



Universidad Católica Argentina

Facultad de Ingeniería y Ciencias Agrarias - Carrera de Ingeniería Electrónica

TRABAJO FINAL DE GRADO

Sistema estabilizador de imagen - Aplicación del Filtro de Kalman en 2 ejes - implementado mediante un DSP de doble núcleo

Alumna: Trinidad Burs

Tutor: Mag. Ing. Ricardo Vecchio

Julio 2022

AGRADECIMIENTOS

Quiero agradecer al Ing. Ricardo Vecchio por su apoyo y guía no solo durante este proyecto, pero durante toda la carrera. Por sus consejos, dedicación y esfuerzo. Por transmitirnos su pasión por la electrónica y la teoría de control.

Una agradecimiento especial también al Ing. Diego Lazcano Colodrero por su ayuda y entusiasmo durante la carrea. Muchas gracias por enseñarnos que lo mejor no se logra con un Arduino.

Quiero dedicar este trabajo a mi familia, especialmente a mis padres. Por su paciencia y por su apoyo incondicional, por siempre estar y por acompañarme en todas mis decisiones. No hubiese llegado a donde estoy hoy sin ellos. Muchas gracias a mis amigos, por escucharme y aguantarme durante todo este proceso.

INDICE

| | | |
|-------|---|----|
| I. | ALCANCE | 4 |
| II. | OBJETIVO..... | 5 |
| III. | INTRODUCCIÓN | 6 |
| IV. | COMPONENTES | 7 |
| 4.1 | dsPIC33CH Curiosity Development Board (DM330028-2) | 7 |
| 4.1.1 | PLL..... | 9 |
| 4.1.2 | PPS – Peripheral Pin Select | 10 |
| 4.1.3 | UART..... | 10 |
| 4.1.4 | I2C | 10 |
| 4.1.5 | MSI – Master Slave Interface | 12 |
| 4.1.6 | SCCP | 12 |
| 4.2 | Servos Futaba3003..... | 13 |
| 4.3 | MPU6050..... | 15 |
| 4.3.1 | Acelerómetro | 16 |
| 4.3.2 | Giróscopo | 17 |
| 4.3.3 | Mediciones..... | 18 |
| 4.3.4 | Calibración | 19 |
| 4.3.5 | Configuración..... | 19 |
| 4.4 | Conversor de nivel HW-221 | 20 |
| V. | MODELIZACIÓN..... | 21 |
| 5.1 | Servos..... | 21 |
| 5.2 | Gimbal | 25 |
| VI. | CONTROL..... | 27 |
| 6.1 | LQR – Linear Quadratic Regulator | 28 |
| 6.2 | Observador..... | 28 |
| 6.3 | Acción Integral | 29 |
| 6.4 | Implementación en MatLab y Simulaciones | 31 |
| 6.5 | Implementación en dsPIC | 33 |
| VII. | DETERMINACIÓN DE ORIENTACIÓN..... | 36 |
| 7.1 | Cálculo del Ángulo de Inclinación – Acelerómetro | 37 |
| 7.2 | Cálculo del Ángulo de Inclinación – Giróscopo | 39 |
| 7.3 | Cálculo del Ángulo de Inclinación – Filtro Complementario | 41 |

| | | |
|-------|---|----|
| 7.4 | Cálculo del Ángulo de Inclinación – Kalman..... | 45 |
| 7.4.1 | “Filter Tuning” | 49 |
| 7.4.2 | Implementación | 49 |
| VIII. | EVALUACIÓN | 53 |
| 8.1 | Plataforma de pruebas..... | 53 |
| 8.2 | Distribución del código en ambos núcleos..... | 53 |
| 8.3 | Ubicación del acelerómetro | 57 |
| 8.4 | Filtrado y Procesamiento de Datos | 58 |
| 8.4.1 | Filtro Pasa Bajos | 59 |
| 8.4.2 | Interferencia entre ejes..... | 59 |
| 8.5 | Kalman VS Complementario en Reposo..... | 61 |
| 8.6 | Costo del dispositivo | 63 |
| 8.7 | Para tener en cuenta en futuros desarrollos | 64 |
| IX. | CONCLUSION | 65 |
| X. | BIBLIOGRAFIA..... | 66 |
| XI. | ANEXOS | 68 |
| 11.1 | ANEXO A - Implementación Sistema en dsPIC | 68 |
| 11.2 | ANEXO B - Implementación del filtro de Kalman en MatLab | 71 |
| 11.3 | ANEXO C - Algunos Tiempos de Procesamiento | 72 |
| 11.4 | ANEXO D - Mecanismo de Bloqueo..... | 72 |

I. ALCANCE

Esta tesis tiene como objetivo diseñar, implementar y evaluar un sistema de estabilización y control en variable de estado para un gimbal de dos ejes de dinámica desacoplada. El sistema de control debe calcular un conjunto óptimo de salidas de control para estabilizar la plataforma en la posición en que se desee. Se considera que el gimbal va a ser utilizado por un persona en condiciones no extremas.

El diseño del sistema debe considerar las limitaciones de implementación, incluidos los requisitos de procesamiento y las limitaciones en oferta y costo de hardware. Actualmente, en la Argentina hay muchos problemas con las importaciones, razón por la cual hay muy poca oferta y variedad de productos electromecánicos. A su vez, los costos de productos especializados son muy elevados.

II. OBJETIVO

Lograr un desarrollo compacto con pocos puntos de soldadura, con alta estabilidad mecánica, gran capacidad de cómputo y bajo costo utilizando un DSC de doble núcleo.

Se desea poder modelizar el funcionamiento del sistema, obteniendo su espacio de estados, e implementando un control óptimo.

A su vez se desea comprobar el funcionamiento del filtro de Kalman y su eficacia. Se comparará su desempeño con el del filtro complementario.

Por último se construirá un prototipo experimental y se evaluarán los resultados obtenidos, comparándolos con las simulaciones.

III. INTRODUCCIÓN

En el presente proyecto se realiza el desarrollo electromecánico de hardware y software de un prototipo conocido comúnmente como "Gimbal de 2 ejes". Un gimbal es un estabilizador externo para fotografía y video, que se basa en un sistema electromecánico que permite amortiguar los movimientos que se producen al caminar o al moverse produciendo una imagen mucho más fluida.

En él se pueden montar cámaras, smartphones u otros objetos, y es común utilizarlo en drones de para corregir para estabilizar y corregir vibraciones.

Se implementará un algoritmo LQR para el control del sistema en variables de estado y se utilizará el algoritmo de Kalman para minimizar el ruido de las mediciones.

IV. COMPONENTES

El sistema desarrollado consta de los siguientes componentes

- dsPIC33CH Curiosity Development Board (DM330028-2). *Cantidad:1*
- servos Futaba3003. *Cantidad:2*
- MPU6050. *Cantidad:1*
- Conversor de nivel HW-221. *Cantidad:1*

La estructura del gimbal se imprimirá utilizando técnicas 3D

4.1 dsPIC33CH Curiosity Development Board (DM330028-2)

Es un placa de desarrollo construida alrededor del dsPIC33CH512MP508. La placa incluye un programador/debugger integrado y no requiere hardware adicional. Opera con una tensión de 3.3V-3.6V.

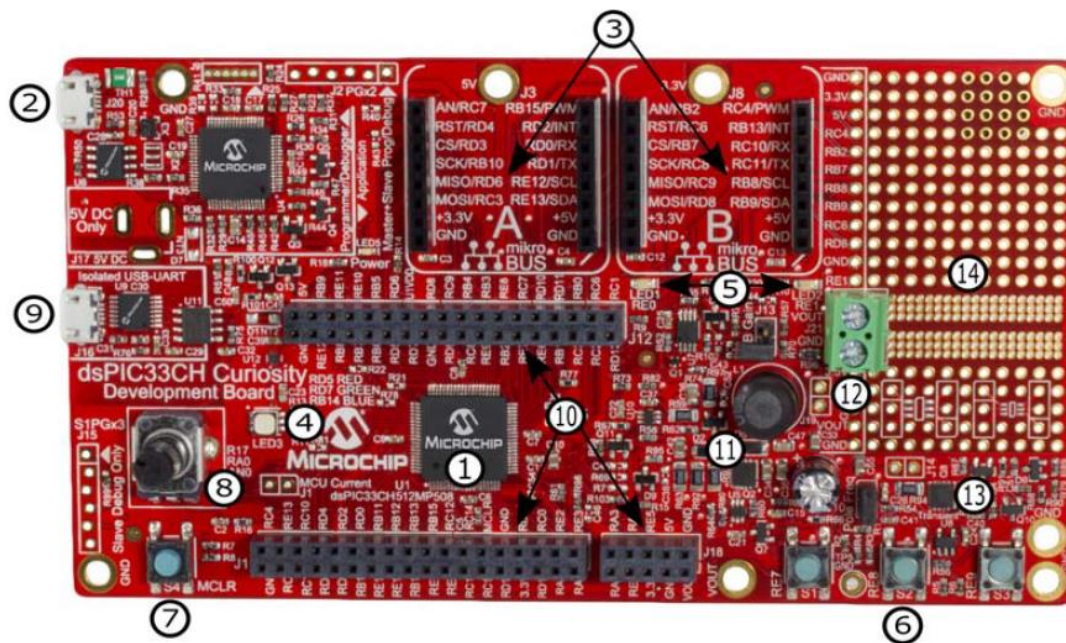


Imagen 1. dsPIC33CH Curiosity Development Board (DM330028-2). Parte 1

Hardware Features:

1. dsPIC33CH512MP508 dual core, 16-bit DSP target device.
2. Integrated PICkit™-On-Board (PKOB) programmer/debugger.
3. 2x mikroBUS™ interfaces for hardware expansion, compatible with a wide range of existing click boards™ from MikroElektronika (www.mikroe.com).
4. 1x Red/Green/Blue (RGB) LED.
5. 2x general purpose red indicator LEDs.
6. 3x general purpose push buttons.
7. 1x MCLR Reset push button.
8. 10k potentiometer.
9. Galvanically isolated USB-UART interface, capable of up to 460,800 baud.
10. Female, 100 mil pitch, I/O pin access headers for probing and connecting to all target microcontroller GPIO pins.
11. Configurable Switch Mode Power Supply (SMPS) test circuit that can be operated in Buck, Boost, or Buck-Boost modes, using either Voltage mode or Peak Current mode control.
12. Converter output voltage screw terminal.
13. Configurable load step transient generator.
14. General purpose through-hole and SMT prototyping area.

Imagen 2. dsPIC33CH Curiosity Development Board (DM330028-2). Parte 2

El dsPIC33CH512MP508 es un 16-bit DSC (digital signal controller - controlador de señal digital) de doble núcleo. El núcleo primario -“Master”- puede trabajar a 90 MIPS¹ y el núcleo secundario -“Slave”- es capaz de trabajar a 100 MIPS. Esto brinda una gran capacidad de cómputo.

Los DSC de la familia dsPIC33CH son ideales para muchas aplicaciones robustas y de uso general de alto rendimiento.

En este proyecto se dedicará el módulo “Master” a la comunicación, sensado, filtrado y procesamiento de datos; mientras que el algoritmo de control del sistema en variables de estado, la implementación del observador y el manejo de los servos se encontraran en el módulo “Slave”.

Al estar ambos módulos en el mismo chip y utilizar un sistema de comunicación conocido como MSI, la comunicación entre ambos es eficiente y robusta.

¹ MIPS: “Million Instructions Per Second” o millones de instrucciones por segundo

4.1.2 PPS – Peripheral Pin Select

La selección de pin en periféricos proporciona una alternativa al permitir la selección y colocación de conjuntos de periféricos en una amplia gama de pines de entrada y salida. Al aumentar las opciones de asignación de pines disponibles en un dispositivo en particular, los usuarios pueden adaptar mejor el dispositivo a su aplicación, en lugar de tener que adaptar la aplicación para que se ajuste al dispositivo. La configuración de PPS opera sobre un subconjunto fijo de pines de I/O digitales. Los usuarios pueden asignar de forma independiente la entrada y/o salida de la mayoría de los periféricos digitales a cualquiera de estos pines.

Para poder realizar esto de forma controlada se incluyen salvaguardas de hardware que evitan accidentes o cambios bruscos en el mapeo de los periféricos.

FIGURE 3-24: REMAPPABLE INPUT FOR U1RX

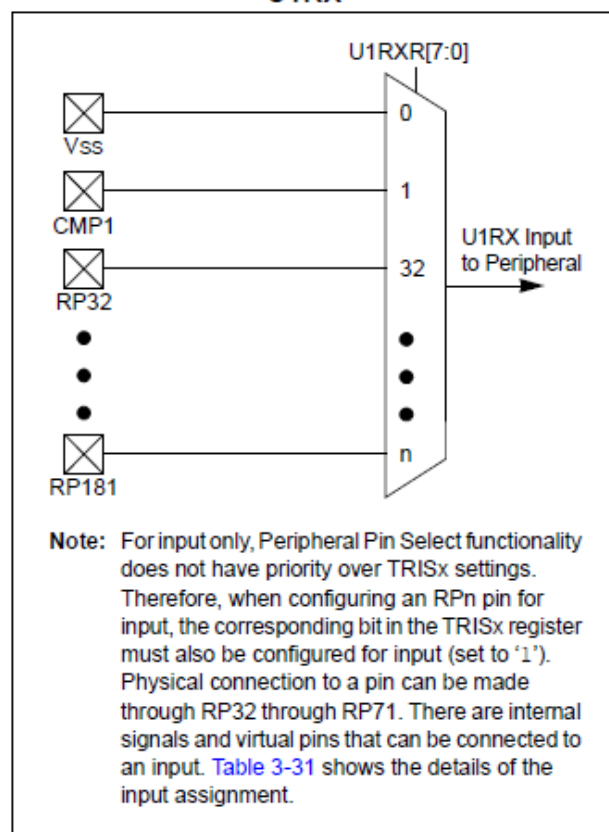


Imagen 4. Funcionamiento del módulo PPS

4.1.3 UART

En un inicio se consideró usar el módulo bluetooth HM-10 pero al final se decidió utilizar la interfaz USB-UART de la placa ya que está aislada galvánicamente (lo que reduce muchísimo el ruido) alcanzando velocidades muy altas de esta 460800 baudios. Esto hace a la rápida ejecución del programa.

4.1.4 I2C

Este es el protocolo necesario para comunicarse con el MPU6050. I2C es un protocolo de comunicación en serie, por lo que los datos se transfieren bit a bit a lo largo de un cable. I2C es un

protocolo síncrono, por lo que los bits de salida están sincronizados con el clock compartido por el maestro y el esclavo. La señal del reloj siempre está controlada por el maestro.

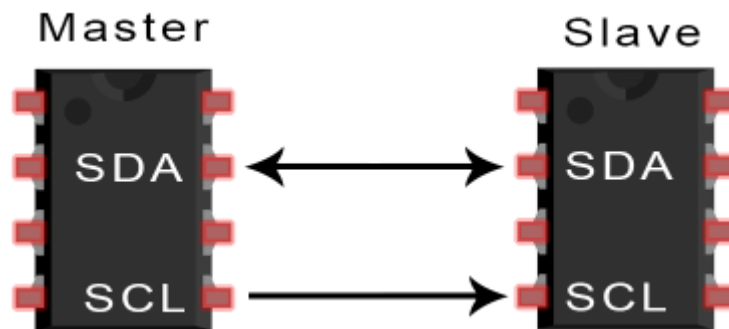


Imagen 5. Conexión entre Master y Esclavo en I2C

- SDA (Serial Data) – Línea de transmisión de datos
- SCL (Serial Clock) – Línea con la señal de reloj

Los datos se transfieren en mensajes. Los mensajes se dividen en frames de datos. Cada mensaje tiene frame de dirección que contiene la dirección binaria del esclavo y una o más frames de datos que contienen los datos que se transmiten. El mensaje también incluye condiciones de inicio y parada, bits de lectura/escritura y bits ACK/NACK entre cada frame.

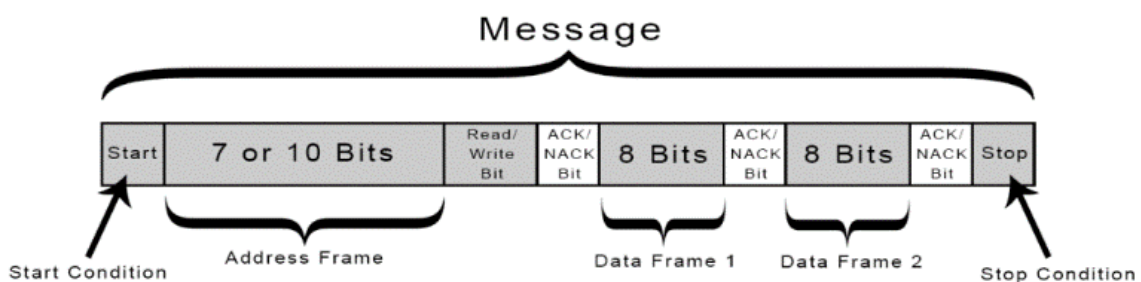


Imagen 6. Mensaje I2C

El maestro envía la dirección del esclavo con el que quiere comunicarse a todos los esclavos conectados a él. Luego, cada esclavo compara la dirección enviada desde el maestro con su propia dirección. Si la dirección coincide, envía un bit ACK al maestro. Si la dirección no coincide, el esclavo no hace nada y la línea SDA permanece alta.

Al frame de dirección le sigue un bit al final que informa al esclavo si el maestro quiere escribirle datos o recibir datos de él. Si el maestro quiere enviar datos al esclavo, el bit de lectura/escritura es un nivel de voltaje bajo. Si el maestro solicita datos del esclavo, el bit es un nivel de alto voltaje.

| Ventajas | Desventajas |
|--|--|
| <ul style="list-style-type: none"> • Solo usa dos cables • Soporta múltiples maestros y múltiples esclavos • El bit ACK/NACK confirma que cada cuadro se transfiere con éxito | <ul style="list-style-type: none"> • Tasa de transferencia de datos más lenta que SPI • El tamaño de la trama de datos está limitado a 8 bits. • Se necesita hardware más complicado de implementar que SPI |

- El hardware es menos complicado comparado con el requerido para implementar un protocolo UART
- Es un protocolo bien conocido y ampliamente utilizado.

Tabla 1. Comparación de las ventajas y desventajas de I2C, SPI y UART

4.1.5 MSI – Master Slave Interface

Este módulo es un puente entre el procesador primario y secundario, cada uno de los cuales opera con su clock independiente.

Se espera que los procesadores primario y secundario operen a velocidades de reloj significativamente diferentes, por lo que el módulo MSI también incluye sincronización de datos y señales que se intercambian entre dominios de reloj.

Existen dos métodos de transferencia de datos entre el Maestro y el Esclavo:

- Transferencia basada en mailbox → Método más confiable
- Transferencia basada en FIFO → Ofrece un mayor throughput pero tienen que manejarse errores y es menos confiable

Se elegirá la transferencia basada en mailboxes ya que se prioriza la robustez de la aplicación.

A grandes rasgos la transferencia basada en mailboxes, consta de 16 pipes de datos independientes, unidireccionales, estilo mailbox. Las direcciones de los pipes es seleccionable.

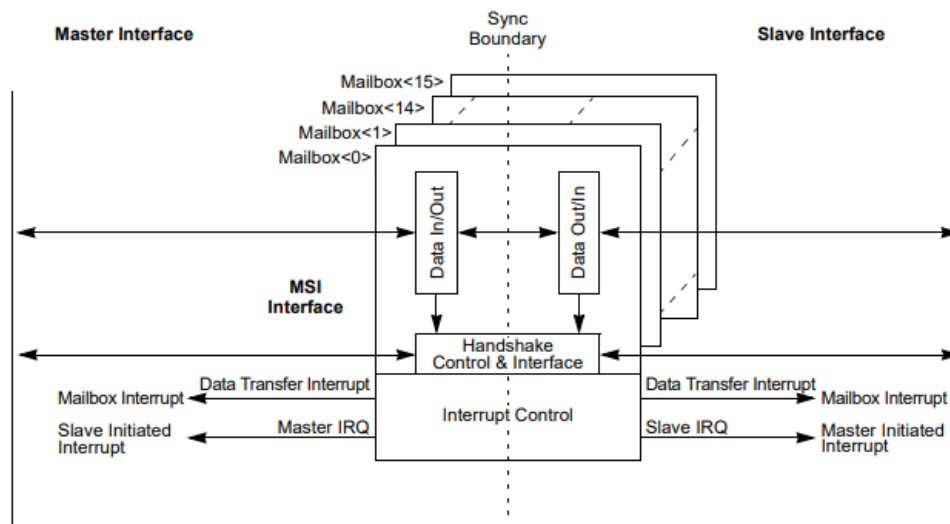


Imagen 7. MSI Diagrama en Bloques

4.1.6 SCCP

Se utilizó el módulo SCCP para generar las señales de control PWM. No se usó el módulo PWM ya que no era capaz de generar la frecuencia necesaria para controlar los servos, los servos funcionan a una frecuencia demasiado baja relativa al módulo PWM.

El módulo SCCP se configuró para trabajar en el modo "Dual Edge Compare (16-bit buffered)", este modo permite generar una señal PWM.

4.2 Servos Futaba3003

Se eligió este servo ya que satisface los requerimientos electromecánicos requeridos, es de bajo costo y está ampliamente disponible en el mercado local.

| Detailed Specifications | | | |
|------------------------------|--|------------------------|---|
| Control System: | ±Pulse Width Control 1520usec Neutral | Current Drain (4.8V): | 7.2mA/idle |
| Required Pulse: | 3-5 Volt Peak to Peak Square Wave | Current Drain (6.0V): | 8mA/idle |
| Operating Voltage: | 4.8-6.0 Volts | Direction: | Counter Clockwise/Pulse Traveling 1520-1900usec |
| Operating Temperature Range: | -20 to +60 Degree C | Motor Type: | 3 Pole Ferrite |
| Operating Speed (4.8V): | 0.23sec/60 degrees at no load | Potentiometer Drive: | Indirect Drive |
| Operating Speed (6.0V): | 0.19sec/60 degrees at no load | Bearing Type: | Plastic Bearing |
| Stall Torque (4.8V): | 44 oz/in. (3.2kg.cm) | Gear Type: | All Nylon Gears |
| Stall Torque (6.0V): | 56.8 oz/in. (4.1kg.cm) | Connector Wire Length: | 12" |
| Operating Angle: | 45 Deg. one side pulse traveling 400usec | Dimensions: | 1.6" x 0.8"x 1.4" (41 x 20 x 36mm) |
| 360 Modifiable: | Yes | Weight: | 1.3oz. (37.2g) |

Tabla 2. Especificaciones técnicas del servo Futaba S3003

Un servo es un motor que cuenta con un sistema de realimentación (encoder), el cual le indica al driver del servo la posición en la que se encuentra el eje, permitiendo la corrección en tiempo real de la posición.

Además de la precisión, otra de las propiedades deseables de los servos es la capacidad de mantener un torque constante en toda su gama de revoluciones. Un servomotor aplica todo su torque disponible para conservar la posición de la carga.

Un servomotor está compuesto por:

- Un motor de corriente continua
- Un sistema de control: controla el movimiento del motor mediante el envío de pulsos
- Un sistema reductor: formado por engranajes gracias a los cuales puede aumentar el torque.
- Un potenciómetro: se encuentra conectado al eje central y permite saber en todo momento el ángulo en el que se encuentra el eje del motor.

