



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

ÁREA DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**CÓDIGO DE LA ASIGNATURA “TRABAJO FIN DE MÁSTER”:
MINGIND2-2-030**

**CALIBRACIÓN Y EVALUACIÓN METROLÓGICA DE UNIDADES
DE MEDIDA INERCIAL PARA LA MONITORIZACIÓN DEL
MOVIMIENTO HUMANO**

**D. CASTELLANOS ALONSO, Alejandro
TUTOR: D. ÁLVAREZ ÁLVAREZ, Juan Carlos**

FECHA: julio de 2023

RESUMEN

El eje central del presente *Trabajo Fin de Máster (TFM)* se configura en torno a la estimación inercial de varios movimientos articulares. En aras de desarrollar dicha actividad, se han seleccionado varias familias de sensores inerciales; cuya calibración ha constituido la etapa previa a la evaluación inercial propuesta. De esta forma, se ha efectuado una comparación entre los resultados obtenidos para cada gama de unidades de medida inercial.

De forma complementaria a lo indicado en el párrafo anterior, se ha analizado el efecto que poseen los algoritmos de cálculo de la orientación sobre la estimación inercial de los resultados finales. El detalle de los algoritmos contemplados puede consultarse posteriormente a lo largo de este documento.

Adicionalmente al estudio indicado con anterioridad, se ha incluido la planificación temporal y la presupuestación ligadas a este proyecto. Asimismo, al final del documento se han elaborado varios anexos; los cuales se han destinado a profundizar en algunos aspectos teóricos y/o de implementación surgidos durante el transcurso del *TFM*.

PALABRAS CLAVE

Sensores inerciales (*IMUs*), calibración, *Xsens DOT*, *IMUs MATRIX*, *SuperSens*, captura del movimiento, *OptiTrack*, *Motive*, robot colaborativo *UR3e*, acelerómetro, giroscopio, magnetómetro, *Sensors Validation*, evaluación metrológica, rotación articular, estimación inercial, *MatLab*.

ÍNDICE

CAPÍTULO 1. MEMORIA	20
CAPÍTULO 2. PRESUPUESTO	239
CAPÍTULO 3. MATERIALES ADJUNTADOS	253

ÍNDICE DE LA MEMORIA

1. INTRODUCCIÓN	21
1.1.- DEFINICIÓN DEL PROBLEMA	21
1.1.1.- <i>Unidades de medida inercial y algoritmos estudiados</i>	22
1.2.- REVISIÓN TÉCNICA (ESTADO DEL ARTE)	24
1.3.- RESUMEN DEL PROCEDIMIENTO DISEÑADO	26
1.4.- SECUENCIA DE HITOS DEFINIDA PARA EL PROYECTO	27
2. MARCO TEÓRICO	30
2.1.- MÉTODO COMPUTACIONAL DESTINADO A CUANTIFICAR EL MOVIMIENTO TRIDIMENSIONAL DE UNA ARTICULACIÓN (DIFERENCIA DE CUATERNIONES)	30
2.1.1.- <i>Incorporación de los errores de orientación y alineación sensor – segmento durante el cálculo del cuaternión final de ROTACIÓN ARTICULAR</i>	35
2.1.2.- <i>Cálculo de los errores generados por el algoritmo durante la estimación de la orientación 3D, asociada al codo</i>	42
2.1.3.- <i>Representación gráfica relativa al problema de evaluación inercial planteado</i>	42
2.2.- ALGORITMO TRIAD DE EVALUACIÓN INERCIAL	44
3. METODOLOGÍA DE TRABAJO	46
3.1.- ESTRUCTURA DE LOS EXPERIMENTOS REALIZADOS	46
3.2.- DESCRIPCIÓN DEL EXPERIMENTO DE CALIBRACIÓN INERCIAL	46
3.2.1.- <i>Metodología experimental de calibración tridimensional (3D)</i>	49
3.2.2.- <i>Metodología experimental de calibración bidimensional (2D)</i>	50
3.2.3.- <i>Algoritmos empleados durante la calibración (2D y 3D) de los sensores inerciales</i>	50
3.3.- METODOLOGÍA EXPERIMENTAL LIGADA A LA EVALUACIÓN INERCIAL	53
4. DESCRIPCIÓN DEL SISTEMA UTILIZADO	57
4.1.- SOFTWARE Y DISPOSITIVOS SELECCIONADOS	57
4.2.- MATERIAL HARDWARE EMPLEADO. RESUMEN DE CARACTERÍSTICAS TÉCNICAS	57
4.2.1.- <i>Sensores inerciales de la familia Xsens (IMUs Xsens DOT)</i>	57
4.2.2.- <i>Unidades de medida inercial pertenecientes a la gama Matrix</i>	61
4.2.3.- <i>SuperSens/bIMUr: Sensor inercial desarrollado por el grupo de investigación SiMuR</i>	62
4.2.4.- <i>Robot colaborativo (cobot) UR3e. Caja de control “Teach Pendant”</i>	63
4.3.- DISEÑO SOFTWARE	66
4.3.1.- <i>Modelo de RoboDK/Python que permite gobernar el funcionamiento del cobot UR3e</i>	67
5. RESULTADOS	68
5.1.- CARACTERIZACIÓN DE LOS SENSORES INERCIALES	68
5.1.1.- <i>Estudio sobre el estado de los sensores considerados, como paso previo a la calibración de dichos dispositivos</i>	68
5.1.2.- <i>Ortogonalización de los sistemas de coordenadas asociados a cada sensor inercial</i>	72
5.2.- CALIBRACIÓN DE LOS SENSORES INERCIALES	77
5.3.- EVALUACIÓN DE LOS SENSORES Y LOS ALGORITMOS DE ESTIMACIÓN INERCIAL	82
5.3.1.- <i>IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: “ZXZ”). Algoritmo de estimación de orientación: Ángulos de Euler suministrados por cada dispositivo Xsens DOT (filtro propietario). Algoritmo de evaluación inercial: diferencia de cuaterniones</i>	83

5.3.2.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: "YXZ"). Algoritmo de estimación de orientación: imufilter. Algoritmo de evaluación inercial: diferencia de cuaterniones.....	85
5.3.3.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: "ZXZ"). Algoritmo de estimación de orientación: ecompass. Algoritmo de evaluación inercial: diferencia de cuaterniones	88
5.3.4.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: "ZYZ"). Algoritmo de estimación de orientación: filtro propietario. Algoritmo de evaluación inercial: TRIAD.....	91
5.3.5.- IMUs MATRIX (orden de obtención de los Ángulos de Euler: "ZXZ"). Algoritmo de estimación de orientación: imufilter. Algoritmo de evaluación inercial: diferencia de cuaterniones.....	94
5.4.- ERROR TEÓRICO GENERADO DURANTE LA ESTIMACIÓN INERCIAL MEDIANTE DATOS SINTÉTICOS	97
5.4.1.- Simulación de los errores de alineación sensor – segmento anatómico	97
5.4.2.- Introducción de las referencias de rotación articular.....	98
5.4.3.- Curva teórica de error obtenida.....	98
5.4.4.- Superficie teórica de error generada	102
5.4.5.- Introducción de los errores de alineación en los dos ejes cartesianos restantes	104
6. DISCUSIÓN	108
7. CONCLUSIONES	111
8. PLANIFICACIÓN TEMPORAL.....	112
8.1.- DIAGRAMA DE GANTT ASOCIADO AL PROYECTO	116
9. LÍNEAS DE TRABAJO FUTURAS	118
9.1.- POTENCIALES SECTORES VINCULADOS A LA EVALUACIÓN METROLÓGICA DE SENSORES INERCIALES.....	119
10. REFERENCIAS BIBLIOGRÁFICAS	120
11. ANEXO A: FUNDAMENTOS TEÓRICOS LIGADOS A LAS TÉCNICAS DE CALIBRACIÓN SENSORIAL.....	128
11.1.- DEFINICIÓN FORMAL DEL PROCESO DE CALIBRACIÓN	128
11.2.- TIPOS DE CALIBRACIÓN	129
11.2.1.- Categorías de calibración existentes en función de la naturaleza intrínseca a los parámetros de ajuste.....	129
A.- Calibración de la sensibilidad s y el offset o	130
B.- Calibración de la alineación ϕ	132
11.2.2.- Tipos de calibración vinculados a la fase actual del proceso de ajuste.....	133
A.- Calibración asociada al sistema de referencia del sensor.....	133
B.- Ajuste paramétrico ligado al encapsulado del dispositivo electrónico.....	133
C.- Etapa de calibración correspondiente a la aplicación final del sensor.....	134
11.3.- ESTRUCTURA MATEMÁTICA DE LA CALIBRACIÓN TRIDIMENSIONAL (3D).....	134
11.4.- ALGORITMOS AVANZADOS DE CALIBRACIÓN SENSORIAL	138

12. ANEXO B: ALGORITMIA VINCULADA AL FILTRO DE KALMAN. ESTIMACIÓN DE LA ORIENTACIÓN.....	139
12.1.- ¿EN QUÉ CONSISTE EL <i>FILTRO DE KALMAN</i> ?	139
12.2.- ANATOMÍA DEL MÉTODO ITERATIVO DE <i>KALMAN</i>	139
12.2.1.- Etapa de predicción del estado $\mu_k k - 1$ asociado al sistema	140
12.2.2.- Fase de corrección/actualización.....	141
12.2.3.- Interpretación del Filtro de Kalman como una Caja Negra	143
12.3.- ESTIMACIÓN DE LA ORIENTACIÓN ESPACIAL MEDIANTE EL <i>FILTRO DE KALMAN</i> . TÉCNICA DE FUSIÓN SENSORIAL	143
13. ANEXO C: FUNDAMENTO ALGORÍTMICO LIGADO AL MÉTODO DE FUSIÓN SENSORIAL <i>ECOMPASS</i>.....	149
13.1.- INTRODUCCIÓN.....	149
13.2.- CONFIGURACIÓN GEOMÉTRICA DEL PROBLEMA PLANTEADO	149
13.3.- LECTURA DEL ACELERÓMETRO PRESENTE EN EL SENSOR INERCIAL	151
13.3.- LECTURA DEL MAGNETÓMETRO EXISTENTE EN LA UNIDAD DE MEDIDA INERCIAL	153
13.4.- NORMALIZACIÓN DE LA MATRIZ DE ROTACIÓN <i>R</i>	155
14. ANEXO D: MODELIZACIÓN DE LA ORIENTACIÓN ESPACIAL MEDIANTE CUATERNIONES.....	157
14.1.- ¿QUÉ SON LOS CUATERNIONES?.....	157
14.1.1.- Principales propiedades asociadas a los cuaterniones	158
14.2.- ¿POR QUÉ SE UTILIZAN LOS CUATERNIONES PARA EXPRESAR LA ORIENTACIÓN TRIDIMENSIONAL (3D)?	158
14.3.- FORMULACIÓN MATEMÁTICA DE UN CUATERNIÓN	159
14.3.1.- Producto de dos cuaterniones	160
14.4.- CÁLCULO DE ROTACIONES MEDIANTE CUATERNIONES. TEOREMA.....	160
14.4.1.- Demostración del teorema anterior.....	161
15. ANEXO E: CÓDIGO FUENTE DESARROLLADO DURANTE LA FASE PREVIA A LA CALIBRACIÓN METROLÓGICA DE LOS SENSORES INERCIALES	166
15.1.- <i>IMU XSENS DOT7</i>	167
15.1.1.- Valores máximos de las señales de aceleración	167
15.1.2.- Valores medios de las señales de aceleración.....	167
15.1.3.- Representación gráfica de las señales de aceleración	167
15.2.- <i>IMU MATA1002195</i>	170
15.2.1.- Valores máximos de las señales de aceleración	170
15.2.2.- Valores medios de las señales de aceleración.....	171
15.2.3.- Representación gráfica de las señales de aceleración	171
16. ANEXO F: CÓDIGO FUENTE CONSTRUIDO PARA COMPROBAR EL ESTADO DE ORTOGONALIDAD INTRÍNSECO AL <i>FRAME</i> CARTESIANO DE CADA <i>IMU</i>	174
16.1.- MEDIDA EXPERIMENTAL	175
16.1.1.- Importación de datos.....	175
16.1.2.- Representación gráfica de los datos de aceleración (<i>Acelerómetro del IMU</i>).....	176

16.2.- RESOLUCIÓN DEL SISTEMA DE ECUACIONES LINEALES RELATIVO AL MODELO DE CALIBRACIÓN. MÉTODO DE MÍNIMOS CUADRADOS.....	180
16.2.1.- Vector $[y]$ de variables dependientes	181
16.2.2.- Matriz de coeficientes $[A]$ del sistema lineal de ecuaciones	182
16.2.3.- Vector de incógnitas θ . Obtención de los parámetros de calibración a través de Mínimos Cuadrados	182
16.3.- FUNCIONES	184
16.3.1- Sistema de ecuaciones no lineales utilizado durante la calibración. Calibración(x).....	184
17. ANEXO G: CÓDIGO FUENTE DESARROLLADO PARA CORREGIR LA DESALINEACIÓN EXISTENTE EN EL PAR SENSOR – SEGMENTO ANATÓMICO	185
17.0.- MATERIAL DE LABORATORIO.....	187
17.1.- IMPORTACIÓN DE DATOS.....	188
17.1.1.- Movimientos realizados.....	189
17.2.- ENSAMBLAJE DE LA MATRIZ B . VALORES CRUDOS DE ACELERACIÓN	190
17.3.- CREACIÓN DE LA MATRIZ INHERENTE A LOS VALORES CALIBRADOS (TEÓRICOS) DE LA ACELERACIÓN DE LA GRAVEDAD.....	193
17.4.- CÓMPUTO DE LA MATRIZ DE ROTACIÓN $[R]$	193
17.5.- REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS OBTENIDOS	195
17.6.- VERIFICACIÓN RELATIVA A LA MATRIZ DE ROTACIÓN $[R]$. ¿ $[R]$ PERTENECE AL GRUPO ORTOGONAL ESPECIAL $SO(3)$?.....	198
17.6.1.- Traspuesta de la matriz de rotación $[R]$. ¿ $RTR = I$?.....	198
17.6.2.- Determinante de la matriz de rotación $[R]$. ¿ $[R] = 1$?	199
17.6.3.- Matriz inversa $R - 1$. ¿ $RT = R - 1$?	200
17.6.4.- Conclusiones	201
18. ANEXO H: CÓDIGO FUENTE IMPLEMENTADO PARA EFECTUAR LA ESTIMACIÓN INERCIAL DE UNA ROTACIÓN ARTICULAR.....	202
18.0.- MATERIALES DE LABORATORIO	204
18.1.- IMPORTACIÓN DE LOS DATOS CAPTURADOS.....	206
18.1.1.- Medidas de los acelerómetros	206
18.1.2.- Medidas del giroscopio	207
18.2.- ESTIMACIÓN DE LA ORIENTACIÓN CORRESPONDIENTE A CADA IMU . FUSIÓN SENSORIAL.....	208
18.3.- ALGORITMO IMPLEMENTADO PARA LA ESTIMACIÓN DEL ÁNGULO DE FLEXIÓN ($3D$) ASOCIADO AL CODO	210
18.3.1.- $BSqF0$	210
18.3.2.- $GBqF$	210
18.3.3.- $BSqA0$	211
18.3.4.- $GBqA$	211
18.3.5.- $BqAF$	211
18.3.6.- Representación gráfica de los parámetros: knee flexion ψ_{FE} , knee abduction θ_{AA} y knee internal rotation ϕ_{IE}	212
18.4.- INCORPORACIÓN DE LOS ERRORES DE ALINEACIÓN $SEGMENTO$ -SENSOR. MÉTODO 1.....	213
18.5.- INCORPORACIÓN DE LOS ERRORES DE ALINEACIÓN $SEGMENTO$ -SENSOR. MÉTODO 2.....	217
18.5.1.- Cálculo y representación de la señal de error durante la estimación del ángulo de abducción del codo	223
18.6.- COMPARATIVA DE VARIOS CASOS DE ESTUDIO. OBSERVACIÓN DEL ERROR GENERADO DURANTE LA ESTIMACIÓN DE LA ROTACIÓN ARTICULAR.....	227
18.7.- INCORPORACIÓN DE LOS ERRORES DE ORIENTACIÓN DEL IMU	230

18.7.1.- $GSqA'$	230
18.7.2.- $GSqF'$	231
18.7.3.- $BqAFest2$	232
18.7.4.- Obtención de los Ángulos de Euler asociados al cuaternión $BqAFest2$	233
18.8.- VALORACIÓN FINAL DE LOS ERRORES DE ESTIMACIÓN DE LA ORIENTACIÓN	236
18.8.1.- $BqAFref$	236
18.8.2.- $\phi ERROR$	237
18.8.3.- $\theta ERROR$	237
18.8.4.- $\psi ERROR$	237

ÍNDICE DE FIGURAS

<i>Figura 1.1.- Representación esquemática del problema planteado en este proyecto. [Elaboración propia].</i>	27
<i>Figura 1.2.- Flujograma asociado a la ruta de trabajo establecida para desarrollar el proyecto (Trabajo Fin de Máster). [Elaboración propia].</i>	29
<i>Figura 2.1.- Representación tridimensional de un experimento de flexión simulado computacionalmente. δDESALINEACIÓN = -18 [°]; $\lambda \in 0^\circ, 5^\circ, 10^\circ$. [Elaboración propia].</i>	40
<i>Figura 2.2.- Representación tridimensional de varios experimentos de flexión simulados computacionalmente. Previamente, se ha realizado la siguiente sintonización de parámetros: δDESALINEACIÓN $\in -18^\circ, 0^\circ$; $\lambda \in 0^\circ, 5^\circ, 10^\circ$. [Elaboración propia].</i>	41
<i>Figura 2.3.- Modelo geométrico diseñado para abordar el problema de estimación de los tres parámetros intrínsecos al movimiento del codo: ángulo de flexión ψ_{FE}, abducción ϑ_{AA} y rotación interna ϕ_{IE}.</i>	43
<i>Figura 3.1.- Ubicación de las unidades de medida inercial en el robot colaborativo UR3e. En la fotografía de la izquierda se muestran los sensores de la familia Xsens DOT. En la imagen derecha pueden observarse los IMUs de la gama MATRIX.</i>	47
<i>Figura 3.2.- Ejes motrices existentes en el brazo robótico UR3e. [Elaboración propia].</i>	48
<i>Figura 3.3.- Instrucciones de movimiento programadas en RoboDK para implementar físicamente los protocolos de calibración inercial. A la izquierda se recoge el programa de calibración 3D. En la zona superior derecha se muestra el software de calibración 2D. Finalmente, en la región inferior derecha se representa el modelo CAD utilizado durante el proceso de simulación.</i>	51
<i>Figura 4.1.- Sistema cartesiano de referencia adoptado por los dispositivos Xsens DOT durante la captura de datos. Imagen extraída como cortesía de [36].</i>	58
<i>Figura 4.2.- Especificaciones geométricas de los dispositivos Xsens DOT (versión v2). Plano extraído como cortesía de [36].</i>	59

<i>Figura 4.3.- Imagen de ejemplo relativa a una unidad de medida inercial MATRIX. [Elaboración propia].</i>	61
<i>Figura 4.4.- Actividad de manufactura mecánica, la cual se ha automatizado mediante la utilización del robot colaborativo UR3e. Esta imagen ha sido extraída como cortesía del sitio web oficial ligado al fabricante Universal Robots [1].</i>	63
<i>Figura 4.5.- Fragmento del código Python desarrollado para realizar una conexión con el software RoboDK y el robot físico UR3e. [Elaboración propia].</i>	67
<i>Figura 5.1.- Componentes cartesianas de la aceleración de la gravedad, registradas mediante el sensor inercial Xsens DOT7.</i>	70
<i>Figura 5.2.- Componentes cartesianas de la aceleración de la gravedad, registradas mediante el sensor inercial MATA1002195; el cual pertenece a la gama MATRIX.</i>	72
<i>Figura 5.3.- Evolución temporal de las componentes cartesianas asociadas a la aceleración de la gravedad, las cuales se han registrado mediante el dispositivo de medida inercial Xsens DOT 7.</i>	75
<i>Figura 5.4.- Evolución temporal de las componentes cartesianas asociadas a la aceleración de la gravedad, las cuales se han registrado mediante el dispositivo de medida inercial MATA1002662.</i>	77
<i>Figura 5.5.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. Nota: se muestran las tres componentes cartesianas de g, capturadas mediante el dispositivo Xsens DOT 6 y calibradas mediante un protocolo 2D.</i>	79
<i>Figura 5.6.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. Nota: se muestran las tres componentes cartesianas de g, capturadas mediante el dispositivo Xsens DOT 8 y calibradas mediante un protocolo 3D.</i>	80
<i>Figura 5.7.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. Nota: se muestran las tres componentes cartesianas de g, capturadas mediante el dispositivo MATA 1002195 (perteneciente a la gama MATRIX) y calibradas mediante un protocolo 2D.</i>	81
<i>Figura 5.8.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. Nota: se muestran las tres componentes cartesianas de g, capturadas</i>	

- mediante el dispositivo MATA 100262 (perteneciente a la gama MATRIX) y calibradas mediante un protocolo 3D..... 82
- Figura 5.9.- Estimación inercial del movimiento articular realizado con el cobot UR3e, la cual se ha expresado mediante sus tres Ángulos de Euler. NOTA: Este resultado se corresponde con el conjunto “A” de datos experimentales..... 83
- Figura 5.10.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción θ° , flexión λ° y rotación interna ϕ° . NOTA: Este resultado se corresponde con el conjunto “A” de datos experimentales..... 84
- Figura 5.11.- Estimación inercial del movimiento articular realizado con el cobot UR3e, la cual se ha expresado mediante sus tres Ángulos de Euler (orden de reconstrucción “ZXZ”). NOTA: Se ha utilizado el algoritmo “imufilter” para calcular la orientación de cada IMU Xsens DOT..... 86
- Figura 5.12.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción θ° , flexión λ° y rotación interna ϕ° . NOTA: Se han utilizado los IMUs Xsens DOT y el algoritmo “imufilter” durante la estimación de la orientación..... 87
- Figura 5.13.- Estimación inercial del movimiento articular realizado con el cobot UR3e, la cual se ha expresado mediante sus tres Ángulos de Euler (orden de reconstrucción “ZXZ”). NOTA: Se ha utilizado el algoritmo “ecompass” para calcular la orientación de cada IMU Xsens DOT. El método de evaluación inercial ha sido la “diferencia de cuaterniones”... 89
- Figura 5.14.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción θ° , flexión λ° y rotación interna ϕ° . NOTA: Se han utilizado los IMUs Xsens DOT junto con los algoritmos “ecompass” y “diferencia de cuaterniones”..... 90
- Figura 5.15.- Estimación inercial del movimiento articular realizado con el cobot UR3e, la cual se ha expresado mediante sus tres Ángulos de Euler (orden de reconstrucción “ZYZ”). NOTA: Se ha utilizado el algoritmo “triad” para calcular el ángulo de flexión del codo. 92
- Figura 5.16.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento

articular: abducción θ° , flexión λ° y rotación interna ϕ° . NOTA: Se han utilizado los IMUs Xsens DOT y el algoritmo “TRIAD” durante la estimación de los tres ángulos de rotación.
..... 93

Figura 5.17.- Estimación inercial del movimiento articular realizado con el cobot UR3e, la cual se ha expresado mediante sus tres Ángulos de Euler. NOTA: Se ha utilizado el algoritmo “imufilter” para calcular la orientación de cada IMU..... 95

Figura 5.18.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción θ° , flexión λ° y rotación interna ϕ° . NOTA: Este resultado se corresponde con los IMUs MATRIX y el algoritmo imufilter, empleado durante la estimación de la orientación..... 96

Figura 5.19.- Representación gráfica del error de estimación (teórico) en función del ángulo de desalineación incorporado. Nota: $\lambda = 70^\circ$ 99

Figura 5.20.- Representación gráfica del error de estimación (teórico) en función del ángulo de desalineación incorporado. Nota: $\lambda = 60^\circ$ 101

Figura 5.21.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. Nota: $\delta_{DESALINEACIÓN} \equiv \phi_{ERRORARM} \in -20^\circ, 20^\circ$... 103

Figura 5.22.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. Nota: $\delta_{DESALINEACIÓN} \equiv \theta_{ERRORARM} \in -20^\circ, 20^\circ$.. 105

Figura 5.23.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. Nota: $\delta_{DESALINEACIÓN} \equiv \psi_{ERRORARM} \in -20^\circ, 20^\circ$... 107

Figura 6.1.- Fotografía de los IMUs Xsens DOT durante los dos experimentos iniciales realizados. [Elaboración propia]..... 109

Figura 8.1.- Planificación temporal, en forma de Diagrama de Gantt, que se ha diseñado para el desarrollo de este Trabajo Fin de Máster. [Elaboración propia]. 117

Figura 11.1.- Representación esquemática del procedimiento diseñado para efectuar la correcta calibración de un acelerómetro. [Elaboración propia]. 131

Figura 11.2.- Modelo geométrico diseñado para estimar el valor de la desalineación ϕ . Esta información será empleada durante la posterior calibración del sensor. [Elaboración propia]..... 132

Figura 11.3.- Representación esquemática de la disposición geométrica considerada durante la metodología de calibración tridimensional. [Elaboración propia]. 137

Figura 12.1.- Representación conceptual del Filtro de Kalman como una Caja Negra. [Elaboración propia]. 143

Figura 12.2.- Arquitectura diseñada (Filtro de Kalman) para estimar la orientación espacial de un sólido rígido. [Elaboración propia]..... 147

Figura 13.1.- Disposición espacial de los sistemas de coordenadas NED y cartesiano, vinculados a una posición 3D genérica. [Elaboración propia]..... 150

Figura 13.2.- Transformación geométrica diseñada para alinear el eje Z del frame cartesiano vinculado al sensor y el sistema de referencia NED. [Elaboración propia]. . 152

Figura 13.3.- Corte realizado en el plano N – D, con la finalidad de representar gráficamente la disposición espacial ligada a los sistemas de coordenadas $\{X, Y, Z\}$ y $\{N, E, D\}$. Adicionalmente, puede visualizarse el resultado generado para el vector de campo magnético m . [Elaboración propia]..... 153

Figura 13.4.- Representación esquemática de la rotación necesaria para alinear el sistema de referencia $\{N, E, D\}$ con el frame cartesiano $\{X, Y, Z\}$; intrínseco al sensor. [Elaboración propia]..... 154

Figura 16.1.- Componentes cartesianas de la aceleración de la gravedad, capturadas durante el procedimiento de ortogonalización. Obsérvese que la información de interés se corresponde con las diferentes posiciones estáticas definidas (véanse las variables g_x y g_y). 180

Figura 17.1.- Sistema de coordenadas vinculado a las unidades de medida inercial de la gama Xsens DOT. Imagen extraída como cortesía de [36]. 187

Figura 17.2.- Vista isométrica del montaje experimental elaborado en el robot colaborativo UR3e, disponible en el RoboLab de la EPI Gijón..... 187

Figura 17.3.- Vista frontal del montaje experimental elaborado en el robot colaborativo UR3e, disponible en el RoboLab de la EPI Gijón..... 188

Figura 17.4.- Movimientos experimentados por el IMU Xsens DOT 7, los cuales han sido programados mediante el cobot UR3e. 189

Figura 17.5.- Representación gráfica de los datos crudos vinculados a la aceleración de la gravedad, junto con sus correspondientes valores calibrados. 198

Figura 18.1.- Dispositivo experimental empleado durante la captura de datos..... 204

Figura 18.2.- Vista adicional sobre el medidor de ángulos utilizado durante la experiencia de laboratorio. 205

Figura 18.3.- Sistema de coordenadas cartesianas asociado a los sensores inerciales de la familia Xsens DOT. 205

ÍNDICE DE TABLAS

<i>Tabla 1.1.- Datos inerciales y algoritmos disponibles para calcular la orientación de un IMU.....</i>	<i>23</i>
<i>Tabla 1.2.- Alternativas existentes en la estimación inercial del movimiento articular.</i>	<i>24</i>
<i>Tabla 3.1.- Variables establecidas durante la calibración inercial de los dispositivos pertenecientes a la familia Xsens DOT. Esta configuración ha sido realizada mediante la aplicación oficial de Android suministrada por el fabricante Xsens.</i>	<i>48</i>
<i>Tabla 3.2.- Variables establecidas durante la calibración inercial de los dispositivos pertenecientes a la familia MATRIX. Dicha configuración se ha definido a través de la app de MATRIX disponible en Android (versión 1.2.7).</i>	<i>49</i>
<i>Tabla 3.3.- Metodología experimental definida para calibrar el sensor inercial colocado en el último eslabón del robot colaborativo UR3e.</i>	<i>49</i>
<i>Tabla 3.4.- Metodología experimental definida para calibrar el sensor inercial colocado en el penúltimo eslabón del robot colaborativo UR3e.....</i>	<i>50</i>
<i>Tabla 3.5.- Metodología experimental definida para evaluar el sensor inercial colocado en el último eslabón del robot colaborativo UR3e. Nota: esta secuencia de movimientos debe repetirse para cada ángulo de desalineación $\delta_{DESALINEACIÓN} \equiv \delta$.....</i>	<i>54</i>
<i>Tabla 3.6.- Metodología experimental definida para evaluar el sensor inercial colocado en el penúltimo eslabón del robot colaborativo UR3e. Nota: téngase presente que todos los ángulos de flexión λ [°] se evaluarán para cada desalineación introducida $\delta_{DESALINEACIÓN}$ [°].....</i>	<i>55</i>
<i>Tabla 3.7.- Síntesis de la metodología experimental diseñada para la evaluación inercial de los IMUs. Nota: obsérvese que esta tabla es válida para todos los ángulos de desalineación “sensor – segmento” propuestos $\delta_{DESALINEACIÓN} \in -20^\circ, -18^\circ, \dots, 0^\circ, \dots, 18^\circ, 20^\circ$.</i>	<i>56</i>
<i>Tabla 4.1.- Características propias del acelerómetro que posee un IMU Xsens DOT.....</i>	<i>59</i>
<i>Tabla 4.2.- Características asociadas al giróscopo que posee un IMU Xsens DOT.....</i>	<i>60</i>
<i>Tabla 4.3.- Características ligadas al magnetómetro que posee un IMU Xsens DOT.</i>	<i>60</i>

<i>Tabla 4.4.- Características técnicas que gobiernan la funcionalidad de un sensor inercial MATRIX. Estos datos han sido obtenidos mediante la información referenciada en [38].</i>	62
<i>Tabla 4.5.- Principales especificaciones mecánicas del brazo robótico UR3e.</i>	63
<i>Tabla 4.6.- Características cinemáticas del mecanismo articulado UR3e.</i>	64
<i>Tabla 4.7.- Propiedades relativas al subsistema eléctrico del brazo robótico UR3e.</i>	64
<i>Tabla 4.8.- Características físicas del robot colaborativo UR3e.</i>	65
<i>Tabla 4.9.- Prestaciones mecánicas suministradas por el brazo robótico UR3e.</i>	66
<i>Tabla 5.1.- Desviaciones típicas vinculadas a las medidas proporcionadas por los sensores inerciales. Cada captura ha tenido una duración de 1 minuto y los IMUs se han colocado en una posición estática.</i>	68
<i>Tabla 5.2.- Muestra de los datos capturados mediante el IMU Xsens DOT7, perteneciente a la familia de sensores Xsens DOT. Obsérvese que estos datos se corresponden con el protocolo experimental diseñado para comprobar la ortogonalidad existente en el frame cartesiano de dicho sensor inercial.</i>	74
<i>Tabla 5.3.- Muestra de los datos capturados mediante el IMU MATA1002662, perteneciente a la familia de sensores MATRIX. Obsérvese que estos datos se corresponden con el protocolo experimental diseñado para comprobar la ortogonalidad existente en el frame cartesiano de dicho sensor inercial.</i>	76
<i>Tabla 5.4.- Valores numéricos (media) obtenidos como resultado de los experimentos de calibración inercial llevados a cabo.</i>	78
<i>Tabla 5.5.- Valores numéricos (desviación típica) obtenidos como resultado de los experimentos de calibración inercial llevados a cabo.</i>	78
<i>Tabla 5.6.- Errores generados durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (EXP_1, Xsens DOT, filtro propietario, diferencia de cuaterniones).</i>	85
<i>Tabla 5.7.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (EXP_1, Xsens DOT, imufilter, diferencia de cuaterniones).</i>	88

<i>Tabla 5.8.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (EXP_2, Xsens DOT, ecompass, diferencia de cuaterniones).</i>	91
<i>Tabla 5.9.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (EXP_3, Xsens DOT, imufilter, TRIAD).</i>	94
<i>Tabla 5.10.- Errores generados durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (EXP_4, MATRIX, imufilter, diferencia de cuaterniones).</i>	97
<i>Tabla 5.11.- Resultados numéricos obtenidos durante el tratamiento estadístico de los datos teóricos, correspondientes a los errores de estimación inercial para $\lambda = 10^\circ$.</i>	100
<i>Tabla 5.12.- Resultados numéricos obtenidos durante el tratamiento estadístico de los datos teóricos, correspondientes a los errores de estimación inercial para $\lambda = 60^\circ$.</i>	101
<i>Tabla 8.1.- Desglose de las tareas que se han definido para desarrollar el presente Trabajo Fin de Máster.</i>	112
<i>Tabla 16.1.- Fragmento de los datos capturados mediante el sensor inercial. Recuérdese que se persigue cuantificar el nivel de ortogonalidad asociado al sistema de referencia interno, existente en dicho dispositivo.</i>	175
<i>Tabla 17.1.- Fragmento de los datos inerciales capturados mediante el dispositivo Xsens DOT 7. Obsérvese que se ha registrado un total de 1.492 muestras de datos; relativos a la orientación espacial (cuaternión), la aceleración lineal y la velocidad angular de giro que posee el IMU monitorizado.</i>	188

CAPÍTULO 1.

MEMORIA

1. Introducción

En la actualidad, la tecnología sensorial se encuentra presente en multitud de sectores; tanto dentro del ámbito doméstico como del industrial y/o de servicios. El presente proyecto se ha centrado en investigar e implementar la **estimación de la orientación** a partir de **unidades de medida inercial**. Concretamente, tras efectuar la correspondiente calibración de los dispositivos de medida, se evaluarán inercialmente varias rotaciones articulares.

Se contrastarán los resultados obtenidos con **distintas familias de sensores inerciales**. Asimismo, se explorará el efecto inherente a los **diversos algoritmos de estimación de la orientación** empleados. El detalle de dichos elementos *hardware* y *software* se describe en posteriores apartados de este documento.

El estudio llevado a cabo permitirá modelar teórica y experimentalmente la influencia de los errores de alineación (vinculados a los respectivos sensores) sobre la estimación inercial final que se ha realizado. Se ha realizado la cuantificación de los errores ligados a cada estimación inercial, así como en obtener una distribución estadística que posibilite ajustar dichos datos de error.

Con objeto de automatizar las metodologías experimentales diseñadas, se ha utilizado el **robot colaborativo UR3e** [1]; disponible en el *RoboLab* de la *EPI* de Gijón. Durante la etapa correspondiente al tratamiento y explotación de los datos adquiridos en las experiencias de laboratorio, se ha hecho uso del *software MatLab R2022b* [2].

1.1.- DEFINICIÓN DEL PROBLEMA

El eje central del presente proyecto de investigación se entronca en torno al estudio inercial de las rotaciones articulares en el espacio (tridimensional). Se estudiará el efecto que poseen sobre el resultado final las unidades de medida inercial empleadas.

En concreto, se pretende **analizar la influencia que poseen los errores de alineación¹** sobre la estimación inercial final. Para desarrollar dicha actividad, se han contemplado **tres algoritmos diferentes destinados a la estimación de la orientación**, vinculados a sendas **familias de sensores inerciales: Xsens DOT y MATRIX**. El detalle de los métodos computacionales definidos puede consultarse en posteriores secciones de esta *Memoria*.

La cuantificación numérica de los errores de estimación, cometidos con cada gama de sensores, permitirá deducir qué dispositivo de medida inercial resulta más adecuado para la aplicación biomecánica en estudio.

Adicionalmente, todo el estudio desarrollado permitirá modelar la fiabilidad de los resultados que serán proporcionados por las estimaciones inerciales implementadas.

1.1.1.- Unidades de medida inercial y algoritmos estudiados

Tal y como se ha indicado anteriormente, en el presente proyecto se han escogido las siguientes familias de sensores inerciales:

- *IMUs Xsens DOT*. Este tipo de dispositivos de medida inercial tienen incorporado un microcontrolador, el cual ya dispone de varios algoritmos de estimación de la orientación implementados. Concretamente, se ha seleccionado el *Filtro de Kalman²* durante la obtención de los **cuaterniones de orientación**. De esta forma, en esta gama de sensores el algoritmo de estimación inercial se *alimentará* directamente con los cuaterniones suministrados por los dispositivos *Xsens DOT*. Por otra parte, también resulta factible utilizar los algoritmos

¹ **Nota:** Téngase en cuenta que los errores de alineación están referidos a los sistemas de referencia que se ubican, respectivamente, en el sensor inercial utilizado y en el correspondiente segmento del sólido rígido analizado.

² Recuérdese que el *Algoritmo de Kalman* posee *memoria* sobre cuál es el estado anterior del sistema controlado. Durante la obtención de cada cuaternión de orientación, dicho método computacional emplea la respectiva muestra de la siguiente triada de datos: *Aceleración + Intensidad de Campo Magnético + Velocidad Angular de Giro*.

Puede consultarse información adicional sobre este algoritmo a través de la información recogida en el apartado “*Anexo B: Algoritmia vinculada al Filtro de Kalman. Estimación de la orientación*” del presente documento.

imufilter (*input*: aceleración + velocidad angular de giro) y *ecompass*³ (*input*: aceleración + intensidad de campo magnético) durante la obtención de los respectivos cuaterniones de orientación.

- *IMUs MATRIX*. Dado que estas unidades de medida inercial solamente están provistas de acelerómetros y giroscopios, se ha decidido utilizar el **algoritmo *imufilter*** para estimar la orientación espacial de cada dispositivo. Por ende, los datos inerciales proporcionados al algoritmo de estimación inercial se corresponden con la aceleración y la velocidad angular de giro del sólido rígido monitorizado.

A continuación, se muestran las posibilidades existentes para estimar la orientación de un sensor inercial:

Tabla 1.1.- Datos inerciales y algoritmos disponibles para calcular la orientación de un *IMU*.

IMU	Acelerómetro	Giróscopo	Magnetómetro	Filtro	ALGORITMO
	X	X	X	X	<i>Filtro propietario</i>
Xsens	X		X	X	<i>TRIAD</i>
DOT	X	X			<i>imufilter</i>
	X		X		<i>ecompass</i>
MATRIX	X	X			<i>imufilter</i>

Con objeto de estimar de estimar inercialmente varias rotaciones articulares, se dispone de las siguientes posibilidades:

³ A través del apéndice “*Anexo C: Fundamento algorítmico ligado al método de fusión sensorial *ecompass*”*, pueden obtenerse datos adicionales sobre el método de fusión sensorial *ecompass*. Téngase presente que dicho algoritmo no tiene *memoria* sobre el estado anterior del sistema.

Por otro lado, la algoritmia de *ecompass* precisa conocer (en cada instante de tiempo analizado) los datos de *Aceleración e Intensidad de Campo Magnético*; vinculados al sólido rígido en estudio.

Tabla 1.2.- Alternativas existentes en la estimación inercial del movimiento articular.

Dispositivos	ORIENTACIÓN DE CADA IMU						ESTIMACIÓN INERCIAL DE ROTACIÓN	
	q	\ddot{y}	$\dot{\theta}$	m	imufilter	ecompass	Diferencia de cuaterniones [3]	TRIAD [4]
	X						EXP_1	
Xsens DOT #1	X	X			X		EXP_1	
Xsens DOT #2	X			X		X	EXP_2	
		X		X				EXP_3
MATRIX #1	--	X	X	--	X	--	EXP_4	--
MATRIX #2	--	X	X	--	X	--	EXP_4	--

#1 y #2 designan a dos dispositivos de la misma familia de sensores inerciales.
 \mathbf{q} designa al cuaternión de orientación calculado por el filtro interno del respectivo *IMU*.
 $\ddot{y} \equiv$ aceleración registrada por el *IMU*, $\dot{\theta} \equiv$ velocidad angular de giro, $\mathbf{m} \equiv$ intensidad de campo magnético.

1.2.- REVISIÓN TÉCNICA (ESTADO DEL ARTE)

Actualmente, existen varios estudios en la literatura sobre la evaluación de dispositivos de estimación de la orientación basados en sensores inerciales; como, por ejemplo, las publicaciones referenciadas en [5] y [6]. Desde el punto de vista de las ciencias biomédicas (deporte, rehabilitación, análisis de la marcha, gerontología, neurociencias), es necesario evaluar la precisión de esta tecnología cuando se aplica a diferentes escenarios del movimiento humano, como revelan dos revisiones sistemáticas [7][8]. Estos estudios dejan claro que las características técnicas (como la calibración interna, el tipo de sensores, la colocación y fijación del sensor o el método de estimación) son determinantes, pero en la aplicación clínica también son relevantes otros dos factores: la complejidad de la articulación y la complejidad del movimiento. Los sensores que se evalúan cuando están fijados al cuerpo humano también dificultan la obtención de conjuntos de datos suficientemente grandes y condiciones experimentales controladas de forma coherente.

Las unidades de medida inercial constituyen una solución interesante para el análisis de la marcha humana. En la actualidad, se han realizado numerosos estudios en este tipo de aplicaciones biomédicas, tales como la literatura referenciada en [9] y [10]. En este escenario, los algoritmos centrados en la fusión de información sensorial cobran especial relevancia [11]-[14].

En la actualidad, muchos de los algoritmos destinados a estimar la orientación tridimensional de un sólido rígido requieren la explotación de datos vinculados a la intensidad del campo magnético. El cuello de botella de esta tecnología son las interferencias electromagnéticas que pueden estar presentes en la zona de estudio. Por este motivo, diversas investigaciones [15][16] están centrando sus esfuerzos en aumentar la robustez de los algoritmos implementados en los *IMUs* frente a la existencia de dicho ruido electromagnético.

Como paso previo a cualquier aplicación biomecánica, es necesario diseñar e implementar un protocolo de calibración vinculado a los sensores inerciales que se van a utilizar. Actualmente, existe abundante literatura sobre técnicas de calibración sensorial [17]-[21]. Esta revisión técnica se ha centrado en metodologías experimentales vinculadas a unidades de medida inercial. En particular, las aplicaciones ligadas a la estimación inercial de la rotación de una articulación requieren dos tipos de calibración: la intrínseca a la orientación del *IMU* y la derivada de la alineación *sensor-segmento*. En [22] J. Favre et al. proponen un método de calibración basado en la estimación de dos cuaterniones (\vec{R}_A y \vec{R}_B), que permiten rotar los *frames* cartesianos ligados a cada sensor inercial; de forma que coincidan con los sistemas de referencia intrínsecos al segmento anatómico correspondiente. Mediante un protocolo de calibración en dos pasos, este algoritmo permite obtener los *frames* $\{\vec{i}, \vec{j}, \vec{k}\}$ correspondientes a cada *IMU* colocado en el brazo o pierna.

Por otro lado, en [23] G. Ligorio et al. han diseñado un método de calibración basado en el registro de varias velocidades angulares de rotación. En primer lugar, se han realizado varios movimientos de flexión/extensión y pronación/supinación del codo; capturando las velocidades angulares de giro mediante los respectivos *IMUs* colocados en el brazo y antebrazo del usuario. A continuación, el protocolo experimental pretende determinar la posición relativa del brazo y antebrazo durante diversas posturas estáticas realizadas por la persona objeto de estudio. Finalmente, la base de datos obtenida será explotada mediante un método de cálculo basado en el análisis de vectores propios, tal y como se describe mediante la bibliografía referenciada en [24].

Otros estudios de investigación, relacionados con la calibración sensorial, prestan especial atención en el ajuste del magnetómetro existente dentro del sensor inercial. N. Choe et al. han empleado un algoritmo de calibración magnética consistente en el cálculo de los coeficientes vinculados a un modelo elipsoidal, utilizando para ello el método de mínimos cuadrados durante el proceso de cómputo [25].

1.3.- RESUMEN DEL PROCEDIMIENTO DISEÑADO

En esta sección se recoge la naturaleza consustancial al problema abordado durante este proyecto. De manera particular, el objetivo del presente *TFM* consiste en cuantificar numéricamente las tres componentes cinemáticas (guiñada – *yaw*, cabeceo – *pitch* y balanceo – *roll*) vinculadas al movimiento del codo.

De acuerdo con el apartado anterior, resulta necesario conocer la orientación espacial del brazo y el antebrazo del robot que se está monitorizando. Por este motivo, se ha decidido colocar una unidad de medida inercial en dos eslabones de dicho robot. Para abordar esta actividad, se requiere calcular los cuaterniones de orientación tridimensionales ${}^{GS}q_i$, $\forall i \in \{1, 2\}$; ligados a cada sensor inercial.

Una vez conocidos los cuaterniones de orientación inherentes a cada *IMU*, los ángulos que definen la rotación relativa entre el brazo y el antebrazo pueden calcularse fácilmente mediante operaciones de conjugación y producto de cuaterniones; entre otras estrategias computacionales.

Cabe destacar que en los modelos propuestos para la estimación inercial de la rotación articular, los cual se detallarán posteriormente, se ha considerado la posible presencia de errores en la orientación del *IMU* y en la alineación *sensor - segmento*.

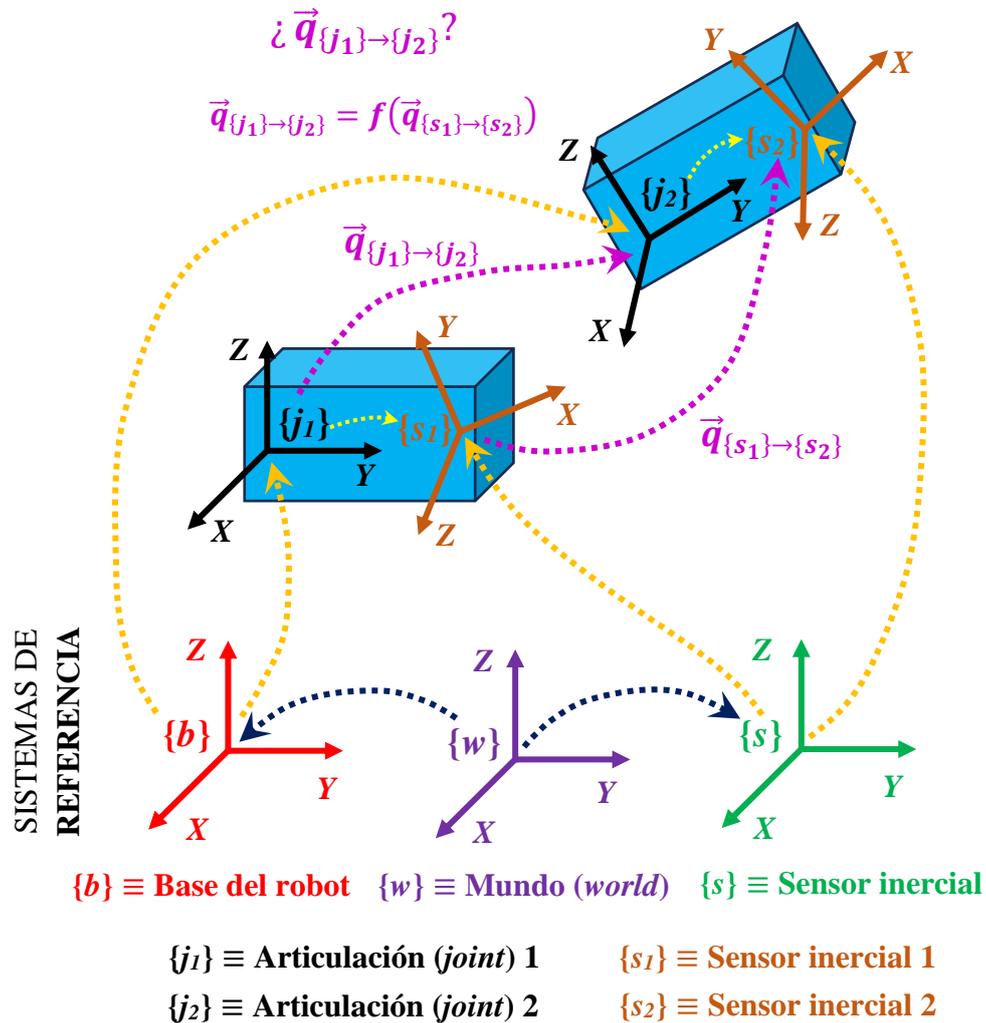


Figura 1.1.- Representación esquemática del problema planteado en este proyecto. [Elaboración propia].

1.4.- SECUENCIA DE HITOS DEFINIDA PARA EL PROYECTO

A través de este apartado, se recoge la filosofía de trabajo seguida durante el desarrollo del presente proyecto.

Como fase inicial, se ha efectuado una revisión sobre la bibliografía existente en materia de calibración de sensores inerciales [26 – 30]. Si bien es cierto que a lo largo de este trabajo se han estudiado varios algoritmos de calibración, se ha prestado mayor atención

en analizar los resultados generados por cada método analítico de trabajo; así como en realizar una comparación estadística entre ellos.

A continuación, se ha procedido a consultar varias referencias bibliográficas relacionadas con la evaluación metrológica de *IMUs* [31 – 35]; incidiendo con mayor énfasis en las potenciales aplicaciones biomecánicas de estos dispositivos de medida inercial.

Tras finalizar las tareas comentadas en los dos párrafos anteriores, ya se está en disposición de definir la experiencia de laboratorio que debe llevarse a cabo. Dicho protocolo experimental se ha diseñado garantizando el máximo rigor posible, con objeto de evitar ambigüedades durante la posterior interpretación estadística de los resultados generados.

El siguiente paso del proyecto ha consistido en la captura de datos, obtenidos durante el desarrollo de los experimentos propuestos. Los sensores utilizados para dicho registro de información han sido, fundamentalmente, unidades de medida inercial (*Inertial Measurement Units – IMUs*).

Tomando como punto de partida la base de datos generada en la experiencia de laboratorio, se han empleado herramientas estadísticas para procesar y analizar⁴ los movimientos capturados a través de los *IMUs*. Seguidamente, los resultados arrojados por el estudio implementado han permitido extraer varias conclusiones; relativas a la evaluación metrológica de los sensores utilizados.

Finalmente, se ha considerado interesante especificar las limitaciones de la solución diseñada; de la misma manera que se proponen algunas posibles alternativas para mejorar los resultados obtenidos durante el presente estudio.

El diagrama de flujo expuesto en la Figura 1.2 ha permitido satisfacer la elaboración del proyecto presentado durante la sección anterior, así como estructurar de forma secuencial las tareas que se deben abordar en el transcurso del *Trabajo Fin de Máster (TFM)*.

⁴ **Nota:** El paquete *software* de cálculo científico seleccionado para desarrollar las tareas de análisis, procesamiento y visualización de datos ha sido *MatLab R2022b*.

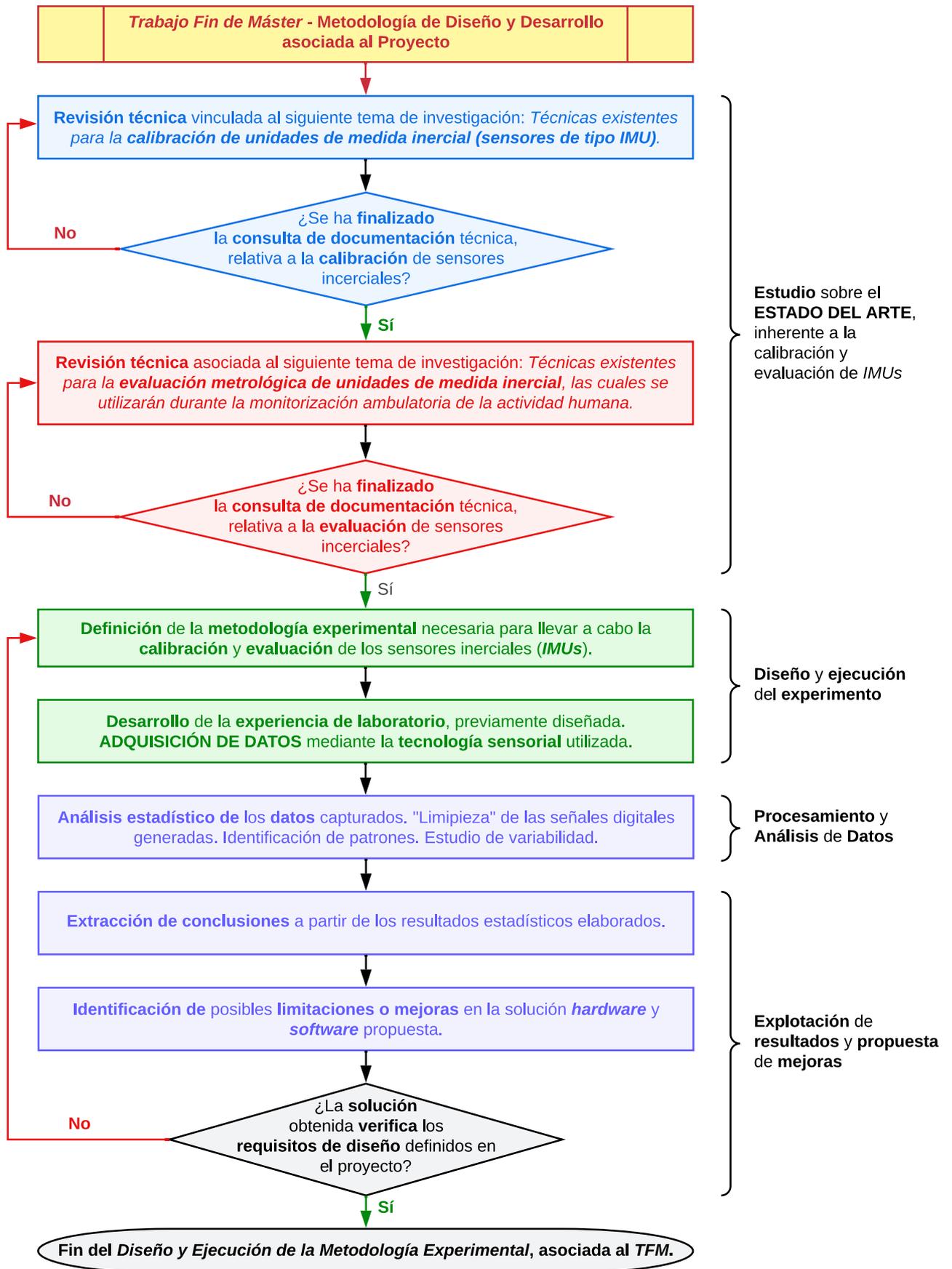


Figura 1.2.- Flujograma asociado a la ruta de trabajo establecida para desarrollar el proyecto (Trabajo Fin de Máster). [Elaboración propia].

2. Marco Teórico

Mediante el actual apartado se exponen varios modelos matemáticos utilizados durante la estimación inercial de rotaciones articulares (*Diferencia de Cuaterniones* [3] y *TRIAD* [4]).

2.1.- MÉTODO COMPUTACIONAL DESTINADO A CUANTIFICAR EL MOVIMIENTO TRIDIMENSIONAL DE UNA ARTICULACIÓN (DIFERENCIA DE CUATERNIONES)

En esta sección, se propone un algoritmo capaz de valorar numéricamente la rotación espacial (3D) experimentada por el codo durante el movimiento de una persona o de un brazo robótico. En este escenario, resulta necesario conocer en tiempo real la orientación tridimensional⁵ de varios sólidos rígidos. Concretamente, la algoritmia diseñada precisa definir dos sólidos rígidos; correspondientes, respectivamente, al brazo y el antebrazo de la persona monitorizada.

Abundando en lo explicado durante el párrafo anterior, se han considerado los cuaterniones principales identificados a continuación:

- ${}^{GB}q$: cuaternión de orientación asociado a un segmento del cuerpo de la persona en estudio ($\{B\}$). Este cuaternión está expresado respecto a un sistema de coordenadas de referencia ($\{G\}$).

⁵ Actualmente, existen numerosas herramientas *software* para modelizar la orientación y la trayectoria 3D de un objeto. En este caso, se ha decidido emplear cuaterniones de orientación; dada su robustez frente a pequeñas variaciones del ángulo de giro θ . Puede consultarse información adicional sobre este tema mediante el anexo del documento *Memoria* titulado como: “Anexo D: Modelización de la orientación espacial mediante cuaterniones”.

- ${}^{GS}q$: cuaternión de orientación vinculado al sensor inercial ($\{S\}$) que se ha colocado sobre el cuerpo del individuo monitorizado ($\{B\}$). Dicho cuaternión está codificado respecto a un sistema de coordenadas de referencia ($\{G\}$).

Particularizando los cuaterniones ${}^{GB}q$ y ${}^{GS}q$ a las diversas situaciones experimentales definidas, se han ido construyendo todas las formulaciones requeridas por el algoritmo propuesto. En primer lugar, debe expresarse la orientación 3D del brazo ($Arm - A$) respecto a un sistema cartesiano de coordenadas; tomado como referencia ($Ground - G$). Para ello, se ha definido el cuaternión ${}^{GB}q_A$; el cual obedece al siguiente producto⁶:

$${}^{GB}q_A = {}^{GS}q_A \otimes {}^{BS}q_{A_0}^* \quad (2.1)$$

Donde:

- ${}^{GS}q_A$ es el cuaternión de orientación correspondiente al *IMU* (*Sensor - S*) colocado en el brazo ($Arm - A$) de la persona. Este cuaternión está codificado respecto a un sistema de coordenadas de referencia ($\{G\}$).
- ${}^{BS}q_{A_0}^*$ denota al cuaternión conjugado⁷ que modela la orientación inicial ($t = 0$), vinculada al sensor inercial fijado en el brazo de la persona. Las 4 componentes de este cuaternión están expresadas respecto al sistema de referencia definido en el brazo ($\{B\}$).

El parámetro ${}^{BS}q_{A_0}^*$, utilizado en la expresión (2.1), puede calcularse aplicando la siguiente ecuación:

$${}^{BS}q_{A_0}^* = {}^{GS}q_{A_0}^* \otimes {}^{GB}q_{A_0} \quad (2.2)$$

⁶ **Nota:** En el presente documento, el operador *producto entre dos cuaterniones* se ha representado mediante el símbolo \otimes . Véase la literatura referenciada en [4].

⁷ A lo largo del texto incluido en este apartado, el operador “*” simboliza al cuaternión conjugado de otro *quaternion* inicial.

A continuación, se detallan los **nuevos términos incorporados en (2.2)**:

- ${}^{GS}q_{A_0}^*$: cuaternión conjugado de orientación inicial ($t = 0$), correspondiente al sensor inercial colocado en el brazo de la persona en estudio. Las 4 componentes de este cuaternión están codificadas respecto al sistema de referencia global ($\{G\}$).
- ${}^{GB}q_{A_0}$: cuaternión de orientación inicial correspondiente a la unidad de medida inercial colocada en el brazo de la persona. Este cuaternión está expresado respecto al sistema de referencia definido en el brazo ($\{B\}$).

De forma análoga al cálculo de orientación realizado en la ecuación (2.1), debe modelarse la **orientación tridimensional del antebrazo** ($F - Forearm$):

$${}^{GB}q_F = {}^{GS}q_F \otimes {}^{BS}q_{F_0}^* \quad (2.3)$$

Notación definida en la ecuación (2.3)

-
-
- ${}^{GS}q_F$: cuaternión de orientación correspondiente al *IMU (Sensor - S)* fijado en el antebrazo ($F - Forearm$) de la persona. Este cuaternión está expresado respecto a un sistema de coordenadas de referencia ($\{G\}$).
 - ${}^{BS}q_{F_0}^*$: cuaternión conjugado que expresa la orientación inicial, asociada al sensor inercial situado en el antebrazo de la persona.
-
-

El cálculo del cuaternión ${}^{BS}q_{F_0}^*$ se ha implementado de acuerdo con la siguiente ecuación:

$${}^{BS}q_{F_0}^* = {}^{GS}q_{F_0}^* \otimes {}^{GB}q_{F_0} \quad (2.4)$$

Notación definida en la ecuación (2.4)

- ${}^{GS}q_{F_0}^*$: cuaternión conjugado de orientación inicial, correspondiente al sensor inercial colocado en el antebrazo de la persona en estudio. Dicho cuaternión está codificado respecto al sistema de referencia global ($\{G\}$).
 - ${}^{GB}q_{F_0}$: cuaternión de orientación inicial, vinculado al *IMU* colocado en el brazo de la persona. Este cuaternión está expresado respecto al sistema de referencia definido en el antebrazo ($\{B\}$).
-
-

Hasta este punto del algoritmo, ya se han calculado los cuaterniones de orientación inicial ${}^{BS}q_{A_0}^*$ y ${}^{BS}q_{F_0}^*$; ligados a sendos *IMUs* colocados en el brazo y el antebrazo de la persona, respectivamente.

El siguiente paso del método algebraico implementado consiste en obtener el cuaternión ${}^Bq_{AF}$, a través del cual se modela el **movimiento tridimensional desarrollado en el codo de la persona**. Esta tarea se ha resuelto mediante la ecuación (2.5), cuya anatomía se detalla a continuación:

$${}^Bq_{AF} = {}^{GB}q_A^* \otimes {}^{GB}q_F \quad (2.5)$$

En la **ecuación (2.5)** se ha incorporado la siguiente **notación**:

- ${}^{GB}q_A^*$: conjugado del cuaternión de orientación correspondiente al brazo (*A – Arm*) de la persona en estudio. Las componentes de ${}^{GB}q_A^*$ están expresadas respecto al sistema de referencia $\{G\}$.

- ${}^{GB}q_F$: cuaternión de orientación ligado al antebrazo ($F - Forearm$) del sujeto monitorizado. Las componentes de ${}^{GB}q_F$ están expresadas respecto al sistema de referencia $\{G\}$.
- Obsérvese que la información espacial almacenada en el cuaternión ${}^Bq_{AF}$ se ha expresado respecto al sistema de referencia vinculado al cuerpo de la persona ($\{B\}$).

A partir de la información contenida en el cuaternión ${}^Bq_{AF}$, pueden obtenerse los tres *Ángulos de Euler* (ψ_{FE} , ϑ_{AA} , ϕ_{IE}) que describen la rotación 3D experimentada por el codo. Por tanto, la siguiente etapa del algoritmo consiste en realizar la transformación matemática enunciada a continuación:

$${}^Bq_{AF} \mapsto \{\psi_{FE}, \vartheta_{AA}, \phi_{IE}\} \quad (2.6)$$

Bajo el supuesto de que el cuaternión indicado en la ecuación (2.6) puede expresarse como ${}^Bq_{AF} = [q_0, q_1, q_2, q_3]^T$, la terna de *Ángulos de Euler* se calcula de la manera que sigue:

- *Obtención del ángulo de flexión del codo (ψ_{FE}) – Elbow Flexion*

$$\psi_{FE} = \text{atan2}(-2q_2q_3 + 2q_0q_1; q_3^2 - q_2^2 - q_1^2 + q_0^2) \quad (2.7)$$

- *Cómputo del ángulo de abducción del codo (ϑ_{AA}) – Elbow Abduction*

$$\vartheta_{AA} = \text{asin}(2q_1q_3 + 2q_0q_2) \quad (2.8)$$

- *Cálculo del ángulo de rotación interna asociado al codo (ϕ_{IE}) – Elbow Internal Rotation*

$$\phi_{IE} = \text{atan2}(-2q_1q_2 + 2q_0q_3; q_1^2 + q_0^2 - q_3^2 - q_2^2) \quad (2.9)$$

2.1.1.- Incorporación de los errores de orientación y alineación sensor – segmento durante el cálculo del cuaternión final de ROTACIÓN ARTICULAR

Las principales dificultades que aparecen durante la obtención de los *Ángulos de Euler* $\{\psi_{FE}, \vartheta_{AA}, \phi_{IE}\}$ consisten en dos fuentes de error, las cuales pueden sesgar la validez de los datos capturados:

- ❖ **Desalineación** existente entre el sensor inercial y el segmento del cuerpo humano que se desea monitorizar.
- ❖ **Orientación errónea** de la unidad de medida inercial, de acuerdo con la naturaleza de la magnitud física que se desea medir.

Las dos formas de error expresadas en los puntos anteriores pueden estudiarse numéricamente mediante el uso de cuaterniones. Inicialmente, deben calcularse las tres componentes asociadas al cuaternión final de orientación (${}^B q_{AF}$); las cuales se corresponden con los movimientos de *Yaw* (Guiñada), *Pitch* (Cabeceo) y *Roll* (Balanceo).

