UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Experiencias de Simulación Interactiva en Sistemas Multifrecuencia

Proyecto Final de Carrera

[Ingeniería Informática]

Autor: [Francisco Palau Romero]

Directores: [Ángel Miguel Cuenca Lacruz]

[Julián José Salt Llobregat]

[09/09/2013]

Resumen

El proyecto de final de carrera de la titulación de Ingeniería Informática "*Experiencias de Simulación Interactiva en Sistemas Multifrecuencia*" tiene como objetivo fundamental la generación de herramientas de análisis basadas en simulación interactiva para el análisis de sistemas multifrecuenciales.

Dada la complejidad de este tipo de sistemas de control digital con distintos muestreadores en diferentes partes del bucle de control, la simulación convencional resulta insuficiente. En el caso que ocupa, la existencia de una cantidad de parámetros interconectados requiere técnicas que permitan observar la implicación de cada variación en distintos aspectos del análisis y diseño de sistemas multifercuenciales.

En la herramienta preparada se han considerado reguladores habituales en los ámbitos industriales: PID, Cancelación-RST y Basados en Modelo.

Con la aplicación preparada se puede observar y lo que es más importante, descubrir, fenómenos atípicos de los sistemas multifrecuenciales a los que, como se decía, es muy difícil acceder con metodologías convencionales de simulación continua.

Finalmente cabe decir que para la realización de estos simuladores se ha utilizado la herramienta SysQuake (http://www.calerga.com/products/Sysquake/) un lenguaje de programación basado en MATLAB (http://www.mathworks.es/products/matlab/). Los ficheros generados en SysQuake se crean con la extensión .sq, y su contenido es texto plano similar al usado en lenguaje MATLAB.

.

Palabras clave: Control, Multifrecuencia, SysQuake, MatLab, Simulación.

Tabla de contenidos

Contenido

1. Introducción

En este documento se puede encontrar una detallada explicación del proyecto consistente en simulaciones en sistemas multifrecuencia, para las cuales se ha utilizado la herramienta SysQuake.

En los siguientes apartados se expondrá un resumen con teoría de control y una explicación también teórica de los controladores utilizados en las simulaciones. El contenido de este apartado, aunque extenso, intenta cubrir únicamente los conceptos más básicos para poder comprender el funcionamiento de los sistemas multifrecuencia y sus peculiaridades.

Una vez explicada la teoría fundamental, se expondrá un breve resumen sobre la herramienta SysQuake y MatLab. Estos apartados son puramente técnicos desde un punto de vista informático y científico, pero son necesarios para comprender el funcionamiento de la herramienta de simulación.

Finalmente se mostrarán los simuladores creados, las opciones que ofrecen al usuario, sus limitaciones, y algunos resultados obtenidos. Con estos resultados cerraremos el documento, ofreciendo conclusiones del trabajo realizado y el agradecimiento debido a todos aquellos que han colaborado en este proyecto de casi un año de duración.

2. Teoría de Control

2.1 Control clásico

Cuando se realiza un control digital de un proceso continuo *G(s)* es necesario que:

a) La señal de entrada al proceso se convierta de discreta a continua.
b) La señal de salida del proceso se convierta de continua a discreta.

Para conseguir a) se pone un bloqueador a la entrada del proceso; y, para conseguir b), un muestreador a su salida, tal y como se muestra en la Figura 1:

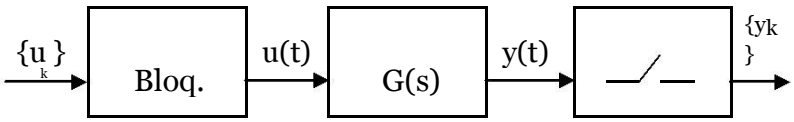


Figura 1: Bloqueador y Muestreador

Un **bloqueador de orden cero** ZOH es un dispositivo que transforma una señal discreta en una señal continua. La transformación se consigue mediante una reconstrucción dada por:

$$y(t) = y(t_k) \quad t_k \leq t \leq t_{k+1}$$

Es decir, la señal reconstruida es una señal continua escalonada cuyo valor se actualiza con cada instante de muestreo de la señal discreta original.

Un **muestreador** S es un dispositivo que transforma una señal continua $v(t)$ en una señal discreta $w(k)$. Esto se representa como $w = Sv$.

Así pues, el esquema elemental de control es el que se muestra en la Figura 2:

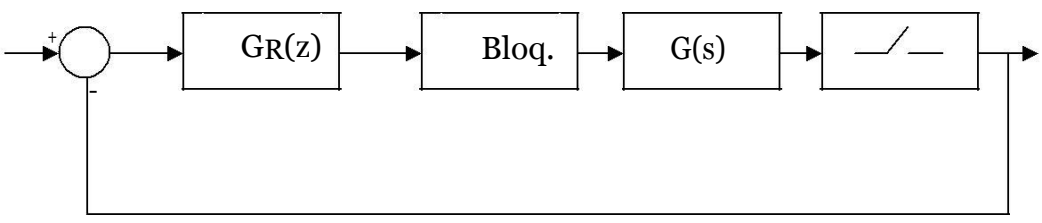


Figura 2: Esquema de control en bucle cerrado

En las aplicaciones industriales, el proceso de adquisición de datos (obtención de la secuencia $\{y_k\}$) tiene más restricciones temporales que la actualización de la señal de control (secuencia $\{u_k\}$). Los motivos principales de que esto ocurra son los que siguen:

- El proceso de medición de algunas variables para hacer la realimentación puede ser difícil, dando lugar a periodos de muestreo significativamente grandes. Esto se suele dar, por ejemplo, con los analizadores químicos, donde la muestra necesita un tiempo de preparación.

- En otros casos, aunque podría ser posible, no es conveniente tener tantas medidas y los recursos son usados para otros propósitos. Éste es el caso del posicionamiento del cabezal de lectura/escritura en el servosistema de las unidades de disco duro. En la técnica moderna del servo, la señal de posición está solamente disponible en los sectores, mejorando la tradicional técnica servo donde un área era dedicada para almacenar la posición del servo. Esto da lugar a una mayor capacidad de almacenaje sin incrementar el número de sectores del servo.

- En los sistemas de control distribuidos varios bucles de control comparten los mismos canales de comunicación. En este caso, los periodos de muestreo deben ser lo suficientemente grandes para que dichos canales de medición no se saturen (y, por tanto, las mediciones puedan ser procesadas y las acciones de control puedan ser aplicadas a tiempo).

Así pues, la conclusión es que el comportamiento de una planta controlada digitalmente suele empeorar a medida que el periodo de muestreo es más grande.

2.2 Control multifrecuencia

En estos sistemas donde dos o más partes del sistema operan a distintas frecuencias las teorías clásicas de control fuerzan a que todos sus elementos trabajen a la misma frecuencia lenta (es decir, el periodo de muestreo más grande de los disponibles). El control multifrecuencia, por el contrario, intenta alcanzar resultados similares a los que obtendríamos si todos los elementos del sistema pudiesen trabajar a la misma frecuencia rápida (es decir, el periodo de muestreo más pequeño), pero usando los datos muestreados con diferentes periodos.

El esquema de control de un sistema multifrecuencia (operando a T y NT) es el que se muestra en la Figura 3.

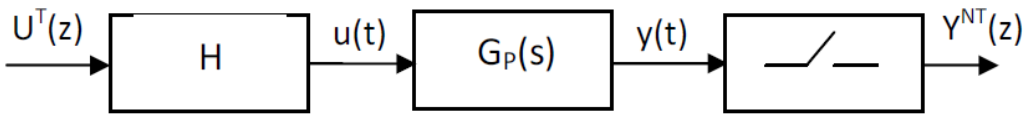


Figura 3: Esquema de control multifrecuencia (T y NT)

El control multifrecuencia no sólo se limita a sistemas en los que $\{y_k\}$ trabaja con una frecuencia más lenta que $\{u_k\}$, sino que también incluye los siguientes casos:

- En sistemas con múltiples procesadores, la presuposición de que todos los instantes de muestreo están sincronizados y operando a la misma frecuencia es falsa. Y esta suposición puede conducir a errores importantes en el modelado de los mismos.
- Los sistemas en los que las variables a controlar tienen naturalezas distintas (y, por tanto, dinámicas distintas) requieren frecuencias de muestreo acordes a cada una de ellas.

La definición de un sistema multifrecuencia, por tanto es la de un sistema discreto en el que dos o más variables son actualizadas a frecuencias distintas. El esquema de muestreo puede ser de varias formas siendo el más común y simple de modelar el correspondiente al muestreo periódico, es decir, aquel en donde el esquema de muestreo se repite cada cierto período global de tiempo *T*; este período global es igual al mínimo común múltiplo de los períodos de muestreo hallados en todas las señales del sistema.

Actualmente el control multifrecuencia es un área de estudio muy activa y de gran interés para los grupos de investigación de control automático. El gran desarrollo de la electrónica digital altamente integrada y de los recursos de cálculo para el modelado, análisis y diseño de sistemas digitales ha permitido un crecimiento acelerado del número de aplicaciones para la solución de problemas de ingeniería de control. El abanico de utilización de estos sistemas es muy amplio: desde los posicionadores del cabezal de los discos duros hasta el control de sistemas robóticos equipados con mecanismos de visión.

El objeto de estudio de este proyecto son los sistemas bifrecuencia, esto es, aquellos sistemas en los que se opera con dos periodos de muestreo distintos (T y NT).

2.3 Aplicaciones prácticas del control multifrecuencia

Consecuencia lógica de lo descrito en los apartados anteriores es que el control multifrecuencia permite obtener lazos de control con características y atributos que son difíciles o imposibles de obtener con otros métodos como los continuos puros o discretos monofrecuencia.

Es por esto que el control multifrecuencia ha sido utilizado con éxito en diversas áreas y sistemas como: robótica, control de discos duros, aeronáutica, columnas de destilación, reactores, procesadores, detección de fallos, industria de plásticos, submarinos, sistemas eléctricos, gas, sistemas neumáticos y motores. En este apartado se van a presentar algunos ejemplos concretos de estas aplicaciones.

Aeronáutica

En el esquema de control multifrecuencia desarrollado en [1] se muestrea la señal de salida de la planta a una alta frecuencia, y mediante un algoritmo se recomponen estas muestras en un vector de pseudomedidas (*pseudomeasurements vector*). El potencial de este esquema es muy grande ya que, entre otras ventajas, asegura el correcto funcionamiento del sistema aunque falle alguno de los sensores.
En dicho artículo se pone como ejemplo de aplicación el diseño de un regulador para un YF-16. Al compararlo con un regulador convencional analógico se demuestra que el controlador multifrecuencia es más eficiente y efectivo.

Imagen 1: YF-16

Reactores Químicos

En [2] se presenta la implementación de un control ador multifrecuencia no lineal para regular la temperatura y el peso molecular medio en un reactor polimérico continuo de radicales libres de metilmetacrilato. La temperatura se mide en un esquema con un período de muestreo muchísimo más pequeño que el peso molecular medio. Ambas medidas se incorporan al algoritmo cuando están disponibles para poder hacer predicciones de las salidas futuras.

El desempeño de este controlador se ha examinado variando los parámetros de sintonía del controlador para comprobar su capacidad para seguir cambios en referencias tipo escalón. Como resultado de esto, se concluyó que es te controlador es eficiente y efectivo para esquema de control con medidas multifrecuencia de la salida (MROC,
Multirate Output Control).

Resultados semejantes se obtienen en un reactor de etileno cuando se emplea un esquema similar; estos son presentados en [3].

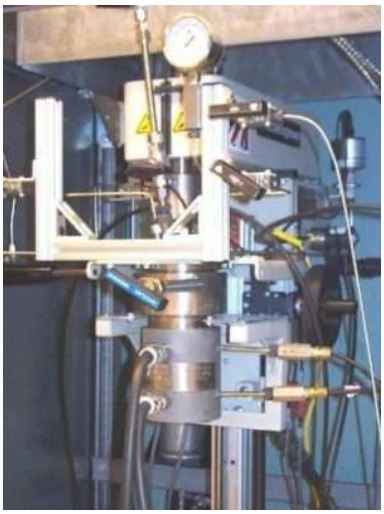


Imagen 2: Reactor para preparar materiales poliméricos

Discos duros

Cuando un sistema de control tiene múltiples actuadores con diferentes anchos de banda puede considerarse que hay varios lazos de realimentación operando con diferentes períodos de muestreo. Una frecuencia de muestreo alta es necesaria para el lazo con ancho de banda más grande y una frecuencia baja par el lazo con el ancho de banda menor. La implementación de un esquema multifrecuencia con un solo lazo de control es posible descomponiendo el algoritmo de control en bloques de operación rápida y lenta. Esta implementación reduce considerablemente la carga computacional del procesador encargado de realizar esta tarea.

Una aplicación de esta propuesta se encuentra en [4], en la cual se implementa un regulador multifrecuencia para el control de un actuador doble (*Dual-actuator*) de un posicionador del cabezal de un disco duro. Dicho sistema de posicionamiento utiliza un VCM (*voice coil motor*) para los movimientos "gruesos", y un PZT (*piezo electric transducer*) para los desplazamientos "finos" del cabezal. El VCM (dispositivo electromagnético) tiene un ancho de banda más estrecho que el PZT dispositivo piezoeléctrico)

Imagen 3: Disco duro

2.5 Discretización de la señal

De una señal continua $y(t)$ se puede obtener una secuencia de datos tomada a una frecuencia $1/T_0\{y(kT_0)\}, k \epsilon Z^+$. A esta secuencia la denotaremos con Y^{T_0}.

Por otra parte, si tenemos un bloqueador H_{T_i} operando a un periodo de muestreo T_i y al que le llega una señal de entrada $u(t)$, la salida tendrá una forma escalonada y quedará denotada por $\bar{\bar{U}}^{Ti}(t)$.

Si $T_0 = T_i = T$, las transformadas Z de las expresiones anteriores se pueden denotar con:

$$Y^T(z) \triangleq Z_T[y(t)] = \sum_{k=0}^{\infty} y(kT)z^{-k}$$

$$U^T(z) \triangleq Z_T[u(t)] = \sum_{k=0}^{\infty} u(kT)z^{-k}$$

Donde la variable z^{-k} representa el operador retardo. De igual forma, si los periodos de muestreo son ampliados de tal forma que $T_0 = T_i = NT, N \in Z^+$ las secuencias de entrada y salida tendrán las siguientes transformadas Z:

$$Y^{NT}(z) \triangleq Z_{NT}[y(t)] = \sum_{k=0}^{\infty} y(kNT)z_N^{-k}$$

$$U^{NT}(z) \triangleq Z_{NT}[u(t)] = \sum_{k=0}^{\infty} u(kNT)z_N^{-k}$$

Donde las series en las partes derechas de la fórmulas sólo presentan potencias de

$$z_N = z^N$$

2.6 Operadores expansión y salto

El operador de **expansión** (**expand**) es aquel que expande la escala de tiempo para una secuencia digital actuando de la siguiente forma:

$$E: [Y^{NT}(z_N)]^T \triangleq \hat{Y}^T(z^N) \triangleq \sum_{k=0}^{\infty} \hat{y}(kT) z^{-kN} \begin{cases} \hat{y}(kT) = y(kT);\ \forall k = \lambda N \\ \hat{y}(kT) = 0;\ \forall k \neq \lambda N \end{cases} \lambda \in z^+$$

Es decir, crea una T-secuencia a partir de una NT-secuencia, donde la secuencia de salida es igual a la secuencia de entrada cada *N* muestreos, fijando en 0 los *N-1* muestreos intermedios.

