

**OFFICIAL GRADUATE PROGRAM IN COMPUTER AND
NETWORKING ENGINEERING**

Department of Computer Architecture and Computer Technology

UNIVERSITY OF GRANADA



Ph.D. Thesis Dissertation:

**Signal Processing of Magnetic and
Inertial Sensor's Signals applied to
Human Body Motion Monitoring**

Written by:

Alberto Olivares Vicente

Supervised by:

Dr. Dr. Juan Manuel Górriz Sáez

Dr. Javier Ramírez Pérez de Inestrosa

Dr. Gonzalo Olivares Ruiz

**PROGRAMA OFICIAL DE POSGRADO EN INGENIERÍA DE
COMPUTADORES Y REDES**

Departamento de Arquitectura y Tecnología de Computadores

Universidad de Granada



TESIS DOCTORAL

**Procesamiento de Señales de Sensores
Magnéticos e Iniciales para la
Monitorización del Movimiento del
Cuerpo Humano**

Realizado por:

Alberto Olivares Vicente

Dirigido por:

Dr. Dr. Juan Manuel Górriz Sáez

Dr. Javier Ramírez Pérez de Inestrosa

Dr. Gonzalo Olivares Ruiz

El doctorando D. Alberto Olivares Vicente y los directores de la tesis D. Juan Manuel Górriz Sáez, D. Javier Ramírez Pérez de Inestrosa, Catedráticos de Universidad del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada y D. Gonzalo Olivares Ruiz, Profesor Titular de Universidad del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada

GARANTIZAMOS AL FIRMAR ESTA TESIS DOCTORAL

que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, a 30 de enero de 2013

Directores de la Tesis

D. Juan Manuel Górriz Sáez D. Javier Ramírez Pérez de Inestrosa D. Gonzalo Olivares Ruiz

Doctorando

D. Alberto Olivares Vicente

Acknowledgements

Foremost I would like to thank Dr. Gonzalo Olivares Ruiz, Dr. Dr. Juan Manuel Górriz Sáez and Dr. Javier Ramírez Pérez de Inestrosa for supervising this work, their valuable comments and suggestions and specially for their continuous support and motivation.

Special thanks go to the members of the Signal Processing and Biomedical Applications (SIPBA) group, who are a role model for me to become a successful researcher, for being such a nice group of people.

I am deeply grateful to everyone that has contributed to the whole project and helped me with some peripheral tasks, their knowledge and ideas. This includes: Dr. Francisco Gómez Mula, Dr. Miguel Damas, Mr. Sascha Poggemann, Mr. Miguel Cabrera, Mr. Rafael López, Mr. Enrique Villarino, Mr. Dennis Huning, Mr. Gonzalo Ruiz and Mr. Oresti Baños.

Many thanks go as well to the supervisors of the Rehabitic Project at Telefónica Digital (formerly Telefónica I+D), Mr. Luis Curiel, Mr. Pedro Antonio de Alarcón and Mr. Ernesto Aranda.

I would also like to express my deep gratitude to the guys at the office for making the writing process more bearable with their jokes and peculiar laughter during their Pomodoro breaks.

Thanks to all my friends and family, and last but not least, to Abi ♡ for supporting me with her affection and love.

This work has been partially funded by the following projects:

- Plataforma Abierta De Procesamiento de Imágenes para Ayuda al Diagnóstico de Alteraciones Neurológicas (PAPI-ADAN). (Open Image Processing Diagnosis Help Platform for Neurological Diseases), funded by the Conserjería de Innovación, Ciencia y Empresa de la Junta de Andalucía.
- AmIVital "Entorno Personal Digital para la Salud y el Bienestar" (Health and Wellness Digital Personal Environment) [1] . CENIT project funded by the CDTI within the INGENIO 2010 program.
- REHABITIC from Telefónica I+D.
- "Optimización Multiobjetivo de Altas Prestaciones y Aplicaciones en Neuroingeniería y Tecnologías para Rehabilitación (hpMooN)", TIN2012-32039.

Preface

The document you have in your hands includes the dissertation entitled "Human body motion monitoring using magnetic and inertial sensors" written to apply for the obtention of the Ph.D. graduation from the University of Granada.

Along this dissertation, we describe a complete system to monitor human body motion using a wireless network of magnetic and inertial sensors.

During the past 4 years I have followed a long way full of obstacles in which I have gained knowledge from many fields. Personally, through this work I have lost the fear to face the unknown.

The fruits of our labor are compiled in the document that you are reading today. We hope you enjoy reading it as much as we did writing it and developing everything behind it. To ease the reading process we have made this document as self-contained as possible so that readers are able to follow it without having to continuously check other resources to understand and reproduce the most important parts of our research.

Abstract

Human body motion monitoring is a field of growing interest mainly due to the fast evolution of miniaturized inertial and magnetic sensors. The analysis of body position and motion is very important in many fields of modern medicine, teleassistance and sports performance monitoring.

This work describes the development of a complete system to monitor the orientation of human body and the analysis of its motion using inertial (accelerometer and gyroscope) and magnetic (magnetometer a.k.a digital compass) sensors, making special emphasis in the design of algorithms to calibrate the sensors and estimate the orientation using the calibrated physical magnitudes. Along this document we describe, in first place, the employed measurement unit (Wagyromag) and the working principles of the embedded sensors as well as the procedures to analyze their quality and performance. In second place, we include an analysis of the existing calibration algorithms and we select the most well-known and efficient ones to carry out a comparative study. This study reveals the great importance of defining a complete mathematical model for the output of the sensors, as well as the necessity of employing homogeneously distributed data to estimate the calibration parameters. In addition, we propose two new algorithms to calibrate the gyroscope without using complex instrumentation. In third place, we analyze the existing possibilities to detect the intensity of the motion which is being monitored and we also carry out a comparative study between four algorithms which have already been applied to this purpose and four new alternatives. This study reveals that two of the proposed algorithms outperform the existing algorithms as they achieve up to 97% of accuracy distinguishing between activity and inactivity. Next, we make a meticulous dissertation about the different procedures and alternatives that can be followed to estimate the attitude (orientation) of a body moving freely in space, using inertial (acceleration and angular rate) and magnetic (Earth's magnetic field) measurements. We analyze the pros and cons of every approach and explain the sensor fusion concept. We also introduce five new fusion alternatives based on variants of LMS and RLS filters, and a fusion model designed from scratch based on the Kalman filter, whose precision is again analyzed by means of a comparative study. This study reveals that the accuracy of the algorithms highly depends on the initial parameters and the intensity of the motion being measured. Furthermore, we propose a new fusion strategy based on the dynamic modification of the parameters of the fusion model based on the Kalman filter so that the algorithm can be used to estimate the orientation of a body moving with changing intensity. This dynamic variant achieves a RMSE of 2.1132° for the estimated orientation angles, which improves the precision of many proprietary algorithms embedded in commercial systems. Finally, we present a summary of the different fields of application in which our system can be used, making special emphasis in those applications related to healthcare.

Resumen

La monitorización del movimiento del cuerpo humano es un campo en auge debido principalmente a la rápida evolución de los sensores iniciales y magnéticos miniaturizados. El análisis del movimiento y la posición corporal es muy importante en muchos campos de la medicina, en aplicaciones de teleasistencia y monitorización de la actividad deportiva.

El presente trabajo presenta el desarrollo completo de un sistema de monitorización del movimiento y la orientación del cuerpo humano basado en sensores iniciales (acelerómetro y giróscopo) y magnéticos (magnetómetro vectorial), haciendo especial énfasis en el desarrollo de algoritmos de calibración de sensores y de estimación de la orientación a partir de las magnitudes físicas ya calibradas. A lo largo del documento presentamos, en primer lugar, el prototipo de medida utilizado (Wagyromag) y los conceptos teóricos de los sensores que lo componen, en conjunto con los procedimientos para analizar su calidad y rendimiento. En segundo lugar, incluimos un análisis de los algoritmos de calibración existentes y seleccionamos los más relevantes para llevar a cabo un estudio comparativo. Dicho estudio revela la gran importancia de la definición de un modelo matemático completo para la salida de los sensores y la necesidad de utilizar datos distribuidos en el espacio de forma homogénea para estimar los parámetros de calibración. Asimismo proponemos dos nuevos algoritmos para la calibración del giróscopo no basados en instrumentación compleja. En tercer lugar analizamos las posibilidades existentes para detectar la intensidad del movimiento que está siendo monitorizado y llevamos a cabo un estudio comparativo entre cuatro algoritmos que ya han sido aplicados para este propósito y cuatro nuevas alternativas. Este estudio revela que dos de los algoritmos propuestos superan el rendimiento de los algoritmos existentes, obteniendo un 97% de precisión en la decisión. Seguidamente hacemos una minuciosa disertación de los distintos procesos y alternativas que se pueden seguir para estimar la orientación de un cuerpo en el espacio a partir de medidas iniciales (aceleración y velocidad angular) y magnéticas (campo magnético terrestre). Analizamos las ventajas e inconvenientes de cada una de ellas y explicamos el concepto de fusión de sensores. Presentamos cinco nuevas alternativas de fusión basadas en variantes de los filtros LMS y RLS, y un modelo de fusión basado en el filtro de Kalman, de las cuales analizamos su precisión mediante un estudio comparativo. El estudio comparativo revela que la precisión de los algoritmos depende en gran medida de los parámetros iniciales seleccionados y de la intensidad del movimiento que está siendo monitorizado. Presentamos también una nueva estrategia de fusión basada en la modificación dinámica de los parámetros del modelo de fusión basado en el filtro de Kalman de forma que el algoritmo pueda ser utilizado para estimar la orientación de un cuerpo moviéndose con intensidad cambiante. Esta variante dinámica obtiene un RMSE de 2.1132° en los ángulos de orientación, el cual es inferior al ofrecido por muchos algoritmos propietarios embebidos en sistemas com-

erciales. Para finalizar, presentamos un resumen de los distintos campos en los que puede ser empleado nuestro sistema, haciendo especial énfasis en aplicaciones relacionadas con la medicina.

Abbreviations

- **ADC:** Analog/Digital Converter.
- **ADL:** Activities of Daily Life.
- **AHRS:** Attitude Heading Reference System.
- **AMD:** Acceleration Magnitude Detector.
- **AMVD:** Acceleration Moving Variance Detector.
- **ARED:** Angular Rate Energy Detector.
- **ARW:** Angle Random Walk.
- **AUC:** Area Under Curve.
- **CUSUM:** Cumulative Sum.
- **CVG:** Coriolis Vibratory Gyro.
- **DCM:** Direction Cosine Matrix.
- **DOF:** Degrees of Freedom.
- **DOM:** Degrees of Movement.
- **DSSS:** Direct Sequence Spread Spectrum.
- **ENU:** East-North-Up.
- **EKF:** Extended Kalman Filter.
- **FIR:** Finite Impulse Response.
- **FQA:** Factored Quaternion Algorithm.
- **FRD:** Filtered Rectifier Detector.
- **FSD:** Framed Spectrum Detector.
- **GKF:** Gated Kalman Filter.
- **HF:** Hydrogen Fluoride.
- **HRLS:** Householder Recursive Least Squares.
- **I2C:** Inter-Integrated Circuit.

- **IMU**: Inertial Measurement Unit.
- **IrDA**: Infrared Data Association.
- **KF**: Kalman Filter.
- **LMS**: Least Mean Squares.
- **LTSD**: Long Term Spectral Detector.
- **MARG**: Magnetic, Angular Rate and Gravity.
- **MBCD**: Memory-Based CUSUM Detector.
- **MBGTD**: Memory-Based Graph Theoretical Detector.
- **MEMS**: MicroElectroMechanical Systems.
- **MIMU**: Magnetic Inertial Measurement Unit.
- **MPU**: Motion Processing Unit.
- **NED**: North-East-Down.
- **NLMS**: Normalized Least Mean Squares.
- **OTG**: On-the-go.
- **PCB**: Printed Circuit Board.
- **QRDRRLS**: QR Decomposition Recursive Least Squares.
- **QUEST**: Quaternion estimator.
- **RLS**: Recursive Least Squares.
- **RMSE**: Root Mean Squared Error.
- **ROC**: Receiver Operating Characteristic.
- **RTCC**: Real Time Clock Calendar.
- **SHOD**: Stance Hypothesis Optimal Detector.
- **SPI**: Serial Peripheral Interface.
- **VAD**: Voice Activity Detection.
- **VRW**: Velocity Random Walk.
- **WDT**: WatchDog Timer.

- **WSN**: Wireless Sensor Network.
- **ZUPT**: Zero Velocity Updates.

Contents

Introducción	27
1 Introduction	43
1.1 General	43
1.2 Motivation	44
1.3 Goals	45
1.4 Project structure	46
1.5 State of the art	47
1.5.1 State of the art at the beginning of the project	47
1.5.2 General view of the state of the art	49
1.5.2.1 Instrumentation	49
1.5.2.2 Calibration	50
1.5.2.3 Determination of motion intensity	51
1.5.2.4 Estimation of the orientation of dynamic bodies	51
1.5.2.5 Estimation of orientation applied to human body position and motion monitoring	52
1.6 Main contributions	54
1.6.1 Published works	54
1.6.1.1 Articles in Refereed Magazines	55

1.6.1.2	Book chapters	55
1.6.1.3	International Conferences	55
1.6.1.4	National Conferences	56
1.7	Document structure	56
2	Instrumentation and tools	59
2.1	MARG Sensors	59
2.1.1	Accelerometers	59
2.1.2	Gyroscopes	65
2.1.3	Magnetometers	68
2.1.3.1	MEMS Based Magnetometers	68
2.2	Hardware	69
2.2.1	Prototypes developed for this project	69
2.2.1.1	WAGYRO	69
2.2.1.2	WAGYROMAG	71
2.2.2	Performance of Sensors	74
2.2.2.1	Analysis of Allan Variance	74
2.2.2.2	Constant Bias	76
2.2.2.3	Thermo-Mechanical White Noise & Angle Random Walk	76
2.2.2.4	Bias Instability	77
2.2.2.5	Temperature Effects	78
2.2.2.6	Calibration Errors	78
2.2.2.7	Analyzing WAGYROMAG's inertial sensors	79
2.3	Firmware	81
2.3.1	Diagram of states	81
2.3.2	Diagram of activity	82
2.3.3	Main routine	83

2.3.4	Auxiliary routines	84
2.3.4.1	Sounding the alarm	84
2.3.4.2	Detecting SD card and creating data file	84
2.3.4.3	Receiving calibration parameters	85
2.3.4.4	Reading data from the Analog/Digital Converter . .	85
2.3.4.5	Reading data from the magnetometer	85
2.3.4.6	Calibrating data	85
2.3.4.7	Writing data in the microSD card	86
2.3.4.8	Sending data wirelessly	86
2.3.5	Interruption routines	86
2.3.5.1	Data gathering interruption	86
2.3.5.2	Mode switching interruption	87
2.4	Data Visualization Software	87
2.4.1	General configuration	88
2.4.2	Connection of sensors and avatar configuration	88
2.4.3	2D Visualization	90
2.4.4	3D Visualization	91
2.4.5	Recording sessions	92
2.4.6	Playing sessions	92
2.5	Wrap-up and future improvement of instrumentation and tools	93
2.5.1	MIMUs	93
2.5.2	Firmware	95
2.5.3	Software	95
3	Calibration	97
3.1	Introduction and chapter's structure	97
3.2	Sensor errors and undesired effects	99

3.2.1	Errors common to MEMS MARG sensors	99
3.2.2	Errors specific to the kind of sensor	100
3.2.2.1	Magnetometer	100
3.2.2.2	Gyroscope	100
3.3	Sensor modeling	101
3.4	Comparative study of calibration algorithms	103
3.4.1	Selected algorithms	103
3.4.1.1	Method 1: Ferrari's in-field procedure	103
3.4.1.2	Method 2: Lotters' quasi-static calibration	107
3.4.1.3	Method 3: Campolo's multi-position procedure	108
3.4.1.4	Method 4: Camps' multi-position procedure	109
3.4.1.5	Method 5: Artese's multi-position procedure	110
3.4.1.6	Method 6: Jurman's multi-position procedure	111
3.4.1.7	Method 7: Won's iterative multi-position procedure .	112
3.4.1.8	Method 8: Frosio's multi-position procedure	113
3.4.1.9	Method 9: Wang's neural network procedure	114
3.4.1.10	Method 10: Skog's multi-position procedure	117
3.4.1.11	Method 11: Gebre-Egziabher's two-step nonlinear procedure	119
3.4.1.12	Method 12. Fong's 18-positions procedure	120
3.4.2	Structure of the study	122
3.4.3	Results	127
3.4.4	Discussion of results	145
3.5	Proposed algorithms	146
3.5.1	Optimizer of spatial distribution of input calibration data . . .	146
3.5.2	A novel accelerometer-aided gyroscope calibration	154
3.5.3	A novel magnetometer-aided gyroscope calibration	159

3.6 Chapter wrap-up	164
4 Determining Motion Intensity	165
4.1 Introduction	165
4.2 Novel approaches	166
4.3 Theoretical basis of motion detection algorithms	166
4.3.1 Magnitude-based methods	167
4.3.1.1 Acceleration Moving Variance Detector (AMVD) . . .	167
4.3.1.2 Acceleration Magnitude Detector (AMD)	167
4.3.1.3 Angular Rate Energy Detector (ARED)	168
4.3.1.4 Stance Hypothesis Optimal Detector (SHOD)	168
4.3.1.5 Filtered Rectifier Detector (FRD)	168
4.3.2 Spectrum-based methods	169
4.3.2.1 A new approach based on the Long Term Spectral Detector (LTSD)	169
4.3.2.2 A new approach based on the Framed Spectrum De- tector (FSD)	169
4.3.3 Memory-based methods	170
4.3.3.1 A new approach based on the Memory-based Theo- retic Graph Detector (MBGTD)	170
4.3.3.2 A new approach based on the Memory-based Cumu- lative Sum Detector (MBCD)	170
4.3.4 Workflow of the algorithms	171
4.4 Experiments	173
4.4.1 Simulations	173
4.4.1.1 Set-up	173
4.4.1.2 Monte Carlo simulation	174
4.4.1.3 Results	175
4.4.2 Real datasets	182

4.4.2.1	Set-up	182
4.4.2.2	Optimization of parameters	182
4.4.2.3	Results	187
4.5	Results discussion	192
4.5.1	Results of our experiments	192
4.5.2	Comparison with results in literature	194
4.6	Chapter wrap-up	195
5	Computation of Attitude Estimation	197
5.1	Basic concepts	198
5.1.1	Coordinate frames	198
5.1.2	Attitude representation	200
5.1.2.1	Euler angles	200
5.1.3	Rotation Matrices	202
5.1.3.1	Direction Cosine Matrix	202
5.1.3.2	Quaternions	204
5.1.4	Projection of gravity vector and Earth's magnetic field	206
5.1.5	Integration of angular rate	207
5.1.6	Sensor fusion	207
5.2	Design and development algorithms	209
5.2.1	Projection of gravity and Earth's magnetic field vectors	209
5.2.1.1	Euler angles-based approach	209
5.2.1.2	Quaternions approach	214
5.2.2	Integration of angular rate	218
5.2.2.1	Cartesian approach	218
5.2.2.2	Quaternions approach	219
5.2.3	Gating	220

5.2.4	Sensor fusion	222
5.2.4.1	Adaptive filtering I: Kalman Filter	223
5.2.4.2	Adaptive filtering II: Least Mean Squares (LMS) and Recursive Least Squares (RLS) filters	232
5.3	Experiments	238
5.3.1	General design	238
5.3.2	Synthesized signals	239
5.3.2.1	Signal synthesizer	240
5.3.2.2	Parameter optimizer	241
5.3.2.3	Structure of the experiments	242
5.3.2.4	Results	243
5.3.3	Real signals	261
5.3.3.1	Reference angle device	261
5.3.3.2	Parameter optimizer	263
5.3.3.3	Structure of the experiments	263
5.3.3.4	Results	263
5.4	Discussion of results	287
5.4.1	Theoretical signals	288
5.4.1.1	High intensity	288
5.4.1.2	Low intensity	293
5.4.1.3	Low and high intensity	293
5.4.2	Real signals	296
5.4.2.1	High intensity	297
5.4.2.2	Low intensity	298
5.4.2.3	Low and high intensity	299
5.5	Chapter wrap-up	301
6	Applications of the System	303

6.1	Telerehabilitation	304
6.2	Gait and Posture analysis	305
6.3	Analysis of activities of daily life	305
6.4	Fall detection	308
6.5	Detection of seizures	309
6.5.1	Epilepsy	309
6.5.2	Parkinson's	311
6.6	Sleep disorders	312
6.7	Sport science	312
6.8	Chapter wrap-up	314
7	Conclusions and Future Work	315
7.1	Conclusions	315
7.2	Future work	320
Conclusiones y Trabajo Futuro		323
Bibliography		331

List of Figures

1	Estructura general del proyecto dividida en capas.	31
1.1	Layer structure of the human body monitoring project. Signal processing layer is highlighted as it includes the core of our work.	47
2.1	Generic structure of a moving mass accelerometer	61
2.2	Single axis accelerometer	61
2.3	Bode's Diagram of the accelerometer frequency response.	63
2.4	Diagram of the generic structure of MEMS accelerometers.	63
2.5	Resistive accelerometer architecture diagram	64
2.6	Capacitive accelerometer architecture diagram	64
2.7	Piezoelectric accelerometer architecture diagram	65
2.8	Basic structure of a vibrating mass gyroscope	66
2.9	Vibrating Plate and Tuning Fork Coriolis Gyroscopes	67
2.10	Vibrating Beam and Vibrating Shell Coriolis Gyroscopes	67
2.11	Concept of the MEMS flux concentrator [2].	69
2.12	WAGYRO's block diagram.	70
2.13	Internal and external appearance of WAGYRO	70
2.14	Wagyromag's block diagram.	71
2.15	External and internal appearance of WAGYROMAG.	74

2.16 Computation of Bias Instability by means of the Allan Deviation (Tri-axial Accelerometer).	79
2.17 Computation of Bias Instability by means of the Allan Deviation (Tri-axial Gyroscope).	80
2.18 Computation of Angle Random Walk by means of the Allan Deviation (Triaxial Accelerometer).	80
2.19 Computation of Angle Random Walk by means of the Allan Deviation (Triaxial Gyroscope).	81
2.20 Diagram of states of WAGYROMAG's firmware. Modified from [3].	82
2.21 Diagram of activity of WAGYROMAG's firmware. Modified from [3].	83
2.22 Attachment of sensor to an avatar's body part.	91
2.23 Example of an acceleration 2D Plot generated by InertialCapture.	92
2.24 InertialCaptures's 3D avatar moving during a data gathering session.	93
2.25 MARG data gathered using Matlab (data extraction and visualization) and SensorList Pro (data reading and storage) running on a Samsung Galaxy SII smartphone.	95
3.1 Representation of non-orthogonal reference frame XYZ' with respect to the orthogonal reference frame XYZ.	102
3.2 Definition of non-orthogonality angles in Wang's algorithm.	115
3.3 Definition of the non-orthogonality angles. The non-orthogonal axes of the accelerometer triad can be aligned with the orthogonal platform axes x ^P ,y ^P ,z ^P through the six angles $\alpha_{xy}, \alpha_{xz}, \alpha_{yx}, \alpha_{yz}, \alpha_{zx}, \alpha_{zy}$	118
3.4 Wagyromag placed with each axis parallel and antiparallel to gravity vector.	123
3.5 Raw acceleration signals at each one of the six positions.	123
3.6 Calibrated acceleration signals at each one of the six positions.	124
3.7 Raw acceleration signals gathered from placing the Wagyromag in 73 different quasi-static positions.	125
3.8 Mode of the raw acceleration gathered at each of the quasi-static positions for each axis. 3D representation.	125

3.9	Mode of the calibrated acceleration gathered at each of the quasi-static positions for each axis. 3D representation	126
3.10	Raw magnetic field values gathered from placing moving the Wagyromag around its axes in a random way. 3D representation and projection over the planes.	126
3.11	Calibrated magnetic field values gathered from placing moving the Wagyromag around its axes in a random way. 3D representation and projection over the planes.	127
3.12	Average calibration errors for 5 six positions runs using a fix reference. Method 1.	128
3.13	Average calibration errors for 5 six positions runs using a fix reference. Method 2.	128
3.14	Average calibration errors for 5 six positions runs using a fix reference. Method 3.	129
3.15	Average calibration errors for 5 six positions runs using a fix reference. Method 4.	129
3.16	Average calibration errors for 5 six positions runs using a fix reference. Method 5.	130
3.17	Average calibration errors for 5 six positions runs using a fix reference. Method 6.	130
3.18	Average calibration errors for 5 six positions runs using a fix reference. Method 7.	131
3.19	Average calibration errors for 5 six positions runs using a fix reference. Method 8.	131
3.20	Average calibration errors for 5 six positions runs using a fix reference. Method 9.	132
3.21	Average calibration errors for 5 six positions runs using a fix reference. Method 10.	132
3.22	Average calibration errors for 5 six positions runs using a fix reference. Method 11.	133
3.23	Average calibration errors for 5 six positions runs using a fix reference. Method 12.	133
3.24	Average calibration errors for 5 six positions runs using a fix reference. All methods.	134

3.25 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 2.	138
3.26 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 3.	138
3.27 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 4.	139
3.28 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 5.	139
3.29 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 6.	140
3.30 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 8.	140
3.31 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 9.	141
3.32 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 10.	141
3.33 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 11.	142
3.34 Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. All methods.	142
3.35 Average magnetic field magnitude errors after calibration for 5 multi-position runs . All methods.	145
3.36 3D Representation of raw magnetic field. Dataset 1.	148
3.37 Representation of θ and ϕ spherical coordinates . Dataset 1.	148
3.38 3D Representation of raw magnetic field. Dataset 2.	149
3.39 Representation of θ and ϕ spherical coordinates . Dataset 2.	149
3.40 3D Representation of raw magnetic field. Dataset 3.	150
3.41 Representation of θ and ϕ spherical coordinates . Dataset 3.	150
3.42 3D Representation of raw magnetic field. Dataset 4.	151
3.43 Representation of θ and ϕ spherical coordinates . Dataset 4.	151
3.44 Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 1. . .	152

3.45 Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 2 . . .	152
3.46 Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 3 . . .	153
3.47 Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 4 . . .	153
3.48 Configuration of the accelerometer-aided gyroscope calibration. The MIMU is attached to the tilted axis of the wheel.	154
3.49 Working principles of a tuning fork Coriolis gyroscope.	156
3.50 Acceleration effects on a tuning fork gyroscope.	156
3.51 Peak detection of the accelerometer signal	157
3.52 Wheel's turn periods obtained from the positive spin	158
3.53 Angular velocity obtained with accelerometer data (positive spin). . .	158
3.54 Calibration line calculated for the X sensitivity axis.	158
3.55 Peak detection of the raw magnetometer signal. Distance between peaks defines the turn period.	160
3.56 Peak detection of the raw magnetometer signal (zoomed). Distance between peaks defines the turn period.	161
3.57 Estimation of turn periods using detected peaks.	161
3.58 Estimation of angular rate using estimated periods.	162
3.59 Estimated calibration lines for the gyroscope triad.	162
3.60 Roll computed using estimation lines. Estimated roll is 176.6° . Actual roll is 180°	163
3.61 Pitch computed using estimation lines. Estimated pitch is 176.7° . Actual pitch is 180°	163
3.62 Yaw computed using estimation lines. Estimated yaw is 185.2° . Actual yaw is 180°	163
4.1 Acceleration, angular rate synthesized signals and activity marker. Activity sequence: walking, lying-standing up, walking, sitting-standing up, running, no angular rate, jumping, walking, lying-standing up, no acceleration.	174
4.2 Theoretical simulation diagram. A Monte Carlo simulation is performed to ensure statistical robustness.	175

4.3	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (AMD)	178
4.4	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (AMVD)	178
4.5	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (ARED)	179
4.6	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (SHOD)	179
4.7	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (FRD-PROD)	180
4.8	Parameter optimization (synthetic signals). Sweep of window length, shift and threshold values to find maximum accuracy (FSD-PROD). The red mark indicates the optimal point of operation.	180
4.9	Parameter optimization (synthetic signals). Sweep of window length, shift and threshold values to find maximum accuracy (LTSD-SUM). The red mark indicates the optimal point of operation.	181
4.10	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (MBCD-PROD) . . .	181
4.11	Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (MBGTD-PROD) . .	182
4.12	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (AMD)	183
4.13	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (AMVD). Real signals. . .	183
4.14	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (ARED). Real signals. . .	184
4.15	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (FRD-PROD). Real signals.	184
4.16	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (SHOD). Real signals. . .	185
4.17	Parameter optimization (real signals). Sweep of window length, shift and threshold values to find maximum accuracy (FSD-PROD). Real signals. The red mark indicates the optimal point of operation.	185

4.18	Parameter optimization (real signals). Sweep of window length, shift and threshold values to find maximum accuracy (LTSD-SUM). Real signals. The red mark indicates the optimal point of operation.	186
4.19	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (MBCD). Real signals. . .	186
4.20	Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (MBGTD). Real signals. .	187
4.21	Input (product of acceleration and angular rate magnitude) and output (vector of characteristics and marker) of the LTSD. Real signals. .	189
4.22	Input and output (vector of characteristics and marker) of the AMVD. Real signals.	190
4.23	ROC curves computed for the eight best methods. Synthesized signals. Complete curves (up), zoomed curves (down).	190
4.24	ROC curves computed for the eight best methods. Real signals. Complete curves (up), zoomed curves (down).	191
5.1	Inertial (XYZ), and coordinate frames: East-North-Up (ENU) and North-East-Down (NED)	199
5.2	Body frame. Turn angle around X axis (Roll), around Y axis (Pitch) and around Z axis (Yaw).	199
5.3	Representation of attitude of body frame with respect to the navigation frame using Euler angles.	201
5.4	Visualization of gimbal lock effect when representing attitude using Euler angles.	201
5.5	Obtention of rotation matrices. North-East-Down system (NED) and East-North-Up system (ENU).	203
5.6	Effects of dynamic bias on the integrated gyroscope output signal. .	208
5.7	Measured acceleration of a moving body with both low and high intensity motion.	221
5.8	Application of MBCUSUM detector to differentiate low intensity motion from high intensity motion (1).	221
5.9	Application of MBCUSUM detector to differentiate low intensity motion from high intensity motion (2).	222

5.10 General diagram of the standard approach of the Kalman filter without sensor fusion.	227
5.11 General diagram of the standard approach of the Kalman filter with sensor fusion.	231
5.12 General diagram of the proposed Gated Kalman Filter.	232
5.13 Diagram of the LMS/RLS filter sensor fusion approach.	238
5.14 Diagram of the Montecarlo Simulation for attitude estimation methods (synthetic signals).	239
5.15 A set of synthesized angle signals.	241
5.16 A set of real gathered angle signals.	242
5.17 Optimization of input parameters. NLMS. High intensity synthetic signals.	244
5.18 Optimization of input parameters. MNLMS. High intensity synthetic signals.	244
5.19 Optimization of input parameters. RLS. High intensity synthetic signals.	245
5.20 Optimization of input parameters. HRLS. High intensity synthetic signals. The cross indicates the optimal point of operation.	245
5.21 Optimization of input parameters. QRDRRLS. High intensity synthetic signals. The cross indicates the optimal point of operation.	246
5.22 Attitude angle computed using sensor fusion algorithms applied on high intensity synthetic signals.	247
5.23 Attitude angle computed using sensor fusion algorithms applied on high intensity synthetic signals (zoomed in).	247
5.24 Optimization of input parameters. NLMS. Low intensity synthetic signals.	248
5.25 Optimization of input parameters. MNLMS. Low intensity synthetic signals.	248
5.26 Optimization of input parameters. RLS. Low intensity synthetic signals.	249
5.27 Optimization of input parameters. HRLS. Low intensity synthetic signals. The cross indicates the optimal point of operation.	249

5.28 Optimization of input parameters. QRDRRLS. Low intensity synthetic signals. The cross indicates the optimal point of operation.	250
5.29 Attitude angle computed using sensor fusion algorithms applied on low intensity synthetic signals.	251
5.30 Attitude angle computed using sensor fusion algorithms applied on low intensity synthetic signals (zoomed in).	251
5.31 Optimization of input parameters. NLMS. Low and high intensity synthetic signals.	252
5.32 Optimization of input parameters. MNLMS. Low and high intensity synthetic signals.	252
5.33 Optimization of input parameters. RLS. Low and high intensity synthetic signals.	253
5.34 Optimization of input parameters. HRLS. Low and high intensity synthetic signals. The cross indicates the optimal point of operation. .	253
5.35 Optimization of input parameters. QRDRRLS. Low and high intensity synthetic signals. The cross indicates the optimal point of operation. .	254
5.36 Orientation angle computed using sensor fusion algorithms applied on synthetic signals presenting both low and high intensity.	255
5.37 Orientation angle computed using sensor fusion algorithms applied on low synthetic signals presenting both low and high intensity (zoomed in).	255
5.38 Validation of optimal synthetic parameters on 100 new synthetic signals. NLMS (high intensity signals and low intensity signals).	256
5.39 Validation of optimal synthetic parameters on 100 new synthetic signals. MNLMS (high intensity signals and low intensity signals).	256
5.40 Validation of optimal synthetic parameters on 100 new synthetic signals. RLS (high intensity signals and low intensity signals).	257
5.41 Validation of optimal synthetic parameters on 100 new synthetic signals. HRLS (high intensity signals and low intensity signals).	257
5.42 Validation of optimal synthetic parameters on 100 new synthetic signals. QRDRRLS (high intensity signals and low intensity signals).	258
5.43 Validation of optimal synthetic parameters on 100 new synthetic signals. NLMS (signals containing both high and low intensity).	258

5.44 Validation of optimal synthetic parameters on 100 new synthetic signals. MNLMS (signals containing both high and low intensity).	259
5.45 Validation of optimal synthetic parameters on 100 new synthetic signals. RLS (signals containing both high and low intensity).	259
5.46 Validation of optimal synthetic parameters on 100 new synthetic signals. HRLS (signals containing both high and low intensity).	260
5.47 Validation of optimal synthetic parameters on 100 new synthetic signals. QRDRRLS (signals containing both high and low intensity).	260
5.48 Angle reference device based on a linear potentiometer.	262
5.49 Spline mapping raw digital potentiometer values to reference angles in degrees.	262
5.50 Estimation of Pitch. Non-fusion algorithms. Complete signal.	265
5.51 Estimation of Pitch. Non-fusion algorithms. Zoomed in on low intensity period.	266
5.52 Estimation of Pitch. Non-fusion algorithms. Zoomed in on high intensity period.	266
5.53 Estimation of Yaw. Non-fusion algorithms. Complete signal.	267
5.54 Estimation of Yaw. Non-fusion algorithms. Zoomed in on low intensity period.	267
5.55 Estimation of Yaw. Non-fusion algorithms. Zoomed in on high intensity period.	268
5.56 NLMS parameter optimization procedure. High intensity real signals.	268
5.57 MNLMS parameter optimization procedure. High intensity real signals.	269
5.58 RLS parameter optimization procedure. High intensity real signals. .	269
5.59 HRLS parameter optimization procedure. High intensity real signals. The cross indicates the optimal point of operation.	270
5.60 QRDRRLS parameter optimization procedure. High intensity real signals. The cross indicates the optimal point of operation.	270
5.61 Pitch angle computed using sensor fusion algorithms applied on high intensity real signals.	272
5.62 Pitch angle computed using sensor fusion algorithms applied on high intensity real signals (zoomed in).	272

5.63 NLMS parameter optimization procedure. Low intensity real signals.	273
5.64 MNLMS parameter optimization procedure. Low intensity real signals.	273
5.65 RLS parameter optimization procedure. Low intensity real signals. . .	274
5.66 HRLS parameter optimization procedure. Low intensity real signals. The cross indicates the optimal point of operation.	274
5.67 QRDRRLS parameter optimization procedure. Low intensity real sig- nals. The cross indicates the optimal point of operation.	275
5.68 Pitch angle computed using sensor fusion algorithms applied on low intensity real signals.	276
5.69 Pitch angle computed using sensor fusion algorithms applied on low intensity real signals (zoomed in).	277
5.70 NLMS parameter optimization procedure. High and low intensity real signals.	277
5.71 MNLMS parameter optimization procedure. High and low intensity real signals.	278
5.72 RLS parameter optimization procedure. High and low intensity real signals.	278
5.73 HRLS parameter optimization procedure. High and low intensity real signals. The cross indicates the optimal point of operation.	279
5.74 QRDRRLS parameter optimization procedure. High and low intensity real signals. The cross indicates the optimal point of operation.	279
5.75 Yaw computed applying sensor fusion methods on real signals (very high and low intensity).	281
5.76 Yaw computed applying NLMS on real signals (very high and low intensity).	282
5.77 Yaw computed applying MNLMS on real signals (very high and low intensity).	282
5.78 Yaw computed applying RLS on real signals (very high and low in- tensity).	283
5.79 Yaw computed applying HRLS on real signals (very high and low in- tensity).	283

5.80 Yaw computed applying QRDRRLS on real signals (very high and low intensity).	284
5.81 Yaw computed applying Madgwick's algorithm on real signals (very high and low intensity).	284
5.82 Yaw computed applying Kalman filter on real signals (very high and low intensity).	285
5.83 Detection of low and high intensity periods. Output of MBCUSUM detector vs. acceleration magnitude.	285
5.84 Estimation of Yaw. Non-fusion algorithms in addition to gating. Complete signal.	286
5.85 Estimation of Yaw. Regular Kalman filter vs. Gated-Kalman filter. First part.	286
5.86 Estimation of Yaw. Regular Kalman filter vs. Gated-Kalman filter. Second part.	287
5.87 Filter taps vs. prediction error vs. angle estimate. NLMS over high intensity synthetic signals using optimal parameters.	288
5.88 Filter taps vs. prediction error vs. angle estimate. RLS over high intensity synthetic signals using optimal parameters.	289
5.89 Filter taps vs. prediction error vs. angle estimate. NLMS over high intensity synthetic signals using non optimal step size ($\mu = 1$).	289
5.90 Comparison of filter taps. RLS over high intensity synthetic signals using two different initial covariance matrices ($\delta = 1e3$ and $\delta = 1e-3$).	291
5.91 Filter taps vs. prediction error vs. angle estimate. HRLS over high intensity synthetic signals using non optimal forgetting factor ($\lambda = 1$).	291
5.92 Filter taps vs. prediction error Vs. angle estimate. QRDRRLS over high intensity synthetic signals using optimal forgetting factor ($\lambda = 3740$).	292
5.93 Filter taps vs. prediction error vs. angle estimate. QRDRRLS over high intensity synthetic signals using non optimal forgetting factor ($\lambda = 1$).	292
5.94 Filter taps vs. prediction error vs. angle estimate. NLMS over low intensity synthetic signals using optimal step size ($\mu = 0.4816$).	294
5.95 Filter taps vs. prediction error vs. angle estimate. QRDRRLS over low intensity synthetic signals using optimal forgetting factor ($\lambda = 1126$).	294

5.96 Filter taps vs. prediction error vs. angle estimate. NLMS over mixed synthetic signals using optimal step size ($\lambda = 0.0120$).	295
5.97 Filter taps vs. prediction error vs. angle estimate. HRLS over mixed synthetic signals using optimal step size ($\lambda = 0.0120$).	296
5.98 Filter taps vs. prediction error vs. angle estimate. HRLS over high intensity real signals using optimal forgetting factor ($\lambda = 0.9$).	298
5.99 Kalman gains vs. prediction error vs. angle estimate. Kalman filter over low intensity real signals using optimal parameters.	299
6.1 Raw acceleration gathered while closing the blinds of a window.	307
6.2 Raw acceleration gathered while cleaning the dust on a table.	307
6.3 Raw acceleration gathered during a front fall.	309
6.4 Raw acceleration gathered while performing a set of push ups. Four MIMUs are placed in both wrists, chest and left ankle.	313
6.5 Raw acceleration gathered while performing a set of squats. Four MIMUs are placed in both wrists, chest and left ankle.	313

List of Tables

2.1	Wagyromag's Bias Instability and Angle Random walk computed using Allan Variance.	79
2.2	Header format of the data file stored in the microSD card.	85
3.1	Average acceleration errors after calibration. 6 positions. Methods 1-4.	135
3.2	Average acceleration errors after calibration. 6 positions. Methods 5-8.	136
3.3	Average acceleration errors after calibration. 6 positions. Methods 9-12.	137
3.4	Average total acceleration errors after calibration. 6 positions. All methods.	137
3.5	Average acceleration errors after calibration. Multi-positions. Methods 2-6.	143
3.6	Average acceleration errors after calibration. Multipositions. Methods 8-11.	144
3.7	Average total acceleration errors after calibration. Multi-positions. All methods.	144
3.8	Average magnetic field magnitude errors after calibration. Multi-positions. All methods.	145
4.1	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Magnitude methods without flexible input).	176

4.2	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Framed Spectrum Detector).	176
4.3	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Long Term Spectral Detector).	177
4.4	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based Graph Theoretic Detector).	177
4.5	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based CUSUM Detector).	177
4.6	Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Filtered Rectifier Detector).	177
4.7	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Magnitude methods without flexible input).	188
4.8	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Framed Spectrum Detector). .	188
4.9	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Long Term Spectral Detector).	188
4.10	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based Graph Theo- retic Detector).	188
4.11	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based CUSUM De- tector).	189
4.12	Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Filtered Rectifier Detector (FRD)).	189
4.13	Area Under Curve (AUC) computed out of ROC curves obtained from application of algorithms on synthesized signals. Number in brackets indicates overall position in performance comparison.	191

4.14	Area Under Curve (AUC) computed out of ROC curves obtained from application of algorithms on real signals. Number in brackets indicates overall position in performance comparison.	192
4.15	Execution times of detection algorithms over a 8000 samples signal. Total execution time, time per sample and percentage of sample period.	194
5.1	Value of the local magnetic vector in our lab in Granada, Spain.	217
5.2	Results of the Monte Carlo simulation (N=50). High intensity synthetic signals. Average RMSE and associated parameters.	246
5.3	Results of the Monte Carlo simulation (N=50). Low intensity synthetic signals. Average RMSE and associated parameters.	250
5.4	Results of the Monte Carlo simulation (N=50). Synthesized signals with both low and high intensity. Average RMSE and associated parameters. (R) indicates that the integration resetting approach was used.	254
5.5	Pitch RMSE: Non-fusion algorithms.	265
5.6	Yaw RMSE: Non-fusion algorithms.	265
5.7	Sensor fusion methods on real signals (high intensity).	271
5.8	Pitch applying sensor fusion methods on real signals (high intensity) using synthetic parameters.	271
5.9	Sensor fusion methods on real signals (low intensity).	275
5.10	Pitch applying sensor fusion methods on real signals (low intensity) using synthetic parameters.	276
5.11	Pitch computed applying sensor fusion methods on real signals (medium and low intensity).	280
5.12	Yaw computed applying sensor fusion methods on real signals (very high and low intensity).	280
5.13	Pitch computed applying sensor fusion methods on real signals using synthetic parameters.	281
5.14	Yaw computed applying sensor fusion methods on real signals using synthetic parameters.	281
5.15	Yaw computed applying gated sensor fusion methods on real signals (very high and low intensity).	287

Introducción

General

Un factor de importancia clave a la hora de analizar la cinemática del cuerpo humano es seleccionar los fenómenos que serán medidos. Con este objeto, analicemos en primer lugar la definición de cinemática incluida en la Enciclopedia Britannica[4].

Cinemática: Rama de la física y subdivisión de la mecánica clásica que estudia el posible movimiento geométrico de un cuerpo o un sistema de cuerpos sin considerar las fuerzas envueltas (causas y efectos de los movimientos). Cuando una partícula se mueve a lo largo de un recorrido curvo, la descripción de su posición es más complicada y requiere dos o tres dimensiones. En dichos casos, las descripciones continuas en forma de un sólo gráfico o una sola fórmula matemática no son factibles. La posición de una partícula moviéndose de forma circular puede ser descrita, por ejemplo, por un radio giratorio de un círculo como si fuese el radio de una rueda que tiene un extremo fijo en el centro de la circunferencia y el otro extremo unido a la partícula. El radio giratorio es conocido como el vector de posición de la partícula, y, si el ángulo entre dicho vector y un radio fijo es conocido en función del tiempo, la magnitud de la velocidad y la aceleración de la partícula pueden ser calculadas.

El movimiento del cuerpo humano describe trayectorias en tres dimensiones, y algunos movimientos, especialmente los de las extremidades, pueden ser modelados como rotaciones alrededor de un eje fijo (la articulación). Como se extrae de la definición anterior, si las magnitudes de la velocidad y la aceleración son conocidas, podemos entonces determinar el ángulo de la posición del vector de la partícula, esto es, podemos determinar los ángulos de posicionamiento u orientación de un

cuero que esté siendo sometido a un movimiento. Por lo tanto, las magnitudes físicas que debe medir un dispositivo cuyo objetivo sea determinar la orientación del cuerpo humano son la aceleración y la velocidad angular.

Otro factor importante es el punto de referencia desde el que realizamos las medidas. El punto de referencia puede ser fijo en el espacio o estar incluido en el cuerpo móvil. Si elegimos la primera opción, una de las soluciones posibles es instalar cámaras en diferentes puntos fijos de una habitación, de forma que actúen como observadores. Esta configuración se usa, por ejemplo, en [5]. Las cámaras Vicon [6] y el sistema Kinect de Microsoft [7] son usados a menudo en sistemas de análisis del movimiento basados en cámaras [8–14]. Por otro lado, si elegimos un punto de medida incluido en el cuerpo sujeto a movimiento, necesitamos un dispositivo que pueda ser colocado en dicho cuerpo y que mida la aceleración y velocidad angular relativas. Por lo tanto, en este caso, nuestro sistema ha de incluir sensores iniciales, en particular sensores de aceleración (acelerómetros) y sensores de velocidad angular (giróscopos).

Uno de los requerimientos iniciales del sistema que pretendíamos utilizar para realizar nuestro trabajo es que estuviese basado en dispositivos que pudiesen ser colocados en diferentes puntos del cuerpo y calcular la orientación de los mismos. Por lo tanto, puesto que esta configuración utiliza puntos de medida en el cuerpo que se mueve, necesitamos sensores iniciales. En particular, utilizaremos dispositivos denominados Unidades de Medida Inercial o IMUs (por sus siglas en inglés), los cuales incluyen sensores iniciales en conjunto con otros elementos que proporcionan otras funcionalidades como el procesamiento y almacenamiento de los datos medidos y su envío de forma inalámbrica, entre otros.

Si usamos acelerómetros y giróscopos triaxiales, nuestra IMU tendrá seis grados de libertad, lo cual es suficiente para medir el movimiento relativo en el plano XZ e YZ. Sin embargo, no podemos determinar la orientación inicial respecto al norte, ya que carecemos de una referencia de posicionamiento absoluta en el plano XZ. Para solucionar este problema necesitamos un sensor que sea capaz de medir la orientación respecto al norte magnético de la Tierra, es decir, que mida el campo magnético terrestre. Por lo tanto, si añadimos un magnetómetro a nuestra unidad de medida inercial añadiremos 3 grados más de libertad al sistema. Las unidades de medida inercial que incluyen magnetómetros son normalmente conocidas como Unidades de Medida Inercial Magnética, MIMUs (por sus siglas en inglés).

Motivación

La idea de desarrollar un sistema inalámbrico de monitorización del movimiento corporal surgió hace cuatro años cuando Telefónica I+D Granada contactó con noso-

tros para el desarrollo de un prototipo de bajo coste y pequeño tamaño que pudiese colocarse en las piernas de un paciente para monitorizar el proceso de rehabilitación tras la cirugía de rodilla.

Los requerimientos iniciales del prototipo eran muy generales, ya que las únicas condiciones eran que fuese inalámbrico, lo menos intrusivo posible y que transmitiese los datos medidos en tiempo real.

Una vez desarrollado el dispositivo, del cual también hablaremos a lo largo de la memoria, fui el encargado de desarrollar los algoritmos de calibración y cálculo de la orientación del dispositivo.

Tras la finalización de la colaboración en conjunto con Telefónica I+D, continuamos, en el marco del proyecto AmIVital [1], con el estudio de nuevos algoritmos orientados a desarrollar aplicaciones de telerehabilitación más complejas, monitorización de deportes y aplicaciones de teleasistencia.

Un aspecto importante para la motivación del trabajo en el ámbito de la monitorización de la posición y el movimiento es el gran número de campos de aplicación que abarca, entre los cuales destacan las aplicaciones de salud.

Como iremos explicando más adelante, el estudio y monitorización del movimiento y la postura del cuerpo humano es una rama muy importante dentro de la medicina ya que un gran número de dolencias causan trastornos en el movimiento y/o pueden ser diagnosticadas a través del análisis del mismo. A modo de ejemplo, nuestro sistema puede ser utilizado para detectar convulsiones tonicoclónicas epilépticas, episodios de bradikinesia, temblor y diskinesia en enfermos de Parkinson, prevención y detección de caídas de personas mayores y estudiar trastornos del sueño, entre muchos otros.

El hecho de que nuestro sistema pueda ser aplicado para mejorar la calidad de vida de muchos enfermos fue la motivación detonante para llevar a cabo este trabajo.

Objetivos

Los principales objetivos al comienzo de nuestro trabajo se basaban en, utilizando el prototipo de monitorización ya desarrollado, estudiar la calibración del mismo y procesar los datos medidos para calcular la posición del sujeto monitorizado. Si los clasificamos de acuerdo a los dos campos de trabajo, tenemos:

- **Calibración:**
 - Realizar un completo estudio del estado del arte de algoritmos de cali-

bración de magnetómetros, acelerómetros y giróscopos de tipo MEMS.

- Seleccionar e implementar los algoritmos más utilizados en los últimos años con el objetivo de analizar su precisión y rendimiento mediante un estudio comparativo.
- Proponer nuevos algoritmos de calibración que no requieran instrumentación compleja.

- **Estimación de la orientación y posición:**

- Realizar un completo estudio del estado del arte de los diversos algoritmos existentes en la literatura.
- Proponer nuevas variantes de estimación de la orientación basadas en filtrado adaptativo y comparar su rendimiento con algunos de los algoritmos más utilizados en los últimos años.
- Proponer algoritmos que modifiquen sus parámetros de forma dinámica en base a la intensidad del movimiento que está siendo monitorizado.

Por otro lado, una vez completados los dos puntos anteriores, se pretende buscar aplicaciones del sistema dentro del ámbito sanitario.

Estructura del proyecto

Si consideramos tanto el desarrollo del sistema de monitorización del movimiento del cuerpo humano, como la aplicación del mismo en diversas aplicaciones como un proyecto, podríamos dividirlo en la estructura de capas reflejada en la figura 1. Dichas capas serían:

- **Instrumentación (Hardware):** Esta capa engloba el diseño y desarrollo de unidades de medida inercial así como el uso de otros dispositivos comerciales que incluyan sensores magnéticos y/o inerciales. A lo largo de este proyecto se ha trabajado con Wagyro y Wagyromag (ambos desarrollados en el departamento de ATC), Equivital [15] y diversos Smartphones como iPhone 4, Samsung Galaxy Nexus y Samsung Galaxy SII.

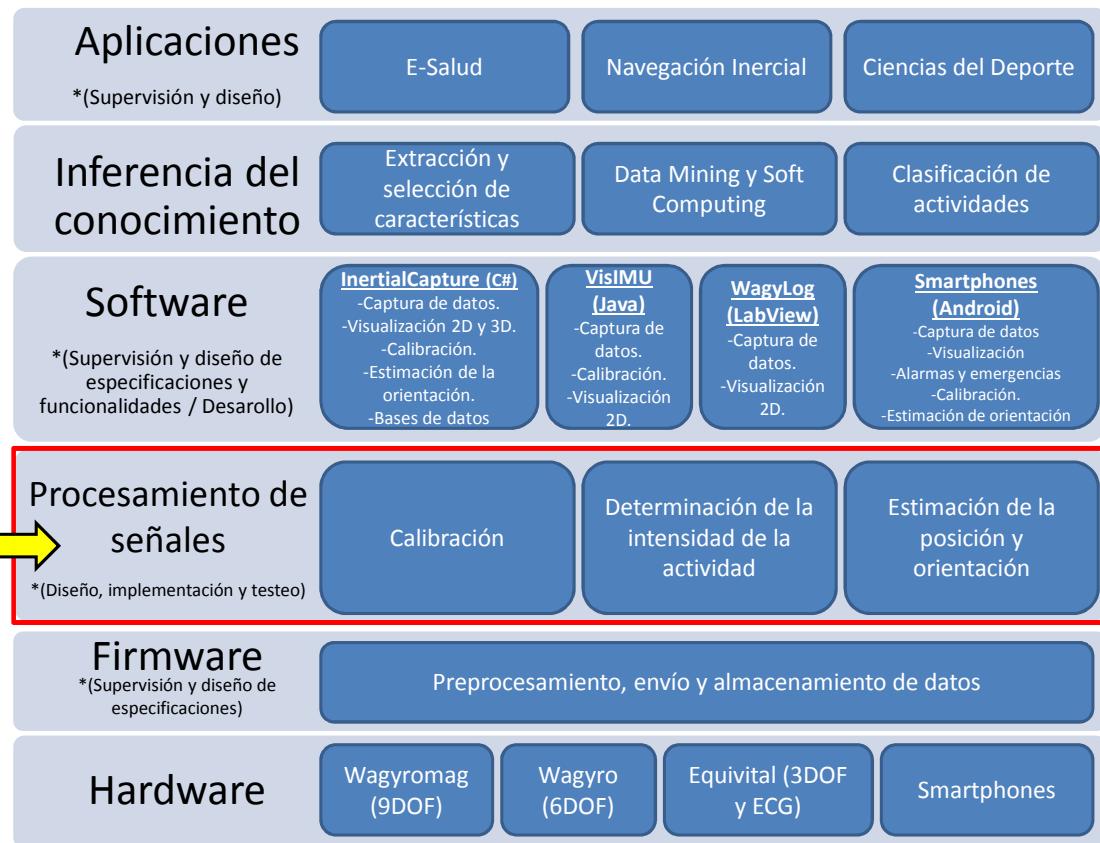


Figure 1: Estructura general del proyecto dividida en capas.

- **Firmware:** Esta capa engloba el diseño e implementación del código que da soporte a las funcionalidades de Wagyromag. A lo largo del proyecto se ha trabajado con un PIC de la familia 24F [16] de Microchip.
- **Procesamiento de señales:** Esta capa engloba el desarrollo de algoritmos óptimos de calibración así como el estudio en profundidad de algoritmos ya existentes. Asimismo también incluye el análisis y desarrollo de algoritmos de detección de la intensidad del movimiento y la estimación de la orientación de un cuerpo en el espacio. A lo largo de este proyecto se ha trabajado de manera profunda y exhaustiva en estos tres ámbitos.
- **Software:** Esta capa engloba el diseño y desarrollo de programas que permitan la integración de los algoritmos de procesado de la señal para captar, procesar, representar gráficamente y almacenar los datos provenientes de las unidades de medida inercial. A lo largo de este proyecto se han desarrollado varias aplicaciones en diversas plataformas: C#, Java, Labview y Android.

- **Inferencia del conocimiento:** Esta capa engloba el uso de los datos ya calibrados y procesados para desarrollar sistemas inteligentes capaces de discernir entre diferentes tipos de actividades (andar, correr, sentarse, etc.) y detectar eventos (caídas, ataques epilépticos, etc.).
- **Aplicaciones:** Esta capa engloba el diseño de aplicaciones del sistema completo. A lo largo del proyecto el sistema ha sido usado en la telerehabilitación de pacientes operados de rodilla y va a ser aplicado en un proyecto piloto de monitorización y detección de ataques epilépticos de tipo motriz.

Estado del arte

Estado del arte al comienzo

Si comenzamos analizando el estado del arte de los dispositivos existentes hace cuatro años que podían ser empleados para monitorizar el movimiento del cuerpo humano, cabe destacar que casi ninguno de ellos era totalmente inalámbrico, y los pocos que lo eran tenían un precio desorbitado. Además, todos estaban basados en protocolos de transmisión tipo Bluetooth, el cual tiene un alto consumo de energía, o directamente no tenían capacidad para enviar datos en tiempo real.

Entre los pocos dispositivos existentes por aquel entonces, se encontraban Xbus Master (ahora Xbus Kit [17]), Dynaport (cuya versión actual puede verse en [18]), KinetiSense (cuya versión actual puede verse en [19]), Physilog [20], Nike+ [21] y Fitbit (cuya versión actual puede consultarse en [22]). La mayoría de ellos consiste en una serie de sensores unidos por cable a una unidad central de gran tamaño que, o bien almacena los datos en una memoria local, o bien los transmite de forma inalámbrica.

Ya que ningún dispositivo cumplía los requerimientos iniciales, se procedió al desarrollo de un prototipo propio.

Respecto a los algoritmos de calibración, ya existía una gran cantidad de métodos para los tres sensores utilizados (acelerómetros, magnetómetros y giróscopos). Sin embargo, al analizarlos de forma detallada, observamos que la gran mayoría de los métodos eran meras repeticiones o modificaciones muy ligeras del mismo. Por lo tanto, vimos la necesidad de hacer una criba importante para seleccionar aquellos algoritmos más citados de forma que pudiésemos llevar a cabo un estudio comparativo entre ellos para determinar cuál ofrecía una mejor precisión.

Observamos que existían dos grandes grupos de métodos, aquellos basados en la minimización de una función de error o coste para estimar los parámetros de calibración óptimos, y aquellos que calculaban los parámetros de forma directa aplicando una serie de transformaciones algebraicas.

En este punto surgió un problema. La mayoría de los algoritmos de calibración del giróscopo estaban basados en el uso de mesas rotatorias multieje controladas por ordenador. Dichas mesas tienen un precio de mercado superior a los 10.000 euros, con lo cual tuvimos que buscar nuevas soluciones no basadas en este tipo de dispositivos.

Analizando el estado del arte de los algoritmos de estimación de la orientación observamos que, prácticamente el 100% de los algoritmos existentes estaban basados en la fusión de la información de los distintos sensores usando el filtro de Kalman. Además, la mayoría de ellos explicaban de manera muy superficial el diseño del filtro, siendo ésta la parte más importante del algoritmo, y a menudo la más compleja.

Observamos también que no había ningún estudio que indicase la precisión de la fusión de sensores para distintos tipos de intensidad del movimiento.

Las primeras pruebas realizadas nos hicieron ver que la precisión de, por ejemplo, el filtro de Kalman, aplicado a señales iniciales de intensidad alta, difería mucho de la obtenida para señales medidas durante la realización de movimientos lentos. Vimos, por tanto, la importancia de analizar la precisión de los filtros para señales recogidas durante situaciones de movimiento de distinta intensidad.

Algunos de los algoritmos más recientes comenzaban a sugerir la posibilidad de aplicar filtros complementarios cuya ganancia fuese adaptada de forma dinámica dependiendo del grado de intensidad del movimiento medida.

Era evidente, por tanto, la necesidad de aplicar un algoritmo de detección de la intensidad del movimiento para poder cambiar de forma dinámica los parámetros del filtro.

Para este efecto, la mayoría de los trabajos existentes utilizaban algoritmos muy básicos basados simplemente en el análisis de la magnitud de la aceleración y velocidad medidas, por lo que decidimos buscar otros algoritmos más completos que pudiesen mejorar la precisión con la que se determinan los instantes de inicio y fin de períodos de distinta intensidad de movimiento.

Estado del arte general

Instrumentación

Como se ha comentado anteriormente, es muy habitual integrar acelerómetros y giróscopos en conjunto con una unidad de procesamiento y módulos de memoria para formar Unidades de Medida Inercial (IMU). Si además, el dispositivo integra también un magnetómetro, entonces es conocido como una Unidad de Medida Inercial Magnética (MIMU). Existe un amplio número de IMUs y MIMUs comerciales con diferentes especificaciones, sin embargo, la mayoría de ellas se producen a baja

escala, con lo cual los precios son relativamente altos.

Antes de decidirnos por desarrollar nuestro propio prototipo de MIMU llevamos a cabo un minucioso sondeo del mercado para analizar las distintas opciones existentes. De entre todos los dispositivos analizados, observamos que, como norma general, las IMUs y las MIMUs pueden ser elegidas en función de los siguientes criterios:

- **Tamaño:** Hay una gran diferencia de tamaño entre, por ejemplo, los dispositivos inerciales utilizados en la aviación comercial y la monitorización del movimiento del cuerpo humano. Como normal general, una reducción del tamaño de la unidad, y por tanto de los sensores, va unida de la mano a una reducción en la precisión de los mismos.
- **Peso:** Los dispositivos de gran tamaño son también bastante pesados ya que están protegidos por carcasa de materiales muy robustos. El peso del dispositivo ha de ser acorde con el tipo de aplicación en la que será utilizado.
- **Transmisión de datos:** Podemos elegir entre transmisión por cable o transmisión inalámbrica. Aparte del tipo de transmisión, también deberemos seleccionar el protocolo utilizado (Bluetooth 2.0, Bluetooth 3.0, Zigbee, ANT+, etc.).
- **Procesador:** Algunos dispositivos incluyen un microcontrolador o microprocesador para dotar al dispositivo de capacidad de procesamiento local.
- **Almacenamiento de datos:** Algunos dispositivos incluyen módulos de memoria integrados y/o ranuras para tarjetas de memoria externas. El almacenamiento de datos es imprescindible en aquellas aplicaciones en las que los datos no pueden ser transmitidos en tiempo real.
- **Especificaciones de los sensores:** El factor más determinante a la hora de elegir una IMU o MIMU es el tipo de sensores que incluyen. Es imprescindible estudiar detalladamente las especificaciones de los sensores, tales como la sensibilidad, precisión, resolución, rango de medida, etc.

Ejemplos de IMUs y MIMUs comerciales que han aparecido recientemente en el mercado, y que poseen diferentes características de peso, precio, tamaño y calidad, pueden ser consultadas en [23–30]. Además, una lista muy completa de IMUS y MIMUs comerciales puede ser también consultada en [31].

Nuevos dispositivos están apareciendo constantemente en el mercado, principalmente debido a que la industria de sensores de tipo Microelectromecánico (MEMS) evoluciona con gran rapidez. Además, hoy en día, la mayoría de los smartphones

de alta gama incluyen acelerómetros, giróscopos y magnetómetros triaxiales, lo que hace que puedan ser utilizados como MIMUs. Esto ofrece la oportunidad de desarrollar aplicaciones que incluyan los algoritmos que serán descritos a lo largo de este trabajo y, consecuentemente, aumentar el nicho de mercado de las aplicaciones de monitorización del movimiento del cuerpo humano, ya que en pocos años prácticamente todas las personas llevarán consigo un smartphone y, por lo tanto, una MIMU. Sin embargo, los smartphones no son una alternativa definitiva a las MIMU, ya que muchas aplicaciones de monitorización del movimiento del cuerpo humano requieren que el sujeto porte varias MIMUs adosadas a distintas partes del cuerpo formando una red inalámbrica de área personal .

Calibración

Durante la última década se han publicado cientos de trabajos relacionados con la calibración de sensores MEMS de tipo MARG. Sin embargo, la mayoría de ellos son meras repeticiones o variaciones muy leves de los métodos publicados anteriormente. A lo largo de esta sección nombraremos algunos de los métodos más representativos.

Comenzamos la revisión del estado del arte con el trabajo más citado a lo largo de estos años; Ferraris et al. presentan en [32] un método para calibrar acelerómetros y giróscopos triaxiales que no necesita instrumentación compleja ni maniobras complicadas. Este trabajo ha servido como base para muchos otros métodos que han sido desarrollados en los años subsecuentes. En este método, la calibración del acelerómetro se lleva a cabo a través de un proceso en el que se coloca el dispositivo en seis posiciones diferentes y se aplican transformaciones algebraicas simples sobre los datos de aceleración recogidos en dichas posiciones. Por otro lado, el giróscopo se calibra basándose en un proceso que sustituye la necesidad de utilizar una referencia de velocidad angular por una rotación conocida. Explicaremos este método en profundidad en el capítulo 3, en conjunto con el método propuesto por Won et al. [33] el cual también está basado en el procedimiento de seis posiciones.

El método de seis posiciones permite únicamente calcular seis parámetros de calibración (factores de escala y desviaciones). Para permitir el uso de modelos de calibración basados en un mayor número de parámetros, varios autores utilizan un método que está basado en colocar el dispositivo en múltiples posiciones para utilizar los datos recogidos en el proceso de minimización de una función de coste. Ejemplos de este procedimiento pueden ser encontrados en [34–52]. Estos trabajos se diferencian en la parametrización utilizada para modelar el sensor, el método utilizado para recoger los datos (posiciones aleatorias o posiciones predefinidas), la instrumentación (mesas rotatorias multieje, dispositivos mecánicos simples o ningún tipo de herramienta) y la función de error o función de coste y el método utilizado para minimizarla (Gauss-Newton, método de mínimos cuadrados lineal o no lineal, descenso de gradiente, Levenberg-Marquadt, etc.).

Existe también un grupo de trabajos que están basados en técnicas diferentes tales como [53] y [54] los cuales presentan algoritmos empíricos muy simples basados en operaciones algebraicas básicas. Por otro lado, en [55] se presenta un método de calibración que utiliza un algoritmo de mínimos cuadrados basado en lógica Fuzzy de tipo Mamdami. Otro método bastante conocido es el denominado procedimiento de dos pasos incluido en [56]. Además, otros autores utilizan algoritmos basados en filtros FIR como los presentados en [57] y [58]. Finalmente, otros trabajos se centran en resolver una parte específica del procedimiento de calibración. Por ejemplo, [59] usa un método de optimización del gradiente conjugado para hallar simplemente el error de alineación horizontal y [60] intenta minimizar los errores de calibración en sistemas que utilizan múltiples sensores.

Detección de la intensidad del movimiento

La detección de la intensidad del movimiento que está siendo monitorizado es un tema que no ha sido tratado por muchos autores. Sin embargo, es un factor muy importante a la hora de mejorar la precisión de los sistemas de estimación de la orientación. Por ejemplo, para recalcular algunos parámetros de calibración [61], resetear desviaciones en las trayectorias estimadas [62, 63] o adaptar de forma dinámica los parámetros de diseño de los filtros de fusión de sensores durante el uso del dispositivo es necesario detectar los instantes en los que el cuerpo se mueve con distinta intensidad.

Los algoritmos de detección pueden ser clasificados de acuerdo al sensor que utilizan como entrada. El detector basado en la varianza móvil de la aceleración (AMVD por sus siglas en inglés), y el detector basado en la magnitud de la aceleración (AMD por sus siglas en inglés), ambos implementados en [64], utilizan únicamente la aceleración medida por el acelerómetro para llevar a cabo la clasificación. Esto puede limitar la capacidad de detectar instantes de actividad en los que no se está midiendo aceleración, pero el cuerpo sí está sujeto a velocidad angular. Por otro lado, en [65] se presenta un detector de actividad basado exclusivamente en el análisis de la magnitud de la velocidad angular, lo que también puede llevar a situaciones en las que se clasifique erróneamente como estático un cuerpo que esté sujeto a aceleraciones pero no tenga velocidad angular. Para solucionar este problema, surge el detector óptimo de hipotética postura (SHOD por sus siglas en inglés) presentado en [66]. Este algoritmo utiliza tanto la aceleración como la velocidad angular para incrementar la precisión de la detección. Finalmente, el detector basado en rectificación y filtrado presentado en [67] tiene entrada flexible y, por lo tanto se pueden utilizar como entrada las magnitudes de la aceleración y la velocidad angular por separado o en conjunto formando combinaciones lineales.

Un estudio comparativo preliminar entre algunos de los algoritmos que acaba-

mos de mencionar puede ser consultado en [66]. En este trabajo se incluye una comparación del rendimiento de los detectores utilizando señales reales obtenidas de diferentes sensores. La definición de los métodos es muy rigurosa pero, como reconocen los autores, la cantidad de señales utilizada es muy baja, por lo que el estudio no está completo desde el punto de vista de robustez estadística.

Estimación de la orientación de cuerpos móviles

Los métodos de estimación de la orientación comenzaron a ser desarrollados en los años setenta para las misiones espaciales. Se desarrollaron una serie de algoritmos con el objetivo de determinar la orientación de la nave u objeto espacial respecto a distintos sistemas de referencia. Ejemplos de estos primeros métodos pueden ser encontrados en [68] y en [69], donde se describen los métodos TRIAD y QUEST respectivamente. El algoritmo TRIAD proporciona una solución determinística para la orientación basándose en dos observaciones vectoriales y dos referencias vectoriales que son utilizadas para calcular la matriz de orientación. A su vez, el algoritmo QUEST minimiza una función de coste para encontrar los cuaterniones óptimos que representan la orientación. En nuestro contexto (determinación de la orientación utilizando sensores magnéticos e iniciales), estos métodos sólo utilizan el acelerómetro y el magnetómetro. Para evitar los problemas en la estimación de la orientación durante instantes de alta intensidad de movimiento, en [70] se fusiona el algoritmo QUEST con la estimación de la orientación calculada mediante integración de la velocidad angular. Para realizar dicha fusión se utiliza un filtro de Kalman extendido (EKF por sus siglas en inglés).

Una alternativa a QUEST se encuentra en [71]. El algoritmo presentado en este trabajo, denominado Algoritmo de Cuaterniones Factorizados (FQA por sus siglas en inglés), a diferencia de QUEST, evita el uso de datos del magnetómetro para calcular el *pitch* y el *roll* (dos de los tres ángulos de posicionamiento), ya que las perturbaciones magnéticas pueden llevar a errores durante la estimación de los mismos. En este caso, el uso del magnetómetro se ha restringido al cálculo del tercer ángulo de orientación, el denominado *yaw*.

Distintas variaciones de estos métodos han sido propuestas a lo largo de las últimas décadas, principalmente por Malcolm D. Shuster [72, 73], F. Landis Markley [74–78], y Xiaoping Yun [79]. En [79] se utiliza un algoritmo de iteración de Gauss-Newton para encontrar el cuaternion que relaciona de forma óptima las medidas de aceleración y campo magnético en el sistema de referencia del cuerpo con las mismas medidas desde el punto de vista del sistema de referencia terrestre. Dicho cuaternion se fusiona posteriormente con la velocidad angular utilizando un filtro de Kalman extendido. Este enfoque proporciona buenos resultados pero añade un alto coste computacional debido al proceso de minimización que tiene que ser ejecutado para cada conjunto de medidas.

Madgwick [80] presenta, de la misma manera, un enfoque basado en cuater-

niones que estima el cuaternion de observación (campos magnético y gravitatorio terrestres) usando un método de descenso de gradiente, para posteriormente fusionarlo con el cuaternion de velocidad angular utilizando un filtro complementario.

Otros ejemplos de algoritmos utilizados para calcular los ángulos de Euler que determinan la orientación de un cuerpo pueden ser encontrados en [81–83].

Asimismo, un completo estudio del estado del arte de los métodos no lineales de estimación de la orientación puede ser consultado en [84]. Por otro lado, Young et al. presentan en [85] un estudio comparativo entre algunos de los métodos más conocidos.

Estimación de la orientación aplicada a la monitorización del movimiento del cuerpo humano

Como se ha comentado anteriormente, existe una gran variedad de campos de aplicación de los sistemas de orientación de la posición del cuerpo humano. Zhang et al. proponen en [86] una red bayesiana dinámica e híbrida para modelar el ángulo de orientación no lineal de la cadera en condiciones dinámicas, y un filtro Gaussiano de partículas para estimar dicho ángulo durante los ciclos de la marcha a partir de las medidas de un acelerómetro vestible adherido a los muslos. Este algoritmo consigue un error mínimo en la estimación del ángulo de orientación de 1.50° RMS. Sin embargo, ya que la estimación de la orientación está basada exclusivamente en las medidas de un acelerómetro, este algoritmo presentará, con toda seguridad, errores mucho mayores al ser aplicado para calcular la orientación de un cuerpo que se mueve con gran intensidad.

Luinge et al. desarrollan en [87] un método que utiliza restricciones en el movimiento del codo para medir la orientación del antebrazo con respecto a la parte superior del brazo. Utilizan una unidad de medida inercial, compuesta por un acelerómetro triaxial y un giróscopo triaxial, en conjunto con un filtro de mínimos cuadrados que estima los errores de orientación de forma que el ángulo de adducción sea nulo. Este algoritmo fracasa a la hora de estimar la orientación de forma correcta ya que presenta errores de hasta 40° RMS.

Roetenberg et al. describen en [88] un filtro de Kalman complementario para estimar la orientación de los segmentos corporales mediante la fusión de los datos del giróscopo, acelerómetro y magnetómetro. Los errores de estimación de ángulo en condiciones de baja intensidad y media intensidad obtenidos son 1.4° and 2.6° RMS respectivamente.

Luinge et al. describen en [89] el diseño y rendimiento de un filtro de Kalman para estimar la inclinación a partir de las señales de un acelerómetro triaxial. Consiguen una precisión de 2° para condiciones de movimiento cuasiestáticas.

Favre et al. presentan en [90] dos métodos para fusionar los datos medidos por un giróscopo triaxial y un acelerómetro triaxial para medir rotaciones. Estos métodos calculan el cuaternion de orientación utilizando las medidas de aceleración

durante los instantes de movimiento cuasiestático y lo actualizan utilizando el cuaternión de rotación obtenido a partir de la velocidad angular medida durante los instantes dinámicos.

De forma similar, Sabati propone en [91] una técnica de interpolación aplicada a los cuaterniones de orientación para mejorar la precisión de la estimación de la orientación. Dicho método obtiene una precisión de 14.6° RMS durante un ciclo de marcha y 14.8° RMS durante dos ciclos. Adicionalmente, describe un algoritmo de fusión basado en cuaterniones y el filtro extendido de Kalman, el cual consigue una precisión de 4° RMS.

Concluimos el repaso al estado del arte con el trabajo realizado por Amasay et al., quienes describen en [92] un método muy simple que se basa en la descomposición de las componentes de la aceleración de la gravedad para calcular el ángulo de *pitch*. Consiguen un RMS de menos de 1° par condiciones cuasiestáticas. Este método es totalmente inválido para estimar la orientación de un cuerpo que se mueva con intensidad media o alta, ya que no aplica ningún tipo de fusión de la aceleración con otras magnitudes.

Contribuciones principales

Las principales contribuciones de nuestro trabajo están enumeradas a continuación:

- (i) Elaboración de un estudio comparativo entre doce de los algoritmos más relevantes utilizados para calibrar sensores MARG de tipo MEMS.
- (ii) Dos nuevos algoritmos de calibración, para giróscopos de tipo MEMS, que no requieren instrumentación compleja.
- (iii) Un nuevo algoritmo que calcula la distribución espacial de los datos utilizados para estimar los parámetros de calibración. Este algoritmo es utilizado para asegurar una distribución espacial homogénea de los datos de entrada de forma que los parámetros de calibración estimados tengan una mayor precisión.
- (iv) Cuatro nuevas propuestas para determinar la intensidad del movimiento que está siendo monitorizado usando sensores MARG de tipo MEMS. Estas nuevas propuestas están basadas en los algoritmos FRD, LTSD, MBGTD y MBCUSUM.
- (v) Elaboración de un estudio comparativo entre cinco de los métodos más conocidos para determinar la intensidad del movimiento y los cuatro nuevos algoritmos mencionados en el punto anterior. Este estudio ha mostrado que el algoritmo LTSD supera el grado de precisión del resto de algoritmos estudiados a la hora de discernir entre períodos de actividad e inactividad.

- (vi) Cinco nuevas propuestas para fusionar las estimaciones de orientación calculadas utilizando los datos medidos por los sensores MARG de forma separada. Estas nuevas propuestas están basadas en los filtros adaptativos NLMS, MNLMS, RLS, HRLS y QRDRRLS.
- (vii) Una nueva propuesta para estimar la orientación de un cuerpo usando una variante del algoritmo basado en el filtro de Kalman. Esta variante se basa en la modificación dinámica de los parámetros de diseño del filtro en función del grado de intensidad del movimiento al que está sujeto el cuerpo.
- (viii) Elaboración de un estudio comparativo entre todos las nuevas propuestas de estimación de la orientación mencionadas en los puntos anteriores y la versión estática del filtro de Kalman y el algoritmo de Madgwick [80]. Este estudio ha demostrado que nuestra versión dinámica del filtro de Kalman supera tanto a su versión estática como al algoritmo de Madgwick. También se ha comprobado que el algoritmo basado en el filtro HRLS obtiene las mejores estimaciones de orientación para un cuerpo que está siendo sometido a un movimiento de muy alta intensidad.

Trabajos publicados

Además, parte del trabajo presentado a lo largo de este documento ha dado lugar a las siguientes publicaciones:

Artículos en revistas indexadas

- A. Olivares, J.M. Górriz, J. Ramírez and G. Olivares, "Accurate human limb angle measurement: sensor fusion through Kalman, least mean squares and recursive least-squares adaptive filtering". *Meas. Sci. Technol.* 2011, 22, doi:10.1088/0957-0233/22/2/025801.
- A. Olivares, G. Olivares, F. Mula, J.M. Górriz, and J. Ramírez. "Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications". *Journal of Systems Architecture*, 57(10):905-915, nov. 2011.
- Olivares A., Ramírez J., Górriz J.M., Olivares G. and Damas M. "Detection of (In)activity Periods in Human Body Motion Using Inertial Sensors: A Comparative Study". *Sensors*. 2012; 12(5):5791-5814.

Capítulos de libros

- Alberto Olivares, Gonzalo Olivares, J. M. Górriz and J. Ramírez, "Human Body Position Monitoring" Chapter 16 of *Recent Advances on Biomedical Signal Processing*, Bentham Science Publishers, pp 77 – 92, 2010.
- Alberto Olivares. 'Periféricos con acelerómetros', *Periféricos avanzados*, Editorial Garceta, Madrid, ISBN: 978-84-15452-04-1.

Conferencias internacionales

- Alberto Olivares, Gonzalo Olivares, J. M. Gorriz, and J. Ramirez, "High-efficiency Low-cost Accelerometer-aided Gyroscope Calibration", International Conference on Test and Measurement, Hong Kong, China, December 2009. vol.1, pp 354–360, ISBN: 978-1-4244-4699-5.
- A. Olivares, J.M. Gorriz, G. Olivares, J. Ramirez and P. Gloesekoetter, "A Study of Vibration-Based Energy Harvesting in Activities of Daily Living", published at 4th International ICST Conference on Pervasive Computing Technologies for Healthcare, ISBN: 987-963-9799-89-9, Munich, Germany, March 2010.
- Alberto Olivares, J.M. Gorriz, J. Ramirez and Gonzalo Olivares, "Accurate Human Limb Angle Measurement in Telerehabilitation: Sensor fusion through Kalman, LMS and RLS Adaptive Filtering", Ambient Intelligence and Future Trends, International Symposium on Ambient Intelligence (ISAMI 2010). Guimaraes, Portugal, June 2010. ISBN: 978-3-642-13267-4. pp 97–104.
- Alberto Olivares, J.M. Gorriz, J. Ramirez and Gonzalo Olivares, "Sensor Fusion Adaptive Filtering for Position Monitoring in Intense Activities", 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010). San Sebastian, Spain, June 2010. ISBN: 978-3-642-13768-6. pp 484–491.

Conferencias nacionales

- Alberto Olivares, Gonzalo Olivares, J. M. Górriz, J. Ramírez, "Wagyromag: red inalámbrica de sensores iniciales para el registro y procesamiento de movimientos corporales", I Simposio en Computación Empotrada, Congreso Español de Informática (CEDI), Valencia, Spain, September 2010.

Estructura del documento

El documento está dividido de la siguiente forma; el capítulo 2 presenta la instrumentación empleada así como los fundamentos teóricos de los sensores iniciales y magnéticos. Asimismo se explican de forma breve las distintas funcionalidades del firmware incluido en el mismo y de la herramienta desarrollada para visualizar los datos en tiempo real; el capítulo 3 está dedicado íntegramente a los algoritmos de calibración que se aplican para transformar los datos en crudo en datos en unidades físicas, así como para compensar los efectos indeseados presentes en la salida de los sensores; el capítulo 4 describe cómo detectar la intensidad del movimiento medido por los sensores. La detección de la intensidad es un paso muy importante para aumentar la precisión de la estimación de la orientación. En este capítulo realizamos un estudio comparativo entre una serie de métodos ya aplicados en este campo y cuatro algoritmos utilizados para detectar cambios abruptos en otras aplicaciones de tipo industrial; el capítulo 5 incluye el grueso del trabajo llevado a cabo. En él describimos las diferentes maneras de representar y estimar la orientación de un cuerpo respecto a un sistema de referencia. Volvemos a realizar un estudio comparativo entre algoritmos muy conocidos y los propuestos por nosotros; seguidamente, en el capítulo 6 discutimos sobre los posibles campos de aplicación de nuestro sistema, haciendo especial énfasis en la monitorización del movimiento del cuerpo humano; finalmente, el capítulo 7 incluye las conclusiones generales de nuestro trabajo y los objetivos que han sido cubiertos así como aquellos que se prevé cubrir en un futuro.

Chapter 1

Introduction

1.1 General

A key factor in analysis of human kinematics is selecting the phenomena to be sensed and measured. Let's take a look to the definition of kinematics included in the Encyclopædia Britannica [4].

"Kinematics, branch of physics and a subdivision of classical mechanics concerned with the geometrically possible motion of a body or system of bodies without consideration of the forces involved (i.e., causes and effects of the motions).

...

When a particle moves on a curved path, a description of its position becomes more complicated and requires two or three dimensions. In such cases continuous descriptions in the form of a single graph or mathematical formula are not feasible. The position of a particle moving on a circle, for example, can be described by a rotating radius of the circle, like the spoke of a wheel with one end fixed at the centre of the circle and the other end attached to the particle. The rotating radius is known as a position vector for the particle, and, if the angle between it and a fixed radius is known as a function of time, the magnitude of the velocity and acceleration of the particle can be calculated".

...

Human body motion describes trajectories in three dimensions and some movements, specially those of the limbs, can be modeled as rotations. As it comes from

the definition, if the magnitude of the velocity and acceleration are known we can determine the angle of the position vector of the particle, i.e. we can determine the positioning angles of the object which is subject to a motion. Therefore, our measurement device has to be able to sense the acceleration and angular velocity of a moving object.

Another important factor is the reference point from which we want to make the measurements. The reference point can be fixed in space or included in the object. If we choose the former, a possible option is setting a room which has cameras acting as observers placed in different fix positions. Such a configuration is employed in [5]. Vicon® cameras [6] and Microsoft's Kinect [7] are regularly used in camera-based motion analysis systems [8–14]. On the other hand, if we choose the latter, we need a device that is attached to human body and measures its relative acceleration and angular rate. Hence, our system needs from inertial sensors, namely acceleration sensors (accelerometers) and angular rate sensors (gyroscopes). Our work will be based on a set of devices that can be attached to different parts of human body and compute their positioning angles, so we will be using inertial sensors. Such devices including inertial sensors are generally known as Inertial Measurement Units (IMUs).

If we use triaxial accelerometers and gyroscopes our IMU will have six degrees of freedom. This is enough to sense the relative motion of objects, however we lack from a heading reference and, therefore, we can not fully determine the initial point in space of the object. To do so, we need from a sensor that is able to measure the heading relative to the Earth's coordinate frame. Compasses have been successfully employed for centuries to point the heading angle with respect to the Earth's magnetic North pole. Then, if we assemble a triaxial magnetic field sensor (magnetometer) in our device in addition to the triaxial accelerometers and gyroscopes, we will be able to measure the heading angle, and, consequently, our IMU will have nine degrees of freedom. IMUs including magnetic sensors are usually referred to as Magnetic Inertial Measurement Units (MIMUs).

1.2 Motivation

We started thinking about the possibility of designing and building systems to monitor the orientation and the movement of rigid bodies almost four years ago. We were contacted by Telefónica I+D which wanted to develop a knee telerehabilitation system and needed to outsource the design and implementation of the system.

Once the first prototype was developed, I started working to develop calibration and attitude estimation algorithms to process the measured data.

Right after the end of our collaboration with Telefónica I+D, we were funded by the AmIVital project[1] and, therefore, we were able to continue studying orientation algorithms oriented to develop more complex telerehabilitation, sports monitoring and teleassistance applications.

A very important factor that motivated us to develop this system comes from its direct application to the healthcare field.

As it will be later explained, the study and monitoring of human body motion and position has many applications in modern medicine, since many diseases cause motion disorders and/or can be diagnosed analyzing them. For example, our system can be applied to detect tonic-clonic epileptic seizures and bradykinesia, dyskinesia and tremor episodes in Parkinson's patients. It can also be used to prevent and detect falls among the elderly and analyze sleep disorders.

Therefore, the fact that our system could be applied to improve the quality of life of many patients was the main reason to design and develop it.

1.3 Goals

The main objectives at the beginning of our work were focused on studying calibration algorithms as well as algorithms to estimate the orientation of moving bodies to process the data measured by inertial and magnetic MEMS sensors. If we classify them according to both work fields, we get:

- **Calibration:**

- Carrying out a complete study of the state of the art of calibration algorithms for MEMS magnetometers, accelerometers and gyroscopes.
- Selecting and implementing the most cited algorithms to analyze their accuracy and performance by means of a comparative study.
- Proposing new calibration procedures not requiring complex instrumentation.

- **Orientation estimation:**

- Carrying out a complete study of the state of the art of existing algorithms.
- Proposing new algorithms based on alternative adaptive filtering approaches and comparing their performance with a set of the most popular algorithms developed during the last years.
- Proposing algorithms that modify their parameters dynamically depending on the intensity of the movement being measured.

Once these initial goals are fulfilled, we intend to look for applications connected with healthcare in which we can apply our system.

1.4 Project structure

If we consider the development of a human body motion monitoring system and its applications as a whole project, we can divide it in layers to help putting our work into context. As we said before, the core of our work is focused on signal processing of data gathered by inertial and magnetic MEMS sensors and testing and developing calibration algorithms. However, the project has many other layers in which other members of the project are currently working. The content of each layer, which is depicted in figure 1.1, is described next:

- **Instrumentation (Hardware):** This layer includes the design and development of inertial measurement units as well as the use of other commercial devices. Along this project we have worked with Wagyro and Wagyromag (both prototypes developed at the Computer Architecture and Computer Technology Department), Equivital[15] and Smartphones such as the iPhone 4, Samsung Galaxy Nexus and Samsung Galaxy SII.
- **Firmware:** This layer includes the design and implementation of the code supporting Wagyromag's functionalities. Along this project we have worked with a PIC from Microchip's 24F Family [16].
- **Signal processing:** This layer includes the design and development of optimal calibration algorithms as well as the analysis and study of activity intensity detection and orientation estimation algorithms. Along this project we have worked exhaustively in this layer.
- **Software:** This layer includes the design and development of software tools that integrate all the signal processing algorithms to gather, calibrate, process, plot and store data coming from the inertial measurement units. Along this project we have developed a series of applications in different platforms, e.g. C#, Java, Labview and Android.
- **Knowledge inference:** This layer includes the use of calibrated and processed data to develop automatic and intelligent systems that distinguish different kinds of daily activities (walking, running, sitting down, etc.) and detect events (falls, epileptic seizures, etc.).
- **Applications:** This layer includes the design of applications for the complete system. Along this project the system has been used in a telerehabilitation

service for knee after surgery patients and will soon be used to monitor and detect tonic-clonic epileptic seizures.

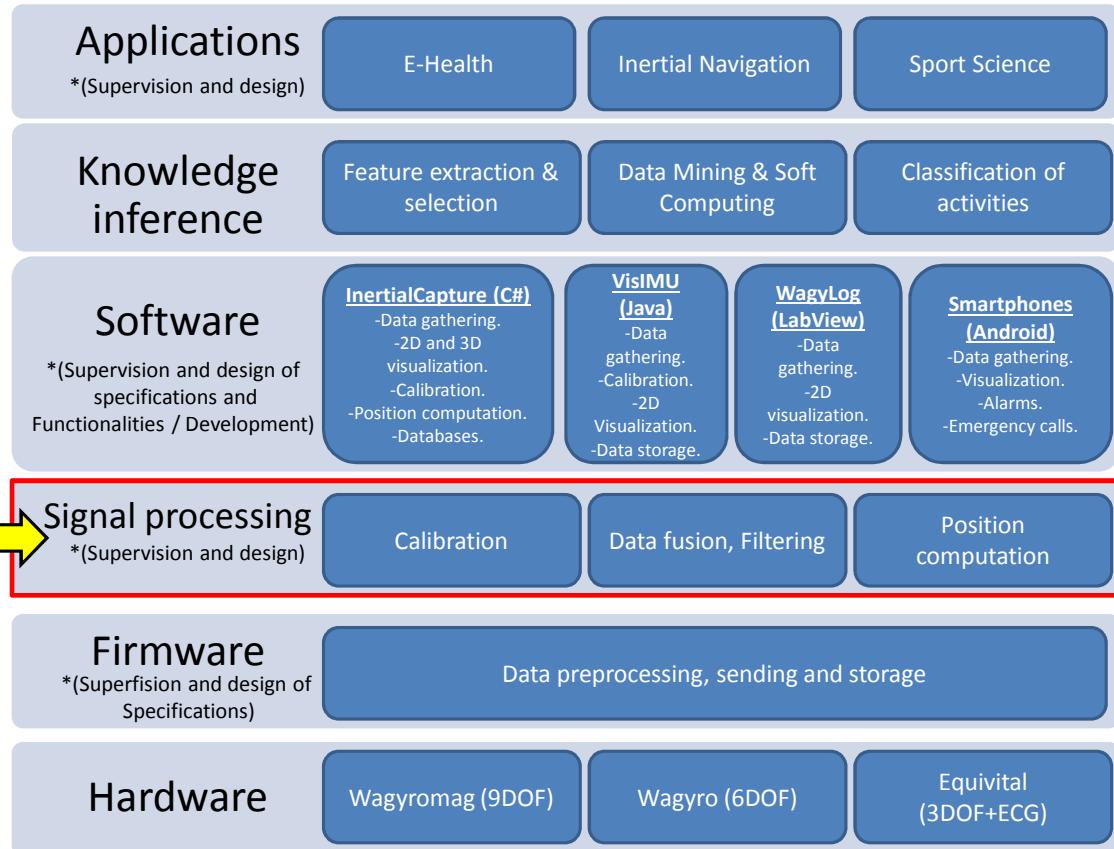


Figure 1.1: Layer structure of the human body monitoring project. Signal processing layer is highlighted as it includes the core of our work.

1.5 State of the art

1.5.1 State of the art at the beginning of the project

If we analyze the state of the art of the existing IMUs four years ago that could be used to monitor human body motion, we see that it was very hard to find completely wireless devices. The very few existing ones were very expensive. Moreover, they were all based on high power consumption protocols such as Bluetooth, or were not able to transmit data in real time.

Among the few existing devices, we could find the 2009 versions of Xbus Master(now Xbus Kit [17]), Dynaport and Kinetisense (which actual versions can be found in [18] and [19], respectively), Physilog [20], Nike+ [21] and fitbit [22].

Given that no devices fulfilled all the initial requirements, we developed our own prototype.

Regarding the calibration algorithms, there was already a large number of published algorithms, but, once we analyzed them in detail, we observed that most of them were mere repetitions or slight variations of classic methods. Hence, we narrowed down the number of selected algorithms in order to implement them and carry out a comparative study.

We observed that calibration algorithms could be classified in two groups; those based on the minimization of an error function to estimate the optimal calibration parameters, and those that compute the calibration algorithms by applying simple algebraic transformations.

At this point in time a problem appeared. We saw that most gyroscope calibration algorithms needed from complex multi-axis numerical rate tables. These tables had a market price over 10,000 euros and, therefore, we had to look for other alternatives to calibrate the gyroscope using simple instrumentation.

With respect to orientation estimation algorithms, we observed that the majority of existing algorithms were based on sensor fusion approaches using the Kalman filter.

Our initial tests made us realize that the precision of, e.g. the Kalman filter applied to estimate the orientation of a body subject to large accelerations, varied considerably with respect to the case in which the accelerations were very low or mild. Thus, we saw the importance of analyzing the performance of sensor fusion filters applied on signals showing large variations of intensity. Therefore, we decided to study the possibility of applying other existing adaptive filters such as the Least Mean Squares filter (LMS), the Recursive Least Squares (RLS) filter and some of their variants, to compare their performance with the Kalman Filter.

We also observed that, to our knowledge, there were no published works indicating the performance of sensor fusion with respect to the intensity of the motion being monitored, since only a few of the most recent works at the time were starting to suggest the possibility of applying complementary filters having different gains depending on the degree of motion intensity.

1.5.2 General view of the state of the art

1.5.2.1 Instrumentation

It is very usual to integrate accelerometers and gyroscopes with a processing unit and memory modules to form Inertial Measurement Units (IMUs). If the device integrates also a magnetometer, then it is known as a Magnetic Inertial Measurement Unit (MIMU). There is a wide range of commercial IMUs and MIMUs each one having different specifications. Most of them are produced in low scale, so the prices are quite large.

As we said in the introduction, we finally opted to develop our own MIMU, but before making the decision we took a look to the market and we carried out an analysis of the state of the art of commercial and experimental units that were for sale. We observed that, IMUs and MIMUs can be chosen according to the following criteria:

- *Size*: There is a large variation of size between inertial devices employed in commercial aviation and, for example, human body motion monitoring. A reduction of size goes hand in hand with a reduction of the precision of the sensors.
- *Weight*: Bulky devices are usually quite heavy as they are protected with robust cases. Weight might be an important constraint of the application in which the device will be used.
- *Data transmission*: We can choose between wired or wireless transmission. There is also a variety of protocols employed for wireless data transmission (Bluetooth 2.0, Bluetooth 3.0, Zigbee, ANT+, etc.).
- *Processor*: Some devices include a microcontroller or a small microprocessor to give the device local processing capability.
- *Data storage*: Some devices include integrated memory modules and/or a memory card slot. Data storage is mandatory in those applications in which data can not be transmitted in real time.
- *Sensors' specifications*: The specifications of the sensors need to be considered before choosing them. Errors, sensitivity, power supply and range are some of the characteristics to be contemplated.

Examples of recent commercial IMUs and MIMUs with different sensor specifications, quality and features can be found in [23–30]. A complete list of commercial and experimental IMUs and MIMUs is also available in [31].

New devices are constantly appearing in the market since the MEMS sensor market is growing and evolving incredibly fast. In addition, nowadays most high-end smartphones include triaxial accelerometers, gyroscopes and magnetometers. This offers the opportunity to develop applications using the algorithms described along this work and increase the niche of human body motion monitoring applications as in a few years almost every person will be carrying a MIMU with themselves. However, smartphones are not an ultimate alternative to MIMUs since some complex motion and position monitoring applications require various MIMUs attached to different parts of the body forming a wireless Personal Area Network (PAN).

1.5.2.2 Calibration

Hundreds of works related to calibration of MEMS sensors have been published during the last decade. However, most of them are mere repetitions or slight variations of previous methods. Along this section we name some of the most cited and most important methods in the literature.

We begin the review with the most cited work by far; Ferraris et al. present in [32] a method to calibrate triaxial accelerometers and gyroscopes without using any complex maneuvers nor instrumentation. This work has been used as the basis for many other methods developed in subsequent years. The calibration of the accelerometer is carried out by setting its axis in six different positions and applying simple algebraic expressions. The gyroscope is calibrated by substituting the need of an angular rate reference by a set of known rotations. We will explain this method in depth in chapter 3 together with Wong's et al. method [33] which is also based on the six positions procedure.

Only six calibration parameters (scale factors and biases) can be computed following the six positions procedure. Therefore, many other works gather data by putting the sensors in many different positions. These methods are known as multiposition methods and use the data to minimize an error or cost function applying a minimization procedure. Examples of multiposition methods can be found in [34–52]. These works only differ in the employed sensor model—which determines the calibration parameters to be estimated—the method to gather the data—random positions or predefined positions—the instrumentation—complex and expensive triaxial rate tables, simple mechanical devices or no devices at all—and the error or cost function they employ and the method they use to minimize such function—Gauss-Newton, linear least squares, non linear least squares, gradient descent, Levenberg-Marquardt, etc.

There is also a group of works that are based in different techniques such [53] and [54] which include simple empirical algorithms based on basic algebraic operations. A method based on a Mamdami type fuzzy batch least square algorithm used

to estimate the calibration bias and the scaling parameters can be found in [55] and the rather famous two-step procedure is presented in [93] and also used in [56]. [57] and [58] both develop algorithms based on digital FIR filters.

Other works are focused on solving just an specific part of the calibration procedure. For example, [59] uses a method based on conjugate gradient optimization to find the horizontal error angles and [60] issues the inter-triad calibration between different sensors.

1.5.2.3 Determination of motion intensity

Determining the intensity of the motion being monitored is a topic which has not been issued by many authors. However, it is a very important factor to improve the accuracy of orientation estimation systems. For example, detecting the instants in which the body is static is necessary to recompute calibration parameters [61] such as the bias, reset trajectory drifts in inertial navigation [62, 63], or adapting the design parameters of sensor fusion filters.

Detection algorithms can be classified according to the sensor they use as an input. The Acceleration Moving Variance Detector (AMVD) proposed in [67], and the Acceleration Magnitude Detector (AMD) implemented in [64], use the acceleration signals to carry out the classification. This fact may limit the detection of possible instants where there is no acceleration but the gyros are measuring angular rate. On the other hand, the Angular Rate Energy Detector (ARED) employed in [65] uses the angular velocity signals as the input, which may also lead to erroneous classification of moments where there is little or no angular rate but accelerometers are sensing acceleration as inactivity periods. The Stance Hypothesis Optimal Detector (SHOD) proposed in [66] uses both the acceleration and angular velocity signals to increase the precision of the detector and, finally, the Filtered Rectifier Detector (FRD) employed in [67] has a flexible input (acceleration and angular rate magnitudes or a linear combination of both).

A comparative study among some of the aforementioned algorithms is also presented in [66], where a performance comparison of the detectors, using real signals gathered from different sensors, is shown. The mathematical definition of the detectors is very rigorous, however, as the authors state, the amount of signals used to compare the methods is rather low, making the study non-complete in statistical terms.

1.5.2.4 Estimation of the orientation of dynamic bodies

Orientation estimation methods started to be developed in the seventies for space missions. Algorithms were developed to determine the orientation of the spaceship

with respect to different reference frames. Examples of these early methods can be found in [68] and [69] which describe the TRIAD and QUEST methods respectively. The TRIAD algorithm provides a deterministic solution for the attitude based on two vector observations and two vector references (which, in our case, can be the a priori known gravity and local Earth magnetic field vectors) which are used to compute the attitude matrix. The QUEST algorithm minimizes a loss function to find the optimal quaternions representing the attitude. In our context (determination of attitude using magnetic and inertial sensors), these methods solely use the accelerometer and the magnetometer. To avoid problems related to high intensity motion, [70] fuses QUEST algorithm with integration of angular rate using an Extended Kalman Filter (EKF).

An alternative to QUEST is presented in [71]. The algorithm, denominated Factored Quaternion Algorithm (FQA), unlike QUEST, avoids the use of magnetic data in calculations of roll and pitch as the magnetic disturbances may lead to errors in their computation. The use of magnetic data is restricted to the estimation of the yaw angle.

Many variations of these methods have been proposed during the last decades mainly by Malcolm D. Shuster [72, 73], F. Landis Markley [74–78], and Xiaoping Yun [79]. In [79] a Gauss-Newton iteration algorithm is employed to find the best quaternion which relates the measured accelerations and earth magnetic field in the body frame to calculated values in the earth coordinate frame. Such quaternion is afterwards fused with angular rate using an EKF. This approach provides good results but adds high computational cost due to the minimization procedure which has to be computed for every set of measurements.

Madgwick [80] also presents a quaternion based approach which estimates the observation (Earth's magnetic field and gravity) quaternion using a gradient-descent method and then fuses it with the dynamical quaternion (angular rate) using a complementary filter.

Examples of algorithms to compute Euler angles without employing quaternions can be found in [81–83].

A very complete survey of nonlinear attitude estimation methods can be found in [84] and a comparison between some of the most popular orientation filter algorithms is also found in [85].

1.5.2.5 Estimation of orientation applied to human body position and motion monitoring

Zhang et al. propose in [86] a Hybrid Dynamic Bayesian Network to model the nonlinear hip angle dynamics and a Gaussian Particle Filter to estimate the hip angle during gait cycles from the measurements of a wearable accelerometer which is attached to the thighs. They achieve a minimum error of 1.50° RMS in the orientation. However, since the angle estimation is only based on the accelerometer this

algorithm will likely show larger errors when applied under more intense motion conditions.

Luinge et al. present in [87] a method that uses constraints in the elbow to measure the orientation of the lower arm with respect to the upper arm. They use an IMU composed of a triaxial accelerometer and a triaxial gyroscope, and apply a least squares filter that estimates the orientation errors in a way that sets the adduction angle to zero. Their algorithm fails to estimate the orientation properly as it shows errors of up to 40° RMS.

Roetenberg et al. describe in [88] a complementary Kalman Filter design to estimate the orientation of human body segments by fusing gyroscope, accelerometer and magnetometer signals from MEMS sensors. The filter estimates the gyroscope error bias, the orientation error and the magnetic disturbance error. The average static and dynamic errors are 1.4° and 2.6° RMS, respectively.

Luinge et al. describe in [89] the design and performance of a Kalman filter to estimate inclination from the signals of a triaxial accelerometer. They achieve an accuracy of 2° RMS for quasi-static motion.

Favre et al. present in [90] two methods to fuse a triaxial gyroscope with a triaxial accelerometer to measure rotations. These methods compute the orientation quaternion using the acceleration measurements during quasi-static instants and update it using the rotation quaternion obtained from the angular rate measurements during dynamic instants.

Similarly Sabati proposes in [91] an interpolation technique applied to attitude quaternions to improve the accuracy of orientation and positioning estimates. It obtains an accuracy of 14.6° RMSE during one gait cycle and 14.8° RMSE during two gait cycles. Additionally, it describes a quaternion based extended Kalman filter (EKF) algorithm that achieves an accuracy close to 4° RMSE.

Favre et al. propose a method based on a leg movement to align two inertial measurement units fixed on the thigh and shank segments. They combine this method with their fusion algorithm [90], to obtain an error within the [0°, 3.4°] range for different knee quasi-static movements. They employ an IMU composed of a triaxial accelerometer and a triaxial gyroscope.

We conclude the review of the state of the art with the work by Amasay et al, who describe in [92] a simple method based only on the decomposition of gravity acceleration to estimate the pitch angle. They obtain a RMSE of less than 1° only during quasi-static conditions. This simple method is absolutely non valid to estimate the orientation of a body moving with medium to high intensity as it does not apply any sensor fusion strategies.

1.6 Main contributions

The main contributions of our work are listed below:

- (i) A comparative study between twelve of the most relevant calibration methods for MEMS magnetic and inertial sensors published within the last years.
- (ii) Two new calibration algorithms for MEMS gyroscopes which do not require complex instrumentation.
- (iii) A new algorithm that computes the spatial distribution of data needed to estimate the calibration parameters of MARG sensors. This algorithm is used to ensure a homogeneous spatial distribution of the input data so the calibration parameters are estimated with a higher degree of accuracy.
- (iv) Four novel approaches to determine the intensity of motion being monitored using MEMS magnetic and inertial sensors. These new approaches are based on the FRD, LTSD, MBGTD and MBCUSUM algorithms.
- (v) A comparative study between five of the most known motion intensity detection algorithms and the four aforementioned new algorithms. This study has revealed that the LTSD algorithm proposed by us outperforms all of the existing methods.
- (vi) Five new approaches to fuse the separate estimations of the orientation of a body computed using different MEMS MARG sensors. These novel approaches are based on the NLMS, MNLMS, RLS, HRLS and QRDRRLS adaptive filters.
- (vii) A new approach based on a version of the Kalman filter which dynamically varies its design parameters according to the degree of intensity of the motion being sensed.
- (viii) A comparative study between all the proposed orientation estimation algorithms and two of the most popular algorithms in literature: the static Kalman Filter and Madgwick's [80] algorithm. This study has revealed that the dynamic version of the Kalman filter outperforms both its static existing version and Madgwick's algorithm. It has also shown that the HRLS algorithm outperforms the regular Kalman Filter to estimate the orientation of a body subject to very intense motion.

1.6.1 Published works

In addition, part of the work we have presented along this document has been published and is already available for the research community.

1.6.1.1 Articles in Refereed Magazines

1. A. Olivares, J.M. Górriz, J. Ramírez and G. Olivares, "Accurate human limb angle measurement: sensor fusion through Kalman, least mean squares and recursive least-squares adaptive filtering". *Meas. Sci. Technol.* 2011, 22, doi:10.1088/0957-0233/22/2/025801.
2. A. Olivares, G. Olivares, F. Mula, J.M. Górriz, and J. Ramírez. "Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications". *Journal of Systems Architecture*, 57(10):905-915, nov. 2011.
3. Olivares A., Ramírez J., Górriz J.M., Olivares G. and Damas M. "Detection of (In)activity Periods in Human Body Motion Using Inertial Sensors: A Comparative Study". *Sensors*. 2012; 12(5):5791-5814.

1.6.1.2 Book chapters

1. Alberto Olivares, Gonzalo Olivares, J. M. Górriz and J. Ramírez, "Human Body Position Monitoring" Chapter 16 of *Recent Advances on Biomedical Signal Processing*, Bentham Science Publishers, pp 77 – 92, 2010.
2. Alberto Olivares. 'Periféricos con acelerómetros', *Periféricos avanzados*, editado por Alberto Prieto. Editorial Garceta, Madrid, 2012. ISBN: 978-84-15452-04-1.

1.6.1.3 International Conferences

1. Alberto Olivares, Gonzalo Olivares, J. M. Górriz, and J. Ramirez, "High-efficiency Low-cost Accelerometer-aided Gyroscope Calibration", International Conference on Test and Measurement, Hong Kong, China, December 2009. vol.1, pp 354–360, ISBN: 978-1-4244-4699-5.
2. A. Olivares, J.M. Górriz, G. Olivares, J. Ramirez and P. Gloesekoetter, "A Study of Vibration-Based Energy Harvesting in Activities of Daily Living", published at 4th International ICST Conference on Pervasive Computing Technologies for Healthcare, ISBN: 987-963-9799-89-9, Munich, Germany, March 2010.
3. Alberto Olivares, J.M. Górriz, J. Ramirez and Gonzalo Olivares, "Accurate Human Limb Angle Measurement in Telerehabilitation: Sensor fusion through Kalman, LMS and RLS Adaptive Filtering", Ambient Intelligence and Future Trends, International Symposium on Ambient Intelligence (ISAMI 2010). Guimaraes, Portugal, June 2010. ISBN: 978-3-642-13267-4. pp 97–104.

4. Alberto Olivares, J.M. Gorriz, J. Ramirez and Gonzalo Olivares, "Sensor Fusion Adaptive Filtering for Position Monitoring in Intense Activities", 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010). San Sebastian, Spain, June 2010. ISBN: 978-3-642-13768-6. pp 484–491.

1.6.1.4 National Conferences

1. Alberto Olivares, Gonzalo Olivares, J. M. Górriz, J. Ramírez, "Wagyromag: red inalambrica de sensores inerciales para el registro y procesamiento de movimientos corporales", I Simposio en Computación Empotrada, Congreso Español de Informática (CEDI), Valencia, Spain, September 2010.

1.7 Document structure

This document presents a slight variation from the standard structure in which there is a chapter including all the theoretical basis, followed by another chapter including all the experiments, results and their discussion and a final chapter drawing the conclusions. Instead, every separate chapter has its own introduction, theoretical fundamentals, experiments, results, discussion of results and conclusions. This structure makes every chapter self-contained and, therefore, they can be read separately. We have proceeded this way to avoid the reader to go through dense chapters including a long string of equations, graphs and results which may make him lose the thread of the reading.

Therefore the document is divided as follows; chapter 2 explains how to design and develop a system to gather raw magnetic and inertial data. This includes the hardware, the firmware and the software we developed for this purpose; chapter 3 deals with calibration algorithms that can be applied to transform raw data into meaningful data in physical units, as well as to compensate for undesired effects in the output of the sensors. We carry out a comparative study between different existing algorithms at the same time we propose a set of novel calibration procedures; chapter 4 describes how to detect the intensity of the motion being gathered by the sensors. Detecting the intensity is a key step to increase the precision of attitude estimation. We also carry out a comparative study between well known methods and a set of new approaches that we propose; chapter 5 includes the core of our research. We describe different ways to represent the attitude of a body and how to estimate it. We, again, carry out a complete comparative study between very well-known algorithms and our proposed methods; chapter 6 talks about some of the possible fields of application of our system, making special emphasis on human body motion monitoring; finally, chapter 7 includes the general conclusions and summarizes the most relevant parts of everything that is presented in the rest of the chapters. We

also discuss about future research lines and how we will orient forthcoming work.

Chapter 2

Instrumentation and tools

As we said in the introduction, accelerometers, gyroscopes and magnetometers are needed to obtain the orientation of a body with nine degrees of freedom. Along this chapter we will talk in depth about their working principles and characteristics. We will also introduce and describe the employed tools and instrumentation as well as the firmware and software used to gather, send, store and visualize inertial and magnetic data to subsequently process it.

2.1 MARG Sensors

MARG (Magnetic, Angular Rate and Gravity) sensors is the general name usually given to embrace magnetometers, gyroscopes and accelerometers.

2.1.1 Accelerometers

Accelerometers sense the movement of an object in an inertial space. Their behavior differs slightly from displacement sensors, since the latter measure the relative movement between two points. However, it is essential to use an element being able to sense the displacements that are produced inside the accelerometer, so a measure of displacement or force is obtained. This measure will be used in turn to calculate the resultant acceleration.

The generic structure of an accelerometer is composed by a seismic mass m

which is joint to the internal surfaces by an elastic structure in such a way that it moves with the same acceleration suffered by the test object. This structure can be seen in Figure 2.1. The mass can only move in one direction, which is set to be the sensitive axis of the accelerometer. As we can observe in Figure 2.2, the displacement of the mass in the direction of \hat{n} is proportional to the difference between the dynamic acceleration a and the gravity static acceleration g . The displacement will vary the voltage at the output terminals, and, therefore, such a variation is interpreted as the acceleration suffered by the object to which the sensor is attached, after a proper calibration process.

The structure and basic working principles of a displacement based accelerometer can be modeled as a system composed by a mass joint to a spring and a resort. The resultant displacement of the mass responding to an applied acceleration is modeled as a second order system. Therefore, the system model can be obtained as a direct application of Newton's Law:

$$u(t) = m \frac{\partial^2 x(t)}{\partial t^2} + b \frac{\partial x(t)}{\partial t} + kx(t) \quad (2.1)$$

where $x(t)$ is the relative displacement of the mass m , b is the damping constant, k is the elastic constant of the spring and $u(t)$ is the applied force.

By applying the Laplace Transform on (2.1) considering null initial conditions ($x(0) = 0, \dot{x}(0) = 0$), we can calculate the system's transfer function, which input and output are the force $u(t)$ and the relative displacement $x(t)$ respectively, as follows

$$u(s) = x(s)[ms^2 + bs + k] \quad (2.2)$$

$$H(s) = \frac{x(s)}{u(s)} = \frac{1}{ms^2 + bs + k} = \frac{1/m}{s^2 + \frac{b}{m}s + \frac{k}{m}} \quad (2.3)$$

which can be finally expressed as:

$$H(s) = \frac{K_e \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.4)$$

where $K_e = \frac{1}{k}$, $\omega_n = \sqrt{k/m}$ is the natural oscillation frequency of the spring and $\xi = b/(2\sqrt{mk})$ is the damping coefficient.

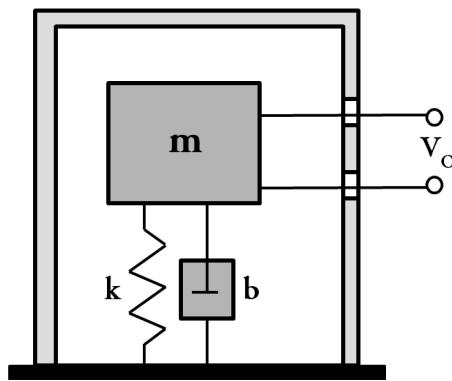


Figure 2.1: Generic structure of a moving mass accelerometer. A mass m is joint to the internal surface by a spring having an elastic constant k and a resort with a damping constant b . The displacement x causes variations in the output voltage V_o .

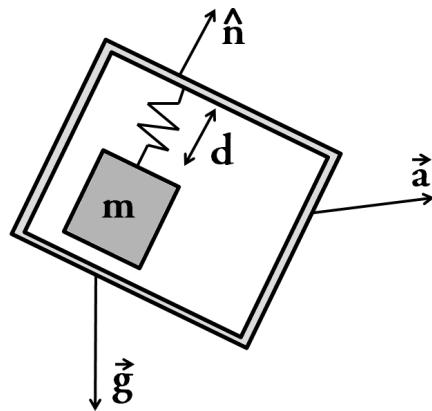


Figure 2.2: Single axis accelerometer. A mass is hanging attached to a spring. Distance d between the mass and the casing is measured as a function of the acceleration and the direction of gravity with respect to the direction of the measured displacement. Unit vector \hat{n} represents the sensitive axis of the sensor.

We can equally express the equations in the system's state space. For this purpose we assume the input of the system to be a sinusoidal force:

$$u(t) = u_0(t) \sin(\omega t) \quad (2.5)$$

Hence, the differential equation representing the system will be:

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = u(t) \quad (2.6)$$

$$\ddot{x}(t) = \frac{1}{m}(-kx(t) - b\dot{x}(t) + u(t)) \quad (2.7)$$

What leads to the following system's state equation:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & \frac{-b}{m} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} u(t) \quad (2.8)$$

where $x_1(t) = x(t)$ and $x_2(t) = \dot{x}(t)$. The resulting output equation is:

$$y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \quad (2.9)$$

If, for example, we represent the system's transfer function Bode Diagram for $m = 10$ grams, $b = 0.05$ and $k = 300$ N/m, we can clearly observe that the accelerometer has an accurate response for all frequencies under ω_n , which for the previous values is:

$$\omega_n = \sqrt{\frac{k}{m}} = 0.5 \sqrt{\frac{300}{0.01}} = 173.2 \text{ rad/s.} \quad (2.10)$$

that, as depicted on Figure 2.3, matches with the frequency value where the phase changes abruptly. Frequency values over this threshold will likely lead to a remarkable difference between the actual acceleration and the sensed value.

Once the accelerometer's generic structure and model are described, we will do a brief description of the different kinds of accelerometers, according to its size and physical sensing principle used to measure the mass displacement, that can be found in the market. According to size criteria, accelerometers can be classified as MEMS and non MEMS.

Microelectromechanical Systems (MEMS) are, as its name suggests, systems that combine mechanical elements with electronic components which are produced in micrometric or nanometric scale. This technology is applied to accelerometers so to obtain a minimization of its size, allowing them to be integrated into a circuit. Their structure is based on the generic system previously presented with some slight modifications. The basic structure of a MEMS accelerometer consists of a suspended mass fixed to a substrate by suspension arms (see Figure 2.4). This kind of accelerometers offer a reasonable price-performance trade-off, yet their precision and robustness is many times lower than non MEMS accelerometers. Non MEMS accelerometers can't be integrated in a circuit due to their size, so they are used in applications where size is not a constraint such as industrial environments, aeronautic systems, and other systems needing great precision and robustness.

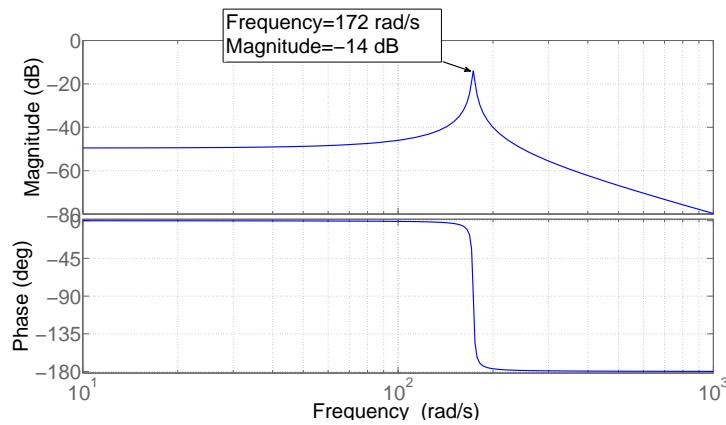


Figure 2.3: Bode's Diagram of the accelerometer frequency response.

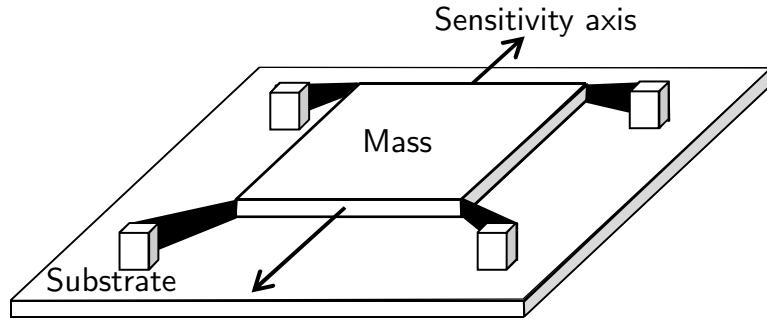


Figure 2.4: Diagram of the generic structure of MEMS accelerometers.

According to the mass displacement sensing method, accelerometers can be classified as resistive, capacitive, and piezoelectric. There exist other less popular sensing methods such as the ones based on optic fiber, servoaccelerometers and vibrating elements that will not be explained here.

Resistive accelerometers sense the mass displacement through a brush which moves along a potentiometer. The system may use magnetic, gas or viscous damping in order to minimize the acoustic noise generated by the contact of the brush with the potentiometer. Potentiometer-based resistive accelerometers typically have a frequency range from 20 to 60 Hz, depending on the stiffness of the spring, and develop a high level output signal. Their frequency response is lower than most of the accelerometers, usually varying from 15 to 30 Hz. A diagram of their architecture can be seen in Figure 2.5.

Capacitive accelerometers sense the mass displacement through variation of the capacity between two electrodes that are fixed to the internal surface and the moving mass respectively. Figure 2.6 shows a diagram of their architecture.

Piezoelectric accelerometers are normally used in applications where vibrations need to be measured, and in shock detection devices (such as cars' airbags). The piezoelectric crystals (usually quartz or ceramic) generate an electric charge when the accelerated mass applies a force on them. The quartz layers are precharged so there is a positive or negative variation when the applied force produces a change in the electric charge. Even though the sensitivity of the piezoelectric accelerometers is rather low compared to other types, they present the largest dynamic range of operation, being able to measure accelerations up to 1000 g, as well as a frequency response higher than 20 kHz. Figure 2.7 shows a diagram of their architecture.

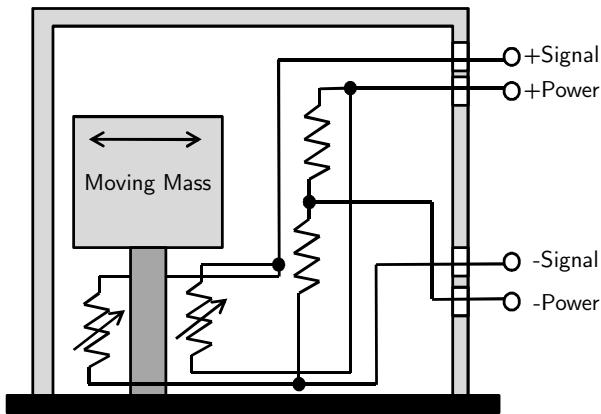


Figure 2.5: Resistive accelerometer architecture diagram. The flexion due to the movement of the mass causes variations in the sensitive resistors that are translated to variations in the output signal. Modified from [94].

