



Caracterización del arreglo de sensores MARG, filtrado y orientación en tiempo real

Trabajo Final

Fundamentos de Procesamiento de Señales

Esp. Ing Nelson Ariel Fortunatti

Descripción general

Introducción

Dado el creciente auge durante los últimos años en temas relacionados con robótica e inteligencia artificial (IA), para el proyecto final de la MSE [\[planificación\]](#) se pretende realizar un sistema robótico con fines educativos para tratar temas relacionados con visión por computadora como el procesamiento de imágenes para detección de objetos, control de estabilidad del robot y manipulación de un brazo robótico. El proyecto consiste en un sistema complejo que puede analizar y desplazarse por una determinada área con la capacidad de configurarse en base a diversas estrategias.

Este proyecto introduce un robot móvil de dos ruedas dotado de un sistema de auto-balanceo que se basa en la utilización de un giróscopo para determinar el grado de inclinación actual del mismo, utilizando una estrategia de control de terminada se accionan ambas ruedas para corregir la inclinación y mantener el robot en posición vertical. La Figura 1 muestra un diagrama simplificado del comportamiento de este tipo de robots donde el sistema se modela en base al control de un péndulo invertido [1].

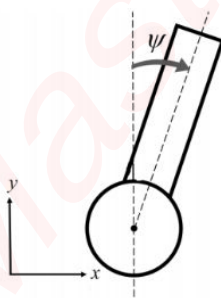


Figura 1: Diagrama simplificado de un robot con dos ruedas. X e Y determinan el eje de coordenadas, ψ determina el ángulo de inclinación.

Cuando surge una perturbación inesperada con respecto al ángulo de inclinación del robot, el robot mantiene su estabilidad en mayor o menor medida dependiendo del tiempo de procesamiento de la señal, tiempo de procesamiento del algoritmo de control y capacidad accionamiento de los motores de manera tal que su respuesta sea rápida y precisa. Ésto último depende principalmente del tipo de motor utilizado, cuyas características recomendables son obtener un torque elevado, momento de inercia reducido y velocidad angular elevada. Si la perturbación recibida excede la capacidad de respuesta del robot, el sistema se torna inestable provocando la caída del robot con una consecuente rotura del mismo.

Los robots móviles de dos ruedas tienen la capacidad de obtener mejor movilidad, rotación en espacios reducidos e incluso se desplazan con mayor velocidad que los robots que

disponen de piernas para su desplazamiento. Es por ello que se utilizan en gran variedad de aplicaciones como son los Segway[\[link\]](#).

Como trabajo final de la materia Fundamentos de Procesamiento de señales se pretende caracterizar el comportamiento de los arreglos de sensores *MARG* (Magnetic, Angular Rate, and Gravity) que consiste en un dispositivo híbrido que posee un sensor *IMU* (Inertial Measurement Unit) e incorpora un magnetómetro de tres ejes. Un *IMU* solo puede medir la posición relativa con respecto a la dirección de la gravedad. En cambio, los sistemas *MARG* también conocidos como *AHRS* (Attitude and Heading Reference Systems) son capaces de proveer una orientación completa ya que se puede determinar la orientación con respecto al centro del planeta tierra y el polo norte magnético.

Un sensor *MARG* debe ser utilizado en el robot móvil a desarrollar ya que se pretende estimar el grado de inclinación del robot con una buena precisión y un tiempo de respuesta acotado para poder ajustar la inclinación del robot rápidamente haciendo uso de un dado algoritmo de control. Por otro lado es importante determinar la orientación del robot y poder en un futuro determinar la posición del mismo a medida que se desplace.

Un giroscopio que mide la velocidad angular, si se conoce la posición inicial, puede ser utilizado para obtener el cambio de la orientación del sensor mediante integración. Al integrar las mediciones del giroscopio se producen errores que se van acumulando a medida que se realiza una nueva integración. Estos sensores no pueden proveer una medición absoluta de la orientación por sí mismos.

Es aquí donde empieza a tener relevancia la incorporación de un acelerómetro y un magnetómetro para poder determinar la dirección de la gravedad y la dirección del campo magnético terrestre respectivamente. Estos sensores son propensos a altos niveles de ruido como por ejemplo aceleraciones dadas por el movimiento o vibraciones del objeto donde se encuentran montados los sensores pueden deteriorar la estimación de la dirección de la gravedad.

La tarea de un filtro de orientación utilizar las cualidades de cada uno de los sensores nombrados y fusionarlas para poder determinar la orientación del dispositivo con el menor error posible. El filtro Kalman es uno de los más utilizados para la fusión de los sensores ya que se puede obtener precisión y efectividad en las mediciones. Sin embargo, son complicados de implementar, requieren una frecuencia de muestreo mínima que en algunos casos puede llegar a los 30 KHz, gran disponibilidad de memoria y poder de procesamiento.

La utilización de un *Sensor Fusion* (algoritmo basado en la implementación de un filtro de orientación) debe ser en algunos casos capaz de ser utilizado en sistemas embebidos donde la capacidad de procesamiento y memoria disponible es acotada. Para estos casos se conocen dos algoritmos *Mahony*[\[link\]](#) y *Madgwick*[\[link\]](#). *Mahony* pretende ser una solución eficiente y efectiva para ser utilizado en microcontroladores de baja performance pero solo es útil para la utilización en un *IMU*. Sebastian *Madgwick* basado en todas las

Maestría en Sistemas Embebidos - UBA

Trabajo Final de la materia Fundamentos de Procesamiento de Señales
Caracterización del arreglo de sensores MARG, filtrado y orientación en tiempo real

características detalladas anteriormente propone un nuevo algoritmo que puede ser utilizado tanto en un *IMU* como en un *MARG*, el costo computacional es mayor al utilizado en el algoritmo de Mahony pero microcontroladores de la familia ARM Cortex M3 en adelante puede implementarlo sin mayores dificultades.

Dadas las características que posee el algoritmo de Madgwick es el elegido para el desarrollo del presente trabajo.

Algoritmo Sensor Fusion Madgwick

Esta sección detalla las características más sobresalientes, la performance obtenida del algoritmo de *Madgwick*. Todos los gráficos presentados son de autoría de *Sebastián Madgwick* y el documento original puede encontrarse en el siguiente [link](#).

Madgwick utiliza la representación de la orientación mediante un quaternion. En su paper presenta el desarrollo matemático del filtro y muestra la utilización de un algoritmo optimizado de descenso de gradiente que permite un rendimiento a bajas frecuencias de muestreo, un algoritmo de compensación de distorsión magnética en tiempo real y compensación de deriva por efecto de integraciones sucesivas del giroscopio y de la temperatura. El diagrama en bloques del mismo puede verse en la Figura 2 el algoritmo que es utilizado para fusionar los sensores y poder compensar los efectos de distorsión magnética y deriva del giroscopio.

