

**UNIVERSIDAD NACIONAL DE EDUCACION A DISTANCIA  
ESCUELA TECNICA SUPERIOR DE INGENIERIA INFORMATICA**

**Máster en Ingeniería de Sistemas y de Control**

**Estudio de Controladores PI para Sistemas con Retardo**

**Autor: FERNANDO GARCIA SEDANO**

**Dirigido por: Dña. MARIA GUINALDO LOSADA  
D. SEBASTIAN DORMIDO BENCOMO**

**Curso: 2.016 – 2.017, Convocatoria de Junio**





**UNIVERSIDAD NACIONAL DE EDUCACION A DISTANCIA  
ESCUELA TECNICA SUPERIOR DE INGENIERIA INFORMATICA**

**Máster en Ingeniería de Sistemas y de Control**

Proyecto Específico (Tipo A)

**Estudio de Controladores PI para Sistemas con Retardo**

**Autor: FERNANDO GARCIA SEDANO**

**Dirigido por: Dña. MARIA GUINALDO LOSADA**

**D. SEBASTIAN DORMIDO BENCOMO**

**Curso: 2.016 – 2.017, Convocatoria de Junio**



Tribunal Calificador:

Presidente: D./D<sup>a</sup>. .....

Secretario: D./D<sup>a</sup>. .....

Vocal: D./D<sup>a</sup>. .....

Fecha de Lectura y Defensa: .....

Calificación: .....





## Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: **Fernando García Sedano**



# Resumen

---

El presente trabajo analiza la estabilidad en bucle cerrado de los sistemas con retardo, a los que se incorpora un regulador PI para alcanzar dicha estabilidad y unas especificaciones dinámicas determinadas. La importancia de este tipo de sistemas proviene del hecho de que muchos procesos físicos e industriales pueden modelarse mediante sistemas de primer orden con un término de retardo, y de que en la mayoría de los casos el regulador PI es el regulador más sencillo posible suficiente para garantizar las especificaciones del sistema.

Sin embargo, la existencia del término de retardo, dificulta el análisis de la estabilidad del sistema, así como la elección de los parámetros del regulador, para lo que se describen algunas herramientas matemáticas que permiten gestionar dicha dificultad con la suficiente precisión y exactitud, a la vez que se analizan técnicas (como la discretización o la aplicación del muestreo multifrecuencia) para llevar a cabo adecuadamente el control de este tipo de sistemas.

Por otra parte, actualmente es cada vez más frecuente la aplicación del concepto de control en red, de manera que el controlador de un proceso (o de varios de ellos), está ubicado lejos de la propia planta a controlar, y por tanto también lejos del sensor y del actuador, quedando los tres elementos antes mencionados unidos a través de una red de transmisión de datos digital. En este trabajo se simula dicha situación, se analiza la problemática que esta situación implica y se propone una solución simple para compensar los retardos que, como se estudiará, la red incorpora al sistema.

En este trabajo se utiliza como herramienta básica para simular las diferentes situaciones planteadas y el comportamiento de los sistemas bajo análisis MATLAB y SIMULINK; a través de los diferentes *scripts* de MATLAB y de las simulaciones realizadas con SIMULINK, se establecen diferentes conclusiones para cada caso analizado, siempre dándole al citado análisis una orientación lo más práctica posible.

# Abstract

---

The present work analyzes closed loop stability of systems with time-delay, in which also a PI controller is used to fulfil such stability and the corresponding dynamic specifications required for each application. The relevance of the systems with this topology is due to the fact that most of the real systems (physical or industrial) can be modelled as a first order dynamical system with an additional time-delay term, and because of PI controllers are usually the simplest ones that are able to achieve and guarantee the desired system performance.

Nevertheless, the existence of the time-delay term makes more difficult the stability analysis of the system, and also the proper selection of the controller parameters. To deal with this problem, some mathematical tools are described. These allow managing the problem with sufficient precision and accuracy; at the same time, different techniques (such as sampling of continuous-time systems with time delays or application of multirate sampling) are analyzed to perform the control of this type of systems.

Moreover, application of the networked control concept is becoming more frequent every day; it implies that the controller of any process (or several of them), is located far away from the plant to be controlled, and therefore also far from the sensor and from the actuator, being both sensor and actuator locally in the plant, and being all these three elements connected through a digital network. In this work, such situation is simulated and the problems that arise are analyzed, in order to propose a simple solution to compensate the delays that, as will be shown, the network add to the system.

In this work, MATLAB and SIMULINK are used as basic the tools to simulate the different situations considered, and the behaviour of the systems under test. With the support of several MATLAB scripts and SIMULINK simulations, different conclusions are established for each situation analyzed, always focusing such analysis as much as possible in a practical approach.

# Palabras Clave

---

Alternancia, Bifrecuencia, Bucle Abierto, Bucle Cerrado, CAN, Comunicaciones, Control, Cuasi-Polinomio, Ecuación Diferencial Funcional, Estabilidad, Estado, FOTD, Función de Transferencia, Hermite-Biehler, Hurwitz, MATLAB, Muestreo, Multifrecuencia, MPC, Naïve, Nodo, Nodo Interferente, Oscilación, Padé, PBC, Perturbación, PI, PID, Polinomio, Predictor de Smith, Raíces, Red, Regulador, Remoto, Retardo, RHC, Routh, SCBR, Simulación, Tarea Interferente, Trama, Vectores Ampliados.

# Key Words

---

Bifrequency, CAN, Closed Loop, Communications, Control, Controller, Delay, Differential Functional Equations, Disturbance, FOTD, Frame, Hermite-Biehler, Hurwitz, Interlacing, MATLAB, Multifrequency, MPC, Naïve, NCS, Network, Node, Interference Node, Interference Task, Oscillation, Padé, PBC, Quasipolynomials, Open Loop, PI, PID, Polynomial, Remote, RHC, Roots, Routh, Sampling, Shifted Vectors, Smith Predictor, Simulation, Stability, State, Transfer Function.



# Agradecimientos

---

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado de una manera o de otra a que este trabajo haya podido realizarse; en especial a María Guinaldo Losada y Sebastián Dormido Bencomo, por su tutorización y por la gran dosis de paciencia que han demostrado cuando el trabajo se me hacía más cuesta arriba de lo esperado.

No quiero dejar de mencionar a Carlos Mario Vélez, profesor e investigador de la Universidad EAFIT, que me ayudó en el aprendizaje del uso de la *Multirate Control Toolbox* (MCT) que el mismo desarrolló para su tesis doctoral, y que tan amablemente resolvió algunas de las dudas relativas a muestreo multifrecuencia con las que me tuve que enfrentar; a George Miminis, profesor de la *Memorial University of Newfoundland*, que actualizó para su uso por mi parte la *Toolbox PolePack* que el mismo desarrolló; a Ben M. Chen, profesor de la *National University of Singapore*, por su ayuda con la herramienta *Linear Systems Toolkit* para MATLAB; a Ángel Miguel Cuenca, profesor de la Escuela Técnica Superior de Ingenieros Industriales de Valencia, por darme ideas interesantes y útiles sobre el uso de la *Toolbox TrueTime* de aplicación en este trabajo; a Anton Cervin, profesor de la Universidad de Lund y coautor de la *Toolbox TrueTime*, por su paciencia conmigo a la hora de resolverme las dudas que tenía sobre el uso de dicha herramienta, y a los compañeros de la UNED, que como Ernesto Aranda, atendieron a mis peticiones de socorro desinteresadamente, y en general a todas aquellas personas que de una manera u otra me han dado su apoyo.

Por supuesto a mis padres Fernando y Adela, ya que soy lo que soy gracias a ellos, y a ellos les debo todo. Y a Alicia, a la que durante demasiado tiempo la he robado demasiadas horas para esta y otras aventuras similares: gracias por tu paciencia y amor.

Fernando Garcia Sedano,

Madrid, Junio de 2017.



<b>Introducción.....</b>	<b>1</b>
1.1. Resumen .....	1
1.2. Definición de Sistema Continuo con Retardo.....	2
1.3. Sistemas FOTD .....	3
1.4. Muestreo de Sistemas.....	5
1.5. Muestreo No Convencional.....	5
1.6. Sistemas de Control Basados en Red .....	7
1.7. Objetivos del Trabajo.....	9
1.8. Estructura de este trabajo .....	9
<b>Estabilidad de Sistemas FOTD.....</b>	<b>11</b>
2.1. Resumen .....	11
2.2. Estabilidad de Sistemas FOTD.....	12
2.3. Tipos de Aproximaciones Racionales .....	13
2.3.1. Aproximación de Naïve.....	13
2.3.2. Aproximación de Padé .....	14
2.4. El Predictor de Smith .....	16
2.5. El Teorema de Hermite-Biehler en la Teoría de Control.....	17
2.5.1. El Teorema de Hermite-Biehler para el caso continuo.....	18
2.5.2. El Teorema de Hermite-Biehler para el caso discreto.....	19
2.5.3. El Teorema de Hermite-Biehler para sistemas con retardo.....	19
2.6. Análisis de Estabilidad .....	20
2.7. Regulador PI .....	23
<b>Muestreo de Sistemas con Retardo .....</b>	<b>33</b>
3.1. Introducción.....	33
3.2. Muestreo de Sistemas con Retardo.....	33
3.3. Ejemplo práctico para un sistema FODT .....	36
3.4. Caso General: Sistemas y Regulador de cualquier orden.....	41

3.5.	Caso General: Ejemplo práctico para un sistema de 2° orden.....	43
<b>Sistemas con Muestreo Multifrecuencia .....</b>		<b>45</b>
4.1.	Introducción.....	45
4.2.	Modelos Multifrecuencia.....	46
4.3.	Método de Modelado Multifrecuencia.....	47
4.4.	Multirate Control Toolbox .....	49
4.5.	Caso de estudio.....	51
4.6.	Comparación caso Monofrecuencia – caso Bifrecuencia ( $\tau = 0$ ).....	51
4.7.	Comparación caso Monofrecuencia – caso Bifrecuencia ( $\tau \neq 0$ ) ....	53
4.8.	Extensión a sistemas de orden superior.....	55
4.9.	Comportamiento frente a perturbaciones.....	56
4.10.	Código MATLAB .....	65
<b>Sistemas de Control Basados en Red .....</b>		<b>69</b>
5.1.	Introducción.....	69
5.2.	La red CAN (Controller Area Network) .....	71
5.3.	<i>Toolbox TrueTime</i> para MATLAB.....	72
5.4.	Descripción Detallada de los bloques <i>TrueTime</i> .....	74
5.5.	Análisis y Simulación de un SCBR.....	75
5.6.	Simulaciones realizadas.....	79
5.6.1.	SCBR sin interferencias ni tareas parásitas ( $\tau = 0$ ).....	80
5.6.2.	SCBR con retardo intrínseco ( $\tau = 2$ s.).....	80
5.6.3.	SCBR con tarea adicional en el nodo controlador ( $\tau = 0$ ) .....	81
5.6.4.	SCBR con tarea en el nodo controlador y retardo ( $\tau = 2$ ).....	82
5.6.5.	SCBR con trafico adicional en red ( $\tau = 0$ ) .....	82
5.7.	Solución propuesta .....	85
5.7.1.	Estimación de los retardos producidos.....	85
5.7.2.	Estimación de los parámetros del regulador ( $\tau = 1$ , $\tau = 2$ ).....	86
5.7.3.	Cálculo de los parámetros del regulador. ....	87
5.7.4.	Resultados obtenidos .....	87
5.8.	Otras estrategias de compensación del retardo de red.....	89
5.8.1.	Packet-Based Control (PBC).....	89

5.8.2.	Receding Horizon Control (RHC) .....	90
5.9.	Comportamiento frente a perturbaciones.....	91
<b>Conclusiones y Trabajo Futuro.....</b>		<b>97</b>
6.1.	Conclusiones.....	97
6.2.	Trabajo Futuro .....	99
<b>Referencias y Bibliografía .....</b>		<b>101</b>
7.1.	Referencias.....	101
7.2.	Bibliografía .....	104



# Abreviaturas y Acrónimos

---

CAN: *Controlled Area Network*

EDF: Ecuaciones Diferenciales Funcionales

EDO: Ecuaciones Diferenciales Ordinarias

E/S: Entradas / Salidas

fdt: Función de Transferencia

FDMA: *Frequency Division Multiple Access*

FOTD: *First Order with Time Delay*

LTI: *Linear Time Invariant*

mcm: Mínimo Común Múltiplo

MCT: *Multirate Control Toolbox*

MIMO: *Multiple Input Multiple Output*

MPC: Mode Predictive Control

MRIC: *Multi-Rate Input Controller*

NCS: *Networked Control System*

P: Proporcional

PBC: *Packet-Based Control*

PD: Proporcional-Diferencial

PI: Proporcional-Integral

PID: Proporcional-Integral-Diferencial

RHC: *Receding Horizont Control*

SCBR: Sistema de Control Basado en Red

SISO: *Single Input Single Output*

TDMA: *Time Division Multiple Access*

URL: *Uniform Resource Locator*

WLAN: *Wireless Local Area Network*



# Índice de Figuras

---

Figura 1: Bucle de control con un sistema FOTD .....	3
Figura 2: Diagrama de Bode del Retardo.....	4
Figura 3: Sistema de Control Clásico .....	5
Figura 4: Esquema General de un Sistema Bifrecuencia .....	6
Figura 5: Esquema General de un Sistema de Control Basado en Red .....	8
Figura 6: Diagrama de Bode de $M(s)$ .....	15
Figura 7: Predictor de Smith .....	16
Figura 8: Predictor de Smith con Perturbación .....	17
Figura 9: Análisis de Estabilidad de un Sistema.....	21
Figura 10: Representación de $Fr(\omega)$ y $Fi(\omega)$ .....	21
Figura 11: Representación de $Fr(\omega)$ y $Fi(\omega)$ (Detalle).....	22
Figura 12: 'Script' MATLAB para representar curvas de estabilidad.....	22
Figura 13: Respuesta al escalón del sistema original.....	22
Figura 14: 'Script' para el cálculo del conjunto $\{k_p, k_i\}$ estable .....	27
Figura 15: 'Script' de la función <code>gzero()</code> .....	28
Figura 16: Lugar geométrico de pares $\{k_p, k_i\}$ estables de $G(s)$ .....	28
Figura 17: Respuesta al escalón de $G(s)$ con el par $\{k_p=4, k_i=2\}$ .....	28
Figura 18: Respuesta al escalón de $G(s)$ con el par $\{k_p=1, k_i=2\}$ .....	29
Figura 19: Respuesta al escalón de $G(s)$ $\{k_p=4, k_i=2\}$ (Padé, orden 1) ...	29
Figura 20: Lugar geométrico de pares $\{k_p, k_i\}$ estables de $H(s)$ .....	30
Figura 21: Respuesta del sistema $H(s)$ con el par $\{1.5, 0.25\}$ .....	30
Figura 22: Respuesta de $H(s)$ con Padé (1) y el par $\{1.5, 0.25\}$ .....	31
Figura 23: Sistemas con retardo temporal.....	34
Figura 24: Sistema FOTD con retardo temporal interno.....	34
Figura 25: Sistema FOTD sin/con retardo temporal interno.....	38
Figura 26: Sistema FOTD con diversos retardos .....	39
Figura 27: Script de MATLAB que calcula la respuesta del sistema FOTD...	40
Figura 28: Script de MATLAB: cálculo la respuesta de cualquier sistema ...	42
Figura 29: Esquema de un sistema multifrecuencia genérico.....	47
Figura 30: Vista de entrada de la ayuda de Multirate Control Toolbox.....	50
Figura 31: Respuesta del sistema mono y bifrecuencia ( $N=2$ ) .....	52
Figura 32: Respuesta del sistema mono y bifrecuencia ( $N=3$ ) .....	52
Figura 33: Respuesta del sistema mono y bifrecuencia.....	53
Figura 34: Respuesta del sistema mono y bifrecuencia ( $\tau=5$ ) .....	54
Figura 35: Respuesta del sistema mono y bifrecuencia ( $\tau=12$ ) .....	54
Figura 36: Respuesta del sistema mono y bifrecuencia ( $\tau=0$ ) .....	55
Figura 37: Respuesta del sistema mono y bifrecuencia ( $\tau=12$ ) .....	56
Figura 38: Respuesta del sistema mono y bifrecuencia ( $\tau=0$ ) .....	56
Figura 39: Sistema $G(s)$ con perturbaciones ( $P$ ) .....	57

Figura 40: Respuesta de $G(s)$ a una perturbación escalón ( $\tau=0$ ) .....	57
Figura 41: Respuesta de $G(s)$ a perturbación senoidal ( $\tau=0$ ) .....	58
Figura 42: Respuesta de $G(s)$ a perturbación cuadrada ( $\tau=0$ ) .....	59
Figura 43: Respuesta de $G(s)$ a una perturbación escalón ( $\tau=5$ s) .....	59
Figura 44: Respuesta de $G(s)$ a perturbación senoidal ( $\tau=5$ ) .....	60
Figura 45: Respuesta de $G(s)$ a perturbación cuadrada ( $\tau=5$ ) .....	60
Figura 46: Respuesta de $G(s)$ a una perturbación escalón ( $N=2, \tau=0$ ) .....	61
Figura 47: Respuesta de $G(s)$ a perturbación senoidal ( $N=2, \tau=0$ ) .....	61
Figura 48: Respuesta de $G(s)$ a perturbación senoidal ( $N=3, \tau=0$ ) .....	62
Figura 49: Respuesta de $G(s)$ a perturbación cuadrada ( $N=2, \tau=0$ ) .....	62
Figura 50: Respuesta de $G(s)$ a perturbación cuadrada ( $N=3, \tau=0$ ) .....	63
Figura 51: Respuesta de $G(s)$ a perturbación escalón ( $N=2, \tau=5$ ) .....	63
Figura 52: Respuesta a perturbación senoidal ( $N=2, \tau=5$ ) .....	64
Figura 53: Respuesta a perturbación cuadrada ( $N=2, \tau=5$ ) .....	64
Figura 54: Respuesta a perturbación cuadrada ( $N=3, \tau=5$ ) .....	65
Figura 55: Elementos de un Sistema de Control Basado en Red .....	70
Figura 56: Trama CAN .....	72
Figura 57: Librería de TrueTime .....	73
Figura 58: Configuración del SCBR simulado con TrueTime .....	76
Figura 59: Código de inicialización del nodo controlador .....	76
Figura 60: Código de la tarea 'controller_code' del nodo controlador .....	77
Figura 61: Código de inicialización del nodo sensor / actuador .....	78
Figura 62: Código de la tarea 'sensor_code' del nodo sensor / actuador ...	78
Figura 63: Código de la tarea 'actuador_code' del nodo sensor / actuador .	78
Figura 64: Código de inicialización del nodo interferente .....	79
Figura 65: Código de la tarea 'interference_code', nodo interferente .....	79
Figura 66: Respuesta de $G(s)$ sin interferencias ni retardos .....	80
Figura 67: Respuesta de $G(s)$ sin interferencias ( $\tau=2$ s) .....	81
Figura 68: Respuesta de $G(s)$ con tarea interferente ( $\tau=0$ s) .....	81
Figura 69: Respuesta de $G(s)$ con tarea interferente ( $\tau=2$ s) .....	82
Figura 70: Respuesta de $G(s)$ con tramas interferentes ( $\tau=0$ ) .....	83
Figura 71: Respuesta de $G(s)$ con tramas interferentes ( $\tau=0$ ) .....	84
Figura 72: Respuesta de $G(s)$ con tramas interferentes ( $\tau=2$ ) .....	84
Figura 73: Código de la función 'interpolate()' .....	87
Figura 74: Respuesta con trama interferente y compensación de retardo..	88
Figura 75: Respuesta con tarea interferente y compensación de retardo...	89
Figura 76: Diagrama de Bloques Simulink para perturbaciones .....	91
Figura 77: Respuesta del sistema a la perturbación escalón ( $\tau=0$ ) .....	92
Figura 78: Respuesta del sistema a la perturbación escalón ( $\tau=2$ ) .....	92
Figura 79: Respuesta del sistema a la perturbación senoidal ( $\tau=0$ ) .....	93

*Figura 80: Respuesta del sistema a la perturbación senoidal ( $\tau=5$ ) ..... 94*  
*Figura 81: Respuesta del sistema a la perturbación cuadrada ( $\tau=0$ ) ..... 94*  
*Figura 82: Respuesta del sistema a la perturbación cuadrada ( $\tau=5$ ) ..... 95*



## Introducción

---

### 1.1. Resumen

El presente trabajo analiza los sistemas LTI de primer y segundo orden con retardos, ya que son estos los sistemas que principalmente se usan en ingeniería para modelar los procesos físicos que se desea controlar [Silva-02]. En efecto: es muy difícil modelar un sistema real sin incluir un término de retardo en el mismo, ya que estos se presentan cuando concurren varias circunstancias, por otra parte muy frecuentes, que se dan en los procesos físicos e industriales [Muro-09]:

- El proceso tiene operaciones de transporte, especialmente en el caso de los fluidos.
- El proceso presenta fases de espera o "incubación".
- Los sensores que informan del estado de la variable que se desea controlar necesita tiempo para tomar mediciones, o para procesarlas.
- Los actuadores necesitan tiempo para cambiar el estado de algún elemento del proceso (válvulas, compuertas, etc.).
- Cuando los modelos de orden elevados se simplifican mediante un sistema de primer (o segundo) orden.
- Se utilizan redes de comunicaciones en el sistema.

Todas estas circunstancias tienen consecuencias negativas para la correcta controlabilidad del sistema, por varios motivos:

- Las perturbaciones no se detectan en el momento oportuno.
- La acción de control (que depende del instante en el que se haya tomado la medición) no se produce en el momento adecuado.
- La acción de control tarda en tener efecto sobre el proceso que se está controlando.
- Complica el análisis y diseño de los sistemas de control realimentados.
- Los tiempos de retardo o tiempos muertos de un sistema siempre reducen la estabilidad del mismo, y además limita la respuesta temporal que puede conseguirse del sistema.

---

## Introducción

---

Por tanto, la existencia de retardos, especialmente cuando son del mismo orden que el tiempo de respuesta del sistema, puede dar lugar a la existencia de bucles cerrados de control que resultan ser inestables, impidiendo la controlabilidad del proceso; esta falta de controlabilidad suele presentarse en la práctica con una reducción de la máxima ganancia utilizable en el controlador para actuar sobre el proceso. Los retardos pueden afectar a la entrada, la salida o el estado interno del proceso.

Por todo ello el estudio de los sistemas y los mecanismos de control de los mismos cuando hay retardos presentes, es un área de estudio muy activo por el interés que presenta al resultar un elemento clave en los sistemas reales industriales.

### 1.2. Definición de Sistema Continuo con Retardo

Es bien sabido que un sistema dinámico se puede caracterizar mediante ecuaciones diferenciales ordinarias (EDO):

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

Donde  $x(t) \in \mathbb{R}^n$  representa las variables de estado del proceso. Por tanto, mediante el modelo en ecuaciones diferenciales del proceso, se puede determinar la evolución futura de las variables de estado a partir de cierto instante  $t_0$  a partir del valor actual del estado  $x(t_0)$ .

No obstante, el comportamiento de ciertos sistemas dinámicos no puede ser modelado a partir de EDO [Hale-03]. Concretamente, los sistemas con retardo no pueden ser modelados con EDO, pues la evolución futura de las variables de estado no solamente depende del valor actual del estado  $x(t_0)$ , sino de los valores pasados dentro de una cierta ventana temporal:

$$x(\vartheta), t_0 - \tau \leq \vartheta \leq t_0, \tau > 0$$

Para modelar formalmente un sistema dinámico con retardos, es necesario recurrir a las ecuaciones diferenciales funcionales (EDF). Estas se diferencian básicamente de las ecuaciones diferenciales ordinarias en que al menos uno de los argumentos de la ecuación no depende directamente del tiempo  $t$ , sino que lo hace a través de una cierta función  $g(t)$  de modo que:

$$\dot{x}(t) = f(t, x(g(t)))$$

El comportamiento de los sistemas con retardos temporales se puede describir por tanto a partir de EDF. Es fácil ver que siempre que el retardo sea constante, la función  $g(t)$  es lineal en  $t$  con pendiente 1. Así, para un

## Introducción

---

cierto retardo de un sistema de valor  $\tau$ , la dinámica del mismo viene gobernada por:

$$\dot{x}(t) = f(t, x(t-\tau))$$

Donde la condición inicial del estado  $x(t)$  en cualquier instante de tiempo arbitrario se define a partir de la evolución pasada del estado en la ventana temporal  $t-\tau \leq t < t$  como:

$$\begin{aligned} x(t+\mathcal{G}) &= \phi(\mathcal{G}) \\ -\tau &\leq \mathcal{G} \leq 0 \end{aligned}$$

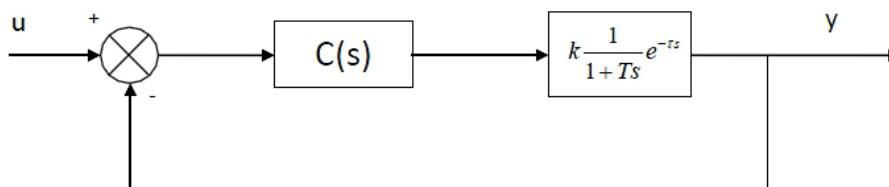
A partir de esta condición se puede obtener unívocamente la evolución futura del estado desde el instante  $t$ , siempre que  $\phi(\mathcal{G})$  sea continua en todos sus argumentos y Lipschitz, de forma análoga a las EDO con las condiciones iniciales definidas en un instante puntual. Existe una taxonomía de EDF, así como una teoría matemática para las EDF paralela a las EDO, que el lector interesado puede consultar en [Hale-93].

### 1.3. Sistemas FOTD

Los sistemas FOTD son aquellos sistemas de primer orden que presentan un retardo (o tiempo muerto) en su respuesta temporal, debido como ya se ha indicado a los tiempos muertos debidos al transporte de masa o energía (calor, por ejemplo); la denominación FOTD proviene de sus siglas en inglés *First Order with Time Delay*. Un sistema FOTD como el representado en la figura 1 tiene la siguiente función de transferencia  $G(s)$ :

$$G(s) = k \frac{1}{1+Ts} e^{-\tau s}$$

Eligiendo adecuadamente los parámetros  $T$  y  $\tau$ , los sistemas FOTD pueden representar con suficiente exactitud multitud de procesos tanto físicos como industriales e incluso sociales o financieros [Rico-07].



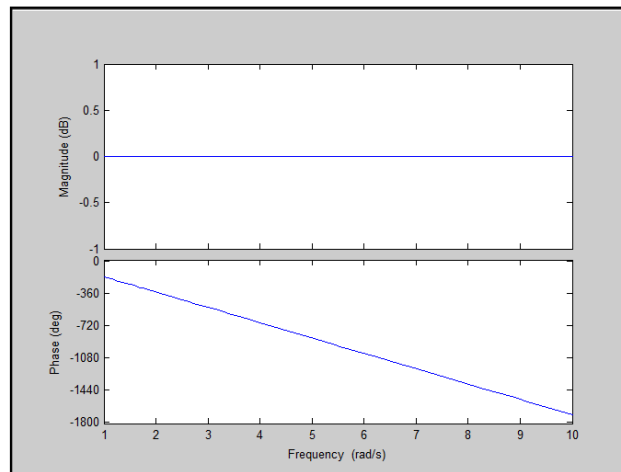
**Figura 1:** *Bucle de control con un sistema FOTD*

## Introducción

Como se puede ver en la expresión de  $G(s)$ , y haciendo  $s = j\omega$  el término que representa el retardo es:

$$e^{-j\tau\omega} = \cos(\tau\omega) - j\text{sen}(\tau\omega)$$

De donde se deduce que  $|e^{-j\tau\omega}| = 1$  y que  $\angle(e^{-j\tau\omega}) = -\omega\tau$  y por lo tanto la gráfica del diagrama de Bode correspondiente es la que se muestra en la figura 2 para  $\tau = 3$ .



**Figura 2:** *Diagrama de Bode del Retardo*

El efecto que el retardo tiene sobre una función de transferencia dada  $G(s)$  tal que  $G(s) = G_R(s) e^{-\tau s}$  donde  $G_R(s)$  representa la parte racional de  $G(s)$ , es entonces:

$$\begin{aligned} |G(s)| &= |G_R(s)| \\ \angle G(s) &= \angle G_R(s) - \tau\omega \end{aligned}$$

En definitiva, el retardo en este caso:

- No modifica el módulo de  $G_R(s)$ .
- Añade un retardo de fase a  $G_R(s)$  proporcional a  $\omega$ , y contribuye por lo tanto a reducir el margen de fase.

El retardo analizado hasta ahora, y en el que se centrará este trabajo, es en el retraso a la entrada o la salida, y que es el que hace aparecer el término ya comentado  $e^{-\tau s}$ ; existen otro tipo de retardos más complejos de analizar, como por ejemplo el que presenta el siguiente sistema (estable):

$$G(s) = \frac{1 - e^{-2\pi s}}{2\pi s}$$

# Introducción

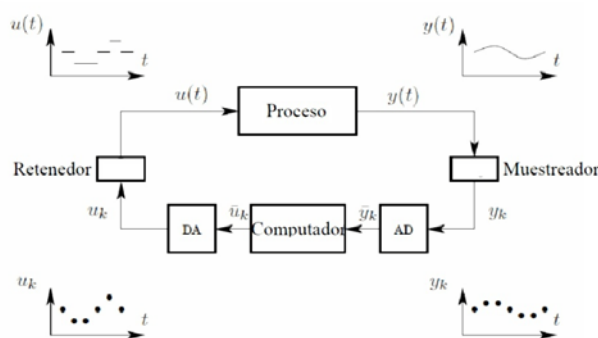
## 1.4. Muestreo de Sistemas

Dado que actualmente el control sobre un sistema con o sin retardo se realiza mediante computador, es imprescindible tener en cuenta el hecho diferencial que supone el uso del mismo, que es precisamente la discretización del proceso: la señal continua es convertida en una secuencia de números que son procesados por el computador digital. Este da una nueva secuencia de números, que posteriormente son convertidos a una señal continua y aplicada al proceso [Wittenmark-02] [Bernhardsson-93].

Este segundo proceso se denomina reconstrucción de la señal; la figura 3 muestra los diferentes elementos que intervienen en este proceso de discretización, donde el computador implementa el código del controlador.

En implementaciones prácticas de control, el sensor (que mide alguna magnitud física relevante desde el punto de vista de control a la salida de la planta) y el controlador (que suele ser un computador digital) ejecutan sus tareas en instantes puntuales de tiempo junto con el actuador (que ejecuta las acciones de control calculadas por el controlador), ya que se implementan con dispositivos digitales.

De este modo, la actualización de la acción de control y la medida de la salida del proceso se realiza en ciertos instantes puntuales  $t_{ak}$  y  $t_{sk}$ ,  $k=1,2,\dots$  respectivamente. Bajo este escenario, se define el periodo de muestreo como la diferencia entre dos instantes de tiempo consecutivos. Por ejemplo, el periodo de muestreo del sensor es  $T_{sk} = t_{sk} - t_{sk-1}$ . Por simplicidad, los periodos de muestreo se asumen constantes:  $T_{sk} = T_s$ ,  $T_{ak} = T_a$ .



**Figura 3:** Sistema de Control Clásico

## 1.5. Muestreo No Convencional

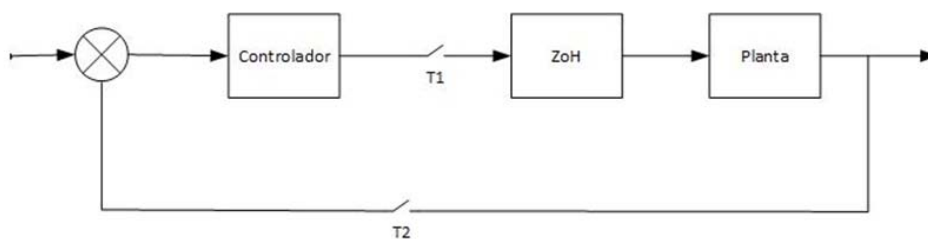
Un sistema muestreado no convencionalmente, es aquel en el que el muestreo de las señales no sigue un patrón periódico. Como caso particular,

## Introducción

---

se tiene el muestreo multifrecuencia, en el que dos o más variables son muestreadas a frecuencias distintas [Vélez-04a]. El muestreo no convencional presenta propiedades interesantes de cara a los sistemas de control, incluidos aquellos que presentan retardos, como se verá en capítulos posteriores.

Una aplicación práctica habitual en la industria es el muestreo bifrecuencia, en el que, dado un sistema realimentado, la salida del sistema se somete a un muestreo (o actualización) lento  $T_2$ , y la entrada al mismo o entrada de control que proviene del regulador, a un muestreo (o actualización) rápido  $T_1$ , tal y como puede verse en la figura 4.



**Figura 4:** *Esquema General de un Sistema Bifrecuencia*

El esquema mostrado en la figura 4, y que es el que se usará a lo largo de este trabajo es el denominado MRIC (*Multi-Rate Input Controller*), al ser la actualización de la señal de control cada  $T_1$  instantes de tiempo, mientras que el muestreo de la salida de la planta se realiza cada  $T_2 = N \times T_1$  instantes de tiempo.

Aunque el sistema mostrado en la figura es de tipo SISO, los sistemas multifrecuencia pueden ser también multivariantes o MIMO. Desde el punto de vista de modelado matemático, los sistemas multifrecuencia son un caso particular de sistemas periódicos, en los que el patrón de muestreo se repite periódicamente, siendo este periodo el mínimo común múltiplo (mcm) de los periodos de muestreo de todas las señales del sistema.

El control multifrecuencia presenta ciertas propiedades que lo hacen interesante al momento de diseñar reguladores digitales que tengan en cuenta las distintas dinámicas de los elementos del sistema. Aunque es muchas veces viable muestrear un sistema a la mayor frecuencia posible, esto en otras ocasiones puede traer problemas numéricos, de resolución y de capacidad de memoria del ordenador o sistema que efectúa el control.

Pero lo realmente interesante del muestreo multifrecuencia, es que la introducción bien estudiada de esquemas multifrecuencia puede mejorar el comportamiento de un sistema en comparación con la aplicación de

## Introducción

---

patrones de muestreo monofrecuencia, siendo especialmente útil en sistemas multivariables donde los sensores y actuadores de los diferentes lazos de control tienen diferentes tiempos de respuesta y anchos de banda; en [Araki-93] se da una detallada explicación de las ventajas directas del control multifrecuencia.

Por último, es preciso resaltar que el control multifrecuencia ha sido utilizado en diversas aplicaciones: robots, discos duros, aviónica, columnas de destilación, reactores, procesadores, detección de fallos, industria del poliéster, submarinos, sistemas eléctricos, gas, sistemas neumáticos, sistemas hidráulicos y motores.

### 1.6. Sistemas de Control Basados en Red

Bajo el término de Sistemas de Control Basados en Red (SCBR o NCS en su acrónimo inglés: *Networked Control Systems*) se entiende un tipo específico de sistemas de control, que se caracterizan por la utilización de un medio de comunicación de datos (en este caso las muestras de los sensores y las acciones de control) compartido para la transferencia de información entre el controlador y la planta controlada. Esto significa que, a diferencia de un sistema de control discreto convencional, en este tipo de sistemas el mismo medio de comunicación es empleado para establecer varios bucles de control e incluso para otro tipo de tareas no relacionadas con el control. El hecho de utilizar un enlace de uso no exclusivo para cerrar el bucle de realimentación presenta una serie de características y de limitaciones que dificultan y limitan el control que se puede efectuar sobre el sistema (se verá que es una nueva fuente de retardo), aunque también presentan algunas ventajas que justifican su amplia utilización y estudio en profundidad: reducción de cableado, facilidad de monitorización y mantenimiento en general barato [Sala-09].

