

**DISEÑO DE UNA INTERFAZ GUI EN MATLAB PARA EL CALCULO DE UN
CONTROL MULTIVARIABLE**

PABLO JOSÉ PEÑUELA ROJAS

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2007**

**DISEÑO DE UNA INTERFAZ GUI EN MATLAB PARA EL CALCULO DE UN
CONTROL MULTIVARIABLE**

PABLO JOSÉ PEÑUELA ROJAS

**Trabajo de grado presentado como requisito para
optar al título de Ingeniero Electrónico**

Director

AGUSTÍN SOTO OTÁLORA

Ingeniero Electrónico

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

NEIVA

2007

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Neiva, 30 de mayo de 2007

A mi Mamá, que con su esfuerzo ha sido mi apoyo y mi guía, a mis viejos, José Ángel y Rossana, que siempre han estado para mí, a Sergio, que me brinda la alegría y esperanza, a mis tíos, Juanito y Kike, que siempre han sido la voz de apoyo, a Ernesto, que ha sido una fuente de inspiración, a Cristina, que ha estado siempre ahí y a todos ellos que con su sonrisa han alumbrado mi camino, les doy las gracias por estar junto a mí en el recorrido de este sueño.

Pablo José.

AGRADECIMIENTOS

El autor expresa su agradecimiento:

A Dios

Al ingeniero Agustín Soto Otálora, Director del proyecto

A los profesores y compañeros, quienes me han acompañado en el transcurso de mis estudios.

CONTENIDO

INTRODUCCIÓN	10
1. MARCO TEÓRICO	11
1.1 FUNCIONES DE TRANSFERENCIA EN SISTEMAS MIMO	11
1.2 SISTEMAS DE CONTROL CON REALIMENTACIÓN NEGATIVA	13
1.3 RESPUESTA EN FRECUENCIA DE SISTEMAS MULTIVARIABLE	14
1.4 DIRECCIONES EN SISTEMAS MULTIVARIABLE	16
1.5 DESCOMPOSICIÓN DE VALORES SINGULARES	17
1.5.1 DIRECCIONES DE ENTRADA Y SALIDA	17
1.5.2 MÁXIMO Y MÍNIMO VALOR SINGULAR	18
1.5.3 USO DEL MÍNIMO VALOR SINGULAR	20
1.5.4 VALORES SINGULARES PARA EL DESEMPEÑO	20
2. CONTROL DE PLANTAS MULTIVARIABLE	24
2.1 CONTROLADOR DIAGONAL	24
2.2 LAZO DEL “MEJOR” CONTROLADOR	25
3. SÍNTESIS DE CONTROL MULTIVARIABLE	26
3.1 DISEÑO DEL LAZO (DISEÑO H_{∞})	26
3.2 CONTROL LQG	29
3.3 ROBUSTEZ	33
3.4 DISEÑO DE SENSIBILIDAD MIXTA (H_{∞})	35
4. APLICACIÓN MATLAB	38
4.1 DISEÑO DEL LAZO	42
4.2 LQG	47
4.3 CONTROL ROBUSTO	50
4.4 SENSIBILIDAD MIXTA H_{∞}	53
5. CÓDIGO DE MATLAB	56
6. CONCLUSIONES	65
BIBLIOGRAFIA	66
ANEXOS	67

LISTA DE FIGURAS

Figura 1: Diagrama de bloques (a) en serie y (b) con realimentación.....	12
Figura 2: Sistema de control convencional con realimentación negativa.....	13
Figura 3: Sistema $G(s)$ con entrada d y salida y	14
Figura 4: Gráficas típicas de valores singulares	19
Figura 5: Relaciones de diseño en lazo cerrado.....	22
Figura 6: Configuración de control por realimentación.....	24
Figura 7: Lazo diseñado y controlador.....	26
Figura 8: Modelo con Incertidumbre.....	26
Figura 9: Controlador LQG y planta con ruido	31
Figura 10: Forma estándar de la optimización de sensibilidad mixta	36
Figura 11: Configuración del péndulo invertido sobre un columpio	39
Figura 12: Forma para establecer la carpeta con los archivos en el "Path"	40
Figura 13: Vista del aplicativo de MATLAB.....	41
Figura 14: Pasos a seguir para manejo del aplicativo	42
Figura 15: Respuesta Paso de la planta controlada (Diseño del Lazo).....	46
Figura 16: SVD de la planta controlada (Diseño del Lazo)	46
Figura 17: Pasos a seguir para diseño del controlador LQG	47
Figura 18: Respuesta Paso de la planta controlada (LQG)	49
Figura 19: SVD de la planta controlada (LQG).....	49
Figura 20: Pasos a seguir para diseño de Control Robusto	50
Figura 21: Respuesta Paso de la planta controlada (Control Robusto).....	51
Figura 22: SVD de la planta controlada sin y con incertidumbre (Control Robusto).....	52
Figura 23: Pasos a seguir para diseño de Sensibilidad Mixta Hinf	53
Figura 24: Respuesta Paso de la planta controlada (Sensibilidad Mixta Hinf) ...	55
Figura 25: SVD de la planta controlada (Sensibilidad Mixta Hinf).....	55

RESUMEN

Nuevas tendencias en temas como control y automatización de procesos están siendo introducidas en el mundo, estas tendencias en su gran mayoría están basadas en la teoría de Control Clásico, pero difieren mucho en su aplicación final. El control Clásico es solo un pequeño tema en todo lo que tiene que ver con la teoría de Control, el Control Moderno es que precede al control Clásico. En nuestro caso, estamos interesados en el Control Multivariable porque es un acercamiento más realista acerca de que realmente sucede en un proceso real. Aunque el Control Clásico es una teoría que solo explica eventos que solo depende de una variable física y puede ser notada como el fundamento de toda la teoría de control, por lo tanto es siempre necesitada.

El Control Moderno y el Control Multivariable son tema que ahora son de nuestro interés. En el mundo real ni un solo proceso está determinado por una sola fuerza (variable), esto quiere decir que, si queremos entender como un proceso trabaja y como es afectado por sus alrededores necesitamos entender las interacciones en un proceso o planta, en el mismo momento que esto es entendido se necesita comprender como controlar dicho proceso.

En este proceso se hará una introducción en Control Moderno y en Sistemas Multivariable, también conocidos como Sistemas MIMO (Multiple Input – Multiple Output), para entender sus dinámicas y comportamientos. Una vez que esta introducción sea realizada, procedemos a estudiar el Control Centralizado para plantas MIMO para así sintetizar (diseñar) un controlador apropiado para dicha planta.