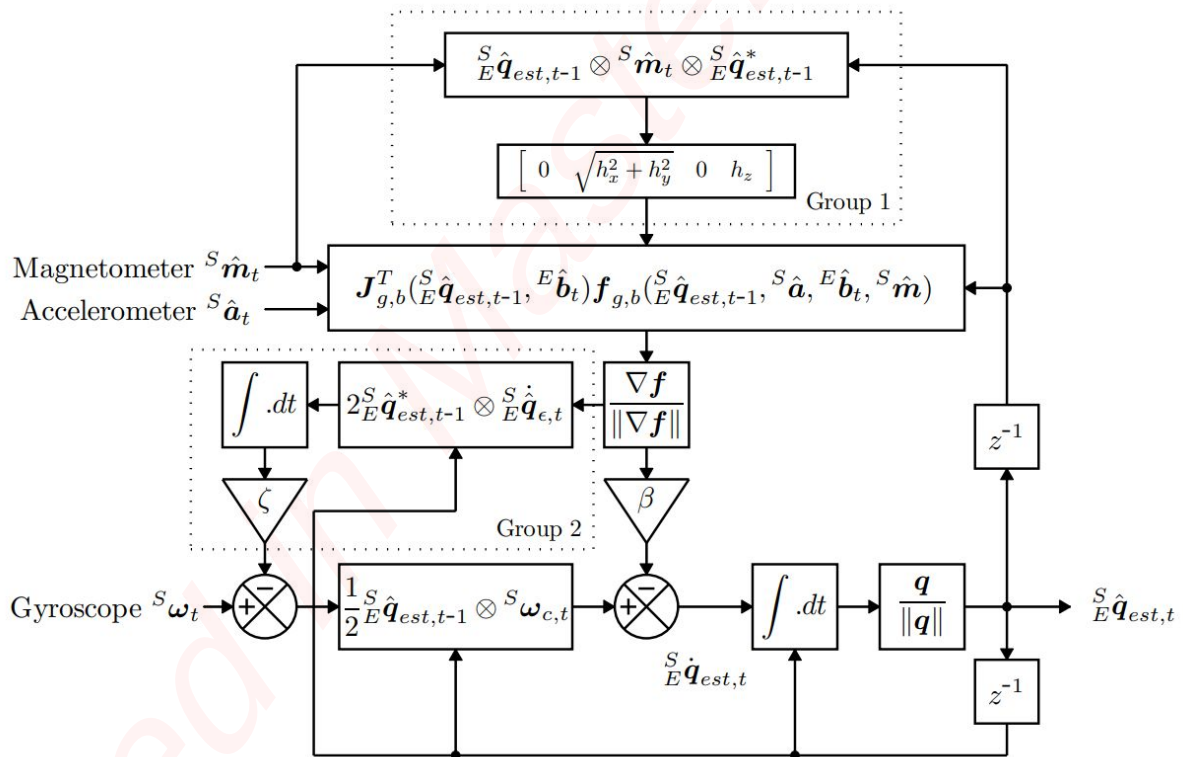


Figura 2: Diagrama en bloques del filtro *Madgwick* para un sensor MARG donde se encuentra implementada la compensación por distorsión magnética (Grupo 1) y la deriva del giroscopio (Grupo 2).

Para evaluar la performance del algoritmo, Madgwick realizó ensayos donde compara los resultados obtenidos con respecto a una medición de alta precisión y a su vez lo compara con una estimación de orientación basada en un filtro Kalman. En este caso se basa en los ángulos de Euler que representan los ángulos de rotación con respecto a los ejes de coordenadas x, y y z, siendo los ángulos de rotación ϕ , θ y ψ respectivamente. Para obtener estos ángulos del quaternion resultante del algoritmo se utilizan las siguientes ecuaciones.

$$\psi = \text{Atan2} (2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1)$$

$$\theta = -\sin^{-1} (2q_2q_4 + 2q_1q_3)$$

$$\phi = \text{Atan2} (2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1)$$

A continuación en la Figura 3, 4 y 5 se muestran los resultados obtenidos de los ángulos de rotación determinados por las ecuaciones anteriores. Puede notarse que en todos los casos que el error obtenido por el algoritmo de *Madgwick* es del mismo orden o, en algunos casos, incluso menor que la solución realizada con un filtro *Kalman*.

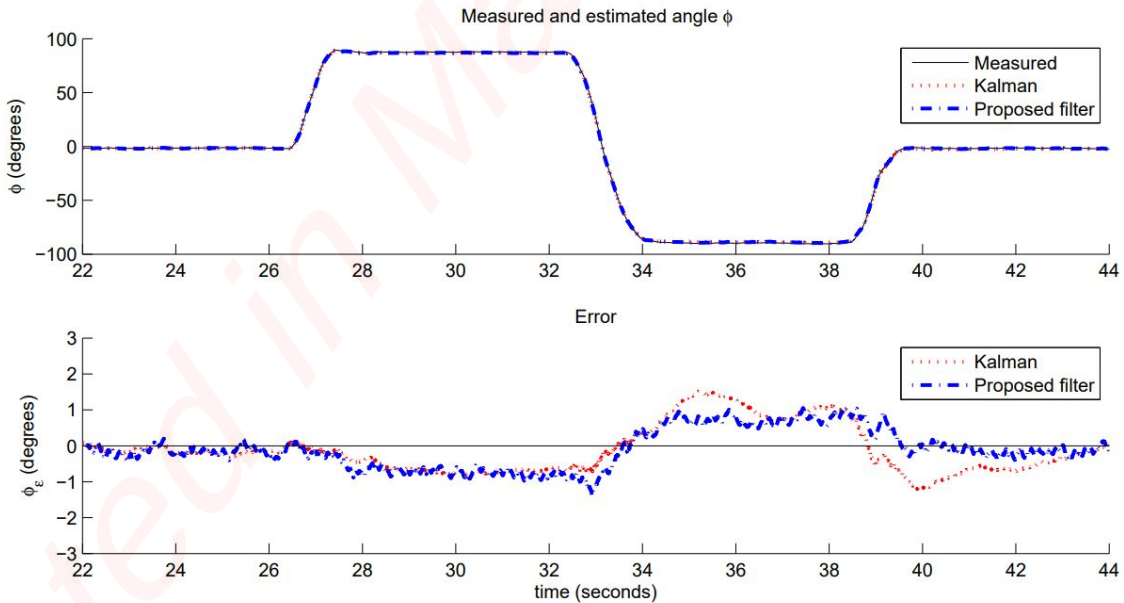


Figura 3: Análisis de los resultados obtenidos del algoritmo de *Madgwick* con respecto a la medición de precisión y a la solución con el filtro *Kalman*. Se estima el ángulo ϕ que representa la rotación con respecto al eje x.

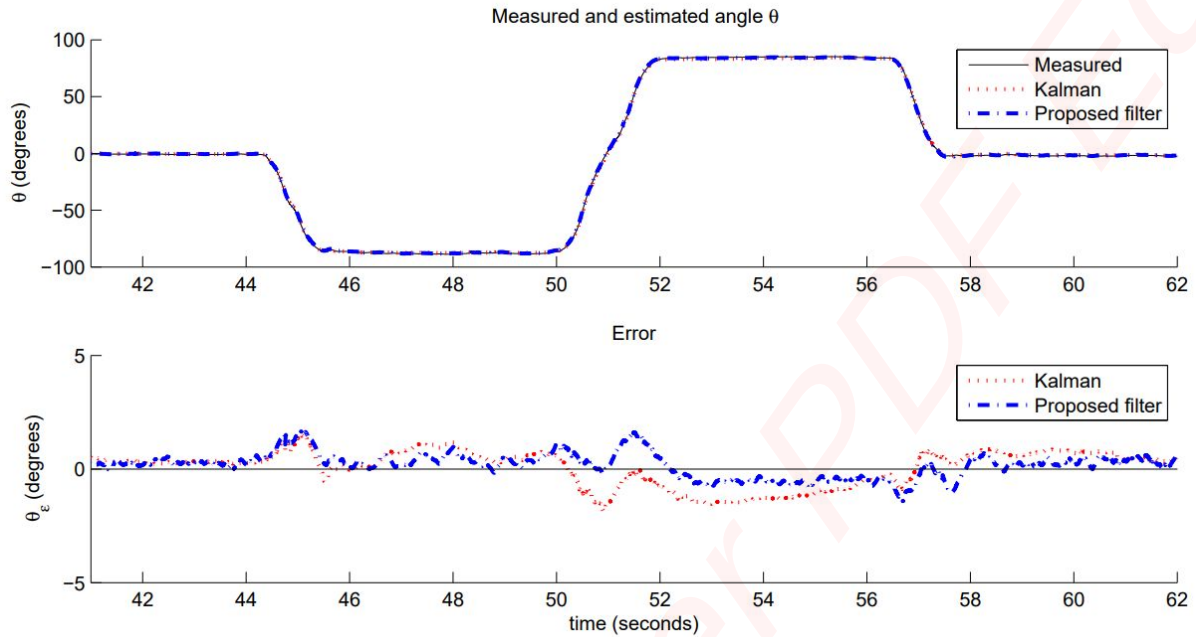


Figura 4: Análisis de los resultados obtenidos del algoritmo de *Madgwick* con respecto a la medición de precisión y a la solución con el filtro *Kalman*. Se estima el ángulo θ que representa la rotación con respecto al eje y.

