



# SERVOMECANISMOS

Modelado, Identificación y Control del  
sistema Aeropéndulo

21-06-2016

Profesor:

Dr. Etchepareborda Andrés

Grupo de trabajo:

Paesani Matías

Sosa Visconti Ariel

## Índice

<b>Introducción</b> .....	3
<b>Objetivos</b> .....	4
<b>1 Modelado e identificación del sistema aeropéndulo</b> .....	5
<b>1.1 Descripción del sistema aeropéndulo</b> .....	5
<b>1.2 Diagrama de cuerpo libre y ecuaciones del sistema</b> .....	6
<b>1.3 Identificación del sistema de propulsión</b> .....	8
<b>1.4 Identificación del sistema péndulo</b> .....	14
<b>1.5 Acoplamiento del sistema de propulsión con el sistema péndulo</b> .....	18
<b>2 Linealización del sistema aeropéndulo y desarrollo del primer controlador</b> .....	21
<b>2.1 Linealización del sistema aeropéndulo</b> .....	21
<b>2.2 Desarrollo del primer controlador</b> .....	25
<b>3 Filtro de Kalman Extendido (EKF)</b> .....	30
<b>3.1 Cálculo e implementación del EKF</b> .....	30
<b>4 Diseño e implementación de distintas técnicas de control</b> .....	35
<b>4.1 Controlador PD en la realimentación</b> .....	35
<b>4.2 Controlador integral por variables de estado</b> .....	37
<b>4.3 Controlador integral por variables de estado con modelo de referencia</b> .....	42
<b>4.4 Controlador por linealización exacta</b> .....	45
<b>4.5 Controlador por linealización exacta y control integral con modelo de referencia</b> .....	50
<b>Conclusiones</b> .....	53
<b>Bibliografía</b> .....	55
<b>Anexo 1</b> .....	57
<b>Anexo 2</b> .....	59
<b>Anexo 3</b> .....	61
<b>Anexo 4</b> .....	69

## Introducción

Con el fin de poner en práctica los conocimientos adquiridos en los sucesivos cursos de Teoría de Control realizados por el grupo a lo largo de la carrera de Ingeniería Electromecánica con orientación en Automatización Industrial, se llevó a cabo, dentro del marco de la asignatura Servomecanismos, la Práctica Profesional Supervisada (PPS) denominada “Modelado, Identificación y Control de un Aeropéndulo”.

El aeropéndulo es un péndulo en cuyo extremo libre se encuentra instalado un motor de corriente continua brushless con una hélice fijada a su rotor, de modo que al girar esta ejerce una fuerza que se opone a la acción de la gravedad levantando así el extremo del péndulo de su posición de reposo.

El objetivo fue lograr el control de la posición final del sistema a lo largo de todo su rango de operación, desde su posición de reposo hasta la posición de no retorno en el punto opuesto. Teniendo en cuenta las no linealidades de la planta, se realizó en primer lugar un modelado e identificación del aeropéndulo, para luego en base a una simulación del sistema poder diseñar y probar distintas alternativas de controladores.

Por su parte, también se implementó un estimador predictivo óptimo, más precisamente un Filtro de Kalman Extendido, con el fin de estimar las mediciones de las variables del sistema de modo de obtener una estimación con menor ruido y más precisa, pudiéndose lograr un control más suave.

Desarrollado este filtro, se implementaron distintas alternativas de control mediante las técnicas de modelo externo (control clásico) y modelo interno (control por variable de estado), así como también control por modelo de referencia y por linealización exacta.

## Objetivos

Los objetivos de esta práctica pueden especificarse en tres:

**Modelado e identificación del sistema aeropéndulo**, mediante lo cual desarrollar una simulación que refleje el comportamiento real de la planta a lazo abierto y poder así diseñar y probar los distintos controladores diseñados. Implementar tanto una simulación que represente las no linealidades del sistema como una simulación del sistema linealizado alrededor de distintos puntos de operación.

**Implementación del Filtro de Kalman Extendido**, con el fin de lograr la mejor estimación de las variables medidas del sistema y reducir el ruido en la lectura de las mismas. Comparar luego el desempeño del sistema controlado tanto con las variables estimadas como con las mediciones.

**Diseño e implementación de diversos controladores**, empleando distintas técnicas de control, evaluando y comparando el desempeño de cada uno, utilizando como realimentación las variables estimadas. En particular, control clásico (técnicas de modelo externo), control por variables de estado (técnicas de modelo interno), control integral con modelo de referencia y control por linealización exacta.

# 1 Modelado e identificación del sistema aeropéndulo

## 1.1 Descripción del sistema aeropéndulo

La planta consta de dos sistemas que fueron modelados e identificados por separado para luego ser acoplados uno al otro:

Por un lado, el sistema de propulsión, solidario al extremo libre del péndulo. Definido por las componentes hélice, motor y driver, ejerce la fuerza opuesta a la acción de la gravedad y es el sistema sobre el cual accede la acción de control que lleva la planta de una posición a otra definida por el usuario. La hélice, fabricada en plástico y con un radio aproximado de 125 mm, se encuentra fijada al rotor del motor de corriente continua (del tipo brushless), comandado por un driver.

Por otro lado, el sistema péndulo, que consta de una varilla rígida acoplada a un eje horizontal apoyado sobre rodamientos. En uno de los extremos se encuentra un sistema de engranajes que se conecta al encoder utilizado para ver la posición angular del péndulo, siendo la relación entre estos de 90/50 (o 1.8), mientras que el encoder ejecuta 4000 cuentas por vuelta.

La comunicación entre el sistema aeropéndulo y la PC, desde la cual se ejecutaron los algoritmos de control, es por medio de un PIC con un tiempo de muestreo de 0.004 s, es decir, 250 ciclos por segundo.

Algo importante a resaltar, es que se estableció un rango de cuentas de PWM dentro del cual alimentar el motor. Esto se debe a la limitación de potencia propia de la fuente que energiza el sistema, estableciéndose un máximo de 2500 cuentas, mientras que el mínimo, dado por la cantidad necesarias para sobrepasar la franja muerta, se fijó en 1000.

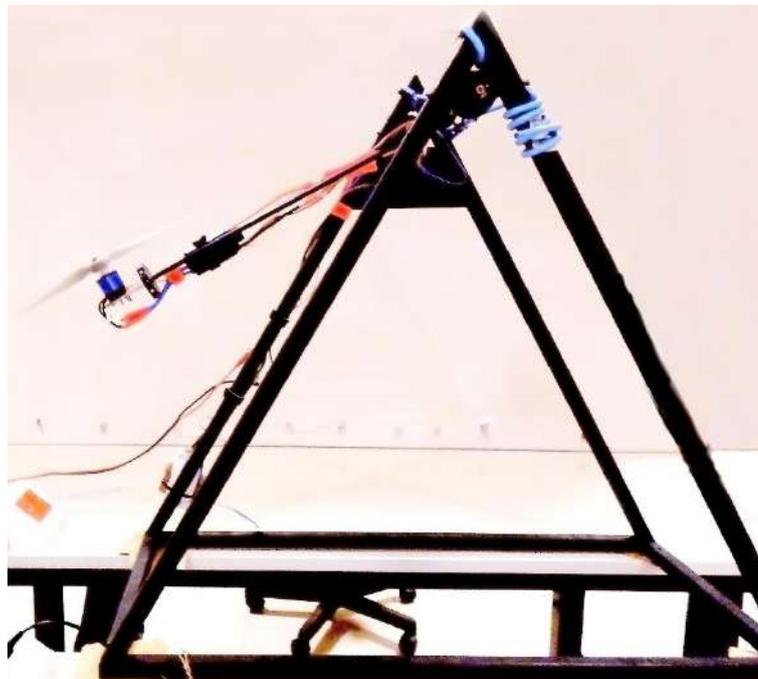


Figura 1.1.1. Aeropéndulo en funcionamiento.

## 1.2 Diagrama de cuerpo libre y ecuaciones del sistema

Para hallar un modelo del sistema, se realizó el diagrama de cuerpo libre del péndulo junto al sistema de propulsión, a partir del cual se derivaron las principales ecuaciones físicas que rigen al aeropéndulo.

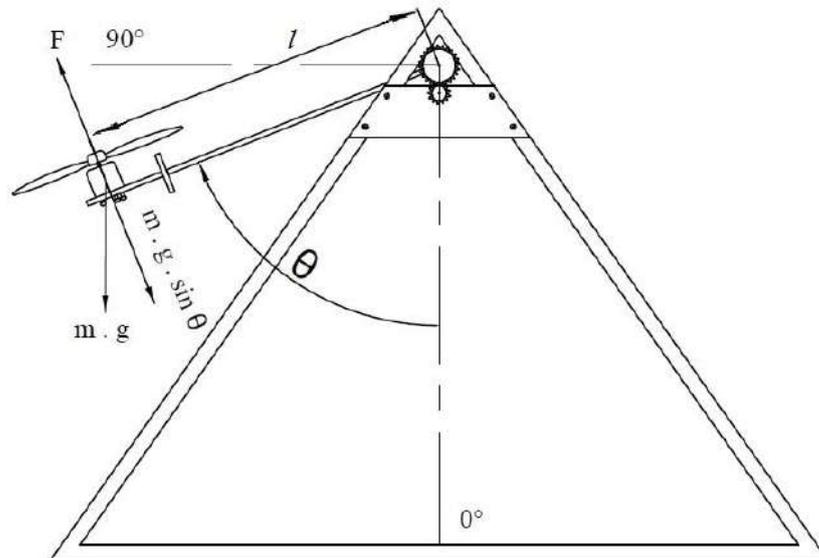


Figura 1.2.1. Diagrama de cuerpo libre del aeropéndulo.

Ecuación de momentos respecto al eje:

$$\sum M = F \cdot l - m \cdot g \cdot \sin(\theta) \cdot l = J_p \cdot \dot{\omega} + f(\omega) \quad (1.2.1)$$

Donde:

$F$  = Fuerza de empuje.

$l$  = Longitud del péndulo.

$m$  = Masa del péndulo.

$g$  = Aceleración de la gravedad.

$\theta$  = Posición angular del péndulo.

$\omega$  = Velocidad angular del péndulo.

$f(\omega)$  = Fricción viscosa.

$J_p$  = Momento de inercia del péndulo.

Ahora bien, la fuerza de empuje ejercida por el sistema de propulsión, al no poder medirse, fue modelada como una función dependiente de la velocidad de giro del motor:

$$F = h(\Omega) \quad (1.2.2)$$

Donde:

$\Omega$  = Velocidad del motor.

De las ecuaciones se desprendió una característica de suma importancia del sistema, y es que este se vuelve inestable al cruzar los  $90^\circ$ . Esto es debido a que, en posiciones por debajo de este punto, la componente del empuje que se opone a la acción de la gravedad debe aumentar si se pretende alcanzar posiciones en ángulos mayores. Pero por encima de  $90^\circ$ , la relación se invierte, por lo que para que el sistema se mantenga estable en ángulos superiores, la fuerza de empuje debe ser cada vez menor.

Por otro lado, debido a que el sentido de la fuerza de empuje es siempre el mismo, el sistema de propulsión no puede regresar la planta a ángulos menores a  $180^\circ$  una vez sobrepasado este límite. El rango de trabajo del sistema quedó así definido entre  $0^\circ$  y  $180^\circ$ , siendo la posición de reposo y el punto opuesto a esta respectivamente.

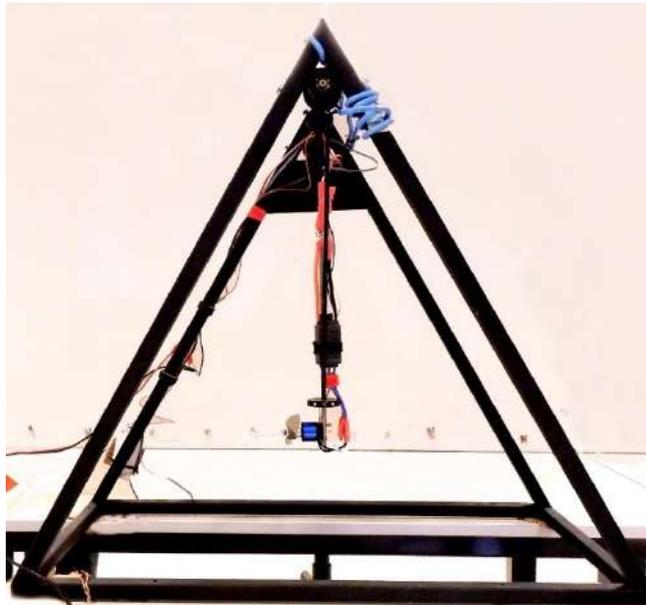


Figura 1.2.2. Aeropéndulo en posición de reposo en  $0^\circ$

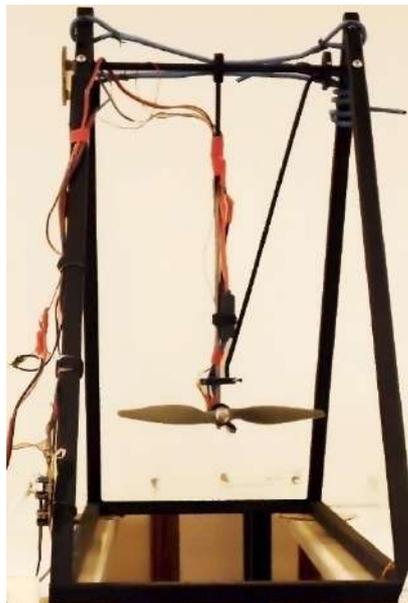


Figura 1.2.3. Vista frontal del aeropéndulo en reposo.

### 1.3 Identificación del sistema de propulsión

Como se denotó en la ecuación (1.2.2), se optó por modelar el sistema de propulsión como la fuerza de empuje en función de la velocidad del motor. Sin embargo, al no ser posible medir esta fuerza de manera directa, se realizó una medición indirecta empleando una secuencia de escalones que llevaron al sistema de una posición de equilibrio a otra. En cada uno de estos puntos, las ecuaciones del sistema se reducen a:

$$F = m \cdot g \cdot \sin(\theta) = a \cdot \sin(\theta) \quad (1.3.1)$$

Donde:

$a$  = constante que agrupa  $m \cdot g$ .

El ensayo realizado se ejecutó dentro del rango estable del sistema ( $0^\circ$  a  $90^\circ$ ), realizando sucesivos escalones de 50 cuentas de PWM, iniciando en 1000 y concluyendo en 1750, registrándose la posición angular del péndulo y la velocidad del motor. Teniendo la precaución de evitar el efecto suelo (ver Anexo 1), los datos obtenidos en este ensayo fueron:

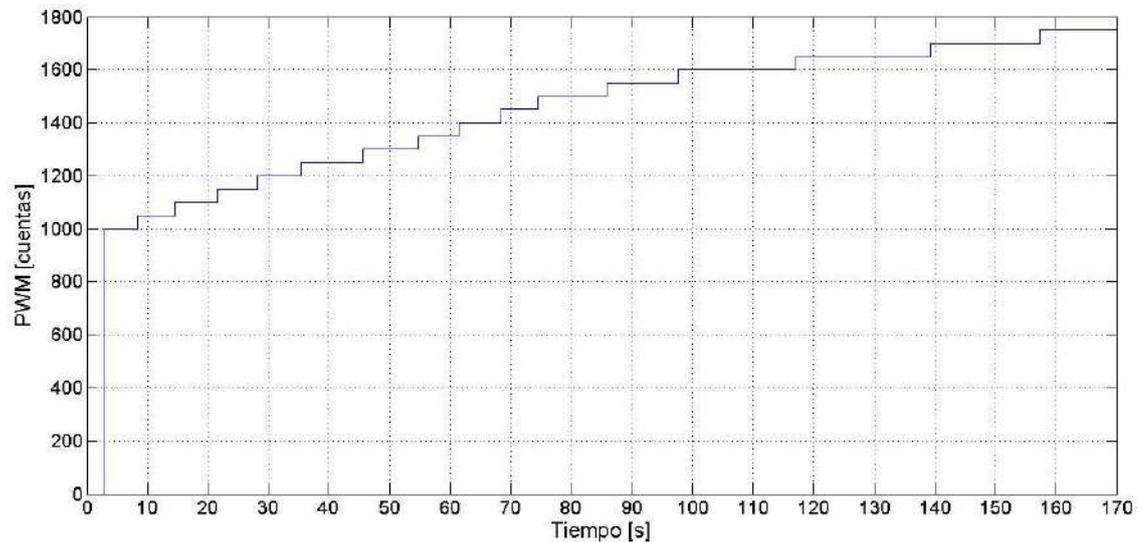


Figura 1.3.1. Secuencia de escalones de PWM en cuentas.

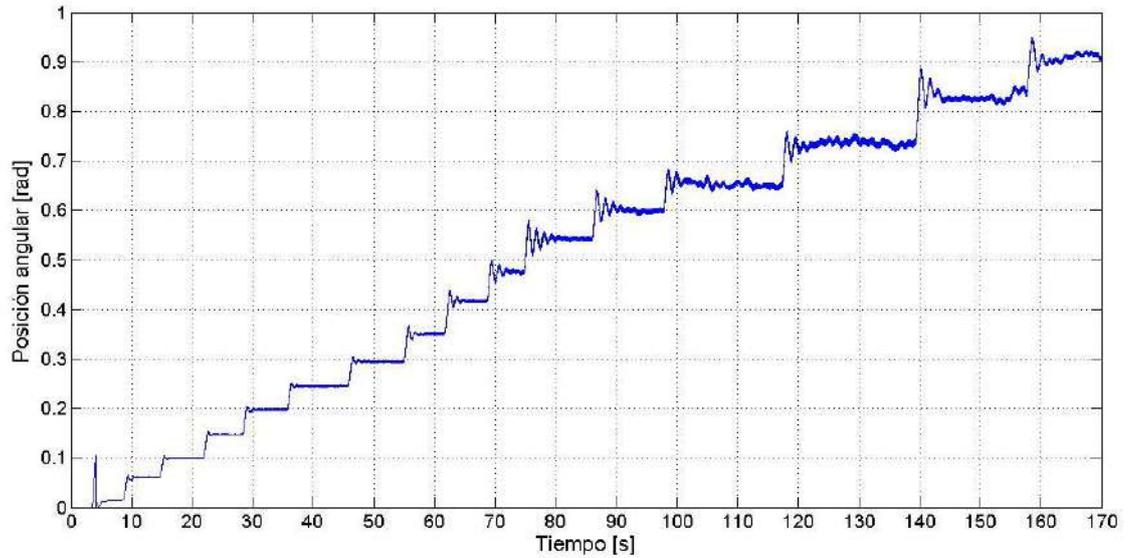


Figura 1.3.2. Posición angular del sistema.

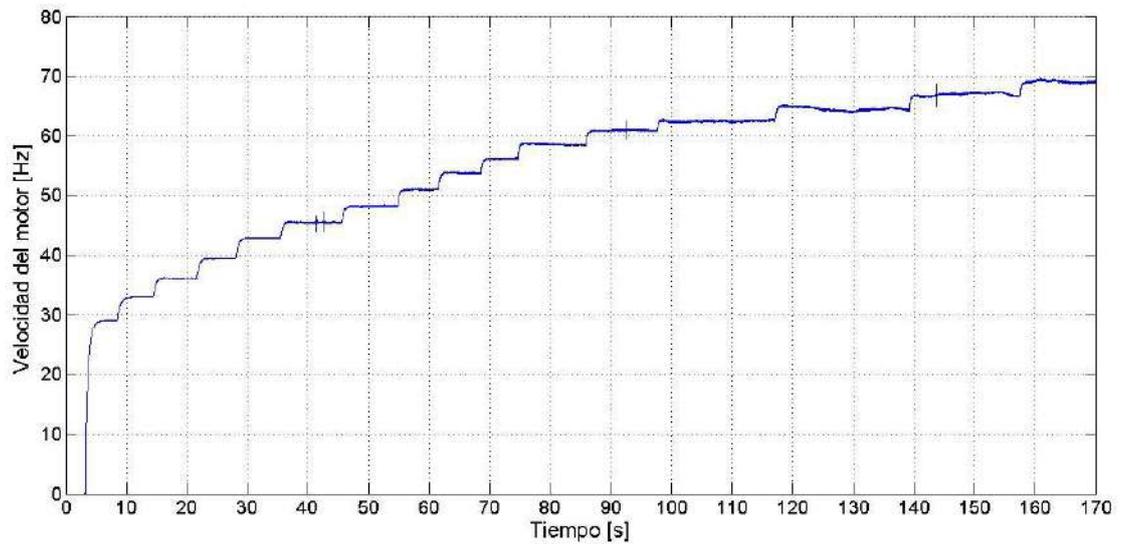


Figura 1.3.3. Velocidad del motor

Con estos datos se realizaron dos análisis: la relación velocidad del motor vs fuerza, y PWM vs fuerza. Usando la relación (1.3.1) pero dejando fuera de esta la constante  $\alpha$ , que luego fue calculada en la etapa de acoplamiento de ambos sistemas, la fuerza obtenida como señal de salida del modelo del sistema de propulsión no tiene unidades de tal, sino que, una vez acoplados ambos sistemas, se vuelve un empuje que genera un momento sobre el eje del péndulo.

Para la primera relación se tomaron los datos del motor en Hz (1/s) y la fuerza como  $\sin(\theta)$ , aproximados al valor estacionario en cada posición, y empleando el comando de *Matlab cftool* se ajustó una función cuadrática a los puntos por el método de mínimos cuadrados, presentándose dicho ajuste en la figura 1.3.4.

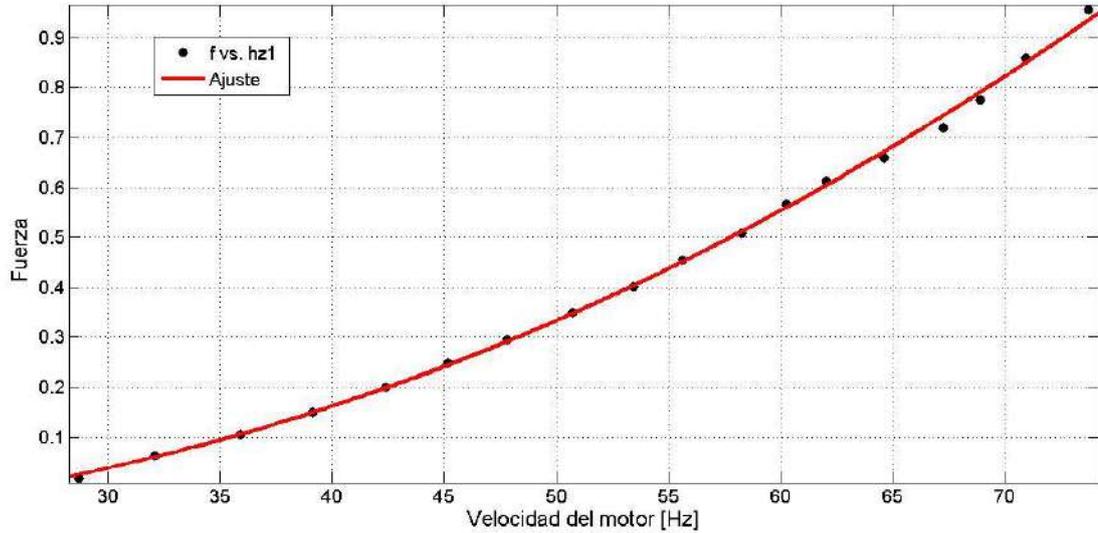


Figura 1.3.4. Ajuste Fuerza vs Velocidad del Motor.

La ecuación obtenida es:

$$f(x) = 0.242 \cdot 10^{-3} \cdot x^2 - 0.00458 \cdot x - 0.04257 \quad (1.3.2)$$

Donde:

$f(x)$  = Fuerza como  $\sin(\theta)$ .

$x$  = Velocidad del motor [Hz].

Para la segunda relación se tomaron nuevamente los datos de la fuerza aproximada como  $\sin(\theta)$  y esta vez la PWM en cuentas. Luego, se ajustó una función cuadrática a estos puntos, el cual se presenta en la siguiente figura:

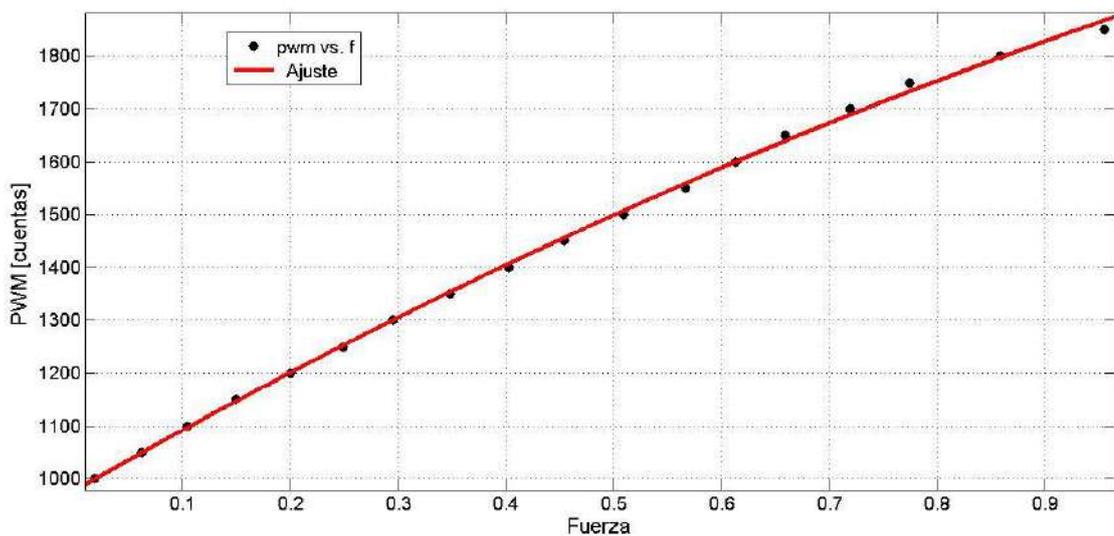


Figura 1.3.5. Ajuste PWM vs fuerza.

La ecuación obtenida es:

$$f(x) = -245 \cdot x^2 + 1165 \cdot x + 977.6 \quad (1.3.3)$$

Donde:

$f(x)$  = PWM en cuentas.