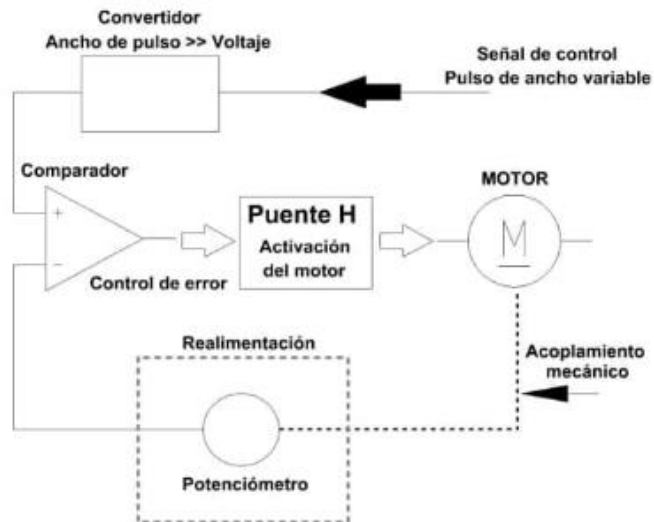


Imagen 8. Diagrama en bloques del interior de un servo

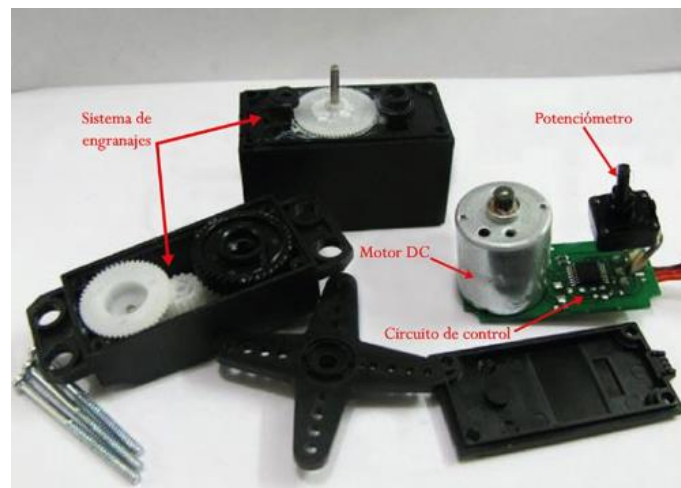


Imagen 9. Partes de un servo

El servo motor Futaba3003 es controlado con una señal PWM.

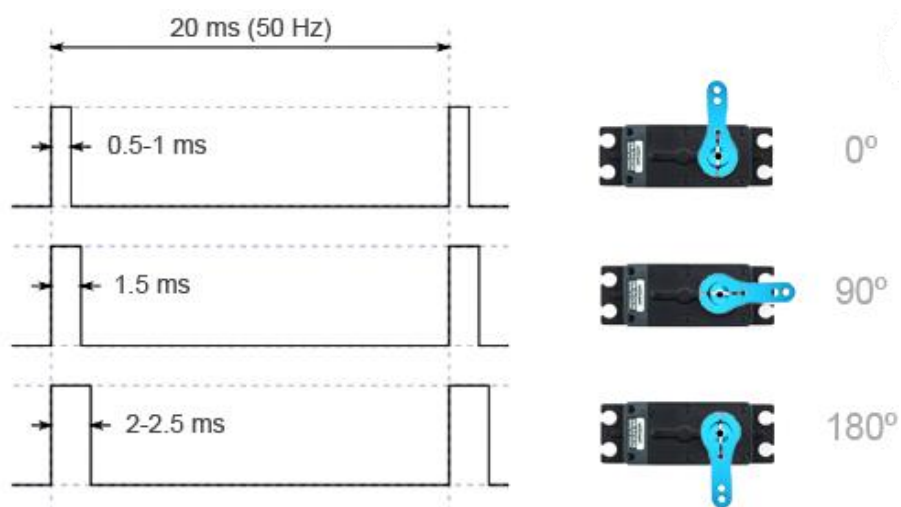


Imagen 10. Señal de PWM comúnmente utilizada para control los servos

A pesar de que en la mayoría de la documentación encontrada se especifica que el periodo de la señal PWM debe ser de 20ms, se realizaron pruebas que demostraron que se los puede controlar de manera eficiente y eficaz empleando un periodo de 5ms. Lo que importa es el tiempo en “ON” de la señal.

4.3 MPU6050

El MPU 6050 es un sensor IMU de 6 DOF (grados de libertad), lo que significa que proporciona seis valores como salida: tres valores del acelerómetro y tres del giroscopio. El MPU 6050 es un sensor basado en tecnología MEMS (sistemas microelectromecánicos). Tanto el acelerómetro como el giroscopio están integrados dentro de un solo chip. El chip es el GY-521. Utiliza I2C para la comunicación.

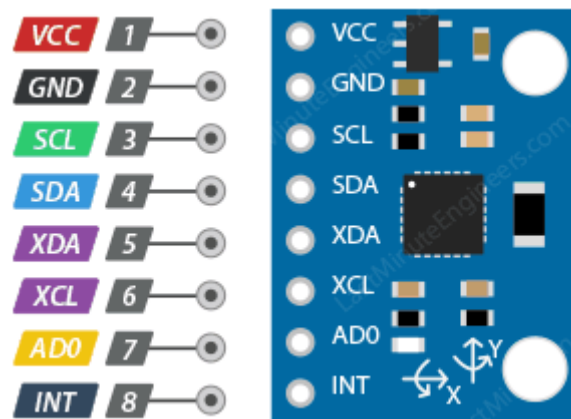


Imagen 11. Diagrama MPU6050

La principal desventaja del GY-521 es que tiene un alto nivel de ruido. A pesar de esto se eligió este sensor ya que era el único disponible en el mercado.

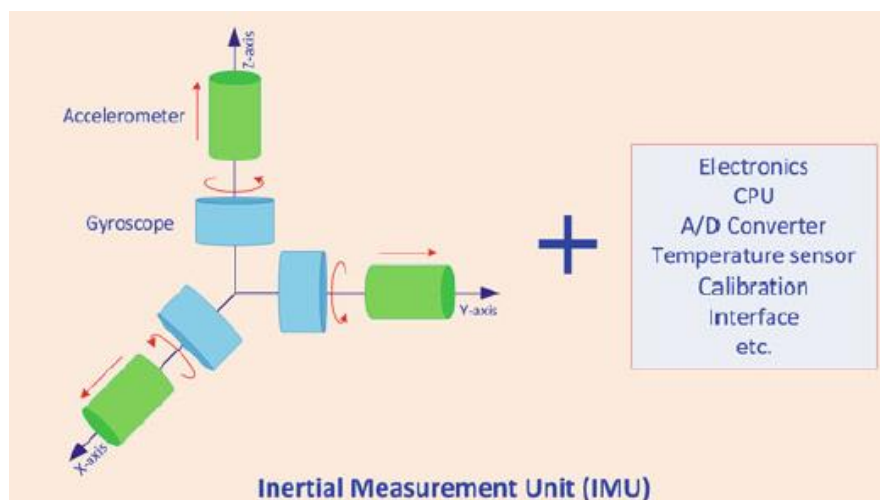


Imagen 12. Diagrama básico de una IMU

MEMS, o Micro Electro-Mechanical System, es una tecnología en la cual los sensores constan de una masa suspendida entre un par de placas capacitivas. De esta manera cuando se produce un movimiento, esta masa suspendida crea una diferencia de potencial eléctrico. La diferencia creada se mide como un cambio en la capacitancia.

Los sensores MEMS son sensores inerciales que tienen un bajo número de piezas, son relativamente baratos de fabricar y son de pequeño tamaño. La principal desventaja es que son menos precisos que dispositivos de otras tecnologías (ej: ópticos). Generalmente tienen mayor nivel de ruido.

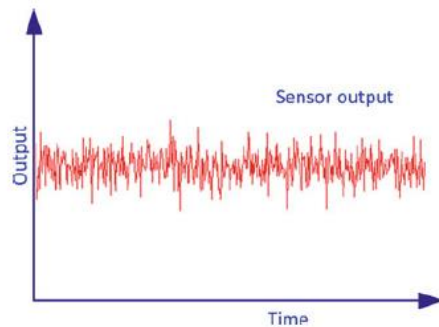


Imagen 13. Medición contaminada por ruido blanco

4.3.1 Acelerómetro

En aquellas aplicaciones donde la fuerza o aceleración neta sobre un sistema a lo largo del tiempo es la gravedad, se puede usar un acelerómetro para medir el ángulo estático de inclinación. Tales aplicaciones incluyen juegos, detección de horizonte en cámaras digitales y detección del rumbo de un dispositivo en aplicaciones industriales y médicas.

El acelerómetro MEMS (Micro Electro Mechanical Systems) consiste en una estructura micromecanizada construida sobre una oblea de silicio. Esta estructura está suspendida por resortes que permiten que la estructura se desvíe en el momento en que se aplica la aceleración en el eje.

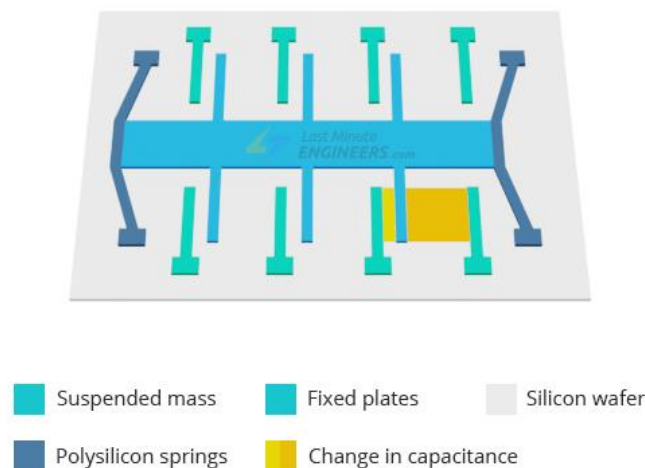


Imagen 14. Diagrama Acelerómetro

Cuando esto ocurre la capacitancia entre las placas fijas y las placas unidas a la estructura suspendida cambia. Este cambio en la capacitancia es proporcional a la aceleración en ese eje. A medida que el dispositivo se mueve, un ADC registra y convierte el cambio en la capacitancia en un valor digital que varía entre -32750 y 32750.

La magnitud de la señal en el caso del acelerómetro está sesgada por la gravedad. Este no es el caso del giróscopo.

4.3.2 Gir6scopo

Los giroscopios MEMS utilizan el efecto Coriolis para medir la rotaci3n angular, que establece que en un marco de referencia rotatorio (a velocidad angular ω), una masa m que se mueve con velocidad v experimenta una fuerza:

$$F_c = -2m (\omega \times v)$$

Los giroscopios MEMS contienen elementos vibratorios para medir el efecto Coriolis. La geometr3a m3s simple de dichos elementos vibratorios consta de una sola masa que es impulsada a vibrar a lo largo de un eje de transmisi3n. Cuando se gira el giroscopio, se induce una vibraci3n secundaria a lo largo del eje de sentido perpendicular debido a la fuerza de Coriolis. La velocidad angular se puede calcular midiendo esta rotaci3n.

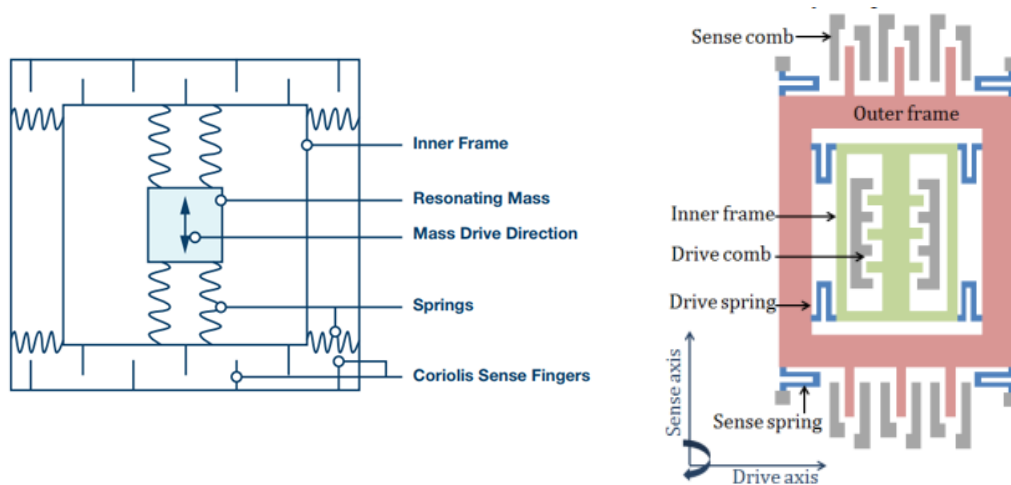


Imagen 15. Diagramas de la estructura mec3nica de un gir6scopo

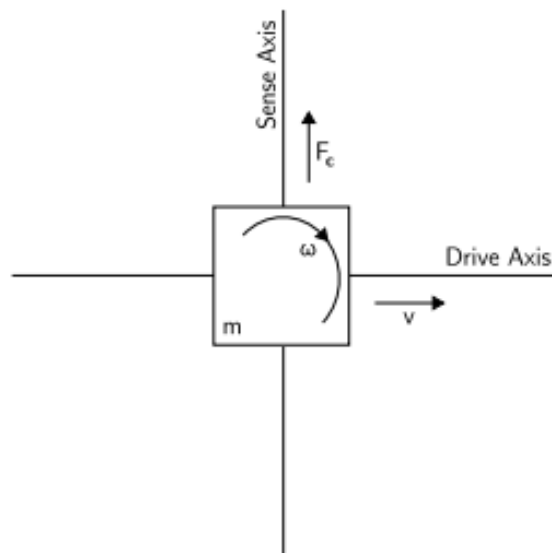


Imagen 16. Diagrama simplificado de un gir6scopo

4.3.3 Mediciones

El MPU6050 puede medir la aceleración simultáneamente a lo largo de los 3 ejes (X,Y,Z) con cuatro rangos de escala completamente programables de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.

| AFS_SEL | Full Scale Range | LSB Sensitivity |
|---------|------------------|-----------------|
| 0 | $\pm 2g$ | 16384 LSB/g |
| 1 | $\pm 4g$ | 8192 LSB/g |
| 2 | $\pm 8g$ | 4096 LSB/g |
| 3 | $\pm 16g$ | 2048 LSB/g |

Tabla 3. Modos de configuración del acelerómetro

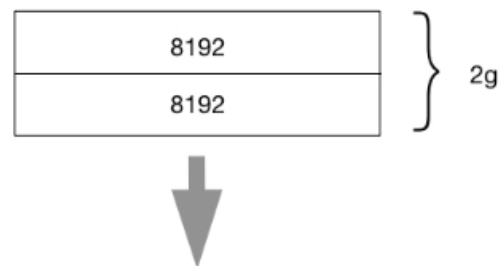
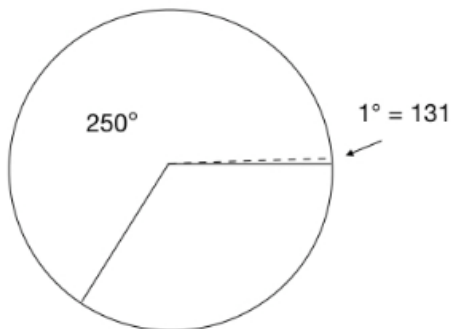
El MPU6050 puede medir la rotación angular simultáneamente alrededor de los ejes (X,Y,Z) con cuatro rangos de escala completa programables de $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$ y $\pm 2000^\circ/s$.

| FS_SEL | Full Scale Range | LSB Sensitivity |
|--------|--------------------|----------------------|
| 0 | $\pm 250^\circ/s$ | 131 LSB/ $^\circ/s$ |
| 1 | $\pm 500^\circ/s$ | 65.5 LSB/ $^\circ/s$ |
| 2 | $\pm 1000^\circ/s$ | 32.8 LSB/ $^\circ/s$ |
| 3 | $\pm 2000^\circ/s$ | 16.4 LSB/ $^\circ/s$ |

Tabla 4. Modos de configuración del giróscopo

Gyro measurement units = 32,750
 $32,750 / \text{degrees} = \text{sensitivity}$
 e.g. $32,750 / 250^\circ =$
 131 measurement units per degree

Accelerometer measurement units = 32,750
 $32,750 / g = \text{sensitivity}$
 e.g. $32,750 / 2g = 16,384$



| Angular Velocity Limit | Sensitivity |
|------------------------|-------------|
| 250°/s | 131 |
| 500°/s | 65.5 |
| 1000°/s | 32.8 |
| 2000°/s | 16.4 |

| Acceleration Limit | Sensitivity |
|--------------------|-------------|
| 2g | 16,384 |
| 4g | 8,192 |
| 8g | 4,096 |
| 16g | 2,048 |

Imagen 17. Cálculo de sensibilidad

4.3.4 Calibración

Antes de utilizar la IMU esta debe calibrarse. La IMU montada en la plataforma no estará perfectamente alineada con el suelo (o por defectos e imperfecciones durante el montaje del sensor en el módulo), por lo que se deben tomar una serie de mediciones del acelerómetro y el giroscopio para calcular dicha desviación ("offset"). Desde una perspectiva física, la compensación de offsets proporciona una translación del "Body Frame" (plano en el cual está montado la IMU) al "Inertial Frame" (marco respecto al cual se efectúan todos los cálculos).

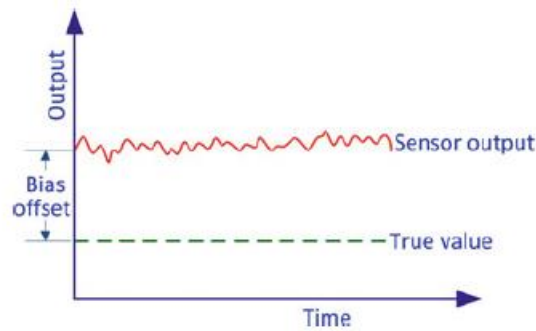


Imagen 18. Efecto de componente de continua ("offset") en una medición

En general, cuanto menor sea el error de offset (o "bias"), menores serán los errores al integrar la salida del giroscopio a lo largo del tiempo. Es decir que será menor el "drift" de la señal. Un giroscopio con menor error de bias conducirá a menores errores en las estimaciones de posición a lo largo del tiempo para una unidad de medida inercial.

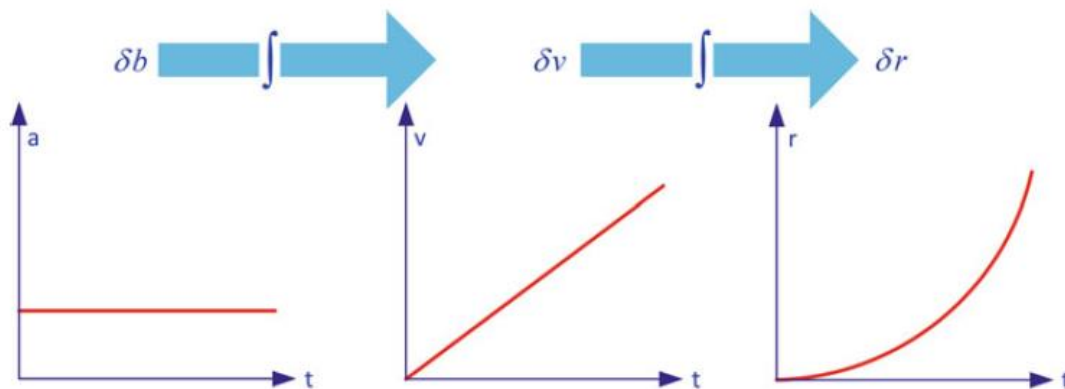


Imagen 19. Efecto de integrar una componente de continua (en aceleración, velocidad y posición)

4.3.5 Configuración

Se configuro el módulo de la siguiente manera:

- Rango Acelerómetro: +/- 2g
- Rango Giroscopio: +/- 250 °/s
- Frecuencia de muestreo del Acelerómetro : 1kHz
- Frecuencia de muestreo del Giroscopio : 1kHz
- DLPF deshabilitado ya que introduce un retardo grande y variable

- Se habilito la generación de interrupción cada vez que haya nuevos datos. Se generarán interrupción al dsPIC cada 1ms
- No se utiliza el módulo DMP de la IMU ya que no hay información acerca de cómo debe configurarse, se podría utilizar código encontrado en internet pero se va a priorizar la transparencia y autoría del código en este trabajo.

4.4 Conversor de nivel HW-221

Dispositivo que permite convertir de manera bidireccional voltajes de 5V y 3,3V. Esto se hace porque la placa trabaja a una tensión de 3,3V, mientras que los servos son alimentados con un voltaje de 5V. Se lo utiliza para poner una barrera entre la placa y el servo (y así proteger a la primera) y para elevar de nivel las señales PWM.

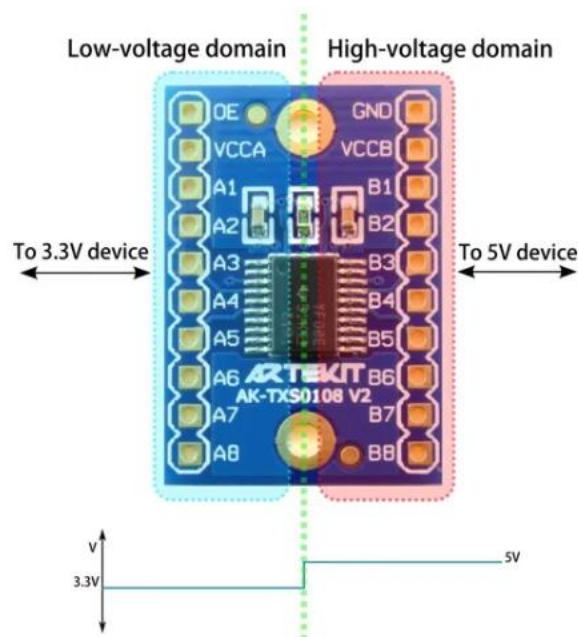


Imagen 20. Conversor de nivel HW221

V. MODELIZACIÓN

La modelización del gimbal se realizó de forma empírica. Se comenzó modelando los servos.

5.1 Servos

En vez de recurrir a modelos matemáticos basados en principios mecánicos y cinemáticos se hizo lo siguiente: se acoplo un potenciómetro al eje del servo y se aplicó una señal conocida al servo (señal PWM que generaba una rotación de 50°).

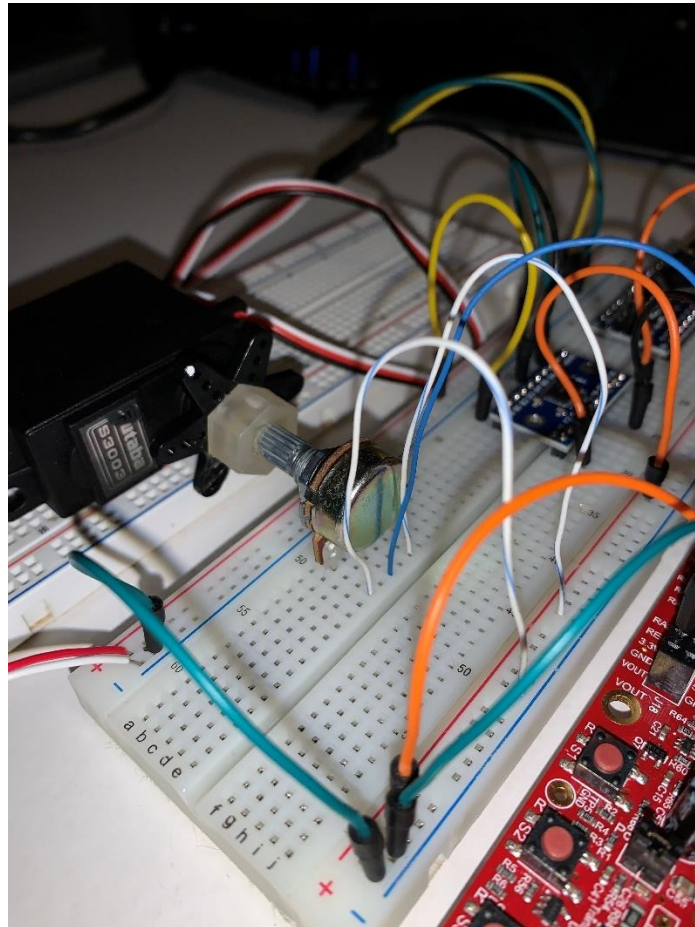


Imagen 21. Potenciómetro acoplado al eje del servo

Con el módulo ADC del dsPIC33CH se observó la variación de tensión sobre los pines del potenciómetro. Se utilizó una velocidad de muestreo de $50\mu s$. Esto se repitió numerosas veces para ambos sentidos de rotación. Se observó que el servo se comporta ligeramente diferente cuando gira en distintos sentidos. Como observación se debe aclarar que un aumento en el valor registrado por el ADC representa una disminución del ángulo de rotación del servo.

