



UNIVERSIDAD DE CASTILLA - LA MANCHA

**ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA INDUSTRIAL**

CIUDAD REAL

TRABAJO FIN DE GRADO EN
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

N.º 21-3-225339

**DISEÑO DE UN ESTIMADOR DE LA FRECUENCIA
DEL TEMBLOR DEL PARKINSON**



Autor/a:
RICARDO CAMACHO DÍAZ - CANO

Tutor/a:
VICENTE FELIU BATLLE

SEPTIEMBRE 2021

AGRADECIMIENTOS

Quiero transmitir mi más sincero agradecimiento a todos aquellos que me han ayudado a lo largo de esta trayectoria.

En primer lugar, a mi familia, sobre todo a mis padres y mi hermana, por haberme brindado la oportunidad de estudiar con la tranquilidad de saber que cuento con su respaldo, apoyándome en todo momento y animándome cuando más lo necesitaba. Soy quien soy gracias a vosotros.

En segundo lugar, a mi tutor Vicente Feliu, por su ayuda en la planificación, información, y sobre todo, la paciencia que ha tenido conmigo.

También, expresar mi más sentido agradecimiento a la universidad por acogerme dentro de sus aulas y hacerme sentir como en casa. Después de esta etapa de aprendizaje, solo tengo palabras de agradecimientos a todos los profesores y compañeros.

A todos ellos, mil gracias.

ÍNDICE GENERAL

1	Introducción	1
1.1	Motivación	1
1.2	Estado del arte	2
1.3	Objetivos del proyecto	3
1.4	Organización del trabajo	3
2	Modelo de estado	5
2.1	Introducción	5
2.2	Concepto de estado	5
2.3	Ecuaciones del modelo de estado	6
2.4	Sistemas dinámicos lineales	7
2.5	Métodos de obtención del modelo de estado	8
2.6	Solución de la ecuación de estado de sistemas lineales	8
2.6.1	Solución de la ecuación homogénea	8
2.6.2	Solución de la ecuación completa	9
2.6.3	Método de la matriz de transición	11
3	Modelo Algebraico	13
3.1	Introducción	13
3.2	Marco matemático	13
4	Resolución de parámetros de una señal sinusoidal ruidosa y sesgada	15
4.1	Introducción	15
4.2	Estimación algebraica de la frecuencia	16
4.2.1	Algoritmos	16
4.2.2	Implementación de las simulaciones de las ecuaciones de estado	21
4.2.3	Implementación de la simulación de las soluciones analíticas	23

4.3	Estimación algebraica de la amplitud y fase	24
4.3.1	Algoritmos	24
4.3.2	Implementación de las simulaciones de las ecuaciones de estado	31
4.3.3	Implementación de las simulaciones de las soluciones analíticas	33
4.4	Estimación por mínimos cuadrados de la amplitud, fase y offset	36
5	Implementación	41
5.1	Introducción	41
5.2	Filtrado	43
5.2.1	Filtro lineal de Butterworth	43
5.2.2	Filtro no lineal mediante integrales de valores absolutos	44
5.3	Selección de la frecuencia	45
5.3.1	Centinela basado en error absoluto	45
5.3.2	Centinela basado en error relativo	46
6	Análisis de robustez	51
6.1	Introducción	51
6.2	Efecto de la frecuencia de la señal	51
6.3	Efecto de la frecuencia de muestreo	52
6.4	Efecto del ratio SNR	53
6.5	Efecto ante señales con alta frecuencia	54
7	Comparación con otros métodos	57
8	Resultados experimentales	61
8.1	Obtención de las señales	61
8.2	Evolución en el tiempo de las señales	62
8.3	Estimación de los parámetros característicos de las señales	62
9	Conclusión	67
9.1	Características del algoritmo desarrollado	67
9.2	Aportaciones y ventajas frente a trabajos previos	68
9.3	Futuros trabajos	69
A	Propiedades del modelo de estado	71
B	Transformaciones lineales	73

C	Métodos de obtención del modelo de estado	75
	C.0.1 Variables de estado como magnitudes físicas	75
	C.0.2 Variables de estado como salida de integradores	75
	C.0.3 Variables de estado de fase	78
	C.0.4 Variables de Jordan	80
D	Propiedades de la matriz de transición	83
E	Casos particulares de la matriz de transición	85
F	Programas en Simulink	87
G	Código en Matlab	91

ÍNDICE DE FIGURAS

4.1	Frecuencia del modelo de estado mediante Simulink.....	22
4.2	Frecuencia del modelo analítico mediante Simulink.....	23
4.3	Frecuencia del modelo analítico mediante Matlab.....	23
4.4	Comparativa entre métodos de obtención de la frecuencia.....	24
4.5	Variable p_3 del modelo de estado mediante Simulink.....	32
4.6	Variable p_2 del modelo de estado mediante Simulink.....	33
4.7	Variable p_3 del modelo analítico mediante Simulink.....	33
4.8	Variable p_3 del modelo analítico mediante Matlab.....	34
4.9	Comparativa entre métodos de obtención de la variable p_3	34
4.10	Variable p_2 del modelo analítico mediante Simulink.....	35
4.11	Variable p_2 del modelo analítico mediante Matlab.....	35
4.12	Comparativa entre métodos de obtención de la variable p_2	36
4.13	Amplitud y fase mediante el método basado en la minimización del error cuadrático con un 20 % de ruido.....	38
4.14	Amplitud, fase y constante mediante el método basado en la minimización del error cuadrático con un 20 % de ruido.....	40
5.1	Frecuencia normalizada sin centinela.....	42
5.2	Frecuencia normalizada con ruido.....	43
5.3	Frecuencia con filtro paso bajo Butterworth.....	44
5.4	Frecuencia con filtro lineal y no lineal.....	45
5.5	Condición vs tiempo de convergencia.....	46
5.6	Condición vs tiempo de convergencia entre 10 y 1000.....	47
5.7	Condición vs tiempo de convergencia entre 100 y 300.....	49
5.8	Condición vs tiempo de convergencia en presencia de ruido.....	49
6.1	Frecuencia vs tiempo de convergencia.....	52
6.2	Muestras vs frecuencia de muestreo.....	53
6.3	Tiempo de convergencia vs SNR.....	54
6.4	Tiempo de convergencia vs SNR.....	55
6.5	Frecuencia de la señal normalizada junto con una señal de alta amplitud.....	55

7.1	Transformada rápida de Fourier (FFT) para la señal normalizada.....	58
7.2	Error de la frecuencia en función del tiempo de simulación y el periodo de muestreo.....	58
8.1	Plataforma experimental con un diagrama esquemático del sistema de estimación.	61
8.2	Evolución temporal de las medidas del acelerómetro por eje.....	62
8.3	Evolución de la estimación de la frecuencia por eje.	63
8.4	Selección del rango y su condición asociada por eje.....	63
8.5	Evolución de la estimación de la frecuencia por eje.	64
8.6	Comparación de la señal estimada con los datos reales.....	65
8.7	Diferencia entre la señal estimada y los datos reales.	66
8.8	FFT de los datos reales.	66
A.1	Propiedad de unicidad.	71
A.2	Propiedad de transición.	72
F.1	Estructura interna del bloque espacio de estado variable.	87
F.2	Frecuencia en Simulink mediante modelo de estado	88
F.3	Frecuencia en Simulink mediante modelo analítico	88
F.4	Variable p_3 en Simulink mediante modelo de estado.....	89
F.5	Variable $p_3(t)$ en Simulink mediante modelo analítico	89
F.6	Variable $p_2(t)$ en Simulink mediante modelo de estado	90
F.7	Variable p_2 en Simulink mediante modelo analítico.....	90

ÍNDICE DE TABLAS

5.1	Valores sobre las condiciones límites del centinela	48
-----	---	----

LISTA DE SÍMBOLOS

A	—	Amplitud de la señal
$A(t)$	—	Matriz del sistema, de dimensión $n \times n$
A_x	—	Variables intermedias que almacenan el valor de la ejecución anterior
$B(t)$	—	Matriz de entrada, de dimensión $n \times m$
$C(t)$	—	Matriz de salida, de dimensión $n \times n$
$D(t)$	—	Matriz, de dimensión $p \times n$
d_1	—	Denominador de la variable intermedia p_1
d_{1f}	—	Denominador de la variable intermedia p_1 filtrado
d_2	—	Denominador de la variable intermedia p_2
d_3	—	Denominador de la variable intermedia p_3
Δ_i	—	Condición del centinela
E	—	Media del vector
e	—	Minimización del error cuadrático
f	—	Frecuencia de la señal (Hz)
f_c	—	Frecuencia de corte (Hz)
\hat{f}	—	Frecuencia estimada (Hz)
f_s	—	Frecuencia de muestreo (Hz)
I_x	—	Variables intermedias que almacenan el valor de la ejecución anterior
K	—	offset de la señal
$k[d/dt]$	—	Anillo de operadores diferenciales lineales
λ_i	—	Autovalores de las matrices
N	—	Numero de muestras
n_1	—	Numerador de la variable intermedia p_1
n_{1f}	—	Numerador de la variable intermedia p_1 filtrado
n_2	—	Numerador de la variable intermedia p_2

n_3	—	Numerador de la variable intermedia p_3
ω	—	Velocidad angular (rad)
ω_n	—	Frecuencia de corte (rad)
$\hat{\omega}$	—	Frecuencia estimada (rad)
$P(\lambda)$	—	Polinomio característico
p_1	—	Variable intermedia
p_2	—	Variable intermedia
p_3	—	Variable intermedia
p_4	—	Variable intermedia
p_5	—	Variable intermedia
ϕ	—	Fase de la señal (rad/s)
Φ	—	Matriz de transición
σ	—	Varianza
t	—	Instante del tiempo
t_0	—	Instante inicial
t_{conv}	—	Tiempo de convergencia
T_s	—	Tiempo de muestreo (seg.)
tol	—	Tolerancia
θ	—	Conjunto de parámetros desconocidos
u	—	Vector de entrada de dimensión m
φ	—	Error cuadrático
x	—	Vector de estado de dimensión n
$x(t_0)$	—	Condiciones iniciales
x_i	—	Variables de estado
\dot{x}	—	Ecuaciones de estado
ξ	—	Ruido
z	—	Solución de la ecuación completa

CAPÍTULO 1

INTRODUCCIÓN

1 1

Motivación

La enfermedad del Parkinson es junto al Alzheimer y la demencia senil un trastorno neurodegenerativo que afecta al sistema nervioso de forma progresiva. Se encuentra extendido por todo el mundo afectando sobre todo a personas de avanzada edad.

El Parkinson se origina por la pérdida (o degeneración) de neuronas en la sustancia negra, generando una falta de dopamina en el organismo. La dopamina es la encargada de transmitir la información en el sistema nervioso central, por lo que la falta de dicha sustancia hace que el control del movimiento se vea alterado, generando el característico temblor.

Los síntomas se caracterizan por un movimiento lento (bradicinesia), rigidez y aumento del músculo, temblores y pérdida del control postural. Estos síntomas comienzan gradualmente, a veces simplemente con un leve temblor en una sola mano o un balanceo leve y suave de los brazos al caminar, pasando totalmente desapercibidos.

Las causas del Parkinson aún se desconocen, pero se considera que pueden deberse a una combinación de factores genéticos, medioambientales y aquellos derivados del propio envejecimiento, haciendo que sea una enfermedad muy presente en personas de edad avanzada, afectando entre un 1 % y un 2 % de la población cercana a los 60 años y entre un 0,5 % y un 5 % de la población mayor de 65 años, presentando los primeros síntomas a partir de los 40 años.

A pesar de ser una enfermedad muy extendida, el Parkinson no tiene cura. Su tratamiento consiste en mejorar, o al menos mantener y prolongar, la funcionalidad del sistema nervioso durante el mayor tiempo posible a través de medicamentos, cirugías y rehabilitación.

Dada la influencia del Parkinson a nivel global, se han realizado incontables estudios sobre esta enfermedad. En este escrito nos vamos a centrar en los temblores, que son producidos por el movimiento repetitivo de los ganglios basales a una determinada frecuencia, pudiéndose estimar dicho comportamiento mediante una señal sinusoidal. El rango de frecuencias en el que se producen los temblores fueron establecidos entre 4-8 Hz por Hassin-Baer y Sirota [8] y Jang [13]. Marcando el principal objetivo en identificar

la señal sinusoidal dentro de dicho rango, para posteriormente tratar y corregir dicho temblor, aumentando de forma considerable la calidad de vida.

1 2

Estado del arte

En la última década se le ha dado mucha importancia al problema de estimar una señal sinusoidal, debido a que es un problema muy presente en numerosos ámbitos, que van desde la biomedicina hasta los sistemas eléctricos y mecánicos.

Los métodos propuestos en el pasado para la solución de este problema utilizan una amplia gama de técnicas, ya sean en el dominio del tiempo como en el dominio de la frecuencia. Una recopilación completa de estos trabajos se encuentra recogida por Stoica en [22].

Los métodos en el dominio de la frecuencia se basan en el análisis del espectro, donde los métodos más clásicos incluyen la maximización del periodograma (MP) y la regresión de cuadrados no lineales (NLS). Ambos métodos son estadísticamente eficientes en el sentido del límite inferior de Cramér-Rao como se puede ver en los artículos de Kay y Marple [14], Rife y Boorstyn [18] y Andersson y Händle [1]. Los principales problemas de estos métodos es que requieren el uso de algoritmos de optimización iterativos y unas condiciones iniciales muy precisas para asegurar la convergencia.

Por otro lado, están los métodos basados en los estimadores espectrales no paramétricos, los cuales son muy adecuados cuando la aplicación presenta una alta relación señal-ruido (SNR), pero tienen el inconveniente de presentar una alta carga numérica, es decir, una alta demanda computacional, siendo métodos no válidos para aplicaciones en tiempo real. El más conocido de todos es la transformada rápida de Fourier (FFT), aunque algunos casos prácticos son los artículos de Haykin [10] y Roy y Kailath [19].

Considerando los inconvenientes que presentan los métodos en el dominio de la frecuencia y que para las aplicaciones de control en tiempo real se requiere un método con una alta eficiencia computacional y una rápida convergencia, se plantea la solución en el dominio del tiempo mediante un identificador convergente adaptativo, propuesto por Hsu, Ortega y Damm en [12], el cual está basado en filtros no lineales y fue introducido como una buena solución matemática, donde su configuración no es una tarea sencilla como se puede ver en el artículo de Clarke [3]. Esta línea de trabajo fue extendida por Mojiri y Bakhshai en [16] para la estimación de los parámetros en señales periódicas.

De forma independiente se proponen los identificadores basados en el tiempo discreto. Algunos de estos identificadores pueden encontrarse en los artículos de Li y Kedem [15] y Xiao y Tadokoro [25]. Los resultados de convergencia en el tiempo discreto son considerablemente más rápidos que en tiempo continuo, pero estos métodos son muy sensibles a la frecuencia de muestreo y a las condiciones iniciales, ya que, si la frecuencia de muestreo es muy alta en comparación con la frecuencia natural de la señal, estos algoritmos presentan problemas de convergencia. En los artículos de Unbehauen y Rao [24] y Young [27] se presentan de forma más detallada, las ventajas de los métodos basados en tiempo continuo en contraposición a los métodos basados en tiempo discreto.

Como solución a la estimación de la señal sinusoidal, se propone la técnica de identificación algebraica introducida por Fliess y Sira-Ramírez en [5], dicha técnica se basa en el análisis de señales en el dominio del tiempo, mediante el método derivativo. Este método obtiene todas las ventajas de ser un método paramétrico en el dominio del tiempo, es decir, es computacionalmente muy eficiente, capaz de estimar la señal en menos de un ciclo, es independiente de las condiciones iniciales, robusto ante situaciones con un alto nivel de ruido y no es sensible ante cambios en la frecuencia de muestreo.

Algunos artículos que usan dicho método son para la identificación algebraica de las dos primeras frecuencias naturales de una estructura flexible por Pereira, Trapero y Feliu-Batlle [17], un estimador algebraico rápido para la estimación de parámetros del sistema y el ajuste del controlador en línea - una aplicación de un nanoposicionador por San-Millan, Aphale y Feliu-Batlle [21], un estimador de la masa de un brazo flexible que se mueve bajo la gravedad por San-Millan Cambera y Feliu-Batlle [20]. Y destacando sobre todo, los artículos de Yasar [26] y Trapero, Sira-Ramírez y Feliu-Batlle [23] para la estimación de la frecuencia de una señal sinusoidal sesgada y ruidosa, que se seguirán de manera muy cercana.

1 3

Objetivos del proyecto

El objetivo principal de este trabajo es identificar el comportamiento de los temblores del Parkinson, mediante la estimación de los parámetros característicos que definen una señal sinusoidal, con el fin de recrearla para posteriormente realizar un tratamiento sobre dicha señal.

La estimación de la amplitud, frecuencia, fase y offset de una señal sinusoidal sesgada y ruidosa se realizarán mediante el uso del método algebraico y el método basado en la minimización del error cuadrático, observando en todo momento el comportamiento de dichos modelos a través de diferentes soluciones y obteniendo la más óptima.

Se mejorará el modelo construido mediante la aplicación de varios filtros, que eliminarán las discontinuidades y aislarán los datos de las frecuencias indeseadas; y se implementará un algoritmo de re-iniciación llamado centinela que determinará cuando la señal ha convergido y finalizará la ejecución, requiriendo menos esfuerzo computacional y uso de memoria, así como conseguir disminuir el tiempo de ejecución.

Tanto para el filtro como para el centinela, se implementarán varias opciones y se elegirá la mejor, ya que en todo momento se busca el equilibrio entre rapidez y precisión. Una vez definido nuestro algoritmo, se adecuará a los datos experimentales proporcionados por la investigadora Claudia Yasar, y se observará su comportamiento aplicado a los temblores del Parkinson.

1 4

Organización del trabajo

Este trabajo está estructurado de la siguiente forma. En el Capítulo 2, se introduce el concepto de modelo de estado, el cual constituye la base fundamental para la

representación e identificación de cualquier sistema, así como el método general para encontrar la solución al modelo de estado. En el Capítulo 3, se explica de forma detallada el método algebraico, con el cual gracias a él se obtiene el modelo de estado que define el comportamiento del sistema. En el Capítulo 4, se resuelve el problema de estimar los parámetros característicos de una señal sinusoidal ruidosa y sesgada, comparando las diferentes soluciones y sacando la conclusión de cuál es la más óptima para cada caso. En el Capítulo 5, se realiza la implementación del algoritmo, incluyendo las mejoras que proporcionan la introducción de los filtros y del centinela, estudiando las diferentes opciones para construir finalmente el algoritmo definitivo. En el Capítulo 6, se realizará el análisis de robustez, donde se prueba el algoritmo en diferentes ambientes desfavorables y se sacan conclusiones de su eficiencia y los defectos que presenta. En el Capítulo 7, se compara con otros métodos introducidos anteriormente, destacando la comparación con la transformada rápida de Fourier (FFT). En el Capítulo 8, se adecúa el algoritmo para nuestro caso práctico fijado en los temblores del Parkinson, y se realizará el estudio con los datos experimentales obtenidos en el laboratorio. Por último, en el Capítulo 9, se realiza una valoración general como conclusión del trabajo los aportes realizados frente a trabajos previos, así como una visión de futuros del proyecto.

CAPÍTULO 2

MODELO DE ESTADO

2 1

Introducción

En este capítulo se introducirá el modelo de estado, el cual constituye la base fundamental para la representación e identificación de cualquier sistema. Se explicarán los conceptos y procedimientos para su obtención y una vez identificado, se explicará el método general para resolver y encontrar la solución que define totalmente su comportamiento.

Todos los procedimientos se explicarán de una forma breve y general, para posteriormente aplicarlo en nuestro caso práctico. Dichos procedimientos se apoyan con información complementaria en los apéndices, así como en el extendido libro de Domínguez [4], dentro del cual vienen definidos todos los conceptos, incluyendo ejemplos adicionales para facilitar su comprensión.

2 2

Concepto de estado

El modelo de estado está basado en los principios de la teoría moderna de control, la cual se centra en el conocimiento de las variables que definen el comportamiento y dinámica de los sistemas, en contraposición a la teoría clásica de control, que utiliza únicamente la relación entre su entrada y salida.

Se define estado de un sistema, como la mínima cantidad de información necesaria en un instante concreto, donde conociendo la entrada en dicho instante, se pueda determinar el valor de sus variables. Dicha cantidad viene dada por un conjunto de variables $x_i(t)$, denominadas variables de estado del sistema, que se agrupan dentro del vector $x(t)$, denominado vector de estado. Todas las variables de estado del sistema dependen del instante de tiempo en el que se encuentren; y en la mayoría de sistemas físicos, el vector de estado es de dimensión finita, n . Además, se representa el conjunto de variables de entrada mediante el vector $u(t)$, definiendo el vector de estado con la expresión matemática:

$$x(t) = \psi(t, t_0, x(t_0), u(\tau)), \quad t_0 < \tau \leq t \quad (2.1)$$

donde el vector de estado depende de las condiciones iniciales t_0 y $x(t_0)$, el instante de tiempo t y la entrada en dicho instante $u(t)$. Dicha definición es aplicable para la representación tanto de sistemas lineales como no lineales.

Se define el concepto de espacio de estado:

Definición 2.1 Espacio de estado es el espacio vectorial en el cual el vector de estado toma valores, teniendo por tanto la misma dimensión que el número de elementos de dicho vector.

El espacio de estado es un espacio vectorial que admite cualquier base, estando relacionadas entre sí mediante transformaciones lineales y existiendo infinitas representaciones para un mismo sistema. Pero esta dependencia no afecta a su comportamiento, pues para la misma entrada devuelve la misma salida.

En todo el escrito se va a trabajar con el espacio de estado para sistemas lineales, pues la aplicación del vector de estado en sistemas no lineales añade un nivel tan alto de complejidad que se encuentra fuera del alcance de la aplicación.

Para comprender de manera más profunda el vector de estado, en el apéndice A se muestran las principales propiedades que marcan la base fundamental para su posterior desarrollo.

2 3

Ecuaciones del modelo de estado

Partiendo de la expresión matemática (2.1), y conociendo que el estado es capaz de recoger toda la información del sistema en un determinado instante, esta ecuación se puede expresar como:

$$y(t) = \eta(t, x(t), u(t)) \quad (2.2)$$

donde la salida en el instante t depende del tiempo t , del estado $x(t)$ y de la entrada $u(t)$ en ese instante.

Los sistemas diferenciales dinámicos se caracterizan porque pueden ser representados de la siguiente forma:

$$\dot{x}(t) = f(t, x(t), u(t)) \quad (2.3)$$

$$y(t) = \eta(t, x(t), u(t)) \quad (2.4)$$

donde la ecuación (2.3) se le conoce como ecuación de estado (representa la evolución del estado del sistema mediante ecuaciones diferenciales) y la ecuación (2.4) se le conoce como ecuación de salida. Al conjunto de las dos ecuaciones se le llama modelo de estado del sistema, así mismo se llama orden del modelo al número de variables que define el estado.

Las ecuaciones del modelo de estado respetan la continuidad del sistema, donde además el orden del modelo coincide con el número mínimo de condiciones iniciales necesarias para resolver la ecuación de estado.

Gracias a la ecuación (2.3) se pueden determinar propiedades como la estabilidad del sistema, asegurándose que funciona en un determinado rango sin que se produzca una discontinuidad, y en el caso de que se produzca, poder corregirla, determinar la naturaleza (si es lineal o no lineal), y la determinación de los estados de equilibrio. Estos últimos se obtienen mediante la siguiente ecuación:

$$f(t, x(t), u(t)) = 0 \quad (2.5)$$

2 4

Sistemas dinámicos lineales

El primer paso es determinar si un sistema es lineal o no, para ello se aplica el test de superposición:

Partiendo de un sistema en un estado inicial cualquiera $x_1(t_0)$, con una entrada cualquiera $u_1(\tau)$, devuelve una salida $y_1(t)$; y a partir de otro estado inicial cualquiera $x_2(t_0)$, con otra entrada cualquiera $u_2(\tau)$, devuelve una salida $y_2(t)$. Se dice que dicho sistema es lineal si para todo a y b reales, partiendo de un estado inicial $x_3(t_0) = ax_1(t_0) + bx_2(t_0)$ con una entrada $u_3(\tau) = au_1(\tau) + bu_2(\tau)$, devuelve como salida $y_3(t) = ay_1(t) + by_2(t)$.

Aplicando esta propiedad a las ecuaciones del modelo de estado, se traduce en:

$$f(t, \alpha x_1(t) + \beta x_2(t), \alpha u_1(t) + \beta u_2(t)) = \alpha f(t, x_1(t), u_1(t)) + \beta f(t, x_2(t), u_2(t)) \quad (2.6)$$

$$\eta(t, \alpha x_1(t) + \beta x_2(t), \alpha u_1(t) + \beta u_2(t)) = \alpha \eta(t, x_1(t), u_1(t)) + \beta \eta(t, x_2(t), u_2(t))$$

donde f y η son funciones vectoriales, cumpliendo la propiedad de linealidad.

La forma más común en la que se representan las ecuaciones del modelo de estado es en su forma matricial:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.7)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (2.8)$$

donde:

$x(t)$ es el vector de estado, de dimensión n .

$u(t)$ es el vector de entradas, de dimensión m .

$y(t)$ es el vector de salida de dimensión p .

$A(t)$ es la matriz del sistema, de dimensión $n \times n$.

$B(t)$ es la matriz de entradas, de dimensión $n \times m$.

$C(t)$ es la matriz de salida, de dimensión $p \times n$.

$D(t)$ tiene dimensiones $p \times n$ (en la mayoría de los sistemas es nula)

Las expresiones de las matrices $A(t)$, $B(t)$, $C(t)$ y $D(t)$ dependen de la representación del estado del sistema.

Si para un sistema con un estado inicial $x_0 = x(t_0)$ se le aplica una entrada $u_1(\tau)$, $t_0 < \tau \leq t$, produce una salida $y_1(t)$. Se dice que es invariante para $\forall t$, si se parte de ese mismo estado x_0 , pero en el instante $t_0 + T$, con la misma entrada en $u_1(\tau + T) = u_1(\tau)$, responde con una salida $y_2(\tau + T) = y_1(t)$. Esto significa que los elementos que componen las matrices A , B , C y D son constantes.

2 5**Métodos de obtención del modelo de estado**

Como se mencionó anteriormente, un modelo de estado tiene infinitas representaciones, todas ellas igual de válidas, que se pueden relacionar entre sí mediante las transformaciones lineales, explicadas en el apéndice B.

En este escrito se tratará en concreto, la obtención del modelo de estado mediante el método algebraico, al cual se dedicará un capítulo entero para su explicación. De forma complementaria en el apéndice C se explican los métodos generales más comunes para la representación del modelo de estado.

2 6**Solución de la ecuación de estado de sistemas lineales**

Una vez obtenida la representación del modelo de estado, se procede a encontrar su solución. Se recuerda que las ecuaciones de estado están compuestas por ecuaciones diferenciales que describen completamente el comportamiento del sistema, por lo que encontrar una solución implica conocer la evolución del sistema y poder estudiar su funcionamiento tanto en régimen dinámico como estático.

Para encontrar dicha solución se usa el método general, donde solo es aplicable a sistemas lineales, es decir, a aquellos sistemas que responden de forma lineal incluso cuando la entrada no lo es. El objetivo del método general es encontrar una solución a cualquier sistema de ecuaciones de estado de la forma:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.9)$$

El método general funciona de igual forma que para sistemas convencionales de ecuaciones diferenciales, donde primero se encuentra una solución a la ecuación homogénea y posteriormente a la ecuación completa. La ecuación homogénea es aquella cuya entrada es nula, es decir, el sistema no recibe ningún tipo de estímulo, siendo la respuesta del sistema libre; mientras que la solución de la ecuación completa viene definida por un estímulo externo.

2 6 1 Solución de la ecuación homogénea

La ecuación homogénea viene definida por la ecuación de estado cuando la entrada $u(t)$ es nula:

$$\dot{x}(t) = A(t)x(t) \quad (2.10)$$

con condiciones iniciales $x(t_0) = x_0$.

Para resolver la ecuación homogénea se puede utilizar el método de integración por aproximaciones sucesivas de Peano-Baker, donde si se tiene una ecuación diferencial de la forma:

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2.11)$$

con condiciones iniciales $x(t_0) = x_0$, se puede obtener la solución de dicho sistema

construyendo una secuencia de integraciones:

$$\begin{aligned}\varphi_0 &= x_0 \\ \varphi_k(t) &= \varphi_0 + \int_{t_0}^t f(\varphi_{k-1}(\tau), u(\tau), \tau) d\tau\end{aligned}\quad (2.12)$$

donde la solución a la ecuación diferencial se obtiene como:

$$x(t) = \lim_{k \rightarrow \infty} \varphi_k(t) \quad (2.13)$$

Para el caso de la ecuación (2.10), se obtiene la siguiente sucesión:

$$\begin{aligned}\varphi_1(t) &= \varphi_0 + \int_{t_0}^t A(\tau) x_0 d\tau = x_0 + \int_{t_0}^t A(\tau) d\tau x_0 \\ \varphi_2(t) &= \varphi_0 + \int_{t_0}^t A(\tau) [x_0 + \int_{t_0}^{\tau} A(\tau_1) d\tau_1 x_0] d\tau \\ &= x_0 + \int_{t_0}^t A(\tau) d\tau x_0 + \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1) d\tau_1 d\tau x_0 \\ &= [I + \int_{t_0}^t A(\tau) d\tau + \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1) d\tau_1 d\tau] x_0\end{aligned}\quad (2.14)$$

A medida que se desarrollan los términos de la serie, se puede observar como aparece el término x_0 en todos ellos. Se saca como factor común, agrupándose como:

$$x(t) = \Phi(t, t_0) x_0 \quad (2.15)$$

donde $\Phi(t, t_0)$ es conocida como matriz de transición, la cual se construye mediante la integración por aproximaciones sucesivas de la ecuación homogénea:

$$\begin{aligned}\Phi(t, t_0) &= \lim_{k \rightarrow \infty} [I + \int_{t_0}^t A(\tau) d\tau + \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1) d\tau_1 + \cdots \\ &\quad \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1) \cdots \int_{t_0}^{\tau_{k-1}} A(\tau_k) d\tau_k \cdots d\tau_1 d\tau]\end{aligned}\quad (2.16)$$

El vector $x(t)$ de la ecuación (2.15) es la solución a la ecuación homogénea, siendo única su solución.

Como conclusión se puede determinar que la ecuación homogénea viene definida por el comportamiento del sistema, definido por la matriz de transición, y las condiciones iniciales del sistema. En el caso de que estas fuesen nulas, la ecuación homogénea también lo sería.

2.6.2 Solución de la ecuación completa

Una vez que se ha obtenido el vector $x(t)$, se procede a resolver la ecuación completa. En el apéndice D se muestran las propiedades que posee la matriz de transición, la cual es de gran utilidad para su desarrollo.

La ecuación de estado para una entrada $u(t)$ es:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.17)$$

Para calcular la solución de la ecuación completa se utiliza el método de la variación de las constantes, donde se parte de que existe un vector $z(t)$ que es la solución de la ecuación completa:

$$x(t) = \Phi(t, t_0)z(t) \quad (2.18)$$

Para desarrollar la solución, se trabaja sobre el vector $z(t)$. Primero se sustituye la expresión (2.18) en la ecuación de estado:

$$\begin{aligned} \dot{\Phi}(t, t_0)z(t) + \Phi(t, t_0)\dot{z}(t) &= A(t)\Phi(t, t_0)z(t) + B(t)u(t) \\ \Phi(t, t_0)\dot{z}(t) + [\dot{\Phi}(t, t_0) - A(t)\Phi(t, t_0)]z(t) &= B(t)u(t) \end{aligned} \quad (2.19)$$

donde el término entre corchetes se anula por la propiedad de la derivada con respecto al tiempo de la matriz de transición.

Se despeja el término $\dot{z}(t)$:

$$\dot{z}(t) = \Phi^{-1}(t, t_0)B(t)u(t) \quad (2.20)$$

Obteniendo el vector $z(t)$:

$$z(t) = z(t_0) + \int_{t_0}^t \Phi^{-1}(\tau, t_0)B(\tau)u(\tau)d\tau \quad (2.21)$$

donde se aplica la propiedad de inversión en el tiempo de la matriz de transición y se tiene en cuenta que $z(t_0) = x(t_0)$:

$$z(t) = x(t_0) + \int_{t_0}^t \Phi(t_0, \tau)B(\tau)u(\tau)d\tau \quad (2.22)$$

Una vez obtenido el vector $z(t)$, se sustituye en la ecuación (2.18), para poder relacionar el vector $x(t)$ con la solución de la ecuación completa:

$$x(t) = \Phi(t, t_0)x(t_0) + \Phi(t, t_0) \int_{t_0}^t \Phi(t_0, \tau)B(\tau)u(\tau)d\tau \quad (2.23)$$

El término $\Phi(t, t_0)$ que no depende de τ , se introduce dentro de la integral como una variable, produciendo la siguiente expresión final:

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau \quad (2.24)$$

Si se observa la solución a la ecuación completa, esta tiene dos términos. El primero de ellos define el comportamiento y la evolución libre del sistema proporcionada por las condiciones iniciales, es decir, comprende la solución a la ecuación homogénea del sistema, mientras que el segundo término determina la evolución del sistema con respecto a la entrada, existiendo solamente cuando se excita el sistema.