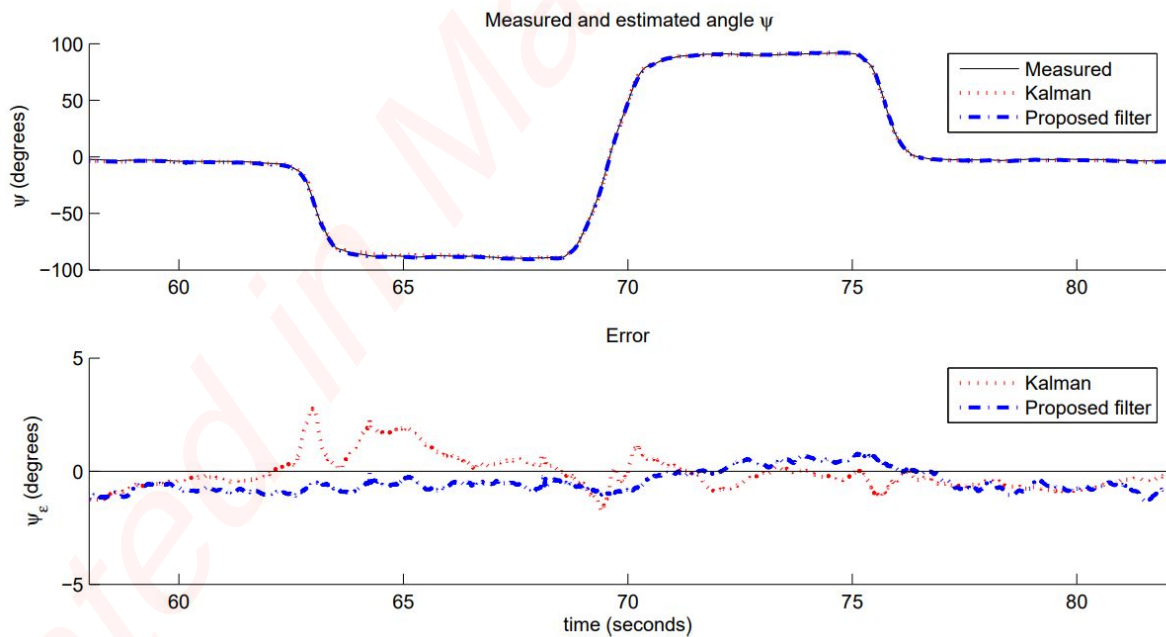


Figura 5: Análisis de los resultados obtenidos del algoritmo de *Madgwick* con respecto a la medición de precisión y a la solución con el filtro *Kalman*. Se estima el ángulo ψ que representa la rotación con respecto al eje z.

Para determinar el error estático y dinámico de la utilización del algoritmo de *Madgwick* y comparar los resultados con el filtro del *Kalman* se asume que todo error estático corresponde a un error menor a $5^\circ/s$ y el error dinámico es todo error mayor a $5^\circ/s$. La Tabla 1 muestra que los resultados del algoritmo de *Madgwick* son mejores en todos los casos ensayados teniendo en cuenta que para el algoritmo que no utiliza magnetómetro no se estima ψ ya que este para que el error sea reducido debe ser corregido haciendo uso del magnetómetro.

Euler parameter	Kalman-based algorithm	Proposed filter (MARG)	Proposed filter (IMU)
RMS $[\phi_e]$ static	0.789°	0.581°	0.594°
RMS $[\phi_e]$ dynamic	0.769°	0.625°	0.623°
RMS $[\theta_e]$ static	0.819°	0.502°	0.497°
RMS $[\theta_e]$ dynamic	0.847°	0.668°	0.668°
RMS $[\psi_e]$ static	1.150°	1.073°	N/A
RMS $[\psi_e]$ dynamic	1.344°	1.110°	N/A

Tabla 1: Análisis de los resultados obtenidos del algoritmo de *Madgwick* tanto para sensores *MARG* como *IMU* con respecto a la solución con el filtro *Kalman*.

El filtro de *Madgwick* tiene dos parámetros que pueden ser configurables, por un lado se tiene la ganancia del filtro β y por otro la frecuencia de muestreo de los datos del sensor. El desempeño del filtro con respecto a la ganancia se puede ver en la Figura 6 donde se puede ver que existe un punto óptimo para cada algoritmo propuesto. El valor de β debe ser lo suficientemente grande como para reducir los errores ocasionados por la integración del giroscopio y por otro lado debe ser lo suficientemente pequeño para no interferir en gran medida introduciendo grandes pasos en la iteración del gradiente descendiente.

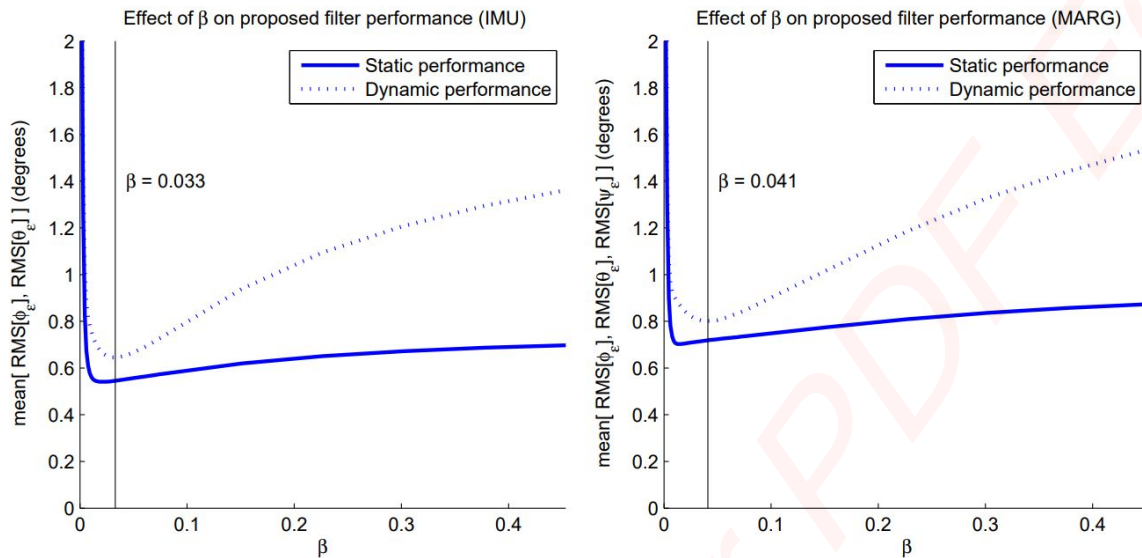


Figura 6: Efecto de β en el error obtenido en el algoritmo de Madgwick para el uso en *IMU* y *MARG*.

Por otro lado tenemos la frecuencia de muestreo que se mencionó anteriormente que para el caso de un filtro Kalman varía entre 512 Hz y 30 KHz. Para el caso del algoritmo de *Madgwick* se pueden utilizar frecuencias de muestreo notablemente menores como pueden verse en la Figura 7. Se puede notar que para este caso no existe una situación de compromiso como sucedió con respecto a la ganancia, en este caso, cuanto mayor es la frecuencia de muestreo menor es el error obtenido, aunque obviamente el costo computacional es mayor y tanto el sensor, como el procesador donde se realice el algoritmo de *Madgwick*, poseen una frecuencia máxima que podrán satisfacer.

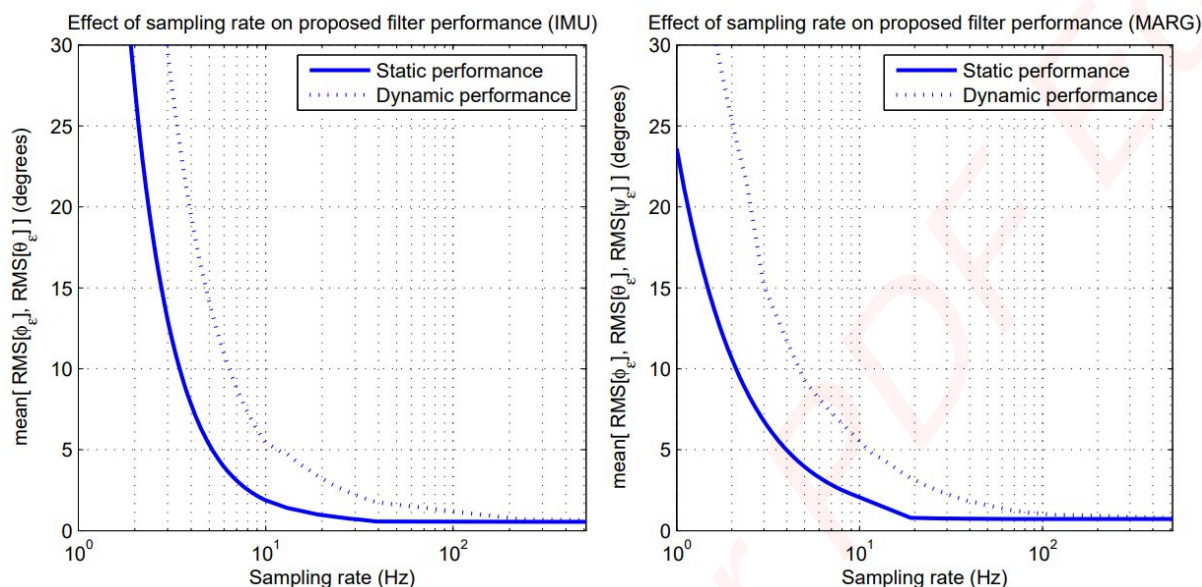


Figura 7: Efecto de la frecuencia de muestreo en el error obtenido en el algoritmo de Madgwick para el uso en *IMU* y *MARG*.

