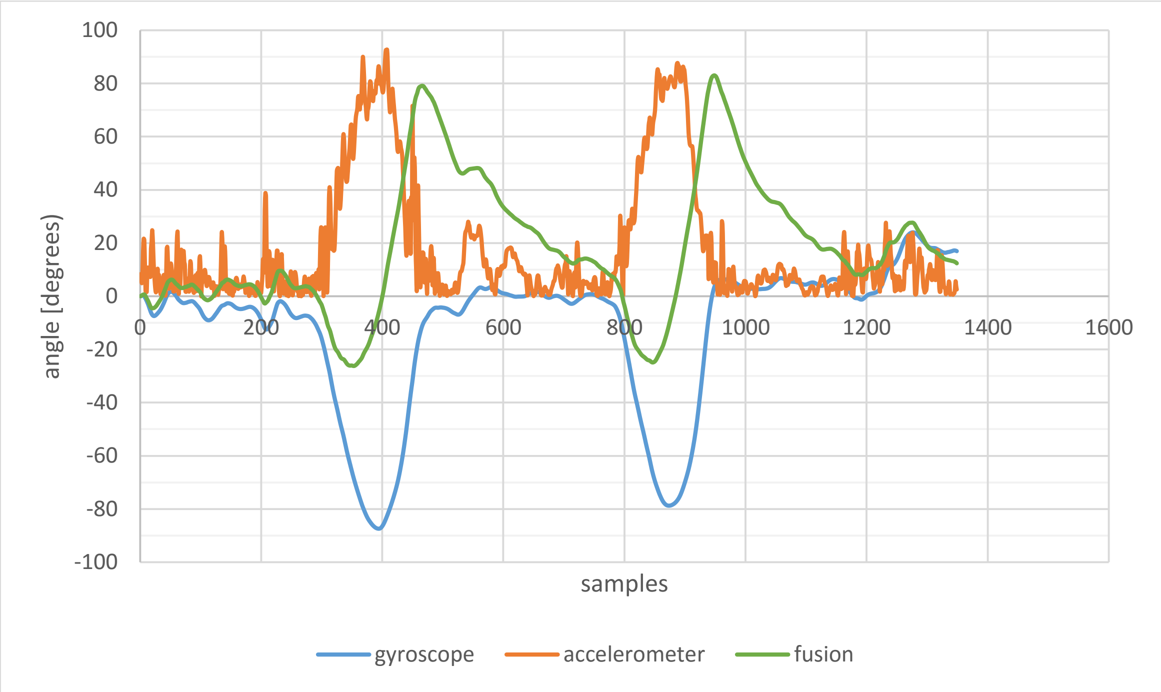

Figure 5.9 Incorrect sensor mounting

Another weakness is that since gyroscope-based angle calculation always starts at zero, leg needs to be straightened. In case leg is bended, result of calculated angle will not be correct.

6 Conclusions and future directions

Goal of this bachelor's thesis was to explore and propose solution for human motion analysis using accelerometers. This was extended to measurement using Inertial Measurement Units which are cheap and allow easy usage. Focus was put on knee angle measurement to allow gait analysis. There are already solutions for this available, however they require special laboratories, complicated setup or are too expensive.

After investigation of different problems and solutions in using IMUs for measurement appropriate method for calibration and measurement was chosen and implemented in MATLAB and various experiments were done. They show that problems caused by drifting gyroscope, accelerometer noise or magnetometer interference were successfully eliminated. There is no drift even after 6 minutes of runtime, there is no magnetic interference and outputs are smooth and does not contain unwanted peaks from accelerometer. Gait analysis shows promising results but this still require more tests to compare with already available systems in order to analyse precision.

During experimentation new problems and future improvements were discovered. Biggest restriction is that measurement needs to start with straightened leg which could be a problem at some patients so further investigation in start point estimation is necessary. Mounting of sensors needs to be done to chosen places with heading up. While mounting restriction is proficient in order to eliminate skin or muscle movement, heading of sensors does not affect measurement and as explained in [18] this can be estimated from calibration data itself. There is also possibility to extend measurement for real time use.

Literature

1. **J. Van Bogart.** *Motion analysis technologies.*: IEEE Conf. on Pediatric Gait, 2000. pp. 166–172.

2. **McClelland JA, Webster KE and Feller JA.** *Gait analysis of patients following total knee replacement: a systematic review.*: The Knee. Volume 14, Issue 4, August 2007, pp 253–263.

3. **Auawise.** *An image showing all three axises.* [online, available at: 2016], https://commons.wikimedia.org/wiki/File:Yaw_Axis_Corrected.svg.

4. **EPSON.** Gyro sensors - How they work and what's ahead. [Online, available at: 2016] http://www5.epsondevice.com/en/information/technical_info/gyro/.