Todo esto será hecho utilizando las herramientas provistas por MATLAB y algunos de sus “Toolbox” tales como el “Toolbox” de Control, el “Toolbox” de Control Robusto, el “Toolbox” de μ análisis y síntesis y otros. Finalmente, todas estas herramientas serán reunidas en otra herramienta de MATLAB.

La Interfaz Gráfica de Usuario (GUI) será utilizada para ser la interface entre la teoría de control, los procedimientos matemáticos y el usuario que quiera sintetizar un controlador para su planta MIMO.

Diferentes tipos de controladores serán sintetizados para dar diferentes opciones y mostrar las diferencias entre uno y otro, esto resulta en una visión más amplia acerca de lo que es el Control Moderno de Plantas MIMO.

Más allá de sintetizar un controlador, será permitido para el usuario ver las respuestas de los procesos controlados con curvas como la “Respuesta Paso” y la “Descomposición de Valores Singulares SVD”.

Todo esto es un esfuerzo para reunir herramientas con facilidad de uso para estudiantes y profesionales que estén interesados en temas como Control de Plantas MIMO y otros temas similares.

ABSTRACT

New trends in subjects like process control and process automation are being introduced in the world, these trends are most based in classical control theory but they differ a lot in the final application. Classical control is just a little issue in all that matters in control theory, Modern control proceeds the Classical Control. In our case, we are interested in the Multivariable Control because it is a more realistic approach about what really happens in any real process.

Although Classical control is a theory that only explains events that only depends on one physic variable and it can be noted as the fundamentals of all control theory, hence it is always needed.

Modern Control and the Multivariable Control are the topics that now are our concern. In real world not a single process is determined by only one force (variable), that means that, if we want to understand how a process works and how it is affected by its surroundings we need to understand the physics interactions on the process or plant, in the very moment that this is understood it is needed to comprehend how to control such a process.

In this work will be done and introduction in Modern Control and in the Multivariable Systems, also known as MIMO systems (Multiple Input – Multiple Output), prior to understand its dynamics and behavior. Once this introduction is done, we proceed to study centralized Control for MIMO plants in order to synthesize (design) and appropriate controller for such a plant.

All of this will be done by using the tools provided by MATLAB and some of its toolboxes such Control Toolbox, Robust Control Toolbox, μ -Analysis and synthesis Toolbox and others. Finally, all this tools will be gathered in other MATLAB Tool. Graphical User Interface (GUI) will be used to be the interface between the control theory, mathematic procedures and the user who wants to synthesize a controller for his or her MIMO plant.

Different types of Controllers will be synthesized to give different options and to show the differences between one and another, this yields in a broader vision about what Modern Control of MIMO plants is.

Furthermore about synthesizing a controller, it will be also allowed for the user to see responses of controlled processes with plots like the “Step Response” and the “Singular Value Decomposition SVD”.

All of this is an effort to bring tools with ease of use for students and professionals that are interested in topics like Controlling MIMO Plants and other similar topics.

INTRODUCCIÓN

Como objetivo tenemos en este documento el de realizar un acercamiento al control de las plantas multivariable, más conocidas como plantas MIMO por sus siglas en inglés Multiple Input – Multiple Output. El control de las plantas MIMO hace parte de la teoría moderna de control, la cual se basa en el control clásico de sistemas SISO (Single Input – Single Output) para desarrollar las teorías que permitan diseñar e implementar un controlador para este tipo de procesos.

Es de suma importancia entender que la teoría de control moderno se fundamenta totalmente en la teoría de control clásica, pues a pesar de haberse logrado nuevas técnicas y teorías aun se siguen aplicando conceptos como polos y ceros de una planta (modos). En este trabajo se ha deseado hacer énfasis en el control centralizado de las plantas, el cual consiste de un solo controlador para una sola planta a diferencia del control descentralizado que consiste en enfocar sus estudios en un desacoplador de entradas y salidas de la planta en cuestión y una vez diseñado este desacoplador, se procede a diseñar un controlador PID para cada par de entrada – salidas identificadas.

Como herramienta para desarrollar esta labor no hemos apoyado en MATLAB y en sus librerías de control, las cuales poseen herramientas específicas para el desarrollo o diseño de controles centralizados.

También aprenderemos nuevas herramientas como la descomposición de valores singulares (SVD) y el arreglo de ganancias relativas de una planta, las cuales serán fundamentales para plantear los objetivos de control y así mismo diseñar el controlador en cuestión.

Se presentarán a continuación cuatro técnicas para el diseño o síntesis de controladores para plantas MIMO, todo esto por medio de una interface de usuario utilizando el GUIDE Toolbox de MATLAB; con el fin de brindar herramientas para aquellos que deseen incursionar en el mundo de la teoría de control moderno en conjunto con el uso de herramientas avanzadas de cálculo para su aplicación. Desafortunadamente no es posible entrar en los pormenores del extenso mundo MIMO pero se da una clara explicación de sus fundamentos, los cuales servirán de base para futuros conocimiento más precisos y puntuales del tema.

1. MARCO TEÓRICO

Ahora se considerará una planta MIMO (múltiple entrada – múltiple salida) con una cantidad de entradas m y salidas l . De esto tenemos que la función de transferencia básica es $Y(s) = G(s)u(s)$, donde $Y(s)$ es un vector de tamaño $l \times 1$, $u(s)$ es un vector de tamaño $m \times 1$ y $G(s)$ es una matriz de función de transferencia de tamaño $l \times m$.

Si hacemos un cambio en la primera entrada, u_1 , entonces esto generalmente afectará a todas las salidas, y_1, y_2, \dots, y_l , de esto tenemos que existe una interacción entre las entradas y las salidas. Una planta que interactúa es aquella donde una entrada u_1 solo afecta a la salida y_1 , u_2 a la salida y_2 , y así sucesivamente.

