

GaitWatch: User Manual



Prof. Dr. med. Kai Bötzl^a
Dr. Alberto Olivares Vicente^b

^a Department of Neurology, Ludwig-Maximilians University, Munich, Germany.

^b Department of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain.



Contents

1	The GaitWatch Device	9
1.1	General description	9
1.2	Hardware description	9
2	Initial configuration	13
2.1	Identification of sensor's axes	13
2.1.1	Identification of Accelerometer's axes	14
2.1.2	Identification of Gyroscope's axes	17
2.1.3	Identification of Magnetometer's axes	19
3	Calibration of sensors	21
3.0.4	Accelerometer calibration	21
3.0.5	Magnetometer calibration	26
3.0.6	Gyroscope calibration	29
3.0.7	Wrap-up	32
4	Computation of Orientation	33
4.1	Computing the orientation using Euler Angles	33
4.1.1	Integration of angular rate	33
4.1.2	Decomposition of gravity	34

4.1.3	Fusing the angular rate with the orientation angle observation computed with the accelerometer	35
4.1.3.1	The Kalman filter	35
4.1.3.2	The Kalman filter applied to orientation estimation .	38
4.2	Quaternion-based orientation determination	43
4.2.1	Using angular rate to compute orientation	43
4.2.2	Using acceleration and magnetic field to compute orientation .	44
4.2.3	Fusing the orientation quaternions computed with the angular rate and the acceleration and magnetic field	48
4.2.3.1	The Extended Kalman Filter	48
4.2.3.2	The Extended Kalman Filter applied to orientation estimation	51
5	Experiments	53
5.1	Qualisys optical motion tracker	53
5.1.1	Computing Euler angles using Qualisys data	53
5.2	Qualisys validation experiments	55
5.2.1	Data gathering protocol	55
5.3	Fast Walking	57
5.4	Jumping	59
5.5	Leg motion	61
5.6	Freezing gait	63
5.7	Normal gait	65
Bibliography		67

List of Figures

1.1	General diagram of the GaitWatch being worn by a subject.	10
2.1	Definition of the desired body axes (a). Convention for the orientation of the axis rotation (b).	13
2.2	Orientation and axes of the GaitWatch box.	14
2.3	Identification of Z axis of the accelerometer in the box unit.	15
2.4	Identification of the acceleration signal gathered in the Z axis.	15
2.5	Identification of the box's acceleration Y (top) and X (bottom) axes. . .	16
2.6	Identification of the Z (left) and X (right) axes of the accelerometer in right shank's unit.	16
2.7	Identification of the sense of rotation of the Y axis of the gyroscope in the left hand unit.	18
2.8	Angular rate signal gathered during maneuvers of identification of rotation.	18
2.9	Magnetic field signal gathered during maneuvers of Z axis identifica- tion.	20
3.1	Acceleration gathered during quasi-static positions and output of the detection algorithm	23
3.2	3D representation of acceleration gathered during quasi-static posi- tions.	23
3.3	3D representation of calibrated quasi-static positions.	24

3.4	Raw acceleration gathered when both axes X and Z are placed parallel and anti-parallel to the gravity vector.	26
3.5	Raw magnetic field gathered by randomly moving the trunk unit containing the magnetometer.	27
3.6	Calibrated magnetic field gathered by randomly moving the trunk unit containing the magnetometer.	28
3.7	Calibrated magnetic field vs. raw magnetic field gathered by randomly moving the trunk unit containing the magnetometer.	28
3.8	Gyroscope calibration device.	29
3.9	Gyroscope calibration routine: selection of static period to compute bias.	31
3.10	Gyroscope calibration routine: selection of positive rotations to compute scale factor.	31
4.1	Effects of dynamic bias on the integrated gyroscope output signal. . .	34
4.2	General diagram of the standard approach of the Kalman filter without sensor fusion.	38
4.3	General diagram of the standard approach of the Kalman filter with sensor fusion.	42
5.1	An Oqus camera part of the Qualisys system.	53
5.2	Diagram of the pitch computation using the Qualisys system.	54
5.3	Pitch angle of left shank measured while walking fast (GaitWatch vs. Qualisys).	57
5.4	Pitch angle of left thigh measured while walking fast (GaitWatch vs. Qualisys).	57
5.5	Pitch angle of right shank measured while walking fast (GaitWatch vs. Qualisys).	58
5.6	Pitch angle of right thigh measured while walking fast (GaitWatch vs. Qualisys).	58
5.7	Pitch angle of left shank measured while jumping (GaitWatch vs. Qualisys).	59

5.8	Pitch angle of left thigh measured while jumping (GaitWatch vs. Qualisys).	59
5.9	Pitch angle of right shank measured while jumping (GaitWatch vs. Qualisys).	60
5.10	Pitch angle of right thigh measured while jumping (GaitWatch vs. Qualisys).	60
5.11	Pitch angle while swinging left shank. (GaitWatch vs. Qualisys).	61
5.12	Pitch angle while swinging left thigh. (GaitWatch vs. Qualisys).	61
5.13	Pitch angle while swinging right shank. (GaitWatch vs. Qualisys).	62
5.14	Pitch angle while swinging right thigh. (GaitWatch vs. Qualisys).	62
5.15	Left shank's pitch angle during freezing gait. (GaitWatch vs. Qualisys).	63
5.16	Left thigh's pitch angle during freezing gait. (GaitWatch vs. Qualisys).	63
5.17	Right shank's pitch angle during freezing gait. (GaitWatch vs. Qualisys).	64
5.18	Right thigh's pitch angle during freezing gait. (GaitWatch vs. Qualisys).	64
5.19	Left shank's pitch angle during normal gait. (GaitWatch vs. Qualisys).	65
5.20	Left thigh's pitch angle during normal gait. (GaitWatch vs. Qualisys).	65
5.21	Right shank's pitch angle during normal gait. (GaitWatch vs. Qualisys).	66
5.22	Right thigh's pitch angle during normal gait. (GaitWatch vs. Qualisys).	66

List of Tables

2.1	List of Gaitwatch's accelerometers and their associated channels.	17
2.2	List of Gaitwatch's gyroscopes and their associated channels.	19
2.3	Magnetometer's offsets and axes.	19
3.1	Calibration parameters of biaxial accelerometers (valid as of December the 10th, 2013).	26
3.2	Calibration parameters of gyroscopes (valid as of December the 10th, 2013).	32

Chapter 1

The GaitWatch Device

1.1 General description

GaitWatch is an Inertial Measurement Unit (IMU) designed for gait monitoring purposes. It was developed by Prof. Dr. Med. Kai Bötzel at the Department of Neurology of Ludwig-Maximilians University in Munich in conjunction with Dr. Alberto Olivares Vicente from the Department of Signal Theory, Telematics and Communications of the University of Granada.

GaitWatch is thought to be used in applications in which it is necessary to monitor the motion of patients.

The system is composed of a box which contains the central processing unit and a set of measuring units which are wired to it. The measuring units are thought to be placed in the patients' thighs, shanks, arms and trunk. Figure XXXX shows the appearance of the complete GaitWatch device.

1.2 Hardware description

As it was aforementioned, GaitWatch has a box containing the central processing unit as well as a set of embedded magnetic and inertial sensors. The microcontroller acting as the central processing unit is in charge of gathering the data from the external measurement units and writing it to the memory card together with the data from the embedded box sensors.

There are two different kinds of external units. The first one (type A IMU) is placed on both thighs and shanks. The second unit (type B IMU) is placed on both

arms. Figure 1.1 shows a diagram of the box (which contains the CPU and the type C IMU) together with the external units.

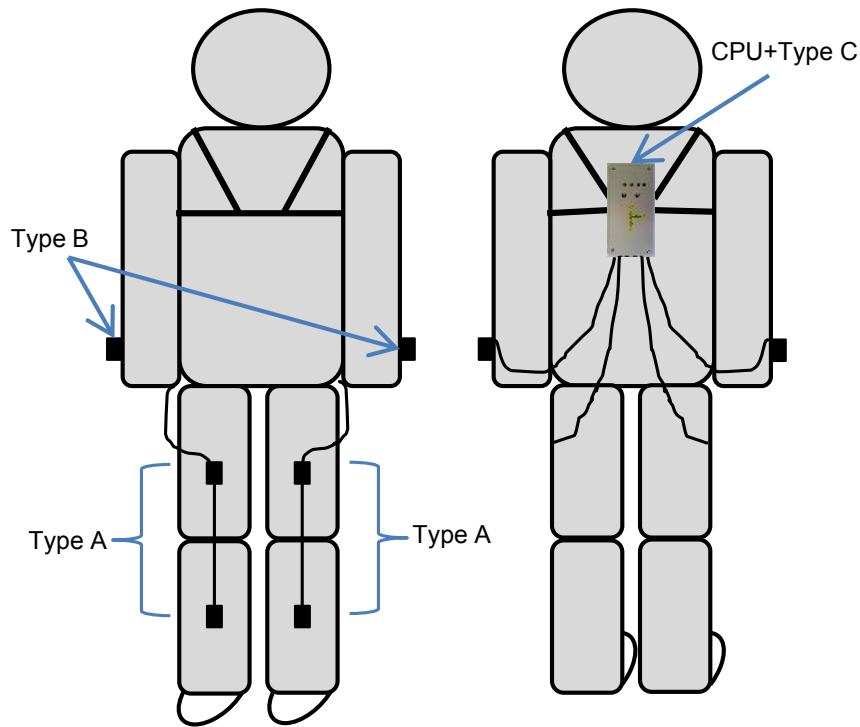


Figure 1.1: General diagram of the GaitWatch being worn by a subject.

The three different kinds of IMUs have the following components:

- Type A (thighs and shanks):
 - IMU 5 [1] from Sparkfun. IMU 5 contains an IDG500 [2] biaxial gyroscope (from which only Y axis is actually used) with a measurement range of $\pm 500\text{deg/s}$ and a $\pm 3g$ triaxial accelerometer, ADXL335 [3].
- Type B (arms):
 - IDG500 [2] biaxial $\pm 500\text{deg/s}$ gyroscope.
- Type C (trunk box):
 - ADXL345 [4] triaxial accelerometer with programmable range ($\pm 16g/\pm 8g/\pm 4g/\pm 2g$).

- IMU3000 [5] triaxial gyroscope with programmable range ($\pm 250/\pm 500/\pm 1000/\pm 3000(deg/s)$).
- Micromag3 [6] triaxial magnetometer ($\pm 11Gauss$).

In addition, the trunk box contains an AL-XAVRB [7] board containing an AVR ATxmega processor which contains the necessary embedded firmware to gather the data from all the measurement units and store it in a microSD card.

Figure XXX shows the schematics of the complete GaitWatch device.

Chapter 2

Initial configuration

2.1 Identification of sensor's axes

The first step that needs to be carried out is to determine the orientation of the axes of the body frame that we wish to use, as well as the orientation of the rotation around those axes. One of the most popular configurations is to set the X axis pointing forwards, the Y axis pointing to the right and the Z axis pointing down. This configuration follows the rule of the right hand for the orientation of the axes and the corkscrew rule for the rotation. Both are shown in Figure 2.1. These two standard navigation frames are usually employed when representing the orientation of a body in space and are known as the North-East-Down (NED) and the East-North-Up (ENU) frames.

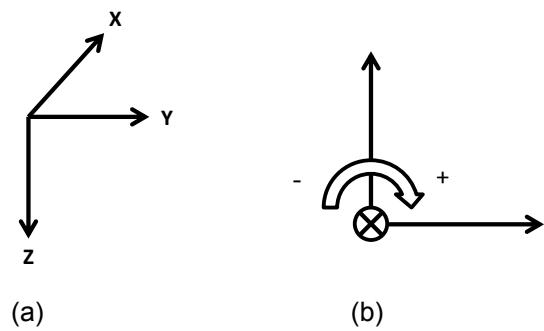


Figure 2.1: Definition of the desired body axes (a). Convention for the orientation of the axis rotation (b).

Since we will be using the GaitWatch device to monitor gait, then we need its X axis to point to the front of the patient, the Y axis pointing to the right of the patient, and the Z axis to the floor. We will start by configuring the orientation of the sensors which are inside the data gathering unit and which will be placed on the back of the patient. Figure 2.2 shows the desired orientation for Gaitwatch's box.

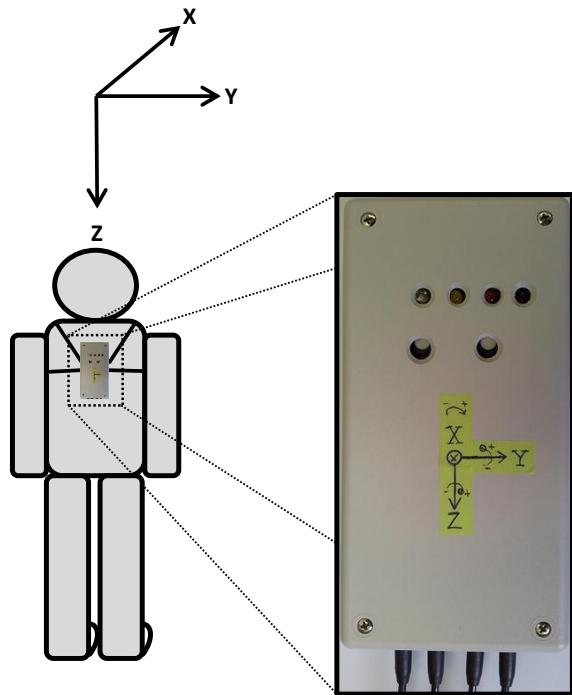


Figure 2.2: Orientation and axes of the GaitWatch box.

Once we know the desired orientation, the next step is to adapt the axes of the sensors' boards to it. We will start by adjusting the accelerometer.

2.1.1 Identification of Accelerometer's axes

The goal here is to determine the orientation of the axes and adapt it to the convention aforementioned. First, we set the box with the desired Z axis pointing downwards and upwards respectively while gathering data, as shown in figure 2.3.

Now we plot the acceleration gathered along the three axes. We have to identify the channel showing a large change in the measured acceleration.

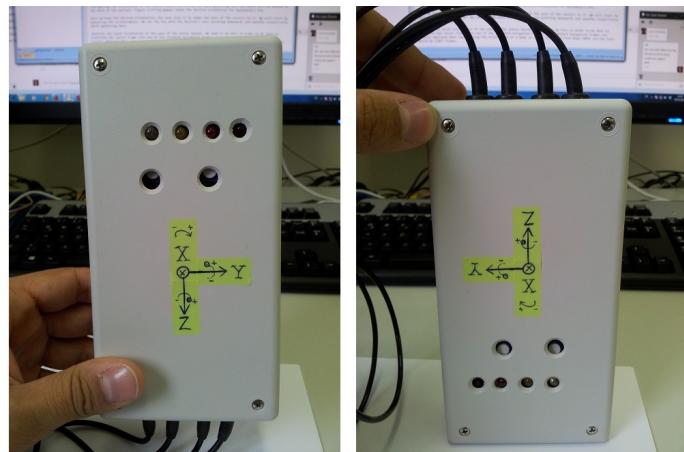


Figure 2.3: Identification of Z axis of the accelerometer in the box unit.

That channel will then be labeled as Z irrespective to which label it may have in the accelerometer board. That is, the sensor board may indicate that the selected channel is 'X' in the sensor frame, but we will label it as 'Z' in our desired frame which is the body frame.

If the Z axis of the sensors has the same orientation of the desired frame, then, when pointing the Z axis of the box downwards, the accelerometer will measure 1g since its axis is parallel to the Earth's gravity vector. If it has the opposite orientation, then the accelerometer will measure -1g. Since we have not calibrated the accelerometer so far and, therefore, the data are in raw units, the way to identify the correct orientation is by knowing that the measured output when setting the accelerometer axis parallel to gravity is larger than its output when it is set anti-parallel to the gravity vector. Figure 2.4 shows the triaxial gathered acceleration and the axis which should be labeled as 'Z'.

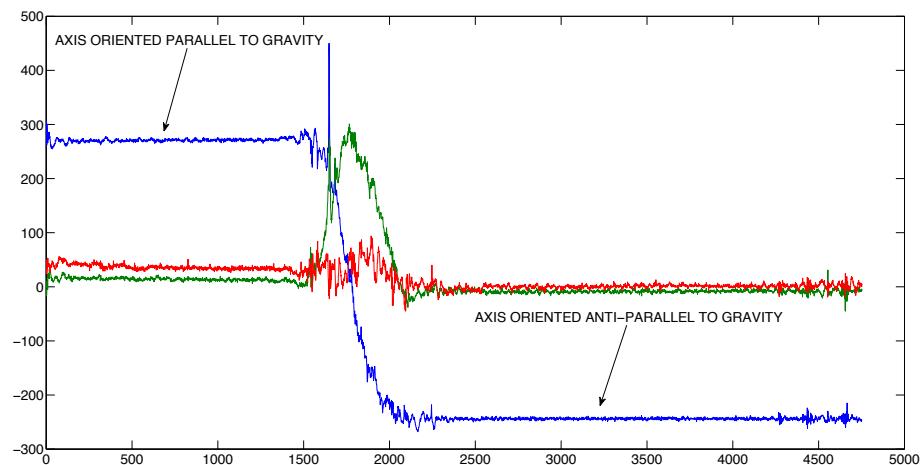


Figure 2.4: Identification of the acceleration signal gathered in the Z axis.

For the case of the GaitWatch, the signal showing both parallel and anti-parallel acceleration values is in channel 20. The orientation of the axis is correct, since we first put the desired Z axis pointing downwards and the accelerometer showed the largest acceleration value in that position. We, thus, set channel 20 to be the acceleration in Z axis. The remaining two axes (X and Y) are identified following the same procedure. Figure 2.5 shows the box being oriented negatively and positively along X and Y axes. Both X and Y axes are properly oriented and need no orientation correction. After inspecting the gathered signals, we observe that X and Y measurements are contained in channels 22 and 21, respectively.



Figure 2.5: Identification of the box's acceleration Y (top) and X (bottom) axes.

Once we have identified the axes of the accelerometer inside the box which will be placed in the back of the patient, we need to apply the same aforementioned procedure to identify, on by one, the axes of the biaxial accelerometers which will be placed in the thighs and the shanks. Figure 2.6 shows the right shank module being placed parallel and anti-parallel to the gravity in both X and Z axes.

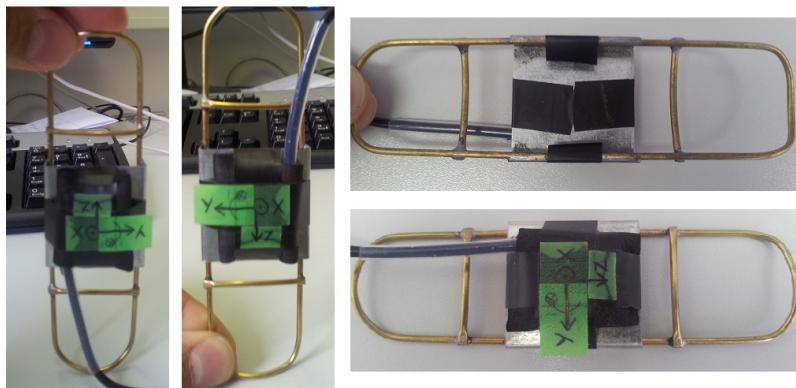


Figure 2.6: Identification of the Z (left) and X (right) axes of the accelerometer in right shank's unit.

Table 2.1: List of Gaitwatch's accelerometers and their associated channels.