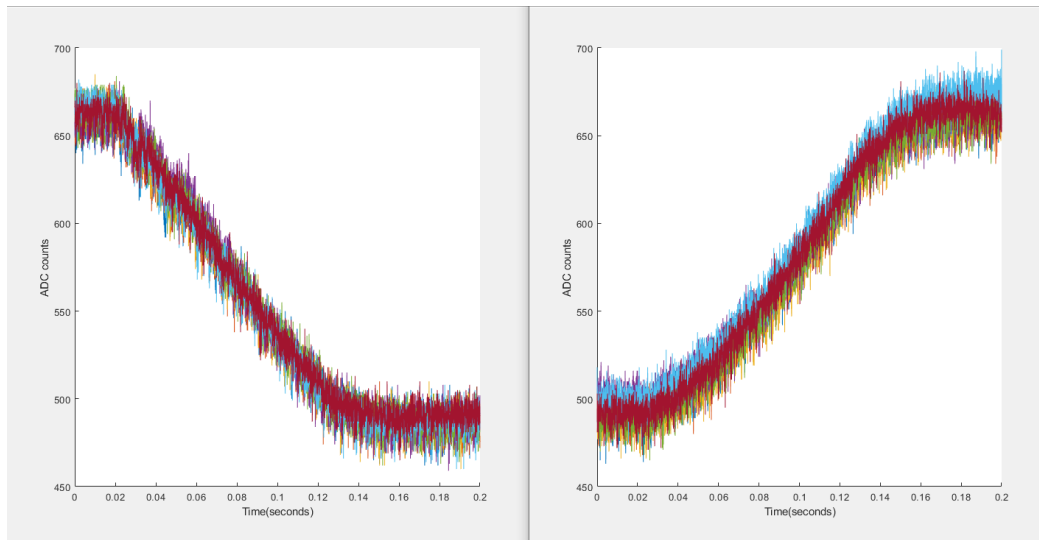


Imagen 22. Curvas obtenidas con el ADC al aplicar una señal PWM al servo que generaba una rotación de $\pm 50^\circ$

Para reducir el ruido de las mediciones se hizo un promedio entre todas las pasadas. Se decidió que la entrada del sistema a modelar será la anchura del pulso y la salida el desplazamiento en grados. Se adaptaron los datos de salida del ADC para que representaran grados en vez de cuentas, se invirtió una de las curvas (para facilidad de visualización), se eliminó el offset y se removieron las muestras antes de que se aplicara la señal PWM. También se creó una función escalón que adquiere el valor el ancho del pulso (en segundos). Estas señales fueron las utilizadas para identificar la planta.

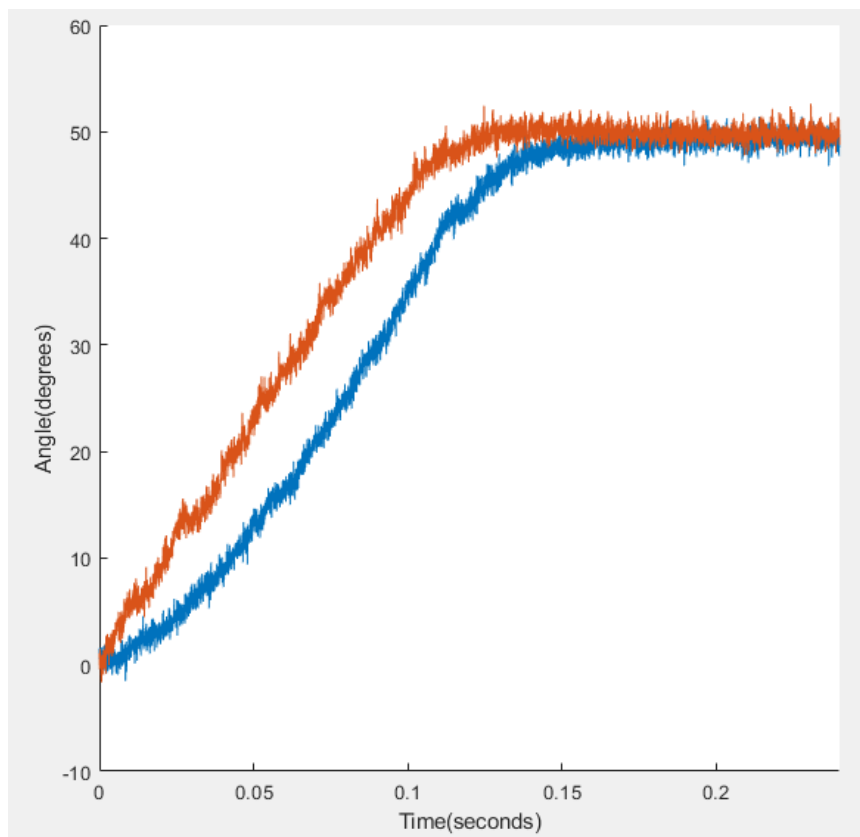


Imagen 23. Curvas de la imagen anterior luego de promediarse, invertirse una de ellas, removerse su offset y adaptarse su escala (cuentas del ADC \rightarrow grados $^\circ$)

Luego se utilizó el “System Identification Toolbox” de MatLab para obtener la función transferencia de la planta. Se plantearon varias posibles transferencias de diverso orden y se terminó eligiendo la que se consideró que mejor se adaptaba.

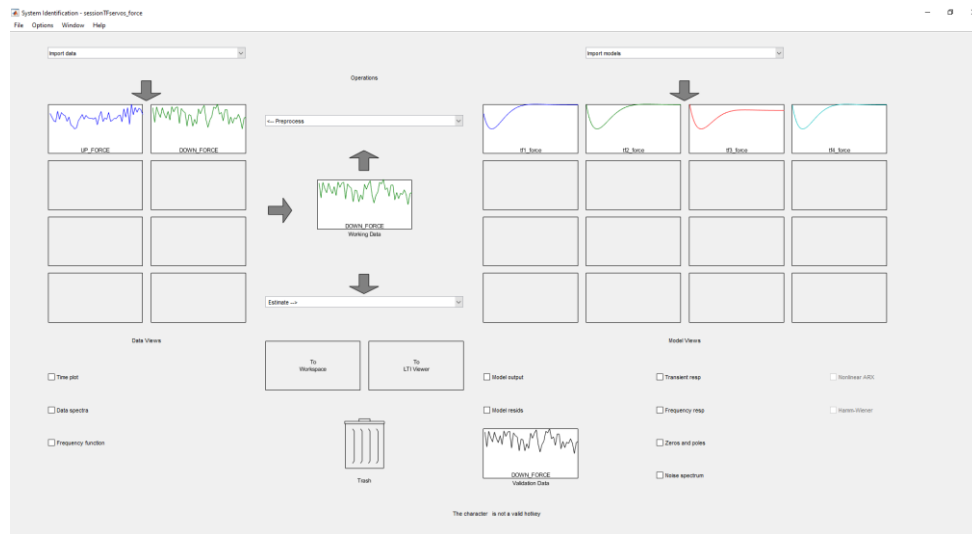


Imagen 24. System Identification Toolbox

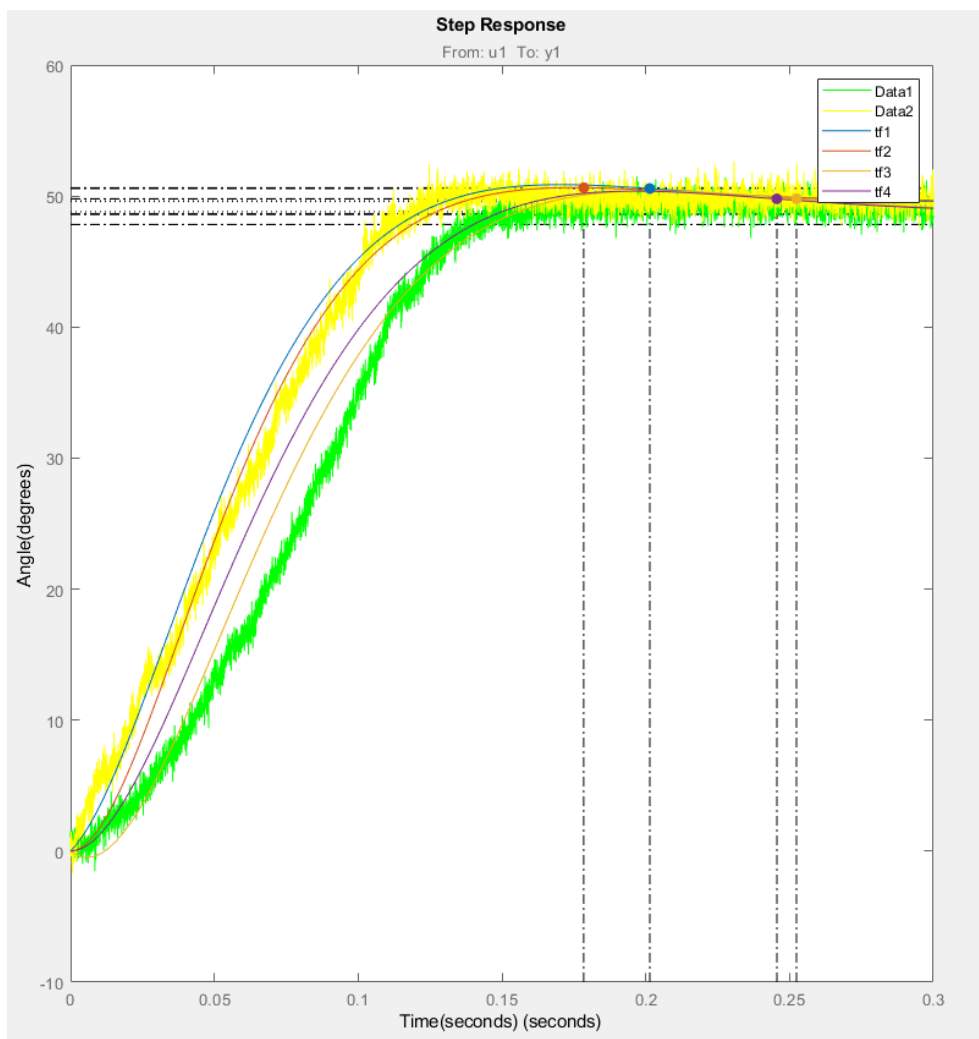


Imagen 25. Posibles transferencias

La “tf2” se descartó porque era la de menor settling time, por lo que estaría siendo optimista si eligiese esta. La “tf3” tampoco se tomó en consideración porque comienza tomando valores negativos. La ventaja de la “tf1” es que posee un menor settling time, mientras que la ventaja de la “tf4” es que no posee zeros. Luego de numerosas pruebas con los motores en las cuales se evaluó visualmente su respuesta se eligió la “tf4”. A su vez esta representa un mejor compromiso entre las curvas de ambos sentido de giro.

```
num=4.951180259830461e+07;  
den = [1 35.061338538384340 5.635617959610166e+02];  
[A,B,C,D] = tf2ss(num,den);
```

Imagen 26. Construcción del espacio de estados a partir de la transferencia elegida (tf4)

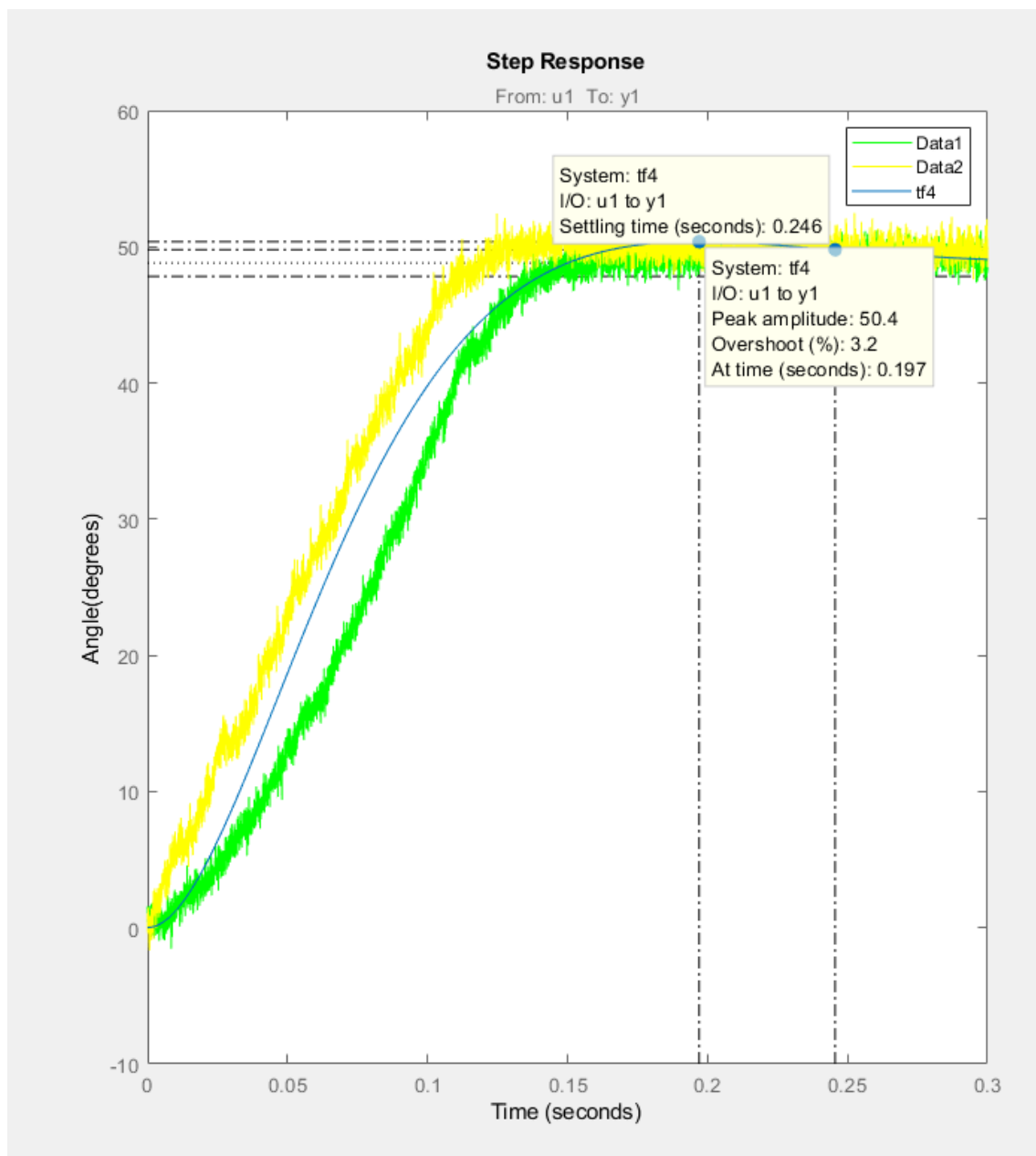


Imagen 27. Transferencia elegida para el servo

5.2 Gimbal

En cualquier sistema de transmisión existirá inercia, tanto del motor como la de la carga . Para un sistema de accionamiento directo (una conexión directa del motor a la carga), las inercias individuales se pueden sumar de forma sencilla. Sin embargo, para cualquier otro tipo de configuración que involucre algunos componentes de accionamiento mecánico adicionales la relación cambia. En el caso de un motor reductor (ej: servomotor) cualquier engranaje reducirá la inercia de la carga reflejada en el motor por un factor del radio del engranaje al cuadrado.

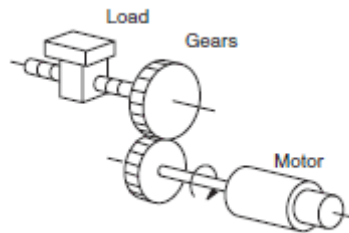


Imagen 28. Diagrama de carga-reducción-motor

De acuerdo con la relación de transmisión, la propia vibración del motor se puede reducir haciendo que esta no pase a la carga.

Los servos Futaba3003 cuentan con una reducción muy grande. Esto simplifica la modelización del sistema ya que se va a poder despreciar el efecto de la inercia de la carga sobre el motor. Esto también puede hacerse porque el peso de la estructura es bajo, por lo cual su inercia es muy pequeña. De esta manera la función transferencia del sistema será la función transferencia de los servos.

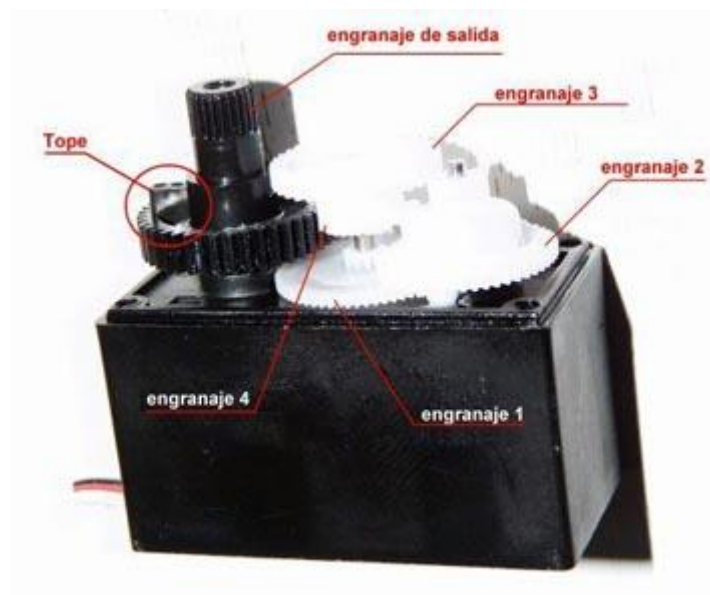


Imagen 29. Reducción del servo Futaba S3003

A esto también se suma que al contar con una gran reducción, los servomotores tendrán una gran capacidad de torque los que les permitirá mover fácilmente la estructura. La reducción aumenta el torque a coste de disminuir la velocidad del motor.

El espacio de estados del sistema será:

$$A_{sist} = \begin{bmatrix} A_{servo1} & 0 \\ 0 & A_{servo2} \end{bmatrix}$$

$$B_{sist} = \begin{bmatrix} B_{servo1} & 0 \\ 0 & B_{servo2} \end{bmatrix}$$

$$C_{sist} = \begin{bmatrix} C_{servo1} & 0 \\ 0 & C_{servo2} \end{bmatrix}$$

$$D_{sist} = \begin{bmatrix} D_{servo1} & 0 \\ 0 & D_{servo2} \end{bmatrix}$$

$$X_{sist} = \begin{bmatrix} \theta_1 \\ d\theta_1/dt \\ \theta_2 \\ d\theta_2/dt \end{bmatrix}$$

$$Y_{sist} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

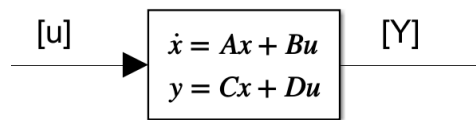


Imagen 30. Espacio de estados en Simulink

VI. CONTROL

La estabilidad del sistema está determinada por los autovalores de la matriz A (polos del sistema), si cualquiera de estos valores es positivo el sistema es inestable.

Un sistema es controlable si es posible llevarlo a cualquier estado desde cualquier estado inicial en un tiempo finito con una señal de actuación finita. Si el espacio de estados es completamente controlable (es decir, que todos los estados son controlables), el rango de la matriz de controlabilidad (C) es igual al orden del sistema (rango de A). La matriz A es una matriz de $n \times n$, la matriz B es una matriz de $n \times m$.

$$C = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

```
39 %% VERIFICACION DE CONTROLABILIDAD
40 - M_controlabilidad=ctrb(A,B);
41 - if (size(A,1)==rank(M_controlabilidad))
42 -     disp("El sistema es controlable")
43 - else
44 -     error("El sistema no es controlable!")
45 - end
```

```
Command Window
El sistema es controlable
fx >>
```

Imagen 31. Verificación de Controlabilidad en MatLab

Se dice que un sistema es observable en el tiempo si con el sistema en el estado $x(t)$, es posible determinar dicho estado a partir de las mediciones de entrada y salida en un lapso finito de tiempo. Si el espacio de estados es completamente observable, el rango de la matriz de observabilidad (O) es igual al orden del sistema. La matriz C es una matriz de $p \times n$.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

```
46 %% VERIFICACION DE OBSERVABILIDAD
47 - M_observabilidad=obsv(A,C);
48 - if (size(A,1)==rank(M_observabilidad))
49 -     disp("El sistema es observable")
50 - else
51 -     error("El sistema no es observable!")
52 - end
```

```
Command Window
El sistema es observable
fx >>
```

Imagen 32. Verificación de Observabilidad en MatLab

Los servos serán totalmente controlables y observables, por lo tanto el sistema también lo será.

Demostrada la controlabilidad y observabilidad del sistema se procede al diseño del controlador y del observador.

6.1 LQR – Linear Quadratic Regulator

El control optimo es una teoría que tiene como propósito la maximización o la minimización de un índice de desempeño que tenga que ver con algún proceso. Es decir, lo que se pretende es encontrar unas condiciones de funcionamiento óptimas para el sistema, la cual está sujeta a determinada función de costo (índice de desempeño) y que debe cumplir con un conjunto de restricciones dependiendo del sistema a optimizar. En el caso particular del controlador LQR, este índice de desempeño es un índice cuadrático y se lo define de la siguiente manera:

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \quad \text{siendo} \quad u = -Kx$$

K es la matriz de ganancias optimas del lazo. Las matrices Q y R ponderan el costo de convergencia del sistema y el costo de actuación, respectivamente. Estas matrices suelen ser diagonales y los elementos diagonales pueden ajustarse para cambiar la importancia relativa de los objetivos de control. Lo que importa son sus valores relativos. Si R adquiere valores mayores a Q , esto indica que el costo de actuación (el esfuerzo de control) será mayor, luego se intentará minimizar este y como consecuencia el sistema convergirá más lento. Si fuese al revés, se implementará un control más agresivo priorizando la convergencia. R es definida positiva y Q es semidefinida positiva. N es la matriz de costos que plantea un compromiso entre la rapidez de convergencia y el costo de control, se la considerara nula.

En definitiva, el objetivo es estabilizar el sistema para que el estado converja rápidamente pero sin gastar demasiado esfuerzo de control.

6.2 Observador

Un observador de estados es un sistema dinámico cuyos estados convergen a los del sistema observado, permitiendo obtener una estimación de los estados no medibles. El observador se debe diseñar de cierta manera para cumplir el objetivo de forzar la convergencia de los estados del observador a los del sistema, tal que $e = x - \hat{x}$ converja a cero.

Se supone la siguiente estructura genérica para el observador:

$$d\hat{x}/dt = A_o \hat{x} + Ly + z$$

Se definen A_o y z de la siguiente manera:

$$A_o = A - LC$$

$$z = Bu$$

Luego el diseño del observador se reduce a encontrar una matriz L que asigne los polos del observador en el lugar deseado.

Consideraciones para la elección de autovalores/polos del observador :

- Deben estar ubicados en el semiplano izquierdo, lo cual asegura la estabilidad del observador
- Deben encontrarse a la izquierda de los autovalores del sistema para asegurar que la dinámica del error de observación sea más rápida que la del sistema, de manera que el observador converja más rápido que este.

6.3 Acción Integral

Los servos consisten en un motor de continua (sistema inherentemente integrador si se considera que entra tensión y que la salida es el ángulo) al cual se le adiciona un lazo de control. Este lazo, aunque facilita el control, elimina la acción integradora del motor. Esto no se considera algo positivo ya que un sistema con un polo en el origen es capaz de eliminar el error al estacionario. Como se desea error nulo al estacionario será necesario agregar acción integral. Esto se lo puede hacer adicionando un integrador en serie con la planta y realizando un par de modificaciones. Esto cambia la estructura de control.

El integrador discretizado tiene la siguiente forma:

$$Integrador = \frac{T_s}{z - 1}$$

La adición del integrador se puede modelar aumentando las ecuaciones de estado con un estado extra que representa el error al estacionario.