En la siguiente figura se expone un ejemplo para N = 3:

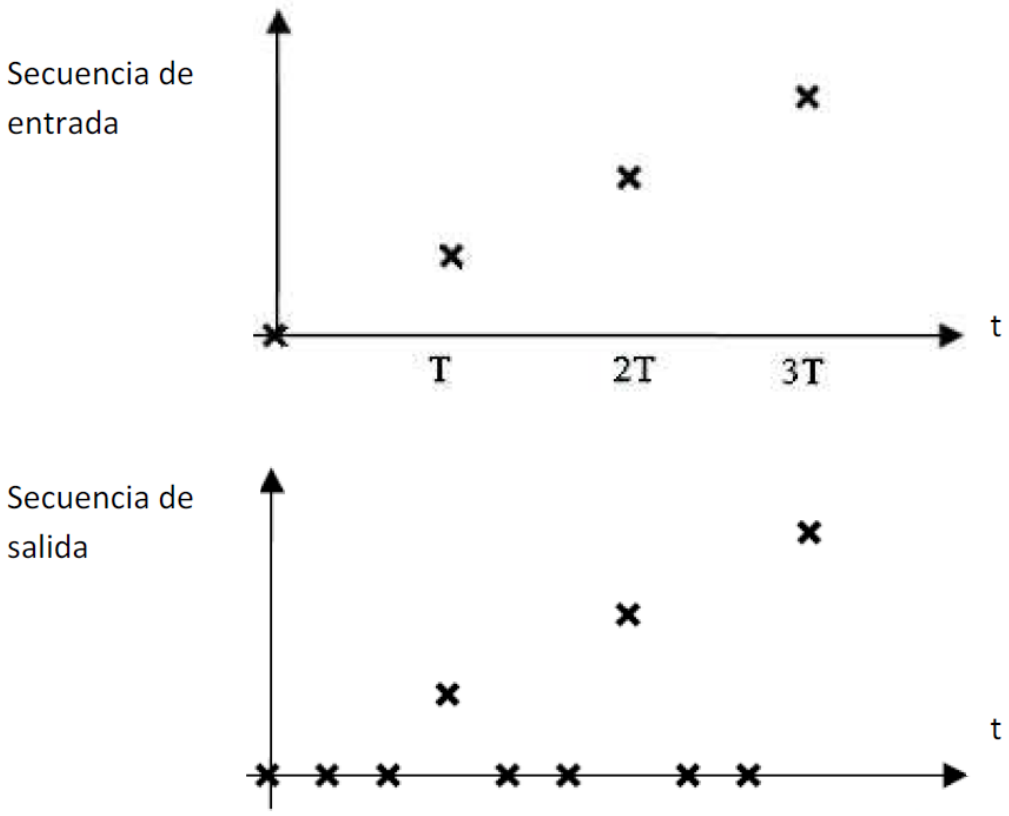


Figura 1: Resultado de aplicar el operador expansión

El operador de **salto** (**skip**) crea una T-secuencia a partir de una NT-secuencia, de forma que la secuencia de salida está formada por las N-ésimas muestras de la secuencia de entrada.

$$S: [Y^T(z)]^{NT} \triangleq \check{Y}^{NT}(z^N) \triangleq \sum_{k=0}^{\infty} y(kNT) z^{-kN} = Y^{NT}(z_N)$$

En la Figura 2 se expone un ejemplo para N = 3:

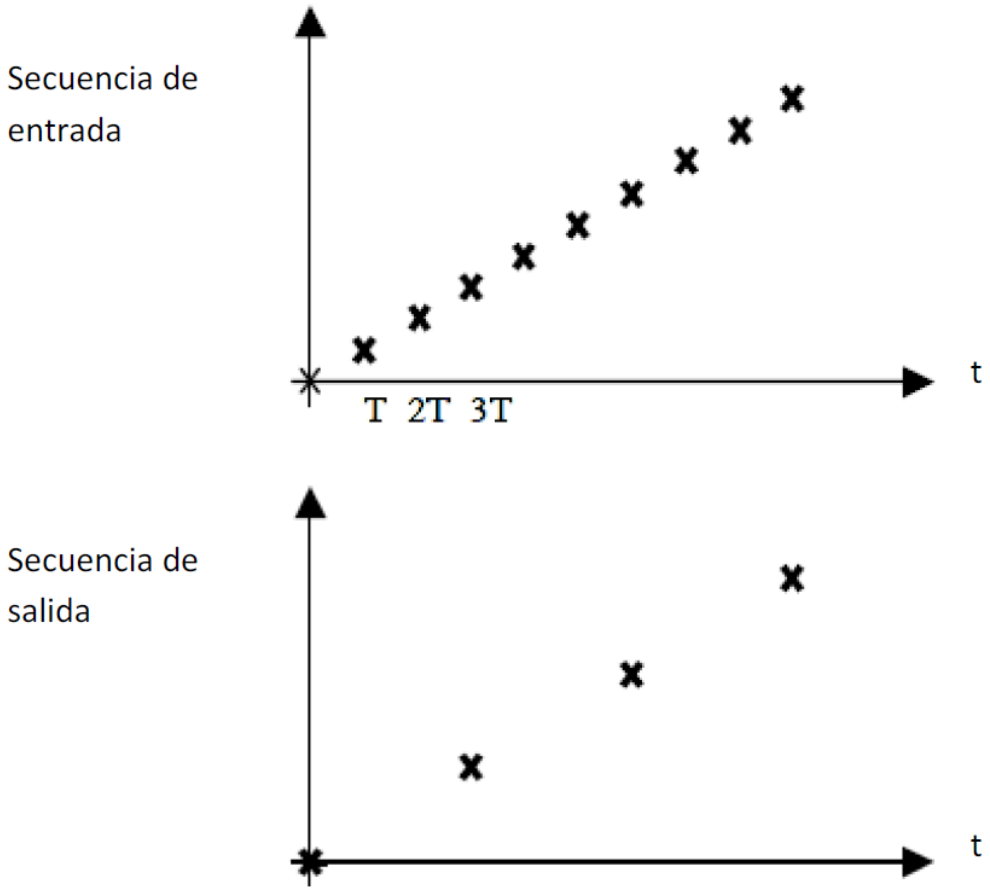


Figura 2: Resultado de aplicar el operador salto

El operador *skip* aplicado a la transformada Z de una señal puede obtenerse mediante la siguiente expresión, debido a [5]:

$$[F^T(z)]^{NT} = \frac{1}{N}\sum_{k=0}^{N-1} F(e^{j\frac{2\pi k}{N}} z|_{z^N=z_N} = \breve{F}^{NT}(z_N)$$

Las siguientes propiedades de los operadores *skip* y *expand* se usarán más adelante [6]:

- El operador *skip* no es conmutativo:
$$[X^T(z)Y^T(z)]^{NT} \neq [X^T]^{NT}[Y^T]^{NT}(z_N)$$
$$[X^T(z)[Y^{NT}]^T(z)]^{NT} \neq [X^T]^{NT}Y^{NT}(z_N)$$
- El operador *expand* sí es conmutativo:
$$[X^{NT}(z_N)Y^{NT}(z)]^T = [X^{NT}]^T[Y^{NT}]^T(z_N)$$

2.7 Lugar de las raíces

El **lugar de raíces** (del inglés, *root locus*) es el lugar geométrico de los polos y ceros de una función de transferencia a medida que se varía la ganancia del sistema K en un determinado intervalo.

El método del lugar de raíces permite determinar la posición de los polos de la función de transferencia en bucle cerrado para un determinado valor de ganancia K a partir de la función de transferencia en lazo abierto

Sea $G(s)H(s)$ una función de transferencia en lazo abierto. Pertenecen al lugar de raíces todos los puntos del plano complejo que satisfacen la ecuación característica:

$$1 + KG(s)H(s) = 0$$

Si $0 \leq K < \infty$ estamos trabajando con el lugar de raíces verdadero. Si no, se trata del lugar de raíces complementario. Una solución de la ecuación para un valor de k dado se llama lugar de la raíz.

Propiedades del lugar de las raíces

- El lugar de raíces es simétrico respecto del eje real.
- Comienza en $K = 0$ en los polos p_i de la función de transferencia en lazo abierto $G(s)H(s)$; y termina para $K \to \pm\infty$. Las soluciones para $K \geq 0$ corresponden al lugar de raíces verdadero, mientras que las soluciones para $K > 0$ corresponden al lugar de raíces complementario.

2.8 Sistemas bifrecuencia

Un sistema LTI bifrecuencia es aquel en el que la secuencia de entrada tiene un periodo de muestreo T_u; y la secuencia de salida, un periodo de muestreo T_y, siendo $T_u \neq T_y$.

Si:

$$\frac{T_u}{T_y} = \frac{M}{N}$$

Siendo N y M enteros –lo que es lo mismo que decir que T_y / T_u es un numero racional-, entonces el comportamiento del sistema puede caracterizarse por una matriz de funciones de transferencia:

$$y_l(z) = G_{lifted}(z)u_l(z)$$

Donde y_l es un vector de longitud M, u_i es un vector de longitud N, y G_{lfted} es una matriz de funciones de transferencia MxN [7].

La secuencia original y(k) y su versión *lifted* $y_l(k)$ se construyen de tal forma que el iésimo elemento de $y_l(k), y_{l,i}(k)$, es $y(k * M + i), i = 0, \dots, N - 1$.

Para recuperar la secuencia original desde la salida múltiple del sistema estirado (*lifted*) se usa un operador *expand,* que en el dominio de la transformada Z puede escribirse:

$$z \rightarrow z^N$$

Este operador inserta $N - 1$ ceros entre las muestras estiradas, con lo que la secuencia original y(z) se obtiene de $y_l(z)$ con:

$$y(z) = \left(1\ z^{-1}\ z^{-2} \dots\ z^{-(N-1)}\right) y_l(z^N)$$

2.9 Operadores monofrecuencia y bifrecuencia

Dado el esquema bifrecuencia en lazo abierto de la Figura 3:

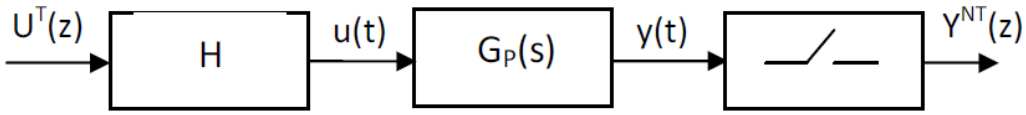


Figura 3: Sistema bifrecuencia en lazo abierto

Para cada periodo de muestreo se puede definir un modelo monofrecuencia:

El modelo discreto muestreado rápido (FSDT, *Fast Sampling Discrete Time*) se define de la siguiente manera:

$$\frac{Y(z)}{U(z)} = G(z) = [H_T G_p(s)]^T = \frac{B(z)}{A(z)} = \frac{\sum_{i=1}^{n} b_{i,T} z^{-i}}{1 + \sum_{i=1}^{n} a_{i,T} z^{-i}} = \frac{Y^T}{U^T}$$

Si denotamos los polos de la función de transferencia por $\alpha_{i,NT}$tenemos que:

$$A(z) = \prod_{i=1}^{n} (z - \alpha_{i,T}) = A^T$$

El modelo discreto muestreado lento (SSDT, *Slow Sampling Discrete Time*) está definido de la siguiente manera:

$$\frac{Y(z^N)}{U(z^N)} = G_N(z) = [H_{NT} G_p(s)]^{NT} = \frac{B_N(z^N)}{A_N(z^N)} = \frac{\sum_{i=1}^{n} b_{i,NT} z^{-iN}}{1 + \sum_{i=1}^{n} a_{i,NT} z^{-iN}} = \frac{Y^{NT}}{U^{NT}}$$

Si denotamos los polos de la función de transferencia por $\alpha_{i,NT}$, tenemos que:

$$A_N(z^N) = \prod_{i=1}^{n} (z^N - \alpha_{i,NT}) = A^{NT}$$

Para un mismo sistema continuo:

$$\alpha_{i,NT} = \alpha_{i,T}^N;\ \forall i = 1, \dots, n (n = orden\ del\ sistema)$$

De lo que se deriva la siguiente relación:

$$W_A(z) = W_A^T = \frac{[A^{NT}(z_N]^T}{A^T(z)} = \frac{[\prod_{i=1}^{n}(z^N - \alpha_{i,NT})]^T}{\prod_{i=1}^{n}(z - \alpha_{i,T})} =$$
$$= \prod_{i=1}^{n}(z^{N-1} + \alpha_{i,T}z^{N-2} + \cdots + \alpha_{i,T}^{N-1})$$

Es decir:

$$[A^{NT}(z_N)]^T = W_A^T A^T(z)$$

Para cualquier polinomio similar a A se puede obtener su correspondiente polinomio W.
Para evitar aliasing[2] [8] hay que tomar dos asunciones técnicas muy suaves:

a) En el FSDT, $G(z)$no tiene ningún cero en $z = 1$.
b) En el SSDT, si μ es un polo de $G_N(z)$, entonces $\mu e^{2\pi kj/N}$ no es un polo de $G_N(z)$.

El FSDT, por tanto, también puede ser expresado de la siguiente forma:

$$\frac{Y(z)}{U(z)} = G(z) = [H_T G_p(s)]^T = \frac{B(z)}{A(z)} = \frac{B^T}{A^T} = \frac{B^T W_A^T}{A^T W_A^T} = \frac{\tilde{B}^T(z)}{[A^{NT}(z_N)]^T} = \frac{Y^T}{U^T}$$

Los parámetros del numerador de $B^T(z)$ son distribuidos en n grupos de N coeficientes en los que la suma de cada uno de estos grupos es igual al coeficiente del numerador del regulador monofrecuencia lento. Así pues, es posible obtener el modelo SSDT a partir del modelo FSDT [9].

De la expresión anterior, se puede despejar lo siguiente:

$$\tilde{B}^T U^T = [A^{NT}]^T Y^T$$

Si a esta ecuación le aplicamos el operador *skip*:

$$\widetilde{[B^T U^T]}^{NT} = [[A^{NT}]^T Y^T]^{NT}$$

2 Aliasing: Fenómeno físico-visual que ocurre a veces al tener más de una frecuencia de muestreo coexistiendo en un mismo plano, y que genera una frecuencia "fantasma" cuyo efecto es el resultado de la suma de las dos.

Aplicando la propiedad del operador Kranc según la cual la operación skip no es conmutativa:

$$[\tilde{B}^T U^T]^{NT} = A^{NT}[Y^T]^{NT}$$

$$[\tilde{B}^T U^T]^{NT} = A^{NT} Y^{NT}$$

Nota: El término $[\tilde{B}^T U^T]^{NT}$ es indivisible ya que el operador *skip* no conmuta (si lo hiciese, a partir de la fórmula anterior podríamos definir una función de transferencia entre una entrada rápida y una salida lenta, lo cual es imposible).

Vamos a considerar ahora la situación contraria: La transformación de una secuencia de datos de frecuencia lenta a una secuencia de datos de frecuencia rápida. Esto es necesario porque el bloqueador de orden o bifrecuencia (DRZOH, *Dual Rate Zero Order Hold*) está formado por un bloqueador de orden cero lento ZOH, cuya entrada será U^{NT} seguido de un muestreador rápido cuya salida será U_H^T. La función de transferencia del DRZOH es:

$$[H_{NT}(s)]^T = \frac{U_H^T}{[U^{NT}]^T} = [\frac{1 - e^{-NTS}}{s}]^T = \frac{1 - z^{-N}}{1 - z^{-1}} = 1 + z^{-1} + \cdots + z^{-(N-1)}$$

Este resultado también se puede obtener si al polinomio $R^-(z) = 1 - z^{-1}$ se le aplica la transformación W definida en la página anterior. De esta forma:

$$[H_{NT}(s)]^T = W_R^T -$$

En este caso, sí que es posible obtener una función de transferencia del proceso más el dispositivo DRZOH, ya que el operador *expand* sí es conmutativo:

$$Y^T = [H_{NT} G_p(s) U^{NT}]^T = [H_{NT} G_p(s)]^T [U^{NT}]^T$$

Si usamos transformación W:

$$[H_{NT} G_p(s)]^T = W_R^T - [H_T G_p(s)]^T$$

A partir de esto, se puede definir un operador bifrecuencia discreto (DRDT, *Dual Rate*
Discrete Time) para expresar la relación entre dos secuencias diferentemente espaciadas en el tiempo:

$$G^{T,NT} \triangleq \frac{Y^T}{[U^{NT}]^T} = W_R^T - [H_T G_p(s)]^T = W_R^T - G^T = W_R^T - \frac{B^T}{A^T} = W_R^T - \frac{B^T W_A^T}{A^T W_A^T} = \frac{W_R^T - \tilde{B}^T}{[A^{NT}]^T}$$

Siendo $G^{T,NT}$ la "función de transferencia" que relaciona una entrada lenta (NT) expandida y una salida rápida (T).

Usando una notación similar a la usada para describir los modelos SSDT y FSDT, el operador DRZOH se representa con $H^{T,NT}$.

Si al operador DRDT se le aplica un *skip*, se obtiene el operador SSDT (ver [9] para detalles).

$$[G^{T,NT}]^{NT} = \frac{[W_R^T - \tilde{B}^T]^{NT}}{[[A^{NT}]^T]^{NT}} = \frac{[Y^T]^{NT}}{[[U^{NT}]^T]^{NT}} = \frac{[Y^T]^{NT}}{U^{NT}} = \frac{B^{NT}}{A^{NT}} = G_N(z)$$

2.10 Control multifrecuencia y modelos muestreados

El esquema básico de control multifrecuencia en lazo cerrado se muestra en la Figura 4:

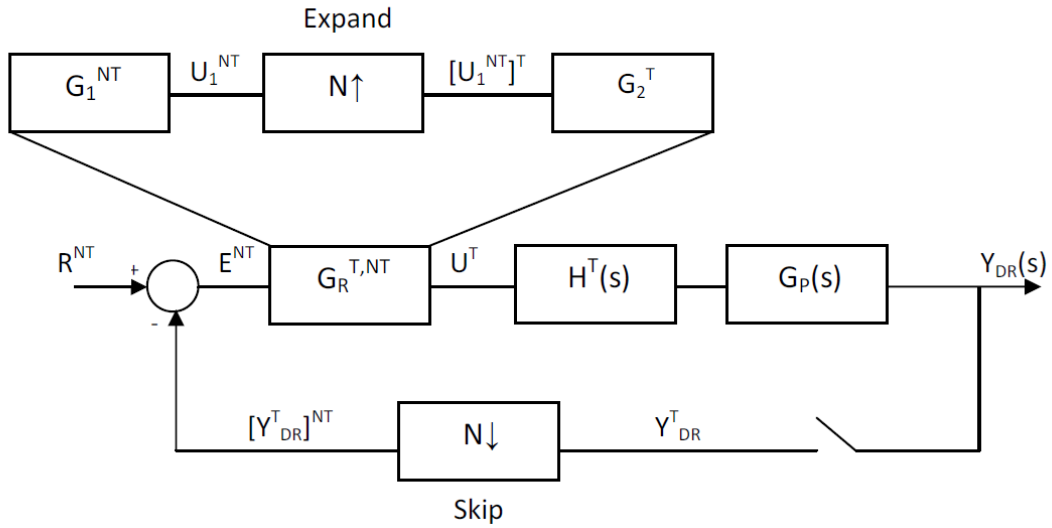


Figura 4: Sistema bifrecuencia en lazo cerrado

En este esquema, la planta se representa por un sistema continuo LTI SISO de orden n, con función de transferencia $G_p(s)$. La salida del regulador bifrecuencia $G_{R,N}^T$ es actualizada a período T a través del dispositivo de retención rápido H. La salida continua $Y(s)$ se discretiza a período NT y se compara con la referencia $R(t)$, que está siempre disponible, obteniéndose el error para el regulador. Para simplificar el cómputo, y sin pérdida de generalidad, se asume que N es un número entero.

El regulador bifrecuencia estará compuesto por tres partes:
1. El error calculado ($R^{NT} - Y^{NT}$) es procesado por la parte lenta del regulador G_1^{NT}.
2. La salida de la parte anterior pasa a través de un bloque Expand para convertir
la secuencia a periodo de muestreo T.
3. Finalmente, la parte rápida del regulador -G_2^{NT}– junto con un bloque que asegura que la conversión de frecuencias es correcta –H_N^T.– proporciona la entrada de control al proceso.