Unit	Axis	Orientation compensation	Channel
Right Shank	Z	No (+1)	2
Right Shank	X	No (+1)	3
Right Thigh	Z	No (+1)	5
Right Thigh	X	No (+1)	6
Left Shank	Z	No (+1)	8
Left Shank	X	No (+1)	9
Left Thigh	Z	No (+1)	11
Left Thigh	X	No (+1)	12
Trunk	Z	No (+1)	20
Trunk	Y	No (+1)	21
Trunk	X	No (+1)	22

Table 2.1 shows the axes of all Gaitwatch's accelerometers, their associated channel and the orientation compensation (if needed).

2.1.2 Identification of Gyroscope's axes

Once we have identified the axes of the accelerometers, we now proceed to identify the axes of the gyroscopes and their orientation. By convention, as it is depicted in figure 2.1, the sense of the rotation around a given axis is positive when the axis is pointing forwards (from the perspective of the user) and it is turned to the right. Analogously, the rotation is negative when it is turned to the left. Knowing this, in order to identify the sense of rotation of the gyroscope and adapt it (if needed) to the body frame we need to proceed as follows:

1. Place the axis that we want to analyze so it is pointing forwards.
2. Rotate it to the right and then to the left. Repeat this three or four times.
3. Identify the channel containing the data and plot them. If the measured angular rate first increases and then decreases, the sense of rotation of the sensor axis is correct. On the other hand, if the angular has the opposite behavior, that is, it first decreases and then increases, we need to change the sense of the rotation.

Figure 2.7 shows the maneuvers to identify the rotation sense of the Y axis gyroscope located in the left hand unit. Figure 2.8 shows the measured angular rate for such maneuvers. Notice how the first rotation is positive (the signal increases and then

decreases), so, for the case of the left hand unit, the Y axis has the correct sense of rotation.

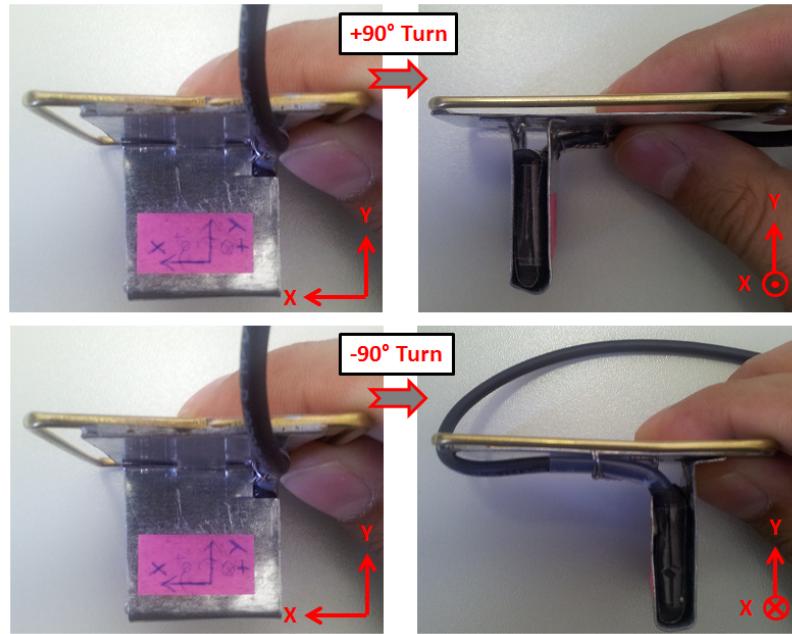


Figure 2.7: Identification of the sense of rotation of the Y axis of the gyroscope in the left hand unit.

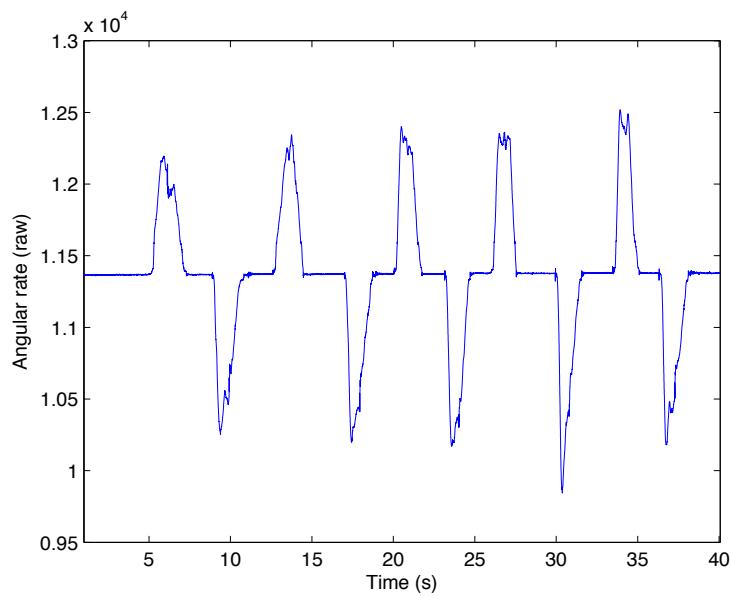


Figure 2.8: Angular rate signal gathered during maneuvers of identification of rotation.

This procedure needs to be repeated for each one of the axis of the gyroscopes located in all the units.

Table 2.2 shows the channels of all GaitWatch's gyroscopes.

Table 2.2: List of Gaitwatch's gyroscopes and their associated channels.

Unit	Axis	Sense compensation	Channel
Right shank	Y	Yes (-1)	1
Right thigh	Y	Yes (-1)	4
Left shank	Y	Yes (-1)	7
Left thigh	Y	Yes (-1)	10
Left arm	Y	No (+1)	13
Left arm	X	Yes (-1)	14
Right arm	Y	No (+1)	15
Right arm	X	Yes (-1)	16
Trunk	Z	No (+1)	17
Trunk	Y	No (+1)	18
Trunk	X	Yes (-1)	19

2.1.3 Identification of Magnetometer's axes

The data of the triaxial magnetometer embedded in the trunk unit are all contained in the same channel (channel 23). Each axis data has a different offset so it is easy to separate them. The axes offsets are +32764, 20000 and 0. Once we have separated the data, we need to identify which axis corresponds to each offset. To do so, we place the desired axis pointing upwards or downwards and turn the box more than 360 degrees. Then, the signal corresponding to the selected axis should be the one showing the least variation (as it is shown in figure 2.9). This operation needs to be repeated two more times to identify the remaining two axes.

Table 2.3: Magnetometer's offsets and axes.

Unit	Axis	Offset
Trunk	X	-20000
Trunk	Y	0
Trunk	Z	+32764

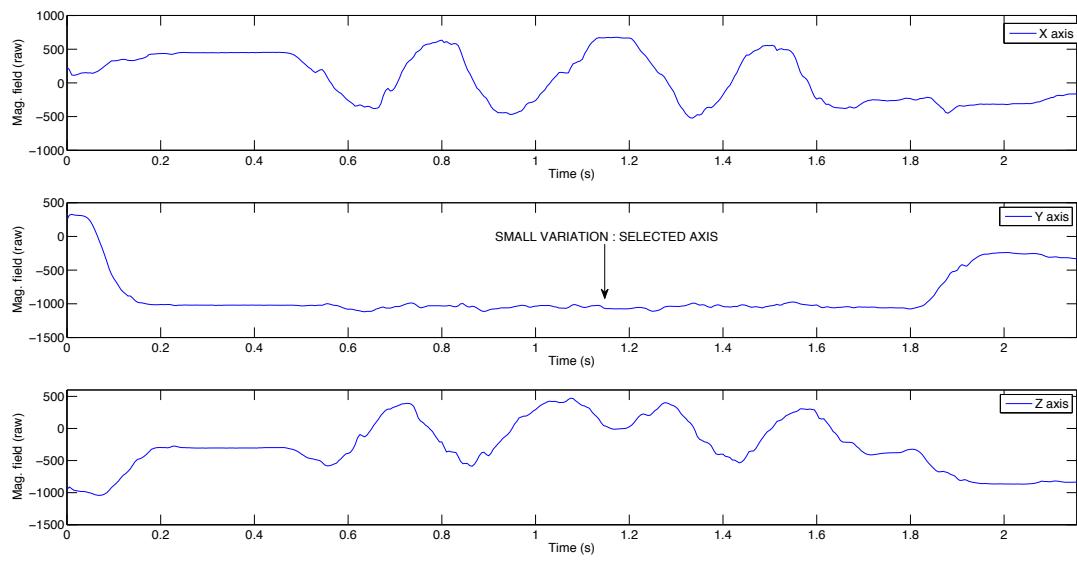


Figure 2.9: Magnetic field signal gathered during maneuvers of Z axis identification.

Chapter 3

Calibration of sensors

Once the axes are properly identified, the next step is to calibrate all its sensors. The main goal of the calibration process is to transform the raw data into meaningful physical units.

3.0.4 Accelerometer calibration

The calibration of the accelerometers is divided in two parts according to the number of axes of the accelerometer. For the triaxial accelerometer included in the trunk unit (the box) we will use the ellipsoid fitting algorithm by Camps et al. [8]. They present a theoretical and experimental method to compute gains, bias and non orthogonality factors of magnetometer and accelerometer sensors. The calibration procedure involves arbitrary rotations of the MIMU, so the set of maneuvers to gather the necessary data is very simple.

- **Sensor Modeling:** The model of the sensor output is given by:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} m_x(t) \\ m_y(t) \\ m_z(t) \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} \quad (3.1)$$

where $\mathbf{b} = (b_x, b_y, b_z)^T$ represents the offset, s_x, s_y, s_z are the sensor gains, $m_x(t), m_y(t), m_z(t)$ are the components of the actual magnetic field or gravity and $\varepsilon_x, \varepsilon_y, \varepsilon_z$ are the components of the noise for each axis.

If we take into consideration the effects of non-orthogonality, then the sensor

output is given by:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix} \begin{bmatrix} m_x(t) \\ m_y(t) \\ m_z(t) \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} \quad (3.2)$$

where s_{ij} for $i \neq j$ represent the orthogonality errors between sensor axes i and j .

- **Calibration Procedure:** Using the fact that the norm of the input vector $\mathbf{m} = [m_x(t) \ m_y(t) \ m_z(t)]^T$ is constant, the following relation is derived:

$$\|\mathbf{m}\|^2 = m_x(t)^2 + m_y(t)^2 + m_z(t)^2 \quad (3.3)$$

$$\|\mathbf{h}\|^2 = \left(\frac{v_x(t) - b_x}{s_x} \right)^2 + \left(\frac{v_y(t) - b_y}{s_y} \right)^2 + \left(\frac{v_z(t) - b_z}{s_z} \right)^2 \quad (3.4)$$

Equation (3.4) is the parametric equation of an ellipsoid with center \mathbf{b} and semi-axes s_x, s_y and s_z . Using the system of equations formed by the various measurements at times t , we estimate the parameters through non-linear least squares minimization of the error function

$$e_p(t) = \|\mathbf{m}\|^2 - (\mathbf{v}(t) - \mathbf{b})^T (S^{-1})^2 (\mathbf{v}(t) - \mathbf{b}) \quad (3.5)$$

where

$$S^{-1} = \begin{bmatrix} 1/s_{xx} & 1/s_{xy} & 1/s_{xz} \\ 1/s_{xy} & 1/s_{yy} & 1/s_{yz} \\ 1/s_{xz} & 1/s_{yz} & 1/s_{zz} \end{bmatrix} \quad (3.6)$$

The cost function $e_p(t)$ is quadratic, and is minimized iteratively by the *Levenberg-Marquardt* algorithm (LMA) [9, 10].

Long story short, if the accelerometer is placed in multiple random quasi-static positions, the gathered data should ideally describe a sphere of radius equal to the magnitude of the gravity vector.

Therefore, the first calibration step is to gather the acceleration data which will be used to find the optimal calibration parameters. As it is said before, we need to place the unit containing the accelerometer in multiple random quasi-static positions. To do so, we put the box in a set of random positions trying to cover all the orientations. Since the transitions from one quasi-static position to other cause the accelerometer to measure linear acceleration which disrupt the gravity acceleration, we can not just take every measured point but we need to apply an algorithm which is able to detect the quasi-static instants.

Figure 3.1 show the triaxial acceleration measured in each quasi-static position in

addition to the undesired linear acceleration. This figure also shows the output of the quasi-static instant detection algorithm.

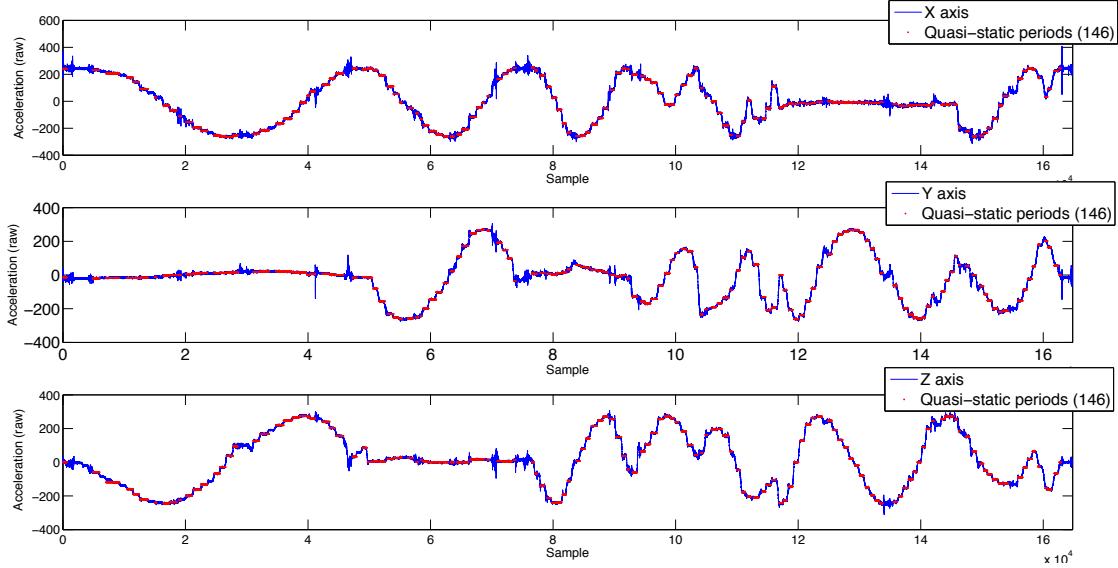


Figure 3.1: Acceleration gathered during quasi-static positions and output of the detection algorithm

If we plot the detected quasi-static accelerations in 3D, they should cover the locus of a sphere as it is shown in figure 3.2

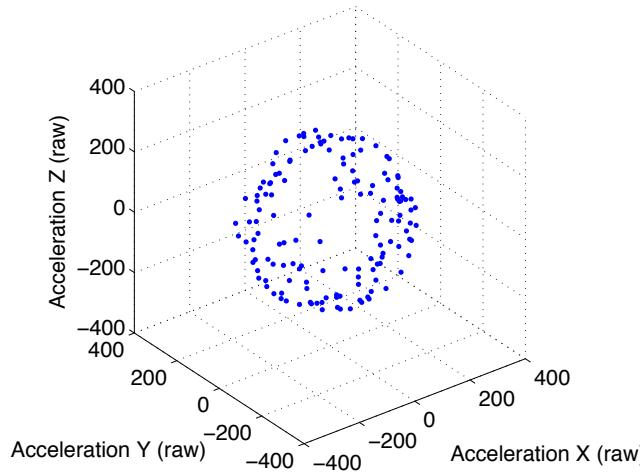


Figure 3.2: 3D representation of acceleration gathered during quasi-static positions.

We then feed the algorithm with these data and the calibration parameters are found. Equation 3.7 shows the estimated calibration parameters for the trunk tri-

axial accelerometer.

$$S = \begin{bmatrix} 256.68 & -4.01 \cdot 10^{-5} & -2.41 \cdot 10^{-5} \\ -4.01 \cdot 10^{-5} & 262.74 & -3.98 \cdot 10^{-7} \\ -2.41 \cdot 10^{-5} & -3.98 \cdot 10^{-7} & 263.04 \end{bmatrix} \quad (3.7)$$

$$\mathbf{b} = [-23.69 \quad -6.95 \quad 22.85]$$

256,689789868850 -4,01238483601604e-05 -2,41541345424999e-05 -4,01238483601604e-05 262,746531495908 -3,98690478210809e-07 -2,41541345424999e-05 -3,98690478210809e-07 263,047756176519 These parameters are then plugged in the following equation to obtain the calibrated triaxial acceleration:

$$\mathbf{a}_{cal} = S^{-1}(\mathbf{a}_{raw} - \mathbf{b}) \quad (3.8)$$

where \mathbf{a}_{cal} is a three dimensional vector containing the calibrated acceleration, S is the computed orthogonality matrix, \mathbf{a}_{raw} is the three dimensional vector containing the raw acceleration and \mathbf{b} is the computed triaxial bias vector.

Figure 3.3 shows the calibrated quasi-static accelerations. Notice how the points define now a sphere centered in the origin with a radius of 1(g).

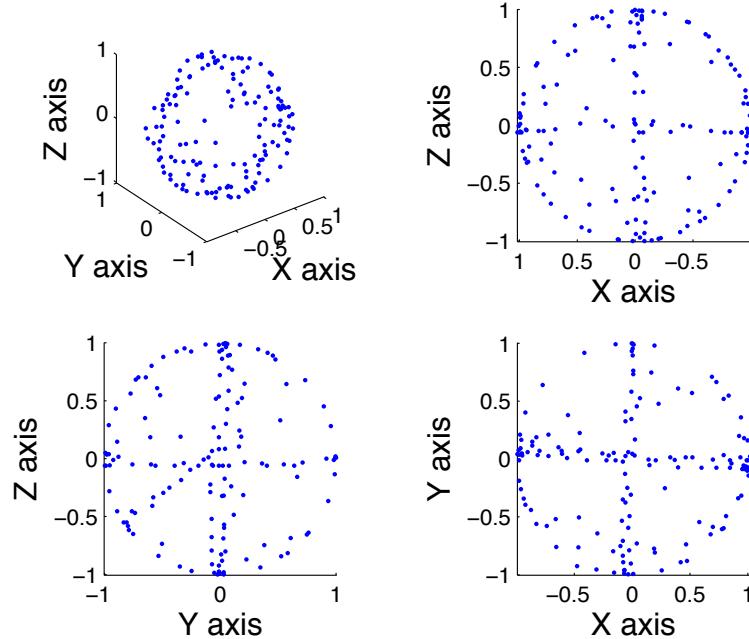


Figure 3.3: 3D representation of calibrated quasi-static positions.

For the biaxial accelerometer we could adapt the method aforementioned to two dimensions. However, the calibration maneuvers would be much more complicated since we would need to lock the unit including the accelerometer in the XZ plane and then describe a complete slow rotation to gather the quasi-static positions.

We opted to carry out a single-axis calibration procedure which only requires to gather data from two positions for each axis. This procedure is very simple; we first set the desired axis parallel to the gravity vector and leave it in that position for a couple of seconds and we then flip it so it is anti-parallel to the gravity vector and leave it static again for another two seconds. Proceeding this way, we know the raw values corresponding to +1g and -1g respectively. Since the accelerometers included in GaitWatch's units are linear, we can then find the calibration equation in a very simple way. The calibration equation is defined as follows,

$$a_{cal} = k \cdot a_{raw} + b \quad (3.9)$$

where a_{cal} is the calibrated acceleration, a_{raw} is the raw acceleration, k is the scale factor and b is the bias. If we substitute the two gathered values we would have the following system of equations from which it is straight forward to find the values of k and b ,

$$\begin{aligned} 1 &= k \cdot a_{raw,+} + b \\ -1 &= k \cdot a_{raw,-} + b \end{aligned} \quad (3.10)$$

where $a_{raw,+}$ and $a_{raw,-}$ are the gathered raw accelerations in the parallel and anti-parallel positions respectively.