$x$  = Fuerza como  $\sin(\theta)$ .

Se realizó también el ajuste inverso para pasar unidades de PWM a fuerza de empuje y luego acoplar al sistema péndulo mediante la constante  $a$ . A continuación, dicho ajuste:

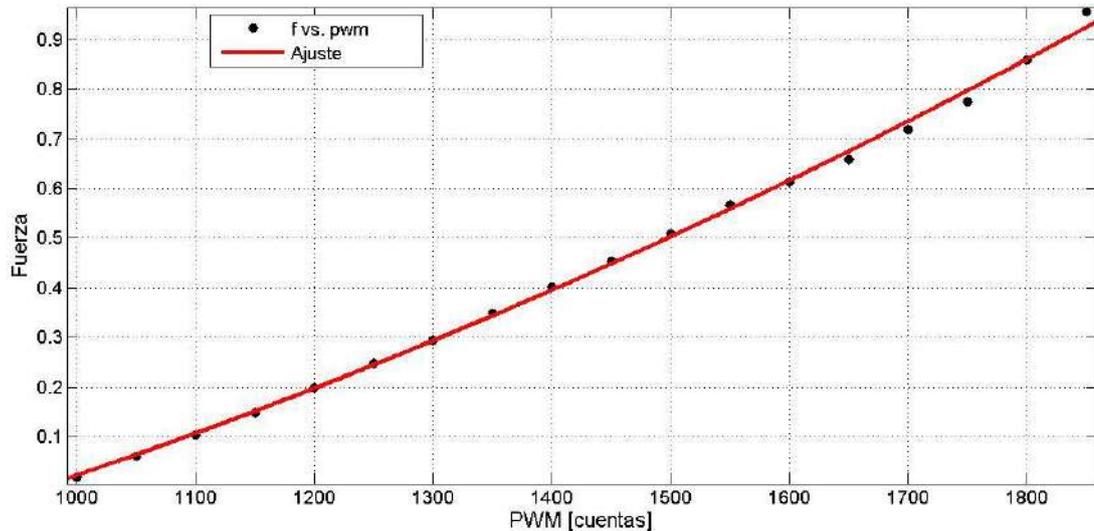


Figura 1.3.6. Ajuste fuerza vs PWM.

La ecuación obtenida es:

$$f(x) = 2.86 \cdot 10^{-7} x^2 + 0.2466 \cdot 10^{-3} \cdot x - 0.5103 \quad (1.3.4)$$

Donde:

$f(x)$  = Fuerza como  $\sin(\theta)$ .

$x$  = PWM en cuentas.

El siguiente paso fue obtener la respuesta dinámica del motor, por lo que se utilizó el ensayo de escalones sucesivos realizado anteriormente. Tomando los datos de la velocidad del motor se analizó el comportamiento dinámico de este en diferentes posiciones. Realizando para cada escalón un ajuste con una ecuación del tipo:

$$f(x) = a \cdot (1 - \exp(-b \cdot x)) + c \quad (1.3.5)$$

Se halló la constante de tiempo correspondiente a cada posición junto con su variación a lo largo del ensayo. Dichos valores se presentan en la siguiente tabla, y fueron calculados empleando la ecuación (1.3.6):

Nº Escalón	PWM	$\tau$
1	1000	0.4142
2	1050	0.3496
3	1100	0.5589
4	1150	0.3394
5	1200	0.3389
6	1250	0.3428
7	1300	0.3384
8	1350	0.3016
9	1400	0.2665

Nº Escalón	PWM	$\tau$
10	1450	0.2027
11	1500	0.2407
12	1550	0.2091
13	1600	0.2271
14	1650	0.341
15	1700	0.2308
16	1750	0.2012
17	1800	0.1209
18	1850	0.1406

Tabla 1.3.1. Valores de constante de tiempo para cada posición

$$\tau = -\frac{1}{b} \quad (1.3.6)$$

Donde podemos ver que para distintas posiciones el sistema se comporta de diferentes maneras, por lo que la dinámica del motor varía en su tiempo de respuesta dependiendo de la posición angular. Sin embargo, al momento de implementar la simulación se consideró un valor promedio de:

$$\tau = 0.3 \text{ s (tau)}.$$

Se muestran a continuación el modelo del sistema de propulsión empleado en las simulaciones, y la comparación de la dinámica de este frente a una de las mediciones de la secuencia de escalones ya mencionada.

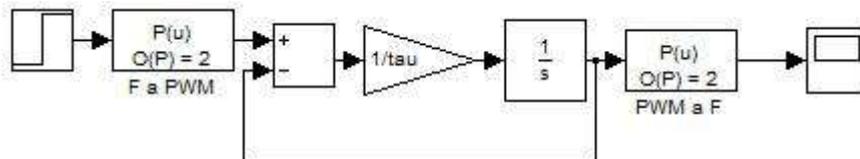


Figura 1.3.7. Modelo de simulación del sistema de propulsión.

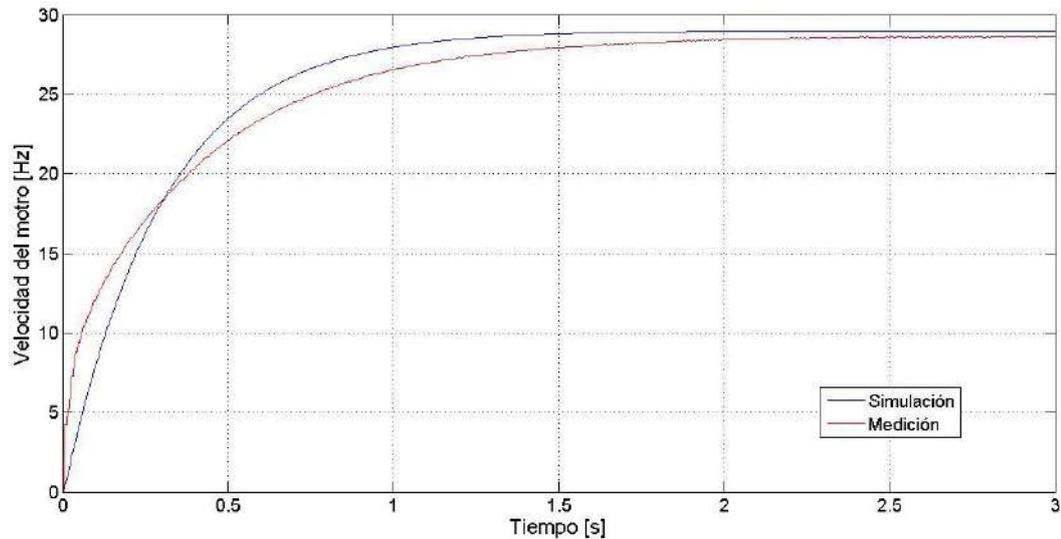


Figura 1.3.8. Respuesta del modelo frente a la medición de la velocidad del motor.

De esta manera, el sistema de propulsión puede verse como una función de transferencia de la forma:

$$G(s) = \frac{1}{\tau \cdot s + 1} \quad (1.3.7)$$

Con  $\tau = 0.3$  s (tau).

Cargar secuencia de escalones (realizar en orden):

- Carpeta Matlab-Simulink
  - o Modelado e identificación
    - Secuencia de escalones
      - carga\_datos.m
      - grafica\_datos.m
      - carga\_datos\_limpios.m
      - grafica\_datos\_limpios.m

Modelado del sistema de propulsión (realizar en orden):

- Carpeta Matlab-Simulink
  - o Modelado e identificación
    - Identificación del motor
      - carga\_datos\_escalones.m
      - Ajuste de fuerza vs hz.cfit
      - Ajuste de fuerza vs pwm.cfit
      - Ajuste de pwm vs fuerza.cfit
      - grafica\_hz1\_vs\_tao.m
      - simulacion\_motor.mdl

## 1.4 Identificación del sistema péndulo

Para la identificación del sistema péndulo se realizó un ensayo en el cual, sin accionar el sistema de propulsión, se lo soltaba en caída libre desde distintas posiciones angulares.

Partiendo de la ecuación (1.2.1) del sistema aeropéndulo pero ahora tomando  $F = 0$ , la expresión que rige al sistema resulta:

$$\dot{\omega} = -\frac{m \cdot g \cdot l}{J_p} \cdot \sin(\theta) - \frac{1}{J_p} \cdot f(\omega) \quad (1.4.1)$$

A lo enunciado anteriormente fue necesario sumarle el efecto de la fricción coulombiana, la cual se opone al movimiento con un esfuerzo normal, quedando la ecuación completa como:

$$\dot{\omega} = -\frac{m \cdot g \cdot l}{J_p} \cdot \sin(\theta) - \frac{1}{J_p} \cdot f(\omega) - \frac{m \cdot g \cdot \mu}{J_p} \cdot \text{signo}(\dot{\theta}) \cdot \cos(\theta) \quad (1.4.2)$$

En la figura 1.4.1 se presenta el modelo realizado en *Simulink* del sistema péndulo.

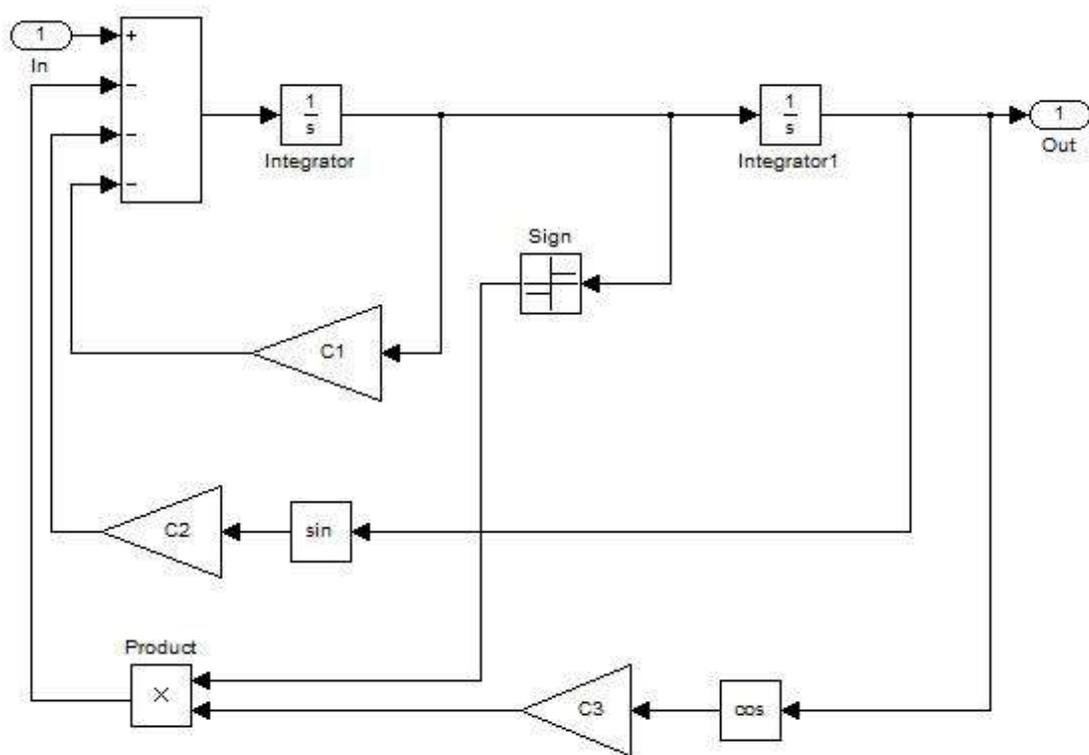


Figura 1.4.1. Modelo de simulación del sistema péndulo

Donde los valores C1, C2 y C3 son las constantes asociadas a los términos de la ecuación (1.4.2): C1 corresponde a la fricción viscosa, C2 al efecto de la gravedad y C3 a la fricción coulombiana. La obtención de los valores numéricos de estas constantes se realizó mediante una parametrización por mínimos cuadrados, donde el sistema planteado fue el siguiente.

$$\dot{\omega} = [-\omega \quad -\sin\theta \quad -\text{signo}(\omega) \cdot \cos\theta] \begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} \quad (1.4.3)$$

Como se conoce la velocidad y la posición del péndulo, ya que son variables medidas, se obtuvo la derivada de la velocidad ( $\dot{\omega}$ ) por medio de un script en *Matlab* y se evaluó el vector fila según la posición y la velocidad angular, despejando de esta manera los valores del vector  $[C1 \ C2 \ C3]^T$  columna por el método de mínimo cuadrados.

A partir de los valores obtenidos de este cálculo y de observar los resultados (ver Anexo 2), se optó por variar los parámetros del modelo manualmente para lograr un mejor ajuste, hasta que de manera iterativa se obtuvieron las siguientes constantes que fueron utilizadas en adelante:

$$C1 = 0.12$$

$$C2 = 27.5$$

$$C3 = 0.09$$

Los resultados obtenidos en simulación se compararon contra las mediciones del ensayo real del péndulo con oscilaciones libres, presentándose dichas comparaciones en las figuras 1.4.2 a 1.4.4:

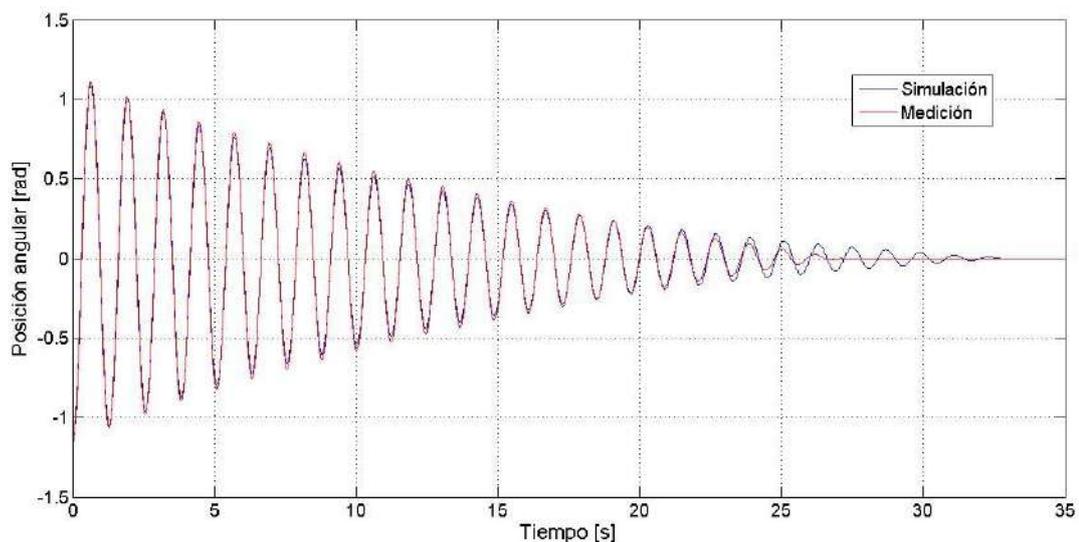


Figura 1.4.2. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente -1.16 rad.

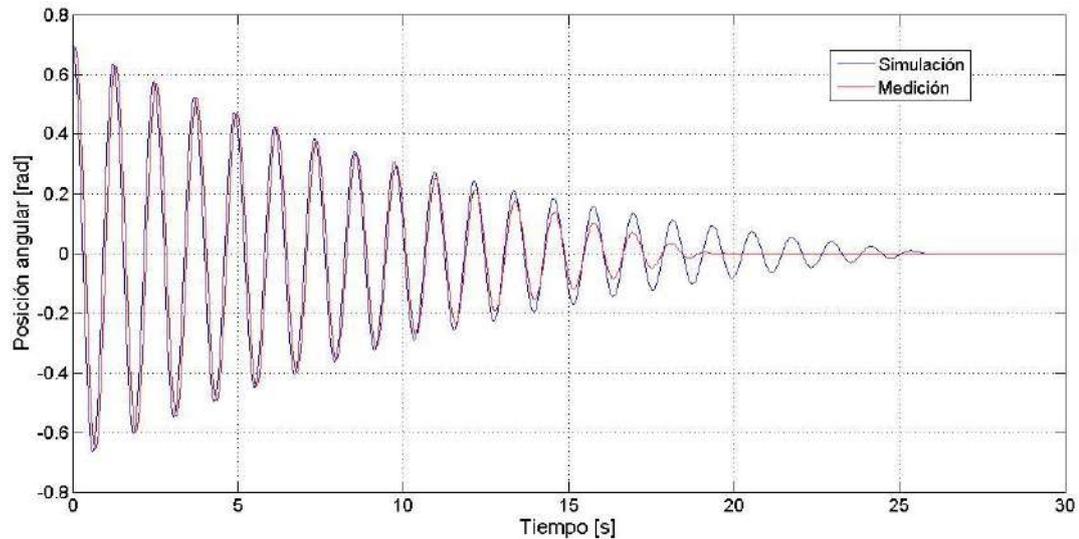


Figura 1.4.3. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente 0.70 rad.

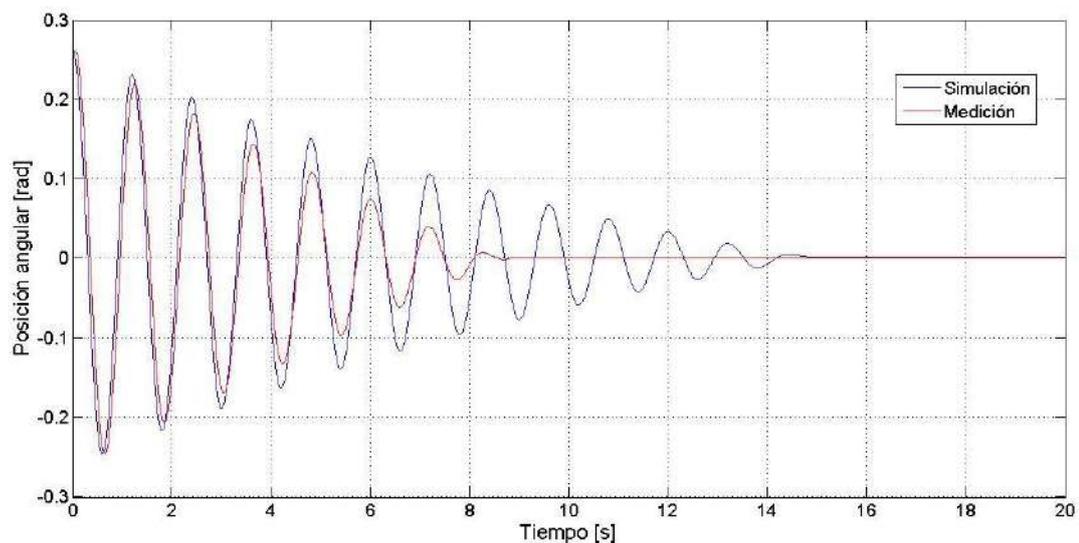


Figura 1.4.4. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente 0.26 rad.

Claramente, la respuesta obtenida del modelo se asemeja a la real, aunque no perfecta. Luego de realizar un análisis de lo que podría estar ocurriendo se arribó a la conclusión que no se estaba considerando la fricción estática. Esta representa la fuerza que se opone al inicio del movimiento de un cuerpo y que se da, por lo tanto, cuando este se encuentra en reposo. Por este motivo, se optó por no considerarla en el modelo del péndulo.

Modelado del sistema péndulo (realizar en orden):

- Carpeta Matlab-Simulink
  - o Identificación del péndulo
    - carga\_datos\_oscilacion.m
    - grafica\_datos\_oscilacion.m
    - identificacion\_parametrica.m
    - simulacion\_pendolo.mdl

### 1.5 Acoplamiento del sistema de propulsión con el sistema péndulo

Una vez modelados e identificados los dos sistemas que componen el aeropéndulo, se procedió al acoplamiento de ambos para conformar así el sistema completo. Partiendo de la ecuación (1.4.2) pero esta vez considerando la fuerza de empuje, se dedujo la siguiente expresión:

$$\dot{\omega} = \frac{l}{J_p} \cdot F - \frac{m \cdot g \cdot l}{J_p} \cdot \sin(\theta) - \frac{1}{J_p} \cdot f(\omega) - \frac{m \cdot g \cdot \mu}{J_p} \cdot \text{signo}(\dot{\theta}) \cdot \cos(\theta) \quad (1.5.1)$$

Ahora bien, debido a que en estado estacionario la ecuación anterior puede reducirse a:

$$\frac{l}{J_p} \cdot F = \frac{m \cdot g \cdot l}{J_p} \cdot \sin(\theta) \quad (1.5.2)$$

Y debido a que  $C2$  representa a la constante que multiplica a la componente de la acción de la gravedad en el sentido de la fuerza de empuje:

$$C2 = \frac{m \cdot g \cdot l}{J_p} \quad (1.5.3)$$

Se decidió establecer la constante de acoplamiento entre los dos sistemas precisamente como la constante  $C2$  ya identificada para el sistema péndulo. Debido a que  $F$  fue ajustada como  $\sin(\theta)$ , esto resultaría en que  $m \cdot g \cong 1$ , que como se comprobó luego, cuenta esta aproximación es lo suficientemente buena como para mantenerla por el resto del proyecto. De este modo, la fuerza de empuje se vuelve un momento sobre el eje del péndulo como:

$$\dot{\omega} = C2 \cdot F - C2 \cdot \sin(\theta) - C1 \cdot \omega - C3 \cdot \text{signo}(\dot{\theta}) \cdot \cos(\theta) \quad (1.5.4)$$

En la figura 1.5.1 se presenta el modelo de simulación en *Simulink* realizado.

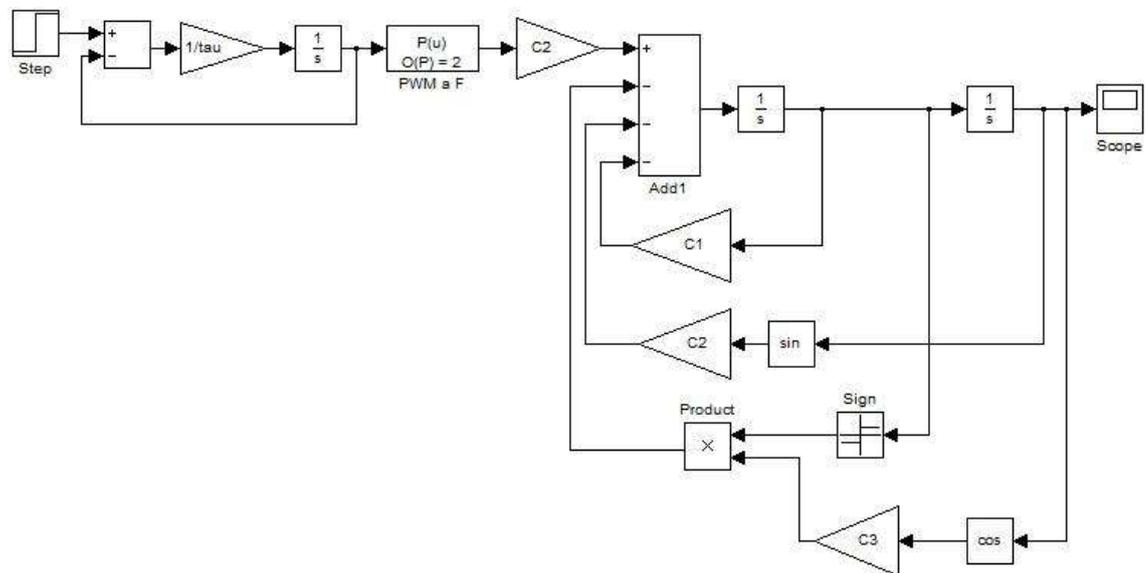


Figura 1.5.1. Modelo de simulación del sistema aeropéndulo completo.

Para comprobar el desempeño de este modelo de simulación, se generó manualmente una secuencia de escalones de PWM (en cuentas) y se registró el comportamiento del sistema real. Las siguientes figuras presentan la comparación entre las mediciones y la respuesta del modelo.

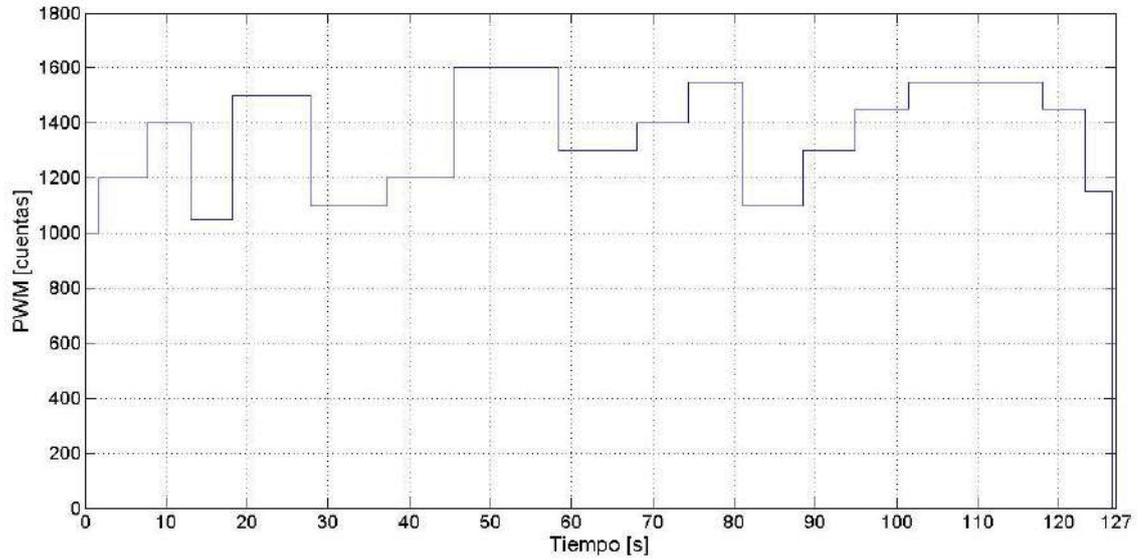


Figura 1.5.2. Secuencia de escalones de entrada a la planta real.