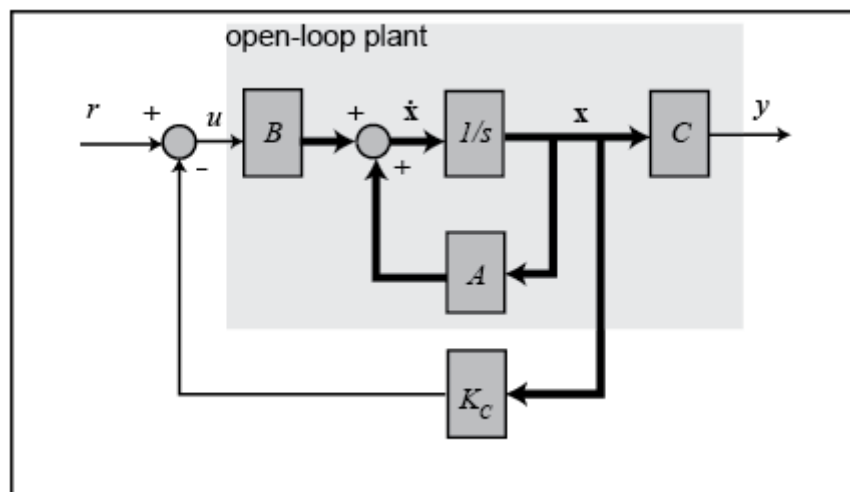


Imagen 33. Diagrama de un sistema a lazo cerrado antes de agregar acción integradora

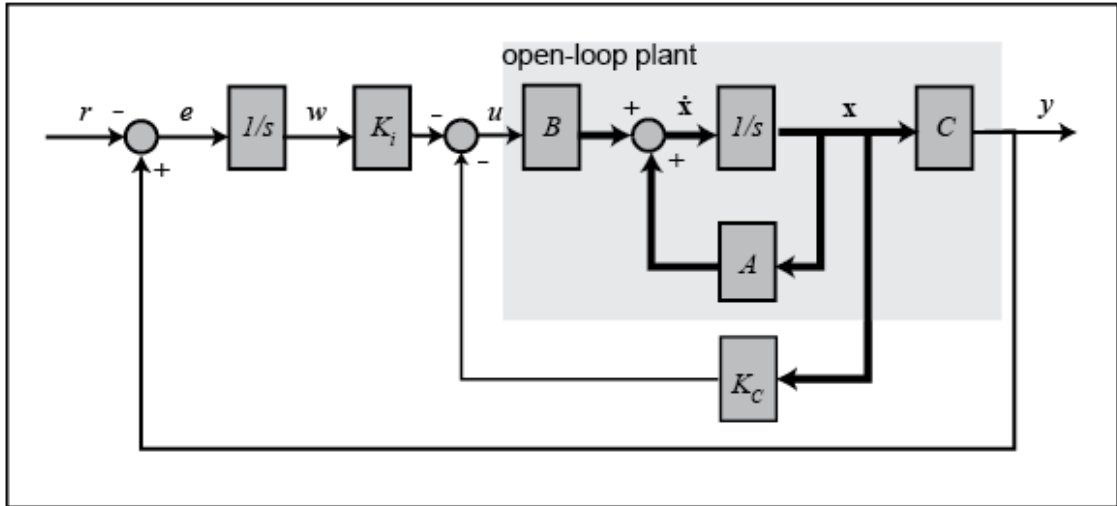


Imagen 34. Diagrama de un sistema a lazo cerrado luego de agregar acción integradora

Se define

$$e(t) = \int_0^t (r(\tau) - y(\tau)) d\tau$$

Tal que

$$\dot{e}(t) = r - y = r - Cx$$

Mi nuevo espacio de estados es:

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r$$

El sistema se lo debe diseñar de cierta forma de que estabilice, luego si $x(\infty)$, $e(\infty)$ y $u(\infty)$ se aproximan a valores constantes, $\dot{e}(\infty) = 0$ y $y(\infty) = r$.

Habiendo dicho esto la ecuación en estado permanente es:

$$\begin{bmatrix} \dot{x}(\infty) \\ \dot{e}(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(\infty) \\ e(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(\infty) + \begin{bmatrix} 0 \\ I \end{bmatrix} r(\infty)$$

Como la referencia se toma constante $r(\infty) = r(t) = r$. Restando las dos ecuaciones anteriores obtengo:

$$\begin{bmatrix} \dot{x}(t) - \dot{x}(\infty) \\ \dot{e}(t) - \dot{e}(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) - x(\infty) \\ e(t) - e(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} [u(t) - u(\infty)]$$

Considerando

$$x(t) - x(\infty) = x_e(t)$$

$$e(t) - e(\infty) = e_e(t)$$

$$u(t) - u(\infty) = u_e(t)$$

Reescribo

$$\begin{bmatrix} \dot{x}_e \\ \dot{e}_e \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_e \\ e_e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e$$

Siendo

$$u_e(t) = -K_c x_e(t) - K_i e_e(t)$$

Mi sistema aumentado será:

$$\dot{x}_a = \begin{bmatrix} \dot{x}_e \\ \dot{e}_e \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_e \\ e_e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e = A_a x_a + B_a u_e$$

$$y_a = \begin{bmatrix} C & 0 \end{bmatrix} x_a = C_a x_a$$

Se diseñará el controlador LQR con A_a y B_a

```
Aa = [A, [0; 0]; [-C 0]];
Ba = [B; 0];
Ca = [C, 0];
Da = D;

Qa=diag([1 1 1]);
Ra=1*45000000;
Na=zeros(3,1);

[Ka,~,pa] = lqr(Aa,Ba,Qa,Ra,Na);
Kc=[Ka(1) Ka(2)];
Kint=[Ka(3)];
```

6.4 Implementación en MatLab y Simulaciones

Se eligieron esos valores para Q y R luego de numerosas iteración. Se ensayaron varios valores produciendo numerosas curvas hasta que se consiguió una que no posee sobre pico y cuyo settling time ronda los 0.35 segundos. Se eligió que el settling time del sistema sea 0.1 seg más lento que el de los servos para no saturarlos ni usarlos al límite considerando que puede haber cierta incertidumbre en la modelización de la planta. A su vez, se encontró que movimientos muy rápidos de la plataforma generaban valores erráticos y muy bruscos en el sensado de la velocidad angular (giróscopo) produciendo errores.

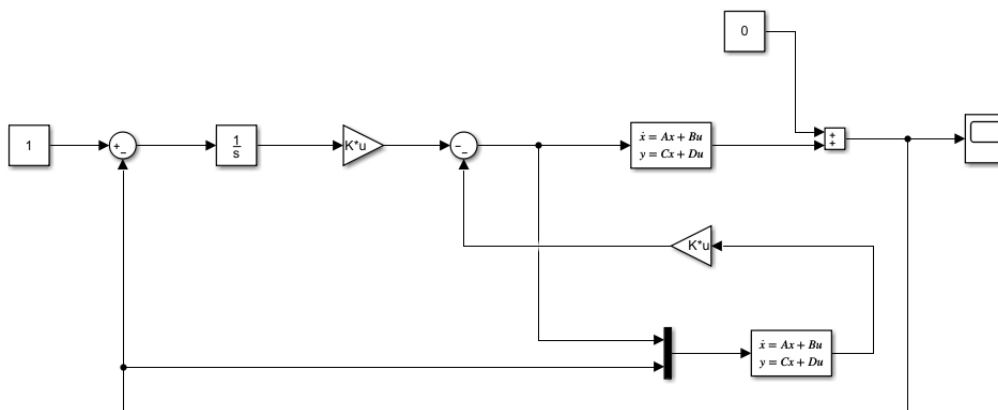


Imagen 35. Construcción del sistema analógico en Simulink

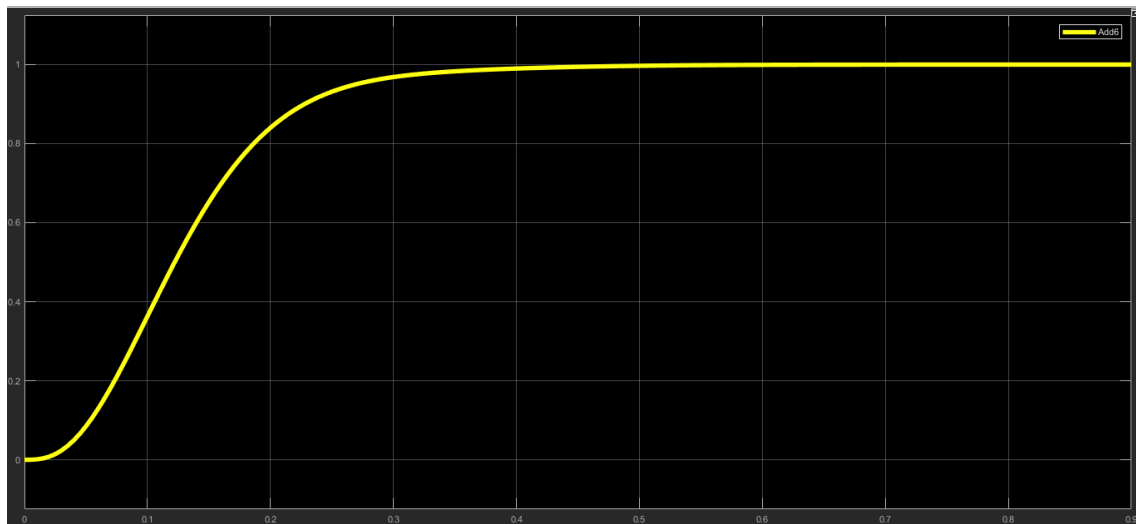


Imagen 36. Respuesta del sistema a un escalón unitario

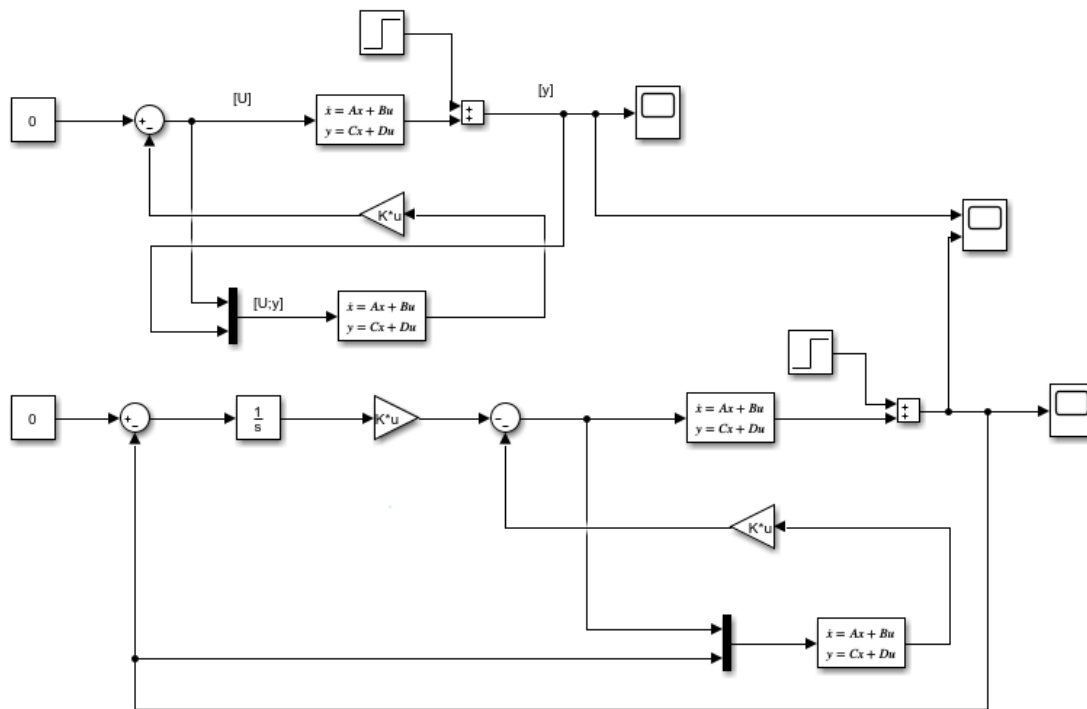


Imagen 37. Diagrama comparativo del sistema con y sin acción integradora

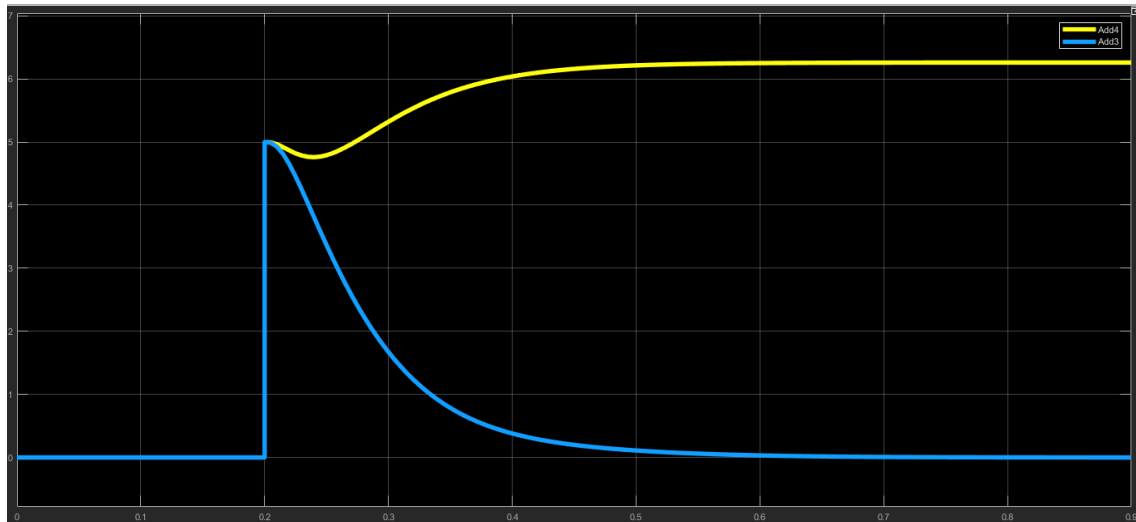
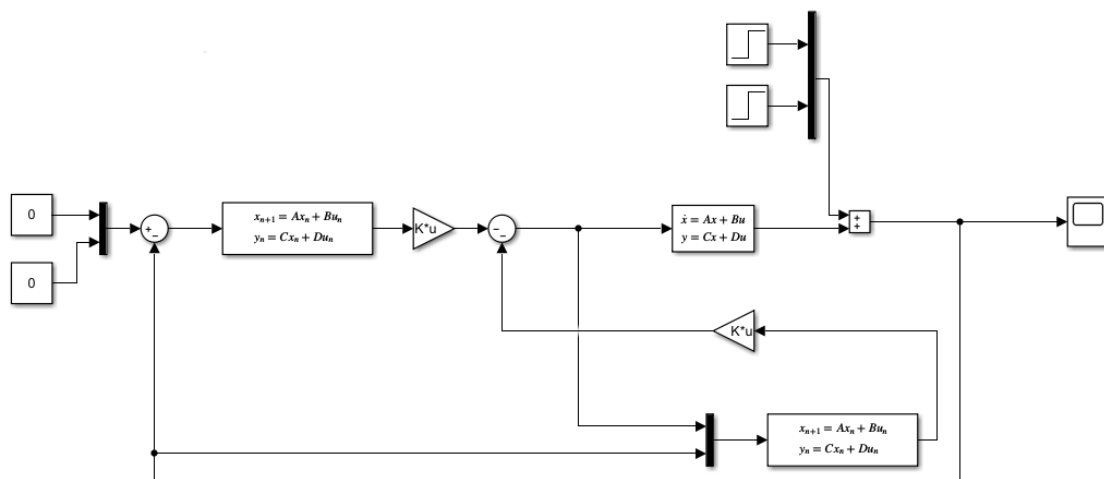


Imagen 38. Diferencia de respuesta ante una perturbación de igual magnitud de los sistema con (azul) y sin acción integradora (amarillo)

Es muy importante señalar que al observador no se lo puede convertir de SISO(un eje) a MIMO(dos ejes) como se hizo con la planta. Se lo debe calcular nuevamente una vez que se modelizo todo el sistema.

6.5 Implementación en dsPIC

Para poder implementar el lazo diseñado se tuvo que convertir el controlador, el integrador y el observador al dominio discreto.



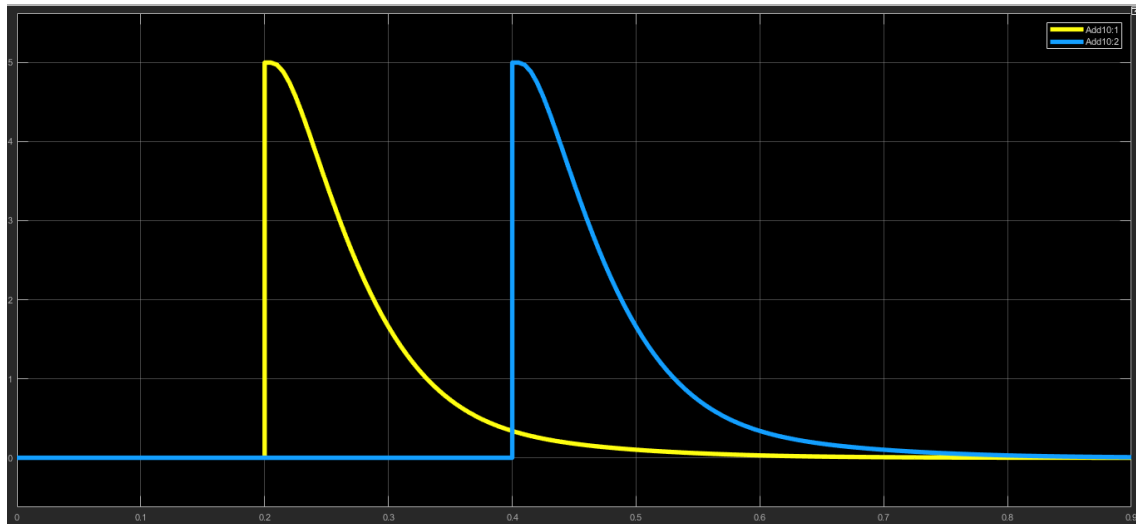


Imagen 39. Respuesta del sistema MIMO discreto a dos perturbación de igual magnitud aplicadas con un diferencia de 0.2 segundos sobre las distintas salidas

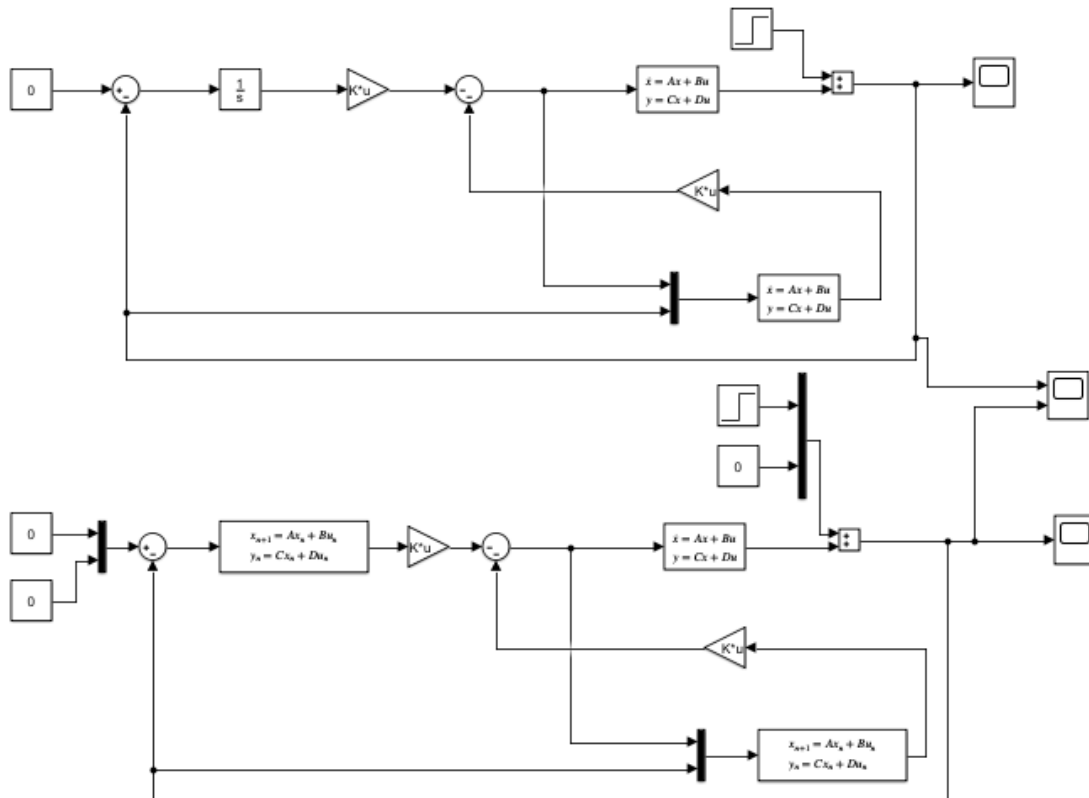


Imagen 40. Diagrama comparativo del sistema SISO analógico (arriba) con el sistema MIMO discreto (abajo)

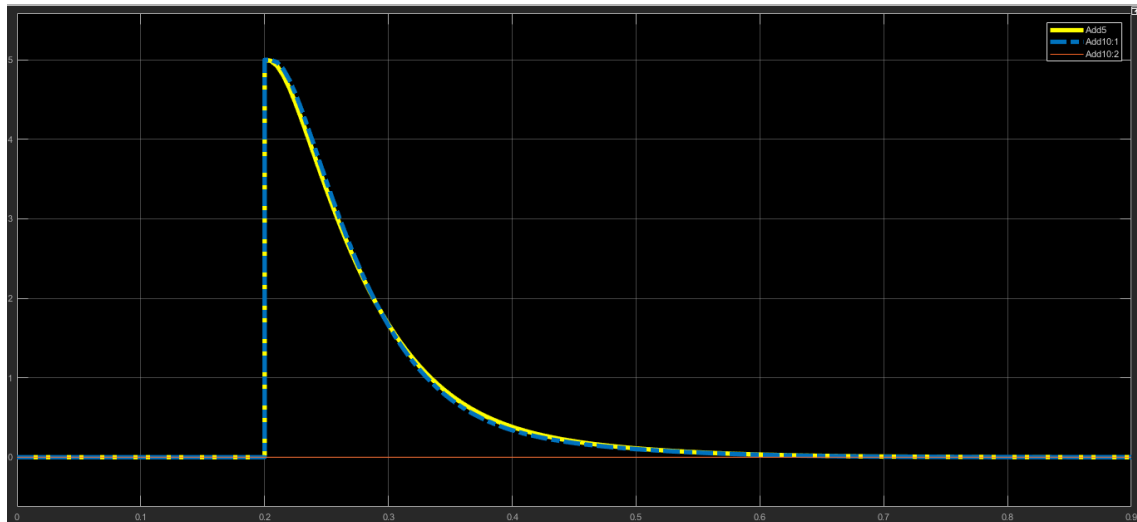


Imagen 41. Comparación de la respuesta del sistema SISO analógico (amarillo) con el sistema MIMO discreto (azul punteada), cuando se aplica la misma perturbación al sistema SISO que a una de las salidas del sistema MIMO.

Para observar cómo se implementó al lazo en el microcontrolador referirse al apéndice.

VII. DETERMINACIÓN DE ORIENTACIÓN

El primer paso para poder controlar la plataforma es determinar su posición partir de los datos del acelerómetro y/o giróscopo. Los datos provistos por estos dos sensores se pueden trabajar y procesar separadamente o se pueden utilizar técnicas de fusión de datos ("sensor fusión").

El problema de utilizar únicamente el acelerómetro para el cálculo de la posición es que es muy susceptible a vibraciones, teniendo en cuenta que esa es parte de su función, lo que afecta el cálculo de la inclinación ya que las mediciones poseen mucho ruido. El problema del giroscopio es que al integrar la velocidad angular, debido a la imprecisiones en la toma de las muestras, a la desviación del valor real, a errores en la calibración y a las del mismo sensor se acumula un error que genera una desviación con el paso del tiempo. Sin embargo, el giróscopo no es tan susceptible a ruido y el acelerómetro no es susceptible a la mencionada desviación o "drift", por lo que en la práctica se utilizan estos dos sensores en conjunto para tener mediciones más confiables. Se utilizarán técnicas de fusión de datos.

La fusión de datos se refiere al procesos en el cual se utilizan señales de dos o más tipos de sensores para actualizar o mantener el estado de un sistema. En el caso de los sistemas de navegación inercial, el estado generalmente consiste en la orientación y velocidad del dispositivo medidas en un marco de referencia global. Un algoritmo de fusión mantendrá este estado utilizando señales provenientes del acelerómetro y del giroscopio. Existen muchas técnicas para realizar la fusión de datos de sensado, como lo es el filtro complementario o el algoritmo de Kalman.

Se hará el desarrollo del cálculo del ángulo de inclinación con respecto a un eje. Este desarrollo se lo puede replicar para el eje restante.

Se comenzó dejando el sensor en reposo y registrando los valores de aceleración y velocidad que enviaba cada 1ms, de esta forma se conoce el valor verdadero del ángulo de inclinación (0°) y se pueden evaluar las distintas técnicas .