The units placed on the shanks and the thighs contain a biaxial accelerometer (X,Z) so we can put the unit in the four required positions (two for each axis) and then apply the quasi-static positions detector to extract the values in these positions.

Figure 3.4 shows the gathered raw acceleration in such four positions and the extracted quasi-static values for the accelerometer in the left shank unit.

Using this procedure the calibration parameters are found for all the units. Table 3.1 shows them.

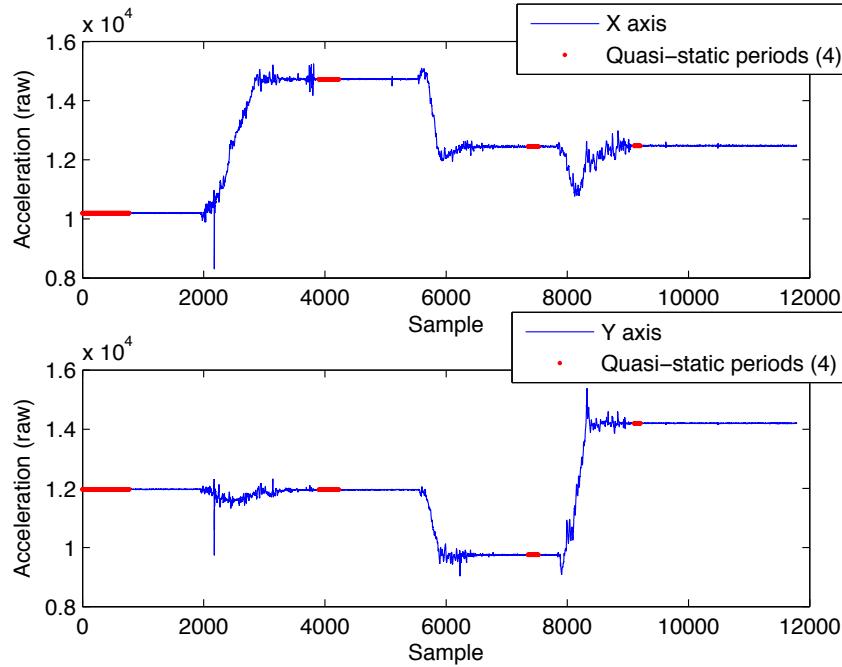


Figure 3.4: Raw acceleration gathered when both axes X and Z are placed parallel and anti-parallel to the gravity vector.

Table 3.1: Calibration parameters of biaxial accelerometers (valid as of December the 10th, 2013).

Unit	Axis	Scale factor	Bias
Left shank	X	4.45e-04	-5.59
Left shank	Z	4.49e-04	-5.36
Left thigh	X	4.58e-04	-5.76
Left thigh	Z	4.74e-04	-5.70
Right shank	X	4.38e-04	-5.50
Right shank	Z	4.77e-04	-5.72
Right thigh	X	4.36e-04	-5.44
Right thigh	Z	4.43e-04	-5.34

3.0.5 Magnetometer calibration

The calibration of the triaxial magnetometer is also based on the algorithm by Camps et al. used for the accelerometer calibration. In this case, we only need to change the data gathering maneuvers and the value of the magnitude of the reference vector (which in this case is the Earth's magnetic field vector).

The maneuvers are much simpler in this case since the magnetometer is not affected by the linear acceleration. Therefore, we can freely move the magnetometer

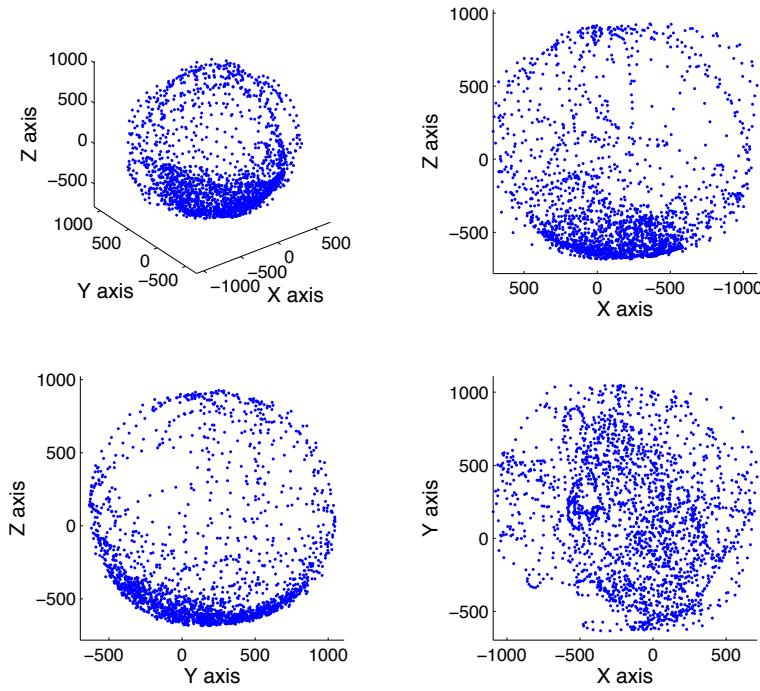


Figure 3.5: Raw magnetic field gathered by randomly moving the trunk unit containing the magnetometer.

at any speed trying to cover the most possible space. Figure 3.5 shows the 3D representation of the raw magnetic field measured carrying these maneuvers.

Ideally, these data should describe an sphere centered in the origin with a radius given by the magnitude of the local Earth's magnetic field. By checking the Magnetic field calculator from the American National Geophysical Data Center [11], we can obtain the value for Munich, which is 0.482352 Gauss. We then feed the algorithm with the data shown in figure 3.5 and the calibration parameters are computed. Analogously to the accelerometer, the calibrated magnetic field is found applying the following equation,

$$\mathbf{h}_{cal} = S^{-1}(\mathbf{h}_{raw} - \mathbf{b}) \quad (3.11)$$

where again \mathbf{h}_{cal} is a three dimensional vector containing the calibrated magnetic field, S is the computed orthogonality matrix, \mathbf{h}_{raw} is the three dimensional vector containing the raw magnetic field and \mathbf{b} is the computed triaxial bias vector. If we apply this equation to the raw values used to compute the parameters, we can see how they now define a sphere centered in 0 and with radius 0.482352 Gauss (figure

3.6).

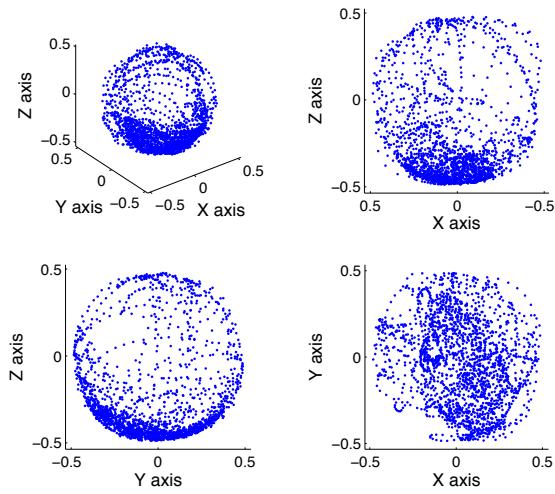


Figure 3.6: Calibrated magnetic field gathered by randomly moving the trunk unit containing the magnetometer.

Figure 3.7 shows the raw and calibrated signals for all the three axes of the magnetometer included in the trunk's unit.

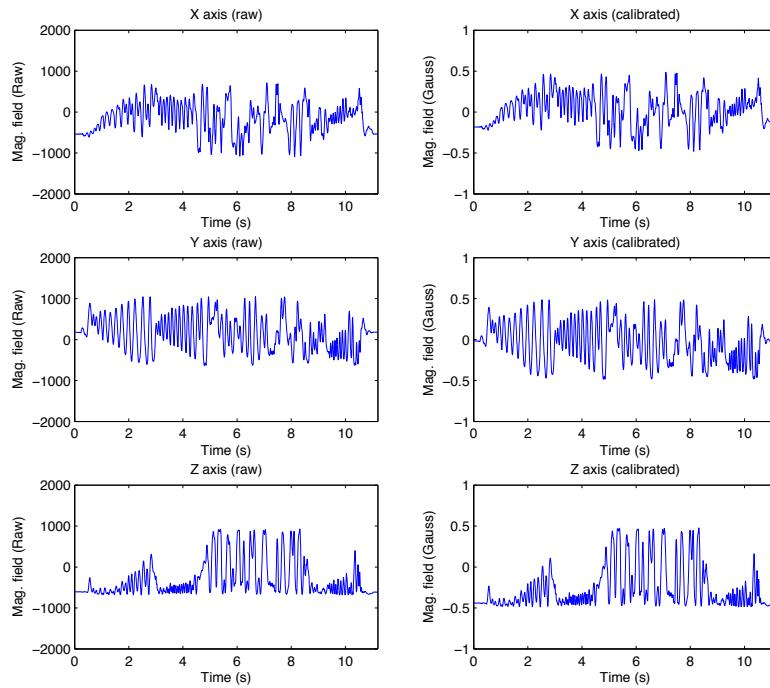


Figure 3.7: Calibrated magnetic field vs. raw magnetic field gathered by randomly moving the trunk unit containing the magnetometer.

The calibration parameters (valid as of December the 10th, 2013) of the magnetometer in the trunk unit are listed below

$$S = \begin{bmatrix} 1.75 \cdot 10^3 & -3.46 \cdot 10^{-8} & -3.04 \cdot 10^{-6} \\ -3.46 \cdot 10^{-8} & 1.97 \cdot 10^3 & 1.42 \cdot 10^{-5} \\ -3.04 \cdot 10^{-6} & 1.42 \cdot 10^{-5} & 1.83 \cdot 10^3 \end{bmatrix} \quad (3.12)$$

$$\mathbf{b} = [14.86 \quad 102.43 \quad -45.04]$$

3.0.6 Gyroscope calibration

The gyroscope is the last remaining kind of sensor to be calibrated. The most accurate way to calibrate a gyroscope is to subject it to different known rotation speeds and then associating them to the raw gathered values to obtain a calibration equation. If a variable rate table is not available, we can use simpler equipment and subject the gyroscope to known rotations instead of known angular rates. To calibrate each one of the axis we need to build a device like the one shown in figure 3.8. This device should allow the rotation of the unit containing the gyroscope around 180° (other configurations of the device allowing different known rotations are also valid).

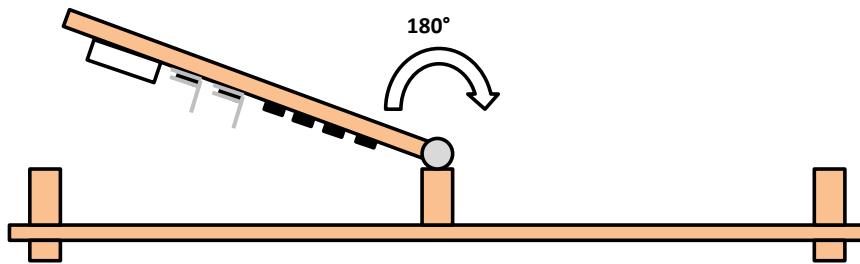


Figure 3.8: Gyroscope calibration device.

Once the calibration device is ready, the calibration maneuvers are not complicated. Each of the units should be attached to the surface of the device and then we need to carry out a set of 180° rotations (at least 5).

To compute the calibration parameters we are going to use the fact that the integration of the angular speed leads to an angle displacement. Therefore, if we subject the gyroscope to a known rotation (180° degrees in this case), we can apply the following equations to find the scale factor,

$$u_{raw} = k \cdot \omega + b \quad (3.13)$$

$$\omega = \frac{u_{raw} - b}{k} \quad (3.14)$$

$$u_{raw,corr} = u_{raw} - b \quad (3.15)$$

$$\int \omega dt = \frac{\int u_{raw,corr} dt}{k} \quad (3.16)$$

$$\Omega = \int \omega dt \quad (3.17)$$

$$\theta = \int u_{raw,corr} dt \quad (3.18)$$

$$\Omega = \frac{\theta}{k} \quad (3.19)$$

$$k = \frac{\theta}{\Omega} \quad (3.20)$$

$$(3.21)$$

where u_{raw} is the raw gathered angular rate, ω is the actual angular rate (in physical units), k is the scale factor and b is the bias. In this case $\Omega = 180^\circ$ and θ is the integrated raw gathered angular rate (corrected in bias) during the rotation of 180° . We can see that prior to the integration of the raw angular rate we need to subtract the bias.

The bias is easily found by leaving the units in a static position before starting the rotation and then visually inspecting the gathered signal. To take into account the effects of noise, we compute the mode of all the samples gathered while the sensors are static.

From a practical point of view, the Matlab calibration routine first displays the gathered raw angular rate signal and asks the user to select the initial and final points of the static period as it is shown in figure 3.9.

Once the bias is found, we need to tell the routine each one of the starting and ending points of the rotations so it knows the initial and final time instants of the integration. To do so we use the *datacursor*. It is important to only select the positive rotations (as indicated in figure 3.10), that is, those in which the angular rate first increases and then decreases. The routine will then integrate the selected segments of the signal and average the computed scale factor for each one of them.

The calibrated angular rate is finally found by substituting the computed calibration parameters in equation (3.14).

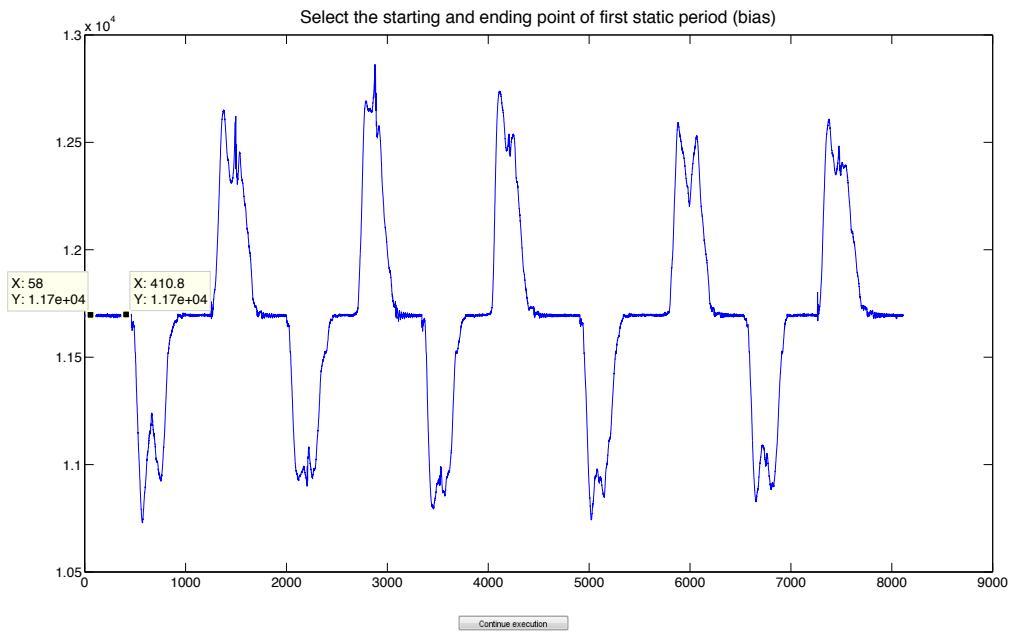


Figure 3.9: Gyroscope calibration routine: selection of static period to compute bias.

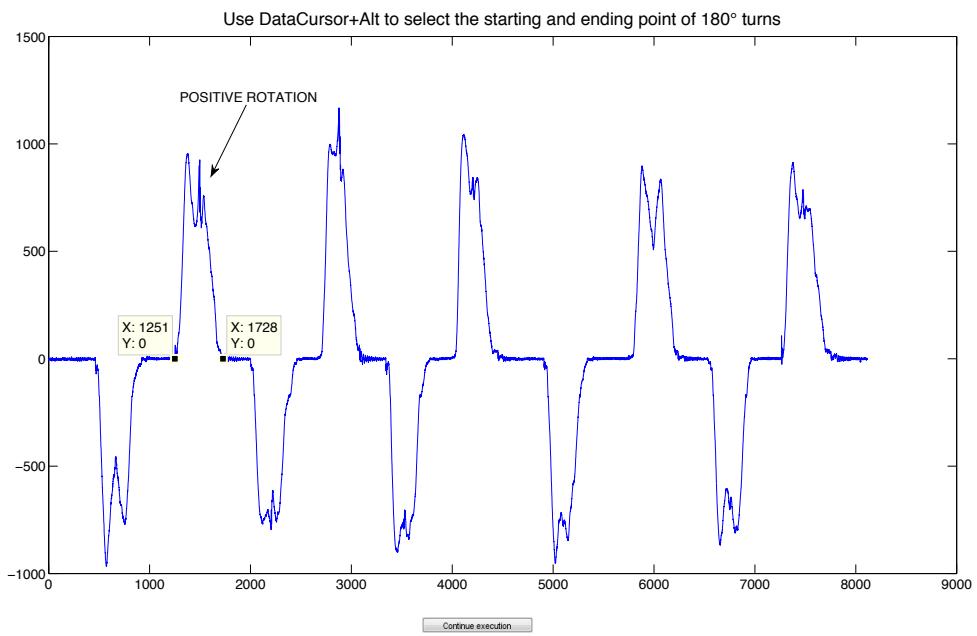


Figure 3.10: Gyroscope calibration routine: selection of positive rotations to compute scale factor.

Table 3.2 lists the computed calibration parameters for all GaitWatch's gyroscopes.

Table 3.2: Calibration parameters of gyroscopes (valid as of December the 10th, 2013).

Unit	Axis	Scale factor	Bias
Left shank	Y	-5.84	11591
Left thigh	Y	-6.36	11720
Right shank	Y	-5.95	11618
Right thigh	Y	-6.40	11740
Left arm	X	-5.89	12087
Left arm	Y	5.75	10937
Right arm	X	-6.02	11543
Right arm	Y	5.64	11415
Trunk	X	-16.15	-10
Trunk	Y	16.02	62
Trunk	Z	16.23	31

3.0.7 Wrap-up

As a final summary, once all the calibration parameters from all the sensors are computed, we only need to apply the following equations,

- **Accelerometer:**

- *Triaxial accelerometer:*

$$\mathbf{a}_{cal} = S^{-1}(\mathbf{a}_{raw} - \mathbf{b}) \quad (3.22)$$

- *Biaxial accelerometer:*

$$a_{cal} = k \cdot a_{raw} + b \quad (3.23)$$

- **Magnetometer:**

$$\mathbf{h}_{cal} = S^{-1}(\mathbf{h}_{raw} - \mathbf{b}) \quad (3.24)$$

- **Gyroscope:**

$$\omega_{cal} = \frac{u_{raw} - b}{k} \quad (3.25)$$

Chapter 4

Computation of Orientation

4.1 Computing the orientation using Euler Angles

4.1.1 Integration of angular rate

Triaxial gyroscopes measure the angular rate around each one of the Cartesian axes relative to the body frame. Such angular rate can be integrated to estimate the rotation angle from time instants t_0 to t_1 . Therefore, we can compute the position of the body relative to the navigation frame if, and only if, the initial orientation of the body frame with respect to the navigation frame is known. A straight forward solution is to use the approach based on the projection of the gravity and earth's magnetic field vectors to compute such initial orientation when the body is static.

