

Grado en Ingeniería Aeroespacial

EQUIPOS Y SISTEMAS DE CONTROL AEROESPACIALES

CURSO 2021-2022

ENTREGA 4



José Luis Dugo Ortega
Elena Gallego Hernández
Alejandro Sánchez Torres
Álvaro Jiménez Melendo

23 de diciembre de 2021

Índice

1. Ejercicio 8: Control del sustentador	2
1.1. Introducción	2
1.2. Modelo lineal de la planta	3
1.2.1. Representación en el espacio de estados	5
1.3. Implementación del modelo lineal	8
1.4. Simulink	8
1.5. Comparativa Hover	11
 2. Ejercicio 9: Filtrado complementario	 13
2.1. Introducción	13
2.2. Filtrado complementario de primer orden. <i>Dominio temporal</i>	13
2.3. Filtrado complementario de primer orden. <i>Dominio de Laplace</i>	16
2.4. Filtrado complementario de segundo orden. <i>Dominio de Laplace</i>	19
2.5. Filtrado complementario discreto	21
 3. Ejercicio 10: Control del helicóptero	 23
3.1. Introducción	23
3.2. Representación en el espacio de estados y modelo lineal	23
3.3. Diagrama de simulación	24
3.4. Resultados obtenidos	25
3.5. Resultados mediante script de matlab	28

1. Ejercicio 8: Control del sustentador

1.1. Introducción

El sustentador Quanser 3-DOF se trata de un modelo de helicóptero montado en una base fija con cuatro hélices impulsadas por motores DC, es decir, se trata de una cuadricóptero. El bastidor contiene una gran articulación que permite el movimiento con tres grados de libertad así la estructura pivotante gira en los tres ejes: balance, cabeceo y guiñada.



Figura 1: Quanser 3-DOF

Cada hélice genera una fuerza de elevación y las fuerzas de elevación se utilizan para controlar los ángulos de cabeceo y balance. Además, cabe decir que el par total generado por los motores de las hélices hace que el cuerpo se mueva alrededor de eje de guiñada y, como las hélices son contrarrotantes dos a dos, el par total del sistema está equilibrado cuando el empuje de las cuatro hélices es aproximadamente igual. Al igual que en prácticas anteriores, el ángulo de los ejes se mide utilizando encoders de alta resolución.

Para los siguientes apartados serán necesarios ciertos valores específicos de el Quanser 3-DOF:

$$\begin{aligned}
K_t &= 0,0036 N \cdot m/V \\
K_f &= 0,1188 N/V \\
L &= 0,197 m \\
m_{hover} &= 2,85 kg \\
J_y &= 0,110 kg \cdot m^2 \\
J_p &= 0,0552 kg \cdot m^2 \\
J_r &= 0,0552 kg \cdot m^2
\end{aligned}$$

1.2. Modelo lineal de la planta

Lo primero a realizar es un convenio de las medidas utilizadas, de manera que definimos que el sustentador está horizontal (paralelo al suelo), cuando los ángulos de cabeceo y balance son cero. Además, el ángulo de guiñada θ_y es creciente cuando el cuerpo gira en el sentido contrario a las agujas del reloj (CCW). El ángulo de cabeceo θ_p crece cuando el eje gira en sentido antihorario y, finalmente, el ángulo de balance θ_r crece cuando el eje gira en sentido antihorario.

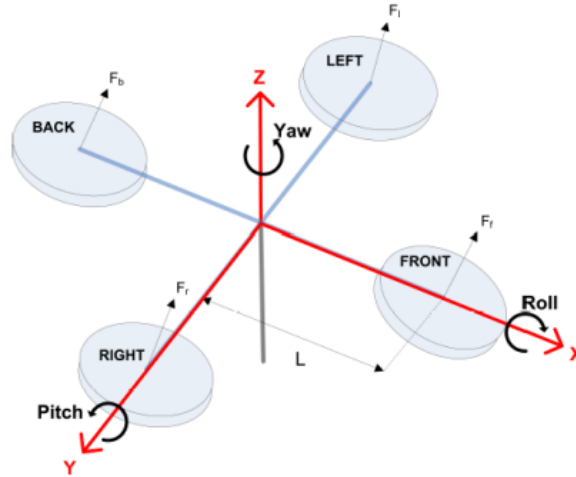


Figura 2: Convenio empleado

Cada uno de los ejes se debe modelar por separado. La dinámica para cada eje se puede describir por la ecuación general siguiente:

$$J\ddot{\theta} = \Delta F \cdot L$$

donde θ es el ángulo del pivote, L es la distancia entre el motor propulsor y el eje del pivote, J es el momento de inercia alrededor del eje, y ΔF es el diferencial de la fuerza de empuje.

Modelo del eje de cabeceo (Pitch)

En cuanto al modelo de la dinámica del eje de cabeceo, se puede particularizar de la siguiente manera:

$$J_p \ddot{\theta}_p = LK_f(V_f - V_b)$$

donde K_f es la constante de empuje, V_f es el voltaje de motor delantero (front), V_b es el voltaje del motor trasero (back), T_p es el ángulo de cabeceo (pitch), J_p es el momento de inercia respecto al eje de cabeceo y L es la distancia al punto de aplicación. Se dice que el ángulo de cabeceo aumenta cuando la tensión frontal es mayor que la trasera.

Modelo del eje de balance (Roll)

En cuanto al modelo de la dinámica del eje de balance, se puede particularizar de la siguiente manera:

$$J_r \ddot{\theta}_r = LK_f(V_r - V_l)$$

donde K_f es la constante de empuje, V_r es el voltaje del motor derecho, V_l es el voltaje del motor izquierdo, T_r es el ángulo de balance, J_r es el momento de inercia respecto al eje del balance y L es la distancia al punto de aplicación. Además sabemos que en este caso el ángulo de balance aumenta cuando la tensión del motor derecho es mayor que la del motor izquierdo.

Modelo del eje de guiñada (Yaw)

Finalmente, en cuanto al modelo de la dinámica del eje de guiñada podemos decir que se particulariza por la siguiente ecuación:

$$J_y \ddot{\theta}_y = \Delta\tau = \tau_l + \tau_r - \tau_f - \tau_b$$

donde τ_l y τ_r son los torques generados por las hélices izquierda y derecha (sentido horario) y τ_f y τ_b son los pares ejercidos por los rotores frontal y posterior (sentido antihorario). Según la convención de signos los pares antihorarios son negativos.

1.2.1. Representación en el espacio de estados

Como ya se ha estudiado hasta ahora, la representación en el espacio de estados del modelo lineal para el sistema sustentador 3DOF usa la siguiente expresión:

$$\begin{aligned}\dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \\ y(t) &= C \cdot x(t) + D \cdot u(t)\end{aligned}$$

donde, como ya se comentó en otras prácticas, x es el vector de estados, y el vector de salida, y u es el vector de control. Para definir el espacio de estados, como ya se sabe, hay que definir cuatro matrices. Para ello hacen falta los vectores nombrados anteriormente:

$$\begin{aligned}x &= [\theta_y \ \theta_p \ \theta_r \ \dot{\theta}_y \ \dot{\theta}_p \ \dot{\theta}_r] \\ y &= [\theta_y \ \theta_p \ \theta_r] \\ u &= [V_f \ V_b \ V_r \ V_l]\end{aligned}$$

Seguidamente se realizan las derivadas correspondientes para hallar el espacio de estados:

$$\begin{aligned}x_1 &= \dot{\theta}_y = x_4 \\ x_2 &= \dot{\theta}_p = x_5 \\ x_3 &= \dot{\theta}_r = x_6 \\ x_4 &= \ddot{\theta}_y = \frac{K_t}{J_y}(V_r + V_l) - \frac{K_t}{J_y}(V_f + V_b) \\ x_5 &= \ddot{\theta}_p = -\frac{LK_f}{J_p}(V_f - V_b) \\ x_6 &= \ddot{\theta}_r = -\frac{LK_f}{J_r}(V_r - V_l)\end{aligned}$$

Definido lo anterior, se puede pasar a definir las matrices que acabarán por completar el espacio de estados:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{LK_t}{J_p} & -\frac{LK_t}{J_p} & 0 & 0 \\ 0 & 0 & \frac{LK_t}{J_p} & -\frac{LK_t}{J_p} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

y con las matrices jacobianas definidas, queda definida también la representación en el espacio de estados.

En cuanto a la estabilidad y la controlabilidad del sistema, se deben analizar los autovalores, es decir, los polos de la ecuación característica del sistema. Estos cálculos se realizan en el Script de Matlab, tanto para lazo abierto como cerrado, aunque es obvio que si en lazo abierto el sistema es estable, en lazo cerrado también lo será.

Una vez se compila, Matlab devuelve que el sistema en lazo abierto es controlable, por lo que se puede concluir que el sistema es estable en lazo cerrado.

Seguidamente se pide que se realice el mismo proceso que anteriormente, pero ahora con un vector de estados $x = [\theta_y \ \theta_p \ \dot{\theta}_r \ \dot{\theta}_y \ \dot{\theta}_p \ \int \theta_p]$ y todos los demás vectores iguales.

De esta manera las matrices jacobianas quedan de la siguiente manera:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -2,92 & 0 & 0 & -2,59 & 0 & 0 \\ 0 & -26,81 & 0 & 0 & -11,2 & 0 \\ 0 & 0 & -26,81 & 0 & 0 & -11,2 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{LK_t}{J_p} & -\frac{LK_t}{J_p} & 0 & 0 \\ 0 & 0 & \frac{LK_t}{J_p} & -\frac{LK_t}{J_p} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

y con eso, de nuevo, la representación en el espacio de estados queda definida.