Teorema 1 [10]: Usando la misma notación que en el apartado 4.1 y que en el esquema de la figura anterior, la salida del proceso en un sistema con un regulador bifrecuencia (*DR, Dual Rate*) viene expresada por:

$$Y_{DR}^{T}(z) = [H_T G_p(s)]^T G_R^{T,NT}[R^{NT} - [Y_{DR}^{T}]^{NT}]^T(z) = G^T G_R^{T,NT}[R^{NT} - [Y_{DR}^{T}]^{NT}]^T(z)$$
$$= G^T G_2^T [G_1^{NT}]^T [R^{NT} - [Y_{DR}^{T}]^{NT}]^T(z)$$

Demostración: Para el sistema representado en la figura anterior, es posible expresar:

$$Y_{DR}^{T}(z) = G^T(z)U^T(z);\ U^T(z) = G_R^{T,NT}(z)[E^{NT}]^T(z^N)$$

Sabiendo que el operador *expand* es conmutativo y usando álgebra básica de bloques:

$$U_1^{NT} = G_1^{NT} E^{NT}$$

$$[U_1^{NT}]^T = [G_1^{NT} E^{NT}]^T [E^{NT}]^T$$

$$U^T = G_2^T [U_1^{NT}]^T = G_2^T [G_1^{NT}]^T [E^{NT}]^T$$

$$E^{NT} = R^{NT} - [Y_{DR}^{T}]^{NT}$$

Combinando estas expresiones se obtiene la formulación buscada.

Conviene remarcar lo siguiente:

- Como se ha dicho antes, el operador *skip* no es conmutativo. Esto implica que, aunque la función de transferencia rápida $Y_{DR}^{T} = M^T R^T$ se puede definir, no es posible obtener una relación entre Y_{DR}^{T} y R^{NT}.
- Es necesario un operador *expand* como parte del regulador bifrecuencia.

2.11 Diseño de reguladores multifrecuencia

A partir del modelo descrito en el apartado anterior se va a plantear la cuestión de cómo diseñar un regulador bifrecuencia. Dado un proceso continuo $G_p(s)$ el objetivo principal es diseñar un controlador bifrecuencia para obtener un comportamiento en bucle cerrado lo más parecido posible al conseguido con una función de transferencia en bucle cerrado $M(s)$.

Es bien sabido por todos que para diseñar un regulador discreto clásico monofrecuencia, se pueden seguir las siguientes opciones:

1) Diseñar un regulador continuo $G_R(s)$ y discretizarlo.
2) Diseñar directamente un controlador discreto para que el sistema en bucle cerrado siga el comportamiento discreto de *M(s)*.

En el primer caso, vamos a denotar con G_R^T *y* G_R^{NT} a los equivalentes discretos de $G_R(s)$para periodos de muestreo T o NT, respectivamente. Las funciones de transferencia en bucle cerrado de la planta más el regulador serán las siguientes:

$$M_R^T(z) = \frac{Y_R^T(z)}{R^T(z)} = \frac{G^T(z)G_R^T(z)}{1 + G^T(z)G_R^T(z)}$$

$$M_R^{NT}(z) = \frac{Y_R^{NT}(z_N)}{R^{NT}(z_N)} = \frac{G^{NT}(z_N)G_R^T(z_N)}{1 + G^{NT}(z_N)G_R^T(z_N)}$$

Ninguna de estas dos funciones se ajustará a los correspondientes equivalentes discretos de *M(s)*. Por lo tanto, ninguna de las correspondientes salidas tiene por qué ajustarse a la salida continua equivalente.

En el segundo caso, los controladores monofrecuencia correspondientes son:

$$M^T(z) = \frac{G^T(z)\bar{G}_R^T(z)}{1 + G^T(z)\bar{G}_R^T(z)}$$

$$M^T(z) + M^T(z)G^T(z)\bar{G}_R^T(z) = G^T(z)\bar{G}_R^T(z)$$

$$\bar{G}_R^T(z) = \frac{1}{G^T(z)}\frac{M^T(z)}{1 - M^T(z)}$$

$$M^{NT}(z_N) = \frac{G^{NT}(z_N)\bar{G}_R^{NT}(z_N)}{1 + G^{NT}(z_N)\bar{G}_R^{NT}(z_N)}$$

$$M^{NT}(z_N) + M^{NT}(z_N)G^{NT}(z_N)\bar{G}_R^{NT}(z_N) = G^{NT}(z_N)\bar{G}_R^{NT}(z_N)$$

$$\bar{G}_R^{NT}(z_N) = \frac{1}{G^{NT}(z_N)}\frac{M^{NT}(z_N)}{1 - M^{NT}(z_N)}$$

Ambas funciones de transferencia tienen dos términos claramente diferenciados. El primer término es la inversa de la función de transferencia de la planta; y el segundo es función únicamente de la función de transferencia deseada para el sistema. Es evidente, por tanto, que estamos ante controladores de cancelación, con todas las restricciones que esto conlleva (condiciones para la estabilidad, oscilaciones ocultas,…)

En este segundo caso las respuestas de los sistemas obtenidos ante las referencias $R^T(z)$ y $R^{NT}(z_N)$ (respectivamente) coincidirán con las correspondientes secuencias discretas tomadas del sistema continuo inicial. Es decir:

$$M^T(z) = \frac{Y_c^T(z)}{R^T(z)} = \frac{\bar{Y}^T(z)}{R^T(z)}$$

$$M^{NT}(z_N) = \frac{Y_c^{NT}(z_N)}{R^{NT}(z_N)} = \frac{\bar{Y}^{NT}(z_N)}{R^{NT}(z_N)}$$

El objetivo es diseñar un regulador multifrecuencia para alcanzar el rendimiento del sistema controlado a frecuencia rápida $M^T(z)$ a partir de una secuencia de medidas de la salida a frecuencia lenta. El resultado se puede expresar como sigue:

Teorema 1 [1]. Dado un proceso continuo $G_p(s) = B(s)/A(s)$ y un modelo de referencia $M(s)$ para el sistema controlado, asumiendo una frecuencia de actualización de control $1/T$, y la salida muestreada a frecuencia $1/NT$:

a) La salida del sistema bifrecuencia cuando en la entrada hay una referencia $r(t)$ discretizada a frecuencia rápida $R^T(z)$ coincidirá con la salida del sistema monofrecuencia rápido si el controlador bifrecuencia está compuesto de una parte lenta dada por $1/R^{NT}(z_N) - [M^T R^T]^{NT}(z_N))$ y una parte rápida dada por $M^T(z)R^T(z)/G^T(z)$. Esto es:

$$G_R^{T,NT}(z) = \frac{M^T(z)R^T(z)}{G^T(z)}\left[\frac{1}{R^{NT}(z_N) - [M^T R^T]^{NT}(z_N)}\right]^T(z)$$

b) Más aún, si la salida rápida una vez que ha pasado por el bloque *skip*, $[Y_{DR}^T]^{NT}$, fuese la misma que la salida del esquema en bucle cerrado monofrecuencia lento, $\bar{Y}^{NT}(z_N) = Y_c^{NT}(z_N)$ entonces las partes del controlador serán las siguientes:

- Una parte rápida dada por:

$$G_2^T(z) = \frac{M^T(z)}{G^T(z)}$$

- Una parte lenta dada por:

$$G_1^{NT}(z_N) = \frac{1}{1 - M^{NT}(z_N)}$$

- Un convertidor de frecuencia puesto después del bloque expand y dado por:

$$H^{T,NT}(z) = \frac{R^T(z)}{[R^{NT}]^T(z)}$$

Comentarios

1) Para cambios en escalón en la referencia, el regulador bifrecuencia alcanzará el mismo rendimiento discreto que el regulador rápido, pero utilizando una frecuencia de muestreo lenta para la salida.

2) Se puede evitar el rizado intermuestreo si en la parte rápida del controlador $G_2^T(z)$ se sustituye $M^T(z)$ por $M_R^T(z)$pero la respuesta no tiene por qué coincidir con la respuesta continua.

3) Resultados similares se pueden derivar para otras referencias de entrada con un convertidor de frecuencia generalizado $H_{\lambda,\mathrm{N}}^T$.

2.12 Algoritmo de diseño

Como conclusión, el siguiente algoritmo resume el proceso de diseño:

Dado el modelo de la planta $G_p(s)$, la función de transferencia en bucle cerrado cuyo comportamiento se pretende imitar, $M(s)$, la entrada al sistema, y los periodos de muestreo rápido (T) y lento (NT); se diseña un controlador multifrecuencia con tres partes:

- Un controlador rápido
- Un controlador lento
- Un convertidor de frecuencia

Para lo cual se usan las discretización rápida de la planta, las discretizaciones lenta y rápida de $M(s)$ y la entrada al sistema.

Si la planta es inestable en bucle abierto, de fase no mínima o aparece algún rizado intermuestreo debido a las cancelaciones planta-regulador, se puede hacer lo siguiente:

Calcular el controlador de cancelación continuo:

$$G_R(s) = \frac{1}{G_p(s)} \frac{M(s)}{1 - M(s)}$$

Y discretizarlo para obtener $G_R^T(z)$ $M(s)$ deberá ser escogido de tal forma que no haya cancelación de polos inestables en este controlador.

3. Controladores

En este apartado se presentan los cuatro controladores que se han utilizado para la realización de este proyecto y el modo de conseguir el control sobre un sistema de cada uno.

3.1 Controlador PID

Un controlador PID es aquel en el que para una entrada $e(t)$ (señal de error), su salida es la combinación lineal de tres acciones básicas de control:

Acción proporcional (P)

La señal de control es proporcional a la señal de error $e(t)$:

$$u_p(t) = K_p e(t)$$
$$U_p(s) = K_p E(s)$$

La acción proporcional $u_p(t)$ varía cada vez que se modifica $e(t)$.
Alcanza un valor estacionario cuando lo alcanza $e(t)$.

Acción integral (I)

La variación de la señal de control es proporcional a la señal de error $e(t)$:

$$\frac{du_i(t)}{dt} = K_i e(t)$$

$$U_i(s) = \frac{K_i}{s} E(s)$$

La acción integral $u_i(t)$ es acumulativa, tiene en cuenta la historia pasada de $e(t)$ y sólo puede tener un valor estacionario cuando $e(t) = 0$.

Acción derivativa (D)

La señal de control es proporcional a la variación de la señal de error $e(t)$:

$$u_d(t) = K_d \frac{de(t)}{dt}$$
$$U_d(s) = K_d s E(s)$$

La acción derivativa $u_d(t)$ es anticipativa, tiene en cuenta las variaciones instantáneas de $e(t)$.

Se anula cuando $e(t)$ alcanza un valor estacionario.

Acción PID

El diseñador del sistema de control debe lograr una combinación adecuada de estas tres acciones. Es decir, debe determinar las tres constantes K_p, K_i, K_d del algoritmo PID para que el sistema cumpla las especificaciones pedidas.

El algoritmo de control PID es:

$$u(t) = u_p(t) + u_i(t) + u_d(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

$$U(s) = \left[K_p + K_d s + \frac{K_i}{s}\right] E(s)$$

Luego la función de transferencia del PID es:

$$G(s) = \frac{U(s)}{E(s)} = [K_p + K_d s + \frac{K_i}{s}]$$

Y en la práctica se usa:

$$G(s) = K_p(1 + T_d s + \frac{1}{T_i s})$$

Donde T_d y T_i representan las constantes de tiempo derivativa e integral:

$$T_d = \frac{K_d}{K_p} \qquad T_i = \frac{K_p}{K_i}$$

Discretización del PID

A partir de un PID continuo se puede obtener un PID discreto mediante la siguiente aproximación [11]:

$$G(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}}$$

Con:

$$q_0 = K_p \left(1 + \frac{T_d}{T}\right)$$

$$q_1 = -K_p(1 - \frac{T}{T_i} + 2\frac{T_d}{T})$$

$$q_2 = K_p \frac{T_d}{T}$$

Siendo T el periodo de muestreo.

Particularización de la estructura del controlador bifrecuencia para un PID

Una vez que la estructura del controlador bifrecuencia ha quedado explicada, se va a presentar su particularización para trabajar con reguladores tipo PID. En estos casos, la descomposición del controlador es la siguiente [12]:

Una parte lenta (periodo de muestreo = *NT*) dada por un controlador PI el cual, basándonos en métodos clásicos de discretización, viene dado por:

$$G_{PI}(z_N) = K_{PI} \frac{z_N - (1 - \frac{NT}{T_i})}{z_N - 1}$$

Una parte rápida (periodo de muestreo = *T*) dada por un controlador PD el cual, basándonos en métodos clásicos de discretización, viene dado por:

$$G_{PD}(z) = K_{PD} \frac{z\left(1 + \frac{T_d}{T}\right) - \frac{T_d}{T}}{z}$$

Un convertidor de frecuencias situado entre las dos partes del controlador dado por:

$$H^{T,NT}(z) = \frac{R^T(z)}{[R^{NT}]^T(z)}$$

3.2 Cancelación

Un controlador de cancelación es aquel que cancela completamente la dinámica de proceso en aras de seguir un modelo de referencia en el que se definen las prestaciones deseadas para el bucle de control.

Siendo $G_p(s)$ el proceso, $G_R(s)$ el regulador y $M(s)$ el modelo, la obtención del controlador $G_R(s)$ se hace del siguiente modo:

$$G_R(s) = \frac{M(s)}{1 - M(s)} \frac{1}{G_p(s)}$$

Para nuestro trabajo, es necesaria la creación de un controlador de cancelación para un sistema bifrecuencia. Este punto se ha explicado en el apartado anterior, Diseño de Reguladores Multifrecuencia.

3.3 Controlador RST

El controlador RST es un controlador de dos grados de libertad obtenido vía un procedimiento de ubicación de polos con diseño basado en modelo entrada-salida. En este documento, para evitar confusiones, los polinomios del RST serán nombrados como $\bar{R}\bar{S}\bar{T}$ (pues R se utiliza para la señal de referencia, y T para el periodo de muestreo). Los polinomios $\bar{R}$, $\bar{S}$ se obtienen resolviendo la ecuación diofántica, El polinomio $\bar{T}$ está diseñado para ajustar la ganancia del sistema de control , y para evitar cancelaciones cero inestables. El controlador $\bar{R}\bar{S}\bar{T}$ propuesto se diseña a un periodo NT, y se define así:

$$\bar{U}^{NT} = \frac{\bar{T}^{NT}(z_N)}{\bar{R}^{NT}(z_N)} R^{NT} - \frac{\bar{S}^{NT}(z_N)}{\bar{R}^{NT}(z_N)} [\tilde{\tilde{Y}}_{DR}^{T}]^{NT}$$

El procedimiento general para obtener esta ley se puede encontrar en [13]. Vamos a detallar un caso particular (pero habitual) basado en la consideración de un proceso discreto de segundo orden con un cero:

Si el proceso SSDT tiene esta forma:

$$G^{NT}(z_N) = \frac{k(z_N + b)}{(z_N - a_1)(z_N - a_2)} - \frac{k(z_N + b)}{(z_N^2 + \alpha_1 z_N + \alpha_2)} = \frac{B^{NT}(z_N)}{A^{NT}(z_N)}$$

El modelo de referencia del SSDT (incluyendo el cero del proceso para evitar problemas de ondulación) se puede proponer como:

$$M^{NT}(z_N) = \frac{(1 + \bar{t}_1 + \bar{t}_2)}{(1 + b)} \frac{(z_N + b)}{(z_N^2 + \bar{t}_1 z_N + \bar{t}_2)} = \frac{B_M^{NT}(z_N)}{A_M^{NT}(z_N)} = \frac{Y^{NT}(z_N)}{R^{NT}(z_N)}$$

Entonces, la ubicación del polo sigue la siguiente ecuación diofántica.

$$A^{NT}(z_N)\bar{R}^{NT}(z_N) + B^{NT}(z_N)\bar{S}^{NT}(z_N) = A_M^{NT}(z_N)A_0^{NT}(z_N)$$

Donde $A_0^{NT}(z_N)$ es el polinomio del observador (teniendo sus polos localizados normalmente en $z_N = 0$), y $\bar{R}$, $\bar{S}$ tendrán la siguiente forma:

$$\bar{R}^{NT}(z_N) = (z_N + \bar{r}_1)$$

$$\bar{S}^{NT}(z_N) = (\bar{s}_0 z_N + \bar{s}_1)$$

Así pues, resolviendo la ecuación diofántica:

$$\bar{s}_0 = \frac{\bar{t}_1 - \alpha_1 - \bar{r}_1}{k}$$

$$\bar{s}_1 = \frac{\alpha_2 \bar{r}_1}{kb}$$

$$\bar{r}_1 = \frac{\bar{t}_2 - \alpha_2 + b\alpha_1 - bt_1}{\alpha_1 - \frac{\alpha_2}{b} - b}$$

Siendo

$$\bar{T}^{NT}(z_N) = \frac{1 + \bar{t}_1 + \bar{t}_2}{k(1+b)} z_N$$

Nota 1: Tal como se dice en [13], las perturbaciones no son usadas en el diseño de la ubicación de polos. Estos pueden ser incluidos indirectamente como restricciones en el modelo de referencia, en el polinomio del observador, y en la ley de control.

Controlador multifrecuencia no uniforme

Una vez el controlador RST ha sido definido, el objetivo es redefinir el controlador multifrecuencia para incluir la fase RST. Como se ha comentado, el controlador multifrecuencia será diseñado vía un procedimiento basado en modelo [10]. La fase de control del RST será incluida en el lado lento del controlador multifrecuencia, siendo obtenida vía ubicación de polos a través de un método basado en modelo entrada-salida [13] (como se detalla en la sección anterior).

