Desarrollo de un Sistema para la Implementación de Controladores Lineales Multivariables

Juan Pablo Argüello Fajardo

Universidad Nacional de Colombia
Facultad de Ingeniería
Bogotá
2010

Desarrollo de un Sistema para la Implementación de Controladores Lineales Multivariables

Juan Pablo Argüello Fajardo

Director:
Leonardo Enrique Bermeo Clavijo

Tesis presentada para obtener el título de
Msc en Ingeniería Automatización Industrial

Universidad Nacional de Colombia
Facultad de Ingeniería
Bogotá
2010

Resumen

Se desarrolló un sistema hardware y software que permite la implementación digital de controladores para sistemas multivariables. La parte software tiene tres tareas principales:

- El diseño de controladores
- El acondicionamiento numérico de realizaciones.
- La comunicación con el hardware.

Para el diseño de controladores se emplea la estructura WKFque consiste en un controlador de dos grados de libertad (K) con red de desacople (W) y pre-compensador (F). La red de desacople se puede para sistemas con ceros de transmisión y se encuentra restringida únicamente a sistemas estables. Para el diseño de controladores se desarrollaron funciones LQG y H_∞ las cuales garantizan desacople de canales y error de estado estacionario nulo y poseen parámetros de diseño para definir la forma de la respuesta de lazo cerrado. El bloque pre-compensador actúa como un ecualizador permitiendo atenuar frecuencias de resonancia o aumentar el ancho de banda del sistema.

Las herramientas para acondicionamiento numérico son un conjunto de funciones que buscan reducir el deterioro de las realizaciones al evaluarse sobre un dispositivo de precisión finita, el cual introduce errores de cuantización y redondeo. El desborde numérico es una no linealidad que puede afectar la realizaciones en punto fijo en el momento en que se debe representar numéricamente un valor mayor al máximo representable. Se da solución a este problema mediante una función que calcula el valor máximo al evaluar una realización y a partir de esto es posible encontrar el formato numérico apropiado para implementar la realización. La estrategia de diseño LQG añade integradores para garantizar el desacople de canales lo cual implica que el controlador es inestable. Se propuso una técnica Antiwindup para acondicionar numéricamente la realizaciones con integradores y evitar el desborde de variables. Se trabajaron las implementaciones con operador q y operador δ con formato numérico en punto fijo y punto flotante.

El software de comunicación está compuesto por conjunto de funciones que permiten descargar realizaciones al hardware y realizar modificaciones. De forma paralela al software en Matlab se desarrolló un módulo de comunicación en lenguaje Python el cual permite programar realizaciones sin necesidad de Matlab, y tiene como ventaja el ser un lenguaje *open source* y multiplataforma.

El hardware se encuentra basado en un microcontrolador ARM7 que junto con los módulos para acondicionamiento de señal de entrada y salida, permite la implementación de sistemas de hasta cuatro entradas y cuatro salidas.

Las posibles aplicaciones de este sistema resultan ser amplias, teniendo en cuenta que a nivel industrial muchas plantas o procesos son multivariables, como es el caso de las calderas, columnas de destilación, biorreactores entre otros. En su mayoría estos sistemas son lentos y pueden ser linealizados, siendo posible usar este dispositivo para su control. En los cursos de Señales y Sistemas y Control se puede llegar a necesitar un dipositivo que actue como un filtro digital o como controlador y que pueda ser programado de forma rápida ya sea para demostraciones o para realizar prácticas de laboratorio.

El documento se organiza de la siguiente forma, en el capítulo dos se hace una revisión de conceptos relacionados con el diseño de sistemas de control multivariable e implementación de sistemas digitales, en el capítulo tres se hace una descripción del software desarrollado, el que se puede clasificar en dos grupos el de las funciones destinadas al diseño de sistemas de control y el de las funciones

orientadas al acondicionamiento numérico de realizaciones donde se propone un esquema antiwindup multivariable para implementaciones con integradores. En el capítulo cuatro se describe el hardware para implementación y se construye un modelo que tiene en cuenta los fenómenos de cuantización, redondeo y desborde siendo posible simular el comportamiento del sistema de control. En el capítulo 5 se describe una herramienta gráfica que integra análisis, diseño e implementación. Por último en el capítulo 6 se realizan algunas pruebas para comparar realizaciones con operador δ y operador q, y comparar realizaciones en punto fijo y punto flotante.

Contenido

Resumen v

1. Introducción 1
- 1.1. Planteamiento del problema . . . 2
- 1.2. Solución propuesta . . . 3

2. Sistemas multivariables y control 5
- 2.1. Notación . . . 5
- 2.2. Control Multivariable . . . 6
- 2.3. Control cásico . . . 6
  - 2.3.1. Control descentralizado . . . 6
  - 2.3.2. Control desacoplado . . . 8
- 2.4. Diseño en espacio de estados . . . 8
  - 2.4.1. Realimentación de estado . . . 8
  - 2.4.2. Realimentación de salida . . . 9
- 2.5. Control Robusto . . . 9
  - 2.5.1. Control H_∞ . . . 9
- 2.6. Reducción de Controladores . . . 11
  - 2.6.1. Truncamiento Balanceado . . . 11
  - 2.6.2. Aproximación de la norma de Hankel . . . 11
- 2.7. Sistemas de control digital . . . 12
  - 2.7.1. Diseño de controladores digitales . . . 12
  - 2.7.2. Parámetros en la implementación . . . 12

3. Sistema de desarrollo 17
- 3.1. Descripción . . . 17
- 3.2. Herramientas de Diseño . . . 18
  - 3.2.1. desacople . . . 18
  - 3.2.2. multivlqg . . . 19
  - 3.2.3. dhinfts . . . 23
- 3.3. Herramientas para acondicionamiento numérico de realizaciones . . . 24
  - 3.3.1. addaw . . . 24
  - 3.3.2. getdelta . . . 27
  - 3.3.3. nob (Número de bits fraccionales) . . . 28
  - 3.3.4. Funciones Auxiliares . . . 30
- 3.4. Herramienta *Python* para descarga de realizaciones . . . 32
  - 3.4.1. Protocolo para descarga de realizaciones . . . 32
  - 3.4.2. Funciones *Python* . . . 33

4. Hardware y Firmware **35**
4.1. Hardware . . . 35
4.1.1. Procesador . . . 36
4.1.2. Módulos del sistema . . . 36
4.2. Firmware . . . 37
4.2.1. Asignación de memoria . . . 37
4.2.2. Programa principal . . . 38
4.2.3. Comunicación . . . 40
4.3. Modelo del sistema . . . 41
4.3.1. Etapas de entrada y salida (Acondicionamiento de señal) . . . 41
4.3.2. Cálculo de realizaciones . . . 42
4.3.3. Ejecución de operaciones . . . 43
4.3.4. Tiempo de procesamiento . . . 45

5. Interfaz de usuario **47**
5.1. Descripción . . . 47
5.1.1. Estructura y planta . . . 48
5.2. Síntesis . . . 48
5.2.1. Síntesis de K . . . 49
5.2.2. Síntesis de W . . . 50
5.2.3. Síntesis de F . . . 50
5.2.4. Síntesis del controlador central . . . 52
5.3. Implementación . . . 53
5.4. Metodología de diseño . . . 54

6. Resultados **57**
6.1. Efecto del operador . . . 57
6.2. Efecto del formato numérico . . . 58
6.3. Casos de estudio . . . 60
6.3.1. Emulación . . . 60
6.3.2. Control . . . 61
6.3.3. Control LQG . . . 62
6.3.4. Control H_∞ . . . 63

Conclusiones **65**
6.4. Conclusiones . . . 65
6.5. Aportes . . . 66
6.6. Trabajo futuro . . . 67

Apendice 1 **69**
A.1. Instalación del driver usb . . . 69
A.1.1. PARA WINDOWS XP - Vista . . . 69
A.1.2. PARA LINUX . . . 69
A.1.3. Instalación Mac Os X leopard . . . 69
A.2. Descarga del firmware . . . 69
A.3. Formato para almacenamiento de realizaciones . . . 70

Apendice 2 **71**
B.2. Manual de usuario . . . 71
B.2.1. Modo de uso . . . 71
B.2.2. Definición de la estructura . . . 71
B.2.3. Definición de la planta . . . 72
B.2.4. Diseño . . . 73
B.2.5. Implementación . . . 76
B.2.6. Recursos en línea . . . 77

Lista de figuras

2.1. Configuración con 1Dof. . . . 6
2.2. Estructura para control descentralizado de un sistema 2×2 . . . 7
2.3. Estructura de control desacoplado . . . 8
2.4. Realimentación del vector de estado . . . 9
2.5. Control con observador (a) Realimentación de salida (b) Con integrador en el lazo . . . 9
2.6. (a) Configuracion General de control, (b)Problema de sensitividad mixta S/T . . . 10
2.7. Metodologías de diseño . . . 11
2.8. Sistema de control digital . . . 12
2.9. (a) Estructura con el operador memoria (b) Estructura δ usando como equivalente el operador q . . . 16

3.1. Fases en la implementación de un sistema . . . 17
3.2. Respuesta al paso de plantas precompensadas con redes de desacople (a) para el caso de una planta estable y de fase mínima (b) para una planta con ceros *RHP* . . . 19
3.3. Sistema de control con saturación en el actuador (a) *1DoF* (b) *2Dof* . . . 21
3.4. Comportamiento del sistema con varios controladores *LQG* basados en pesos escalares . 21
3.5. Comportamiento del sistema con control *LQG* con pesos matriciales, . . . 22
3.6. Respuesta en lazo cerrado con un controlador LQG de dimensión 2×1 . . . 22
3.7. Respuesta en lazo cerrado con controlador H_∞(a) para $t = [0{,}5\ 2]$ (b) para $t = [1\ 1]$ (c) para $t = [2\ 0{,}5]$ . . . 23
3.8. Controlador *AW* (a) para la configuración *1DoF* (b) para la configuración *2Dof* . . . 24
3.9. Controlador con *AW* implementado en Simulink . . . 26
3.10. Salida y Señal de control del sistema en lazo cerrado (1) con actuador ideal, (2) con actuador limitado en rago (3) con actuador limitado y controlador con *AW* . . . 27
3.11. Esquema propuesto para evaluar γ . . . 28

4.1. Esquema general del sistema de control . . . 35
4.2. Diagrama funcional del microcontrolador ADuC702X . . . 36
4.3. Circuitos para adecuación de señal (a) etapa de salida (b) etapa de entrada . . . 37
4.4. Mapa de memoria . . . 37
4.5. Tipos de canales de entrada y ejemplos . . . 38
4.6. Modos de operación programa principal . . . 39
4.7. Diagrama de flujo detección de puerto y sincronización . . . 41
4.8. Modelo entrada salida característica entrada salida . . . 42
4.9. Región de implementación . . . 46

5.1. Paneles de elección de estructura y definición de la planta . . . 48
5.2. Estructuras de control, los subíndices corresponden a la dimensión de la señal y la región sombreada al componente implementado . . . 49
5.3. Panel para síntesis de K con los subpaneles para diseño *LQG*, H_∞ y control *SISO*. . . . 49
5.4. Panel para síntesis del elemento de desacople W . . . 50
5.5. Panel para síntesis del precompensador F . . . 51

5.6. Ejemplo de aplicación de un filtro notch 51
5.7. Representación en espacio de estados del controlador central 53
5.8. Panel de implementacion 53

6.1. Sistema emulado con operador δ y operador q en punto fijo con diferentes formatos numéricos 58
6.2. Características de la planta multivariable (a) Respuesta al paso (b) gráfico de valores singulares 58
6.3. Respuesta en lazo cerrado usando controladores LQG y H_∞ 59
6.4. Respuesta en lazo cerrado sistema con controlador H_∞implementado en punto flotante y punto fijo 59
6.5. Señal de control sistema con controlador H_∞implementado en punto flotante y punto fijo 60
6.6. Respuesta del sistema emulando una planta 2×2 desacoplada 60
6.7. Respuesta del sistema emulando una planta 2×2 acoplada 61
6.8. Respuesta en lazo cerrado para una planta siso con control LQG 62
6.9. Respuesta en lazo cerrado para un sistema multivariable con control LQG 62
6.10. Respuesta en lazo cerrado para un sistema multivariable con control H_∞ 63

Lista de tablas

1.1. Relación número de estados dimensión para algunos sistemas multivariables 3

2.1. Métodos de discretización de controladores 13

3.1. Comandos en Matlab para diseño LQG con pesos escalares 21
3.2. Comandos en Matlab para diseño LQG con pesos matriciales 22
3.3. Comandos en Matlab para diseño H_∞con restricción en el tiempo de establecimiento . . 23
3.4. Diseño LQG con antiwidup 26
3.5. Número de bits enteros mínimos para implementación en punto fijo 30
3.6. Ejemplo de protocolo para comunicación con el dispositivo hardware 32
3.7. Funciones Python 33

4.1. Contenido de los bloques de memoria 38
4.2. Solicitudes al microcontrolador 41
4.3. Tiempo de procesamiento para realizaciones $SISO$ con operador q 45

Capítulo 1
Introducción

1.1. Planteamiento del problema 2
1.2. Solución propuesta 3

A finales de los años 50 se inicia el interés por el control de procesos multivariables con los trabajos de *Povejsil(1955), Kavanagh(1956), Freeman(1957)*, donde mediante un tratamiento algebraico, y criterios para asegurar desacople de lazos se intenta sintetizar los primeros controladores multivariables, posteriormente en 1960, *Kalman* con su articulo *" On the general theory of control systems"* mostró la dualidad entre sistemas de control realimentado y el filtros realimentados, esto da origen al conocido filtro de Kalman en 1961, a los criterios de observabilidad y controlabilidad de sistemas multivariables desarrollados por Gilbert en 1963, seguido de trabajos como el de Brockett [1] donde se analiza el efecto de la realimentación de estados, Luenberger desarrolla los observadores para sistemas multivariables (1966) donde se pretende recuperar estados no observables, Womham y Pearson [2, 3] desarrollan técnicas para la asignación de polos en lazo cerrado, simultáneamente se desarrolla la teoría de control óptimo con el LQR (Linear Quadratic Regulator) y LQG (Linear Quadratic Gaussian) donde se pretende asignar los polos en lazo cerrado para obtener respuestas óptimas como tiempo mínimo, mínimo consumo de energía. En 1960 Horowitz [4] propone un método algebraico para diseño, y analiza el efecto de las variaciones en el modelo de la planta posteriormente su investigación se centra en el diseño de sistemas de control descentralizados con publicaciones como [5], donde se asumen pocesos fácilmente diagonalizables. En 1966 Bristol publica [6], donde propone el RGA (Relative Gain Array) como medida de la interacción en sistemas multivariables, con este articulo se presenta un criterio para la elección de lazos de control que serviría de inspiración a trabajos como Liu [7], donde se proponen condiciones necesarias para el desacople de sistemas multivariables.

Entre los 70's y 80's surgen aportes significativos referentes al análisis en frecuencia de sistemas multivariables, en particular el concepto de matriz coprima introducido por *Youla* en 1976, también se generalizó a sistemas multivariables el criterio de estabilidad de *Nyquist* por parte de *Rosenbrock, Macfarlne*[8] y *Postlewaite* y posteriormente en 1977 *Safonov* [9] demostró que la definición de margen de fase y margen de ganancia puede ser extendida a sistemas multivariables. En 1979 *Doyle* y *Stein* proponen el *LTR* (loop transfer recovery) como una versión robusta del *LQG* y en 1980 *Paraskevopoulos* [10] proponen métodos de dise no para control de sistemas multivariables mediante lazos PID. 1981 es uno de los años más productivos donde se encuentran importantes trabajos en su mayoría orientados al uso de valores singulares en el análisis y diseño de sistemas multivariables, ejemplo de esto es Klema,Moore y Postlethwaite [11, 12, 13], por otra parte se encuentra en este mismo año la publicación de Pernebo [14, 15] donde se propone la topología general de control o de modelo generalizado de la planta con un diseño basado en polinomios generalizados. Quizá el trabajo mas significativo de ese año fue el presentado por Doyle [16] titulado *"Multivariable Feedback design: Conceps for a Clasical/modern Synthesis"* donde se analizan los efectos de las incertidumbres estructuradas, dando origen al problema de control H_∞, en 1982 se introducen los procedimientos para síntesis de controladores multivariables,

el primero de estos propuesto por *Zames* y *Francis*, posteriormente en 1987 *Doyle* propone una solución en espacio de estados al problema H_∞.

Una vez se diseña un controlador o filtro digital se debe llevar a una realización para su implementación sobre un dispositivo de procesamiento, la realización está compuesta por un conjunto de coeficientes que conforman una ecuación recursiva. Un sistema posee infinitas realizaciones cuando es implementado con presición infinita, pero cuando la implementación se realiza sobre un dispositivo de procesamiento los coeficientes que componen la realización deben ser ajustados al formato numérico del dispositivo, las señales de entrada son cuantizadas por los conversores analógico-digital *(ADC)* y digital- analógico *(DAC)* , las operaciones internas involucran el redondeo típico de la matemática de presición finita en general la realización se comportará de forma diferente. Es de esperarse que existan conjuntos de realizaciones que por su condicionamiento numérico presentan un mejor comportamiento. En la literatura hay una buena cantidad de documentos relacionados con la busqueda de realizaciones óptimas, siendo tres los principales objetivos de la optimización, por una parte se proponen técnicas que baśandose en la caracterización del dispositivo de procesamiento buscan una matriz de transformación que permite minimizar el error de los coeficientes que componen la realización [17], otros buscan minimizar el número de bits necesarios para representar los coeficientes 'mantenido un índice de desempeño [18] , por último se encuentran los interesados en conocer el factor de escala necesario para evitar el desborde cuando las implementaciones son en punto fijo [19] basandose en el valor máximo que podrían presentar las variables de estado.

Los primeros trabajos relacionados con el análisis de sistemas de control implementados con precisión finita se remontan a [20], donde se analiza el efecto de la cuantización en sistemas de control digital , en [21] se analizan sistemas con longitud de palabra finita, este análisis se realiza basado en la configuración general de control, donde los ruidos de cuantización son señales exógenas al sistema, adicionalmente se imponen condiciones para el escalamiento del sistema que previenen el desborde. En [22, 19] se propone el problema de escalización de controladores en punto fijo como un problema de optimización con el fin de evitar el desborde, uno de los aspectos importantes de este documento es que destaca la importancia de factores de escala no diagonales para sistemas multivariables. Algunos trabajos mas recientes como [23], [17] orientan la optimización a minimizar el rango dinámico de la realización y proponen realizaciones no mínimas como la forma δ y ρ [24] o la forma implícita las cuales buscan reducir la sensitividad de la implementación.

1.1. Planteamiento del problema

El desarrollo de un sistema de control implica el diseño y síntesis de controladores, en un curso de control para pregrado o posgrado se enfatiza en los aspectos teóricos del diseño, en el cumplimiento de especificaciones de desempeño o de robustez pero generalmente se omite la implementación ya que en estos cursos la simulación suele ser la principal herramienta de validación. En los casos en que se aborda el problema de implementación el estudiante debe destinar la mayor parte del tiempo a la creación de controladores con amplificadores operacionales o con microcontroladores desperdiciando tiempo que podría ser empleado en la validación de otras estrategias de control. Es muy usual que en un curso de señales se hable de filtros digitales, pero nunca se habla de como implementarlos porque se asume que eso se debe aprender en el curso de sistemas digitales. También se observa que muchos estudiantes que están a punto de terminar sus estudios de pregrado tienen bien fundamentados los conceptos de control y además tienen un muy buen background de sistemas digitales, pero cuando se les habla de control digital no tienen muy claro e incluso lo llegan a confundir con el control de eventos discretos, esto debido a que no se contó con los recursos para implementar sistemas digitales de forma rápida y fácil.

Por otra parte se encuentra que a nivel industrial los procesos son controlados de forma sub óptima debido a que las estrategias de control suelen ser muy limitadas, se podría hablar de dos el control *on-off* y el control *PID*. La implementación de cualquier otra estrategia implica el montaje de un controlador industrial el cual resulta ser un computador adecuado para operar en ambientes industriales y que se caracteriza por ser muy costoso, una revisión bibiográfica mostró que muchos procesos industriales son multivariables y que además poseen un modelo lineal como se muestra en la tabla 1.1.

Por ejemplo para controlar una columna de destilación como la propuesta en [25] se puede sintetizar

Nombre	Número Estados	Tamaño	Fuente
Columna de destilación	4	2×2	[25]
Condensador de un reactor químico	4	2×2	[26]
Caldera	3	3×3	[28]
Reactor de polimerización	2	2×2	[29]
Disco duro	4	2×2	[30]
Jet-dinámica longitudinal	6	2×2	[31]

Tabla 1.1: Relación número de estados dimensión para algunos sistemas multivariables

un controlador LQG que con integradores resulta ser de orden 6, entonces viene la pregunta ¿ Es necesario un computador industrial para una realización de orden 6?. ¿ No podría un pequeño sistema de control embebido en un microcontrolador realizar esta tarea?

1.2. Solución propuesta

Se propone la construcción de una plataforma programable para ejecución de realizaciones lineales con multiples entradas y salidas, la plataforma podrá ser controlada desde una herramienta como Matlab en la cual se podrá asistir el proceso de diseño de algoritmos de control. Debido a que Matlab es un software comercial se desarrollará una interfaz en software libre que permita que la implementación se pueda realizar sin Matlab.

La herramienta se orientará principalmente al control de procesos multivariables aunque se podrá ser empleada como filtro digital multicanal o para emulación de sistemas. También servirá como herramienta didáctica en la validación de estrategias de control.

Adicionalmente la herramienta podrá ser empleada en el estudio de técnicas de acondicionamiento numérico para implementaciones con precisión finita, siendo posible analizar los efectos del formáto numérico, operador, estructura y estrategias de control antiwindup.

Capítulo 2

Sistemas multivariables y control

2.1. Notación **5**
2.2. Control Multivariable **6**
2.3. Control cásico **6**
2.3.1. Control descentralizado 6
2.3.2. Control desacoplado 8
2.4. Diseño en espacio de estados **8**
2.4.1. Realimentación de estado 8
2.4.2. Realimentación de salida 9
2.5. Control Robusto **9**
2.5.1. Control H_∞ 9
2.6. Reducción de Controladores **11**
2.6.1. Truncamiento Balanceado 11
2.6.2. Aproximación de la norma de Hankel 11
2.7. Sistemas de control digital **12**
2.7.1. Diseño de controladores digitales 12
2.7.2. Parámetros en la implementación 12

2.1. Notación

Se considera un sistema, proceso o planta multivarible lineal a un conjunto de entradas y salidas que interactuan y se puede expresar de la forma $y = Gu$, siendo u las señales de entrada al sistema, y las señales de salida y G su matriz de transferencia. Para referirse al tamaño del sistema se usa la notación $(p \times q)$ donde q es el número de entradas y p el número de salidas. de modo que $y \in \mathbb{R}^{1\times p}$, $u \in \mathbb{R}^{q\times 1}$ y $G \in \mathbb{R}^{p\times q}$.

Un sistema lineal con función de transferencia $G(\sigma)$ donde $\sigma = s, z$ según sea el caso, tiene una representación en espacio de estados de la forma(2.1) para sistemas continuos o (2.2) para sistemas discretos.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.1}$$

$$\begin{aligned} x[n+1] &= Ax[n] + Bu[n] \\ y[n] &= Cx[n] + Du[n] \end{aligned} \tag{2.2}$$

Siendo n el número de estados, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{q\times n}$ y $D \in \mathbb{R}^{p\times q}$ esta notación representa a un sistema con función de transferencia $G(s)$ de p entradas, q salidas y orden n.

También puede ser representado de una forma compacta como la mostrada en (2.3) donde se presentan únicamente los coeficientes que componen la realización.

$$G(\sigma) = C(\sigma I - A)^{-1}B + D \triangleq \left[\begin{array}{c|c} \text{A} & \text{B} \\ \hline \text{C} & \text{D} \end{array}\right] \tag{2.3}$$

Un sistema en espacio de estados con presición infinita posee infinitas realizaciones dadas por $[TAT^{-1}, TB, CT^{-1}, D]$ siendo T una matriz de transformación que lleva de un conjunto de variables de estado a otro; necesariamente $T \in \mathbb{R}^{n\times n}$.

2.2. Control Multivariable

El esquema de control mas común es el de 1DoF (un grado de libertad) mostrado en la figura (2.1), este puede manipular los polos del sistema para lograr características tales como tiempo de asentamiento y sobrepico máximo. Por definición el producto $L \triangleq GK$ se denomina función de lazo, $S \triangleq (I + GK)^{-1}$ es la función de sensitividad y $T \triangleq (I + GK)^{-1}GK$ es la función de sensitividad complementaria. S y T cumplen $S + T = 1$.

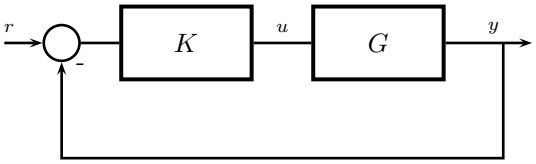


Figura 2.1: Configuración con 1Dof.

Se denominará canal a una pareja entrada-salida un sistema de control tiene tantos canales como salidas tenga la planta. Un canal se encuentra desacoplado cuando es independiente del valor de los otros canales, un canal esta acoplado si depende del valor de los otros canales. En un sistema multivarible los acoples entre canales aumentan la complejidad en el diseño del controlador pues se busca que por cada salida exista una referencia pero el acople hace que las salidas sean dependientes de varias entradas. Las primeras estrategias de diseño conocidas como control por desacople o control desacoplado buscaban garantizar una mínima interacción entre parejas de entrada y salida de modo que el sistema podía ser controlado por varios compensadores SISO ejemplos de esto se encuentran en [5, 7, 32].

Posteriormente las estrategias de control centralizado por variables de estado, y el control H_∞ permitieron la obtención de controladores que neutralizan las interacciones. En un control centralizado se dispone de un conjunto de sensores, y mediante un algoritmo de control se calcula una acción para todos los actuadores disponibles. Esta estrategia generalmente requiere el uso de computadores industriales, y suele ser la única forma de controlar sistemas fuertemente acoplados.

2.3. Control cásico

2.3.1. Control descentralizado

Sea $G(s)$ una planta cuadrada su controlador descentralizado es un controlador con matriz de transferencia diagonal $K = diag\,[K_1, K_2, \cdots]$, para un caso (2×2) puede ser representado por una estructura como la mostrada en la figura (2.2). Para el diseño de este controlador se seleccionan los pares entrada salida, y posteriormente se diseña un controlador para cada lazo, al implementar una estrategia como esta K_1 y K_2 son independientes el uno del otro, incluso pueden pertenecer a distintos dominios (eléctrico, mecánico,..). El uso de esta estrategia se restringe a sistemas con baja interacción entre lazos.

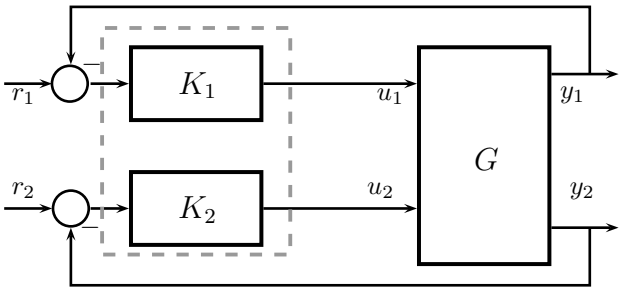


Figura 2.2: Estructura para control descentralizado de un sistema 2×2

Selección de lazos de control

Para la selección de lazos se emplea el *RGA (Relative Gain Array)* propuesto en [6], el *RGA* es una herramienta de análisis que brinda información acerca de las interacciones en sistemas *MIMO* y además permite predecir el comportamiento de las respuestas controladas. La matriz de ganancias relativas tiene la forma mostrada en (2.4), donde los elementos λ_{ij} definidos en (2.5) son la relación entre la ganancia estática manteniendo constante todas la entradas $u \neq i$ y la ganancia estática manteniendo constante todas las salidas $y \neq j$

$$\Lambda = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1j} \\ \vdots & \ddots & \vdots \\ \lambda_{i1} & \cdots & \lambda_{ij} \end{bmatrix} \tag{2.4}$$

$$\lambda_{ij} = \left[\frac{\Delta y_i}{\Delta u_j}\bigg|_{u_m=cte} \right] \left[\frac{\Delta y_i}{\Delta u_j}\bigg|_{y_n=cte} \right]^{-1} \qquad m \neq i, n \neq j \tag{2.5}$$

Otra alternativa es evaluar la matriz de ganancias relativas en forma matricial haciendo un producto elemento a elemento de.

$$RGA(G(j\omega)) = G(j\omega). \times (G(j\omega)^{-1})^T$$

La selección de pares se realiza dependiendo del punto de operación, para lo cual se prefiere evaluar el *RGA* cerca a la frecuencia de cruce ω_0 y seleccionar parejas cuyos valores se encuentren cercanos a uno, se evita elegir parejas cuyo valor λ_{ij} sea alto o negativo. Los valores negativos indican la presencia de comportamientos inestables en lazo cerrado, debido ceros en el semiplano derecho, los valores muy grandes indican que el control no se realizará de forma efectiva.

Diseño de controladores

En un control descentralizado se asume que la interacción entre lazos es mínima por lo tanto se trabaja con modelo aproximado de la planta $\hat{G}$ el cual está compuesto por la diagonal de G. Con lo cual es posible diseñar controladores *SISO* usando métodos tradicionales.

Cuando la planta no es diagonal $\hat{G} \neq G$ y el comportamiento en lazo cerrado será diferente debido a que la interacción de lazos hace que el cambio de referencia en un canal afecte a los demas. Una forma de evaluar la estabilidad del sistema es usando el teorema de *Gershgorin* [33], el cual define la condición (2.6) como condición suficiente para determinar la estabilidad del sistema en lazo cerrado. Esta condición se basa en el acotamiento del error entre el sistema real y el aproximado asumiendo $E = (G - \hat{G})\hat{G}^{-1}$, es posible expresar la función de lazo abierto $(1 + GK)$ en términos del error E y la función de lazo cerrado aproximada $\hat{T}$ siendo $(I + GK) = (I + E\hat{T})(I + \hat{G}K)$, es posible afirmar que el sistema en lazo cerrado es estable si $(I + E\hat{T})$ no tiene ceros en el semiplano derecho.

$$\|T(j\omega)\| \leq \frac{\|g_{ii}(j\omega)\|}{\sum_{j \neq i} \|g_{ij}(j\omega)\|} = \frac{1}{\rho_i(\omega)} \tag{2.6}$$

La teorema de *Gershgorin* define el radio de Gershgorin ρ_i como el inverso de la relación entre la norma del lazo principal y las suma de las normas de las otras funciones (funciones cruzadas o que no hacen parte de la diagonal) esto guarda relación con la función de lazo cerrado $\hat{T}$, ya que si no hay dominancia diagonal ρ_i es grande lo cual indica que $\hat{T}$ sufrirá un gran deterioro con respecto a T.

2.3.2. Control desacoplado

El control desacoplado se basa en el uso de un compensador $W(s)$ previo a la planta el cual tiene como finalidad neutralizar o minimizar la interacción entre canales, el objetivo es hacer que el producto $G(s)W(s)$ sea aproximadamente diagonal y por tanto más fácil de controlar. La red de desacople puede ser calculada para tres diferentes casos.