1.3. Implementación del modelo lineal

De la misma manera que en la práctica anterior, se va a proceder al diseño de un controlador LQR que permita el control del sistema. Como ya se ha explicado en varias ocasiones, el controlador LQR viene definido por dos matrices de ponderación, Q y R . El objetivo es hallar el valor de los elementos de estas matrices para que estas se ajusten a lo deseado.

Hay que tener claro que $Q(1,1)$ repercute sobre el ángulo de guiñada y $Q(4,4)$ a la velocidad de guiñada, $Q(2,2)$ al ángulo de cabeceo y $Q(5,5)$ a la velocidad de cabeceo y, finalmente, $Q(3,3)$ al ángulo de balance y $Q(6,6)$ a la velocidad de balance.

Como se puede observar, todos ellos son valores de la diagonal, por lo que se usará un comando en matlab para definirlos. Por otro lado, también hay que definir los valores de R , que se corresponden con los voltajes aplicados a cada motor, por lo que se va a suponer que son todos iguales:

$$Q = \text{diag}([100 \ 100 \ 100 \ 10 \ 10 \ 10])$$

$$R = \text{diag}([0,05 \ 0,05 \ 0,05 \ 0,05])$$

Para estos valores matriciales, debe corresponder una ganancia K , la cual se va hallar haciendo uso de Matlab.

$$K = \begin{pmatrix} -22,36 & 31,62 & 0 & -19,78 & 13,21 & 0 \\ -22,36 & -31,62 & 0 & -19,78 & -13,21 & 0 \\ 22,36 & 0 & 31,62 & 19,78 & 0 & 13,21 \\ 22,36 & 0 & -31,62 & 19,78 & 0 & -13,21 \end{pmatrix}$$

La ganancia tiene el valor anterior. Se puede concluir que este valor es estable ya que los polos obtenidos en Matlab son todos pertenecientes al semiplano izquierdo del lugar de las raíces y por tanto concluimos que el sistema será estable.

1.4. Simulink

Para acabar de determinar si los valores de las matrices de ponderación son correctos o no se va a hacer uso de Simulink. De esta manera veremos la respuesta del sistema.

Para ello se implementa el siguiente diagrama de simulación:

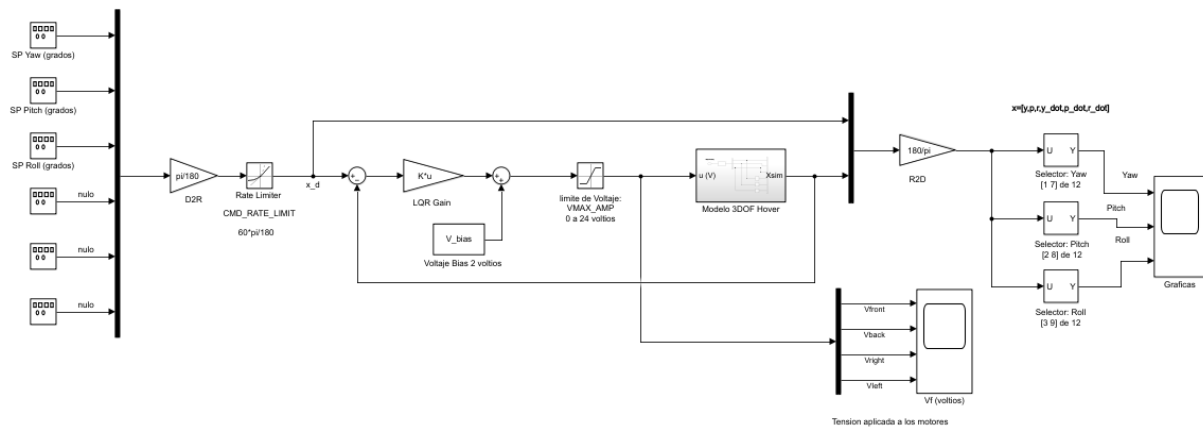


Figura 3: Diagrama de bloques

En primer lugar se tiene la respuesta del sistema de posición angular, como se representa en la siguiente imagen:

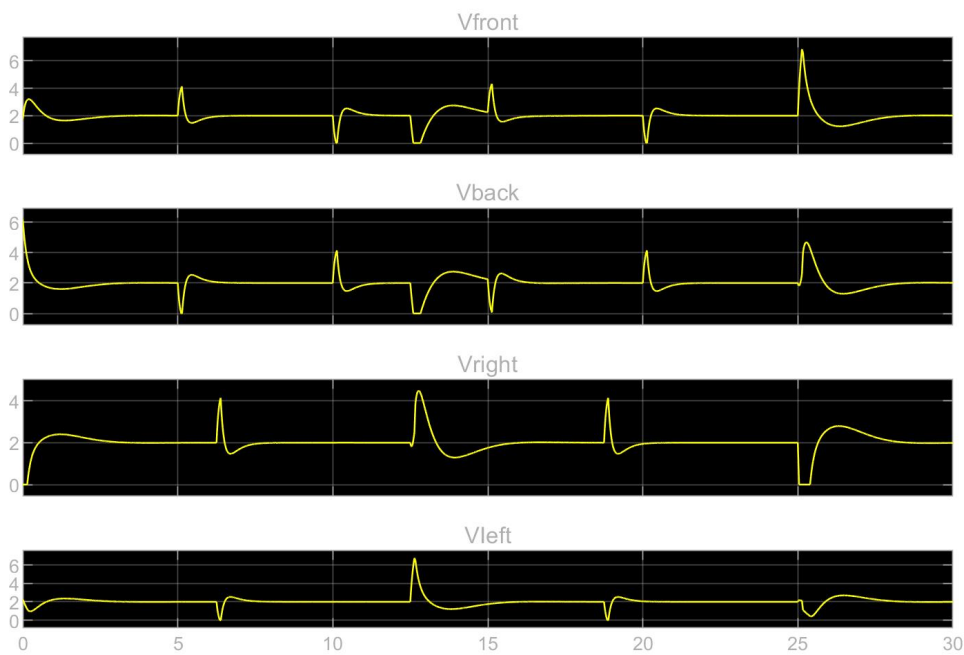


Figura 4: Respuesta del sistema de posición angular

Por último, se puede observar la respuesta del sistema de la tensión aplicada a los motores.

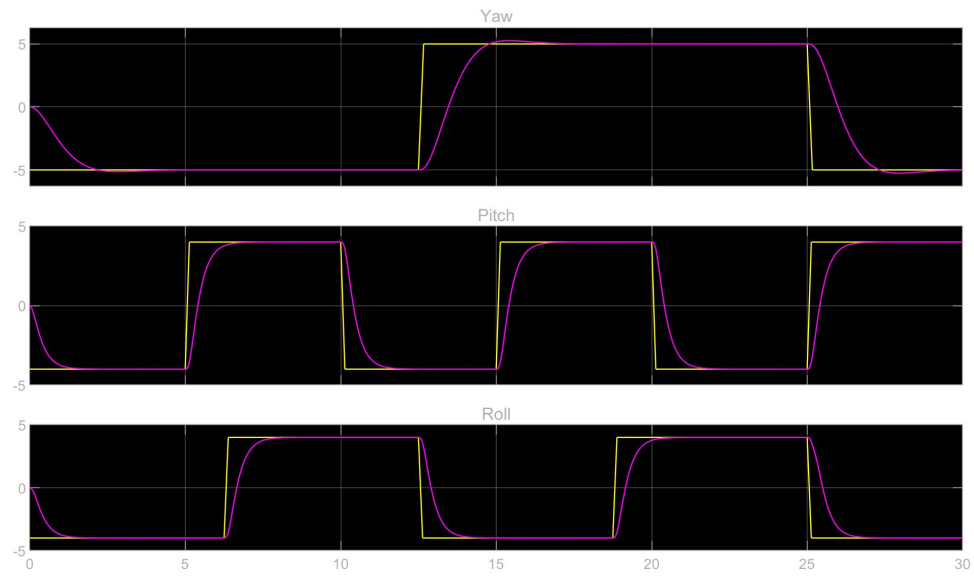


Figura 5: Respuesta del sistema de la tensión aplicada en los motores

1.5. Comparativa Hover

Ha continuación, se va a realizar el estudio con los datos obtenidos en el laboratorio, llevando a cabo una comparación con los resultados obtenidos anteriormente:

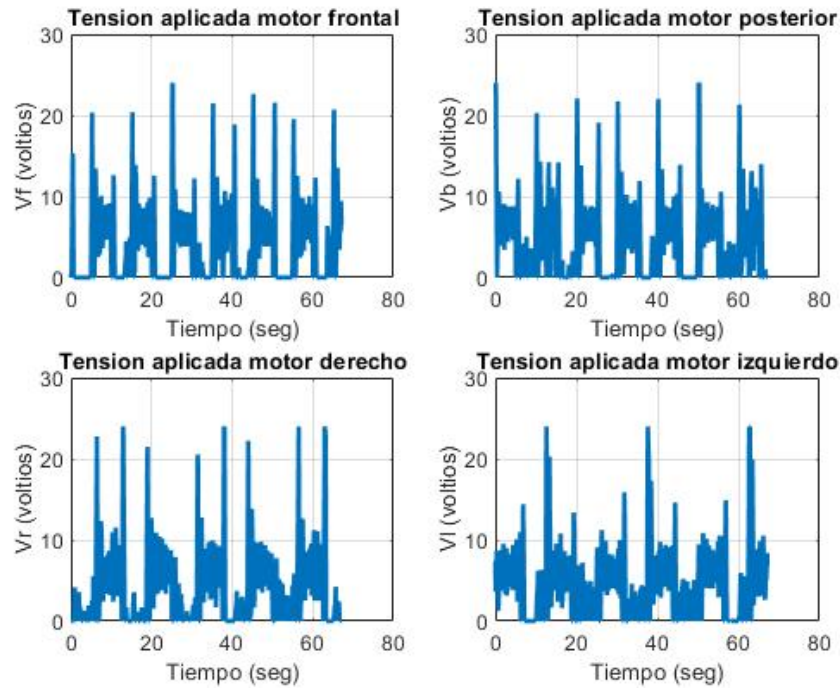


Figura 6: Respuesta del sistema de posición angular hover

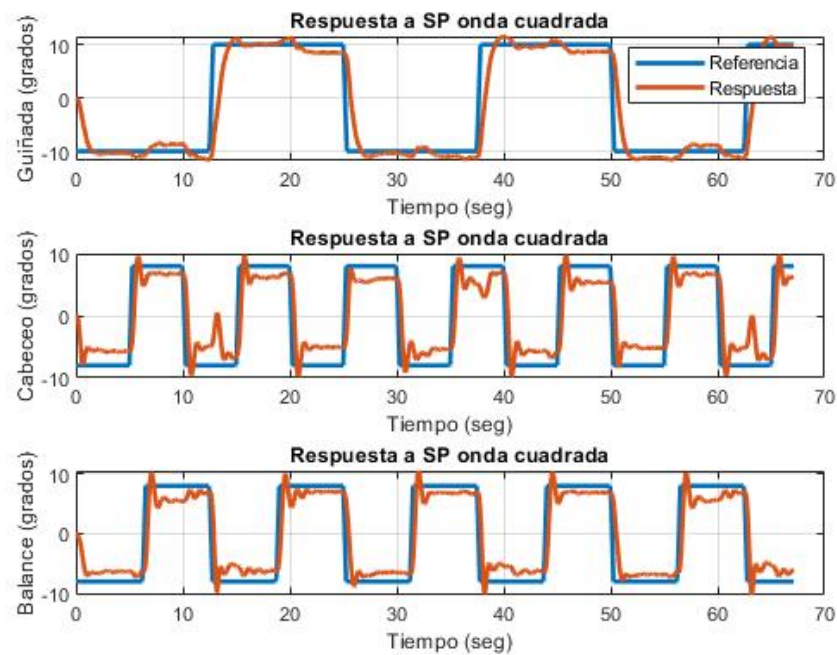


Figura 7: Respuesta del sistema de tensión aplicada a los motores hover

Como puede apreciarse mirando las gráficas, los resultados obtenidos son casi idénticos, existiendo pequeñas variaciones que son producto del ruido en la señal.

2. Ejercicio 9: Filtrado complementario

2.1. Introducción

En el siguiente informe se van a recopilar la novena práctica de la asignatura, mediante el uso de los softwares *Matlab* y *Simulink*.

Durante esta práctica se tratarán las diferentes formas de filtrado complementario tanto en dominio del tiempo como en discreto. En la figura 8 se observan donde se encuentran los filtros de cada uno de los apartados.

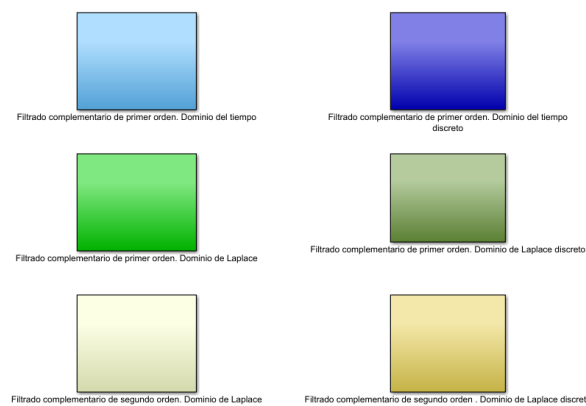


Figura 8: Diagrama de bloques Filtrado complementario orden 1 en el dominio temporal

2.2. Filtrado complementario de primer orden. *Dominio temporal*

En primer lugar se implementará un filtrado complementario de primer orden (dominio temporal). A continuación se presenta el filtro en *Simulink*.

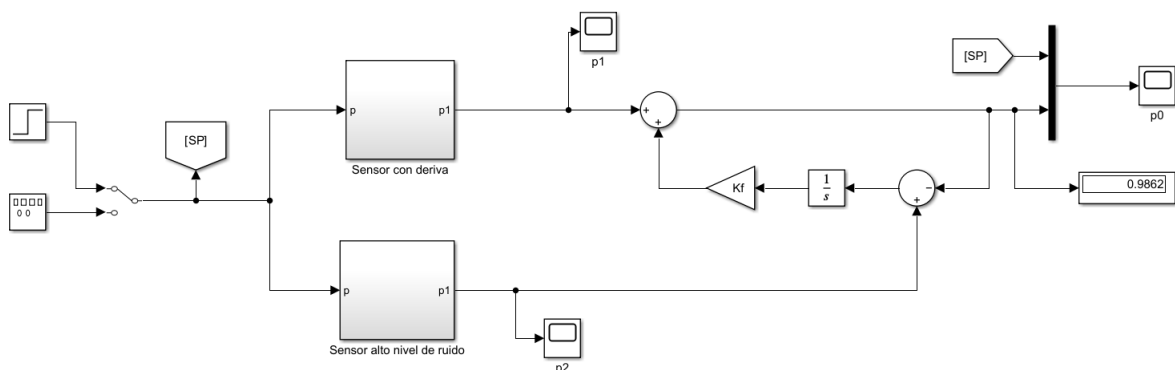


Figura 9: Diagrama de bloques Filtrado complementario orden 1 en el dominio temporal

Se calcula la ganancia del filtro de la siguiente forma:

$$K_f = \frac{1}{\tau_f} = 20$$

Posteriormente, se representan la respuesta del filtro washout (p_1) y del filtro paso-bajo (p_2) en la figura 10 por separado. Cabe destacar que estos filtros tienen una entrada escalón unidad y la duración de tiempo de simulación es de 30s.

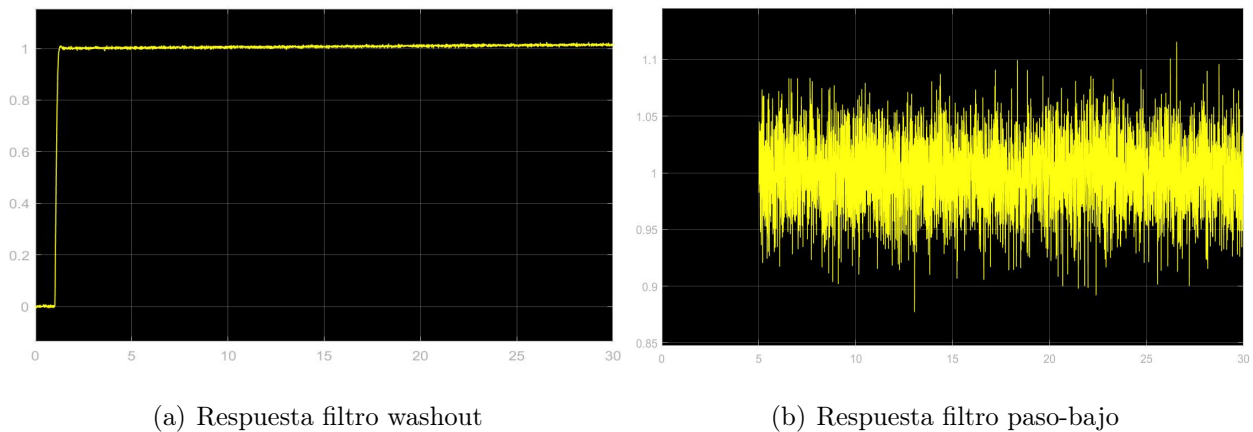


Figura 10: Respuesta de los filtros

Después de observar las respuestas por separado, se realiza la simulación del filtrado complementario (combinación de ambos (p_0))

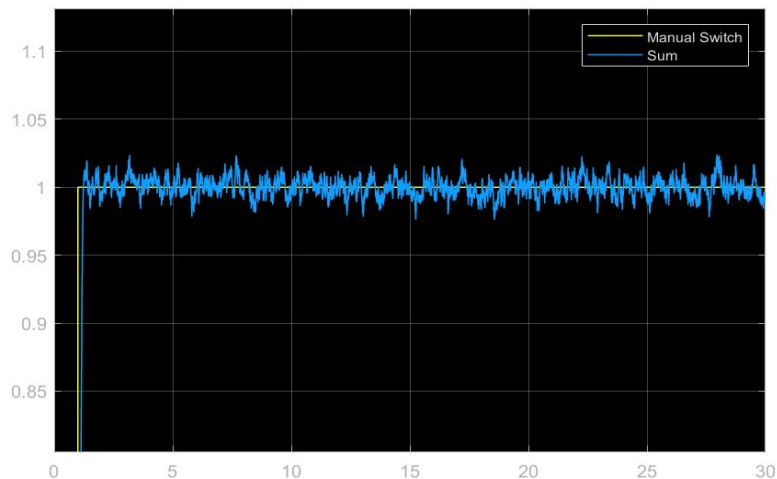
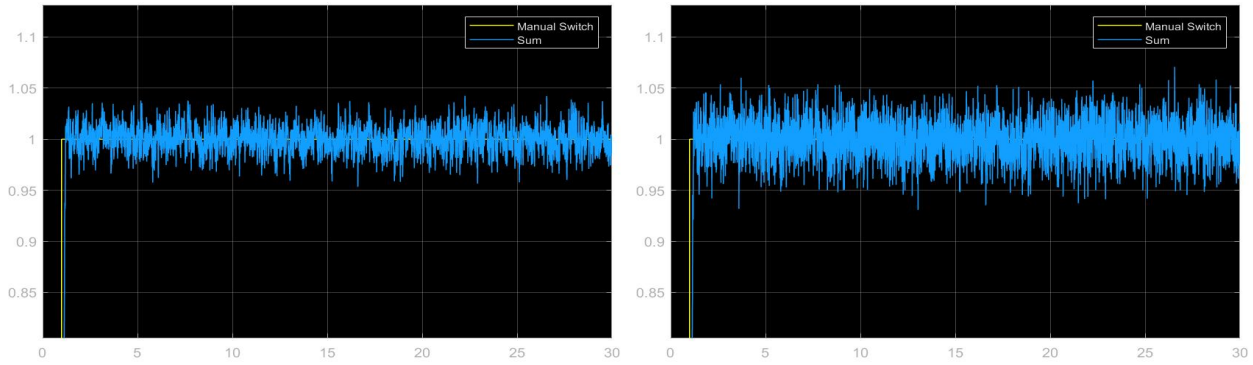


Figura 11: Respuesta del sistema con filtrado complementario

Como se aprecia en la figura 11, al combinar ambos filtros la respuesta del sistema tiene menos ruido y la señal que deriva es mucho menor.