Obviamente, una manera de actuar para evitar estas limitaciones en el control del sistema, consiste en actuar para mejorar las prestaciones del medio de comunicación compartido, aumentando con ello el ancho de banda disponible, y por lo tanto reduciendo su impacto negativo en los objetivos de control que se desean alcanzar.

Por otra parte, se puede asumir (como en este trabajo), que el medio de comunicación existe e impone limitaciones que no es posible evitar, y tratar de reducir o eliminar su influencia mediante el diseño de estructuras de control apropiadas que sean capaces de contrarrestar el efecto negativo del medio de comunicación.

La característica típica de este tipo de sistemas, es que presentan retardos en la transmisión de la información, retardos que en muchos casos

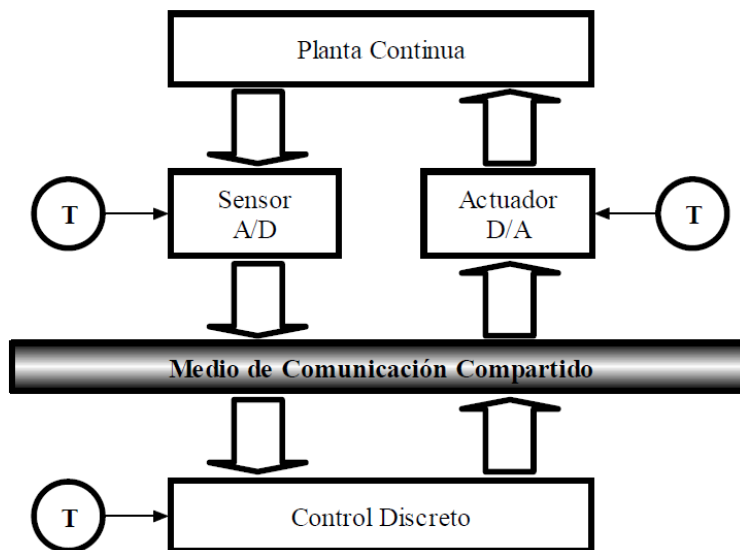
## Introducción

---

son aleatorios y difíciles de prever, dada la naturaleza estocástica del tráfico de información que circula por la red; más aún: en muchos casos, no solo se producen retardos en la transmisión de la información, sino que ésta en todo o en parte (cuando se emite en paquetes), se puede perder, al ser un medio compartido y no estar necesariamente disponible cuando el control lo necesita. En definitiva, el hecho de introducir una red de comunicaciones en el sistema, es otra fuente de retardo.

Otro problema característico de este tipo de sistemas, es la falta de sincronismo entre los instantes en que se muestrea la variable a controlar, y la salida del controlador. En un sistema de control convencional el muestreo de la variable de interés, el cálculo de las acciones de control y la aplicación de las mismas sobre el proceso controlado coinciden en el tiempo, cosa que debido a que el medio no tiene porqué estar disponible cuando los requisitos de transmisión del sistema lo exigen, no se da en los sistemas basados en red.

Algunos de los efectos negativos que introduce la red de comunicaciones a través de la que se intercambian los datos y en general la información en el sistema de control se describe en detalle en [Zhang-01]. La figura 5 muestra un esquema general de este tipo de sistemas con sus elementos más significativos: sensor, actuador, controlador (remoto), planta y red de comunicaciones.



**Figura 5:** *Esquema General de un Sistema de Control Basado en Red*

En la citada figura puede verse como entre el sistema de control y la planta, sensores y actuadores, está el medio de transmisión compartido. En este caso, se supone que todo el sistema está muestreado a frecuencia  $1/T$ .

## 1.7. Objetivos del Trabajo

Los objetivos del presente trabajo se resumen en el estudio de la estabilidad de sistemas con retardos, en el análisis de la discretización de los mismos, en la aplicación de técnicas de muestreo no convencional a este tipo de sistemas para la mejora de su controlabilidad, y en el análisis y compensación de los retardos introducidos por las redes de comunicaciones, cuando estas son el medio de transmisión de los datos que en formato digital, se usan para controlar los sistemas. En particular:

- Se analizarán las herramientas matemáticas disponible para tratar los sistemas con retardos, haciendo particular énfasis en el Teorema de Hermite-Biehler, y su aplicación práctica a sistemas de primer orden con retardos, para calcular el espacio de parámetros de los reguladores asociados a dichos sistemas que los hacen estables.
- Se estudiará la discretización de los sistemas de primer orden (y órdenes superiores) con retardos, y se implementará un algoritmo que realice dicha discretización, y simule la respuesta del sistema en bucle cerrado.
- Se analizarán los sistemas con muestreo no convencional, y en particular se analizará el efecto del muestreo bifrecuencia en sistemas FOTD, estudiando si este tipo de muestreo mejora la controlabilidad de los sistemas con retardo, en función de las especificaciones dinámicas de los mismos.
- Por último, se analizará el efecto de la inclusión de una red de comunicaciones industrial (CAN) como medio de transmisión entre los diferentes elementos que intervienen en un sistema de control en bucle cerrado, como dicha red es una fuente de retardos, y un mecanismo de compensación de los mismos para mejorar el comportamiento dinámico del sistema.

## 1.8. Estructura de este trabajo

El presente trabajo está estructurado en 5 capítulos, siendo el primero de ellos esta introducción. El resto de los capítulos son los siguientes:

- Capítulo 2: **Estabilidad de los Sistemas FOTD**: en este capítulo se describen diversas herramientas matemáticas para tratar los términos de retardo que aparecen en la función de transferencia de un sistema que los presente, se enuncia el Teorema de Hermite-Biehler y se muestra como este último puede ser una herramienta poderosa para determinar la estabilidad de un sistema, así como el espacio de parámetros del regulador que estabilizan un sistema con retardo en bucle cerrado, en base a trabajos basados en el citado teorema [Silva-02] y [Silva-05].

## Introducción

---

- Capítulo 3: **Muestreo de Sistemas con Retardo:** En este capítulo se analiza en detalle el muestreo de sistemas con retardo de cualquier orden, y se desarrolla un algoritmo en MATLAB para calcular la ecuación de estados de un sistema con retardo discretizado basado en [Aström-97] y [Wittenmark-85]. Se simula también con MATLAB el comportamiento del sistema frente a escalones y perturbaciones en la entrada del mismo.
- Capítulo 4: **Sistemas con Muestreo Multifrecuencia:** En este capítulo se estudian los sistemas con Muestreo No Convencional, y en particular el muestreo bifrecuencia aplicado a sistemas de primer y segundo orden con retardos, analizando en base a simulaciones realizadas con MATLAB las mejoras que el muestreo bifrecuencia aporta a la controlabilidad de sistemas con retardo. La herramienta MCT '*Multirate Control Toolbox*' utilizada para ello está fundamentada en [Salt-00], [Velez-02], [Vélez-03] y [Velez-04a].
- Capítulo 5: **Sistemas de Control Basados en Red:** en este capítulo se analizan las características específicas de un sistema de control en el que sus elementos intercambian datos mediante una red digital de comunicaciones mediante simulaciones con MATLAB, así como sus efectos sobre un sistema FOTD, y un posible mecanismo sencillo de compensación de dichos efectos, todo ello con la herramienta *TrueTime* para MATLAB [Henriksson-03] y [Cervin-03].

## Estabilidad de Sistemas FOTD

---

### 2.1. Resumen

En sabido que en el análisis de estabilidad de un sistema continuo es necesario que todas las raíces de su polinomio característico asociado se encuentren en  $\mathbb{C}^-$ ; así el problema de determinar las condiciones bajo las cuales todas las raíces reales de un polinomio característico se encuentren en el semiplano izquierdo complejo, juega un papel importante en el análisis de la estabilidad de sistemas. Un polinomio que posee esta propiedad se denomina polinomio de Hurwitz.

En el estudio de la distribución de las raíces de un polinomio sobre el plano complejo, uno de los primeros problemas fue, históricamente, el de determinar el número de raíces reales del citado polinomio; esto es, dado un polinomio con coeficientes reales, determinar únicamente en base a sus coeficientes, y sin calcular analíticamente sus raíces, si este tiene raíces reales, y cuantas raíces positivas y negativas tiene.

Diversos criterios se han propuesto para alcanzar el objetivo expuesto, siendo algunos muy conocidos, como los de Routh y Hurwitz. En el caso de sistemas discretos, para tener estabilidad se necesita que el polinomio característico asociado tenga todas sus raíces dentro del círculo unitario: estos polinomios reciben el nombre de polinomios de Schur.

El análisis del número de raíces reales de un polinomio (sin resolver la ecuación correspondiente), y la determinación de cuantas tienen su parte real positiva y cuantas su parte real negativa, se planteo ya en 1868 por Maxwell, si bien otros matemáticos (como Hermite) ya habían abordado el problema de la localización de las raíces de un polinomio sin resolver analíticamente el problema, aunque siempre dentro del mundo de las matemáticas. La contribución más importante de Maxwell fue demostrar que el comportamiento de un sistema de control automático en la proximidad de una posición de equilibrio, se podía aproximar por una ecuación diferencial lineal y por lo tanto la estabilidad se podía así discutir en términos de las raíces de la ecuación algebraica asociada.

Finalmente fueron Routh y Hurwitz quienes encontraron un método que se ha hecho muy popular (aunque no sea el único): Routh resuelve este

problema en 1877 [Routh-77]; en su trabajo presenta por primera vez su conocido criterio de estabilidad; Hurwitz en 1885, de una forma independiente y utilizando las técnicas de Cauchy y Hermite, resuelve el mismo problema en términos de un conjunto de determinantes. Fue Bompiani quien demostraría en 1911 la equivalencia de los criterios de Routh y Hurwitz [Bompiani 11].

En este capítulo se presenta de forma resumida un método para determinar la estabilidad y el espacio de parámetros que hacen estable un sistema continuo en bucle cerrado basado en el Teorema de Hermite-Biehler, aplicado a sistemas continuos FOTD:

$$G_R(s) = k \frac{1}{1+Ts} e^{-\tau s}$$

Que como se verá más adelante, se caracterizan por tener un número infinito de raíces en su polinomio característico, lo que añade una dificultad al cálculo de sus raíces, y por tanto a la determinación de la estabilidad del sistema.

Este tipo de polinomios se denominan cuasi-polinomios, y el Teorema de Hermite-Biehler (también denominado Teorema de la alternancia) permite abordar el estudio de su estabilidad.

### 2.2. Estabilidad de Sistemas FOTD

Si el sistema FOTD  $G_R(s)$  antes presentado no tuviera el término de retardo  $e^{-\tau s}$  (es decir  $\tau=0$ ), y siendo la fdt del controlador  $C(s)$ , la ecuación característica del mismo sería:

$$1 + C(s)G_R(s) = 0$$

Que tiene un número finito de raíces, y por lo tanto, también tendrá un número finito de polos la función de transferencia en bucle cerrado  $M(s)$ :

$$M(s) = \frac{C(s)G_R(s)}{1 + C(s)G_R(s)}$$

Ahora bien, si  $\tau \neq 0$ , entonces la ecuación característica es:

$$1 + C(s)G_R(s) e^{-\tau s} = 0$$

Nótese que a diferencia de los sistemas convencionales, la ecuación característica es trascendente con infinitas raíces, lo que implica que el número de polos de  $M(s)$  también es infinito.

Si se tiene un sistema realimentado como los definidos en el capítulo anterior y se considera que la función de transferencia del controlador es  $C(s)$ , asumiendo que este es un controlador de tipo PID, entonces:

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

Donde  $k_p$  es la constante del término proporcional,  $k_i$  la del término integral y  $k_d$  la del término diferencial. Si  $\tau = 0$ , entonces la ecuación característica del sistema es la siguiente:

$$(T + k k_d) s^2 + (1 + k k_p) s + k k_i = 0$$

El sistema es estable si todos los coeficientes tienen el mismo signo:

$$k_p > -\frac{1}{k}, k_i > 0, k_d > -\frac{T}{k}$$
$$k_p < -\frac{1}{k}, k_i < 0, k_d < -\frac{T}{k}$$

Si ahora se considera que  $\tau \neq 0$ , la ecuación característica es la que se muestra a continuación:

$$(k k_i + k k_p s + k k_d s^2) e^{-\tau s} + (1 + T s) s = 0$$

Que como se ha indicado presenta infinitos ceros, circunstancia que dificulta el análisis de la misma. Para resolver esta dificultad, existen varias herramientas matemáticas, las que se presentan aquí son las siguientes:

- Aproximaciones Racionales
- El predictor de Smith
- Teorema de Hermite-Biehler ampliado

### 2.3. Tipos de Aproximaciones Racionales

La idea general en la que se basan las aproximaciones racionales del término de retardo  $e^{-\tau s}$ , es el truncado de una serie de potencias, que suelen dar un resultado aceptable en un entorno suficientemente amplio de  $s = 0$ .

#### 2.3.1. Aproximación de Naïve

La aproximación de Naïve [Klerk-02], se basa en la igualdad siguiente:

$$e^{-s} = \frac{e^{-\frac{s}{2}}}{e^{\frac{s}{2}}}$$

Desarrollando en series de Taylor numerador y denominador, y truncando, se obtiene:

$$e^{-s} = \frac{\sum_{i=0}^n \frac{1}{i!} (-s)^i}{\sum_{i=0}^n \frac{1}{i!} s^i}$$

El desarrollo da los siguientes resultados:

$$n = 1 \rightarrow e^{-\tau s} \approx \frac{1 - \frac{\tau s}{2}}{1 + \frac{\tau s}{2}}, \quad n = 2 \rightarrow e^{-\tau s} = \frac{1 - \frac{\tau s}{2} - \frac{s^2 \tau^2}{8}}{1 + \frac{\tau s}{2} + \frac{s^2 \tau^2}{8}}$$

La última expresión para  $n=2$  es la denominada fórmula de Kautz.

Continuando el desarrollo indicado, puede demostrarse que las aproximaciones obtenidas son inestables para  $n > 4$ .

### 2.3.2. Aproximación de Padé

La aproximación de Padé, muy popular, se basa en la igualdad siguiente [Marshall-92]:

$$e^{-s} = \frac{P_m(s)}{Q_n(s)} = R_{m,n}(s)$$

En donde  $P_m(s)$  y  $Q_n(s)$  son polinomios en  $s$  de grados  $m$  y  $n$  respectivamente. El desarrollo de Taylor en  $s=0$  da la siguiente expresión:

$$e^{-s} = 1 - \frac{s}{1!} + \frac{s^2}{2!} - \frac{s^3}{3!} + \dots$$

$$R_{m,n}(s) = R_{m,n}(0) + \frac{R'_{m,n}(0)s}{1!} + \frac{R''_{m,n}(0)s^2}{2!} + \frac{R'''_{m,n}(0)s^3}{3!} + \dots$$

La aproximación de Padé [Silva-05], se basa en encontrar los coeficientes de  $R_{m,n}(s)$  haciéndolos coincidir con los primeros  $n+m+1$  coeficientes de Taylor de estas dos series. Se puede demostrar que si  $n = m$ , entonces  $P_n(s) = -Q_n(-s)$ . Operando:

$$e^{-s} = \frac{\sum_{i=0}^n \binom{n}{i} \frac{(2n-i)!}{(2n)!} (-s)^i}{\sum_{i=0}^n \binom{n}{i} \frac{(2n-i)!}{(2n)!} s^i}$$

Particularizando la expresión anterior para los dos primeros valores de  $n$  se obtiene:

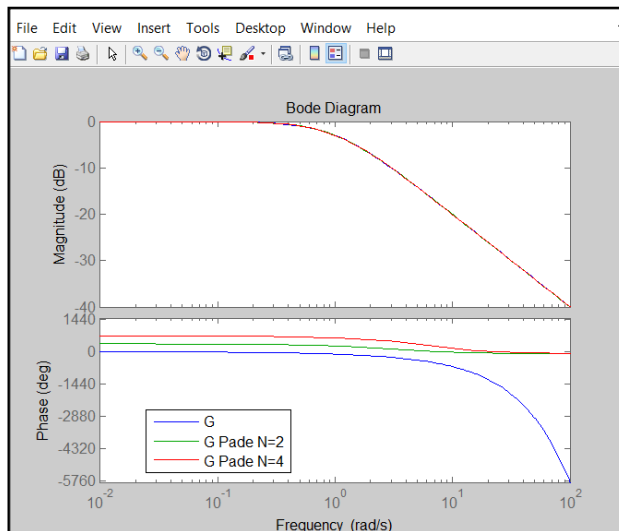
$$n=1 \rightarrow \frac{1 - \frac{\tau s}{2}}{1 + \frac{\tau s}{2}} \quad n=2 \rightarrow \frac{1 - \frac{\tau s}{2} - \frac{\tau^2 s^2}{12}}{1 + \frac{\tau s}{2} + \frac{\tau^2 s^2}{12}}$$

A diferencia de lo que sucedía en el caso de la aproximación de Naïve, la aproximación de Padé es estable para todo valor de  $n$ .

La figura 6 muestra el diagrama de Bode de la función de transferencia:

$$M(s) = \frac{1}{1+s} e^{-s}$$

Junto con las aproximaciones del retardo de Padé para  $n=2$  y  $n=4$ . En general, el aumento del orden de la aproximación hace cada vez más precisa la misma (excepto para  $e^{-s}$ ), aunque también aumenta mucho la complejidad de los cálculos a realizar, por lo que habitualmente no se usan aproximaciones de Padé de orden superior al segundo, salvo que los cálculos se hagan por ordenador.



**Figura 6:** Diagrama de Bode de  $M(s)$

Las aproximaciones anteriores, aportan ceros en el semiplano positivo, lo que implica que los sistemas resultantes son sistemas de fase no mínima; por tanto, los sistemas con retardo tienen un comportamiento de fase no mínima.

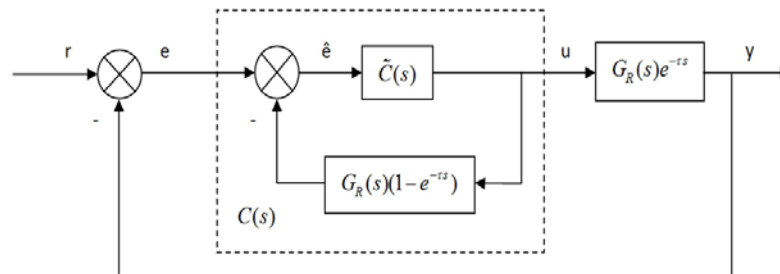
### 2.4. El Predictor de Smith

El predictor de Smith fue introducido hace más de medio siglo por Otto J. Smith [Brosilow-79]; es una de las estructuras de control avanzado más populares de todas las que tratan de solucionar los problemas de los sistemas realimentados con retardos. Aunque por la fecha en la que fue introducido se concibió para sistemas continuos, es más apropiado para sistemas discretos (digitales).

La idea básica que subyace detrás de esta estructura de control, es el adelanto del sensor, usando la discrepancia entre la salida de la planta y un "modelo rápido" retrasado para corregir la estimación de la salida.

El esquema de dicho predictor es el que se muestra en la figura 7. En dicho esquema,  $\tilde{C}(s)$  es el controlador primario, y  $G_R(s)(1-e^{-\tau s})$  es el predictor de Smith propiamente dicho. Nótese que:

$$\hat{e} = (r - y) - G_R(s)(1 - e^{-\tau s})u = r - G_R(s)u$$



**Figura 7: Predictor de Smith**

Y que  $r - G_R(s)u$  es la "predicción" del error  $e$  cuando se cumple que el retardo es nulo. Operando sobre el esquema de la figura 4 se tiene:

$$u = \tilde{C} [(r - G_R(s)e^{-\tau s}u) - G_R(1 - e^{-\tau s})u] = \tilde{C}(r - G_R(s)u)$$

De donde las funciones de transferencia desde  $r$  a  $u$  e  $y$ :

$$G_u(s) = \frac{\tilde{C}(s)}{1 + G_R(s)\tilde{C}(s)}; \quad G_y(s) = \frac{G_R(s)\tilde{C}(s)}{1 + G_R(s)\tilde{C}(s)} e^{-\tau s}$$

## Estabilidad de Sistemas FOTD

---

Obsérvese que el término de retardo  $e^{-\tau s}$  queda eliminado de la ecuación característica:

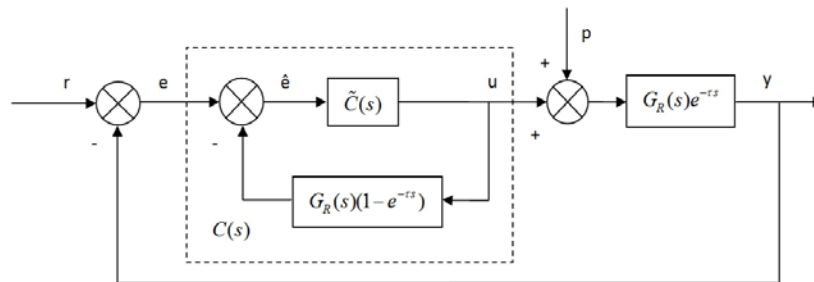
$$1 + G_R(s)\tilde{C}(s) = 0$$

Quedando pues la estabilización del sistema con retardo  $(G_R(s)e^{-\tau s}, C(s))$  reducida a la estabilización del sistema sin retardo  $(G_R(s), \tilde{C}(s))$ .

El método es sencillo y elegante, pero presenta inconvenientes; para analizarlos, veamos la situación representada en la figura 8, en la que se añade al sistema una perturbación  $p$ :

Calculando la función de transferencia desde  $p$  hasta  $y$  se tiene que:

$$\begin{aligned} u &= \tilde{C} [(r - G_R(s)e^{-\tau s}u - G_R(s)e^{-\tau s}p) - G_R(s)(1 - e^{-\tau s})u] = \\ &= \tilde{C} (r - G_R(s)u - G_R(s)e^{-\tau s}p) \end{aligned}$$



**Figura 8:** *Predicador de Smith con Perturbación*

Por lo tanto:

$$G_p(s) = \frac{G_R(s)}{1 + G_R(s)\tilde{C}(s)} [1 + \tilde{C}(s)G_R(s)(1 - e^{-\tau s})]e^{-\tau s}$$

Ahora bien:  $G_p(s)$  puede ser inestable si  $G_R(s)$  también lo es. En definitiva: el método del predicador de Smith puede usarse en sistemas o plantas estables, pero no es útil si el sistema es inestable.

Tampoco es útil si no se conoce adecuadamente el modelo del sistema, o si se estima incorrectamente el retardo.

### 2.5. El Teorema de Hermite-Biehler en la Teoría de Control

Como ya se ha descrito anteriormente, el análisis de la estabilidad de sistemas libres de retardo es relativamente sencillo debido a que el número

de raíces de la ecuación característica es finito, sin embargo cuando el mismo sistema presenta retardos, el número de raíces de la ecuación característica no es finito, y el análisis de estabilidad se complica. El teorema de Hermite-Biehler constituye una poderosa herramienta para llevar a cabo el análisis de estabilidad de sistemas con retardos temporales, existiendo este teorema para el caso continuo, el discreto y para cuasi-polinomios, siendo este último el que se presenta en los sistemas con retardo.

El teorema de Hermite-Biehler (también conocido como Teorema de la alternancia) es un teorema que da un criterio geométrico para determinar si un polinomio tiene sus raíces en el semiplano izquierdo del conjunto  $\mathbb{C}^-$ .

En este trabajo se describirá el teorema de Hermite-Biehler para los tres casos citados, si bien no se demostrará, ya que existen múltiples referencias bibliográficas para ello, siendo quizás una de las más interesantes [Silva-05], a la que se remite al lector interesado en ellas.

Recuérdese que el análisis de estabilidad de un sistema, se basa en el estudio de la posición de las raíces de la ecuación característica, de modo que dichas raíces estén todas ellas en el semiplano izquierdo del conjunto  $\mathbb{C}^-$  de los números complejos (caso continuo, polinomio Hurwitz), o en el conjunto de los números complejos con módulo menor que la unidad (caso discreto, polinomio Schur).

### 2.5.1. El Teorema de Hermite-Biehler para el caso continuo Sea el polinomio de grado $n$ :

$$\delta(s) = a_0 + a_1s + a_2s^2 + \dots + a_ns^n$$

Que puede escribirse como:

$$\delta(s) = \delta_p(s^2) + s\delta_i(s^2)$$

$$\delta(j\omega) = p(\omega) + jq(\omega)$$

Siendo  $\delta_p(s^2)$ ,  $s\delta_i(s^2)$  las componentes de potencias pares e impares respectivamente de  $\delta(s)$ , y siendo  $p(\omega) = \delta_p(-\omega^2)$  y  $q(\omega) = \omega\delta_i(-\omega^2)$ . Sean por otra parte  $\omega_{p1}, \omega_{p2}, \dots; \omega_{i1}, \omega_{i2}, \dots$  los ceros reales no negativos de  $\delta_p(-\omega^2)$  y  $\delta_i(-\omega^2)$  respectivamente, dispuestos en orden ascendente de módulo; el Teorema de Hermite-Biehler afirma que se verifica que  $\delta(s)$  es Hurwitz si y solo si:

- Todos los ceros de  $\delta_p(-\omega^2)$  y  $\delta_i(-\omega^2)$  son reales y distintos.
- Los coeficientes de  $\delta_p(s^2)$ ,  $s\delta_i(s^2)$  tienen el mismo signo.
- Se cumple la propiedad de la alternancia, esto es:

$$\omega_{p1} < \omega_{i1} < \omega_{p2} < \omega_{i2} < \dots$$

### 2.5.2. El Teorema de Hermite-Biehler para el caso discreto

Sea el polinomio de grado  $n$  en  $z$ :

$$P(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0$$

Que puede escribirse en función de  $P_s(z)$  y  $P_a(z)$  como se indica a continuación:

$$P(z) = P_s(z) + P_a(z)$$

Siendo:

$$P_s(z) = \frac{1}{2} \left[ P(z) + z^n P\left(\frac{1}{z}\right) \right]$$

$$P_a(z) = \frac{1}{2} \left[ P(z) - z^n P\left(\frac{1}{z}\right) \right]$$

Entonces el Teorema de Hermite-Biehler afirma que se verifica que  $P(z)$  es Schur si y solo si:

- $P_s(z)$  y  $P_a(z)$  son polinomios de grado  $n$  con coeficientes del mismo signo.
- $P_s(z)$  y  $P_a(z)$  tienen solo ceros simples en el círculo unidad.
- Los ceros de  $P_s(z)$  y  $P_a(z)$  se alternan en el círculo unidad.

### 2.5.3. El Teorema de Hermite-Biehler para sistemas con retardo

En los párrafos anteriores se ha mostrado el teorema de Hermite-Biehler para sistemas continuos y discretos sin retardos; sin embargo, cuando el sistema tiene retardos, no se pueden aplicar de manera directa los resultados vistos en ellos. Los sistemas con retardos dan lugar a ecuaciones características con términos trascendentes denominados cuasi-polinomios. Cuando los retardos son conmensurables (múltiplos enteros de un número positivo  $\tau$ ), la ecuación característica, como ya se ha visto es de la forma:

$$P(s) = a_0(s) + a_1(s)e^{-\tau s} + a_2(s)e^{-2\tau s} + \dots + a_k(s)e^{-k\tau s}$$

Definiendo previamente:

$$\sum_k \sum_i a_{ik} s^i t^k = P(s, t)$$

Donde  $r$  es el mayor grado en  $s$  y  $p$  el mayor grado en  $t$ , se dice que  $P(s, t)$  tiene un término principal si y solo si  $a_{rp} \neq 0$ ; entonces se tiene que un polinomio  $F(s) = P(s, e^s)$ , donde  $P(s, t)$  tiene un término principal, y está escrito de la forma:

$$F(j\omega) = F_r(\omega) + jF_i(\omega)$$

Con  $F_r(\omega)$  y  $jF_i(\omega)$  representando respectivamente la parte real e imaginaria de  $F(j\omega)$ , el Teorema de Hermite-Biehler afirma que  $F(j\omega)$  tiene todos sus ceros en el semiplano izquierdo complejo si y solo si:

- $F_r(\omega)$  y  $F_i(\omega)$  tienen solo raíces reales simples y estas se alternan.
- $\forall \omega \in \mathbb{R}, F_i'(\omega)F_r(\omega) - F_i(\omega)F_r'(\omega) > 0$ .
- La segunda condición puede sustituirse por la siguiente (menos restrictiva):

$$\exists \omega_0 \in \mathbb{R} / F_i'(\omega_0)F_r(\omega_0) - F_i(\omega_0)F_r'(\omega_0) > 0.$$

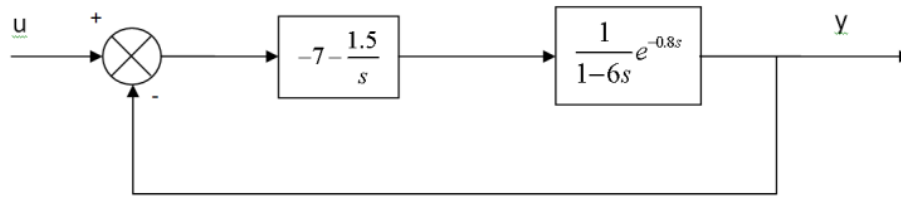
### 2.6. Análisis de Estabilidad

Existe una descripción detallada [Silva-02] de la metodología para analizar la estabilidad de un sistema con un regulador PID en el espacio de parámetros  $\{k_p, k_i, k_d\}$ ; en este trabajo, se implementará la metodología descrita en [Silva-05] en MATLAB para hallar el lugar geométrico de los parámetros  $\{k_p, k_i\}$  de un regulador PI que hacen que un sistema FOTD sea estable.

El teorema de Hermit-Biehler permite, como aplicación más inmediata, determinar si un sistema es estable, como se muestra en el ejemplo siguiente:

Dada la planta (inestable):  $G(s) = \frac{1}{1-6s} e^{-0.8s}$ , se analiza la estabilidad con el siguiente regulador:

$$C(s) = -7 - \frac{1.5}{s}$$



**Figura 9:** *Análisis de Estabilidad de un Sistema*

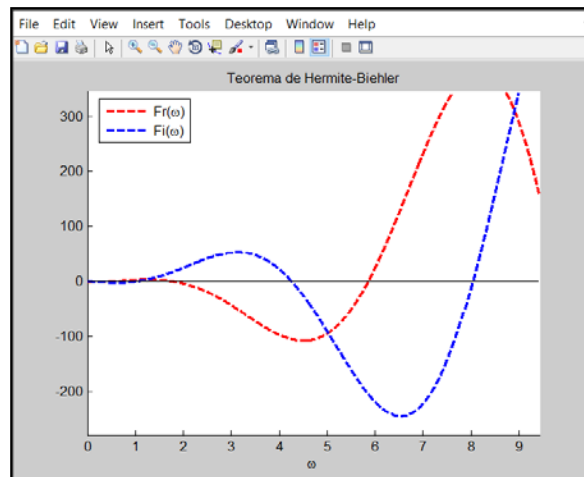
El sistema realimentado resultante del esquema de la figura 9 es:

$$M(s) = -\frac{1.5s + 7}{(1 - 6s)se^{0.8s} - 7s - 1.5}$$

Operando según dicta el teorema de Hermite-Biehler, se tienen las siguientes expresiones para la parte real e imaginaria respectivamente:

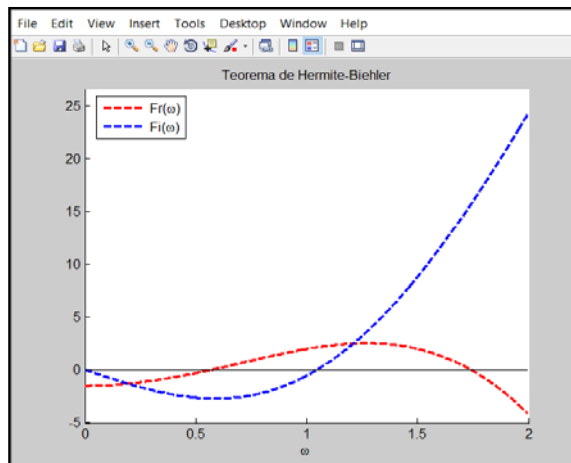
$$F_r(j\omega) = kk_i - \omega \text{sen}(\tau\omega) - T\omega^2 \cos(\tau\omega) = -1.5 - \omega \text{sen}\left(\frac{4}{5}\omega\right) + 6\omega^2 \cos\left(\frac{4}{5}\omega\right)$$

$$F_i(j\omega) = \omega[kk_p \cos(\tau\omega) - T\omega \text{sen}(\tau\omega)] = -7\omega \cos\left(\frac{4}{5}\omega\right) + 6\omega^2 \text{sen}\left(\frac{4}{5}\omega\right)$$



**Figura 10:** *Representación de  $F_r(\omega)$  y  $F_i(\omega)$*

Representando ambas curvas con MATLAB obtenemos como resultado el mostrado en la figura 10 y 11 (esta última es un detalle de la anterior). El 'script' usado para la representación de las curvas mostradas en las figuras 10 y 11 se muestra en la figura 12.

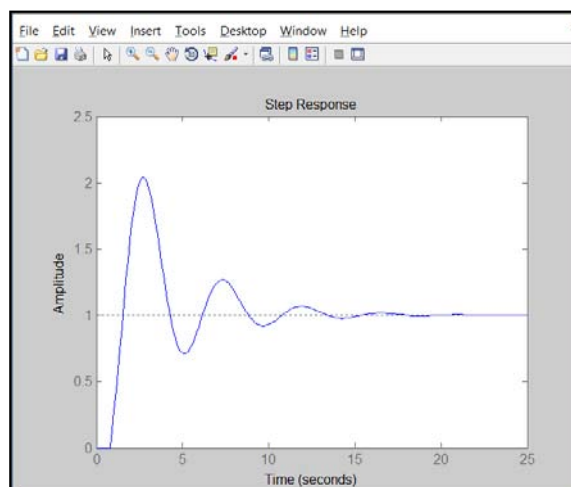


**Figura 11:** Representación de  $F_r(\omega)$  y  $F_i(\omega)$  (Detalle)

```
function dibuja()
    syms w;
    Xlim = 2;
    hold on
    p1 = ezplot('-1.5-w*sin((4/5)*w)+6*w^2*cos((4/5)*w)', [0, Xlim]);
    p2 = ezplot('-7*w*cos((4/5)*w)+6*w^2*sin((4/5)*w)', [0, Xlim]);
    p3 = ezplot('0', [0, Xlim]);
    set(p1,'Color','red','LineStyle','--','LineWidth',2);
    set(p2,'Color','blue','LineStyle','--','LineWidth',2);
    set(p3,'Color','black');
    title('Teorema de Hermite-Biehler');
    legend([p1 p2], 'Fr(\omega)', 'Fi(\omega)', 'Location', 'Best');
    xlabel('\omega');
    hold off
end
```

**Figura 12:** 'Script' MATLAB para representar curvas de estabilidad

Vemos que efectivamente, de acuerdo con la respuesta del sistema a un escalón mostrada en la figura 13, el sistema es estable.