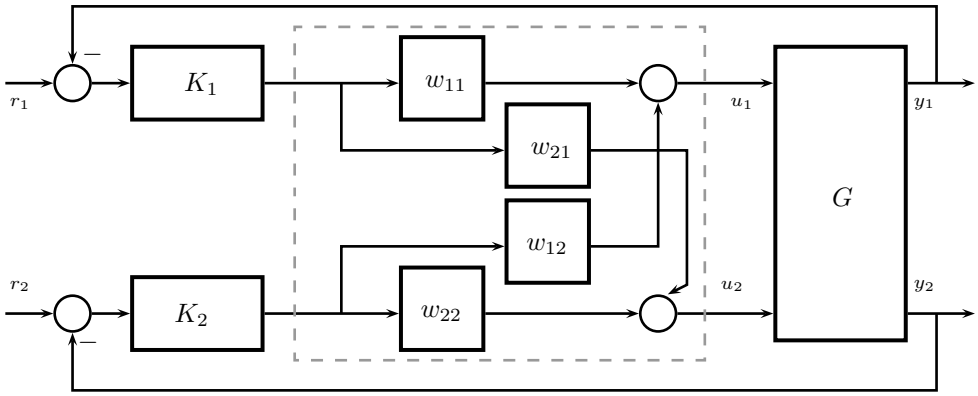


Figura 2.3: Estructura de control desacoplado

- **Desacople dinámico:** Se diagonaliza la planta haciendo $G_d(s) = G(s)W(s) = I$ donde $W(s) = G(s)^{-1}$, la planta debe ser invertible y su inversa es realizable. Cuando la planta no es cuadrada se puede cambiar D^{-1} por $D^{\dagger}$
- **Desacople en estado estable:** $G_d(0)$ es diagonal al hacer $W = G^{-1}(0)$. Este desacople garantiza error de estado estacionario cero.
- **Desacople aproximado en la frecuencia ω_0:** Hace que $G_d(j\omega_0)$ sea diagonal, usualmente se calcula la red de desacople haciendo que $W = G_0^{-1}$ donde G_0 es una aproximación de la planta en la frecuencia ω_0.

En general las redes de desacople pueden presentar una alta sensibilidad a los errores en el modelamiento y las incertidumbres, además las plantas con ceros de fase no mínima puede introducir cancelaciones no permitidas.

2.4. Diseño en espacio de estados

2.4.1. Realimentación de estado

En una realimentación de estado hay un sensor asociado a cada uno de los estados del sistema [34]. En la figura (2.4) se presenta la estructura de una realimentación del vector de estado, la señal de control esta dada por $u = -K_e x + r$ donde K_e es una matriz constante que permite establecer los polos del sistema en lazo cerrado ya que el sistema realimentado tiene la forma $\left[\begin{array}{c|c} A - BK_e & B \\ \hline C & D \end{array}\right]$, su matriz de transferencia es $A_{lc} = A - BK_e$ y es posible asignar valores propios o polos del sistema solucionando.

$$det(sI - A_{Lc}) = det(sI - (A + BK_e))$$

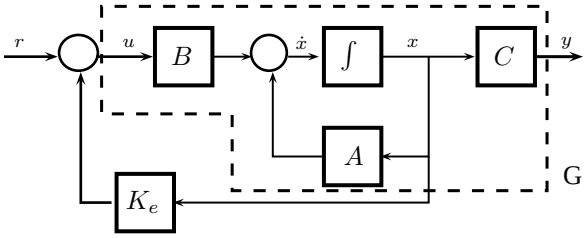


Figura 2.4: Realimentación del vector de estado

En caso de no poseer una idea muy clara de donde situar los polos del sistema se puede usar una función de peso como (2.7) donde se buscan posiciones adecuadas para los polos dependiendo de los pesos Q y R que penalizan los estados y la señal de cotrol.

$$J_r = \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t)\right) dt \tag{2.7}$$

La solucion a este problema es de la forma $K = -B^T P x$ donde P es una matriz semidefinida positiva $P > 0$ y es solución de la ecuación de *Riccati*.

$$A^T P + PA - PBB^T P + C^T C = 0$$

2.4.2. Realimentación de salida

Cuando no es posible medir algún estado del sistema directamente para aplicar una estrategia de realimentación de estados, es necesario reconstruir los estados faltantes [35], para lo cual se emplea un *filtro de Kalman* también conocido como observador resultando una estructura como la mostrada en la figura (2.5.a).

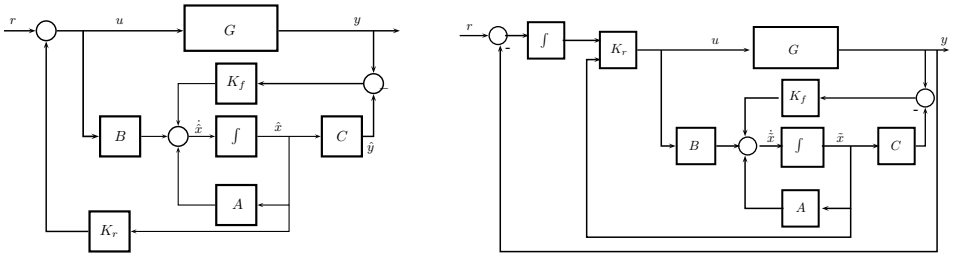


Figura 2.5: Control con observador (a) Realimentación de salida (b) Con integrador en el lazo

Los estados estimados están descritos por $\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y})$ y la salida estimada esta dada por $\hat{y} = C\hat{x}$. Aunque usando esta configuración es posible situar los polos del sistema siempre se presenta error de estado estacionario lo cual se corrige con el uso de un precompensador o con la adición de integradores.

2.5. Control Robusto

2.5.1. Control H_∞

Sea P una planta generalizada descrita en espacio de estados de la forma (2.8) y con una estructura como la mostrada en la figura (2.6)

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_1 x + D_{12} u \\ y &= C_2 x + D_{21} w \end{aligned} \tag{2.8}$$

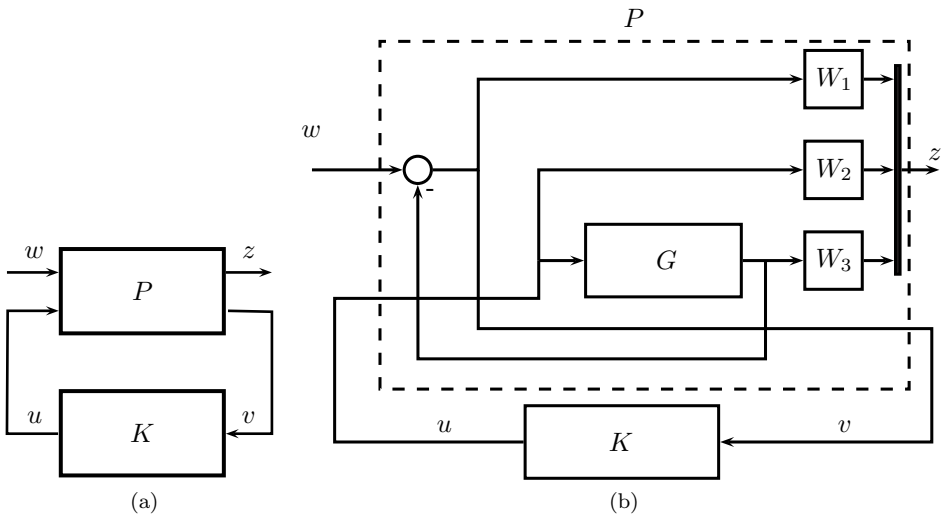


Figura 2.6: (a) Configuracion General de control, (b)Problema de sensitividad mixta S/T

Donde u son las variables de control, v son las variables medidas, w son señales exogenas (perturbaciones, señales de referencia) y z son las señales de error. El objetivo de esta estrategia es encontrar un controlador K el cual permita minimizar la norma infinito de w a z, esto es:

$$\|\mathcal{F}(P,K)\| = \max_{w\neq 0} \frac{\|z\|}{\|w\|} < \gamma$$

Donde $\mathcal{F}(:)$ es una LFT (Linear Fractional Transformation) y γ es un índice de desempeño. El problema se descompone en dos: la síntesis de un controlador H_∞ con información completa y el diseño de un filtro H_∞(Observador). El diseño del controlador con información completa consiste en encontrar un $P > 0$ que sea solución de la ecuación de Ricatti

$$A^T P + PA + C^T C + P(\gamma^{-2} B_1 B_1^T - B_2 B_2^T)P = 0$$

Para asegurar estabilidad los valores propios de $A + (\gamma^{-2} B_1 B_1^T - B_2 B_2^T)P$ deben permanecer en el semiplano izquierdo (SPI) . El filtro H_∞consiste en encontrar un $Q > 0$ que cumpla la ecuación de Ricatti

$$QA^T + AQ + C^T C + B_1 B_1^T + Q(\gamma^{-2} C_1^T C_1^T - C_2^T C_2)Q = 0$$

Tal que $\lambda\left[A + (\gamma^{-2} B_1 B_1^T - B_2 B_2^T)P\right] \in SPI$ y $\rho(PQ) < \gamma^2$ donde ρ es el radio espectral.

El controlador K tendrá la forma (2.9):

$$K = (I - \gamma^{-2} QP)^{-1} QC_2^T (sI - (A + \gamma^{-2} B_1 B_1^T P - B_2 B_2^T P + (I - \gamma^{-2} QP)^{-1}))B_2^T P \tag{2.9}$$

Es posible usar un esquema de bisección para lograr un valor deseado de γ, o incluso hacer que la salida del modelo generalizado z sean señales equalizadas en frecuencia mediante filtros o pesos que penalizan el error o la amplitud de la señal de control en un determinado ancho de banda obteniendo características como el rechazo a perturbaciones, mínimo consumo de energía y tolerancia a incertidumbres. Para establecer el desempeño del sistema en lazo cerrado se suele ulilizar la sintesis mixta que consiste en sintetizar un controlador H_∞ minimizando la función

$$\left\| \begin{bmatrix} W_1 S \\ W_2 T \\ W_3 K S \end{bmatrix} \right\|_\infty$$

Con lo cual se puede establecer características como ancho de banda, sobrepico de la función de lazo cerrado o sobrepico de la función de sensibilidad.

2.6. Reducción de Controladores

Los métodos de diseño como H_∞y LQG resultan en controladores con un orden por lo menos igual al orden de la planta o mayor si se incluyen pesos, esto hace que la implementación sea compleja. En la figura (2.7) se resume la forma como es posible hacer un diseño y obtención de un controlador de orden reducido. En [36, 37] se proponen un método para el diseño H_∞de orden reducido basado en LMIs para obtener un observador de orden reducido. El objetivo es encontrar un sistema de orden reducido $G_r(s)$ tal que la norma del error $\|G(s) - G_r(s)\|$ es pequeña. Se presentan los siguientes métodos para llevar a cabo esta reducción.

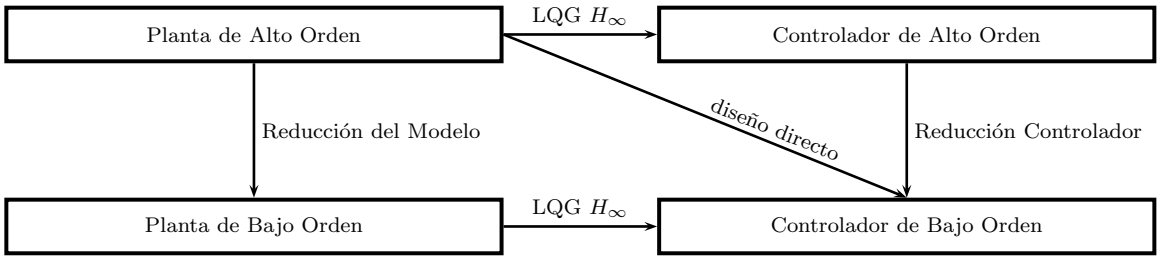


Figura 2.7: Metodologías de diseño

2.6.1. Truncamiento Balanceado

La principal idea de este método es el no tener en cuenta las partes del sistema que son menos observables y controlables, obteniendo un sistema de orden reducido con una dinámica fundamental similar al sistema original. El error introducido por el truncamiento es significativamente menor si esta se aplica a sistemas balanceados, en sistemas desbalanceados se puede aplicar el siguiente algoritmo para calcular la realización balanceada. ver [38].

- Calcular los gramianos de observabilidad P y de controlabilidad Q.
- Calcular el factor de Cholesky R tal que $Q = R^T R$
- Aplicar descomposición en valores singulares a RPR^T tal que $RPR^T = U\Sigma U^T$.
- Se calcula $T = \Sigma^{-\frac{1}{2}} U^T R$
- la realización balanceada será $[TAT^{-1},\ TB,\ CT^{-1},\ D]$

Una vez obtenida la realización mínima se descomponen la matriz $\Sigma = [\Sigma_1, \Sigma_2]$ con lo cual las matrices (A, B, C) se pueden expresar de la forma $A = \left(\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right)$, $B = \left(\begin{smallmatrix} B_1 \\ B_2 \end{smallmatrix}\right)$ y $C = \left(\begin{smallmatrix} C_1 & C_2 \end{smallmatrix}\right)$ el sistema de orden reducido será.

$$G_r(s) = C_1(sI - A_{11})^{-1}B_1 + D = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array}\right]$$

$G_r(s)$ es llamado el truncamiento balanceado del sistema $G(s)$ y cumple

$$\|G(s) - G_r(s)\|_\infty \leq 2tr(\Sigma_2)$$

2.6.2. Aproximación de la norma de Hankel

La norma Hankel denota la mayor ganancia $\mathcal{L}_2$ en una transición. El sistema $G(s)$ con representación $[A, B, C, D]$ y con gramianos $P = Q = diag\,[\Sigma_1, \sigma I]$ donde σ es el menor valor singular y Σ_1 es diagonal mayor que σ. Se puede construir un sistema $G_r(s) = [\hat{A}, \hat{B}, \hat{C}, \hat{D}]$ a partir de

$$\begin{array}{ll} \hat{A} = \Gamma^{-1}(\sigma^2 A_{11}^T + \Sigma_1 A_{11}\Sigma_1 - \sigma C_1^T U B_1^T) & \hat{B} = \Gamma^{-1}(\Sigma_1 B_1 + \sigma C_1^T U) \\ \hat{C} = C_1\Sigma_1 + \sigma U B_1^T & \hat{D} = D - \sigma U \end{array}$$

Donde U satisface $B_2 = -C_2^T U$ y $\Gamma = \Sigma_1^2 - \sigma^2 I$. Se cumple que

$$\|G(s) - G_r(s)\|_\infty = \sigma$$

2.7. Sistemas de control digital

El desarrollo de la microelectrónica ha dado como resultado dispositivos de procesamiento cada vez más veloces, con mayor memoria y menor precio haciendo que los controladores preferiblemente sean implementados de forma digital, embebidos en un procesador que almacena y ejecuta el algoritmo de control. El control analógico se reserva para casos especiales con anchos de banda grande, o donde se requieren sistemas altamente confiables y de bajo consumo energético. Actualmente el control de plantas multivariables se lleva a cabo por medio de controladores digitales implementados en un computador industrial.

El lazo de control mostrado en la figura (2.8) consta de una fase de adquisición que involucra una conversión análoga a digital (A/D), una etapa de procesamiento donde se ejecuta el algoritmo de control y una etapa de salida de señal donde realiza una conversión digital análoga *(D/A)*. Un sistema digital es comandado por un reloj que opera a una velocidad (T) conocida como periodo de muestreo, esta determina el tiempo entre muestras para el conversor A/D, el tiempo que existe entre el ćalculo de dos estados para el algoritmo $K(z)$ y la velocidad a la cual se actualiza el conversor *(D/A)*.

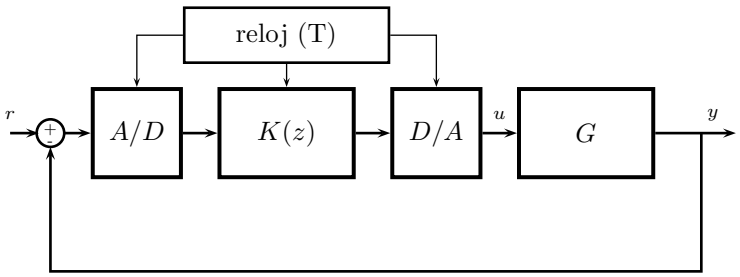


Figura 2.8: Sistema de control digital

2.7.1. Diseño de controladores digitales

Aunque la tendencia es el diseño en tiempo continuo y la posterior discretización del controlador en [39] se proponen métodos para el diseño en tiempo discreto y en [40] se proponen métodos para el diseño de controladores H_∞ en tiempo discreto. Para la discretización se disponen de diferentes métodos los más comunes se resumen en la tabla (2.1) donde $\mathcal{Z}[]$ corresponde a la transformada Z y $\mathcal{L}$ a la transformada de *Laplace*, $C(s)$ es el controlador en tiempo continuo, T el periodo de muestreo, y p_i, c_i los polos y ceros del sistema en tiempo continuo. Para una descripción más detallada de los métodos de discretización consultar [41].

2.7.2. Parámetros en la implementación

La implementación consiste en programar un dispositivo digital para evaluar el algoritmo de control, con respecto al hardware se deben tener en cuenta los siguientes aspectos.

- Los registros para almacenamiento de coeficientes tienen longitud finita por lo tanto no se almacena el valor real del coeficiente sino una versión cuantizada.

- Existe un tiempo de procesamiento debido a las operaciones que debe realizar el dispositivo, este restringe el periodo de muestreo al no poder evaluar el estado siguiente sin haber terminado de evaluar el estado presente.

Invarianza al impulso	$C(z) = \mathcal{Z}\left[\mathcal{L}^{-1}C(s)\right]$
Invariaza al paso	$C(z) = \frac{z-1}{z}\mathcal{Z}\left[\mathcal{L}^{-1}\left[\frac{C(s)}{s}\right]\right]$
Aproximación hacia adelante	$C(z) = C(s)\mid_{s=\frac{z-1}{T}}$
Aproximación hacia atras	$C(z) = C\left(s\right)\mid_{s=\frac{z-1}{Tz}}$
Aproximación bilinear	$C(z) = C\left(s\right)\mid_{s=\frac{2}{T}\frac{z-1}{z+1}}$
Mapeo de polos y ceros	$C(z) = K\frac{\prod_{j=0}^{nc}(z-e^{-c_j T})}{\prod_{i=0}^{np}(z-e^{-p_i T})}$

Tabla 2.1: Métodos de discretización de controladores

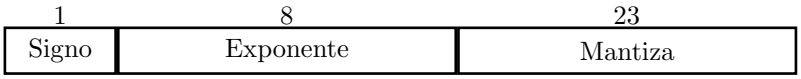
- El uso de conversores A/D y D/A adiciona ruido de cuantización al sistema y limita la precisión debido a que el número de bits de los conversores es menor al número de bits de los registros encargados del procesamiento.

En [42, 19] se proponen estrategias para reducir los efectos de la precisión finita en implementaciones digitales, se observa que la representación numérica juega un papel fundamental en la reducción de los errores computacionales mientras que los efectos del *A/D* y *D/A* dependen directamente del dispositivo hardware. A continuación se analizan con mayor detalle otros aspectos importantes en la implementación de controladores.

Representación numérica

Se refiere a el formato en el cual es almacenado y procesado un valor numérico. Las representaciones más utilizadas son la representación en punto y fijo y en punto flotante.

Representación numérica punto flotante Un número se expresa en punto flotante de la forma $x = (-1)^s \times m \times 2^e$ donde s es el signo $(+,-)$, m es la mantisa $m \in [0{,}5, 1)$ y e es el exponente, el estándar *IEEE 754* establece que los números en punto flotante de longitud 32 bits asignan un bit de signo$\beta_s = 1$, ocho bits de mantisa $\beta_b = 8$ y veintitrés bits de exponente $\beta_e = 23$. el rango que se maneja es de $1{,}8 \times 10^{-38}$ a $3{,}4 \times 10^{38}$.

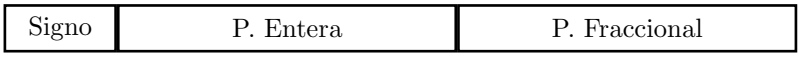


Representación numérica en punto fijo Un número en punto fijo se escribirá de la forma $A(a, b)$ donde a representa la longitud de la parte entera y b la longitud de la parte fraccional. Una palabra binaria de longitud N cuando es interpretada como racional con signo en punto fijo representa un valor

$$x = \frac{1}{2^b}\left(-2^{N-1}x_{N-1} + \sum_{0}^{N-2} 2^n x_n\right) = A(a, b)$$

x_n representa el bit número n, el rango de la representación es

$$-2^{N-1-b} \leq x \leq 2^{N-1-b} - \frac{1}{2^b}$$



Cuantización

La cuantización es una no linealidad introducida al intentar representar variables continuas mediante valores discretos. En [20] se presenta un análisis de la cuantización en sistemas de control digital y en [39] se presenta un modelo lineal y no lineal para el análisis, en [22] se presenta un modelo generalizado que analiza la cuantización como una incertidumbre. En un sistema de control digital se presentan dos tipos de cuantización la introducida por los conversores A/D y D/A y la cuantización interna realizada por dispositivos de punto fijo cuando efectúan multiplicaciones en punto fijo, debido a que la multiplicación de dos registros de longitud β da como resultado un registro de longitud 2β el cual debe ser reducido tomando los bits mas significativos despreciando los bits menos significativos. Los esquemas de cuantización mas utilizados son:

- **QBM(Quantization Before Multiplication:)** Consiste en reducir el tamaño de los operandos antes de realizar la multiplicación.
- **QAM(Quantization After Multiplication:)** El resultado es reducido luego de la multiplicación

Escalamiento

Consiste en amplificar o atenuar las señales de entrada y salida del controlador para garantizar que no se presenten saturaciones o desbordes, esto se hace necesario para representaciones en punto fijo debido a que el rango numérico es menor al de una representación en punto flotante.

Representación en espacio de estados

Un sistema tiene infinitas representaciones dependiendo del conjunto de variables de estado, es posible pasar de una representación a otra aplicando (2.10).

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} TAT^{-1} & TB \\ \hline CT^{-1} & D \end{array}\right] \tag{2.10}$$

Cuando la implementación es sobre un dispositivo de presición finita existen representaciones más adecuadadas que otras debido a sus propiedades numéricas, a continuación se citan algunas de las formas mas utilizadas.

Forma modal Se distingue porque A es diagonal cuando todos los valores propios son reales o diagonal por bloques cuando hay valores propios complejos o valores propios con multiplicidad mayor a uno, la realización tiene la forma (2.11) donde hay dos valores propios reales α_1 y α_2 y dos complejos conjugados de la forma $\alpha_x \pm \beta_x$.

$$G = \left[\begin{array}{ccccc|ccc} \alpha_1 & & & & & b_{11} & \cdots & b_{m1} \\ & \alpha_2 & & 0 & & b_{12} & \cdots & b_{m2} \\ & & \ddots & & & \vdots & \cdots & \vdots \\ & 0 & & \alpha_x & -\beta_x & b_{1x-1} & \cdots & b_{mx-1} \\ & & & \beta_x & \alpha_x & b_{1x} & \cdots & b_{mx} \\ \hline c_{11} & c_{21} & \cdots & c_{x-11} & c_{x1} & d_{11} & \cdots & d_{m1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{1n} & c_{2n} & \cdots & c_{x-1n} & c_{xn} & d_{1n} & \cdots & d_{mn} \end{array}\right] \tag{2.11}$$

Formas de Luenberger La forma mas conocida es la forma controlable [43] la cual es una extensión de la forma controlable para sistemas *SISO*, sea (A, B, C, D) una realización con (A, B) controlable y la matriz de controlabilidad $\mathcal{C}(A, B) = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ tiene rango completo. Es posible encontrar una matriz de transformación T de vectores independientes calculando (2.12)

$$T = \begin{bmatrix} b_1 & Ab_1 & \cdots & A^{r_1-1}b_1 & b_2 & \cdots & A^{r_2-1}b_2 & \cdots & A^{r_m-1}b_m \end{bmatrix} \tag{2.12}$$

En [44] se proponen algoritmos de búsqueda por fila y columna para encontrar los vectores linealmente independientes. Al aplicar la transformación (2.10) la realización tendrá la forma (2.13).

$$G = \left[\begin{array}{cccccc|ccc} 0 & 0 & \alpha_{11} & \cdots & 0 & \alpha_{x1} & 1 & \cdots & 0 \\ 1 & 0 & \alpha_{12} & \cdots & 0 & \alpha_{x2} & 0 & \cdots & 0 \\ 0 & 1 & \alpha_{13} & \cdots & 0 & \alpha_{x3} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \alpha_{1x} & \cdots & 1 & \alpha_{xx} & 0 & \cdots & 1 \\ \hline c_{11} & c_{21} & c_{31} & \cdots & c_{x-11} & c_{x1} & d_{11} & \cdots & d_{m1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{1n} & c_{2n} & c_{3n} & \cdots & c_{x-1n} & c_{xn} & d_{1n} & \cdots & d_{mn} \end{array}\right] \tag{2.13}$$

En [43] se propone una forma análoga a la forma observable la cual se obtiene a partir de la matriz de transformación (2.12), sea $T = [t_1, \cdots, t_n]^T$ donde t_i son las filas de la matriz se calcula la matriz de transformación (2.14) obteniendose la forma canónica (2.15).

$$S = \begin{bmatrix} t_1 & t_1A & t_1A^2 & \cdots & t_1A^n & t_2 & t_2A \cdots & t_nA^n \end{bmatrix}^T \tag{2.14}$$

$$G = \left[\begin{array}{cccccc|ccc} 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1x-1} & \alpha_{1x} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \alpha_{x1} & \alpha_{x2} & \alpha_{x3} & \cdots & \alpha_{xx-1} & \alpha_{xx} & 0 & \cdots & 1 \\ \hline c_{11} & c_{21} & c_{31} & \cdots & c_{x-11} & c_{x1} & d_{11} & \cdots & d_{m1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{1n} & c_{2n} & c_{3n} & \cdots & c_{x-1n} & c_{xn} & d_{1n} & \cdots & d_{mn} \end{array}\right] \tag{2.15}$$

Realizaciones Balanceadas Las formas balanceadas presentan un especial interés a la hora de la implementación debido a la baja sensibilidad cuando son implementadas en dispositivos de presición finita por otra parte las formas canónicas controlables y observables para sistemas multivariables descritas en [43], poseen una considerable cantidad de ceros que a la final reduce el número de operaciones, sin embargo estas formas canónicas presentan una alta sensitividad a la variación de parámetros la forma modal es menos sensitiva aunque presenta desventaja cuando el sistema presenta polos imaginarios y la matriz de transición de estado no es completamente diagonal. A continuación se describe un procedimiento para el cálculo de formas balanceadas.

Sea un sistema $K = [A, B, C, D]$, W_c su gramiano de controlabilidad y W_o su gramiano de observabilidad además A es estable con sus valores propios con parte real negativa existe una solución a las ecuaciones (2.16)y(2.17) si y solo si K es controlable y observable.

$$AW_c + W_cA' = -BB' \tag{2.16}$$

$$A'W_o + W_oA = -C'C \tag{2.17}$$

Y existe una transformación T tal que $W_c = W_o = \Sigma$, el Gramiano de observabilidad y controlabilidad son iguales y está dada por.

$$W_c = \Sigma = \Sigma^{-1/2}U'(R')^{-1}W_cR^{-1}U\Sigma^{1/2} \tag{2.18}$$

$$W_o = \Sigma = \Sigma^{1/2}U'RW_oR'U\Sigma^{-1/2} \tag{2.19}$$

Una característica de las formas balanceadas es que los coeficientes de una realización balanceada generalmente son diferentes de cero lo cual implica un aumento en el número de operaciones.

Operador

Una realización se evalúa de forma iterativa usando como variables intermedias los estados del sistema, el operador se refiere a la relación entre el estado presente y futuro. Una representación en espacio de estados *(A,B,C,D)* como la presentada en la ecuación (2.2) donde $x[n+1] = Ax[n] + Bu[n]$ se puede expresar de la forma $q[X[n]] = Ax[n] + Bu[n]$ donde (q) es el operador memoria y esta definido como $q[x[n]] = x[n+1]$.

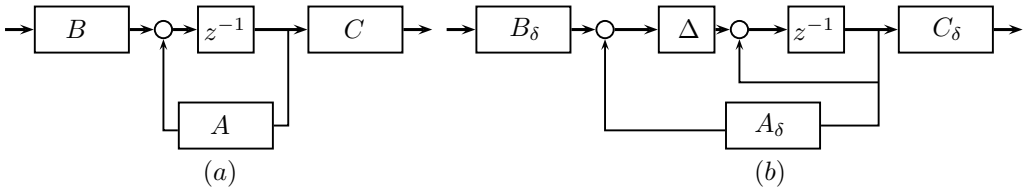


Figura 2.9: (a) Estructura con el operador memoria (b) Estructura δ usando como equivalente el operador q

Operador delta δ Una representación en espacio de estados basada en el operador δ tiene la forma 2.20

$$\begin{array}{lcl} \delta[x[n+1]] & = & A_\delta x[n] + B_\delta u[n] \\ y[n] & = & C_\delta x[n] + D_\delta u[n] \end{array} \tag{2.20}$$

El operador guarda relación con el operador q siendo $\delta = \frac{q-1}{\Delta}$, donde Δ es una constante que usualmente es el periodo de muestreo. Con lo cual se puede encontrar un equivalente entre las dos realizaciones siendo $A = \Delta A_\delta + I$, $B = \Delta B_\delta$, $C = C_\delta$ y $D = D_\delta$. Evaluar esta realización implica el cálculo de una variable intermedia x_i, como se observa en el algoritmo (1).

algoritmo 1 Cálculo de la forma δ

1: $x_i \leftarrow A_\delta x[n] + B_\delta u$
2: $x[n+1] = x[n] + \Delta x_i$
3: $y[k] \leftarrow Cx[n] + Du[n]$

Conclusiones

- Al adicionar integradores en el lazo de control se garantiza la reducción del acople entre canales pero se compromete la estabilidad de la realización lo cual puede traer problemas por los posibles desbordes numéricos; se debe garantizar que las implementaciones digitales sean estables o por lo menos implementar técnicas de saturación numérica o de detección de saturación en actuadores para prevenir el crecimiento de las variables de estado.
- Las formas canónicas se caracterizan por tener un considerable número de coeficientes cero disminuyendo el número de operaciones necesarias para evaluar la realización, esto trae como consecuencia que la realización se hace mas sensitiva a la variación de coeficientes.
- La forma δ es una realización no mínima que puede resultar de ayuda en implementaciones donde no hay restricción en el tiempo de procesamiento y se cuenta con frecuencias de muestreo elevadas, en esta realización el valor de los coeficientes y de los estados tiende a ser grande de forma que los decimales se pueden hacer despreciables.

Capítulo 3

Software

3.1. Descripción 17
3.2. Herramientas de Diseño 18
3.2.1. desacople 18
3.2.2. multivlqg 19
3.2.3. dhinfts 23
3.3. Herramientas para acondicionamiento numérico de realizaciones 24
3.3.1. addaw 24
3.3.2. getdelta 27
3.3.3. nob (Número de bits fraccionales) 28
3.3.4. Funciones Auxiliares 30
3.4. Herramienta *Python* para descarga de realizaciones 32
3.4.1. Protocolo para descarga de realizaciones 32
3.4.2. Funciones *Python* 33

3.1. Descripción

El sistema propuesto se compone de un conjunto de herramientas hardware y software que en conjunto facilitan el diseño y síntesis de controladores. En la figura (3.1) se presenta la metodología propuesta para el diseño y implementación de un sistema de control, se parte de un modelo del sistema a controlar y se inicia con una fase de diseño cuyo objetivo es obtener una realización interpretable por el hardware, sigue una etapa de implementación en la cual se programa el dispositivo permitiendo que el hardware interactue con la planta, en la etapa de supervisión se hace una validación realizando una adquisición de datos y verificando las especificaciones de diseño

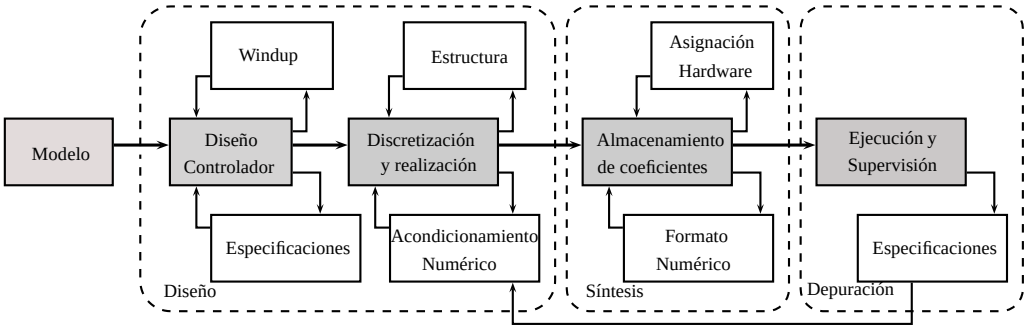


Figura 3.1: Fases en la implementación de un sistema

3.2. Herramientas de Diseño

Las funciones diseñadas hacen uso del *Control System Toolbox*, *Robust Control Toolbox* y *Symbolic Toolbox* las funciones asisten el diseño de controladores requiriendo un mínimo de parámetros.