Si se modifican los valores de la ganancia para el filtro complementario, se obtiene:



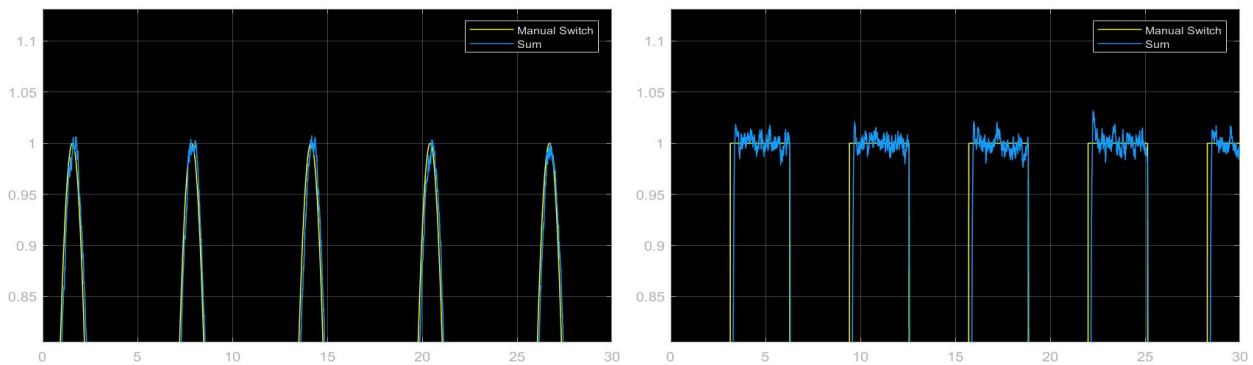
(a) Respuesta p_0 para $K_f = \frac{1}{\tau_f} \cdot 10$

(b) Respuesta p_0 para $K_f = \frac{1}{\tau_f} \cdot 25$

Figura 12: Respuesta del filtrado complementario (p_0) en función de K

Las respuestas se muestran en la figura 11, que se comentarán posteriormente tras realizar el filtro de Laplace, para poder comparar resultados.

Finalmente, utilizando un Signal Generator en Simulink, se introducirán diferentes señales de entrada para observar la respuesta del sistema.



(a) Respuesta ante una entrada Sine

(b) Respuesta ante una entrada Square

Figura 13: Respuestas del filtrado complementario ante SP de 1s de amplitud

Como se puede observar, el sistema responde adecuadamente a diferentes formas de entrada. Si comparamos ambas, la entrada Square presenta una respuesta del filtro con más ruido que la entrada Sawtooth.

2.3. Filtrado complementario de primer orden. *Dominio de Laplace*

Para realizar este filtro complementario de primer orden, se utilizará el dominio de Laplace. Al igual que en el apartado anterior, se implementará en *Simulink*.

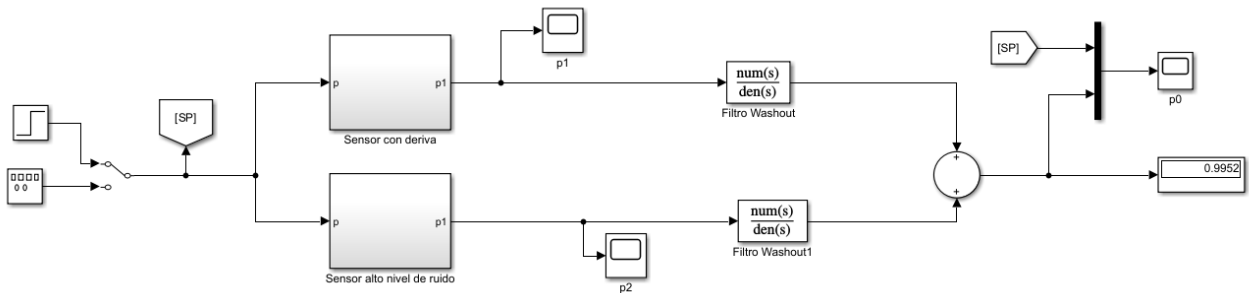


Figura 14: Diagrama filtrado complementario de primer orden en dominio de Laplace

A continuación se representa la respuesta del filtro complementario en el dominio de Laplace (Figura 15).

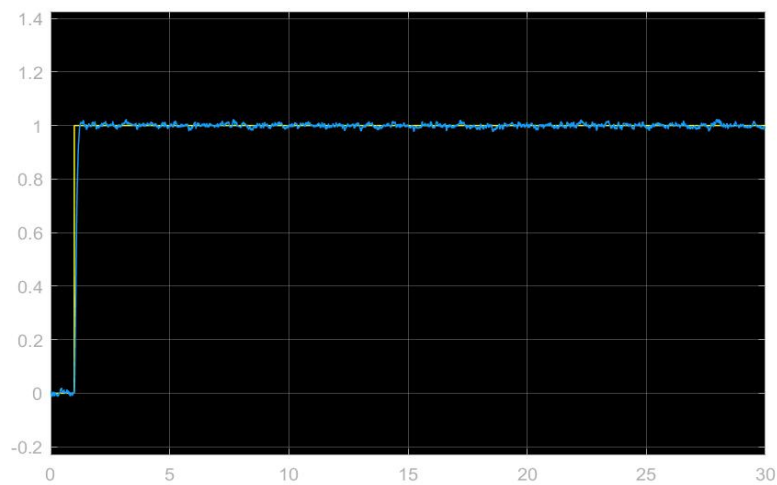
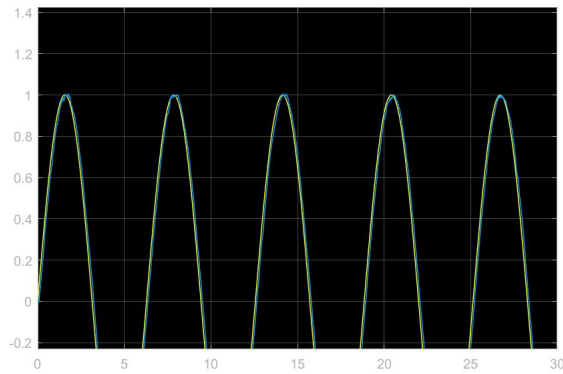
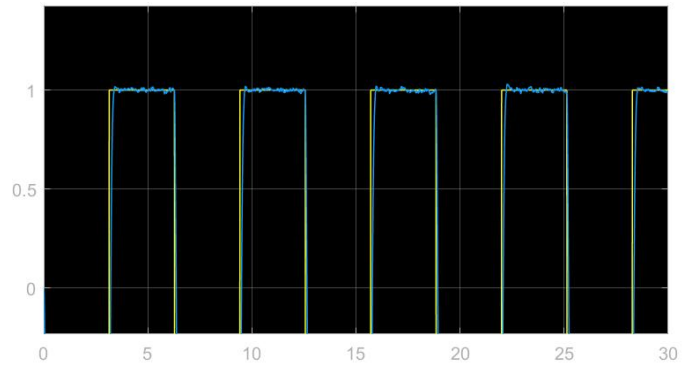


Figura 15: Respuesta del sistema con filtrado complementario

Para poder llegar a unas conclusiones sobre el filtro, se comparará la resultante de la combinación de las señales obtenidas por los dos sensores probando diferentes tipos de señales mediante el generador de señales, mostrándose en la figura 16.