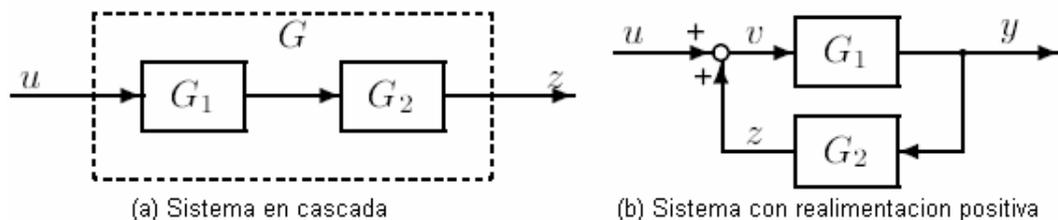
La principal diferencia entre un sistema escalar SISO y un sistema MIMO es la presencia de direcciones en este último. Las Direcciones son relevantes para vectores y matrices, no para escalares. Sin embargo, a pesar del factor de Direccionalidad, muchas de las teorías aplicadas a sistemas SISO pueden ser aplicadas a sistemas MIMO.

La descomposición de valores singulares ó SVD (por sus siglas en inglés) provee un método útil para cuantificar la direccionalidad multivariable.

Más adelante se mostrarán algunas reglas para determinar funciones de transferencia multivariable, aunque la mayoría de las formulas para sistemas escalares es aplicable, a pesar de ellos debemos tener en cuenta que la multiplicación en el caso multivariable no es conmutativa. Además se observará la descomposición de valores singulares y se verá cómo se puede utilizar para estudiar las direcciones en sistemas multivariable y se analizará la robustez de un proceso multivariable

1.1 FUNCIONES DE TRANSFERENCIA EN SISTEMAS MIMO

Para poder definir funciones de transferencia para bloques MIMO podemos evaluar tres reglas:



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 1: Diagrama de bloques (a) en serie y (b) con realimentación

CONEXIÓN SERIE: Para una interconexión serie (Cascada) de G_1 y G_2 como se muestra en la Figura 1(a), la función de transferencia total es $G = G_2G_1$.

CONEXIÓN CON REALIMENTACIÓN: Con referencia al sistema de realimentación positiva en la Figura 1(b), tenemos que:

$$\begin{aligned}y &= G_1v & z &= G_2y \\v &= u + z \\v &= u + G_2y \\v &= u + G_2G_1v \\v - G_2G_1v &= u \\(I - G_2G_1)v &= u \\v &= (I - L)^{-1}u\end{aligned}$$

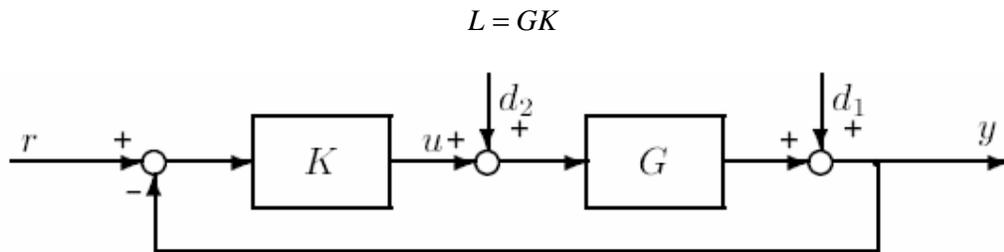
donde $L = G_2G_1$ es la función de transferencia alrededor del lazo y I es la matriz identidad (Ver Anexo A).

EQUIVALENCIA: Para matrices de dimensiones apropiadas (la división y multiplicación de matrices no es conmutativa y las matrices deben tener dimensiones apropiadas para que puedan ser sometidas a estas operaciones) tenemos:

$$G_1(I - G_2G_1) = (I - G_1G_2)G_1$$

1.2 SISTEMAS DE CONTROL CON REALIMENTACIÓN NEGATIVA

Para la Figura 2 del sistema de realimentación negativa, definimos L como la función de transferencia del lazo, para el caso cuando el lazo consta de una planta G y un controlador de realimentación K tenemos:



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 2: Sistema de control convencional con realimentación negativa

Considerando los sistemas dinámicos se pueden derivar relaciones fundamentales que describen propiedades de todo el sistema. Estas relaciones dependen de las propiedades dinámicas y estructurales de la planta (polos y ceros de la función de transferencia) y están expresadas en términos de funciones (de sensibilidad y sensibilidad complementaria) que cuantifican el desempeño de todo el sistema (especifican la cantidad de cambio de la variable de salida por unidad de cambio en la variable de entrada).

La Sensibilidad y la Sensibilidad Complementaria están definidas como:

$$S = (I + L)^{-1}; T = I - S = L(I + L)^{-1}$$

En la Figura 2, T es la función de transferencia de r hasta y , y S es la función de transferencia de d_1 .

S y T también son llamados sensibilidad de salida y sensibilidad complementaria de salida, respectivamente.

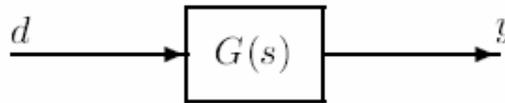
Además tenemos algunas relaciones que son útiles:

$$\begin{aligned} (I + L)^{-1} + L(I + L)^{-1} &= S + T = I \\ G(I + GK)^{-1} &= (I + GK)^{-1}G \\ GK(I + GK)^{-1} &= G(I + GK)^{-1}K = (I + GK)^{-1}GK \\ T &= L(I + L)^{-1} = (I + (L)^{-1})^{-1} \end{aligned}$$

Hay que notar que las matrices G y K NO necesitan ser cuadradas, en cambio, $L = GK$ sí es cuadrada.

1.3 RESPUESTA EN FRECUENCIA DE SISTEMAS MULTIVARIABLE

La función de transferencia $G(s)$ es una función de la variable de Laplace s y puede ser utilizado para representar un sistema dinámico. Sin embargo, si se fija $s = s_0$ entonces veremos $G(s_0)$ simplemente como una matriz compleja, la cual puede ser analizada utilizando herramientas estándares en el algebra de matrices. En particular, la elección de $s_0 = j\omega$ es de interés cuando $G(j\omega)$ representa la respuesta a una señal sinusoidal de frecuencia ω .