5. **Sparkfun.** Accelerometer Basics. *sparkfun.* [Online, available at: 2016] https://learn.sparkfun.com/tutorials/accelerometer-basics.

6. **Preeti Jain.** Magnetometers. *Engineers Garage.* [Online, available at: 2016] http://www.engineersgarage.com/articles/magnetometer.

7. **Alessandro Giacomel.** Kalman filter vs Complementary filter. *Robottini | Little robots with Arduino.* [Online, available at: 2016] http://robottini.altervista.org/kalman-filter-vs-complementary-filter.

8. **Harsh Vathsangam.** My IMU estimation experience. [Online, available at: 2016] https://sites.google.com/site/myimuestimationexperience/.

9. **Xsens.** MTw User Manual. [Online] November 2013. https://www.xsens.com/download/usermanual/MTw_usermanual.pdf.

10. **Sebastian O.H. Madgwick, Andrew J. L. Harrison, Ravi Vaidyanathan.** *Estimation of IMU and MARG orientation using a gradient descent algorithm.* Rehab Week Zurich, ETH Zurich Science City, Switzerland : 2011 IEEE International Conference on Rehabilitation Robotics, June 29 - July 1, 2014.

11. **Yost Engineering Inc.** *Calculating Angles Between Two YEI 3-Space Sensor™ Devices on a Human Body. AN2013.01* USA, 2007.

12. **T. Seel, T. Schauer and J. Raisch** *Joint axis and position estimation from inertial measurement data by exploiting kinematic constraints.* Dubrovnik, Croatia : In Proceedings of the IEEE International Conference on Control Applications, 2012. pp. 45–49.

13. **Newcastle University and Northumbria University.** Gait and Motor Milestones. *Paediatric Musculoskeletal Matters.* [Online, available at: 2016] http://www.pmmonline.org/page.aspx?id=753.

14. **Kim Bennell, Fiona Dobson and Rana Hinman.** *Measures of Physical Performance Assessments.*: Arthritis Care & Research, 2011. Vol. 63, No. S11, pp S350–S370.

15. **Jeeshan Rahman.** *Gait assessment as a functional outcome measure in total knee arthroplasty: a cross-sectional study.*: BMC Musculoskeletal Disorders, 2015.

16. **K. Liu.** *Ambulatory measurement and analysis of the lower limb 3D.* Changchun, China : Proceedings of International Conference on Mechatronics, 2009. pp. 3065–3069.

17. **J. Favre, R.Aissaoui, B. M. Jolles, J. A. deGuise, K. Aminian.** *Functional calibration procedure for 3D knee joint angle description using inertial sensors.*: Journal of Biomechanics, 42, 2330–2335, 2009.

18. **T. Seel, T. Schauer, and J. Raisch.** *IMU-Based Joint Angle Measurement for Gait Analysis.*: Sensors, 14 (4), 6891–6909, 2014.

19. **T. Seel.** Inertial Sensor-Based Gait Analysis. *Technical University Berlin.* [Online, available at: 2016] 2014. http://www.control.tu-berlin.de/IMU-Based_Gait_Analysis.

Appendix A

Manual

This application runs in Matlab and it contains multiple files with scripts. For measurement MT Manager application is required.

In case data are already stored, they can be read from **.mtb* file, Xsens MT SDK needs to be installed in system. This can be done by downloading and installing MT Software Suite (https://www.xsens.com/mt-software-suite/). Then by running script `main(calibFile, dataFile)` where *calibFile* is file for calibration and *dataFile* is file containing measured data, everything is computed automatically and results are plotted. In case multiple measurements on one patient were done, output from *main* (calibration vectors) can be stored to Matlab variable and used instead of *calibFile* in order to avoid calibration vectors recalculation. Examples can be seen here:

```
%show results from first measurement and store calibration data
vectors = main('calibFile.mtb', 'dataFile.mtb')

%show results from second measurement and use stored calibration data
main(vectors, 'data2File.mtb')
```

Another possibility to read data without Xsens MT SDK is to use already exported data in **.txt* format. One **.txt* is for each sensor, this means two files (one set) for calibration data and at least one another data set with measurement itself. First this data needs to be imported and stored in Matlab, then calibration vectors need to be computed and finally angles from measured data itself could be processed, for example:

```
%store calibration data to variable
calibStruct = importXSensData('calibFileTxt.txt', 'calibFileTxt2.txt')

%store measurement data to variable
dataStruct = importXSensData('dataFileTxt.txt', 'dataFileTxt2.txt')

%compute and store calibration data
vectors = computeSeel(calibStruct)

%show results from measurement
calcJointAngles(vectors, dataStruct)
```

Example calibration and measurement data can be found on attached CD along with Matlab source files.