Por último *Madgwick* analiza el efecto de la deriva térmica y por integración del giroscopio utilizando su algoritmo. Analiza si su algoritmo puede seguir los efectos mencionados para poder determinar si los errores producidos por este efecto son considerables o no. Como se puede notar en la Figura 8 el algoritmo sigue apropiadamente a los efectos producidos por la deriva para poder compensarlos de forma acorde.

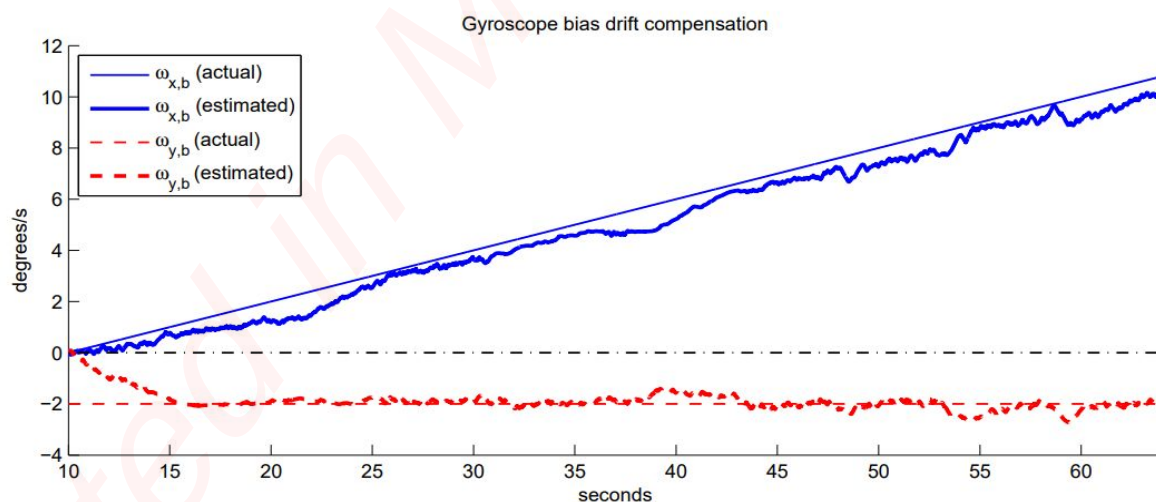


Figura 8: Algoritmo de *Madgwick* detectando los efectos de deriva térmica y por integración del giroscopio.

Por los resultados mencionados en esta sección donde puede verse que el algoritmo es capaz de estimar la orientación con respecto a cada ángulo de Euler con un error aproximadamente de 1° y a su vez tiene la capacidad de ser utilizado en un

microcontrolador sin requerir un costo computacional excesivo, este algoritmo es el utilizado para implementar la fusión de sensores del trabajo.

Representación en tiempo real del estado del MARG - Análisis temporal y en frecuencia

A continuación se detalla la estructura de programas asociados a verificar el comportamiento del *MARG* en tiempo real tanto para las señales provenientes del giroscopio, acelerómetro y magnetómetro para cada uno de los 3 ejes involucrados y también para analizar el comportamiento del algoritmo propuesto por *Madgwick*.

La Figura 9 muestra en un diagrama en bloques como se obtienen los datos provenientes del sensor, estos datos son procesados con el algoritmo de Sensor Fusión de *Madgwick* y son enviados a través de la UART para poder ser analizados en la PC. Dentro de la PC se realizan la transformada de Fourier a cada uno de los datos de entrada excepto para los 4 valores correspondiente a quaternion entregado por *Madgwick*, en este caso, se realiza la transformación de quaternion a ángulos de Euler como se detalló anteriormente. Luego a éstos ángulos también se le aplica la transformada rápida de Fourier para poder graficar en pantalla a medida que llega un nuevo dato la transformada de los últimos N valores recibidos. Siendo N un parámetro configurable dentro del script de Python.

Mediante una serie de botones se puede seleccionar qué datos se pretenden ver. Esto da la posibilidad de ver los valores de los ejes X, Y y Z tanto para el acelerómetro, giroscopio y magnetómetro. Y a su vez también se pueden ver los ángulos de Euler. Como al seleccionar un botón solo se muestra los gráficos correspondiente a alguna de estas 4 opciones. El programa solo calcula la transformada de Fourier para los datos presentados. Esto ahorra tiempo de procesamiento y mejora el ancho de banda para la lectura, procesamiento, transmisión, cálculo de transformadas y presentación de los datos.

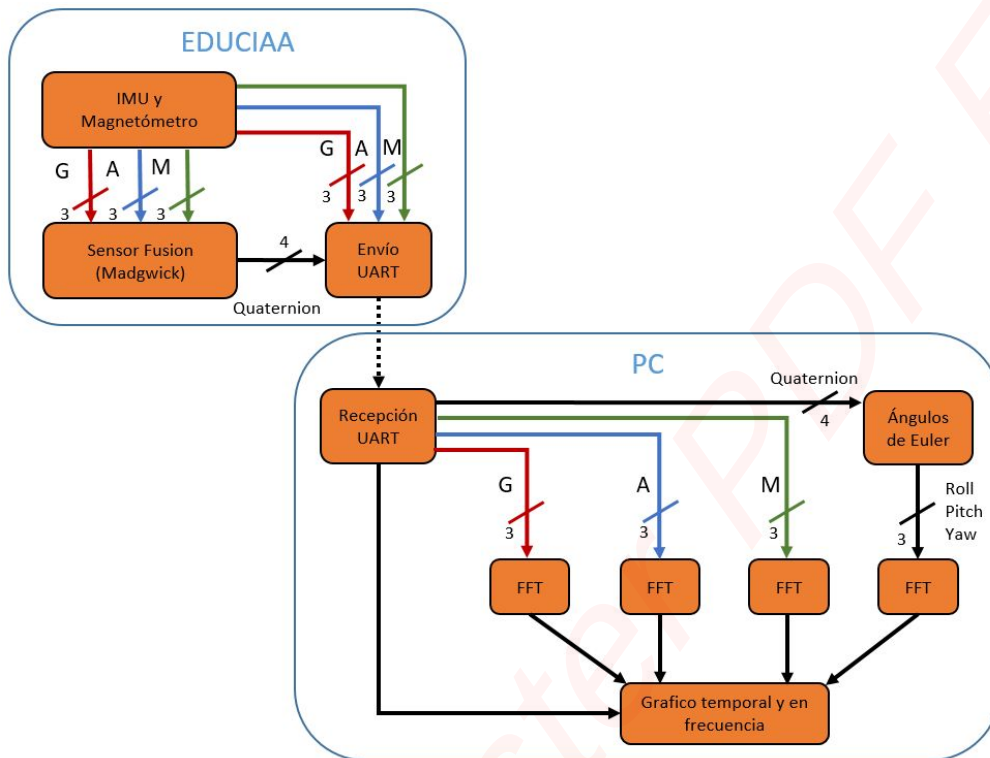


Figura 9: Diagrama en bloques de los programas utilizados para graficar en tiempo real los valores obtenidos por el MARG y el comportamiento del filtro *Madgwick*

El acelerómetro si bien se muestra correctamente en la Figura 10, presenta una saturación en el valor de 5 m/s^2 si bien se encuentra configurado el MARG a 16G se debe verificar este comportamiento con mayor detalle.