Many different numerical integration methods can be used to integrate the angular rate. In this case we will use the numerical integration based on the trapezoidal rule which calculates the area under the function performing a trapezoidal approximation as follows:

$$\alpha(n) = \alpha_0 + \frac{T}{2} \sum_{k=1}^n [\omega_g(k) + \omega_g(k-1)] \quad (4.1)$$

Where T is the sample period, n the sample number, $\omega_g(n)$ the gyro measurement at instant n and α_0 the angle at $n = 0$. The angle can also be computed in a recursive way using

$$\alpha(n) = \alpha(n-1) + \frac{T}{2} [\omega_g(n) + \omega_g(n-1)] \quad (4.2)$$

where $\alpha(0) = \alpha_0$.

The main advantage of this method is that, once the initial orientation is known,

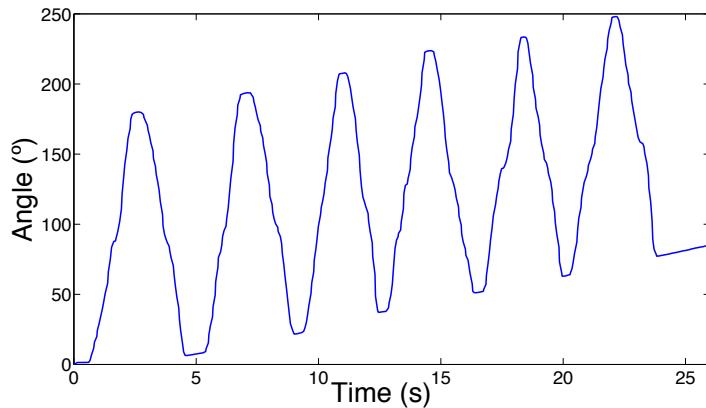


Figure 4.1: Effects of dynamic bias on the integrated gyroscope output signal.

it can be used either under low or high intensity motion conditions as the gyroscope measurements are not affected by linear acceleration. However, integration of noise and dynamic bias leads to a quick time growing offset which disrupts the computed angle. The effects of such integration are so devastating that the orientation estimate is absolutely erroneous after a few seconds as it can be seen in figure 4.1).

This method is, itself, unfeasible if not fused with the accelerometer and magnetometer approach.

4.1.2 Decomposition of gravity

We can use the projection of the Earth's gravity vector on the accelerometer's axes to estimate the orientation angle of the body frame with respect to the reference frame. In our case, this approach is used to compute the orientation angle of the thighs and shanks. Since only a biaxial accelerometer is available in the IMUs placed on these segments, we can only compute one of the two orientation angles (roll and pitch) which can be computed by means of this technique. We are interested on the angle of the segments defined in the XZ plane, that is, the angle around Y axis, which is defined as the pitch angle. The computation of the pitch will only be valid if the roll angle of the legs is close to zero. Since the legs move in the sagittal plane during gait, we can consider that the following expression will be valid to compute the pitch angle,

$$\theta_k = \text{atan2}(a_{z,k}, a_{x,k}) \quad (4.3)$$

where *atan2* is the quadrant-corrected arctangent function.

4.1.3 Fusing the angular rate with the orientation angle observation computed with the accelerometer

4.1.3.1 The Kalman filter

The Kalman filter [12], [13], [14] is a set of mathematical equations that provides a means to estimate the state of a process minimizing the mean of the squared error.

If we consider a time invariant linear continuous model:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) + \mathbf{C}\mathbf{u}(t) + \mathbf{I}_n\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) + \mathbf{I}_q\mathbf{v}(t)\end{aligned}\tag{4.4}$$

where,

- the operator $\dot{\mathbf{x}}$ denotes the first derivative of \mathbf{x} .
- $\mathbf{x}(t) \in \mathbb{R}^n$ is the $(n \times 1)$ state vector including the n state variables that describe the system.
- $\mathbf{z}(t) \in \mathbb{R}^q$ is the $(q \times 1)$ output vector including the q real observations (measurements) of the state variables of the process.
- $\mathbf{u}(t) \in \mathbb{R}^p$ is the $(p \times 1)$ control vector including the p input control signals.
- \mathbf{F} is the $(n \times n)$ state transition matrix, or simply the state matrix.
- \mathbf{C} is the $(n \times p)$ input matrix.
- \mathbf{H} is the $(q \times n)$ output matrix.
- \mathbf{I}_n is the $(n \times n)$ identity matrix.
- \mathbf{I}_q is the $(q \times q)$ identity matrix.
- $\mathbf{w}(t) \in \mathbb{R}^n$ is the $(n \times 1)$ process noise vector.
- $\mathbf{v}(t) \in \mathbb{R}^q$ is the $(q \times 1)$ observation noise vector.

In the absence of a control input we have:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) + \mathbf{I}_n\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) + \mathbf{I}_q\mathbf{v}(t)\end{aligned}\tag{4.5}$$

If we apply the forward-Euler approximation:

$$\dot{\mathbf{x}}(t) \simeq \frac{\mathbf{x}(t + dt) - \mathbf{x}(t)}{dt}\tag{4.6}$$

we get:

$$\begin{aligned}\frac{\mathbf{x}(t + dt) - \mathbf{x}(t)}{dt} &\simeq F\mathbf{x}(t) + I_n \mathbf{w}(t) \\ \mathbf{x}(t + dt) &\simeq [F\mathbf{x}(t) + I_n \mathbf{w}(t)] dt + \mathbf{x}(t) \\ \mathbf{x}(t + dt) &\simeq F\mathbf{x}(t) dt + I_n \mathbf{w}(t) dt + \mathbf{x}(t) \\ \mathbf{x}(t + dt) &\simeq [F dt + I_n] \mathbf{x}(t) + I_n \mathbf{w}(t) dt\end{aligned}\tag{4.7}$$

If we now discretize the model:

$$\begin{aligned}\mathbf{x}_{k+1} &= [F dt + I_n] \mathbf{x}_k + I_n dt \mathbf{w}_k \\ \mathbf{x}_{k+1} &= \Phi \mathbf{x}_k + I_n dt \mathbf{w}_k \\ \mathbf{z}_k &= H \mathbf{x}_k + I_q \mathbf{v}_k\end{aligned}\tag{4.8}$$

or how it is usually represented:

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + B \mathbf{w}_{k-1}\tag{4.9}$$

$$\mathbf{z}_k = H \mathbf{x}_k + I_q \mathbf{v}_k\tag{4.10}$$

where,

- $\mathbf{x} \in \mathbb{R}^n$ is again the ($n \times 1$) state vector of the linear dynamic system.
- $\mathbf{z} \in \mathbb{R}^q$ is again the ($q \times 1$) vector of observations.
- Φ is now the ($n \times n$) state transition matrix of the discrete linear dynamic system.
- H is again the ($q \times n$) measurement sensitivity matrix defining the linear relationship between the state of the dynamic system and measurements that can be made.
- $B = I_n dt$, where dt is the sampling period.
- $\mathbf{w} \in \mathbb{R}^n$ is again the ($n \times 1$) process noise vector.
- $\mathbf{v} \in \mathbb{R}^q$ is again the ($q \times 1$) observation noise vector.

So, when applied to discrete signals, the Kalman filter aims to solve the problem of trying to estimate the state $\mathbf{x} \in \mathbb{R}^n$ of a discrete-time controlled process that is defined by the model in equations (4.9) and (4.10). In addition, \mathbf{w}_k and \mathbf{v}_k are assumed to be independent of each other, white, and with normal probability distributions

$$p(\mathbf{w}) \sim N(0, Q)\tag{4.11}$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}) \quad (4.12)$$

where, in turn, \mathbf{Q} and \mathbf{R} are the process noise covariance and measurement noise covariance matrices respectively and can be considered to be invariant in practice.

When deriving the equations for the Kalman filter, the goal is to find an equation that computes an a posteriori (updated/corrected with the measurement) state estimate $\hat{\mathbf{x}}_k$ as a linear combination of an a priori (predicted) estimate $\hat{\mathbf{x}}_k^-$ and a weighted difference between an actual measurement \mathbf{z}_k and a measurement prediction $H\hat{\mathbf{x}}_k^-$,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-). \quad (4.13)$$

The difference in (4.13) is known as the measurement innovation, or the residual. The residual indicates the discordance between the predicted measurement $H\hat{\mathbf{x}}_k^-$ and the actual measurement \mathbf{z}_k .

The matrix K_k in (4.13) is known as the Kalman gain and is chosen to be the gain that will minimize the a posteriori error covariance. Another way to see the weighting by K_k is that as the measurement error covariance gets close to zero, the actual measurement \mathbf{z}_k is trusted more, while the predicted measurement $H\hat{\mathbf{x}}_k^-$ is trusted less. Analogously, as the a priori estimate error covariance approaches zero the actual measurement is trusted less, while the predicted measurement is trusted more.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. As such, the equations for the Kalman filter are divided into two groups: time update equations and measurement update equations.

The time update equations can also be thought of as prediction equations, while the measurement update equations can be thought of as correction equations. If we assume Φ , H , \mathbf{Q} and \mathbf{R} to be constant, the time update equations are as follows:

$$\hat{\mathbf{x}}_k^- = \Phi\hat{\mathbf{x}}_{k-1}^- \quad (4.14)$$

$$\mathbf{P}_k^- = \Phi\mathbf{P}_{k-1}^-\Phi^T + \mathbf{Q} \quad (4.15)$$

while the measurement update equations are

$$K_k = \mathbf{P}_k^-H^T(H\mathbf{P}_k^-H^T + \mathbf{R})^{-1} \quad (4.16)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - H\hat{\mathbf{x}}_k^-) \quad (4.17)$$

$$\mathbf{P}_k = (\mathbf{I} - K_k H)\mathbf{P}_k^- \quad (4.18)$$

where \mathbf{P}_k^- and \mathbf{P}_k are the a priori and a posteriori estimate error covariance respectively and K_k is a factor known as the Kalman Gain.

Consequently in order to build a computational algorithm the following steps must be implemented:

- Known parameters.

1. Φ : State transition matrix.
 2. H : Measurement matrix.
 3. Q : Process noise covariance matrix.
 4. R : Measurement noise covariance matrix.
- Computations.
 1. Compute P_k^- by substituting P_{k-1} , Φ and Q in (4.15).
 2. Compute K_k by substituting P_k^- , H and R in (4.16).
 3. Compute P_k by substituting K_k and P_k^- in (4.18).
 4. Compute successive values of \hat{x}_k (which is the output of the filter) recursively using (4.14), (4.17) and the computed values of K_k . Start with the given initial estimates \hat{x}_0 and P_0 .

Figure 4.2 shows the flow diagram of the standard approach of the Kalman filter without sensor fusion.

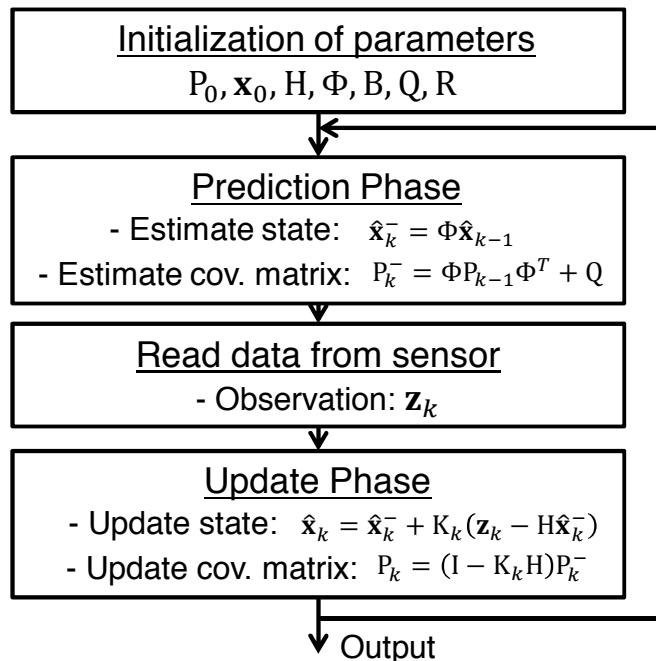


Figure 4.2: General diagram of the standard approach of the Kalman filter without sensor fusion.

4.1.3.2 The Kalman filter applied to orientation estimation

After this brief introduction to the basic theoretical fundamentals of the Kalman filter, we now explain how to employ it in a sensor fusion approach, and more

specifically how to apply it to fuse the information coming from the two orientation estimation approaches explained to the moment.

Let the state vector that we want to estimate be the following:

$$\mathbf{x}(t) = \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} \quad (4.19)$$

where $\alpha(t)$ is any of the orientation angles (roll, pitch or yaw), and $\text{bias}(t)$ is the difference between the measured angular rate and the actual angular rate for α .

$$\text{bias}(t) = \omega_{\text{meas}}(t) - \omega_{\text{actual}}(t) \quad (4.20)$$

Also, let $\mathbf{w}(t) = [w_\alpha(t) \ w_{\text{bias}}(t)]^T$, be the vector composed of the angle estimation noise and the angular rate bias noise respectively. The observation of the process, $\mathbf{z}(t)$ is the estimation of the angle computed using the accelerometer-magnetometer approach.

$$\mathbf{z}(t) = z(t) = \alpha(t) \quad (4.21)$$

We now proceed to obtain the expressions of the derivatives of the state variables:

$$\frac{d\alpha(t)}{dt} = \omega_{\text{actual}}(t) \quad (4.22)$$

knowing that $\omega_{\text{actual}}(t) = \omega_{\text{meas}}(t) - \text{bias}(t)$, i.e. the measured angular rate presents an offset with respect to its actual value:

$$\frac{d\alpha(t)}{dt} = \omega_{\text{meas}}(t) - \text{bias}(t) \quad (4.23)$$

If we consider $\omega_{\text{meas}}(t)$ as a part of the process noise, $\omega_{\text{meas}}(t) = w_\alpha(t)$, we get:

$$\frac{d\alpha(t)}{dt} = -\text{bias}(t) + w_\alpha(t) \quad (4.24)$$

Thus, the first component of the process noise power, $w_\alpha(t)$ should be the variance of the angle signal computed with the accelerometer-magnetometer approach.

On the other hand, let the derivative of the second state variable, $\text{bias}(t)$, be the second component of the process noise:

$$\frac{dbias(t)}{dt} = w_{\text{bias}}(t) \quad (4.25)$$

Then, the model of the system is as follows:

$$\begin{aligned} \mathbf{x}(t) &= \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix}; \\ \frac{d\mathbf{x}(t)}{dt} &= \frac{d}{dt} \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_\alpha(t) \\ w_{\text{bias}}(t) \end{bmatrix}; \\ z(t) &= [1 \ 0] \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} + v(t) \end{aligned} \quad (4.26)$$

If we now apply the approximations to discretize the system we get:

$$\begin{aligned}\mathbf{x}_{k+1} &= \begin{bmatrix} \alpha_{k+1} \\ \text{bias}_{k+1} \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -dt \\ 0 & 0 \end{bmatrix} \right] \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} w_{\alpha,k} \\ w_{\text{bias},k} \end{bmatrix}; \\ \mathbf{z}_{k+1} &= [1 \ 0] \begin{bmatrix} \alpha_{k+1} \\ \text{bias}_{k+1} \end{bmatrix} + v_{k+1}\end{aligned}\quad (4.27)$$

or, equivalently:

$$\begin{aligned}\mathbf{x}_k &= \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} = \left[\begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{k-1} \\ \text{bias}_{k-1} \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} w_{\alpha,k-1} \\ w_{\text{bias},k-1} \end{bmatrix}; \\ \mathbf{z}_k &= [1 \ 0] \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} + v_k\end{aligned}\quad (4.28)$$

Then, identifying terms, we have:

$$\Phi = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix}, \quad H = [1 \ 0] \quad (4.29)$$

Once we have defined the state equations of the discrete system, we repeat the equations of the Kalman filter and their order of application for the sake of clarity:

PREDICTION EQUATIONS

$$\hat{\mathbf{x}}_k^- = \Phi \hat{\mathbf{x}}_{k-1} \quad (4.30)$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q \quad (4.31)$$

UPDATE EQUATIONS

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (4.32)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (z_k - H \hat{\mathbf{x}}_k^-) \quad (4.33)$$

$$P_k = (I - K_k H) P_k^- \quad (4.34)$$

where

- $\hat{\mathbf{x}}_k^-$ is the a priori state estimate.
- P_k^- is the a priori system covariance matrix.

- Q is the process noise covariance matrix and is defined as

$$Q = \begin{bmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\omega \end{bmatrix} \quad (4.35)$$

where σ_α is the variance of the angle computed with the accelerometer-magnetometer approach and σ_ω the variance of the angular rate measured by the gyroscope.

- R , which in this case is a scalar, is the power of the observation noise and is defined as the variance of the accelerometer-magnetometer angle estimation (σ_α).
- K_k is the Kalman filter gain.
- \hat{x}_k is the a posteriori state estimate.
- P_k is the a posteriori system covariance matrix.

The reader may be wondering where exactly in the equations we have used the measured angular rate. Truth is that equations (4.30)-(4.34) do not yet implement a sensor fusion approach as we are only using data coming from the accelerometer-magnetometer. The key of inserting sensor fusion to the system defined by these equations is to change the computation of the a priori prediction of the states \hat{x}_k^- as follows:

$$\hat{x}_k^- = \begin{bmatrix} \alpha_k^- \\ \text{bias}_k^- \end{bmatrix} = \begin{bmatrix} \alpha_{k-1} + (\omega_{\text{meas},k} - \text{bias}_{k-1}) dt \\ \text{bias}_{k-1} \end{bmatrix} \quad (4.36)$$

This way, the a priori state estimate is being computed using information from both accelerometer-magnetometer and angular rate approaches.

To end the explanation of the classical approach of the Kalman filter applied to fuse the information of the accelerometer-magnetometer and gyroscope, we include below the step by step operations that need to be carried out by the complete algorithm:

1. Initialize the covariance matrix as a 2×2 identity matrix, $P = I_{2 \times 2}$.
2. Define the covariance matrix using equation (4.35). This matrix remains constant during the execution of the algorithm.
3. Define the measurement noise variance, R , which in this case is the variance of the accelerometer-magnetometer angle estimations, $R = \sigma_\alpha$. This value remains constant during the execution of the algorithm.
4. Define the state transition and the measurement matrices, Φ and H using equation (4.29).

5. Read data from accelerometer (\mathbf{a}_k), gyroscope (ω_k) and magnetometer (\mathbf{h}_k).
6. Using \mathbf{a}_k and \mathbf{h}_k , compute the accelerometer-magnetometer angle estimation, α_k , by applying any of the non-fusion orientation estimation methods explained within the previous subsections.
7. Compute the a priori estimation of the state vector using equation (4.36).
8. Compute the a priori system covariance matrix using equation (4.31).
9. Compute the Kalman gain using equation (4.32), where $z_k = \alpha_k$.
10. Update the state vector using equation (4.33).
11. Update the system covariance matrix using equation (4.34).
12. Return to step 5.

Figure 4.3 shows the flow diagram of the standard approach of the Kalman filter with sensor fusion.