Antes de comenzar se presenta la siguiente tabla en la cual se calculó la media y la desviación estándar de los datos (antes de considerar la sensibilidad):

| Datos Calibrados | | |
|------------------|-----------------|--------------------------|
| | Media (cuentas) | Desv. Estandar (cuentas) |
| ACCEL_X | 10,66799047 | 58,11934175 |
| ACCEL_Y | -2,728355838 | 49,98682734 |
| ACCEL_Z | 16387,11676 | 79,894574 |
| GYRO_X | -0,312152502 | 10,89450512 |
| GYRO_Y | -20,75694996 | 12,76937496 |
| GYRO_Z | -6,909451946 | 11,41177806 |

Tabla 5. Media y Desviación Estándar de los datos registrado cuando el sensor se encontraba en reposo. A estos datos no se les aplico ningún tipo de procesamiento

El eje Z medirá la gravedad por lo que tiene sentido que la media ronde las 16387 cuentas. A su vez se puede notar que aunque se calibren los ejes la media no será 0.

7.1 Cálculo del Ángulo de Inclinación – Acelerómetro

Considerando solamente las mediciones del acelerómetro, el ángulo de inclinación se lo puede calcular de la siguiente manera:

$$A_x = 1g \times \sin(\alpha) \quad \therefore \quad \alpha = \arcsin\left(\frac{A_x}{1g}\right)$$

A_x = aceleración medida con respecto al eje X del sensor

$1g$ = gravedad

α = ángulo del eje X del sensor con respecto al vector gravedad

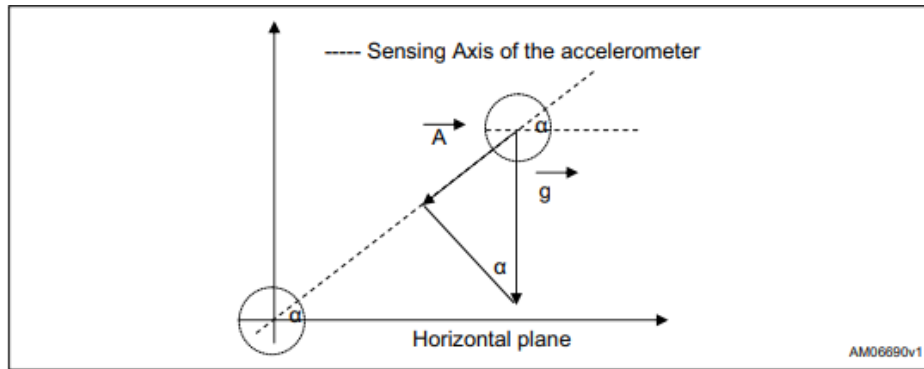


Imagen 42. Angulo de inclinación

Si el acelerómetro tiene dos ejes también se puede calcular α de la siguiente forma:

$$A_y = 1g \times \cos(\alpha) \quad \therefore \quad \alpha = \arccos\left(\frac{A_y}{1g}\right)$$

A_y = aceleración medida con respecto al eje Y del sensor

Si se grafica el ángulo en función de las mediciones en cada eje se obtiene la siguiente curva:

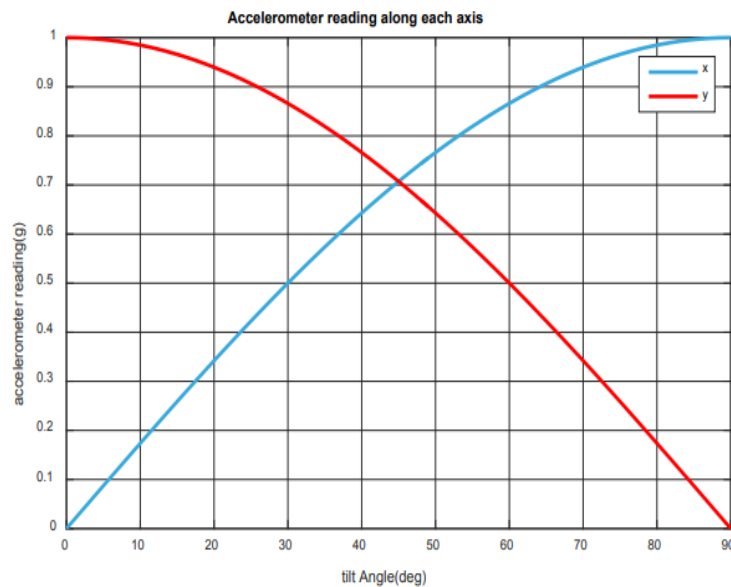


Imagen 43. Valor escalado registrado por el eje de un acelerómetro VS ángulo de inclinación del sensor

Se puede observar que el sensor responde mejor a los cambios en el ángulo de inclinación cuando el eje de sentido es perpendicular a la fuerza de la gravedad. En este caso, la sensibilidad² es de aproximadamente 17,45 mg/° [= sin(1°) - sin(0°)]. El sensor tiene una sensibilidad más baja (responde menos a los cambios del ángulo de inclinación) cuando el eje de detección está cerca de su posición de +1g (o -1 g). En este caso, la sensibilidad es de solo 0,15 mg/° [= sin(90°) - sin(89°)]

Otra forma de calcular el ángulo es la siguiente:

$$\frac{A_X}{A_Y} = \frac{1g \times \sin(\alpha)}{1g \times \cos(\alpha)} \quad \therefore \quad \alpha = \arctan\left(\frac{A_X}{A_Y}\right)$$

Usar arctan permite mantener la sensibilidad constante.

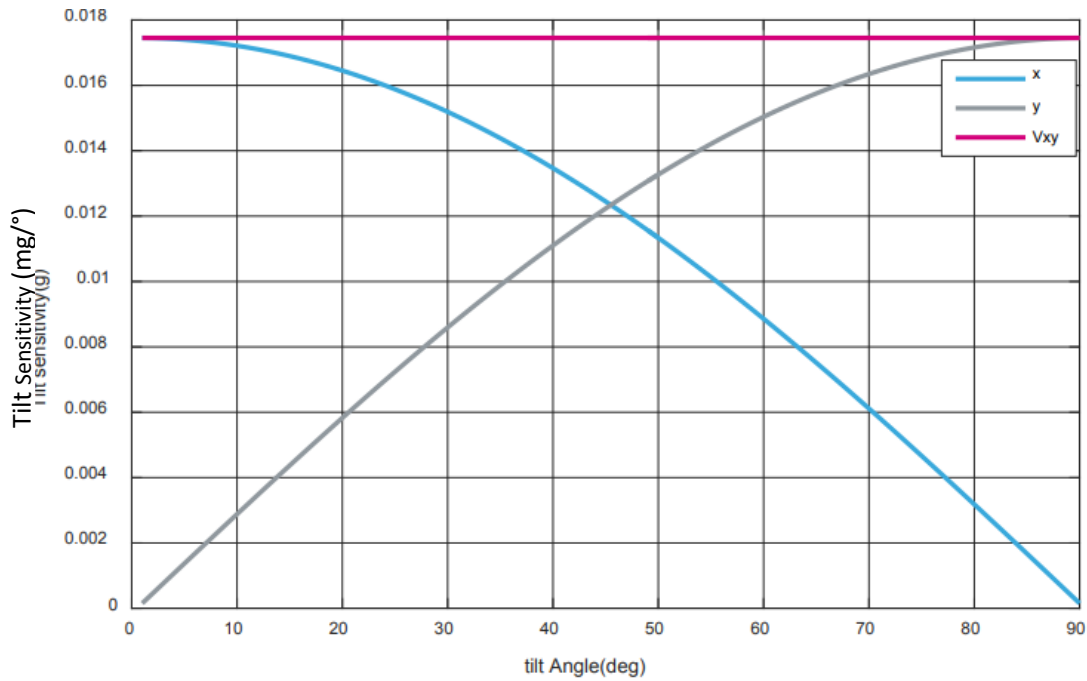


Imagen 44. Sensibilidad VS ángulo de inclinación

Cuando el acelerómetro tiene 3 ejes, la ecuación anterior pasa a ser la siguiente:

$$pitch = \alpha = \arctan\left(\frac{A_X}{\sqrt{A_Y^2 + A_Z^2}}\right)$$

Trasladando esto al cálculo del ángulo β (ángulo del eje Y del sensor respecto al vector gravedad):

$$roll = \beta = \arctan\left(\frac{A_Y}{\sqrt{A_X^2 + A_Z^2}}\right)$$

Como el vector gravedad suele apuntar en el sentido del eje Z, γ , el ángulo de este eje con respecto al vector gravedad se calcula como:

$$yaw = \gamma = \arctan\left(\frac{\sqrt{A_X^2 + A_Y^2}}{A_Z}\right)$$

² La sensibilidad se puede medir calculando la pendiente en cada α o diferenciando las ecuaciones anteriores

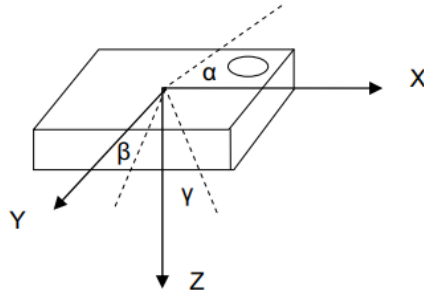


Imagen 45. Ángulos de inclinación de un acelerómetro de tres ejes

En este proyecto se calcularán α y β

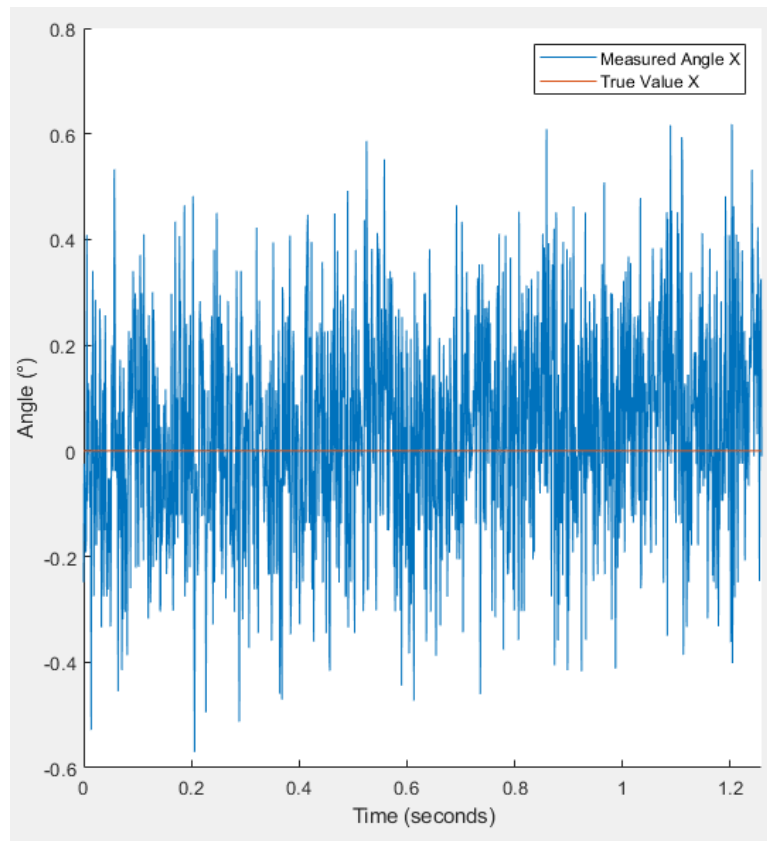


Imagen 46. Medición con ruido del ángulo de inclinación del eje X del acelerómetro

Se puede apreciar el ruido que posee la medición cuando se utiliza únicamente el acelerómetro para el cálculo de la inclinación.

7.2 Cálculo del Ángulo de Inclinación – Giróscopo

Considerando solamente los valores provistos por el giróscopo, se hace lo siguiente: al tener la velocidad angular lo que se debe hacer para obtener el ángulo simplemente es integrar. Se considera que la velocidad angular es constante en el intervalo $\Delta Ts = 1ms$. Se utiliza la siguiente formula

$$\theta_k = \theta_{k-1} + \omega_k \Delta Ts$$

Para utilizar esta fórmula se debe conocer el ángulo inicial.

Como ya se mencionó anteriormente la principal fuente de error en este método es la deriva (o “drift”). Hay dos componentes principales de la señal de entrada que contribuyen a la deriva, estas son la componente de bias y el ruido.

Al integrar una constante (bias) a lo largo del tiempo se obtiene una salida que crece linealmente. Por otro lado, cuando se integra el ruido (incluso el ruido con media cero) se obtiene lo que se llama “random walk”. Si bien la media de la salida aún puede ser cero, la magnitud de la salida más probable crece proporcionalmente a la raíz cuadrada del tiempo transcurrido.

El ruido puede provenir de varias fuentes:

- ruido térmico en el sensor y la electrónica es la fuente más obvia
- señales con alias (esto será una fuente de error si los datos del giroscopio cambian más rápidamente que la frecuencia de muestreo)
- ruido de cuantificación asociado con el ADC
- el error asociado a la precisión numérica.
- asumir que la velocidad angular es constante en el intervalo ΔT s

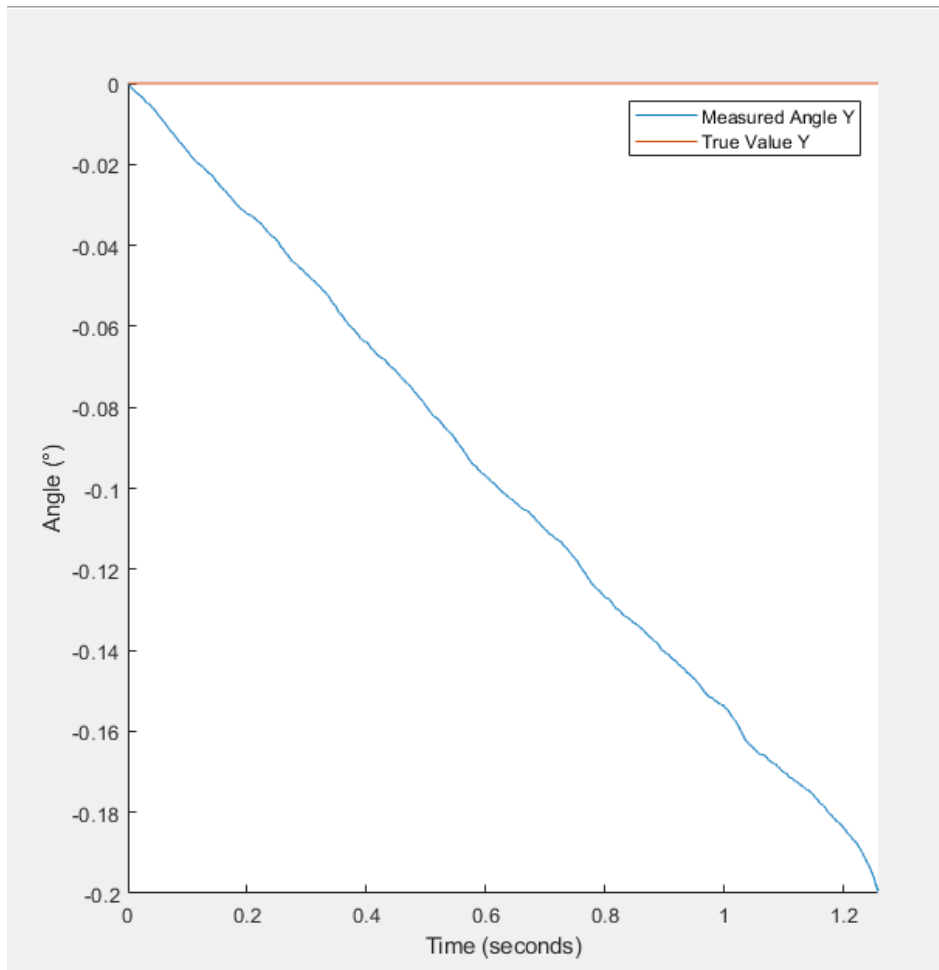


Imagen 47. Drifteo de la señal cuando se calcula el ángulo de inclinación utilizando únicamente el giróscopo

El error que domina sobre los otros es el error de bias.

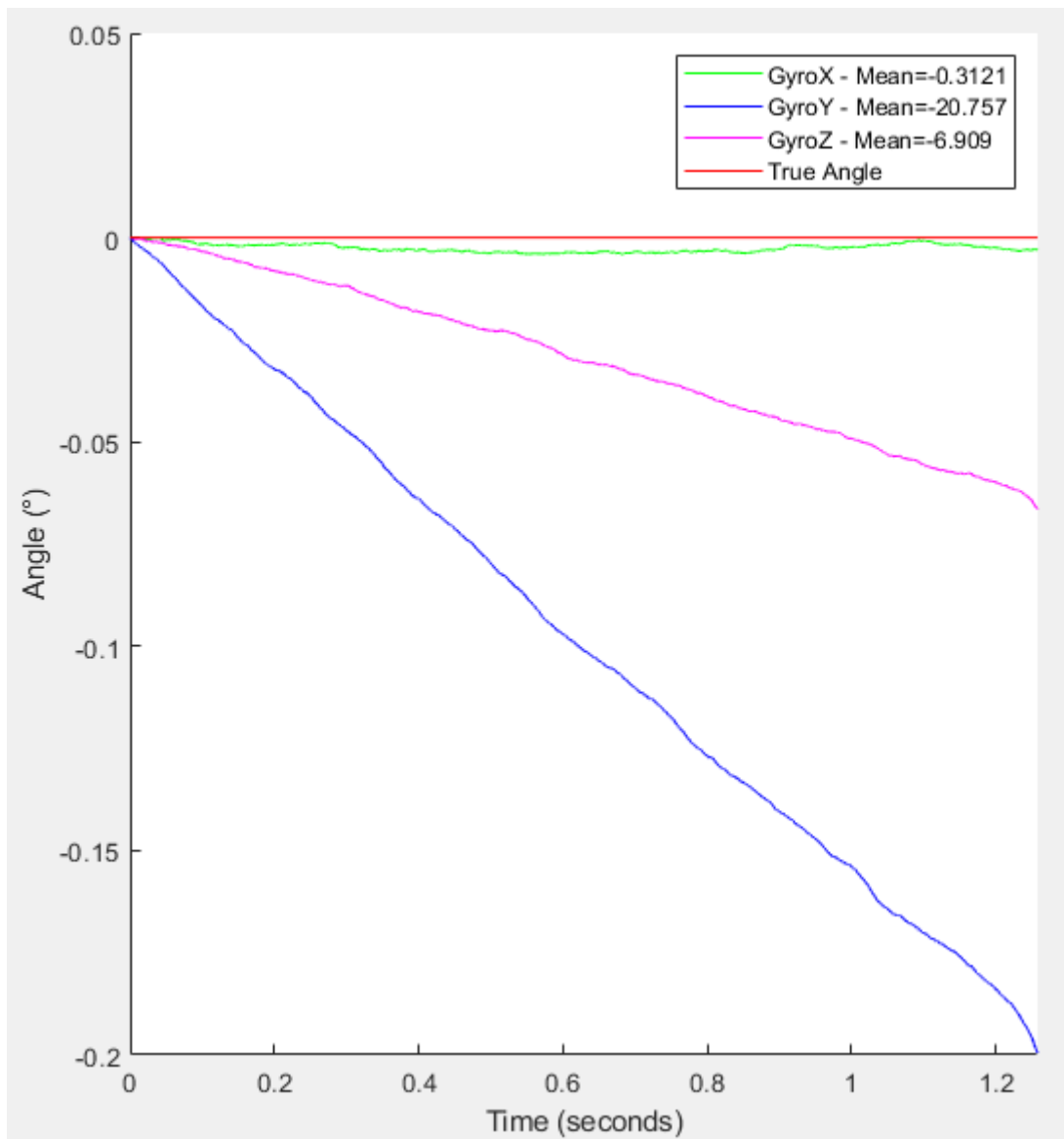


Imagen 48. Efecto del error de bias sobre el drift

7.3 Cálculo del Ángulo de Inclinación – Filtro Complementario

El acelerómetro es muy rápido y sensible, por lo que sus datos no son fiables a corto plazo, por lo que se ha de utilizar un filtro paso bajo que dejará pasar sólo a valores que cambien a largo plazo que son estables en el tiempo, eliminando el ruido y amortiguando las fluctuaciones esporádicas. La desventaja de esto es que produce un retardo en la percepción del ángulo, por lo que puede provocar problemas en la estabilización.

Por otro lado, los datos del giroscopio son fiables sólo a corto plazo, ya que su medida tiene tendencia a la deriva como se explicó anteriormente. Al contrario que el filtro de paso bajo usado en el acelerómetro, para el giroscopio se utiliza un filtro de paso alto que sólo permite el paso a las señales de corta duración (o a corto plazo).

Los datos del giróscopo sirven para eliminar el retardo producido por el filtro de paso bajo aplicado al acelerómetro. La fórmula resultante de combinar/complementar los dos filtros es:

$$\theta_k = \rho (\theta_{k-1} + \omega_{k_{GYRO}} \Delta Ts) + (1 - \rho) \theta_{accel}$$

Dónde ρ es la ganancia del filtro de paso alto y $(1 - \rho)$ es la ganancia del filtro de paso bajo.

El filtro se llama complementario porque combina sus dos partes de manera que siempre suman uno, consiguiendo una estimación precisa y lineal en la salida.

En un principio se pensó en usar un valor de ρ de 0.94, ya que ofrecía un buen compromiso entre el ruido y el delay. Posteriormente, al implementarlo en el dsPIC se encontró que la gimbal es muy sensible al ruido por lo que se tuvieron que utilizar valores más altos. En la mayoría de la literatura encontrada se utiliza un valor de 0.98

Se usa el ángulo calculado únicamente con el acelerómetro como referencia.