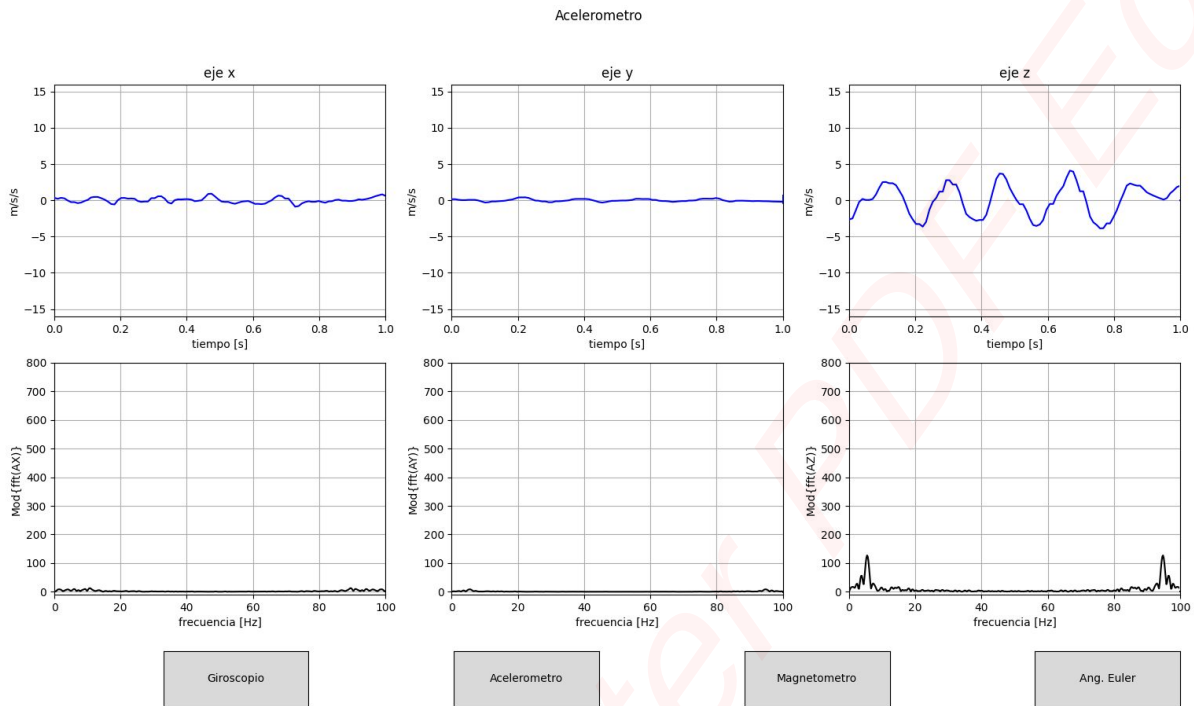


Figura 10: Comportamiento del acelerómetro frente a pequeñas variaciones en el sensor.

El giroscopio como se muestra en la Figura 11 presenta errores fijos que no pudieron ser mitigados con la calibración del MPU 9250, si bien se puede ver un gran error en la medición esto no se ve reflejado al utilizar el filtro de madgwick como se puede ver más adelante.

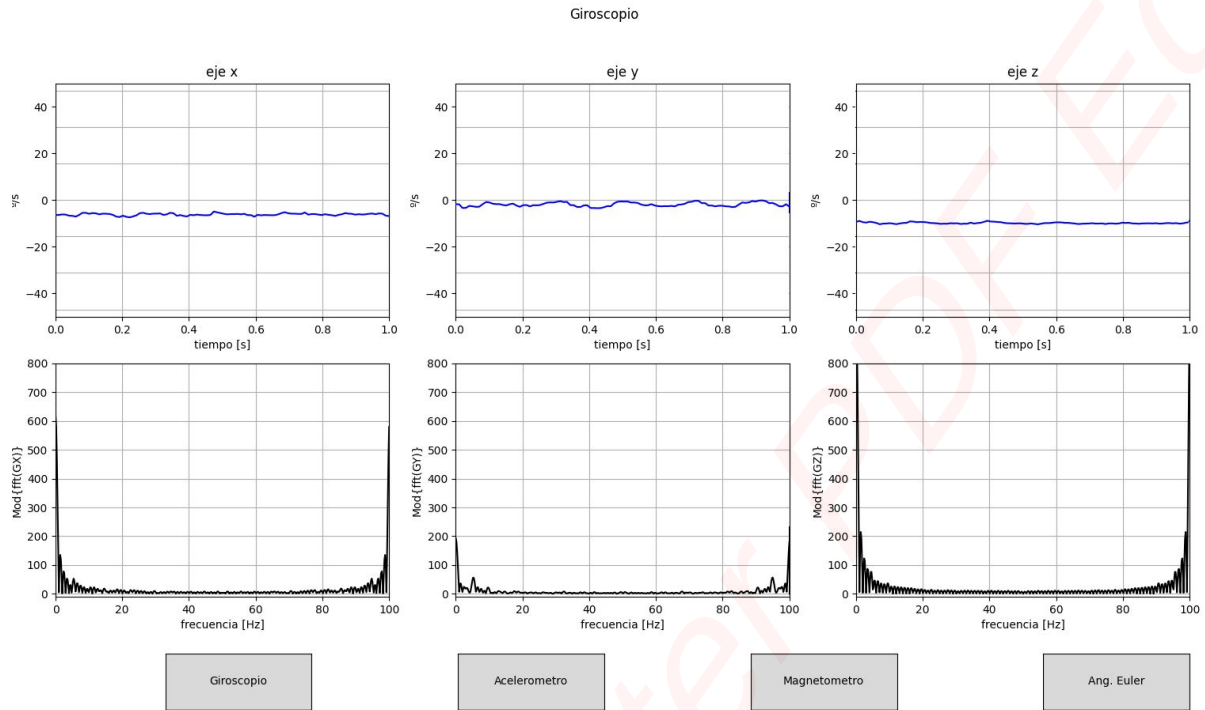


Figura 11: Comportamiento del giroscopio frente a pequeñas variaciones en el sensor.

La Figura 12 muestra los valores de entrada del magnetómetro, en este caso no se pudo determinar ningún comportamiento fuera de lo normal.

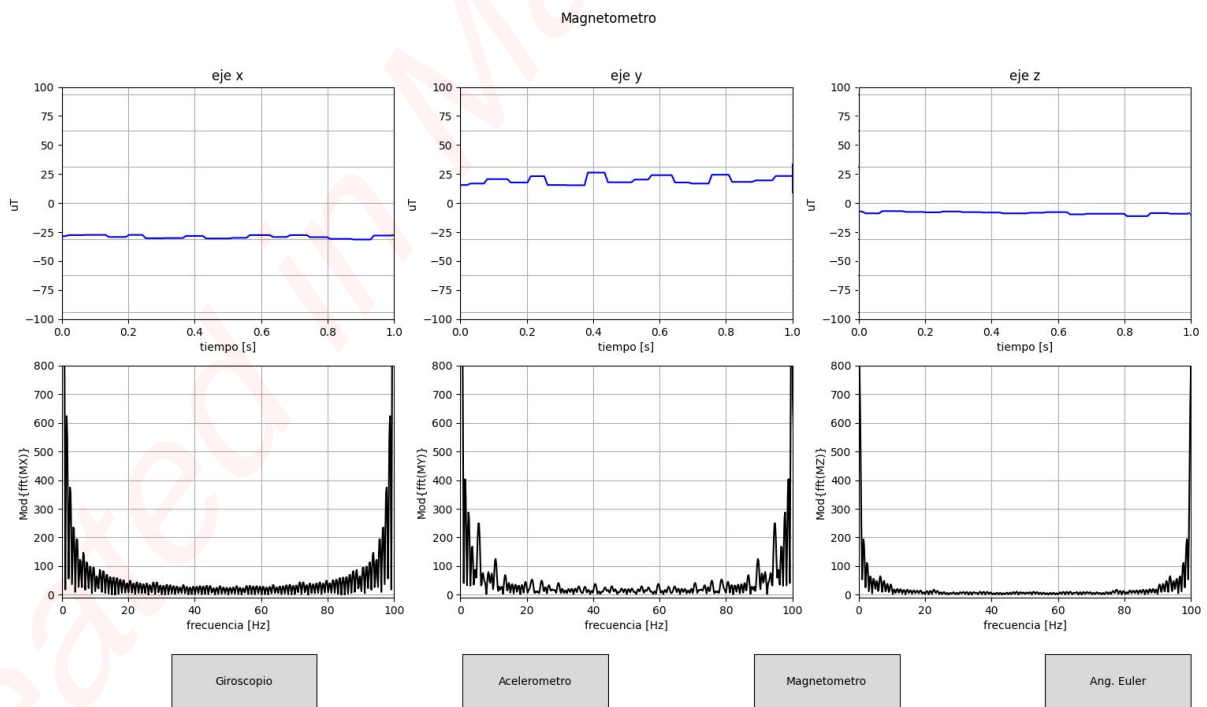


Figura 12: Comportamiento del magnetómetro frente a pequeñas variaciones en el sensor.

Los ángulos de Euler se mantienen constante a lo largo del tiempo y no son condicionados por el error percibido en el giroscopio. Esto se debe a que el algoritmo de *Madgwick* genera buenos resultados. La Figura 13 muestra este comportamiento.