(a) Respuesta ante una entrada Sine

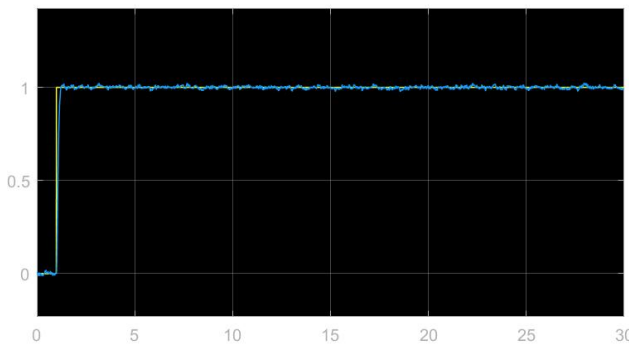
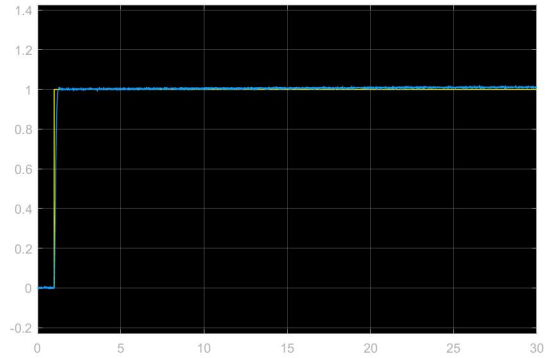
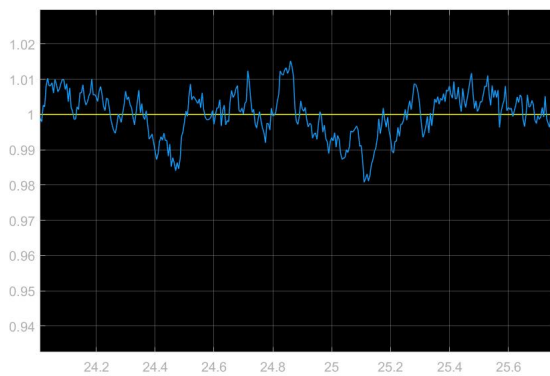
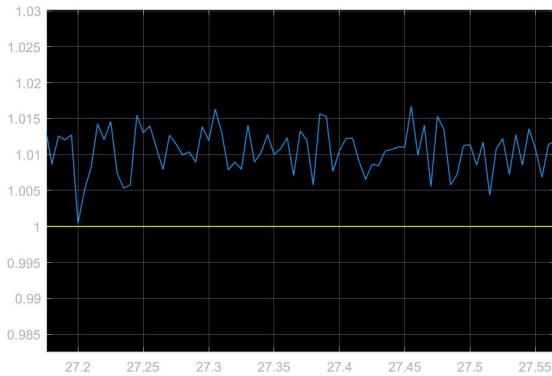


(b) Respuesta ante una entrada Square

Figura 16: Respuestas del filtrado complementario ante SP de 1s de amplitud

A continuación se compara la figura 13 con la figura 16. Utilizando un filtro complementario de primer orden se obtiene mejor resultado en el dominio de Laplace que en el dominio temporal. Si se observa la parte superior de las señales, se observa como utilizando Laplace la señal del filtro es mas ajustada a la señal de entrada que en el dominio temporal. No obstante, ambos dominios están más ajustados con tipos de señales sinodales y cuadradas que con una señal única de escalón.

Se va a mostrar una comparativa de la resultante de los sensores para distintos valores de la constante de tiempo asociada a los filtros (τ_f).

(a) Resultado para $\tau_f = 0,05$ (b) Resultado para $\tau_f = 50$ (c) Resultado para $\tau_f = 0,05$ ampliado(d) Resultado para $\tau_f = 50$ ampliadoFigura 17: Resultados tras el filtrado complementario para distintos valores de τ_f

Al ser la constante inversamente proporcional con la frecuencia es necesario buscar un valor óptimo, ya que para constantes de tiempo bajas la deriva se filtraría mucho mejor que el ruido. A medida que se incrementa el valor de la ganancia del filtro, el ruido de la respuesta obtenida en el filtrado complementario es mayor. Además la respuesta empieza a divergir de la señal deseada.

Representando el diagrama de Bode para $\tau_f = 0,05$ ya que se considera que es una respuesta aceptable.

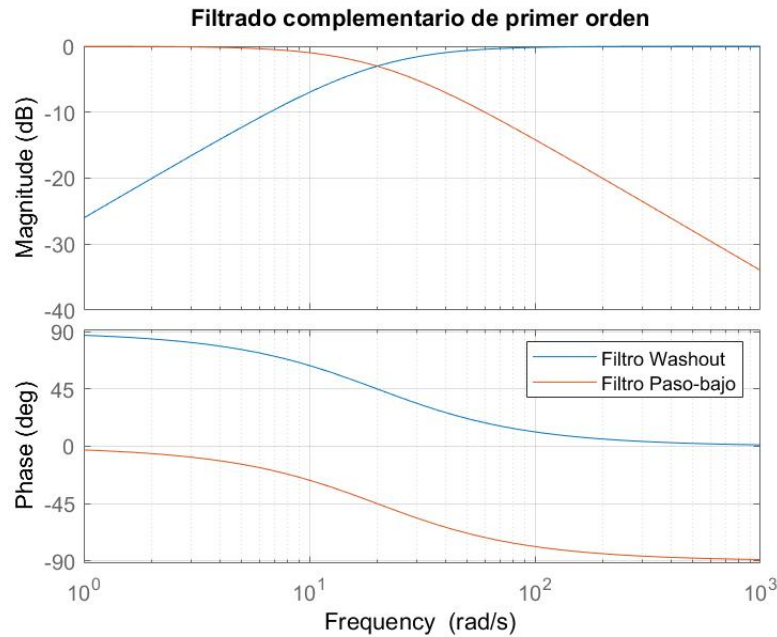


Figura 18: Resultados tras el filtrado complementario

Evidentemente, el filtro de washout atenúa las frecuencias bajas, mientras que el filtro paso-bajo atenúa las frecuencias altas. Como se ha ido probando durante la práctica.

2.4. Filtrado complementario de segundo orden. *Dominio de Laplace*

En este apartado se trata el mismo filtrado complementario pero de segundo orden. El diagrama será el siguiente:

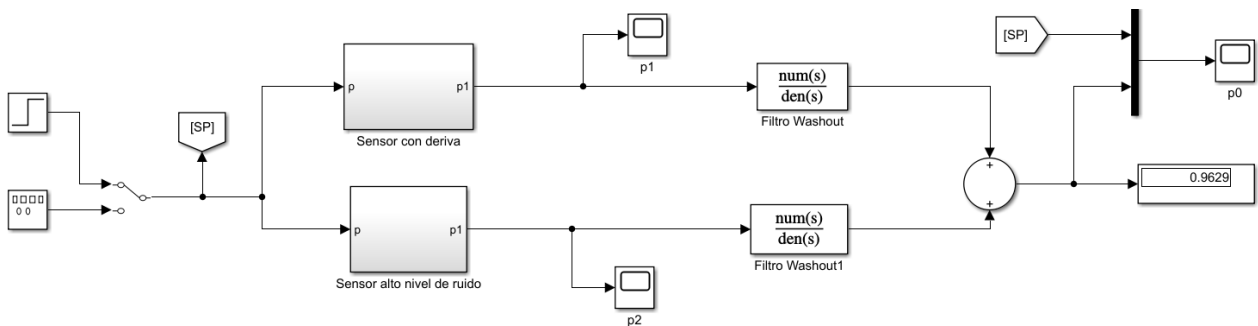


Figura 19: Diagrama para un filtrado complementario de segundo orden en dominio de Laplace

Ahora se va a realizar el filtrado para diferentes valores de frecuencia de corte (ω_{cf}) y coeficiente de amortiguamiento (ξ_f). Se representa en la siguiente figura.

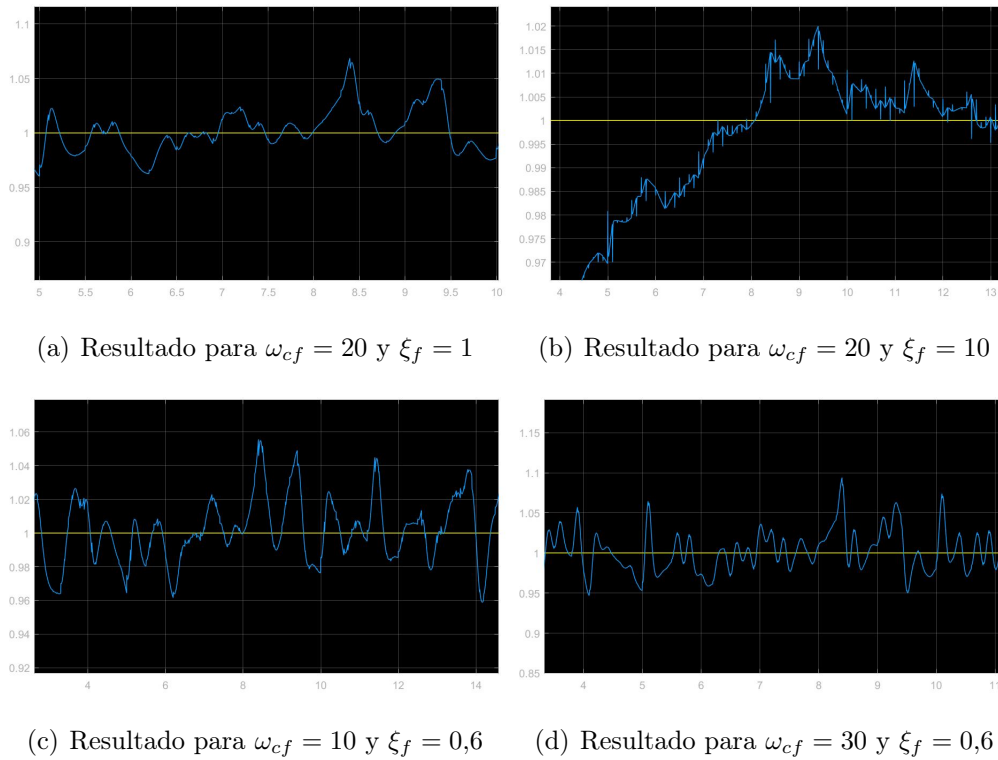


Figura 20: Resultados tras el filtrado complementario para distintos valores de ω_{cf} y ξ_f

A continuación, se presenta el diagrama de Bode para el sistema de segundo orden.