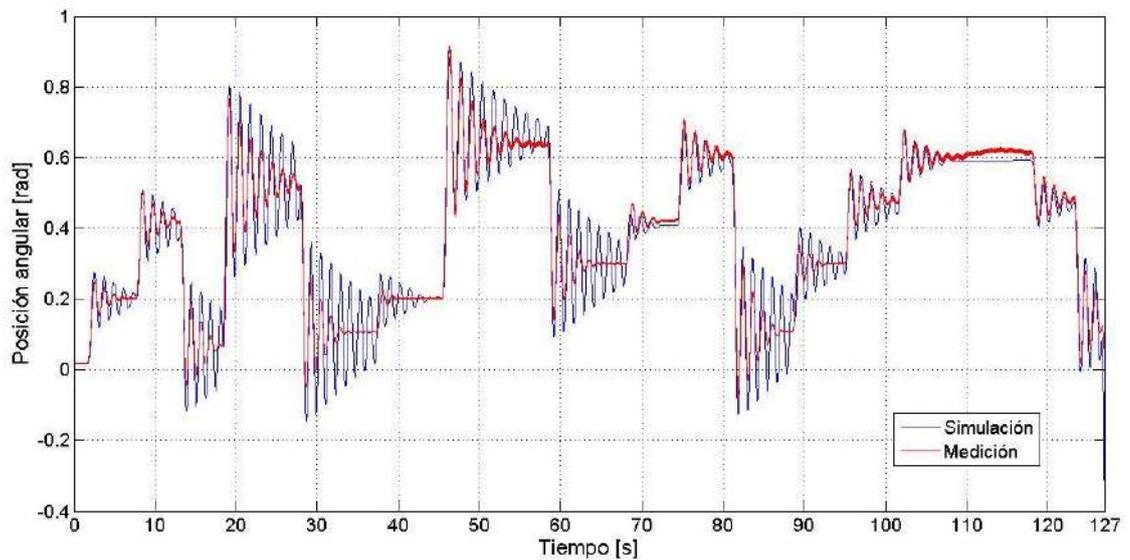


Figura 1.5.3. Respuesta del modelo frente a la medición de la posición angular del sistema aeropéndulo completo para  $C1 = 0.09$ .

Como puede observarse, frente a escalones de 300 cuentas de PWM el modelo resulta más oscilante que la planta real. Para intentar resolver esto, se ajustó la constante  $C1$  que representa la fricción viscosa sobre el sistema, aumentando su valor. Al hacer esto, se llegó a obtener  $C1 = 0.4$  y el resultado fue el siguiente:

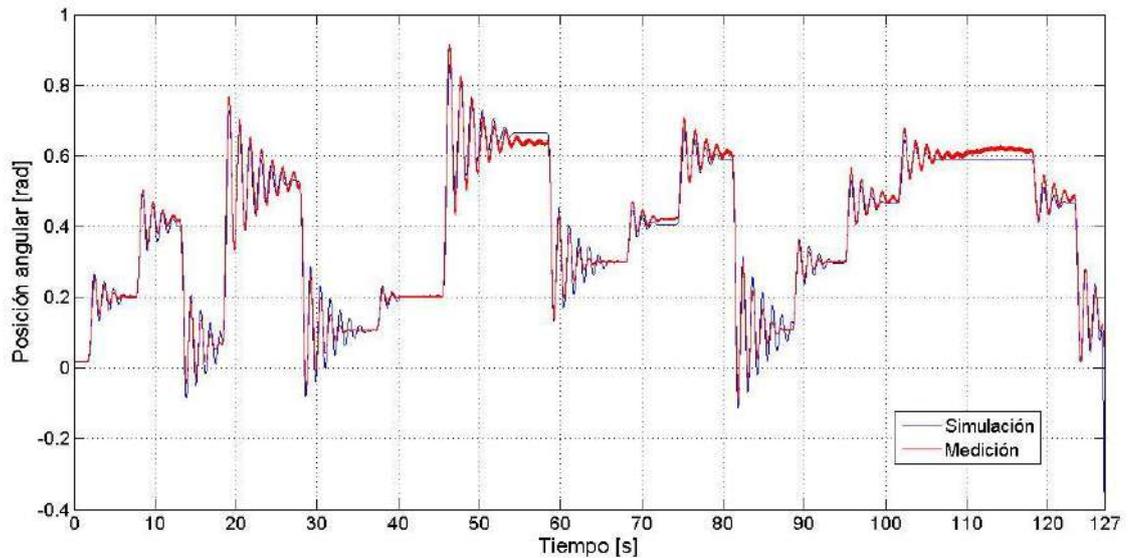


Figura 1.5.4. Respuesta del modelo frente a la medición de la posición angular del sistema aeropéndulo completo para  $C1 = 0.4$ .

Si bien esto disminuía las oscilaciones en los escalones “inferiores”, para los valores de PWM más elevados se observó un amortiguamiento excesivo del modelo a comparación de la planta real. Mientras que, para escalones de pequeña amplitud el modelo no era capaz de copiar la misma respuesta que la planta. Por estos motivos, pero principalmente el último, se decidió conservar los valores de  $C1$ ,  $C2$  y  $C3$  que ya habían sido obtenidos para la primera simulación del sistema aeropéndulo completo.

Modelado sistema completo (realizar en orden):

- Carpeta Matlab-Simulink:
  - o Modelado e identificación
    - carga\_seriale\_random.m
    - sistema\_acoplado.mdl

## 2 Linealización del sistema aeropéndulo y desarrollo del primer controlador

### 2.1 Linealización del sistema aeropéndulo

Con el fin de diseñar un primer controlador empleando las técnicas de control clásico (modelo externo), se linealizó la planta real alrededor de distintos puntos de equilibrio dentro del rango de 0° a 90° y se construyó un modelo lineal. Para esto, se halló la expresión linealizada de la ecuación (1.2.1) alrededor de los distintos puntos de equilibrio seleccionados, claramente dejándose de lado la componente de la fricción coulombiana dependiente del signo de la velocidad del péndulo. La ecuación resultante es:

$$\delta E - m \cdot g \cdot l \cdot \cos(\theta_0) \cdot \delta\theta = J_p \cdot \delta\ddot{\theta} + f \cdot \delta\dot{\theta} \quad (2.1.1)$$

Donde:

$\theta_0$  = Posición de equilibrio.

$\delta E$  = Variación del momento alrededor del eje del péndulo.

$\delta\theta$  = Variación de la posición angular del péndulo.

$f$  = Constante correspondiente a la fricción viscosa.

De esta manera, despejando la segunda derivada de la posición angular del péndulo  $\delta\ddot{\theta}$ , se obtiene la ecuación con la cual se construyó el modelo linealizado:

$$\delta\ddot{\theta} = \frac{1}{J_p} \cdot \delta E - \frac{m \cdot g \cdot l \cdot \cos(\theta_0)}{J_p} \cdot \delta\theta - \frac{f}{J_p} \cdot \delta\dot{\theta} \quad (2.1.2)$$

Para la identificación de las constantes del modelo se utilizó nuevamente una parametrización por mínimos cuadrados, tomándose como señal de entrada la secuencia de escalones mostrada en la figura 1.3.1. Esta señal fue recortada alrededor de distintos puntos de equilibrio y la ecuación (2.1.2) parametrizada para cada caso, utilizando:

$$\delta\ddot{\theta} = \begin{bmatrix} -\delta\theta & -\delta\dot{\theta} & \delta E \end{bmatrix} \begin{bmatrix} \text{theta1} \\ \text{theta2} \\ \text{theta3} \end{bmatrix} \quad (2.1.3)$$

A continuación se presenta una tabla con los distintos puntos de equilibrio tomados, las constantes obtenidas y el rango útil de PWM de cada linealización (ver Anexo 3):

Punto de equilibrio (PWM en cuentas)	<i>theta1</i>	<i>theta2</i>	<i>theta3</i>	Rango útil (PWM en cuentas)
1200	27.25	3.601	1.01	1000-1300
1350	25.23	1.642	1.197	1300-1450
1450	22.86	0.6902	0.9802	1400-1600
1550	17.29	0.6261	0.6138	1450-1650
1650	17.73	1.035	1.07	1600-1700

Tabla 2.1.1. Constantes del sistema linealizado.

De esta manera, puede construirse para cada caso una función de transferencia que simula la dinámica del sistema alrededor de los puntos de equilibrio respectivos, cuya señal de entrada es el momento ejercido por el sistema de propulsión.

$$G_1(s) = \frac{1.01}{s^2 + 3.601 \cdot s + 27.25}$$

$$G_2(s) = \frac{1.197}{s^2 + 1.642 \cdot s + 25.23}$$

$$G_3(s) = \frac{0.9802}{s^2 + 0.6902 \cdot s + 22.86}$$

$$G_4(s) = \frac{0.6138}{s^2 + 0.6261 \cdot s + 17.29}$$

$$G_5(s) = \frac{1.07}{s^2 + 1.035 \cdot s + 17.73}$$

Ahora bien, como ya se comentó al inicio de esta sección, el rango de linealización fue establecido entre  $0^\circ$  y  $90^\circ$ , por lo tanto, las funciones de transferencia obtenidas representan al sistema dentro de su zona estable. Para determinar el comportamiento en la zona inestable de  $90^\circ$  a  $180^\circ$ , se dedujeron las expresiones de otras cinco plantas simétricas a las ya calculadas. Esto es: la relación entre empuje y posición angular del sistema es simétrica respecto a la horizontal del péndulo, de modo que el mínimo empuje necesario se da cerca de  $0^\circ$  y de  $180^\circ$ , mientras que el máximo se encuentra sobre  $90^\circ$ . En base a esto, se estableció la siguiente simetría de la ecuación (1.3.1):

$$F_0 = m \cdot g \cdot \sin(\theta_0) = m \cdot g \cdot \sin(\pi - \theta_0) \quad (2.1.4)$$

Donde:

$F_0$  = Fuerza de empuje en el  $\theta_0$ .

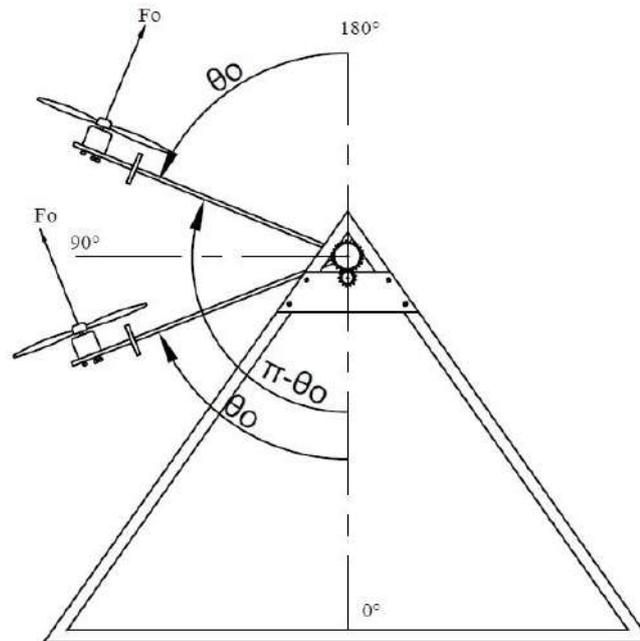


Figura 2.1.1. Diagrama de simetría.

De este modo, la ecuación del sistema linealizado sobre la zona inestable resulta:

$$\delta E - m \cdot g \cdot l \cdot \cos(\pi - \theta_0) \cdot \delta \theta = J_p \cdot \delta \ddot{\theta} + f \cdot \delta \dot{\theta} \quad (2.1.5)$$

Y volviendo a despejar:

$$\delta \ddot{\theta} = \frac{1}{J_p} \cdot \delta E - \frac{m \cdot g \cdot l \cdot \cos(\pi - \theta_0)}{J_p} \cdot \delta \theta - \frac{f}{J_p} \cdot \delta \dot{\theta} \quad (2.1.6)$$

De una simple observación, se dedujo que solo la constante que acompaña a  $\delta \theta$  resulta diferente a la planta simétrica de la zona estable:

$$theta3' = \frac{1}{J_p} = theta3$$

$$theta2' = \frac{f}{J_p} = theta2$$

$$theta1' = \frac{m \cdot g \cdot l \cdot \cos(\pi - \theta_0)}{J_p} = -\frac{m \cdot g \cdot l \cdot \cos(\theta_0)}{J_p} = -theta1$$

Así, para cada planta estable, corresponde una expresión simétrica de una planta inestable:

$$G_1'(s) = \frac{1.01}{s^2 + 3.601 \cdot s - 27.25}$$

$$G_2'(s) = \frac{1.197}{s^2 + 1.642 \cdot s - 25.23}$$

$$G_3'(s) = \frac{0.9802}{s^2 + 0.6902 \cdot s - 22.86}$$

$$G_4'(s) = \frac{0.6138}{s^2 + 0.6261 \cdot s - 17.29}$$

$$G_5'(s) = \frac{1.07}{s^2 + 1.035 \cdot s - 17.73}$$

Linealización del sistema aeropéndulo (realizar en orden):

- Carpeta Matlab-Simulink
  - o Linealización
    - datos\_para\_calculo\_empuje\_equilibrio.m
    - calculo\_empuje\_equilibrio.mdl
    - para\_esc\_1250.m
    - para\_esc\_1400.m
    - para\_esc\_1500.m
    - para\_esc\_1600.m
    - para\_esc\_1700.m
    - pendulo\_linealizado.mdl
    - plantas\_inestables.m
    - polos\_vivero.m

## 2.2 Desarrollo del primer controlador

Con las plantas linealizadas que representa al sistema aeropéndulo, tanto en su zona estable como inestable, se procedió al diseño del primer controlador utilizando las técnicas de control clásico (modelo externo). Para ello, en primer lugar se ubicaron en un mismo gráfico los polos de cada una de las funciones de transferencia, tanto para la zona estable como para la zona inestable.

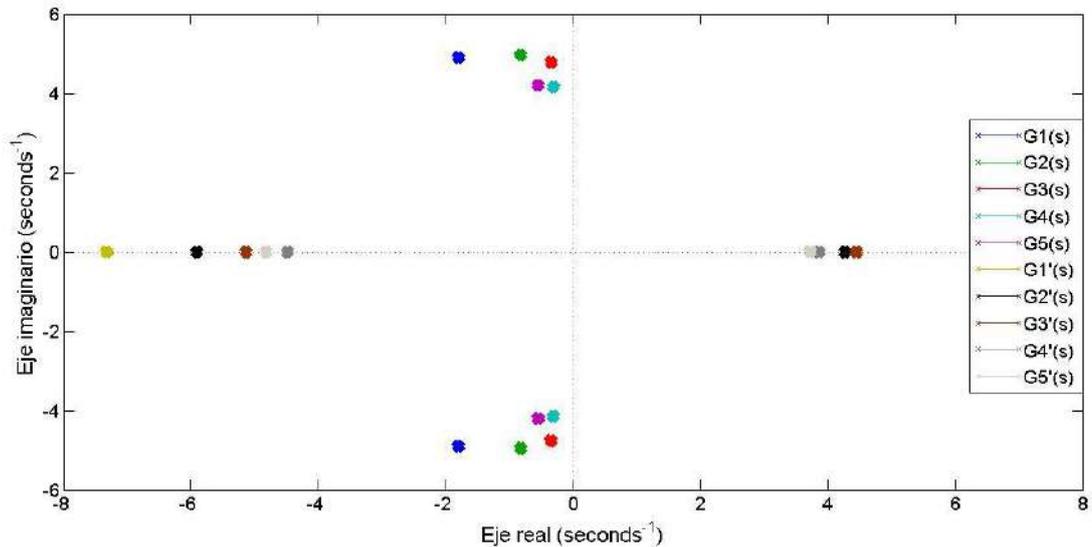


Figura 2.2.1. Ubicación de los polos de las plantas estables e inestables.

Como puede apreciarse, las plantas estables poseen polos complejos conjugados dentro del semiplano izquierdo, aproximándose hacia el eje imaginario a medida que el sistema se aproxima a los  $90^\circ$ , aumentando las oscilaciones en la respuesta. En el caso de las plantas inestables, los polos son reales, con uno estable y otro inestable. Con esto presente, el primer controlador diseñado se realizó en base a la planta  $G_3'(s)$  cuyo polo inestable es el más alejado. Acoplado la función de transferencia del sistema de propulsión, la planta a controlar resultó:

$$G(s) = G_3'(s) \cdot \frac{27.5}{(0.3 \cdot s + 1)}$$

Empleando el diseño por emulación, con un tiempo de muestreo de  $T = 0.004$  s, se obtuvo el controlador de la ecuación (2.2.1), que luego fue transformado a su forma discreta, ecuación (2.2.2) (ver Anexo 4).

$$D(s) = \frac{200 \cdot (s+5.139) \cdot (s+3.333)}{(s+200) \cdot (s+10)} \quad (2.2.1)$$

$$D(z) = \frac{137.2534 \cdot (z-0.9868) \cdot (z-0.9797)}{(z-0.9608) \cdot (z-0.4493)} \quad (2.2.2)$$

Antes de realizar la implementación sobre el sistema real, se construyó la siguiente simulación con el fin de comprobar el desempeño de este controlador, tanto en el rango estable como en el inestable:

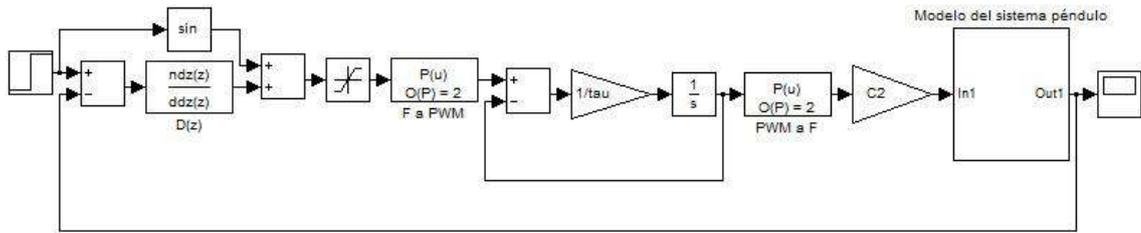


Figura 2.2.2. Modelo del sistema aeropéndulo a lazo cerrado.

El bloque denotado como “Modelo del sistema péndulo” es un subsistema que agrupa el modelo presentado en la figura 1.4.1, mientras que la referencia al sistema es un escalón de posición angular en radianes.

Como puede observarse, además de la implementación del controlador (2.2.2), también se implementó un bloque de *feed-forward* cuya función es sumar a la acción de control un valor calculado como  $\sin(\theta_r)$ , donde  $\theta_r$  es la posición angular de referencia en radianes. De este modo, logra reducirse el error en estado estacionario del sistema.

Además de este bloque de *feed-forward*, se colocó también un bloque de saturación sobre la acción de control, para contemplar las limitaciones reales de la fuente de alimentación del sistema, como ya se explicó en la sección 1.1. Por este motivo, tanto en las simulaciones como en cada algoritmo, se estableció esta saturación de la acción de control, cuyo rango es entre 1.92 N (aproximadamente 2300 cuentas de PWM) y 0 N (977.5 cuentas de PWM).

El resultado obtenido de la simulación puede apreciarse en las siguientes figuras:

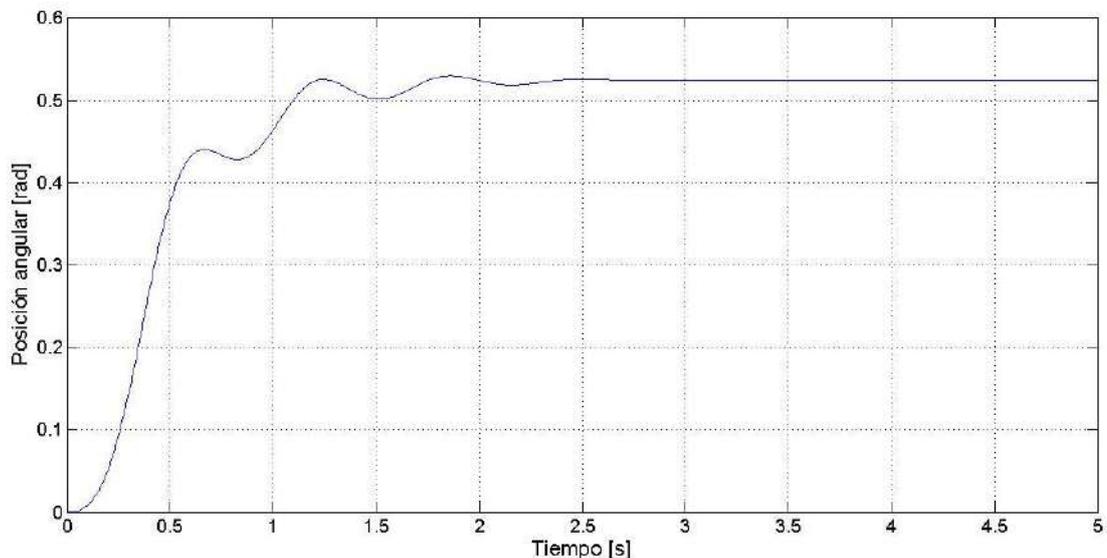


Figura 2.2.3. Respuesta del modelo de simulación a lazo cerrado para una referencia de 0.5236 rad ( $30^\circ$ ).

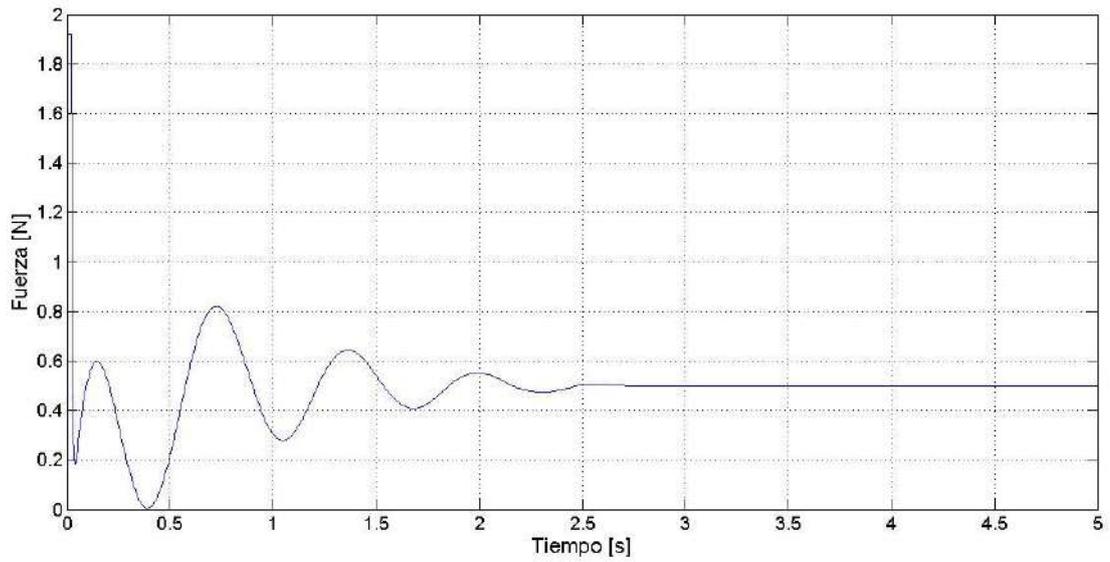


Figura 2.2.4. Acción de control sobre el sistema de propulsión para una referencia de  $0.5236 \text{ rad}$  ( $30^\circ$ ).

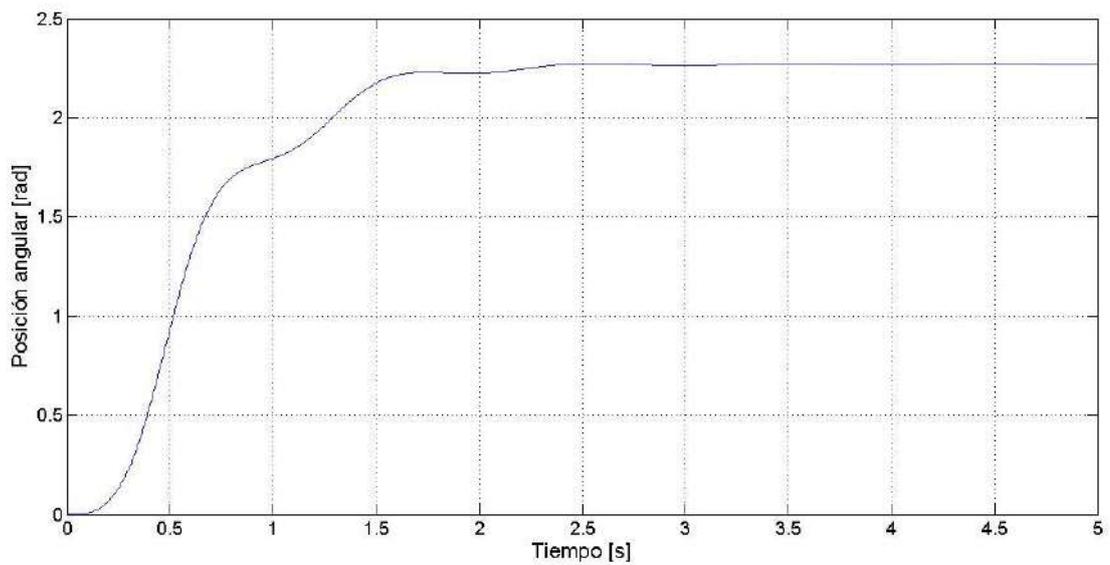


Figura 2.2.5. Respuesta del modelo de simulación a lazo cerrado para una referencia de  $2.2689 \text{ rad}$  ( $130^\circ$ ).