Teoría: Dado un orden n y un proceso continuo de una entrada y una salida $G_p(s)$, aquellos comportamientos de bucle cerrado deben seguir el siguiente modelo de referencia continuo $M(s)$, asumiendo un control no periódico uniforme acorde a los subperiodos τ_i, y la salida siendo muestreada a un periodo 1/NT; si la salida rápida $[\tilde{\tilde{Y}}_{DR}^{T}]^{NT}$ debe encajar con la salida del bucle de control lento $Y^{NT}(z_N)$, esto es, la señal muestreada lenta, entonces el controlador debería ser:

$$G_R^{T,NT}(z) = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{\propto \tilde{\tilde{B}}^T(z)} \frac{R^T(z)}{[R^{NT}]^T(z)} [G_{RST}^{NT}(z_N) \frac{B^{NT}(z_N)}{B_M^{NT}(z_N)}]^T(z)$$

Donde:

-$G_2^T(z) = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{\propto \tilde{\tilde{B}}^T(z)}$ es el lado rápido del controlador multifrecuencia, siendo $\propto_M \tilde{\tilde{B}}_M^T(z)$ el numerador de la FSDT del modelo de referencia (ya que ha sido definido como $\tilde{\tilde{M}}^T(z) = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{[A_M^{NT}]^T(z)}$) y $\propto \tilde{\tilde{B}}^T(z)$ el numerador de la FSDT del modelo del proceso.

-$G_1^{NT}(z_N) = \frac{B^{NT}(z_N)}{B_M^{NT}(z_N)}$ es el lado lento del controlador multifrecuencia, donde $B^{NT}(z_N)$ es el numerador de la SSDT del modelo de referencia, y $B^{NT}(z_N)$ el numerador de la SSDT del modelo del proceso.

-$G_{RST}^{NT}(z_N)$ representa el controlador RST.

-$H^{T,NT}(z) = \frac{R^T(z)}{[R^{NT}]^T(z)}$ es un convertidor de frecuencia localizado después de expandir el bloque. Depende de la señal de referencia, Así, por una entrada escalón, actua como un ZOH digital, entonces $H^{T,NT}(z) = \frac{(1-z^{-N})}{(1-z^{-1})}$.

Nótese que la operación de expansión se aplica al lado lento de $G_R^{T,NT}(z)$, esto es, $[G_{RST}^{NT}(z_N)G_1^{NT}(z_N)]^T(z)$ (ver Figura 1).

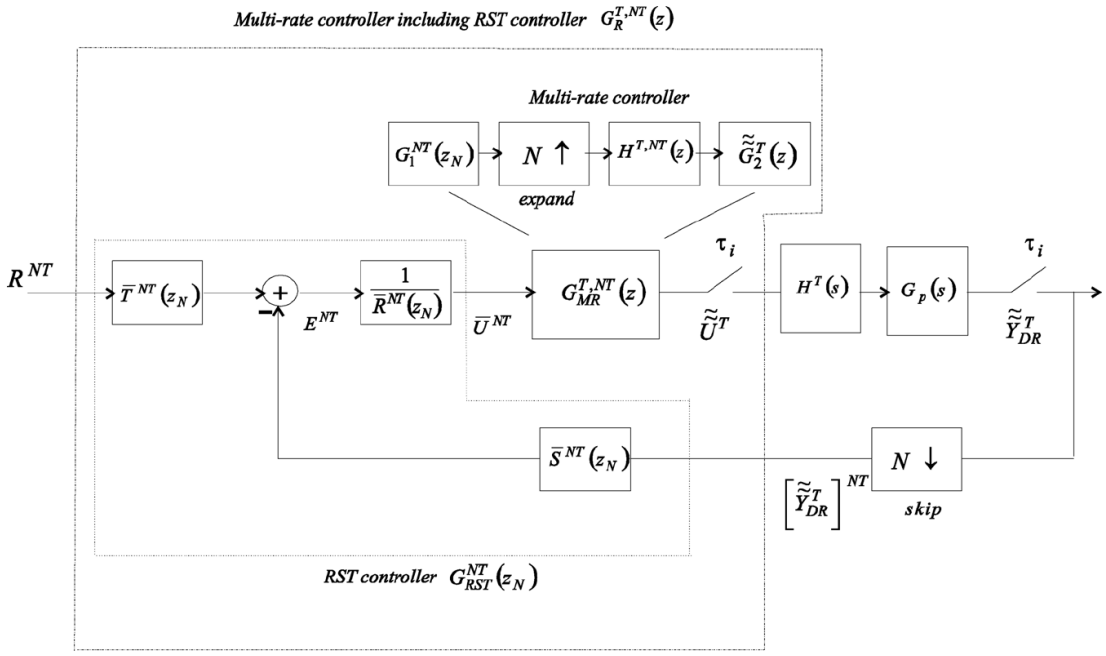


Figura 1: Diagrama de bloques para el sistema de control: controlador multifrecuencia incluyendo RST como fase.

Para el caso particular en el cual se considere una actualización uniforme del control a periodo T, el controlador debería ser:

$$G_R^{T,NT}(z) = \frac{\tilde{\tilde{B}}_M^T(z)}{\tilde{\tilde{B}}^T(z)} \frac{R^T(z)}{[R^{NT}]^T(z)} [G_{RST}^{NT}(z_N) \frac{B^{NT}(z_N)}{B_M^{NT}(z_N)}]^T(z)$$

Donde sólo el lado rápido cambia ($G_2^T(z) = \frac{\tilde{\tilde{B}}_M^T(z)}{\tilde{\tilde{B}}^T(z)}$), siendo $\tilde{\tilde{B}}_M^T(z)$ el numerador de la FSDT del modelo de referencia (ya que se ha definido en este caso como $M^T(z) = \frac{\tilde{B}_M^T}{[A_N^{NT}]^T}(z)$), y $\tilde{\tilde{B}}^T(z)$ el numerador del FSDT del modelo del proceso.

Prueba: Del resultado sobre controladores multifrecuencia uniformes [14]:

$$G_R^{T,NT}(z) = \frac{M^T(z)}{G^T(z)} \frac{R^T(z)}{[R^{NT}]^T(z)} [\frac{1}{1 - M^{NT}(z_N)}]^T(z)$$

Donde

- $G_2^T(z) = \frac{M^T(z)}{G^T(z)}$ es el lado rápido del controlador multifrecuencia.

-$[G_1^{NT}(z_N)]^T(z) = [\frac{1}{1-M^{NT}(z_N)}]^T(z)$ es el lado lento del controlador multifrecuencia expandido.

-$H^{T,NT}(z) = \frac{R^T(z)}{[R^{NT}]^T(z)}$ es el convertidor de frecuencia localizado después del bloque de expansión.

Si el SSDT del modelo de referencia en se redefine como:

$$M^{NT}(z_N) = \frac{G_{RST}^{NT}(z_N)G^{NT}(z_N)}{1 + G_{RST}^{NT}(z_N)G^{NT}(z_N)}$$

Entonces el lado lento del controlador multifrecuencia puede ser re-escrito de este modo:

$$\frac{1}{1 - M^{NT}(z_N)} = G_{RST}^{NT}(z_N)\frac{G^{NT}(z_N)}{M^{NT}(z_N)}$$

Así, el controlador multifrecuencia puede ser redefinido del siguiente modo:

$$G_R^{T,NT}(z) = \frac{M^T(z)}{G^T(z)}\frac{R^T(z)}{[R^{NT}]^T(z)}[G_{RST}^{NT}(z_N)\frac{G^{NT}(z_N)}{M^{NT}(z_N)}]^T(z)$$

Donde

-$G_1^{NT}(z_N) = \frac{G^{NT}(z_N)}{M^{NT}(z_N)}$ es el nuevo lado lento del controlador multifrecuencia.

-$G_{RST}^{NT}(z_N)$ representa el controlador RST.

Aplicando al lado lento $G_1^{NT}(z_N)$ tanto el operador de expansión y la definición para $G^T(z)$, este lado lento puede ser adaptado de este modo:

$$[G_1^{NT}(z_N)]^T = [\frac{G_1^{NT}(z_N)}{[M^{NT}(z_N)]^T}]^T = \frac{[B^{NT}(z_N)]^T}{[A^{NT}(z_N)]^T}\frac{[A_M^{NT}(z_N)]^T}{[B_M^{NT}(z_N)]^T} = \frac{[B^{NT}(z_N)]^T}{[B_M^{NT}(z_N)]^T}\frac{A_M^T(z)W_{MA}^T}{A^T(z)W_A^T}$$

Adicionalmente, al usar los modelos no uniformes, el lado rápido $\tilde{\tilde{G}}_2^T(z)$ puede ser representado por:

$$\tilde{\tilde{G}}_2^T(z) = \frac{\tilde{\tilde{M}}^T(z)}{\tilde{\tilde{G}}^T(z)} = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{A_M^T(z)W_{MA}^T}\frac{A^T(z)W_A^T}{\propto \tilde{\tilde{B}}^T(z)}$$

Así pues, ahora que ambos lados del controlador multifrecuencia están ahora definidos al mismo periodo, pueden multiplicarse:

$$\tilde{\tilde{G}}_2^T(z)[G_1^{NT}(z_N)]^T = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{A_M^T(z)W_{MA}^T}\frac{A^T(z)W_A^T}{\propto \tilde{\tilde{B}}^T(z)}\frac{[B^{NT}(z_N)]^T}{[B_M^{NT}(z_N)]^T}\frac{A_M^T(z)W_{MA}^T}{A^T(z)W_A^T} = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{\propto \tilde{\tilde{B}}^T(z)}\frac{[B^{NT}(z_N)]^T}{[B_M^{NT}(z_N)]^T}$$

Resultando:

-$G_1^{NT}(z_N) = \frac{B^{NT}(z_N)}{B_M^{NT}(z_N)}$ (sin incluir la operación expansión)

-$\tilde{\tilde{G}}_2^T(z) = \frac{\propto_M \tilde{\tilde{B}}_M^T(z)}{\propto \tilde{\tilde{B}}^T(z)}$, o $G_2^T(z) = \frac{\tilde{B}_M^T(z)}{\tilde{B}^T(z)}$ (dependiendo del tipo de patrón de muestreo)

Nota 2: $\tilde{\tilde{G}}_2^T(z)$ o $G_2^T(z)$ contienen en el denominador el numerador del modelo del proceso. Entonces, por otro lado, si este modelo no tiene fase mínima, se generará una señal de control inestable. Así, el controlador multifrecuencia propuesto no es apropiado para este caso. En [15], el lector puede encontrar un caso de controlador bifrecuencia adecuado para sistemas inestables y sin fase mínima. Por otro lado, si el proceso está en fase mínima, puede definirse una región de seguridad en el plano complejo que incluysa algunos valores absolutos y relativos deseados. Así, para evitar problemas de cancelación sólo aquellos ceros del modelo que se encuentren dentro de esta región serán cancelados [13].

Nota 3: El diseño de controlador multifrecuencia propuesto sólo implica el diseño del controlador a un periodo NT, ya que ambos lados, lento y rápido, son obtenidos a través del modelado correspondiente. Esto reduce mucho la complejidad de la fase de diseño y de la implementación del controlador.

3.4 PID Basado en Modelo

Dado un proceso continuo $G_p(s)$, diseñaremos un controlador continuo $G_R(s)$. Si se aplica un PID discretizado [11], el rendimiento dinámico del proceso controlado se ve degradado si se escoge un muestreo lento. Asumiendo que el comportamiento discreto de la planta controlada es aceptable para un periodo de muestreo T, con el controlador rápido G_R^T, pero demasiado pobre si se escoge un muestreo NT, G_R^{NT}. La pregunta en la configuración del multifrecuencia es: ¿si la entrada pudiera ser actualizada a un periodo 1/T, es posible obtener un rendimiento similar si la salida se mide a NT?

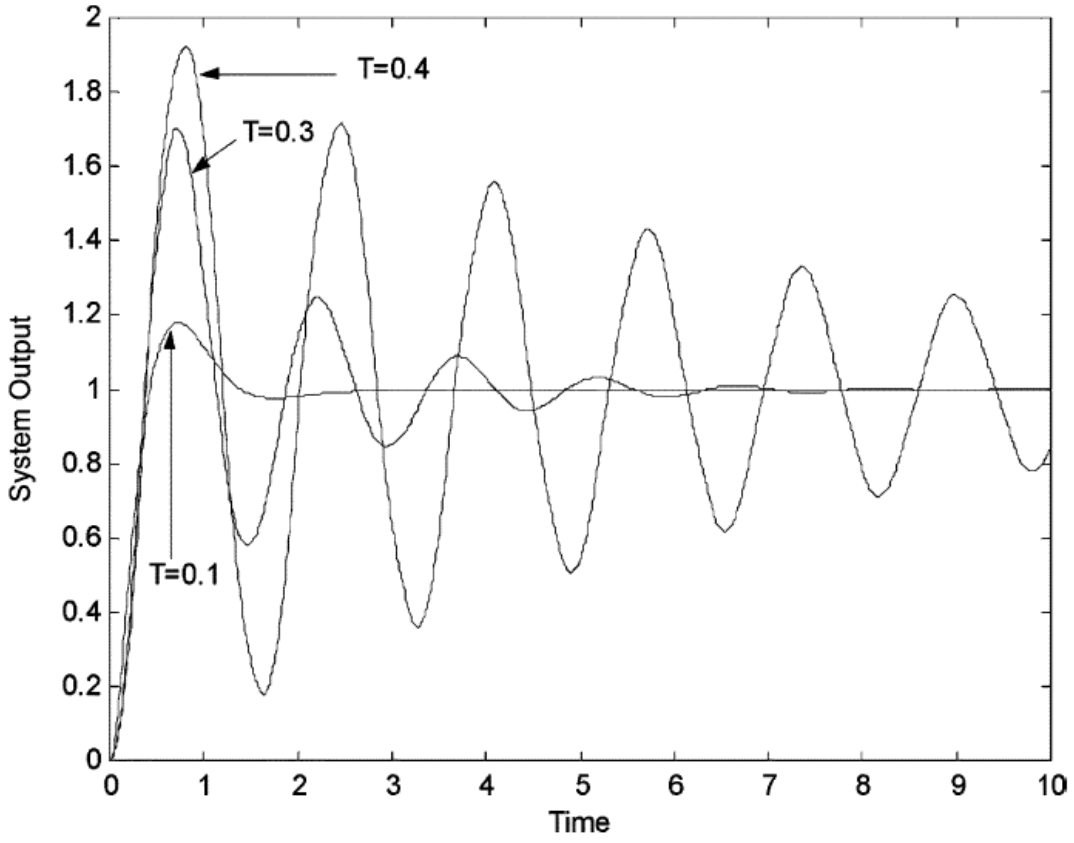


Figura 2: Control PID

Siguiendo con el diseño propuesto, la función de transferencia de la planta controlada por el PID, $M(s)$ se presupone como el objetivo de diseño. En este caso, la parte rápida del controlador multifrecuencia, G_2^T, adapta la dinámica del proceso de tal modo que el nuevo proceso puede ser controlado por una parte discreta lenta, G_1^{NT}.

Como ejemplo, considerando el proceso:

$$G_p(s) = \frac{1.5}{(s+0.5)(s+1.5)}$$

Un controlador PID continuo aceptado podría ser:

$$u(t) = K_p[e(t) + T_D e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau]$$

Para $K_p = 8$, $T_D = 0.2$, $T_I = 3.2$. Una aproximación para un controlador PID discretizado a T sería:

$$G_R^T(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^2}{1 - z^{-1}}$$

$$q_0 = K_p(1 + \frac{T_D}{T})$$

$$q_1 = K_p(1 + \frac{T}{T_I} + 2\frac{T_D}{T})$$

$$q_2 = K_p(\frac{T_D}{T})$$

Para un periodo de muestreo distinto la respuesta del sistema controlado se muestra en la Figura 2, haciéndose inestable para $T = 0.5s$.

Asumiendo que el periodo de muestreo de la salida no puede ser menor que 0.3 s se diseña e implementa un controlador bifrecuencia, con $T = 0.1$, $N = 3$. La primera solución es implementar un controlador de cancelación uniendo una parte lenta y una parte rápida.

$$G_1^{NT}(z_N) = \frac{1}{1 - M^{NT}(z_N)}$$

$$G_2^T(z) = \frac{M^T(z)}{G^T(z)}$$

La respuesta encaja los puntos del sistema controlado FSDT y SSDT, tal como se expresa en la parte b) del resultado principal, pero aparecen oscilaciones, como se muestra en la Figura 3, donde a) es la respuesta en bucle cerrado para $M^T(z)$, b) corresponde a $M^{NT}(z_N)$ y c) es la respuesta de la planta controlada multifrecuencia a una entrada escalón.

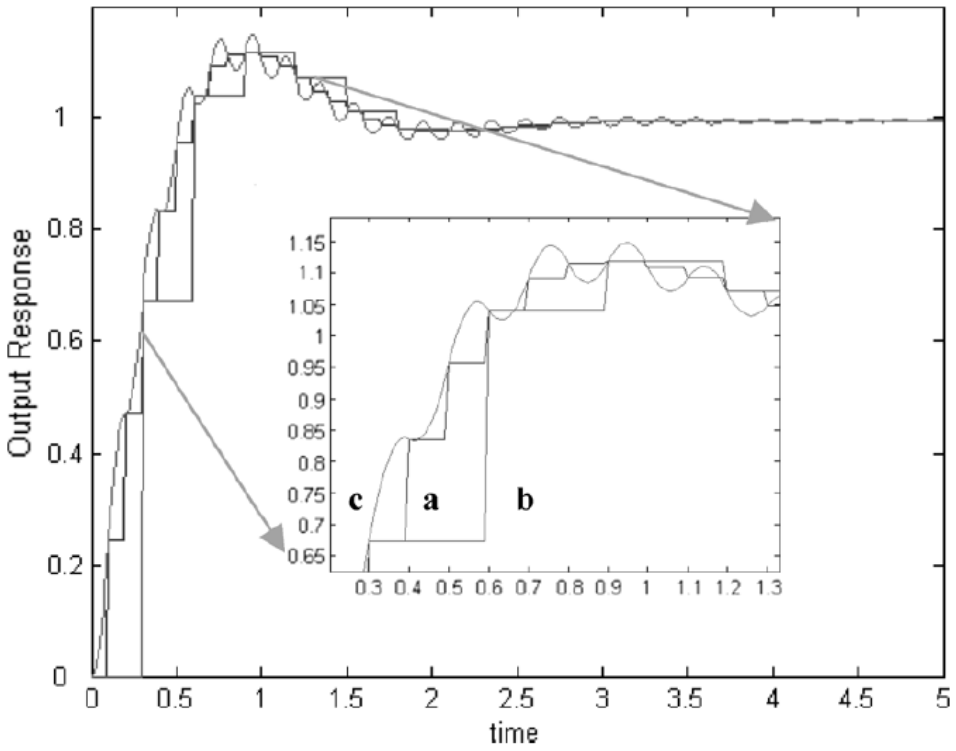


Figura 3: Control de cancelación multifrecuencia basado en modelo.