La principal ventaja es que se trata de un método general y totalmente válido para encontrar la solución del modelo de estado de cualquier sistema lineal, siendo un proceso sencillo, pues solo hace falta calcular la matriz de transición que se explicará en el siguiente apartado y aplicarla a la ecuación general (2.24).

2.6.3 Método de la matriz de transición

Para encontrar la solución al modelo de estado es necesario calcular la matriz de transición, donde el cálculo de la matriz de transición es equivalente al cálculo de la exponencial de una matriz:

$$\Phi(t, t_0) = e^{\int_{t_0}^t A(\tau) d\tau} \quad (2.25)$$

Se utiliza el método de Cayley-Hamilton para la obtención de la matriz de transición, el cual dice, que para cualquier matriz $A(t)$ que sea factorizable, la matriz de transición se puede expresar mediante su desarrollo en serie, que no es otra cosa que un polinomio que depende de M y de t :

$$e^{M \int_{t_0}^t \alpha(\tau) d\tau} = \sum_{k=0}^{\infty} a_k(t) M^k = Q(M, t) \quad (2.26)$$

donde:

$$a_k(t) = \frac{1}{k!} \left[\int_{t_0}^t \alpha(\tau) d\tau \right]^k \quad (2.27)$$

El teorema de Cayley-Hamilton demuestra que existe un polinomio $R(M, t)$, de grado finito y menor que n , el cual es calculado a partir del polinomio infinito particularizado en $A(t)$. Se llama $P(\lambda)$ al polinomio característico de la matriz M , cuyo grado coincide con el número de variables del sistema n , de forma que la expresión (2.26) se puede reescribir como:

$$Q(\lambda, t) = C(\lambda, t)P(\lambda) + R(\lambda, t) \quad (2.28)$$

donde $C(\lambda, t)$ es el polinomio cociente, $R(\lambda, t)$ un polinomio de grado $n-1$ y $P(\lambda, t)$ el polinomio característico. Donde:

$$R(\lambda, t) = \sum_{j=0}^{n-1} h_j(t) \lambda^j \quad (2.29)$$

Mientras que el polinomio $P(\lambda, t)$ se calcula:

$$P(M) = 0 \quad (2.30)$$

Si la ecuación (2.28) se particulariza para $\lambda = M$, se obtiene la siguiente expresión:

$$Q(M, t) = R(M, t) \quad (2.31)$$

Para obtener los coeficientes del polinomio $R(M, t)$ se usa la propiedad de los valores propios que anulan el polinomio característico:

$$P(\lambda_i) = 0 \quad \forall \lambda_i \quad (2.32)$$

donde λ_i son los valores propios de la matriz M , por lo que:

$$Q(\lambda_i, t) = R(\lambda_i, t) \quad (2.33)$$

gracias a esto se obtiene un sistema de n ecuaciones:

$$e^{\lambda_i \int_{t_0}^t \alpha(\tau) d\tau} = R(\lambda_i, t) = \sum_{j=0}^{n-1} h_j(t) \lambda_i^j \quad (2.34)$$

que permite obtener las n incógnitas $h_j(t)$.

En el caso de que la matriz M presente valores propios múltiples, se debe derivar la expresión hasta encontrar un orden inferior a la unidad. Con esto se obtienen tantas ecuaciones como incógnitas. Por lo que, dado un valor propio λ_i con multiplicada m_i , el polinomio característico es de la forma:

$$P(\lambda) = (\lambda - \lambda_i)^{m_i} P_i(\lambda) + R(\lambda, t) \quad (2.35)$$

Se deriva con respecto a λ :

$$\begin{aligned} \frac{dQ(\lambda, t)}{d\lambda} = \frac{dC(\lambda, t)}{d\lambda} P_i(\lambda) (\lambda - \lambda_i)^{m_i} + C(\lambda, t) \frac{dP_i(\lambda)}{d\lambda} (\lambda - \lambda_i)^{m_i} \\ + C(\lambda, t) P_i(\lambda) m_i (\lambda - \lambda_i)^{m_i-1} + \frac{dR(\lambda, t)}{d\lambda} \end{aligned} \quad (2.36)$$

donde la expresión se anula en todos los términos cuando se particularizan en λ_i salvo en el último, quedando:

$$\frac{dQ(\lambda, t)}{d\lambda} \Big|_{\lambda=\lambda_i} = \frac{dR(\lambda, t)}{d\lambda} \Big|_{\lambda=\lambda_i} \quad (2.37)$$

dichas condiciones se cumplirán hasta la derivada de orden $m_i - 1$, a partir de ese momento, aparecen sumandos que no dependen de $(\lambda - \lambda_i)$ y que por lo tanto no se anularán.

En el apéndice E se pueden observar algunos casos particulares más comunes en el cálculo de la matriz de transición.

CAPÍTULO 3

MODELO ALGEBRAICO

3 1

Introducción

En este capítulo, se expondrán tanto los desarrollos matemáticos como los fundamentos en los que se basa la técnica de identificación algebraica utilizada para estimar los parámetros característicos de una señal sinusoidal sesgada y perturbada. El cual mediante el uso del método derivativo se consigue obtener el modelo de estado que define el comportamiento del sistema. Todo este capítulo está fuertemente ligado al artículo de Fliess y Sira-Ramírez [5]. De forma complementaria, este método también es explicado en los artículos de Trapero, Sira-Ramírez y Feliu-Batlle [23], Fliess y Glad [6] y García-Rodríguez y Cortes-Romero [7].

3 2

Marco matemático

Sea k el campo de números reales o complejos, se define $k[d/dt]$ como anillo de operadores diferenciales lineales o anillos diferenciales de la forma siguiente:

$$\sum_{j=0}^{finite} c_v \frac{d^j}{dt^j}, \quad c_v \in k. \quad (3.1)$$

Se considera un conjunto finito de valores indeterminados (y_1, \dots, y_m) como valores de salida asociados al anillo de operadores diferenciales lineales, es decir, estos valores representan las variables que describen las señales en el dominio del tiempo, siendo necesarias para extraer los parámetros desconocidos.

La manera más común en la que los parámetros aparecen en los coeficientes de una ecuación diferencial, incluyendo las expresiones algebraicas que involucran, es por medio de operaciones matemáticas, como las sumas, multiplicaciones, raíces, etc., es decir, las expresiones de los parámetros surgen como soluciones de las ecuaciones algebraicas que involucran dichos parámetros, siendo necesario despejar sus expresiones. La complejidad de obtener dichas expresiones viene dada por una extensión algebraica finita del campo de los números reales o complejos, denotada como K .

Sea K una extensión algebraica finita del campo $k(\Theta)$ de los números reales o complejos, generados por un conjunto finito $\Theta = (\theta_1, \dots, \theta_r)$, de parámetros constantes desconocidos.

Sea S/K una extensión algebraica finita diferencialmente generada, donde una señal es un elemento de S . Sea L/K una extensión algebraica diferencial y N un subconjunto de L . El sobrecampo diferencial de K generado por N se escribe como $K\langle N \rangle$.

Se dice que el conjunto Θ de parámetros desconocidos es algebraicamente identificable, si, y solo si, cualquier componente de Θ pertenece al campo $k\langle t, y \rangle$. Y se dice que es linealmente identificable cuando se cumpla:

$$P \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_r \end{pmatrix} = Q \quad (3.2)$$

Donde:

- P y Q son matrices $r \times r$ y $r \times 1$ respectivamente,
- las entradas de P y Q pertenecen a $\text{span}_{k[d/dt]}(1, y)$,
- $\det(P) \neq 0$

donde $\text{span}_{k[d/dt]}(1, y)$ es el conjunto de todas las combinaciones lineales de $(1, y_1, \dots, y_m)$, por ejemplo $D_0 \cdot 1 + \sum_{i=1}^m D_i y_i$. Los coeficientes D_i pertenecen a $k[d/dt]$. Por tanto, D_i es un operador diferencial de la forma

$$\sum_{j=1}^n \left(\frac{\sum_{k=0}^{k_j} a_{k,j} t^k}{\sum_{l=0}^{l_j} b_{l,j} t^l} \right) \quad (3.3)$$

con $a_{k,j}, b_{l,j} \in k$.

Se dice que el conjunto $\Theta = (\theta_1, \dots, \theta_r)$ de parámetros desconocidos es débilmente identificable linealmente si, y solo si, existe un conjunto finito $\Theta' = (\theta'_1, \dots, \theta'_{q'})$ de parámetros desconocidos relacionados tales que:

- Θ' es linealmente identificable.
- Los componentes de Θ' , (resp. Θ) son algebraicos sobre $C(\Theta)$ (resp. $C(\Theta')$)

CAPÍTULO 4

RESOLUCIÓN DE PARÁMETROS DE UNA SEÑAL SINUSOIDAL RUIDOSA Y SESGADA

4.1

Introducción

Una vez descrito todo el marco matemático, este se va a emplear para encontrar la solución a la estimación de los parámetros característicos de una señal sinusoidal.

La señal sinusoidal ruidosa y sesgada es definida por:

$$y(t) = A \sin(\omega t + \phi) + K + \xi(t) \quad (4.1)$$

donde el parámetro K corresponde al offset, es decir, la componente continua que presenta la señal, siendo un parámetro que se puede estimar; mientras que $\xi(t)$ corresponde con el ruido que presenta la señal, siendo un valor totalmente aleatorio. En este caso $\xi(t)$ será tratado como un ruido blanco de media cero.

El objetivo es estimar los parámetros desconocidos de la amplitud (A), frecuencia (ω), fase (ϕ) y offset (K), con el objetivo de recrear con la mayor precisión posible la señal original. Para obtener los diferentes parámetros se aplicará el método algebraico, donde dichas ecuaciones de estado se implementarán en Simulink, gracias al paquete de 'Control de sistemas', y posteriormente se usará el método general para encontrar la solución al modelo de estado, que se implementará tanto en Simulink como en Matlab. De forma alternativa se propondrá un método basado en la minimización del error por mínimos cuadrados para la amplitud, fase y offset. En todas las representaciones se observará y se comparará tanto su evolución respecto al tiempo, como el tiempo de ejecución para un número de muestras fijo, con el objetivo de determinar cuál de los diferentes métodos es más válido, considerando que nuestro algoritmo tiene que ser lo más eficiente posible.

Los datos con los que se realizarán las pruebas son: una amplitud $A = 20$, una velocidad angular $\omega = 5$ rad/s, una fase $\phi = 1$ rad, un offset $K = 10$ y por supuesto un ruido nulo $\xi(t) = 0$. Se procesarán los datos durante 2 segundos con una frecuencia de muestreo $f_s = 1$ kHz ($T_s = 0.001$ seg.). Además, el sistema tendrá unas condiciones iniciales nulas.

4 2**Estimación algebraica de la frecuencia****4 2 1 Algoritmos****4 2 1 1 Modelo de estado**

Se parte de la ecuación original (4.1), considerando la señal libre de ruido, $\xi(t) = 0$:

$$u(t) = A \sin(\omega t + \phi) + K \quad (4.2)$$

Se realiza la transformada de Laplace de la señal sinusoidal:

$$U(s) = \frac{A\omega \cos(\phi)}{s^2 + \omega^2} + \frac{sA \sin(\phi)}{s^2 + \omega^2} + \frac{K}{s} \quad (4.3)$$

donde se supone que $-\pi < \phi < \pi$.

Para simplificar los cálculos, se definen unos parámetros intermedios:

$$\begin{aligned} p_3 &= A\omega \cos(\phi) \\ p_2 &= A \sin(\phi) \\ p_1 &= \omega^2 \end{aligned} \quad (4.4)$$

De forma que los parámetros característicos vienen definidos por:

$$\begin{aligned} A &= \sqrt{\frac{p_3^2}{p_1} + p_2^2} \\ \omega &= +\sqrt{p_1} \\ \phi &= \tan^{-1}\left(\frac{p_2\sqrt{p_1}}{p_3}\right) \end{aligned} \quad (4.5)$$

Se reescribe la ecuación (4.3) con respecto a los nuevos parámetros:

$$U(s) = \frac{p_3}{s^2 + p_1} + \frac{sp_2}{s^2 + p_1} + \frac{K}{s} \quad (4.6)$$

Se multiplica la expresión (4.6) por $(s^2 + p_1)$ y s , obteniendo la expresión final con la que se van a estimar los diferentes parámetros:

$$\begin{aligned} s(s^2 + p_1)U(s) &= sp_3 + s^2p_2 + (s^2 + p_1)K \\ s^3U(s) + sp_1U(s) &= sp_3 + s^2p_2 + s^2K + p_1K \end{aligned} \quad (4.7)$$

Primero se va a estimar el parámetro p_1 . El motivo se debe a que dicho parámetro no depende del valor de ningún otro; y además, el valor de la frecuencia solamente depende de p_1 , obteniendo directamente su valor. Para ello, mediante el método algebraico, se deriva sucesivamente hasta eliminar los términos desconocidos p_2 , p_3 y K :

$$\begin{aligned} \frac{d}{ds} &\rightarrow [s^3 + sp_1]\dot{U}(s) + [3s^2 + p_1]U(s) = p_3 + 2sp_2 + 2sK \\ \frac{d^2}{ds^2} &\rightarrow [s^3 + sp_1]\ddot{U}(s) + [6s^2 + 2p_1]\dot{U}(s) + 6sU(s) = 2p_2 + 2K \\ \frac{d^3}{ds^3} &\rightarrow [s^3 + sp_1]\dddot{U}(s) + [9s^2 + 3p_1]\ddot{U}(s) + 18s\dot{U}(s) + 6U(s) = 0 \end{aligned} \quad (4.8)$$

Se multiplica la tercera derivada por s^{-3} :

$$[1 + s^{-2}p_1]\ddot{U}(s) + [9s^{-1} + 3s^{-3}p_1]\ddot{U}(s) + 18s^{-2}\dot{U}(s) + 6s^{-3}U(s) = 0 \quad (4.9)$$

Y se despeja el término de p_1 de la ecuación:

$$p_1 = -\frac{\ddot{U}(s) + 9s^{-1}\ddot{U}(s) + 18s^{-2}\dot{U}(s) + 6s^{-3}U(s)}{s^{-2}\ddot{U}(s) + 3s^{-3}\dot{U}(s)} \quad (4.10)$$

Una vez obtenida la expresión para p_1 , se divide en dos ecuaciones. La del numerador n_1 y la del denominador d_1 , las cuales van a ser tratadas de forma independiente:

$$p_1 = \frac{n_1(s)}{d_1(s)} \quad (4.11)$$

$$n_1(s) = -[\ddot{U}(s) + 9s^{-1}\ddot{U}(s) + 18s^{-2}\dot{U}(s) + 6s^{-3}U(s)] \quad (4.12)$$

$$d_1(s) = s^{-2}\ddot{U}(s) + 3s^{-3}\dot{U}(s) \quad (4.13)$$

Ambas ecuaciones se trasladan al dominio del tiempo como una combinación lineal de integrales:

$$n_1(t) = t^3u(t) - 9 \int t^2u(t) + 18 \int^{(2)} tu(t) - 6 \int^{(3)} u(t) \quad (4.14)$$

$$d_1(t) = - \int^{(2)} t^3u(t) + 3 \int^{(3)} t^2u(t) \quad (4.15)$$

A partir de las ecuaciones que definen el comportamiento de p_1 en el dominio del tiempo, se obtienen las ecuaciones de estado para el numerador y el denominador.

Las variables de estado se definen como salida de los integrados; para el numerador:

$$n_1(t) = t^3u(t) + x_1 \quad (4.16)$$

$$\begin{aligned} \dot{x}_1(t) &= -9t^2u(t) + x_2 \\ \dot{x}_2(t) &= 18tu(t) + x_3 \\ \dot{x}_3(t) &= -6u(t) \end{aligned} \quad (4.17)$$

mientras que para el denominador:

$$d_1(t) = x_4 \quad (4.18)$$

$$\begin{aligned} \dot{x}_4(t) &= x_5 \\ \dot{x}_5(t) &= -t^3u(t) + x_6 \\ \dot{x}_6(t) &= 3t^2u(t) \end{aligned} \quad (4.19)$$

La sintaxis que define un modelo de estado, viene dada por las siguientes ecuaciones:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (4.20)$$

Se escribe de forma matricial y se adecúa con respecto a la definición, donde las ecuaciones de estado son:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} -9t^2 \\ 18t \\ -6 \end{pmatrix} u(t) \quad (4.21)$$

$$\begin{pmatrix} \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 0 \\ -t^3 \\ 3t^2 \end{pmatrix} u(t) \quad (4.22)$$

Mientras que las ecuaciones de salida son:

$$n_1(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} t^3 \end{pmatrix} u(t) \quad (4.23)$$

$$d_1(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} \quad (4.24)$$

Se identifican las matrices del modelo de estado para el numerador:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} -9t^2 \\ 18t \\ -6 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}; \quad D = \begin{pmatrix} t^3 \end{pmatrix} \quad (4.25)$$

y el denominador:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} 0 \\ -t^3 \\ 3t^2 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}; \quad D = \begin{pmatrix} 0 \end{pmatrix} \quad (4.26)$$

4.2.1.2 Solución analítica

Una vez obtenido el modelo de estado que define el comportamiento de p_1 , se encuentra su solución mediante el método general. El primer paso es encontrar la matriz de transición, donde la matriz A es idéntica tanto en el modelo de estado del numerador como del denominador; además, es invariante en el tiempo, por lo que se puede aplicar el método de Cayley-Hamilton.

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (4.27)$$

Primero se determinan los autovalores de la matriz:

$$Q(\lambda) = |\lambda I - A| = 0 \quad \rightarrow \quad \begin{pmatrix} \lambda & -1 & 0 \\ 0 & \lambda & -1 \\ 0 & 0 & \lambda \end{pmatrix} = 0 \quad \rightarrow \quad \lambda^3 = 0 \quad (4.28)$$

El sistema tiene un autovalor en $\lambda = 0$ (triple).

Aplicando el método general:

$$\Phi(t) = P(A) = e^{A(t-t_0)} = R(A) \quad (4.29)$$

y conociendo que el sistema solo tiene un autovalor triple, queda de la siguiente forma:

$$\begin{aligned} \Phi(t) &= e^{A(t-t_0)} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 \\ P(\lambda) &= e^{\lambda(t-t_0)} \\ R(\lambda) &= \alpha_0 + \alpha_1 \lambda + \alpha_2 \lambda^2 \end{aligned} \quad (4.30)$$

Se realizan las diferentes operaciones:

$$\begin{aligned} P(\lambda)|_{\lambda=0} = R(\lambda)|_{\lambda=0} &\rightarrow e^{\lambda(t-t_0)}|_{\lambda=0} = \alpha_0 + \alpha_1 \lambda + \alpha_2 \lambda^2|_{\lambda=0} \rightarrow \alpha_0 = 0 \\ \frac{dP(\lambda)}{d\lambda}|_{\lambda=0} = \frac{dR(\lambda)}{d\lambda}|_{\lambda=0} &\rightarrow (t-t_0)e^{\lambda(t-t_0)}|_{\lambda=0} = \alpha_1 + 2\alpha_2 \lambda|_{\lambda=0} \\ &\rightarrow \alpha_1 = (t-t_0) \\ \frac{dP^2(\lambda)}{d\lambda^2}|_{\lambda=0} = \frac{dR^2(\lambda)}{d\lambda^2}|_{\lambda=0} &\rightarrow (t-t_0)^2 e^{\lambda(t-t_0)}|_{\lambda=0} = 2\alpha_2|_{\lambda=0} \rightarrow \alpha_2 = \frac{(t-t_0)^2}{2} \end{aligned} \quad (4.31)$$

Y se construye la matriz de transición para cualquier valor de t_0 :

$$\begin{aligned} \Phi(t, t_0) &= e^{A(t-t_0)} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 \\ \Phi(t, t_0) &= e^{A(t-t_0)} = \begin{pmatrix} 1 & (t-t_0) & \frac{(t-t_0)^2}{2} \\ 0 & 1 & (t-t_0) \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (4.32)$$

Una vez obtenida la matriz de transición se aplica el método general para encontrar la solución a la ecuación completa:

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)B(\tau)U(\tau)d\tau \quad (4.33)$$

A partir de aquí, los cálculos para el modelo de estado del numerador y el denominador se tratan por separado, pues la matriz $B(t)$ es diferente.

En el caso del numerador:

$$B(t) = \begin{pmatrix} -9t^2 \\ 18t \\ -6 \end{pmatrix} \quad (4.34)$$

Sustituyendo la matriz $B(t)$ y la matriz de transición en la ecuación general (4.33), se obtiene el siguiente resultado final:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} = \begin{pmatrix} 1 & (t-t_0) & \frac{(t-t_0)^2}{2} \\ 0 & 1 & (t-t_0) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{pmatrix} + \int_{t_0}^t \begin{pmatrix} -9\tau^2 + 18\tau(t-\tau) - 6\frac{(t-\tau)^2}{2} \\ 18\tau - 6(t-\tau) \\ -6 \end{pmatrix} U(\tau) d\tau \quad (4.35)$$

Conociendo la salida de $n_1(t)$:

$$n_1(t) = t^3 u(t) + x_1(t) \quad (4.36)$$

Se puede observar que solo depende del parámetro $x_1(t)$, siendo solamente necesario desarrollar dicho término:

$$\begin{aligned} x_1(t) &= x_1(0) + x_2(0)(t-t_0) + x_3(0)\frac{(t-t_0)^2}{2} - 30 \int_{t_0}^t \tau^2 U(\tau) d\tau + \\ &+ 24t \int_{t_0}^t \tau U(\tau) d\tau - 3t^2 \int_{t_0}^t U(\tau) d\tau \end{aligned} \quad (4.37)$$

Sustituyendo en la ecuación de salida e implementando las condiciones iniciales nulas, se obtiene la expresión de $n_1(t)$:

$$n_1(t) = t^3 u(t) - 30 \int_{t_0}^t \tau^2 U(\tau) d\tau + 24t \int_{t_0}^t \tau U(\tau) d\tau - 3t^2 \int_{t_0}^t U(\tau) d\tau \quad (4.38)$$

Se repite el mismo proceso para el denominador, donde la matriz $B(t)$ es:

$$B(t) = \begin{pmatrix} 0 \\ -t^3 \\ 3t^2 \end{pmatrix} \quad (4.39)$$

Sustituyendo la matriz de transición y la matriz $B(t)$ en la ecuación general, se obtiene el siguiente resultado:

$$\begin{pmatrix} x_4(t) \\ x_5(t) \\ x_6(t) \end{pmatrix} = \begin{pmatrix} 1 & (t-t_0) & \frac{(t-t_0)^2}{2} \\ 0 & 1 & (t-t_0) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_4(0) \\ x_5(0) \\ x_6(0) \end{pmatrix} + \int_{t_0}^t \begin{pmatrix} -\tau^3(t-\tau) + 3\tau^2\frac{(t-\tau)^2}{2} \\ -\tau^3 + 3\tau^2(t-\tau) \\ 3\tau^2 \end{pmatrix} U(\tau) d\tau \quad (4.40)$$

Conociendo la salida de $d_1(t)$:

$$d_1(t) = x_4(t) \quad (4.41)$$

Se puede observar que la salida solo depende del parámetro $x_4(t)$, por lo que solo se desarrolla dicho término:

$$\begin{aligned} x_4(t) = & x_4(0) + x_5(0)(t - t_0) + x_6(0)\frac{(t - t_0)^2}{2} + \frac{5}{2} \int_{t_0}^t \tau^4 U(\tau) d\tau - 4t \int_{t_0}^t \tau^3 U(\tau) d\tau \\ & + \frac{3}{2} t^2 \int_{t_0}^t \tau^2 U(\tau) d\tau \end{aligned} \quad (4.42)$$

Sustituyendo en la ecuación de salida y aplicando las condiciones iniciales nulas, se obtiene la expresión de $d_1(t)$:

$$d_1(t) = \frac{5}{2} \int_{t_0}^t \tau^4 U(\tau) d\tau - 4t \int_{t_0}^t \tau^3 U(\tau) d\tau + \frac{3}{2} t^2 \int_{t_0}^t \tau^2 U(\tau) d\tau \quad (4.43)$$

Gracias a las ecuaciones que definen el comportamiento de $n_1(t)$ y $d_1(t)$, se obtiene el valor del parámetro p_1 :

$$p_1 = \frac{t^3 u(t) - 30 \int_{t_0}^t \tau^2 U(\tau) d\tau + 24t \int_{t_0}^t \tau U(\tau) d\tau - 3t^2 \int_{t_0}^t U(\tau) d\tau}{\frac{5}{2} \int_{t_0}^t \tau^4 U(\tau) d\tau - 4t \int_{t_0}^t \tau^3 U(\tau) d\tau + \frac{3}{2} t^2 \int_{t_0}^t \tau^2 U(\tau) d\tau} \quad (4.44)$$

4.2.2 Implementación de las simulaciones de las ecuaciones de estado

En este apartado se implementarán las ecuaciones de estado para el parámetro p_1 en Simulink gracias al bloque 'Espacio de estado variable', que se encuentra dentro de la toolbox de sistemas de control. Su sintaxis corresponde a la del modelo de estado y su estructura interna se puede ver en la figura F.1 del apéndice F.

El bloque de 'Espacio de estado variable' tiene la particularidad de que las matrices tienen que ser invariantes en el tiempo, teniendo que transformar la entrada $u(t)$, como un vector que incluya las combinaciones necesarias entre la entrada $u(t)$ y la variable de tiempo t :

$$u(t) = \begin{pmatrix} t^3 u(t) \\ t^2 u(t) \\ t u(t) \\ u(t) \end{pmatrix} \quad (4.45)$$

Modificando las matrices que definen el modelo de estado para acoplarse al nuevo formato de entrada. Siendo para el numerador:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} & B &= \begin{pmatrix} 0 & -9 & 0 & 0 \\ 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & -6 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \\ C &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}; & D &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (4.46)$$

y el denominador:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad (4.47)$$

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

Una vez definidas las matrices de estado, se implementan en Simulink. En el programa, se distinguen los modelos de estado para el numerador y el denominador. Además, se ha aprovechado la ventaja de que la frecuencia se puede obtener directamente con el parámetro p_1 , mediante su definición en las ecuaciones (4.5), devolviendo como argumento de salida la frecuencia en radianes. El esquema se puede ver en la figura F.2 del apéndice F.

Se pasa como argumento de entrada la señal definida en la introducción, dando como resultado:

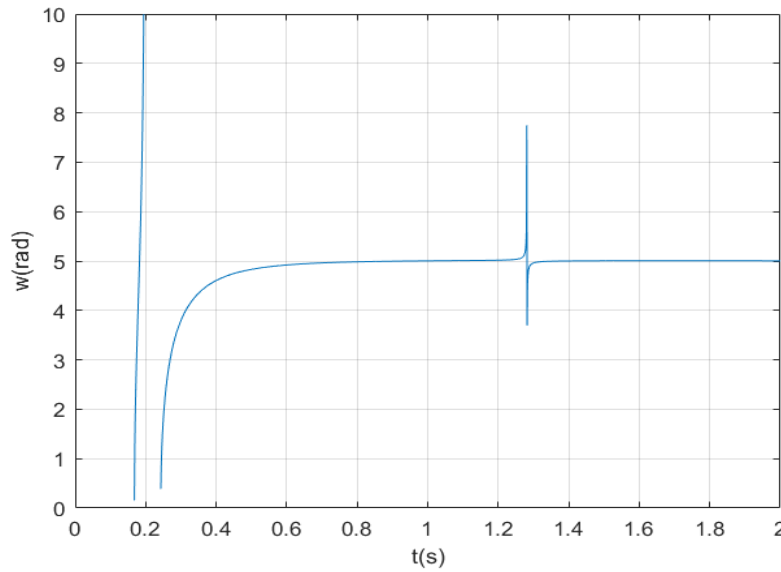


Figura 4.1: Frecuencia del modelo de estado mediante Simulink

El tiempo aproximado de ejecución del programa es de 0.56 segundos.

4.2.3 Implementación de la simulación de las soluciones analíticas

A continuación, se va a implementar la ecuación de p_1 obtenida mediante la solución del modelo de estado en Simulink, donde el esquema se puede ver en la figura F.3 del apéndice F. La salida del sistema es:

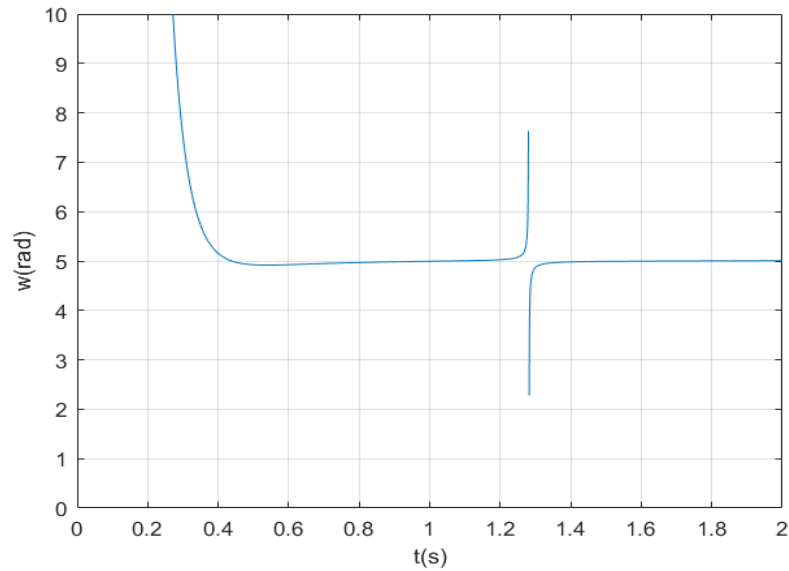


Figura 4.2: Frecuencia del modelo analítico mediante Simulink

El tiempo de ejecución del programa es alrededor de 0.90 segundos.

Si la ecuación se implementa en el entorno de Matlab, el resultado es:

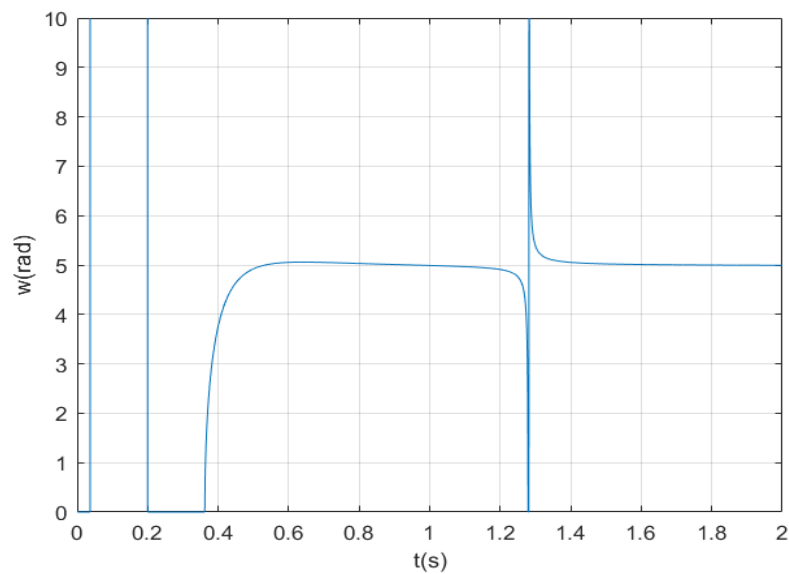


Figura 4.3: Frecuencia del modelo analítico mediante Matlab

Ejecutándose en un tiempo medio de 0.19 segundos.

Si se comparan los tres resultados obtenidos:

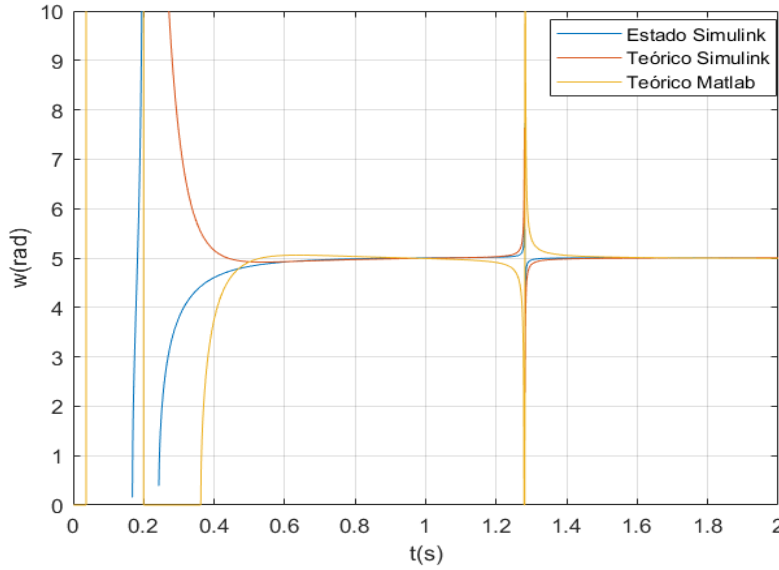


Figura 4.4: Comparativa entre métodos de obtención de la frecuencia

Se puede observar que su comportamiento es similar en los tres modelos.

Donde sí hay una gran diferencia es en el tiempo de ejecución, donde el mejor método es el modelo basado en la solución del modelo de estado implementado en Matlab, pero que en el caso de usar la herramienta de Simulink, el método más adecuado es el basado en el modelo de estado.

4.3

Estimación algebraica de la amplitud y fase

4.3.1 Algoritmos

4.3.1.1 Modelos de estados

Para la estimación de la amplitud y la fase es necesario conocer las variables intermedias p_1 , p_2 y p_3 , realizando el mismo procedimiento sobre dichas variables, es decir, partiendo de la transformada de Laplace (4.6), se aplica el método algebraico para despejar las diferentes variables a través de su derivada. En este punto, se asume que se conoce la variable p_1 .