Nombre	Sintaxis	Descripción
desacople	W=desacople(G)	Encuentra una red de desacople que diagonaliza la planta
lqgcontroller	*D=lqg (G,q,r)*	Calcula un controlador LQG bansándose en pesos Q , R establecidos por el usuario
dhinfts	K=dhinf(G,Bw)	Diseña un controlador H_∞ especificando el tiempo de establecimiento de cada canal.

3.2.1. desacople

Esta función evalua de forma algebraica una red de desacople que al actuar como precompensador diagonaliza la planta.

Sintaxis `W=desacople(G)` Donde G es la planta y W es la red de desacople.

Algoritmo

1. Si la planta es estable y no tiene ceros de transmisión se expresa de la forma $G = D^{-1}N$, donde D es diagonal y es común denominador por fila, la red de desacople se obtiene haciendo $W = N^{-1}$.
2. Si la planta es estable con ceros *RHP* $z = [z_1, z_2, \cdots, z_n]$ se hace $W = \left(I \prod_{i=0}^{n} (s + z_i)\right) G^{-1}$, con lo cual se busca evitar las cancelaciones en el semiplano derecho.

Ejemplo 1 planta minifase: Se asume la planta

$$G = \begin{bmatrix} \frac{6}{s+7} & \frac{5}{s+14} \\ \frac{3}{s+7} & -\frac{0{,}25}{s+4} \end{bmatrix}$$

Haciendo *W=desacople(G)* se obtiene

$$W = \begin{bmatrix} \frac{0{,}015152}{s+4{,}909} & \frac{0{,}18182(s+4)}{(s+7)(s+4{,}909)} \\ \frac{0{,}30303}{(s+4{,}909)} & \frac{-0{,}36364(s+14)}{(s+7)(s+4{,}909)} \end{bmatrix}$$

La respuesta al paso de la planta en cascada con la red de desacople se muestra en la figura (3.2a).

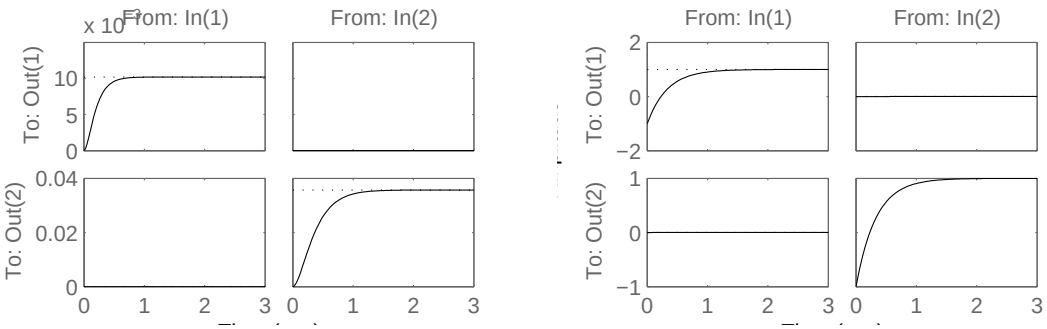
Ejemplo 2 planta con ceros RHP: Se asume la planta

$$G = \begin{bmatrix} \frac{s+1}{s^2+s+1} & \frac{1}{s+1} \\ \frac{s-2}{s+3} & \frac{1}{s+4} \end{bmatrix}$$

En este caso la planta tiene un cero *RHP* en $s = 3{,}0541$. Haciendo *W=desacople(G)* se obtiene.

$$W = \begin{bmatrix} \frac{(s+1)(s+3)(s^2+s+1)}{(s+3{,}896)(s+3{,}054)(s^2+1{,}158s+0{,}9244)} & \frac{-(s+1)(s-2)(s+4)(s^2+s+1)}{(s+3{,}896)(s+3{,}054)(s^2+1{,}158s+0{,}9244)} \\ \frac{-(s+3)(s+4)(s^2+s+1)}{(s+3{,}896)(s+3{,}054)(s^2+1{,}158s+0{,}9244)} & \frac{(s+1)^2(s+3)(s+4)}{(s+3{,}896)(s+3{,}054)(s^2+1{,}158s+0{,}9244)} \end{bmatrix}$$

La respuesta al paso de la planta en cascada con la red de desacople se muestra en la figura (3.2b).



(a) (b)

Figura 3.2: Respuesta al paso de plantas precompensadas con redes de desacople (a) para el caso de una planta estable y de fase mínima (b) para una planta con ceros *RHP*

Restricciones No aplicable a plantas inestables

3.2.2. multivlqg

Esta función diseña controladores *LQG*.

Sintaxis `[K,T]=multivlqg(sys,Q,R,tipo,plot)`
Donde los parámetros son:

sys	*Planta*, puede ser cualquier objeto ss tf o zpk.
q	*Peso asociado al error:* define la velocidad de respuesta del sistema, si se ingresa un escalar se aplica el mismo para cada canal, también puede ser ingresado como un vector de pesos.
r	*Peso asociado a la señal de control:* define la robustez del sistema si se ingresa un escalar se aplica el mismo peso para cada canal, también puede ser ingresado como un vector de pesos.
Tipo	*Estructura del controlador* puede ser 'cen' para un controlador de un grado de libertad o 'c2p' para un controlador de dos parámentros
plot	al ingresar algun valor se grafica la respuesta en lazo cerrado.
K	Controlador
T	Función de lazo cerrado.

Descripción Asumiendo que la dinámica de la planta es lineal y conocida y que la medición de ruido y perturbaciones es estocástica con propiedades estadísticas conocidas, se tiene un modelo de la planta

$$\dot{x} = Ax + bu + w_d \tag{3.1}$$
$$y = Cx + w_n \tag{3.2}$$

Donde w_d y w_n son el ruido del proceso y el ruido de medición y cumplen

$$E\left\{w_d(t)w_d(\tau)^T\right\} = W\delta(t - tau) \qquad E\left\{w_n(t)w_n(\tau)^T\right\} = V\delta(t - tau)$$

Esto implica una densidad espectral constante dada por las matrices W y V. El problema consiste en encontrar un controlador que minimize la función

$$J = E\left\{\lim_{T\to\infty} \frac{1}{T}\int_0^T \left[x^T Q x + u^T R u\right] dt\right\} \tag{3.3}$$

Donde Q es una matriz que pondera los estados y R es una matriz que pondera la entrada, en [45] se encuentra la solución al problema la cual se basa en el principio de separación que consiste en encontrar un LQR asumiendo $wd, wn = 0$ que es conocido como *problema del regulador* y tiene una

solución de la forma $u(t) = -K_r x(t)$ con $K_r = R^{-1}B^T P$, siendo P una matriz simétrica semidefinida positiva y es solución de la ecuación de Riccati.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

Luego se diseña un filtro de *Kalman* el cual es independiente de Q y R, y tiene como objetivo estimar el valor de $x(t)$ minimizando $E\left\{[x - \tilde{x}][x - \tilde{x}]^T\right\}$ siendo $\tilde{x}(t)$ el estado estimado, este es un observador con una estructura

$$\dot{\tilde{x}} = A\tilde{x} + Bu + K_f(y - C\tilde{x})$$

El valor de K_f esta dado por $K_f = LC^T V^{-1}$ donde Y es una matriz simétrica semidefinida positiva y es solución de la ecuación de Riccati

$$LA^T + AL - LC^T V^{-1} CL + W = 0$$

finálmente la solución es $u(t) = K_r\tilde{x}(t)$

Algoritmo

1. Se agregan integradores a la planta incluyendo el signo se la realimentación negativa haciendo $G_i(s) = \left[\begin{array}{cc|c} \text{A} & 0 & \text{B} \\ \text{-C} & 0 & \text{-D} \\ \hline 0 & \text{I} & 0 \end{array}\right]$ donde $A_i = \left[\begin{array}{cc} \text{A} & 0 \\ \text{-C} & 0 \end{array}\right]$ y $B_i = \left[\begin{array}{c} \text{B} \\ \text{-D} \end{array}\right]$

2. Se busca una matriz P semidefinida positiva que sea solución a la ecuación $A_i^T P + PA_i - PB_i R^{-1} B_i^T P + Q = 0$ se calcula el vector de realimentación de estado $K_r = R^{-1}B^T P$, que está conformado por $K_r = \begin{bmatrix} K_{re} & K_{ri} \end{bmatrix}$ siendo K_{re} la ganancia de realimentación debida a los estados de la planta y K_{ri} debida a los estados del integrador.

3. Se asume que el sistema tiene un ruido w_d el cual cumple $E\left\{w_d(t)w_d(\tau)^T\right\} = W\delta(t - tau)$, este modelo de ruido se adiciona a la matriz B siendo $B_n = \begin{bmatrix} B & w_d I_n \end{bmatrix}$

4. Se busca una matriz L semidefinida positiva que sea solución a la ecuación $A^T L + L^T A - L^T B_n R^{-1} B_n^T L + Q = 0$, la ganancia del observador se calcula siendo $K_f = LC^T V^{-1}$, está ganancia se puede dividir en dos $K_f = \begin{bmatrix} K_{fe}, K_{fn} \end{bmatrix}$ una parte debida a las entradas propias del sistema y la otra debida a las entradas de ruido.

5. Se conforma un controlador central , se manejan dos estructuras, la estructura de *1DoF* (un grado de libertad) (fig. 2.1) en la cual el cálculo del error no es realizado por el controlador, en este caso la representacion es:

$$K_{1Dof} = \left[\begin{array}{cc|c} 0 & 0 & \text{-I} \\ -BK_{ri} & A - BK_{re} - K_{fn}C & K_{fn} \\ \hline K_{ri} & K_{rp} & 0 \end{array}\right]$$

   La estructura de dos grados (fig. 3.3b) de libertad resulta en un controlador de la forma $u = K_{2Dof}\begin{bmatrix} r & y \end{bmatrix}^T$ donde el controlador tiene como entradas las referencias y las salidas del sistema, el cálculo del error es interno. En este caso el controlador central tiene la forma.

$$K_{2Dof} = \left[\begin{array}{cc|cc} 0 & 0 & I & \text{-I} \\ -BK_{ri} & A - BK_{re} - K_{fn}C & 0 & K_{fn} \\ \hline K_{ri} & K_{rp} & 0 & 0 \end{array}\right]$$

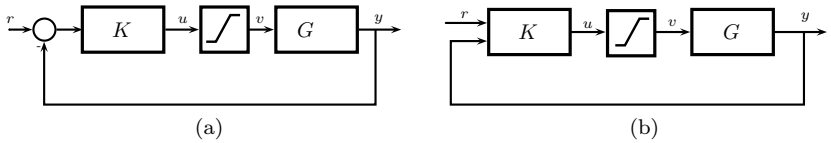


Figura 3.3: Sistema de control con saturación en el actuador (a) *1DoF* (b) *2Dof*

Ejemplo 1: caso $p = q$ Se asume la planta (3.4), la cual representa a un sistema 2×2 con un cero de transmisión en $s = 3{,}05$. Se diseñan tres controladores variando los pesos Q y R. El código con el cual se diseña se presenta en la tabla 3.1y la respuesta resultante se presenta en la figura (3.4).

$$G = \begin{bmatrix} \frac{s+1}{s^2+s+1} & \frac{1}{s+1} \\ \frac{s-2}{s+3} & \frac{1}{s+4} \end{bmatrix} \tag{3.4}$$

En la figura (3.4) se observa que al aumentar el valor de Q el sistema se hace más rápido pero es más notorio el efecto de los ceros de fase no mínima, por otra parte si se aumenta el valor de R es equivalente a hacer que Q sea pequeño; en este caso los pesos utilizados son escalares lo cual indica que se aplicó el mismo peso para cada canal, es posible especificar un peso para cada canal como se muestra a continuación.

Tabla 3.1: Comandos en Matlab para diseño LQG con pesos escalares

```
s=tf( s );
G=[(s+1)/(s^2+s+1) 1/(s+1); (s-2)/(s+3) 1/(s+4)];
[K,T1]=multivlqg(G,10,1); %Q=10 y R=1
[K,T2]=multivlqg(G,1,1);%Q= 1 y R=1
[K,T3]=multivlqg(G,1,10); %Q= 1 y R=10
paso2d(T1,[3 20 35],[1 -1]);
paso2d(T2,[3 20 35],[1 -1]);
paso2d(T3,[3 20 35],[1 -1]);
```

La respuesta de la figura (3.5) se obtuvo modificando el peso Q y dejando el peso R constante, en la parte izquierda se eligió $P = \left(\begin{smallmatrix} 0{,}1 & 0 \\ 0 & 10 \end{smallmatrix}\right)$ de modo que se da mas importancia a la velocidad del canal dos, mientras que en la parte izquierda se eligió $P = \left(\begin{smallmatrix} 10 & 0 \\ 0 & 0{,}1 \end{smallmatrix}\right)$ dando más velocidad el código se muestra en la tabla (3.2).

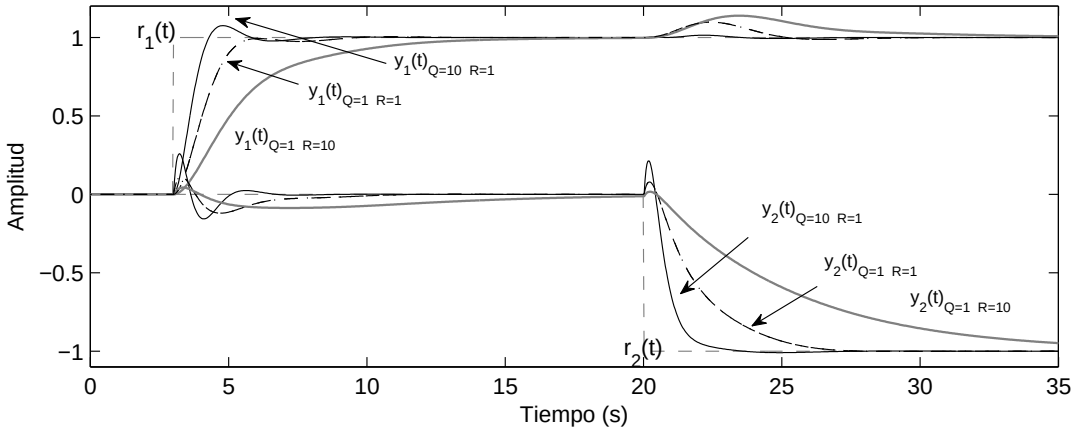


Figura 3.4: Comportamiento del sistema con varios controladores LQG basados en pesos escalares

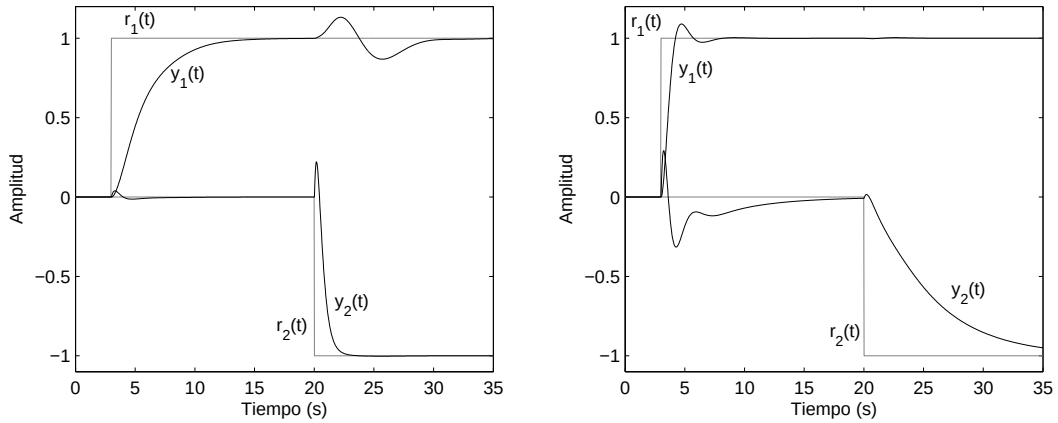


Figura 3.5: Comportamiento del sistema con control LQG con pesos matriciales,

Tabla 3.2: Comandos en Matlab para diseño LQG con pesos matriciales

```
s=tf( s );
G=[(s+1)/(s^2+s+1) 1/(s+1); (s-2)/(s+3) 1/(s+4)];
[K,T1]=multivlqg(G,[0.1 10],[1 1]); %Q=[0.1 0 ; 0 10]
subplot(121); paso2d(T1,[3 20 35],[1 -1]);
[K,T1]=multivlqg(G,[10 0.1],[1 1]); %Q=[ 10 0; 0 0.1]
subplot(122); paso2d(T1,[3 20 35],[1 -1]);
```

Ejemplo 2: Planta rectangular caso $p < q$ Se asume una planta con dos entradas y una salida dada por la ecuación 3.5, esta planta tiene un cero de transmisión en $s = 1$.

$$G = \begin{bmatrix} \frac{s-1}{s^2+s+1} & -2\frac{s-1}{(s+5)^2} \end{bmatrix} \tag{3.5}$$

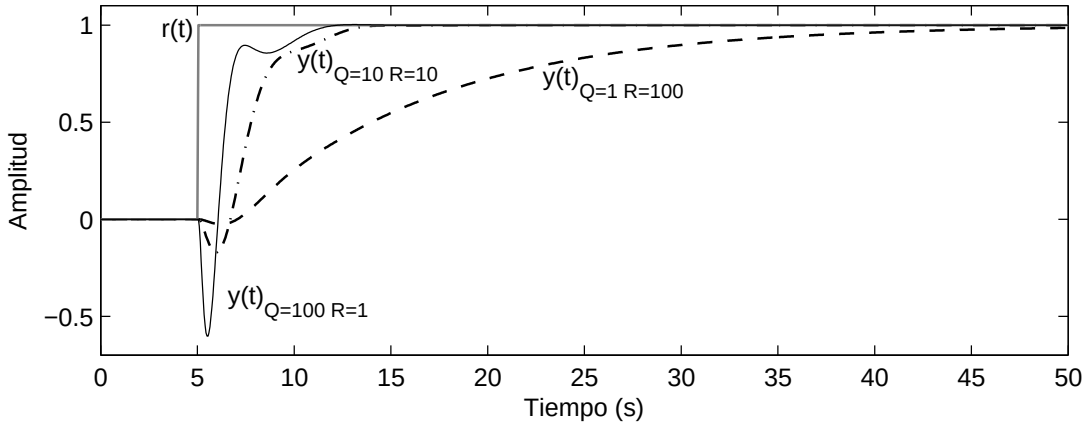


Figura 3.6: Respuesta en lazo cerrado con un controlador LQG de dimensión 2×1

En este caso el controlador debe ser de dimensión 2×1 de modo que el sistema de lazo cerrado es un sistema *SISO*. Se diseña un controlador aplicando diferentes pesos utilizando el código mostrado en la tabla. Se obtiene una respuesta en lazo cerrado con error de posición nulo (fig 3.6), se observa que cuando se aumenta el valor de R se reduce el tiempo de establecimiento pero se disminuye el sub-pico debido al cero de transmisión.

Restricciones Esta función es aplicable a plantas de dimensión $p \times q$ con $p \leq q$. El algoritmo evalua un controlador para el caso $p > q$ quedando $q - p$ salidas no controladas.

3.2.3. dhinfts

Diseña un controlador H_∞especificando el tiempo de establecimiento que se espera para cada canal.

Sintaxis `[K,T]=dhinfts(G,t)`
Donde K es el controlador T es la función de lazo cerrado, G es la planta y t es un vector de la forma $t = \begin{bmatrix} t_1 & t_2 & \cdots & t_n \end{bmatrix}$ siendo t_i el tiempo de establecimiento del canal i.

Tabla 3.3: Comandos en Matlab para diseño H_∞con restricción en el tiempo de establecimiento

```
s=tf( s );
G=[(s+1)/(s^2+s+1) 1/(s+1); (s-2)/(s+3) 1/(s+4)];
[K1,T1]=dhinfts(G,[0.5 2]);
[K2,T2]=dhinfts(G,[1 1]);
[K3,T3]=dhinfts(G,[2 0.5]);
subplot(311);paso2d(T1,[1 5 11],[1 -1]);
subplot(312);paso2d(T2,[1 5 11],[1 -1]);
subplot(313);paso2d(T3,[1 5 11],[1 -1]);
```

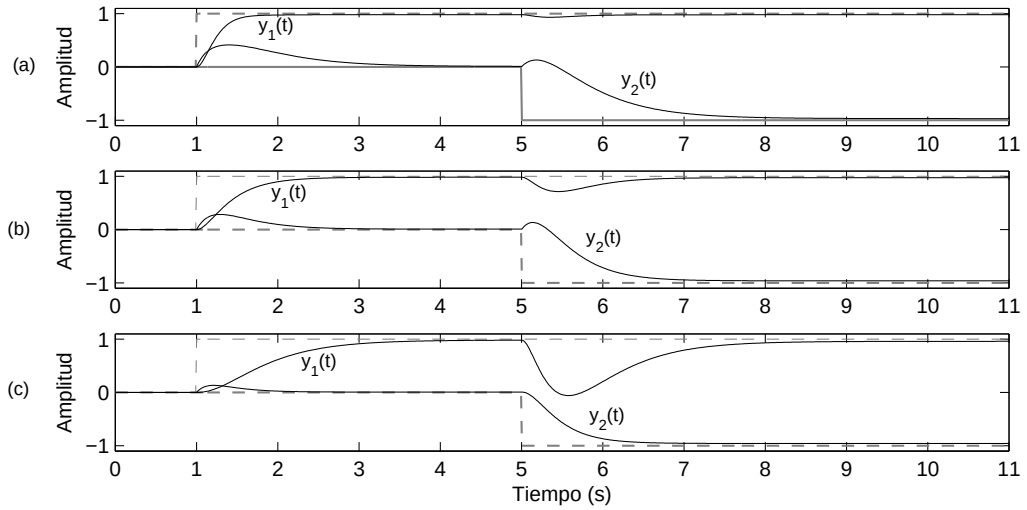


Figura 3.7: Respuesta en lazo cerrado con controlador H_∞(a) para $t = [0{,}5\ 2]$ (b) para $t = [1\ 1]$ (c) para $t = [2\ 0{,}5]$

Descripción La función se basa en una síntesis $[WS]$, donde el peso W se diseña para describir la función de sensibilidad de forma que se puede especificar el ancho de banda, se busca que

$$\overline{\sigma}\left[S(j\omega)\right] \leq \gamma \underline{\sigma}\left[W^{-1}(j\omega)\right]$$

Al ser $S + T = I$, S y T tienen el mismo ancho de banda. Cuando W es escalar el tiempo de establecimiento es aproximadamente el mismo para cada canal, cuando se desea velocidades diferentes W es una matriz diagonal con los respectivos pesos para cada canal.

Ejemplo Dada la planta (3.4) se diseñan tres controladores H_∞especificando como tiempo de establecimiento $t_1 = [0{,}5\ 2]$, $t_2 = [1\ 1]$ y $t_3 = [2\ 0{,}5]$, esto se realiza con el código mostrado en la tabla 3.3.

En la figura (3.7) se observa la respuesta al paso para diferentes especificaciones de diseño.

3.3. Herramientas para acondicionamiento numérico de realizaciones

El conjunto de funciones descritas a continuación, se utilizan para acondicionar el controlador mejorando el desempeño en lazo cerrado.

Nombre	Sintaxis	Descripción
addaw	*Kaw=addaw(sys,chanels,n)*	Sintetiza un controlador central con antiwindup
getdelta	Kd=getdelta(K)	Lleva un sistema *LTI* a una realización δ
nob	ne=nob(K)	Evalua el número de bits enteros para una implementación en punto fijo

3.3.1. addaw

Esta función sintetiza un controlador con *AW (AntiWindup)*.

Sintaxis `Kaw=addaw(K,chanels,n)`
Donde K es un controlador , *chanels* son los canales sujetos a saturación y n es un criterio que establece la ubicación de los polos del observador basandose en los polos de K.

Descripción Es necesario tener en cuenta la estructura del controlador para los casos de uno y dos parámetros, a continuación se analiza cada caso todo el análisis se hace en términos de la variable σ ya que el procedimiento es válido tanto para sistemas continuos y discretos.

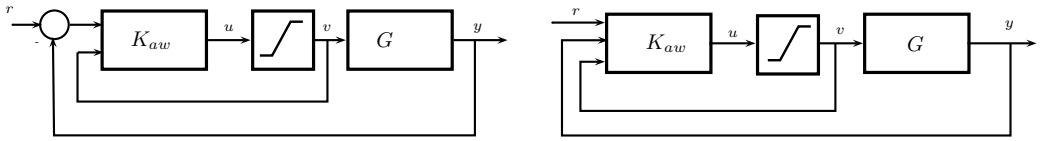


Figura 3.8: Controlador *AW* (a) para la configuración *1DoF* (b) para la configuración *2Dof*

Controlador 1DoF Se asume un sistema como el mostrado en la figura (3.3a) donde K es un controlador central de un parámetro para una planta G de n entradas m salidas, $(m \times n)$, el controlador será de dimensión $n \times m$, con r un vector de referencias de la forma $r = \begin{bmatrix} r_1 & r_2 & \cdots & r_m \end{bmatrix}^T$, y u un vector de señales de control de la forma $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}^T$, y es un vector de salidas del sistema de la forma $y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}^T$, se define un vector de señales de error $e = r - y = \begin{bmatrix} r_1 - y_1 & \cdots & r_m - y_m \end{bmatrix} = \begin{bmatrix} e_1 & \cdots & e_m \end{bmatrix}$. El controlador se puede escribir en la forma de *Smith McMillan* donde $D(k)$ es el polinómio característico del sistema.

$$K = \frac{1}{D_k(\sigma)} N_k(\sigma)$$

Que es equivalente a tener

$$K = \frac{1}{D_k(\sigma)} \begin{bmatrix} N_{11}(\sigma) & N_{12}(\sigma) & \cdots & N_{1m}(\sigma) \\ N_{21}(\sigma) & N_{22}(\sigma) & \cdots & N_{2m}(\sigma) \\ \vdots & \vdots & \ddots & \vdots \\ N_{n1}(\sigma) & N_{n2}(\sigma) & \cdots & N_{nm}(\sigma) \end{bmatrix}$$

Donde N_{nm} es el numerador de la función de transferencia que relaciona la *n-ésima* señal de control con la *m-ésima* señal de error, esto es.

$$u_{nm} = \frac{N_{nm}(\sigma)}{D_k(\sigma)} e_m$$

Cada fila de la matriz de transferencia del controlador se puede expresar de la forma

$$D_k(\sigma)u_n = N_{n1}(\sigma)e_1 + N_{n2}(\sigma)e_2 + \cdots + N_{nm}(\sigma)e_m$$

Si se adiciona el término $A_w(\sigma)u_n$ a cada lado de la expresión siendo A_w un polinómio mónico de igual grado que $D_k(\sigma)$, el cual actua como polinómio observador, siendo necesario que tenga polos más rápidos que el controlador se llega a la expresión.

$$D_k(\sigma)u_n + A_w(\sigma)u_n = N_{n1}(\sigma)e_1 + N_{n2}(\sigma)e_2 + \cdots + N_{nm}(\sigma)e_m + A_w(\sigma)u_n \tag{3.6}$$

Que al ser organizada y reemplazando $v = sat(u)$ resulta

$$u_n = \frac{N_{n1}(\sigma)}{A_w(\sigma)}e_1 + \frac{N_{n2}(\sigma)}{A_w(\sigma)}e_2 + \cdots + \frac{N_{nm}(\sigma)}{A_w(\sigma)}e_m + \left(1 - \frac{D_k(\sigma)}{A_w(\sigma)}\right) v_n \tag{3.7}$$

Finalmente este procedimiento se aplica para cada una de las salidas del controlador resultando en un controlador con n entradas adicionales (ver figura 3.8a) debidas a las entradas saturadas, la forma del controlador es.

$$K_{aw} = \frac{1}{A_w(\sigma)} \begin{bmatrix} N_k & diag\left[A_w(\sigma) - D_k(\sigma)\right] \end{bmatrix}$$

K_{aw} relaciona las entradas y salidas del controlador de la forma (3.10).

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \frac{1}{A_w} \begin{bmatrix} N_{11} & N_{12} & \cdots & N_{1m} & A_w - D_k & 0 & \cdots & 0 \\ N_{21} & N_{22} & \cdots & N_{2m} & 0 & A_w - D_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ N_{n1} & N_{n2} & \cdots & N_{nm} & 0 & 0 & \cdots & A_w - D_w \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_m \\ v_1 \\ \vdots \\ v_n \end{bmatrix} \tag{3.8}$$

Controlador de 2DoF Se asume un sistema de control como el mostrado en la figura (3.3b) donde G es de dimensión $m \times n$, y el controlador K es de dimensión $n \times 2m$, el cual puede ser representado de la forma $K = \begin{bmatrix} K_r & K_y \end{bmatrix}$ asumiendo que es posible fraccionar el controlador en dos, K_r que depende del vector de referencias r y K_y que depende del vector de salidas del sistema y, es equivalente a tener la representación

$$K = \frac{1}{D_k} \begin{bmatrix} N_{r(11)} & \cdots & N_{r(1m)} & N_{y(11)} & \cdots & N_{r(1m)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ N_{r(11)} & \cdots & N_{r(nm)} & N_{y(n1)} & \cdots & N_{r(nm)} \end{bmatrix} \tag{3.9}$$

Aplicando el procedimiento desarrollado anteriormente es fácil demostrar que el controlador con AW para esta configuración es de la forma (3.10) donde las dimesión del controlador es $n \times (2m + n)$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \frac{1}{A_w} \begin{bmatrix} N_{r(11)} & \cdots & N_{r(1m)} & N_{y(11)} & \cdots & N_{y(1m)} & A_w - D_k & 0 & \cdots & 0 \\ N_{r(21)} & \cdots & N_{r(2m)} & N_{y(21)} & \cdots & N_{y(2m)} & 0 & A_w - D_k & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ N_{r(n1)} & \cdots & N_{r(nm)} & N_{y(n1)} & \cdots & N_{y(1m)} & 0 & 0 & \cdots & A_w - D_k \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_m \\ y_1 \\ \vdots \\ y_m \\ v_1 \\ \vdots \\ v_n \end{bmatrix} \tag{3.10}$$

El controlador con AW se puede escribir de forma simplificada

$$K_{aw} = \begin{bmatrix} \frac{N_r}{A_w} & \frac{N_y}{A_w} & \frac{A_w - D_w}{A_w} \end{bmatrix}$$

Con una estructura como la mostrada en la figura (3.8b)

Ejemplo

Para una planta multivariable de la forma

$$G = \frac{0{,}2}{(s+1)(s+2)} \begin{bmatrix} s+2 & 4 \\ -2 & s-2 \end{bmatrix}$$

Se diseña un controlador LQG con efecto integral, el controlador se sintetiza añadiendo una estrategia AW de forma que se pretende evitar que las variables de estado asociadas a los integradores se desborden numéricamente cuando el actuador presente saturación. El código es mostrado en la tabla 3.4.

Tabla 3.4: Diseño LQG con antiwidup

```
s=tf( s );
G=0.2/((s+1)*(s+2))*[(s+2) 4; -2 -(2-s)];
% se dise¬na un controlador muy rapido que precisa se¬nales de control de gran amplitud
[K,T]=multivlqg(G,[100 200],1, cen , a );
% controlador con antiwindup en los canales [1,2] y polos 5 veces mas rapidos que la planta
Kaw=addaw(K,[1,2],5);
```

El sistema en lazo cerrado se simuló usando *Simulink* debido a que es necesario incluir la saturación en el actuador (fig. 3.9).