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 3: Sistema $G(s)$ con entrada d y salida y

El dominio de frecuencias es ideal para estudiar direcciones en sistemas multivariable. Considerando el sistema de la Figura 3 con entrada $d(s)$ y salida $y(s)$

$$y(s) = G(s)d(s)$$

Si consideramos los elementos $g_{ij}(j\omega)$ de la matriz $G(j\omega)$, los cuales representan la respuesta sinusoidal desde la entrada j hasta la salida i , es decir, se aplica a la entrada del canal j una señal sinusoidal escalar dada por

$$d_j(t) = d_{j0} \text{sen}(\omega t + \alpha_j)$$

Esta señal es persistente, es decir, es aplicada desde $t = -\infty$. La señal de salida persistente correspondiente en el canal i es también una señal sinusoidal con la misma frecuencia

$$y_i(t) = y_{i0} \text{sen}(\omega t + \alpha_i)$$

Donde la ganancia y el cambio de fase pueden ser obtenidos del número complejo como sigue

$$\frac{y_{i0}}{d_{j0}} = |g_{ij}(j\omega)|, \quad \beta_i - \alpha_j = \angle g_{ij}(j\omega)$$

En notación fasorial se tiene

$$d_j(\omega) = d_{j0}e^{j\alpha}, y_i(\omega) = y_{i0}e^{j\beta}$$

Aquí el uso de ω (y no $j\omega$) como argumento de $d_j(\omega)$ y $y_i(\omega)$ implica que son números complejos, representando a cada frecuencia de ω la magnitud y la fase de las señales sinusoidales de la entrada y la salida.

La respuesta total del sistema a entradas simultáneas en varios canales a la misma frecuencia es, igual a la suma de las respuestas individuales, por principio de superposición

$$y_i(\omega) = g_{i1}(j\omega)d_1(\omega) + g_{i2}(j\omega)d_2(\omega) + \dots = \sum_j g_{ij}(j\omega)d_j(\omega)$$

O en forma matricial

$$y(\omega) = G(j\omega)d(\omega)$$

Donde se representan los vectores de las entradas y salidas sinusoidales

$$d(\omega) = \begin{bmatrix} d_1(\omega) \\ d_2(\omega) \\ \vdots \\ d_j(\omega) \\ \vdots \\ d_m(\omega) \end{bmatrix} \quad \Upsilon \quad y(\omega) = \begin{bmatrix} y_1(\omega) \\ y_2(\omega) \\ \vdots \\ y_i(\omega) \\ \vdots \\ y_l(\omega) \end{bmatrix}$$

1.4 DIRECCIONES EN SISTEMAS MULTIVARIABLE

Las cosas no son muy sencillas para sistemas MIMO donde ambos, la entrada y la salida son ambos vectores, como es necesario hacer la suma de los elementos en el vector, utilizamos la norma a 2, entonces a una frecuencia ω dada la magnitud del vector de entrada es

$$\|d(\omega)\|_2 = \sqrt{\sum_j |d_j(\omega)|^2} = \sqrt{d_{10}^2 + d_{20}^2 + \dots}$$

Y la magnitud del vector de salida es

$$\|y(\omega)\|_2 = \sqrt{\sum_j |y_j(\omega)|^2} = \sqrt{y_{10}^2 + y_{20}^2 + \dots}$$

La ganancia del sistema $G(s)$ para una señal de entrada $d(\omega)$ en particular está dada por la relación

$$\frac{\|y(\omega)\|_2}{\|d(\omega)\|_2} = \frac{\|G(j\omega)d(\omega)\|_2}{\|d(\omega)\|_2} = \frac{\sqrt{y_{10}^2 + y_{20}^2 + \dots}}{\sqrt{d_{10}^2 + d_{20}^2 + \dots}}$$

Aquí se puede notar que la ganancia de un sistema depende de la frecuencia ω y es independiente de la magnitud de la entrada $\|d(\omega)\|_2$. Sin embargo para un sistema MIMO hay adicionales grados de libertad y la ganancia depende también en la dirección de la entrada d .

El máximo valor de ganancia mientras varía la dirección de la entrada, es el máximo valor singular de G ,

$$\bar{\sigma}(G) = \max_{d \neq 0} \frac{\|Gd\|_2}{\|d\|_2} = \max_{\|d\|_2=1} \|Gd\|_2$$

Donde la mínima ganancia es el mínimo valor singular de G ,

$$\underline{\sigma}(G) = \min_{d \neq 0} \frac{\|Gd\|_2}{\|d\|_2} = \min_{\|d\|_2=1} \|Gd\|_2$$

1.5 DESCOMPOSICIÓN DE VALORES SINGULARES

Considerando una frecuencia fija ω donde $G(j\omega)$ es una matriz compleja constante de tamaño $l \times m$, y se denota $G(j\omega)$ como G . Cualquier matriz G puede ser descompuesta en sus valores singulares y se escribe de la siguiente manera

$$G = U \Sigma V^H$$

Donde:

Σ es una matriz de $l \times m$ con valores singulares no negativos $k = \min\{l, m\}$, σ_i , ordenados en orden descendente a lo largo de su diagonal principal; los otros valores son iguales a cero. Los valores singulares son las raíces cuadradas de los valores propios de $G^H G$, donde G^H es la transpuesta conjugada compleja de G .

$$\sigma_i(G) = \sqrt{\lambda_i(G^H G)}$$

U es una matriz unitaria de $l \times l$ de los vectores de salida singulares, u_i .

V es una matriz unitaria de $m \times m$ de los vectores de entrada singulares, v_i .

1.5.1 DIRECCIONES DE ENTRADA Y SALIDA

Los vectores columna de U representan las direcciones de salida de la planta. Son ortogonales y de magnitud unitaria, esto es

$$\|u_i\|_2 = \sqrt{|u_{i1}|^2 + |u_{i2}|^2 + \dots + |u_{il}|^2} = 1$$

$$u_i^H u_i = 1, u_i^H u_j = 0, i \neq j$$

Asimismo, los vectores columna de V son ortogonales y de magnitud unitaria. Estas direcciones de entrada y salida están relacionadas por medio de los

valores singulares. Podemos ver que el i -ésimo valor singular σ_i , da directamente la ganancia de la matriz G en esta dirección, es decir

$$\sigma_i(G) = \|Gv_i\|_2 = \frac{\|Gv_i\|_2}{\|v_i\|_2}$$

1.5.2 MÁXIMO Y MÍNIMO VALOR SINGULAR

La ganancia más grande para cualquier dirección de entrada es igual al máximo valor singular

$$\max_{d \neq 0} \frac{\|Gd\|_2}{\|d\|_2} = \frac{\|Gv_1\|_2}{\|v_1\|_2} = \sigma_1(G) \stackrel{\Delta}{=} \bar{\sigma}(G)$$