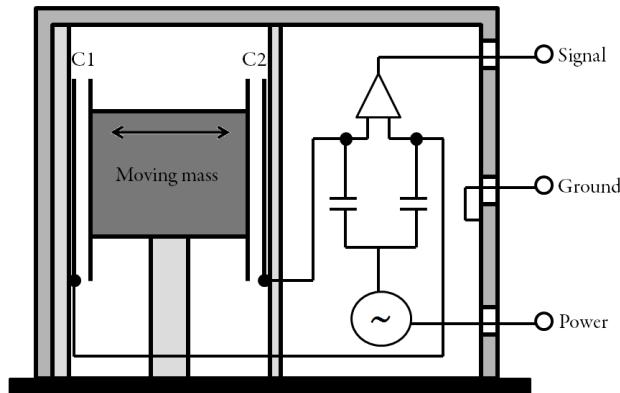


Figure 2.6: Capacitive accelerometer architecture diagram. The flexion due to the movement of the mass causes variations in the sensitive capacitors that are translated to variations in the output signal. Modified from [94].

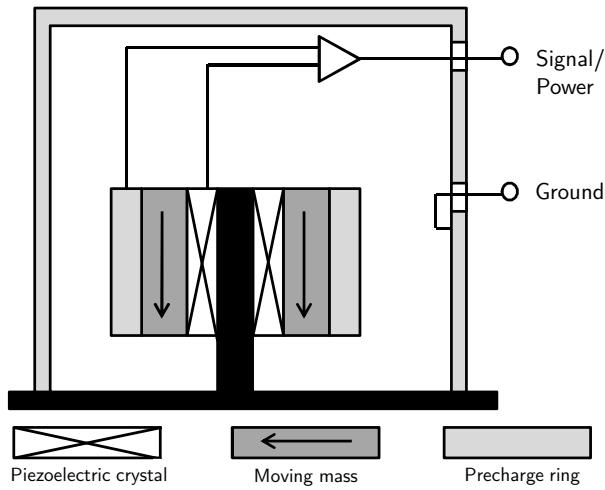


Figure 2.7: Piezoelectric accelerometer architecture diagram. The flexion of the crystals due to the movement of the mass causes variations in the electric charge that are translated to variations in the output signal. Modified from [94].

2.1.2 Gyroscopes

A gyroscope is a device which is essentially formed by a rotation symmetry object that spins around its symmetry axis. They are mainly used for measuring and maintaining the orientation of an object. When used in IMUs, gyroscopes are used to measure angular velocity. Gyroscopes used for this purpose are based on the Coriolis Effect.

The Coriolis Effect originates the relative acceleration that is suffered by an object, which is moving inside a non inertial rotating reference system, when there is a variation of its distance with respect to the spin axis. The Coriolis Force causes an acceleration or deceleration of the object moving along the radius with respect to the rotating system, that causes a variation in its trajectory. The Coriolis acceleration can be computed applying the following expression:

$$\mathbf{a}_c = -2\omega \times \mathbf{v} \quad (2.11)$$

where ω is the angular velocity of the rotating reference system and \mathbf{v} is the velocity of the moving object with respect to the rotating reference system.

Due to their miniaturization, it is impossible to build a MEMS gyroscope based on the classic rotating disc structure. We need then, a structure based on the Coriolis effect which is feasible to be built in nanometric scale.

Vibrating mass gyroscopes are the most usually employed MEMS gyroscopes. Their basic concept consists of a mass which is suspended by two arms that permit its oscillation. When an angular velocity is applied on the device, the mass, which is vibrating, will experiment a displacement caused by the Coriolis effect. If the displacement is known, then, after proper calibration, the angular velocity can be measured. The vibration of the mass is set so it moves with a velocity v , allowing the apparition of Coriolis Effect. This vibration is usually produced by a piezoelectric crystal that expands and contracts under a variation of voltage. Figure 2.8 shows the basic diagram of the vibrating mass gyroscopes.

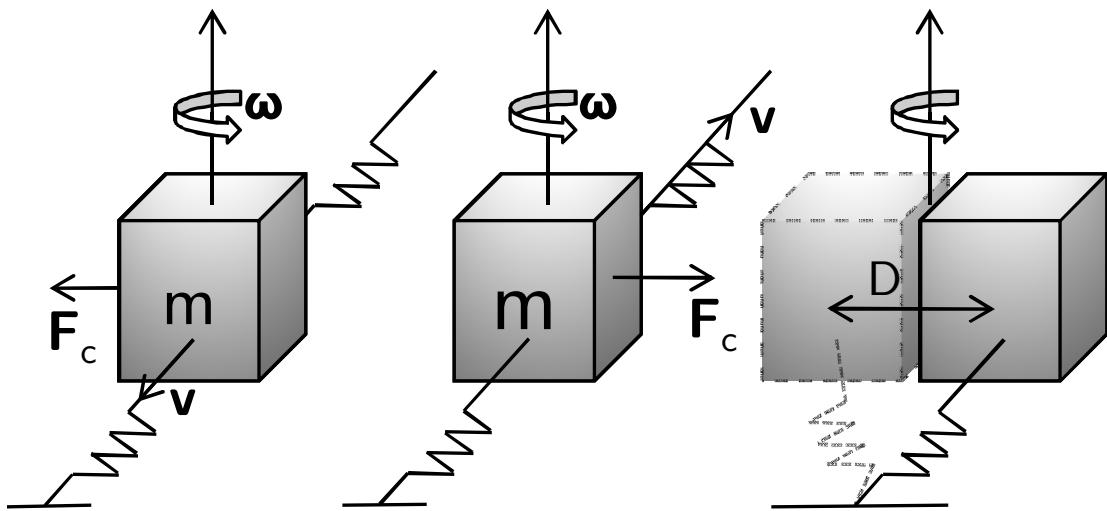
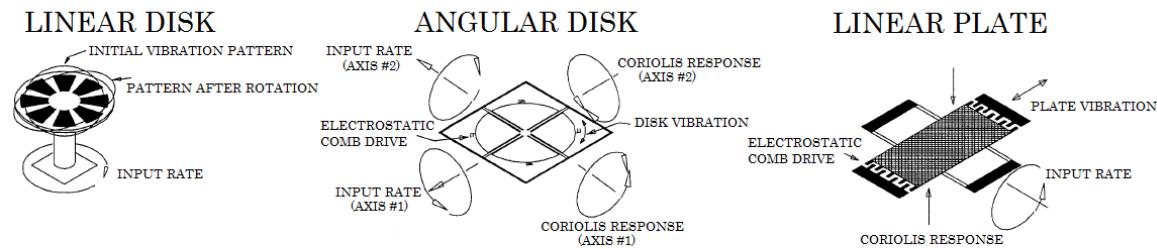


Figure 2.8: Basic structure of a vibrating mass gyroscope. The vibration of the mass causes it to move with a speed v in each one of the oscillation senses (left and middle figures), which makes the Coriolis Force cause a displacement D (figure on the right).

A Coriolis Vibratory Gyro (CVG) is a gyroscope based on the coupling of a structural, driven, vibrating mode into at least one other structural mode (pickoff) via Coriolis acceleration, where the Coriolis force arises from the motion of the vibrating structure relative to the CVG case fixed frame that is rotating relative to inertial space. Figures 2.9 and 2.10 depict the various types of CVG that have been commercially constructed.

Tuning fork gyroscopes are the most common vibrating mass MEMS gyroscopes. As its name says, they are based on a tuning fork, which arms are made vibrate, that

VIBRATING PLATES



TUNING FORKS

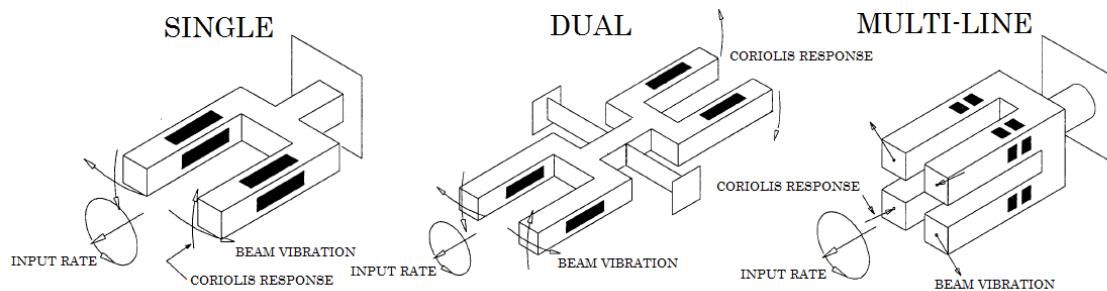
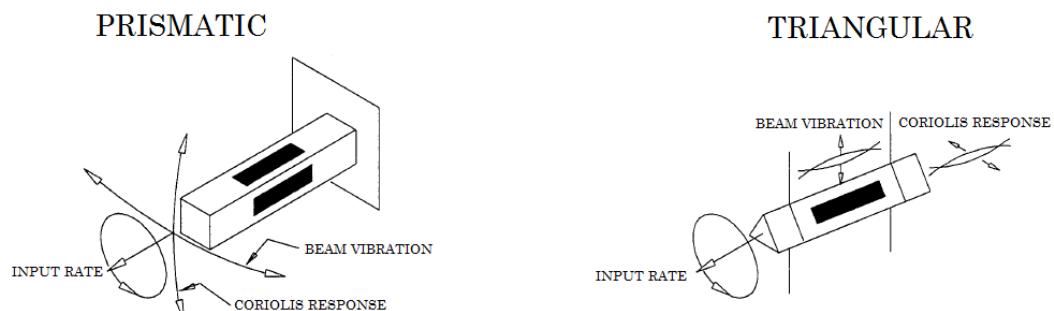


Figure 2.9: Vibrating Plate and Tuning Fork Coriolis Gyroscopes [95].

VIBRATING BEAMS



VIBRATING SHELLS

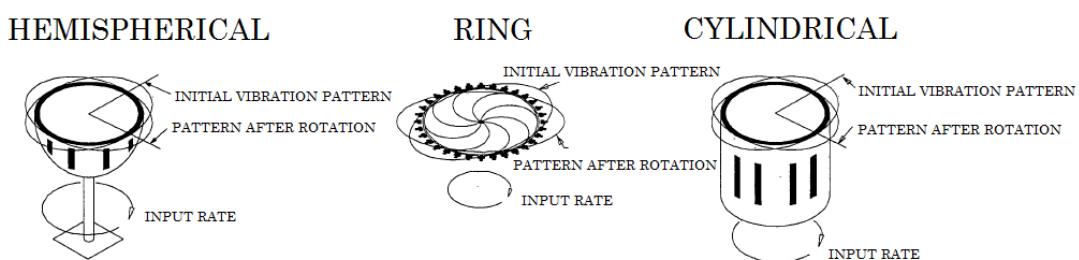


Figure 2.10: Vibrating Beam and Vibrating Shell Coriolis Gyroscopes [95].

spins with the same angular velocity applied to the sensor. An array of capacitors forming a comb is attached to the central part of the tuning fork in order to measure the displacement of the arms. Like in the case of the capacitive accelerometers, the variation of the capacity is used to measure the displacement and, consequently, the angular velocity.

2.1.3 Magnetometers

Magnetometers can be divided in two groups; those measuring the magnitude of the field and those that are able to measure each component of the magnetic field. Most magnetometers use the relationship between movement, magnetic fields and induced currents. A very complete survey of different kinds of magnetometers can be found in [96]. We reproduce here the section in which they describe MEMS magnetometers as this is the kind of magnetometers we will work with.

2.1.3.1 MEMS Based Magnetometers

According to [96], "[...] with the development of microelectromechanical systems (MEMS), the idea of using movement to sense magnetic fields is being reexamined. Fabricating these devices has turned out to be quite complicated. This is especially true if the fabrication process requires using different technologies that are not naturally compatible. For example, the use of HF (Hydrogen Fluoride), often required to perform the release step needed to fabricate the MEMS structure, can damage other parts of the sensor. The majority of these sensors use the Lorentz force. An example of this is a magnetometer based on detecting the motion of a miniature bar magnet. The hard magnetic material used was deposited by electrodeposition. The choice of materials for the hard magnet was limited by the need to use HF in the release step. The bar magnetic responds to the field without drawing any power.

MEMS technology can improve magnetic sensors by minimizing the effect of noise. The concept for a device that can accomplish this, the MEMS flux concentrator [97], [2], is shown in figure 2.11. In the device, the flux concentrators composed of soft magnetic material, are placed on MEMS flaps. The flux concentrators enhance the field. Decreasing the separation between the flaps increases the enhancement. The two MEMS flaps are forced to oscillate by applying an AC voltage to the electrostatic comb drives. By tuning the frequency, one can excite the normal mode in which the distance between the flaps oscillates. The resonant frequency for the MEMS structure is designed to be about 10 kHz. The oscillation of the MEMS flaps modulates the field at the position of the sensor and, hence, shifts the operating frequency of the sensor above the frequency where noise dominates. Depending on the type of magnetic sensor used, this shift in operating frequency should increase the sensitivity of magnetometers by one to three orders of magnitude. [...]"

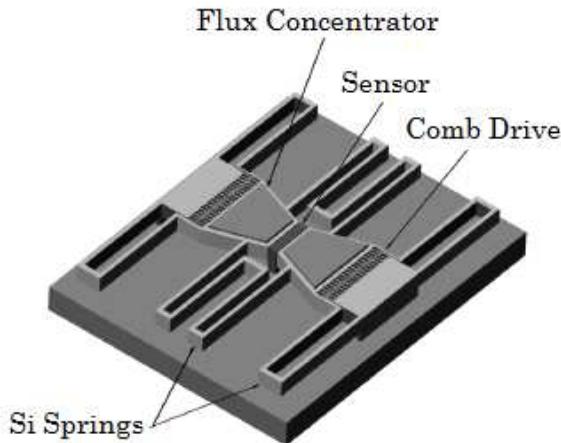


Figure 2.11: Concept of the MEMS flux concentrator [2].

2.2 Hardware

2.2.1 Prototypes developed for this project

In spite of the existence of many different commercial IMUs nowadays in the market, it is very important to emphasize that there were no complete wireless devices which could be used to implement a Wireless Sensor Network by the time we started this project. Therefore, we decided to develop a new line of prototypes to adequate the design to our needs and have more control over the different features and functionalities.

In the following, we will introduce two different versions of the prototype we developed during the early stages of our work: Wagyro and Wagyromag.

2.2.1.1 WAGYRO

In the first phase of the project we designed a simple IMU called WAGYRO (Wireless Accelerometer and GYROscope). WAGYRO includes an Analog Devices ADXL330 triaxial accelerometer [98] with a range of ± 3 g and an InvenSense IDG-300 biaxial gyroscope [99] with a theoretical range of ± 500 deg/s. Data are sampled at a rate of 50 Hz and sent using Digi's XBee®[100] module based on IEEE 802.15.4 [101]. Analog signals are converted with a 10 bit ADC, so raw data will be bounded to a $[0,1023]$ range. It also includes a 1100 mAh rechargeable battery. Figure 2.12 shows the WAGYRO's block diagram.

WAGYRO has a few limitations since the gyroscope is lacking from the Z sensitive axis and it does not include a magnetometer. It only allows online data gathering, so a computer has to act as a receiver in every moment. This fact limits the space where the subject carrying the IMU can be monitored as he can not leave the transmission range. Moreover, WAGYRO can only send raw data, that is, it does not process data locally in any means.

To overcome these limitations, we decided to design a newer version of WAGYRO including more features that had revealed to be important.

It is important to remark that WAGYRO has been used by Telefónica I+D in a knee telerehabilitation program which is taking place in various hospitals in Spain.

Figure 2.13 shows both the internal and external appearance of WAGYRO. Its dimensions are 40 mm×80 mm×20 mm and it has a weight of 58 g.

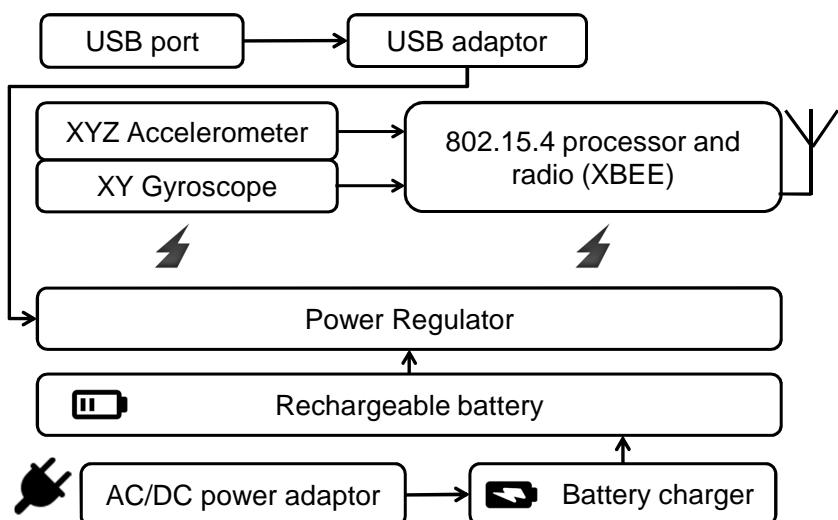


Figure 2.12: WAGYRO's block diagram.



Figure 2.13: Internal (left) and external (right) appearance of WAGYRO.

2.2.1.2 WAGYROMAG

WAGYROMAG (Wireless Accelerometer, Gyroscope and MAGnetometer) [102] is the second version of WAGYRO. It improves its predecessor by adding new sensors and features. More specifically, now we have in addition a triaxial gyroscope, a triaxial magnetometer, a temperature sensor and the data can be stored into a microSD card or be transmitted in real time again by the XBee chip. There is also a Microchip's® PIC24FJ64GB004 microcontroller [103] that provides local process capability, a buzzer, and a 1 Mb serial F-RAM memory. It operates at 3.3 V and uses a battery, which can be charged via USB or an external 5 V power supply.

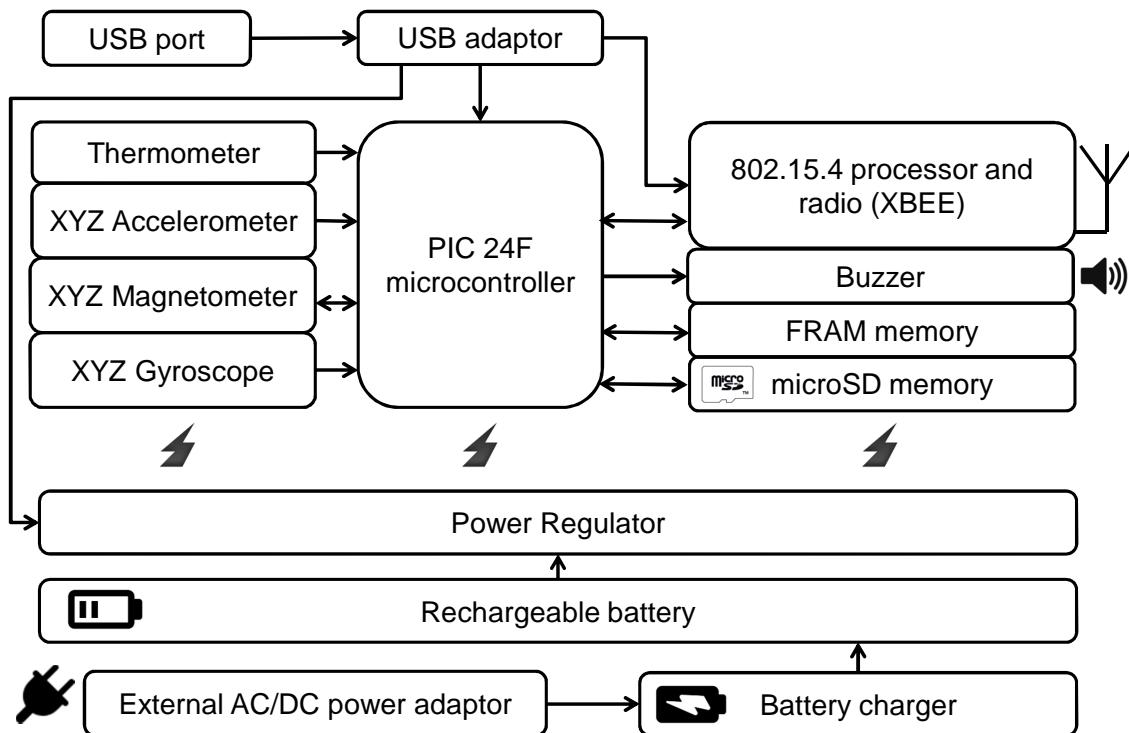


Figure 2.14: Wagyromag's block diagram.

Figure 2.14 shows the general com diagram of WAGYROMAG. Its components are listed below:

- *PIC24FJ64GB004*: The *PIC24FJ64GB004* is a 44 Pin, 16 Bit, Flash Microcontroller with USB On-The-Go (OTG) and nanoWatt XLP Technology.

The *PIC24FJ64GB004* Features are:

- 64 kBytes of Program Memory.
- 8 kByte SRAM.
- 5 Timers Modules (16 Bit).
- 2 UART w/IrDA Interfaces.
- 2 SPI Interfaces.
- 2 I2C Interfaces.
- 10 Bit A/D converter, 13 Channels.
- 1 RTCC Module.
- 1 USB OTG (On-The-Go).

The CPU includes a modified Harvard Architecture with up to 16 MIPS @ 32 MHz, an 8 MHz internal oscillator - PLL up to 96 MHz with multiple divide options, and an C Compiler optimized instruction set architecture with only 76 base instructions and flexible addressing modes. Additional features are an 17-times-17 Bit Single-Cycle Hardware fractional/integer multiplier an 32-by-16 Bit hardware divider and a 16-times-16 Bit working register array. Different power management modes are supported, for example, the Deep-Sleep-Mode allows to nearly shut down the Microcontroller whereas it still has the ability to wake up on external Trigger, WDT or RTTC alarm. The regular Sleep-Mode shuts down the peripherals and the core for power saving but offers a fast wake-up.

The operating Voltage can vary from 2.0 V to 3.6 V, but the digital input pins are all 5 V tolerant. Microchip warrants a minimum of 10.000 erase/write cycle endurance and 20-years of data retention. In-Circuit Serial Programming (ICSP), In-Circuit Debugging (ICD) via 2 PINS and JTAG Boundary Scan are supported.

- **ADXL335:** The ADXL335 [104] is a small, low-power, 3-axis ± 3 g accelerometer. Its operating voltage can vary from 1.8 V to 3.6 V consuming 350 μ A (typical). It is 10.000 g shock resistant and temperature stable ($\pm 0.01\text{ }^{\circ}\text{C}$). The ADXL335 emits an analog voltage for each axis which goes from 0 V to 3 V representing ± 3 g.
- **LPR550AL:** The LPR550AL [105] is a MEMS gyroscope with dual axis pitch and roll functionality produced by ST Microelectronics. Basically it is a combination out of one actuator and one accelerometer for each axis. The operating voltage can vary from 2.7 V to 3.6 V consuming 6.8 mA. It is used to measure angular velocity in x-axis and y-axis within a range of $\pm 2000\text{ }^{\circ}/\text{s}$.
- **LY550ALH:** The LY550ALH [106] is a high performance low-power single-axis micromachined gyroscope able to measure angular rate along yaw axis. The LY550ALH is capable of detecting rates with a -3 dB bandwidth up to 140 Hz. It is used to measure angular velocity in z-axis within a range of $\pm 2000\text{ }^{\circ}/\text{s}$.
- **FM24V10:** The FM24V10 [107] is a 1 Mb Serial 3 V F-RAM Memory IC with a fast two-wire serial interface (I2C). The maximum bus frequency is 3.4 MHz.

The operating voltage can vary from 2.0 V to 3.6 V and the current depends on the bus clock frequency. $175 \mu\text{A}$ @ SCL = 100 kHz, $400 \mu\text{A}$ @ SCL = 1 MHz, $1000 \mu\text{A}$ @ SCL = 3.4 MHz.

- *HMC5843*: The HMC5843 [108] is a 3-axis digital compass IC produced by Honeywell. It includes an I2C module for communication with a Microcontroller, which supports standard (100 kHz bus frequency) and fast mode (400 kHz bus frequency). The operating voltage can vary from 2.5 V to 3.3 V. In measurement mode the device consumes 0.9 mA and in sleep mode (single supply) $110 \mu\text{A}$. The measurement range is ± 4 Gauss.
- *MCP9700A-TT*: The MCP9700A-TT [109] is a tiny analog temperature sensor with a temperature range from -40°C to $+125^\circ\text{C}$. The accuracy is stated with a maximum of $\pm 2^\circ\text{C}$ (0°C to $+70^\circ\text{C}$). This IC is optimized for analog-to-digital converters with an output of $10\text{mV}^\circ\text{C}$. Its operating voltage can vary from 2.3V to 5.5V consuming $6 \mu\text{A}$ (typical).
- *Micro-SD card*: A general micro-SD card is used for data storage. Any micro-SD card up to 4GB can be employed.
- *XBee*: XBee [100] is a 802.15.4 based radio module from Digi International that includes a microcontroller with 6 analog inputs which also supports serial communication. The XBee chip also includes a UART interface which can be used to send messages and commands between the XBee Modules. It admits a transmission rate up to 250 kbps and uses the 2.4 GHz ISM band. It is able to achieve a transmission range up to 30 meters indoors and 90 meters outdoors at 1 mW.
- *Rechargeable Battery*: Wagyromag includes a 1100 mAh rechargeable battery which provides an autonomy of 12-15 hours.

Its dimensions are identical to Wagyrō's (40 mm \times 80 mm \times 20 mm) and it has a weight of 50 g.

WAGYROMAG's microcontroller is programmed with a firmware that reads data from sensors and stores them in the microSD card or sends them wirelessly. This firmware will be briefly described in section 2.3.

Data can be sampled at different rates by modifying the firmware. In the following, all data that is analyzed and shown, have been sampled at a frequency of 40 sps.

Figure 2.15 shows the internal and external appearance of WAGYROMAG.



Figure 2.15: External and internal appearance of WAGYROMAG.

2.2.2 Performance of Sensors

Prior to the calibration procedure of the sensors (later explained in detail in chapter 3), it is advisable to carry out a performance analysis of the sensors included in the IMU that will be used to monitor motion. The performance of the sensors can be analyzed by studying the noise which is present at their output. MEMS MARG sensors present different kinds of noise processes that affect the measurements to a greater or lesser extent.

A complete analysis of MEMS error characteristics can be found in [110], which is based, in turn, on the *IEEE 952-1997 Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros* [95]. According to it, the main MEMS error characteristics are: constant bias, thermo-mechanical white noise/angle random walk, flicker noise/bias instability, temperature effects and calibration errors.

2.2.2.1 Analysis of Allan Variance

Before describing the error characteristics we should introduce the technique that is used to compute their numerical value. The Allan Variance is a time domain analysis technique that was originally developed by Dr. David W. Allan [111] to

study the frequency stability of oscillators. It can be used to determine the character of the underlying random processes that give rise to the data noise in any kind of instrument. The mathematical definition, extracted from [95] is as follows.

Consider N samples of data with a sample time of τ_0 . Then, form data clusters of lengths $\tau_0, 2\tau_0, \dots, k\tau_0$ ($k < N/2$) and obtain averages of the sum of the data points contained in each cluster over the length of that cluster. The Allan variance is defined as a function of cluster time. Since we will be analyzing the effects of integrating the noise present in the output signals, the Allan Variance can also be expressed in terms of integrated output $\theta(t)$,

$$\theta(t) = \int^t \Omega(t') dt' \quad (2.12)$$

where $\Omega(t)$ is the output of the sensor (rate or acceleration). The average rate between times t_k and $t_k + t$ is given by

$$\bar{\Omega}_k(\tau) = \frac{\theta_{k+m} - \theta_k}{\tau} \quad (2.13)$$

where $\tau = m\tau_0$. Then the Allan Variance is defined as

$$\sigma^2(\tau) = \frac{1}{2} \left\langle (\bar{\Omega}_{k+m} - \bar{\Omega}_k)^2 \right\rangle = \frac{1}{2\tau^2} \left\langle (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \right\rangle \quad (2.14)$$

where $\langle \rangle$ is the ensemble average. The final expression to estimate the Allan Variance is, then,

$$\sigma^2(\tau) = \frac{1}{2\tau^2(N-2m)} \sum_{k=1}^{N-2m} (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \quad (2.15)$$

The Allan Variance is also related to the two-sided PSD of the noise by applying

$$\sigma^2(\tau) = 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df \quad (2.16)$$

A straightforward interpretation of the previous equation is that the Allan Variance is proportional to the total noise power of the output when passed through a filter that has a transfer function of $\frac{\sin^4(x)}{x^2}$. This transfer function comes from the form in which we divide the data in clusters and operate with them. As we see in equation 2.16 the bandpass of the filter depends on the value of τ . Therefore, by inspecting the value of the Allan Variance at different values of τ we will be able to quantify different noise terms. This analysis is easily done by plotting the square root of the Allan variance versus τ on a log-log plot.

Now that we know the expression to quantify the noise terms we can proceed to briefly explain them.

2.2.2.2 Constant Bias

The constant bias of MARG sensors is the offset of its output signal from the actual value. It can be computed by averaging in long term the output of the sensors while they are in a static position. We should take into consideration that the accelerometer will measure the acceleration of the gravity while being static and that the magnetometer may measure other magnetic fields apart from the Earth's one if the IMU is in a magnetically perturbed environment. Ideally the bias should be constant from power-up to power-down, but this is not the case as the bias suffers from dynamic drift which forces us to periodically recompute its value.

2.2.2.3 Thermo-Mechanical White Noise & Angle Random Walk

Since the sensors have mechanical parts in the of order micrometers and nanometers, their output will be distorted by thermo-mechanical noise with a frequency many times higher than the sampling rate of the sensor. This noise can be modeled as a white sequence, that is, the noise has zero-mean and is composed by a set of random uncorrelated variables with constant finite variance σ^2 .

When using accelerometers and gyroscopes to monitor motion and compute positioning angles, we will need to carry out integrations over the acceleration and angular rate signals as it is explained in chapter 5. The presence of thermo-mechanical white noise causes an effect known as the Angle Random Walk (ARW) [95, 112] in the case of the gyroscope and Velocity Random Walk (VRW) in the case of the accelerometer. The ARW and VRW specify the average deviation that occurs when the inertial signals are integrated. It is usually included as a noise specification with units of deg/ \sqrt{h} and m/s $^2/\sqrt{h}$ respectively. To study their effect we have to compute the expected value and variance of the integrated white noise signal.

Let define N_i as the i^{th} random variable of the white noise sequence. Assuming that each N_i is identically distributed with zero mean and finite variance σ^2 , the result of integrating the white noise signal using the rectangular rule $\epsilon(t)$ over a time period $t = n\delta t$ is

$$\int_0^t \epsilon(\tau)d\tau = \delta t \sum_{i=1}^n N_i \quad (2.17)$$

where n is the number of samples in the time period and $\delta t = 1/f$, where f is the sampling frequency and δt is the time between two consecutive samples. If we use the variance and expected value properties that reflect that $E(aX+bY) = aE(X)+bE(Y)$ and $Var(aX+bY) = a^2Var(X) + b^2Var(Y) + 2abCOV(X,Y)$, and the fact that $E[X]=0$ and

$\text{COV}(X, Y) = 0$ for a white sequence, then

$$\begin{aligned} E\left(\int_0^t \epsilon(\tau) d\tau\right) &= E\left(\delta t \sum_{i=1}^n N_i\right) = E\left(\delta t \left[\sum_{i=1}^{n/2} N_i + \sum_{i=n/2+1}^n N_i\right]\right) = \\ &= \delta t E\left(\sum_{i=1}^{n/2} N_i\right) + \delta t E\left(\sum_{i=n/2+1}^n N_i\right) = \delta t E\left(\frac{n}{2}N\right) + \delta t E\left(\frac{n}{2}N\right) = \\ &= \delta t \frac{n}{2} E(N) + \delta t \frac{n}{2} E(N) = \delta t n E(N) = 0 \end{aligned} \quad (2.18)$$

$$\begin{aligned} \text{Var}\left(\int_0^t \epsilon(\tau) d\tau\right) &= \text{Var}\left(\delta t \sum_{i=1}^n N_i\right) = \text{Var}\left(\delta t \left[\sum_{i=1}^{n/2} N_i + \sum_{i=n/2+1}^n N_i\right]\right) = \\ &= \delta t^2 \text{Var}\left(\sum_{i=1}^{n/2} N_i\right) + \delta t^2 \text{Var}\left(\sum_{i=n/2+1}^n N_i\right) + 2\delta t^2 \text{Cov}\left(\sum_{i=1}^{n/2} N_i, \sum_{i=n/2+1}^n N_i\right) = \\ &= \delta t^2 \left[\text{Var}\left(\sum_{i=1}^{n/2} N_i\right) + \text{Var}\left(\sum_{i=n/2+1}^n N_i\right) \right] = \delta t^2 \left[\text{Var}\left(\frac{n}{2}N\right) + \text{Var}\left(\frac{n}{2}N\right) \right] = \\ &= \delta t^2 \left[\frac{n}{2} \text{Var}(N) + \frac{n}{2} \text{Var}(N) \right] = \delta t^2 n \text{Var}(N) = \delta t t \sigma^2 \end{aligned} \quad (2.19)$$

Therefore, we have shown that integrating the noise introduces a zero-mean random walk error into the integrated signal.

The PSD of the thermo-mechanical noise can be represented by $S_\Omega(f) = N^2$ where N is the angle random walk coefficient that we want to compute. If we plug $S_\Omega(f)$ into equation 2.16 we will be able to obtain the value of the Allan Variance in terms of τ for the thermo-mechanical noise.

$$\sigma^2(\tau) = 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df = 4N^2 \int_0^\infty \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df = 4N^2 \left(\frac{1}{4\tau} \right) = N^2/\tau \quad (2.20)$$

Finally, by inspecting the log-log plot depicting the square root of the Allan Variation with respect to the time division τ we can obtain the value of N , that is, the ARW, by checking the value of the curve for $\tau = 1$.

2.2.2.4 Bias Instability

Bias instability, as its name suggests, is the variation present in bias when we measure it over a determined period of time while the sensor is in a static position and at a constant temperature. Bias instability is usually expressed as a 1σ value

whose units are deg/s or deg/h. This low-frequency noise is originated by the random flickering present in the electronics. Its associated PSD is

$$S_\Omega(f) = \begin{cases} \frac{B^2}{2\pi f} & f \leq f_0 \\ 0 & f > f_0 \end{cases} \quad (2.21)$$

where B is the bias instability coefficient that we aim to find and f_0 is the cutoff frequency. If we substitute the previous equation in equation 2.16, we obtain the Allan variance of the flicker noise

$$\begin{aligned} \sigma^2(\tau) &= 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df = 4 \int_0^\infty \frac{B^2}{2\pi f} \frac{1}{(\pi f \tau)^2} \sin^4(\pi f \tau) df = \\ &= \frac{2B^2}{\pi} \left[\ln 2 - \frac{\sin^3(x)}{2x^2} (\sin(x) + 4x \cos(x) + C_i(2x) - C_i(4x)) \right] \end{aligned} \quad (2.22)$$

where x is $\pi f_0 \tau$ and C_i is the cosine-integral function. If we plot the Allan Variance obtained for bias instability we observe that there is a flat region for τ much longer than the inverse cut off frequency. Therefore, this flat area is used to determine the limit of the bias instability.

2.2.2.5 Temperature Effects

The bias can also be affected by temperature changes in the environment and those due to the sensor self heating. When using MEMS inertial sensors we may need to wait until the sensor has reached a stable temperature. This may take from seconds up to some minutes from power up. Most commercial IMUs include a built-in temperature sensor that can be used to compensate for temperature effects by computing the calibration parameters at different temperatures. Then, a function determining the value of the parameters depending on the temperature can be computed and stored at the IMU to locally compensate for temperature errors. As stated in section 2.2.1.2, our prototype includes such a temperature sensor.

2.2.2.6 Calibration Errors

The performance of the sensor is also affected by errors in calibration parameters. When defining the mathematical function modeling the output of the sensor we include several parameters such as the scale factors, the cross-axis sensitivity, the non-orthogonality between sensor triplets and different biases among others. Chapter 3 is dedicated to study in depth the definition, determination and correction of the errors in calibration parameters.

2.2.2.7 Analyzing WAGYROMAG's inertial sensors

Once we have described the main errors, and their sources, present in the output of MEMS inertial sensors we proceed to analyze the performance of WAGYROMAG's inertial sensors.

To that purpose we have gathered acceleration and angular rate data of both the triaxial accelerometer and triaxial gyroscope during five and a half hours while the IMU was static. After extracting the gathered data from the SD card we computed the Allan Variance of each one of the sensors signals. Then, the Bias Instability value is computed by measuring the minimum value of the Allan Variance and the ARW is obtained by fitting a line through the negative slope and reading its value at $t = 1$. These values are depicted in table 2.1 and figures 2.16-2.19 show the Allan Variance curves and the points used to extract the ARW and Bias Instability values for both the accelerometer and the gyroscope.

Table 2.1: Wagyromag's Bias Instability and Angle Random walk computed using Allan Variance.

Sensor	Bias Instability	ARW
Accel. (X axis)	$0.0031 \text{ m/s}^2 = 11.1677 \text{ m/h}^2$	$0.0160 \text{ m/s}^2/\sqrt{s} = 0.9648 \text{ m/s}^2/\sqrt{h}$
Accel. (Y axis)	$0.0018 \text{ m/s}^2 = 6.5592 \text{ m/h}^2$	$0.0073 \text{ m/s}^2/\sqrt{s} = 0.4418 \text{ m/s}^2/\sqrt{h}$
Accel. (Z axis)	$0.0025 \text{ m/s}^2 = 9.0108 \text{ m/h}^2$	$0.0142 \text{ m/s}^2/\sqrt{s} = 0.8526 \text{ m/s}^2/\sqrt{h}$
Gyro. (X axis)	$0.1080 \text{ }^\circ/\text{s} = 388.8000 \text{ }^\circ/\text{h}$	$0.3389 \text{ }^\circ/\sqrt{s} = 20.3340 \text{ }^\circ/\sqrt{h}$
Gyro. (Y axis)	$0.1040 \text{ }^\circ/\text{s} = 374.4000 \text{ }^\circ/\text{h}$	$0.3108 \text{ }^\circ/\sqrt{s} = 18.6400 \text{ }^\circ/\sqrt{h}$
Gyro. (Z axis)	$0.0280 \text{ }^\circ/\text{s} = 100.8000 \text{ }^\circ/\text{h}$	$0.2342 \text{ }^\circ/\sqrt{s} = 14.0500 \text{ }^\circ/\sqrt{h}$

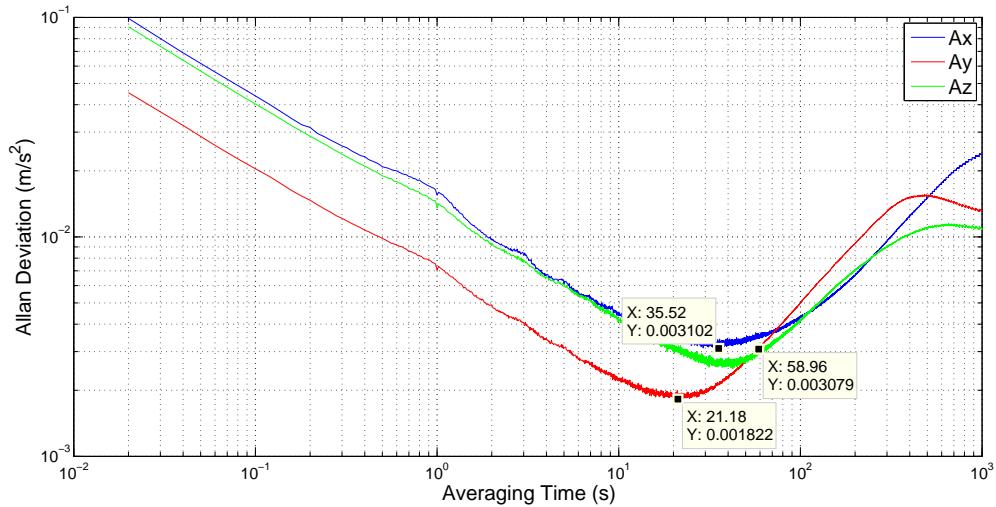


Figure 2.16: Computation of Bias Instability by means of the Allan Deviation (Triaxial Accelerometer).

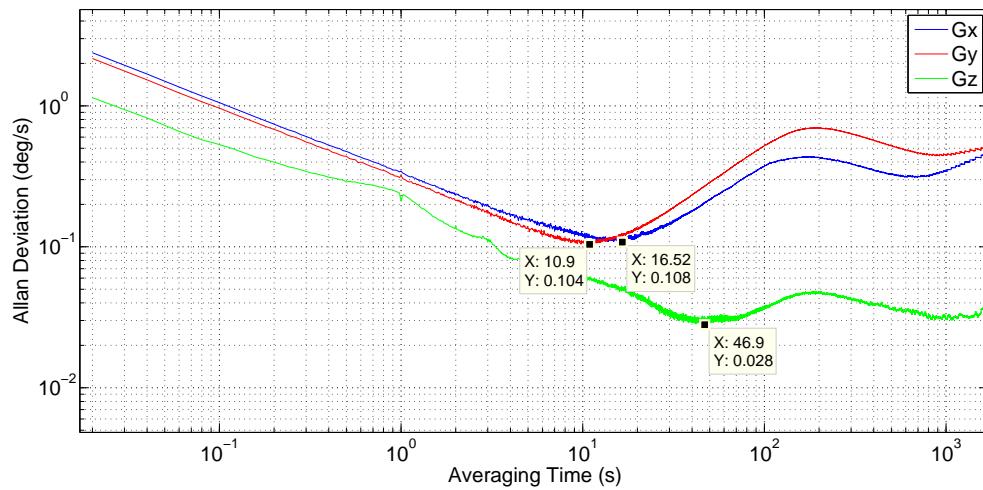


Figure 2.17: Computation of Bias Instability by means of the Allan Deviation (Triaxial Gyroscope).

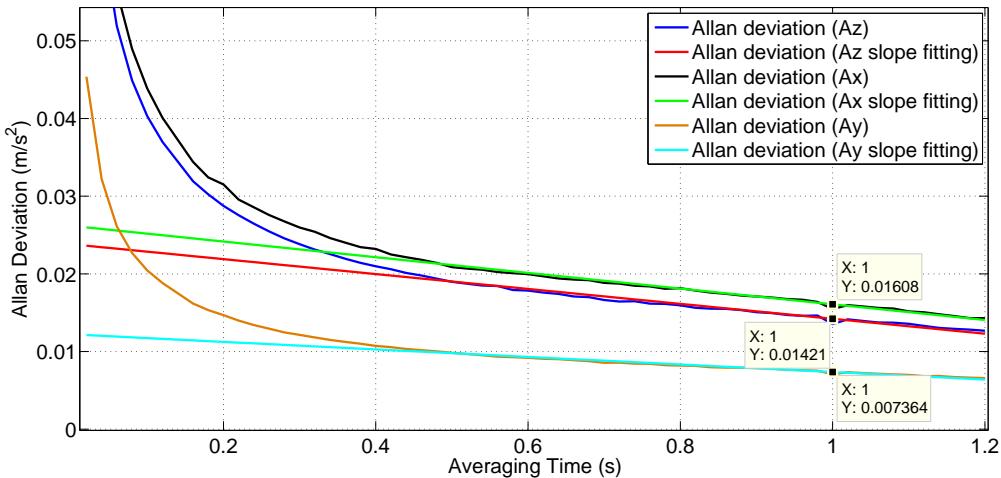


Figure 2.18: Computation of Angle Random Walk by means of the Allan Deviation (Triaxial Accelerometer).

Results depicted in the table and the figures show that, even though they are slightly larger than those of commercial devices [110], they are acceptable for position monitoring applications as long as the ARW and bias drift are compensated by means of in-use calibration or adaptive filtering sensor fusion approaches.

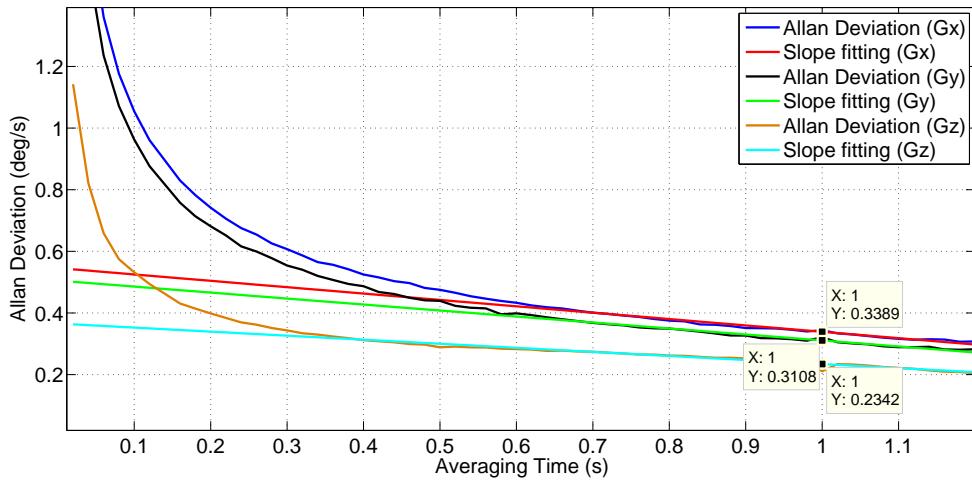


Figure 2.19: Computation of Angle Random Walk by means of the Allan Deviation (Triaxial Gyroscope).

2.3 Firmware

As it was explained in section 2.2.1.2, we have implemented the firmware that is included in WAGYROMAG's microcontroller. This firmware is in charge of reading data from the sensors, store them in the microSD card or pack them and send them via the XBEE transceiver. This section aims to explain the firmware's workflow in a general way without going into code details.

When the device is switched on, the firmware reads the internal memory to check if the calibration parameters are stored. If not, it waits until it receives them wirelessly. Then it checks if the SD card is inserted in the slot; if so, it gathers the data from the sensors and stores them in it. If not inserted, the data is sent wirelessly to be processed in a remote station. An external switch allows to switch between local storage mode and wireless mode at any moment of execution.

2.3.1 Diagram of states

In order to ease the understanding of the firmware, its workflow is depicted in detail in figure 2.20. As we can see, the device is configured within the first two states. Then, the calibration parameters are checked and, if not already stored, the device waits until they are received and stores them. The calibration parameters are used to transform raw data into data expressed in physical units and to correct for non-orthogonalities, axis misalignments, cross-axis sensitivity and other undesired

effects which are later explained in depth in chapter 3. Once the main routine is done with the calibration parameters it searches for a microSD card where the data read from the sensors will be stored. The existence of a memory card determines the operation mode of the device (data storage or data transmission).

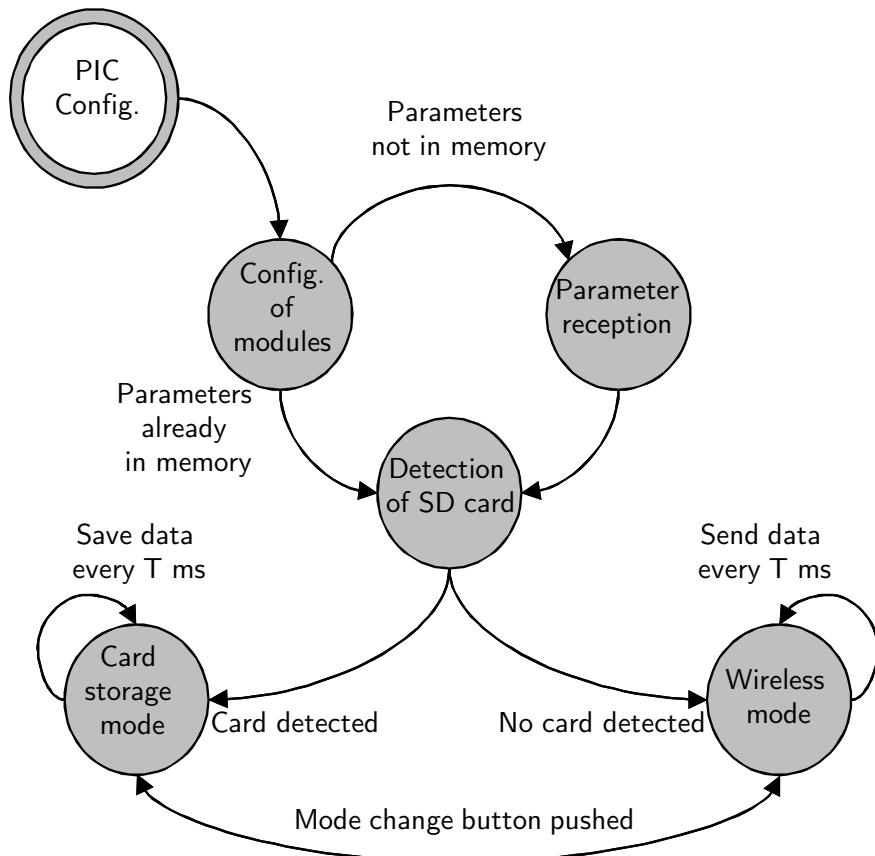


Figure 2.20: Diagram of states of WAGYROMAG's firmware. Modified from [3].

2.3.2 Diagram of activity

Now, if we take a look to figure 2.21 we can see the structure of the main routines of the program emphasizing their dependency. The diagram is divided into three columns each one representing the three main routines; general configuration of the device; data processing; data transmission/storage. Each level represents the dependency of the routines, that is, those routines depicted in lower levels can not start until the ones above have finished their tasks.

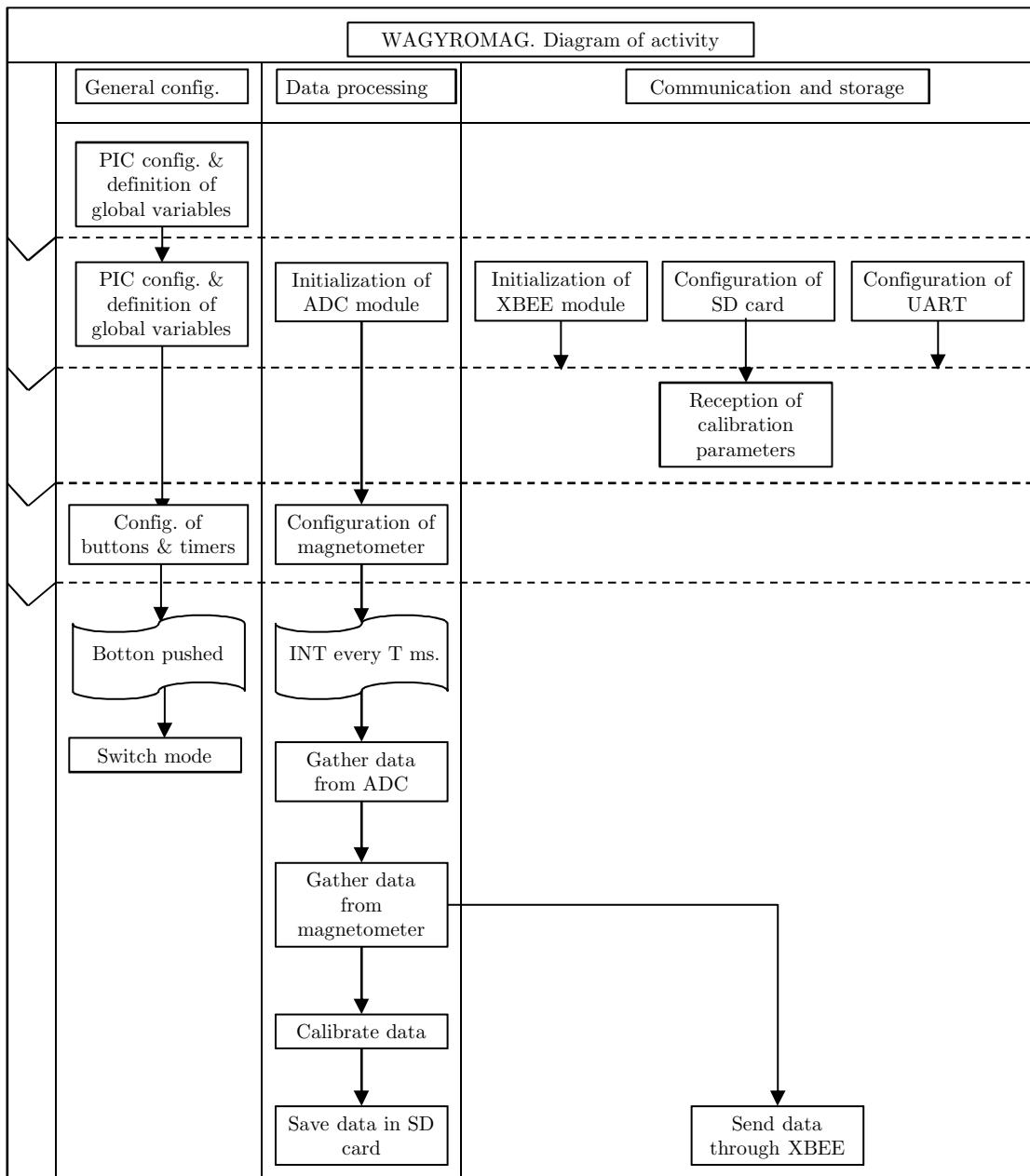


Figure 2.21: Diagram of activity of WAGYROMAG's firmware. Modified from [3].

2.3.3 Main routine

Once the timers, inputs, outputs, sensors and the rest of the MIMU devices are configured, if in the data storage mode, the firmware proceeds to create the .CSV

file where the data coming from the sensors will be stored in columns. On the other hand, if in the data transmission mode, the wireless transmission flag is activated and the execution continues. The next step consists on checking the correct functioning of the ferromagnetic memory by writing a known datum in a known position which is read right afterwards to see if it matches the written value. If not matching, a continuous buzzing indicates the user that there is an error in the memory. Then, the parameter reception routine, explained in 2.3.4.3, is called and once the calibration parameters are received, or found in memory, the buzzer indicates that all the initialization routines are done, followed by the activation of timer that controls the data-gathering interruption and the execution of the main loop. A counter is incremented every time that the data-gathering interruption is executed, and, when it reaches the value of the size of the data buffer, the data are calibrated and stored in the memory card.

2.3.4 Auxiliary routines

The main loop calls a series of auxiliary routines that are in charge of different tasks such as sounding the buzzer, detecting the memory card, receiving the calibration parameters and son on. These auxiliary routines are briefly described within the next paragraphs below.

2.3.4.1 Sounding the alarm

The buzzer is used as an indicator of the status of the execution. A short buzzing is used to notify the user that the program is running as expected and a continuous buzzing indicates that something went wrong. The buzzer uses a signal which oscillates at a frequency close to the sound that we want to generate. Such signal is produced by using the internal timers of the microcontroller.

2.3.4.2 Detecting SD card and creating data file

The microSD card is managed using a set of libraries provided my Microchip that include all the necessary routines to create, write and read files. Every time the device is powered up a new file, named *WAGYROX.CSV*, where X is the number from 0 to 999, is created. Using a method that allows to search files in the root folder, the routine seeks files starting from *WAGYRO0.CSV* and ending when the method returns a negative result; in that point it creates a new file.

The last task of the routine consists of writing the header of the file, that is, the first row of the .CSV that indicates the name of the variable that will be stored in each column. The format of the header is shown in table 2.2 where A, G and H

represent the acceleration, angular rate and magnetic field respectively, and x , y and z represent the cartesian coordinate axes.

2.3.4.3 Receiving calibration parameters

The calibration parameters are received just before entering the main loop. In the first place, the routine reads the 0x01 position of the memory, which is defined as a flag that indicates if the parameters are stored. If 0xA is read, the parameters are already stored and the routine ends its execution. Alternatively, if any other different value is read, the routine puts the execution in an idle mode waiting for the reception of the calibration parameters which are sent from a base station. After their reception, the auxiliary routine ends and returns control to the main one.

2.3.4.4 Reading data from the Analog/Digital Converter

Data coming from the accelerometer and the gyroscope are read using the microcontroller's built-in ADC. The ADC is read every time the interruption corresponding to the main timer is triggered. The method that is in charge of reading the data from the sensors packs them into a structure in which all the current ADC reads are stored together with the magnetometer data.

2.3.4.5 Reading data from the magnetometer

The magnetometer does not provide an analog signal corresponding to the measured magnetic field since it has its own ADC and outputs the measurements using an I^2C bus. The main timer interruption sends the address of the magnetometer through the I^2C bus followed by the read instruction and, then, it waits for magnetometer to send the corresponding data.

2.3.4.6 Calibrating data

The data is calibrated before storing them in the memory card by applying the calibration parameters on the buffered data. This procedure is repeated every second, that is, every time a group of $1/T$ of samples is read, where T is the sampling

Time	Ax	Ay	Az	Gx	Gy	Gz	Hx	Hy	Hz	Temp	AnBat	ExtAngle
------	----	----	----	----	----	----	----	----	----	------	-------	----------

Table 2.2: Header format of the data file stored in the microSD card.

period. The whole process followed to compute the calibration parameters is explained in detail in chapter 3.

2.3.4.7 Writing data in the microSD card

After calibrating the data, they are stored in the memory card. An specific method receives as input parameters the name of the file, the write/read mode and the data array pointer. The method calls Microchip's *FSfopen* method to open the file in write mode.

Once the file is open, a loop goes through the data array and, using an auxiliary method, converts every data structure into a string array that is written in comma separated values following the format previously shown in table 2.2. Once the data array has been completely scanned, the file is closed and a flag is returned to indicate that the data has been successfully written.

2.3.4.8 Sending data wirelessly

If the device is working in the wireless data transmission mode, the data are wirelessly sent through the XBee transceiver to the base station. Prior to their transmission, the data are encapsulated into the frame specified in table 2.2 and the CRC of the XBee package is computed. Finally, the data are sent byte by byte.

2.3.5 Interruption routines

There are two interruptions that make the program jump out of the main routine; the one generated by timer 1 and the one generated by pushing the mode button.

2.3.5.1 Data gathering interruption

This routine is called every time that timer 1 is overflowed, that is, every T ms (T is set to 25 ms in the default configuration). This timer controls how often the sensors are read, that is, the sampling frequency. The routine calls, in the first place, the function that captures data from the ADC. Then, it calls the function that reads the magnetometer data and stores them together with the accelerometer, gyroscope, thermometer and battery level data in the data frame. Then the operation mode (data storage or data transmission) is checked and it calls the corresponding function to store or send the data. Finally the data buffer indicator is incremented and the control is returned to the main routine.

2.3.5.2 Mode switching interruption

This routine is called whenever the mode button is pushed. It first checks the value of a flag that indicates the operation mode. If in storage mode it automatically switches to the transmission mode by following the next steps:

- (i) Stop timer 1 so the data gathering interruption can not get the control of the execution.
- (ii) Change the value of the operation mode indicator.
- (iii) Generate a buzzing to indicate that the mode is being switched.
- (iv) Re-enable timer 1.

On the other hand, if in the transmission mode the routine proceeds as follows:

- (i) Stop timer 1 so the data gathering interruption can not get the control of the execution.
- (ii) Check if SD card is inserted in the slot. If not the buzzer indicates it playing a sound for a few seconds and execution continues in transmission mode.
- (iii) If the card is in its place the last existing file in memory is opened (or created if the memory is blank) and a new header is added.
- (iv) Change the value of the operation mode indicator.
- (v) Re-enable timer 1.

Once the firmware is implemented and programmed in the MIMUs, the next step consists of developing software to receive the data being sent by the MIMU or process and visualize the data stored in the memory card. This software will be executed in a computer that acts as a base station.

2.4 Data Visualization Software

When we started thinking about building an application to receive, analyze, visualize and store the data gathered by our MIMU, we opted to use *LabView* as it permitted fast development of applications to read, visualize and store data. The first application we built, named *WagyLog* [113] was intended to read data from the USB port to which the *XBee* dongle is connected, plot it in real time and store it in a

file.

This first tool was very limited and required pre-installation of LabView so we decided to develop another application including a set of new features.

The subsequent tool, named *InertialCapture*, is able to read data from any MIMU, plot measurements in 2D, compute positioning angles with customizable algorithms, plot human body position in real time using a 3D avatar, store data and load and play stored data. This tool was built mainly in C# in conjunction with Mr. Luis Curiel, application developer and interfaces expert, from *Telefónica I+D*. The main idea was to build a very flexible tool in which it is very easy to add new supported MIMUs and data processing algorithms.

The main window of the application is divided in four sections; general configuration; 2D visualization; 3D visualization and avatar and sensors configuration.

Along the following sections we briefly describe their structure and features, as well as its functioning principles.

2.4.1 General configuration

The general configuration window, located in the upper left corner of the main window, allows us to choose the mode of the application (configuration, real time, record session and play session) as well as to take notes of every session and visualize the debug window where the status messages appear. From this panel we can play saved sessions, record new sessions and add them to the database or activate the real time mode to see real 2D plots and movement of the avatar.

2.4.2 Connection of sensors and avatar configuration

In order to let the application detect the MIMU that is gathering the data, we need to previously define a XML file that contains all the information about the sensors, the format of the data frame, the sampling rate, the baud rate and other parameters that fully characterize the MIMU. The header of the XML file indicates the name of the sensor, the vendor, the COM port it is connected to and the baud rate. An example of the header of the XML file corresponding to Wagyro is as follows,

```
<gtsensor name="wagyro" vendor="icr" comport="3" baudrate="57600">
```

By reading this header, the application knows that it should read the information coming from COM3 port at 56000 bps. The next part of the XML file indicates the type of each one of the data that will be sent by the IMU. If we want the application to read 2 bytes of binary data where the MSB is sent prior to the LSB we will write,

```
<tag type="value" byteoffset="2" datatype="12" datasize="2" name="gx" />
```

Where *byteoffset* indicates the number of additional bytes that are required to read the complete data, *datatype* uses a dictionary to know the format of the incoming data and *datasize* indicates the size in bytes corresponding to the measured variable indicated by *name*. When the data parser finds a complete frame it returns an structure that contains a vector with the received values measured by the sensors. In case we want to read ascii data with an unknown size we will also need to specify the delimiters as depicted in the following example,

```
<tag type="value" datatype="57" name="ax" />
<tag type="constraint" byteoffset="1" value="127" />
```

This chunk of code indicates the application it should read an integer ascii datum (57) named '*ax*' (Acceleration measured along the X axis). Since we do not know the size of the datum, the application stops reading it when it reaches the constraint, a 127 character in this case. In short, the binary parser has the following features,

- Decodes of ascii, binary, hexadecimal and other data formats.
- Supports data vectors.
- Supports conditionals through the option *tag* to apply different data processing algorithms based on conditioning values.
- Access data at bit level.
- Allows high level XML description.
- Loads the format in memory to maximize efficiency and memory consumption.
- Automatically detects the end of the frame.
- Automatically computes date type data.
- Supports different CRC algorithms.

Finally, as an example, we include the complete XML file needed by the program to properly read and interpret data coming from Wagyromag.

```

<gtsensor name="wagyro" vendor="ICR-UGR" comport="3" baudrate="57600">
  <parser>
    <tag type="constraint" byteoffset="1" datatype="1" value="255" />
    <tag type="value" byteoffset="5" datatype="5" name="" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="gx" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="gy" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="gz" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="accx" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="accy" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="accz" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="mx" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="my" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="mz" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="tx" />
    <tag type="value" byteoffset="2" datatype="12" datasize="2" name="battery" />
    <tag type="value" byteoffset="7" datatype="6" name="" />
  </parser>
</gtsensor>

```

The next step consists in configuring the position of the sensors on the 3D avatar. We need to tell the application what sensor is attached to what body joint. This is done by associating one of the precharged IMU XML files. We must also select the COM port, the sensor ID (in case that two or more sensors are paired with one receiver dongle) and the algorithm that will be employed to compute the positioning angles. The body of the avatar has been divided in 10 parts including the most important body joints. Figure 2.22 shows such division.

2.4.3 2D Visualization

The left lower window shows the 2D real time plots of any of the sensors' signals. When configuring the body joints we can also specify which signals we want to visualize. The selected signals are depicted in the same plot and can be easily differentiated as they are painted with different colors. The visualizer has two working modes; continuous, where only the last 100 samples are displayed, and complete where the whole signal is depicted since the sensors start to be monitored.

The visualizer is able to zoom the signals in and out and save the graphs in different image formats. Figure 2.23 depicts the output of the triaxial accelerometer.

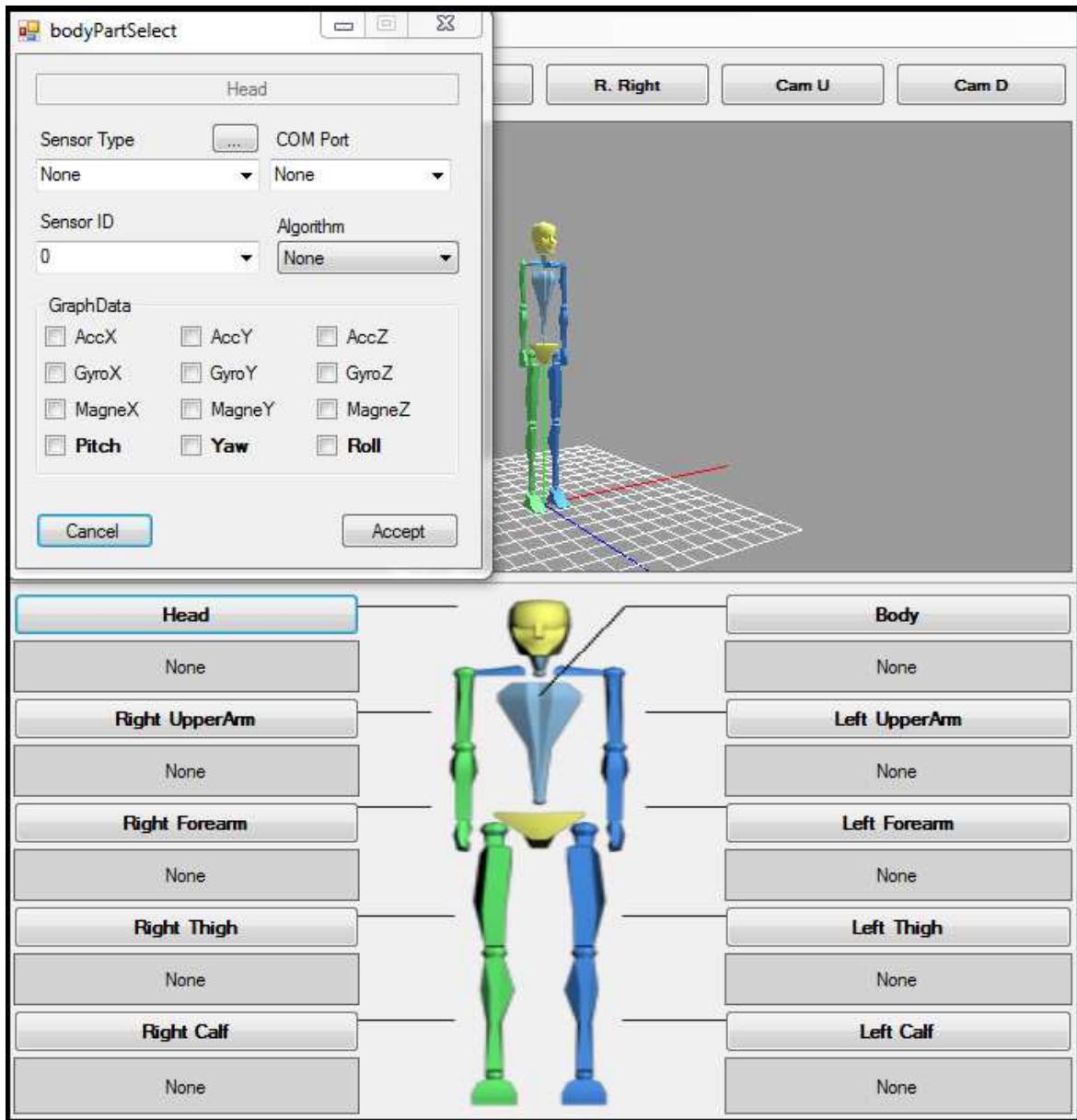


Figure 2.22: Attachment of sensor to an avatar's body part.

2.4.4 3D Visualization

As aforementioned, the application includes a 3D human body skeleton-like avatar which is located in the right upper corner. The avatar is divided in 10 parts; right and left arms, right and left forearms, right and left thighs, right and left calves, trunk and head. In order to visualize the movement of the avatar we need to select and algorithm that computes the Euler angles using the measured inertial signals. The movement of the avatar can be recorded from different angles using the cam-

era view controls. Then, the movement of the avatar can be subsequently played together with the 2D plots to ease the association of the sensor signals with body movement. A capture of the moving 3D avatar can be seen in figure 2.24.

2.4.5 Recording sessions

To record and play the monitoring sessions we have also used a XML file that includes all the necessary variables that specify the Euler angles, the inertial signals and the information about the sensors and where they are placed. The file format is very intuitive and can be easily read and understood by the operator. Through the *File* menu we can also generate a simpler CSV file also including all inertial signals and positioning angles.

2.4.6 Playing sessions

Through the *File* menu the user can import any of the sessions that were previously recorded. This allows to analyze the movement of the avatar as well as any of the gathered signals whenever it is required. Each session can include information about the person being monitored such as his name, age, height and weight, etc. the place and date of the monitoring session and any other aspect considered as relevant by the operator controlling the session. The ability to record and play sessions

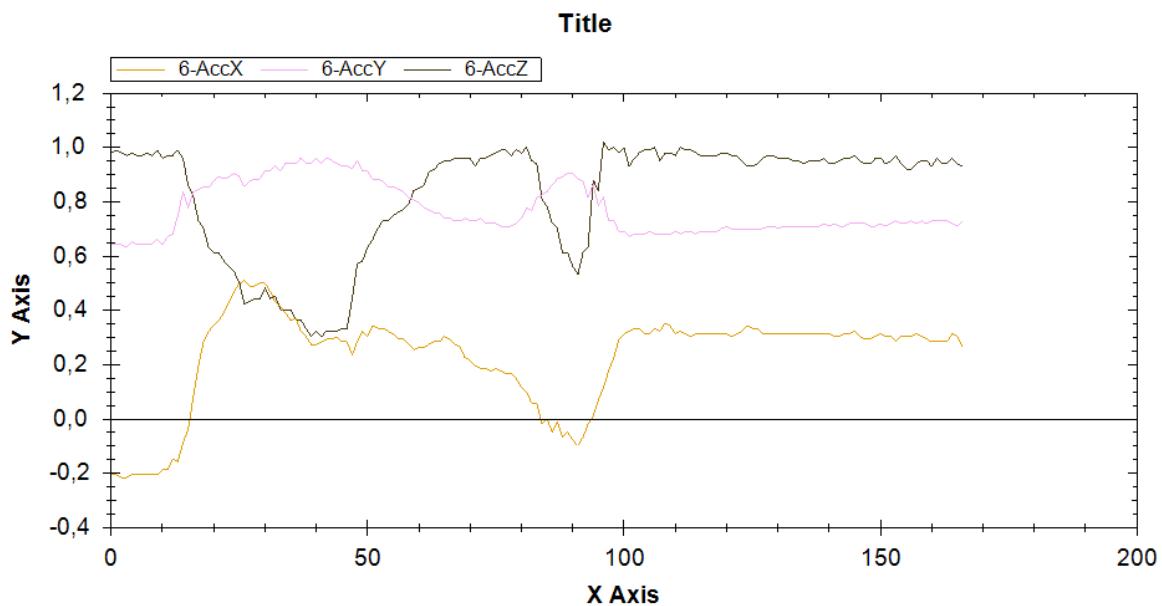


Figure 2.23: Example of an acceleration 2D Plot generated by InertialCapture.

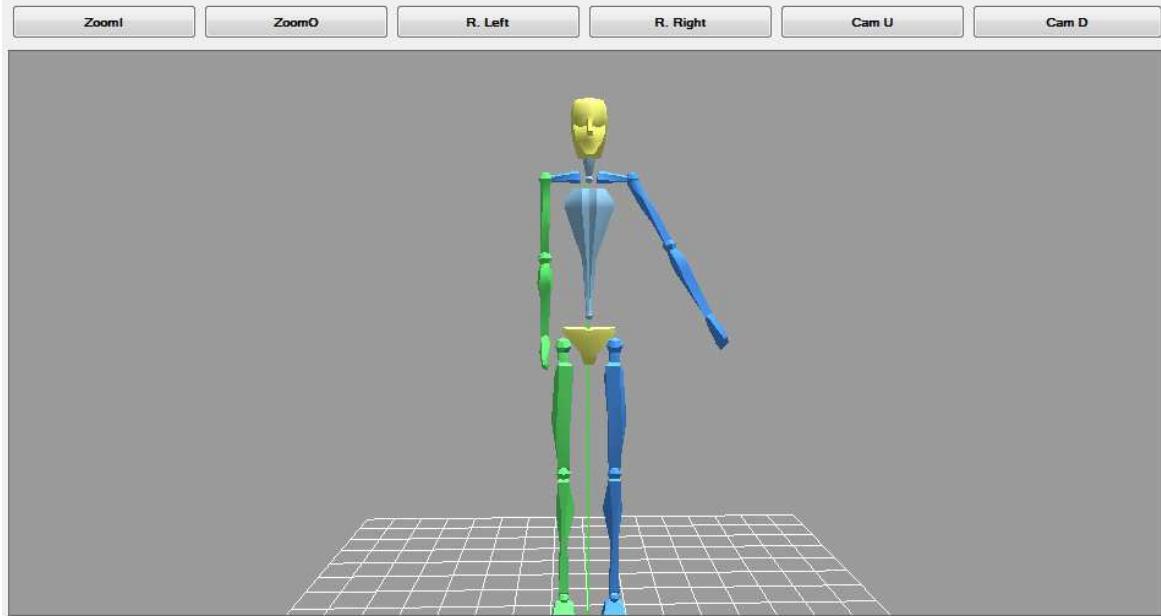


Figure 2.24: InertialCaptures's 3D avatar moving during a data gathering session.

gives our application the potential of building and supporting a complete database of monitoring sessions.

2.5 Wrap-up and future improvement of instrumentation and tools

Along this chapter we have presented all the necessary hardware, firmware and software to successfully gather MARG data measured by sensors attached to human body. We now summarize the key aspects of each one of the previous sections at the same time we comment the future steps in our research.

2.5.1 MIMUs

It is still possible to continue developing new prototypes of MIMUs. Forthcoming MIMUs should include the most recent sensors in the market—as their precision and quality is continuously improving—and have a higher local processing capability by adding small processors or small DSPs (such as the dsPIC hybrid) in order to locally perform the calibration, preprocessing and processing algorithms. They should also be lighter and smaller to facilitate their attachment to the body of the subject under test. A very important field of research is to decrease power consump-

tion. Recent studies [114], [115] have been carried out in order to find systems to harvest energy from human body motion. One way to scavenge energy from human activity is to use vibrating beam piezoelectric harvesters. These generators [116] contain piezoelectric layers that are attached to a mass at the tip. When the system moves, the mass starts to oscillate and the layers start to expand and compress which derives in the generation of power.

Nowadays, it is not possible yet to completely remove batteries as the piezoelectric harvesters do not produce enough power for the complete system. Some published works [117] show that future harvesters attached to the body might be enough to extend battery life, or even completely power the MIMU or other sensor motes in a wireless personal area sensor network.

Research could be also focused on developing low consumption transmission algorithms for the sensor network. Such algorithms may decrease the power consumption of the transducers, which represent the highest percentage of system power consumption.

An important factor to improve the usability of MIMUs and their commercialization would be their encapsulation into a hermetic case. This way they would be water resistant and could be worn by the users during all their daily routine (including shower time). It is equally important to include the possibility of charging the battery wirelessly through magnetic induction. This would ease the use of the monitoring device by the elderly as they usually have unsteady hands and issues with small connectors.

Smartphones are having a very fast development and their degree of penetration to population is constantly increasing. Therefore, we have found very interesting to use smartphones to capture and monitor human body movement. Most medium-end/high-end smartphones are equipped with MEMS triaxial MARG sensors which are similar to those of many commercial MIMUs. Taking advantage of the high computation power of processors embedded in recent smartphones, we will be developing applications to capture, monitor and classify body motion. We have preliminarily developed a series of scripts that extract and plot MARG data from files gathered with existing Android and iOS applications such as SensorList Pro [118] and SensorLog [119]. However, it is not very likely that smartphones will substitute MIMUs as they are not a cost-efficient solution for applications that need multiple measurement units placed in different points, e.g. WSNs. In addition, neither their power consumption nor size are appropriate for many applications.

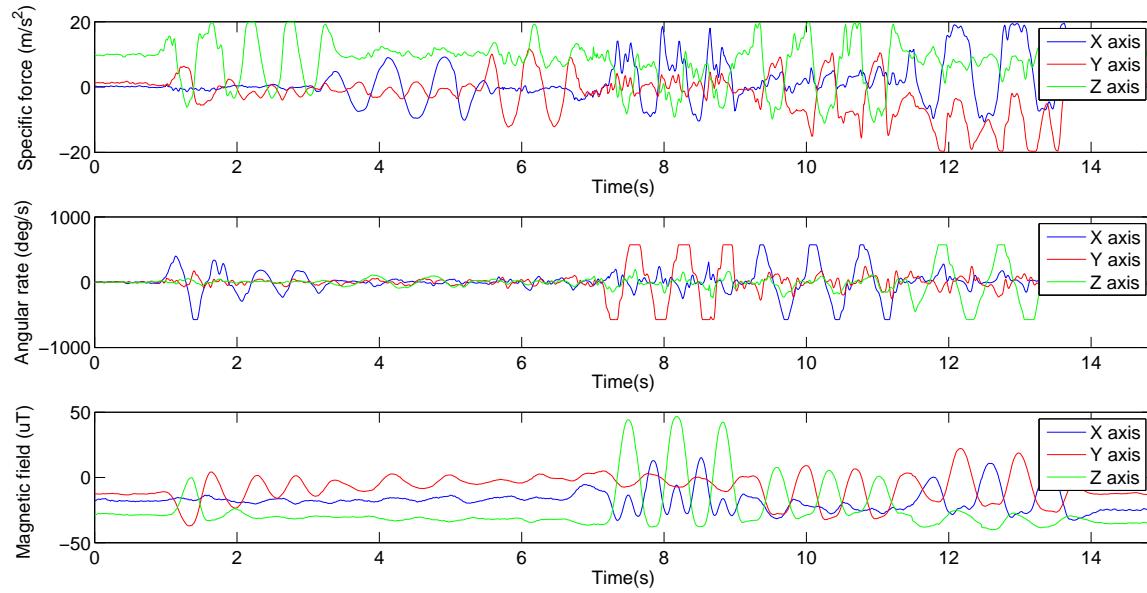


Figure 2.25: MARG data gathered using Matlab (data extraction and visualization) and SensorList Pro (data reading and storage) running on a Samsung Galaxy SII smartphone.

2.5.2 Firmware

The Firmware that is included in Wagyromag carries out a set of basic tasks to read, pack, store and/or transmit data measured by the MARG sensors.

In section 2.3 we have explained the general structure of the firmware and how it performs each one of the main process subtasks. This firmware can be improved and extended to support autocalibration and to send and receive configuration commands of the MIMU's different components (change sampling frequency, change XBEE transceiver parameters, modify dynamic range of sensors, etc.). Moreover it is desirable that future versions support remote reading of data already stored in the microSD card and optimize power consumption of all components. Moreover, algorithms specific to different applications could be implemented in the firmware.

2.5.3 Software

Furthermore we have presented the beta version of a software tool that we use to receive, plot and store data sent by IMUs to a base station. The tool is conceived as a flexible and scalable environment so it supports any MIMU by just configuring a XML file specifying all the communication parameters and the frame format.

The application represents graphically the measured values in different configurable plots that can be zoomed in and out as well as saved in different formats. In addition, we can specify the position of the MIMUs that are placed on the body

of the subject that is being monitored, so a 3D avatar moves in the same way the subject does.

Future versions of the tool will include a wizard that eases up the configuration of new MIMUs by non-advanced users, as well as spatial movement of the avatar (including inertial navigation algorithms). We also aim to give the tool the ability to support Matlab scripts including data processing algorithms and to include a sensor calibration wizard that guides the user through the different steps that need to be followed. Other feature that could be included is synchronization of video with plots and the movement of the avatar so it is easier to identify and analyze the movement of the subject under study.

Finally, we have plans on developing a web platform that allows remote visualization in real time of the movement of the subject being monitored as well as storage of gathered data so they can be checked in any moment.

Chapter 3

Calibration

3.1 Introduction and chapter's structure

One of the most accurate and correct definitions of calibration was proposed by Chatfield [120], who stated:

...

"Calibration is the process of comparing instrument outputs with known reference information and determining the coefficients that force the output to agree with the reference information over a range of output values".

...

Therefore, along this chapter we will introduce the main errors that disrupt the output of MARG sensors and the large number of existing techniques and algorithms to compensate them and minimize their associated negative effects.

When working with sensors, calibration is one of the most important, delicate and meticulous processes that needs to be carried out. Prior to the calibration process, the information at the sensors' output will be a signal composed of integer numbers, or real numbers (if the information is not processed by an Analog/Digital Converter) bounded into a range which is determined by the precision of the sensors and converters. These numeric values lack of physical value, so it is strictly necessary to convert them into a scale that can be measured in physical units. Moreover, the calibration will help to reduce the undesired components and errors of the

output caused by different kinds of effects that will be explained in this section.

Depending on the sensor being calibrated, there is a wide range of calibration procedures that are based on different principles. First of all, we can classify them according to the moment where the procedure is carried out. Most algorithms are applied before the use of the MIMU, but, occasionally, we can apply automatic algorithms that recalibrate the sensor while it is being used. Such automatic systems are recommended for lower quality sensors that present a time varying behavior and are also very sensitive to temperature changes.

On the other hand, we can classify the calibration procedures according to the employed instrumentation. For military, industrial and aviation applications the sensors need to be very accurate and so has the calibration equipment. Accurate calibration equipment, usually composed of computer controlled multi axis rate tables including a thermal chamber [121–123] is really expensive.

The companies using military, industrial or aviation sensors can afford buying such instrumentation as their cost represents a very small percentage of the whole budget. However, for low-cost motion monitoring applications the employed instrumentation has to be very cheap and simple as the cost of a rate table is at least up to 250 times the cost of our prototypes and some commercial MIMUs.

Most calibration algorithms are based on the work developed by Franco Ferraris [32] in 1995. Ferraris proposed a procedure to calibrate triaxial gyroscopes and accelerometers without using any kind of instrumentation. From there, many works have proposed variations of the original algorithm. To calibrate magnetometers most authors use the ellipsoid fitting procedure [124], [125] and [126], which is based on gathering a large dataset by randomly moving the MIMU around all the axes and trying to cover all the space. The gathered points are fitted into an ellipsoid which semi-axes and origin are computed to transform it into an ideal sphere centered in the origin of radius equal to the magnitude of the local Earth's magnetic field value.

This chapter aims to explain the main calibration errors present in MEMS MARG sensors (section 3.2) and the mathematical models used to define them. Section 3.4 includes a comparative study which is carried out between a selection of the most complete and relevant algorithms. The chapter continues with the definition of a novel algorithm that helps the user to gather well-distributed data to be used in calibration algorithms to optimize the estimation of calibration parameters and two calibration procedures that can be employed to calibrate gyroscopes without sophisticated equipment. All three algorithms are presented in section 3.5. Finally, the chapter concludes with a wrap-up of the contents developed along it and with a brief reflection of future research in the calibration field.

3.2 Sensor errors and undesired effects

Ideally, assuming that we are using linear sensors, if we wish to transform raw sensor data into physical units, we would only need to find a calibration line defined by a scale factor and a bias. Such calibration line can be expressed as follows,

$$u_{\text{calibrated}_i} = k_i u_{\text{raw}_i} - b_i \quad (3.1)$$

where $u_{\text{calibrated}_i}$ is the output of axis i once it has been transformed to physical units, k_i is the scale factor of axis i , u_{raw_i} is the raw output of axis i and b_i is the offset, also from axis i . The scale factor and the bias are constant under ideal circumstances.

However, the sensor triad presents a number of errors which are not modeled, and consequently not compensated for, using 3.1. To begin with, biases are usually time varying and scale factors may not be linear. In addition, there are other effects such as axis non-orthogonality and frame misalignment, which are common to accelerometers, magnetometers and gyroscopes, and other phenomena that are specific to each kind of sensor. Therefore we need to define a more complex model to calibrate the sensor and compensate for these undesired effects. Prior to defining the model, we will briefly define and explain the most relevant errors of MEMS MARG sensors.

3.2.1 Errors common to MEMS MARG sensors

Triaxial MARG MEMS sensors may suffer from non-orthogonality and misalignment effects.

- Non-orthogonality effects appear when the axes of the sensor triad are not perfectly orthogonal to each other. When the triad is perfectly orthogonal, each axis measures only the acceleration and magnetic field along its axis or the angular rate around it. For example, when the object is being accelerated only along Z axis, the other two axes should have a null output. This does not happen when the triad is not orthogonal as X and Y axes will also measure acceleration. The amount of acceleration they measure depends on the size of the deviation angles of each axis from the orthogonal position.
- Misalignment effects come from an improper alignment of the sensor frame with the body frame. We may consider different chained misalignments, for example, the MEMS chip with respect to the PCB, the PCB with respect to the MIMU case and occasionally the case of the MIMU with respect to the surface of the object to which it is attached. We will consider that the chip and the PCB and the chip are perfectly aligned and will only consider the misalignment of the PCB with respect to the case of the MIMU. Misalignment is, therefore, a

displacement of one reference frame with respect to another usually caused during the manufacturing or assembly process.

3.2.2 Errors specific to the kind of sensor

3.2.2.1 Magnetometer

Besides the unwanted effects mentioned above, the output of magnetometers is distorted by wide band measurement noise and random biases caused by magnetic interference around the sensors [127]. These undesired magnetic fields can be classified according to their cause. The first category consists of constant or slow time-varying magnetic fields which are generated by ferromagnetic materials placed in the vicinity. Errors caused by these perturbations are known as hard iron biases and are translated into null-shifts (variations of the null bias). The effect of hard iron errors can be avoided or minimized by properly designing the MIMU so no materials which may potentially cause distortion are placed close to the magnetometer. It can also be easily eliminated by removing the constant bias.

The second group of disruptive magnetic fields comes from materials that generate a magnetic field when they are excited with an applied external field. The generated field depends on the magnitude and the direction of the external field. These kind of materials are often referred to as soft irons so the errors caused by them will be known as soft iron errors. Soft iron errors translate into time varying biases.

3.2.2.2 Gyroscope

Gyroscopes suffer from a time-varying bias which is mainly caused by self-heating effects derived from power-on. The speed of the variation depends on the quality of the sensor. Once the sensor has been powered on and left idle for a period of time which can vary from a few seconds to a few minutes, the bias stabilizes or continues to grow much more slowly. This dynamic drift behavior leads to an even larger drift when the angular rate is integrated to estimate the rotation around an axis (see chapter 5). Since the bias varies according to the variation of temperature during power-on, it also varies according to the temperature of the environment in which it is being used. Under normal operation conditions, the temperature of the environment will vary very slowly, but the bias will need to be recomputed if the ambient temperature has considerably varied from last calibration.

A possible way to compensate for the time growing bias is to characterize the output of the gyroscope according to the temperature and applying the following

expression [128]:

$$b_g(t) = b_{g0} + \alpha_t \Delta T \quad (3.2)$$

where $b_g(t)$ is the bias at instant t , b_{g0} is the bias at $t = 0$, α_t is the temperature coefficient and ΔT is the time period between $t = 0$ and $t = T$.

All the effects enumerated along this section can be modeled mathematically in the form of parameters that can be estimated and subsequently compensated for.

3.3 Sensor modeling

In this section we will explain the mathematical model used in [44] to represent the output of the sensors and include the parameters and errors common to all sensors. The model is general and very complete and can be used for the three sensors by simplifying it or completing it depending on the employed sensor.

The general triaxial sensor model is written in matrix form, where index k indicates the type of the sensor (a for accelerometer, g for gyroscope and m for magnetometer). The, the sensor model is as follows:

$$\mathbf{y}_k = S_k T_k M_k \mathbf{u}_k + \mathbf{b}_k \quad (3.3)$$

where \mathbf{y}_k represents the output of sensor, S_k the scale factors matrix (also known as sensitivity matrix), T_k the orthogonalization matrix, M_k the misalignment matrix, \mathbf{u}_k the actual measured quantity and \mathbf{b}_k the bias, all from sensor k . The vectors are composed in the following way:

$$\mathbf{u}_k = \begin{bmatrix} u_{kx} \\ u_{ky} \\ u_{kz} \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} y_{kx} \\ y_{ky} \\ y_{kz} \end{bmatrix}, \quad \mathbf{b}_k = \begin{bmatrix} b_{kx} \\ b_{ky} \\ b_{kz} \end{bmatrix} \quad (3.4)$$

and the matrices have the following form:

$$S_k = \begin{bmatrix} s_{kx} & 0 & 0 \\ 0 & s_{ky} & 0 \\ 0 & 0 & s_{kz} \end{bmatrix} \quad (3.5)$$

where s_{kx} , s_{ky} and s_{kz} are the scale factors of X, Y and Z axes, respectively.

$$T_k = \begin{bmatrix} 1 & 0 & 0 \\ \cos \alpha_k & 1 & 0 \\ \cos \beta_k & \cos \gamma_k & 1 \end{bmatrix} \quad (3.6)$$

where α represents the displacement angle of Y' axis with respect to X axis, β the displacement angle of Z' axis with respect to X axis and γ the displacement angle of

Z' axis with respect to Y axis. In a nutshell, the orthogonalization matrix T_k transforms the vector expressed in the orthogonal sensor reference frame XYZ into the vector expressed in the non-orthogonal sensor reference frame XYZ' as depicted in figure 3.1. The misalignment matrix M_k is a rotation matrix (see chapter 5 for more information about rotation matrices) which converts the frame XYZ' in a rotated frame XYZ'' .

$$M_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_k & \sin \phi_k \\ 0 & -\sin \phi_k & \cos \phi_k \end{bmatrix} \begin{bmatrix} \cos \theta_k & 0 & -\sin \theta_k \\ 0 & 1 & 0 \\ \sin \theta_k & 0 & \cos \theta_k \end{bmatrix} \begin{bmatrix} \cos \psi_k & \sin \psi_k & 0 \\ -\sin \psi_k & \cos \psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_k = \begin{bmatrix} \cos \theta_k \cos \psi_k & \cos \theta_k \sin \psi_k & -\sin \theta_k \\ \sin \phi_k \sin \theta_k \cos \phi_k - \cos \phi_k \sin \psi_k & \sin \phi_k \sin \theta_k \sin \psi_k + \cos \phi_k \cos \psi_k & \sin \phi_k \cos \theta_k \\ \cos \phi_k \sin \theta_k \cos \psi_k + \sin \phi_k \sin \psi_k & \cos \phi_k \sin \theta_k \sin \psi_k - \sin \phi_k \cos \psi_k & \cos \phi_k \cos \theta_k \end{bmatrix} \quad (3.7)$$

where ϕ_k, θ_k and ψ_k are the rotation angles around X' , Y' and Z' axes respectively.

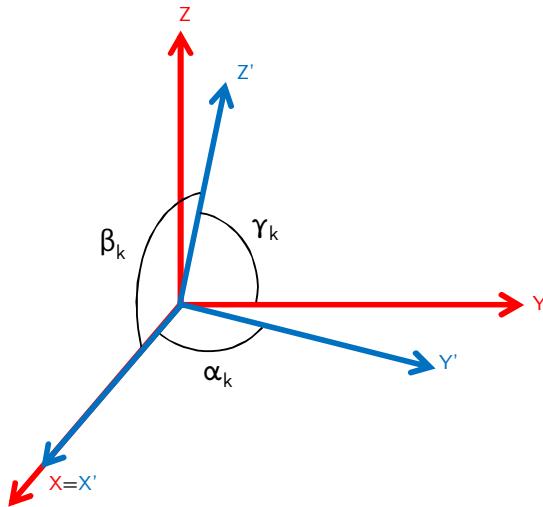


Figure 3.1: Representation of non-orthogonal reference frame XYZ' with respect to the orthogonal reference frame XYZ .

Therefore, twelve parameters must be determined to compensate for all the effects. These parameters can be divided into two sets: mechanical parameters and electrical parameters. Mechanical parameters—orthogonalization and misalignment parameters—, are independent of temperature and time and, therefore, need to be determined only once. On the other hand, electrical parameters—like the biases—, have to be periodically recomputed.

Finally, we can obtain the estimate $\hat{\mathbf{u}}_k$ for the observed physical quantity \mathbf{u}_k by reverting the effects, that is, multiplying the output of the sensor by the inverse matrices aforementioned as follows:

$$\hat{\mathbf{u}}_k = M_k^{-1} T_k^{-1} S_k^{-1} (\mathbf{y}_k - \mathbf{b}_k) \quad (3.8)$$

As we said before, this is a general model that can be extended or reduced to add more errors or neglect some of them. If we are using high quality sensors and the MIMU has been carefully assembled we may be able to neglect orthogonalization and misalignment errors.

3.4 Comparative study of calibration algorithms

At the end of the state of the art we noticed that there was a large number of works presenting calibration algorithms for MEMS MARG sensors but also that most of them were very similar and differed only in some aspects related to the complexity of the sensor model, the procedure to gather the data, the function to be minimized and the algorithm to minimize it. Therefore, we decided to select a set of highly cited works in order to implement them and carry out a preliminary comparison to study their performance.

Since most gyroscope calibration methods employ complex and expensive rate tables and, unfortunately, we were lacking of such equipment, we were forced to study just accelerometer and magnetometer methods. After discarding a large number of methods, we selected twelve of them which are next described in detail.

3.4.1 Selected algorithms

3.4.1.1 Method 1: Ferrari's in-field procedure

The choice of the first method was clear since the beginning of the selection. We thought that Ferraris [32] procedure had to be one of the selected algorithms since it is the most cited and most famous calibration method in the last fifteen years.

Ferraris describes an efficient calibration method which is based on a set of full angle rotations for the gyroscope and the local value of the gravity vector for the accelerometer. In addition, the required equipment is very simple.

- **Sensor modeling:** The output of the accelerometer and the gyroscope— \mathbf{u}_a and \mathbf{u}_g respectively—, are modeled through the following expressions:

$$\mathbf{u}_g = K_g R_g \omega + \mathbf{d}_{g,a} + \mathbf{b}_g \quad (3.9)$$

$$\mathbf{u}_a = K_a R_a \mathbf{a} + \mathbf{b}_a \quad (3.10)$$

where ω and \mathbf{a} represent the actual values of the sensed angular rate and the acceleration, respectively. R is the orientation (misalignment) matrix, K is the scale factor matrix, \mathbf{b} is the bias vector and $\mathbf{d}_{g,a}$ is used to model the effect of

linear acceleration on the gyroscope.

The scale factor matrix is a diagonal matrix formed by the scale factor of each axis expressed in a Cartesian reference system $\Sigma = \langle O, x, y, z \rangle$.

$$K = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \quad (3.11)$$

On the other hand, R rectifies the orientation of each sensitivity axis so they form a reference system which axes are parallel to Σ . Each one of the rows represent the transposition of the unit vectors r_x, r_y and r_z that identify the three axis of the sensor. That is, the orientation matrix is equivalent to the misalignment matrix defined at the beginning of the chapter. Therefore, the orientation matrix is as follows:

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \quad (3.12)$$

- **Calibration procedure:** To calibrate the gyroscope we need to estimate eighteen parameters: three biases, three scale factors, three acceleration sensitivity components and the nine components of the orientation matrix.

The estimation of bias can be carried out in a separate way for each one of the axes of the gyroscope by averaging the output while the MIMU is held static. Likewise, the linear acceleration sensitivity is estimated by setting each axis parallel and antiparallel, gathering the output of the gyroscope and applying the following equation:

$$k_{g,a,\nu} = \frac{u_{g,P,\nu} - u_{g,A,\nu}}{2g} \quad (3.13)$$

where $k_{g,a,\nu}$ is the gyro acceleration sensitivity coefficient of axes $\nu=x, y, z$, $u_{g,P,\nu}$ and $u_{g,A,\nu}$ are the measured outputs of the gyroscope when its axes are set parallel and antiparallel to gravity respectively, and g stands for the acceleration magnitude.

The estimation of the orientation matrix and the scale factors can be carried out applying a known angular rate to each one of the axes. The nine measurements—the output of the three axes when each one of them is being rotated—, are subsequently corrected in bias and acceleration and can be grouped into the following matrix:

$$U_g = K_g R_g \Omega \quad (3.14)$$

where Ω is a matrix containing the three applied angular velocities $\omega_x, \omega_y, \omega_z$ in the diagonal. So, knowing Ω , we can estimate the product $K_g R_g$ using

$$K_g R_g = U_g \Omega^{-1} \quad (3.15)$$

We can separate the product to obtain K_g and R_g using the fact that the rows of R_g are unitary vectors,

$$\begin{bmatrix} k_{g,x}^2 \\ k_{g,y}^2 \\ k_{g,z}^2 \end{bmatrix} = \text{diag}\left[\left(U_g\Omega^{-1}\right)\left(U_g\Omega^{-1}\right)^T\right]$$

$$R_g = K_g^{-1}U_g\Omega^{-1} \quad (3.16)$$

This procedure requires a reference angular rate to build Ω but, if no equipment generating a known angular rate is available, we can integrate both sides of (3.14) and substitute the angular rate reference by a known rotation angle.

$$\int_{t_1}^{t_2} U_g dt = \int_{t_1}^{t_2} K_g R_g \Omega dt = K_g R_g \int_{t_1}^{t_2} \Omega dt \quad (3.17)$$

If we designate the leftmost integral of the equation above as W_g and Θ as the rotation angle between t_1 and t_2 ,

$$W_g = \int_{t_1}^{t_2} U_g dt \quad \text{and} \quad \Theta = \int_{t_1}^{t_2} \Omega dt \quad (3.18)$$

we can rewrite equation (3.14) as follows:

$$W_g = K_g R_g \Theta \quad (3.19)$$

This way the calibration only requires three known rotations around each one of the axes and the integration of the measured angular rate during such rotations.

The calibration of the accelerometer is carried out by estimating the corresponding biases, scale factors and orientation matrix. The bias can be computed by measuring the output of the accelerometer before and after a rotation of 180° around a horizontal axis, so the gravity effect is reversed. In practice, we will measure the output of the accelerometer when each axis is set parallel (\mathbf{u}_{a1}) and antiparallel (\mathbf{u}_{a2}) to the gravity vector and apply the following equations:

$$\mathbf{u}_{a1} = K_a R_a g + \mathbf{b}_a \quad (3.20)$$

$$\mathbf{u}_{a2} = K_a R_a (-g) + \mathbf{b}_a$$

$$\mathbf{b}_a = \frac{\mathbf{u}_{a1} + \mathbf{u}_{a2}}{2} \quad (3.21)$$

The next step is to compute the elements of K_a and R_a . This is done by defining the following system,

$$U_{a+} = K_a R_a g + B_a \quad (3.22)$$

$$U_{a-} = K_a R_a (-g) + B_a \quad (3.23)$$

where U_{a+} is a 3×3 matrix which contains the output of the accelerometer triad when each axis is set parallel to the gravity vector placed by columns. U_{a-} is analogous to U_{a+} but placing the accelerometer antiparallel to the gravity vector. Thus, the U_{a+} and U_{a-} matrices have the following form:

$$\begin{aligned} U_{a+} &= \begin{bmatrix} U_{a+xx} & U_{a+yx} & U_{a+zx} \\ U_{a+xy} & U_{a+yy} & U_{a+zy} \\ U_{a+xz} & U_{a+yz} & U_{a+zz} \end{bmatrix} \\ U_{a-} &= \begin{bmatrix} U_{a-xx} & U_{a-yx} & U_{a-zx} \\ U_{a-xy} & U_{a-yy} & U_{a-zy} \\ U_{a-xz} & U_{a-yz} & U_{a-zz} \end{bmatrix} \end{aligned} \quad (3.24)$$

where U_{a+ij} and U_{a-ij} are the output of j axis when i axis is set parallel and antiparallel to the gravity vector, respectively. So, if we subtract U_{a+} to U_{a-} we can extract the scale factor and the orientation matrices. The steps to be followed are:

$$U_{aD} = U_{a+} - U_{a-} \quad (3.25)$$

$$U_{aD} = 2gK_a R_a \quad (3.26)$$

Analogously to the extraction of R_g and K_g , we also take into consideration that the rows of R_a are unit vectors.

$$\begin{bmatrix} k_{ax}^2 \\ k_{ay}^2 \\ k_{az}^2 \end{bmatrix} = diag[(K_a R_a)(K_a R_a)^T] = \frac{1}{4g^2} diag[U_{aD} U_{aD}^T] \quad (3.27)$$

$$K_a = \begin{bmatrix} k_{ax} & 0 & 0 \\ 0 & k_{ay} & 0 \\ 0 & 0 & k_{az} \end{bmatrix} \quad (3.28)$$

Finally, the orientation matrix can be found applying

$$R_a = \frac{1}{2g} K_a^{-1} U_{aD} \quad (3.29)$$

Once we have computed all the calibration parameters, we can correct the output of the accelerometer and the gyroscope using

$$\begin{aligned} \hat{\omega} &= R_g^{-1} K_g^{-1} (\mathbf{u}_g - \mathbf{d}_{g,a} - \mathbf{b}_g) \\ \hat{\mathbf{a}} &= R_a^{-1} K_a^{-1} (\mathbf{u}_a - \mathbf{b}_a) \end{aligned} \quad (3.30)$$

3.4.1.2 Method 2: Lotters' quasi-static calibration

Lotters et al.[124] present an algorithm which is based on the fact that the magnitude of the acceleration vector has to be close to 1 g under quasi-static conditions. Their calibration method does not require explicit knowledge of the orientation of the accelerometer's axes with respect to the gravity vector and, therefore, the data needed to estimate the calibration parameters is gathered when the MIMU is placed in random positions. The algorithm tries to compensate for the dynamic bias by recomputing the null bias when the MIMU is static, quasi-static or moving under a quasi-constant acceleration. Quasi-static instants are computed using a filtered-rectifier detector which is tested and explained in next chapter's section 4.3.1.5.

- **Sensor modeling:** The actual acceleration measured by the accelerometer can be linearly related to its output applying the following model:

$$v_\mu = s_\mu a_\mu + b_\mu \quad (3.31)$$

where v (V) is the output voltage, s ($V \cdot g^{-1}$) the scale factor, a (g) the measured acceleration, b (V) the bias and μ (X, Y or Z) is the sensitivity axis.

- **Calibration procedure:** The goal of the algorithm is to estimate the calibration parameters defining a function which is to be adjusted employing a minimum variance linear estimator. The function establishes that the measured acceleration magnitude must be equal to the local value of the gravity.

$$h(\mathbf{v}, \mathbf{p}) = \sqrt{\left(\frac{v_x - b_x}{s_x}\right)^2 + \left(\frac{v_y - b_y}{s_y}\right)^2 + \left(\frac{v_z - b_z}{s_z}\right)^2} \quad (3.32)$$

The general equation of the minimum variance linear estimator is

$$y = h(\mathbf{v}, \mathbf{p}) + \mu \quad (3.33)$$

where y is the actual measured acceleration, $\mathbf{p} = [s_x \ s_y \ s_z \ b_x \ b_y \ b_z]$ is the parameters vector, $\mathbf{v} = [v_x \ v_y \ v_z]$ is the output of the accelerometer, and μ the noise representing the deviations from 1 g when the measurements are gathered under quasi-static conditions.

The authors define the initial values of the parameter vectors applying Ferraris' procedure. The complete definition of the minimum variance linear estimator has been omitted for the sake of brevity, but can be found in the original paper [124].

3.4.1.3 Method 3: Campolo's multi-position procedure

Campolo et al. [41], present a novel calibration procedure for accelerometers and magnetometers. Its novelty lies in not needing to know the magnitude nor the direction of the gravity and Earth's magnetic field and also in the simplicity of the calibration maneuvers.

- **Sensor modeling:** The measured magnitude (acceleration or earth's magnetic field) is related to the output of the sensor as follows:

$$m_\mu = \frac{v_\mu - b_\mu}{s_\mu} \quad (3.34)$$

where s_μ and b_μ are the scale factor and the bias of axis μ , respectively, and are the parameters to be estimated in order to calibrate the sensor.

- **Calibration procedure:** The proposed algorithm is based on the fact that rotation movements in the Euclidean 3D space describe linear transformations in which vectors preserve their magnitude so that:

$$m_x^2 + m_y^2 + m_z^2 = \|m\|^2 \quad (3.35)$$

that, is, when the MIMU is being displaced and rotated in space, assuming constant linear acceleration and no external magnetic perturbations, the magnitudes of the Earth's magnetic and gravity fields are constant. Hence, plugging (3.34) in (3.35) yields:

$$\left(\frac{v_x - b_x}{s_x}\right)^2 + \left(\frac{v_y - b_y}{s_y}\right)^2 + \left(\frac{v_z - b_z}{s_z}\right)^2 = m^2 \quad (3.36)$$

where m is the magnitude of the local value of Earth's magnetic or gravity vector. Equation (3.36) can be rewritten as:

$$\left(\frac{v_x - b_x}{ms_x}\right)^2 + \left(\frac{v_y - b_y}{ms_y}\right)^2 + \left(\frac{v_z - b_z}{ms_z}\right)^2 = 1 \quad (3.37)$$

This last equation is equivalent to the equation of an ellipsoid which is characterized by six parameters, namely $\mathbf{p}=(b_x, b_y, b_z, \|m\|s_x, \|m\|s_y, \|m\|s_z)$, where the three first ones indicate the coordinates of the ellipsoid center while the last three determine the length of each semi-axis.

The algorithm needs $N > 6$ measurements derived from different positions of the MIMU to carry out a least squares adjustment and estimate the calibration parameters. To carry out such adjustment, we first have to linearize the following N non-linear equations:

$$\left(\frac{v_{ix} - b_x}{ms_x}\right)^2 + \left(\frac{v_{iy} - b_y}{ms_y}\right)^2 + \left(\frac{v_{iz} - b_z}{ms_z}\right)^2 - 1 = 0 \quad (3.38)$$

which leads to the following cost function:

$$e_i(\mathbf{p}) \triangleq \sqrt{(v_{ix} - b_x)^2 + (v_{iy} - b_y)^2 + (v_{iz} - b_z)^2} \cdot \left(1 - \frac{1}{\left(\frac{v_{ix}-b_x}{ms_x} \right)^2 + \left(\frac{v_{iy}-b_y}{ms_y} \right)^2 + \left(\frac{v_{iz}-b_z}{ms_z} \right)^2} \right) \quad (3.39)$$

Consequently, the parameters vector \mathbf{p} is estimated minimizing the cost function

$$\min_{\mathbf{p}} \sum_{i=1}^N e_i^2(\mathbf{p}) \quad (3.40)$$

The authors use Matlab's *lsqnonlin* built-in function to minimize the cost function and employ the following values for the initial parameters vector \mathbf{p}_0 :

$$\mathbf{p}_0 \triangleq [\bar{v}_x \quad \bar{v}_y \quad \bar{v}_z \quad \bar{r} \quad \bar{r} \quad \bar{r}] \quad (3.41)$$

where \bar{v}_x , \bar{v}_y , \bar{v}_z and \bar{r} are defined as:

$$\begin{aligned} \bar{v}_x &= \frac{1}{N} \sum_{i=1}^N v_{xi}; & \bar{v}_y &= \frac{1}{N} \sum_{i=1}^N v_{yi}; & \bar{v}_z &= \frac{1}{N} \sum_{i=1}^N v_{zi}; \\ \bar{r} &= \frac{1}{N} \sum_{i=1}^N \sqrt{(v_{xi} - \bar{v}_x)^2 + (v_{yi} - \bar{v}_y)^2 + (v_{zi} - \bar{v}_z)^2}; \end{aligned} \quad (3.42)$$

3.4.1.4 Method 4: Camps' multi-position procedure

Camps et al. present in [52] theoretical and experimental steps of a 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:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} m_x(t) \\ m_y(t) \\ m_z(t) \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} \quad (3.43)$$

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:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{xy} & s_{yy} & s_{yz} \\ s_{xz} & s_{yz} & s_{zz} \end{bmatrix} \begin{bmatrix} m_x(t) \\ m_y(t) \\ m_z(t) \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} \quad (3.44)$$

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:

$$\|\mathbf{m}\|^2 = m_x(t)^2 + m_y(t)^2 + m_z(t)^2 \quad (3.45)$$

$$\|\mathbf{h}\|^2 = \left(\frac{v_x(t) - b_x}{s_x} \right)^2 + \left(\frac{v_y(t) - b_y}{s_y} \right)^2 + \left(\frac{v_z(t) - b_z}{s_z} \right)^2 \quad (3.46)$$

Equation (3.46) 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

$$e_p(t) = \|\mathbf{m}\|^2 - (\mathbf{v}(t) - \mathbf{b})^T (S^{-1})^2 (\mathbf{v}(t) - \mathbf{b}) \quad (3.47)$$

where

$$S^{-1} = \begin{bmatrix} 1/s_{xx} & 1/s_{xy} & 1/s_{xz} \\ 1/s_{xy} & 1/s_{yy} & 1/s_{yz} \\ 1/s_{xz} & 1/s_{yz} & 1/s_{zz} \end{bmatrix} \quad (3.48)$$

The cost function $e_p(t)$ is quadratic, and is minimized iteratively by the *Levenberg-Marquardt* algorithm (LMA) [129, 130].

3.4.1.5 Method 5: Artese's multi-position procedure

G. Artese and A. Tecroci [131] show a multi-position calibration algorithm for accelerometers (and also gyroscopes) which does not require precise alignment of the sensors with the case of the device nor perfect orthogonality of the sensor triad.

- **Sensor modeling:** The accelerometer model is described by the model below:

$$\begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} = \begin{bmatrix} 1+s_x & 0 & 0 \\ -\theta_{yz} & 1+s_y & 0 \\ \theta_{zx} & -\theta_{zy} & 1+s_z \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (3.49)$$

where l_μ is the output of axis, s_μ the scale factor and b_μ the bias, all of axes $\mu = x, y, z$. θ_{ij} represents the angle formed by axis i with respect to axis j and m_μ is the actual acceleration to which the MIMU is being subjected.

- **Calibration procedure:** Under quasi-static conditions the magnitude of the measured acceleration has to be equal to the magnitude of the gravity vector. Using this fact we get,

$$l_x^2 + l_y^2 + l_z^2 - g^2 = 0 \quad (3.50)$$

If we plug equation (3.49) in (3.50), the error function is expressed in terms of the parameters. Next, we need to gather acceleration data coming from more than 9 quasi-static positions in order to be able to estimate the parameters. Once we have gathered the data, we use it to find the optimal calibration parameters that minimize the error function. This is done applying a least squares procedure.

3.4.1.6 Method 6: Jurman's multi-position procedure

Jurman et al. describe in [44] a procedure to calibrate an inertial device composed of three triaxial sensors: an accelerometer, a gyroscope and a magnetometer. It presents a very complete and general model for the sensors and defines an error function which is minimized to find the calibration parameters. Data is gathered from different random positions.

- **Sensor modeling:** The model presented in this work is the one we have explained in section 3.3. In our opinion this is the most complete, general and accurate model as it includes the most important undesired effects present in the output of MARG sensors.
- **Calibration procedure:** The calibration procedure is composed of two parts: first, the scale factors, biases, and orthogonality errors are estimated and then, in the second part, the misalignment parameters are found. The error function is defined as the difference between a reference quantity and the magnitude of the measured quantity. For the case of the accelerometer and the magnetometer, the reference is the magnitude of the Earth's gravity vector and Earth's magnetic field vector, respectively. For the gyroscope, a rate table is needed to rotate the MIMU at a known angular rate. The error function is defined below.

$$e(\mathbf{p}) = \frac{1}{N} \sum_{n=1}^N (u_{\text{ref}} - u_n(\mathbf{p}))^2 \quad (3.51)$$

Where u_{ref} is the reference magnitude and u_n is the output of the sensor modeled by equations in section 3.3. Data are gathered differently for each sensor. The acceleration data are gathered by setting the MIMU in N different

quasi-static positions while the magnetic field data are gathered by moving the MIMU randomly trying to cover all the space. These data can be gathered while the MIMU is subjected to linear acceleration so the procedure is much faster than for the accelerometer. On the other hand, the angular rate data are gathered while fixing the MIMU in N different orientations with respect to the surface of a table which spins at a known rate. If no multi-axis rate tables are available the procedure is very cumbersome to carry out.

Finally, once we have gathered the necessary data, it is plugged in the error function and the parameters are found applying the Gauss-Newton minimization algorithm.

3.4.1.7 Method 7: Won's iterative multi-position procedure

A six parameters model is presented in [33]. The authors state that the main advantages of the proposed algorithm are the low computational cost due to the high convergence speed and the easy steps that need to be followed to carry out the complete calibration. More specifically, the MIMU has to be put only in the famous six positions.

- **Sensor modeling:** The relation between the output of each one of the accelerometer's axes (S_{axis}) and the actual value of the acceleration (A_{axis}) is modeled through the next equation:

$$S_{\text{axis}} = G_{\text{axis}} A_{\text{axis}} + B_{\text{axis}} \quad (3.52)$$

where G_{axis} and B_{axis} are the scale factor and bias of each axis.

- **Calibration procedure:** The proposed method determines the three scale factors and biases in an iterative way. We can rewrite (3.52) in the following way:

$$\begin{aligned} S_{\text{axis}} &= G_{\text{axis}} A_{\text{axis}} + B_{\text{axis}} = \left(\prod_{i=0}^k \tilde{G}_{\text{axis},i} \right) \hat{A}_{\text{axis},k} + \sum_{i=0}^k \tilde{B}_{\text{axis},i} = \\ &= \hat{G}_{\text{axis},k} \hat{A}_{\text{axis},k} + \hat{B}_{\text{axis},k} \end{aligned} \quad (3.53)$$

where $\tilde{G}_{\text{axis},i}$ and $\tilde{B}_{\text{axis},i}$ represent the scale factor and bias for each one of the axes at the i^{th} iteration. $\hat{A}_{\text{axis},k}$, $\hat{G}_{\text{axis},k}$ y $\hat{B}_{\text{axis},k}$ represent the estimation of acceleration, scale factor and bias of each axis at the k^{th} iteration. Hence, the acceleration can be estimated as:

$$\hat{A}_{\text{axis},k-1} = \frac{S_{\text{axis}} - \hat{B}_{\text{axis},k-1}}{\hat{G}_{\text{axis},k-1}} \quad (3.54)$$

The iteration process is stopped when the error parameter is lower than a pre-defined threshold. The error is defined as follows:

$$\begin{aligned} E_{k-1} = & \left(1 - 1/\tilde{G}_{X,k}^2\right) \hat{A}_{X,k-1}^2 + 2\tilde{B}_{X,k}\hat{A}_{X,k-1}/\left(\hat{G}_{X,k-1}\tilde{G}_{X,k}^2\right) + \\ & + \left(1 - 1/\tilde{G}_{Y,k}^2\right) \hat{A}_{Y,k-1}^2 + 2\tilde{B}_{Y,k}\hat{A}_{Y,k-1}/\left(\hat{G}_{Y,k-1}\tilde{G}_{Y,k}^2\right) + \\ & + \left(1 - 1/\tilde{G}_{Z,k}^2\right) \hat{A}_{Z,k-1}^2 + 2\tilde{B}_{Z,k}\hat{A}_{Z,k-1}/\left(\hat{G}_{Z,k-1}\tilde{G}_{Z,k}^2\right) \end{aligned} \quad (3.55)$$

The error can also be expressed as a matrix,

$$\begin{aligned} [\text{Error}_{k-1}] &= [\text{Accel}_{k-1}] \cdot [\text{Cal}_k] \\ \begin{bmatrix} E_{1,k-1} \\ E_{2,k-1} \\ E_{3,k-1} \\ E_{4,k-1} \\ E_{5,k-1} \\ E_{6,k-1} \end{bmatrix} &= \begin{bmatrix} \hat{A}_{X,k-1,1}^2 & \hat{A}_{Y,k-1,1}^2 & \hat{A}_{Z,k-1,1}^2 & \hat{A}_{X,k-1,1} & \hat{A}_{Y,k-1,1} & \hat{A}_{Z,k-1,1} \\ \hat{A}_{X,k-1,2}^2 & \hat{A}_{Y,k-1,2}^2 & \hat{A}_{Z,k-1,2}^2 & \hat{A}_{X,k-1,2} & \hat{A}_{Y,k-1,2} & \hat{A}_{Z,k-1,2} \\ \hat{A}_{X,k-1,3}^2 & \hat{A}_{Y,k-1,3}^2 & \hat{A}_{Z,k-1,3}^2 & \hat{A}_{X,k-1,3} & \hat{A}_{Y,k-1,3} & \hat{A}_{Z,k-1,3} \\ \hat{A}_{X,k-1,4}^2 & \hat{A}_{Y,k-1,4}^2 & \hat{A}_{Z,k-1,4}^2 & \hat{A}_{X,k-1,4} & \hat{A}_{Y,k-1,4} & \hat{A}_{Z,k-1,4} \\ \hat{A}_{X,k-1,5}^2 & \hat{A}_{Y,k-1,5}^2 & \hat{A}_{Z,k-1,5}^2 & \hat{A}_{X,k-1,5} & \hat{A}_{Y,k-1,5} & \hat{A}_{Z,k-1,5} \\ \hat{A}_{X,k-1,6}^2 & \hat{A}_{Y,k-1,6}^2 & \hat{A}_{Z,k-1,6}^2 & \hat{A}_{X,k-1,6} & \hat{A}_{Y,k-1,6} & \hat{A}_{Z,k-1,6} \end{bmatrix} \cdot \\ &\cdot \begin{bmatrix} 1 - 1/\tilde{G}_{X,k}^2 \\ 1 - 1/\tilde{G}_{Y,k}^2 \\ 1 - 1/\tilde{G}_{Z,k}^2 \\ 2 \cdot \tilde{B}_{X,k}/(\hat{G}_{X,k-1} \cdot \tilde{G}_{X,k-1}^2) \\ 2 \cdot \tilde{B}_{Y,k}/(\hat{G}_{Y,k-1} \cdot \tilde{G}_{Y,k-1}^2) \\ 2 \cdot \tilde{B}_{Z,k}/(\hat{G}_{Z,k-1} \cdot \tilde{G}_{Z,k-1}^2) \end{bmatrix} \end{aligned} \quad (3.56)$$

where $\tilde{A}_{\mu,k-1,n}$ is the output of the accelerometer of axis μ in position n (1 to 6) at instant $k-1$. Since both matrices $[\text{Error}_{k-1}]$ and $[\text{Accel}_{k-1}]$ are known, we can compute the calibration parameters vector $[\text{Cal}_k]$ as follows:

$$[\text{Cal}_k] = [\text{Accel}_{k-1}]^{-1} \cdot [\text{Error}_{k-1}] \quad (3.57)$$

Once we have computed it, we extract the scale factors from the first three rows and use them to find the bias parameters from the last three rows. The estimation of the acceleration is finally found from equation (3.52):

$$A_{\text{axis}} = \frac{S_{\text{axis}} - B_{\text{axis}}}{G_{\text{axis}}} \quad (3.58)$$

3.4.1.8 Method 8: Frosio's multi-position procedure

Frosio et al. propose in [42] a multi-position calibration method which is also based on the fact that any inertial device is only affected by the gravity acceleration under quasi-static conditions.