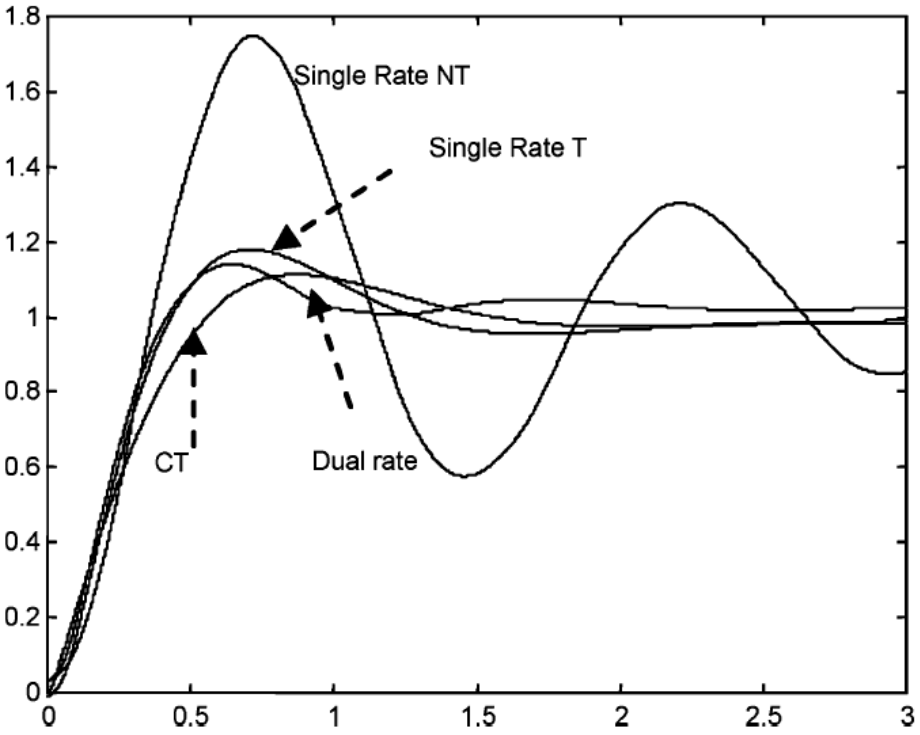


Figura 4: Planta controlada por PID

Para evitar las oscilaciones, la parte rápida del controlador multifrecuencia se computa del siguiente modo:

$$G_2^T(z) = \frac{M_R^T(z)}{G^T(z)} = \frac{G_R^T(z)}{1 + G_R^T(z)G^T(z)}$$

Los excelentes resultados se muestran en la Figura 4, donde las respuestas a una entrada escalón para la planta controlada por el PID para un controlador continuo, el periodo de muestreo de los controladores PID para un T=0.1 y NT=0.3 s, al igual que la planta controlada basada en modelo bifrecuencia se dibujan en la gráfica. Usando el modelo perfecto de planta, la parte inicial de la respuesta de la planta controlada a bifrecuencia sigue lo logrado por el controlador rápido. Sin embargo, en el instante de tiempo t = NT = 0.3 s. una nueva medida es tomada y la respuesta mejora.

Obviamente, el controlador completo es mucho más complejo que cualquiera de estos dos controladores simples discretizados a monofrecuencia. También es mejor que el diseño controlador lento de cancelación para alcanzar la función de transferencia objetivo $M^{NT}(z_N)$.

4. SysQuake

Sysquake es un software creado especialmente para el diseño de sistemas aprovechando la visualización gráfica. Se diferencia de los programas tradicionales en que utiliza gráficos interactivos, permitiendo su manipulación directa por parte del usuario. En base a esto, el uso de Sysquake es altamente recomendable cuando se busca conseguir uno de los siguientes dos objetivos:

- **Entender conceptos básicos:** En Ingeniería muchas veces nos encontramos con que la teoría no es muy intuitiva al principio al relacionar variables de diferentes dominios: Energías y posiciones, tiempo y frecuencia, temperaturas y entropías,...Los mecanismos básicos que relacionan estas variables pueden ser ilustrados de forma muy efectiva con el uso de Sysquake.
- **Diseño de sistemas:** En la gran mayoría de aplicaciones ingenieriles hay que buscar un compromiso de funcionamiento, debido a que los objetivos que hay que alcanzar (por ejemplo, velocidad, precisión y coste) tienen requerimientos contradictorios. Si nos centramos en los sistemas de control, es bien conocido que su diseño no es un proceso secuencial, sino más bien una integración entre cálculo numérico, análisis de resultados, y ajuste de los parámetros. Las características de Sysquake se adaptan muy bien a este tipo de procesos, ya que permiten no sólo observar la actualización de los gráficos en tiempo real mientras se varían los parámetros, sino también manipular los gráficos directamente y ver cómo los parámetros cambian en consecuencia.

4.1 Características generales

En este proyecto se ha trabajado con Sysquake 5 Pro; aunque puede obtenerse una versión gratuita de este programa que puede obtenerse de la página de su distribuidor Calerga (*www.calerga.com)*. Ésta versión gratuita posee la mayoría de las características de la versión comercial y se puede utilizar sin ningún tipo de restricción.

Programación en Sysquake

El lenguaje de programación es muy similar a otros lenguajes de alto nivel, sobre todo al usado en MATLAB. Ahora bien, a diferencia de éste, incorpora comandos y funciones particulares para lograr la interactividad de los gráficos.

El intérprete de información numérica usado por Sysquake se llama "motor de matriz de poco peso" (LME, *Lightweight Math Engine*). Éste incorpora más de 700 funciones, operadores y comandos nativos. En la versión LE, las funciones utilizadas para leer datos externos al programa creados por el usuario no están disponibles. Como resultado de esto, no se pueden hacer llamadas a funciones externas, por lo que si se crean nuevas funciones éstas deben incorporarse directamente al código.

4.2 La Interfaz de Sysquake

Sysquake puede ejecutarse de dos maneras distintas. Por una parte, los archivos con extensión .sq se pueden abrir directamente, con lo que el programa contenido en ese fichero se ejecutará automáticamente. Como segunda opción, se puede abrir el archivo Sysquake.exe, abriéndose una pantalla en blanco y dándonos la opción de cargar algún archivo con el código del programa o utilizar la línea de comandos. Para esto último debe marcarse la casilla *command panel* en la pestaña *view* de la barra de herramientas, tal y como se observa en la Figura 1:

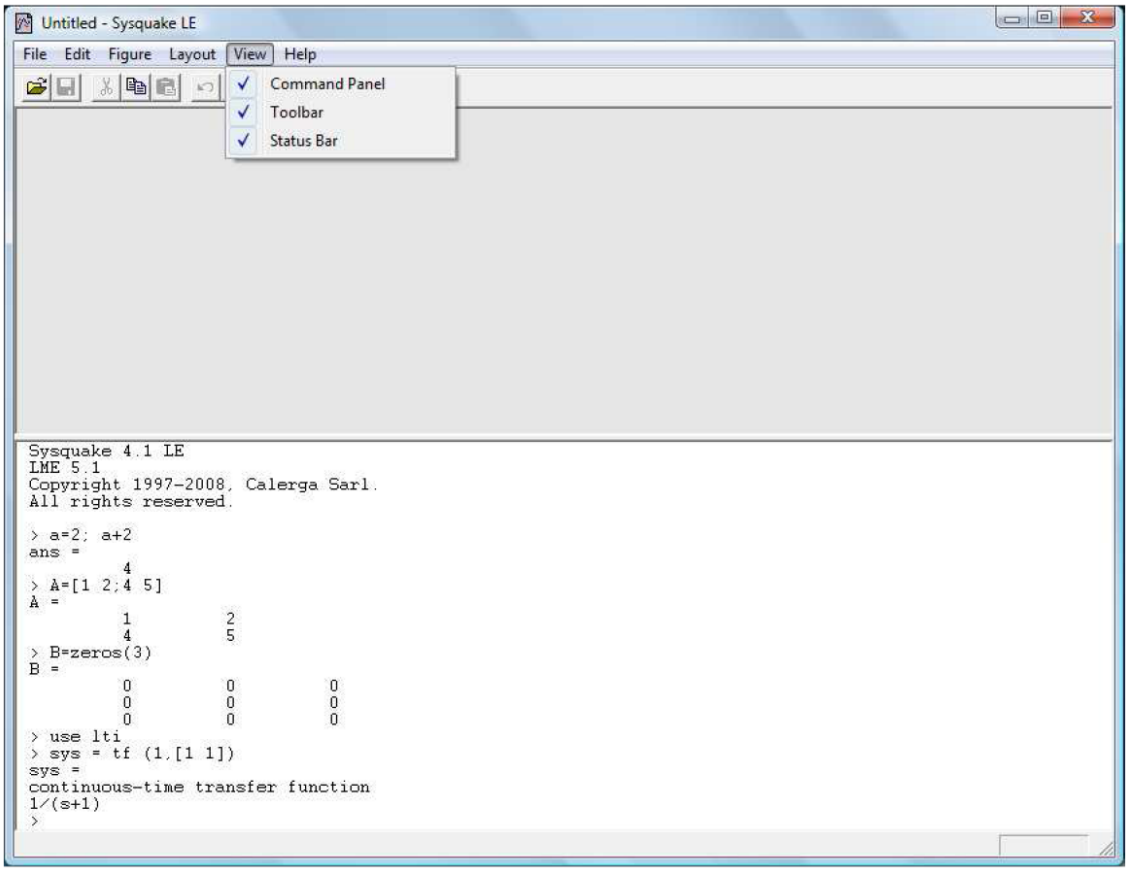


Figura 1: Habilitación de la línea de comandos.

En esta misma figura se observa que se pueden introducir variables, manipular matrices, definir funciones, etc. de la misma forma que se hace en MATLAB.

4.3 Menús

En este apartado se va a hacer una descripción general de los comandos presentes en los menús de Sysquake.

4.3.1 Menú file

Los comandos que contiene este menú se presentan en la Figura 2:

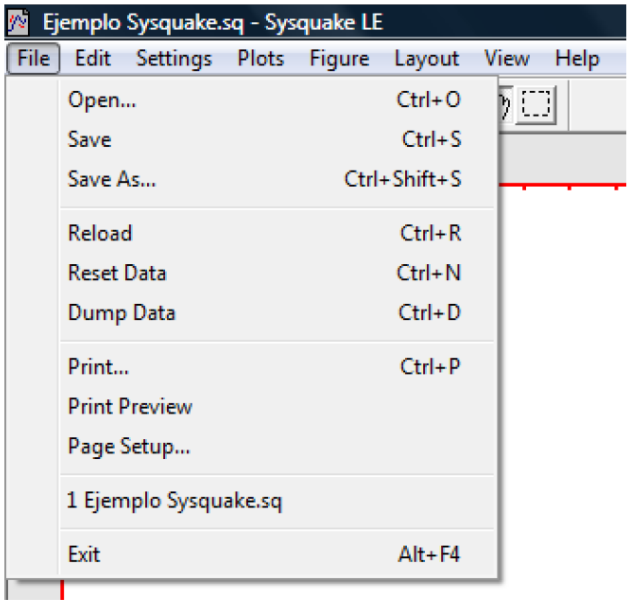


Figura 2: Elementos del menú File

Open sirve para abrir ficheros con extensión .sq o con la extensión del editor de
texto utilizado.

Save y *Save as* sirven para guardar la sesión de trabajo en un fichero con extensión .sqd.

Open y *Save* son comandos que también están disponibles en la barra de herramientas,
de la que hablaremos en un apartado posterior.

Reload sirve para volver a cargar el fichero sq.

Reset Data sirve para restablecer los valores originales del archivo que se encuentra
en uso.

Dump Data sirve para mandar datos relacionados con las variables de trabajo a la línea de comandos.

Print sirve para imprimir las gráficas activas.

Print Preview sirve para obtener una vista previa de lo que se va a imprimir.

Page Setup sirve para configurar la página de impresión.

Exit sirve para salir del programa.

4.3.2 Menú edit

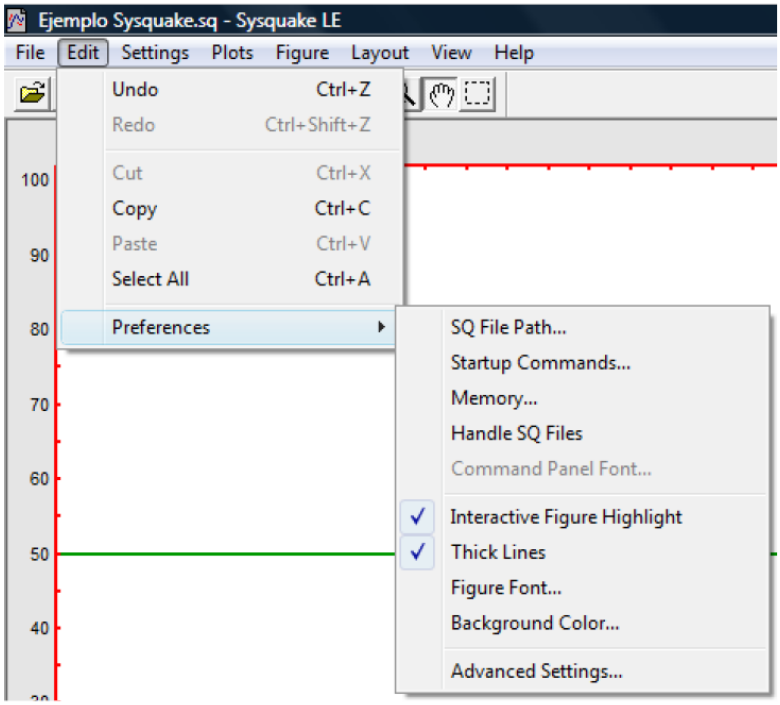


Figura 3: Elementos del menú Edit

Como se ha mostrado en la Figura 3, este menú contiene los comandos necesarios para el intercambio de datos dentro de Sysquake y con otros programas.

Undo permite volver a la situación existente antes de la última acción del usuario del programa.

Redo deshace el último *Undo*.

Cut, Copy y Paste permiten cortar, copiar y pegar (respectivamente) la gráfica seleccionada. Estos cinco primeros comandos también estarán disponibles en la barra de herramientas, de la que hablaremos en un apartado posterior.

Select All selecciona todas las gráficas activas.

Preferences permite cierta personalización de la forma en que Sysquake interactúa con el sistema operativo a través de los siguientes comandos:

- *SQ File Path* nos deja definir la ruta de acceso a los archivos y bibliotecas que Sysquake llama en los programas.
- *Startup Commands* se utiliza si se desea ejecutar un comando cada vez que se inicia Sysquake. Esto es útil si a menudo se usan funciones de una biblioteca específica que no está cargada por defecto. Para llamar a una librería, debe escribirse la palabra *use* seguida del nombre de la biblioteca. Por ejemplo, use stdlib llama a las funciones básicas incorporadas en Sysquake.

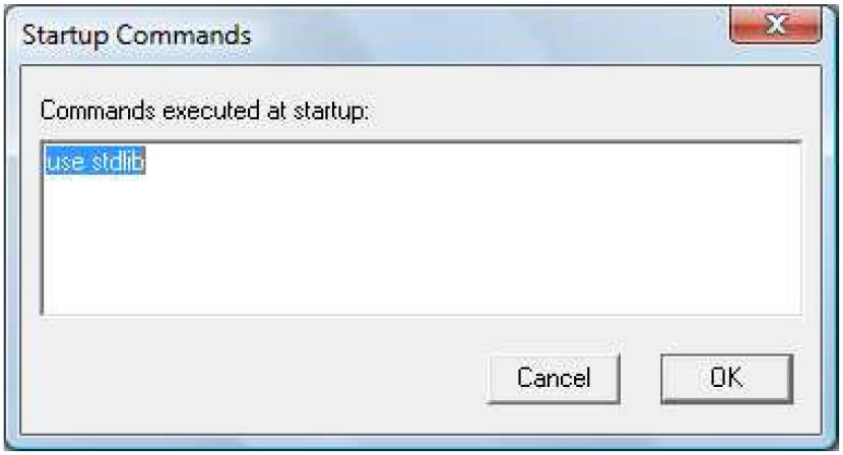


Figura 4: Ejemplo de llamada a una librería con Startup Commands

- *Memory* permite al usuario establecer un mínimo o máximo de acceso a memoria, aunque Sysquake ajusta automáticamente su uso conforme el programa lo necesita.
- *Handle SQ Files* asocia los ficheros SQ a Sysquake.
- *Command Panel Font* permite elegir el tipo de letra para la barra de comandos.
- *Interactive Figure Highlight* permite hacer una distinción entre las figuras que contienen elementos manipulables directamente con el cursor y las que no. Si está activada esta opción, el recuadro alrededor de las gráficas interactivas se verá en rojo; y el recuadro alrededor de las que no lo son, en negro.
- *Thick Lines* hace que en todas las gráficas aparezcan las líneas más gruesas.
- *Figure Font* muestra una caja de diálogos que ofrece la posibilidad de cambiar el tipo de letra en los gráficos.
- *Background Color* muestra una caja de diálogos para cambiar el color de fondo del programa.
- *Advanced Settings* permite activar/desactivar una serie de casillas para pedir un mensaje de confirmación antes de cerrar el programa (*Ask before closing*), mostrar en la línea de comandos posibles mensajes de error al iniciar un programa (*SQ file posible error warnings*), convertir los archivos de SQ a un código intermedio para acelerar la ejecución del programa (*LME code optimization*), etc.