En consonancia con la idea transmitida durante el párrafo anterior, un cuaternión genérico puede calcularse a partir de sus tres componentes de la siguiente forma:

$$q = q_y \otimes q_p \otimes q_r \quad (2.10)$$

Notación introducida en la ecuación (2.10)

-
-
- q_y : cuaternión de orientación correspondiente al movimiento *yaw* del sólido rígido.
 - q_p : cuaternión de orientación asociado al movimiento *pitch* del sólido rígido.
 - q_r : cuaternión de orientación inherente al movimiento *roll* del sólido rígido.
-
-

Los tres cuaterniones ligados a los movimientos de *yaw*, *pitch* y *roll* pueden calcularse a partir de los tres *Ángulos de Euler* anteriormente referenciados; sin más que aplicar las ecuaciones (2.11) a (2.13):

$$q_r = \cos\left(\frac{\phi_{IE}}{2}\right) (1000) + \sin\left(\frac{\phi_{IE}}{2}\right) (0100) \quad (2.11)$$

$$q_p = \cos\left(\frac{\vartheta_{AA}}{2}\right) (1000) + \sin\left(\frac{\vartheta_{AA}}{2}\right) (0010) \quad (2.12)$$

$$q_y = \cos\left(\frac{\psi_{FE}}{2}\right) (1000) + \sin\left(\frac{\psi_{FE}}{2}\right) (0001) \quad (2.13)$$

En este marco teórico⁸, resulta factible definir los cuaterniones de error intrínsecos a cada componente (q_y , q_p , q_r) del movimiento tridimensional. Dados los tres *Ángulos de Euler* (ϕ_{ERROR} , ϑ_{ERROR} , ψ_{ERROR}) que representan el error cometido (ya sea de orientación o desalineación sensor – segmento), se obtienen los cuaterniones de error (q_{r_e} , q_{p_e} , q_{y_e}) de forma análoga a lo indicado en las tres ecuaciones anteriores:

$$q_{r_e} = \cos\left(\frac{\phi_{ERROR}}{2}\right) (1000) + \sin\left(\frac{\phi_{ERROR}}{2}\right) (0100) \quad (2.14)$$

$$q_{p_e} = \cos\left(\frac{\vartheta_{ERROR}}{2}\right) (1000) + \sin\left(\frac{\vartheta_{ERROR}}{2}\right) (0010) \quad (2.15)$$

$$q_{y_e} = \cos\left(\frac{\psi_{ERROR}}{2}\right) (1000) + \sin\left(\frac{\psi_{ERROR}}{2}\right) (0001) \quad (2.16)$$

⁸ **Nota:** Durante el cálculo de los cuaterniones vinculados a los movimientos de *roll*, *pitch* y *yaw*, se han definido 4 cuaterniones auxiliares; los cuales se detallan a continuación:

$$q_1 = (1000); q_2 = (0100); q_3 = (0010); q_4 = (0001)$$

Estos cuaterniones adicionales permiten indicar las orientaciones espaciales involucradas en el cálculo de q_r , q_p , q_y , q_{r_e} , q_{p_e} y q_{y_e} .

De este modo, ya se dispone de seis cuaterniones $(q_r, q_p, q_y, q_{r_e}, q_{p_e}, q_{y_e})$ que permitirán incorporar las medidas de error al modelo de estimación propuesto en la sección "B.3". Como resultado de esta actividad, se obtendrá un cuaternión final \hat{q} ; cuyo cómputo se rige por la siguiente expresión:

$$\hat{q} = (q_y \otimes q_{y_e}) \otimes (q_p \otimes q_{p_e}) \otimes (q_r \otimes q_{r_e}) \quad (2.17)$$

Teniendo presente la anatomía del cuaternión expresado en la ecuación (2.17), ya se está en disposición de modelizar el efecto cuantitativo que poseen los errores de orientación y desalineación sobre la estimación final del algoritmo $({}^B q_{AF})$.

En el primer análisis, relativo al **estudio sobre la influencia de la desalineación sensor – segmento en la estimación de ${}^B q_{AF}$** , ha de emplearse la siguiente ecuación:

$${}^B q_{AF_{Est1}} = \left({}^{GS} q_A \otimes ({}^{GS} q_{A_0}^* \otimes {}^{GB} q'_{A_0}) \right)^* \otimes \left({}^{GS} q_F \otimes ({}^{GS} q_{F_0}^* \otimes {}^{GB} q'_{F_0}) \right) \quad (2.18)$$

El cálculo de los tres *Ángulos de Euler*, a partir del cuaternión ${}^B q_{AF_{Est1}}$, posibilita analizar la influencia de los errores de orientación de los *IMUs* sobre el valor final de la terna de ángulos $\{\psi_{FE}, \vartheta_{AA}, \phi_{IE}\}$.

Nueva notación utilizada en la ecuación (2.18)

- ${}^{GB} q'_{A_0}$: cuaternión auxiliar generado tras incorporar los errores de orientación del *IMU* o de alineación *segmento – sensor*. Dicho cuaternión se corresponde con la unidad de medida inercial colocada en el brazo de la persona y almacena la información registrada en el instante inicial ($t = 0$). Las componentes del *quaternion* ${}^{GB} q'_{A_0}$, correspondientes al *frame* cartesiano $\{B\}$, están expresadas respecto a un sistema de referencia global ($\{G\}$).

El parámetro ${}^{GB} q'_{A_0}$ puede calcularse como:

$${}^{GB}q'_{A_0} = (q_{A_0y_e} \otimes q_{A_0y}) \otimes (q_{A_0p} \otimes q_{A_0p_e}) \otimes (q_{A_0r} \otimes q_{A_0r_e}) \quad (2.19)$$

- ${}^{GB}q'_{F_0}$: Ídem que en el caso del cuaternión ${}^{GB}q'_{A_0}$, con la salvedad de que ${}^{GB}q'_{F_0}$ se vincula al sensor inercial situado en el antebrazo del individuo monitorizado. El *quaternion* ${}^{GB}q'_{F_0}$ presenta la siguiente estructura matemática:

$${}^{GB}q'_{F_0} = (q_{F_0y_e} \otimes q_{F_0y}) \otimes (q_{F_0p} \otimes q_{F_0p_e}) \otimes (q_{F_0r} \otimes q_{F_0r_e}) \quad (2.20)$$

El segundo estudio propuesto consiste en **analizar la influencia que poseen los errores de orientación**, correspondientes a los *IMUs* utilizados, sobre el cálculo final de los *Ángulos de Euler*; arrojados como resultado del algoritmo $(\phi_{ERROR}, \vartheta_{ERROR}, \psi_{ERROR})$. Las ecuaciones que permiten resolver este problema presentan numerosas similitudes con las expresiones (2.18) a (2.20), tal y como puede observarse a continuación:

$${}^Bq_{AF_{Est2}} = ({}^{GS}q'_A \otimes ({}^{GS}q^*_{A_0} \otimes {}^{GB}q_{A_0}))^* \otimes ({}^{GS}q'_F \otimes ({}^{GS}q^*_{F_0} \otimes {}^{GB}q_{F_0})) \quad (2.21)$$

Nueva notación empleada en la ecuación (2.21)

Los nuevos cuaterniones ${}^{GS}q'_A$ y ${}^{GS}q'_F$, introducidos en la expresión anterior, poseen un significado físico análogo al ya indicado para ${}^{GB}q'_{A_0}$ y ${}^{GB}q'_{F_0}$. Las únicas diferencias radican en los siguientes aspectos:

- En el presente caso de estudio, ${}^{GS}q'_A$ y ${}^{GS}q'_F$ no se vinculan exclusivamente al instante inicial $t = 0$, sino que hacen referencia a un tiempo genérico t .
- Por otro lado, los cuaterniones ${}^{GS}q'_A$ y ${}^{GS}q'_F$ almacenan la orientación tridimensional del *frame* correspondiente al sensor (*S*); a diferencia de ${}^{GB}q'_{A_0}$ y

${}^{GB}q'_{F_0}$, los cuales codifican la orientación del sistema de referencia asociado al brazo o antebrazo ($\{B\}$).

$${}^{GS}q'_A = (q_{A_{y_e}} \otimes q_{A_y}) \otimes (q_{A_p} \otimes q_{A_{p_e}}) \otimes (q_{A_r} \otimes q_{A_{r_e}}) \quad (2.22)$$

$${}^{GS}q'_F = (q_{F_{y_e}} \otimes q_{F_y}) \otimes (q_{F_p} \otimes q_{F_{p_e}}) \otimes (q_{F_r} \otimes q_{F_{r_e}}) \quad (2.23)$$

Finalmente, transformando el cuaternión ${}^Bq_{AF_{Est2}}$ a sus respectivos *Ángulos de Euler*, puede comprobarse numéricamente el sesgo introducido por los errores de estimación sobre las estimaciones finales de $\{\psi_{FE}, \vartheta_{AA}, \phi_{IE}\}$.

En la siguiente figura, puede visualizarse gráficamente el funcionamiento de la ecuación (2.18); relativa a la introducción de un error de alineación *sensor - segmento* durante la estimación inercial de la rotación articular. En este caso, dicha desalineación inicial es $\delta_{DESALINEACIÓN} = -18 [^\circ]$; mientras que las referencias de rotación articular toman los siguientes valores: $\lambda \in \{0^\circ, 5^\circ, 10^\circ\}$:

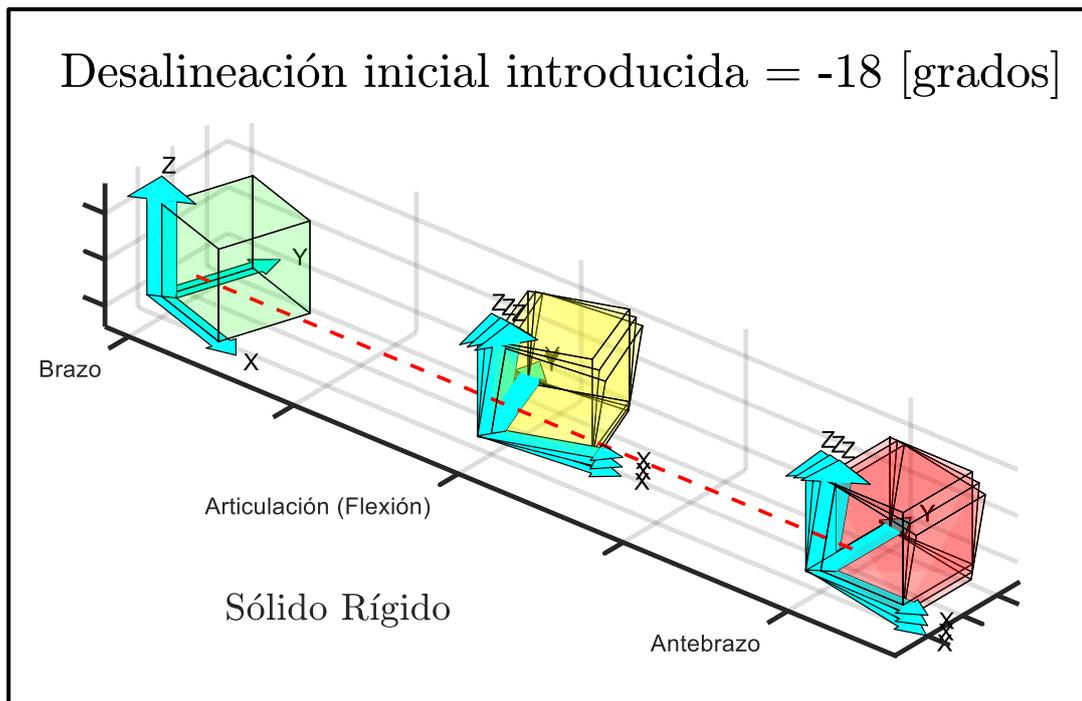


Figura 2.1.- Representación tridimensional de un experimento de flexión simulado computacionalmente. $\delta_{DESALINEACIÓN} = -18 [^{\circ}]$; $\lambda \in \{0^{\circ}, 5^{\circ}, 10^{\circ}\}$. [Elaboración propia].

A continuación, se recoge una ilustración sobre el experimento de flexión realizado con varios ángulos de desalineación inicial ($[-18^{\circ}, 0^{\circ}]$). Al igual que en el caso anterior, las referencias de rotación articular han sido 0° , 5° y 10° .

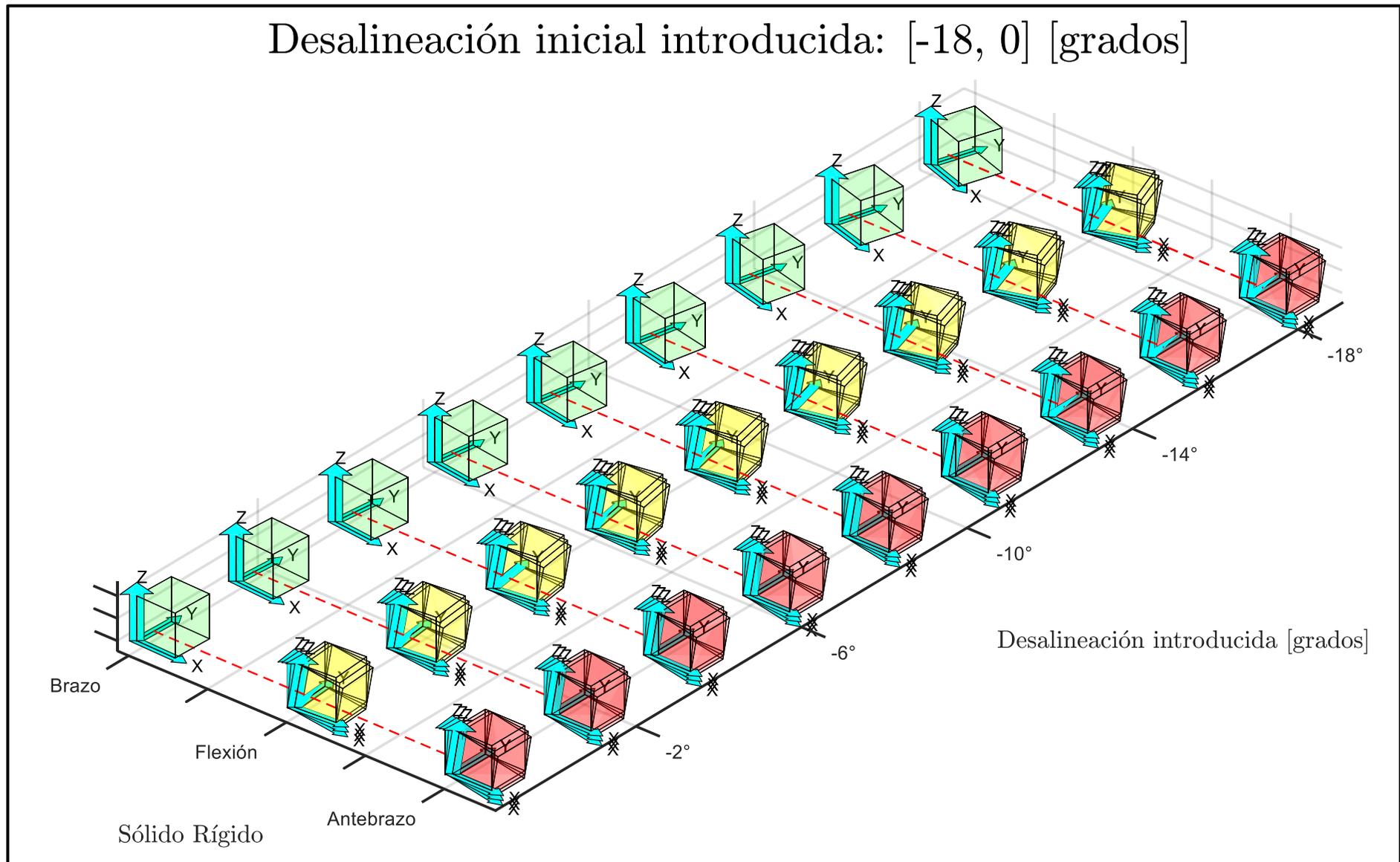


Figura 2.2.- Representación tridimensional de varios experimentos de flexión simulados computacionalmente. Previamente, se ha realizado la siguiente sintonización de parámetros: $\delta_{DESALINEACIÓN} \in [-18^\circ, 0^\circ]$; $\lambda \in \{0^\circ, 5^\circ, 10^\circ\}$. [Elaboración propia].

2.1.2.- Cálculo de los errores generados por el algoritmo durante la estimación de la orientación 3D, asociada al codo

Como corolario a todo el estudio de orientación desarrollado, en este subapartado se recogen las expresiones analíticas destinadas a cuantificar los errores de estimación; arrojados por el algoritmo implementado:

❖ *Error originado en el ángulo de rotación interna del codo (ϕ_{ERROR})*

$$\phi_{ERROR} = \phi_{q_{AF_{est}}} - \phi_{q_{AF_{ref}}} \quad (2.24)$$

❖ *Error asociado al ángulo de abducción del codo (ϑ_{ERROR})*

$$\vartheta_{ERROR} = \vartheta_{q_{AF_{est}}} - \vartheta_{q_{AF_{ref}}} \quad (2.25)$$

❖ *Error vinculado al ángulo de flexión del codo (ψ_{ERROR})*

$$\psi_{ERROR} = \psi_{q_{AF_{est}}} - \psi_{q_{AF_{ref}}} \quad (2.26)$$

2.1.3.- Representación gráfica relativa al problema de evaluación inercial planteado

A continuación, se maqueta una figura que permite observar esquemáticamente la relación geométrica existente entre los diversos sistemas de coordenadas cartesianas; ya definidos con anterioridad. Obsérvese que la posición de dichos *frames* cartesianos afecta a las estimaciones inerciales realizadas por los algoritmos implementados.

Nota: Se ha tomado la simplificación de que el sistema de referencia del mundo ($\{w\}$), de la base del robot ($\{b\}$) y del sensor ($\{s\}$) son coincidentes (frame $\{G\}$ representado en la siguiente figura).

RESUMEN VISUAL SOBRE LOS SISTEMAS DE REFERENCIA Y LOS CAMBIOS DE BASE DEFINIDOS EN EL PRESENTE PROBLEMA

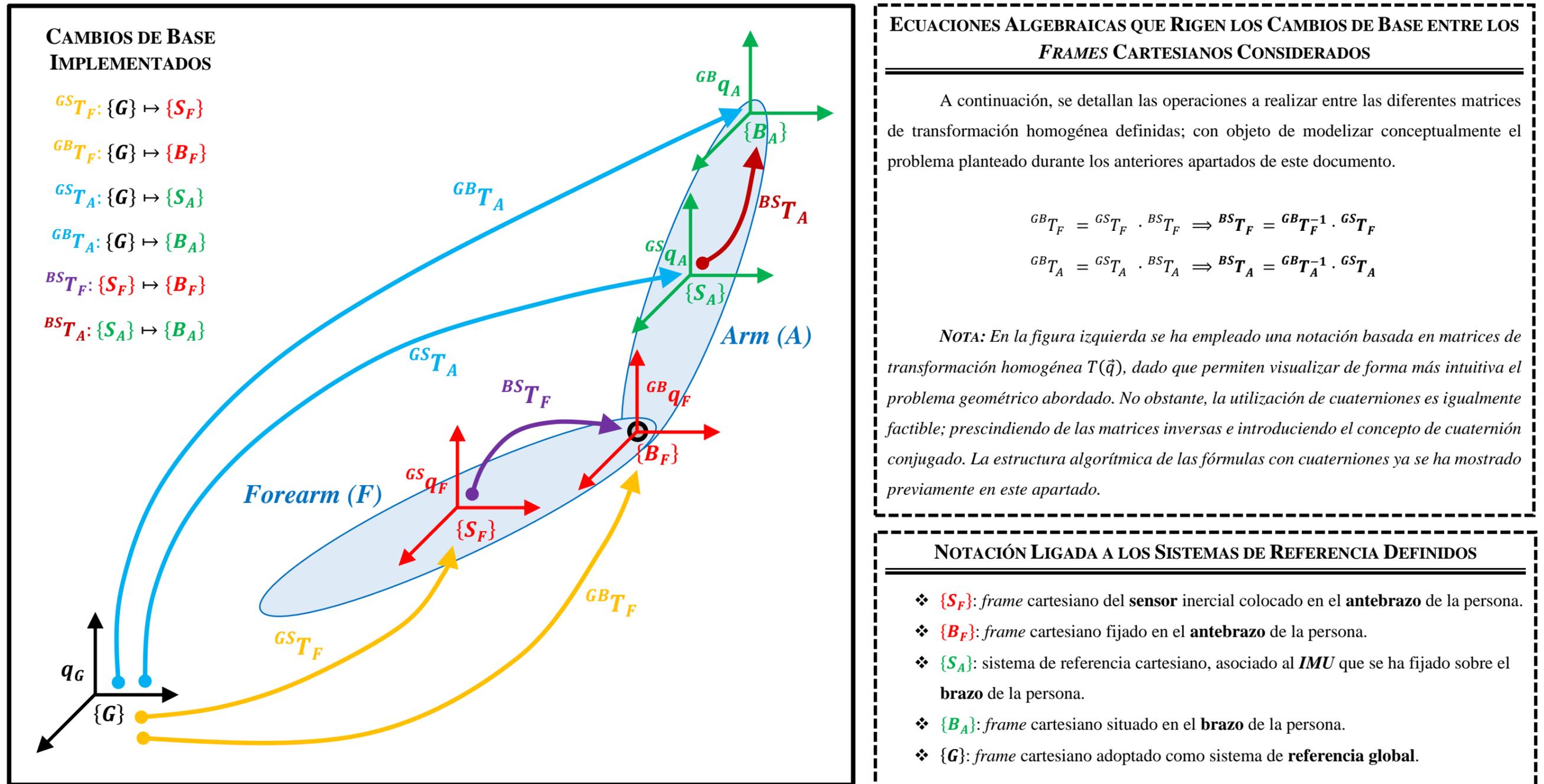


Figura 2.3.- Modelo geométrico diseñado para abordar el problema de estimación de los tres parámetros intrínsecos al movimiento del codo: ángulo de flexión (ψ_{FE}), abducción (ϑ_{AA}) y rotación interna (ϕ_{IE}).

2.2.- ALGORITMO *TRIAD* DE EVALUACIÓN INERCIAL

En este subapartado se describe brevemente la anatomía del algoritmo *TRIAD* [4]. Dicha solución computacional permite trabajar en coordenadas locales del sensor inercial (o del sólido rígido en estudio), sin necesidad de conocer en todo momento la relación existente con un marco cartesiano de referencia global.

Particularizando el estudio para el caso de un *IMU*, supóngase que se conocen los vectores de aceleración de la gravedad (\mathbf{b}_1) e intensidad de campo magnético (\mathbf{b}_2); ambos expresados en coordenadas locales del sensor inercial. De esta forma, se construirá el triedro $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ de la manera que sigue:

$$\mathbf{w}_1 = \mathbf{b}_1 \quad (2.27)$$

$$\mathbf{w}_2 = \mathbf{b}_\times = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{\|\mathbf{b}_1 \times \mathbf{b}_2\|} \quad (2.28)$$

$$\mathbf{w}_3 = \mathbf{b}_1 \times \mathbf{b}_\times = \mathbf{b}_1 \times \frac{\mathbf{b}_1 \times \mathbf{b}_2}{\|\mathbf{b}_1 \times \mathbf{b}_2\|} \quad (2.29)$$

Aplicando el procedimiento indicado en las tres ecuaciones anteriores, se obtendrán los sistemas de referencia locales a cada sensor inercial; los cuales están ubicados en el último ($\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\} \equiv \mathbf{W}$) y penúltimo ($\{\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3\} \equiv \mathbf{W}'$) eslabón del robot *UR3e*, respectivamente.

Finalmente, mediante la comparación de los triedros \mathbf{W} y \mathbf{W}' se generará la estimación inercial de la rotación articular analizada. Para ello, se recurrirá a los correspondientes *Ángulos de Euler*.

De forma complementaria, también resulta posible calcular la relación existente entre un sistema de referencia global y el *frame* local obtenido mediante *TRIAD*. Para ello, han de conocerse (en el marco global de referencia) los vectores de aceleración de la gravedad (\mathbf{r}_1)

e intensidad de campo magnético (\mathbf{r}_2). Análogamente al caso anterior, se obtendrá el triedro cartesiano $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$:

$$\mathbf{v}_1 = \mathbf{r}_1 \quad (2.30)$$

$$\mathbf{v}_2 = \mathbf{r}_\times = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{\|\mathbf{r}_1 \times \mathbf{r}_2\|} \quad (2.31)$$

$$\mathbf{v}_3 = \mathbf{r}_1 \times \mathbf{r}_\times = \mathbf{r}_1 \times \frac{\mathbf{r}_1 \times \mathbf{r}_2}{\|\mathbf{r}_1 \times \mathbf{r}_2\|} \quad (2.32)$$

Con todo ello, se tiene que la matriz de orientación que permite relacionar las coordenadas globales $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ con el triedro local $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ presenta la siguiente estructura:

$$\hat{A}_{TRIAD} = \mathbf{b}_1 \mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_\times)(\mathbf{r}_1 \times \mathbf{r}_\times)^T + \mathbf{b}_\times \mathbf{r}_\times^T \quad (2.33)$$

De esta manera, se pueden obtener las coordenadas locales asociadas a la rotación articular. Nótese que este algoritmo no tiene memoria: en cada instante no utiliza señales de los períodos anteriores.

3. Metodología de trabajo

3.1.- ESTRUCTURA DE LOS EXPERIMENTOS REALIZADOS

En primer lugar, se han adherido dos sensores inerciales al penúltimo y último eslabón del *cobot UR3e*; respectivamente. Inicialmente, se ha trabajado con la familia de *IMUs Xsens DOT* y, a continuación, con los *IMUs* de la gama *MATRIX*.

Seguidamente, se ha efectuado un protocolo de calibración ligado a cada sensor inercial; los cuales pueden consultarse mediante el siguiente subapartado. Para finalizar, se han capturado los datos relativos al movimiento de rotación articular; cuyo análisis permitirá evaluar los algoritmos de estimación inercial propuestos. El detalle de los movimientos programados durante la evaluación inercial se recoge en el subapartado "3.3.- Metodología experimental ligada a la evaluación inercial".

3.2.- DESCRIPCIÓN DEL EXPERIMENTO DE CALIBRACIÓN INERCIAL

En este apartado, se expone el protocolo de calibración⁹ diseñado para la aplicación biomecánica propuesta. A nivel *hardware*, se ha hecho uso del robot colaborativo *UR3*; cuyo montaje experimental puede consultarse en la siguiente figura:

⁹ *Nota:* Puede consultarse información adicional sobre la calibración inercial a través del apéndice "Anexo A: Fundamentos teóricos ligados a las técnicas de calibración sensorial".

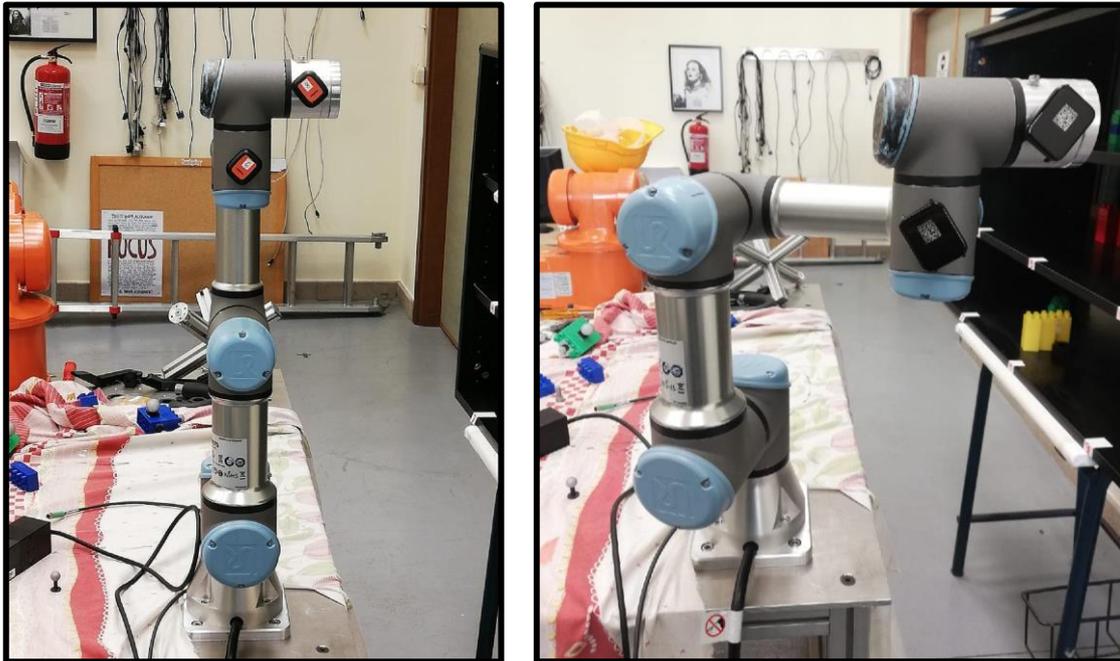


Figura 3.1.- Ubicación de las unidades de medida inercial en el robot colaborativo *UR3e*. En la fotografía de la izquierda se muestran los sensores de la familia *Xsens DOT*. En la imagen derecha pueden observarse los *IMUs* de la gama *MATRIX*.

Téngase en cuenta que se han desarrollado dos procesos de **calibración inercial**, correspondientes respectivamente a la **tipología 2D y 3D**. La ejecución de los protocolos de calibración resulta imprescindible para *alinearse* el sistema de referencia de cada *IMU* con el *frame* cartesiano asociado al respectivo eslabón del cobot *UR3e*.

El **último eslabón** del brazo robótico dispone de **tres grados de libertad**, lo cual permite *enfrentar* cada uno de los tres ejes cartesianos del correspondiente *IMU* contra la aceleración de la gravedad. De este modo, efectuando tres rotaciones de valor $\{\phi_X, \theta_Y, \beta_Z\}$, la **calibración 3D** conseguirá alinear el sistema de referencia ubicado en la base del robot y el *frame* intrínseco al sensor adherido al último eslabón de dicho *cobot*.

Por otro lado, la **calibración 2D** puede entenderse como un caso particular del procedimiento tridimensional. En esta situación, dado que el penúltimo eslabón del robot *UR3e* solamente posee dos grados de libertad, la calibración se *reduce* a rotar el plano *XY* del sensor inercial alrededor de su eje *Z* local.

A través de la siguiente figura, se esquematizan los grados de libertad disponibles en cada articulación del robot *UR3e*:

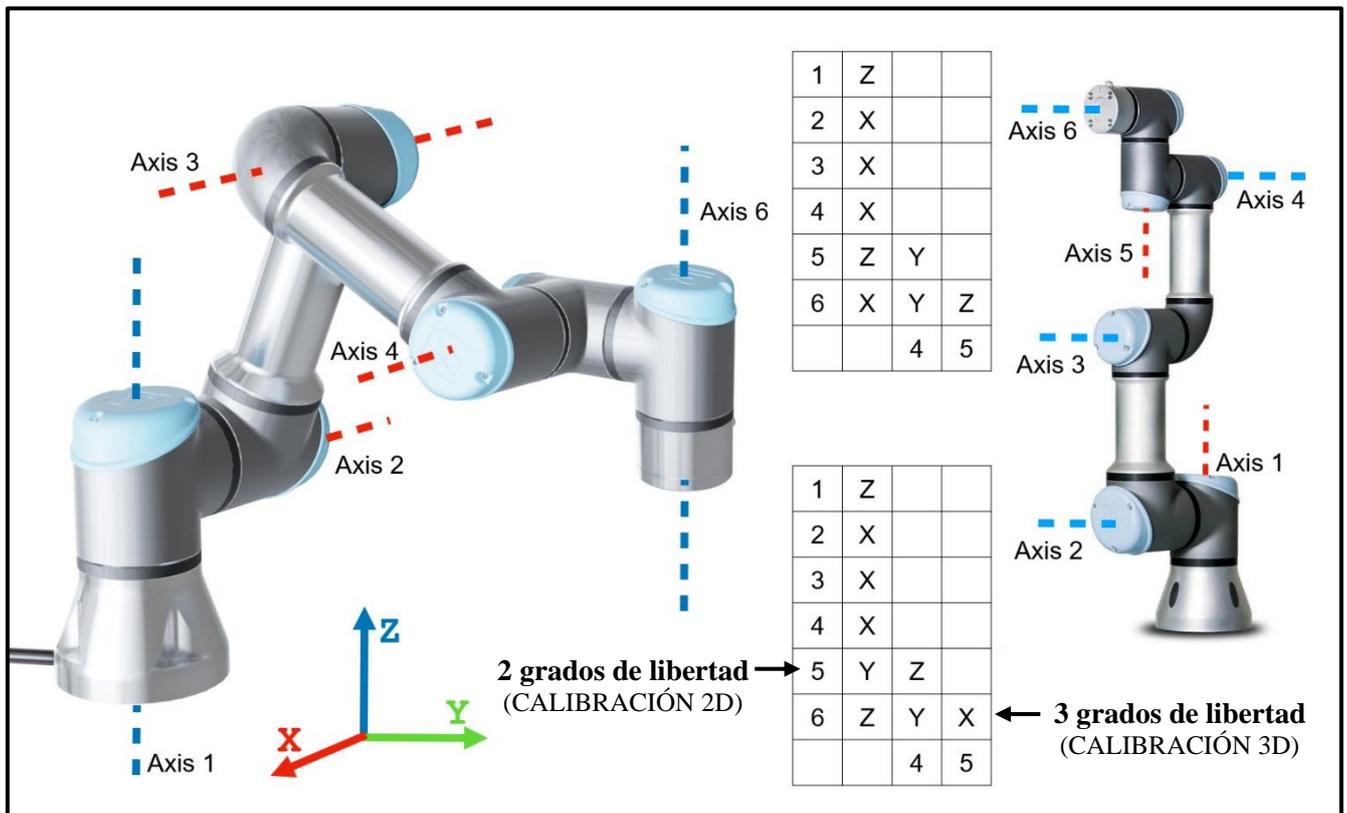


Figura 3.2.- Ejes motrices existentes en el brazo robótico *UR3e*. [Elaboración propia].

En los dos procedimientos experimentales desarrollados, se han sintonizado los siguientes parámetros como paso previo a la calibración de cada sensor inercial:

Tabla 3.1.- Variables establecidas durante la calibración inercial de los dispositivos pertenecientes a la familia *Xsens DOT*. Esta configuración ha sido realizada mediante la aplicación oficial de Android suministrada por el fabricante *Xsens*.

<i>Dispositivos Xsens DOT</i>	
Frecuencia de muestreo	60 [Hz]
Modo de captura	Modo 5 (Cuaterniones de Orientación + Aceleración + Velocidad)

Tabla 3.2.- Variables establecidas durante la calibración inercial de los dispositivos pertenecientes a la familia *MATRIX*. Dicha configuración se ha definido a través de la app de *MATRIX* disponible en Android (versión 1.2.7).

<i>Unidades de medida inercial MATRIX</i>	
Frecuencia de muestreo	100 [Hz]
Acelerómetro/Giroscopio	6 [ejes]
Rango de aceleración	8 [g]

La secuencia de movimientos que debe realizar el robot *UR3e* se recoge en las siguientes secciones.

3.2.1.- Metodología experimental de calibración tridimensional (3D)

A continuación, puede consultarse el experimento propuesto (**6 posiciones estáticas**) para calibrar el *IMU* situado en el último eslabón¹⁰ del brazo robótico *UR3e*:

Tabla 3.3.- Metodología experimental definida para calibrar el sensor inercial colocado en el último eslabón del robot colaborativo *UR3e*.

Sensor inercial (ubicado en el antebrazo)			Ejes del brazo robótico (movimiento articular)					
<i>Roll</i>	<i>Pitch</i>	<i>Yaw</i>	<i>Eje 1</i>	<i>Eje 2</i>	<i>Eje 3</i>	<i>Eje 4</i>	<i>Eje 5</i>	<i>Eje 6</i>
180°	0°	0°	0°	-90°	-90°	-90°	90°	180°
0°	0°	0°	0°	-90°	-90°	90°	90°	180°
0°	90°	0°	0°	-90°	-90°	-90°	180°	180°
0°	-90°	0°	0°	-90°	-90°	-90°	0°	180°
90°	-90°	0°	0°	-90°	-90°	-180°	0°	180°
90°	0°	-90°	0°	-90°	-90°	0°	0°	180°

¹⁰ Efectuando una comparación con la aplicación biomecánica propuesta, el último eslabón del *robot UR3e* adquiere el rol del antebrazo de la persona monitorizada.

3.2.2.- Metodología experimental de calibración bidimensional (2D)

De forma análoga a la secuencia de movimientos recogida en la Tabla 3.3, se ha creado otro protocolo experimental (**3 posiciones estáticas**) destinado a calibrar el sensor inercial adherido al penúltimo eslabón¹¹ del *cobot UR3e*. En la siguiente tabla se ilustran las referencias articulares del robot inherentes a dicha metodología experimental de calibración:

Tabla 3.4.- Metodología experimental definida para calibrar el sensor inercial colocado en el penúltimo eslabón del robot colaborativo *UR3e*.

Sensor inercial (situado en el brazo)			Ejes del brazo robótico (movimiento articular)					
<i>Roll</i>	<i>Pitch</i>	<i>Yaw</i>	<i>Eje 1</i>	<i>Eje 2</i>	<i>Eje 3</i>	<i>Eje 4</i>	<i>Eje 5</i>	<i>Eje 6</i>
90°	0°	0°	0°	-90°	-90°	0°	90°	0°
180°	0°	0°	0°	-90°	-90°	-90°	90°	0°
-90°	0°	0°	0°	-90°	-90°	90°	90°	0°

3.2.3.- Algoritmos empleados durante la calibración (2D y 3D) de los sensores inerciales

Con objeto de *corregir* las medidas proporcionadas por cada *IMU*, se han diseñado dos **procedimientos de calibración estática**; consistentes en implementar físicamente las secuencias de movimientos definidas en las secciones "3.2.1.- Metodología experimental de calibración tridimensional (3D)" y "3.2.2.- Metodología experimental de calibración bidimensional (2D)". El *software* empleado durante la programación del robot colaborativo *UR3e* ha sido **RoboDK**.

A continuación, se recogen varias imágenes sobre sendos protocolos de calibración sintonizados en el programa *RoboDK*:

¹¹ En este caso, el penúltimo eslabón del *cobot UR3e* desempeña la función motriz correspondiente al brazo de la persona en estudio.

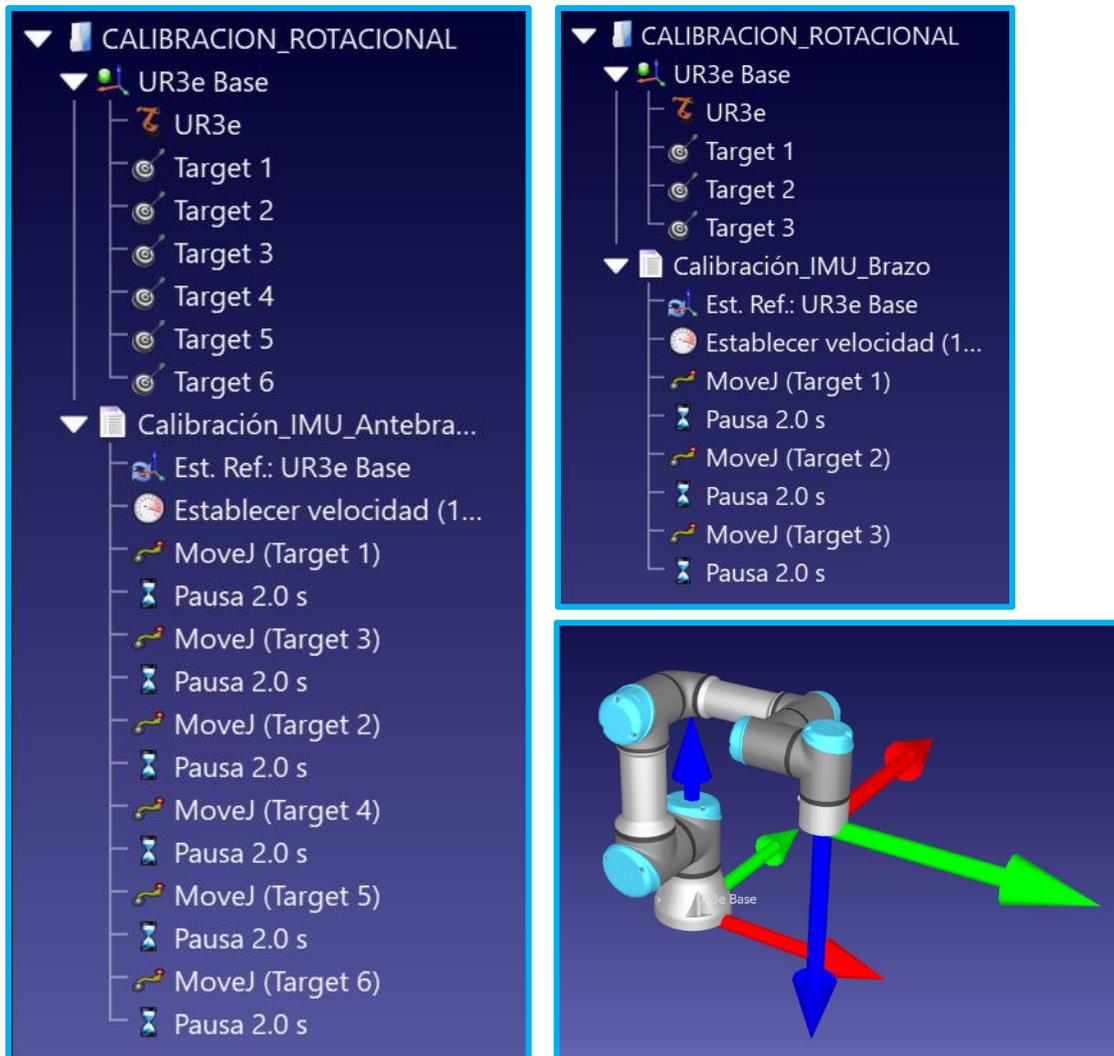


Figura 3.3.- Instrucciones de movimiento programadas en *RoboDK* para implementar físicamente los protocolos de calibración inercial. A la izquierda se recoge el programa de calibración *3D*. En la zona superior derecha se muestra el *software* de calibración *2D*. Finalmente, en la región inferior derecha se representa el modelo *CAD* utilizado durante el proceso de simulación.

Con objeto de garantizar que los valores de aceleración registrados durante la calibración se corresponden con posiciones estáticas, se han definido varias instrucciones de *pausa* entre cada par de movimientos; tal y como aparece reflejado en la Figura 3.3. Concretamente, cada *estado de reposo* del robot posee una duración de 2 [s].

Una vez capturados los datos inerciales, se ha ensamblado la siguiente matriz correspondiente a los **valores crudos de aceleración** (de la gravedad¹²):

$$[B]_{3 \times 3} = \begin{bmatrix} g_x^I & g_y^I & g_z^I \\ g_x^{II} & g_y^{II} & g_z^{II} \\ g_x^{III} & g_y^{III} & g_z^{III} \end{bmatrix}; \{I, II, III\} \equiv \text{muestra de datos} \quad (3.1)$$

Los vectores de aceleración de la gravedad g_x y g_y , resaltados mediante color rojo en la expresión anterior, son utilizados durante el proceso de calibración bidimensional. Por lado, la calibración 3D emplea la matriz $[B]$ completa (*datos en color rojo y azul*); puesto que precisa corregir la desalineación en los tres ejes cartesianos $\{X, Y, Z\}$ del sensor inercial.

El siguiente paso consiste en definir los **valores teóricos de \vec{g}** que se han de registrar en el triedro cartesiano del sensor, en función de cada posición estática definida. En el caso de la calibración 2D, resulta la siguiente matriz de referencia inercial:

$$[A]_{2D} = \begin{bmatrix} -g & 0 \\ 0 & -g \\ 0 & g \end{bmatrix}_{3 \times 2}, \text{ siendo } g = 9,8066 \left[\frac{m}{s^2} \right] \quad (3.2)$$

Para la situación de calibración tridimensional, se tiene que la matriz teórica $[A]$ adquiere la siguiente anatomía:

¹² **Nota:** Cabe destacar que las componentes cartesianas de la aceleración de la gravedad, registradas por cada IMU, poseen **intervalos de datos transitorios**; como consecuencia del movimiento realizado por el *cobot UR3e* **entre cada par de posiciones estáticas** definidas. Por ende, resulta necesario identificar de forma automatizada los datos de \vec{g} que se vinculan a cada una de las posiciones cartesianas de calibración establecidas.

Para llevar a cabo la actividad indicada en el párrafo anterior, se han utilizado las **señales suministradas por el giroscopio** de cada IMU; con objeto de **detectar** los índices asociados a los **máximos locales** de dichos datos digitales.

Téngase en consideración que un valor elevado en las señales angulares de giro informa sobre la realización de un nuevo movimiento, por parte del brazo robótico. De esta forma, empleando el vector de índices obtenido tras la *detección de picos* en la señal del giroscopio, pueden extraerse los valores de régimen permanente en las señales de aceleración; correspondientes a las respectivas posiciones estáticas definidas para la calibración inercial.

La algoritmia comentada durante la presente *Nota* se ha traducido a código *MATLAB* mediante la implementación de la función *DetectarValoresDeRegimenPermanente_En_UnaSenial_v3.mlx*, cuyo código fuente puede consultarse a través de los anexos adjuntados en este proyecto.

$$[A]_{3D} = \begin{bmatrix} 0 & -g & 0 \\ 0 & g & 0 \\ 0 & 0 & g \\ 0 & 0 & -g \\ -g & 0 & 0 \\ g & 0 & 0 \end{bmatrix}_{6 \times 3}, \text{ siendo } g = 9,8066 \left[\frac{m}{s^2} \right] \quad (3.3)$$

Finalmente, los datos inerciales contenidos en $[A]$ (aceleración de la gravedad teórica) y $[B]$ (medidas crudas de aceleración) se relacionan mediante una matriz de rotación $[R]$; la cual satisface la siguiente ecuación algebraica:

$$[A]_{3 \times m} = [R]_{3 \times 3}([B]_{m \times 3})_{3 \times m}^T \Leftrightarrow [A]([B]^T)^{-1} = [R][B]^T([B]^T)^{-1} = [R][I] \quad (3.4)$$

De acuerdo con el desarrollo matricial indicado en la expresión (3.4), puede deducirse que la matriz de calibración (rotación) $[R]$, tanto para el caso $2D$ como $3D$, toma la siguiente forma:

$$[R]_{2D|3D} = [A]([B]^T)^{-1} \quad (3.5)$$

A través de la matriz de rotación $[R]$, pueden obtenerse los valores calibrados de aceleración; sin más que aplicar la fórmula indicada a continuación:

$$[g_{CALIBRADA}]_{m \times 3} = ([R]_{3 \times 3}([g_{CRUDA}]_{m \times 3})_{3 \times m}^T)^T \quad (3.6)$$

3.3.- METODOLOGÍA EXPERIMENTAL LIGADA A LA EVALUACIÓN INERCIAL

La presente sección se ha dedicado a sintetizar el protocolo experimental diseñado para la evaluación metrológica de los sensores inerciales. Concretamente, se han evaluado ángulos de flexión comprendidos en el intervalo $[0^\circ, 70^\circ]$; mientras que la desalineación

sensor - segmento se encuentra dentro del rango $[-20^\circ, 20^\circ]$. Téngase en cuenta que dicha desalineación se ha asignado al *IMU* colocado en el penúltimo eslabón del brazo robótico.

En las siguientes tablas se recoge la secuencia de movimientos que debe realizar el *cobot UR3e* durante la evaluación inercial:

Tabla 3.5.- Metodología experimental definida para evaluar el sensor inercial colocado en el último eslabón del robot colaborativo *UR3e*. *Nota: esta secuencia de movimientos debe repetirse para cada ángulo de desalineación ($\delta_{DESALINEACIÓN} \equiv \delta$).*

Sensor inercial (ubicado en el <i>antebrazo</i>)			Ejes del brazo robótico (movimiento articular)					
<i>Roll</i>	<i>Pitch</i>	<i>Yaw</i>	<i>Eje 1</i>	<i>Eje 2</i>	<i>Eje 3</i>	<i>Eje 4</i>	<i>Eje 5</i>	<i>Eje 6</i>
0°	0°	0°	180°	-90°	-90°	0°	90°	0°
0°	5°	0°	180°	-90°	-90°	0°	90°	5°
0°	10°	0°	180°	-90°	-90°	0°	90°	10°
0°	15°	0°	180°	-90°	-90°	0°	90°	15°
0°	20°	0°	180°	-90°	-90°	0°	90°	20°
0°	25°	0°	180°	-90°	-90°	0°	90°	25°
0°	30°	0°	180°	-90°	-90°	0°	90°	30°
0°	35°	0°	180°	-90°	-90°	0°	90°	35°
0°	40°	0°	180°	-90°	-90°	0°	90°	40°
0°	45°	0°	180°	-90°	-90°	0°	90°	45°
0°	50°	0°	180°	-90°	-90°	0°	90°	50°
0°	55°	0°	180°	-90°	-90°	0°	90°	55°
0°	60°	0°	180°	-90°	-90°	0°	90°	60°
0°	65°	0°	180°	-90°	-90°	0°	90°	65°
0°	70°	0°	180°	-90°	-90°	0°	90°	70°

Tabla 3.6.- Metodología experimental definida para evaluar el sensor inercial colocado en el penúltimo eslabón del robot colaborativo *UR3e*. *Nota: téngase presente que todos los ángulos de flexión (λ [°]) se evaluarán para cada desalineación introducida ($\delta_{DESALINEACIÓN}$ [°]).*

Sensor inercial (situado en el brazo)			Ejes del brazo robótico (movimiento articular)					
Roll	Pitch	Yaw	Eje 1	Eje 2	Eje 3	Eje 4	Eje 5	Eje 6
-20°	0°	0°	160°	0°	0°	0°	110°	0°
-18°	0°	0°	162°	0°	0°	0°	108°	0°
-16°	0°	0°	164°	0°	0°	0°	106°	0°
-14°	0°	0°	166°	0°	0°	0°	104°	0°
-12°	0°	0°	168°	0°	0°	0°	102°	0°
-10°	0°	0°	170°	0°	0°	0°	100°	0°
-8°	0°	0°	172°	0°	0°	0°	98°	0°
-6°	0°	0°	174°	0°	0°	0°	96°	0°
-4°	0°	0°	176°	0°	0°	0°	94°	0°
-2°	0°	0°	178°	0°	0°	0°	92°	0°
0°	0°	0°	180°	0°	0°	0°	90°	0°
2°	0°	0°	182°	0°	0°	0°	88°	0°
4°	0°	0°	184°	0°	0°	0°	86°	0°
6°	0°	0°	186°	0°	0°	0°	84°	0°
8°	0°	0°	188°	0°	0°	0°	82°	0°
10°	0°	0°	190°	0°	0°	0°	80°	0°
12°	0°	0°	192°	0°	0°	0°	78°	0°
14°	0°	0°	194°	0°	0°	0°	76°	0°
16°	0°	0°	196°	0°	0°	0°	74°	0°
18°	0°	0°	198°	0°	0°	0°	72°	0°
20°	0°	0°	200°	0°	0°	0°	70°	0°

Combinando la información expresada en la Tabla 3.5 y la Tabla 3.6, el protocolo experimental destinado a la evaluación de las unidades de medida inercial puede resumirse de la manera que sigue:

Tabla 3.7.- Síntesis de la metodología experimental diseñada para la evaluación inercial de los IMUs. Nota: *obsérvese que esta tabla es válida para todos los ángulos de desalineación “sensor – segmento” propuestos ($\delta_{DESALINEACIÓN} \in \{-20^\circ, -18^\circ, \dots, 0^\circ, \dots, 18^\circ, 20^\circ\}$).*

Sensor inercial (ubicado en el brazo)			Sensor inercial (ubicado en el antebrazo)			Ejes del brazo robótico (movimiento articular)					
Roll	Pitch	Yaw	Roll	Pitch	Yaw	Eje 1	Eje 2	Eje 3	Eje 4	Eje 5	Eje 6
0°	0°	δ°	0°	0°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	0°
0°	0°	δ°	0°	5°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	5°
0°	0°	δ°	0°	10°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	10°
0°	0°	δ°	0°	15°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	15°
0°	0°	δ°	0°	20°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	20°
0°	0°	δ°	0°	25°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	25°
0°	0°	δ°	0°	30°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	30°
0°	0°	δ°	0°	35°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	35°
0°	0°	δ°	0°	40°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	40°
0°	0°	δ°	0°	45°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	45°
0°	0°	δ°	0°	50°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	50°
0°	0°	δ°	0°	55°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	55°
0°	0°	δ°	0°	60°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	60°
0°	0°	δ°	0°	65°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	65°
0°	0°	δ°	0°	70°	$-\delta^\circ$	$180^\circ + \delta^\circ$	-90°	-90°	0°	$90^\circ - \delta^\circ$	70°

4. Descripción del sistema utilizado

El presente apartado tiene como finalidad recoger los principales aspectos *hardware* y *software* que se vinculan al sistema utilizado durante la experiencia de laboratorio.

4.1.- SOFTWARE Y DISPOSITIVOS SELECCIONADOS

La principal herramienta *software* empleada durante el manejo y análisis de los datos inerciales adquiridos ha sido *MatLab R2022b*. Asimismo, también se ha hecho uso de los siguientes programas informáticos: *RoboDK*, **aplicaciones oficiales de Xsens** y *MATRIX* creadas para *Android*, así como el entorno de desarrollo *Spyder* (lenguaje *Python*).

A nivel *hardware*, los equipos seleccionados en este proyecto son: un **robot colaborativo UR3e**, varios **sensores inerciales** de las familias *Xsens DOT*, *MATRIX* y *bIMUr* (desarrollo propio por parte del grupo de investigación *SiMuR*), y un **dispositivo Android** (utilizado durante las capturas de datos realizadas en el *TFM*).

4.2.- MATERIAL HARDWARE EMPLEADO. RESUMEN DE CARACTERÍSTICAS TÉCNICAS

Esta entrada del documento *Memoria* se ha destinado a revisar los dispositivos *hardware* utilizados para el desarrollo del *Trabajo Fin de Máster*. A continuación, se exponen brevemente las especificaciones técnicas correspondientes a dicha tecnología.

4.2.1.- Sensores inerciales de la familia Xsens (IMUs Xsens DOT)

En primera instancia, se presentan las unidades de medidas inercial incluidas dentro de la familia *Xsens DOT* [36, 37]. Dichos *IMUs* (*Inertial Measurement Units*) permiten obtener medidas de aceleración, velocidad angular de giro e intensidad de campo magnético.

A modo ilustrativo, la siguiente figura ofrece una representación gráfica sobre el sistema de referencia cartesiano considerado para cada dispositivo *Xsens DOT*:



Figura 4.1.- Sistema cartesiano de referencia adoptado por los dispositivos *Xsens DOT* durante la captura de datos. *Imagen extraída como cortesía de [36].*

Las dimensiones físicas de este tipo de sensores inerciales obedecen al plano geométrico recogido a continuación:

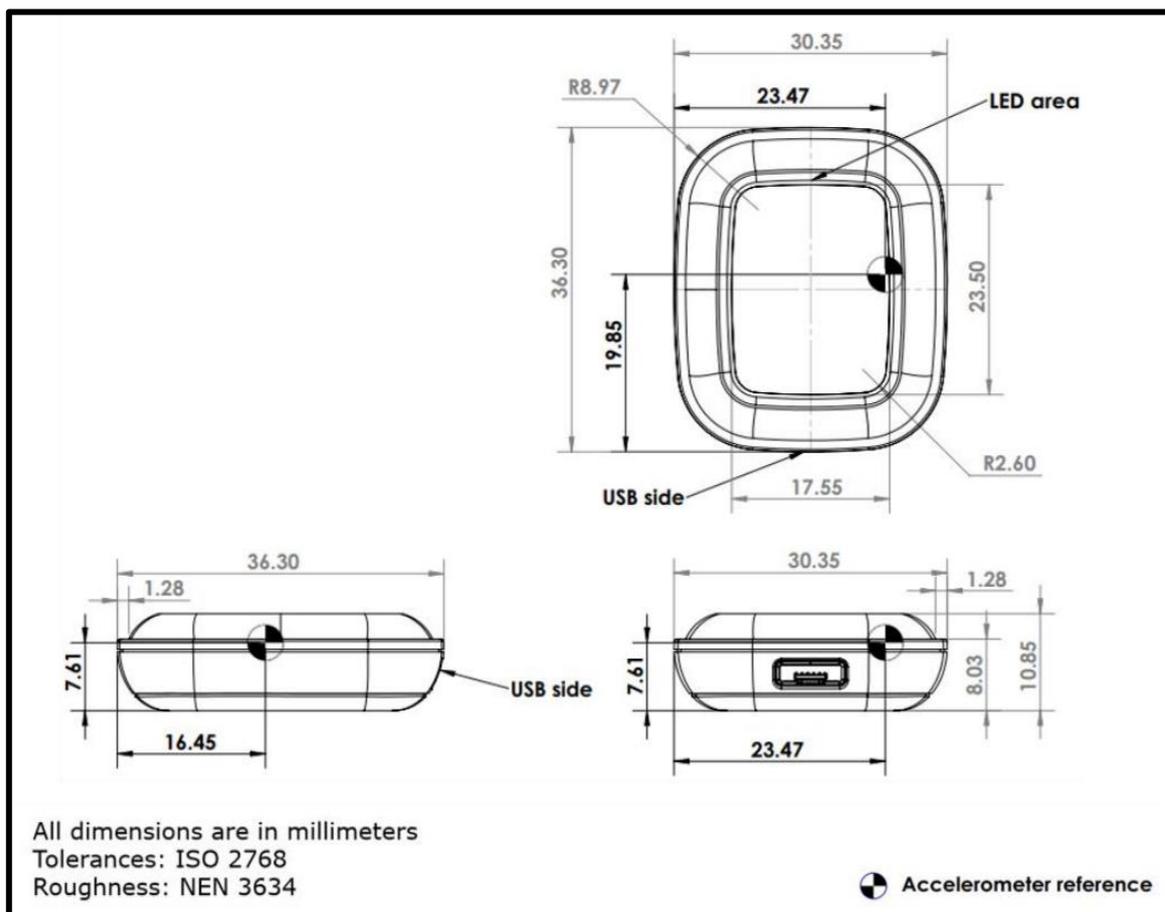


Figura 4.2.- Especificaciones geométricas de los dispositivos *Xsens DOT* (versión v2). Plano extraído como cortesía de [36].

A través de las tablas mostradas a continuación, se han sintetizado las características técnicas asociadas a los sensores inerciales *Xsens DOT*; los cuales están internamente conformados por acelerómetros, giroscopios y magnetómetros:

Tabla 4.1.- Características propias del acelerómetro que posee un *IMU Xsens DOT*.

ESPECIFICACIONES INSTRUMENTALES DEL ACCELERÓMETRO PRESENTE EN UN <i>IMU XSENS DOT</i>	
Característica electrónica	Valor de la magnitud física
Resolución	16 [bits]
Sensibilidad (ganancia o factor de escala)	2048 [LSB/g]
Fondo de escala (amplitud del rango de medida)	± 16 [g]
Frecuencia de muestreo durante la captura de datos	800 [Hz]

Densidad de ruido en la etapa de salida	120 $\left[\frac{\mu\text{g}}{\sqrt{\text{Hz}}}\right]$
<i>Bias</i> existente en condiciones normales de funcionamiento	0,03 [mg]
Ancho de banda	324 [Hz]
No – linealidad	$\pm 0,5$ [% FS]

Tabla 4.2.- Características asociadas al giróscopo que posee un *IMU Xsens DOT*.

ESPECIFICACIONES INSTRUMENTALES DEL GIROSCOPIO PRESENTE EN UN <i>IMU XSENS DOT</i>	
Característica electrónica	Valor de la magnitud física
Resolución	16 [bits]
Sensibilidad (ganancia o factor de escala)	16,4 [LSB/g]
Fondo de escala (amplitud del rango de medida)	± 2000 [°/s]
Frecuencia de muestreo durante la captura de datos	800 [Hz]
Densidad de ruido en la etapa de salida	0,007 $\left[\frac{°/s}{\sqrt{\text{Hz}}}\right]$
<i>Bias</i> existente en condiciones normales de funcionamiento	10 [°/h]
Ancho de banda	255 [Hz]
No – linealidad	0,1 [% FS]
G – sensibilidad	0,1 $\left[\frac{°/s}{g}\right]$
Variación de la sensibilidad (valor típico)	0,5 [%]

Tabla 4.3.- Características ligadas al magnetómetro que posee un *IMU Xsens DOT*.

ESPECIFICACIONES INSTRUMENTALES DEL MAGNETÓMETRO PRESENTE EN UN <i>IMU XSENS DOT</i>	
Característica electrónica	Valor de la magnitud física
Resolución	0,25 [mGauss]
Fondo de escala (amplitud del rango de medida)	± 8 [Gauss]
Frecuencia de muestreo durante la captura de datos	60 [Hz]

No - linealidad	0,2 [% FS]
Valor medio (RMS) de ruido total	0,5 [mGauss]
Resolución (expresada en bits)	16 [bits]

4.2.2.- Unidades de medida inercial pertenecientes a la gama *Matrix*

La segunda gama de sensores inerciales considerados en este proyecto se corresponde con la familia *Matrix*. Estas unidades de medida inercial permiten registrar valores de aceleración lineal y velocidad angular de giro¹³. Sin embargo, no poseen la capacidad de monitorizar la intensidad de campo magnético existente en la zona de medición.

En conformidad con lo expresado durante el párrafo anterior, los sensores de la familia *MATRIX* constituyen una solución económica para resolver multitud de problemas en el ámbito de la Ingeniería Biomecánica; a pesar de no ofrecer tantas funcionalidades como los *IMUs Xsens DOT*.

Mediante la siguiente figura, puede observarse una fotografía del sensor inercial incluido en la familia *MATRIX*:

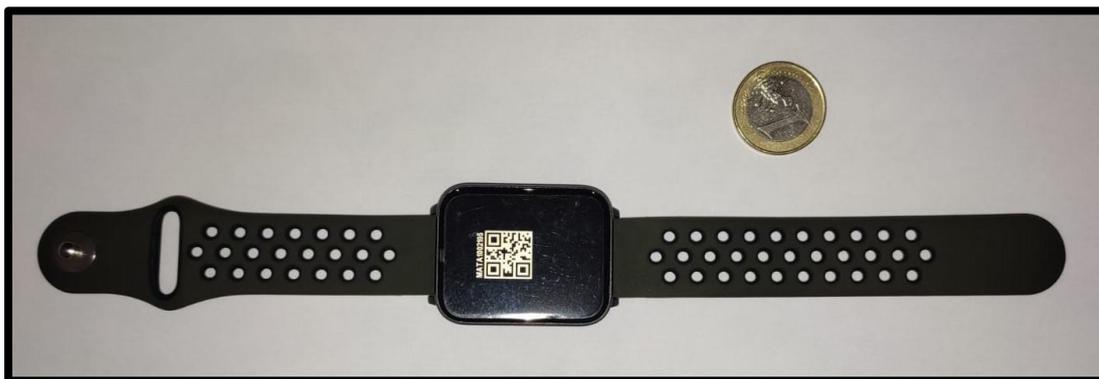


Figura 4.3.- Imagen de ejemplo relativa a una unidad de medida inercial *MATRIX*. [Elaboración propia].

¹³ **Nota:** De forma adicional a los datos inerciales, proporcionados por el acelerómetro y el giroscopio del *IMU MATRIX*, este sensor es capaz de medir las temperaturas ambiente y corporal de la persona monitorizada. Asimismo, dicho dispositivo posee la tecnología necesaria para capturar la frecuencia cardíaca del sujeto en estudio. Esta magnitud es proporcionada por el *IMU MATRIX* en valores crudos, junto con la señal de *heart rate* generada tras un procesamiento digital inicial.