- **Sensor modeling:** The model that characterizes the output of the accelerometer can be expressed in matrix form as follows:

$$\mathbf{a} = \mathbf{S}(\mathbf{v} - \mathbf{b}) \quad (3.59)$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \left(\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) \quad (3.60)$$

where \mathbf{a} , \mathbf{v} and \mathbf{b} are the vectors containing the actual acceleration components, the output of the accelerometer and the biases, respectively. Matrix \mathbf{S} includes the scale factors and the misalignment angles. It is important to remark that it is symmetric as $s_{xy} = s_{yx}$, $s_{xz} = s_{zx}$ and $s_{yz} = s_{zy}$. Therefore, nine different parameters need to be estimated: three scale factors, three misalignment angles and three biases.

- **Calibration procedure:** Prior to the estimation of the parameters, we need to place the MIMU in at least 9 different static positions. These data are then plugged in the following error function e_k :

$$e_k = a_x^2 + a_y^2 + a_z^2 - g^2 = \sum_{i=x,y,z} \left\{ \sum_{j=x,y,z} [s_{ij}(v_{j,k} - b_j)]^2 \right\} - g^2 \quad (3.61)$$

which is used to form a cumulative quadratic error function E :

$$E = E(b_x, b_y, b_z, s_{xx}, s_{yy}, s_{zz}, s_{xy}, s_{xz}, s_{yz}) = \frac{\sum_{k=1}^N e_k^2}{N} \quad (3.62)$$

which is, in turn, minimized applying Newton's procedure.

3.4.1.9 Method 9: Wang's neural network procedure

Wang et al. present in [132] a scalar calibration method that uses an artificial neural network to correct the diversionary errors in a tri-axial orthogonal magnetometer. When a tri-axial orthogonal magnetometer is used to estimate the intensity of a measured magnetic field, different results will be obtained in different orientations. This phenomenon is called diversionary errors. A calibration model with nine unknown parameters is established and a special neural network structure is devised, which can adaptively update the nine calibration parameters through the relationship between the outputs of the magnetometer and the magnetic field applied. They apply a scalar calibration method, in which only the intensity of the magnetic field is used without having to know its direction.

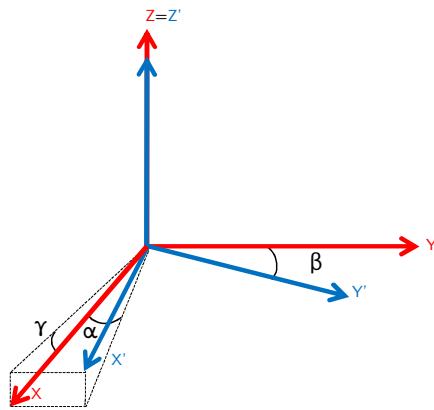


Figure 3.2: Definition of non-orthogonality angles in Wang's algorithm.

- **Sensor modeling:** For a particular magnetic measurement, the magnetometer output $\mathbf{v}_h = [v_x, v_y, v_z]^T$ is connected to the actual magnetic field measured $\mathbf{h} = [h_x, h_y, h_z]^T$ according to

$$\mathbf{v} = f(\mathbf{h}) = \mathbf{S}\mathbf{R} \cdot \mathbf{h} + \mathbf{b} \quad (3.63)$$

where \mathbf{S} is the diagonal matrix of scale factors,

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \quad (3.64)$$

\mathbf{R} is a 3×3 upper triangular matrix, which transforms a vector from the ideal orthogonal coordinate system to the non-orthogonal magnetometer axes coordinate system

$$\mathbf{R} = \begin{bmatrix} \cos \alpha & \sin \gamma & \sin \alpha \cos \gamma \\ 0 & \cos \beta & \sin \beta \\ 0 & 0 & 1 \end{bmatrix} \quad (3.65)$$

where α, β and γ are the three non-orthogonality angles (see figure 3.2). β is the angle between the Y' -axis and Y -axis and α and γ respectively denote the angles between the X' -axis and the two planes ($X'OZ'$ and $Z'OY'$). The vector $\mathbf{b} = [b_x, b_y, b_z]^T$ is the bias vector.

- **Calibration procedure:** Nine parameters need to be determined: three non-orthogonality angles, three scale factors and three output offsets. The magnetic field measured can be estimated by applying:

$$\hat{\mathbf{h}} = f^{-1}(\mathbf{v}) = \mathbf{R}^{-1}\mathbf{S}^{-1}(\mathbf{v} - \mathbf{b}) = \mathbf{R}^{-1}\mathbf{S}^{-1}\mathbf{v} - \mathbf{b}' \quad (3.66)$$

where

$$\mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1/s_z \end{bmatrix} = \begin{bmatrix} s'_x & 0 & 0 \\ 0 & s'_y & 0 \\ 0 & 0 & s'_z \end{bmatrix} \quad (3.67)$$

and the offset vector $\mathbf{b}' = R^{-1}S^{-1}\mathbf{b} = [b'_x, b'_y, b'_z]^T$. The inverse R^{-1} can be derived as:

$$R^{-1} = \begin{bmatrix} \frac{1}{\cos \alpha \cos \gamma} & \frac{-\sin \gamma}{\cos \alpha \cos \beta \cos \gamma} & \frac{\sin \beta \sin \gamma - \sin \alpha \cos \beta \cos \gamma}{\cos \alpha \cos \beta \cos \gamma} \\ 0 & \frac{1}{\cos \beta} & \frac{-\sin \beta}{\cos \beta} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.68)$$

The scalar intensity of the measured geomagnetic field can be calculated as:

$$\|\hat{\mathbf{h}}(t)\| = \sqrt{\hat{\mathbf{h}}^T(t)\hat{\mathbf{h}}(t)} = \sqrt{(R^{-1}S^{-1}\mathbf{v}(t) - \mathbf{b}')^T(R^{-1}S^{-1}\mathbf{v}(t) - \mathbf{b}')} \quad (3.69)$$

To estimate the parameters a neural network is built. Considering that the non-orthogonality angles α, β, γ are generally $< 1^\circ$, the following approximations can be made: $\cos \alpha \approx 1, \cos \beta \approx 1, \cos \gamma \approx 1, \sin \alpha \approx \alpha, \sin \beta \approx \beta, \sin \gamma \approx \gamma, \sin \beta \sin \gamma \approx 0$. Thus, the calibration model can be simplified as follows:

$$\hat{\mathbf{h}} \approx \begin{bmatrix} s'_x & -\gamma & -\alpha \\ 0 & s'_y & -\beta \\ 0 & 0 & s'_z \end{bmatrix} \mathbf{v} + \begin{bmatrix} b'_x \\ b'_y \\ b'_z \end{bmatrix} \quad (3.70)$$

Then, the square error of the training samples is defined as the objective function of the neural network

$$J(\cdot) = \frac{1}{2} \sum_{t=1}^n e(t)^2 = \frac{1}{2} \sum_{t=1}^n (\|\hat{\mathbf{h}}(t)\|^2 - M^2)^2 \quad (3.71)$$

where M is the magnitude of the local Earth's magnetic field vector. If the weight vector is defined as $\omega = [\alpha, \beta, \gamma, s'_x, s'_y, s'_z, b'_x, b'_y, b'_z]^T$, then according to the correlation principle, the updating values $\Delta\omega$ in the t^{th} step can be expressed as

$$\Delta\omega = -a \cdot e(t) \frac{\partial e(t)}{\partial \omega} \quad (3.72)$$

where $a > 0$ is the learning rate that controls the stability and rate of convergence (generally between 0.01 and 0.1). We can estimate the partial derivatives of errors $e(t)$ as follows:

$$\begin{aligned} \frac{\partial e(t)}{\partial b'_x} &= 2\hat{h}_x(t), \quad \frac{\partial e(t)}{\partial b'_y} = 2\hat{h}_y(t), \quad \frac{\partial e(t)}{\partial b'_z} = 2\hat{h}_z(t) \\ \frac{\partial e(t)}{\partial s'_x} &= 2\hat{h}_x(t)v_x(t), \quad \frac{\partial e(t)}{\partial s'_y} = 2\hat{h}_y(t)v_y(t), \quad \frac{\partial e(t)}{\partial s'_z} = 2\hat{h}_z(t)v_z(t) \\ \frac{\partial e(t)}{\partial \alpha} &= -2\hat{h}_x(t)v_z(t), \quad \frac{\partial e(t)}{\partial \beta} = -2\hat{h}_y(t)v_z(t), \quad \frac{\partial e(t)}{\partial \gamma} = -2\hat{h}_x(t)v_y(t) \end{aligned} \quad (3.73)$$

Summarizing, each step of the algorithm is divided in the following two steps

1. Computation of the estimation error:

$$e(t) = \|\hat{\mathbf{h}}(t)\|^2 - M^2 \quad (3.74)$$

2. Computation of the parameter update weights:

$$\begin{aligned} \Delta b'_x &= -2ae(t)\hat{h}_x(t), \Delta b'_y = -2ae(t)\hat{h}_y(t), \Delta b'_z = -2ae(t)\hat{h}_z(t) \\ \Delta s'_x &= -2ae(t)\hat{h}_x(t)v_x(t), \Delta s'_y = -2ae(t)\hat{h}_y(t)v_y(t), \Delta s'_z = -2ae(t)\hat{h}_z(t)v_z(t) \\ \Delta \alpha &= 2ae(t)\hat{h}_x(t)v_z(t), \Delta \beta = 2ae(t)\hat{h}_y(t)v_z(t), \Delta \gamma = 2ae(t)\hat{h}_x(t)v_y(t) \end{aligned} \quad (3.75)$$

The initial values of the parameters are set to $b'_x = b'_y = b'_z = 0, \alpha = \beta = \gamma = 0, s'_x = s'_y = s'_z = 1$. When the decreasing quantity of the objective function $J(\cdot)$ is less than a predefined threshold ε , the iterations are stopped as the neural network is considered to have reached convergence.

The data which is needed to carry out the calibration procedure is gathered by putting the MIMU in 20 random spatial attitudes about the test point, where the intensity of the magnetic field is assumed to have an uniform and time-constant value.

3.4.1.10 Method 10: Skog's multi-position procedure

Skog and Händel propose in [45] a low cost calibration method which is also based in the minimization of the difference between the estimated magnitude and the actual magnitude of the acceleration (or the Earth's magnetic field) and the angular rate. In our case, we will only use this algorithm to calibrate the accelerometer and the magnetometer as we do not have a device providing a known rotation rate. The minimization is carried out applying Newton's algorithm.

- **Sensor modeling:** The specific forced measured by the accelerometer (\mathbf{v}^a) can be expressed with respect to the platform reference axes as follows:

$$\mathbf{v}^p = T_a^p \mathbf{v}^a, \quad T_a^p = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{bmatrix} \quad (3.76)$$

where \mathbf{v}^p and \mathbf{v}^a are the specific forces measured in the platform axes and the accelerometer axes respectively. Hence, α_{ij} is the rotation of the i^{th} sensitivity axis of the accelerometer with respect to the j^{th} platform axis (see 3.3). Matrix T_a^p can be simplified if axes x^a and x^p are made coincident so axis y^p is contained in the plane defined by axes x^a and y^a yielding:

$$T_a^p = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ 0 & 1 & -\alpha_{zx} \\ -0 & 0 & 1 \end{bmatrix} \quad (3.77)$$

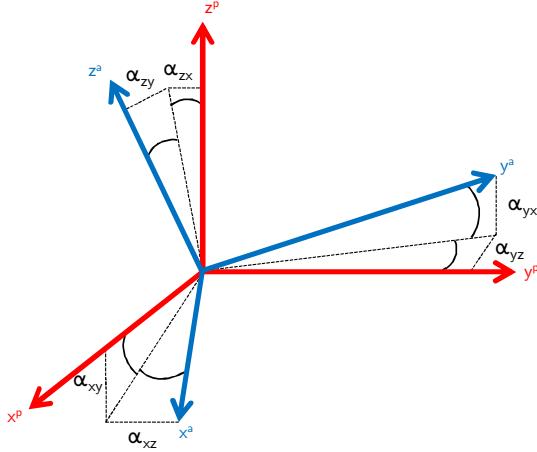


Figure 3.3: Definition of the non-orthogonality angles. The non-orthogonal axes of the accelerometer triad can be aligned with the orthogonal platform axes x^p, y^p, z^p through the six angles $\alpha_{xy}, \alpha_{xz}, \alpha_{zy}, \alpha_{yz}, \alpha_{zx}, \alpha_{zy}$

Therefore, the output of the accelerometer is expressed as follows:

$$\tilde{\mathbf{v}}^a = K_a(T_a^p)^{-1} \mathbf{v}^p + \mathbf{b}_a + \epsilon_a \quad (3.78)$$

where K_a is the scale factor matrix, \mathbf{b}_a is the bias vector and ϵ_a is the noise vector.

- **Calibration procedure:** Nine different parameters—three scale factors, three non-orthogonality angles and three biases—have to be found to calibrate the accelerometer. The following estimator is chosen by the authors:

$$\mathbf{u}_k = \mathbf{h}(\mathbf{v}_k, \theta) = T K^{-1} (\mathbf{v}_k - \mathbf{b}) \quad (3.79)$$

where the parameter vector is defined as,

$$\theta = [k_x \ k_y \ k_z \ \alpha_{yz} \ \alpha_{zy} \ \alpha_{zx} \ b_x \ b_y \ b_z \ \sigma^2]^T \quad (3.80)$$

Ideally, irrespective to the orientation of the MIMU, the measured magnitude (Earth's magnetic field or gravity) has to be equal to the magnitude of its local value. Then, we can build the cost function as the quadratic error between the squared magnitude of the input \mathbf{u}_k and the squared magnitude of the output from the compensated MIMU output:

$$\hat{\theta} = \arg \min_{\theta} \{L(\theta)\} \quad (3.81)$$

$$L(\theta) = \sum_{k=0}^{K-1} (\|\mathbf{u}_k\|^2 - \|\mathbf{h}(\mathbf{v}_k, \theta)\|^2)^2 \quad (3.82)$$

being $K=MN$, where M is the number of orientations the MIMU has been placed in and N is the number of gathered samples in each orientation. In this case, M has to be equal or greater than nine. Knowing that during each static orientation, the magnitude of the measured acceleration has to be constant and equal to the gravity (or the Earth's magnetic field), the cost function can be simplified as:

$$L(\theta) = \sum_{m=0}^{M-1} (\|\bar{\mathbf{u}}^m\|^2 - \|\mathbf{h}(\bar{\mathbf{v}}^m, \theta)\|^2)^2 \quad (3.83)$$

where \mathbf{u}^m and \mathbf{v}^m are the magnitude of the local value of the acceleration or the Earth's magnetic field and the magnitude of the output of the accelerometer or the magnetometer.

The last step remaining is to minimize the optimal values of the calibration parameters that minimize the cost function. This is done applying Newton's algorithm.

3.4.1.11 Method 11: Gebre-Egziabher's two-step nonlinear procedure

Gebre-Egziabher et al. develop in [127] an algorithm to calibrate magnetometers in the magnetic field domain. The calibration algorithm uses an iterated, batch least-squares estimator that is initialized using a two-step nonlinear estimator.

- **Sensor modeling:** This work presents a very complete model for the output error of a strapdown magnetometer.

$$\hat{\mathbf{h}}^w = C^{b \rightarrow w} [C_m C_{sf} C_{si} (h^b + b^b + w^b)] \quad (3.84)$$

where $\hat{\mathbf{h}}^w$ is the measured field measurement in the w plane while $\mathbf{b}^b = [b_x^b, b_y^b, b_z^b]^T$ and $\mathbf{w}^b = [w_x^b, w_y^b, w_z^b]^T$ represent hard iron biases and wide-band noise, respectively, and are additive errors that corrupt the true field measurement \mathbf{h}^b . The variables C_{si} , C_m and C_{sf} are 3×3 matrices that model soft iron, scale factor, and misalignment errors, respectively. The matrix $C^{b \rightarrow w}$ is the 3×3 body frame to navigation frame rotation matrix.

$$C_{si} = \begin{bmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{bmatrix}; C_{sf} = \begin{bmatrix} (1+s_x) & 0 & 0 \\ 0 & (1+s_y) & 0 \\ 0 & 0 & (1+s_z) \end{bmatrix}$$

$$C_m = \begin{bmatrix} 1 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 1 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 1 \end{bmatrix} \quad (3.85)$$

where s_x, s_y and s_z are the scale factors, α_{ij} represents the constants of proportionality between the magnetic field which is applied to a soft iron material and the resulting induced magnetic field. The index i indicates the component of the field along axis i which has been generated in response to an applied field in the direction indicated by j . The α_{ij} terms represent the effective soft iron coefficients and are the constants of proportionality between the magnetic field applied to a soft iron and the resulting induced magnetic field. The three parameters defining the C_m matrix represent small rotations about the body axes of the body that will bring the platform axes into perfect alignment with the body axes. Thus, C_m is constant and only needs to be estimated once.

- **Calibration procedure:** The calibration algorithm aims to solve the problem of determining the parameters of an ellipsoid that best fits the data collected from a magnetometer triad. Mathematically, the locus of measurements is described by the following equation:

$$\|\mathbf{h}\|^2 = h^2 = \left(\frac{\hat{h}_x^b - b_x}{\gamma_x} \right)^2 + \left(\frac{\hat{h}_y^b - b_y}{\gamma_y} \right)^2 + \left(\frac{\hat{h}_z^b - b_z}{\gamma_z} \right)^2 \quad (3.86)$$

where

$$\begin{aligned} \gamma_x &= (1 + s_{fx})(1 + \alpha_{xx}) \\ \gamma_y &= (1 + s_{fy})(1 + \alpha_{yy}) \\ \gamma_z &= (1 + s_{fz})(1 + \alpha_{zz}) \end{aligned} \quad (3.87)$$

The parameters to be estimated are the hard iron biases denoted by b_x, b_y and b_z and the combined effect of scale-factor error and the α_{ii} soft iron terms denoted by γ_x, γ_y and γ_z . The given or known inputs to the calibration algorithm are the measured magnetometer outputs, \hat{h}_x^b, \hat{h}_y^b and \hat{h}_z^b and the magnitude of Earth's magnetic field vector, $\|\mathbf{h}\| = h$, in the geographic area where the calibration is being performed.

The ellipsoid of revolution is fitted to the measured magnetic field data by using a two-step batch least-squares estimator which is explained in depth in the original paper.

3.4.1.12 Method 12. Fong's 18-positions procedure

Fong et al. [133] present a method to calibrate and compensate for non-zero biases, non-unit scale factors, axis misalignments and cross-axis sensitivities for MEMS triaxial accelerometers. Like in the majority of the works we have reviewed up to this point, this method is also based on the fact that the magnitude of the measured acceleration in static positions must equal that of the gravity. A cost function

defined by the difference between the magnitude of the local gravity and the magnitude of the measured acceleration is minimized employing the Downhill simplex optimization method [134].

- **Sensor modeling:** The model is very similar to that presented by Skog and Händel (method 10). The accelerometer model which converts the k^{th} acceleration vector measured in the sensor frame, $\mathbf{a}_{s,k}$, to the platform frame $\mathbf{a}_{p,k}$, is

$$\mathbf{a}_{s,k} = \mathbf{MS}(\mathbf{a}_{s,k} - \mathbf{b}_a) \quad (3.88)$$

where \mathbf{b}_a is the bias vector, \mathbf{M} is the misalignment matrix,

$$\mathbf{M} = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ 0 & 1 & -\alpha_{zx} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.89)$$

and \mathbf{S} is the scale factor and cross-axis sensitivities matrix,

$$\mathbf{S} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \quad (3.90)$$

where s_{ij} is the sensitivity of the i th axis to the accelerations along the j th axis. Now, \mathbf{M} and \mathbf{S} matrices are multiplied to get

$$\mathbf{E} = \begin{bmatrix} s_{xx} - s_{yx}\alpha_{yz} + s_{zx}\alpha_{zy} & s_{xy} - s_{yy}\alpha_{yz} + s_{zy}\alpha_{zy} & s_{xz} + s_{yz}\alpha_{yz} + s_{zz}\alpha_{zy} \\ s_{yx} - s_{zx}\alpha_{zx} & s_{yy} - s_{zy}\alpha_{zx} & s_{yz} - s_{zz}\alpha_{zx} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \quad (3.91)$$

The off-diagonal elements which are close to zero can be neglected, yielding

$$\mathbf{E} \approx \begin{bmatrix} s_{xx} & s_{xy} - s_{yy}\alpha_{yz} & s_{xz} + s_{zz}\alpha_{zy} \\ s_{yx} & s_{yy} & s_{yz} - s_{zz}\alpha_{zx} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} = \begin{bmatrix} e_{00} & e_{01} & e_{02} \\ e_{10} & e_{11} & e_{12} \\ e_{20} & e_{21} & e_{22} \end{bmatrix} \quad (3.92)$$

Therefore, the final expression of the model proposed by the authors is as follows:

$$\mathbf{a}_{p,k} = \mathbf{E}(\mathbf{a}_{s,k} - \mathbf{b}_a) \quad (3.93)$$

where $\theta_a = [e_{00} \ e_{01} \ e_{02} \ e_{10} \ e_{20} \ e_{21} \ e_{22} \ b_x \ b_y \ b_z]$ are the twelve parameters that need to be found to calibrate the accelerometer.

- **Calibration procedure:** The cost function is also very similar to the one proposed by Skog and Händel. It measures the amount of deviation from the actual magnitude of the gravity (1 g) for K sets of measurements.

$$L(\theta_a) = \sum_{k=0}^{K-1} \left(1 - \|h(\mathbf{a}_{s,k}, \theta_a)\|^2 \right)^2 \quad (3.94)$$

The necessary data is gathered by placing the MIMU in 18 different positions (resting on its 6 flat faces and 12 edges) and averaging the accelerometer output over a period of 1 s to reduce noise. Moreover, the Filtered-rectifier motion intensity detector, proposed in [135] and explained in section 4.3.1.5, is employed to ensure that the MIMU is not subjected to motion.

Finally, the Downhill simplex [134] optimization method is used to minimize the cost function and estimate the calibration parameters.

3.4.2 Structure of the study

The comparative study of all the aforementioned methods is divided in two parts: first, we will study their application to calibrate the triaxial accelerometer which is included in Wagyromag, and then, we will also apply them to calibrate its triaxial magnetometer.

The accelerometer calibration is, in turn, subdivided in two parts; we will start by placing the device in six static positions so that each one of the axis is set parallel and antiparallel to the direction of the gravity vector (see figure 3.4). The mode of the measured acceleration values in each position will be the input of the algorithms (see figure 3.5). Once the calibration parameters are estimated, we will apply them to the same six positions signal to calibrate it (see figure 3.6). Next, we will compute the error of the calibrated signal with respect to the reference, which, in our case, is the local gravity vector. To do so, we will subtract the mode of the calibrated acceleration to the reference acceleration at each one of the six positions. For example, if we place Z axis parallel to the gravity vector, the reference acceleration values will be $a_{x\text{-ref}} = 0 \text{ g}$, $a_{y\text{-ref}} = 0 \text{ g}$ and $a_{z\text{-ref}} = 1 \text{ g}$, i.e. the accelerometer should ideally measure no acceleration in both X and Y axes and 1 g in the Z axis. Therefore, subtracting the calibrated signal to the reference and computing the absolute value of the result, we will obtain the acceleration error for each one of the six positions after the calibration procedure.

Once we have computed the 18 acceleration errors—6 positions×3 axes—the whole procedure (placing the Wagyromag in six positions, gathering the acceleration signals, estimating the calibration parameters, calibrating the signals and computing the errors) is repeated four more times to average the resulting errors.

Since most of the calibration algorithms included in this study are thought to compute the calibration parameters using a larger set of static positions, in the second part of the study we will test those allowing multi-point estimation (algorithms 2-6 and 8-11) using a set of values gathered from 73 different random quasi-static positions covering all the space (see figure 3.7). Again, the mode of each quasi-static period is computed for each axis and used as the input for the algorithms. Such multi-point dataset, should ideally define a sphere of radius 1 g and centered in the origin after the calibration procedure. Figures 3.8 and 3.9 depict the three

dimensional representation of the multi-point acceleration data set before and after calibration, respectively.

After the estimation of the calibration parameters, we will also calibrate the same six positions signals which were used in the first part and compute again the average acceleration error. This way we will be able to check if the precision of the algorithms has increased by using a larger input dataset.

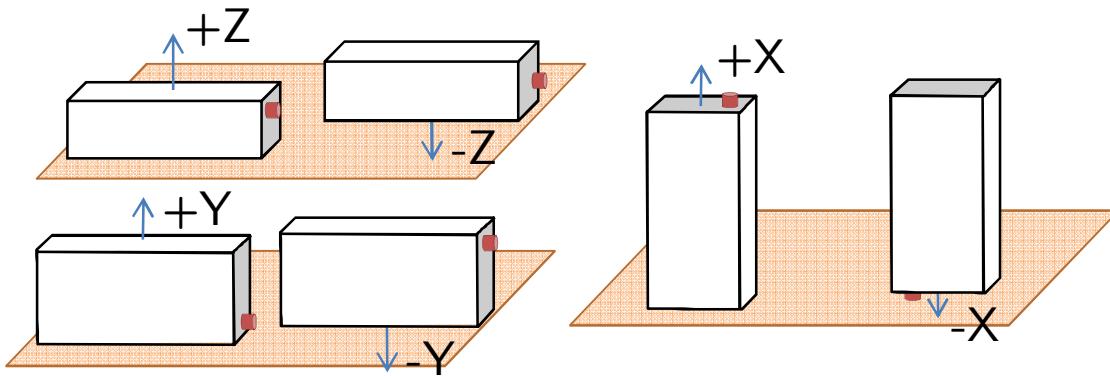


Figure 3.4: Wagyromag placed with each axis parallel and antiparallel to gravity vector.

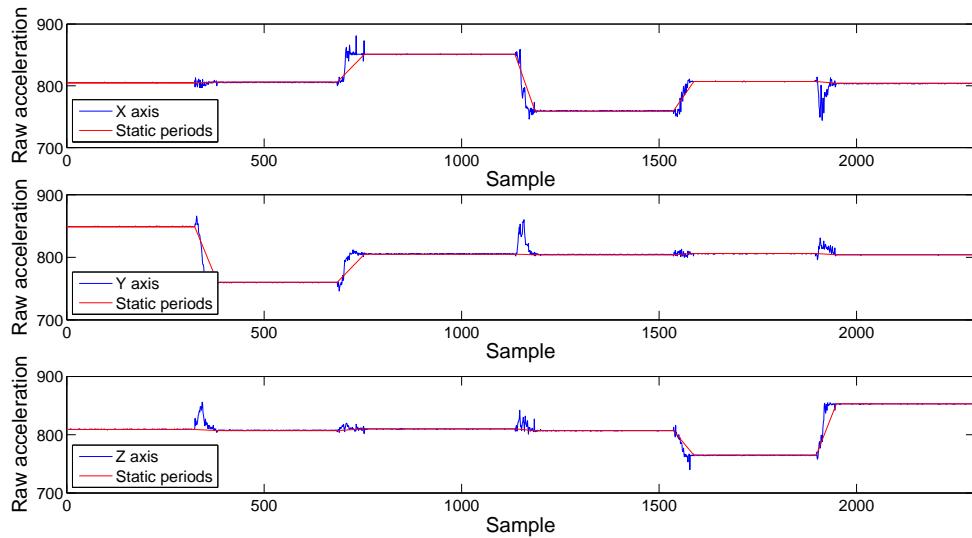


Figure 3.5: Raw acceleration signals gathered at each one of the six positions. The mode of the raw acceleration is computed automatically for each position.

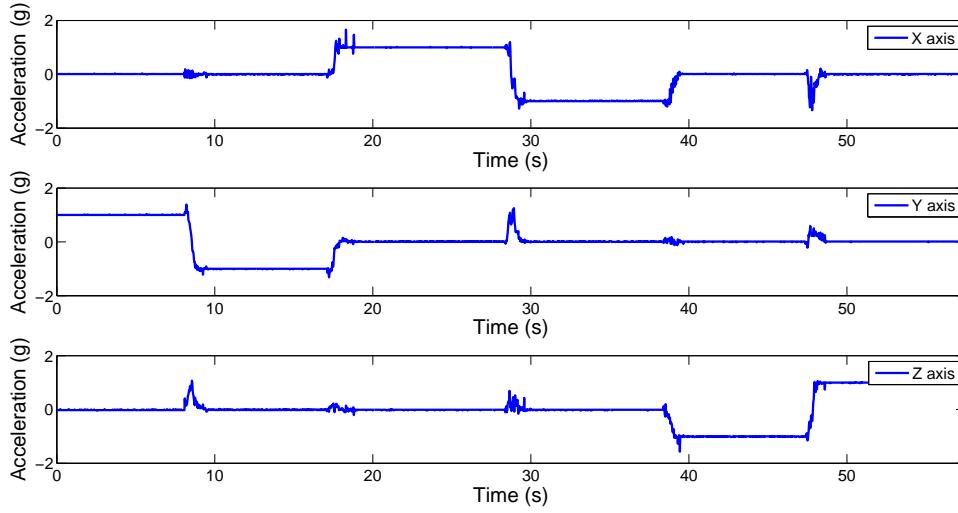


Figure 3.6: Calibrated acceleration signals gathered at each one of the six positions. The mode of the raw acceleration is computed automatically for each position.

The second part of the general calibration comparative study consists of applying algorithms 2-6 and 8-11 to calibrate Wagyromag's triaxial magnetometer. The input of these algorithms is a dataset which is gathered by randomly moving the device around its axes trying to span all the space. Unlike with the accelerometer, we do not need to place the Wagyromag in quasi-static or static positions since the magnetic field is not affected by linear acceleration. Hence, the calibration maneuvers are simpler, much faster and a way less cumbersome. This, in turn, allows the obtention of a larger number of data points which improves the precision of the estimated calibration parameters [136].

Ideally, the gathered dataset should define a sphere centered in the origin of radius equal to the local value of Earth's magnetic field magnitude. However, the undesired effects explained at the beginning of this chapter distort the sphere into an off-centered ellipsoid.

We will gather five sets having a large number of raw data points which will be used to estimate the calibration parameters that fit the ellipsoid back into the ideal sphere. These parameters are subsequently used to calibrate the raw datasets and, in addition, the magnitude of the resultant calibrated Earth's magnetic field is also computed. Our goal is to calculate the error between the estimated magnitude with respect to the known local magnitude. Therefore, we subtract every sample of the estimated magnitude to the value of the local magnitude (0.432 Gauss in Granada, Spain [137]). This procedure is carried out for the five datasets to find the average value of the magnitude error.

All the computed errors are used as indicators to compare the performance and precision of the algorithms for both the accelerometer and magnetometer.

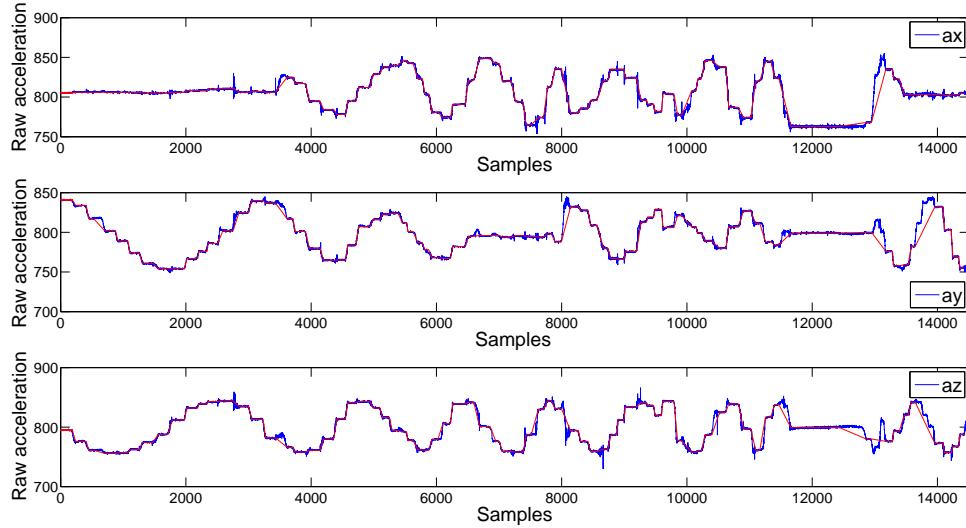


Figure 3.7: Raw acceleration signals gathered from placing the Wagyromag in 73 different quasi-static positions.

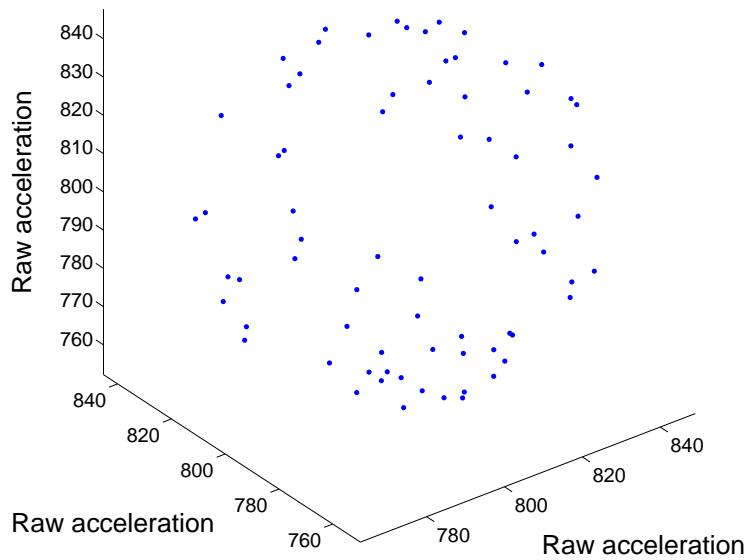


Figure 3.8: Mode of the raw acceleration gathered at each of the quasi-static positions for each axis. 3D representation.

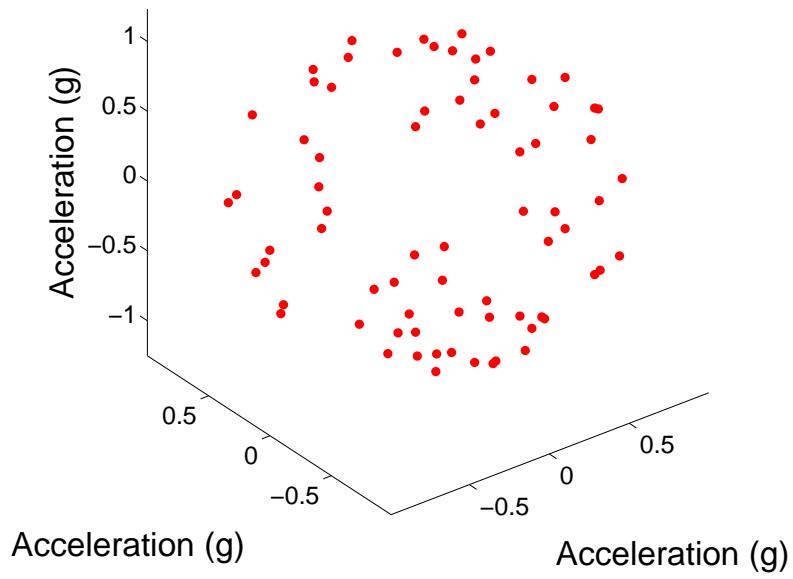


Figure 3.9: Mode of the calibrated acceleration gathered at each of the quasi-static positions for each axis. 3D representation .

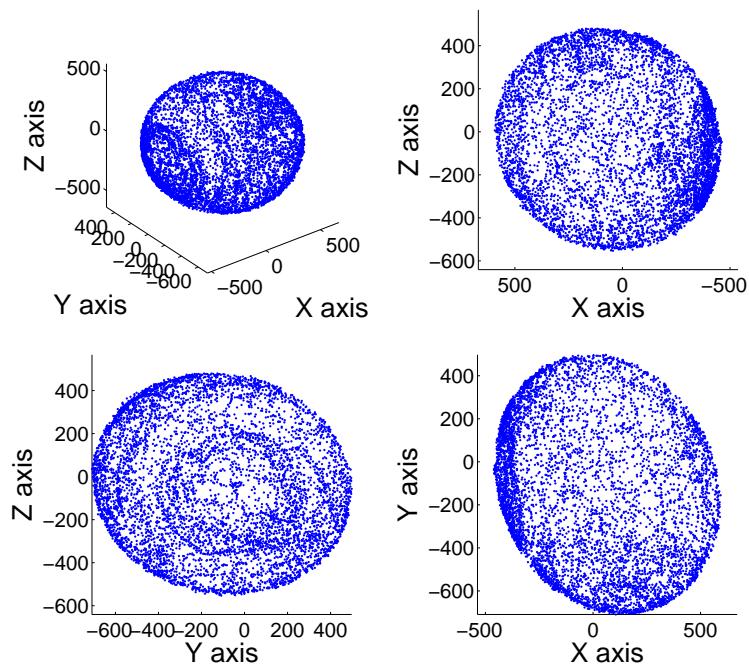


Figure 3.10: Raw magnetic field values gathered from placing moving the Wagymag around its axes in a random way. 3D representation and projection over the planes.

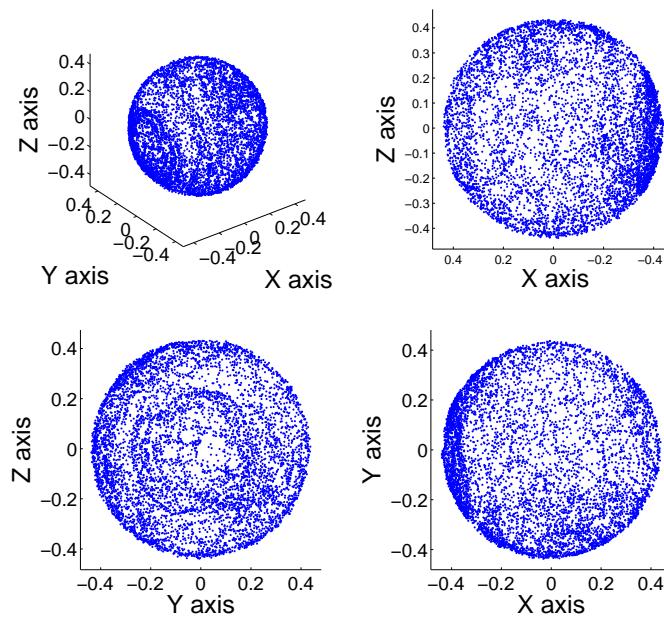


Figure 3.11: Calibrated magnetic field values gathered from placing moving the Wagyromag around its axes in a random way. 3D representation and projection over the planes.

3.4.3 Results

The results of the comparison between algorithms are presented following the structure below:

- *Accelerometer:* The acceleration errors are denoted as $a_{ij\pm}$, where i indicates the axis which is set in the direction of the gravity vector, j indicates the axis measuring the acceleration and \pm indicates the sense of the axis i (+ denotes positive sense and - negative sense).
 - Calibration parameters computed using six-positions dataset:
 - * Average calibration errors for five datasets are depicted included in tables 3.1-3.3 and also depicted in figures 3.12-3.23.
 - * Average calibration total errors (mean of all the 18 $a_{ij\pm}$ errors) for five datasets are shown in Table 3.4.
 - * All the bar plots (figures 3.12-3.23) are put together in figure 3.24 to provide a quick visual way to compare the errors.
 - Calibration parameters computed using multi-positions dataset:
 - * Average calibration errors for five datasets are depicted in figures 3.25-3.33.

- * Average calibration errors for five datasets are also included in tables 3.5-3.6.
- * Average calibration total errors (mean of all the $18 a_{ij\pm}$ errors) for five datasets are shown in Table 3.7.
- * All the bar plots (figures 3.25-3.33) are put together in figure 3.34 to provide a quick visual way to compare the errors.
- *Magnetometer:*
 - Average magnetic field magnitude errors after calibration for the five multi-position runs are included in table 3.8 and also depicted in figure 3.35.

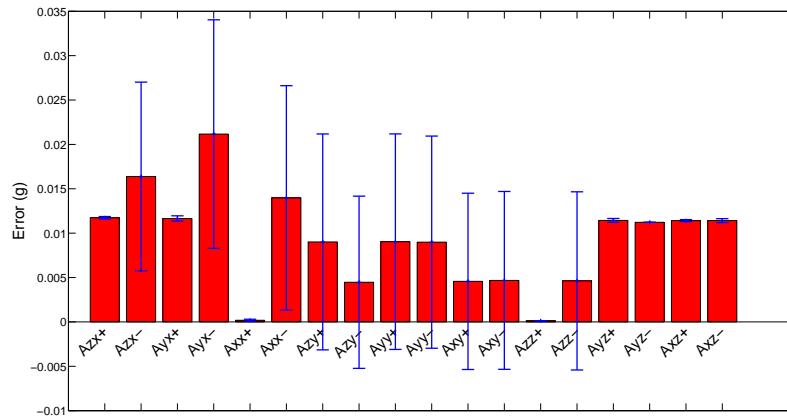


Figure 3.12: Average calibration errors for 5 six positions runs using a fix reference. Method 1.

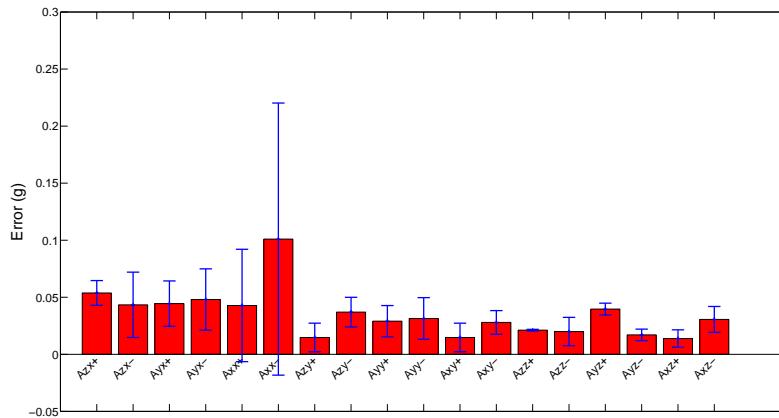


Figure 3.13: Average calibration errors for 5 six positions runs using a fix reference. Method 2.

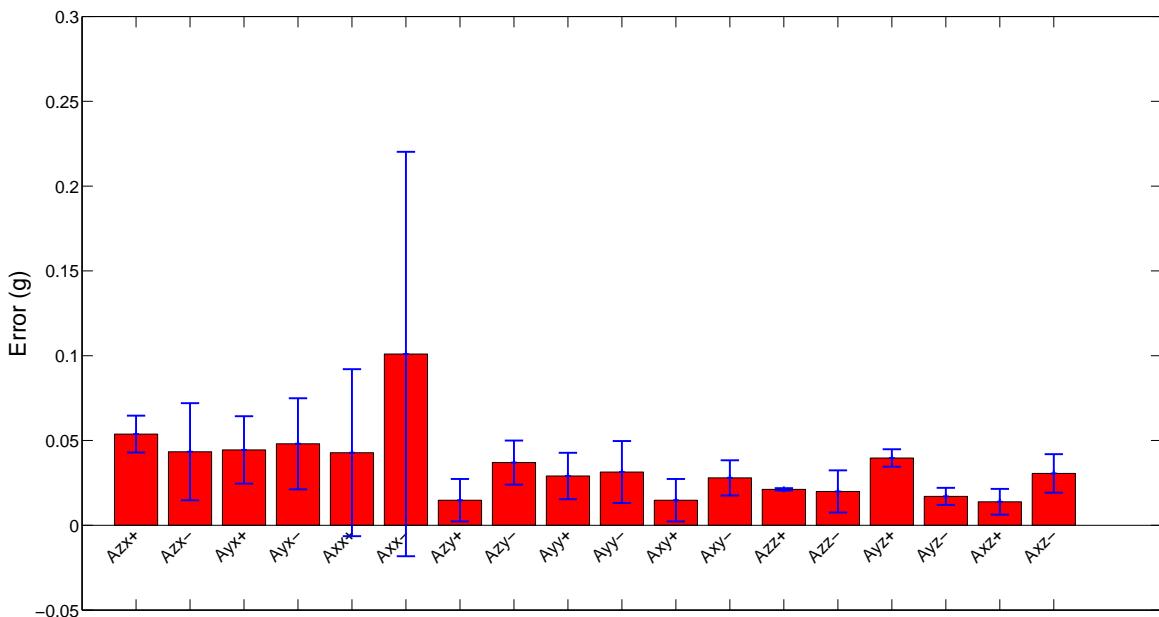


Figure 3.14: Average calibration errors for 5 six positions runs using a fix reference. Method 3.

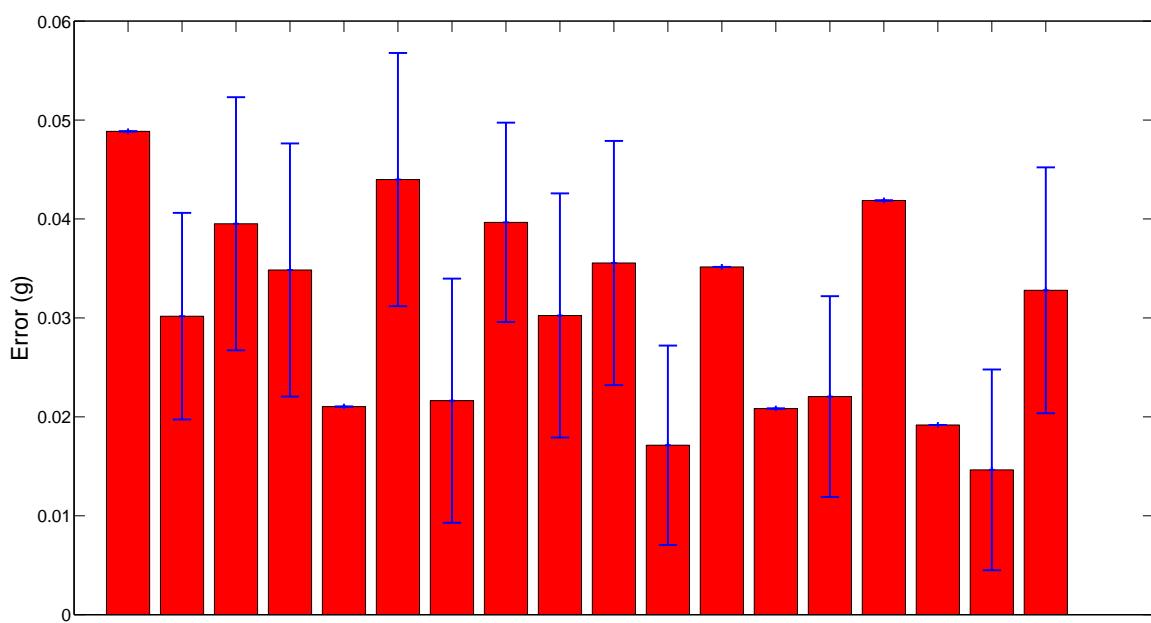


Figure 3.15: Average calibration errors for 5 six positions runs using a fix reference. Method 4.

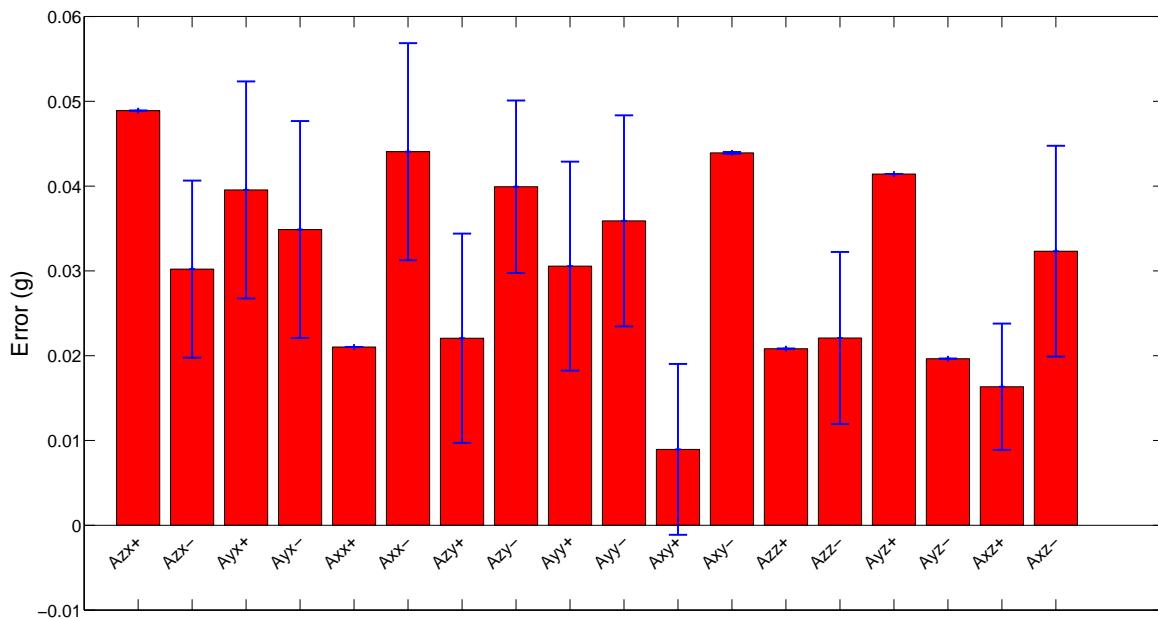


Figure 3.16: Average calibration errors for 5 six positions runs using a fix reference. Method 5.

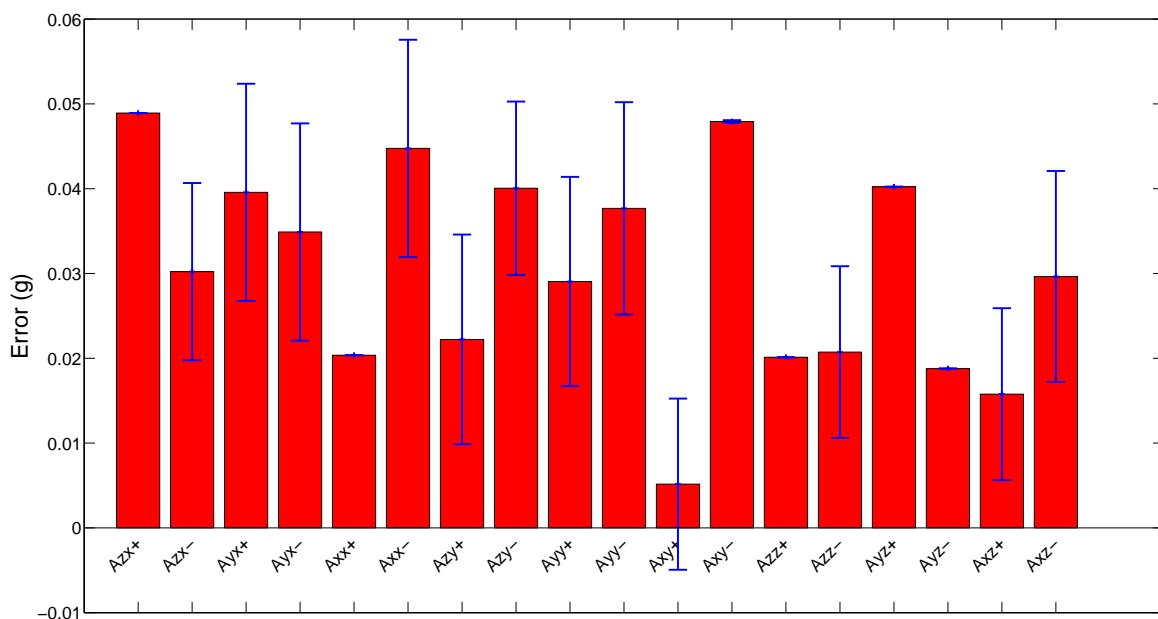


Figure 3.17: Average calibration errors for 5 six positions runs using a fix reference. Method 6.

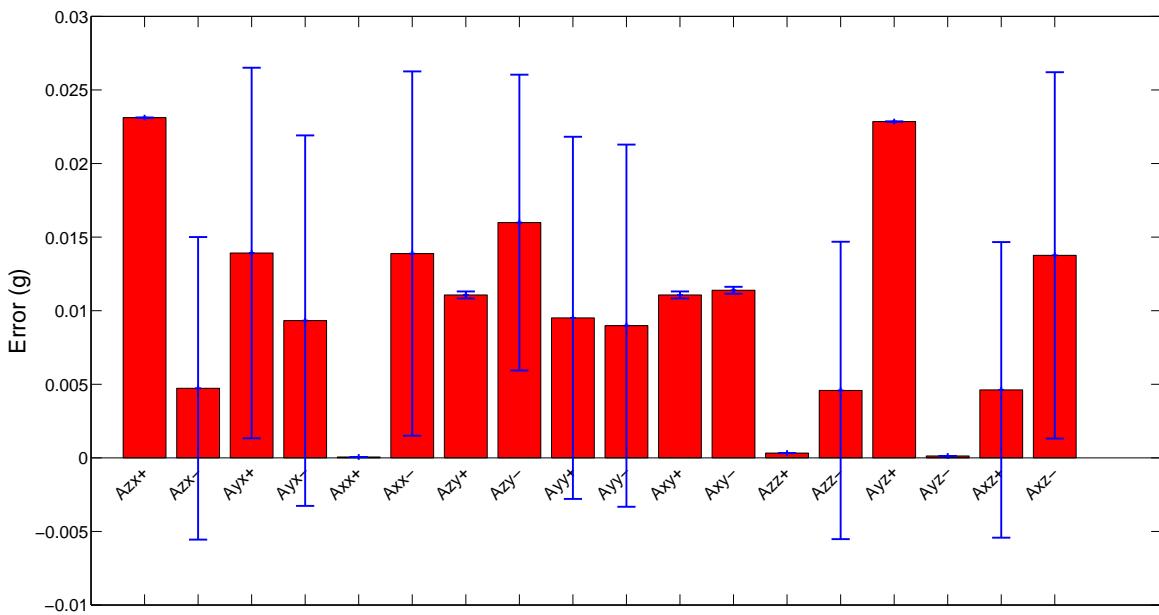


Figure 3.18: Average calibration errors for 5 six positions runs using a fix reference. Method 7.

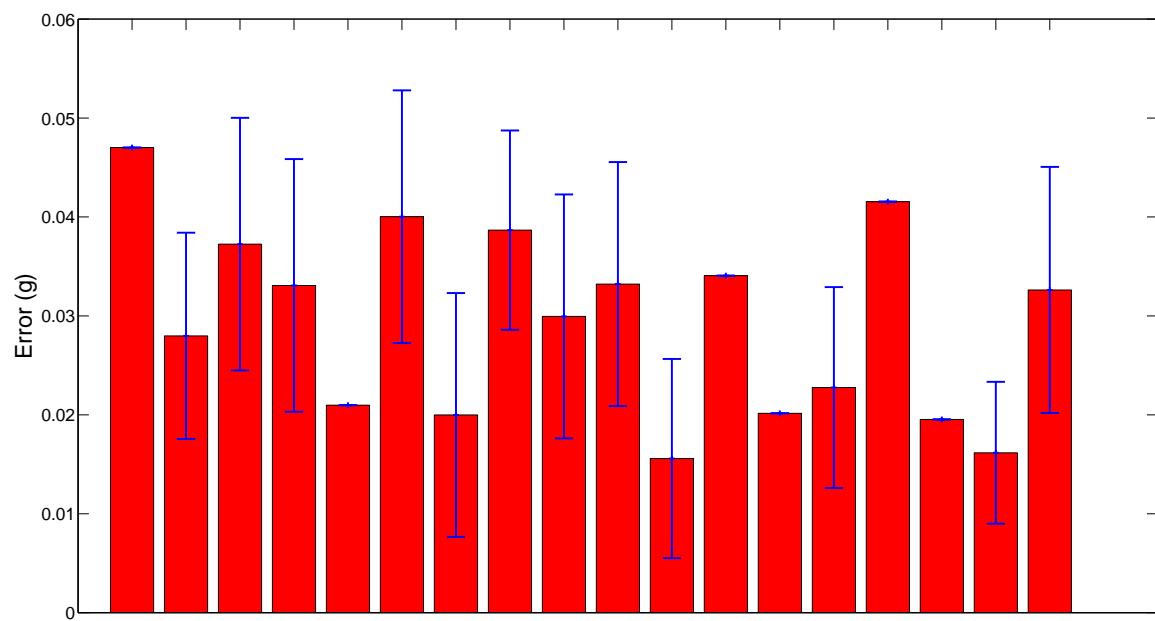


Figure 3.19: Average calibration errors for 5 six positions runs using a fix reference. Method 8.

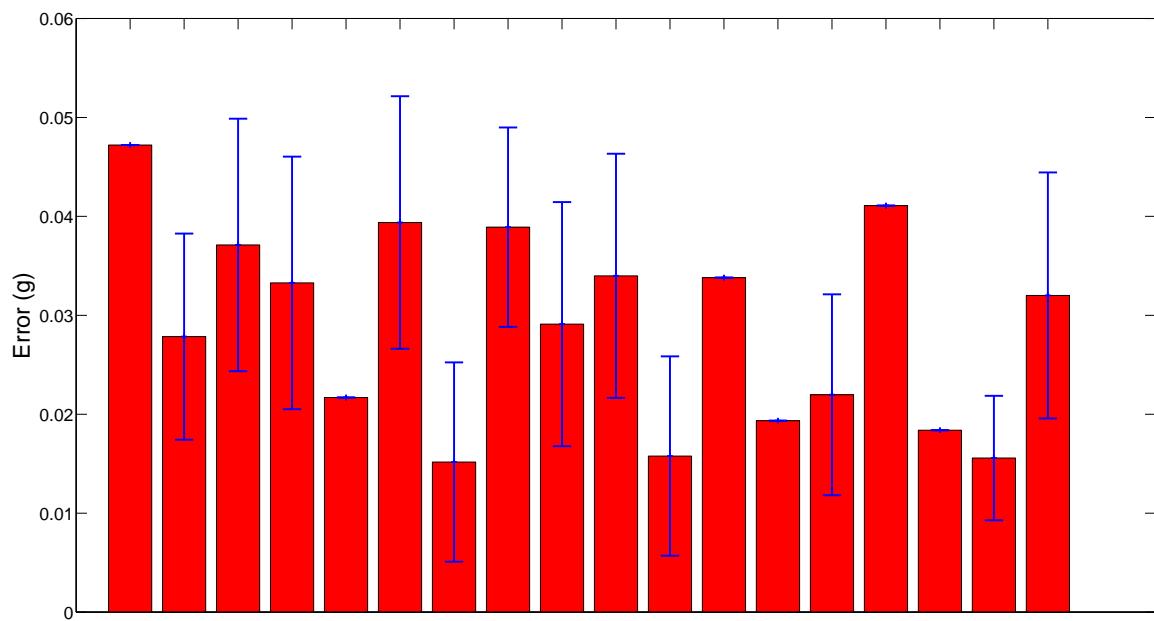


Figure 3.20: Average calibration errors for 5 six positions runs using a fix reference. Method 9.

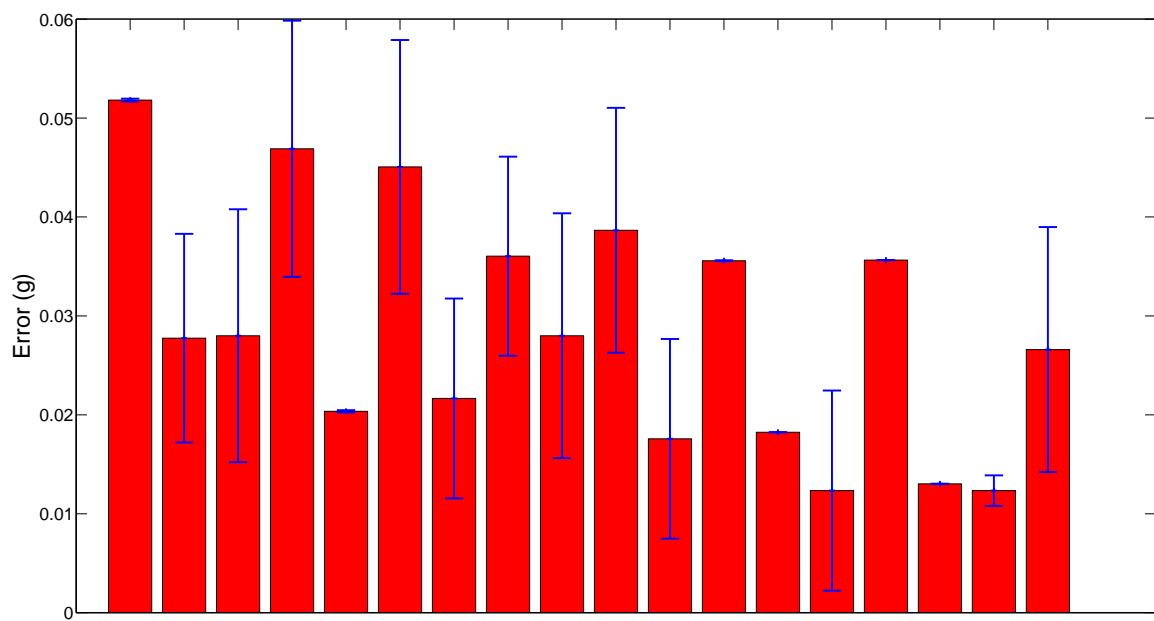


Figure 3.21: Average calibration errors for 5 six positions runs using a fix reference. Method 10.

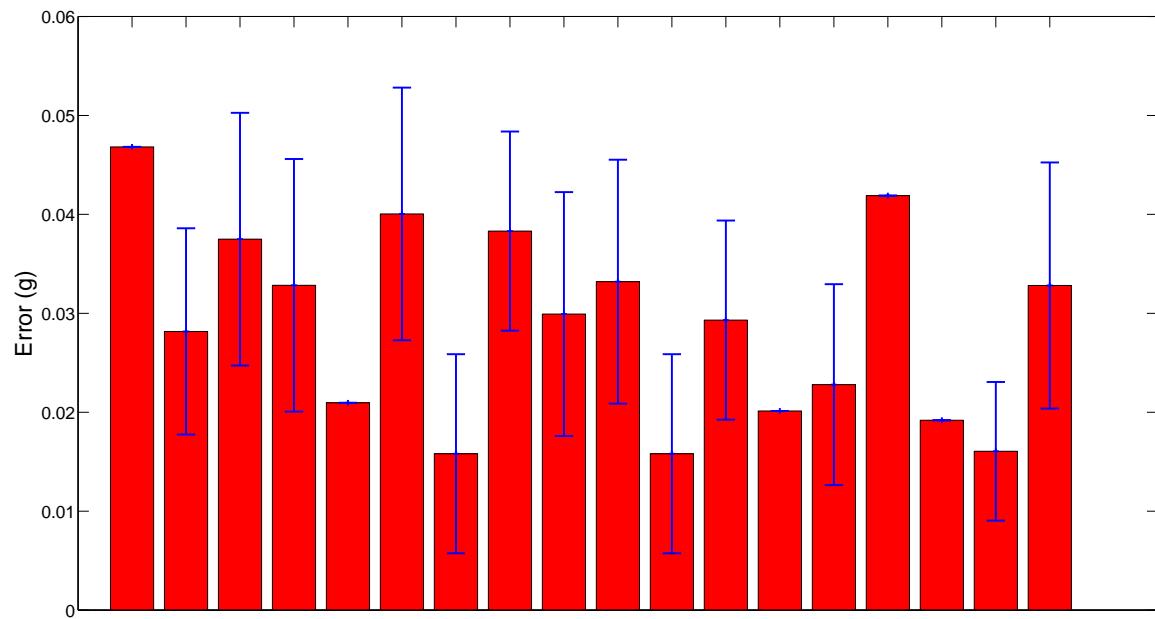


Figure 3.22: Average calibration errors for 5 six positions runs using a fix reference. Method 11.

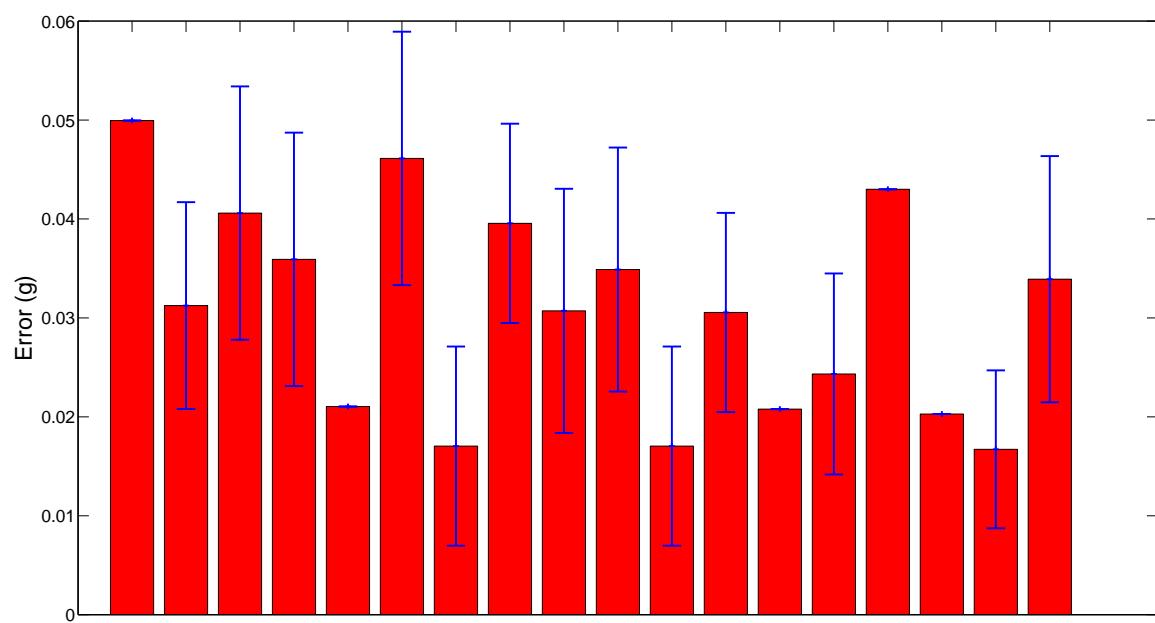


Figure 3.23: Average calibration errors for 5 six positions runs using a fix reference. Method 12.

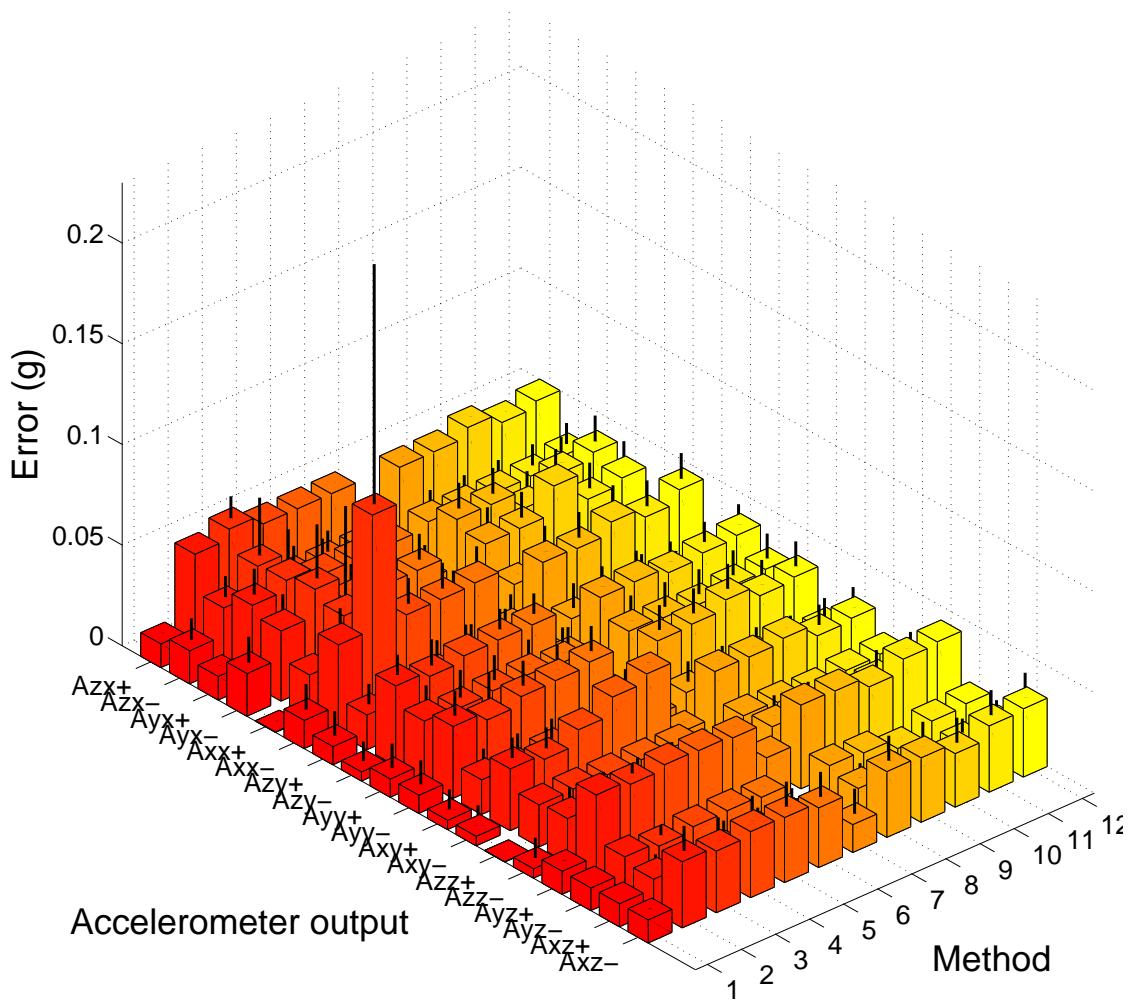


Figure 3.24: Average calibration errors for 5 six positions runs using a fix reference. All methods.

Table 3.1: Average errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using only 6 positions. Methods 1 to 4.

	Method 1	Method 2	Method 3	Method 4
a_{zx+}	$0.0117 \pm 1.14469e-4$	0.0488 ± 0.0000	0.0538 ± 0.0108	0.0489 ± 0.0000
a_{zx-}	0.0164 ± 0.0106	0.0301 ± 0.0301	0.0434 ± 0.0286	0.0302 ± 0.0104
a_{yx+}	$0.0117 \pm 2.9977e-4$	0.0395 ± 0.0127	0.0444 ± 0.0198	0.0395 ± 0.0127
a_{yx-}	0.0212 ± 0.0128	0.0348 ± 0.0127	0.0481 ± 0.0268	0.0348 ± 0.0127
a_{xx+}	$1.8300e-4 \pm 1.1814e-4$	0.0210 ± 0.0000	0.0428 ± 0.0492	0.0210 ± 0.0000
a_{xx-}	0.0140 ± 0.0126	0.0440 ± 0.0127	0.1010 ± 0.1192	0.0440 ± 0.0127
a_{zy+}	0.0090 ± 0.0121	0.0172 ± 0.0100	0.0148 ± 0.0125	0.0216 ± 0.0123
a_{zy-}	0.0044 ± 0.0099	0.0397 ± 0.0100	0.0370 ± 0.0101	0.0397 ± 0.0100
a_{yy+}	0.0090 ± 0.0121	0.0303 ± 0.0123	0.0291 ± 0.0136	0.0302 ± 0.0123
a_{yy-}	0.0089 ± 0.0119	0.0356 ± 0.0123	0.0314 ± 0.0182	0.0355 ± 0.0123
a_{xy+}	0.0045 ± 0.0099	0.0172 ± 0.0100	0.0148 ± 0.0125	0.0171 ± 0.0100
a_{xy-}	0.0046 ± 0.0100	0.0307 ± 0.0100	0.0208 ± 0.0103	0.0351 ± 0.0000
a_{zz+}	$1.2800e-4 \pm 8.1287e-6$	0.0208 ± 0.0000	$0.0212 \pm 7.1173e-4$	0.0208 ± 0.0000
a_{zz-}	0.0046 ± 0.0100	0.0221 ± 0.0101	0.0200 ± 0.0124	0.0221 ± 0.0101
a_{yz+}	$0.0114 \pm 2.1736e-4$	0.0419 ± 0.0000	0.0397 ± 0.0050	0.0419 ± 0.0000
a_{yz-}	$1.1200e-4 \pm 8.1287e-6$	0.0192 ± 0.0000	0.0170 ± 0.0050	0.0192 ± 0.0000
a_{xz+}	$0.0114 \pm 1.1283e-4$	0.0146 ± 0.0101	0.0139 ± 0.0076	0.0146 ± 0.0101
a_{xz-}	$0.0114 \pm 2.2350e-4$	0.0328 ± 0.0124	0.0306 ± 0.0113	0.0328 ± 0.0124

Table 3.2: Average errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using only 6 positions. Methods 5 to 8.

	Method 5	Method 6	Method 7	Method 8
a_{zx+}	0.0489±0.0000	0.0489±0.0000	0.0231±0.0000	0.0470±0.0000
a_{zx-}	0.0302±0.0104	0.0302±0.0104	0.0047±0.0102	0.0280±0.0104
a_{yx+}	0.0396±0.0127	0.0396±0.0128	0.0139±0.0125	0.0373±0.0127
a_{yx-}	0.0349±0.0127	0.0349±0.0128	0.0093±0.0125	0.0331±0.0127
a_{xx+}	0.0210±0.0000	0.0204±0.0000	6.0200e-5±0.0000	0.0210±0.0000
a_{xx-}	0.0441±0.0127	0.0448±0.0128	0.0139±0.0123	0.0400±0.0127
a_{zy+}	0.0221±0.0123	0.0222±0.0123	0.0111±2.3634e-4	0.0200±0.0123
a_{zy-}	0.0399±0.0101	0.0401±0.0102	0.0160±0.0100	0.0387±0.0100
a_{yy+}	0.0306±0.0123	0.0291±0.0123	0.0095±0.0123	0.0299±0.0123
a_{yy-}	0.0359±0.0124	0.0377±0.0125	0.0089±0.0123	0.0332±0.0123
a_{xy+}	0.0089±0.0100	0.0051±0.0100	0.0111±2.3634e-4	0.0156±0.0100
a_{xy-}	0.0439±1.0714e-4	0.0479±1.5629e-4	0.0114±2.3634e-4	0.0341±0.0000
a_{zz+}	0.0208±0.0000	0.0201±0.0000	3.2700e-4±0.0000	0.0202±0.0000
a_{zz-}	0.0221±0.0101	0.0207±0.0101	0.0045±0.0101	0.0228±0.0101
a_{yz+}	0.0414±0.0000	0.0402±0.0000	0.0229±0.0000	0.0415±0.0000
a_{yz-}	0.0196±0.0000	0.0188±0.0000	1.3100e-4±0.0000	0.0195±0.0000
a_{xz+}	0.0163±0.0074	0.0158±0.0101	0.0046±0.0100	0.0162±0.0071
a_{xz-}	0.0323±0.0124	0.0296±0.0124	0.0138±0.0124	0.0326±0.0124

Table 3.3: Average errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using only 6 positions. Methods 9 to 12.

	Method 9	Method 10	Method 11	Method 12
a_{zx+}	0.0472±0.0000	0.0472±0.0000	0.0468±0.0000	0.0499±0.0000
a_{zx-}	0.0278±0.0104	0.0278±0.0104	0.0282±0.0104	0.0312±0.0104
a_{yx+}	0.0371±0.0127	0.0371±0.0127	0.0375±0.0127	0.0406±0.0128
a_{yx-}	0.0333±0.0127	0.0333±0.0127	0.0328±0.0127	0.0359±0.0128
a_{xx+}	0.0217±4.7803e-6	0.0217±4.7803e-6	0.0209±0.0000	0.0210±0.0000
a_{xx-}	0.0394±0.0127	0.0394±0.0127	0.0400±0.0127	0.0461±0.0128
a_{zy+}	0.0152±0.0100	0.0152±0.0100	0.0158±0.0100	0.0170±0.0100
a_{zy-}	0.0389±0.0100	0.0389±0.0100	0.0383±0.0100	0.0396±0.0100
a_{yy+}	0.0291±0.0123	0.0291±0.0123	0.0299±0.0123	0.0307±0.0123
a_{yy-}	0.0340±0.0123	0.0340±0.0123	0.0332±0.0123	0.0349±0.0123
a_{xy+}	0.0158±0.0100	0.0158±0.0100	0.0158±0.0100	0.0170±0.0100
a_{xy-}	0.0338±7.5369e-6	0.0338±7.5369e-6	0.0293±0.0100	0.0305±0.0100
a_{zz+}	0.0193±0.0000	0.0193±0.0000	0.0201±0.0000	0.0208±0.0000
a_{zz-}	0.0220±0.0101	0.0220±0.0101	0.0228±0.0101	0.0243±0.0101
a_{yz+}	0.0411±0.0000	0.0411±0.0000	0.0419±0.0000	0.0430±0.0000
a_{yz-}	0.0184±0.0000	0.0184±0.0000	0.0192±0.0000	0.0203±0.0000
a_{xz+}	0.0156±0.0062	0.0156±0.0062	0.0161±0.0070	0.0167±0.0079
a_{xz-}	0.0320±0.0124	0.0320±0.0124	0.0328±0.0124	0.0339±0.0000

Table 3.4: Average total errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using only 6 positions. All methods.

	Average total error		Average total error
Method1	0.0092±0.0054	Method7	0.0100±0.0069
Method2	0.0310±0.0103	Method8	0.0295±0.0095
Method3	0.0350±0.0206	Method9	0.0290±0.0099
Method4	0.0305±0.0102	Method10	0.0286±0.0122
Method5	0.0307±0.0113	Method11	0.0290±0.0097
Method6	0.0303±0.0123	Method12	0.0308±0.0106

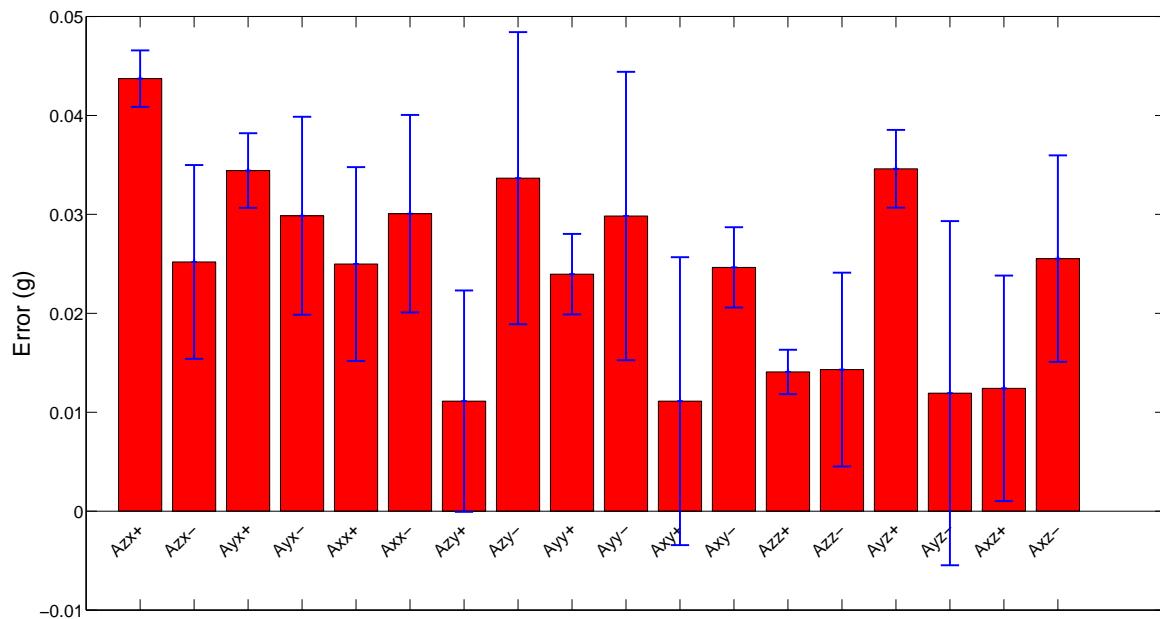


Figure 3.25: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 2.

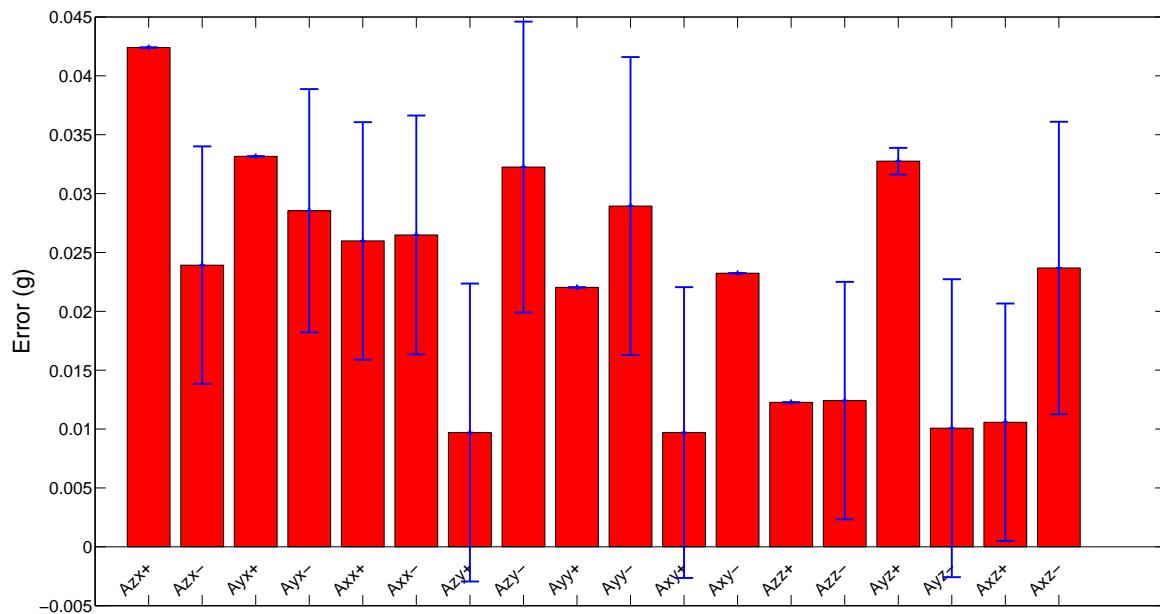


Figure 3.26: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 3.

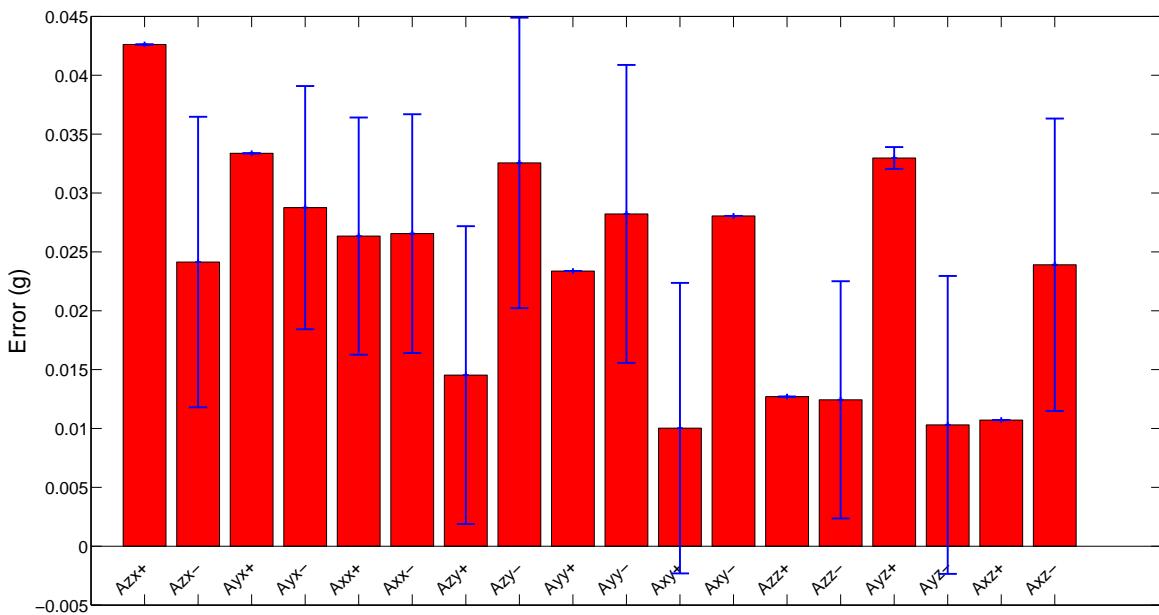


Figure 3.27: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 4.

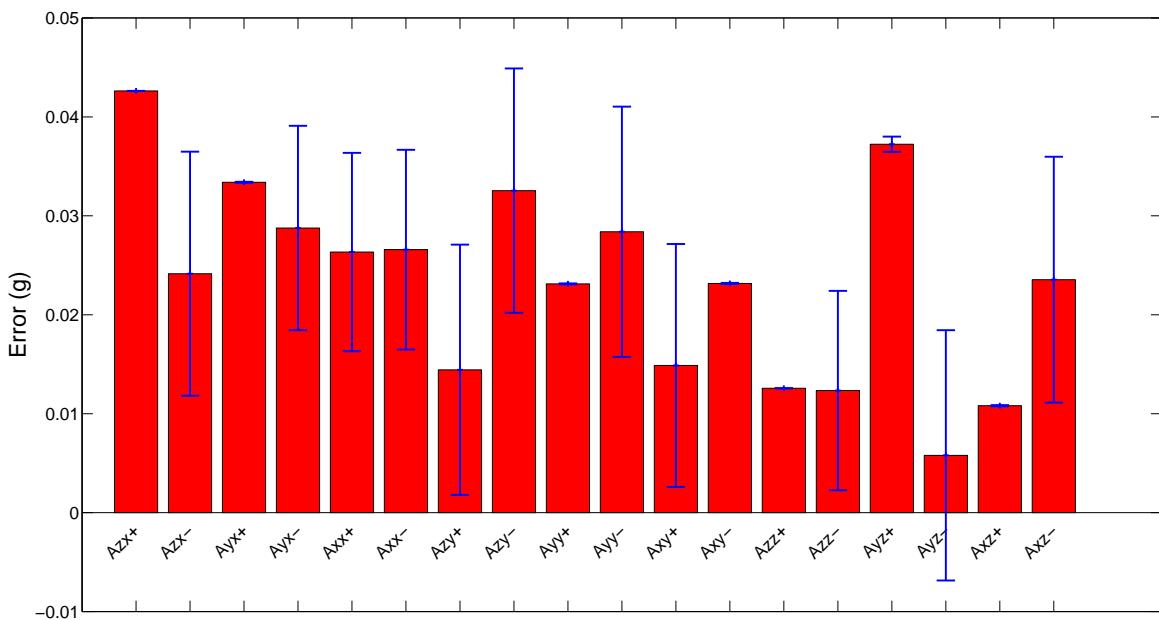


Figure 3.28: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 5.

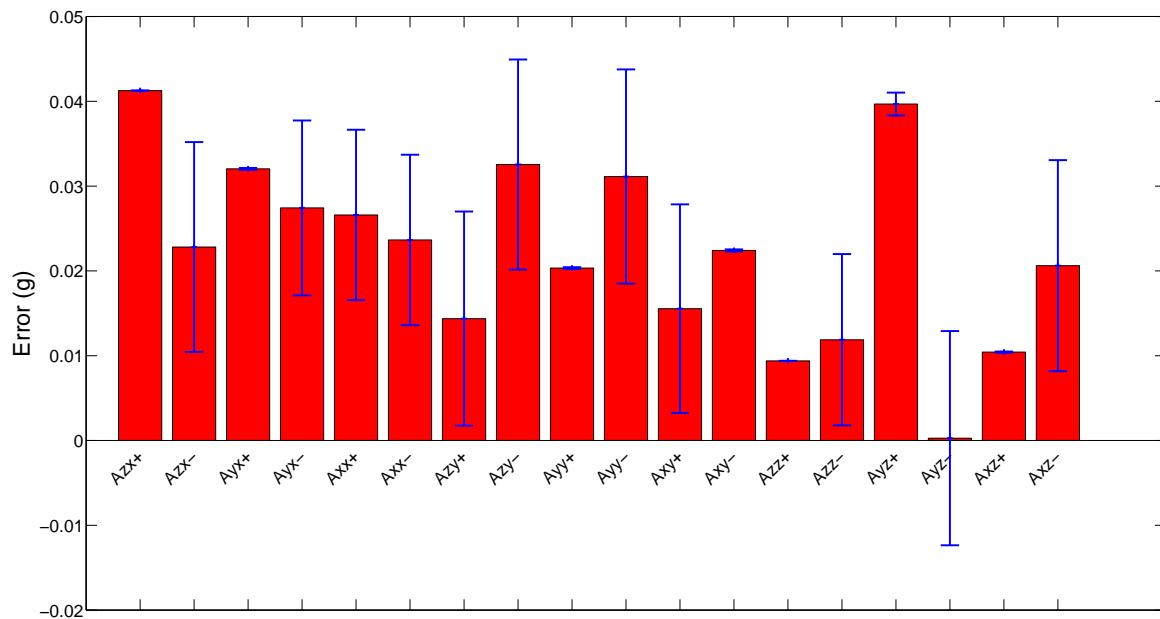


Figure 3.29: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 6.

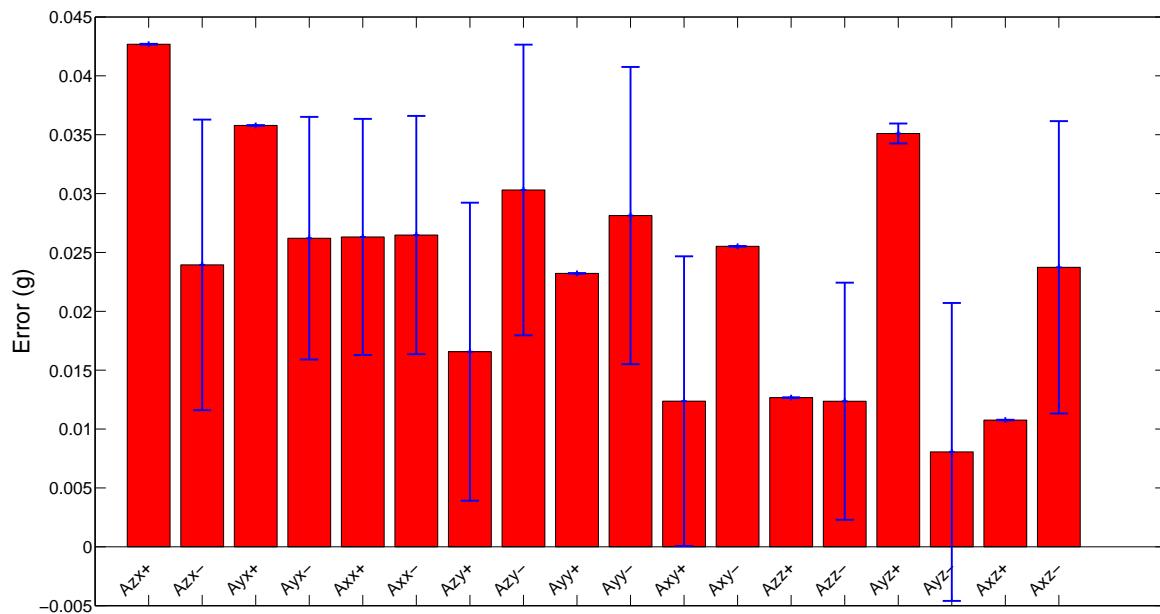


Figure 3.30: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 8.

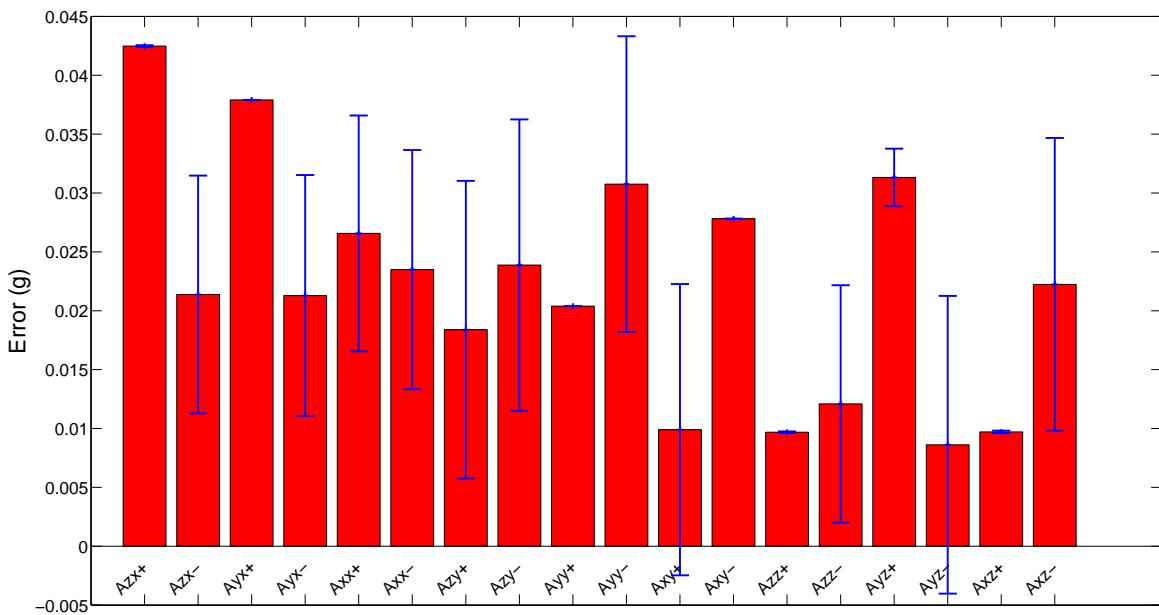


Figure 3.31: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 9.

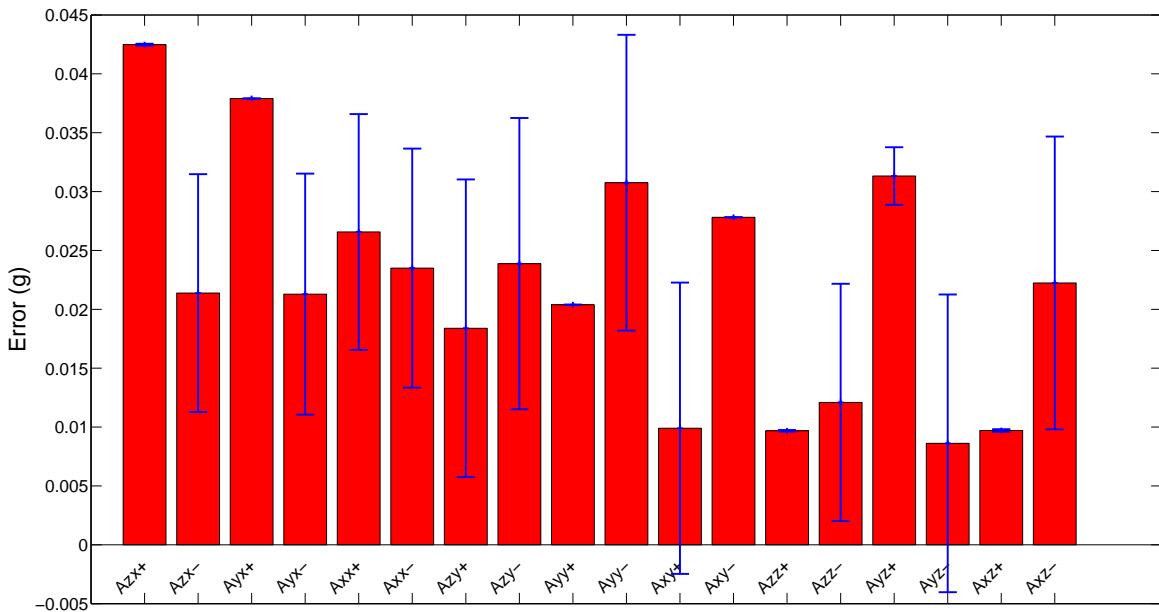


Figure 3.32: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 10.

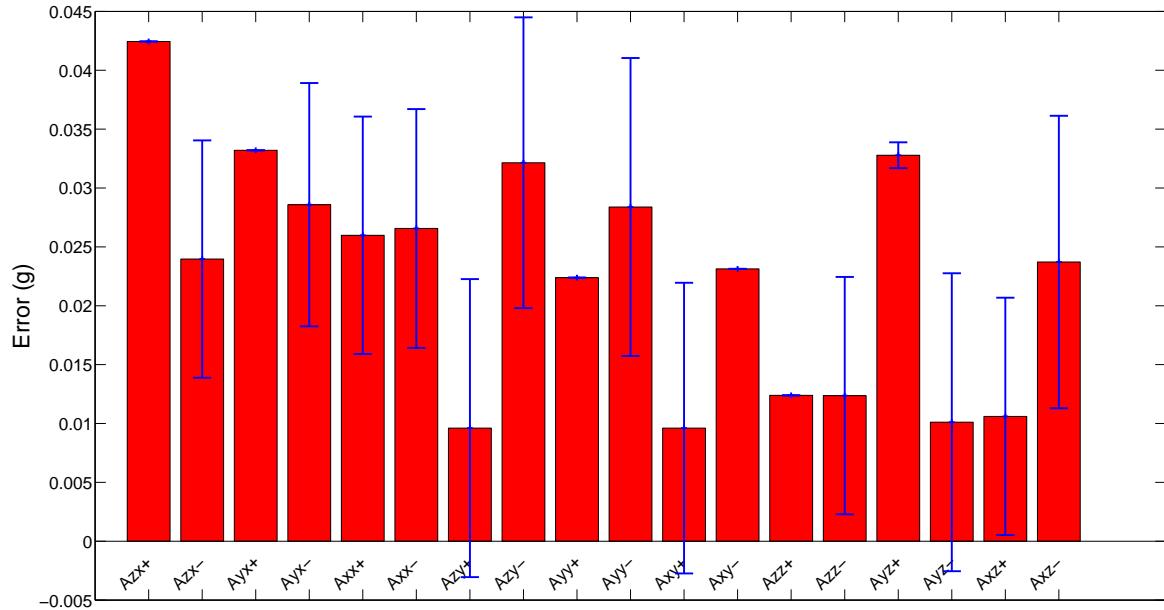


Figure 3.33: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. Method 11.

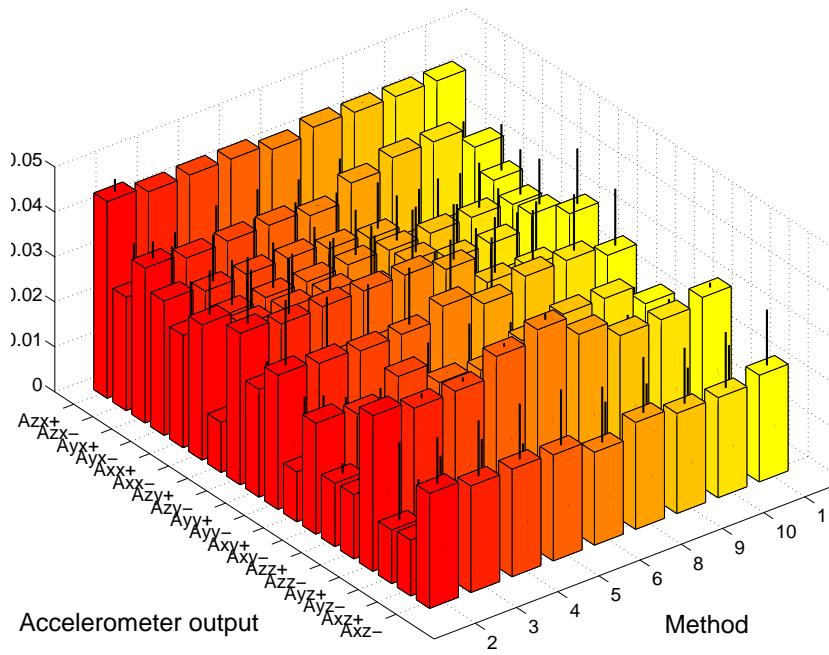


Figure 3.34: Average calibration errors for 5 six positions runs using parameters computed applying multi-position methods. All methods.

Table 3.5: Average errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using random multi-positions. Methods 2 to 6.

	Method 2	Method 3	Method 4	Method 5	Method 6
a_{zx+}	0.0437±0.0029	0.0424±0.0000	0.0426±1.5890e-8	0.0426±0.0000	0.0413±0.0000
a_{zx-}	0.0111±0.0098	0.0239±0.0100	0.0241±0.0123	0.0241±0.0123	0.0228±0.0123
a_{yx+}	0.0141±0.0038	0.0332±0.0000	0.0334±1.3674e-8	0.0334±1.3674e-8	0.0320±1.0614e-4
a_{yx-}	0.0252±0.0100	0.0285±0.0103	0.0288±0.0103	0.0288±0.0103	0.0274±0.0274
a_{xx+}	0.0337±0.0098	0.0260±0.0100	0.0263±0.0100	0.0263±0.0100	0.0266±0.0100
a_{xx-}	0.0143±0.0100	0.0265±0.0101	0.0265±0.0101	0.0266±0.0100	0.0236±0.0100
a_{zy+}	0.0344±0.0112	0.0097±0.0097	0.0145±0.0126	0.0144±0.0126	0.0144±0.0126
a_{zy-}	0.0240±0.0148	0.0323±0.0123	0.0326±0.0123	0.0325±0.0123	0.0325±0.0123
a_{yy+}	0.0346±0.0041	0.0022±0.0000	0.0234±1.3574e-8	0.0231±5.3627e-5	0.0203±1.0466e-4
a_{yy-}	0.0299±0.0146	0.0289±0.0126	0.0282±0.0126	0.0284±0.0126	0.0311±0.0126
a_{xy+}	0.0298±0.0146	0.0097±0.0123	0.0100±0.0123	0.0149±0.0122	0.0155±0.0123
a_{xy-}	0.0119±0.0041	0.0232±0.0000	0.0281±1.4381e-8	0.0232±5.6826e-5	0.0224±1.2311e-4
a_{zz+}	0.0250±0.0022	0.0123±0.0000	0.0127±1.3134e-8	0.0126±0.0000	0.0093±0.0000
a_{zz-}	0.0111±0.0098	0.0124±0.0100	0.0124±0.0100	0.0123±0.0100	0.0119±0.0101
a_{yz+}	0.0124±0.0039	0.0328±0.0011	0.0330±9.2474e-4	0.0372±7.7240e-4	0.0397±0.0013
a_{yz-}	0.0301±0.0174	0.0101±0.0126	0.0103±0.0126	0.0057±0.0126	2.7100e-4±2.7062e-4
a_{xz+}	0.0246±0.0114	0.0106±0.0100	0.0107±1.2240e-8	0.0108±6.1596e-5	0.0104±7.0887e-5
a_{xz-}	0.0255±0.0104	0.0237±0.0124	0.0239±0.0124	0.0235±0.0124	0.0206±0.0124

Table 3.6: Average errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using random multipositions. Methods 8 to 11.

	Method 8	Method 9	Method 10	Method 11
a_{zx+}	$0.0427 \pm 3.0548e-5$	$0.0425 \pm 7.4058e-5$	$0.0425 \pm 7.4058e-5$	0.0424 ± 0.0000
a_{zx-}	0.0239 ± 0.0123	0.0214 ± 0.0101	0.0214 ± 0.0101	0.0240 ± 0.0104
a_{yx+}	$0.0358 \pm 2.6900e-5$	0.0379 ± 0.0000	0.0379 ± 0.0000	0.0332 ± 0.0128
a_{yx-}	0.0262 ± 0.0103	0.0213 ± 0.0102	0.0213 ± 0.0102	0.0286 ± 0.0103
a_{xx+}	0.0263 ± 0.0100	0.0266 ± 0.0100	0.0266 ± 0.0100	0.0260 ± 0.0100
a_{xx-}	0.0265 ± 0.0101	0.0235 ± 0.0101	0.0235 ± 0.0101	0.0266 ± 0.0101
a_{zy+}	0.0166 ± 0.0126	0.0184 ± 0.0126	0.0184 ± 0.0126	0.0096 ± 0.0126
a_{zy-}	0.0303 ± 0.0123	0.0239 ± 0.0123	0.0239 ± 0.0123	0.0321 ± 0.0123
a_{yy+}	$0.0232 \pm 2.6707e-5$	0.0204 ± 0.0000	0.0204 ± 0.0000	0.0224 ± 0.0000
a_{yy-}	0.0281 ± 0.0126	0.0308 ± 0.0125	0.0308 ± 0.0125	0.0284 ± 0.0126
a_{xy+}	0.0124 ± 0.0123	0.0099 ± 0.0123	0.0099 ± 0.0123	0.0096 ± 0.0123
a_{xy-}	$0.0255 \pm 2.8256e-5$	0.0278 ± 0.0000	0.0278 ± 0.0000	0.0231 ± 0.0000
a_{zz+}	$0.0127 \pm 2.5242e-5$	$0.0096 \pm 6.4902e-5$	$0.0096 \pm 6.4902e-5$	0.0124 ± 0.0000
a_{zz-}	0.0124 ± 0.0100	0.0121 ± 0.0100	0.0121 ± 0.0100	0.0124 ± 0.0100
a_{yz+}	$0.0351 \pm 8.4784e-4$	0.0313 ± 0.0024	0.0313 ± 0.0024	0.0328 ± 0.0010
a_{yz-}	0.0080 ± 0.0126	0.0086 ± 0.0126	0.0086 ± 0.0126	0.0101 ± 0.0126
a_{xz+}	$0.0108 \pm 2.3688e-5$	$0.0097 \pm 1.0620e-4$	$0.0097 \pm 1.0620e-4$	0.0106 ± 0.0100
a_{xz-}	0.0237 ± 0.0124	0.0222 ± 0.0124	0.0222 ± 0.0124	0.0237 ± 0.0124

Table 3.7: Average total errors of calibrated acceleration with respect to local gravity when each axis is set parallel and antiparallel to gravity. Calibration parameters are computed using random multi-positions. All methods.

	Av. total error		Av. total error		Av. total error
Method 2	0.0242 ± 0.0098	Method 5	0.0234 ± 0.0099	Method 9	0.0221 ± 0.0098
Method 3	0.0227 ± 0.0099	Method 6	0.0224 ± 0.0109	Method 10	0.0221 ± 0.0098
Method 4	0.0234 ± 0.0096	Method 8	0.0233 ± 0.0096	Method 11	0.0227 ± 0.0098

Table 3.8: Average magnetic field magnitude error with respect to local Earth's magnetic field. Calibration parameters are computed using random multi-positions. All methods.

	Av. total error		Av. total error		Av. total error
Method 2	$0.0177 \pm 6.5345e-4$	Method 5	$0.0030 \pm 6.8702e-4$	Method 9	$0.0030 \pm 6.8702e-4$
Method 3	$0.0176 \pm 6.3087e-4$	Method 6	$0.0031 \pm 6.9857e-4$	Method 10	$0.0030 \pm 6.8702e-4$
Method 4	$0.0179 \pm 4.3012e-4$	Method 8	$0.0030 \pm 6.8702e-4$	Method 11	$0.2458 \pm 4.4721e-5$

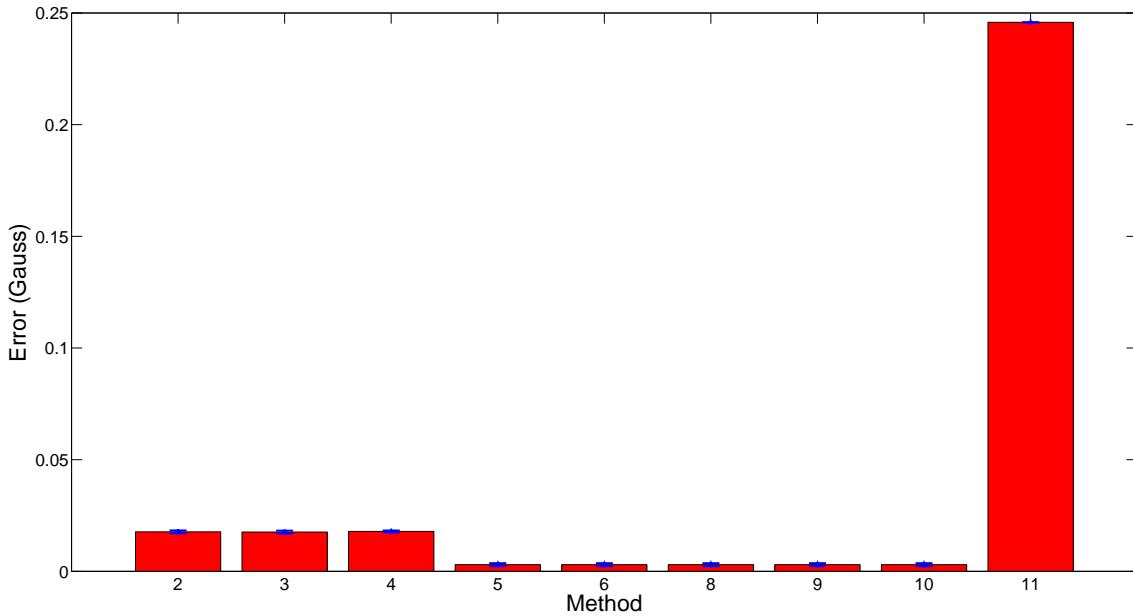


Figure 3.35: Average magnetic field magnitude errors after calibration for 5 multi-position runs . All methods.

3.4.4 Discussion of results

We start the discussion of results analyzing the calibration of the accelerometer based on the six positions dataset. From all methods, number 1 proposed by Ferraris et al. achieves the lowest total average error (0.0092 ± 0.0054 g). This was expectable as this method is specifically designed for a 6 positions dataset and the calibration parameters are calculated applying direct and simple algebraic operations. Moreover, the sensor modeling is very complete. Method number 7, which is also designed for a 6 positions dataset has obtained the second lowest error (0.0100 ± 0.0690 g). A reason that may explain its inferiority against method 1 is the fact that its model is simpler as it does not include non-orthogonality nor misalignment errors.

The rest of the algorithms present errors ranging from 0.0286 ± 0.0122 g obtained by method 10, to 0.0360 ± 0.0206 g obtained by method 3. The poorer performance of these methods can be associated to the low number of data points that are used to minimize the cost function. It is well known that the larger the number of points used to minimize a function is, the better is the estimation of the parameters.

This fact is also verified in the second part of the accelerometer calibration study. Using a larger dataset (73 points versus 6 points) has decreased the errors around a 20%. Methods 9 and 10 obtain the lowest error both with 0.0221 ± 0.0098 g. However, even with a larger dataset, multi-position methods do not outperform methods 1 and 7. In general, the estimation of parameters through the minimization of cost functions seems to be less accurate than direct algebraic transformations of six positions methods.

Another reason that may explain the larger errors of multi-point methods may lie in the good quality of the employed sensors and their good integration in the PCB. This fact causes the misalignment and non-orthogonality errors to be almost negligible. Therefore, forcing a cost function to include them may reduce the quality of the scale factor and bias estimates.

Finally, going through the results obtained for the magnetometer calibration, we see that methods 5, 8, 9 and 10 achieve the lowest error with $0.0030 \pm 6.8702e - 4$ Gauss. These methods have similar and very complete output models and the minimization procedure finds parameters which lead to a very low magnitude error. On the other hand, method 11 has shown a very poor performance. This may be explained by the different model it uses—for example, the scale factor matrix has $1 + S_i$ terms in the diagonal instead of just S_i . This causes the sphere to have a wrong radius, which, in turn, leads to a large magnitude error. Anyhow, it manages to properly convert the ellipsoid into a sphere centered in the origin, so the hard iron and soft iron effects are dealt with.

3.5 Proposed algorithms

3.5.1 Optimizer of spatial distribution of input calibration data

In the previous section we have seen the importance of having a large dataset to run the calibration algorithms. However, the number of points is not enough to warrant a good estimation of the calibration parameters. In addition to a large number of points we also need them to be well distributed around the space i. e. they should cover as much surface of the sphere (accelerometer) or ellipsoid (magnetometer) as possible.

On the other hand, a very large dataset may lead to a slower convergence of the

minimization algorithms, making them unsuitable for calibration in real time.

Therefore, it is desirable to find an optimal balance between distribution and number of data points. Gathering a set of N points is very simple as we just have to stop reading data from the sensors when the buffer or the memory has reached N samples. On the other hand, determining the quality of the data distribution is complicated by just eyeballing it, so [52] proposed a software application that visualizes both ϕ and θ spherical coordinates in real time while the data is being gathered. This way, the operator can move the MIMU to cover all the space and fill in any possible gaps. Figures 3.36-3.42 show the three dimensional representation of four magnetic field datasets having a different distribution in space. We can see that the visualization of ϕ with respect to θ helps to determine the distribution of the points in space.

However, this solution needs from online connection between the MIMU and a base station where data is to be visualized. We thought it would be a good idea to develop a system that automatically detects the distribution of the data points and enables the calibration procedure only when the distribution is adequate. Moreover, such a system could be implemented in the MIMU's firmware so it tells the operator—with a sound or a LED—to stop moving the MIMU while the input data is being gathered.

To this purpose, we have developed a system that computes both θ and ϕ coordinates of the dataset and calculates their histogram using 4000 bins. Then, the estimation of the parameters is only enabled when a predefined number of K bins is empty.

Figures 3.44-3.47 show the histograms associated to the four datasets indicated before, as well as the number of empty bins.

The predefined number of tolerated empty bins will depend on the employed algorithm.

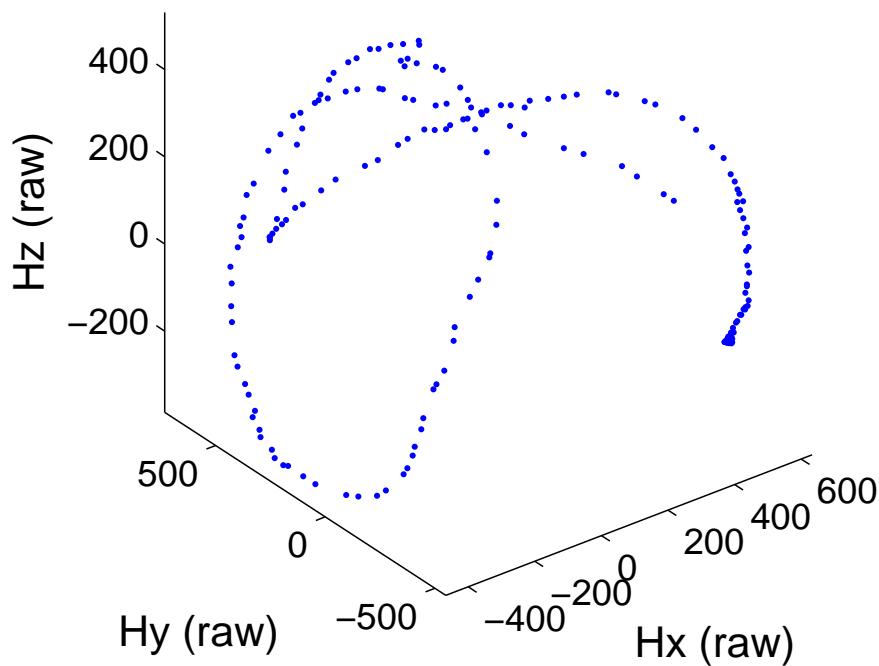


Figure 3.36: 3D Representation of raw magnetic field. Dataset 1.

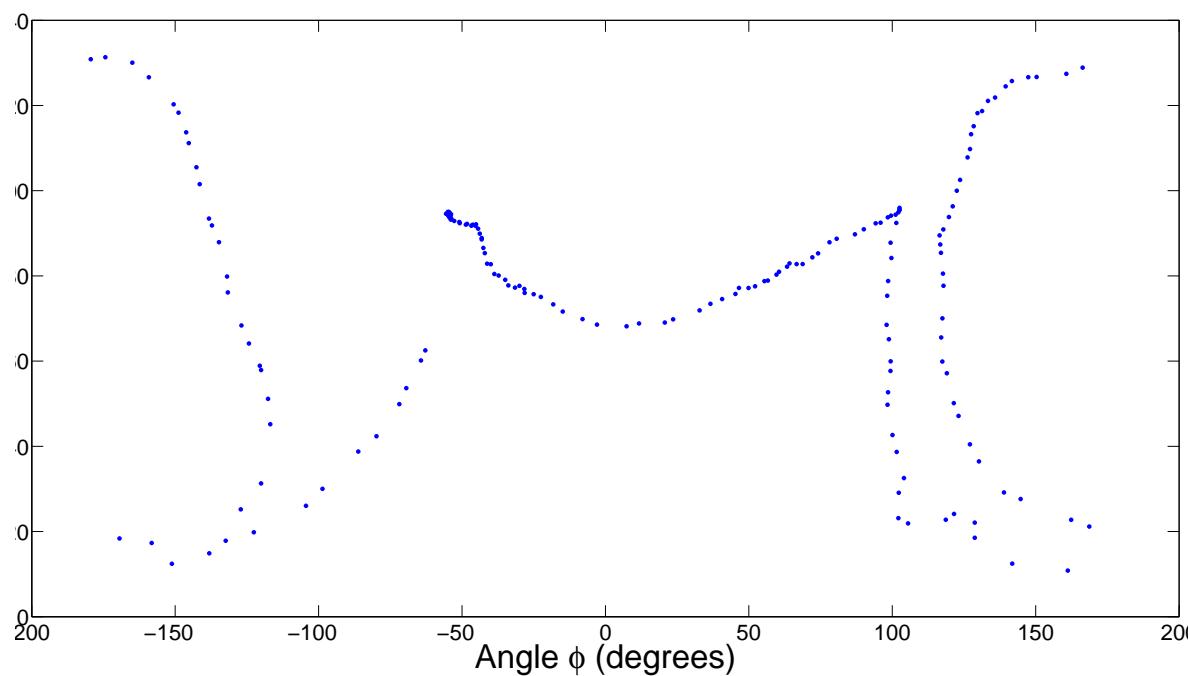


Figure 3.37: Representation of θ and ϕ spherical coordinates . Dataset 1.