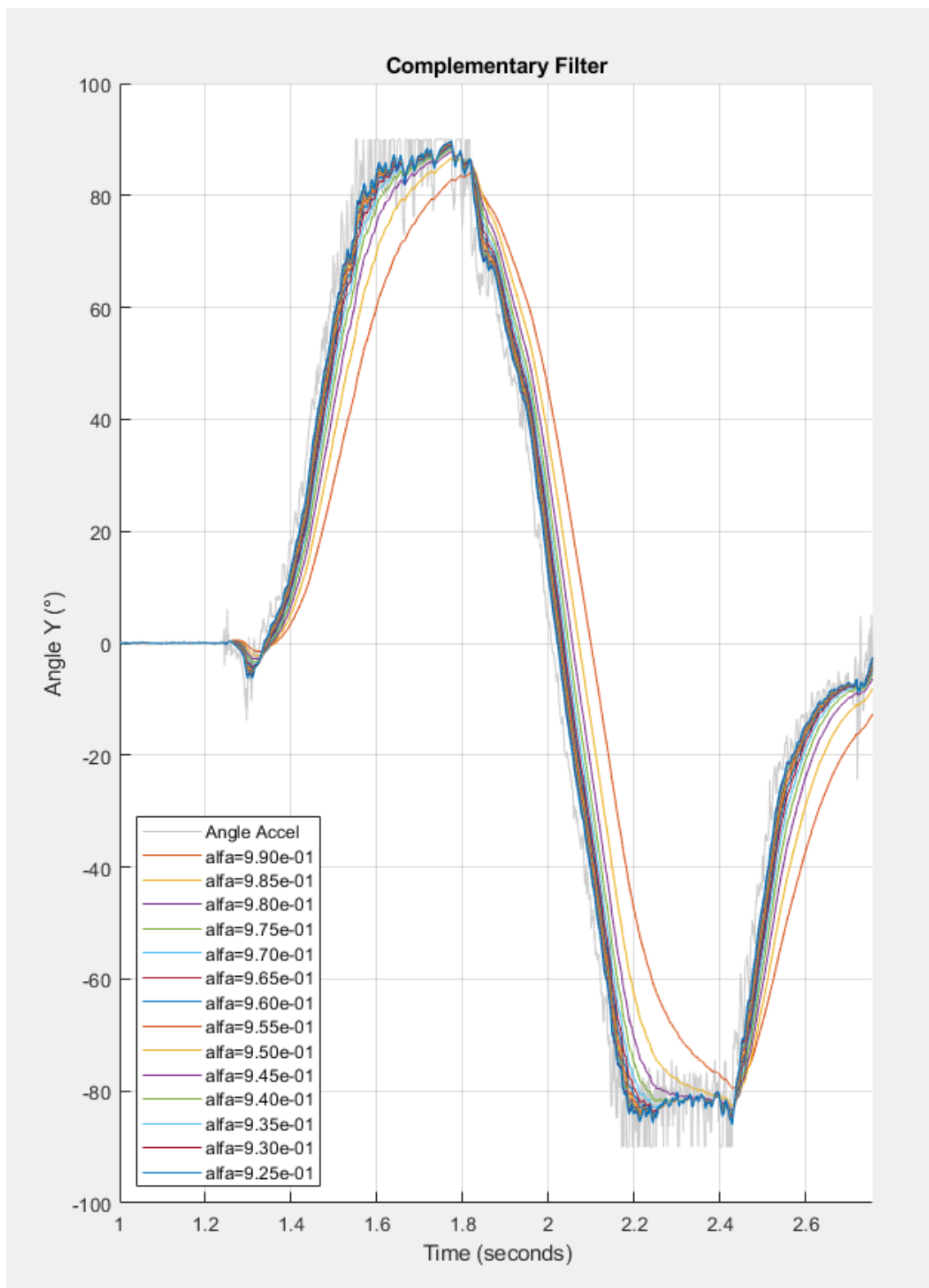


Imagen 49. Aplicación del filtro complementario variando el ρ

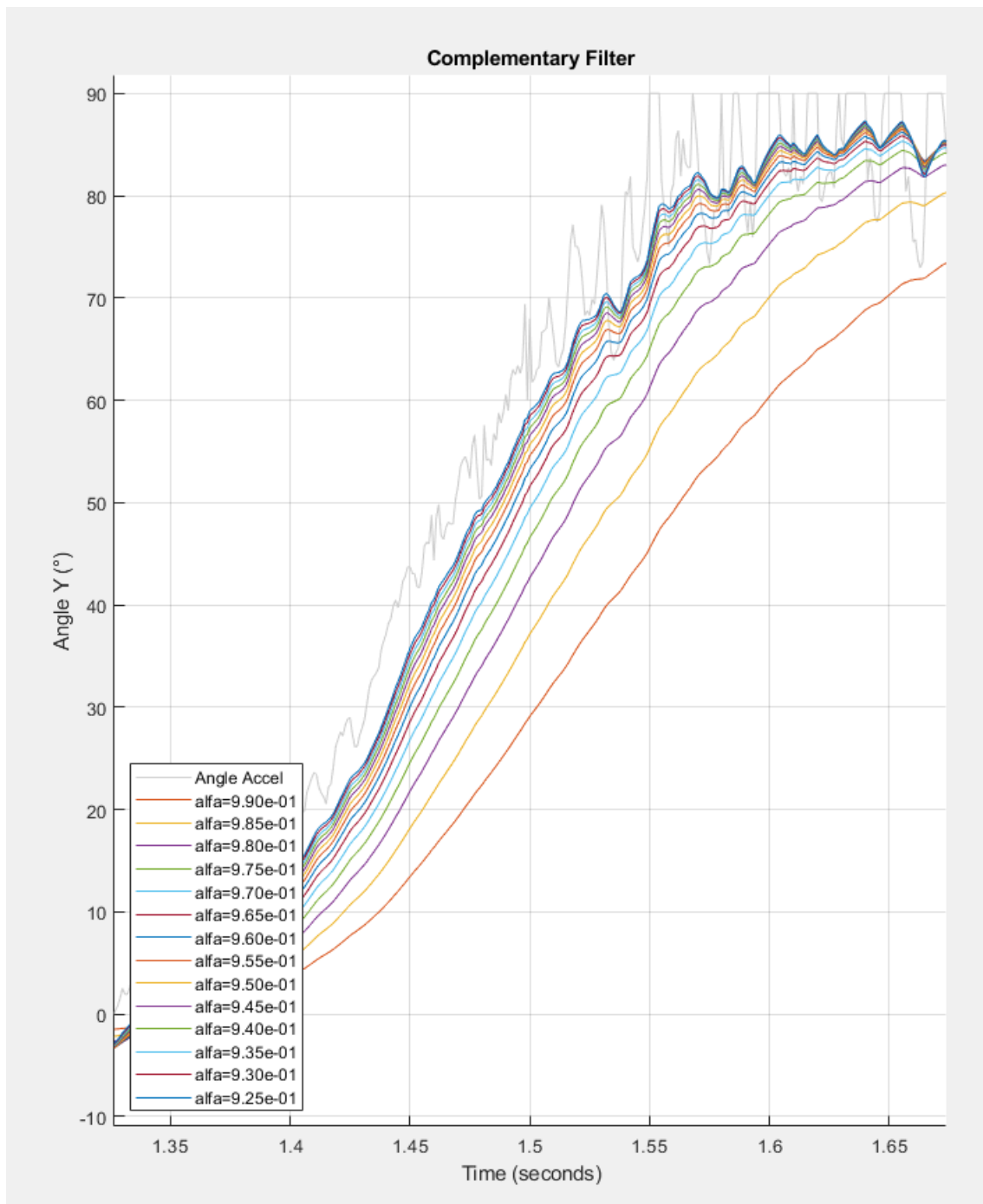


Imagen 50. Zoom en aplicación del filtro complementario variando el ρ

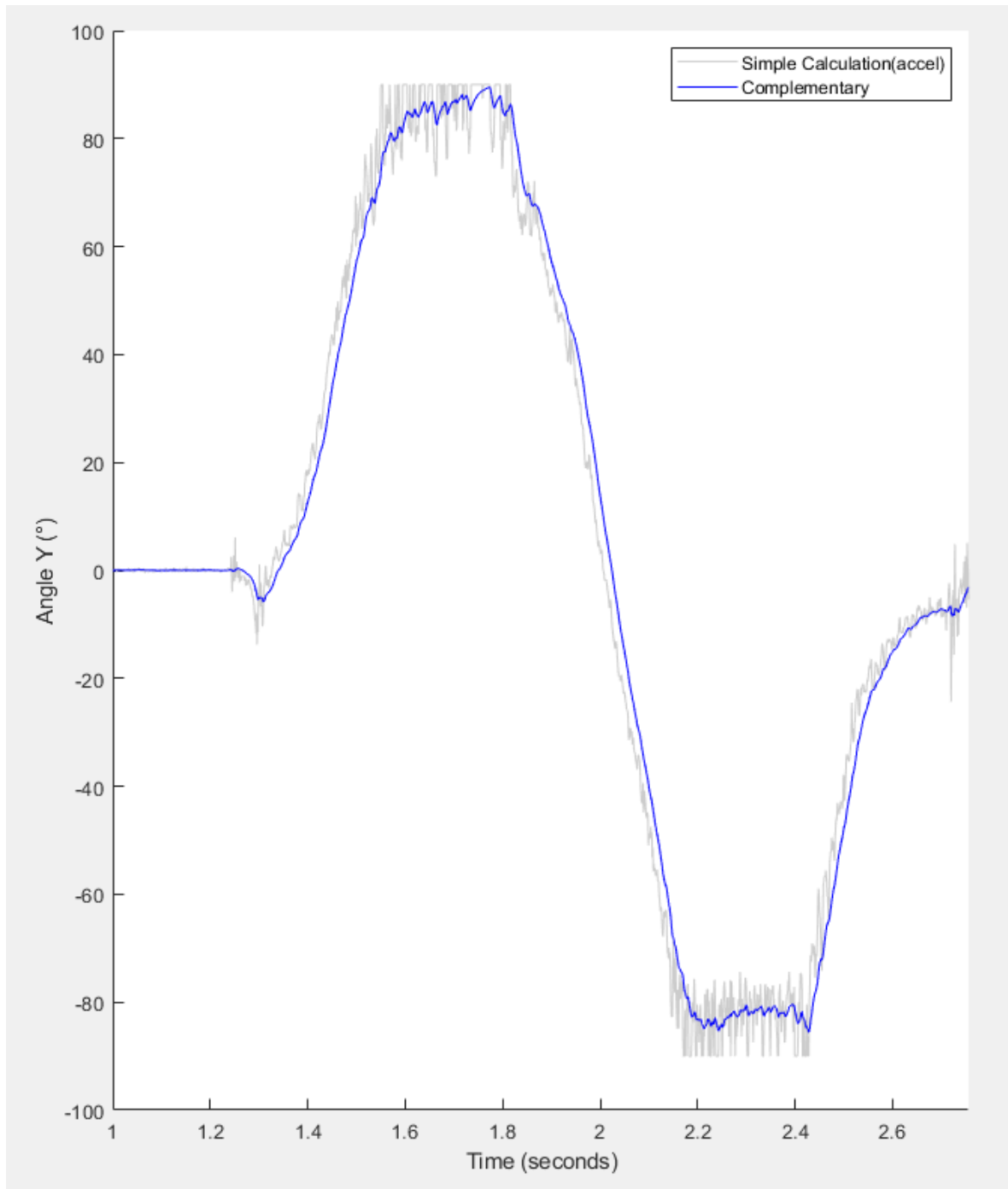


Imagen 51. Filtro Complementario de $\rho = 0.96$

7.4 Cálculo del Ángulo de Inclinación – Kalman

Kalman es un algoritmo para estimar de manera óptima el estado de un sistema a partir de mediciones contaminadas por ruido. Produce estimaciones precisas de las variables ocultas basadas en mediciones imprecisas e inciertas. Además, el filtro de Kalman proporciona una predicción del estado futuro del sistema en función de estimaciones anteriores.

El filtro de Kalman es uno de los algoritmos de estimación más importantes y ampliamente usados. La ventaja de Kalman es que utiliza todas las medidas disponibles, independientemente de su

precisión o el nivel de ruido que poseen, para estimar el valor actual del estado del sistema mediante la ponderación adecuada de estas medidas.

El filtro de Kalman requiere de la siguiente información:

- 1) Información sobre el modelo del sistema y sus medidas.
- 2) Información estadística del ruido del sistemas, de los errores de medición y la incertidumbre en el modelo del sistema.
- 3) Conocimiento sobre las condiciones iniciales de los estados del sistema.

El filtrado de Kalman se basa en los siguientes supuestos:

- 1) El sistema (tanto el proceso como las mediciones) se puede describir mediante modelos lineales.
- 2) El ruido del sistema y el ruido de las mediciones son procesos de ruido blanco gaussiano de media cero no correlacionados con función de covarianza conocida.
- 3) El estado inicial del sistema $X_{k=0}$ es un vector aleatorio no correlacionado con los ruidos del proceso y de la medición.
- 4) Se conoce el valor medio del estado inicial $X_{k=0}$ y su matriz de covarianza $P_{k=0}$

El filtro de Kalman es un proceso de dos pasos: "predicción" y "corrección". El filtro puede comenzar en cualquiera de ambos. En el primer paso mencionado, se predice el estado con el modelo dinámico. En el segundo paso, se corrige con el modelo de observación, de manera que se minimiza la covarianza del error del estimador.

El filtro de Kalman aborda el problema general de tratar de estimar el estado $x \in \mathbb{R}^n$ de un proceso controlado en tiempo discreto que se rige por la siguiente ecuación diferencial estocástica lineal:

$$x_k = A x_{k-1} + B u_k + w_{k-1}$$

Con mediciones $z \in \mathbb{R}^m$

$$y_k = C x_k + v_k$$

x es el vector de estados del sistema, A es la matriz de transición de estados, B es la ganancia de la matriz de señales de control y u es el vector de entrada; y es el vector de medición (también se lo suele llamar z), C relaciona el estado actual con la medición (a esta matriz también se la suele llamar H), w es el ruido del proceso y v es el ruido de medición.

Se asume que w y v son variables aleatorias independientes (entre ellas), de media cero con distribución normal

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

Q : process noise covariance

R : measurement noise covariance

Las ecuaciones del algoritmo de Kalman son las siguientes:

Predicción

$$\hat{x}_k^- = A \hat{x}_{k-1} + B u_k$$

$$P_k^- = A P_{k-1} A^T + Q$$

Actualización (o Corrección)

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C \hat{x}_k^-)$$

$$P_k = (I - K_k C) P_k^-$$

El supra índice negativo indica que lo que se está calculando es una estimación de dicha variable a *priori* de tomar en consideración la medición. Las matrices P_k y K_k se explican a continuación.

Se pueden definir los errores de estimación a *priori* y a *posteriori* (de la medición) de la siguiente manera:

$$e_k^- = x_k - \hat{x}_k^-$$

$$e_k = x_k - \hat{x}_k$$

La covarianza del error de estimación a *priori* será:

$$P_k^- = E(e_k^- e_k^{-T})$$

La covarianza del error de estimación a *posteriori* será:

$$P_k = E(e_k e_k^T)$$

La diferencia $(y_k - C \hat{x}_k^-)$ se conoce como “innovación” de la medición (o “residuo”). El residuo refleja la discrepancia entre la predicción y la medición del estado medido.

Uno de los elementos más importantes del algoritmo es K_k . Se la conoce como la ganancia de Kalman. Es la matriz de ponderación óptima que permite minimizar la covarianza del error de estimación a *posteriori*.

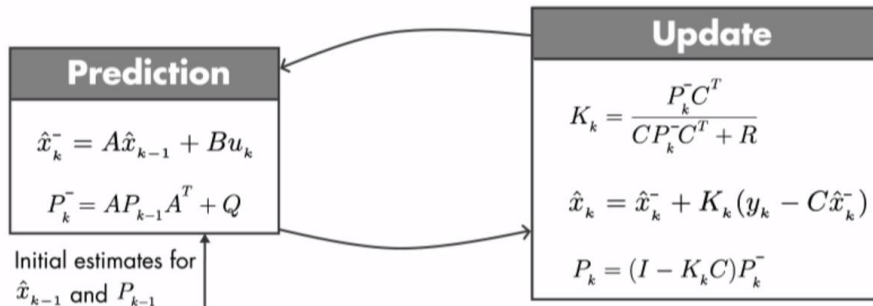


Imagen 52. Algoritmo de Kalman

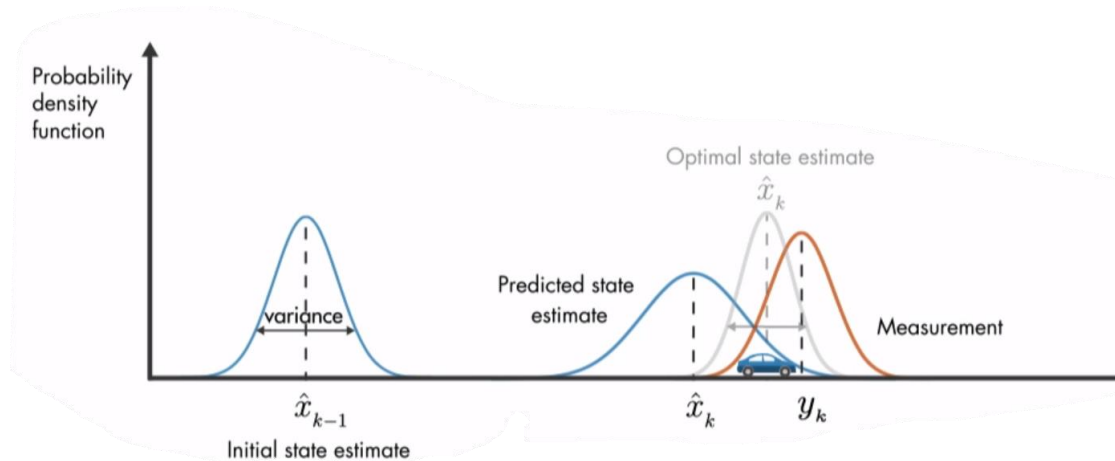


Imagen 53. Curvas de probabilidad de Kalman

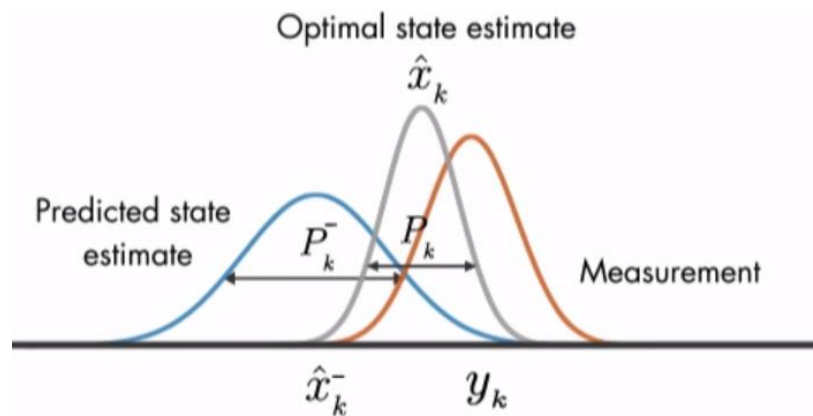


Imagen 54. Covarianza de las curvas de probabilidad

Calculation of \hat{x}_k

\hat{x}_k^- y_k

$$\lim_{R \rightarrow 0} K_k = \lim_{R \rightarrow 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{R \rightarrow 0} \frac{P_k^- C^T}{C P_k^- C^T + 0} = C^{-1} \quad C^{-1} = 1$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C \hat{x}_k^-) = \hat{x}_k^- + C^{-1} (y_k + C \hat{x}_k^-)$$

$$= \cancel{\hat{x}_k^-} + C^{-1} y_k + \cancel{C^{-1} C} \hat{x}_k^-$$

$$\hat{x}_k = y_k$$

Calculation of \hat{x}_k

\hat{x}_k^- y_k

$$\lim_{P_k^- \rightarrow 0} K_k = \lim_{P_k^- \rightarrow 0} \frac{P_k^- C^T}{C P_k^- C^T + R} = \lim_{P_k^- \rightarrow 0} \frac{0}{0 + R} = 0$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C \hat{x}_k^-) = \hat{x}_k^- + 0 (y_k + C \hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^-$$

Imagen 55. Efecto de R o P_k^- nulos en el calculo de \hat{x}_k

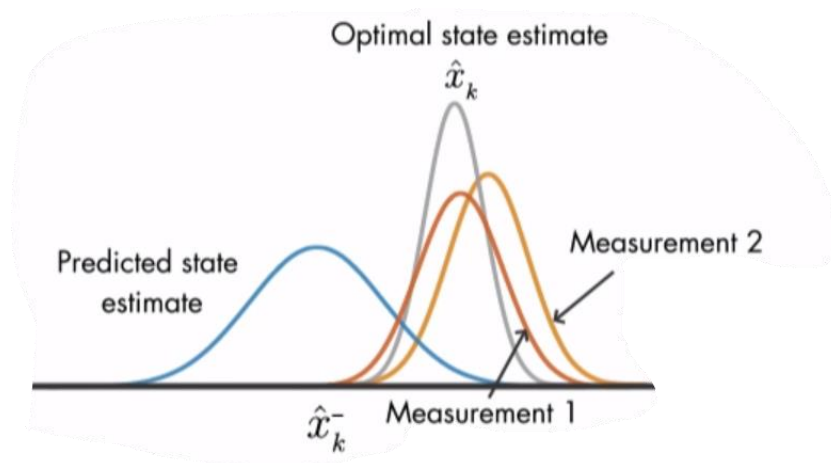


Imagen 56. Efecto de adicionar un nuevo sensor

7.4.1 “Filter Tuning”

La estimación de los parámetros del sistema: el estado inicial $x_{k=0}$ y su covarianza $P_{k=0}$, junto con Q y R se denomina ajuste/sintonización del filtro de Kalman.

En la práctica, se comienza con una suposición inicial razonable para R y Q , y luego se las va ajustando experimentalmente. En la mayoría de los casos es suficiente utilizar matrices diagonales. Se debe tener en cuenta que en aplicaciones reales no conocemos las estadísticas reales del ruido de la variable oculta. Físicamente es posible conocer R , pero no Q , ya que este ruido corresponde a un proceso ficticio (no hay ningún dispositivo físico que lo produzca). Solo refleja cuánto uno confía en su modelo matemático (cuanto más grandes son los valores de la matriz Q , menos confianza se tiene en el modelo y en el ruido asociado).

A su vez, el ruido a menudo no tiene una distribución gaussiana. La práctica común es establecer de manera conservadora R y Q , asignándole valores ligeramente más grandes que los valores esperados. Esto contribuye a aumentar la robustez del sistema.

Al igual que en el caso de LQR, lo importante es la proporción de R y Q (en lugar de sus valores individuales), ya que esta relación es lo que determina el rendimiento del filtro de Kalman.

Elegir el valor inicial de $x_{k=0}$ y su covarianza $P_{k=0}$ es mas sencillo. Se debe tener en cuenta la siguiente consideración: para que el filtro funcione correctamente, los valores de la matriz de covarianza deben definirse de manera que la diferencia entre el estado inicial y la estimación del estado inicial se encuentre dentro del rango permitido por dicha matriz. Si la estimación del estado inicial es bastante precisa, la matriz de covarianza solo necesita permitir un pequeño error. Es por esta razón que es importante comenzar con el sensor en reposo, así conozco el estado inicial y puedo fijar el estado inicial y su matriz de covarianza de forma correcta, de lo contrario el filtro puede no converger.

7.4.2 Implementación

Partiendo de la ecuación mencionada en la sección 7.2 que establecía que:

$$\theta_k = \theta_{k-1} + \omega_k \Delta Ts$$

Asumiremos que la velocidad angular del gir6scopo tiene una desviaci6n constante, la llamaremos b . Podemos asumir que es constante ya que al analizar los gr6ficos de la secci6n 7.2, vemos como el 6ngulo diverge pr6cticamente de forma lineal cuando solo se utilizan los valores del gir6scopo para calcularlo.

$$\frac{db}{dt} = cte \Rightarrow b_{k+1} = b_k$$

Luego se puede plantear lo siguiente:

$$\theta_k = \theta_{k-1} + (\omega_k - b_k) \Delta Ts$$

Representando las dos ecuaciones anteriores en un espacio de estados sin ruido obtenemos

$$\begin{bmatrix} \theta_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta Ts \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_k \\ b_k \end{bmatrix} + \begin{bmatrix} \Delta Ts \\ 0 \end{bmatrix} \omega_k$$

Podemos decir entonces que:

$$x_k = \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -\Delta Ts \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \Delta Ts \\ 0 \end{bmatrix}$$

$$u_k = \omega_k$$

Los datos provistos por el aceler6metro ser6n tenidos en cuenta en la etapa de actualizaci6n

$$y_k = C x_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_{ACCEL_k} \\ b_k \end{bmatrix} = \theta_{ACCEL_k}$$

Una vez modelado el filtro se procedi6 a su sintonizaci6n. Se eligieron los siguientes par6metros suponiendo que el sistema inicia en reposo:

$$x_{k=0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$P_{k=0} = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$$

$$R = [10]$$

Los valores iniciales de Q y R se eligieron iterando y calculando el error cuadr6tico medio respecto a una referencia (la referencia ser6 0, el sensor est6 en reposo):

```
count=1;
for q1=0.001:0.001:1
    for q2=0.001:0.001:1
        for r=1:1:200
            rmse=kalmanPrecision(accel,gyro,q1,q2,r,reference);
```

```

        result(count,1:4)=[rmse q1 q2 r];
        count=count+1;
    end
end
end
rmse=result(:,1);
q1=result(:,2);
q2=result(:,3);
r=result(:,4);

```

Se eligió aquellos valores que permitían el menor RMSE. Luego evaluando estos valores en otros datos (en los cuales el sensor no estaba en reposo) se los fue adaptando para lograr el desempeño requerido.