Y la ganancia más pequeña para cualquier dirección de entrada es igual al mínimo valor singular

$$\min_{d \neq 0} \frac{\|Gd\|_2}{\|d\|_2} = \frac{\|Gv_k\|_2}{\|v_k\|_2} = \sigma_k(G) \stackrel{\Delta}{=} \underline{\sigma}(G)$$

Donde $k = \min\{l, m\}$. Entonces, para cualquier vector d tenemos que

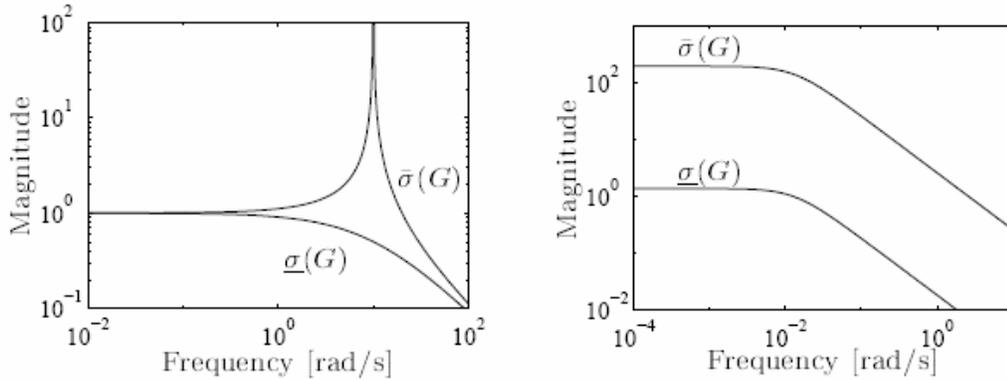
$$\underline{\sigma}(G) \leq \frac{\|Gd\|_2}{\|d\|_2} \leq \bar{\sigma}(G)$$

Si se define $v_1 = \bar{v}$ como la dirección de entrada con mayor amplificación, y $u_1 = \bar{u}$ como la dirección de salida en la cual las entradas son más efectivas, es decir, que a un mismo valor de entrada generan el mayor valor en la salida.

La dirección que involucra a \bar{v} y \bar{u} es designada como la dirección más “fuerte”, de “Alta ganancia” o “más importante”. La siguiente dirección más importante es asociada con v_2 y u_2 , y así sucesivamente hasta la dirección “menos importante”, “débil” o de “Baja ganancia” la cual está asociada con \underline{v} y \underline{u} .

Definiendo $u_1 = \bar{u}, v_1 = \bar{v}, u_k = \underline{u}, v_k = \underline{v}$ luego entonces tenemos

$$G\bar{v} = \bar{\sigma}\bar{u}, \quad G\underline{v} = \underline{\sigma}\underline{u}$$



(a) Spinning satellite

(b) Distillation process

Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 4: Gráficas típicas de valores singulares

Para sistemas dinámicos, los valores singulares y sus direcciones asociadas, varían con la frecuencia, y para propósitos de control, usualmente el rango de frecuencia de mayor interés corresponde al ancho de banda en lazo cerrado.

Gráfica típica de la descomposición de valores singulares (SVD) para un satélite que gira es mostrada en la figura 4(a), siendo la siguiente la planta:

$$G(s) = \frac{1}{s^2 + a^2} \begin{bmatrix} s - a^2 & a(s+1) \\ -a(s+1) & s - a^2 \end{bmatrix}; \quad a = 10$$

Otra gráfica típica de descomposición de valores singulares (SVD) para una columna de destilación con el modelo siguiente se muestran en la Figura 4(b):

$$G(s) = \frac{1}{75s + 1} \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix}$$

Y su descomposición de valores singulares (SVD) en estado estable es:

$$G = \begin{bmatrix} 0.625 & -0.781 \\ 0.781 & 0.625 \end{bmatrix} \begin{bmatrix} 197.2 & 0 \\ 0 & 1.39 \end{bmatrix} \begin{bmatrix} 0.707 & -0.708 \\ -0.708 & -0.707 \end{bmatrix}^H$$

La descomposición de valores singulares es útil cuando se trata de plantas cuyas dimensiones no son cuadradas. Por ejemplo, considerando una planta de dos entradas y tres salidas, en este caso, el tercer vector singular de salida dice en cual dirección de salida la planta no puede ser controlada. Así mismo, para una planta con una cantidad mayor de entradas que salidas, el vector singular de entrada nos dice en cuales direcciones la entrada no tiene efecto en la salida de la planta.

1.5.3 USO DEL MÍNIMO VALOR SINGULAR

El mínimo valor singular de una planta, $\underline{\sigma}(G(j\omega))$, evaluado como una función de la frecuencia es una medición útil para evaluar fehacientemente la capacidad de alcanzar control aceptable. Luego entonces, con una entrada manipulada de magnitud unitaria (medida con norma a dos), podemos obtener una salida de magnitud de por lo menos $\underline{\sigma}(G)$, en cualquier dirección de salida. Generalmente se desea que $\underline{\sigma}(G)$ tan grande como sea posible, al menos para evitar saturaciones de entrada $\underline{\sigma}(G)$ deber ser mayor a uno en todas las frecuencias donde se realiza control.

1.5.4 VALORES SINGULARES PARA EL DESEMPEÑO

El valor singular máximo es útil en términos del desempeño en el dominio de la frecuencia. Para un sistema de control realimentado, la respuesta de lazo cerrado desde la referencia, r , hacia el error de control, $e = y - r$, $e = -Sr$. Tenemos que para sistemas MIMO se considera la relación $\|e(\omega)\|_2 / \|r(\omega)\|_2$ donde $\|\bullet\|_2$ es la norma a dos del vector. Como vemos, la relación depende de la dirección de $r(\omega)$, luego entonces

$$\underline{\sigma}(S(j\omega)) \leq \frac{\|e(\omega)\|_2}{\|r(\omega)\|_2} \leq \bar{\sigma}(S(j\omega))$$

En términos de desempeño, se requiere que la relación $\|e(\omega)\|_2 / \|r(\omega)\|_2$ permanezca pequeño para cualquier dirección de $r(\omega)$, incluyendo el “peor caso” de dirección la cual da $\underline{\sigma}(S(j\omega))$. Dejando que a cada frecuencia $1/|W_p(j\omega)|$ represente la magnitud máxima permitida de $\|e\|_2 / \|r\|_2$. Lo cual resulta en un requerimiento de desempeño

$$\begin{aligned}\overline{\sigma}(S(j\omega)) \leq 1/|W_p(j\omega)|, \forall \omega &\Leftrightarrow \overline{\sigma}(W_p S) \leq 1, \forall \omega \\ &\Leftrightarrow \|W_p S\|_{\infty} \leq 1\end{aligned}$$