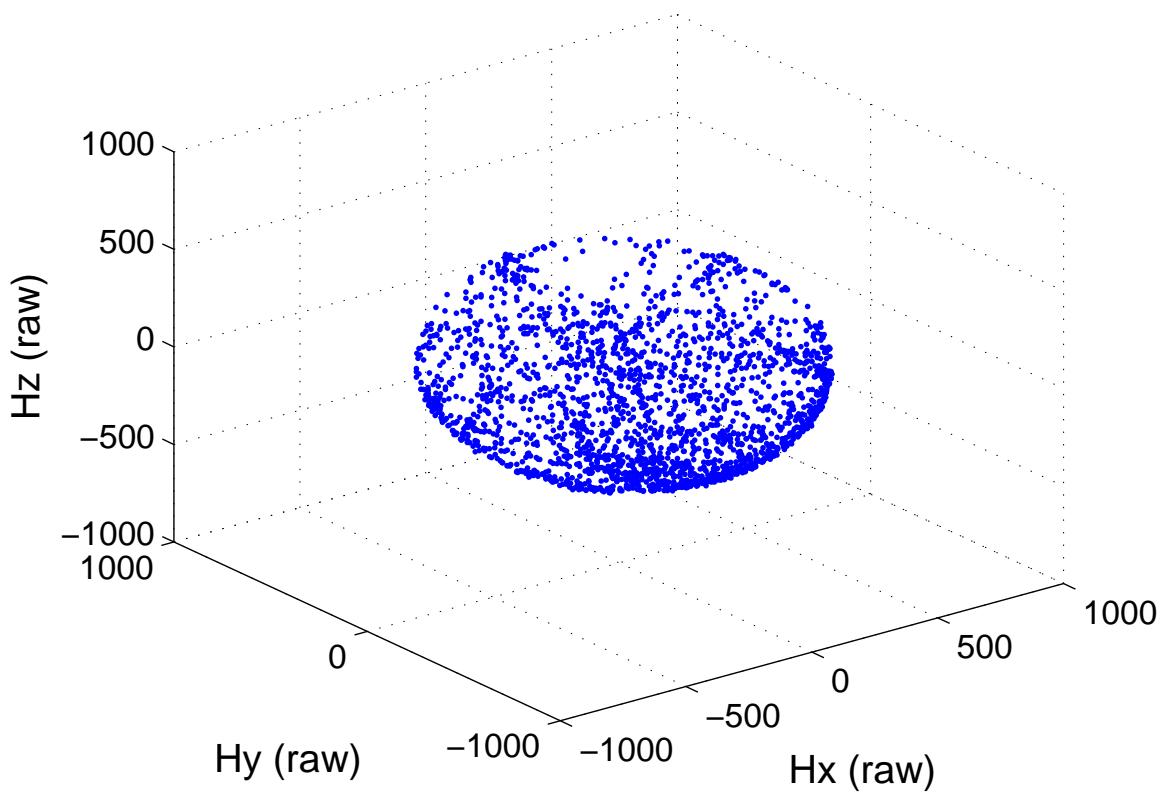


Figure 3.38: 3D Representation of raw magnetic field. Dataset 2.

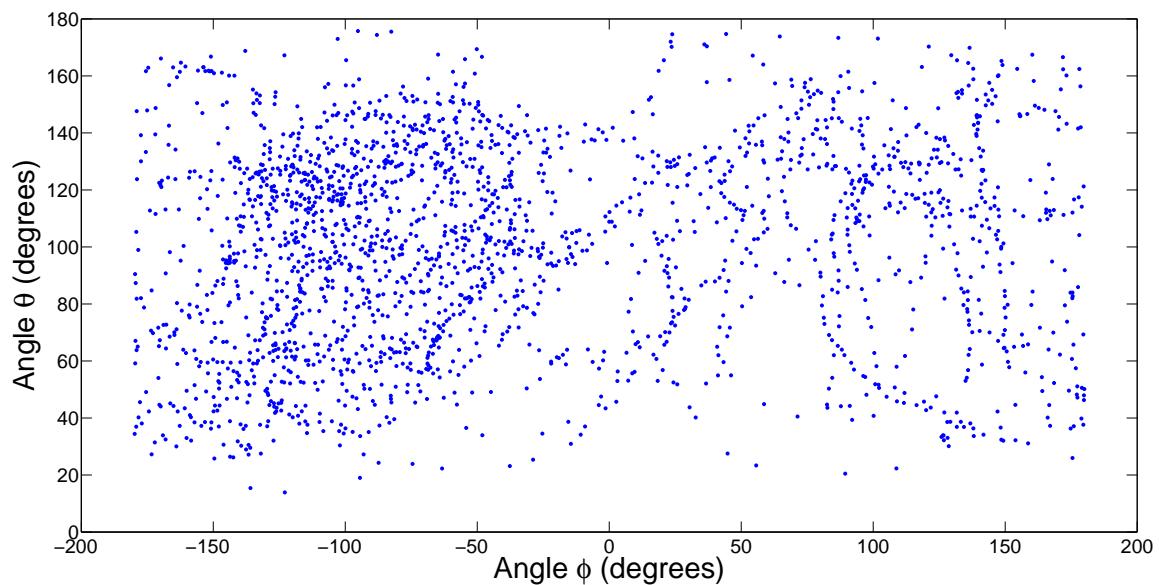


Figure 3.39: Representation of θ and ϕ spherical coordinates . Dataset 2.

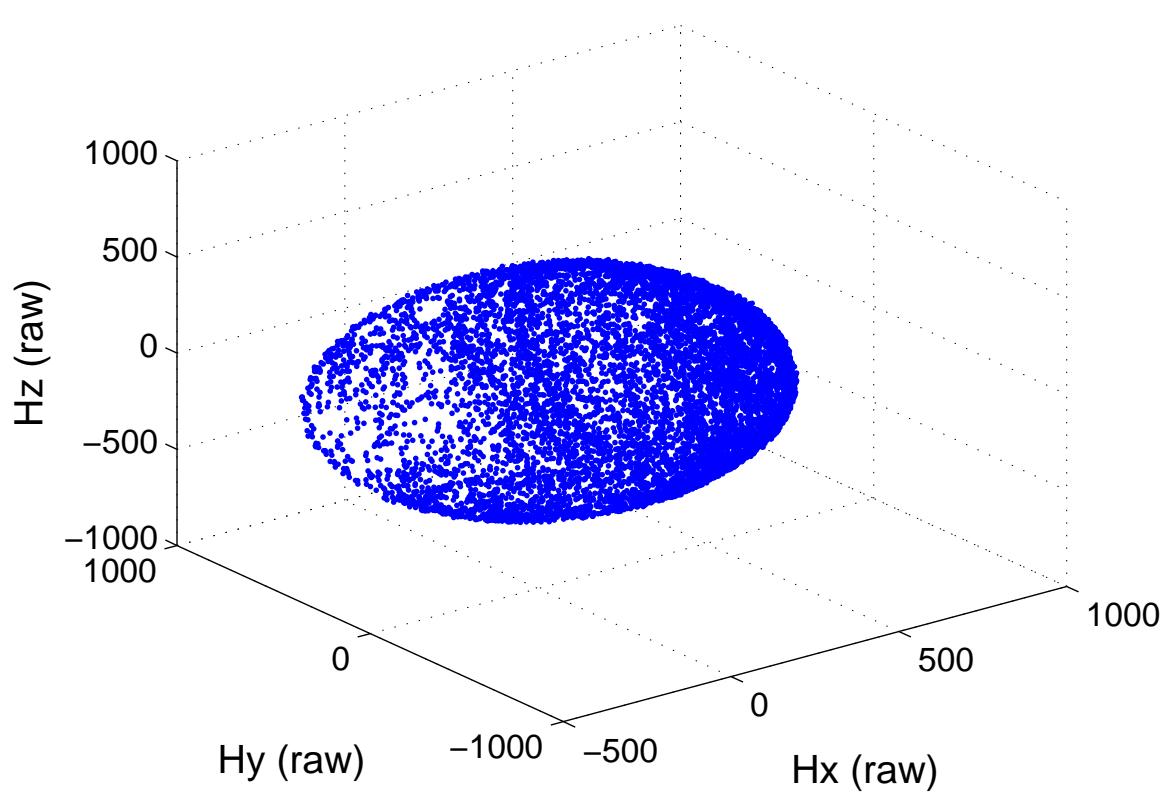


Figure 3.40: 3D Representation of raw magnetic field. Dataset 3.

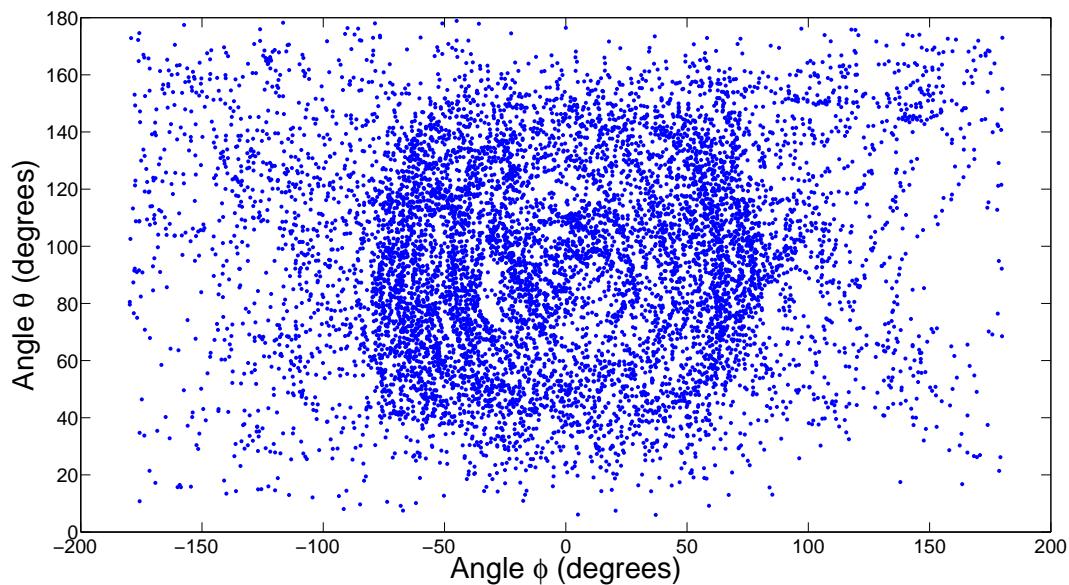


Figure 3.41: Representation of θ and ϕ spherical coordinates . Dataset 3.

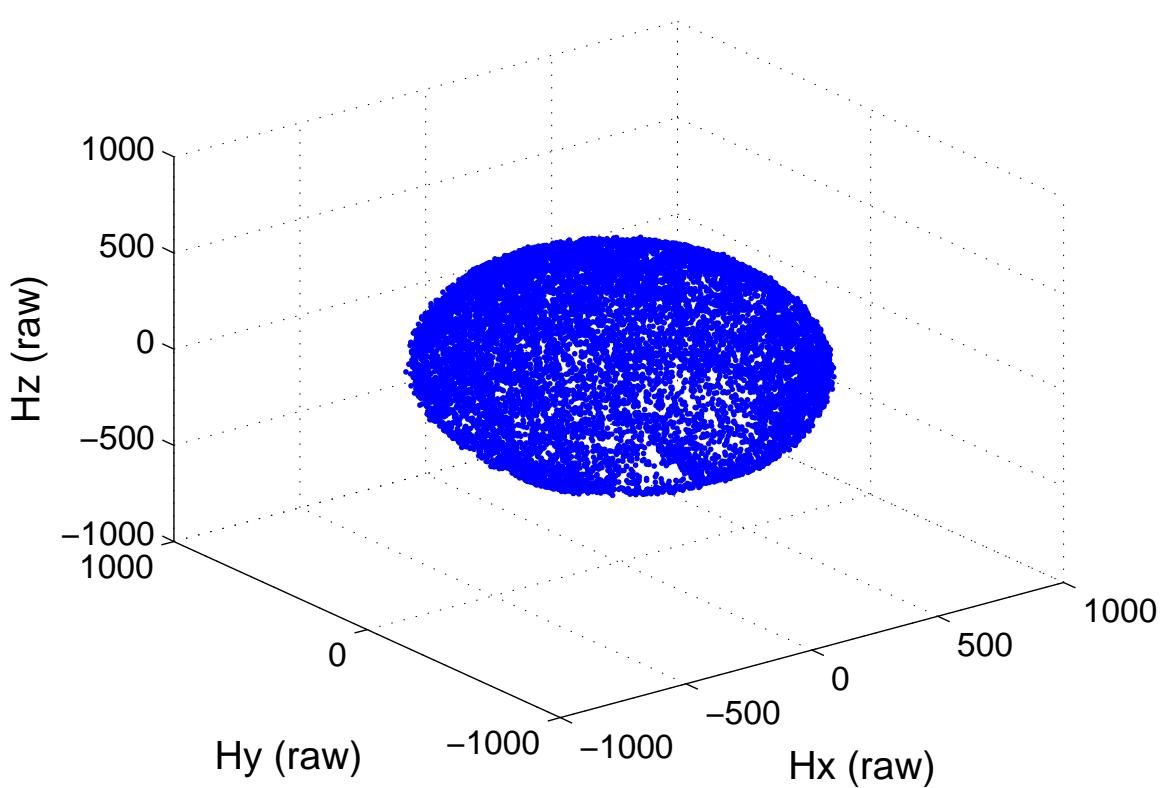


Figure 3.42: 3D Representation of raw magnetic field. Dataset 4.

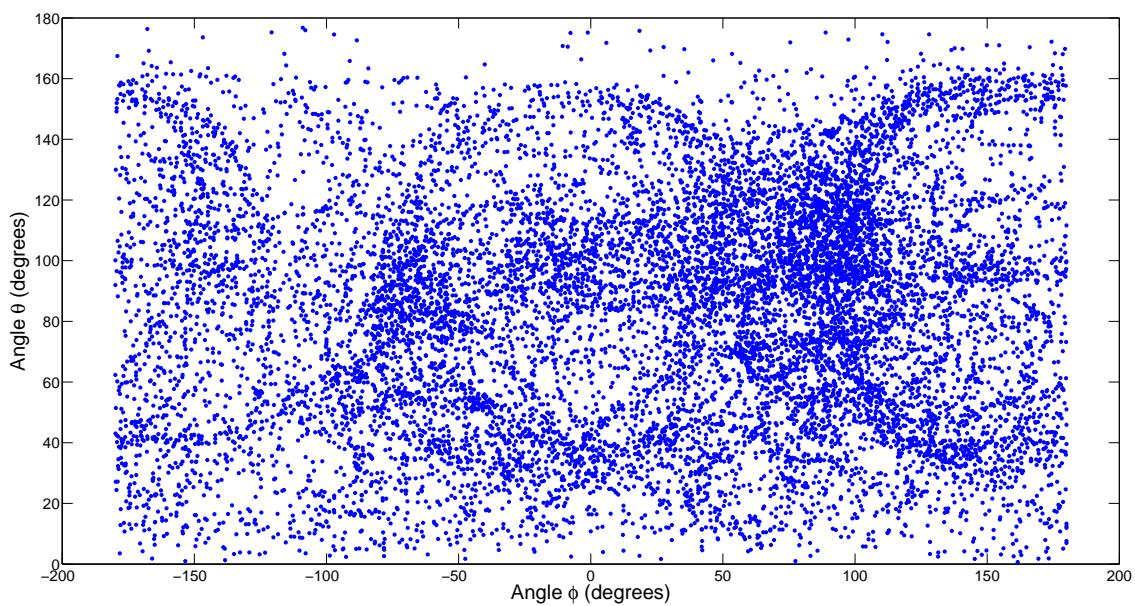


Figure 3.43: Representation of θ and ϕ spherical coordinates . Dataset 4.

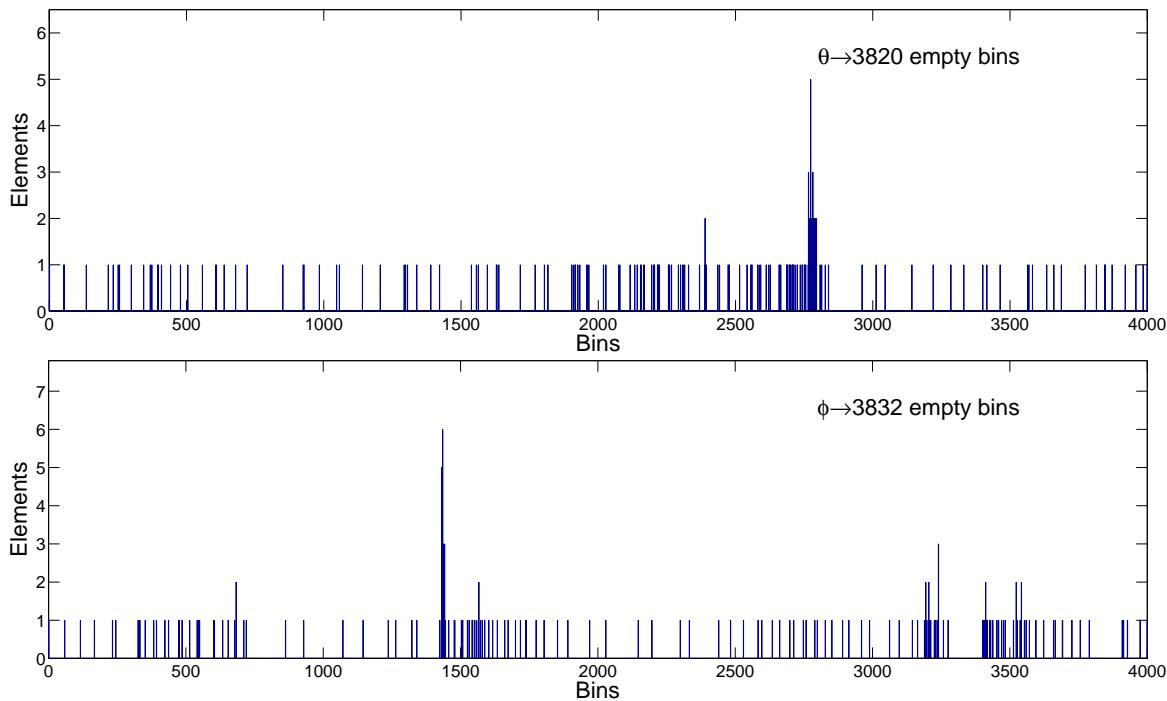


Figure 3.44: Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 1.

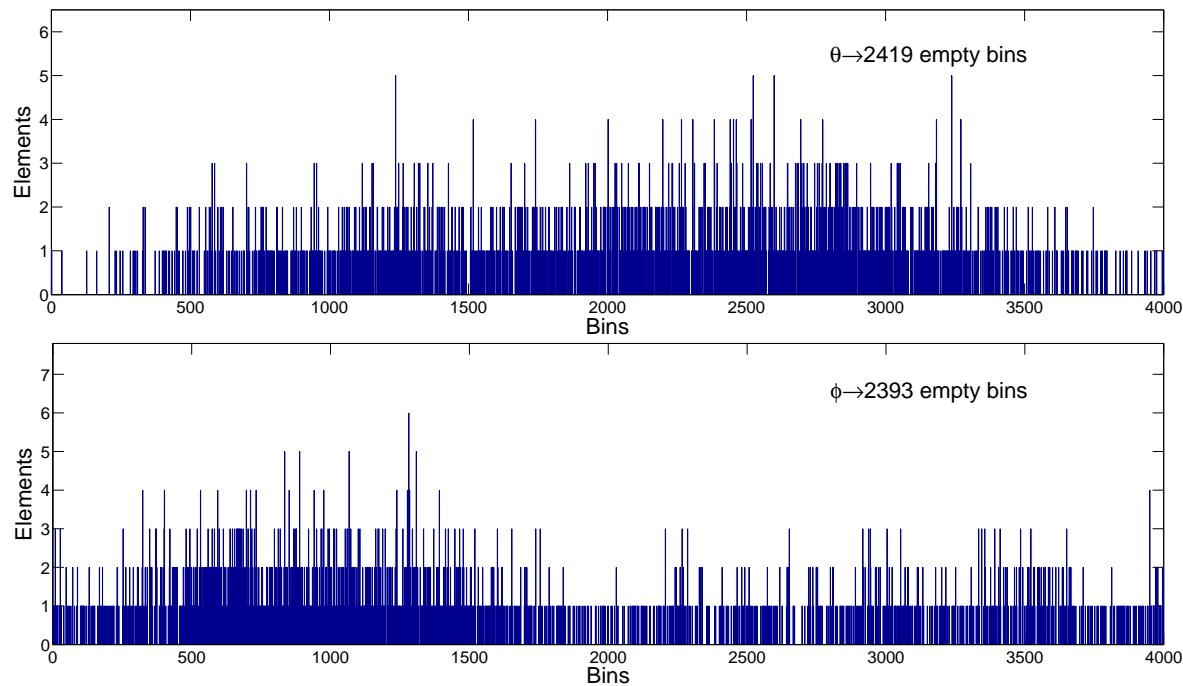


Figure 3.45: Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 2.

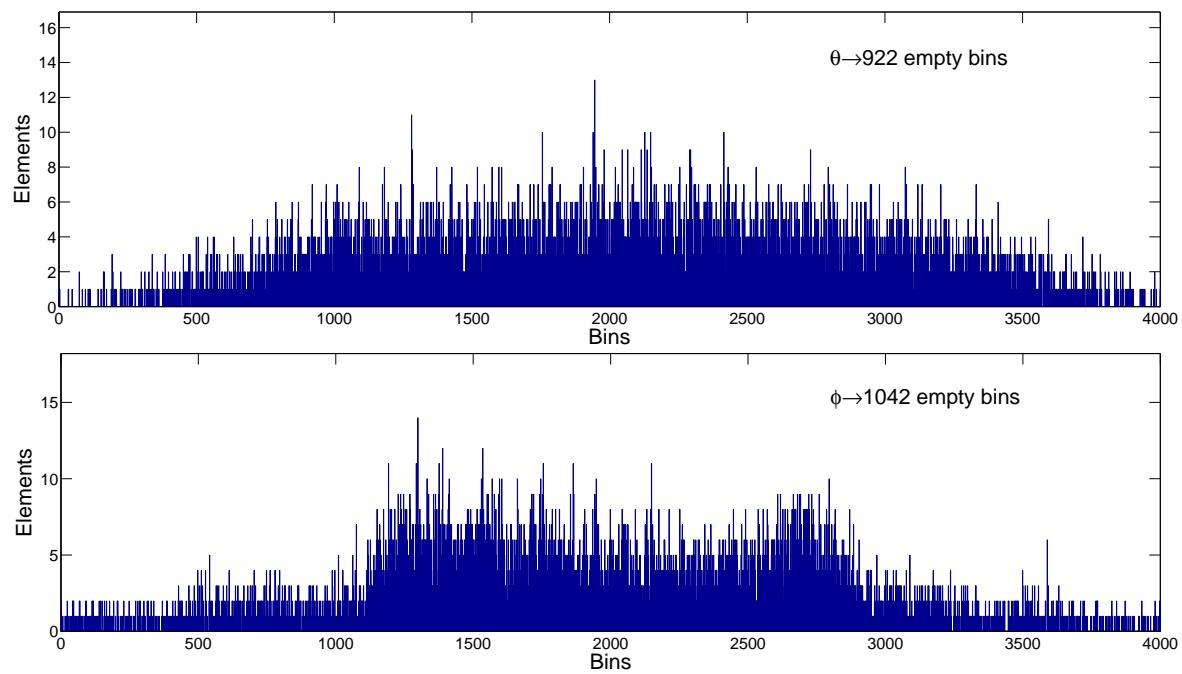


Figure 3.46: Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 3.

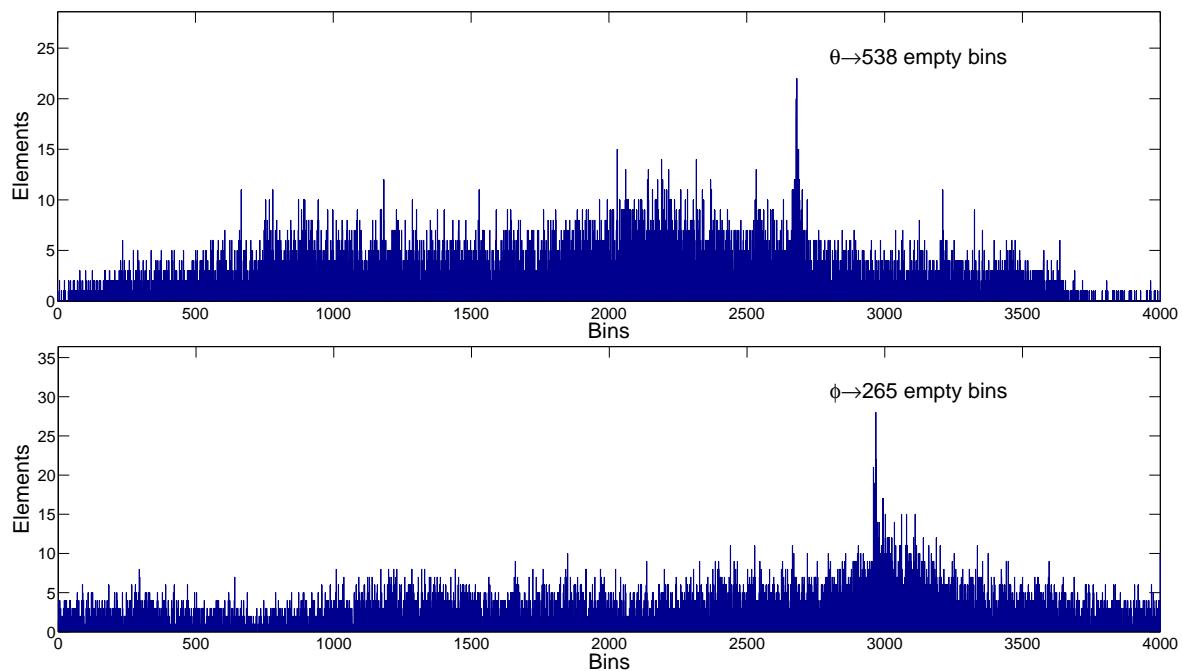


Figure 3.47: Histogram of θ and ϕ spherical coordinates (4000 bins). Dataset 4.

3.5.2 A novel accelerometer-aided gyroscope calibration

Up to this point the reader may be aware that we have not discussed the calibration of gyroscopes so far. Most gyroscope calibration algorithms need from complex and expensive multi-axis rate tables. We were lacking such an equipment so we were not able to carry out a comparative study between different existing algorithms. However, the lack of rate tables put us thinking of other alternatives to calibrate the gyroscope without the need of expensive equipment. This is how we came out with the idea of using a bicycle wheel—mainly because of its high rotation momentum—to which we could attach the Wagymag in order to rotate it around its axes.

Nonetheless, we still had the problem of computing the angular rate of the wheel while it was spinning. Then, we realized that, if the wheel axis was slightly tilted, the signal gathered by the accelerometer showed a sinusoidal behavior. This is caused by the influence of gravity on the centripetal acceleration, which will make the output of the accelerometer increase when the centripetal acceleration component in the X axis has the same sense than the gravity component along the X axis, and decrease when they have opposed senses. The sinusoidal signal will complete a period when a wheel turn is completed. This behavior is better understood when observing figure 3.48

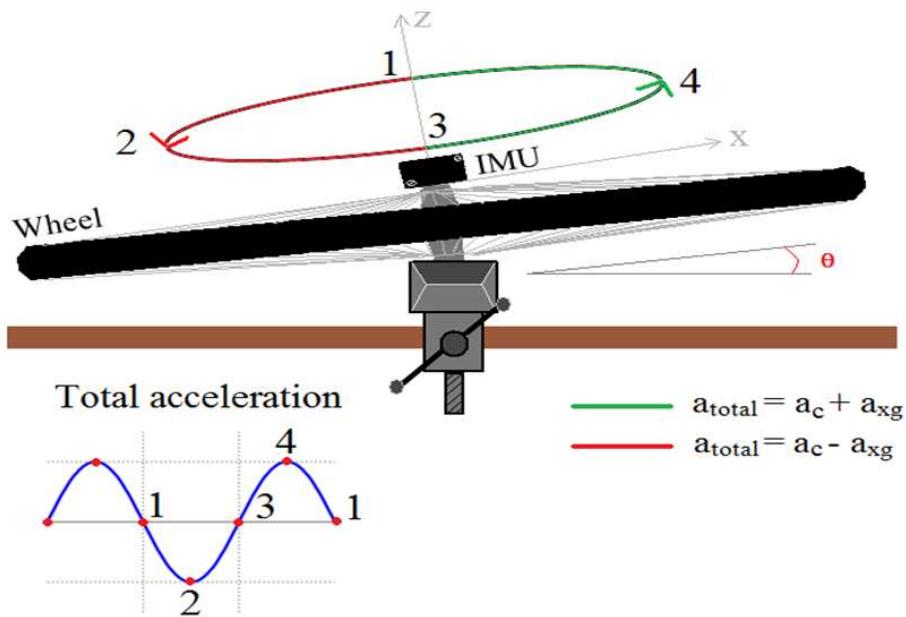


Figure 3.48: Configuration of the accelerometer-aided gyroscope calibration. The MIMU is attached to the tilted axis of the wheel.

This figure shows that the total measured acceleration $\mathbf{a}_{\text{total}}$ is composed of the centripetal acceleration \mathbf{a}_c together with the gravity acceleration component along the X axis \mathbf{a}_{gx} .

$$\mathbf{a}_c = -\omega r^2 \quad (3.95)$$

$$\mathbf{a}_{gx} = \mathbf{g} \sin \theta \quad (3.96)$$

$$\omega = 2\pi f = \frac{2\pi}{T} \quad (3.97)$$

In equation (3.95), r has a small value because the MIMU is placed on the spin axis to gather proper data from the gyroscope; still this fact is not significant since we will be able to observe the sinusoidal signal.

The accelerometer and gyroscope data are stored along with a time stamp column which will let us to detect the wheel turns in the acceleration signal and, therefore, determine the period of each turn. A vector of periods is built and by applying (3.97) a vector with the angular velocity at instant 2 (figure 3.48) of each wheel turn is found. This vector is then paired with the gyroscope raw data, so pairs of $[\omega_{\text{raw}}, \omega_{\text{acc}}]$, where ω_{raw} is the raw angular rate data and ω_{acc} is the estimated angular rate using the accelerometer signal, are computed. These pairs are subsequently used to build the calibration line by fitting the points using least squares.

The friction will reduce the angular velocity of the wheel in a progressive way until it stops; this way the whole sensing range of the gyroscope is comprised, from the highest rate to static state. The wheel has to be spun clockwise and anti-clockwise in order to obtain values in the positive and negative range.

The tilt angle of the spin axis has to be chosen with care because the gyroscope is sensitive to gravity acceleration due to its structure and working principle. Micro-machined Silicon Vibration Gyroscopes are based on the Coriolis force to sense angular velocity.

Among MEMS gyroscopes tuning fork is a much extended structure. Its working principle is very simple as stated in figure 3.49; a tuning fork is made vibrate when spinning, which makes the Coriolis force move the arms of the tuning fork. The measured displacement is, therefore, directly dependent to the angular velocity.

Figure 3.50 shows the behavior of the tuning fork when a tilt is applied to its rotating axis. The acceleration resulting from the force applied to the fork to generate the vibration is distorted by the influx of the component of the gravity in the displacement axis. The total acceleration (3.98) will be greater when the vibration acceleration, \mathbf{a}_{vib} , has the same sense than the gravity component (3.99) and, on the other hand, it will be smaller when they have opposite senses. The same behavior is present in all vibration based gyroscopes.

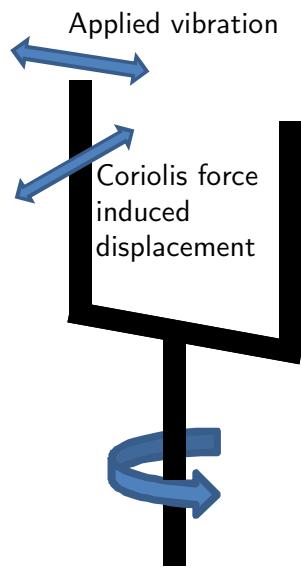


Figure 3.49: Working principles of a tuning fork Coriolis gyroscope.

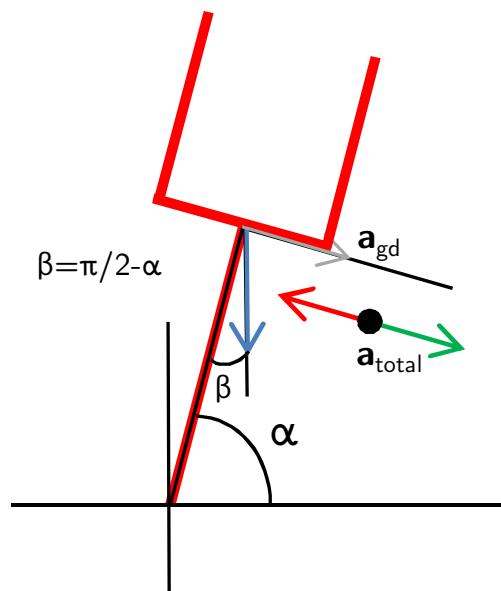


Figure 3.50: Acceleration effects on a tuning fork gyroscope.

$$\mathbf{a}_{\text{total}} = \mathbf{a}_{\text{vib}} + \mathbf{a}_{\text{gd}} \quad (3.98)$$

$$\mathbf{a}_{\text{gd}} = \mathbf{g} \sin(\pi/2 - \alpha) \quad (3.99)$$

Hence, the gyroscope output presents a sinusoidal nature which is more pronounced the higher the tilt angle gets, having its peak when $\alpha = \pi/2$.

The smaller the tilt angle, the least gravity acceleration effects are present. However when the tilt angle comes close to zero the sinusoidal acceleration signal gets very noisy to a point where it is impossible to differentiate the periods.

A trade-off between the two situations is needed. After several tests a tilt angle of $\alpha = 7^\circ$ showed to be the best option, as the periods of the acceleration signal were observable and the effects of the gravity on the gyroscope output were almost negligible.

The whole explained procedure was used to estimate the calibration line of the biaxial gyroscope included in Wagyro, a previous version of Wagymag which only included a triaxial accelerometer and a biaxial gyroscope.

So, by spinning the wheel with a tilt angle of $\alpha = 7^\circ$ clockwise and anti-clockwise for both MIMU's gyroscope sensitivity axis the calibration lines are obtained. Figures 3.51, 3.52, 3.53 and 3.54 show the process. Figure 3.51 shows the acceleration signal sensed in the Z axis when the wheel is spinning clockwise. The progressive decrease of the frequency reveals the decrease of the wheel's angular velocity.

Time difference between consecutive peaks is computed to obtain a vector including the period of each wheel turn. We can observe at a glance in Figure 3.52, how the periods augment due to the effect of the friction. By applying (3.97) on the periods vector, a vector with the angular velocity of each turn is calculated (Figure 3.53). Finally the trend line of the calculated angular velocity is paired with the stored gyroscope data and put together with the trend line obtained following the same procedure for the data recorded during the negative spin. Figure 3.54 shows the calculated calibration line for the X sensitivity axis. Experimental values show a very clear linear behavior so they are accurately fit by the trend line estimation.

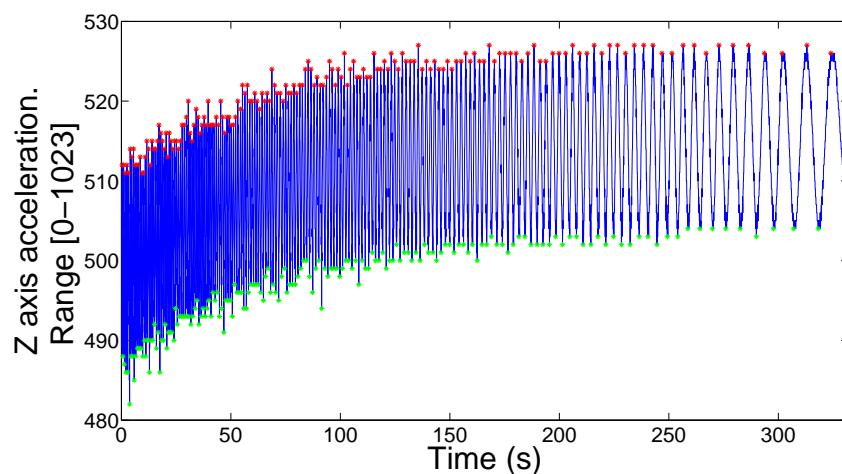


Figure 3.51: Peak detection of the accelerometer signal recorded during the positive spin. Turn periods increase with time due to friction.

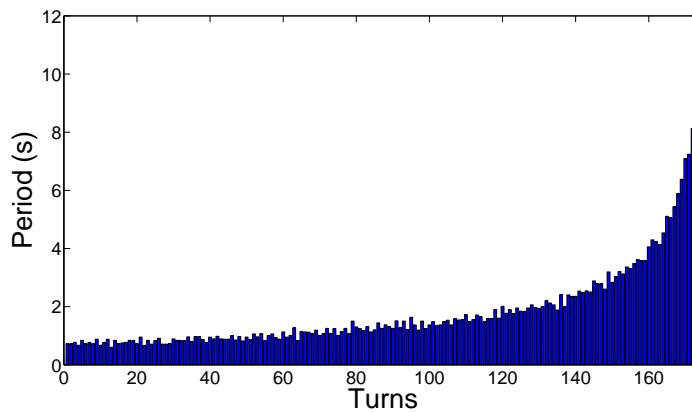


Figure 3.52: Wheel's turn periods obtained from the positive spin using the accelerometer data.

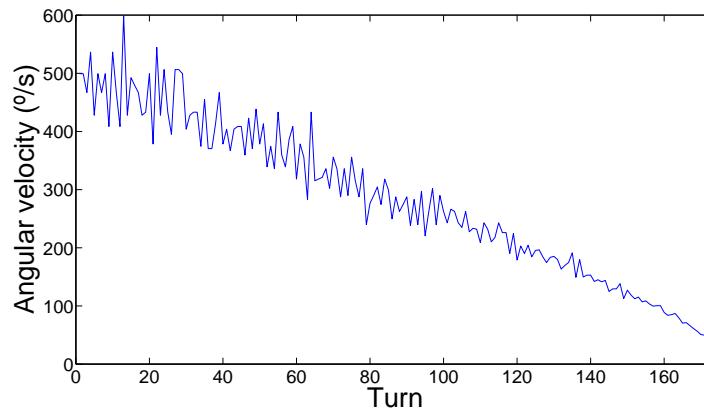


Figure 3.53: Angular velocity obtained with accelerometer data (positive spin).

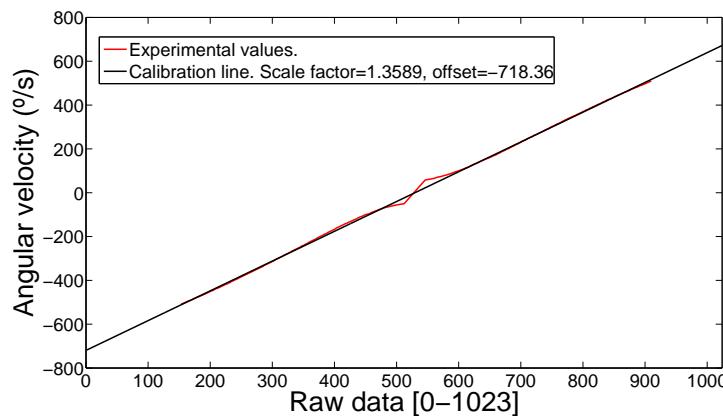


Figure 3.54: Calibration line calculated for the X sensitivity axis.

Expressions (3.100) and (3.101) show the calculated calibration lines.

$$u_{gx} = 1.3589n_{gx} - 718.34 \quad (3.100)$$

$$u_{gy} = 1.3658n_{gy} - 788.06 \quad (3.101)$$

Where u_{gx} and u_{gy} are the angular velocities in $^{\circ}/s$ for the X and Y axes respectively and n_{gx} and n_{gy} are the gyroscope raw data outputs for X and Y axes respectively.

3.5.3 A novel magnetometer-aided gyroscope calibration

With the integration of a magnetometer in the second version of our MIMU (Wagyromag), we observed a new possibility of measuring the turn rate of the bicycle wheel. This time, instead of using the accelerometer signal, which requires a slight inclination of the wheel's axis to show a sinusoidal behavior, we used the magnetic field readings. This signal does not need any inclination of the spin axis and, therefore, the gyroscope is not affected by the gravity effect.

The Earth's magnetic field has been used to obtain a reference of the magnetic north for centuries. Any rotation around the axis orthogonal to the Earth's plane will generate a signal oscillating with a sinusoidal behavior.

Analogously to what we did with the accelerometer, we use the sinusoidal signal to extract the periods of the wheel's turns and, subsequently, its angular rate. Similarly, we pair the estimated angular rate with the raw values measured by the gyroscope.

Facing the advantage of being able to set the wheel's axis, and therefore the gyroscope's axis, orthogonal to the Earth's plane, the magnetometer presents a larger amount of noise which may distort the sinusoidal signal. This noise makes more difficult to identify the peaks of the signal as there might be other local maximums distorting them. To mitigate the effects of the noise, we apply a first-order Butterworth low-pass filter ($\omega_{\text{norm}} = 0.12$) to the magnetometer output. Figures 3.55 and 3.56 depict the measured magnetic field when spinning the MIMU around its X axis. Notice how the unfiltered signal presents local peaks around the maximums.

The rest of the procedure is shown in figures 3.57-3.59, where the latter represents the three estimated calibration lines for each one of the gyroscope's axes. To check the precision of the estimated lines, we rotate the Wagyromag 180° around each one of its axes. Then, the gathered angular rate is calibrated and subsequently integrated to estimate the rotation angle. Figures 3.60-3.62 show the estimated rotation angles. These are 176.5° , 176.7° and 185.2° for X, Y and Z axes respectively. As we can see, the obtained error ranges from 3 to 5 degrees. This error is not exclusively caused by the inaccuracy of the estimated calibration lines, but also by the drift associated to the integration of noise present in the output of the gyroscope

(see 5.2.2).

We can conclude that the proposed method offers an acceptable degree of precision at the same time it removes the need of expensive and complex instrumentation.

The estimated calibration lines are:

$$\begin{aligned}\omega_x &= 12.569u_x - 9619 \text{ deg/s} \\ \omega_y &= 12.518u_y - 9577.2 \text{ deg/s} \\ \omega_z &= 5.1493u_z - 1973.3 \text{ deg/s}\end{aligned}\quad (3.102)$$

and, therefore, the gyroscope range is:

$$\begin{aligned}\omega_x &= [-9169, 3239] \text{ deg/s} \\ \omega_y &= [-9577.2, 3228.7] \text{ deg/s} \\ \omega_z &= [-1973.3, 3294.43] \text{ deg/s}\end{aligned}\quad (3.103)$$

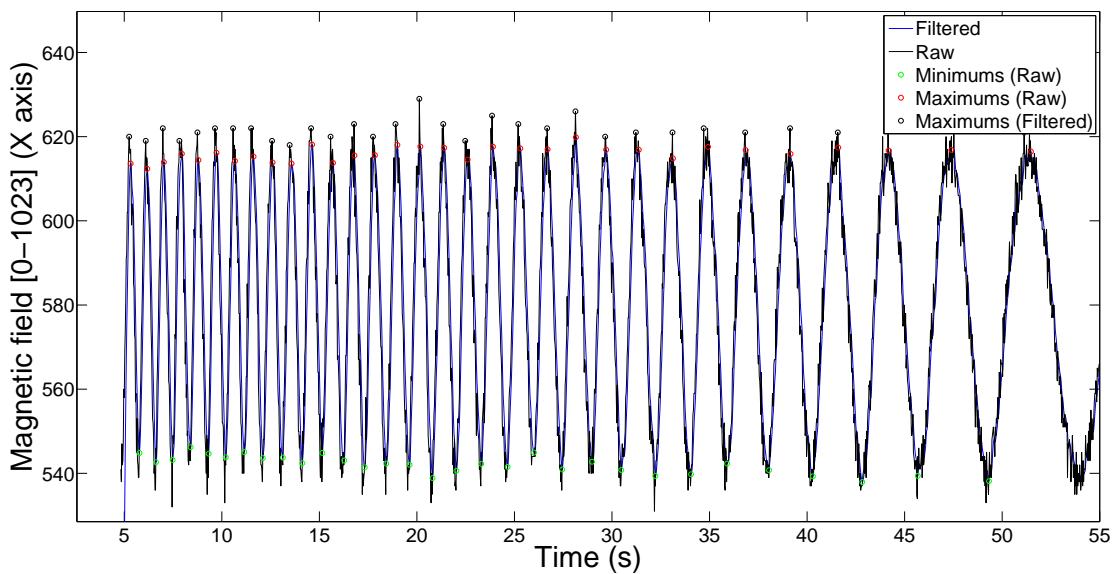


Figure 3.55: Peak detection of the raw magnetometer signal. Distance between peaks defines the turn period.

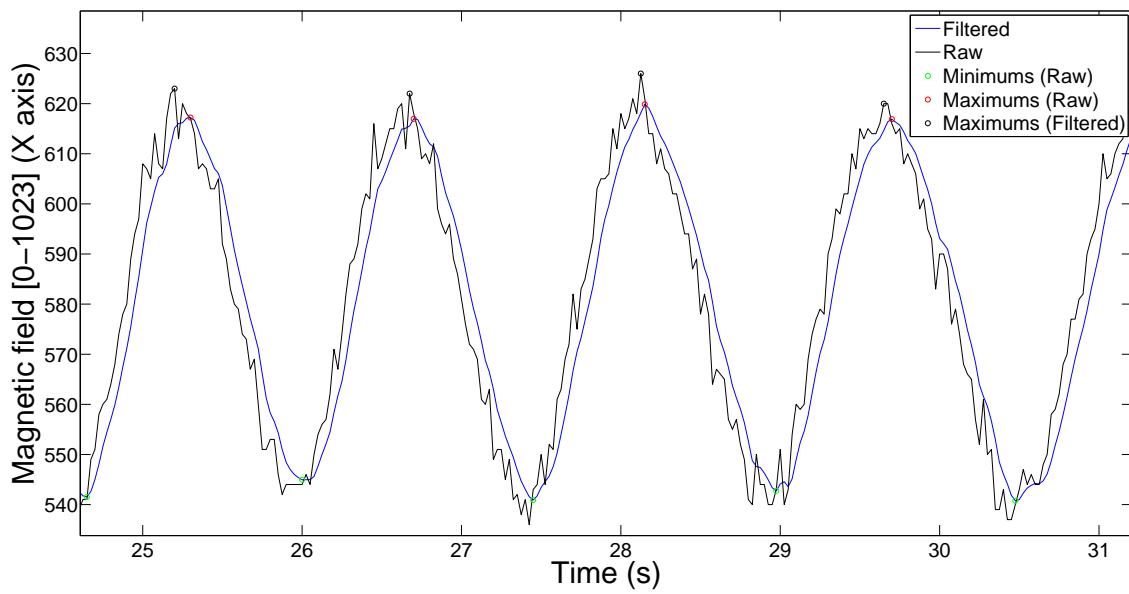


Figure 3.56: Peak detection of the raw magnetometer signal (zoomed). Distance between peaks defines the turn period.

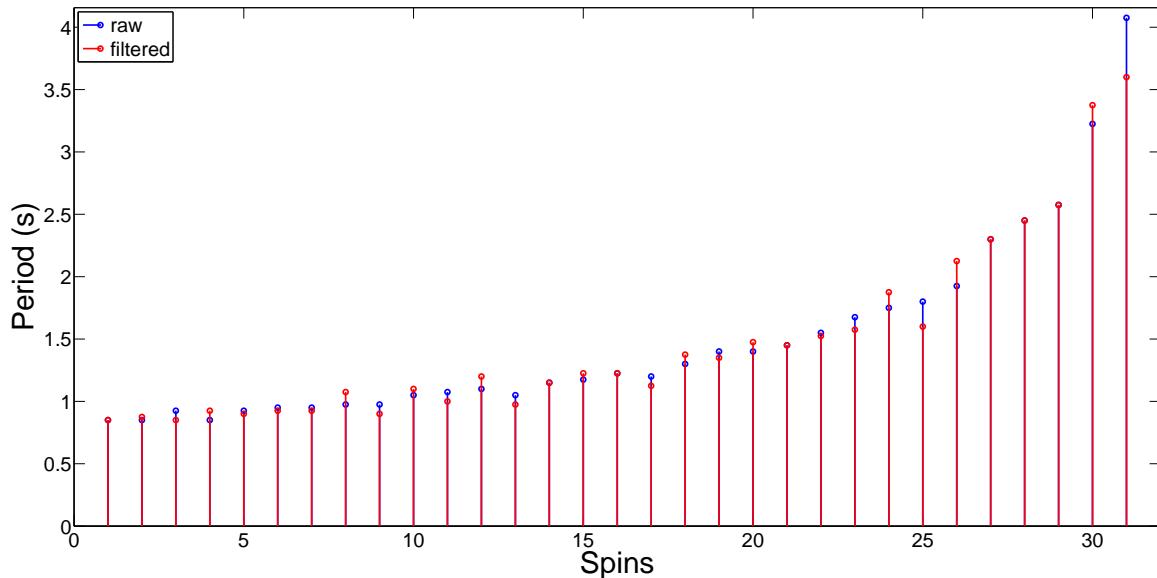


Figure 3.57: Estimation of turn periods using detected peaks.

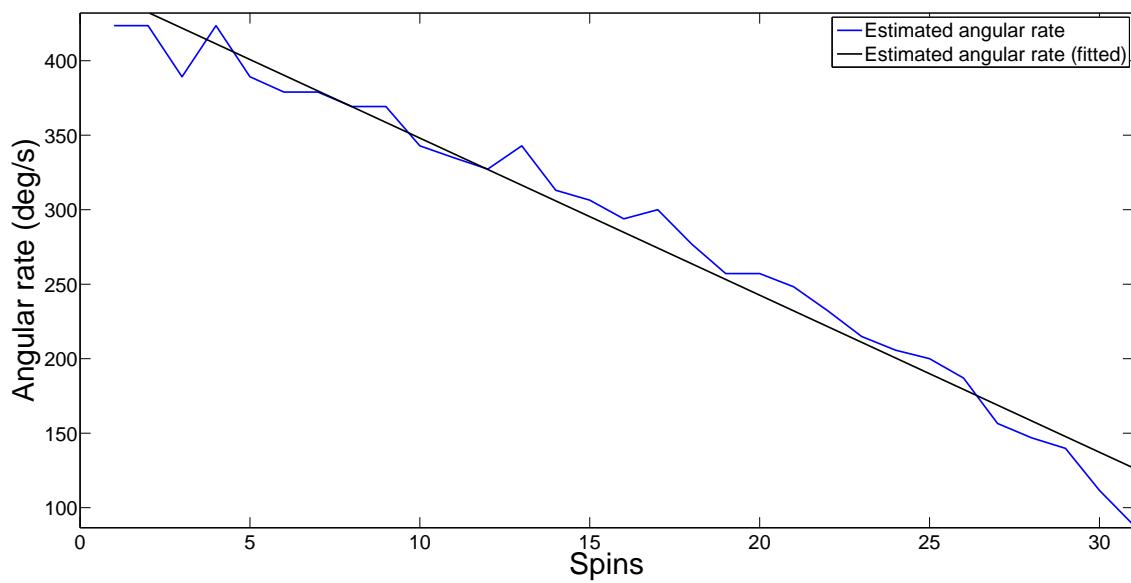


Figure 3.58: Estimation of angular rate using estimated periods.

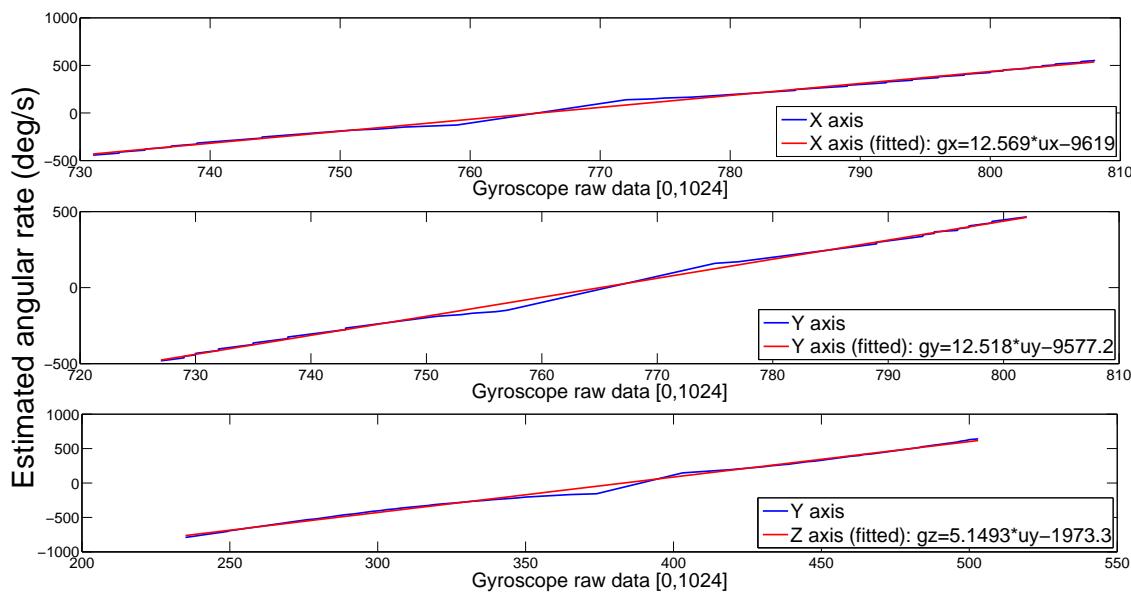


Figure 3.59: Estimated calibration lines for the gyroscope triad.

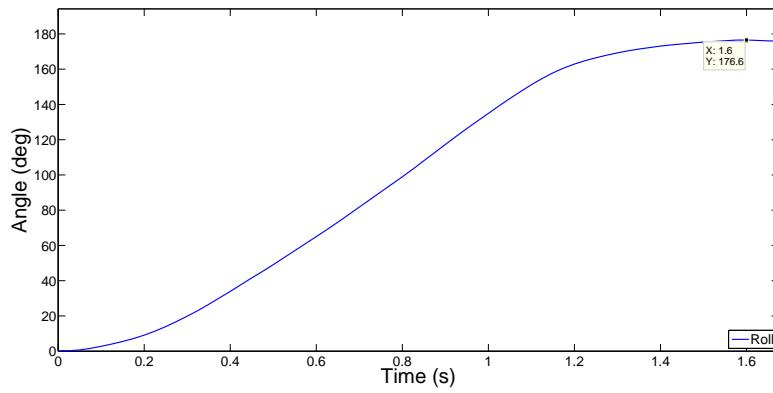


Figure 3.60: Roll computed using estimation lines. Estimated roll is 176.6° . Actual roll is 180° .

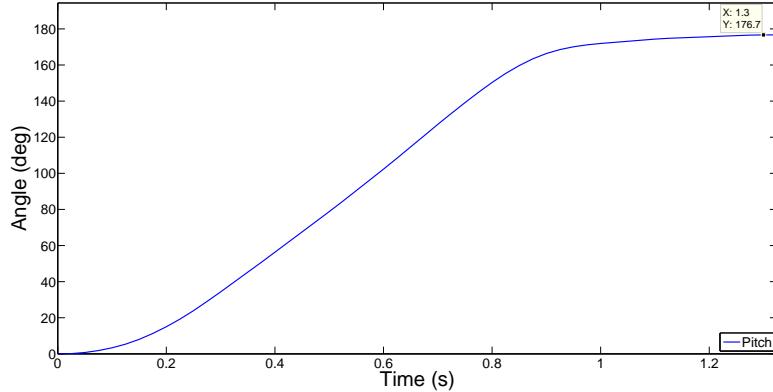


Figure 3.61: Pitch computed using estimation lines. Estimated pitch is 176.7° . Actual pitch is 180° .

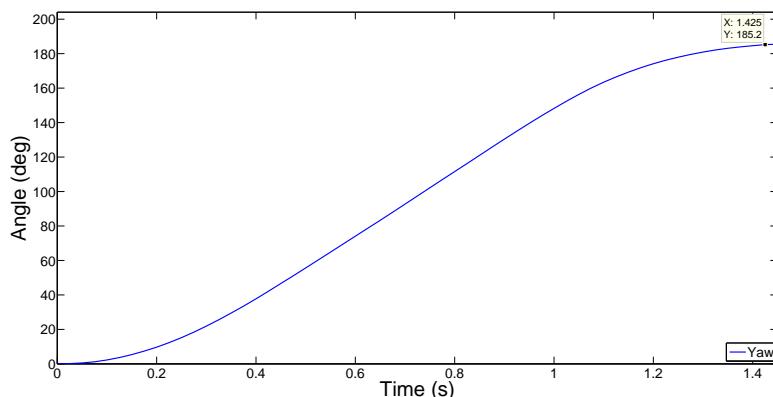


Figure 3.62: Yaw computed using estimation lines. Estimated yaw is 185.2° . Actual yaw is 180° .

3.6 Chapter wrap-up

Along this chapter we carried out a deep analysis of the current techniques to calibrate MARG MEMS sensors. We explained the different kinds of undesired effects and errors, both generic and specific, present in their output, followed by a mathematical framework that models these errors using a set of parameters that are to be estimated to revert the undesired effects.

After setting the theoretical basis, we commented the state of the art of the MEMS MARG calibration algorithms published during the last years. Among them, we selected 12 of the most relevant ones which were explained in depth, including the proposed sensor model, the calibration maneuvers and the algorithm employed to estimate the parameters. We tested afterwards these 12 methods to compare their precision to estimate the calibration parameters and compensate for the errors and undesired effects. To that purpose, we performed a series of experiments that threw a set of results that allowed us to draw some preliminary conclusions about the precision of the studied algorithms as well as their pros and cons.

Finally, we presented an algorithm that supports the data gathering procedure and warrants that the dataset is well distributed in the space which, in turn, increases the precision of multi-point algorithms. This algorithm computes the histograms of both θ and ϕ spherical coordinates of the dataset and enables the parameter estimation procedure only when a predefined limited number of bins are still empty. Using a well distributed dataset increases the precision of the ellipsoid fitting and minimization algorithms.

Moreover, we proposed two novel methods to calibrate the gyroscope without the need of complex and expensive equipment like multi-axis numerical rate tables. These methods use the magnetometer or the accelerometer (if a magnetometer is not included in the measuring device) to determine the angular rate to which each one of the axes is subjected. Such estimated angular rate is used as a reference to obtain the calibration lines mapping raw converting raw gyroscope data into angular rate measured in deg/s.

Chapter **4**

Determining Motion Intensity

4.1 Introduction

Detection of human body movement and inactivity periods is a critical step for motion monitoring applications based on inertial or MARG (Magnetic, Angular Rate and Gravity) sensors. When body movement is being monitored using inertial or MARG sensors their output signals can be used to discriminate periods in which the subject being monitored is moving with different intensities. For example, the distinction between static and active instants is usually imperative for sensor calibration [61] and different motion monitoring applications like inertial navigation [62] and human activity classifiers [138].

Most sensors present random time variations in the parameters of their mathematical model, such as the scale factors or biases [139, 140]. Some works show different techniques to reduce drifts in inertial measurements using Kalman Filtering [141] as well as other adaptive filtering algorithms [142]. Such a drifting behavior requires the periodical recomputation of the model parameters in order to maintain a satisfactory degree of precision during the complete monitoring session [61]. However, we can only recalculate them when there is no acceleration nor angular velocity, for example, when the subject that is carrying them is stationary, since we need to know the zero level noise signal.

Determination of motion intensity, and more specifically, discrimination of (in)-activity periods, is of critical importance in inertial navigation algorithms so the Zero Velocity Updates (ZUPT) [66, 143, 144] can be computed to determine when to reset the offset parameters and perform corrections in order to help avoid erroneous drift in the trajectory of the body [145, 146].

It is also a very important preprocessing step for inertial-based human activity classifiers since it helps to divide the signals into periods that are later analyzed.

Let's take a look now to the already existing approaches and to the new ones that we propose along this chapter.

4.2 Novel approaches

In addition to implementing and testing the five motion intensity detectors mentioned in the introductory chapter, we will complete the comparative study by implementing and testing four more detection methods. The first two are based on the computation of the spectrum (Fourier transform) of the input signals. We will use the Long Term Spectral Detector (LTSD) presented in [147] and a variation that we will refer as to the Framed Spectrum Detector (FSD). Spectrum-based methods have been widely used with success in Voice Activity Detection (VAD) applications [148–150]. By applying such algorithms we aim to find other possible detectors that may outperform those in [66], as they are very robust in conditions of low SNR. The last two methods that we will test are thought to detect abrupt changes in signals coming from sensors located in an industrial environment. These are the Memory-Based Graph Theoretic Detector (MBGTD) and the Memory-Based Cumulative Sum Detector (MBCD) both developed in [151]. Therefore, this chapter also presents for the first time the results of the application of LTSD, FSD, MBGTD and MBCD algorithms in the context of motion intensity detection using MARG sensors.

The rest of the chapter is divided as follows; section 4.3 presents the different detection methods that will be tested in the comparative study; section 4.4 shows both the simulations and the application of the algorithms on real signals; section 4.5 analyzes both results from theoretical and real experiments and compares our results to those obtained in previous studies; section 4.6 draws the conclusions and future evolution of the research.

4.3 Theoretical basis of motion detection algorithms

As said in the introduction, we will be testing nine different methods. These methods can be grouped in three different sets; those based on the magnitude of the

acceleration and/or the angular rate (AMVD, AMD, ARED, SHOD and FRD); those based on the spectrum of the acceleration and the angular rate (LTSD and FSD); those based on abrupt changes in data distributions (MBGTD and MBCD). The following subsections present the mathematical core of each one of the detectors which is essential to program them, that is, we derive the expressions of the figures of merit that are used for the classification. At the end of the section, we have also included the workflow of the algorithms explaining step by step their general structure to ease their understanding and implementation.

4.3.1 Magnitude-based methods

The following methods use the magnitude of the acceleration, the magnitude of the angular velocity or a linear combination of both as the input signal. All the computations are carried out in the time domain of the signals.

4.3.1.1 Acceleration Moving Variance Detector (AMVD)

The AMVD exclusively uses the acceleration signals to carry out the distinction of periods in which motion presents different intensities. A sliding window is applied over the signal in which the variance of the acceleration is computed. The figure of merit of the detection algorithm is computed as follows,

$$V(\mathbf{n}) = \frac{1}{N} \sum_{k=1}^N \|\mathbf{a}_k - \bar{\mathbf{a}}_n\|^2 < \gamma \quad (4.1)$$

where \mathbf{n} is the frame at instant n , i.e. the content of the sliding window at instant n , \mathbf{a}_k is the acceleration vector at instant k , $\bar{\mathbf{a}}_n$ is the mean of the acceleration of the frame at instant n , N is the length of the frame and γ is the predefined threshold that characterizes the decision based on the resultant value of the figure of merit. When wishing to detect various levels of motion intensity, we will need to empirically set different thresholds, each one associated to an specific level of intensity.

4.3.1.2 Acceleration Magnitude Detector (AMD)

The AMD is also solely based on the acceleration signals. The magnitude of the gravity acceleration vector is subtracted from the magnitude of the acceleration vector which is computed at every instant. The figure of merit used as the input of the classifier can be computed as;

$$V(\mathbf{n}) = \frac{1}{\sigma_a^2 N} \sum_{k=1}^N (\|\mathbf{a}_k\| - g)^2 < \gamma \quad (4.2)$$

where g is the magnitude of the gravity acceleration (1 g or 9.8 m/s^2) and σ_a^2 is the variance of the acceleration signal noise that is used as a scaling factor to make the threshold less sensitive to noise.

As with AMVD, and the rest of the methods to be presented, different thresholds can be set to implement a multi-level motion intensity detector.

4.3.1.3 Angular Rate Energy Detector (ARED)

On the other hand, the ARED, uses only the angular rate signals as the input. The squared magnitude of the angular rate vector at each instant is compared with a predefined threshold. This can be expressed in the following way;

$$V(\mathbf{n}) = \frac{1}{\sigma_\omega^2 N} \sum_{k=1}^N \|\omega_k\|^2 < \gamma \quad (4.3)$$

where ω_k is the angular rate vector at instant k and σ_ω^2 is the variance of the angular rate noise signal, which is also used as a scaling factor.

4.3.1.4 Stance Hypothesis Optimal Detector (SHOD)

The SHOD uses both acceleration and angular rate signals. Its goal is to increase the precision of the previous detectors by taking into consideration that there might be instants where human body movement presents angular rate but no acceleration and vice versa. The figure of merit used as the input of the classifier is;

$$V(\mathbf{n}) = \frac{1}{N} \sum_{k=1}^N \left(\frac{1}{\sigma_a^2} \left\| \mathbf{a}_k - g \frac{\bar{\mathbf{a}}_n}{\|\bar{\mathbf{a}}_n\|} \right\|^2 + \frac{1}{\sigma_\omega^2} \|\omega_k\|^2 \right) < \gamma \quad (4.4)$$

4.3.1.5 Filtered Rectifier Detector (FRD)

The FRD was developed by Veltink [67] as a preprocessing step for a simple classifier of Activities of Daily Life (ADL). The operating principle of the detector is very simple; first, the frame of the input signal is high-pass filtered, then rectified and finally low-pass filtered. Therefore, its figure of merit can be expressed as;

$$V(\mathbf{n}) = LPF \{RECT [HPF(\mathbf{n})]\} < \gamma \quad (4.5)$$

In their work the tangential acceleration is used as the input of the detector but, as seen later, the performance of the detector can be improved by using other inputs such as the magnitude of the acceleration, the magnitude of the angular rate, or a

linear combination of both.

More information about the definition and statistical performance of AMVD, AMD, ARED and SHOD can be found in [66].

4.3.2 Spectrum-based methods

Motion intensity can be easily identified by inspecting the frequency of inertial signals. Therefore, instead of using the time domain to detect possible transitions from inactivity to activity and viceversa, now the detectors operate in the frequency domain of the input signals.

4.3.2.1 A new approach based on the Long Term Spectral Detector (LTSD)

The LTSD computes the Long Term Spectral Envelope (LTSE) of the signal. Let $x(k)$ be the sensor signal which is segmented into frames with a certain degree of overlapping and $X(l, n)$ its amplitude spectrum for the l band at frame \mathbf{n} . The N -order long-term spectral envelope can be computed as;

$$\text{LTSE}_N(l, n) = \max\{X(l, n+j)\}_{j=-N}^{j=+N} \quad (4.6)$$

The figure of merit used as the input of the classification process for each frame can be obtained by applying;

$$V(\mathbf{n}) = 10 \log_{10} \left(\frac{1}{N_{\text{FFT}}} \sum_{l=0}^{N_{\text{FFT}}-1} \frac{\text{LTSE}^2(l, n)}{N^2(l)} \right) < \gamma \quad (4.7)$$

where $N_{\text{FFT}} = 512$ in our case, is the resolution of the Fast Fourier Transform and $N(l)$ is the average noise spectrum magnitude for the band l , ($l = 0, 1, \dots, N_{\text{FFT}} - 1$). For further information about the definition of the LTSD see [147].

4.3.2.2 A new approach based on the Framed Spectrum Detector (FSD)

The FSD is similar to the LTSD, but instead of computing the Long Term Spectral Envelope, it uses the spectrum of each frame in which the input signal is divided. Its expression is as follows;

$$V(\mathbf{n}) = 10 \log_{10} \left(\frac{1}{N_{\text{FFT}}} \sum_{l=0}^{N_{\text{FFT}}-1} \frac{X^2(l, n)}{N^2(l)} \right) < \gamma \quad (4.8)$$

where again, N_{FFT} is the resolution of the Fast Fourier Transform, $N(l)$ is the average noise spectrum magnitude for the band l and $X(l, n)$ is the spectrum of the input signal for the band l at frame n .

4.3.3 Memory-based methods

4.3.3.1 A new approach based on the Memory-based Theoretic Graph Detector (MBGTD)

The MBGT algorithm, presented in [151], is based on computing the distance between two distributions, which are indirectly specified by means of two sample sets. Consider that we have a buffer which is filled with the last N sensor readings. Instead of splitting the sample buffer into two equal parts, and testing for difference between them, the MBGT algorithm considers all possible pairs of indices (i, j) , such that $1 \leq i < j \leq N$, which split the sample frame into two adjacent windows $\alpha_{i,j-1}$ and $\alpha_{j,N}$, where $\alpha_{a,b} = \{x_a, x_b\}$ is a window that contains all samples from index a to index b , i.e. the starting points of the first and second window respectively. Once we have divided the frame into two sub-windows, we can compute the average Euclidean distance between two points included in the pair (i,j) applying the following expression;

$$C_{i,j} = \frac{\sum_{k=i}^{j-1} \sum_{l=j}^N d_{k,l}}{(j-i)(N-j+1)} \quad (4.9)$$

where $d_{k,l}$ is the Euclidean distance between points k and l .

The overall figure of merit of the detection algorithm is the maximum $C_{i,j}$ computed over all the possible frame splits that is;

$$V_{MBGTD} = \max_{1 \leq i < j \leq N} C_{i,j} < \gamma \quad (4.10)$$

Further information about the algorithm, and how to implement it so its complexity is suitable for practical applications, can be found in [151].

4.3.3.2 A new approach based on the Memory-based Cumulative Sum Detector (MBCD)

The Memory-Based Cumulative Sum Detector, also presented in [151], is a variation of the well-known Cumulative Sum (CUSUM) algorithm first proposed in [152] and explained in depth in [153]. The CUSUM algorithm accumulates the log-likelihood of the current reading with respect to the distributions, before ($p_{\theta_0}(x_i)$) and after ($p_{\theta_1}(x_i)$), which is the hypothesized change point. The procedure is as

follows;

$$\begin{aligned} g_k &= S_k - m_k, \quad S_k = \sum_{i=1}^k s_i \\ s_i &= \ln \frac{p_{\theta_1}(x_i)}{p_{\theta_0}(x_i)}, \quad m_k = \min_{1 \leq j \leq k} S_j \end{aligned} \quad (4.11)$$

The change point will be detected when $g_k \leq h$ where h is an empirically set threshold.

However, the CUSUM algorithm can only be applied when both the distributions $p_{\theta_0}(x_i)$ and $p_{\theta_1}(x_i)$ are known. The MBCD solves this drawback by estimating both pre-change and post-change distributions via Parzen kernel density estimates [154] by using the following expression (included here for clarity),

$$\hat{p}_\theta(x_k) = \frac{1}{N(2\lambda^2\pi)^{\frac{1}{2}}} \sum_{i=1}^N e^{\frac{1}{2}(\|x_i - x_k\|/\lambda)^2} \quad (4.12)$$

where λ stands for the standard deviation of the distribution and N is the total number of samples included in the frame.

Once we know how to estimate the distributions of each one of the sub-windows, we can proceed to compute the log-likelihood ratio as follows;

$$S_{i,j} = \sum_{l=j}^N \log \frac{\frac{1}{N-j+1} \sum_{k=j}^N w_{l,k}}{\frac{1}{j-i} \sum_{k=i}^{j-1} w_{l,k}} \quad (4.13)$$

where $w_{l,k}$ is a kernel weight for the pair of samples (x_l, x_k) computed using Parzen's approximation.

The general figure of merit of the algorithm is;

$$V_{MBCD} = \max_{1 \leq i < j \leq N} S_{i,j} < \gamma \quad (4.14)$$

Further information about the algorithm and how to reduce its computational complexity can also be found in [151].

4.3.4 Workflow of the algorithms

All the presented algorithms have been implemented following the structure which is explained by the steps mentioned below.

- (I) Set input parameters of the algorithm.

- (II) Algorithm starts a swipe, using a sliding window, through the input signal.
- (III) For every signal frame, compute the resultant value of the figure of merit by applying equations (4.1-4.5), (4.7), (4.8), (4.10), (4.14) accordingly. Save computed value in a vector that grows in length with the iterations.
- (IV) Compare figures of merit obtained for each one of the aforementioned methods with the predefined threshold. If an (in)activity detector is being implemented, for every instant k , if the value of the figure of merit is lower than the threshold, we will mark the instant as "static" and a 0 is added to a marker vector. On the other hand, if the value of the vector is equal or higher than the threshold, the instant will be marked as "active" and a value of 1 is added to the marker vector. At the end of the application of every algorithm we will have a binary marker vector what will be used for performance evaluation. In addition, if a multi-level intensity detector is being implemented, a m -ary vector will be computed.

Finally, the following list clarifies the corresponding inputs and outputs of every presented algorithm:

- Input:
 - Signal to be analyzed:
 - * Acceleration (X, Y and Z axes): AMVD and AMD.
 - * Angular Rate (X, Y and Z axes): ARED.
 - * Acceleration and Angular rate (X, Y and Z axes): SHOD.
 - * Flexible input: FRD, LTSD, FSD, MBGTD and MBCD.
 - Window length (size of sliding window): AMVD, AMD, ARED, SHOD, LTSD, FSD, MBGTD and MBCD.
 - Threshold (empirically predefined): AMVD, AMD, ARED, SHOD, FRD, LTSD, FSD, MBGTD and MBCD.
 - Shift (sliding window overlapping): LTSD and FSD.
- Output:
 - Figure of merit.
 - Binary activity marker (computed by comparing the figure of merit with the predefined threshold).

4.4 Experiments

Once the detectors are implemented we proceed to design a comparative study that computes different statistic parameters to determine the performance of each algorithm. Such a comparative study is divided in two parts; the first part includes simulations derived from the application of the detectors on a large set of synthesized signals; the second part aims to complete the study by applying the algorithms on real datasets gathered from inertial sensors.

4.4.1 Simulations

The main goal of the theoretical simulations is to apply the algorithm over a very large set of signals since this will allow the computed performance parameters to have statistic significance. Specifically, we will be calculating the Accuracy and Correlation coefficient of the resultant activity marker with respect to the actual activity marker. The actual markers can be obtained by visually inspecting each one of the gathered acceleration and angular rate signals and hand labeling the starting and ending points of each activity period. This should be done by averaging the observed starting and ending points. Due to the cumbersomeness and almost impracticality of carrying out such a procedure over a large set of signals, we decided to design a synthesizer that is able to mimic signals coming out of an accelerometer and a gyroscopic sensor. The synthesizer is designed not only to avoid the hand-labeling procedure but to be able to generate large data sets as gathering many real signals is very time consuming. Therefore, the synthesizer will also generate the marker with the actual starting and ending points of each activity period so we do not have to label them manually.

4.4.1.1 Set-up

At the start of the simulations we need to generate the synthetic signals and to that purpose we use the signal synthesizer. The signal synthesizer has been built to generate acceleration-like and angular rate-like signals coming from five different basic activities; walking, sitting on a chair and standing up, lying on a bed and standing up, running and jumping. Two more general activities have been implemented. The first one includes no acceleration and shows a constant angular rate and the second one includes no angular rate and shows a constant acceleration period. Although this may look like an unrealistic activity, there exist instants of time where this may happen. Thus, we have included them to ensure that the detector is as much robust as possible. The intensity of each activity, i.e. frequency, amplitude, and also the length of each activity period are set randomly every time the synthe-

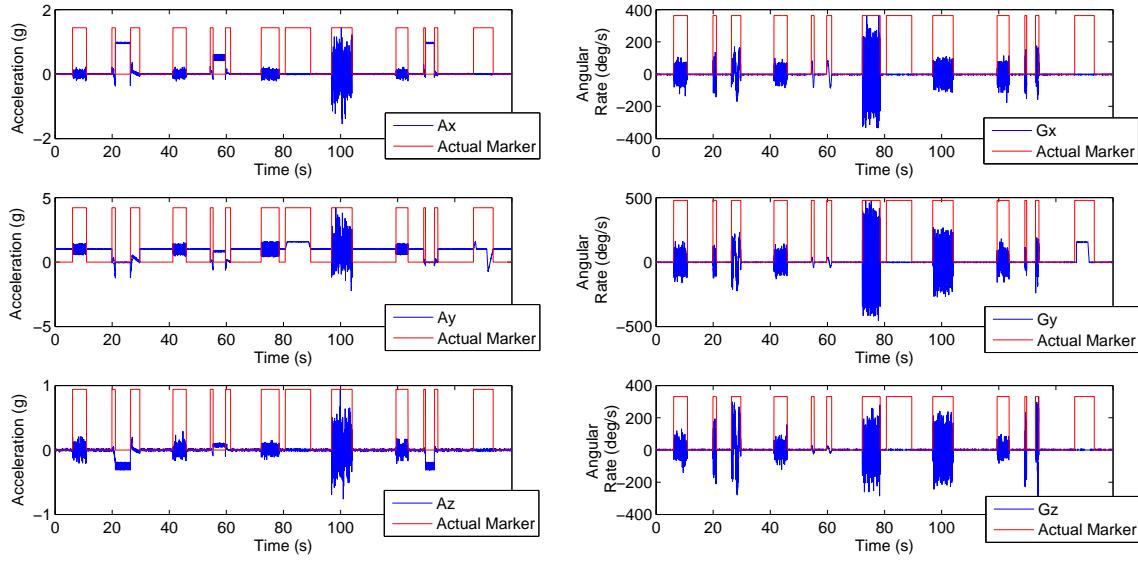


Figure 4.1: Acceleration, angular rate synthesized signals and activity marker. Activity sequence: walking, lying-standing up, walking, sitting-standing up, running, no angular rate, jumping, walking, lying-standing up, no acceleration.

sizer is called. The sensing axis that we want to be parallel to the gravity vector can also be set. In addition, random noise is added to the signals in order to get a better approximation of real sensor signals. The magnitude of the acceleration is set to be 1 g at every static instant. The acceleration and angular rate ranges can also be set according to the level of expected intensity of the exercise that we are simulating, and also to simulate similar ranges to those of commercial MEMS inertial sensors.

Figure 4.1 shows the synthesized signals for the following activity sequence; walking, lying down-standing up, walking, sitting down-standing up, running, no angular rate, jumping, walking, lying down-standing up, no acceleration. The depicted binary activity marker is normalized to the signal magnitude to allow visibility.

4.4.1.2 Monte Carlo simulation

Once the signal synthesizer is set, a Monte Carlo simulation of N repetitions can be performed. At every repetition a new set of signals is synthesized. Then, an optimization routine based on a grid search procedure is called for every algorithm. We opted for a grid search procedure since we observed that convergence of the objective functions depended highly on the initial values of the parameters to be optimized and this was causing the optimizers to stop in local minima, when not diverge, that were far from the optimal values. The optimizer performs a sweep

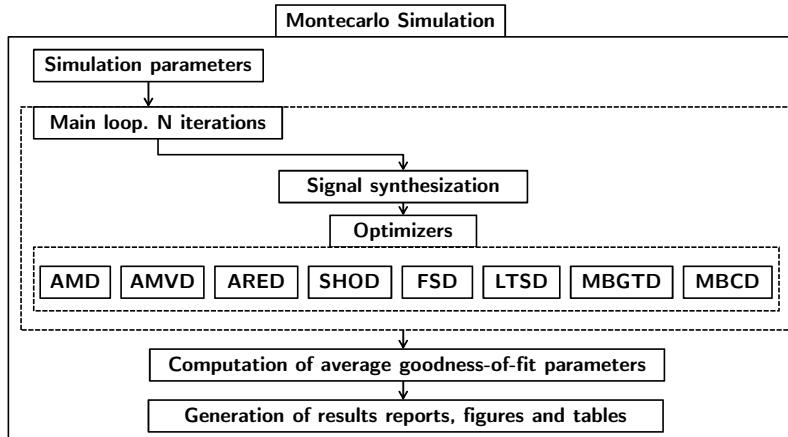


Figure 4.2: Theoretical simulation diagram. A Monte Carlo simulation is performed to ensure statistical robustness.

through the different parameters of each method, for example, window length and threshold for magnitude-based and memory-based methods and window length, frame shift and threshold for the spectrum-based methods. For every parameter configuration, the accuracy and correlation coefficient are computed. After the sweep, we extract the maximum values of the statistics and also store the value of the parameters for which they maximize. At the end of the Monte Carlo simulation, we obtain the average value of every statistic and the average value of the optimal configuration parameters for each one of the eight methods. Figure 4.2 depicts a diagram showing the steps to be followed during the theoretical simulation.

Spectrum-based and memory-based methods can be computed using different combinations of sensor inputs. We have used four different combinations; the magnitude of the acceleration; the magnitude of the angular rate; and the sum and product of both acceleration and angular rate magnitudes. Proceeding this way, we will be able to determine which of the sensor combinations offers the best performance.

The following subsection presents the results coming from a Monte Carlo simulation of $N=500$ iterations.

4.4.1.3 Results

All the nine methods have been tested using a set composed of 500 unique synthetic signals. Figures 4.4-4.11 show the average Accuracy values obtained from the optimization procedure for each one of the nine methods, when sweeping values of the input parameters, searching for maximum accuracy. The depicted surfaces may

help to identify the optimal range of the input parameters and convergence points, as well as to identify the weight of each parameter in the computation of the performance indicator.

Tables 4.1-4.6 show the average Accuracy and Correlation coefficient values, as well as the associated parameter values, for each one of the detection methods put into the Monte Carlo simulation of N=500 runs. The optimal value of each parameter is highlighted in bold characters for every performance indicator.

The suffixes *-Acc* (magnitude of the acceleration), *-Ang* (magnitude of the angular rate), *-SUM* (sum of the acceleration and angular rate magnitudes) and *-PROD* (product of the acceleration and angular rate magnitudes), indicate the algorithm input signal.

Table 4.1: Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Magnitude methods without flexible input).

	AMVD	AMD	ARED	SHOD
Accuracy	0.8741±0.0181	0.9641±0.0087	0.9431±0.0136	0.9817±0.0124
Correlation coeff.	0.7137±0.0360	0.9205±0.0175	0.8752±0.0270	0.9592±0.0269
Window length	26.065±1.1011	96.9560±9.8260	21.9300±1.5849	10.7556±1.3007
Threshold	0.0188±0.0049	0.0008±0.0002	2.3712±2.1621	1.3426±0.2898

Table 4.2: Results of the Monte Carlo simulation (N=500). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Framed Spectrum Detector).

	FSD-Acc.	FSD-Ang.	FSD-SUM.	FSD-PROD.
Accuracy	0.9395±0.0155	0.9344±0.0153	0.9330±0.0351	0.9470±0.0441
Correlation coeff.	0.8639±0.0331	0.8534±0.0311	0.8520±0.058906	0.8835±0.0788
Window length	18.0836±1.8278	17.3496±1.8112	13.0636±0.9983	9.5292±1.7607
Threshold	2.6441±0.6272	13.8640±3.0741	10.4764±2.6726	8.3344±2.8890
Shift	15.1304±2.2318	16.9824±0.9055	13.1048±0.4916	7.2364±2.9423

Table 4.3: Results of the Monte Carlo simulation ($N=500$). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Long Term Spectral Detector).

	LTSD-Acc.	LTSD-Ang.	LTSD-SUM.	LTSD-PROD.
Accuracy	0.9252 ± 0.0179	0.9150 ± 0.0926	0.9355 ± 0.0138	0.9318 ± 0.0884
Correlation coeff.	0.8328 ± 0.0388	0.8209 ± 0.15256	0.8556 ± 0.0282	0.8585 ± 0.1266
Window length	14.6012 ± 0.7580	5.2376 ± 0.8156	3.3848 ± 0.2816	11.3200 ± 0.3639
Threshold	4.9140 ± 1.7976	17.3252 ± 3.0170	16.1172 ± 3.6251	8.7112 ± 1.7633
Shift	1.6348 ± 0.7648	1.3936 ± 0.5699	1.2744 ± 0.5312	1.7776 ± 0.7616

Table 4.4: Results of the Monte Carlo simulation ($N=500$). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based Graph Theoretic Detector).

	MBGTD-Acc.	MBGTD-Ang.	MBGTD-SUM.	MBGTD-PROD.
Accuracy	0.9243 ± 0.0139	0.9114 ± 0.0179	0.9115 ± 0.0179	0.9349 ± 0.0159
Correlation coeff.	0.8295 ± 0.02813	0.8040 ± 0.0345	0.8041 ± 0.0345	0.85312 ± 0.03532
Window length	12.8088 ± 3.4643	5.9932 ± 3.4687	5.9900 ± 3.4701	9.7260 ± 3.7743
Threshold	1.1286 ± 0.7215	84.9760 ± 79.7386	84.8680 ± 79.275	151.8720 ± 81.4212

Table 4.5: Results of the Monte Carlo simulation ($N=500$). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based CUSUM Detector).

	MBCD-Acc.	MBCD-Ang.	MBCD-SUM.	MBCD-PROD.
Accuracy	0.9257 ± 0.0145	0.9098 ± 0.0180	0.9100 ± 0.0180	0.9373 ± 0.0168
Corr. coeff.	0.8339 ± 0.0289	0.80002 ± 0.034774	0.80066 ± 0.034845	0.85877 ± 0.037214
W. length	8.4584 ± 3.1602	7.6108 ± 2.4729	6.0644 ± 2.1265	11.1192 ± 3.2323
Threshold	$1.749e-6 \pm 5.389e-7$	0.1117 ± 0.0179	0.0925 ± 0.0182	0.1068 ± 0.0267

Table 4.6: Results of the Monte Carlo simulation ($N=500$). Synthesized signals. Average Accuracy, Correlation coefficient and associated parameters (Filtered Rectifier Detector).

	FRD-Acc.	FRD-Ang.	FRD-SUM.	FRD-PROD.
Accuracy	0.7921 ± 0.0178	0.7608 ± 0.0207	0.7610 ± 0.0207	0.7944 ± 0.0223
Correlation coeff.	0.5228 ± 0.0389	0.4823 ± 0.0464	0.4825 ± 0.0463	0.5362 ± 0.0509
Threshold	0.0100 ± 0.0070	0.1720 ± 0.2862	0.1780 ± 0.2990	0.1680 ± 0.2777

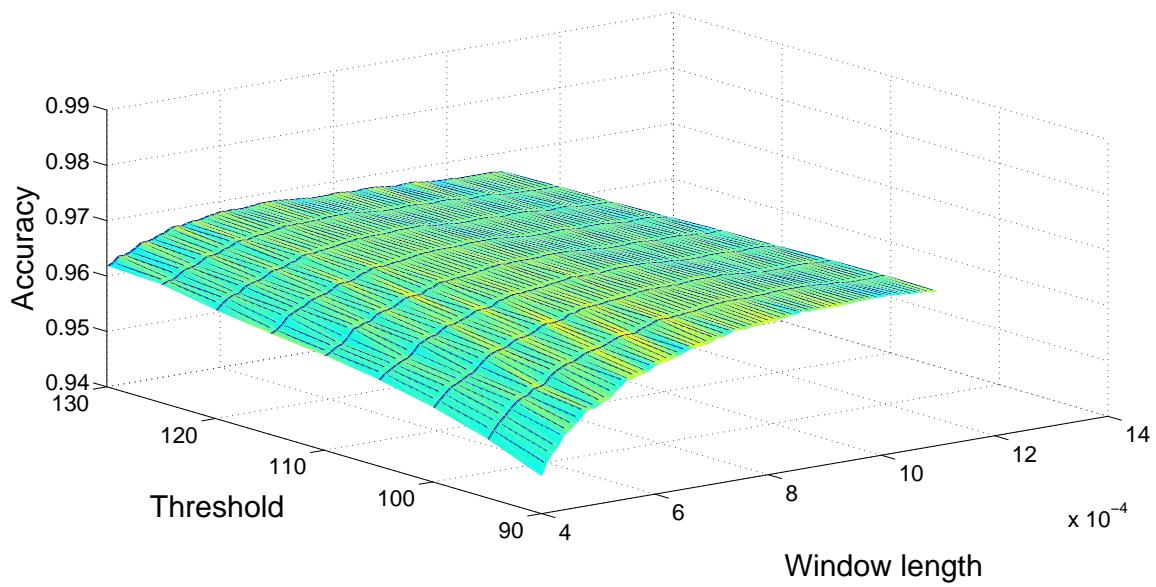


Figure 4.3: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (AMD).

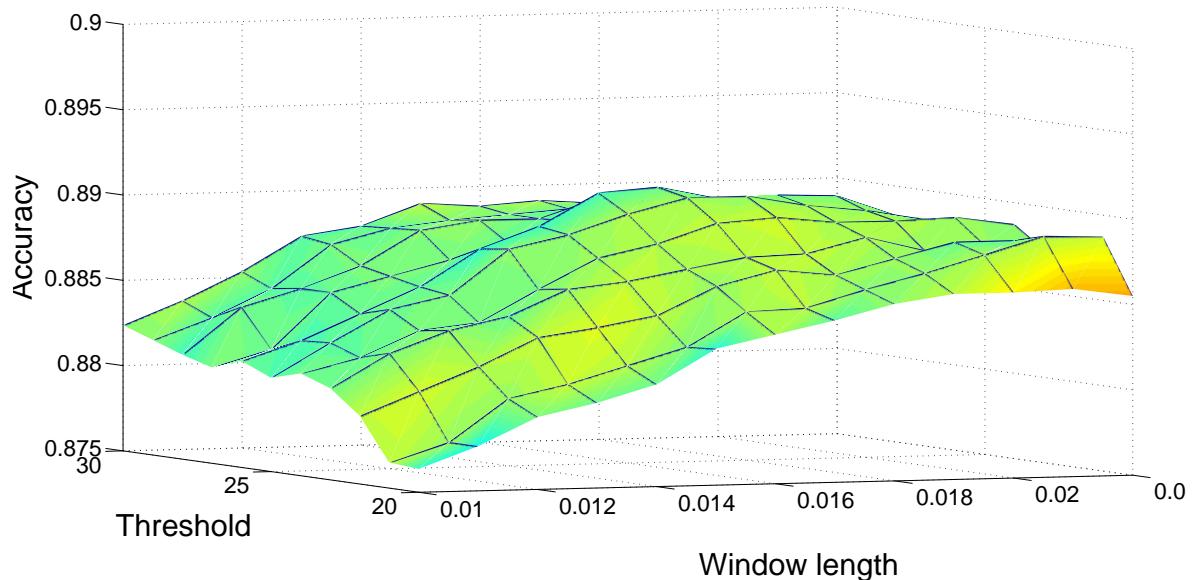


Figure 4.4: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (AMVD).

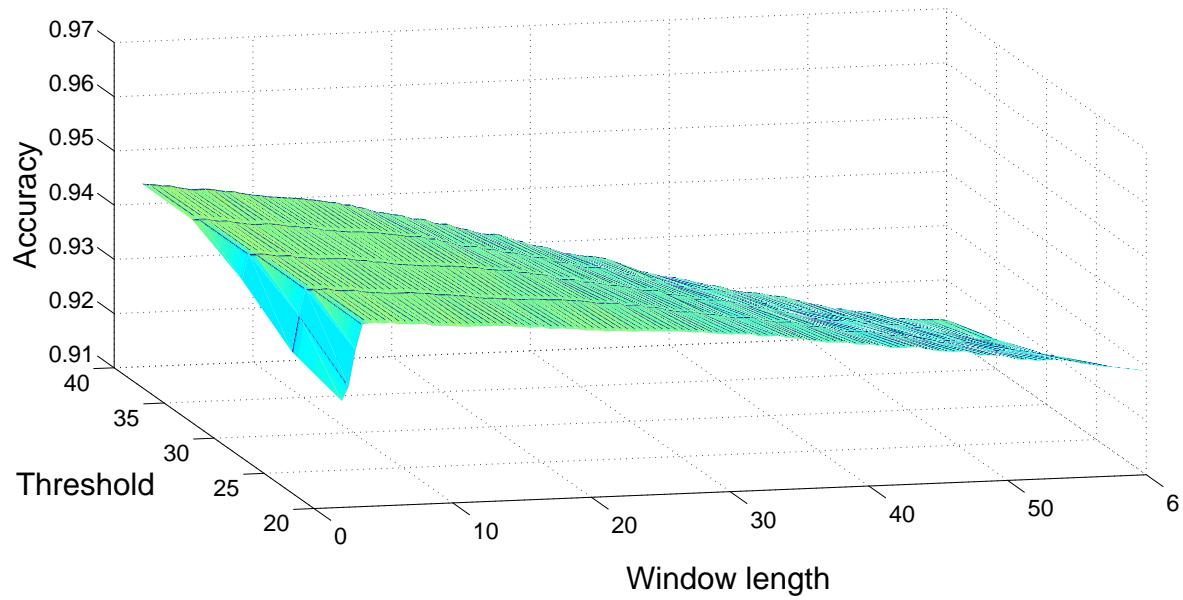


Figure 4.5: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (ARED).

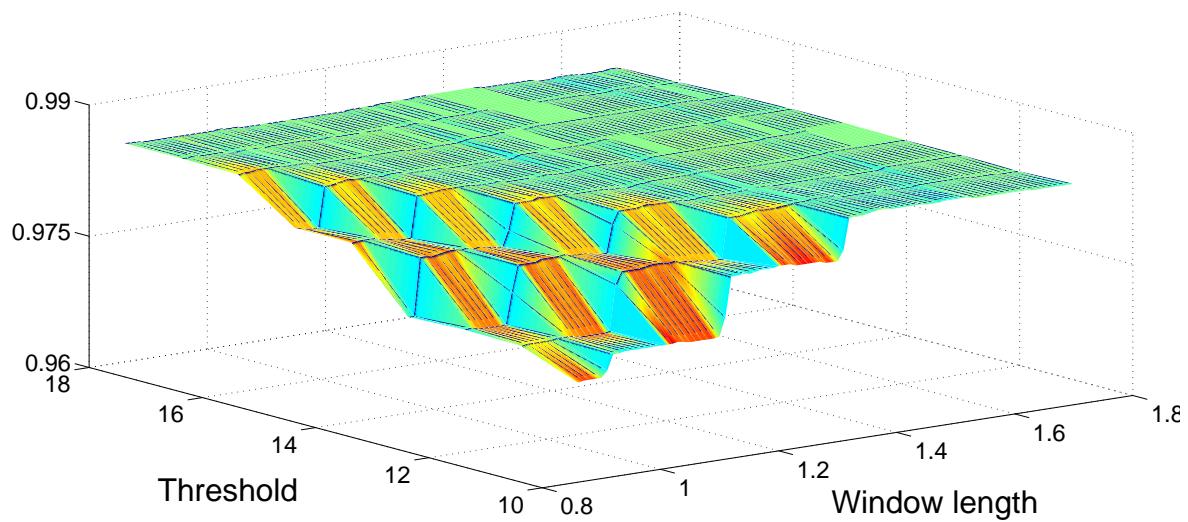


Figure 4.6: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (SHOD).

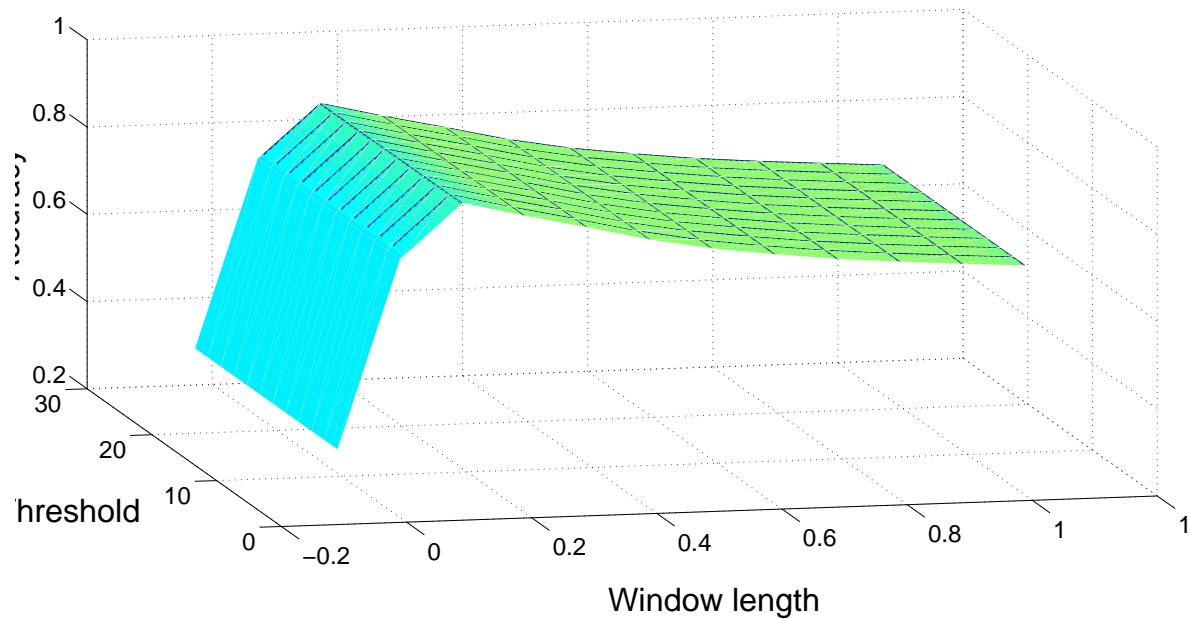


Figure 4.7: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (FRD-PROD).

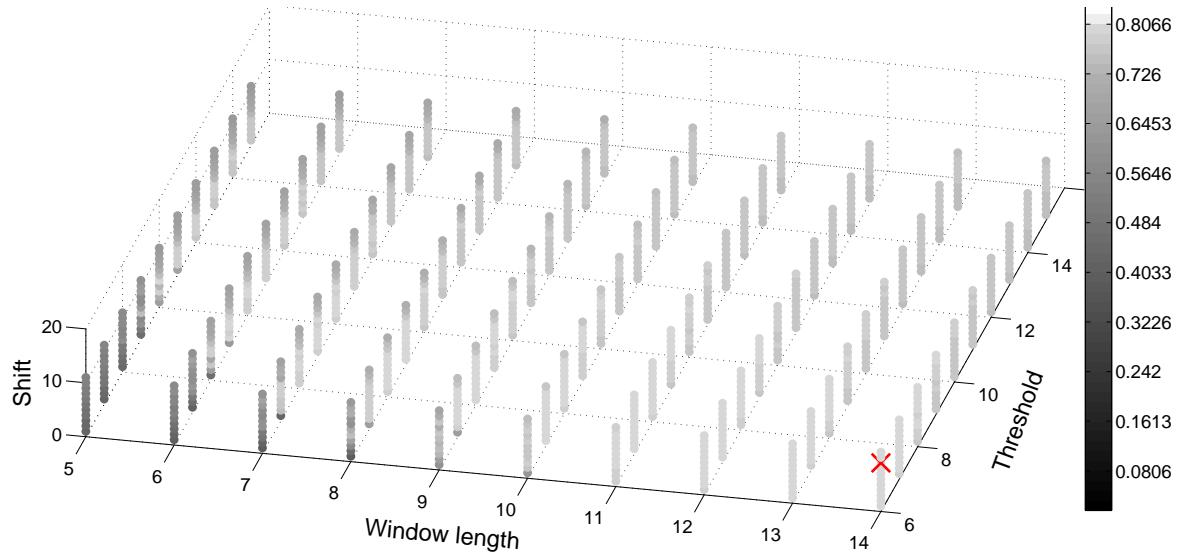


Figure 4.8: Parameter optimization (synthetic signals). Sweep of window length, shift and threshold values to find maximum accuracy (FSD-PROD). The red mark indicates the optimal point of operation.

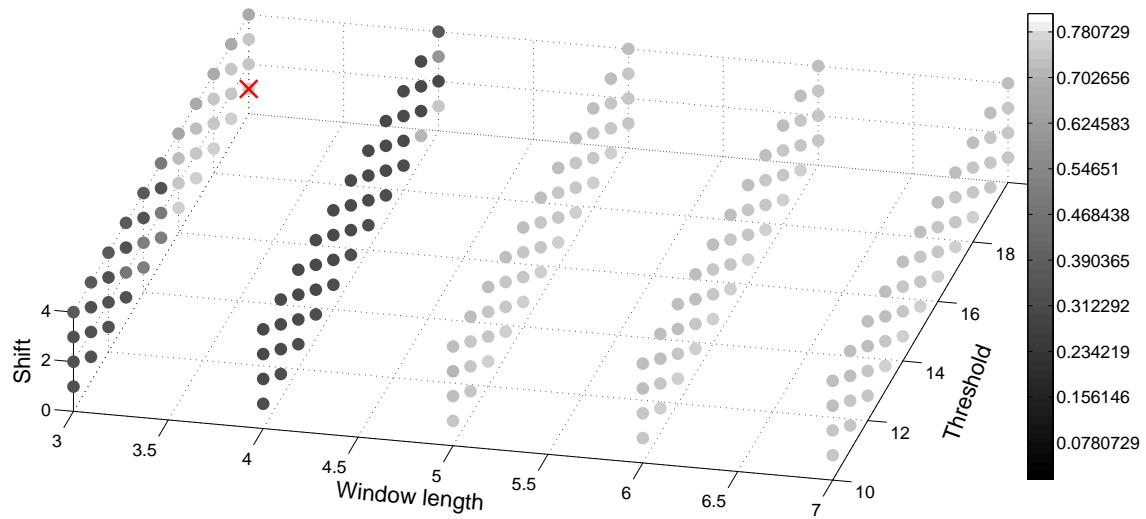


Figure 4.9: Parameter optimization (synthetic signals). Sweep of window length, shift and threshold values to find maximum accuracy (LTSD-SUM). The red mark indicates the optimal point of operation.

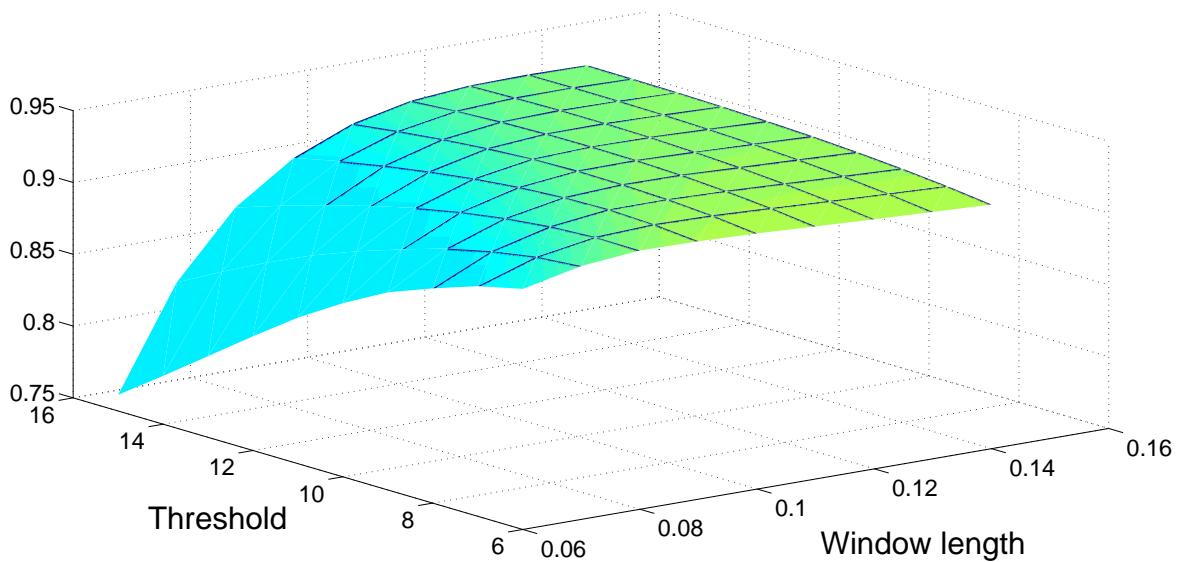


Figure 4.10: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (MBCD-PROD).

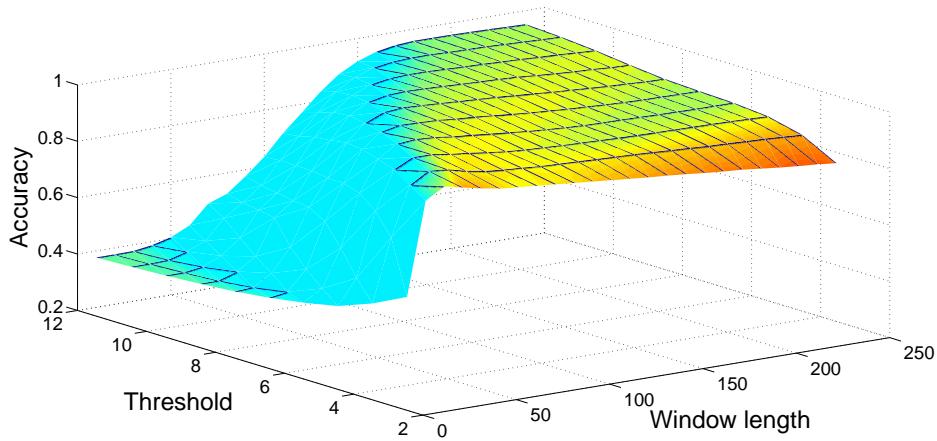


Figure 4.11: Parameter optimization (synthetic signals). Sweep of window length and threshold values to find maximum accuracy (MBGTD-PROD).

4.4.2 Real datasets

Once the algorithms have been tested over a large set of synthetic signals, we need to confirm that the obtained results are also valid for real signals. To do so we have gathered a set of signals using two Wagyromag (remember section 2.2.1.2) units.

4.4.2.1 Set-up

Four male healthy subjects (179.33 ± 4.04 cm, 72.33 ± 7.09 kg, 25 ± 1 years), wearing two Wagyromag units placed at the hip and the ankle respectively, followed twice a circuit composed of the following activities; walk 20 m, stop, sit down-stand up, stop, run 20 m, stop, jump 5 times, stop, and lie down-stand up. A total of 96 signals were gathered (3 acceleration axes + 3 angular rate axes) $\times 2$ MIMUs $\times 4$ subjects $\times 2$ runs) and used as the input for all detection algorithms.

The (in)activity markers were set manually by visually inspecting the gathered signals.

4.4.2.2 Optimization of parameters

Like in the theoretical simulations, an analogous optimization procedure was carried out using the real dataset in order to obtain the average maximum Accuracy and Correlation coefficient values and their associated algorithm configuration pa-

rameters. By doing this we aimed to verify those results previously obtained from the theoretical simulation and check for possible differences.

Figures 4.12-4.20 show the average output of the optimization process when maximizing the accuracy and using the MBGTD.

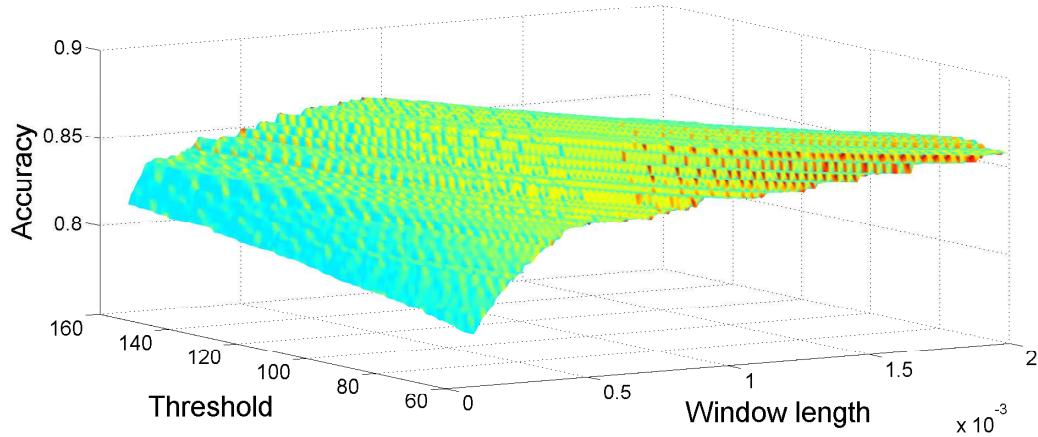


Figure 4.12: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (AMD).

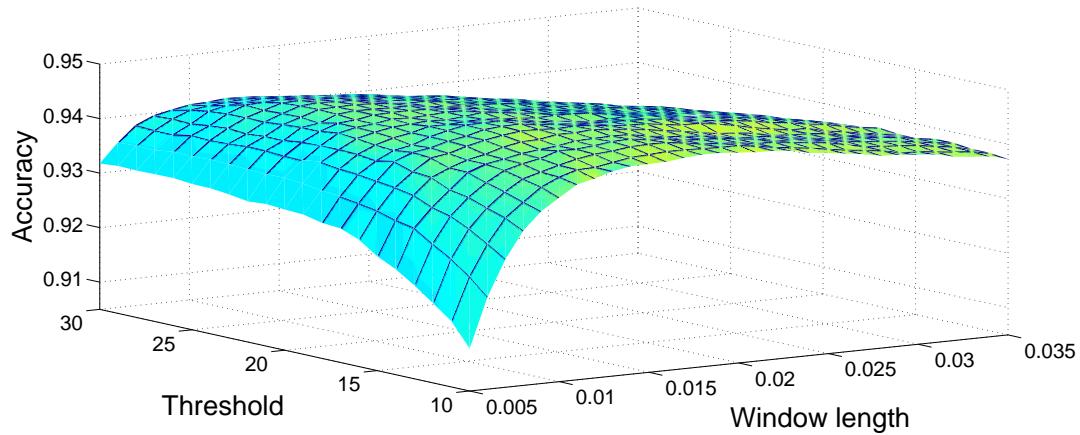


Figure 4.13: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (AMVD). Real signals.

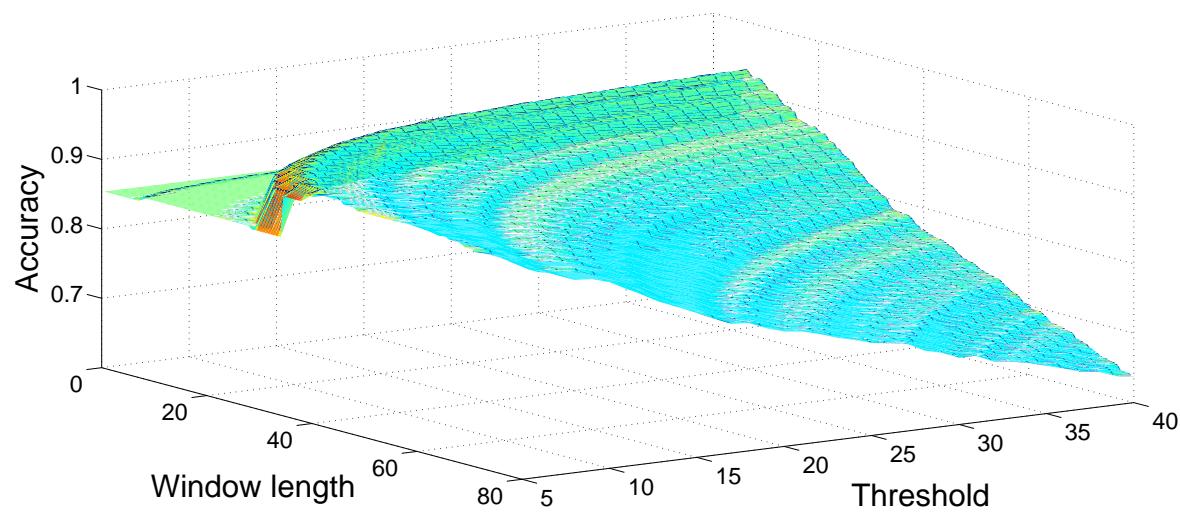


Figure 4.14: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (ARED). Real signals.

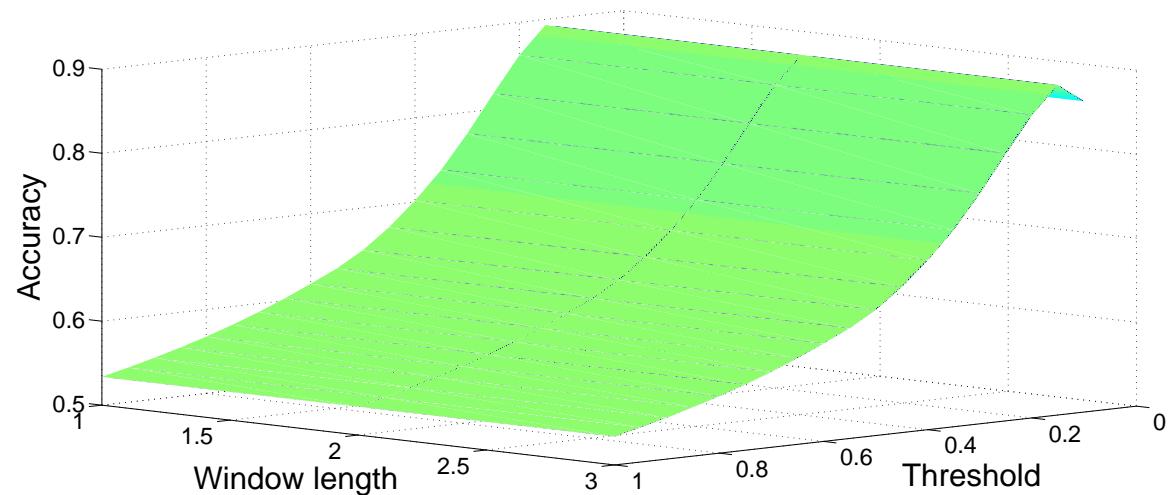


Figure 4.15: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (FRD-PROD). Real signals.

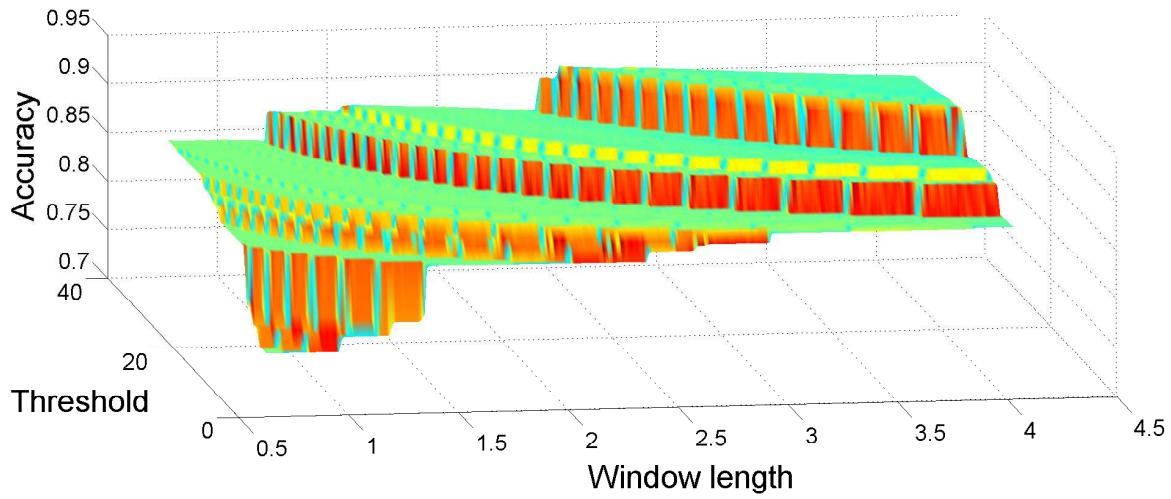


Figure 4.16: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (SHOD). Real signals.

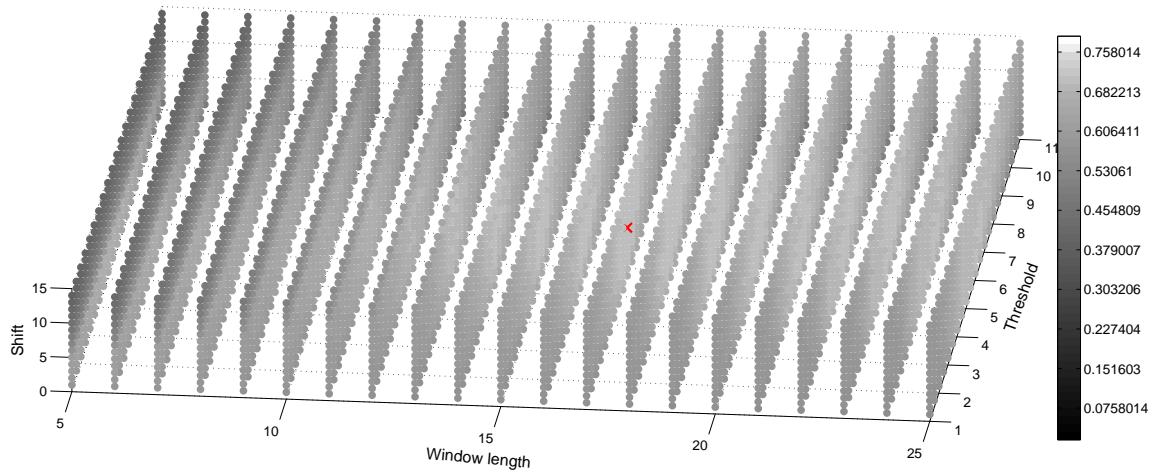


Figure 4.17: Parameter optimization (real signals). Sweep of window length, shift and threshold values to find maximum accuracy (FSD-PROD). Real signals. The red mark indicates the optimal point of operation.

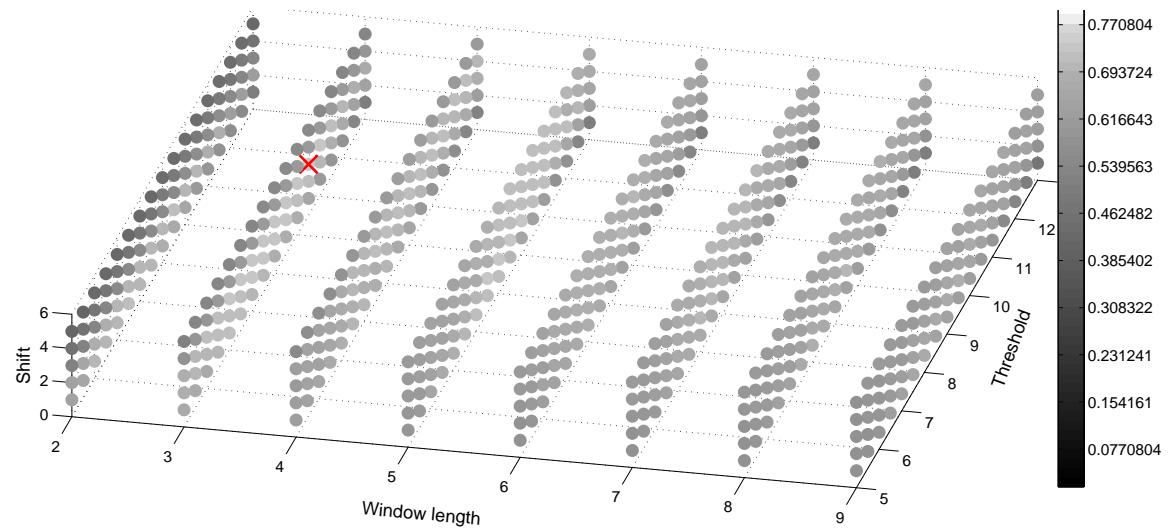


Figure 4.18: Parameter optimization (real signals). Sweep of window length, shift and threshold values to find maximum accuracy (LTSD-SUM). Real signals. The red mark indicates the optimal point of operation.