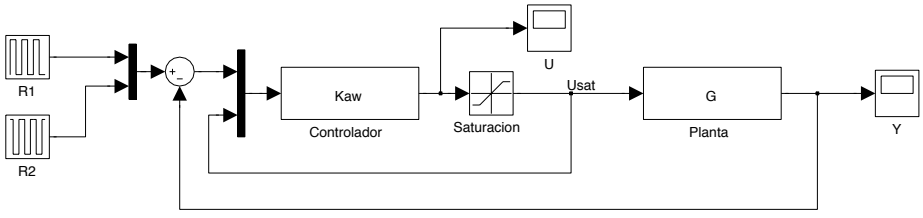


Figura 3.9: Controlador con AW implementado en Simulink

Se hizo un análisis de la respuesta del sistema realimentado para tres casos (fig.3.10), primero se asume un actuador ideal observandose un seguimiento de la señales de referencia, en el segundo caso se asume una saturación en el actuador en el rango $[-1\ 1]$ sin AW, siendo las salidas del sistema $y_{1\ sat}$ y $y_{2\ sat}$, se observa un deterioro en la respuesta debido a que la cantidad de energía suministrada por

el actuador no es suficiente para llevar la planta al punto de operación deseado, en cuanto a las señales de control $u_{1\ sat}$ y $u_{2\ sat}$ se observa un crecimiento desmesurado debido a la acumulación de error, este crecimiento tiene como resultado que en $t = 20seg$ cuando $r_1(t)$ cambia el actuador se encuentre tan saturado que demora un tiempo considerable para entrar de nuevo en región lineal, igual sucede en $t = 30s$ con el cambio de $r_2(t)$, en ambos casos se observa que $y_{1\ sat}$ y $y_{2\ sat}$ mantienen un error constante. Por último se hizo el análisis usando la estrategia AW, se observa que las señales de control $u_{1\ aw}$ y $u_{2\ aw}$ se encuentran en región lineal, por otra parte se observa que en principio la respuesta de $y_{1\ aw}$ y y_{2_aw} es similar a $y_{1\ sat}$ y $y_{2\ sat}$ respectivamente, pero la gran diferencia se aprecia a partir de $t = 20s$ debido a que al no estar saturado el actuador el controlador puede responder más rápidamente ante los cambios en la referencia, siendo importante el hecho que la estrategia AW no permitió el crecimiento de las variables internas del controlador evitando un posible desborde numérico.

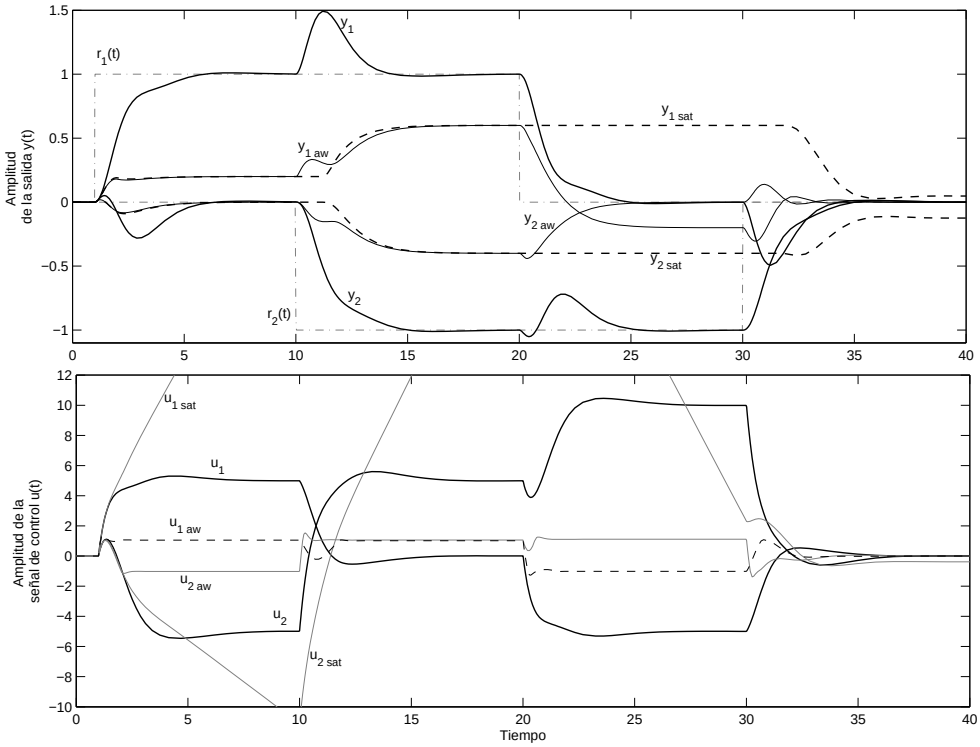


Figura 3.10: Salida y Señal de control del sistema en lazo cerrado (1) con actuador ideal, (2) con actuador limitado en rago (3) con actuador limitado y controlador con AW

Restricciones No se garantiza estabilidad en lazo cerrado, en especial si la planta es inestable.

3.3.2. getdelta

Esta función toma una representación en espacio de estados en términos del operador q y la transforma en una realización en términos del operador δ

sintaxis `Kd=getdelta(K,D)`
Siendo K un sistema en términos del operador q y K_d un sistema equivalente en términos del operador δ, se puede especificar el parámetro $D = \Delta$ en caso de no especificase se elige $\Delta = T_s$ (periodo de muestreo).

descripción Se evalúa la forma δ para una realización $[A, B, C, D]$ en términos del operador q siendo

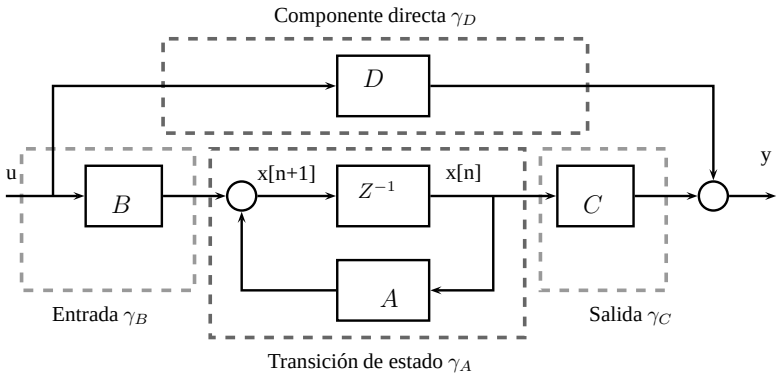


Figura 3.11: Esquema propuesto para evaluar γ

$$\left[\begin{array}{c|c} A_\delta & B_\delta \\ \hline C_\delta & D_\delta \end{array}\right] = \left[\begin{array}{c|c} \Delta^{-1}(A-I) & \Delta^{-1}B \\ \hline C & D \end{array}\right]$$

Restricciones Aplicable solo a sistemas discretos, en sistemas continuos no tiene sentido.

3.3.3. nob (Número de bits fraccionales)

```
nf=nob(sys,lp)
```

Esta función es acrónimo de *(number of bits)*, y retorna el número de bits fraccionales necesarios para implementar un sistema en punto fijo. Este procedimiento es aplicable únicamente a sistemas estables.

Descripción Evaluar una realización implica realizar un cálculo recursivo de estados y salidas, cuando se emplean representaciones en punto fijo en formato $A(\alpha, \beta)$ siendo α la longitud de la parte entera y β la longitud de la parte fraccional, el mayor deterioro que puede sufrir una realización se presenta cuando ocurre desborde numérico debido a que alguna operación matemática arrojó un número con parte entera mayor al valor representable que para este caso es 2^α. Para evitar este fenómeno se debe garantizar que cualquier operación evaluada por el procesador sea representable con el número de bits disponibles.

Se definirá γ como el máximo valor registrado al evaluar una realización, de forma que el rango dinámico de la realización corresponde a $\begin{bmatrix} u_{min}\gamma & u_{max}\gamma \end{bmatrix}$ siendo u_{max} y u_{min} las entradas máxima y mínima del sistema de forma que γ se encuentra normalizado para entradas en el rango $[-1\ 1]$.

En la figura (3.11) se presenta el esquema para evaluar el valor de γ siendo

- γ_B el máximo número para evaluar Bu cuando $u \in [-1\ 1]$
- γ_A el máximo valor al evaluar $x[n+1] = Ax[n]$ asumiento estados normalizados en el rango $[-1\ 1]$.
- γ_C el máximo valor al evaluar la salida del sistema $y = Cx$ asumiento estados normalizados en el rango $[-1\ 1]$.
- γ_D el máximo valor al evaluar la trayectoria directa $y = Du$ asumiento entradas normalizadas en el rango $[-1\ 1]$

Se propone que

$$\gamma = \text{máx}\left[\gamma_A,\ \gamma_B,\ \gamma_C,\ \gamma_D, \gamma_B\gamma_A\gamma_C + \gamma_D\right]$$

Para la verificación se asumirá el sistema (3.11)

$$H = \frac{1}{z^3 - 0{,}7z^2 + 0{,}1676z - 0{,}0338}\begin{bmatrix} z^3 + 45{,}3z^2 - 17{,}35z - 1{,}468 & -46z^2 + 34{,}16z - 5{,}145 \\ 27z^2 + 0{,}6z - 3{,}133 & z^3 - 36{,}7z^2 + 22{,}01z - 0{,}6482 \end{bmatrix}_{T_s=0{,}1} \tag{3.11}$$

Con representación en espacio de estados

$$\left[\begin{array}{ccc|cc} 0.5 & 0 & 0 & 6 & 2 \\ 0 & 0.1 & 0.24 & 4 & 3 \\ 0 & -0.24 & 0.1 & 5 & -6 \\ \hline 1 & 0 & 8 & 1 & 0 \\ 3 & -4 & 5 & 0 & 1 \end{array}\right]$$

Se debe tener en cuenta algunos conceptos relacionados con la norma de matrices, sea B la matriz de entrada, La norma de $\|B\|$ se define como el mayor valor singular $\bar{\sigma}$ o máxima ganancia.

La descomposición en valores singulares de B es:

$$B = usv^T = \begin{bmatrix} -0{,}6190 & -0{,}4580 & -0{,}6380 \\ -0{,}3748 & -0{,}5416 & 0{,}7525 \\ -0{,}6902 & 0{,}7049 & 0{,}1636 \end{bmatrix} \begin{bmatrix} 8{,}8449 & 0 \\ 0 & 6{,}9115 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0{,}9796 & -0{,}2011 \\ 0{,}2011 & -0{,}9796 \end{bmatrix}^T$$

Que indica que esta matriz tiene una ganancia máxima de 8.84, si se observa para el caso en que las entradas al sistema son $u_1 = 1$ y $u_2 = -1$ se tiene

$$Bu = \begin{bmatrix} 6 & 2 \\ 4 & 3 \\ 5 & -6 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 6-2 \\ 4-3 \\ 5+6 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 11 \end{bmatrix}$$

Lo cual indica que $\|B\|$ no es un buen indicador de γ_B debido a que la ganancia máxima se evalua para entradas ortonormales. La norma infinito es $\|B\|_\infty = 11$ que corresponde al máximo obtenido a partir de esto se puede deducir que.

$$\boxed{\|B\|_\infty \geq \gamma_B \text{ para } u \in [-1\ 1].}$$

Demostración: a partir de la definición de la norma H_∞que establece que para una matriz $B \in \mathbb{R}^{m\times n}$

$$\|B\|_\infty = max\left(\sum_{i=1}^{n} |B_i|\right)$$

Para cualquier canal i se cumple

$$\text{máx}\, B_i < B_{i,1}u_1 + B_{i,2}u_2 + \cdots < |B_{i1}|\, u_1 + |B_{i,2}|\, u_2 \cdots < \gamma_B$$

De forma se garantiza que para una entrada normalizada en el rango $[-1\ 1]$ la norma infinito será mayor o igual que el valor absoluto de la suma de canales con sus respectivas ganancias.

De igual forma

$$\gamma_B = \|B\|_\infty \qquad \gamma_C = \|C\|_\infty \qquad \gamma_C = \|C\|_\infty$$

Una vez definido el máximo de la entrada al sistema se analiza el cálculo del siguiente estado, el máximo valor implicado en el calculo se puede evaluar como

$$\boxed{\gamma_A = \gamma_B(1 + \|A\|_\infty)\left\|(zI - A)^{-1}\right\|_\infty}$$

Demostración: Al garantizar que el sistema es estable se debe cumplir que para cualquier estado inicial la respuesta dinámica de $x[n+1] = Ax[n]$ tiene un comportamiento decreciente. Teniendo como máxima ganancia $\left\|(zI - A)^{-1}\right\|_\infty$.

De forma que luego de calcular el siguiente estado la peor condición se presenta cuando la dirección de entrada hace que Bu tenga la máxima amplificación para más de dos periodos de muestreo esto es $Bu = \gamma_B$ y simultaneamente el bloque $(zI - A)^{-1}$ presenta la máxima ganancia coincidiendo con la dirección en que Ax es máxima.

De esta forma se llega a la expresión propuesta anteriormente

$$\gamma = \text{máx}\left[\gamma_A,\ \gamma_B,\ \gamma_C,\ \gamma_D, \gamma_B\gamma_A\gamma_C + \gamma_D\right]$$

Se evalúa el número de bits enteros mínimo como

$$n_e = \overline{int}\left[\log_2\left[\text{máx}\left[|u_{max}|, |u_{u_{min}}|\right]\gamma\right]\right]$$

Donde $\overline{int}[x]$ es una función que aproxima el número x al entero inmediatamente superior. El número de bits fraccionales corresponderá a la longitud de palabra menos los bits enteros menos el bit de signo $n_f =$ longitud de palabra $- n_e - 1$.

Ejemplo Se realizará un análisis en el cual se caculará el número de bits enteros necesarios para implementar una realización con cada una de las representaciones :

- **Realización balanceada (Balanceo del gramiano):** Corresponde a la función `balreal` de Matlab la cual trata de balancear los gramianos de observabilidd y controlabilidad.
- **Realización balanceada (Balanceo de estado):** Corresponde a la función `ssbal` la cual busca que la norma de filas y columnas sea aproximadamente igual.
- **Forma modal:** Corresponde a la función `canon` la cual lleva la matriz de transición de estado a la forma diagonal de Jordan.
- **Forma Observable:** Para sistemas *SISO* se puede obtener haciendo $G = ss2ss(G, obsv(G))$ que consiste en aplicar como transformación la matriz de observabilidad, en sistemas *MIMO* se puede emplear la función `sobserver` para obtener la forma observable de *Luenberger*
- **Forma Controlable:** Para sistemas *SISO* se puede obtener haciendo `G=ss2ss(G,inv(ctrb(G)))` , que consiste en aplicar como matriz de transformación la matriz de controlabilidad del sistema, para el caso *MIMO* se puede usar la función `scontroller` para obtener la forma observable de *Luenberger*.

El análisis se realizó para dos funciones de transferencia una *SISO* (eq.3.12) y una *MIMO* (eq.3.11) obteniendose los resultados mostrados en la tabla (3.5).

$$H = \frac{5}{z^6 - 0{,}202z^5 + 1{,}37z^4 - 0{,}1451z^3 + 0{,}3951z^2 + 0{,}0147z - 0{,}04739} \tag{3.12}$$

	Balanceada Gramiano	Balanceada norma	Modal	Observable	Controlable
Sistema SISO	12	12	19	16	13
Sistema MIMO	16	18	17	17	14

Tabla 3.5: Número de bits enteros mínimos para implementación en punto fijo

De esto se puede deducir que para implementaciones en punto fijo las realizaciones balanceadas por gramiano presentan el mejor condicionamiento numérico ya que al requerir menos bits enteros es posible asignar mayor número de bits a la parte fraccional mejorando la resolución.

3.3.4. Funciones Auxiliares

Peso

`W=peso(w)`

Retorna una función de transferencia de primer orden que puede ser empleada como peso para la función se sensitividad en el diseño H_∞. La función tiene la forma

$$W = \frac{s/M + \omega}{s + \omega A} \qquad M = 1{,}2 \;\; A = 0{,}01M$$

El único parámetro de entrada corresponde a la frecuencia central del peso.

to2dof

`K2p=to2dofs(K)`

Esta función lleva un controlador de un grado de libertad (1Dof) a una configuración de dos grados de libertad *(2Dof)*.

Asumiendo que la entrada al controlador de un grado de libertad es una señal de error $e = r - y$ se tiene la siguiente representación.

$$\begin{array}{rcl} \dot{x} & = & Ax + B(r-y) \\ u & = & Cx + D(r-y) \end{array} \quad = \quad \left[\begin{array}{c|cc} \text{A} & \text{B} & \text{-B} \\ \hline \text{C} & \text{D} & \text{-D} \end{array}\right]$$

findint

`chanel=findint(G)`

Dado un sistema G retorna un vector con el identificador de los canales que poseen integrador.

evaldelta

`y=evaldelta(K,u)`

Dado un sistema K en tiempo discreto y expresado en términos del operador δ y dado u un vector que corresponde a la entrada al sistema se calcula la salida del sistema y evaluando la realización de forma recursiva.

scontroller

`[Hc,T]=scontroller(H)`

Esta función lleva el sistema H a la forma controlable de *Luengberger* retornando el sistema en Hc en forma controlable y la matriz de transformación T, se encuentra basada en el algoritmo descrito en [43].

Ejemplo Para el sistema (3.11) la forma controlable se evalúa haciendo `[Hc,T]=scontroller(H)` obteniendose

$$Hc = \left[\begin{array}{cccccc|cc} 0 & 0 & 0.0338 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & -0.1676 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0338 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -0.1676 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.7 & 0 & 0 \\ \hline 46 & 14.6824 & 1.1339 & -46 & 1.96 & 3.9366 & 1 & 0 \\ 27 & 19.5 & 5.9918 & -36 & -3.3576 & 3.0689 & 0 & 1 \end{array}\right]$$

En esta representación se puede observar una diagonal de unos y dos columnas de coeficientes, debido a que el sistema es 2×2 resulta muy similar a dos bloques de realizaciones controlables.

sobserver

`[Ho,T]=observer(H)`

Esta función lleva el sistema H a la forma observable de *Luengberger* retornando el sistema en Ho en forma observable y la matriz de transformación T, se encuentra basada en el algoritmo descrito en [43].

Ejemplo Para el sistema 3.11 la forma observable es

$$Ho = \left[\begin{array}{cccccc|cc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0338 & -0.1676 & 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.0338 & -0.1676 & 0.7 & 0 & 0 \\ \hline 46 & -270.5266 & 434.3905 & -46 & 75.8757 & 57.9882 & 1 & 0 \\ 27 & -226.5740 & 576.9231 & -36 & 160.3314 & -99.3373 & 0 & 1 \end{array}\right]$$

En este caso se conserva la diagonal de unos y dos filas de coeficientes resultando similar a dos bloques de realizaciones observables.

3.4. Herramienta *Python* para descarga de realizaciones

Python es un lenguaje interpretado libre y multiplataforma. Se describe un conjunto de funciones desarrolladas en lenguaje *Python* las cuales pueden ser empleadas para descargar realizaciones al dispositivo hardware.

Se programó la función `guardar.m` para Matlab la cual crea un archivo de texto con toda la información relacionada con el controlador el formato de este archivo se puede observar en el anexo 1. Este archivo de texto puede ser modificado en cualquier editor de texto.

3.4.1. Protocolo para descarga de realizaciones

Para descargar una realización al dispositivo hardware no se es necesario poseer un protocolo específico debido a que descargar la realización consiste en actualizar la memoria del procesador, lo único que se debe tener en cuenta es el borrar previamente el bloque de memoria en que se va a escribir (En el capitulo 4 se describe el mapa de memoria del dispositivo).

Software				**Hardware**
Solicitud de transmisión	`0x7F3F1F0F`	→		
		←	`0x11`	Palabra de confirmación
Borrar Página	`0x23452345`	→		
Dirección a borrar	`0x2800`	→		
		←	`0xEE`	Palabra de confirmación
Solictud escritura	`0x15151515`	→		
número de bits	`0x08`	→		
dirección	`0x2800`	→		
enviar datos	`0x00101010`	→		
		⋮		
		←	`0xEE`	Palabra de confirmación
Solictud lectura	`0x7CC77CC7`	→		
número de bits	`0x08`	→		
dirección	`0x2800`	→		
		←	`0x00101010`	Datos
		⋮		
Finalizar Tx	`0xABCDABCD`	→		

Tabla 3.6: Ejemplo de protocolo para comunicación con el dispositivo hardware

En la tabla 3.6 se muestra un posible protocolo el cual se basa en tres tareas principales que son

Función	**Sintaxis**	**Descripción**
iniciar	`p=iniciar(port)`	Esta función inicializa un puerto serial y retorna su manipulador
finalizar	`finalizar(p)`	Libera el puerto con manipulador p
txfloat	`txfloat(valor,p)`	Envia `valor` a través del puerto p en formato flotante single
txint	`txint(valor,p)`	Envia `valor` a través del puerto p en como un entero con signo de 32 bits
txuint	`txuint(valor,p)`	Envia `valor` a través del puerto p en como un entero sin signo de 32 bits
txshort	`txshort(valor,p)`	Envia `valor` a través del puerto p en como un entero con signo de 16 bits
txushort	`txushort(valor,p)`	Envia `valor` a través del puerto p en como un entero de 16 bits sin signo
rxint	`valor=rxint(p)`	Lee del puerto p un entero de 32 bits
rxfloat	`valor=rxfloat(p)`	Lee del puerto p un flotante single
rxshort	`valor=rxshort(p)`	Lee del puerto p un entero de 16 bits
rxch	`valor=rxch(p)`	Lee del puerto p un caracter (char)
inittx	`ok=inittx(p)`	Inicia comunicación con el dispositivo enviando la solicitud de transmisión
fintx	`ok=fintx(p)`	Envia la palabra de fin de transmisión
borrar	`ok=borrar(p,dir)`	Borra el bloque de memoria especificado por *dir*
writefloat	`ok=writefloat(p,valor,dir)`	Escribe un flotante en la direccion dir
writeint	`ok=writeint(p,valor,dir)`	Escribe un entero de 32 bits en la direccion dir
writeshort	`ok=writeshort(p,valor,dir)`	Escribe un entero de 16 bits en la direccion dir
leerfloat	`valor=leerfloat(p,n,dir)`	Lee n datos tipo flotante apartir de la direccion dir
leerint	`valor=leerint(p,n,dir)`	Lee n datos tipo entero apartir de la direccion dir
leershort	`valor=leershort(p,n,dir)`	Lee n datos tipo short apartir de la direccion dir
leercontrol	`valor=leercontrol(direccion)`	Lee un controlador diseñado en matlab y lo entrega como un diccionario

Tabla 3.7: Funciones Python

- Borrar el bloque de memoria
- Solicitar la escritura especificando dirección y número de bits a escribir.
- Solicitar la lectura del bloque recien escrito para verificar su correcta programación.

3.4.2. Funciones *Python*

Dado que no se precisa un protocolo específico se construyó un conjunto de funciones que realizan tareas de bajo nivel las cuales se resumen en la tabla 3.7 para ejecutar estas funciones basta con instalar python dependiendo del sistema operativo (en linux viene por defecto) e instalar el complemento *pyserial* que permite el manejo del puerto serie.

Conclusiones

- En sistemas estables el cálculo de redes de desacople resulta simple y permite que el diseño de controladores multivariables se convierta en un problema *SISO*.
- La solución del problema de sensibilidad mixta permite especificar mediante pesos características como el tiempo de establecimiento, sobrepico y amplitud de la señal de control, el uso de los

pesos tiene como desventaja el incremento en el orden del controlador, en un controlador LQG los pesos aplicados nunca alteran el orden del controlador resultando en un controlador con el mismo orden de la planta para regulación o planta mas número de entradas cuando se adicionan integradores para seguimiento de referencia.

- El esquema antiwindup propuesto es aplicable únicamente a controladores con integrador en plantas estables. Debido a la función saturación que relaciona la salida del controlador con la entrada, el controlador es no lineal y el análisis de estabilidad no puede realizarse aplicando la teoría de sistemas lineales, se podría emplear el teorema de estabilidad de *Lyapunov*.

- En una implementación de punto fijo la elección del número de bits fraccionales es fundamental, cuando el controlador es estable es posible recurrir a la norma infinito para estimar el valor máximo de las variables de estado de forma que se puede determinar el número de bits enteros mínimos necesarios y dejando los sobrantes como bits fraccionales, cuando el controlador es inestable (en lo posible solo por poseer integradores) el número de bits para la parte entera tiende a infinito siendo obligatorio el uso de una técnica antiwindup.

- Se propuso una método para determinar el formato numérico para implementaciones en punto flotante con operador q.

- Para implementaciones en punto fijo las formas balanceadas resultan ser la mejor alternativa , el estudio comparativo mostró que en general este tipo de realizaciones debido a su condicionamiento numérico pueden ser evaluadas empleando un menor número de bits enteros siendo posible aumentar en número de bits fraccionales mejorando la presición.

- Las formas observable y controlable presentan un significativo número de coeficientes iguales a cero de forma que su tiempo de procesamiento resultaría menor, pero al analizar el número de bits enteros se concluye que restan presición a la implementación.

Capítulo 4

Hardware y Firmware

4.1. Hardware 35
4.1.1. Procesador 36
4.1.2. Módulos del sistema 36
4.2. Firmware 37
4.2.1. Asignación de memoria 37
4.2.2. Programa principal 38
4.2.3. Comunicación 40
4.3. Modelo del sistema 41
4.3.1. Etapas de entrada y salida (Acondicionamiento de señal) 41
4.3.2. Cálculo de realizaciones 42
4.3.3. Ejecución de operaciones 43
4.3.4. Tiempo de procesamiento 45

Los sistemas discretos pueden ser implementados sobre dispositivos como *FPGA (Field Programmable Gate Array)*, *DSP (Digital signal processor)* y microcontroladores. La principal ventaja de las *FPGA* es la elevada velocidad de procesamiento, pero al ser dispositivos de implementación hardware cualquier modificación a una realización implicaría la síntesis del hardware y la descarga al dispositivo, por otra parte se encuentran los microcontroladores y *DSP* que son reconfigurables y de bajo costo, pero con tiempos de procesamiento mayores a los obtenidos con una *FPGA*.

4.1. Hardware

El sistema debe contar con algunas etapas para acondicionamiento de señales analógicas que corresponden a las entradas y salidas del controlador y un protocolo de comunicación para realizar tareas de programación y supervisión.

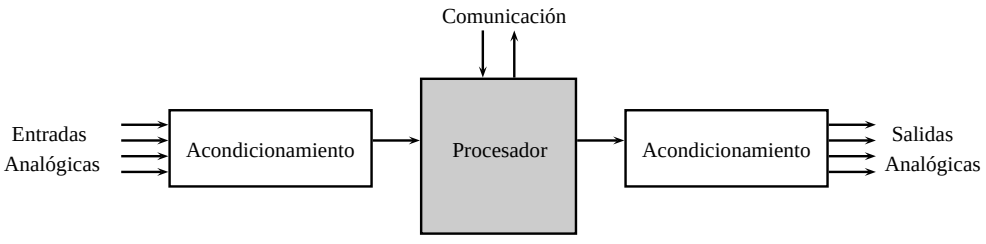


Figura 4.1: Esquema general del sistema de control

4.1.1. Procesador

Se eligió el microcontrolador *ADuC7026* fabricado por *Analog Devices* basado en un procesador *ARM7TDMI* o *Advanced Risc Machine 7 Thumb, Debug, Multiply, ICE* , *Thumb* representa la capacidad del procesador para operar con instrucciones de 16 y 32 bits dependiendo del desempeño deseado, *Debug* se refiere a la capacidad para entrar en modo de depuración mediante la interfaz JTAG (*Joint Test Action Group*). *Multiply* hace referencia a la capacidad para hacer multiplicaciones de 32 bits por 32 bits con resultados de 64 bits y *ICE* hace referencia al módulo integrado *EmbeddedICE* el cual es un soporte para depuración embebido.

Aspectos generales

El dispositivo opera a una frecuencia máxima de 44Mhz con una longitud de palabra de 32 bits y una alimentación de 3V. Dependiendo de la referencia el puede tener hasta 40 puertos de entrada y salida. El dispositivo cuenta con 4 timers, un módulo PWM trifásico, y lo que lo hace ideal para aplicaciones de control es un módulo de 16 canales de A/D de 12 bits que opera a 1MSPS y un modulo de cuatro canales D/A de 12 bits, en cuanto a memoria cuenta con 62kB de memoria Flash y 8KB de Ram. En la figura (4.2) se muestra su diagrama funcional.

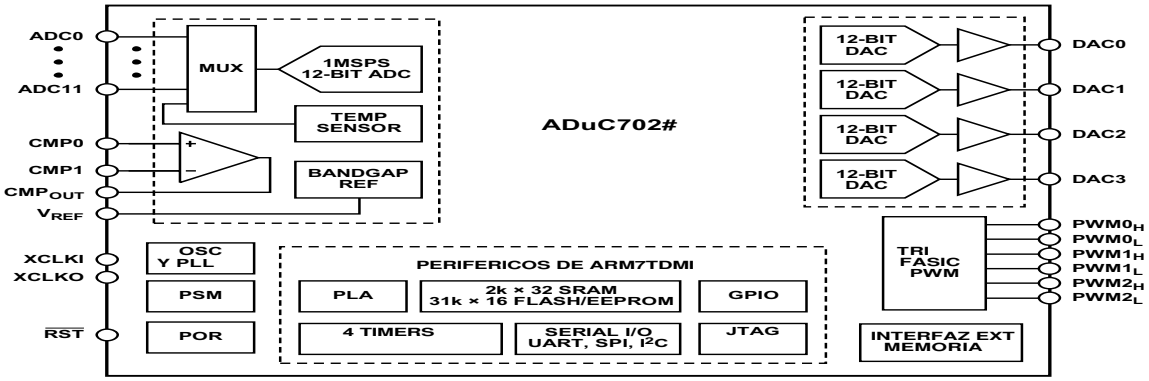


Figura 4.2: Diagrama funcional del microcontrolador ADuC702X

Las carácteristicas que hacen ideal a este dispositivo para esta aplicación es:

- Incorpora 12 de entrada analógica.
- Incorpora 4 canales de salida analógica.
- Opera sobre registros de 32 bits.
- Posee hardware para multiplicar.

4.1.2. Módulos del sistema

Etapa de salida Se diseñó un circuito basado en amplificador operacional que al ingresar una señal en el rango [0 3,3] entrega una salida en el rango [−10 10] el circuito se muestra en la figura (4.3.a).

Etapa de entrada Se diseñó un circuito basado en amplificador operacional que permite entradas en el rango [−10 10] y una salida en el rango [0 3,3] para ingreso directo al microcontrolador, el dispositivo maneja una tierra virtual en $1{,}65V$, el circuito se muestra en la figura (4.3.b).

Referencias en circuito El dispositivo cuenta con cuatro referencias las cuales son potenciómetros ubicados en el circuito y conectados directamente a un canal analógico de entrada.

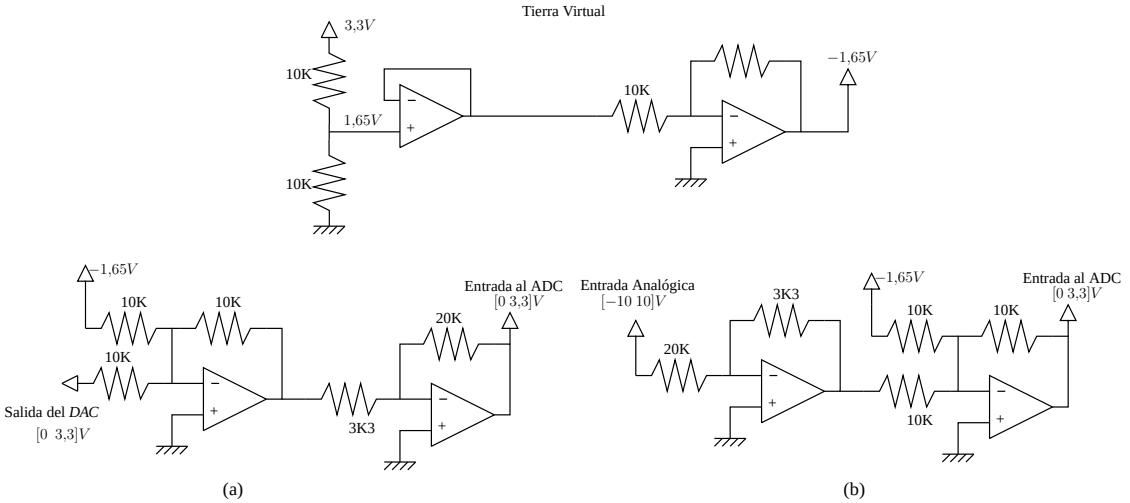


Figura 4.3: Circuitos para adecuación de señal (a) etapa de salida (b) etapa de entrada

Comunicación El dispositivo soporta comunicación *Rs-232* adicionalmente se incorporó un conversor *FTDI* el cual realiza una conversión usb- serial permitiendo que el dispositivo sea reconocido como un puerto COM consultar el anexo *A.1* para ayuda sobre la instalación del driver.