A continuación, se enumeran las principales características que rigen el funcionamiento de un sensor inercial perteneciente al grupo *MATRIX*:

Tabla 4.4.- Características técnicas que gobiernan la funcionalidad de un sensor inercial *MATRIX*. *Estos datos han sido obtenidos mediante la información referenciada en [38].*

ESPECIFICACIONES TÉCNICAS DE UN <i>IMU MATRIX</i>	
ACELERÓMETRO QUE POSEE SENSOR INERCIAL	
Característica en estudio	Valor de la magnitud física
Resolución	16 [bits]
Fondo de escala	± 16 [g]
Ejes de medida	6 [ejes] ó 9 [ejes]
Tolerancia a los golpes	2000 [g]
<i>Chip</i> integrado	ICM – 40607
OTRAS CARACTERÍSTICAS INSTRUMENTALES DEL <i>IMU MATRIX</i>	
Resolución vinculada a la medida de la temperatura ambiente	1 [°C]
Resolución correspondiente a la medida de la temperatura corporal	0,1 [°C]
Capacidad de la batería incorporada en el sensor inercial	200 [mA·h]
Duración estimada de la energía almacenada en la batería	Más de 8 [días] capturando datos a una frecuencia de 100 [Hz]
Resistencia del sensor al agua	Hasta 50 [m] de profundidad
Tipo de conectividad con la aplicación móvil, destinada a la adquisición de datos	<i>Bluetooth</i>
Memoria integrada en el dispositivo electrónico	512 [MB] (ampliable a 1 [GB])

4.2.3.- *SuperSens/bIMUr*: Sensor inercial desarrollado por el grupo de investigación *SiMuR*

Las características técnicas de este dispositivo de medida inercial son propiedad del grupo de investigación *SiMuR* [39].

4.2.4.- Robot colaborativo (*cobot*) UR3e. Caja de control “Teach Pendant”

En esta sección se recogen las principales características intrínsecas al brazo robótico UR3e [40, 41]. En primer lugar, cabe señalar que el *cobot* UR3e dispone de seis articulaciones cilíndricas. A modo ilustrativo, en la siguiente figura se expone dicho robot realizando una tarea automatizada de fabricación mecánica:

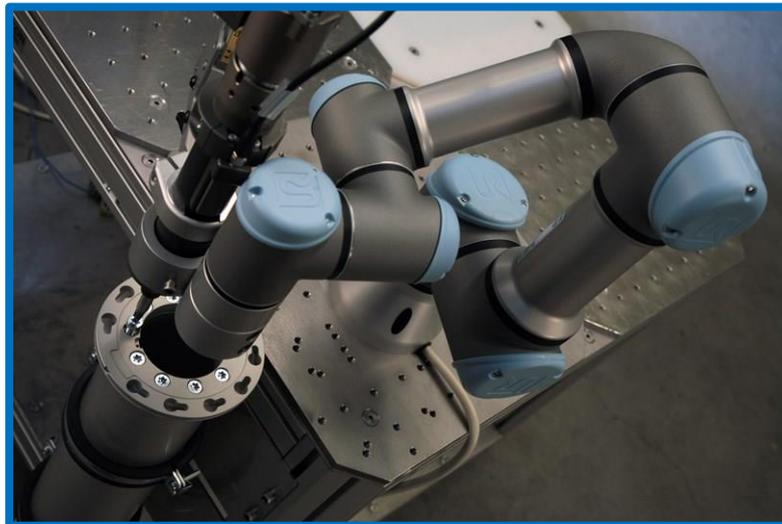


Figura 4.4.- Actividad de manufactura mecánica, la cual se ha automatizado mediante la utilización del robot colaborativo UR3e. Esta imagen ha sido extraída como cortesía del sitio web oficial ligado al fabricante Universal Robots [1].

De forma general, pueden destacarse tres **características** del *cobot* UR3e; vinculadas a su **naturaleza mecánica**:

Tabla 4.5.- Principales especificaciones mecánicas del brazo robótico UR3e.

CARACTERÍSTICAS GENERALES DEL ROBOT COLABORATIVO UR3E	
Especificación del mecanismo robotizado	Valor de la magnitud física
Máxima carga soportada por el efector final del brazo robótico	3 [kg]
Máximo alcance del mecanismo articulado	500 [mm]
Grados de libertad (<i>DOF – Degrees Of Freedom</i>)	6 [-] (articulaciones cilíndricas)

En combinación con la información recogida mediante la tabla anterior, las **características propias del movimiento** desarrollado por el *cobot UR3e* se han resumido de la manera que sigue:

Tabla 4.6.- Características cinemáticas del mecanismo articulado *UR3e*.

ESPECIFICACIONES DE MOVIMIENTO ASOCIADAS AL <i>COBOT UR3E</i>		
Característica cinemática del robot	Valor de la magnitud física	
Repetibilidad de la pose adoptada por el robot (según la norma <i>ISO 9283</i>)	$\pm 0,03$ [mm]	
Velocidad de trabajo habitual	1 [m/s]	
ESPECIFICACIONES CINEMÁTICAS INHERENTES A CADA ARTICULACIÓN DEL BRAZO ROBÓTICO		
Ubicación del respectivo eje motriz	Rango de operación	Velocidad máxima de trabajo
<i>Base</i>	± 360 [°]	± 180 [°/s]
<i>Hombro</i>	± 360 [°]	± 180 [°/s]
<i>Codo</i>	± 360 [°]	± 180 [°/s]
<i>Primera muñeca</i>	± 360 [°]	± 360 [°/s]
<i>Segunda muñeca</i>	± 360 [°]	± 360 [°/s]
<i>Tercera muñeca</i>	<i>Infinito</i>	± 360 [°/s]

Por otro lado, en cuanto a **términos eléctricos** se refiere, el robot *UR3e* ofrece las **prestaciones** indicadas a continuación:

Tabla 4.7.- Propiedades relativas al subsistema eléctrico del brazo robótico *UR3e*.

CARACTERÍSTICAS ELÉCTRICAS DEL <i>COBOT UR3E</i>	
Especificación eléctrica considerada	Valor de la magnitud correspondiente
Consumo máximo de potencia eléctrica ¹⁴	300 [W]

¹⁴ **Nota:** El máximo consumo de potencia eléctrica indicado (300 [W]) ha sido obtenido por el fabricante *Universal Robots* a través de una media estadística. Por ende, puede existir un pequeño margen de error ($\pm \varepsilon$ [%]) en el valor *real* asociado a la potencia eléctrica máxima que es demandada por el equipo robótico colaborativo *UR3e*.

Consumo típico de potencia eléctrica en condiciones normales de funcionamiento	100 [W]
Tensión de alimentación requerida por los puertos de entrada / salida (I / O), disponibles en el <i>cobot UR3e</i>	12 [V] / 24 [V]
Corriente de alimentación demandada por los puertos de entrada / salida (I / O)	600 [mA]
Número de líneas digitales de entrada	2 [-]
Número de líneas digitales de salida	2 [-]
Número de líneas analógicas de entrada	2 [-]

Adicionalmente a las características comentadas con anterioridad, resulta interesante conocer las **especificaciones físicas** del robot *UR3e*; las cuales se recogen en la siguiente tabla:

Tabla 4.8.- Características físicas del robot colaborativo *UR3e*.

PROPIEDADES FÍSICAS DEL BRAZO ROBÓTICO UR3E	
Característica física	Detalle (cuantitativo o cualitativo)
Diámetro (\emptyset) de la base que posee el robot	128 [mm]
Materiales de fabricación	Acero, aluminio y plástico
Tipo de conexión con la herramienta (efector final del <i>cobot</i>)	Ocho pines de métrica 8 (M8)
Rango de temperatura de operación recomendado	[0, 50] [°C]
Humedad relativa (HR) soportada	90 [%]
Longitud del cable que permite la alimentación eléctrica del robot	6 [m]
Peso del brazo robótico (incluido el cable de alimentación)	11,2 [kg]
Montaje del robot	En cualquier orientación
Nivel de ruido	< 60 dB (A)
Clasificación IP	IP 54

Finalmente, se exponen las principales prestaciones que ofrece el *cobot UR3e* en materia de funcionalidad mecánica¹⁵.

Tabla 4.9.- Prestaciones mecánicas suministradas por el brazo robótico *UR3e*.

FUNCIONALIDAD MECÁNICA DEL ROBOT COLABORATIVO UR3E	
FUERZA PROPORCIONADA EN LOS EJES CARTESIANOS X, Y, Z	
Prestación mecánica	Valor numérico de la propiedad física
Precisión	3,5 [N]
Rango de aplicabilidad	30,0 [N]
Resolución	1,0 [N]
PAR MECÁNICO GENERADO EN LOS EJES CARTESIANOS X, Y, Z	
Precisión	0,10 [N·m]
Rango de aplicabilidad	10,00 [N·m]
Resolución	0,02 [N·m]

Nota Adicional: El funcionamiento del robot colaborativo *UR3e* se gobierna por medio de la Caja de Control “Teach Pendant”, la cual pertenece a la clasificación IP54. Puede consultarse más información sobre este dispositivo de regulación mediante la bibliografía referenciada en [39, 40].

4.3.- DISEÑO SOFTWARE

En el presente subapartado se exponen algunos aspectos *software* relacionados con la calibración y evaluación inerciales, efectuadas durante el transcurso del proyecto.

¹⁵ En cualquier aplicación de robótica, resulta indispensable conocer detalladamente los rangos de fuerza y par mecánico que es capaz de proporcionar la cadena cinemática considerada. Por ello, se ha considerado adecuado construir la Tabla 4.9.

Concretamente, se ofrecerá una breve panorámica sobre el modelo construido en *RoboDK* y *Python* para regular el funcionamiento del robot colaborativo *UR3e*.

4.3.1.- Modelo de *RoboDK/Python* que permite gobernar el funcionamiento del *cobot UR3e*

A continuación, se exponen las herramientas *software* construidas para regular el movimiento del brazo robótico *UR3e*. Con la finalidad de automatizar la implementación del protocolo experimental diseñado, se ha elaborado un código en lenguaje *Python*. Dicha utilidad computacional permitirá introducir fácilmente cualquier modificación sobre la metodología experimental inicialmente prevista.

```
# PROGRAMA PRINCIPAL
if __name__ == '__main__':

    RDK = Robolink()
    robot = RDK.ItemUserPick('Select a robot', ITEM_TYPE_ROBOT)
    if not robot.Valid():
        raise Exception('No robot selected or available')

    RUN_ON_ROBOT = True

    if RDK.RunMode() != RUNMODE_SIMULATE:
        RUN_ON_ROBOT = False

    if RUN_ON_ROBOT:
        success = robot.Connect()
        status, status_msg = robot.ConnectedState()
        if status != ROBOTCOM_READY:
            print(status_msg)
            raise Exception("Failed to connect: " + status_msg)

    RDK.setRunMode(RUNMODE_RUN_ROBOT)    [...]
    print(status)
```



Figura 4.5.- Fragmento del código *Python* desarrollado para realizar una conexión con el *software RoboDK* y el robot físico *UR3e*. [Elaboración propia].

Cabe resaltar que el algoritmo *Python* construido permite almacenar en un fichero de texto (extensión ".txt") todas las referencias articular comandadas al robot *UR3e*; así como los ángulos de desalineación introducidos. Finalmente, téngase en cuenta que todos los movimientos ejecutados por el robot colaborativo *UR3e* pueden visualizarse en tiempo real a través del simulador *RoboDK*.

5. Resultados

A lo largo de este apartado se exponen los principales resultados, tanto teóricos (modelo matemático) como experimentales, obtenidos en el presente *Trabajo Fin de Máster*.

5.1.- CARACTERIZACIÓN DE LOS SENSORES INERCIALES

A continuación, se muestra la desviación estándar inherente a las medidas de cada sensor inercial. Los cálculos se han particularizado para los tres ejes cartesianos presentes en el acelerómetro, giroscopio o magnetómetro de cada sensor.

Tabla 5.1.- Desviaciones típicas vinculadas a las medidas proporcionadas por los sensores inerciales. Cada captura ha tenido una duración de 1 minuto y los *IMUs* se han colocado en una posición estática.

STD (Desviación estándar)	Acelerómetro [mg]			Giróscopo [grados/s]			Magnetómetro [μT]		
	x	y	z	x	y	z	x	y	z
Xsens DOTx #1	0.73	0.73	1.17	0.04	0.04	0.04	0.07	0.08	0.08
Xsens DOTx #2	0.75	0.78	1.19	0.04	0.04	0.03	0.07	0.08	0.09
MATRIX #1	2.78	2.99	2.88	0.15	0.72	0.08	--	--	--
MATRIX #2	2.61	2.91	2.52	0.12	0.08	0.06	--	--	--

#1 y #2 designan a dos dispositivos de la misma familia de sensores inerciales. Frecuencia de captura de datos: 60 [Hz] (*Xsens DOT*), 100 [Hz] (*MATRIX*). Duración de cada captura: 1 [min].

5.1.1.- Estudio sobre el estado de los sensores considerados, como paso previo a la calibración de dichos dispositivos

En esta sección se recoge un análisis relativo al estado inicial (ruido presente en las señales de aceleración) vinculado a los sensores inerciales considerados, en cuanto a términos de calibración metrológica se refiere. Para llevar a cabo dicha actividad, se han analizado de forma independiente las dos familias de sensores inerciales empleadas en este proyecto: *Xsens DOT* y *MATRIX*.

La metodología experimental, seguida durante la adquisición de los datos de aceleración de la gravedad, ha consistido en situar horizontalmente cada *IMU* sobre una mesa; la cual se ha utilizado como plano horizontal de referencia ($\varphi_{INCLINACIÓN} = 0$). Durante la recogida de datos, todas las unidades de medida inercial han permanecido en estado de reposo.

A continuación, se muestran varias representaciones gráficas vinculadas a los valores de aceleración de la gravedad; los cuales han sido registrados a través de los correspondientes sensores inerciales.

❖ Unidad de medida inercial Xsens DOT 7

En la Figura 5.1 puede observarse la evolución temporal de las componentes cartesianas inherentes al vector \vec{g} (aceleración de la gravedad): $[g_x, g_y, g_z] \left[\frac{m}{s^2} \right]$. Dichas representaciones gráficas se han resaltado mediante color azul. Asimismo, se muestra el valor medio vinculado a sendas señales de aceleración de la gravedad; a través de una línea de color rojo y trazo discontinuo.

Aceleración de la gravedad registrada mediante el IMU Xsens DOT7, apoyado sobre una mesa

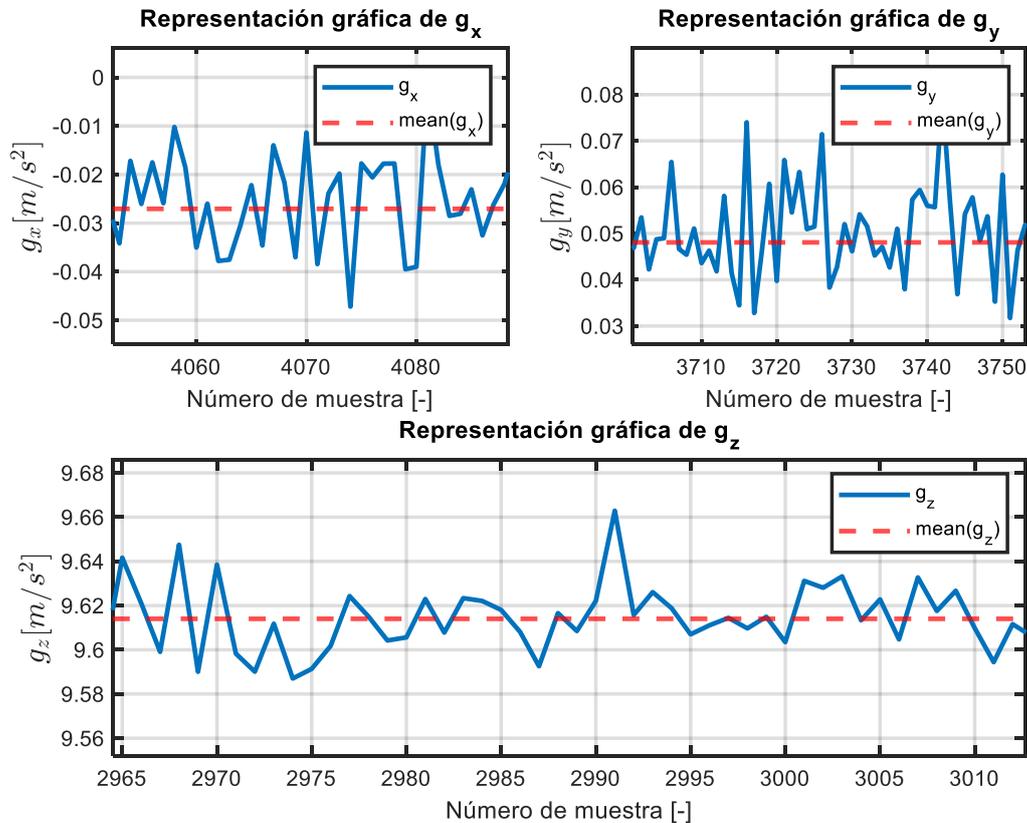


Figura 5.1.- Componentes cartesianas de la aceleración de la gravedad, registradas mediante el sensor inercial Xsens DOT7.

De la figura anterior se desprende que el *offset* presente en las componentes cartesianas g_x y g_y es prácticamente nulo. Concretamente, el valor medio ligado a la señal g_x es $\overline{g_x} = -0,0271 \left[\frac{m}{s^2} \right]$; mientras que para g_y se verifica que $\overline{g_y} = 0,0481 \left[\frac{m}{s^2} \right]$.

Por otra parte, la media estadística ligada a la señal g_z asciende a $\overline{g_z} = 9,614 \left[\frac{m}{s^2} \right]$. Considerando que el valor teórico esperado para $\overline{g_z}$ es $9,81 \left[\frac{m}{s^2} \right]$, puede calcularse el error presente en la medición de la siguiente forma:

$$\varepsilon_{\overline{g_z}} = \frac{|\overline{g_z} - \overline{g_z}^{Teórico}|}{\overline{g_z}^{Teórico}} \cdot 100 \quad (5.1)$$

Incorporando en la expresión (5.1) los datos conocidos sobre $\overline{g_z}$, resulta:

$$\varepsilon_{\overline{g_z}}^{Xsens DOT7} = \frac{\left| 9,614 \left[\frac{m}{s^2} \right] - 9,81 \left[\frac{m}{s^2} \right] \right|}{9,81 \left[\frac{m}{s^2} \right]} \cdot 100 \cong 2,00 [\%] \quad (5.2)$$

De acuerdo con el análisis realizado, el **error (bias)** existente en las **mediciones de aceleración** suministradas por el sensor inercial *Xsens DOT7* puede acotarse superiormente mediante la cifra **2,00 [%]** ($= \varepsilon_{\overline{g_z}}^{Xsens DOT7}$).

❖ Sensor inercial MATA1002195 (familia MATRIX)

Análogamente al estudio elaborado para el *IMU Xsens DOT 7*, se ha comprobado el estado inicial de la calibración presente en la unidad de medida inercial *MATA1002195*. Una vez registrados los datos vinculados a la aceleración de la gravedad, se ha maquetado la representación gráfica mostrada en la Figura 5.2.

En este caso, se verifica que el valor medio asociado a la componente *X* de la aceleración de la gravedad asciende a $\overline{g_x} = -0,2287 \left[\frac{m}{s^2} \right]$. Por otro lado, se tiene que $\overline{g_y} = 0,0554 \left[\frac{m}{s^2} \right]$. Finalmente, la componente principal de la magnitud vectorial \vec{g} toma el siguiente valor medio: $\overline{g_z} = -9,4381 \left[\frac{m}{s^2} \right]$.

Teniendo presente la información indicada con anterioridad, el cálculo del error aparejado a las medidas de aceleración suministradas por el *IMU MATA1002195* se traduce en la siguiente expresión matemática:

$$\varepsilon_{\overline{g_z}}^{MATA1002195} = \frac{\left| -9,4381 \left[\frac{m}{s^2} \right] - (-9,81) \left[\frac{m}{s^2} \right] \right|}{\left| -9,81 \left[\frac{m}{s^2} \right] \right|} \cdot 100 \cong 3,791 [\%] \approx 3,80 [\%] \quad (5.3)$$

En conformidad con el análisis efectuado, el *offset (bias)* existente en las **mediciones de aceleración** suministradas por el sensor inercial *MATA1002195* puede acotarse superiormente mediante la cifra **3,80 [%]** ($= \varepsilon_{g_z}^{MATA1002195}$).

Aceleración de la gravedad registrada mediante el IMU MATA1002195

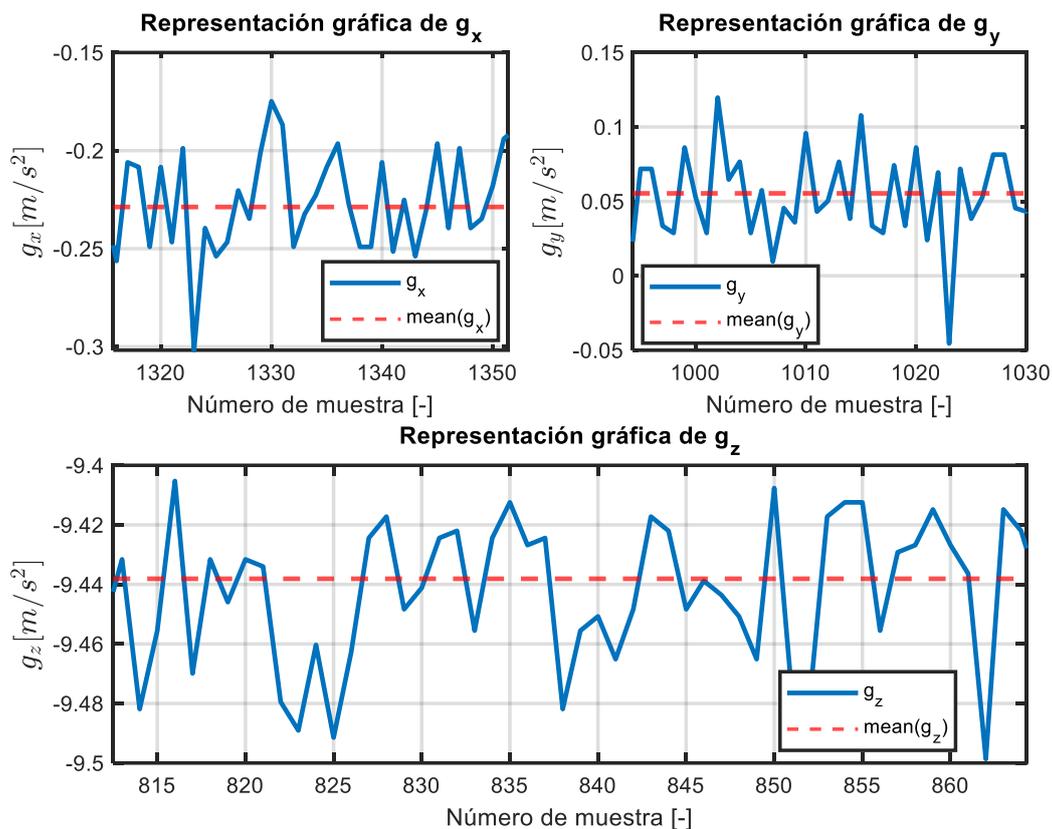


Figura 5.2.- Componentes cartesianas de la aceleración de la gravedad, registradas mediante el sensor inercial *MATA1002195*; el cual pertenece a la gama *MATRIX*.

5.1.2.- Ortogonalización de los sistemas de coordenadas asociados a cada sensor inercial

Uno de los pasos previos a cualquier calibración anatómica que se desee realizar sobre una unidad de medida inercial consiste en comprobar el nivel de ortogonalidad, presente en dicho *IMU*. Esto es, el estudio que debe efectuarse pasa por identificar el ángulo formado por cada par de ejes que conforman el *frame* cartesiano $\{X, Y, Z\}$ del sensor inercial.

De manera particular, se estudiará el nivel de ortogonalidad intrínseco al sistema cartesiano de referencia definido en el interior de dichos dispositivos.

Con objeto de abordar la actividad de calibración enunciada durante el párrafo anterior, se ha implementado¹⁶ el algoritmo detallado en la sección "11.3.- ESTRUCTURA MATEMÁTICA DE LA CALIBRACIÓN TRIDIMENSIONAL (3D)"; la cual puede consultarse dentro del apartado "*Anexo A: Fundamentos teóricos ligados a las técnicas de calibración sensorial*". Concretamente, el método computacional expuesto en dicha sección de la presente *Memoria* permite obtener como solución el siguiente vector de datos metrológicos: $[s, u_x, u_y, u_z, o]$.

En este escenario, se ha estudiado la ortogonalidad existente en cada una de las familias de sensores inerciales consideradas en este estudio: *Xsens DOT*, *MATRIX* y *SuperSens*. A continuación, se recogen los principales resultados obtenidos:

➤ *Ortogonalidad asociada a los IMUs de la gama Xsens DOT*

Con objeto de verificar el nivel de ortogonalidad existente en el sistema de coordenadas cartesianas vinculado un sensor inercial *Xsens DOT*, se ha realizado la captura de datos mostrada en la Tabla 5.2.

El protocolo experimental diseñado, durante la adquisición de los datos inerciales, se corresponde con varias rotaciones (γ_z) del sensor inercial alrededor de su eje Z. De manera particular, el parámetro γ_z toma los siguientes valores preestablecidos:

$$\gamma_z \in \{0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ\} \quad (5.4)$$

A continuación, se recoge el registro de datos realizado mediante el dispositivo inercial *XSens DOT7*. Téngase en cuenta que la información necesaria, para implementar el

¹⁶ **Nota:** El código fuente desarrollado para analizar el estado de ortogonalidad, inherente al sistema cartesiano de coordenadas de un sensor inercial, puede consultarse a través del apéndice indicado a continuación: "*Anexo F: Código Fuente construido para comprobar el estado de ortogonalidad intrínseco al *frame* cartesiano de cada IMU*".

proceso de ortogonalización planteado, se corresponde con la siguiente matriz de aceleración lineal cartesiana: $[\mathbf{Acc}_{X_{m \times 1}}, \mathbf{Acc}_{Y_{m \times 1}}, \mathbf{Acc}_{Z_{m \times 1}}]$.

Tabla 5.2.- Muestra de los datos capturados mediante el *IMU Xsens DOT7*, perteneciente a la familia de sensores *Xsens DOT*. Obsérvese que estos datos se corresponden con el protocolo experimental diseñado para comprobar la ortogonalidad existente en el *frame cartesiano de dicho sensor inercial*.

<i>Packet Counter</i>	<i>Sample TimeFine</i>	<i>Quat_W</i>	<i>Quat_X</i>	<i>Quat_Y</i>	<i>Quat_Z</i>	<i>Acc_X</i>	<i>Acc_Y</i>	<i>Acc_Z</i>	<i>Gyr_X</i>	<i>Gyr_Y</i>	<i>Gyr_Z</i>
1	1296765816	0.308877	-0.304444	-0.650356	0.623656	-0.001530	0.103020	-0.006737	-0.123482	-0.067595	-0.015765
2	1296782483	0.308961	-0.304368	-0.650093	0.623925	0.230834	-9.812376	-0.302961	0.872452	-1.218473	0.667858
3	1296799150	0.309023	-0.304255	-0.649977	0.624070	0.238591	-9.804417	-0.307012	0.870672	-1.167478	0.588420
4	1296815817	0.309052	-0.304140	-0.649957	0.624133	0.243156	-9.813269	-0.309342	0.755043	-1.144337	0.634574
5	1296832484	0.309073	-0.304026	-0.649963	0.624172	0.225606	-9.806728	-0.293527	0.834532	-1.139855	0.591660
6	1296849151	0.309060	-0.303952	-0.649934	0.624244	0.236724	-9.807701	-0.321841	0.802721	-1.151099	0.635701
7	1296865818	0.309015	-0.303878	-0.649939	0.624297	0.225238	-9.807431	-0.305049	0.799507	-1.176647	0.576574
8	1296882485	0.308938	-0.303802	-0.649930	0.624382	0.220857	-9.804692	-0.293071	0.879278	-1.209175	0.623945
9	1296899152	0.308876	-0.303742	-0.649906	0.624467	0.234743	-9.819306	-0.302329	0.808689	-1.187482	0.508150
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5684	1391484377	-0.409726	-0.306629	0.144427	-0.846903	6.265825	0.076416	7.350135	0.896041	-1.187802	0.606221

La evolución temporal asociada a las tres componentes cartesianas de la aceleración puede observarse gráficamente a través de la siguiente figura:

Datos de aceleración registrados mediante el IMU Xsens DOT7. Proceso de ortogonalización

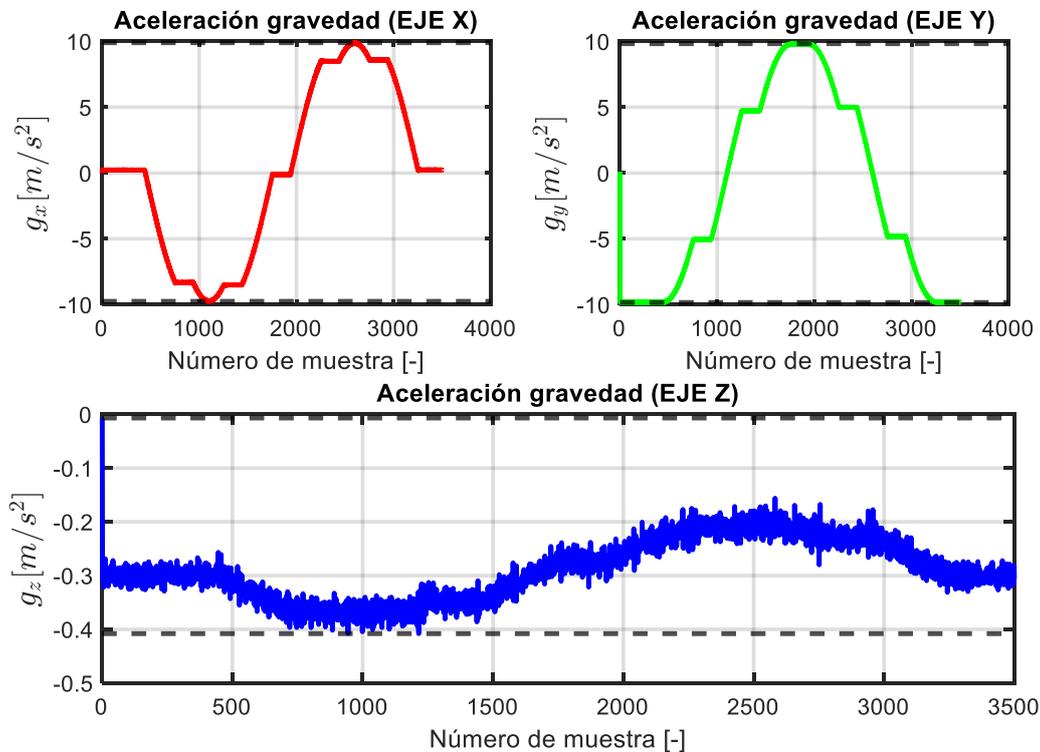


Figura 5.3.- Evolución temporal de las componentes cartesianas asociadas a la aceleración de la gravedad, las cuales se han registrado mediante el dispositivo de medida inercial *Xsens DOT 7*.

En conformidad con la naturaleza del movimiento realizado, el sistema lineal de ecuaciones planteado ha arrojado la siguiente solución:

$$[s, u_x, u_y, u_z, o]^T = [1, 0, 0, 1, 0]^T \quad (5.5)$$

El resultado analítico recogido en la expresión (5.5) verifica que la rotación del sensor *Xsens DOT7* se ha realizado alrededor del eje *Z*, tal y como se había definido previamente en el protocolo experimental implementado.

➤ *Ortogonalidad correspondiente a los sensores inerciales de la familia MATRIX*

Al igual que en el caso de los dispositivos *Xsens DOT*, el protocolo seguido durante la metodología experimental de ortogonalización (ligada al *IMU MATA1002662*) ha

consistido en una captura estática de datos inerciales de aceleración. Particularmente, se ha monitorizado el movimiento de rotación experimentado por el *IMU MATRIX* alrededor del eje cartesiano *Z*; local a dicha unidad de medida inercial. La posición angular γ_Z en el plano *XY*, vinculada al sensor inercial de *MATRIX*, satisface la siguiente restricción:

$$\gamma_Z \in \{0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ\} \quad (5.6)$$

Con todo ello, la matriz de datos generada por el sensor *MATRIX* durante la experiencia de laboratorio toma la siguiente anatomía:

Tabla 5.3.- Muestra de los datos capturados mediante el *IMU MATA1002662*, perteneciente a la familia de sensores *MATRIX*. Obsérvese que estos datos se corresponden con el protocolo experimental diseñado para comprobar la ortogonalidad existente en el *frame cartesiano de dicho sensor inercial*.

<i>dateTime</i>	<i>acc_x</i>	<i>acc_y</i>	<i>acc_z</i>	<i>gyr_x</i>	<i>gyr_y</i>	<i>gyr_z</i>	<i>bodySurface_temp</i>	<i>Ambient_temp</i>	<i>hr_raw</i>	<i>hr</i>	<i>remarks</i>
04/05/2023 09:16	0.99319437	-0.00073242	0.01562548	0.36622211	-2.13623047	1.40385144	20.0	18.0	8191	0	Medida del antebrazo
04/05/2023 09:16	0.99075289	-0.00439453	0.01464888	0.36622211	-2.01416016	1.46488845			8191	0	
04/05/2023 09:16	0.99685659	-0.00537109	0.01464888	0.36622211	-2.13623047	1.46488845			8191	0	
04/05/2023 09:16	0.99197363	-0.00561523	0.01464888	0.30518509	-2.19726562	1.46488845			8191	0	
04/05/2023 09:16	0.99172948	-0.00488281	0.01342814	0.36622211	-2.07519531	1.52592547			8191	0	
04/05/2023 09:16	0.99783319	-0.00976562	0.01318400	0.36622211	-2.13623047	1.46488845			8191	0	
04/05/2023 09:16	0.99832148	-0.00292969	0.01611377	0.30518509	-2.13623047	1.46488845			2162	0	
04/05/2023 09:16	0.99465926	-0.00512695	0.01098666	0.24414808	-2.13623047	1.46488845			2158	0	
04/05/2023 09:16	0.99954222	-0.00048828	0.01123081	0.36622211	-2.19726562	1.46488845			2154	0	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
04/05/2023 09:17:23	0.50587481	-0.86352539	0.01855525	0.24414808	-2.31933594	1.46488845					

En el presente caso de estudio, los datos de interés se corresponden con los valores inerciales recogidos en las columnas *acc_x*, *acc_y* y *acc_z* de la Tabla 5.3; es decir, las tres componentes cartesianas de la aceleración registrada por el *IMU MATRIX*.

De forma ilustrativa, se ha representado gráficamente la evolución temporal ligada a los parámetros cartesianos $[a_x, a_y, a_z]$; la cual se muestra en la siguiente figura:

Datos de aceleración registrados mediante el IMU MATA1002662. Proceso de ortogonalización

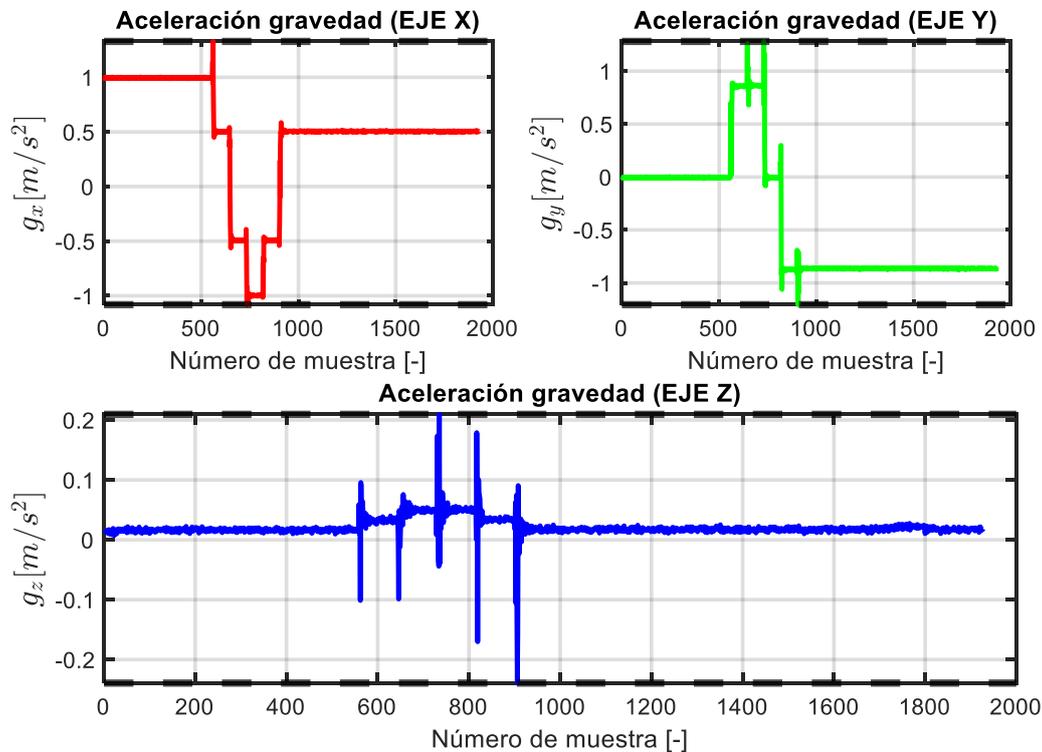


Figura 5.4.- Evolución temporal de las componentes cartesianas asociadas a la aceleración de la gravedad, las cuales se han registrado mediante el dispositivo de medida inercial *MATA1002662*.

Con todo ello, se tiene que la ganancia (s) del sensor es unitaria y el *offset* (o) adquiere valor nulo. Por otra parte, las componentes cartesianas del vector director inherente a la orientación del sensor inercial toman la siguiente forma: $\vec{u} = [u_x, u_y, u_z]^T = [0, 0, 1]^T$. Este resultado corrobora que el movimiento rotacional se ha efectuado alrededor del eje Z, definido en el sensor inercial *MATA1002662*.

5.2.- CALIBRACIÓN DE LOS SENSORES INERCIALES

Tras ejecutar los protocolos de calibración indicados anteriormente en el apartado "3.2.- DESCRIPCIÓN DEL EXPERIMENTO DE CALIBRACIÓN INERCIAL", se han generado los siguientes resultados numéricos:

Tabla 5.4.- Valores numéricos (media) obtenidos como resultado de los experimentos de calibración inercial llevados a cabo.

	Familia de sensores	$ \beta_{g_x} $ [mg]	$ \beta_{g_y} $ [mg]	$ \beta_{g_z} $ [mg]
CALIBRACIÓN 2D	<i>Xsens DOT</i>	1.62	10.15	14.96
	<i>MATRIX</i>	1.02	1.02	3.06
CALIBRACIÓN 3D	<i>Xsens DOT</i>	21.20	17.78	8.81
	<i>MATRIX</i>	26.20	10.78	20.84

Tabla 5.5.- Valores numéricos (desviación típica) obtenidos como resultado de los experimentos de calibración inercial llevados a cabo.

	Familia de sensores	σ_{g_x} [mg]	σ_{g_y} [mg]	σ_{g_z} [mg]
CALIBRACIÓN 2D	<i>Xsens DOT</i>	1.02	0.97	0.66
	<i>MATRIX</i>	2.77	3.03	3.02
CALIBRACIÓN 3D	<i>Xsens DOT</i>	1.86	0.92	0.81
	<i>MATRIX</i>	3.23	4.32	2.79

A continuación, se detalla la notación introducida en la Tabla 5.4 y la Tabla 5.5:

- ❖ $|\beta_{g_i}|$ [mg] = $|\hat{\mu}_{g_i} - \mu_{g_i}|$ [mg], $\forall i \in \{x, y, z\} \equiv$ Valor absoluto de la diferencia existente entre las componentes (cartesianas) teóricas de \vec{g} y los valores experimentales registrados para dicha aceleración de la gravedad.
- ❖ $\hat{\mu}_{g_i}$ [mg] \equiv Cantidad teórica ($\hat{\mu}_{g_x} = \hat{\mu}_{g_y} = 0$ [mg], $\hat{\mu}_{g_z} = 1000$ [mg]) asociada a la respectiva componente cartesiana de la aceleración de la gravedad.
- ❖ μ_{g_i} [mg] \equiv Valor experimental obtenido para la componente cartesiana i de \vec{g} .
- ❖ σ_{g_i} [mg] \equiv Desviación típica (experimental) asociada a la información inercial contenida en la variable $|\beta_{g_i}|$.

Tras ejecutar los protocolos de calibración 2D y 3D, se verifica que el valor medio de error vinculado a las medidas proporcionadas por los sensores inerciales es inferior a 27 [mg]. Por otro lado, las desviaciones típicas de las medidas inerciales no se han modificado significativamente.

A continuación, se muestra gráficamente el resultado obtenido tras la calibración de los datos inerciales de aceleración:

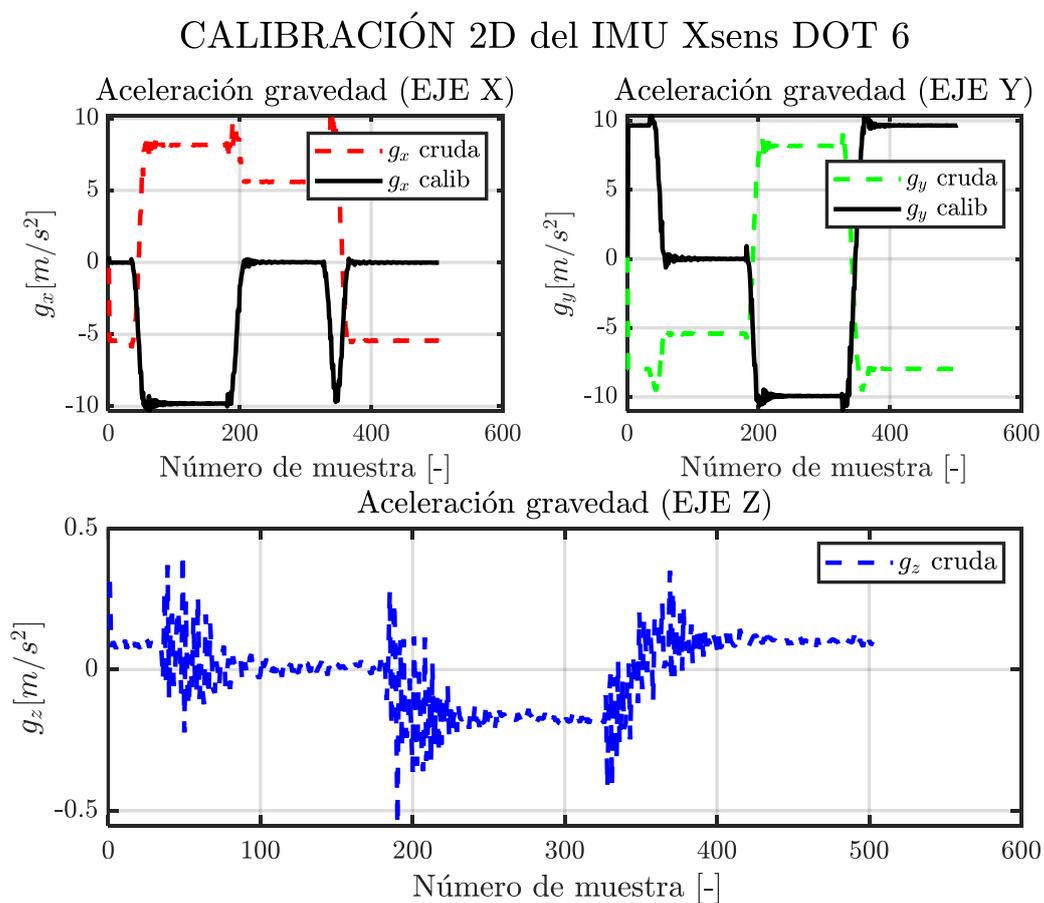


Figura 5.5.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. *Nota: se muestran las tres componentes cartesianas de \vec{g} , capturadas mediante el dispositivo Xsens DOT 6 y calibradas mediante un protocolo 2D.*

La matriz de calibración adquiere la siguiente estructura:

$$[R]_{2D} = \begin{bmatrix} -0,8260 & 0,5654 \\ -0,5532 & -0,8342 \end{bmatrix} \quad (5.7)$$

CALIBRACIÓN 3D del IMU Xsens DOT 8

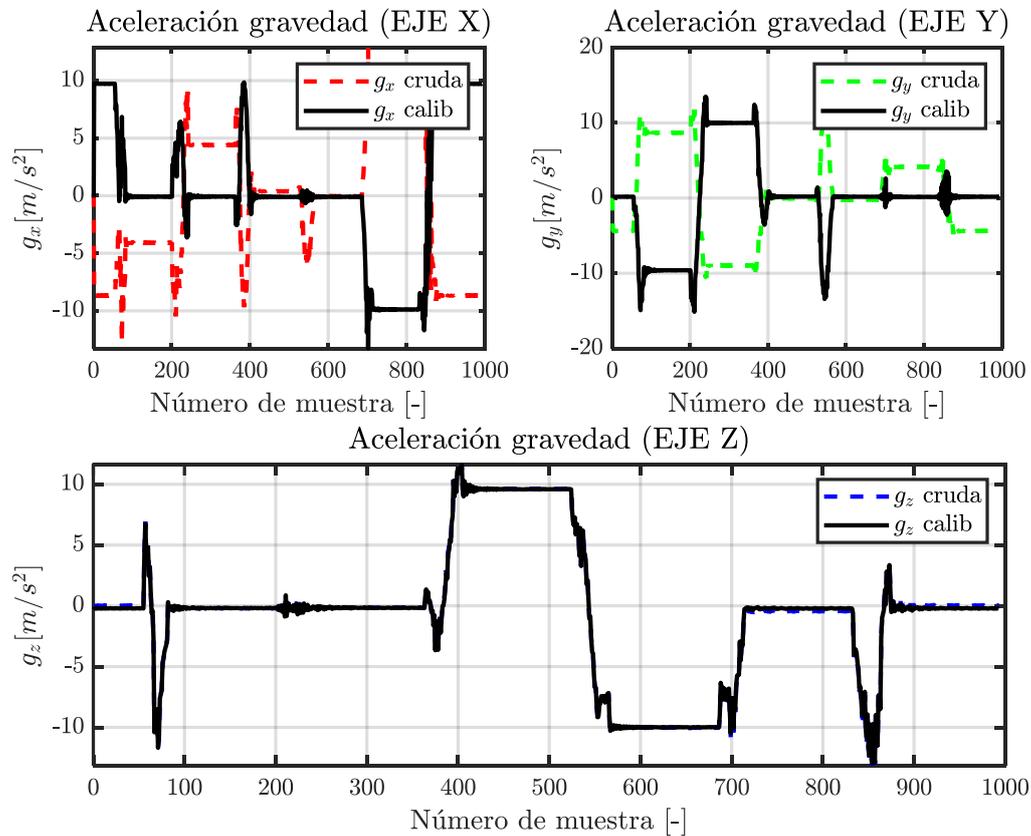


Figura 5.6.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. *Nota: se muestran las tres componentes cartesianas de \vec{g} , capturadas mediante el dispositivo Xsens DOT 8 y calibradas mediante un protocolo 3D.*

La matriz de calibración obtenida es:

$$[R]_{3D} = \begin{bmatrix} -0,9014 & -0,4342 & 0,0226 \\ 0,4347 & -0,9013 & -0,0009 \\ 0,0202 & 0,0108 & 0,9958 \end{bmatrix}_{3 \times 3} \quad (5.8)$$

CALIBRACIÓN 2D del IMU MATA 1002195 (MATRIX)

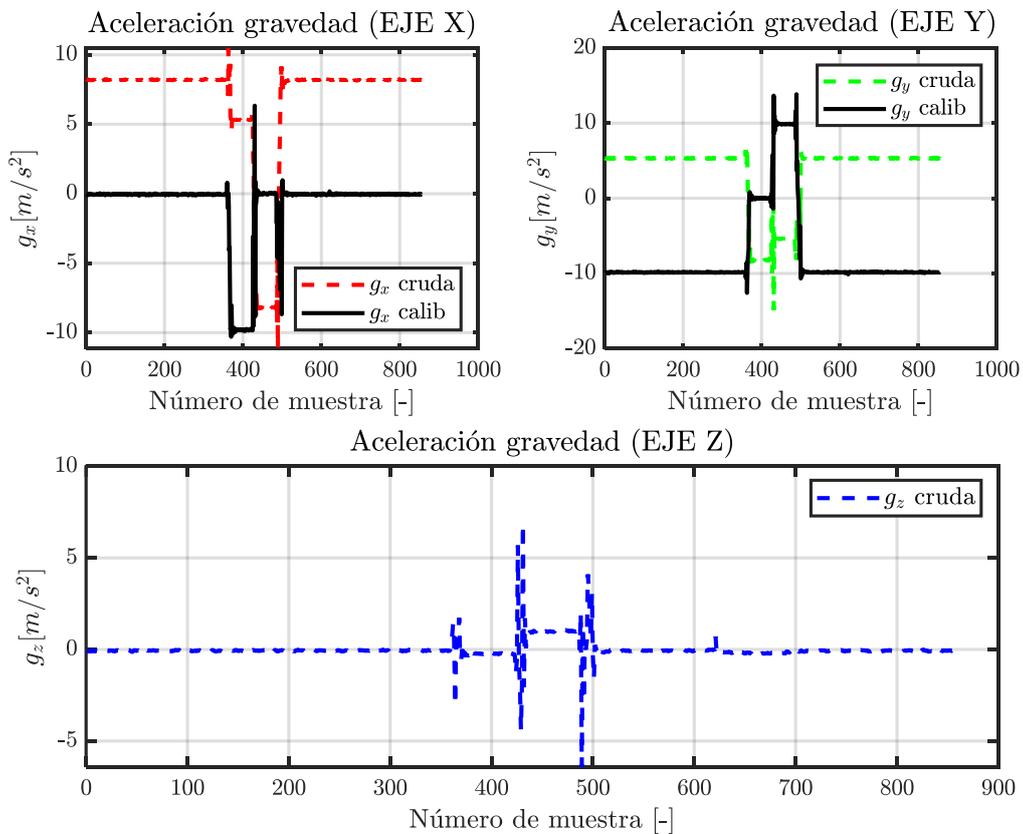


Figura 5.7.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. *Nota: se muestran las tres componentes cartesianas de \vec{g} , capturadas mediante el dispositivo MATA 1002195 (perteneciente a la gama MATRIX) y calibradas mediante un protocolo 2D.*

La matriz de calibración obtenida es:

$$[R]_{2D} = \begin{bmatrix} -0,5492 & 0,8368 \\ -0,8471 & -0,5454 \end{bmatrix}_{2 \times 2} \quad (5.9)$$

CALIBRACIÓN 3D del IMU MATA 1002662

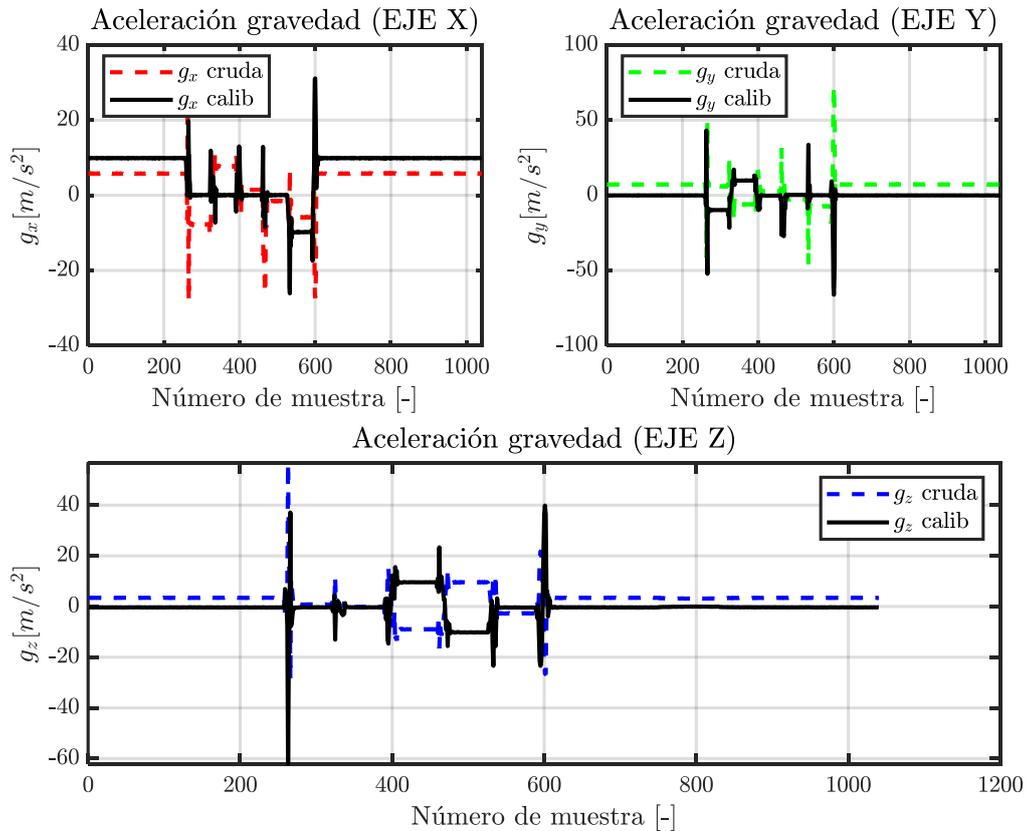


Figura 5.8.- Comparación gráfica entre los valores crudos y calibrados de la aceleración de la gravedad. *Nota: se muestran las tres componentes cartesianas de \vec{g} , capturadas mediante el dispositivo MATA 100262 (perteneciente a la gama MATRIX) y calibradas mediante un protocolo 3D.*

La matriz de calibración obtenida es:

$$[R]_{3D} = \begin{bmatrix} 0,5976 & 0,7454 & 0,3161 \\ 0,7829 & -0,6264 & -0,0283 \\ 0,1555 & 0,2835 & -0,9571 \end{bmatrix}_{3 \times 3} \quad (5.10)$$

5.3.- EVALUACIÓN DE LOS SENSORES Y LOS ALGORITMOS DE ESTIMACIÓN INERCIAL

En este apartado se recogen los resultados generados durante la evaluación inercial de los métodos computacionales y de los sensores contemplados en este estudio.

5.3.1.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: “ZXZ”). Algoritmo de estimación de orientación: Ángulos de Euler suministrados por cada dispositivo Xsens DOT (filtro propietario). Algoritmo de evaluación inercial: diferencia de cuaterniones

A continuación, se recogen los resultados obtenidos para la primera alternativa de evaluación inercial. Recuérdese que la desalineación introducida ($\delta_{DESALINEACIÓN}$) es $\{-20^\circ, -18^\circ, -16^\circ, \dots, 0^\circ, 16^\circ, 18^\circ, 20^\circ\}$, mientras los ángulos de flexión (λ) pertenecen al intervalo $[0^\circ, 70^\circ]$; siendo $\Delta\lambda_{FLEXIÓN} = 5^\circ$.

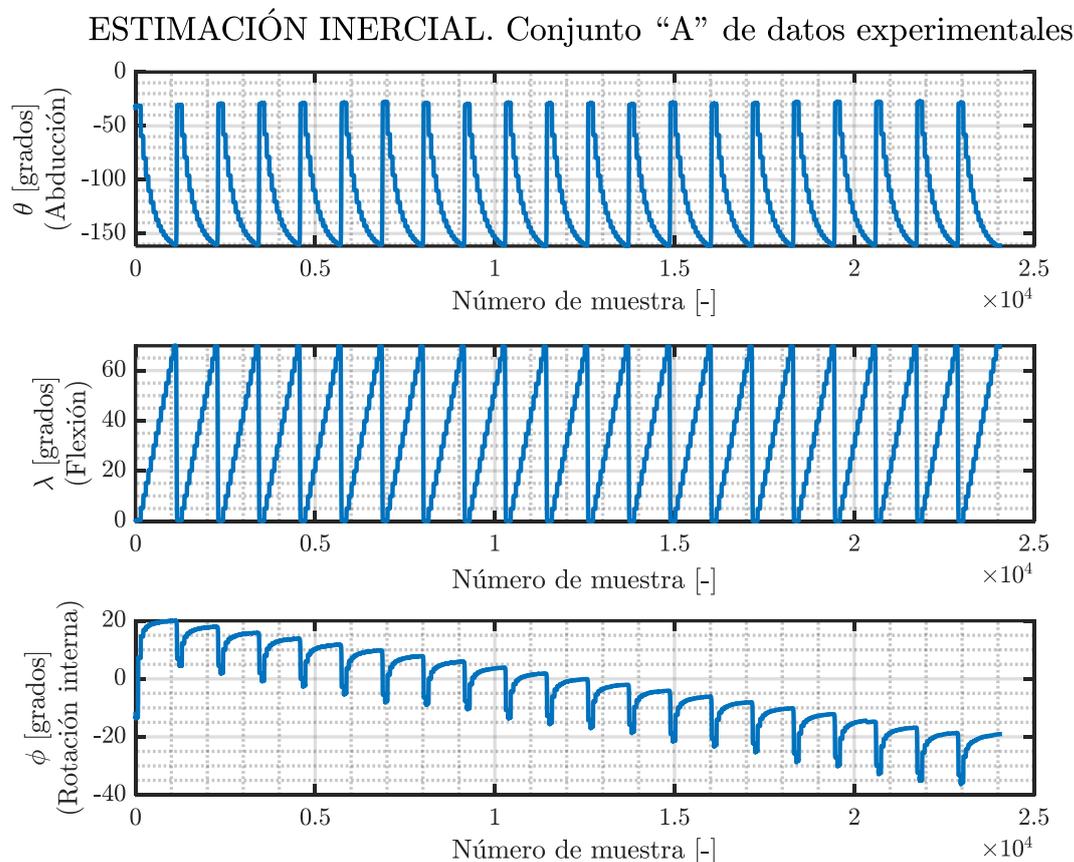


Figura 5.9.- Estimación inercial del movimiento articular realizado con el *cobot UR3e*, la cual se ha expresado mediante sus tres Ángulos de Euler. *NOTA: Este resultado se corresponde con el conjunto “A” de datos experimentales.*

Maquetando una vista detallada para los casos en los que el ángulo de desalineación ($\delta_{DESALINEACIÓN}$) toma los valores -20° , 0° ó 20° , se ha obtenido la siguiente figura:

ESTIMACIÓN INERCIAL (VISTA DE DETALLE).

Conjunto "A" de datos experimentales

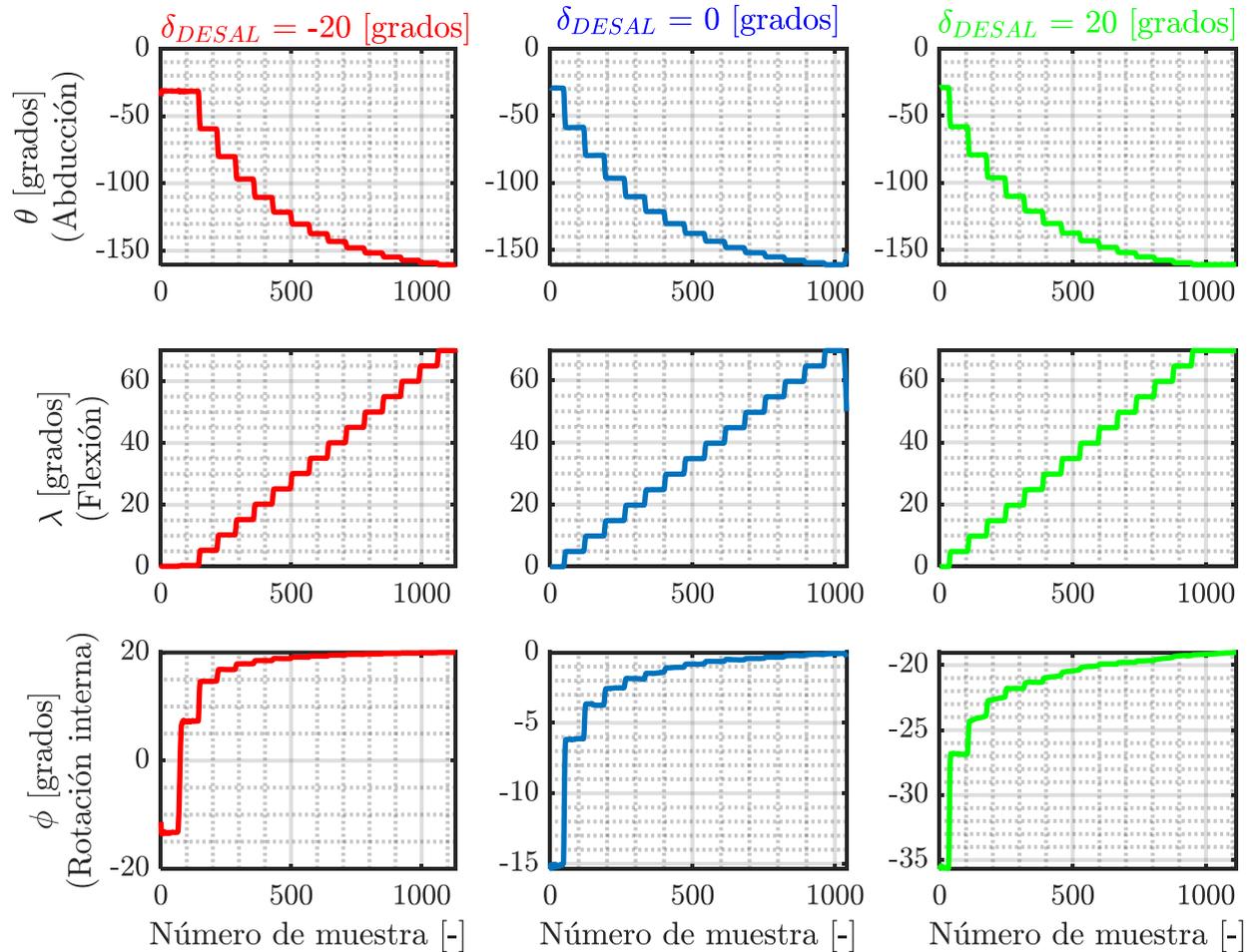


Figura 5.10.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción (θ°), flexión (λ°) y rotación interna (ϕ°). *NOTA: Este resultado se corresponde con el conjunto "A" de datos experimentales.*

En las figuras anteriores puede observarse que utilizando los cuaterniones proporcionados los *IMUs Xsens DOT* y el algoritmo "Diferencia de cuaterniones", se obtiene una óptima estimación de los ángulos de flexión. Asimismo, el efecto de la desalineación introducida cada 2° aparece reflejado en el tercer *Ángulo de Euler* (ϕ , rotación interna).

Numéricamente, se han obtenido los siguientes resultados:

Tabla 5.6.- Errores generados durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (*EXP_1*, *Xsens DOT*, *filtro propietario*, *diferencia de cuaterniones*).

IMUs: Xsens DOT Algoritmo de evaluación inercial: <i>Diferencia de cuaterniones (filtro propietario)</i> Orden de obtención de los Ángulos de Euler: "ZXZ" $\delta_{DESALINEACIÓN} = 0^\circ$															
Parámetro estadístico	λ [grados] (Flexión)														
	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
$\mu_{ e }$ [°]	0.00	0.09	0.13	0.15	0.18	0.21	0.21	0.22	0.24	0.27	0.29	0.33	0.36	0.41	0.46
$10 \cdot \sigma_{ e }$ [°]	0.02	0.03	0.06	0.07	0.03	0.05	0.05	0.04	0.01	0.02	0.02	0.03	0.10	0.03	0.05

$\mu_{|e|}$ [°] se corresponde con la media del error cometido durante la estimación inercial, expresada en grados.
 $\sigma_{|e|}$ [°] denota la desviación típica del error presente en dicha estimación inercial, expresada en grados.

De la tabla anterior se desprende que el error de estimación inercial se incrementa linealmente con el ángulo de flexión λ . Finalmente, cabe destacar que dicho error de estimación presenta un orden de magnitud de décimas de grado; ofreciendo una gran estabilidad para cada ángulo de flexión λ .