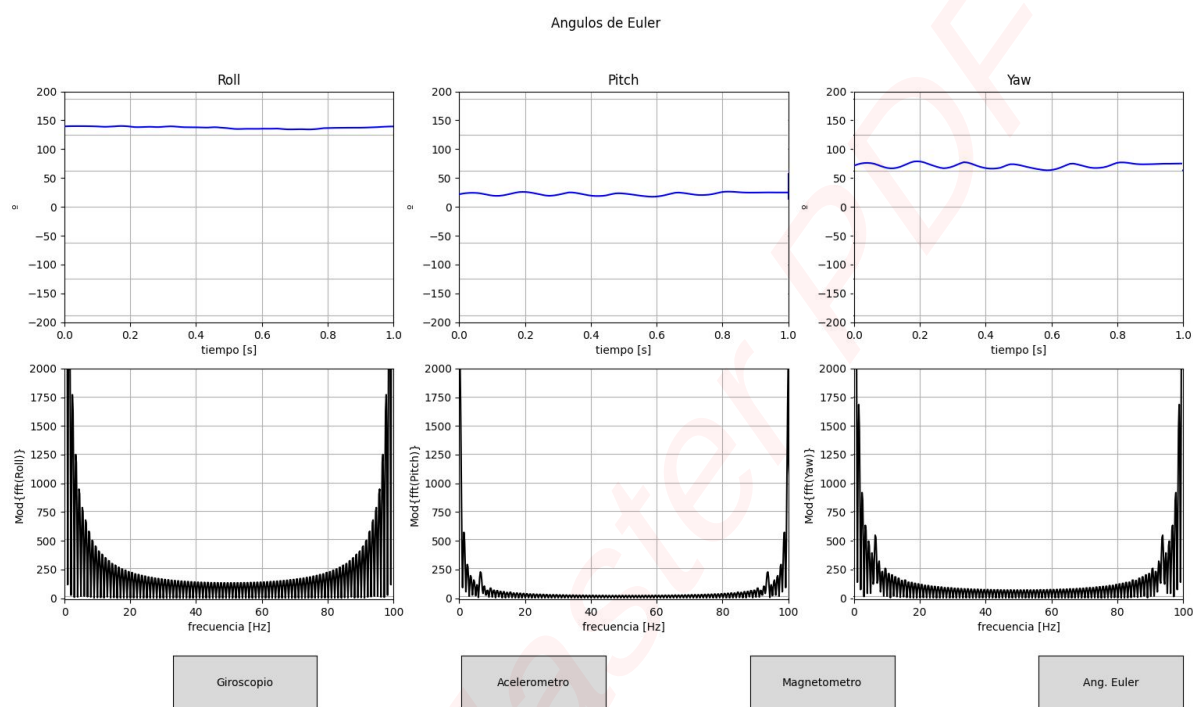


Figura 13: Ángulos de Euler frente a pequeñas variaciones en el sensor.

El detalle y comportamiento de los siguiente programas puede verse en los siguientes links:

- [Explicación de programas.](#)
- [Demostración.](#)

Filtro propuesto para ser utilizado dentro del microcontrolador

En base al comportamiento en frecuencia obtenido del programa anterior se pudo detectar que los datos obtenidos no son limitados en frecuencia. Por otro lado no se puede limitar el ancho de banda de la señal proveniente del Magnetómetro. Para asegurarnos que el ancho de banda de transmisión de los datos se ajusta a la frecuencia que se procesan los datos y se transmiten, es deseable diseñar un filtro para limitar en banda el comportamiento del MPU 9250.

Dadas estas características se pretende crear un programa para analizar el comportamiento de un filtro que pueda ser utilizado para todas las variables involucradas (3-Giroscopio, 3-Magnetómetro y 3-Acelerómetro).

Por otro lado, el diseño del filtro no debe retrasar en demasía el procesamiento de los datos dentro del microcontrolador utilizado y no debe utilizar grandes cantidades de memoria, ya que tanto la performance como la memoria son recursos acotados en este tipo de dispositivos.

Existen dos tipos de filtros a utilizar *FIR* (Finite Impulse Response) y *IIR* (Infinite Impulse Response). Para el caso de *FIR* se puede obtener un filtro de mejor calidad obteniendo una fase lineal que provoca que la variable siga siendo controlable, aunque requiere un gran número de elementos a evaluar. Con respecto al *IIR*, la fase obtenida no es lineal por lo cual no es posible el control de la variable aunque requiere significativamente menos coeficientes a evaluar.

Dadas las características mencionadas anteriormente se opta por utilizar un filtro IIR Chebyshev del tipo II que posee una respuesta monótonamente decreciente en la banda de paso para asegurarnos en no interferir en los valores obtenidos por parte del sensor. Las características de la base serán evaluadas luego de diseñar completamente el sistema y se determinará si la respuesta es aceptable o debe cambiarse a un diseño con filtro FIR.

La figura 14 muestra el procedimiento del diseño del filtro y cómo se evalúa el comportamiento del mismo utilizando un script de Python.

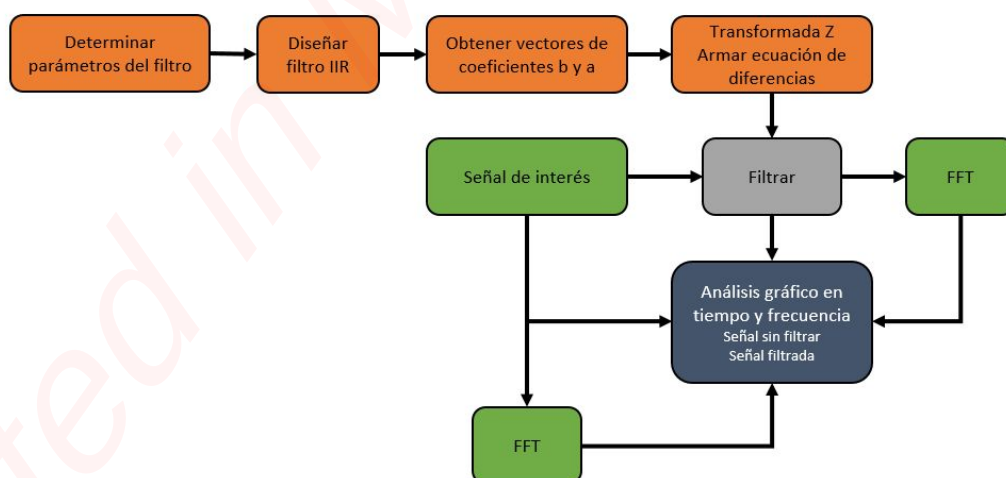


Figura 14: Diagrama en bloques para el desarrollo del filtro Chebyshev II utilizando transformada z y ecuaciones de diferencias.

Haciendo uso de la herramienta Pyfda se diseña el filtro para la frecuencia de corte 80 Hz, del tipo Chebyshev II, con una atenuación en la banda de rechazo de 120 dB, como puede verse en las Figuras 15 y 16 el módulo y fase respectivamente.

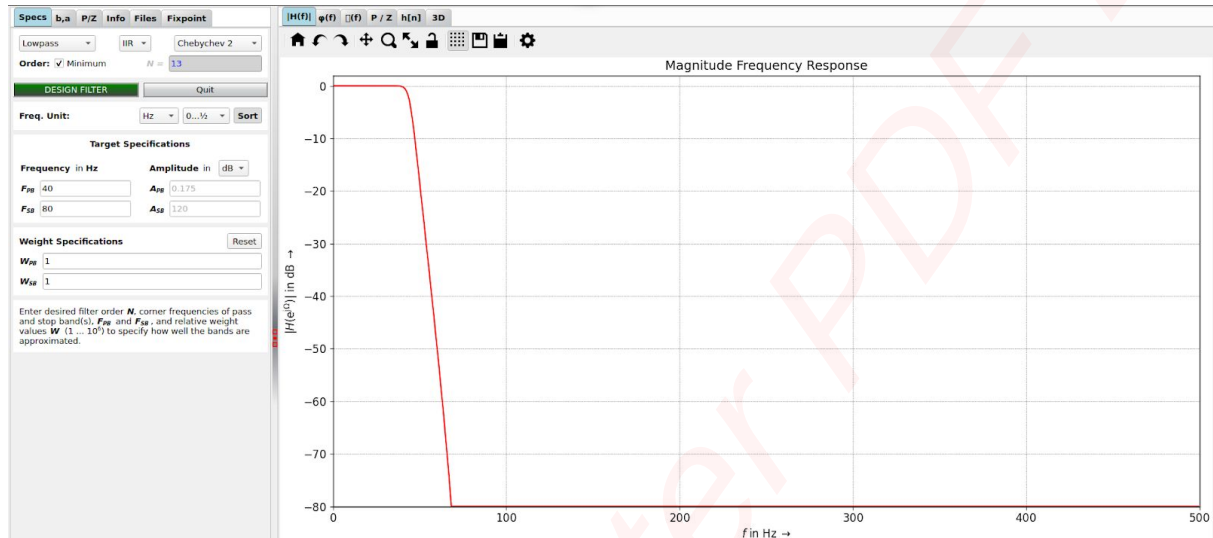


Figura 15: Respuesta en frecuencia del módulo del filtro diseñado haciendo uso de Pyfda.