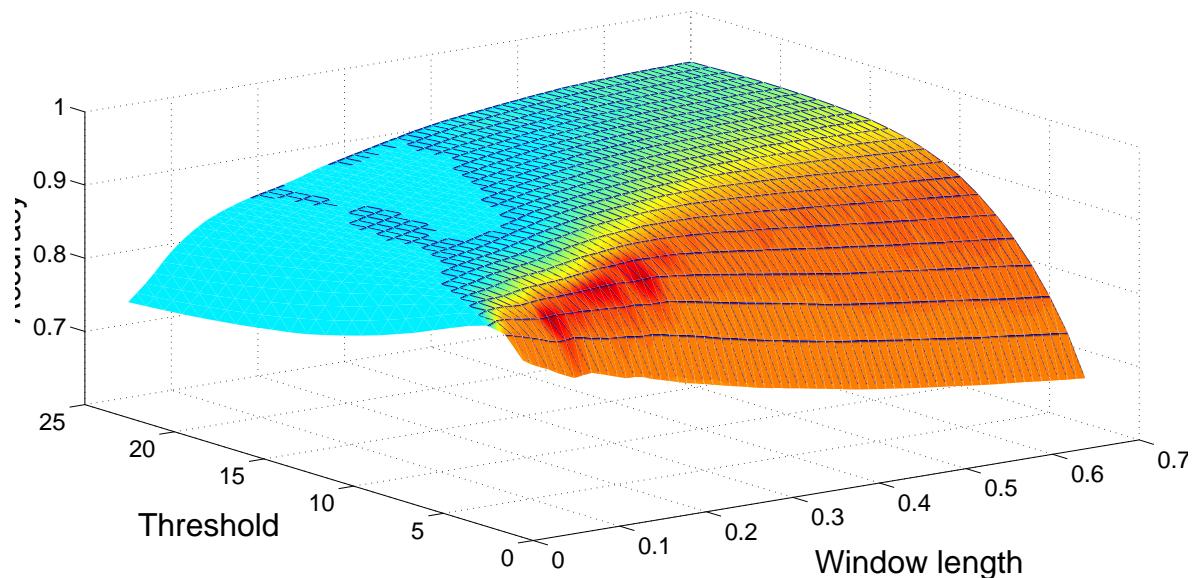


Figure 4.19: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (MBCD). Real signals.

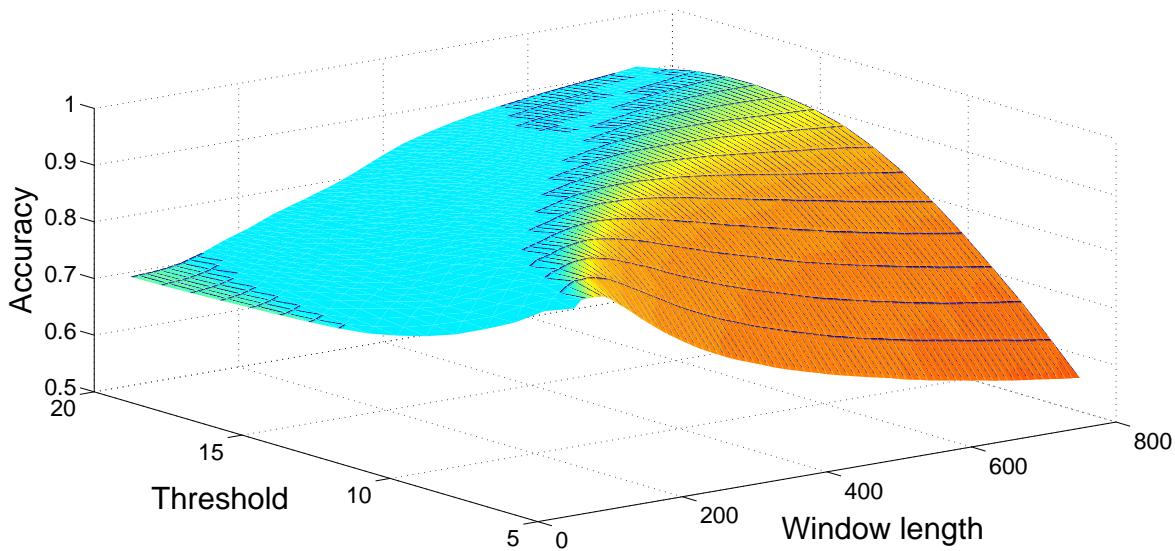


Figure 4.20: Parameter optimization (real signals). Sweep of window length and threshold values to find maximum accuracy (MBGTD). Real signals.

4.4.2.3 Results

Tables 4.7- 4.12 show the average values obtained when the detectors are applied on the 96 gathered real signals. Figures 4.21 and 4.22 show the input (product of acceleration and angular rate magnitudes) and output (figure of merit) of the AMVD and the LTSD for a set of gathered signals when subject number 1 is following the activity circuit wearing the MIMU at the ankle. Binary activity markers have been normalized to the input amplitude in order to allow visibility in the same plot. Notice how AMVD produces a larger number of false positives due to the figure of merit wrongly crossing the threshold when intensity has not changed.

In addition to the Accuracy and the Correlation coefficient, we have also computed the ROC curves and Area Under Curve (AUC) values to follow the standards used to compare detectors and to ease the performance classification of all tested methods. Figures 4.23 and 4.24 show the average ROC curves for the best eight methods when applied on synthesized signals and gathered real signals, respectively. Tables 4.13 and 4.14 show the computed AUC values for each one of the algorithms when they are applied to synthesized and real signals, respectively. As we can see, those methods having high Accuracy rates also present high AUC values. A high AUC value means that the detection algorithm has low False Positive rates when the True Positive rate is high, which is the desired behavior of a classifier.

Table 4.7: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Magnitude methods without flexible input).

	AMVD	AMD	ARED	SHOD
Accuracy	0.9529±0.0113	0.8875±0.0196	0.9418±0.0185	0.9447±0.0236
Correlation coeff.	0.8899±0.02536	0.7610±0.0411	0.8678±0.0381	0.8730±0.0473
Window length	16.7333±2.3851	86.2000±36.2165	8.5167±6.5721	19.6167±9.2617
Threshold	0.0173±0.0106	0.0011±0.0006	38.3250±26.9008	2.3995±1.1856

Table 4.8: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Framed Spectrum Detector).

	FSD-Acc.	FSD-Ang.	FSD-SUM.	FSD-PROD.
Accuracy	0.9702±0.0064	0.9533±0.0194	0.9479±0.0151	0.9515±0.0162
Correlation coeff.	0.9302±0.0155	0.8918±0.0420	0.8804±0.0359	0.8886±0.0385
Window length	20.2000±9.2214	16.4500±4.8910	13.5167±2.5943	14.3000±6.3390
Threshold	3.2433±1.3441	5.0667±2.3935	5.0583±2.72227	5.2917±2.6446
Shift	18.6667±7.0711	15.9667±2.6592	9.4667±1.4477	9.5667±2.5293

Table 4.9: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Long Term Spectral Detector).

	LTSD-Acc.	LTSD-Ang.	LTSD-SUM.	LTSD-PROD.
Accuracy	0.9711±0.0072	0.9682±0.0096	0.9523±0.0591	0.9670±0.0122
Correlation coeff.	0.9318±0.0186	0.9261±0.0228	0.9165±0.0428	0.9264±0.0225
Window length	13.8500±6.4327	5.1167±2.4056	4.6833±0.8023	10.7500±2.5498
Threshold	5.4167±1.9185	8.9167±3.0781	8.7083±2.3045	9.0500±2.9711
Shift	2.4500±0.6390	2.2333±0.8628	1.9167±0.6640	2.5833±1.1134

Table 4.10: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based Graph Theoretic Detector).

	MBGTD-Acc.	MBGTD-Ang.	MBGTD-SUM.	MBGTD-PROD.
Accuracy	0.9626±0.0071	0.9452±0.0120	0.9452±0.0121	0.9468±0.0109
Corr. coeff.	0.9125±0.0186	0.8632±0.0383	0.8634±0.0384	0.8670±0.0359
W. length	13.1833±4.5759	13.6833±5.0705	13.6167±5.0182	13.6500±5.0040
Threshold	1.5467±0.7218	454.0000±285.2036	447.333±279.4448	453.5000±280.7149

Table 4.11: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Memory Based CUSUM Detector).

	MBCD-Acc.	MBCD-Ang.	MBCD-SUM.	MBCD-PROD.
Accuracy	0.9576±0.0080	0.9414±0.0164	0.9414±0.0165	0.9434±0.0154
Correlation coeff.	0.9010±0.0153	0.8588±0.0465	0.8583±0.0469	0.8635±0.0429
Window length	12.6167±4.1869	15.2500±6.4345	15.0500±6.8720	14.5167±6.6832
Threshold	3.468e-6±2.049e-6	0.3588±0.1731	0.3551±0.1746	0.4346±0.1690

Table 4.12: Algorithms applied on real signals. Average Accuracy, Correlation coefficient and associated parameters (Filtered Rectifier Detector (FRD)).

	FRD-Acc.	FRD-Ang.	FRD-SUM.	FRD-PROD.
Accuracy	0.8136±0.0282	0.8417±0.0218	0.8414±0.0218	0.8248±0.0305
Correlation coeff.	0.5754±0.0616	0.6319±0.0657	0.6313±0.0653	0.5878±0.0703
Threshold	0.0055±0.0077	0.1508±0.0928	0.1566±0.0899	0.1558±0.0903

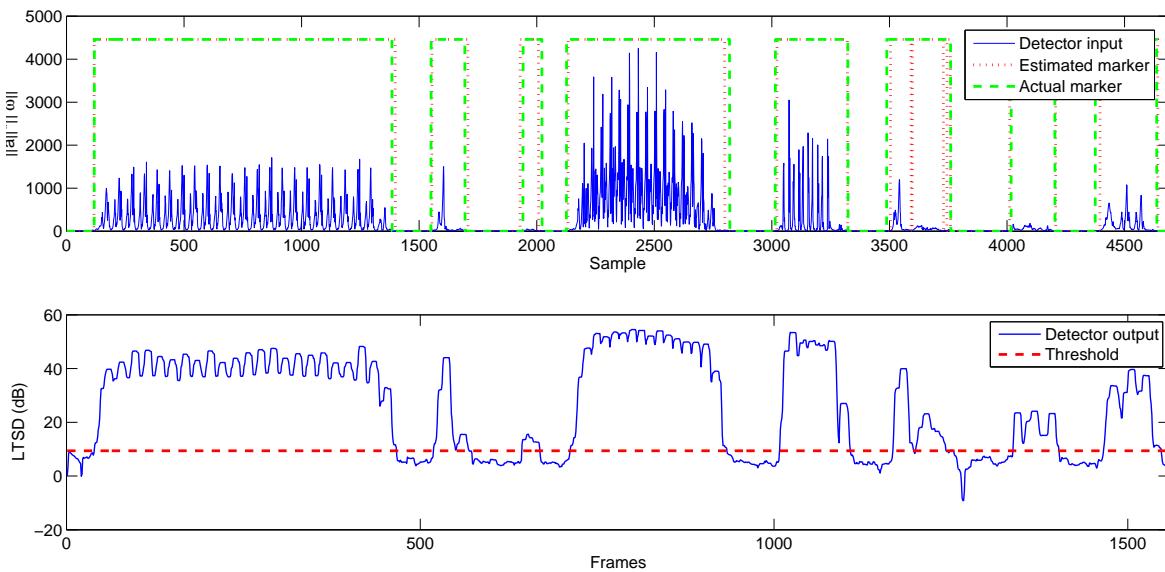


Figure 4.21: Input (product of acceleration and angular rate magnitude) and output (vector of characteristics and marker) of the LTSD. Real signals.

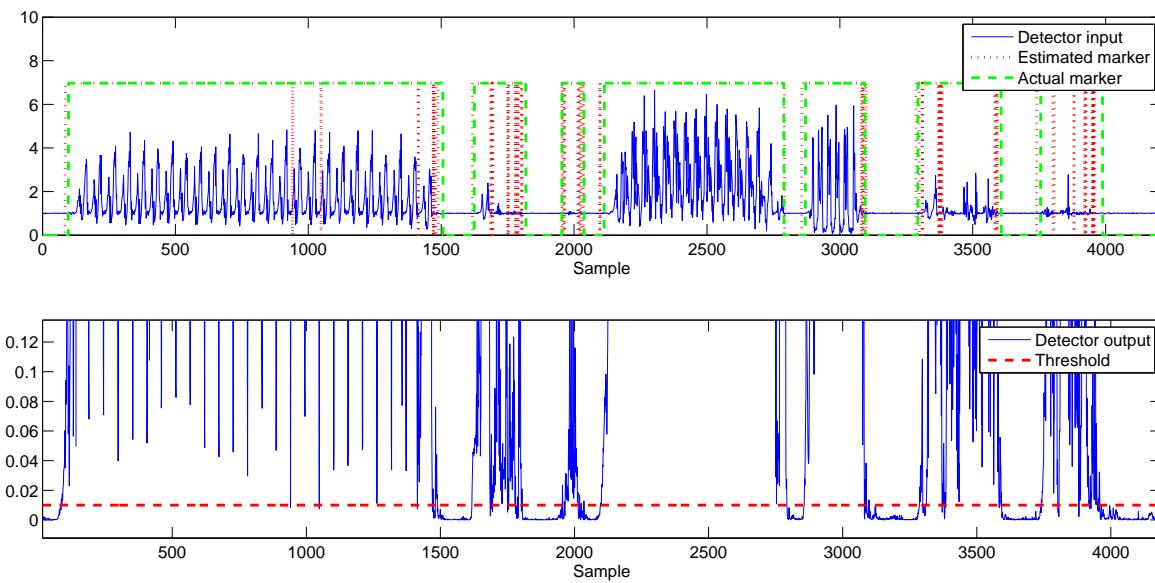


Figure 4.22: Input and output (vector of characteristics and marker) of the AMVD. Real signals.

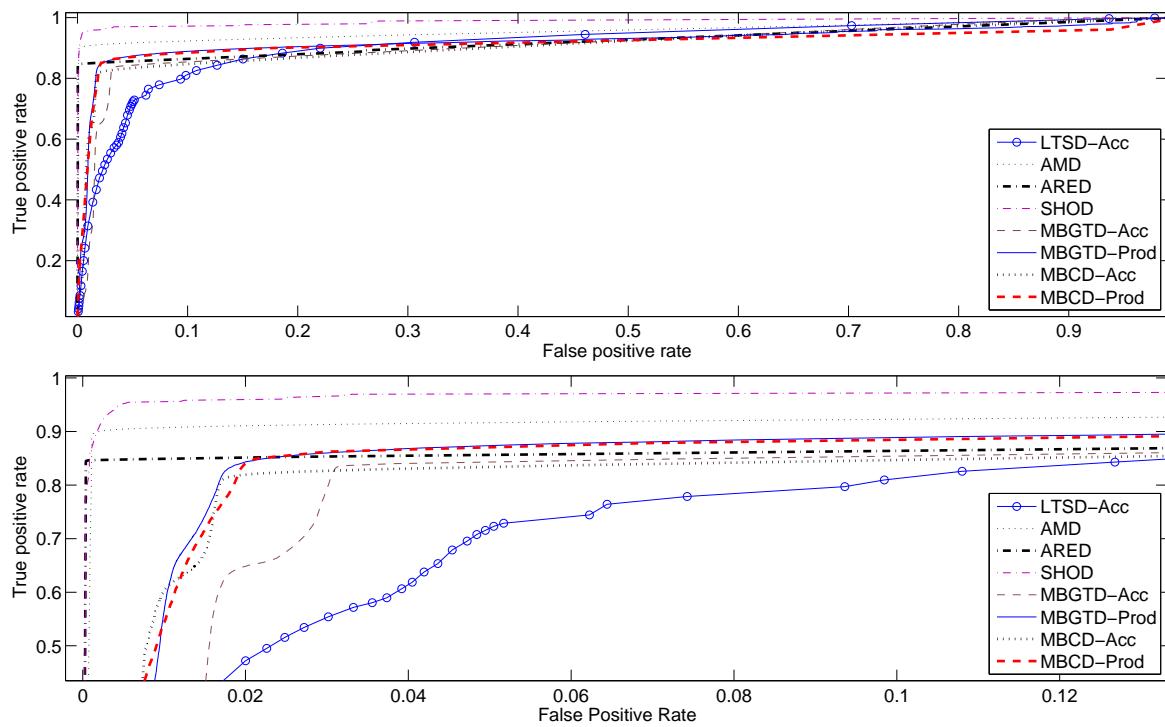


Figure 4.23: ROC curves computed for the eight best methods. Synthesized signals. Complete curves (up), zoomed curves (down).

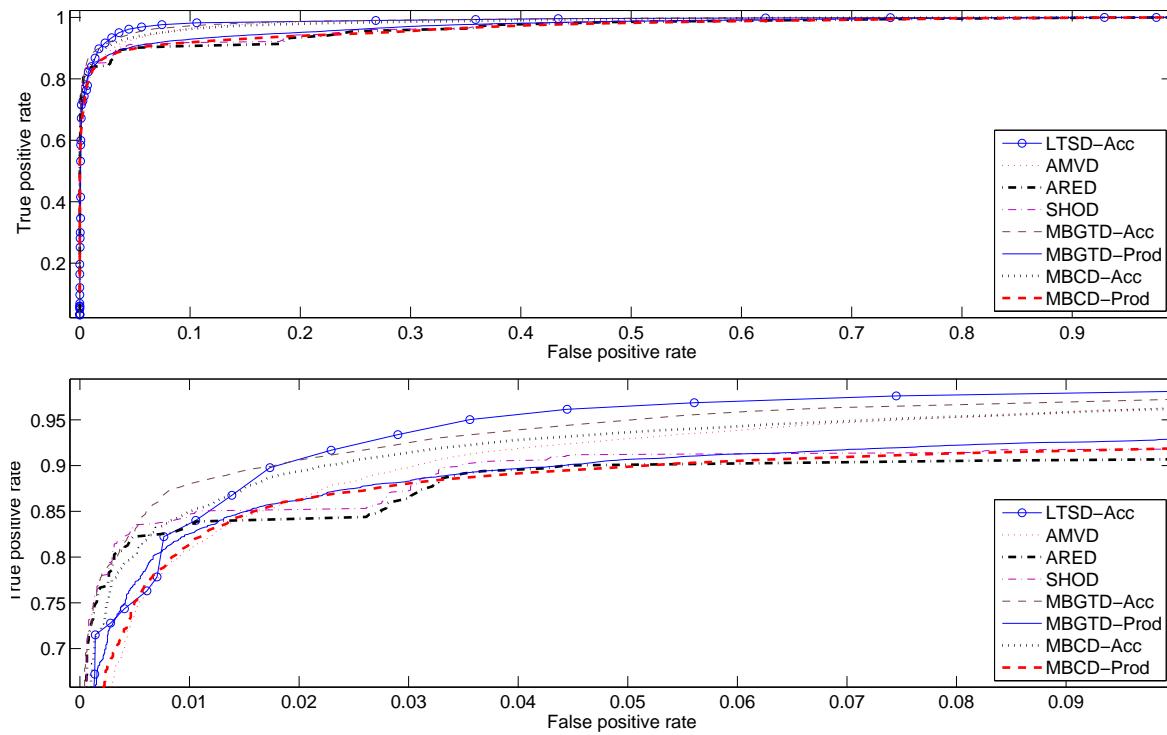


Figure 4.24: ROC curves computed for the eight best methods. Real signals. Complete curves (up), zoomed curves (down).

Table 4.13: Area Under Curve (AUC) computed out of ROC curves obtained from application of algorithms on synthesized signals. Number in brackets indicates overall position in performance comparison.

	AMVD	AMD	ARED	SHOD
AUC	0.8778 (12)	0.9576 (2)	0.9256 (3)	0.9880 (1)
	FSD-Acc.	FSD-Ang.	FSD-Sum	FSD-Prod
AUC	0.8897 (11)	0.7909 (18)	0.7722 (21)	0.7422 (24)
	LTSD-Acc.	LTSD-Ang.	LTSD-Sum	LTSD-Prod
AUC	0.9127 (7)	0.7468 (23)	0.8940 (10)	0.8959 (9)
	MBGTD-Acc.	MBGTD-Ang.	MBGTD-Sum	MBGTD-Prod
AUC	0.9090 (8)	0.7478 (22)	0.8648 (15)	0.9250 (4)
	MBCD-Acc.	MBCD-Ang.	MBCD-Sum	MBCD-Prod
AUC	0.9133 (6)	0.8655 (13)	0.8650 (14)	0.9172 (5)
	FRD-Acc.	FRD-Ang.	FRD-Sum	FRD-Prod
AUC	0.8451 (16)	0.7787 (19)	0.7786 (20)	0.8177 (17)

Table 4.14: Area Under Curve (AUC) computed out of ROC curves obtained from application of algorithms on real signals. Number in brackets indicates overall position in performance comparison.

	AMVD	AMD	ARED	SHOD
AUC	0.9847 (2)	0.9239 (13)	0.9662 (8)	0.9695 (6)
	FSD-Acc.	FSD-Ang.	FSD-Sum	FSD-Prod
AUC	0.8850 (14)	0.7183 (21)	0.7083 (22)	0.6153 (24)
	LTSD-Acc.	LTSD-Ang.	LTSD-Sum	LTSD-Prod
AUC	0.9870 (1)	0.6965 (23)	0.8610 (18)	0.9284 (12)
	MBGTD-Acc.	MBGTD-Ang.	MBGTD-Sum	MBGTD-Prod
AUC	0.9798 (4)	0.8211 (20)	0.9536 (10)	0.9726 (5)
	MBCD-Acc.	MBCD-Ang.	MBCD-Sum	MBCD-Prod
AUC	0.9845 (3)	0.9557 (9)	0.9431 (11)	0.9666 (7)
	FRD-Acc.	FRD-Ang.	FRD-Sum	FRD-Prod
AUC	0.8702 (15)	0.8690 (16)	0.8689 (17)	0.8461 (19)

4.5 Results discussion

We now proceed to discuss the results obtained in the experiments we carried out. In the first part of the section we will analyze and compare all the tested algorithms between them. Additionally, in the second part, we compare our results to those obtained in other works present in the literature.

4.5.1 Results of our experiments

When analyzing the results thrown by the theoretical simulations using magnitude-based methods (table 4.1) we observe that SHOD has the highest accuracy and correlation coefficient and also the highest AUC. Since SHOD uses both acceleration and angular rate signals its detection rate is less affected by non-accelerated or non-spinning movements. AMD classifies second in the performance evaluation even when it is not able to detect the non accelerated movements as it solely relies on the acceleration signals to carry out the detection. AMVD shows the poorest performance of magnitude-based methods. This is due to the fact that when there is an abrupt change in the signal, the variance value will be high, which causes the detector to prematurely detect the transition from a static state to an active state. Analogously, it also prematurely detects the transition from an active state to a static state. These shifts in the estimated marker are the main cause of its poor performance. FRD has a very poor performance in general. This happens because

abrupt changes are smoothed by the filtering process and, therefore, large shifts are introduced at the starting and ending points of each activity period.

On the other hand, the performance of the spectrum methods is somewhere between the performance of the AMVD and the ARED. Amongst them, FSD using the product of the acceleration and the angular rate magnitudes as the input, does the best in terms of accuracy. This is due to the fact that the product of the magnitudes will increase the resultant amplitude of activity periods leading to values much higher than the threshold i.e. detectable values. The LTSD method has a worse decision rate as it is designed to work under conditions where the SNR is low, i.e the sensor signals present large noise which is not the case for our synthesized signals.

Memory-based methods are thought for detecting any abrupt change in signals. This means they also detect changes during active periods. For example, if the subject starts to run faster, the resultant inertial signals will have a larger amplitude and frequency and the figure of merit of the detector will have a higher output. This can be a drawback because if the intensity change during an activity period is very radical, which is similar to a change from inactivity to activity, the detector may wrongly detect the change as a transition from activity to inactivity.

In terms of parameter configuration, we would prefer a shorter window length if we are monitoring movement in real time. Most methods have an optimal window size of around 10 samples which is an adequate latency for real time applications. Only AMD has a latency of 80 samples until it is able to start the detection procedure. This translates to a continuous delay of two seconds during the whole monitoring session when we use an MIMU having a sampling frequency of 40 Hz like the one we used in the present work.

Now, if we look at the results when real signals are used, we can see that the effectiveness of the spectrum and memory-based methods has improved. LTSD using just the acceleration magnitude as input has the best accuracy of all tested methods (0.9711 ± 0.0072). Acceleration signals gathered using the MIMU showed a slightly larger noise than the synthesized signals. This may have caused the performance increase of spectrum methods. Both MBGTD and MBCD present a raise of 3% in the accuracy rate, as the subjects did not perform abrupt changes of intensity while running or walking, which decreased the rate of false changes from activity to inactivity. Alternatively, the raise in the general performance of spectrum and memory-based methods could also be a result of the lower number of real signals that were used to run the tests compared to the number of synthesized signals.

AMD presents a lower performance when monitoring real signals as the zero-crossing-rate was higher than in the theoretical case. Its poorer performance is caused by the high amount of instants where the acceleration crosses the zero level. After computing the magnitude of the acceleration, the values corresponding to zero-crossing instants will still be zero or close to zero; they will be below the threshold and the instant will be erroneously classified as "static". AMVD does better as the transitions from states in real signals are smoother than in synthesized signals.

Table 4.15: Execution times of detection algorithms over a 8000 samples signal. Total execution time, time per sample and percentage of sample period.

	Total execution time	Time per sample	% of sample period
ARED	0.0776s	9.7033e-6s	0.038%
AMD	0.0993s	1.2414e-5s	0.049%
AMVD	0.3661s	4.5765e-5s	0.180%
SHOD	0.4186s	5.2337e-5s	0.200%
MBGTD-PROD	0.4991s	6.2399e-5s	0.240%
FSD-PROD	0.7649s	9.5617e-5s	0.380%
MBCD-PROD	1.0118s	1.2648e-4s	0.500%

Computation times of both memory-based and spectrum-based methods are larger than magnitude based methods when executed in a regular computer. Difference in computation time can be much higher if the algorithms are implemented in processors embedded in mobile devices or MIMUs. This may lead to unacceptable delays in real time monitoring applications. However, this is not a problem if signals are being processed both online or offline in a regular computer. Implementation of magnitude-based methods such as SHOD should be considered when using devices that have low computation power.

Table 4.15 shows the execution time of each one of the algorithms over a signal signal of 8000 samples, the execution time per sample and the percentage of the sample period (0.025 s) it consumes. The algorithms are run using Matlab R2011a in a Intel® Core™ CPU 860 @2.80 Ghz with 8 GB RAM running Windows 7 professional 64 bits. The algorithms are run using their optimal parameters.

4.5.2 Comparison with results in literature

Our main contribution in this work is the proposal of new, and more accurate algorithms to detect human body (in)activity periods using inertial sensors, as well as other existing detection algorithms that had not been applied to this field yet which have outperformed the traditional algorithms. We have also extended the work in [66] by using a larger amount of algorithms and signals to increase the statistical significance of the results.

We have obtained similar results for the methods tested in [66] since SHOD has revealed to be superior to the rest of magnitude-based methods. To our knowledge, one of the first methods developed to detect (in)activity using inertial sensors was presented in [67]. We have shown that, while their method has acceptable rates of accuracy ($\sim 85\%$), the subsequently developed magnitude-based algorithms, as well as our proposals, outperform it.

Not all works presenting detection methods contain an explicit performance study, as in most cases the algorithms were developed as a part of a more complex system with different goals (activity classification, human body positioning algorithms, inertial navigation, etc.). Therefore, it is not easy to compare our results with those obtained by them.

In summary, average maximum accuracy rates and correlation coefficients between the actual activity markers and the markers computed by the algorithms, have been presented, together with the optimal configuration parameters, in tables 4.1-4.6 and 4.7-4.12, for synthesized and real signals respectively. ROC curves of the best methods, as well as their associated AUC values, have been revealed in figures 4.23, 4.24 and tables 4.13, 4.14.

4.6 Chapter wrap-up

The main motivation of the current chapter was to study the best way to implement a motion intensity detector for human body movement monitoring (and also other applications such as inertial navigation). To do so, we have carried out a rigorous and complete comparative study between different algorithms that have been applied in recent literature to detect (in)activity periods in human body motion by means of inertial sensors. To extend the study, we have proposed and tested other methods that are being applied to detect abrupt changes in signals in different applications (industrial processes, voice detection, etc.) that had never been applied to the motion detection field.

Along the chapter, we have presented a comparative study among different magnitude-based algorithms provided in literature such as; the Acceleration Moving Variance Detector (AMVD); the Acceleration Magnitude Detector (AMD); the Angular Rate Energy Detector (ARED); the Stance Hypothesis Optimal Detector (SHOD); and the Filtered Rectifier Detector (FRD). The study presented in [66] has been completed by using a larger data set of theoretical signals. Moreover, a new approach has been tested. It includes spectrum-based algorithms such as the Framed Spectrum Detector (FSD) and the Long Term Spectral Detector (LTSD) and memory-based algorithms such as the Memory-Based Graph Theoretical Detector and The Memory-Based Cumulative Sum Detector (MBCD). The objective was to carry out a statistically robust comparison. To do so, we developed an acceleration and angular rate signal synthesizer that mimics the output of a triaxial accelerometer and a triaxial gyroscope when a subject is performing basic activities such as walking, running, lying down, sitting, standing up and jumping. The theoretical tests show that SHOD is the method with the highest accuracy rate achieving ROC values higher than 0.96. In contrast, tests applied using real signals place LTSD, using the magnitude of the acceleration as input, as the best detector with an accuracy rate of 0.9711 ± 0.0072 . This method is closely followed by FSD-Acc. achieving a correlation coefficient of

0.9302 ± 0.0155 and an accuracy rate of 0.9702 ± 0.0064 .

The use of SHOD is strongly recommended when the system has a reduced computation power and/or when lower delay is preferred over higher precision. Alternatively, LTSD is the best option if movement is being analyzed using a powerful computer and/or in an off-line way.

The importance of motion intensity detection is made patent in chapter 5, where we will see how the precision of attitude estimates can be substantially increased by discriminating between low intensity and high intensity motions.

The experiments presented along this chapter have been published, and are freely available in the following open access publication,

- Olivares A., Ramírez J., Górriz J.M., Olivares G., Damas M. Detection of (In)-activity Periods in Human Body Motion Using Inertial Sensors: A Comparative Study. *Sensors*. 2012; 12(5):5791-5814. <http://www.mdpi.com/1424-8220/12/5/5791>

Computation of Attitude Estimation

Once we know the procedures to be carried out in order to calibrate each one of the sensors and determine the intensity of motion being monitored by the inertial sensors, we can proceed to explain the operations required to compute the attitude estimates.

Attitude is known in the motion monitoring field and specially in aircraft navigation, as the position of the coordinate frame of the body carrying the MIMU with respect to a fix reference coordinate axis. Therefore, from now on, **attitude** is used as an absolute synonym of **orientation**. Attitude is frequently represented using navigation angles (Euler angles), known as *pitch*, *roll* and *yaw*. That is, we will be able to determine the position of the MIMU (and, consequently that of the body to which it is attached), with respect to, for example, the Earth's surface. However, although the three-angle approach is the most usual one, there exist other alternatives to represent the attitude of a body.

Along this chapter we will explain the different employed reference systems as well as the different ways to express and compute attitude. We will see that there exist different ways to compute attitude using inertial sensors and that each one of them has a series of pros and cons when used separately. One of the main goals of the present chapter is to introduce the reader to the concept of sensor fusion, which refers to the conjoint use of information coming from all the sensors included in the MIMU to improve the overall precision of the attitude estimates. We will introduce the difference strategies and mathematical tools which have been used during the last decade to that effect, as well as a series of new approaches to solve some of the existing problems.

To analyze these new approaches, as well as those already existing , we have car-

ried out a comparative study similar to that of chapter 4.

The structure of the chapter is as follows. In section 5.1 we explain the basic concepts which are necessary for the general comprehension of the chapter. Then, in section 5.2, we explain how to design, develop and apply different approaches and algorithms to estimate the attitude from a theoretical and practical point of view. Section 5.3 includes all the experiments we have carried out as well as their results. These results are then analyzed in section 5.4. Finally, section 5.5 closes the chapter with a summary of the most important concepts, approaches, algorithms, results and conclusions.

5.1 Basic concepts

This section is an introduction to the concepts that will be used through this chapter. We will explain some of them in depth while others are just introduced as they will be developed thoroughly during the rest of the chapter.

5.1.1 Coordinate frames

Essential to the determination of the attitude of a body with respect to a fix reference frame is the definition of the coordinate frames. The main coordinate frames are defined by Titterton in [155] as follows,

- *The inertial frame (i-frame)* has its origin at the center of the Earth and axes which are non-rotating with respect to the fixed stars, defined by the axes X , Y , Z , with Z coincident with the Earth's polar axis (which is assumed to be invariant in direction).
- *The navigation frame (n-frame)* is a local geographic frame which has its origin at the location of the navigation system, point P , and axes aligned with the directions of north, east and the local vertical downwards (NED), or sometimes east, north, up (ENU).
- *The body frame (b-frame)*, is an orthogonal axis set which is aligned with the roll, pitch and yaw axes of the vehicle in which the navigation system is installed.

Figures 5.1 depicts the inertial and navigation frames while 5.2 shows the body frame.

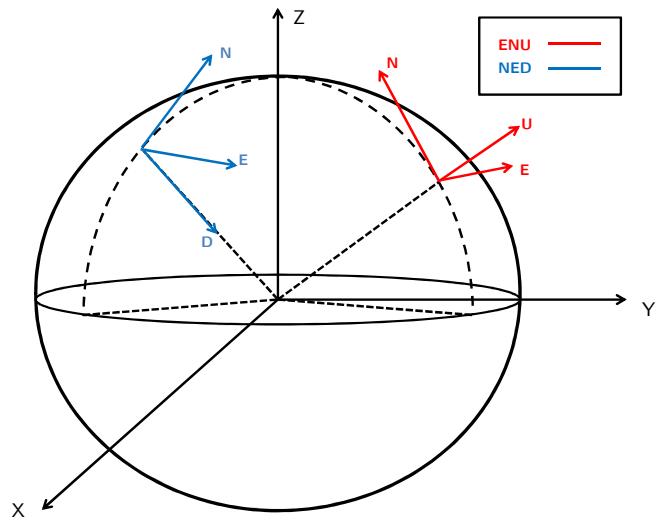


Figure 5.1: Inertial (XYZ), and coordinate frames: East-North-Up (ENU) and North-East-Down (NED)

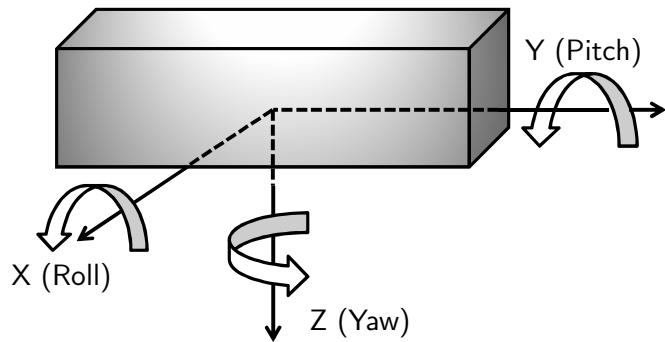


Figure 5.2: Body frame. Turn angle around X axis (Roll), around Y axis (Pitch) and around Z axis (Yaw).

The attitude of the frames with respect to each other can be mapped in many different ways. More specifically, we can use Euler angles, the Direction Cosine Matrix, Rotation Matrices and Quaternions.

5.1.2 Attitude representation

5.1.2.1 Euler angles

Euler angles are used to relate the rotational orientation of an object (its associated body frame) in a three-dimensional space, with respect to the navigation frame, using three angles. So, by rotating each one of the three axes over a known angle, we can map the body frame to the navigation frame and viceversa.

The three Euler angles are defined as follows,

- The *roll* (ϕ) angle determines the rotation around the X axis. It is usually bounded to the range $(-\pi, \pi]$.
- The *pitch* (θ) angle determines the rotation around the Y axis. It is usually bounded to the range $[-\pi/2, \pi/2]$.
- the *yaw* (ψ) angle determines the rotation around the Z axis. It is usually bounded to the range $(-\pi, \pi]$.

Figure 5.3 shows the three rotations that map the navigation frame (red) to the body frame (red). So by undoing the rotations depicted in the figure we carry such a mapping.

Euler angles have the advantage of being easy to represent in a visual way. However, they are subjected to an undesired effect known as the gimbal lock.

The gimbal lock causes the loss of one degree of freedom in a three-dimensional space when two of the three gimbals are in a position in which their axes are parallel. During such a configuration, we will not be able to determine all three Euler angles properly. Figure 5.4 shows one of the positions of a body that lead to a gimbal lock situation. The left drawing shows a normal situation in which all three Euler angles can be computed. The drawing on the right depicts a position in which the yaw angle can not be distinguished from the roll angle (Z axis and X axis gimbals are coincident in space).

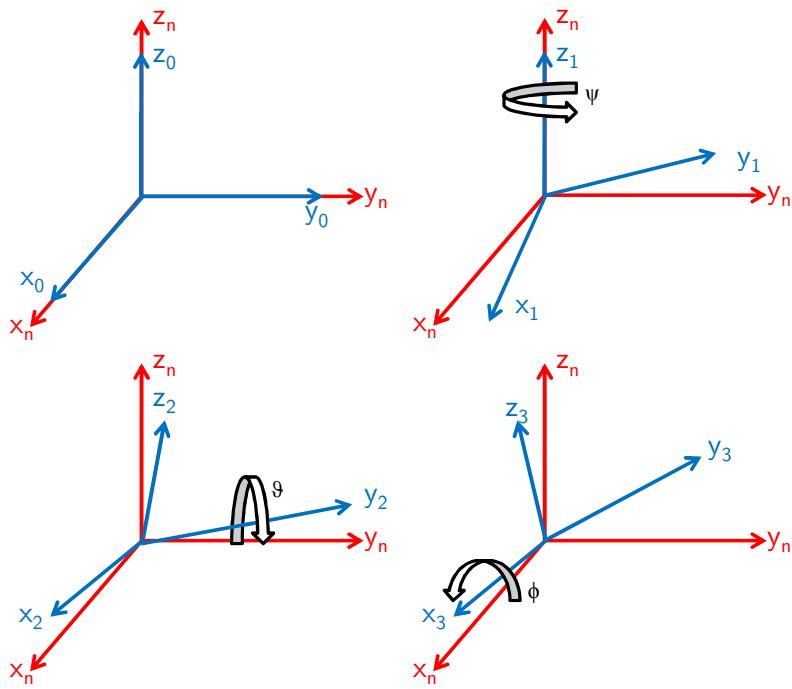


Figure 5.3: Representation of attitude of body frame with respect to the navigation frame using Euler angles.

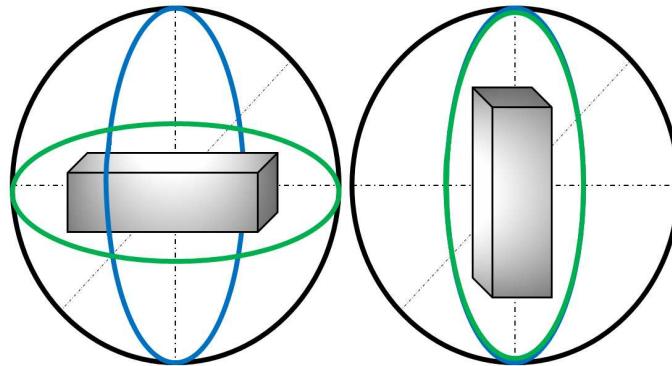


Figure 5.4: Visualization of gimbal lock effect when representing attitude using Euler angles.

As we will see later on, Euler angles can be computed by decomposing the acceleration of the gravity by means of trigonometric relationships. Such an approach leads to discontinuities when the pitch angle approaches $\pm 90^\circ$. This undesired effect is explained in depth in sections 5.1.4 and 5.2.1.

5.1.3 Rotation Matrices

Rotation matrices are used to carry out rotations in Euclidean space. We will use rotation matrices as a way of mapping the Euler rotations into matrices. This way, by multiplying a vector by a rotation matrix we will obtain its rotated version in the space.

$$\mathbf{v}' = \mathbf{R}\mathbf{v} \quad (5.1)$$

The obtention of rotation matrices implies a series of basic trigonometric operations that are summarized in figure 5.5. Depending on which navigation frame we use (NED or ENU) we will have to carefully set the axes and the sense of the rotations. By inspecting figure 5.5 we can extract the rotation matrices for both reference frames.

$$\left. \begin{array}{l} x_1 = x_0 \cos \psi + y_0 \sin \psi \\ y_1 = y_0 \cos \psi - x_0 \sin \psi \\ z_1 = z_0 \end{array} \right\} \Rightarrow R_{\text{NED}}(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left. \begin{array}{l} z_1 = z_0 \cos \theta + x_0 \sin \theta \\ x_1 = x_0 \cos \theta - z_0 \sin \theta \\ y_1 = y_0 \end{array} \right\} \Rightarrow R_{\text{NED}}(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$\left. \begin{array}{l} y_1 = y_0 \cos \phi + z_0 \sin \phi \\ z_1 = z_0 \cos \phi - y_0 \sin \phi \\ x_1 = x_0 \end{array} \right\} \Rightarrow R_{\text{NED}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (5.2)$$

$$\left. \begin{array}{l} y_1 = y_0 \cos \psi + x_0 \sin \psi \\ x_1 = x_0 \cos \psi - y_0 \sin \psi \\ z_1 = z_0 \end{array} \right\} \Rightarrow R_{\text{ENU}}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left. \begin{array}{l} x_1 = x_0 \cos \theta + z_0 \sin \theta \\ z_1 = z_0 \cos \theta - x_0 \sin \theta \\ y_1 = y_0 \end{array} \right\} \Rightarrow R_{\text{ENU}}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$\left. \begin{array}{l} z_1 = z_0 \cos \phi + y_0 \sin \phi \\ y_1 = y_0 \cos \phi - z_0 \sin \phi \\ x_1 = x_0 \end{array} \right\} \Rightarrow R_{\text{ENU}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (5.3)$$

5.1.3.1 Direction Cosine Matrix

The rotation matrices can be grouped together to form a single matrix, known as the Direction Cosine Matrix (DCM), that maps all the rotations and, therefore, the position of the body frame with respect to the navigation frame and viceversa. The

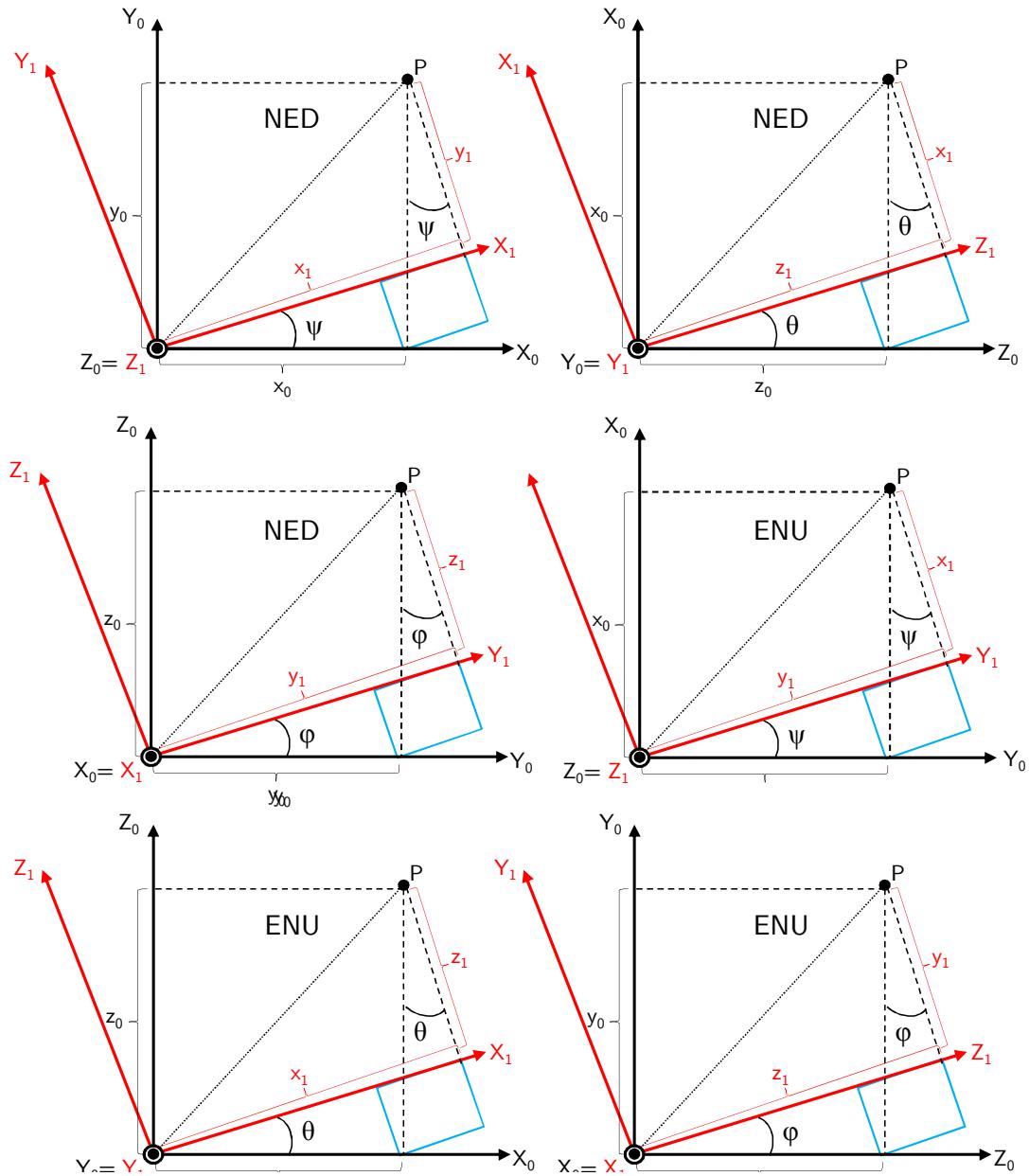


Figure 5.5: Obtention of rotation matrices. North-East-Down system (NED) and East-North-Up system (ENU).

most common sequence of rotations is known as the aerospace sequence and defines yaw as the first rotation, then pitch and finally roll. Let \mathbf{v}^n be a vector expressed in the navigation frame, then the direction cosine matrix C_n^b that transforms it into a vector \mathbf{v}^b expressed in the object's body frame is as follows (from now on we assume that the NED frame is used),

$$C_n^b = R(\phi)R(\theta)R(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

which yields

$$C_n^b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\phi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (5.5)$$

Analogously we can compute the DCM C_b^n that transforms a vector expressed in the object's body frame to the navigation frame by applying

$$C_b^n = (C_n^b)^{-1} = (C_n^b)^T \quad (5.6)$$

leading to

$$C_b^n = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\phi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (5.7)$$

5.1.3.2 Quaternions

Quaternions are an extension of real numbers similar to complex numbers. While complex numbers extend real numbers by the definition an imaginary unit i so that $i^2 = -1$, quaternions are an extension which is generated by adding three imaginary units i, j, k so that the following relations are verified,

$$\begin{aligned} ij &= k, & ji &= -k \\ jk &= i, & kj &= -i \\ ki &= j, & ik &= -j \\ i^2 &= j^2 = k^2 = ijk = -1 \end{aligned} \quad (5.8)$$

Therefore, a quaternion is a number defined as follows,

$$q = a + bi + cj + dk \quad (5.9)$$

where a, b, c and d are arbitrary real numbers. Usually, a is called the scalar part of the quaternion and $bi+cj+dk$ is called the vector part. Quaternions can also be expressed as a compact vector,

$$\mathbf{q} = (a, b, c, d) = (q_0, q_1, q_2, q_3) \quad (5.10)$$

From (5.10) it can be seen that all quaternions, as a set, are equal to \mathbb{R}^4 . The quaternions space, which is usually denominated as \mathbb{H} , has three operations; addition, scalar multiplication and quaternion multiplication. The sum of two quaternions is straight forward as it is defined to be their sum as elements of \mathbb{R}^4 . Analogously, the product of a quaternion by a real number is the same as the product in \mathbb{R}^4 . In order to compute the product of two quaternions we must follow the relations in (5.8) of the $(1, i, j, k)$ basis. It is very important to remember that multiplication of quaternions is not commutative.

From all elements in \mathbb{H} , *unit quaternions* are of special interest to represent rotations. Unit quaternions are quaternions of norm one. They are produced by dividing a non-zero quaternion \mathbf{q} by its norm.

$$U_q = \frac{\mathbf{q}}{\|\mathbf{q}\|} \quad (5.11)$$

The reciprocal of \mathbf{q} is defined as

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \quad (5.12)$$

where \mathbf{q}^* is the conjugated quaternion and is defined as $a - bi - cj - dk$.

A quaternion rotation can be constructed similarly to the Euler formula by using the following formula,

$$\mathbf{q} = e^{\frac{1}{2}\theta(a_x i + a_y j + a_z k)} = \cos \frac{1}{2}\theta + (a_x i + a_y j + a_z k) \sin \frac{1}{2}\theta \quad (5.13)$$

where θ is the angle of rotation and (a_x, a_y, a_z) is the axis of rotation.

The quaternion rotation can, thus, be also expressed as,

$$\mathbf{q} = (q_0, q_1, q_2, q_3) = \left(\cos \frac{\alpha}{2}, u_x \sin \frac{\alpha}{2}, u_y \frac{\alpha}{2}, u_z \frac{\alpha}{2} \right) \quad (5.14)$$

where $\mathbf{u} = (u_x, u_y, u_z)$ is a \mathbb{R}^3 unit vector of magnitude equal to unity. Quaternions have the following properties related to rotations, viz:

- Let $\mathbf{a} = (a_x, a_y, a_z)$ be an arbitrary vector in a \mathbb{R}^3 space, and its associated quaternion $\mathbf{q}_a = (0, a_x, a_y, a_z)$. Let also $\mathbf{q} = (q_0, q_1, q_2, q_3)$ an arbitrary unit quaternion describing a rotation in a three-dimensional space. Then, we can rotate \mathbf{a} the angle defined by \mathbf{q} in the following way,

$$\mathbf{q}'_a = \mathbf{q} \cdot \mathbf{q}_a \cdot \mathbf{q}^{-1} \quad (5.15)$$

where \cdot indicates the quaternion product and $\mathbf{q}'_a = (0, a'_x, a'_y, a'_z)$

- Similarly, if $\mathbf{q}'_a = \mathbf{q} \cdot \mathbf{q}_a \cdot \mathbf{q}^{-1}$ describes a rotation, the inverse rotation can be computed using

$$\mathbf{q}_a = \mathbf{q}^{-1} \cdot \mathbf{q}'_a \cdot \mathbf{q} \quad (5.16)$$

- Let \mathbf{q}_1 and \mathbf{q}_2 be two unit quaternions describing two rotations in a three-dimensional space. Then, the rotation composed of rotating first the angle determined by \mathbf{q}_1 followed by the angle determined by \mathbf{q}_2 can be described by the following product,

$$\mathbf{q}' = \mathbf{q}_2 \cdot \mathbf{q}_1 \quad (5.17)$$

Quaternions are numerically more stable than rotation matrices and in many cases more efficient. When used to represent an orientation, i.e. a rotation relative to a reference position, they are usually called orientation or attitude quaternions.

The main drawback of quaternions is, as they define a \mathbb{R}^4 space, they can not be directly represented in a three-dimensional space, making their visualization non representative and non intuitive. To avoid this, most applications extract Euler angles after performing all calculations using quaternions. Euler angles can be extracted from quaternions using the following transformations,

$$\begin{aligned}\phi &= \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \\ \theta &= \arcsin(2(q_0q_2 - q_3q_1)) \\ \psi &= \arctan\left(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}\right)\end{aligned} \quad (5.18)$$

Since arctan and arcsin functions have a result range of $[-\pi/2, \pi/2]$ we can not define all possible orientations having three rotations with such a range. As we aforementioned, to cover all possible orientations, roll (ϕ) and yaw (ψ) are generally defined within a $(-\pi, \pi]$ range while pitch (θ) is bounded to $[-\pi/2, \pi/2]$. Therefore, in order to define all possible positions and rotations of a body in a three-dimensional space we need to substitute the arctan function in (5.18) by the function *atan2* which is defined in $(-\pi, \pi]$.

5.1.4 Projection of gravity vector and Earth's magnetic field

One of the approaches to compute the attitude of an object is to project the gravity vector on the accelerometer axes. As it was explained in chapter 2, accelerometers measure the acceleration to which they are subjected. Under static, quasi-static (object moving at a constant velocity) or low acceleration conditions we can assume that the sensed acceleration will be mainly that of the gravity and neglect dynamic accelerations. Therefore, by carrying out simple trigonometric transformations we

will be able to obtain estimates for pitch and roll angles. The yaw angle can not be computed by means of this method since the XY plane is perpendicular to the gravity vector and, then, any rotation around the Z axis will not cause any variation in the measured acceleration.

This drawback can be solved by using a magnetometer which measures the variation of Earth's magnetic field around the X, Y and Z axis and, therefore, provides an extra degree of freedom. However, the readings of the Earth's magnetic field can be distorted by any other magnetic fields nearby, introducing large errors in the estimation of yaw. The algorithms employed to estimate pitch, roll and yaw angles using both the accelerometer and the magnetometer will be explained in depth in section 5.2.1.

This approach to compute the attitude is only valid under static, quasi-static or low acceleration conditions. When the object is suffering from high accelerations the gravity vector will not be predominant anymore, that is we are not able to neglect dynamic accelerations.

5.1.5 Integration of angular rate

Another approach which is solely based on the gyroscope consists on computing roll, pitch and yaw angles by integrating the measured angular rate around X, Y and Z axes respectively. From a theoretical point of view this method would be enough to achieve very accurate estimates of the attitude. However, as we stated in chapter 2, integrating a noisy angular rate signal leads to Angle Random Walk. ARW and integration of dynamic bias cause the computed angle signal to dramatically drift in a very short period of time making the estimate absolutely non reliable. Generally, as the noise can be modeled as random white noise of zero mean, the effects of ARW are not as disruptive as those of the integration of dynamic bias. Figure 5.6 shows the effects of integrating angular rate to obtain an estimate of the rotation angle. As we can see, the signal drifts with time, leading to an error of around 75 degrees in just 25 seconds. Unlike with accelerometers, attitude estimates computed integrating angular rate are not affected by dynamic accelerations.

The algorithms employed to estimate pitch, roll and yaw angles using the gyroscope will be explained in depth in section 5.2.2.

5.1.6 Sensor fusion

Each one of the MARG sensors has its advantages and its drawbacks. There are situations in which they provide good estimates of attitude and others in which the estimates are absolutely erroneous. Therefore, the solution is to combine the information coming from all sensors in a weighted way. This is known as sensor fusion.

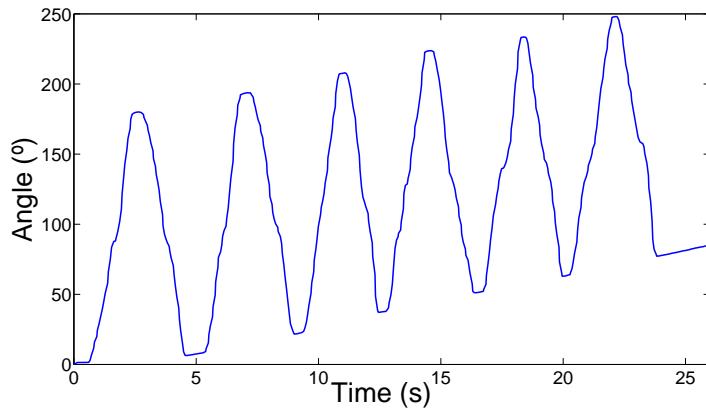


Figure 5.6: Effects of dynamic bias on the integrated gyroscope output signal.

The term sensor fusion is referred to the use of information coming from multiple sensors so that the degree of precision is higher than it would be if the sensors were used separately. Therefore, we combine data coming from different kinds of sensors to predict and/or estimate states of an observed process. Raol [156] defines the following advantages which are expected from the use of sensor fusion:

1. Multiple sensors would provide redundancy which, in turn, would enable the system to provide information in case of partial failure, data loss from one sensor —i.e., fault tolerance capability- robust functional and operational performance.
2. Measurements of one sensor are confirmed by the measurements of the other sensors, obtaining cooperative arrangement and enhancing the confidence— increased confidence in the inferred results.
3. Joint information would tend to reduce ambiguous interpretations and hence less uncertainty.
4. Multiple independent measurements when fused would improve the resolution —enhanced spatial resolution.
5. Extended temporal coverage—the information is continually available.

When aiming to compute attitude estimates using MARG sensors, the best approach is to use the information coming from the accelerometer and the magnetometer when the intensity of the motion is low and the information coming from the gyroscope when the intensity is high. There are different solutions to fuse the information [156] but we will be focusing in adaptive filtering. All the sensor fusion approaches we have used will be explained in depth in section 5.2.4.

5.2 Design and development algorithms

The previous sections have introduced the most important basic concepts to understand the representation of the orientation of a body through the estimation of its attitude. Along this section we will explain in detail the operation principles of the aforementioned approaches and how they can be implemented from a practical point of view.

5.2.1 Projection of gravity and Earth's magnetic field vectors

As it was explained before, we can compute the attitude by projecting the components of the local Earth's magnetic field vector and the Earth's gravitational vector on the sensor's axes. The accelerometer and the magnetometer measure the Earth's gravitational and Earth's magnetic field local vectors in the body frame. Therefore, we can compute the orientation angles by using the known values of such vectors in the navigation frame. Next, we will explain two different approaches, the first one is based on decomposing the vectors using trigonometric functions to find the Euler angles, and the second one is based on the computation of the quaternions that map all three rotations defining the orientation of the body frame with respect to the navigation frame.

5.2.1.1 Euler angles-based approach

The very first thing we need to do is to decide the axes convention that will be employed. In our case, we followed the convention which is used in aerospace applications.

- The X axis is aligned along the longitudinal axis of the body. Our MIMU has a rectangular case, so, the X axis goes along the longest side.
- The Y axis is aligned along the lateral axis of the body. In our case, the Y axis goes along the shortest side.
- The Z axis points downwards and is aligned with the vertical axis of the body so that it is parallel to the earth's gravitational vector when the plane XY is flat on a surface.

We have to take into account that we should not use the accelerometer's own coordinate system as it is not convenient to define the orientation of the object to which it is attached because it may be mounted at any orientation on the circuit board. It

is advisable, then, to use a coordinate system which is aligned with the axes of the MIMU case. Therefore, we first have to identify the accelerometer's coordinate system by placing the MIMU's axes parallel and antiparallel to gravity and then apply the corresponding corrections (redefining axes, changing the sense of axes, etc.) so they match the desired coordinate system. Most systems follow the convention that the accelerometer output should be +1 g in any axis aligned with the earth's gravitational vector and pointing downwards.

Once we have identified the coordinate system of the accelerometer and adjusted its measurements to the coordinate system defined by the MIMU case, we can continue the definition of the expressions of the orientation Euler angles.

As explained in [81], an accelerometer experiencing a linear acceleration \mathbf{a}_n measured in the navigation frame, will have an output \mathbf{u} defined as follows,

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = R(\mathbf{g} - \mathbf{a}_n) \quad (5.19)$$

where R is the compose rotation matrix (or direction cosine matrix) describing the orientation of the MIMU's body frame with respect to the navigation frame as seen in section 5.1.3.1.

To extract the orientation angles we have to assume that $\mathbf{a}_n \approx 0$, that is, the object is static, quasi-static, slowly moving or moving at constant velocity. Then, if we make $\mathbf{a}_n = 0$ we have

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = R\mathbf{g} = R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.20)$$

➤ Estimation of Pitch and Roll

The rotation matrix R can be formed as a set of six different combinations of the three rotations defined by $R(\phi)$, $R(\theta)$ and $R(\psi)$. These six different combinations lead to six different rotation matrices (see [81]) which, when substituted in equation (5.20), produce six different systems of equations with three unknowns (ϕ , θ and ψ).

As explained in (see [81]), the accelerometer output has three components but only two degrees of freedom since the vector magnitude must always be equal to 1 g if $\mathbf{a}_n \approx 0$. Therefore, we can only find the solution for two of the three Euler angles. The only two systems (out of the six) which depend only on two angles (pitch and roll) are

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = R\mathbf{g} = R(\phi)R(\theta)R(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.21)$$

$$\begin{aligned}
\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} &= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \phi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \\
&= \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix}; \\
\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} &= \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \tag{5.22}
\end{aligned}$$

and

$$\begin{aligned}
\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} &= Rg = R(\theta)R(\phi)R(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \\
\begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta + \sin \theta \sin \phi \cos \psi & -\sin \theta \cos \phi \\ -\cos \phi \sin \psi & \cos \phi \cos \psi & \sin \phi \\ \cos \theta \sin \phi \sin \psi + \sin \theta \cos \psi & -\cos \psi \cos \theta \sin \phi + \sin \psi \sin \theta & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \\
&= \begin{bmatrix} -\sin \theta \cos \phi \\ \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \tag{5.23}
\end{aligned}$$

We will be using the convention defined by system 5.21 originated from the sequence of rotations $R(\phi)R(\theta)R(\psi)$. This set of rotations is known as the aerospace rotation sequence. Now, if we pre-multiply equation 5.21 by the inverse roll and pitch rotation matrices we get,

$$R(-\theta)R(-\phi) \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = R(-\theta)R(-\phi)R(\phi)R(\theta)R(\psi) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = R(\psi) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{5.24}$$

substituting the values of $R(-\theta)$ and $R(-\phi)$ and $R(\psi)$ we get

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{5.25}$$

which yields

$$\begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{5.26}$$

To solve for the roll angle (ϕ) we use the value of the component u_y from equation (5.26) as follows,

$$u_y \cos \phi - u_z \sin \phi = 0 \quad (5.27)$$

$$\tan \phi = \frac{u_y}{u_z}; \phi = \text{atan2}(u_z, u_y) \quad (5.28)$$

Now, to solve for the pitch angle (θ), we use the value of the component u_x also from equation (5.26),

$$u_x \cos \theta + u_y \sin \theta \sin \phi + u_z \sin \theta \cos \phi = 0 \quad (5.29)$$

which gives

$$\tan \theta = \frac{-u_x}{u_y \sin \phi + u_z \cos \phi} \quad (5.30)$$

and, after some cumbersome algebraic manipulations, we finally get

$$\tan \theta = \frac{-u_x}{\sqrt{u_y^2 + u_z^2}}; \theta = \text{atan2}(\sqrt{u_y^2 + u_z^2}, -u_x) \quad (5.31)$$

where *atan2* is the quadrant-corrected *atan* function present in almost any programming language out there.

In order to eliminate duplicate solutions in (5.28) and (5.31) we need to limit the range of one of the two angles as the *atan2* function returns values within the $[-\pi, \pi]$ range. The convention adopted in aerospace applications is to limit the pitch angle to $[-\pi/2, \pi/2]$. Therefore, we need to map the values laying in $[\pi/2, \pi]$ and $[\pi, \frac{3}{2}\pi]$ to $[0, \pi/2]$ and $[-\pi/2, 0]$ respectively.

As we commented in section 5.1.4, the Euler angles-based trigonometric approach has some issues with singularities of the *atan2* function. Equation (5.28), will not be defined when both the numerator and the denominator are zero, that is, $u_y = 0$ and $u_z = 0$. The output of the accelerometer will return this values when the X axis is set parallel or antiparallel to the gravity vector, that is, both Y and Z axes are contained in the Earth's surface plane and are, thus, perpendicular to the gravity vector. Under these circumstances the roll angle will not be defined and the system will momentarily loose a degree of freedom. This situation was defined as *gimbal lock* in section 5.1.2.1. On the other hand, (5.31) will always be defined as, when the linear acceleration is zero or close to zero, the magnitude of the accelerometer's output has to be 1 g or close to it. Therefore, u_x, u_y and u_z will never be zero at the same time.

➤ Estimation of Yaw

We have seen how, using just an accelerometer we are only able to obtain two degrees of freedom (roll and pitch angles). To obtain the third degree of freedom,

that is, the rotation angle around Z axis (yaw) we need an extra sensor. Magnetometers measure the intensity of magnetic field projected to its axes. Under ideal conditions, the magnetometer would only measure the intensity of Earth's magnetic field. However, there will always exist magnetic disturbances originated by nearby ferromagnetic materials and many other sources. If the MIMU is carefully designed so the magnetic sensor has no ferromagnetic materials in the surroundings we can get clean enough measurements from Earth's magnetic field. Spontaneous and transitory disturbances can be dealt with by means of sensor fusion [88].

Let's imagine that the MIMU's XY plan is set parallel to the plane normal to the Earth's magnetic field (local horizontal plane). This means that both pitch and roll angles are zero, so the yaw angle would be computed just by applying

$$\psi = \arctan(h_y/h_x) \quad (5.32)$$

where h_x and h_y are the Earth's magnetic field components projected to the magnetometer's X and Y axes respectively. In this case, we can directly compute the arctangent as both X and Y body frame's axes are in the same plane than navigation frame's X and Y axes. What happens if the body frame is rotated in pitch and roll with respect to the navigation frame? Well, in this case we need to rotate back X and Y axes such pitch and roll angles. This can be easily done by applying $R(-\theta)$ and $R(-\phi)$ rotation matrices to the magnetometer's output vector as in equation (5.2.1.1). Once the output is projected to the XY plane of the navigation frame we can apply equation (5.32).

$$\begin{aligned} \mathbf{h}_n &= R(-\theta)R(-\phi)\mathbf{h}_b = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} h_x^b \\ h_y^b \\ h_z^b \end{bmatrix} = \\ &= \begin{bmatrix} \cos\theta & \sin\theta \sin\phi & \sin\theta \cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta \sin\phi & \cos\theta \cos\phi \end{bmatrix} \begin{bmatrix} h_x^b \\ h_y^b \\ h_z^b \end{bmatrix} = \\ &= \begin{bmatrix} h_x^b \cos\theta + h_y^b \sin\theta \sin\phi + h_z^b \sin\theta \cos\phi \\ h_y^b \cos\phi - h_z^b \sin\phi \\ -h_x^b \sin\theta + h_y^b \cos\theta \sin\phi + h_z^b \cos\theta \cos\phi \end{bmatrix} \end{aligned} \quad (5.33)$$

$$\begin{aligned} h_x^n &= h_x^b \cos\theta + h_y^b \sin\theta \sin\phi + h_z^b \sin\theta \cos\phi \\ h_y^n &= h_y^b \cos\phi - h_z^b \sin\phi \end{aligned} \quad (5.34)$$

Then, the X and Y components of the de-rotated magnetic vector \mathbf{h}_n can be used to compute the yaw angle just by substituting h_x and h_y in equation (5.32) by h_x^n and h_y^n in equation (5.34).

$$\psi = \text{atan2}(h_x^n, h_y^n) \quad (5.35)$$

5.2.1.2 Quaternions approach

It is important to remark that the method based on rotations defined through Euler angles needs from the evaluation of trigonometric functions and inverse trigonometric functions. These functions add a computational cost that may make this approach unsuitable for real time applications or low power systems having low processing power. In addition to his drawback, the Euler angles approach has also the aforementioned problem with singularities when computing the roll angle.

On the other hand, if we define the rotations using quaternions, we can compute the attitude estimates without having to evaluate trigonometric functions and with nothing more than basic operations such multiplications, additions and subtractions. In this section, we explain in depth the Factorized Quaternion Algorithm (FQA) developed by Yun et al. in [71]. This method computes the pitch, roll and yaw quaternions separately and then puts them together to form a single quaternion representing the three rotations that map the body frame to the navigation frame. The resulting quaternion can be then used to extract the Euler angles—so the visualization of the orientation is more intuitive—, or as the input of a sensor fusion algorithm.

➤ Pitch quaternion

The quaternion reflecting a rotation of a pitch angle θ around the Y axis can be expressed using equation (5.13) as follows,

$$\mathbf{q}_\theta = \cos \frac{\theta}{2} [1 \ 0 \ 0 \ 0] + \sin \frac{\theta}{2} [0 \ 0 \ 1 \ 0] \quad (5.36)$$

where the vector (a_{xi}, a_{yj}, a_{zk}) in equation (5.13) is substituted by the unity vector $\mathbf{n} = (n_x, n_y, n_z)$ representing the Y axis.

As we have just explained, the tilt angle of a body about the Y axis can be obtained from the projection of the gravity vector measured by the accelerometer. So, if a body is tilted around the Y axis we have that

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (5.37)$$

so,

$$\begin{aligned} u_x &= -g \sin \theta \\ u_z &= g \cos \theta \end{aligned} \quad (5.38)$$

Now, for convenience, we normalize the measured acceleration vector so that,

$$\bar{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|} \quad (5.39)$$

$$\sin \theta = \bar{u}_x \quad (5.40)$$

In order to get the quaternion \mathbf{q}_θ we need to compute the values of $\cos \frac{\theta}{2}$ and $\sin \frac{\theta}{2}$. To do so, we need to use the trigonometric half-angle formulas given by

$$\sin \frac{\theta}{2} = \text{sign}(\sin \theta) \sqrt{(1 - \cos \theta)/2} \quad (5.41)$$

$$\cos \frac{\theta}{2} = \sqrt{(1 + \cos \theta)/2} \quad (5.42)$$

where $\text{sign}(\cdot)$ is the sign function which is defined as,

$$\text{sign}(x) = \begin{cases} +1, & x > 0 \\ 0, & x=0 \\ -1, & x < 0 \end{cases} \quad (5.43)$$

The sign function has been removed from equation (5.42) as $\cos \frac{\theta}{2}$ will always be positive within the pitch angle range, $[-\pi/2, \pi/2]$.

The value of $\cos \theta$ needed in (5.41) can be easily obtained from

$$\cos \theta = \sqrt{1 - \sin^2 \theta} \quad (5.44)$$

Now that we know the values of $\cos \frac{\theta}{2}$ and $\sin \frac{\theta}{2}$, we can compute the pitch quaternion just by applying equation (5.36).

➤ Roll quaternion

The next step is to compute the roll quaternion. To do so, we proceed in a similar way to the computation of the pitch quaternion. First, we need to know that the quaternion defining a turn angle ϕ about the X axis is defined as

$$\mathbf{q}_\phi = \cos \frac{\phi}{2} [1 \ 0 \ 0 \ 0] + \sin \frac{\phi}{2} [0 \ 1 \ 0 \ 0] \quad (5.45)$$

Therefore, we need to compute the value of $\cos \frac{\phi}{2}$ and $\sin \frac{\phi}{2}$. To do so, we remember from section 5.2.1.1 that the acceleration measured by an accelerometer which is attached to a body that is tilted first around and angle ϕ and θ around X and Y axes respectively is as follows,

$$\begin{aligned} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \\ &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \phi \sin \theta & \cos \phi & \sin \phi \cos \theta \\ \cos \phi \sin \theta & -\sin \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \end{aligned} \quad (5.46)$$

yielding

$$u_y = g \sin \phi \cos \theta \quad (5.47)$$

$$u_z = g \cos \phi \cos \theta \quad (5.48)$$

We normalize the accelerometer readings as we did with the pitch quaternion

$$\bar{u}_y = \sin \phi \cos \theta \quad (5.49)$$

$$\bar{u}_z = \cos \phi \cos \theta \quad (5.50)$$

At this point we can compute the values of $\sin \phi$ and $\cos \phi$ using the value of $\cos \theta$ computed with equation (5.44)

$$\sin \phi = \frac{\bar{u}_y}{\cos \theta} \quad (5.51)$$

$$\cos \phi = \frac{\bar{u}_z}{\cos \theta} \quad (5.52)$$

Finally, we compute the values of the half angles and plug them in equation (5.45).

$$\sin \frac{\phi}{2} = \text{sign}(\sin \phi) \sqrt{(1 - \cos \phi)/2} \quad (5.53)$$

$$\cos \frac{\phi}{2} = \sqrt{(1 + \cos \phi)/2} \quad (5.54)$$

➤ Yaw quaternion

The last step is to compute the quaternion that defines a rotation of an angle ψ about the Z axis. Analogously to the pitch and roll, the yaw quaternion is defined as

$$\mathbf{q}_\psi = \cos \frac{\psi}{2} [1 \ 0 \ 0 \ 0] + \sin \frac{\psi}{2} [0 \ 0 \ 0 \ 1] \quad (5.55)$$

Similarly to the computation of the yaw angle based on Euler angles, we first have to rotate in pitch and roll the magnetic field vector measured in the body frame \mathbf{h}^b back to an intermediate frame $\mathbf{h}^{n'}$ which is coincident with the navigation frame except from the rotation about the Z axis. This can be done by applying the inverse quaternion rotation (5.16)

$$\mathbf{h}^{n'} = \mathbf{q}_\theta^{-1} \mathbf{q}_\phi^{-1} \mathbf{h}^b \mathbf{q}_\phi \mathbf{q}_\theta \quad (5.56)$$

where the reciprocal quaternion \mathbf{q}^{-1} can be computed applying equation (5.12).

After rotating back the output of the magnetometer in pitch and roll we have that the known local magnetic vector \mathbf{n} —which value can be consulted in [137]—is coincident with $\mathbf{h}^{n'}$ except from the azimuth (yaw) rotation. The estimated value of \mathbf{n} in our lab in Granada, Spain, is depicted in table 5.1. Therefore, we have that

Table 5.1: Value of the local magnetic vector in our lab in Granada, Spain.

Date	North Comp. (+ N -S)	East Comp. (+ E -W)	Vertical Comp. (+ D -U)	Total Field
2012-06-28	27,195.08 nT	-633.83 nT	33,740.50 nT	43,340.46 nT

$$\begin{aligned} \mathbf{n} &= R(\psi) \mathbf{h}^{n'} \\ \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_x^{n'} \\ h_y^{n'} \\ h_z^{n'} \end{bmatrix} \\ \begin{bmatrix} n_x \\ n_y \end{bmatrix} &= \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} h_x^{n'} \\ h_y^{n'} \end{bmatrix} \end{aligned} \quad (5.57)$$

Equation (5.57) implies that \mathbf{n} and $\mathbf{h}^{n'}$ have the same magnitude. However, magnetic interferences and noise present in the output of the magnetometer make both vectors differ in length. To avoid this drawback, we normalize both vectors as follows

$$N = \begin{bmatrix} N_x \\ N_y \end{bmatrix} = \frac{1}{\sqrt{n_x^2 + n_y^2}} \begin{bmatrix} n_x \\ n_y \end{bmatrix} \quad (5.58)$$

$$H^{n'} = \begin{bmatrix} H_x^{n'} \\ H_y^{n'} \end{bmatrix} = \frac{1}{\sqrt{h_x^{n'^2} + h_y^{n'^2}}} \begin{bmatrix} h_x^{n'} \\ h_y^{n'} \end{bmatrix} \quad (5.59)$$

We plug now the normalized vectors in equation (5.57) to obtain

$$\begin{bmatrix} N_x \\ N_y \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} H_x^{n'} \\ H_y^{n'} \end{bmatrix} \quad (5.60)$$

which, after some algebraic manipulations leads to

$$\begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix} = \begin{bmatrix} H_x^{n'} & -H_y^{n'} \\ H_y^{n'} & H_x^{n'} \end{bmatrix} \begin{bmatrix} N_x \\ N_y \end{bmatrix} \quad (5.61)$$

Finally, we proceed to apply the half-angle formulas to find the values of $\cos \frac{\psi}{2}$ and $\sin \frac{\psi}{2}$

$$\begin{aligned} \sin \frac{\psi}{2} &= \text{sign}(\sin \psi) \sqrt{(1 - \cos \psi)/2} \\ \cos \frac{\psi}{2} &= \sqrt{(1 + \cos \psi)/2} \end{aligned} \quad (5.62)$$

Once we have computed all the rotation quaternions we put them together to form a single quaternion that maps the orientation of the body frame with respect of the navigation frame.

$$\hat{\mathbf{q}}_n^b = \mathbf{q}_\psi \mathbf{q}_\theta \mathbf{q}_\phi \quad (5.63)$$

> Avoiding singularities

Up to this point, we have said many times that one of the advantages of using quaternions is that, since no trigonometric functions are evaluated, we avoid the singularities associated to them. However the reader may be wondering what happens in equation (5.51) and (5.52) when $\cos \theta = 0$. The truth is that, in this case, there is indeed a singularity. Good news is that we can implement a very simple mechanism to avoid the singularity related to this specific case.

When we detect that the value of $\cos \theta$ coming from (5.44) is close to zero, for example by comparing it with a predefined threshold, that means that the pitch angle is close to either 90° or -90° . What we do is to rotate the magnetic field \mathbf{h}^b and the acceleration \mathbf{u}^b measured vectors around the body coordinate Y_b axis by an angle large enough—e.g. $\alpha = 20^\circ$ —to obtain a rotated version of them:

$$\bar{\mathbf{u}}_{\text{rotated}}^b = \mathbf{q}_\alpha \bar{\mathbf{u}}^b \mathbf{q}_\alpha^{-1} \quad (5.64)$$

$$\bar{\mathbf{h}}_{\text{rotated}}^b = \mathbf{q}_\alpha \mathbf{h}^b \mathbf{q}_\alpha^{-1} \quad (5.65)$$

where \mathbf{q}_α is the quaternion representing the rotation of an angle α about the body Y_b axis given by

$$\mathbf{q}_\alpha = \cos \frac{\alpha}{2} [1 \ 0 \ 0 \ 0] + \sin \frac{\alpha}{2} [0 \ 0 \ 1 \ 0] \quad (5.66)$$

From this point, once we have computed the new values of $\bar{\mathbf{u}}^b$ and $\bar{\mathbf{h}}^b$, the pitch, roll and yaw quaternions are computed and multiplied to form the estimated conjoint offset quaternion $\hat{\mathbf{q}}_{\text{offset}}$ in the same way as it was explained within the whole section. The last step is to multiply $\hat{\mathbf{q}}_{\text{offset}}$ by the quaternion representing the inverse rotation of α to revert the initial rotation of 20° .

$$\hat{\mathbf{q}}_n^b = \hat{\mathbf{q}}_{\text{offset}} \mathbf{q}_\alpha^{-1} \quad (5.67)$$

5.2.2 Integration of angular rate

5.2.2.1 Cartesian approach

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:

$$\alpha(n) = \alpha_0 + \frac{T}{2} \sum_{k=1}^n [\omega_g(k) + \omega_g(k-1)] \quad (5.68)$$

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

$$\alpha(n) = \alpha(n-1) + \frac{T}{2} [\omega_g(n) + \omega_g(n-1)] \quad (5.69)$$

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

The main advantage of this method is that, once the initial orientation is known, it can be used either under low or high intensity motion conditions as the gyroscope measurements are not affected by linear acceleration. However, as it was commented in section 5.1.5, 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 (remember figure 5.6).

This method is, itself, unfeasible if not fused with the accelerometer and magnetometer approach. The growing offset effect can be partially mitigated by periodically resetting the integration using the orientation provided by the accelerometer-magnetometer approach only under low intensity motion. This strategy, often referred as *gating*, is explained in depth in section 5.2.3.

5.2.2.2 Quaternions approach

An alternative approach can also be developed to estimate the orientation quaternion through the integration of angular rate. This method, explained in depth in [70], has been widely employed during the last decades in the navigation field.

Starting from the definition of the derivative, we can compute the value of the quaternion at every time instant using the following expression

$$\begin{aligned} \frac{\partial \mathbf{q}(t)}{\partial t} &= \dot{\mathbf{q}}(t) \\ \mathbf{q}(t) &= \mathbf{q}_0 + \int_0^t \dot{\mathbf{q}}(t') dt' \end{aligned} \quad (5.70)$$

The derivative of the quaternion can be computed as a function of the angular rate as follows

$$\dot{\mathbf{q}}(t) = \frac{1}{2} [\mathbf{q}(t) \omega_q(t)] \quad (5.71)$$

where $\omega_q(t)$ is the pure vector quaternion of the body angular rate vector itself defined as $\omega_q(t) = [0, \omega_{g_x}^b, \omega_{g_y}^b, \omega_{g_z}^b]$.

Equations (5.70) and (5.71) can be translated to discrete time, yielding

$$\begin{aligned} \mathbf{q}(n) &= \mathbf{q}_0 + \sum_{k=1}^n \left[\dot{\mathbf{q}}(k-1) dt + \frac{\dot{\mathbf{q}}(n) - \dot{\mathbf{q}}(n-1)}{2} dt \right] = \\ &= \mathbf{q}_0 + \frac{dt}{2} \sum_{k=1}^n [\dot{\mathbf{q}}(k) + \dot{\mathbf{q}}(k-1)] = \mathbf{q}_0 + \frac{dt}{2} \sum_{k=1}^{n-1} [\dot{\mathbf{q}}(k) + \dot{\mathbf{q}}(k-1)] + \frac{dt}{2} [\dot{\mathbf{q}}(n) + \dot{\mathbf{q}}(n-1)] = \\ &= \mathbf{q}(n-1) + \frac{dt}{2} [\dot{\mathbf{q}}(n) + \dot{\mathbf{q}}(n-1)] \end{aligned} \quad (5.72)$$

where $\mathbf{q}(0) = \mathbf{q}_0$ and

$$\dot{\mathbf{q}}(n) = \frac{1}{2} \mathbf{q}(n-1) \omega_q(n) \quad (5.73)$$

The initial quaternion can be computed using the Factorized Quaternion Algorithm approach which is based on the accelerometer and the magnetometer.

5.2.3 Gating

A strategy that can be followed to mitigate the effects of time growing bias in the integrated angular rate signal is to periodically reset the integration to a reference value which is computed using the accelerometer-magnetometer approach. Therefore, since this approach is only accurate under low intensity motion, we need a motion intensity detector to determine the moment in which the integration can be reset.

Figure 5.7 shows the measured acceleration signals of a body which is moving alternating between high and low intensity motion. Figure 5.8 depicts the figure of merit of the MBCUSUM detector, the detection threshold and the resultant binary marker (which amplitude has been adapted to improve visibility) of such moving body. The figure also shows the acceleration magnitude which is usually employed as the detection signal by some authors [85, 157, 158]. As we can see, the magnitude of the acceleration crosses several times the threshold which would have led the algorithm to erroneously reset the integration with the orientation estimate obtained from the accelerometer-magnetometer approach. Notice how MBCUSUM has a much better performance and crosses the threshold at correct instants.

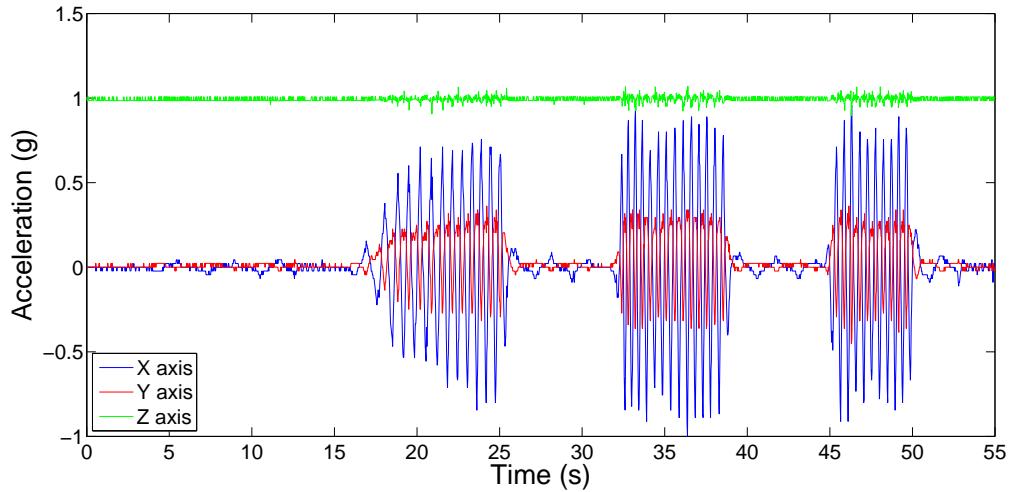


Figure 5.7: Measured acceleration of a moving body with both low and high intensity motion.

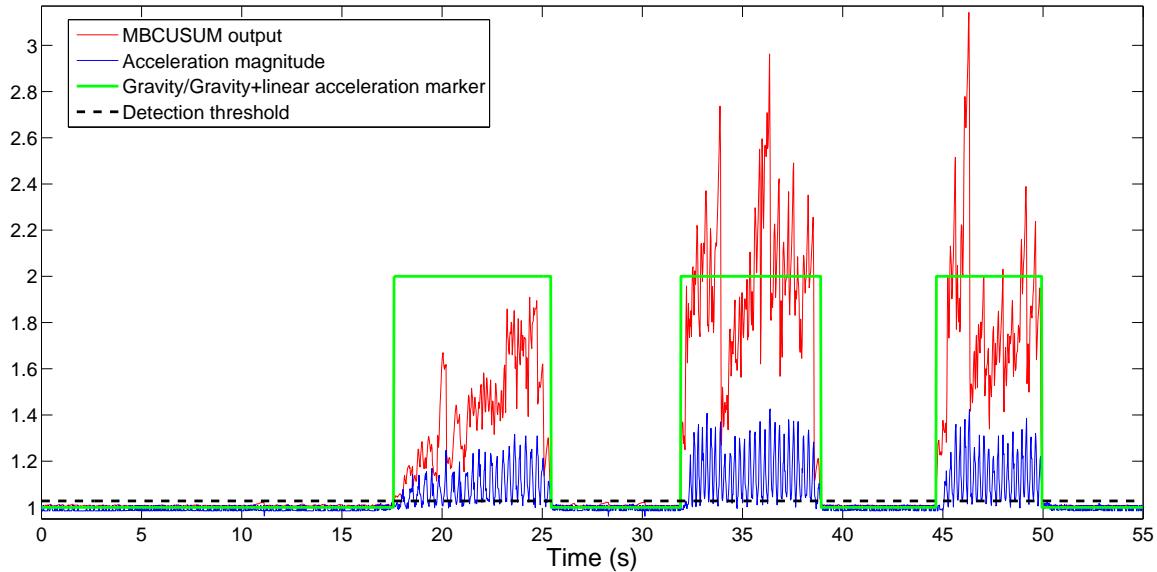


Figure 5.8: Application of MBCUSUM detector to differentiate low intensity motion from high intensity motion (1).

Figure 5.9 shows the yaw angle computed using the accelerometer-magnetometer quaternion approach (computed using FQA), the yaw angle computed using the integrated angular rate quaternion approach—and its reset version—the actual angle reference—computed with a mechanical device presented in section 5.3.3.1 and the MBCUSUM marker (also with modified amplitude and offset to allow visibility) which differentiates between low and high intensity motion. The angle computed

integrating the angular rate is reset only once at the beginning of each low intensity period. To avoid the effect of long low intensity periods, the algorithm could reset the signal not only at the beginning of the period but every t seconds. Such resetting period should not be too short in order to avoid the integrated signal to be too similar to the one computed with FQA. Notice how the yaw estimate computed with FQA is absolutely erroneous during high intensity periods and how the integrated angular rate yaw estimate has a time increasing bias.

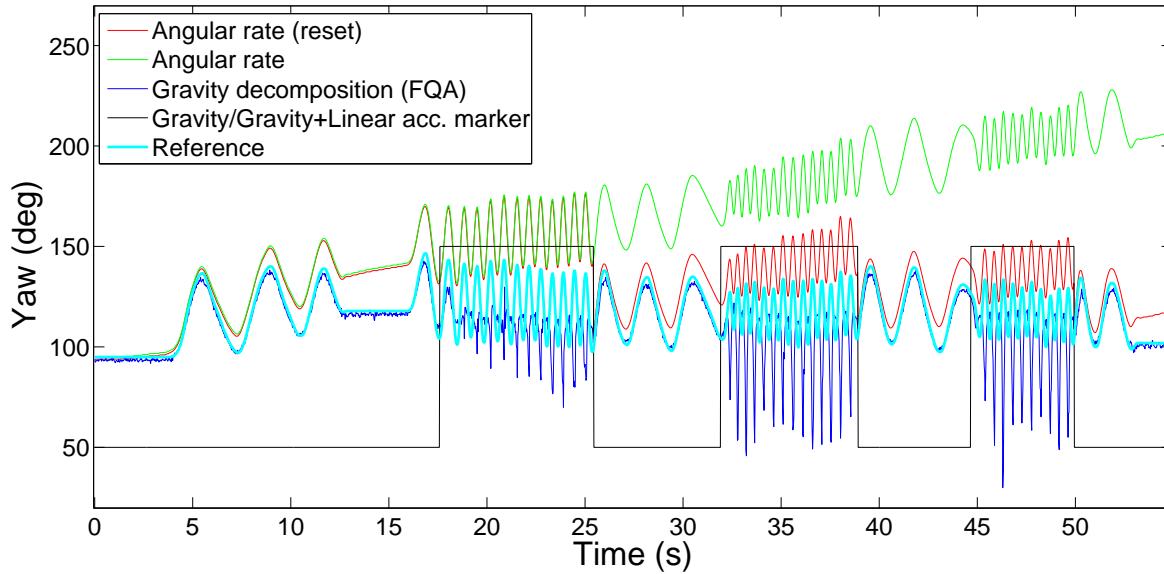


Figure 5.9: Application of MBCUSUM detector to differentiate low intensity motion from high intensity motion (2).

The gating strategy can be also successfully employed to increase the performance of adaptive filters as it will be later explained in section 5.2.4.1.

5.2.4 Sensor fusion

Up to this point we have shown the working principles of the approach based on the projection of reference vectors and the approach based on the integration of angular rate. We have also explained why a solution solely based on any of the two techniques can not be employed to achieve high accuracy attitude estimates. The accelerometer-magnetometer approach works fine under low intensity motion conditions, and the angular rate approach is not affected by high intensity motion but its estimates are only valid during a short period of time until they start diverging. It is clear then, that we need a strategy to combine the benefits of both approaches, a system that relies more on the accelerometer-magnetometer technique when the

intensity of the motion is low, while it relies more on the angular rate technique when the intensity is high. It would also be of great help if the bias of the integrated angular rate could be reduced.

As we explained briefly in section 5.1.6, sensor fusion is a powerful solution to this problem and offers means to implement a system with the aforementioned desired behavior. A popular approach is to estimate some determined aspects of a process and correct the estimation observing the output of a sensor which is used as a reference. A common form of this approach is to estimate various aspects of the angular rate approach (orientation angle and dynamic bias of the angular rate signal or time increasing offset of the integrated angular rate signal) and correct the estimate observing the orientation angle estimate computed using the accelerometer-magnetometer approach. Such procedure can be implemented using Kalman filters, Least Mean Square Filters and Recursive Least Squares filters among other kinds of adaptive filters.

5.2.4.1 Adaptive filtering I: Kalman Filter

The Kalman filter [159], [160], [161] 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:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) + \mathbf{C}\mathbf{u}(t) + \mathbf{I}_n\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) + \mathbf{I}_q\mathbf{v}(t)\end{aligned}\tag{5.74}$$

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.

- I_n is the $(n \times n)$ identity matrix.
- 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:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= F\mathbf{x}(t) + I_n \mathbf{w}(t) \\ \mathbf{z}(t) &= H\mathbf{x}(t) + I_q \mathbf{v}(t)\end{aligned}\quad (5.75)$$

If we apply the forward-Euler approximation:

$$\dot{\mathbf{x}}(t) \simeq \frac{x(t+dt) - x(t)}{dt} \quad (5.76)$$

we get:

$$\begin{aligned}\frac{\mathbf{x}(t+dt) - \mathbf{x}(t)}{dt} &\simeq F\mathbf{x}(t) + I_n \mathbf{w}(t) \\ \mathbf{x}(t+dt) &\simeq [F\mathbf{x}(t) + I_n \mathbf{w}(t)]dt + \mathbf{x}(t) \\ \mathbf{x}(t+dt) &\simeq F\mathbf{x}(t)dt + I_n \mathbf{w}(t)dt + \mathbf{x}(t) \\ \mathbf{x}(t+dt) &\simeq [Fdt + I_n]\mathbf{x}(t) + I_n \mathbf{w}(t)dt\end{aligned}\quad (5.77)$$

If we now discretize the model:

$$\begin{aligned}\mathbf{x}_{k+1} &= [Fdt + I_n]\mathbf{x}_k + I_n dt \mathbf{w}_k \\ \mathbf{x}_{k+1} &= \Phi \mathbf{x}_k + I_n dt \mathbf{w}_k \\ \mathbf{z}_k &= H\mathbf{x}_k + I_q \mathbf{v}_k\end{aligned}\quad (5.78)$$

or how it is usually represented:

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + B \mathbf{w}_{k-1} \quad (5.79)$$

$$\mathbf{z}_k = H\mathbf{x}_k + I_q \mathbf{v}_k \quad (5.80)$$

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.
- $w \in \mathbb{R}^n$ is again the $(n \times 1)$ process noise vector.
- $v \in \mathbb{R}^q$ 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 $x \in \mathbb{R}^n$ of a discrete-time controlled process that is defined by the model in equations (5.79) and (5.80). In addition, w_k and v_k are assumed to be independent of each other, white, and with normal probability distributions

$$p(w) \sim N(0, Q) \quad (5.81)$$

$$p(v) \sim N(0, R) \quad (5.82)$$

where, in turn, Q and 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{x}_k as a linear combination of an a priori (predicted) estimate \hat{x}_k^- and a weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k^-$,

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-). \quad (5.83)$$

The difference in (5.83) is known as the measurement innovation, or the residual. The residual indicates the discordance between the predicted measurement $H\hat{x}_k^-$ and the actual measurement z_k .

The matrix K_k in (5.83) 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 z_k is trusted more, while the predicted measurement $H\hat{x}_k^-$ 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 , Q and R to be constant, the time update equations are as follows:

$$\hat{x}_k^- = \Phi\hat{x}_{k-1}^- \quad (5.84)$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q \quad (5.85)$$

while the measurement update equations are

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (5.86)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (5.87)$$

$$P_k = (I - K_k H) P_k^- \quad (5.88)$$

where P_k^- and 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 (5.85).
 2. Compute K_k by substituting P_k^- , H and R in (5.86).
 3. Compute P_k by substituting K_k and P_k^- in (5.88).
 4. Compute successive values of \hat{x}_k (which is the output of the filter) recursively using (5.84), (5.87) and the computed values of K_k . Start with the given initial estimates \hat{x}_0 and P_0 .

Figure 5.10 shows the flow diagram of the standard approach of the Kalman filter without sensor fusion.

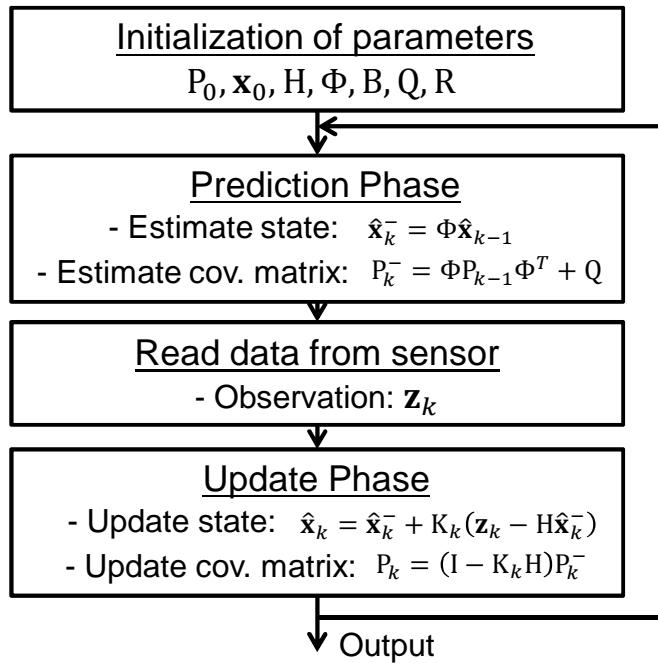


Figure 5.10: General diagram of the standard approach of the Kalman filter without sensor fusion.

➤ Kalman filtering applied to sensor fusion: classic approach

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:

$$\mathbf{x}(t) = \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} \quad (5.89)$$

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 .

$$\text{bias}(t) = \omega_{\text{meas}}(t) - \omega_{\text{actual}}(t) \quad (5.90)$$

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.

$$\mathbf{z}(t) = z(t) = \alpha(t) \quad (5.91)$$

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

$$\frac{d\alpha(t)}{dt} = \omega_{\text{actual}}(t) \quad (5.92)$$

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:

$$\frac{d\alpha(t)}{dt} = \omega_{\text{meas}}(t) - \text{bias}(t) \quad (5.93)$$

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

$$\frac{d\alpha(t)}{dt} = -\text{bias}(t) + w_\alpha(t) \quad (5.94)$$

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:

$$\frac{dbias(t)}{dt} = w_{\text{bias}}(t) \quad (5.95)$$

Then, the model of the system is as follows:

$$\begin{aligned} \mathbf{x}(t) &= \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix}; \\ \frac{d\mathbf{x}(t)}{dt} &= \frac{d}{dt} \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_\alpha(t) \\ w_{\text{bias}}(t) \end{bmatrix}; \\ z(t) &= [1 \ 0] \begin{bmatrix} \alpha(t) \\ \text{bias}(t) \end{bmatrix} + v(t) \end{aligned} \quad (5.96)$$

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

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} \alpha_{k+1} \\ \text{bias}_{k+1} \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -dt \\ 0 & 0 \end{bmatrix} \right] \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} w_{\alpha,k} \\ w_{\text{bias},k} \end{bmatrix}; \\ \mathbf{z}_{k+1} &= [1 \ 0] \begin{bmatrix} \alpha_{k+1} \\ \text{bias}_{k+1} \end{bmatrix} + v_{k+1} \end{aligned} \quad (5.97)$$

or, equivalently:

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} = \left[\begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{k-1} \\ \text{bias}_{k-1} \end{bmatrix} + \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} w_{\alpha,k-1} \\ w_{\text{bias},k-1} \end{bmatrix} \right]; \\ \mathbf{z}_k &= [1 \ 0] \begin{bmatrix} \alpha_k \\ \text{bias}_k \end{bmatrix} + v_k \end{aligned} \quad (5.98)$$

Then, identifying terms, we have:

$$\Phi = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix}, \quad H = [1 \ 0] \quad (5.99)$$

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

$$\hat{x}_k^- = \Phi \hat{x}_{k-1} \quad (5.100)$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q \quad (5.101)$$

UPDATE EQUATIONS

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (5.102)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (5.103)$$

$$P_k = (I - K_k H) P_k^- \quad (5.104)$$

where

- \hat{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

$$Q = \begin{bmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\omega \end{bmatrix} \quad (5.105)$$

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). This parameter can be dynamically modified to improve the performance of the Kalman Filter. What we do is to apply a motion intensity detector and then set a very high variance (high R) when the intensity is high and a low variance when the motion is low. The higher R , the lower Kalman gain and, therefore, less weight is given to the measurement update. In other words, when we detect that the motion is rather high, we tell the filter not to trust the angle measurements offered by the accelerometer-magnetometer by increasing their variance R . Such a variation of the Kalman Filter will be referred as the Gated Kalman Filter. Results of the Gated Kalman Filter are shown later in section 5.3.3.4.
- K_k is the Kalman filter gain.

- $\hat{\mathbf{x}}_k$ 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 (5.100)-(5.104) 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 $\hat{\mathbf{x}}_k^-$ as follows:

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} \alpha_k^- \\ \text{bias}_k^- \end{bmatrix} = \begin{bmatrix} \alpha_{k-1} + \omega_{\text{meas},k} dt \\ \omega_{\text{meas},k} - \text{bias}_{k-1} \end{bmatrix} \quad (5.106)$$

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 (5.105). 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 (5.99).
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 (5.106).
8. Compute the a priori system covariance matrix using equation (5.101).
9. Compute the Kalman gain using equation (5.102), where $z_k = \alpha_k$.
10. Update the state vector using equation (5.103).
11. Update the system covariance matrix using equation (5.104).

12. Return to step 5.

Figure 5.11 shows the flow diagram of the standard approach of the Kalman filter with sensor fusion.

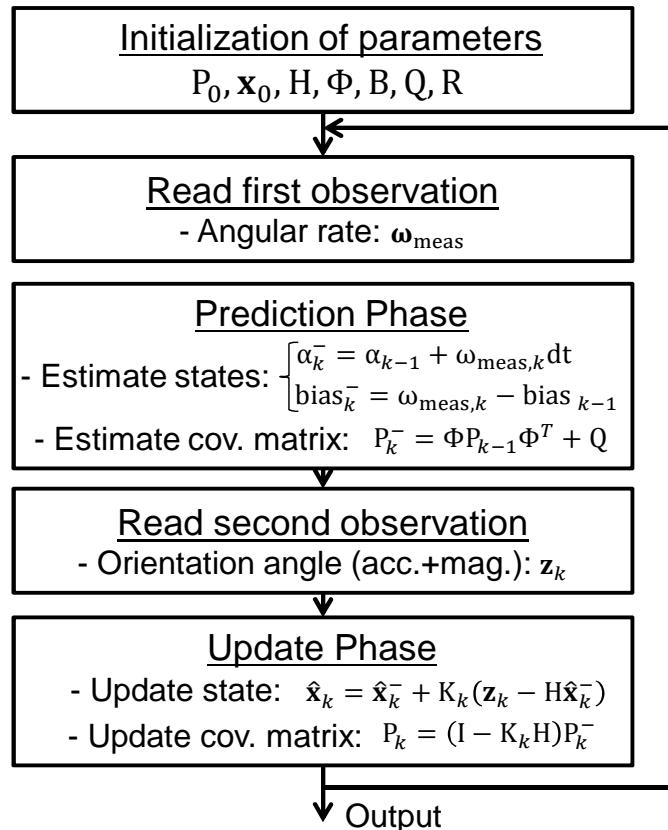


Figure 5.11: General diagram of the standard approach of the Kalman filter with sensor fusion.

➤ Gated Kalman Filter: a new approach based on a dynamical Kalman Filter.

Human body motion is usually variable in terms of intensity. We move with both smooth and intense movements and, therefore, a MIMU attached to our body will measure both low and high angular rate and accelerations. This can be a problem for the accuracy of the classic Kalman Filter approach since both the variances of the process noise and the measurement noise are set at the beginning of the algorithm and remain constant during its execution. If we consider the linear acceleration as one of the elements of the noise disrupting both the process and the measurement, then, it is straightforward to see that smooth motion will introduce low linear acceleration in the measurements, and therefore, low noise. On the other hand, intense motion is originated by large linear accelerations, so the acceleration readings will

be corrupted with these disrupting accelerations which are modeled as noise.

Thus, if we set low process and measurement noises, we are telling the filter that the gravity acceleration prevails the linear acceleration and then, it will rely more on the acceleration readings to estimate the orientation. This can be a problem if the intensity of the motion starts to be changing (which, as we have said, is the usual behavior of human motion) as the adaptation of the filter is burdened by the unchanging process and measurement variances.

We thought that the general performance of the classic Kalman Filter could be improved by dynamically changing both process and measurement variances according to the intensity of the motion. To do so, we first set a motion intensity detector to distinguish between low and high intensity. Then, we set the value of the measurement and process noise between two predefined values according to the value of the intensity indicator. Therefore by setting low measurement and process noises when the subject is moving smoothly and setting high values when it is moving intensely, we improve the adapting capability of the filter, and consequently, the precision of the orientation estimation. Figure 5.12 shows the general diagram of the Gated Kalman Filter.

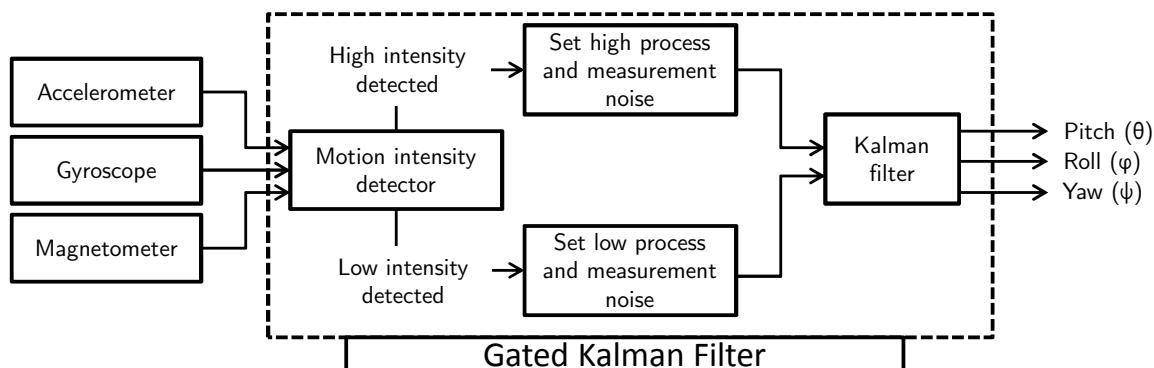


Figure 5.12: General diagram of the proposed Gated Kalman Filter.

The next section presents an alternative sensor fusion approach which is based on Least Mean Squares (LMS) and Recursive Least Squares (RLS) Filters.

5.2.4.2 Adaptive filtering II: Least Mean Squares (LMS) and Recursive Least Squares (RLS) filters

Least Mean Squares Filters (LMS) are linear adaptive filtering algorithms [162], which, in general consists of two basic processes:

1. A filtering process, which involves:

- Computing the output of a linear filter in response to an input signal.
 - Generating an estimation error by comparing this output with a desired response
2. An adaptive process, which involves the automatic adjustment of the parameters of the filter in accordance with the estimation error.

The combination of these two processes working together constitutes a feedback loop. A significative feature of the LMS algorithms family is its simplicity since it does not require measurements of the pertinent correlation functions, nor does it require matrix inversions.

➤ Standard LMS

Standard LMS algorithm [163] is the simplest and first algorithm to be developed of the LMS algorithms family. During the filtering process, the desired response $d(n)$ is supplied for processing alongside the tap-input vector $\mathbf{u}(n)$. Given this input, the transversal filter produces an output $\hat{d}(n)$ used as an estimate of the desired response $d(n)$. Accordingly, we may define an estimation error $e(n)$ as the difference between the desired response and the actual filter output.

We assume that the desired response $d(n)$ is linearly related to the input vector $\mathbf{u}(n)$ by a multiple regression model whose parameter vector is unknown, hence the need for adaptive filtering.

In the adaptive weight-control mechanism, a scalar version of the inner product of the estimation error $e(n)$ and the tap input $\mathbf{u}(n-k)$ is computed for $k = 0, 1, \dots, M-1$ where M is the number of coefficients of the filter. The result to the previous product defines the correction $\delta\hat{\mathbf{w}}_k(n)$ applied to the tap weight $\hat{\mathbf{w}}_k(n)$ at the iteration $n+1$. The scaling factor used in this computation is denoted by a positive quantity μ known as the step-size parameter of the filter.

The feedback loop acting around the tap weights behaves like a low-pass filter, whose average time constant is inversely proportional to the step-size parameter μ . Hence by assigning a small value to μ , the adaptive process is made to progress slowly, and the effects of gradient noise on the tap weights are largely filtered out.

The standard LMS algorithm is, then, computed as follows,

1. Parameters:

- M : number of taps, that is, the length of the filter.
- μ : step-size parameter.

2. Initialization:

- Set $\hat{\mathbf{w}}_0 = \mathbf{0}$ unless prior knowledge of the tap-weight vector $\hat{\mathbf{w}}_k(n)$ is available. In that case use it to select an appropriate value for $\hat{\mathbf{w}}_0$.

3. Given data:

- $\mathbf{u}(n)$: ($M \times 1$) tap-input vector at time n . $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$.
- $d(n)$: desired response at time n .

4. Data to be computed:

- $\hat{\mathbf{w}}_{n+1}$: estimate of tap-weight vector at time $n + 1$.

5. Computation: For $n = 0, 1, 2, \dots$, compute

$$\mathbf{e}(n) = d(n) - \hat{\mathbf{w}}_n^H \mathbf{u}(n) \quad (5.107)$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu \mathbf{u}(n) \mathbf{e}^*(n) \quad (5.108)$$

where the superscripts H and * in equations (5.107) and (5.108) denote the complex conjugate transpose matrix and the complex conjugate respectively.

➤ Normalized LMS (NLMS)

Normalized LMS [164] normalizes by $\|\mathbf{u}(n)\|^2$ equation (5.108) altering the magnitude, but not the direction, of the estimated gradient vector. This leads to a faster convergence. The update equation is, then,

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu \frac{\mathbf{u}^*(n) \mathbf{e}(n)}{\|\mathbf{u}(n)\|^2} \quad (5.109)$$

where μ is the step size, $e(n)$ is the error at instant n , $\mathbf{u}(n)$ is the input signal of the filter, and $\|\cdot\|$ denotes the Euclidean norm.

➤ Momentum Normalized LMS (MNLMS)

The Momentum Normalized Least Mean Squares [165] filter tries to normalize the weighted average square error function and leads to a different weight vector update equation as follows

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \frac{\mu \mathbf{u}^*(n) e(n)}{\mu \|\mathbf{u}(n)\|^2 + 1} \quad (5.110)$$

where μ is the step size, $e(n)$ is the error at instant n and $\mathbf{u}(n)$ is the input signal of the filter.

➤ Standard RLS

The Recursive least squares (RLS) adaptive filter [163] is an algorithm which recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is in contrast to the LMS filter that aims to reduce the mean square error.

Its rate of convergence is typically an order of magnitude faster than that of the simple LMS filter due to the fact that the RLS filter whitens the input data by using the inverse correlation matrix of the data, assumed to be of zero mean. However, it is achieved at the expense of an increase in computational complexity, specifically the LMS filter requires $2N + 1$ operations per filter update, whereas the RLS filter requires $2.5N^2 + 4N$, being N the number of filter coefficients. Its standard formulation is computed as follows

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \alpha(n)\mathbf{g}(n) \quad (5.111)$$

where

$$\mathbf{g}(n) = \frac{\mathbf{z}(n)}{\lambda + \mathbf{u}^T(n)\mathbf{z}(n)} \quad (5.112)$$

$$\alpha(n) = d(n) - \hat{\mathbf{w}}_{n-1}^T \mathbf{u}(n) \quad (5.113)$$

$$\mathbf{z}(n) = \mathbf{P}(n-1)\mathbf{u}^*(n) \quad (5.114)$$

where \mathbf{P} denotes the inverse of the ($M \times M$) correlation matrix of the input vector $\mathbf{u}(n)$ at instant n . \mathbf{P} is updated through the following equation:

$$\mathbf{P}(n) = \lambda^{-1}[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{z}^H(n)] \quad (5.115)$$

where $0 < \lambda \leq 1$ is known as the forgetting factor which gives exponentially less weight to older error samples and is intended to ensure that data in the distant past are forgotten so to afford the possibility of following the statistical variations of the observable data when the filter operates in a nonstationary environment. In our case, we will use the *growing window RLS* which sets $\lambda = 1$.

To build a computational RLS algorithm, the next steps must be followed:

1. Initialization:

- $\hat{\mathbf{w}}_0 = \mathbf{0}$
- $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$ where δ is called the regularization parameter and is a small or large positive constant for high and low SNR respectively.

2. Computation: For each instant of time $n = 1, 2, \dots$,

- Compute $\mathbf{z}(n)$ using (5.114).

- Compute $\mathbf{g}(n)$ using (5.112).
- Compute $\alpha(n)$ using (5.113).
- Update the filter weight $\hat{\mathbf{w}}_n$ vector using (5.111).
- Update $P(n)$ using (5.115).

➤ Householder RLS (HRLS)

The Householder RLS [166] algorithm is a variation of the standard RLS filter that is based on the Householder transformation [167]. Its scheme avoids, almost entirely, costly square roots and divisions (present in other numerically well-behaved adaptive LS schemes) and provides directly the estimates of the unknown system coefficients. Furthermore, it offers enhanced parallelism, which leads to efficient implementations. The computing process is as follows,

$$\mathbf{V} = \mathbf{G}\mathbf{u}(n) \quad (5.116)$$

$$\mathbf{U} = \mathbf{G}^T \mathbf{V} \quad (5.117)$$

$$\gamma = \lambda + \mathbf{V}'\mathbf{V} \quad (5.118)$$

where the operator $(\cdot)'$ represents the complex conjugate transposition of the matrix (\cdot) .

$$\zeta = \frac{1}{\gamma + \sqrt{\lambda + \gamma}} \quad (5.119)$$

Now the filter gain can be computed,

$$\mathbf{K} = \frac{1}{\gamma \mathbf{U}}. \quad (5.120)$$

Next, the square root covariance matrix \mathbf{G} is updated,

$$\mathbf{G}(n) = \frac{1}{\sqrt{\lambda}} \mathbf{G}(n-1) - \zeta \mathbf{V} \mathbf{U}' \quad (5.121)$$

And finally the weight vector is updated,

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mathbf{e}(n)^* \mathbf{K}' \quad (5.122)$$

where again λ is the forgetting factor, $\mathbf{e}(n)$ is the error at instant n and $\mathbf{u}(n)$ is the input of the filter.

➤ QR Decomposition RLS (QRD-RLS)

The widespread acceptance of RLS filters has been impeded by a sometimes unacceptable numerical performance in limited precision environments. This degradation of performance is especially noticeable for the family of techniques collectively known as fast RLS filters. These fast algorithms are typically characterized by an absence of matrix multiplications and require $O(N)$ operations per sampling interval, where N is the number of coefficients in the filter. Finite precision implementations of these fast RLS filters have sometimes been observed to be numerically unstable. A possible solution comes from applying stable orthogonal linear algebraic transformations to the original RLS problem. These linear algebraic transformations are used to triangularize the multiplying matrix in a set of simultaneous linear equations.

One popular method for producing such a triangular structure is the QR decomposition [168], [169]. QR decomposition transforms the original RLS problem, which uses data covariance values, into a problem that uses only transformed data values. This, in turn, causes the numerical dynamic range of the transformed computational problem to be reduced to one-half the dynamic range of the original RLS problem [170], which has immediate benefit concerning stability considerations. The filter weight equation is calculated by applying

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \frac{\mathbf{c}(n)}{b(n)} e(n) \quad (5.123)$$

where

$$\mathbf{c}(n) = \frac{\mathbf{R}^{-1}(n-1)\mathbf{a}(n)}{b(n)\sqrt{\lambda}} \quad (5.124)$$

$$\mathbf{a}(n) = \frac{\mathbf{R}^{-T}(n-1)\mathbf{u}(n)}{\sqrt{\lambda}} \quad (5.125)$$

$$b(n) = \sqrt{1 + \mathbf{a}(n)^T \mathbf{a}(n)} \quad (5.126)$$

➤ A new sensor fusion approach based on LMS and RLS filtering

After the theoretical introduction to the LMS and RLS filters, we can proceed to develop a sensor fusion approach using such filtering algorithms. As said in the theoretical introduction, in LMS/RLS algorithms we can assume the desired response $d(n)$ is linearly related to the input vector $\mathbf{u}(n)$.

By letting $d(n)$ be the accelerometer+magnetometer angle signal $\alpha_{ah}(n)$ and $\mathbf{u}(n)$ be the gyroscope angle signal $\alpha_g(n)$, we are satisfying that condition since $\alpha_{ah}(n)$ and $\alpha_g(n)$ are related linearly.

LMS and RLS filters are widely used to cancel undesired components in signals,

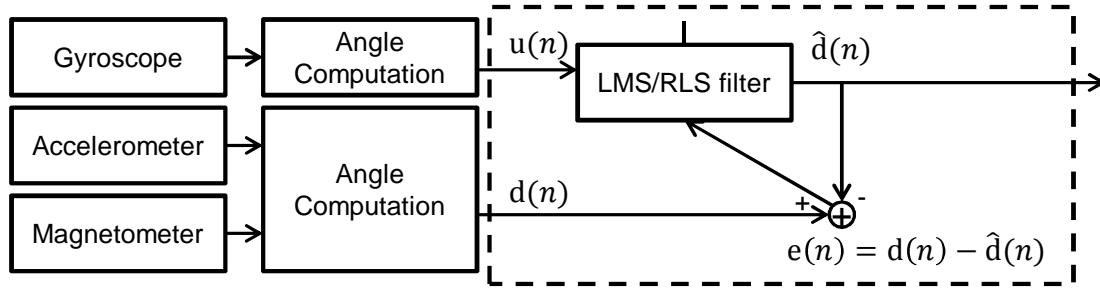


Figure 5.13: Diagram of the LMS/RLS filter sensor fusion approach.

such as noise and echo. Therefore, we will use the gyroscope signal as the input to be filtered and the accelerometer signal as the desired signal. This configuration will remove the gyroscope's dynamic bias while it will filter the accelerometer's noise, since resultant LMS and RLS filters will have low-pass nature in almost all configurations.

Figure 5.13 shows the diagram of the developed LMS/RLS sensor fusion approach.

5.3 Experiments

After the long theoretical introduction to the fundamentals of attitude estimation based on MARG sensors, we now proceed to explain the experiments that we have carried out. The main goal of the experiments is to carry out a complete comparative study among all the aforementioned approaches.

5.3.1 General design

The general design of the experiments is very similar to those carried out in chapter 4. The first part of the experiments consists of a Monte Carlo simulation that applies all algorithms over a large set of synthetic signals to ensure statistical robustness. To that purpose, we developed a signal synthesizer that mimics angle signals computed using the accelerometer-magnetometer and the angular rate approaches. As with the motion intensity detectors, a parameter optimization procedure is carried out for every algorithm during each iteration.

The second part of the experiments consists of the application of the algorithms over a smaller set of real signals gathered using a Wagyromag MIMU.

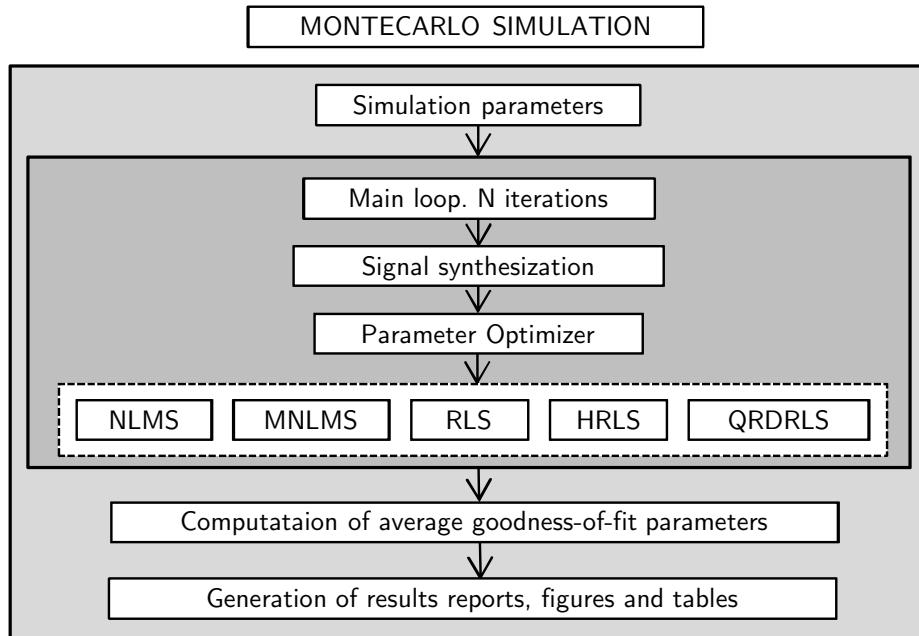


Figure 5.14: Diagram of the Montecarlo Simulation for attitude estimation methods (synthetic signals).

5.3.2 Synthesized signals

It is complicated to know the exact actual angle for every position. Different systems can be used to estimate the real signal, some of them showing great accuracy. Among them, optical trackers [6] and systems based on mechanical systems are good solutions. However, even though the estimates that these systems can be very accurate, it is not possible to know the exact actual signal unless we synthesize it and define the rest of the signals from it.

Along this section, we will explain how the signal synthesizer and the parameter optimizer work and how they are put together into a Monte Carlo simulation. Figure 5.14 shows the structure of the Monte Carlo simulation carried out using synthetic signals. The structure is as follows:

1. Definition of the general parameters of the Monte Carlo simulations (number of iterations, status flags, etc.).
2. Definition of the signal synthesizer parameters (signal length, noise amplitude, number of activity periods, etc.).
3. Definition of optimizer parameters (starting and ending values of grid search, goodness-of-fit indicators, etc.).
4. Start of Monte Carlo simulation:

- 4.1 Synthesization of actual angle signal, accelerometer-magnetometer angle signal and angular rate angle signal.
- 4.2 Parameter optimization:
 - 4.2.1 NLMS.
 - 4.2.2 MNLMS.
 - 4.2.3 RLS.
 - 4.2.4 HRLS.
 - 4.2.5 QRDRRLS.
- 4.3 Computation of goodness-of-fit indicators and optimal parameters update.
- 5. Construction of figures, tables and reports.