5.3.2.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: "YXZ"). Algoritmo de estimación de orientación: *imufilter*. Algoritmo de evaluación inercial: diferencia de cuaterniones

Modificando la secuencia utilizada durante la obtención de los Ángulos de Euler a partir de los cuaterniones de orientación, puede comprobarse que las estimaciones inerciales realizadas se ven alteradas. En este caso, el orden de conversión "YXZ" y el algoritmo *imufilter* (*datos inerciales empleados: aceleración + velocidad angular de giro*) han arrojado los siguientes resultados:

ESTIMACIÓN INERCIAL. IMUs Xsens DOT. Algoritmo imufilter

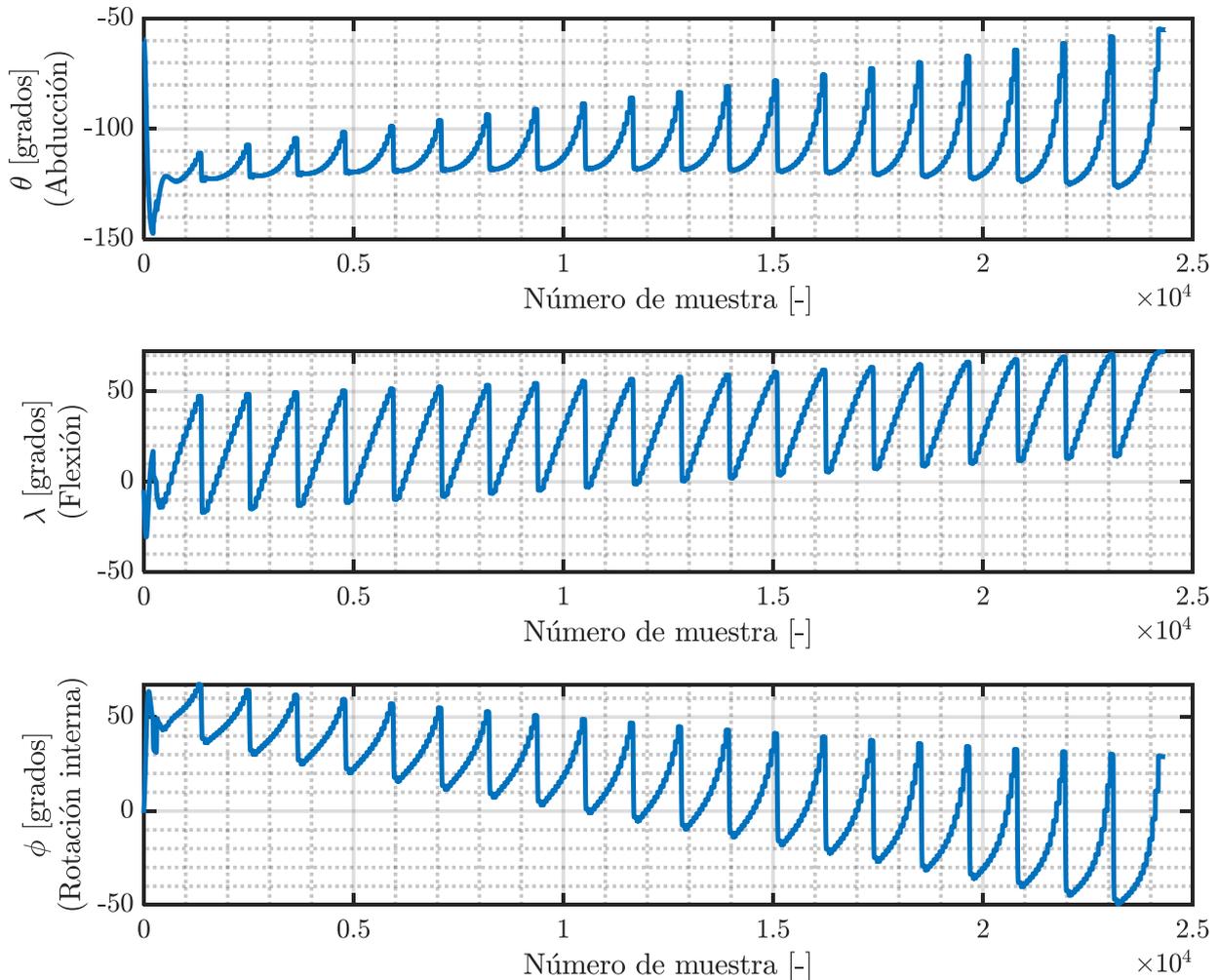


Figura 5.11.- Estimación inercial del movimiento articular realizado con el *cobot UR3e*, la cual se ha expresado mediante sus tres *Ángulos de Euler* (orden de reconstrucción “ZXZ”). *NOTA: Se ha utilizado el algoritmo “imufilter” para calcular la orientación de cada IMU Xsens DOT.*

Efectuando un *zoom* para los casos en los que $\delta_{DESALINEACIÓN} \in \{-20^\circ, 0^\circ, 20^\circ\}$, resulta la siguiente representación gráfica:

ESTIMACIÓN INERCIAL (VISTA DE DETALLE) IMUs Xsens DOT. Algoritmo imufilter

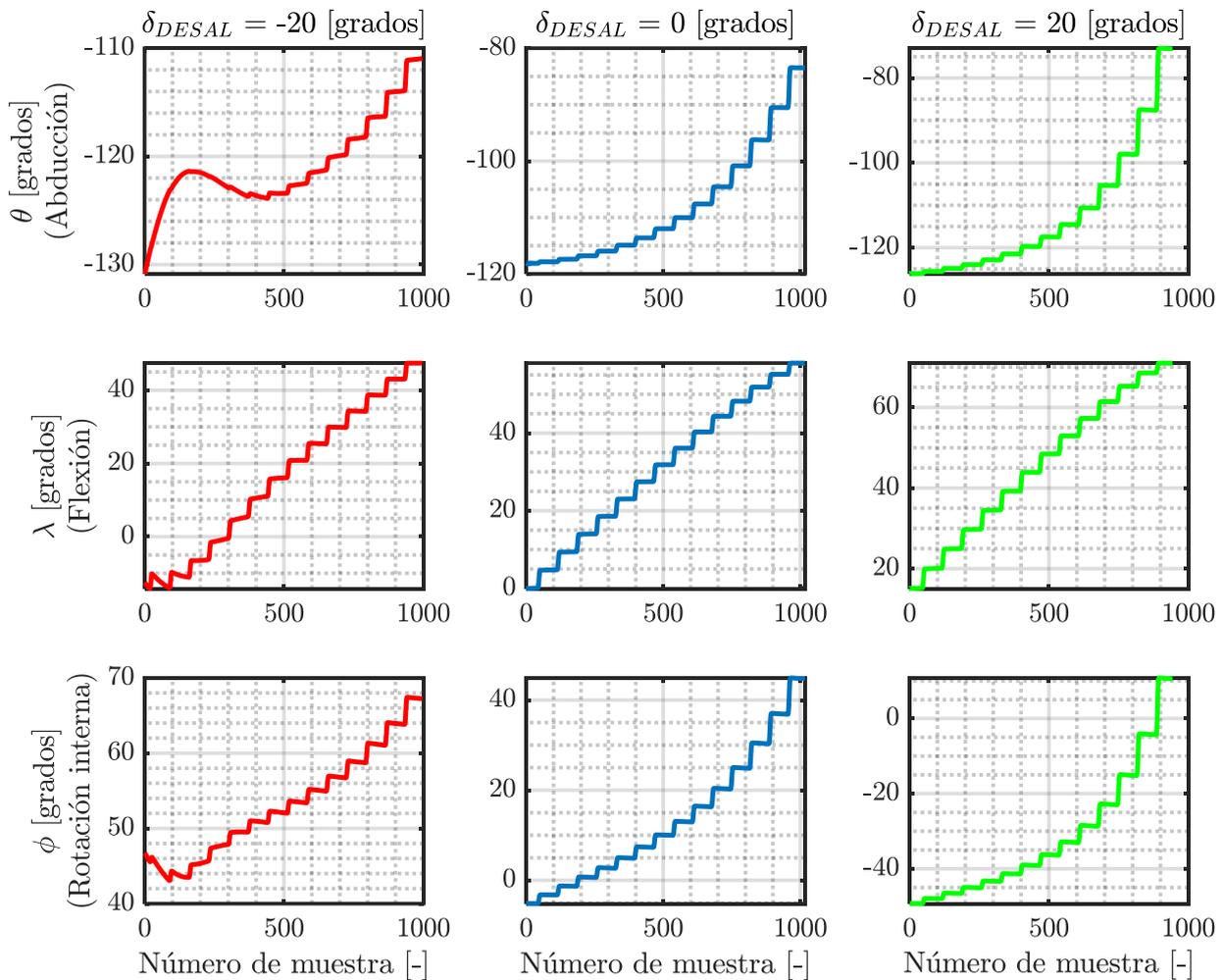


Figura 5.12.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción (θ°), flexión (λ°) y rotación interna (ϕ°). *NOTA: Se han utilizado los IMUs Xsens DOT y el algoritmo “imufilter” durante la estimación de la orientación.*

A continuación, se recogen algunos resultados numéricos:

Tabla 5.7.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (*EXP_1*, *Xsens DOT*, *imufilter*, *diferencia de cuaterniones*).

IMUs: <i>Xsens DOT</i> Algoritmo de evaluación inercial: <i>Diferencia de cuaterniones (imufilter)</i> Orden de obtención de los Ángulos de Euler: “YXZ” $\delta_{DESALINEACIÓN} = 0^\circ$															
Parámetro estadístico	λ [grados] (Flexión)														
	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
$\mu_{ \epsilon }$ [°]	3.68	3.42	3.07	2.68	2.22	1.71	1.16	0.53	0.20	1.01	1.97	3.11	4.47	6.17	8.29
$10 \cdot \sigma_{ \epsilon }$ [°]	0.5	0.3	0.2	0.2	0.04	0.1	0.1	0.05	0.02	0.03	0.02	0.04	0.05	0.02	0.01

$\mu_{|\epsilon|}$ [°] se corresponde con la media del error cometido durante la estimación inercial, expresada en grados.
 $\sigma_{|\epsilon|}$ [°] denota la desviación típica del error presente en dicha estimación inercial, expresada en grados.

En caso de emplear el algoritmo *imufilter* para obtener los cuaterniones de orientación de cada *IMU Xsens DOT*, el error en la estimación inercial de la flexión presenta un valor máximo de $8,29^\circ$; siendo $\lambda = 70^\circ$.

Las desviaciones típicas generadas en la señal de error pueden considerarse despreciables.

5.3.3.- *IMUs Xsens DOT* (orden de obtención de los Ángulos de Euler: “ZXZ”). Algoritmo de estimación de orientación: *ecompass*. Algoritmo de evaluación inercial: *diferencia de cuaterniones*

En este caso, se han obtenido los cuaterniones de orientación asociados a cada *IMU* mediante la algoritmia *ecompass*. A continuación, se muestran los resultados obtenidos:

ESTIMACIÓN INERCIAL. IMUs Xsens DOT.
Algoritmos: ecompass, diferencia de cuaterniones

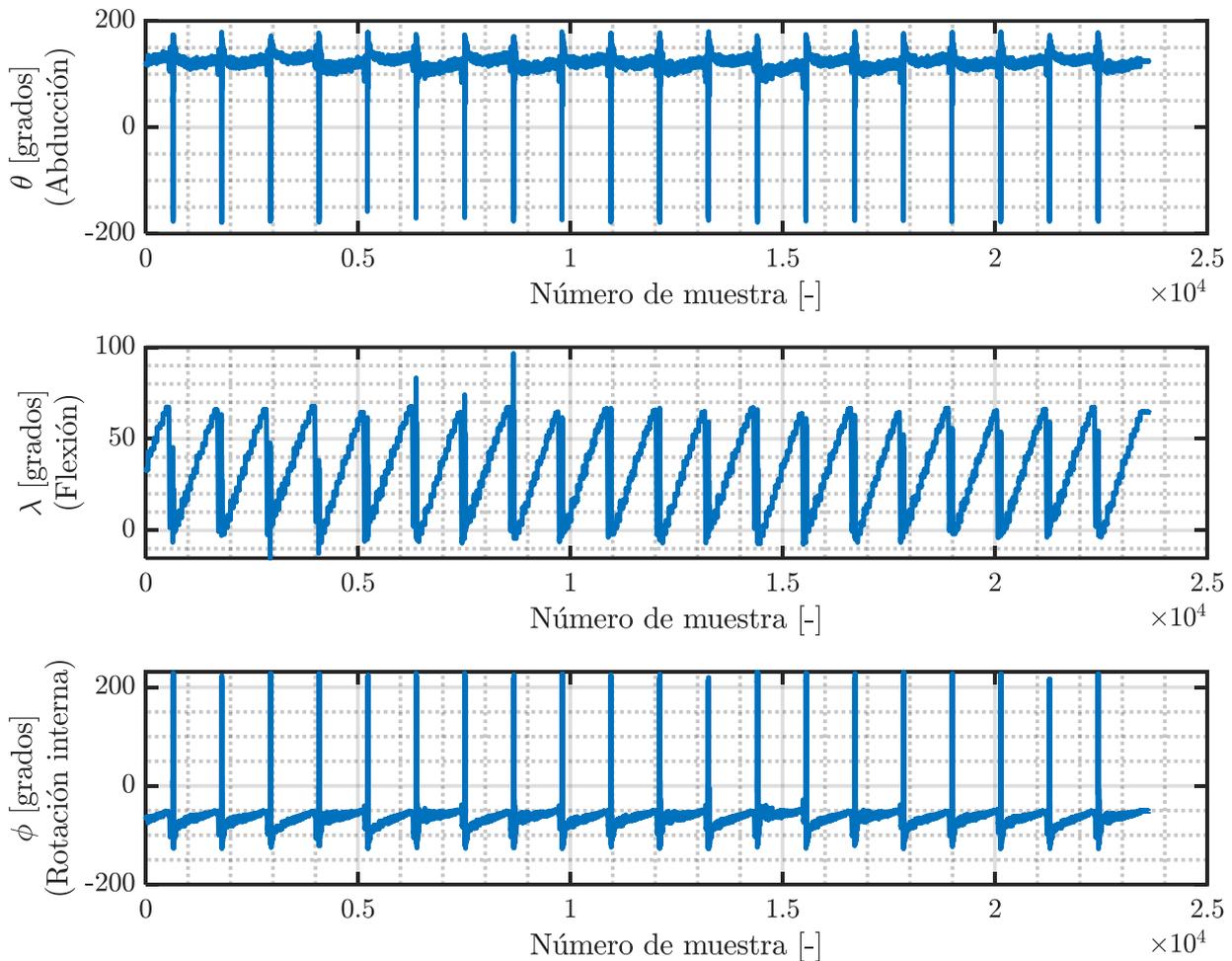


Figura 5.13.- Estimación inercial del movimiento articular realizado con el *cobot UR3e*, la cual se ha expresado mediante sus tres *Ángulos de Euler* (orden de reconstrucción “ZXZ”). *NOTA: Se ha utilizado el algoritmo “ecompass” para calcular la orientación de cada IMU Xsens DOT. El método de evaluación inercial ha sido la “diferencia de cuaterniones”.*

Realizando un *zoom* para los casos en los que $\delta_{DESALINEACIÓN} \in \{-20^\circ, 0^\circ, 20^\circ\}$, se ha construido la siguiente representación gráfica:

ESTIMACIÓN INERCIAL (DETALLE). IMUs Xsens DOT.

Algoritmos: ecompass, diferencia de cuaterniones

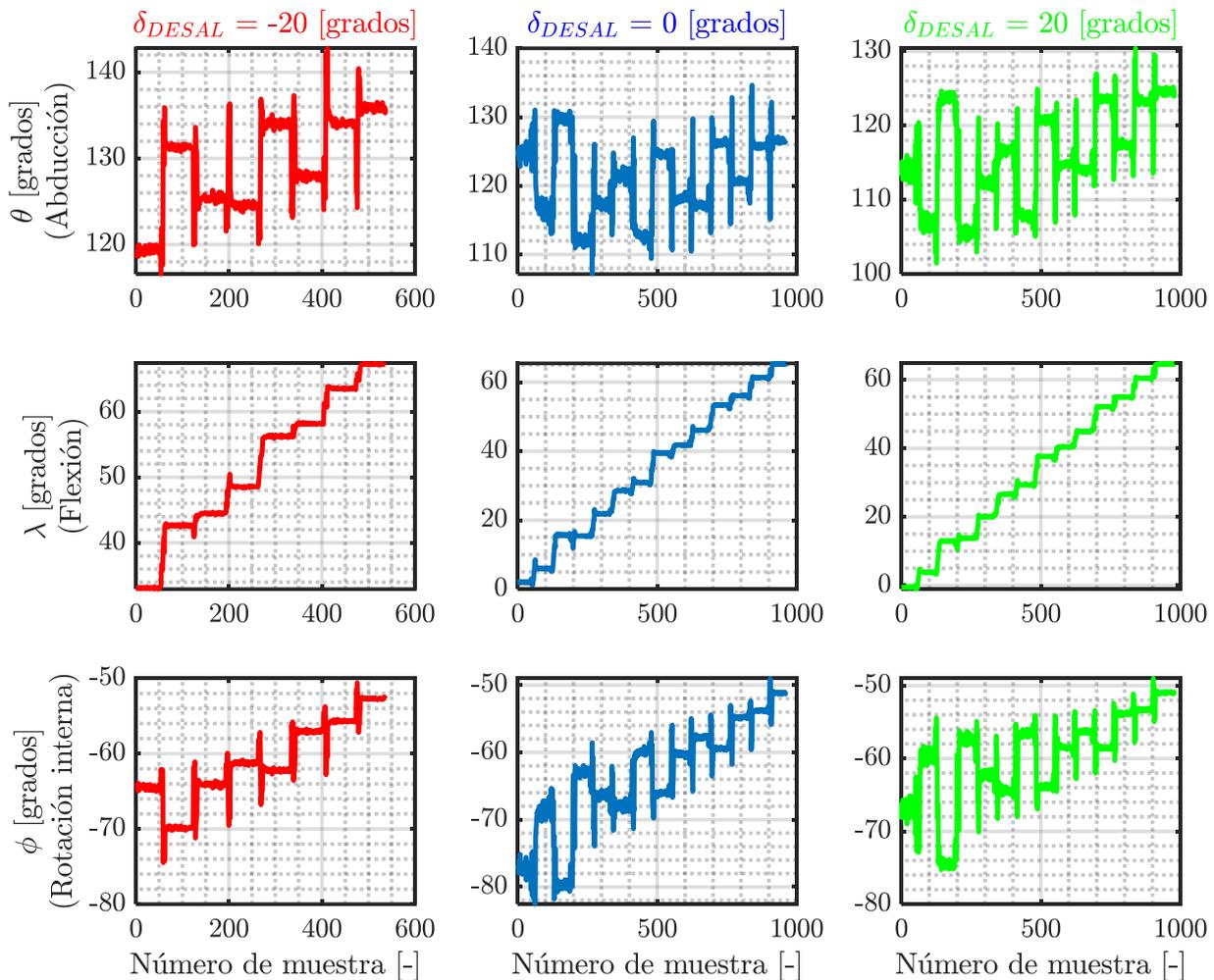


Figura 5.14.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción (θ°), flexión (λ°) y rotación interna (ϕ°). *NOTA: Se han utilizado los IMUs Xsens DOT junto con los algoritmos “ecompass” y “diferencia de cuaterniones”.*

Las señales de estimación inercial obtenidas presentan un nivel de ruido bastante notorio.

Por otra parte, se han generado los siguientes resultados numéricos:

Tabla 5.8.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (*EXP_2*, *Xsens DOT*, *ecompass*, *diferencia de cuaterniones*).

IMUs: Xsens DOT Algoritmo de evaluación inercial: <i>Diferencia de cuaterniones (ecompass)</i> Orden de obtención de los Ángulos de Euler: “ZXZ” $\delta_{DESALINEACIÓN} = 0^\circ$															
Parámetro estadístico	λ [grados] (Flexión)														
	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
$\mu_{ \epsilon }$ [°]	2.00	0.95	5.74	0.45	1.80	3.56	0.91	4.48	1.80	1.10	3.42	1.10	1.38	0.28	--
$10 \cdot \sigma_{ \epsilon }$ [°]	1.11	1.01	1.27	0.67	0.93	1.30	0.70	0.95	0.78	0.58	0.78	0.72	0.70	0.81	--

$\mu_{|\epsilon|}$ [°] se corresponde con la media del error cometido durante la estimación inercial, expresada en grados.
 $\sigma_{|\epsilon|}$ [°] denota la desviación típica del error presente en dicha estimación inercial, expresada en grados.

Se verifica que, si no hay desalineación sensor - segmento, el error cometido en la estimación inercial de los ángulos de flexión se encuentra acotado por el intervalo [0.28°, 5.74°]. La desviación estándar asociada oscila en torno a las décimas de grado.

5.3.4.- IMUs Xsens DOT (orden de obtención de los Ángulos de Euler: “ZYZ”). Algoritmo de estimación de orientación: *filtro propietario*. Algoritmo de evaluación inercial: *TRIAD*

A continuación, se recoge la estimación inercial realizada a través del algoritmo *TRIAD* (cuya implementación también se ha efectuado en *MatLab*).

ESTIMACIÓN INERCIAL - IMUs Xsens DOT.
Algoritmo “TRIAD” (Acelerómetro + Magnetómetro)

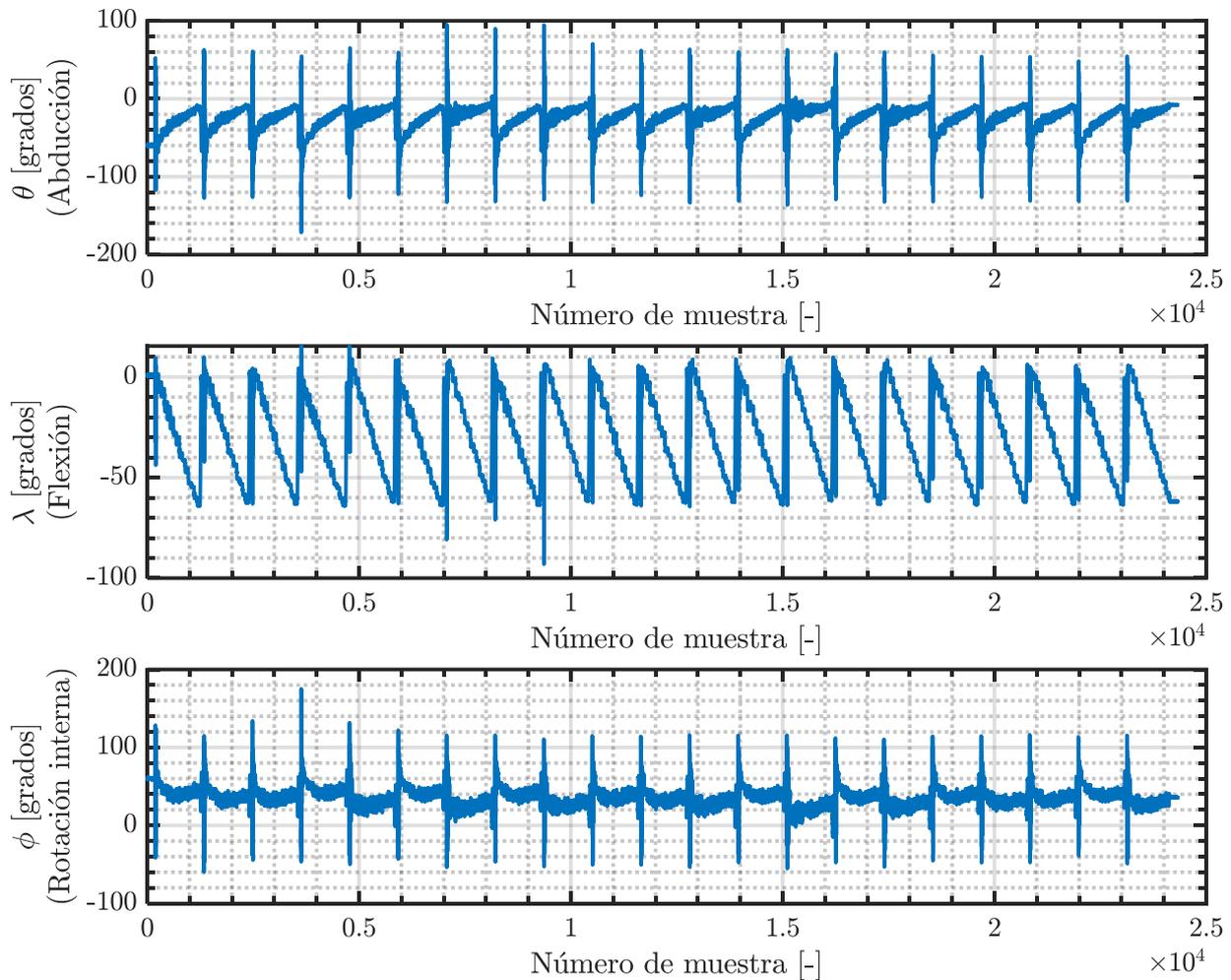


Figura 5.15.- Estimación inercial del movimiento articular realizado con el *cobot UR3e*, la cual se ha expresado mediante sus tres *Ángulos de Euler* (orden de reconstrucción “ZYZ”). *NOTA: Se ha utilizado el algoritmo “triad” para calcular el ángulo de flexión del codo.*

Efectuando un *zoom* para los casos en los que $\delta_{DESALINEACIÓN} \in \{-20^\circ, 0^\circ, 20^\circ\}$, se ha maquetado la siguiente representación gráfica:

ESTIMACIÓN INERCIAL (VISTA DE DETALLE).
Algoritmo “TRIAD” (Acelerómetro + Magnetómetro)

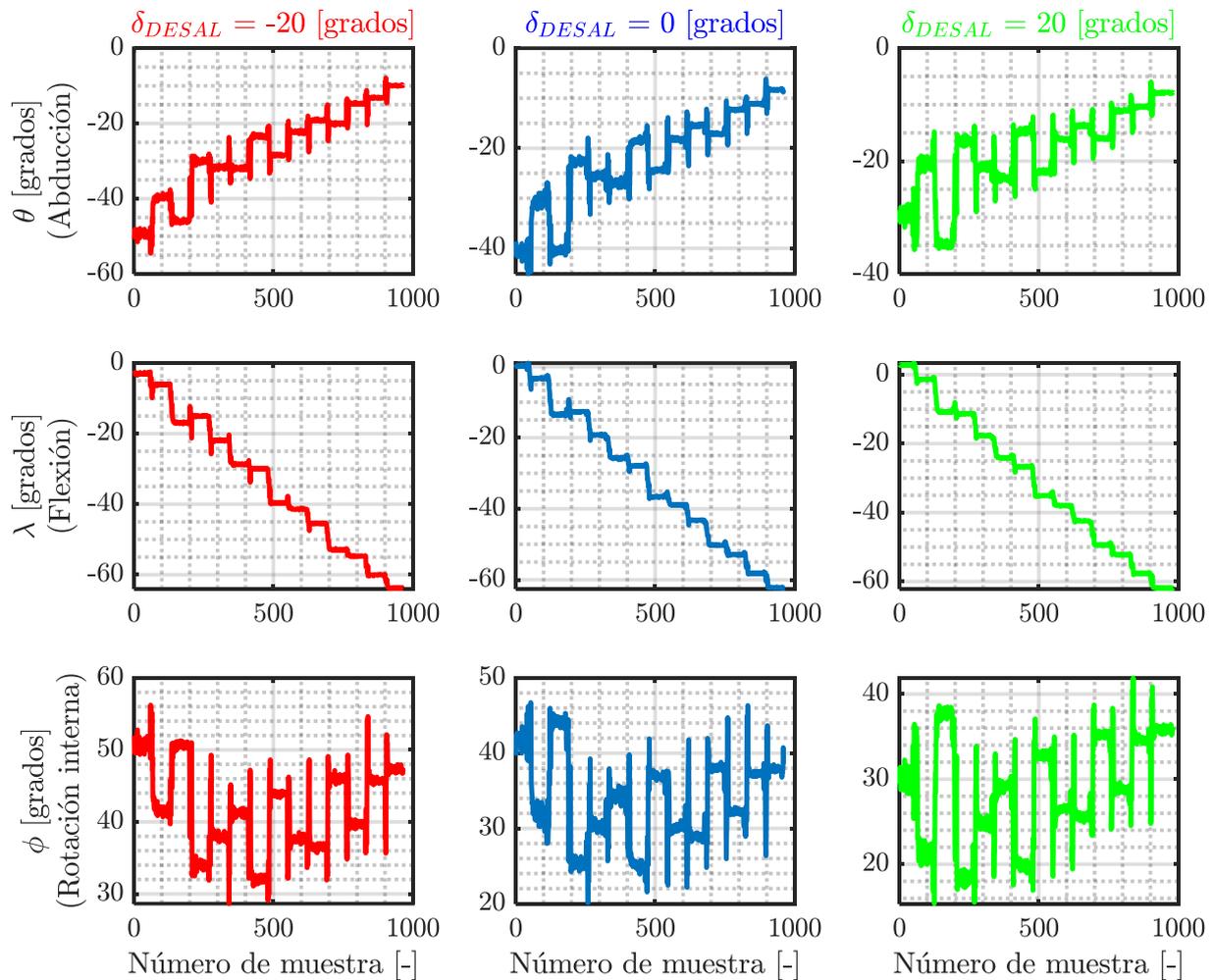


Figura 5.16.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción (θ°), flexión (λ°) y rotación interna (ϕ°). *NOTA: Se han utilizado los IMUs Xsens DOT y el algoritmo “TRIAD” durante la estimación de los tres ángulos de rotación.*

En la siguiente tabla se muestran varios resultados numéricos asociados al experimento de evaluación inercial:

Tabla 5.9.- Errores cometidos durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (*EXP_3*, *Xsens DOT*, *imufilter*, *TRIAD*).

IMUs: <i>Xsens DOT</i> Algoritmo de evaluación inercial: <i>TRIAD</i> Orden de obtención de los Ángulos de Euler: “ <i>ZYZ</i> ” $\delta_{DESALINEACIÓN} = 0^\circ$															
Parámetro estadístico	λ [grados] (Flexión)														
	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
$\mu_{ \epsilon }$ [°]	0.26	1.72	3.46	2.32	0.79	0.77	2.12	1.61	1.06	1.78	0.29	2.10	1.83	2.83	--
$10 \cdot \sigma_{ \epsilon }$ [°]	1.60	1.17	1.70	0.78	1.06	1.35	0.72	0.11	0.80	0.72	0.81	0.77	0.68	0.82	--

$\mu_{|\epsilon|}$ [°] se corresponde con la media del error cometido durante la estimación inercial, expresada en grados.
 $\sigma_{|\epsilon|}$ [°] denota la desviación típica del error presente en dicha estimación inercial, expresada en grados.

El método *TRIAD*, en combinación con los *IMUs Xsens DOT*, también permite observar el ruido presente en el campo magnético del robot. En este caso, el efecto de la desalineación se recoge en el primer *Ángulo de Euler*. La estimación inercial de la flexión λ ofrece errores medios inferiores a 3.46° , en la situación de no existir desalineación sensor - segmento. Nuevamente, la desviación típica obtenida está en el orden de las décimas de grado.

5.3.5.- *IMUs MATRIX* (orden de obtención de los *Ángulos de Euler*: “*ZXZ*”). Algoritmo de estimación de orientación: *imufilter*. Algoritmo de evaluación inercial: diferencia de cuaterniones

A continuación, se presentan los resultados obtenidos para las unidades de medida inercial pertenecientes a la gama *MATRIX*.

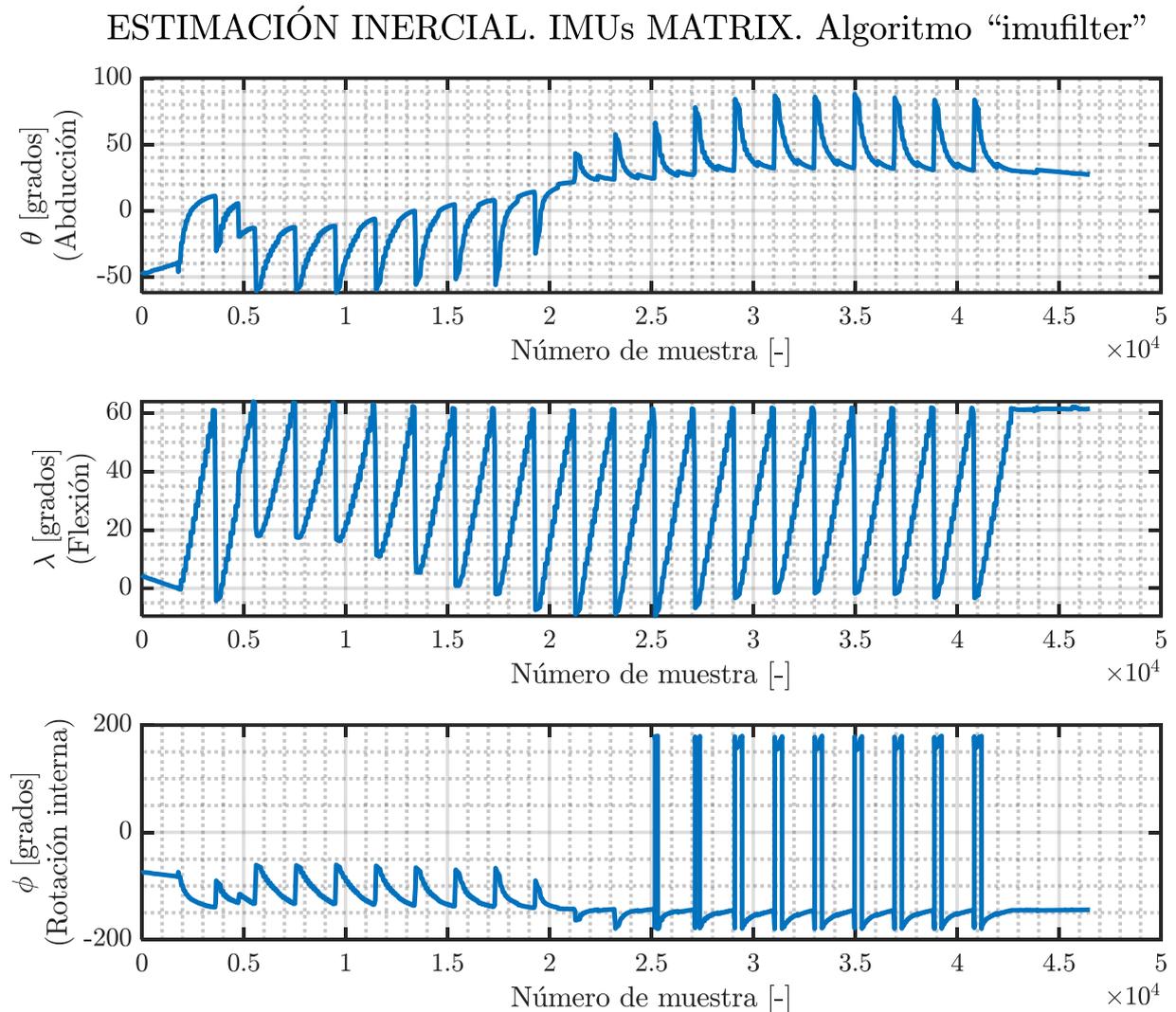


Figura 5.17.- Estimación inercial del movimiento articular realizado con el *cobot UR3e*, la cual se ha expresado mediante sus tres *Ángulos de Euler*. *NOTA: Se ha utilizado el algoritmo “imufilter” para calcular la orientación de cada IMU.*

Efectuando un *zoom* para los casos en los que $\delta_{DESALINEACIÓN} \in \{-20^\circ, 0^\circ, 20^\circ\}$, se ha generado la siguiente representación gráfica:

ESTIMACIÓN INERCIAL (VISTA DE DETALLE).
IMUs MATRIX. Algoritmo: “imufilter”

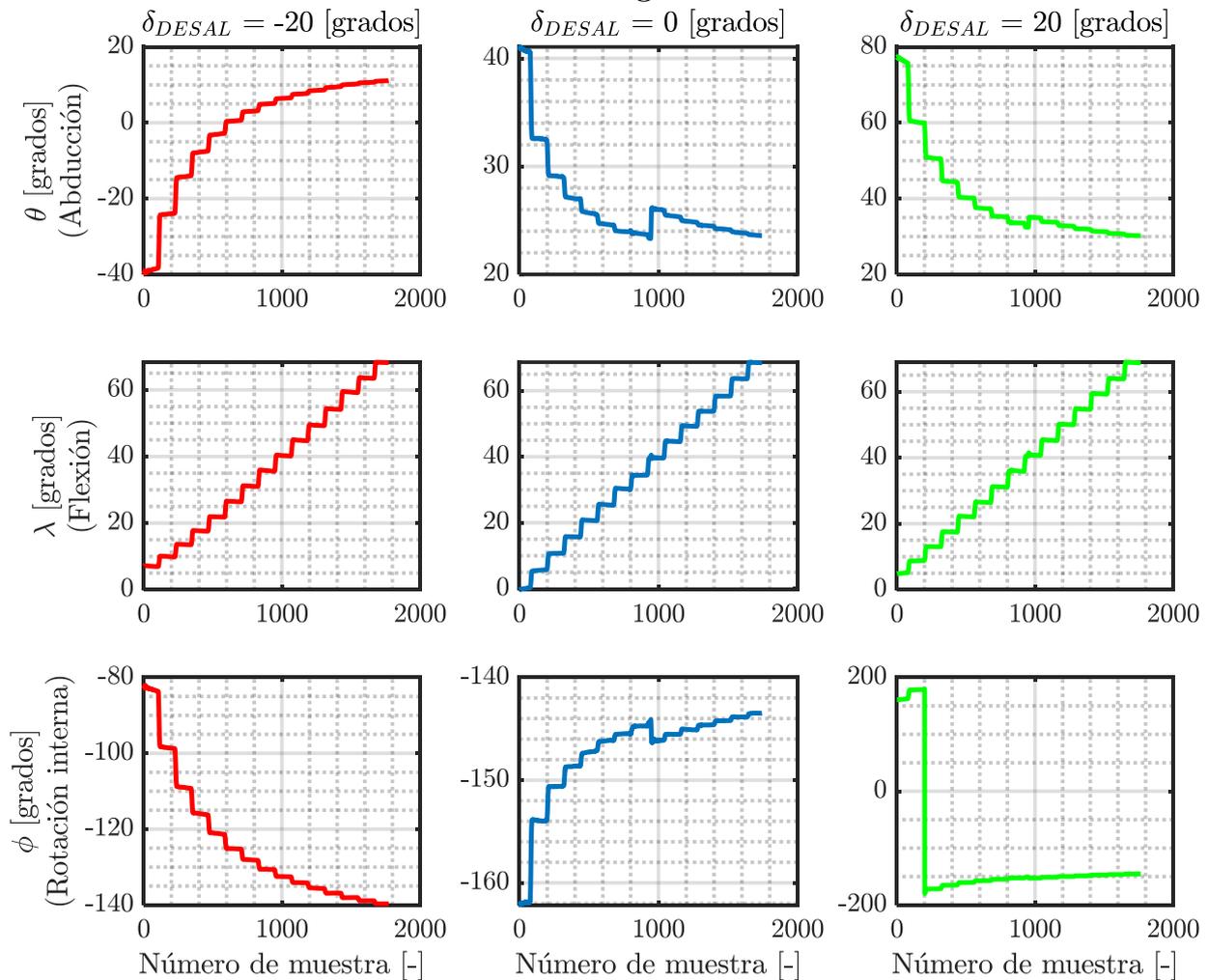


Figura 5.18.- Vista ampliada de la estimación inercial efectuada para los siguientes ángulos de desalineación: $\{-20^\circ, 0^\circ, 20^\circ\}$. Pueden observarse las 3 componentes del movimiento articular: abducción (θ°), flexión (λ°) y rotación interna (ϕ°). *NOTA: Este resultado se corresponde con los IMUs MATRIX y el algoritmo imufilter, empleado durante la estimación de la orientación.*

De forma análoga a los casos anteriores, se han calculado los siguientes resultados numéricos:

Tabla 5.10.- Errores generados durante la estimación de la flexión en caso de no existir desalineación sensor – segmento (*EXP_4, MATRIX, imufilter, diferencia de cuaterniones*).

Parámetro estadístico	λ [grados] (Flexión)														
	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°
$\mu_{ \epsilon }$ [°]	0.11	0.59	0.71	0.77	0.75	0.47	0.33	0.61	0.33	0.33	0.70	1.23	1.62	1.33	1.47
$10 \cdot \sigma_{ \epsilon }$ [°]	0.84	0.41	0.11	0.11	0.30	0.39	0.35	0.09	0.35	0.31	0.21	0.09	0.04	0.18	0.15

$\mu_{|\epsilon|}$ [°] se corresponde con la media del error cometido durante la estimación inercial, expresada en grados.
 $\sigma_{|\epsilon|}$ [°] denota la desviación típica del error presente en dicha estimación inercial, expresada en grados.

Los *IMUs MATRIX*, en conjunto con los algoritmos "imufilter" y "diferencia de cuaterniones", generan una buena estimación de los ángulos de flexión cuando no existe desalineación sensor - segmento. En dicho escenario, se verifica que el error máximo en la estimación de la flexión asciende a 1.62° ($\lambda = 60^\circ$). Analizando las desviaciones generadas en la señal de error, puede observarse que se tiene una variabilidad muy poco significativa.

5.4.- ERROR TEÓRICO GENERADO DURANTE LA ESTIMACIÓN INERCIAL MEDIANTE DATOS SINTÉTICOS

Las ecuaciones que permiten estimar inercialmente la rotación articular (${}^B q_{AF_{Est1}}$), ya detalladas en anteriores secciones de este documento, pueden utilizarse para analizar la tendencia que ofrece el error teórico de estimación. La base datos utilizada para ejecutar el algoritmo implementado ha sido elaborada de forma sintética, esto es, sin la captura de datos experimentales.

5.4.1.- Simulación de los errores de alineación sensor – segmento anatómico

Como primer paso para estudiar el error teórico de estimación inercial, se han definido varios ángulos de desalineación inicial. Dichos valores de desalineación estarán presentes en los cuaterniones de orientación ligados a los sensores inerciales; colocados, respectivamente, en el brazo (${}^{GS} q_A$) y el antebrazo (${}^{GS} q_F$). Esta disposición geométrica

también se puede generalizar para cualquier binomio de sólidos rígidos, cuya diferencia de orientación espacial se desee calcular. Con todo ello, se han establecido las siguientes desalineaciones:

$$\delta_{DESALINEACIÓN} \in \left\{ \begin{array}{l} -20^{\circ}, -18^{\circ}, -16^{\circ}, -14^{\circ}, -12^{\circ}, -10^{\circ}, -8^{\circ}, \\ -6^{\circ}, -4^{\circ}, -2^{\circ}, 0^{\circ}, 2^{\circ}, 4^{\circ}, 6^{\circ}, 8^{\circ}, 10^{\circ}, 12^{\circ}, 14^{\circ}, \\ 16^{\circ}, 18^{\circ}, 20^{\circ} \end{array} \right\} \quad (5.11)$$

Los ángulos de desalineación registrados en la expresión (5.11) pueden introducirse en cualquiera de los tres *parámetros de Euler* (ϕ_{ERROR} , ϑ_{ERROR} , ψ_{ERROR}) que conforman el cuaternión de error. Inicialmente, se ha considerado que la desalineación ha sido incorporada en el *eje X* del sensor correspondiente; de modo que $\phi_{ERROR} = \delta_{DESALINEACIÓN}$. Téngase en cuenta que dicho error de alineación solamente se ha incorporado para el par *sensor - segmento* del brazo ($\phi_{ERROR}^{ARM} = \delta_{DESALINEACIÓN}$).

5.4.2.- Introducción de las referencias de rotación articular

Por otra parte, el algoritmo precisa conocer las referencias de rotación articular comandadas; las cuales se han identificado, en este caso, mediante la notación λ [grados]. A continuación, se indican los valores definidos para el parámetro λ :

$$\lambda \in \{0^{\circ}, 5^{\circ}, 10^{\circ}, 15^{\circ}, 20^{\circ}, 25^{\circ}, 30^{\circ}, 35^{\circ}, 40^{\circ}, 45^{\circ}, 50^{\circ}, 55^{\circ}, 60^{\circ}, 65^{\circ}, 70^{\circ}\} \quad (5.12)$$

5.4.3.- Curva teórica de error obtenida

En la siguiente figura¹⁷ se muestra la evolución teórica del error cometido durante la estimación inercial, en función de la desalineación simulada.

¹⁷ **Nota:** Cabe destacar que durante la construcción de la Figura 5.19 se ha tomado una muestra de los datos totales simulados para la referencia de rotación articular λ :

$$\delta_{DESALINEACIÓN} \in \{-20^{\circ}, -18^{\circ}, -16^{\circ}, \dots, 0^{\circ}, \dots, 16^{\circ}, 18^{\circ}, 20^{\circ}\}; \quad \lambda = 70^{\circ}.$$

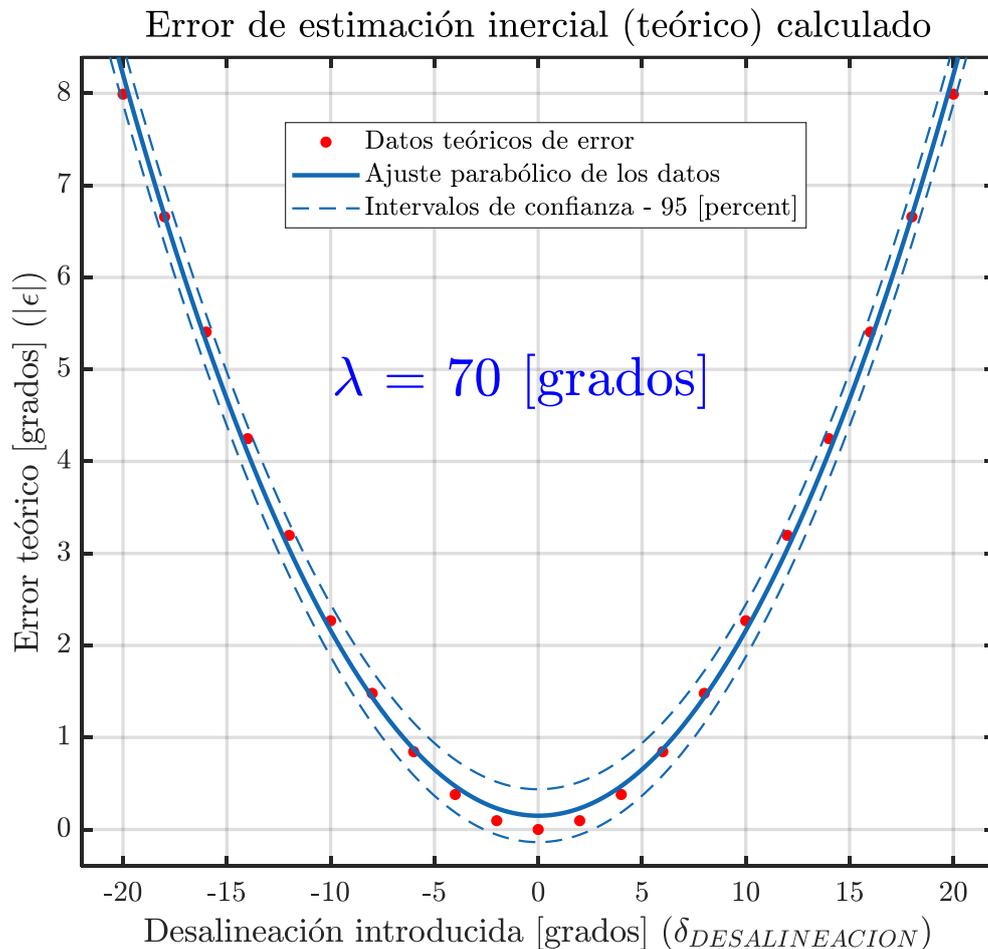


Figura 5.19.- Representación gráfica del error de estimación (teórico) en función del ángulo de desalineación incorporado. **Nota:** $\lambda = 70^\circ$.

Durante el ajuste estadístico de los datos teóricos de error (puntos de color rojo en la Figura 5.19), se ha empleado una expansión polinómica de orden 2 (ecuación parabólica). Por ende, dicha curva de ajuste (β_{AJUSTE}) presenta la siguiente estructura:

$$\beta_{AJUSTE} = ax^2 + bx + c \quad (5.13)$$

Donde:

- La variable explicada (β_{AJUSTE}) denota al error teórico, en valor absoluto, generado por el algoritmo de estimación ($|\varepsilon|$).

- La variable explicativa (x) se corresponde con el error de alineación ($\phi_{ERROR}^{ARM} = \delta_{DESALINEACIÓN}$) emulado en el par *sensor - segmento* del brazo.

A continuación, se recogen los principales resultados numéricos del ajuste polinómico comentado anteriormente:

Tabla 5.11.- Resultados numéricos obtenidos durante el tratamiento estadístico de los datos teóricos, correspondientes a los errores de estimación inercial para $\lambda = 10^\circ$.

<i>a</i>	<i>b</i>	<i>c</i>	<i>R-square</i>	<i>Adjusted R-square</i>	<i>RMSE</i>
0,02013	2,247e-19	0,1497	0,9979	0,9977	0,13
<i>Intervalo de confianza del 95 [%]</i>					
(0,01967; 0,02058)	(-0,00492; 0,00492)	(0,06021; 0,2393)			

El estudio detallado anteriormente se ha repetido para el caso de $\lambda = 60^\circ$ (**referencia de rotación articular**), obteniéndose los siguientes resultados:

Error de estimación inercial (teórico) calculado

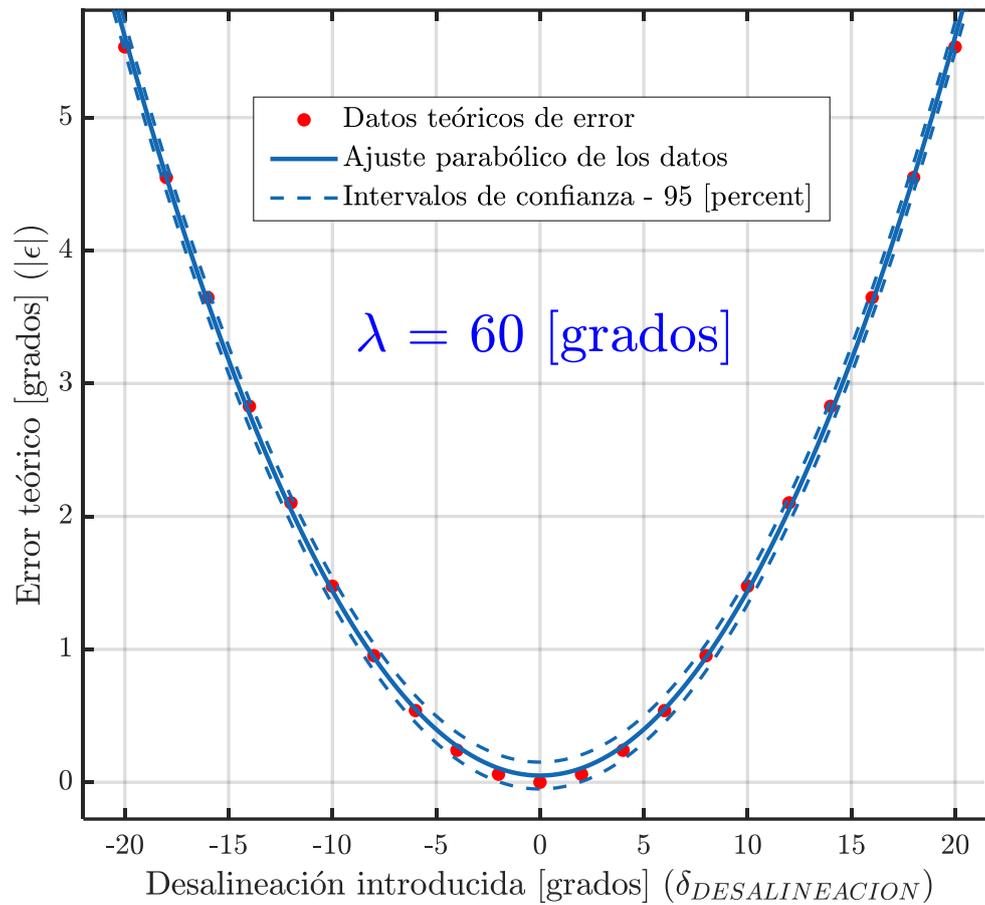


Figura 5.20.- Representación gráfica del error de estimación (teórico) en función del ángulo de desalineación incorporado. *Nota:* $\lambda = 60^\circ$.

Tabla 5.12.- Resultados numéricos obtenidos durante el tratamiento estadístico de los datos teóricos, correspondientes a los errores de estimación inercial para $\lambda = 60^\circ$.

<i>a</i>	<i>b</i>	<i>c</i>	<i>R-square</i>	<i>Adjusted R-square</i>	<i>RMSE</i>
0,01389	4,077e-18	0,05043	0,9995	0,9994	0,04576
<i>Intervalo de confianza del 95 [%]</i>					
(0,01373; 0,01405)	(-0,001732; 0,001732)	(0,0189; 0,08196)			

A partir de los resultados obtenidos, puede deducirse que el error de estimación inercial depende de la referencia de rotación articular y la desalineación sensor - segmento inicialmente presente:

$$|\epsilon| = f(\lambda, \delta_{DESALINEACIÓN}) \quad (5.14)$$

De acuerdo con lo indicado en la expresión (5.14), la tendencia que adopta el error teórico de estimación puede aproximarse mediante una superficie; tal y como se muestra en la siguiente sección.

5.4.4.- Superficie teórica de error generada

En la Figura 5.21 se recoge el alzado y la vista isométrica asociados a la superficie teórica de error ($|\epsilon|$). Durante la interpolación de toda la nube de puntos, se ha utilizado un algoritmo de ajuste lineal¹⁸.

¹⁸ **Nota:** Durante el método de ajuste lineal, el parámetro λ ha sido normalizado a través de la media $\mu_\lambda = 35$ [°] y la desviación típica $\sigma_\lambda = 21,64$ [°]. Por otra parte, la normalización de la variable $\delta_{DESALINEACIÓN}$ se ha llevado a cabo mediante los siguientes parámetros estadísticos:

$$\begin{aligned} \mu_{\delta_{DESALINEACIÓN}} &= 0 \text{ [°]} \\ \sigma_{\delta_{DESALINEACIÓN}} &= 12,13 \text{ [°]} \end{aligned}$$

SUPERFICIE TEÓRICA DE ERROR. $\lambda \in [0^\circ, 70^\circ]$. $\delta_{DESALINEACION} \equiv \phi_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$

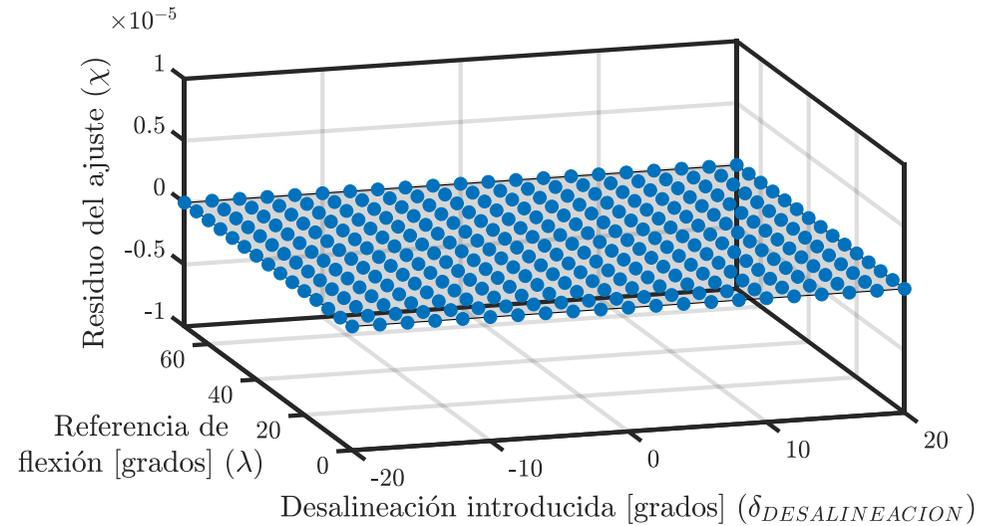
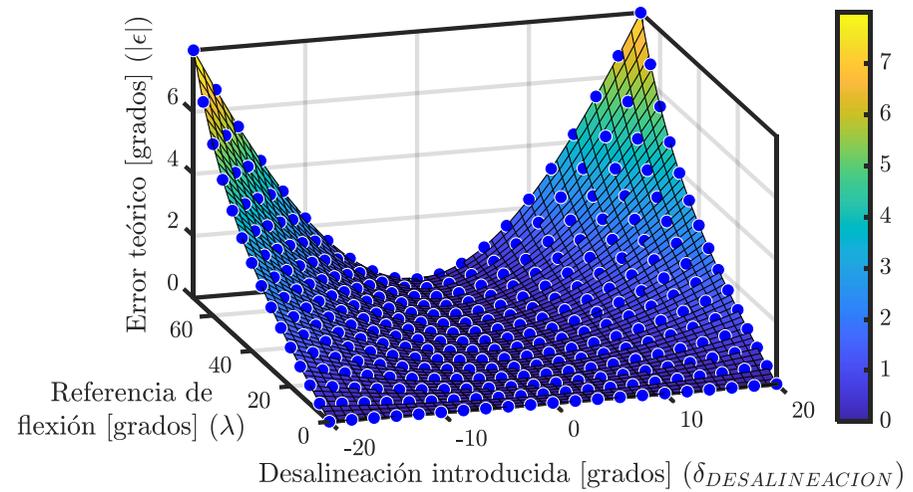
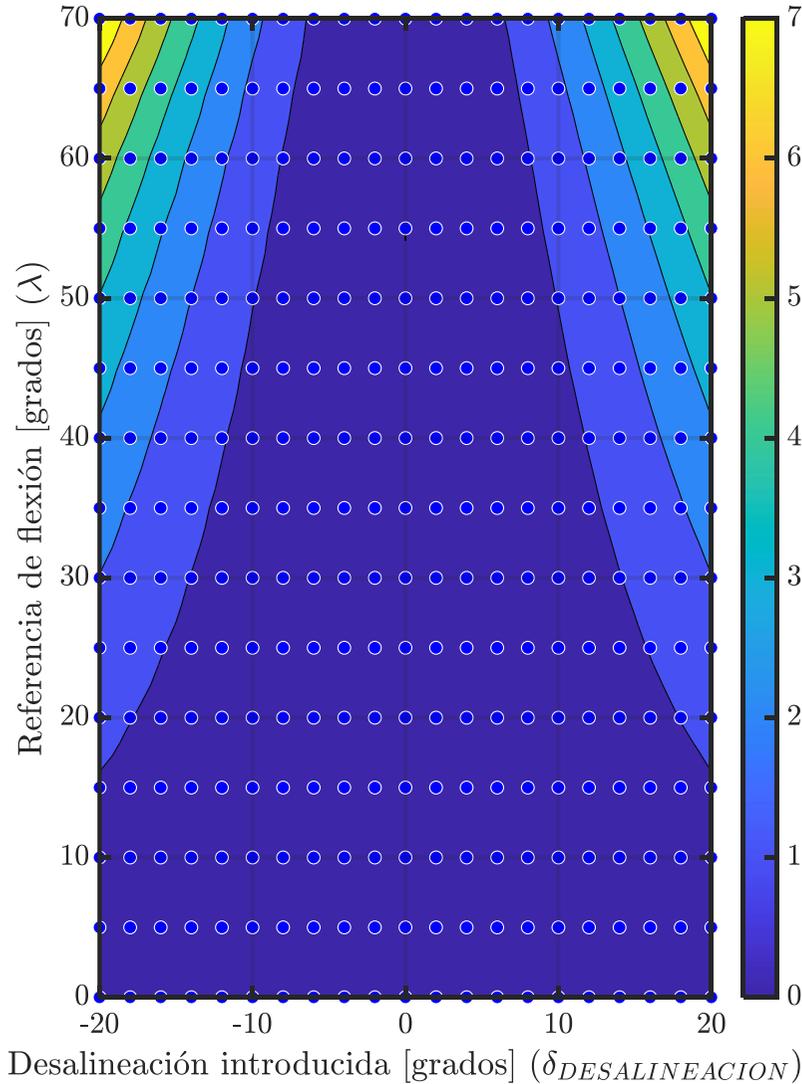


Figura 5.21.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. **Nota:** $\delta_{DESALINEACION} \equiv \phi_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$.

De acuerdo con los resultados recogidos en la figura anterior, se verifica que **la desalineación geométrica** introducida en el par sensor - segmento ($\delta_{DESALINEACIÓN}$) posee un **efecto parabólico** sobre el error final de la estimación inercial ($|\epsilon|$):

$$|\epsilon| = k \cdot \delta_{DESALINEACIÓN}^2, \quad k \in \mathbb{R} \quad (5.15)$$

En la otra cara de la moneda, puede comprobarse que **el valor de la flexión realizada** (λ) tiene una **relación** aproximadamente **lineal con el error** de estimación inercial ($|\epsilon|$):

$$|\epsilon| = k \cdot \lambda, \quad k \in \mathbb{R} \quad (5.16)$$

5.4.5.- Introducción de los errores de alineación en los dos ejes cartesianos restantes

Durante la sección anterior, recuérdese que la desalineación geométrica inicial ha sido incorporada en el *eje X*. En el presente apartado, se analizará qué efecto produce la desalineación definida en los ejes cartesianos *Z* e *Y* del sensor.

❖ *Desalineación existente a lo largo del eje Y del IMU*

En este estudio cabe destacar que el *eje Y* del sensor es coincidente con el eje motriz simulado. Por ende, el error introducido se traducirá directamente como un *offset* en la estimación inercial final. Este hecho puede visualizarse gráficamente a través de la siguiente figura:

SUPERFICIE TEÓRICA DE ERROR. $\lambda \in [0^\circ, 70^\circ]$. $\delta_{DESALINEACION} \equiv \vartheta_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$

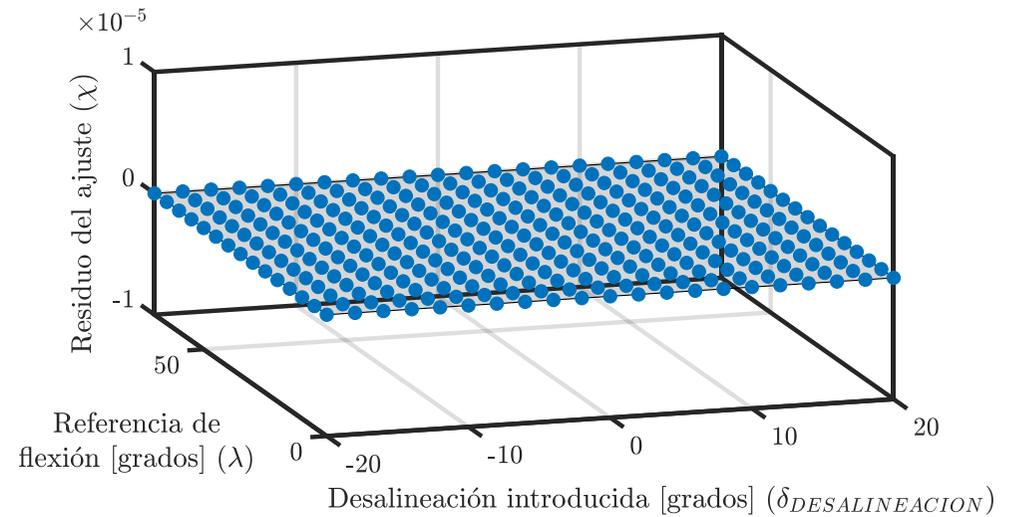
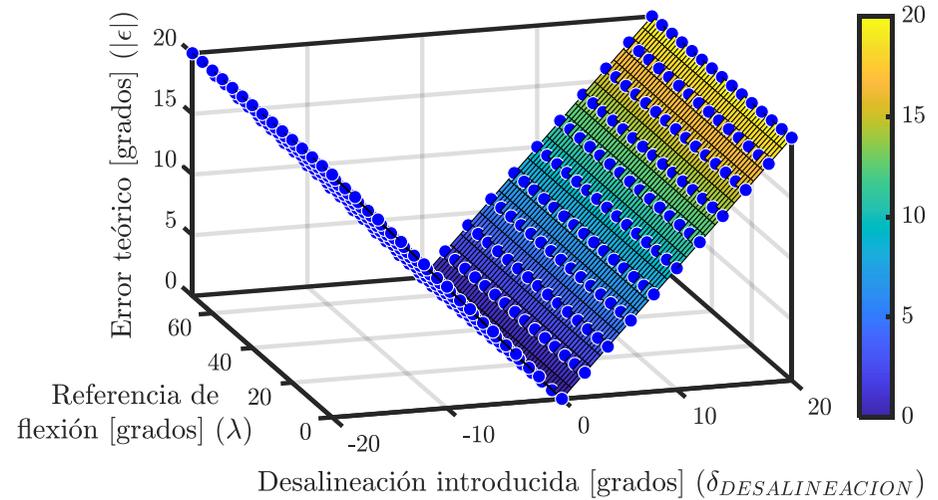
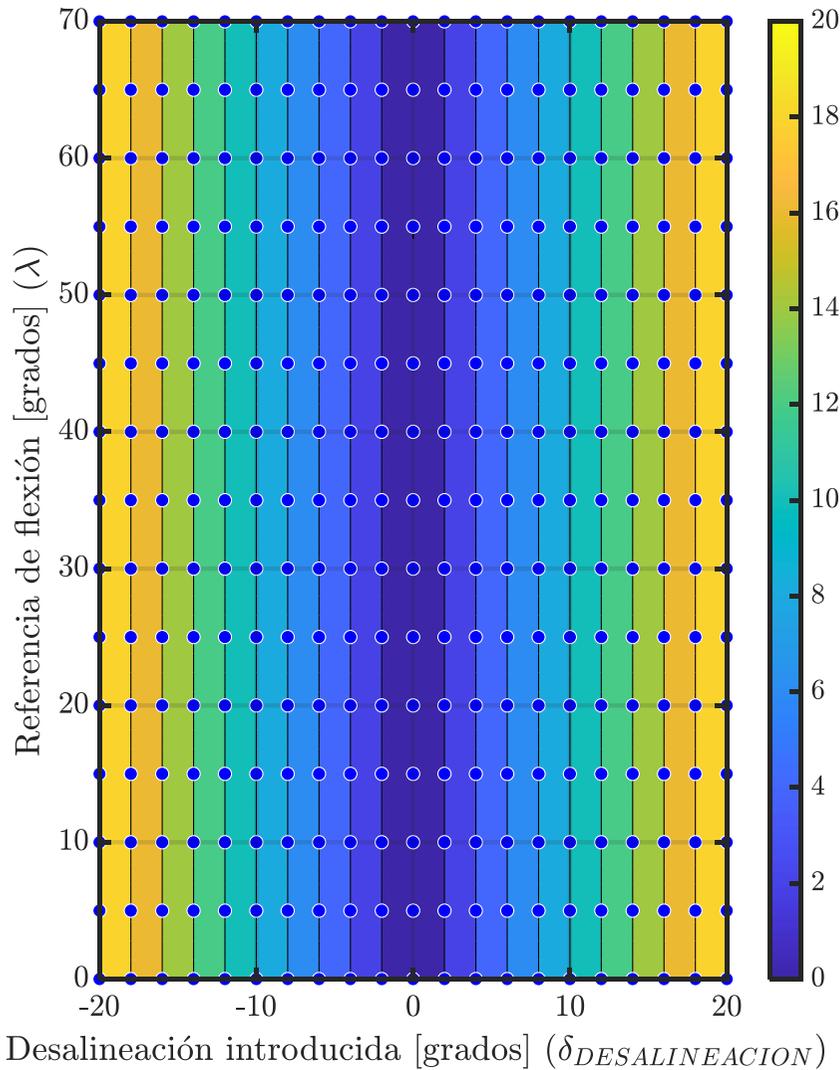


Figura 5.22.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. **Nota:** $\delta_{DESALINEACION} \equiv \vartheta_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$.