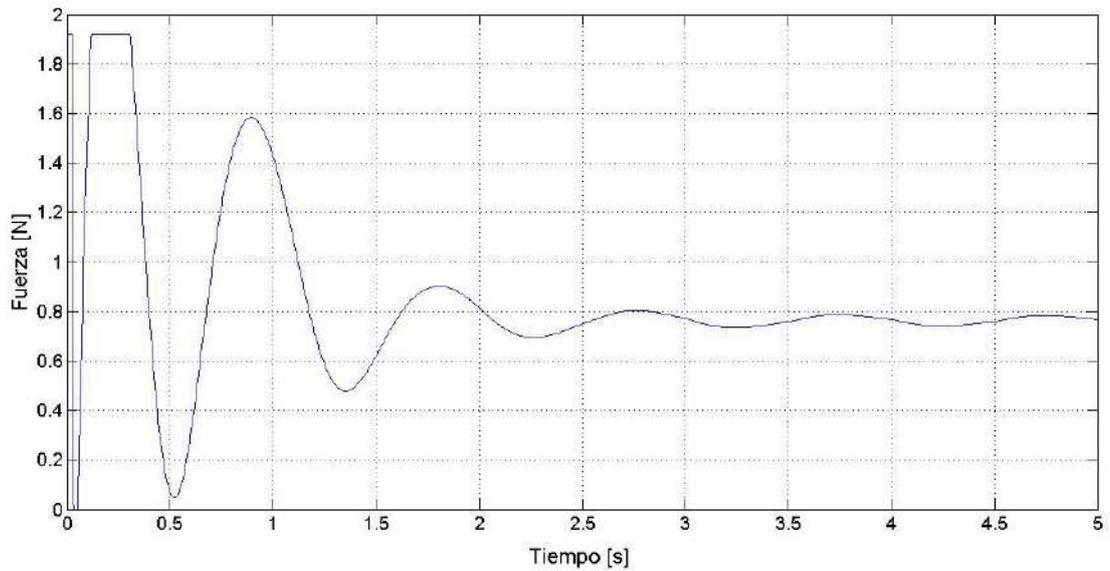


Figura 2.2.6. Acción de control sobre el sistema de propulsión para una referencia de 2.2689 rad (130°).

Como puede observarse, la respuesta es estable tanto para valores de referencia dentro del rango 0° a 90° como también de 90° a 180°.

Comprobado esto, se procedió a la implementación del algoritmo de control sobre el sistema real, cuya respuesta ante una referencia de 0.2618 rad (15°) fue la siguiente:

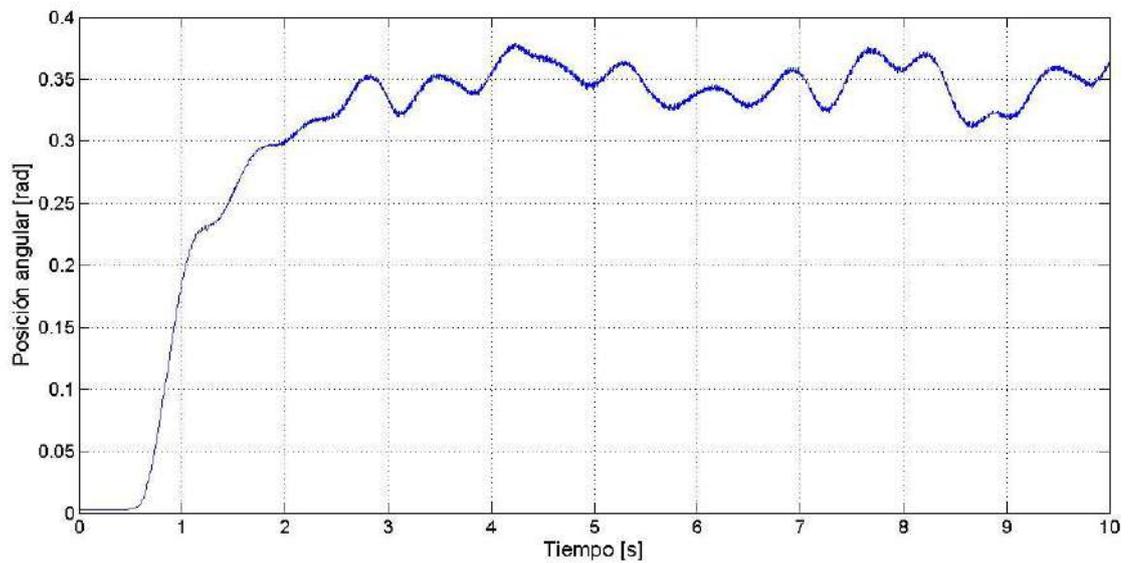


Figura 2.2.7. Respuesta del sistema real ante una referencia de 0.2618 rad (15°).

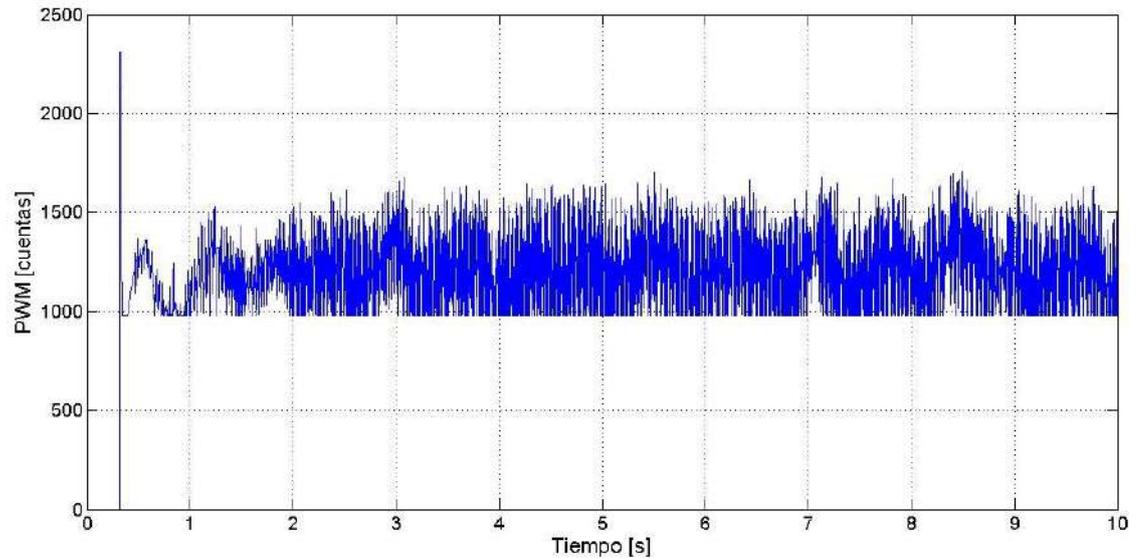


Figura 2.2.8. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Claramente, esta no fue la respuesta esperada, y menos aún la acción de control. Analizando el motivo de esto, se comprobó que el problema era el ruido en la medición de la posición, la cual al ser derivada por la doble red de adelanto del controlador produce una señal de salida como la observada en la figura 2.2.8.

Calculo del controlador clásico (realizar en orden):

- Carpeta Matlab-Simulink
  - o Controlador Clásico y PD
    - controlador\_emulacion.m
    - pendulo\_controlador\_clasico.mdl
  - o Filtro de Kalman
    - log\_ruido
      - log2.txt

Código en C:

- Carpeta Aeropendolo
  - o controlclasico.c

### 3 Filtro de Kalman Extendido (EKF)

#### 3.1 Cálculo e implementación del EKF

Con el objetivo de solucionar el problema de ruido en las mediciones del sistema, se implementó un Filtro de Kalman Extendido (debido a que la planta no es lineal), cuyas ecuaciones son:

**Predicción:**

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k)$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_k$$

**Actualización:**

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1})$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

Donde:

$$F_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_k}$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k|k-1}}$$

Ahora bien, las variables de estado del sistema son tres: posición angular del péndulo, velocidad angular del péndulo y fuerza ejercida por el sistema de propulsión, de modo que la matriz  $F_{k-1}$ , calculada en base a las ecuaciones (1.3.7) y (1.5.4), resulta:

$$F_{k-1} = \begin{bmatrix} 1 & T & 0 \\ -C2 \cdot \cos(\theta_0) \cdot T & 1 - C1 \cdot T & C2 \cdot T \\ 0 & 0 & 1 - \frac{T}{\tau} \end{bmatrix} \quad (3.1.1)$$

La matriz  $H_k$  de medición se construyó con la posición angular del péndulo y la fuerza ejercida por el motor y la hélice (medida indirectamente a través de la ecuación (1.3.2)):

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1.2)$$

Mientras, las matrices de covarianza  $Q_{EKF}$  (ruido del proceso) y  $R_{EKF}$  (ruido en la medición) fueron establecidas como:

$$Q_{EKF} = \begin{bmatrix} 1e-7 & 0 & 0 \\ 0 & 3e-5 & 0 \\ 0 & 0 & 4e-7 \end{bmatrix} \quad (3.1.3)$$

$$R_{EKF} = \begin{bmatrix} 6.04e-6 & 0 \\ 0 & 1.4e-6 \end{bmatrix} \quad (3.1.4)$$

Para la construcción de  $Q_{EKF}$  se tomó como valores de la diagonal la inversa del cuadrado del máximo valor de la variable correspondiente, y luego fue ajustada de manera iterativa reduciendo el orden de magnitud hasta obtener la matriz ya presentada. En el caso de  $R_{EKF}$  se calculó la varianza del ruido en las mediciones de posición angular y de fuerza de empuje, valores que se mantuvieron en su expresión final.

Una vez programado el EKF en el algoritmo, se implementó un control en el ingreso de referencia al sistema, siendo esta alcanzada por medio de una rampa en vez de un escalón con el fin de no generar un elevado consumo de corriente. La respuesta del sistema a lazo cerrado mejoró así sustancialmente, como puede observarse en las siguientes figuras:

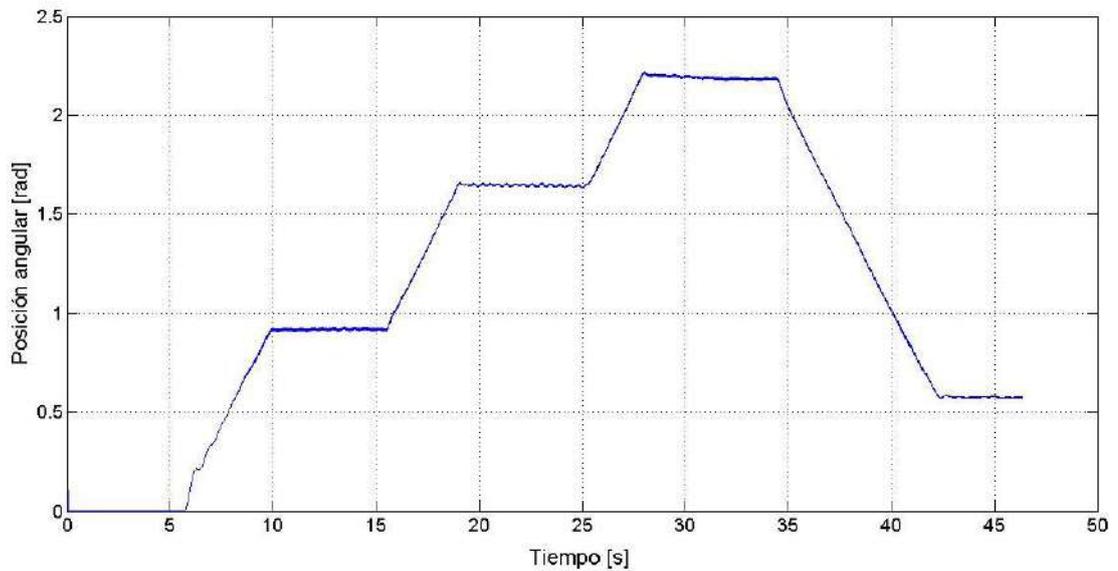


Figura 3.1.1. Respuesta del sistema ante diferentes referencias de posición angular en radianes.

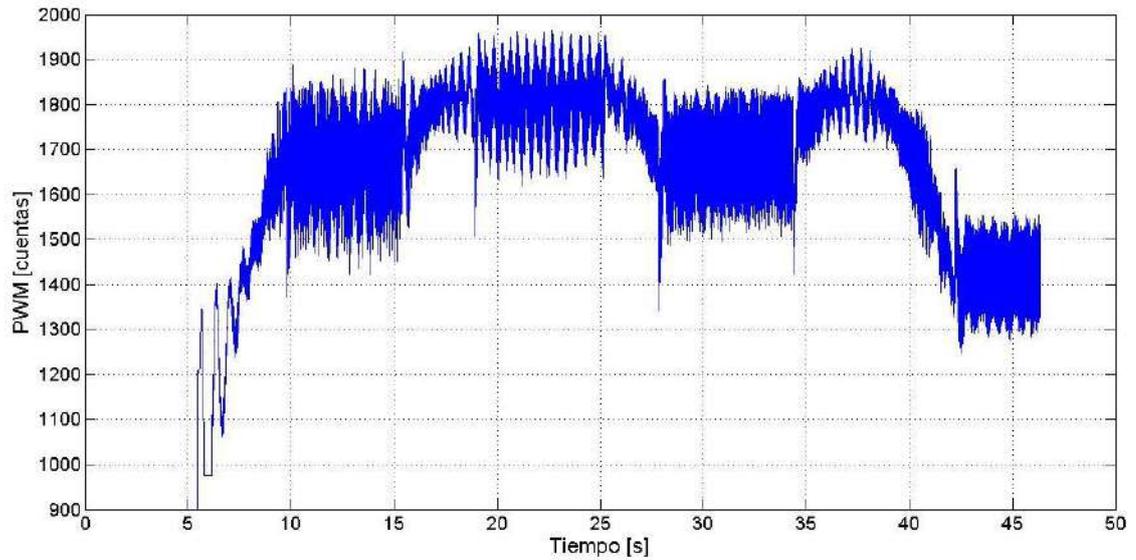


Figura 3.1.2. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Como muestran estas imágenes, el sistema controlado es estable tanto en el rango  $0^\circ$  a  $90^\circ$  como de  $90^\circ$  a  $180^\circ$ , y su comportamiento ya no es tan fluctuante como en la figura 2.2.7. La acción de control aún sigue siendo oscilante, pero con una amplitud menor que en la figura 2.2.8, más de 500 cuentas de PWM contra poco más de 300 observadas en esta última imagen.

Para apreciar la acción del EKF, se muestra una comparación entre las mediciones de las variables de estado y las estimaciones de estas, producto del filtro. Puede verse que la estimación de las variables disminuye el ruido presente en las mediciones, especialmente en la velocidad angular del péndulo:

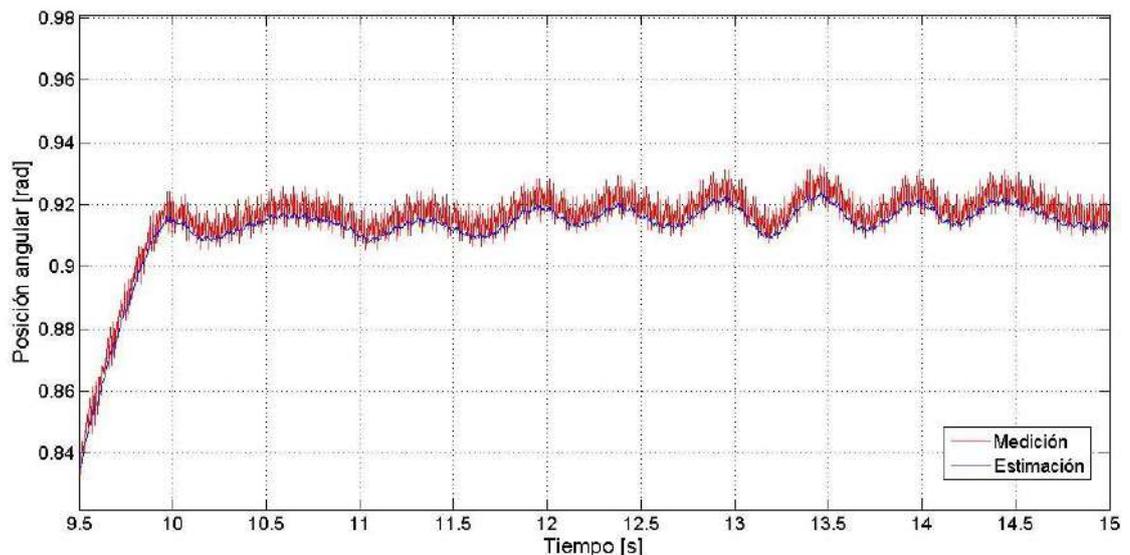


Figura 3.1.3. Comparación entre la variable posición angular medida y la estimada.

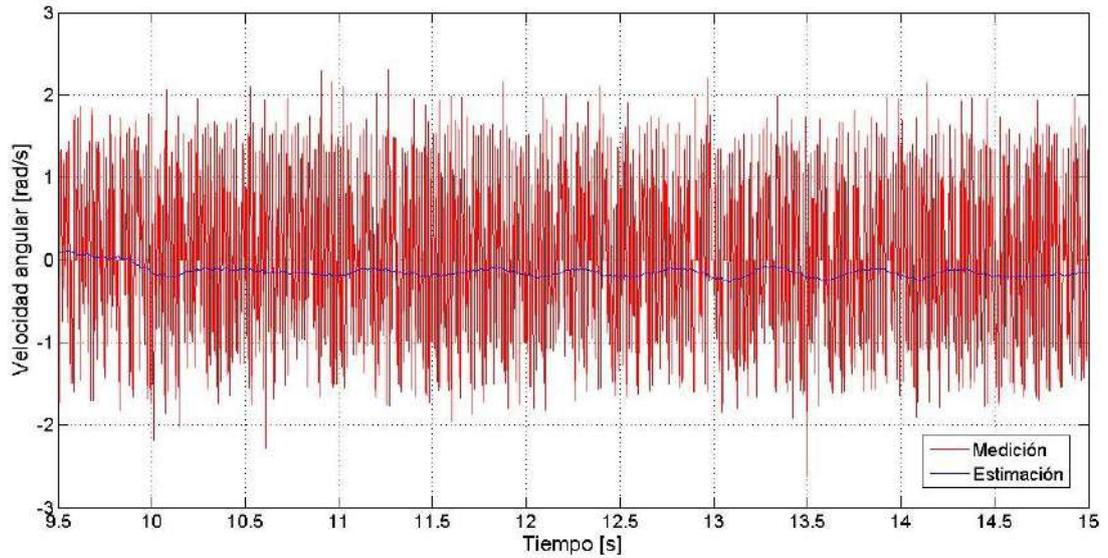


Figura 3.1.4. Comparación entre la variable velocidad angular medida y la estimada.

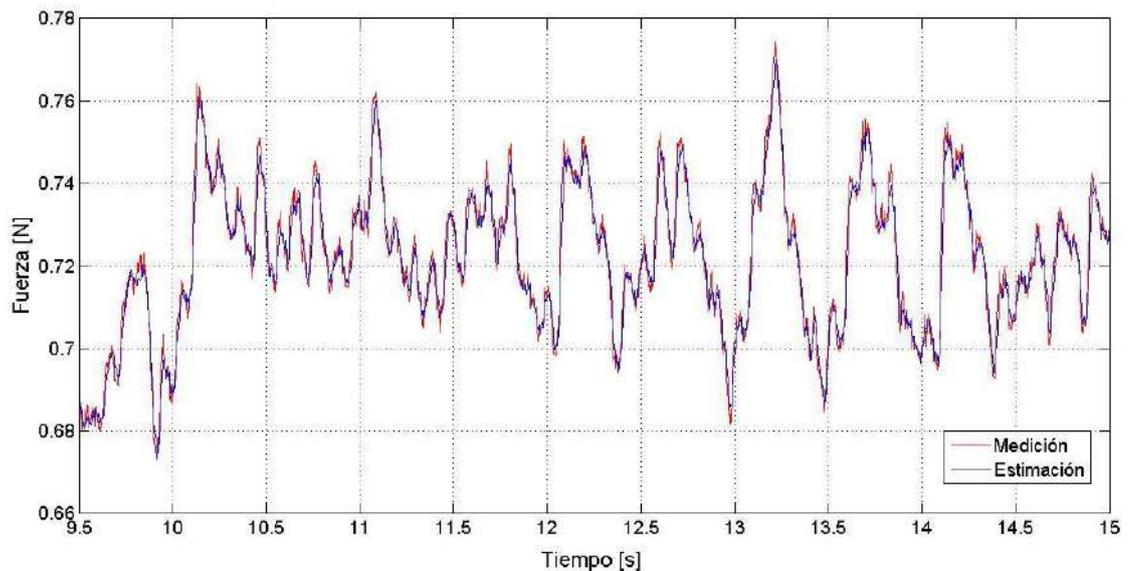


Figura 3.1.5. Comparación entre la variable fuerza de empuje “medida” y la estimada.

Nota: la información presentada en estas últimas tres figuras son acercamientos del mismo ensayo presentado en la figura 3.1.1, notar que el intervalo de tiempo graficado es de 9.5 s a 15 s.

Para los ensayos de los distintos controladores diseñados, se implementó un modelo del EKF en *Simulink* capaz de realizar el algoritmo de estimación en simulación. Para ello, se desarrolló una función en *Matlab* denominada *FiltroK* que es llamada por un bloque del tipo *Interpreted MATLAB Fcn* al momento de ejecutarse la simulación. En la figura 3.1.6 se presenta el diagrama de bloques creado en *Simulink* a tal efecto, el cual luego sería agrupado en subsistema denominado EKF, con tres entradas: las mediciones de posición angular y de fuerza de empuje, y la acción de control que ingresa

al sistema de propulsión ya saturada. Las salidas en esta simulación son cuatro: por un lado, cada una de las estimaciones, y por otro, un vector formado por dichas estimaciones que luego sería utilizado en la realimentación de estados.

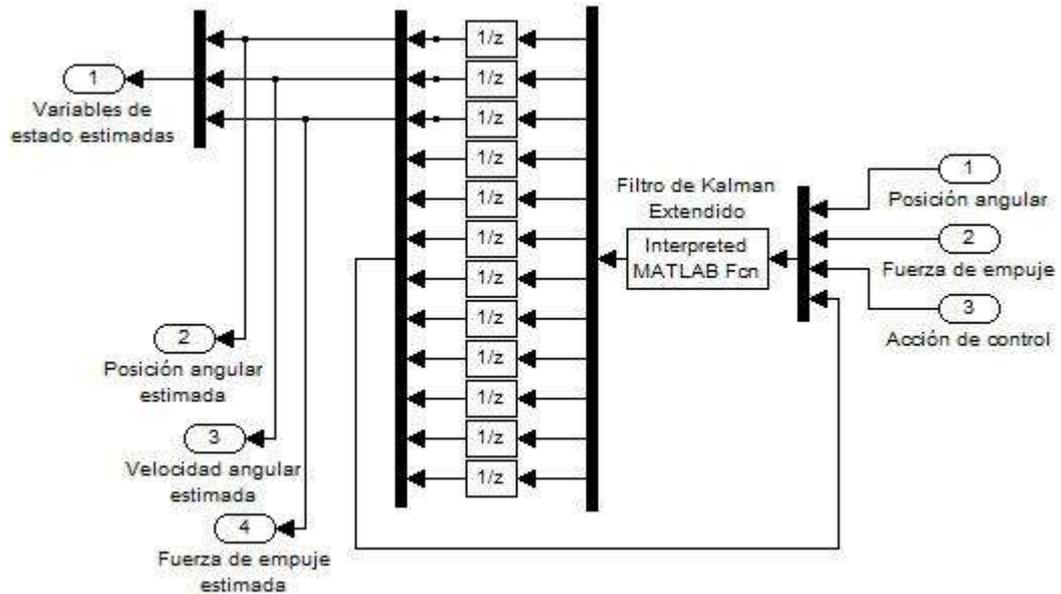


Figura 3.1.6. Simulación de Filtro de Kalman Extendido.

Filtro de Kalman Extendido:

- Carpeta Matlab-Simulink
  - o Filtro de Kalman
    - carga\_grafica\_datos\_kalman.m

Datos del controlador clásico con EKF:

- Carpeta Matlab-Simulink
  - o Logs varios controladores
    - ControlClasicoKalman.txt

Código en C:

- Carpeta Aeropendulo
  - o ControlclasicoKalman.cpp

## 4 Diseño e implementación de distintas técnicas de control

### 4.1 Controlador PD en la realimentación

La primera alternativa que se planteó luego de la doble red de adelanto fue un controlador PD en la realimentación de las variables estimadas por el EKF. La ecuación de este controlador se definió como:

$$U(s) = K \cdot (1 + \tau_1 \cdot s) \cdot (1 + \tau_2 \cdot s) \cdot \theta(s) \quad (4.1.1)$$

De este modo, la acción de control resulta, efectivamente, en una realimentación de las tres variables de estado estimadas:  $\hat{\theta}$ ,  $\hat{\theta} = \hat{\omega}$  y  $\hat{\theta} = C2 \cdot \hat{F} - C2 \cdot \sin(\hat{\theta}) - C1 \cdot \hat{\omega}$ . Haciendo uso de la herramienta *sisotool*, los valores finales de  $K$ ,  $\tau_1$  y  $\tau_2$  que fueron implementados en el controlador son:

$$K = 6$$

$$\tau_1 = 0.39$$

$$\tau_2 = 0.14$$

Y la respuesta obtenida fue la siguiente:

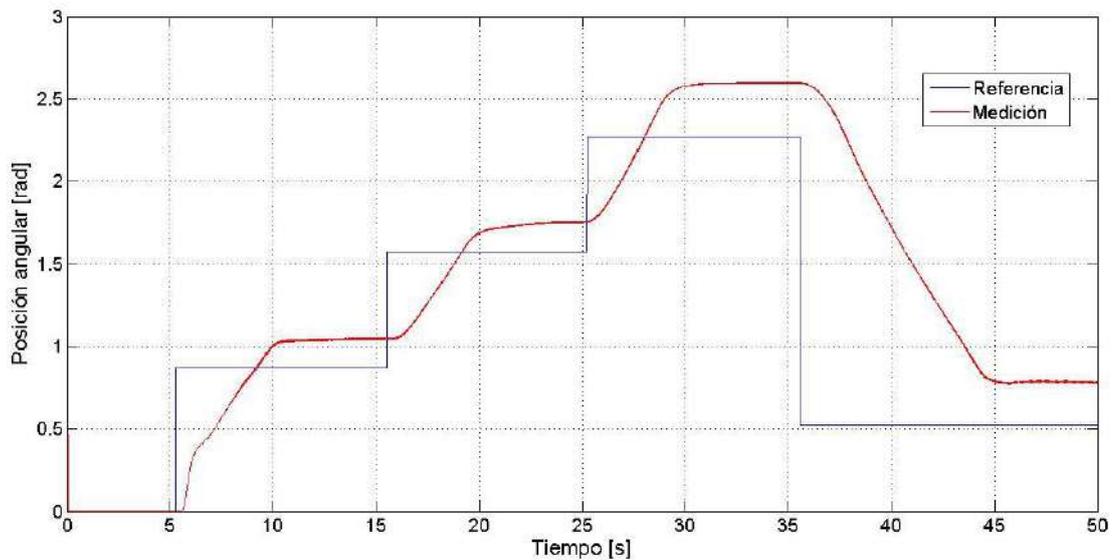


Figura 4.1.1. Respuesta del sistema con controlador PD en la realimentación.