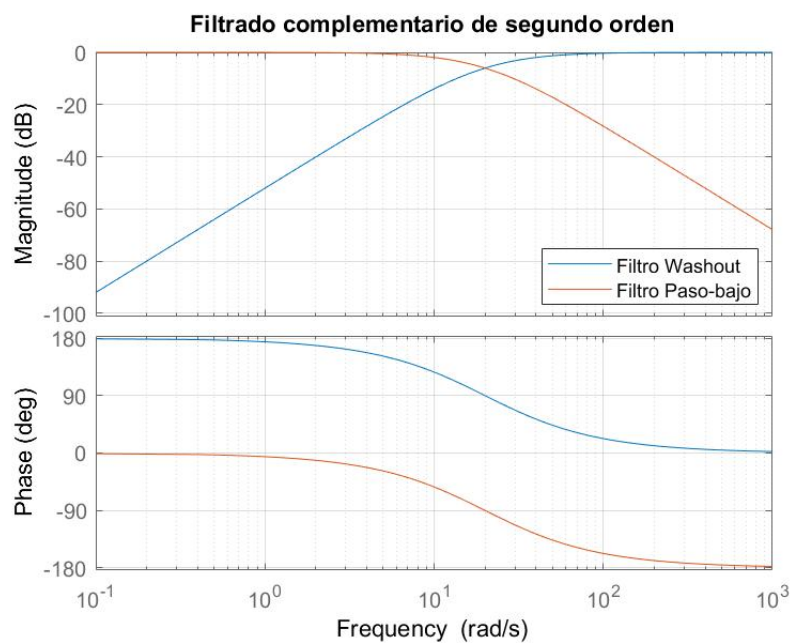


Figura 21: Diagrama de Bode indicando en azul el filtro paso-alto y en verde el filtro paso-bajo

En la figura 21 se observa que el filtro paso-alto o filtro Washout atenúa a frecuencias bajas, mientras que el filtro paso-bajo se atenúa a frecuencias altas.

2.5. Filtrado complementario discreto

Por último, se analizará el filtrado complementario en tiempo discreto. Para realizar esta parte de la práctica se utilizarán los diagramas utilizados en apartados anteriores y se modificaran para el tiempo discreto.

El tiempo de muestreo, para los convertidores A/D y el sistema de computo, será de la siguiente forma:

$$T_m = \frac{\tau_f}{100} \quad (1)$$

Cabe destacar que también se ha cambiado el paso de integración para la parte discreta.

Para el filtrado complementario de primer orden en el dominio temporal, se ha cambiado el integrador continuo por uno en tiempo discreto. El resultado se muestra en la Figura 22.

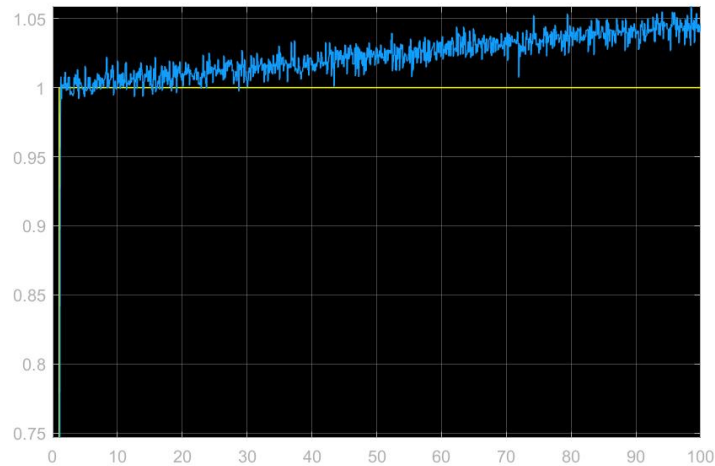


Figura 22: Resultado Dominio temporal (primer orden) en tiempo discreto

Para el dominio de Laplace, ya bien sea filtro de primer o segundo orden, se necesita convertir la función transferencia de tiempo continuo a tiempo discreto. El resultado para el filtro de primer orden se observa en la Figura 23.

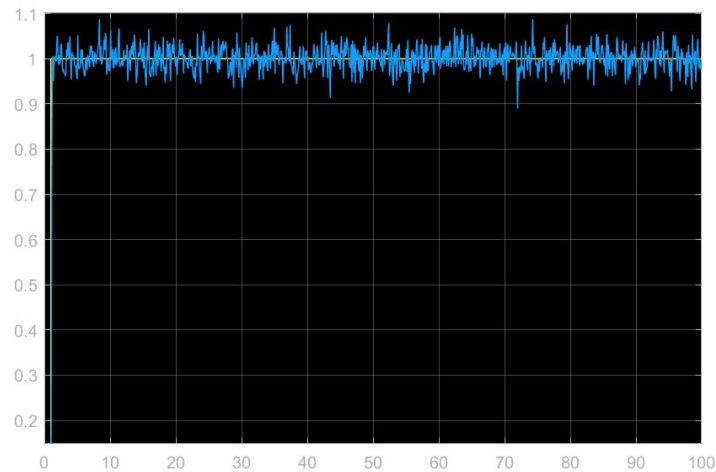


Figura 23: Resultado Dominio de Laplace (primer orden) en tiempo discreto

Al igual que en tiempo continuo, se obtienen mejores resultados en el dominio de Laplace (en cuanto a filtros de primer orden).

Finalmente, se muestra la respuesta del filtrado complementario en tiempo discreto para el dominio de Laplace de segundo orden (ver figura 24).

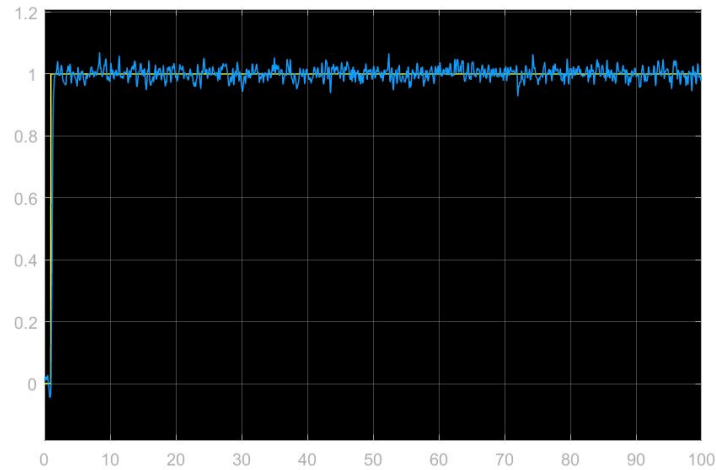


Figura 24: Resultado Dominio de Laplace (segundo orden) en tiempo discreto

Si se comparan los filtros en tiempo discreto y continuo, no se aprecia grandes diferencias puesto que se han obtenido respuestas similares para cada uno de los dominios y ordenes correspondientes.

3. Ejercicio 10: Control del helicóptero

3.1. Introducción

El objetivo de esta práctica es diseñar un controlador de actitud para un helicóptero, (que tendrá dos grados de libertad) mediante el uso de un controlador Feed Forward, mediante la técnica de diseño LQR con acción integral.

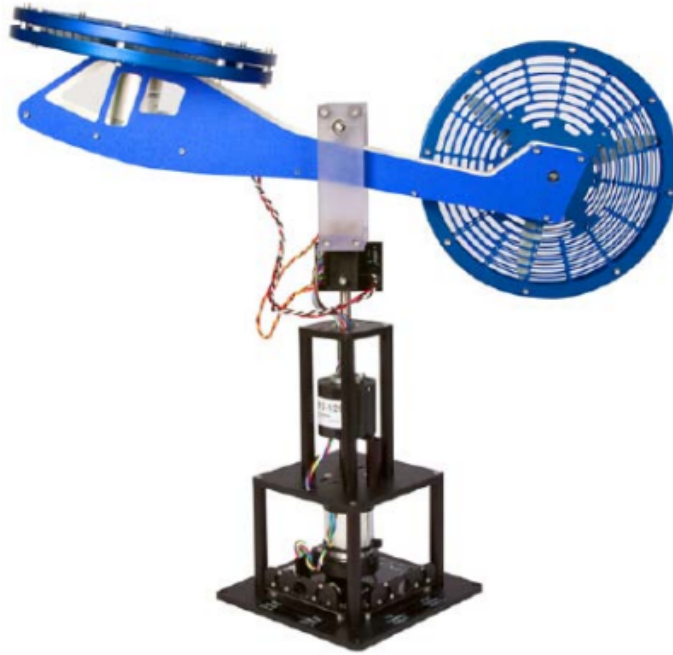


Figura 25: Modelo de helicóptero utilizado

3.2. Representación en el espacio de estados y modelo lineal

Las ecuaciones que definen el modelo no lineal del sistema son:

$$J_p \ddot{\theta} = \tau_{Bp} + \tau u_p = -B_p \dot{\theta} + \tau u_p \quad (2)$$

$$J_y \ddot{\psi} = \tau_{By} + \tau u_y = -B_y \dot{\psi} + \tau u_y \quad (3)$$

Este sistema de ecuaciones se puede transformar en el siguiente:

$$J_{Tp}\ddot{\theta} + B_p\dot{\theta} = K_{pp}u_p + K_{py}u_y \quad (4)$$

$$J_{Ty}\ddot{\psi} + B_y\dot{\psi} = K_{yp}u_p + K_{yy}u_y \quad (5)$$

Las matrices A, B, C y D del sistema vendrán dadas por las siguientes expresiones:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-B_p}{J_{Tp}} & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{-B_y}{J_{Ty}} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{Tp}} & \frac{K_{py}}{J_{Tp}} \\ \frac{K_{yp}}{J_{Ty}} & \frac{K_{yy}}{J_{Ty}} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

3.3. Diagrama de simulación

El simulink utilizado para esta práctica es el siguiente:

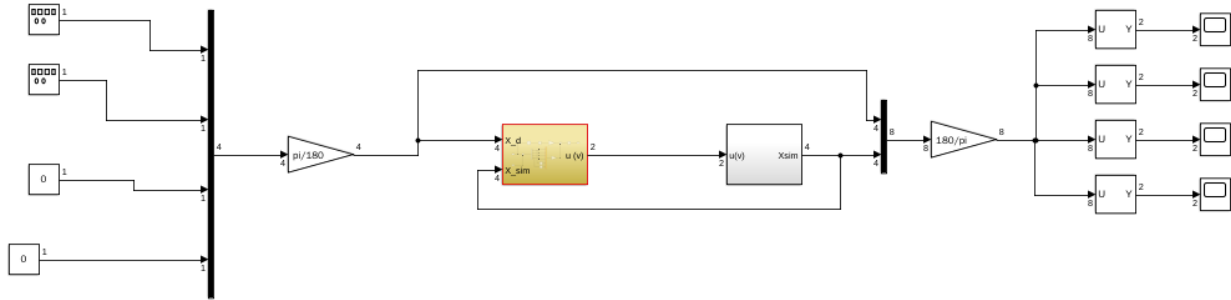


Figura 26: Diagrama de simulación utilizado

El código de la práctica fue proporcionado por el profesor, quedando a cargo de los alumnos el cálculo de las matrices de estado (tal y como se muestra en el apartado anterior) y el valor de las matrices Q_c y R_c .