Donde la norma H_{∞} está definida como el pico del valor singular máximo de la respuesta de frecuencia, es decir, la norma a infinito es el máximo valor que toma el vector para toda frecuencia

$$\|M(s)\|_{\infty} \triangleq \max_{\omega} \overline{\sigma}(M(j\omega))$$

Los valores singulares de $S(j\omega)$ pueden ser graficados como una función de la frecuencia. Típicamente, son pequeños a bajas frecuencias donde la realimentación es efectiva, y se acercan a uno a altas frecuencias porque cualquier sistema real es estrictamente propio. El máximo valor singular, $\overline{\sigma}(G(j\omega))$, tiene un pico mayor que uno a la frecuencia de cruce. Este pico resulta indeseable, pero es inevitable para sistemas Reales.

El ancho de banda dependerá en las direcciones, y se tiene una región de ancho de banda entre una frecuencia menor donde el máximo valor singular, $\overline{\sigma}(S)$, alcanza 0.7 (Dirección del “peor caso”), y una frecuencia más alta, donde el valor singular mínimo, $\underline{\sigma}(S)$, alcanza 0.7 (Dirección del “mejor caso”). El ancho de banda puede ser definido como

$$W_B : \text{Frecuencia donde } \overline{\sigma}(S) \text{ cruza } 1/\sqrt{2} = 0.7.$$

Se entiende que el ancho de banda es por lo menos W_B para cualquier dirección de la señal de entrada, para lo cual

$$\underline{\sigma}(L) - 1 \leq \frac{1}{\overline{\sigma}(S)} \leq \underline{\sigma}(L) + 1$$

Esto es, a frecuencias donde la realimentación es efectiva (donde $\underline{\sigma}(L) \gg 1$) tenemos que $\overline{\sigma}(S) \approx 1/\underline{\sigma}(L)$, y a la frecuencia de ancho de banda (donde $1/\overline{\sigma}(S(jW_B)) = \sqrt{2} = 1.41$) tenemos que $\underline{\sigma}(L(jW_B))$ esta entre 0.41 y 2.41. Esto es que el ancho de banda es aproximadamente donde $\underline{\sigma}(L(j\omega))$ cruza 1.

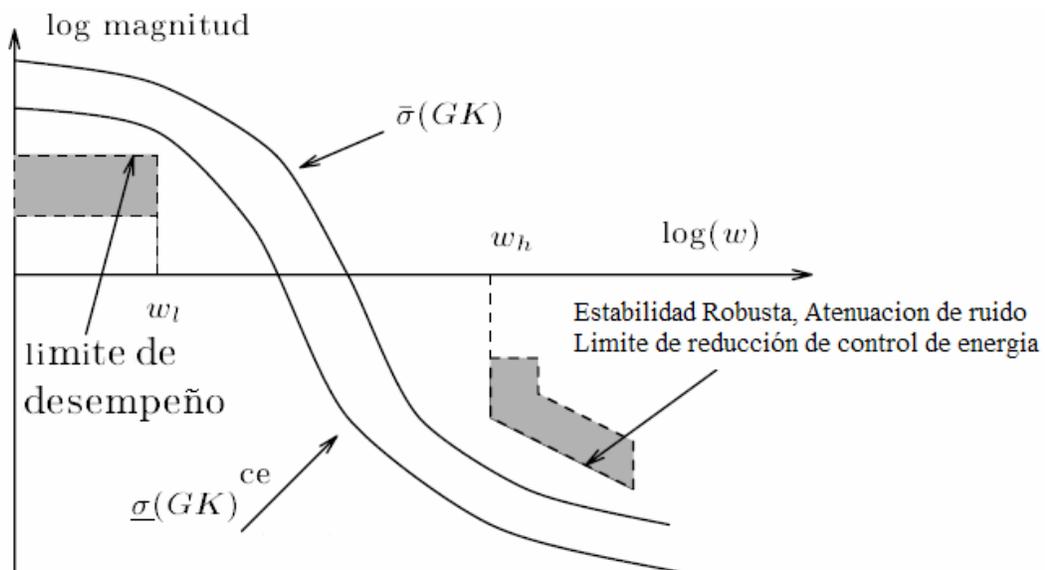
Finalmente, a mayor frecuencia donde para cualquier sistema real $\underline{\sigma}(L)$ (y $\overline{\sigma}(L)$) tenemos que $\overline{\sigma}(S) \approx 1$.

El hecho de “darle” forma o diseñar el lazo de una función de transferencia está basado en la idea de que una definición satisfactoria de ganancia para una matriz de función de transferencia está dada por los valores singulares de la función de transferencia. Para lo cual se plantean algunos objetivos de control

1. Para el rechazo de perturbaciones se hace C pequeño.
2. Para la atenuación del ruido se hace $\overline{\sigma}(T)$ pequeño.
3. Para realizar Tracking se hace $\overline{\sigma}(T) \cong \underline{\sigma}(T) \cong 1$.
4. Para reducción de energía de control se hace $\overline{\sigma}(KS)$ pequeño.
5. Para estabilidad robusta en presencia de perturbaciones aditivas se hace $\overline{\sigma}(KS)$.
6. Para estabilidad robusta en presencia de perturbaciones multiplicativas en la salida se hace $\overline{\sigma}(T)$ pequeño.

Estos requerimientos de lazo cerrado no pueden ser satisfechos simultáneamente. Por lo tanto en el momento de diseño realimentado hay una relación entre las frecuencias y los requerimientos porque los rangos entre los cuales los objetivos son importantes pueden ser bastante diferentes.