4.2. Firmware

El firmware se programó en lenguaje C, usando la herramienta *Keil* la cual es gratuita y es ofrecida por el fabricante del microcontrolador.

4.2.1. Asignación de memoria

La memoria debe almacenar el programa principal, los coeficientes que componen la realización y la configuración del dispositivo, los bloques de memoria restantes se pueden emplear en el almacenamiento de algunas variables monitoreadas a modo de supervisión.

El dispositivo tiene *64Kb* de memoria flash la cual se encuentra entre los registros $0h00080000$ al $0h0008FFFFh$, y se divide en 32 bloques de *2Kb*, cada bloque está compuesto por cuatro páginas de modo que la memoria se divide en 128 paginas de 512 bits, el direccionamiento se realiza con registros de 16 bits.

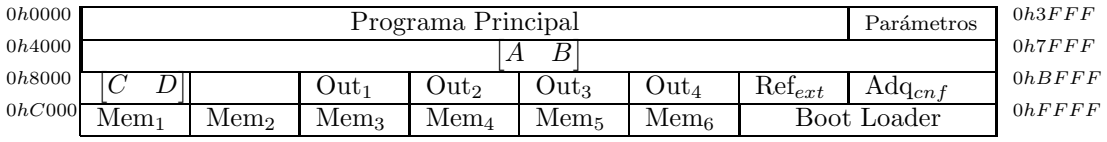


Figura 4.4: Mapa de memoria

Almacenamiento de realizaciones Disponiendo de *16Kb* para almacenar $\begin{bmatrix} A & B \end{bmatrix}$ se asume que esta matriz esta compuesta por $(n_x + n + i) \times n_x$ coeficientes, teniendo en cuenta que cada coeficiente ocupa 4 bits y que el número máximo de entradas es $n_i = 12$, asumiendo cuatro canales de entrada diferenciales con cuatro entradas de antiwindup, el valor máximo de n_x es la solución de la ecuación

$$n_x^2 + 12n_x - \frac{16384}{4} = 0 \quad n_x \approx 58$$

Por limitación en el hardware el número máximo de salidas es $n_o = 4$ entonces la matriz $\begin{bmatrix} C & D \end{bmatrix}$ será de dimesión $(n_x + n_i) \times n_o = 70 \times 4 = 280$ Coeficientes de modo que ocupará $1120Kb$ siendo necesario un bloque de cuatro páginas para su almacenamiento.

El la figura (4.4) se presenta la organización de la memoria por bloques de cuatro páginas. Se destinan *14Kb* al programa principal, *16kb* a almacenar la matriz $\begin{bmatrix} A & B \end{bmatrix}$, *2Kb* para almacenar $\begin{bmatrix} C & D \end{bmatrix}$, adicionalmente en la tabla 4.1 se encuentra la descripción de los bloques de memoria.

Bloque	Parámetro	Significado
Parámetros	n_i	Número de entradas
	n_o	Número de salidas
	n_x	Número de estados
	T_s	Periodo de muestreo
	$Tipo$	Tipo de realización (Punto fijo o punto flotante)
	$delta$	Valor de Δ para implementaciones en forma δ
	$nbpf$	Número de bits fraccionales
	TipoCh	Procedencia del canal de entrada 1 (Ver tipos de entradas)
Out	U_{max}	Excursión máxima
	U_{min}	Excursión mínima

Tabla 4.1: Contenido de los bloques de memoria

Tipos de entradas Dada una realización con n_i entradas cada una de ellas posee un identificador que determina la procedencia de dicha entrada. Por hardware se tiene un total de doce entradas de la cuales ocho son canales analógicos externos y cuatro referencias en circuito basadas en potenciómetros adicionalmete se cuenta con cuatro referencias externas modificables por hardware y cuatro entradas debidas a los canales de antiwindup.

La palabra que determina el tipo de canal ver tab. 4.1 consiste en una palabra de tipo entero la cual se distribuye como se observa en la figura (4.5).

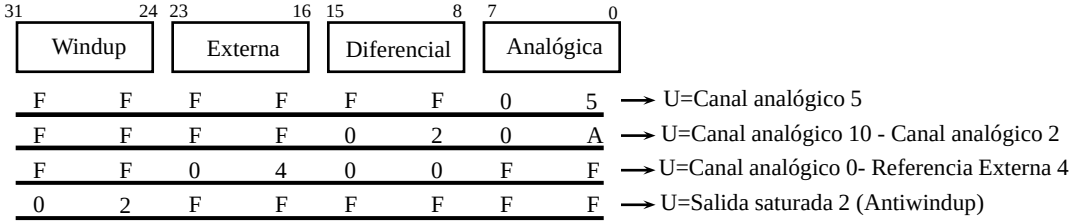


Figura 4.5: Tipos de canales de entrada y ejemplos

Se observa que de la palabra de *32 bit* se asigna un byte para describir el tipo de canal, siendo posible elegir dos canales en modo diferencial para conformar una entrada ver fig.(4.5), en tal caso se evalua la entrada a la realización como la diferencia de los dos canales.

4.2.2. Programa principal

El microcontrolador posee dos modos de operación, el modo ejecución en el cual se evalúa la realización almacenada en memoria y el modo descarga en el cual el dispositivo permite la actualización total o parcial de un controlador la figura (4.6) presenta un diagrama de bloques en el cual se presentan los modos de operación .

Modo ejecución En el modo ejecución se tienen cuatro rutinas principales la evaluación de realizaciones en punto fijo y en punto flotante usando operadores q y δ.

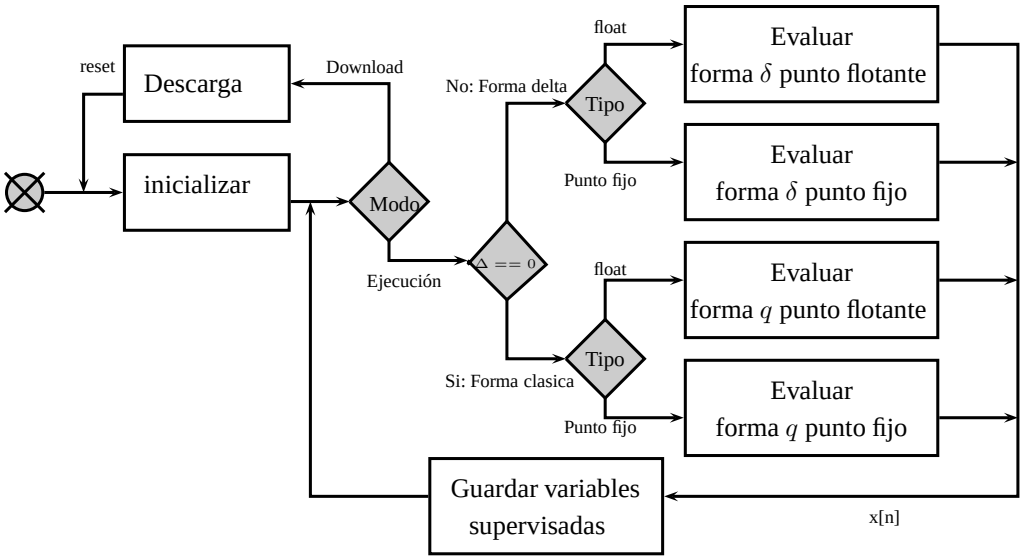


Figura 4.6: Modos de operación programa principal

Realización con operador q El cáculo de la realización es recursivo siendo necesario realimentar el valor de $x[n]$ de un ciclo al otro en el algoritmo (2) se describe la función calcular realización q implementada en el procesador basada en la representación clásica en espacio de estados.

$$x[n+1] = Ax[n] + Bu[n]$$
$$y[n] = Cx[n] + Du[n]$$

algoritmo 2 Cálculo de realización en operador retardo

```
1:  función CalcularRealizaciónq(x)
2:      u ← leercanales
3:      x = [x  u]
4:      Para i ← 1, N_o hacer                          ▷ Cálculo de la salida
5:          Acumulado = 0
6:          Para j ← 1, N_x + N_i hacer
7:              Si C[i, j] ≠ 0 Entonces
8:                  Acumulado = C[i, j]x[j] + Acumulado
9:              Fin Si
10:         Fin Para
11:         y[i] = Sat(Acumulado, u_i max, u_i min)    ▷ Saturación de salida
12:     Fin Para
13:     Para i ← 1, N_x hacer                          ▷ Cálculo del siguiente estado
14:         Acumulado = 0
15:         Para j ← 1, N_x + N_i hacer
16:             Si A[i, j] ≠ 0 Entonces
17:                 Acumulado = A[i, j]x[j] + Acumulado
18:             Fin Si
19:         Fin Para
20:         x[i] = Acumulado
21:     Fin Para
22:     Return x[0 : N_x]                              ▷ se retorna el estado siguiente X[n + 1]
23: Fin función
```

Realización con operador δ La realización delta tiene la forma

$$\begin{aligned} \delta[x[n+1]] &= A_\delta x[n] + B_\delta u[n] \\ y[n] &= C_\delta x[n] + D_\delta u[n] \end{aligned}$$

el cálculo del estado siguiente se hace en dos pasos primero se evalúa la variable intermedia T y luego se evalúa $x = \Delta T + x$, la función descrita en el algoritmo 3 tiene como parámetro de entrada el vector de estados $x[n]$ y Δ.

algoritmo 3 Cálculo de realización en operador δ

```
1:  función CalcularRealizaciónDelta(x,Δ)
2:      u ← leercanales
3:      x = [x  u]
4:      Para i ← 1, N_o hacer                          ▷ Cálculo de la salida
5:          Acumulado = 0
6:          Para j ← 1, N_x + N_i hacer
7:              Si C[i, j] ≠ 0 Entonces
8:                  Acumulado = C[i, j]x[j] + Acumulado
9:              Fin Si
10:         Fin Para
11:         y[i] = Sat(Acumulado, u_i max, u_i min)    ▷ Saturación de salida
12:     Fin Para
13:     Para i ← 1, N_x hacer                          ▷ Cálculo de la variable intermedia
14:         Acumulado = 0
15:         Para j ← 1, N_x + N_i hacer
16:             Si A[i, j] ≠ 0 Entonces
17:                 Acumulado = A[i, j]x[j] + Acumulado
18:             Fin Si
19:         Fin Para
20:         T[i] = Acumulado
21:     Fin Para
22:     Para i ← 1, N_estados hacer                    ▷ Cálculo del siguiente estado
23:         x[i] = ΔT[i] + x[i]
24:     Fin Para
25:     Return x[0 : N_estados]                        ▷ solo X[n + 1]
26: Fin función
```

4.2.3. Comunicación

El dispositivo opera en base a solicitudes hechas por el usuario permitiendo que cualquier aplicación capaz de manejar puertos serie sea capaz de entablar comunicación con el dispositivo. Las solicitudes se pueden clasificar en dos tipos.

- `Solicitud de información:` Se emplea para leer una variable del microcontrolador, la solicitud va acompañada de la dirección a ser leida y el número de datos leidos el dispositivo responde entregando la información solicitada.

- `Solicitud de acción:` Se emplea para dar una orden al procesador ya sea para sincronizar el dispositivo, actualizar un registro o forzar una acción, este responde con la palabra de confirmación ($0hEE$).

Las instrucciones son códigos de 32 bits que realizan una tarea específica, se tiene el siguiente conjunto de instrucciones.

El modo *ping* se utiliza para verificar si el dispositivo se encuentra conectado a un puerto, el dispositivo responde con la palabra de confirmación, el modo ajuste se emplea para verificar los canales analógicos y hace que los canales de salida entregen lo que se esta leyendo en el canal analógico 1.

Códigos para lectura		Códigos para escritura	
$0hF5F5F5F5$	Leer float	$0h1F3F1F3F$	Escribir float
$0h7CC77CC7$	Leer entero de 32 bits	$0h15151515$	Escribir entero de 32 bits
$0h9EE99EE9$	Leer entero de 16 bits	$0h26262626$	Escribir entero de 16 bits
		$0h3A3A3A3A$	Asignar$Ref_{ext}1$
		$0h1C1C1C1C$	Asignar$Ref_{ext}2$
		$0h5D5D5D5D$	Asignar$Ref_{ext}3$
		$0h2B2B2B2B$	Asignar$Ref_{ext}4$
		$0hA12BA12B$	ping
		$0hC5B6C5B6$	Modo ajuste
		$0h7F3F1F0F$	Entrar al modo descarga
		$0hABCDABCD$	Salir del modo descarga

Tabla 4.2: Solicitudes al microcontrolador

Sincronización

Al conectar el dispositivo en un equipo por primera vez el dispositivo es reconocido como un puerto serial y el sistema operativo asigna un identificador al puerto, la aplicación debe detectar el dispositivo inicializando los puertos disponibles y enviando una solicitud, si la solicitud es contestada se sabrá que el hardware se encuentra conectado de lo contrario se debe intentar con otro puerto. Este procedimiento se describe en el diagrama mostrado en la figura (4.7) el identificador del puerto puede variar de un sistema operativo a otro.

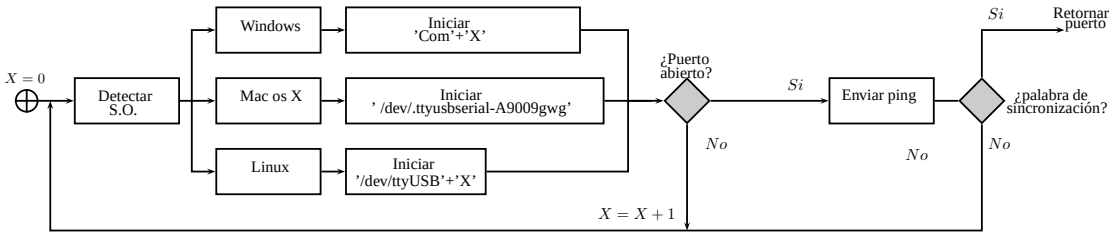


Figura 4.7: Diagrama de flujo detección de puerto y sincronización

4.3. Modelo del sistema

Se construyó un modelo del sistema con el fin de simular y analizar el comportamiento de una realización antes de implementarla en la plataforma. Para construir el modelo se deben tener en cuenta todas no linealidades como truncaciones, saturaciones, cuantización de señales, y aritmética empleada para realizar las operaciones.

4.3.1. Etapas de entrada y salida (Acondicionamiento de señal)

La etapa de acondicionamiento consta de un arreglo de amplificadores operacionales mostrado en la figura (4.3), se asume que la respuesta es lineal y solo se encuentra sujeto a saturación por parte de la alimentación del circuito. El conversor ADC y DAC trabajan con un formato numérico entero de 12 bits, el conversor ADC opera en región lineal para tensiones de entrada de $[0\ 3{,}3]V$ entregando una salida en el rango $[0\ 4095]$ fuera de región lineal se satura a el nivel máximo o mínimo de salida como se observa en la figura (4.8).

El conversor DAC recibe un entero sin signo de 12 bits y lo convierte a una salida analógica en el rango $[0\ 3{,}3]V$, fuera de región lineal presenta un fenómeno de desborde que hace que cantidades con un valor numérico alto sean confundidas con un valor bajo es posible añadir una saturación por software con el fin de prevenir el desborde.

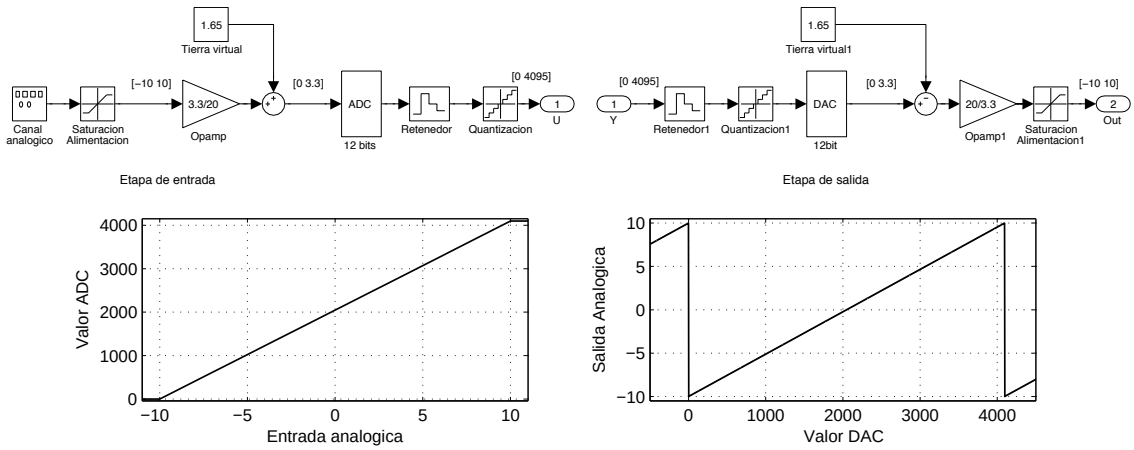


Figura 4.8: Modelo entrada salida característica entrada salida

4.3.2. Cálculo de realizaciones

Debido a que la implementación se realiza sobre un dispositivo de presición finita el modelo debe contemplar la truncación a los coeficientes durante la descarga a la memória y la cuantización. Para simular el comportamiento del dispositivo se emplea el *Fixed Point Toolbox* el cual permite simular dispositivos de precisión finita.

Para definir el sistema de precisión finita basta con definir dos estructuras

- **fi:** Se relaciona con el formato númerico con el cual son almacenados los coeficientes de la realización sus propiedades son:
  - Longitud de palabra: número total de bits necesarios para almacenar un coeficiente.
  - Signo: Habilita el bit mas significativo como bit de signo.
  - Parte fraccional: Número de bits reservados para representar la parte fraccional.
- fimat: Especifica la forma como se realizan las operaciones en el dispositivo de procesamiento.
  - Longitud de palabra de la ALU.
  - Longitud de palabra máxima para el resultado de una suma: Se relaciona con la capacidad para manejar el acarreo luego de una suma.
  - Longitud de palabra máxima para e resultado de una multiplicación: Se relaciona con el esquema de cuantización adoptado para la multiplicación.
  - Manejo del desborde
  - Longitud de la parte fraccional luego de una suma.
  - Longitud de la parte fraccional luego de una multiplicación.

Realizaciones en punto fijo

Para una realización en punto fijo con representación (p, q), el hardware posee las siguientes características :

Longitud de palabra de la ALU	32=p+q+1
Longitud de palabra máxima para el resultado de una suma	32
Longitud de palabra máxima para e resultado de una multiplicación	64
Manejo del desborde	No
Longitud de la parte fraccional luego de una suma	q
Longitud de la parte fraccional luego de una multiplicación	q

4.3.3. Ejecución de operaciones

Para evaluar la ecuación de estado $x[n+1] = Ax[n] + Bu[n]$ $y[n] = Cx[n] + Du[n]$ se lleva a la forma matricial

$$x[n+1] = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix} \qquad y[n] = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix}$$

Dicha operación se realiza haciendo

$$x[n+1] = \begin{bmatrix} \left(\sum_{i=0}^{nx+ni} \begin{bmatrix} A & B \end{bmatrix}_{(i,1)} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix}_i \right) \\ \vdots \\ \left(\sum_{i=0}^{nx+ni} \begin{bmatrix} A & B \end{bmatrix}_{(i,nx)} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix}_i \right) \end{bmatrix} \qquad y[n] = \begin{bmatrix} \left(\sum_{i=0}^{nx+ni} \begin{bmatrix} C & D \end{bmatrix}_{(i,1)} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix}_i \right) \\ \vdots \\ \left(\sum_{i=0}^{nx+ni} \begin{bmatrix} C & D \end{bmatrix}_{(i,no)} \begin{bmatrix} x[n] \\ u[n] \end{bmatrix}_i \right) \end{bmatrix}$$

Número de operaciones

Para una realización $[A, B, C, D]$ con n_i entradas, n_o salidas y n_x estados el número de operaciones depende del operador.

Operador q

Para evaluar el siguiente estado $x[n+1] = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$ se requiere de $n_x(n_x + n_i)$ multiplicaciones y $n_x(nx + n_1 - 1)$ sumas en cuanto a la ecuación de salida $y = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$ se requieren $n_o(n_x + n_i)$ multiplicaciones y $n_o(n_x + n_i - 1)$ sumas. Entonces se requieren:

- Sumas: $(n_x + n_i - 1)(n_x + n_o)$
- Multiplicaciones $(n_x + x_i)(n_x + n_o)$

Se realizará el analisis de adición de estados, entradas y salidas para unificar el tiempo de procesamiento de realizaciones *MIMO*, en términos de realizaciones *SISO*.

Adición de un estados Cuando se adiciona un estado a la realización se tiene que el número de multiplicaciones será

$$N_\times = (n_x+1+n_i)(n_x+1+n_o) = (nx+n_i)(n_x+1+n_o)+(n_x+1+n_o) = (n_x+1)(n_x+n_o)+(nx+n_i)+(n_x+1+n_o)$$

$$N_\times = (n_x+1)(n_x+n_o) + \overbrace{2n_x+n_i+n_o+1}^{\Delta_{x,\times}}$$

En cuanto al número de sumas.

$$N_+ = (n_x + 1 + n_i - 1)(n_x + 1 + n_o) = (nx + n_i - 1)(n_x + 1 + n_o) + (n_x + 1 + n_o)$$

$$= (n_x + n_i - 1)(n_x + n_o) + (nx + 1 + n_o) + (nx + n_i - 1) = (n_x + n_i - 1)(n_x + n_o) + \overbrace{2n_x + n_i + n_o}^{\Delta_{x,+}}$$

Siendo $\Delta_{x,\times} = 2n_x + n_i + n_o + 1$ el incremento en el número de multiplicaciones debido a la adición de un estado y $\Delta_{x,+} = 2n_x + n_i + n_o$ el incremento en el número de sumas debido a la adición de un estado.

Adición de una entrada Cuando se adiciona una entrada al sistema el número de multiplicaciones será

$$N_\times = (n_x + n_i + 1)(n_x + n_o) = (nx + n_i)(n_x + n_o) + \overbrace{(n_x + n_o)}^{\Delta_{i,\times}}$$

donde $\Delta_{i,\times}$ corresponde al incremento en multiplicaciones, en cuanto al número de sumas se tiene

$$N_+ = (n_x + x_i)(n_x + n_o) = (n_x + x_i - 1)(n_x + n_o) + \overbrace{(nx + n_o)}^{\Delta_{i,+}}$$

Se puede afirmar que $\Delta_{x,\times} > \Delta_{i,\times}$ y $\Delta_{x,+} > \Delta_{x,+}$ de forma que la adición de una entrada trae menor carga computacional que la adición de un estado.

Adición de una salida Cuando se adiciona una salida al sistema el número de multiplicaciones será

$$N_x = (n_x + n_i)(n_x + n_o + 1) = (nx + n_i)(n_x + n_o) + \overbrace{(n_x + n_i)}^{\Delta_{o,\times}}$$

donde $(nx + n_i)$ corresponde al incremento en multiplicaciones, en cuanto al número de sumas se tiene

$$N_+ = (n_x + n_i - 1)(n_x + n_o + 1) = (n_x + x_i - 1)(n_x + n_o) + \overbrace{(n_x + n_i - 1)}^{\Delta_{o,+}}$$

Se cumple $\Delta_{x,\times} > \Delta_{i,\times}$ y $\Delta_{x,+} > \Delta_{x,+}$ requiriendo menor tiempo de procesamiento la adición de una salida con respecto a la adición de un estado, con respecto a la adición de una entrada el costo computacional es similar.

Adición de una entrada y una salida Este caso corresponde al aumento de dimensión del controlador, el número de multiplicaciones será.

$$N_\times = (n_x+x_i+1)(n_x+n_o+1) = (nx+n_i)(n_x+1+n_o)+(n_x+1+n_o) = (n_x+1)(n_x+n_o)+(nx+n_i)+(n_x+1+n_o)$$

$$N_\times = (n_x + 1)(n_x + n_o) + \overbrace{2n_x + n_i + n_o + 1}^{\Delta_{io,\times}}$$

En cuanto al número de sumas se tiene

$$N_+ = (n_x + n_i - 1 + 1)(n_x + n_o + 1) = (nx + n_i - 1)(n_x + 1 + n_o) + (n_x + 1 + n_o)$$

$$= (n_x + n_i - 1)(n_x + n_o) + (nx + 1 + n_o) + (nx + n_i - 1) = (n_x + n_i - 1)(n_x + n_o) + \overbrace{2n_x + n_i + n_o}^{\Delta_{io,+}}$$

Se observa que $\Delta_{io,+} = \Delta_{x,+}$ y $\Delta_{io,\times} = \Delta_{x,\times}$ de forma que computacionalmente es equivalente aumentar un estado a aumentar una entrada y una salida.

Operador δ

En una realización con operador delta se requieren $(n_x + n_i)n_x$ multiplicaciones y $(n_x + n_i - 1)n_x$ sumas evaluar la variable intermedia $T = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$, luego se requieren n_x multiplicaciones y n_x sumas para evaluar el siguiente estado $x[n+1] = \Delta T + x[n]$, por último se requieren $(n_x + ni)n_o$ multiplicaciones y $(n_x + ni - 1)n_o$ sumas para evaluar la salida del sistema $y[n] = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$.

De forma que el incremento en el número de operaciones con operador δ corresponde a $\Delta_\times = n_x$ y $\Delta_+ = n_x$ entonces el tiempo de procesamiento de una realización δ de orden n_x es menor que el tiempo de una realización en operador q de orden $n_x + 1$ y mayor que el tiempo de una realización q de orden n_x.

4.3.4. Tiempo de procesamiento

A partir del análisis planteado anteriormente se puede deducir que para una realización de orden n_x con un tiempo de procesamiento t_x y una realización de orden $n_x + n$ con tiempo de procesamiento t_{x+n} se cumple que

- Al adicionar una entrada o salida el tiempo de procesamiento t_p cumple $t_x < t_p < t_{x+1}$
- Al adicionar en conjunto una entrada y una salida el tiempo de procesamiento t_p cumple $t_x < tp = t_{x+1} < tx + 2$ asumiendo conversores D/A y A/D ideales con tiempo de adquisición y entrega de señales nulo.
- Al evaluar una realización de orden n_x en operador δ el tiempo de procesamiento t_p es $t_x < t_p < t_{x+1}$

De esta forma basta con caracterizar el dispositivo para el caso *SISO* siendo posible deducir el tiempo de establecimiento de sistemas *MIMO* para cantidad de entradas o salidas e incluso de realizaciones en forma δ

Según mediciones realizadas empleando como sistema de prueba

$$H = \frac{\prod_{i=0}^{n_x} (1{,}3 + i/2)(s + 1 + i/2)}{\prod_{i=0}^{n_x} (1 + i/2)(s + 1{,}3 + i/2)}$$

Este corresponde a un sistema propio estable con ganancia unitaria se midió y registró el tiempo de procesamiento en punto flotante para varios ordenes estos se presentan en la tabla (4.3), se observó que para realizaciones en punto fijo el tiempo de procesamiento podía presentar variaciones dependiendo del número de bits fraccionales debido a las operaciones de cuantización, para el caso en que el número de bits fraccionales es 31 se presenta el máximo tiempo de procesamiento en punto fijo, obteniendose una reducción del 20 % la cual no se tendrá en cuenta y se asumirá como tiempo de procesamiento el obtenido para realizaciones en punto flotante .

Orden	1	2	3	4	5	6	7	8	9	10
Tiempo(μS)	100	168	248	344	476	620	820	1020	1300	1500
Orden	11	12	13	14	15	16	17	18	19	20
Tiempo(mS)	1.6	1.84	2.1	2.4	2.7	3.1	3.5	3.9	4.3	4.7

Tabla 4.3: Tiempo de procesamiento para realizaciones $SISO$ con operador q

Para que la realización se evalúe correctamente el tiempo de procesamiento debe ser menor al periodo de muestreo, de otra forma el dispositivo sufrirá bloqueos por acumulación de interrupciones .

Por este motivo se hizo una regresión cuadrátrica para relacionar el orden de la realización con respecto al tiempo de procesamiento obteniendose la función.

$$t_{\mu S} = 10n^2 + 35n + 60 \qquad \text{donde n es: } n = n_x + \text{máx}\,[n_i, n_o]\text{y} + 1 \text{ si es realizacion } \delta$$

Esta función permite calcular el tiempo mínimo de procesamiento pero es necesario tener en cuenta que el dispositivo debe realizar algunas otras tareas de modo que se propone que el periodo de muestreo sea un poco mayor al tiempo de procesamiento para garantizar operaciones como la reprogramación del dispositivo o supervisión de señales, en la figura 4.9 se presenta el tiempo de procesamiento mínimo, la región bajo la curva corresponde a una región no implementable existe una región de tolerancia en la cual se garantiza que la realización operará de forma adecuada pero no se garantizan funciones como reprogramación y supervisión de variables, todo lo que se encuentre sobre la región punteada es implementable de forma que para una realización con 50 estados se tiene un tiempo de procesamiento de procesamiento de $t = 22mS$ pero se recomienda que el periodo de muestreo sea $Ts = 40mS$.

Conclusiones

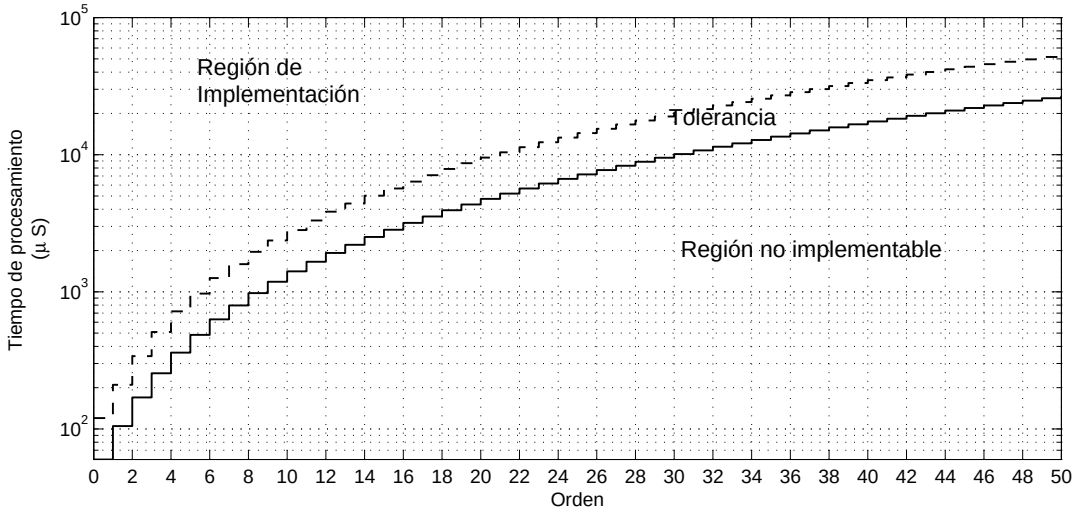


Figura 4.9: Región de implementación

- El tiempo de procesamiento de una realización depende principalmente del número de estados, en sistemas multivariables la adición de un canal (entrada- salida) resulta equivalente a aumentar la realización un estado.
- Se esperaba que para realizaciones en punto flotante el tiempo de procesamiento resultara considerablemente mayor con respecto a las implementaciones en punto fijo, pero las mediciones realizadas muestran que son muy similares, un análisis detallado reveló que la componente dominante del tiempo de procesamiento se encuentra en el direccionamiento, lectura de memoria y adquisición de señales analógicas.
- Debido a que no hay diferencias significativas en el tiempo de procesamiento al cambiar el formato numérico se recomienda el uso de realizaciones en punto flotante y de esta forma evitar problemas de desborde numérico y escalamiento.