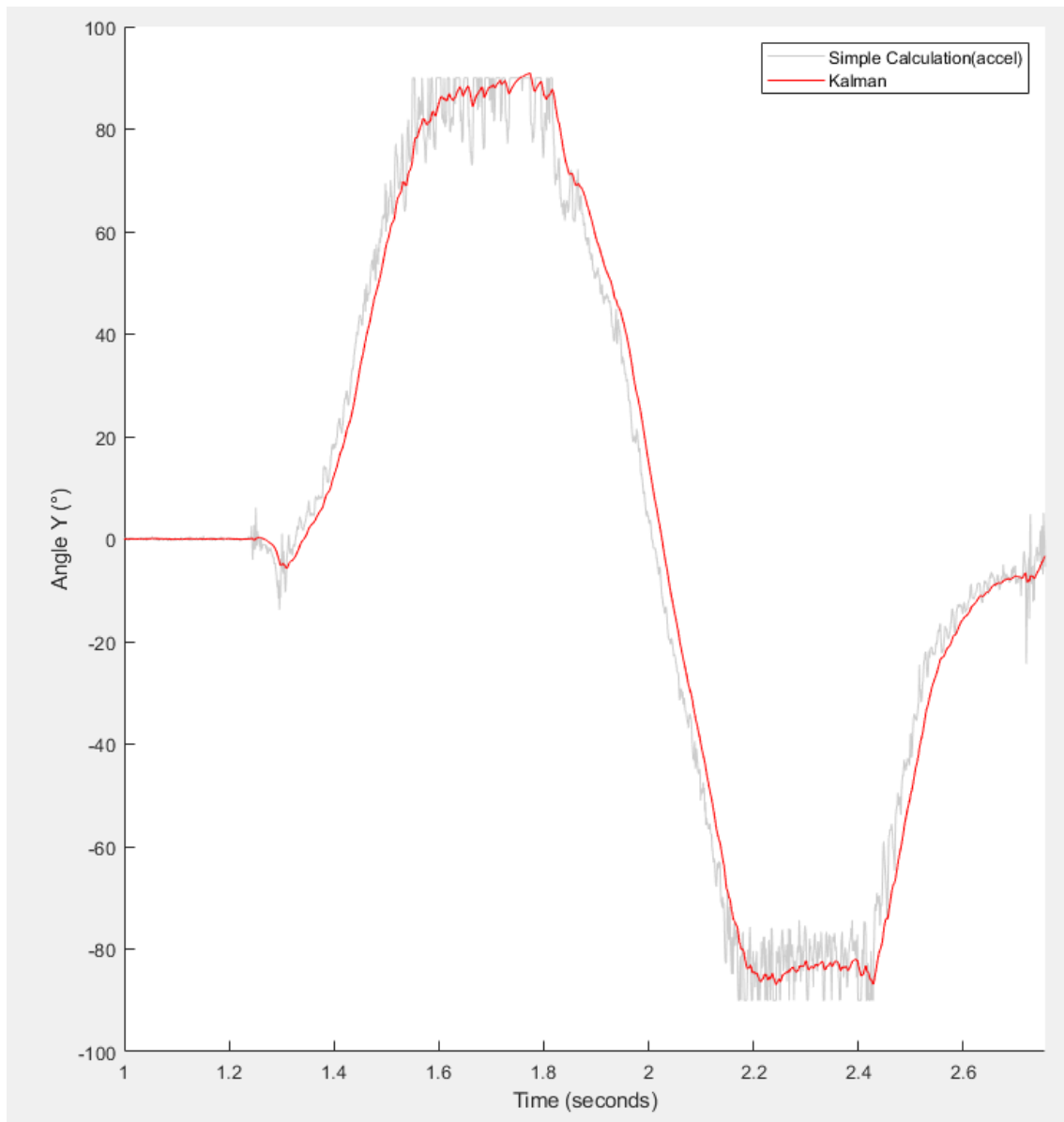


Imagen 57. Filtro de Kalman

El filtro de Kalman se puede aplicar de manera separada a cada eje o se puede plantear un espacio de estados generalizado que combina ambos ejes, de igual manera que se hizo al modelar el gimbal. En este caso, se mantendrán los ejes separados para reducir los tiempos de cálculo.

Si x_k posee solo dos dimensiones, la operación $C P_k^{-1} C^T + R$ dará como resultado un valor numérico de una dimensión; de esta manera no será necesario calcular su inversa para obtener el nuevo valor de K_k en cada ciclo de Kalman. Se debe efectuar simplemente una división.

VIII. EVALUACIÓN

8.1 Plataforma de pruebas

Para testear el sistema diseñado se construyó una plataforma a escala. Para la estructura del gimbal se utilizó un modelo hallado en Thingiverse³. Este se imprimió por medio de técnicas 3D

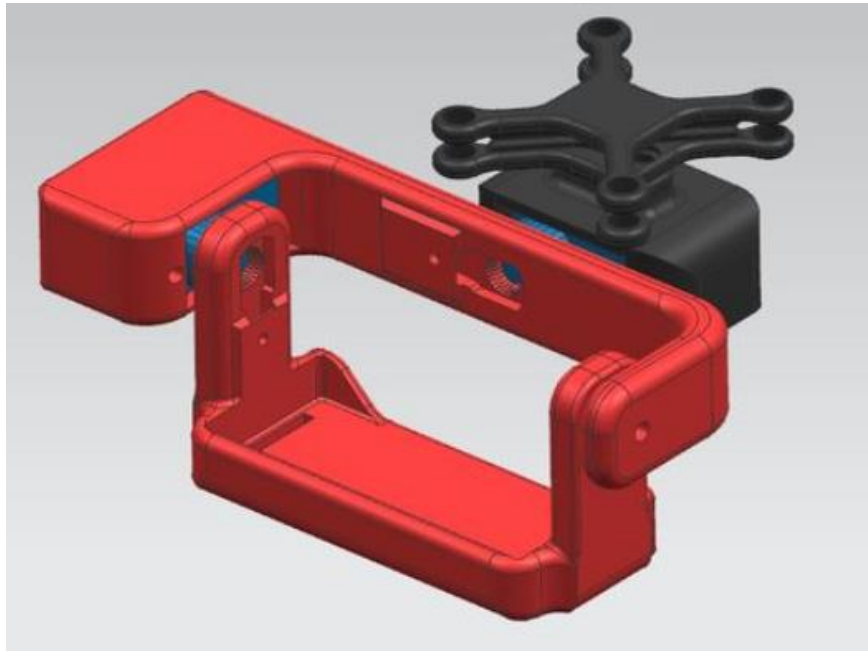


Imagen 58. Estructura 3D

8.2 Distribución del código en ambos núcleos

En un principio se quiso dedicar el núcleo Primario a todo lo que hacía a las funciones generales y a la comunicación con el sensor y el usuario; mientras que el núcleo Secundario iba a realizar las tareas de procesamiento y filtrado de datos e iba a ser el encargado de cerrar el lazo implementando el observador y el compensador. Es decir que se quería dedicar el Secundario a todas las tareas que involucraran un cálculo numérico. Esto se planeó así ya que el Secundario funciona a mayor velocidad que el Primario.

El Primario iba a recibir los datos enviados por el MPU6050 a través de I2C e iba a enviarlos a través del protocolo MSI al Secundario. El Secundario, habiendo recibido los datos “crudos”, iba a estar encargado de procesarlos, filtrarlos, estimar las variables ocultas, realimentar el lazo y calcular el nuevo ciclo de trabajo de cada una de las señales PWM.

Se encontró que esta distribución no es razonable si se desea aplicar el filtro de Kalman. Se debe tener en cuenta lo siguiente:

- Se reciben datos nuevos del sensor cada 1ms
- El duty de las señales PWM solo se puede actualizar cada 5ms (como se indicó anteriormente el periodo del PWM al cual responden los servos es 5ms)

³ Sitio web dedicado a compartir archivos de diseño digital. <https://www.thingiverse.com/>

- La porción de código que involucra la ejecución del observador y el compensador es capaz de demorar hasta $652,8 \mu s$ (se midió la demora utilizando el módulo SCCP3 como timer)
- La aplicación del filtro de Kalman demora aproximadamente lo mismo que demora el cierre del lazo

Consideraciones:

- Se desea recibir datos del sensor a la máxima velocidad posible ya que esto me permitirá realizar un mejor traqueo de los movimientos, reducir el ruido y el delay apreciable por el usuario al implementar el filtro.
- Se desea utilizar “double” como tipo de dato siempre que sea posible para tener más precisión y que el comportamiento del sistema sea lo más parecido posible a las simulaciones (MatLab trabaja con alta precisión)

Al tener en cuenta esto se encontró que la mejor distribución es la siguiente:

| <u>Master/Primario</u> | <u>Slave/Secundario</u> |
|-------------------------------|----------------------------------|
| - Funciones Generales | - Implementación del Observador |
| - Comunicación con el usuario | - Implementación del compensador |
| - Sensado | LQR |
| - Filtrado y fusión de datos | - Control de los servos |

De esta manera el Primario recibirá interrupción cada 1ms del módulo de sensado, procesará los datos (Kalman), y cada cinco interrupciones recibidas comunicará los ángulo de inclinación respecto al eje “X” e “Y” al Secundario a través de MSI, generando una interrupción en el Secundario cada 5ms.

El secundario recibirá los nuevos ángulos de inclinación, implementará el observador y el compensador cerrando el lazo y actualizara del ciclo de operación de las señales PWM que controlan a los servos.

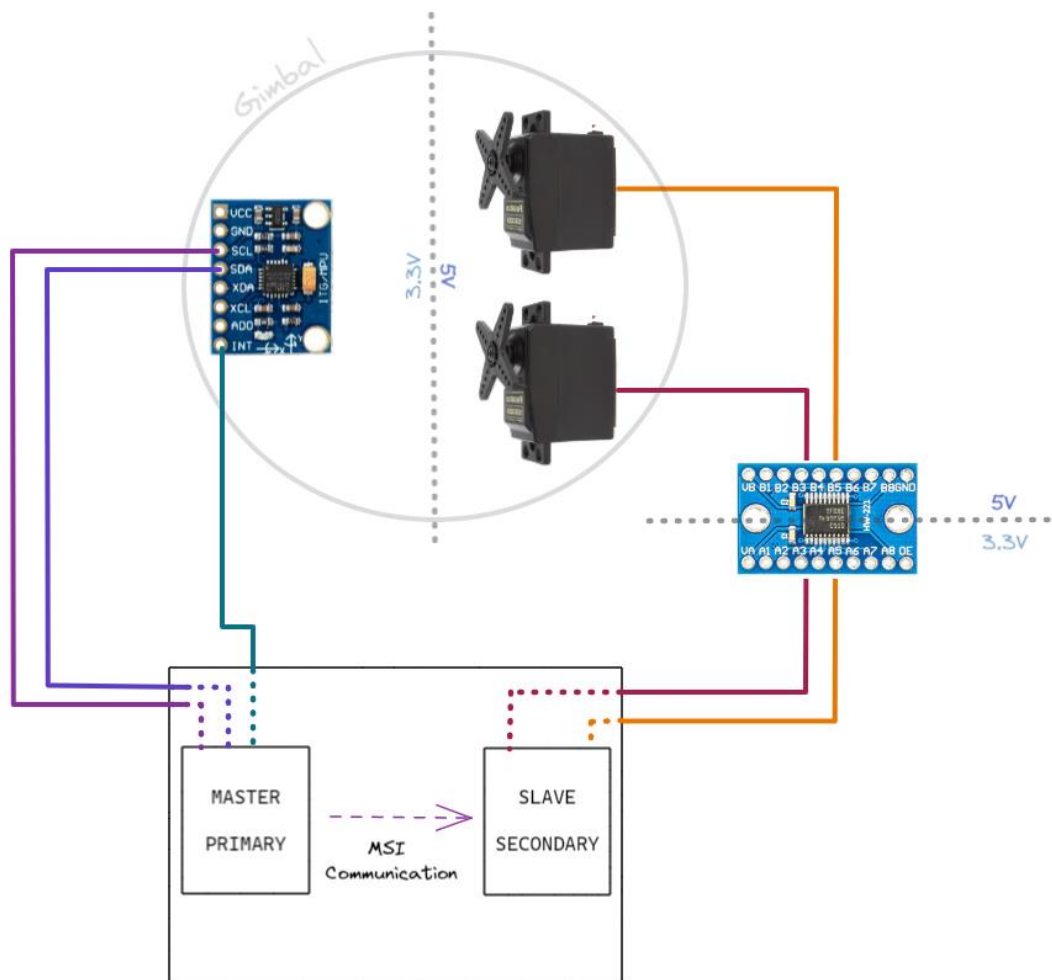


Figure 1. Diagrama del lazo cerrado incluyendo los componentes

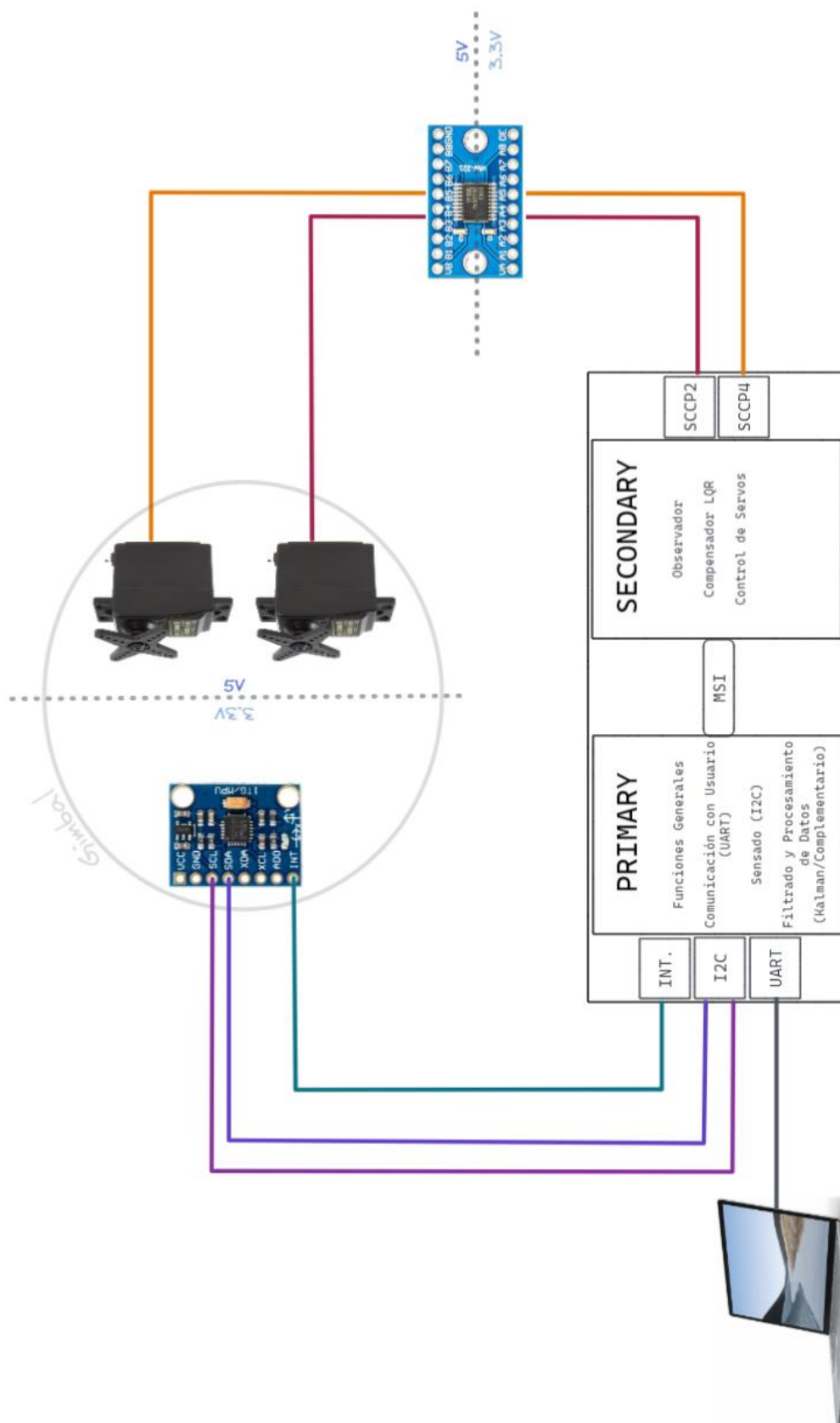


Figure 2. Diagrama del lazo cerrado incluyendo módulos principales

8.3 Ubicación del acelerómetro

La suposición subyacente en la detección de inclinación con un acelerómetro es que el único estímulo de aceleración es el asociado con la gravedad. En la práctica, se puede procesar la señal de salida para eliminar el contenido de alta frecuencia, por lo que se puede tolerar cierta aceleración de AC.

La detección de inclinación utiliza el vector de gravedad y su proyección sobre los ejes del acelerómetro para determinar el ángulo de inclinación. Debido a que la gravedad es una aceleración de DC, cualquier fuerza que resulte en una aceleración de DC adicional corrompe la señal de salida y da como resultado un cálculo incorrecto. Las fuentes de aceleración de DC incluyen: periodos de tiempo en los cuales el objeto acelera a una velocidad constante, o en dispositivos giratorios que inducen una aceleración centrípeta en el acelerómetro. Además, la rotación de un acelerómetro a través de la gravedad provoca una aparente aceleración de AC a medida que cambia la proyección de la gravedad en los ejes de interés.

Por estas razones mencionadas, es muy importante señalar que el acelerómetro, de acuerdo con el diseño del gimbal, debe estar alineado con el eje de ambos motores.

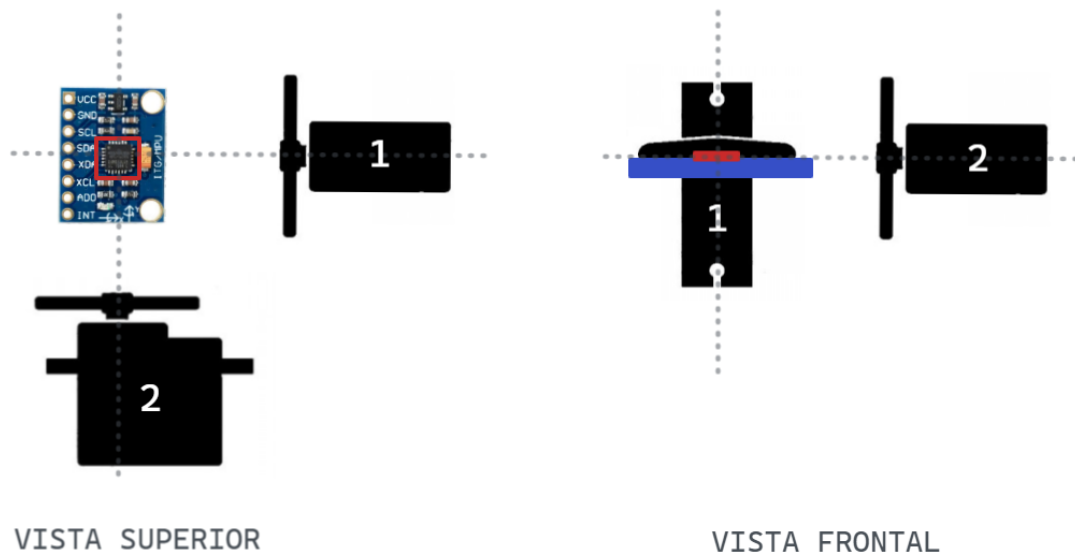


Figure 3. Ubicación deseable del sensor

Cuanto más alejado este de dichos ejes, aumentará la aceleración centrípeta y tangencial registrada, induciendo a mayores errores en el cálculo del ángulo de inclinación. Al modelo 3D anteriormente mencionado, fue necesario agregarle un soporte adicional para poder centrar correctamente al sensor.

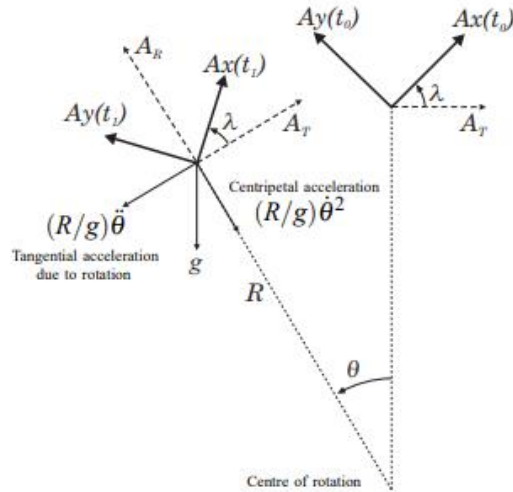


Imagen 59. Efecto de un movimiento rotatorio sobre la aceleración cuando el sensor no se encuentra en el eje de giro

8.4 Filtrado y Procesamiento de Datos

Luego de obtener tan buenos resultados en las simulaciones de fusión de datos, en las cuales incluso el filtro complementario presentaba un desempeño similar al de Kalman, se esperaba que al estar bien diseñado el lazo de compensación la plataforma estabilizara. Este no fue el caso. Esto se debió principalmente a que se ensayó la respuesta de ambos filtros con datos que cambiaban relativamente lento. En el sistema real, los datos cambian más rápidamente y existen diferentes fuentes de error, como la vibración de los servos que se traslada a la plataforma y por ende al sensor. En este contexto el filtro de Kalman va a presentar un mucho mejor desempeño que el filtro complementario.

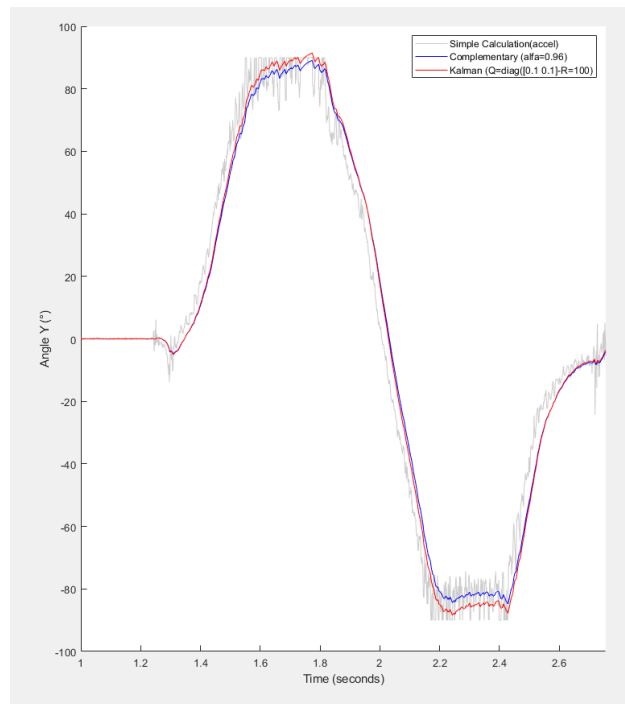


Imagen 60. Filtro Complementario VS Kalman

La plataforma era capaz de seguir el movimiento pero no era capaz de mantener una posición estable. Cuando no se debían registrar desplazamientos y la plataforma estaba alineada con los valores de referencia, esta empezaba a vibrar rápida y notablemente alrededor del supuesto punto de reposo. Este era un comportamiento muy poco deseable. En este punto se tuvo que priorizar la estabilidad y la inmunidad al ruido y a las vibraciones frente al retardo de posicionamiento. Se decidió realizar un filtrado más fuerte, sintonizando nuevamente el filtro de Kalman.

$$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$R = [100]$$

A pesar de esto el retardo del filtro no es tan apreciable como se esperaba ya que domina la respuesta más lenta de los servos.

8.4.1 Filtro Pasa Bajos

Se encontró que además de volver a sintonizar el filtro de Kalman era conveniente filtrar los datos provistos por el acelerómetro debido a que tenían mucho ruido. No se utilizó el filtro pasa bajos embebido en el módulo MPU6050 ya que introducía un delay notorio y variable.

El filtro implementado tendrá la siguiente forma:

$$\theta_{filtered\ accel_k} = \alpha \theta_{filtered\ accel_{k-1}} + (1 - \alpha) \theta_{accel_k} \quad \alpha \leq 1$$

Se elegirá $\alpha = 0.975$

8.4.2 Interferencia entre ejes

Luego de todo esto, a pesar de estar aplicado un filtrado más fuerte, seguía sin poder estabilizarse a la plataforma. Se realizó un gran estudio del tema y se encontró que movimientos en un eje causaban grandes aumentos en la velocidad angular medida en los otros ejes.