En conformidad con los resultados mostrados en la Figura 5.22, puede deducirse que **el ángulo de flexión (λ) carece de influencia sobre el error final de la estimación inercial**. Sin embargo, **la desalineación existente en el binomio *sensor – segmento* se traslada directamente sobre dicho error de estimación**. Matemáticamente, el error final del algoritmo ($|\epsilon|$) toma el valor absoluto ligado al ángulo de desalineación ($\delta_{DESALINEACIÓN}$):

$$|\epsilon| = |\delta_{DESALINEACIÓN}| \quad (5.17)$$

❖ *Desalineación existente a lo largo del eje Z del IMU*

Análogamente al caso de estudio vinculado al *eje Y* del sensor inercial, se ha realizado un análisis sobre la desalineación introducida en el *eje Z* de dicho dispositivo. Los resultados obtenidos, representados en la Figura 5.23, constatan que el error ($|\epsilon|$) generado durante la estimación inercial es prácticamente nulo para cualquier valor de desalineación ($\delta_{DESALINEACIÓN}$); integrada respecto al *eje Z* del sensor.

Complementariamente a lo indicado durante el párrafo anterior, puede establecerse la siguiente relación de variables¹⁹:

$$|\epsilon| \neq f(\lambda, \delta_{DESALINEACIÓN}) \Leftrightarrow |\epsilon| = K_0 \cong 0 \quad (5.18)$$

¹⁹ **Nota:** En la notación desarrollada durante la ecuación (5.18) se ha definido la constante arbitraria K_0 .

SUPERFICIE TEÓRICA DE ERROR. $\lambda \in [0^\circ, 70^\circ]$. $\delta_{DESALINEACIÓN} \equiv \psi_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$

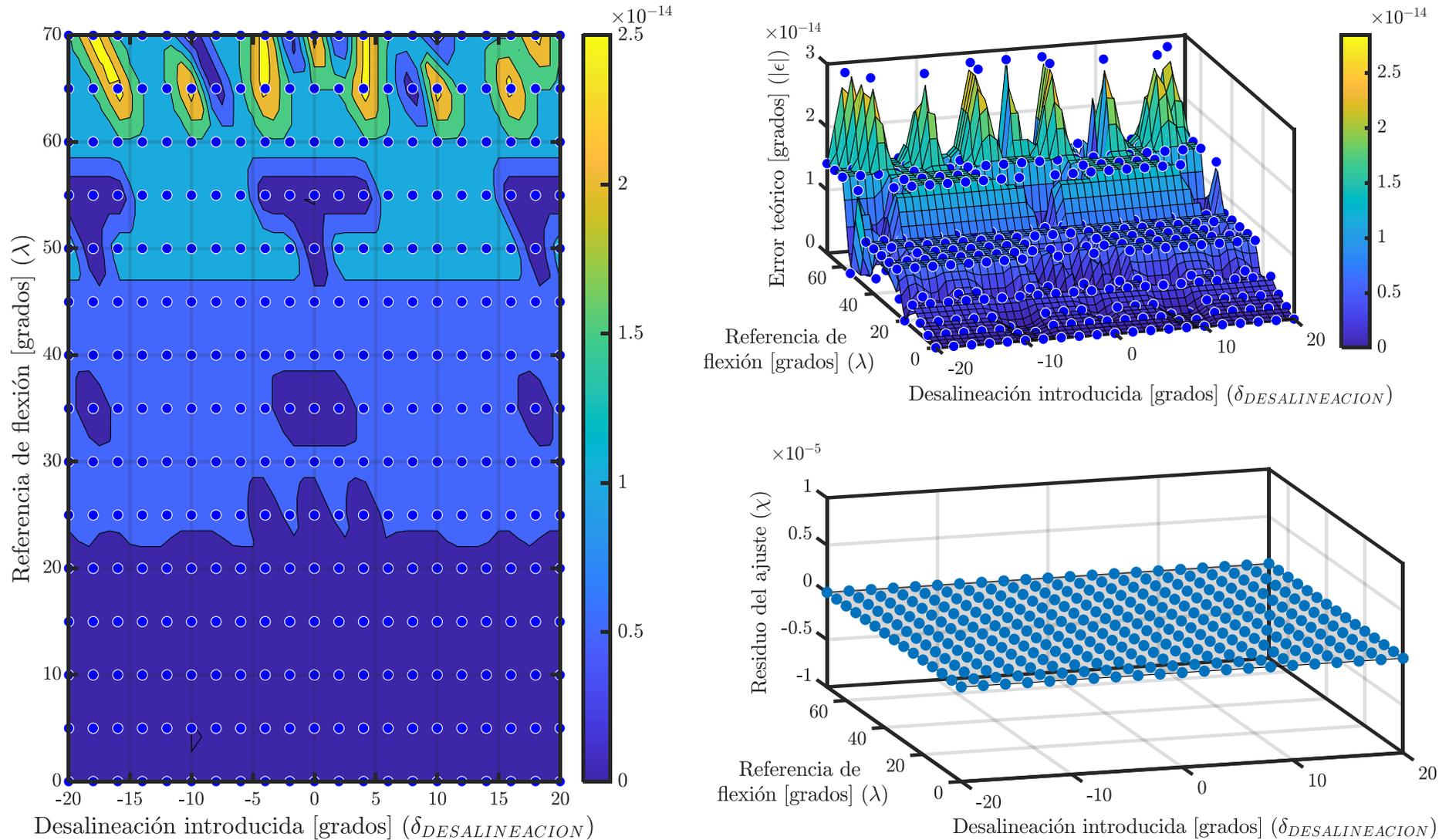


Figura 5.23.- Representación tridimensional del error teórico inherente al modelo matemático desarrollado. *Nota:* $\delta_{DESALINEACIÓN} \equiv \psi_{ERROR}^{ARM} \in \{-20^\circ, 20^\circ\}$.

6. Discusión

En este capítulo se procede a discutir la validez de los resultados expuestos durante el apartado anterior. Asimismo, se justificarán las posibles causas que apoyan los datos empíricos obtenidos.

En primer lugar, puede deducirse que los datos inerciales obtenidos mediante los dispositivos *Xsens DOT* ofrecen menos variabilidad que los capturados por los *IMUs* pertenecientes a la gama *MATRIX*; de acuerdo con las desviaciones típicas previamente calculadas. Esta afirmación es aplicable tanto para el acelerómetro como para el giróscopo de los *IMUs*. Recuérdese que los sensores *MATRIX* no poseen magnetómetro.

Los protocolos de calibración *2D* y *3D* no se han traducido en una modificación de las desviaciones típicas vinculadas a los datos inerciales, puesto que dichos parámetros estadísticos son una característica intrínseca a cada *IMU*.

Para los ángulos de flexión considerados, el algoritmo *imufilter* genera un error de estimación 10 veces mayor que en la situación de utilizar los cuaterniones suministrados por el filtro propietario de los *IMUs Xsens DOT*. El algoritmo de orientación "ecompass" (*IMUs Xsens DOT*) y de evaluación inercial "diferencia de cuaterniones" generan resultados con una mayor presencia de ruido. Este hecho puede deberse a la variabilidad del campo magnético existente en las cercanías del robot *UR3e*.

De acuerdo con lo indicado en el párrafo anterior, los resultados experimentales avalan que el algoritmo *ecompass* ofrece mejores estimaciones en la evaluación inercial frente al método *imufilter*; en el caso de *IMUs Xsens DOT*.

Por otro lado, puede concluirse que el algoritmo *TRIAD* ofrece mejores estimaciones que el método constituido por los siguientes ítems: "ecompass" + Acelerómetro (*Xsens DOT*) + Magnetómetro (*Xsens DOT*).

Finalmente, se ha comprobado que el filtro propietario de los *IMUs Xsens DOT* (acoplado con el algoritmo "Diferencia de cuaterniones") ofrece la mejor estimación inercial de los ángulos de flexión; dentro de las posibilidades contempladas en el presente estudio. No obstante, téngase en cuenta que se ha considerado la simplificación de que los dos sistemas de referencia de cada sensor son coincidentes (hipótesis ya indicada en la sección "2.1.3"). Experimentalmente, se puede comprobar que esta hipótesis es discutible y los dos *IMUs* pueden estar referenciados a *frames* ligeramente diferentes.

El primer paso necesario para estimar inercialmente una rotación articular consiste en calcular la orientación de cada uno de los dos sensores inerciales utilizados. En el caso de los dispositivos *Xsens DOT*, el microcontrolador presente en dichos *IMUs* suministra la orientación en forma de *Ángulos de Euler* (orden de conversión "ZYX") o cuaterniones. El problema de esta información es que el sistema de referencia respecto al cual están codificadas todas las medidas de orientación es desconocido y varía con cada experimento de laboratorio. Por ejemplo, si se desea inicializar a "0" todos los *Ángulos de Euler* de varios *IMUs Xsens DOT*, se ha comprobado experimentalmente que no existe una secuencia de movimientos única para los diversos sensores inerciales considerados.

En la siguiente figura puede observarse la orientación de varios *Xsens DOT* que garantiza un valor nulo para los correspondientes *Ángulos de Euler*. El experimento se ha realizado dos veces, corroborando que el sistema de referencia de inicial es diferente para dispositivo de medida inercial y se modifica cada vez que se reinicia el respectivo *IMU*.



Figura 6.1.- Fotografía de los *IMUs Xsens DOT* durante los dos experimentos iniciales realizados. [Elaboración propia].

Este problema de la orientación inicial de cada *IMU* también se manifiesta al aplicar un filtro sobre los datos crudos suministrados por el sensor. Este es el caso de los algoritmos *imufilter* y *ecompass* implementados en *MatLab* [42].

7. Conclusiones

En consonancia con el proyecto desarrollado, resulta correcto afirmar que la estimación inercial de movimientos articulares presenta una elevada complejidad.

Los resultados generados no dependen solamente del algoritmo de evaluación inercial seleccionado, sino que también tiene influencia la familia de sensores considerada. En función de los datos crudos necesarios por cada algoritmo, las condiciones ambientales de los experimentos pueden influir notoriamente en los resultados generados.

El estudio desarrollado en este proyecto tiene aplicaciones en los sectores de la biomecánica y la robótica, permitiendo realizar la monitorización ambulatoria del movimiento humano o incrementar la fiabilidad en las tareas de colaboración *humano - robot*.

Las estrategias computacionales diseñadas en este trabajo pueden refinarse en futuras líneas de investigación, así como incorporar nuevos algoritmos para la evaluación de las unidades de medida inercial.

Finalmente, téngase presente que la metodología expuesta en este trabajo se ha particularizado para el caso de los ángulos formados entre dos eslabones de un brazo robótico. No obstante, dicho desarrollo teórico-experimental es extrapolable a cualquier situación biomecánica en la que sea necesario conocer analíticamente el ángulo formado por dos sólidos rígidos adyacentes de una cadena cinemática genérica.

8. Planificación temporal

El presente apartado se ha destinado a reflejar la línea temporal establecida para desarrollar este *Trabajo Fin de Máster (TFM)*.

A través de la siguiente tabla, se exponen todas las tareas realizadas durante el transcurso del *TFM*; así como los períodos temporales programados para cada de hito del proyecto. Posteriormente, dicha relación de actividades se traducirá de forma gráfica en un *Diagrama de Gantt*.

Tabla 8.1.- Desglose de las tareas que se han definido para desarrollar el presente *Trabajo Fin de Máster*.

	Tarea	Comienzo planificado	Fin planificado	Comienzo real	Fin real	Variación de duración	Precedencias
1.-	Estudios previos, análisis preliminares.						
1.1.-	Diseño inicial de las fases que conforman el proyecto (<i>TFM</i>).	mié 01/02/23	mié 08/02/23	mié 01/02/23	mié 08/02/23	0 días	
1.2.-	Búsqueda y lectura de <i>papers</i> relacionados con la evaluación biomecánica de <i>IMUs</i> .	mié 08/02/23	mié 15/02/23	mié 08/02/23	vie 17/02/23	2 días	Finalizada la tarea 1.1.
1.3.-	Exploración de la documentación asociada al <i>cobot UR3e</i> .	mié 15/02/23	mar 21/02/23	vie 17/02/23	lun 20/02/23	- 3 días	Finalizada la tarea 1.2.
1.4.-	Aprendizaje sobre el funcionamiento vinculado a las siguientes familias de sensores inerciales: <i>Xsens DOT, MATRIX, SuperSens</i> .	mar 21/02/23	lun 27/02/23	lun 20/02/23	lun 27/02/23	1 día	Finalizada la tarea 1.3.

1.5.-	Definición inicial de objetivos a satisfacer durante el trabajo.	lun 27/02/23	vie 03/03/23	lun 27/02/23	vie 03/03/23	0 días	Comenzada la tarea 1.2.
2.-	Análisis exhaustivo del problema que se debe resolver.						
2.1.-	Diseño de la metodología experimental asociada al <i>TFM</i> .	vie 03/03/23	vie 10/03/23	vie 03/03/23	vie 10/03/23	0 días	Finalizada la tarea 1.
2.2.-	Estudio sobre la viabilidad y la validez del experimento.	vie 10/03/23	mar 14/03/23	vie 10/03/23	mar 14/03/23	0 días	Finalizada la tarea 2.1.
3.-	Descripción matemática de los algoritmos diseñados.						
3.1.-	Estimación del ángulo de flexión del codo empleando cuaterniones de orientación.	mar 14/03/23	lun 20/03/23	mar 14/03/23	lun 20/03/23	0 días	Finalizada la tarea 2.
3.2.-	Cálculo de la orientación mediante el <i>Filtro de Kalman</i> . Dicha técnica se aplica a los datos de aceleración, velocidad angular de giro y campo magnético.	lun 20/03/23	mié 22/03/23	lun 20/03/23	jue 23/03/23	1 día	Finalizada la tarea 3.1.
3.3.-	Estimación de la orientación a través del método <i>TRIAD</i> . Este algoritmo hace uso de datos de aceleración e intensidad de campo magnético.	mié 22/03/23	lun 27/03/23	jue 23/03/23	lun 27/03/23	- 1 día	Finalizada la tarea 3.2.
3.4.-	Cálculo de la orientación mediante un <i>Filtro inicial</i> aplicado sobre datos de	lun 27/03/23	jue 30/03/23	lun 27/03/23	jue 30/03/23	0 días	Finalizada la tarea 3.3.

	aceleración y velocidad angular de giro: Filtro + Acelerómetro + Giroscopio.						
4.-	Programación en <i>MatLab</i> de los algoritmos propuestos.	jue 30/03/23	vie 05/05/23	jue 30/03/23	jue 04/05/23	- 1 día	Finalizada la tarea 3.
5.-	Ejecución de la etapa de <i>test</i> mediante datos sintéticos (manuales).						
5.1.-	Depuración a través del brazo robótico <i>UR3e</i> .	vie 05/05/23	lun 15/05/23	jue 04/05/23	lun 15/05/23	1 día	Finalizada la tarea 4.
5.2.-	Evaluación de los errores intrínsecos a los algoritmos implementados. Diseño de un <i>test de desalineación</i> .	lun 15/05/23	lun 22/05/23	lun 15/05/23	lun 22/05/23	0 días	Finalizada la tarea 5.1.
6.-	Desarrollo experimental con personas. Realización de pruebas biomecánicas, en sujetos humanos, para los algoritmos implementados.	lun 22/05/23	lun 05/06/23	--	--	--	Finalizada la tarea 5.
7.-	Desarrollo, maquetación y revisión de los documentos asociados al presente Trabajo Fin de Máster.	mié 01/02/23	lun 12/06/23	mié 01/02/23	lun 12/06/23	0 días	Comenzada la tarea 1.1.
8.-	Construcción de la presentación que se deriva del trabajo desarrollado durante el TFM.	mié 01/02/23	lun 12/06/23	mié 01/02/23	lun 12/06/23	0 días	Comenzada la tarea 1.1.
9.-	Lectura y Defensa del TFM.	A establecer por la <i>EPIG</i>	A establecer por la <i>EPIG</i>	--	--	--	Finalizadas las tareas 7 y 8.

En consonancia con la información recogida mediante la Tabla 8.1, puede observarse que este *Trabajo Fin de Máster* posee una **duración temporal** aproximada de **6 [meses]**; aproximadamente. La **fecha de inicio** del proyecto se ha fijado como el **miércoles 01/02/23**, mientras su correspondiente **día de finalización** ha sido el **lunes 12/07/23**. Este marco temporal se traduce en un total de **114 días laborables**.

El trabajo asociado al presente proyecto se ha desglosado en nueve ítems principales. Con objeto de dosificar la carga semanal de actividades a realizar, se han diseñado **22 subtareas en total**; las cuales ya se han enunciado en la Tabla 8.1.

La programación temporal, trazada con anterioridad, se ha verificado en la mayoría de las tareas definidas para el proyecto. No obstante, cabe reseñar que en algunas actividades sí se han producido ligeras desviaciones temporales; respecto a la planificación inicialmente prevista. A continuación, se justifican brevemente las variaciones temporales en los hitos del proyecto indicados con anterioridad:

- ❖ La exploración y revisión de artículos sobre la evaluación biomecánica de unidades de medida inercial (*tarea 1.2*) se ha prolongado dos días adicionales. En esta tarea, se ha considerado interesante dedicar el tiempo necesario a analizar el estado del arte del problema; prestando especial atención en las metodologías experimentales diseñadas durante la evaluación de *IMUs*.
- ❖ Seguidamente, a través del *hito 1.3*, se ha revisado la documentación técnica disponible sobre el robot colaborativo *UR3e*. Puesto que la búsqueda y selección de dichos archivos resulta menos laboriosa que en el caso anterior, la duración de la presente tarea se ha visto reducida tres días.
- ❖ La diligencia empleada en la *actividad 1.2* ha permitido prolongar un día la duración de la *tarea 1.4*. De este modo, se ha analizado profundamente el funcionamiento asociado a las unidades de medida inercial contempladas en este proyecto (*Xsens DOT* y *MATRIX*).

- ❖ Por otra parte, durante las *tareas* 3.2 y 3.3 se ha producido una variación temporal de un día en cada una de ellas. Estas pequeñas modificaciones de la planificación se han visto compensadas entre las dos actividades. Por ende, este hecho no ha afectado perjudicialmente a la duración global del proyecto.
- ❖ El *hito* 4, vinculado a la programación en *MatLab* de los algoritmos diseñados, ha finalizado un día antes de lo inicialmente previsto; debido a la necesidad de revisar y depurar todos los códigos implementados.
- ❖ Complementariamente al punto anterior, el *test* de los algoritmos de estimación de la orientación (*actividad* 5.1) se ha retrasado un día laborable. La duración global de todo el proyecto se ha prolongado, aproximadamente, 1 mes adicional; con objeto de refinar el trabajo previamente desarrollado.

8.1.- DIAGRAMA DE GANTT ASOCIADO AL PROYECTO

Con objeto de plasmar gráficamente la programación temporal enunciada en la Tabla 8.1, se ha hecho uso de la herramienta conocida como *Diagrama de Gantt* (véase la Figura 8.1). En dicha representación jerárquica del *Trabajo Fin de Máster*, cada una de las tareas que conforman la arquitectura del proyecto se corresponde con una barra horizontal; resaltada mediante color azul o rojo²⁰.

Por otro lado, la escala temporal asociada al proyecto puede observarse en la zona superior del *Diagrama de Gantt* construido.

²⁰ **Nota:** En la Gestión de Proyectos de Ingeniería resulta habitual utilizar el concepto de *Tarea Crítica*. Estas actividades son determinantes para la duración final del proyecto, puesto que cualquier demora en la finalización de dichas tareas se traducirá en un incremento de la fecha de finalización; planificada para el trabajo global. Por ello, se ha considerado interesante representar en el *Diagrama de Gantt* maquetado las tareas críticas existentes en el presente *TFM*.

Continuando con lo expresado durante el párrafo anterior, en el *Diagrama de Gantt* de la Figura 8.1 se ha etiquetado mediante color rojo las tareas críticas definidas para el proyecto.

Finalmente, cabe destacar que la definición de tareas y el cálculo de las actividades críticas del *TFM* se ha realizado mediante la herramienta *MS Project*; la cual ha sido facilitada por el *software corporativo* de la Universidad de Oviedo.

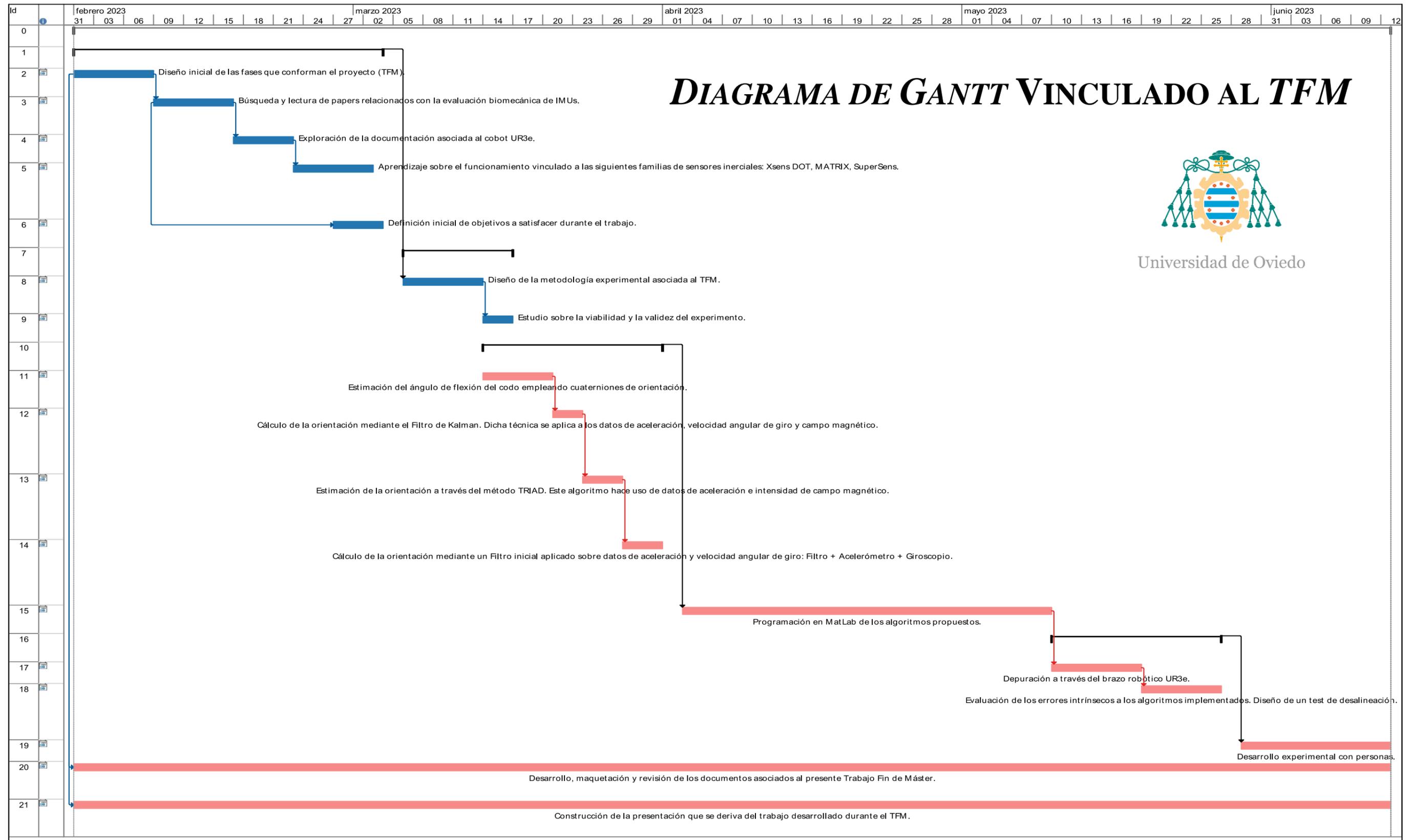


Figura 8.1.- Planificación temporal, en forma de *Diagrama de Gantt*, que se ha diseñado para el desarrollo de este *Trabajo Fin de Máster*. [Elaboración propia].

9. Líneas de trabajo futuras

El presente apartado se ha destinado a proponer varias actividades que servirán como continuación al proyecto desarrollado durante este *Trabajo Fin de Máster*.

Una interesante línea de investigación consiste en estudiar la estimación inercial de rotaciones articulares bajo un régimen dinámico. Esto es, se analizarán los errores generados por los diversos algoritmos propuestos durante el movimiento de los *IMUs*. Recuérdese que, en este proyecto, se han aplicado los métodos computacionales de estimación inercial bajo posiciones estáticas de los sensores.

En combinación con lo expresado mediante el párrafo anterior, se plantea maquetar dos *Diagramas de Bode*. El primero de ellos permitirá relacionar la precisión de los cálculos realizados y la frecuencia empleada por los *IMUs* durante la adquisición de datos. El segundo *Diagrama de Bode* hará posible observar el efecto que posee la frecuencia del movimiento programado sobre los errores de estimación inercial.

Se propone explorar nuevos algoritmos de evaluación inercial que utilicen el convenio eje - ángulo para expresar la orientación espacial, tal como el estudio referenciado en [6].

Otra posible ampliación del presente proyecto está consituida por incorporar la tecnología de captura del movimiento *OptiTrack* [43]. Se propone aplicar los algoritmos desarrollados en este *TFM* a una actividad biomecánica con personas. En dicho caso, se recurrirá al *software Motive* [44] para efectuar la adquisición de los datos espaciales; a partir de los marcadores colocados en los individuos monitorizados.

Finalmente, se propone realizar la evaluación de los algoritmos de estimación inercial con el sensor *bIMUr*; cuyo *hardware* y *firmware* ha sido desarrollado por el grupo de investigación *Sistemas Multisensor y Robótica* [39]. En este caso, puede estimarse la orientación espacial mediante datos de Aceleración e Intensidad de Campo Magnético.

9.1.- POTENCIALES SECTORES VINCULADOS A LA EVALUACIÓN METROLÓGICA DE SENSORES INERCIALES

En este apartado se detallan brevemente algunos campos de estudio en los que resulta interesante efectuar una evaluación inercial de sensores electrónicos:

- ❖ **Robótica colaborativa.** Las tareas que involucran la *interacción humano - robot* abren un gran abanico de posibilidades para las aplicaciones de los sensores inerciales. Existen numerosos estudios sobre la utilización de *IMUs* en este tipo de actividades, tales como los referenciados en [45], [46] y [47].
- ❖ **Sector de atención primaria.** En este campo de investigación existen numerosas propuestas para estimar inercialmente el ángulo de elevación del codo; como, por ejemplo, la literatura referenciada en [48] y [49]. Por otro lado, las variables cinemáticas del *gait* son ampliamente estudiadas [50].
- ❖ **Neurología.** El sector de la neurología se encuentra intrínsecamente ligado a las actividades rehabilitación física, ofreciendo novedosas soluciones clínicas mediante la evaluación inercial de *IMUs* [51].
- ❖ **Actividades deportivas.** Bajo este grupo se engloban la monitorización ambulatoria de la marcha [52], estudios sobre la evolución del centro de gravedad de una persona durante la actividad ciclista [53], cálculo de la fuerza de reacción contra el suelo durante el movimiento de una persona [54]; entre otros ejemplos.

10. Referencias Bibliográficas

[1] Página web oficial del *cobot UR3e*, la cual pertenece al fabricante *Universal Robots* y se encuentra disponible mediante el siguiente *link*: <https://www.universal-robots.com/es/productos/robot-ur3/>, *dirección web visitada el 11 de febrero de 2023*.

[2] Sitio web del programa de cálculo *MatLab R2022b*: https://es.mathworks.com/products/new_products/release2022b.html, *dirección web visitada el 28 de junio de 2023*.

[3] B. Fan, Q. Li, T. Tan, P. Kang and P. B. Shull, "Effects of IMU Sensor-to-Segment Misalignment and Orientation Error on 3-D Knee Joint Angle Estimation," in *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2543-2552, 1 Feb.1, 2022, doi: 10.1109/JSEN.2021.3137305.

[4] "Fundamentals of Spacecraft Attitude Determination and Control". F. Landis Markley, John L. Crassidis. *Space Technology Library*. DOI: <https://doi.org/10.1007/978-1-4939-0802-8>. Springer New York, NY. Hardcover ISBN: 978-1-4939-0801-1 (*Published: 01 June 2014*).

[5] Caruso, M.; Sabatini, A.M.; Laidig, D.; Seel, T.; Knaflitz, M.; Della Croce, U.; Cereatti, A. "Analysis of the Accuracy of Ten Algorithms for Orientation Estimation Using Inertial and Magnetic Sensing under Optimal Conditions: One Size Does Not Fit All". *Sensors* 2021, 21, 2543. <https://doi.org/10.3390/s21072543>.

[6] X. Xu, Y. Sun, X. Tian and L. Zhou, "A Novel Joint Angle Estimation Method for Serial Manipulator Using Micro-Electromechanical Systems Sensors," in *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10610-10620, Dec. 2020, doi: 10.1109/TIE.2019.2962442.

- [7] C. P. Walmsley, S. A. Williams, T. Grisbrook, C. Elliott, C. Imms, and A. Campbell, “Measurement of Upper Limb Range of Motion Using Wearable Sensors: A Systematic Review,” *Sports Med Open*, vol. 4, no. 1, 2018, doi: 10.1186/s40798-018-0167-7.
- [8] D. T. P. Fong and Y. Y. Chan, “The use of wearable inertial motion sensors in human lower limb biomechanics studies: A systematic review,” *Sensors (Switzerland)*, vol. 10, no. 12, pp. 11556–11565, Dec. 2010. doi: 10.3390/s101211556.
- [9] S. An, Y. Tuncel, T. Basaklar, G. K. Krishnakumar, G. Bhat, and U. Y. Ogras, “MGait : Model-Based Gait Analysis Using Wearable Bend and Inertial Sensors”, *ACM Transactions on Internet of Things*, vol. 3, no. 1, pp. 1–24, Feb. 2022, doi: 10.1145/3485434.
- [10] L. Carcreff, C. Moulin, B. Mariani, and S. Armand, “Simple rule to automatically recognize the orientation of the sagittal plane foot angular velocity for gait analysis using IMUs on the feet of individuals with heterogeneous motor disabilities,” *J Biomech*, vol. 135, Apr. 2022, doi: 10.1016/j.jbiomech.2022.111055.
- [11] D. Dong, C. Ma, M. Wang, H. T. Vu, B. Vanderborght, and Y. Sun, “A low-cost framework for the recognition of human motion gait phases and patterns based on multi-source perception fusion,” *Eng Appl Artif Intell*, vol. 120, Apr. 2023, doi: 10.1016/j.engappai.2023.105886.
- [12] J. Li et al., “Multi-body sensor data fusion to evaluate the hippotherapy for motor ability improvement in children with cerebral palsy,” *Information Fusion*, vol. 70, pp. 115–128, Jun. 2021, doi: 10.1016/j.inffus.2021.01.002.
- [13] Y. Sun, X. Xu, X. Tian, L. Zhou, and Y. Li, “A quaternion-based sensor fusion approach using orthogonal observations from 9D inertial and magnetic information”, *Information Fusion*, vol. 90, pp. 138–147, Feb. 2023, doi: 10.1016/j.inffus.2022.08.020.
- [14] S. Qiu et al., “Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges,”

Information Fusion, vol. 80. Elsevier B.V., pp. 241–265, Apr. 01, 2022. doi: 10.1016/j.inffus.2021.11.006.

[15] D. Laidig and T. Seel, “VQF: Highly accurate IMU orientation estimation with bias estimation and magnetic disturbance rejection,” *Information Fusion*, vol. 91, pp. 187–204, Mar. 2023, doi: 10.1016/j.inffus.2022.10.014.

[16] D. A. Aligia, B. A. Rocca, C. H. De Angelo, G. A. Magallán, and G. N. González, “An orientation estimation strategy for low cost IMU using a nonlinear Luenberger observer,” *Measurement (Lond)*, vol. 173, Mar. 2021, doi: 10.1016/j.measurement.2020.108664.

[17] A. Nez, L. Fradet, P. Laguillaumie, T. Monnet, and P. Lacouture, “Comparison of calibration methods for accelerometers used in human motion analysis,” *Medical Engineering and Physics*, vol. 38, no. 11. Elsevier Ltd, pp. 1289–1299, Nov. 01, 2016. doi: 10.1016/j.medengphy.2016.08.004.

[18] A. Nez, L. Fradet, P. Laguillaumie, T. Monnet, and P. Lacouture, “Comparison of calibration methods for accelerometers used in human motion analysis,” *Medical Engineering and Physics*, vol. 38, no. 11. Elsevier Ltd, pp. 1289–1299, Nov. 01, 2016. doi: 10.1016/j.medengphy.2016.08.004.

[19] M. A. Brodie, A. Walmsley, and W. Page, “The static accuracy and calibration of inertial measurement units for 3D orientation,” *Comput Methods Biomech Biomed Engin*, vol. 11, no. 6, pp. 641–648, 2008, doi: 10.1080/10255840802326736.

[20] S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, and A. Barraud, “Calibration methods for inertial and magnetic sensors,” *Sens Actuators A Phys*, vol. 156, no. 2, pp. 302–311, Dec. 2009, doi: 10.1016/j.sna.2009.10.008.

[21] D. Jurman, M. Jankovec, R. Kamnik, and M. Topič, “Calibration and data fusion solution for the miniature attitude and heading reference system,” *Sens Actuators A Phys*, vol. 138, no. 2, pp. 411–420, Aug. 2007, doi: 10.1016/j.sna.2007.05.008.

- [22] J. Favre, R. Aissaoui, B. M. Jolles, J. A. de Guise, and K. Aminian, “Functional calibration procedure for 3D knee joint angle description using inertial sensors,” *J Biomech*, vol. 42, no. 14, pp. 2330–2335, Oct. 2009, doi: 10.1016/j.jbiomech.2009.06.025.
- [23] G. Ligorio, D. Zanotto, A. M. Sabatini, and S. K. Agrawal, “A novel functional calibration method for real-time elbow joint angles estimation with magnetic-inertial sensors,” *J Biomech*, vol. 54, pp. 106–110, Mar. 2017, doi: 10.1016/j.jbiomech.2017.01.024.
- [24] H. J. Luinge, P. H. Veltink, and C. T. M. Baten, “Ambulatory measurement of arm orientation,” *J Biomech*, vol. 40, no. 1, pp. 78–85, 2007, doi: 10.1016/j.jbiomech.2005.11.011.
- [25] N. Choe, H. Zhao, S. Qiu, and Y. So, “A sensor-to-segment calibration method for motion capture system based on low cost MIMU,” *Measurement (Lond)*, vol. 131, pp. 490–500, Jan. 2019, doi: 10.1016/j.measurement.2018.07.078.
- [26] Nez, A., Fradet, L., Laguillaumie, P., Monnet, T., & Lacouture, P. (2016). “Comparison of calibration methods for accelerometers used in human motion analysis”. In *Medical Engineering and Physics* (Vol. 38, Issue 11, pp. 1289–1299). Elsevier Ltd. <https://doi.org/10.1016/j.medengphy.2016.08.004>.
- [27] Krohn, A., Beigl, M., Decker, C., & Kochendörfer..., U. (2005). *Inexpensive and automatic calibration for acceleration sensors*.
- [28] Brodie, M. A., Walmsley, A., & Page, W. (2008). “The static accuracy and calibration of inertial measurement units for 3D orientation”. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(6), 641–648. <https://doi.org/10.1080/10255840802326736>.
- [29] Bonnet, S., Bassompierre, C., Godin, C., Lesecq, S., & Barraud, A. (2009). “Calibration methods for inertial and magnetic sensors”. *Sensors and Actuators, A: Physical*, 156(2), 302–311. <https://doi.org/10.1016/j.sna.2009.10.008>.

- [30] Jurman, D., Jankovec, M., Kamnik, R., & Topič, M. (2007). “Calibration and data fusion solution for the miniature attitude and heading reference system”. *Sensors and Actuators, A: Physical*, 138(2), 411–420. <https://doi.org/10.1016/j.sna.2007.05.008>.
- [31] Alvarez, J. C., Álvarez, D., & López, A. M. (2018). “Accelerometry-based distance estimation for ambulatory human motion analysis”. *Sensors (Switzerland)*, 18(12), 4441. <https://doi.org/10.3390/s18124441>.
- [32] Ricci, L., Taffoni, F., & Formica, D. (2016). “On the orientation error of IMU: Investigating static and dynamic accuracy targeting human motion”. *PLoS ONE*, 11(9), 1–15. <https://doi.org/10.1371/journal.pone.0161940>.
- [33] Chen, H., Schall, M. C., & Fethke, N. (2018). “Accuracy of angular displacements and velocities from inertial-based inclinometers”. *Applied Ergonomics*. <https://doi.org/10.1016/j.apergo.2017.09.007>.
- [34] Beange, K. H. E., Chan, A. D. C., & Graham, R. B. (2018). “Evaluation of wearable IMU performance for orientation estimation and motion tracking”. *MeMeA 2018 - 2018 IEEE International Symposium on Medical Measurements and Applications, Proceedings*, 1–6. <https://doi.org/10.1109/MeMeA.2018.8438623>.
- [35] Bergamini, E., Ligorio, G., Summa, A., Vannozzi, G., Cappozzo, A., & Sabatini, A. (2014). “Estimating Orientation Using Magnetic and Inertial Sensors and Different Sensor Fusion Approaches: Accuracy Assessment in Manual and Locomotion Tasks”. *Sensors*, 14(10), 18625–18649. <https://doi.org/10.3390/s141018625> PMID - 25302810.
- [36] *Manual de usuario* correspondiente a los sensores inerciales de la gama *Xsens DOT*, la cual se encuentra disponible para su consulta a través del siguiente enlace: <https://www.xsens.com/hubfs/Downloads/Manuals/Xsens%20DOT%20User%20Manual.pdf>, *dirección web visitada el 20 de febrero de 2023*.
- [37] Resumen de las especificaciones técnicas correspondientes a los *IMUs Xsens DOT*, disponible mediante el siguiente *link*:

<https://drive.google.com/file/d/1SiEJLX3958qerDORfkOrFv3dE7ARwfH6/view?usp=sharing>, *dirección web visitada el 21 de febrero de 2023.*

[38] Guía para el usuario de los dispositivos de medida inercial *MATRIX*, disponible para su descarga mediante el siguiente *link*:
https://drive.google.com/file/d/1u2wSKulZOA4nRYdP8hZ3G5guDP9pcHJ/view?usp=share_link, *dirección web visitada el 22 de febrero de 2023.*

[39] Página web del grupo de investigación *SiMuR*, perteneciente a la Universidad de Oviedo: <https://simur.dieecs.com/>, *dirección web visitada el 28 de junio de 2023.*

[40] Hoja de características vinculada al robot colaborativo *UR3e* (fabricante: *Universal Robots*), disponible para su descarga a través del siguiente enlace: <https://www.universal-robots.com/media/1807464/ur3e-rgb-fact-sheet-landscape-a4.pdf>; *dirección web visitada el 11 de febrero de 2023.*

[41] Manual de usuario vinculado al robot colaborativo *UR3e*, disponible para su consulta a través del siguiente enlace:
https://drive.google.com/file/d/1H3sCE7Uuv0m4QklVcpcxVJvDqgZi5EMW/view?usp=drive_link, *dirección web visitada el 28 de junio de 2023.*

[42] Información sobre la estimación inercial de la orientación - Sitio web oficial de MathWorks: <https://es.mathworks.com/help/fusion/gs/determine-orientation-through-sensor-fusion.html>, *dirección web visitada el 28 de junio de 2023.*

[43] Sitio web oficial de la tecnología de captura del movimiento (*MOCAP*) *OptiTrack*: <https://optitrack.com/>, *dirección web visitada el 28 de junio de 2023.*

[44] Página *online* de la herramienta *software Motive*, destinada a la monitorización del movimiento mediante marcadores: <https://optitrack.com/software/motive/>, *dirección web visitada el 28 de junio de 2023.*

- [45] Safeea, M., & Neto, P. (2019). "Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction". *Robotics and Computer-Integrated Manufacturing*, 58, 33–42. <https://doi.org/10.1016/j.rcim.2019.01.008>.
- [46] Cifuentes, C. A., Frizera, A., Carelli, R., & Bastos, T. (2014). "Human-robot interaction based on wearable IMU sensor and laser range finder". *Robotics and Autonomous Systems*, 62(10), 1425–1439. <https://doi.org/10.1016/j.robot.2014.06.001>.
- [47] Amorim, A., Guimares, D., Mendona, T., Neto, P., Costa, P., & Moreira, A. P. (2021). "Robust human position estimation in cooperative robotic cells". *Robotics and Computer-Integrated Manufacturing*, 67. <https://doi.org/10.1016/j.rcim.2020.102035>.
- [48] Chapman, R. M., Torchia, M. T., Bell, J. E., & van Citters, D. W. (2022). "Using inertial measurement units to quantify shoulder elevation after reverse total shoulder arthroplasty: a pilot study comparing goniometric measures captured clinically to inertial measures captured ‘in-the-wild’". *Seminars in Arthroplasty JSES*. <https://doi.org/10.1053/j.sart.2022.08.002>.
- [49] Digo, E., Gastaldi, L., Antonelli, M., Pastorelli, S., Cereatti, A., & Caruso, M. (2022). "Real-time estimation of upper limbs kinematics with IMUs during typical industrial gestures". *Procedia Computer Science*, 200, 1041–1047. <https://doi.org/10.1016/j.procs.2022.01.303>.
- [50] Mc Ardle, R., del Din, S., Donaghy, P., Galna, B., Thomas, A. J., & Rochester, L. (2021). "The impact of environment on gait assessment: Considerations from real-world gait analysis in dementia subtypes". *Sensors (Switzerland)*, 21(3), 1–15. <https://doi.org/10.3390/s21030813>.
- [51] Nasrabadi, A. M., Eslaminia, A. R., Bakhshayesh, P. R., Ejtehadi, M., Alibiglou, L., & Behzadipour, S. (2022). "A new scheme for the development of IMU-based activity recognition systems for telerehabilitation". *Medical Engineering and Physics*, 108. <https://doi.org/10.1016/j.medengphy.2022.103876>.

- [52] Prigent, G., Aminian, K., Cereatti, A., Salis, F., Bonci, T., Scott, K., . . . for the Mobilise-D consortium. (2023). "A robust walking detection algorithm using a single foot-worn inertial sensor: Validation in real-life settings". *Medical and Biological Engineering and Computing*, doi:10.1007/s11517-023-02826-x.
- [53] Wilkinson, R. D., & Lichtwark, G. A. (2021). "Evaluation of an inertial measurement unit-based approach for determining centre-of-mass movement during non-seated cycling". *Journal of Biomechanics*, 126. <https://doi.org/10.1016/j.jbiomech.2021.110441>.
- [54] Shahabpoor, E., & Pavic, A. (2018). "Estimation of vertical walking ground reaction force in real-life environments using single IMU sensor". *Journal of Biomechanics*, 79, 181–190. <https://doi.org/10.1016/j.jbiomech.2018.08.015>.
- [55] Trabajo Fin de Grado titulado como: *El grupo especial ortogonal $SO(3)$* , el cual se ha desarrollado en la Universidad de Cantabria y se encuentra disponible para su consulta a través del siguiente enlace: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/25673/AhumadaVelezIgnacio-TFG-Matematicas.pdf?sequence=1>; *dirección web visitada el 11 de febrero de 2023*.
- [56] Mohinder, S. Grewal (2001). *Kalman Filtering*. John Wiley & Sons, Inc. ISBN 0-471-26638-8.
- [57] Sitio web oficial de *MathWorks*, el cual contiene abundante información sobre el método computacional *ecompass*; disponible para su consulta a través del siguiente enlace: <https://es.mathworks.com/help/nav/ref/ecompass.html>, *dirección web visitada el 20 de febrero de 2023*.
- [58] "Modern Robotics: Mechanics, Planning, and Control," Kevin M. Lynch and Frank C. Park, Cambridge University Press, 2017, ISBN 9781107156302.

11. *Anexo A*: Fundamentos teóricos ligados a las técnicas de calibración sensorial

A lo largo de esta sección se recoge la base teórica que subyace a la metodología de calibración propuesta en el *Trabajo Fin de Máster*.

11.1.- DEFINICIÓN FORMAL DEL PROCESO DE CALIBRACIÓN

La calibración puede formularse como el cálculo²¹ de la relación matemática existente entre la medida *cruda*, proporcionada por el sensor, y la magnitud *calibrada*.

Analíticamente, la información enunciada en el párrafo anterior puede expresarse de la manera que sigue:

$$[y] = s[x] + [o] \quad (11.1)$$

Donde:

- ❖ $[y]$ denota la magnitud calibrada.
- ❖ s designa al factor de escala, también denominado *sensibilidad* o *ganancia*.
- ❖ $[x]$ se corresponde con la medida cruda (*raw measurement*) suministrada por el sensor.

²¹ Nótese que las magnitudes involucradas en el proceso de calibración del sensor pueden presentar una naturaleza tanto escalar como vectorial.

De este modo, se tiene que el proceso de calibración consiste en obtener²² el par de parámetros $(s, o) \equiv (\text{sensibilidad}, \text{offset})$.

11.2.- TIPOS DE CALIBRACIÓN

Las estrategias de calibración pueden clasificarse atendiendo, fundamentalmente, a dos criterios²³:

- **Naturaleza de los parámetros a ajustar:** factor de escala, *offset* y/o alineación.
- **Etapas actuales del proceso de calibración,** existiendo tres fases bien diferenciadas:
 - (1) Ajuste paramétrico asociado al **sistema de referencia** del sensor.
 - (2) Calibración ligada al **encapsulado** de dicho **dispositivo electrónico**.
 - (3) Etapa de calibración correspondiente a la **aplicación final** del sensor.

11.2.1.- Categorías de calibración existentes en función de la naturaleza intrínseca a los parámetros de ajuste

Tal y como se ha indicado anteriormente, durante el proceso de calibración intervienen tres variables principales: la ganancia (s), el *offset* (o) y la alineación (ϑ); las cuales determinan la anatomía de los diversos algoritmos de calibración que deben utilizarse.

²² En conformidad con la información recogida en esta sección, la metodología de calibración precisa de, al menos, dos parejas de valores $(y, x) \equiv (\text{magnitud calibrada}, \text{medida cruda})$. Dichos datos permitirán resolver el sistema de ecuaciones lineales construido a partir de la expresión (11.1).

²³ **Nota:** A lo largo de este apartado, se ha decidido detallar el procedimiento de calibración para el caso de los sensores de tipo acelerómetro. Dada la ingente cantidad de tipologías de sensores existentes en la actualidad, la metodología propuesta en este documento no puede aplicarse *directamente* a cualquier dispositivo sensorial. No obstante, la filosofía de calibración es similar en la mayoría de sensores disponibles actualmente. Por ende, la algoritmia recogida en esta sección de la *Memoria* puede extrapolarse a situaciones similares que requieran calibrar varios dispositivos electrónicos de medida.

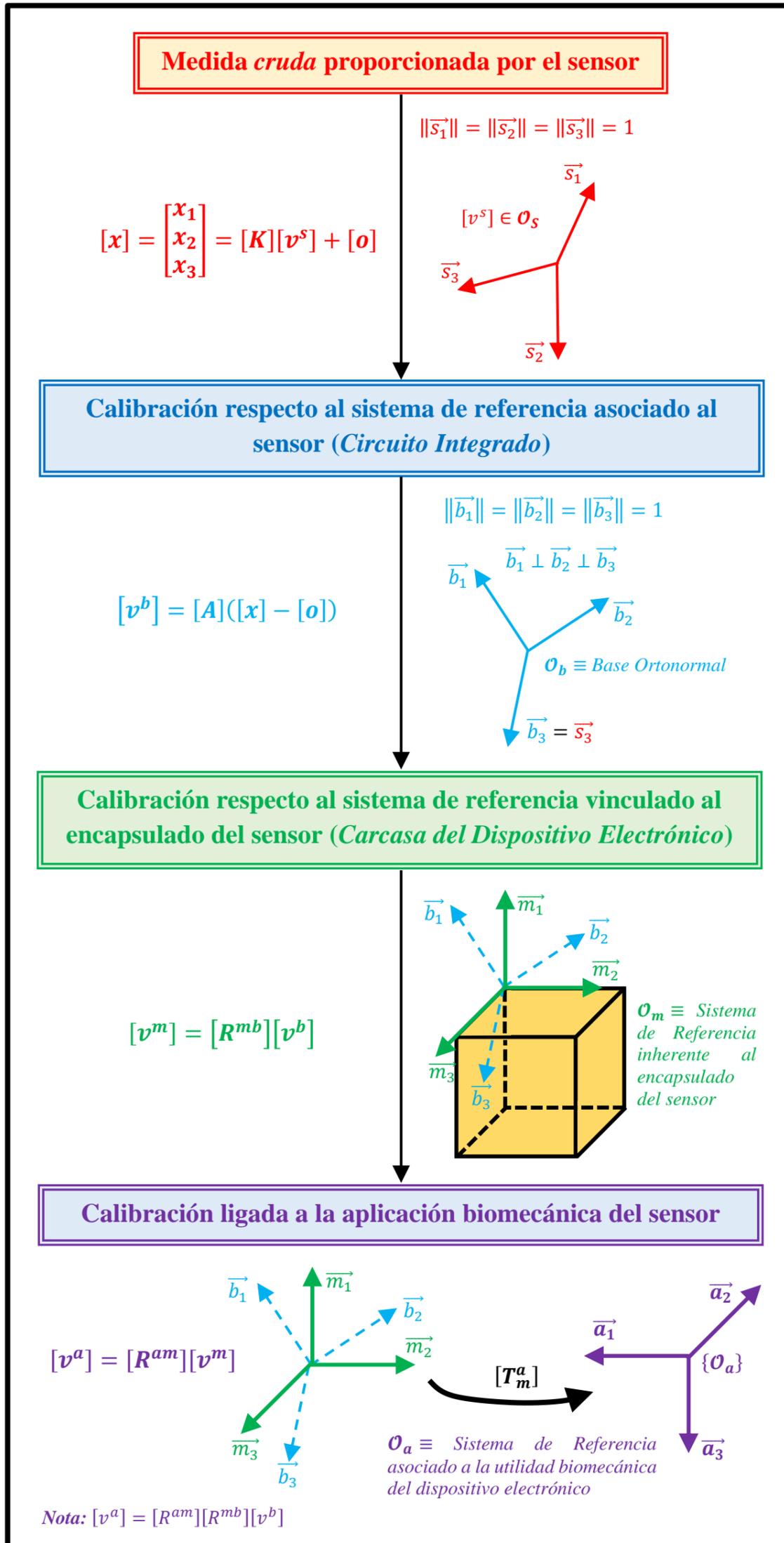
A.- Calibración de la sensibilidad (s) y el offset (o)

Los acelerómetros, y la mayor parte de los equipos de medición disponibles hoy en día, ofrecen una señal eléctrica en su etapa de salida; generada como resultado de la magnitud (física, química, biológica, etc.) medida. De aquí en adelante, dicha señal eléctrica se identificará mediante la letra y ; la cual estará expresada en Voltios [V].

Por otro lado, el valor *real* (medición cruda – *raw measurement*) de la aceleración capturada por el sensor se ha etiquetado como x ; y está medida en unidades del Sistema Internacional $\left(\left[\frac{m}{s^2}\right]\right)$.

La relación lineal existente entre x e y está condicionada por los valores de s (sensibilidad inherente al sensor) y o (*offset* presente en el dispositivo de medida), de acuerdo con la expresión matemática recogida en la ecuación (11.1).

METODOLOGÍA DE CALIBRACIÓN PROPUESTA PARA UN ACELERÓMETRO



NOTACIÓN EMPLEADA

Durante la maquetación de la figura representada en la zona izquierda de esta página, se ha hecho uso de la siguiente terminología:

- $[x]$ \equiv Diferencia de potencial eléctrico, medida en Voltios [V], suministrada por el sensor tras finalizar el proceso de medida.
- $[K]$ \equiv Matriz diagonal de sensibilidad vinculada al sensor ($[K] \in \mathbb{R}^{3 \times 3}$).
- $[v^s]$ \equiv Magnitud física monitorizada a través del sensor (medida *cruda*).
- $[o]$ \equiv Offset intrínseco a la naturaleza del dispositivo sensorial ($[o] \in \mathbb{R}^3$).
- $[v^b]$ \equiv Componentes del vector $[v^s]$ EXPRESADAS EN el sistema de referencia $\{\mathcal{O}_b\}$.
- $[A]$ \equiv Matriz triangular superior que permite efectuar la siguiente conversión de información:
 $[A]: [x, o] \mapsto [v^b] \equiv \{\mathcal{O}_s\} \mapsto \{\mathcal{O}_b\}$
 (siendo $[A] = [T]^{-1}[K]^{-1} = ([K][T])^{-1}$).
- $[v^m]$ \equiv Magnitud física capturada por el sensor y EXPRESADA EN el sistema de referencia $\{\mathcal{O}_m\}$.
- $[R^{mb}]$ \equiv Matriz de rotación que posibilita implementar el cambio de base de coordenadas indicado a continuación: $[R^{mb}]: \{\mathcal{O}_b\} \mapsto \{\mathcal{O}_m\}$
 (Nota: $[R^{mb}] = [m_1^b, m_2^b, m_3^b]^T$).
- $[v^a]$ \equiv Componentes de la variable medida, EXPRESADAS EN el sistema de referencia $\{\mathcal{O}_a\}$.
- $[R^{am}]$ \equiv Matriz de rotación utilizada durante la siguiente transformación entre sistemas de coordenadas: $\{\mathcal{O}_m\} \mapsto \{\mathcal{O}_a\}$.
- $[T_m^a]$ \equiv Matriz de Transformación Homogénea, construida para realizar el siguiente cambio de base: $\{\mathcal{O}_m\} \mapsto \{\mathcal{O}_a\}$.

Figura 11.1.- Representación esquemática del procedimiento diseñado para efectuar la correcta calibración de un acelerómetro.
[Elaboración propia].

B.- Calibración de la alineación (φ)

Adicionalmente al procedimiento de calibración diseñado para la ganancia y el *offset* del sensor, puede resultar necesario *corregir* la desalineación inicial (φ); la cual está presente con frecuencia en la mayoría de los dispositivos de medida. De este modo, utilizando la terna de parámetros (s , o , φ), se ha formulado un nuevo modelo de calibración sensorial²⁴; el cual obedece a la siguiente expresión:

$$[y] = s([x] \cdot \cos(\varphi)) + [o] \quad (11.2)$$

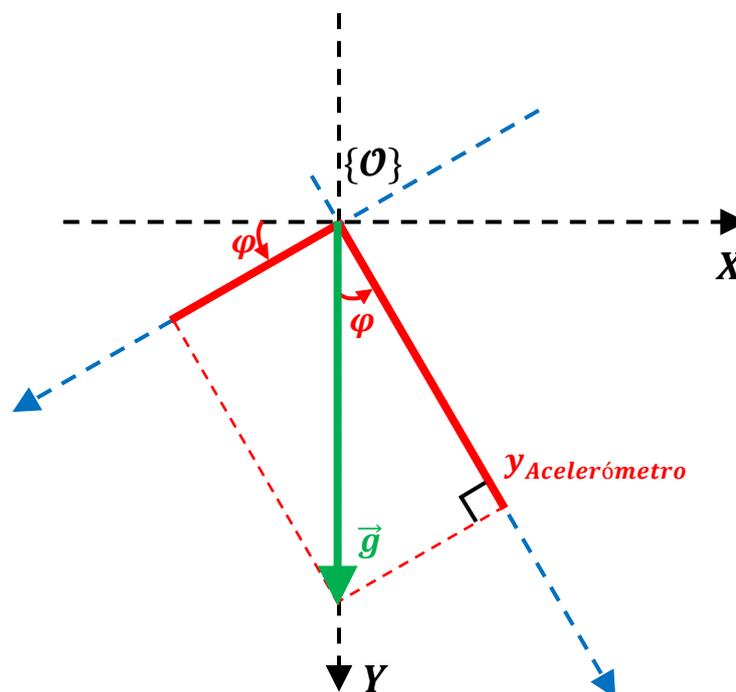


Figura 11.2.- Modelo geométrico diseñado para estimar el valor de la desalineación φ . Esta información será empleada durante la posterior calibración del sensor. [Elaboración propia].

²⁴ Obsérvese que la incorporación de la nueva incógnita φ genera una *no – linealidad* en el modelo de calibración inicialmente propuesto. Asimismo, en conformidad con la expresión recogida en la ecuación (11.2), en este caso es necesario llevar a cabo un mínimo de tres experimentos para obtener el valor de los parámetros (s , o , φ) \equiv (*factor de escala, offset, desalineación con el eje de medida*).

11.2.2.- Tipos de calibración vinculados a la fase actual del proceso de ajuste

Tal y como se ha reflejado mediante la Figura 11.1, resulta preciso diferenciar tres tipologías de calibración [29]; inherentes a la correspondiente etapa del proceso metrológico. Dicha información se muestra durante los siguientes subapartados.

A- Calibración asociada al sistema de referencia del sensor

En esta fase inicial de calibración, se pretende ajustar el *factor de escala* del dispositivo sensorial; así como el *offset* (también llamado *bias*) ineludiblemente presente en cualquier proceso de medida.

Complementariamente a los parámetros de calibración indicados en el párrafo anterior, ha de efectuarse una cuantificación del error de medida; introducido por la posible desalineación mecánica entre los ejes del sistema de referencia. Dicho *frame* cartesiano se encuentra asociado al sensor inercial que se empleará durante la captura de datos.

Como corolario, tras finalizar esta primera etapa de calibración, obsérvese que se ha conseguido *traducir* la diferencia de potencial eléctrico (suministrada por el sensor) hacia unidades de la magnitud física que se está estudiando (por ejemplo: $\left[\frac{m}{s^2}\right]$ en el caso de la aceleración de la gravedad).

B.- Ajuste paramétrico ligado al encapsulado del dispositivo electrónico

A través de esta segunda estrategia de calibración, se busca determinar una matriz de transformación homogénea²⁵ $[T] \in SO(3)$ que permita relacionar el *frame* del sensor (dispositivo electrónico) con el sistema de referencia asociado al encapsulado mecánico de dicho instrumento de medida. Esto es, ha de calcularse la matriz $[T]$ tal que:

²⁵ Téngase en cuenta que la matriz de transformación $[T]$ pertenece al grupo ortogonal especial $SO(3)$, también conocido como grupo rotacional. Por ende, dicha matriz $[T]$ verifica las dos propiedades siguientes:

- $[T]^T [T] = [I]$
- $\|T\| = 1$

Puede consultarse información adicional sobre el grupo algebraico ortogonal $SO(3)$ mediante la literatura referenciada en [55].

$$[T]: \{O_{sensor}\} \mapsto \{O_{encapsulado}\} \quad (11.3)$$

Donde el símbolo O denota al triedro cartesiano vinculado a sendos sistemas de referencia (sensor y encapsulado del dispositivo).

C.- Etapa de calibración correspondiente a la aplicación final del sensor

En última instancia, debe efectuarse una calibración determinada por la utilidad biomecánica que se desea suministrar al sensor (inercial). Para abordar esta tarea, ha de calcularse la transformación homogénea $[T]$ que posibilita relacionar el sistema de referencia $\{O_{encapsulado}\}$ con el *frame* intrínseco al sólido rígido cuya actividad se desea monitorizar. En definitiva, el problema geométrico planteado durante esta sección puede expresarse analíticamente de la manera que sigue:

$$\exists [T]? \text{ tal que } [T]: \{O_{encapsulado}\} \mapsto \{O_{Sólido Rígido Monitorizado}\} \quad (11.4)$$

Cabe destacar que la primera etapa de calibración (asociada al *frame* cartesiano del sensor) puede repetirse durante las iteraciones que se consideren adecuadas. Por otra parte, el cálculo matricial enunciado en la sección “B.” de este apartado solamente es necesario realizarlo 1 única vez; durante la vida útil del dispositivo sensorial.

Finalmente, la fase de calibración expuesta en el subapartado “C.-” ha de implementarse para cada una de las aplicaciones biomecánicas diseñadas.

11.3.- ESTRUCTURA MATEMÁTICA DE LA CALIBRACIÓN TRIDIMENSIONAL (3D)

En la mayoría de las aplicaciones prácticas, tanto biomecánicas como ingenieriles, resulta habitual implementar una calibración tridimensional de los dispositivos de medida. En este escenario, el presente apartado se ha destinado a especificar la arquitectura matemática intrínseca a dicho proceso de calibración 3D.

Recuérdese que el modelo de calibración considerado en este estudio responde a la siguiente ecuación²⁶:

$$[\mathbf{y}] = \mathbf{s}(\vec{\mathbf{x}} \cdot \vec{\mathbf{u}}) + [\mathbf{o}] = s[a_x \quad a_y \quad a_z] \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} + [\mathbf{o}] = [a_x \quad a_y \quad a_z \quad 1] \begin{bmatrix} su_x \\ su_y \\ su_z \\ 0 \end{bmatrix} \quad (11.5)$$

De acuerdo con la información indicada anteriormente, el problema de calibración se traduce en la resolución del sistema lineal de ecuaciones recogido en la expresión (11.6):

$$[\mathbf{y}]_{m \times 1} = [\mathbf{A}]_{m \times 4} [\boldsymbol{\theta}]_{4 \times 1} \Leftrightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{x_1} & a_{y_1} & a_{z_1} & 1 \\ a_{x_2} & a_{y_2} & a_{z_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x_m} & a_{y_m} & a_{z_m} & 1 \end{bmatrix} \begin{bmatrix} su_x \\ su_y \\ su_z \\ 0 \end{bmatrix} \quad (11.6)$$

Donde el subíndice m se corresponde con el número de experimentos realizados²⁷ durante el proceso de calibración sensorial.