Capítulo 5
Interfaz de usuario

5.1. Descripción 47
5.1.1. Estructura y planta 48
5.2. Síntesis 48
5.2.1. Síntesis de K 49
5.2.2. Síntesis de W 50
5.2.3. Síntesis de F 50
5.2.4. Síntesis del controlador central 52
5.3. Implementación 53
5.4. Metodología de diseño 54

5.1. Descripción

La interfaz de usuario es una herramienta desarrollada en *Matlab* y que tiene como finalidad facilitar el uso del sistema de desarrollo multivariable. Las principales caráteristicas de la interfaz son:

- Permite la implementación de controladores y emulación de sistemas multivariables.
- Acepta plantas $(m \times n)$ con $m \leq n$
- En el modo control posee una estructura general la cual esta basada en los bloques W, K, F.
  - **W (Elemento de desacople):** es un bloque mediante el cual se desea que el producto GW sea aproximadamente diagonal, los métodos de diseño son la compensación estática evaluada para una frecuencia y la compensación dinámica para todas las frecuencias.
  - **K (Compensador):** es el controlador el cual garantiza la respuesta de lazo cerrado deseada, el usuario tiene la posibilidad diseñar este bloque usando las estrategias LQG y H_∞.
  - **F (Prefiltro):** es un elemento que acondiciona la señal de referencia para ajustar el comportamiento frecuencial del sistema de lazo cerrado, para el diseño del prefiltro se cuenta con redes de adelanto, redes de atraso y filtros *notch*.
- Se pueden visualizar variables del sistema de lazo cerrado como la sensitividad, sensitividad inversa y señal de control en el dominio del tiempo y frecuencial.
- Interfaz para descarga de controladores en la cual se puede adicionar una estrategia antiwindup a los canales inestables del controlador.
- Implementación de controladores usando operador q y δ, con aritmética de punto fijo o punto flotante.

- Modelo del sistema el cual permite simular teniendo en cuenta todos los elementos no lineales del hardware y firmware.

5.1.1. Estructura y planta

En la figura (5.1.a) se muestra el panel de selección de estructura, este panel se relaciona con el modo de operación del dispositivo hardware el cual puede ser emulación y control. El el modo control se trabaja con una estructura WKF teniendo las siguientes restricciones:

- F siempre es cuadrada y diagonal.
- Por software se satura la señal de control de forma que es posible hacer que K tenga un esquema antiwindup realimentando la señal de control saturada.
- K puede tener forma de *1Dof* o *2Dof* siendo la más general la forma de $2Dof$ con antiwindup.
- W siempre es cuadrada de forma que su uso se limíta a plantas cuadradas.

Cuando el sistema es empleado para la emulación de plantas digitales se omiten los bloques F y W, en este caso se deshabilita la pestaña síntesis.

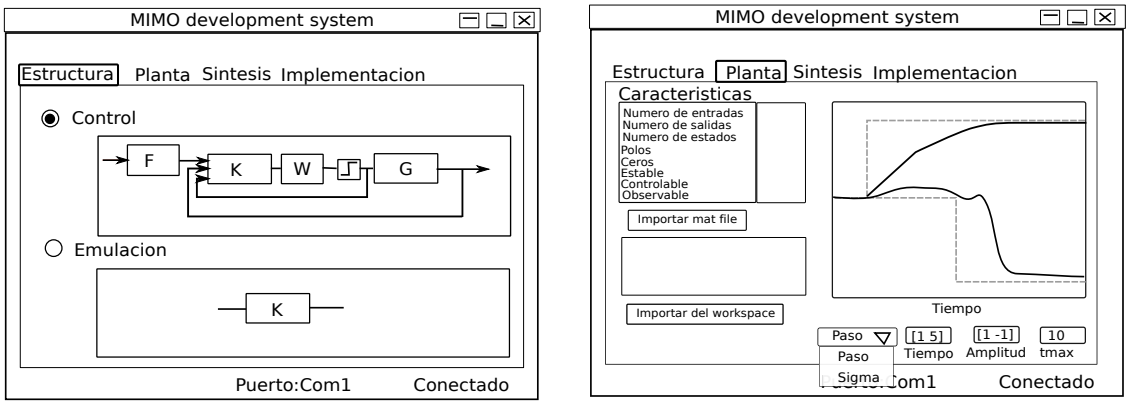


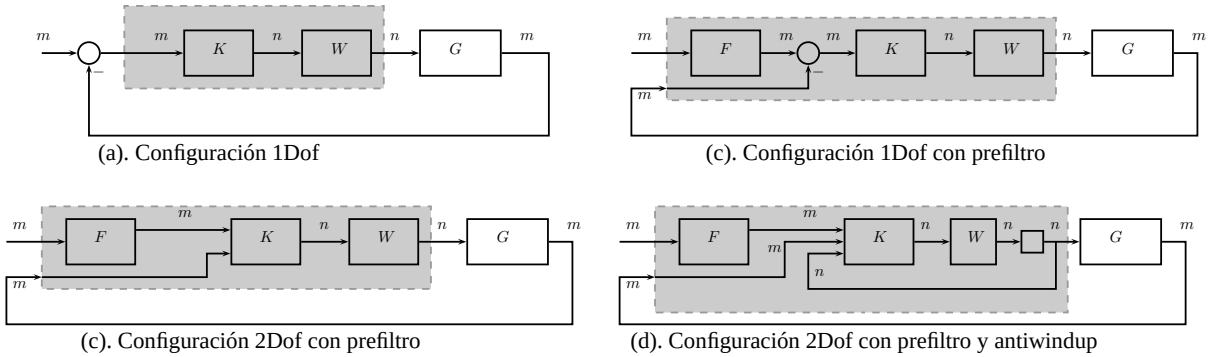
Figura 5.1: Paneles de elección de estructura y definición de la planta

Cuando se elige la estructura de control se debe especificar cual es la planta a controlar, para ello se destina un panel en el cual se puede definir la planta ya sea importando un archivo de extensión *mat* (Formato matlab para almacenamiento de objetos) o tomando el valor de una variable del workspace. Una vez se define la planta (5.1.b) se visualizan características como número de entradas, salidas y estados, polos, ceros, controlabilidad y observabilidad a demás se puede visualizar la respuesta al paso y el gráfico de valores singulares.

5.2. Síntesis

La síntesis consiste en asignar valores a W, K y F y llevarlas a la forma de un controlador centralizado. Las dimensiones de cada bloque dependen directamente de las dimensiones de la planta; para una planta G de dimensión $m \times n$, W tendrá dimensión $n \times n$, K será de dimensión $n \times m$ para una configuración *1Dof*, $n \times 2m$ para una una configuración de dos grados de libertad y $n \times 2m + n$ para una configuración *2Dof* con antiwindup, por último F será de dimensión $m \times m$ y es diagonal ver figura (5.2).

La interfaz cuenta con un panel destinado al diseño del controlador, este panel cuenta con algunos paneles auxiliares los cuales permiten ingresar los parámetros para el diseño. Es posible asignar a cualquier bloque una variable almacenada en un *matfile* o en una variable del espacio de trabajo *(workspace)*, se cuenta con una tabla en la cual se visualiza las caraterísticas de controlador central



(a). Configuración 1Dof

(c). Configuración 1Dof con prefiltro

(c). Configuración 2Dof con prefiltro

(d). Configuración 2Dof con prefiltro y antiwindup

Figura 5.2: Estructuras de control, los subíndices corresponden a la dimensión de la señal y la región sombreada al componente implementado

tales como número de entradas, salidas y estados, dimensión de cada bloque y estabilidad de lazo cerrado.

5.2.1. Síntesis de K

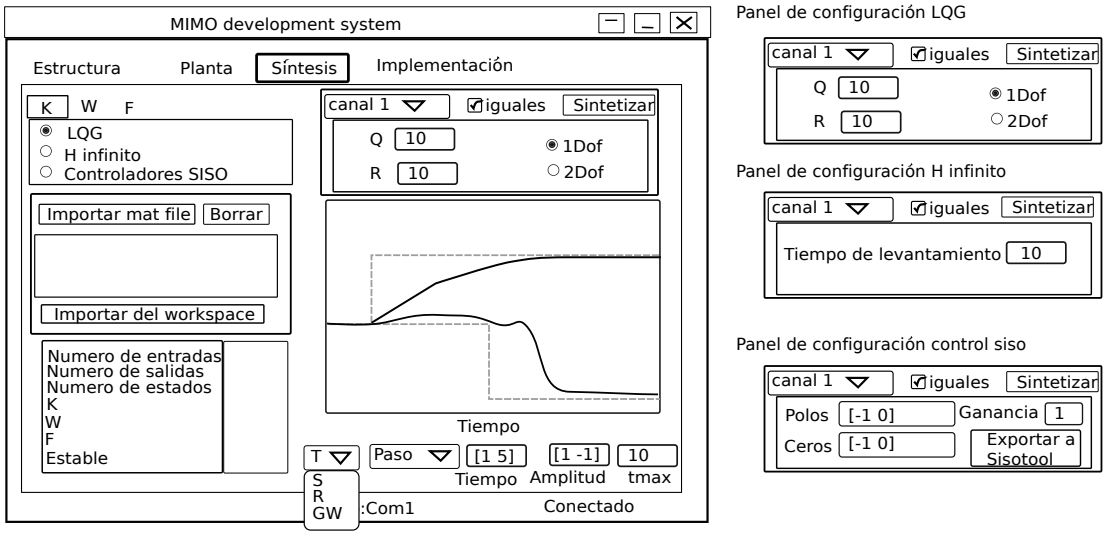


Figura 5.3: Panel para síntesis de K con los subpaneles para diseño LQG, H_∞ y control *SISO*.

En la figura (5.5) se muestra el panel para síntesis del bloque K, es posible elegir entre diseño LQG, H_∞ y controles SISO; cada método de diseño posee un panel auxiliar que permite ingresar los parámetros de diseño.

LQG: Los parámetros de diseño son los pesos Q y R asociados a velocidad y señal de control respectivamente, adicionalmente es posible elegir entre la estructura de *1Dof* o *2Dof*.

Se puede elegir el mismo peso para todos los canales o elegir un peso diferente para cada canal empleando la función de diseño LQG descrita en el capítulo 3. Durante el diseño se adiciona un integrador a cada canal para garantizar el desacople en lazo cerrado de forma que se recomienda aplicar un esquema antiwindup al implementar esta estrategia.

$\mathbf{H}_\infty$: El diseño se basa en la función *dhinfts* el único parámetro de diseño es el tiempo de levantamiento el cual se puede ser escalar o vectorial, cuando se especifica el tiempo para todos los canales o para cada canal respectivamente.

Controladores SISO: Al asumir controladores *SISO* se asume que la planta es aproximadamente diagonal. Los parámetros corresponden a la descripción *zpk*(zero-pole-gain), para el controlador *SISO* asociado al canal de forma que el controlador será de la forma.

$$K = \begin{bmatrix} k_1 \frac{\prod_{i=0}^{nc}(s+c_i^1)}{\prod_{i=0}^{np}(s+p_i^1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & k_n \frac{\prod_{i=0}^{nc}(s+c_i^n)}{\prod_{i=0}^{np}(s+p_i^n)} \end{bmatrix}$$

En plantas no cuadradas se ajusta la dimensión del compensador agregando filas de ceros para el caso $m < n$ o columnas de ceros para el caso $m > n$.

5.2.2. Síntesis de W

La síntesis del elemento de desacople se puede realizar de forma estática o dinámica, cuando el desacople es estático se busca que el producto GW sea una identidad para una frecuencia, un desacople dinámico diagonaliza la planta para todas las frecuencias.

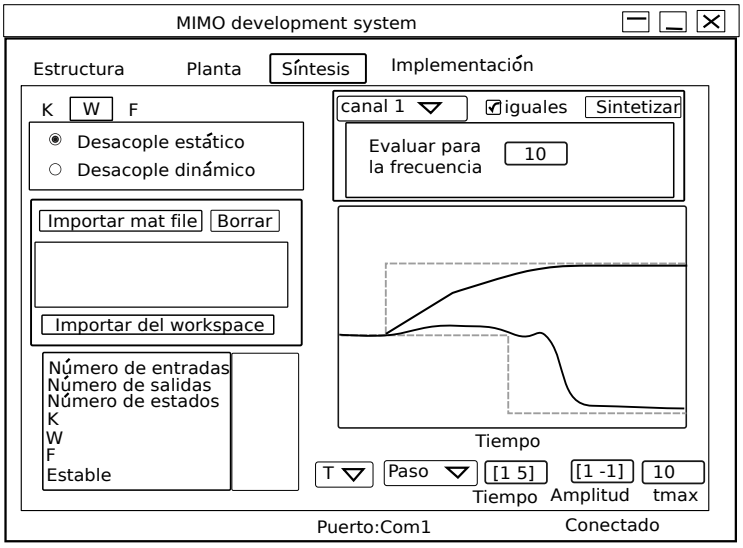


Figura 5.4: Panel para síntesis del elemento de desacople W

En la interfaz se debe definir la frecuencia angular para la cual se desea evaluar la red de desacople estático usualmente se evalua para DC ($\omega = 0$) o para la frecuencia cruce (ω_{-3dB}) . La función de desacople dinámico esta basada en la función *desacople* descrita en el capítulo 3.

5.2.3. Síntesis de F

Como elemento de precompensación se cuenta con filtros *notch*, pasaaltos y pasabajos, los parámetros son similares para los tres casos, estos parámetros corresponden a la frecuencia del filtro, la profundidad y el ancho para el caso del filtro notch. Teniendo en cuenta que el filtro se aplica a los canales de entrada del sistema realimentado se adicionó la posibilidad de sintonizar el lazo visualizando la respuesta en frecuencia del filtro y el lazo con y sin prefiltro.

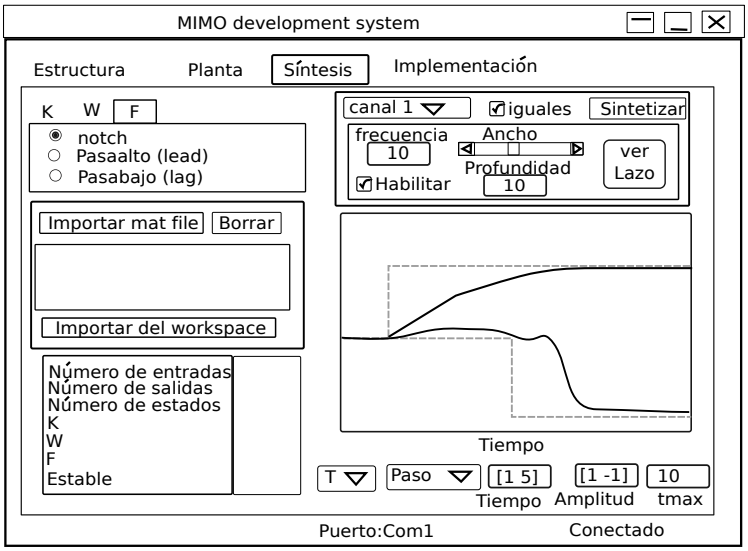


Figura 5.5: Panel para síntesis del precompensador F

Notch: El filtro notch rechaza una frecuencia de forma que se puede emplear para atenuar fenómenos de resonancia en el sistema realimentado la forma general del filtro notch es.

$$W = \frac{s^2 + 2\alpha_1 s\omega + \omega^2}{s^2 + 2\alpha_2\omega + \omega^2}$$

Donde la profundidad depende de la relación α_1 y α_2 la profundidad en decibeles esta dada por

$$\text{profundidad}_{dB} = 20\log_{10}\left(\frac{\alpha_1}{\alpha_2}\right)$$

El ancho del filtro define el ancho de banda para el cual es efectiva la atenuación.

Ejemplo Se asume que la función de lazo cerrado $GWK(I+GWK)^{-1}$ tiene de la primer entrada a la primer salida la función de transferencia

$$Slc_{1,1} = \frac{1}{s^2 + s + 1}$$

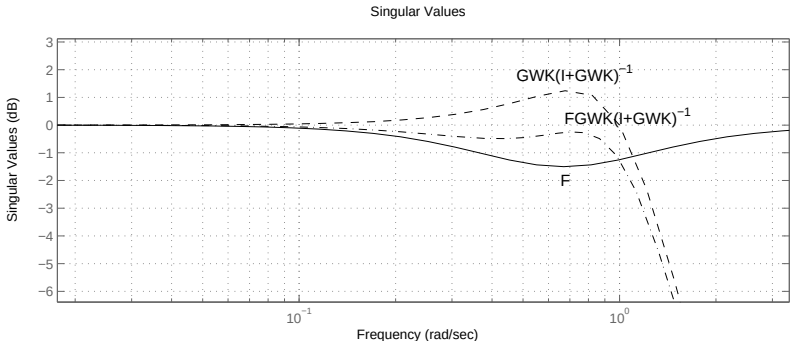


Figura 5.6: Ejemplo de aplicación de un filtro notch

Esta función posee un pico a en $\omega = 0{,}67 rad/seg$, usando un prefiltro se propone atenuar el pico en frecuencia para lo cual se propone un filtro notch con frecuencia central en $\omega = 0{,}67 rad/seg$,

profundidad de 1,5dB y ancho de 0,5rad el comportamiento en frecuencia resultante se presenta en la figura (5.6) donde se observa que el sistema precompensado presenta un ancho de banda similar al sistema original pero no presenta sobrepico.

Pasaalto: El filtro pasaalto se emplea para aumentar el ancho de banda del sistema realimentado se emplea la función prototipo

$$F = \frac{\alpha_2}{\alpha_1}\frac{(s+\alpha_1)}{(s+\alpha_2)} \qquad \text{con} \quad \alpha_1 = \omega \quad \alpha_2 = \omega 10^{\frac{\text{profundidad}}{20}}$$

Donde la profundidad especifica la ganancia máxima del filtro el cual se encuentra normalizado para tener ganancia unitaria en la banda de rechazo.

Pasabajo: Se emplea para reducir el ancho de banda del sistema realimentado emplea la función prototipo.

$$F = \frac{\alpha_2}{\alpha_1}\frac{(s+\alpha_1)}{(s+\alpha_2)} \qquad \text{con} \quad \alpha_1 = \omega 10^{\frac{\text{profundidad}}{20}} \quad \alpha_2 = \omega$$

La profundidad especifica la atenuación en decibeles cuando ω tiende a infinito.

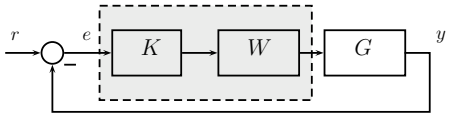
5.2.4. Síntesis del controlador central

El controlador central para la estructura WKF es un único elemento que agrupa estos elementos preservando el comportamiento en lazo cerrado en la figura (5.7) su síntesis depende de la estructura.

Estructura *1Dof* sin prefiltro

En este caso el controlador resulta ser

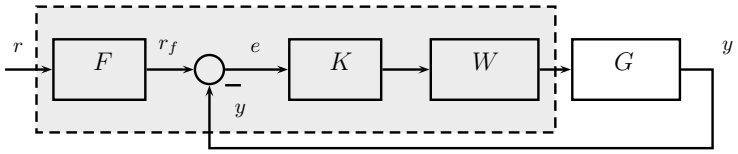
Controlador:	$K_{cen} = WK$
Sensitividad Complementaria:	$T = GWK(I+GWK)^{-1}$
Sensitividad:	$S = (I+GWK)^{-1}$
Señal de control:	$R = WK(I+GWK)^{-1}$



Estructura *1Dof* con prefiltro

Siendo r_f la referencia filtrada se plantea que $r_f = Fr$, $e = r_f - y$ entonces $e = Fr - y$ finalmente $u = WKe = WKFr - WKy$.

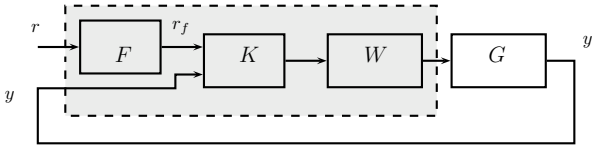
C: $K_{cen} = \begin{bmatrix} WKF & -WK \end{bmatrix}$
T: $T = GWK(I+GWK)^{-1}F$
S: $S = (I+GWK)^{-1}$
R: $R = WKF(I+GWKF)^{-1}$



Estructura *2Dof*

Se tiene la siguiente estructura en la cual $rf = rF$ de forma que la señal de control es

$$u = WK \begin{bmatrix} rf \\ y \end{bmatrix} = WKFry$$



De forma que el controlador central solo puede ser planteado en espacio de estados o fraccionando el controlador K de la forma $K = \begin{bmatrix} K_r & K_y \end{bmatrix}$. Se presenta la representación en espacio en la figura (5.7) obteniendose la representación en espacio de estados del controlador central . Esta corresponde a

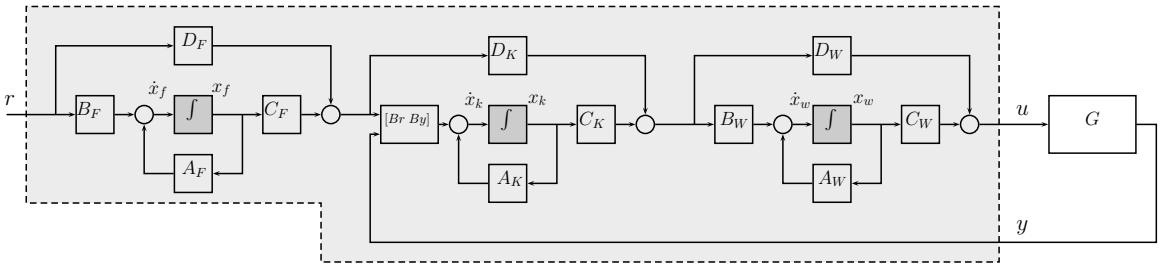


Figura 5.7: Representación en espacio de estados del controlador central

$$\begin{bmatrix} \dot{x}_W \\ \dot{x}_K \\ \dot{x}_F \end{bmatrix} = \begin{bmatrix} A_W & B_W C_K & B_W D_K C_F \\ 0 & A_K & B_r C_F \\ 0 & 0 & A_F \end{bmatrix} \begin{bmatrix} x_W \\ x_K \\ x_F \end{bmatrix} + \begin{bmatrix} B_W D_K D_F & 0 \\ B_r D_F & B_y \\ B_F & 0 \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix} \tag{5.1}$$

$$u = \begin{bmatrix} C_W & D_W C_K & D_W D_K C_F \end{bmatrix} \begin{bmatrix} x_W \\ x_K \\ x_F \end{bmatrix} + \begin{bmatrix} D_W D_K D_F & 0 \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix} \tag{5.2}$$

5.3. Implementación

La implementación consiste en descargar los coeficientes que describen el controlador y almacenarlos en el dispositivo de procesamiento. La interfaz cuenta con un panel que permite la configuración del dispositivo hardware dependiendo de la realización, es posible modificar el tipo de operador, el tipo de aritmética, ajustar el periodo de muestreo y añadir canales de antiwindup a la realización. Cuando se elige que el procesador debe realizar las operaciones usando punto fijo es necesario especificar el número de bits fraccionales; cuando se elige la implementación con antiwindup se debe especificar el valor de señal de control máxima U_{max} y mínima U_{min}.

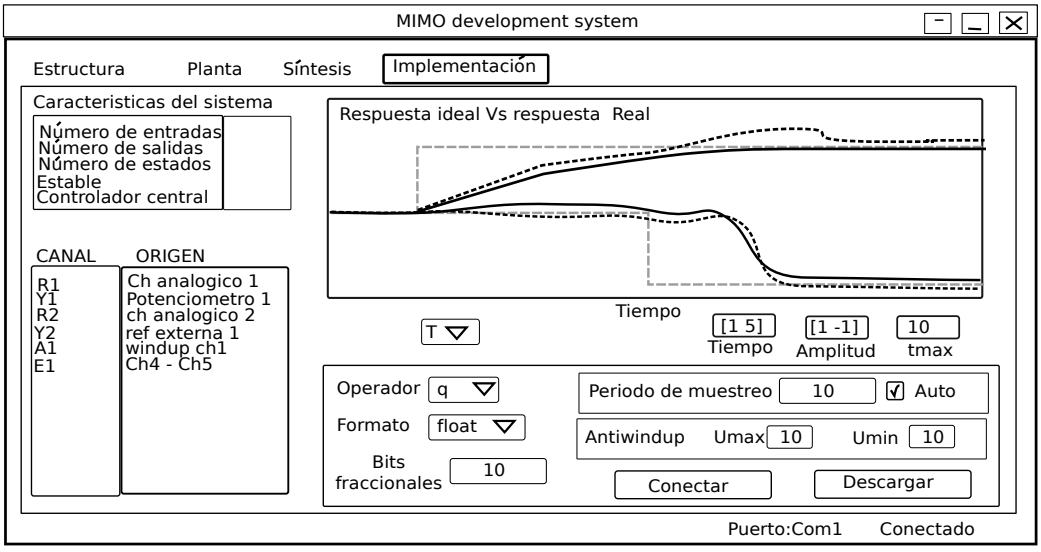


Figura 5.8: Panel de implementacion

Se cuenta con una ventana para configurar el origen de los canales de entrada al dispositivo hardware, dependiendo del método de diseño en la etapa de síntesis los canales son etiquetados con las letras E (error), R (referencia), Y (salida) y A (antiwindup). E hace referencia a una señal de error la cual

corresponde a la entrada de un controlador de *1Dof*, R y Y hacen referencia a canales de referencia y salida respectivamente los cuales constituyen la entrada a un controlador *2Dof*, A hace referencia a un canal de antiwindup por el cual ingresa una señal de control saturada.

- **Canal analógico:** Se emplea cuando la entrada corresponde a una señal analógica en el rango $[-10\ 10]V$, sirve cuando la entrada es de tipo R o de tipo Y.
- **Canal diferencial:** Se emplea para asignar la entrada a la diferencia de dos canales analógicos, puede ser empleada para ingresar señales tipo E.
- **Referencia externa:** Se emplea para enviar un valor de referencia de forma remota a traves del puerto de comunicación *USB* o serial.
- **Potenciometro:** Se utiliza para señales tipo R cuando se desea que el dispositivo opere basado en el voltaje suministrado por un potenciometro presente en la tarjeta.

Una vez se tiene la configuración del dispositivo es posible simular el comportamiento que tendrá el dispositivo basándose en el comportamiento descrito por el modelo la interfaz sobrepone el comportamiento ideal con respecto al modelo obtenido empleando *Fixed Point Toolbox*.

5.4. Metodología de diseño

El desarrollo de esta interfaz permitió el desarrollo de una estrategia para diseño y síntesis de sistemas multivariables. Esta metodología se basa en las características de la planta.

Plantas cuadradas de fase mínima Este es el caso más fácil debido a que no se precisa el uso de redes de desacople se puede seguir el siguiente procedimiento de diseño.

1. Verificar si el sistema estabiliza usando síntesis H_∞, de lo contrario se emplea la síntesis LQG.
2. Se visualiza la señal de control eligiendo como señales de referencia las señales máximas que se esperan en la entrada.
3. En el caso de una síntesis H_∞ reducir el tiempo de establecimiento hasta donde sea posible garantizando que la señal de control se encuentra en región lineal si la síntesis es LQG se procede de igual manera aumentando el peso Q.
4. Si es necesario se calculan filtros *notch* para eliminar sobrepico.
5. Al garantizar que la senãl de control se encuentra en región lineal no se precisa el uso de estrategias AW.

Plantas estables con ceros *RHP* La existencia de ceros de transmisión dificulta levemente el desacople de canales, se puede proceder de forma similar al caso anterior, teniendo cuidado en la interacción entre canales.

1. Se intenta estabilizar la planta usando H_∞o LQG para evaluar la interacción entre canales se visualiza la respuesta de lazo cerrado T y se verifica para varias combinaciones de entrada.
2. Si la interacción es considerable es mejor diseñar una red de desacople dinámico y realizar nuevamente la síntesis del bloque K.
3. Se ajustan especificaciones de tiempo de establecimiento mediante la elección de pesos.

Plantas rectangulares Solo para plantas $m \times n$ con $m < n$, en este caso no es posible usar redes de desacople de forma que se dispone de la síntesis LQG y la formulación de controladores *SISO* para definir el bloque K, no existe limitación en cuanto al uso del precompensadores.

Plantas inestables En plantas inestables no es posible sintetizar redes de desacople, la síntesis H_∞presenta dificultades cuando hay polos sobre el eje imaginario, de forma que se cuenta con la síntesis LQG como medio de diseño.

Durante la síntesis de debe tener en cuenta las siguientes recomendaciones.

- Verificar si la relación orden periodo de muestreo satisface las condiciones de implementación ver la figura (4.9).
- Visualizar características como la señal de control para garantizar que la señal de control no presenta desborde, verificar esto para varias señales de entrada.
- En lo posible utilizar punto flotante como formato numérico con operador q para aplicaciones prácticas, las implementaciones en punto fijo y operador δ se pueden emplear con fines académicos.
- Cuando el controlador es diseñado empleando LQG y la planta es estable se recomienda realizar la implementación con AW.

Conclusiones

- Es posible encontrar una red de desacople para cualquier sistema estable y cuadrado, permitiendo reducir la interacción de canales y facilitar la aplicación de otras técnicas de diseño pero teniendo como inconveniente que el orden del controlador implementado puede resultar elevado.
- El uso del bloque F permite ajustar la respuesta final del sistema ecualizando la respuesta frecuencial de la función de lazo cerrado, es posible eliminar el sobrepico usando filtros notch, incrementar el ancho de banda con un filtro pasaalto o reducirlo con un pasabajo.

Capítulo 6
Resultados

6.1. Efecto del operador 57
6.2. Efecto del formato numérico 58
6.3. Casos de estudio 60
6.3.1. Emulación 60
6.3.2. Control 61
6.3.3. Control LQG 62
6.3.4. Control H_∞ 63

6.1. Efecto del operador

Se emulará un sistema multivariable implementado con operador δ y operador q. El sistema de prueba es una planta 2×2 de 7 estados dada por la ecuación (6.1).

$$G = \begin{bmatrix} \frac{4}{s^2+s+4} & -\frac{1}{s+1} \\ -\frac{3}{s^2+s+5} & -\frac{4(s-2)}{s^2+5s+8} \end{bmatrix} \tag{6.1}$$

En la figura (6.1) se muestra el comportamiento de las realizaciones al emplear representaciones en punto fijo con 20, 19 , 8 y 5 bits fraccionales, en los dos casos las realizaciones se implementaron en forma balanceada con un periodo de muestreo $T_s = 0{,}1s$, del comportamiento observado se puede concluir que

1. El rango dinámico de la realización con operador δ es mayor entonces se pueden presentar de desbordes numéricos cuando se asignan muy pocos bits a la parte entera esto se demuestra en la figura (6.1) en la cual se asignó 20 bits para la parte fraccional, se observa que la realización δ presenta desborde mientras que la realización q se asemeja a la respuesta ideal.

2. Cuando se eligió la representación A(12,19) la respuesta con operador q y operador δ son identicas.

3. La realización con operador δ es más sensible a la perdida de resolución esto se verifica en las gráficas en las cuales se emplea el formato A(23,8) y A(26,5) donde se observa que para la realización δ con 8 bits fraccionales hay un deterioro significativo en la respuesta y para 5 bits fraccionales se pierde estabilidad, por otra parte la realización con operador q sufre un menor deterioro.

4. En general una realización con operador q presenta mejor condicionamiento numérico.

5. Algunas pruebas no mostradas en este documento mostraron que cuando la implementación es en punto flotante el cambio de operador no afecta la implementación.