4.3.3 Menú settings

Este menú sólo estará disponible en aquellos programas que se hayan escrito usando una estructura de menús. Si se ha hecho así, esta pestaña permitirá cambiar los valores de las variables usadas en el programa.

4.3.4 Menú plots

Este menú contiene la lista de figuras que pueden ser mostradas. Un programa medianamente complejo en Sysquake tiene una mayor cantidad de gráficos de los que se pueden mostrar simultáneamente por motivos de espacio en la ventana, de ahí la necesidad de este menú.

4.3.5 Menú figure

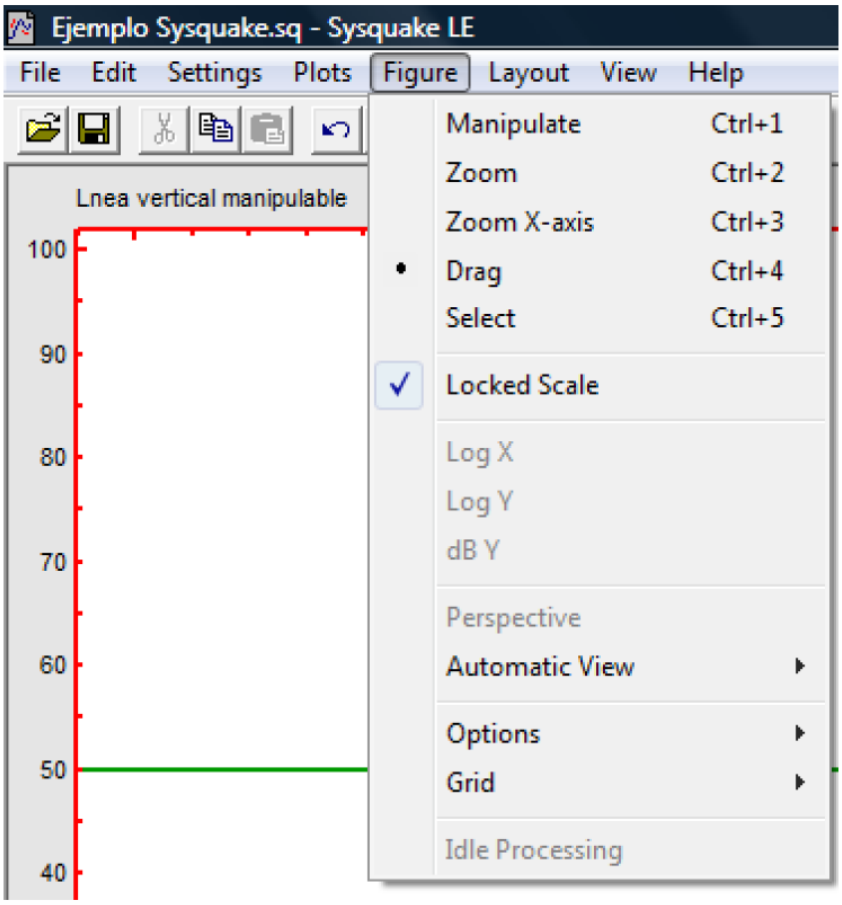


Figura 5: Elementos del menú Figure

Tal y como se muestra en la Figura 5, este menú contiene los comandos necesarios para el intercambio de datos dentro de Sysquake y con otros programas.

Manipulate: Cuando está habilitado permite modificar los elementos interactivos de las distintas gráficas. Si el cursor está sobre un elemento modificable, tomará la forma de una mano; en caso contrario, conservará la forma estándar de flecha.

Zoom: Cuando esta opción se activa permite observar con mayor detalle determinadas áreas de las gráficas en pantalla. Para ello, basta con arrastrar el cursor con el botón izquierdo pulsado para formar un cuadrilátero que englobe el área que será aumentada.

Zoom X Axis: Zoom que sólo tiene efecto sobre el eje x.

Drag: Si esta casilla está activa, al mover el ratón con su botón izquierdo presionado los límites del gráfico seleccionado cambiarán de tal forma que el punto bajo el cursor se mueve con ellos.

Select se utiliza para escoger una de las figuras, y, o bien sustituirla por otra usando comandos del menú *plots*, o bien arrastrarla encima de otra gráfica activa, con lo que ambas gráficas intercambiarán sus posiciones.

Estas cinco primeras opciones constituyen los modos de manipulación de las figuras, y también estarán accesibles en la barra de herramientas, de la que hablaremos en un apartado posterior

Locked Scale: A no ser que se defina en el código del programa, Sysquake define una escala implícita para cada gráfico. Cuando se amplía (*Zoom*, *zoom X Axis*) o se arrastra (*Drag*) la figura la escala cambia, y con este comando se puede volver a la escala original.

Log X: La escala horizontal se vuelve logarítmica, y los valores negativos se descartan.

Log Y: La escala vertical se convierte a logarítmica, y los valores negativos se descartan.

dB Y: La escala vertical se convierte a logarítmica en decibelios.

Options brinda la posibilidad de habilitar o inhabilitar las siguientes cuatro opciones:

- *Frame* activa/desactiva el recuadro que rodea a la gráfica con los valores de la escala.
- *Label* activa/desactiva los títulos de las gráficas.
- *Margin* funciona como *Frame*, pero sin quitar las barras divisorias que nos dan una noción de la escala y expandiendo el contenido de la gráfica para que llene toda la ventana
- A cada gráfico se le puede añadir una leyenda que será mostrada en cualquiera de las cuatro esquinas de la gráfica (se puede mover con el ratón de una esquina a otra). En caso de que esta leyenda se convierta en un obstáculo para visualizar el gráfico (especialmente cuando se están mostrando muchos gráficos en pantalla a la vez) se puede deshabilitar con el comando *Legend*.

Grid: brinda la posibilidad de habilitar la rejilla o cuadrícula (líneas en el gráfico con separación constante) en el eje *x* y/o el eje *y*. La disponibilidad y el tipo de cuadrícula dependen de la figura.

Idle Processing: Ciertos programas ejecutan cálculos aún cuando no se está interactuando con ellos (ejemplo: simulaciones animadas). Si se deshabilita este comando se suspenderá la ejecución de estos cálculos.

4.3.6 Menú layout

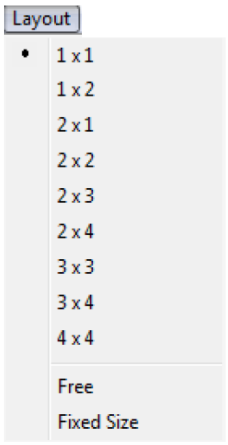


Figura 6: Elementos del menú Layout

Tal y como se muestra en la Figura 6, este menú se utiliza para seleccionar el número de gráficos mostrados simultáneamente. Las opciones disponibles incluyen 1x1, 1x2,
2x1, 2x2, 2x3, 2x4, 3x2, 3x4, 3x4, 4x4; donde el primer número corresponde al número de filas y el segundo al de las columnas. Por tanto, utilizando este menú se pueden observar de 1 a 16 figuras simultáneamente en pantalla.
Vale la pena resaltar que para cada programa sólo existe una ventana activa, por lo que si (por ejemplo) se desea observar el comportamiento de tres gráficos simultáneamente, estos deben visualizarse en una misma ventana. En la mayoría de los casos no es recomendable utilizar Sysquake en modos superiores a 2x3, ya que la disminución en el tamaño de los gráficos aumenta el grado de dificultad para la manipulación; y, evidentemente, su detalle se reduce.
Hay además dos opciones adicionales:

Free: Cuando está habilitado se puede cambiar libremente el tamaño y posición de las figuras con el ratón (siempre y cuando el modo de manipulación esté en *select*).
Cuando se cambia el tamaño de la ventana del programa, todas las gráficas quedan escaladas proporcionalmente.

Fixed Size funciona igual que *free*, pero cuando se cambia el tamaño de la ventana del programa las gráficas no cambian su tamaño.

4.3.7 Menú view

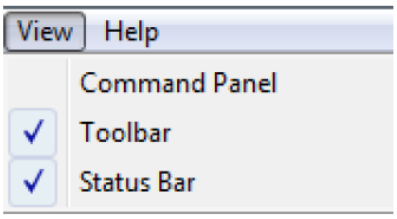


Figura 7: Elementos del menú View

La figura 7 nos muestra que este menú contiene los siguientes tres comandos:

Command Panel permite mostrar u ocultar la línea de comandos, que estará disponible en la parte inferior de la ventana principal. El tamaño de la línea de comandos puede cambiarse arrastrando la separación entre ésta y las gráficas con el cursor del ratón. Si se está ejecutando un programa en Sysquake esta línea, en la mayoría de los casos, no se usará.

Toolbar oculta o muestra la barra de herramientas. Esta barra contiene los comandos *Open, Save, Cut, Copy, Paste, Undo, Redo, Manipulate, Zoom, Zoom X Axis, Drag* y *Select* , de los que ya hemos hablado en apartados anteriores.

Status Bar oculta o muestra una región en la parte inferior de la ventana del programa donde se pueden desplegar distintos mensajes.

En la figura 8 se muestra la posición en pantalla de estos elementos:

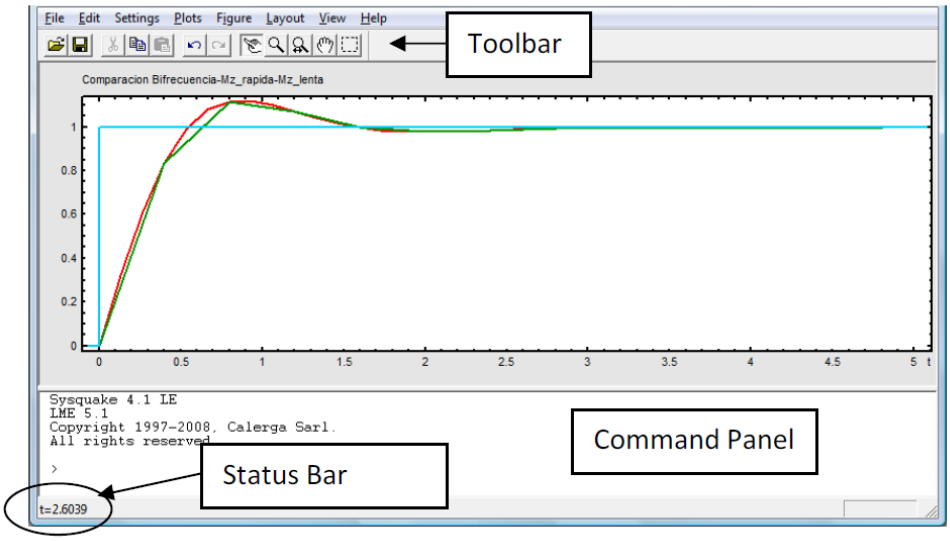


Figura 8: Posición de la línea de comandos, de la barra de estatus y de la barra de herramientas

4.3.8 Menú help

Los comandos que contiene este menú se presentan en la figura 9:

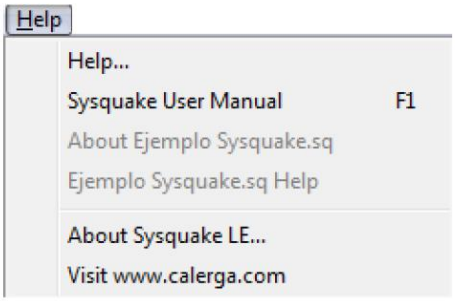


Figura 9: Elementos del menú Help

Help despliega una ventana que contiene un par de comentarios con las principales características de Sysquake. Seleccionando *example* se abre el archivo de ejemplo *triangle.sq*.

Sysquake User manual abre el manual del usuario de Sysquake en formato htm.

About Nombre_del_archivo abre, si se ha creado, una ventana con un comentario acerca del programa cargado.

Nombre_del_archivo File Help despliega, si se ha creado, una ventana con la ayuda para el programa cargado.

About Sysquake muestra una ventana con la versión del programa instalada, el copyright y la página web del distribuidor.

Go to www.calerga.com Abre la página web del distribuidor de Sysquake.

5. Manual de Usuario

A continuación se procede a explicar la interfaz del programa a la que tiene acceso el usuario, obviando las opciones generales de SysQuake, que han sido debidamente explicadas en el apartado anterior.

5.1 Interfaz general

Todos los controladores comparten la mayoría de aspectos de su interfaz, aunque cada controlador posee sus peculiaridades. En la Figura 1 podemos ver la interfaz del controlador PID.

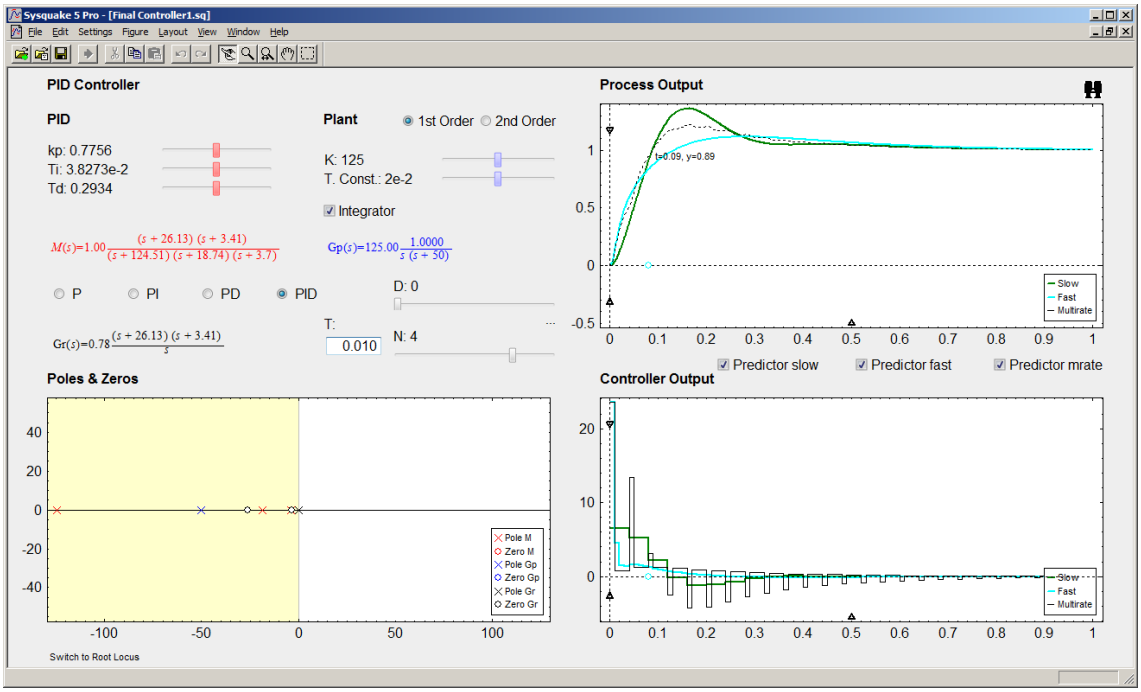


Figura 1: Interfaz Controlador PID

Podemos dividir la pantalla en dos partes: la de entrada de datos (mitad izquierda) y la de salida de datos (mitad derecha).

La parte izquierda del programa es para la visualización y entrada de datos requeridos del programa, entre estos valores se encuentran: Controlador o Modelo del sistema, Planta del sistema, visualización de las funciones de transferencia (Planta, Controlador y/o Modelo) y Mapa de polos y ceros, entre otros.

En la parte derecha del programa podemos observar las salidas de los sistemas y de los controladores en forma de gráficas. En las gráficas se pueden observar 3 gráficos distintos, uno de color verde (representa el sistema lento), otro de color celeste

(representa el sistema rápido) y otra de color negro punteado (representa el sistema multifrecuencia). Pasar el cursor por encima de cualquier gráfico permite ver el valor del tiempo e intensidad de la señal en ese punto.

Como hemos visto en el apartado anterior, SysQuake ofrece una interfaz gráfica al usuario que le permite manipular dinámicamente el sistema. Este dinamismo nos permite modificar en tiempo real los valores de las variables del programa, así pues, las herramientas para realizar estas modificaciones en este proyecto son:

- A través de la opción de menú *Settings*. Este menú varía dependiendo del controlador, pero normalmente se permitirá cambiar todos los valores básicos del sistema, el tipo de controlador y el modo en que se modifican los deslizadores.
- A través de cajas de texto, cajas de selección, botones de selección o deslizadores. Los deslizadores pueden aumentar/disminuir los valores de un modo lineal o porcentual.
- Arrastrando y soltando los elementos gráficos (los símbolos en el mapa de polos y ceros, por ejemplo).

Para facilitar el uso del programa, se ha realizado esta interfaz buscando la máxima coherencia en su elaboración. Se ha tenido en cuenta la distribución (todos los sistemas tienen una interfaz similar), el color de los elementos (por ejemplo, se representa con color azul los deslizadores, la función de transferencia y los polos y ceros de la planta), la ubicación bien diferenciada entre entrada y salida de datos y se ha utilizado una simbología que debería resultar familiar al usuario medio.

5.2 Interfaz del controlador PID

Menu Settings

A continuación se detallan las opciones de menú *Settings* del controlador PID:

- **System Parameters**:
  - **Set mapping parameters**: Permite modificar los valores T y N
  - **Set transfer function Gp parameters**: Permite introducir el numerador y denominador de la función de transferencia Gp.
  - **Set transfer function Gr parameters**: Permite introducir el numerador y denominador de la función de transferencia Gr.
- **Controller**
  - **PID Controller**: Cambia al controlador PID.
  - **Cancellation Controller**: Cambia al controlador de cancelación.
  - **RST Controller**: Cambia al controlador RST.
  - **Model Based Controller**: Cambia al controlador PID basado en modelo.
- **Sliders Config**
  - **Slider by percentage**: Permite modificar los sliders en base a un porcentaje introducido por el usuario.
  - **Slider by value**: Permite modificar los sliders en base a un valor introducido por el usuario.

Interfaz

En la Figura 2 se puede observar la interfaz del controlador PID con sus elementos numerados. Posteriormente se explica con detalle cada apartado.