5.3.2.1 Signal synthesizer

The signal synthesizer builds four angle signals, namely:

- *Actual angle signal*: It is a sine-like signal which takes values from the $[-\pi, \pi]$ interval. It has two predefined frequencies which emulate low intensity and high intensity motion. The starting and ending points of the low and high intensity motion are computed randomly.
- *Acceleration-Magnetic field decomposition angle signal*: It is a signal that mimics angle signals computed with the accelerometer-magnetometer approach. This signal is close to the values of the actual angle signal during low intensity periods and presents large errors during high intensity motion. This signal is built by increasing the amplitude of the actual signal during high intensity motion periods and adding random noise to simulate noise at the output of the accelerometer and magnetometer.
- *Integration of angular rate angle signal*: It is a signal that mimics the angle signals computed through the integration of measured angular rate. This signal is built by adding a time increasing bias to the actual angle signal.
- *Integration of angular rate angle signal (reset version)*: It is a signal which is computed from the regular integrated angular rate angle signal by resetting it at the start of every period of low intensity motion. The intensity marker is computed using the MBCUSUM detector.

We can also select the intensity of the signals between three possibilities: only high intensity, only low intensity and low intensity combined with high intensity.

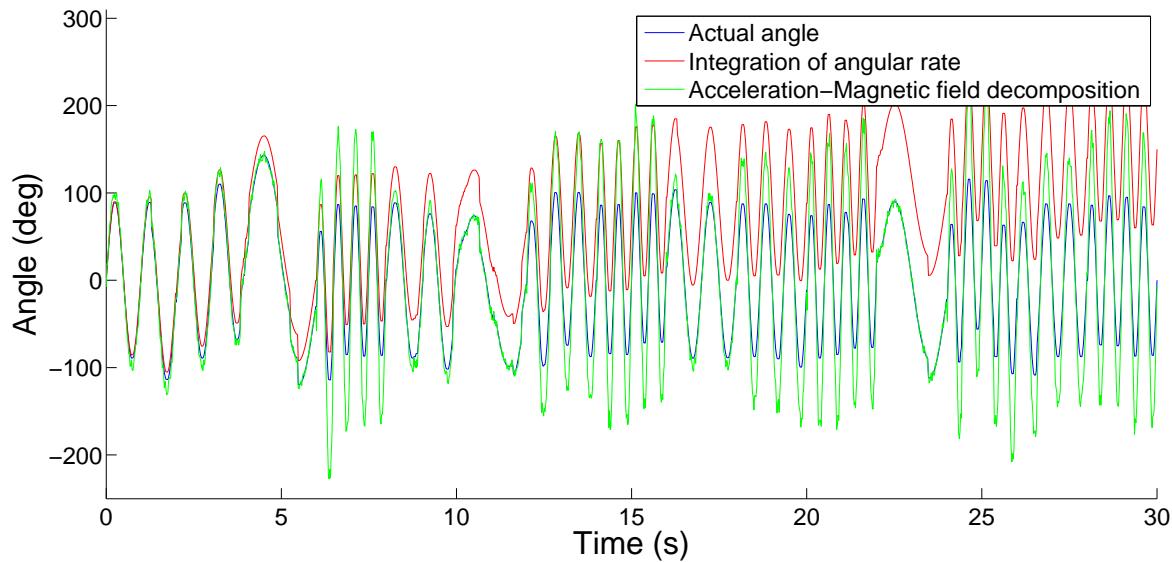


Figure 5.15: A set of synthesized angle signals.

Figures 5.15 and 5.16 show a set of synthesized signals and a set of real signals gathered using a Wagyromag respectively. Notice how synthetic signals are very similar to those gathered with a real MIMU.

5.3.2.2 Parameter optimizer

The parameter optimizer is called after building the set of synthetic signals. It carries out a grid search optimization procedure to find the optimal values of the parameters of each one of the sensor fusion algorithms. As it happened with experiments in section 4, we originally developed optimization routines that were based on the Gauss-Newton algorithm. Even though the minimization of the cost function was performed in a fast way in many executions, there still were a large number of executions in which the algorithm did not converge. In order to ease convergence of the Gauss-Newton procedure the user has to carefully select the initial parameters of the sensor fusion algorithm. This was a problem since some of the algorithms showed to be not very robust in terms of parameter repeatability. Therefore, to avoid the Monte Carlo simulation from getting stuck we opted to implement an optimization procedure based on grid-search.

The optimizer carries proceeds in the following way in every iteration:

1. Select the value of algorithm's input parameters from the predefined parameter vector.
2. Execute the sensor fusion algorithm using the integrated angular rate and the

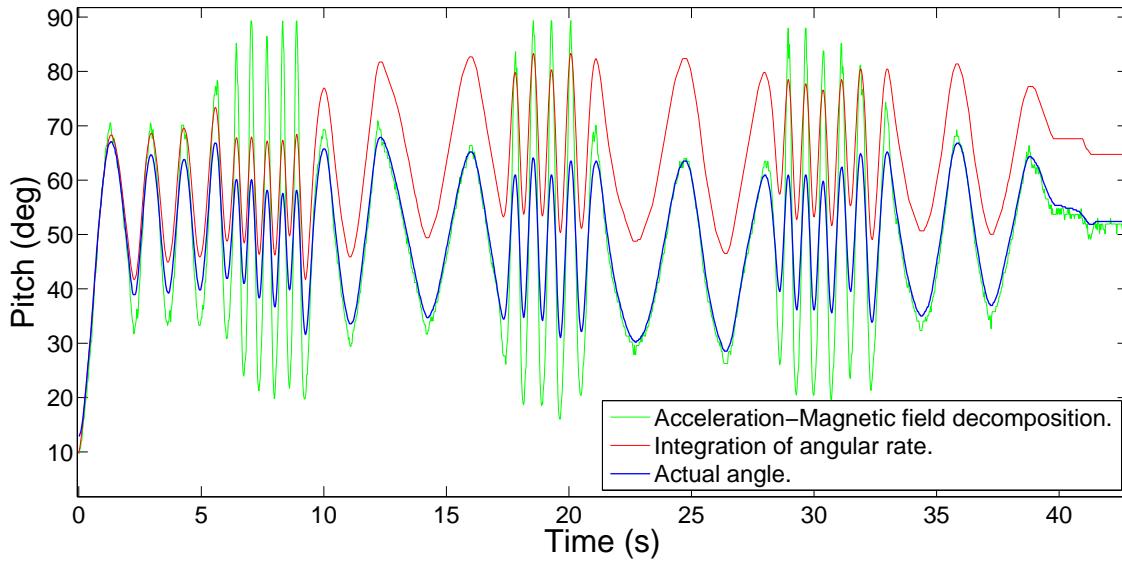


Figure 5.16: A set of real gathered angle signals.

accelerometer-magnetometer synthesized angle signals as inputs.

3. Compute the Root Mean Square Error (RMSE) between the actual angle signal and the output of the sensor fusion algorithm.
4. Store the computed RMSE in a vector.

Once it has executed the algorithm using all the values included in the parameter vectors, it searches the minimum RMSE and the associated input parameters and saves them in a vector which length is equal to the number of iterations of the Monte Carlo simulation.

This optimization procedure is carried out for each one of the five adaptive filtering algorithms.

5.3.2.3 Structure of the experiments

We have first carried out a Monte Carlo simulation of $N=50$ iterations synthesizing only low intensity signals and high intensity signals. The goal is to study the behavior of the algorithms in extreme situations in which high dynamic acceleration is being sensed during a large period of time and, on the other hand, situations in which the body being monitored is moving slowly with low dynamic acceleration. We will also obtain the average optimal parameters for both situations. Then, another Monte Carlo simulation is carried out, this time synthesizing signals containing both low and high intensity periods. Now the goal is to observe how the

algorithms adapt to signals of changing intensity with a set of fix input parameters.

At the end of the last simulation, we generate three new sets of 100 signals and test all the algorithms with the average optimal parameters. This way, we aim to study the validity and generality of the computed optimal parameters by applying the algorithms over a set of signals which has not been used to optimize the parameters.

5.3.2.4 Results

When the Monte Carlo simulation is over we have a vector having N RMSE different values for each one of the algorithms, where N is the number of iterations of the simulation. Then, the average RMSE is computed and stored. Analogously, each algorithm has also vector of N elements for each one of its parameters which is also averaged. Five algorithms were tested—NLMS, MNLMS, RLS, HRLS, QRDRRLS—with two different input combinations, one using the normal rate angle signal and the other one using its reset version.

Results are presented in the following order:

1. Sensor fusion algorithms applied on high intensity signals:
 - Parameter optimization procedure. Figures 5.17-5.21.
 - Average RMSE and average optimal parameters. Table 5.2.
 - Estimated orientation angles. Figures 5.22 and 5.23.
2. Sensor fusion algorithms applied on low intensity signals:
 - Parameter optimization procedure. Figures 5.24-5.28.
 - Average RMSE and average optimal parameters. Table 5.3.
 - Estimated orientation angles. Figures 5.29 and 5.30.
3. Sensor fusion algorithms applied on signals having both low and high intensity:
 - Parameter optimization procedure using grid search. Figures 5.31-5.35.
 - Average RMSE and optimal parameters. Table 5.4.
 - Estimated orientation angles. Figures 5.36 and 5.37.
4. Application of algorithms using optimal parameters on 100 new synthetic signals that were not used in the optimization procedure:
 - High intensity signals vs. low intensity signals. Figures 5.38-5.42.

- Low and high intensity signals. Figures 5.43-5.47.

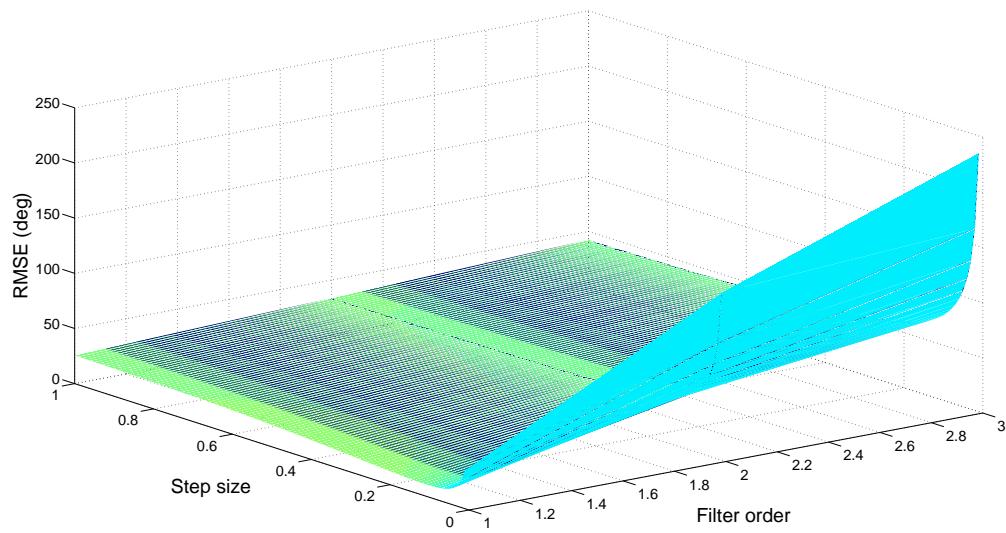


Figure 5.17: Optimization of input parameters. NLMS. High intensity synthetic signals.

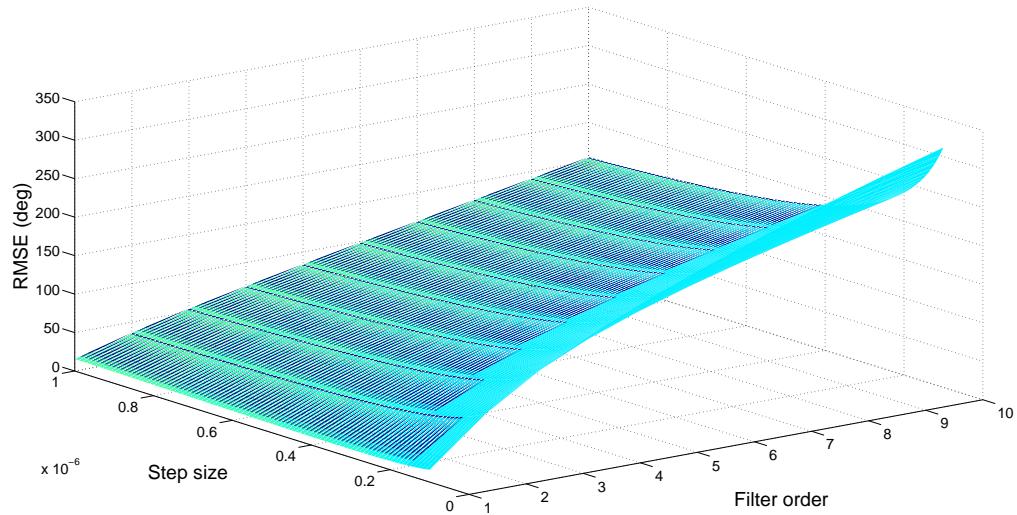


Figure 5.18: Optimization of input parameters. MNLMS. High intensity synthetic signals.

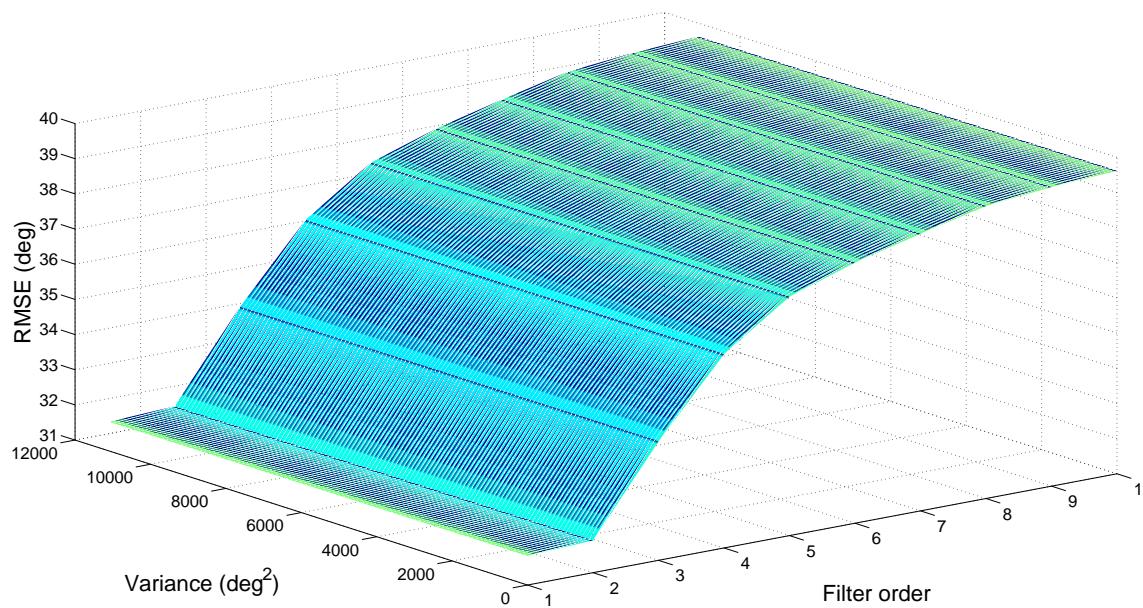


Figure 5.19: Optimization of input parameters. RLS. High intensity synthetic signals.

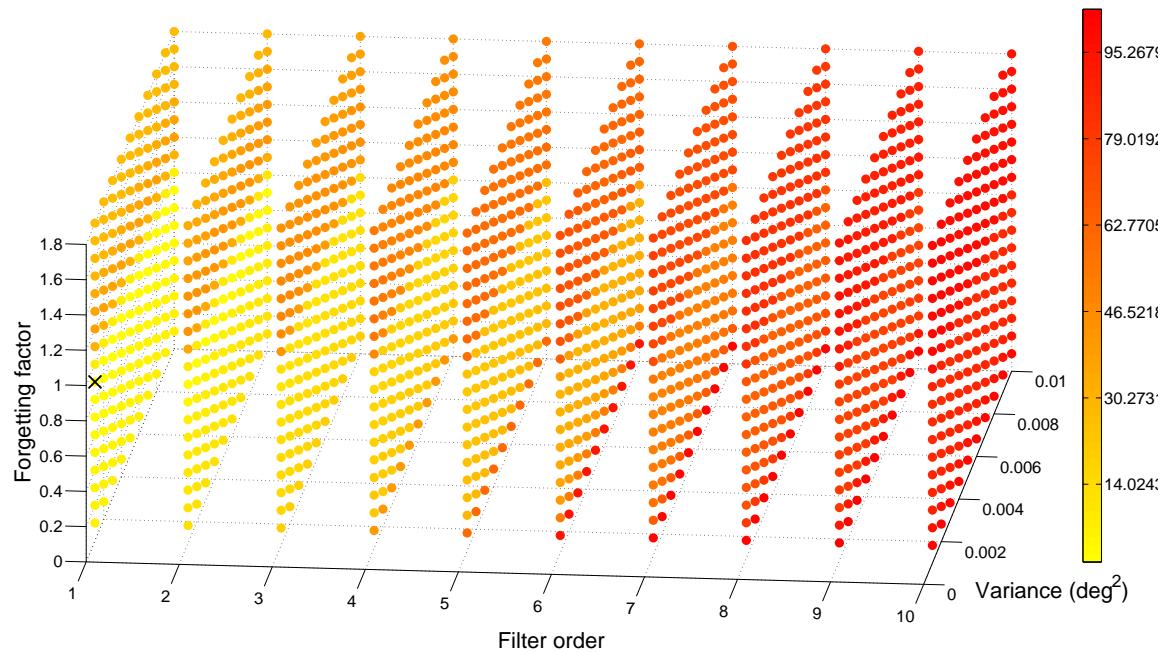


Figure 5.20: Optimization of input parameters. HRLS. High intensity synthetic signals. The cross indicates the optimal point of operation.

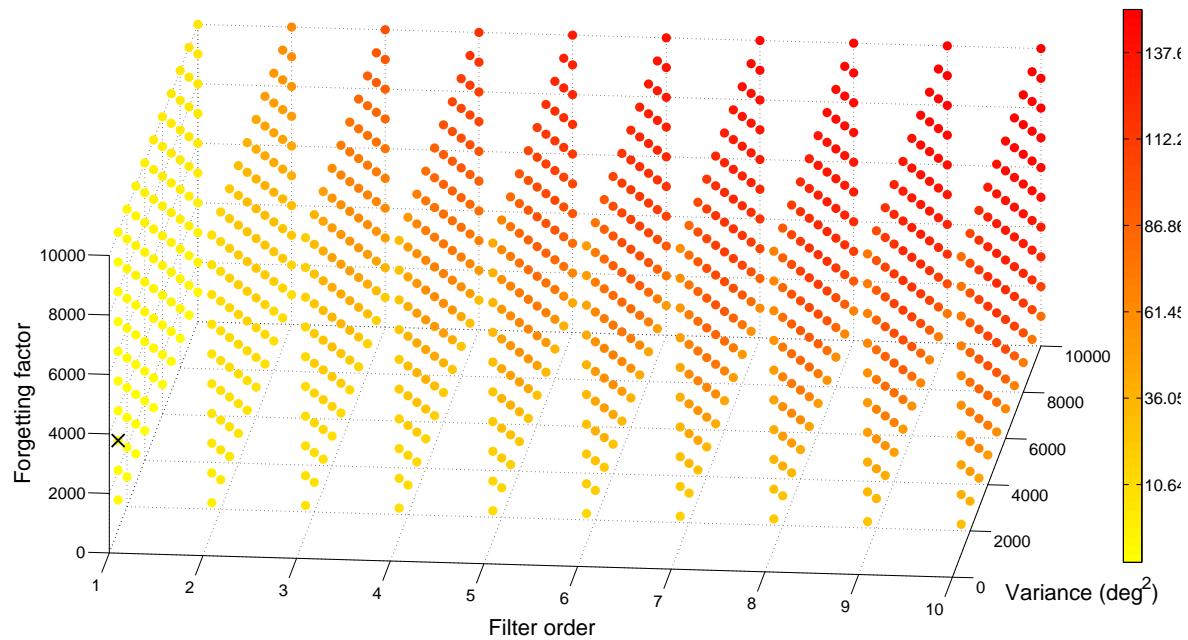


Figure 5.21: Optimization of input parameters. QRDRRLS. High intensity synthetic signals. The cross indicates the optimal point of operation.

Table 5.2: Results of the Monte Carlo simulation ($N=50$). High intensity synthetic signals. Average RMSE and associated parameters.

Algorithm	Average RMSE	First Parameter	Second Parameter	Third Parameter
Acc-Mag	30.3316 ± 0.9881	—	—	—
NLMS	10.5963 ± 2.5222	$L=1.0000 \pm 0.0000$	$\mu=0.0274 \pm 0.0072$	—
MNLMS	12.7064 ± 2.4052	$L=1.0000 \pm 0.0000$	$\mu=2.5600e-7 \pm 5.8134e-8$	—
RLS	31.8950 ± 4.0899	$L=1.0000 \pm 0.0000$	$\delta=8.7750e3 \pm 3.0952e3$	—
HRLS	16.5855 ± 1.6179	$L=1.0000 \pm 0.0000$	$\lambda=0.9000 \pm 8.9720e-16$	$\delta=0.0010 \pm 6.5713e-19$
QRDRRLS	12.7252 ± 2.4115	$L=1.0000 \pm 0.0000$	$\lambda=3.7400e3 \pm 1.2257e3$	$\delta=1.2800e3 \pm 8.3339e2$

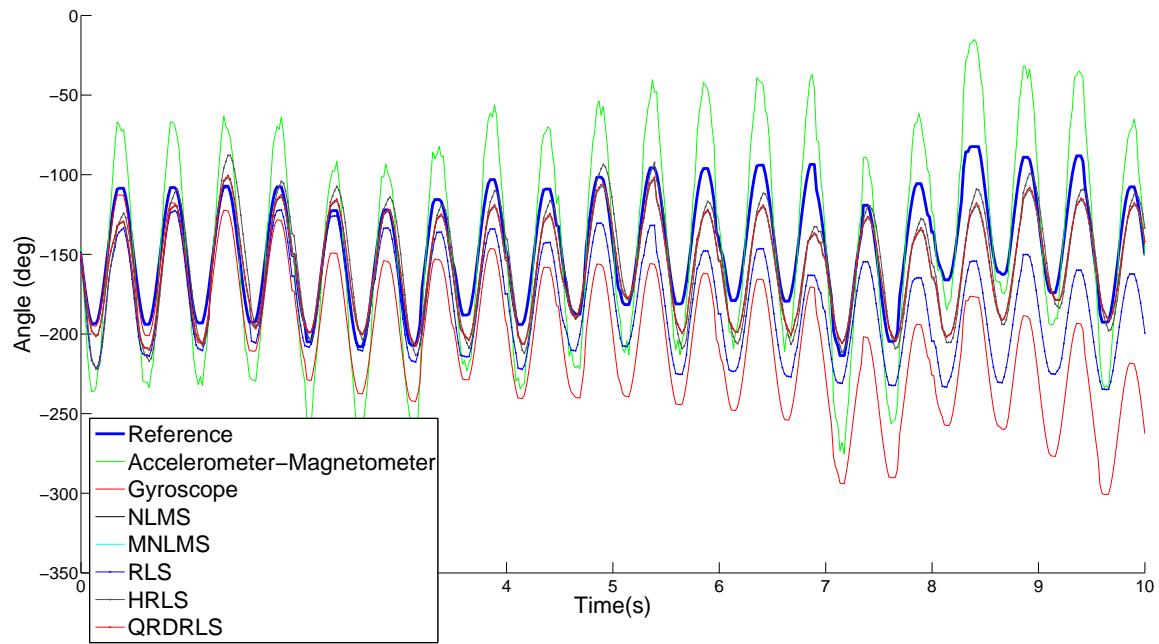


Figure 5.22: Attitude angle computed using sensor fusion algorithms applied on high intensity synthetic signals.

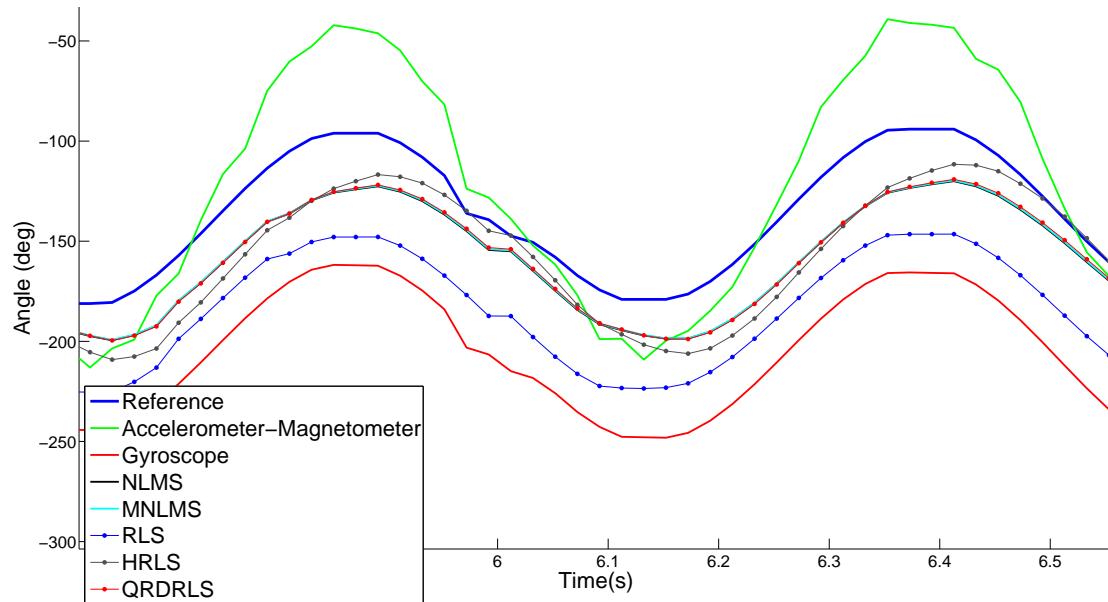


Figure 5.23: Attitude angle computed using sensor fusion algorithms applied on high intensity synthetic signals (zoomed in).

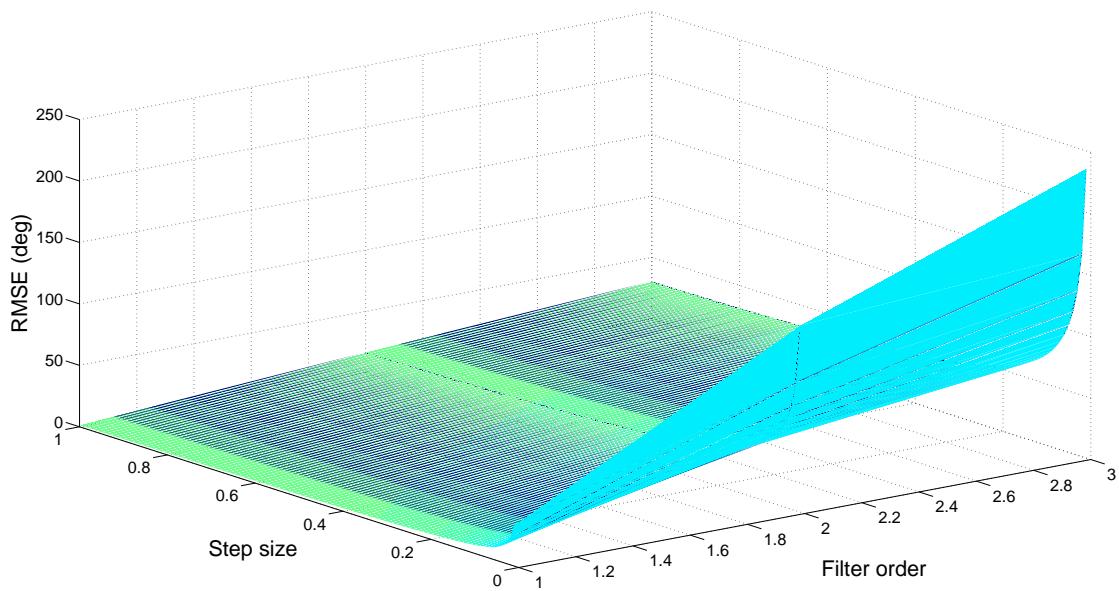


Figure 5.24: Optimization of input parameters. NLMS. Low intensity synthetic signals.

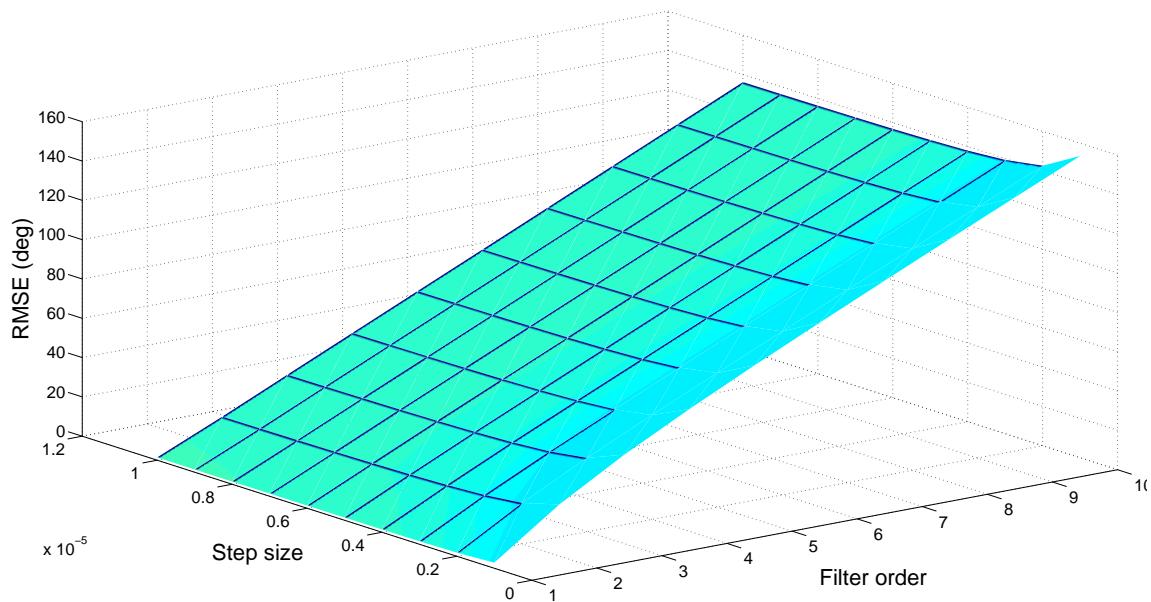


Figure 5.25: Optimization of input parameters. MNLMS. Low intensity synthetic signals.

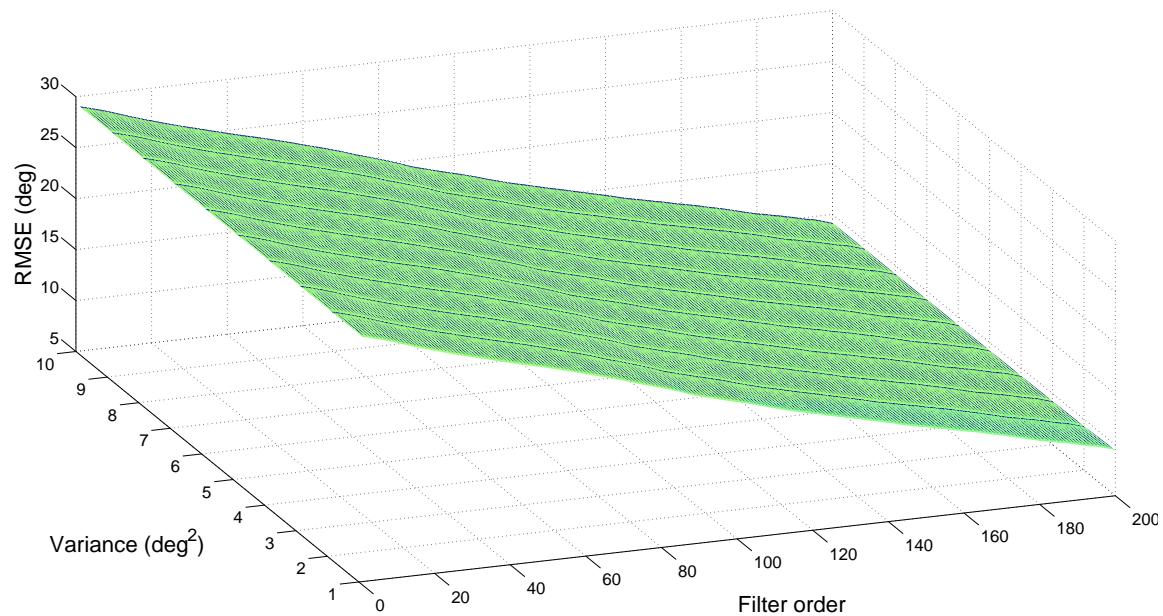


Figure 5.26: Optimization of input parameters. RLS. Low intensity synthetic signals.

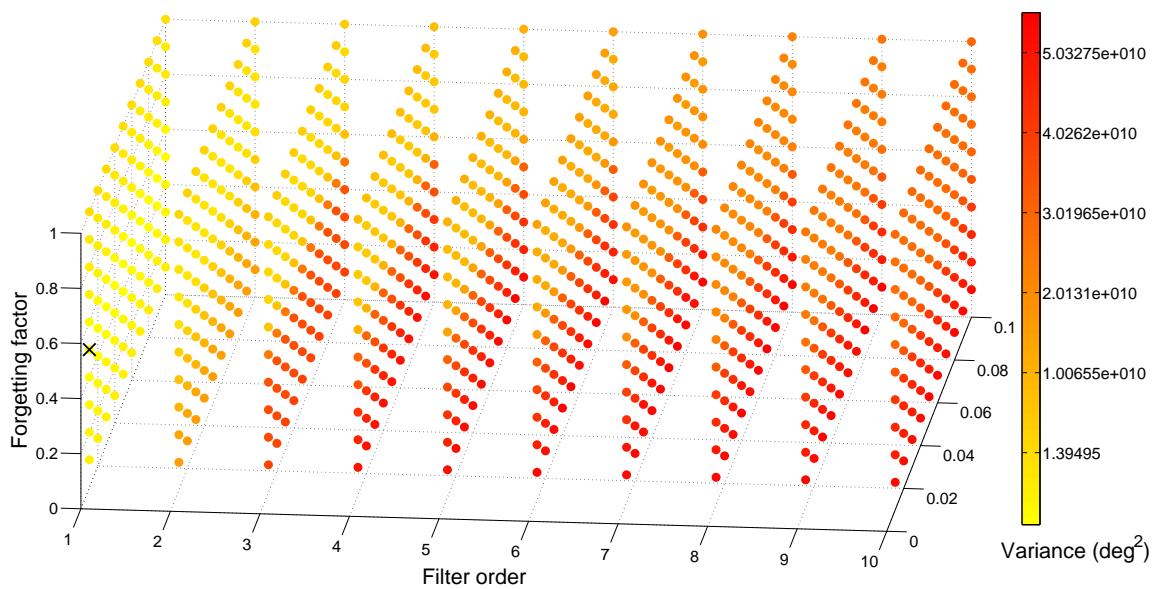


Figure 5.27: Optimization of input parameters. HRLS. Low intensity synthetic signals. The cross indicates the optimal point of operation.

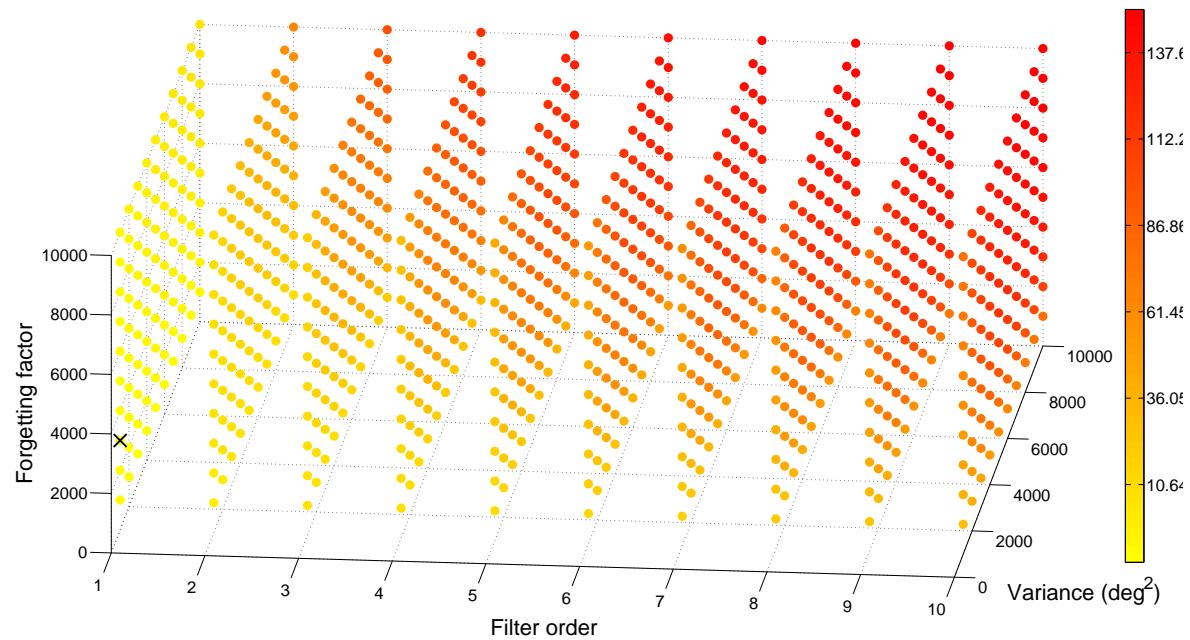


Figure 5.28: Optimization of input parameters. QRDRRLS. Low intensity synthetic signals. The cross indicates the optimal point of operation.

Table 5.3: Results of the Monte Carlo simulation ($N=50$). Low intensity synthetic signals. Average RMSE and associated parameters.

Algorithm	Average RMSE	First Parameter	Second Parameter	Third Parameter
Acc-Mag	1.7416 ± 0.0323	—	—	—
NLMS	1.2940 ± 0.0633	$L=1.0000 \pm 0.0000$	$\mu=0.4816 \pm 0.0226$	—
MNLMS	1.3078 ± 0.0970	$L=1.0000 \pm 0.0000$	$\mu=7.0600e-6 \pm 1.4902e-6$	—
RLS	14.3433 ± 3.6394	$L=2.0000e2 \pm 0.0000$	$\delta=9.9600 \pm 0.19795$	—
HRLS	1.3248 ± 0.0756	$L=1.0000 \pm 0.0000$	$\lambda=0.5040 \pm 0.0282$	$\delta=0.0100 \pm 3.5047e-18$
QRDRRLS	1.3066 ± 0.0781	$L=1.0000 \pm 0.0000$	$\lambda=1.1260e3 \pm 2.4143e2$	$\delta=1.3680e2 \pm 35.8250$

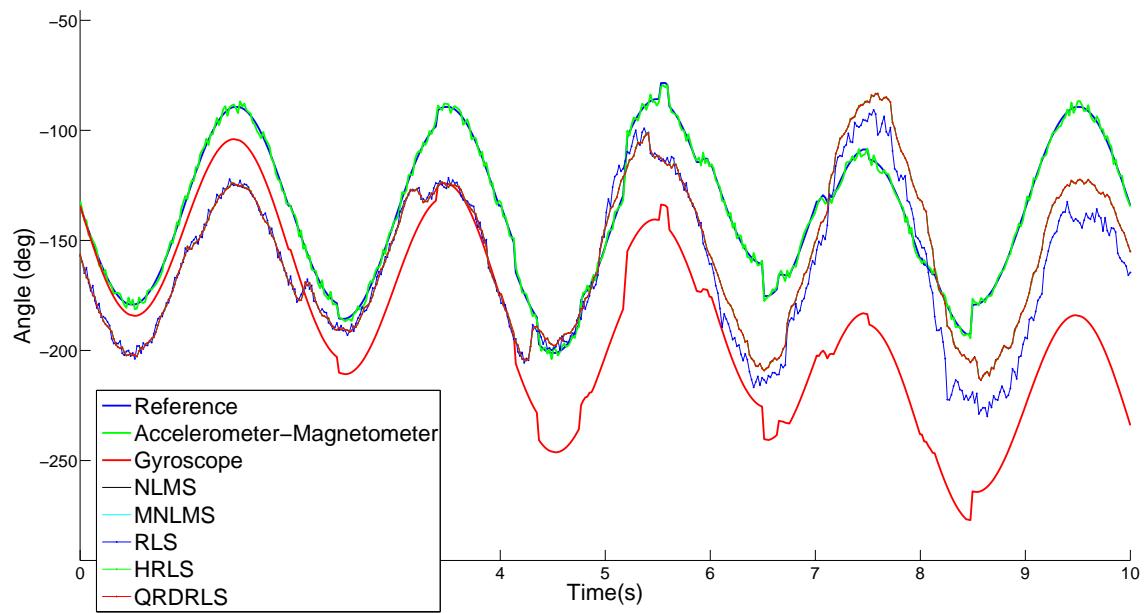


Figure 5.29: Attitude angle computed using sensor fusion algorithms applied on low intensity synthetic signals.

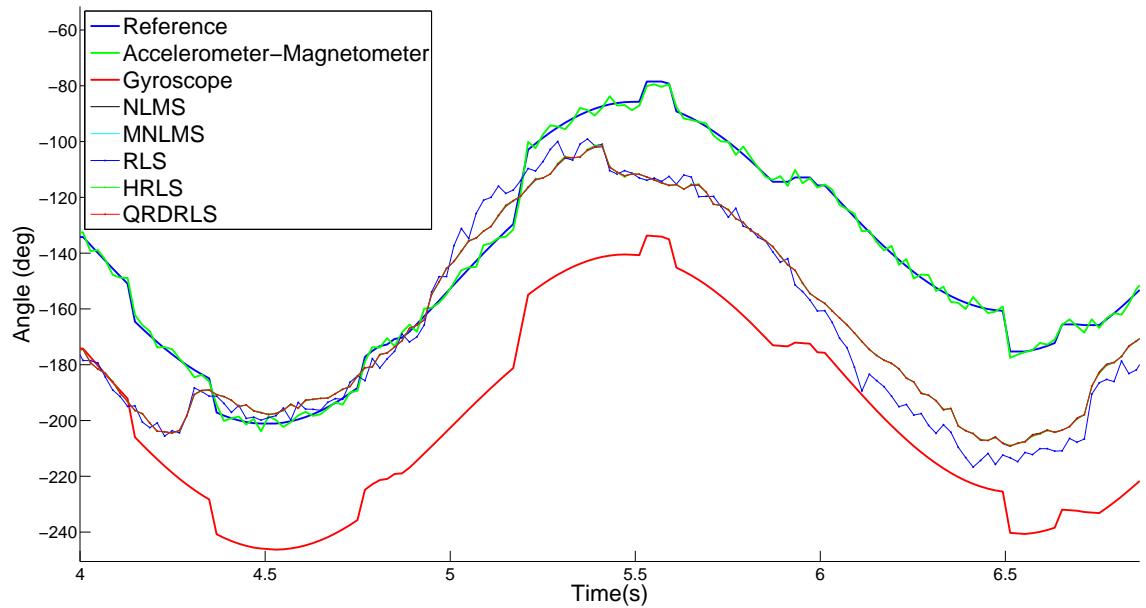


Figure 5.30: Attitude angle computed using sensor fusion algorithms applied on low intensity synthetic signals (zoomed in).

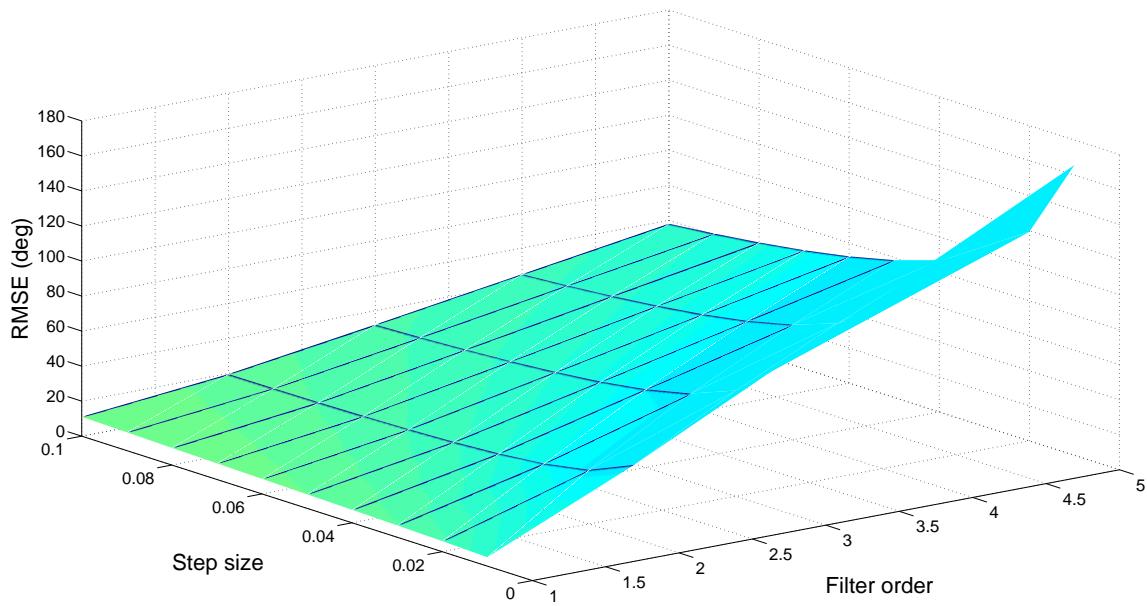


Figure 5.31: Optimization of input parameters. NLMS. Low and high intensity synthetic signals.

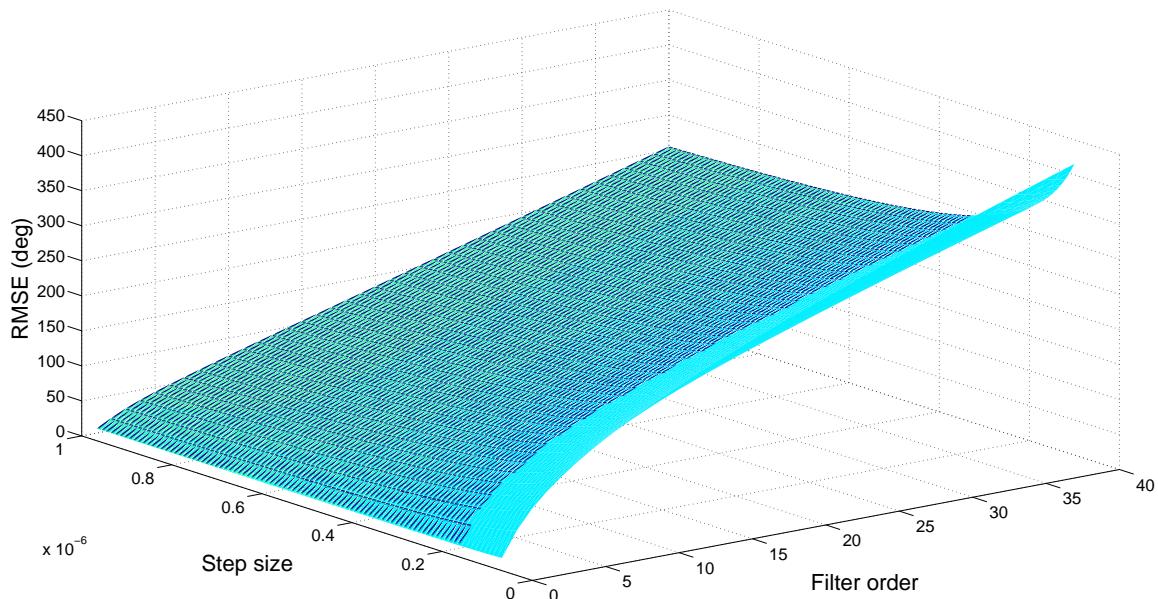


Figure 5.32: Optimization of input parameters. MNLMS. Low and high intensity synthetic signals.

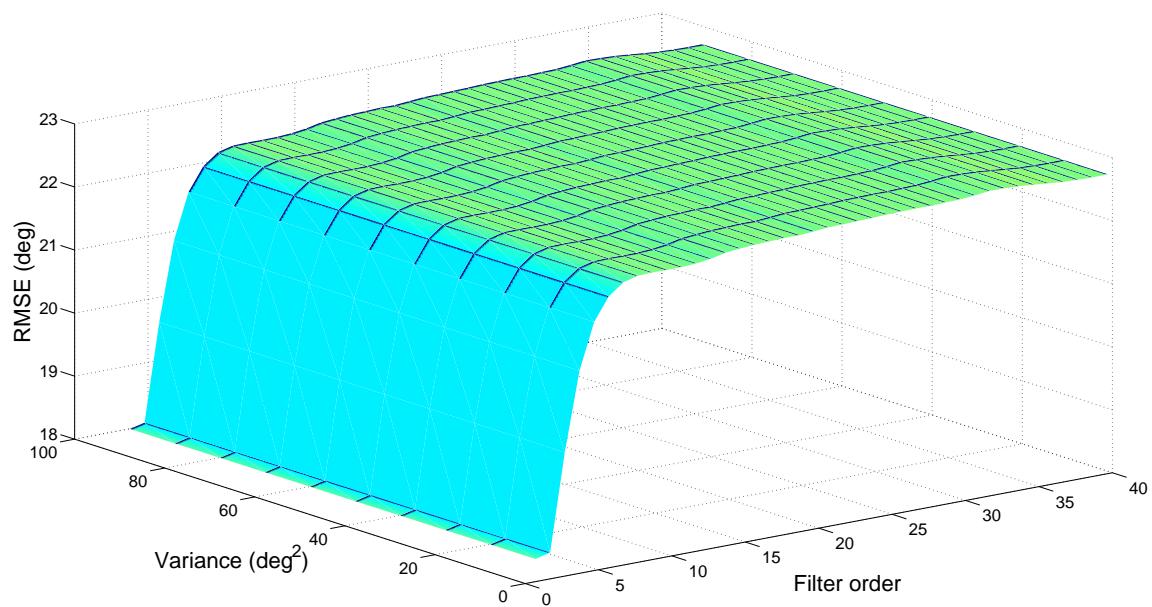


Figure 5.33: Optimization of input parameters. RLS. Low and high intensity synthetic signals.

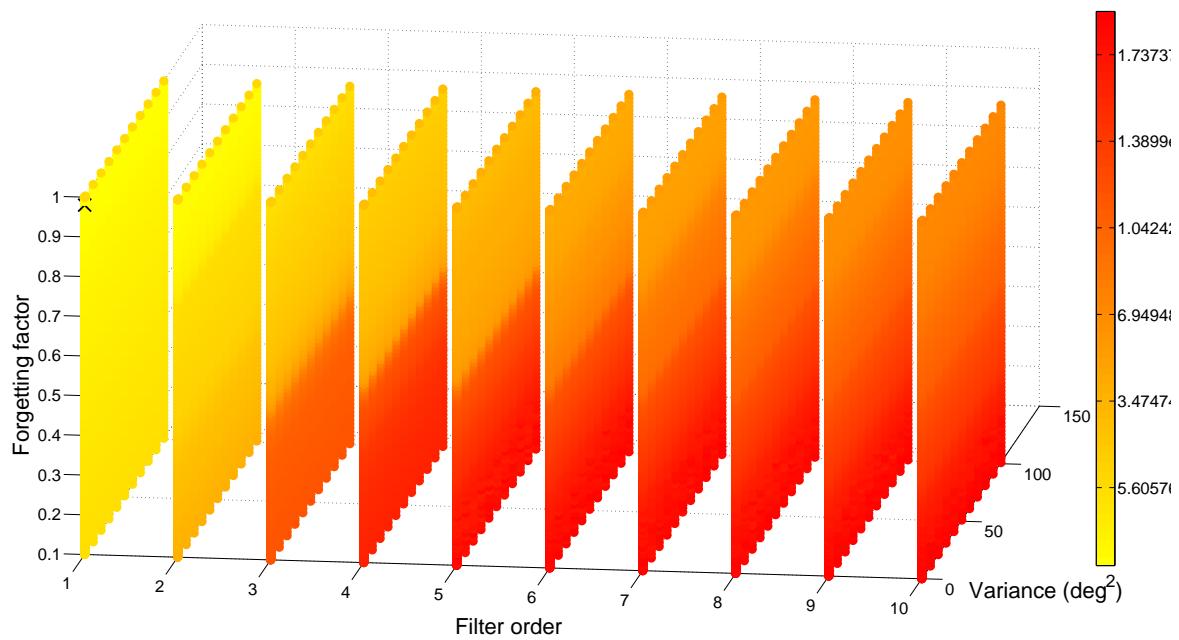


Figure 5.34: Optimization of input parameters. HRLS. Low and high intensity synthetic signals. The cross indicates the optimal point of operation.

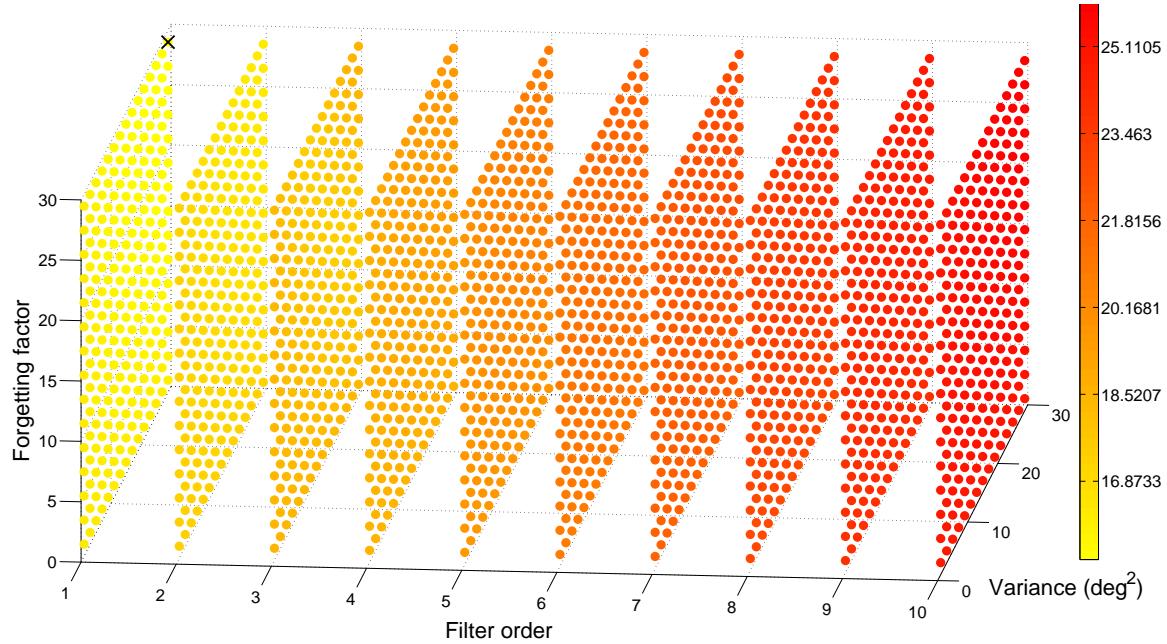


Figure 5.35: Optimization of input parameters. QRDRRLS. Low and high intensity synthetic signals. The cross indicates the optimal point of operation.

Table 5.4: Results of the Monte Carlo simulation ($N=50$). Synthesized signals with both low and high intensity. Average RMSE and associated parameters. (R) indicates that the integration resetting approach was used.

Algorithm	Av. RMSE	First parameter	Second parameter	Third parameter
NLMS	5.2347 ± 0.3834	$p=1.0000 \pm 0.0000$	$\mu=0.0120 \pm 0.0040$	—
NLMS(R)	21.7478 ± 2.7113	$p=1.0000 \pm 0.0000$	$\mu=1.2500 \pm 0.9090$	—
MNLMS	6.4350 ± 0.8272	$p=1.0000 \pm 0.0000$	$\mu=2.8940e-7 \pm 0.6488e-7$	—
MNLMS(R)	5.8135 ± 1.0072	$p=1.0000 \pm 0.0000$	$\mu=2.5200e-7 \pm 1.2673e-7$	—
RLS	19.7383 ± 1.7054	$p=1.0000 \pm 0.0000$	$\delta=46.0000 \pm 45.4569$	—
RLS(R)	8.0267 ± 2.3209	$p=1.0000 \pm 0.0000$	$\delta=42.4000 \pm 45.5690$	—
HRLS	6.7670 ± 0.8161	$p=1.0000 \pm 0.0000$	$\lambda=0.9768 \pm 0.0051$	8.0892 ± 27.6761
HRLS(R)	6.1789 ± 1.0526	$p=1.0000 \pm 0.0000$	$\lambda=0.9832 \pm 0.0099$	24.2476 ± 43.5690
QRDRRLS	6.3372 ± 0.6942	$p=1.0000 \pm 0.0000$	$\lambda=1.7320e3 \pm 8.9680e2$	$2.568e3 \pm 1.0815e3$
QRDRRLS(R)	21.8692 ± 2.6962	$p=1.0000 \pm 0.0000$	$\lambda=0.0010 \pm 6.5713e-19$	$0.0010 \pm 6.5713e-19$

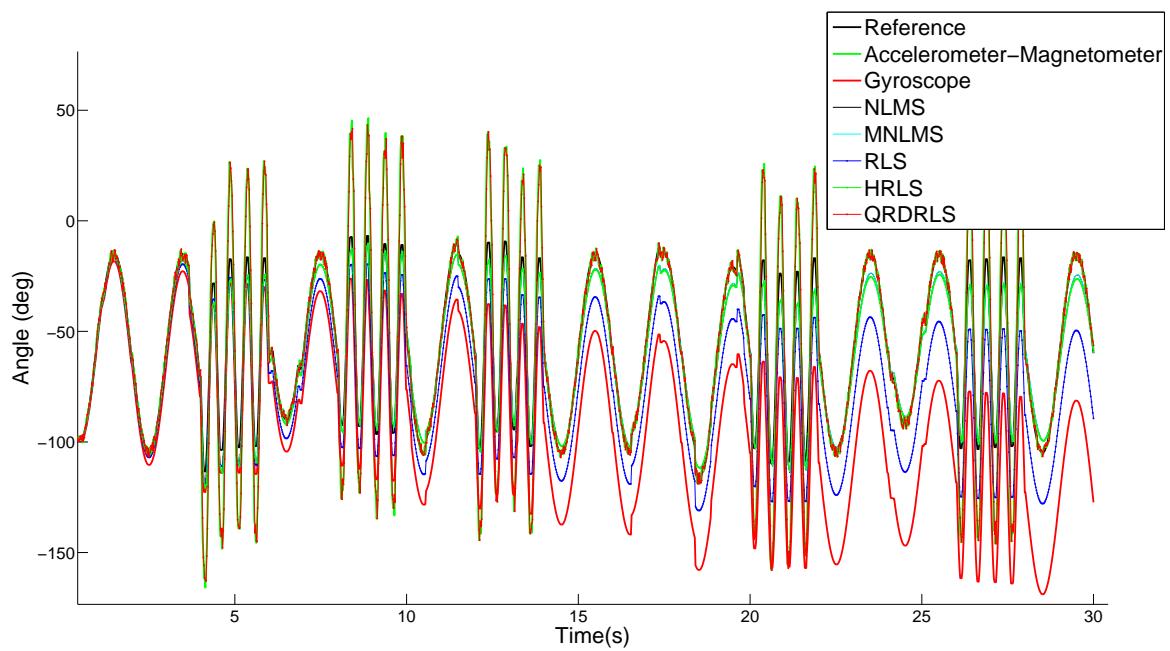


Figure 5.36: Orientation angle computed using sensor fusion algorithms applied on synthetic signals presenting both low and high intensity.

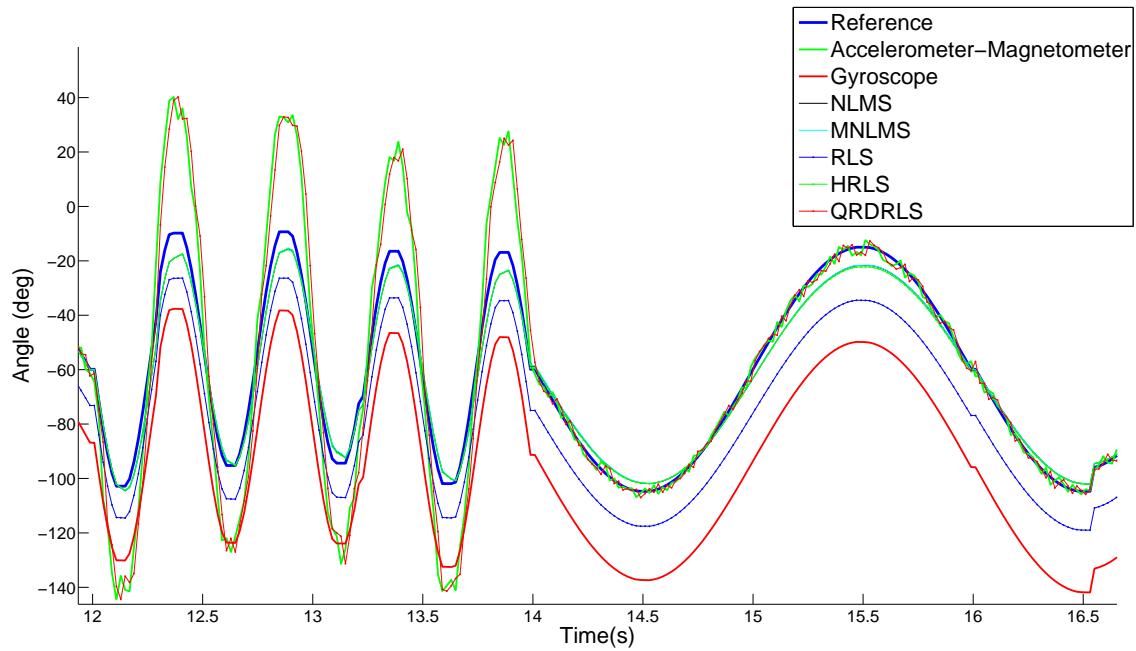


Figure 5.37: Orientation angle computed using sensor fusion algorithms applied on low synthetic signals presenting both low and high intensity (zoomed in).

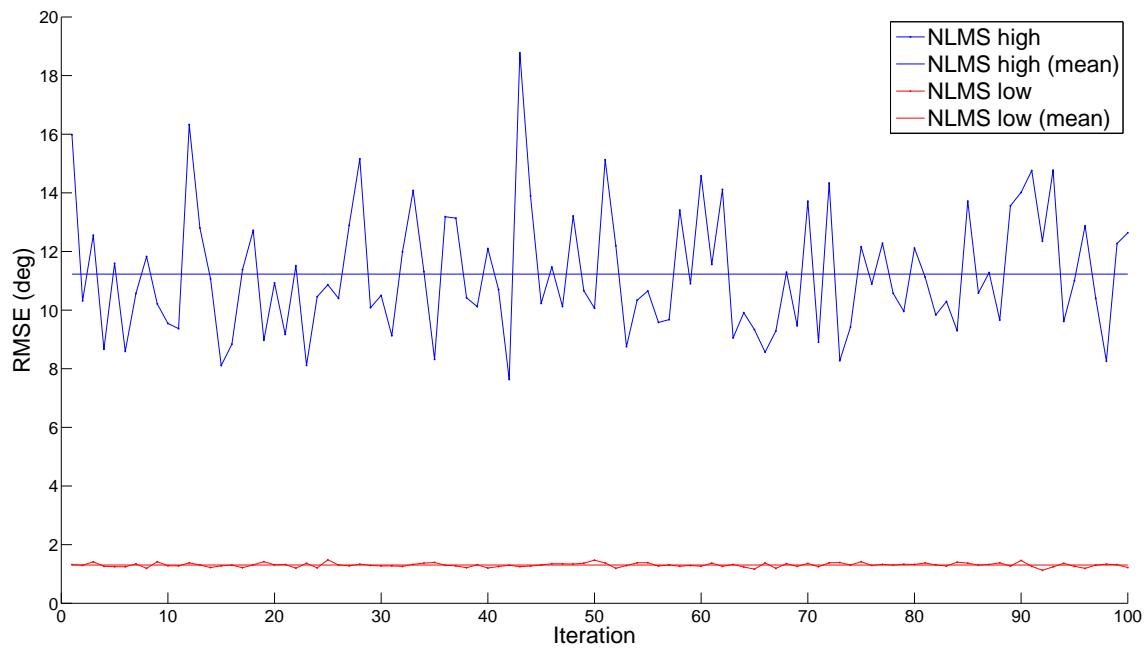


Figure 5.38: Validation of optimal synthetic parameters on 100 new synthetic signals. NLMS (high intensity signals and low intensity signals).

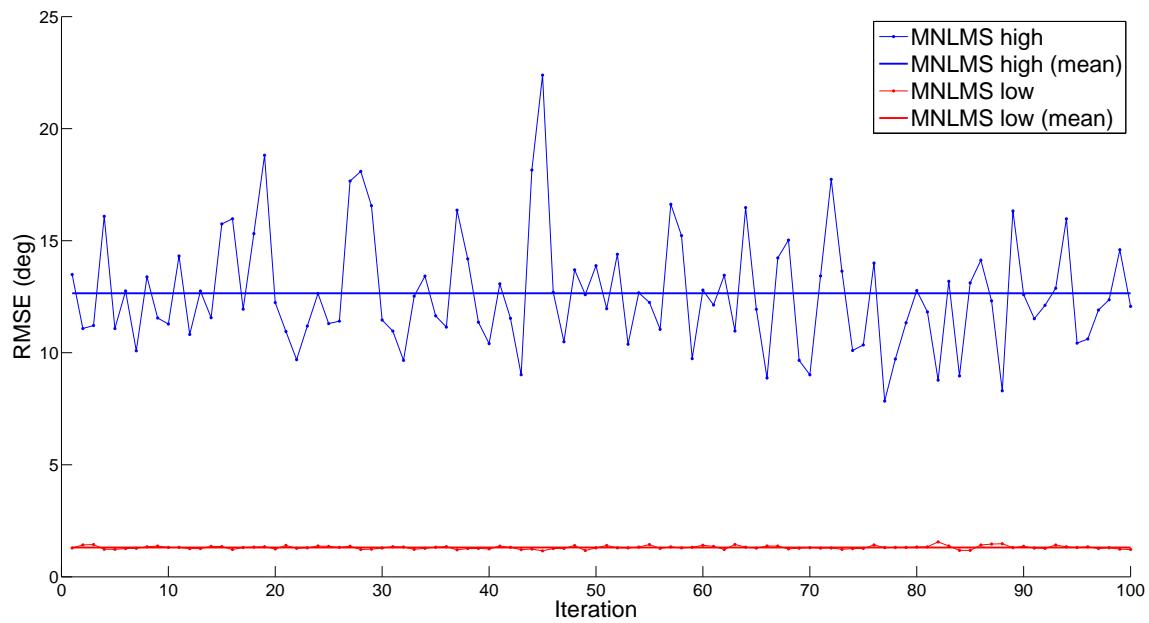


Figure 5.39: Validation of optimal synthetic parameters on 100 new synthetic signals. MNLMS (high intensity signals and low intensity signals).

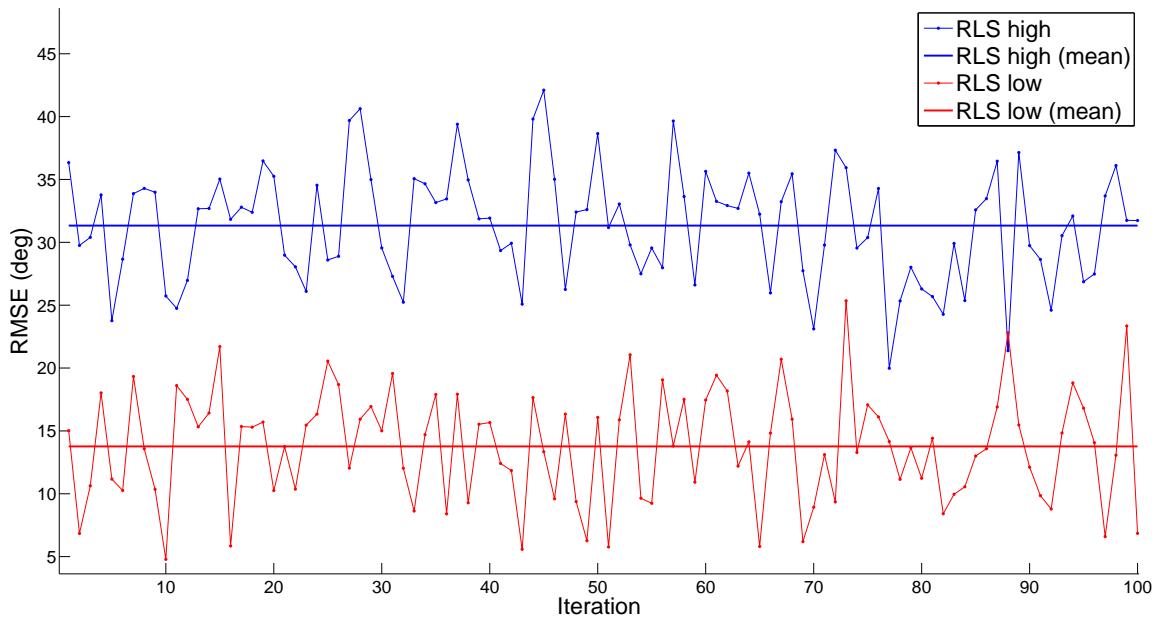


Figure 5.40: Validation of optimal synthetic parameters on 100 new synthetic signals. RLS (high intensity signals and low intensity signals).

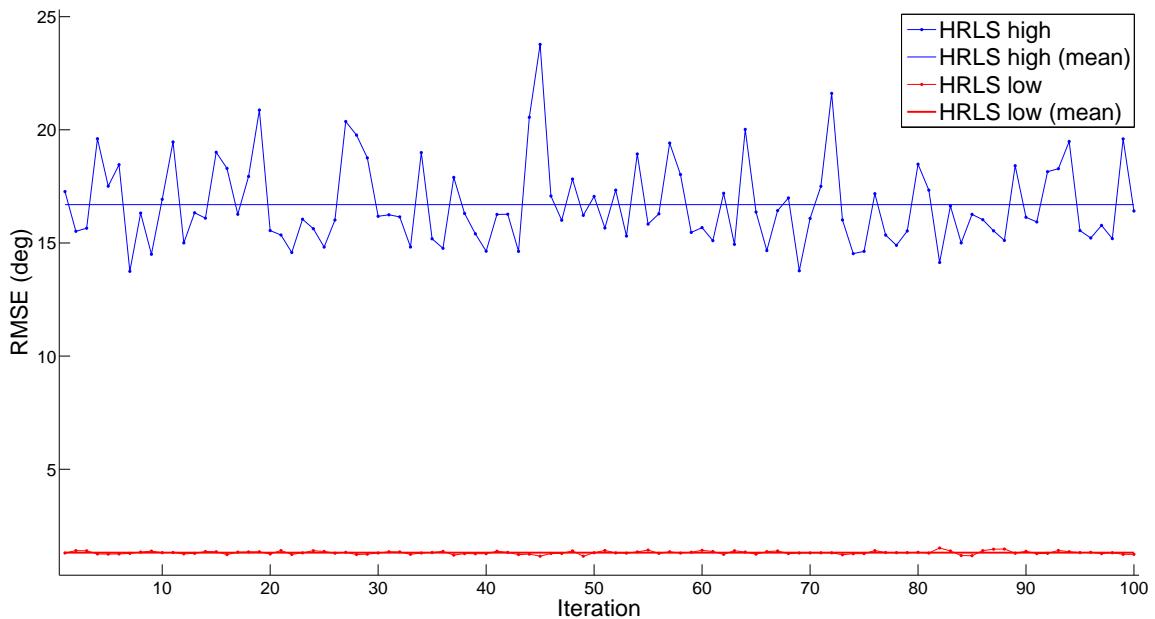


Figure 5.41: Validation of optimal synthetic parameters on 100 new synthetic signals. HRLS (high intensity signals and low intensity signals).

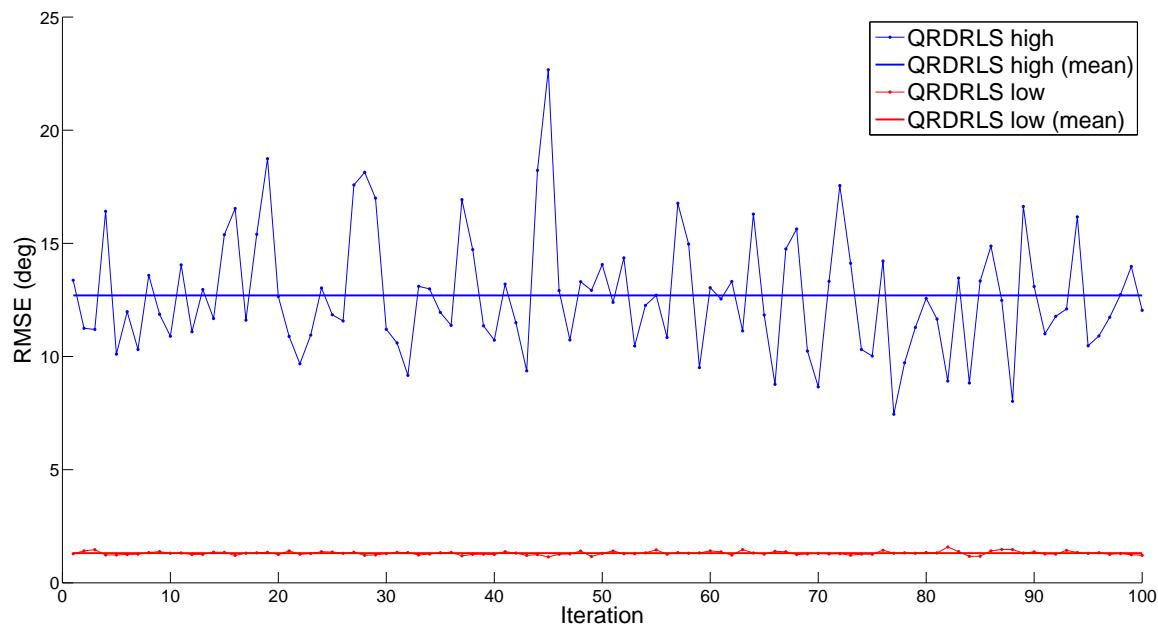


Figure 5.42: Validation of optimal synthetic parameters on 100 new synthetic signals. QRDRLS (high intensity signals and low intensity signals).

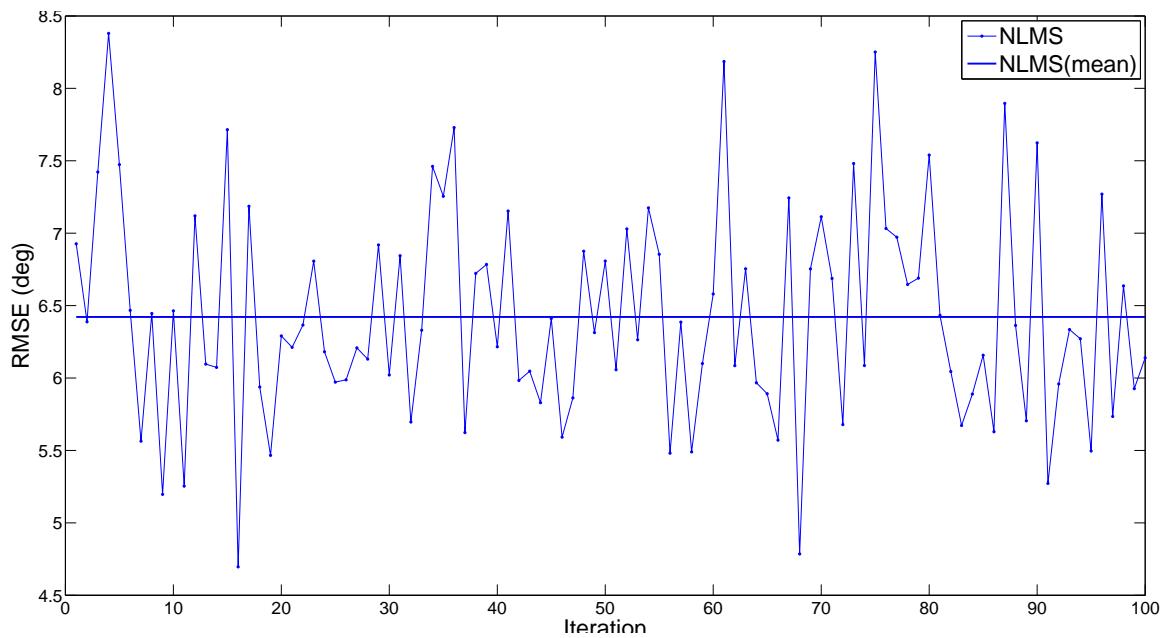


Figure 5.43: Validation of optimal synthetic parameters on 100 new synthetic signals. NLMS (signals containing both high and low intensity).

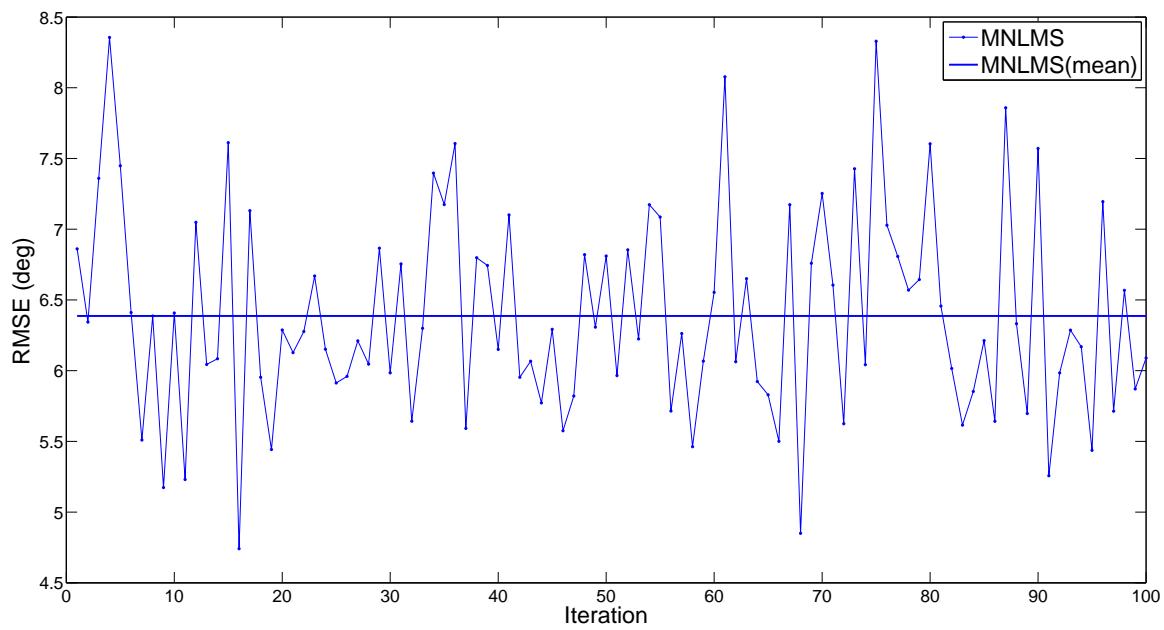


Figure 5.44: Validation of optimal synthetic parameters on 100 new synthetic signals. MNLMS (signals containing both high and low intensity).

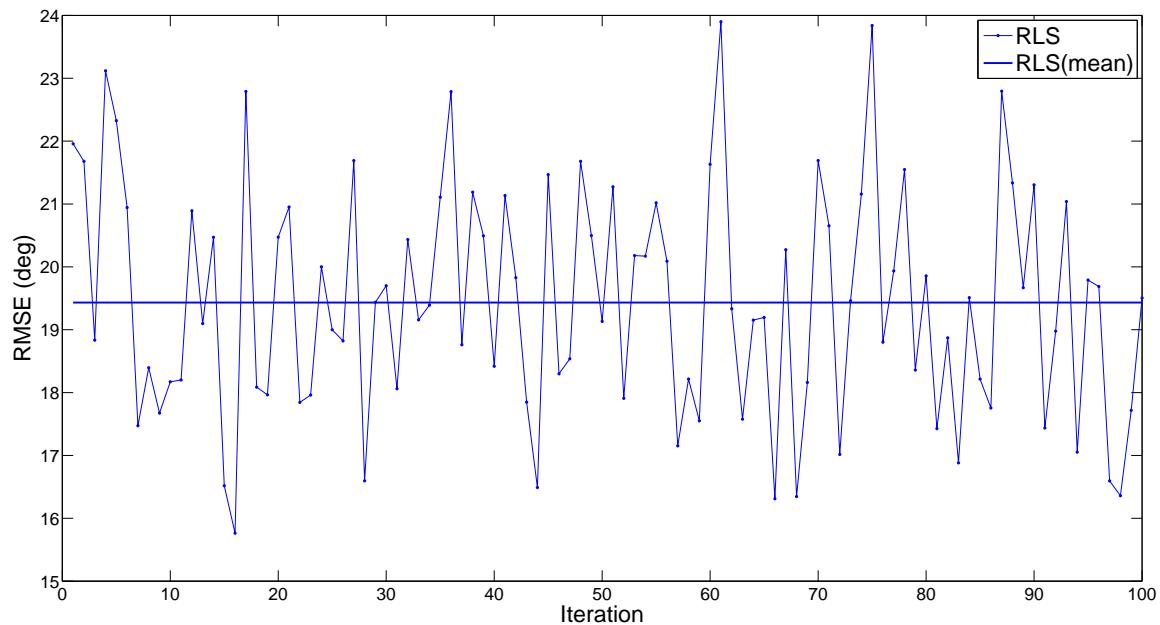


Figure 5.45: Validation of optimal synthetic parameters on 100 new synthetic signals. RLS (signals containing both high and low intensity).

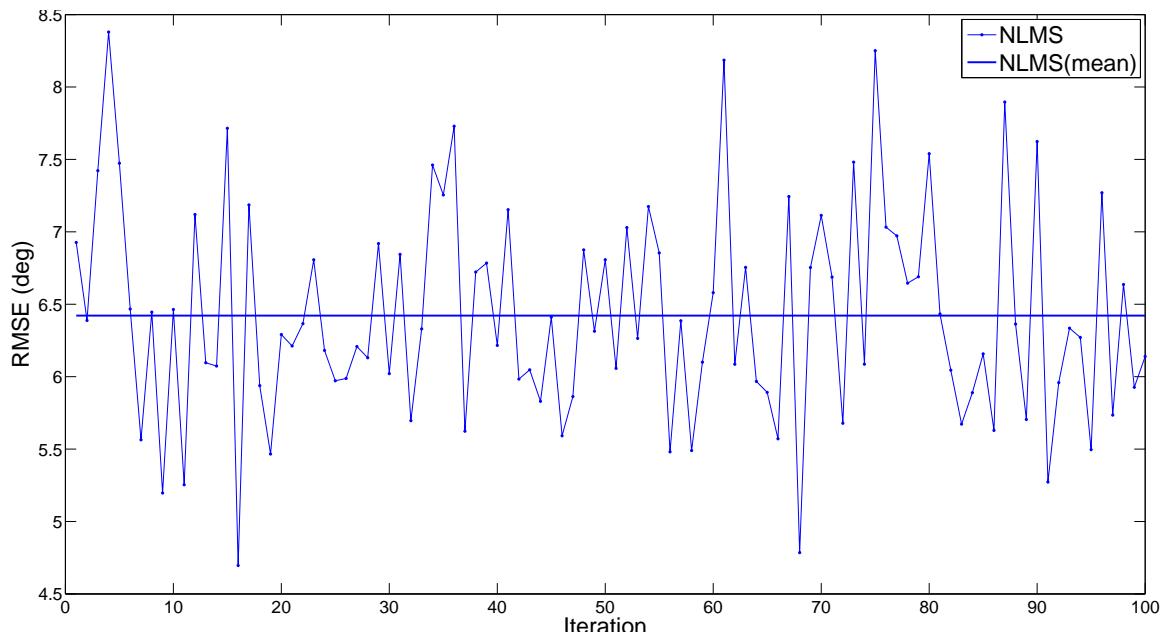


Figure 5.46: Validation of optimal synthetic parameters on 100 new synthetic signals. HRLS (signals containing both high and low intensity).

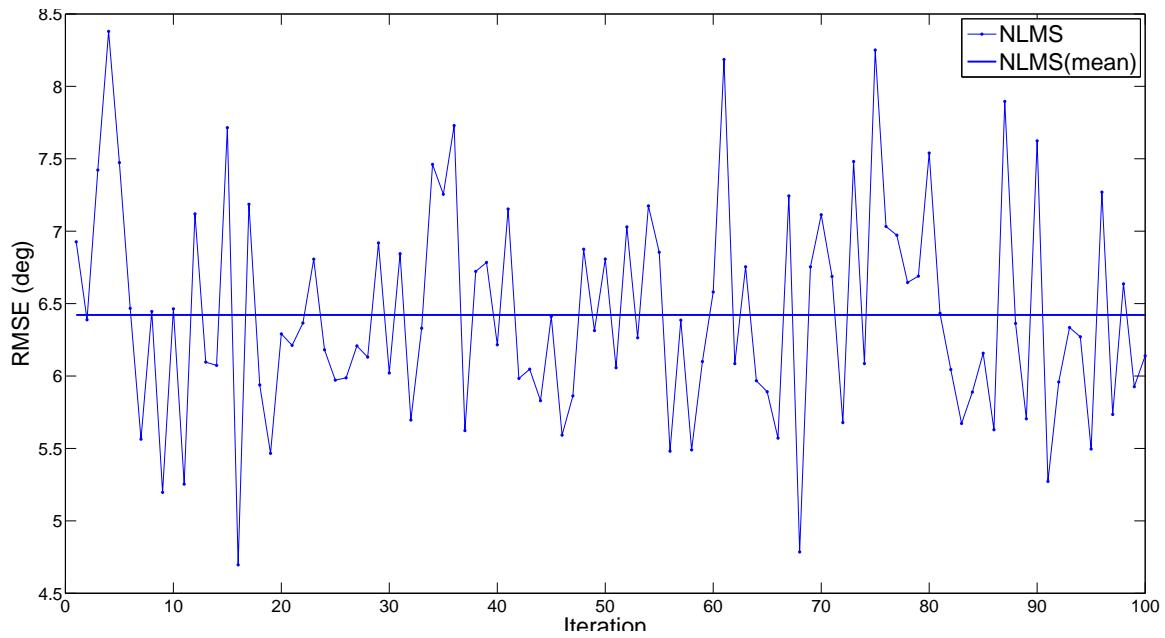


Figure 5.47: Validation of optimal synthetic parameters on 100 new synthetic signals. QRDRRLS (signals containing both high and low intensity).

5.3.3 Real signals

After the theoretical simulations we apply the algorithms over a set of real signals gathered using a Wagyromag MIMU. The MIMU is placed on a mechanical device which output is used as a reference angle, and then, a series of rotations around Y and Z axes is carried out. Rotation around X axis is not included here as it is equivalent to rotation around Y axis (as none of them uses the magnetometer to compute the estimate) and, therefore, the behavior of sensor fusion algorithms is equivalent.

The following sensor fusion algorithms are tested and compared over real signals:

- Madgwick's algorithm [80], based on the minimization of a cost function to find the optimal attitude accelerometer-magnetometer quaternion and its fusion with the integrated angular rate quaternion using a complementary filter.
- Kalman Filter and Gated Kalman Filter (both explained in section 5.2.4.1).
- Normalized Least Mean Squares (NLMS) and Momentum-Normalized Least Mean squares (MNLMS) filters.
- Recursive Least Squares (RLS), Household Recursive Least Squares (HRLS) and QR-Decompositon Recursive Least Squares (QRDRLS) filters.

5.3.3.1 Reference angle device

We built a mechanical device composed of two wooden planks joint with three hinges, one of which has an axial linear potentiometer inserted in its axis. The output of the potentiometer is connected to one of the spare analog inputs of Wagyromag's microcontroller. The analog value is sampled and digitalized and stored in the memory card together with the readings of the rest of the sensors.

The reference angle is calibrated using a protractor, taking measurements every 0.5° . Then a spline is used to build a lookup table that transforms the raw potentiometer digital readings into a reference angle in degrees. Figure 5.49 shows the reference angle calibration spline. Notice how the output of the potentiometer is highly linear. Figure 5.48 shows the mechanical reference angle device.

To gather the signals, the MIMU was placed first with its Y axis parallel to the rotation axis of the reference device. Then, we opened and closed the device alternating fast rotations with slow rotations. The data are stored in the memory card and later downloaded to the computer.

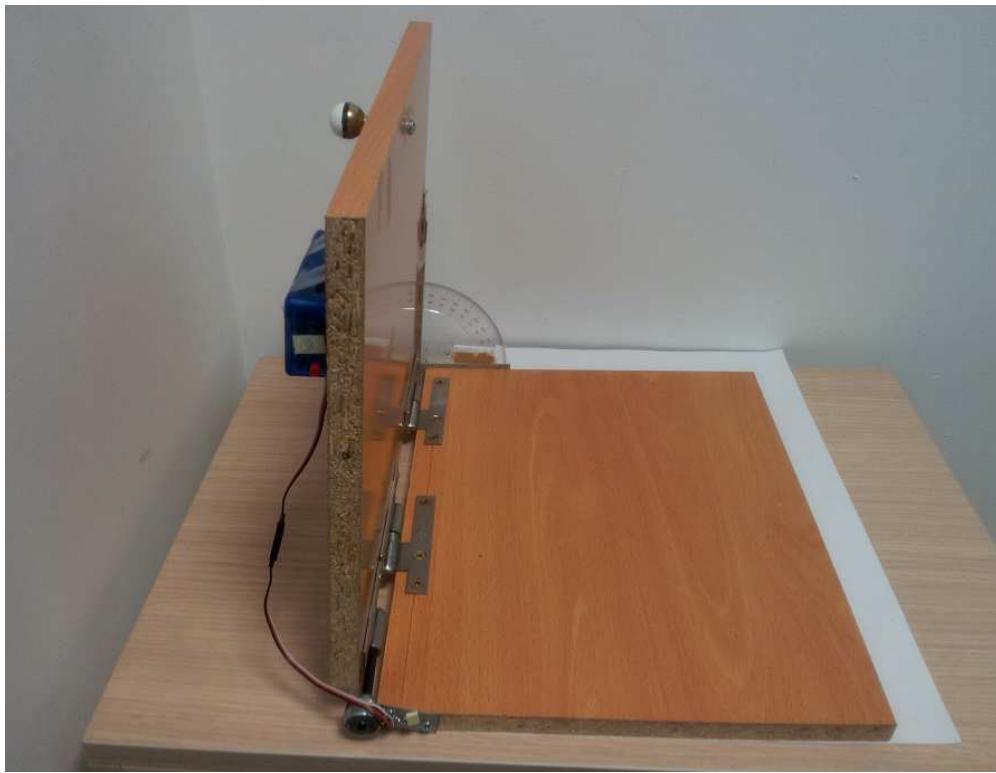


Figure 5.48: Angle reference device based on a linear potentiometer.

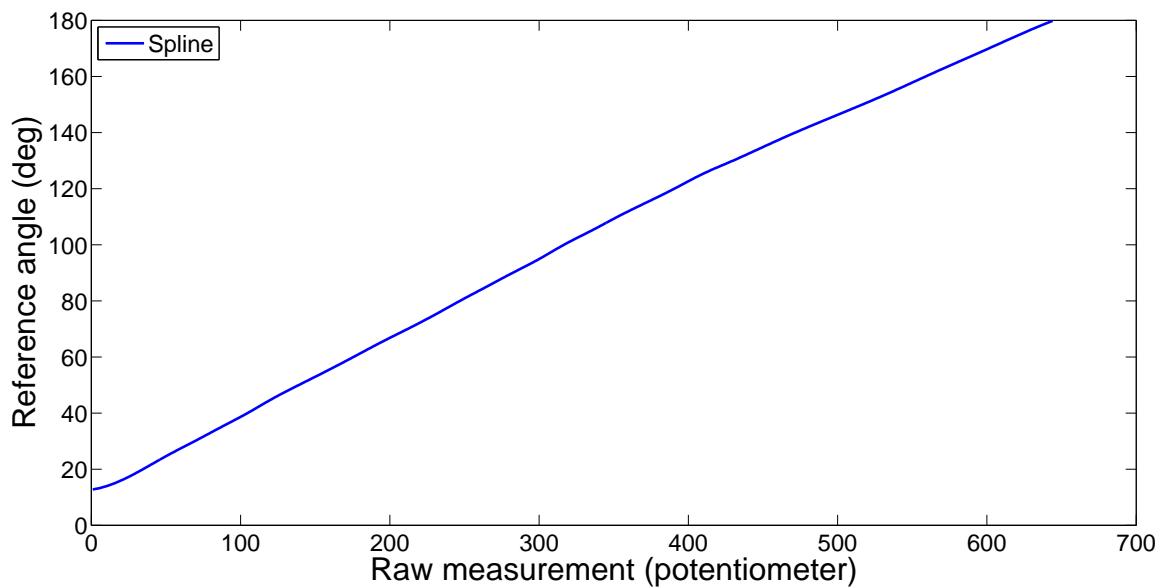


Figure 5.49: Spline mapping raw digital potentiometer values to reference angles in degrees.

5.3.3.2 Parameter optimizer

As we did with synthetic signals, an optimization procedure is also called to optimize the parameters of each one of the algorithms. In this case, as we are not running a Monte Carlo simulation, and therefore, we can manually select the initial value of the parameters, some of the algorithms have been optimized by minimizing the RMSE using the Gauss-Newton algorithm (more specifically, Kalman Filter, Gated Kalman Filter and Madgwick's algorithm).

The optimal parameters are then used to compute the estimates of the orientation angles.

5.3.3.3 Structure of the experiments

The structure of the experiments is very similar to those carried out using synthetic signals. We first show an example of angle estimates computed with the different non-fusion approaches explained during this chapter (decomposition of gravity-Earth's magnetic field and integration of angular rate). Next, we apply all the algorithms over a small set of high intensity real signals and calculate the average RMSE and optimal parameters. We repeat the calculations but this time employing the optimal synthetic parameters to further check their validity. The same procedure is carried out again using low intensity real signals and, subsequently, signals presenting both low and high intensity.

Finally, we apply the same algorithms over the last set of signals (low and high intensity) employing an intensity detector which resets the bias of the integrated angular rate signal and changes the parameters of the Kalman filter (so it becomes a Gated-Kalman filter) based on the intensity of the motion.

5.3.3.4 Results

Results are presented in the following order:

1. Non-fusion algorithms on real signals presenting both low and high intensity:
 - RMSE of an example of angle estimation (pitch and yaw). Tables 5.5 and 5.6
 - Estimated orientation angles (pitch and yaw). Figures 5.50-5.55.
2. Sensor fusion algorithms applied on high intensity signals:
 - Parameter optimization procedure. Figures 5.56-5.60.

- Average RMSE and average optimal parameters. Table 5.7.
- Average RMSE of the application of the algorithms on real signals using the optimal parameters obtained from theoretical simulations. Table 5.8.
- Estimated orientation angles. Figures 5.61 and 5.62.

3. Sensor fusion algorithms applied on low intensity signals:

- Parameter optimization procedure. Figures 5.63-5.67.
- Average RMSE and average optimal parameters. Table 5.9.
- Average RMSE of the application of the algorithms on real signals using the optimal parameters obtained from theoretical simulations. Table 5.10.
- Estimated orientation angles. Figures 5.68 and 5.69.

4. Sensor fusion algorithms applied on signals having both low and high intensity:

- Parameter optimization procedure using grid search. Figures 5.70-5.74.
- RMSE and optimal parameters for pitch and yaw angles (using both grid search and Gauss-Newton). Tables 5.11 and 5.12 .
- RMSE of the application of the algorithms on real signals using the optimal parameters from synthetic signals. Tables 5.13 and 5.14.
- Estimated orientation angles. Figures 5.75 and 5.82.

5. Sensor fusion algorithms applied on signals having both low and high intensity. Application of gating to reset gyroscope angle signal and change Kalman's parameters dynamically:

- RMSE and optimal parameters. Table 5.15.
- Detection of low and high intensity periods. Figure 5.83.
- Resetting integrated angular rate signal using the intensity marker. Figure 5.84.
- Estimation of yaw using Kalman Filter vs. Gated-Kalman Filter. Figures 5.86 and 5.85.

Table 5.5: Example of application of non-fusion attitude estimation algorithms on real signals presenting both mild and low intensity: Pitch RMSE.

Algorithm	RMSE
Accelerometer-Magnetometer (Euler)	7.4772
Accelerometer-Magnetometer (FQA)	7.4516
Gyroscope (Euler)	14.4412
Gyroscope (Quaternion)	14.1259

Table 5.6: Example of application of non-fusion attitude estimation algorithms on real signals presenting both high and low intensity: Yaw RMSE.

Algorithm	RMSE
Accelerometer-Magnetometer (Euler)	59.8250
Accelerometer-Magnetometer (FQA)	16.7800
Gyroscope (Euler)	56.938
Gyroscope (Quaternion)	56.96

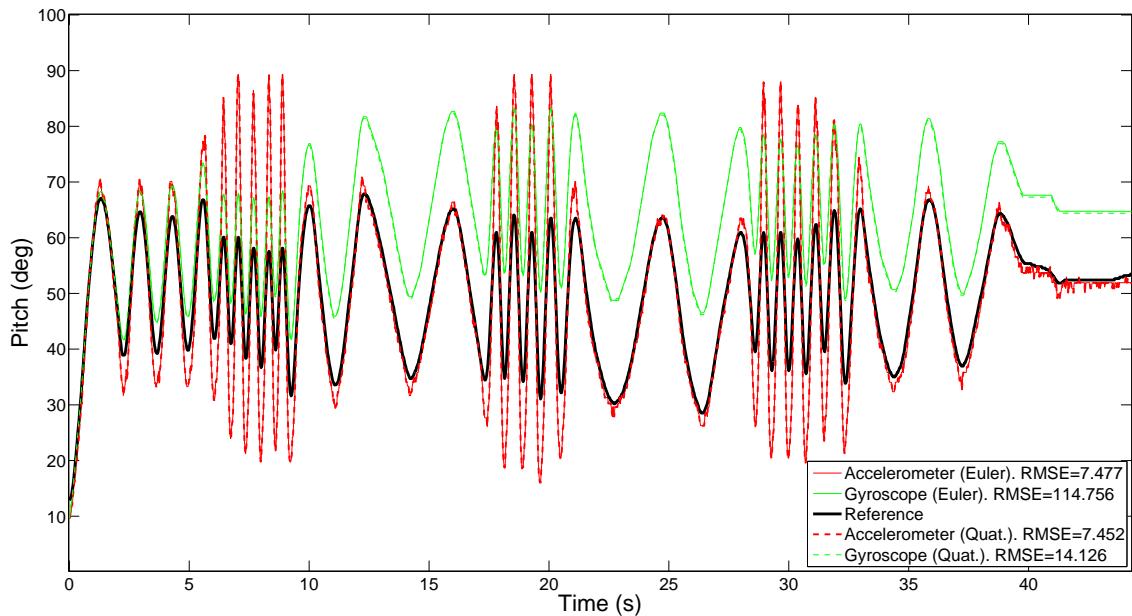


Figure 5.50: Estimation of Pitch. Non-fusion algorithms. Complete signal.

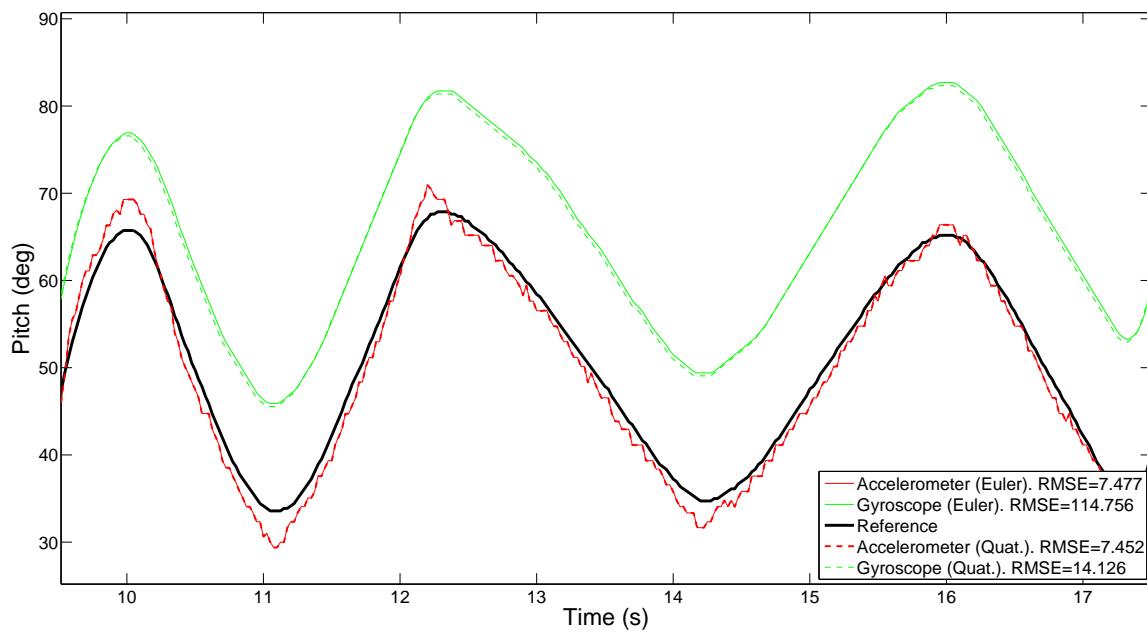


Figure 5.51: Estimation of Pitch. Non-fusion algorithms. Zoomed in on low intensity period.

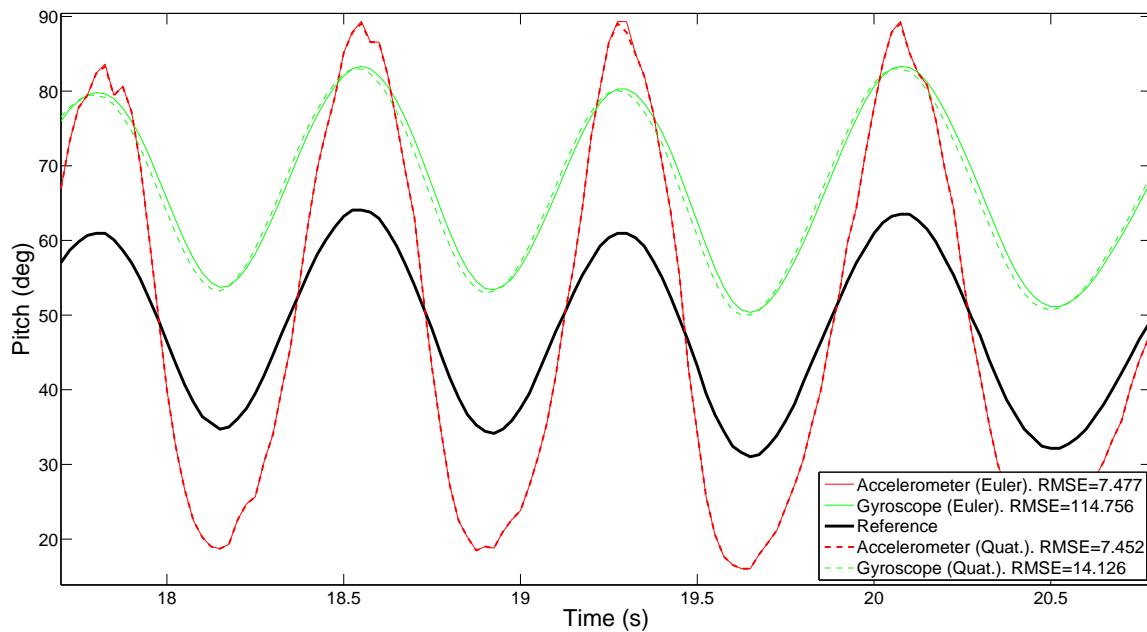


Figure 5.52: Estimation of Pitch. Non-fusion algorithms. Zoomed in on high intensity period.

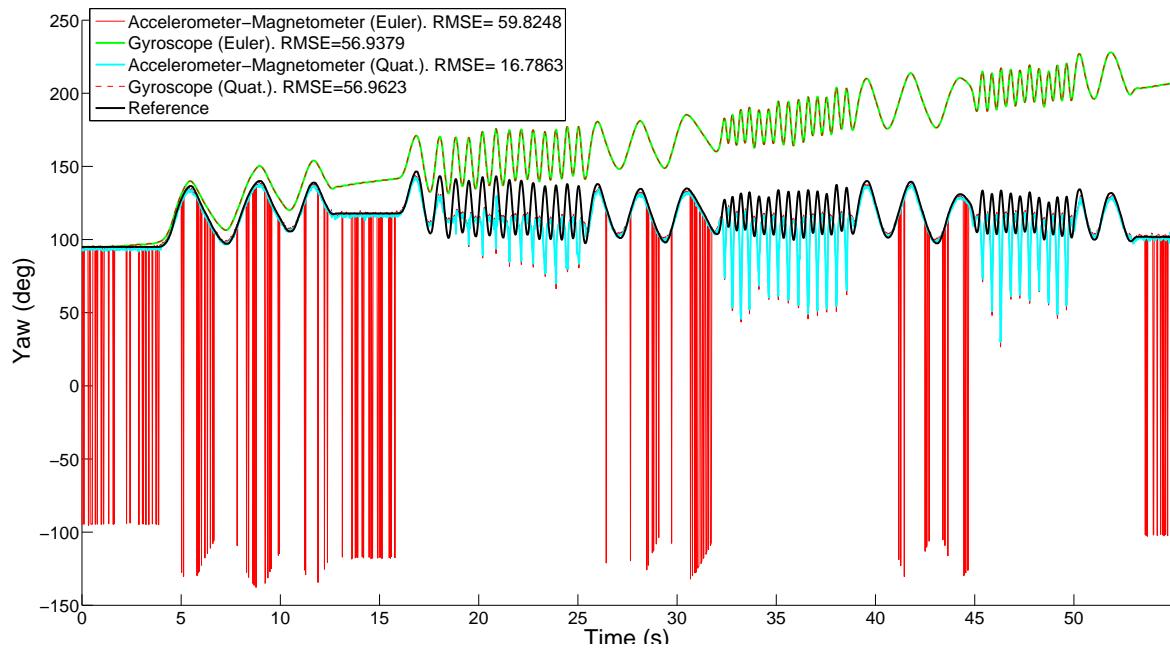


Figure 5.53: Estimation of Yaw. Non-fusion algorithms. Complete signal.

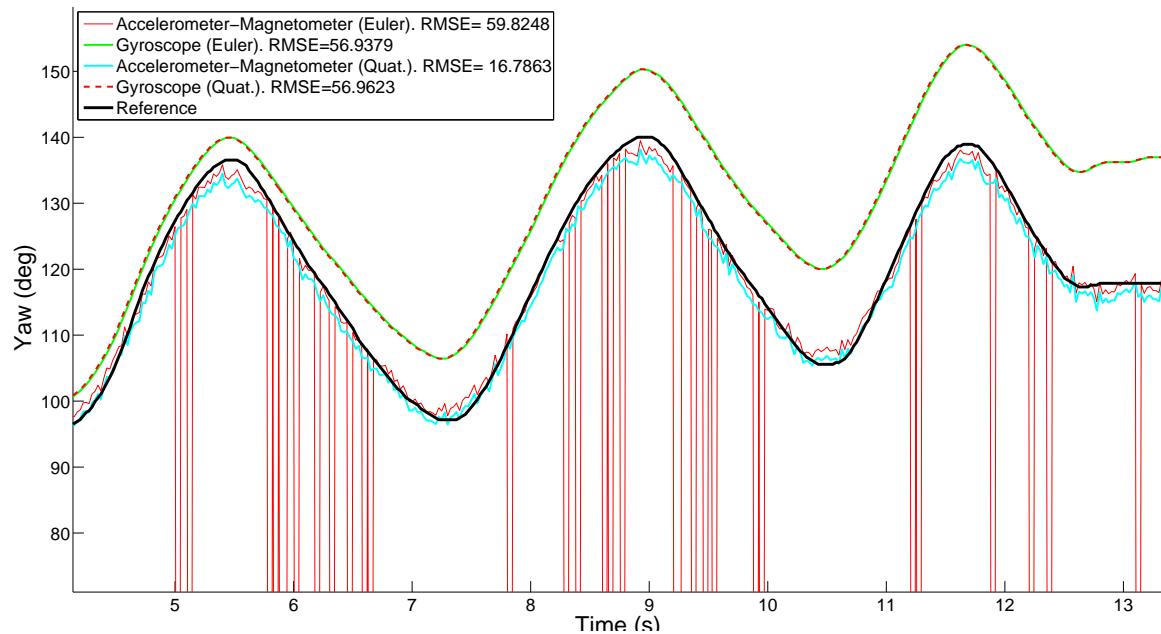


Figure 5.54: Estimation of Yaw. Non-fusion algorithms. Zoomed in on low intensity period.

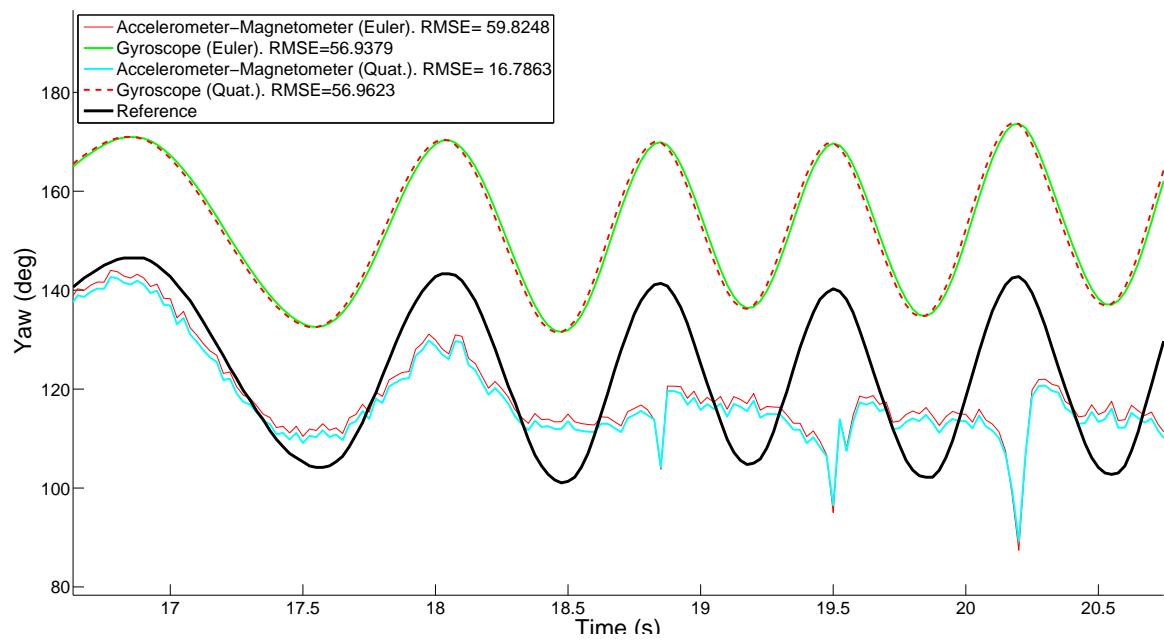


Figure 5.55: Estimation of Yaw. Non-fusion algorithms. Zoomed in on high intensity period.

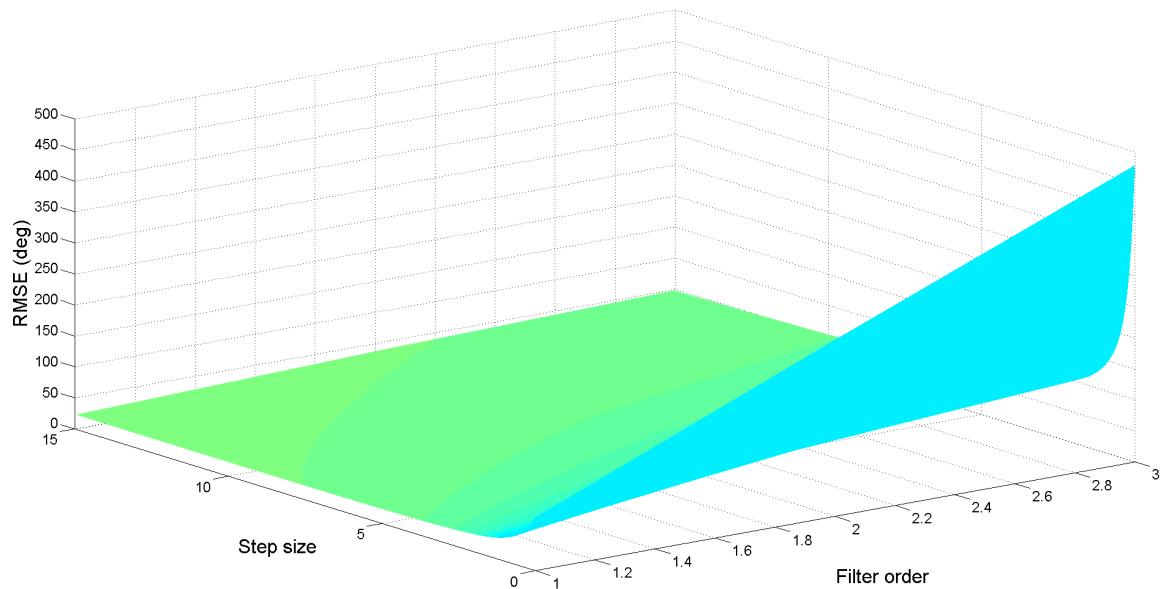


Figure 5.56: NLMS parameter optimization procedure. High intensity real signals.

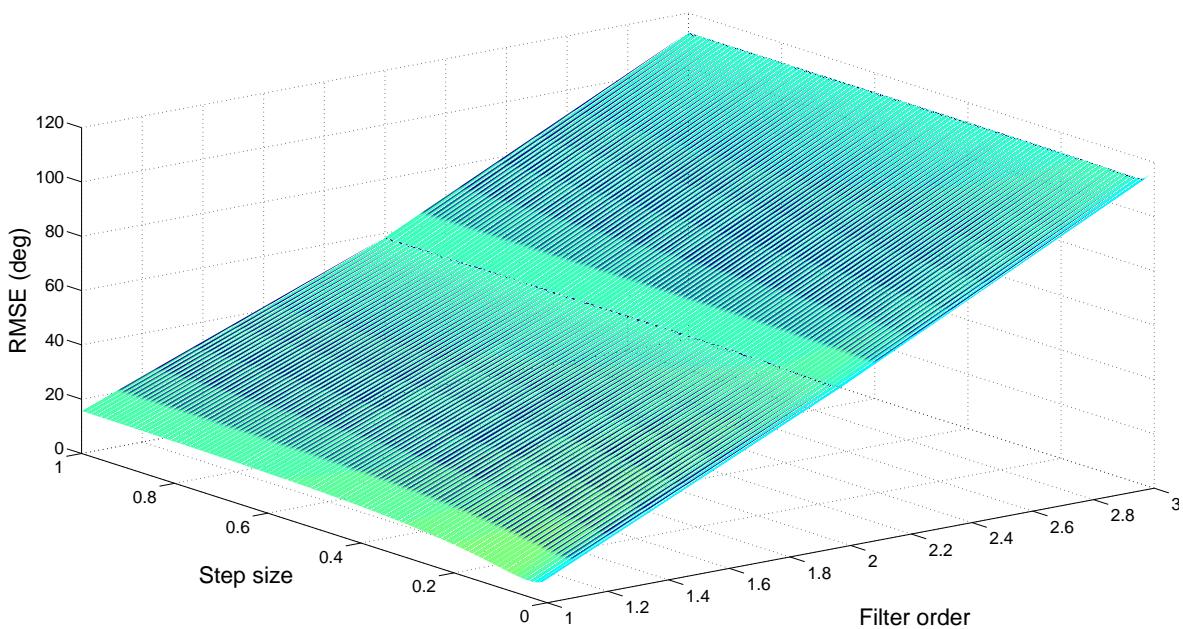


Figure 5.57: MNLMS parameter optimization procedure. High intensity real signals.

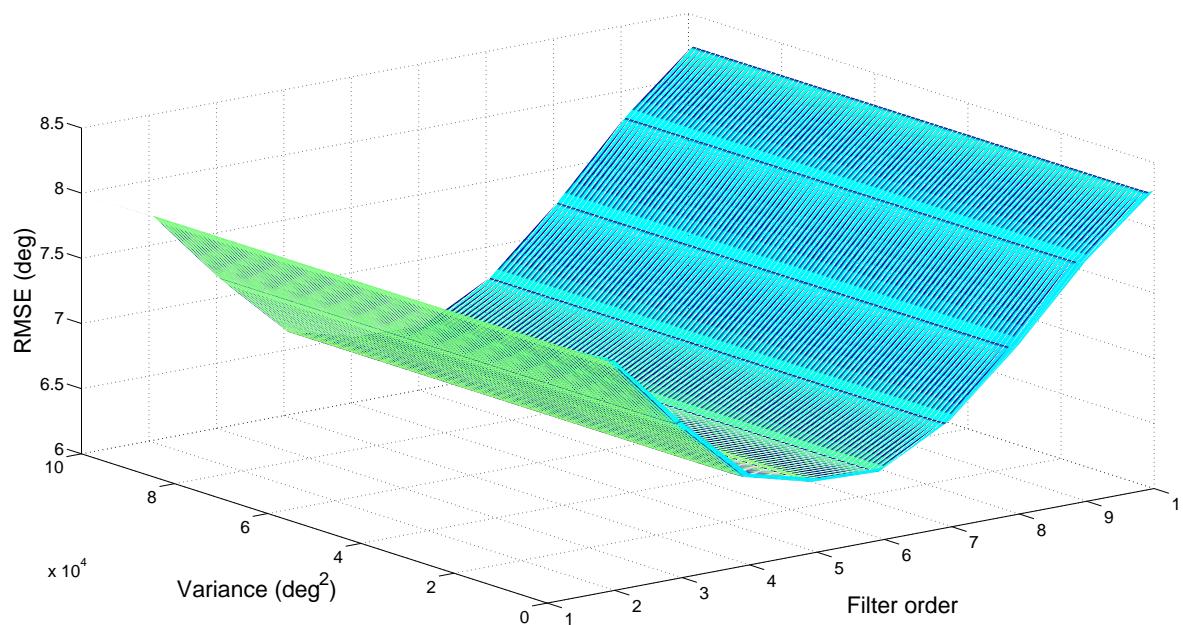


Figure 5.58: RLS parameter optimization procedure. High intensity real signals.