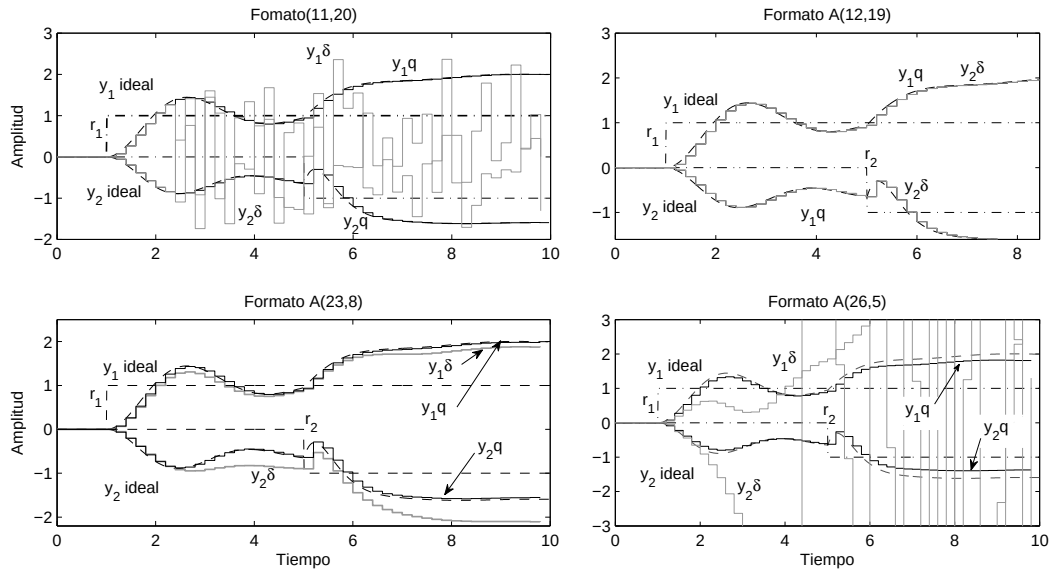


Figura 6.1: Sistema emulado con operador δ y operador q en punto fijo con diferentes formatos numéricos

6.2. Efecto del formato numérico

Se diseñará un sistema de control para la planta (6.2) este es un sistema estable 3×3 sin ceros *RHP* es un sistema fuertemente acoplado como se observa en la figura (6.2.a) y además se encuentra mal condicionado como se observa en la gráfica de valores singulares (6.2.b).

$$G = \begin{bmatrix} \frac{1}{s^2+s+1} & \frac{1}{s+1} & 0 \\ \frac{s+9}{s+4} & \frac{s-2}{s+8} & \frac{1}{s+3} \\ \frac{1}{s+1} & \frac{3}{s^2+3s+2} & \frac{1}{s+3} \end{bmatrix} \tag{6.2}$$

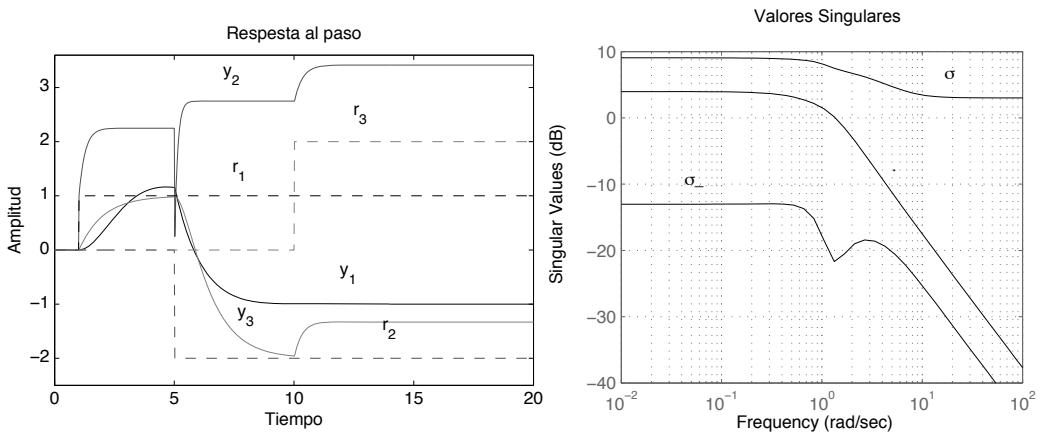


Figura 6.2: Características de la planta multivariable (a) Respuesta al paso (b) gráfico de valores singulares

Elección de la estrategia de control Se intentará sintetizar una red de desacople dinámico para este sistema resulta de orden 60 de forma que no es viable su implementación entonces se realiza el diseño usando la síntesis LQG y H_∞.

En la figura (6.3) se presenta la respuesta al paso para controladores LQG y H_∞, se observa que el diseño H_∞presenta un mejor desacople de canales y un menor tiempo de establecimiento, en cuanto al orden del controlador los dos son de orden 12, pero el controlador H_∞tiene como ventaja el ser estable mientras que el controlador LQG posee un integrador por cada canal.

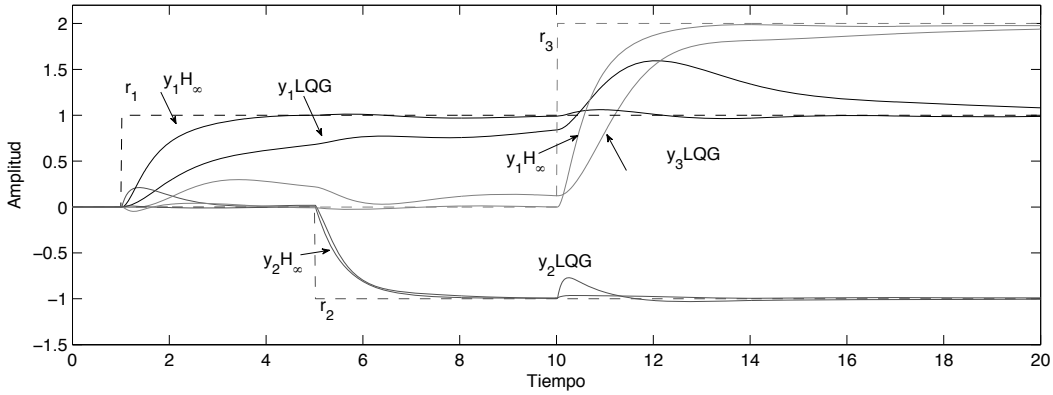


Figura 6.3: Respuesta en lazo cerrado usando controladores LQG y H_∞

Implementación Para la implementación se hace el análisis con el controlador H_∞empleando formatos numéricos de punto fijo y punto flotante los resultados se muestran en la figura 6.4.

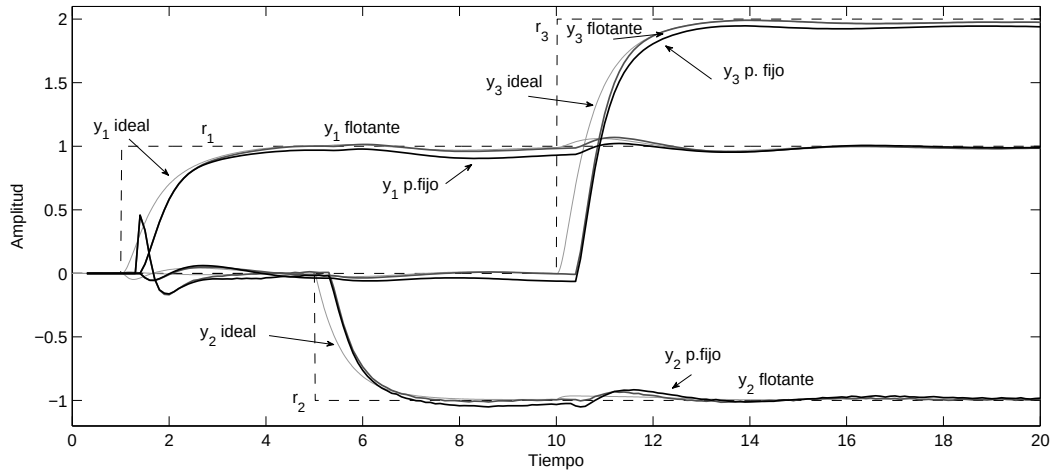


Figura 6.4: Respuesta en lazo cerrado sistema con controlador H_∞implementado en punto flotante y punto fijo

El periodo de muestreo sugerido para la implementación es $T_s = 30ms$ pero se realiza la implementación con un periodo $T_s = 0{,}1s$ para degradar la respuesta. En la figura (6.4) se observa que la respuesta se encuentra cercana a la respuesta ideal para implementación en punto flotante en cuanto al implementación en punto flotante se tienen algunos errores pequeños. en la figura (6.5) se presenta la señal de control donde la linea punteada corresponde a la respuesta ideal, se observa que las señales de control evaluadas para las dos implementaciones resultan ser similares para los canale uno y dos y presenta un leve error para el canal tres.

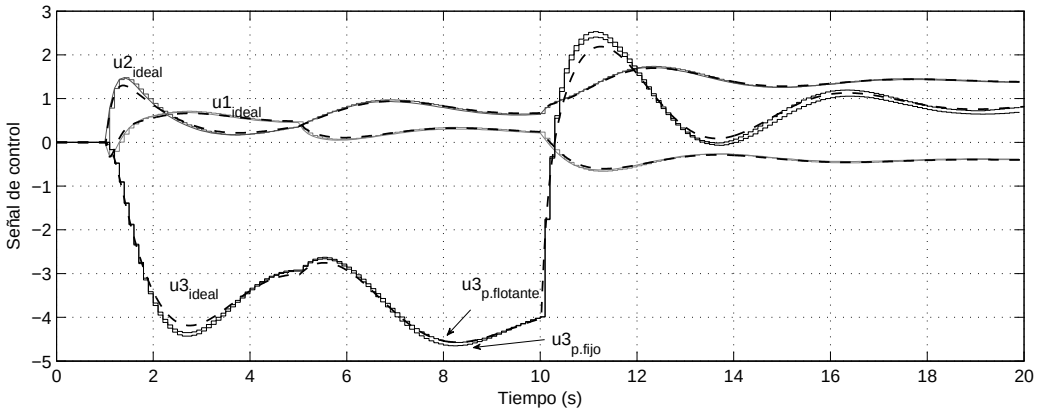


Figura 6.5: Señal de control sistema con controlador H_∞implementado en punto flotante y punto fijo

6.3. Casos de estudio

Para las pruebas mencionadas a continuación se implementó un esquema *hardware in the loop* el cual está compuesto por un software de tiempo real (Real time Windows target) y una tarjeta de adquisición *PCI 6024E* de National Instruments.

Esta configuración permite que el computador pueda comportarse como la planta donde sus entradas y salidas pueden ser tomadas directamente de la tarjeta de adquisición. Adicionalmente en el anexo B se estudia un reactor de polimerización y se muestra la interfaz gráfica de usuario.

6.3.1. Emulación

sistema desacoplado

Se implementa sobre el dispositivo el sistema

$$\begin{bmatrix} -\frac{s-0,2}{s^2+2s+4} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix}$$

El cual corresponde a un sistema de dos entradas dos salidas desacopladas, este sistema se discretizó con un periodo de muestreo $T_s = 10ms$ y se implementó en punto fijo con un fomato $A(19, 12)$. En

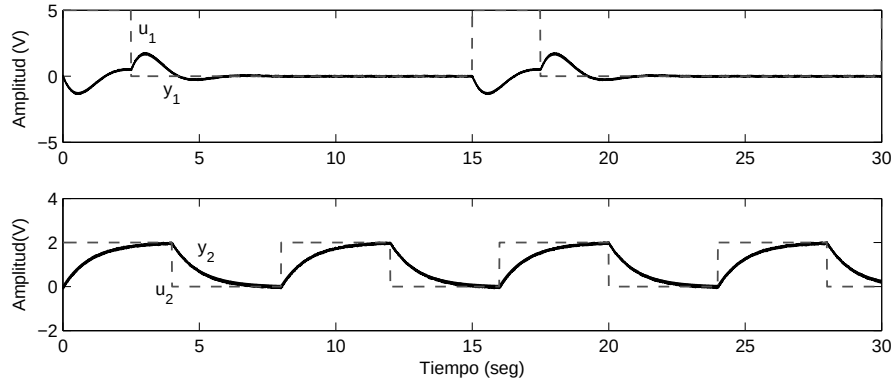


Figura 6.6: Respuesta del sistema emulando una planta 2×2 desacoplada

la figura 6.6 se muestra el comportamiento del sistema desacoplado, en esta misma gráfica se presenta el comportamiento ideal del sistema resultando igual al sistema implementado.

sistema acoplado

Se tomó el sistema desacoplado y se añadió funciones de transferencia para crear interacción entre lazos resultando el siguiente sistema.

$$H = \begin{bmatrix} -\frac{s-0,2}{s^2+2s+4} & \frac{1}{s+6} \\ -\frac{s-2}{s+5} & \frac{1}{s+1} \end{bmatrix}$$

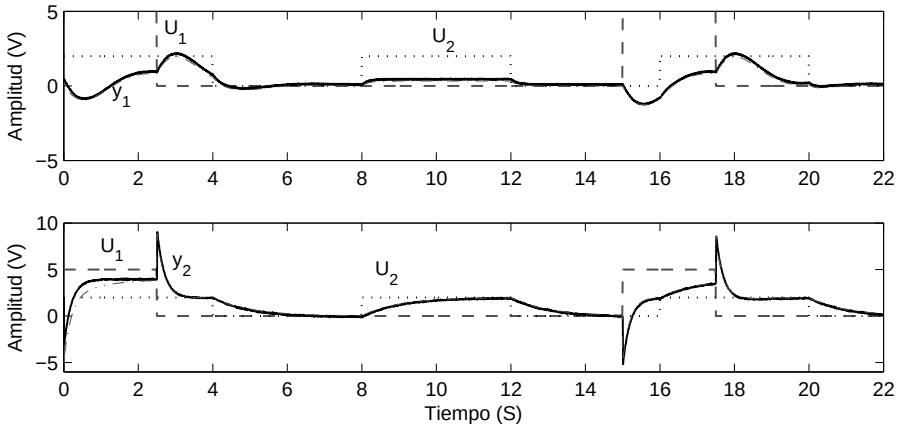


Figura 6.7: Respuesta del sistema emulando una planta 2×2 acoplada

La implementación se realizó con un periodo de muestreo $T_s = 10mS$ usando operador δ en punto flotante , en la figura 6.7 se presenta el comportamiento del sistema con respecto al comportamiento ideal, donde se observa que se encuentran completamente sobrepuestas .

6.3.2. Control

Las pruebas de control de sistemas tienen como finalidad el verificar la capacidad del dispositivo para operar en lazo cerrado y lograr controlar un sistema.

Control de un sistema Siso

Se propone el control del sistema

$$G = \frac{1}{s^2 + 0{,}5s + 1}$$

El cual es un sistema *SISO* con frecuencia natural $\omega_n = 1$ y $\zeta = 0{,}25$ lo cual lo hace un sistema con poco amortiguamiento y un sobrepico mayor al 40 %.

Usando la interfaz de usuario se implementó un controlador LQG con función de transferencia:

$$K = \frac{2s^2 + s + 1{,}5}{s^3 + 2{,}5s^2 + 3{,}5s}$$

En la 6.8 se muestrá el comportamiento en lazo cerrado para una implementación en punto fijo $A(23, 8)$ con operador q, donde la línea continua corresponde a la respuesta del lazo *hardware in the loop*, y la linea punteada corresponde a la respuesta ideal. En este caso se observan algunas diferencias debido a que la implementación asigna muy pocos bits a la parte fraccional sin embargo la respuesta obtenida muestra una tendencia hacia la respuesta ideal.

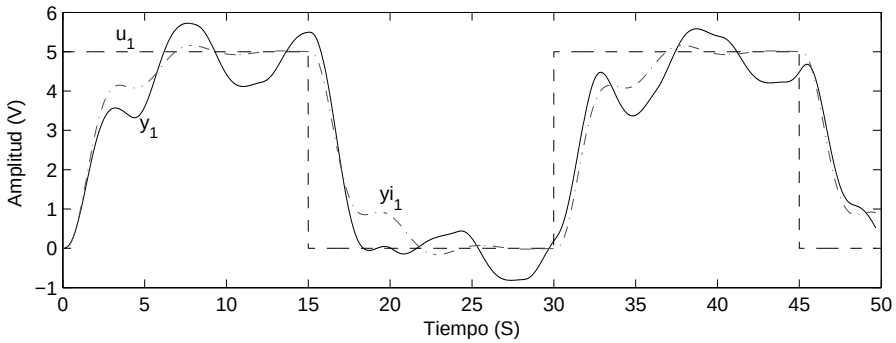


Figura 6.8: Respuesta en lazo cerrado para una planta siso con control LQG

6.3.3. Control *LQG*

Se propone el control de un sistema acoplado representado por la matriz de transferencia

$$G = \begin{bmatrix} \frac{1}{s+3} & \frac{3}{s+5} \\ \frac{5}{s+10} & \frac{2}{s^2+s+8} \end{bmatrix}$$

Usando la interfaz de usuario se diseño un controlador LQG usando como pesos [Q=10, R=1] para ambos canales. El controlador está dado por la siguiente expresión:

$$K = \frac{1}{Den} \begin{bmatrix} 0{,}29(s-6{,}27)(s+5{,}53)(s+3{,}05)(s+10{,}33)(s+1{,}52) & 3{,}59(s+3{,}05)(s+6{,}95)(s+10{,}02)(s^2+1{,}137s+7{,}76) \\ 3{,}39(s+11{,}98)(s+5{,}02)(s+2{,}99)(s^2+1{,}38s+8{,}63) & -0{,}21(s+16{,}17)(s+10{,}15)(s+5{,}03)(s^2+1{,}31s+8{,}40) \end{bmatrix}$$

$$Den = s(s+11{,}99)(s+6{,}97)(s+3{,}04)(s^2+1{,}37s+8{,}55)$$

La implementación se realizó con operador q y formato de punto flotante y un periodo de muestreo de $T_s = 10mS$.

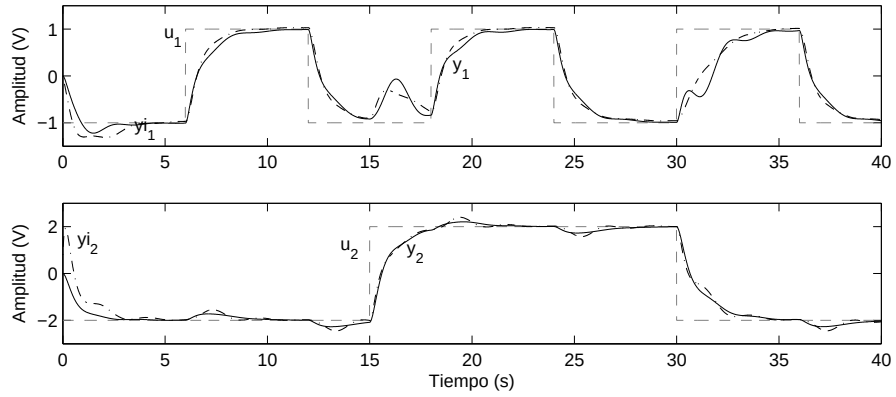


Figura 6.9: Respuesta en lazo cerrado para un sistema multivariable con control LQG

La figura 6.9 presenta en linea continua el comportamiento en lazo cerrado obtenido bajo el esquema hardware in the loop, la línea punteada corresponde al comportamiento ideal, se observan algunas diferencias sobretodo para $t = 0$ ya que en el inicio de la prueba los estados del controlador no eran cero como en el modelo ideal. también se observan algunas diferencias no muy notorias durante el transcurso de la prueba.

6.3.4. Control H_∞

Para el diseño H_∞ se sintetizó un controlador que permitiera que los canales tuvieran un tiempo de establecimiento de 5seg.

A continuación se muestra la matriz de transferencia del controlador:

$$K = \frac{1}{Den}\begin{bmatrix} 0{,}86s^5 + 10{,}64s^4 - 80{,}16s^3 - 1780s^2 - 9180s - 14330 & 5{,}8s^5 + 237{,}2s^4 + 2476s^3 + 8788s^2 + 22540s + 376000 \\ 5{,}873s^5 + 309{,}9s^4 + 2503s^3 + 8476s^2 + 21110s + 31140 & -0{,}85s^5 - 69{,}18s^4 - 891{,}9s^3 - 3845s^2 - 9015s - 19860 \end{bmatrix}$$

$$Den = s^6 + 74{,}17s^5 + 1428s^4 + 5200s^3 + 1{,}215 \times 10^4 s^2 + 2{,}105 \times 10^4 s + 150{,}9$$

Este controlador se implementó en punto flotante con operador q y un periodo de muestreo $T_s = 10mS$.

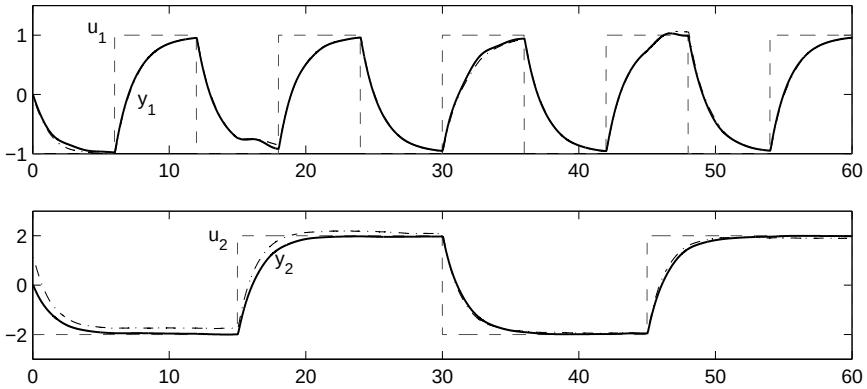


Figura 6.10: Respuesta en lazo cerrado para un sistema multivariable con control H_∞

La figura 6.10 muestra el comportamiento en lazo cerrado, donde se observa que al igual que con el controlador LQG las diferencias significativas se deben principalmente a la condición inicial del controlador.

Conclusiones

- En una implementación en punto flotante usando operador δ y operador q la respuesta resultan indistinguibles, debido a que este formato numérico presenta un gran rango dinámico, cuando la implementación se realiza en punto fijo la realización con operador δ requiere un estudio para determinar el formato numérico apropiado y así evitar problemas de desborde numérico o perdida de resolución.

- Se observó que las realizaciones con operador δ presentan mayor sensibilidad al formato numérico, como la matriz A_δ se encuentra normalizada por Δ^{-1} y Δ suele ser muy pequeño se trabaja con cantidades grandes necesitando un mayor número de bits enteros, por otra parte durante el cálculo del siguiente estado se desnormaliza la variable intermedia $x[n+1] = \Delta T + x$ y las cantidades involucradas en el cálculo suelen ser pequeñas requiriendo un número considerable de bits fraccionales entrando en contradicción con lo anteriormente planteado ya que no es posible asignar bits fraccionales y enteros a la vez.

- Una realización en punto fijo con operador δ tiene mas posibilidad de no operar adecuadamente ya que el rango de formatos numéricos apropiados para implementación es menor que para una implementación con operador q.

6.4. Conclusiones

- Al adicionar integradores en el lazo de control se garantiza la reducción del acople entre canales pero se compromete la estabilidad de la realización lo cual puede traer problemas por los posibles desbordes numéricos; se debe garantizar que las implementaciones digitales sean estables o por lo menos implementar técnicas de saturación numérica o de detección de saturación en actuadores para prevenir el crecimiento de las variables de estado.
- Las formas canónicas se caracterizan por tener un considerable número de coeficientes cero disminuyendo el número de operaciones necesarias para evaluar la realización, esto trae como consecuencia que la realización se hace mas sensitiva a la variación de coeficientes.
- La forma δ es una realización no mínima que puede resultar de ayuda en implementaciones donde no hay restricción en el tiempo de procesamiento y se cuenta con frecuencias de muestreo elevadas, en esta realización el valor de los coeficientes y de los estados tiende a ser grande de forma que los decimales se pueden hacer despreciables.
- En sistemas estables el cálculo de redes de desacople resulta simple y permite que el diseño de controladores multivariables se convierta en un problema *SISO*.
- La solución del problema de sensibilidad mixta permite especificar mediante pesos características como el tiempo de establecimiento, sobrepico y amplitud de la señal de control, el uso de los pesos tiene como desventaja el incremento en el orden del controlador, en un controlador LQG los pesos aplicados nunca alteran el orden del controlador resultando en un controlador con el mismo orden de la planta para regulación o planta mas número de entradas cuando se adicionan integradores para seguimiento de referencia.
- El esquema antiwindup propuesto es aplicable únicamente a controladores con integrador en plantas estables. Debido a la función saturación que relaciona la salida del controlador con la entrada, el controlador es no lineal y el análisis de estabilidad no puede realizarse aplicando la teoría de sistemas lineales, se podría emplear el teorema de estabilidad de *Lyapunov*.
- En una implementación de punto fijo la elección del número de bits fraccionales es fundamental, cuando el controlador es estable es posible recurrir a la norma infinito para estimar el valor máximo de las variables de estado de forma que se puede determinar el número de bits enteros mínimos necesarios y dejando los sobrantes como bits fraccionales, cuando el controlador es inestable (en lo posible solo por poseer integradores) el número de bits para la parte entera tiende a infinito siendo obligatorio el uso de una técnica antiwindup.
- Se propuso una método para determinar el formato numérico para implementaciones en punto flotante con operador q.
- Para implementaciones en punto fijo las formas balanceadas resultan ser la mejor alternativa , el estudio comparativo mostró que en general este tipo de realizaciones debido a su condicionamiento numérico pueden ser evaluadas empleando un menor número de bits enteros siendo posible aumentar en número de bits fraccionales mejorando la presición.

- Las formas observable y controlable presentan un significativo número de coeficientes iguales a cero de forma que su tiempo de procesamiento resultaría menor, pero al analizar el número de bits enteros se concluye que restan presición a la implementación.
- El tiempo de procesamiento de una realización depende principalmente del número de estados, en sistemas multivariables la adición de un canal (entrada- salida) resulta equivalente a aumentar la realización un estado.
- Se esperaba que para realizaciones en punto flotante el tiempo de procesamiento resultara considerablemente mayor con respecto a las implementaciones en punto fijo, pero las mediciones realizadas muestran que son muy similares, un análisis detallado reveló que la componente dominante del tiempo de procesamiento se encuentra en el direccionamiento, lectura de memoria y adquisición de señales analógicas.
- Debido a que no hay diferencias significativas en el tiempo de procesamiento al cambiar el formato numérico se recomienda el uso de realizaciones en punto flotante y de esta forma evitar problemas de desborde numérico y escalamiento.
- Es posible encontrar una red de desacople para cualquier sistema estable y cuadrado, permitiendo reducir la interacción de canales y facilitar la aplicación de otras técnicas de diseño pero teniendo como inconveniente el orden de implementación que puede resultar elevado.
- El uso del bloque F permite ajustar la respuesta final del sistema ajustando la respuesta frecuencial de la función de lazo cerrado, es posible eliminar el sobrepico usando filtros notch, incrementar el ancho de banda con un filtro pasaalto o reducirlo con un pasabajo.
- En una implementación en punto flotante usando operador δ y operador q la respuesta resultan indistinguibles, debido a que este formato numérico presenta un gran rango dinámico, cuando la implementación se realiza en punto fijo la realización con operador δ requiere un estudio para determinar el formato numérico apropiado y así evitar problemas de desborde numérico o perdida de resolución.
- Se observó que las realizaciones con operador δ presentan mayor sensibilidad al formato numérico, como la matriz A_δ se encuentra normalizada por Δ^{-1} y Δ suele ser muy pequeño se trabaja con cantidades grandes necesitando un mayor número de bits enteros, por otra parte durante el cálculo del siguiente estado se desnormaliza la variable intermedia $x[n+1] = \Delta T + x$ y las cantidades involucradas en el cálculo suelen ser pequeñas requiriendo un número considerable de bits fraccionales entrando en contradicción con lo anteriormente planteado ya que no es posible asignar bits fraccionales y enteros a la vez.
- Una realización en punto fijo con operador δ tiene mas posibilidad de no operar adecuadamente ya que el rango de formatos numéricos apropiados para implementación es menor que para una implementación con operador q.

6.5. Aportes

- El desarrollo de una interfaz de usuario para diseño de controladores para plantas *MIMO*, con soporte hardware para implementaciones de hasta 4 entradas, 4 salidas y 50 estados.
- Se desarrolló un modelo no lineal que describe de forma precisa el funcionamiento del hardware y permite simular el comportamiento de la implementación.
- Se propuso un esquema antiwindup para sistemas multivariables.
- Se propuso un método para determinar el rango dinámico de realizaciones en punto fijo.
- Se desarrollaron los algoritmos para ejecución de realizaciones q y δ sobre microcontrolador.

- Se propuso una función para síntesis H_∞ fijando el tiempo de establecimiento esperado para canal.
- Se propuso la estructura de control WKF y una metodología de diseño.
- Se desarrolló una interfaz de usuario para asistir el diseño y la síntesis de controladores.

6.6. Trabajo futuro

En este trabajo se abordó el problema de realización de sistemas lineales para control, lo cual resulta en una idealización en la cual se suele asumir un modelo lineal para la planta y resulta eficaz cerca al punto de operación. Resulta interesante desarrollar un dispositivo capaz de implementar ecuaciones de estado no lineales y otras estrategias como implementaciones *bumpless* y *control en modo deslizante*.

Apendice 1

A.1. Instalación del driver usb

Cuando la plataforma es utilizada empleando un puerto usb se hace necesario instalar el driver del conversor usb serial.

A.1.1. PARA WINDOWS XP - Vista

Se debe descargar e instalar el driver
http://www.ftdichip.com/Drivers/CMD/CMD20602.zip
Se instala y luego se puede verificar en sistema - administración de hardware - puertos serie el número de puerto *COM* que le fue asignado.

A.1.2. PARA LINUX

La tarjeta no necesita de drivers para ninguna versión basta con conectar el dispositivo y verificar el nombre que el sistema le asigna esto lo puede hacer con el comando `ls/dev/` el nombre suele ser `/dev/ttyUSB0`.

A.1.3. Instalación Mac Os X leopard

Se debe descargar e instalar el driver
http://www.ftdichip.com/Drivers/VCP/MacOSX/UniBin/FTDIUSBSerialDriver_v2_2_14.dmg el dispositivo se instala como un puerto *Posix* el nombre suele ser `/dev/tty.usbserial-A9009gwg`

A.2. Descarga del firmware

Cuando al programar una realización se elige un periodo de muestreo menor al tiempo de procesamiento el microcontrolador puede llegar a bloquearse debido a la acumulación de interrupciones, siendo imposible ingresar al menú de programación para realizar modificaciones, en ese caso se debe programar el dispositivo con el firmware, para lo cual se emplea la aplicación *ARMWIN.exe* para iniciar el dispositivo en el modo debug se sigue el siguiente procedimiento.

1. Se inicia la aplicación ARMWIN, seleccionando como archivo a descargar *firmware.hex* y se selecciona download.

2. Se mantiene presionado el botón *reset* mientras se presiona una vez el botón *download*

3. Se suelta el bontón de reset.

A.3. Formato para almacenamiento de realizaciones

Se programó la función guardar para Matlab la cual crea un archivo de texto con toda la información relacionada con el controlador Por ejemplo es posible hacer

```
s=tf( s );
guadar(ss(c2d(1/(s^2+s+1),0.1)), nuevo.txt )
```

Que corresponde a la realización

$$H = \left[\begin{array}{cc|c} 0.9706 & 0.08541 & -0.2126 \\ -0.08541 & 0.9247 & -0.2009 \\ \hline -0.2126 & 0.2009 & 0 \end{array}\right]$$

Creandose un archivo de texto como el mostrado a continuación.

```
Multivariable controller implementation plataform--2010-5-1_at_17:59:2.4771
Ni:
1
No:
1
Nx:
2
T1sc:
26113
T1con:
196
Ch1:
252645135
A:
6
0.9706
0.08541
-0.2126
-0.08541
0.9247
-0.2009
C:
3
-0.2126
 0.2009
0
```

Observese que se maneja una etiqueta para indicar el significado del dato e inmediatamente el dato excepto para las matrices en las cuales el primer dato corresponde al numero de elementos, observerse también que se asume $A = \begin{bmatrix} A & B \end{bmatrix}$ y $C = \begin{bmatrix} C & D \end{bmatrix}$, T1sc y T1con son registros que establecen el periodo de muestreo.

Apendice 2

B.2. Manual de usuario

La herramienta *MIMO DEVELOPMENT SYSTEM* es una interfaz *GUI* programada en *MATLAB* para asistir el diseño de sistemas de control lineal para plantas SISO o Multivaribles.

Requisitos: A demás de tener una versión de Matlab *2007a* o posterior se deben tener instalados las siguientes cajas de herramientas.

- Control systems toolbox.
- Robust control toolbox.
- Fixed point toolx

Ejecución: Para ejecutar la aplicación en se deben seguir los siguientes pasos:

- Abrir Matlab
- Entrar al menú *file/set path*
- Agregar la carpeta donde se encuentran los archivos que componen la aplicación.
- Teclear en el *command window* interfaz

El uso de esta herramienta es intuitivo aunque precisa el conocimiento básico de algunos aspectos relacionados con teória de control y sistemas dinamicos lineales.