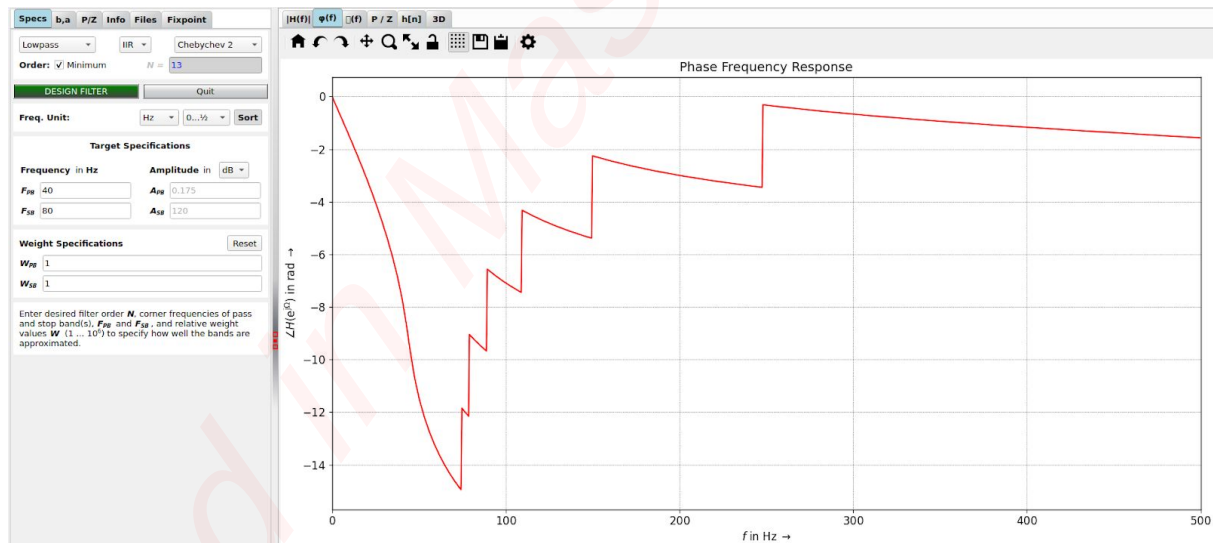


Figura 16: Respuesta en frecuencia de la fase del filtro diseñado haciendo uso de Pyfda.

Luego de determinar el filtro a utilizar, se obtienen de Pyfda los coeficientes b y a de la función de transferencia del filtro diseñado dados por la siguiente ecuación [\[link1\]](#):

$$H(z) = B(z)/A(z)$$

Maestría en Sistemas Embebidos - UBA

Trabajo Final de la materia Fundamentos de Procesamiento de Señales Caracterización del arreglo de sensores MARG, filtrado y orientación en tiempo real

$$H(z) = (b(0) + b(1) * z^{-1} + \dots + b(n) * z^{-n}) / (a(0) + a(1) * z^{-1} + \dots + a(n) * z^{-n})$$

Siendo siempre $a(0) = 1$.

Suponiendo “x” como los valores de entrada obtenidos del sensor MPU 9250 e “y” los valores de salida filtrados. La ecuación de diferencias queda de la siguiente forma [\[link2\]](#):

$$Y[n] = (b(0)x[n] + b(1) * x[n-1] + \dots) - (a(1) * Y[n-1] + a(2) * Y[n-2] + \dots)$$

Para nuestro caso el filtro diseñado posee 13 coeficientes de b y de a, por lo cual debemos almacenar los 12 valores previos de x e y para poder multiplicarlos con los coeficientes y así obtener nuestro nuevo valor de salida Y[n]. Podemos ver que un filtro IIR se basa en un algoritmo recursivo ya que necesita resultados obtenidos de los cálculos previos.

A continuación, en la Figura 17 se muestran los resultados obtenidos para una señal de entrada con diferentes tonos en frecuencia donde puede notarse que el filtro correctamente para la frecuencia de corte. Se puede notar que el filtrado introduce un corrimiento en fase de la señal filtrada con respecto al tono real graficado. También debe notarse que en algunos momentos se produce un corrimiento de fase de la señal de salida debido a las características del filtro IIR.

La señal de entrada a evaluar está dada por la siguiente ecuación:

$$x_data = \sin(2\pi \cdot 10 \cdot t_data) + \sin(2\pi \cdot 150 \cdot t_data) + \sin(2\pi \cdot 200 \cdot t_data) + \sin(2\pi \cdot 400 \cdot t_data)$$

Filtrado IIR Chebyshev II utilizando transformada Z

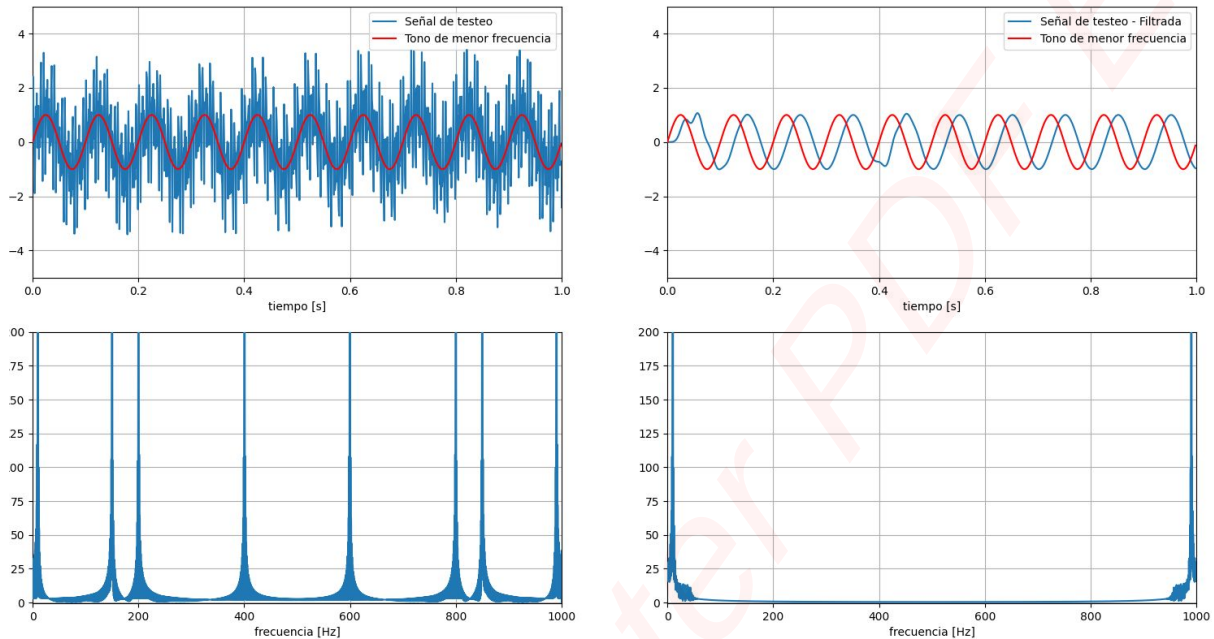


Figura 17: Gráfico de la señal original en tiempo y frecuencia (izquierda) y señal filtrada en tiempo y frecuencia (derecha).

En base al filtro diseñado como trabajo futuro se pretende verificar el comportamiento del filtro para las variables obtenidas del sensor, en primera instancia filtrando las señales obtenidas desde el microcontrolador como se muestra en la Figura 18. Luego de verificar que el desempeño del filtro es correcto el mismo va a ser introducido dentro del microcontrolador para ecualizar las señales obtenidas del sensor MPU 9250 y enviárselas al filtro de *Madgwick* para asegurarnos que las señales estén limitadas en frecuencia y no se produzcan interferencias por solapamientos, esto se puede ver en la Figura 19.