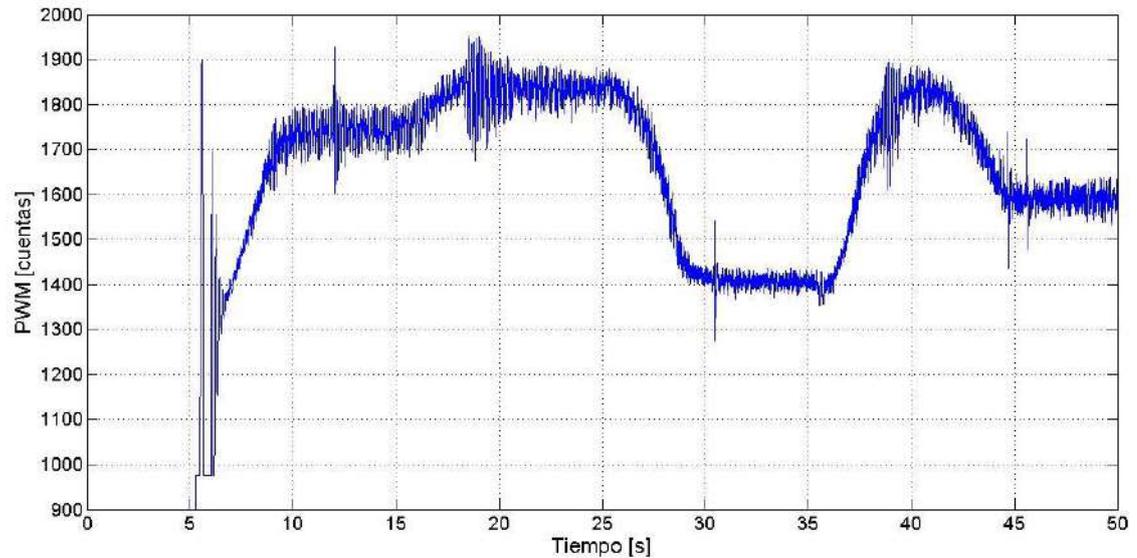


Figura 4.1.2. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Como puede verse, la respuesta del sistema mejora sustancialmente a comparación de lo mostrado en la figura 3.1.1 presentando menos oscilaciones en estado estacionario, al igual que la acción de control, cuyas variaciones se encuentran en un promedio de 100 cuentas de PWM en estado estacionario. Sin embargo, al igual que el primer controlador, la respuesta del sistema presenta error en estado estacionario. Para solucionar esto, se diseñó e implementó un control integral por variables de estado.

Algo interesante que puede apreciarse ahora que la acción de control es más nítida, es el intervalo entre los 25 y 40 s. Como se explicó en la sección 2.1, el sistema es simétrico a los  $90^\circ$ , por lo que un aumento de la posición angular en la zona inestable se debe corresponder con una disminución de la fuerza de empuje y viceversa. En particular, puede verse como, durante el último escalón, la acción de control aumenta de manera progresiva hasta los 40 s. En instante el sistema pasa por  $90^\circ$  y la fuerza de empuje es máxima, para luego disminuir hasta alcanzar la posición final en  $30^\circ$ .

#### Calculo de controlador PD

- Carpeta Matlab-Simulink
  - o Controlador Clásico y PD
    - controlador\_con\_ceros

#### Datos del controlador PD:

- Carpeta Matlab-Simulink
  - o Logs varios controladores
    - ControlCeros.txt

#### Código en C:

- Carpeta Aeropendolo
  - o ControlCero.cpp

## 4.2 Controlador integral por variables de estado

El tercer controlador diseñado fue mediante la técnica de control integral por variables de estado, con el fin de eliminar el error en estado estacionario que poseen los anteriores algoritmos implementados. Esto se logró aumentando el tipo del sistema, es decir, agregando una nueva variable de estado que fuera proporcional a la integral de la salida.

Con base en la linealización de los modelos en continuo, tanto del sistema de propulsión como del sistema péndulo, se definieron las siguientes variables de estado (estimadas por el EKF): la posición angular, la velocidad angular y la fuerza de empuje. De este modo, el sistema queda descrito por las ecuaciones:

$$\begin{cases} \dot{x} = F \cdot X + G \cdot u \\ y = H \cdot x \end{cases} \quad (4.2.1)$$

Vector de estados:

$$x = \begin{bmatrix} \theta \\ \omega \\ F \end{bmatrix} \quad (4.2.2)$$

Matrices  $F$ ,  $G$  y  $H$ :

$$F = \begin{bmatrix} 0 & 1 & 0 \\ tetha(1) & tetha(2) & tetha(3) \\ 0 & 0 & -1/\tau \end{bmatrix} \quad (4.2.3)$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 1/\tau \end{bmatrix} \quad (4.2.4)$$

$$H = [1 \ 0 \ 0] \quad (4.2.5)$$

Al agregarse el estado integral definido como  $x_I = \theta - \theta_{ref}$ , el sistema extendido pudo ser expresado como:

Vector de estados extendido:

$$x_{ext} = \begin{bmatrix} x_I \\ \theta \\ \omega \\ F \end{bmatrix} \quad (4.2.6)$$

Matrices  $F$ ,  $G$  y  $H$  extendidas:

$$F_{ext} = \begin{bmatrix} 1 & H \\ 0 & F \end{bmatrix} \quad (4.2.7)$$

$$G_{ext} = \begin{bmatrix} 0 \\ G \end{bmatrix} \quad (4.2.8)$$

$$H_{ext} = [0 \ 1 \ 0 \ 0] \quad (4.2.9)$$

Por lo tanto, la acción de control que se implementó en el sistema realimentado fue:

$$u = -K_I \cdot x_I - K_1 \cdot \theta - K_2 \cdot \omega - K_3 \cdot F \quad (4.2.10)$$

Para calcular estos valores de ganancia en la realimentación, se utilizó la técnica de control óptimo con la función *lqr* de *Matlab*, donde las matrices de pesos empleadas fueron:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

$$R = 0.1$$

Obteniéndose el siguiente resultado:

$$K = [32.22 \quad 16.67 \quad 14.51 \quad 7.50]$$

Para ensayarse estos valores, se construyó el siguiente modelo de simulación:

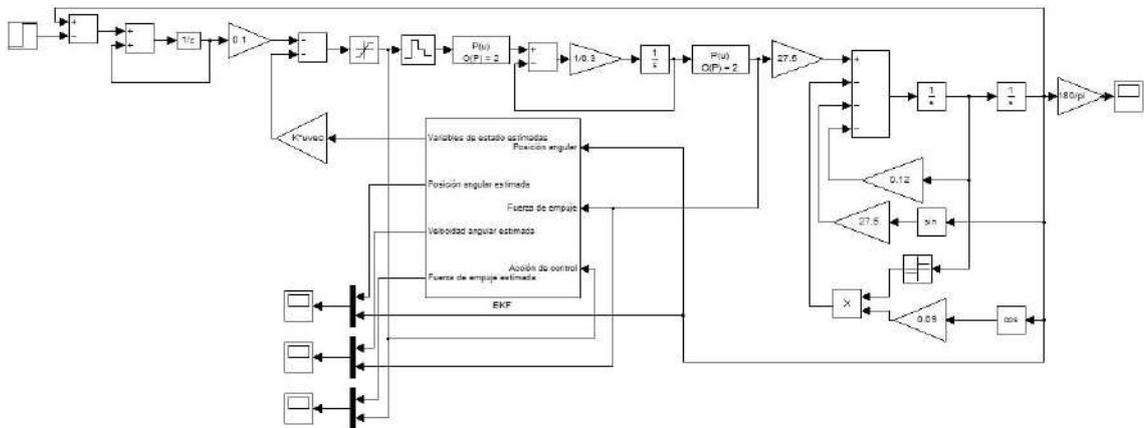


Figura 4.2.1. Modelo de simulación del control integral.

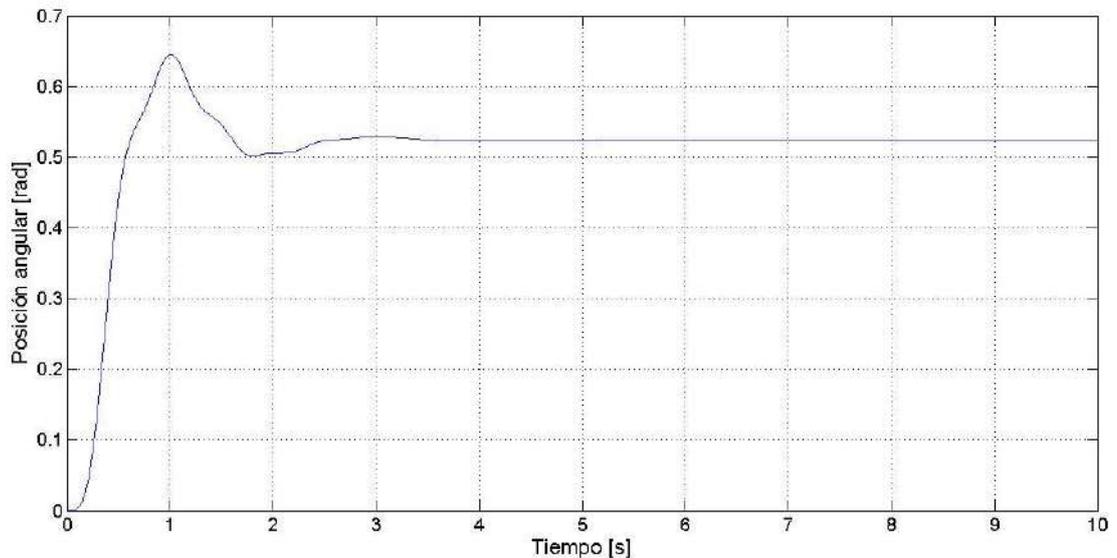


Figura 4.2.2. Respuesta del modelo de simulación de la figura 4.2.1 para una referencia de 0.5236 rad (30°).

Al implementarse estas ganancias en el modelo de simulación, el sistema se tornó inestable. Se especuló primeramente con que el control era muy agresivo, por lo que se disminuyeron los valores de ganancia del vector  $K$  de manera manual, pero sin encontrar solución al inconveniente.

Como ya se mencionó anteriormente en la sección 3.1, el sistema posee ruido en la medición, motivo por el cual se implementó el EKF. Considerando esto, se ajustaron los parámetros  $Q_{EKF}$  y  $R_{EKF}$  del estimador óptimo hasta tener una respuesta estable, resultando:

$$Q_{EKF} = \begin{bmatrix} 1 e - 7 & 0 & 0 \\ 0 & 3 e - 4 & 0 \\ 0 & 0 & 4 e - 7 \end{bmatrix}$$

$$R_{EKF} = \begin{bmatrix} 6 e - 6 & 0 \\ 0 & 1.4 e - 6 \end{bmatrix}$$

$$K = [0.1 \quad 5 \quad 1.8 \quad 1]$$

Una vez estabilizado el sistema en *Simulink*, se implementó el algoritmo en C, donde todos los parámetros tuvieron que ser nuevamente ajustados para obtener una respuesta estable. La constante que involucra la variable de estado integral tuvo que ser aumentada con el fin de obtener un tiempo de respuesta menor, mientras que las constantes de  $Q_{EKF}$  se tuvieron que incrementar algunos órdenes de magnitud para lograr la estabilidad del sistema:

$$Q_{EKF} = \begin{bmatrix} 1 e - 4 & 0 & 0 \\ 0 & 3 e - 1 & 0 \\ 0 & 0 & 4 e - 1 \end{bmatrix}$$

$$R_{EKF} = \begin{bmatrix} 6 e - 6 & 0 \\ 0 & 1.4 e - 6 \end{bmatrix}$$

$$K = [15 \quad 5 \quad 1.8 \quad 1]$$

A continuación se presenta la respuesta del sistema controlado con estos valores, ingresando la referencia ya no como una rampa, sino como un escalón para observar el comportamiento del sistema.

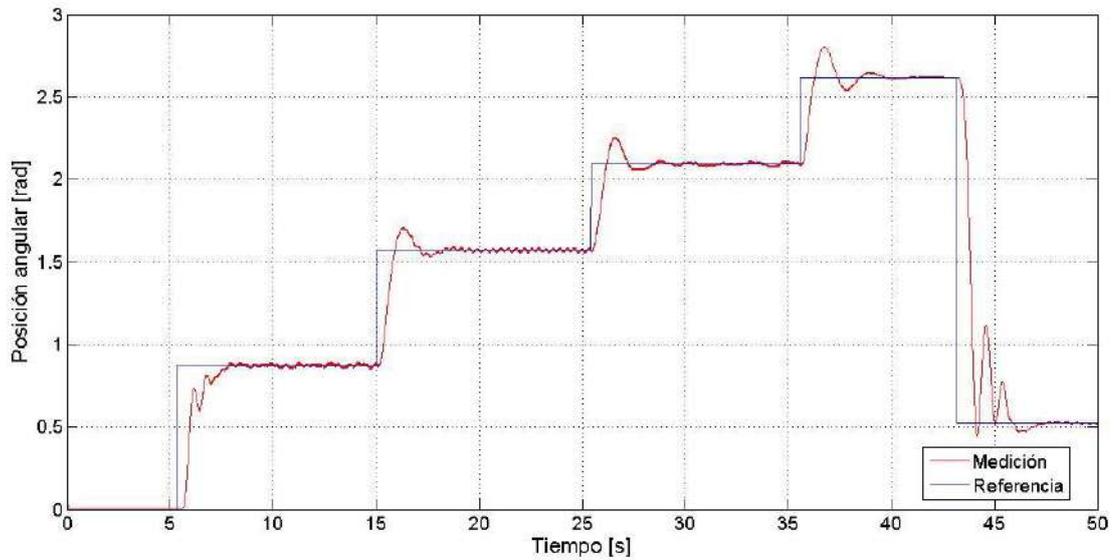


Figura 4.2.3. Respuesta del sistema con controlador integral por variables de estado.

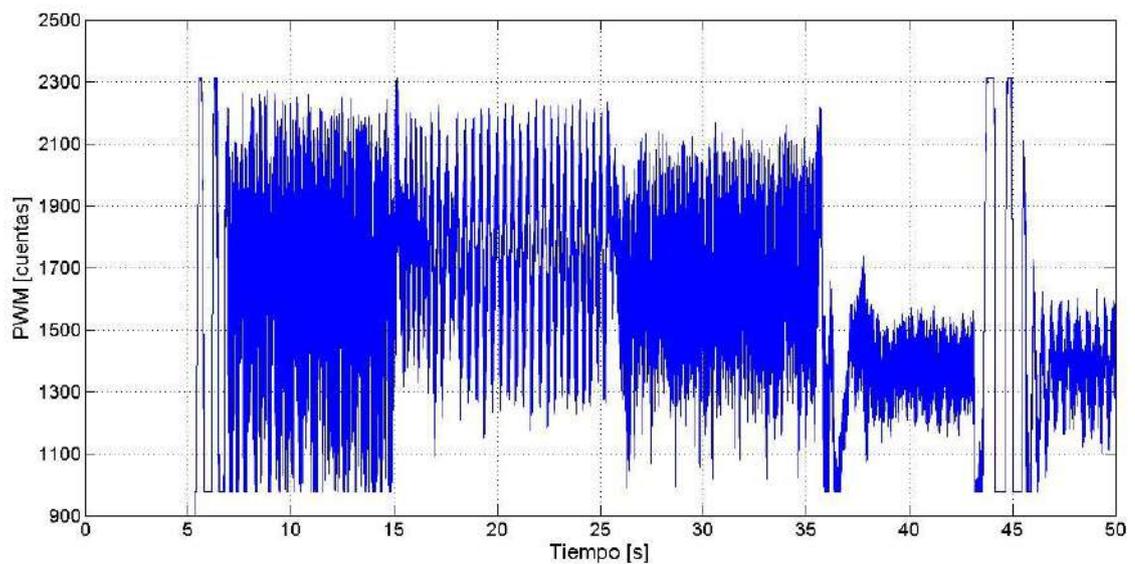


Figura 4.2.4. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Como puede verse, la respuesta no presenta error en estado estacionario. Sin embargo, la dinámica posee ahora sobre pico en el transitorio y oscilaciones alrededor del punto de establecimiento. Este sobre pico resulta ser un serio problema si se pretende alcanzar valores de posición cercanos a los  $180^\circ$ , ya que puede hacer que el sistema sobrepase este límite y ya no pueda regresar. Además, la acción de control es mucho más agresiva que en los controladores anteriores.

#### Calculo de controlador integral:

- Carpeta Matlab-Simulink
  - o Controlador Integral
    - controlador\_integral.m
    - FiltroK.m
    - simulacion\_control\_integral.mdl

#### Datos del controlador integral:

- Carpeta Matlab-Simulink
  - o Logs varios controladores
    - ControllIntegral.txt

#### Código en C:

- Carpeta Aeropendolo
  - o ControllIntegralKalman.cpp

### 4.3 Controlador integral por variables de estado con modelo de referencia

En la sección anterior (4.2) se hizo referencia al hecho de que, al ingresarse escalones amplios de referencia, el sistema podía volverse inestable. Analizando el cálculo de la integral realizado por la variable de estado  $x_I$ , como puede verse en la figura 4.3.1, en un primer instante el valor de esta integral es elevada, lo que al ser multiplicado por la ganancia produce una acción de control excesiva.

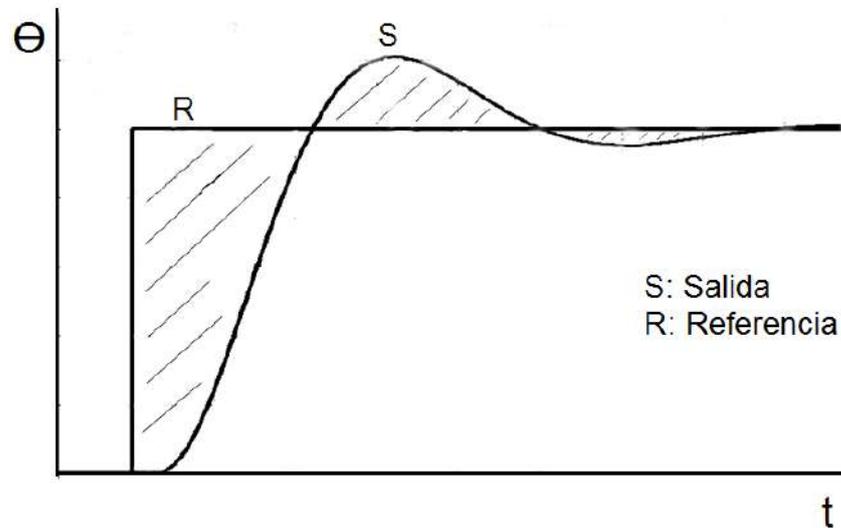


Figura 4.3.1. Integral del error entre la salida del sistema y la referencia.

Para solucionar esto, se planteó un nuevo algoritmo de control que en lugar de calcular  $x_I = \theta - \theta_{ref}$ , toma como referencia a ingresar al sistema, la salida de un modelo de primer orden definido como:

$$\frac{\theta_S}{\theta_{ref}} = \frac{1}{\tau \cdot s + 1} \rightarrow \theta_S = \theta_{ref} \cdot \frac{1}{\tau \cdot s + 1} \quad (4.3.1)$$

$$x_I = \theta - \theta_S \quad (4.3.2)$$

La figura 4.3.2 presenta una explicación gráfica que ilustra el objetivo buscado.

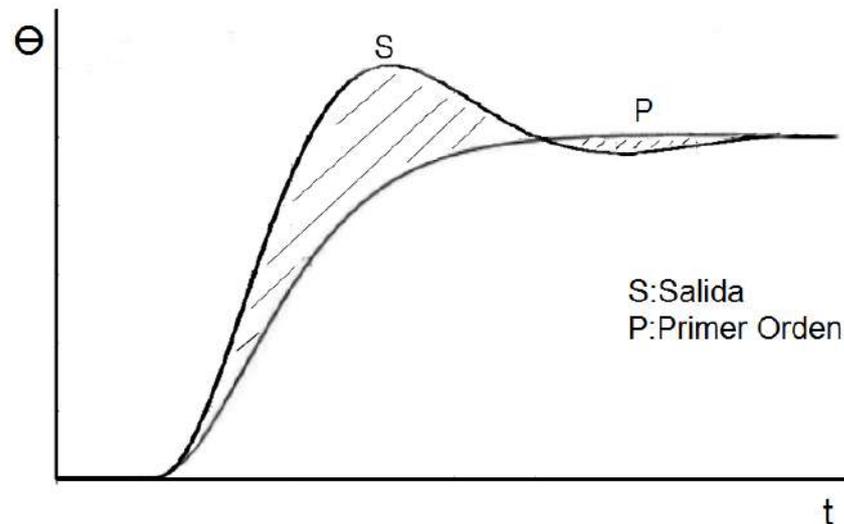


Figura 4.3.2. Integral del error entre la salida del sistema y el modelo de referencia.

Una vez obtenido el error de esta manera, las variables de estado y las ganancias utilizadas fueron las mismas a las calculadas anteriormente en la sección 4.2. A la hora de implementar este nuevo cálculo de error, se supuso  $\tau = 0.3$  s, pero al implementarlo no se observó una mejora sustancial en la respuesta. Por este motivo se ajustó dicho valor de manera iterativa hasta obtener una dinámica aceptable. Finalmente, el modelo de primer orden resultante es:

$$\theta_s = \theta_{ref} \cdot \frac{1}{0.7 \cdot s + 1}$$

La respuesta del sistema lograda se presenta en las figuras 4.3.3 y 4.3.4.

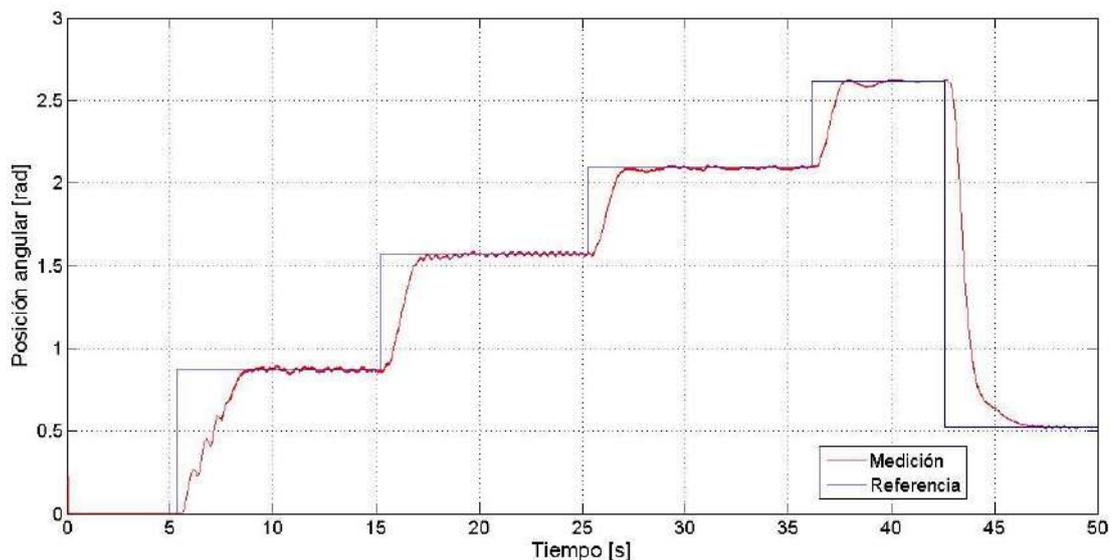


Figura 4.3.3. Respuesta del sistema con controlador integral por variables de estado con modelo de referencia.

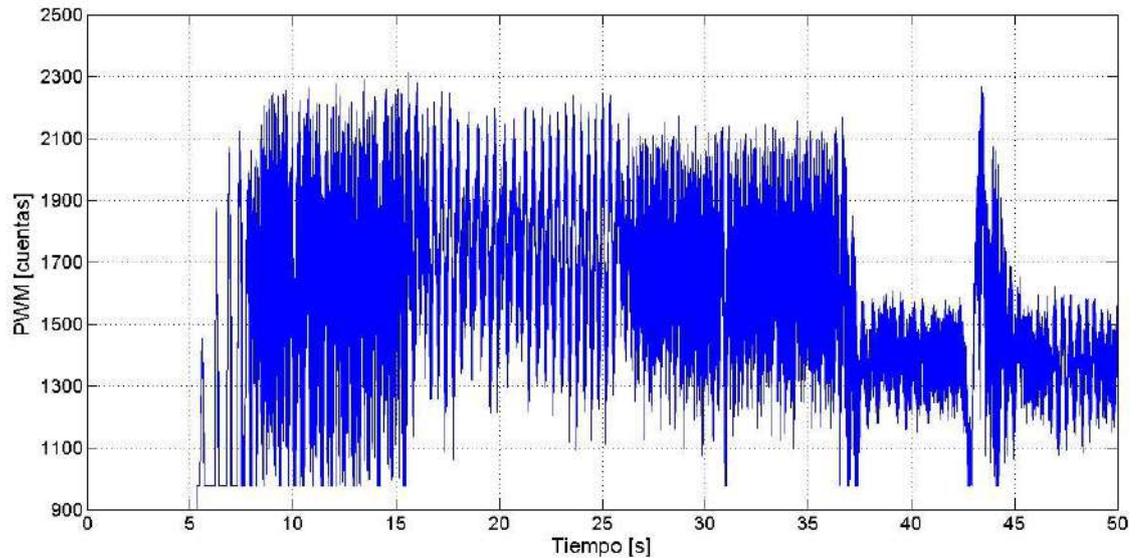


Figura 4.3.4. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Como puede observarse, el sobre pico en el transitorio de la respuesta fue eliminado, pero las oscilaciones en estado estacionario y la elevada acción de control siguen siendo parecidas al caso anterior. Para salvar esto, se planteó un controlador por linealización exacta.