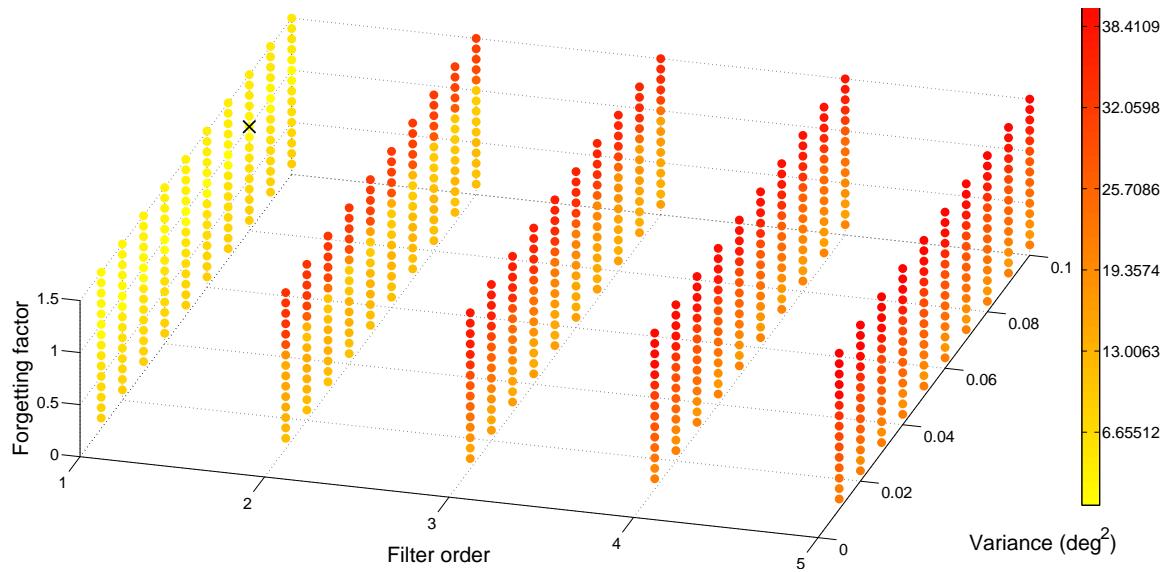


Figure 5.59: HRLS parameter optimization procedure. High intensity real signals. The cross indicates the optimal point of operation.

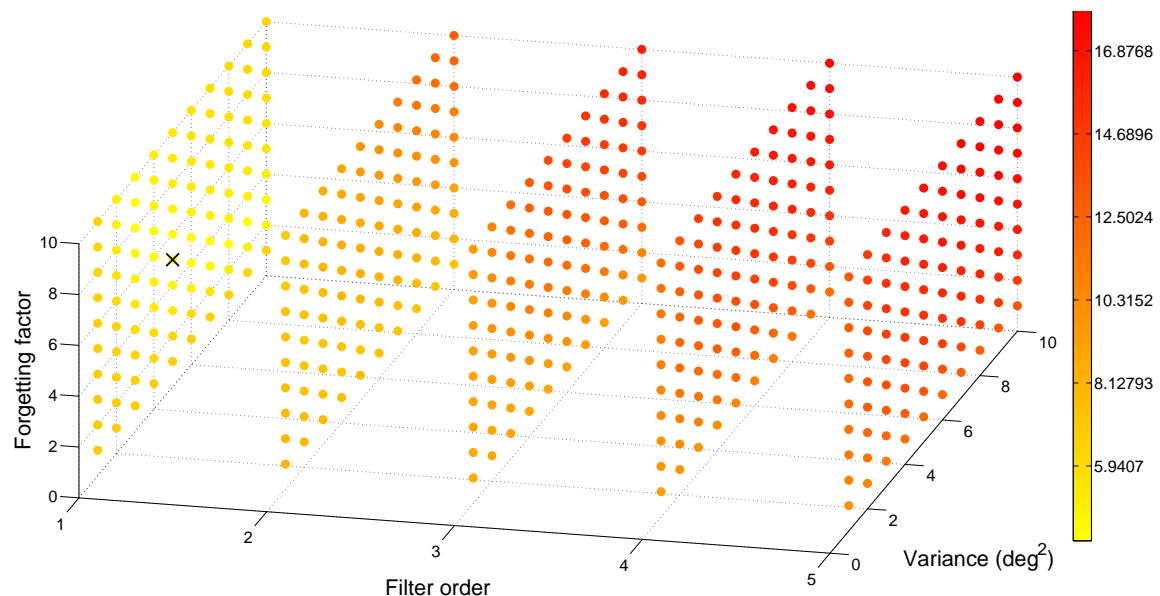


Figure 5.60: QRDRRLS parameter optimization procedure. High intensity real signals. The cross indicates the optimal point of operation.

Table 5.7: Application of sensor fusion attitude estimation algorithms on real signals presenting only high intensity. Average RMSE and average parameters.

Algorithm	Av. RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	7.7941 ± 1.5741	$L=1.0000 \pm 0.0000$	$\mu=4.6667 \pm 0.5774$	—
MNLMS	7.7014 ± 1.5718	$L=1.0000 \pm 0.0000$	$\mu=0.0467 \pm 0.0058$	—
RLS	6.2510 ± 0.2409	$L=2.6667 \pm 2.8868$	$\delta=3.3070e4 \pm 5.7270e4$	—
HRLS	4.3924 ± 1.9770	$L=1.0000 \pm 0.0000$	$\lambda=1.3667 \pm 0.3215$	$\delta=4.0267 \pm 5.2614$
QRDRRLS	7.7008 ± 1.5715	$L=1.0000 \pm 0.0000$	$\lambda=3.8667 \pm 0.0000$	$\delta=7.3333 \pm 4.9329$
Kalman	5.2274 ± 0.5976	$\alpha=4.633 \pm 2.0502$	—	—
Madgwick	15.1540 ± 1.9619	$\beta=7.7432 \pm 4.0820$	—	—

Table 5.8: Pitch angle computed applying sensor fusion methods using optimal parameters from synthetic signals on a set of real signals presenting only high intensity.

Algorithm	Average RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	11.4554 ± 0.2170	$L=1$	$\mu=9.8600 \pm 1.8737$	—
MNLMS	7.7014 ± 1.5718	$L=1$	$\mu=2.5600e-7 \pm 5.8134e-8$	—
RLS	6.8652 ± 0.9924	$L=1$	$\delta=8.7750e3 \pm 3.0952e3$	—
HRLS	10.3954 ± 2.9118	$L=1$	$\lambda=0.9000 \pm 8.972e-16$	$\delta=0.001 \pm 6.5713e-19$
QRDRRLS	14.8504 ± 5.3977	$L=1$	$\lambda=3.7400e3 \pm 1.2257e3$	$\delta=1.2800e3 \pm 8.3339e2$

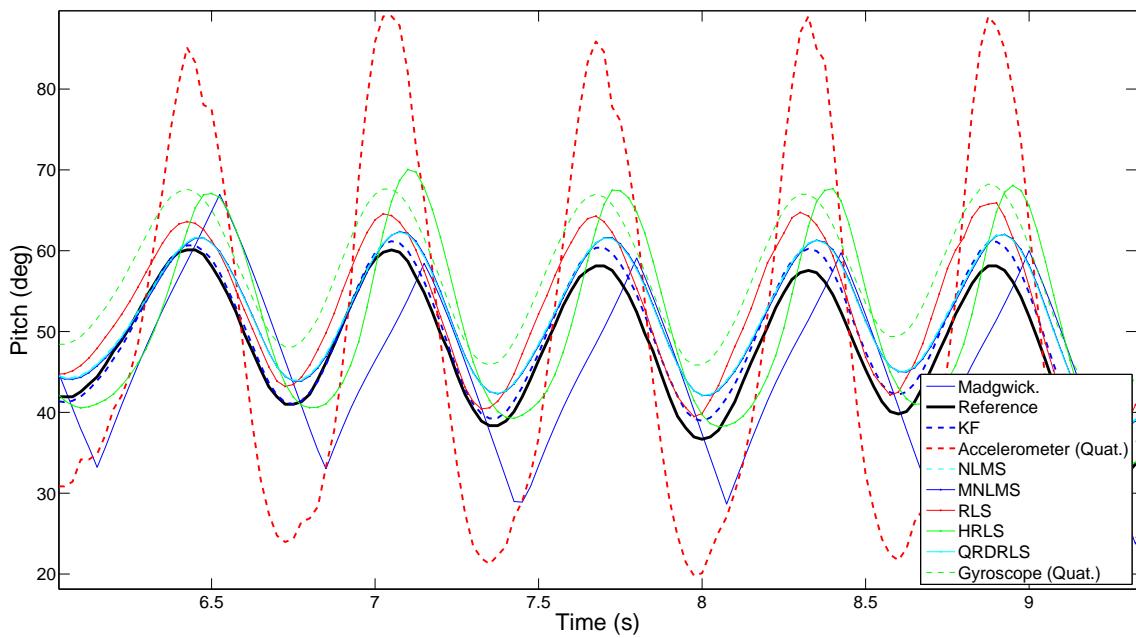


Figure 5.61: Pitch angle computed using sensor fusion algorithms applied on high intensity real signals.

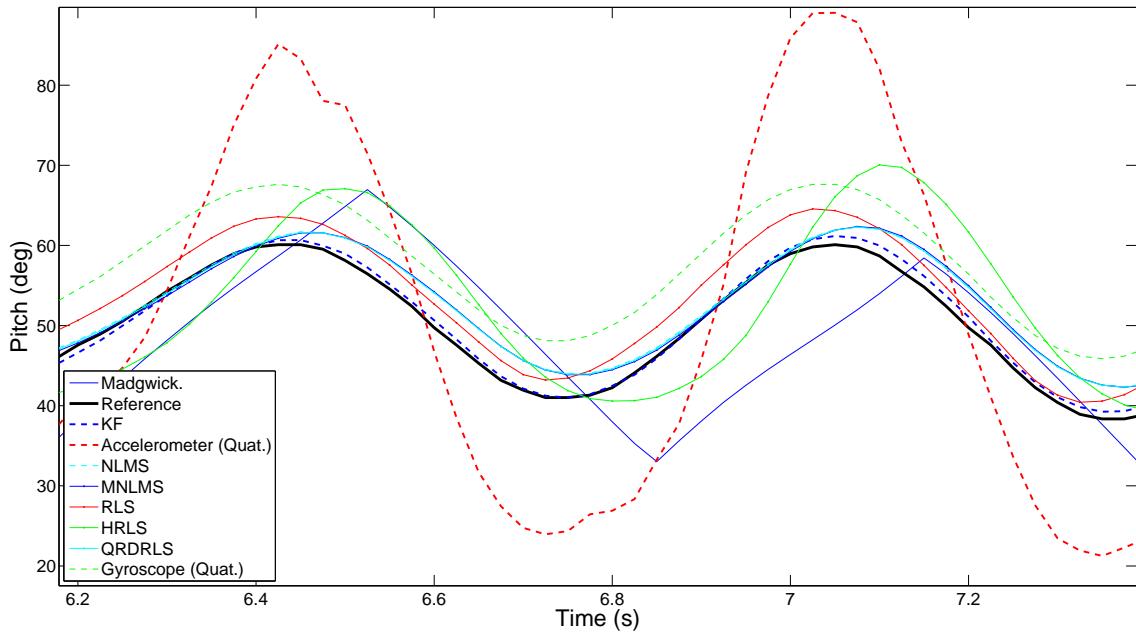


Figure 5.62: Pitch angle computed using sensor fusion algorithms applied on high intensity real signals (zoomed in).

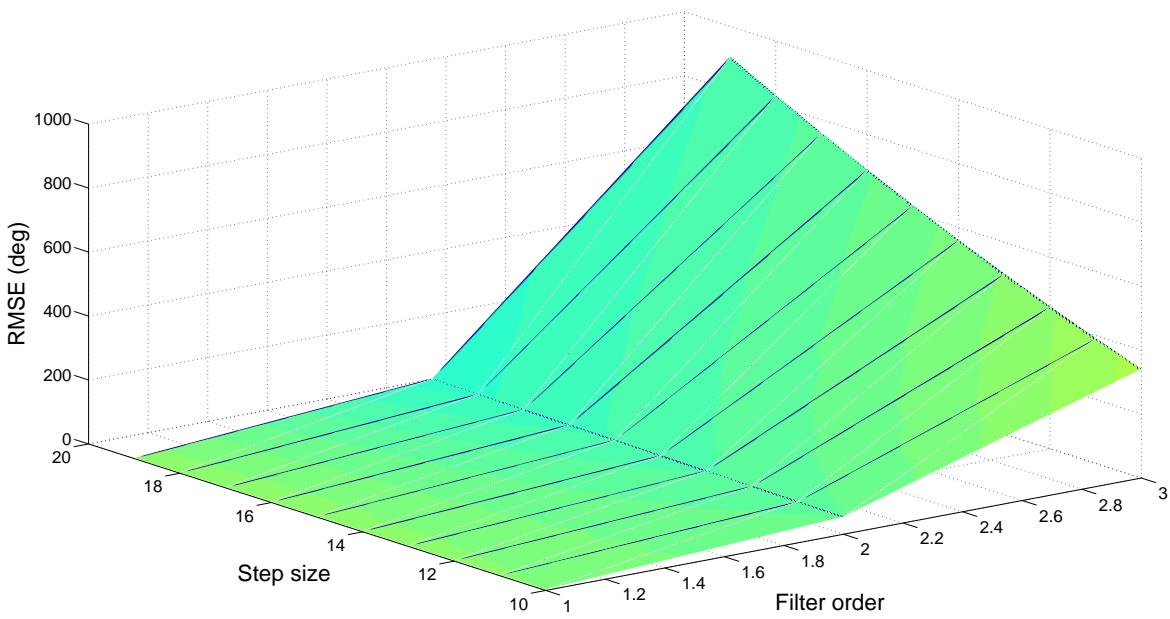


Figure 5.63: NLMS parameter optimization procedure. Low intensity real signals.

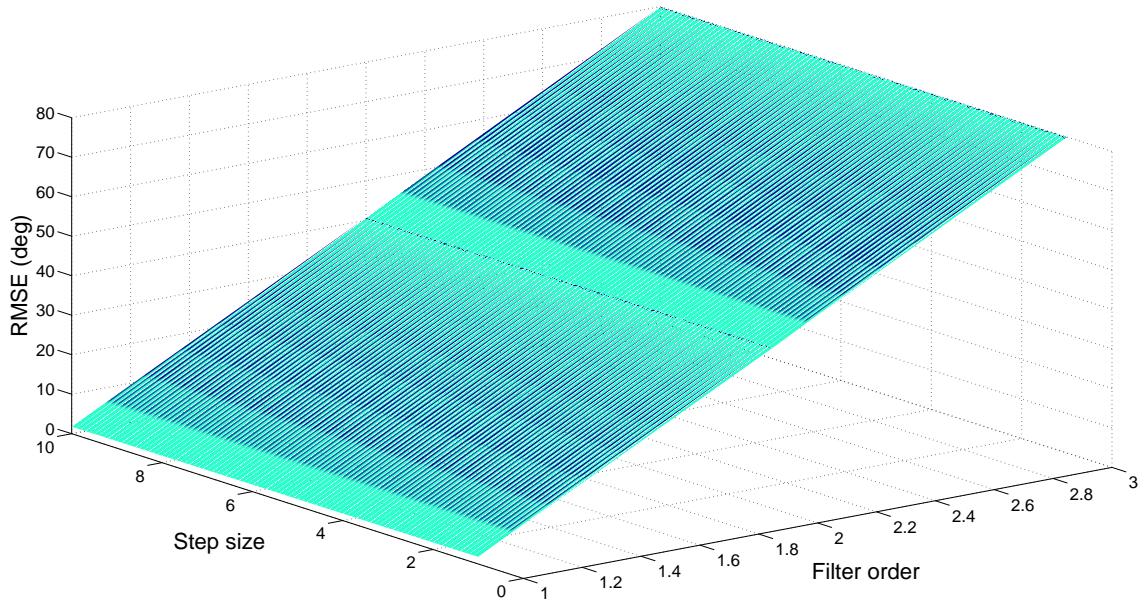


Figure 5.64: MNLMS parameter optimization procedure. Low intensity real signals.

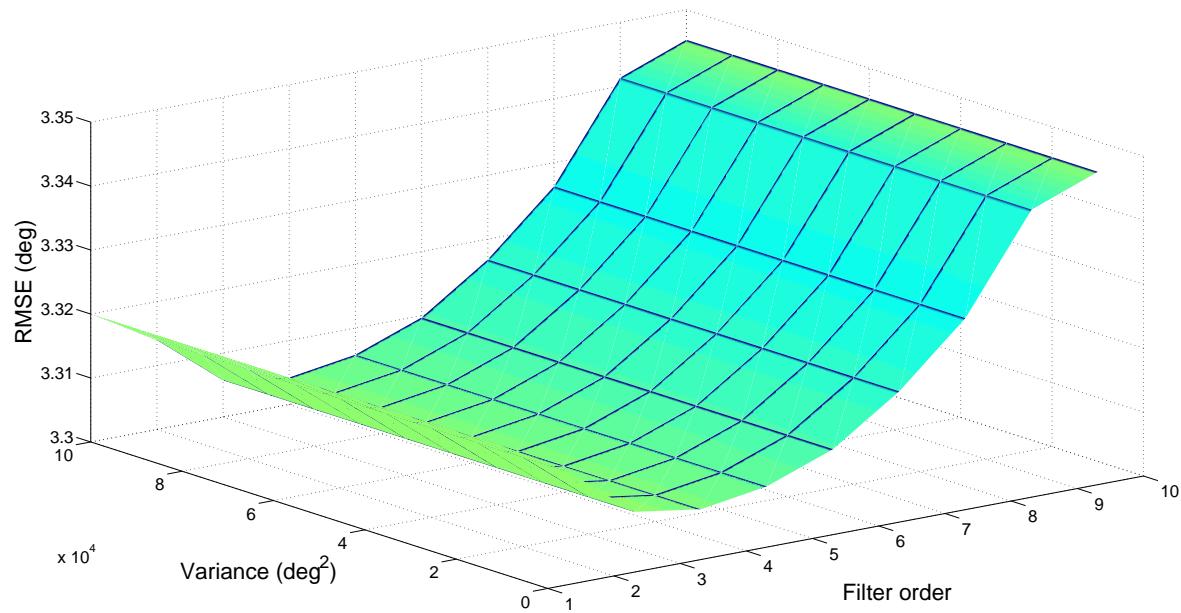


Figure 5.65: RLS parameter optimization procedure. Low intensity real signals.

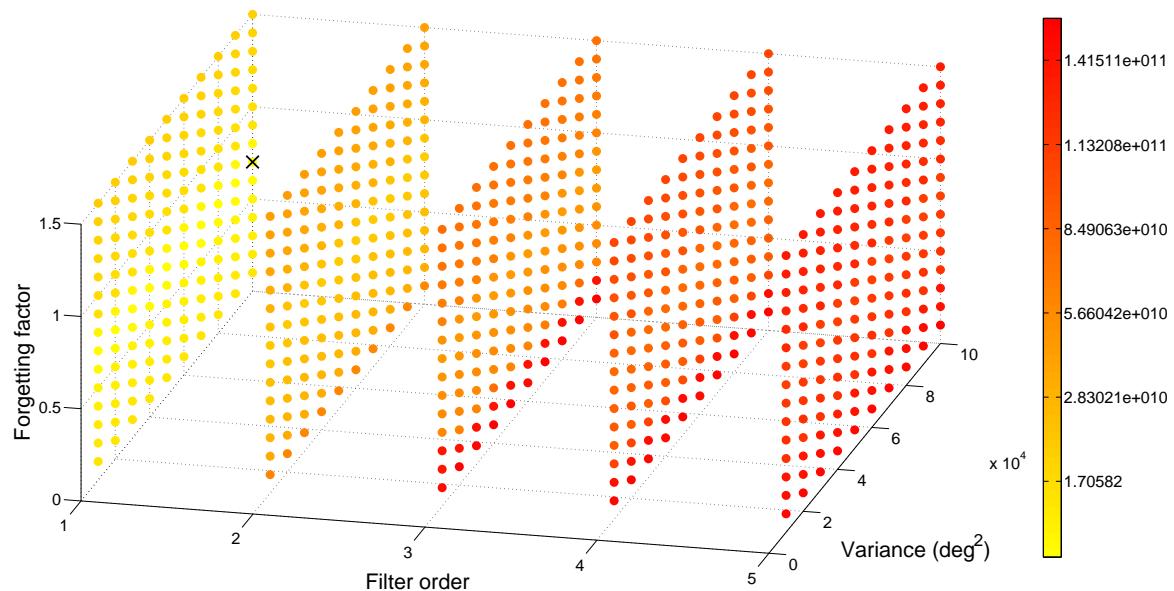


Figure 5.66: HRLS parameter optimization procedure. Low intensity real signals. The cross indicates the optimal point of operation.

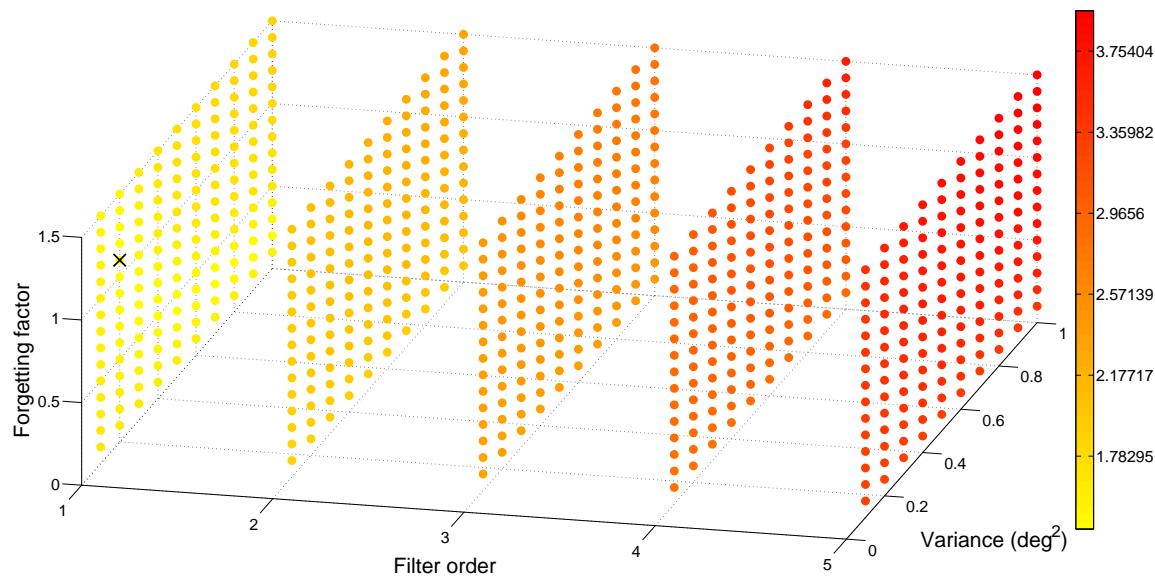


Figure 5.67: QRDRLS parameter optimization procedure. Low intensity real signals. The cross indicates the optimal point of operation.

Table 5.9: Application of sensor fusion attitude estimation algorithms on real signals presenting only low intensity. Average RMSE and average parameters.

Algorithm	Av. RMSE	First Parameter	Second Parameter	Third Parameter
Acc-Mag.	1.7974 ± 0.4760	—	—	—
NLMS	2.1249 ± 0.4565	$L=1.0000 \pm 0.0000$	$\mu=86.5000 \pm 8.3612$	—
MNLMS	2.1725 ± 0.4785	$L=1.0000 \pm 0.0000$	$\mu=3.3360e4 \pm 5.7700e4$	—
RLS	2.1614 ± 0.9999	$L=88.3300 \pm 78.7676$	$\delta=3.3350e4 \pm 5.7716e4$	—
HRLS	2.1204 ± 0.5221	$L=1.0000 \pm 0.0000$	$\lambda=0.5000 \pm 0.4358$	$\delta=1.0000e5 \pm 0.0000$
QRDRLS	2.1725 ± 0.4785	$L=1.0000 \pm 0.0000$	$\lambda=0.7667 \pm 0.6658$	$\delta=0.1667 \pm 0.1527$
Kalman	1.5643 ± 0.5327	$\alpha=0.0915 \pm 0.0851$	—	—
Madgwick	7.2513 ± 0.9167	$\beta=6.4000 \pm 0.6083$	—	—

Table 5.10: Pitch angle computed applying sensor fusion methods using optimal parameters from synthetic signals on a set of real signals presenting only low intensity.

Algorithm	Average RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	2.5229 ± 0.3999	$L=1.0000 \pm 0.0000$	$\mu=1.2094e2 \pm 7.0982$	—
MNLMS	15.8590 ± 2.4892	$L=1.0000 \pm 0.0000$	$\mu=7.0600e-6 \pm 1.4902e-6$	—
RLS	3.1029 ± 0.1002	$L=2.0000e2 \pm 0.0000$	$\delta=9.9600 \pm 0.19795$	—
HRLS	3.0580 ± 0.6860	$L=1.0000 \pm 0.0000$	$\lambda=0.5040 \pm 0.0282$	$\delta=0.0100 \pm 3.5047e-18$
QRDRRLS	15.8607 ± 2.4895	$L=1.0000 \pm 0.0000$	$\lambda=1.1260e3 \pm 2.4143e2$	$\delta=1.3680e2 \pm 35.8250$

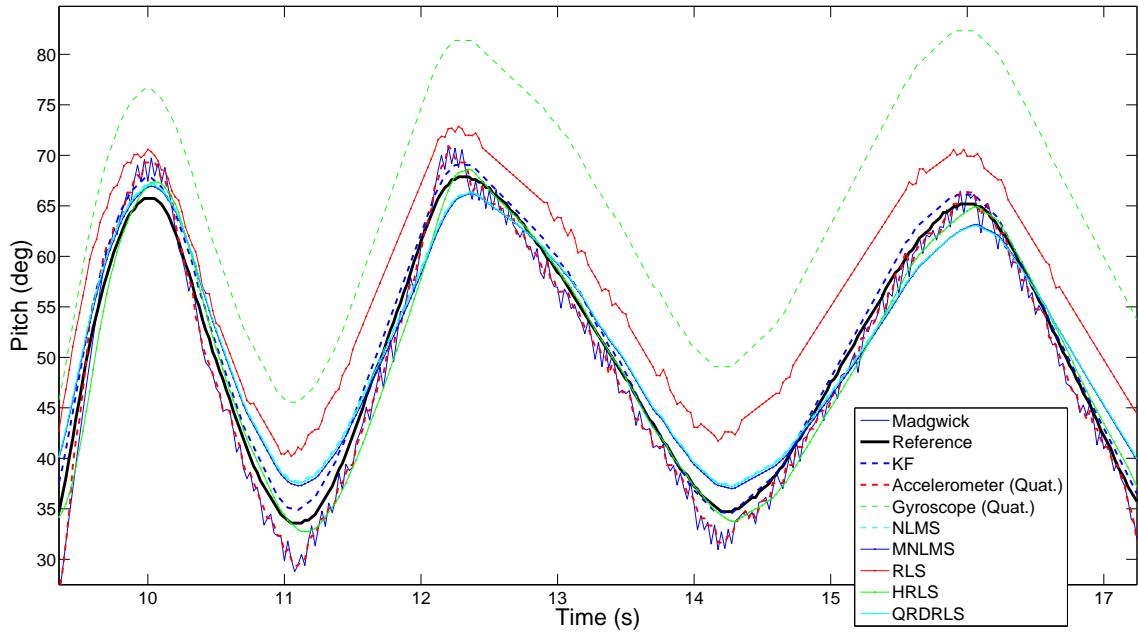


Figure 5.68: Pitch angle computed using sensor fusion algorithms applied on low intensity real signals.

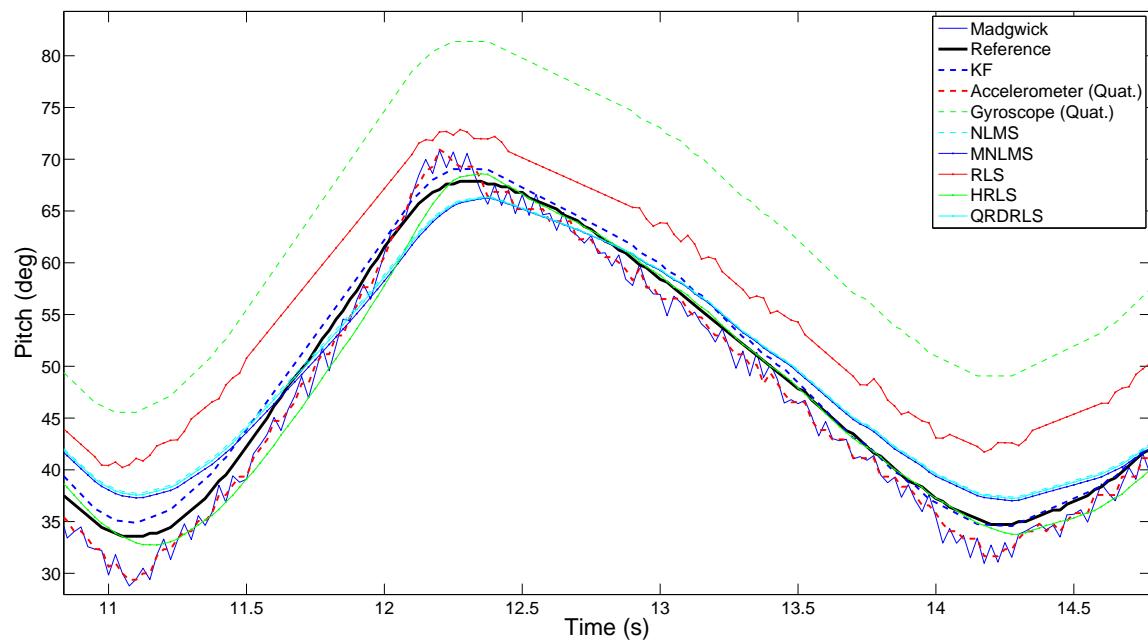


Figure 5.69: Pitch angle computed using sensor fusion algorithms applied on low intensity real signals (zoomed in).

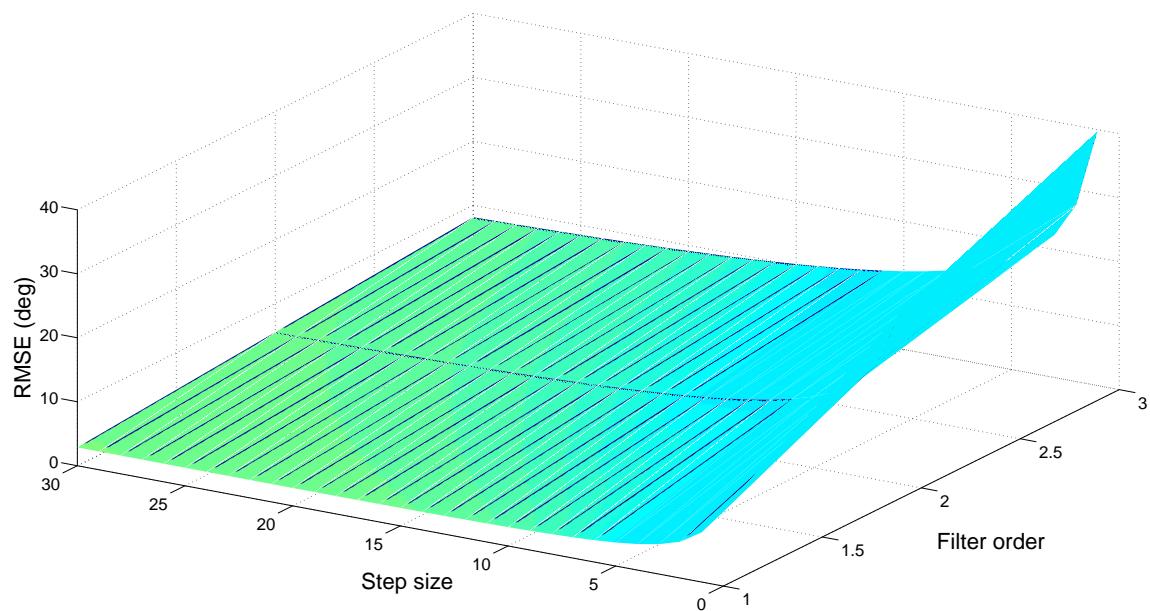


Figure 5.70: NLMS parameter optimization procedure. High and low intensity real signals.

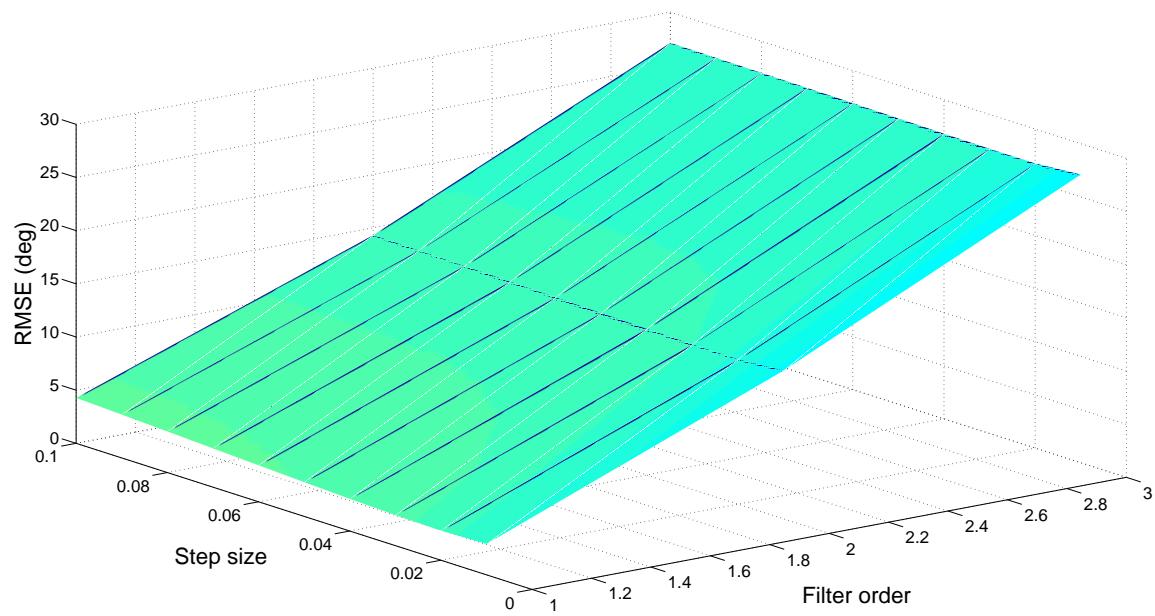


Figure 5.71: MNLMS parameter optimization procedure. High and low intensity real signals.

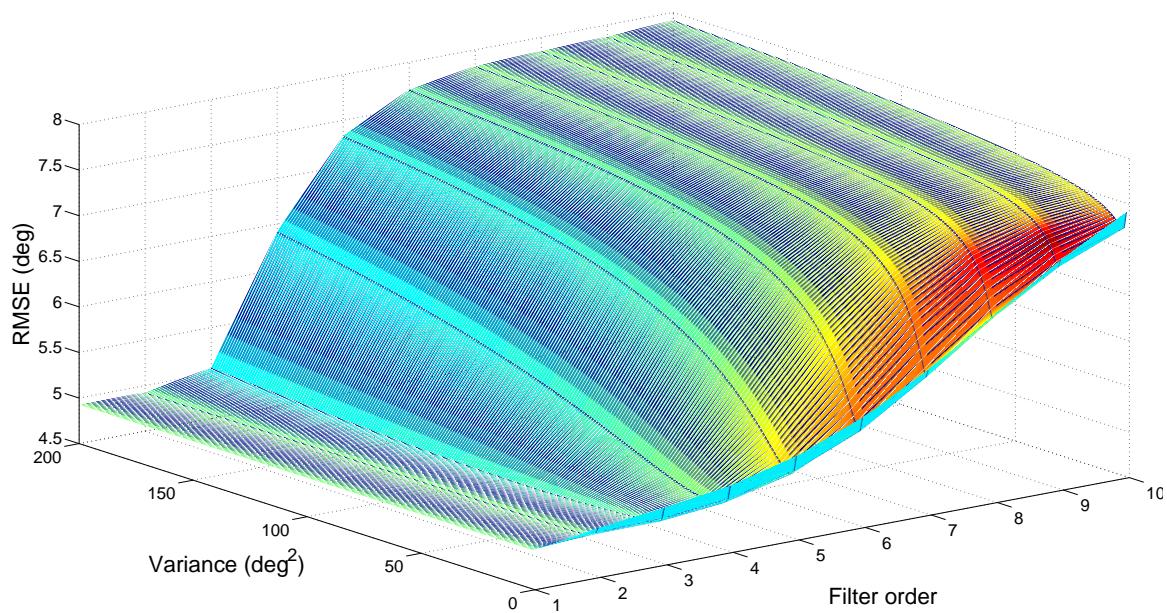


Figure 5.72: RLS parameter optimization procedure. High and low intensity real signals.

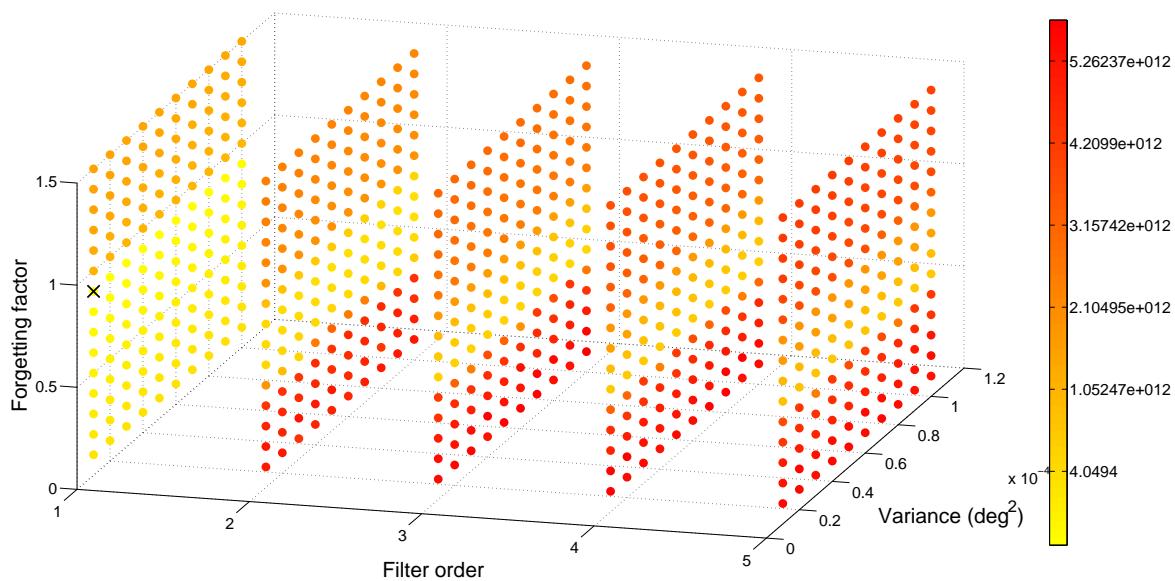


Figure 5.73: HRLS parameter optimization procedure. High and low intensity real signals. The cross indicates the optimal point of operation.

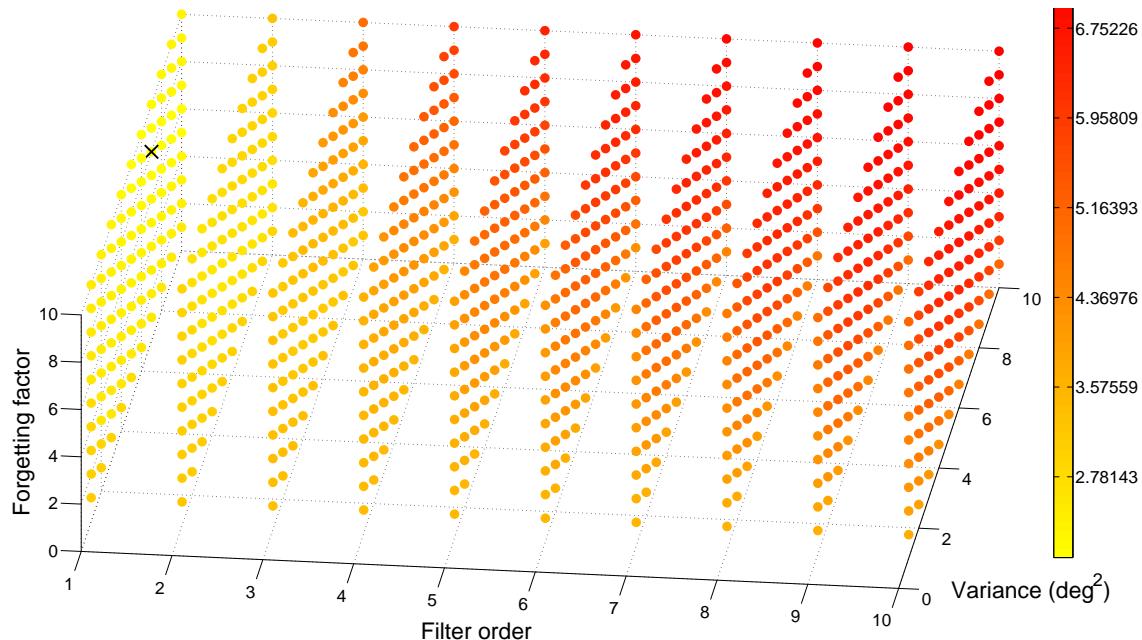


Figure 5.74: QRDRRLS parameter optimization procedure. High and low intensity real signals. The cross indicates the optimal point of operation.

Table 5.11: Pitch angle computed applying sensor fusion methods on a set of real signals presenting both medium and high intensity.

Algorithm	Optimizer	RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	Grid search	2.7987	L=1.0000	$\mu=19.0000$	—
NLMS	Gauss-Newton	2.6670	L=1.0000	$\mu=17.1323$	—
MNLMS	Grid search	2.7855	L=1.0000	$\mu=0.0300$	—
MNLMS	Gauss-Newton	2.6546	L=1.0000	$\mu=0.0165$	—
RLS	Grid search	4.8803	L=3.0000	$\delta=61.0000$	—
RLS	Gauss-Newton	4.8964	L=1.0000	$\delta=4.0672e4$	—
HRLS	Grid search	4.0494	L=1.0000	$\lambda=0.9000$	$\delta=1.0000e-5$
HRLS	Gauss-Newton	2.6271	L=1.0000	$\lambda=0.9807$	$\delta=0.0907$
QRDRRLS	Grid search	2.7814	L=1.0000	$\lambda=8.0000$	$\delta=7.0000$
QRDRRLS	Gauss-Newton	2.6546	L=1.0000	$\lambda=7.9050$	$\delta=7.6759$
Kalman	Gauss-Newton	2.1632	$\alpha=1.2200e3$	—	—
Madgwick	Gauss-Newton	5.4061	$\beta=1.0844$	—	—

Table 5.12: Yaw angle computed applying sensor fusion methods on a set of real signals presenting both low and very high intensity (complete signal).

Algorithm	Optimizer	RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	Grid search	6.6841	L=1.0000	$\mu=6.0000$	—
NLMS	Gauss-Newton	6.7964	L=1.0000	$\mu=5.1517$	—
MNLMS	Grid search	6.6945	L=1.0000	$\mu=7.0000e-4$	—
MNLMS	Gauss-Newton	6.7881	L=1.0000	$\mu=5.4900e-4$	—
RLS	Grid search	13.1165	L=1.0000	$\delta=1.0000e3$	—
RLS	Gauss-Newton	16.2992	L=1.0000	$\delta=6.9100e3$	—
HRLS	Grid search	5.1677	L=1.0000	$\lambda=0.9960$	$\delta=8.08$
HRLS	Gauss-Newton	5.1881	L=1.0000	$\lambda=0.9966$	$\delta=1.9190e3$
QRDRRLS	Grid search	6.6749	L=1.0000	$\lambda=48.0000$	$\delta=32.0000$
QRDRRLS	Gauss-Newton	6.7881	L=1.0000	$\lambda=59.3790$	$\delta=30.6759$
Kalman	Gauss-Newton	3.5430	$\alpha=143.8600$	—	—
Madgwick	Gauss-Newton	9.6170	$\beta=2.7440$	—	—

Table 5.13: Pitch angle computed applying sensor fusion methods using optimal parameters from synthetic signals on a set of real signals presenting both low and very high intensity (complete signal).

Algorithm	RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	2.8019	L=1	$\mu=17.2000$	—
MNLMS	14.4372	L=1	$\mu=2.894e-7$	—
RLS	4.9286	L=1	$\delta=46$	—
HRLS	2.7906	L=1	$\lambda=0.9768$	$\delta=8.0892$
QRDRLS	6.9034	L=1	$\lambda=29$	$\delta=29$

Table 5.14: Yaw angle computed applying sensor fusion methods using optimal parameters from synthetic signals on a set of real signals presenting both low and very high intensity (complete signal).

Algorithm	RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	8.3300	L=1	$\mu=17.2000$	—
MNLMS	56.8116	L=1	$\mu=2.894e-7$	—
RLS	16.1165	L=1	$\delta=46$	—
HRLS	8.8627	L=1	$\lambda=0.9768$	$\delta=8.0892$
QRDRLS	7.4741	L=1	$\lambda=29$	$\delta=29$

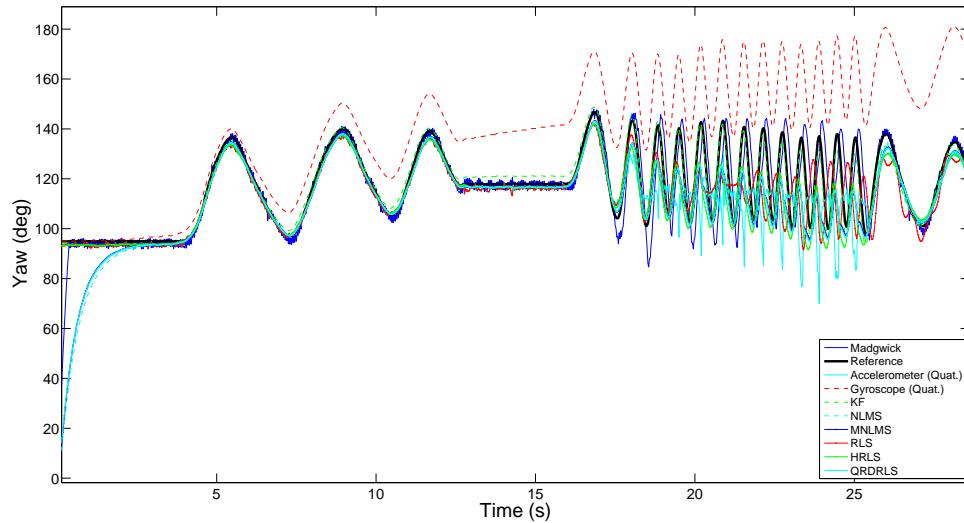


Figure 5.75: Yaw computed applying sensor fusion methods on real signals (very high and low intensity).

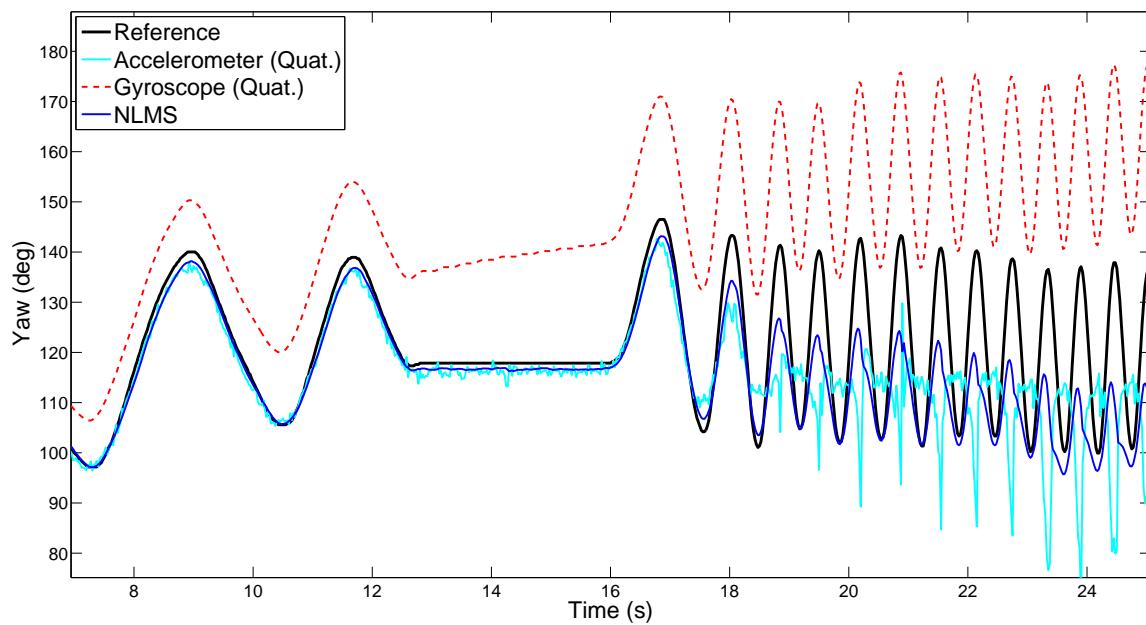


Figure 5.76: Yaw computed applying NLMS on real signals (very high and low intensity).

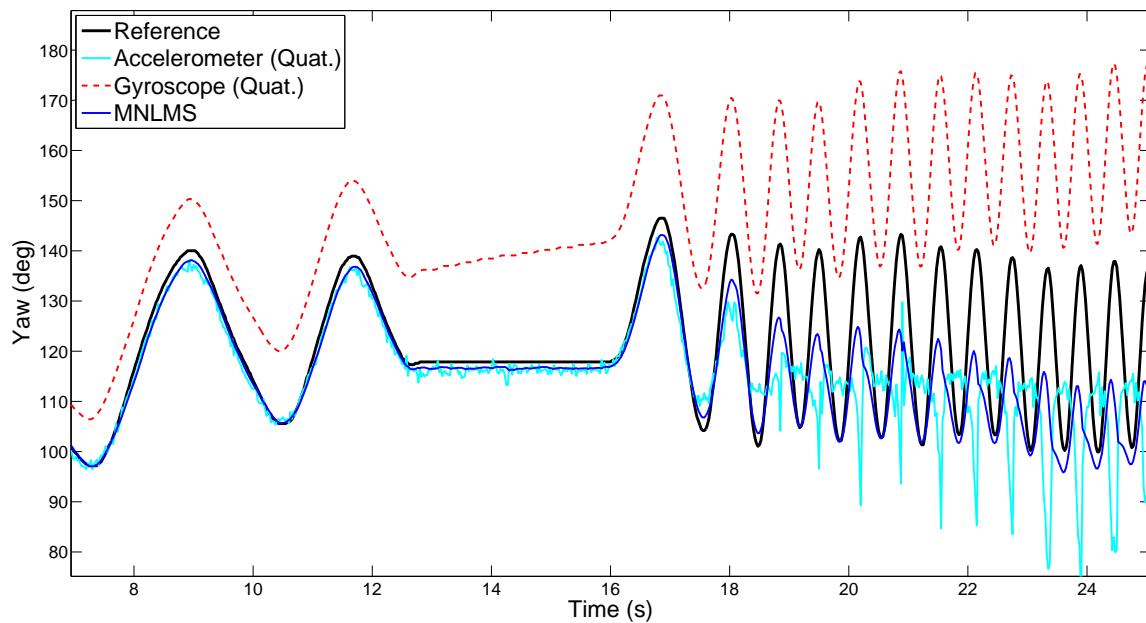


Figure 5.77: Yaw computed applying MNLMS on real signals (very high and low intensity).

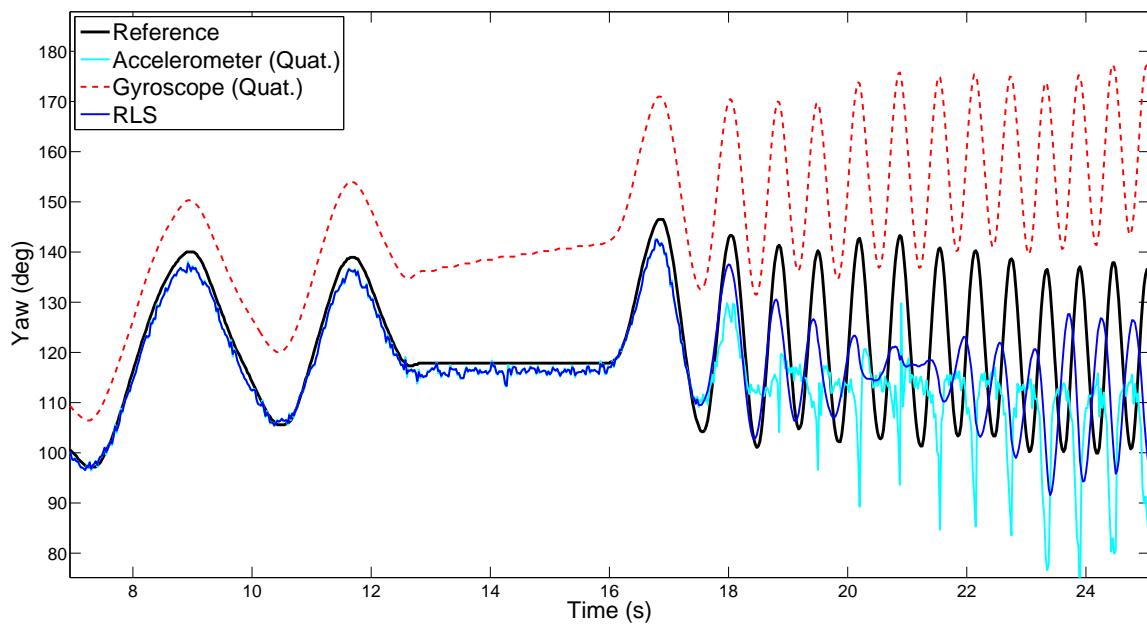


Figure 5.78: Yaw computed applying RLS on real signals (very high and low intensity).

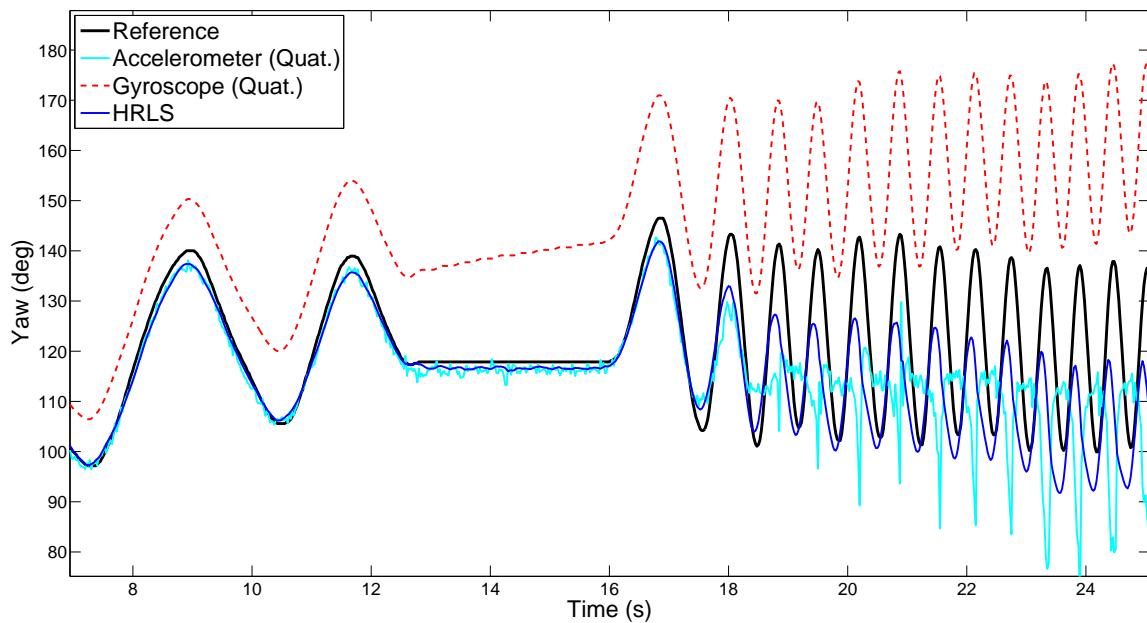


Figure 5.79: Yaw computed applying HRLS on real signals (very high and low intensity).

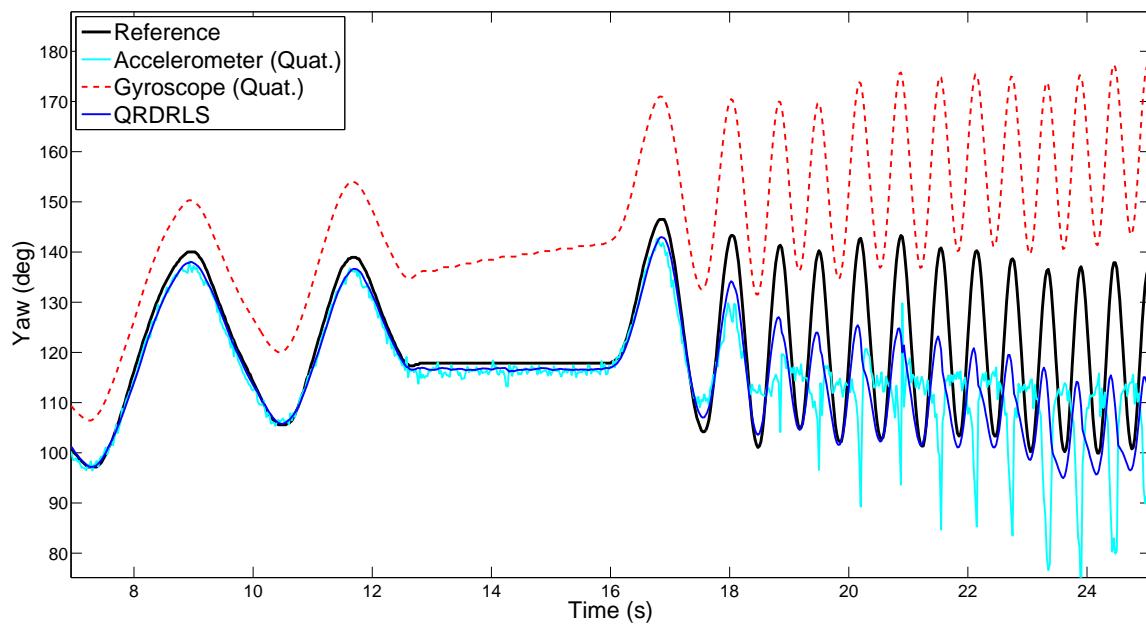


Figure 5.80: Yaw computed applying QRDRRLS on real signals (very high and low intensity).

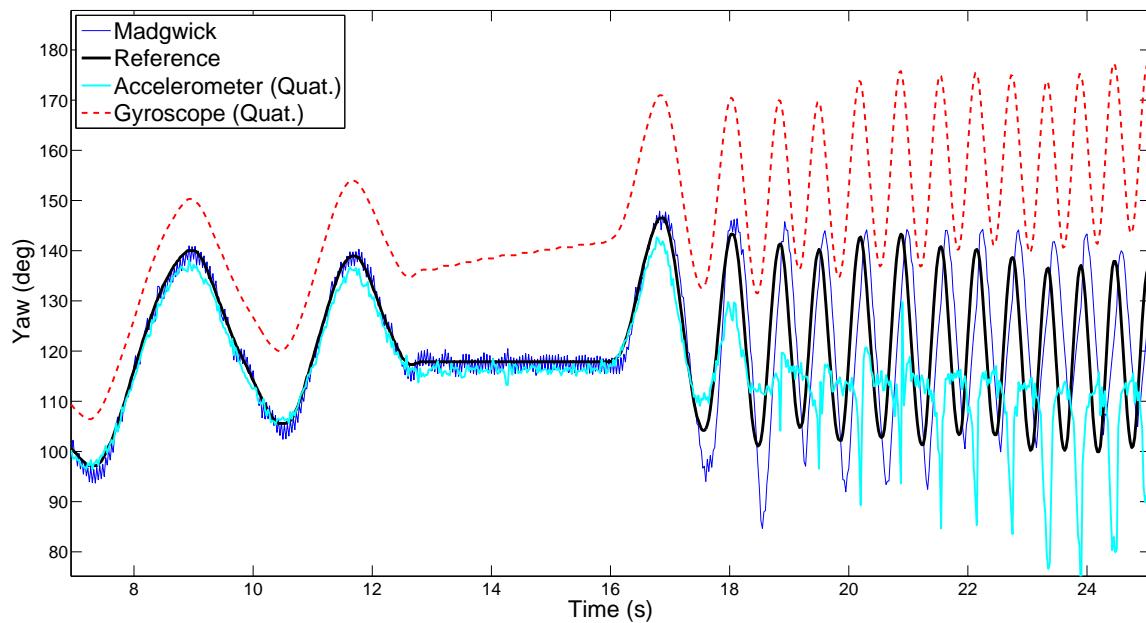


Figure 5.81: Yaw computed applying Madgwick's algorithm on real signals (very high and low intensity).

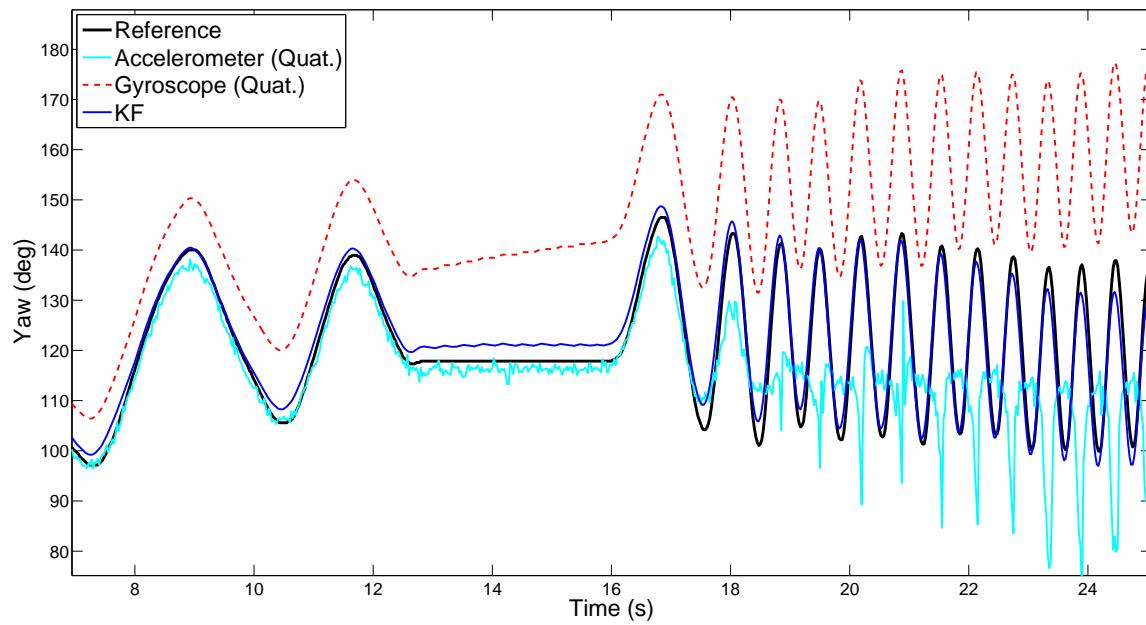


Figure 5.82: Yaw computed applying Kalman filter on real signals (very high and low intensity).

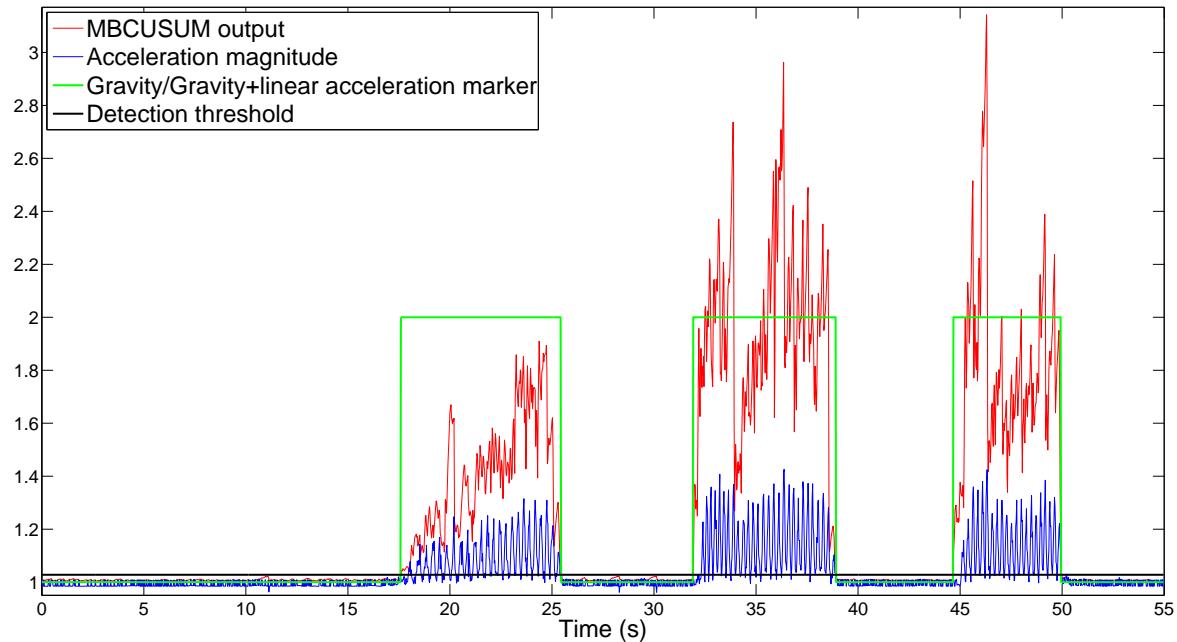


Figure 5.83: Detection of low and high intensity periods. Output of MBCUSUM detector vs. acceleration magnitude.

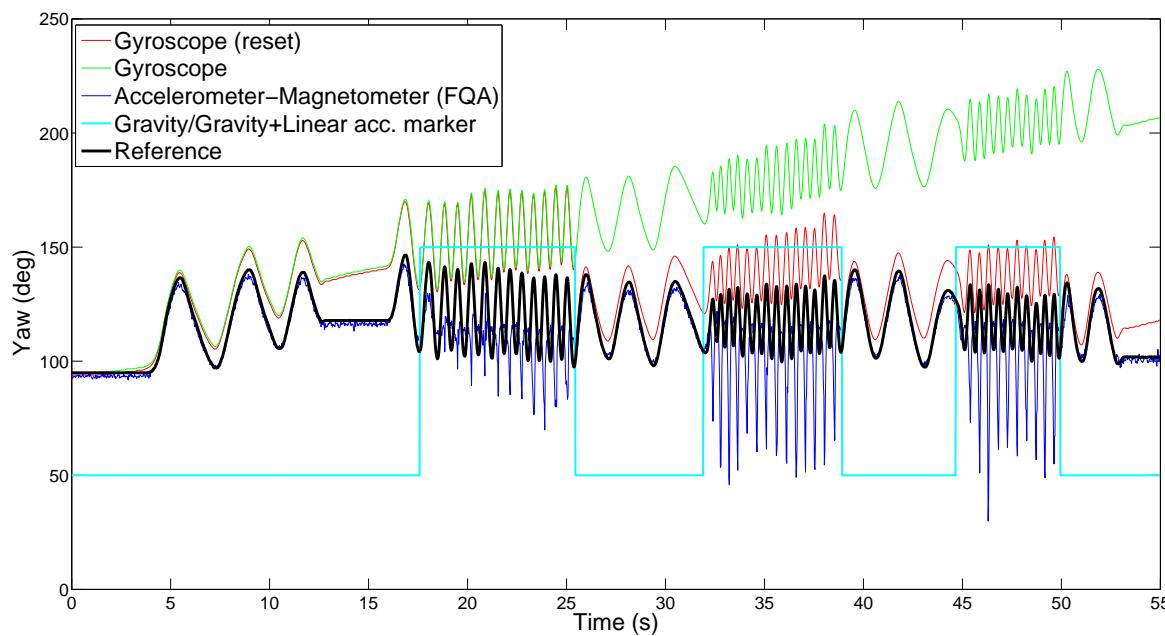


Figure 5.84: Estimation of Yaw. Non-fusion algorithms in addition to gating. Complete signal.

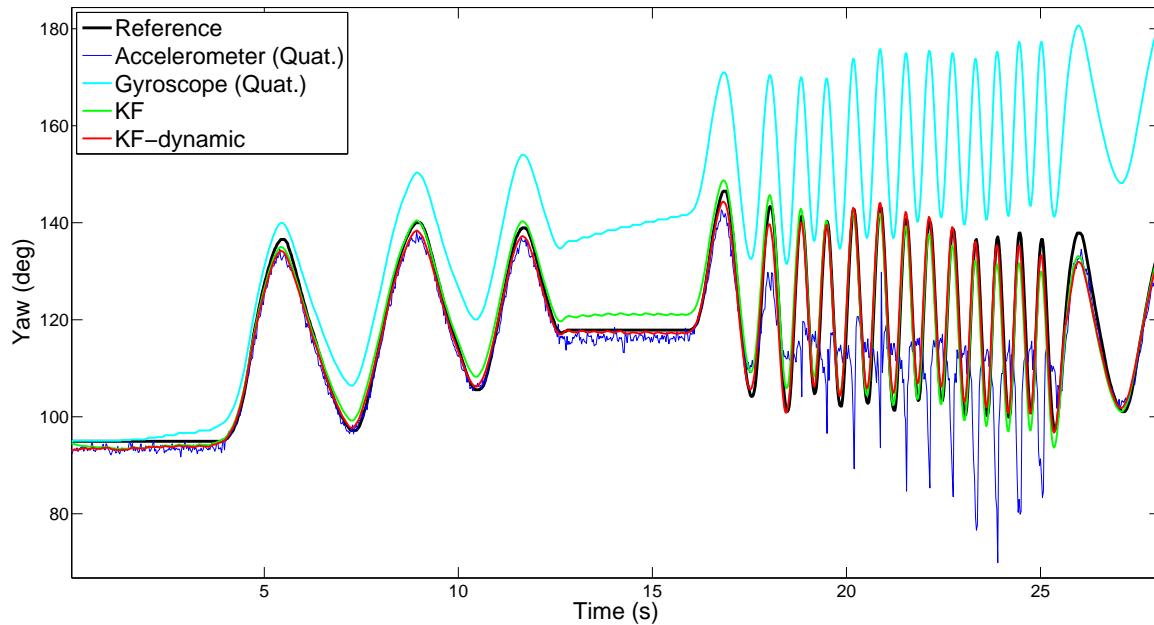


Figure 5.85: Estimation of Yaw. Regular Kalman filter vs. Gated-Kalman filter. First part.

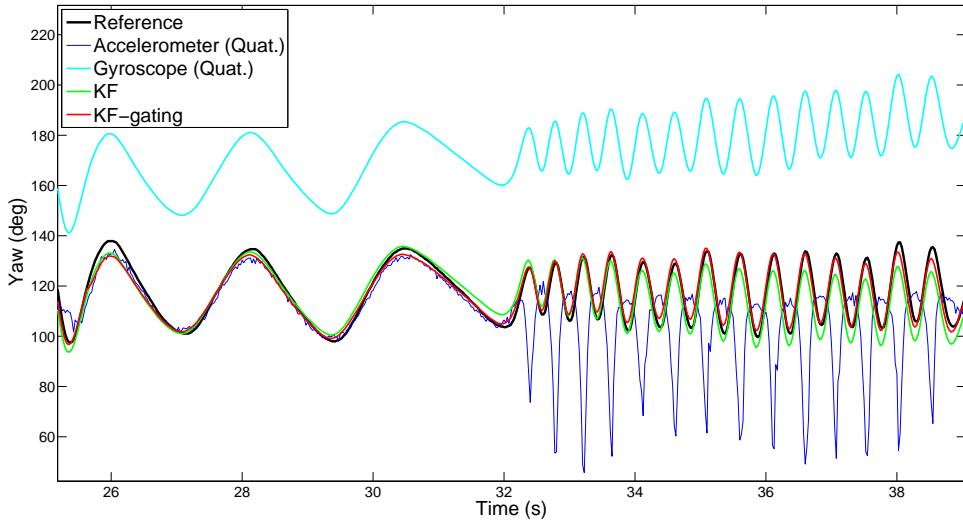


Figure 5.86: Estimation of Yaw. Regular Kalman filter vs. Gated-Kalman filter. Second part.

Table 5.15: Yaw angle computed applying gated sensor fusion methods on a set of real signals presenting both low and very high intensity (complete signal).

Algorithm	RMSE	First Parameter	Second Parameter	Third Parameter
NLMS	12.5256	L=6	$\mu=22$	—
MNLMS	12.3191	L=5	$\mu=0.005$	—
RLS	9.8036	L=1.0000	$\delta=10$	—
HRLS	9.5355	L=1.0000	$\lambda=1.4$	$\delta=1$
QRDRRLS	12.3094	L=6.0000	$\lambda=46$	$\delta=5$
Kalman	2.1132	$\alpha_1=1.1450$	$\alpha_2=168.8207$	—

5.4 Discussion of results

We have presented a large set of results that we now proceed to analyze in detail. We will determine which filter has shown the best behavior under each one of the three different intensity conditions (low, high and both low and high). In addition, we will explain the observed behavior of the algorithms with respect to their input parameters and we will discuss the repeatability of their optimal values. Finally, we will explain the advantages and disadvantages of using a motion intensity detector to reset the integrated angular rate signal.

We start the discussion by analyzing the results thrown by the theoretical simulations.

5.4.1 Theoretical signals

5.4.1.1 High intensity

When the algorithms are applied on theoretical signals presenting only long periods of high intensity motion, the obtained average RMSE is higher than 10° in all the cases. Figures 5.22 and 5.23 show how all the filters have some trouble to completely remove the time-growing bias present in the Gyroscope angle signal. Among them, NLMS is the one achieving a higher degree of precision, while RLS has the most trouble converging to a bias free signal. Observing figures 5.87 and 5.88 we observe that both filter taps decrease with time to compensate for the growing bias. However, the final value of the filter taps for NLMS is 0.7359 and 0.881 for RLS, i.e. NLMS compensates better for the bias.

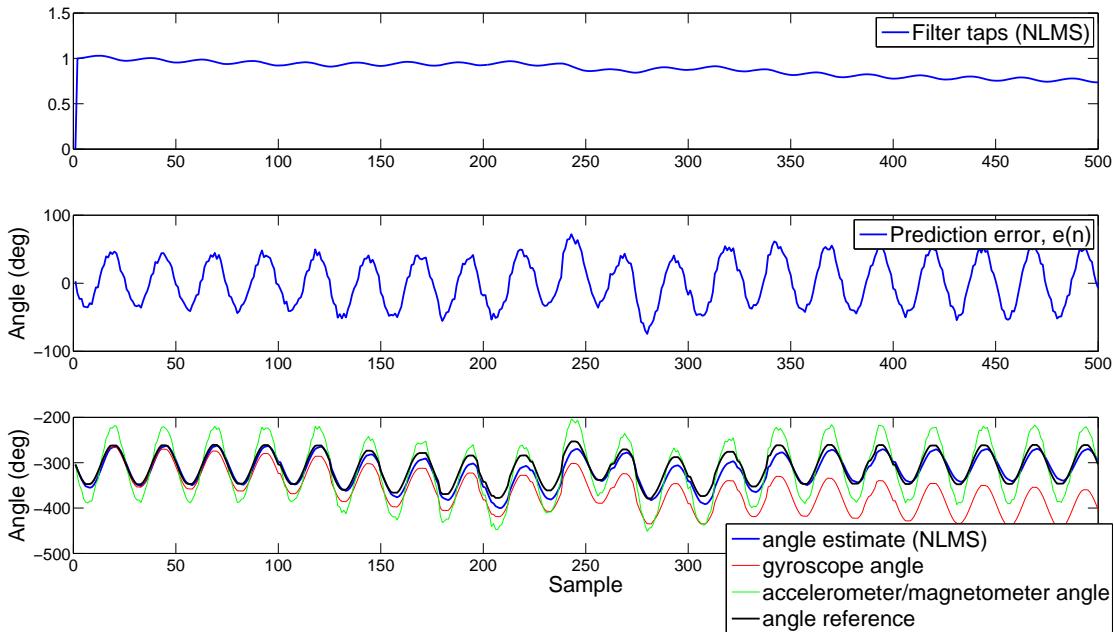


Figure 5.87: Filter taps vs. prediction error vs. angle estimate. NLMS over high intensity synthetic signals using optimal parameters.

If we take a look to the parameters, we see that all algorithms reach their optimal performance for a filter order of $p=1$. Increasing the filter order leads to a larger converge time and to a slight time shift which causes the RMSE to augment.

The optimal value of the step size is rather small for both NLMS and MNLMS

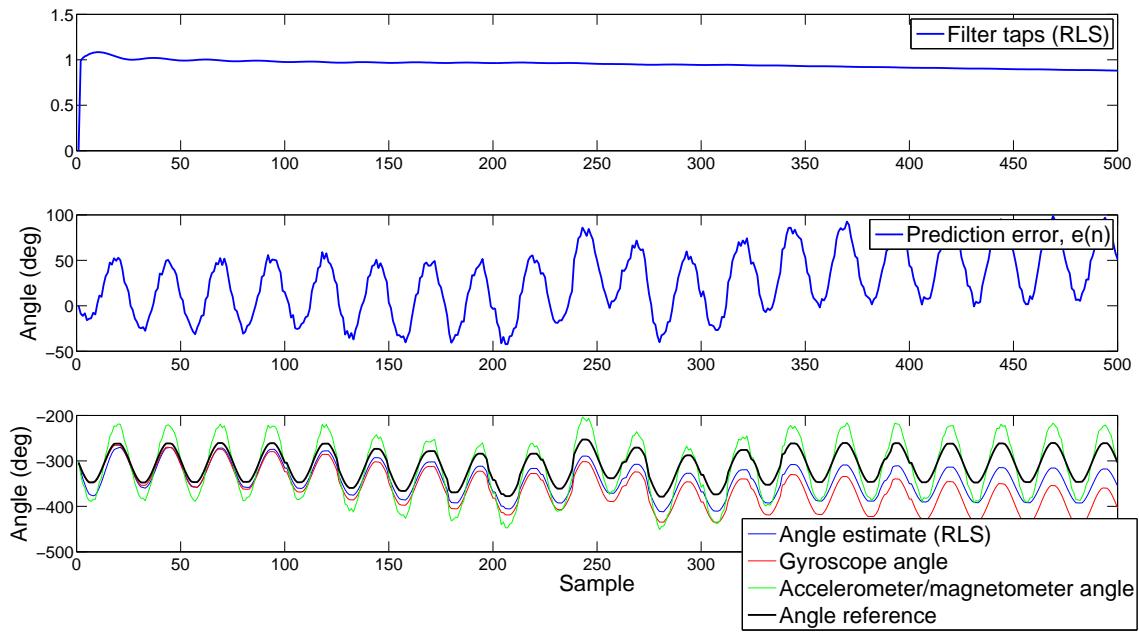


Figure 5.88: Filter taps vs. prediction error vs. angle estimate. RLS over high intensity synthetic signals using optimal parameters.

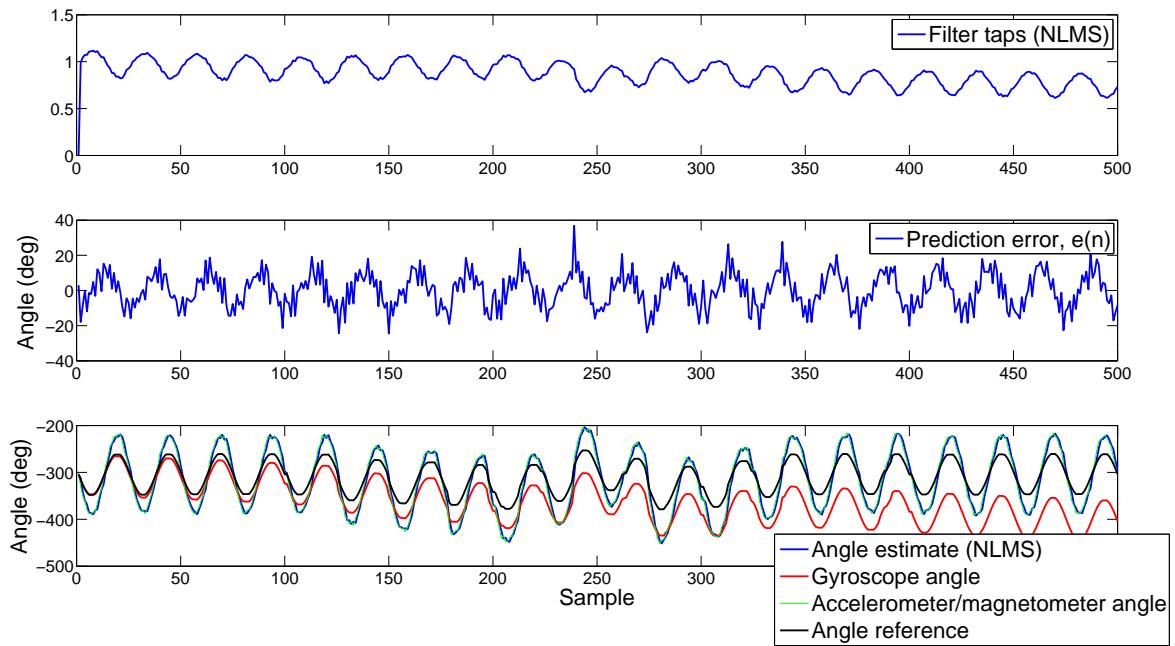


Figure 5.89: Filter taps vs. prediction error vs. angle estimate. NLMS over high intensity synthetic signals using non optimal step size ($\mu = 1$).

to prevent the filter from following the abrupt changes in the accelerometer-magnetometer angle signal. A larger step size gives more weight to the update of the taps based on the prediction error (equations 5.109-5.110). The prediction error becomes larger during the instants in which the accelerometer-magnetometer signal is more affected by large dynamic acceleration. Therefore, for high intensity signals, the prediction error can become quite large during these time instants (see figure 5.87). If the step size is large, then the taps will be updated in a more abrupt way in order to reduce the prediction error (see figure 5.89). We, thus, need to keep the step size low to filter out the erroneous peaks of the accelerometer-magnetometer signal by preventing the taps from being adapted to the observation signal (which is absolutely erroneous during high intensity periods). Therefore, the step size acts as a control parameter. By modifying it we can adapt the degree of adaptation of the estimated signal to the observation signal.

The value to initialize the covariance matrix (δ), is the second parameter of the RLS filter. Since it is only used to set the initial value, it has very short influence in the performance of the filter as, for example, two distant values like $\delta = 1e3$ and $\delta = 1e-3$ converge to an almost coincident set of tap filters almost immediately (see figure 5.90). As we said in section 5.2.4.2, the RLS algorithm recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. Such linear function is $\alpha(n) = d(n) - \hat{w}_{n-1}^T u(n)$, where, in our case, $d(n)$ is the value of the accelerometer-magnetometer angle signal, $u(n)$ is the segment of the gyroscope angle signal which is multiplied by the filter taps \hat{w}_{n-1}^T . It happens that, under high motion conditions, the accelerometer-magnetometer and the gyroscope angle signals are not related linearly, which makes the problem ill posed.

The initial value of the covariance matrix is almost indifferent for the performance of HRLS. On the other hand, the value of the forgetting factor (λ) is decisive for its behavior. A large value of λ gives more weight to the previous samples of the input signal. It also means a very low weight of the prediction error when updating the taps (equations 5.120 and 5.122). Figure 5.91 shows how the filter taps are stuck at the unity and do not compensate for the time-growing bias.

On the other hand, QRDRRLS has the opposite behavior in terms of λ . Looking at equations (5.123)-(5.126) we can see how a large λ gives less weight to the prediction error in the tap update equation, while a small value causes the taps to tightly follow the prediction error. Figures 5.92 and 5.93 show the filter taps, prediction error and estimated angle signal for $\lambda_{\text{optimal}} = 3740$ and $\lambda = 1$ respectively.

In conclusion, when the signal presents high intensity motion, the filter taps need to compensate for the time-growing bias present in the gyroscope angle signal without following in excess the prediction error. A following error which, in

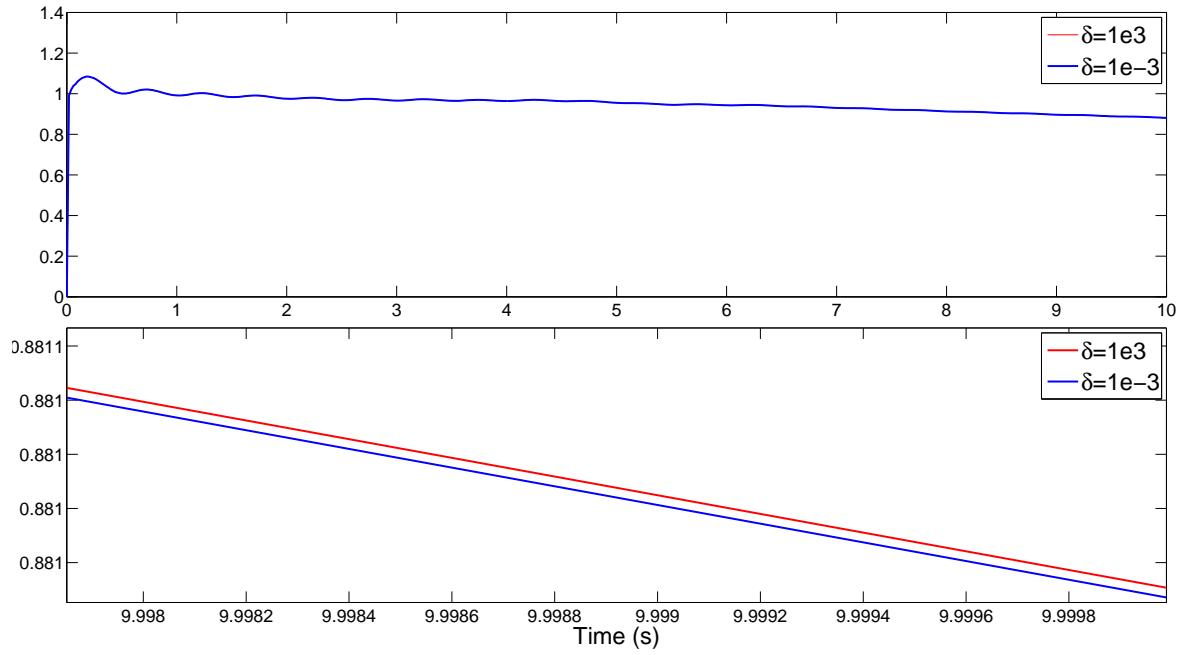


Figure 5.90: Comparison of filter taps. RLS over high intensity synthetic signals using two different initial covariance matrices ($\delta = 1e3$ and $\delta = 1e - 3$).

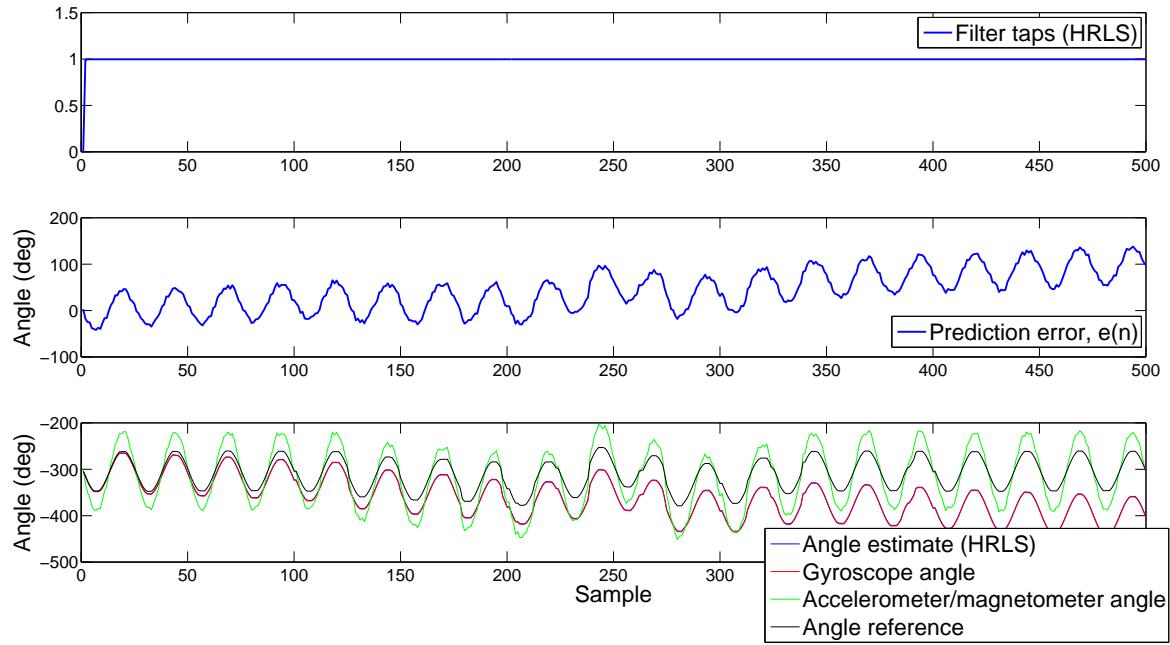


Figure 5.91: Filter taps vs. prediction error vs. angle estimate. HRLS over high intensity synthetic signals using non optimal forgetting factor ($\lambda = 1$).

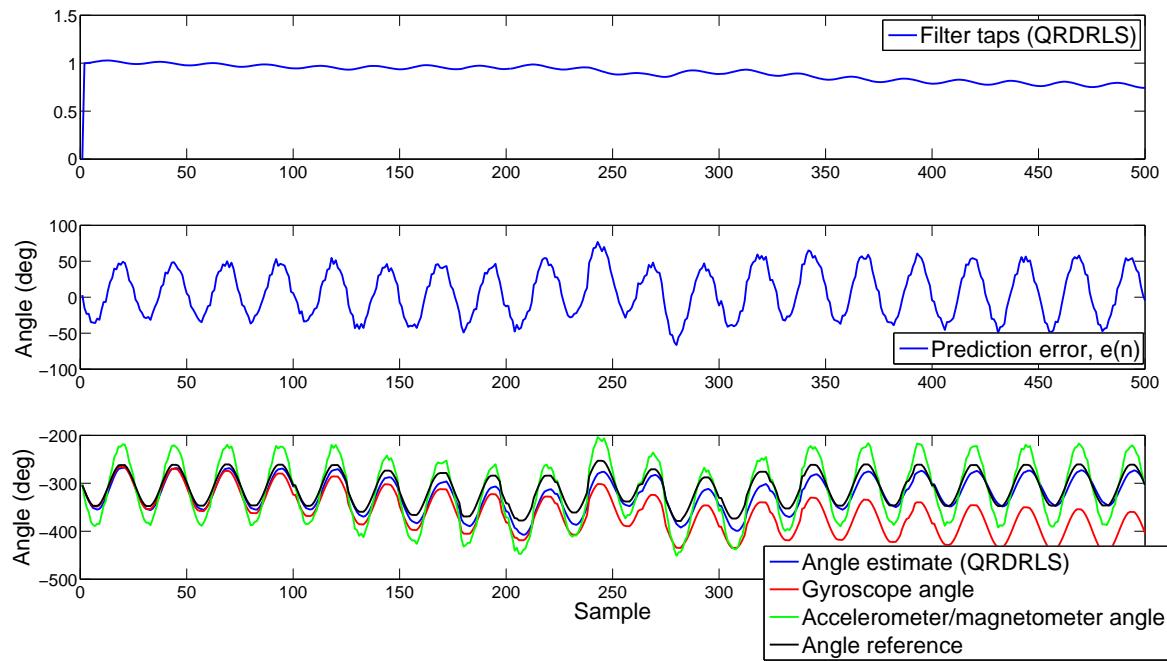


Figure 5.92: Filter taps vs. prediction error Vs. angle estimate. QRDRRLS over high intensity synthetic signals using optimal forgetting factor ($\lambda = 3740$).

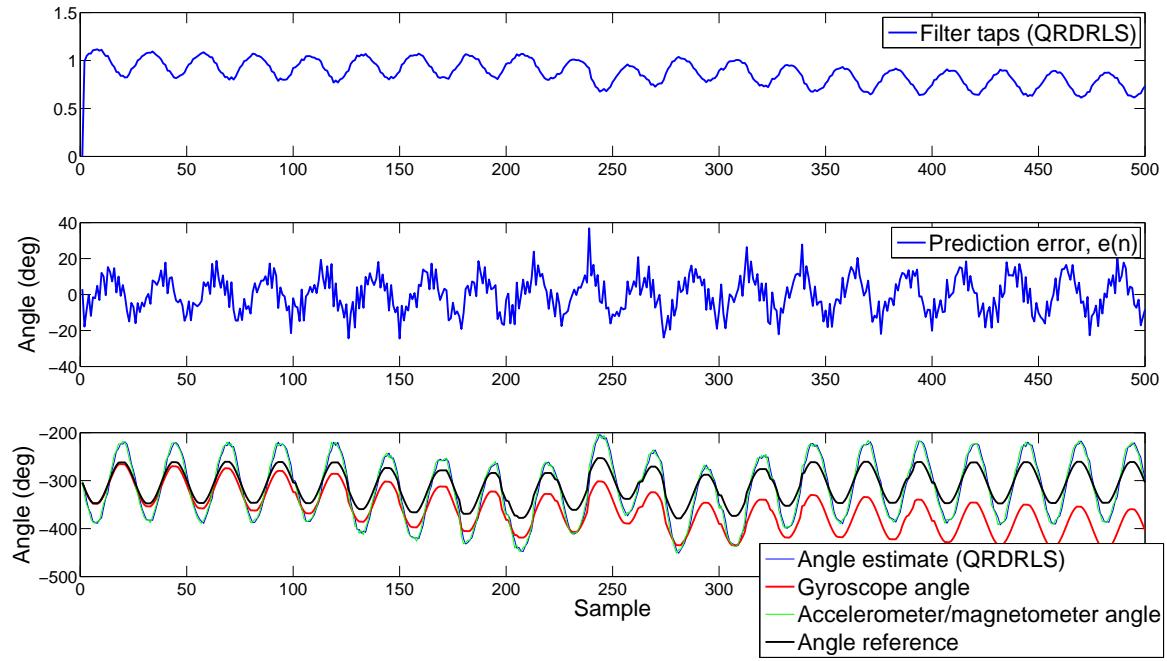


Figure 5.93: Filter taps vs. prediction error vs. angle estimate. QRDRRLS over high intensity synthetic signals using non optimal forgetting factor ($\lambda = 1$).

this case, is rather high due to the undesired effects of the linear acceleration in the accelerometer-magnetometer angle signal. Let's continue now studying the behavior of the filters in the opposite situation, that is, signals gathered under low intensity motion conditions.

5.4.1.2 Low intensity

Again, all filters but RLS perform better for a filter order of $p = 1$. This time, for low intensity signals, RLS reaches its optimal point of operation for $p = 200$. The performance now is slightly better now as both input signals can be assumed to be related linearly.

If we compare the step size of both NLMS and MNLMS filters we see that now, its value is higher. This time, since the accelerometer-magnetometer angle signal is always very accurate, the error of the prediction will not be large, and therefore, we can give it more importance to update the taps. Now the weights are able to compensate for the time-growing bias present in the gyroscope angle signal without being distorted by occasional large prediction errors.

As it happened with high intensity signals, the initial value of the covariance matrix has again very limited impact in the performance of RLS, HRLS and QRDRRLS filters. As for the forgetting factor in HRLS and QRDRRLS, the optimal values decrease from $0.9000 \pm 8.927e - 16$ to 0.5040 ± 0.0282 and from 3740 ± 1225 to 1126 ± 241.43 respectively. Like we explained before, since now the prediction error is smaller we can allow a faster convergence of the taps without fearing occasional disturbances coming from large prediction errors. Figures 5.94 and 5.95 show the filter taps, prediction error and estimated angle signal computed from low intensity signals using NLMS and QRDRRLS respectively.

The synthetic simulations have revealed that NLMS outperforms the rest of the tested algorithms and might be a good option to estimate the attitude when the motion is not very intense.

5.4.1.3 Low and high intensity

Until now, we have studied two extreme situations in which the signals are only composed of high intensity or low intensity. The behavior of the filters based on the value of their parameters has been described and we now know what values should be used for each one of the situations. However, in real applications, the body being monitored will usually move alternating high intensity motion with low intensity motion. Therefore, the value of the parameters should be different to those obtained in the two first situations. Depending on the predominant kind of motion, the value

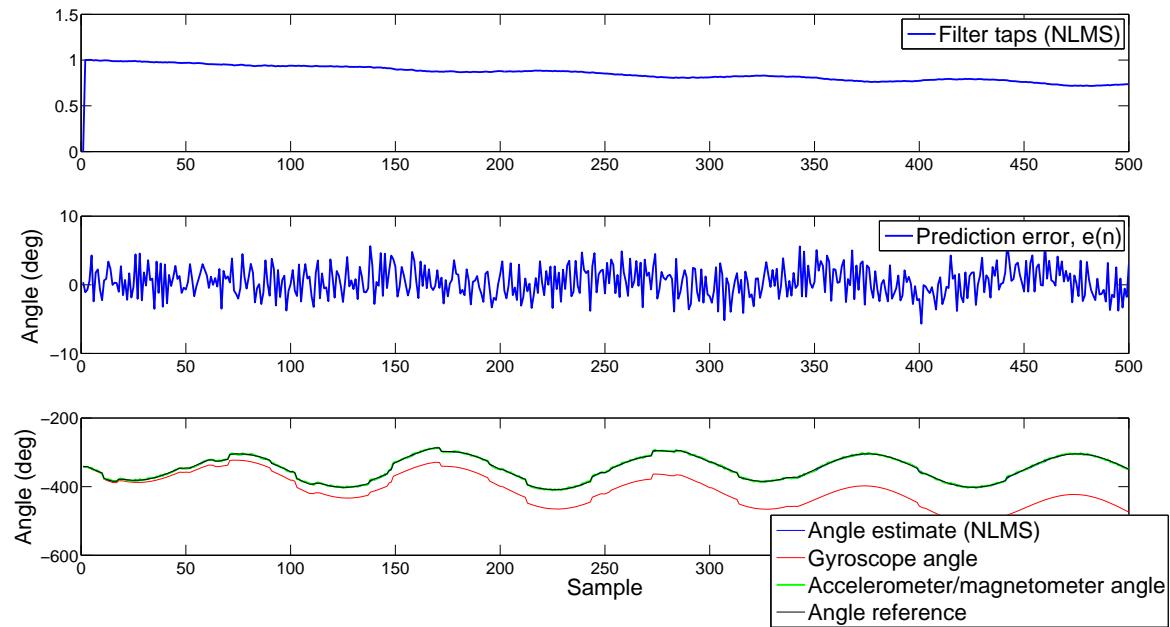


Figure 5.94: Filter taps vs. prediction error vs. angle estimate. NLMS over low intensity synthetic signals using optimal step size ($\mu = 0.4816$).

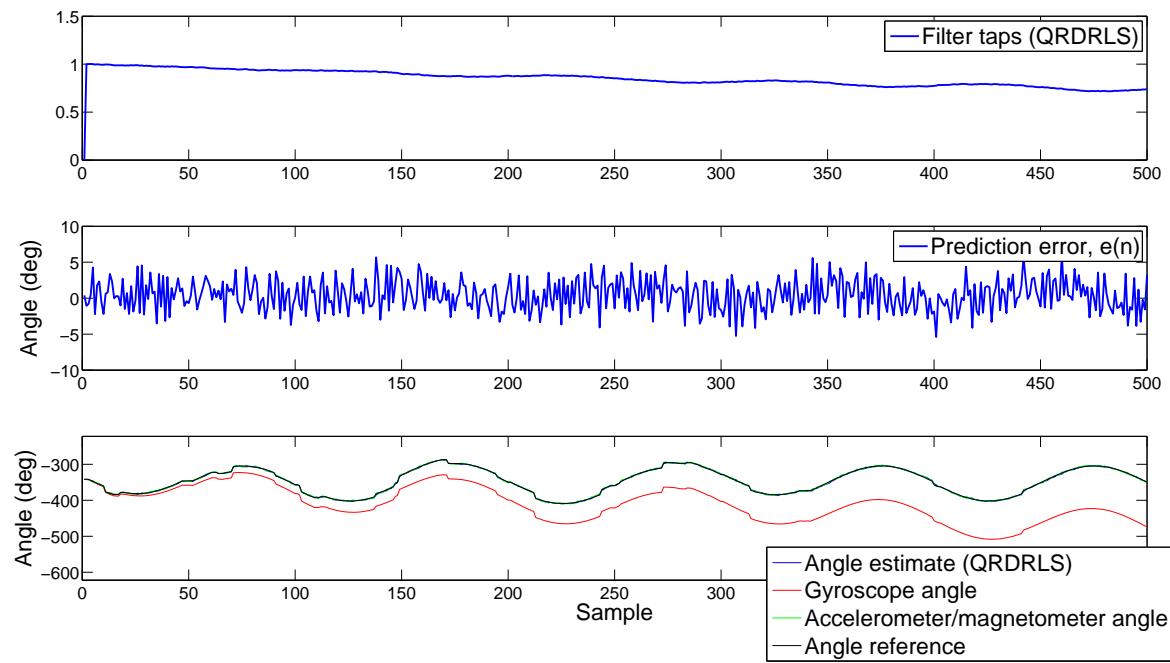


Figure 5.95: Filter taps vs. prediction error vs. angle estimate. QRDRRLS over low intensity synthetic signals using optimal forgetting factor ($\lambda = 1126$).

of the parameters will be closer to some of the extreme values. The simulation shows that, for mixed signals, the best performance is achieved for those values closer to those obtained in the only high intensity motion simulation. The objective of the optimizers is to find the parameters minimizing the RMSE of the complete signal. Then, since a much higher RMSE is obtained when applying low intensity parameters on high intensity signals than when applying high intensity parameters on low intensity signals, the optimizer returns values closer to high intensity parameters (see table 5.4). Figures 5.96 and 5.97 show the taps of the filter, the estimation error and the angle estimate for NLMS and HRLS applied on mixed signals respectively. Notice how the taps oscillate slightly during high intensity motion periods and how they converge smoothly during low intensity periods. The final value of the taps (0.7829 for NLMS and 0.7774 for HRLS) compensates successfully for the bias of the gyroscope signal since, when multiplied by the value of the gyroscope signal (-275.3°) at the last time instant K yields a value close to the actual angle (-215.4°) also for $t = K$:

$$\begin{aligned}\hat{\alpha}_{\text{NLMS}}(K) &= -275.3^\circ \cdot 0.7890 = -217.21^\circ \approx -215.4^\circ \\ \hat{\alpha}_{\text{HRLS}}(K) &= -275.3^\circ \cdot 0.7774 = -214.01^\circ \approx -215.4^\circ\end{aligned}\quad (5.127)$$

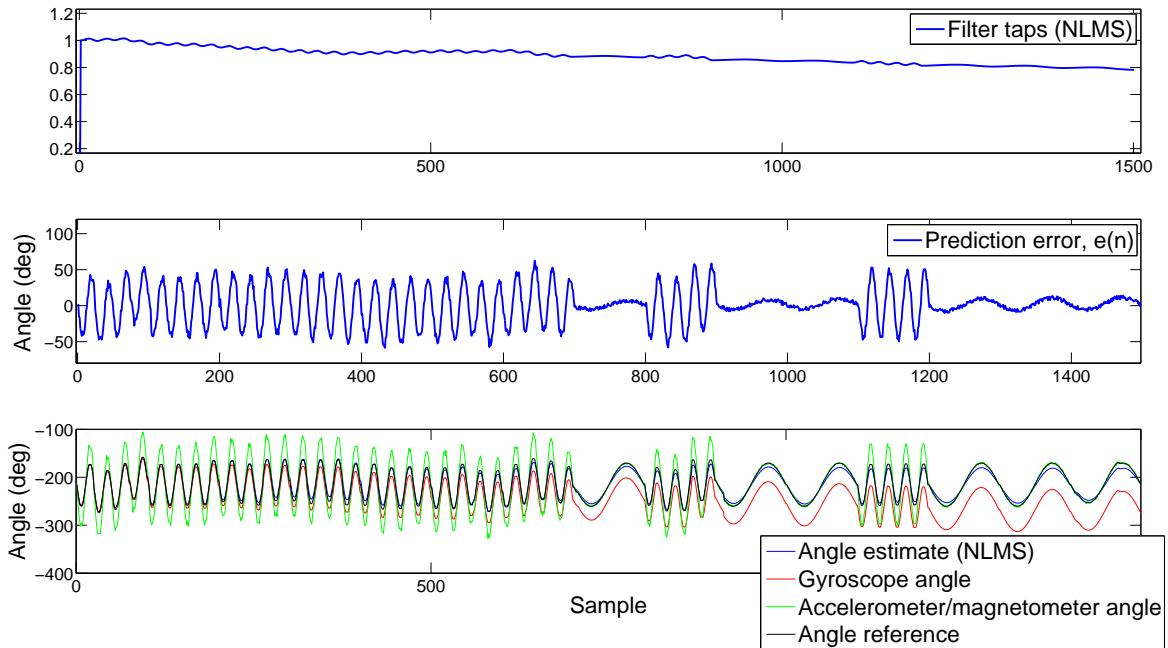


Figure 5.96: Filter taps vs. prediction error vs. angle estimate. NLMS over mixed synthetic signals using optimal step size ($\lambda = 0.0120$).

Resetting the bias of the gyroscope angle signal has revealed not to be a good option for all the tested filters. The resetting introduces a discontinuity that affects

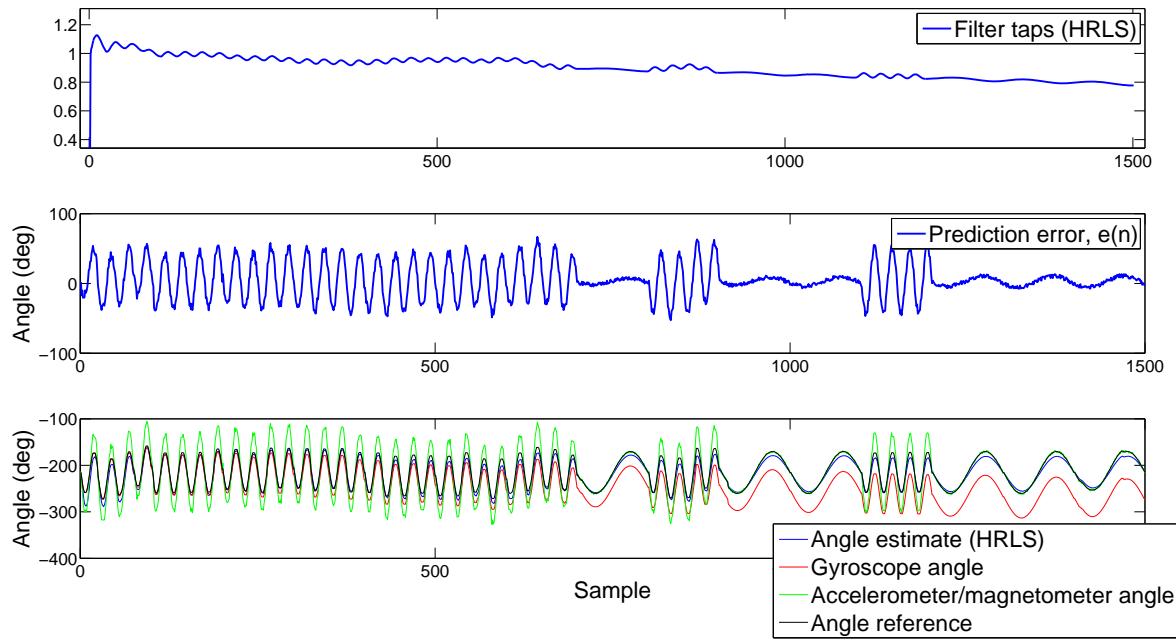


Figure 5.97: Filter taps vs. prediction error vs. angle estimate. HRLS over mixed synthetic signals using optimal step size ($\lambda = 0.0120$).

the convergence of the filter as it introduces a large error between the previous state and the current state.

5.4.2 Real signals

The results obtained from applying the algorithms on real signals are expected to differ slightly from those obtained by the theoretical experiments because the synthetic signals are similar, but not equal to the real signals gathered using a MIMU. Another factor that may cause the results to be different is the smaller number of signals employed to carry out the experiments. However, we expect some of the theoretical results to be reproducible with real signals.

The experiments were started applying non-fusion attitude estimation approaches. Tables 5.5 and 5.6 show the RMSE for an example of computation of pitch and yaw angles respectively. All the methods present a very high RMSE because of the undesired effects inherent to each one of the approaches. The accelerometer-magnetometer based method reaches very large error during the time instants in which the measured acceleration is mainly composed of linear acceleration or under the effect of magnetic perturbations. The FQA algorithm has shown to be more robust than the Euler-based approach. For the yaw estimation (figure 5.53) we can see how the Euler method presents many discontinuities, caused by the evaluation

of trigonometric expressions explained earlier in the chapter, which are not present in the quaternion approach. From these poor results, the already explained necessity of using a sensor fusion approach has been made patent. We begin now by analyzing the results coming from the applications of adaptive filters on signals which are exclusively composed of high intensity motion.

5.4.2.1 High intensity

Table 5.7 shows the average results of the application of the algorithms on a set composed of five high intensity real signals. The reason for such a reduced number of signals is mainly due to the large amount of time needed to mount the sensors in the angle reference device, carry out the desired maneuvers, extract, correct and calibrate the data and compute the orientation angles for each one of the signals. We aim to complete the database in the future by automatizing the complete process.

Returning to the table, we see that the results with real signals improve those obtained with synthetic signals. It is specially remarkable the case of RLS which showed poor performance with synthetic signals and which error has now decreased to 6° . This may be due to various reasons: the fact that dynamic bias in synthetic signals is larger than in real signals—the synthesizer was designed this way to simulate worst case scenarios in which gyroscopes have large drift—the smaller number of signals used in the experiments—5 versus 50—or other possible differences of synthetic signals with respect to real signals. We aim to solve this issue by designing in the future a synthesizer based on physical laws modeling real signals more accurately. Among all the tested methods, HRLS achieves the highest precision followed by Kalman filter.

On the other hand, if we analyze the optimal parameters, we see that those having more weight in the performance (filter order, step size and forgetting factor), show similar results. It is worth pointing out that the optimal parameters of QR-DRLS and HRLS are almost identical for synthetic and real signals.

If we take a look to figures 5.56-5.60 we notice that, the filter order is again the most decisive parameter for the behavior of the algorithms. As an example, figure 5.98 shows the evolution of the filter taps, the prediction error and estimation of the pitch angle for HRLS. Observe how the prediction error is low and, therefore, the taps do not oscillate and slowly follow the time-growing bias.

Finally, we have computed the average RMSE again, this time using the optimal parameters obtained from high intensity signals. The RMSE is almost identical for RLS and MNLMS, while for the rest of the methods it worsens between 3° and 6° . We can conclude that the results obtained for the simulations have an acceptable repeatability considering the differences between synthetic and real signals.

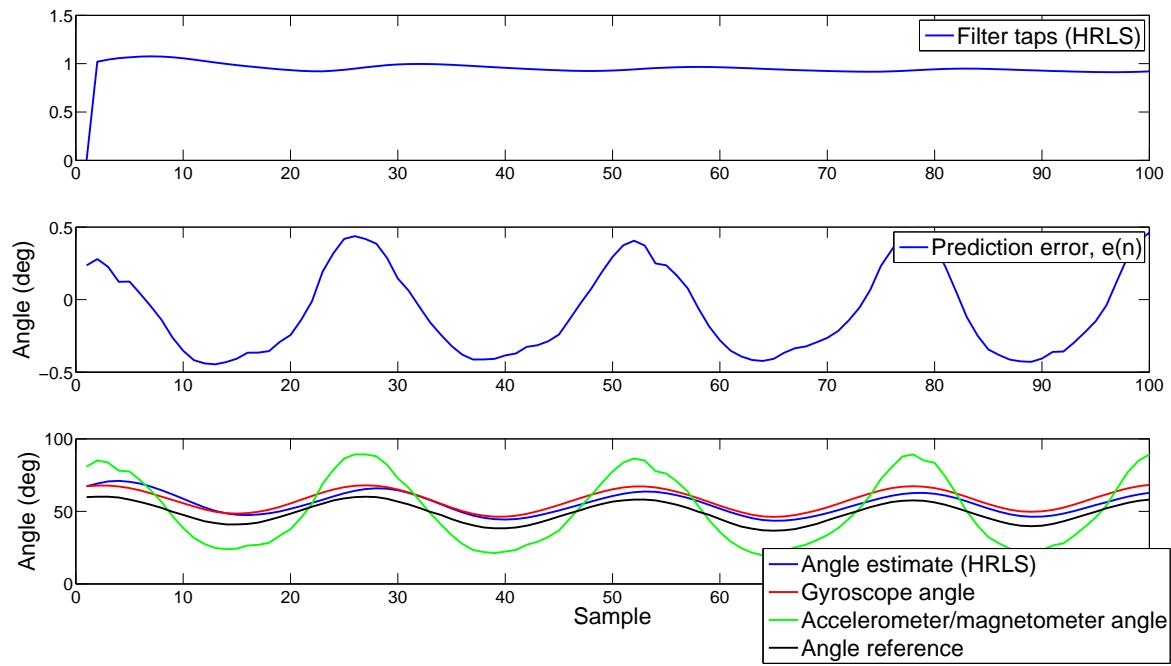


Figure 5.98: Filter taps vs. prediction error vs. angle estimate. HRLS over high intensity real signals using optimal forgetting factor ($\lambda = 0.9$).

5.4.2.2 Low intensity

As it has been commented many times through this chapter, under low intensity conditions, the gravity predominates in the measured acceleration. Therefore, the orientation estimates are very precise in a magnetically stable environment using only the accelerometer-magnetometer approach. However, it is still interesting to apply sensor fusion to filter out noise and avoid possible upcoming magnetic perturbations.

Since no large peaks derived from linear acceleration have to be filtered out, the filters only have to compensate for dynamic bias. Consequently, the average RMSE is around 2° for all algorithms. This time, Kalman filter outperforms the rest of the filters achieving an average RMSE of 1.5430 ± 0.5327 . We can observe how the α parameter—which multiplies the variance of the measurement noise—has decreased from 4.633 to 0.0915. This happens because the variance of the measurement noise has actually decreased, given that we consider as measurement noise the erroneous peaks caused by linear acceleration. This way, decreasing the value of Q we tell the filter to rely more on the observation of the accelerometer-magnetometer angle signal.

Figure 5.99 shows the Kalman gain for the angle estimation, together with the estimation error and the pitch estimate. Notice how the gain converges to zero once the angle estimates are corrected in bias.

The repeatability and generality of synthetic parameters has been also checked by applying the optimal theoretical parameters on real low intensity signals. Excepting QRDRRLS and MNLLMS, the rest of algorithms present an error around only 2° larger.

By now, NLMS and Kalman filter lead the performance comparison. The next step is to check their ability to adapt themselves to signals of changing intensity.

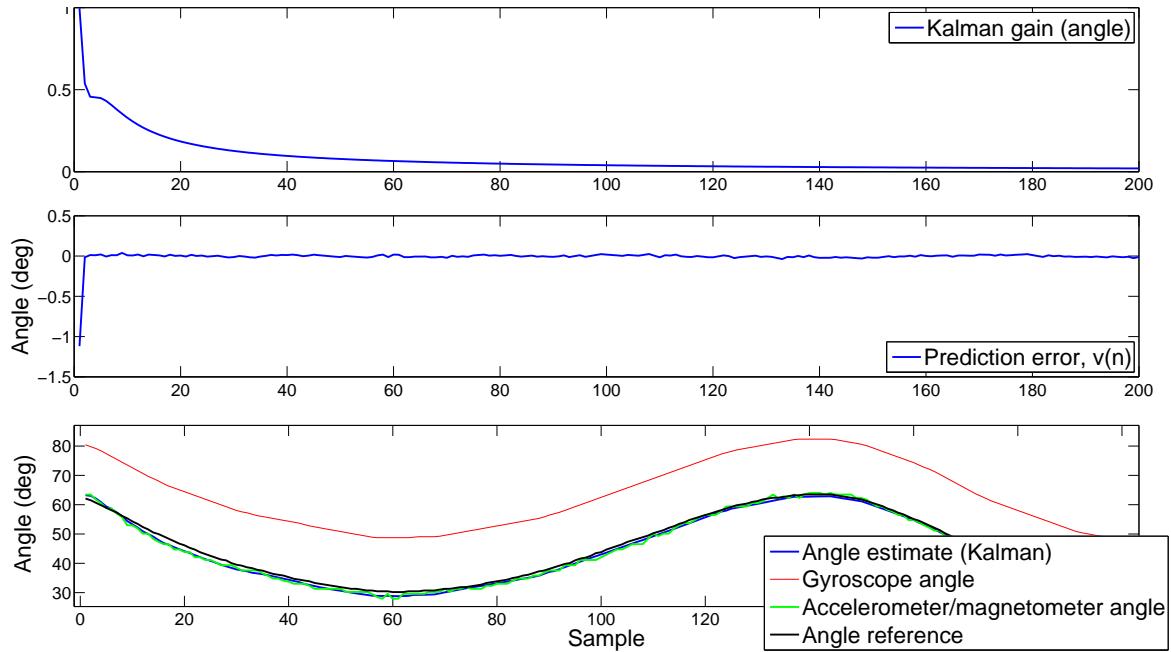


Figure 5.99: Kalman gains vs. prediction error vs. angle estimate. Kalman filter over low intensity real signals using optimal parameters.

5.4.2.3 Low and high intensity

Performance of all algorithms is summarized in tables 5.11 and 5.12. In this tables we show the optimal parameters for two different optimization procedures, one based on grid search and the other one based on the Gauss-Newton minimization algorithm. Gauss-Newton procedure is much faster but in some occasions it can be very sensitive to initial parameters and, if they are far from the optimal value, the minimization process may not converge. Moreover, it also occasionally stops in local minimums which are not close to the optimal value. By inspecting the table we can see that in some occasions it leads to better optimal parameters, but it is generally outperformed by grid search.

The pitch estimates achieve a rather good RMSE for all algorithms and again Kalman filter shows the lowest error. Yaw estimates show a higher RMSE than pitch estimates. This may be caused by two reasons: the intensity of the motion being higher when the signals were gathered and/or the distortion of magnetic disturbances. Most proprietary algorithms obtain a RMSE two to three degrees higher for yaw than for pitch and roll. Kalman filter shows its power achieving an error of just 3° using fix parameters.

We end the discussion of results analyzing the results coming from the use of an intensity detector to reset the integrated angular rate signal—for NLMS, MNLMS, RLS, HRLS and QRDRRLS—and to change the parameters of the filter—Kalman filter.

Figure 5.83 shows the output of the MBCUSUM detector employed to detect the intensity of the motion. The acceleration magnitude is also depicted as it is usually employed as an intensity detector. It is easy to see that MBCUSUM outperforms the magnitude detector as its output crosses the threshold very close to the actual points where the intensity changes. The magnitude detector erroneously crosses the threshold many times which leads to a high rate of false positives. The binary marker shown in the figure will be used to reset the gyro signal or change the filter parameters.