Inspeccionando la anatomía de la matriz de medidas *crudas* $[\mathbf{A}]$, referenciada en la ecuación anterior, puede deducirse que se precisan cuatro posiciones angulares distintas ($m = 4$) para el sensor en estudio. De esta forma, el vector de incógnitas $[\boldsymbol{\theta}]$ puede calcularse algebraicamente sin más que aplicar la siguiente expresión:

$$[\boldsymbol{\theta}] = [\mathbf{A}]^{-1}[\mathbf{y}] \quad (11.7)$$

²⁶ **Nota:** El vector unitario $\vec{\mathbf{u}}$ se encuentra alienado con la dirección del eje dominante de medida, presente en el sensor. La inclusión de esta variable en el modelo de calibración permite conocer con exactitud la orientación espacial (3D) del dispositivo sensorial.

Por otro lado, téngase presente que en la notación presentada durante la ecuación (11.5), se ha considerado que $\vec{\mathbf{x}} = \vec{\mathbf{a}}$.

²⁷ Obsérvese que los elementos $a_{ij} \in [\mathbf{A}]_{m \times 4}$, $\forall i \in \{x, y, z\} \wedge \forall j \in \{1, 2, \dots, m\}$ se corresponden los valores (*crudos*) de aceleración; medidos por el sensor inercial en cada uno de los tres ejes cartesianos $\{X, Y, Z\}$.

En caso de disponer de más medidas experimentales ($m > 4$), el sistema de ecuaciones indicado en la notación (11.6) se resuelve aplicando *Mínimos Cuadrados*; haciendo uso de la ecuación mostrada a continuación:

$$[\theta_{\text{MÍNIMOS CUADRADOS}}] = ([A]^T[A])^{-1}[A]^T[y] \quad (11.8)$$

En este punto del problema de calibración, ya se conoce el valor del vector de parámetros $(su_x, su_y, su_z, o)^T$. Sin embargo, todavía no se dispone de la cantidad asociada a la ganancia del sensor (s) y el correspondiente vector unitario de orientación ($\vec{u} = [u_x, u_y, u_z]^T$). Para abordar dicha tarea, debe introducirse una nueva restricción al modelo de calibración; la cual se vincula al módulo del vector \vec{u} ($\|\vec{u}\| = 1$):

$$\|\vec{u}\| = u_x^2 + u_y^2 + u_z^2 = 1 \Rightarrow [\vartheta] = \begin{bmatrix} s \\ o \\ u_x \\ u_y \\ u_z \end{bmatrix} \equiv \text{Parámetros de calibración} \quad (11.9)$$

Por otra parte, una interpretación geométrica alternativa de este proceso de calibración consiste en definir la latitud (ϑ) y la longitud (ψ); vinculadas al eje del sensor adoptado durante la medición que se desea realizar. Este planteamiento puede visualizarse gráficamente en la siguiente figura:

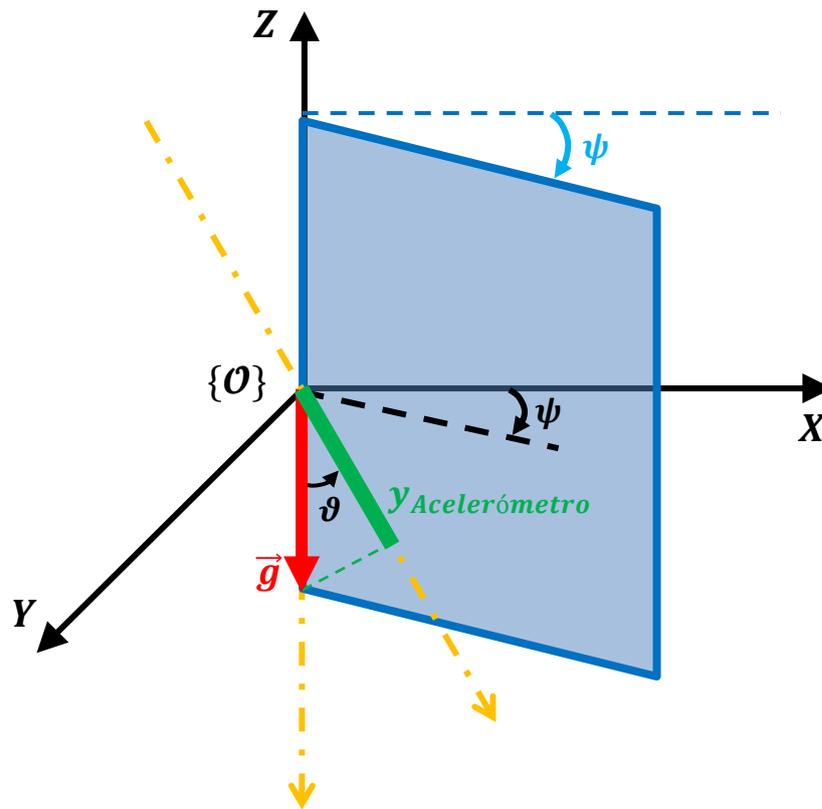


Figura 11.3.- Representación esquemática de la disposición geométrica considerada durante la metodología de calibración tridimensional. [Elaboración propia].

Para la situación planteada en la Figura 11.3, deberían sintonizarse cuatro variables de calibración: (*ganancia, offset, latitud, longitud*) $\equiv (s, o, \vartheta, \psi)$.

Finalmente, resulta interesante modelar la situación en la que se emplean varios acelerómetros durante la calibración. Para ello, han de ampliarse las dimensiones vinculadas a los vectores $[\theta]$ e $[y]$; obteniéndose la expresión recogida a continuación:

$$[y]_{m \times n} = [A]_{m \times 4} [\vartheta \equiv x]_{4 \times n} \quad (11.10)$$

Considerando que se han utilizado n acelerómetros ($[a, b, c, \dots, n]$) durante los m experimentos de calibración diseñados, el sistema de ecuaciones resultante adopta la siguiente forma:

$$\begin{aligned}
 & \begin{bmatrix} y_1^a & y_1^b & y_1^c & \cdots & y_1^n \\ y_2^a & y_2^b & y_2^c & \cdots & y_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m^a & y_m^b & y_m^c & \cdots & y_m^n \end{bmatrix}_{m \times n} = \\
 & = \begin{bmatrix} a_{x_1} & a_{y_1} & a_{z_1} & 1 \\ a_{x_2} & a_{y_2} & a_{z_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x_m} & a_{y_m} & a_{z_m} & 1 \end{bmatrix}_{m \times 4} \begin{bmatrix} s^a u_x^a & s^b u_x^b & s^c u_x^c & \cdots & s^n u_x^n \\ s^a u_y^a & s^b u_y^b & s^c u_y^c & \cdots & s^n u_y^n \\ s^a u_z^a & s^b u_z^b & s^c u_z^c & \cdots & s^n u_z^n \\ o^a & o^b & o^c & \cdots & o^n \end{bmatrix}_{4 \times n} \quad (11.11)
 \end{aligned}$$

Por ende, se necesita registrar el vector 3D de aceleración (\vec{a}_i) en cada uno de los m experimentos llevados a cabo. Finalmente, los parámetros de calibración definidos ($[s, o, u_x, u_y, u_z]^T \forall i \in \{1, 2, \dots, m\}; \forall j \in \{a, b, c, \dots, n\}$) se obtendrán mediante la resolución algebraica del sistema de ecuaciones planteado en (11.11).

11.4.- ALGORITMOS AVANZADOS DE CALIBRACIÓN SENSORIAL

En la actualidad, existen algoritmos más complejos destinados a implementar el procedimiento *Sensor Frame Calibration*. A continuación, se recoge el esquema de cálculo vinculado a uno de ellos [29]:

- (1) $\{F_S\} = [\vec{s}_1 \ \vec{s}_2 \ \vec{s}_3]; \vec{v}^S = [v_1^S \ v_2^S \ v_3^S]; v_i^S = \vec{v} \cdot \vec{s}_i$.
- (2) En (1) se tiene la siguiente relación: $\vec{v} = f(\vec{h}_{MAG}^S, \vec{g}_{GRAV}^S - \vec{a}_{GRAV}^S)$.
- (3) Tensión de salida del sensor: $x_i = k_i v_i^S + o_i$.
- (4) Expresión matricial de (3): $[x] = [k][v^S] + [o] = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix} [v^S] + \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix}$.
- (5) $[s] = [T][b]$, $T_{ij} = \vec{s}_i \cdot \vec{b}_j$. Ortonormalización de $\{F_b\}$.
- (6) $[v^S] = [T][v^b]$; siendo $\cos(\alpha_{ij}) = \vec{s}_i \cdot \vec{s}_j$.
- (7) $[v^b] = [T]^{-1}[v^S] = [T]^{-1}\{[k]^{-1}([x] - [o])\} = [A]([x] - [o])$.
- (8) En (7) se ha incluido la siguiente notación: $[A] = [T]^{-1}[k]^{-1} = (kT)^{-1}$.
- (9) $\|v_b\|^2 = 1 \Rightarrow \|[A]([x] - [o])\|^2 = 1 \Rightarrow ([x_n] - [o])^T ([A]^T [A]) ([x_n] - [o]) = 1$.
Téngase en cuenta que $([A]^T [A]) \in \mathbb{R}^{3 \times 3}$.

12. *Anexo B*: Algoritmia vinculada al *Filtro de Kalman*. Estimación de la orientación

12.1.- ¿EN QUÉ CONSISTE EL *FILTRO DE KALMAN*?

El *Filtro de Kalman*²⁸ es un método iterativo que permite conocer el estado actual correspondiente a un determinado sistema dinámico.

En conformidad con la arquitectura matemática de este algoritmo, la cual se detallará posteriormente, el *Filtro de Kalman* resulta idóneo para su ejecución en tiempo real; puesto que solamente requiere datos vinculados al estado anterior del sistema, las señales de entrada actuales y la respectiva incertidumbre de las mediciones [56].

A pesar de que este método computacional inicialmente fue diseñado para abordar problemas propios de la robótica móvil, su rango de aplicación se ha extendido hacia sectores como la econometría, el Procesamiento Digital de Señales (*DSP - Digital Signal Processing*), la geolocalización (*GPS*), las Comunicaciones por Satélite; entre otros.

12.2.- ANATOMÍA DEL MÉTODO ITERATIVO DE *KALMAN*

El *Algoritmo Recursivo de Kalman* pertenece a la familia de filtros identificados como *Gaussianos*. El procedimiento de cálculo, destinado a estimar el estado del sistema en

²⁸ El origen histórico de este algoritmo se remonta al año 1960, en el cual el Ingeniero Eléctrico y Matemático Rudolf Emil Kalman propuso dicha estrategia numérica para ofrecer una solución a los problemas de estimación de trayectorias. Inicialmente, el *Filtro de Kalman* tuvo un gran impacto; debido a su incorporación en los sistemas de control diseñados durante el *Programa Apolo*.

A través de la literatura referenciada en [56], se recoge información adicional sobre la historia, el funcionamiento y las potenciales aplicaciones que posee el *Filtro de Kalman*.

un instante de tiempo genérico t , puede dividirse en dos grandes bloques; vinculados, respectivamente, a la fase de predicción y actualización de dicho estado.

12.2.1.- Etapa de predicción del estado ($\hat{\boldsymbol{\mu}}_{k|k-1}$) asociado al sistema

Inicialmente, el *Filtro de Kalman* implementa una **estimación a priori del estado actual** ($\hat{\boldsymbol{\mu}}_{k|k-1}$); ligado al sistema en estudio. Matemáticamente, esta predicción se formula mediante la siguiente ecuación:

$$\hat{\boldsymbol{\mu}}_{k|k-1} = \mathbf{A}_k \boldsymbol{\mu}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \quad (12.1)$$

Donde:

- $\hat{\boldsymbol{\mu}}_{k|k-1}$ se corresponde con la predicción efectuada para el estado actual del sistema.
- $\boldsymbol{\mu}_{k-1|k-1}$ es el estado del sistema durante la muestra digital inmediatamente anterior.
- \mathbf{A}_k denota a la matriz de cambio de estado, la cual permite realizar la siguiente transición en tiempo discreto: $\mathbf{A}_k: \boldsymbol{\mu}_{k-1|k-1} \mapsto \hat{\boldsymbol{\mu}}_{k|k-1}$.
- \mathbf{u}_k es la señal de entrada al sistema que se recibe en la muestra digital k .
- \mathbf{B}_k designa a la matriz encargada de desarrollar la siguiente transformación matemática: $\mathbf{u}_k \mapsto \hat{\boldsymbol{\mu}}_{k|k-1}$.

A continuación, debe cuantificarse el **error vinculado a la estimación de estado** realizada con anterioridad. Para ello, se procede a la construcción de la matriz de covarianzas $\hat{\boldsymbol{\Sigma}}_{k|k-1}$; de acuerdo con la ecuación recogida a continuación:

$$\hat{\boldsymbol{\Sigma}}_{k|k-1} = \mathbf{A}_k \hat{\boldsymbol{\Sigma}}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{R}_k \quad (12.2)$$

En la expresión (12.2) se ha utilizado la siguiente notación:

- $\hat{\boldsymbol{\Sigma}}_{k|k-1}$: matriz de covarianzas calculada durante la iteración k del *Filtro de Kalman*. Dichas covarianzas se vinculan a la predicción del estado llevada a cabo.

- $\hat{\Sigma}_{k-1|k-1}$: *idem* que en el caso anterior, con la salvedad de que la matriz de covarianzas generada se asocia a la iteración $k - 1$ del algoritmo.
- \mathbf{A}_k^T : matriz traspuesta de transición de estado.
- \mathbf{R}_k : varianza correspondiente al ruido blanco introducido por el dispositivo sensorial.

12.2.2.- Fase de corrección/actualización

Durante esta segunda etapa del proceso de cálculo, se realiza una actualización sobre el estado inicialmente estimado para el sistema. En este aspecto, se emplean los datos suministrados por los sensores disponibles durante el estudio.

A continuación, se enumeran las estrategias computacionales que rigen la presente fase de corrección del estado estimado:

- ❖ **Integración en el modelo predictivo de las medidas suministradas (\mathbf{z}_k) por la tecnología sensorial.** Esta actividad permite obtener el *residuo* ligado a la medición implementada. Analíticamente, la actualización de dicho *residuo* puede formularse como:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{C}_k \cdot \hat{\mathbf{u}}_{k|k-1} \quad (12.3)$$

Obsérvese que la notación empleada en la ecuación (12.3) responde a la siguiente información:

- $\tilde{\mathbf{y}}_k$: estimación del error cometido (*vector columna*) durante la actualización del estado, asociado al sistema dinámico en estudio.
- \mathbf{z}_k : datos medidos (*vector columna*) a través del sensor considerado.
- \mathbf{C}_k : esta matriz permite combinar la información medida por el sensor y el estado actual ($\hat{\mathbf{u}}_{k|k-1}$) del sistema dinámico. *Nótese que la anatomía de \mathbf{C}_k considera que las mediciones realizadas son ideales, esto es, sin presencia de ruido.*

❖ **Cómputo de la Ganancia de Kalman (\mathbf{K}_k)**

$$\mathbf{K}_k = \hat{\Sigma}_{k|k-1} \mathbf{C}_k^T (\mathbf{C}_k \hat{\Sigma}_{k|k-1} \mathbf{C}_k^T + \mathbf{Q}_k)^{-1} \quad (12.4)$$

En la siguiente lista se especifican los nuevos parámetros introducidos durante el cálculo de la ganancia \mathbf{K}_k :

- \mathbf{K}_k : *Ganancia de Kalman*, la cual se desea cuantificar en la presente fase de actualización/corrección del algoritmo.
 - \mathbf{C}_k^T : traspuesta de la matriz \mathbf{C}_k , cuyo significado físico ya ha sido referenciado con anterioridad.
 - \mathbf{Q}_k : matriz de covarianza que modela el ruido inherente a las capturas de datos realizadas por los sensores.
- **Evaluación *a posteriori* del estado actual ($\mu_{k|k-1}$) ligado al sistema dinámico.** En este paso, combinando el valor del *residuo* $\tilde{\mathbf{y}}_k$ y la estimación *a priori* del estado $\hat{\mu}_{k|k-1}$, se obtendrá el estado *real* del sistema modelado ($\mu_{k|k-1}$); de acuerdo con la siguiente expresión:

$$\mu_{k|k-1} = \hat{\mu}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (12.5)$$

- Finalmente, se debe calcular la matriz de covarianzas $\Sigma_{k|k-1}$; la cual permite identificar el error cometido en la estimación de $\mu_{k|k-1}$ enunciada durante la ecuación (12.5). La obtención de la matriz $\Sigma_{k|k-1}$ obedece a la expresión mostrada a continuación:

$$\Sigma_{k|k-1} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \hat{\Sigma}_{k|k-1} \quad (12.6)$$

Donde \mathbf{I} es la *Matriz Identidad* del orden correspondiente.

12.2.3.- Interpretación del *Filtro de Kalman* como una *Caja Negra*

En conformidad con la información expuesta anteriormente, resulta posible considerar al *Filtro de Kalman* como una *Caja Negra* (*Black Box*) que recibe cuatro señales de entrada y retorna el valor de dos parámetros de salida.

Abundando en lo expresado durante el párrafo anterior, cada ejecución del *Filtro de Kalman* precisa conocer el estado del sistema en la iteración $k - 1$ ($\mu_{k-1|k-1}$), su correspondiente matriz de covarianzas ($\Sigma_{k-1|k-1}$), el valor actual de las señales de entrada al sistema (u_k) y las medidas realizadas por los sensores (z_k) en la muestra digital k . Por otro lado, el algoritmo *devuelve* el estado actual del sistema dinámico ($\mu_{k|k-1}$); junto con su respectivo error de estimación en forma de matriz de covarianzas ($\Sigma_{k|k-1}$).



Figura 12.1.- Representación conceptual del *Filtro de Kalman* como una *Caja Negra*.
[Elaboración propia].

12.3.- ESTIMACIÓN DE LA ORIENTACIÓN ESPACIAL MEDIANTE EL *FILTRO DE KALMAN*. TÉCNICA DE FUSIÓN SENSORIAL

En conformidad con la información recogida durante los anteriores subapartados, el *Filtro de Kalman* constituye una herramienta muy utilizada en el modelado y control de sistemas mediante espacio de estados. Asimismo, una de sus principales aplicaciones prácticas consiste en el cálculo de la orientación tridimensional; asociada a un determinado sólido rígido.

A continuación, se procede a detallar el algoritmo de `imufilter`; el cual ya se encuentra implementado en *MatLab* y permite estimar la orientación de 3D de un objeto.

El método `imufilter` emplea el *Filtro de Kalman*²⁹ para obtener el error inherente a la orientación espacial, la aceleración lineal y el movimiento del giroscopio. Con dichos datos se calculará, en tiempo real, la orientación tridimensional del objeto; junto con la velocidad angular desarrollada.

El vector de error obtenido en cada iteración del algoritmo se ha etiquetado como $\boldsymbol{\mu}_{k|k-1}$, el cual contiene la siguiente información:

$$\boldsymbol{\mu}_{k|k-1} = \begin{bmatrix} \boldsymbol{\vartheta}_k \\ \boldsymbol{\delta}_k \\ \boldsymbol{\varepsilon}_{a_k} \end{bmatrix}_{9 \times 1} = \mathbf{F}_k \begin{bmatrix} \boldsymbol{\vartheta}_{k-1} \\ \boldsymbol{\delta}_{k-1} \\ \boldsymbol{\varepsilon}_{a_{k-1}} \end{bmatrix} + \boldsymbol{\tau}_k \quad (12.7)$$

Donde:

- $[\boldsymbol{\vartheta}_k]_{3 \times 1}$ se vincula al error de orientación generado durante la iteración k del algoritmo, el cual se expresa en grados decimales [$^\circ$].
- $[\boldsymbol{\delta}_k]_{3 \times 1}$ denota al *offset* inherente al giroscopio en el instante de tiempo k . Dicho parámetro posee unidades de velocidad angular ($[^\circ/s]$).
- $[\boldsymbol{\varepsilon}_{a_k}]_{3 \times 1}$ se corresponde con el error de aceleración existente en iteración k del *Filtro de Kalman*. Este vector se ha expresado utilizando unidades de \vec{g} (en tanto por uno).
- $[\mathbf{F}_k]$ modela la matriz de transición de estado entre $\boldsymbol{\mu}_{k-1|k-1}$ y $\boldsymbol{\mu}_{k|k-1}$.
- $[\boldsymbol{\tau}_k]_{9 \times 1}$ permite incorporar en el modelo de *Kalman* el efecto producido por la existencia de ruido blanco.

De forma a las expresiones analíticas detalladas exhaustivamente durante el apartado "12.2.- ANATOMÍA DEL MÉTODO ITERATIVO DE *KALMAN*", se muestra a

²⁹ **Nota:** En el presente caso de estudio, la naturaleza geométrica del algoritmo se corresponde con un *Filtro de Kalman* de seis ejes.

continuación la nomenclatura adoptada para formular el Filtro de Kalman durante la estimación de la orientación espacial tridimensional:

$$\hat{\boldsymbol{\mu}}_{k|k-1} = \mathbf{F}_k \boldsymbol{\mu}_{k-1|k-1} \quad (12.8)$$

$$\hat{\boldsymbol{\Sigma}}_{k|k-1} = \mathbf{F}_k \boldsymbol{\Sigma}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (12.9)$$

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\boldsymbol{\mu}}_{k|k-1} \quad (12.10)$$

$$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_{k|k-1} \mathbf{H}_k^T \quad (12.11)$$

$$\mathbf{K}_k = \hat{\boldsymbol{\Sigma}}_{k|k-1} \mathbf{H}_k^T (\mathbf{S}_k)^{-1} \quad (12.12)$$

$$\boldsymbol{\mu}_{k|k-1} = \hat{\boldsymbol{\mu}}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \quad (12.13)$$

$$\boldsymbol{\Sigma}_{k|k-1} = \hat{\boldsymbol{\Sigma}}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_{k|k-1} \quad (12.14)$$

Con objeto de simplificar las ecuaciones mostradas anteriormente, se han realizado las siguientes hipótesis:

- La estimación *a priori* del vector error en la iteración k ($\hat{\boldsymbol{\mu}}_{k|k-1}$) se ha considerado de valor 0; puesto que dicha variable modeliza el sesgo inicial ($k = 0$) presente en la metodología de cálculo de la orientación.
- Como consecuencia del hecho indicado en el punto anterior, el parámetro de transición de estado \mathbf{F}_k es la matriz nula.

Por ende, las ecuaciones (12.8) a (12.14) pueden reformularse de la siguiente forma:

$$\hat{\boldsymbol{\mu}}_{k|k-1} = 0 \quad (12.15)$$

$$\hat{\Sigma}_{k|k-1} = \mathbf{Q}_k \quad (12.16)$$

$$\mathbf{y}_k = \mathbf{z}_k \quad (12.17)$$

$$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \hat{\Sigma}_{k|k-1} \mathbf{H}_k^T \quad (12.18)$$

$$\mathbf{K}_k = \hat{\Sigma}_{k|k-1} \mathbf{H}_k^T (\mathbf{S}_k)^{-1} \quad (12.19)$$

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{K}_k \mathbf{y}_k \quad (12.20)$$

$$\Sigma_{k|k-1} = \hat{\Sigma}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \hat{\Sigma}_{k|k-1} \quad (12.21)$$

Nota Adicional 1: A lo largo del presente apartado, no se ha detallado el significado matemático de la notación utilizada en las ecuaciones expuestas. La justificación de este hecho radica en que dichas expresiones son muy similares a las ya enunciadas durante la sección “12.2.- ANATOMÍA DEL MÉTODO ITERATIVO DE KALMAN”. Dentro de dicho subapartado puede consultarse detalladamente la estructura y el significado de las ecuaciones que conforman el Filtro de Kalman.

Nota Adicional 2: Obsérvese que el algoritmo de estimación de la orientación, propuesto durante este apartado, se basa en la fusión de dos conjuntos de datos. Dichos datasets se vinculan, respectivamente, a la información proporcionada por el acelerómetro y el giroscopio; presentes en el sensor inercial considerado durante el estudio.

DIAGRAMA DE BLOQUES LIGADO A LA ESTIMACIÓN DE LA ORIENTACIÓN MEDIANTE EL *FILTRO DE KALMAN*

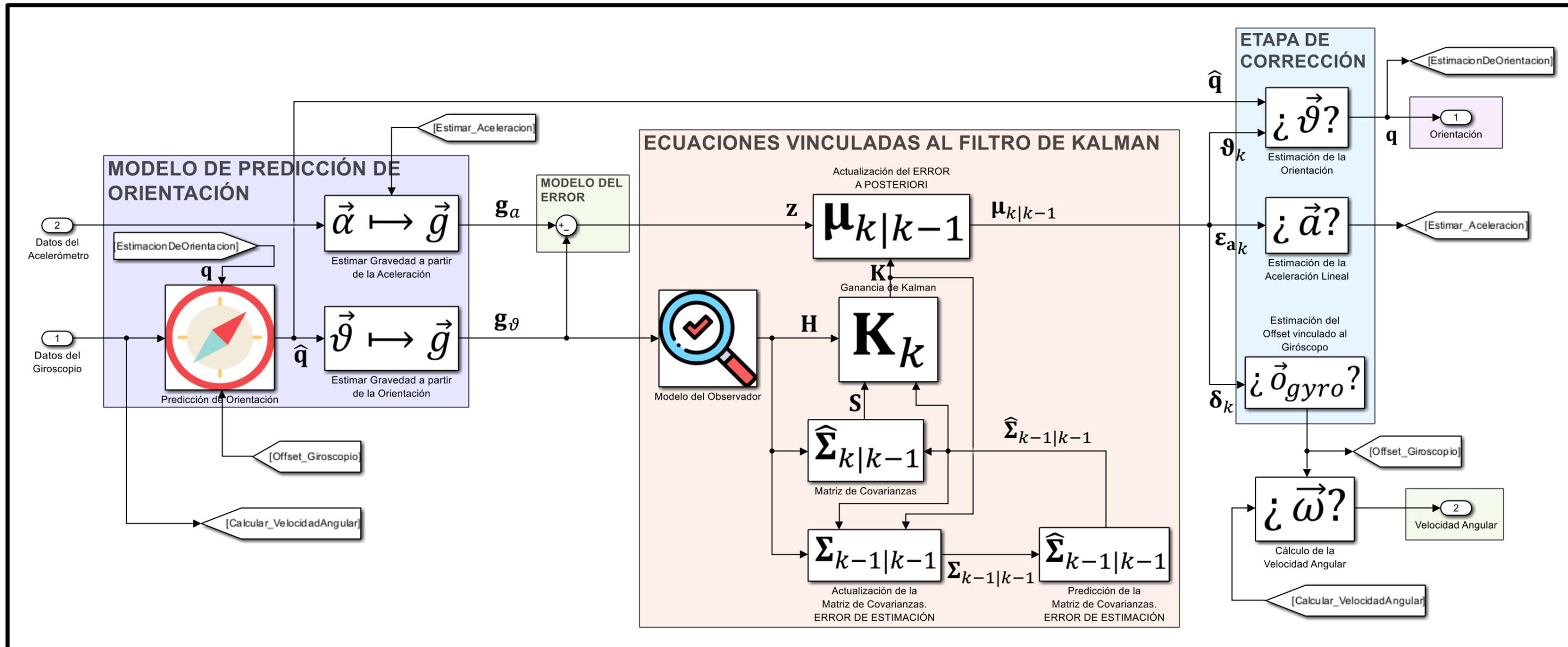


Figura 12.2.- Arquitectura diseñada (*Filtro de Kalman*) para estimar la orientación espacial de un sólido rígido. [Elaboración propia].

DETALLE DE LA NUEVA TERMINOLOGÍA MATEMÁTICA

INTRODUCIDA EN LA FIGURA ANTERIOR

A continuación, se recoge el significado de las señales etiquetadas en el diagrama de bloques expuesto mediante la Figura 12.2:

- **q**: estimación *a posteriori* de la orientación espacial, vinculada al sólido rígido en estudio.
- **q̂**: estimación *a priori* de la orientación tridimensional, ligada al sólido rígido en estudio.
- **g_a**: aceleración de la gravedad calculada a partir de los datos de aceleración, suministrados por el sensor inercial (acelerómetro) utilizado.
- **g_g**: aceleración de la gravedad calculada a partir de los datos de velocidad angular de giro, suministrados por el sensor inercial (giroscopio) utilizado.
- **z**: información proporcionada por la tecnología sensorial disponible en el sistema.
- **H**: matriz auxiliar destinada a incluir la información sensorial (**z**) dentro de la predicción del estado actual ($\boldsymbol{\mu}_{k|k-1}$).
- **K**: *Ganancia de Kalman* calculada por el algoritmo en la ejecución número *k*.
- **S**: matriz auxiliar empleada durante la obtención de la *Ganancia de Kalman K* en la iteración *k* del método numérico. Recuérdese que $\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_{k|k-1} \mathbf{H}_k^T$.
- $\hat{\boldsymbol{\Sigma}}_{k-1|k-1}$: estimación *a priori* de la matriz de covarianzas, vinculada al estado del sistema, durante la iteración *k* - 1 del algoritmo.
- $\boldsymbol{\Sigma}_{k-1|k-1}$: cálculo *a posteriori* de la matriz de covarianzas, vinculada al estado del sistema, durante la iteración *k* - 1 del algoritmo numérico.
- Los parámetros $\boldsymbol{\mu}_{k|k-1}$, $\boldsymbol{\vartheta}_k$, $\boldsymbol{\varepsilon}_{a_k}$ y $\boldsymbol{\delta}_k$ ya han sido referenciados al comienzo del presente subapartado “12.3”.

13. *Anexo* C: Fundamento algorítmico ligado al método de fusión sensorial *ecompass*

13.1.- INTRODUCCIÓN

Este *Anexo* del documento *Memoria* se ha confeccionado para recoger las bases teóricas que rigen el funcionamiento del algoritmo *ecompass* [57], disponible para su utilización en *MatLab*. Dicha estrategia de fusión sensorial precisa conocer los datos proporcionados por el **acelerómetro** y el **magnetómetro**, existentes en la unidad de medida inercial considerada.

De forma análoga a la mayoría de los métodos computacionales destinados a estimar la orientación espacial de un sólido rígido, el algoritmo *ecompass* proporciona una matriz de rotación³⁰ ($[R]$) como resultado final.

13.2.- CONFIGURACIÓN GEOMÉTRICA DEL PROBLEMA PLANTEADO

Inicialmente, la matriz de rotación indicada en la sección anterior adopta la siguiente anatomía:

$$[R] = [(a \times m) \times a \quad a \times m \quad a] \quad (13.1)$$

³⁰ Alternativamente, la orientación espacial calculada por la función *ecompass* también puede expresarse en forma de cuaternión.

Donde el parámetro \mathbf{a} se corresponde con los datos de **aceleración** registrados, mientras que \mathbf{m} denota a los valores de **intensidad magnética**.

La matriz de rotación $[R]$ permite relacionar las coordenadas tridimensionales (x, y, z) de un sensor³¹ con el sistema de referencia local *NED* (*North, East and Down Coordinates*); ubicado en dicho punto del espacio. Esta configuración geométrica puede observarse gráficamente mediante la siguiente figura:

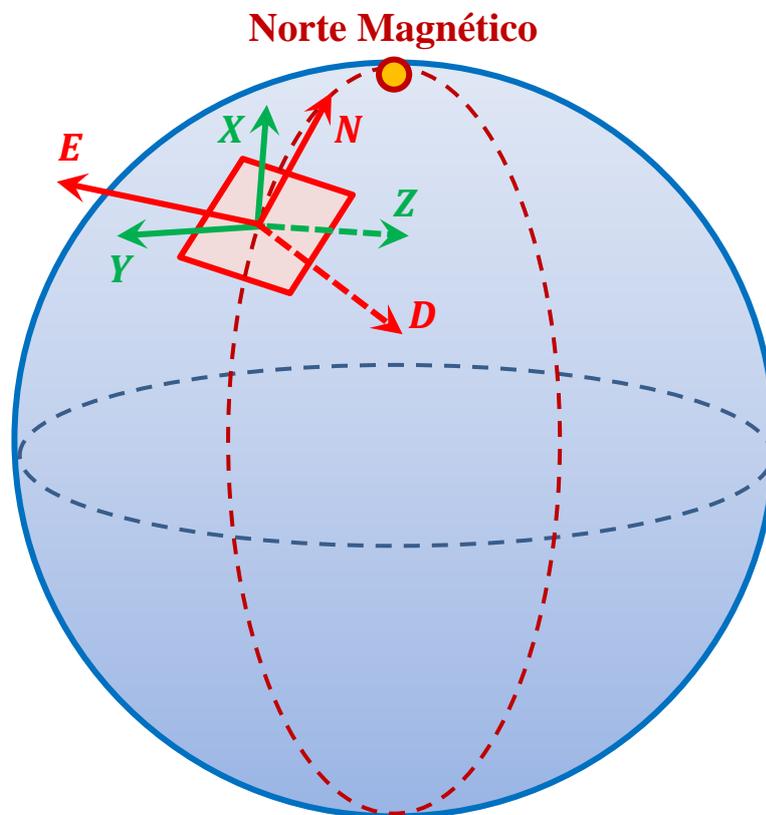


Figura 13.1.- Disposición espacial de los sistemas de coordenadas *NED* y cartesiano, vinculados a una posición *3D* genérica. [Elaboración propia].

De este modo, la orientación del sólido rígido vinculado al sensor puede calcularse de la manera que sigue:

³¹ **Nota:** Considérese que la posición cartesiana (x, y, z) , definida para la unidad de medida sensorial, está ligada a un punto genérico en el espacio.

$$[R]_{3 \times 3} \cdot [p_{PADRE}]_{3 \times 1} = [p_{HIJO}]_{3 \times 1} \quad (13.2)$$

En la expresión (13.2) se ha utilizado la notación indicada a continuación:

- ❖ $[p_{PADRE}]$: vector columna (dimensión 3 x 1) en el que se almacenan las coordenadas cartesianas de posición espacial, las cuales están expresadas respecto al sistema de referencia *padre*.
- ❖ $[R]$: matriz de rotación (dimensión 3 x 3) que implementa la transformación de coordenadas cartesianas $[p_{PADRE}] \mapsto [p_{HIJO}]$. Dicha matriz se encuentra ligada a la orientación tridimensional del *frame* cartesiano *hijo*.
- ❖ $[p_{HIJO}]$: vector columna (dimensión 3 x 1) en el que se almacenan las coordenadas cartesianas de posición espacial, las cuales están expresadas respecto al sistema de referencia *hijo*.

13.3.- LECTURA DEL ACELERÓMETRO PRESENTE EN EL SENSOR INERCIAL

Los datos de aceleración capturados mediante una unidad de medida inercial se encuentran condicionados por la orientación 3D de dicho *IMU*. En caso de efectuar la adquisición de datos durante posiciones estáticas, los valores arrojados por el acelerómetro se corresponden con la aceleración de la gravedad (\vec{g}).

De acuerdo con lo expresado anteriormente, resulta preciso emplear una operación de rotación entre el *frame* *NED*³² y el sistema cartesiano de referencia inherente al sensor; tal y como se refleja en la Figura 13.2:

³² **Nota:** Cabe resaltar que, en la construcción del sistema de referencia *NED*, se utiliza el norte magnético (*N*) terrestre. Este aspecto cobra importancia, puesto que los nortes geográfico y magnético no son coincidentes.

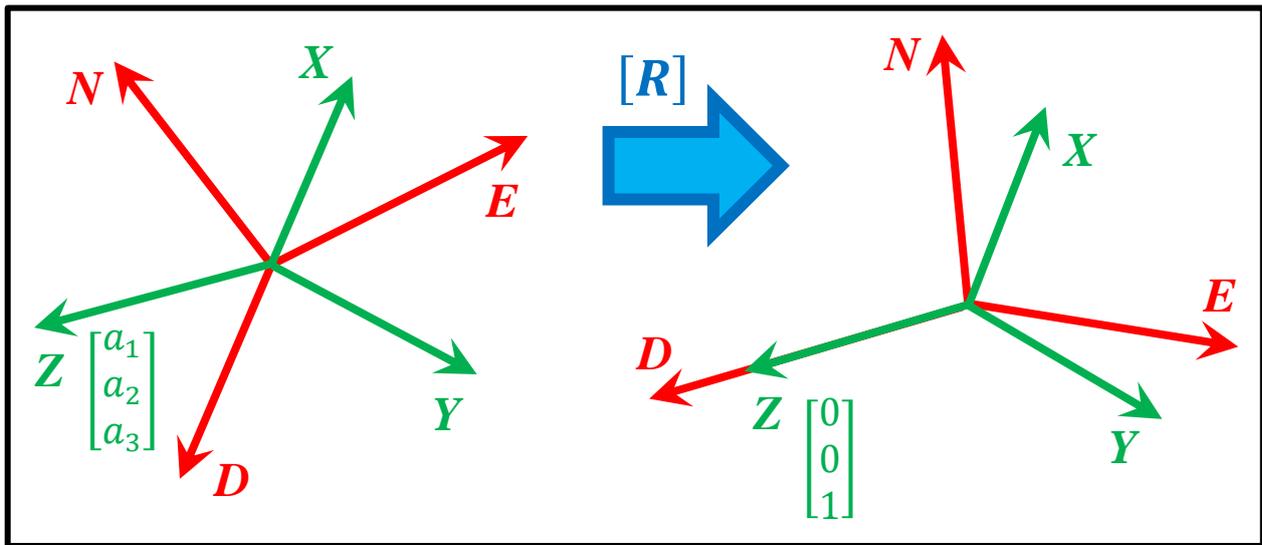


Figura 13.2.- Transformación geométrica diseñada para alinear el eje Z del *frame* cartesiano vinculado al sensor y el sistema de referencia *NED*. [Elaboración propia].

En la figura anterior puede observarse la alineación geométrica conseguida entre el eje Z del sistema de coordenadas del sensor y la componente *D* (*Down*) asociada al *frame* *NED*. De esta forma, la lectura proporcionada por el acelerómetro será $[0 \ 0 \ g]^T \left[\frac{m}{s^2} \right]$. Alternativamente, dicho resultado también puede expresarse en unidades de $[g]$ como $[0 \ 0 \ 1]^T [g]$.

Por otro lado, considerando una orientación genérica para el sistema cartesiano $\{x, y, z\}$ se tiene que la lectura suministrada por el acelerómetro es $[a_1 \ a_2 \ a_3]^T \left[\frac{m}{s^2} \right]$.

Matemáticamente, la rotación geométrica maquetada en la Figura 13.2 se traduce en la siguiente operación matricial:

$$\begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (13.3)$$

Mediante inspección, puede deducirse que los datos de aceleración capturados a través del sensor inercial se corresponden con la tercera columna de la matriz de rotación $[R]$. Por ende, resulta correcto afirmar que:

$$\begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (13.4)$$

13.3.- LECTURA DEL MAGNETÓMETRO EXISTENTE EN LA UNIDAD DE MEDIDA INERCIAL

Análogamente al caso anteriormente expuesto, relativo a la aceleración de la gravedad, también resulta necesario analizar la **orientación relativa de los frames** $\{N, E, D\} - \{x, y, z\}$ en materia de intensidad de campo magnético. Esto es, dicha orientación espacial condicionará el resultado obtenido en el vector de campo magnético \vec{m} .

Sea $P(x, y, z)$ un punto genérico ubicado en el espacio $3D$. La lectura del magnetómetro correspondiente a dicho punto se encuentra orientada hacia el norte magnético. Asimismo, dicho resultado verifica la propiedad de estar contenido en el plano $N - D$. Este aspecto puede observarse en la siguiente figura:

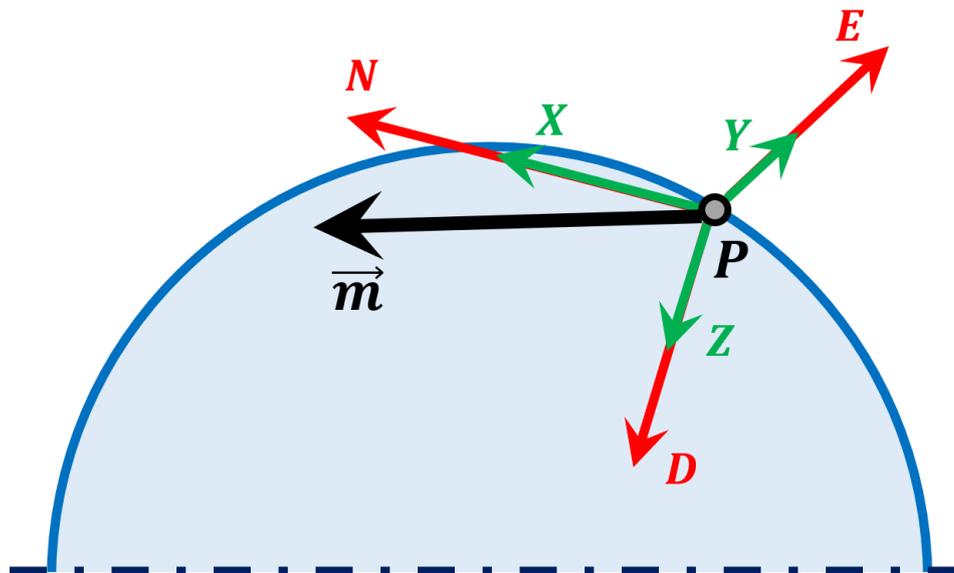


Figura 13.3.- Corte realizado en el plano $N - D$, con la finalidad de representar gráficamente la disposición espacial ligada a los sistemas de coordenadas $\{X, Y, Z\}$ y $\{N, E, D\}$. Adicionalmente, puede visualizarse el resultado generado para el vector de campo magnético (\vec{m}). [Elaboración propia].

Puesto que el triedro NED es ortogonal, se verifica la siguiente propiedad geométrica³³:

$$E = D \times N \quad (13.5)$$

En conformidad con la expresión (13.5), puede deducirse que el vector E es perpendicular al plano $N - D$.

A través de la Figura 13.4, se ha representado gráficamente el efecto tridimensional que posee la matriz de rotación $[R]$; destinada a *corregir* la desalineación existente entre los sistemas de coordenadas $\{N, E, D\}$ y $\{X, Y, Z\}$. En caso de que los sistemas de referencia $\{X, Y, Z\}$ y $\{N, E, D\}$ se encuentren alineados, puede demostrarse que los vectores \vec{a} (aceleración registrada) y \vec{m} (intensidad de campo magnético) se ubican en el interior del plano $N - D$.

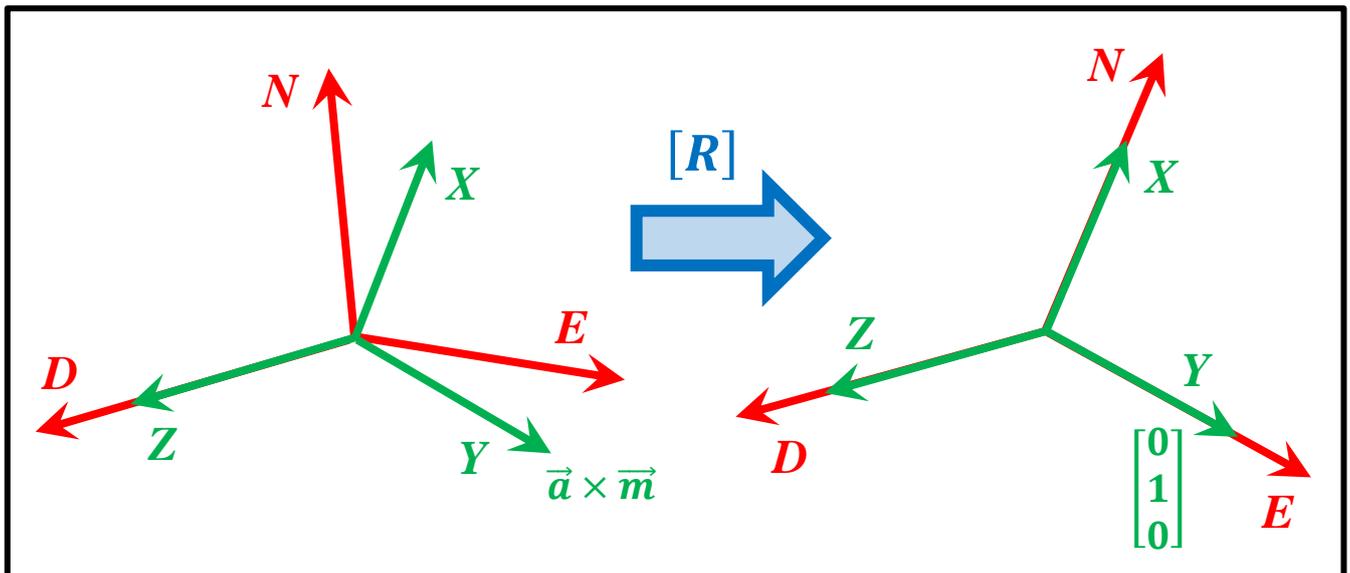


Figura 13.4.- Representación esquemática de la rotación necesaria para alinear el sistema de referencia $\{N, E, D\}$ con el *frame* cartesiano $\{X, Y, Z\}$; intrínseco al sensor. [Elaboración propia].

A partir de la figura anterior, se llega a la conclusión de que la coordenada y (expresada en el sistema de referencia del sensor) verifica la siguiente relación:

³³ **Nota:** Téngase presente que el operador “ \times ”, empleado en la ecuación (13.5), simboliza el producto vectorial entre las componentes D y N del sistema de referencia NED .

$$y = \vec{a} \times \vec{m} \quad (13.6)$$

La matriz de rotación $[R]$, indicada en la Figura 13.4, posibilita reorientar el *frame* $\{N, E, D\}$; de modo que sea coincidente con el sistema de referencia del sensor ($\{X, Y, Z\}$). Para ello, se efectúa el siguiente producto matricial:

$$\begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (13.7)$$

De acuerdo con la anatomía de la matriz de rotación ($[R] = [r_{ij}]$; $\forall i, j \in \{1, 2, 3\}$), se deduce que el producto vectorial entre las magnitudes \vec{a} y \vec{m} equivale a la segunda columna de la matriz $[R]$. Este hecho puede expresarse analíticamente como:

$$\begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (13.8)$$

Recordando que la matriz de rotación $[R]$ es ortogonal, dado que pertenece al *Grupo Rotacional* $SO(3)$, la primera columna de $[R]$ puede calcularse como el producto vectorial entre las dos restantes. Por tanto, resulta correcto implementar la siguiente operación algebraica:

$$\begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} = \begin{bmatrix} r_{21} \\ r_{22} \\ r_{23} \end{bmatrix} \times \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \end{bmatrix} = \left(\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \right) \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = (\vec{a} \times \vec{m}) \times \vec{a} \quad (13.9)$$

13.4.- NORMALIZACIÓN DE LA MATRIZ DE ROTACIÓN $[R]$

La etapa final del algoritmo *ecompass* se corresponde con un proceso de normalización, aplicado sobre la matriz de rotación $[R]$. Dicha normalización se efectúa sobre cada columna de la matriz $[r_{ij}]$, de acuerdo con lo recogido en la ecuación (13.10):

$$r_{ij} = \frac{r_{ij}}{\sqrt{\sum_{i=1}^3 r_{ij}^2}}, \forall j \quad (13.10)$$

Nota: Téngase en cuenta que el símbolo j hace referencia a cada una de las columnas que conforman la matriz de rotación $[R]$. Por otro lado, la notación i designa a las filas de dicha matriz.

14. Anexo D: Modelización de la orientación espacial mediante cuaterniones

A través de este anexo, se exponen las bases teóricas que fundamentan la interpretación física de los cuaterniones³⁴.

Si bien es cierto que las operaciones matemáticas con este tipo de objetos computacionales ya se encuentran implementadas en la mayoría de *softwares* comerciales de cálculo, se ha considerado interesante ofrecer un panorámica general acerca de la utilidad de los cuaterniones en la obtención de rotaciones y orientaciones 3D.

14.1.- ¿QUÉ SON LOS CUATERNIONES?

Los cuaterniones son estructuras matemáticas que pueden interpretarse como *números complejos extendidos a cuatro dimensiones*, tal y como se expresa en la siguiente notación³⁵:

$$\bar{q} = q_0 + q_1\vec{i} + q_2\vec{j} + q_3\vec{k} \quad (14.1)$$

³⁴ **Nota:** Durante elaboración de este anexo de la memoria, se ha tomado como bibliografía fundamental la literatura referenciada en [58].

³⁵ Como es habitual en la configuración vectorial cartesiana, los vectores unitarios \vec{i} , \vec{j} y \vec{k} verifican las siguientes propiedades:

$$\vec{i}^2 = \vec{j}^2 = \vec{k}^2 = \vec{i} \cdot \vec{j} \cdot \vec{k} = -1$$

$$\vec{i} \times \vec{j} = \vec{k}$$

$$\vec{j} \times \vec{k} = \vec{i}$$

$$\vec{k} \times \vec{i} = \vec{j}$$

14.1.1.- Principales propiedades asociadas a los cuaterniones

Los cuaterniones, al igual que la mayoría de objetos matemáticos definidos, cumplen una serie de propiedades; las cuales se enuncian a continuación:

- (1) Todo cuaternión \bar{q} posee un conjugado que puede expresarse como:

$$\bar{q}^* = q_0 - q_1\vec{i} - q_2\vec{j} - q_3\vec{k} \quad (14.2)$$

- (2) El producto escalar entre $\lambda \in \mathbb{R}$ y un cuaternión $\bar{q} \in \mathbb{R}^4$ puede calcularse de acuerdo con la siguiente expresión:

$$\lambda \cdot \bar{q} = \lambda q_0 + \lambda q_1\vec{i} + \lambda q_2\vec{j} + \lambda q_3\vec{k} \quad (14.3)$$

- (3) La suma de dos cuaterniones se obtiene sin más que operar componente a componente; tal y como puede observarse a continuación:

$$\bar{p} + \bar{q} = (p_0 + q_0) + (p_1 + q_1)\vec{i} + (p_2 + q_2)\vec{j} + (p_3 + q_3)\vec{k} \quad (14.4)$$

14.2.- ¿POR QUÉ SE UTILIZAN LOS CUATERNIONES PARA EXPRESAR LA ORIENTACIÓN TRIDIMENSIONAL (3D)?

Los cuaterniones ofrecen una solución robusta al problema de modelizar matemáticamente la orientación espacial, vinculada a un sólido rígido. En este escenario, pierde importancia la sensibilidad numérica ineludiblemente presente en las coordenadas exponenciales de orientación tridimensional.

Recuérdese que una matriz de rotación $[R] \in SO(3)$, codificada en coordenadas exponenciales, depende de la función inversa de $\sin(\theta)$. En consecuencia, se verifica la siguiente relación de proporcionalidad:

$$[R] \propto \sin^{-1}(\theta) \quad (14.5)$$

En consonancia con lo expresado en la ecuación (14.5), cualquier pequeña variación en el parámetro θ ocasionará modificaciones significativas en la matriz de rotación $[R]$.

Obsérvese que el peaje a pagar por la implementación de los cuaterniones, durante la modelización de la orientación 3D, es la incorporación de un cuarto parámetro en su expresión matemática.

14.3.- FORMULACIÓN MATEMÁTICA DE UN CUATERNIÓN

Sea $[R] \in SO(3)$ una matriz de rotación expresada en coordenadas exponenciales mediante la siguiente ecuación:

$$[R] = e^{[\omega]\theta} \quad (14.6)$$

Donde:

- $[\omega]$ se corresponde con el eje definido para la rotación 3D del sólido rígido considerado, verificándose que $\|\omega\| = 1$.
- θ denota al ángulo de rotación, expresado en radianes, que se ha desarrollado alrededor del eje $[\omega]$.

De acuerdo con la información recogida anteriormente, puede definirse un cuaternión unitario ($[\varepsilon] \in \mathbb{R}^4$) de la manera que sigue:

$$[\varepsilon] = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ [\omega] \cdot \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \in \mathbb{R}^4 \quad (14.7)$$

El vector $[\varepsilon] = [\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3]^T$ está conformado por los denominados *Parámetros de Euler* y satisface la condición de ser unitario, puesto que:

$$\|\varepsilon\| = \sqrt{\varepsilon_0^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2} = 1 \quad (14.8)$$

14.3.1.- Producto de dos cuaterniones

La operación más habitual en el cálculo de rotaciones tridimensionales se corresponde con la multiplicación encadenada de matrices pertenecientes al *Grupo Ortogonal Especial* o, en su defecto, el producto de cuaterniones de orientación.

Dadas dos matrices de rotación $[R_q], [R_p] \in SO(3)$ resulta posible establecer una equivalencia geométrica a través de los cuaterniones $[q] = [q_0, q_1, q_2, q_3]^T$ y $[p] = [p_0, p_1, p_2, p_3]^T$. De este modo, se tiene que el cuaternión producto $[\varepsilon] = [q][p]$ se obtiene mediante el siguiente cálculo matricial:

$$[\varepsilon] = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + p_0 q_1 + q_2 p_3 - q_3 p_2 \\ q_0 p_2 + p_0 q_2 - q_1 p_3 + q_3 p_1 \\ q_0 p_3 + p_0 q_3 + q_1 p_2 - q_2 p_1 \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (14.9)$$

14.4.- CÁLCULO DE ROTACIONES MEDIANTE CUATERNIONES. TEOREMA

Además de ser utilizados en la modelización de la orientación espacial, los cuaterniones constituyen una potente herramienta matemática para el cómputo de las rotaciones efectuadas por un sólido rígido. En relación a esta idea, se ha formulado el

siguiente Teorema; el cual permite observar la relación existente entre una matriz de rotación y el producto equivalente de cuaterniones.

Teorema.- La ecuación $r' = qrq^*$ es matemáticamente equivalente a $r' = Rr$, donde $R \in SO(3)$.

En el enunciado anterior, los cuaterniones r' , r y q han sido obtenidos de la siguiente forma:

$$\begin{aligned} r' &= 0 + \bar{r}' \\ r &= 0 + \bar{r} \\ q &= q_0 + \bar{q} \\ q^* &= q_0 - \bar{q} \end{aligned} \tag{14.10}$$

14.4.1.- Demostración del teorema anterior

A modo de curiosidad, se incluye en este subapartado la demostración matemática del teorema enunciado durante la sección anterior.

En primer lugar, ha de desarrollarse el producto de los tres cuaterniones inicialmente propuesto ($r' = qrq^*$). De esta forma, se obtiene la expresión indicada a continuación:

$$\begin{aligned} r' &= qrq^* \Rightarrow \\ \Rightarrow 0 + \bar{r}' &= (q_0 + \bar{q})(0 + \bar{r})(q_0 - \bar{q}) = \\ &= (q_0\bar{r} + \bar{q} \times \bar{r} - \bar{q} \cdot \bar{r})(q_0 - \bar{q}) = \\ &= [-\bar{q} \cdot \bar{r} + (q_0\bar{r} + \bar{q} \times \bar{r})](q_0 - \bar{q}) \end{aligned} \tag{14.11}$$

Aplicando nuevamente la expresión correspondiente al producto de cuaterniones, resulta:

$$\bar{r}' = -q_0(\bar{q} \cdot \bar{r}) + [q_0\bar{r} + \bar{q} \times \bar{r}] \cdot \bar{q} + (\bar{q} \cdot \bar{r})\bar{q} + \tag{14.12}$$

$$\begin{aligned}
 &+q_0(q_0\bar{r} + \bar{q} \times \bar{r}) - (q_0\bar{r} + \bar{q} \times \bar{r}) \times \bar{q} = \\
 &= -q_0(\bar{q} \cdot \bar{r}) + q_0\bar{r} \cdot \bar{q} + (\bar{q} \times \bar{r}) \cdot \bar{q} + (\bar{q} \cdot \bar{r})\bar{q} + q_0^2\bar{r} + \\
 &+q_0(\bar{q} \times \bar{r}) - q_0(\bar{r} \times \bar{q}) - \bar{q} \times \bar{r} \times \bar{q}
 \end{aligned}$$

Reordenando la ecuación anterior, se tiene que:

$$\begin{aligned}
 \bar{r}' &= (\bar{q} \times \bar{r}) \cdot \bar{q} + (\bar{q} \cdot \bar{r}) \cdot \bar{q} + q_0^2\bar{r} - 2q_0(\bar{r} \times \bar{q}) - \bar{q} \times \bar{r} \times \bar{q} = \\
 &= (\bar{q} \times \bar{r}) \cdot \bar{q} + (\bar{q} \cdot \bar{r}) \cdot \bar{q} + q_0^2\bar{r} - \\
 &-2q_0(\bar{r} \times \bar{q}) - (\bar{q} \cdot \bar{q}) \cdot \bar{r} + (\bar{r} \cdot \bar{q}) \cdot \bar{q} = \\
 &= q_0^2\bar{r} - (\bar{q} \cdot \bar{q}) \cdot \bar{r} - 2q_0(\bar{r} \times \bar{q}) + 2\bar{q} \cdot (\bar{q} \cdot \bar{r})
 \end{aligned} \tag{14.13}$$

Alternativamente, el cuaternión \bar{r}' también puede expresarse como³⁶:

$$\bar{r}' = q_0^2\bar{r} - (\bar{q} \cdot \bar{q}) \cdot \bar{r} + 2q_0(\bar{q} \times \bar{r}) + 2\bar{q} \cdot (\bar{q} \cdot \bar{r}) \tag{14.14}$$

El siguiente paso de la demostración del teorema consiste en incorporar la notación recogida en las dos ecuaciones siguientes, propia del cálculo vectorial:

$$\bar{q} \times \bar{r} = [q]\bar{r} \tag{14.15}$$

$$(\bar{q} \cdot \bar{r}) \cdot \bar{q} = (\bar{q} \cdot \bar{q}^T) \cdot \bar{r} \tag{14.16}$$

De este modo, el cuaternión \bar{r}' adopta la siguiente anatomía matricial:

$$\bar{r}' = \left[\begin{array}{c} (q_0^2 - q_1^2 - q_2^2 - q_3^2)[I] + \\ +2q_0 \left[\begin{array}{ccc} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{array} \right] + \left[\begin{array}{ccc} q_1^2 & q_1q_2 & q_1q_3 \\ q_1q_2 & q_2^2 & q_2q_3 \\ q_1q_3 & q_2q_3 & q_3^2 \end{array} \right] \end{array} \right] \cdot \bar{r} = \tag{14.17}$$

³⁶ Cabe resaltar que durante la obtención de la ecuación (14.14) se ha hecho uso de la siguiente relación geométrica, relativa al producto vectorial de dos vectores tridimensionales:

$$\bar{p} \times \bar{q} \times \bar{r} = \bar{q} \cdot (\bar{p} \cdot \bar{r}) - \bar{r} \cdot (\bar{p} \cdot \bar{q})$$

$$= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \cdot \bar{\mathbf{r}}$$

Seguidamente, se procede a deducir la *Fórmula de Rodrigues*³⁷ mediante la notación matricial (14.17) y la definición inicialmente enunciada para un cuaternión de orientación genérico, la cual se recuerda a continuación:

$$[\varepsilon] = \varepsilon_0 + \bar{\varepsilon} = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) \cdot [\omega] \quad (14.18)$$

Introduciendo el cuaternión $[\varepsilon]$, enunciado en la ecuación (14.18), dentro de la expresión matricial (14.17) se obtiene el siguiente desarrollo³⁸:

$$\begin{aligned} \bar{\mathbf{r}}' &= q_0^2 \bar{\mathbf{r}} - (\bar{\mathbf{q}} \cdot \bar{\mathbf{q}}) \cdot \bar{\mathbf{r}} + 2q_0(\bar{\mathbf{q}} \times \bar{\mathbf{r}}) + 2\bar{\mathbf{q}} \cdot (\bar{\mathbf{q}} \cdot \bar{\mathbf{r}}) = \\ &= q_0^2 \bar{\mathbf{r}} - (\bar{\mathbf{q}} \cdot \bar{\mathbf{q}}) \cdot \bar{\mathbf{r}} + 2q_0[\bar{\mathbf{q}}]\bar{\mathbf{r}} + 2(\bar{\mathbf{q}} \cdot \bar{\mathbf{q}}^T) \cdot \bar{\mathbf{r}} = \end{aligned} \quad (14.19)$$

³⁷ **Nota:** Durante el proceso de obtención de la *Fórmula de Rodrigues*, deben emplearse varias relaciones trigonométricas vinculadas al *ángulo mitad*; las cuales se enuncian a continuación:

$$2 \cdot c\left(\frac{\theta}{2}\right) \cdot s\left(\frac{\theta}{2}\right) = s(\theta)$$

$$c^2\left(\frac{\theta}{2}\right) - s^2\left(\frac{\theta}{2}\right) = c(\theta)$$

$$s^2\left(\frac{\theta}{2}\right) + c^2\left(\frac{\theta}{2}\right) = 1$$

$$2 \cdot s^2\left(\frac{\theta}{2}\right) = 1 - c(\theta)$$

³⁸ Téngase en cuenta que el cuaternión $[\bar{\mathbf{q}}]$, expresado mediante la notación matricial *Eje – Ángulo*, responde a la siguiente anatomía matemática:

$$[\bar{\mathbf{q}}] = \begin{bmatrix} 0 & -\omega_3 s\left(\frac{\theta}{2}\right) & \omega_2 s\left(\frac{\theta}{2}\right) \\ 0 & 0 & -\omega_1 s\left(\frac{\theta}{2}\right) \\ 0 & 0 & 0 \end{bmatrix} = s\left(\frac{\theta}{2}\right) \cdot [\omega]$$

$$= \left[\begin{array}{l} \cos^2\left(\frac{\theta}{2}\right) - \left(\omega_1^2 \cdot \sin^2\left(\frac{\theta}{2}\right) + \omega_2^2 \cdot \sin^2\left(\frac{\theta}{2}\right) + \omega_3^2 \cdot \sin^2\left(\frac{\theta}{2}\right)\right) + \\ + 2 \cos\left(\frac{\theta}{2}\right) \cdot \sin\left(\frac{\theta}{2}\right) \cdot [\omega] + 2 \sin^2\left(\frac{\theta}{2}\right) \cdot ([\omega][\omega]^T) \end{array} \right] \cdot \bar{r}$$

El resultado de \bar{r}^T , mostrado anteriormente, puede reformularse tal y como se indica a continuación³⁹:

$$\bar{r}^T = [\cos(\theta) + \sin(\theta) \cdot [\omega] + (1 - \cos(\theta)) \cdot ([\omega][\omega]^T)] \cdot \bar{r} \quad (14.20)$$

La ecuación recogida anteriormente en (14.20) ya se corresponde con la *Fórmula de Rodrigues*, de modo que la demostración matemática puede darse por concluida.

A modo de ejemplo, se ha decidido comprobar el correcto funcionamiento de la ecuación final deducida. Para abordar dicha tarea, se ha definido una rotación θ alrededor del eje $[\omega] = [z]$; la cual se corresponde con el cuaternión $\varepsilon = \left[c\left(\frac{\theta}{2}\right), 0, 0, s\left(\frac{\theta}{2}\right) \right]^T$. Aplicando el resultado algebraico demostrado con anterioridad, puede calcularse⁴⁰ la siguiente matriz de rotación $[R] \in SO(3)$:

$$[R]_{3 \times 3} = \begin{bmatrix} c^2\left(\frac{\theta}{2}\right) - s^2\left(\frac{\theta}{2}\right) & -2 \cdot s\left(\frac{\theta}{2}\right) \cdot c\left(\frac{\theta}{2}\right) & 0 \\ 2 \cdot s\left(\frac{\theta}{2}\right) \cdot c\left(\frac{\theta}{2}\right) & c^2\left(\frac{\theta}{2}\right) - s^2\left(\frac{\theta}{2}\right) & 0 \\ 0 & 0 & s^2\left(\frac{\theta}{2}\right) + c^2\left(\frac{\theta}{2}\right) \end{bmatrix}_{3 \times 3} = \quad (14.21)$$

³⁹ Durante la deducción de la fórmula (14.20), se ha empleado la expresión indicada a continuación; relativa al producto de un cuaternión por su representación matricial traspuesta:

$$\bar{q} \cdot \bar{q}^T = s\left(\frac{\theta}{2}\right) \cdot [\omega] \cdot s\left(\frac{\theta}{2}\right) \cdot [\omega]^T = s^2\left(\frac{\theta}{2}\right) \cdot ([\omega][\omega]^T) = \frac{1}{2}(1 - c(\theta))$$

⁴⁰ Durante los cálculos trigonométricos expuestos en este apartado, se ha empleado una notación abreviada para referenciar a las funciones *seno* y *coseno*:

$$s(\theta) \equiv \sin(\theta), \forall \theta \in [0, 2\pi] [rad]$$

$$c(\theta) \equiv \cos(\theta), \forall \theta \in [0, 2\pi] [rad]$$

$$= \begin{bmatrix} \mathbf{c}(\theta) & -\mathbf{s}(\theta) & \mathbf{0} \\ \mathbf{s}(\theta) & \mathbf{c}(\theta) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}_{3 \times 3}$$

15. Anexo E: Código Fuente desarrollado durante la fase previa a la calibración metrológica de los sensores inerciales

Como etapa anterior al análisis de los resultados mostrados en la sección "5.1.1.- Estudio sobre el estado de los sensores considerados, como paso previo a la calibración de dichos dispositivos", se ha desarrollado una porción de código fuente; la cual se recoge en el presente anexo. Téngase en cuenta que el lenguaje de programación empleado durante dicho desarrollo *software* ha sido *MATLAB*.