En primer lugar, se despeja la variable intermedia p_3 , ya que únicamente depende de la variable conocida p_1 , multiplicando la ecuación (4.6) por s :

$$sU(s) = \frac{sp_3}{s^2 + p_1} + \frac{s^2 p_2}{s^2 + p_1} + K \quad (4.48)$$

Se deriva con respecto a s , para eliminar el término K :

$$U(s) + s\dot{U}(s) = p_3 \frac{(s^2 + p_1) - 2s^2}{(s^2 + p_1)^2} + p_2 \frac{2s(s^2 + p_1) - 2s^3}{(s^2 + p_1)^2} \quad (4.49)$$

Se multiplica por $(s^2 + p_1)^2$:

$$(s^2 + p_1)^2[U(s) + s\dot{U}(s)] = -s^2p_3 + p_1p_3 + 2sp_1p_2 \quad (4.50)$$

Se deriva sucesivamente con respecto a s hasta eliminar el término desconocido p_2 :

$$\begin{aligned} \frac{d}{ds} &\rightarrow (4s^3 + 4sp_1)[U(s) + s\dot{U}(s)] + (s^2 + p_1)^2[2\dot{U}(s) + s\ddot{U}(s)] = -2sp_3 + 2p_2p_1 \\ \frac{d^2}{ds^2} &\rightarrow s^5\ddot{U}(s) + 11s^4\ddot{U}(s) + s^3(28\dot{U}(s) + 2p_1\ddot{U}(s)) + s^2(12U(s) + 14p_1\ddot{U}(s)) \\ &\quad + s(20p_1\dot{U}(s) + p_1^2\ddot{U}(s)) + 4p_1U(s) + 3p_1^2\ddot{U}(s) = -2p_3 \end{aligned} \quad (4.51)$$

Se multiplica por s^{-5} :

$$\begin{aligned} \ddot{U}(s) + 11s^{-1}\ddot{U}(s) + s^{-2}(28\dot{U}(s) + 2p_1\ddot{U}(s)) + s^{-3}(12U(s) + 14p_1\ddot{U}(s)) \\ + s^{-4}(20p_1\dot{U}(s) + p_1^2\ddot{U}(s)) + s^{-5}(4p_1U(s) + 3p_1^2\ddot{U}(s)) = -2s^{-5}p_3 \end{aligned} \quad (4.52)$$

Por último, se despeja la variable p_3 , la cual solo depende de p_1 :

$$\begin{aligned} p_3 = & -\frac{1}{2s^{-5}}[\ddot{U}(s) + 11s^{-1}\ddot{U}(s) + s^{-2}(28\dot{U}(s) + 2p_1\ddot{U}(s)) + s^{-3}(12U(s) + 14p_1\ddot{U}(s)) \\ & + s^{-4}(20p_1\dot{U}(s) + p_1^2\ddot{U}(s)) + s^{-5}(4p_1U(s) + 3p_1^2\ddot{U}(s))] \end{aligned} \quad (4.53)$$

Una vez obtenida la expresión para p_3 , se divide en el numerador n_3 , y el denominador d_3 , y se tratan de forma independiente:

$$p_3 = \frac{n_3(s)}{d_3(s)} \quad (4.54)$$

$$\begin{aligned} n_3(s) = & \ddot{U}(s) + 11s^{-1}\ddot{U}(s) + s^{-2}(28\dot{U}(s) + 2p_1\ddot{U}(s)) + s^{-3}(12U(s) + 14p_1\ddot{U}(s)) \\ & + s^{-4}(20p_1\dot{U}(s) + p_1^2\ddot{U}(s)) + s^{-5}(4p_1U(s) + 3p_1^2\ddot{U}(s)) \end{aligned} \quad (4.55)$$

$$d_3(s) = -2s^{-5} \quad (4.56)$$

Ambas ecuaciones se trasladan al dominio del tiempo:

$$n_3(t) = -t^3u(t) + 11 \int t^2u(t) - 28 \int^{(2)} tu(t) - 2p_1 \int^{(2)} t^3u(t) \quad (4.57)$$

$$\begin{aligned} & + 12 \int^{(3)} u(t) + 14p_1 \int^{(3)} t^2u(t) - 20p_1 \int^{(4)} tu(t) \\ & - p_1^2 \int^{(4)} t^3u(t) + 4p_1 \int^{(5)} u(t) + 3p_1^2 \int^{(5)} t^2u(t) \\ d_3(t) = & -\frac{t^4}{12} \end{aligned} \quad (4.58)$$

Una vez obtenidas las ecuaciones que definen el comportamiento de p_3 , se obtienen las ecuaciones de estado correspondientes obteniendo las variables de estado como

salida de los integrados. Para el numerador:

$$n_3(t) = -t^3 u(t) + x_1 \quad (4.59)$$

$$\begin{aligned} \dot{x}_1(t) &= 11t^2 u(t) + x_2 \\ \dot{x}_2(t) &= -28tu(t) - 2p_1 t^3 u(t) + x_3 \\ \dot{x}_3(t) &= 12u(t) + 14p_1 t^2 u(t) + x_4 \\ \dot{x}_4(t) &= -20p_1 t u(t) - p_1^2 t^3 u(t) + x_5 \\ \dot{x}_5(t) &= 4p_1 u(t) + 3p_1^2 t^2 u(t) \end{aligned} \quad (4.60)$$

mientras que para el denominador:

$$d_3(t) = -\frac{t^4}{12} \quad (4.61)$$

El modelo de estado se escribe de forma matricial, donde las ecuaciones de estado son:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} + \begin{pmatrix} 11t^2 \\ -28t - 2p_1 t^3 \\ 12 + 14p_1 t^2 \\ -20p_1 t - p_1^2 t^3 \\ 4p_1 + 3p_1^2 t^2 \end{pmatrix} u(t) \quad (4.62)$$

Mientras que las ecuaciones de salida:

$$n_3(t) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} + \begin{pmatrix} -t^3 \end{pmatrix} u(t) \quad (4.63)$$

$$d_3(t) = -\frac{t^4}{12} \quad (4.64)$$

Como se puede observar, el denominador carece de ecuaciones de estado, dependiendo únicamente de su ecuación de salida, siendo solamente necesario identificar las matrices de estado para el numerador:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; & B &= \begin{pmatrix} 11t^2 \\ -28t - 2p_1 t^3 \\ 12 + 14p_1 t^2 \\ -20p_1 t - p_1^2 t^3 \\ 4p_1 + 3p_1^2 t^2 \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}; & D &= \begin{pmatrix} -t^3 \end{pmatrix} \end{aligned} \quad (4.65)$$

Una vez obtenido el modelo de estado para p_3 , el siguiente paso es obtener el término p_2 , para conocer el comportamiento tanto de la amplitud como de la fase. Se parte de la ecuación:

$$(s^2 + p_1)^2[U(s) + s\dot{U}(s)] = -s^2p_3 + p_1p_3 + 2sp_2p_1 \quad (4.66)$$

A la hora de hallar p_2 , se tiene en cuenta que en este punto se conoce tanto p_1 como p_3 , por lo que se desarrolla este término:

$$2sp_2p_1 = s^4U(s) + 2p_1s^2U(s) + p_1^2U(s) + s^5\dot{U}(s) + 2p_1s^3\dot{U}(s) + p_1^2s\dot{U}(s) + s^2p_3 - p_1p_3 \quad (4.67)$$

Se multiplica por s^{-5} :

$$2s^{-4}p_2p_1 = s^{-1}U(s) + 2p_1s^{-3}U(s) + p_1^2s^{-5}U(s) + \dot{U}(s) + 2p_1s^{-2}\dot{U}(s) + p_1^2s^{-4}\dot{U}(s) + s^{-3}p_3 - s^{-5}p_1p_3 \quad (4.68)$$

Y se despeja la variable p_2 :

$$p_2 = \frac{s^{-5}(p_1^2U(s) - p_1p_3) + p_1^2s^{-4}\dot{U}(s) + s^{-3}(p_3 + 2p_1U(s))}{2s^{-4}p_1} + \frac{2p_1s^{-2}\dot{U}(s) + s^{-1}U(s) + \dot{U}(s)}{2s^{-4}p_1} \quad (4.69)$$

Se divide la ecuación, una para el numerador y otra para el denominador:

$$p_2 = \frac{n_2(s)}{d_2(s)} \quad (4.70)$$

$$n_2(s) = \dot{U}(s) + s^{-1}U(s) + 2p_1s^{-2}\dot{U}(s) + s^{-3}(p_3 + 2p_1U(s)) + p_1^2s^{-4}\dot{U}(s) + s^{-5}(p_1^2U(s) - p_1p_3) \quad (4.71)$$

$$d_2(s) = 2s^{-4}p_1 \quad (4.72)$$

Ambas ecuaciones se trasladan al dominio del tiempo:

$$n_2(t) = -tu(t) + p_3\frac{t^2}{2} - p_1p_3\frac{t^4}{24} + \int u(t) - 2p_1 \int^{(2)} tu(t) + 2p_1 \int^{(3)} u(t) - p_1^2 \int^{(4)} tu(t) + p_1^2 \int^{(5)} u(t) \quad (4.73)$$

$$d_2(t) = -p_1\frac{t^3}{3} \quad (4.74)$$

Una vez obtenidas las ecuaciones que definen el comportamiento de p_2 , se obtienen las ecuaciones de estado correspondientes, obteniendo las variables de estado como salida de los integrados. Para el numerador:

$$n_2(t) = -tu(t) + p_3\frac{t^2}{2} - p_1p_3\frac{t^4}{24} + x_1 \quad (4.75)$$

$$\begin{aligned} \dot{x}_1(t) &= u(t) + x_2 \\ \dot{x}_2(t) &= -2p_1tu(t) + x_3 \\ \dot{x}_3(t) &= 2p_1u(t) + x_4 \\ \dot{x}_4(t) &= -p_1^2tu(t) + x_5 \\ \dot{x}_5(t) &= p_1^2u(t) \end{aligned} \quad (4.76)$$

mientras que para el denominador:

$$d_3(t) = -p_1 \frac{t^3}{3} \quad (4.77)$$

Se escribe de forma matricial, donde las ecuaciones de estado son:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} + \begin{pmatrix} 1 \\ -2p_1 t \\ 2p_1 \\ -p_1^2 t \\ p_1^2 \end{pmatrix} u(t) \quad (4.78)$$

Las ecuaciones de salida son:

$$n_2(t) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} + (-t) u(t) + p_3 \frac{t^2}{2} - p_1 p_3 \frac{t^4}{24} \quad (4.79)$$

$$d_2(t) = -p_1 \frac{t^3}{3} \quad (4.80)$$

Se identifican las matrices que definen el modelo de estado tanto para el numerador como para el denominador. Al igual que en p_3 , el denominador no tiene ecuaciones de estado, solo depende de la ecuación de salida. Además, se tiene que tener en cuenta que para la ecuación de salida del numerador se les suman dos constantes a las matrices. Las matrices que definen el modelo de estado para el numerador son:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; B = \begin{pmatrix} 1 \\ -2p_1 t \\ 2p_1 \\ -p_1^2 t \\ p_1^2 \end{pmatrix}; C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}; D = \begin{pmatrix} -t \end{pmatrix} \quad (4.81)$$

4.3.1.2 Soluciones analíticas

Para encontrar la solución a los modelos de estado que definen p_3 y p_2 , se tiene que encontrar la matriz de transición a partir de la matriz A . En este caso la matriz A es idéntica tanto para p_3 como para p_2 , es decir, comparten matriz de transición. Se realiza el mismo procedimiento que anteriormente, donde se determinan los autovalores de la matriz A :

$$Q(\lambda) = |\lambda I - A| = 0 \rightarrow \begin{pmatrix} \lambda & -1 & 0 & 0 & 0 \\ 0 & \lambda & -1 & 0 & 0 \\ 0 & 0 & \lambda & -1 & 0 \\ 0 & 0 & 0 & \lambda & -1 \\ 0 & 0 & 0 & 0 & \lambda \end{pmatrix} = 0 \rightarrow \lambda^5 = 0 \quad (4.82)$$

El sistema tiene un único autovalor en $\lambda = 0$ (quintuple).

Se aplica el método general, conociendo que el sistema solo tiene un autovalor de grado 5:

$$\begin{aligned}\Phi(t) &= e^{A(t-t_0)} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \alpha_3 A^3 + \alpha_4 A^4 \\ P(\lambda) &= e^{\lambda(t-t_0)} \\ R(\lambda) &= \alpha_0 + \alpha_1 \lambda + \alpha_2 \lambda^2 + \alpha_3 \lambda^3 + \alpha_4 \lambda^4\end{aligned}\quad (4.83)$$

Se realizan las diferentes operaciones:

$$\begin{aligned}P(\lambda)|_{\lambda=0} = R(\lambda)|_{\lambda=0} &\rightarrow e^{\lambda(t-t_0)}|_{\lambda=0} = \alpha_0 + \alpha_1 \lambda + \alpha_2 \lambda^2 + \alpha_3 \lambda^3 + \alpha_4 \lambda^4|_{\lambda=0} \rightarrow \alpha_0 = 0 \\ \frac{dP(\lambda)}{d\lambda}|_{\lambda=0} = \frac{dR(\lambda)}{d\lambda}|_{\lambda=0} &\rightarrow (t-t_0)e^{\lambda(t-t_0)}|_{\lambda=0} = \alpha_1 + 2\alpha_2 \lambda + 3\alpha_3 \lambda^2 + 4\alpha_4 \lambda^3|_{\lambda=0} \\ &\rightarrow \alpha_1 = (t-t_0) \\ \frac{dP^2(\lambda)}{d\lambda^2}|_{\lambda=0} = \frac{dR^2(\lambda)}{d\lambda^2}|_{\lambda=0} &\rightarrow (t-t_0)^2 e^{\lambda(t-t_0)}|_{\lambda=0} = 2\alpha_2 + 6\alpha_3 \lambda + 12\alpha_4 \lambda^2|_{\lambda=0} \\ &\rightarrow \alpha_2 = \frac{(t-t_0)^2}{2} \\ \frac{dP^3(\lambda)}{d\lambda^3}|_{\lambda=0} = \frac{dR^3(\lambda)}{d\lambda^3}|_{\lambda=0} &\rightarrow (t-t_0)^3 e^{\lambda(t-t_0)}|_{\lambda=0} = 6\alpha_3 + 24\alpha_4 \lambda|_{\lambda=0} \rightarrow \alpha_3 = \frac{(t-t_0)^3}{6} \\ \frac{dP^4(\lambda)}{d\lambda^4}|_{\lambda=0} = \frac{dR^4(\lambda)}{d\lambda^4}|_{\lambda=0} &\rightarrow (t-t_0)^4 e^{\lambda(t-t_0)}|_{\lambda=0} = 24\alpha_4|_{\lambda=0} \rightarrow \alpha_4 = \frac{(t-t_0)^4}{24}\end{aligned}\quad (4.84)$$

La matriz de transición para cualquier valor de t_0 es:

$$\begin{aligned}\Phi(t, t_0) &= e^{A(t-t_0)} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \alpha_3 A^3 + \alpha_4 A^4 \\ \Phi(t, t_0) &= e^{A(t-t_0)} = \begin{pmatrix} 1 & (t-t_0) & \frac{(t-t_0)^2}{2} & \frac{(t-t_0)^3}{6} & \frac{(t-t_0)^4}{24} \\ 0 & 1 & (t-t_0) & \frac{(t-t_0)^2}{2} & \frac{(t-t_0)^3}{6} \\ 0 & 0 & 1 & (t-t_0) & \frac{(t-t_0)^2}{2} \\ 0 & 0 & 0 & 1 & (t-t_0) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\end{aligned}\quad (4.85)$$

Una vez obtenida la matriz de transición, se desarrolla la ecuación general con $B(t)$ del numerador $n_3(t)$:

$$\begin{aligned}\begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{pmatrix} &= \begin{pmatrix} 1 & (t-t_0) & \frac{(t-t_0)^2}{2} & \frac{(t-t_0)^3}{6} & \frac{(t-t_0)^4}{24} \\ 0 & 1 & (t-t_0) & \frac{(t-t_0)^2}{2} & \frac{(t-t_0)^3}{6} \\ 0 & 0 & 1 & (t-t_0) & \frac{(t-t_0)^2}{2} \\ 0 & 0 & 0 & 1 & (t-t_0) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \\ x_5(0) \end{pmatrix} \\ &+ \int_{t_0}^t \begin{pmatrix} 1 & (t-\tau) & \frac{(t-\tau)^2}{2} & \frac{(t-\tau)^3}{6} & \frac{(t-\tau)^4}{24} \\ 0 & 1 & (t-\tau) & \frac{(t-\tau)^2}{2} & \frac{(t-\tau)^3}{6} \\ 0 & 0 & 1 & (t-\tau) & \frac{(t-\tau)^2}{2} \\ 0 & 0 & 0 & 1 & (t-\tau) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 11\tau^2 \\ -28\tau - 2p_1\tau^3 \\ 12 + 14p_1\tau^2 \\ -20p_1\tau - p_1^2\tau^3 \\ 4p_1 + 3p_1^2\tau^2 \end{pmatrix} U(\tau) d\tau\end{aligned}\quad (4.86)$$

Conociendo que la salida para $n_3(t)$ es:

$$n_3(t) = -t^3 u(t) + x_1(t) \quad (4.87)$$

Solo es necesario desarrollar el parámetro $x_1(t)$ de la ecuación general:

$$\begin{aligned} x_1(t) = & x_1(0) + x_2(0)(t - t_0) + x_3(0)\frac{(t - t_0)^2}{2} + x_4(0)\frac{(t - t_0)^3}{6} + z_5(0)\frac{(t - t_0)^4}{24} + \int_{t_0}^t [11\tau^2 \\ & + (-28\tau - 2p_1\tau^3)(t - \tau) + (12 + 14p_1\tau^2)\frac{(t - \tau)^2}{2} + (-20p_1\tau - p_1^2\tau^3)\frac{(t - \tau)^3}{6} \\ & + (4p_1 + 3p_1^2\tau^2)\frac{(t - \tau)^4}{24}]U(\tau)d\tau \end{aligned} \quad (4.88)$$

Se sustituye en la ecuación de salida y se implementan las condiciones iniciales nulas, obteniendo la expresión para $n_3(t)$:

$$\begin{aligned} n_3(t) = & -t^3 u(t) + \int_{t_0}^t [11\tau^2 + (-28\tau - 2p_1\tau^3)(t - \tau) + (12 + 14p_1\tau^2)\frac{(t - \tau)^2}{2} \\ & + (-20p_1\tau - p_1^2\tau^3)\frac{(t - \tau)^3}{6} + (4p_1 + 3p_1^2\tau^2)\frac{(t - \tau)^4}{24}]U(\tau)d\tau \end{aligned} \quad (4.89)$$

Gracias a las ecuaciones de salida que definen el comportamiento de $n_3(t)$ y $d_3(t)$, se obtiene p_3 :

$$\begin{aligned} p_3 = & -\frac{12}{t^4}[-t^3 u(t) + \int_{t_0}^t [11\tau^2 + (-28\tau - 2p_1\tau^3)(t - \tau) + (12 + 14p_1\tau^2)\frac{(t - \tau)^2}{2} \\ & + (-20p_1\tau - p_1^2\tau^3)\frac{(t - \tau)^3}{6} + (4p_1 + 3p_1^2\tau^2)\frac{(t - \tau)^4}{24}]U(\tau)d\tau] \end{aligned} \quad (4.90)$$

Una vez obtenida la solución para el parámetro p_3 , se aplica la ecuación general a la matriz $B(t)$ del numerador $n_2(t)$:

$$\begin{aligned} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{pmatrix} = & \begin{pmatrix} 1 & (t - t_0) & \frac{(t - t_0)^2}{2} & \frac{(t - t_0)^3}{6} & \frac{(t - t_0)^4}{24} \\ 0 & 1 & (t - t_0) & \frac{(t - t_0)^2}{2} & \frac{(t - t_0)^3}{6} \\ 0 & 0 & 1 & (t - t_0) & \frac{(t - t_0)^2}{2} \\ 0 & 0 & 0 & 1 & (t - t_0) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \\ x_5(0) \end{pmatrix} \\ & + \int_{t_0}^t \begin{pmatrix} 1 & (t - \tau) & \frac{(t - \tau)^2}{2} & \frac{(t - \tau)^3}{6} & \frac{(t - \tau)^4}{24} \\ 0 & 1 & (t - \tau) & \frac{(t - \tau)^2}{2} & \frac{(t - \tau)^3}{6} \\ 0 & 0 & 1 & (t - \tau) & \frac{(t - \tau)^2}{2} \\ 0 & 0 & 0 & 1 & (t - \tau) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2p_1\tau \\ 2p_1 \\ -p_1^2\tau \\ p_1^2 \end{pmatrix} U(\tau)d\tau \end{aligned} \quad (4.91)$$

Conociendo la salida para $n_2(t)$:

$$n_2(t) = -tu(t) + p_3\frac{t^2}{2} - p_1p_3\frac{t^4}{24} + x_1(t) \quad (4.92)$$

Solo es necesario desarrollar el término $x_1(t)$:

$$\begin{aligned} x_1(t) = & x_1(0) + x_2(0)(t - t_0) + x_3(0)\frac{(t - t_0)^2}{2} + x_4(0)\frac{(t - t_0)^3}{6} + x_5(0)\frac{(t - t_0)^4}{24} \\ & + \int_{t_0}^t [1 - 2p_1\tau(t - \tau) + 2p_1\frac{(t - \tau)^2}{2} - p_1^2\tau\frac{(t - \tau)^3}{6} \\ & + p_1^2\frac{(t - \tau)^4}{24}]U(\tau)d\tau \end{aligned} \quad (4.93)$$

Sustituyendo en la ecuación de salida e implementando las condiciones iniciales nulas, se obtiene la expresión $n_2(t)$:

$$\begin{aligned} n_2(t) = & -tu(t) + p_3\frac{t^2}{2} - p_1p_3\frac{t^4}{24} + \int_{t_0}^t [1 - 2p_1\tau(t - \tau) + 2p_1\frac{(t - \tau)^2}{2} \\ & - p_1^2\tau\frac{(t - \tau)^3}{6} + p_1^2\frac{(t - \tau)^4}{24}]U(\tau)d\tau \end{aligned} \quad (4.94)$$

Mientras que $d_2(t)$ es:

$$d_2(t) = -p_1\frac{t^3}{3} \quad (4.95)$$

Una vez obtenidas las ecuaciones que definen el comportamiento para $n_2(t)$ y $d_2(t)$, se obtiene el parámetro p_2 :

$$\begin{aligned} p_2 = & -p_1\frac{3}{t^3}[-tu(t) + p_3\frac{t^2}{2} - p_1p_3\frac{t^4}{24} + \int_{t_0}^t [1 - 2p_1\tau(t - \tau) + 2p_1\frac{(t - \tau)^2}{2} \\ & - p_1^2\tau\frac{(t - \tau)^3}{6} + p_1^2\frac{(t - \tau)^4}{24}]U(\tau)d\tau] \end{aligned} \quad (4.96)$$

4.3.2 Implementación de las simulaciones de las ecuaciones de estado

Se implementan las ecuaciones de estado en el entorno de Simulink, en ambos casos solo se encuentra un modelo de estado, el del numerador, debido a que el denominador solo posee la ecuación de salida.

Primero se implementan las ecuaciones de estado para p_3 , donde al igual que anteriormente, la matriz $B(t)$ del numerador es variante en el tiempo, teniendo que transformar la entrada al sistema como un vector que incluya las combinaciones necesarias entre la entrada $u(t)$ y la variable tiempo t :

$$u(t) = \begin{pmatrix} t^4u(t) \\ t^3u(t) \\ t^2u(t) \\ tu(t) \\ u(t) \end{pmatrix} \quad (4.97)$$

Modificando a su vez las matrices que definen el modelo de estado para acoplarse al nuevo formato:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; & B &= \begin{pmatrix} 0 & 0 & 11 & 0 & 0 \\ 0 & -2p_1 & 0 & -28 & 0 \\ 0 & 0 & 14p_1 & 0 & 12 \\ 0 & -p_1^2 & 0 & -20p_1 & 0 \\ 0 & 0 & 3p_1^2 & 0 & 4p_1 \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}; & D &= \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (4.98)$$

Se implementa en Simulink, donde el esquema se puede ver en la figura F.4 del apéndice F. La salida de la variable p_3 es la siguiente:

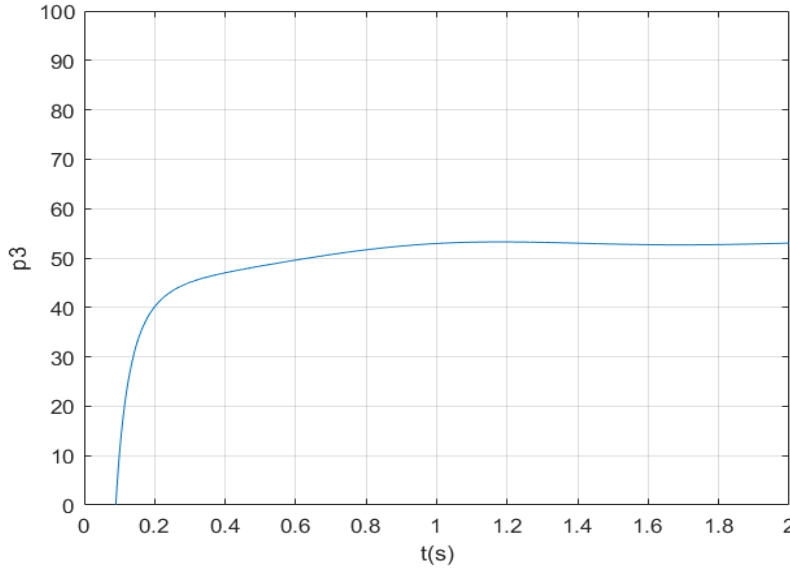


Figura 4.5: Variable p_3 del modelo de estado mediante Simulink

El tiempo de ejecución promedio del programa es de 0.55 segundos.

Se realiza el mismo procedimiento para el parámetro p_2 , donde la entrada al sistema se modifica siendo:

$$u(t) = \begin{pmatrix} tu(t) \\ u(t) \end{pmatrix} \quad (4.99)$$

A su vez, se acoplan las matrices al nuevo formato, quedando:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; & B &= \begin{pmatrix} 0 & 1 \\ -2p_1 & 0 \\ 0 & 2p_1 \\ -p_1^2 & 0 \\ 0 & p_1^2 \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}; & D &= \begin{pmatrix} -1 & 0 \end{pmatrix} \end{aligned} \quad (4.100)$$

Se implementa en Simulink, donde el esquema se puede ver en la figura F.6 del apéndice F. La salida de la variable p_2 es la siguiente:

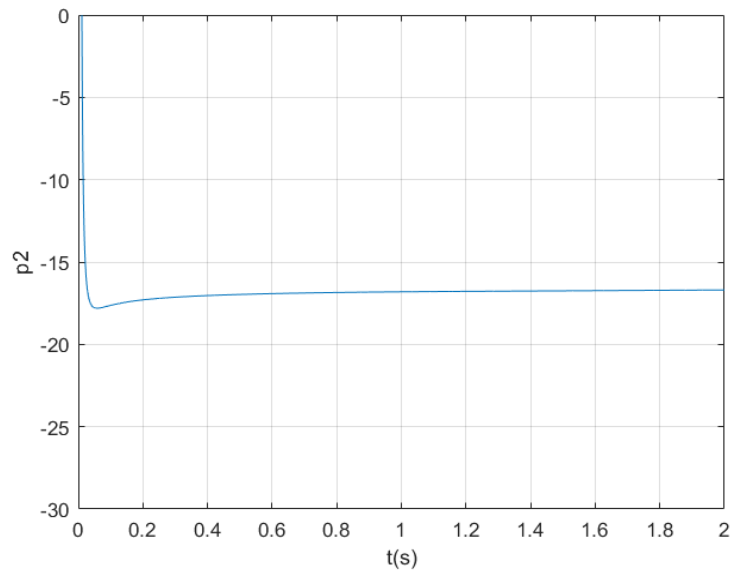


Figura 4.6: Variable p_2 del modelo de estado mediante Simulink

El tiempo de ejecución del programa ronda los 0.62 segundos.

4.3.3 Implementación de las simulaciones de las soluciones analíticas

Se implementa la solución analítica de p_3 en Simulink, donde el esquema se puede ver en la figura F.5 del apéndice F. La salida del sistema es:

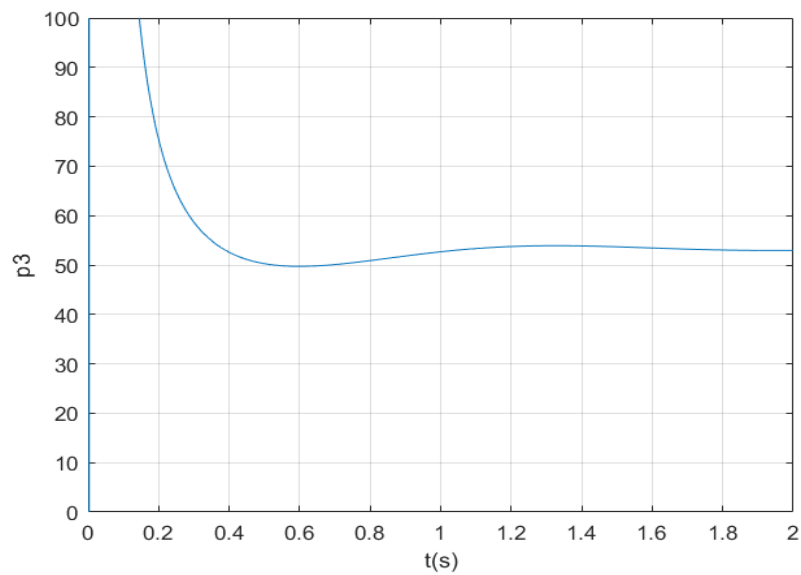


Figura 4.7: Variable p_3 del modelo analítico mediante Simulink

El tiempo de ejecución del programa se encuentra alrededor de los 0.85 segundos.

Si la ecuación se implementa en el entorno de Matlab, el resultado es el siguiente:

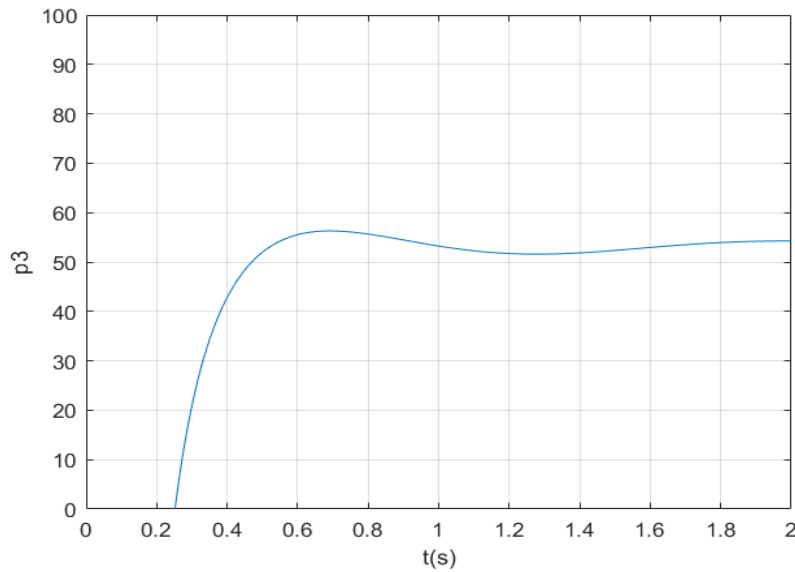


Figura 4.8: Variable p_3 del modelo analítico mediante Matlab

Con un tiempo de ejecución aproximado de 0.20 segundos.

Si se comparan las tres señales obtenidas para el parámetro p_3 :

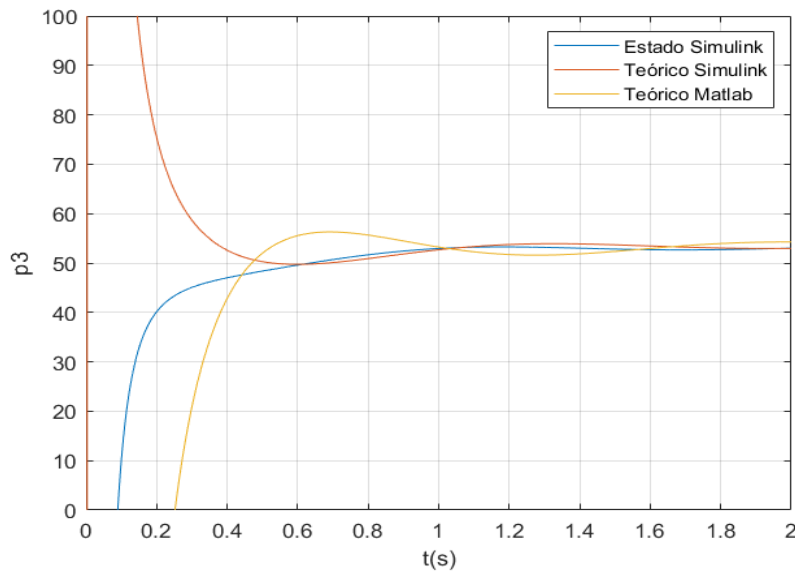


Figura 4.9: Comparativa entre métodos de obtención de la variable p_3

Se puede observar que su comportamiento es similar en los tres modelos, entrando en la zona de convergencia a partir del mismo punto. Sin embargo, si se observan los tiempos de ejecución, el que menor tiempo presenta es el modelo analítico en Matlab.