**Figura 13:** Respuesta al escalón del sistema original

### 2.7. Regulador PI

La teoría de control asienta sus bases esencialmente en el concepto de realimentación; este concepto se concreta en una estructura de control en la cual el regulador se puede entender como un controlador, que en función de la salida de referencia de la planta, y la salida real medida, proporciona la acción de control a aplicar sobre el sistema.

De los diversos tipos de reguladores clásicos de los que se dispone (P, PD, PI, PID), los reguladores PI son los más ampliamente utilizados en la mayoría de procesos [Aström-88], procesos que además, generalmente pueden modelarse mediante un sistema de primer orden con o sin retardo [Barraud-05]. Por ejemplo, en procesos industriales químicos, se necesita casi siempre incluir un término de retardo para su correcto modelado.

Existen razones que desaconsejan el uso de la acción proporcional en sistemas con retardo, aparte de la ya conocida de la excesiva sensibilidad al ruido debido a la alta ganancia que presenta a frecuencias elevadas (que puede mitigarse filtrando el ruido del sistema); la segunda razón, importante para los procesos estudiados en este trabajo, es que en sistemas que presentan retardos elevados, la acción anticipativa del término derivativo deja de ser efectiva, ya que la aproximación lineal correspondiente al término derivativo:

$$e(t+T_d) = e(t) + T_d \frac{de(t)}{dt}$$

Tan sólo tiene validez para pequeños valores de  $T_d$ : debido a los tiempos muertos hay un retardo antes de que los efectos de cualquier acción de control se puedan detectar sobre la variable de interés del proceso que se está controlando.

Por este motivo, es mucho más efectivo en caso de sistemas con retardo (tanto más cuanto mayor sea el término de retardo que presenten), predecir su acción futura analizando la señal de control en combinación con un modelo previamente establecido del proceso que se está controlando.

Esto es lo que hace en realidad el predictor de Smith descrito en el punto 2.4, así como otras estrategias de control, como la MPC que se describirá brevemente en el punto 5.8.2.

El enfoque basado en el uso de un predictor, es aplicable al control de procesos fuertemente no lineales y/o que no pueden ser aproximados eficientemente por algún modelo lineal, como es el caso descrito de sistemas con un término de retardo relativamente grande, y existen numerosos trabajos basados en este concepto para su aplicación a sistemas de control automático.

## Estabilidad de Sistemas FOTD

---

De ahí que en este trabajo se utilicen fundamentalmente los reguladores de tipo PI, ya que van a estar asociados al control de procesos con retardos que pueden ser arbitrariamente grandes. En este apartado, se utilizará el Teorema de Hermite-Biehler para obtener los parámetros de los reguladores PI de un sistema de primer orden con retardo (FOTD) como el mostrado en la figura 9, con la ayuda de un ejemplo.

Recordemos que la fdt de la planta tipo que se va a analizar es la correspondiente a un FOTD, que necesita tan solo tres parámetros para ser completamente definida ( $k$ ,  $T$ ,  $\tau$ ):

$$G(s) = k \frac{1}{1+Ts} e^{-\tau s}$$

Siendo  $k$  la ganancia estática de la planta,  $T$  la constante de tiempo y  $\tau$  el retardo. La fdt del regulador es:

$$C(s) = k_p + \frac{k_i}{s}$$

Siendo  $k_p$  la ganancia proporcional, y  $k_i$  la ganancia integral. El objetivo de este epígrafe es encontrar el lugar geométrico de los parámetros del regulador  $\{k_p, k_i\}$  que hacen que la planta sea estable también en bucle cerrado.

Cuando  $\tau = 0$ , ya se ha visto que se cumple que la ecuación característica del sistema es:

$$F(s) = Ts^2 + (kk_p + 1)s + kk_i$$

Del análisis de la misma se deduce, para un sistema sin retardos, que este será estable en bucle cerrado si se cumple en bucle abierto que:

$$kk_i > 0, \quad kk_p + 1 > 0, \quad T > 0 \quad (\text{Sistemas estables})$$

$$kk_i < 0, \quad kk_p + 1 < 0, \quad T < 0 \quad (\text{Sistemas inestables})$$

Si como es habitual, la ganancia estática de la planta  $k$  es positiva, las condiciones anteriores para bucle cerrado se pueden reducir a las siguientes:

$$k_p > -\frac{1}{k}, \quad k_i > 0$$

$$k_p < -\frac{1}{k}, \quad k_i < 0$$

## Estabilidad de Sistemas FOTD

---

Analizando ahora la situación en la que existe retardo, esto es:  $\tau > 0$ , la ecuación característica de sistema a bucle cerrado es la siguiente:

$$F(s) = (kk_i + kk_p)e^{-\tau s} + (1+Ts)s$$

Aplicando 2.4.3, se halla el cuasi-polinomio del sistema:

$$F^*(s) = e^{\tau s}F(s) = kk_i + kk_p s + (1+Ts)s e^{\tau s}$$

Haciendo  $s = j\omega$  se puede escribir:

$$F^*(j\omega) = F_r(j\omega) + jF_i(j\omega)$$

En donde identificando términos:

$$F_r(j\omega) = kk_i - \omega \operatorname{sen}(\tau\omega) - T\omega^2 \cos(\tau\omega)$$

$$F_i(j\omega) = \omega[kk_p \cos(\tau\omega) - T\omega \operatorname{sen}(\tau\omega)]$$

Si el sistema es inestable en bucle abierto ( $T < 0$ ), aplicando las condiciones vistas más arriba, se debe de cumplir si  $\tau = 0$ :

$$k_p < -\frac{1}{k}, k_i < 0$$

De acuerdo con [Silva-05], la condición necesaria para que un controlador PI estabilice simultáneamente el sistema sin retardo ( $\tau = 0$ ) y el sistema con retardo ( $\tau > 0$ ) cuando la fdt del sistema es inestable en bucle abierto, es la siguiente:

$$\left| \frac{T}{\tau} \right| > 1$$

Si esta condición se satisface, entonces el conjunto de valores de  $k_p$  para el sistema FOTD de partida debe de cumplir para que el sistema sea estable [Silva-05]:

$$\frac{T}{k\tau} \sqrt{\alpha_1 + \frac{\tau^2}{T^2}} < k_p < -\frac{1}{k}$$

Siendo  $\alpha_1$  la solución de la ecuación:  $\operatorname{tg}(\alpha_1) = -\frac{T}{\tau}\alpha_1$

En el intervalo abierto  $(0, \frac{\pi}{2})$  para sistemas inestable, o  $(\frac{\pi}{2}, \pi)$  para sistemas inestables. Esta expresión, permite reducir el conjunto de valores  $k_p, k_i$  sobre los que hacer que se verifique 2.5.3.

### Ejemplo:

Considérese el sistema FOTD:

$$G(s) = \frac{1}{1+4s} e^{-s}$$

Si se analiza el lugar geométrico de parámetros que hacen que el sistema siga siendo estable se tiene que:

$$\left| \frac{T}{\tau} \right| = \left| \frac{4}{1} \right| = 4 > 1$$

Por lo que existe el lugar geométrico pedido. Los pasos del algoritmo a calcular son los siguientes:

- Paso 1: Se hace  $k_p = -\frac{1}{k}$ ,  $pasos = \frac{1}{N+1} \left( \frac{T}{k\tau} \sqrt{\alpha_1^2 + \frac{\tau^2}{T^2}} + \frac{1}{k} \right)$  y  $p = 1$ ;
- Paso 2: Incrementar  $k_p$  como se indica:  $k_p = k_p + pasos$ ;
- Paso 3: Si se cumple que  $k_p < \frac{T}{k\tau} \sqrt{\alpha_1^2 + \frac{\tau^2}{T^2}}$  ir al paso 4; si no es así, el algoritmo ha concluido;
- Paso 4: Calcular las raíces  $z_p$  de la ecuación:

$$kk_p + \cos(z) - \frac{T}{\tau} z \operatorname{sen}(z) = 0$$

- Paso 5: Calcular el valor de  $a_p$  para cada  $z_p$  anterior con la ecuación:

$$a(z) = \frac{z}{k\tau} \left[ \operatorname{sen}(z) + \frac{T}{\tau} z \cos(z) \right]$$

- Paso 6: Si  $\cos(z_p) > 0$  ir al paso 7. Si no, hacer  $p = p + 2$  e ir al paso 4;
- Paso 7: Determinar las cotas inferior y superior de  $k_i$  con la expresión:

$$0 < k_i < \min_{l=1,3,5,\dots,p} \{a_p\}$$

- Paso 8: Volver al paso 2.

La implementación de los 8 pasos antes indicados, da lugar al algoritmo cuyo código en MATLAB se muestra en la figura 14.

## Estabilidad de Sistemas FOTD

En dicha figura puede observarse que se hace uso de la '*Symbolic Toolbox*' de MATLAB, así como de la resolución numérica de ecuaciones con variable simbólica mediante la función *vpasolve()*.

La función entrega una matriz de dos columnas, una para los valores de  $k_p$  y otra para los valores de  $k_i$  que definen el lugar geométrico que hace el sistema estable. También se usa la función *gzero()*, que resuelve la ecuación:

$$\operatorname{tg}(\alpha_1) = -\frac{T}{\tau}\alpha_1$$

El código de dicha función se muestra en la figura 15. La gráfica ( $\Omega$ ) obtenida con los '*scripts*' mencionados, puede verse en la figura 16.

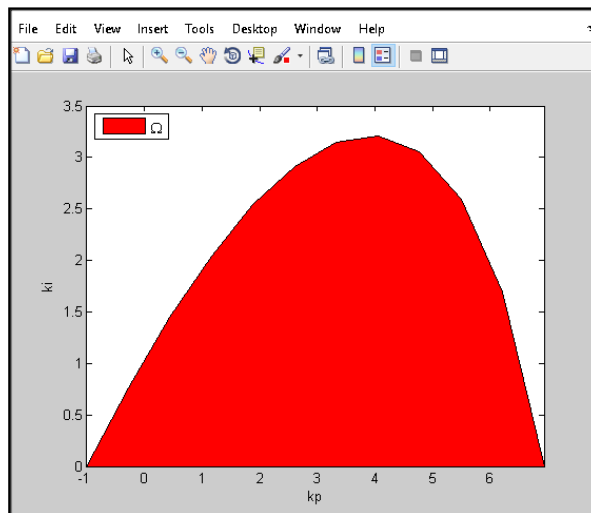
Para comprobar de una manera sencilla los resultados que da la gráfica, se tomará un par estable y otro inestable de parámetros  $\{k_p, k_i\}$ , y se simulará la respuesta del sistema en bucle cerrado a un escalón unitario con ambos pares de parámetros.

```
function [y] = hbpi(K, T, tau)
syms x;
N = 10.0; delta = 0.01; r = 0; y = [];
if T > 0
    if tau > 0
        kp = -1/K;
        alfa1 = gzero(T, tau);
        kmax = (T/(K*tau))*sqrt(alfa1^2 + (tau/T)^2);
        paso = (1/(N + 1))*(kmax + (1/K));
        while kp < kmax + delta
            p = 1; q = 1; a = []; z = [];
            while q > 0
                z(p) = vpasolve(K*kp+cos(x)-(T/tau)*x*sin(x) == 0, x, [(p-1)*pi p*pi]);
                a(p) = (z(p)/(K*tau))*(sin(z(p)))+(T/tau)*z(p)*cos(z(p));
                if cos(z(p)) > 0
                    ki = min(a(1:2:length(a)));
                    if ki > 0
                        y = [y; kp, ki];
                    elseif ki <= 0 && (kp > kmax - delta || kp == -1/K)
                        y = [y; kp, 0];
                    else
                        % No se hace nada;
                    end
                    q = 0;
                else
                    p = p + 2;
                end
            end
            kp = kp + paso;
        end
    else
        if abs(T/tau) <= 1
            disp('El sistema es inestable');
        else
            disp('El sistema esta libre de retardo');
        end
        y = [0 0];
    end
end
y;
```

**Figura 14:** '*Script*' para el cálculo del conjunto  $\{k_p, k_i\}$  estable

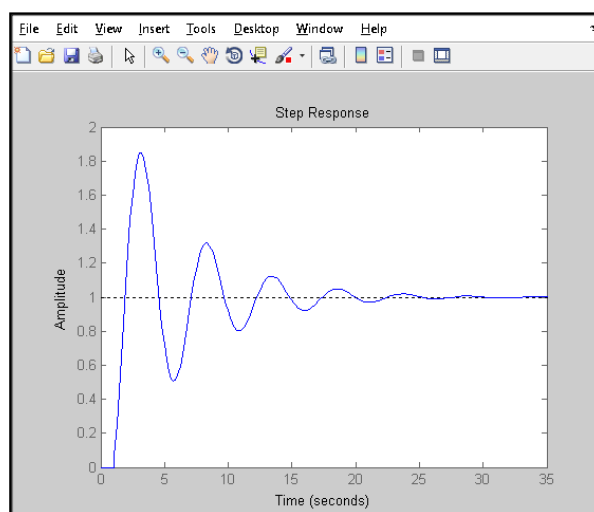
```
function y = gzero(T, tau)
syms x;
if T > 0
    y = double(vpasolve(tan(x) + (T/tau)*x == 0, x, [pi/2 pi]));
elseif T < 0
    y = double(vpasolve(tan(x) + (T/tau)*x == 0, x, [0.01 pi/2]));
else
    y = 0;
end
end
```

**Figura 15:** *'Script' de la función gzero()*



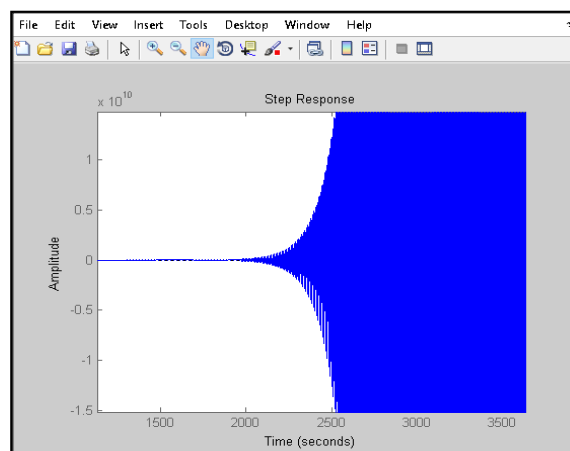
**Figura 16:** *Lugar geométrico de pares  $\{k_p, k_i\}$  estables de  $G(s)$*

Se puede ahora simular con MATLAB la respuesta al escalón con el par estable  $\{k_p = 4, k_i = 2\}$ , que se muestra en la figura 17:



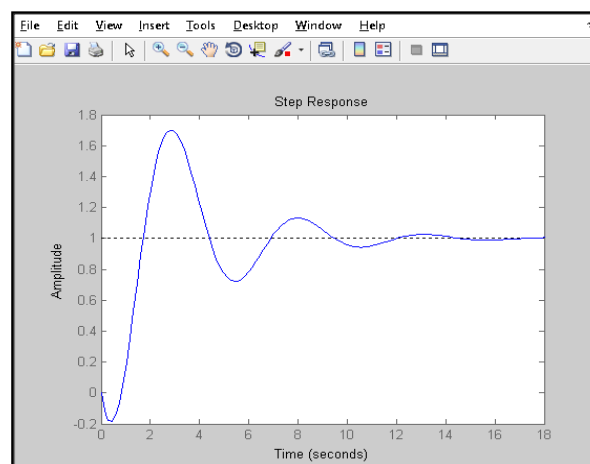
**Figura 17:** *Respuesta al escalón de  $G(s)$  con el par  $\{k_p=4, k_i=2\}$*

Y simular la respuesta al escalón con el par inestable  $\{k_p = 1, k_i = 2\}$ , según se muestra en la figura 18. Claramente se comprueba que el par  $\{k_p = 4, k_i = 2\}$  da lugar a un sistema estable, mientras que el par  $\{k_p = 1, k_i = 2\}$  da lugar a un sistema que es inestable, de acuerdo con los resultados obtenidos por el algoritmo de la figura 14 basado en el Teorema de Hermite-Biehler desarrollado en este capítulo. Es interesante comparar la respuesta del sistema original con el par estable  $\{k_p = 4, k_i = 2\}$  correspondiente a la figura 17, con la que dan las aproximaciones de Padé.



**Figura 18:** *Respuesta al escalón de  $G(s)$  con el par  $\{k_p=1, k_i=2\}$*

Para una aproximación de Padé de orden 1, la respuesta del sistema al escalón puede verse en la figura 19.



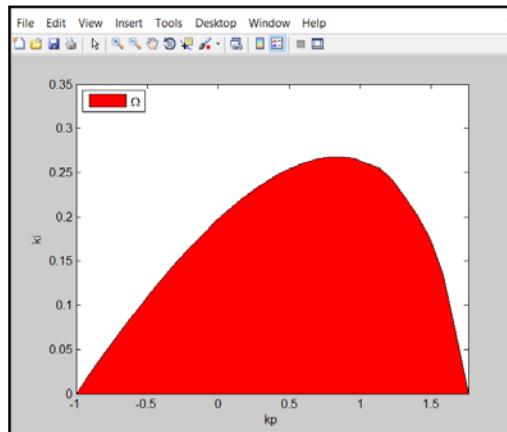
**Figura 19:** *Respuesta al escalón de  $G(s)$   $\{k_p=4, k_i=2\}$  (Padé, orden 1)*

## Estabilidad de Sistemas FOTD

Podemos preguntarnos, a la vista de dicha respuesta, razonablemente aproximada, qué aporta la metodología de Hermite-Biehler sobre la aproximación de Padé. Para responder a esta pregunta, veamos un ejemplo extraído de [Silva-05], con la planta representada por la fdt:

$$H(s) = \frac{1}{1+4s} e^{-6s}$$

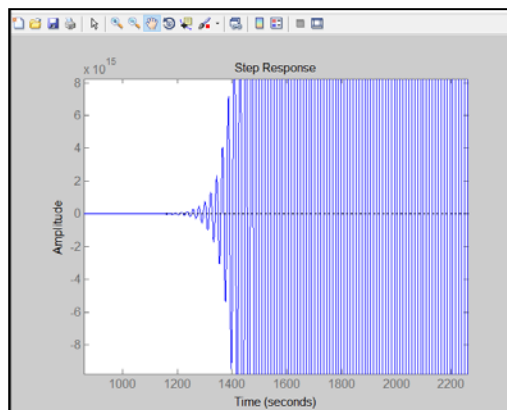
Que es evidentemente estable. Veamos ahora cual es el área de estabilidad de los pares  $\{k_p, k_i\}$  que mantienen la estabilidad del sistema en bucle cerrado obtenida con el algoritmo de la figura 14; dicho área puede verse en la figura 20.



**Figura 20:** Lugar geométrico de pares  $\{k_p, k_i\}$  estables de  $H(s)$

Si se toma ahora un par  $\{k_p, k_i\}$  cercano a la frontera de la zona estable, pero en el área no coloreada en rojo (inestable), como por ejemplo:

$$\{k_p, k_i\} = \{1.5, 0.25\}$$



**Figura 21:** Respuesta del sistema  $H(s)$  con el par  $\{1.5, 0.25\}$

## Estabilidad de Sistemas FOTD

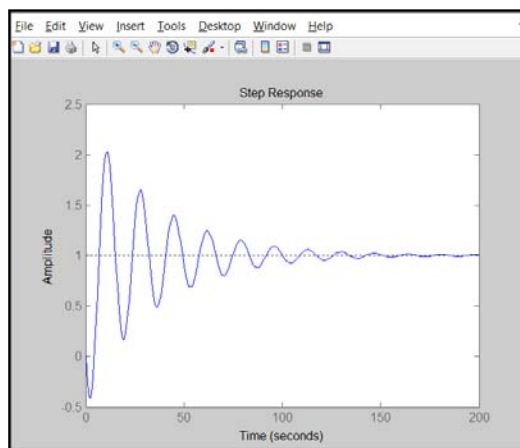
---

Se tiene la respuesta del sistema en bucle cerrado mostrada en la figura 21, que tal y como predice el algoritmo y era por tanto de esperar, es claramente inestable.

Sin embargo, si ahora se hace una aproximación de  $G(s)$  con Padé de orden 1, se tiene la respuesta al escalón del sistema de la figura 22.

Se ve que dicha respuesta al escalón con el mismo regulador y con el retardo aproximado por Padé de orden 1, resulta ser estable, lo cual es claramente erróneo. Puede comprobarse que haciendo el mismo ejercicio aproximando con Padé de orden 2, la respuesta resulta ser inestable.

Queda comprobado por tanto que la metodología aquí descrita es mucho más exacta que otras aproximaciones algebraicas, como la de Padé.



**Figura 22:** *Respuesta de  $H(s)$  con Padé (1) y el par  $\{1.5, 0.25\}$*



---

## Muestreo de Sistemas con Retardo

---

### 3.1. Introducción

De la Teoría de control, es sabido que en el caso de sistemas LTI sin retardo, si se parte de la ecuación de estado del sistema:

$$\begin{aligned}\frac{dx}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Entonces según [Aström-97] se tiene que:

$$\begin{aligned}x(t_{k+1}) &= \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k) \\ y(t_k) &= Cx(t_k) + Du(t_k)\end{aligned}$$

Con

$$\begin{aligned}\Phi(t_{k+1}, t_k) &= e^{A(t_{k+1}, t_k)} \\ \Gamma(t_{k+1}, t_k) &= \int_0^{t_{k+1}-t_k} e^{As} ds B\end{aligned}$$

En este capítulo se analizará la discretización de sistemas con retardo, y se implementará un 'script' de MATLAB para calcular las ecuaciones de estado de un sistema muestreado cuando aquel presenta retardo, analizando como es la respuesta de un sistema FOTD (y posteriormente de un sistema de 2º orden) en función de los valores de dicho retardo.

Se verá que el retardo introduce estados adicionales en la ecuación de estado del sistema, y que el aumento del retardo tiende a hacer el sistema cada vez más oscilatorio, hasta que se vuelve completamente inestable.

### 3.2. Muestreo de Sistemas con Retardo

El muestreo de sistemas con retardo está descrito en [Aström-97] y [Wittenmark-85] por ejemplo; aquí se mostrarán los resultados de dichos estudios.

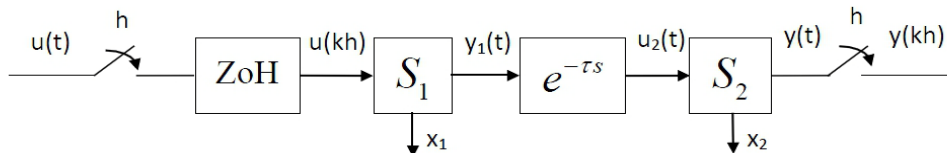
Para ello, se plantea el problema de manera general, es decir: dados dos sistemas:

## Muestreo de Sistemas con Retardo

$$S_1 \begin{cases} \dot{x}_1(t) = A_1 x_1(t) + B_1 u_1(t) \\ y_1(t) = C_1 x_1(t) + D_1 u_1(t) \end{cases}$$

$$S_2 \begin{cases} \dot{x}_2(t) = A_2 x_2(t) + B_2 u_2(t) \\ y_2(t) = C_2 x_2(t) + D_2 u_2(t) \end{cases}$$

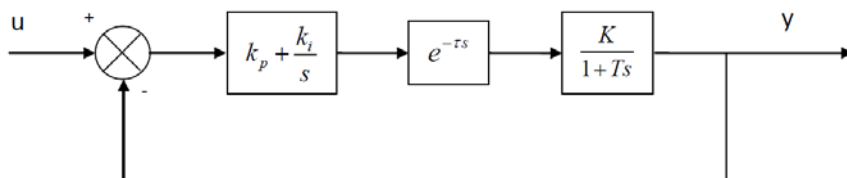
Cuyo orden es respectivamente  $n_1$  y  $n_2$  dispuestos según se indica en la figura 23, donde el periodo de muestreo es  $h$ .



**Figura 23:** *Sistemas con retardo temporal*

Se trata de calcular la ecuación de estado discreta del sistema conjunto; para ello se particulariza el esquema de la figura 23, obteniendo el sistema mostrado en la figura 24, en el que se usa un regulador PI, y donde:

$$S_1 = k_p + \frac{k_i}{s}, \quad S_2 = \frac{K}{1+Ts}$$



**Figura 24:** *Sistema FOTD con retardo temporal interno*

Las ecuaciones de estado del regulador PI son siguientes [Dominguez-06]:

$$S_1 = \begin{cases} \dot{x}_1(t) = u(t) \\ y_1(t) = k_i x_1(t) + k_p u(t) \end{cases}$$

Y por lo tanto se cumple que:  $[A]=0$ ,  $[B]=1$ ,  $[C]=k_i$ ,  $[D]=k_p$ . El sistema se puede modelar mediante una ecuación diferencial de primer orden, de

## Muestreo de Sistemas con Retardo

---

manera que solo hay una única variable de estado  $x_1(t)$ . Para el sistema de primer orden (y por tanto también con una sola variable de estado  $x_2(t)$ ) se tiene que:

$$S_2 \begin{cases} \dot{x}_2(t) = -\frac{1}{T}x_2(t) + \frac{K}{T}u_2(t) \\ y_2(t) = x_2(t) \end{cases}$$

Y en este caso:  $[A] = -\frac{1}{T}$ ,  $[B] = \frac{K}{T}$ ,  $[C] = 1$ ,  $[D] = 0$ .

Además:

$$u_2(t) = y_1(t - \tau)$$

Una vez obtenida la ecuación de estado del sistema en bucle abierto mostrado en la figura 24, puede obtenerse la fdt de dicho sistema.

Para encontrar la ecuación de estado discreta, se puede considerar que el controlador PI no sufre retardo alguno; en este supuesto y según [Aström-97], el sistema muestreado se modela con la expresión:

$$S1 \begin{cases} x_1(kh + h) = \Phi x_1(kh) + \Gamma u_1(kh) \\ y_1(kh) = C_1 x_1(kh) + D_1 u_1(kh) \end{cases}$$

Donde:

$$\Phi = e^{Ah} = 1$$

$$\Gamma = \int_0^h e^{As} ds = h$$

Para el controlador PI (sistema S1) se tiene:

$$S1 \begin{cases} x_1(kh + h) = x_1(kh) + hu_1(kh) \\ y_1(kh) = k_i x_1(kh) + k_p u_1(kh) \end{cases}$$

Y por lo tanto se tiene en este caso que:

$$[A_1] = 1, [B_1] = h, [C_1] = k_i, [D_1] = k_p.$$

A continuación se calcula la expresión del sistema muestreado de primer orden S2, que sufre un retardo  $\tau$  a su entrada; se supondrá además que este retardo es tal que  $\tau > h$ , siendo:

$$\tau = (d - 1)h + \tau'$$

## Muestreo de Sistemas con Retardo

---

De donde:  $h - \tau' = dh - \tau$

Según [Aström-97] y [Wittenmark-85], la expresión del sistema S2 muestreado es la siguiente:

$$x_2(kh + h) = \Phi x_2(kh) + \Gamma_0 u(kh - (d-1)h) + \Gamma_1 u(kh - dh)$$

Con:

$$\begin{aligned}\Phi &= e^{A_2 h} \\ \Gamma_0 &= \int_0^{h-\tau'} B_2 e^{A_2 s} ds \\ \Gamma_1 &= e^{A_2 (h-\tau')} \int_0^{\tau'} B_2 e^{A_2 s} ds\end{aligned}$$

### 3.3. Ejemplo práctico para un sistema FODT

A continuación se va a llevar a cabo una aplicación práctica del análisis anterior: como sistema FODT de trabajo, se va a tomar en este caso el siguiente:

$$G(s) = \frac{0.25}{24s + 1} e^{-\tau s}$$

Que según [Campo-12] representa con bastante exactitud la variación de presión en una columna de destilación en función de la actuación de una válvula de control de vapor. Aunque las medidas experimentales para la columna en cuestión daban para el retardo un valor:

$$\tau \approx 3$$

En este trabajo se asumirán arbitrariamente otros valores del retardo con el objeto de analizar la estabilidad de un sistema FODT dado con un retardo mayor que el periodo de muestreo  $h$ .

Del análisis de la respuesta al escalón de un sistema genérico de primer orden, se sabe que el tiempo de establecimiento  $t_s$  es de  $3T$ , siendo  $T$  la constante de tiempo del sistema. Por otra parte, como periodo de muestreo, se elige aquel que cumpla [Aström-97]:

$$N_r = \frac{t_r}{h}$$

Con  $N_r = 10$ , y siendo  $t_r$  el tiempo de subida.

## Muestreo de Sistemas con Retardo

---

El tiempo de subida se puede definir de varias formas; en este trabajo se asume que el tiempo de subida es el que transcurre desde que la salida del sistema pasa de un 5% de a un 95% de su valor final ante una respuesta en escalón, y se asumirá también que:

$$t_r \approx t_s$$

Y por lo tanto:

$$h = \frac{t_s}{10} = 0.1 \cdot t_s = 0.1 \cdot 3 \cdot T \approx 0.33 \cdot T = \frac{T}{3}$$

Calculando valores, se tiene:

$$\begin{aligned} \Phi &= e^{Ah} = e^{\frac{h}{T}} \\ \Gamma_0 &= \int_0^{h-\tau'} B \cdot e^{As} ds = \frac{K}{T} \int_0^{h-\tau'} e^{\frac{1}{T}s} ds = -K \cdot [e^{\frac{3\tau-dT}{3T}} + 1] \\ \Gamma_1 &= e^{A(h-\tau')} \int_0^{\tau'} B \cdot e^{As} ds = -K \cdot e^{\frac{\tau-dh}{T}} [e^{\frac{(d-1)h-\tau}{T}} - 1] \end{aligned}$$

En el sistema FODT definido, se usan los siguientes valores para los parámetros:  $h = 8$ ,  $\tau = 10$ ,  $d = 2$ ,  $\tau' = 2$ . Sustituyendo estos valores en las expresiones anteriores, se obtiene:

$$\begin{aligned} \Phi &= e^{\frac{h}{T}} = e^{\frac{1}{3}} = 0.716 \\ \Gamma_0 &= -K \cdot [e^{\frac{3\tau-dT}{3T}} - 1] = -0.25(e^{-0.25} - 1) = 0.015 \\ \Gamma_1 &= -K \cdot e^{\frac{\tau-dh}{T}} [e^{\frac{(d-1)h-\tau}{T}} - 1] = -0.25 e^{-0.25} [e^{-0.083} - 1] = 0.055 \end{aligned}$$

De manera que la ecuación de estado extendida para S2 es la siguiente:

$$\begin{pmatrix} x(8k+8) \\ u(8k-8) \\ u(8k) \end{pmatrix} = \begin{pmatrix} 0.716 & 0.015 & 0.055 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x(8k) \\ u(8k-16) \\ u(8k-8) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(8k)$$

$$y(8k+8) = x(8k)$$

Para completar las ecuaciones de estado del regulador PI, se realiza el ajuste en bucle cerrado del sistema realimentado de la figura 24 con los valores indicados más arriba suponiendo que el sistema de primer orden tiene un retardo:

$$\tau = 0$$

## Muestreo de Sistemas con Retardo

Dicho ajusta arroja los valores para  $k_p$  y  $k_i$  calculados con la función `pidtune()` de MATLAB siguientes:

$$k_p = 1.8928$$

$$k_i = 0.2674$$

Por lo que la ecuación de estado del regulador PI (sistema S1) es:

$$x_1(8k+8) = x_1(8k) + 8 u_1(8k)$$

$$y_1(8k) = 0.2674 x_1(8k) + 1.8928 u_1(8k)$$

Una vez que se dispone de la expresión muestreada del controlador (sistema S1) y del sistema FODT (sistema S2), y teniendo en cuenta la composición de sistemas en serie definidos por sus ecuaciones de estado:

$$\dot{x} = \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix} x + \begin{pmatrix} B_1 \\ B_2 D_1 \end{pmatrix} u$$

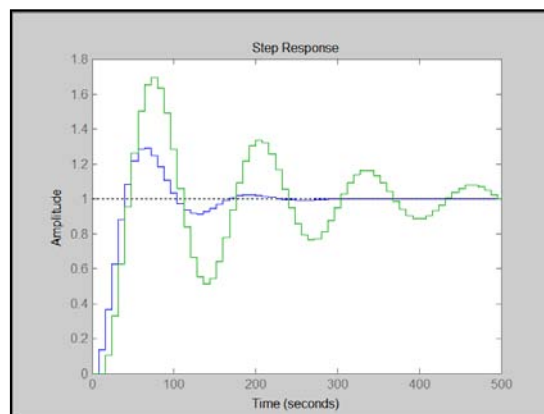
$$y = (D_2 C_1 \quad C_2) x + (D_2 \quad D_1) u$$

Y por lo tanto puede escribirse:

$$\begin{pmatrix} x_1(8k+8) \\ x_2(8k+8) \\ u(8k-8) \\ u(8k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.716 & 0.015 & 0.055 \\ 0 & 0 & 0 & 1 \\ 0.2674 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(8k) \\ x_2(8k) \\ u(8k-16) \\ u(8k-8) \end{pmatrix} + \begin{pmatrix} 8 \\ 0 \\ 0 \\ 1.8928 \end{pmatrix} u(8k)$$

$$y(8k+8) = x_2(8k)$$

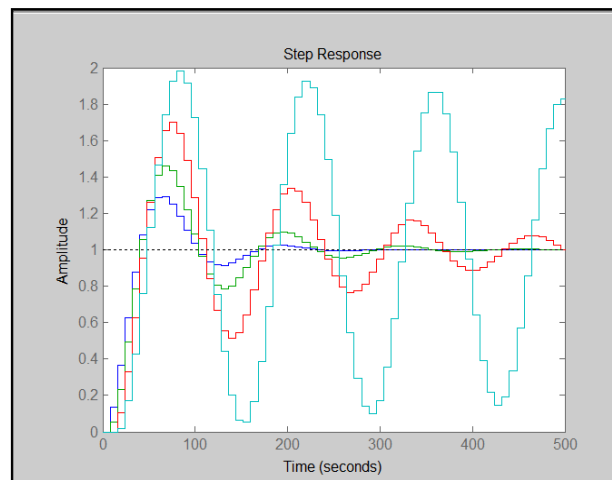
Que es la ecuación de estado del sistema de la figura 3 en bucle abierto. La respuesta del sistema con retardo en bucle cerrado (color verde) calculada y comparada con la respuesta del sistema sin retardo (color azul) por MATLAB se muestra en la figura 25.



**Figura 25:** Sistema FODT sin/con retardo temporal interno

## Muestreo de Sistemas con Retardo

Puede comprobarse que con los valores dados para los parámetros  $h$ ,  $\tau$ ,  $T$  y  $K$  ambas respuestas son estables, si bien el aumento del retardo hace el sistema progresivamente más oscilante, hasta que llega a ser completamente inestable. Para visualizar este hecho, se ha hecho la misma simulación con MATLAB, pero representando la respuesta para cuatro valores del retardo:  $\tau = 0, 5, 10, 15$ . En estos casos, la respuesta es la que se muestra en la figura 26, en la que puede verse que para  $\tau = 15$  (curva azul) la respuesta del sistema en bucle cerrado es muy oscilatoria; aumentando  $\tau$  todavía más, el sistema alcanza la inestabilidad.