Datos del controlador integral con modelo de referencia:

- Carpeta Matlab-Simulink
  - o Logs varios controladores
    - ControllIntegral2.txt

Código en C:

- Carpeta Aeropendulo
  - o ControllIntegral2.cpp

#### 4.4 Controlador por linealización exacta

El último controlador diseñado para el sistema aeropéndulo fue realizado mediante linealización exacta. Este método de control para sistemas no lineales consiste básicamente en transformar, parcial o totalmente, la dinámica no lineal del sistema en una dinámica lineal. Esto permite aplicar técnicas de control lineal de modo de obtener las especificaciones deseadas a lazo cerrado. Sin embargo, esto trae aparejado como principal inconveniente, el disponer de un modelo exacto.

El concepto de linealización exacta para transformar la dinámica no lineal contempla la realización de un cambio de variables de estado. Así, partiendo del modelo no lineal de la ecuación (1.5.4), pero sin tener en cuenta la fricción coulombiana, se obtuvo:

$$\dot{\theta} = \omega \quad (4.4.1)$$

$$\dot{\omega} = K_f \cdot F - K_\omega \cdot \omega - K_\theta \cdot \sin(\theta) \quad (4.4.2)$$

$$\dot{F} = u \cdot \frac{1}{\tau} - F \cdot \frac{1}{\tau} \quad (4.4.3)$$

Donde:

$$K_f = C2 = 27.5$$

$$K_\omega = C1 = 0.12$$

$$K_\theta = C2 = 27.5$$

$$\tau = 0.3$$

$u$  = Acción de control.

Siendo el cambio de variables de estado planteado el siguiente:

$$\hat{\theta} = \theta - \theta_r \quad (4.4.4)$$

$$\hat{\omega} = \omega \quad (4.4.5)$$

$$\zeta = F \cdot K_f - K_\theta \cdot \sin(\hat{\theta} + \theta_r) \quad (4.4.6)$$

Donde:

$\theta_r$  = Referencia de posición angular en radianes.

Por lo que al derivar estas tres últimas expresiones, se obtuvo:

$$\hat{\dot{\theta}} = \dot{\theta} - \dot{\theta}_r \quad (4.4.7)$$

$$\hat{\dot{\omega}} = \dot{\zeta} - \dot{\hat{\omega}} \quad (4.4.8)$$

$$\dot{\zeta} = \dot{F} \cdot K_f - K_\theta \cdot \cos(\hat{\theta} + \theta_r) \cdot \hat{\dot{\theta}} \quad (4.4.9)$$

Ahora bien, utilizando la expresión (4.4.3), se reescribió la ecuación (4.4.9) para obtenerse la relación:

$$\dot{\zeta} = u \cdot \frac{K_f}{\tau} - \frac{1}{\tau} \cdot \zeta + \frac{K_\theta}{\tau} \cdot \sin(\hat{\theta} + \theta_r) - K_\theta \cdot \cos(\hat{\theta} + \theta_r) \cdot \hat{\dot{\theta}} \quad (4.4.10)$$

Lo cual permitió definir la acción de control como:

$$u = \frac{\tau}{K_f} \cdot \left( \frac{1}{\tau} \cdot \zeta + \frac{K_\theta}{\tau} \cdot \sin(\hat{\theta} + \theta_r) - K_\theta \cdot \cos(\hat{\theta} + \theta_r) \cdot \hat{\omega} \right) + \frac{\tau}{K_f} \cdot \hat{u} \quad (4.4.11)$$

Donde  $\hat{u}$  es una nueva acción de control para el sistema:

$$\begin{aligned} \hat{\theta} &= \hat{\omega} \\ \hat{\omega} &= \zeta - K_\omega \cdot \hat{\omega} \\ \dot{\zeta} &= \hat{u} \end{aligned}$$

Así, la acción de control  $\hat{u}$  resultó en la realimentación de estado del nuevo sistema de variables de estado como:

$$\hat{u} = -K_1 \cdot \hat{\theta} - K_2 \cdot \hat{\omega} - K_3 \cdot \zeta$$

Finalmente, las matrices de la ecuación de estado de este nuevo sistema resultaron:

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -K_\omega & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Para calcular los valores de ganancia en la realimentación se utilizó control óptimo con la función *lqr* de *Matlab*, donde las matrices de pesos utilizadas fueron:

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

$$R = 0.01$$

Al realizar el cálculo del vector de ganancias  $K$ , los valores que se obtuvieron fueron los siguientes:

$$K = [100 \quad 43.62 \quad 9.39]$$

Esto fue ensayado en un modelo de simulación en el entorno de *Simulink* utilizando el diagrama presentado en la figura 4.4.1.

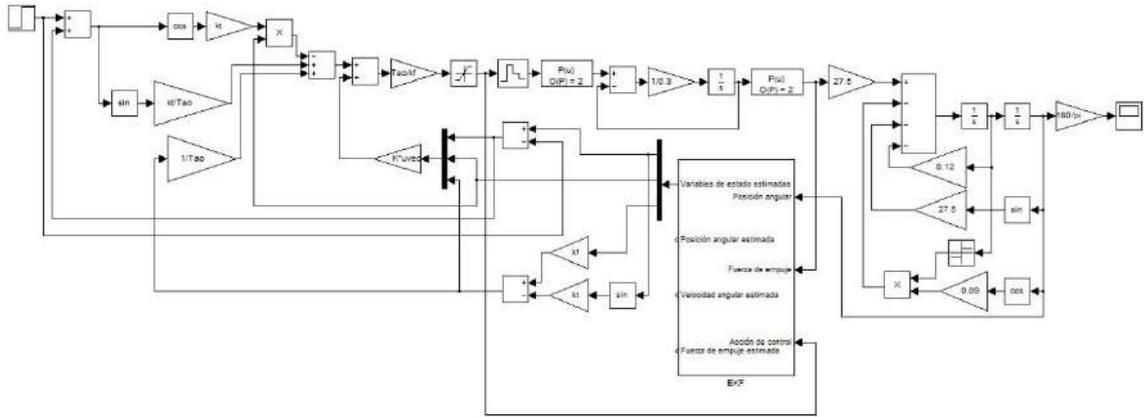


Figura 4.4.1. Modelo de simulación del control por linealización exacta.

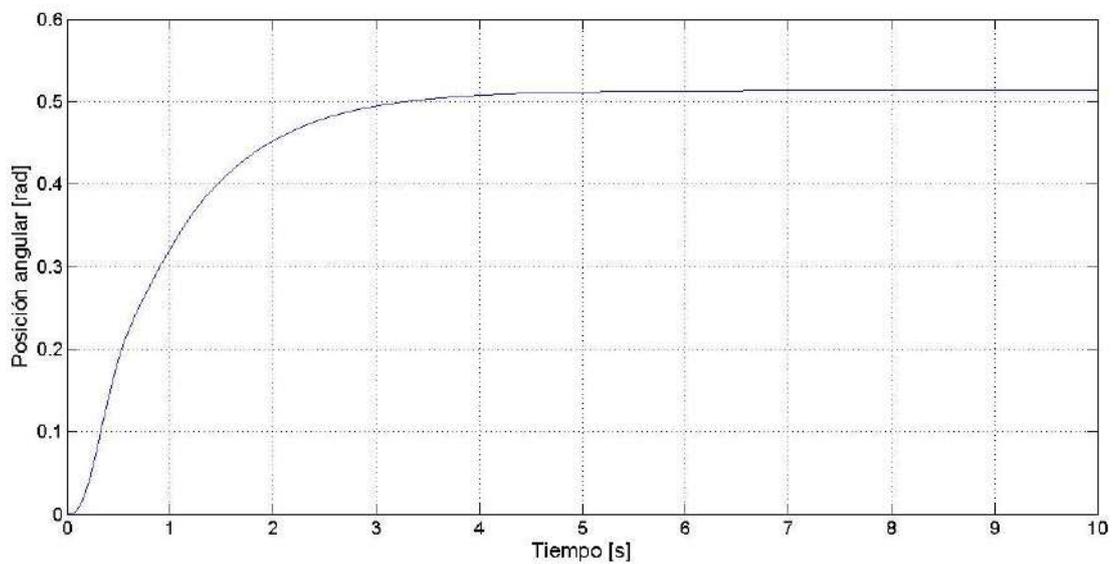


Figura 4.4.2. Respuesta del modelo de simulación de la figura 4.4.1 para una referencia de 0.5236 rad (30°).

Sin embargo, al implementarse estos valores en el algoritmo del sistema real, fueron ajustados manualmente de manera iterativa para obtener una mejor respuesta temporal, siendo los valores finales utilizados:

$$K = [114 \quad 46 \quad 9]$$

En las figuras 4.4.3 y 4.4.4 puede observarse la respuesta del sistema controlado. En este caso, al igual que durante la implementación del controlador integral, no se ingresó la referencia mediante una rampa, sino que se utilizaron escalones de posición angular.

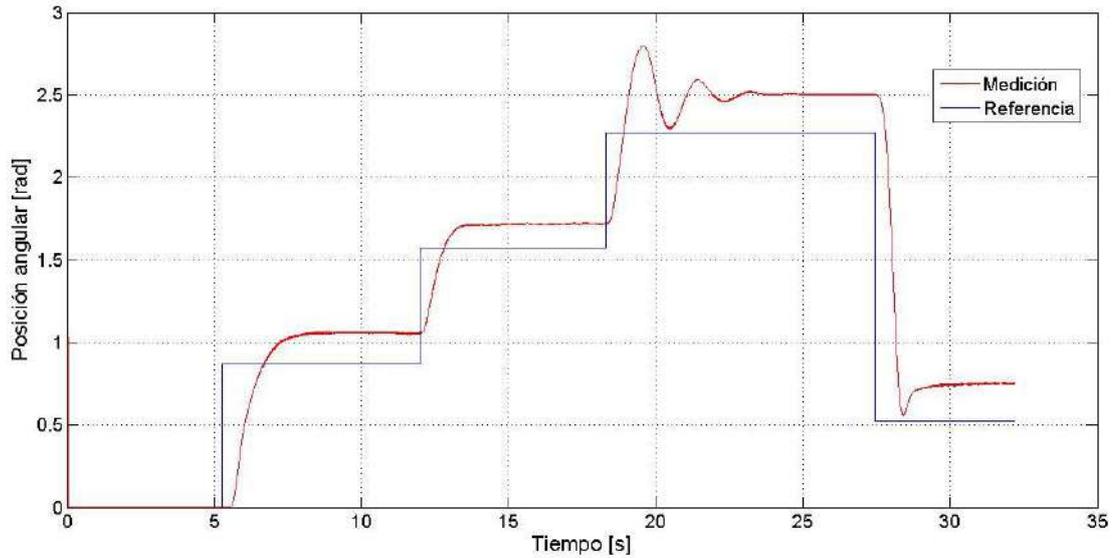


Figura 4.4.3. Respuesta del sistema con controlador por linealización exacta.

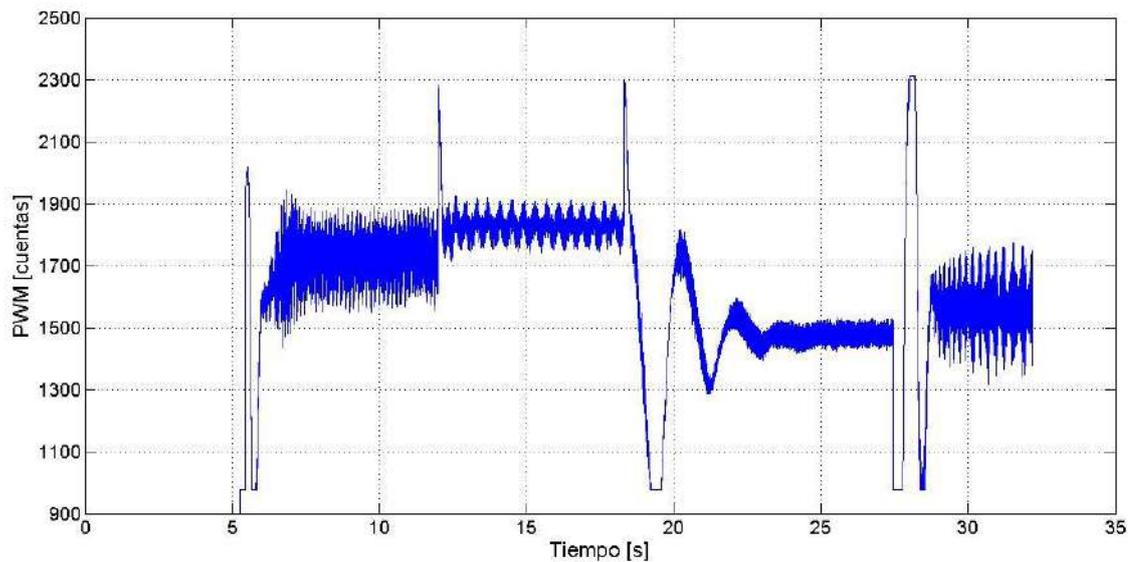


Figura 4.4.4. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Al analizar la respuesta se apreció primero que el sistema era estable con una muy buena respuesta dinámica, pero con error en estado estacionario. Esto último se debe a lo aclarado en el inicio de esta sección: el modelo debe ser exacto, y en este caso no es muy preciso el modelado realizado anteriormente, además de que al momento de realizar los cálculos no se tuvo en consideración la fuerza coulombiana.

Calculo de controlador por linealización exacta:

- Carpeta Matlab-Simulink
  - o Linealización Exacta
    - linealizacion\_exacta\_sinintegral.m
    - FiltroK.m
    - simulacion\_linealizacion\_exacta.mdl

Datos del controlador por linealización exacta:

- Carpeta Matlab-Simulink
  - o Logs varios controladores
    - LinealizacionExacta.txt

Código en C:

- Carpeta Aeropendolo
  - o Linealizacionexacta.cpp

#### 4.5 Controlador por linealización exacta y control integral con modelo de referencia

Al ver que el diseño anterior traía aparejado error en estado estacionario, se propuso realizar un controlador por linealización exacta y control integral con modelo de referencia, explicado en la sección 4.3, donde la variable integral se tomó igual a  $\hat{\theta}$ . De esta manera, el nuevo sistema de variables de estado a controlar resultó:

$$\begin{aligned}\hat{\theta} &= \hat{\omega} \\ \dot{\hat{\omega}} &= \zeta - K_{\omega} \cdot \hat{\omega} \\ \dot{\zeta} &= \hat{u} \\ \dot{x}_I &= \hat{\theta}\end{aligned}$$

Así, las matrices de la ecuación de estado de este nuevo sistema resultaron:

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -K_w & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

En este caso, se utilizó nuevamente el comando *lqr* de *Matlab* para obtener el vector de ganancias  $K$ , siendo las matrices de peso las siguientes:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

$$R = 0.01$$

Los valores que se obtuvieron fueron:

$$K = [114.05 \quad 46.04 \quad 9.63 \quad 31.62]$$

Y nuevamente, estas ganancias fueron ajustadas manualmente, siendo los valores finales utilizados:

$$K = [114 \quad 46 \quad 9 \quad 32]$$

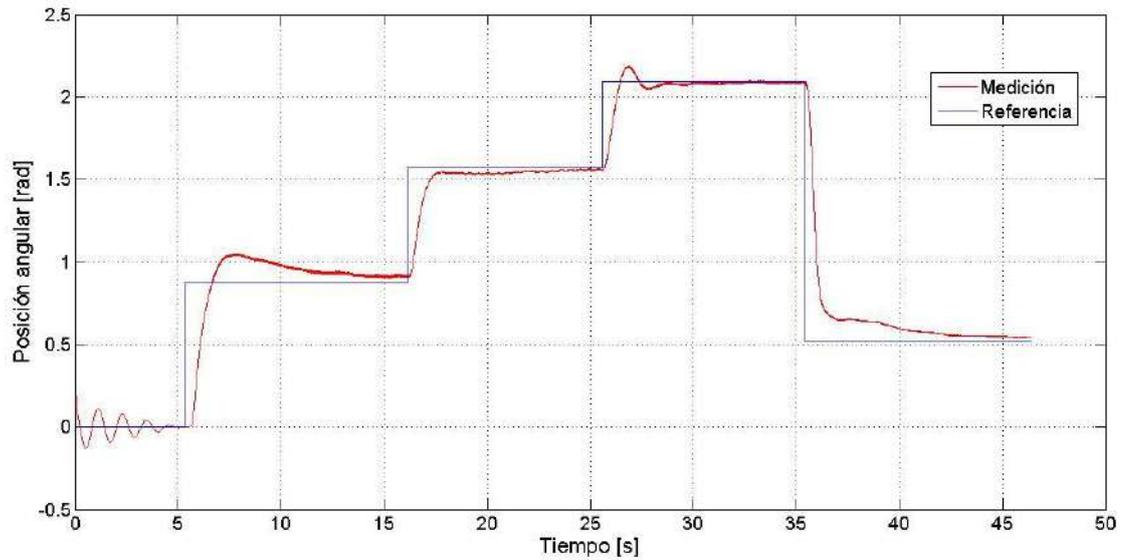


Figura 4.5.1. Respuesta del sistema con controlador por linealización exacta y control integral con modelo de referencia.

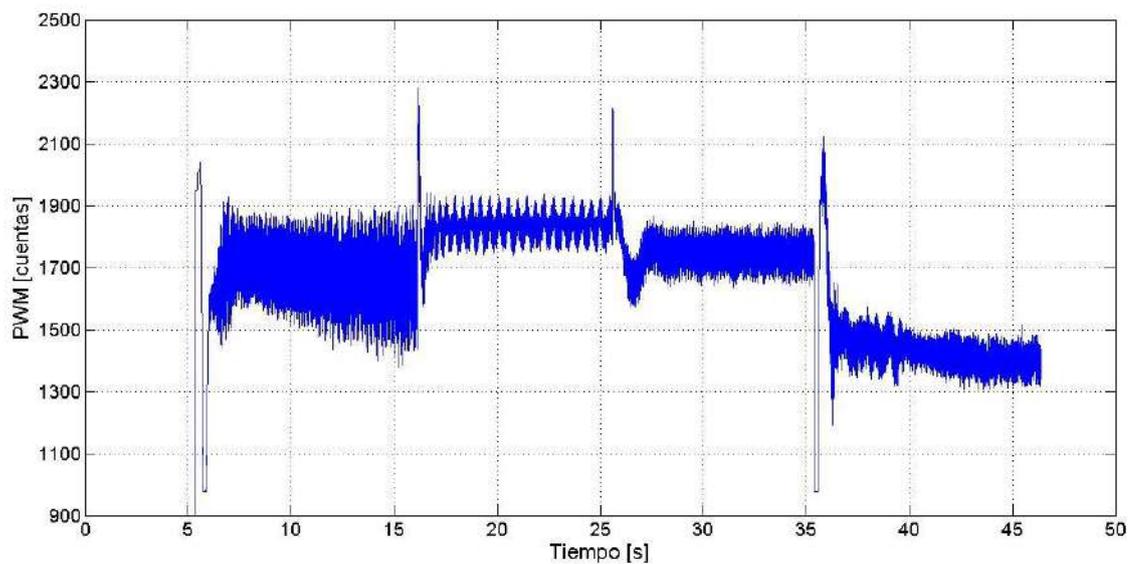


Figura 4.5.2. Acción de control sobre el sistema de propulsión en cuentas de PWM.

Al analizar la respuesta obtenida, puede verse claramente que el error en estado estacionario es nulo. Sin embargo, la dinámica del sistema en algunos escalones se ve empeorada, presentando picos donde antes no había, e incluso en estado estacionario parece presentar pequeñas oscilaciones.

Datos del controlador por linealización exacta con control integral por modelo de referencia:

- Carpeta Matlab-Simulink
  - Logs varios controladores
    - LinealizacionExactaIntegral.txt

Código en C:

- Carpeta Aeropendolo
  - LinealizacionexactaIntegral.cpp

## Conclusiones

Si bien a lo largo de esta práctica se obtuvieron distintas conclusiones que llevaron a la toma de decisiones en cada tarea desarrollada, es importante resaltar aquellas que llevaron a cumplir los objetivos planteados.

En la primera etapa del proyecto se llevó a cabo el modelado e identificación del sistema a controlar. El estudio del comportamiento físico del aeropéndulo permitió definirlo como un sistema “caja gris” capaz de ser modelado según sus ecuaciones principales. Por su parte, el reconocimiento de que variables de dicho sistema podían medirse permitió la identificación de aquellas que no eran accesibles, como la fuerza ejercida por la hélice del sistema de propulsión. A su vez, el análisis de estas ecuaciones llevó a la conclusión de la inestabilidad del sistema en aquellas posiciones por encima de los  $90^\circ$ , factor de suma importancia al momento de desarrollar el primer controlador.

Durante la identificación del sistema de propulsión, la obtención de la relación entre cuentas de PWM, enviadas al driver, y la fuerza ejercida por la hélice, permitió más adelante poder trabajar directamente con esta última como acción de control.

En el caso de la identificación del sistema péndulo, si bien fue sumado el efecto de la fricción coulombiana con el objetivo de mejorar el modelo, se tomó la decisión de no considerar la fricción estática en la simulación. Esto se debió a que el ajuste logrado resultaba ya lo suficientemente aproximado como para continuar con los demás pasos del modelado e identificación del sistema completo.

Quizás la conclusión más relevante de esta primera etapa es que, una vez acoplados ambos sistemas y puesto a prueba el modelo de simulación, frente a una señal “pseudo-random”, pudo observarse claramente que dicho modelo se encontraba lejos de ser perfecto. Aun así, se lo utilizó durante todo el desarrollo del proyecto, debido a que presentaba el mejor ajuste ante pequeños escalones, siendo estos los mayormente empleados durante las primeras pruebas de cada controlador diseñado.

La implementación del filtro de Kalman fue un punto más que interesante en el desarrollo del proyecto. El hecho de utilizar un estimador óptimo, una vez llevado a la práctica y probado su funcionamiento, permitió entender conceptos teóricos adquiridos durante el tiempo de cursada. Se verificó el papel que juegan las matrices de peso  $Q$  y  $R$ , donde primeramente se les dieron valores y luego se ajustaron teniendo en cuenta una situación de compromiso, entre obtener una señal con menor ruido y tener offset. De todas maneras, se redujo el ruido de las variables de estado, lo que permitió la implementación del primer controlador.

Durante la etapa de control se propusieron y utilizaron distintas alternativas en las cuales todas cumplieron con el objetivo principal, controlar el sistema en todo su rango operable con un solo controlador. De todos modos, si bien el objetivo se logró, no todos tuvieron el mismo comportamiento: algunos presentaron una buena dinámica, pero error en estado estacionario, mientras que otros lograron anular dicho error a coste de una peor dinámica en la respuesta.

El primer controlador implementado, si bien fue estable, tenía una respuesta muy oscilatoria, la cual se debía al ruido en la medición. Una vez implementado el filtro de Kalman, el mismo controlador cambió radicalmente su respuesta, disminuyendo sus

oscilaciones pero manteniendo error en estado estacionario. En el caso del controlador PD en la realimentación se obtuvo una respuesta similar pero con menores oscilaciones (aun con error en estado estacionario.)

El controlador integral permitió seguir la referencia sin error, pero presentando sobre pico y oscilaciones. El controlador integral con modelo de referencia permitió eliminar este sobre pico y obtener mejor dinámica para escalones de referencias de más de  $100^\circ$ .

La implementación del controlador por linealización exacta mejoró la respuesta dinámica, pero con error en estado estacionario, problema solucionado al utilizar linealización exacta con control integral.

De todos los controladores diseñados, son de destacar las respuestas logradas por el controlador integral con modelo de referencia, y el controlador por linealización exacta con control integral, ambos con buena dinámica y sin error en estado estacionario.

Como conclusión final, nos encontramos frente a un proyecto de características diferentes a las que normalmente nos enfrentamos. Al ser un sistema no lineal, y sumados los problemas de ruido en las mediciones, llevó a emplear nuevas técnicas de control y estimación de variables que no habían sido cubiertas en cursos anteriores de control, como lo fue linealización exacta y filtro de Kalman.

## Bibliografía

- Philips, Charles L (1995): *Digital Control System Analysis and design*.
- Ogata, Katsuhiko (1996): *Sistemas de Control en Tiempo Discreto, Segunda Edición*.
- Franklin, Gene. F, J. David, Powell, Abbas, Emami-Naeini (2002): *Feedback Control of Dynamic Systems*.
- Franklin, Gene. F, J. David, Powell, Michael. L, Workman (1998): *Digital Control of Dynamic Systems*.
- MATLAB Documentation. The MathWorks, Inc (1994-2016). Disponible en: <http://www.mathworks.com/help/index.html>.
- Kalman Filter. Wikimedia Foundation, Inc. Disponible en: [https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter).
- Extended Kalman Filter (EKF). Wikimedia Foundation, Inc. Disponible en: [https://en.wikipedia.org/wiki/Extended\\_Kalman\\_filter](https://en.wikipedia.org/wiki/Extended_Kalman_filter).
- Teoría de Control II, Carpeta Teórica. Instituto Balseiro (2016).

# Anexos

## Anexo 1

Durante el primer ensayo realizado sobre el aeropéndulo, el soporte de este se encontraba apoyado sobre una mesa, de modo que debajo de todo el sistema había una superficie plana a la misma altura que la base de la estructura. Una vez levantados los datos y graficados para observar el resultado obtenido, se apreciaron oscilaciones considerables de la posición entre  $30^\circ$  y  $50^\circ$ . Se supuso que estas podrían ser producidas por el efecto suelo, un fenómeno aerodinámico que sucede cuando un cuerpo con una diferencia de presiones entre la zona que hay por encima de él y por debajo (en este caso la hélice), al estar muy cerca de una superficie provoca alteraciones en el flujo del aire. El fenómeno puede observarse en la figura A.1.2.