Se implementa en Simulink la ecuación de p_2 , donde el esquema se puede ver en la figura F.7 del apéndice F. La salida del sistema es:

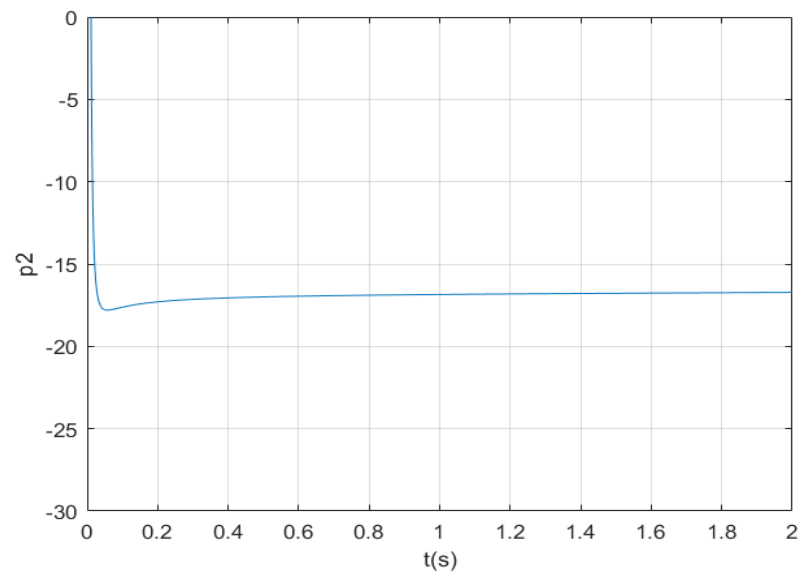


Figura 4.10: Variable p_2 del modelo analítico mediante Simulink

El tiempo de ejecución promedio es de 0.90 segundos.

Si la ecuación se implementa en el entorno de Matlab, el resultado es el siguiente:

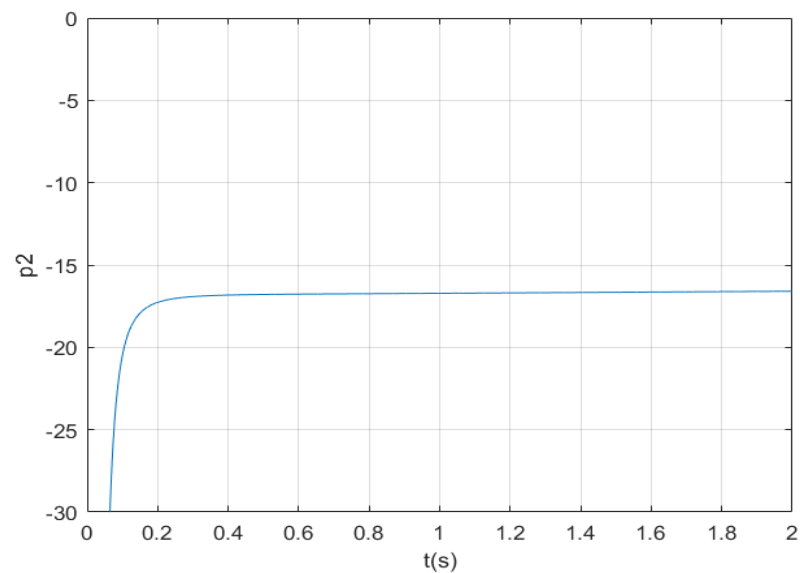


Figura 4.11: Variable p_2 del modelo analítico mediante Matlab

Con un tiempo de ejecución en torno a 0.14 segundos.

Si se comparan las tres señales obtenidas para la constante p_2 :

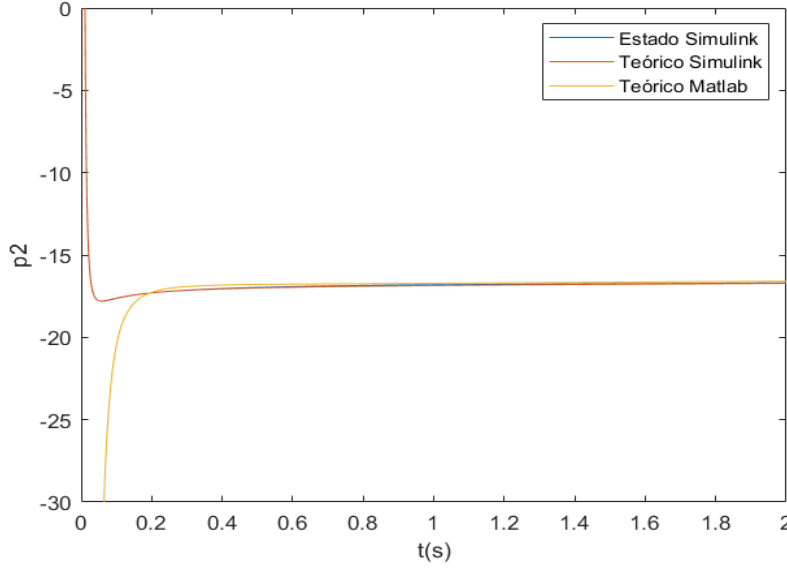


Figura 4.12: Comparativa entre métodos de obtención de la variable p_2

Se puede observar que los modelos en Simulink convergen antes que el modelo de Matlab. Sin embargo, si se observan los tiempos de ejecución, el que menor tiempo presenta es el modelo analítico en Matlab.

Una vez calculadas las variables intermedias p_2 y p_3 , se pueden obtener tanto la amplitud como la fase:

$$A = \sqrt{\frac{p_3^2}{p_1} + p_2^2} \quad (4.101)$$

$$\phi = \tan^{-1}\left(\frac{p_2\sqrt{p_1}}{p_3}\right) \quad (4.102)$$

El principal defecto que tiene este sistema de estimación, tanto en Matlab como en Simulink, es que los métodos para los diferentes parámetros se tienen que ejecutar en cadena, es decir, primero tiene que converger p_1 , posteriormente p_3 y por último p_2 , debido a que dependen del valor del modelo anterior, alargando de manera considerable el tiempo en el que se obtiene los parámetros característicos de la sinusoidal. Al igual que la precisión con la que se obtiene el parámetro influye en la estimación del siguiente.

4 4

Estimación por mínimos cuadrados de la amplitud, fase y offset

Una solución para el problema anterior, es obtener los parámetros de la amplitud y fase de forma trivial o directa. Para ello solamente es necesario conocer la frecuencia y los valores de la señal para dos instantes cualesquiera. Se parte de la ecuación:

$$u(t) = A \sin(\omega t + \phi) \quad (4.103)$$

Se consideran dos valores conocidos de la señal:

$$\begin{aligned} u(t_1) &= A \sin(\omega t_1 + \phi) \\ u(t_2) &= A \sin(\omega t_2 + \phi) \end{aligned} \quad (4.104)$$

Se desarrollan las ecuaciones, conociendo las identidades trigonométricas:

$$\begin{aligned} u(t_1) &= A \sin(\omega t_1) \cos(\phi) + A \cos(\omega t_1) \sin(\phi) \\ u(t_2) &= A \sin(\omega t_2) \cos(\phi) + A \cos(\omega t_2) \sin(\phi) \end{aligned} \quad (4.105)$$

Y se expresan de forma matricial:

$$\begin{pmatrix} u(t_1) \\ u(t_2) \end{pmatrix} = \begin{pmatrix} \sin(\omega t_1) & \cos(\omega t_1) \\ \sin(\omega t_2) & \cos(\omega t_2) \end{pmatrix} \begin{pmatrix} A \cos(\phi) \\ A \sin(\phi) \end{pmatrix} \quad (4.106)$$

Se definen unas variables intermedias para simplificar las operaciones.

$$\begin{aligned} p_4 &= A \cos(\phi) \\ p_5 &= A \sin(\phi) \end{aligned} \quad (4.107)$$

De forma que los parámetros característicos vienen definidos por:

$$\begin{aligned} A &= \sqrt{p_4^2 + p_5^2} \\ \phi &= \tan^{-1}(p_5/p_4) \end{aligned} \quad (4.108)$$

Aplicando la inversa de la matriz se pueden obtener los valores de las variables p_4 y p_5 . El principal problema de este método es que solo trabaja con datos exactos, y debido a que tanto la señal que se va a proporcionar, presenta ruido, como el valor de la frecuencia no es 100 % exacto, el resultado de la amplitud y la fase no se acercará al valor esperado. Una solución a este problema es trabajar con un rango de datos comprendido entre t_1 y t_2 y se aplica la minimización del error cuadrático en dicho rango, de forma que:

$$e(t) = u(t) - p_4 \sin(\omega t) - p_5 \cos(\omega t) \quad (4.109)$$

$$\begin{aligned} \varphi = \int_{t_1}^{t_2} e^2(\tau) d\tau &\rightarrow \frac{d\varphi}{dp_4} = 0 \\ &\rightarrow \frac{d\varphi}{dp_5} = 0 \end{aligned} \quad (4.110)$$

Si se sustituye:

$$\begin{aligned} \varphi &= \int_{t_1}^{t_2} [u(\tau) - p_4 \sin(\omega \tau) - p_5 \cos(\omega \tau)]^2(\tau) d\tau \\ \frac{d\varphi}{dp_4} = 0 &\rightarrow \int_{t_1}^{t_2} [u(\tau) \sin(\omega \tau)] d\tau = \int_{t_1}^{t_2} [p_4 \sin^2(\omega \tau)] d\tau + \int_{t_1}^{t_2} [p_5 \sin(\omega \tau) \cos(\omega \tau)] d\tau \\ \frac{d\varphi}{dp_5} = 0 &\rightarrow \int_{t_1}^{t_2} [u(\tau) \cos(\omega \tau)] d\tau = \int_{t_1}^{t_2} [p_5 \cos^2(\omega \tau)] d\tau + \int_{t_1}^{t_2} [p_4 \sin(\omega \tau) \cos(\omega \tau)] d\tau \end{aligned} \quad (4.111)$$

Y se pasa a forma matricial:

$$\begin{pmatrix} \int_{t_1}^{t_2} [u(\tau) \sin(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [u(\tau) \cos(\omega\tau)] d\tau \end{pmatrix} = \begin{pmatrix} \int_{t_1}^{t_2} [\sin^2(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\cos^2(\omega\tau)] d\tau \end{pmatrix} \begin{pmatrix} p_4 \\ p_5 \end{pmatrix}$$

$$\begin{pmatrix} p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} \int_{t_1}^{t_2} [\sin^2(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\cos^2(\omega\tau)] d\tau \end{pmatrix}^{-1} \begin{pmatrix} \int_{t_1}^{t_2} [u(\tau) \sin(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [u(\tau) \cos(\omega\tau)] d\tau \end{pmatrix}$$

Una vez conocidas las variables p_4 y p_5 , se calcula la amplitud y la fase.

Se simula el método basado en la minimización del error cuadrático con la señal descrita en la introducción, pero, al no tener ruido, su comportamiento es constante, es decir, el valor de la amplitud es 20 y la fase 1 rad para cualquier rango. Por tanto, se añade un ruido blanco del 20 % de la amplitud de la señal fijando t_1 en la primera muestra y desplazando t_2 sobre la señal:

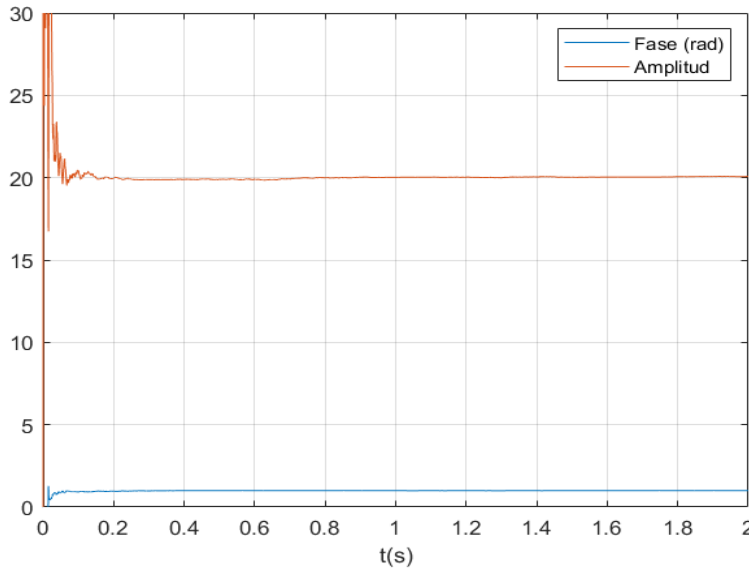


Figura 4.13: Amplitud y fase mediante el método basado en la minimización del error cuadrático con un 20 % de ruido

Como se puede observar, la convergencia se produce mucho antes que en los métodos basados en el modelo de estado y su solución, incluso con un 20 % de ruido sobre la señal. El tiempo de ejecución promedio es de 0.00254 segundos, siendo el método más eficiente.

Pero, debido a que el objetivo es obtener todos los parámetros que definen la señal original y recrearla con la mayor precisión posible, es necesario conocer el offset K , el cual se añade en el procedimiento, siendo:

$$e(t) = u(t) - p_4 \sin(\omega t) - p_5 \cos(\omega t) - K \quad (4.112)$$

$$\begin{aligned}
\varphi = \int_{t_1}^{t_2} e^2(\tau) d\tau & \rightarrow \frac{d\varphi}{dp_4} = 0 \\
& \rightarrow \frac{d\varphi}{dp_5} = 0 \\
& \rightarrow \frac{d\varphi}{dK} = 0
\end{aligned} \tag{4.113}$$

Si se sustituye:

$$\begin{aligned}
\varphi &= \int_{t_1}^{t_2} [u(\tau) - p_4 \sin(\omega\tau) - p_5 \cos(\omega\tau) - K]^2(\tau) d\tau \tag{4.114} \\
\frac{d\varphi}{dp_4} = 0 &\rightarrow \int_{t_1}^{t_2} [u(\tau) \sin(\omega\tau)] d\tau = \int_{t_1}^{t_2} [p_4 \sin^2(\omega\tau)] d\tau + \int_{t_1}^{t_2} [p_5 \sin(\omega\tau) \cos(\omega\tau)] d\tau \\
&\quad + \int_{t_1}^{t_2} [K \sin(\omega\tau)] d\tau \\
\frac{d\varphi}{dp_5} = 0 &\rightarrow \int_{t_1}^{t_2} [u(\tau) \cos(\omega\tau)] d\tau = \int_{t_1}^{t_2} [p_5 \cos^2(\omega\tau)] d\tau + \int_{t_1}^{t_2} [p_4 \sin(\omega\tau) \cos(\omega\tau)] d\tau \\
&\quad + \int_{t_1}^{t_2} [K \cos(\omega\tau)] d\tau \\
\frac{d\varphi}{dK} = 0 &\rightarrow \int_{t_1}^{t_2} u(\tau) d\tau = \int_{t_1}^{t_2} [p_4 \sin(\omega\tau)] d\tau + \int_{t_1}^{t_2} [p_5 \cos(\omega\tau)] d\tau + \int_{t_1}^{t_2} K d\tau
\end{aligned}$$

Se expresa de forma matricial:

$$\begin{pmatrix} p_4 \\ p_5 \\ K \end{pmatrix} = \begin{pmatrix} \int_{t_1}^{t_2} [\sin^2(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\sin(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [\sin(\omega\tau) \cos(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\cos^2(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\cos(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [\sin(\omega\tau)] d\tau & \int_{t_1}^{t_2} [\cos(\omega\tau)] d\tau & (t_2 - t_1) \end{pmatrix}^{-1} \begin{pmatrix} \int_{t_1}^{t_2} [u(\tau) \sin(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} [u(\tau) \cos(\omega\tau)] d\tau \\ \int_{t_1}^{t_2} u(\tau) d\tau \end{pmatrix} \tag{4.115}$$

Una vez conocidas las variables p_4 y p_5 se sustituyen en las ecuaciones:

$$\begin{aligned}
A &= \sqrt{p_4^2 + p_5^2} \\
\phi &= \tan^{-1}(p_5/p_4) \\
K &= K
\end{aligned} \tag{4.116}$$

Se realiza la misma prueba que en el caso anterior, con un 20% de la amplitud como ruido blanco, fijando t_1 en la primera muestra y desplazando t_2 .

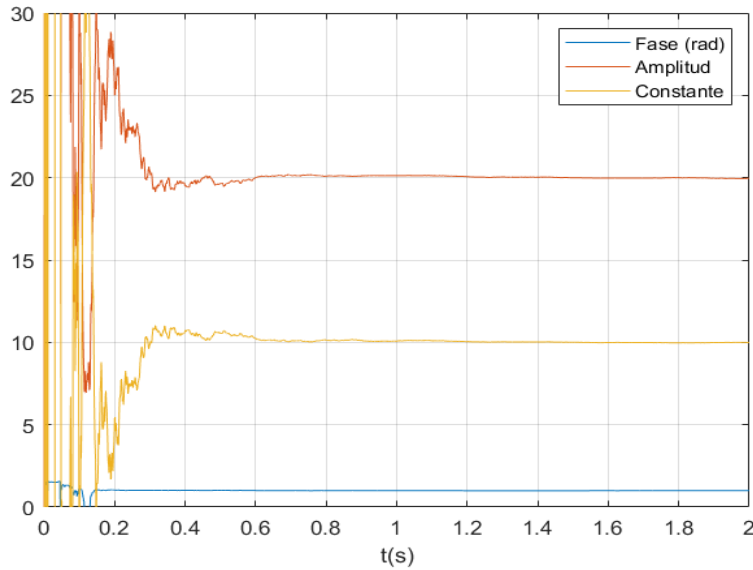


Figura 4.14: Amplitud, fase y constante mediante el método basado en la minimización del error cuadrático con un 20 % de ruido

Como se puede observar, cuanto mayor es el rango de datos, más precisa es la estimación. El tiempo de ejecución promedio es de 0.016 segundos, siendo definitivamente el método más rápido y con mayor precisión para calcular la amplitud, fase y el offset. La única condición de este método es que se tiene que conocer la frecuencia de la señal con la mayor precisión posible.

Como conclusión, el método basado en la minimización del error cuadrático sustituye a los métodos basados en el modelo de estado y su solución analítica para la estimación de los parámetros de la amplitud, fase y offset, gracias a su gran precisión y eficiencia. El inconveniente de este método es la necesidad de conocer la frecuencia, pero mediante el uso del modelo de estado para p_1 se estima su valor.

El algoritmo para obtener la frecuencia se puede implementar tanto en Simulink como en Matlab, ofreciendo cada entorno ventajas e inconvenientes. El entorno de Simulink posee la ventaja de ser una herramienta intuitiva y sencilla, gracias a su diseño mediante bloques; mientras que el entorno de Matlab, requiere encontrar una solución a las ecuaciones de estado, pero se gana una mayor eficiencia computacional. Además, el algoritmo de Matlab posee una mayor portabilidad a otros lenguajes como pueden ser Arduino, Labview, Python, etc., ya que la única complejidad de su código es el uso de integrales, en contra posición a los bloques de Simulink.

CAPÍTULO 5

IMPLEMENTACIÓN

5.1

Introducción

Una vez propuestas las diferentes formas de obtener los parámetros característicos de la señal y comparar tanto su comportamiento como el tiempo de ejecución bajo un número de muestras fijo, se pasa a la creación del programa que recibiendo como argumento de entrada la señal, sea capaz de recrearla con la mayor eficiencia posible.

Los métodos utilizados para construir el programa son: el método basado en la solución de las ecuaciones de estado para la frecuencia; y el método basado en la minimización del error cuadrático para la amplitud, fase y offset en el entorno de Matlab.

El funcionamiento del programa se basa en ejecutar el algoritmo de la solución analítica para la frecuencia sobre las muestras obtenidas, hasta que su valor converja. Una vez obtenida la frecuencia, se pasa al método basado en la minimización del error cuadrático para estimar la amplitud, fase y el offset. En las ecuaciones que definen el comportamiento de la frecuencia, se puede observar como el programa integra desde t_0 hasta t , siendo el instante t , donde se encuentra la muestra. Este proceso se puede realizar de forma recursiva, disminuyendo considerablemente el nivel computacional. Para ello, se usa la siguiente propiedad de las integrales:

$$\int_{t_0}^t X(t)dt \longleftrightarrow \int_{t_0}^{t_1} X(t)dt + \int_{t_1}^t X(t)dt \quad (5.1)$$

Gracias a esta propiedad, se reescribe la ecuación (4.44), siendo solamente necesario conocer los valores de las integrales en la ejecución anterior y el valor de la muestra en el instante t ; salvo en la primera ejecución, pues no hay ningún valor previo. Siendo para la primera ejecución:

$$p_1 = \frac{t^3 u(t_1) - 30 \int_{t_0}^{t_1} \tau^2 U(\tau) d\tau + 24t_1 \int_{t_0}^{t_1} \tau U(\tau) d\tau - 3t_1^2 \int_{t_0}^{t_1} U(\tau) d\tau}{\frac{5}{2} \int_{t_0}^{t_1} \tau^4 U(\tau) d\tau - 4t_1 \int_{t_0}^{t_1} \tau^3 U(\tau) d\tau + \frac{3}{2} t_1^2 \int_{t_0}^{t_1} \tau^2 U(\tau) d\tau} \quad (5.2)$$

Mientras que, para ejecuciones posteriores, hay que tener en cuenta que la ecuación depende de la variable t , siendo necesario cancelar el valor del instante anterior por el

instante actual, de forma que la ecuación (4.44) queda de la siguiente forma:

$$p_1 = \frac{t_2^3 u(t_2) + [A_1 - 30 \int_{t_1}^{t_2} \tau^2 U(\tau) d\tau] + [A_2 \frac{t_2}{t_1} + 24 \int_{t_1}^{t_2} \tau U(\tau) d\tau] + [A_4 + \frac{5}{2} \int_{t_1}^{t_2} \tau^4 U(\tau) d\tau] + [A_5 \frac{t_2}{t_1} - 4 \int_{t_1}^{t_2} \tau^3 U(\tau) d\tau] + [A_3 \frac{t_2^2}{t_1^2} - 3 \int_{t_1}^{t_2} U(\tau) d\tau]}{[A_6 \frac{t_2^2}{t_1^2} + \frac{3}{2} \int_{t_1}^{t_2} \tau^2 U(\tau) d\tau]} \quad (5.3)$$

donde las constantes A_X guardan los valores de las integrales anteriores.

Una vez determinada la frecuencia de la señal, se utiliza para encontrar los parámetros de la amplitud, fase y offset. Hay que tener en cuenta que, dichos parámetros van en función de la frecuencia, es decir, cuanta más precisión se consiga al determinar la frecuencia, mayor precisión tendrán estos parámetros. Como ya se vio anteriormente, la principal ventaja es la sencillez a la hora de obtenerla y el tiempo de ejecución, mientras que la principal desventaja es que los datos a tratar por las integrales no se pueden realizar de forma paralela, ya que es necesario conocer el valor de la frecuencia, teniendo que volver a tratar los datos obtenidos desde cero.

En la práctica, la señal presenta distorsiones y frecuencias no deseadas, introducidas por el sensor y el ambiente. Una de las principales mejoras es disminuir el efecto del ruido y aislar los datos que se van a tratar de la señal, aplicando sobre ella una serie de filtros. Otra mejora es determinar en qué instante se ha obtenido la frecuencia con un cierto nivel de precisión y finalizar la ejecución del estimador. Todas estas mejoras serán tratadas en detalle dentro de este capítulo.

Para la realización de las pruebas, se trabaja con una señal normalizada, donde la amplitud de la señal es la unidad, una frecuencia de 1 Hz y tanto la fase como el offset son nulos:

$$\hat{u} = \frac{1}{A} [A \sin(2\pi t)] \quad (5.4)$$

con una frecuencia de muestreo de 1 kHz ($T_s = 0.001$ seg.) durante 2 segundos.

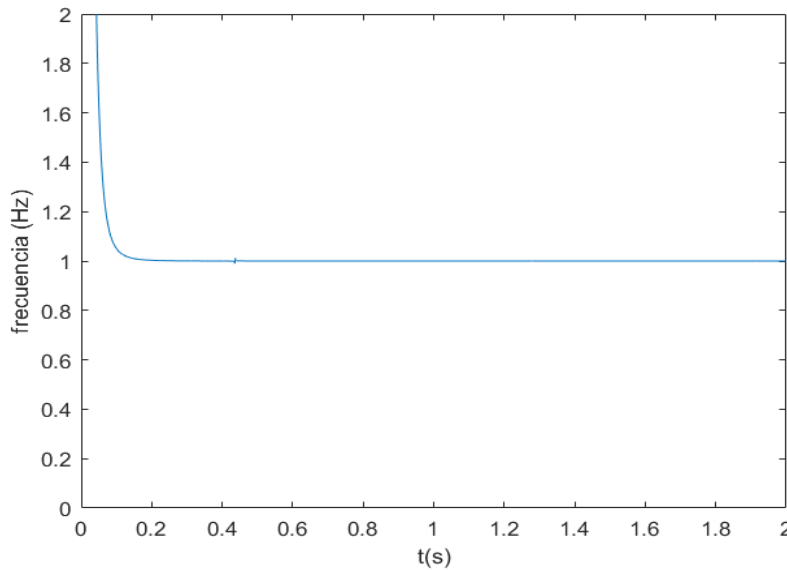


Figura 5.1: Frecuencia normalizada sin centinela

Como se puede observar, la frecuencia entra en la zona de convergencia a partir del instante 0.2 segundos y aparecen unas leves discontinuidades cuando el denominador cruza por cero.

5 2

Filtrado

El objetivo de los filtros es eliminar los componentes no deseados y aislar los datos a tratar de la señal, aumentando de forma considerable la precisión a la hora de estimar la señal sinusoidal. Para ello, se añade a la señal normalizada un ruido blanco de media cero del 10 % de la amplitud.

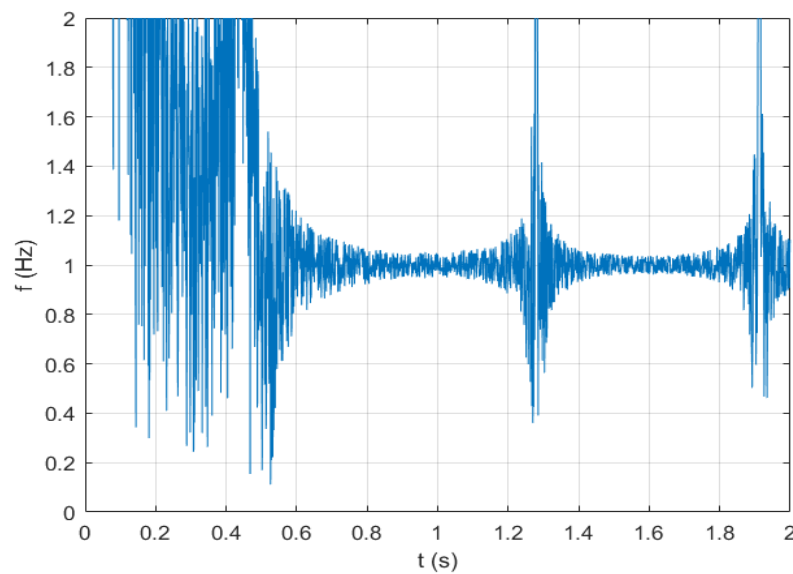


Figura 5.2: Frecuencia normalizada con ruido

5 2 1 Filtro lineal de Butterworth

En primer lugar, se implementa el filtro más básico y común, el filtro paso bajo de Butterworth de segundo orden. El objetivo de este filtro es aislar las frecuencias deseadas, además de disminuir la influencia del ruido.

$$F(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5.5)$$

El principal problema de este filtro es su configuración, siendo necesario conocer el rango de frecuencias en el que se va a trabajar para poder definir la frecuencia de corte. En este caso, el rango de frecuencias en el que se va a mover la estimación, se encuentra entre los 4 y 8 Hz, por lo que se diseña un filtro paso de bajo con una frecuencia de corte ω_n igual a 10 Hz, es decir, atenuarán aquellas frecuencias cuyo valor sea mayor a la frecuencia de corte y una ζ igual a 0.707.

Se aplica el filtro paso bajo de forma independiente al numerador y denominador, garantizando la invarianza del filtro sobre el parámetro p_1 , siempre que la configuración

sea la misma.

$$p_1 = \frac{n_{1f}(t)}{d_{1f}(t)} = \frac{F(s)n_1(t)}{F(s)d_1(t)} \quad (5.6)$$

Gracias a esto, se consigue asegurar que el filtro no modifica el valor de p_1 . Se implementa dando como resultado:

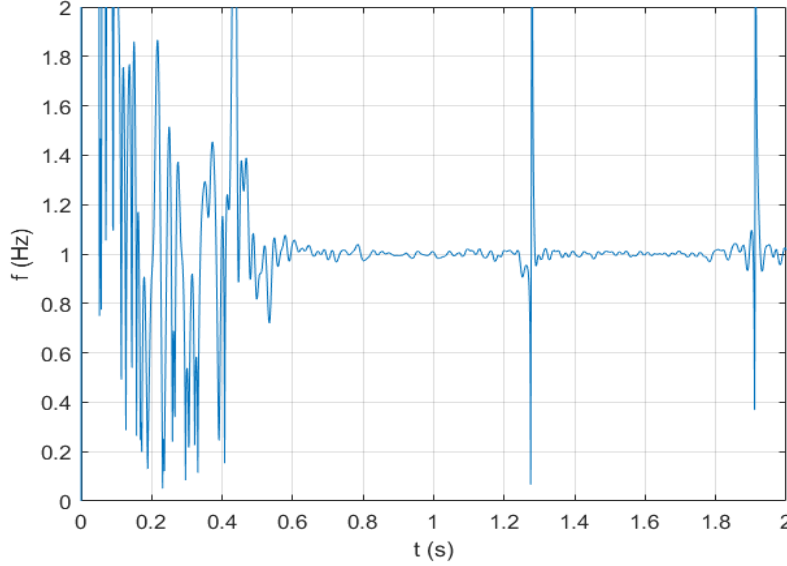


Figura 5.3: Frecuencia con filtro paso bajo Butterworth

Como se puede observar, se disminuye de forma considerable el ruido sobre la señal, pero siguen apareciendo distorsiones que dificultan la obtención de la frecuencia. Además, aumentan las discontinuidades que se producen cuando el denominador cruza por cero.

5.2.2 Filtro no lineal mediante integrales de valores absolutos

Una solución es añadir de forma adicional un filtro no lineal basado en las integrales de los valores absolutos del numerador y el denominador durante todo el rango de datos:

$$f(t) = \frac{\int_0^t |n_{1f}(\tau)| d\tau}{\int_0^t |d_{1f}(\tau)| d\tau} \quad (5.7)$$

Sus principales ventajas son su sencillez y su falta de configuración. Además de la posibilidad de implementarlo de forma recursiva:

$$f(t_2) = \frac{I n_0^{t_1} + \int_{t_1}^{t_2} |n_{1f}(\tau)| d\tau}{I d_0^{t_1} + \int_{t_1}^{t_2} |d_{1f}(\tau)| d\tau} \quad (5.8)$$

donde $I_0^{t_1}$ almacena el valor de la integral en la ejecución anterior.

Se implementa el filtro no lineal después del filtro paso bajo, dando como resultado:

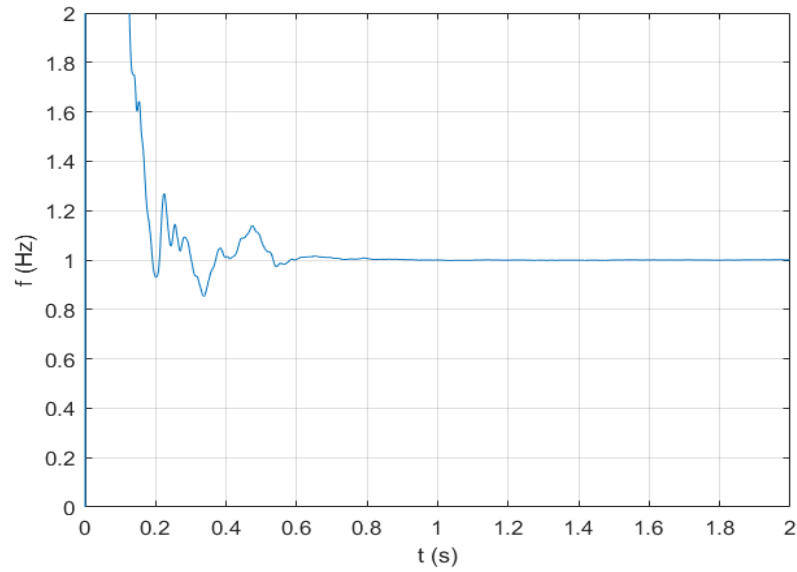


Figura 5.4: Frecuencia con filtro lineal y no lineal

Como se puede observar, el filtro es capaz de eliminar las discontinuidades provocadas cuando el denominador cruza por cero, y de suavizar la influencia del ruido de forma progresiva. Además, dicho filtro ayuda a converger la señal sobre su valor, obteniendo la frecuencia con un mayor grado de precisión.