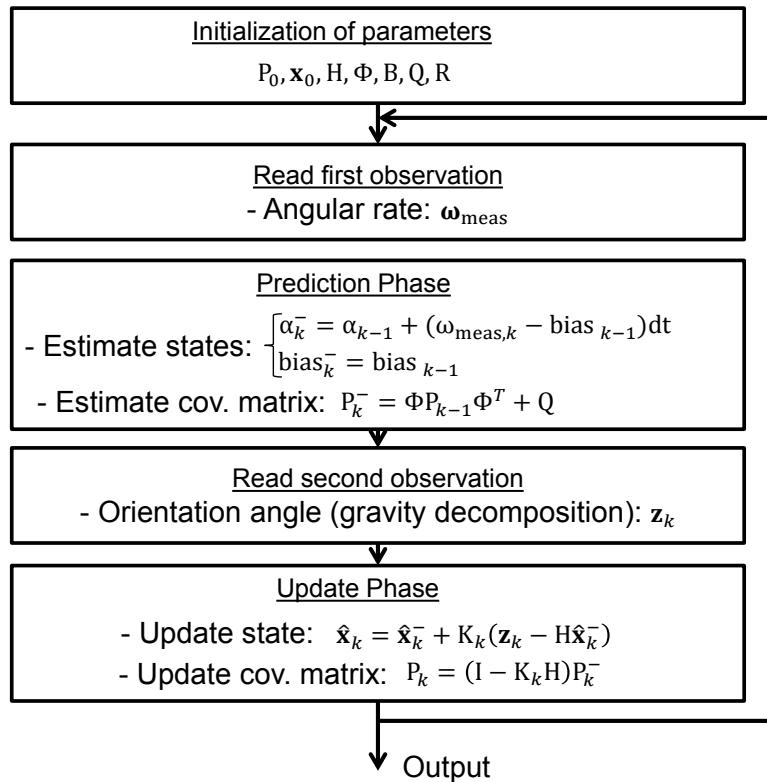


Figure 4.3: General diagram of the standard approach of the Kalman filter with sensor fusion.

4.2 Quaternion-based orientation determination

4.2.1 Using angular rate to compute orientation

A way to compute the rate of change of the orientation of the Earth frame relative to the sensor frame is to compute the quaternion derivative ${}^S_E \dot{\mathbf{q}}$, which is defined as follows,

$${}^S_E \dot{\mathbf{q}} = \frac{1}{2} {}^S_E \mathbf{q} \otimes {}^S \boldsymbol{\omega} \quad (4.37)$$

where ${}^S_E \mathbf{q}$ is the quaternion determining the orientation of the Earth frame relative to the sensor frame, ${}^S \boldsymbol{\omega} = [0 \ \omega_x \ \omega_y \ \omega_z]$ is the measured angular rate vector in the sensor frame in quaternion form, and \otimes is the quaternion (or Hamiltonian) product.

Therefore, we can compute the orientation of the Earth frame relative to the sensor frame (computed from the angular rate) at time t ${}^S_E \mathbf{q}_{\omega,t}$ by simply numerically integrating the quaternion derivative in equation (4.37).

$${}^S_E \mathbf{q}_{\omega,t} = {}^S_E \hat{\mathbf{q}}_{t-1} + {}^S_E \dot{\mathbf{q}}_{\omega,t} \Delta t \quad (4.38)$$

where ${}^S_E \hat{\mathbf{q}}_{t-1}$ is the orientation quaternion estimated in $t-1$ by applying a sensor fusion algorithm (this will be later explained), and Δt is the sampling frequency. So, if we plug equation (4.37) into (4.38), we get

$$\begin{aligned} {}^S_E \mathbf{q}_{\omega,t} &= [\hat{q}_{1,t-1} \ \hat{q}_{2,t-1} \ \hat{q}_{3,t-1} \ \hat{q}_{4,t-1}] + \frac{1}{2} \Delta t [\hat{q}_{1,t-1} \ \hat{q}_{2,t-1} \ \hat{q}_{3,t-1} \ \hat{q}_{4,t-1}] \otimes [0 \ \omega_x \ \omega_y \ \omega_z] = \\ &= [\hat{q}_{1,t-1} \ \hat{q}_{2,t-1} \ \hat{q}_{3,t-1} \ \hat{q}_{4,t-1}] + \frac{1}{2} \Delta t \left[\underbrace{-\hat{q}_{2,t-1} \omega_x - \hat{q}_{3,t-1} \omega_y - \hat{q}_{4,t-1} \omega_z}_{y_1'} \right. \\ &\quad \underbrace{\hat{q}_{1,t-1} \omega_x + \hat{q}_{3,t-1} \omega_z - \hat{q}_{4,t-1} \omega_y}_{y_2'} \underbrace{\hat{q}_{1,t-1} \omega_y - \hat{q}_{2,t-1} \omega_z + \hat{q}_{4,t-1} \omega_x}_{y_3'} \\ &\quad \left. \underbrace{\hat{q}_{1,t-1} \omega_z + \hat{q}_{2,t-1} \omega_y - \hat{q}_{3,t-1} \omega_x}_{y_4'} \right] \end{aligned} \quad (4.39)$$

4.2.2 Using acceleration and magnetic field to compute orientation

Another possible way to compute the orientation is by using the accelerometer and the magnetometer measurements. The accelerometer measures the magnitude and direction of the Earth's gravitational field in addition to linear acceleration. The magnetometer measures as well the magnitude and direction of the Earth's magnetic field together with local magnetic flux and field perturbations.

If the body is moving with quasi-static motion and the environment is magnetically stable, we can assume that the accelerometer will only measure the gravity and the magnetometer will only measure Earth's magnetic field. We will assume this in the following.

In order to obtain a complete 9-DOF orientation estimate we need to combine both the accelerometer and the magnetometer as the accelerometer is not able to measure the orientation around the Z axis as it is parallel to the gravitational field. In some applications it is only necessary to measure pitch and/or roll, so there is no need for a magnetometer. However, for a quaternion-based solution we need both sensors as it is a 9-DOF estimation.

Therefore, to estimate the complete orientation we can formulate an optimization problem where an orientation of the sensor ${}^S_E \dot{\mathbf{q}}$, is that which aligns a predefined reference direction of the field in the Earth frame, ${}^E \hat{\mathbf{d}}$, with the measured direction of the field in the sensor frame, ${}^S \hat{\mathbf{s}}$, using the quaternion rotation operation

$${}^E \mathbf{v} = {}_S^E \hat{\mathbf{q}} \otimes {}^S \mathbf{v} \otimes {}_S^E \hat{\mathbf{q}} \quad (4.40)$$

where ${}^E \mathbf{v}$ and ${}^S \mathbf{v}$ represent the vector \mathbf{v} measured in the Earth frame and the sensor frame respectively, and ${}^E_S \hat{\mathbf{q}}$ is the quaternion representing the orientation of the sensor frame with respect to the Earth frame.

Hence, ${}^E_S \hat{\mathbf{q}}$ can be found as the solution of

$$\min_{\substack{{}^E_S \hat{\mathbf{q}} \in \mathbb{R}^4}} f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}}) \quad (4.41)$$

where

$$f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}}) = {}_S^E \hat{\mathbf{q}}^* \otimes {}^E \hat{\mathbf{d}} \otimes {}_S^E \hat{\mathbf{q}} - {}^S \hat{\mathbf{s}} \quad (4.42)$$

In this case, we will use the gradient descent algorithm to optimize the function. The orientation quaternion at instant k+1 is computed applying

$${}^E_S \hat{\mathbf{q}}_{k+1} = {}^E_S \hat{\mathbf{q}}_k - \mu \frac{\nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}})}{\| \nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}}) \|} \quad (4.43)$$

where

$$\nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}}) = J^T({}^E_S \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{d}}) f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{s}}) \quad (4.44)$$

Where, in turn,

$${}^E_S \hat{\mathbf{q}}^* \otimes {}^E \hat{\mathbf{d}} \otimes {}^E_S \hat{\mathbf{q}} - {}^S \hat{\mathbf{s}} = \begin{bmatrix} 2d_x\left(\frac{1}{2} - q_3^2 - q_4^2\right) + 2d_x + 2d_y(q_1q_4 + q_2q_3) + 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y\left(\frac{1}{2} - q_2^2 - q_4^2\right) + 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + 2d_z\left(\frac{1}{2} - q_2^2 - q_3^2\right) - s_z \end{bmatrix} \quad (4.45)$$

As we aforementioned,

- ${}^E_S \hat{\mathbf{q}} = [q_1 \ q_2 \ q_3 \ q_4] \rightarrow$ Estimated orientation quaternion.
- ${}^E \hat{\mathbf{d}} = [0 \ d_x \ d_y \ d_z] \rightarrow$ Predefined reference direction of the field in the Earth frame.
- ${}^S \hat{\mathbf{s}} = [0 \ s_x \ s_y \ s_z] \rightarrow$ Measured direction of the field in the sensor frame.

In our case, we have two reference vectors, the Earth's gravitational and magnetic field vectors. Therefore, we will use both to have a complete 9DOF orientation estimation. We will start by deriving the equations for the Earth's gravitational field and the accelerometer. For this case, ${}^E \hat{\mathbf{d}} = {}^E \hat{\mathbf{g}} = [0 \ 0 \ 0 \ 1]$ and ${}^S \hat{\mathbf{s}} = {}^S \hat{\mathbf{a}} = [0 \ a_x \ a_y \ a_z]$. Then, equation (4.45) is reduced to,

$$f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}, {}^S \hat{\mathbf{a}}) = {}^E_S \hat{\mathbf{q}}^* \otimes {}^E \hat{\mathbf{g}} \otimes {}^E_S \hat{\mathbf{q}} - {}^S \hat{\mathbf{a}} = \begin{bmatrix} \underbrace{2(q_2q_4 - q_1q_3) - a_x}_{F_1} \\ \underbrace{2(q_1q_2 + q_3q_4) - a_y}_{F_2} \\ \underbrace{2\left(\frac{1}{2} - q_2^2 - q_3^2\right) - a_z}_{F_3} \end{bmatrix} \quad (4.46)$$

As it was shown in equation (4.44),

$$\nabla f \left({}_S^E \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}, {}^S \hat{\mathbf{a}} \right) = J^T \left({}_S^E \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{g}} \right) f \left({}_S^E \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}, {}^S \hat{\mathbf{a}} \right) \quad (4.47)$$

Hence, we are only missing the computation of the Jacobian $J \left({}_S^E \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{g}} \right)$, which is carried out by applying the following equation,

$$J \left({}_S^E \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{g}} \right) = \begin{bmatrix} \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial q_2} & \frac{\partial F_1}{\partial q_3} & \frac{\partial F_1}{\partial q_4} \\ \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial q_2} & \frac{\partial F_2}{\partial q_3} & \frac{\partial F_2}{\partial q_4} \\ \frac{\partial F_3}{\partial q_1} & \frac{\partial F_3}{\partial q_2} & \frac{\partial F_3}{\partial q_3} & \frac{\partial F_3}{\partial q_4} \end{bmatrix} = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (4.48)$$

We are now done with the accelerometer part, let's continue with the magnetometer which is going to provide the Yaw angle which can not be computed with the accelerometer. For the magnetometer, ${}^E \hat{\mathbf{d}} = {}^E \hat{\mathbf{b}} = [0 b_x 0 b_z]$ and ${}^S \hat{\mathbf{s}} = {}^S \hat{\mathbf{h}} = [0 h_x h_y h_z]$.

$$f \left({}_S^E \hat{\mathbf{q}}, {}^E \hat{\mathbf{b}}, {}^S \hat{\mathbf{h}} \right) = {}_S^E \hat{\mathbf{q}}^* \otimes {}^E \hat{\mathbf{b}} \otimes {}_S^E \hat{\mathbf{q}} - {}^S \hat{\mathbf{h}} = \begin{bmatrix} \underbrace{2b_x \left(\frac{1}{2} - q_3^2 - q_4^2 \right) + 2b_z (q_2 q_4 - q_1 q_3) - h_x}_{F_4} \\ \underbrace{2b_x (q_2 q_3 - q_1 q_4) + 2b_z (q_1 q_2 + q_3 q_4) - h_y}_{F_5} \\ \underbrace{2b_x (q_1 q_3 + q_2 q_4) + 2b_z \left(\frac{1}{2} - q_2^2 - q_3^2 \right) - h_z}_{F_6} \end{bmatrix} \quad (4.49)$$

Analogously as in equation (4.48),

$$J \left({}_S^E \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{b}} \right) = \begin{bmatrix} \frac{\partial F_4}{\partial q_1} & \frac{\partial F_4}{\partial q_2} & \frac{\partial F_4}{\partial q_3} & \frac{\partial F_4}{\partial q_4} \\ \frac{\partial F_5}{\partial q_1} & \frac{\partial F_5}{\partial q_2} & \frac{\partial F_5}{\partial q_3} & \frac{\partial F_5}{\partial q_4} \\ \frac{\partial F_6}{\partial q_1} & \frac{\partial F_6}{\partial q_2} & \frac{\partial F_6}{\partial q_3} & \frac{\partial F_6}{\partial q_4} \end{bmatrix} = \begin{bmatrix} -2b_z q_3 & 2b_z q_4 & -2b_z q_1 & 2b_z q_2 \\ -2b_x q_4 + 2b_z q_2 & 2b_x q_3 + 2b_z q_1 & 2b_x q_2 + 2b_z q_4 & -2b_x q_1 + 2b_z q_3 \\ 2b_x q_3 & 2b_x q_4 - 4b_z q_2 & 2b_x q_1 - 4b_z q_3 & 2b_x q_2 \end{bmatrix} \quad (4.50)$$

So, finally, if we put everything together, these are the equations that need to be implemented to compute the orientation quaternion using accelerometer and magnetometer data,

$${}^E_S \hat{\mathbf{q}}_{k+1} = {}^E_S \hat{\mathbf{q}}_k - \mu \frac{\nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}\mathbf{b}, {}^S \hat{\mathbf{a}}\mathbf{h})}{\| \nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}\mathbf{b}, {}^S \hat{\mathbf{a}}\mathbf{h}) \|} \quad (4.51)$$

$$\nabla f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}\mathbf{b}, {}^S \hat{\mathbf{a}}\mathbf{h}) = J^T({}^E_S \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{g}}\mathbf{b}) f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{a}}\mathbf{h}, {}^S \hat{\mathbf{s}}) \quad (4.52)$$

$$f({}^E_S \hat{\mathbf{q}}, {}^E \hat{\mathbf{a}}\mathbf{h}, {}^S \hat{\mathbf{s}}) = \begin{bmatrix} 2(q_2 q_4 - q_1 q_3) - a_x \\ 2(q_1 q_2 + q_3 q_4) - a_y \\ 2\left(\frac{1}{2} - q_2^2 - q_3^2\right) - a_z \\ 2b_x\left(\frac{1}{2} - q_3^2 - q_4^2\right) + 2b_z(q_2 q_4 - q_1 q_3) - h_x \\ 2b_x(q_2 q_3 - q_1 q_4) + 2b_z(q_1 q_2 + q_3 q_4) - h_y \\ 2b_x(q_1 q_3 + q_2 q_4) + 2b_z\left(\frac{1}{2} - q_2^2 - q_3^2\right) - h_z \end{bmatrix} \quad (4.53)$$

$$J({}^E_S \hat{\mathbf{q}}_k, {}^E \hat{\mathbf{g}}\mathbf{b}) = \begin{bmatrix} \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial q_2} & \frac{\partial F_1}{\partial q_3} & \frac{\partial F_1}{\partial q_4} \\ \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial q_2} & \frac{\partial F_2}{\partial q_3} & \frac{\partial F_2}{\partial q_4} \\ \frac{\partial F_3}{\partial q_1} & \frac{\partial F_3}{\partial q_2} & \frac{\partial F_3}{\partial q_3} & \frac{\partial F_3}{\partial q_4} \\ \frac{\partial F_4}{\partial q_1} & \frac{\partial F_4}{\partial q_2} & \frac{\partial F_4}{\partial q_3} & \frac{\partial F_4}{\partial q_4} \\ \frac{\partial F_5}{\partial q_1} & \frac{\partial F_5}{\partial q_2} & \frac{\partial F_5}{\partial q_3} & \frac{\partial F_5}{\partial q_4} \\ \frac{\partial F_6}{\partial q_1} & \frac{\partial F_6}{\partial q_2} & \frac{\partial F_6}{\partial q_3} & \frac{\partial F_6}{\partial q_4} \end{bmatrix} =$$

$$= \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \\ -2b_z q_3 & 2b_z q_4 & -2b_z q_1 & 2b_z q_2 \\ -2b_x q_4 + 2b_z q_2 & 2b_x q_3 + 2b_z q_1 & 2b_x q_2 + 2b_z q_4 & -2b_x q_1 + 2b_z q_3 \\ 2b_x q_3 & 2b_x q_4 - 4b_z q_2 & 2b_x q_1 - 4b_z q_3 & 2b_x q_2 \end{bmatrix} \quad (4.54)$$

The routine implementing the process above should be structured as follows for every instant k ,

- (i) Read the orientation quaternion for instant $k-1$, ${}^E_S \hat{\mathbf{q}}_{k-1}$.
- (ii) Read the data from the accelerometer and the magnetometer for instant k . We then have $a_{x,k}$ $a_{y,k}$ $a_{z,k}$ $h_{x,k}$ $h_{y,k}$ and $h_{z,k}$.

(iii) Compute the Jacobian in equation (4.54) knowing that,

$$\mathbf{m} = {}_S^E \hat{\mathbf{q}}_{k-1} \otimes [0 \ h_{x,k} \ h_{y,k} \ h_{z,k}] \otimes {}_S^E \hat{\mathbf{q}}_{k-1}^* \quad (4.55)$$

$$b_x = \sqrt{m_2^2 + m_3^2} \quad (4.56)$$

$$b_z = m_4 \quad (4.57)$$

(iv) Compute the expression of $f({}_S^E \hat{\mathbf{q}}, {}^E \hat{\mathbf{a}}, {}^S \hat{\mathbf{s}})$ in (4.53).

(v) Substitute the value of the expressions computed in the two previous steps in equation (4.51) to compute the orientation quaternion for instant k , ${}_S^E \hat{\mathbf{q}}_k$.

4.2.3 Fusing the orientation quaternions computed with the angular rate and the acceleration and magnetic field

In the previous two sections we have explained how to compute the orientation quaternion using two different approaches. The first one is based on the numerical integration of the angular rate. This approach would be perfect in an ideal situation in which there was no noise at the output of the gyroscope and no bias time drift. The second approach is based on the computation of the orientation of the gravity and the Earth's magnetic field vectors measured in the sensor frame with respect to their known reference value in the Earth's frame. This approach, as it was aforementioned is only valid if the body is measuring with negligible linear acceleration (i.e. low intensity motion) and/or there are not magnetic distortions nearby the sensors.

It is clear then, that none of the two approaches can be used solely to compute an accurate orientation estimate. Therefore, we can apply a fusion scheme to fuse both approaches. The Extended Kalman Filter is a well known solution to carry out the sensor fusion.

4.2.3.1 The Extended Kalman Filter

The Extended Kalman Filter is used when the process to be estimated and (or) the measurement relationship to the process is non-linear. A solution to the non-linearity is to linearize about the current mean and covariance. Let us assume that our process again has a state vector \mathbb{R}^n , but that the process is now governed by the non-linear stochastic difference equation,

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (4.58)$$

with a measurement $z \in \mathbb{R}^m$,

$$z_k = h(x_k, v_k) \quad (4.59)$$

where the random variables w_k and v_k again represent the process and measurement noise. In this case, the non-linear function f in the difference equation (4.58) relates the state at the previous time step $k - 1$ to the state at the current time step k . It includes as parameters any driving function u_{k-1} and the zero-mean process noise w_k . The non-linear function h in the measurement equation (??) relates the state x_k to the measurement z_k .

In practice one does not know the individual values of the noise w_k and v_k at each time step. However, one can approximate the state and measurement vector without them as,

$$\bar{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (4.60)$$

and

$$\bar{z}_k = h(\bar{x}_k, 0) \quad (4.61)$$

where \hat{x}_k is some a posteriori estimate of the state from the previous time step k .

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that linearize an estimate about (4.60) and (4.61),

$$x_k \approx \bar{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k_1} \quad (4.62)$$

$$z_k \approx \bar{z}_k + H(x_k - \bar{x}_k) + V v_k \quad (4.63)$$

where,

- x_k and z_k are the actual state and measurement vectors.
- \bar{x}_k and \bar{z}_k are the approximate state and measurement vectors from (4.60) and (4.61).
- The random variables w_k and v_k represent the process and measurement noise as in (4.58) and (4.61).