**Figura 26:** Sistema FOTD con diversos retardos

El script de MATLAB que calcula las ecuaciones de estado de los sistemas anteriores, así como la respuesta al escalón de los mismos, es el que se muestra en la figura 27. El propio código, junto con los comentarios contenidos en el mismo, permite entender cómo se realiza el cálculo de las ecuaciones de estado de los sistemas (regulador y FOTD), así como la respuesta al escalón del sistema en bucle cerrado anteriormente mencionado.

Como puede observarse, el script de la figura 27 solo puede ser utilizado para reguladores de tipo PI y sistemas de primer orden con retardo. La gráfica de respuesta al escalón que se representa, mantiene los parámetros del regulador PI calculado en el caso de que el sistema de primer orden no posee retardo alguno, de manera que pueda verse el efecto que tiene sobre un sistema de primer orden el aumento del retardo: el sistema inicialmente estable, llega a convertirse en inestable aumentando el valor del retardo hasta un límite.

Obviamente sería interesante conocer cuál sería la respuesta de un sistema de orden superior con cualquier tipo de regulador.

## Muestreo de Sistemas con Retardo

```
% parámetros de entrada
Nr = 10.0;
modo = 'zoh';
paso = 3.0;
tau = 5.0;
tipo_pid = 'pi';
K = 0.25;
T = 24.0;

% Ecuación de estado para S2 (sistema G(s) de primer orden supuesto retardo nulo)
% A2c = -1/T; B2c = K/T; C2c = 1; D2c = 0;
num = K;
dem = [T 1];
[A2c, C2c, B2c, D2c] = tf2ss(num, dem);
G = tf(num, dem);

% Cálculo de las constantes del regulador R(s) (con retardo nulo en G(s)) y de
las ecuación % de estado para S1 (regulador R(s) de tipo PI)
% Alc = 0; B1c = 1.0; C1c = ki; D1c = kp;
R = pidtune(tf(num, dem), tipo_pid);
[Alc, B1c, C1c, D1c] = ssdata(R);

% Estimación del periodo de muestreo
h = ceil(3.0/Nr*T);

% Discretización de S1 (regulador R(s) de tipo PI)
F1 = expm(Alc*h);
fun1 = @(s) exp(Alc*s);
S01 = integral(fun1, 0, h);
A1d = F1; B1d = S01; C1d = C1c; D1d = D1c; E1d = 1.0;

taul = (0:tau:paso*tau);
hold on;

for i = 1:length(taul)

    if taul(i) > 0
        % Discretización de S2 (sistema de primer orden G(s) con retardo taul)
        fun2 = @(s) B2c*exp(A2c*s);
        F2 = expm(A2c*tau);
        D2d = zeros(1);
        D = D2d*D1d;

        if taul(i) <= h
            d = 1;
        else
            d = floor(taul(i)/h) + 1;
            tau2 = taul(i) - (d-1)*h;
        end

        if d > 1
            S02 = integral(fun2, 0, h-tau2);
            S12 = expm(A2c*(h-tau2))*integral(fun2, 0, tau2);
            A2d = zeros(d+1);
            A2d(1, 1) = F2; A2d(1, 2) = S12; A2d(1, 3) = S02;
            A2d(2:d, 3:d+1) = eye(d-1);
            B2d = zeros(1, d+1)'; B2d(d+1) = 1;
            C2d = eye(1, d+1);
        else
            S02 = integral(fun2, 0, h-taul(i));
            S12 = expm(A2c*(h-taul(i)))*integral(fun2, 0, taul(i));
            A2d = zeros(2); A2d(1, 1) = F2; A2d(1, 2) = S12;
            B2d = zeros(1, 2)'; B2d(1) = S02; B2d(2) = 1;
            C2d = eye(1, 2);
        end

        % Composición de S1 (discretizado) con S2 (discretizado)
        A = zeros(d+2); A(1, 1) = A1d; A(2:d+2, 1) = B2d*C1d; A(2:d+2,
2:d+2) = A2d;
        B = zeros(1, d+2)'; B(1) = B1d; B(2:d+2, 1) = (B2d*D1d);
        C = zeros(1, d+2); C(1) = D2d*C1d; C(1, 2:d+2) = C2d;
        E = eye(d+2);

        H = dss(A, B, C, D, E, h);
    else
        H = series(dss(A1d, B1d, C1d, D1d, E1d, h), c2d(ss(A2c, B2c, C2c,
D2c), h, modo));
    end

    W = feedback(H, 1);
    step(W, 500);
end
hold off;
```

**Figura 27:** Script de MATLAB que calcula la respuesta del sistema FOTD

### 3.4. Caso General: Sistemas y Regulador de cualquier orden

Una vez resuelto el problema del cálculo de las ecuaciones de estado discretas del regulador PI y del sistema FOTD, se puede plantear como reescribir dicho script para que pueda ser utilizado para sistemas de orden superior al primero, y reguladores de todo tipo (P, PI, PD y PID), junto con sus diversas variantes.

Para ello, se utilizará el denominado *método de Wittenmark* [Wittenmark-85], que se basa en el hecho de que la expresión de los sistemas muestreados mostrada en 3.2 presenta la importante característica de que puede ser obtenida muestreando primero el sistema sin retardo para los periodos  $h$ ,  $h-\tau$  y  $\tau$ ; esto implica que los cálculos pueden ser obtenidos mediante algoritmos estándar de muestreo de sistemas.

Teniendo esto en cuenta y la ecuación que expresa la discretización del sistema:

$$x_2(kh+h) = \Phi x_2(kh) + \Gamma_0 u(kh - (d-1)h) + \Gamma_1 u(kh - dh) \quad (1)$$

Utilizando algunas funciones de MATLAB como  $c2dm()$ , que convierte un sistema LTI continua a su versión discreta con un periodo de muestreo  $h$  y que permite además elegir dicha conversión en base a un 'modo' que puede ser con mantenedor de orden cero, mantenedor de orden 1 y otros (ver ayuda de MATLAB para más detalle), se puede calcular:

- $[Ad, Bd, Cd, Dd] = c2dm(Ac, Bc, Cc, Dc, h, \text{modo})$   
Esta sentencia proporciona en la matriz Ad la matriz  $\Phi$  de la expresión (1), siendo Xc las matrices en tiempo continuo del sistema que se está discretizando.
- $[Ad0, Bd0, Cd0, Dd0] = c2dm(Ac, Bc, Cc, Dc, h - \tau', \text{modo})$   
Esta sentencia proporciona a través de la matriz Bd0 proporciona la matriz  $\Gamma_0$  de la ecuación (1), siendo Xc como antes las matrices en tiempo continuo del sistema que se está discretizando.
- $[Ad1, Bd1, Cd1, Dd1] = c2dm(Ac, Bc, Cc, Dc, \tau', \text{modo})$   
Esta sentencia proporciona a través de la matriz Bd1 la integral de la matriz multiplicada por Bc de la matriz  $\Gamma_1$  de la ecuación (1), siendo Xc como antes las matrices en tiempo continuo del sistema que se está discretizando.
- Ahora multiplicando las matrices Bd1 y Ad0:  $Bd1 = Ad0 \times Bd1$  se tiene la matriz  $\Gamma_1$  de la ecuación (1).

Todo lo anterior puede verse con mayor nivel de detalle, en el script de la figura 28. Puede verse que además de la representación de las respuestas al escalón de un sistema con diversos valores de retardo, se representa el diagrama de Nyquist de cada uno de ellos. Este script permite ya calcular la respuesta de cualquier sistema con retardo, con cualquier tipo de regulador y con cualquier mantenedor (de orden 0, de orden 1), etc.

# Muestreo de Sistemas con Retardo

```
% parametros de entrada
Nr = 10.0; paso = 3.0; tau = 5.0; t = 500; K = 0.25; Df = [24.0 0.0]; modo = 'zoh'; pid_type = 'pi';
% Parametros de posicionado de los graficos
screensize = get(groot, 'Screensize'); swidth = screensize(3); sheight = screensize(4); wwidth = fix(swidth/3.2); wheight = fix(sheight/2.0);
% Ecuacion de estado para S2 (sistema G(s) de primer orden supuesto retardo nulo)
numG = [K/Df(1)];
demG = poly(-1./Df);
G = tf(numG, demG);
[A2c, B2c, C2c, D2c] = tf2ss(numG, demG);
% Calculo de las constantes kp, ki, kd del regulador R(s) con retardo nulo en G(s).
R = pidtune(G, pid_type);
[kp, ki, kd, Tf] = piddata(R);
[A1c, B1c, C1c, D1c, E1c] = dssdata(R);
% Estimacion del periodo de muestreo
if min(Df) > 0
    h = ceil(min(Df)*(3.0/Nr));
else
    h = ceil(max(Df)*(3.0/Nr));
end
% Discretizacion de S1 (regulador R(s))
[A1d, B1d, C1d, D1d] = c2dm(A1c, B1c, C1c, D1c, h, modo);
E1d = eye(length(A1d));
tau1 = (0:tau:paso*tau);
t_est = zeros(1, length(tau1));
for i = 1:length(tau1)
    if tau1(i) > 0
        tau2 = rem(tau1(i), h);
        d = fix(tau1(i) / h);
        % Discretizacion de S2 (sistema de primer orden G(s) con retardo tau1).
        % A2d=Phi=expm(A2c*h)
        [A2d, ~, C2d, D2d] = c2dm(A2c, B2c, C2c, D2c, h, modo);
        % El termino Ad0 nos proporciona expm(A2c*(h-tau1(i)))
        % El termino Bd0 nos proporciona la matriz Gamma0
        [Ad0, Bd0, Cd0, Dd0] = c2dm(A2c, B2c, C2c, D2c, h-tau2, modo);
        % El termino Bd1 nos proporciona la integral de la matriz multiplicada por B2c de la matriz Gamma1
        if tau2 > 0
            [Ad1, Bd1, Cd1, Dd1] = c2dm(A2c, B2c, C2c, D2c, tau2, modo);
        else
            n = length(A2c); Ad1 = eye(n, n); Bd1 = zeros(n, 1); Cd1 = C2c; Dd1 = D2c;
        end
        % Ahora multiplicando estas dos cantidades se tiene la matriz Gamma1 de la ecuación 3.20
        Bd1 = Ad0*Bd1;
        % Construcción del modelo en espacio de estados discreto de la planta
        if tau1(i) <= h
            A2d = [A2d Bd1; zeros(1, size(A2d, 2)) 0]; B2d = [Bd0; 1]; C2d = [C2d 0];
        else
            A2d = [A2d Bd1 Bd0 zeros(size(A2d, 1), d-1); zeros(d, size(A2d, 2)+1) eye(d); zeros(1, size(A2d, 2)+d+1)]; B2d = [zeros(size(A2d, 1)-1, 1);
1]; C2d = [C2d zeros(1, d+1)];
        end
        E2d = eye(length(A2d)); H = series(dss(A2d, B2d, C2d, D2d, E2d, h), dss(A1d, B1d, C1d, D1d, E1d, h));
    else
        H = series(dss(A1d, B1d, C1d, D1d, E1d, h), c2d(ss(A2c, B2c, C2c, D2c), h, modo));
    end
    W = feedback(H, 1);
    [Ad, Bd, Cd, Dd] = dssdata(W);
    [numW, demW] = ss2tf(Ad, Bd, Cd, Dd, 1);
    Wc = tf(numW, demW);
    Vf = polyval(numW, 0)/polyval(demW, 0);
    [y, ts] = step(W);
    j = length(ts);
    while (y(j) > 0.9*Vf) && (y(j) < 1.1*Vf)
        j = j-1;
    end
    t_est(i) = ts(j); sort(fix(t_est));
    if (std(t_est) > mean(t_est)) && (i > 1)
        t_est(i) = t_est(i-1);
    end
    hold on;
    figure(1);
    set(1, 'Position', [swidth/3-wwidth/2 (sheight-wheight)/2 wwidth wheight]);
    set(1, 'Name', 'Respuesta Temporal', 'NumberTitle', 'off');
    title('Respuesta al escalon unitario');
    step(W, 1.25*max(t_est));
    figure(2);
    set(2, 'Position', [2*swidth/3-wwidth/2 (sheight-wheight)/2 wwidth wheight]);
    set(2, 'Name', 'Nyquist', 'NumberTitle', 'off');
    title('Diagrama de Nyquist');
    nyquist(W);
end
hold off;
```

**Figura 28:** Script de MATLAB: cálculo la respuesta de cualquier sistema

### 3.5. Caso General: Ejemplo práctico para un sistema de 2º orden

Para ilustrar la aplicación de lo expuesto en el punto anterior, se elige un sistema de segundo orden; en particular, se usará la función de transferencia simplificada de un intercambiador de calor controlado por válvula.

La función de transferencia real de un intercambiador de calor es de tercer orden; es decir, el denominador es un polinomio de tercer orden en  $s$ . Se puede demostrar [Corripio-91] que dicha función de transferencia es esencialmente el producto de tres sistemas FOTD. Si la temperatura de entrada del agua de enfriamiento cambia, lo primero que se ve afectado es la temperatura del agua en la camisa de enfriamiento; por lo tanto esta afecta, a su vez, a la de la pared de metal; y finalmente, esta última influye en la del fluido que se está procesando: de ahí los tres sistemas de primer orden en serie; no obstante puede utilizarse la siguiente fdt con suficiente exactitud:

$$G(s) = \frac{10e^{-10s}}{(50s+1)(10s+1)}$$

El periodo de muestreo, será  $h=3$ , de manera que será necesario incluir 3 variables de estado adicionales a las propias del sistema. Las matrices para el sistema  $G(s)$  que se deben de calcular son las de la expresión (1):

$$x_2(kh+h) = \Phi x_2(kh) + \Gamma_0 u(kh - (d-1)h) + \Gamma_1 u(kh - dh)$$

Utilizando la función  $c2dm()$  tal y como se ha indicado en el punto 7, se tiene:

$$\Phi = \begin{pmatrix} 0.69 & -0.005 \\ 2.51 & 0.99 \end{pmatrix}$$

$$\Gamma_0 = \begin{pmatrix} 1.77 \\ 1.84 \end{pmatrix}$$

$$\Gamma_1 = \begin{pmatrix} 0.73 \\ 2.15 \end{pmatrix}$$

El regulador obtenido con MATLAB (supuesto  $\tau = 0$ ) es el siguiente:

$$R(s) = 0.0166 + \frac{0.0005}{s}$$

## Muestreo de Sistemas con Retardo

---

Una vez incluidos los 3 estados adicionales a los 2 del sistema  $G(s)$  y el del regulador  $R(s)$  al construir el modelo de espacios de estados discreto del se tienen las siguientes matrices:

$$A = \begin{pmatrix} 0.69 & -0.005 & 0.73 & 1.77 & 0 & 0 \\ 2.51 & 0.99 & 2.15 & 1.84 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$C(0 \ 0.2 \ 0 \ 0 \ 0 \ 0)$$

Que describen completamente el sistema de segundo orden con retardo analizado.

Obsérvese que mediante la discretización del sistema con retardo, se han introducido variable  $s$  de estado extra, que representan los valores anteriores a la acción de control. El sistema continuo con retardo, que se ha visto es de dimensión infinita, sin embargo, el discretizado es de dimensión finita: los retardos son por tanto más manejables si se trabaja con el sistema discretizado.

## Sistemas con Muestreo Multifrecuencia

---

### 4.1. Introducción

En las aplicaciones de control, pronto hubo que enfrentarse al problema del tratamiento de las diferentes constantes de tiempo de los elementos que formaban parte de un sistema dinámico, que inicialmente se abordaba descomponiendo el sistema dinámico en diferentes sistemas monofrecuencia aplicando técnicas clásicas, como la transformada  $z$  modificada.

La primera técnica propiamente dicha de análisis de sistemas multifrecuencia conocida como descomposición frecuencial se debe a Sklansky, y data del año 1955 [Sklansky-55]. En 1957, Krank [Krank-57] desarrolló el método de análisis que denominó descomposición de conexiones; posteriormente Jury en 1967 demostró que ambos métodos eran equivalentes [Jury-67].

En 1980 Whitbeck [Whitbeck-80] desarrolló el método de descomposición vectorial de conexiones introduciendo vectores en la fase de modelado, lo que permitió también trabajar con sistemas de muestreo irregular, denominado así por no existir un periodo de muestreo fijo, aunque el patrón de muestreo debe de repetirse periódicamente.

Los métodos descritos solo son aplicables en el dominio de la frecuencia siendo difíciles de usar en la práctica por sus extremas complejidades matemáticas; además, no se puede obtener con ellos ninguna función de transferencia del sistema bajo análisis. En 1986 Thompson [Thompson-86] desarrolla el método de los operadores de Kranc, que resuelve este inconveniente al permitir, mediante la introducción de los llamados operadores de Kranc, el estudio sistemático de la descomposición de conexiones y su adecuada implementación por ordenador. El método de los operadores de Kranc es matricial, basado en la ecuación de estado de aplicación en los sistemas SISO.

Broussard [Broussard-80] desarrolla en 1980 un método de análisis de sistemas multifrecuencia en el dominio del tiempo para controladores óptimos, aunque anteriormente ya hubo algunos trabajos poco desarrollados.

En 1975, Meyer [Meyer-75] analiza los sistemas periódicos y sistemas multifrecuencia a partir de una representación equivalente invariable en el tiempo. Este planteamiento sería adoptado por la mayoría de autores en trabajos futuros.

En 1986 Araki y Yamamoto desarrollan el método de sustituciones progresivas [Araki-86] para el modelado sistemas MIMO y logran resolver algunos de los problemas de los desarrollos anteriores. Este modelo es de elevada dimensión, con vectores ampliados de estado, entradas y salidas.

En 1986, Thompson [Thompson-86] desarrolla un modelo que constituye un interesante híbrido entre la descomposición vectorial de conexiones y la expansión de estados. Estaba pensado para sistemas SISO sin restricciones en los reguladores a utilizar.

En 1990 Godbout, Jordan y Apostolakis [Apostolakis-91] desarrollaron un modelo para un sistema de control en lazo cerrado que incorpora muestreo multifrecuencia con compensación dinámica, pero su dimensión es aún demasiado grande para poder aplicarse de una manera práctica; finalmente, en 1991, Apostolakis y Jordan [Apostolakis-91] lograron reducir las dimensiones de dicho modelo.

A Finales de los años ochenta se comienza a ver un auge en la teoría y aplicación del control multifrecuencia, debido a la aparición de modelos más fáciles de aplicar en un ordenador y, sobre todo, a la comercialización de procesadores más rápidos y potentes.

Una referencia histórica más amplia y detallada de la evolución histórica de los modelos y herramientas desarrollados para el análisis de los sistemas multifrecuencia, puede verse en [Salt-00].

### 4.2. Modelos Multifrecuencia

Para estudiar correctamente un sistema multifrecuencia, es necesario establecer un modelo matemático que represente de manera razonable dicho sistema. En el área de multifrecuencia, se puede asegurar que existe una amplia gama de modelos muy diferentes, tanto desde el punto de vista conceptual como de aplicación; no se va a entrar en este trabajo en el análisis general de los mismos, que puede consultarse en [Salt-00], si bien los métodos descritos en dicha referencia se aplican exclusivamente a sistemas multifrecuencia con muestreo regular (aunque cada variable puede muestrearse con un periodo arbitrario, este siempre es constante).

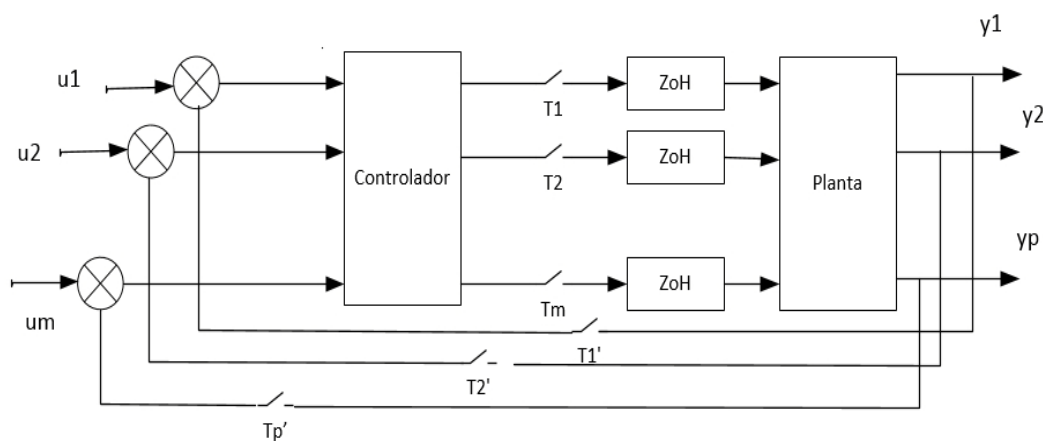
En este trabajo se utilizará el modelo desarrollado por Carlos Mario Vélez [Velez-02], que tiene la ventaja de que generaliza los descritos en la

## Sistemas con Muestreo Multifrecuencia

referencia anterior, y permite también modelar sistemas con muestreo irregular (es decir, cada variable se muestrea en intervalos arbitrarios, pero la secuencia de muestreo se repite cada periodo global  $T_0$ ), incluso permite considerar retardos si fuera necesario. Existe una *Toolbox* de MATLAB desarrollada por el autor de esta metodología [Velez-04b] que implementa la misma, y que puede descargarse a través de la comunidad MATLAB de MATHWORKS en la siguiente URL:

<https://es.mathworks.com/matlabcentral/fileexchange/4496-multirate-control-toolbox-5-1?requestedDomain=www.mathworks.com>

El modelo desarrollado por Vélez parte de un sistema multifrecuencia genérico irregular de orden  $n$  con  $m$  entradas y  $p$  salidas, tal y como se muestra en la figura 29.



**Figura 29:** *Esquema de un sistema multifrecuencia genérico*

Es preciso destacar que el hecho de que el sistema pueda ser muestreado de manera irregular, aporta dos importantes ventajas:

- Incorpora un grado de libertad más para el diseño del regulador del sistema, lo que en determinados casos redundará en una mejora de la respuesta dinámica del mismo (por ejemplo, ayudando a garantizar la observabilidad y controlabilidad del sistema).
- Se adapta mejor a la realidad de algunos sistemas; en particular se adapta mejor al funcionamiento real de los sistemas controlados por eventos.

### 4.3. Método de Modelado Multifrecuencia

En este apartado se describirá brevemente el método utilizado de modelado multifrecuencia, de acuerdo con la figura 29.

El modelo aquí presentado se basa en ampliar o extender los vectores de entrada y de salidas, y consiste básicamente en la obtención de un modelo

monofrecuencia invariante en el tiempo equivalente del sistema multifrecuencia, pero con mayor número de entradas y salidas que el sistema original (vectores extendidos o ampliados). Este proceso de 'ampliado' o 'extendido' conserva las operaciones algebraicas en los sistemas, en las señales (picos, potencia, sumas) y en la estabilidad de los sistemas. Sin embargo, el diseño introduce la restricción de causalidad del sistema.

El método más sencillo de diseño de reguladores multifrecuencia que garantiza la causalidad, consiste en la discretización directa de reguladores continuos (con retenedores de orden cero) obtenidos previamente (en este caso con MATLAB). Tiene la ventaja adicional de no presentar problemas de rizado, pero la aproximación es mejor cuanto menor sean los periodos de muestreo de las señales; de hecho, para periodos largos de muestreo del sistema continuo equivalente, no se puede garantizar la estabilidad en lazo cerrado del sistema.

El modelo (en variable continua) de la planta, que se considera observable y controlable como hipótesis de partida, es el que se indica a continuación:

$$\begin{aligned}\dot{x} &= A_p x + B_p u \\ y &= C_p x + D_p u\end{aligned}$$

Donde:  $u = (u_1, u_2, \dots, u_m)^T$ ,  $y = (y_1, y_2, \dots, y_p)^T$ ,  $x = (x_1, x_2, \dots, x_n)$

Las ecuaciones de la planta en variable discreta son, por tanto:

$$\begin{cases} x((k+1)T) = Ax(kT) + Bu(kT) \\ y(kT) = Cx(kT) + Du(kT) \end{cases}$$

La obtención del modelo discreto a periodo base  $T$  está descrita en detalle en [Aström-97], que es el mostrado en la expresión siguiente:

$$\begin{aligned}A &= A(T) = e^{A_p T}, \quad B = B(T) = \int_0^T e^{A_p \tau} B_p d\tau, \quad B = (B_1, B_2, \dots, B_m), \\ C &= (C_1, C_2, \dots, C_p)^T \quad \text{y} \quad D = (D_1, D_2, \dots, D_p)\end{aligned}$$

Siendo  $m$  el número de entradas del sistema, y  $p$  el número de salidas del sistema. Por sustituciones progresivas se llega finalmente a un modelo descrito en detalle en [Vélez-04a], y que a continuación se muestra de manera resumida:

$$\begin{cases} x((k+1)T_0) = \tilde{A}x(kT_0) + \tilde{B}u^D(kT_0) \\ y^D(kT_0) = \tilde{C}x(kT_0) + \tilde{D}u^D(kT_0) \end{cases}$$

Donde:

$$\begin{aligned} x(kT_0) &= (x_1(kT_0), \dots, x_n(kT_0))^T \\ u^D(kT_0) &= (u_1(kT_0 + \tilde{T}_{11}), \dots, u_1(kT_0 + \tilde{T}_{1\tilde{N}_j}) | \dots)^T \\ y^D(kT_0) &= (y_1(kT_0 + T_{11}), \dots, y_1(kT_0 + T_{1N_i}) | \dots)^T \end{aligned}$$

Y  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$  y  $\tilde{D}$  son las matrices del sistema discretizado. Los coeficientes e índices de las expresiones anteriores están descritos con todo detalle en [Vélez-04a], y se indican a continuación:

- $T$  es el periodo base.
- $T_0 = NT$  es el periodo global.
- $\tilde{T}_{jv}$  son los instantes de muestreo de la entrada.
- $\tilde{N}_j$  es el número de muestras de la entrada  $j$  en el periodo global  $T_0$ .
- $T_{iv}$  son los instantes de muestreo de la salida  $i$  en el periodo global  $T_0$ .
- $N_i$  es el número de muestras de la salida  $i$  en el periodo global  $T_0$ .

Como se ha indicado anteriormente, este modelo soporta sistemas SISO o MIMO, con muestreo regular o irregular, con retardo o sin retardo, y con sincronismo de entradas y salidas o sin él; además es causal por la forma de las matrices, de dimensión mínima, y de fácil implementación en un ordenador, para diversos patrones de muestreo multifrecuencia.

#### 4.4. Multirate Control Toolbox

El *Multirate Control Toolbox* o MCT, es una *Toolbox* desarrollada para MATLAB / SIMULINK con el fin de facilitar el análisis de sistemas multifrecuencia y la síntesis de reguladores para dichos sistemas.

La simulación y análisis de estos sistemas se lleva a cabo con esta *Toolbox* de manera análoga a como se hace para los sistemas monofrecuencia clásicos.

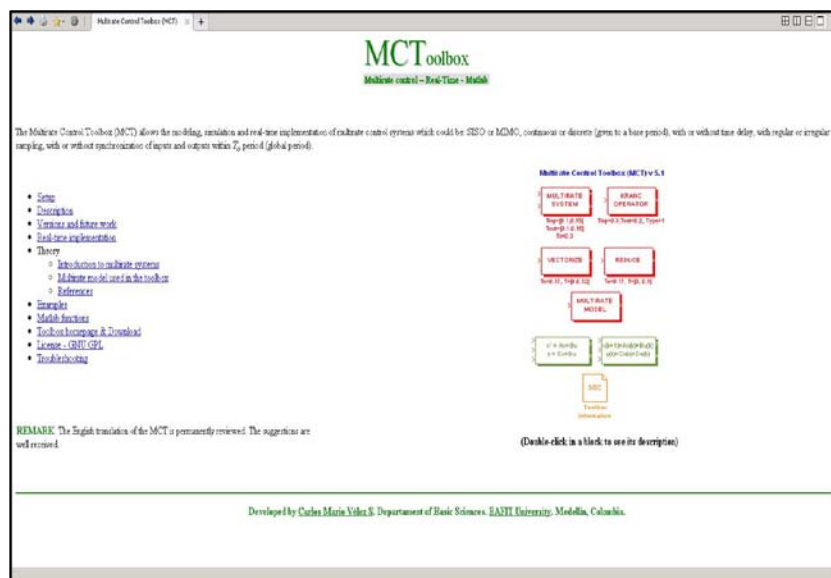
A pesar de la amplia utilización de MATLAB y SIMULINK en el análisis, estudio y aplicación de sistemas de control en todos los ámbitos de aplicación de estos, no existían para esta plataforma herramientas

# Sistemas con Muestreo Multifrecuencia

específicas para aplicación en sistemas, vacío que el desarrollo de MCT ha cubierto, de manera que con MCT puede trabajarse con los sistemas multifrecuencia como se trabaja para los sistemas convencionales monofrecuencia.

Conceptualmente, la *Toolbox* MCT se basa en tres fases bien definidas:

- Muestreo de cada señal de entrada según su patrón a periodo base, y generación del vector de entradas ampliado.
- Cálculo de la salida ampliada según el modelo multifrecuencia.
- Aplicación de la salida correspondiente al instante de muestreo actual (Reducción).



**Figura 30:** Vista de entrada de la ayuda de Multirate Control Toolbox

Todos los cálculos quedan encapsulados en las funciones correspondientes de la *Toolbox*. Los detalles de desarrollo y aplicación de la misma pueden verse en detalle y con ejemplos en [Vélez-04a] y [Vélez-04b], aparte de los de la ayuda de la propia aplicación.

En la figura 30 puede verse la vista de entrada de la ayuda de MCT, en la que por cierto se pueden encontrar varios ejemplos, tanto de sistemas SISO como MIMO, y con muestreo regular o irregular. La función más importante de esta Toolbox es la siguiente:

$$sm = multirate(S, T_{inp}, T_{out}, T_0)$$

Esta función calcula el modelo multifrecuencia del sistema  $S$  (continuo o discreto, con muestreo regular o irregular, con o sin retardos), mediante el

método desarrollado por Vélez anteriormente descrito. Los argumentos de la función se describen a continuación:

- $sm$ : Modelo multifrecuencia lineal e invariante (LTI) con entradas y salidas ampliadas del sistema  $S$ .
- $S$ : Ecuación de estado o función de transferencia del sistema (continuo o discreto, SISO o MIMO, con o sin retardos) del sistema cuyo modelo multifrecuencia se desea calcular.
- $T_{inp}$ : Patrón de muestreo de las señales de entrada en segundos. Para muestreo irregular  $T_{inp} = \{\dots, [T_{j1}, \dots, T_{jN_j}], \dots\}$ ; para muestreo regular:  $T_{inp} = [T_1, T_2, \dots]$ .
- $T_{out}$ : Patrón de muestreo de las señales de salida en segundos. Para muestreo irregular:  $T_{out} = \{\dots, [T_{i1}, \dots, T_{iN_i}], \dots\}$ ; para muestreo regular:  $T_{out} = [T_1, T_2, \dots]$ .
- $T_0$ : Periodo global de muestreo, con  $T_0 = NT$  ( $N \in \mathbb{N}$ ).

### 4.5. Caso de estudio

En el presenta trabajo se analizará el comportamiento de un sistema SISO, inicialmente de primer orden, con un muestreo bifrecuencia síncrono con el siguiente patrón de muestreo:

- Muestreo lento a la salida de la planta (de periodo global  $T_0$ ).
- Muestreo rápido de la señal de control de entrada a la planta (de periodo base  $T$ ).
- Se cumple que:  $T_0 = NT$  ( $N \in \mathbb{N}$ ).
- Este sistema se analizará sin retardo, y con un retardo  $\tau$  a la salida de la planta, con la condición:  $\tau < T_0$ .
- El sistema bajo análisis es el mismo sistema de primer orden usado en capítulos anteriores:

$$G(s) = \frac{0.25}{24s + 1} e^{-\tau s}$$

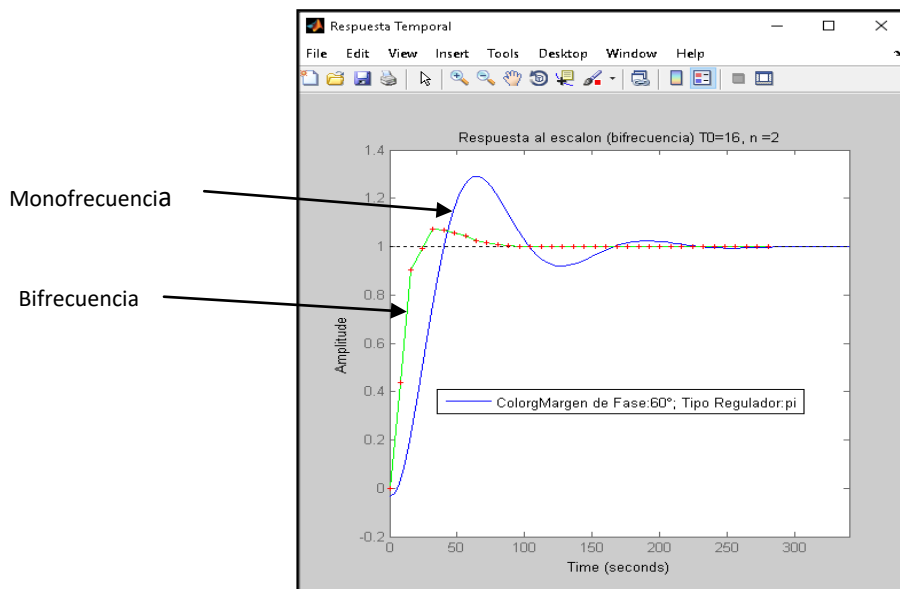
Se usará la *Toolbox* MCT desarrollada para MATLAB por Vélez [Vélez-04a] siempre que sea posible, por su flexibilidad a la hora de modelar matemáticamente los diferentes esquemas de muestreo que pueden darse en un sistema real.

### 4.6. Comparación caso Monofrecuencia – caso Bifrecuencia ( $\tau = 0$ )

En este epígrafe se trata de analizar las diferencias más significativas de la respuesta del sistema indicado con un regulador PI en el caso de muestreo convencional monofrecuencia, y muestreo bifrecuencia.

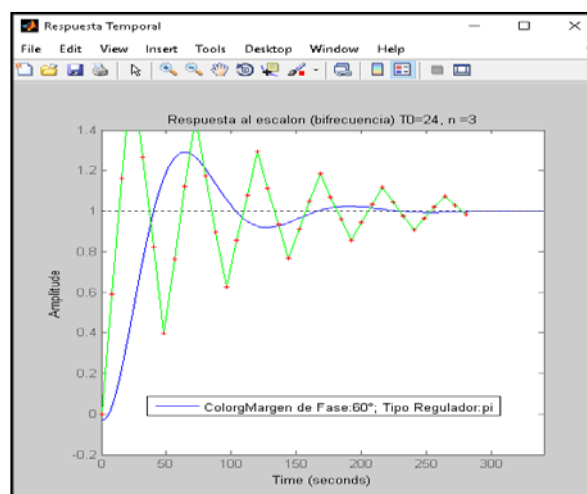
## Sistemas con Muestreo Multifrecuencia

Se establece inicialmente un margen de fase de  $60^\circ$ , y se compara la respuesta del sistema monofrecuencia con el sistema bifrecuencia; pueden verse los resultados en la figura 31: el sistema bifrecuencia (en línea verde,  $N=2$ ) es más rápido y presenta menos sobreoscilación que el sistema muestreado con una única frecuencia (en línea azul). El tiempo de establecimiento  $t_s$  es prácticamente el triple en el sistema monofrecuencia que en el bifrecuencia.