5 3

Selección de la frecuencia

Uno de los principales problemas a los que nos enfrentamos es conocer el tiempo necesario o las muestras necesarias para determinar la frecuencia, ya que dicho valor depende de varios parámetros como pueden ser la propia frecuencia de la señal, el tiempo de muestreo o el nivel de ruido que presenta la señal. Si usa un tiempo de estimación fijo y genérico para cualquier tipo de señal, que esté diseñado para el peor caso, el programa será muy ineficiente para señales que se pueden determinar en un tiempo mucho más corto. Para solventar este problema, se implementa un estimador que seguirá a la señal en todo momento y una vez que se cumpla una condición, que determine que la frecuencia es válida, finalizará el programa, siendo capaz de adaptarse a cualquier tipo de datos.

5 3 1 Centinela basado en error absoluto

En primer lugar, se implementará un estimador sencillo, el cual finalizará el programa cuando la frecuencia estimada supere un umbral, conocido como tolerancia (definido por la diferencia máxima entre la frecuencia estimada y la frecuencia esperada):

$$t_{conv} = |\hat{\omega}_k - \omega| < tol, \quad k = 1, 2, \dots, N \quad (5.9)$$

donde tol es el valor que representa la tolerancia.

Para determinar la tolerancia, se estudia el comportamiento de la señal normalizada ante diferentes situaciones de ruido, para ello se usa el concepto de relación señal-ruido (SNR), que viene definido por:

$$SNR = \frac{|A^2|}{|\sigma^2|} \quad (5.10)$$

donde A corresponde con la amplitud de la señal normalizada y σ con la varianza del ruido. El SNR se expresa en decibelio (dB) y la varianza se relaciona con la amplitud del ruido blanco añadido, quedando la siguiente expresión:

$$SNR = 10\log_{10} \frac{A_1^2}{A_2^2} = 10\log_{10} \frac{1}{A_1^2} \quad (5.11)$$

donde A_2 corresponde con la amplitud del ruido.

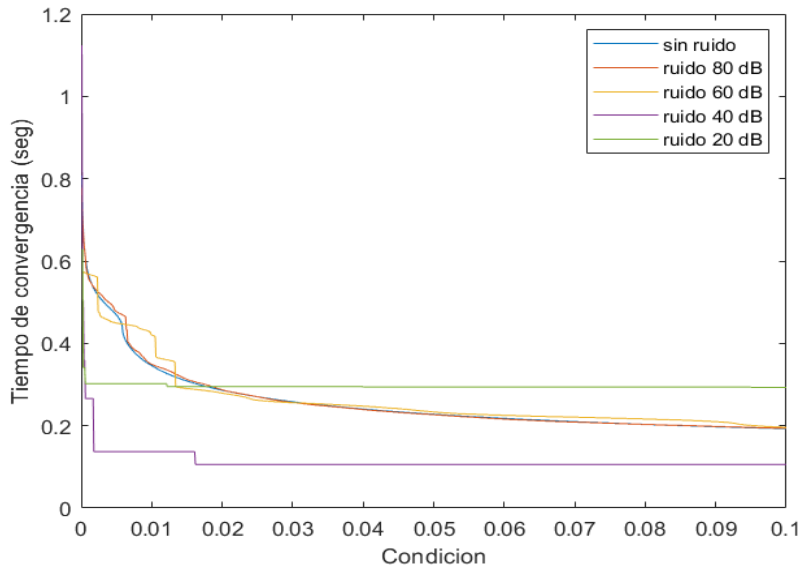


Figura 5.5: Condición vs tiempo de convergencia

Si se observa la figura 5.4, la influencia del ruido hace oscilar la señal sobrepasando el umbral definido sin llegar a estabilizarse, esto se refleja en el estimador devolviendo la frecuencia cuando su valor no ha convergido. Su verdadero funcionamiento se produce cuando la señal se encuentra libre de ruido, devolviendo el valor de la frecuencia cuando se estabiliza sobre el umbral marcado. Cuanto más restrictiva es la condición, más aumenta el tiempo de convergencia, y por tanto, la precisión de la frecuencia.

La principal desventaja de este método es que se desconoce la frecuencia que se va a estimar, es decir, se desconoce la variable ω de la configuración, además de ser un estimador no válido en entornos ruidosos.

5.3.2 Centinela basado en error relativo

Para solventar el problema del centinela basado en el error absoluto, se diseña un centinela que trabaje sobre los datos obtenidos, en lugar de una frecuencia que se desconoce, siendo un modelo válido para cualquier estimación. El funcionamiento del

centinela es igual al anterior, donde se escogerá un rango de datos que deben cumplir una condición de precisión. Una vez cumplido, se detendrá y escogerá el último valor del rango como válido. La condición a superar es:

$$\frac{\sigma_i}{|E[\omega_i]|} \leq \Delta_i \quad (5.12)$$

donde σ_i corresponde a la desviación estándar, $E[\omega_i]$ corresponde a la media de los valores dentro del rango del centinela y Δ_i corresponde al valor a cumplir. Las ecuaciones de la desviación estándar y la media son:

$$\sigma_i = \sqrt{\frac{\sum_{k=1}^N (\omega_k - \bar{\omega}_i)^2}{N}} \quad (5.13)$$

$$E[\omega_i] = \frac{1}{N} \sum_{k=1}^N \omega_k \quad (5.14)$$

El caso ideal se produce cuando la desviación estándar es nula, es decir, los valores de la frecuencia son invariantes dentro del rango de datos, pero en la práctica esto es imposible, haciendo necesario ajustar la condición Δ_i . En aplicaciones que requieren una alta precisión, el valor de Δ_i debe ser pequeño, mientras que para aplicaciones que requieren una rápida estimación, el valor de Δ_i debe ser grande. En nuestro caso debemos encontrar un equilibrio entre rapidez y precisión.

Para implementar el centinela, primero se debe realizar la ejecución del algoritmo sin este, con el objetivo de ver el comportamiento de la señal. A partir de ese momento, se tiene que determinar el tiempo de convergencia donde la señal ha alcanzado el valor de la frecuencia con la precisión deseada, y se diseñará el centinela en torno a ese punto, realizando un barrido para diferentes valores del rango y condiciones.

Se realiza un barrido sobre la señal normalizada sin ruido para rangos de datos entre 10 y 1000, y para cada rango se realiza un barrido sobre la condición entre 0.1 y 0.0001.

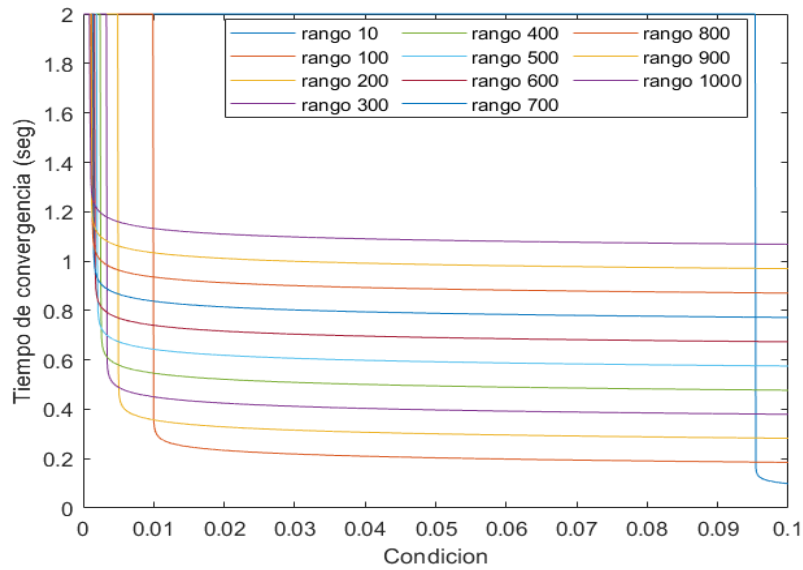


Figura 5.6: Condición vs tiempo de convergencia entre 10 y 1000

Como se puede observar, el parámetro de la condición va ligado al rango de datos, pues para un rango de datos tiene asociado una condición límite. Cuanto mayor es el rango de datos, menor puede ser su condición, y a su vez, mayor su precisión. Por otro lado, el rango y su condición van ligados al tiempo de convergencia, ya que, al aumentar el rango, el programa tiene que trabajar con un mayor número de muestras, aumentando de forma considerable su tiempo de convergencia. En definitiva, cuanto menor sea la condición y mayor el rango de datos, mayor precisión se obtiene en la salida del centinela, pero, a su vez, mayor es el tiempo de convergencia.

Se representa una tabla con la precisión obtenida de los rangos anteriores, con sus condiciones límites asociadas y el tiempo de convergencia, con el objetivo de ver su comportamiento:

Rango	Tiempo de convergencia	Condición	Precisión
10	0.130	0.0958	91.5270 %
100	0.307	0.0103	99.7552 %
200	0.422	0.0054	99.9049 %
300	0.533	0.0037	99.9711 %
400	0.638	0.0029	99.9933 %
500	0.734	0.0024	99.9974 %
600	0.834	0.0022	99.9987 %
700	0.933	0.0020	99.9994 %
800	1.035	0.0018	99.9997 %
900	1.140	0.0016	99.9999 %
1000	1.240	0.0015	99.9999 %

Tabla 5.1: Valores sobre las condiciones límites del centinela

Como se puede observar, la relación de la condición y la precisión con respecto al rango de datos es exponencial.

A partir de aquí, se tiene que decidir en qué punto se quiere obtener la frecuencia. Para esta aplicación, el objetivo es identificar la frecuencia en menos de un periodo de la señal, es decir, un segundo, y asegurar un mínimo de precisión. Si se retorna a la figura 5.1, se puede observar como a partir del instante 0.2 segundos la frecuencia converge con una precisión del 95 %. En este caso, se establece el tiempo de convergencia en 0.3 segundos, debido a que su comportamiento exponencial hace rentable aumentar la precisión, aunque el tiempo de convergencia aumente. Se realiza un nuevo barrido para encontrar el punto que verifique el tiempo de convergencia en el menor valor de la condición.

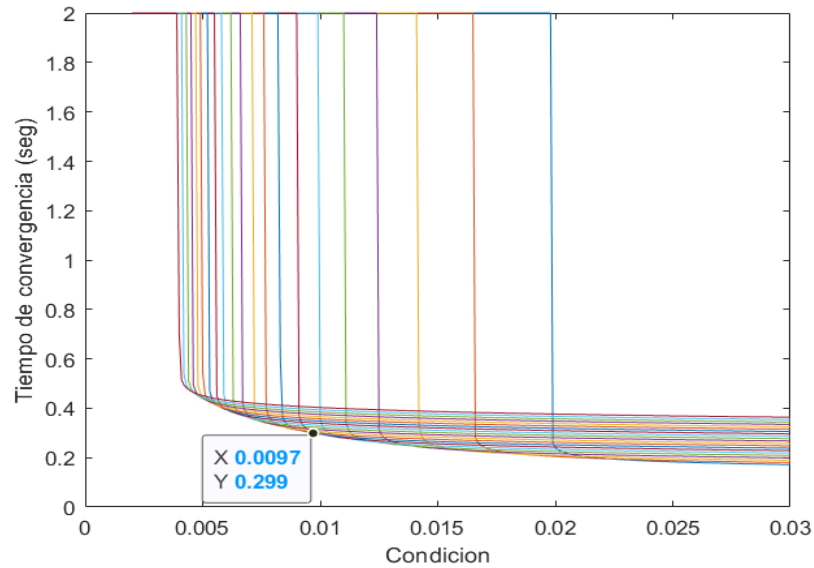


Figura 5.7: Condición vs tiempo de convergencia entre 100 y 300

El punto con menor condición que alcanza el instante 0.3 segundos, se encuentra en el rango de 120 muestras y una condición de 0.0097, obteniendo la frecuencia con una precisión mínima de 99.7329 %.

Uno de los principales problemas de la implementación del centinela se produce en entornos ruidosos, siendo necesario el estudio de su comportamiento en presencia de diferentes niveles de ruido.

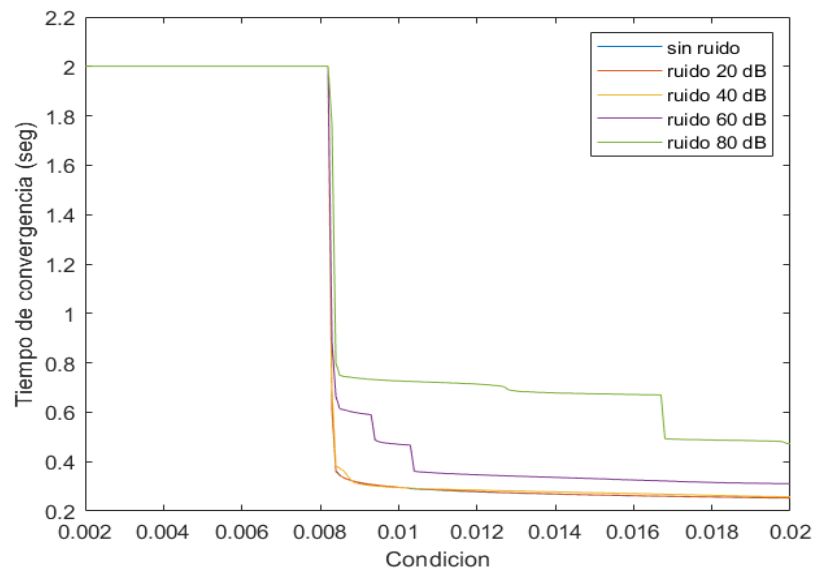


Figura 5.8: Condición vs tiempo de convergencia en presencia de ruido

Como se puede observar, ante la presencia de ruido en la señal, aumenta considerablemente el tiempo de convergencia manteniendo su condición, es decir, devuelve la frecuencia con la misma precisión a costa de aumentar el tiempo. El principal problema de este efecto, viene cuando el ruido es muy elevado, de forma que aumente

considerablemente el tiempo de convergencia. Para solventar este problema, se añade un tiempo límite que detendrá la ejecución del programa cuando se supere dicho límite, devolviendo el valor de la frecuencia, aunque no haya alcanzado la condición del rango. Para este caso se establece en 1 segundo, asegurando obtener la frecuencia en un ciclo de la señal.

De forma adicional, se establece un margen de tiempo donde el centinela omitirá los primeros valores de la frecuencia, debido a que se encuentran muy dispersos y lejos del objetivo, ahorrando de este modo operaciones innecesarias. Dicho valor irá ligado al número de muestras. En este caso tarda 200 muestras en converger, por lo que se omitirán las primeras 100 muestras.

La principal ventaja de este método es su validez para cualquier aplicación y señal de entrada, siendo capaz de adaptarse con niveles de ruido muy altos, a costa de aumentar el tiempo de convergencia.

ANÁLISIS DE ROBUSTEZ

6 1

Introducción

Una vez definido el programa, se procede a realizar un análisis de robustez ante diferentes situaciones, donde los escenarios a tener en cuenta son la sensibilidad ante diferentes valores de la frecuencia de la señal ω , diferentes valores de la frecuencia de muestreo f_s , diferentes valores del índice señal-ruido (SNR) y ante la presencia de altas frecuencias.

Para la realización de las pruebas, se va a trabajar con la señal normalizada configurada en el capítulo 5, es decir, con muestras tomadas durante 2 segundos con una frecuencia de muestreo de 1 kHz, una frecuencia de corte del filtro paso bajo de 10 Hz y la configuración del centinela con un rango de 120 muestras y una condición de 0.0097. Para el análisis de robustez se omite la configuración tanto del tiempo límite como del tiempo de omisión de muestras, para poder ver el comportamiento completo de la señal.

6 2

Efecto de la frecuencia de la señal

El comportamiento del algoritmo ante diferentes valores de la frecuencia de la señal es un aspecto muy importante, debido a la influencia del filtro paso bajo para atenuar aquellas frecuencias que se encuentren fuera de su rango.

Como se puede observar en la figura 6.1, el tiempo de convergencia disminuye de forma exponencial con respecto al aumento de la frecuencia de la señal dentro del rango del filtro. Pero como era de esperar, a partir de la frecuencia 10 Hz, el comportamiento del centinela se vuelve inestable, ya que, el filtro paso bajo actúa sobre la señal filtrando las frecuencias a estimar.

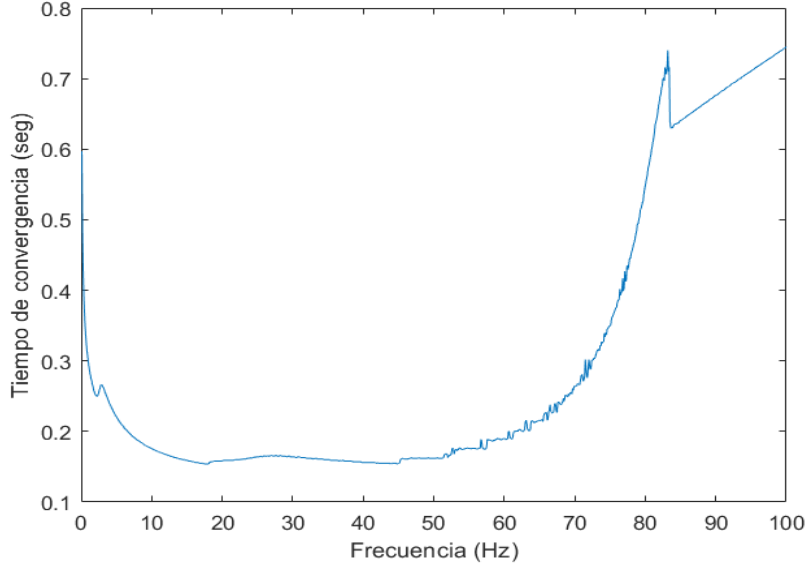


Figura 6.1: Frecuencia vs tiempo de convergencia

En el caso de utilizar dicho método para otra aplicación donde los datos se encuentren en otro rango diferente, se debe modificar la frecuencia de corte del filtro paso bajo, dejando pasar los datos de interés.

6.3

Efecto de la frecuencia de muestreo

El periodo o frecuencia de muestreo es un parámetro que afecta de forma directa a cualquier estimador diseñado en el dominio discreto. Rife y Boorstyn en [18] obtuvieron los límites para la estimación de los parámetros de la sinusoidal, en particular, para la obtención de la frecuencia, desconociéndose la amplitud y la fase, estando la varianza limitada por:

$$\text{var}(\hat{\omega}) \geq \frac{12}{SNR^2 T_s^2 N(N^2 - 1)} \quad (6.1)$$

donde $\hat{\omega}$ es la frecuencia estimada, SNR es el índice señal-ruido, T_s es la frecuencia de muestreo y N es el número de muestras tomadas. Como se puede observar, cuanto mayor es la frecuencia de muestreo, mayor es la varianza de la frecuencia estimada. El tiempo de muestreo, T_s , el índice SNR y el número de muestras N , producen el efecto contrario, ya que, cuanto mayor es su valor, menor es la varianza de la frecuencia estimada.

El método propuesto en este trabajo, está basado en el dominio del tiempo donde, al igual que en el dominio discreto, se estima que la precisión de la frecuencia estimada empeore con respecto al aumento de la frecuencia de muestreo. Para ello se estudia el efecto de la frecuencia de muestreo sobre la señal normalizada:

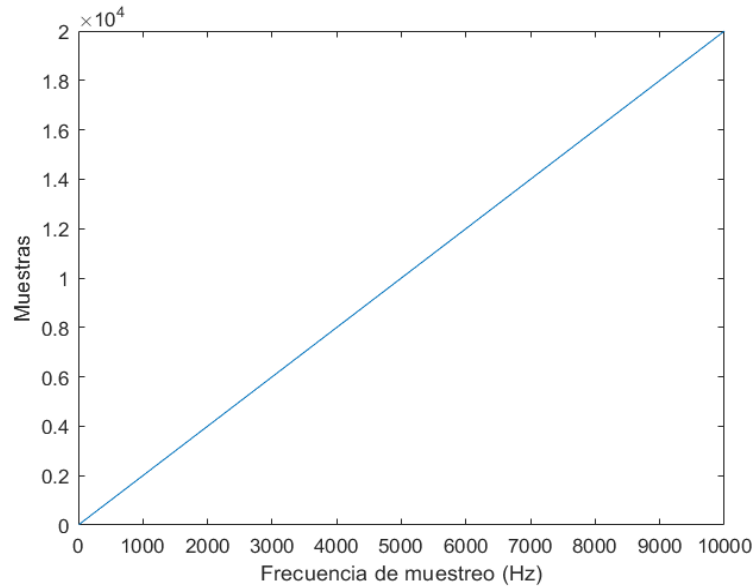


Figura 6.2: Muestras vs frecuencia de muestreo

Como se puede observar, el número de muestras necesarias para obtener la frecuencia con la misma precisión aumenta de forma lineal con respecto al aumento de la frecuencia de muestreo. Si el número de muestras se mantiene fijo, significa que la precisión disminuye, o lo que es lo mismo, la varianza de la frecuencia estimada aumenta.

Al variar la frecuencia de muestreo, este influye sobre la configuración del centinela, existiendo otras configuraciones mejor optimizadas o incluso no llegando a cumplir la condición, siendo necesario ajustar el rango y la condición con respecto a la frecuencia de muestreo de cada aplicación.

6 4

Efecto del ratio SNR

La señal con la que se trabajará presentará un nivel de ruido, por lo que es importante ver el comportamiento del ruido sobre la señal normalizada, donde la relación señal-ruido viene definida por:

$$SNR = \frac{|A^2|}{|\sigma_v^2|} \quad (6.2)$$

donde A^2 corresponde con la amplitud de la señal normalizada y σ_v^2 corresponde a la varianza del ruido.

Se realiza un barrido del ruido blanco desde 0 % al 10 % de la amplitud de la señal normalizada como ruido.

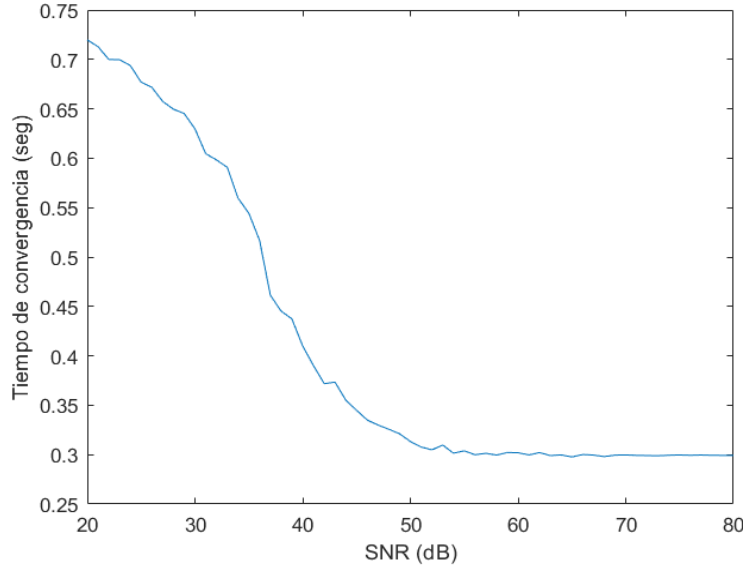


Figura 6.3: Tiempo de convergencia vs SNR

Como se puede observar, el tiempo de convergencia aumenta de forma exponencial si aumenta la amplitud del ruido. Además, como se vio anteriormente, el ruido no afecta ni a la condición ni el rango del centinela, solo al tiempo de convergencia.

6 5

Efecto ante señales con alta frecuencia

Se estudia la influencia de la señal en presencia de señales con altas frecuencias, que se sumarán a la señal normalizada de forma:

$$\hat{u}(t) = \sin(2\pi t) + A_2 \sin(2\pi f_2 t)$$

donde la amplitud A_2 y la frecuencia f_2 variarán. Dicha señal se identificará como ruido, donde se puede relacionar con la señal normalizada por medio del SNR:

$$SNR = 10 \log_{10} \frac{A_1^2}{A_2^2} = 10 \log_{10} \frac{1}{A_2^2} \quad (dB)$$

Se variará la amplitud A_2 mediante el SNR y el valor de la frecuencia f_2 :

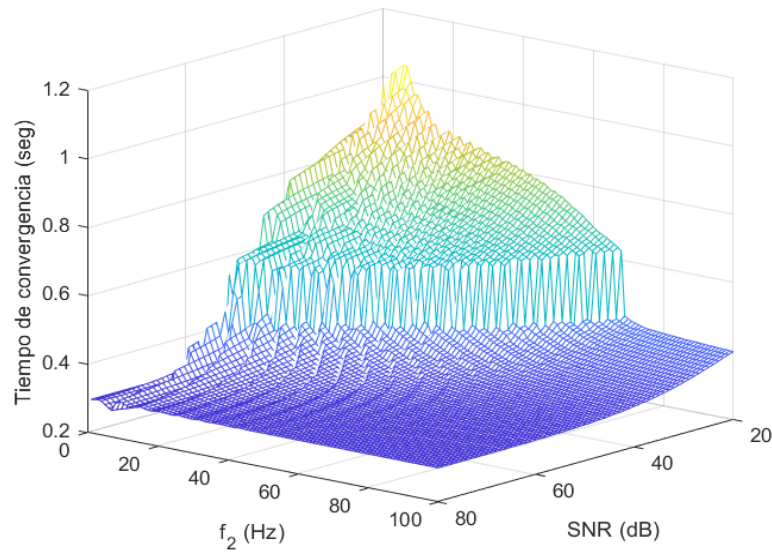


Figura 6.4: Tiempo de convergencia vs SNR

Se puede observar el efecto que provoca el filtro paso bajo, ya que, disminuye de forma considerable la influencia del ruido para frecuencias muy altas. Sin embargo, cuando la señal presenta una frecuencia dentro del rango del filtro con una alta amplitud, esta distorsiona la señal, aumentando el tiempo de convergencia.

Se estudia el caso particular de combinar la señal normalizada junto a una sinusoidal de igual amplitud y frecuencia 4 Hz.

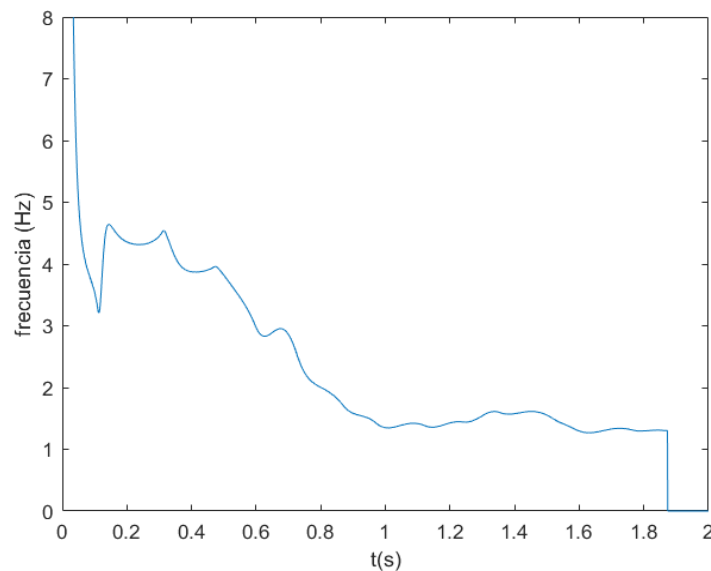


Figura 6.5: Frecuencia de la señal normalizada junto con una señal de alta amplitud

Como se puede observar el tiempo de convergencia aumenta de forma considerable. Además, el valor de salida no corresponde con la frecuencia normalizada ni cumple con la precisión del centinela. Aquí se encuentra uno de los principales problemas de

este método, ya que, cuando coexisten varias frecuencias dentro del rango del filtro paso bajo con una amplitud similar, el algoritmo no es capaz de realizar la estimación correctamente.

COMPARACIÓN CON OTROS MÉTODOS

En esta sección, se va a comparar el método algebraico propuesto con otros métodos publicados que resuelven la identificación de los parámetros característicos de una señal sinusoidal, con el fin de ver su rendimiento y robustez frente a dichos métodos.

En el artículo de Trapero, Sira-Ramírez y Feliu [23] comparan el método con la técnica de estimación de los parámetros de la sinusoidal presentada por Ziarani y Konrad [28]. Este método es un estimador adaptativo no lineal, robusto e inmune a niveles de ruido muy altos, basado en el dominio del tiempo, siendo un método válido para operaciones en tiempo real. Su mayor inconveniente es la dificultad a la hora de configurar sus parámetros de diseño y definir las condiciones iniciales. Como se puede observar en dicho artículo, cuando adaptan el estimador adaptativo bajo condiciones ruidosas y sesgadas, la frecuencia no termina de converger en el valor real debido a la presencia de ruido constante, afectando a la estimación de los demás parámetros. En este artículo también se puede encontrar la comparación con el método desarrollado por Hou en [11], con un estimador globalmente convergente basado en el dominio del tiempo. Al igual que anteriormente, el principal problema es la configuración de los parámetros de diseño, donde algunos autores han tenido problemas, ya que, no siguen unas reglas. En este caso, la señal consigue converger adecuadamente, demorando hasta 25 segundos en obtener el valor de la frecuencia con precisión, mientras que, la amplitud y la fase tarda 0.4 segundos.

En definitiva, el método algebraico propuesto en este texto tiene un rendimiento superior comparado con los métodos vistos anteriormente, siendo capaz de conseguir una rápida estimación, incluso bajo condiciones de ruido y perturbaciones constantes. Además, tiene la ventaja de que no necesita ningún parámetro de diseño ni depende de las condiciones iniciales, pues la configuración de dichos parámetros suele ser el principal inconveniente de estos métodos, siendo difícil optimizarlos a los datos de entrada.

De forma paralela, se compara el método propuesto con la transformada rápida de Fourier (FFT), siendo el método más extendido y usado en el ámbito de la ingeniería para el tratamiento de señales. Se utilizan los datos obtenidos con señal normalizada, añadiendo un alto nivel de ruido blanco de media cero y una frecuencia de muestreo de 1 kHz.

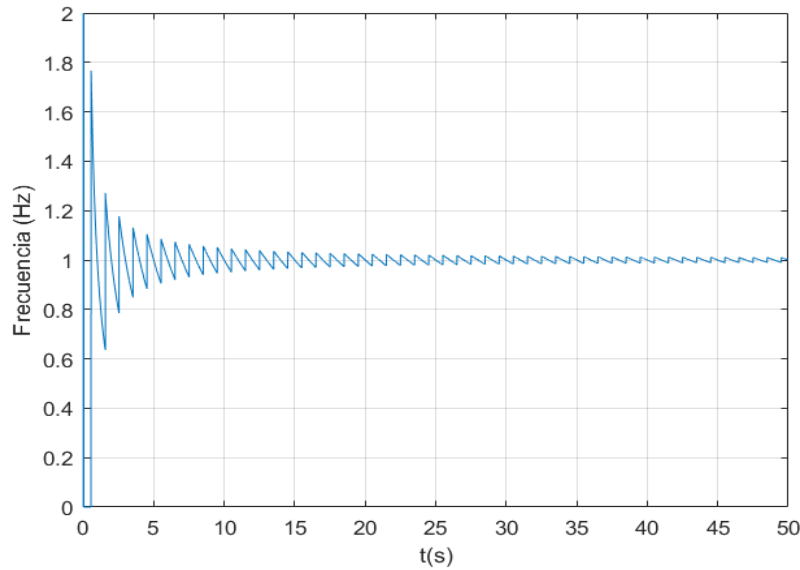


Figura 7.1: Transformada rápida de Fourier (FFT) para la señal normalizada

La principal ventaja de los estimadores espectrales no paramétricos es que son aplicaciones muy válidas para entornos muy ruidosos, pero estos estimadores tienen la desventaja de tener una alta carga numérica, es decir, necesitan un número de muestras elevado para obtener el valor de la frecuencia con precisión. Además, la FFT tiene una limitación, ya que para obtener el valor exacto de la frecuencia necesita que el número de muestras en la serie sean potencia de 2, es decir, tienen que existir 2^n muestras.

Se estudia el efecto que produce la frecuencia de muestreo y el tiempo ejecución sobre la FFT:

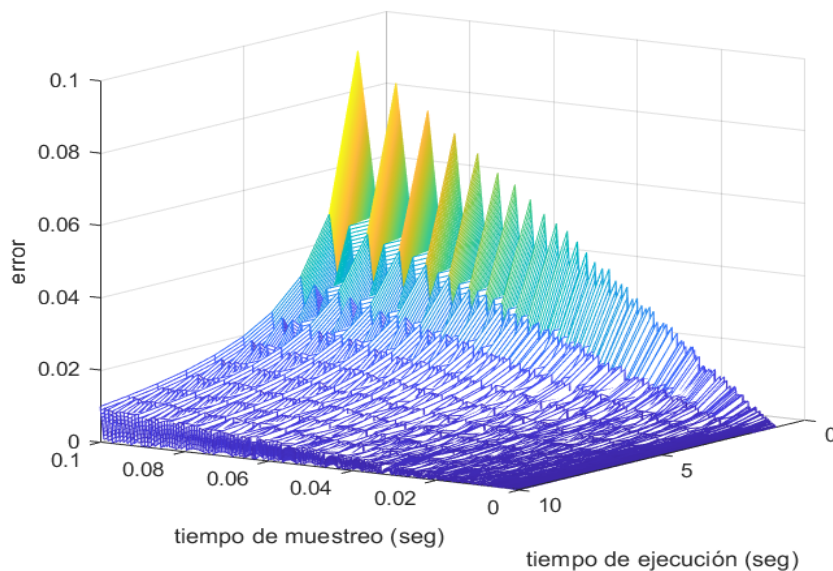


Figura 7.2: Error de la frecuencia en función del tiempo de simulación y el periodo de muestreo.