[LAB]: Comprobación del estado inicial de calibración asociado a los sensores inerciales

Índice de Contenidos

15.1.- *IMU Xsens DOT7*

- 15.1.1.- Valores máximos de las señales de aceleración
- 15.1.2.- Valores medios de las señales de aceleración
- 15.1.3.- Representación gráfica de las señales de aceleración

15.2.- *IMU MATA1002195*

- 15.2.1.- Valores máximos de las señales de aceleración
- 15.2.2.- Valores medios de las señales de aceleración
- 15.2.3.- Representación gráfica de las señales de aceleración

15.1.- IMU XSENS DOT7

```
% Limpieza de la ventana de comandos. Liberación de la memoria
inicialmente ocupada en el Workspace de MatLab
clc, clear

% Lectura de los datos capturados mediante la unidad de medida inercial,
perteneiente a la gama Xsens DOT.
% Llamada a la función readtable.
Datos_XsensDOT7 = readtable('./Medidas_Xsens_DOT7.csv')

% Componentes cartesianas (gx, gy, gz) de la aceleración de la gravedad
gx = Datos_XsensDOT7.Acc_X;
gy = Datos_XsensDOT7.Acc_Y;
gz = Datos_XsensDOT7.Acc_Z;
```

15.1.1.- Valores máximos de las señales de aceleración

```
% Valor máximo de la señal gx
ValorMaximo_gx_XsensDOT = max(gx)

% Valor máximo de la señal gy
ValorMaximo_gy_XsensDOT = max(gy)

% Valor máximo de la señal gz
ValorMaximo_gz_XsensDOT = max(gz)
```

15.1.2.- Valores medios de las señales de aceleración

```
% Valor medio de la señal gx
ValorMedio_gx_XsensDOT = mean(gx)

% Valor medio de la señal gy
ValorMedio_gy_XsensDOT = mean(gy)

% Valor medio de la señal gz
ValorMedio_gz_XsensDOT = mean(gz)
```

15.1.3.- Representación gráfica de las señales de aceleración

```
% Creación de una nueva figura
figure
```

```
% Activación de la primera casilla dentro de la matriz de
% figuras creada (dimensiones 2 x 2)
subplot(2,2,1)

% Representación gráfica de la componente gx, asociada a
% la aceleración de la gravedad. Se emplea un ancho de línea
% de 1.5 [puntos]:
plot(gx, 'LineWidth',1.5)

% Incorporación de una etiqueta al eje X de la figura
xlabel('Número de muestra [-]')

% Incorporación de una etiqueta al eje Y de la figura
ylabel('$g_x [m/s^2]$', 'Interpreter', 'latex', 'FontSize',12)

% Creación de un título para dicha figura
title('Representación gráfica de g_x')

% Definición de los límites vinculados a los ejes de la primera
% representación gráfica maquetada, con objeto de mejorar la
% visualización de los resultados obtenidos:
xlim([4052.4 4088.3])
ylim([-0.055 0.006])

% Activación de la rejilla existente en la figura creada con anterioridad
grid on

% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gx, capturada mediante el dispositivo Xsens DOT7. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gx_XsensDOT, 'r--', 'LineWidth',1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_x', 'mean(g_x)')

%-----

% Activación de la segunda casilla dentro de la matriz de
% figuras creada (dimensiones 2 x 2)
subplot(2,2,2)

% Representación gráfica de la componente gy, asociada a
% la aceleración de la gravedad
plot(gy, 'LineWidth',1.5)

% Incorporación de una etiqueta al eje X de la figura
xlabel('Número de muestra [-]')
```

```
% Incorporación de una etiqueta al eje Y de la figura
ylabel('$g_y [m/s^2]$', 'Interpreter', 'latex', 'FontSize', 12)

% Creación de un título para dicha figura
title('Representación gráfica de g_y')

% Definición de los límites vinculados a los ejes de la segunda
% representación gráfica maquetada, con objeto de mejorar la
% visualización de los resultados obtenidos:
xlim([3700.8 3753.1])
ylim([0.026 0.090])

% Activación de la rejilla existente en la figura creada con anterioridad
grid on

% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gy, capturada mediante el dispositivo Xsens DOT7. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gy_XsensDOT, 'r--', 'LineWidth', 1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_y', 'mean(g_y)')

% Activación de la tercera y cuarta casilla dentro de la matriz de
% figuras creada (dimensiones 2 x 2)
subplot(2,2,[3,4])

% Representación gráfica de la componente gz, asociada a
% la aceleración de la gravedad
plot(gz, 'LineWidth', 1.5)

% Incorporación de una etiqueta al eje X de la figura
xlabel('Número de muestra [-]')

% Incorporación de una etiqueta al eje Y de la figura
ylabel('$g_z [m/s^2]$', 'Interpreter', 'latex', 'FontSize', 12)

% Creación de un título para dicha figura
title('Representación gráfica de g_z')

% Definición de los límites vinculados a los ejes de la tercera
% representación gráfica maquetada, con objeto de mejorar la
% visualización de los resultados obtenidos:
xlim([2964.5 3012.7])
ylim([9.552 9.686])

% Activación de la rejilla existente en la figura creada con anterioridad
```

```
grid on

% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gz, capturada mediante el dispositivo Xsens DOT7. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gz_XsensDOT, 'r--', 'LineWidth', 1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_z', 'mean(g_z)')

% Inclusión de un título global a toda la figura creada mediante las
líneas
% de código anteriores:
sgtitle({'Aceleración de la gravedad registrada', 'mediante el IMU Xsens
DOT7'})
```

15.2.- IMU MATA1002195

```
% Lectura de los datos capturados mediante la unidad de medida inercial,
% perteneciente a la familia MATRIX. Llamada a la función readtable.
Datos_IMU_MATA1002195 = readtable('./MedidasIMU_MATRIX_MATA1002195.csv')

% Definición del valor teórico de g, el cual se utilizará posteriormente
% para reescalar los valores suministrados por el IMU empleado durante la
% metodología experimental:
g = 9.81;

% Reescalado de gx a unidades de [m/s^2]. Inicialmente, dicha señal estaba
% expresada en tanto por uno de g.
gx = Datos_IMU_MATA1002195.acc_x * g;

% Reescalado de gy a unidades de [m/s^2]. Inicialmente, dicha señal estaba
% expresada en tanto por uno de g.
gy = Datos_IMU_MATA1002195.acc_y * g;

% Reescalado de gz a unidades de [m/s^2]. Inicialmente, dicha señal estaba
% expresada en tanto por uno de g.
gz = Datos_IMU_MATA1002195.acc_z * g;
```

15.2.1.- Valores máximos de las señales de aceleración

```
% Obtención del valor máximo ligado a la señal gx
ValorMaximo_gx = max(abs(gx))

% Obtención del valor máximo ligado a la señal gy
```

```
ValorMaximo_gy = max(abs(gy))  
  
% Obtención del valor máximo ligado a la señal gz  
ValorMaximo_gz = max(abs(gz))
```

15.2.2.- Valores medios de las señales de aceleración

```
% Cálculo del valor medio inherente a la señal gx  
ValorMedio_gx = mean(gx)  
  
% Cálculo del valor medio inherente a la señal gy  
ValorMedio_gy = mean(gy)  
  
% Cálculo del valor medio inherente a la señal gz  
ValorMedio_gz = mean(gz)
```

15.2.3.- Representación gráfica de las señales de aceleración

```
% Creación de una nueva figura  
figure  
  
% Activación de la primera casilla dentro de la matriz de  
% figuras creada (dimensiones 2 x 2)  
subplot(2,2,1)  
  
% Representación gráfica de la componente gx, asociada a  
% la aceleración de la gravedad. Se emplea un ancho de línea  
% de 1.5 [puntos]:  
plot(gx, 'LineWidth',1.5)  
  
% Incorporación de una etiqueta al eje X de la figura  
xlabel('Número de muestra [-]')  
  
% Incorporación de una etiqueta al eje Y de la figura  
ylabel('$g_x [m/s^2]$', 'Interpreter', 'latex', 'FontSize', 12)  
  
% Creación de un título para dicha figura  
title('Representación gráfica de g_x')  
  
% Definición de los límites intrínsecos al eje X de la figura  
% maquetada con anterioridad  
xlim([1315.7 1351.4])  
  
% Activación de la rejilla existente en la figura creada  
grid on
```

```
% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gx, capturada mediante el dispositivo MATA1002195. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gx,'r--','LineWidth',1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_x', 'mean(g_x)')

% Identificación de la posición ligada a la leyenda creada mediante la
% instrucción anterior:
legend("Position", [0.27902,0.53498,0.17679,0.082143])

% -----

% Activación de la segunda casilla dentro de la matriz de
% figuras creada (dimensiones 2 x 2)
subplot(2,2,2)

% Representación gráfica de la componente gy, asociada a
% la aceleración de la gravedad. Se emplea un ancho de línea
% de 1.5 [puntos]:
plot(gy, 'LineWidth',1.5)

% Incorporación de una etiqueta al eje X de la figura
xlabel('Número de muestra [-]')

% Incorporación de una etiqueta al eje Y de la figura
ylabel('$g_y$ [m/s^2]','$','Interpreter','latex','FontSize',12)

% Creación de un título para dicha figura
title('Representación gráfica de g_y')

% Definición de los límites intrínsecos al eje X de la figura
% maquetada con anterioridad
xlim([994.3 1030])

% Activación de la rejilla existente en la figura creada
grid on

% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gy, capturada mediante el dispositivo MATA1002195. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gy,'r--','LineWidth',1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_y', 'mean(g_y)')

% Identificación de la posición ligada a la leyenda creada mediante la
```

```
% instrucción anterior:
legend("Position", [0.59577,0.53152,0.17679,0.082143])

% Activación de la tercera y cuarta casilla dentro de la matriz de
% figuras creada (dimensiones 2 x 2)
subplot(2,2,[3,4])

% Representación gráfica de la componente gz, asociada a
% la aceleración de la gravedad
plot(gz, 'LineWidth',1.5)

% Incorporación de una etiqueta al eje X de la figura
xlabel('Número de muestra [-]')

% Incorporación de una etiqueta al eje Y de la figura
ylabel('$g_z$ [m/s^2]$', 'Interpreter', 'latex', 'FontSize',12)
% Creación de un título para dicha figura
title('Representación gráfica de g_z')

% Definición de los límites intrínsecos al eje X de la figura
% maquetada con anterioridad
xlim([812.5 864.3])

% Activación de la rejilla existente en la figura creada
grid on

% Representación gráfica (línea horizontal) del valor medio inherente a la
% señal gz, capturada mediante el dispositivo MATA1002195. Se utiliza una
% línea de color rojo y trazo discontinuo, cuyo ancho es de 1.5 [puntos]:
yline(ValorMedio_gz, 'r--', 'LineWidth',1.5)

% Incorporación de una leyenda en la figura anterior
legend('g_z', 'mean(g_z)')

% Identificación de la posición correspondiente a la leyenda creada
% mediante la instrucción anterior:
legend("Position", [0.68568,0.11998,0.16786,0.072619])

% Inclusión de un título global a toda la figura creada mediante las
líneas de código anteriores:
sgtitle({'Aceleración de la gravedad registrada', 'mediante el IMU
MATA1002195'})
```

16. *Anexo F: Código Fuente* construido para comprobar el estado de ortogonalidad intrínseco al *frame* cartesiano de cada *IMU*

Como etapa previa a la calibración anatómica, se ha verificado el nivel de ortogonalidad presente en los sistemas de referencia vinculados a los respectivos *IMUs*.

[LAB]: Calibración del *IMU Xsens DOT 8*. Resolución del modelo de calibración mediante *Mínimos Cuadrados*

Índice de Contenidos

16.1.- Medida Experimental

16.1.1.- Importación de datos

16.1.2.- Representación gráfica de los datos de aceleración (Acelerómetro del *IMU*)

16.2.- Resolución del sistema de ecuaciones lineales relativo al modelo de calibración. *Método de Mínimos Cuadrados*

16.2.1.- Vector $[y]$ de variables dependientes

16.2.2.- Matriz de coeficientes $[A]$ del sistema lineal de ecuaciones

16.2.3.- Vector de incógnitas $[\theta]$. Obtención de los parámetros de calibración a través de

Mínimos Cuadrados

16.3.- Funciones

16.3.1- Sistema de ecuaciones no lineales utilizado durante la calibración.
Calibracion(x)

```
% Liberación de la memoria inicialmente ocupada en el Workspace de MatLab
clear

% Limpieza de la ventana de comandos
clc
```

16.1.- MEDIDA EXPERIMENTAL

16.1.1.- Importación de datos

```
% A continuación, se realiza la importación de los datos capturados mediante
el IMU Xsens DOT8. Para ello, se invoca a la
% función importfile_XsensDOT8MedicionCada15grados; previamente diseñada.
Datos_IMU = readtable("Datos_XsensDOT7.csv")
```

Tabla 16.1.- Fragmento de los datos capturados mediante el sensor inercial. Recuérdese que se persigue cuantificar el nivel de ortogonalidad asociado al sistema de referencia interno, existente en dicho dispositivo.

Datos_IMU = 5674x12 table

	PacketCounter	SampleTimeFine	Quat_W	Quat_X	Quat_Y	...
1	1	1.2968e+09	0.3089	-0.3044	-0.6504	
2	2	1.2968e+09	0.3090	-0.3044	-0.6501	
3	3	1.2968e+09	0.3090	-0.3043	-0.6500	
4	4	1.2968e+09	0.3091	-0.3041	-0.6500	
5	5	1.2968e+09	0.3091	-0.3040	-0.6500	
6	6	1.2968e+09	0.3091	-0.3040	-0.6499	
7	7	1.2969e+09	0.3090	-0.3039	-0.6499	
:						

```
% Datos correspondientes a la aceleración (de la gravedad) registrada en el
eje X del sensor inercial
Acc_X_m_s2 = Datos_IMU.Acc_X;

% Indexado del vector Acc_X_m_s2. Se seleccionan las muestras 1 a 3500.
Acc_X_m_s2 = Acc_X_m_s2(1:3500);

% Datos correspondientes a la aceleración (de la gravedad) registrada en el
eje Y del sensor inercial
Acc_Y_m_s2 = Datos_IMU.Acc_Y;

% Indexado del vector Acc_Y_m_s2. Se seleccionan las muestras 1 a 3500.
Acc_Y_m_s2 = Acc_Y_m_s2(1:3500);

% Datos correspondientes a la aceleración (de la gravedad) registrada en el
eje Z del sensor inercial
Acc_Z_m_s2 = Datos_IMU.Acc_Z;

% Indexado del vector Acc_Z_m_s2. Se seleccionan las muestras 1 a 3500.
Acc_Z_m_s2 = Acc_Z_m_s2(1:3500);
```

16.1.2.- Representación gráfica de los datos de aceleración (Acelerómetro del *IMU*)

```
% Creación de una nueva figura
figure

% Activación de la primera casilla dentro de la figura generada
subplot(2,2,1)

% Representación gráfica de la aceleración de la gravedad capturada en la
dirección del eje X
plot(Acc_X_m_s2, 'r-', 'LineWidth', 2)

% Activación de la rejilla en la figura maquetada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Número de muestra [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel('$g_x [m/s^2]$', 'interpreter', 'latex', 'FontSize', 13)

% Incorporación del título a la representación gráfica
title('Aceleración gravedad (EJE X)')
```

```
% Especificación de valores límite en el eje X, con objeto de ajustar la
representación efectuada
% xlim([1440, length(Acc_X_m_s2)])

% Extracción de una muestra de datos, asociados a gx
% Datos_gx_representados = Acc_X_m_s2(1440:end);
Datos_gx_representados = Acc_X_m_s2;

% Cálculo del valor máximo de gx
Maximo_gx = max(Datos_gx_representados);

% Cálculo del valor mínimo de gx
Minimo_gx = min(Datos_gx_representados);

% Con esta instrucción, se mantiene activa la figura actual
hold on

% Representación gráfica del valor máximo ligado a gx
yline(Maximo_gx, 'k--', 'LineWidth', 2)

% Representación gráfica del valor mínimo ligado a gx
yline(Minimo_gx, 'k--', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Activación de la segunda casilla dentro de la figura generada
subplot(2,2,2)

% Representación gráfica de la aceleración de la gravedad capturada en la
dirección del eje Y
plot(Acc_Y_m_s2, 'g-', 'LineWidth', 2)

% Activación de la rejilla en la figura maquetada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Número de muestra [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel('$g_y [m/s^2]$', 'interpreter', 'latex', 'FontSize', 13)

% Incorporación del título a la representación gráfica
title('Aceleración gravedad (EJE Y)')
```

```
% Especificación de valores límite en el eje X, con objeto de ajustar la
representación efectuada
% xlim([1440, length(Acc_Y_m_s2)])

% Extracción de una muestra de datos, asociados a gy
% Datos_gy_representados = Acc_Y_m_s2(1440:end);
Datos_gy_representados = Acc_Y_m_s2;

% Cálculo del valor máximo de gy
Maximo_gy = max(Datos_gy_representados);

% Cálculo del valor mínimo de gy
Minimo_gy = min(Datos_gy_representados);

% Se mantiene activa la figura actual
hold on

% Representación gráfica del valor máximo ligado a gy
yline(Maximo_gy, 'k--', 'LineWidth', 2)

% Representación gráfica del valor mínimo ligado a gy
yline(Minimo_gy, 'k--', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Activación de la tercera y cuarta casilla dentro de la figura generada
subplot(2,2,[3,4])

% Representación gráfica de la aceleración de la gravedad capturada en la
dirección del eje Z
plot(Acc_Z_m_s2, 'b-', 'LineWidth', 2)

% Activación de la rejilla en la figura maquetada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Número de muestra [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel('$g_z [m/s^2]$', 'interpreter', 'latex', 'FontSize', 13)

% Incorporación del título a la representación gráfica
title('Aceleración gravedad (EJE Z)')
```

```
% Especificación de valores límite en el eje X, con objeto de ajustar la
representación efectuada
% xlim([1440, length(Acc_Z_m_s2)])

% Extracción de una muestra de datos, asociados a gz
% Datos_gz_representados = Acc_Z_m_s2(1440:end);
Datos_gz_representados = Acc_Z_m_s2;

% Cálculo del valor máximo de gz
Maximo_gz = max(Datos_gz_representados);

% Cálculo del valor mínimo de gz
Minimo_gz = min(Datos_gz_representados);

% Se mantiene activa la figura actual
hold on

% Representación gráfica del valor máximo ligado a gz
yline(Maximo_gz, 'k--', 'LineWidth', 2)

% Representación gráfica del valor mínimo ligado a gz
yline(Minimo_gz, 'k--', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Creación de un título global para toda la figura
sgtitle({'Datos de aceleración registrados mediante', ...
        'el IMU Xsens DOT7. Proceso de ortogonalización'})
```

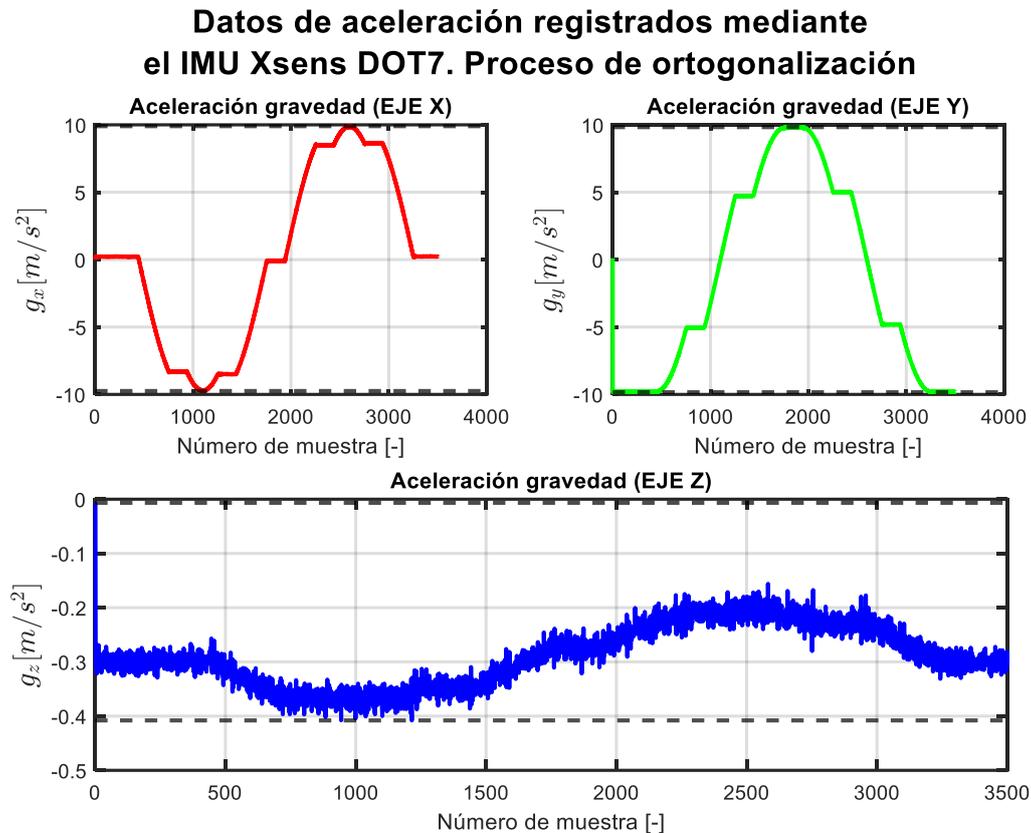


Figura 16.1.- Componentes cartesianas de la aceleración de la gravedad, capturadas durante el procedimiento de ortogonalización. Obsérvese que la información de interés se corresponde con las diferentes posiciones estáticas definidas (véanse las variables g_x y g_y).

16.2.- RESOLUCIÓN DEL SISTEMA DE ECUACIONES LINEALES RELATIVO AL MODELO DE CALIBRACIÓN. MÉTODO DE MÍNIMOS CUADRADOS

Modelo de calibración planteado:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{x_1} & a_{y_1} & a_{z_1} & 1 \\ a_{x_2} & a_{y_2} & a_{z_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x_m} & a_{y_m} & a_{z_m} & 1 \end{bmatrix} \begin{bmatrix} s \cdot u_x \\ s \cdot u_y \\ s \cdot u_z \\ o \end{bmatrix} \equiv [Y] = [A][\theta]$$

16.2.1.- Vector [y] de variables dependientes

```
% Creación del vector y. Inicialmente, no contiene ningún elemento
y = [];

% Inicialización de la posición angular asociada al sensor, la cual se irá
incrementando progresivamente durante
% el procedimiento de calibración
% theta = 0;

% Asignación del primer elemento que conforma el vector y (magnitudes
calibradas)
y(1) = Acc_Z_m_s2(1);

% Definición del incremento [rad] aplicado sobre la posición angular del IMU
Xsens DOT8
% incremento_theta = 60 * pi/180;

% Creación del vector fila theta, en el cual se almacenan los ángulos de
rotación definidos
theta = [0, 60, 120, 180, 240, 300] * pi/180;

% Variable empleada para indexar el vector theta
indice_theta = 1;

% Parámetro auxiliar utilizado para detectar cambios significativos en las
señales de aceleración registradas
epsilon = 0.3;

% Para i desde 2 hasta length(Acc_Z_m_s2) --> HACER:
for i = 2:length(Acc_Z_m_s2)
    % Si el valor absoluto de la diferencia (Acc_Z_m_s2(i) - Acc_Z_m_s2(i-
1)) es mayor que epsilon --> HACER:
    if abs(Acc_Z_m_s2(i) - Acc_Z_m_s2(i-1)) > epsilon
        % Se incrementa una unidad el contenido de la variable indice_theta
        indice_theta = indice_theta + 1;
        % Fin del bucle for i = 2:length(Acc_Z_m_s2)
    end
    % Obtención de la poyección vertical asociada a la aceleración de la
gravedad
    y(i) = Acc_Z_m_s2(i) * cos(theta(indice_theta));
end

% Para i desde 1 hasta length(Acc_Z_m_s2) --> HACER:
% for i=1:length(Acc_Z_m_s2)
%     Incremento del ángulo [rad] inherente al sensor inercial
%     theta = theta + incremento_theta;
```

```
% Incorporación del elemento que ocupa la posición "i" dentro del
vector [y]
%   y(i) = Acc_Z_m_s2(i) * cos(theta);
% Fin del bucle for i=1:length(Acc_Z_m_s2)
% end

% Mediante esta instrucción, se traspone la variable y; con la finalidad de
obtener un vector columna
y = y'
```

```
y = 3500x1
-0.0067
-0.3030
-0.3070
-0.3093
-0.2935
-0.3218
-0.3050
-0.2931
-0.3023
-0.2740
⋮
```

16.2.2.- Matriz de coeficientes [A] del sistema lineal de ecuaciones

```
% Contrucción de la matriz de coeficientes [A], utilizada durante el proceso
de calibración
A = [Acc_X_m_s2, Acc_Y_m_s2, Acc_Z_m_s2, ones(length(Acc_X_m_s2),1)]
```

```
A = 3500x4
-0.0015    0.1030   -0.0067    1.0000
 0.2308   -9.8124   -0.3030    1.0000
 0.2386   -9.8044   -0.3070    1.0000
 0.2432   -9.8133   -0.3093    1.0000
 0.2256   -9.8067   -0.2935    1.0000
 0.2367   -9.8077   -0.3218    1.0000
 0.2252   -9.8074   -0.3050    1.0000
 0.2209   -9.8047   -0.2931    1.0000
 0.2347   -9.8193   -0.3023    1.0000
 0.2356   -9.7833   -0.2740    1.0000
⋮
```

16.2.3.- Vector de incógnitas [θ]. Obtención de los parámetros de calibración a través de *Mínimos Cuadrados*

$$\theta_{\text{MÍNIMOS CUADRADOS}} = (A^T A)^{-1} A^T y$$

```
% Resolución algebraica del sistema lineal de ecuaciones
theta = pinv(A'*A) * A' * y
```

```
theta = 4x1
-0.0000
-0.0000
 1.0000
 0.0000
```

Hasta este punto, se tiene que $[\theta] = \begin{bmatrix} s \cdot u_x \\ s \cdot u_y \\ s \cdot u_z \\ o \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.

Ahora se debe resolver el siguiente sistema no lineal de 4 ecuaciones (teniendo presente que el vector asociado a la dirección del sensor es unitario):

$$\begin{bmatrix} s \cdot u_x \\ s \cdot u_y \\ s \cdot u_z \\ u_x^2 + u_y^2 + u_z^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Incógnitas: $[s, u_x, u_y, u_z]$.

```
% Invocación de la función Calibracion
fun = @Calibracion;

% Semilla inicial para resolver el sistema de ecuaciones no lineales
x0 = [10,0.5,0.5,0.5];

% Resolución numérica (procedimiento iterativo) del sistema no lineal de
ecuaciones planteado
x = fsolve(fun,x0)
```

Equation solved.

fsolve completed because the vector of function values is near zero as measured by the value of the function tolerance, and the problem appears regular as measured by the gradient.

<stopping criteria details>

```
x = 1x4
 1.0000 -0.0000 -0.0000 1.0000
```

SOLUCIÓN OBTENIDA DURANTE EL PROCESO DE CALIBRACIÓN:

$[s, u_x, u_y, u_z, o] = [1, 0, 0, 1, 0]$.

16.3.- FUNCIONES

16.3.1- Sistema de ecuaciones no lineales utilizado durante la calibración. Calibracion(x)

Función de calibración.

```
function F = Calibracion(x)
% Codificación de la primera ecuación homogénea presente en el sistema
F(1) = x(1)*x(2) - 0;
% Codificación de la segunda ecuación homogénea presente en el sistema
F(2) = x(1)*x(3) - 0;
% Codificación de la tercera ecuación homogénea presente en el sistema
F(3) = x(1)*x(4) - 1;
% Codificación de la cuarta ecuación homogénea presente en el sistema
F(4) = x(2)^2 + x(3)^2 + x(4)^2 - 1;

% Fin de la función Calibracion
end
```

17. Anexo G: Código Fuente desarrollado para corregir la desalineación existente en el par *sensor – segmento anatómico*

El presente *Anexo* se ha destinado a recoger el algoritmo implementado en *MatLab* para eliminar el error introducido por la *desalineación* del sensor inercial. En esta actividad, se ha calculado una matriz de rotación $[R]$; la cual posibilita transformar los datos crudos de aceleración (\vec{a}_{CRUDA}) en los valores calibrados que se desean hallar ($\vec{a}_{CALIBRADA}$). Esto es, la matriz $[R]$ permite realizar la siguiente operación:

$$[R]: \vec{a}_{CRUDA} \mapsto \vec{a}_{CALIBRADA} \quad (17.1)$$

A continuación, puede consultarse el código fuente indicado en el párrafo anterior:

[LAB]: Calibración Rotacional del *IMU Xsens DOT 7*. Resolución del modelo de calibración mediante Mínimos Cuadrados

En este laboratorio virtual se ha implementado un método de *Calibración Rotacional*, aplicado sobre el dispositivo de medida inercial *Xsens DOT 7*. Para garantizar la repetibilidad de los movimientos establecidos, se ha hecho uso del robot colaborativo *UR3e*.

Índice de Contenidos

17.0.- Material de laboratorio

17.1.- Importación de datos

17.1.1.- Movimientos realizados

17.2.- Ensamblaje de la matriz $[B]$. Valores *crudos* de aceleración

17.3.- Creación de la matriz inherente a los valores calibrados (teóricos) de la aceleración de la gravedad

17.4.- Cómputo de la matriz de rotación $[R]$

17.5.- Representación gráfica de los resultados obtenidos

17.6.- Verificación relativa a la matriz de rotación $[R]$. ¿ $[R]$ pertenece al grupo ortogonal especial $SO(3)$?

17.6.1.- Traspuesta de la matriz de rotación $[R]$. ¿ $[R]^T[R] = [I]$?

17.6.2.- Determinante de la matriz de rotación $[R]$. ¿ $|[R]| = 1$?

17.6.3.- Matriz inversa ($[R]^{-1}$). ¿ $[R]^T = [R]^{-1}$?

17.6.4.- Conclusiones

```
% Liberación de la memoria inicialmente ocupada en el Workspace de Matlab  
clear
```

```
% Limpieza de la ventana de comandos  
clc
```

17.0.- MATERIAL DE LABORATORIO



Figura 17.1.- Sistema de coordenadas vinculado a las unidades de medida inercial de la gama *Xsens DOT*. Imagen extraída como cortesía de [36].



Figura 17.2.- Vista isométrica del montaje experimental elaborado en el robot colaborativo *UR3e*, disponible en el *RoboLab* de la *EPI Gijón*.



Figura 17.3.- Vista frontal del montaje experimental elaborado en el robot colaborativo UR3e, disponible en el *RoboLab* de la *EPI Gijón*.

17.1.- IMPORTACIÓN DE DATOS

```
% A continuación, se realiza la importación de los datos capturados mediante
el IMU Xsens DOT8. Para ello, se invoca a la función
importfile_XsensDOT8MedicionCada15grados; previamente diseñada.
Datos_IMU = readtable("DATOS_Calibracion_rotacional_XsensDOT7.csv")
```

Tabla 17.1.- Fragmento de los datos inerciales capturados mediante el dispositivo *Xsens DOT 7*. Obsérvese que se ha registrado un total de 1.492 muestras de datos; relativos a la orientación espacial (cuaternión), la aceleración lineal y la velocidad angular de giro que posee el *IMU* monitorizado.

Datos_IMU = 1492x12 table

	PacketCounter	SampleTimeFine	Quat_W	Quat_X	Quat_Y	...
1	1	41026754	0.4107	-0.1856	-0.6908	
2	2	41043421	0.4106	-0.1858	-0.6910	
3	3	41060088	0.4106	-0.1858	-0.6911	
4	4	41076755	0.4105	-0.1860	-0.6912	
5	5	41093422	0.4105	-0.1860	-0.6913	
6	6	41110089	0.4104	-0.1861	-0.6915	

⋮

```
% Lectura de los datos de aceleración de la gravedad
```

```
% Aceleración en el eje X
```

```
Acc_X_m_s2 = Datos_IMU.Acc_X;
```

```
% Aceleración en el eje Y
```

```
Acc_Y_m_s2 = Datos_IMU.Acc_Y;
```

```
% Aceleración en el eje Z
```

```
Acc_Z_m_s2 = Datos_IMU.Acc_Z;
```

17.1.1.- Movimientos realizados

En la presente sección se muestra una representación esquemática sobre los sistemas de coordenadas vinculados al sensor inercial *Xsens DOT 7*, durante las respectivas etapas de la trayectoria programada.

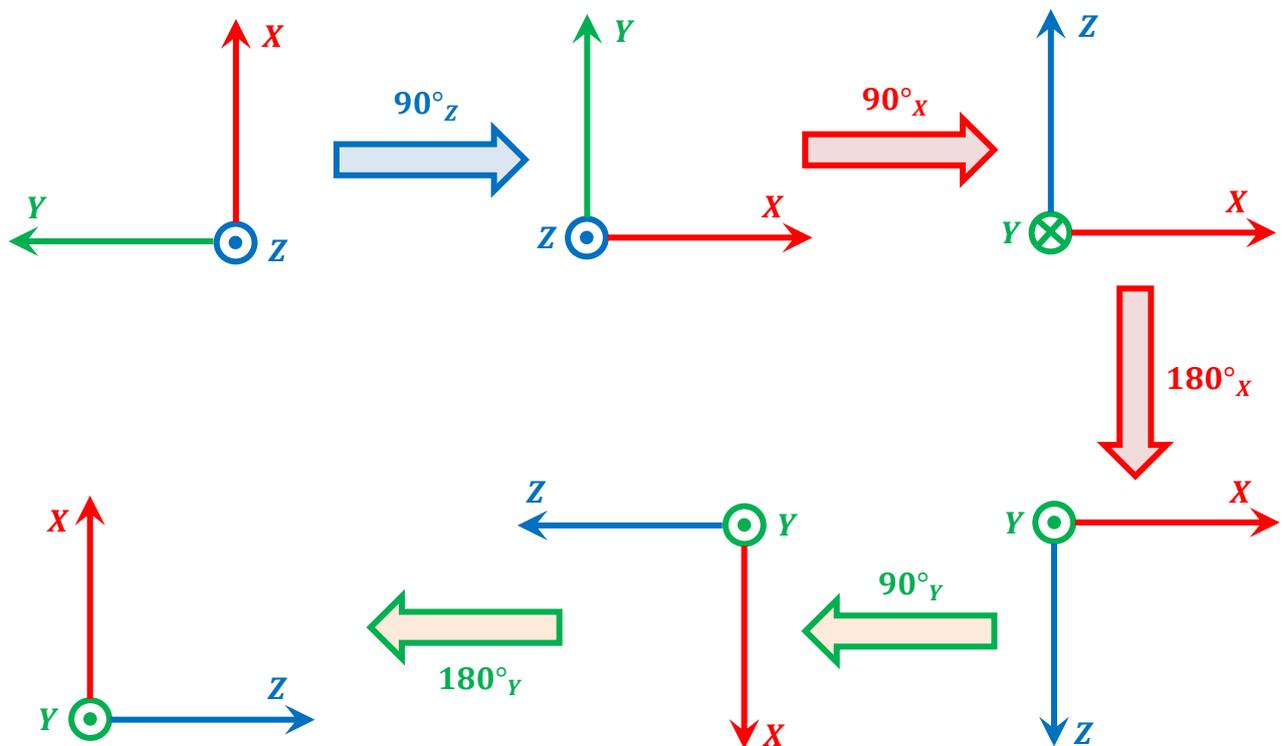


Figura 17.4.- Movimientos experimentados por el *IMU Xsens DOT 7*, los cuales han sido programados mediante el *cobot UR3e*.

17.2.- ENSAMBLAJE DE LA MATRIZ [B]. VALORES *CRUDOS* DE ACCELERACIÓN

```
% Construcción, por inspección, de la matriz de coeficientes [B]

% Extracción, por inspección, de los datos vinculados al parámetro ax_1
ax_1 = Acc_X_m_s2(20:304);

% Cálculo del valor medio inherente al parámetro ax_1
ax_1 = mean(ax_1);

% Extracción, por inspección, de los datos vinculados al parámetro ay_1
ay_1 = Acc_Y_m_s2(20:304);

% Cálculo del valor medio inherente al parámetro ay_1
ay_1 = mean(ay_1);

% Extracción, por inspección, de los datos vinculados al parámetro az_1
az_1 = Acc_Z_m_s2(20:304);

% Cálculo del valor medio inherente al parámetro az_1
az_1 = mean(az_1);

% Extracción, por inspección, de los datos vinculados al parámetro ax_2
ax_2 = Acc_X_m_s2(390:506);

% Cálculo del valor medio inherente al parámetro ax_2
ax_2 = mean(ax_2);

% Extracción, por inspección, de los datos vinculados al parámetro ay_2
ay_2 = Acc_Y_m_s2(390:506);

% Cálculo del valor medio inherente al parámetro ay_2
ay_2 = mean(ay_2);

% Extracción, por inspección, de los datos vinculados al parámetro az_2
az_2 = Acc_Z_m_s2(390:506);

% Cálculo del valor medio inherente al parámetro az_2
az_2 = mean(az_2);

% Extracción, por inspección, de los datos vinculados al parámetro ax_3
ax_3 = Acc_X_m_s2(613:717);

% Cálculo del valor medio inherente al parámetro ax_3
```

```
ax_3 = mean(ax_3);

% Extracción, por inspección, de los datos vinculados al parámetro ay_3
ay_3 = Acc_Y_m_s2(613:717);

% Cálculo del valor medio inherente al parámetro ay_3
ay_3 = mean(ay_3);

% Extracción, por inspección, de los datos vinculados al parámetro az_3
az_3 = Acc_Z_m_s2(613:717);

% Cálculo del valor medio inherente al parámetro az_3
az_3 = mean(az_3);

% Extracción, por inspección, de los datos vinculados al parámetro ax_4
ax_4 = Acc_X_m_s2(830:942);

% Cálculo del valor medio inherente al parámetro ax_4
ax_4 = mean(ax_4);

% Extracción, por inspección, de los datos vinculados al parámetro ay_4
ay_4 = Acc_Y_m_s2(830:942);

% Cálculo del valor medio inherente al parámetro ay_4
ay_4 = mean(ay_4);

% Extracción, por inspección, de los datos vinculados al parámetro az_4
az_4 = Acc_Z_m_s2(830:942);

% Cálculo del valor medio inherente al parámetro az_4
az_4 = mean(az_4);

% Extracción, por inspección, de los datos vinculados al parámetro ax_5
ax_5 = Acc_X_m_s2(1027:1152);

% Cálculo del valor medio inherente al parámetro ax_5
ax_5 = mean(ax_5);

% Extracción, por inspección, de los datos vinculados al parámetro ay_5
ay_5 = Acc_Y_m_s2(1027:1152);

% Cálculo del valor medio inherente al parámetro ay_5
ay_5 = mean(ay_5);

% Extracción, por inspección, de los datos vinculados al parámetro az_5
az_5 = Acc_Z_m_s2(1027:1152);
```

```
% Cálculo del valor medio inherente al parámetro az_5
az_5 = mean(az_5);

% Extracción, por inspección, de los datos vinculados al parámetro ax_6
ax_6 = Acc_X_m_s2(1262:1463);

% Cálculo del valor medio inherente al parámetro ax_6
ax_6 = mean(ax_6);

% Extracción, por inspección, de los datos vinculados al parámetro ay_6
ay_6 = Acc_Y_m_s2(1262:1463);

% Cálculo del valor medio inherente al parámetro ay_6
ay_6 = mean(ay_6);

% Extracción, por inspección, de los datos vinculados al parámetro az_6
az_6 = Acc_Z_m_s2(1262:1463);

% Cálculo del valor medio inherente al parámetro az_6
az_6 = mean(az_6);

% Construcción de la matriz relativa a los datos de aceleración registrados
% ([ACELERACION_CRUDA] <- Medidas de aceleración registradas mediante el
dispositivo Xsens DOT)
ACELERACION_CRUDA = [ax_1 ay_1 az_1;
                    ax_2 ay_2 az_2;
                    ax_3 ay_3 az_3;
                    ax_4 ay_4 az_4;
                    ax_5 ay_5 az_5;
                    ax_6 ay_6 az_6]
```

```
ACELERACION_CRUDA = 6x3
    7.6467    6.1838   -0.2367
    6.3236   -7.5506   -0.2625
    0.0265   -0.0547    9.6081
    0.0657   -0.0071  -10.0659
   -7.5156   -6.2212   -0.2421
    7.6009    6.2452   -0.2297
```

17.3.- CREACIÓN DE LA MATRIZ INHERENTE A LOS VALORES CALIBRADOS (TEÓRICOS) DE LA ACELERACIÓN DE LA GRAVEDAD

```
% Valor teórico ligado a la aceleración de la gravedad [m/s^2]
g = 9.80665;

% Matriz de aceleración calibrada (TEÓRICA)
ACELERACION_CALIBRADA = [-g  0  0;
                          0 -g  0;
                          0  0 -g;
                          0  0  g;
                          g  0  0;
                          -g  0  0]
```

```
ACELERACION_CALIBRADA = 6x3
-9.8066        0        0
         0   -9.8066        0
         0        0   -9.8066
         0        0    9.8066
    9.8066        0        0
   -9.8066        0        0
```

17.4.- CÓMPUTO DE LA MATRIZ DE ROTACIÓN [R]

Sea [A] la matriz correspondiente a los datos calibrados de aceleración; y sea [B] la matriz vinculada a los valores crudos de aceleración, registrados mediante el sensor inercial empleado. Ambos conjuntos de datos están relacionados mediante una matriz de rotación [R], verificando la siguiente expresión matemática:

$$[A]_{3 \times m} = [R]_{3 \times 3}([B]_{m \times 3})_{3 \times m}^T$$

$$[A]([B]^T)^{-1} = [R][B]^T([B]^T)^{-1} = [R][I]$$

$$[R] = [A]([B]^T)^{-1}$$

```
% ACELERACION_CALIBRADA = R * ACELERACION_CRUDA'
R = ACELERACION_CALIBRADA' * pinv(ACELERACION_CRUDA')
```

```
R = 3x3
    -0.7665    -0.6419    -0.0032
    -0.6310     0.7702     0.0004
    -0.0230     0.0160    -0.9956
```

```
% Componente X de g_cruda
gx_cruda = Acc_X_m_s2;

% Componente Y de g_cruda
gy_cruda = Acc_Y_m_s2;

% Componente Z de g_cruda
gz_cruda = Acc_Z_m_s2;

% Ensamblaje de la matriz asociada a la aceleración cruda de la gravedad
(g_cruda)
g_cruda = [gx_cruda, gy_cruda, gz_cruda]
```

```
g_cruda = 1492x3
         0         0         0
    7.6443    6.1871   -0.2229
    7.6527    6.1780   -0.2424
    7.6492    6.1851   -0.2334
    7.6528    6.1854   -0.2406
    7.6432    6.1881   -0.2353
    7.6564    6.1728   -0.2331
    7.6484    6.1756   -0.2348
    7.6463    6.1745   -0.2352
    7.6504    6.1860   -0.2561
    :
```

```
% Obtención de los datos calibrados de aceleración. Para ello, se efectúa
el producto con la matriz de rotación [R]; previamente calculada.
g_calibrada = R * g_cruda';

% Valores calibrados de g. Se muestran las tres componentes cartesianas de
la aceleración de la gravedad
g_calibrada = g_calibrada'
```

```
g_calibrada = 1492x3
         0         0         0
   -9.8304   -0.0584    0.1447
   -9.8309   -0.0707    0.1638
   -9.8328   -0.0630    0.1550
```

```
-9.8358   -0.0651   0.1620  
-9.8302   -0.0569   0.1570  
-9.8305   -0.0770   0.1543  
-9.8261   -0.0698   0.1562  
-9.8239   -0.0693   0.1567  
-9.8343   -0.0631   0.1775  
⋮
```

```
% Componente X de la aceleración de la gravedad calibrada  
gx_calibrada = g_calibrada(:,1);  
  
% Componente Y de la aceleración de la gravedad calibrada  
gy_calibrada = g_calibrada(:,2);  
  
% Componente Z de la aceleración de la gravedad calibrada  
gz_calibrada = g_calibrada(:,3);
```

17.5.- REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS OBTENIDOS

```
% Creación de una nueva figura  
figure  
  
% Activación de la primera casilla dentro de la figura generada  
subplot(2,2,1)  
  
% Representación gráfica de la aceleración de la gravedad capturada en la  
dirección del eje X  
plot(gx_cruda, 'r--', 'LineWidth', 2)  
  
% Activación de la rejilla en la figura maquetada  
grid on  
  
% Inclusión de una etiqueta en el eje X de la gráfica  
xlabel('Número de muestra [-]')  
  
% Inclusión de una etiqueta en el eje Y de la gráfica  
ylabel('$g_x [m/s^2]$', 'interpreter', 'latex', 'FontSize', 13)  
  
% Incorporación del título a la representación gráfica  
title('Aceleración gravedad (EJE X)')  
  
% Con esta instrucción, se mantiene activa la figura actual
```

```
hold on

% Representación gráfica del parámetro gx_calibrada
plot(gx_calibrada, 'k-', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Incorporación de una leyenda a la figura maquetada
legend('gx cruda', 'gx calib')

% Activación de la segunda casilla dentro de la figura generada
subplot(2,2,2)

% Representación gráfica de la aceleración de la gravedad capturada en la
dirección del eje Y
plot(gy_cruda, 'g--', 'LineWidth', 2)

% Activación de la rejilla en la figura maquetada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Número de muestra [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel('$g_y [m/s^2]$', 'interpreter', 'latex', 'FontSize', 13)

% Incorporación del título a la representación gráfica
title('Aceleración gravedad (EJE Y)')

% Se mantiene activa la figura actual
hold on

% Representación gráfica del parámetro gy_calibrada
plot(gy_calibrada, 'k-', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Incorporación de una leyenda a la figura maquetada
legend('gy cruda', 'gy calib')

% Activación de la tercera y cuarta casilla dentro de la figura generada
subplot(2,2,[3,4])
```

```
% Representación gráfica de la aceleración de la gravedad capturada en la
dirección del eje Z
plot(gz_cruda, 'b--', 'LineWidth', 2)

% Activación de la rejilla en la figura maquetada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Número de muestra [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel('$g_z$ [m/s^2]', 'interpreter', 'latex', 'FontSize', 13)

% Incorporación del título a la representación gráfica
title('Aceleración gravedad (EJE Z)')

% Se mantiene activa la figura actual
hold on

% Representación gráfica del parámetro gz_calibrada
plot(gz_calibrada, 'k-', 'LineWidth', 2)

% Se desactiva la representación gráfica actual, puesto que no se va a seguir
pintando dentro de ella
hold off

% Incorporación de una leyenda a la figura maquetada
legend('gz cruda', 'gz calib')

% Creación de un título global para la figura construida con anterioridad
sgtitle('Calibración del Sensor Inercial')
```

Calibración del Sensor Inercial

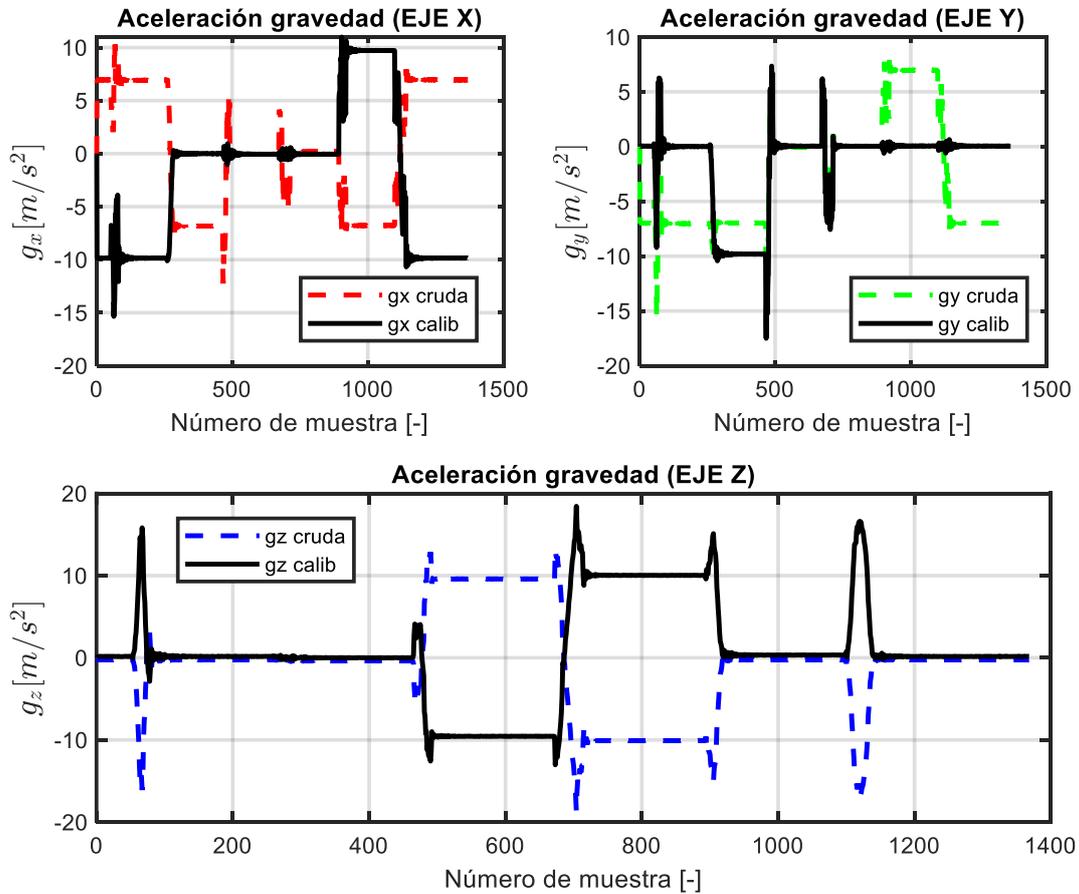


Figura 17.5.- Representación gráfica de los datos *crudos* vinculados a la aceleración de la gravedad, junto con sus correspondientes valores calibrados.

17.6.- VERIFICACIÓN RELATIVA A LA MATRIZ DE ROTACIÓN $[R]$. ¿ $[R]$ PERTENECE AL GRUPO ORTOGONAL ESPECIAL $SO(3)$?

En este apartado se comprueba que la matriz de rotación obtenida cumple las propiedades consustanciales al grupo ortogonal especial $SO(3)$.

17.6.1.- Traspuesta de la matriz de rotación $[R]$. ¿ $[R]^T[R] = [I]$?

```
% Matriz traspuesta de [R]
R_traspuesta = R'
```

```
R_traspuesta = 3x3
-0.7665   -0.6310   -0.0230
-0.6419    0.7702    0.0160
-0.0032    0.0004   -0.9956
```

```
% Determinante vinculado a la matriz traspuesta de [R]
Determinante_DE_R_traspuesta = det(R_traspuesta)
```

```
Determinante_DE_R_traspuesta = 0.9910
```

```
% Cómputo de la matriz Identidad de orden 3
Matriz_Identidad = R_traspuesta * R
```

```
Matriz_Identidad = 3x3
 0.9862   0.0057   0.0252
 0.0057   1.0055  -0.0135
 0.0252  -0.0135   0.9913
```

La matriz obtenida con anterioridad presenta una anatomía muy similar a la matriz

Identidad (de orden 3) teórica: $[I]_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3}$. OK !

17.6.2.- Determinante de la matriz de rotación [R]. ¿|[R]| = 1?

```
% Cálculo del determinante correspondiente a la matriz de rotación [R]
Determinante_R = det(R)
```

```
Determinante_R = 0.9910
```

```
% Diferencia existente entre la unidad y el determinante de [R]
Error_Estimado = 1 - Determinante_R
```

```
Error_Estimado = 0.0090
```

En los resultados anteriores, puede observarse que el determinante asociado a la matriz de rotación [R] ofrece un valor próximo a la unidad. |[R]| \cong 1. OK !

17.6.3.- Matriz inversa ($[R]^{-1}$). ¿ $[R]^T = [R]^{-1}$?

```
% Mediante esta instrucción se muestra por pantalla la matriz traspuesta de [R]
R_traspuesta
```

```
R_traspuesta = 3x3
-0.7665    -0.6310    -0.0230
-0.6419     0.7702     0.0160
-0.0032     0.0004    -0.9956
```

```
% Obtención de la matriz pseudoinversa (Moore-Penrose) asociada a [R]
R_inversa = pinv(R)
```

```
R_inversa = 3x3
-0.7738    -0.6450     0.0023
-0.6339     0.7700     0.0023
 0.0077     0.0273    -1.0044
```

```
% Cuantificación de la diferencia (error) existente entre R_traspuesta y R_inversa.
% El resultado se expresa como una nueva matriz de error.
Matriz_Error = abs(R_traspuesta) - abs(R_inversa)
```

```
Matriz_Error = 3x3
-0.0072    -0.0140     0.0207
 0.0080     0.0002     0.0136
-0.0045    -0.0269    -0.0088
```

```
% Cómputo de un valor medio para la matriz de error anteriormente calculada
EstimacionFinal_Del_Error = abs(mean(mean(Matriz_Error)))
```

```
EstimacionFinal_Del_Error = 0.0021
```

En este subapartado se ha comprobado que, efectivamente, las matrices traspuesta e inversa de $[R]$ son prácticamente idénticas: $[R]^T \cong [R]^{-1}$. *OK!*

17.6.4.- Conclusiones

En conformidad con el estudio realizado, puede afirmarse que tanto la matriz de rotación $[R]$ como su traspuesta ($[R]^T$) pertenecen al *Grupo Ortogonal Especial* (*Grupo Rotacional - $SO(3)$*):

$$[R] \in SO(3)$$

$$[R]^T \in SO(3)$$

```
% Mediante la siguiente instrucción, se guarda la matriz de calibración  
% calculada anteriormente:  
save('R_calibracion.mat', 'R')
```

18. Anexo H: Código Fuente implementado para efectuar la estimación inercial de una rotación articular

Con la finalidad de automatizar los cálculos vinculados a la modelización tridimensional del movimiento desarrollado por una articulación, se ha programado el código fuente recogido en el presente apéndice.

Cabe destacar que el fundamento teórico subyacente al algoritmo implementado se basa en el *manejo* de cuaterniones. El correspondiente desarrollo matemático, construido durante el diseño de dicha estrategia computacional, ya ha sido detallado exhaustivamente en la sección "2.1.- Método computacional destinado a cuantificar el movimiento tridimensional de una".

A continuación, puede consultarse el código fuente desarrollado en lenguaje *MATLAB*:

[LAB]: Cálculo del Ángulo de Flexión del codo mediante dos *IMUs* de la gama *Xsens DOT*. Incorporación de errores de orientación y desalineación *segmento – sensor*

En este *script* se implementa un algoritmo de fusión sensorial (*Filtro de Kalman*) que permite estimar la orientación espacial de un sensor inercial, perteneciente a la familia *MATRIX*.

Índice de Contenidos

18.0.- Materiales de laboratorio

18.1.- Importación de los datos capturados

18.1.1.- Medidas de los acelerómetros

18.1.2.- Medidas del giroscopio

18.2.- Estimación de la orientación correspondiente a cada IMU. Fusión sensorial

18.3.- Algoritmo implementado para la estimación del ángulo de flexión (3D) asociado al codo

18.3.1.- ${}^{BS}q_{F0}$

18.3.2.- ${}^{GB}q_F$

18.3.3.- ${}^{BS}q_{A0}$

18.3.4.- ${}^{GB}q_A$

18.3.5.- ${}^Bq_{AF}$

18.3.6.- Representación gráfica de los parámetros: *knee flexion* (ψ_{FE}), *knee abduction* (θ_{AA}) y *knee internal rotation* (ϕ_{IE})

18.4.- Incorporación de errores de alineación *segmento-sensor*. Método 1

18.5.- Incorporación de errores de alineación *segmento-sensor*. Método 2

18.5.1.- Cálculo y representación de la señal de error durante la estimación del ángulo de abducción del codo

18.6.- Comparativa de varios casos de estudio. Observación del error generado durante la estimación de la rotación articular

18.7.- Incorporación de los errores de orientación del *IMU*

18.7.1.- ${}^{GS}q'_A$

18.7.2.- ${}^{GS}q'_F$

18.7.3.- ${}^Bq_{AF_{est2}}$

18.7.4.- Obtención de los *Ángulos de Euler* asociados al cuaternión ${}^Bq_{AF_{est2}}$

18.8.- Valoración final de los errores de estimación de la orientación

18.8.1.- ${}^Bq_{AF_{ref}}$

18.8.2.- ϕ_{ERROR}

18.8.3.- θ_{ERROR}

18.8.4.- ψ_{ERROR}

18.0.- MATERIALES DE LABORATORIO



Figura 18.1.- Dispositivo experimental empleado durante la captura de datos.



Figura 18.2.- Vista adicional sobre el medidor de ángulos utilizado durante la experiencia de laboratorio.



Figura 18.3.- Sistema de coordenadas cartesianas asociado a los sensores inerciales de la familia *Xsens DOT*.