Las matrices de estado se obtienen simplemente sustituyendo los valores de las constantes que se muestran en dichas matrices, valores que están proporcionados por el profesor, y que se muestran en la siguiente figura:

Symbol	Description	Value	Unit
K_{pp}	Thrust torque constant of yaw motor/propeller	0.204	N.m/V
K_{yy}	Thrust torque constant of yaw axis from yaw motor/propeller	0.072	N.m/V
K_{py}	Thrust torque constant acting on pitch axis from yaw motor/propeller	0.0068	N.m/V
K_{yp}	Thrust torque constant acting on yaw axis from pitch motor/propeller	0.0219	N.m/V
$B_{eq,p}$	Equivalent viscous damping about pitch axis	0.800	N/V
$B_{eq,y}$	Equivalent viscous damping about yaw axis	0.318	N/V
m_{heli}	Total moving mass of the helicopter (body, two propeller assemblies, etc.)	1.3872	kg
l_{cm}	Center of mass length along helicopter body from pitch axis	0.186	m
$J_{eq,p}$	Total moment of inertia about pitch axis	0.0384	kg.m ²
$J_{eq,y}$	Total moment of inertia about yaw axis.	0.0432	kg.m ²

Figura 27: Diagrama de simulación utilizado

3.4. Resultados obtenidos

Los resultados obtenidos se muestran en las siguientes gráficas.

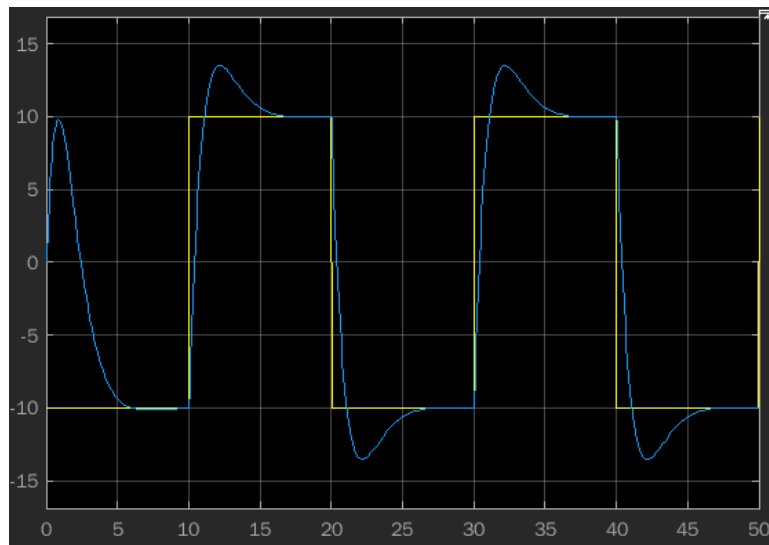


Figura 28: Resultados de cabeceo

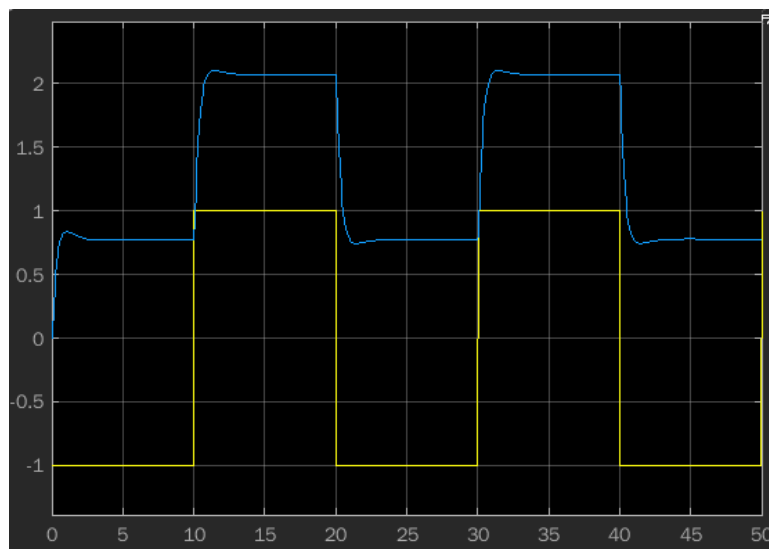


Figura 29: Resultados de guiñada

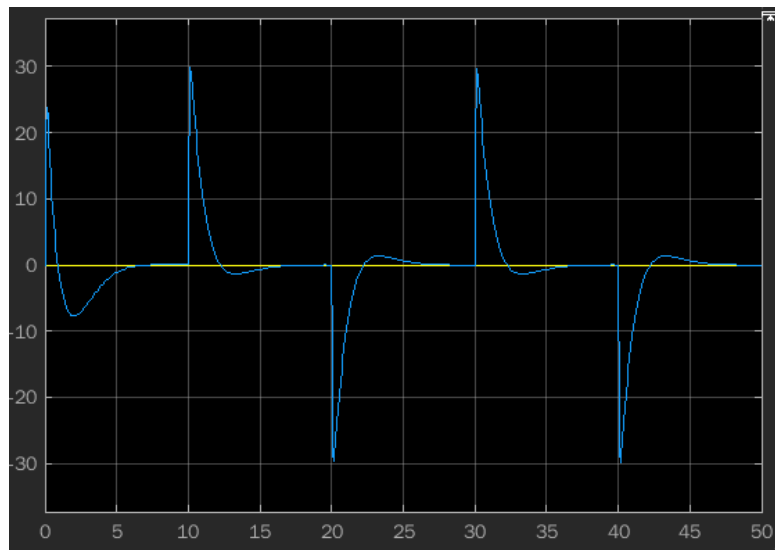


Figura 30: Resultados de velocidad de cabeceo

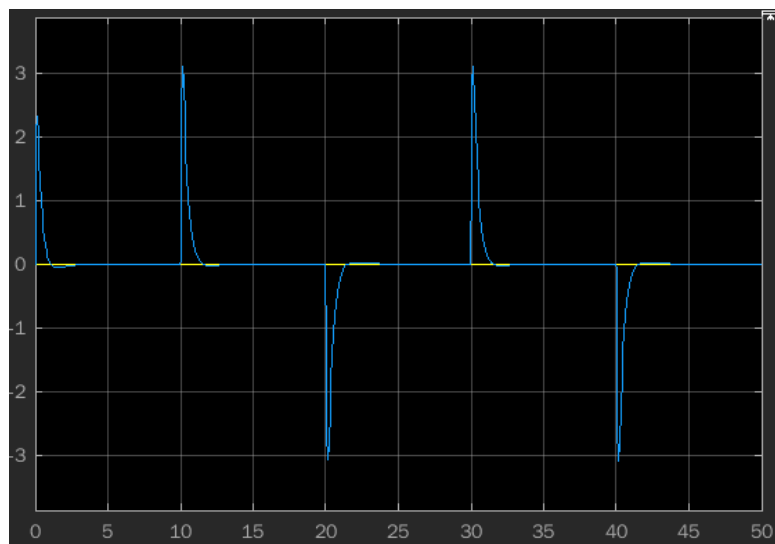


Figura 31: Resultados de velocidad de guiñada

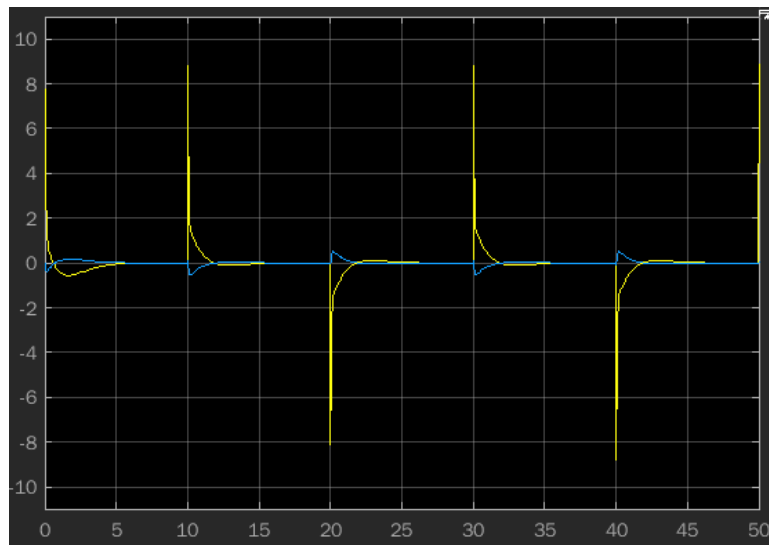


Figura 32: Resultados de la tensión aplicada a los rotores

Como se puede ver en las gráficas, estos resultados concuerdan con lo que debería obtenerse de acuerdo a lo visto en clase.