Como se puede observar cuanto mayor es el tiempo de simulación y menor el periodo

de muestreo, la frecuencia se obtiene con mayor precisión. Esto se debe al aumento del número de muestras con el que se trabaja. Sabiendo que la frecuencia de muestreo viene definida por el sensor, da lugar a la necesidad de obtener un largo tiempo de tomas de datos, viéndose reforzado el principal problema que presentan estos estimadores y su incapacidad para aplicarlos en aplicaciones en tiempo real como la aquí planteada.

RESULTADOS EXPERIMENTALES

En esta sección, se describirá el proceso realizado por Claudia F. Yasar en [26] para obtener las señales de temblor del Parkinson y se condicionará el método propuesto para la resolución de los parámetros de dichas señales.

8.1

Obtención de las señales

Para obtener las señales que definen el comportamiento del Parkinson, Claudia F. Yasar usó la placa Arduino MKR1000, incluyendo en ella una batería de litio y un módulo WiFi, para garantizar la comunicación en todo momento. Para obtener las señales de temblor usó el sensor BNO055 IMU, el cual mide la aceleración, la velocidad angular y el campo magnético a lo largo de los ejes X, Y y Z.

En la siguiente figura se muestra la placa usada en el experimento.

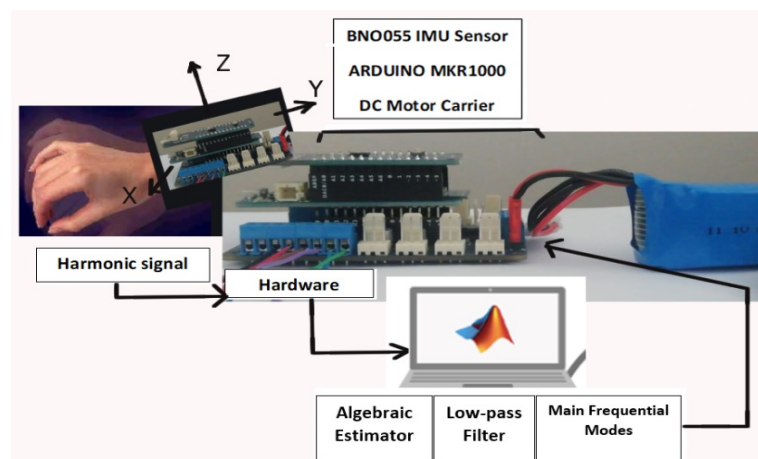


Figura 8.1: Plataforma experimental con un diagrama esquemático del sistema de estimación.

El sensor BNO055 IMU está configurado para devolver como salida una orientación absoluta en relación al campo magnético, gravitacional y la velocidad angular a lo

largo de las coordenadas locales del sensor. Dicho sensor pasa los datos en formato hexadecimal 0x28 mediante el protocolo I2C a la placa MKR1000, con una frecuencia de muestreo de 100 Hz.

Se realiza la captación de los datos por varios segundos, pero solo se trabaja con los datos en un periodo de tiempo de 1 segundo, ya que la captación de la frecuencia de las señales se realiza en un tiempo inferior a un segundo. Se recuerda que tanto Hassin-Baer y Sirota en [8] como Jang, Han y Park en [13] definieron la frecuencia del Parkinson dentro del rango de 4-8 Hz.

8 2

Evolución en el tiempo de las señales

Se observa la evolución en el tiempo de las señales que definen el temblor del Parkinson en los diferentes ejes (X, Y y Z).

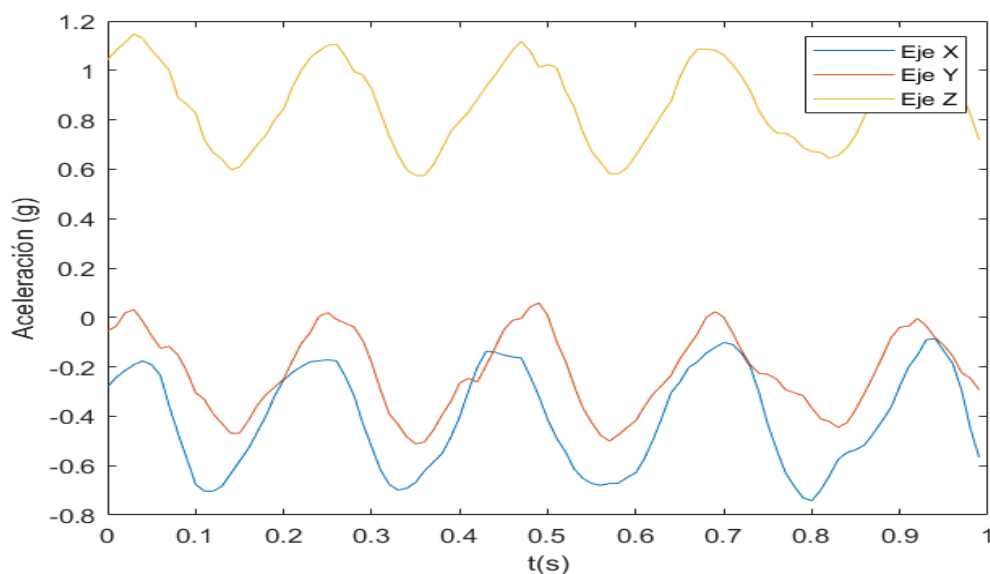


Figura 8.2: Evolución temporal de las medidas del acelerómetro por eje.

8 3

Estimación de los parámetros característicos de las señales

Se aplica el algoritmo construido para estimar los parámetros característicos de las señales. En primer lugar, se aplica el modelo sobre las frecuencias sin el centinela, con el objetivo de ver el comportamiento completo que tiene con las muestras proporcionadas.

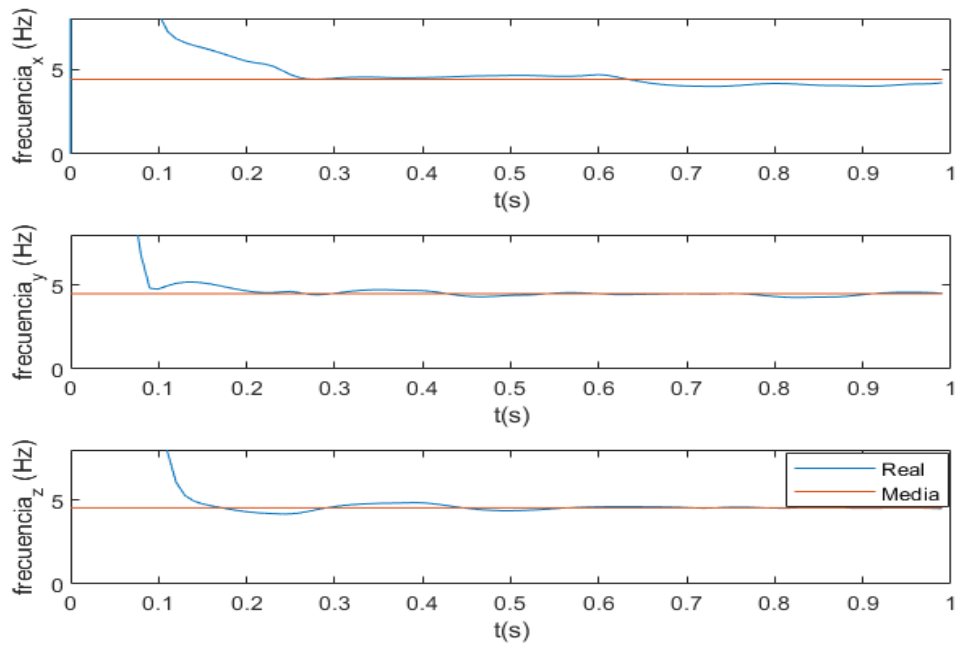


Figura 8.3: Evolución de la estimación de la frecuencia por eje.

Como se puede observar, el tiempo que tarda la señal en converger depende del nivel de ruido. A partir del instante 0.2 segundos las frecuencias empiezan a estabilizarse.

Se realiza un barrido de diferentes valores del rango, comprendidos entre 10 y 25 muestras y condiciones entre 0.04 y 0.08, con el objetivo de ver su comportamiento y seleccionar aquella que sea más óptima:

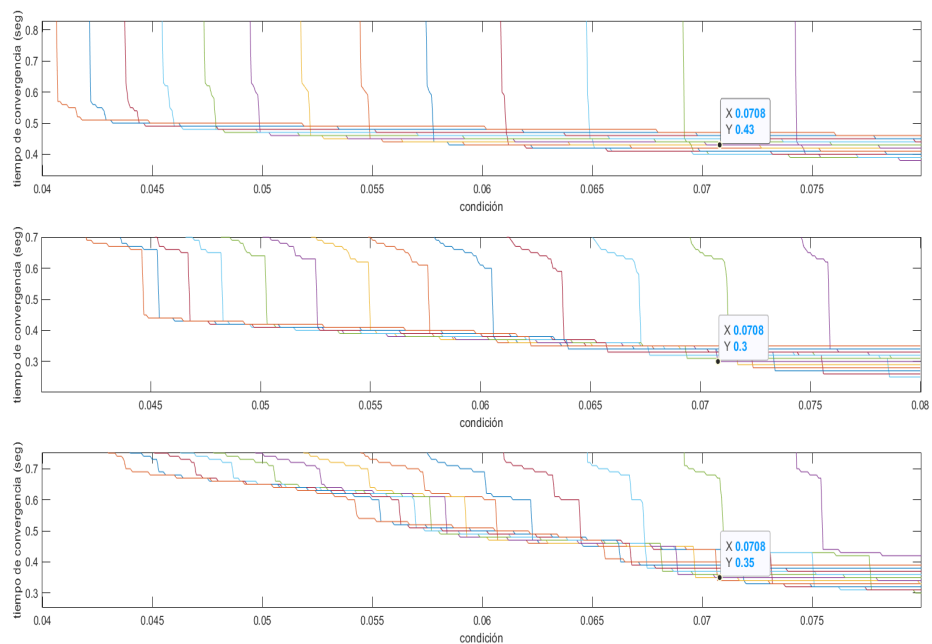


Figura 8.4: Selección del rango y su condición asociada por eje.

El punto de diseño se fija dentro del rango 0.3 y 0.4 segundos, con un rango de 20 muestras y una condición de 0.0708. Estos valores varían dependiendo de la aplicación, aumentando o disminuyendo el tiempo de convergencia. De forma adicional, se van a omitir las primeras 10 muestras debido a que se encuentran lejos de la zona de convergencia. Una vez configurado el centinela se ejecuta para observar las frecuencias en cada eje.

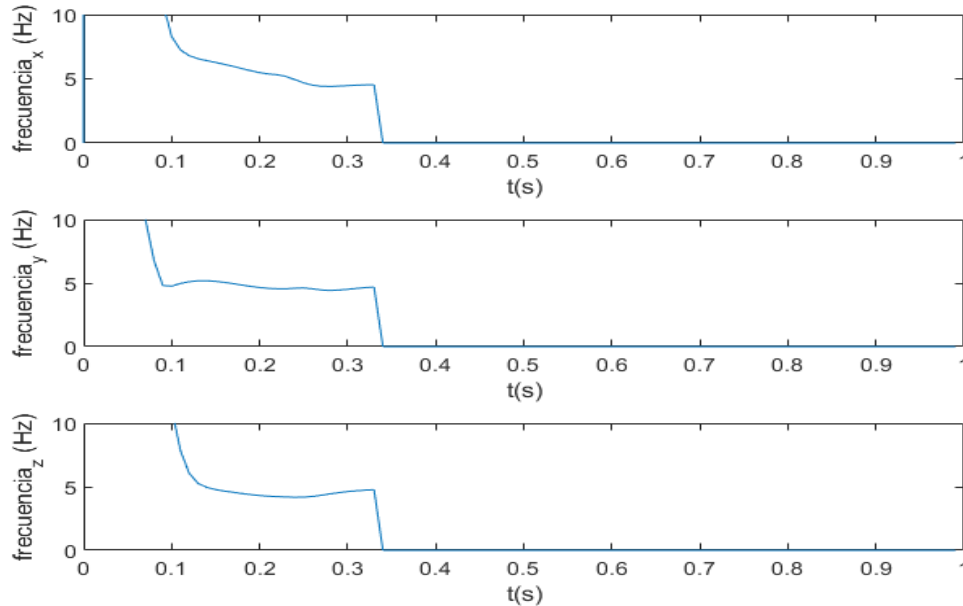


Figura 8.5: Evolución de la estimación de la frecuencia por eje.

Como se puede observar, el tiempo de estimación depende en gran medida del nivel de ruido y la distorsión que presente la señal proporcionada por el sensor. La frecuencia del eje x es 4.5270 Hz (28.3183 rad/s), para el eje y 4.6891 Hz (29.4625 rad/s) y para el eje z 4.7911 Hz (30.1034 rad/s). Como es lógico, las frecuencias de cada eje tienen que ser la misma, donde a partir de aquí hay dos posibilidades: detener la ejecución cuando se obtenga la primera frecuencia y devolver la frecuencia obtenida o realizar la media entre las tres frecuencias en dicho instante, ganando rapidez a la hora de obtener el valor, o seguir la ejecución hasta que las tres frecuencias cumplan la condición del centinela y se realice la media de las tres frecuencias, ganando precisión a la hora de obtener el valor.

Se realizan pruebas sobre los diferentes datos proporcionados y las diferentes formas de obtener el valor de la frecuencia, llegando a la conclusión de que el mejor método es obtener la primera frecuencia que encuentre el centinela, ya que no hay una gran diferencia con respecto a que todos los ejes cumplan la condición, debido a que se trata de la misma frecuencia. Y con respecto a realizar la media de las frecuencias, existe el principal problema de que un eje contenga tanto ruido que dicho valor proporcionado se encuentre lejos de la zona de convergencia, distorsionando el valor real de la frecuencia. En este caso al aplicar dicho método, la primera frecuencia que se encuentra es la del eje y con un valor de 4.6891 Hz (29.4625 rad/s) al tiempo de 0.34 segundos.

Una vez obtenida la frecuencia se pasa a determinar la amplitud, fase y el offset mediante el método basado en minimizar el error cuadrático. Se recuerda que, cuanto más precisión se obtenga la frecuencia, mayor precisión van a tener los demás parámetros.

Los valores parámetros para el eje x son: $amplitud_x = 0.2606$, $fase_x = 0.9629$ rad y $offset_x = -0.4097$; para el eje y: $amplitud_y = 0.2263$, $fase_y = 0.4808$ rad; y $offset_y = -0.2264$ y para el eje z: $amplitud_z = 0.2485$, $fase_z = 0.4705$ rad y $offset_z = 0.8381$. Una vez obtenidos los parámetros se recrean las diferentes señales y se comparan con las originales.

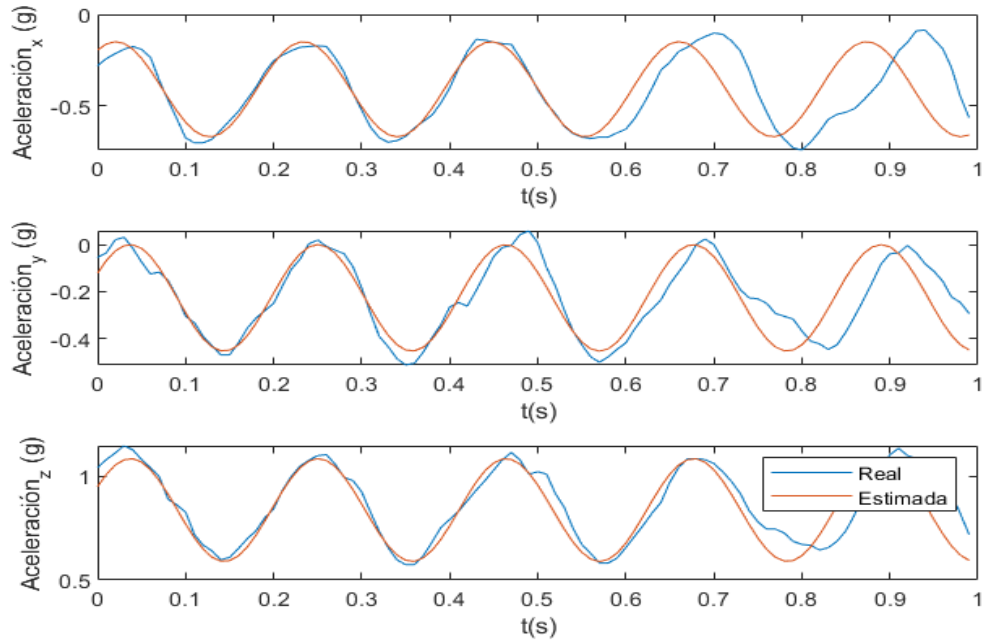


Figura 8.6: Comparación de la señal estimada con los datos reales.

El resultado obtenido es bastante positivo, ya que recrea con mucha precisión el temblor del Parkinson durante los primeros instantes. Pero a partir del instante 0.5 segundos, la señal estimada se desfasa con respecto a los datos originales. Para observar con más detalle el comportamiento, se realiza la diferencia entre la señal estimada y los datos reales.

Como se puede observar en la figura 8.7, la diferencia que hay entre ambas señales aumenta de forma progresiva. Esto se debe a que los parámetros de los temblores no son constantes, es decir, sus valores varían de forma ligera con respecto al tiempo, provocando que la señal estimada se descompense con respecto a los datos reales. Este problema se resuelve gracias a la implementación del centinela, ya que reinicia la ejecución del algoritmo dentro de un rango de tiempo entre 0.3 y 0.4 segundos, devolviendo como argumento de salida los nuevos parámetros de la señal. Gracias a esto, la señal estimada se ajusta a los datos reales antes de que se produzcan descompensaciones.

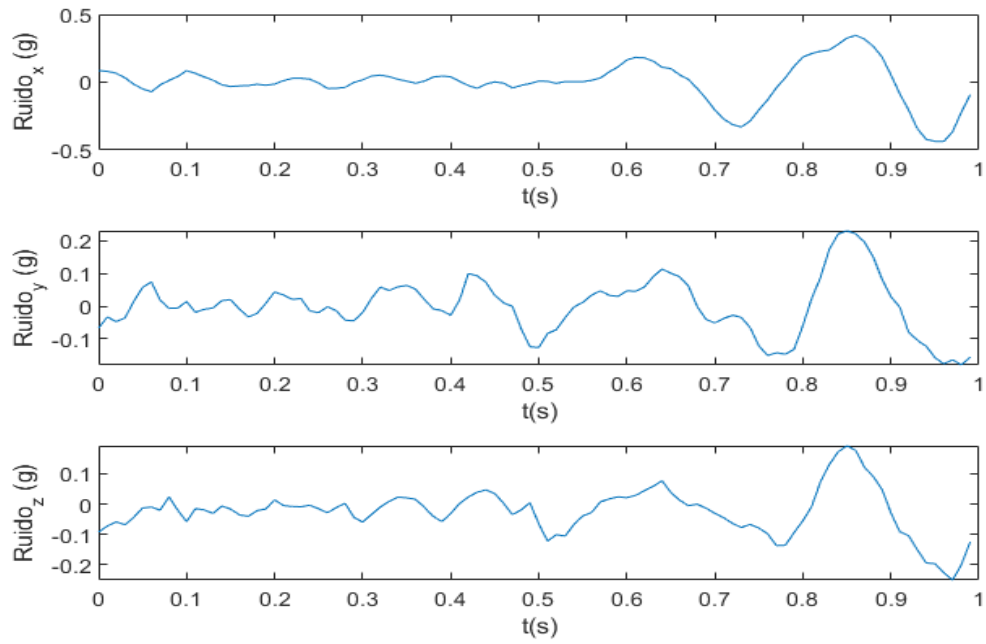


Figura 8.7: Diferencia entre la señal estimada y los datos reales.

Para ver la principal ventaja de dicho método se compara con la FFT, con el mismo número de muestras.

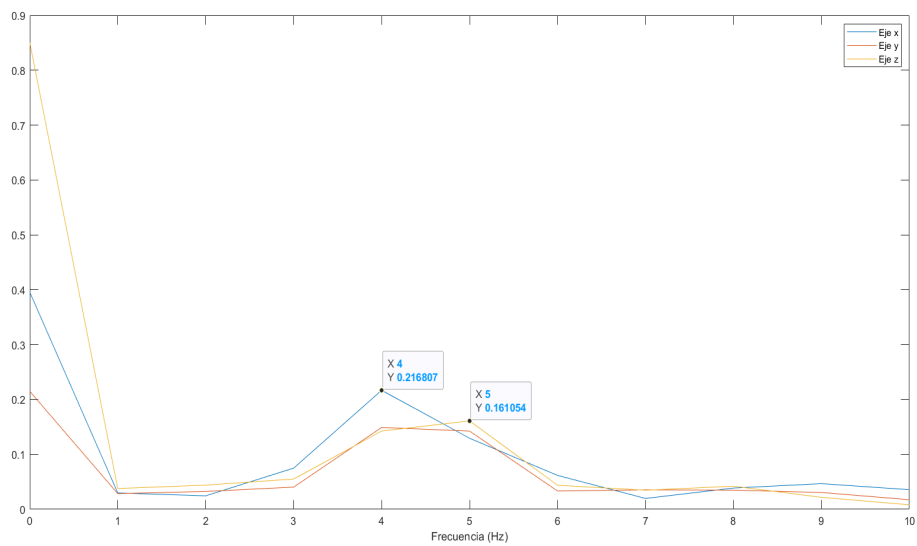


Figura 8.8: FFT de los datos reales.

Como se puede observar, el número de muestras es insuficiente para definir las frecuencias correspondientes, siendo imposible de determinar con un mínimo de exactitud sus valores reales.

CONCLUSIÓN

9 1

Características del algoritmo desarrollado

En este trabajo se ha presentado un método algebraico rápido y robusto basado en el dominio del tiempo para la estimación de los parámetros característicos de una sinusoidal.

Las principales ventajas que presenta son: su alta eficiencia, proporcionando datos en menos de un segundo; su robustez ante entornos ruidosos; y su independencia con respecto a parámetros de diseño y condiciones iniciales. Por ello, se convierte en un método ideal para aplicaciones en tiempo real, a la vez que universal para cualquier tipo de entrada, gracias a su falta de configuración.

Una vez identificado el comportamiento de cada parámetro mediante las ecuaciones de estado y su solución analítica, se han implementado tanto en Simulink como en Matlab. A partir de ahí, conociendo las ventajas y desventajas de cada uno, se ha optado por la eficiencia y portabilidad que ofrece Matlab. Siendo en este punto cuando ha aparecido el primer inconveniente; y es que el algoritmo se tiene que ejecutar en cadena, es decir, primero tiene que converger un parámetro y obtener su valor para poder encontrar el siguiente.

De forma alternativa se ha propuesto un método basado en la minimización del error cuadrático para la obtención de los parámetros de la amplitud, fase y offset. La principal ventaja de este método es que solo hace falta conocer la frecuencia para obtener las variables restantes. Realizándose en un tiempo de computación muy corto, sin necesidad de realizar otra estimación y esperar que la señal converja, resolviendo el problema del método algebraico.

El algoritmo ha sido mejorado con la implementación de un filtro paso bajo, con el objetivo de aislar los datos con los que se va a trabajar y un filtro no lineal basado en la integral de valores absolutos, que elimina el ruido no deseado y ayuda a la señal a converger. Además de resolverse uno de los principales problemas, como es el determinar cuándo se conoce con precisión el valor de la frecuencia para detener la ejecución del algoritmo mediante un centinela, que sigue la función en todo momento hasta que se cumpla la condición de precisión.

Para justificar la robustez del método se ha integrado en diferentes ambientes y situaciones; desde variaciones en la frecuencia de la señal, ambientes con una frecuencia de muestreo variable, en un ambiente con niveles de ruido muy altos e incluso en ambientes con diferentes señales que serán tratadas como ruido.

El resultado de las diferentes pruebas es positivo, comportándose de manera correcta ante los diferentes ambientes, salvo en un caso particular, que aparece cuando existe una señal con una frecuencia mayor a la de los temblores dentro del rango del filtro paso bajo, pues el algoritmo realiza la estimación de la frecuencia fundamental, ignorando los temblores del Parkinson.

Se ha aplicado el estimador propuesto para obtener los parámetros característicos de los temblores, gracias a los datos proporcionados por Claudia F. Yasar. Para aplicar el método, los únicos parámetros que se deben modificar son: la frecuencia de corte del filtro paso bajo y el ajuste de los valores del centinela. Configurándose este último, dando prioridad a la rapidez o la precisión en función de la aplicación y de los resultados que se pretendan obtener. Los resultados obtenidos son bastante satisfactorios, siendo el método capaz de converger y obtener la frecuencia en menos de un segundo, incluso con una alta presencia de ruido añadida por el entorno y el sensor.

Cabe señalar, que debido a la fácil implementación y el bajo esfuerzo computacional del método, lo hacen ideal para aplicaciones en tiempo real. Pero este método conlleva un inconveniente, y es que está diseñado para estimar los parámetros de una señal sinusoidal donde dichos parámetros son constantes, es decir, tanto la frecuencia como la amplitud, fase y el offset no varían. Pero gracias a su rápida estimación, el programa puede estar reiniciando constantemente el algoritmo, obteniendo cada 0.3 segundos los valores de los parámetros que se adecúan en todo momento a la señal obtenida.

9 2

Aportaciones y ventajas frente a trabajos previos

En este trabajo se han tomado como punto de partida los artículos de Claudia F. Yasar [26] y Trapero, Sira-Ramírez y Feliu-Batlle [23] para la estimación de los parámetros característicos de una señal sinusoidal mediante el método algebraico, mejorando de forma considerable su eficiencia.

En el artículo de Trapero, Sira-Ramírez y Feliu-Batlle [23] se ha propuesto el método de estimación basado en las soluciones analíticas de las ecuaciones de estado, ganando eficiencia y portabilidad a otros lenguajes de programación. Y el método basado en la minimización del error cuadrático para la estimación de la amplitud, fase y offset; resolviendo unos de los principales problemas de su ejecución en cadena, además de ganar precisión y rapidez a la hora de estimar los diferentes parámetros.

Con respecto al artículo de Claudia F. Yasar [26] se ha mejorado el tratamiento de los datos gracias a la implementación del filtro paso bajo de Butterworth y el filtro no lineal basado en la integral del valor absoluto. Además, se ha resuelto su principal problema de estimar la frecuencia de los temblores del Parkinson, debido a que sus parámetros no son constantes, produciéndose unas pequeñas variaciones que producen descompensaciones a lo largo del tiempo mediante la implementación del centinela, que reinicia el algoritmo obteniendo nuevos valores que se ajusten a la realidad.

9 3**Futuros trabajos**

A partir de este trabajo se pueden desarrollar principalmente tres líneas para futuros trabajos.

La primera de ellas consiste en mejorar el trabajo realiza por Claudia F. Yasar, desarrollando un sensor/captador que mejore la eficiencia en la toma de datos.

La segunda consiste en mejorar el trabajo realizado desarrollando un algoritmo que permita una estimación más eficiente a base de integrar las señales de cada eje y construir un algoritmo que sea capaz de distinguir cuándo el movimiento es involuntario, como lo es en el Parkinson, y cuándo el movimiento es voluntario para realizar una cierta tarea.

Y por último, una línea de trabajo aplicada a compensar los temblores del Parkinson, desarrollando un lápiz actuador mediante cristales piezoeléctricos, que proporcionando la señal estimada sea capaz de compensar los temblores.

PROPIEDADES DEL MODELO DE ESTADO

El vector de estado para un sistema lineal y casual presenta una serie de propiedades:

1. Unicidad.

$$\forall t \geq t_0, x_0 = x(t_0), \quad u(\tau) \quad t_0 < \tau \leq t \quad \Rightarrow \quad x(t) \text{ es única} \quad (\text{A.1})$$

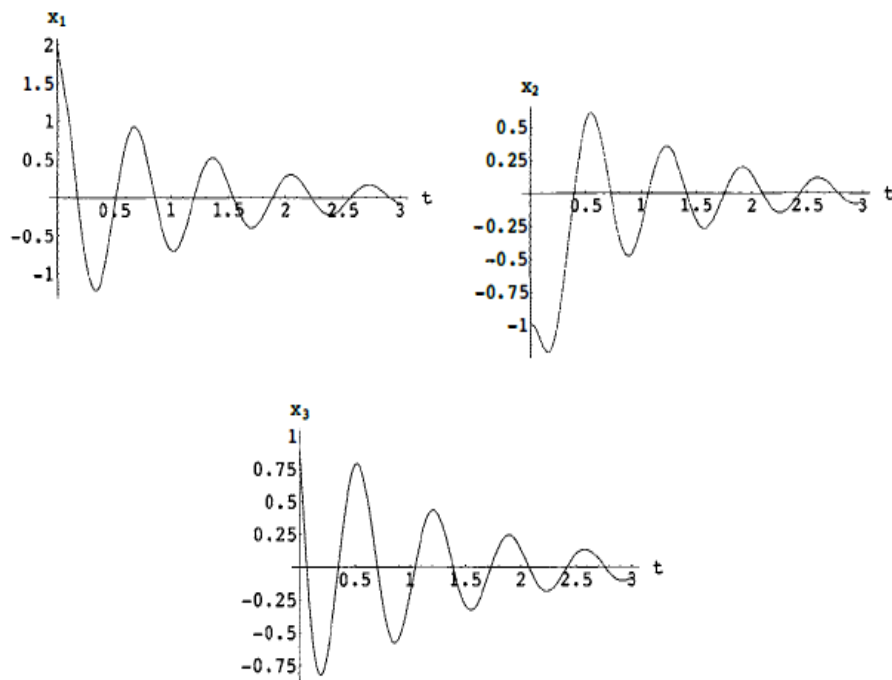


Figura A.1: Propiedad de unicidad.

2. Continuidad.

Las trayectorias en el espacio de estado son funciones continuas:

$$\lim_{t \rightarrow t_0} x(t) = x(t_0) \quad \forall t, t_0 \quad (\text{A.2})$$

3. Transitividad o propiedad de transición.

Si se considera en una trayectoria tres instantes, estos están relacionados por la propiedad de transición:

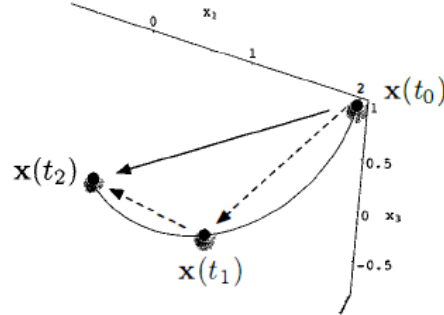


Figura A.2: Propiedad de transición.

$$x(t_2) = \psi(t_2, t_0, x(t_0), u(\tau)) \quad t_0 < \tau \leq t_2 \quad (\text{A.3})$$

$$x(t_2) = \psi(t_2, t_1, x(t_1), u(\tau)) \quad t_1 < \tau \leq t_2 \quad (\text{A.4})$$

Siendo:

$$x(t_1) = \psi(t_1, t_0, x(t_0), u(\tau)) \quad t_0 < \tau \leq t_1 \quad (\text{A.5})$$

Por lo que se puede conocer el instante t_2 cuando se cumplan las siguientes condiciones:

- Conocer el estado en t_0 y la entrada entre t_0 y t_2
- Conocer el estado en t_1 y la entrada entre t_1 y t_2

APÉNDICE **B**

TRANSFORMACIONES LINEALES

Las transformaciones lineales permiten trasladar un modelo de estado de una base a otra, mediante la denominada matriz de transición, la cual está compuesta por las coordenadas de los vectores de una base con respecto a otra. Las condiciones que debe cumplir la matriz de transformación es que tiene que ser no singular y tiene que existir la derivada de su inversa.

Para aplicar la transformación lineal se parte de las ecuaciones (2.3) y (2.4) y una matriz de transformación $t_{n \times n}$. El nuevo vector de estado $\tilde{x}(t)$ se define como:

$$x(t) = T(t)\tilde{x}(t) \quad (\text{B.1})$$

$$\tilde{x}(t) = T^{-1}(t)x(t) \quad (\text{B.2})$$

Se deriva la expresión $\tilde{x}(t)$ con respecto al tiempo:

$$\dot{\tilde{x}}(t) = \dot{T}^{-1}(t)x(t) + T^{-1}(t)\dot{x}(t) \quad (\text{B.3})$$

Cumpléndose que:

$$f(x) = \begin{cases} x(t) = T(t)\tilde{x}(t) \\ \dot{x}(t) = A(t)x(t) + B(t)u(t) \end{cases} \quad (\text{B.4})$$

Se sustituyen las ecuaciones (B.4) sobre la ecuación (B.3), quedando:

$$\begin{aligned} \dot{\tilde{x}}(t) &= \dot{T}^{-1}(t)T(t)\tilde{x}(t) + T^{-1}(t)[A(t)T(t)\tilde{x}(t) + B(t)u(t)] \\ &= [\dot{T}^{-1}(t)T(t) + T^{-1}(t)A(t)T(t)]\tilde{x}(t) + T^{-1}(t)B(t)u(t) \end{aligned} \quad (\text{B.5})$$

Que junto con la ecuación:

$$y(t) = C(t)T(t)\tilde{x}(t) + D(t)u(t) \quad (\text{B.6})$$

Definen el nuevo modelo de estado.