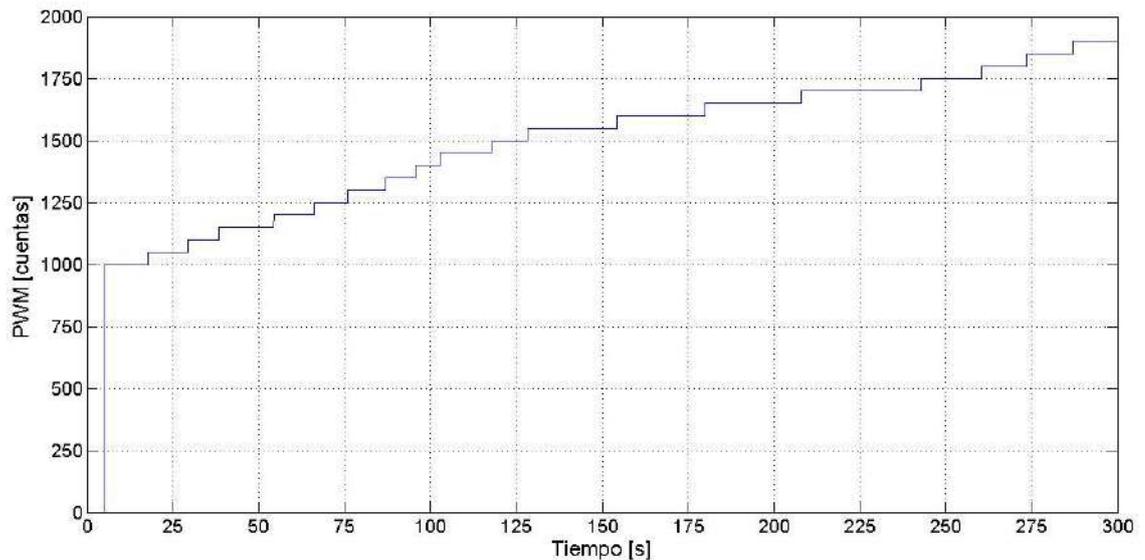


Figura A.1.1. Secuencia de escalones de PWM en cuentas.

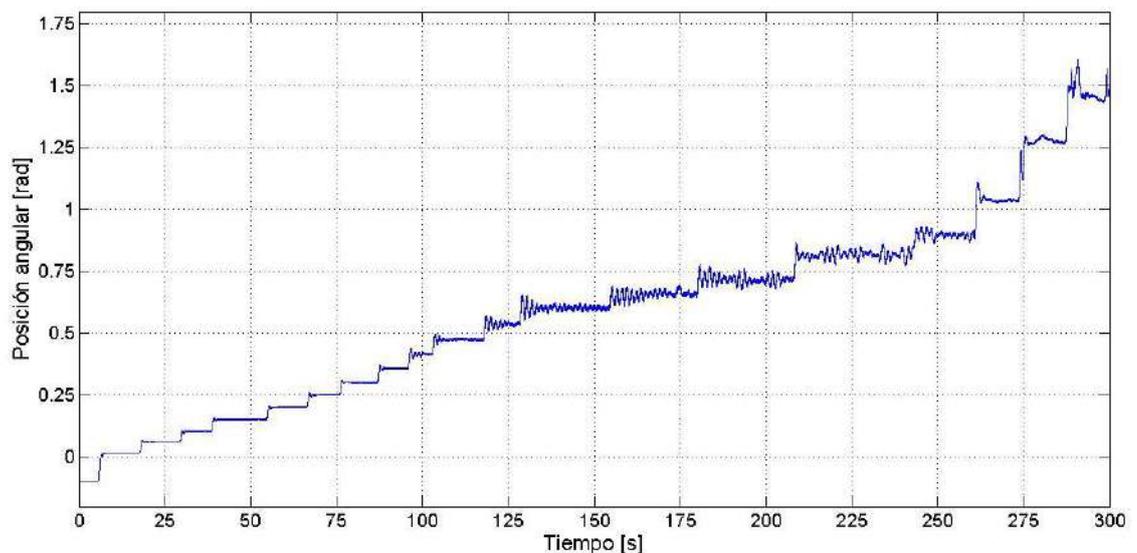


Figura A.1.2. Posición angular del sistema

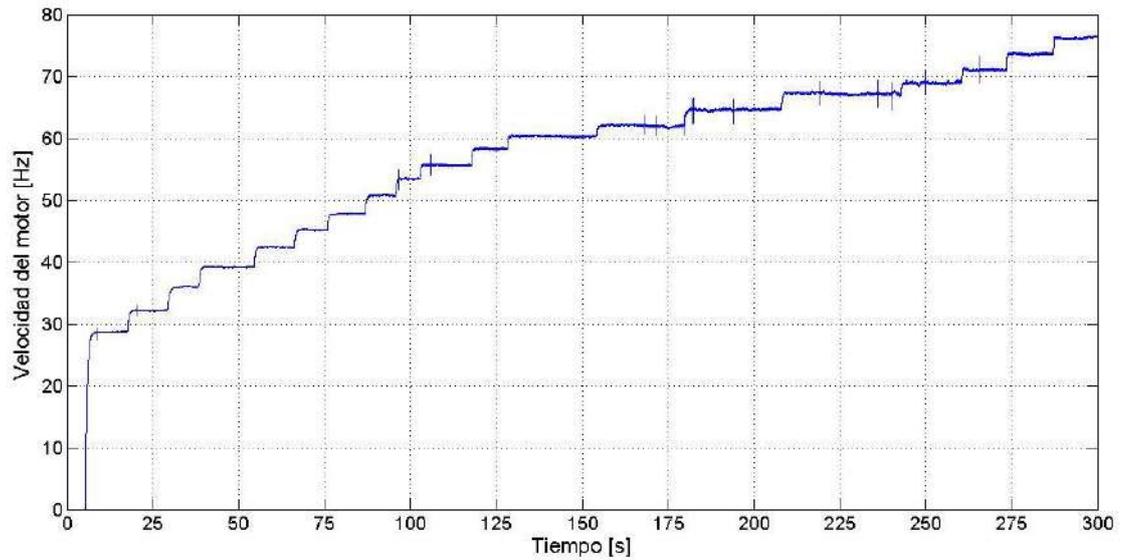


Figura A.1.3. Velocidad del motor.

Para solucionar esto, se repitió el ensayo con el soporte del péndulo apoyado en los extremos de dos mesas separadas, de manera de dejar un hueco debajo de este y así disminuir el efecto suelo. Desde ese momento, todo ensayo realizado sobre el sistema se realizó de esta manera, así como las pruebas de funcionamiento del sistema controlado.

## Anexo 2

Los valores obtenidos del ajuste por mínimos cuadrados de la ecuación (1.4.2) fueron:

$$C1 = 0.1175$$

$$C2 = 27.4593$$

$$C3 = 0.0085$$

Estos valores fueron probados en simulación comparando con las mediciones del ensayo real del péndulo con oscilaciones libres, cuyo resultado fue:

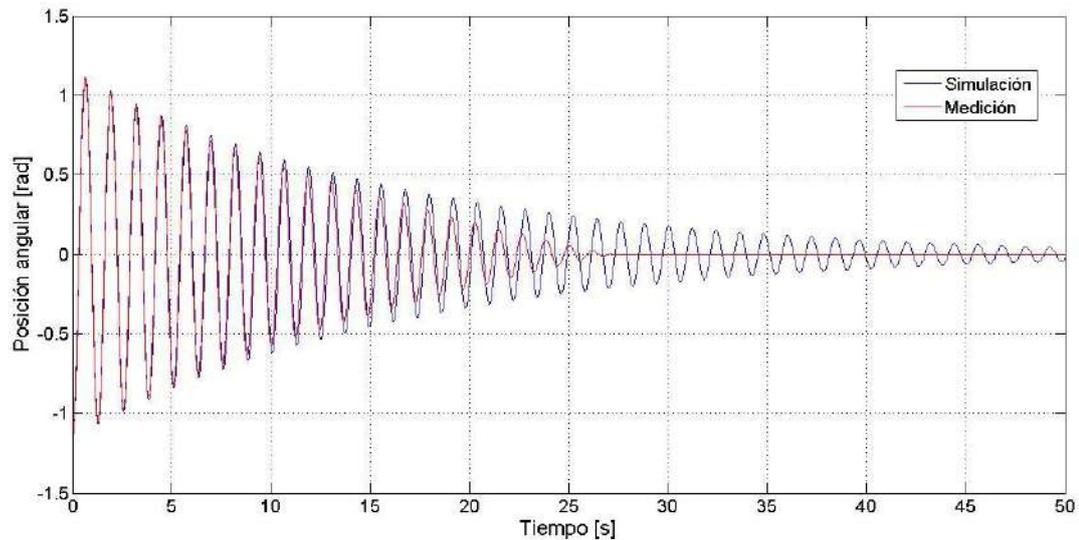


Figura A.2.1. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente -1.16 rad.

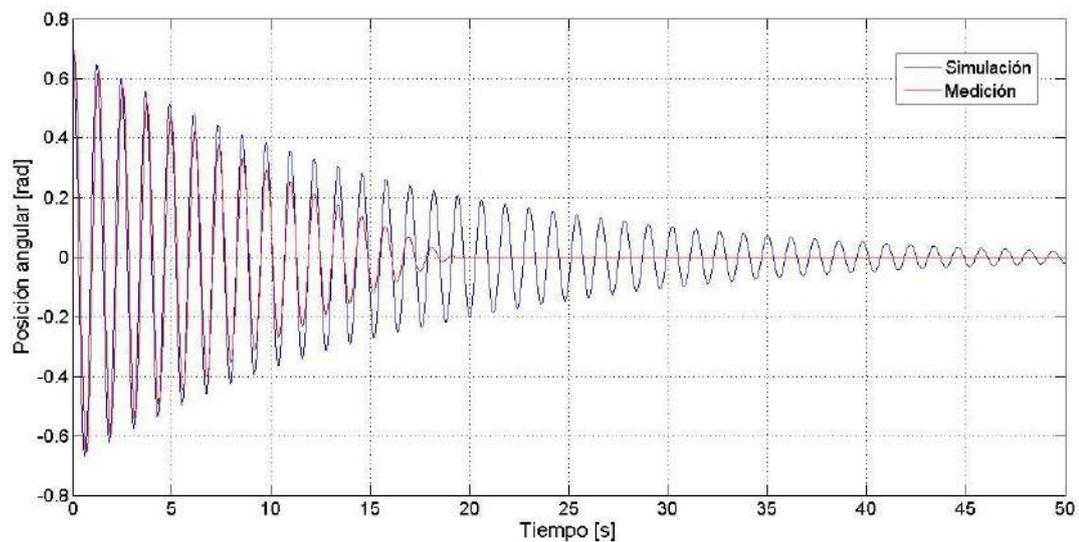


Figura A.2.2. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente 0.70 rad.

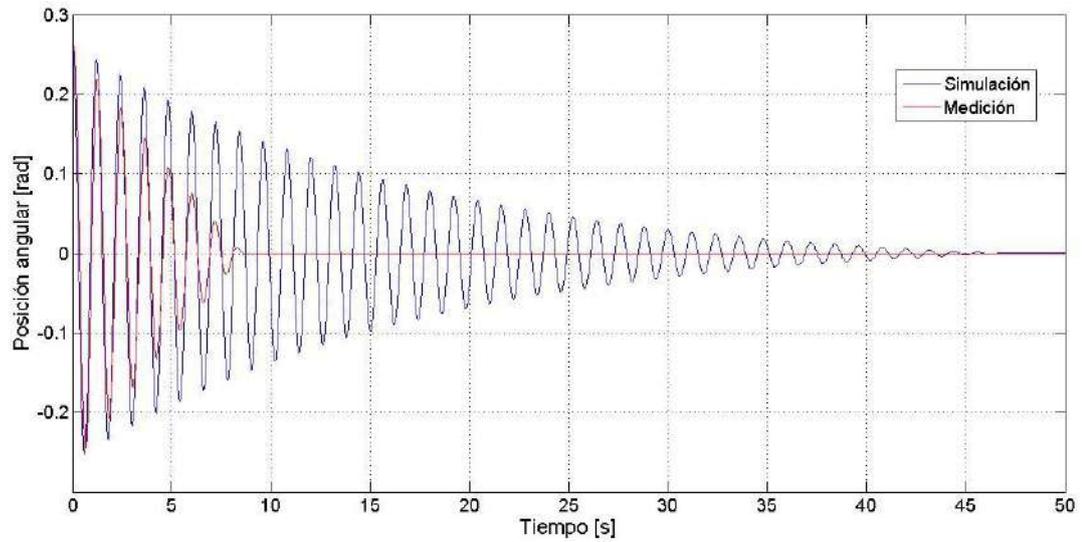


Figura A.2.3. Respuesta del modelo frente a la medición de la posición angular del péndulo para un ángulo inicial de aproximadamente 0.26 rad.

Puede observarse que la respuesta en los instantes iniciales es similar pero hacia el final de las oscilaciones la simulación se aleja de la respuesta real. Por este motivo, se optó por continuar con el ajuste de manera manual, partiendo de lo ya calculado.

### Anexo 3

A continuación se presenta la simulación de cada una de estas plantas ante un escalón de 50 cuentas de PWM partiendo del punto de equilibrio respectivo:

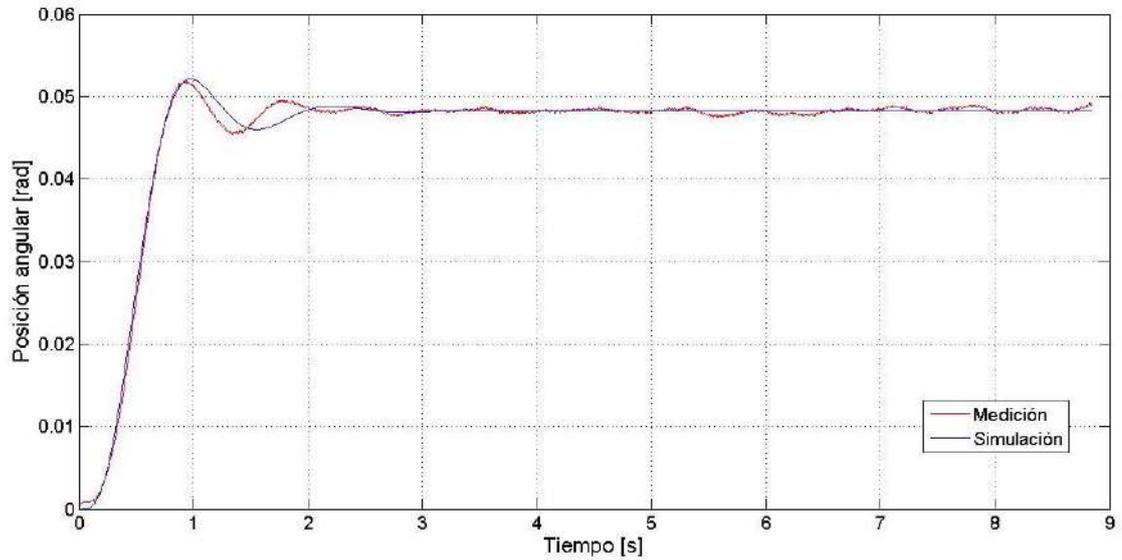


Figura A.3.1. Ajuste logrado por  $G_1(s)$  para un escalón de 1200 a 1250 cuentas de PWM.

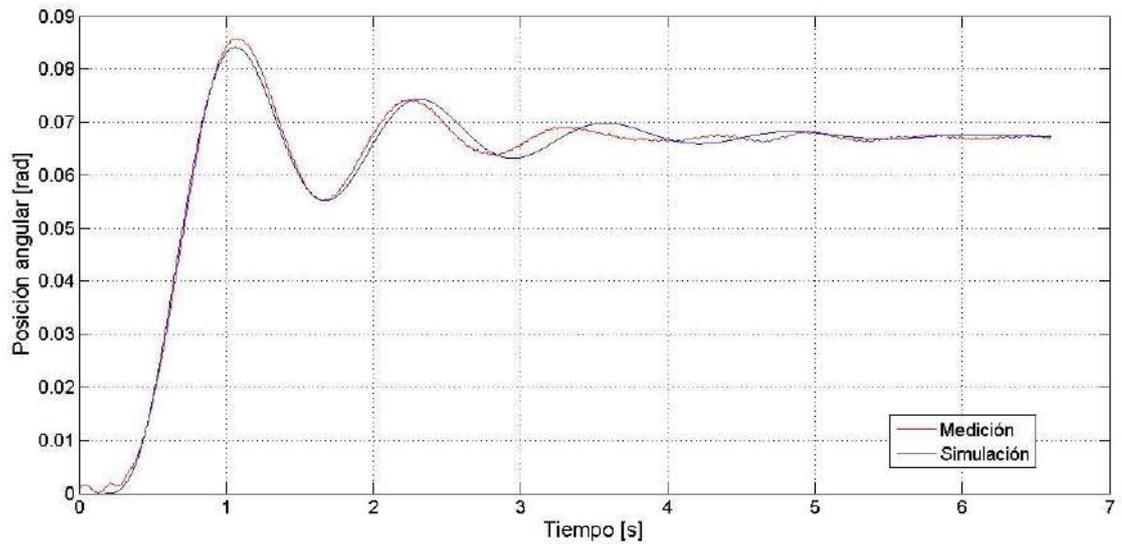


Figura A.3.2. Ajuste logrado por  $G_2(s)$  para un escalón de 1450 a 1500 cuentas de PWM.

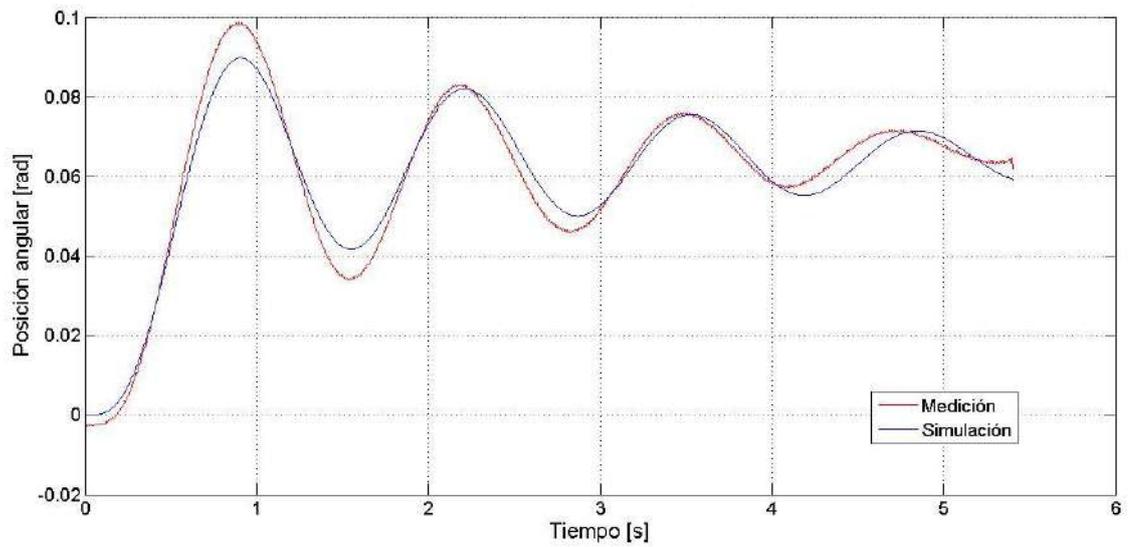


Figura A.3.3. Ajuste logrado por  $G_3(s)$  para un escalón de 1350 a 1400 cuentas de PWM.

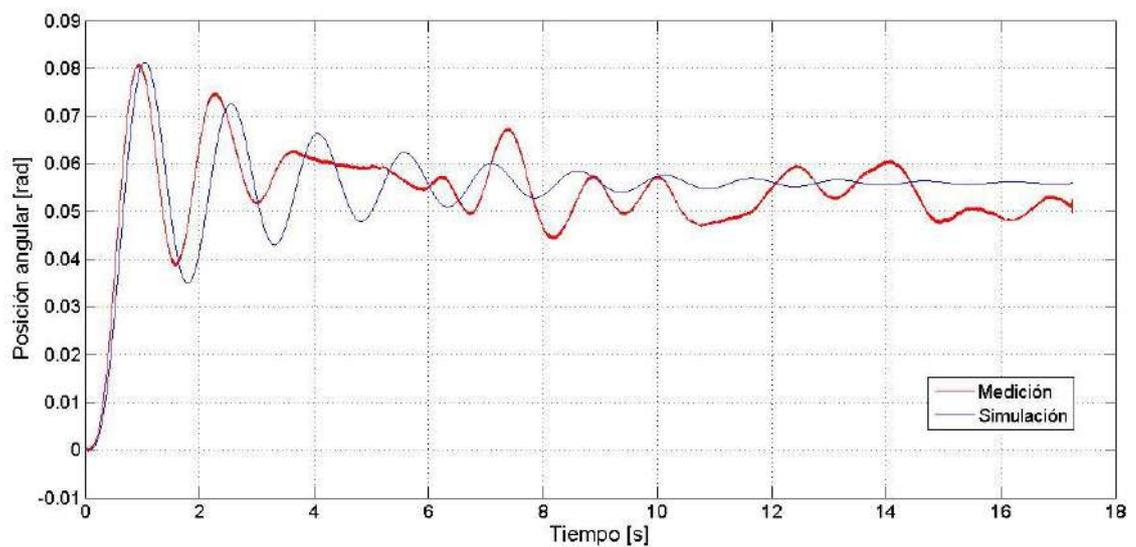


Figura A.3.4. Ajuste logrado por  $G_4(s)$  para un escalón de 1550 a 1600 cuentas de PWM.

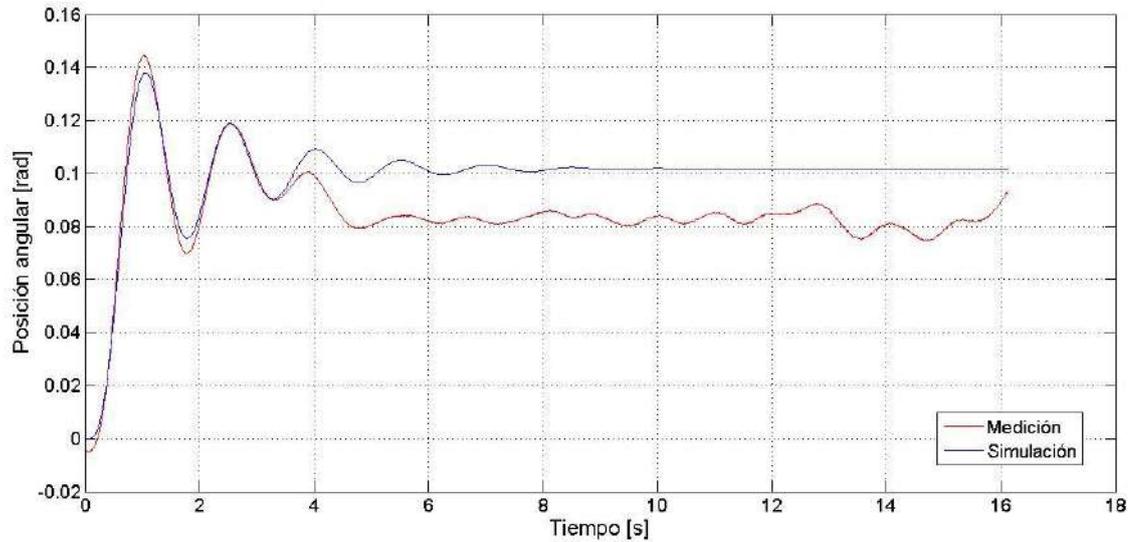


Figura A.3.5. Ajuste logrado por  $G_5(s)$  para un escalón de 1650 a 1700 cuentas de PWM.

Una vez realizado estas comprobaciones, se empleó la siguiente simulación para decidir el rango útil de cada linealización:

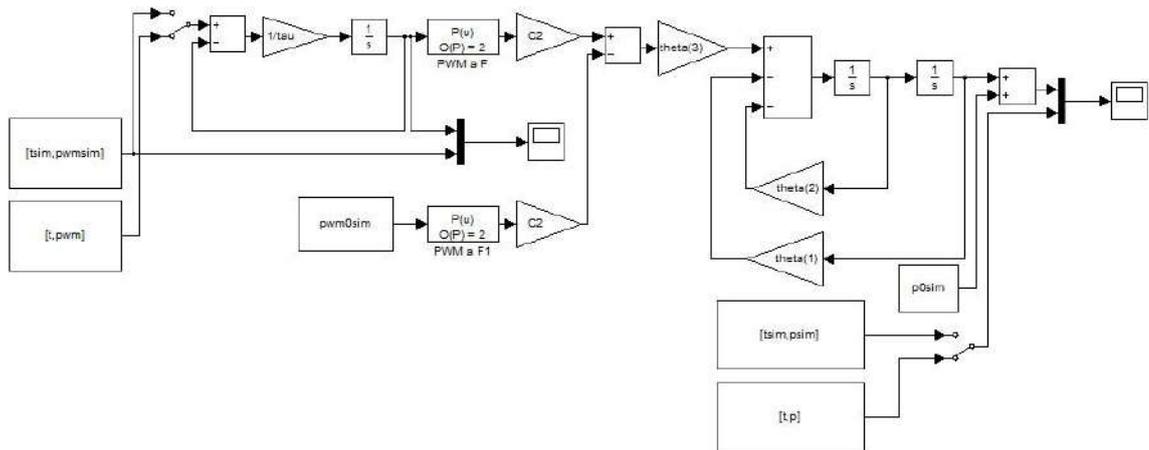


Figura A.3.6. Modelo de simulación del sistema aeropéndulo linealizado.