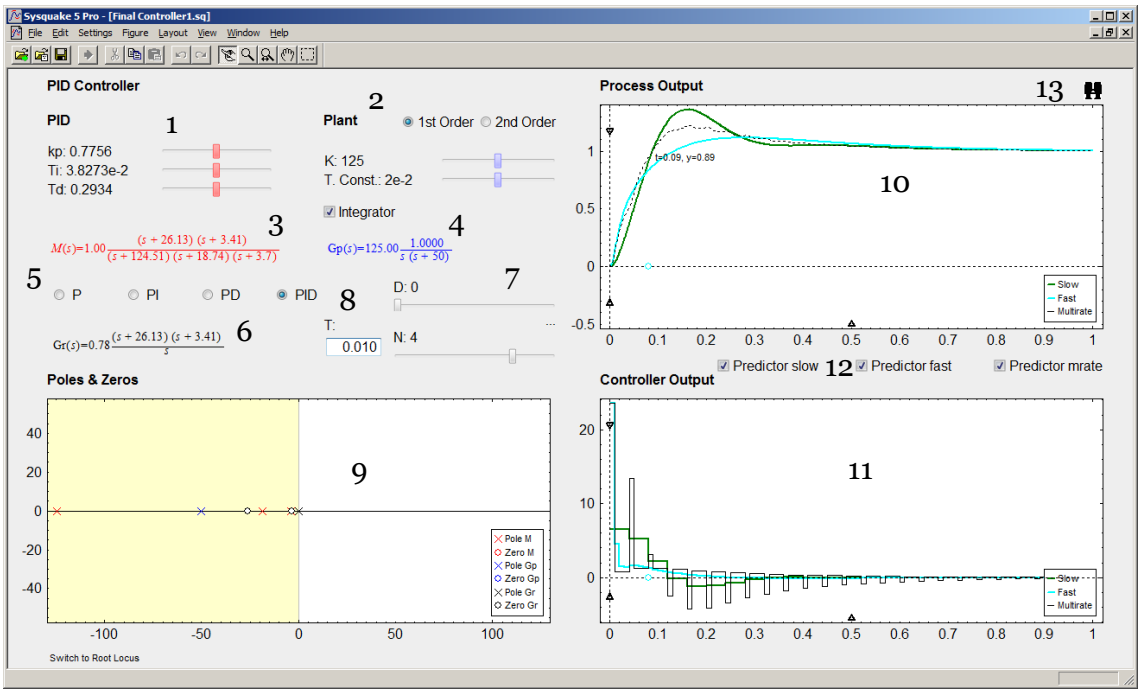


Figura 2: Interfaz Controlador PID detallada

1: Modificación de los parámetros del controlador PID a través de deslizadores.

2: Modificación de los parámetros de la planta del sistema a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2) o de añadir un integrador con la caja de selección.

3: Función de transferencia del modelo del sistema.

4: Función de transferencia de la planta del sistema.

5: Cambiar el tipo de PID a usar como controlador (P, PI, PD o PID).

6: Función de transferencia del controlador.

7: Añadir o eliminar retardo (*delay*).

8: Modificación del periodo de muestreo (T) y del multiplicador del periodo para el caso lento (N).

9: Mapa interactivo de polos y ceros. Dispone de un botón en la parte inferior para cambiar a la visualización del lugar de las raíces (ver Figura 3).

10: Gráfica con la salida del sistema.

11: Gráfica con la salida del controlador

12: Cajas de selección para activar/desactivar el uso del predictor de Smith[2] cuando añadimos retardo al sistema.

13: Botón para auto escalar las dos gráficas (salida del sistema y salida del controlador).

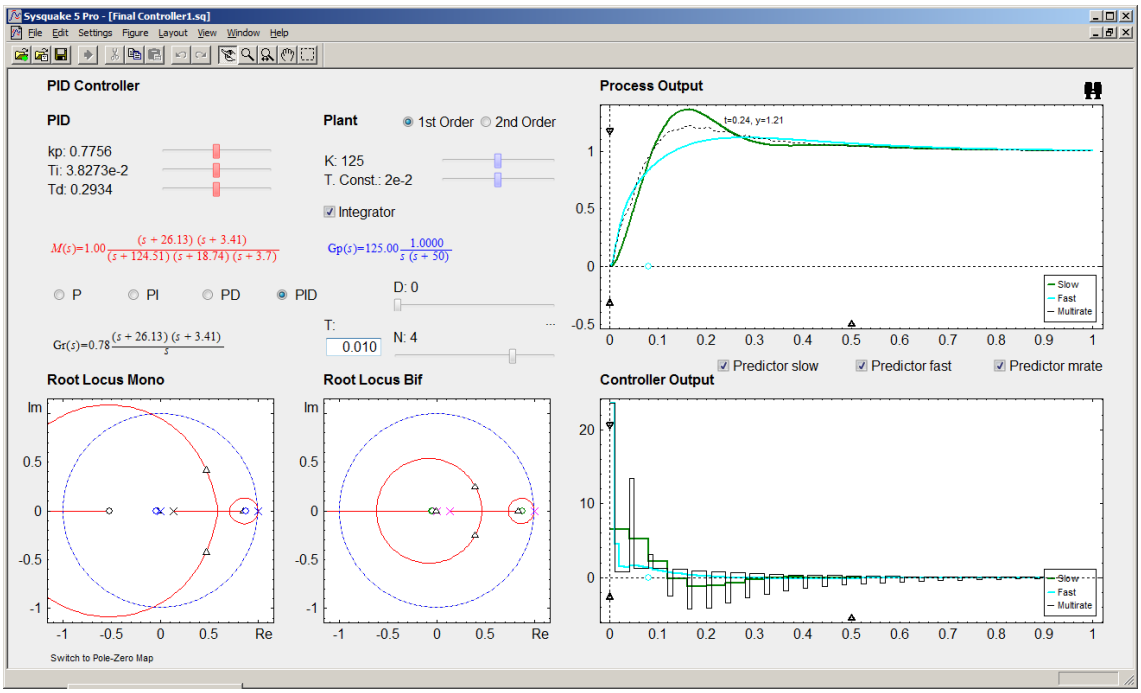


Figura 3: Controlador PID con lugar de las raíces.

2: http://en.wikipedia.org/wiki/Smith_predictor

5.3 Interfaz del controlador de Cancelación

Menu Settings

A continuación se detallan las opciones de menú *Settings* del controlador de cancelación:

- **System Parameters**:
  - **Set mapping parameters**: Permite modificar los valores T y N
  - **Set transfer function Gp parameters**: Permite introducir el numerador y denominador de la función de transferencia Gp.
  - **Set transfer function Gr parameters**: Permite introducir el numerador y denominador de la función de transferencia Gr.
- **Controller**
  - **PID Controller**: Cambia al controlador PID.
  - **Cancellation Controller**: Cambia al controlador de cancelación.
  - **RST Controller**: Cambia al controlador RST.
  - **Model Based Controller**: Cambia al controlador PID basado en modelo.
- **Sliders Config**
  - **Slider by percentage**: Permite modificar los sliders en base a un porcentaje introducido por el usuario.
  - **Slider by value**: Permite modificar los sliders en base a un valor introducido por el usuario.

Interfaz

En la Figura 4 se puede observar la interfaz del controlador de cancelación con sus elementos numerados. Posteriormente se explica con detalle cada apartado.

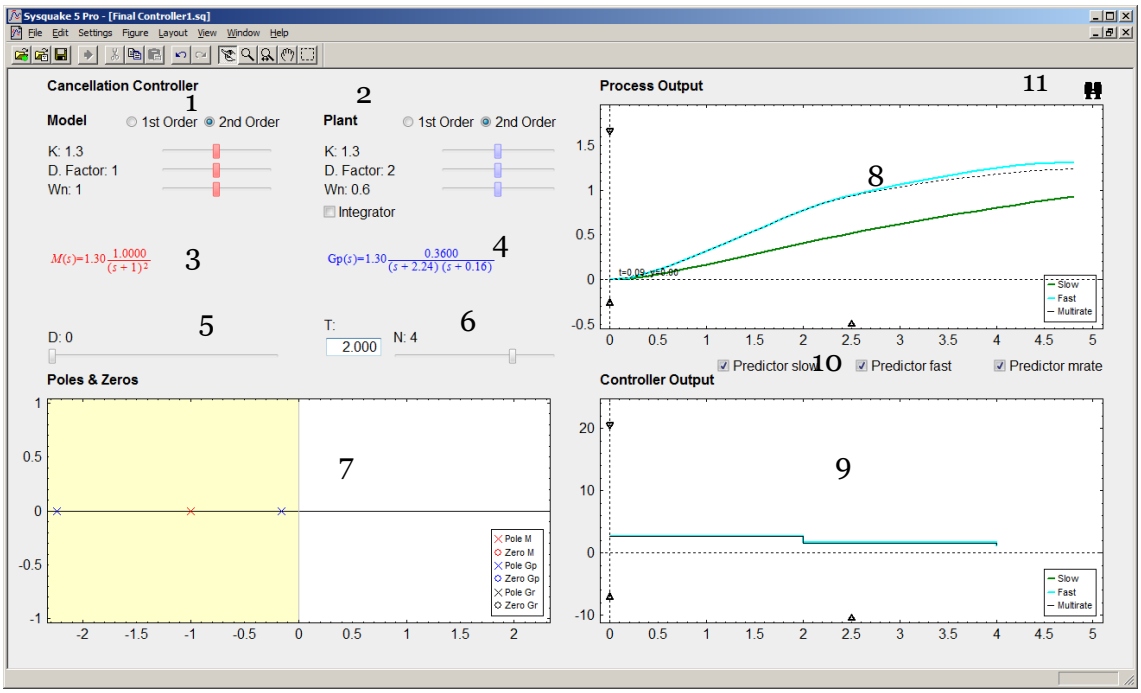


Figura 4: Interfaz Controlador de Cancelación detallada

1: Modificación de los parámetros del controlador a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2) o de añadir un integrador con la caja de selección.

2: Modificación de los parámetros de la planta del sistema a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2) o de añadir un integrador con la caja de selección.

3: Función de transferencia del modelo del sistema.

4: Función de transferencia de la planta del sistema.

5: Añadir o eliminar retardo (*delay*).

6: Modificación del periodo de muestreo (T) y del multiplicador del periodo para el caso lento (N).

7: Mapa interactivo de polos y ceros.

8: Gráfica con la salida del sistema.

9: Gráfica con la salida del controlador

10: Cajas de selección para activar/desactivar el uso del predictor de Smith cuando añadimos retardo al sistema.

11: Botón para auto escalar las dos gráficas (salida del sistema y salida del controlador).

5.4 Interfaz del controlador RST

Menu Settings

A continuación se detallan las opciones de menú *Settings* del controlador de RST:

- **System Parameters**:
  - **Set mapping parameters**: Permite modificar los valores T y N
  - **Set transfer function Gp parameters**: Permite introducir el numerador y denominador de la función de transferencia Gp.
  - **Set transfer function Gr parameters**: Permite introducir el numerador y denominador de la función de transferencia Gr.
- **Controller**
  - **PID Controller**: Cambia al controlador PID.
  - **Cancellation Controller**: Cambia al controlador de cancelación.
  - **RST Controller**: Cambia al controlador RST.
  - **Model Based Controller**: Cambia al controlador PID basado en modelo.
- **Sliders Config**
  - **Slider by percentage**: Permite modificar los sliders en base a un porcentaje introducido por el usuario.
  - **Slider by value**: Permite modificar los sliders en base a un valor introducido por el usuario.

Interfaz

En la Figura 5 se puede observar la interfaz del controlador de RST con sus elementos numerados. Posteriormente se explica con detalle cada apartado.

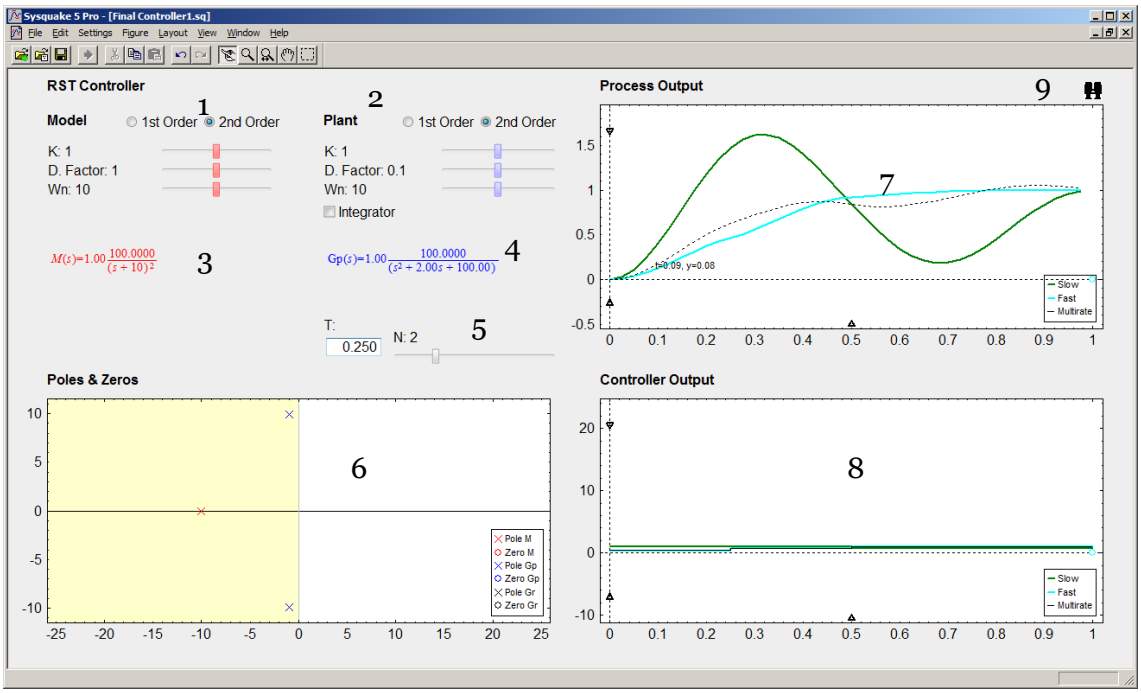


Figura 5: Interfaz Controlador RST detallada

1: Modificación de los parámetros del controlador a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2).

2: Modificación de los parámetros de la planta del sistema a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2) o de añadir un integrador con la caja de selección.

3: Función de transferencia del modelo del sistema.

4: Función de transferencia de la planta del sistema.

5: Modificación del periodo de muestreo (T) y del multiplicador del periodo para el caso lento (N).

6: Mapa interactivo de polos y ceros.

7: Gráfica con la salida del sistema.

8: Gráfica con la salida del controlador

9: Botón para auto escalar las dos gráficas (salida del sistema y salida del controlador).

5.5 Interfaz del controlador PID basado en modelo

Menu Settings

A continuación se detallan las opciones de menú *Settings* del controlador PID:

- **System Parameters**:
  - **Set mapping parameters**: Permite modificar los valores T y N
  - **Set transfer function Gp parameters**: Permite introducir el numerador y denominador de la función de transferencia Gp.
  - **Set transfer function Gr parameters**: Permite introducir el numerador y denominador de la función de transferencia Gr.
- **Controller**
  - **PID Controller**: Cambia al controlador PID.
  - **Cancellation Controller**: Cambia al controlador de cancelación.
  - **RST Controller**: Cambia al controlador RST.
  - **Model Based Controller**: Cambia al controlador PID basado en modelo.
- **Sliders Config**
  - **Slider by percentage**: Permite modificar los sliders en base a un porcentaje introducido por el usuario.
  - **Slider by value**: Permite modificar los sliders en base a un valor introducido por el usuario.

Interfaz

En la Figura 6 se puede observar la interfaz del controlador de PID basado en modelo con sus elementos numerados. Posteriormente se explica con detalle cada apartado.

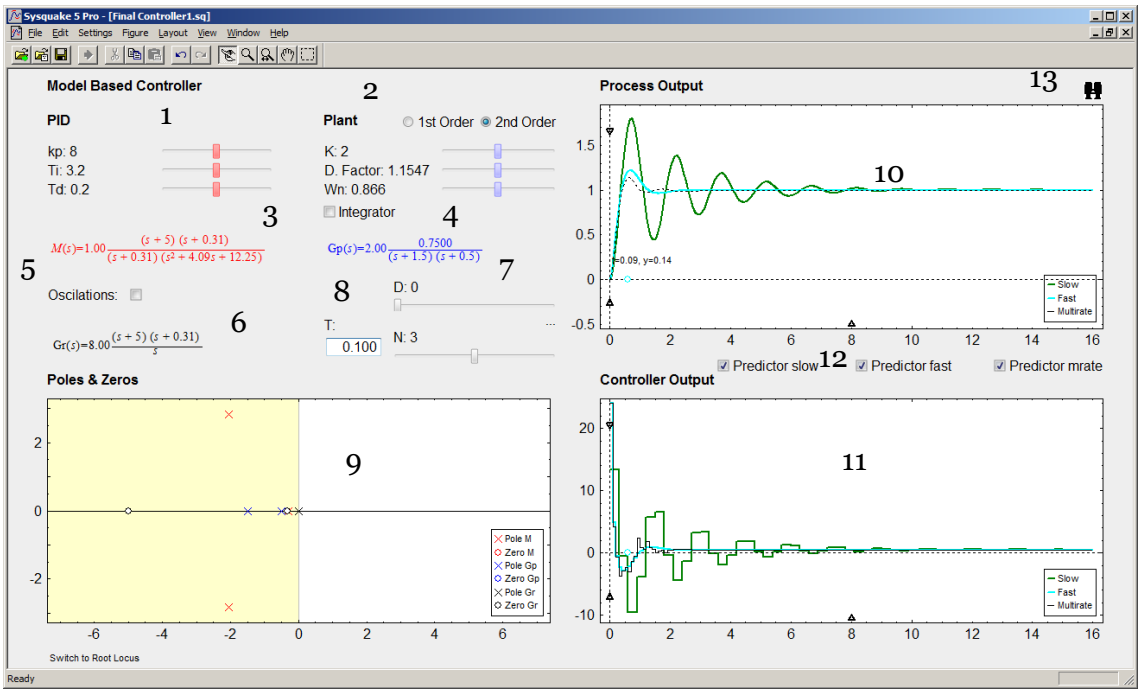


Figura 6: Interfaz Controlador PID basado en modelo detallada

1: Modificación de los parámetros del controlador PID a través de deslizadores.