Se pueden aplicar de una forma más sencilla las transformaciones lineales si se realizan directamente sobre las matrices que definen el modelo de estado. Se parte de un modelo de estado ya existente $x(t)$; se puede obtener una nueva representación del modelo de estado $\tilde{x}(t)$, aplicando la matriz de transformación sobre las matrices, dando lugar a unas nuevas matrices del modelo de estado:

$$\tilde{A}(t) = [T^{-1}(t)T(t) + T^{-1}(t)A(t)T(t)]$$

$$\tilde{B}(t) = T^{-1}(t)B(t)$$

$$\tilde{C}(t) = C(t)T(t)$$

$$\tilde{D}(t) = D(t)$$

En la mayoría de casos, la matriz de transformación es invariante, es decir, sus elementos son constantes y no dependen del tiempo, por lo que las nuevas matrices del modelo de estado se simplifican, resultando:

$$\tilde{A}(t) = T^{-1}A(t)T$$

$$\tilde{B}(t) = T^{-1}B(t)$$

$$\tilde{C}(t) = C(t)T$$

$$\tilde{D}(t) = D(t)$$

MÉTODOS DE OBTENCIÓN DEL MODELO DE ESTADO

Las metodologías presentadas a continuación, están ordenadas de manera que las variables de estado presentan un orden decreciente en su significado físico y un orden creciente en cuanto a la simplicidad del modelo matemático. Considerando la primera de todas, la más intuitiva pero más tediosa, mientras que la última es la que presenta mayor simplicidad en las ecuaciones del modelo de estado resultante.

3 0 1 Variables de estado como magnitudes físicas

La idea básica es escoger como variables de estado las magnitudes físicas, es decir aquellos elementos que son capaces de acumular energía. La principal ventaja es que dichas variables de estados impiden que se presenten discontinuidades en el sistema. Estas variables pueden ser desde condensadores y corrientes, en sistemas eléctricos; posiciones y velocidades, en sistemas mecánicos; alturas de un fluido en sistemas hidráulicos; como temperaturas en sistemas térmicos. Es decir, toda variable que sea capaz de transmitir energía hasta conseguir un equilibrio energético (siempre que dicha transmisión no sea instantánea).

3 0 2 Variables de estado como salida de integradores

En este método se escogen las variables que presentan condiciones iniciales en las ecuaciones diferenciales que definen el comportamiento dinámico del sistema. Esto se debe a que dicha elección es equivalente a expresar dichas ecuaciones diferenciales en forma de integraciones sucesivas. Elegir como variable de estado la salida de cada uno de estos integradores tiene la ventaja de que su salida no presenta discontinuidades, incluso cuando la entrada es discontinua, pudiéndose aplicar tanto a sistemas lineales como no lineales, siendo válido para cualquier sistema. Dentro de este método se puede aplicar tanto a sistemas monovariable, multivariable y de orden reducido.

C 0 2 1 Variables de estado como salida de integrados en sistemas monovariantes

Para obtener las variables de estado en sistemas monovariantes, se parte de la ecuación general:

$$(s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0)y(t) = (b_ns^n + \cdots + b_1s + b_0)u(t) \quad (C.1)$$

Primero se reordena la ecuación y se saca factor común el operador derivada, s :

$$s[(s^{n-1} + a_{n-1}s^{n-2} + \cdots + a_1)y - (b_ns^{n-1} + \cdots + b_1)u] = b_0u - a_0y \quad (C.2)$$

El término de la izquierda se obtiene a través de la integración del término de la derecha, y es denominada como variable x_1 . La ecuación resultante es:

$$\begin{aligned} \dot{x}_1 &= b_0u - a_0y \\ x_1 &= (s^{n-1} + a_{n-1}s^{n-2} + \cdots + a_1)y - (b_ns^{n-1} + \cdots + b_1)u \end{aligned} \quad (C.3)$$

Dichas ecuaciones corresponden con la ecuación que describe la variable de estado y la ecuación de salida del integrador.

Si escoge la ecuación de salida del integrador x_1 y se agrupa, se obtiene:

$$s[(s^{n-2} + a_{n-1}s^{n-3} + \cdots + a_2)y - (b_ns^{n-2} + \cdots + b_2)u] = x_1 + b_1u - a_1y \quad (C.4)$$

Se repite el proceso anterior:

$$\begin{aligned} \dot{x}_2 &= x_1 + b_1u - a_1y \\ x_2 &= (s^{n-2} + a_{n-1}s^{n-3} + \cdots + a_2)y - (b_ns^{n-2} + \cdots + b_2)u \end{aligned} \quad (C.5)$$

Se realiza sucesivamente esta operación hasta obtener las siguientes ecuaciones:

$$\dot{x}_n = x_{n-1} + b_{n-1}u - a_{n-1}y \quad (C.6)$$

$$x_n = y - b_nu \quad (C.7)$$

Si se observa la ecuación (C.7) se puede obtener la relación de la salida con la variable estado, mientras que si se sustituye la salida dentro de la ecuación (C.6), se obtiene:

$$\begin{aligned} \dot{x}_i &= x_{i-1} - a_{i-1}x_n + (b_{i-1} - a_{i-1}b_n)u \\ y &= x_n + b_nu \end{aligned} \quad (C.8)$$

Se expresa de forma matricial, dando como resultado el siguiente modelo de estado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_0 - b_na_0 \\ b_1 - b_na_1 \\ b_2 - b_na_2 \\ \vdots \\ b_{n-1} - b_na_{n-1} \end{bmatrix} u \quad (C.9)$$

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} x + b_nu \quad (C.10)$$

C022 Variables de estado como salida de integradores en sistemas multivariables

El procedimiento anterior se puede aplicar para sistemas multivariables, que se representan por medio de un conjunto de ecuaciones diferenciales lineales. Se denomina u_i , con $i = 1, \dots, m$ a las distintas entradas del sistema y w_i , con $i = 1, \dots, q$ al resto de las variables involucradas, en las que dentro de estas se encuentran las variables de salida. Para los sistemas multivariables, hay q ecuaciones linealmente independientes, y cada una de ellas va a relacionar un conjunto U_j de entradas con un conjunto W_j de variables intermedias, de forma que:

$$\sum_{i \in U_j} (a_{ji,n_j} s^{n_j} + a_{ji,n_j-1} s^{n_j-1} + \dots + a_{ji,1} s + a_{ji,0}) \omega_i \quad (C.11)$$

$$= \sum_{i \in U_j} (b_{ji,n_j} s^{n_j} + b_{ji,n_j-1} s^{n_j-1} + \dots + b_{ji,1} s + b_{ji,0}) u_i, \text{ para } j = 1, \dots, q$$

verificándose que existe al menos uno de los coeficientes a_{ji,n_j} , que es no nulo.

Para cada una de las ecuaciones diferenciales se van eligiendo las variables de estado y se aplica el procedimiento descrito para sistemas monovariabes. El primer paso es ordenar y sacar factor común al operador derivada s :

$$s \left[\sum_{i \in W_j} (a_{ji,n_j} s^{n_j-1} + \dots + a_{ji,1}) w_i - \sum_{i \in U_j} (b_{ji,n_j} s^{n_j-1} + \dots + b_{ji,1}) u_i \right]$$

$$= \sum_{i \in U_j} b_{ji,0} u_i - \sum_{i \in W_j} a_{ji,0} w_i \quad (C.12)$$

Donde sí se elige como variable de estado el término dentro de los corchetes, la ecuación se puede descomponer de la forma:

$$x_{j1} = \sum_{i \in W_j} (a_{ji,n_j} s^{n_j-1} + \dots + a_{ji,1}) w_i - \sum_{i \in U_j} (b_{ji,n_j} s^{n_j-1} + \dots + b_{ji,1}) u_i$$

$$\dot{x}_{j1} = \sum_{i \in U_j} b_{ji,0} u_i - \sum_{i \in W_j} a_{ji,0} w_i \quad (C.13)$$

De igual forma que para los sistemas monovariabes, el procedimiento se reitera con la primera de las anteriores ecuaciones, obteniendo un conjunto de ecuaciones diferenciales de primer orden de la forma:

$$\dot{x}_{j,k} = x_{j,k-1} + \sum_{i \in U_j} b_{ji,k-1} u_i - \sum_{i \in W_j} a_{ji,k-1} w_i \quad \text{para } k = 2, \dots, n_j \quad (C.14)$$

junto con la ecuación:

$$x_{j,n_j} = \sum_{i \in W_j} a_{ji,n_j} \omega_i - \sum_{i \in U_j} b_{ji,n_j} u_i \quad (C.15)$$

La aplicación de este procedimiento a todas las ecuaciones diferenciales da lugar a un conjunto de n ecuaciones diferenciales de primer orden:

$$n = \sum_{j=1}^q n_j \quad (C.16)$$

Dichas ecuaciones diferenciales se pueden expresar de forma matricial:

$$\dot{x} = E_1x + B_1u + A_1\omega \quad (\text{C.17})$$

de igual forma que se obtiene las ecuaciones algebraicas de forma matricial:

$$E_2x + B_2u + A_2\omega = 0 \quad (\text{C.18})$$

A partir de esta ecuación se puede despejar ω , ya que la matriz A_2 está compuesta por ecuaciones iniciales que son linealmente independientes, siendo no singular:

$$\omega = -A_2^{-1}(E_2x + B_2u) \quad (\text{C.19})$$

sustituyendo en la ecuación (C.17) queda la siguiente expresión para las ecuaciones diferenciales:

$$\dot{x} = E_1x + B_1u - A_1A_2^{-1}(E_2x + B_2u) = (E_1 - A_1A_2^{-1}E_2)x + (B_1 - A_1A_2^{-1}B_2)u \quad (\text{C.20})$$

Las variables de salida comprenden un subconjunto de vectores y_i dentro de las variables de ω , obteniéndose directamente dicho vector mediante una matriz de selección S , de modo que:

$$y = S\omega - SA_2^{-1}(E_2x + B_2u) = -SA_2^{-1}E_2x - SA_2^{-1}B_2u \quad (\text{C.21})$$

C 0 2 3 Variables de estado como salida de sistemas de orden reducido

El procedimiento tanto para sistemas monovariantes como multivariantes es un proceso sistemático, donde las ecuaciones diferenciales del sistema se han descompuesto en integraciones sucesivas de los diferentes términos, elegidos como variables de estado, que representan la información del sistema en cada instante y no presentan discontinuidades en sus valores ante discontinuidades en la entrada.

Para elegir las variables de estado no es necesario descomponer las ecuaciones diferenciales del sistema hasta obtener los integradores puros, se pueden elegir alternatively como variables de estado, la salida de una ecuación diferencial de primer orden; o la salida y la derivada de una de segundo orden. El sistema se descompone en bloques de funciones de transferencia de orden reducido, donde se pueden elegir las salidas y las derivadas de sus salidas como variables de estado sin necesidad de profundizar de forma tan detallada con el método general.

3 0 3 Variables de estado de fase

Se vuelve a la expresión general para un sistema monovariante:

$$(s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0)y = (b_ms^m + \dots + b_1s + b_0)u \quad (\text{C.22})$$

Se puede reescribir como cociente entre dos polinomios:

$$y = \frac{b_ms^m + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}u \quad (\text{C.23})$$

El proceso para elegir las variables de estado consiste en elegir como variable de estado x_1 :

$$x_1 = \frac{1}{s^n + a_{n-1}s^{n-1} + \dots + a_0}u \quad (\text{C.24})$$

por lo que se ha elegido x_1 como solución de la ecuación diferencial en instante de u . Se eligen el resto de variables por sucesión de la derivada.

$$\begin{aligned} x_2 &= \dot{x}_1 \\ x_3 &= \dot{x}_2 \\ &\dots \\ x_n &= \dot{x}_{n-1} \end{aligned} \quad (C.25)$$

Por lo que las ecuaciones de estado se escriben como:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\dots \\ \dot{x}_{n-1} &= x_n \\ \dot{x} = s^n x_1 &= -a_{n-1}s^{n-1}x_1 - a_{n-2}s^{n-2} - \dots - a_1sx_1 - a_0x_1 + u = \\ &= -a_{n-1}x_n - a_{n-2}x_{n-1} - \dots - a_1x_2 - a_0x_1 + u \end{aligned} \quad (C.26)$$

Dichas ecuaciones de estado, se puede expresar de forma matricial:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (C.27)$$

La ecuación de salida es:

$$y = (b_ms^m + \dots + b_1s + b_0)x_1 \quad (C.28)$$

Expresada de forma matricial:

$$y = (b_0 \ b_1 \ b_2 \ b_m \ 0 \ \dots \ 0) \quad (C.29)$$

Cuando $n=m$, es decir, cuando el grado del numerador y el denominador de la función de transferencia son iguales, la ecuación de salida es de la forma:

$$y = (b_0 - b_na_0)x_1 + (b_1 - b_na_1) + \dots + (b_{n-1} - b_na_{n-1})x_n + b_nu \quad (C.30)$$

que se puede expresar de forma matricial mediante:

$$y = Cx + Du \quad (C.31)$$

La matriz D sólo aparece cuando el grado del numerador y el denominador es igual, y es la respuesta directa de la salida con respecto a la entrada, donde se superpone la respuesta dinámica.

3 0 4 Variables de Jordan

Se parte de la expresión:

$$y = \frac{b_m s^m + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} u \quad (\text{C.32})$$

donde se descompone la función de transferencia mediante las variables de Jordan. Hay dos casos posibles:

- Los polos son simples y se descompone en fracciones simples de la forma:

$$y = (b_n + \frac{\rho_1}{s - \lambda_1} + \frac{\rho_2}{s - \lambda_2} + \dots + \frac{\rho_n}{s - \lambda_n}) u \quad (\text{C.33})$$

A cada polo se le asigna una variable:

$$\begin{aligned} x_1 &= \frac{1}{s - \lambda_1} u & \dot{x}_1 &= \lambda_1 x_1 + u \\ x_2 &= \frac{1}{s - \lambda_2} u & \dot{x}_2 &= \lambda_2 x_2 + u \\ x_3 &= \frac{1}{s - \lambda_3} u & \dot{x}_3 &= \lambda_3 x_3 + u \\ &\dots & & \\ x_n &= \frac{1}{s - \lambda_n} u & \dot{x}_n &= \lambda_n x_n + u \end{aligned} \quad (\text{C.34})$$

donde la representación matricial es:

$$\dot{x} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} u \quad (\text{C.35})$$

Mientras que la ecuación de salida es:

$$y = \begin{pmatrix} \rho_1 & \rho_2 & \dots & \rho_n \end{pmatrix} x + b_n u \quad (\text{C.36})$$

- Al menos un polo tiene multiplicidad r , donde la descomposición de la funciones de transferencia es de la forma:

$$y = (b_n + \frac{\rho_1}{(s - \lambda_1)^r} + \dots + \frac{\rho_{r-1}}{(s - \lambda_n)^2} + \frac{\rho_r}{s - \lambda_1} + \frac{\rho_{r+1}}{s - \lambda_{r+1}} + \dots + \frac{\rho_n}{s - \lambda_n}) u \quad (\text{C.37})$$

A cada polo se le asigna una variable:

$$\begin{aligned} x_1 &= \frac{1}{(s - \lambda_1)^r} u = \frac{1}{s - \lambda_1} x_2 & \dot{x}_1 &= \lambda_1 x_1 + x_2 \\ x_2 &= \frac{1}{(s - \lambda_1)^{r-1}} u = \frac{1}{s - \lambda_1} x_3 & \dot{x}_2 &= \lambda_1 x_2 + x_3 \\ &\dots & & \\ x_{r-1} &= \frac{1}{(s - \lambda_1)^2} u = \frac{1}{s - \lambda_1} x_r & \dot{x}_{r-1} &= \lambda_1 x_{r-1} + x_r \\ &\dots & & \\ x_r &= \frac{1}{s - \lambda_1} u & \dot{x}_r &= \lambda_1 x_r + u \\ &\dots & & \\ x_n &= \frac{1}{s - \lambda_n} u & \dot{x}_n &= \lambda_n x_n + u \end{aligned} \quad (\text{C.38})$$

donde la representación matricial es:

$$\dot{x} = \begin{pmatrix} \lambda_1 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & \lambda_1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \lambda_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \lambda_{r+1} & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \lambda_n \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} u \quad (\text{C.39})$$

Mientras que la ecuación de salida es:

$$y = \begin{pmatrix} \rho_1 & \rho_2 & \cdots & \rho_{r-1} & \rho_r & \rho_{r+1} & \cdots & \rho_n \end{pmatrix} x + b_n u \quad (\text{C.40})$$

Cuando hay raíces múltiples, la representación se realiza por medio de bloques cuya dimensión depende del grado de la raíz y número superior es uno. Si la matriz presenta raíces múltiples deja de ser diagonal y la matriz correspondiente B contiene ceros salvo en el último elemento de la caja que es uno

PROPIEDADES DE LA MATRIZ DE TRANSICIÓN

Se describirán algunas propiedades de la matriz de transición que serán útiles en el desarrollo de la ecuación completa:

- Derivada con respecto al tiempo

Si se deriva la expresión genérica de la matriz de transición con respecto al tiempo se obtiene:

$$\frac{d\Phi(t, t_0)}{dt} = A(t) + A(t) \int_{t_0}^t A(\tau) d\tau + \dots = A(t)\Phi(t, t_0) \quad (\text{D.1})$$

Como se puede observar, se puede sacar factor común la matriz $A(t)$, quedando la suma infinita que define la matriz de transición.

- Valor de la matriz de transición en t_0

El valor de la variable $x(t)$ para el instante $t = t_0$ coincide con el valor especificado como condición inicial, x_0 , por lo que particularizando en ese instante, la matriz de transición es la matriz de identidad:

$$x(t_0) = \Phi(t_0, t_0)x(t_0) \quad \rightarrow \quad \Phi(t, t_0) = I \quad (\text{D.2})$$

De igual manera se llega a dicha conclusión mediante la serie de Peano-Baker, ya que todas las integrales se anulan, permaneciendo únicamente la matriz de identidad.

- Transitividad de la solución de la ecuación homogénea Se calcula la solución de la ecuación homogénea en un cualquier instante t_1 :

$$x(t_1) = \Phi(t_1, t_0)x(t_0) \quad (\text{D.3})$$

Se hace lo mismo para cualquier instante t_2 :

$$x(t_2) = \Phi(t_2, t_0)x(t_0) \quad (\text{D.4})$$

Como la solución de la ecuación de estado es única, se alcanzará el mismo estado para un instante determinado siempre que se parta de las mismas condiciones iniciales, de forma que:

$$x(t_2) = \Phi(t_2, t_1)x(t_1) = \Phi(t_2, t_1)\Phi(t_1, t_0)x(t_0) \quad (\text{D.5})$$

cumpliendo:

$$\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0) \quad (\text{D.6})$$

■ Inversión de tiempos

Si en la propiedad de transitividad se supone que $t_2 = t_0$, se cumple que:

$$\Phi(t_0, t_0) = I = \Phi(t_0, t_1)\Phi(t_1, t_0) \quad \rightarrow \quad \Phi(t_0, t_1) = \Phi^{-1}(t_1, t_0) \quad (\text{D.7})$$

■ Cambio de representación del estado

Si se realiza un cambio de base en la representación del estado mediante la matriz de transformación $T(t)$, siempre que exista $T^{-1}(t)$ se cumple que:

$$\tilde{\Phi}(t, t_0) = T^{-1}(t)\Phi(t, t_0)T(t_0) \quad (\text{D.8})$$

donde si se sustituye en la solución de la ecuación homogénea se obtiene:

$$x(t) = \Phi(t, t_0)x(t_0) \quad T(t)\tilde{x} = \Phi(t, t_0)T(t_0)\tilde{t}_0 \quad (\text{D.9})$$

CASOS PARTICULARES DE LA MATRIZ DE TRANSICIÓN

Existe un caso particular para la matriz de transición, que se produce cuando el producto de $A(t)$ y $\int_{t_0}^t A(\tau)d\tau$ es conmutativo. Entonces la matriz de transición se puede definir como:

$$\Phi(t, t_0) = e^{\int_{t_0}^t A(\tau)d\tau} \quad (\text{E.1})$$

Teniendo en cuenta que:

$$\frac{d}{dt} \left[\int_{t_0}^t A(\tau)d\tau \right]^n = n \left[\int_{t_0}^t A(\tau)d\tau \right]^{n-1} A(t) = n A(t) \left[\int_{t_0}^t A(\tau)d\tau \right]^{n-1} \quad (\text{E.2})$$

y volviendo al método general de Peano-Baker. Se aplica la propiedad anterior, quedando:

$$\begin{aligned} & \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1) \cdots \int_{t_0}^{\tau_{k-2}} A(\tau_{k-1}) \int_{t_0}^{\tau_{k-1}} A(\tau_k) d\tau_k d\tau_{k-1} \cdots d\tau_1 d\tau \\ &= \int_{t_0}^t A(\tau) \cdots \int_{t_0}^{\tau_{k-2}} \frac{1}{2} \frac{d}{d\tau_{k-1}} \left[\int_{t_0}^{\tau_{k-1}} A(\tau_k) d\tau_k \right]^2 d\tau_{k-1} d\tau_{k-2} \cdots d\tau \\ &= \int_{t_0}^t A(\tau) \cdots \int_{t_0}^{\tau_{k-3}} A(\tau_{k-2}) \left[\frac{1}{2} \left[\int_{t_0}^{\tau_{k-2}} A(\tau_k) d\tau_k \right]^2 \right] d\tau_{k-2} \cdots d\tau \end{aligned} \quad (\text{E.3})$$

Si se observa su comportamiento, se puede extraer un termino que define las sucesiones de la serie de Peano-Baker:

$$\frac{1}{k!} \left[\int_{t_0}^t A(\tau) d\tau \right]^k \quad (\text{E.4})$$

Definiendo la matriz de transición como:

$$\Phi(t, t_0) = I + \int_{t_0}^t A(\tau) d\tau + \frac{1}{2!} \left[\int_{t_0}^t A(\tau) d\tau \right]^2 + \cdots + \frac{1}{k!} \left[\int_{t_0}^t A(\tau) d\tau \right]^k + \cdots \quad (\text{E.5})$$

Se desarrolla la expresión mediante la conocida serie de Taylor, llegando a la conclusión que dicha expresión deriva a la ecuación (E.1).

Dentro de este caso particular, se pueden distinguir otros casos, que dependen de las propiedades de la matriz $A(t)$:

- Si la matriz $A(t)$ es diagonal, la ecuación (E.1) se puede desarrollar como:

$$\begin{aligned}
 \Phi(t, t_0) = e^{\int_{t_0}^t A(\tau) d\tau} &= e^{\begin{pmatrix} a_{11}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn}(\tau) \end{pmatrix} d\tau} = e^{\begin{pmatrix} \int_{t_0}^t a_{11}(\tau) d\tau & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \int_{t_0}^t a_{nn}(\tau) d\tau \end{pmatrix}} \\
 &= \begin{pmatrix} e^{\int_{t_0}^t a_{11}(\tau) d\tau} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\int_{t_0}^t a_{nn}(\tau) d\tau} \end{pmatrix} \quad (E.6)
 \end{aligned}$$

- Si la matriz $A(t)$ se puede factorizar, la ecuación (E.1) se puede desarrollar como:

$$\begin{aligned}
 A(t) &= M\alpha(t) \\
 \Phi(t, t_0) &= e^{\int_{t_0}^t A(\tau) d\tau} = e^{\int_{t_0}^t M\alpha(\tau) d\tau} = e^{M \int_{t_0}^t \alpha(\tau) d\tau} \quad (E.7)
 \end{aligned}$$

- Si la matriz $A(t)$ es invariante, la matriz conmuta con su integral de forma que la ecuación (E.1) se puede desarrollar como:

$$\Phi(t, t_0) = e^{\int_{t_0}^t A(\tau) d\tau} = e^{A \int_{t_0}^t d\tau} = e^{A(t-t_0)} \quad (E.8)$$

PROGRAMAS EN SIMULINK

La estructura externa del bloque Espacio de estado variable de la toolbox de sistemas de control viene dada por la definición de modelo de estado:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (\text{F.1})$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (\text{F.2})$$

Mientras que su estructura interna es:

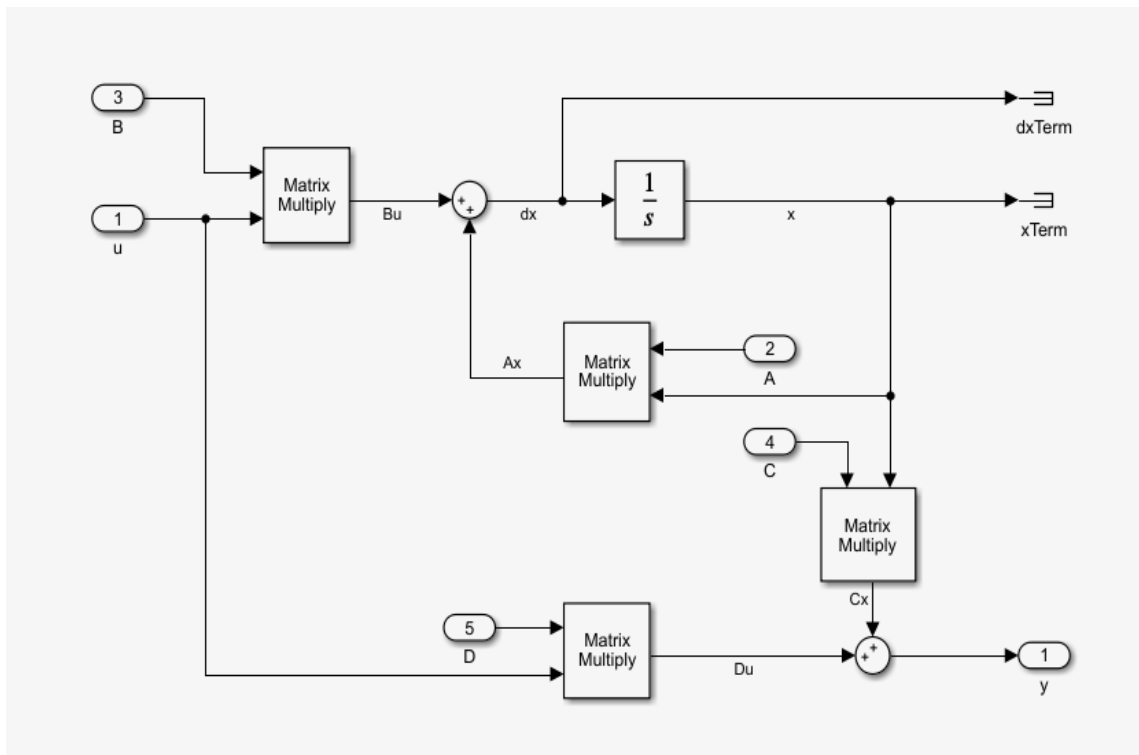


Figura F.1: Estructura interna del bloque espacio de estado variable.

Programas en Simulink para al resolución de la frecuencia:

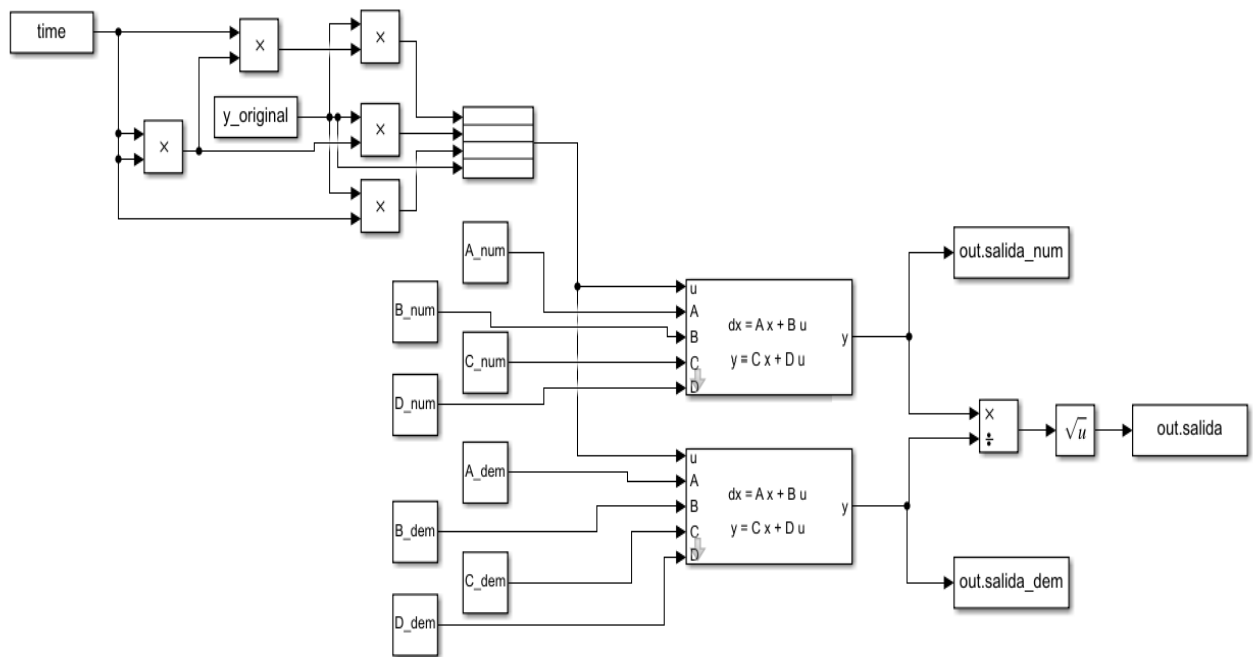


Figura F.2: Frecuencia en Simulink mediante modelo de estado

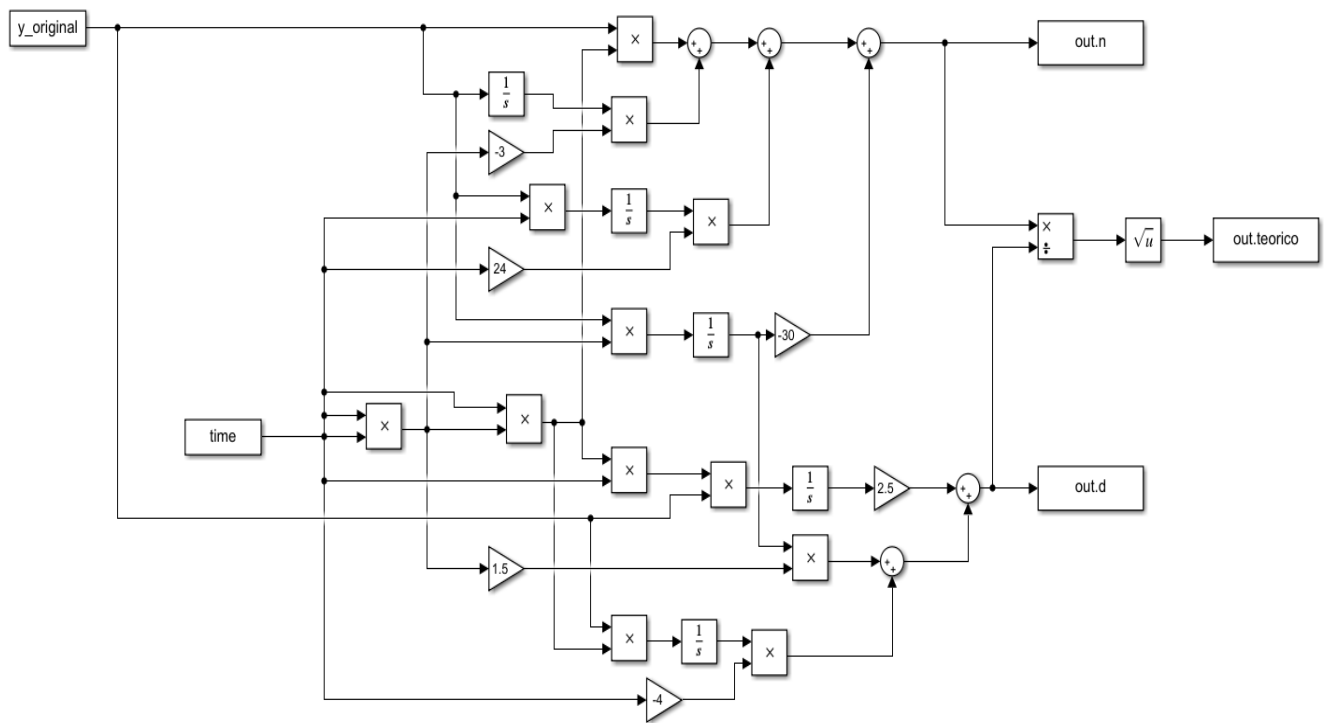


Figura F.3: Frecuencia en Simulink mediante modelo analítico

Programas en Simulink para la resolución de la variable p_3 :