18.1.- IMPORTACIÓN DE LOS DATOS CAPTURADOS

```
% Limpieza de la ventana de comandos. Liberación de la memoria inicialmente
ocupada en el Workspace de MatLab
clc, clear

% Lectura de los datos capturados mediante la unidad de medida inercial.
Llamada a la función readtable.
DatosIMU_Xsens_Antebrazo = readtable("Forearm_XsensDOT7.csv")

% Lectura de los datos capturados mediante otro IMU. Llamada a la función
readtable.
DatosIMU_Xsens_Brazo = importfile_ArmXsensDOT8("Arm_XsensDOT8.csv")
```

18.1.1.- Medidas de los acelerómetros

```
% A continuación, se construye el vector de aceleración (ejes X, Y, Z) --
ANTEBRAZO
LecturaDelAcelerometro_Antebrazo = [DatosIMU_Xsens_Antebrazo.Acc_X,
DatosIMU_Xsens_Antebrazo.Acc_Y, DatosIMU_Xsens_Antebrazo.Acc_Z]; % m/s^2

% Creación de una nueva figura
figure

% Representación gráfica de la aceleración asociada al antebrazo monitorizado
plot(LecturaDelAcelerometro_Antebrazo, 'LineWidth', 1.5)

% Activación de rejilla en la figura anterior
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel ('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Accelerometer readings [m/s^2]')

% Incorporación del título a la figura maquetada
title('Datos del Acelerómetro. Antebrazo')
% A continuación, se construye el vector de aceleración (ejes X, Y, Z) --
BRAZO
LecturaDelAcelerometro_Brazo = [DatosIMU_Xsens_Brazo.Acc_X,
DatosIMU_Xsens_Brazo.Acc_Y, DatosIMU_Xsens_Brazo.Acc_Z]; % m/s^2

% Creación de una nueva figura
```

figure

```
% Representación gráfica de la aceleración asociada al brazo monitorizado  
plot(LecturaDelAcelerometro_Brazo, 'LineWidth', 1.5)
```

```
% Activación de la rejilla en la figura anterior  
grid on
```

```
% Inclusión de una etiqueta al eje X de la gráfica  
xlabel ('Sample [-]')
```

```
% Inclusión de una etiqueta al eje Y de la gráfica  
ylabel('Accelerometer readings [m/s^2]')
```

```
% Incorporación del título a la figura maquetada  
title('Datos del Acelerómetro. Brazo')
```

18.1.2.- Medidas del giroscopio

```
% A continuación, se construye el vector de velocidad angular de giro (ejes  
X, Y, Z) --> ANTEBRAZO  
% [degrees/s]
```

```
LecturaDelGiroscopio_Antebrazo = [DatosIMU_Xsens_Antebrazo.Gyr_X -  
mean(DatosIMU_Xsens_Antebrazo.Gyr_X), ...  
    DatosIMU_Xsens_Antebrazo.Gyr_Y - mean(DatosIMU_Xsens_Antebrazo.Gyr_Y),  
DatosIMU_Xsens_Antebrazo.Gyr_Z - mean(DatosIMU_Xsens_Antebrazo.Gyr_Z)];
```

```
% Creación de una nueva figura  
figure
```

```
% Representación gráfica de la aceleración asociada al antebrazo monitorizado  
plot(LecturaDelGiroscopio_Antebrazo, 'LineWidth', 1.5)
```

```
% Activación de la rejilla en la figura anterior  
grid on
```

```
% Inclusión de una etiqueta al eje X de la gráfica  
xlabel ('Sample [-]')
```

```
% Inclusión de una etiqueta al eje Y de la gráfica  
ylabel('Gyroscope readings [degrees/s]')
```

```
% Incorporación del título a la figura maquetada  
title('Datos del Giroscopio. Antebrazo')
```

```
% A continuación, se construye el vector de velocidad angular de giro (ejes X, Y, Z) --> BRAZO
% degrees/s
LecturaDelGiroscopio_Brazo = [DatosIMU_Xsens_Brazo.Gyr_X -
mean(DatosIMU_Xsens_Brazo.Gyr_X), ...
    DatosIMU_Xsens_Brazo.Gyr_Y - mean(DatosIMU_Xsens_Brazo.Gyr_Y),
DatosIMU_Xsens_Brazo.Gyr_Z - mean(DatosIMU_Xsens_Brazo.Gyr_Z)];

% Creación de una nueva figura
figure

% Representación gráfica de la aceleración asociada al brazo monitorizado
plot(LecturaDelGiroscopio_Brazo, 'LineWidth', 1.5)

% Activación de la rejilla en la figura anterior
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel ('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Gyroscope readings [degrees/s]')

% Incorporación del título a la figura maquetada
title('Datos del Giroscopio. Brazo')
```

18.2.- ESTIMACIÓN DE LA ORIENTACIÓN CORRESPONDIENTE A CADA IMU. FUSIÓN SENSORIAL

```
% Forearm (antebrazo)
% Construcción del cuaternión de orientación vinculado a la orientación
% espacial (3D) del antebrazo
GS_q_F = quaternion(DatosIMU_Xsens_Antebrazo.Quat_W,
DatosIMU_Xsens_Antebrazo.Quat_X, DatosIMU_Xsens_Antebrazo.Quat_Y,
DatosIMU_Xsens_Antebrazo.Quat_Z);

% Creación de una nueva figura
figure

% Representación gráfica de los tres ángulos de Euler ligados al cuaternión
anterior
plot(180/pi*(quat2eul(GS_q_F, 'ZYX')), 'LineWidth', 2)
```

```
% Incorporación del título a la figura maquetada
title('Estimación de Orientación (Antebrazo)')

% Adición de una leyenda en la figura creada
legend('Z-axis', 'Y-axis', 'X-axis')

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Time (s)')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Rotation (degrees)')

% Activación de la rejilla en la figura anterior
grid on
% Arm (brazo)
% Construcción del cuaternión de orientación vinculado a la orientación
% espacial (3D) del brazo
GS_q_A = quaternion(DatosIMU_Xsens_Brazo.Quat_W,
DatosIMU_Xsens_Brazo.Quat_X, DatosIMU_Xsens_Brazo.Quat_Y,
DatosIMU_Xsens_Brazo.Quat_Z);

% Creación de una nueva figura
figure

% Representación gráfica de los tres ángulos de Euler ligados al cuaternión
anterior
plot(180/pi*(quat2eul(GS_q_A,'ZYX')), 'LineWidth', 2)

% Incorporación del título a la figura maquetada
title('Estimación de Orientación (Brazo)')

% Adición de una leyenda en la figura creada
legend('Z-axis', 'Y-axis', 'X-axis')

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Time (s)')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Rotation (degrees)')

% Activación de la rejilla en la figura anterior
grid on
```

18.3.- ALGORITMO IMPLEMENTADO PARA LA ESTIMACIÓN DEL ÁNGULO DE FLEXIÓN (3D) ASOCIADO AL CODO

Notación: $A \equiv arm$, $F \equiv forearm$.

```
% Número de cuaterniones contenidos dentro del vector GS_q_F
m1 = length(GS_q_F);

% Número de cuaterniones contenidos dentro del vector GS_q_A
m2 = length(GS_q_A);

% Identificación del valor mínimo entre m1 y m2 --> Número mínimo de
cuaterniones que intervienen en los cálculos
m = min(m1, m2);
```

18.3.1.- ${}^{BS}q_{F0}$

$${}^{BS}q_{F0} = {}^{GS}q_{F0} \otimes {}^{GB}q_{F0}$$

```
% BS_q_F0
% Matriz de transformación homogénea, en forma de cuaternión, que permite
relacionar
% la orientación del sensor y el antebrazo
BS_q_F0 = tform2quat([1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]);

% Creación del objeto de tipo cuaternión (BS_q_F0) vinculado a las coordenadas
anteriores
BS_q_F0 = quaternion(BS_q_F0);
```

18.3.2.- ${}^{GB}q_F$

$${}^{GB}q_F = {}^{GS}q_F \otimes {}^{BS}q_{F0}$$

```
% GB_q_F
% Multiplicación del vector de cuaterniones GS_q_F por el cuaternión conjugado
de BS_q_F0. El resultado es otro vector de cuaterniones (GB_q_F)
GB_q_F = quatmultiply(GS_q_F, quatconj(BS_q_F0));
```

18.3.3.- ${}^{BS}q_{A0}$

$${}^{BS}q_{A0}^* = {}^{GS}q_{A0}^* \otimes {}^{GB}q_{A0}$$

```
% BS_q_A0
% Matriz de transformación homogénea, en forma de cuaternión, que permite
relacionar
% la orientación del sensor y el brazo
BS_q_A0 = tform2quat([1 0 0 0;0 1 0 0;0 0 1 0;0 0 0 1]);

% Creación del objeto de tipo cuaternión (BS_q_A0) vinculado a las coordenadas
anteriores
BS_q_A0 = quaternion(BS_q_A0);
```

18.3.4.- ${}^{GB}q_A$

$${}^{GB}q_A = {}^{GS}q_A \otimes {}^{BS}q_{A0}^*$$

```
% GB_q_A
% Multiplicación del vector de cuaterniones GS_q_A por el cuaternión conjugado
de BS_q_A0. El resultado es
% otro vector de cuaterniones (GB_q_A)
GB_q_A = quatmultiply(GS_q_A, quatconj(BS_q_A0));

% Cálculo del cuaternión conjugado de GB_q_A --> Resultado: GB_q_A_conjugado
GB_q_A_conjugado = quatconj(GB_q_A);
```

18.3.5.- ${}^Bq_{AF}$

$${}^Bq_{AF} = {}^{GB}q_A^* \otimes {}^{GB}q_F$$

```
% B_q_AF -> Cuaternión final de orientación
% Multiplicación de los vectores de cuaterniones GB_q_F y GB_q_A_conjugado.
El resultado se almacena en
% otro vector de cuaterniones --> B_q_AF
B_q_AF = quatmultiply(GB_q_F(1:m), GB_q_A_conjugado(1:m))

% Cálculo de los Ángulos de Euler intrínsecos al cuaternión de orientación
B_q_AF. El segundo parámetro rige la codificación
% de ángulos empleada durante la conversión:
```

```

AngulosDeEuler_ZYX_OrientacionFinal = quat2eul(B_q_AF, 'ZYX');

% Obtención de los tres ángulos asociados al movimiento 3D del codo. NOTA:
Estos 3 ángulos se expresan en grados [°]
% Flexión del codo
elbow_flexion_grados = 180/pi * AngulosDeEuler_ZYX_OrientacionFinal(:,3);

% Abducción del codo
elbow_abduction_grados = 180/pi * AngulosDeEuler_ZYX_OrientacionFinal(:,2);

% Rotación interna del codo
elbow_internal_rotation_grados = 180/pi *
AngulosDeEuler_ZYX_OrientacionFinal(:,1);

```

18.3.6.- Representación gráfica de los parámetros: *knee flexion* (ψ_{FE}), *knee abduction* (θ_{AA}) y *knee internal rotation* (ϕ_{IE})

```

% Creación de una nueva figura
figure

% Activación de la primera casilla en la matriz de figuras creada
subplot(311)

% Representación gráfica del parámetro elbow_flexion_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_flexion_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow flexion', '[degrees]'})

% Activación de la segunda casilla en la matriz de figuras creada
subplot(312)

% Representación gráfica del parámetro elbow_abduction_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_abduction_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

```

```
% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow abduction', '[degrees]'})

% Mediante las siguientes líneas de código se añade una etiqueta que indica
% el último ángulo estimado durante el experimento
ax = gca;
chart = ax.Children(1);
datatip(chart,968,-54.87,"Location","northeast");

% Activación de la tercera casilla en la matriz de figuras creada
subplot(313)

% Representación gráfica del parámetro elbow_internal_rotation_grados. Se
% utiliza un ancho de línea de 1.5 [puntos]
plot(elbow_internal_rotation_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow internal', 'rotation [degrees]'})
```

18.4.- INCORPORACIÓN DE LOS ERRORES DE ALINEACIÓN SEGMENTO-SENSOR. MÉTODO 1

$$q_r = \cos\left(\frac{\phi}{2}\right)(1000) + \sin\left(\frac{\phi}{2}\right)(0100)$$

$$q_p = \cos\left(\frac{\theta}{2}\right)(1000) + \sin\left(\frac{\theta}{2}\right)(0010)$$

$$q_y = \cos\left(\frac{\varphi}{2}\right)(1000) + \sin\left(\frac{\varphi}{2}\right)(0001)$$

$$\varphi \equiv \psi$$

$$q = q_y \otimes q_p \otimes q_r$$

$$\hat{q} = (q_y \otimes q_{y_e}) \otimes (q_p \otimes q_{p_e}) \otimes (q_r \otimes q_{r_e})$$

```
% Obtención de los 3 ángulos de Euler asociados al cuaternión final de
orientación
% phi
phi = elbow_internal_rotation_grados * pi/180;
% theta
theta = elbow_abduction_grados * pi/180;
% chi
chi = elbow_flexion_grados * pi/180;

% Definición de los ángulos de error simulados
% X [grados] de error en el roll
phi_error = 5 * pi/180;
% Y [grados] de error en el pitch
theta_error = 5 * pi/180;
% Z [grados] de error en el yaw
chi_error = 7 * pi/180;

% Creación de 4 cuaterniones de referencia
% q1
q1 = quaternion(1, 0, 0, 0);
% q2
q2 = quaternion(0, 1, 0, 0);
% q3
q3 = quaternion(0, 0, 1, 0);
% q4
q4 = quaternion(0, 0, 0, 1);

% -----

% Cálculo de los tres cuaterniones asociados a los movimientos de roll, pitch
y yaw:
% ROLL
qr = cos(phi/2)*q1 + sin(phi/2)*q2;
% PITCH
qp = cos(theta/2)*q1 + sin(theta/2)*q3;
% YAW
qy = cos(chi/2)*q1 + sin(chi/2)*q4;

% Creación de tres cuaterniones de error, vinculados a los movimientos de
roll, pitch y yaw:
% error en el ROLL
qr_e = cos(phi_error/2)*q1 + sin(phi_error/2)*q2;
% error en el PITCH
qp_e = cos(theta_error/2)*q1 + sin(theta_error/2)*q3;
```

```

% error en el YAW
qy_e = cos(chi_error/2)*q1 + sin(chi_error/2)*q4;

% Incorporación de los errores de alineación al cuaternión final de
orientación. Multiplicación de cuaterniones:
q_estimado = quatmultiply(quatmultiply(qy, qy_e), quatmultiply(qp, qp_e),
quatmultiply(qr, qr_e))

% Cálculo de los Ángulos de Euler intrínsecos al cuaternión de orientación
q_estimado. El segundo parámetro rige la codificación
% de ángulos empleada durante la conversión:
AngulosDeEuler_ZYX_OrientacionFinal_CON_ERRORES = quat2eul(q_estimado,
'ZYX');

% Obtención de los tres ángulos asociados al movimiento 3D del codo. NOTA:
Estos 3 ángulos se expresan en grados [°]
% Flexión del codo
elbow_flexion_grados = 180/pi *
AngulosDeEuler_ZYX_OrientacionFinal_CON_ERRORES(:,3);
% Abducción del codo
elbow_abduction_grados = 180/pi *
AngulosDeEuler_ZYX_OrientacionFinal_CON_ERRORES(:,2);
% Rotación interna del codo
elbow_internal_rotation_grados = 180/pi *
AngulosDeEuler_ZYX_OrientacionFinal_CON_ERRORES(:,1);

% -----

% Creación de una nueva figura
figure

% Activación de la primera casilla en la matriz de figuras creada
subplot(311)

% Representación gráfica del parámetro elbow_flexion_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_flexion_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow flexion', '[degrees]'})

```

```
% Activación de la segunda casilla en la matriz de figuras creada
subplot(312)

% Representación gráfica del parámetro elbow_abduction_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_abduction_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow abduction', '[degrees]'})

% Mediante las siguientes líneas de código se añade una etiqueta que indica
% el último ángulo estimado durante el experimento (52.1 [°],
aproximadamente):
ax = gca;
chart = ax.Children(1);
datatip(chart,968,-54.87,"Location","northeast");

% Activación de la tercera casilla en la matriz de figuras creada
subplot(313)

% Representación gráfica del parámetro elbow_internal_rotation_grados. Se
utiliza un ancho de línea de 1.5 [puntos]
plot(elbow_internal_rotation_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow internal', 'rotation [degrees]'})
```

18.5.- INCORPORACIÓN DE LOS ERRORES DE ALINEACIÓN SEGMENTO-SENSOR. MÉTODO 2

```
% ----- BRAZO -----  
% Cálculo de los Ángulos de Euler intrínsecos al cuaternión de orientación  
GS_q_A. El segundo parámetro rige la codificación  
% de ángulos empleada durante la conversión. Esta información se encuentra  
vinculada al IMU colocado en el brazo:  
AngulosDeEuler_q_A = quat2eul(GS_q_A, 'ZYX');  
  
% A continuación, se crean tres vectores columna inherentes a sendos ángulos  
de Euler (chi_A, theta_A, phi_A):  
chi_A = AngulosDeEuler_q_A(:, 3);  
theta_A = AngulosDeEuler_q_A(:, 2);  
phi_A = AngulosDeEuler_q_A(:, 1);  
  
% auxiliar = [0.5, 1, 3, 5, 8];  
  
% Definición de los ángulos de error simulados:  
chi_error_A = 0 * pi/180;  
theta_error_A = 00 * pi/180;  
phi_error_A = 0 * pi/180;  
  
% Construcción de los cuaterniones involucrados en el cálculo de GB_qprima_A0:  
% Cuaternión de error asociado al movimiento yaw del brazo, en el instante de  
tiempo inicial  
q_A0_ye = cos(chi_error_A/2)*q1 + sin(chi_error_A/2)*q4;  
  
% Cuaternión asociado al movimiento yaw del brazo, durante un tiempo genérico  
t  
q_A_y = cos(chi_A/2)*q1 + sin(chi_A/2)*q4;  
  
% Cuaternión asociado al movimiento yaw del brazo, en el instante de tiempo  
inicial  
q_A0_y = q_A_y(1);  
  
% Cuaternión correspondiente al movimiento de tipo pitch del brazo  
q_A_p = cos(theta_A/2)*q1 + sin(theta_A/2)*q3;  
  
% Cuaternión ligado al movimiento pitch del brazo en el instante inicial  
q_A0_p = q_A_p(1);  
  
% Cuaternión de error vinculado al movimiento pitch del brazo en el instante  
inicial
```

```

q_A0_pe = cos(theta_error_A/2)*q1 + sin(theta_error_A/2)*q3;

% Cuaternión correspondiente al movimiento de tipo roll del brazo
q_A_r = cos(phi_A/2)*q1 + sin(phi_A/2)*q2;

% Cuaternión ligado al movimiento roll del brazo en el instante inicial
q_A0_r = q_A_r(1);

% Cuaternión de error vinculado al movimiento roll del brazo en el instante
inicial
q_A0_re = cos(phi_error_A/2)*q1 + sin(phi_error_A/2)*q2;

```

$${}^{GB}q'_{A0} = \left(q_{A0_{y_e}} \otimes q_{A0_y} \right) \otimes \left(q_{A0_p} \otimes q_{A0_{p_e}} \right) \otimes \left(q_{A0_r} \otimes q_{A0_{r_e}} \right)$$

```

% GB_qprima_A0
% A continuación, se calcula el cuaternión GB_qprima_A0; mediante la
concatenación de operaciones de tipo quatmultiply:
GB_qprima_A0 = quatmultiply(quatmultiply(q_A0_je,q_A0_jy),
quatmultiply(q_A0_p,q_A0_pe), quatmultiply(q_A0_r,q_A0_re))

% ----- ANTEBRAZO -----

% Cálculo de los Ángulos de Euler intrínsecos al cuaternión de orientación
GS_q_F. El segundo parámetro rige la codificación
% de ángulos empleada durante la conversión. Esta información se encuentra
vinculada al IMU colocado en el antebrazo:
AngulosDeEuler_q_F = quat2eul(GS_q_F, 'ZYX');

% A continuación, se crean tres vectores columna inherentes a sendos ángulos
de Euler (chi_F, theta_F, phi_F):
chi_F = AngulosDeEuler_q_F(:, 3);
theta_F = AngulosDeEuler_q_F(:, 2);
phi_F = AngulosDeEuler_q_F(:, 1);

% Definición de los ángulos de error simulados:
chi_error_F = 15 * pi/180;
theta_error_F = 0 * pi/180;
phi_error_F = 0 * pi/180;

% Cuaternión de error asociado al movimiento yaw del antebrazo, en el instante
de tiempo inicial
q_F0_je = cos(chi_error_F/2)*q1 + sin(chi_error_F/2)*q4;

```

```

% Cuaternión asociado al movimiento yaw del antebrazo, durante un tiempo
genérico t
q_F_y = cos(chi_F/2)*q1 + sin(chi_F/2)*q4;

% Cuaternión asociado al movimiento yaw del antebrazo, en el instante de
tiempo inicial
q_F0_y = q_F_y(1);

% Cuaternión correspondiente al movimiento de tipo pitch del antebrazo
q_F_p = cos(theta_F/2)*q1 + sin(theta_F/2)*q3;

% Cuaternión ligado al movimiento pitch del antebrazo en el instante inicial
q_F0_p = q_F_p(1);

% Cuaternión de error vinculado al movimiento pitch del antebrazo en el
instante inicial
q_F0_pe = cos(theta_error_F/2)*q1 + sin(theta_error_F/2)*q3;

% Cuaternión correspondiente al movimiento de tipo roll del antebrazo
q_F_r = cos(phi_F/2)*q1 + sin(phi_F/2)*q2;

% Cuaternión ligado al movimiento roll del antebrazo en el instante inicial
q_F0_r = q_F_r(1);

% Cuaternión de error vinculado al movimiento roll del antebrazo en el instante
inicial
q_F0_re = cos(phi_error_F/2)*q1 + sin(phi_error_F/2)*q2;

```

$${}^{GB}q'_{F0} = \left(q_{F0_{y_e}} \otimes q_{F0_y} \right) \otimes \left(q_{F0_p} \otimes q_{F0_{p_e}} \right) \otimes \left(q_{F0_r} \otimes q_{F0_{r_e}} \right)$$

```

% GB_qprima_F0

% A continuación, se calcula el cuaternión GB_qprima_F0; mediante la
concatenación de operaciones de tipo quatmultiply:
GB_qprima_F0 = quatmultiply(quatmultiply(q_F0_je, q_F0_j),
quatmultiply(q_F0_p, q_F0_pe), quatmultiply(q_F0_r, q_F0_re));

% Valor del cuaternión de orientación GS_q_A en el instante de tiempo inicial
GS_q_A0 = GS_q_A(1);

% Valor del cuaternión de orientación GS_q_F en el instante de tiempo inicial
GS_q_F0 = GS_q_F(1);

```

$${}^B q_{AF_{est1}} = ({}^{GS} q_A \otimes ({}^{GS} q_{A0}^* \otimes {}^{GB} q'_{A0}))^* \otimes ({}^{GS} q_F \otimes ({}^{GS} q_{F0}^* \otimes {}^{GB} q'_{F0}))$$

```

% B_q_AF_est1

% En las siguientes líneas de código, se crean 3 variables auxiliares;
necesarias durante el cálculo del cuaternión
% final de orientación B_q_AF_est1

% Cuaternión1 auxiliar
producto1 = {};

% Conjugado del Cuaternión1 auxiliar
producto1_conjugado = {};

% Cuaternión2 auxiliar
producto2 = {};

% Para i desde 1 hasta length(GS_q_A) --> HACER:
for i = 1:length(GS_q_A)
    % Cómputo del primer factor presente en el producto final de cuaterniones
    producto1{i} = quatmultiply(GS_q_A(i),
    quatmultiply(quatconj(GS_q_A0),GB_qprima_A0));
    % Cálculo del conjugado asociado al cuaternión obtenido con anterioridad
    producto1_conjugado{i} = quatconj(producto1{i});
    % Fin del bucle for i = 1:length(GS_q_A)
end

% Para i desde 1 hasta length(GS_q_F) --> HACER:
for i = 1:length(GS_q_F)
    % Cómputo del segundo factor presente en el producto final de cuaterniones
    producto2{i} = quatmultiply(GS_q_F(i),
    quatmultiply(quatconj(GS_q_F0),GB_qprima_F0));
    % Fin del bucle for i = 1:length(GS_q_F)
end

% Contabilización del número de cuaterniones almacenados en el vector
producto1
m1 = length(producto1);

% Contabilización del número de cuaterniones almacenados en el vector
producto2
m2 = length(producto2);

% Identificación del valor mínimo entre m1 y m2 --> Número mínimo de
cuaterniones

```

```

% que intervienen en los cálculos de orientación
m = min(m1, m2);

% Inicialización, como un vector vacío, del cuaternión B_q_AF_est1
B_q_AF_est1 = {};

% Inicialización, como un vector vacío, de los ángulos de euler a calcular
(AngulosDeEuler_est1)
AngulosDeEuler_est1 = {};

% Para i desde 1 hasta m --> HACER:
for i = 1:m
    % Cálculo del cuaternión de orientación i-ésimo que debe almacenarse en
    la variable B_q_AF_est1
    B_q_AF_est1{i} = quatmultiply(producto1_conjugado{i}, producto2{i});
    % Codificación de dicho cuaternión en forma de Ángulos de Euler
    AngulosDeEuler_est1{i} = quat2eul(B_q_AF_est1{i}, 'ZYX');
    % Fin del bucle for i = 1:m
end

% A continuación, se muestran por pantalla los 3 ángulos de Euler calculados
en cada iteración del algoritmo:
AngulosDeEuler_est1
% Inicialización, como un vector vacío, de los ángulos de flexión del codo
elbow_flexion_grados = [];

% Para i desde 1 hasta length(AngulosDeEuler_est1) --> HACER:
for i = 1:length(AngulosDeEuler_est1)
    % Obtención de los 3 Ángulos de Euler en la iteración i del algoritmo
    AngulosDeEuler_est1_i = AngulosDeEuler_est1{i};
    % Ángulo de flexión del codo, expresado en grados
    elbow_flexion_grados(i) = 180/pi * AngulosDeEuler_est1_i(3);
    % Ángulo de abducción del codo, expresado en grados
    elbow_abduction_grados(i) = 180/pi * AngulosDeEuler_est1_i(2);
    % Ángulo de rotación interna del codo, expresado en grados
    elbow_internal_rotation_grados(i) = 180/pi * AngulosDeEuler_est1_i(1);
    % Fin del bucle for i = 1:length(AngulosDeEuler_est1)
end

% corrección del offset de error (constante)
elbow_abduction_grados = elbow_abduction_grados - 59.9187;

% ----- REPRESENTACIONES GRÁFICAS -----

% Creación de una nueva figura
figure

```

```
% Activación de la primera casilla en la matriz de figuras creada
subplot(311)

% Representación gráfica del parámetro elbow_flexion_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_flexion_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow flexion', '[degrees]'})

% Activación de la segunda casilla en la matriz de figuras creada
subplot(312)

% Representación gráfica del parámetro elbow_abduction_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_abduction_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow abduction', '[degrees]'})

% Mediante las siguientes líneas de código se añade una etiqueta que indica
% el último ángulo estimado durante el experimento (52.1 [°],
aproximadamente):
ax2 = gca;
chart2 = ax2.Children(1);
datatip(chart2,941,-50.32);

% Activación de la tercera casilla en la matriz de figuras creada
subplot(313)

% Representación gráfica del parámetro elbow_internal_rotation_grados. Se
utiliza un ancho de línea de 1.5 [puntos]
```

```
plot(elbow_internal_rotation_grados, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow internal', 'rotation [degrees]'})
```

18.5.1.- Cálculo y representación de la señal de error durante la estimación del ángulo de abducción del codo

```
% Constante auxiliar definida para representar gráficamente las señales que
se desean analizar
k=2.8;

% Número de muestras digitales consideradas para el primer intervalo
t1 = 42:68;

% Primer valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y1 = (-30.1+k) * ones(1, length(t1));

% Cálculo del error de estimación correspondiente al primer intervalo temporal
del experimento
error1 = (y1' - elbow_abduction_grados(t1));

% Número de muestras digitales consideradas para el segundo intervalo
t2 = 78:190;

% Segundo valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y2 = (-17.2+k) * ones(1, length(t2));

% Cálculo del error de estimación correspondiente al segundo intervalo
temporal del experimento
error2 = (y2' - elbow_abduction_grados(t2));

% Número de muestras digitales consideradas para el tercer intervalo
t3 = 196:322;
```

```
% Tercer valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y3 = (-29.5+k) * ones(1, length(t3));

% Cálculo del error de estimación correspondiente al tercer intervalo temporal
del experimento
error3 = (y3' - elbow_abduction_grados(t3));

% Número de muestras digitales consideradas para el cuarto intervalo
t4 = 328:458;

% Cuarto valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y4 = (-44.1+k) * ones(1, length(t4));

% Cálculo del error de estimación correspondiente al cuarto intervalo temporal
del experimento
error4 = (y4' - elbow_abduction_grados(t4));

% Número de muestras digitales consideradas para el quinto intervalo
t5 = 466:614;

% Quinto valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y5 = (-60.2+k) * ones(1, length(t5));

% Cálculo del error de estimación correspondiente al quinto intervalo temporal
del experimento
error5 = (y5' - elbow_abduction_grados(t5));

% Número de muestras digitales consideradas para el sexto intervalo
t6 = 623:750;

% Sexto valor de referencia adoptado para la estimación del ángulo de abducción
del codo
y6 = (-34.9+k) * ones(1, length(t6));

% Cálculo del error de estimación correspondiente al sexto intervalo temporal
del experimento
error6 = (y6' - elbow_abduction_grados(t6));

% Número de muestras digitales consideradas para el séptimo intervalo
t7 = 757:885;
```

```
% Séptimo valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y7 = (-14.3+k) * ones(1, length(t7));

% Cálculo del error de estimación correspondiente al séptimo intervalo
temporal del experimento
error7 = (y7' - elbow_abduction_grados(t7));

% Número de muestras digitales consideradas para el octavo intervalo
t8 = 893:1035;

% Octavo valor de referencia adoptado para la estimación del ángulo de
abducción del codo
y8 = (-52.1+k) * ones(1, length(t8));

% Cálculo del error de estimación correspondiente al octavo intervalo temporal
del experimento
error8 = (y8' - elbow_abduction_grados(t8));

% Creación del vector de tiempo final mediante la concatenación de vectores
t = [t1, t2, t3, t4, t5, t6, t7, t8];

% Creación del vector asociado al ángulo de abducción estimado, mediante la
concatenación de vectores
y = [y1, y2, y3, y4, y5, y6, y7, y8];

% Generación de la señal de error calculada durante la estimación del ángulo
de abducción. En este paso,
% se efectúa el ensamblaje de los 8 vectores columna de error creados
anteriormente.
error = [error1; error2; error3; error4; ...
         error5; error6; error7; error8];

% Creación de una nueva figura
figure

% Activación de la primera casilla en la matriz de figuras creada (dimensiones
2 x 1)
subplot(2,1,1)

% Representación gráfica del ángulo de referencia vinculado a la abducción
del codo. Se utiliza un
% ancho de línea de 1.5 [puntos]
plot(t, y, 'b-', 'LineWidth', 1.5)
```

```
% Se mantiene activa la figura actual, con objeto de seguir dibujando dentro
de ella.
hold on

% Representación gráfica del parámetro elbow_abduction_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_abduction_grados, 'r--', 'LineWidth', 1.5)

% Desactivación de la figura actual, puesto que no se van a incluir más
representaciones gráficas.
hold off

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Elbow abduction [degrees]')

% Incorporación de un título a la representación gráfica maquetada
title('Estimation error. Comparative')

% Creación de una leyenda para dicha figura
legend('Reference [°]', 'Sensor to segment error: 0 [°]', 'Location', 'north')

% Especificación de la posición asociada a la leyenda anterior
legend("Position", [0.26274,0.83814,0.34464,0.072619])

% Activación de la segunda casilla en la matriz de figuras creada (dimensiones
2 x 1)
subplot(2,1,2)

% Representación gráfica del error ligado a la estimación del ángulo de
abducción.
% Se utiliza un ancho de línea de 1.5 [puntos]:
plot(error, 'k-', 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')
```

```
% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'Estimate error in elbow', 'abduction [degrees]'})

% Incorporación de un título a la representación gráfica maquetada
title('Estimation error. Comparative')
```

18.6.- COMPARATIVA DE VARIOS CASOS DE ESTUDIO. OBSERVACIÓN DEL ERROR GENERADO DURANTE LA ESTIMACIÓN DE LA ROTACIÓN ARTICULAR

```
% Carga de la señal de referencia definida para el ángulo de abducción del
codo.
load('.\Seniales\ElbowAbduction_ref.mat')

% Carga de la señal vinculada al ángulo de abducción calculado, considerando
el siguiente
% error para la alineación segmento - sensor: 0 [grados].
X1 = load('.\Seniales\ElbowAbduction_0degrees_ERROR.mat');

% Carga de la señal inherente al error generado durante la estimación del
ángulo de abducción.
Y1 = load('.\Seniales\0degrees_ERROR.mat');

% Carga de la señal vinculada al ángulo de abducción calculado, considerando
el siguiente
% error para la alineación segmento - sensor: 3 [grados].
X2 = load('.\Seniales\ElbowAbduction_3degrees_ARM_ERROR.mat');

% Carga de la señal inherente al error generado durante la estimación del
ángulo de abducción.
Y2 = load('.\Seniales\3degrees_ARM_ERROR.mat');

% Carga de la señal vinculada al ángulo de abducción calculado, considerando
el siguiente
% error para la alineación segmento - sensor: -5 [grados].
X3 = load('.\Seniales\ElbowAbduction_-5degrees_ARM_ERROR.mat');

% Carga de la señal inherente al error generado durante la estimación del
ángulo de abducción.
Y3 = load('.\Seniales\_-5degrees_ARM_ERROR.mat');
```

```
% Carga de la señal vinculada al ángulo de abducción calculado, considerando
el siguiente
% error para la alineación segmento - sensor: 10 [grados].
X4 = load('.\Seniales\ElbowAbduction_10degrees_ARM_ERROR.mat');

% Carga de la señal inherente al error generado durante la estimación del
ángulo de abducción.
Y4 = load('.\Seniales\10degrees_ERROR.mat');

% Creación de una nueva figura.
figure

% Activación de la primera casilla en la matriz de figuras creada (dimensiones
2 x 1).
subplot(2,1,1)

% Representación gráfica del ángulo de referencia vinculado a la abducción
del codo.
% Se utiliza un ancho de línea de 1.5 [puntos].
plot(t, y, 'b-', 'LineWidth', 2)

% Se mantiene activa la figura actual, con objeto de seguir dibujando dentro
de ella.
hold on

% Representación gráfica del parámetro elbow_abduction_grados, el cual es un
campo de la estructura X1.
% Se utiliza un ancho de línea de 1.5 [puntos].
plot(X1.elbow_abduction_grados, 'r--', 'LineWidth', 1.5)

% Representación gráfica del parámetro elbow_abduction_grados, el cual es un
campo de la estructura X2.
% Se utiliza un ancho de línea de 1.5 [puntos].
plot(X2.elbow_abduction_grados, 'g--', 'LineWidth', 1.5)

% Representación gráfica del parámetro elbow_abduction_grados, el cual es un
campo de la estructura X3.
% Se utiliza un ancho de línea de 1.5 [puntos].
plot(X3.elbow_abduction_grados, 'c--', 'LineWidth', 1.5)

% Representación gráfica del parámetro elbow_abduction_grados, el cual es un
campo de la estructura X4.
% Se utiliza un ancho de línea de 1.5 [puntos].
plot(X4.elbow_abduction_grados, 'k--', 'LineWidth', 1.5)
```

```
% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel('Elbow abduction [degrees]')

% Incorporación de un título a la representación gráfica maquetada
title('Estimation error. Comparative')

% Desactivación de la figura actual, puesto que no se van a incluir más
representaciones gráficas.
hold off

% Creación de una leyenda para dicha figura
legend('Reference [°]', 'S to S Error: 0 [°]', 'S to S Error: 3 [°]', ...
       'S to S Error: -5 [°]', 'S to S Error: 10 [°]')

% Activación de la segunda casilla en la matriz de figuras creada (dimensiones
2 x 1)
subplot(2,1,2)

% Representación gráfica del error ligado a la estimación del ángulo de
abducción, el cual es un campo
% de la estructura Y1. Se utiliza un ancho de línea de 1.5 [puntos].
plot(Y1.error, 'r-', 'LineWidth', 1.5)

% Se mantiene activa la figura actual, con objeto de seguir dibujando dentro
de ella.
hold on

% Representación gráfica del error ligado a la estimación del ángulo de
abducción, el cual es un campo
% de la estructura Y2. Se utiliza un ancho de línea de 1.5 [puntos].
plot(Y2.error, 'g-', 'LineWidth', 1.5)

% Representación gráfica del error ligado a la estimación del ángulo de
abducción, el cual es un campo
% de la estructura Y3. Se utiliza un ancho de línea de 1.5 [puntos].
plot(Y3.error, 'c-', 'LineWidth', 1.5)

% Representación gráfica del error ligado a la estimación del ángulo de
abducción, el cual es un campo
% de la estructura Y4. Se utiliza un ancho de línea de 1.5 [puntos].
plot(Y4.error, 'k-', 'LineWidth', 1.5)
```

```
% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'Estimate error in elbow', 'abduction [degrees]'})

% Creación de una leyenda para dicha figura
legend('S to S Error: 0 [°]', 'S to S Error: 3 [°]', ...
       'S to S Error: -5 [°]', 'S to S Error: 10 [°]')

% Desactivación de la figura actual, puesto que no se van a incluir más
representaciones gráficas.
hold off
```

18.7.- INCORPORACIÓN DE LOS ERRORES DE ORIENTACIÓN DEL *IMU*

Estudio sobre el efecto de los errores de orientación del *IMU* en la estimación 3D final (movimiento del codo):

18.7.1.- ${}^{GS}q'_A$

$${}^{GS}q'_A = \left(q_{A_{y_e}} \otimes q_{A_y} \right) \otimes \left(q_{A_p} \otimes q_{A_{p_e}} \right) \otimes \left(q_{A_r} \otimes q_{A_{r_e}} \right)$$

```
% ----- BRAZO -----
% Definición de los ángulos de error simulados (asociados a la orientación
del IMU ubicado en el brazo):
chi_error_A_orientation_IMU = 0 * pi/180;
theta_error_A_orientation_IMU = 0 * pi/180;
phi_error_A_orientation_IMU = 0 * pi/180;

% Construcción de los cuaterniones involucrados en el cálculo de GS_qprima_A:
% Cuaternión de error asociado al movimiento yaw del brazo, en un de tiempo
genérico
q_A_ye = cos(chi_error_A_orientation_IMU/2)*q1 +
sin(chi_error_A_orientation_IMU/2)*q4;
```

```

% Cuaternión asociado al movimiento yaw del brazo, durante un tiempo genérico
t
q_A_y;

% Cuaternión correspondiente al movimiento de tipo pitch del brazo
q_A_p;

% Cuaternión de error vinculado al movimiento pitch del brazo en el instante
de tiempo t
q_A_pe = cos(theta_error_A_orientation_IMU/2)*q1 +
sin(theta_error_A_orientation_IMU/2)*q3;

% Cuaternión correspondiente al movimiento de tipo roll del brazo
q_A_r;

% Cuaternión de error vinculado al movimiento roll del brazo en el instante
de tiempo t
q_A_re = cos(phi_error_A_orientation_IMU/2)*q1 +
sin(phi_error_A_orientation_IMU/2)*q2;

% GS_qprima_A
% A continuación, se calcula el cuaternión GS_qprima_A; mediante la
concatenación de operaciones de tipo quatmultiply:
GS_qprima_A = quatmultiply(quatmultiply(q_A_ye, q_A_y), quatmultiply(q_A_p,
q_A_pe), quatmultiply(q_A_r, q_A_re))

```

18.7.2.- ${}^{GS}q'_F$

$${}^{GS}q'_F = \left(q_{F_{y_e}} \otimes q_{F_y} \right) \otimes \left(q_{F_p} \otimes q_{F_{p_e}} \right) \otimes \left(q_{F_r} \otimes q_{F_{r_e}} \right)$$

```

% ----- ANTEBRAZO -----
% Definición de los ángulos de error simulados (asociados a la orientación
del IMU ubicado en el antebrazo):
chi_error_F_orientation_IMU = 0 * pi/180;
theta_error_F_orientation_IMU = 0 * pi/180;
phi_error_F_orientation_IMU = 0 * pi/180;

% Cuaternión de error asociado al movimiento yaw del antebrazo, en el instante
de tiempo t

```

```

q_F_ye = cos(chi_error_F_orientation_IMU/2)*q1 +
sin(chi_error_F_orientation_IMU/2)*q4;

% Cuaternión asociado al movimiento yaw del antebrazo, durante un tiempo
genérico t
q_F_y;

% Cuaternión correspondiente al movimiento de tipo pitch del antebrazo
q_F_p;

% Cuaternión de error vinculado al movimiento pitch del antebrazo en el
instante de tiempo t
q_F_pe = cos(theta_error_F_orientation_IMU/2)*q1 +
sin(theta_error_F_orientation_IMU/2)*q3;

% Cuaternión correspondiente al movimiento de tipo roll del antebrazo
q_F_r;

% Cuaternión de error vinculado al movimiento roll del antebrazo en el instante
de tiempo t
q_F_re = cos(phi_error_F_orientation_IMU/2)*q1 +
sin(phi_error_F_orientation_IMU/2)*q2;

% GS_qprima_F
% A continuación, se calcula el cuaternión GS_qprima_F; mediante la
concatenación de operaciones de tipo quatmultiply:
GS_qprima_F = quatmultiply(quatmultiply(q_F_ye, q_F_y), quatmultiply(q_F_p,
q_F_pe), quatmultiply(q_F_r, q_F_re))

```

18.7.3.- ${}^B q_{AF_{est2}}$

$${}^B q_{AF_{est2}} = ({}^{GS} q'_A \otimes ({}^{GS} q_{A0}^* \otimes {}^{GB} q_{A0}))^* \otimes ({}^{GS} q'_F \otimes ({}^{GS} q_{F0}^* \otimes {}^{GB} q_{F0}))$$

```

% Mediante las siguientes 4 líneas de código, se obtienen varios cuaterniones
auxiliares involucrados durante el cálculo de B_q_AF_est2:
GS_q_A0 = GS_q_A(1);
GB_q_A0 = GB_q_A(1);
GS_q_F0 = GS_q_F(1);
GB_q_F0 = GB_q_F(1);

```

```
% Cómputo del primer factor asociado al cálculo del cuaternión final de
orientación B_q_AF_est2
factor1 = quatmultiply(GS_qprima_A,
quatmultiply(quatconj(GS_q_A0),GB_q_A0));

% Cálculo del conjugado ligado al cuaternión anterior
factor1_conjugado = quatconj(factor1);

% Cómputo del segundo factor asociado al cálculo del cuaternión final de
orientación B_q_AF_est2
factor2 = quatmultiply(GS_qprima_F,
quatmultiply(quatconj(GS_q_F0),GB_q_F0));

% Cuantificación del número de cuaterniones almacenados en la variable factor1
z1 = length(factor1);

% Cuantificación del número de cuaterniones almacenados en la variable factor2
z2 = length(factor2);

% Identificación del valor mínimo entre z1 y z2 --> Número mínimo de
cuaterniones
% que intervienen en los cálculos de orientación final (cuaternión
B_q_AF_est2)
z = min(z1, z2);

% B_q_AF_est2
% A continuación, se calcula el cuaternión B_q_AF_est2; mediante la operación
quatmultiply:
B_q_AF_est2 = quatmultiply(factor1_conjugado(1:z), factor2(1:z))
```

18.7.4.- Obtención de los *Ángulos de Euler* asociados al cuaternión ${}^B q_{AF_{est2}}$

```
% Cálculo de los Ángulos de Euler intrínsecos al cuaternión de orientación
B_q_AF_est2. El segundo parámetro rige la codificación de ángulos empleada
durante la conversión:
AngulosDeEuler_est2 = quat2eul(B_q_AF_est2, 'ZYX');

% Inicialización, como un vector vacío, de los ángulos de flexión del codo
elbow_flexion_grados_est2 = [];

% Inicialización, como un vector vacío, de los ángulos de abducción del codo
elbow_abduction_grados_est2 = [];
```

```
% Inicialización, como un vector vacío, de los ángulos de rotación interna
del codo
elbow_internal_rotation_grados_est2 = [];

% Para i desde 1 hasta length(AngulosDeEuler_est1) --> HACER:
for i = 1:length(AngulosDeEuler_est2)
    % Obtención de los 3 Ángulos de Euler en la iteración i del algoritmo
    AngulosDeEuler_est2_i = AngulosDeEuler_est2(i,:);
    % Ángulo de flexión del codo, expresado en grados
    elbow_flexion_grados_est2(i) = 180/pi * AngulosDeEuler_est2_i(3);
    % Ángulo de abducción del codo, expresado en grados
    elbow_abduction_grados_est2(i) = 180/pi * AngulosDeEuler_est2_i(2);
    % Ángulo de rotación interna del codo, expresado en grados
    elbow_internal_rotation_grados_est2(i) = 180/pi *
AngulosDeEuler_est2_i(1);
    % Fin del bucle for i = 1:length(AngulosDeEuler_est1)
end

% --- REPRESENTACIONES GRÁFICAS ---

% Creación de una nueva figura
figure

% Activación de la primera casilla en la matriz de figuras creada
subplot(311)

% Representación gráfica del parámetro elbow_flexion_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_flexion_grados_est2, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow flexion', '[degrees]'})

% Incorporación de un título a la figura maquetada
title('Introduction of the orientation error associated with the IMU')

% Activación de la segunda casilla en la matriz de figuras creada
subplot(312)
```

```
% Representación gráfica del parámetro elbow_abduction_grados. Se utiliza un
ancho de línea de 1.5 [puntos]
plot(elbow_abduction_grados_est2, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta en el eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta en el eje Y de la gráfica
ylabel({'elbow abduction', '[degrees]'})

% Mediante las siguientes líneas de código se añade una etiqueta que indica
% el último ángulo estimado durante el experimento (52.1 [°],
aproximadamente):
ax3 = gca;
chart3 = ax3.Children(1);
datatip(chart3,943,54.74);

% Activación de la tercera casilla en la matriz de figuras creada
subplot(313)

% Representación gráfica del parámetro elbow_internal_rotation_grados. Se
utiliza un ancho de línea de 1.5 [puntos]
plot(elbow_internal_rotation_grados_est2, 'LineWidth', 1.5)

% Activación de una rejilla en la gráfica creada
grid on

% Inclusión de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Inclusión de una etiqueta al eje Y de la gráfica
ylabel({'elbow internal', 'rotation [degrees]'})
```

18.8.- VALORACIÓN FINAL DE LOS ERRORES DE ESTIMACIÓN DE LA ORIENTACIÓN

18.8.1.- ${}^B q_{AF_{ref}}$

$${}^B q_{AF_{ref}} = ({}^{GS} q_A \otimes ({}^{GS} q_{A0}^* \otimes {}^{GB} q_{A0}))^* \otimes ({}^{GS} q_F \otimes ({}^{GS} q_{F0}^* \otimes {}^{GB} q_{F0}))$$

```

% Cómputo del primer factor involucrado en el producto global de cuaterniones.
El resultado final
% de dicho producto es el cuaternión B_q_AF_ref
p1 = quatmultiply(GS_q_A, quatmultiply(quatconj(GS_q_A0),GB_q_A0));

% Cálculo del conjugado asociado al cuaternión p1
p1_conjugado = quatconj(p1);

% Cómputo del segundo factor involucrado en el producto global de cuaterniones
p2 = quatmultiply(GS_q_F, quatmultiply(quatconj(GS_q_F0),GB_q_F0));

% Cuantificación del número de cuaterniones almacenados en la variable p1
z1 = length(p1);

% Cuantificación del número de cuaterniones almacenados en la variable p2
z2 = length(p2);

% Identificación del valor mínimo entre z1 y z2 --> Número mínimo de
cuaterniones
% que intervienen en los cálculos de orientación final (cuaternión B_q_AF_ref)
z = min(z1, z2);

% Cálculo del cuaternión B_q_AF_ref, mediante el producto de p1_conjugado y
p2
B_q_AF_ref = quatmultiply(p1_conjugado(1:z), p2(1:z))

% Obtención de los tres Ángulos de Euler intrínsecos al cuaternión B_q_AF_ref
AngulosDeEuler_referencia = quat2eul(B_q_AF_ref, 'ZYX')

% Ángulo de flexión de la articulación
elbow_flexion_grados_ref = 180/pi * AngulosDeEuler_referencia(:,3);

% Ángulo de abducción
elbow_abduction_grados_ref = 180/pi * AngulosDeEuler_referencia(:,2);

% Ángulo de rotación interna
elbow_internal_rotation_grados_ref = 180/pi * AngulosDeEuler_referencia(:,1);

```

18.8.2.- ϕ_{ERROR}

$$\phi_{ERROR} = \phi_{qAF_{est}} - \phi_{qAF_{ref}}$$

```
% Error asociado al ángulo de rotación interna
PHI_ERROR = -elbow_internal_rotation_grados - y(:,3)
```

18.8.3.- θ_{ERROR}

$$\theta_{ERROR} = \theta_{qAF_{est}} - \theta_{qAF_{ref}}$$

```
% Error vinculado al ángulo de abducción
THETA_ERROR = error
```

18.8.4.- ψ_{ERROR}

$$\psi_{ERROR} = \psi_{qAF_{est}} - \psi_{qAF_{ref}}$$

```
% Error ligado al ángulo de flexión
CHI_ERROR = elbow_flexion_grados' - y(:,3)

% Creación de una nueva figura
figure

% Activación de la primera figura, existente en la matriz de
% dimensiones 3 x 1 previamente creada
subplot(3,1,1)

% Representación gráfica de la variable PHI_ERROR. Se utiliza color
% rojo y trazo continuo. El ancho de línea empleado es de 1.5 [puntos]
plot(PHI_ERROR, 'r-', 'LineWidth', 1.5)

% Incorporación de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Incorporación de una etiqueta al eje Y de la gráfica
ylabel('\$\phi_e$', 'Interpreter', 'latex', 'FontSize', 14)

% Activación de la rejilla en dicha figura
grid on
```

```
% Definición del título vinculado a la representación gráfica
% maquetada anteriormente
title('Final error comparison', 'FontSize', 15)

% Activación de la segunda figura, existente en la matriz de
% dimensiones 3 x 1 previamente creada
subplot(3,1,2)

% Representación gráfica de la variable THETA_ERROR. Se utiliza color
% verde y trazo continuo. El ancho de línea empleado es de 1.5 [puntos]
plot(THETA_ERROR, 'g-', 'LineWidth', 1.5)

% Incorporación de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Incorporación de una etiqueta al eje Y de la gráfica
ylabel('\theta_e$', 'Interpreter', 'latex', 'FontSize', 14)

% Activación de la rejilla en dicha figura
grid on

% Activación de la tercera figura, existente en la matriz de
% dimensiones 3 x 1 previamente creada
subplot(3,1,3)

% Representación gráfica de la variable CHI_ERROR. Se utiliza color
% azul y trazo continuo. El ancho de línea empleado es de 1.5 [puntos]
plot(CHI_ERROR, 'b-', 'LineWidth', 1.5)

% Incorporación de una etiqueta al eje X de la gráfica
xlabel('Sample [-]')

% Incorporación de una etiqueta al eje Y de la gráfica
ylabel('\chi_e$', 'Interpreter', 'latex', 'FontSize', 14)

% Activación de la rejilla en dicha figura
grid on
```

CAPÍTULO 2.

PRESUPUESTO

ÍNDICE DEL PRESUPUESTO

1. COSTES DE EJECUCIÓN MATERIAL	242
1.1.- COSTES DE MATERIAL SOFTWARE.....	242
1.2.- COSTES DE EQUIPOS INFORMÁTICOS	244
1.3.- COSTES DE MANO DE OBRA.....	245
1.4.- COSTES DE LOS ELEMENTOS HARDWARE SELECCIONADOS	247
1.5.- OTROS CONCEPTOS	248
1.6.- COSTE TOTAL DEL PRESUPUESTO DE EJECUCIÓN MATERIAL (PEM) .	249
2. GASTOS GENERALES Y BENEFICIO INDUSTRIAL	251
3. COSTE TOTAL DEL PROYECTO (PRESUPUESTO DE EJECUCIÓN POR CONTRATA – PEC).....	252

ÍNDICE DE TABLAS

<i>Tabla 1.1.- Costes derivados de las herramientas software seleccionadas para realizar este Trabajo Fin de Máster.....</i>	<i>243</i>
<i>Tabla 1.2.- Coste propiciado por las herramientas hardware informáticas que se han utilizado para realizar este Trabajo Fin de Máster.....</i>	<i>245</i>
<i>Tabla 1.3.- Coste asociado a la mano de obra empleada en el desarrollo de este proyecto.....</i>	<i>245</i>
<i>Tabla 1.4.- Coste vinculado a los dispositivos hardware empleados durante este TFM. Dicho material físico se ha utilizado en la metodología experimental del proyecto, destinada a la adquisición de datos inerciales.....</i>	<i>247</i>
<i>Tabla 1.5.- Costes derivados de la partida Otros Conceptos, los cuales no habían sido contemplados en secciones anteriores del actual presupuesto.....</i>	<i>249</i>
<i>Tabla 1.6.- Coste final ligado al PEM (Presupuesto de Ejecución Material), el cual se ha calculado mediante la adición de todas las partidas presupuestarias indicadas en secciones anteriores.....</i>	<i>250</i>
<i>Tabla 2.1.- Costes inherentes al Beneficio Industrial y a los Gastos Generales definidos en el presente Trabajo Fin de Máster.....</i>	<i>251</i>
<i>Tabla 3.1.- Coste total del proyecto – Presupuesto de Ejecución por Contrata (PEC).....</i>	<i>252</i>

1. Costes de ejecución material

En el presente apartado se desglosa el procedimiento de cálculo seguido durante la elaboración del **Presupuesto de Ejecución Material (PEM)**, asociado a este *Trabajo Fin de Máster*. Para ello, se ha considerado que las necesidades económicas del proyecto emanan de cuatro posibles sectores: **herramientas software**, **dispositivos informáticos**, **mano de obra**⁴¹ y otros **materiales hardware**. Asimismo, se ha reservado un apartado adicional para los costes derivados de otros conceptos.

En combinación con lo expresado anteriormente, resulta preciso reseñar que durante la elaboración de este presupuesto se ha contemplado la vida útil de cada recurso utilizado en el proyecto. Asimismo, el coste asociado a cada elemento del *PEM* también tiene en cuenta las necesidades horarias impuestas por el *Trabajo Fin de Máster*. Obsérvese que se ha considerado interesante valorar la posible reutilización de equipos y programas *software* durante las líneas de trabajo futuras; ya propuestas anteriormente para este proyecto.

De acuerdo con la introducción comentada en los dos párrafos anteriores, el coste incurrido por cada ítem considerado será función de la duración global del proyecto y la vida útil de dicho recurso.

1.1.- COSTES DE MATERIAL SOFTWARE

Inicialmente, se recogen los costes vinculados a las licencias *software* necesarias para desarrollar la metodología experimental del *TFM*. Durante la presupuestación realizada, se ha considerado que el presente *Trabajo Fin de Máster* posee una duración temporal⁴² de 5 [meses]; aproximadamente.

⁴¹ **Nota:** En la sección destinada a la mano de obra, se ha contabilizado la dedicación horaria de los recursos humanos; necesarios durante el correcto desarrollo del *TFM*.

⁴² A través del apartado “7. Planificación temporal”, puede consultarse un desglose sobre los plazos de tiempo establecidos para el *TFM*. La fecha de inicio del proyecto ha sido el miércoles 01/02/23 y su finalización ha tenido lugar el lunes 12/06/23.

Por otra parte, dado que las herramientas *software* empleadas en este *TFM* pueden reutilizarse en futuros proyectos, se ha considerado relevante tener presente la vida útil de dichos activos. En este caso, se ha tomado una duración de 4 [años] (= 48 [meses]) para el ciclo de vida correspondiente a los materiales *software* elegidos.

Tabla 1.1.- Costes derivados de las herramientas *software* seleccionadas para realizar este *Trabajo Fin de Máster*.

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
Sistema operativo <i>Windows: Windows 11 Professional</i>	259 €	10,42 %	26,99 €
<i>Matlab R2022b (Academic)</i>	262 €/año	10,42 % (durante 5 meses ⁴³)	11,47 €
<i>Microsoft Office 365 (Empresa Estándar)</i>	11,70 €/mes	5 meses	58,50 €
Aplicación oficial (<i>Android</i>) de <i>Xsens DOT</i>	Gratuita	5 meses	--
Aplicación oficial (<i>Android</i>) de los sensores <i>MATRIX</i>	Gratuita	5 meses	--
Programa de simulación de robots <i>RoboDK (Educational)</i>	72,50 €/año	5 meses (= 0,42 años)	30,45 €
<i>Software</i> de captura del movimiento <i>Motive:Body (OptiTrack)</i>	4.556,99 €	10,42 %	474,84 €

Subtotal: **602,25 €**

⁴³ Téngase presente que, en el cálculo del coste vinculado al ítem *Matlab R2022b (Academic)*, se ha utilizado la siguiente equivalencia temporal:

$$5 \text{ [meses]} = \frac{5}{12} \text{ [años]} \cong 0,42 \text{ [años]}$$

En la tabla anterior se ha incluido una columna que indica la *cantidad* necesaria de cada concepto definido. De acuerdo con los datos asociados al presente caso de estudio, se verifica que la cantidad necesaria para cada ítem recogido en la Tabla 1.1 asciende⁴⁴ a 10,42 [%]. Dicho resultado numérico responde a la ecuación (18.1), teniendo presente que la duración del proyecto es 5 [meses] y la vida útil de cada concepto se corresponde con 48 [meses]:

$$CANTIDAD_{CONCEPTO} [\%] = \frac{Duración\ del\ proyecto\ [meses]}{Vida\ útil\ del\ concepto\ [meses]} \cdot 100 \quad (18.1)$$

En combinación con lo expresado durante el párrafo anterior, obsérvese que el apartado *Cantidad* de la Tabla 1.1 también incorpora la duración temporal del presente *TFM*.

Como conclusión a la información expuesta con anterioridad, se tiene que los costes intrínsecos a la adquisición de materiales *software* ascienden a 602,25 €.

1.2.- COSTES DE EQUIPOS INFORMÁTICOS

El siguiente apartado, incluido en el *Presupuesto de Ejecución Material*, se vincula a la utilización de los equipos informáticos (dispositivos *hardware*) durante el desarrollo del *Trabajo Fin de Máster*.

En esta partida del presupuesto, se ha considerado que el ciclo de vida asociado a los equipos informáticos asciende a 5 [años] (= 60 [meses]). Análogamente al caso anterior, la duración total del proyecto se ha cuantificado en 5 [meses].

Al igual que en la sección destinada a la presupuestación de las herramientas *software*, en este caso resulta preciso calcular la cantidad necesaria para el concepto

⁴⁴ El porcentaje de 10,42 [%] se ha obtenido realizando la siguiente operación:

$$CANTIDAD_{CONCEPTO} [\%] = \frac{5 [meses]}{48 [meses]} \cdot 100 = 10,42 [\%]$$

referenciado en la Tabla 1.2. En este escenario, se ha realizado la siguiente estimación temporal:

$$CANTIDAD_{CONCEPTO} [\%] = \frac{5 [meses]}{5 [años] \cdot 12 \left[\frac{meses}{año} \right]} \cdot 100 = 8,33 [\%] \quad (18.2)$$

Tabla 1.2.- Coste propiciado por las herramientas *hardware* informáticas que se han utilizado para realizar este *Trabajo Fin de Máster*.

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
PC (Ordenador portátil)	1.300 €	8,33 %	108,29 €

Subtotal: **108,29 €**

En consonancia con los datos indicados anteriormente, el coste asociado a los equipos informáticos considerados toma el valor de 108,29 €.

1.3.- COSTES DE MANO DE OBRA

El coste unitario (horario) que se asocia a la mano de obra de ingeniería se ha establecido en 6 €/h. El calendario laboral definido para efectuar el *TFM* se corresponde con una carga horaria de 5 horas al día, asignada 5 días a la semana.

Tabla 1.3.- Coste asociado a la mano de obra empleada en el desarrollo de este proyecto.

CONCEPTO	HORAS DE TRABAJO	COSTE HORARIO	SUBTOTAL
Estudios previos, análisis preliminares.	65 horas	6 €/h	390 €

Análisis exhaustivo del problema que se debe resolver.	60 horas	6 €/h	360 €
Descripción matemática de los algoritmos diseñados.	70 horas	6 €/h	420 €
Programación en <i>MatLab</i> de los algoritmos propuestos.	95 horas	6 €/h	570 €
Ejecución de la etapa de <i>test</i> mediante datos sintéticos (manuales).	65 horas	6 €/h	390 €
Desarrollo experimental con personas. Realización de pruebas biomecánicas, en sujetos humanos, para los algoritmos implementados.	55 horas	6 €/h	330 €
Desarrollo, maquetación y revisión de los documentos asociados al presente <i>Trabajo Fin de Máster</i> .	80 horas	6 €/h	480 €
Construcción de la presentación que se deriva del trabajo desarrollado durante el <i>TFM</i> .	10 horas	6 €/h	60 €

Subtotal: ... 500 horas 3.000 €

De acuerdo con la información recogida en la Tabla 1.3, el coste económico vinculado a la mano de obra de ingeniería toma el valor de 3.000 €.

Nota: Obsérvese que la dedicación horaria de la mano de obra, presupuestada en este apartado, ofrece concordancia con la planificación temporal previamente definida. Véase la sección "8.1.- *DIAGRAMA DE GANTT* ASOCIADO AL PROYECTO" de la presente *Memoria*.

1.4.- COSTES DE LOS ELEMENTOS *HARDWARE* SELECCIONADOS

A continuación, se realiza un desglose del presupuesto vinculado a los dispositivos *hardware* escogidos; los cuales se han utilizado durante la implementación física de este proyecto. Téngase en cuenta que cada uno de los conceptos, referenciados en la presente sección, se corresponden con los ítems *hardware* mostrados previamente mediante el apartado "4.2.- Material *hardware* empleado. Resumen de características técnicas".

Cabe resaltar que la vida útil definida para los sensores inerciales presupuestados en la "Tabla 1.4" (*Xsens DOT*, *MATRIX* y *SuperSens*) es de 6 [años], respectivamente. En consecuencia, la cantidad porcentual asignada a dichos activos se ha calculado de la manera que sigue:

$$CANTIDAD_{\text{Sensores Inerciales}} [\%] = \frac{5 [\text{meses}]}{6 [\text{años}] \cdot 12 \left[\frac{\text{meses}}{\text{año}} \right]} \cdot 100 = 6,94 [\%] \quad (18.3)$$

Tabla 1.4.- Coste vinculado a los dispositivos *hardware* empleados durante este *TFM*.
Dicho material físico se ha utilizado en la metodología experimental del proyecto, destinada a la adquisición de datos inerciales.

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
Robot Colaborativo ⁴⁵ <i>UR3e</i>	20.670,10 €	1 unidad (2,78 [%])	574,63 €
Unidades de medida inercial <i>Xsens DOT</i>	140 €	2 unidades (6,94 [%])	19,43 €

⁴⁵ Al igual que en los análisis de costes realizados con anterioridad, se ha efectuado una estimación porcentual sobre la cantidad necesaria del concepto "Robot Colaborativo *UR3e*". Para ello, se ha hecho uso de la siguiente expresión:

$$CANTIDAD_{\text{Cobot } UR3e} [\%] = \frac{5 [\text{meses}]}{15 [\text{años}] \cdot 12 \left[\frac{\text{meses}}{\text{año}} \right]} \cdot 100 = 2,78 [\%]$$

En la ecuación anterior, puede observarse que la vida útil consustancial al robot colaborativo *UR3e* se ha estimado en 15 [años].

Sensores inerciales <i>MATRIX</i>	50 €	2 unidades (6,94 [%])	6,94 €
<i>IMU</i> desarrollado por el grupo de investigación <i>SiMuR: SuperSens</i>	90 €	2 unidades (6,94 [%])	12,50 €
Sistema de captura del movimiento – <i>MOCAP</i> (<i>OptiTrack</i> ⁴⁶)	987,67 €	1 unidad (4,17 [%])	41,19 €

Subtotal: **654,69 €**

Por tanto, el coste intrínseco a los elementos *hardware* seleccionados asciende a 654,69 €.

1.5.- OTROS CONCEPTOS

En adición a las partidas presupuestarias anteriormente expuestas, en este apartado se incluyen los costes incurridos por otros conceptos del proyecto. Estos costes complementarios se agrupado, fundamentalmente, en papelería (impresión de documentos para su consulta) y material de oficina. Los costes vinculados a cada ítem pueden consultarse a través de la "Tabla 1.5".

Por otro lado, resulta preciso indicar que no se han presupuestado las licencias de acceso a bases de datos bibliográficas. En este caso, se han utilizado las fuentes de documentos suministradas de forma gratuita por la Universidad de Oviedo. Concretamente, se han empleado las siguientes herramientas *software* de gestión bibliográfica:

❖ *Repositorio Institucional de la Universidad de Oviedo (RUO).*

⁴⁶ En el caso del sistema de captura del movimiento *OptiTrack*, el parámetro de vida útil se ha establecido en el valor 10 [años]. Por tanto, empleando un razonamiento similar al expuesto para el *cobot UR3e*, la cantidad porcentual del concepto *OptiTrack* asciende a 4,17 [%]; tal y como puede visualizarse en la siguiente expresión:

$$CANTIDAD_{OptiTrack} [\%] = \frac{5 [meses]}{10 [años] \cdot 12 \left[\frac{meses}{año} \right]} \cdot 100 = 4,17 [\%]$$

- ❖ *Elsevier y Scopus.*
- ❖ *Researchgate.*
- ❖ *ScienceDirect.*

Tabla 1.5.- Costes derivados de la partida *Otros Conceptos*, los cuales no habían sido contemplados en secciones anteriores del actual presupuesto.

CONCEPTO	SUBTOTAL
Material de oficina	20 €
Papelería, impresión de documentos	25 €

Subtotal: **45 €**

Mediante el resultado anterior, puede observarse que el coste ligado a la partida *Otros Conceptos* toma el valor de 45 €.

1.6.- COSTE TOTAL DEL PRESUPUESTO DE EJECUCIÓN MATERIAL (*PEM*)

De acuerdo con el desglose de costes del proyecto mostrado durante las secciones anteriores, ya se está en disposición de calcular el Presupuesto de Ejecución Material. En la siguiente tabla se recoge un resumen sobre los costes derivados de cada partida presupuestaria:

Tabla 1.6.- Coste final ligado al *PEM* (Presupuesto de Ejecución Material), el cual se ha calculado mediante la adición de todas las partidas presupuestarias indicadas en secciones anteriores.

CONCEPTO	SUBTOTAL
COSTES DE MATERIAL <i>SOFTWARE</i>	602,25 €
COSTES DE EQUIPOS INFORMÁTICOS	108,29 €
COSTES DE MANO DE OBRA	3.000 €
COSTES DE LOS ELEMENTOS <i>HARDWARE</i> SELECCIONADOS	654,69 €
OTROS CONCEPTOS	45 €

Subtotal: 4.410,23 €

De acuerdo con la información recogida en este apartado, el *Presupuesto de Ejecución Material* consustancial al proyecto se ha estimado en 4.410,23 €.

2. *Gastos Generales y Beneficio*

Industrial

En base al *Presupuesto de Ejecución Material* detallado durante el apartado anterior, se han elaborado unos *Gastos Generales*⁴⁷; ineludiblemente presentes en cualquier proyecto de ingeniería. Asimismo, se ha definido un *Beneficio Industrial*⁴⁸ que permita garantizar, *a priori*, la rentabilidad económica del proyecto presupuestado.

Para el presente caso de estudio, los *Gastos Generales* han ascendido al 13,00 %; mientras que el *Beneficio Industrial* se ha establecido en el valor 9,00 %. Téngase en cuenta que ambos porcentajes se aplican al coste total bruto del proyecto, el cual ha sido estimado previamente mediante el *Presupuesto de Ejecución Material*.

Tabla 2.1.- Costes inherentes al *Beneficio Industrial* y a los *Gastos Generales* definidos en el presente *Trabajo Fin de Máster*.

CONCEPTO	SUBTOTAL
GASTOS GENERALES (13,00 % del coste total bruto - <i>PEM</i> : 4.410,23 €)	573,33 €
BENEFICIO INDUSTRIAL (9,00 % del coste total bruto - <i>PEM</i> : 4.410,23 €)	396,92 €

Subtotal: **970,25 €**

Los *Gastos Generales* y el *Beneficio Industrial* representan 970,25 €.

⁴⁷ Los *Gastos Generales* se han incluido en el presupuesto de este *Trabajo Fin de Máster* con la finalidad de considerar los supuestos enumerados a continuación:

- Gastos derivados del consumo de energía eléctrica.
- Costes vinculados a los trámites administrativos del proyecto.
- Gastos económicos originados repentinamente durante el desarrollo del *TFM*.
- Otros costes generados habitualmente en el entorno laboral de ingeniería.

⁴⁸ El *Beneficio Industrial* es un concepto aplicado en materia de viabilidad económica del proyecto. De este modo, dicho beneficio se define como un porcentaje sobre el *Presupuesto de Ejecución Material (PEM)*; permitiendo incorporar una ventaja económica en la realización del trabajo presupuestado.

3. Coste total del proyecto (Presupuesto de Ejecución por Contrata – *PEC*)

En este capítulo final del presupuesto maquetado, se indica el coste global de todo el proyecto. Para ello, se hará uso del Presupuesto de Ejecución Material previamente calculado; así como del Beneficio Industrial y los Gastos Generales considerados.

Tabla 3.1.- Coste total del proyecto – *Presupuesto de Ejecución por Contrata (PEC)*.

CONCEPTO	SUBTOTAL
COSTE TOTAL DEL PRESUPUESTO DE EJECUCIÓN MATERIAL	4.410,23 €
GASTOS GENERALES Y BENEFICIO INDUSTRIAL	970,25 €

COSTE TOTAL SIN IMPUESTOS: 5.380,48 €
IVA 21 %: 1.129,90 €
COSTE TOTAL, IVA INCLUIDO 6.510,38 €

El coste total del proyecto asciende a la expresada cantidad de **SEIS MIL QUINIENTOS DIEZ EUROS CON TREINTA Y OCHO CÉNTIMOS.**

Gijón, a 10/07/2023



Fdo. Alejandro Castellanos Alonso

CAPÍTULO 3.

MATERIALES ADJUNTADOS

1. *Software*

En el interior del directorio *Material*, se ha incluido la subcarpeta *Software*; a través de la cual se anexan las últimas versiones asociadas a los ficheros de código fuente desarrollados durante este *Trabajo Fin de Máster*.

2. Hojas de características

Adicionalmente, se han adjuntado en el directorio del proyecto las *datasheets* vinculadas a los dispositivos *hardware* que han sido considerados durante el transcurso del *Trabajo Fin de Máster*.

3. Fotos realizadas durante las experiencias de laboratorio

Finalmente, se han incorporado dentro de la carpeta "Fotos_Laboratorio" varias imágenes capturadas durante los experimentos realizados en el *RoboLab*; inherentes a la calibración y evaluación inerciales de los diversos *IMUs*.