Por ello, para frecuencias específicas, se pueden aproximar estos requerimientos de lazo cerrado por los siguientes requerimientos de lazo abierto:



: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 5: Relaciones de diseño en lazo cerrado

1. Para rechazo de perturbación se hace $\underline{\sigma}(GK)$ grande; válido para frecuencias a las cuales $\underline{\sigma}(GK) \gg 1$.
2. Para atenuación de ruido se hace $\bar{\sigma}(GK)$ pequeño; válido para frecuencias a las cuales $\bar{\sigma}(GK) \gg 1$.
3. Para realizar Tracking se hace $\underline{\sigma}(GK)$ grande; válido para frecuencias a las cuales $\underline{\sigma}(GK) \gg 1$.
4. Para reducción de la energía de control se hace $\bar{\sigma}(K)$ pequeño; valido para frecuencias a las cuales $\bar{\sigma}(GK) \ll 1$.
5. Para estabilidad robusta ante una perturbación aditiva se hace $\bar{\sigma}(K)$ pequeño; valido para frecuencias en las cuales $\bar{\sigma}(GK) \ll 1$.
6. Para estabilidad robusta ante una perturbación multiplicativa en la salida se hace $\bar{\sigma}(GK) \ll 1$.

Típicamente, los requerimientos de lazo abierto 1 y 3 son validos e importantes a bajas frecuencias, mientras que 2, 4, 5 y 6 son condiciones las cuales son importantes a altas frecuencias. En la Figura 5 se observa que $\bar{\sigma}(GK)$ y $\underline{\sigma}(GK)$ deben evitar las regiones sombreadas

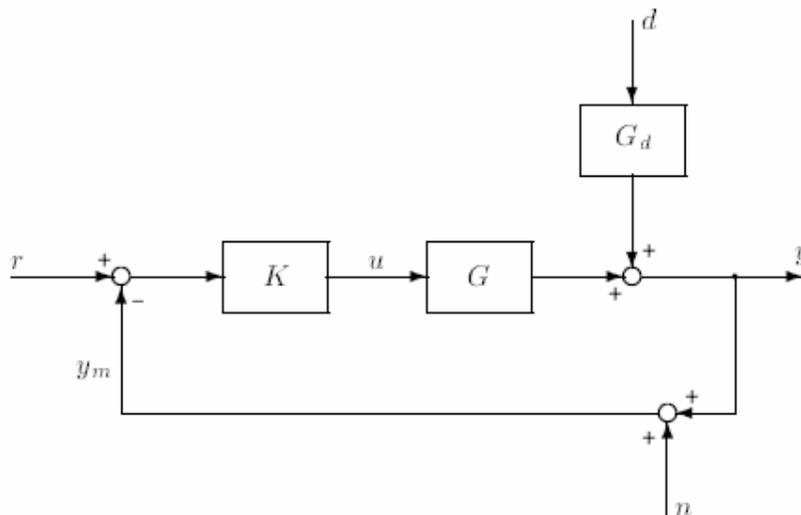
2. CONTROL DE PLANTAS MULTIVARIABLE

Un concepto simple de control multivariable para un sistema como el de la Figura 6 consta de un procedimiento de dos pasos generales en el cual primero se diseña un “compensador” para manejar las interacciones con G , y luego diseñar un controlador diagonal.

El procedimiento más común es diseñar un pre-compensador, $W_1(s)$, el cual contrarresta las interacciones en la planta y resulta en una nueva planta formada:

$$G_s(s) = G(s)W_1(s)$$

La cual es más diagonal y fácil de controlar que la planta original $G(s)$



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 6: Configuración de control por realimentación

2.1 CONTROLADOR DIAGONAL

Otro acercamiento utilizado es el uso de un controlador diagonal $K(s)$. Esto suele llamarse como control descentralizado. Esto funciona bien si $G(s)$ se acerca a la diagonal, porque así básicamente la planta es esencialmente un conjunto de subplantas y cada elemento en $K(s)$ puede ser diseñado independientemente. Sin embargo, si los elementos fuera de la diagonal son grandes, el desempeño de un controlador diagonal será pobre porque no se intenta eliminar las interacciones dentro de la planta.

2.2 LAZO DEL “MEJOR” CONTROLADOR

Teniendo en cuenta el rechazo de las perturbaciones. La respuesta de las perturbaciones en lazo cerrado es $y = SG_d d$. Suponiendo que las perturbaciones a cada frecuencia son de magnitud uno, $\|d\|_2 \leq 1$, y nuestro criterio de diseño es que $\|y\|_2 \leq 1$. Lo cual es equivalente a requerir que $\bar{\sigma}(SG_d) \leq 1$. En ciertas ocasiones hay intercambios entre el uso de la entrada y el desempeño, tal que el controlador que minimiza la magnitud de la señal de entrada es uno que produce todos los valores singulares de SG_d igual a uno, es decir, $\sigma_i(SG_d) = 1, \forall \omega$.

Esto corresponde a

$$S_{\min} G_d = U_1$$

Donde $U_1(s)$ es una función de transferencia pasa todo la cual a cada frecuencia tiene todos sus valores singulares iguales a uno. El subíndice *min* se refiere al uso de la ganancia de lazo más pequeña que satisface el objetivo de desempeño. En frecuencias donde la realimentación es efectiva se tiene que $S = (I + L)^{-1} \approx L^{-1}$, luego entonces, $L_{\min} = GK_{\min} \approx G_d U_1^{-1}$. Por lo tanto, el controlador y la forma del lazo con mínima ganancia serán

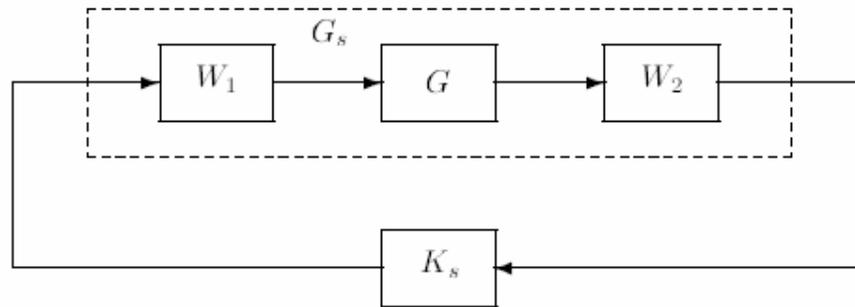
$$K_{\min} \approx G^{-1} G_d U, L_{\min} \approx G_d U$$

Donde $U = U_1^{-1}$ es una función de transferencia pasa todo.

3. SÍNTESIS DE CONTROL MULTIVARIABLE

3.1 DISEÑO DEL LAZO (DISEÑO H_∞)

Para realizar el controlador se hace necesario tener un pre-compensador y un post-compensador para darle forma a la planta y a los valores singulares de lazo abierto, tal cual se ve en la Figura 7.