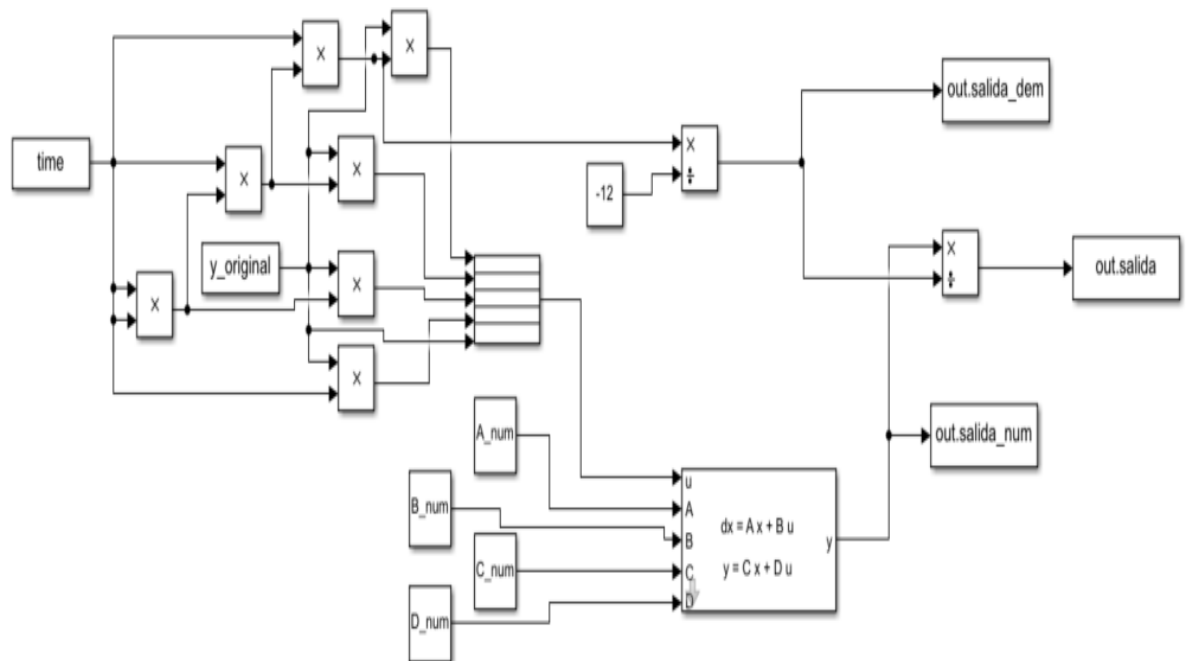


Figura F.4: Variable p_3 en Simulink mediante modelo de estado

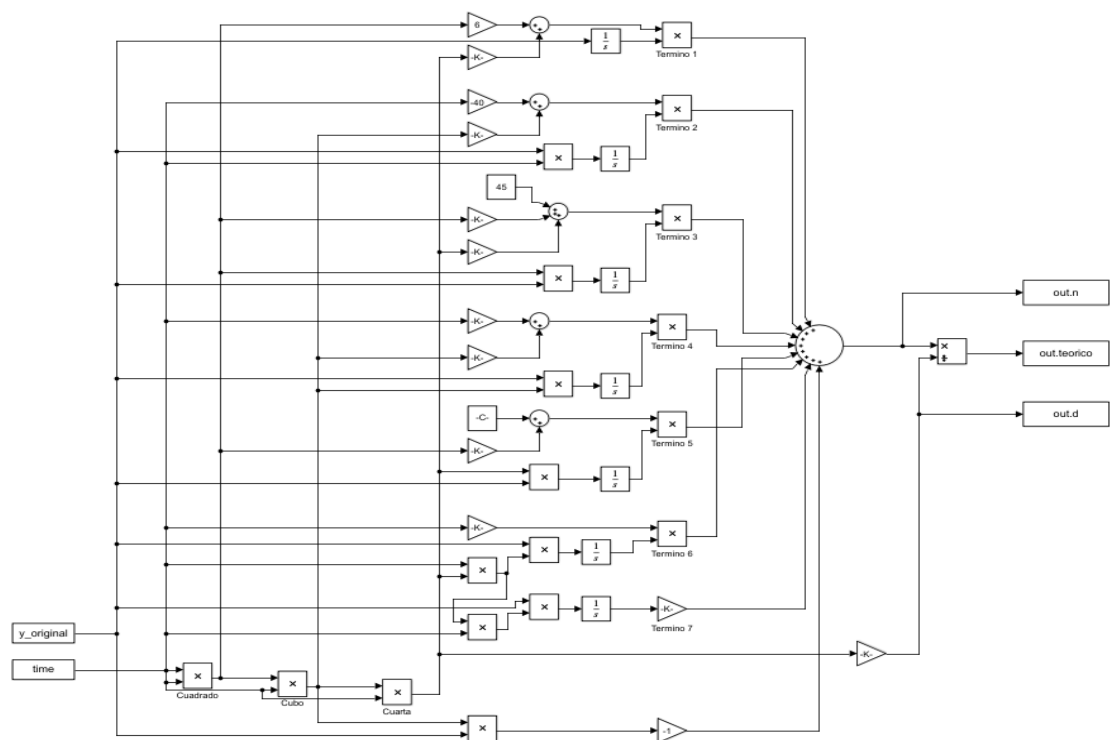


Figura F.5: Variable $p_3(t)$ en Simulink mediante modelo analítico

Programas en Simulink para la resolución de la variable p_2

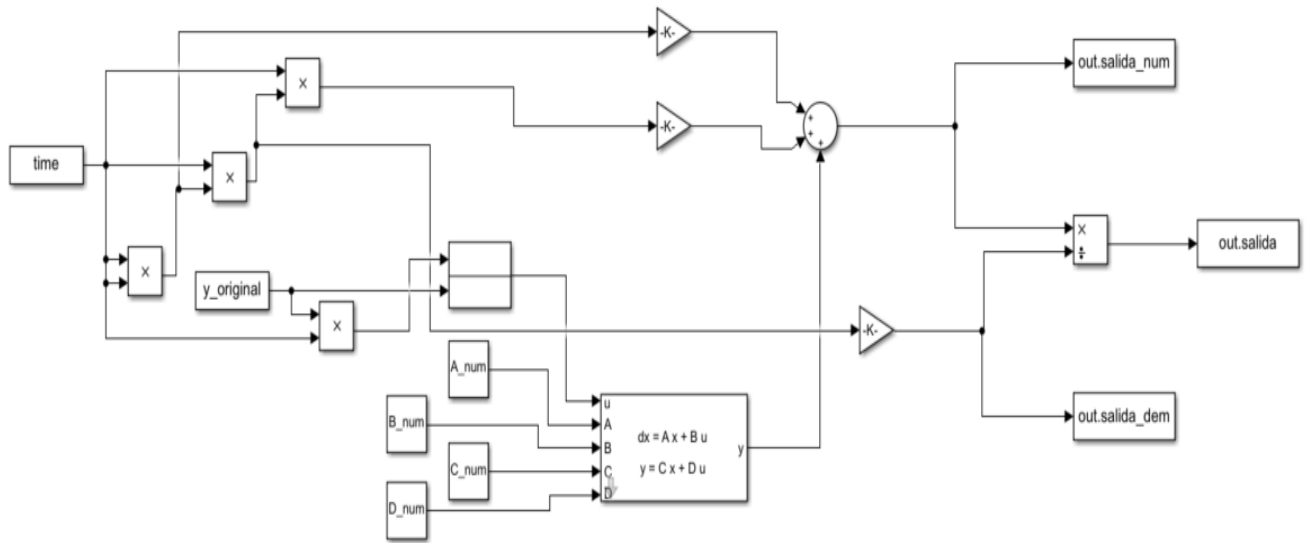


Figura F.6: Variable $p_2(t)$ en Simulink mediante modelo de estado

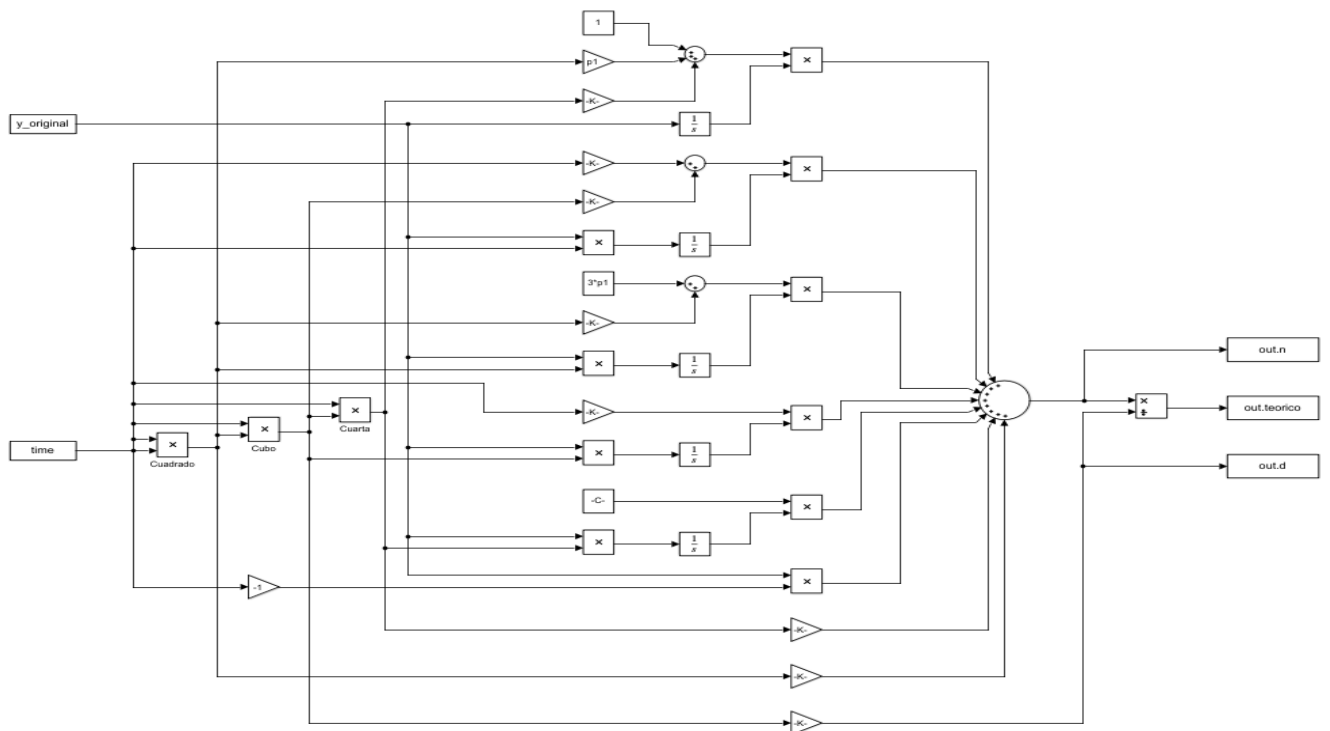


Figura F.7: Variable p_2 en Simulink mediante modelo analítico

CÓDIGO EN MATLAB

```

1  % Lectura de los datos de la señal.
2
3  filename='Tremor.xlsx';
4  datos=xlsread(filename);
5
6  % La columna 1 representa el número de muestra, la columna 2 el
7  % valor en el
8  % eje X, la columna 3 en el eje Y y la columna 4 en el eje Z.
9
10 % Se define el marco de tiempo en el que se trabaja.
11 t_sim=1;
12 h=0.01;
13
14 t=0:h:t_sim-h;
15
16 % Se representa los valores que se obtiene del sensor:
17 figure;
18 plot(t,datos(:,2)); hold on;
19 plot(t,datos(:,3)); hold on;
20 plot(t,datos(:,4)); hold on;
21 xlabel('t(s)'); ylabel('Aceleración (g)');
22 legend('Eje X', 'Eje Y', 'Eje Z');
23
24 %% Configuración del centinela.
25 rango=20; % Toma de datos.
26 margen=10;
27
28 condicion=0.0708;
29 limite=1;
30
31 t=0:h:limite-h;
32
33 %% Estimación de las frecuencias.
34 % Se inicia un contador para ver cuanto tiempo tarda en procesar
35 % los datos.

```

```

34  tic;
35
36  %Condiciones iniciales y asignación de los datos a cada eje.
37  pos=2;
38  tolerancia_x=Inf();
39  tolerancia_y=Inf();
40  tolerancia_z=Inf();
41
42  y_valores_x=datos(:,2);
43  y_valores_x=y_valores_x.';
44  y_valores_x=lowpass(y_valores_x,10,1/h);
45
46  n_x=zeros(1,length(t));
47  d_x=zeros(1,length(t));
48  f_x=zeros(1,length(t));
49  resultados_x=zeros(1,length(t));
50
51  y_valores_y=datos(:,3);
52  y_valores_y=y_valores_y.';
53  y_valores_y=lowpass(y_valores_y,10,1/h);
54
55  n_y=zeros(1,length(t));
56  d_y=zeros(1,length(t));
57  f_y=zeros(1,length(t));
58  resultados_y=zeros(1,length(t));
59
60  y_valores_z=datos(:,4);
61  y_valores_z=y_valores_z.';
62  y_valores_z=lowpass(y_valores_z,10,1/h);
63
64  n_z=zeros(1,length(t));
65  d_z=zeros(1,length(t));
66  f_z=zeros(1,length(t));
67  resultados_z=zeros(1,length(t));
68
69  % Ejecucion inicial del programa.
70  % Numerador Eje x
71  n1_x=(t(pos).^3).*y_valores_x(pos);
72  var=y_valores_x(1:pos);
73  n2_x=-3*(t(pos).^2).*trapz(t(1:pos),var);
74  var=var.*t(1:pos);
75  n3_x=+24*t(pos).*trapz(t(1:pos),var);
76  var=var.*t(1:pos);
77  n4_x=-30.*trapz(t(1:pos),var);
78  n_x(pos)=abs(n1_x+n2_x+n3_x+n4_x);
79  % Denominador Eje x
80  d1_x=1.5*(t(pos).^2).*trapz(t(1:pos),var);
81  var=var.*t(1:pos);
82  d2_x=-4*t(pos).*trapz(t(1:pos),var);
83  var=var.*t(1:pos);
84  d3_x=2.5*trapz(t(1:pos),var);

```

```

85 d_x(pos)=abs(d1_x+d2_x+d3_x);
86 % Frecuencia Eje x
87 f1_x=trapz(t(1:pos),n_x(1:pos));
88 f2_x=trapz(t(1:pos),d_x(1:pos));
89 f_x(pos)=sqrt(f1_x/f2_x);
90 resultados_x(pos)=tolerancia_x;
91
92 % Numerador Eje Y
93 n1_y=(t(pos).^3).*y_valores_y(pos);
94 var=y_valores_y(1:pos);
95 n2_y=-3*(t(pos).^2).*trapz(t(1:pos),var);
96 var=var.*t(1:pos);
97 n3_y=+24*t(pos).*trapz(t(1:pos),var);
98 var=var.*t(1:pos);
99 n4_y=-30.*trapz(t(1:pos),var);
100 n_y(pos)=abs(n1_y+n2_y+n3_y+n4_y);
101 % Denominador Eje Y
102 d1_y=1.5*(t(pos).^2).*trapz(t(1:pos),var);
103 var=var.*t(1:pos);
104 d2_y=-4*t(pos).*trapz(t(1:pos),var);
105 var=var.*t(1:pos);
106 d3_y=2.5*trapz(t(1:pos),var);
107 d_y(pos)=abs(d1_y+d2_y+d3_y);
108 % Frecuencia Eje Y
109 f1_y=trapz(t(1:pos),n_y(1:pos));
110 f2_y=trapz(t(1:pos),d_y(1:pos));
111 f_y(pos)=sqrt(f1_y/f2_y);
112 resultados_y(pos)=tolerancia_y;
113
114 % Numerador Eje Z
115 n1_z=(t(pos).^3).*y_valores_z(pos);
116 var=y_valores_z(1:pos);
117 n2_z=-3*(t(pos).^2).*trapz(t(1:pos),var);
118 var=var.*t(1:pos);
119 n3_z=+24*t(pos).*trapz(t(1:pos),var);
120 var=var.*t(1:pos);
121 n4_z=-30.*trapz(t(1:pos),var);
122 n_z(pos)=abs(n1_z+n2_z+n3_z+n4_z);
123 % Denominador Eje Z
124 d1_z=1.5*(t(pos).^2).*trapz(t(1:pos),var);
125 var=var.*t(1:pos);
126 d2_z=-4*t(pos).*trapz(t(1:pos),var);
127 var=var.*t(1:pos);
128 d3_z=2.5*trapz(t(1:pos),var);
129 d_z(pos)=abs(d1_z+d2_z+d3_z);
130 % Frecuencia Eje Z
131 f1_z=trapz(t(1:pos),n_z(1:pos));
132 f2_z=trapz(t(1:pos),d_z(1:pos));
133 f_z(pos)=sqrt(f1_z/f2_z);
134 resultados_z(pos)=tolerancia_z;
135

```

```

136
137 pos=pos+1;
138 % Ejecución de forma recursiva
139 while( pos<=length(t) && (tolerancia_x>=condicion &&
140     tolerancia_y>=condicion && tolerancia_z>=condicion))
141 % Numerador Eje X
142 n1_x=(t(pos).^3).*y_valores_x(pos);
143 var=y_valores_x(pos-1:pos);
144 n2_x=(t(pos).^2).*(n2_x./(t(pos-1).^2)-3*trapz(t(pos-1:pos),var))
145 ;
146 var=var.*t(pos-1:pos);
147 n3_x=t(pos).*(n3_x./t(pos-1)+24*trapz(t(pos-1:pos),var));
148 var=var.*t(pos-1:pos);
149 n4_x=n4_x-30.*trapz(t(pos-1:pos),var);
150 n_x(pos)=abs(n1_x+n2_x+n3_x+n4_x);
151 % Denominador Eje X
152 d1_x=(t(pos).^2).*(d1_x./(t(pos-1).^2)+1.5*trapz(t(pos-1:pos),var)
153 );
154 var=var.*t(pos-1:pos);
155 d2_x=t(pos).*(d2_x./t(pos-1)-4*trapz(t(pos-1:pos),var));
156 var=var.*t(pos-1:pos);
157 d3_x=d3_x+2.5*trapz(t(pos-1:pos),var);
158 d_x(pos)=abs(d1_x+d2_x+d3_x);
159 % Frecuencia Eje X
160 f1_x=f1_x+trapz(t(pos-1:pos),n_x(pos-1:pos));
161 f2_x=f2_x+trapz(t(pos-1:pos),d_x(pos-1:pos));
162 f_x(pos)=sqrt(f1_x/f2_x);
163
164 % Numerador Eje Y
165 n1_y=(t(pos).^3).*y_valores_y(pos);
166 var=y_valores_y(pos-1:pos);
167 n2_y=(t(pos).^2).*(n2_y./(t(pos-1).^2)-3*trapz(t(pos-1:pos),var))
168 ;
169 var=var.*t(pos-1:pos);
170 n3_y=t(pos).*(n3_y./t(pos-1)+24*trapz(t(pos-1:pos),var));
171 var=var.*t(pos-1:pos);
172 n4_y=n4_y-30.*trapz(t(pos-1:pos),var);
173 n_y(pos)=abs(n1_y+n2_y+n3_y+n4_y);
174 % Denominador Eje Y
175 d1_y=(t(pos).^2).*(d1_y./(t(pos-1).^2)+1.5*trapz(t(pos-1:pos),var)
176 );
177 var=var.*t(pos-1:pos);
178 d2_y=t(pos).*(d2_y./t(pos-1)-4*trapz(t(pos-1:pos),var));
179 var=var.*t(pos-1:pos);
180 d3_y=d3_y+2.5*trapz(t(pos-1:pos),var);
181 d_y(pos)=abs(d1_y+d2_y+d3_y);
182 % Frecuencia Eje Y
183 f1_y=f1_y+trapz(t(pos-1:pos),n_y(pos-1:pos));
184 f2_y=f2_y+trapz(t(pos-1:pos),d_y(pos-1:pos));
185 f_y(pos)=sqrt(f1_y/f2_y);

```

```

182 % Numerador Eje Z
183 n1_z=(t(pos).^3).*y_valores_z(pos);
184 var=y_valores_z(pos-1:pos);
185 n2_z=(t(pos).^2).*(n2_z./(t(pos-1).^2)-3*trapz(t(pos-1:pos),var));
186 ;
187 var=var.*t(pos-1:pos);
188 n3_z=t(pos).*(n3_z./t(pos-1)+24*trapz(t(pos-1:pos),var));
189 var=var.*t(pos-1:pos);
190 n4_z=n4_z-30.*trapz(t(pos-1:pos),var);
191 n_z(pos)=abs(n1_z+n2_z+n3_z+n4_z);
192 % Denominador Eje Z
193 d1_z=(t(pos).^2).*(d1_z./(t(pos-1).^2)+1.5*trapz(t(pos-1:pos),var));
194 ;
195 var=var.*t(pos-1:pos);
196 d2_z=t(pos).*(d2_z./t(pos-1)-4*trapz(t(pos-1:pos),var));
197 var=var.*t(pos-1:pos);
198 d3_z=d3_z+2.5*trapz(t(pos-1:pos),var);
199 d_z(pos)=abs(d1_z+d2_z+d3_z);
200 % Frecuencia Eje Z
201 f1_z=f1_z+trapz(t(pos-1:pos),n_z(pos-1:pos));
202 f2_z=f2_z+trapz(t(pos-1:pos),d_z(pos-1:pos));
203 f_z(pos)=sqrt(f1_z/f2_z);
204
205 % Seleccion de la frecuencia mediante el centinela
206 %Eje X
207 if (pos > rango + margen)
208 s1=sum(f_x(pos-rango:pos));
209 media=s1/rango;
210 if (media ~= Inf())
211 s2=sum((f_x(pos-rango:pos)-media).^2);
212 tolerancia_x=sqrt(s2/rango)/abs(media);
213 end
214 end
215 resultados_x(pos)=tolerancia_x;
216
217 %Eje Y
218 if (pos > rango + margen)
219 s1=sum(f_y(pos-rango:pos));
220 media=s1/rango;
221 if (media ~= Inf())
222 s2=sum((f_y(pos-rango:pos)-media).^2);
223 tolerancia_y=sqrt(s2/rango)/abs(media);
224 end
225 end
226 resultados_y(pos)=tolerancia_y;
227
228 %Eje Z
229 if (pos > rango + margen)
230 s1=sum(f_z(pos-rango:pos));
231 media=s1/rango;
232

```

```

231 if (media ~= Inf())
232 s2=sum((f_z(pos-rango:pos)-media).^2);
233 tolerancia_z=sqrt(s2/rango)/abs(media);
234 end
235 end
236 resultados_z(pos)=tolerancia_z;
237
238 pos=pos+1;
239 end
240
241 %Asignación de la frecuencia para cada eje.
242 freq_x=f_x(pos-1);
243 frecuencia_x=freq_x/(2*pi)
244 freq_y=f_y(pos-1);
245 frecuencia_y=freq_y/(2*pi)
246 freq_z=f_z(pos-1);
247 frecuencia_z=freq_z/(2*pi)
248
249 % Opción 1. Se elige las frecuencias y se realiza una media.
250 % freq=(freq_x+freq_y+freq_z)/3;
251 % frecuencia=freq/(2*pi)
252
253 % Opción 2. Se elige la primera frecuencia que cumpla la
    condición.
254 if (pos>length(t))
255 freq=(freq_x+freq_y+freq_z)/3;
256 frecuencia=freq/(2*pi)
257 else
258 real=[tolerancia_x<condicion tolerancia_y<condicion tolerancia_z<
    condicion];
259 freq=max(real.*[freq_x freq_y freq_z]);
260 frecuencia=freq/(2*pi)
261 end
262 %Se estima los parámetros de la amplitud, fase y constante y se
    recrea
263 %cada señal de forma independiente.
264 t1=1;
265 t2=pos-1;
266
267 % Eje X
268 B1=[trapz(y_valores_x(t1:t2).*sin(freq*t(t1:t2))); trapz(
    y_valores_x(t1:t2).*cos(freq*t(t1:t2))); trapz(y_valores_x(t1:t2
    ))];
269 A1=[trapz(sin(freq*t(t1:t2)).^2),trapz(cos(freq*t(t1:t2)).*sin(
    freq*t(t1:t2))), trapz(sin(freq*t(t1:t2))), trapz(cos(freq*t(t1:
    t2)).*sin(freq*t(t1:t2))),trapz(cos(freq*t(t1:t2)).^2), trapz(
    cos(freq*t(t1:t2))), trapz(sin(freq*t(t1:t2))), trapz(cos(freq*
    t(t1:t2))), t2-t1];
270 x_x=A1\B1;
271
272 fase_x=atan(x_x(2)/x_x(1))

```

```

273 Amplitud_x=sqrt(x_x(1)^2+x_x(2)^2)
274 Const_x=x_x(3)
275
276 %Eje Y
277 B1=[ trapz(y_valores_y(t1:t2).*sin(freq*t(t1:t2))); trapz(
    y_valores_y(t1:t2).*cos(freq*t(t1:t2))); trapz(y_valores_y(t1:t2)
    )]);
278 A1=[ trapz(sin(freq*t(t1:t2)).^2),trapz(cos(freq*t(t1:t2)).*sin(
    freq*t(t1:t2))), trapz(sin(freq*t(t1:t2))); trapz(cos(freq*t(t1:
    t2)).*sin(freq*t(t1:t2))),trapz(cos(freq*t(t1:t2)).^2), trapz(
    cos(freq*t(t1:t2))); trapz(sin(freq*t(t1:t2))), trapz(cos(freq*
    t(t1:t2))), t2-t1];
279 x_y=A1\B1;
280
281 fase_y=atan(x_y(2)/x_y(1))
282 Amplitud_y=sqrt(x_y(1)^2+x_y(2)^2)
283 Const_y=x_y(3)
284
285
286 %EJE Z
287 B1=[ trapz(y_valores_z(t1:t2).*sin(freq*t(t1:t2))); trapz(
    y_valores_z(t1:t2).*cos(freq*t(t1:t2))); trapz(y_valores_z(t1:t2)
    )]);
288 A1=[ trapz(sin(freq*t(t1:t2)).^2),trapz(cos(freq*t(t1:t2)).*sin(
    freq*t(t1:t2))), trapz(sin(freq*t(t1:t2))); trapz(cos(freq*t(t1:
    t2)).*sin(freq*t(t1:t2))),trapz(cos(freq*t(t1:t2)).^2), trapz(
    cos(freq*t(t1:t2))); trapz(sin(freq*t(t1:t2))), trapz(cos(freq*
    t(t1:t2))), t2-t1];
289 x_z=A1\B1;
290
291 fase_z=atan(x_z(2)/x_z(1))
292 Amplitud_z=sqrt(x_z(1)^2+x_z(2)^2)
293 Const_z=x_z(3)
294
295 t=0:h:t_sim-h;
296 y_aprox_x=Amplitud_x*sin(freq*t+fase_x)+Const_x;
297 y_aprox_y=Amplitud_y*sin(freq*t+fase_y)+Const_y;
298 y_aprox_z=Amplitud_z*sin(freq*t+fase_z)+Const_z;
299
300 %Se finaliza el contador para observar cuanto tiempo
    computacional ha
301 %ardado en estimar las frecuencias.
302 toc;
303
304
305 %Se representa los diferentes datos que se ha obtenido:
306 %Se representa los datos de la FFT.
307 figure;
308 subplot(3,1,1); plot(t,f_x./(2*pi));
309 xlabel('t(s)'); ylabel('frecuencia_x (Hz)');
310 axis([0 t_sim 0 10]);

```

```

311 subplot(3,1,2); plot(t,f_y./(2*pi));
312 xlabel('t(s)'); ylabel('frecuencia_y (Hz)');
313 axis([0 t_sim 0 10]);
314 subplot(3,1,3); plot(t,f_z./(2*pi));
315 xlabel('t(s)'); ylabel('frecuencia_z (Hz)');
316 axis([0 t_sim 0 10]);
317
318 %Se representa los datos del estimador algebraico en comparación
319 % con los
320 %datos de entrada.
321 figure;
322 subplot(3,1,1); plot(t,datos(:,2)); hold on; plot(t,y_aprox_x);
323 xlabel('t(s)'); ylabel('Aceleración_x (g)');
324 subplot(3,1,2); plot(t,datos(:,3)); hold on; plot(t,y_aprox_y);
325 xlabel('t(s)'); ylabel('Aceleración_y (g)');
326 subplot(3,1,3); plot(t,datos(:,4)); hold on; plot(t,y_aprox_z);
327 xlabel('t(s)'); ylabel('Aceleración_z (g)');
328 legend('Real','Estimada');
329
330 %Se realiza la inversa para de la señal estimada sobre los datos
331 % reales
332 %pero ver la cantidad de ruido que se produce.
333 figure;
334 subplot(3,1,1); plot(t,y_aprox_x-datos(:,2));
335 xlabel('t(s)'); ylabel('Ruido_x (g)');
336 subplot(3,1,2); plot(t,y_aprox_y-datos(:,3));
337 xlabel('t(s)'); ylabel('Ruido_y (g)');
338 subplot(3,1,3); plot(t,y_aprox_z-datos(:,4));
339 xlabel('t(s)'); ylabel('Ruido_z (g)');

```

BIBLIOGRAFÍA

- [1] Andersson T.; Händel P., Cramér-Rao bound and the Parsimony principle, *IEEE Trans. Instrum. Meas.*, Vol. 55, 2006, 44-53.
- [2] Bobtosov A.; Nikolaev N.; Slita O., Frequency identification of biased harmonic disturbance, *IFAC Proceedings Volumes*, Vol. 41, 2008, 11403-11408.
- [3] Clarke D.W., On the design of adaptive notch filter, *International Journal of Adaptive Control and Signal Processing*, Vol. 15, 2001, 715-744.
- [4] Domínguez S.; Campoy P.; Sebastián J.M.; Jiménez A., *Control en el espacio de Estado*, 2º ed., Pearson Educación S.A., Madrid, 2006, 1-107.
- [5] Fliess M.; Sira-Ramírez H., An algebraic framework for linear identification, *ESAIM Control Optim, Calc*, Vol. 9, 2003, 151-168.
- [6] Fliess M.; Glad S.T., An algebraic approach to linear and nonlinear control, *Essays in Control. Perspective in the Theory and its Applications*, Vol 14, Birkhäuser, 223-267.
- [7] Garcia-Rodriguez C.; Cortes-Romero J.A. y Sira-Ramírez H., Algebraic identification and discontinuous control for trajectory tracking in a perturbed 1-DOF suspension system, *IEEE Transactions on Industrial Electronics*, Vol. 56, 2009, 3665-3674.
- [8] Hassin-Baer S.; Sirota P., Clinical characteristics of neuroleptic-induced Parkinsonism, *Journal of Neural Transmission*, Vienna, 2001, 1299-1308.
- [9] Hayes M., *Digital Signal Processing*, 2º ed., McGraw Hill, 2012, 262-273.
- [10] Haykin S., *Adaptive Filter Theory*, 2º ed., Englewood Cliffs, Prentice-Hall, New Jersey, 1991.
- [11] Hou M., Amplitude and frequency estimator of a sinusoid, *IEEE Trans. Automat. Control*, Vol. 50, 2005, 855-858.
- [12] Hsu L.; Ortega R.; Damm G., A globally convergent frequency estimator, *IEEE Trans. Automat. Control*, Vol. 44, 1999, 698-713.
- [13] Jang W.; Han J.; Park J., Waveform analysis of tremor may help to differentiate Parkinson's disease from drug induced Parkinsonism, *Physiological Measurement*, 2013, 15-24.

- [14] Kay S.M.; Marple S.L., Spectrum Analysis - A modern perspective, Proceedings of IEEE, Vol. 69, 1981, 1380-1419.
- [15] Li T.H.; Kedem B., Strong consistency of the contraction mapping method for frequency estimation, IEEE Trans. Inform. Theory, Vol. 39, 1993, 989-998.
- [16] Mojiri M.; Bakhshai A.R., An adaptative notch filter for frequency estimation of a periodic signal, IEEE Trans. Automat. Control, Vol. 49, 2004, 314-318.
- [17] Pereira E.; Trapero JR.; Muñoz I.; Feliu-Batlle V., Algebraic identification of the first two natural frequencies of flexible-beamlike structures, Mechanical Systems and Signal Processing, Vol. 25, 2011, 2324-2335.
- [18] Rife D.C.; Boorstyn R.R., Single-tone parameter estimation from discrete-time observations, IEEE Trans. Inform. Theory, Vol. 20, 1974, 591-598.
- [19] Roy R.; Kailath T., ESPRIT-estimation of signal parameters via rotational invariance techniques, IEEE Trans. Signal Process, Vol. 37, 1989, 984-995.
- [20] San-Millan A.; Cambera J., Feliu-Batlle V., Online Algebraic Identification of the Payload Changes in a Single-Link Flexible Manipulator Moving under Gravity IFAC Proceedings Volumes, 2014, 8397-8402.
- [21] San-Millan A.; Aphale S.; Feliu-Batlle V., A Fast Algebraic Estimator for System Parameter Estimation and Online Controller Tuning - A Nanopositioning Application, Transactions on Industrial Electronics, Vol. 66, 2019, 4534-4543.
- [22] Stoica P., List of references on spectral line analysis, Signal Process, Vol. 31, 1993, 329-340.
- [23] Trapero JR.; Sira-Ramírez H.; Feliu-Batlle V., An algebraic frequency estimator for a biased and noisy sinusoidal signal, Signal Processing Vol. 87, 2007, 1188-1201.
- [24] Unbehauen H.; Rao G., A review of identification in cotinuous-time systems, Annual Reviews in Control, Vol. 22, 1998, 145-171.
- [25] Xiao Y.; Tadokoro Y., LMS-based notch filter for the estimation of sinusoidal signals in noise, Signal Processing, Vol. 46, 1995, 223-231
- [26] Yasar C., Algebraic estimator of Parkinson's tremor frequency from biased and noisy sinusoidal signals, Transactions of the Institute of Measurement and Control, Vol. 43, 2021, 679-686.
- [27] Young P., Parameter estimation for continuous-time models - a survey, Automatica, Vol.17, 1981, 23-29.
- [28] Ziarani A.K.; Konrad A., A method of extraction of nonstationary sinusoids, Signal Process, Vol. 84, 2004, 1323-1346.
- [29] Federación Española de Parkinson, Conocer la enfermedad, link: <https://www.esparkinson.es/espacio-parkinson/conocer-la-enfermedad/>.
- [30] Wikipedia, Enfermedad de Parkinson, link: https://es.wikipedia.org/wiki/Enfermedad_de_Parkinson.