**Figura 31:** *Respuesta del sistema mono y bifrecuencia ( $N=2$ )*

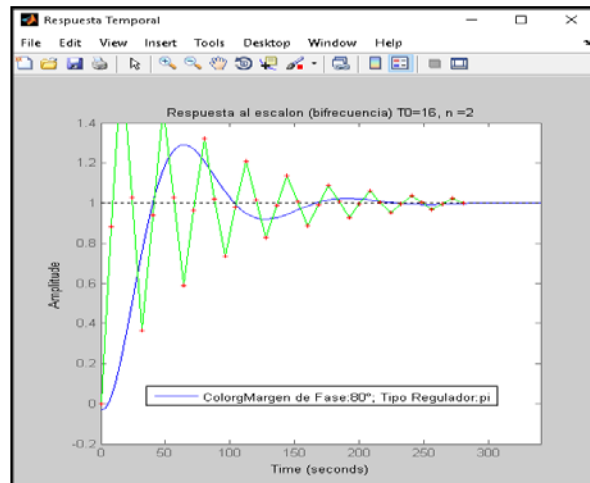
Sin embargo, si se aumenta la relación de frecuencias a  $N=3$  (es decir: la salida de la planta se muestrea a una frecuencia 3 veces inferior a la entrada), manteniendo el resto de parámetros iguales, entonces el sistema se hace oscilatorio, como puede verse en la figura 32.



**Figura 32:** *Respuesta del sistema mono y bifrecuencia ( $N=3$ )*

## Sistemas con Muestreo Multifrecuencia

Adicionalmente, un cambio del margen de fase, manteniendo el resto de parámetros del sistema, también provoca una tendencia a la oscilación del sistema, como pone de manifiesto la figura 33, en la que puede verse que el sistema oscila y el tiempo de establecimiento se alarga de manera considerable, igual que sucede en la figura 32.



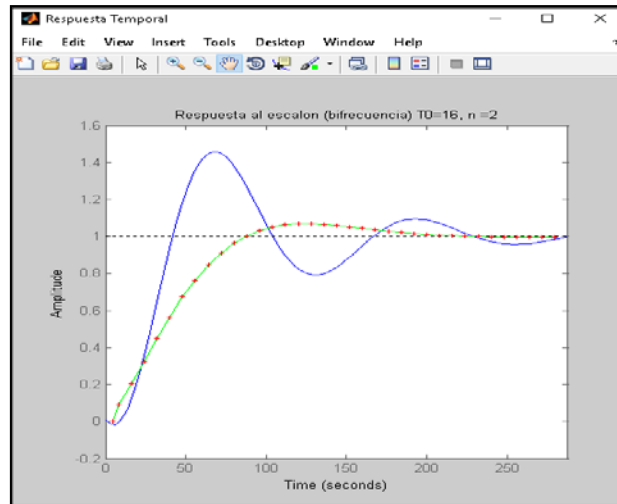
**Figura 33:** *Respuesta del sistema mono y bifrecuencia*

En resumen, se puede afirmar en base a los resultados de la simulación, que cuando no hay retardo el sistema bifrecuencia puede ser más rápido y presentar menos sobreoscilación que el sistema monofrecuencia. Si la relación de muestreo  $N$  se va haciendo más grande, el sistema se vuelve más oscilatorio, hasta que para un  $N$  suficientemente grande, se hace inestable, por lo que la mejora que se puede obtener ha de ser cuidadosamente evaluada, para evitar la tendencia a la inestabilidad antes indicada. Estos resultados están también de acuerdo con las conclusiones de otros trabajos sobre reguladores multifrecuencia, como los realizados por [Cuesta-04] y [Cuenca-04].

### 4.7. Comparación caso Monofrecuencia – caso Bifrecuencia ( $\tau \neq 0$ )

El caso en el que existe retardo es más complejo de simular; la figura 34 muestra una simulación del sistema monofrecuencia con  $\tau = 5$  s y del sistema con muestreo bifrecuencia ( $N=2$ ) y con el mismo retardo  $\tau = 5$  s.

Puede verse que el muestreo bifrecuencia reduce la oscilación del sistema, la sobreoscilación es mucho más pequeña, y el tiempo de establecimiento  $t_s$  es más reducido (120 s en el caso bifrecuencia, y unos 250 s. en el caso monofrecuencia). En las graficas puede verse el comportamiento del sistema de primero orden y el sistema bifrecuencia, en el que la planta es un sistema continuo.

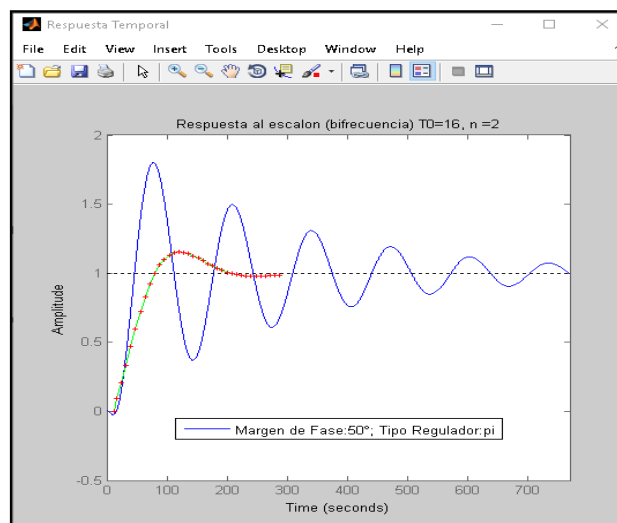


**Figura 34:** *Respuesta del sistema mono y bifrecuencia ( $\tau=5$ )*

Si ahora se aumenta el retardo a 12 s, se tiene la respuesta que se muestra en la figura 35. Es claro en base a las simulaciones obtenidas para el sistema de primer orden:

$$G(s) = \frac{0.25}{24s+1} e^{-\tau s}$$

Que el muestreo bifrecuencia consigue una estabilidad mejorada respecto del monofrecuencia, con reducción de oscilaciones, con un tiempo de establecimiento  $t_s$  más reducido (de 150 s frente al caso monofrecuencia, en el que  $t_s > 500$  s), y una sobreoscilación del 18% frente al 76%. Puede comprobarse que el aumento de N (por ejemplo N=3), hace al sistema bifrecuencia más oscilante, para un mismo valor del retardo  $\tau = 5$ .



**Figura 35:** *Respuesta del sistema mono y bifrecuencia ( $\tau=12$ )*

## Sistemas con Muestreo Multifrecuencia

Estos resultados están también de acuerdo con las conclusiones de otros trabajos sobre reguladores multifrecuencia, como los realizados por [Cuesta-04] y [Cuenca-04].

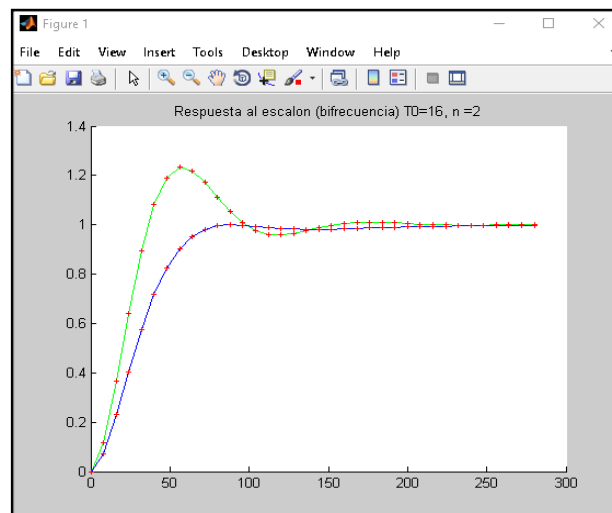
### 4.8. Extensión a sistemas de orden superior

Cabe preguntarse si se obtienen resultados similares con sistemas de orden mayor que uno; para ello se realizarán algunas simulaciones con un sistema de 2° orden; en particular, con el sistema:

$$G(s) = \frac{10e^{-\tau s}}{(50s+1)(10s+1)}$$

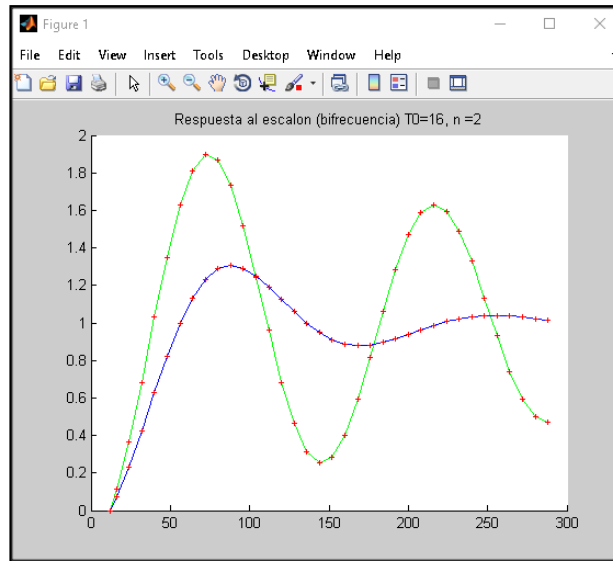
Este sistema ya fue utilizado en el capítulo anterior; recordamos aquí únicamente que se trata de la fdt simplificada de un intercambiador de calor controlado por válvula; como todos los sistemas químicos o termodinámicos, es un sistema con respuestas lentas y en general retardadas, debido a los retardos de transporte de las materias involucradas en estos procesos.

La figura 36 nos muestra los resultados obtenidos para muestreo bifrecuencia del sistema de segundo orden indicado con dos márgenes de fase diferentes: 60° y 75°. En ambos casos, el sistema es estable, y puede controlarse la respuesta del sistema de manera eficaz.



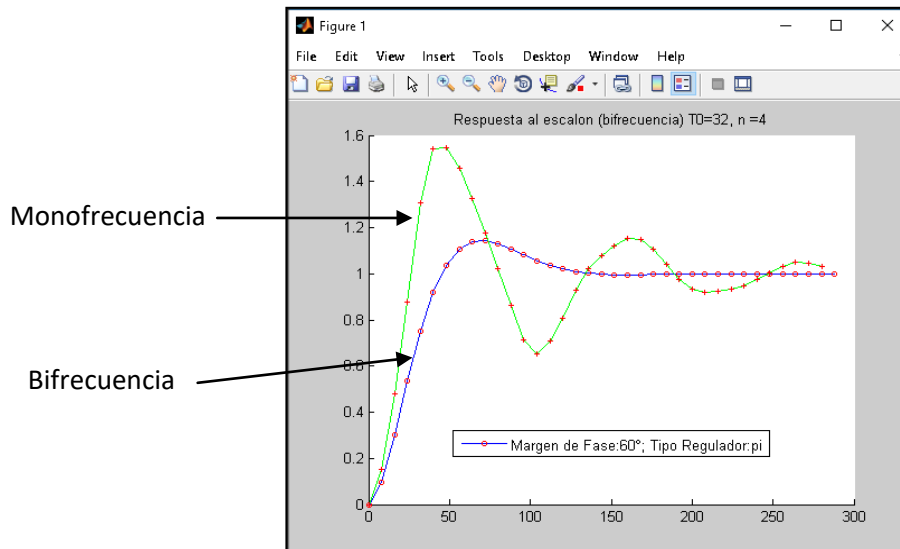
**Figura 36:** *Respuesta del sistema mono y bifrecuencia ( $\tau=0$ )*

La figura 37 muestra el comportamiento del sistema con diferentes márgenes de fase cuando la planta presenta un retardo de 12 s; puede verse como el sistema se vuelve más inestable y tiende a oscilar cuando el margen de fase se modifica.



**Figura 37:** *Respuesta del sistema mono y bifrecuencia ( $\tau=12$ )*

La figura 38 muestra la respuesta del sistema sin retardo, con un valor de  $N=4$ . El aumento del valor de  $N$ , y por tanto de la diferencia entre la frecuencia de muestreo a la salida y entrada del sistema, aumenta la sobreoscilación, y en general vuelve al sistema más inestable. El mismo resultado se obtiene si dejando  $N$  constante, se aumenta el valor del retardo.



**Figura 38:** *Respuesta del sistema mono y bifrecuencia ( $\tau=0$ )*

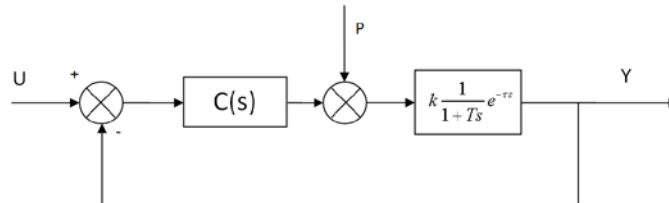
#### 4.9. Comportamiento frente a perturbaciones

A continuación se analizará el comportamiento del sistema  $G(s)$  de primer orden frente a perturbaciones, siendo  $G(s)$ :

## Sistemas con Muestreo Multifrecuencia

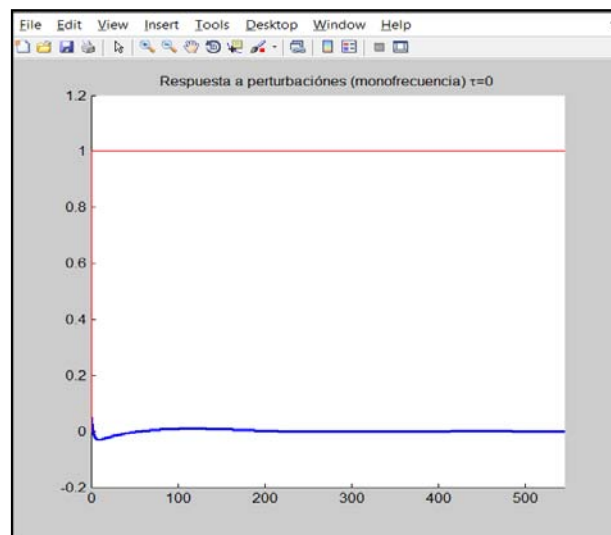
$$G(s) = \frac{0.25}{24s+1} e^{-\tau s}$$

Se simulará utilizando un regulador de tipo PI y con muestreo monofrecuencia, supuesta la entrada nula,  $\tau=0$ ,  $\tau=5$  y una perturbación de tipo escalón senoidal o cuadrada introducida en el sistema según se indica en la figura 39.



**Figura 39:** Sistema  $G(s)$  con perturbaciones ( $P$ )

En este caso el comportamiento frente a la perturbación escalón es el mostrado en la figura 40, en la que en color rojo se representa la perturbación (escalón unitario), y en color azul la respuesta del sistema a la misma. El error instantáneo máximo que se produce es del 2,81%



**Figura 40:** Respuesta de  $G(s)$  a una perturbación escalón ( $\tau=0$ )

La simulación de una perturbación de tipo escalón es sencilla; dado el sistema mostrado en la figura 39, la fdt que relaciona la salida del sistema con la perturbación, supuesta la entrada nula es:

$$Z(s) = \frac{1}{C(s)} \times \frac{C(s)G(s)}{1+C(s)G(s)} = \frac{G(s)}{1+C(s)G(s)}$$

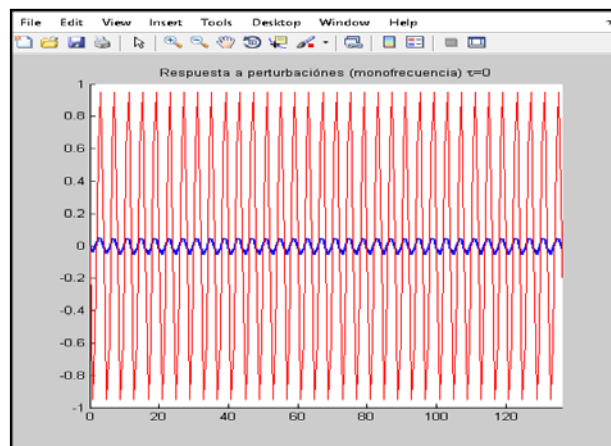
## Sistemas con Muestreo Multifrecuencia

Fácil de calcular con MATLAB mediante las funciones *series()* y *feedback()* en el caso monofrecuencia [Kamman-17]. Por otra parte, la función MATLAB *gensig()* permite generar señales periódicas de test de tipo senoidal, cuadrada o impulso; y finalmente la función *lsim()* nos permite simular la respuesta de cualquier sistema  $Z(s)$  a cualquier señal de entrada arbitraria.

Con todo ello, no es difícil simular la respuesta del sistema de la figura 39 a una señal de entrada senoidal de frecuencia 1 s y amplitud 1 mediante las sentencias:

```
[up, ti] = gensig('sin', 1, 1.25*max(t_est(i)), tm);  
[y1, tf] = lsim(Z, up, ti);  
plot(tf, y1, 'LineWidth', 2, 'Color', 'b');  
plot(ti, up, 'Color', 'red');
```

Cuya representación gráfica es la mostrada en la figura 41; en rojo se representa la señal perturbadora senoidal, y en color azul la respuesta del sistema; el error instantáneo máximo producido es menor del 5.0%.

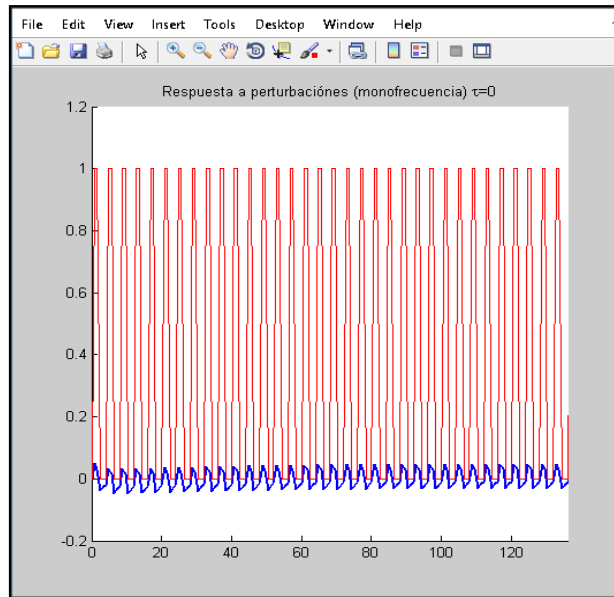


**Figura 41:** Respuesta de  $G(s)$  a perturbación senoidal ( $\tau = 0$ )

Del mismo modo, podemos simular la respuesta del sistema a una señal cuadrada de amplitud 1 y periodo 1 s, mediante las sentencias:

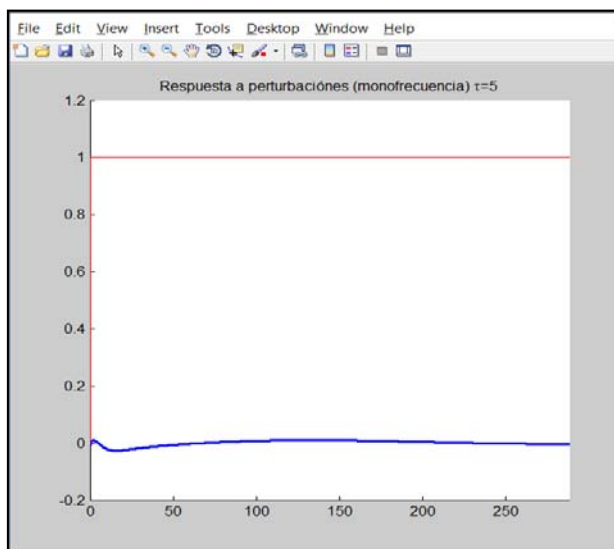
```
[up, ti] = gensig('square', 1, 1.0*max(t_est(i)), tm);  
[y1, tf] = lsim(Z, up, ti);  
plot(tf, y1, 'LineWidth', 2, 'Color', 'b');  
plot(ti, up, 'Color', 'red');
```

Cuya respuesta cuando se simula en el sistema bajo análisis se muestra en la figura 42; en color rojo, la señal perturbadora, en azul la respuesta. El máximo error instantáneo medido es del 5%.



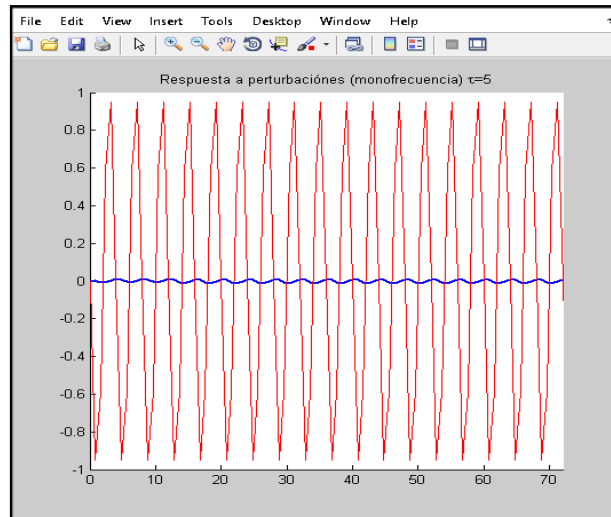
**Figura 42:** *Respuesta de  $G(s)$  a perturbación cuadrada ( $\tau = 0$ )*

Considerando ahora muestreo monofrecuencia con un retardo  $\tau = 5$  s, en el caso de una perturbación tipo escalón la respuesta del sistema es la mostrada en la figura 43; el máximo error instantáneo medido fue del 2.64%



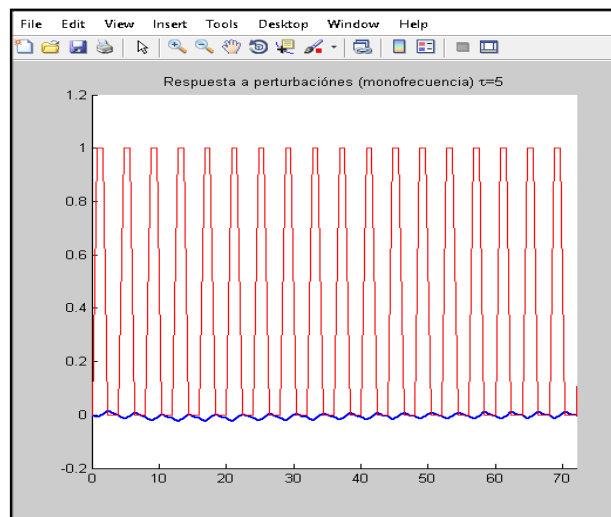
**Figura 43:** *Respuesta de  $G(s)$  a una perturbación escalón ( $\tau = 5$  s)*

Que como puede verse, apenas difiere de la respuesta del sistema en el caso  $\tau = 0$ , y con un valor del error muy similar. En el caso de la perturbación senoidal, la respuesta del sistema con el retardo indicado, se muestra en la figura 44.



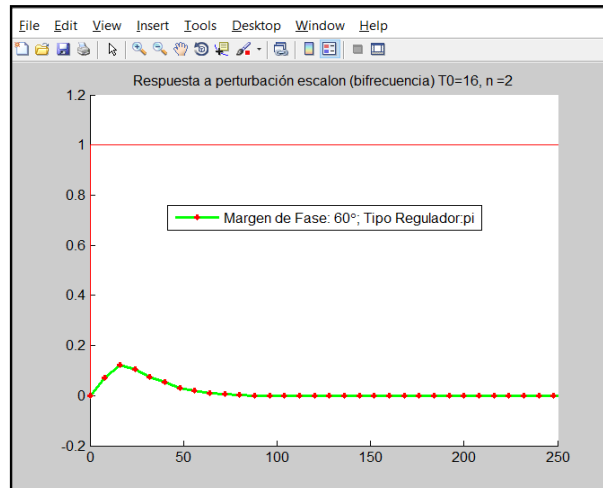
**Figura 44:** *Respuesta de  $G(s)$  a perturbación senoidal ( $\tau=5$ )*

Ahora, la amplitud de la perturbación se ve algo más atenuada que en el caso sin retardo; el error instantáneo pasa al 1.20% (frente al 5.0%); procediendo de igual modo con la perturbación cuadrada, se obtiene la respuesta indicada en la figura 45; donde el error instantáneo medido respecto a la referencia es del 2.35%, menor también que el medido con  $\tau = 0$  (5% en este último caso).



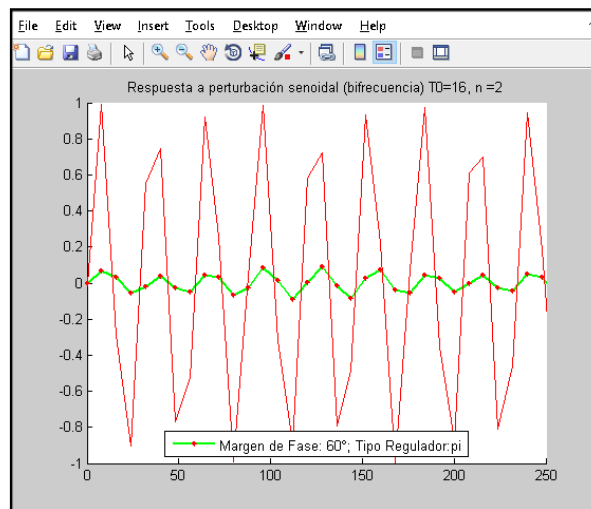
**Figura 45:** *Respuesta de  $G(s)$  a perturbación cuadrada ( $\tau=5$ )*

Para muestreo bifrecuencia, con  $N=2$  la respuesta del sistema para el caso de la perturbación tipo escalón unitario es la mostrada en la figura 46, donde el error máximo producido a la salida respecto la referencia (que es nula) es del 12.15%, apreciablemente superior al medido en el caso monofrecuencia.



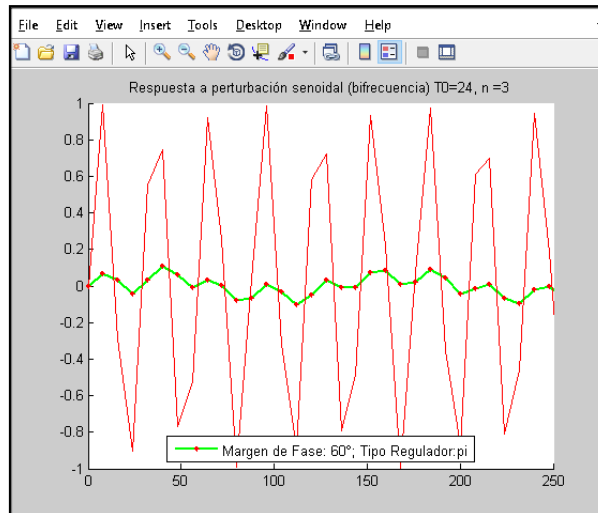
**Figura 46:** *Respuesta de  $G(s)$  a una perturbación escalón ( $N=2$ ,  $\tau=0$ )*

Para el caso de una perturbación senoidal de amplitud 1 y frecuencia 1 se obtiene la respuesta mostrada en la figura 47 ( $\tau=0$ ,  $N=2$ , regulador tipo PI), con un error instantáneo medido del 9.5%, superior al medido en el caso monofrecuencia (5%).



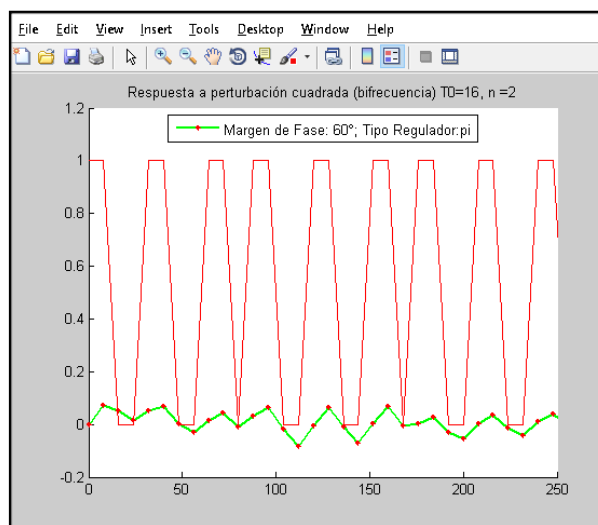
**Figura 47:** *Respuesta de  $G(s)$  a perturbación senoidal ( $N=2$ ,  $\tau=0$ )*

Aumentando el valor de  $N=3$  y manteniendo la perturbación senoidal con regulador tipo PI, la respuesta del sistema es la mostrada en la figura 48, donde se aprecia una atenuación similar, aunque ligeramente menor en valores absolutos respecto a la obtenida con  $N=2$  (figura 47). Recuérdese no obstante, que de acuerdo con los resultados obtenidos en simulaciones anteriores, el aumento de  $N$  tiende a hacer el comportamiento del sistema más oscilante. El error instantáneo máximo medido a la salida es del 10.9%, algo superior al obtenido con  $N=2$ .



**Figura 48:** *Respuesta de  $G(s)$  a perturbación senoidal ( $N=3$ ,  $\tau=0$ )*

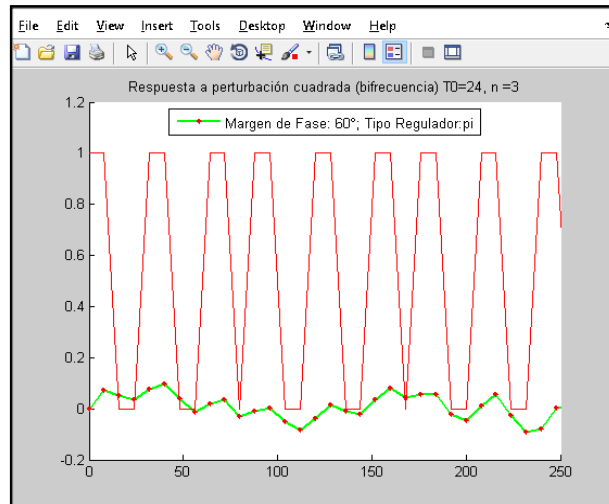
Para el caso de una perturbación de onda cuadrada, amplitud 1, periodo 1 s y  $N=2$ , la respuesta obtenida se muestra en la figura 49. Se observa que la citada perturbación se atenúa de forma similar al caso senoidal, midiéndose un error instantáneo máximo del 8.3%.



**Figura 49:** *Respuesta de  $G(s)$  a perturbación cuadrada ( $N=2$ ,  $\tau=0$ )*

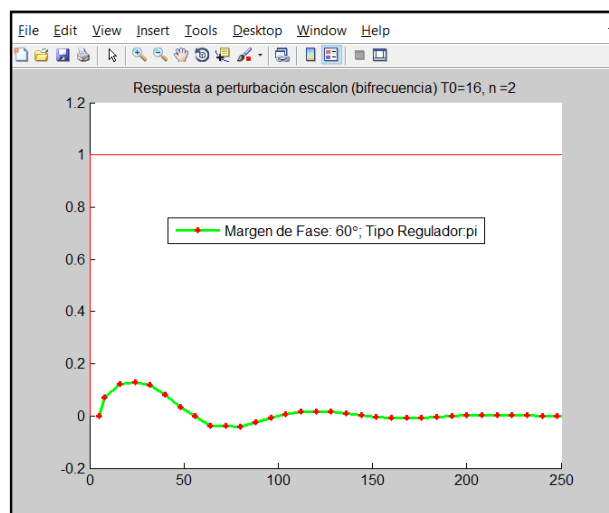
En la figura 50, se ve que las amplitudes y frecuencia de la salida ante una perturbación cuadrada de periodo 1, pero con  $N=3$ , es algo superior al caso anterior, con  $N=2$ , obteniéndose un error instantáneo máximo a la salida del 9.8%.

No conviene olvidar que el aumento de  $N$  tiende a hacer la respuesta del sistema más oscilante, llegando a hacerla inestable para un cierto valor.



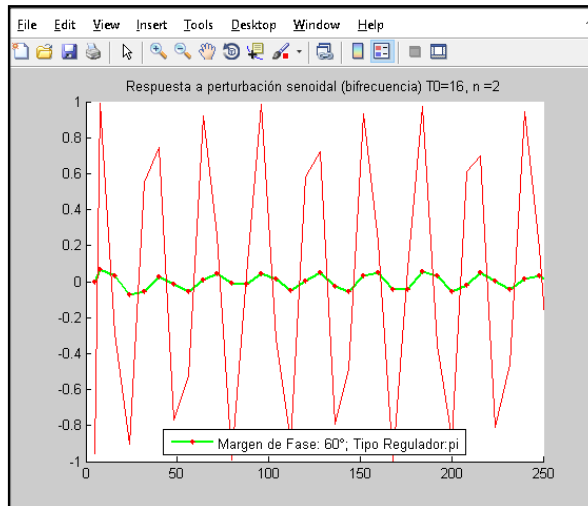
**Figura 50:** *Respuesta de  $G(s)$  a perturbación cuadrada ( $N=3$ ,  $\tau=0$ )*

Por último, para muestreo bifrecuencia ( $N=2$ ) con retardo  $\tau=5$  y frente a una perturbación tipo escalón unitario, la respuesta del sistema es la mostrada en la figura 51, que es peor (mayor sobreoscilación y tiempo de establecimiento) que la respuesta obtenida con  $\tau=0$ ; el error instantáneo máximo medido es del 12.8%, bastante elevado, aunque similar al medido con  $N=2$  y  $\tau=0$ .



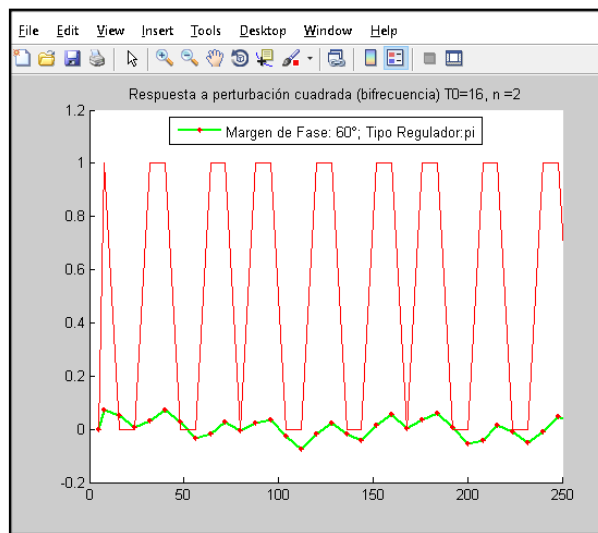
**Figura 51:** *Respuesta de  $G(s)$  a perturbación escalón ( $N=2$ ,  $\tau=5$ )*

Cuando la perturbación es senoidal y el retardo vale  $\tau=5$  y  $N=2$ , la respuesta del sistema es algo inferior al caso en que  $\tau=0$  según se puede ver en la figura 52. La señal perturbadora se atenúa a la salida del sistema, como se ha podido apreciar también en las representaciones gráficas de las simulaciones anteriores, con un error instantáneo máximo del 7.4%.