3.5. Resultados mediante script de matlab

Mediante el archivo de datos obtenido en clase, se puede obtener con matlab las respuestas obtenidas:

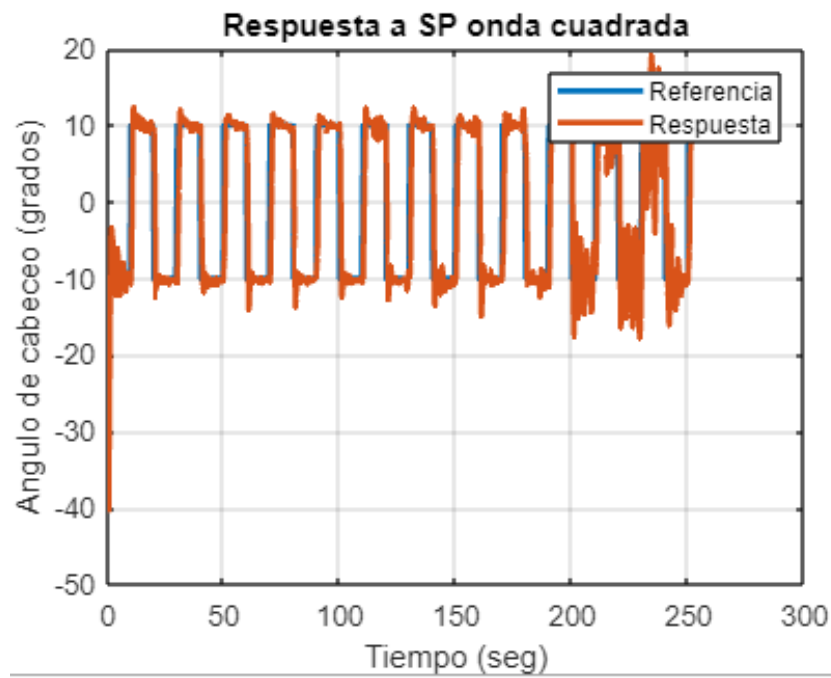


Figura 33: Ángulo de cabeceo

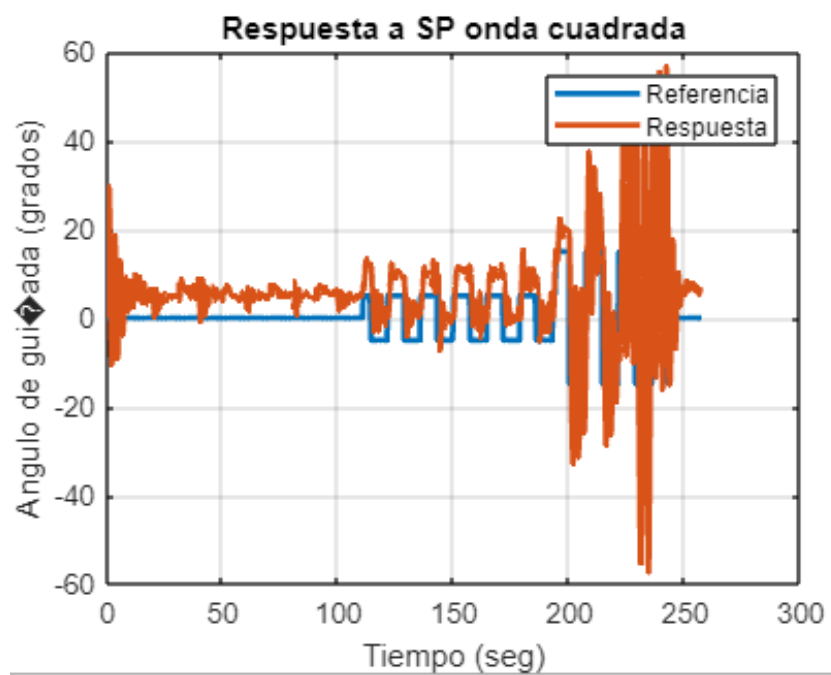


Figura 34: Ángulo de guiñada

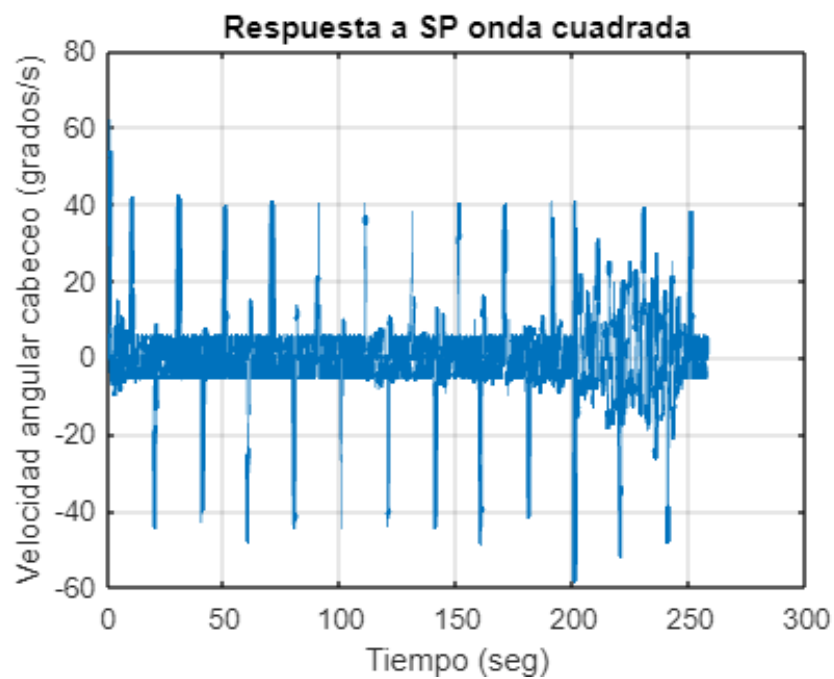


Figura 35: Velocidad angular de cabeceo

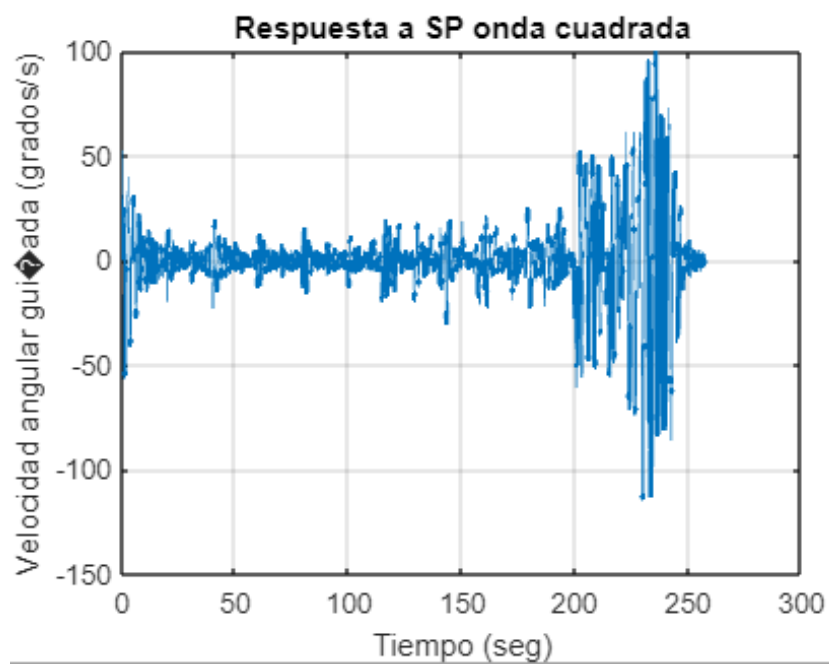


Figura 36: Velocidad angular de guiñada

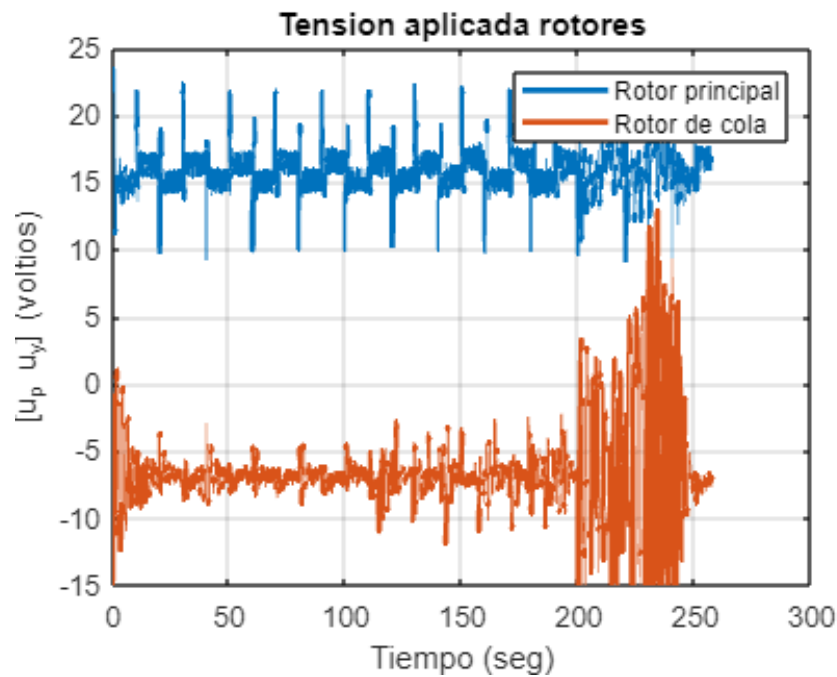


Figura 37: Tensión aplicada a los rotores

A pesar del ruido que presentan los resultados obtenidos con matlab, se puede observar una muy buena aproximación y similitud en ambos resultados, por lo que se puede dar por finalizado el proceso de diseño del controlador.