Este modelo de simulación fue construido con dos posibles modos de ejecución, pudiéndose elegir entre uno y otro por medio de los conmutadores: por un lado, la ejecución de un solo escalón de 50 cuentas de PWM partiendo del punto de equilibrio correspondiente, y por otro lado, la ejecución de la secuencia completa de escalones de PWM. De esta manera, se pudo seleccionar el rango útil de cada linealización, resultados que ya fueron expuestos en la tabla 2.1.1 y que pueden comprobarse en las siguientes figuras:

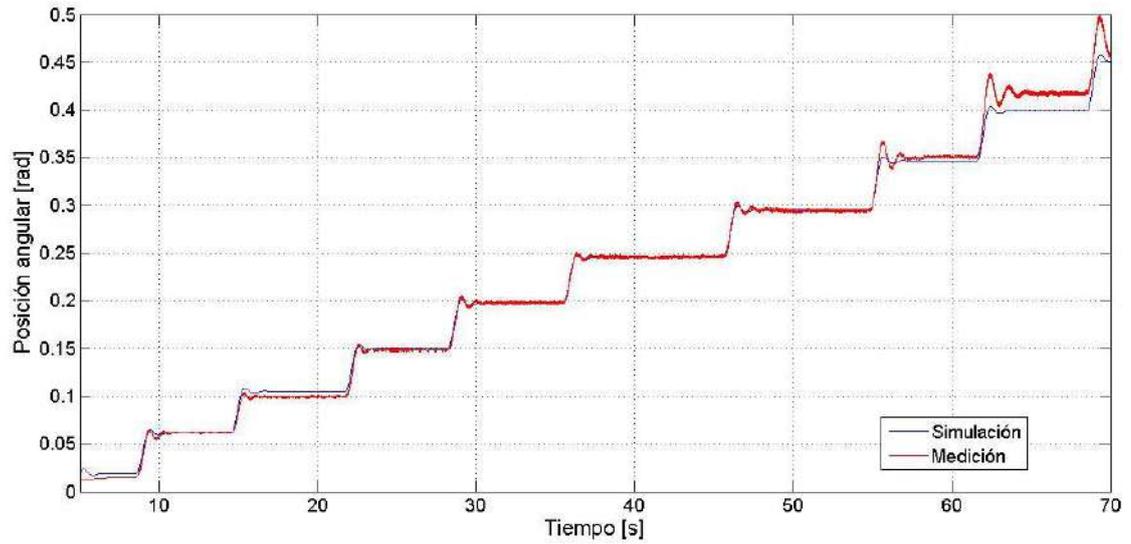


Figura A.3.7. Rango útil de la planta  $G_1(s)$ , de 1000 a 1300 cuentas de PWM.

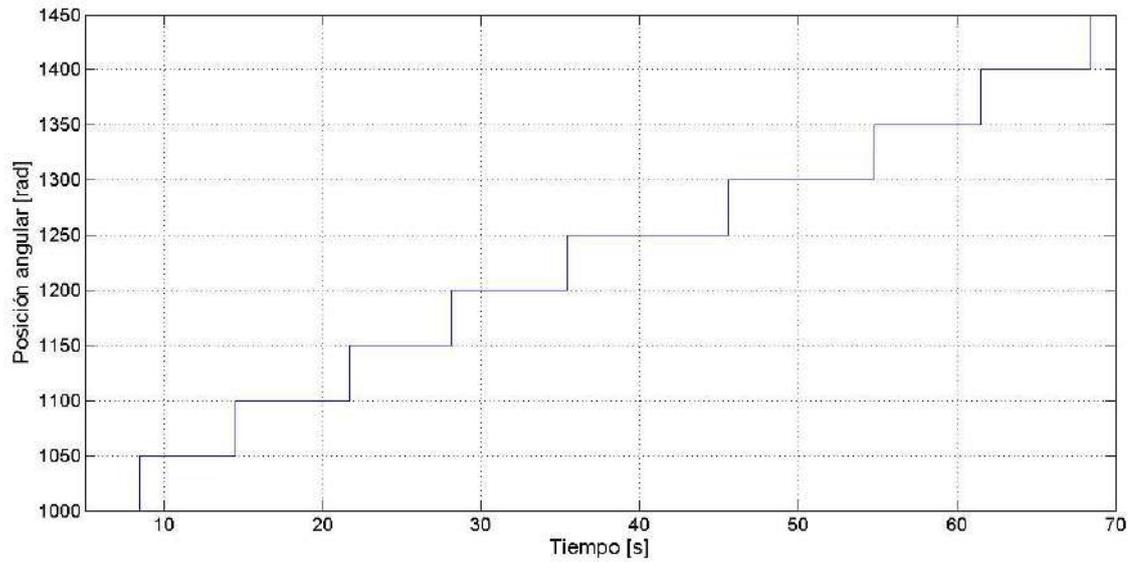


Figura A.3.8. Secuencia de escalones de PWM correspondiente a la figura anterior.

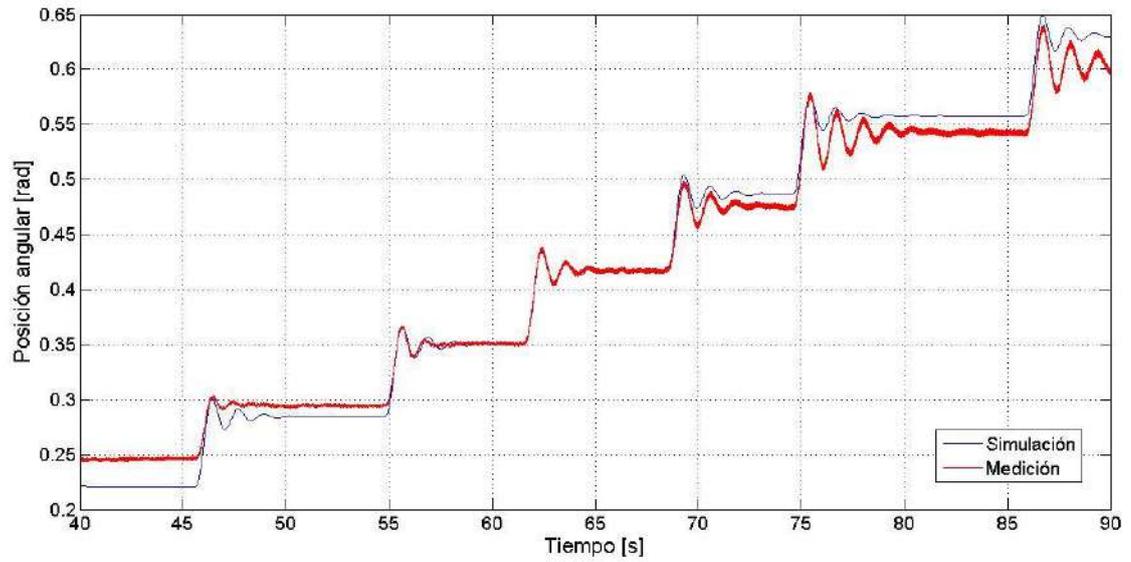


Figura A.3.9. Rango útil de la planta  $G_2(s)$ , de 1300 a 1450 cuentas de PWM.

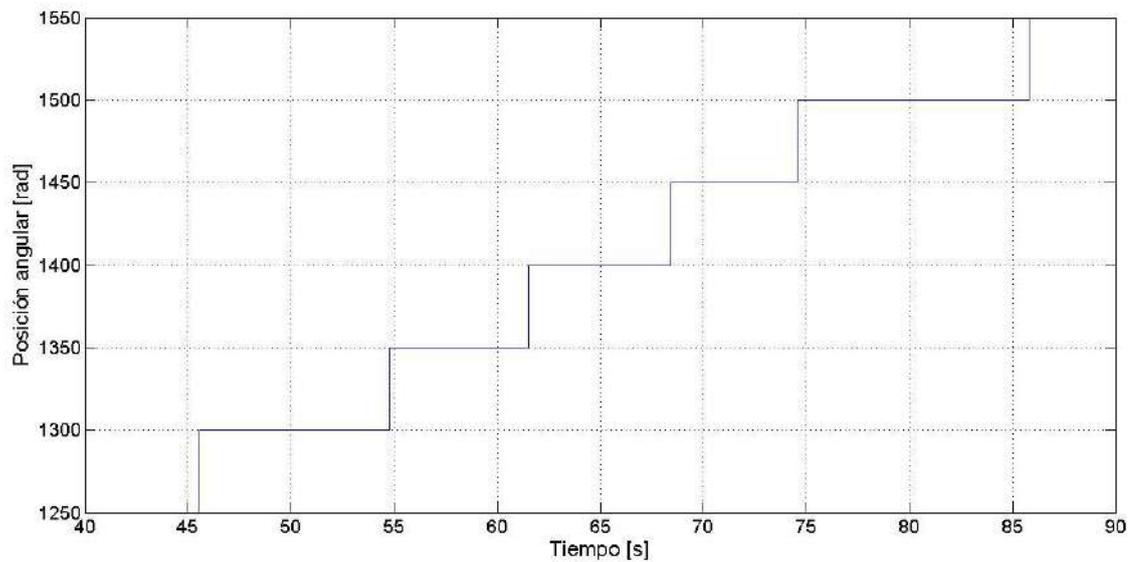


Figura A.3.10. Secuencia de escalones de PWM correspondiente a la figura anterior.

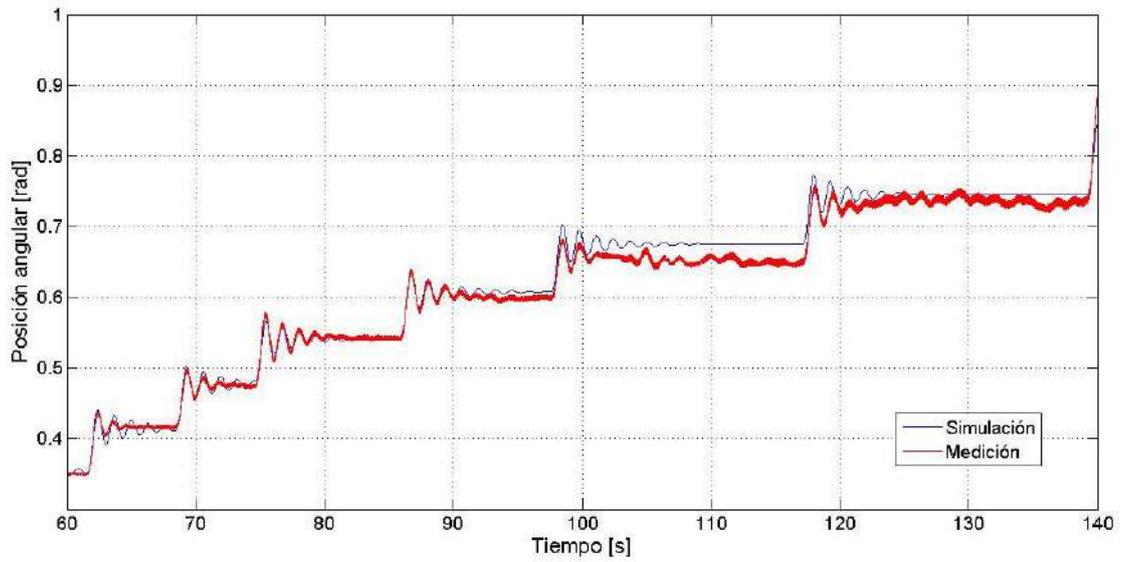


Figura A.3.11. Rango útil de la planta  $G_3(s)$ , de 1400 a 1600 cuentas de PWM.

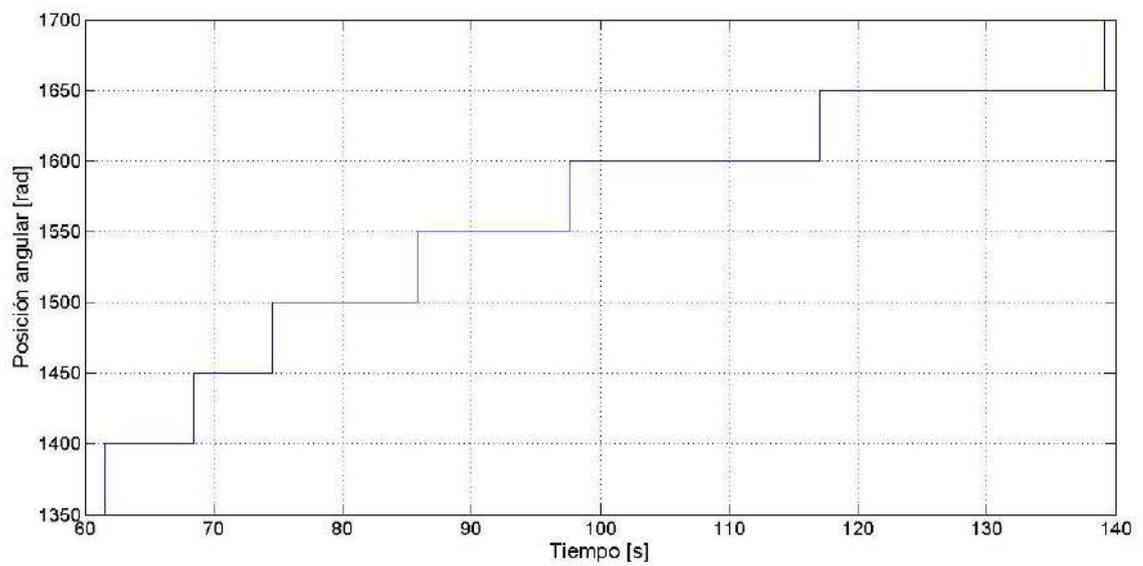


Figura A.3.12. Secuencia de escalones de PWM correspondiente a la figura anterior.

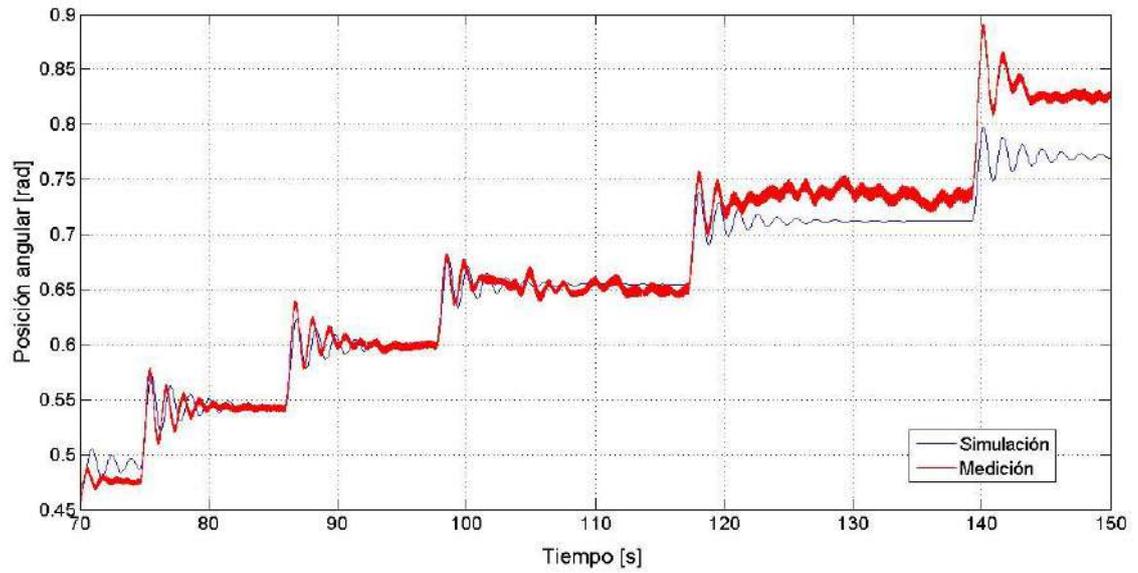


Figura A.3.13. Rango útil de la planta  $G_4(s)$ , de 1450 a 1650 cuentas de PWM.

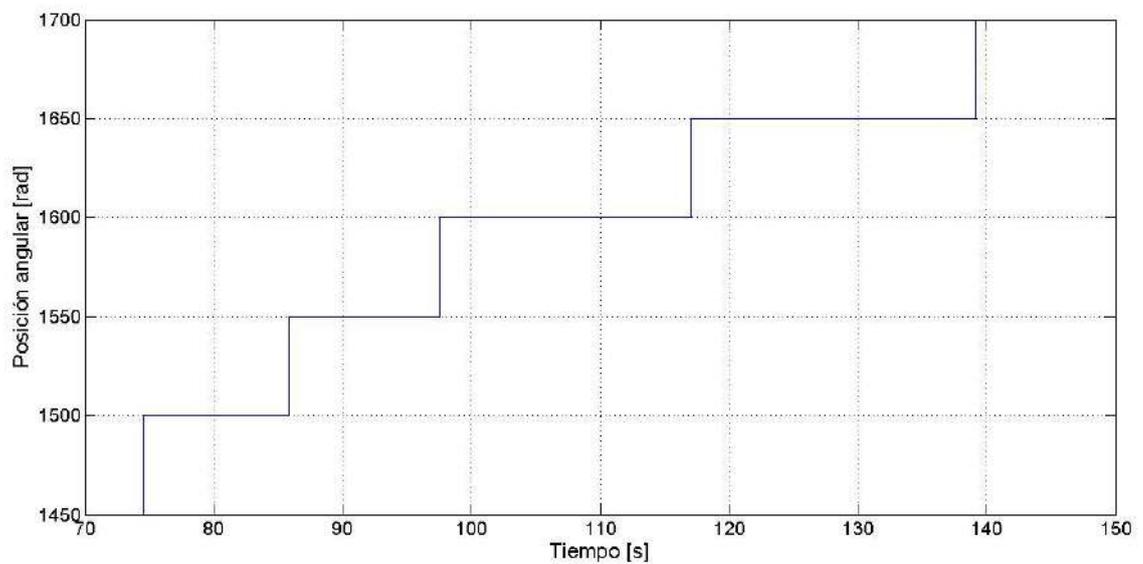


Figura A.3.14. Secuencia de escalones de PWM correspondiente a la figura anterior.

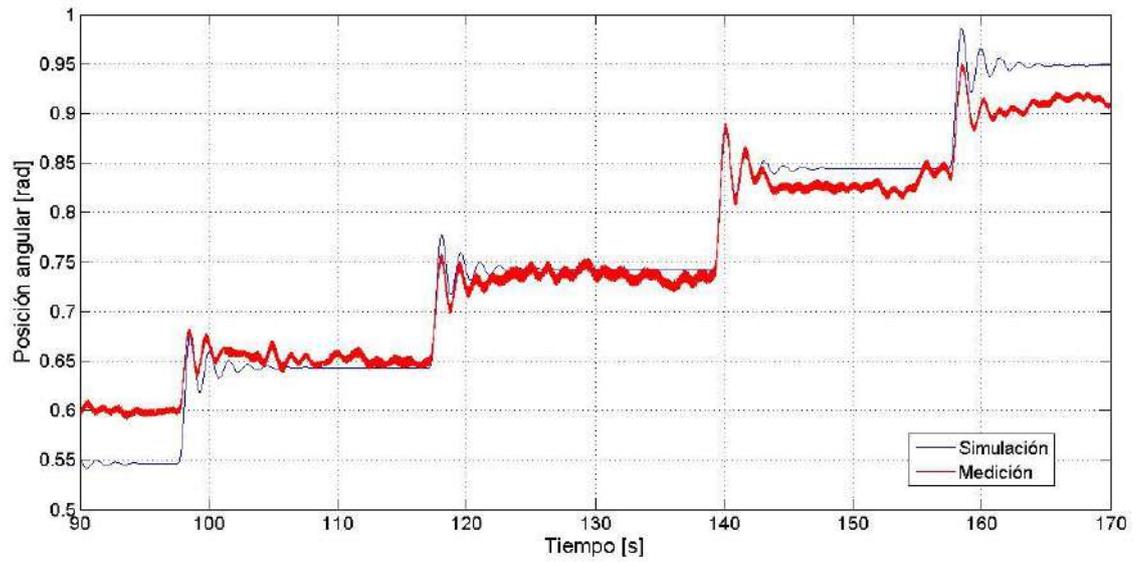


Figura A.3.15. Rango útil de la planta  $G_5(s)$ , de 1600 a 1700 cuentas de PWM.

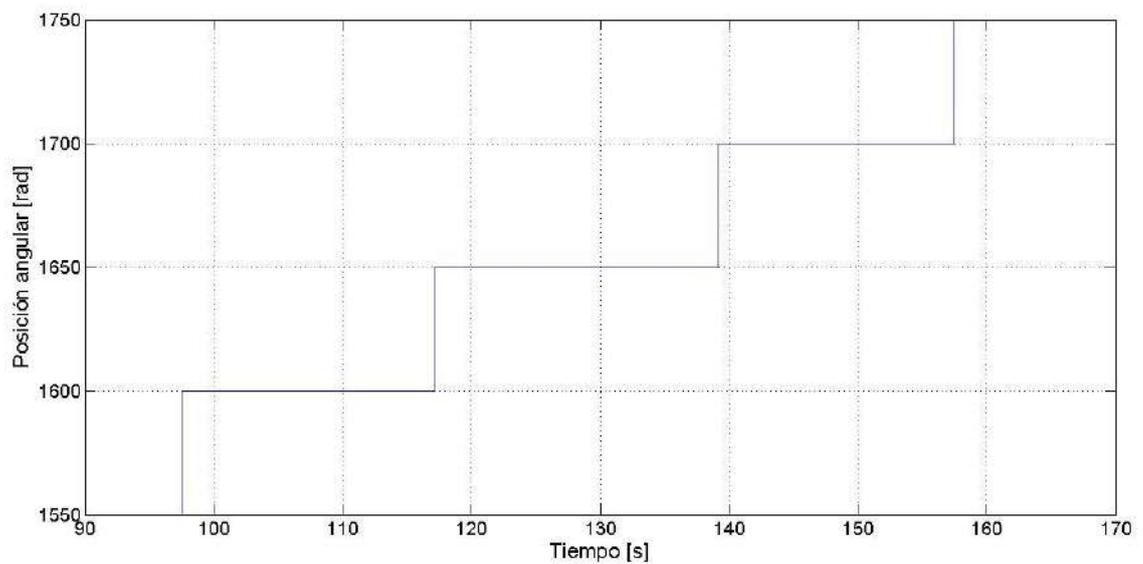


Figura A.3.16. Secuencia de escalones de PWM correspondiente a la figura anterior.

## Anexo 4

Para el diseño y construcción del controlador presentado en la ecuación (2.2.1), se construyó el lugar de las raíces de:

$$G(s) = G_3'(s) \cdot \frac{27.5}{(0.3 \cdot s + 1)}$$

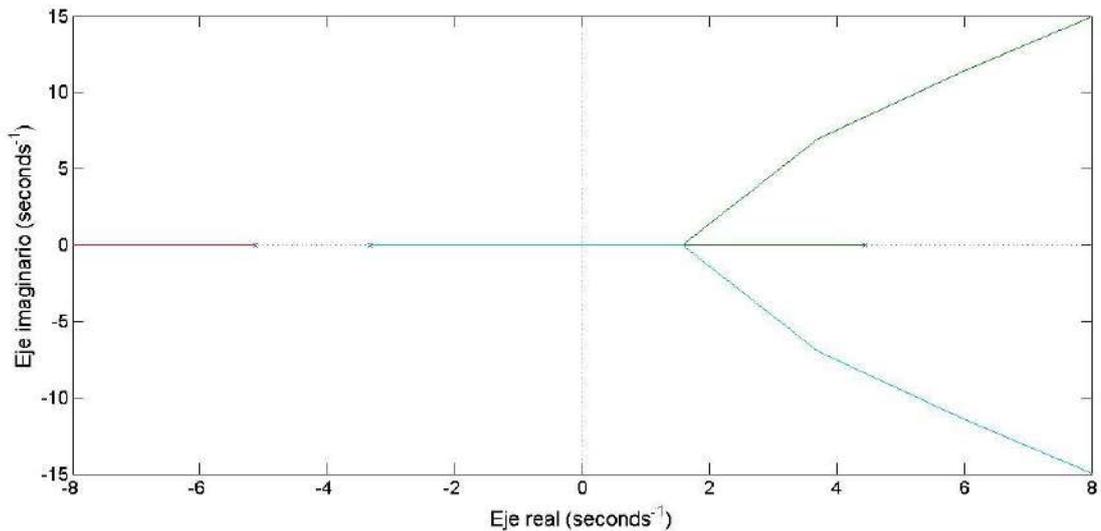


Figura A.4.1. Lugar de las raíces del sistema a lazo abierto.

Al observar esto, se tomó la decisión construir un controlador formado por dos redes de adelanto, ubicando los ceros de modo de cancelar los polos estables del sistema, mientras que los polos se colocaron en -10 y -200 de modo de obtener el siguiente lugar de las raíces:

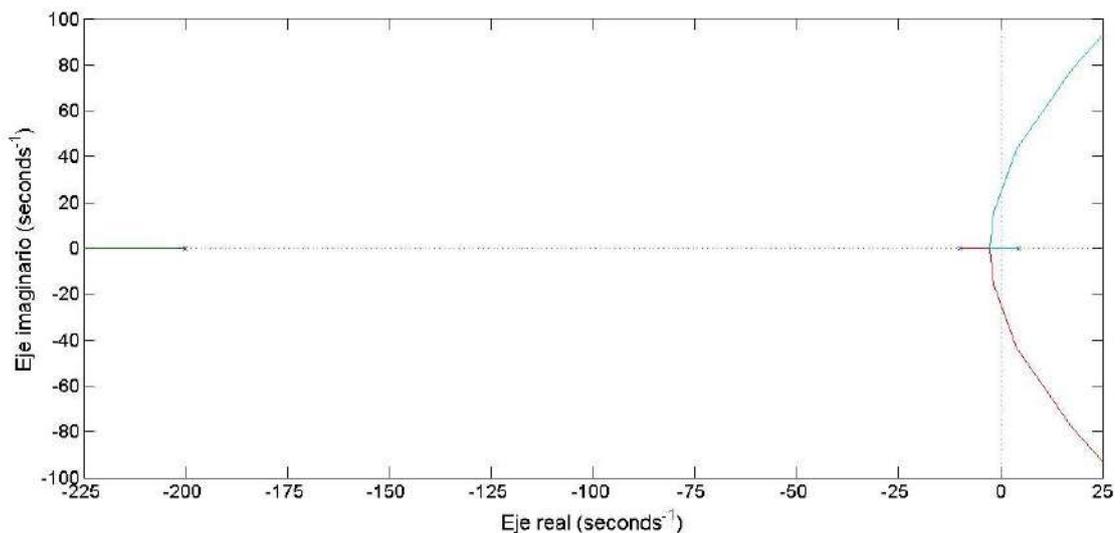


Figura A.4.2. Lugar de las raíces del sistema controlado.

La ecuación de este controlador se presenta en la ecuación (2.2.1):

$$D(s) = \frac{200 \cdot (s + 5.139) \cdot (s + 3.333)}{(s + 200) \cdot (s + 10)}$$

Donde la ganancia fue establecida observando el lugar de las raíces de la figura A.4.2, de modo de que los polos a lazo cerrado queden en el semiplano izquierdo. Este controlador fue discretizado utilizando el equivalente discreto de mapeo de polos y ceros, *matched*, con el fin de poder implementarlo en un algoritmo en C:

$$D(z) = \frac{137.2534 \cdot (z - 0.9868) \cdot (z - 0.9797)}{(z - 0.9608) \cdot (z - 0.4493)}$$