- A is the Jacobian matrix of partial derivatives of f with respect to x , that is,

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}} (\hat{x}_{k-1}, u_{k-1}, 0) \quad (4.64)$$

- W is the Jacobian matrix of partial derivatives of f with respect to w ,

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}} (\hat{x}_{k-1}, u_{k-1}, 0) \quad (4.65)$$

- H is the Jacobian matrix of partial derivatives of h with respect to x ,

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}} (\bar{x}_k, 0) \quad (4.66)$$

- V is the Jacobian of partial derivatives of h with respect to v ,

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}} (\bar{x}_k, 0) \quad (4.67)$$

We can now define a new notation for the prediction error,

$$\bar{e}_{x_k} \equiv x_k - \bar{x}_k \quad (4.68)$$

and the measurement residual,

$$\bar{e}_{z_k} \equiv z_k - \bar{z}_k \quad (4.69)$$

We can redefine these equations as,

$$\bar{e}_{x_k} \equiv A(x_{k-1} - \hat{x}_{k-1}) + \epsilon_k \quad (4.70)$$

$$\bar{e}_{z_k} \equiv H\bar{e}_{x_k} + \eta_k \quad (4.71)$$

where ϵ_k and η_k represent new independent random variables having zero mean and covariance matrices QWW^T and VRV^T , with Q and R equal to the ones used in the standard Kalman filter.

In the end, we can compute the state estimate by applying the following equation,

$$\hat{x}_k = \bar{x}_k + K_k (z_k - \bar{z}_k) \quad (4.72)$$

The complete set of EKF equations is shown below. Note that \bar{x}_k has been substituted by \hat{x}_k^- to remain consistent with the notation used in the standard Kalman Filter definition, and that the Jacobians A, W, H and V have the subscript k to reinforce the notion that they are different at each time step.

- Time update equations:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (4.73)$$

$$\hat{P}_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (4.74)$$

- Measurement update equations:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R_k V_k^T \right)^{-1} \quad (4.75)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (4.76)$$

$$P_k = (I - K_k H_k) P_k^- \quad (4.77)$$

4.2.3.2 The Extended Kalman Filter applied to orientation estimation

The quaternion fusion is done by first defining the equations governing the process that we want to estimate. In our case, the state vector will be ${}^S_E \mathbf{q}_t = [q_1 q_2 q_3 q_4]$ and the function f is defined by equation 4.78,

$$\begin{aligned} {}^S_E \mathbf{q}_{\omega, k} &= [\hat{q}_{1,k-1} \hat{q}_{2,k-1} \hat{q}_{3,k-1} \hat{q}_{4,k-1}] + \frac{1}{2} \Delta t [\hat{q}_{1,k-1} \hat{q}_{2,k-1} \hat{q}_{3,k-1} \hat{q}_{4,k-1}] \otimes [0 \ \omega_x \ \omega_y \ \omega_z] = \\ &= [\hat{q}_{1,k-1} \hat{q}_{2,k-1} \hat{q}_{3,k-1} \hat{q}_{4,k-1}] + \frac{1}{2} \Delta t \underbrace{[-\hat{q}_{2,k-1} \omega_x - \hat{q}_{3,k-1} \omega_y - \hat{q}_{4,k-1} \omega_z]}_{y_1'} \\ &\quad \underbrace{\hat{q}_{1,k-1} \omega_x + \hat{q}_{3,k-1} \omega_z - \hat{q}_{4,k-1} \omega_y}_{y_2'} \underbrace{\hat{q}_{1,k-1} \omega_y - \hat{q}_{2,k-1} \omega_z + \hat{q}_{4,k-1} \omega_x}_{y_3'} \\ &\quad \underbrace{\hat{q}_{1,k-1} \omega_z + \hat{q}_{2,k-1} \omega_y - \hat{q}_{3,k-1} \omega_x}_{y_4'} \end{aligned} \quad (4.78)$$

$$\begin{aligned}
 y_1 &= \hat{q}_{1,k-1} + y'_1 = y_1 = \hat{q}_{1,k-1} - \hat{q}_{2,k-1}\omega_x - \hat{q}_{3,k-1}\omega_y - \hat{q}_{4,k-1}\omega_z & (4.79) \\
 y_2 &= \hat{q}_{2,k-1} + y'_2 = \hat{q}_{2,k-1} + \hat{q}_{1,k-1}\omega_x + \hat{q}_{3,k-1}\omega_z - \hat{q}_{4,k-1}\omega_y \\
 y_3 &= \hat{q}_{3,k-1} + y'_3 = \hat{q}_{3,k-1} + \hat{q}_{1,k-1}\omega_y - \hat{q}_{2,k-1}\omega_z + \hat{q}_{4,k-1}\omega_x \\
 y_4 &= \hat{q}_{4,k-1} + y'_4 = \hat{q}_{4,k-1} + \hat{q}_{1,k-1}\omega_z + \hat{q}_{2,k-1}\omega_y - \hat{q}_{3,k-1}\omega_x
 \end{aligned}$$

The measurement is given by the orientation quaternion computed using the acceleration and the magnetic field. Therefore, since the state vector is also composed of the orientation quaternion, the relationship between the state and the measurement is linear and the matrix H is simply a 4×4 identity matrix.

The matrix A_k is computed applying equation (4.64) which gives,

$$A_k = \left[\begin{array}{cccc} \frac{\partial y_{1,k}}{\partial} & \frac{\partial y_{1,k}}{\partial} & \frac{\partial y_{1,k}}{\partial} & \frac{\partial y_{1,k}}{\partial} \end{array} \right] \quad (4.80)$$

Chapter 5

Experiments

5.1 Qualisys optical motion tracker

The Qualisys optical motion tracking system uses high speed digital cameras to capture the motion of a measurement object with passive or active markers attached. The technology is precise and robust and delivers high quality data to the observer in real-time. The provided software tools allows to perform basic motion calculations, such as speed, acceleration, rotation and angle, as well as other more complex calculations. The system has eight cameras which are distributed around a room. The precision of this system allows its use as a reference system to evaluate portable motion tracking systems such as the GaitWatch.



Figure 5.1: An Oqus camera part of the Qualisys system.

5.1.1 Computing Euler angles using Qualisys data

If the subject is wearing two markers per segment, then the pitch angle of such segment is computed between the vector defined by the upper and lower markers

and the vector normal to the Earth's surface. To be able to compute it we first have to define a third point which has the same X coordinate as the lower marker and the same Z coordinate as the upper marker. This will define a right triangle in which one of the contiguous cathetus is normal to the Earth's surface and the hypotenuse is defined by the line between the upper and the lower point. Therefore, by calculating the arctangent we can easily find the angle of the right triangle, which is, in turn, the pitch angle. Figure 5.2 shows this process.

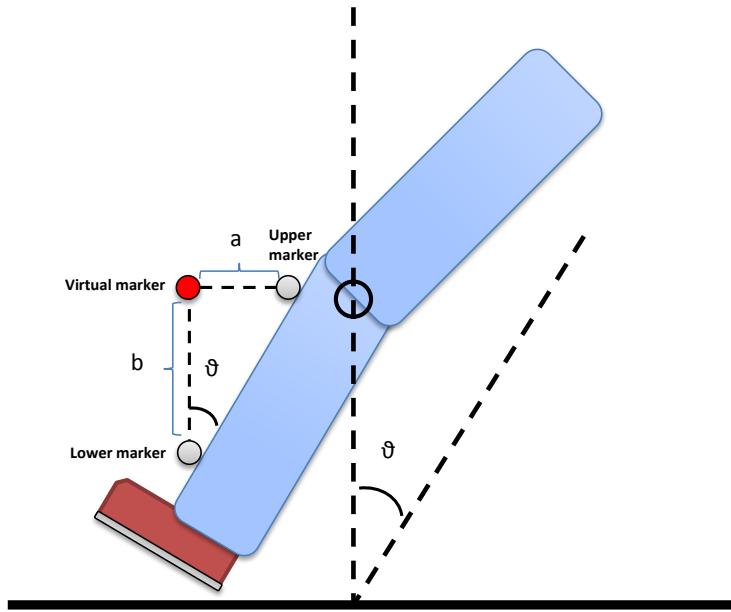


Figure 5.2: Diagram of the pitch computation using the Qualisys system.

Therefore, the pitch angle is computed applying,

$$\theta_{QS} = \arctan\left(\frac{a}{b}\right) \quad (5.1)$$

$$a = \sqrt{(x_{upper} - x_{lower})^2 + (z_{upper} - z_{lower})^2} \quad (5.2)$$

$$b = \sqrt{(x_{lower} - x_{lower})^2 + (z_{lower} - z_{upper})^2} \quad (5.3)$$

where $[x_{lower}, z_{lower}]$ and $[x_{upper}, z_{upper}]$ are the coordinates of the projections of the lower and upper markers in the XZ plane, respectively.

5.2 Qualisys validation experiments

5.2.1 Data gathering protocol

- Switch on the computer.
- Switch on the Qualisys system.
- Start the computer and select the "Tracking" account.
- Write any password, the system will say that the password is wrong and it will show the secret question. The answer to the secret question is "tracking".
- Once the session has started, open the "Qualisys Track Manager".
- Place the calibration frame (which is in the big metallic box and has an L shape) on the floor.
- Press the red button and check that the maximum number of cameras see the four markers which are attached to the calibration frame. You may need to adjust the position of the cameras. The cameras also have a small display which show the ID of the camera and the number of markers which are seen in each instant.
- When all the cameras are ready, go to "capture→calibrate". Select the desired delay and the desired calibration time (30 seconds should be fine).
- Grab the calibration wand (the device which is shaped as a T), click the button to start the calibration and move around the room moving the wand trying to cover all the space.
- When the calibration is done, hide both the calibration frame and the calibration wand somewhere where the markers are not seen by the cameras (you may need to put them back into the big metallic box or just put them out of the room).
- The system is now ready to start the measurements.
- Place the GaitWatch system on the subject and place the markers on the desired segments of the subject.
- Press the red button (new capture) and complete the required fields: initial delay, number of samples (200 by default), duration and name of the file.
- Before starting the capture, switch on the GaitWatch system and press the button to start the measurement. Tell the subject to stay static.

- Go back to the computer, start the capture and write down the current time together with the name you have given to the file.
- The subject may now move around the room and carry out the movements he is instructed to.
- When the capture is done, press the GaitWatch button again to stop the measurement.
- Go back to the computer and label one by one all the markers that were identified by the program. This is done by clicking on the markers, going to "identify→label".
- Once all the markers are identified, save the file and go to "file→export→MAT", then click OK. This will export all the data to a .mat file. Be careful because the .mat file is created when we start a new capture but it will remain empty until the exportation.

The next step would be to download the GaitWatch data to the computer. To do so we use the *GW.comm.m* routine. This routine will also ask the user to give each one of the files a number. Be sure to give the file the same number which was given to the Qualisys file. Once all the data are identified, we can run the *param_optimization.m* routine which loads both the GaitWatch and Qualisys data, synchronizes them and computes the optimal parameters of the algorithm used to compute the orientation using the GaitWatch data.

The following sections contain a set of figures depicting the pitch angle of different segments computed using both the Qualisys and the GaitWatch systems.

5.3 Fast Walking

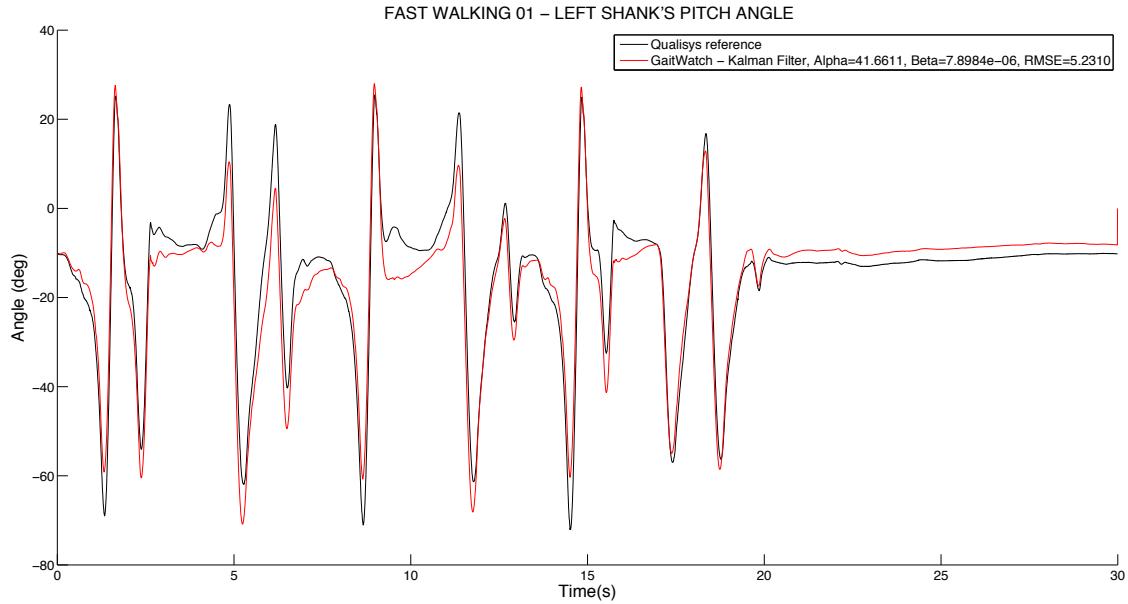


Figure 5.3: Pitch angle of left shank measured while walking fast (GaitWatch vs. Qualisys).

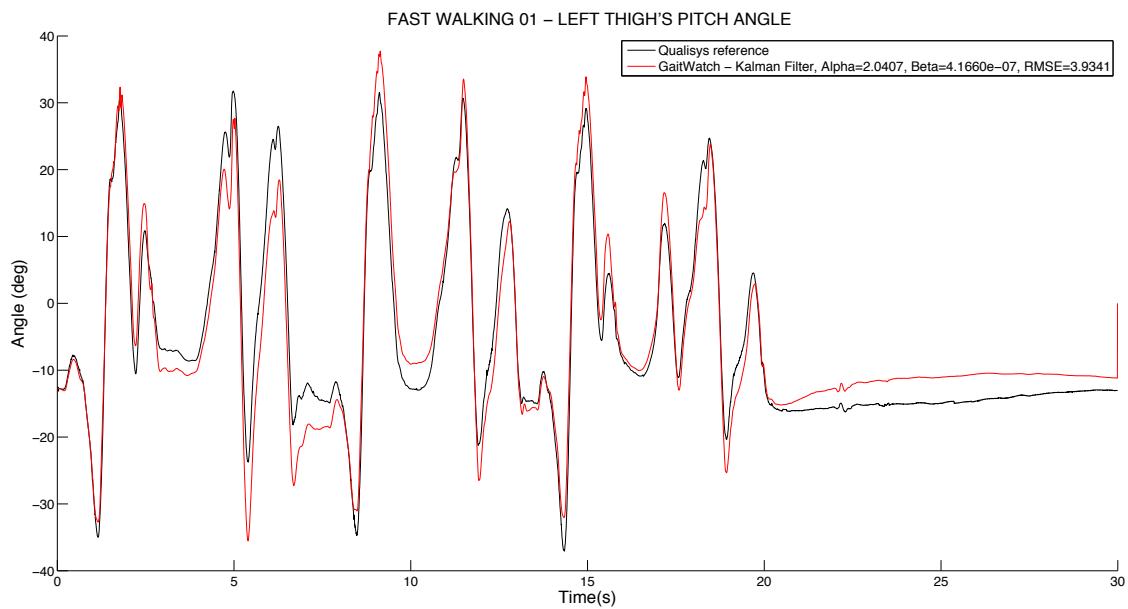


Figure 5.4: Pitch angle of left thigh measured while walking fast (GaitWatch vs. Qualisys).

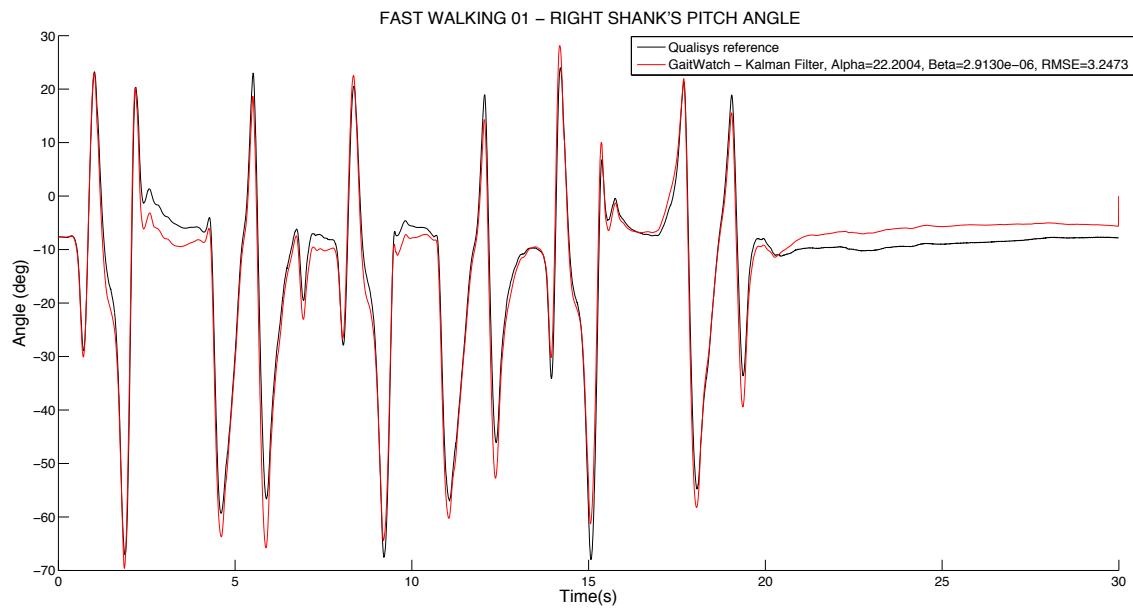


Figure 5.5: Pitch angle of right shank measured while walking fast (GaitWatch vs. Qualisys).

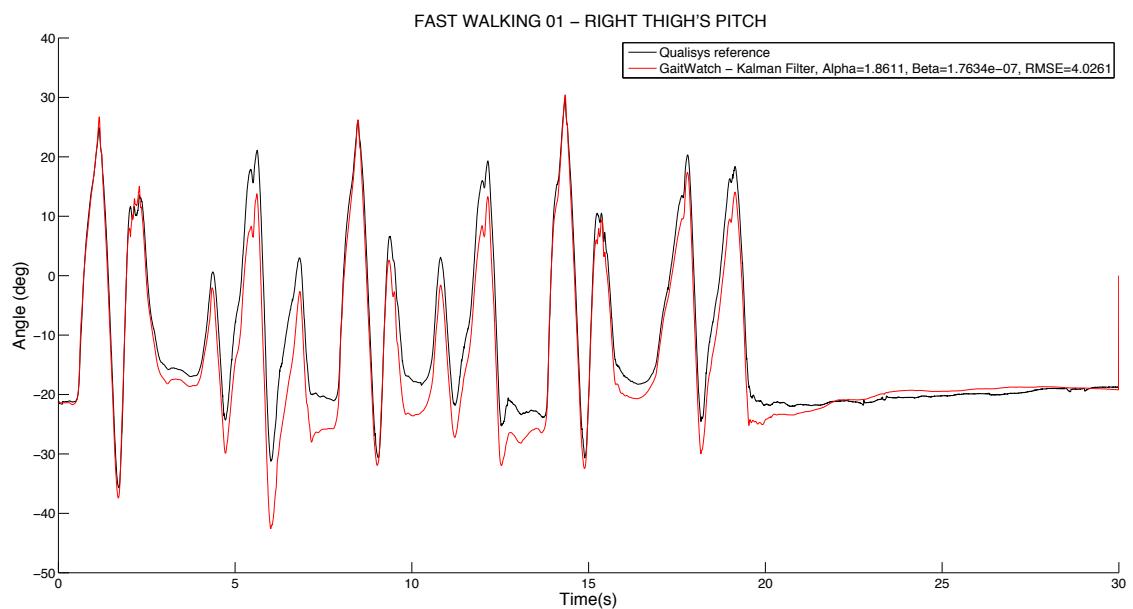


Figure 5.6: Pitch angle of right thigh measured while walking fast (GaitWatch vs. Qualisys).