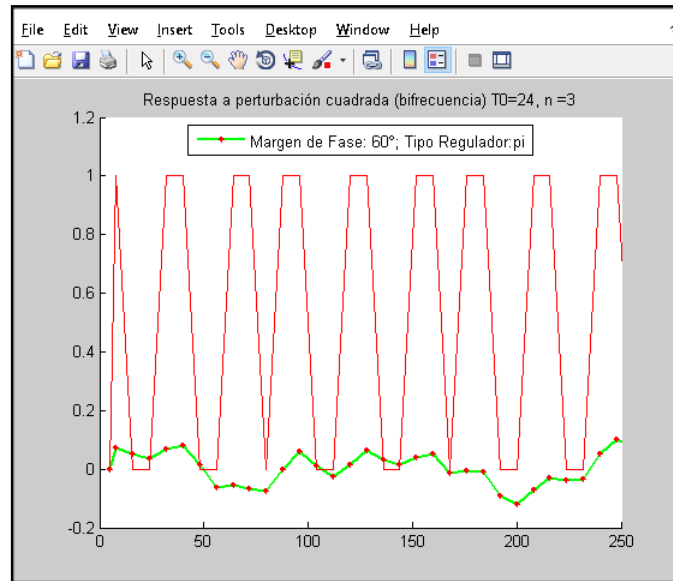
**Figura 52:** *Resposta a perturbación senoidal ( $N=2$ ,  $\tau=5$ )*

Aplicando ahora una perturbación cuadrada, la respuesta del sistema es muy similar a la senoidal, ya que la amplitud de la salida y el error instantáneo son parecidos; este último llega hasta el 7.5%, inferior aunque muy similar al medido en el caso en que  $\tau=0$ . Se observa una peor atenuación de la perturbación cuadrada cuando  $N=3$ , al aumentar la amplitud de la salida frente a la misma y el error instantáneo máximo de salida hasta el 12.7%, lo que se puede observar en la figura 54.



**Figura 53:** *Resposta a perturbación cuadrada ( $N=2$ ,  $\tau=5$ )*

En general, a la vista de las simulaciones realizadas, no puede afirmarse que el retardo empeore la respuesta del sistema a las perturbaciones, aunque si empeora la respuesta con el aumento de  $N$ .



**Figura 54:** *Respuesta a perturbación cuadrada ( $N=3$ ,  $\tau=5$ )*

Los resultados obtenidos se resumen en la tabla que se muestra a continuación, en la que se representa el tipo de perturbación, el retardo, la relación N (el caso monofrecuencia se representa en la tabla como  $N=1$ ), y el máximo error obtenido a la salida del sistema en esas condiciones.

Perturbación	Retardo (s)	N	Error (%)
Escalón	0	1	2.81
Senoidal	0	1	5
Cuadrada	0	1	5
Escalón	5	1	2.64
Senoidal	5	1	1.2
Cuadrada	5	1	2.35
Escalón	0	2	12.15
Senoidal	0	2	9.5
Senoidal	0	3	10.9
Cuadrada	0	2	8.3
Cuadrada	0	3	9.8
Escalón	5	2	12.8
Senoidal	5	2	7.4
Cuadrada	5	2	7.5
Cuadrada	5	3	12.7

#### 4.10. Código MATLAB

A continuación se va a explicar de manera breve el código MATLAB generado para obtener las respuestas y figuras anteriores. Este código comienza con la introducción de varios parámetros del sistema: margen de

## Sistemas con Muestreo Multifrecuencia

---

fase, tipo de regulador, simulación monofrecuencia o bifrecuencia, relación entre las dos frecuencias de muestreo utilizadas, etc.

Se calcula la función de transferencia del regulador y su ecuación de estado mediante las sentencias:

```
R = pidtune(G, pid_type);  
[A1c, B1c, C1c, D1c, E1c] = dssdata(R);
```

En donde *pid\_type* es un parámetro de tipo '*string*' que indica el tipo deseado de regulador; se ha trabajado fundamentalmente con reguladores PI; el par regulador planta utilizado en la simulación de la figura 38 ha sido el siguiente:

$$R(s) = 0.166 + \frac{0.005}{s}, \quad G(s) = \frac{10e^{-\tau s}}{(50s+1)(10s+1)}$$

Y para el sistema de primer orden de las simulaciones realizadas:

$$G(s) = \frac{0.25}{24s+1} e^{-\tau s}$$

El regulador calculado ha sido el siguiente:

$$R(s) = 1.89 + \frac{0.267}{s}$$

Posteriormente se calcula *h* y *T<sub>0</sub>*, y se discretiza el regulador con la sentencia:

```
[A1d, B1d, C1d, D1d] = c2dm(A1c, B1c, C1c, D1c, T0, 'zoh');
```

El sistema se modela mediante la función:

```
function xdot = fofd_model(-, x, sysC2, u)  
  
    xdot = sysC2.A*x + sysC2.B*u;  
  
end
```

Que permite calcular la respuesta del sistema mediante la función *ode45()*, ya que en el caso del muestreo bifrecuencia, no pueden utilizarse convenientemente las funciones que provee MATLAB para manipular sistemas (*ss()*, *series()*, *feedback()*...), pues no manejan adecuadamente

los estados adicionales que en la representación de ecuaciones de estado introduce el retardo del propio sistema.

Con ello, se calcula la respuesta del sistema realimentado mediante el siguiente fragmento de código:

```
% Inicialización de estados controlador y planta
xout = [];
tout = [];
x0 = zeros(np, 1);
u = 0;
y = sysC2.c*x0 + sysC2.d*u;
y0 = y;
if strcmp(pert, '1')
    yref = 1;
else
    yref = 0;
end
xc = zeros(nc, 1);
yout = y0;
ts = [];
contador = 0;
m = floor(tend/T0);

% Cálculo de la respuesta al escalón
for i1 = 1:m
    if i1 == 1
        xc = A1d*xc + B1d*(yref - y0);
        Um = C1d*xc + D1d*(yref - y0);
    else
        indice = 1 + fix(contador/h) - goback(tau, h, N);
        xc = A1d*xc + B1d*(yref - yout(indice));
        Um = C1d*xc + D1d*(yref - yout(indice));
    end

    for i2 = 1:N
        t0 = (i1 - 1)*T0 + (i2 - 1)*h;
        tf = (i1 - 1)*T0 + i2*h;
        u = Um(i2) + eval(fun)/sysC2.b;
        [t, x] = ode45(@(t, x) fofd_model(t, x, sysC2, u), [t0 tf], x0);
        x0 = x(end, :);
        tout = [tout; t(2:end)];
        xout = [xout; x(2:end, :)];
        ts = [ts contador];
        contador = contador + h;
        y = sysC2.c*x0 + sysC2.d*u;
    end
    if i2 == N
        yout = [yout y];
    end
end
y = sysC2.c*x0 + sysC2.d*u;
if i1 == m
    yout = [yout y];
end
end
```

La función  $goback(\tau, h, N)$ , calcula la entrada que debe de aplicarse a la realimentación del sistema en función del retardo  $\tau$  que aparece a la salida del sistema.

Para el caso de análisis de perturbaciones, el código anterior se modifica haciendo  $y_{ref} = 0$  y sumando la perturbación a la entrada mediante la función  $eval()$ , que permite evaluar una expresión que se ha seleccionado previamente, según se muestra en el fragmento de código siguiente, correspondiente a dicha funcionalidad:

```
try
    q = strjoin(strcat('Análisis de perturbaciones (1: No, 2: Escalon, 3: Senoidal, 4: Cuadrada):', {' ' }));
    pert = input(q, 's');
    if isempty(pert)
        pert = 'No';
    end
catch exception
    pert = 'No';
end
switch pert
    case '1'
        fun = '0';
    case '2'
        fun = '1';
    case '3'
        fun = 'sin(tf)';
    otherwise
        fun = '0.5*square(tf)+0.5';
end
```

Parte de los gráficos se han realizado también con el 'script' 'fotd\_w.m' usado en el capítulo 2, que se ha modificado ligeramente respecto del código mostrado en la figura 28 para poder trabajar con perturbaciones, y para su representación, este 'script' también se adjunta con este documento.

## Sistemas de Control Basados en Red

---

### 5.1. Introducción

El objetivo de este capítulo es simular en las condiciones más reales posibles un Sistema de Control Basado en Red (SCBR), donde se pueda apreciar el efecto que los retardos introducidos por la red tienen sobre las acciones de control aplicadas a un sistema de primer orden, y una posible solución a esta situación. A continuación se describe el contexto en el que el Sistema de Control Basado en Red bajo estudio desarrolla su actividad.

Una estructura lógica de los SCBR, es aquella en la que los nodos sensor y actuador están localmente en la planta, y el nodo controlador es remoto, conectado a los nodos sensor y actuador a través de una red de comunicaciones, de manera que el nodo controlador puede controlar varios bucles diferentes; este es el caso que se estudiará en este capítulo.

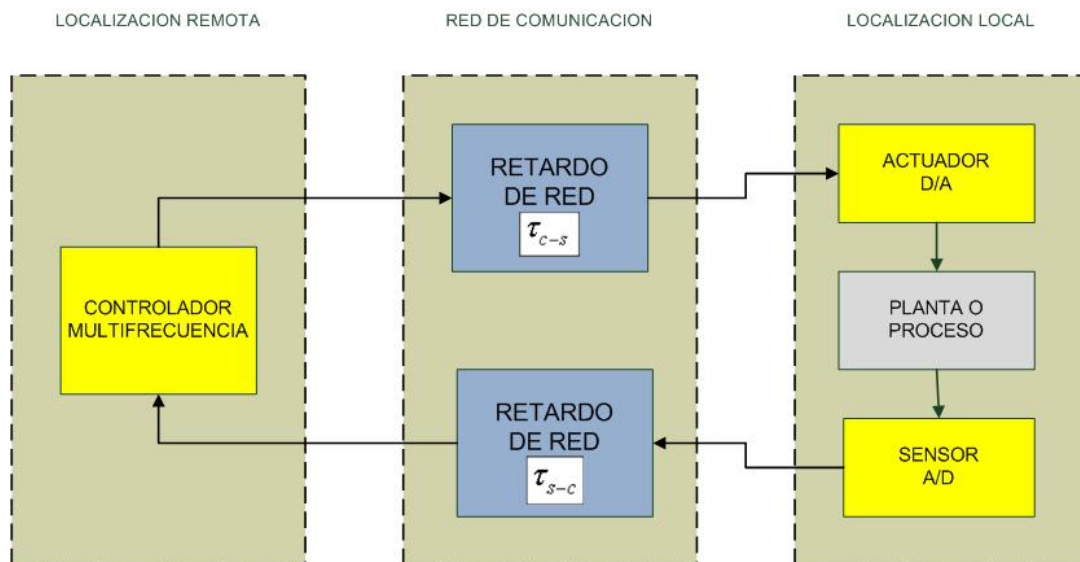
Con esta configuración, es claro que las medidas de la salida de la planta que se desea controlar obtenidas por el sensor llegarán al controlador pasado un periodo de tiempo, en función de la disponibilidad de la red, su velocidad de transmisión, etc.; es decir: llegarán al controlador con un retardo, en principio, variable y aleatorio aunque si se asume sincronización de todos los elementos del bucle de control, medible; lo mismo sucederá con las acciones de control calculadas por el controlador con destino al actuador de la planta, como ya se ha mencionado.

El sensor muestrea la salida de la planta con un periodo  $T$ , es decir: el muestreo del sensor depende únicamente del tiempo; sin embargo, el controlador solo puede calcular la acción de control cuando ha recibido los datos necesarios del sensor, lo que ya hemos visto ocurre con un retardo variable por la propia naturaleza del SCBR, es decir: el controlador está actuando por eventos, siendo estos eventos las llegadas de los mensajes del sensor.

Como puede verse, tampoco puede hablarse aquí de sistemas multifrecuencia (salvo en algunos casos específicos en los que la parte derivativa del regulador general PID es local a la planta, y el controlador ejecuta la parte PI, como en [Salt-07] y [Cuenca-10]).

## Sistemas de Control Basados en Red

Ahora los datos viajan del sensor al controlador y del controlador al sensor a través de una red de comunicaciones que provoca que la información llegue a su destino con unos retardos en general variables y aleatorios ( $\tau_{s-c}$  y  $\tau_{c-a}$ ), aunque supondremos que en todo caso estos retardos son inferiores al periodo de muestreo. La figura 55 muestra un diagrama de bloques que representa los elementos descritos que integran un SCBR.



**Figura 55:** *Elementos de un Sistema de Control Basado en Red*

Elegiremos en este capítulo para las simulaciones un controlador PID por ser el más utilizado en la práctica con buenos resultados. La estructura y parámetros de dicho regulador, para luego programar el regulador fácilmente en MATLAB, es la indicada en [Wittenmark-02].

Esta red de comunicación puede ser por ejemplo Ethernet cableada o inalámbrica, aunque el modelo es útil para cualquier otra; de hecho en las simulaciones realizadas, se ha usado una red CAN, por ser una de las más extendidas en el ámbito industrial y disponer de control de acceso por prioridades.

Podemos suponer además que el nodo controlador, además de llevar a cabo los cálculos pertinentes para el bucle de control en el que estamos interesados, ejecuta otras tareas (que bien pudieran ser el control de otros bucles de la misma o diferentes plantas), y que por tanto tarda un tiempo  $\tau_c$  en atender al bucle y realizar los cálculos correspondientes, de manera que el tiempo total  $\tau_r$  transcurrido desde que el sensor envía una medida hasta que el actuador ejecuta la acción de control es:

$$\tau_r = \tau_{s-c} + \tau_c + \tau_{c-a}$$

Es importante destacar aquí que con esta notación, el retardo de intrínseco de la planta se representa por  $\tau$ .

### 5.2. La red CAN (Controller Area Network)

Como se ha mencionado anteriormente, para simular la situación descrita en el punto anterior se supondrá que los nodos sensor, actuador y controlador están intercomunicados a través de una red CAN, por ser una de las más ampliamente utilizadas en la industria en general, y dentro de ella, en el sector de automoción en particular. El bus CAN [Bosch-91] es un protocolo de transmisión de datos sobre un medio físico balanceado, diseñado por BOSCH junto con INTEL en los años 80, con la idea de poder enlazar cualquier equipo electrónico con otro usando un sistema de comunicación serie barato y seguro.

Este proyecto, aparentemente con poco futuro, cobraba cada vez más valor e importancia al permitir interconectar muchos dispositivos entre sí aportando un extra de fiabilidad y seguridad, además de su reducido coste en comparación con los antiguos sistemas cableados, que necesitaban grandes longitudes de cable. El primer vehículo en usar este sistema de serie fue el Mercedes-Benz W140, que disponía de cinco nodos de comunicación interconectados por este bus.

En la figura 56 puede verse la estructura de la trama o datagrama según el protocolo CAN, que se resume a continuación.

Cabecera:

- Campo de comienzo (SOF): Este campo tiene un bit, y es el comienzo del datagrama, un aviso de que hay un mensaje.
- Dirección (SID): Este campo indica la dirección del nodo que transmite y recibe en la red, y además se usa para determinar la prioridad del nodo en cuestión (cuantos más ceros tenga la dirección en su comienzo, mayor es la prioridad asignada al nodo que transmite). Este campo tiene 11 bits.
- Campo de control (IDE+r0+DLC). Antes de este campo existe un bit libre (RTR), en blanco para diferenciar los campos 2 y 3, en este campo se especifican los datos que van a recibir las unidades de mando, para que las unidades se aseguren los datos que están recibiendo, dando fe que tienen todos los datos. Este campo consta de 6 bit como máximo.

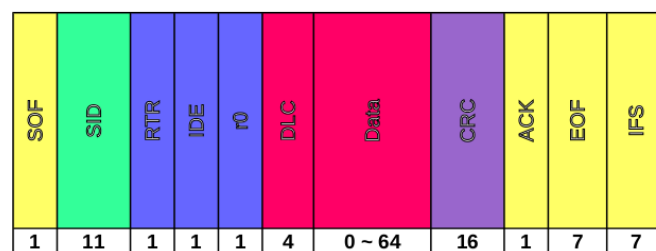
Datos:

## Sistemas de Control Basados en Red

- Campo de datos (DATA): en este campo los nodos envían la información a través del bus. Consta de 64 bit.

Cola:

- Campo de Control (CRC): Sirve para determinar la integridad de la trama. Consta de 16 bit.
- Confirmación (ACK): si hay algún fallo de transmisión en este campo el nodo receptor avisa del fallo para que vuelva a enviar la información el nodo emisor; tiene 2 bit.
- Final de trama (EOF): 7 bits para indicar la terminación del mensaje.



**Figura 56:** *Trama CAN*

En total, cada trama CAN está compuesta por 108 bits, de los que un máximo de 64 son datos; por tanto la sobrecarga de protocolo es de 44 bits (en el caso de usar los 64 bits de datos útiles).

Las velocidades de transmisión utilizadas en un CAN bus dependen de la longitud en metros de dicha red; en las simulaciones se supondrá una velocidad de 115,2 kbps, velocidad para la que la red puede tener una longitud de hasta 500 m, longitud que permite simular una red local en una localización industrial estándar. Pueden encontrarse más detalles sobre CAN bus en <http://www.ni.com/white-paper/2732/es/>, aunque existen muchas publicaciones abordando este bus de comunicaciones, por su uso extenso en automoción, principalmente.

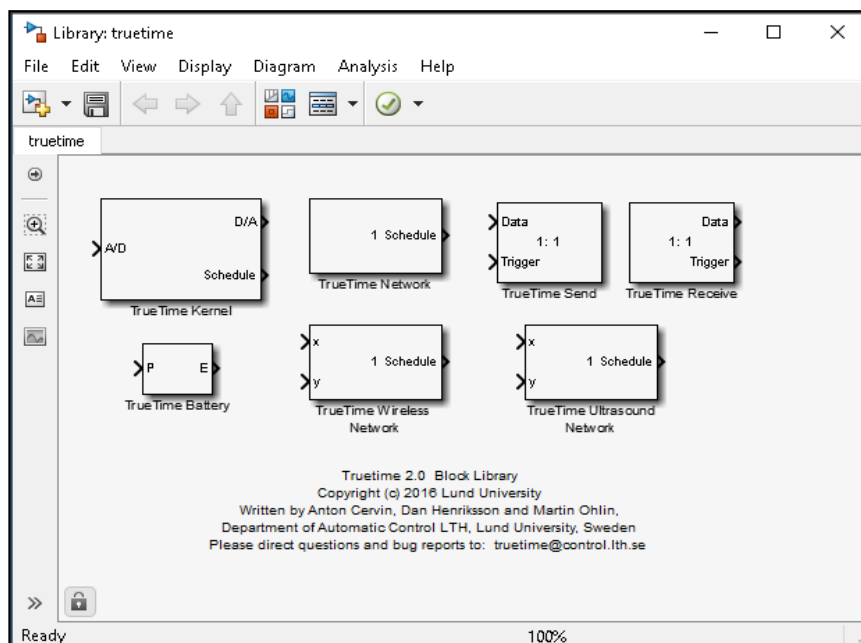
### 5.3. *Toolbox TrueTime* para MATLAB

Para llevar a cabo la simulación del SCBR descrito más arriba se utilizará como hasta ahora MATLAB, y la *Toolbox TrueTime* desarrollada a este efecto por la Universidad de Lund (Suecia). *TrueTime* es un simulador para sistemas de control en tiempo real. *TrueTime* facilita la simulación de tareas de control ejecutándose en *kernel*s en tiempo real, redes de transmisión de datos y sistemas dinámicos de control. Puede encontrarse información más amplia en la página Web de la universidad de Lund, donde se desarrolló la *Toolbox*: <http://www.control.lth.se/truetime/>.

## Sistemas de Control Basados en Red

Está escrito en C++ MEX, utiliza la simulación basada en eventos y las interrupciones externas, permitiendo escribir código para 'tareas' mediante ficheros '.m', funciones C++ o llamadas directas a bloques de SIMULINK desde el código de las funciones utilizadas.

*TrueTime* por otra parte, permite simular las redes más utilizadas en la industria: *Ethernet*, CAN, TDMA (FlexRay), FDMA, *Round Robin* o *switched Ethernet*, así como redes inalámbricas (WLAN y ZigBee), e incluso dispositivos alimentados por baterías. Se puede afirmar que *TrueTime* es una librería de bloques de simulación que extiende la usabilidad de MATLAB / SIMULINK para simular SCBR.



**Figura 57:** *Librería de TrueTime*

Los modelos realizados con *TrueTime* tienen tres elementos clave que están casi siempre presentes en todos los modelos de simulación: el *kernel TrueTime* (que simula computadores, dispositivos E/S, sistemas embebidos...), La red *TrueTime* representa el modelo de red de transmisión de datos utilizada, y el proceso o planta a controlar, que se muestran en la figura 57. Existe también un bloque *TrueTime* para simular una batería, que permite simular diferentes tipos de funcionamiento de los bloques cuando la carga de la batería está baja. El bloque *TrueTime kernel* es responsable de las E/S necesarias para la simulación, de la conexión con la red *TrueTime* y del procesamiento de datos o ejecución de cálculos, siendo por tanto el elemento principal de este modelo, pudiendo ejecutar algoritmos, servicio de interrupciones y otras muchas actividades en base a ficheros '.m' estándar de MATLAB, como tareas que pueden ser periódicas, por eventos,

etc. El bloque de Red de *TrueTime* modela los diferentes tipos de red que pueden simularse con la *Toolbox*, y que ya se han mencionado anteriormente.

### 5.4. Descripción Detallada de los bloques *TrueTime*

#### Bloque *TrueTime Kernel*:

Como ya se ha indicado, este bloque proporciona la lógica de control y el procesamiento de datos necesario para el modelo del sistema que se esté analizando. Es necesario establecer los parámetros más importantes de configuración de este bloque inicialmente:

- Función de inicialización: define el número y nombres de los ficheros '.m' que codifican las funcionalidades de las tareas e interrupciones.
- Argumentos de entrada: son los parámetros que se le pasan al kernel para controlar y personalizar su funcionamiento.
- N° de E/S analógicas: define el número de este tipo de entradas / Salidas.
- N° de nodo: indica el número de nodo que ocupa en la red y el número de red (puede haber más de una red en una sola simulación).
- Drift y offset: valores que permiten definir velocidades de reloj más rápidas que el tiempo real base por una parte, y por otra parte diferencias en el arranque del bloque respecto de dicho tiempo real base.

#### Redes *TrueTime*

Este bloque simula la transferencia de paquetes de información entre los nodos a él comentados (que son los bloques *kernel*). Cuando un nodo necesita transferir un mensaje se activa una señal de 'transferencia comenzada' en el canal apropiado de este bloque, y el paquete de datos es transferido al nodo receptor, cuando la transferencia del paquete de datos se ha concluido, se activa una señal de 'transferencia concluida' en el canal de salida. Los paquetes transferidos se almacenan en un buffer del nodo receptor. Los parámetros de configuración de este bloque son:

- N° de Red: permite identificar la red en un modelo, pues puede haber más de una.
- N° de nodos: es el número de nodos conectados a la red.
- Velocidad de transmisión: velocidad de transmisión de datos en bps.
- Mínimo tamaño de paquete: número mínimo de bits a transmitir con cada paquete. Si se transfiere un paquete de menor tamaño que este valor, se rellena el paquete hasta llegar a este valor.
- Probabilidad de pérdida: es la probabilidad de que el paquete una vez enviado, no llegue a su destino.

### 5.5. Análisis y Simulación de un SCBR

El sistema que se simulará en este capítulo es el mismo que se ha utilizado en los anteriores:

$$G(s) = \frac{0.25}{24s+1} e^{-\tau s}$$

Como ya se ha indicado, este sistema de primer orden con retardo representa según [Campo-12] con bastante exactitud la variación de presión en una columna de destilación en función de la actuación de una válvula de control de vapor. Este es un sistema 'lento', cuya respuesta se alarga varios segundos, dependiendo del valor del retardo, que se variará en la simulación.

En este modelo de simulación se han considerado tres nodos, la red de comunicaciones (CAN bus) y la planta; el modelo se puede ver en la figura 58. En la misma se pueden ver los nodos que intervienen en el sistema, la planta (a la que se le añade un retardo seleccionable mediante un interruptor), y la referencia, que se materializa con un escalón unitario de subida y otro de bajada complementario. Puede introducirse en el sistema como señal de referencia un tren de impulsos, que simulan la aplicación continua de escalones unitarios de subida y bajada al sistema.

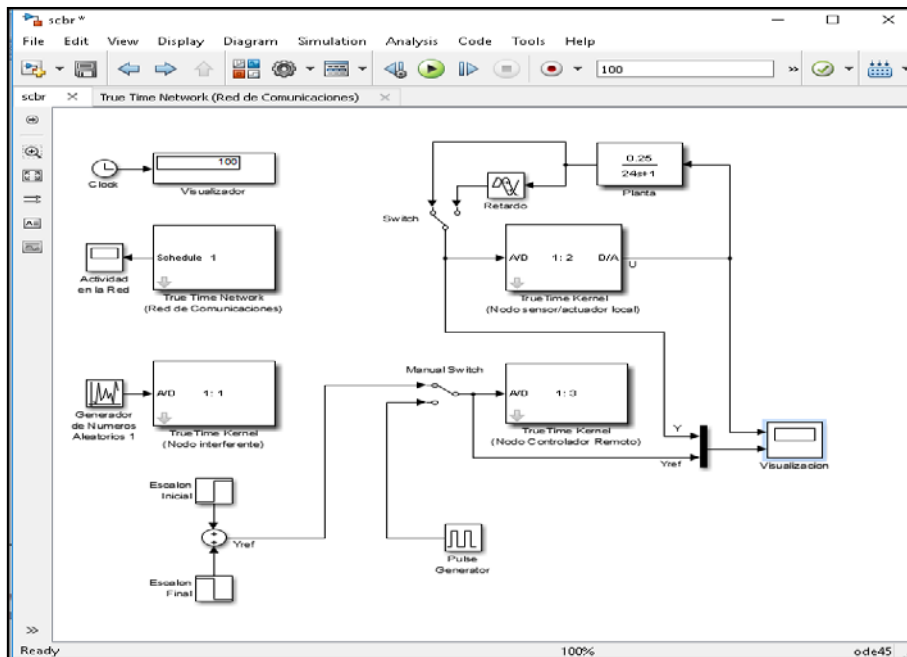
Uno de los nodos es el nodo controlador (nodo 3), que suponemos remoto respecto de la planta, y conectado al sensor y actuador a través del bus CAN, siendo estos últimos locales a la planta.

El nodo controlador incluye una tarea 'interferente' de alta prioridad denominada '*dummy\_task*', concebida para poder analizar posteriormente el efecto de dicha tarea en la ejecución del bucle de control. Esta tarea bien puede simular una tarea relacionada con otro bucle de control, o simplemente una aplicación 'interferente' que deba de ser realizada por el controlador, aún sin estar relacionada con el sistema que se desea controlar.

La función de inicialización de dicho nodo puede verse en la figura 59, y las líneas correspondientes a la tarea de alta prioridad (de hecho, la máxima prioridad posible) antes indicada, se definen en las dos últimas líneas del citado código. En este fichero '.m' de inicialización se crea una estructura denominada '*data*' donde se almacenan los valores anteriores del regulador a memorizar, y los parámetros correspondientes al regulador PID cuyos valores a enviar al nodo actuador se calculan en este nodo.

El resto de las líneas de código son fáciles de entender en base al contexto ya descrito, y los comentarios insertados en el código.

# Sistemas de Control Basados en Red



**Figura 58:** Configuración del SCBR simulado con TrueTime

```
function controller_init(arg)

% Sistema de Control Remoto
%
% Se reciben los mensajes del nodo sensor, se calcula la señal de control y se
% envía al nodo actuador. Los nodos sensor y actuado son locales a la Planta.
% Se introduce también una tarea correspondiente a un nodo interferente de alta
% prioridad.

% Inicialización del kernel del simulador TrueTime.
% Prioridad Fija.
ttInitKernel('prioFP')

% Generación del 'buffer' de datos (en memoria local)
data.h = 8; % periodo de muestreo de 8 s.

data.K = 0; data.K_0 = 14.0; data.K_1 = 12.0;
data.Ti = 0; data.Ti_0 = 27.5; data.Ti_1 = 32.0;
data.Td = 0; data.Td_0 = -5.0; data.Td_1 = -9.0;
data.N = 0; data.N_0 = 50.0; data.N_1 = 250.0;

data.yold = 0;
data.Dold = 0;
data.Iold = 0;
data.u = 0;

% Esta tarea del controlador solo se ejecuta cuando se reciben datos del
deadline = data.h;
ttCreateTask('controller_task', deadline, 'controller_code', data);
ttAttachNetworkHandler('controller_task')
ttSetPriority(2, 'controller_task');

% Tarea periodica interferente de alta prioridad. arg(1) indica el periodo
% de dicha tarea; arg(2) el tiempo de ejecución de la misma, que se pasa a
% dummy_code.m
starttime = 0.0;
period = arg(1);

if arg(2) > 10
    data1 = arg(1);
else
    data1 = arg(1)*arg(2)/10;
end
ttCreatePeriodicTask('dummy_task', starttime, period, 'dummy_code', data1);
ttSetPriority(1, 'dummy_task');
```

**Figura 59:** Código de inicialización del nodo controlador

El código de la tarea '*controller\_code*' se muestra en la figura 60.

```
function [exectime, data] = controller_code(seg, data)
global z
switch seg
case 1
    % Lectura del valor del sensor
    y = ttGetMsg;
    if isempty(y)
        disp('Error: Mensaje No Recibido');
        y = 0.0;
    end
    % Lectura del valor de referencia
    r = ttAnalogIn(1);

    % Calculo de las constantes del regulador en función del retardo
    data = interpolate(data, z);

    % Cálculo de los términos del regulador P, PI, PD ó PIDF
    P = data.K*(r-y);
    I = data.Iold;
    D = data.Td/(data.N*data.h+data.Td)*data.Dold+
        data.N*data.K*data.Td/(data.N*data.h+data.Td)*(data.yold-y);

    if data.Ti <= 0 && data.Td <= 0
        data.u = P;
    elseif data.Ti <= 0 && data.Td ~= 0
        data.u = P + D;
    elseif data.Td <= 0 && data.Ti ~= 0
        data.u = P + I;
    else
        data.u = P + I + D;
    end
    % Guardo en memoria valor parte integral (si es regulador PI ó PID)
    if data.Ti > 0
        data.Iold = data.Iold + data.K*data.h/data.Ti*(r-y);
    end
    % Guardo en memoria resto de términos del regulador
    data.Dold = D;
    data.yold = y;
    exectime = 0.0005;
case 2
    % Envío el mensaje CAN (108 bits en total)
    ttSendMsg(2, data.u, 108);
    exectime = -1;
end
```

**Figura 60:** Código de la tarea '*controller\_code*' del nodo controlador

Se tiene también el nodo sensor / actuador (nodo 2), local a la planta, donde la parte sensora lee los valores de la variable controlada periódicamente, en función únicamente del tiempo; este pasa los datos leídos al nodo controlador, y a su vez el nodo controlador pasa las acciones del actuador al nodo sensor / actuador de nuevo, ejecutando dichas acciones de control. El código de inicialización del nodo sensor / actuador puede verse en la figura 61, y el código de las tareas del sensor ('*sensor\_code*') y actuador ('*actuator\_code*') en las figuras 62 y 63. Se ha incluido un nodo adicional (nodo 1) denominado '*interference\_node*' cuya función es introducir tráfico 'parásito' adicional en la red, entendiendo por 'parásito' tráfico no relacionado con el bucle de control, de manera que el esquema de la figura 58 no solo simula el bucle de control, sino también la tarea interferente del nodo controlador (nodo 3), y las comunicaciones 'parásitas' adicionales a través de la red generadas por el nodo interferente (nodo 1). En la figura 64 se muestra el código de inicialización del nodo interferente, y en la figura 65, el código de la tarea interferente '*interference\_task*'.

```
function sensor_actuator_init
global t
t = [];

% Sistema de Control REmoto: Nodo sensor / actuador.
% Muestra la variable controlada de la planta (y) periodicamente
% y envia los datos al nodo de control. Aplica la acción de control
% recibida por el controlador remoto.

% Inicialización del kernel del simulador TrueTime. Prioridad Fija
ttInitKernel('prioFP');

% Tarea periodica del sensor
starttime = 0.0;
period = 8;
ttCreatePeriodicTask('sensor_task', starttime, period, 'sensor_code');
ttSetPriority(2, 'sensor_task');

% Activación de la tarea del actuador, cuando se reciben datos
% correspondiente a las acciones de control calculadas por el
% controlador
deadline = 0;
ttCreateTask('actuator_task', deadline, 'actuator_code');
ttSetPriority(2, 'actuator_task');
ttAttachNetworkHandler('actuator_task')
```

**Figura 61:** Código de inicialización del nodo sensor / actuador

```
function [exectime, data1] = sensor_code(seg, data1)
persistent y

switch seg
case 1
    y = [ttAnalogIn(1) ttCurrentTime];
    exectime = 0.0005;
case 2
    % Se envían 108 bits al nodo 3 (controlador)
    ttSendMsg([1 3], y, 108, 2);
    exectime = 0.0004;
case 3
    exectime = -1;
end
```

**Figura 62:** Código de la tarea 'sensor\_code' del nodo sensor / actuador

```
function [exectime, data1] = actuator_code(seg, data1)
global t
persistent u

switch seg
case 1
    u = ttGetMsg;
    exectime = 0.0005;
case 2
    if ~isempty(u)
        ttAnalogOut(1, u(1));
        t = [t ttCurrentTime - u(2)];
    else
        disp('Error: Recepción de Mensaje Vacío')
    end
    exectime = 0.0;

otherwise
    exectime = -1;
end
```

**Figura 63:** Código de la tarea 'actuator\_code' del nodo sensor / actuador

Más detalles del funcionamiento de *TrueTime* y de la codificación de tareas, interrupciones, mensajes, etc. que esta *Toolbox* soporta, pueden verse en [Henriksson-03] y [Chvostek-06], así como en el manual de la propia *Toolbox*.

```
function interference_init(arg)

% Sistema de Control Remoto: Nodo interferente.
% Generar trafico parásito de ancho de banda arg
% Inicialización del kernel del simulador TrueTime
% Prioridad Fija
ttInitKernel('prioFP');

% Tarea interferente
period = arg(1);
offset = 0;
data = arg(2);
ttCreatePeriodicTask('interference_task', offset, period, 'interference_code', data);
```

**Figura 64:** *Código de inicialización del nodo interferente*

```
function [exectime, data] = interference_code(seg, data)

global contador1 contador2
contador2 = contador2 + 1;
BWshare = data;
ran = ttAnalogIn(1);
if ran < BWshare
    % Autoenvío de 108 bits
    k = round(1+2*rand);
    ttSendMsg([1 1], 1, 108*k, 1);
    if k == 1
        contador1 = contador1 + 1;
    else
        contador1 = contador1 + 2;
    end
end

end
```

**Figura 65:** *Código de la tarea 'interference\_code', nodo interferente*

El generador de números aleatorios del nodo interferente sirve para garantizar el envío aleatorio de un número de tramas a través de la red que actúan como tramas interferentes a las propias de control del bucle de interés. Su uso permite simular cargas variables en la red de comunicaciones, tal y como sucedería en una red real, en función de las circunstancias de funcionamiento de la planta.

Finalmente, se visualizan dos canales: por una parte la señal de acción de control  $u(t)$  y por otra el escalón que actúa como señal de referencia  $y_{ref}$  y la respuesta del sistema al escalón de referencia:  $y$ .