Table 5.15 shows the RMSE for the same yaw estimation example shown before. We observe that the gyro resetting strategy has not decreased the RMSE but it has actually augmented it.

All the tested filters have shown to be rather good at compensating the time-growing bias as it has low frequency nature and all filters have a low-pass behavior. On the other hand, they have more trouble when filtering out abrupt changes, which may be modeled as high frequency disruptions. Therefore, when we reset the gyro signal, we introduce a sudden change in the amplitude of the signal which affects the behavior of the filters as they have to start converging again. In conclusion, the gyro angle resetting approach has shown not to be a good strategy to follow. However, the dynamic change of parameters, which is usually referred to as *gating*, on the Kalman filter has shown better performance than using fix parameters. The gating procedure consists of setting a low measurement noise variance (R_k) when smooth motion is detected, and on the other hand, increasing it by at least an order of 100 when the detector senses intense motion.

For our example, the optimal parameters are $\alpha_{\text{low}} = 1.1450$ and $\alpha_{\text{high}} = 168.82$.

Future research will be oriented to apply the gating procedure to analogously change the parameters of the rest of the tested algorithms. The optimal parameters observed for low intensity signals and high intensity signals will be used as a starting point of upcoming experiments.

5.5 Chapter wrap-up

We started the chapter explaining a series of basic concepts which are necessary for attitude estimation. We explained the different coordinate frames and the different ways to represent the orientation of a body with respect to a reference frame. A brief explanation of Euler angles, rotation matrices, direct cosine matrices and quaternions was given together with their advantages and inherent problems.

Next, we introduced the basics of the two separate ways that can be used to estimate orientation: the projection of the gravity vector and the Earth's magnetic field in the sensor axes and the integration of the angular rate. We put special emphasis on describing the pros and cons of these two approaches to then describe how they can be combined using a sensor fusion approach.

After the introduction to the basics of attitude estimation, we presented a brief analysis of the state of the art of the most well-known attitude estimation methods. This was followed by a detailed explanation of the projection and integration methods.

Afterward, we presented the theoretical principles of some adaptive filters like the Kalman filter, NLMS, MNLMS, RLS, HRLS and QRDRRLS, and how to implement them to fuse the projection and integration methods.

In this point, we continued the chapter presenting the experiments that we carried out to study the performance of all the aforementioned experiments. The experiments were divided in two parts: theoretical simulations and application of the algorithms on real signals.

Theoretical experiments were based on a Monte Carlo simulation that generated synthetic signals at every iteration which were used to optimize the parameters of each algorithm and find the minimum RMSE. At the end of the simulation the results were averaged and presented in tables and figures. Three different types of signals were employed: low intensity signals mimicking low intensity motion, high intensity signals and signals alternating low with high intensity. The optimal parameters were found for each one of the situations.

The next step was to verify the theoretical results using real signals. To do so, we gathered a set of real signals putting a MIMU in a mechanical angle reference device and we ran again the optimizers. We found some differences, but in general most theoretical results were reproducible using real signals. In conclusion, Kalman filter and NLMS were the ones that showed better performance.

We ended the experiments applying an intensity detector to dynamically adjust the parameters of the Kalman filter to the intensity of the motion. This strategy improved the minimum RMSE. In addition, we also reset the integrated angular rate signal at the beginning of each low intensity period but this strategy was not successful.

After the experiments we analyzed the results in detail and set the basis for future research which will be focused on applying the gating procedure to change the parameters of the NLMS and RLS filters and also to change the algorithm on the fly for a better estimation of the orientation angles.

A previous version of the experiments and results similar to those presented along this chapter is available in the following publication:

- A. Olivares, J.M. Górriz, J. Ramírez, G. Olivares, 'Accurate human limb angle measurement: sensor fusion through Kalman, least mean squares and recursive least-squares adaptive filtering'. *Meas. Sci. Technol.* 2011, 22, doi:10.1088/0957-0233/22/2/025801.

Applications of the System

After explaining and developing in a detailed way a complete motion monitoring system using MEMS MARG sensors, we proceed now to briefly explain some of the possible applications of our system.

Without a doubt, the two main fields of applications of the system are inertial navigation and human body motion monitoring.

Inertial navigation can be defined as the use of MARG sensors to determine the trajectory and/or position of a moving body with respect to a reference system. It is frequently combined with the Global Positioning System [171–175] to increase the precision of the system and allow positioning in GPS shadow areas. In addition, inertial navigation is used in military systems [176, 177], aerospace systems [178, 179], aircraft systems [180–182], indoor positioning [183–186], among others.

Thousands of works related to human body motion monitoring have been published during the last years. The great majority of these works is connected with Health Science and Sport Science.

There exists a large amount of diseases that distort the motor control of human body or present symptoms that can be identified by the analysis of human body posture and motion. Along this chapter, we will briefly comment how our system has been applied, is being applied or can be applied to telerehabilitation, analysis of Activities of Daily Life (ADL), detection of tonic-clonic epileptic seizures, episodes of bradykinesia and dyskinesia of Parkinson patients, and the study of sleep disorders.

On the other hand, as we have commented, motion monitoring systems are being intensively used in the sports field to analyze and improve the technique and performance of top athletes. MEMS MARG systems are of great help to monitor

motion in sports as the sensors can be easily attached to the body of the athletes using no wiring at all so they can carry out their routines with almost no restrictions derived from the monitoring equipment.

After this short introduction, we develop now all the aforementioned topics related to human body motion as this was the initial motivation of our system, and therefore, it is the field to which we have applied our system.

6.1 Telerehabilitation

Most surgical procedures in patients suffering from diverse trauma in the limbs entail rehabilitation programs with a duration oscillating between several months and a few years [187]. The rehabilitation programs usually start with a phase where the patient gets treatment from physiotherapists including massages, infrared heat, short wave stimulation, etc. During this first phase the patient has to move unavoidably to the medical center. When this first treatment phase is complete the patient gets into a phase where a series of low intensity exercises have to be performed to strengthen the muscles and rehabilitate the affected zone [188], [189]. This kind of exercises do not require the continuous presence of supervising staff since they are usually simple to carry out and can be easily done by the patient at home once they have been told how to proceed. Doctors can then supervise the sessions carried out by the patients by remotely checking the logs.

Due to the overcrowding of rehabilitation centers there is a long waiting list to start most of the treatments. When the patient is called out, he is usually scheduled during working hours.

As a consequence of the overcrowding, patients do not receive a complete treatment and are usually sent home prematurely where they perform the exercises under no supervision.

Telerehabilitation improves the efficiency and shortens the duration of the treatment since it allows the patient to remotely carry out the second phase without having to renounce to medical supervision.

The telerehabilitation system needs instrumentation to monitor the patient, algorithms to process the gathered data, a human-machine interface where the patient can observe in real time the execution of the exercises through a 3d avatar, and a patient-doctor communication platform that allows the latter to supervise the rehabilitation process, check the proper execution of the exercises, update the rehabilitation plan and schedule new sessions and exercises.

Wagyro (the first version of our prototype) has been used during the last two years in Telefonica's I+D Rehabitic project. Rehabitic is a knee telerehabilitation ongoing project that uses Wagyro hardware and offers real time 3D animation of the movements a patient is performing. The patient can select between different

exercises. The duration, intensity and frequency of occurrence of every exercise is advised and adjusted by the therapist, who has access to the recorded data through a web platform. The system is also able to ask questions to the patient, for example: How much pain do you feel today? This way the therapist gets a better understanding of how the patient is feeling. Also the therapist is able to give feedback to the patient. Rehabitic is used in the Esperanza and Clinic Hospitals in Barcelona, Son Llátzer Hospital in Palma de Mallorca, Virgen de las Nieves Hospital in Granada and Torrevieja Hospital in Alicante.

Wagyromag will be used in a second phase of the project where other parts of the body will be included, such as the joints of the arms or the back. The more complex kinematics of the back needed from a sensor with a higher degree of freedom which Wagyromag can provide.

Other works related to the application of MARG systems to telerehabilitation can be found in [190–198].

6.2 Gait and Posture analysis

Gait and posture analysis are of utmost importance in prevention and diagnosis of osseus, muscular, degenerative and nervous injuries and diseases. Many studies have been carried out [199], [200] to try to determine when a patient is more likely to fall due to their faulty gait and posture patterns, as well as other studies that focus on detection of limbs injuries based on gait. Most of these studies use human body position monitoring systems that are based on video or inertial sensors. Inertial sensors allow the study to be extended to different scenarios, as the patient can be monitored wherever he is as long as he is wearing the MIMU. Video systems are limited to a closed scenario which is surrounded by cameras. Such video systems offer better accuracy but their price is many times higher than inertial sensor systems.

The use of neural networks and classification methods permits the system to detect movement patterns that can be identified with several diseases.

Other works related to the application of MARG systems to the analysis of Posture and Gait can be found in [87, 201–216]

6.3 Analysis of activities of daily life

Nowadays, most elderly people have different teleassistance and emergency systems installed at home. These systems are based on wireless devices known as 'panic

buttons'. This kind of buttons are usually worn by the patients in a necklace or in a bracelet, so when they are in an emergency situation, such suffering a dangerous fall, a heart attack or other kind of discomfort, they can push it and an operator from the emergency call center contacts them by telephone. If the person is not responding, medical help is sent immediately.

This kind of systems are useless when the person falls unconscious and is unable to push the button. As a consequence, inertial systems are starting to be embedded on panic buttons so falls and long inactivity periods can be detected. Likewise, it is very interesting for the relatives of elderly people to track and control their daily activity. Due to the reticence of elderly people to have cameras installed at home so their relatives can periodically observe their activity to check how are they doing, inertial systems can be employed to monitor their basic ADL such as walking, sitting down and standing up, lying down, etc. This kind of systems are able to alert from abnormal activities during their routine, such as periods of inactivity, falls, sleep habits, etc. Researchers are also working to develop mobile teleassistance so care systems' range is extended, offering assistance while the person is out of home.

Therefore, it is very helpful and quite straightforward to use human body position monitoring systems in complete and reliable assistance systems, since inertial sensors can be carried by the person in a necklace or in a bracelet.

We used a set of Wagyromags to monitor two young adults while they were performing a circuit composed of different ADL: opening and closing a door, a window and its blinds, walking, stair climbing and descending, sweeping the floor, filling the dustpan after sweeping the floor, washing the floor, wringing the mop, cleaning the dust on a table, cleaning a stain on a table, brushing teeth, writing a letter, typing a keyboard, sitting down and standing up, vacuuming the floor, standing still, sitting still, lying still, jumping, bending down and ducking flexing the knees. To this purpose, we used a Matlab GUI application which sends a synchronization frame to the Wagyromags placed on the body of the subject and starts the data gathering process. Once the Wagyromags are activated and gathering data, the subject starts to perform the ADL under the supervision of an operator who uses the same Matlab application to mark the starting and ending times of each separated activity. In addition, another Matlab application uses the activity log to divide the complete signal into subsets corresponding to each activity.

Figures 6.1 and 6.2 show the raw acceleration gathered while closing the blinds of a window and cleaning the dust on a table, respectively.

Other works related to the application of MARG systems to the analysis of ADL can be found in [190, 217–226].

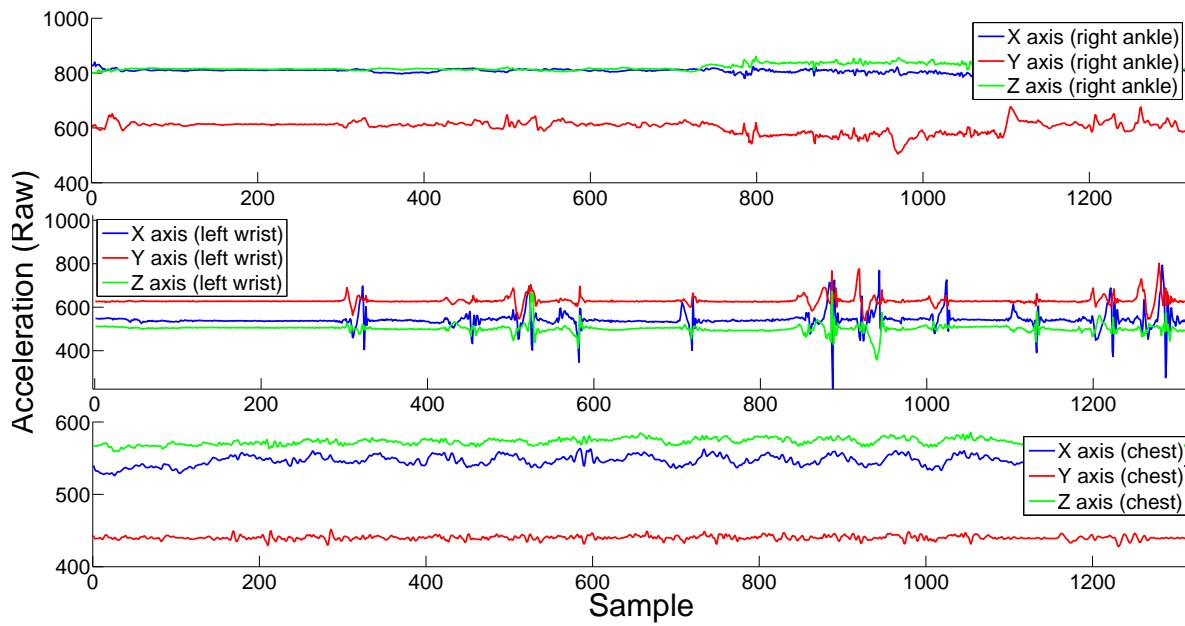


Figure 6.1: Raw acceleration gathered while closing the blinds of a window.

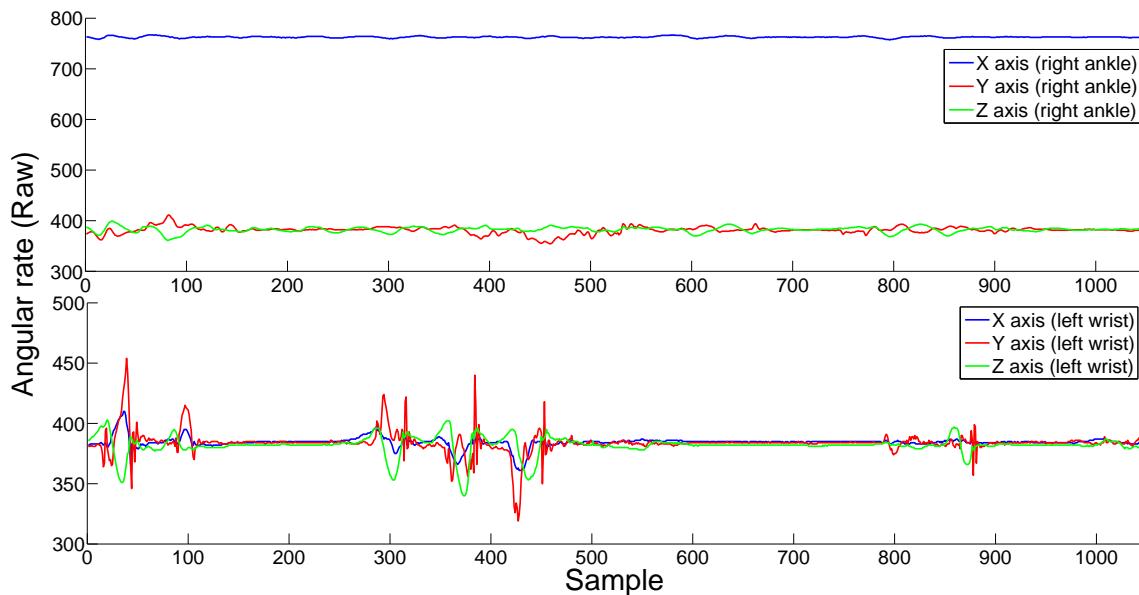


Figure 6.2: Raw acceleration gathered while cleaning the dust on a table.

6.4 Fall detection

Fall detection is a subfield of the analysis of ADL, but, since it has become a very wide and productive field of study, we are presenting it separately from the ADL section.

A critical moment during the daily life of elderly people living alone, or left home alone during certain periods of time, takes place when they suffer a fall. Shocking statistics taken from [227] reveal that:

- About one third of the elder population over the age of 65 falls each year, and the risk of falls increases proportionately with age. At 80 years, over half of seniors fall annually.
- Those who fall are two to three times more likely to fall again.
- About half (53%) of the older adults who are discharged for fall-related hip fractures will experience another fall within six months.
- Falls are the leading cause of death due to injury among the elderly. 87% of all fractures in the elderly are due to falls.
- Falls account for 25% of all hospital admissions, and 40% of all nursing home admissions. 40% of those admitted do not return to independent living; 25% die within a year.
- Many falls do not result in injuries, yet a large percentage of non-injured fallers (47%) cannot get up without assistance.
- For the elderly who fall and are unable to get up on their own, the period of time spent immobile often affects their health outcome. Muscle cell breakdown starts to occur within 30-60 minutes of compression due to falling. Dehydration, pressure sores, hypothermia, and pneumonia are other complications that may result.
- Getting help after an immobilizing fall improves the chance of survival by 80% and increases the likelihood of a return to independent living.
- Up to 40% of people who have a stroke have a serious fall within the next year.

From this statistics, it is very clear that falls are a serious health issue and that systems automatically detecting a fall and calling for help could be of great help to solve it. By attaching a Wagyromag to, for example, the belt or the pocket of a person, or processing the orientation and the intensity of the motion of the subject using our algorithms in a smartphone, we could be able to detect anomalous acceleration profiles, as well as the orientation of the body before and after the alleged

fall. In addition, we could also detect if the person has gotten up after the fall or is lying on the floor.

Following this line of research, we are currently developing new fall detection algorithms and carrying out a comparative study between the most popular published methods. An example of the acceleration profiles gathered during a front fall, placing a Wagyromag on the back of a subject, are shown in figure 6.3.

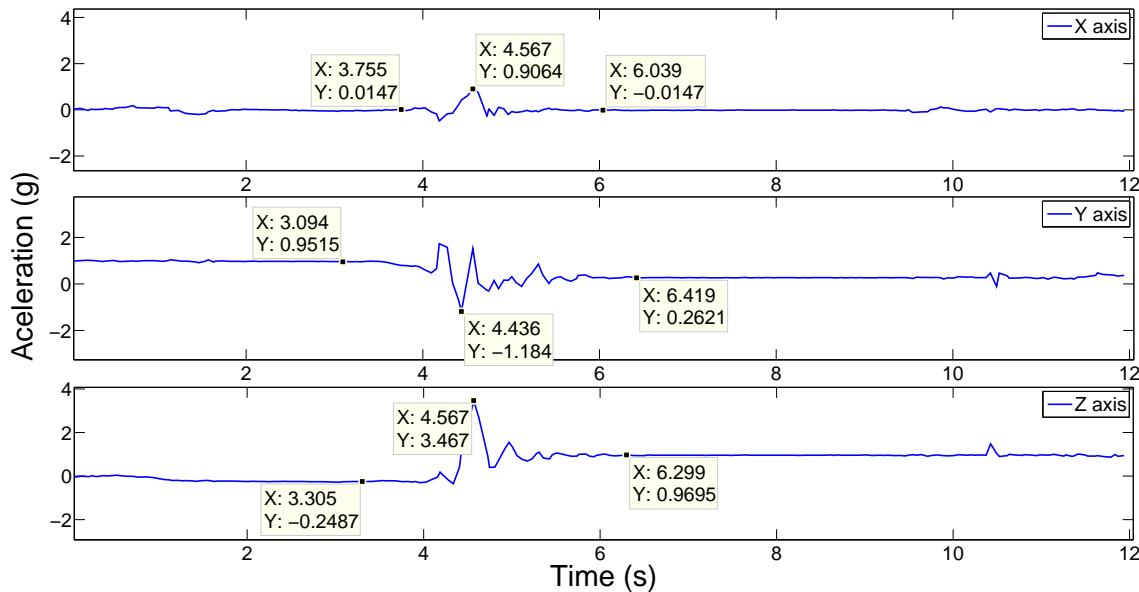


Figure 6.3: Raw acceleration gathered during a front fall.

6.5 Detection of seizures

As we said in the introduction, some diseases cause episodes of uncontrolled motion of the patients body. Detecting them using a MIMU attached to the body of the patient may be of great importance, specially under the event of a violent seizure. In this section, we will talk about the possibilities of applying our system to the detection of tonic-clonic epileptic seizures and bradykinesia and dyskinesia episodes in Parkinson's patients.

6.5.1 Epilepsy

According to the American Epilepsy Foundation [228], "Epilepsy is a medical condition that produces seizures affecting a variety of mental and physical functions. It is also called a seizure disorder. When a person has two or more unprovoked seizures, they

are considered to have epilepsy. Seizures happen when clusters of nerve cells in the brain signal abnormally, which may briefly alter a person's consciousness, movements or actions". Some facts revealing the magnitude of Epilepsy, extracted from [229–232] are listed below:

- Epilepsy affects over 3 million Americans, 400,000 Spaniards and 50 million people worldwide.
- One in 100 people will develop Epilepsy.
- One in 10 people will suffer a seizure in their lifetime.
- Each year over 125,000 to 150,000 are newly diagnosed with Epilepsy in the USA and 20,000 in Spain.
- In two-thirds of patients diagnosed with Epilepsy, the cause is unknown.
- In over 30% of patients, seizures cannot be controlled with treatment.
- Uncontrolled seizures may lead to brain damage and death.
- The mortality rate among people with Epilepsy is two to three times higher than the general population.
- Risk of sudden death among those with Epilepsy is twenty-four times greater.
- Of people with epilepsy, around 20% to 25% have tonic-clonic seizures.

Some of the patients suffer tonic-clonic seizures while they are asleep. These seizures are known as tonic-clonic nocturnal seizures.

Epilepsy patients usually keep a register of the suffered attacks, so they can hand it to their doctor during control and medication revision sessions. However, nocturnal seizures may go unnoticed to the patient, thus, not being annotated in the register. It may happen that a patient suffers many nocturnal seizures and few or no seizures during the day. As a consequence, during the control sessions with the doctor the patient would tell the doctor the medication is working properly, when it actually does not. To avoid such a situation, the patient may sleep with an attached MIMU so nocturnal seizures can be detected and stored in the memory, allowing the doctor to notice that nocturnal attacks are happening. Hence, MIMUs may be an useful tool to improve the monitoring and detection of epileptic crisis.

A pilot project will soon be started in collaboration with the Epilepsy Unit of the Virgen de las Nieves Rehabilitation and Traumatology Hospital. The goal is to monitor patients having nocturnal epileptic seizures in order to identify their frequency and duration, so the patient can have a log including them, which he can

hand in to the doctor.

Some preliminary works have already been carried out aiming to detect nocturnal epileptic seizures using MARG sensors. Examples of them are [233–238].

6.5.2 Parkinson's

According to [239], Parkinson's disease is a progressive, neurodegenerative disease that occurs when the neurons within the brain responsible for producing the chemical dopamine become impaired or die. Dopamine is essential for the smooth control and coordination of the movement of voluntary muscle groups. Once approximately 80% of the brain's dopamine producing cells no longer function, the symptoms of Parkinson's disease begin to appear. Parkinson's disease may be termed as a progressive movement disorder that is distinguished by marked slow movements, tremors, and unstable posture.

Some statistics [240] on Parkinson's are listed below:

- As many as one million Americans and 60,000 Spaniards and 10 million people worldwide live with Parkinson's disease.
- Men are one and a half times more likely to have Parkinson's than women.
- The combined direct and indirect cost of Parkinson's, including treatment, social security payments and lost income from inability to work, is estimated to be nearly \$25 billion per year in the United States alone.
- There is a large number of cases that go undetected.
- It is estimated that 1.10% of the U. S. population, or 1 in 90 people, have the disease and do not yet know it.
- Only two studies have investigated the accuracy of clinical diagnosis and estimated the accuracy of diagnosis to be somewhere around 80%.

Therefore, Inertial systems could also be used to detect tremor [241], bradykinesia [242] and dyskinesia [243] episodes of Parkinson's disease patients and help to diagnose and monitor them. Bradykinesia is the term for defining slow execution of the movement, tremor is the term used to define repetitive periodic movements within a certain frequency range and dyskinesia is a movement disorder which consists of effects including diminished voluntary movements and the presence of involuntary movements. Since all these disorders are related to motor activities, they can be tracked and detected by a MIMU.

Examples of works applying MARG sensors to monitor Parkinson's disease patients can be found in [244–260].

6.6 Sleep disorders

Sleep disorders can also be monitored by inertial sensors. From time to time patients suffering from sleep disorders have to spend the night at sleep units located in hospitals so their heart rate, brain and motor activity are monitored. This test is known as polysomnography. Actigraphy, the use of actigraphs (simple IMUs) to monitor sleep cycles, is a good alternative to polysomnography as the patients can move freely while they are being monitored. A low cost MIMU would allow patients to be monitored at home during periods between control sessions at the hospital. By monitoring them at home, the cost of the study is reduced and sleep specialists can follow the evolution of the patient daily and offer a better medical treatment. There exist some commercial devices such as Fitbit [22] and other actigraphs [261, 262] that measure and monitor activity during sleep periods and measure the quality of the rest. Wagyromag could also be used to this purpose together with the application of our orientation estimation algorithms to extract useful data to monitor sleep cycles and their quality and improve the precision and features of the already existing devices.

Examples of works applying MARG sensors to monitor sleep quality can be found in [263, 264].

6.7 Sport science

The use of MARG sensors to study and monitor human body motion and position in sport has become very popular during the last few years. A wireless motion monitoring system based on MARG sensors like ours can be applied to track and visualize a golf swing, a football, a tennis racket or a swimming stroke. Therefore, with this technology, it is possible to optimize the performance of the athlete through the analysis of the technique.

As described in [265] such systems can also be used for judging in a sports competition for example a snowboarding competition, which gives the judges the opportunity to focus on skills they normally are not able to focus on. Nowadays the judges are busy focusing on the height, turns or the amount of loops, which could be measured by a wireless measurement unit. So it might be possible to develop an automatic scoring system that uses the data recorded by the wireless measurement unit.

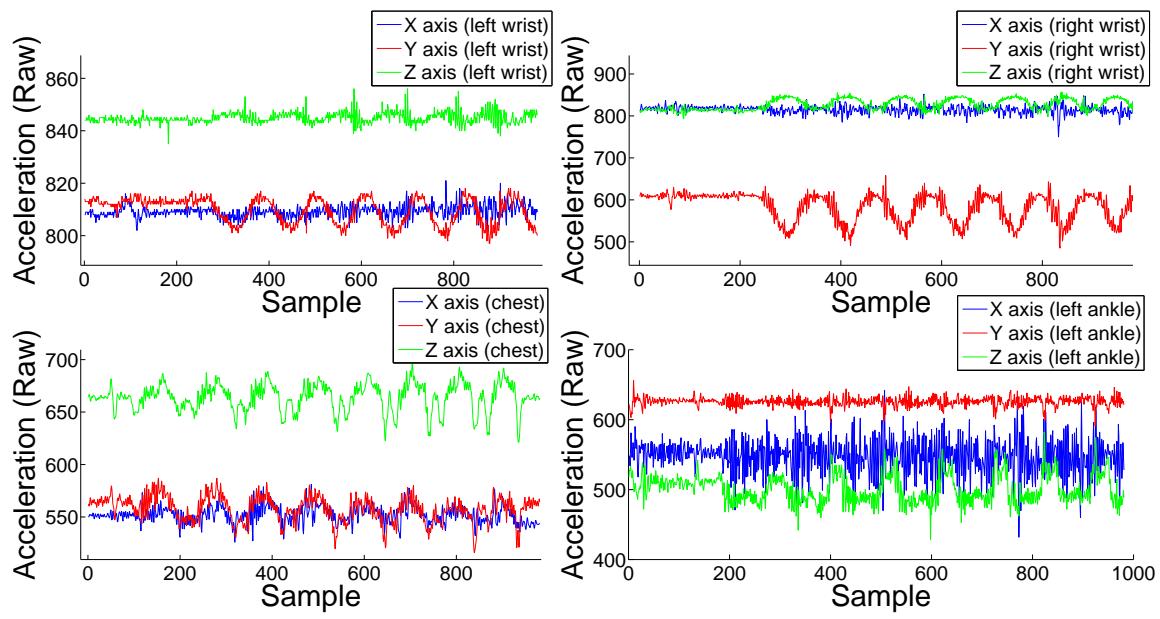


Figure 6.4: Raw acceleration gathered while performing a set of push ups. Four MIMUs are placed in both wrists, chest and left ankle.

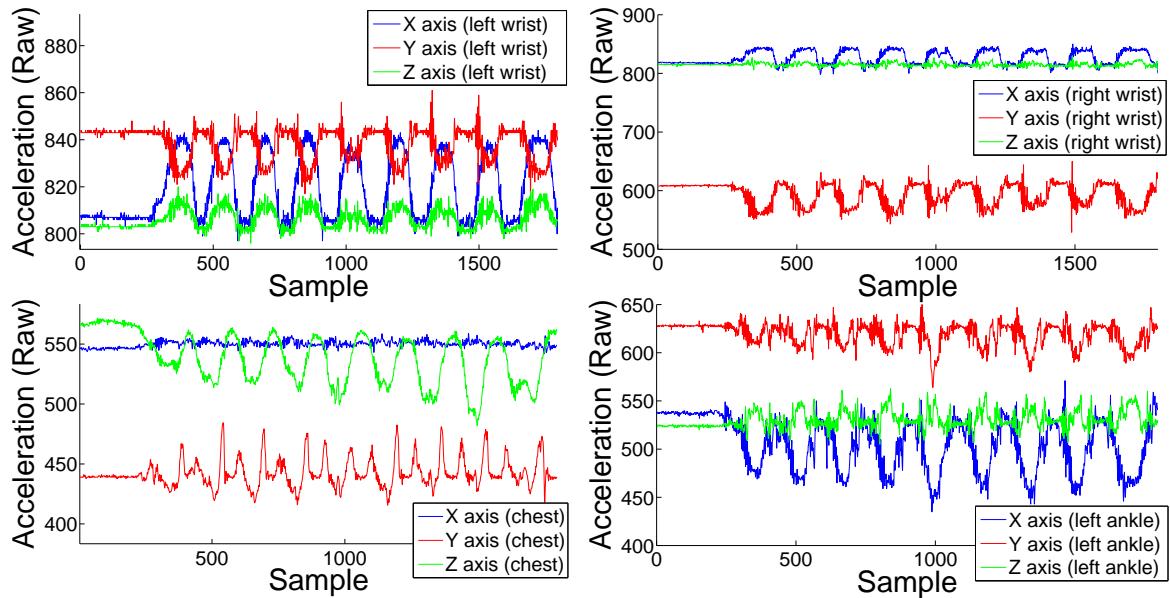


Figure 6.5: Raw acceleration gathered while performing a set of squats. Four MIMUs are placed in both wrists, chest and left ankle.

A complete analysis of the application of inertial sensors in sport science can be found in [266]. It includes measurements of the acceleration of a golf player's hip, wrist and the golf swing.

Figures 6.4 and 6.5 show the gathered raw acceleration while a subject was performing a set of push ups and squats, respectively. Three Wagyromags were placed on both wrists and the left ankle. A fourth IMU—an Equivital unit [15]—was placed on the chest. Notice how the repetitions, as well as their frequency and intensity can be easily observed at a glance.

Examples of works applying MARG sensors to monitor human body motion in sports can be found in [267–299].

6.8 Chapter wrap-up

Along this chapter we have presented the most popular applications of orientation monitoring systems like the one we have developed. We have made special emphasis on human body motion and position monitoring since this was the main motivation that originated the current project.

We have also commented how Wagyromag can be used in conjunction to the developed algorithms to monitor patients suffering from diseases, such as Epilepsy and Parkinson, that affect the normal motion of their bodies. Monitoring patients out of the hospital during their daily life increases their quality of life at the same time it saves many government resources.

On the other hand, we have explained how MIMUs can be applied to monitor body motion in sports and build reports detailing the performance and technique of athletes. Commercial applications in this field are also possible. Inertial sensors could be used to monitor the movement of the body during an exercise session and a system could act as a personal trainer. Such system could be installed in gyms or at home. We will work together with experts of Sport Science to develop to adapt our system to their needs.

There is an enormous field of applications of human body monitoring systems. In order to find possibilities of application is always good to talk to experts of different fields, such as doctors. Engineers lack of knowledge to find applications of human body monitoring in health applications, so such interaction with medicine experts is more than recommended.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The analysis of human body motion and posture has gained special relevance within the past years in many medicine fields, as well as in teleassistance applications for the handicapped and analysis of autonomous sport activities among other disciplines. Using low cost devices based on magnetic and inertial MEMS microsensors to monitor human body motion is only possible through the adequate processing of the signals provided by these sensors, in order to achieve the highest degree of precision for real time orientation measurements of different body segments.

In a nutshell, the ultimate goal is to increase the performance of the system by fusing the signals provided by the accelerometers, gyroscopes and magnetometers given that they are corrupted with error characteristics specific to MEMS sensors (constant bias, thermo-mechanical white noise, angle random walk, flicker noise, bias instability, etc.).

After studying different sensor modeling approaches, which include the main calibration parameters, and describing the state of the art of the current trends on calibration algorithms, we dealt with accelerometer and magnetometer calibration. We selected twelve of the most significative and popular algorithms to carry out a preliminary experimental comparative study. Among all them, those that showed the best performance and accuracy were the algorithm presented by Ferraris et al. in [32] for the accelerometer, with a total average error of 0.0092 ± 0.0054 g with respect to the reference (the gravity vector), and those presented in [42, 45, 132] for the magnetometer, with an average error of $0.0030 \pm 6.87e-4$ Gauss in the measured magnitude of Earth's magnetic field with respect to the known local value

of Earth's magnetic field in Granada, Spain. Then, we studied the possibility of calibrating gyroscopes without using expensive and complex multi-axis rate tables, and we proposed two new algorithms to calibrate gyroscopes which use just a simple bicycle wheel. The spin velocity of the wheel is estimated using the gathered magnetometer and accelerometer signals. These signals have a sinusoidal nature in which every period corresponds to a wheel turn. This way, the angular rate at every time instant is estimated and paired with the raw gyroscope data so a calibration line is built for every axis.

In addition to these two methods, we also proposed an algorithm that enables the estimation of the calibration algorithms only when the input data is uniformly distributed in space. To this purpose, the algorithm computes the histogram of θ and ϕ spherical coordinates for a determined number of bins. Thus, counting the number of empty bins and comparing it to a minimum tolerated value, the algorithm will enable, or not, the process to estimate the calibration parameters that is included in most existing calibration procedures.

Determining the intensity of motion is a very important part of the system since many calibration algorithms need to recompute their parameters during the periods in which the object or the subject being monitored is static. Additionally, some attitude estimation algorithms modify their parameters dynamically depending on the intensity of the motion to increase their precision. Therefore, we analyzed the state of the art of intensity detection algorithms present in the literature and we included in this analysis a couple of Voice Activity Detection (VAD) algorithms used to distinguish voice from silence which we intuited that could show good performance when applied to MARG signals. In addition to the VAD algorithms, we also decided to use two algorithms employed in industrial control to detect abrupt changes in signals.

In order to determine the accuracy of each algorithm, we carried out a comparative study between all the algorithms using both synthesized and real signals. We found out that FSD (Framed Spectrum Detector) and LTSD (Long Term Spectral Detector) algorithms outperformed the precision of the existing algorithms used as the baseline, reaching 97% of accuracy when distinguishing between motion and inactivity.

Then, we focused our work on the estimation of the orientation of a body in space. We explained the diverse reference frames used in inertial navigation and presented the most popular ways of describing the orientation with respect to them. We defined the Euler angles, the rotation matrices, the Direction Cosine Matrix (DCM) and quaternions, to subsequently introduce the different existing strategies to estimate the orientation of a body in space using MARG signals. We saw how the measured acceleration and magnetic field can be compared to the known reference vectors (Earth's gravitational and magnetic fields) to extract the angles between them, and we also explained how to integrate the angular rate to compute

the displacement angle. After analyzing these two separate strategies and commenting their pros and cons, we explained how to fuse them applying adaptive filters to achieve a higher degree of precision. Among adaptive filters, we selected the Kalman filter since it has been widely used in attitude estimation and inertial navigation applications, and some variants of the LMS (Least Mean Squares) and RLS (Recursive Least Squares) filters. We detailed step by step the theoretical basis of each filter and their fusion schemes. We also proposed a new alternative to apply the Kalman filter consisting in dynamically adjusting its parameters to the degree of intensity of the motion. The intensity of the motion is determined by applying one of the algorithms explained in chapter 4. More specifically, they are applied to distinguish the inactivity/low intensity periods from the high intensity ones. Thus, during low intensity periods, the parameters of the Kalman filter are adjusted to give more reliability to the orientation computed using the accelerometer, and, on the other hand, during high intensity periods, they are adjusted to weigh more the orientation estimate given by the integration of the angular rate.

Afterward, we carried out a comparative study between the filters. To do so we computed the RMSE between the real orientation of the MIMU with the estimated orientation provided by each algorithm, and compared the results. We also built a signal synthesizer that mimics both angle signals computed with the accelerometer/magnetometer and the gyroscope, that we used to carry out a comparative study based on a Monte Carlo simulation.

Among all the tested algorithms, we found out that the dynamic version of the Kalman filter outperformed the rest of the filters when used on motion signals with changing intensity. More specifically, the dynamic Kalman Filter obtained a RMSE of 2.1132° , compared with the 3.5430° error achieved by the static version.

We also observed that the HRLS filter outperformed the precision of the Kalman filter when applied to signals showing only very high intensity ($4.3924 \pm 1.9770^\circ$ versus $5.2274 \pm 0.5976^\circ$).

Our work has a direct application in physical telerehabilitation of after surgery patients, analysis of gait and posture, analysis of activities of daily life, fall detection, detection of tonic-clonic epileptic seizures, monitoring of bradykinesia, dyskinesia and tremor seizures of Parkinson's patients, as well as analysis of sleep disorders and sports activity monitoring oriented to increase the performance of professional athletes or supervise autonomous training.

We have opened new ways for MEMS MARG sensors calibration, activity intensity detection and orientation estimation by proposing new algorithms and carrying three comparative studies which include a selection of the most relevant methods in every field. The results of our work are being used in telerehabilitation programs running in various Spanish hospitals; Esperanza and Clinic Hospitals in Barcelona, Son Llátzer Hospital in Palma de Mallorca, Virgen de las Nieves Hospital in Granada and Torrevieja Hospital in Alicante. Our system will also be used in other applications oriented to detect falls of elderly and/or handicapped persons, analysis of

sport activity and detection of tonic-clonic epileptic seizures (at the Virgen de las Nieves Hospital in Granada), among others.

Finally, we proceed to analyze the initial objectives that were enumerated in the introductory chapter to determine to what degree they have been fulfilled.

- **Calibration:**

- *Initial objective:* Carry out a complete study of the state of the art of calibration algorithms for MEMS magnetometers, accelerometers and gyroscopes.
 - * *Results:* We have analyzed more than 100 papers which describe calibration algorithms for MEMS magnetic and inertial sensors.
- *Initial objective:* Select and implement the most cited algorithms to analyze their accuracy and performance by means of a comparative study.
 - * *Results:* From the 100 analyzed methods we have selected and implemented 12 magnetometer and accelerometer calibration methods. These methods have been compared in a preliminary study that aimed to determine the error of the calibrated physical units (acceleration or magnetic field) with respect to the Earth's gravitational and magnetic field vectors. During the study, we observed the importance of the definition of a complete theoretical model for the sensor output, as well as the good spatial distribution of the input data used to estimate the calibration parameters. To warrant proper spatial distribution of these data, we have implemented and automatic system that computes the distribution by calculating the histogram of the spherical coordinates. Then, the system only allows the estimation of the parameters when the number of empty bins of the histogram is lower than a predefined threshold. In addition, we have extended the use of some algorithms that were designed only for accelerometers or only for magnetometers so that they can be applied to calibrate both sensors.
- *Initial objective:* Propose new calibration procedures not requiring complex instrumentation.
 - * *Results:* We have proposed two new algorithms to calibrate the gyroscope included in IMUs or MIMUs, which do not require multi axis numeric rate tables. These two algorithms only require a rotating device having a large rotation moment, e.g. a bike wheel, and are based on a very simple set of maneuvers that can be carried out by non-specialized operators.

- **Orientation estimation:**

- *Initial objective:* Carry out a complete study of the state of the art of existing algorithms.
 - * *Results:* We have analyzed more than 80 papers which describe orientation estimation algorithms fusing magnetic and inertial sensors using adaptive filters.
- *Initial objective:* Propose new algorithms based on alternative adaptive filtering approaches and compare their performance with a set of the most popular algorithms developed during the last years.
 - * *Results:* We have proposed 5 novel sensor fusion approaches based on the RLS filter and two of its variations, as well as two variations of the LMS filter. Moreover, we have developed in a detailed way the theoretical basis of a fusion scheme based on the standard Kalman Filter. In addition, we have implemented Madgwick's [80] and FQA [71] algorithms and included them in the comparative study. We have also studied the performance of the algorithms for signals derived from large accelerations, low accelerations and both large and low accelerations. We have shown that our Kalman filter model offers the best performance for low acceleration and mixed signals, while the HRLS filter outperforms the rest when estimating the orientation of a body subject to large accelerations.
- *Initial objective:* Propose algorithms that modify their parameters dynamically depending on the intensity of the movement being measured.
 - * *Results:* We have proposed a variation of our algorithm based on the Kalman filter which adapts the model parameters in a dynamical way depending on the degree of intensity of the motion that is being monitored. This novel model decreases the RMSE in 1.4° with respect to the static alternative.
- *Initial objective:* Build a mechanical position reference system to compute the actual orientation of the body to which the output of the attitude estimation algorithms is compared.
 - * *Results:* We have built a device that measures the orientation angles based on a linear potentiometer. We have used this device as a reference to compute the error of the attitude estimations obtained by the algorithms.

In addition to the initial objectives, we have also accomplished other objectives that came out while we were working. In particular,

- *Objective:* Propose new algorithms to detect the intensity of the activity being monitored to, for example, distinguish between periods in which the body being monitored is static or in movement.

- *Results:* We have carried out a comparative study between four already existing detection algorithms and four other algorithms which had been applied in other fields such as voice activity detection and abrupt signal change detection. We have shown that the algorithms based on the analysis of the spectrum of the signal outperform the accuracy of the baseline algorithms. Moreover, we have developed a synthesizer that mimics acceleration and angular rate signals gathered while the subject is carrying out basic activities of daily life. This synthesizer can be very useful for applications of analysis and classification of ADL as it allows testing new algorithms over large sets of signals.
- *Objective:* Modify Wagyromag's firmware to synchronize the beginning of data gathering between multiple units forming a network, so data coming from different Wagyromags placed on different parts of the body is synchronized in time.
 - *Results:* We have developed a new version of the firmware that triggers the data gathering process only when it receives a frame which is sent wirelessly from a Matlab application running on a base station.

7.2 Future work

As we have done for the final conclusions, we proceed to analyze future work for each one of the main parts of our system.

- **Hardware and Instrumentation:** We will design a new version of Wagyromag so it includes the last commercial MEMS sensors available in the market. MEMS sensors are evolving very fast and the last models achieve high precision rates and present less artifacts in the output. Moreover, we will add more processing capability so the orientation estimation algorithms can be run in real time. We will also revise the technology employed to transmit the data and consider the possibility of including new sensors such a microphone to improve the quality of fall detection algorithms by analyzing the sound derived from the impact.
- **Calibration of sensors:** We will develop new calibration algorithms based on the minimization of an error function. We have observed that all existing algorithms minimize the error between the measured magnitude and the reference magnitude without considering the direction of the latter. Therefore, we will try to develop new error functions which also consider the deviation of the measured components with respect to the components of the reference vec-

tor. In addition, we will also continue working on the development of new calibration algorithms which do not require from complex instrumentation.

- **Detection of motion intensity:** We will continue improving the quality of the synthesizer of inertial signals so it includes a larger number of activities in its repertoire. Furthermore, we will work on the development of algorithms able to detect multiple levels of intensity.
- **Orientation estimation:** We will develop new sensor fusion models based on other adaptive filters such as the Extended Kalman filter, the Unscented Kalman filter, particle filters, etc. We will also continue developing the concept of dynamic variation of parameters and we will consider the possibility of alternating between different filters depending of the intensity of the activity being monitored.
- **Applications of the system:** We will soon start the nocturnal tonic-clonic epileptic seizure detection pilot project that is currently in the design phase. We will also continue working in conjunction of a medical team to adapt our system to their needs. We will also work to develop fall detection algorithms as well as in telerehabilitation and teleassistance projects needing multiple measurement nodes placed on different body segments and requiring high precision for the estimation of the dynamic orientation.

Conclusiones y trabajo futuro

El análisis del movimiento y la posición corporal está siendo en los últimos años de especial relevancia en algunos campos de la medicina, en aplicaciones de teleasistencia para personas dependientes y en la monitorización de actividades deportivas autónomas, entre otras disciplinas. El uso de dispositivos corporales de bajo coste, construidos mediante microsensores MEMS magnéticos e iniciales para la monitorización del movimiento del cuerpo humano sólo es posible mediante el adecuado procesamiento de las señales suministradas por dichos sensores, a objeto de conseguir la mayor precisión posible en la medida en tiempo real de las trayectorias de los diferentes segmentos corporales.

Se trata, en definitiva, de conseguir las mejores prestaciones, fusionando las señales de acelerómetros, giróscopos y magnetómetros a pesar de las características de error propias de los sensores MEMS (desviaciones, ruido blanco termomecánico, deriva aleatoria del ángulo, ruido 1/f, inestabilidad térmica de la desviación, etc).

Tras el estudio del modelado de los sensores, incluidos los principales parámetros de medida de los efectos no deseados y tras el estudio del estado del arte de los distintos métodos, el primer aspecto abordado fue la calibración de acelerómetros y magnetómetros. A continuación seleccionamos los doce más significativos, para posteriormente llevar a cabo un estudio comparativo experimental preliminar. De todos ellos, los que mostraron un mejor comportamiento y precisión son el método presentado por Ferraris [32] para acelerómetros, con un error medio total de 0.0092 ± 0.0054 g respecto a la referencia (el vector gravitatorio terrestre), y los presentados por Frosio et al. Skog et al. y Wang et al. [42, 45, 132] para magnetómetros, con un error de $0.0030 \pm 6.87E^{-4}$ Gauss en la magnitud del campo magnético terrestre, respecto al valor local conocido de la magnitud del campo magnético terrestre en Granada. Seguidamente estudiamos la posibilidad de calibrar giróscopos sin utilizar costosas mesas rotatorias multieje, y propusimos dos nuevos algoritmos de calibración para giróscopos de forma que son acoplados a una rueda, cuya velocidad de giro es estimada analizando las señales de un acelerómetro o alternativamente

de un magnetómetro. Dichas señales presentan una naturaleza sinusoidal, coincidiendo cada periodo con un giro completo de la rueda. De esta manera, la velocidad angular es estimada y relacionada con las muestras de las señales originales suministradas por el giróscopo, obteniéndose una recta de calibración para cada uno de los ejes del giróscopo.

Hemos propuesto además un algoritmo que sólo permite iniciar la estimación de los parámetros de calibración cuando la distribución espacial de los datos de entrada del algoritmo es lo suficientemente uniforme. Para ello, se calcula el número de segmentos vacíos en el histograma de las coordenadas esféricas θ y ϕ para un número determinado de segmentos y se compara con un valor mínimo tolerado.

La determinación de la intensidad del movimiento es esencial, ya que muchos algoritmos de calibración necesitan recalcular sus parámetros durante los períodos en los que el sujeto monitorizado está estático. Asimismo, algunos algoritmos de estimación de la orientación cambian de manera dinámica sus parámetros para adaptarse a la naturaleza del movimiento medido y así aumentar la precisión de la estimación. A continuación analizamos los algoritmos hasta ahora utilizados por diversos autores para detectar la intensidad del movimiento partiendo de las señales MARG, y otros utilizados en el ámbito del procesamiento de la voz para discernir entre períodos de silencio y de voz, ya que intuimos que podrían dar buen resultado al ser aplicados sobre nuestras señales. Además de los algoritmos de detección de voz, analizamos también varios algoritmos utilizados habitualmente para detectar cambios abruptos en señales generadas por sensores industriales. Para poder emitir un juicio sobre qué algoritmos poseen mejor precisión, llevamos a cabo un estudio comparativo, tanto sobre señales sintetizadas como reales, de todos los algoritmos seleccionados. Descubrimos que los algoritmos FSD (Framed Spectrum Detector) y LTSD (Long Term Spectral Detector) superaban a los utilizados por otros autores, ya que alcanzan una precisión ligeramente superior al 97% al ser aplicados para discernir entre movimiento e inactividad.

A continuación enfocamos nuestro trabajo en la estimación de la orientación de un cuerpo en el espacio. Estudiamos los diversos sistemas de referencia utilizados en el ámbito de la navegación inercial y las formas más convenientes de representar la orientación respecto a los mismos (ángulos de Euler, matrices de rotación, matriz de dirección del coseno y cuaterniones), para posteriormente introducirnos en las distintas estrategias de estimación de la orientación de un cuerpo en el espacio partiendo de señales MARG. Determinamos cómo la aceleración y el campo magnético medidos pueden ser comparados con los vectores de referencia conocidos (campos gravitatorio y magnético terrestres) para extraer el ángulo entre ellos. También explicamos cómo la integración de la velocidad angular alrededor de un eje determina el ángulo recorrido. Tras el estudio correspondiente se discutieron las ventajas e inconvenientes de cada uno de los métodos y cómo ambos pueden ser fusionados utilizando filtros adaptativos para obtener una mayor precisión. De entre los filtros

adaptativos analizados, seleccionamos el filtro de Kalman, ampliamente utilizado en el ámbito de la estimación de la orientación y en navegación, y algunas variantes del filtro LMS (Least Mean Squares) y RLS (Recursive Least Squares). Describimos de manera detallada el fundamento teórico de los filtros y del esquema de fusión de sensores a través de ellos. Hemos propuesto también una nueva alternativa del filtro de Kalman consistente en la variación dinámica de sus parámetros dependiendo de la intensidad de la actividad. La novedad del mismo reside en la inclusión de los nuevos métodos de detección de intensidad también propuestos por nosotros.

Seguidamente, procedimos a llevar a cabo un estudio comparativo entre los diversos filtros utilizados para calcular la orientación real en cada instante, la cual comparamos con la estimada por cada uno de los métodos. Utilizamos la raíz del error cuadrático medio (RMSE) como medida de comparación entre ellos. También utilizamos un sintetizador que simula señales de ángulo calculadas con el acelerómetro/magnetómetro y el giróscopo, que fue empleado en un estudio comparativo basado en simulación Monte Carlo. De entre todos los algoritmos, observamos que el filtro de Kalman, en su variante dinámica (Gated Kalman Filter), obtuvo la mejor precisión para señales de intensidad mixta, con un error de 2.1132° frente a su variante estática cuyo error fue de 3.5430° . Vimos también que, para señales de mucha intensidad, el filtro HRLS mejoraba la precisión del filtro de Kalman, ($4.3924 \pm 1.9770^\circ$, frente a $5.2274 \pm 0.5976^\circ$).

Nuestro trabajo tiene aplicación directa en la telerehabilitación física de pacientes, el análisis postural, el análisis de las actividades diarias, la detección de caídas, la detección de crisis epilépticas de tipo motriz y en la monitorización, detección y seguimiento de episodios de bradicinesia, discinesia y temblores en enfermos de Parkinson, así como en el análisis de la calidad del sueño y en la monitorización del movimiento del cuerpo en actividades deportivas. Se han abierto nuevos caminos tanto en los algoritmos utilizados para calibrar los sensores, como para estimar la orientación, mediante la propuesta de nuevos métodos. Asimismo se ha realizado un profundo análisis de los métodos existentes incluyendo algunos de ellos en los tres estudios comparativos que se han llevado a cabo. Los resultados de nuestro trabajo se están utilizando en aplicaciones de telerehabilitación implantadas en varios hospitales españoles (Esperanza de Barcelona, Clínico de Barcelona, Son Llátzer de Palma de Mallorca, Virgen de las Nieves de Granada y Torrevieja de Alicante), y está siendo, o va a ser utilizado en otras aplicaciones orientadas a la detección de caídas de personas dependientes, el análisis de la actividad deportiva y la detección de crisis epilépticas motrices (Hospital Virgen de las Nieves de Granada), entre otros.

Cabe también remarcar que se están analizando distintas líneas de trabajo para comercializar los resultados de nuestro trabajo en colaboración con la empresa Ingeniería y Control Remoto S.A. (Spin-off de la Universidad de Granada).

Para finalizar las conclusiones, procedemos a analizar a los objetivos iniciales que se enumeraron en el capítulo de introducción para determinar en qué grado se han visto cubiertos.

- **Calibración:**

- *Objetivo inicial:* Realizar un completo estudio del estado del arte de algoritmos de calibración de magnetómetros, acelerómetros y giróscopos de tipo MEMS.
 - * *Resultados:* Se han analizado más de 100 artículos publicados en los que se presentan algoritmos de calibración de sensores magnéticos e inerciales de tipo MEMS.
- *Objetivo inicial:* Seleccionar e implementar los algoritmos más utilizados en los últimos años con el objetivo de analizar su precisión y rendimiento mediante un estudio comparativo.
 - * *Resultados:* De entre los 100 métodos analizados se han seleccionado e implementado 12 métodos de calibración de acelerómetros y magnetómetros. Dichos métodos han sido comparados en un estudio preliminar que tenía como objetivo determinar el error de las magnitudes físicas medidas una vez calibradas respecto a los vectores gravitatorio y magnético terrestres. En la realización del estudio hemos observado la importancia de la definición de un modelo teórico completo del sensor así como la buena distribución espacial de los datos que se utilizan para estimar los parámetros de calibración. Para ello, se ha desarrollado un sistema que determina de forma automática la distribución espacial de los datos, y sólo permite la estimación de los parámetros cuando ésta es lo suficientemente correcta. Asimismo, se ha extendido el uso de algunos algoritmos que estaban diseñados sólo para calibrar magnetómetros o acelerómetros, de forma que puedan ser usados para calibrar ambos sensores.
- *Objetivo inicial:* Proponer nuevos algoritmos de calibración que no requieran instrumentación compleja.
 - * *Resultados:* Se han propuesto dos nuevos algoritmos para calibrar el giróscopo incluido en IMUs o en MIMUs que no requieren mesas rotatorias multieje con velocidad controlable. Estos algoritmos sólo requieren una superficie rotatoria que posea un elevado momento de rotación (por ejemplo una rueda de bicicleta) y están basados en unas maniobras de calibración muy simples que pueden ser llevadas a cabo por personal no especializado.

- **Estimación de la orientación y posición:**

- *Objetivo inicial:* Realizar un completo estudio del estado del arte de los diversos algoritmos existentes en la literatura.
 - * *Resultados:* Se han analizado más de 80 artículos publicados en las que se presentan algoritmos de estimación de la orientación utilizando fusión de sensores magnéticos y/o iniciales basada en filtrado adaptativo.
 - *Objetivo inicial:* Proponer nuevas variantes de estimación de la orientación basadas en filtrado adaptativo y comparar su rendimiento con algunos de los algoritmos más utilizados en los últimos años.
 - * *Resultados:* Se han propuesto cinco alternativas nuevas de fusión de sensores basadas en el filtro RLS y dos de sus variantes, así como el filtro LMS y dos de sus variantes. Asimismo se ha desarrollado de manera detallada el fundamento teórico de un esquema de fusión basado en el filtro de Kalman estándar. También se ha implementado el algoritmo de Madgwick [80] y el algoritmo FQA [71] (no basado en fusión) y han sido incluidos en el estudio comparativo. Además se ha estudiado el comportamiento de los distintos algoritmos para señales compuestas exclusivamente por alta aceleración, aceleración moderada y señales de aceleración cambiante. Hemos demostrado que el modelo de filtro de Kalman propuesto ofrece el mejor rendimiento para señales de baja aceleración y señales mixtas, mientras que para señales de gran aceleración (movimiento intenso), el filtro HRLS presenta un menor RMSE.
 - *Objetivo inicial:* Proponer algoritmos que modifiquen sus parámetros de forma dinámica en base a la intensidad del movimiento que está siendo monitorizado.
 - * *Resultados:* Se ha propuesto una variante de nuestro algoritmo basado en el filtro de Kalman que adapta los parámetros del modelo de forma dinámica dependiendo del grado de intensidad del movimiento que está siendo monitorizado. Dicho modelo disminuye el RMSE en 1.4° respecto a la variante estática del filtro de Kalman.
 - *Objetivo inicial:* Construir un sistema mecánico de cálculo de la posición que sirva como referencia para el estudio comparativo de los algoritmos existentes seleccionados y los propuestos.
 - * *Resultados:* Se ha construido un sistema de medición de ángulos basado en un potenciómetro lineal que ha servido como referencia para calcular el error de las estimaciones obtenidas por los diversos algoritmos.

Aparte de los objetivos iniciales, también se han cubierto otros objetivos que fueron surgiendo a medida que se avanzaba en el trabajo. En particular:

- *Objetivo:* Proponer nuevos algoritmos de detección de la intensidad de la actividad monitorizada para, por ejemplo, discernir entre períodos en los que el cuerpo monitorizado está estático o en movimiento.
 - *Resultados:* Se ha realizado un estudio comparativo entre cuatro algoritmos de detección de intensidad existentes y cuatro algoritmos aplicados en otros campos, como por ejemplo la detección de actividad de voz y la detección de cambios abruptos en señales. Se ha demostrado que los algoritmos basados en el análisis del espectro de la señal superan la precisión de los algoritmos utilizados como baseline. Además, se ha desarrollado un sintetizador de señales que mimetiza la aceleración y la velocidad angular medidas durante actividades básicas del día a día. Este sintetizador puede resultar de utilidad para aplicaciones de análisis y clasificación de la actividad diaria ya que permite probar nuevos algoritmos sobre un gran número de señales de forma que el estudio sea robusto desde el punto de vista estadístico.
- *Objetivo:* Modificar el firmware de Wagyromag para sincronizar el inicio de lectura de datos de múltiples dispositivos configurados en una red. Este aspecto es necesario para que los datos medidos por distintos Wagyromags colocados en varias partes del cuerpo estén sincronizados en el tiempo.
 - *Resultados:* Se ha desarrollado una versión del firmware que inicia la lectura de datos al recibir de forma inalámbrica una trama de sincronización. Dicha trama de sincronización se difunde a todos los dispositivos utilizando una aplicación de Matlab.

Trabajo futuro

Al igual que hemos hecho con las conclusiones finales, procedemos a analizar el trabajo futuro para cada uno de las partes de nuestro sistema.

- **Hardware e Instrumentación:** Diseñaremos una nueva versión de Wagyromag de forma que incluya los últimos sensores MEMS disponibles en el mercado, ya que durante los últimos años ha habido grandes avances en el aumento de la precisión de los mismos y la reducción de artefactos indeseados en sus mediciones. Asimismo se le dotará de una mayor capacidad de computación de forma que pueda ejecutar los algoritmos de estimación de la orientación desarrollados. Se revisará también la tecnología utilizada para la

transmisión inalámbrica y se analizará la posibilidad de inclusión de otros sensores como por ejemplo un micrófono para aumentar la precisión de los detectores de caídas mediante el análisis del sonido producido por el impacto.

- **Calibración de los sensores:** Estudiaremos el desarrollo de nuevos algoritmos de calibración basados en la minimización de una función de error. Se ha detectado que todos los algoritmos existentes minimizan el error de la magnitud del vector medido respecto al vector de referencia sin tener en cuenta la dirección del mismo. Buscaremos, por tanto, nuevas funciones de error que sí tengan en cuenta la desviación de las componentes medidas respecto a las componentes del vector de referencia. Por otro lado, seguiremos estudiando el desarrollo de algoritmos de calibración del giróscopo basados en instrumentación simple.
- **Detección de la intensidad del movimiento:** Seguiremos mejorando el diseño del sintetizador de señales inerciales de forma que incluya un mayor número de actividades en su repertorio. Asimismo, estudiaremos algoritmos que sean capaces de discernir entre varios niveles de intensidad de actividad.
- **Estimación de la orientación:** Continuaremos desarrollando nuevos modelos de filtrado adaptativo basados en otros filtros, como por ejemplo el filtro extendido de Kalman, el filtro de Kalman "Unscented", filtros de partículas, etc. Por otro lado, seguiremos desarrollando el concepto de variación dinámica de parámetros y contemplaremos la posibilidad de alternar entre diferentes filtros dependiendo de la intensidad de la actividad medida.
- **Aplicaciones del sistema:** Se iniciará el proyecto piloto de detección de crisis tónico-clónicas epilépticas durante el sueño, que se encuentra actualmente en su fase de diseño y se seguirá trabajando en conjunto con equipos médicos para adaptar nuestro sistema a sus necesidades. Se trabajará asimismo en algoritmos de detección de caídas y en proyectos de telerehabilitación y de teleasistencia que necesitan nodos múltiples de medida acoplados a diversos segmentos corporales y requieren gran precisión en la estimación dinámica de la orientación.

Bibliography

- [1] CENIT AmIVital. <http://amivital.ugr.es/>. [Accessed January 14th, 2013].
- [2] A. S. Edelstein, G. A. Fischer, M. Pedersen, E. R. Nowak, Shu Fan Cheng, and C. A. Nordman. Progress toward a thousandfold reduction in 1/f noise in magnetic sensors using an ac microelectromechanical system flux concentrator (invited). *Journal of Applied Physics*, 99(8):08B317 –08B317–6, apr. 2006.
- [3] Miguel Cabrera. Desarrollo de firmware de WAGYROMAG, un dispositivo de medida inercial orientado a la monitorización de la posición del cuerpo humano. Master's thesis, Universidad de Granada, Spain, 2011.
- [4] Kinematics (physics). *Britannica Online Encyclopedia*.
- [5] P. Cerveri, A. Pedotti, and G. Ferrigno. Robust recovery of human motion from video using kalman filters and virtual humans. *Human Movement Science*, 22(3):377–404, August 2003.
- [6] Motion capture systems from vicon. <http://www.vicon.com/>. [Accessed January 14th, 2013].
- [7] Kinect for XBOX 360. <http://www.xbox.com/kinect>. [Accessed January 14th, 2013].
- [8] M.N. Nyan, Francis E.H. Tay, and Matthew Z.E. Mah. Application of motion analysis system in pre-impact fall detection. *Journal of Biomechanics*, 41(10):2297–2304, July 2008.
- [9] MB Greenberg, JA Gronley, J Perry, and R Lawthwaite. Concurrent validity of observational gait analysis using the vicon motion analysis system. *Gait & Posture*, 4(2):167–168, April 1996.
- [10] N. Sakai and S. Shimawaki. Hand motion analysis during touch-typing using VICON system with finger force plate. *Journal of Biomechanics*, 39, Supplement 1(0):S166, 2006.

- [11] W.M. Chung, S. Yeung, W.W. Chan, and R. Lee. Validity of VICON motion analysis system for upper limb kinematic measurement - a comparison study with inertial tracking xsens system. *Hong Kong Physiotherapy Journal*, 29(2):97, December 2011.
- [12] Yao-Jen Chang, Shu-Fang Chen, and An-Fu Chuang. A gesture recognition system to transition autonomously through vocational tasks for individuals with cognitive impairments. *Research in Developmental Disabilities*, 32(6):2064–2068, December 2011.
- [13] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32(6):2566–2570, December 2011.
- [14] Loren Arthur Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, (0), 2011.
- [15] Equivital-mobile human monitoring. <http://www.equivital.co.uk/>. [Accessed January 14th, 2013].
- [16] Microchip PIC24F family. <http://goo.gl/NghlQ>. [Accessed January 14th, 2013].
- [17] Xsens xbus kit : Measurement of human motion - xsens. <http://www.xsens.com/en/general/xbus-kit>. [Accessed January 14th, 2013].
- [18] DynaPort | McRoberts - moving technology. <http://goo.gl/bHbpH>. [Accessed January 14th, 2013].
- [19] Great lakes NeuroTechnologies: movement disorders :: KinetiSense. <http://www.glnurotech.com/KinetiSense/>. [Accessed January 14th, 2013].
- [20] Physilog from LMAM | EPFL. <http://lmam.epfl.ch/>. [Accessed January 14th, 2013].
- [21] Nike+. <http://nikeplus.nike.com/plus/>. [Accessed January 14th, 2013].
- [22] Fitbit. <http://www.fitbit.com>. [Accessed January 14th, 2013].
- [23] Atair aerospace's circinus gps-integrated imu. <http://goo.gl/5HdjP>. [Accessed January 14th, 2013].
- [24] Ch robotics' UM6 miniature orientation sensor. <http://goo.gl/Eujvj>. [Accessed January 14th, 2013].

- [25] Intersense's inertiacube bluetooth. <http://goo.gl/1ux2O>. [Accessed January 14th, 2013].
- [26] X-io technologies' x-IMU IMU/AHRS platform. <http://goo.gl/jBkkZ>. [Accessed January 14th, 2013].
- [27] Xsens' MTw wireless 3D motion tracker. <http://goo.gl/Jk81M>. [Accessed January 14th, 2013].
- [28] Freeimu: a libre 9-10 DOM/DOF inertial measurement unit. <http://goo.gl/Ga7Et>. [Accessed January 14th, 2013].
- [29] Realtrack systems' WIMU. <http://goo.gl/o40gJ>. [Accessed January 14th, 2013].
- [30] Memsense's bluetooth IMU. <http://goo.gl/F5uM7>. [Accessed January 14th, 2013].
- [31] A small list of IMU / INS / INU. <http://goo.gl/43FF7>. [Accessed January 14th, 2013].
- [32] F. Ferraris, U. Grimaldi, and M. Parvis. Procedure for effortless in-field calibration of three-axis rate gyros and accelerometers. *Sensors and Materials*, 42:311–330, 1995.
- [33] S. P Won and F. Golnaraghi. A triaxial accelerometer calibration method using a mathematical model. *IEEE Transactions on Instrumentation and Measurement*, 59(8):2144–2153, August 2010.
- [34] Zhi Li, Xiang Li, and Yongjun Wang. A calibration method for magnetic sensors and accelerometer in tilt-compensated digital compass. In *9th International Conference on Electronic Measurement & Instruments, 2009. ICEMI '09*, pages 2–868–2–871. IEEE, August 2009.
- [35] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira. Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation. *IEEE Transactions on Control Systems Technology*, 19(5):1128–1137, September 2011.
- [36] Zhen Shi, Jie Yang, Peng Yue, and ZiJian Cheng. A discrimination method for accelerometer static model parameter based on nonlinear iterative least squares estimation. In *2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA)*, pages 963–966. IEEE, June 2010.

- [37] Wei Dong, Kwang Yong Lim, Young Koon Goh, Kim Doang Nguyen, I-Ming Chen, Song Huat Yeo, and Been-Lirn Duh. A low-cost motion tracker and its error analysis. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 311 –316, May 2008.
- [38] Shin Eun-Hwan and Naser El-Sheimy. A new calibration method for strap-down inertial navigation systems. *Z. Vermess.wes.*, 127:1–10, 2002.
- [39] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy. A new multi-position calibration method for MEMS inertial navigation systems. *Measurement Science and Technology*, 18(7), 2007.
- [40] Xiaoming Zhang and Lizhen Gao. A novel auto-calibration method of the vector magnetometer. In *9th International Conference on Electronic Measurement & Instruments, 2009. ICEMI '09*, pages 1–145–1–150. IEEE, August 2009.
- [41] D. Campolo, M. Fabris, G. Cavallo, D. Accoto, F. Keller, and E. Guglielmelli. A novel procedure for in-field calibration of sourceless Inertial/Magnetic orientation tracking wearable devices. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006*, pages 471–476. IEEE, February 2006.
- [42] I. Frosio, F. Pedersini, and N. A. Borghese. Autocalibration of MEMS accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 58(6):2034–2041, June 2009.
- [43] Pylvänäinen Timo. Automatic and adaptive calibration of 3D field sensors. *Applied Mathematical Modelling*, 32(4):575–587, April 2008.
- [44] David Jurman, Marko Jankovec, Roman Kamnik, and Marko Topic. Calibration and data fusion solution for the miniature attitude and heading reference system. *Sensors and Actuators A: Physical*, 138(2):411 – 420, 2007.
- [45] Isaac Skog and Peter Handel. Calibration of a MEMS inertial measurement unit. In *in Proc. XVII IMEKO WORLD CONGRESS, (Rio de Janeiro*, 2006.
- [46] S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, and A. Barraud. Calibration methods for inertial and magnetic sensors. *Sensors and Actuators A: Physical*, 156(2):302–311, December 2009.
- [47] V. Petrucha and P. Kaspar. Calibration of a triaxial fluxgate magnetometer and accelerometer with an automated non-magnetic calibration system. In *2009 IEEE Sensors*, pages 1510–1513. IEEE, October 2009.
- [48] L. Sahawneh and M. A. Jarrah. Development and calibration of low cost MEMS IMU for UAV applications. In *Mechatronics and Its Applications, 2008. ISMA 2008. 5th International Symposium on*, pages 1 –9, May 2008.

- [49] Antonio Angrisano, IMU low cost calibration method. <http://goo.gl/NwcfD>. [Accessed January 14th, 2013].
- [50] E. Dorveaux, D. Vissiere, A. -P Martin, and N. Petit. Iterative calibration method for inertial and magnetic sensors. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, pages 8296–8303. IEEE, December 2009.
- [51] Valerie Renaudin, Muhammad Haris Afzal, and Gerard Lachapelle. New method for magnetometers based orientation estimation. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 348–356. IEEE, May 2010.
- [52] 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.
- [53] M. Gietzelt, K. H Wolf, M. Marschollek, and R. Haux. Automatic self-calibration of body worn triaxial-accelerometers for application in health-care. In *Second International Conference on Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008*, pages 177–180. IEEE, February 2008.
- [54] E. R. Bachmann, Xiaoping Yun, D. McKinney, R.B. McGhee, and M. J. Zyda. Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 1171 – 1178 vol.1, September 2003.
- [55] Jafar Keighobadi. Fuzzy calibration of a magnetic compass for vehicular applications. *Mechanical Systems and Signal Processing*, 25(6):1973–1987, August 2011.
- [56] D. Gebre-egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson. A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In *in 8th International St. Petersburg Conference on Navigation Systems*, pages 28–30, 2001.
- [57] Mengchun Pan, Hongfeng Pang, Shitu Luo, Qi Zhang, and Ruifang Xie. Calibration improvement of three-axis magnetometer in disturbing magnetic circumstance based on FIR digital filter. In *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, pages 247–250. IEEE, October 2010.
- [58] Sascha Eichstädt, Alfred Link, Thomas Bruns, and Clemens Elster. Online dynamic error compensation of accelerometers by uncertainty-optimal filtering. *Measurement*, 43(5):708–713, June 2010.

- [59] Hui Yan, Changhan Xiao, Shengdao Liu, and Zhaoyang Zhang. Horizontal error calibration method for triaxial fluxgate magnetometer. In *Automation Congress, 2008. WAC 2008. World*, pages 1–5. IEEE, October 2008.
- [60] Hongliang Zhang, Yuanxin Wu, Wenqi Wu, Meiping Wu, and Xiaoping Hu. Improved multi-position calibration for inertial measurement units. *Measurement Science and Technology*, 21(1):015107, January 2010.
- [61] Ashutosh Saxena, Gaurav Gupta, Vadim Gerasimov, and Sebastien Ourselin. In use parameter estimation of inertial sensors by detecting multilevel quasi-static states. In *Proceedings of the 9th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part IV*, KES'05, pages 595–601, Berlin, Heidelberg, 2005. Springer-Verlag.
- [62] I. Skog, J.-O. Nilsson, and P. Handel. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 –6, September 2010.
- [63] OpenShoe–Publications. http://www.openshoe.org/?page_id=54. [Accessed January 14th, 2013].
- [64] B. Krach and P. Robertson. Integration of foot-mounted inertial sensors into a bayesian location estimation framework. In *5th Workshop on Positioning, Navigation and Communication, 2008. WPNC 2008*, pages 55 –61, March 2008.
- [65] Lauro Ojeda and Johann Borenstein. Non-GPS navigation for security personnel and first responders. *The Journal of Navigation*, 60(03):391–407, 2007.
- [66] I. Skog, P. Handel, J. O Nilsson, and J. Rantakokko. Zero-Velocity Detection algorithm evaluation. *IEEE Transactions on Biomedical Engineering*, 57(11):2657–2666, November 2010.
- [67] P. H Veltink, HansB. J Bussmann, W. de Vries, WimL. J Martens, and R. C Van Lumel. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering*, 4(4):375–385, December 1996.
- [68] Gerald M. Lerner. *"Three-Axis Attitude Determination" in Spacecraft Attitude Determination and Control*. Springer Scientific + Business Media, 1978.
- [69] M D Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance Control and Dynamics*, 4(1):70–77, 1981.
- [70] Xiaoping Yun and E.R. Bachmann. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *Robotics, IEEE Transactions on*, 22(6):1216 –1227, December 2006.

- [71] Yun Xiaoping, E.R. Bachmann, and R.B. McGhee. A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *Instrumentation and Measurement, IEEE Transactions on*, 57(3):638 –650, March 2008.
- [72] Malcolm d. shuster. publications 1970-1998. <http://goo.gl/CNqo5>. [Accessed January 14th, 2013].
- [73] Malcolm d. shuster. publications 2000-present. <http://goo.gl/lhePZ>. [Accessed January 14th, 2013].
- [74] F. Landi Markley and NASA. *Attitude Determination Using Two Vector Measurements*. 1998.
- [75] Francis Landis Markley and Daniele Mortari. Quaternion Attitude Estimation using Vector Observations. *The Journal of the Astronautical Sciences*.
- [76] Francis Landis Markley. Attitude Determination using Vector Observations and the Singular Value Decomposition. *The Journal of the Astronautical Sciences*.
- [77] F. Landi Markley and NASA. *Attitude determination using vector observations: A fast optimal matrix algorithm*. February 1993.
- [78] F. Landis Markley, Daniel Mortari, and NASA. *How to Estimate Attitude from Vector Observations*. 1999.
- [79] J.L. Marins, Xiaoping Yun, E.R. Bachmann, R.B. McGhee, and M.J. Zyda. An extended kalman filter for quaternion-based orientation estimation using MARG sensors. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, volume 4, pages 2003 –2011 vol.4, 2001.
- [80] An efficient orientation filter for inertial and inertial/magnetic sensor arrays. <http://goo.gl/uRpbl>. [Accessed January 14th, 2013].
- [81] Tilt sensing using linear accelerometers (freescale semiconductor application note). <http://goo.gl/0dULQ>. [Accessed January 14th, 2013].
- [82] Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors (freescale semiconductor application note). <http://goo.gl/pGMFX>. [Accessed January 14th, 2013].
- [83] S. luczak. Advanced algorithm for measuring tilt with MEMS accelerometers. In *Recent Advances in Mechatronics*, pages 511–515. Springer Berlin Heidelberg, 2007.

- [84] John L. Crassidis, F. Landis Markley, and Yang Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, January 2007.
- [85] A. D. Young. Comparison of orientation filter algorithms for realtime wireless inertial posture tracking. In *Wearable and Implantable Body Sensor Networks, International Workshop on*, volume 0, pages 59–64, Los Alamitos, CA, USA, 2009. IEEE Computer Society.
- [86] Zhiqiang Zhang, Zhipei Huang, and Jiankang Wu. Ambulatory hip angle estimation using gaussian particle filter. *Journal of Signal Processing Systems*, 58(3):341–357, June 2009.
- [87] H.J. Luinge, P.H. Veltink, and C.T.M. Baten. Ambulatory measurement of arm orientation. *Journal of Biomechanics*, 40(1):78–85, 2007.
- [88] D. Roetenberg, H.J. Luinge, C.T.M. Baten, and P.H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):395–405, September 2005.
- [89] H. J Luinge and P. H Veltink. Inclination measurement of human movement using a 3-D accelerometer with autocalibration. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(1):112–121, March 2004.
- [90] J. Favre, B.M. Jolles, O. Siegrist, and K. Aminian. Quaternion-based fusion of gyroscopes and accelerometers to improve 3D angle measurement. *Electronics Letters*, 42(11):612 – 614, May 2006.
- [91] A. M. Sabatini. Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis. *Medical & Biological Engineering & Computing*, 43(1):94–101, February 2005.
- [92] Tal Amasay, Keely Zodrow, Laurel Kincl, Jennifer Hess, and Andrew Karduna. Validation of tri-axial accelerometer for the calculation of elevation angles. *International Journal of Industrial Ergonomics*, 39(5):783–789, September 2009. [Accessed January 14th, 2013].
- [93] C. C Foster and G. H Elkaim. Extension of a two-step calibration methodology to include nonorthogonal sensor axes. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1070–1078, July 2008.
- [94] Accelerometer types. <http://goo.gl/wy335>. [Accessed January 14th, 2013].
- [95] IEEE standard specification format guide and test procedure for single-axis interferometric fiber optic gyros. *IEEE Std 952-1997*, page i, 1998.

- [96] J. Lenz and S. Edelstein. Magnetic sensors and their applications. *IEEE Sensors Journal*, 6(3):631 –649, June 2006.
- [97] Alan S. Edelstein and Gregory A. Fischer. Minimizing 1/f noise in magnetic sensors using a microelectromechanical system flux concentrator. *J. Appl. Phys.*, 91:7795–7797, 2002.
- [98] ADXL330. small, low power, 3-axis accelerometer. <http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl330/products/product.html>. [Accessed January 14th, 2013].
- [99] Invensense IDG-300. integrated Dual-Axis MEMS gyroscope. <http://invensense.com/mems/gyro/idg300.html>. [Accessed January 14th, 2013].
- [100] Digi XBee wireless RF modules - digi international. <http://www.digi.com/xbee/>. [Accessed January 14th, 2013].
- [101] IEEE 802.15.4 2009 standard. <http://goo.gl/nJt1Y>. [Accessed January 14th, 2013].
- [102] A. Olivares, G. Olivares, F. Mula, J.M. Gorri, and J. Ramirez. Wagyromag: Wireless sensor network for monitoring and processing human body movement in healthcare applications. *Journal of Systems Architecture*, 57(10):905–915, nov 2011.
- [103] Microchip Technology Inc. *PIC24FJ64GB004 Family Data Sheet*, 2009. 28/44-Pin, 16-Bit, Flash Microcontrollers with USB On-The-Go (OTG) and nanoWatt XLP Technology.
- [104] ADXL335, small, low power, 3-Axis $\pm 3\text{ g}$ accelerometer. <http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl335/products/product.html>. [Accessed January 14th, 2013].
- [105] LPR550AL-Dual axis MEMS gyroscope. <http://goo.gl/PyKLF>. [Accessed January 14th, 2013].
- [106] LY550ALH single axis MEMS yaw gyroscope. <http://goo.gl/VXMni>. [Accessed January 14th, 2013].
- [107] FM24V10 F-RAM memory product, ramtron international. <http://ramtron.com/products/nonvolatile-memory/serial-product.aspx?id=109>. [Accessed January 14th, 2013].
- [108] HMC5843, 3-Axis digital compass IC, honeywell. <http://goo.gl/0WF7c>. [Accessed January 14th, 2013].

- [109] MCP9700A, Low-Power linear active thermistor ICs. <http://goo.gl/Iy1oc>. [Accessed January 14th, 2013].
- [110] Oliver J Woodman, C Oliver J Woodman, and Oliver J Woodman. An introduction to inertial navigation. 2007.
- [111] David W. Allan. Allan variance - overview. <http://www.allanstime.com/AllanVariance/>. [Accessed January 14th, 2013].
- [112] Angle random walk, walter stockwell, crossbow technology inc. <http://www.xbow.com/pdf/AngleRandomWalkAppNote.pdf>. [Accessed January 14th, 2013].
- [113] Alberto Olivares. Monitoring and capture of human body motion. University of Granada, 2010.
- [114] Joel Feenstra, Jon Granstroma, and Henry Sodano. Energy harvesting through a backpack employing a mechanically amplified piezoelectric stack. *Mechanical Systems and Signal Processing*, 22:721–734, 2008.
- [115] Q. Li, V. Naing, J.A. Hoffer, D.J. Weber, A.D. Kuo, and J. M. Donelan. Biomechanical energy harvesting: Apparatus and method. *IEEE International Conference on Robotics and Automation*, 2008.
- [116] PEH20w-piezoelectric vibration energy harvester. <http://goo.gl/Qnz6u>. [Accessed January 14th, 2013].
- [117] A. Olivares, G. Olivares, P. Glösekötter, J. M. Gorri, and J. Ramirez. A study of vibration-based energy harvesting in activities of daily living. *4th International ICST Conference on Pervasive Computing Technologies for Healthcare 2010, Munich, Germany*.
- [118] Sensor list pro (android app). lists all sensors available on the device along with their current values and a plot to display their changes over time. <http://goo.gl/yxi8b>. [Accessed January 14th, 2013].
- [119] Sensorlog (ios app). With Sensorlog you can log the sensor data of the Core-location Framework, Coremotion Framework, and gyro raw data as CSV data file to be sent as e-mail. <http://goo.gl/WoLiC>. [Accessed January 14th, 2013].
- [120] A. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. AIAA, January 1997.
- [121] Ideal aerosmith's multi-axis table systems. <http://goo.gl/Nt6qt>. [Accessed January 14th, 2013].

- [122] Ixmotion's 3-axis motion simulators. <http://goo.gl/lNRGx>. [Accessed January 14th, 2013].
- [123] Mut aviation's 2012 series multi-axis table. <http://goo.gl/hSKHG>. [Accessed January 14th, 2013].
- [124] J.C. Lötters, J. Schipper, P.H. Veltink, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. *Sensors and Actuators A: Physical*, 68(1-3):221 – 228, 1998. Eurosensors XI.
- [125] D. Gebre-egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson. A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers. In *in 8th International St. Petersburg Conference on Navigation Systems*, pages 28–30, 2001.
- [126] C. C. Foster and G. H. Elkaim. Extension of a two-step calibration methodology to include nonorthogonal sensor axes. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(3):1070 –1078, jul. 2008.
- [127] Demoz Gebre-Egziabher, Gabriel H Elkaim, J. David Powell, and Bradford W Parkinson. Calibration of strapdown magnetometers in magnetic field domain. *Journal of Aerospace Engineering*, 19(2):87, 2006.
- [128] J. K Bekkeng. Calibration of a novel MEMS inertial reference unit. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1967–1974, June 2009.
- [129] F. Ferraris, U. Grimaldi, and M. Parvis. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [130] F. Ferraris, U. Grimaldi, and M. Parvis. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [131] G. Artese and A. Tecroci. Calibration of a low cost mems ins sensor for an integrated navigation system. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.*, 2006. BioRob 2006, pages 877–882, 2008.
- [132] Xiaoh Wang. Automatic and adaptive correction of diversionary errors in tri-axial magnetometer using neural networks. In *IEEE International Symposium on Knowledge Acquisition and Modeling Workshop*, 2008. KAM Workshop 2008, pages 271–274. IEEE, December 2008.
- [133] W T Fong, S K Ong, and A Y C Nee. Methods for in-field user calibration of an inertial measurement unit without external equipment. *Measurement Science and Technology*, 19(8):085202, August 2008.

- [134] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, September 2007.
- [135] Lotters, J. Schipper, P. Veltink, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. *Sensors and Actuators A: Physical*, 68(1-3):221–228, 1998.
- [136] Rafael López Cano. Estudio de algoritmos de calibración para sensores mems en unidades de medida inerciales. Master's thesis, Universidad de Granada, Spain, 2012.
- [137] National Geophysical Data Center. NGDC geomagnetic calculators - magnetic field. <http://www.ngdc.noaa.gov/geomag-web/#igrfwmm>. [Accessed January 14th, 2013].
- [138] M J Mathie, B G Celler, N H Lovell, and A C F Coster. Classification of basic daily movements using a triaxial accelerometer. *Medical & Biological Engineering & Computing*, 42(5):679–687, September 2004. PMID: 15503970.
- [139] Li Jianli and Du Min. Fuzzy modeling and compensation of scale factor for MEMS gyroscope. In *2010 International Conference on Digital Manufacturing and Automation (ICDMA)*, volume 1, pages 766 –771, December 2010.
- [140] M. Lemkin and B.E. Boser. A three-axis micromachined accelerometer with a CMOS position-sense interface and digital offset-trim electronics. *IEEE Journal of Solid-State Circuits*, 34(4):456–468, April 1999.
- [141] Huiyu Zhou and Huosheng Hu. Reducing drifts in the inertial measurements of wrist and elbow positions. *IEEE T. Instrumentation and Measurement*, 59(3):575–585, 2010.
- [142] A Olivares, J M Gorri, J Ramirez, and G Olivares. Accurate human limb angle measurement: sensor fusion through kalman, least mean squares and recursive least-squares adaptive filtering. *Measurement Science and Technology*, 22(2):025801, February 2011.
- [143] Sang Kyeong Park and Young Soo Suh. A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems. *Sensors*, 10(10):9163–9178, October 2010.
- [144] Young Soo Suh and Sangkyung Park. Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. In *4th International Conference on Autonomous Robots and Agents, 2009. ICARA 2009*, pages 336 –341, February 2009.

- [145] S. Godha, G. Lachapelle, and M. E. Cannon. Integrated gps/ins system for pedestrian navigation in a signal degraded environment. In *Proc. ION GNSS 2006*, pages 2151–2164, 2006.
- [146] S. P. Kwakkel, G. Lachapelle, and M. E. Cannon. GNSS aided in situ human lower limb kinematics during running. pages 1388–1397, September 2008.
- [147] Javier Ramirez, Jose C. Segura, Carmen Benitez, Angel De La Torre, and Antonio Rubio. Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, 42:3–4, 2004.
- [148] Javier Ramirez, Juan Manuel Gorriz, Jose Carlos Segura, Carlos G. Puntonet, Antonio J. Rubio, Senior Member, and Senior Member. Speech/non-speech discrimination based on contextual information integrated bispectrum lrt. In *IEEE Signal Processing Letters*, 2006.
- [149] Juan Manuel Gorriz, Javier Ramirez, Carlos G. Puntonet, Jose Carlos Segura, and Senior Member. Generalized lrt-based voice activity detector. In *IEEE Signal Processing Letters*, 2006.
- [150] J. Ramirez, J. C. Segura, J. M. Gorriz, and L. Garcia. Improved voice activity detection using contextual multiple hypothesis testing for robust speech recognition. *Trans. Audio, Speech and Lang. Proc.*, 15(8):2177–2189, November 2007.
- [151] D. Nikovski and A. Jain. Memory-based algorithms for abrupt change detection in sensor data streams. In *2007 5th IEEE International Conference on Industrial Informatics*, volume 1, pages 547–552. IEEE, June 2007.
- [152] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1-2):100–115, January 1954.
- [153] Michele Basseville and Igor Vladimirovich Nikiforov. *Detection of abrupt changes: theory and application*. Prentice Hall, 1993.
- [154] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.
- [155] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology, Second Edition*. AIAA, 2nd edition, January 2005.
- [156] Jitendra R. Raol. *Multi-Sensor Data Fusion with MATLAB®*. CRC Press, December 2009.
- [157] A.M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*, 53(7):1346 –1356, July 2006.

- [158] A. D. Young, M. J. Ling, and D. K. Arvind. Distributed estimation of linear acceleration for improved accuracy in wireless inertial motion capture. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN '10*, pages 256–267, New York, NY, USA, 2010. ACM.
- [159] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, pages 35–45, 1960.
- [160] G. Welch and G. Bishop. An introduction to the kalman filter. *Notes of ACM SIGGRAPH tutorial on the Kalman filter*, 2001.
- [161] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, December 2008.
- [162] Simon Haykin. *Adaptive Filter Theory (4th Edition)*. Prentice Hall, September 2001. Published: Hardcover.
- [163] Simon Haykin. *Adaptive Filter Theory (4th Edition)*. Prentice Hall, September 2001.
- [164] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong Jr, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692 – 1716, dec. 1975.
- [165] S. C. Douglas and T. H. Y. Meng. Normalized data nonlinearities for lms adaptation. *Signal Processing, IEEE Transactions on*, 42(6):1352 –1365, jun 1994.
- [166] Athanasios Rontogiannis and Sergios Theodoridis. On inverse factorization adaptive least-squares algorithms. *Signal Process.*, 52(1):35–47, 1996.
- [167] Alston Householder. Unitary triangularization of a nonsymmetric matrix. *J. ACM*, 5:339–342, 1958.
- [168] J. G. F. Francis. The qr transformation, part 1. *Comput. J.*, 4:265–271, 1961.
- [169] J. G. F. Francis. The qr transformation , part 2. *Comput. J.*, 4:332–345, 1962.
- [170] S. T. Alexander and A. L. Ghirnikar. A method for recursive least squares filtering based upon an inverse qr decomposition. *IEEE Trans. Signal Process*, 41:20–30, 1993.
- [171] He Jie, Huang Xianlin, and Wang Guofeng. Design and application of single-antenna gps/accelerometers attitude determination system. *Journal of Systems Engineering and Electronics*, 19(2):220 – 227, 2008.

- [172] Farhan A. Faruqi and Kenneth J. Turner. Extended kalman filter synthesis for integrated global positioning/inertial navigation systems. *Applied Mathematics and Computation*, 115(23):213–227, October 2000.
- [173] Xin long Wang and Ya feng Li. An innovative scheme for sins/gps ultra-tight integration system with low-grade imu. *Aerospace Science and Technology*, (0):–, 2011.
- [174] Aboelmagd Noureldin, Ahmed El-Shafie, and Mohamed Bayoumi. Gps/ins integration utilizing dynamic neural networks for vehicular navigation. *Information Fusion*, 12(1):48 – 57, 2011. *|ce:title|Special Issue on Intelligent Transportation Systems|ce:title|*.
- [175] Zhi Shen, Jacques Georgy, Michael J. Korenberg, and Aboelmagd Noureldin. Low cost two dimension navigation using an augmented kalman filter/fast orthogonal search module for the integration of reduced inertial sensor system and global positioning system. *Transportation Research Part C: Emerging Technologies*, 19(6):1111 – 1132, 2011.
- [176] Sheng jie Qin and Fu xue Zhang. A combination of silicon micro-gyroscope that application rotary missile attitude control system. *Physics Procedia*, 22(0):487 – 492, 2011. 11 International Conference on Physics Science and Technology (ICPST 2011).
- [177] ZHANG Long-jie, XIE Xiao-fang, LI De-dong, and WANG Yan. Error analysis of strapdown inertia navigation system in tactical missiles. *Procedia Engineering*, 15(0):1456 – 1460, 2011. CEIS 2011.
- [178] Shuang Li, Yuming Peng, Yuping Lu, Liu Zhang, and Yufei Liu. Mcav/imu integrated navigation for the powered descent phase of mars edl. *Advances in Space Research*, 46(5):557 – 570, 2010.
- [179] Shuang Li and Yuming Peng. Radio beacons/imu integrated navigation for mars entry. *Advances in Space Research*, 47(7):1265 – 1279, 2011.
- [180] Jan Wendel, Oliver Meister, Christian Schlaile, and Gert F. Trommer. An integrated gps/mems-imu navigation system for an autonomous helicopter. *Aerospace Science and Technology*, 10(6):527 – 533, 2006.
- [181] Hyo-Sung Ahn and Chang-Hee Won. Dgps/imu integration-based geolocation system: Airborne experimental test results. *Aerospace Science and Technology*, 13(6):316 – 324, 2009.
- [182] M. Carminati, G. Ferrari, M. Sampietro, and R. Grassetti. Fault detection and isolation enhancement of an aircraft attitude and heading reference system based on mems inertial sensors. *Procedia Chemistry*, 1(1):509 – 512, 2009. Proceedings of the Eurosensors XXIII conference.

- [183] Jorge Torres-Solis and Tom Chau. Wearable indoor pedestrian dead reckoning system. *Pervasive and Mobile Computing*, 6(3):351 – 361, 2010.
- [184] Christoph Fuchs, Nils Aschenbruck, Peter Martini, and Monika Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive and Mobile Computing*, 7(1):1 – 15, 2011.
- [185] Manu Akula, Suyang Dong, Vineet R. Kamat, Lauro Ojeda, Adam Borrell, and Johann Borenstein. Integration of infrastructure based positioning systems and inertial navigation for ubiquitous context-aware engineering applications. *Advanced Engineering Informatics*, 25(4):640 – 655, 2011. Special Section: Advances and Challenges in Computing in Civil and Building Engineering.
- [186] J.-O. Nilsson, I. Skog, P. Handel, and K.V.S. Hari. Foot-mounted ins for everybody - an open-source embedded implementation. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 140 –145, april 2012.
- [187] Jennifer Morris. The value of continuous passive motion in rehabilitation following total knee replacement. *Physiotherapy*, 81(9):557–562, 1995.
- [188] Julie Ratcliffe Sally A. Jessep, Nicola E. Walsh and Michael V. Hurley. Long-term clinical benefits and costs of an integrated rehabilitation programme compared with outpatient physiotherapy for chronic knee pain. *Physiotherapy*, 95(2):94–102, 2009.
- [189] Paolo Bifulco Mario Cesarelli and Marcello Bracale. Quadriceps muscles activation in anterior knee pain during isokinetic exercise. *Medical Engineering and Physics*, 21(6-7):469–478, 1999.
- [190] Yaqin Tao and Huosheng Hu. A novel sensing and data fusion system for 3-d arm motion tracking in telerehabilitation. *IEEE Transactions on Instrumentation and Measurement*, 57(5):1029–1040, May 2008.
- [191] Pedro De Alarcon. A physical telerehabilitation system based on new generation web services. 2008.
- [192] Stephane Choquette, Mathieu Hamel, and Patrick Boissy. Accelerometer-based wireless body area network to estimate intensity of therapy in post-acute rehabilitation. *Journal of Neuroengineering and Rehabilitation*, 5:20, 2008. PMID: 18764954.
- [193] Paolo Bonato. Advances in wearable technology and applications in physical medicine and rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2:2, February 2005. PMID: 15733322 PMCID: 552335.

- [194] D. Kairy, C. Vincent, and P. Lehoux. How do the properties of telerehabilitation technologies change clinical practice and interprofessional communication? a qualitative case-study. In *Virtual Rehabilitation International Conference, 2009*, pages 16–22. IEEE, July 2009.
- [195] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation survey. *Biomedical Signal Processing and Control*, 3(1):1–18, January 2008.
- [196] M. Hamel, R. Fontaine, and P. Boissy. In-home telerehabilitation for geriatric patients. *IEEE Engineering in Medicine and Biology Magazine*, 27(4):29–37, August 2008.
- [197] J. E Deutsch, J. A Lewis, and G. Burdea. Virtual reality-integrated telerehabilitation system: Patient and technical performance. In *2006 International Workshop on Virtual Rehabilitation*, pages 140–144. IEEE.
- [198] Michael Sung, Carl Marci, and Alex Pentland. Wearable feedback systems for rehabilitation. *Journal of Neuroengineering and Rehabilitation*, 2:17, 2005. PMID: 15987514.
- [199] Anita SantAnna and Nicholas Wickstrom. Developing a motion language: Gait analysis from accelerometer sensor systems. pages 1 –8, april 2009.
- [200] K. Zhang, A. Gorjian, and D. K. Lester. Gait change after local anesthetic of chronically arthritic knee. *J Long Term Eff Med Implants*, 16(3):223–234, 2006.
- [201] BA MacWilliams, MC Sardelli, and RZ Tashjian. A functional axis based upper extremity model and associated calibration procedures. *Gait & posture*, 31(2):289–291, February 2010.
- [202] Joanne E. Wittwer, Kate E. Webster, and Hylton B. Menz. A longitudinal study of measures of walking in people with alzheimer’s disease. *Gait & Posture*, 32(1):113–117, May 2010.
- [203] Marianne B. van Iersel, Marcel G.M. Olde Rikkert, and George F. Borm. A method to standardize gait and balance variables for gait velocity. *Gait & Posture*, 26(2):226–230, July 2007.
- [204] Alberto Ferrari, Andrea Giovanni Cutti, and Angelo Cappello. A new formulation of the coefficient of multiple correlation to assess the similarity of waveforms measured synchronously by different motion analysis protocols. *Gait & Posture*, 31(4):540–542, April 2010.
- [205] Oliver Rettig, Laetitia Fradet, Philip Kasten, Patric Raiss, and Sebastian I. Wolf. A new kinematic model of the upper extremity based on functional joint parameter determination for shoulder and elbow. *Gait & Posture*, 30(4):469–476, November 2009.

- [206] J.H.M. Bergmann, R.E. Mayagoitia, and I.C.H. Smith. A novel method for determining ground-referenced contacts during stair ascent: Comparing relative hip position to quiet standing hip height. *Gait & Posture*, 31(2):164–168, February 2010.
- [207] Susan P. Barker, William Freedman, and Howard Hillstrom. A novel method of producing a repetitive dynamic signal to examine reliability and validity of gait analysis systems. *Gait & Posture*, 24(4):448–452, December 2006.
- [208] Thomas D. Collins, Salim N. Ghoussayni, David J. Ewins, and Jenny A. Kent. A six degrees-of-freedom marker set for gait analysis: Repeatability and comparison with a modified helen hayes set. *Gait & Posture*, 30(2):173–180, August 2009.
- [209] R. Senden, B. Grimm, I.C. Heyligers, H.H.C.M. Savelberg, and K. Meijer. Acceleration-based gait test for healthy subjects: Reliability and reference data. *Gait & Posture*, 30(2):192–196, August 2009.
- [210] Justin J. Kavanagh and Hylton B. Menz. Accelerometry: A technique for quantifying movement patterns during walking. *Gait & Posture*, 28(1):1–15, July 2008.
- [211] Andrea Giovanni Cutti, Andrea Giovanardi, Laura Rocchi, Angelo Davalli, and Rinaldo Sacchetti. Ambulatory measurement of shoulder and elbow kinematics through inertial and magnetic sensors. *Medical & Biological Engineering & Computing*, 46(2):169–178, February 2008. PMID: 18087742.
- [212] A.L. Hof and E. Otten. Assessment of two-dimensional induced accelerations from measured kinematic and kinetic data. *Gait & Posture*, 22(3):182–188, November 2005.
- [213] Claudine J.C. Lamoth, Rob C. van Lummel, and Peter J. Beek. Athletic skill level is reflected in body sway: A test case for accelerometry in combination with stochastic dynamics. *Gait & Posture*, 29(4):546–551, June 2009.
- [214] E. Gallasch and T. Kenner. Characterisation of arm microvibration recorded on an accelerometer. *European Journal of Applied Physiology*, 75(3):226–232, February 1997.
- [215] Antonia Hartmann, Susanna Luzi, Kurt Murer, Rob A. de Bie, and Eling D. de Bruin. Concurrent validity of a trunk tri-axial accelerometer system for gait analysis in older adults. *Gait & Posture*, 29(3):444–448, April 2009.
- [216] A. Santanna and N. Wickstrom. Developing a motion language: Gait analysis from accelerometer sensor systems. In *3rd International Conference on Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009*, pages 1–8. IEEE, April 2009.

- [217] Werghi Naoufel. A discriminative 3D wavelet-based descriptors: Application to the recognition of human body postures. *Pattern Recognition Letters*, 26(5):663–677, April 2005.
- [218] Eirik Arsand, Odd-Arne Olsen, Ragnhild Varmedal, Willy Mortensen, and Gunnar Hartvigsen. A system for monitoring physical activity data among people with type 2 diabetes. *Studies in Health Technology and Informatics*, 136:113–118, 2008. PMID: 18487717.
- [219] C. V.C Bouten, K. T.M Koekkoek, M. Verduin, R. Kodde, and J. D Janssen. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering*, 44(3):136–147, March 1997.
- [220] Jordi Parera and Cecilio Angulo. Accelerometer signals analysis using SVM and decision tree in daily activity identification. *Gerontechnology*, 7(2):184–184, 2008.
- [221] Merryn J Mathie, Adelle C F Coster, Nigel H Lovell, and Branko G Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological Measurement*, 25(2):R1–R20, April 2004.
- [222] Nishkam Ravi, Nikhil D, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 1541—1546, 2005.
- [223] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. pages 1—17, 2004.
- [224] James Huddleston, Amer Alaiti, Dov Goldvasser, Donna Scarborough, Andrew Freiberg, Harry Rubash, Henrik Malchau, William Harris, and David Krebs. Ambulatory measurement of knee motion and physical activity: preliminary evaluation of a smart activity monitor. *Journal of NeuroEngineering and Rehabilitation*, 3:21, September 2006. PMID: 16970818 PMCID: 1592500.
- [225] F. R Allen, E. Ambikairajah, N. H Lovell, and B. G Celler. An adapted gaussian mixture model approach to accelerometry-based movement classification using time-domain features. In *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006. EMBS '06*, pages 3600–3603. IEEE, September 2006.
- [226] Han-Yi Huang, Wei-Chun Chen, Wei-Chun Cheng, Yu-Ta Lai, and Jiu-Jenq Lin. RT3 accelerometer shoulder activity: Discrimination of activity levels by

- the RT3 accelerometer for the assessment of shoulder physical activity (SPA). *Manual Therapy*, 16(2):172–176, April 2011.
- [227] Falls and injury statistics for seniors and elderly. <http://www.learnnottofall.com/content/fall-facts/how-often.jsp>. [Accessed January 14th, 2013].
- [228] About epilepsy. <http://www.epilepsyfoundation.org/aboutepilepsy/>. [Accessed January 14th, 2013].
- [229] Tonic clonic seizure: Overview. <http://www.freemd.com/tonic-clonic-seizure/overview.htm>. [Accessed January 14th, 2013].
- [230] Epilepsia-estadisticas. <http://goo.gl/h4NRD>. [Accessed January 14th, 2013].
- [231] CDC - epilepsy - basics - fast facts. <http://goo.gl/3FKfx>. [Accessed January 14th, 2013].
- [232] Epilepsy stats and facts | epilepsy.com. <http://goo.gl/D6W8V>. [Accessed January 14th, 2013].
- [233] T. M. E. Nijzen, P. J. M. Cluitmans, J. B. A. M. Arends, and P. A. M. Griep. Detection of subtle nocturnal motor activity from 3-d accelerometry recordings in epilepsy patients. *Biomedical Engineering, IEEE Transactions on*, 54(11):2073 –2081, nov. 2007.
- [234] K. Cuppens, L. Lagae, B. Ceulemans, S. Van Huffel, and B. Vanrumste. Detection of nocturnal frontal lobe seizures in pediatric patients by means of accelerometers: A first study. pages 6608 –6611, sep. 2009.
- [235] Joachim Elevant. *Monitoring Epilepsy With a Wrist Carried Motion Sensor*. PhD thesis.
- [236] Juliana Lockman, Robert S. Fisher, and Donald M. Olson. Detection of seizure-like movements using a wrist accelerometer. *Epilepsy & Behavior*, 20(4):638–641, 2011.
- [237] Ming-Zher Poh, Tobias Loddenkemper, Claus Reinsberger, Nicholas C. Swenson, Shubhi Goyal, Mangwe C. Sabtala, Joseph R. Madsen, and Rosalind W. Picard. Convulsive seizure detection using a wrist-worn electrodermal activity and accelerometry biosensor. *Epilepsia*, 53(5):e93–e97, 2012.
- [238] Eva Schulc, Iris Unterberger, Samrend Saboor, Johannes Hilbe, Markus Ertl, Elske Ammenwerth, Eugen Trinka, and Christa Them. Measurement and quantification of generalized tonic-clonic seizures in epilepsy patients by means of accelerometry - an explorative study. *Epilepsy Research*, 95(1-2):173–183, 2011.

- [239] Definition of parkinson's disease. <http://goo.gl/A5HO8>. [Accessed January 14th, 2013].
- [240] Statistics on parkinson's - parkinson's disease foundation (PDF). http://www.pdf.org/en/parkinson_statistics. [Accessed January 14th, 2013].
- [241] A. Salarian, H. Russmann, F.J.G. Vingerhoets, P.R. Burkhard, Y. Blanc, C. Dehollain, and K. Aminian. An ambulatory system to quantify bradykinesia and tremor in parkinson's disease. pages 35 – 38, april 2003.
- [242] P.H. Veltink, E.G.O. Engberink, B.J. Van Hilten, R. Dunnewold, and C. Jacobi. Towards a new method for kinematic quantification of bradykinesia in patients with parkinson's disease using triaxial accelerometry. volume 2, pages 1303 –1304 vol.2, sep 1995.
- [243] S. Patel, D. Sherrill, R. Hughes, T. Hester, N. Huggins, T. Lie-Nemeth, D. Standaert, and P. Bonato. Analysis of the severity of dyskinesia in patients with parkinson's disease via wearable sensors. pages 4 pp. –126, april 2006.
- [244] S. L Smith and K. Shannon. Vector-based analysis of motor activities in patients with parkinson's disease. In *EUROMICRO 97. 'New Frontiers of Information Technology'. Short Contributions., Proceedings of the 23rd Euromicro Conference*, pages 50–55. IEEE, September 1997.
- [245] S. Patel, T. Hester, R. Hughes, N. Huggins, D. Standaert, Alice, A. Flaherty, and P. Bonato. Using wearable sensors to enhance DBS parameter adjustment for parkinson's disease patients through measures of motor response. In *3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors, 2006*, pages 141–144. IEEE, September 2006.
- [246] O. Sprdlik, Z. Hurak, M. Hoskovcova, and E. Ruzicka. Tremor analysis by decomposition of acceleration into gravity and inertial acceleration using inertial measurement unit. In *9th International Conference on Information Technology and Applications in Biomedicine, 2009. ITAB 2009*, pages 1–4. IEEE, November 2009.
- [247] P.H. Veltink, E.G.O. Engberink, B.J. Van Hilten, R. Dunnewold, and C. Jacobi. Towards a new method for kinematic quantification of bradykinesia in patients with parkinson's disease using triaxial accelerometry. volume 2, pages 1303–1304. IEEE, 1995.
- [248] Ge Wu and Zvi Ladin. The study of kinematic transients in locomotion using the integrated kinematic sensor. *IEEE Transactions on Rehabilitation Engineering*, 4(3):193–200, September 1996.

- [249] Arash Salarian, Heike Russmann, Christian Wider, Pierre R Burkhard, François J G Vingerhoets, and Kamiar Aminian. Quantification of tremor and bradykinesia in parkinson's disease using a novel ambulatory monitoring system. *IEEE Transactions on Bio-Medical Engineering*, 54(2):313–322, February 2007. PMID: 17278588.
- [250] R. LeMoine, C. Coroian, and T. Mastroianni. Quantification of parkinson's disease characteristics using wireless accelerometers. In *ICME International Conference on Complex Medical Engineering, 2009. CME*, pages 1–5. IEEE, April 2009.
- [251] J. Ghika, A. W Wiegner, J. J Fang, L. Davies, R. R Young, and J. H Growdon. Portable system for quantifying motor abnormalities in parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 40(3):276–283, March 1993.
- [252] W. T Latt, U. X Tan, K. C Veluvolu, C. Y Shee, and W. T Ang. Physiological tremor sensing using only accelerometers for real-time compensation. In *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008*, pages 474–479. IEEE, February 2009.
- [253] A. Salarian, H. Russmann, F. J.G Vingerhoets, C. Dehollain, Y. Blanc, P. R Burkhard, and K. Aminian. Gait assessment in parkinson's disease: toward an ambulatory system for long-term monitoring. *IEEE Transactions on Biomedical Engineering*, 51(8):1434–1443, August 2004.
- [254] M. Pavel, T. Hayes, I. Tsay, D. Erdoganmus, A. Paul, N. Larimer, H. Jimison, and J. Nutt. Continuous assessment of gait velocity in parkinson's disease from unobtrusive measurements. In *3rd International IEEE/EMBS Conference on Neural Engineering, 2007. CNE '07*, pages 700–703. IEEE, May 2007.
- [255] Aner Weiss, Talia Herman, Meir Plotnik, Marina Brozgol, Inbal Maidan, Nir Giladi, Tanya Gurevich, and Jeffrey M. Hausdorff. Can an accelerometer enhance the utility of the timed up & go test when evaluating patients with parkinson's disease? *Medical Engineering & Physics*, 32(2):119–125, March 2010.
- [256] S. Patel, D. Sherrill, R. Hughes, T. Hester, N. Huggins, T. Lie-Nemeth, D. Standaert, and P. Bonato. Analysis of the severity of dyskinesia in patients with parkinson's disease via wearable sensors. In *International Workshop on Wearable and Implantable Body Sensor Networks, 2006. BSN 2006*, pages 4 pp.–126. IEEE, April 2006.
- [257] Shyamal Patel, Konrad Lorincz, Richard Hughes, Nancy Huggins, John H Growdon, Matt Welsh, and Paolo Bonato. Analysis of feature space for monitoring persons with parkinson's disease with application to a wireless wear-

- able sensor system. *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, 2007*:6291–6294, 2007. PMID: 18003459.
- [258] A. Salarian, H. Russmann, F. J.G Vingerhoets, P. R Burkhard, Y. Blanc, C. Dehollain, and K. Aminian. An ambulatory system to quantify bradykinesia and tremor in parkinson's disease. In *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003*, pages 35– 38. IEEE, April 2003.
- [259] Patrick Esser, Helen Dawes, Johnny Collett, Max G. Feltham, and Ken Howells. Validity and inter-rater reliability of inertial gait measurements in parkinson's disease: A pilot study. *Journal of Neuroscience Methods*, 205(1):177 – 181, 2012.
- [260] Otakar Sprdlik, Zdenek Hurak, Martina Hoskovcova, Olga Ulmanova, and Evzen Ruzicka. Tremor analysis by decomposition of acceleration into gravity and inertial acceleration using inertial measurement unit. *Biomedical Signal Processing and Control*, 6(3):269 – 279, 2011. ITAB 2009.
- [261] GCDC USB-Accelerometer actigraph. <http://www.gcdadataconcepts.com/xlr8r-actigraph.html>.
- [262] WakeMate–sleep monitor. <http://www.wakemate.com/>.
- [263] Nicole K.Y Tang and Allison G Harvey. Correcting distorted perception of sleep in insomnia: a novel behavioural experiment? *Behaviour Research and Therapy*, 42(1):27 – 39, 2004.
- [264] D.S. Morillo, J.L.R. Ojeda, L.F.C. Foix, and A.L. Jimenez. An accelerometer-based device for sleep apnea screening. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):491 –499, march 2010.
- [265] J. W. Harding, C. G. Mackintosh, A. G. Hahn, and D. A. James. Classification of aerial acrobatics in elite half-pipe snowboarding using body mounted inertial sensors. *The Engineering of Sport 7 - Vol. 2*, 2009.
- [266] Yuji Ohgi. Mems sensor application for the motion analysis in sports science. *ABCM Symposium Series in Mechatronics*, 2009.
- [267] Margaret Estivalet, Pierre Brisson, JI Cowie, JA Flint, and AR Harland. Wireless impact measurement for martial arts (P43). In *The Engineering of Sport 7*, pages 231–237. Springer, 2008.

- [268] E. A Heinz, K. S Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In *2006 IEEE Symposium on Computational Intelligence and Games*, pages 98–102. IEEE, May 2006.
- [269] Aino Ahtinen, Minna Isomursu, Jussi Huhtala, Ykaand Kaasinen, Jukka Salminen, and Jonna Hakkila. Tracking outdoor sports: a user experience perspective. In *Proceedings of the European Conference on Ambient Intelligence, AmI '08*, pages 192–209, Berlin, Heidelberg, 2008. Springer-Verlag.
- [270] A. V Fechan, Y. B Okup, O. Y Sushynsky, O. Z Blavt, and M. O Shymchyshyn. The goalkeepers reaction control by using 3D accelerometer. In *2008 Proceedings of International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, pages 139–139. IEEE, February 2008.
- [271] Daniel A James. The application of inertial sensors in elite sports monitoring. *Sensors Peterborough NH*, pages 289–294–294, 2006.
- [272] D. Alpini, A. Hahn, and D. Riva. Static and dynamic postural control adaptations induced by playing ice hockey. *Sport Sciences for Health*, 2(3):85–92, April 2008.
- [273] Margaret Estivalet, Pierre Brisson, S. N. Omkar, and D. K. Ganesh. Stability training and measurement system for sportsperson (P84). In *The Engineering of Sport 7*, pages 435–442. Springer Paris, Paris, 2009.
- [274] M. Baichlin, M. Kusserow, G. Troister, and H. Gubelmann. Ski jump analysis of an olympic champion with wearable acceleration sensors. In *2010 International Symposium on Wearable Computers (ISWC)*, pages 1–2. IEEE, October 2010.
- [275] A. Wixted, D. Thiel, D. James, A. Hahn, C. Gore, and D. Pyne. Signal processing for estimating energy expenditure of elite athletes using triaxial accelerometers. In *2005 IEEE Sensors*. IEEE, November 2005.
- [276] Margaret Estivalet, Pierre Brisson, M. Wijnen, M. B. Hoppenbrouwers, and J. W. M. Willems. Runalyser: Real time analysis of running technique in practice (P196). In *The Engineering of Sport 7*, pages 289–295. Springer Paris, Paris, 2008.
- [277] Zhen Gao, Quanjun Song, Ming Liu, Jianhe Lei, Yong Yu, and YunJian Ge. Research on a throwing multidimensional force sensing system for advanced training of shot-put athletes. In *2006 IEEE International Conference on Information Acquisition*, pages 376–381. IEEE, August 2006.

- [278] Shuangwei Xie, Quanjun Song, Ming Liu, and Yunjian Ge. Research on a throwing information acquiring system for discus-throw athletes. In *IEEE International Conference on Robotics and Biomimetics, 2007. ROBIO 2007*, pages 394–398. IEEE, December 2007.
- [279] Eckehard Fozzy Moritz, Steve Haake, and Jon Iribarri Berrostegieta. Relation between specific force tests and chained degree in high level sport climbers. In *The Engineering of Sport 6*, pages 275–280. Springer New York, New York, NY, 2006.
- [280] R Foster, L Lanningham-Foster, and J Levine. Optimization of accelerometers for measuring walking. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 222(1):53 –60, March 2008.
- [281] R. C Gonzalez, D. Alvarez, A. M Lopez, and J. C Alvarez. Modified pendulum model for mean step length estimation. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007*, pages 1371–1374. IEEE, August 2007.
- [282] Yuji Ohgi, Kazuya Seo, Nobuyuki Hirai, and Masahide Murakami. Measurement of jumpers body motion in ski jumping. In Eckehard Fozzy Moritz and Steve Haake, editors, *The Engineering of Sport 6*, pages 275–280. Springer New York, New York, NY, 2006.
- [283] Andrew J. Wixted, David V. Thiel, Allan G. Hahn, Christopher J. Gore, David B. Pyne, and Daniel A. James. Measurement of energy expenditure in elite athletes using MEMS-Based triaxial accelerometers. *IEEE Sensors Journal*, 7(4):481–488, April 2007.
- [284] Maxine Kwan, Ching-Lung Cheng, Wen-Tzu Tang, and John Rasmussen. Measurement of badminton racket deflection during a stroke. *Sports Engineering*, 12(3):143–153, March 2010.
- [285] Chang Tai Kiang, Chan Kuan Yoong, and A. C Spowage. Local sensor system for badminton smash analysis. In *IEEE Instrumentation and Measurement Technology Conference, 2009. I2MTC '09*, pages 883–888. IEEE, May 2009.
- [286] P. V Bayly, R. Naunheim, J. Standeven, J. S Neubauer, L. Lewis, and G. M Genin. Linear and angular accelerations of the human head during heading of a soccer ball. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 3, pages 2577– 2578 vol.3. IEEE, October 2002.

- [287] Yosuke Yamada, Keiichi Yokoyama, Risa Noriyasu, Tomoaki Osaki, Tetsuji Adachi, Aya Itoi, Yoshihiko Naito, Taketoshi Morimoto, Misaka Kimura, and Shingo Oda. Light-intensity activities are important for estimating physical activity energy expenditure using uniaxial and triaxial accelerometers. *European Journal of Applied Physiology*, 105(1):141–152, January 2009. PMID: 18853176.
- [288] A. Ahmadi, D. D Rowlands, D. A James, and A. Ahmadi. Investigating the translational and rotational motion of the swing using accelerometers for athlete skill assessment. In *5th IEEE Conference on Sensors, 2006*, pages 980–983. IEEE, October 2006.
- [289] Yann Le Meur, Christophe Hausswirth, Sylvain Dorel, Frank Bignet, Jeanick Brisswalter, and Thierry Bernard. Influence of gender on pacing adopted by elite triathletes during a competition. *European Journal of Applied Physiology*, 106(4):535–545, July 2009. PMID: 19340453.
- [290] S. Daukantas, V. Marozas, and A. Lukosevicius. Inertial sensor for objective evaluation of swimmer performance. In *Electronics Conference, 2008. BEC 2008. 11th International Biennial Baltic*, pages 321–324. IEEE, October 2008.
- [291] Margaret Estivalet, Pierre Brisson, Nicola Petrone, and Giuseppe Marcolin. Impact behaviour of ski-boots against different obstacles (P243). In *The Engineering of Sport 7*, pages 489–497. Springer Paris, Paris, 2008.
- [292] G. S Chambers, S. Venkatesh, G. A.W West, and H. H Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *16th International Conference on Pattern Recognition, 2002. Proceedings*, volume 2, pages 1082– 1085 vol.2. IEEE, 2002.
- [293] Magnus Dencker, Ola Thorsson, Magnus K Karlsson, Christian Lindén, Stig Eiberg, Per Wollmer, and Lars Bo Andersen. Gender differences and determinants of aerobic fitness in children aged 8-11 years. *European Journal of Applied Physiology*, 99(1):19–26, January 2007. PMID: 17024465.
- [294] Matthew Brodie, Alan Walmsley, and Wyatt Page. Fusion motion capture: a prototype system using inertial measurement units and GPS for the biomechanical analysis of ski racing. *Sports Technology*, 1(1):17–28, 2008.
- [295] Margaret Estivalet, Pierre Brisson, Devinder Grewal, Eric Rossetter, Chris Lund, and Benjamin J. Ewers. Experimental measurement of selected snowboard mechanical properties (P194). In *The Engineering of Sport 7*, pages 279–287. Springer Paris, Paris, 2008.
- [296] Mitchell Page and Andrew Vande Moere. Evaluating a wearable display jersey for augmenting team sports awareness. In *Proceedings of the 5th international*

- conference on Pervasive computing*, PERVASIVE'07, pages 91–108, Berlin, Heidelberg, 2007. Springer-Verlag.
- [297] Heyoung Lee, Jung Wook Park, and Abdelsalam Helal. Estimation of indoor physical activity level based on footstep vibration signal measured by MEMS accelerometer in smart home environments. In *Proceedings of the 2nd international conference on Mobile entity localization and tracking in GPS-less environments*, MELT'09, pages 148–162, Berlin, Heidelberg, 2009. Springer-Verlag.
 - [298] Scott E Crouter, James R Churilla, and Jr Bassett, David R. Estimating energy expenditure using accelerometers. *European Journal of Applied Physiology*, 98(6):601–612, December 2006. PMID: 17058102.
 - [299] Eckehard Fozzy Moritz, Steve Haake, and Jarg F. Wagner. Employing modern elements of vehicle navigation for integrated motion measurement in sport. In *The Engineering of Sport 6*, pages 51–56. Springer New York, New York, NY, 2006.