5.4 Jumping

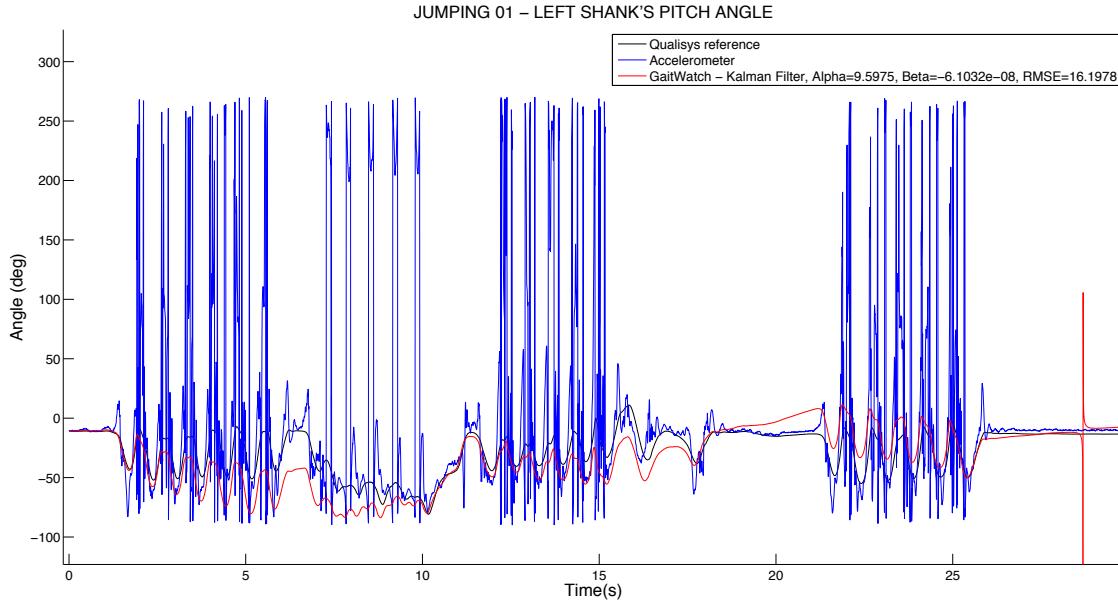


Figure 5.7: Pitch angle of left shank measured while jumping (GaitWatch vs. Qualisys).

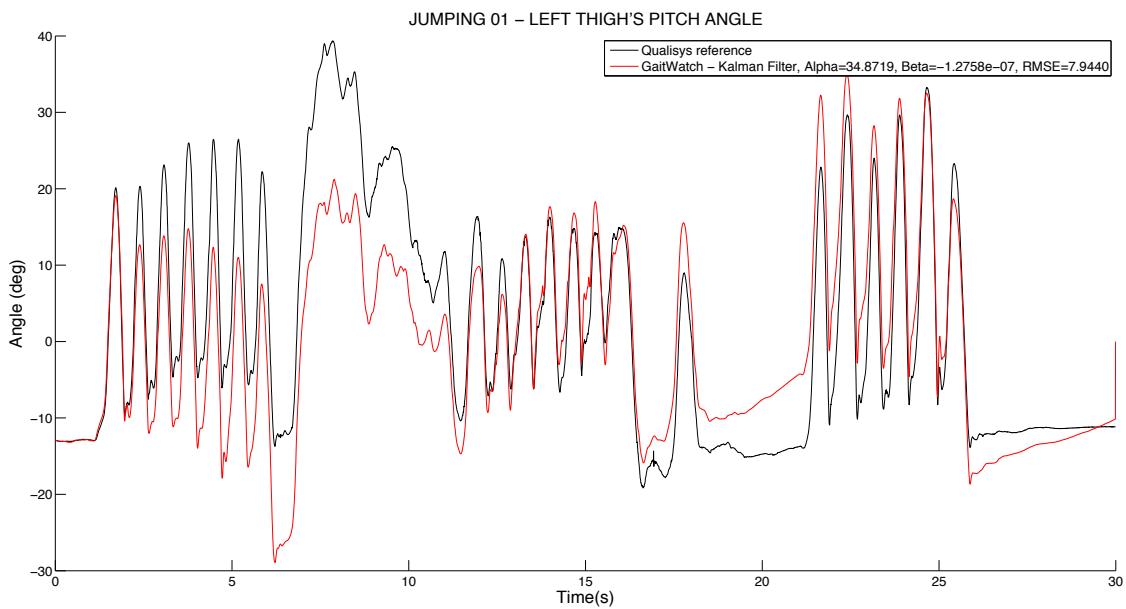


Figure 5.8: Pitch angle of left thigh measured while jumping (GaitWatch vs. Qualisys).

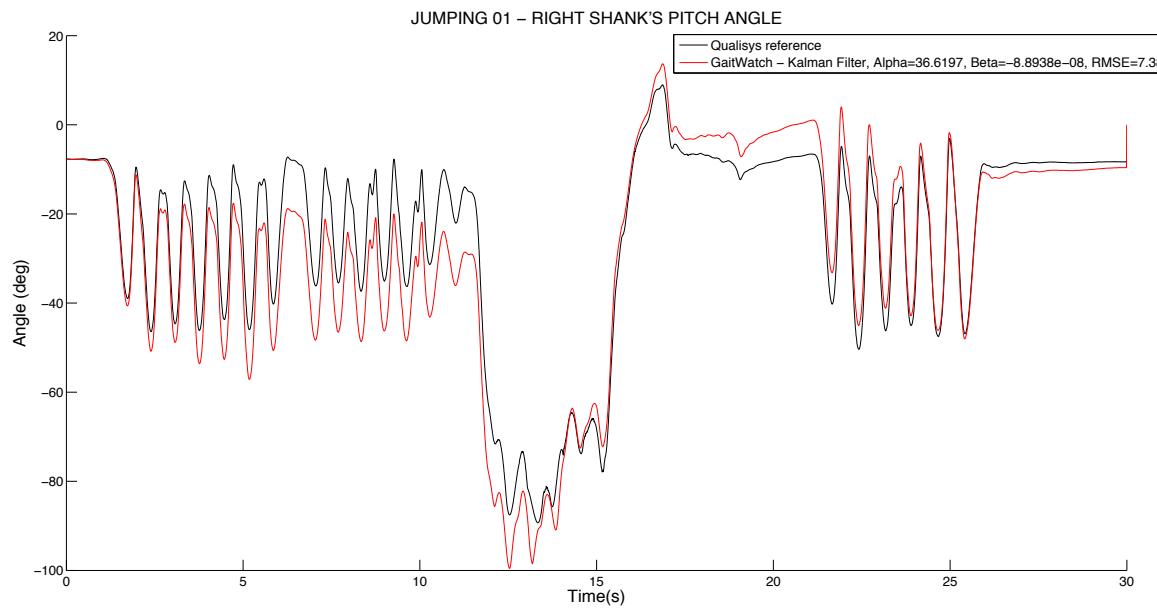


Figure 5.9: Pitch angle of right shank measured while jumping (GaitWatch vs. Qualisys).

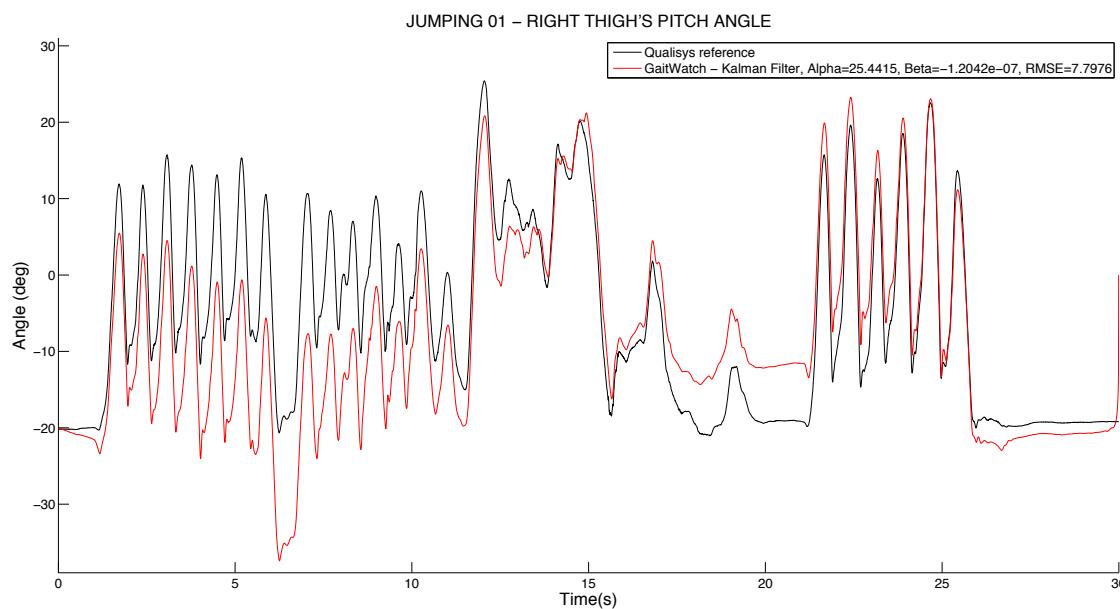


Figure 5.10: Pitch angle of right thigh measured while jumping (GaitWatch vs. Qualisys).

5.5 Leg motion

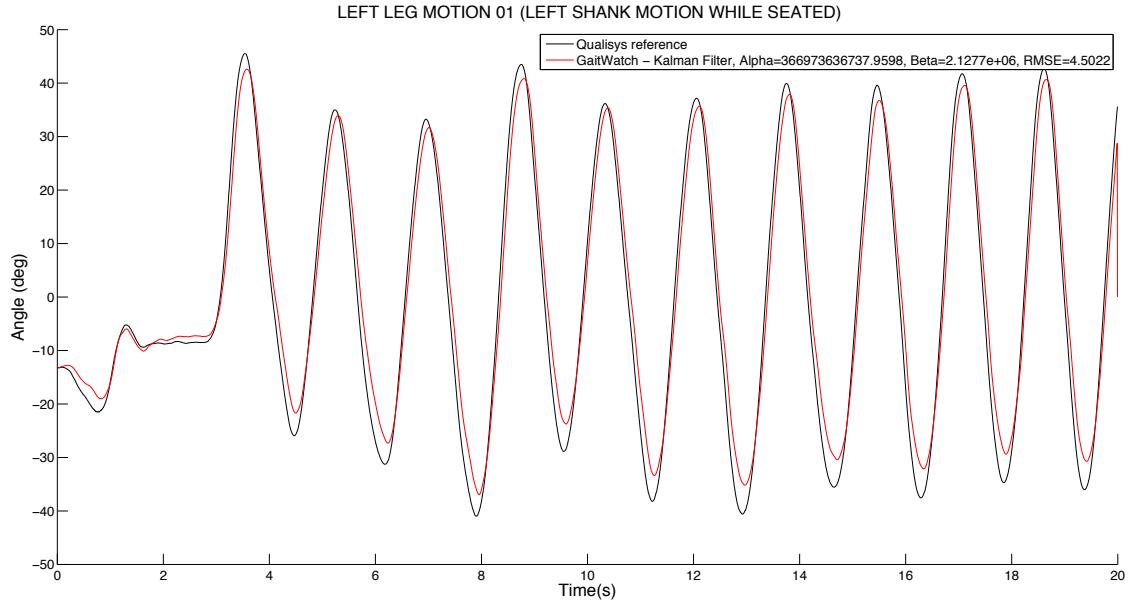


Figure 5.11: Pitch angle while swinging left shank. (GaitWatch vs. Qualisys).

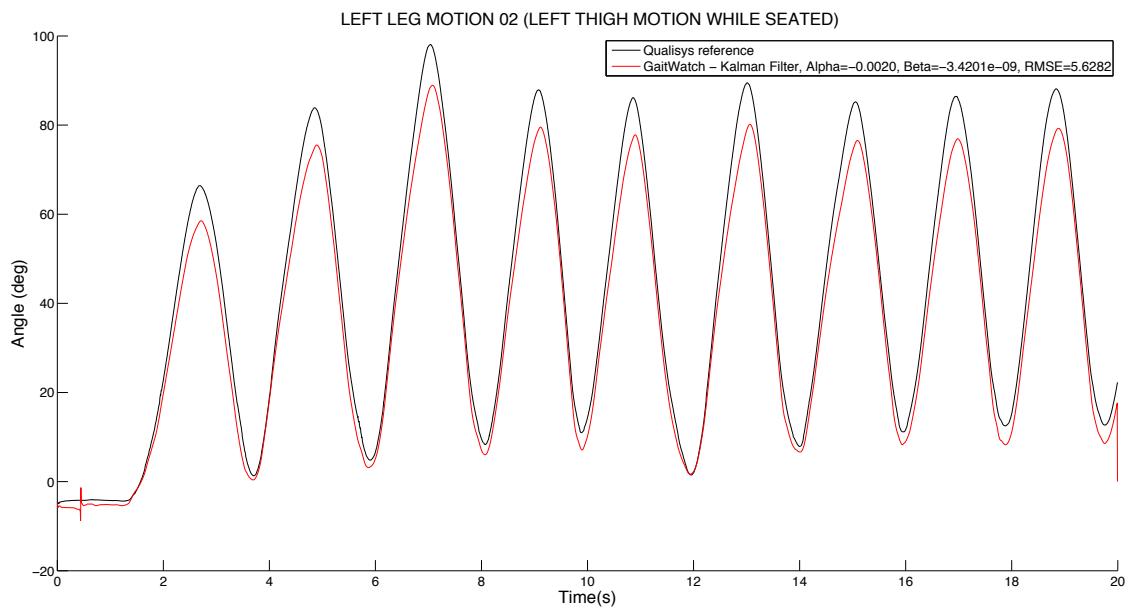


Figure 5.12: Pitch angle while swinging left thigh. (GaitWatch vs. Qualisys).

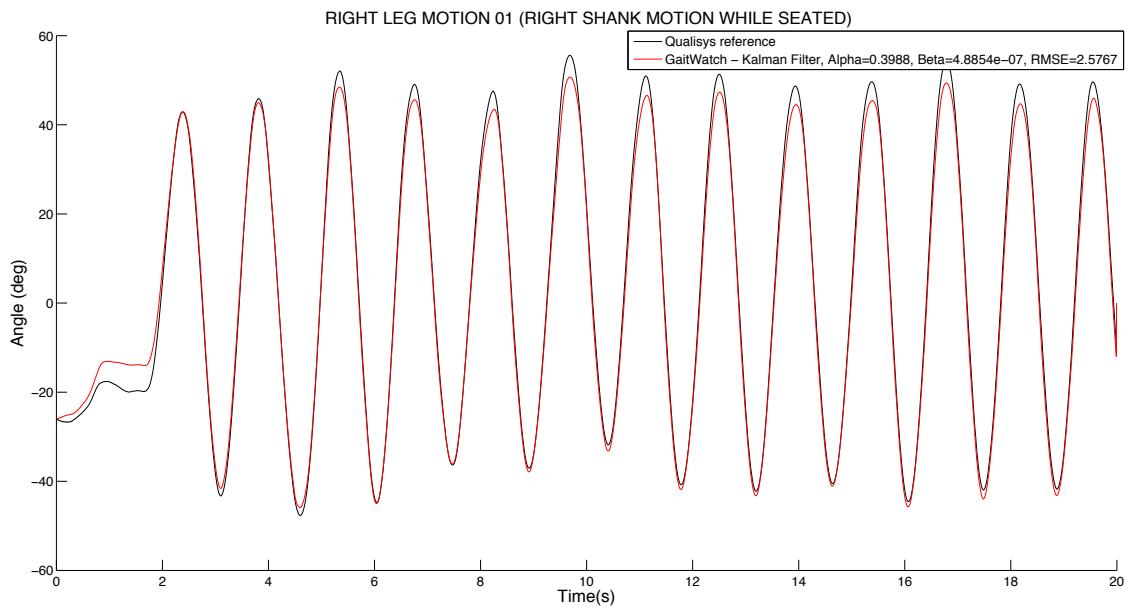


Figure 5.13: Pitch angle while swinging right shank. (GaitWatch vs. Qualisys).

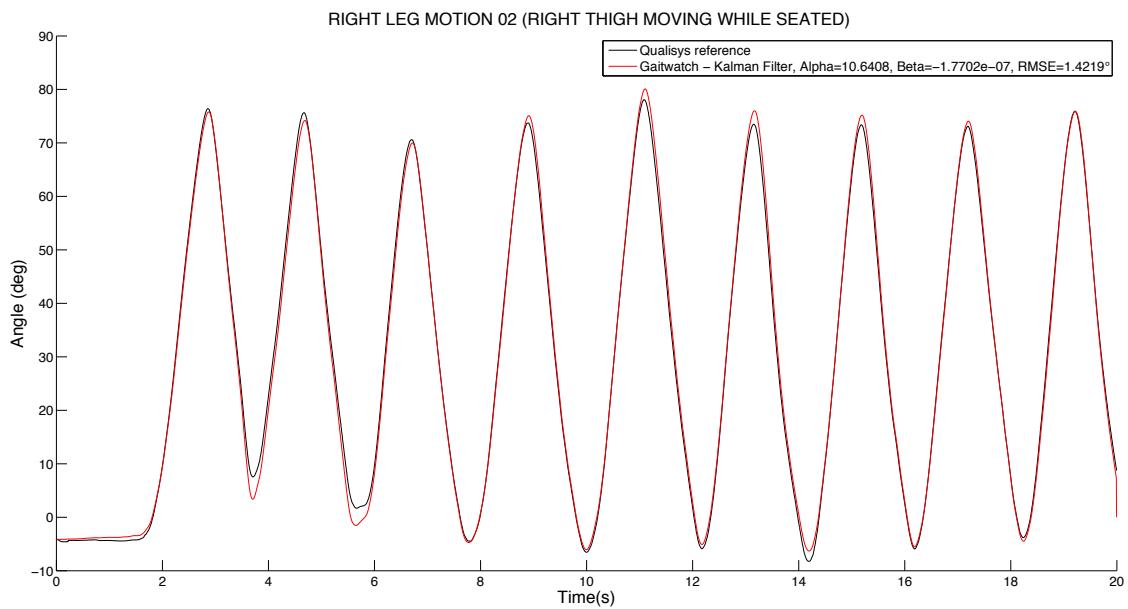


Figure 5.14: Pitch angle while swinging right thigh. (GaitWatch vs. Qualisys).