Los parámetros de partida del regulador PID con los que se trabaja corresponden a  $\tau = 0$ , y según [Wittenmark-02] son:

- data.k = 14.0
- data.Ti = 27.5
- data.Td = -5.0
- data.N = 50.0

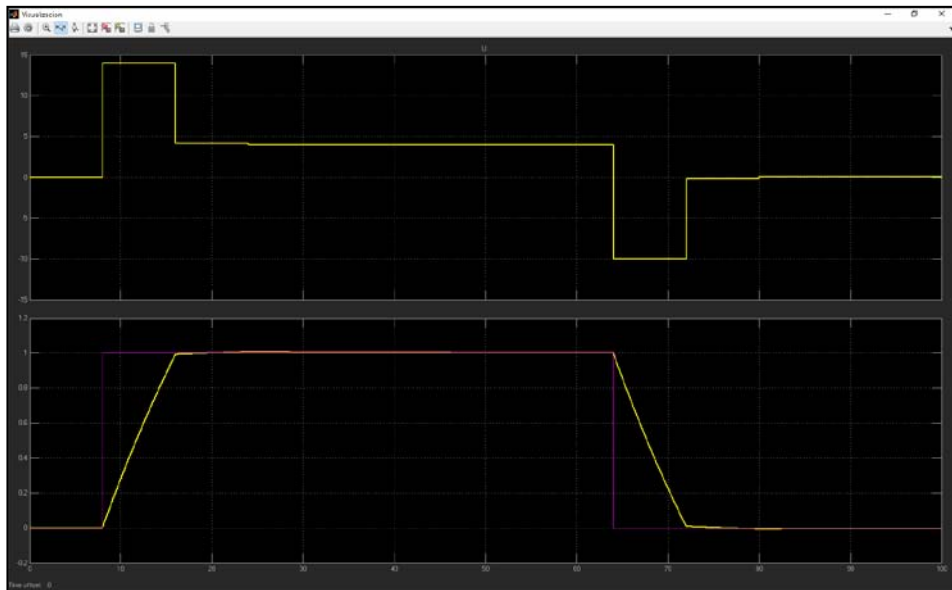
## 5.6. Simulaciones realizadas

Han sido varias las simulaciones realizadas, reproduciendo muchas de las situaciones que pueden darse en un caso real. A continuación se muestran los resultados obtenidos en las mismas.

### 5.6.1. SCBR sin interferencias ni tareas parásitas ( $\tau = 0$ )

Esta simulación se basa en la inexistencia de actuación por parte del nodo interferente, para lo cual se debe de poner a cero el segundo argumento de la función de inicialización del nodo interferente de la figura 64, y sin tarea de alta prioridad en el nodo controlador, lo que se consigue comentando las dos últimas líneas de la función de inicialización del nodo controlador mostrado en la figura 59. La planta tampoco presenta retardos, es decir: ( $\tau = 0$ ).

La planta a controlar, como ya hemos indicado, representa un reactor químico con un cierto retardo de transporte, muestreado cada  $h=8$  s, y aplicando un escalón de subida primero en  $t=9$  s. y otro de bajada superpuesto al anterior en  $t=64$  s, de manera que se produce la subida unitaria en  $t=9$  y la vuelta otra vez a cero de la señal de referencia en  $t=64$  s. La gráfica obtenida de  $u$  e  $(y, y_{ref})$  es la mostrada en la figura 13. La gráfica superior corresponde a  $u$  y la inferior a  $y, y_{ref}$ .



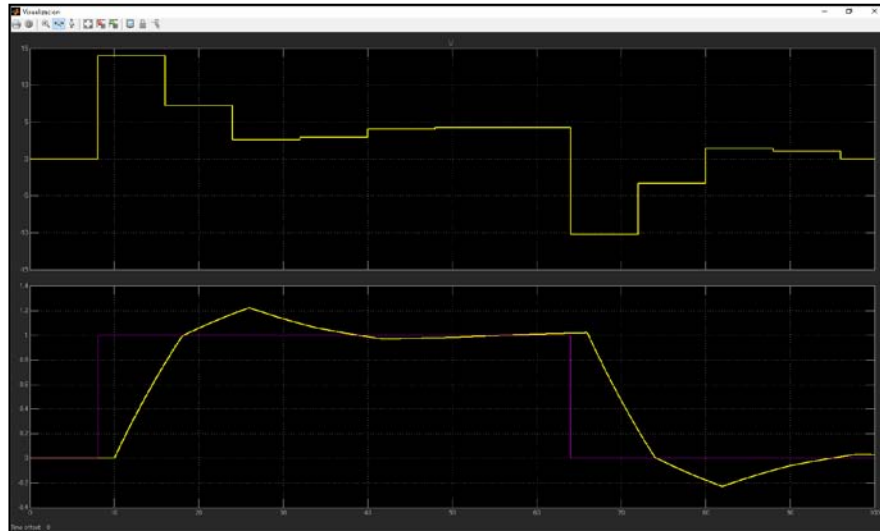
**Figura 66:** *Respuesta de  $G(s)$  sin interferencias ni retardos*

Puede verse que el sistema sigue al escalón aplicado sin ninguna sobreoscilación, y en un tiempo de establecimiento  $t_s = 15$  s.

### 5.6.2. SCBR con retardo intrínseco ( $\tau = 2$ s.)

En este caso, no hay ni interferencias en la red ni tareas parásitas ejecutándose en ninguno de los nodos de la red, pero el sistema presenta un retardo de transporte  $\tau = 2$  s; puede

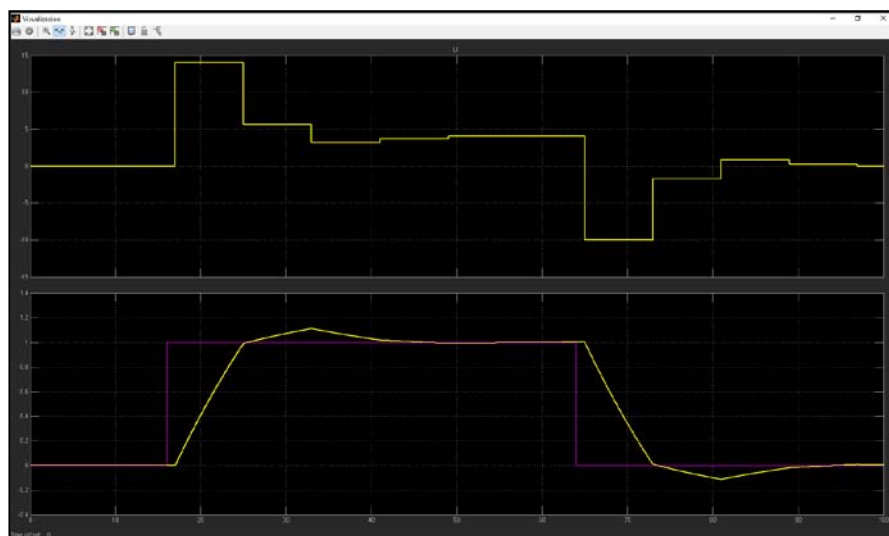
verse en la figura 67 como la respuesta del sistema empeora de manera visible: aparece una sobreoscilación del 22%, el tiempo de establecimiento se alarga hasta  $t_s = 35$  s. Estos resultados son compatibles con lo que ya se había visto a lo largo de este trabajo: los retardos del sistema empeoran la respuesta del mismo.



**Figura 67:** *Respuesta de  $G(s)$  sin interferencias ( $\tau=2$  s)*

### 5.6.3. SCBR con tarea adicional en el nodo controlador ( $\tau=0$ )

En este caso no existe el retardo de la planta, pero el nodo controlador, además de ejecutar el bucle de control, realiza una tarea que le ocupa el 50% del ancho de banda. La respuesta del sistema puede verse en la figura 68.

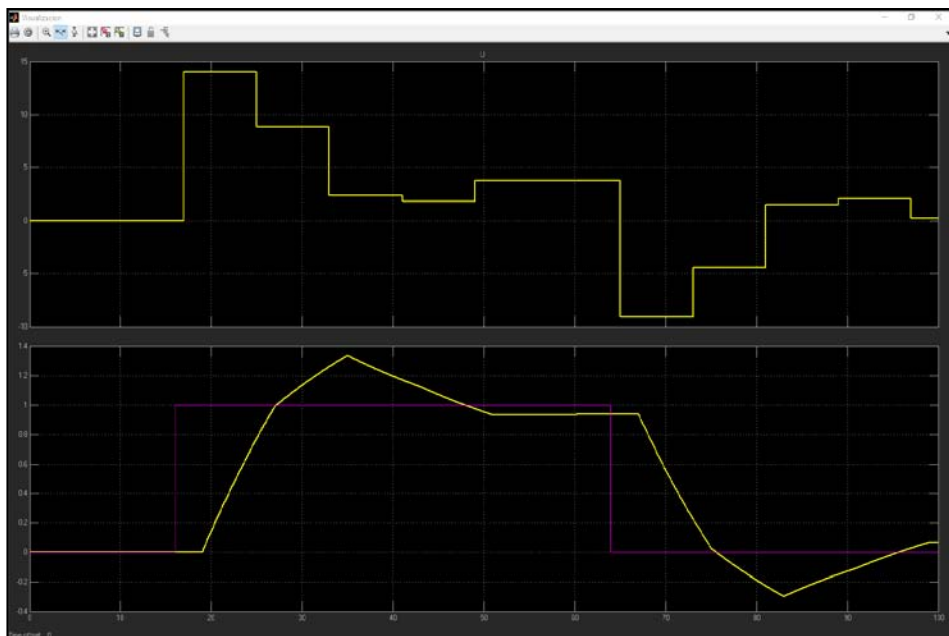


**Figura 68:** *Respuesta de  $G(s)$  con tarea interferente ( $\tau=0$  s)*

Se puede apreciar que dicha respuesta es muy similar a la de la figura 67, cuando era la planta la que presentaba un retardo intrínseco; se puede ver también que la respuesta del sistema se inicia con un retardo de aproximadamente 1.5 s., que aparece una sobreoscilación del 12% y que el tiempo de establecimiento se alarga hasta los 34 s.

### 5.6.4. SCBR con tarea en el nodo controlador y retardo ( $\tau = 2$ )

En este caso, la respuesta del sistema se muestra en la figura 69, en la que se ve que el sistema responde con un retardo aún mayor de 3.5 s., que la sobreoscilación sube hasta el 37%, que el tiempo de establecimiento es de 47.5 s. y que el sistema no alcanza el valor final del escalón durante el tiempo en el que este está aplicado.

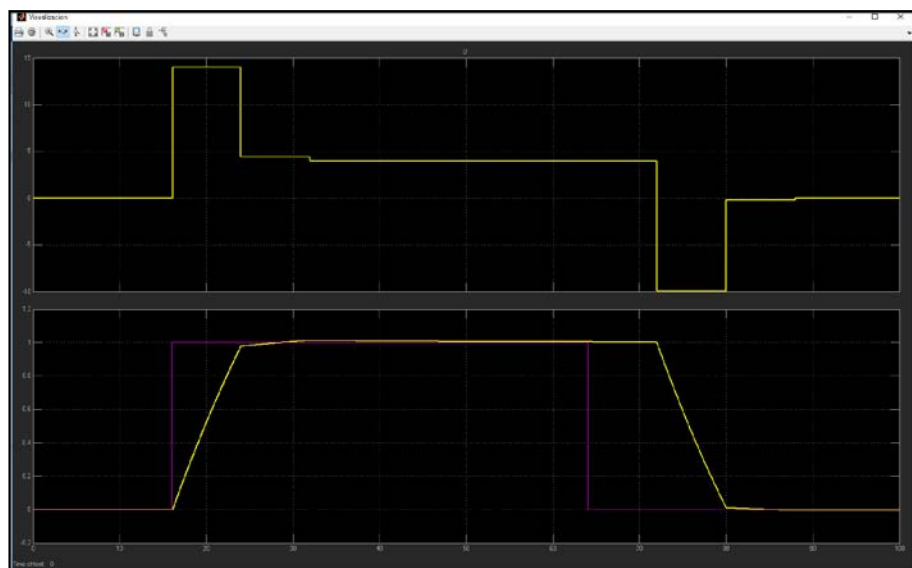


**Figura 69:** *Respuesta de  $G(s)$  con tarea interferente ( $\tau = 2$  s)*

### 5.6.5. SCBR con tráfico adicional en red ( $\tau = 0$ )

Como ya se ha indicado anteriormente, el nodo interferente tiene conectada a su entrada analógica un generador de números aleatorios, de manera que cuando el número generado es menor de un cierto umbral (fijado por el argumento de la función de inicialización del cuadro de dialogo de los parámetros del bloque kernel correspondiente), se envían con la máxima prioridad entre 1 y 2 paquetes de 108 bits; es decir: en ese caso se produce un envío a través de la red de entre 1 y 2 tramas CAN, simulando de este modo tráfico en red adicional al del bucle de control bajo análisis.

Fijando el umbral antes indicado a 0.25, y haciendo que la tarea de envío de tramas interferentes tenga una frecuencia de 1 kHz (lo que indica que esta tarea intentará enviar entre 1000 y 2000 tramas interferentes por segundo, en función del número aleatorio generado y del umbral fijado, y que un 25% del tiempo se envían tramas adicionales a través de nuestro bus CAN), se puede observar en nuestro sistema de control un comportamiento anormal en la bajada del escalón al valor inicial, en forma de un retardo significativo de unos 15 s. adicionales, como puede apreciarse en la figura 70.



**Figura 70:** *Respuesta de  $G(s)$  con tramas interferentes ( $\tau=0$ )*

Durante los 100 segundos que ha durado la simulación, se han enviado un total de 43779 tramas interferentes.

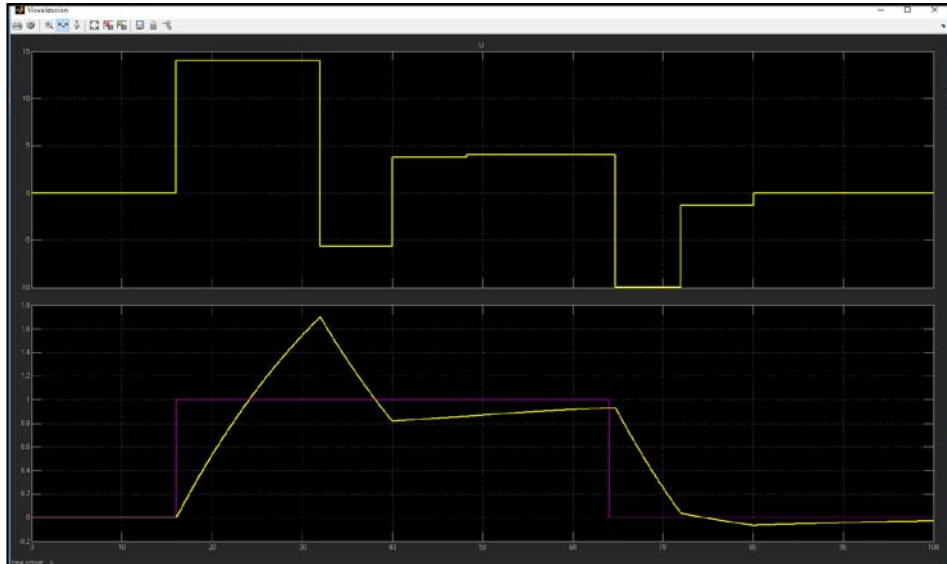
Aumentando la probabilidad de envío de tramas al 40%, sí puede apreciarse una degradación importante en la respuesta del sistema, como se muestra en la figura 71.

Si ahora se repite la misma simulación, pero incluyendo un retardo en el sistema  $G(s)$  de  $\tau = 2$  s, se obtiene la respuesta de la figura 72, que como puede verse empeora sensiblemente, tanto en el momento del inicio del escalón, como en la vuelta del mismo al valor inicial.

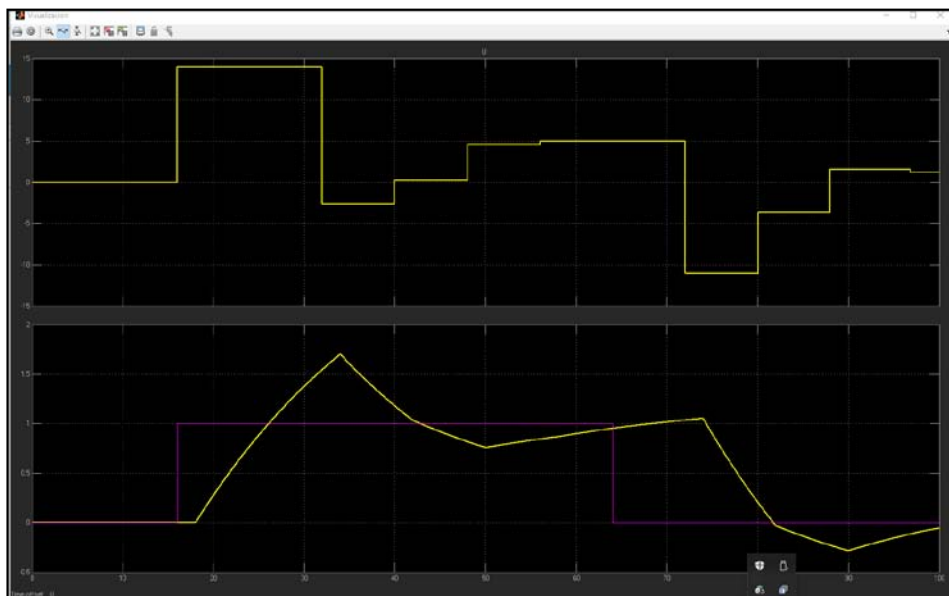
Con estos parámetros de simulación, durante los 100 segundos de duración de la misma, se han enviado a través de la red CAN simulada un total de 70733 tramas interferentes.

## Sistemas de Control Basados en Red

Se ha observado también que el envío de tramas interferentes, junto con la ejecución en el nodo controlador de una tarea interferente, produce efectos similares en el sistema, aunque lógicamente mucho antes, al sumarse ambos efectos negativos.



**Figura 71:** *Respuesta de  $G(s)$  con tramas interferentes ( $\tau=0$ )*



**Figura 72:** *Respuesta de  $G(s)$  con tramas interferentes ( $\tau=2$ )*

En definitiva, las simulaciones realizadas demuestran, como ya se había adelantado en el punto 1, que los SCBR sufren unos retardos adicionales por las circunstancias de su concepción que dificultan y empeoran la controlabilidad de los mismos, aún

siendo simples (como en este caso), y que es difícil predecir cuándo tareas adicionales de los nodos involucrados o tráfico adicional en la red afectarán a nuestro sistema, pues depende del tipo de actividad que estén realizando los nodos, o de la intensidad de tráfico (aleatoria) que soporte la red.

### 5.7. Solución propuesta

Si en el proceso de diseño del regulador utilizado en el SCBR se conociera de antemano el retardo que se van a producir en la red, sería posible optimizar este, y por lo tanto mitigar, si no compensar totalmente, los efectos de dicho retardo en la planta, y en particular en la variable controlada de la misma.

Lo que se propone en este trabajo, es incluir en los mensajes una marca de tiempo, de manera que el nodo actuador conozca uno de los componentes del valor total del retardo, y a partir de esta pueda estimar el retardo total producido en la red:

$$\tau_r = \tau_{s-c} + \tau_c + \tau_{c-a}$$

Y sin modificar el tipo de regulador utilizado (PI, PID...), si pueda adaptar los parámetros característicos del mismo ( $k_p$ ,  $k_i$ ,  $k_d$ ...). Para ello, se calculan los parámetros del regulador suponiendo que no hay ni tareas ni tramas interferentes en los nodos del bucle de control, y fijando retardos conocidos en la planta. De esta manera se obtiene una familia de ternas parámetros de control ( $k_p$ ,  $k_i$ ,  $k_d$ ...) para cada valor del retardo. Posteriormente, en el funcionamiento real de la planta, y gracias a la marca de tiempo de las tramas enviadas, o bien mediante el envío de tramas específicas cuyo objetivo sea estimar los retardos, el nodo actuador puede elegir (mediante interpolación) el mejor valor del conjunto de parámetros que previamente ha calculado el controlador en función de los retardos de red, para así compensar en lo posible los efectos indeseados de las tareas y/o tramas interferentes.

#### 5.7.1. Estimación de los retardos producidos

En este apartado se realizará una estimación de los retardos producido en red; se asumirá que la planta no presenta retardo alguno ( $\tau = 0$ ), y que solo hay una tarea interferente en el nodo controlador, y tramas interferentes generadas por un nodo interferente conectado a la misma red desde la que se ejecuta el bucle de control.

En el caso de la tarea interferente en el nodo controlador, ocupando un 50% del ancho de banda, los retardos medidos son:

$$\tau_r = \tau_{s-c} + \tau_c + \tau_{c-a} = 1.002 \text{ s}$$

En el caso de tramas interferentes, los retardos encontrados (para una media de 706 tramas por segundo) presentan las siguientes características:

- Mínimo retardo: 34 ms.
- Máximo retardo: 849 ms.
- Retardo medio: 108 ms.

Por ello se calcularán los parámetros óptimos del regulador para un retardo de 1 s, y para retardo menores se interpolará entre los valores  $k_p$ ,  $k_i$  y  $k_d$  correspondientes a  $\tau=0$  y esos mismos valores para  $\tau=1$ . Obviamente, operando del mismo modo para retardos mayores de 1 s, se puede extender este método a cualquier valor del retardo de red  $\tau_r$ .

Conviene insistir una vez más en que en base a los resultados de las simulaciones realizadas en el punto anterior, el cálculo de los parámetros del regulador se llevará a cabo suponiendo  $\tau_r=0$ , y asumiendo que dicho retardo es de la planta, es decir: se concentrará el retardo  $\tau_r$  de la red en el parámetro  $\tau$  de la planta.

### 5.7.2. Estimación de los parámetros del regulador ( $\tau=1$ , $\tau=2$ )

Procediendo según se ha indicado anteriormente, y con ayuda de MATLAB, se encuentra el conjunto de parámetros del regulador PID que cumplen especificaciones para  $\tau=1$ :

- data.k = 12.0
- data.Ti = 32.0
- data.Td = -9.0
- data.N = 250.0

Análogamente se pueden hallar los parámetros óptimos del regulador para otros valores del retardo  $\tau$ . Por ejemplo, para  $\tau=2$  los parámetros calculados del regulador son:

- data.k = 10.0
- data.Ti = 36.0
- data.Td = -12.0
- data.N = 400.0

Dado que estos parámetros son sensiblemente iguales a los que se obtendrían interpolando sobre los obtenidos para  $\tau=1$ , la recta de interpolación hallada para  $\tau=1$  se puede dar por válida también para valores  $\tau(1) < \tau < \tau(2)$ .

### 5.7.3. Cálculo de los parámetros del regulador.

Se está ya en condiciones de calcular los valores del regulador PID del sistema para cualquier valor del retardo entre 0 y 1. Para ello, se incorporan en el código de inicialización '*controller\_init*' (ver figura 59) los valores de los parámetros  $k_p$ ,  $k_i$ ,  $k_d$  y N para  $\tau=0$  (valores '*data.X\_0*') y para  $\tau=1$  (valores '*data.X\_1*'), y posteriormente en '*controller\_code*' (ver figura 7), se llama a la función '*interpolate(data, z)*', que actualiza las constantes que se usan en el regulador ('*data.K*', '*data.Ti*', '*data.Td*' y '*data.N*') de la estructura '*data*' en función del valor del retardo estimado  $2 \times z(2)$ ; nótese que cuando se ejecuta el código del controlador, el retardo medido, contenido en  $z(2)$ , es el retardo  $\tau_{s-c}$  y no el retardo  $\tau_r$ , el retardo  $\tau_r$  se estima haciendo la siguiente simplificación:

$$\tau_r = 2 \times z(2) \approx 2 \times \tau_{s-c}$$

Nótese aquí que el valor del retardo  $\tau_{s-c}$  si es conocido y es real, no tratándose de ninguna aproximación o estimación. El código de la función '*interpolate.m*' puede verse en la figura 73.

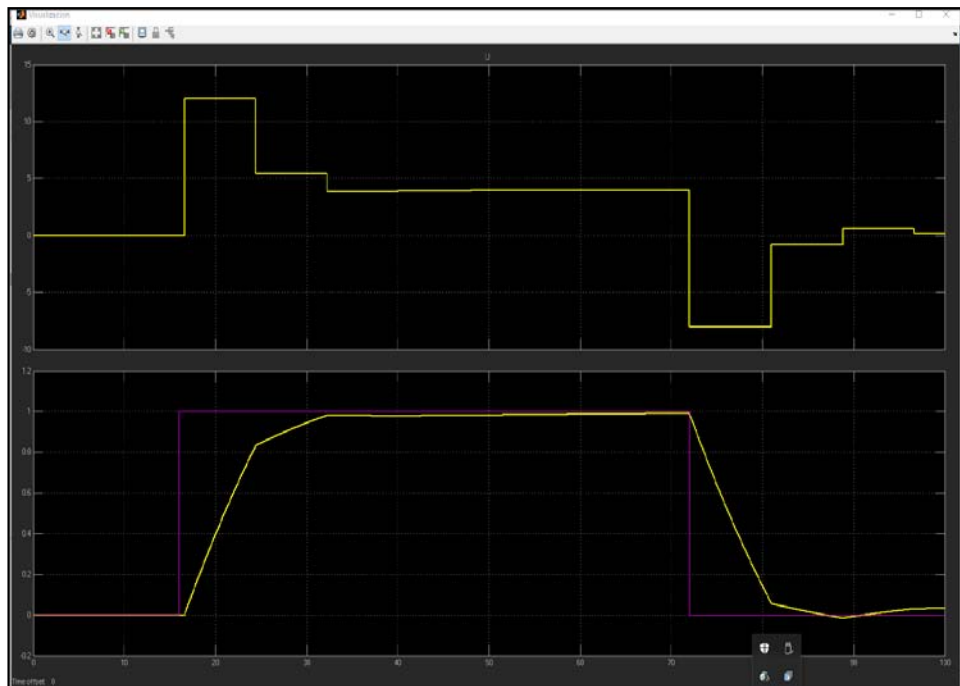
```
function data = interpolate(data, z, fix)
tau = 2*(ttCurrentTime - z(2));
if fix == 0
    data.K = data.K_0;
    data.Ti = data.Ti_0;
    data.Td = data.Td_0;
    data.N = data.N_0;
else
    if tau < 2
        data.K = data.K_0 - (data.K_1 - data.K_0)*tau;
        data.Ti = data.Ti_0 + (data.Ti_1 - data.Ti_0)*tau;
        data.Td = data.Td_0 - (data.Td_1 - data.Td_0)*tau;
        data.N = data.N_0 + (data.N_1 - data.N_0)*tau;
    else
        data.K = data.K_1;
        data.Ti = data.Ti_1;
        data.Td = data.Td_1;
        data.N = data.N_1;
    end
end
end
```

**Figura 73:** Código de la función '*interpolate()*'

### 5.7.4. Resultados obtenidos

Para evaluar la bondad del método de compensación de los retardos de red, se va a repetir la simulación del punto 5.6.5, figura 70 (40% de probabilidad de envío de tramas, con un total de unas 70000 tramas), pero en esta ocasión compensando el retardo tal y como se descrito más arriba. Los resultados pueden verse en la figura 74.

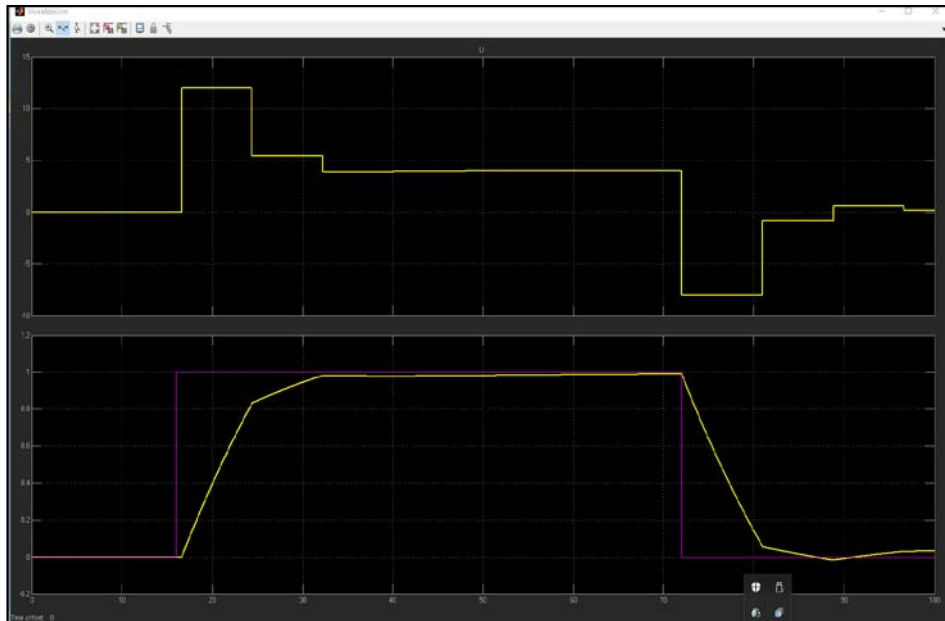
Del examen de la citada figura, se desprende que la respuesta del sistema ante retardos variables de red, mejora aplicando el método de compensación de retardos descrito en los apartados anteriores; además, la compensación del retardo se hace en tiempo real en cada instante de muestreo, por lo que el regulador se adapta de manera más adecuada a los retardos instantáneos que se producen en la red. Por otra parte, los resultados obtenidos están de acuerdo con los trabajos descritos en [Henriksson-03].



**Figura 74:** *Respuesta con trama interferente y compensación de retardo*

Repitiendo ahora la simulación en las mismas condiciones descritas en 5.6.3 (nodo controlador con tarea interferente adicional, ocupando un 50% del ancho de banda), pero compensando los retardos en tiempo real con la función '*interpolate.m*', se obtiene la respuesta mostrada en la figura 75.

Comparada con la respuesta obtenida sin compensar los retardos (figura 68), se aprecia que el tiempo de establecimiento se alarga algo, pero sin embargo se observa que se evita la sobreoscilación en la respuesta de subida y bajada del escalón, siendo por lo tanto una respuesta mejor la obtenida en la figura 74 aplicando la compensación del retardo, que la de la figura 68, sin aplicar dicha compensación.



**Figura 75:** *Respuesta con tarea interferente y compensación de retardo*

### 5.8. Otras estrategias de compensación del retardo de red

La simplificación adoptada en el punto anterior:

$$\tau_r = 2 \times z(2) \approx 2 \times \tau_{s-c}$$

Tiene la ventaja de ser sencilla y fácil de calcular, así como la de dar en las simulaciones unos buenos resultados, además con muy pocos cálculos adicionales, ya que se basa en una medida real ( $\tau_{s-c}$ ), y asume que  $\tau_{c-s}$  será muy similar a  $\tau_{s-c}$ , cuando en realidad es imposible saber si esto será así o no. Hay otras estrategias que tratan de afinar más los retardos y compensar de manera más precisa sus efectos sobre el sistema controlado; aquí se describen dos de las más tratadas por distintos autores.

#### 5.8.1. Packet-Based Control (PBC)

La inclusión de estos dos bloques se basa en las características propias de las transmisiones por red, en los que se necesita la misma cantidad de recursos de red si se envía un solo bit que si se envían varios cientos de bits de datos; esta situación es clara en aquellos casos en los que la red transmite un paquete de longitud de datos mínima: si los datos útiles no son suficientes, la trama se rellena con datos no útiles hasta alcanzar la longitud mínima.

En base a esta observación se desarrolla la estrategia Packet-Based Control, donde se diseña de un paquete de datos especial

en el controlador, sobre el cual, aplicando una regla de comparación en el actuador, se pueden compensar de forma explícita los retardos, pérdida de paquetes o desorden de estos producidos durante la transmisión. El controlador utiliza un modelo de la planta para predecir futuros estados de ésta y calcular futuras acciones de control, en base a una colección de posibles retardos en la transmisión, seleccionando en el actuador los valores adecuados. Algunos trabajos que consideran estrategias de este tipo son [Zhao-11], [Guinaldo-11].

### 5.8.2. Receding Horizon Control (RHC)

En este caso, el controlador envía una secuencia de acciones de control al actuador para el instante actual y los futuros. Obviamente, puede ocurrir que algo imprevisto suceda en el instante  $i$ , y a partir de ahí no sirva la secuencia de acciones de control enviadas; por otra parte, al aproximarse el tiempo final, las acciones de control enviadas (y predichas con anterioridad), tienden a ser cada vez más inexactas. RHC se basa en algunas de las ideas del denominado '*Mode Predictive Control*' o MPC:

- En el instante  $i$  y para el estado actual  $x_i$ , se diseña un conjunto de acciones de control hasta el instante final, digamos  $[i, i+N-1]$ , tomando en cuenta las limitaciones actuales y futuras (conocidas).
- Aplicar solo la primera acción de control calculada en el paso anterior.
- Medir el estado alcanzado en el instante  $i+1$ .
- Se repite la optimización desde el instante  $i+1$  hasta el instante final  $[i+1, i+N]$  a partir del estado actual (en este instante, estado  $x_{i+1}$ ).

En ausencia de perturbaciones, el estado medido en el paso 3, será igual al predicho por el modelo. Algunos trabajos que utilizan esta estrategia son, por ejemplo, [Quevedo-04], [Quevedo-07], [Mattingley-10].

Como puede apreciarse, ambas estrategias se basan en esencia en la comparación de los estados medidos en un instante determinado en el actuador con los estados predichos por un determinado modelo, usados en el controlador. Cuando hay retardos, la idea de estas estrategias para compensar los retardos es:

- 1) Elegir de las acciones de control la más adecuada
- 2) Cuando no llega la acción de control a tiempo, seguir aplicando las acciones de la secuencia.

## Sistemas de Control Basados en Red

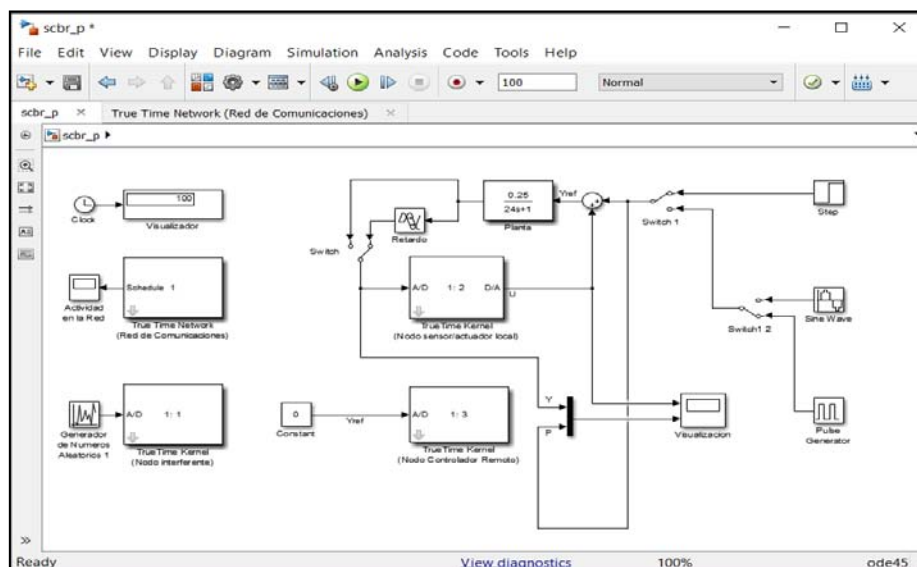
En ambas estrategias, el controlador "base" no cambia, y se diseña bajo condiciones ideales de la red, tratando de compensar esta con el conocimiento que aporta el modelo de la planta. El enfoque en este trabajo es diferente: es el diseño del controlador el que cambia en función de las condiciones de la transmisión.