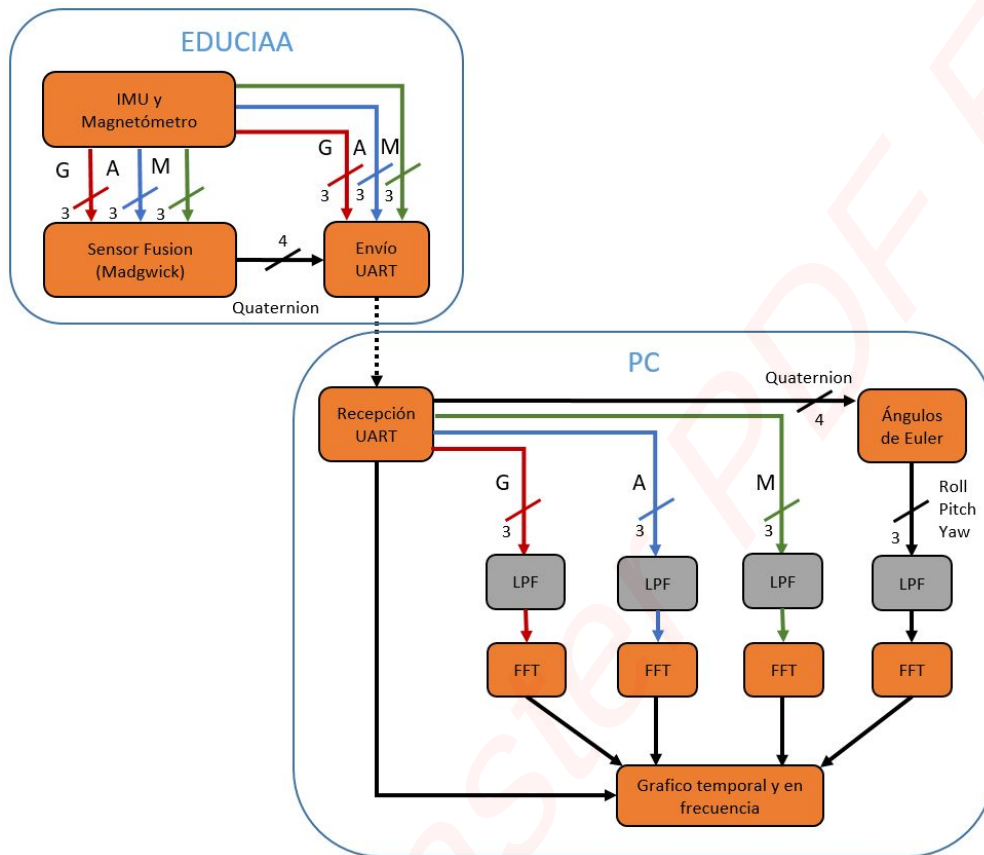


Figura 18: Diagrama en bloques de los programas utilizados para graficar en tiempo real los valores obtenidos por el MARG y el comportamiento del filtro Madgwick junto con etapas de filtrado denominadas LPF.

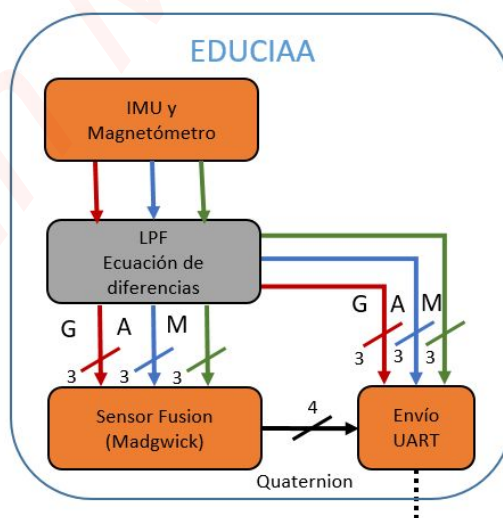


Figura 19: Diagrama en bloques que muestra la implementación dentro del microcontrolador del filtro pasa bajos implementado con ecuaciones de diferencias.

Utilización de OpenGL para mover un objeto en tiempo real en base a movimientos del MARG

En última instancia del trabajo se creó un programa para mover un objeto en tres dimensiones en base a la orientación leída desde el sensor MPU 9250. Para que el comportamiento sea en tiempo real fue necesario limitar la tasa de transferencia de datos a 50Hz ya que el objeto en pantalla solo puede actualizarse a razón de 60 cuadros por segundo. Esto produjo que en primera medida el comportamiento del programa no fuera el óptimo por cuestiones de no estar limitadas en frecuencia las señales transmitidas a 50Hz. Por otro lado fue necesario reducir en 10 Hz la transferencia de datos con respecto a la tasa de refresco en pantalla ya que en algunos momentos no era posible procesar los datos a tiempo dentro de la PC lo que producía que los datos enviados desde el microcontrolador queden alojados en el buffer serie de la PC. Por lo cual los datos presentados en pantalla no eran los actuales sino valores que habían quedado almacenados previamente. Este efecto era acumulativo por lo cual la respuesta del objeto era errática hasta que el sensor se encontrase durante un período de tiempo una posición estable.

El filtrado de las señales queda como trabajo futuro al proyecto como se especificó en el apartado anterior.

En la Figura ZZ se muestra el diagrama en bloques asociado a la estructura de programas utilizados para poder realizar la presentación del objeto junto con los movimientos en tiempo real dados por los cambios en el arreglo de sensores MPU 9250.

En este caso además de determinar los ángulos de Euler fue necesario crear matrices de rotación para cada uno de los ángulos obtenidos. Se obtienen tres matrices de 4 filas x 4 columnas que deben ser multiplicadas entre sí, el orden no es arbitrario porque se obtienen distintos resultados. Las ecuaciones relacionadas al procedimiento descrito pueden notarse a continuación[\[link\]](#):

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicando las matrices obtenidas de la siguiente forma se obtiene la matriz de rotación generalizada R :

$$R = R_z(\phi)R_y(\theta)R_x(\psi)$$

$$= \begin{bmatrix} \cos \theta \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \cos \theta \sin \phi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi \\ -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta \end{bmatrix}$$

La representación del objeto en tres dimensiones actuando a los movimientos del sensor se pueden ver en las Figura 21. Para poder notar en video el comportamiento del mismo dirigirse al siguiente [link](#). El comportamiento del sistema desarrollado es correcto en la mayor parte del tiempo pero se pudo determinar un comportamiento errático en algunas situaciones o errores en la representación del objeto frente a cambios abruptos. Todos estos inconvenientes vienen aparejados a la necesidad de la utilización de un filtro para tasas de refresco de datos bajas.

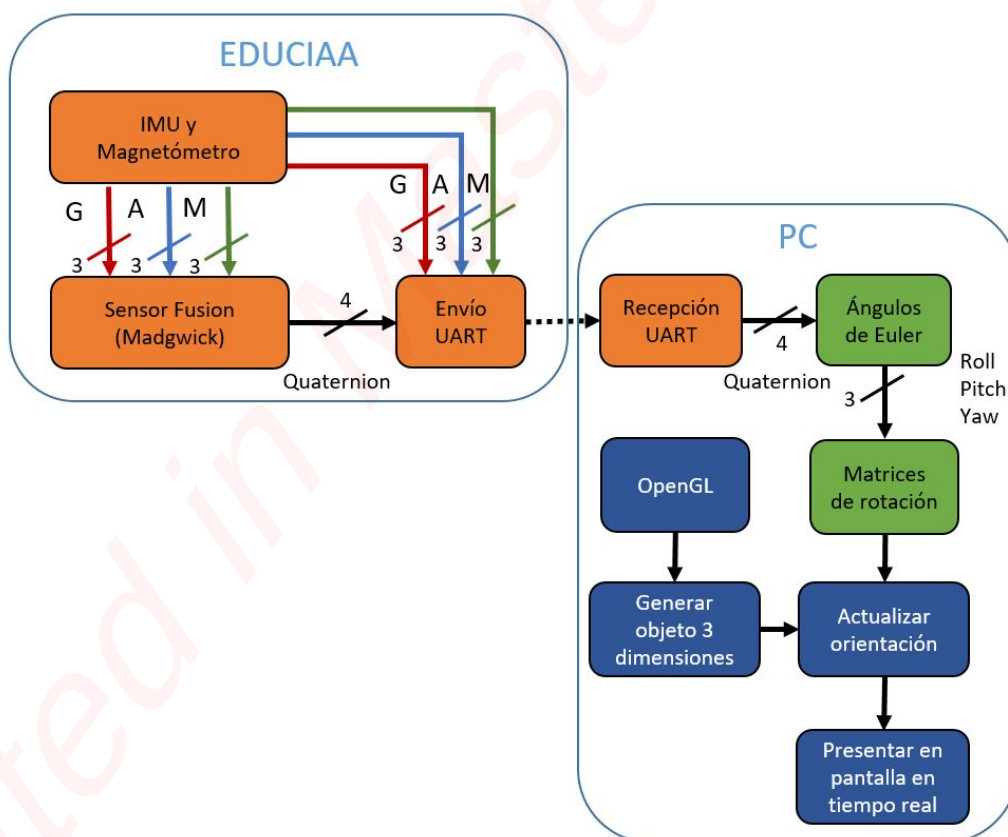


Figura 20: Diagrama en bloques de los programas utilizados para mostrar en tiempo real el movimiento de un cubo en tres dimensiones frente a cambios en el MPU 9250.



Figura 21: Programa funcionando en tiempo real rotando un cubo en tres dimensiones con respecto a cambios en el MPU 9250.