5.6 Freezing gait

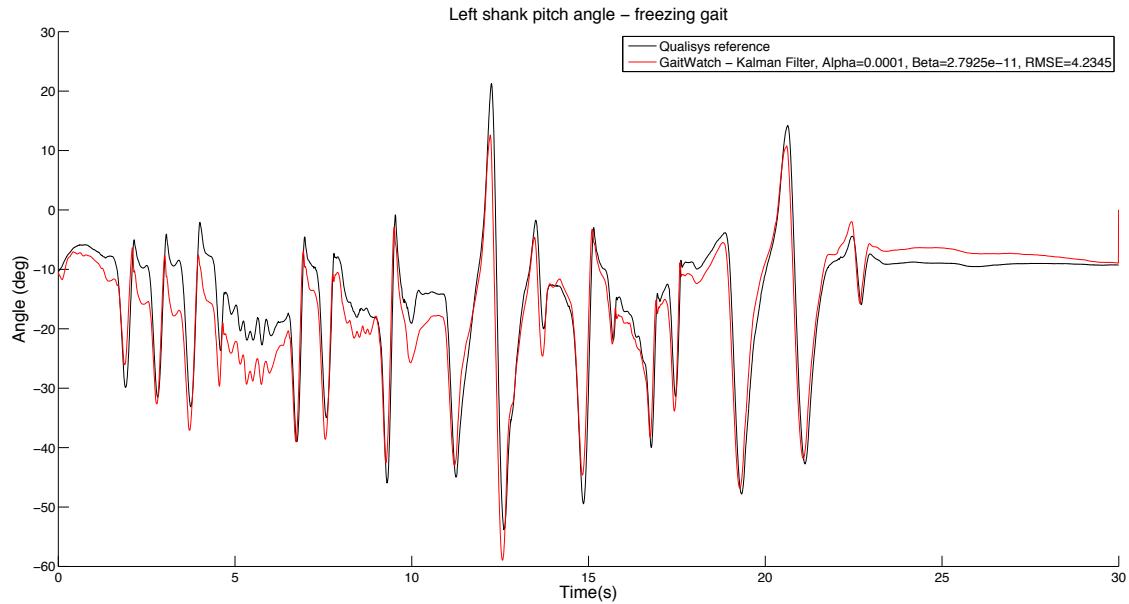


Figure 5.15: Left shank's pitch angle during freezing gait. (GaitWatch vs. Qualisys).

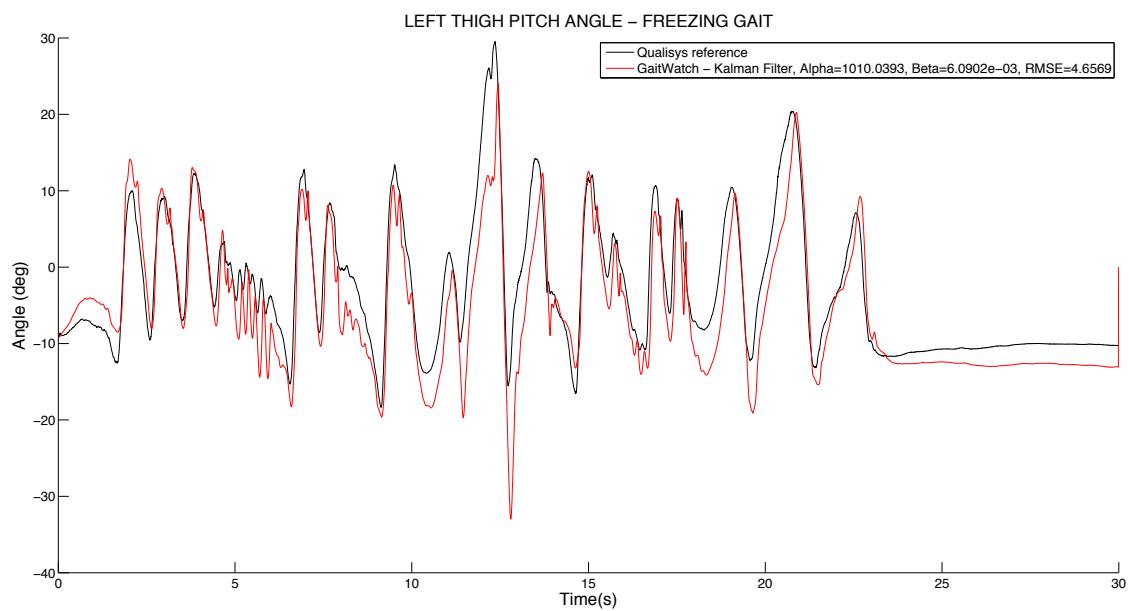


Figure 5.16: Left thigh's pitch angle during freezing gait. (GaitWatch vs. Qualisys).

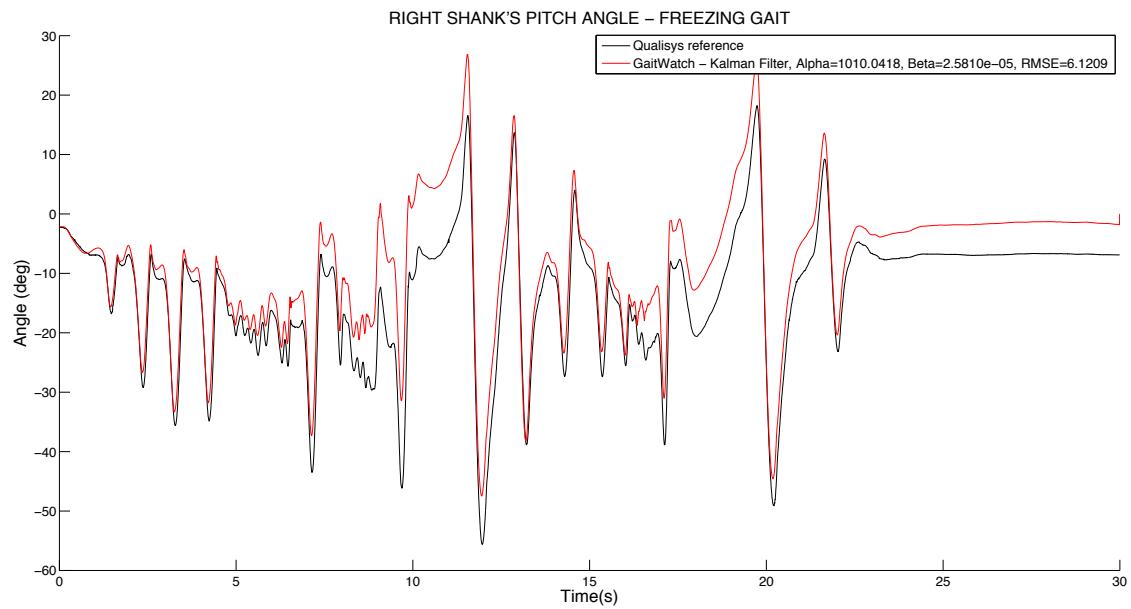


Figure 5.17: Right shank's pitch angle during freezing gait. (GaitWatch vs. Qualisys).

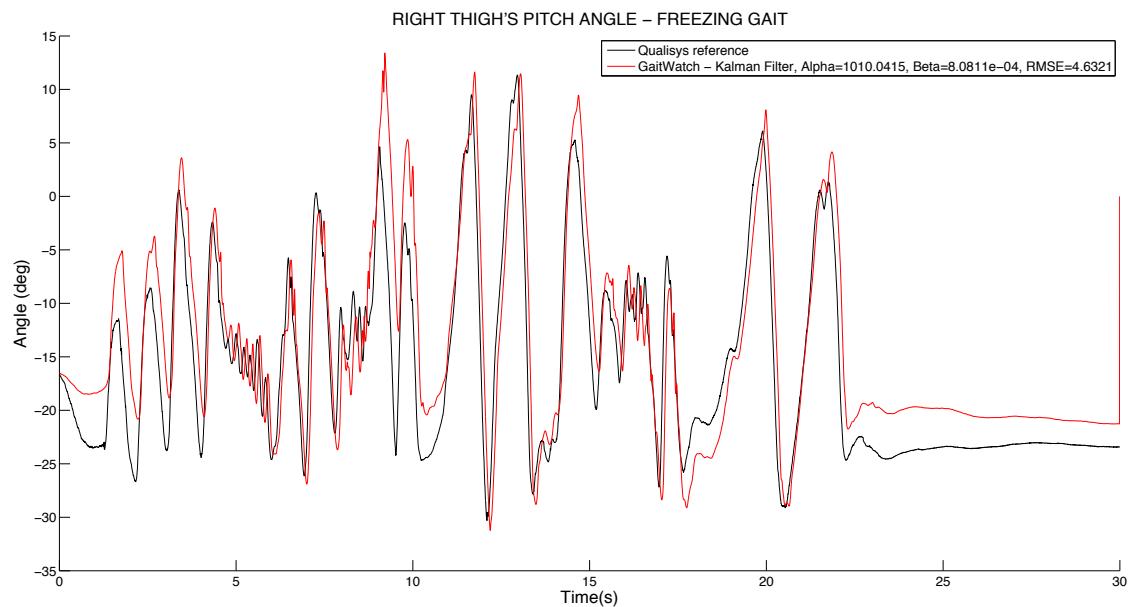


Figure 5.18: Right thigh's pitch angle during freezing gait. (GaitWatch vs. Qualisys).

5.7 Normal gait

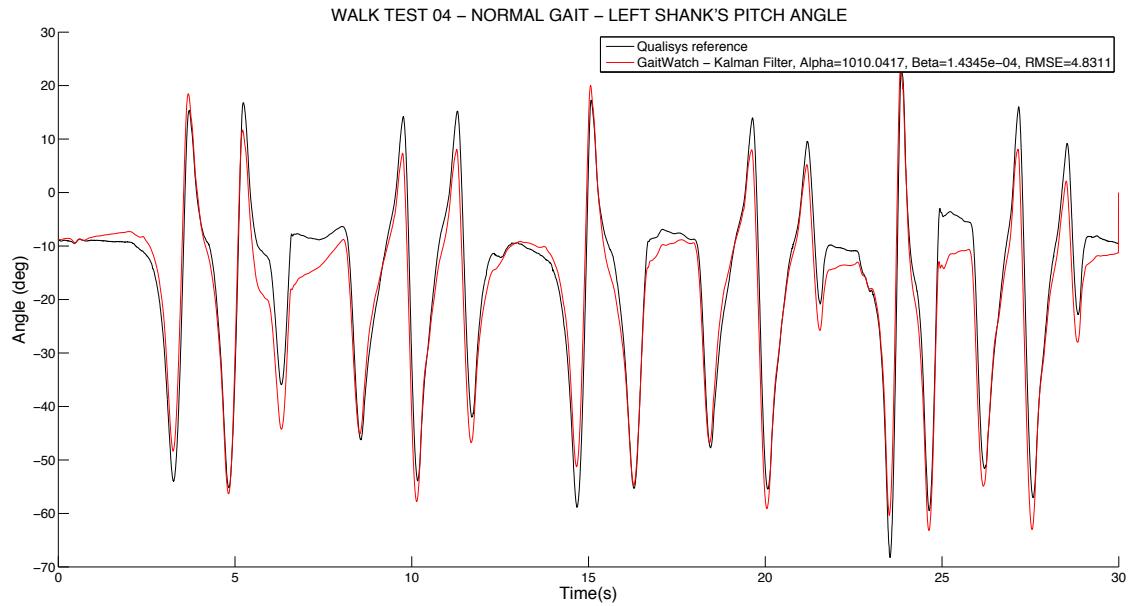


Figure 5.19: Left shank's pitch angle during normal gait. (GaitWatch vs. Qualisys).

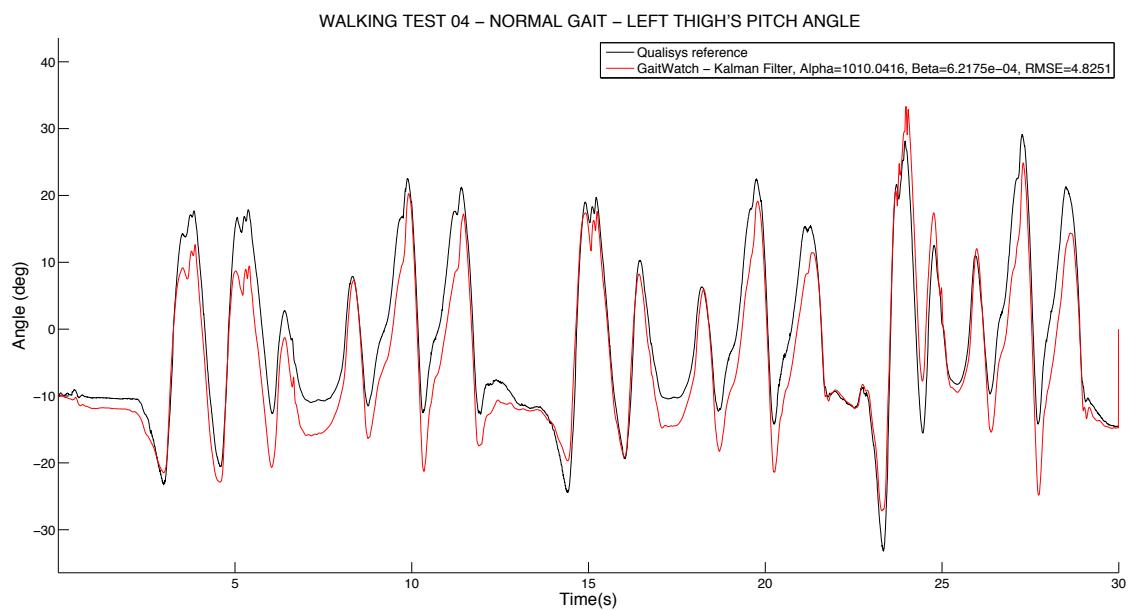


Figure 5.20: Left thigh's pitch angle during normal gait. (GaitWatch vs. Qualisys).

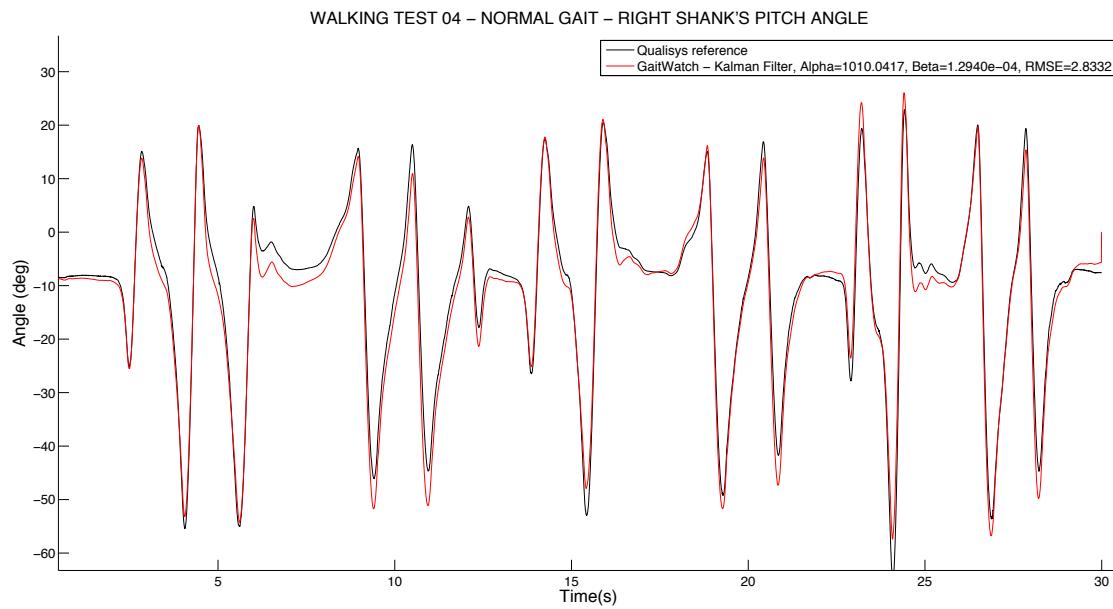


Figure 5.21: Right shank's pitch angle during normal gait. (GaitWatch vs. Qualisys).

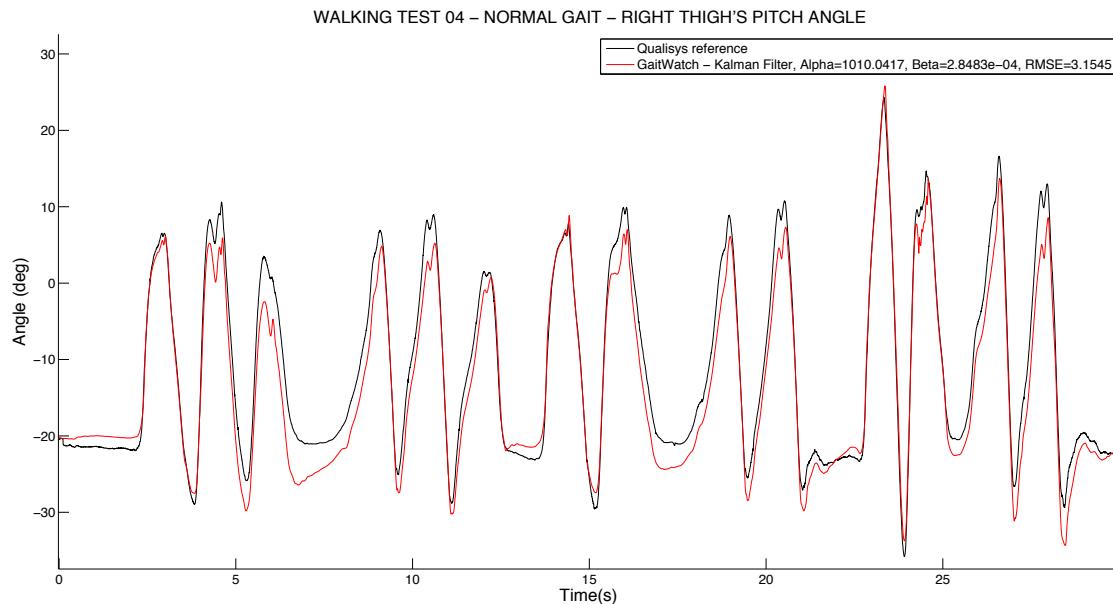


Figure 5.22: Right thigh's pitch angle during normal gait. (GaitWatch vs. Qualisys).

Bibliography

- [1] IMU analog combo board - 5 degrees of freedom idg500/adxl335. <http://goo.gl/0A3Zp9>. [Accessed December 9th, 2013].
- [2] IDG-500 integrated dual-axis gyroscope. <http://goo.gl/w1trOl>. [Accessed December 9th, 2013].
- [3] ADXL335: Small, low power, 3-axis 3g accelerometer. <http://goo.gl/y1QfrL>. [Accessed December 9th, 2013].
- [4] ADXL345: 3-axis, $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ digital accelerometer. <http://goo.gl/38chVM>. [Accessed December 10th, 2013].
- [5] Imu-3000 triple axis motionprocessor gyroscope. <http://goo.gl/rNCpWn>. [Accessed December 10th, 2013].
- [6] Micromag 3-axis magnetometer. <http://goo.gl/f4pCgc>. [Accessed December 10th, 2013].
- [7] AVR ATxmega - development module. <http://goo.gl/HxjWGa>. [Accessed December 10th, 2013].
- [8] F. Camps, S. Harasse, and A. Monin. Numerical calibration for 3-axis accelerometers and magnetometers. In *IEEE International Conference on Electro/Information Technology, 2009. eit '09*, pages 217–221. IEEE, June 2009.
- [9] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [10] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

- [11] National Geophysical Data Center. NGDC geomagnetic calculators - magnetic field. <http://www.ngdc.noaa.gov/geomag-web/#igrfwmm>. [Accessed January 14th, 2013].
- [12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, pages 35–45, 1960.
- [13] G. Welch and G. Bishop. An introduction to the kalman filter. *Notes of ACM SIGGRAPH tutorial on the Kalman filter*, 2001.
- [14] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, December 2008.