2: Modificación de los parámetros de la planta del sistema a través de deslizadores. También existe la opción de seleccionar el orden de la planta (orden 1 u orden 2) o de añadir un integrador con la caja de selección.

3: Función de transferencia del modelo del sistema.

4: Función de transferencia de la planta del sistema.

5: Activar o desactivar las oscilaciones del modelo.

6: Función de transferencia del controlador.

7: Añadir o eliminar retardo (*delay*).

8: Modificación del periodo de muestreo (T) y del multiplicador del periodo para el caso lento (N).

9: Mapa interactivo de polos y ceros. Dispone de un botón en la parte inferior para cambiar a la visualización del lugar de las raíces (ver Figura 7).

10: Gráfica con la salida del sistema.

11: Gráfica con la salida del controlador

12: Cajas de selección para activar/desactivar el uso del predictor de Smith cuando añadimos retardo al sistema.

13: Botón para auto escalar las dos gráficas (salida del sistema y salida del controlador).

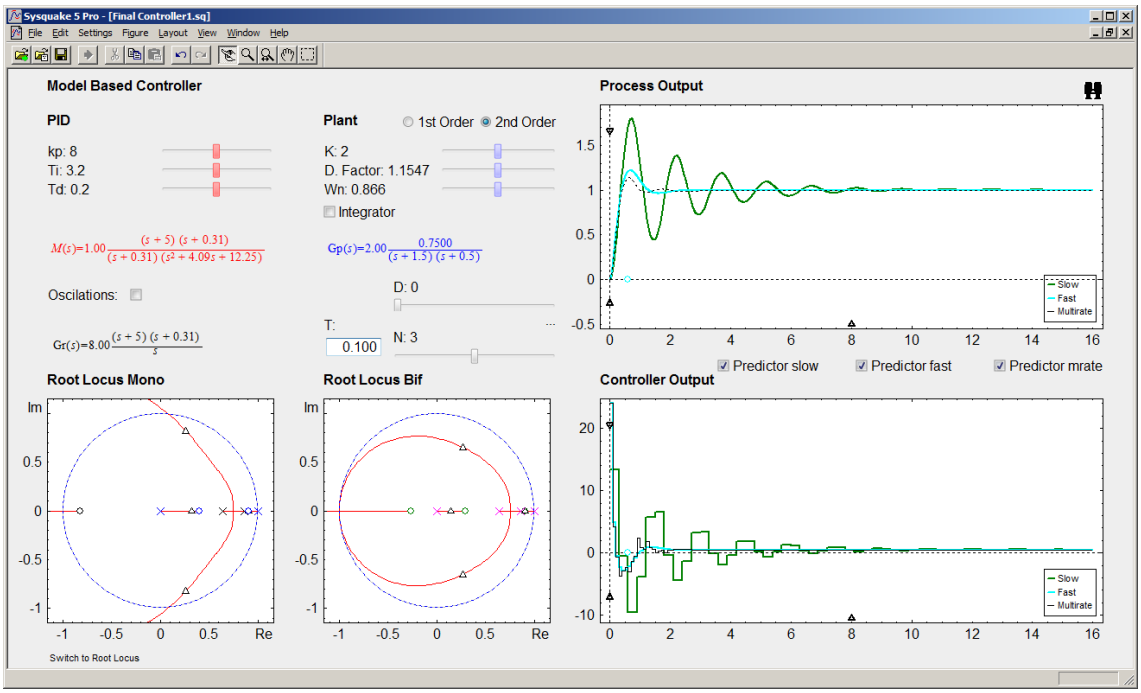


Figura 7: Interfaz Controlador PID basado en modelo con lugar de las raíces

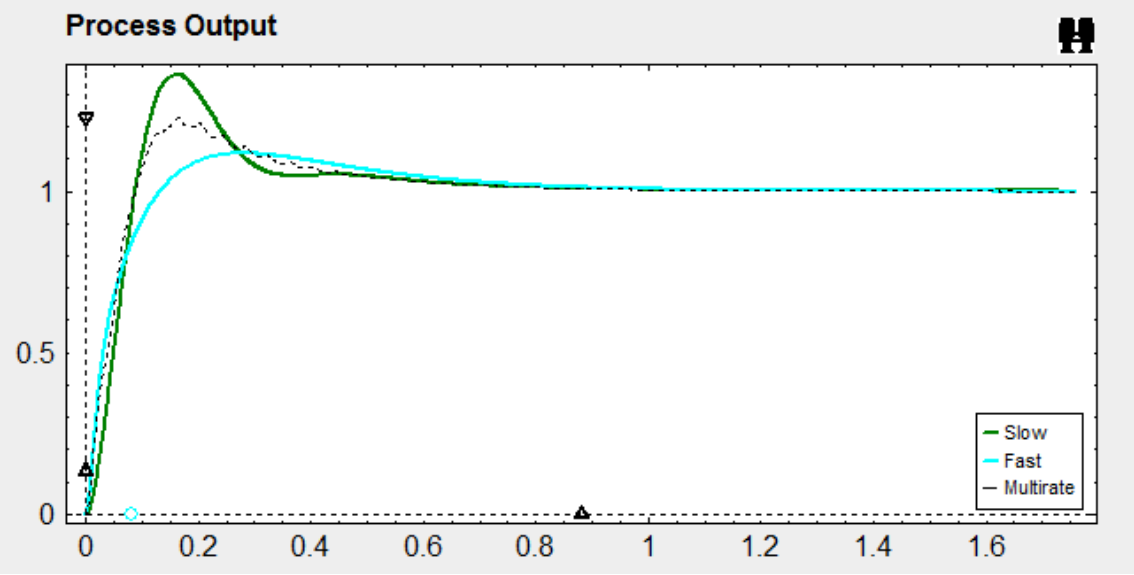
6. Resultados

Como fruto del proyecto se ha realizado una publicación, cuyos autores son Julián Salt Llobregat, Ángel Miguel Cuenca Lacruz, Sebastián Dormido (que ha sido quien ha presentado la publicación en el congreso) y Francisco Palau. Esta publicación se ha presentado en congreso internacional de investigación, concretamente en el “10th IFAC Symposium Advances in Control Education” llevado a cabo en la Universidad de Sheffield, en Reino Unido, cuyo título es "An Interactive Simulation Tool for the Study of Multirate Systems". En dicha publicación se resumen los resultados más relevantes del trabajo realizado.

En todos los controladores podemos observar el mismo resultado: el sistema multifrecuencia ofrece mejores resultados en cuestión de tiempo de establecimiento y menor sobreoscilaciones que los monofrecuencia lentos (aunque en ocasiones, se tienen más sobreoscilaciones que el rápido), aun midiendo en la misma frecuencia que el lento (esto es, NT).

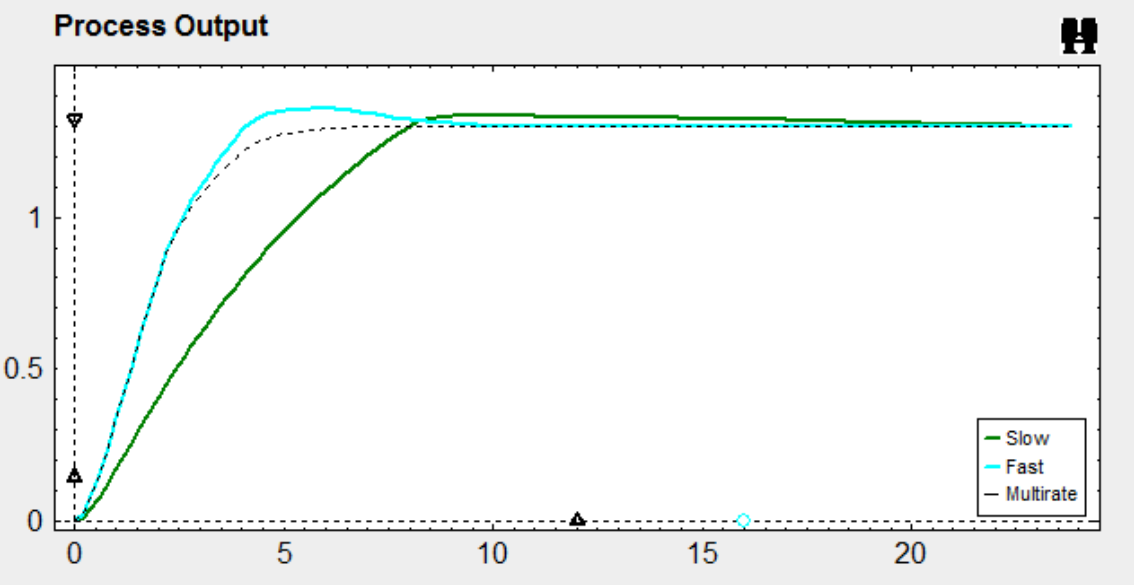
A continuación, analizaremos las salidas de los cuatro sistemas controlados.

6.1 Resultados del controlador PID



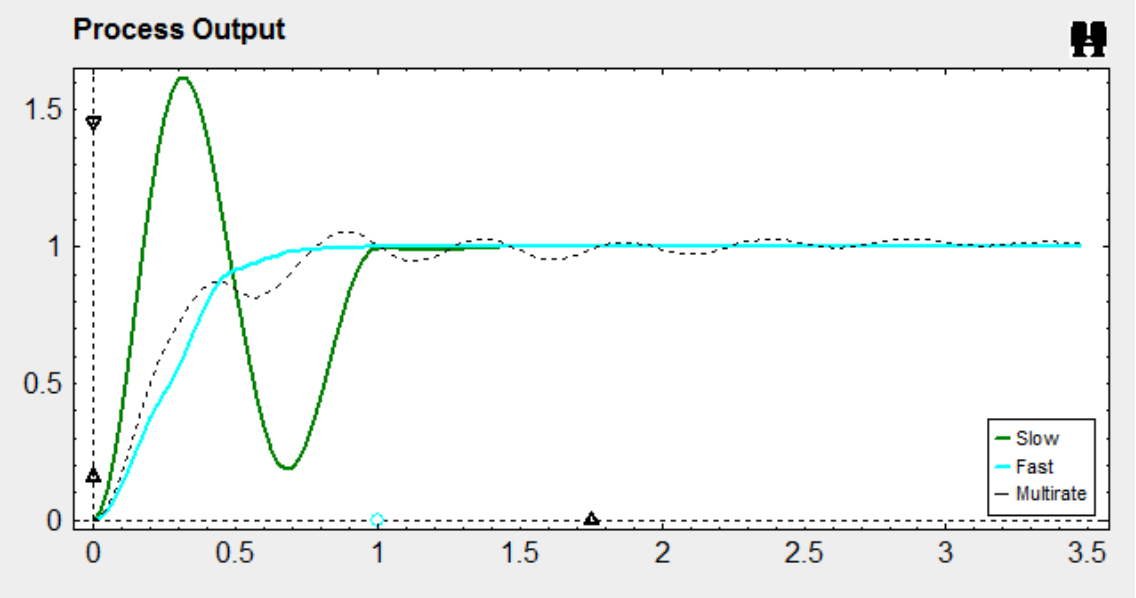
A causa de superposición de frecuencias, vemos oscilaciones en la salida multifrecuencia, pero podemos observar igualmente que, aunque el tiempo de establecimiento es similar a los sistemas monofrecuencia, la sobreoscilación máxima es mejor que en el caso monofrecuencia lento.

6.2 Controlador de Cancelación



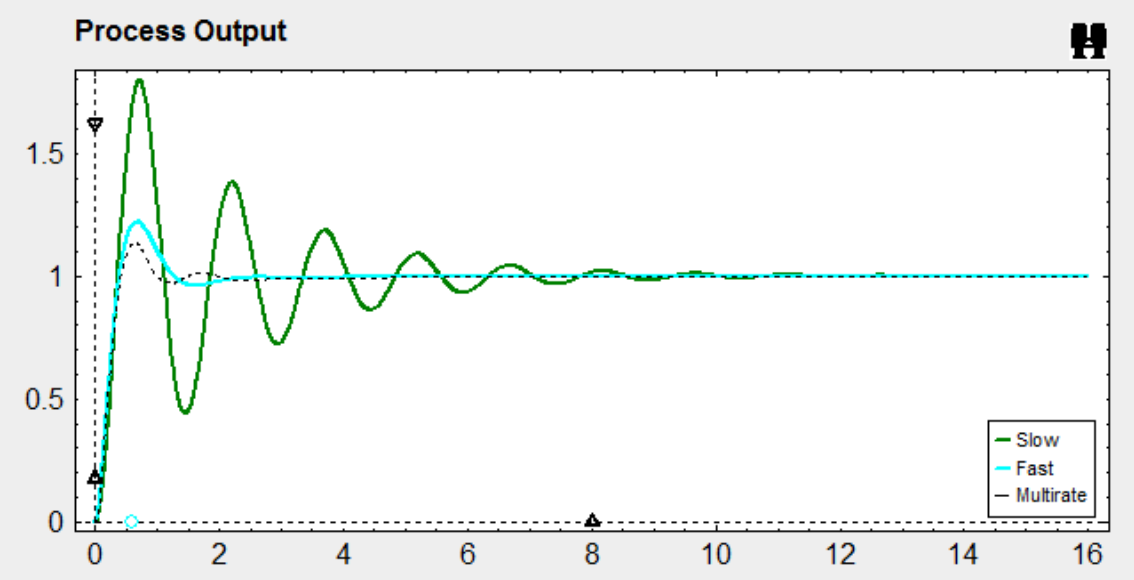
En este caso vemos que el tiempo de establecimiento del multifrecuencia es claramente mejor respecto a los monofrecuencia (incluso al monofrecuencia rápido).

6.3 Controlador RST



El controlador RST multifrecuencia muestra nuevamente la mejora relativa a la sobreoscilación máximas (menor sobreoscilación máxima que el controlador monofrecuencia lento). No obstante, vuelven a aparecer las oscilaciones intermuestreo afectando al tiempo de establecimiento.

6.4 Controlador PID basado en modelo



Podemos observar finalmente que en el controlador PID basado en modelo, el sistema multifrecuencia es claramente mejor que sus iguales monofrecuencia. Además podemos observar como el sistema lento ofrece unas oscilaciones muy acentuadas en el transitorio, que el sistema multifrecuencia (actuando a la misma frecuencia NT que el lento) elimina casi por completo, por lo que tenemos mejoras a nivel de sobreoscilación máxima y tiempo de establecimiento (incluso con respecto al monofrecuencia rápido).

7. Conclusiones y agradecimientos

Con este proyecto hemos sido capaces de generar una herramienta interactiva que nos ha permitido observar los beneficios de los sistemas de control multifrecuencia respecto a los clásicos monofrecuencia. Así hemos visto mejoras a nivel de reducción de tiempo de establecimientos y de sobreoscilación máxima, además hemos visto acciones de control menos enérgicas, y todo ello redunda en reducción de costes energéticos y computacionales.

Quisiera agradecer el trabajo y apoyo aportado por las siguientes personas:

Julián Salt Llobregat, catedrático del departamento ISA de la UPV.

Ángel Miguel Cuenca Lacruz, contratado doctor del departamento ISA de la UPV.

Sebastián Dormido, catedrático de la UNED.

8. Bibliografía

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

[2] M. Niemiec, C. Kravaris, *Nonlinear multirate control of a polymerization reactor: an experimental study*, Proceedings of the 2000 American Control Conference, vol. 4, pp. 2270-2274, 2000.

[3] M. Embiruçu, C. Fontes, *Multirate multivariable generalized predictive control and its application to a slurry reactor for ethylene polymerization*, Chemical Engineering Science, Elsevier Ltd., vol. 61, issue 17, pp. 5754-5767, 2006.

[4] J. Ding, F. Marcassa, S. Ch. Wu, M. Tomizuka, *Multirate control for computing saving*, IEEE Transactions on control systems technology, vol. 14, no 1, 2006.

[5] J. Sklansky y J.R. Ragazzini, *Analysis of errors in sampled-data feedback systems*, AIEE Trans., pt. II, vol. 74, pp. 65-71, 1955.

[6] T.C. Coffey y I.J. Williams, Stability analysis of multiloop, multirate sampled systems, AIAA J.Guild. Control Dyna., nº 4, pp 2178-2190,1966.

[7] Julian J. Salt, Antonio Sala, Jesus Sandoval, Perturbation and Exact Intersample Ripple detection in Discrete Dual-Rate Systems.

[8] H. In y C. Zhang, A multirate digital controller for model matching, Automatica, vol. 30, nº 6, pp 1053-1050, 1994.

[9] P. Albertos y J. Salt, Dual rate adaptive control, Automatica, vol. 32, nº 7, pp 1027-1030, 1996.

[10] Julian J. Salt y Pedro Albertos, Model-Based Multirate Controllers Design, IEEE Transactions on control systems technology, vol. 13, nº 6, 2005.

[11] R. Isermann, Digital Control Systems. New York: Springer-Verlag, 1981.

[12] J. Salt, V. Mascarós, A. Cuenca y V. Casanova, A PID dual rate controller implementation over a networked control system, IEEE International Conference on Control Applicationes, 2006.

[13] K.J. Astrom, B. Wittenmark, Computer-Controlled Systems: Theory and Design, Prentice Hall, 1984.

[14] E.K. Larsson, M. Mossberg, T. Soderstrom, Identification of continuous-time ARX models from irregularly sampled data, IEEE Transactions on Automatic Control 52 (3) (2007) 417–427.

[15] J. Salt, A. Sala, P. Albertos, A transfer-function approach to dual-rate controller design for unstable and non-minimum-phase plants, IEEE Transactions on Control Systems Technology 19 (5) (2011) 1186–1194.