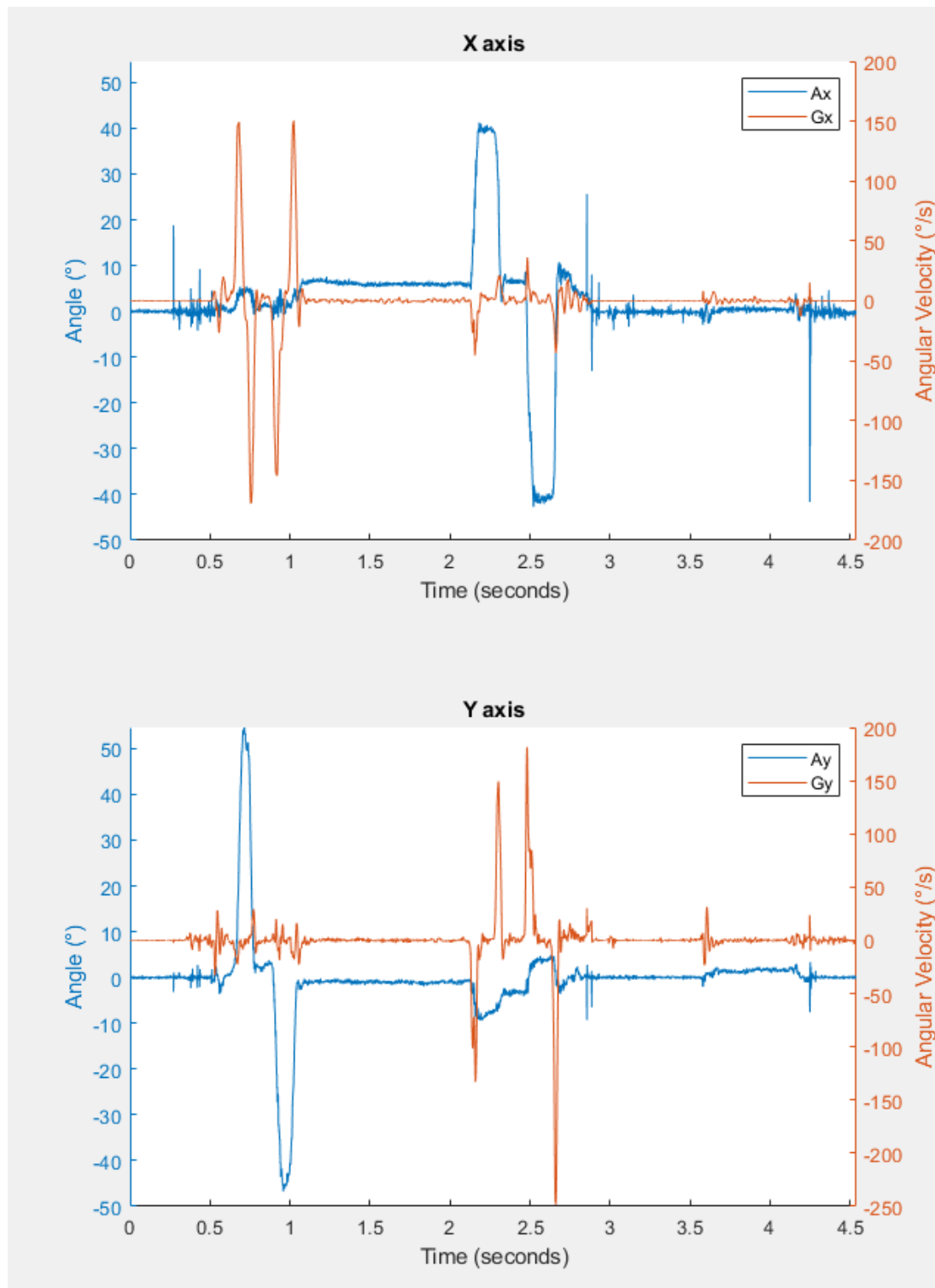


Imagen 61. Interferencia entre ejes

Por esta razón se implementó un mecanismo de bloque que entra en funcionamiento cuando hay una diferencia muy grande entre el ángulo medido por el acelerómetro y la velocidad angular registrada. Con esto se logró estabilizar la plataforma. Referirse al apéndice para más información.

8.5 Kalman VS Complementario en Reposo

Como se mencionó, lo más difícil de lograr fue lograr que la plataforma estabilizara en reposo. Por esta razón se evaluó el desempeño de los filtros en esta condición.

Simplemente se tuvieron que cambiar dos líneas del programa.

```
145 fused_angle_X=kalman_fusion_X(gX,aX);  
146 fused_angle_Y=kalman_fusion_Y(gY,aY);  
147 //fused_angle_X=alfa_complementary*(fused_angle_X+Ts*gX)+(1-alfa_complementary)*aX;  
148 //fused_angle_Y=alfa_complementary*(fused_angle_Y+Ts*gY)+(1-alfa_complementary)*aY;
```

Imagen 62. Líneas a intercambiar el filtro de Kalman por el filtro complementario

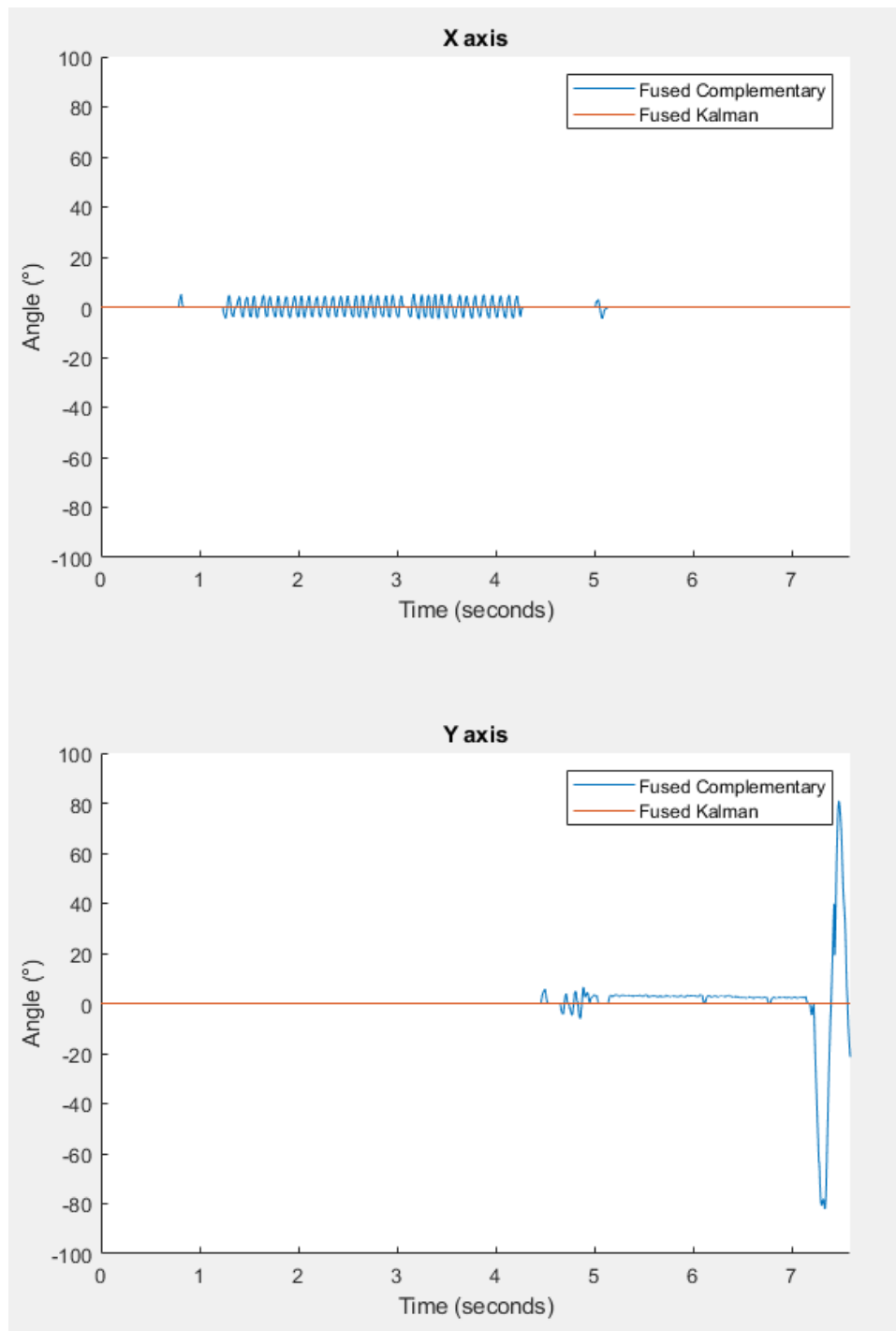


Imagen 63. Respuesta de la plataforma al utilizar el filtro complementario o el filtro de Kalman

A simple vista es fácil de ver que el filtro de Kalman tiene un mejor comportamiento. Al utilizar el filtro complementario notamos que por momentos la plataforma estabiliza, pero que sale de este estado sin recibir ningún estímulo externo, luego de esto comienza a oscilar. También notamos al analizar el eje "Y" que en un momento se produce un bloque del ángulo registrado lo que posteriormente va a producir movimientos erráticos de la plataforma.

8.6 Costo del dispositivo

El costo del dispositivo no excede los USD30 . Los dos servos, el sensor y el convertor de nivel costaron menos de USD14, y aunque en este trabajo se utilizó la placa de desarrollo de DM330028-2, alcanza con utilizar el dsPIC33CH512MP508 ya que no se utiliza ninguna de las funcionalidades de la placa. El DSC cuesta USD8.69.



[Futaba - Servo S3003 3.2kg](#)
[Excelente Calidad III](#)
\$ 1.090,00
Mercadolibre.co...



Giróscopo Acelerómetro Mpu6050 Arduino
\$ 640,00
Mercadolibre.co...



[Conversor De Nivel Arduino Txs0108e Hw221 5v 3 3v 8ch](#)
\$ 479,00
Mercadolibre.co...

DÓLAR BLUE

↑ 6,23%

| | |
|------------------|------------------|
| Compra | Venta |
| \$ 268,00 | \$ 273,00 |

Actualizado: 11.07.2022 01:42



Part Number:
DSPIC33CH512MP508-H/PT
Quantity:
1
Price per Unit (in USD):
8.69
Total Amt:
8.69

Imagen 64. Costo de los insumos

8.7 Para tener en cuenta en futuros desarrollos

A continuación se enumerarán un conjunto de modificaciones que permitirán tener un aumento en la performance. Se recomienda:

- Utilizar sensores de mejor calidad. Al no tener que aplicar un mecanismo de bloque ni un filtro tan fuerte disminuirá el retardo. Igual se debe tener en cuenta que el acceso a los buenos acelerómetros está restringido militarmente.
- Enfriar el sensor con una celda Peltier para reducir el ruido. No se lo debe sobre enfriar porque esto trae efectos adversos.
- Agregar un magnetómetro. Con Kalman será posible fusionar los datos de los 3 sensores minimizando el ruido.
- Aplicar el “Extended Kalman Filter” (EKF). Esta es una extensión del filtro de Kalman a sistemas no lineales. Considerando que los movimientos circulares de la plataforma pueden ser más precisamente representados con senos y cosenos, hubiese sido relevante implementarlo. Esto no fue posible, ya que al requerir operaciones más complejas la velocidad del microprocesador no lo permitía.
- Utilizar servos con una respuesta más rápida, o reemplazarlos por motores de DC. Al aceptar un aumento en el tiempo de estabilización se redujo el alcance de aplicación de la plataforma presentada en este trabajo ya que no va a responder de la mejor manera a cambios amplios y bruscos de posición. De cualquier forma, teniendo en cuenta que los movimientos de una persona caminando, portando un gimbal no son sumamente erráticos la plataforma sigue cumpliendo el alcance de diseño.
- Utilizar comunicación inalámbrica para la transmisión de datos.

IX. CONCLUSION

Se logró implementar un prototipo funcional de bajo costo de un “gimbal” apto para fotografía y video, desarrollando un control LQR y un filtro de Kalman. Consideramos que se alcanzaron excelentes resultados ya que lograr un buen desempeño en un gimbal es una tarea muy difícil, es deseable bajo retardo, alta precisión y buena estabilidad.

Para esto es fundamental contar con datos de posicionamiento fiables. Es complicado lograr un equilibrio entre estos requerimientos ya que, al momento de aplicar un filtro, si quisiésemos minimizar el ruido aumentaremos el retardo. A pesar de esto, considerando que el módulo MPU6050 es un módulo demasiado ruidoso (es un módulo para “gaming”), se logró una buena respuesta del sistema. En la mayoría de las aplicaciones donde se lo utiliza se logra acotar el ruido dentro de una banda de 3° , mientras que en este trabajo se logró reducirlo a una banda de 0.1° . Por lo tanto, concluimos que la aplicación del filtro de Kalman resultó exitosa.

A su vez se verificó que para poder controlar un espacio de estados de estas características es fundamental contar con una alta capacidad de procesamiento.

Utilizar un procesador de doble núcleo permitió dividir el código de una manera organizada, donde la comunicación entre ambas procesadoras era fiable y sencilla de configurar. Esto permitió una buena sincronización y eficiencia al realizar las distintas tareas.

Al trabajar con elementos de bajo costo y calidad se encontraron muchos obstáculos en el camino que no se hubiesen presentado, de todas maneras, resolverlos dejó una interesante experiencia.

X. BIBLIOGRAFIA

<https://www.microchip.com/en-us/product/dsPIC33CH512MP508>

<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>

https://www.editores-srl.com.ar/revistas/aa/4/micro_servomotores

<https://mjwhite8119.github.io/Robots/mpu6050>

<https://www.motioncontroltips.com/how-do-gearmotors-impact-reflected-mass-inertia-from-the-load/>

<https://blog.orientalmotor.com/10-advantages-of-geared-motors>

<https://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition§ion=ControlStateSpace>

<https://iopscience.iop.org/article/10.1088/1742-6596/890/1/012056/pdf>

https://www.st.com/resource/en/application_note/an4509-tilt-measurement-using-a-lowg-3axis-accelerometer-stmicroelectronics.pdf

<https://www.analog.com/en/app-notes/an-1057.html#:~:text=Inclination%20sensing%20uses%20the%20gravity,result%20in%20an%20incorrect%20calculation.>

https://www.st.com/resource/en/application_note/dm00729932-precise-and-accurate-tilt-sensing-in-industrial-applications-stmicroelectronics.pdf

<https://ebuah.uah.es/dspace/bitstream/handle/10017/30784/TFG-Tof%C3%A9-Blanco-2015.pdf?sequence=1&isAllowed=y>

https://www.researchgate.net/publication/275957000_Experimental_comparison_of_Kalman_and_complementary_filter_for_attitude_estimation/figures?lo=1

<https://la.mathworks.com/videos/understanding-kalman-filters-part-4-optimal-state-estimator-algorithm--1493129749201.html>

https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

<https://arxiv.org/ftp/arxiv/papers/1702/1702.00884.pdf>

https://www.researchgate.net/publication/224289587_A_New_Approach_to_Accelerometer-based_Head_Tracking_for_Augmented_Reality_Other_Applications

<https://www.thingiverse.com/thing:2909483>

<https://zipcpu.com/dsp/2017/08/19/simple-filter.html>

<http://ww1.microchip.com/downloads/en/DeviceDoc/51459b.pdf>

“An introduction to inertial navigation”-Oliver J. Woodman

“Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration” -
Aboelmagd Noureldin , Tashfeen B. Karamat , Jacques Georgy

“Global Positioning Systems, Inertial Navigation, and Integration” - Lawrence Randolph Weill,
Angus P. Andrews, Mohinder S. Grewal

XI. ANEXOS

11.1 ANEXO A - Implementación Sistema en dsPIC

```
#ifndef SIMULACION_VAR_Y_OP_TF4_H
#define SIMULACION_VAR_Y_OP_TF4_H

#include "operaciones_matematicas.h"

//Valores usados para modelar el sistema
float Ts=0.005;

//Matrices para implementar el observador discreto
double Ao[4][4]={
{0.9220625814133491,0.00439986477711214,8.036709265956757e-
18,1.949296258796061e-20},
{0.02013847128264393,0.8392491610649864,-
8.598148984527296e-18,-2.123117268762312e-20},
{1.005012297350524e-17,2.43502992379126e-
20,0.9220625814133493,0.004399864777112141},
{2.420672684497533e-18,5.865300226703109e-
21,0.02013847128264394,0.8392491610649864} };
double Bo[4][4]={ {1.14831115608714e-05,3.343557974297475e-
23,1.443412920995785e-09,-1.62699636421671e-25},
{0.004586345091131213,-3.675751481299918e-23,-
5.261023051151722e-08,1.740769607100813e-25},
{4.174351922088184e-23,1.14831115608714e-05,-
2.034595287894454e-25,1.443412920995781e-09},
{1.005509613178101e-23,0.004586345091131213,-
4.900527197963911e-26,-5.261023051151722e-08} };
double Co[4][4]={ {1,0,0,0},
{0,1,0,0},
{0,0,1,0},
{0,0,0,1} };
double Do[4][4]={ {0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0} };

//Kcd+Kid ((LQR+Integrator))
volatile double Kcd[2][4]={
{424.818741494127,10.57019043727758,0,0},

{0,0,424.818741494127,10.57019043727758} };
volatile double nKcd[2][4]={ {-424.818741494127,-
10.57019043727758,0,0},
{0,0,-424.818741494127,-
10.57019043727758} }; //Hacer -Kd*Y[n] == +(-Kd)*Y[n]
volatile double Kid[2][2]={ {-0.0001451313980376144,0},
{0,-0.0001451313980376144} };
volatile double nKid[2][2]={ {0.0001451313980376144,0},
```

```

                                {0,0.0001451313980376144} };
//Hacer  $-K_d Y[n] == +(-K_d) * Y[n]$ 

//Matrices para implementar el integrador discreto
double Ai[2][2]={ {1,0},
                  {0,1} };
double Bi[2][2]={ {0.005,0},
                  {0,0.005} };//Ts
double Ci[2][2]={ {1,0},
                  {0,1} };
double Di[2][2]={ {0,0},
                  {0,0} };

//Declaraciones de valores de Referencia
double reference[2][1]={ {0},
                          {0} };

//X[n],X[n+1],Y[n] y U[n]. Condiciones Iniciales
volatile double X_0[4][1]={0},{0}}; //X[n]
volatile double X_1[4][1]={0},{0}}; //X[n+1]
volatile double Y[2][1]={0}};
volatile double U[2][1]={0}}; //Debo fijarla como matriz ya que
debo multiplicarla
volatile double Xo_0[4][1]={0},{0}}; //Xo[n]
volatile double Xo_1[4][1]={0},{0}}; //Xo[n+1]
volatile double Yo[4][1]={0},{0}}; //Yo==Xo_hat
//OUTER LOOP
volatile double Xi_0[2][1]={0}}; //Xi[n]
volatile double Xi_1[2][1]={0}}; //Xi[n+1]
volatile double Yi[2][1]={0}}; //Yi==Xi_hat

//Variables auxiliares necesarias para la ejecucion del programa
//OUTER LOOP
volatile double nY[2][1];
volatile double Ui[2][1];
volatile double nOutputOL[2][1]; // Analogo MUX Simulink. Will
output the multiplication by  $-K_i$  so then I can sum it instead of
resting it
//INNER LOOP
volatile double aux_1_planta[4][1]; //  $A * X[n]$ 
volatile double aux_2_planta[4][1]; //  $B * U[n]$ 
volatile double aux_1[4][1]; //  $Ao * Xo[n]$ 
volatile double aux_2[4][1]; //  $B * U[n]$ 
volatile double aux_3[2][1]; //  $-K_d * Y[n]$  (en el caso de
planta()) o  $-K_d * Yo[n]$  (en el caso de plantaPLUSobservador())
volatile double inputObs[4][1]; // Analogo MUX Simulink

#endif

#ifdef SIMULACION_H
#define SIMULACION_H

#include "simulacion_variables_y_operaciones.h"

//Implementacion Sistema

```

```

void system();
void inner_loop();
void outer_loop();

void outerLoop(){
//FIRST ADDER (|+-)
    nY[0][0]=Y[0][0]*(-1);
    nY[1][0]=Y[1][0]*(-1);
    Ui[0][0]=reference[0][0] + nY[0][0];
    Ui[1][0]=reference[1][0] + nY[1][0];

//INTEGRATOR
    Xi_1[0][0]=Ai[0][0]*Xi_0[0][0]+Bi[0][0]*Ui[0][0];
    Xi_1[1][0]=Ai[1][1]*Xi_0[1][0]+Bi[1][1]*Ui[1][0];
    Yi[0][0]=Xi_0[0][0]; //C es la identidad entonces puedo
hacer esto
    Yi[1][0]=Xi_0[1][0]; //C es la identidad entonces puedo
hacer esto

    Xi_0[0][0]=Xi_1[0][0];
    Xi_0[1][0]=Xi_1[1][0];

//Ki
    nOutputOL[0][0] = Yi[0][0]*nKid[0][0];
    nOutputOL[1][0] = Yi[1][0]*nKid[1][1];
}

void innerLoop(){
//OBSERVADOR
    //"Multiplexo" la entrada al observador para poder usar la
representacion del espacio de estados usada en simulink
    inputObs[0][0]=U[0][0];
    inputObs[1][0]=U[1][0];
    inputObs[2][0]=Y[0][0];
    inputObs[3][0]=Y[1][0];

    //Xo[n+1]=A*Xo[n]+B*inputObs[n]
    fM1=4;matrix_multiply_c4_c1(Ao,Xo_0,aux_1); //Ao*Xo[n]
    fM1=4;matrix_multiply_c4_c1(Bo,inputObs,aux_2);
//Bo*inputObs[n]
    fM1=4;matrix_sum_c1(aux_1,aux_2,Xo_1);

    //Como Co es la matriz identidad se puede dar una asignación
directa (reduzco los tiempos de ejecutado)
    //Si no fuese la matriz identidad, debo reemplazar por:
fM1=2;matrix_multiply_c2_c1(Co,Xo_0,Yo);
    Yo[0][0]=Xo_0[0][0];
    Yo[1][0]=Xo_0[1][0];
    Yo[2][0]=Xo_0[2][0];
    Yo[3][0]=Xo_0[3][0];

    //Xo[n]=Xo[n+1] (Para la próxima vez que se invoque al
observador)
    Xo_0[0][0]=Xo_1[0][0];
    Xo_0[1][0]=Xo_1[1][0];
    Xo_0[2][0]=Xo_1[2][0];

```

```

        Xo_0[3][0]=Xo_1[3][0];

//REALIMENTACION Y TUNNING
//U[n]=r[n]*Nbard-Kcd*Yo[n]=r[n]*Nbard+(-Kcd)*Yo[n]
fM1=2;matrix_multiply_c4_c1(nKcd,Yo,aux_3); //-Kd*X[n]
fM1=2;matrix_sum_c1(nOutputOL,aux_3,U);
}

void system(){
    innerLoop();
    outerLoop();
}

#endif

```

11.2 ANEXO B - Implementación del filtro de Kalman en MatLab

```

%% Kalman
global A B H
Ts=0.001;

A = [1 -Ts; 0 1];
B = [Ts; 0];
H = [1 0]; %C

Q = [0.3 0.7;0.7 0.3];
R = [100];

state_estimate_actual = [0;0];
P_post = eye(2)*0.02;

kalman=[];

for i=1:length(ang_vel_gyro)
    %PREDICT - Prior Measurement
    state_estimate_prior = A*state_estimate_actual+B*ang_vel_gyro(i);
    P_prior = A*P_post*A'+Q;
    %UPDATE
    K_gain = P_prior*H'/(H*P_prior*H'+R);
    Z_meas = angle_accel(i); %y
    state_estimate_actual=state_estimate_prior+K_gain*(Z_meas-
H*state_estimate_prior);
    P_post =(eye(2)-K_gain*H)*P_prior;

    kalman(end+1)=state_estimate_actual(1,1);
end

```


11.3 ANEXO C - Algunos Tiempos de Procesamiento

| Function Group | Function | Performance (Cycles) ^{1, 3, 4} |
|------------------------------|----------------|---|
| Basic Floating Point | Addition | 122 |
| | Subtraction | 124 |
| | Multiplication | 109 |
| | Division | 361 |
| | Remainder | 385 |
| Trigonometric and Hyperbolic | acos | 478 |
| | asin | 363 |
| | atan | 696 |
| | atan2 | 3206 |
| | cos | 3249 |
| | sin | 2238 |
| | tan | 2460 |
| | cosh | 1049 |
| | sinh | 525 |
| | tanh | 338 |
| Logarithmic and Exponential | exp | 530 |
| | frexp | 39 |
| | ldexp | 44 |
| | log | 2889 |
| | log10 | 3007 |
| Power Functions | pow | 2134 |
| | sqrt | 493 |
| Rounding Functions | ceil | 94 |
| | floor | 51 |
| Absolute Value Function | fabs | 6 |
| Modular Arithmetic Functions | modf | 151 |
| | fmod | 129 |

Tabla 6. Algunos tiempos de procesamiento

11.4 ANEXO D - Mecanismo de Bloqueo

```

aX=alfa*aX+(1-alfa)*angle_accel.ax;
aY=alfa*aY+(1-alfa)*angle_accel.ay;

gX=ang_vel_gyro.gx;
gY=ang_vel_gyro.gy;

if ( fabs(gX/aX)>2.7 ) {gX=0;aX=0;bufferX=1;countX=0;}
if ( fabs(gY/aY)>2.7 ) {gY=0;aY=0;bufferY=1;countY=0;}

if( fabs(angle_accel.ax)>4.4 ) {bufferX=0;countX=0;}
if( fabs(angle_accel.ay)>4.4 ) {bufferY=0;countY=0;}

if ( bufferX==1 ) {
    countX++;
    aX=0;
    gX=0;
    if (countX==60) {countX=0; bufferX=0;}
}
if ( bufferY==1 ) {
    countY++;
    aY=0;
    gY=0;
    if (countY==60) {countY=0; bufferY=0;}
}

count++;

```