B.2.1. Modo de uso

Se describirá la interfaz de usuario mediante el diseño de un controlador para un sistema multivariable. El modelo corresponde a un reactor de polimerización a escala indutrial es un modelo académico citado en [46] y [47] y está descrito por la matriz de transferencia:

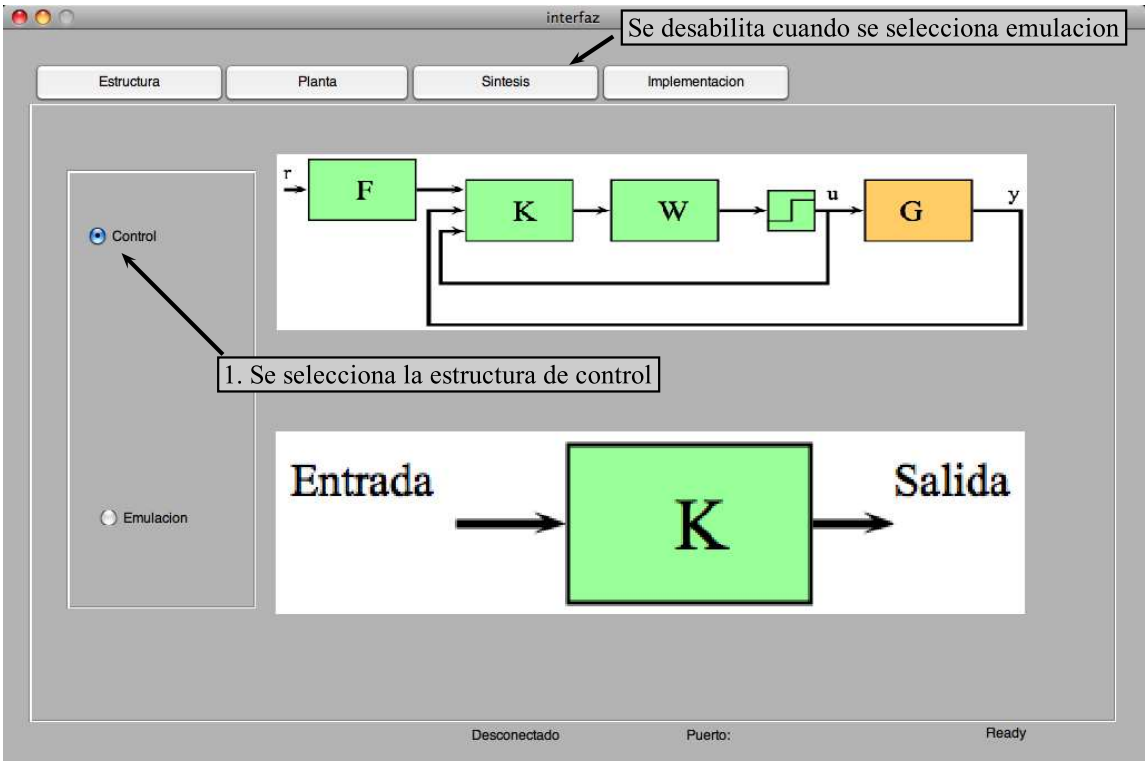
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{22{,}89e^{-0,2s}}{4{,}572s+1} & \frac{-11{,}64e^{0,4s}}{1{,}807s+1} \\ \frac{4{,}689e^{-0,2s}}{2{,}174s+1} & \frac{5{,}80e^{-0,4s}}{1{,}801s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

B.2.2. Definición de la estructura

Definir la estructura consiste en especificar el proposito de la implementación los cuales pueden ser:

- Control: Se define una planta G y se diseña un controlador asignando valores a los bloques W, K y F. Finalmente se implementa un controlador central compuesto por los tres bloques.
- Emulación : Se define un sistema K y se implementa. Cuando se selecciona esta opción se deshabilita el menú de síntesis.

selección de estructura donde se muestra mediante un gráfico la



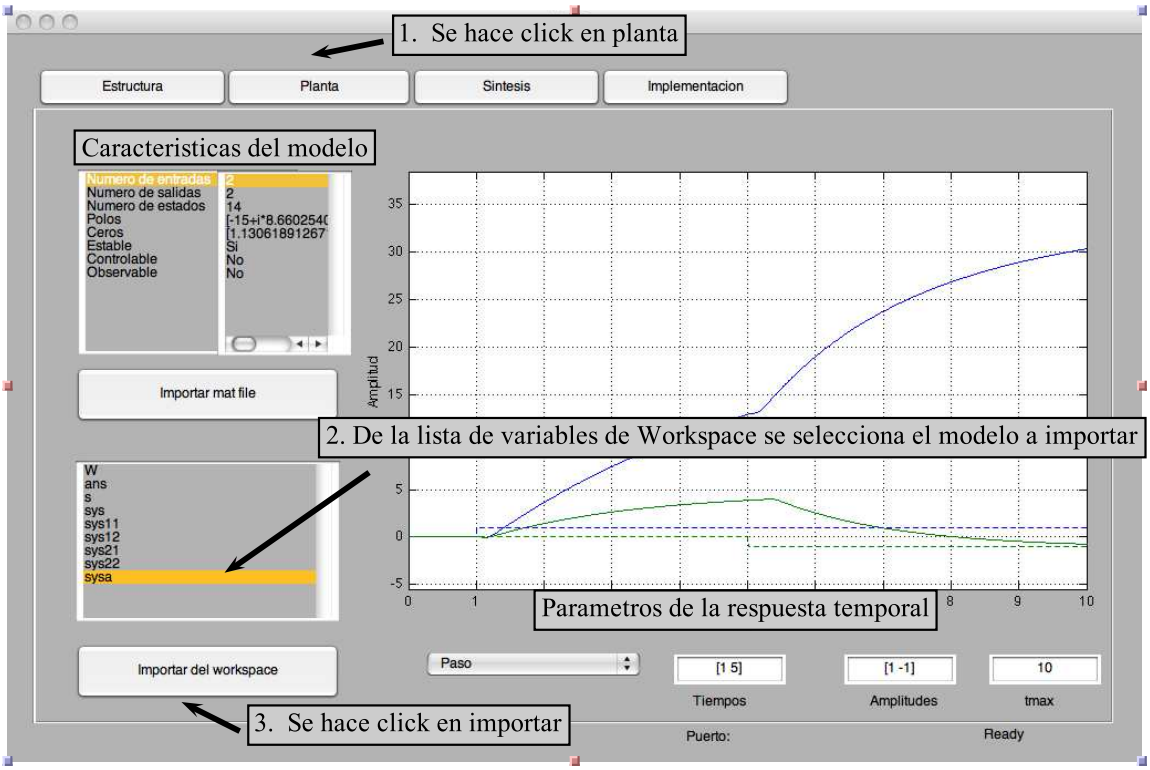
Cuando se realiza emulación se pasa de la definición del sistema directamente a la síntesis sobre el dispositivo hardware de forma que se debe contar con el modelo antes de importarlo a la interfaz debido a que esta no ofrece ninguna herramienta para manipulación o modificación de modelos.

B.2.3. Definición de la planta

La planta puede ser definida en el Workspace definiendo cada función de transferencia independientemente y luego se conforma el sistema multivariable de forma matricial como se muestra en el siguiente código.

```
s=tf( s );
sys11=22.89/(4.572*s+1);     %Sistema siso entrada 1 salida 1
sys11.inputdelay=0.2;        %Se adiciona el retardo
sys12=-11.64/(1.807*s+1);
sys12.inputdelay=0.2;
sys21=4.68/(2.174*s+1);
sys21.inputdelay=0.2;
sys22=5.8/(1.801*s+1);
sys22.inputdelay=0.4;
sys=[sys11,sys12;sys21 sys22]; %planta multivariable
sysa=pade(sys,2);            %Aproximacion de pade de segundo orden
```

Debido a que ninguna función permite el tratamiento de sistemas con retardos puros es necesario trabajar con una aproximación de *Pade* de segundo orden, en la figura se presenta el menú para importar la planta.



Las principales características del panel planta son:

- Permite visualizar la respuesta temporal y frecuencial (gráfico de valores singulares).
- En el caso de la respuesta temporal se visualiza la respuesta al paso siendo posible ajustar el tiempo y amplitud del paso para cada canal.
- Se visualizan características como número de entradas, salidas, estados, polos, ceros.

B.2.4. Diseño

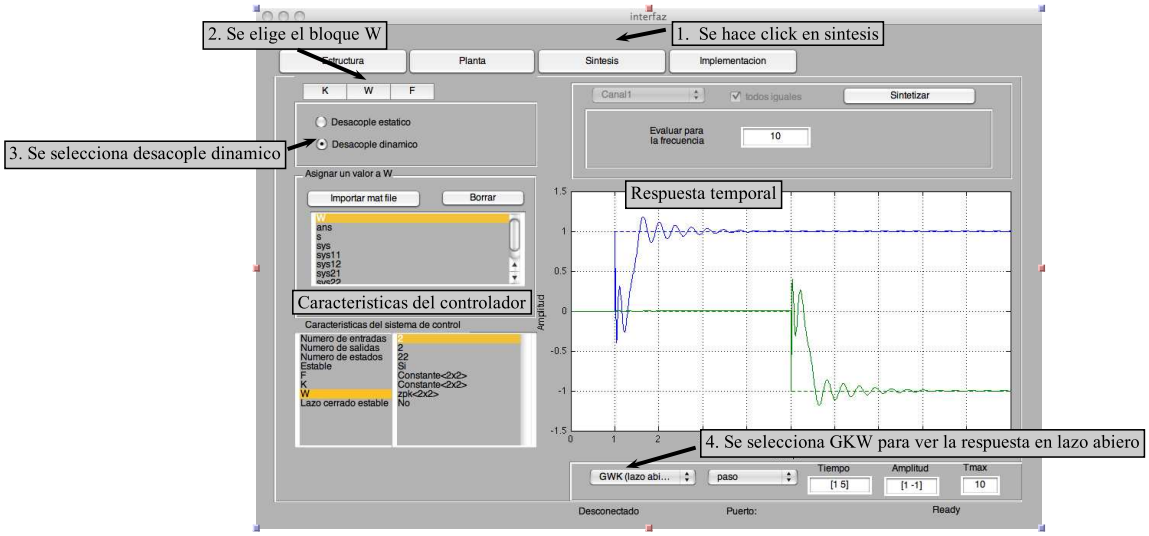
Pulsando la pestaña síntesis se tiene acceso al panel de diseño, inicialmente los bloques K, W y F poseen por defecto matrices identidad que luego pueden ser remplazadas seleccionando el bloque a modificar, eligiendo una estrategia de diseño y realizando la síntesis con el botón síntesis.

Bloque W

Una de las primeras estrategias podría ser el control en lazo abierto mediante una red de desacople dinámico para esto se debe seleccionar la pestaña W y la estrategia desacople dinámico.

En la parte inferior de la interfaz se encuentra un menú desplegable que permite seleccionar la variable a visualizar, es posible elegir entre

- T: Para observar la respuesta de r a y en lazo cerrado
- R: Para observar la respuesta de r a u en lazo cerrado.
- S: Para observar la sensitividad
- GKW : para observar el sistema en lazo abierto.

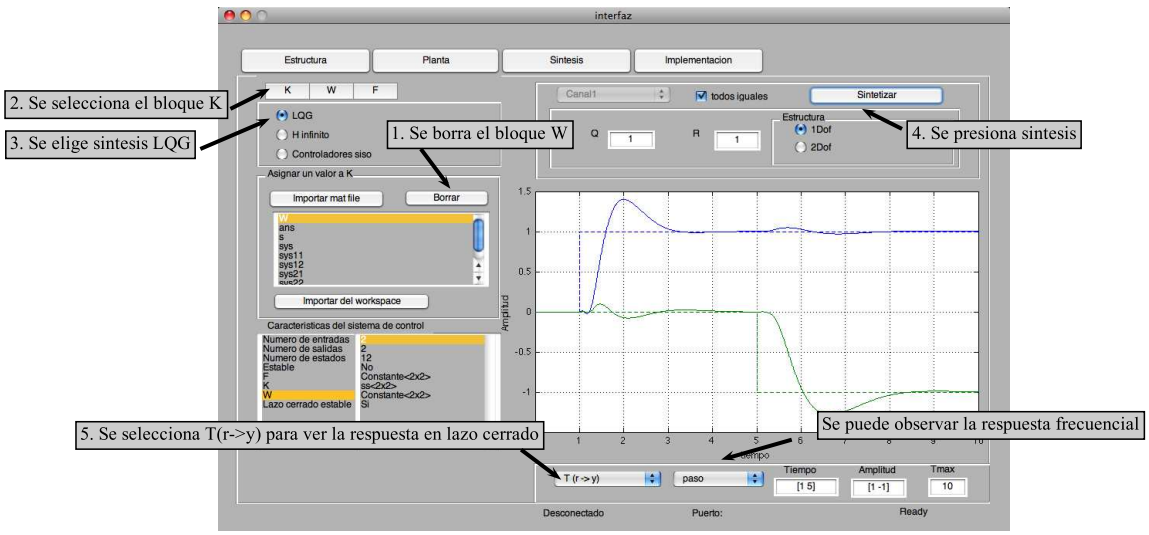


En este caso se seleccionó la respuesta en lazo abierto observandosé que el sistema presenta un comportamiento subamortiguado y el controlador es de orden 22. Ahora se intentara el diseño del bloque K y la red de desacople no se tendrá en cuenta debido a que no resultó ser una buena elección.

Es posible borrar un bloque volviendo a su valor por defecto seleccionando el bloque y pulsando el botón borrar.

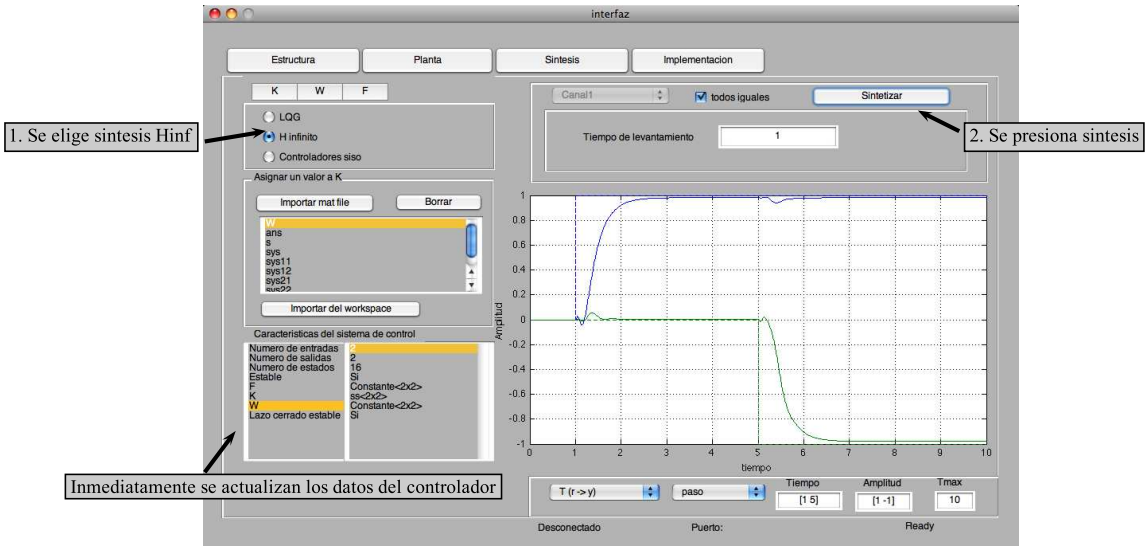
Bloque K

Cuando se hace la síntesis del bloque K es importante tener en cuenta que la síntesis se realiza para la planta desacoplada GW, ahora se selecciona la pestaña de K y se elige la estrategia LQG. Una vez se configuran los parámetros de diseño, (los pesos Q y R) se realiza la síntesis como se observa en la figura.



La respuesta obtenida muestra un comportamiento con baja interacción con sobrepico. Cuando se realiza otra síntesis el valor almacenado es inmediatamente remplazado. Otra posibilidad es realizar la síntesis aplicando un peso diferente para cada canal para lo cual se debe deshabilitar la casilla todos iguales.

Ahora se aplicará la estrategia H_∞ seleccionando un tiempo de establecimiento para lo cual basta con seleccionar la estrategia y luego realizar la síntesis.

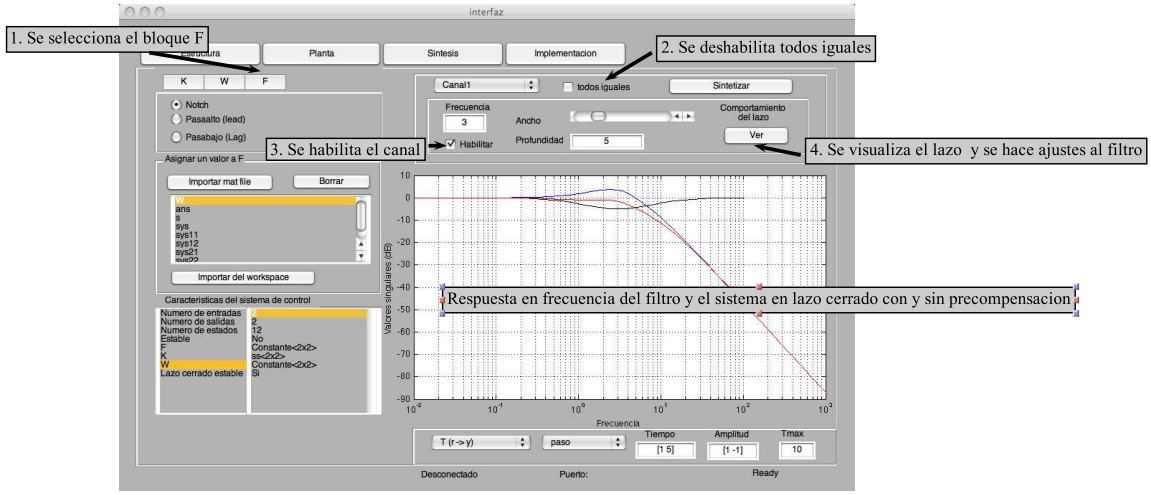


Bloque F

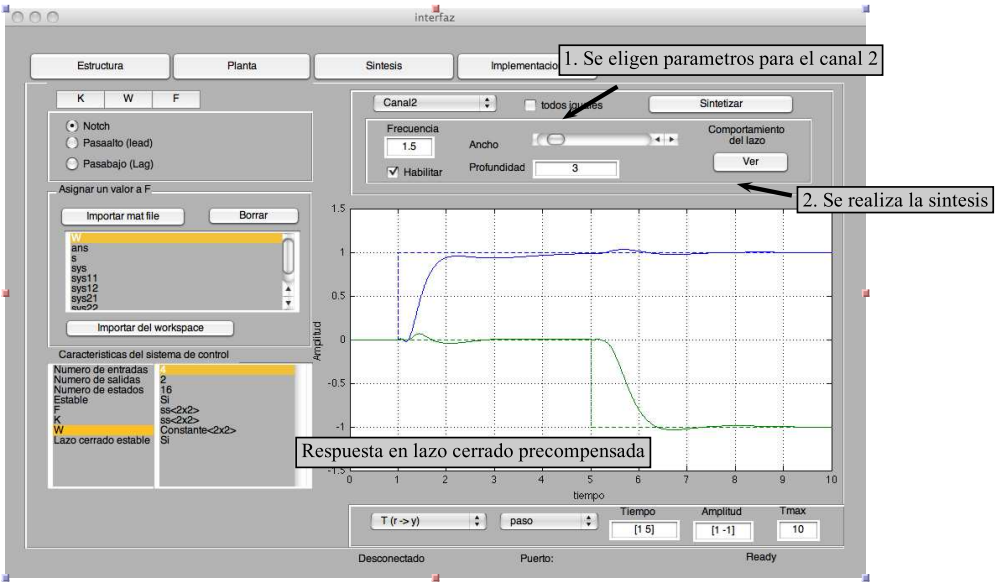
Volviendo al diseño LQG en el cual se presentaban sobrepicos, es posible utilizar el bloque F para ecualizar la respuesta en frecuencia. Se debe seleccionar el bloque F y elegir alguno de los filtros según sea el caso.

Para este ejemplo el sobrepico puede ser eliminado colocando un filtro notch a la frecuencia de resonancia con una profundidad igual a la ganancia máxima. En este punto del diseño se asume que se cuenta con un sistema en lazo cerrado desacoplado de forma que es posible ver el sistema como un conjunto de sistemas *SISO* siendo posible realizar la síntesis del pre-compensador para cada canal. Es necesario habilitar el canal sobre el cual se esta haciendo el proceso de síntesis. Cuando un canal se encuentra deshabilitado guardarda su valor de forma que es posible habilitar un canal para diseñar un filtro notch y luego deshabilitarlo para que no sufra modificaciones al aplicar otros tipos de filtros en los otros canales.

Al pulsar el botón “ver lazo” se visualiza la respuesta frecuencial del filtro, el sistema pre-compensado y sin pre-compensar. En la figura se muestra la respuesta frecuencial del canal uno con el controlador LQG, la elección de los parámetros del filtro recae en el diseñador.



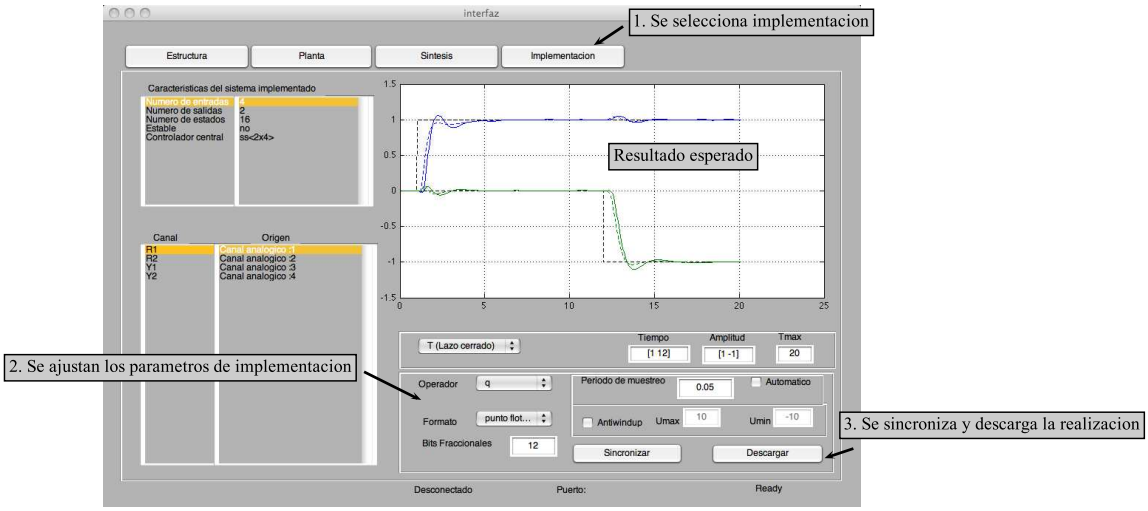
Una vez se seleccionaron los parámetros adecuados para cada canal se realiza la síntesis obteniendo una respuesta como la mostrada en la figura.



B.2.5. Implementación

Una vez se tiene definido el sistema o controlador se debe ser discretizar y enviar al dispositivo hardware. Se cuenta con un panel que permite la simulación del sistema en lazo cerrado sujeto a saturaciones, cuantización y redondeo. Como opciones para la implementación es posible elegir

- El periodo de muestreo
- El formato numérico
- El operador
- Para implementaciones en punto fijo es posible elegir la longitud de la parte fraccional.



B.2.6. Recursos en línea

Se grabó un total de 6 video tutoriales los cuales se encuentran disponibles en el servidor de *youtube* y se describen a continuación.

- Emulación de sistemas: Este video decribe los páneles de estructura y muestra como importar sistemas y simular su comportamiento sobre el dispositivo hardware.
  `http://www.youtube.com/watch?v=Ia07roKLJ4c`

- Diseño multivariable parte 1: Se decribe el proceso de diseño e implementación para un sistema 2×2 para configuración de un grado de libertad sin red de desacople
  `http://www.youtube.com/watch?v=vMMkFUJnTvg`

- Diseño multivariable parte 2: Se muestra el procedimiento para la síntesis de un precompensador para un sistema 2×2 teniendo como objetivo la reducción del sobrepico de un canal y el aumento el ancho de banda del otro.
  `http://www.youtube.com/watch?v=tEy4yjLU5HQ`

- Diseño multivariable parte 3: Se presenta la síntesis H_∞ y sus características.
  `http://www.youtube.com/watch?v=vfz_PYRuLyM`

- Diseño multivariable parte 4: Se diseña un sistema con red de desacople, compensador y precompensador.
  `http://www.youtube.com/watch?v=p-uYYhpeobM`

- Diseño de controladores: Se aborda el diseño de controladores para plantas SISO y plantas multivariables rectangulares.
  `http://www.youtube.com/watch?v=cNje1EDvMgg`

Bibliografía

[1] R. Brockett, "Poles, zeros, and feedback: State space interpretation," *Automatic Control, IEEE Transactions on*, vol. 10, no. 2, pp. 129–135, 1965.

[2] W. Wonham, "On pole assignment in multi-input controllable linear systems," *Automatic Control, IEEE Transactions on*, vol. 12, no. 6, pp. 660–665, 1967.

[3] J. Pearson and C. Ding, "Compensator design for multivariable linear systems," *Automatic Control, IEEE Transactions on*, vol. 14, no. 2, pp. 130–134, 1969.

[4] I. Horowitz, "Synthesis of linear, multivariable feedback control systems," *Automatic Control, IRE Transactions on*, vol. 5, no. 2, pp. 94–105, 1960.

[5] I. Horowitz, "Design of multiple-loop feedback control systems," *Automatic Control, IRE Transactions on*, vol. 7, no. 3, pp. 47–57, 1962.

[6] E. Bristol, "On a new measure of interaction for multivariable process control," *Automatic Control, IEEE Transactions on*, vol. 11, no. 1, pp. 133–134, 1966.

[7] C. Liu and N. Bergman, "On necessary conditions for decoupling multivariable control systems," *Automatic Control, IEEE Transactions on*, vol. 15, no. 1, pp. 131–133, 1970.

[8] A. MacFarlane, "The development of frequency-response methods in automatic control," *Automatic Control, IEEE Transactions on*, vol. 24, no. 2, pp. 250–265, 1979.

[9] M. Safonov and M. Athans, "Gain and phase margin for multiloop LQG regulators," *Automatic Control, IEEE Transactions on*, vol. 22, no. 2, pp. 173–179, 1977.

[10] P. Paraskevopoulos, "On the Design of PID Output Feedback Controllers for Linear Multivariable Systems," *Industrial Electronics and Control Instrumentation, IEEE Transactions on*, pp. 16–18, 1980.

[11] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *Automatic Control, IEEE Transactions on*, vol. 25, no. 2, pp. 164–176, 1980.

[12] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *Automatic Control, IEEE Transactions on*, vol. 26, no. 1, pp. 17–32, 1981.

[13] I. Postlethwaite, J. Edmunds, and A. MacFarlane, "Principal gains and principal phases in the analysis of linear multivariable feedback systems," *Automatic Control, IEEE Transactions on*, vol. 26, no. 1, pp. 32–46, 1981.

[14] L. Pernebo, "An algebraic theory for design of controllers for linear multivariable systems–Part I: Structure matrices and feedforward design," *Automatic Control, IEEE Transactions on*, vol. 26, no. 1, pp. 171–182, 1981.

[15] L. Pernebo, "An algebraic theory for design of controllers for linear multivariable systems–Part II: Feedback realizations and feedback design," *Automatic Control, IEEE Transactions on*, vol. 26, no. 1, pp. 183–194, 1981.

[16] J. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *Automatic Control, IEEE Transactions on*, vol. 26, no. 1, pp. 4–16, 1981.

[17] J. Wu, S. Chen, G. Li, and J. Chu, "Optimal controller realisation with the smallest dynamic range," 2006.

[18] P. Moroney, A. Willsky, and P. Houpt, "The digital implementation of control compensators: The coefficient wordlength issue," *Automatic Control, IEEE Transactions on*, vol. 25, no. 4, pp. 621–630, 1980.

[19] D. Viassolo and M. Rotea, "Optimal scaling of digital controllers," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions*, vol. 46, no. 4, pp. 500–505, 1999.

[20] J. Slaughter, "Quantization errors in digital control systems," *Automatic Control, IEEE Transactions on*, vol. 9, no. 1, pp. 70–74, 1964.

[21] D. W. Mario Rotea, "Optimal realizations of finite wordlength digital filters and controllers," *IEEE transactions on circuits and systems*, vol. 42, pp. 61–72, February 1995.

[22] D. Viassolo, "Implementation of Digital Controllers," *Master's thesis, Purdue University, USA*, 1996.

[23] J. Wu, S. Chen, G. Li, and J. Chu, "Digital finite-precision controller realizations with sparseness-considerations," *Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress on*, vol. 4, 2000.

[24] J. Hao and G. Li, "Optimal/spl rho/-realisation for digital controller implementation with stability consideration," *Electronics Letters*, vol. 40, no. 4, pp. 226–227, 2004.

[25] P. Lundstrom, S. Skogestad, and J. Doyle, "Two-degree-of-freedom controller design for an ill-conditioneddistillation process using μ-synthesis," *Control Systems Technology, IEEE Transactions on*, vol. 7, no. 1, pp. 12–21, 1999.

[26] V. U. V. A P Loh, "Multiloop Controller Design for Multivariable Plants," *Proceedings of the 31 st Conference on Decision and Control*, pp. 181–182, 1992.

[27] L. Bermeo, "Sistema de desarrollo de controladores lineales siso sobre microcontrolador embebido," *Master's thesis, Universidad Nacional de Colombia*, 2003.

[28] W. Tan, H. Marquez, and T. Chen, "Multivariable robust controller design for a boiler system," *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 5, pp. 735–742, 2002.

[29] A. C. D. Y. Zhang, L Shieh, "PID controller design for disturbed multivariable systems," *Proc. Control Theory Appl.*, vol. 151, no. 5, pp. 567–577, 2004.

[30] B. C. C. Daniel Chunlong Hu, "Multivariable Controller Design for Hard Drive Dual-stage Actuator Servo Systems," *International conference on mechatronics*, pp. 404–410, 2005.

[31] R. Chiang and M. Safonov, "MATLAB Robust Control Toolbox,"

[32] K. Astrom and K. Wang, "Design of decoupled PID controllers for MIMO systems," *American Control Conference, 2001. Proceedings of the 2001*, vol. 3, 2001.

[33] P. Albertos and A.Sala, *Multivariable Control System: An Engineering Approach*. Springer, August 2003.

[34] R. Kalman, "Mathematical description of linear dynamical systems," *Siam*, vol. 1, no. 2, pp. 152–192, 1963.

[35] D. Luenberger, "Observers for multivariable systems," *Transactions on Automatic Control*, vol. AC-11, pp. 190–198, April 1966.

[36] P. A. Gahinet P, "A linear matrix inequality approach to h_∞ control," *International Journal on Robust and Nonlinear Control*, pp. 656–661, 1993.

[37] A. I. Pascal Gahinet, "Low order h_∞ synthesis via lmis," *American control conference*, pp. 1499–1450, June 1994.

[38] D. L. Michael Green, *Linear Robust Control*. Pearson Education, 1995.

[39] K. Astrom and B. Wittenmark, *Computer controlled systems*. Prentice Hall Englewood Cliffs, NJ, 1990.

[40] L. D. Green M, *Linear Robust Control*. Prentice Hall, 1995.

[41] C. Chen, *Analog and Digital Control System Design: Transfer-function, State-space, and Algebraic Methods*. Oxford University Press, Inc. New York, NY, USA, 1995.

[42] J. Wu, S. Chen, J. Whidborne, and J. Chu, "Optimal floating-point realizations of finite-precision digital controllers," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, 2002.

[43] D. Luenberger, "Canonical forms for linear multivariable systems," *Automatic Control, IEEE Transactions on*, vol. 12, no. 3, pp. 290–293, 1967.

[44] K. Tomas, *Linear System Theory*, vol. 1. Prentice Hall, 1980.

[45] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. Wiley, 1996.

[46] D. Chen and D. Seborg, "Multiloop PI/PID controller design based on Gershgorin bands," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 5, 2001.

[47] Y. Zhang, L. Shieh, and A. Dunn, "PID controller design for disturbed multivariable systems," *IEE Proceedings-Control Theory and Applications*, vol. 151, no. 5, pp. 567–576, 2004.

[48] O. Duarte, *Análisis de sistemas dinámicos*. Universidad Nacional de Colombia, 2006.