### 5.9. Comportamiento frente a perturbaciones

Como en el capítulo anterior, y para terminar el análisis de Sistemas de Control Basados en Red, se simulará el comportamiento frente a perturbaciones del sistema de primer orden:

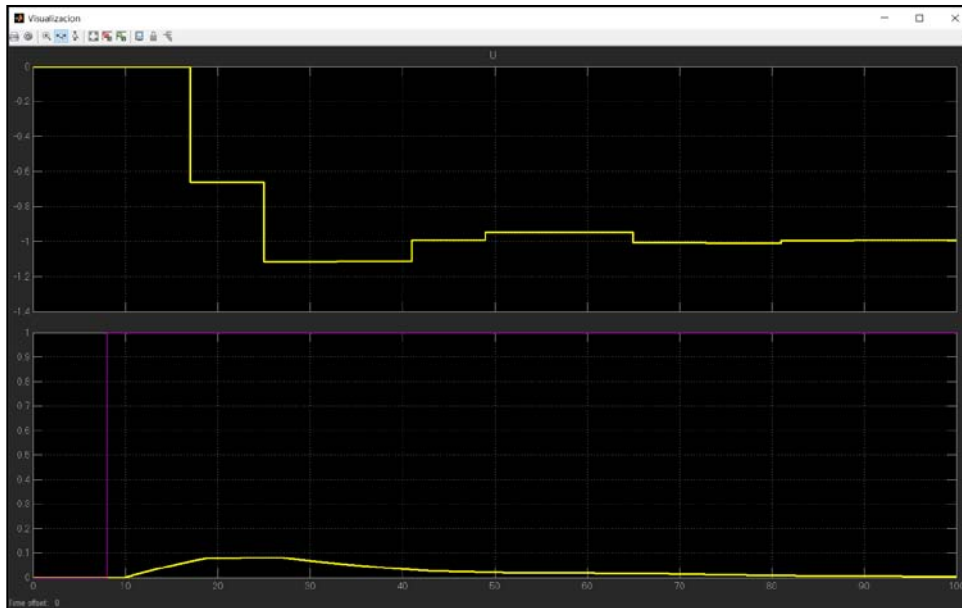
$$G(s) = \frac{0.25}{24s + 1} e^{-\tau s}$$

Que se ha estado utilizando hasta ahora como sistema objetivo a lo largo del capítulo, con los mismos parámetros de simulación y con la misma estrategia de compensación de retardo de la red que se ha descrito en el punto 7. Para ello, se ha modificado el diagrama de bloques en Simulink; el nuevo diagrama de bloques se presenta en la figura 76. Puede observarse en dicho diagrama que se ha fijado  $y_{ref} = 0$ , y que se ha sumado a la salida del actuador (entrada de la planta) la perturbación, que puede seleccionarse mediante un sistema de dos conmutadores de entre tres tipos diferentes: escalón, senoidal y tren de onda cuadrada.



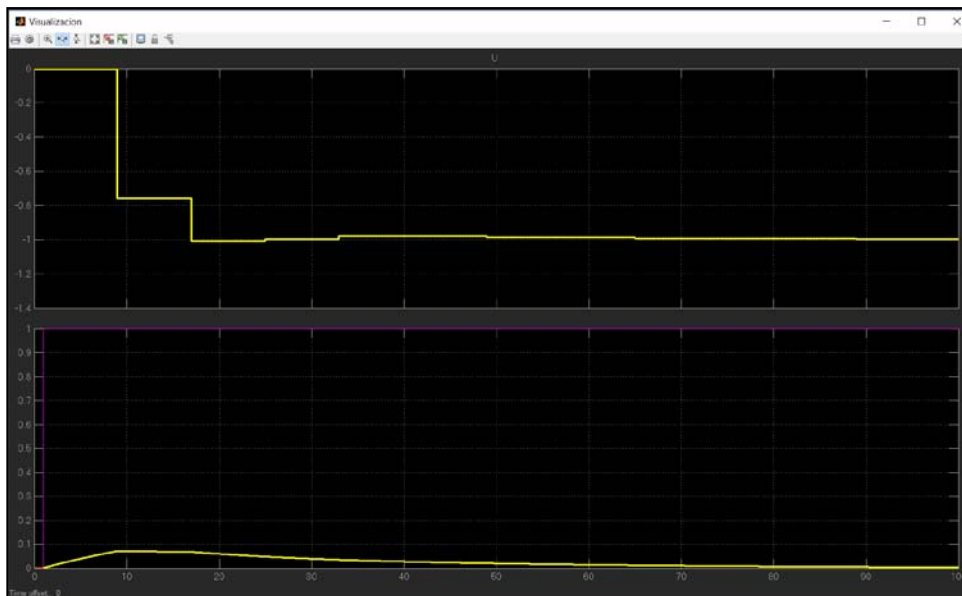
**Figura 76:** *Diagrama de Bloques Simulink para perturbaciones*

Simulando el caso de perturbación escalón unitario, la salida que se obtiene,  $y$ , se representa en la figura 77.



**Figura 77:** *Respuesta del sistema a la perturbación escalón ( $\tau=0$ )*

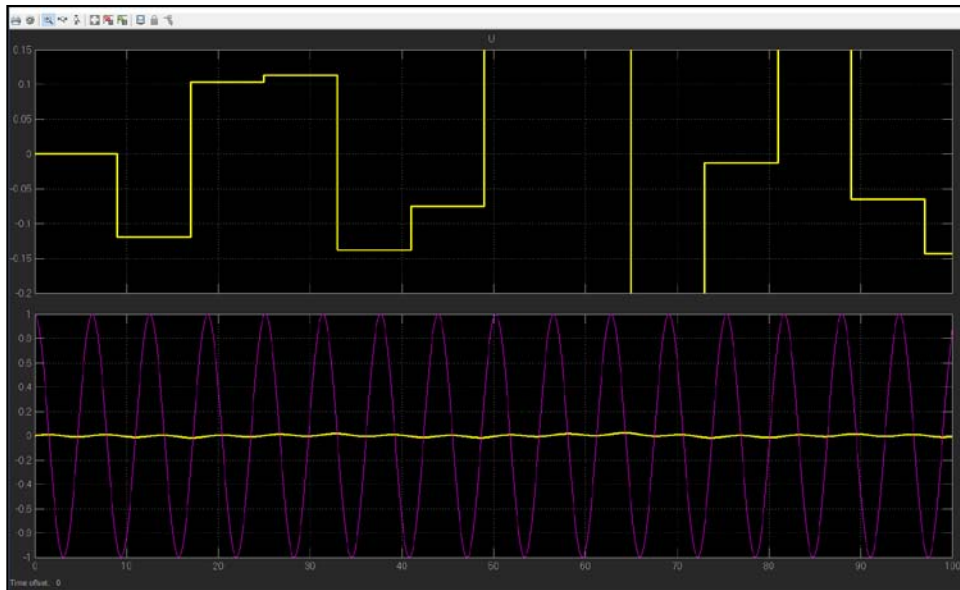
La amplitud de la salida  $y$  es solo un 8% de la amplitud de la perturbación, y aunque esta es continua una vez que el escalón se ha producido, la salida  $y$  vuelve al valor  $y_{ref}$  en unos 50 s.



**Figura 78:** *Respuesta del sistema a la perturbación escalón ( $\tau=2$ )*

La figura 78 muestra la simulación cuando la planta tiene un retardo  $\tau = 2$  s; la respuesta del sistema  $y$  es similar a la obtenida con  $\tau = 0$ .

Mediante simulaciones adicionales, puede comprobarse que el aumento del retardo  $\tau$  no altera de forma apreciable la respuesta del sistema.



**Figura 79:** *Respuesta del sistema a la perturbación senoidal ( $\tau=0$ )*

La figura 79 muestra la respuesta del sistema a una perturbación senoidal, que responde a la expresión:

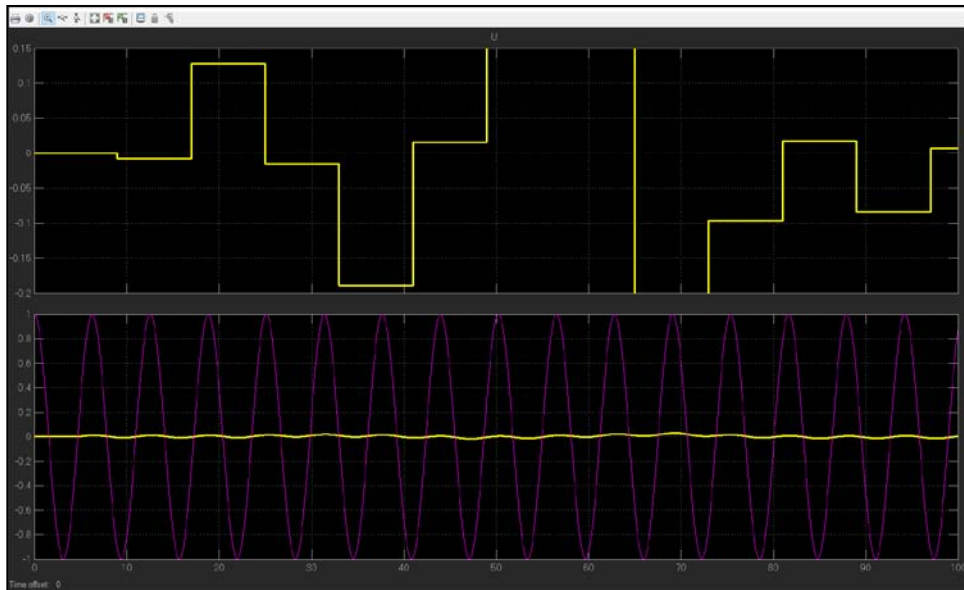
$$p(t) = \text{sen}\left(t + \frac{\pi}{2}\right)$$

De manera que la amplitud es la unidad. Puede observarse que la salida  $y$  está muy atenuada, siendo incluso menor del 1% de la amplitud de la perturbación de entrada.

La respuesta del sistema a la misma perturbación senoidal, cuando la planta tiene un retardo  $\tau = 5$  s se puede ver en la figura 80; se observa que la respuesta es muy similar a la de la figura 79, si bien el sistema acusa el retardo de la planta. La amplitud de la señal de salida es nuevamente menor del 1% de la amplitud de la señal perturbadora.

Por último, la figura 81 muestra la respuesta del sistema cuando la perturbación es una onda cuadrada de amplitud pico a pico la unidad, y con una frecuencia de 2 s. En este caso, la perturbación altera muy poco la respuesta del sistema, que no supera el 5% de la amplitud de aquella, volviendo a 0 en unos 40 s, tiempo similar al obtenido en el caso de la perturbación senoidal.

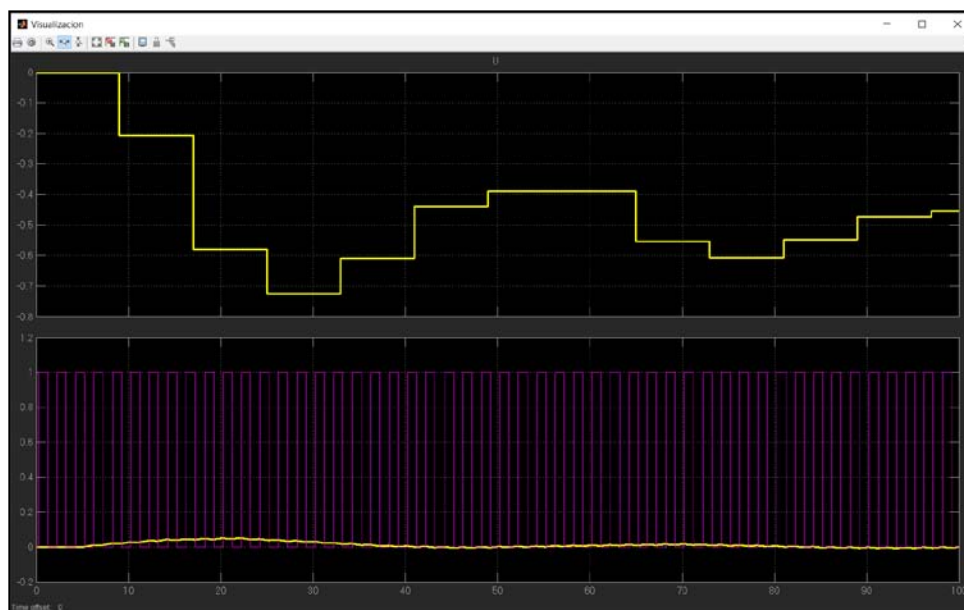
Se puede afirmar que el sistema atenúa en gran medida las perturbaciones introducidas en el sistema, independientemente de que estas sean de tipo escalón, senoidal o de onda cuadrada.



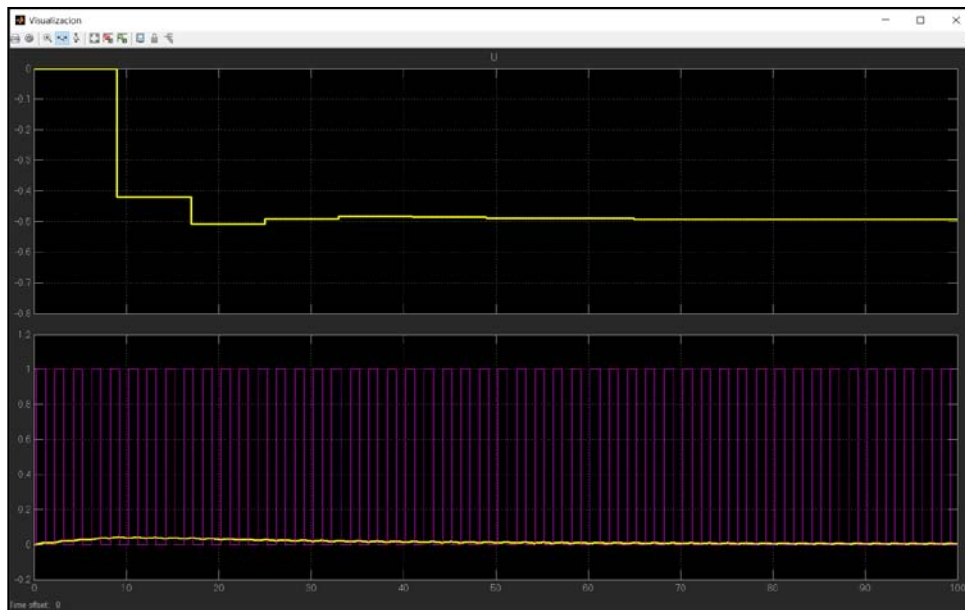
**Figura 80:** *Respuesta del sistema a la perturbación senoidal ( $\tau=5$ )*

Si ahora hacemos la simulación incluyendo un retardo de la planta de  $\tau = 5$  s, la figura 82 nos muestra que la salida del sistema apenas cambia respecto de la del caso  $\tau = 0$ .

Se ha comprobado mediante simulaciones adicionales, que retardos superiores dan una respuesta similar del sistema, por lo que este no afecta a la respuesta del sistema a este tipo de perturbación, que como en los casos de la perturbación de tipo escalón y senoidal, queda muy atenuada a la salida.



**Figura 81:** *Respuesta del sistema a la perturbación cuadrada ( $\tau=0$ )*



**Figura 82:** *Respuesta del sistema a la perturbación cuadrada ( $\tau=5$ )*



## Conclusiones y Trabajo Futuro

---

### 6.1. Conclusiones

En este trabajo se ha estudiado con detalle el retardo en los LTI, abarcando desde su origen, herramientas matemáticas disponibles para abordarlo, efectos y compensación de los mismos.

En particular, en el capítulo 1 se ha hecho un breve análisis del origen físico del retardo en los LTI, su significado físico, de sus efectos (negativos) en las prestaciones del sistema, así como de su modelado matemático, y se ha manifestado la dificultad adicional que el retardo incorpora en un sistema de control, al incorporar infinitos ceros en la ecuación característica.

En el capítulo 2, se ha analizado en detalle las aproximaciones algebraicas que permiten reducir un sistema con el término de retardo en su fdt  $e^{-\tau s}$  a un sistema con una fdt racional, sin dicho término retardo, y como esta aproximación produce una nueva fdt de fase no mínima.

Un problema esencial en la teoría de control, es determinar si un sistema dado es estable, así como los parámetros adecuados del regulador que se use para que el sistema sea estable en bucle cerrado.

En los sistemas con retardos, ambos objetivos se ven dificultados: es necesario disponer de herramientas para poder gestionar dichos retardos, y sus efectos en los bucles de control. Existen dichas herramientas, como las aproximaciones de Padé, los predictores de Smith y el teorema de Hermite-Biehler.

También se ha estudiado en este capítulo 2 el teorema de Hermite-Biehler por ser una herramienta más poderosa que otras aproximaciones, ya que Hermite-Biehler no solo da resultados exactos de estabilidad, sino también los relativos al lugar geométrico de parámetros admisibles para la estabilidad en bucle cerrado, tanto de sistemas estables como inestables en bucle abierto cuando estos presentan retardo, y por tanto sus polinomios característicos infinitas raíces, complementando los criterios clásicos de Routh y Hurwitz.

## Conclusiones y Trabajo Futuro

---

En el Capítulo 3 se ha analizado de manera específica el muestreo y discretización de sistemas FOTD con retardo, se ha implementado un algoritmo que lleva a cabo dicha discretización, y se ha visto como esta genera una descripción en ecuaciones de estado con estados adicionales, que es preciso incorporar para modelar el retardo. Posteriormente se ha ampliado el algoritmo para que lleve a cabo dicha discretización con sistemas de orden superior al primero. Se vio también como la discretización de sistemas con retardo tiene la ventaja de reducir su dimensión inicialmente infinita a una finita, pues para especificar el estado del sistema es necesario memorizar la entrada durante un intervalo de tiempo igual al retardo que presenta el sistema. Utilizando un mantenedor de orden 0, la señal de entrada siempre puede representarse por un número finito de valores.

También se han simulado las respuestas de un sistema con distintos valores de retardo inicialmente usando un regulador del tipo PI, y posteriormente, cualquier tipo de regulador, viendo que el incremento del retardo, como era de esperar, hace al sistema más inestable.

En el capítulo 4 se han estudiado los sistemas multifrecuencia en general, incidiendo de manera particular en los sistemas bifrecuencia, y su efecto cuando se aplican la técnica bifrecuencia a sistemas con retardo, comprobando mediante simulaciones que los esquemas de muestreo multifrecuencia son útiles allá donde se quieran mejorar las especificaciones temporales o frecuenciales del sistema (sobreimpulso, tiempo de establecimiento, margen de fase y de ganancia, oscilaciones, rizado...). Se trabajó primero con reguladores PI, y posteriormente con todo tipo de reguladores. Se ha comprobado que el modelado de los sistemas multifrecuencia exige también la incorporación de estados adicionales, de manera análoga al caso del modelado de los retardos.

Permiten mayor libertad de diseño (asignación de polos y ceros, mapeo del modelo, control descentralizado), simplifican la implementación de los sistemas de control, al requerir equipos de menos prestaciones y memoria, especialmente en los sistemas MIMO, donde hay varias entradas y salidas.

Se adaptan mejor a las condiciones reales de los bucles de control, permitiendo reproducir esquemas de muestreo regulares, irregulares, sincronizados o no, que se acercan mucho más a la realidad que los esquemas de muestreo teóricos monofrecuencia o clásicos.

La respuesta a perturbaciones puede resumirse en que aquella se ve afectada por las características del muestreo aplicado, y por tanto por el valor de  $N$  (en muestreo bifrecuencia): al aumentar  $N$ , en general aumenta la oscilación en la salida del sistema y el error en la misma; y menos por el

## Conclusiones y Trabajo Futuro

---

retardo  $\tau$  del sistema; al aumentar el retardo, las oscilaciones a la salida del sistema a causa de las perturbaciones tienden a mantenerse e incluso reducirse en algunos casos, aunque nunca de manera significativa.

El capítulo 5 analiza los SCBR, sistemas con retardo en el que este último se incorpora al sistema como consecuencia del uso de una red de transmisión digital de datos entre los elementos del bucle de control: sensor, controlador y actuador. Tras el análisis realizado en este capítulo, se ha podido constatar que los SCBR presentan una problemática adicional a la de cualquier otro sistema de control, por las circunstancias especiales que concurren en un sistema descentralizado, y un medio de intercambio de datos (la red de comunicaciones) que debe de compartirse con otros sistemas o con otras funciones de datos. Los retardos que presentan los SCBR se deben a la naturaleza estocástica de la información que circula por los mismos, y a las diferentes tareas adicionales que los nodos involucrados en el bucle de control pueden realizar potencialmente.

Se ha comprobado también que los retardos producidos por el tráfico interferente en la red y las tareas interferentes que se llevan a cabo en los nodos involucrados en el bucle de control, pueden asimilarse a retardos variables en el tiempo de la planta, a efectos de cálculo de mejor valor de los parámetros del regulador utilizado.

Finalmente, se ha propuesto un mecanismo sencillo de compensación del retardo producido en red, mediante el uso de marcas de tiempo insertadas en las tramas, lo que permite:

- Estimar los retardos en tiempo real que se producen en la red.
- Compensar dichos retardos mediante la modificación en tiempo real de los parámetros del regulador, manteniendo la estructura esencial de este último.
- Presentar una buena respuesta frente a perturbaciones introducidas en el sistema, comparable a la de los sistemas convencionales.

### 6.2. Trabajo Futuro

En el capítulo 2 se ha trabajado con el Teorema de Hermite-Biehler, y se ha codificado un algoritmo para determinar el lugar geométrico en el espacio de parámetros que hacen el sistema estable en bucle cerrado para sistemas de primer orden y reguladores PI; este algoritmo puede extenderse a reguladores PID y a todo tipo de sistemas; hay ya trabajos en ese sentido, como en [Oliveira-05] y [Silva-02].

El algoritmo puede hacerse aún más útil si da dicho lugar geométrico en función de especificaciones propias de los sistemas de control (tiempos de establecimiento, sobreoscilación, etc.). En particular, se puede pensar en una aplicación GUI basada en Hermite-Biehler para determinar las ternas de

## Conclusiones y Trabajo Futuro

---

valores  $\{k_p, k_i, k_d\}$  que hacen estable un determinado sistema, y elegir entre las mismas en función de determinadas prestaciones del sistema.

Por otra parte, en la extensión del algoritmo a todo tipo de controladores y retardos (no solo a los conmensurables), puede analizarse si existe una estructura general para el mismo que permita derivar cualquiera de los reguladores P, PI y PD del regulador PID, sin más que anular la constante del regulador PID correspondiente.

En el capítulo 4 se han encontrado algunas dificultades para trabajar en MATLAB con las ecuaciones de estado extendidas (con estados adicionales) tanto de sistemas con retardo como de sistemas multifrecuencia, por lo que sería un trabajo interesante encontrar implementaciones del código realizado que se pudieran integrar fácilmente con MATLAB, ya que no existen muchas herramientas que permitan trabajar con sistemas multifrecuencia. La ampliación de la *'Multirate Control Toolbox'* usada en este trabajo sería sin duda muy interesante en ese sentido.

En el capítulo 5 se ha implementado un mecanismo de compensación del retardo que aunque es sencillo de entender e implantar, no es exacto: es posible implantar otras técnicas de compensación ya mencionadas en dicho capítulo, como la PPC y RHC, descritas en [Zhao-11] y [Mattingley-10] entre otros. Sería interesante evaluar la relación coste-beneficio de las mismas, entendiendo por coste la complejidad de implementación y configuración de las mismas.

Aunque los SCBR pueden considerarse en cierto modo un caso particular de los Sistemas de Control Basados en Eventos (el controlador solo calcula las acciones de control cuando se produce el evento de la llegada de una medida del sensor), una línea a desarrollar de la que ya existen algunos trabajos, es el estudio de estos sistemas en el caso de la existencia de retardo. Los sistemas de control periódicos o gestionados por tiempo han dominado los trabajos en ingeniería de control y de sistemas. El control basado en eventos parece una alternativa con gran potencial, especialmente cuando se consideran sistemas con capacidades limitadas de cálculo y de comunicaciones. En un sistema de control basado en eventos es la detección de un evento, en lugar del tiempo, lo que decide cuándo se van a aplicar nuevas acciones de control, o cuando se va a realizar un nuevo cálculo, del tipo que sea [Ruiz-13] [Dormido-08].

En todo este trabajo se han usado simulaciones con MATLAB; la corroboración de los resultados obtenidos mediante experimentos reales sería un trabajo importante, para asegurar que las simulaciones realizadas están en línea con la realidad.

## Referencias y Bibliografía

---

### 7.1. Referencias

[Apostolakis-91]: Apostolakis I.S., Jordan D., "A Time Invariant Approach to Multirate Optimal Regulator Design", *International Journal of Control*, vol. 53, pp. 1233-1254, 1991.

[Araki-86]: Araki M., Yamamoto K., "Multivariable Multirate Sampled-Data Systems: State Space Description, Transfer Characteristics, and Nyquist Criterion", *IEEE Transactions on Automatic Control*, vol. 31, pp. 145-154, 1986.

[Araki-93]: Araki, M., "Recent Development in Digital Control Theory", *Proceedings 12<sup>th</sup> IFAC World Congr.*, vol. 9, pp. 951-960, 1993.

[Aström-88]: Aström K. J., Hägglund T., "Automatic Tuning of PID Controllers", *Instrument Society of America*, 1988.

[Barraud-05]: Barraud J., Creff Y., Petit N., "PI Controllers Performances for a Process Model with Varying Delay", *Institut Français du Pétrole, École des Mines de Paris, France*, 2005.

[Bernhardsson-93]: Sampling of state space systems with several time delays. Bernhardsson B., *Preprints IFAC 12th World Congress'93*, 1993.

[Brosilow-79]: Brosilow C. B., "The structure and design of Smith predictors from the viewpoint of inferential control", *In Proceedings of Joint American Control Conference, Denver, Colorado*, 1979.

[Broussard-80]: Broussard J.R., Glasson D.P., "Optimal Multirate Flight Control Design", *Proceedings of the 1980 Joint Automatic Control Conference*, 1980.

[Campo-12]: campo A. B., "PID Control Design", *Computer and Information Science Numerical Analysis and Scientific Computing "MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1"*, September 2012.

## Referencias y Bibliografía

---

[Cervin-03]: Cervin A., "Integrated Control and Real-Time Scheduling", Department of Automatic Control, Lund Institute of Technology, Lund, April 2003.

[Chvostek-06]: Chvostek T., Kratky A., "Simulation Of Network Using TrueTime Toolbox," Institute of Control and Industrial Informatics, Slovak Republic, 2006.

[Cuenca-10]: Cuenca A., Salt J., Sala A., Pizá R., "A Delay-Dependent Dual-Rate PID Controller Over an Ethernet Network", IEEE Transactions on Industrial Informatics, pp 1-12, 2010.

[Dormido-08]: Dormido S., Sánchez J., Kofman E., "Muestreo, Control y Comunicación Basados en Eventos", Revista Iberoamericana De Automática e Informática Industrial 5 (1), 1697-7912, 2008.

[Guinaldo-11]: Guinaldo M., Sánchez J., Dormido S., "Co-design Strategy of Networked Control Systems for Treacherous Network Conditions". IET control theory & applications, 5(16), 1906-1915, 2011.

[Henriksson-03]: Henriksson D., Cervin A., Arzen K. E., "TRUETIME: Real-time Control System Simulation with MATLAB/Simulink", Department of Automatic Control Lund Institute of Technology, Box 118, SE-221 00 Lund, Sweden, 2003.

[Jury-67]: Jury E.I., "A Note on Multirate Sampled-Data Systems", IEEE Transactions on Automatic Control, vol. 12, pp. 319-320, 1967.

[Kranc-57]: Kranc G.M., "Input-Output Analysis of Multirate Feedback Systems", IRE Transactions on Automatic Control, PGAC-3, vol 3, pp. 21-28, 1957.

[Mattingley-10]: Mattingley J., Wang Y., Boyd S., "Code generation for Receding Horizon Control", Information Systems Laboratory, Electrical Engineering Department, Stanford University, CA 94305-9510 USA, 2010.

[Marshall-92]: Marshall, J. E., Gorecki, H., Korytowski, A., and Walton, K. Time-Delay Systems: Stability and Performance Criteria with Applications, Ellis Horwood, New York, 1992.

[Meyer-75]: Meyer R.A., Burrus C.S., "An Unified Analysis of Multirate and Periodically Timeinvariant Digital Filters", IEEE Transactions on Circuits Systems, vol. 22, pp. 162-168, 1975.

[Muro-09]: Del Muro B., Gonzalez O. A., Pedraza Y. A., "ESTABILIZACIÓN DE SISTEMAS DE SEGUNDO ORDEN CON RETARDO: APLICACIÓN A SISTEMAS DE ALTO ORDEN", Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacan, Instituto Politécnico Nacional, Santa Ana 1000, Coyoacán, D.F., 04430, México, 2009.

## Referencias y Bibliografía

---

[Oliveira-05]: Oliveira V. A., Cossi L. V., Silva M. F., Teixeira C. M., "PID Stabilization of a Class of Time Delay System", Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, 2005.

[Quevedo-04]: Quevedo D. E., Goodwin G. C., De Dona J. A., "Finite Constraint Set Receding Horizon Quadratic Control". International journal of robust and nonlinear control", 14(4), 355-377, 2004.

[Quevedo-07]: Quevedo D. E., Silva E. I., Goodwin G. C., "Packetized Predictive Control Over Erasure Channels". In American Control Conference, ACC'07 (pp. 1003-1008). IEEE, 2007.

[Routh-77] Routh E. J., "A Treatise on the Stability of a Given State of Motion". Macmillan, London, 1877.

[Ruiz-13]: Ruiz A., Jimenez J. E., Sánchez J., Dormido S., "Control Basado en Eventos de Sistemas de Primer Orden con Retardos", Revista Iberoamericana de Automática e Informática industrial 10, 302–312, 2013.

[Sala-09]: Sala A., Cuenca A., Salt J., "A Retunable PID Multi-rate Controller for a Networked Control System", Preprint submitted to Elsevier Science, January 2009.

[Salt-00]: Salt J., Cuenca A., "Análisis de la Respuesta Temporal de un Sistema con Muestreo No Convencional Basado en el Lugar de las Raíces", Dpto. de Ingeniería de Sistemas y Automática Universidad Politécnica de Valencia, 2000.

[Salt-07]: Salt J., Cuenca A., Casanova V., "Multi-rate PID Controller for a Networked Control System", Department of Systems Engineering and Control Polytechnic University of Valencia, Cno de Vera s/n, 46022 Valencia (Spain), 2007.

[Silva-02]: Silva G. J., Datta A., Bhattacharyya S. P., "New Results on the Synthesis of PID Controllers", IEEE Transactions on Automatic Control, vol. 47, pp. 241-252, 2002.

[Sklansky-55]: Sklansky J., Ragazzini J.R., "Analysis of errors in sampled-data feedback systems", AIEE Transactions, vol. 74, part II, pp. 65-71, 1955.

[Thompson-86]: Thompson P.M., "Gain and phase margin of multirate sampled-data feedback systems", International Journal of Control, vol. 44, pp. 833-846, 1986.

[Vélez-02]: Vélez C. M., "Modelado, Simulación y Control de Sistemas con Muestreo No Convencional", REVISTA Universidad EAFIT No. 125, Febrero-Marzo de 2002.

## Referencias y Bibliografía

---

[Vélez-03]: Vélez C.M., "Diseño e implementación de sistemas de control con muestreo no convencional", (Informe de investigación 2002), Universidad EAFIT (2003).

[Vélez-04a]: Vélez C. M., "Control Multifrecuencia en Tiempo Real Utilizando Herramientas CACSD". Revista Iberoamericana de Automática e Informática Industrial (RIAI), ISSN 16977912, Vol. 1, No. 3, pp 43-52, 2004.

[Vélez-04b]: Vélez C. M., "Ideas y Aplicaciones de los Sistemas de Control Multifrecuencia", Revista Colombiana de Tecnología Avanzada, ISSN:1692-7257 Volumen 1 No 3 – 2004.

[Whitbeck-80]: Whitbeck R.F., Didaleusky D.C.J., "Multirate digital control systems in simulation applications", Report AFWAL-TR-80-3101, vols I, II and III, Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratory, Wright Patterson Air Force Base, Ohio, 1980.

[Wittenmark-02]: Wittenmark B., Aström K. J., Arzen K. E., "Computer Control: An Overview", IFAC Professional Brief, pp. 4-14, 2002.

[Wittenmark-85]: Wittenmark B., "Sampling of a System with Time Delay", IEEE Transactions on Automatic Control, vol. 30, pp. 507-510, 1985.

[Zhao-11]: Zhao Y. B., Kang Y., Liu G. P., Rees D., "Stochastic Stabilization of Packet-Based Networked Control Systems", International Journal of Innovative Computing, Information and Control, Volume 7, Number 4, April 2011.

[Zhang-01]: Zhang W., Branicky S., Phillips S. M., "Stability of Networked Control Systems", IEEE Control Systems Magazine, February 2001.

### 7.2. Bibliografía

[Aström-97]: Aström K. J., Wittenmark B., "Computer-controlled Systems. Theory and Design" 3<sup>th</sup> Edition, Prentice-Hall Information and System Science Series, 1.997.

[Bompiani 11] Bompiani E., "Sulle condizioni sotto le quali un equazione a coefficienti reale ammette solo radici con parte reale negativa". Giornale di Matematica, 1911, 49, pp. 33-39.

[Bosch-91]: BOSCH, "CAN Specification" versión 2.0, Robert Bosch GmbH, Postfach 50, D-7000 Stuttgart, 1991.

[Corripio-91]: Corripio B., Smith C., "Control Automático de Procesos", Noriega Editores (Editorial Limusa) 1991.

## Referencias y Bibliografía

---

[Cuenca-04]: Cuenca A., "Modelado, Análisis y Diseño de Sistemas de Control con Muestreo No Convencional". Tesis Doctoral. Departamento de ingeniería de sistemas, Computadores y Automática. Universidad Politécnica de Valencia, 2004.

[Cuesta-04]: Cuesta A., Grau L., "Modelado de Reguladores PID Multifrecuencia". XXV Jornadas de Automática, Ciudad Real 2004.

[Dominguez-06]: Domínguez S., Campoy P., Sebastián J. M., Jiménez A., "Control en el Espacio de Estado", CEA, Pearson prencite-Hall, 2006.

[Hale-93]: Hale J., Lunel S. M., "Introduction to functional differential equations", Applied Mathematical Sciences, Springer-Verlang, 1993.

[Kamman-17]: Kamman J., "An Introduction to Three Dimensional, Rigid Body Dynamics", PhD Volume I–Kinematics: Summary of Contents, 2017.

[Klerk-02]: Klerk E., "Aspects of Semidefinite Programming. Interior Point Algorithms and Selected Applications", Springer, Applied Optimization, Volume 65, 2002.

[Rico-07]: Normey-Rico J. E., Camacho E. F., "Control of Dead-Time Processes", Advanced Textbooks in Control and Signal Processing, Springer 2007.

[Silva-05]: Silva G. J., Datta A., Bhattacharyya S. P., "PID Controllers for Time-delay Systems", Control Engineering Series, Birkhäuser 2005.