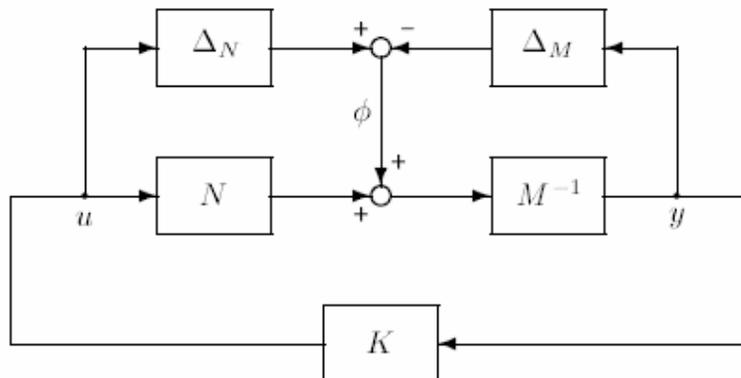
Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 7: Lazo diseñado y controlador

Si W_1 y W_2 son el pre-compensador y el post-compensador respectivamente, entonces la planta formada es

$$G_s = W_2 G W_1$$

Primero se tiene en cuenta el modelo de la planta con incertidumbre tal como se ve en la Figura 8



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 8: Modelo con Incertidumbre

Para obtener el controlador hacemos que la planta formada sea

$$G_s = M^{-1}N$$

El modelo de la planta perturbada puede ser escrito como

$$G_p = (M + \Delta_M)(N + \Delta_N)$$

Donde Δ_M y Δ_N son funciones de transferencia estables desconocidas que representan la incertidumbre en el modelo nominal de la planta G . El objetivo de la estabilización robusta es no solamente estabilizar la planta G , sino también a la familia de plantas perturbadas definidas por

$$G_p = \left\{ (M + \Delta_M)^{-1}(N + \Delta_N) : \|\begin{bmatrix} \Delta_N & \Delta_M \end{bmatrix}\|_\infty \leq \epsilon \right\}$$

Donde $\epsilon > 0$ es el margen de estabilidad.

La propiedad de estabilidad es robusta si y solo si la realimentación nominal del sistema es estable y

$$\gamma = \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_\infty \leq \frac{1}{\epsilon}$$

Donde γ es la norma H_∞ desde ϕ a $\begin{pmatrix} u \\ y \end{pmatrix}$ y $(I - GK)^{-1}$ es la función de sensibilidad para este arreglo de realimentación positiva.

El valor más pequeño de γ alcanzable y el máximo margen de estabilidad ϵ correspondiente está dado como

$$\gamma_{\min} = \epsilon_{\max}^{-1} = \left\{ 1 - \|[N - M]\|_H^2 \right\}^{-1/2} = (1 + \rho(XZ))^{1/2}$$

Donde $\|\bullet\|_H$ es la norma de Hankel, ρ denota el radio espectral (máximo valor propio), y una realización de ecuaciones de estado (A, B, C, D) de G , Z es la única solución positiva definitiva de la solución de la ecuación algebraica de Riccati

$$(A - BS^{-1}D^T C)Z + Z(A - BS^{-1}D^T C)^T - ZC^T R^{-1} CZ + BS^{-1}B^T = 0$$

Donde

$$R = I + DD^T, S = I + D^T D$$

Y X es la única solución positiva definitiva de la siguiente ecuación de Riccati

$$(A - BS^{-1}D^T C)^T X + X(A - BS^{-1}D^T C) - XBS^{-1}B^T X + C^T R^{-1}C = 0$$

Un controlador el cual garantice que

$$\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_{\infty} \leq \gamma$$

Para $\gamma > \gamma_{\min}$ especificado, está dado por

$$K = \begin{bmatrix} A + BF + \gamma^2 (L^T)^{-1} ZC^T (C + DF) & \gamma^2 (L^T)^{-1} ZC^T \\ B^T X & -D^T \end{bmatrix}$$

$$F = -S^{-1}(D^T C + B^T X)$$

$$L = (1 - \gamma^2)I + XZ$$

De esta manera se obtiene el controlador K una vez ya se han obtenido el pre-compensador y el post-compensador.

El código que se implementa en MATLAB para desarrollar este tipo de control es el siguiente

```
G=ss(a,b,c,d);           % Funcion de transferencia en ss
s=zpk('s');             % Variable Laplace (s)
Gd=20/s;                 % Curva del lazo deseado
handles.mydata.Gd=Gd;
% Se computa el controlador optimal de lazo (K)
[K CL GAM]=loopsyn(G,Gd);
handles.mydata.GAM=GAM;
% LIMITACION: G debe ser estabilizable, detectable, y de rango
completo
L=G*K;                   % Se halla el lazo abierto L
S=feedback(I,L);         % Se halla la sensibilidad (S)
T=I-S;                   % Se halla la sensibilidad complementaria (T)
```

Para este tipo de controlador se ha omitido el pre-compensador y el post-compensador ya que la función *loopsyn* funciona con la planta a controlar y la curva del lazo deseado, para este caso se utiliza la función del lazo como

$$G_d = \frac{\omega_b}{s}$$

Donde ω_b corresponde al ancho de banda de las frecuencias a controlar.

3.2 CONTROL LQG

Para Control LQG se deben conocer muy bien las dinámicas de la planta la cual debe ser lineal y adema conocer también los ruidos de medición y las perturbaciones (ruido del proceso), los cuales deben ser estocásticos con sus propiedades estadísticas conocidas. Es decir, teniendo el modelo de la planta

$$\begin{aligned} \dot{x} &= Ax + Bu + \omega_d \\ y &= Cx + \omega_n \end{aligned}$$

Donde ω_d y ω_n son las perturbaciones (ruido del proceso) y el ruido de entrada de las mediciones respectivamente, los cuales se asumen como proceso estocásticos gaussianos no correlacionado con media cero con matrices de densidad espectral con potencia constante W y V respectivamente. Esto es, ω_d y ω_n son procesos de ruido blanco con covarianzas

$$\begin{aligned} E\{\omega_d(t)\omega_d(t)^T\} &= W\delta(t-\tau) \\ E\{\omega_n(t)\omega_n(t)^T\} &= V\delta(t-\tau) \end{aligned}$$

Y

$$E\{\omega_d(t)\omega_n(t)^T\} = 0, E\{\omega_n(t)\omega_d(t)^T\} = 0$$

Donde E es la esperanza y $\delta(t-\tau)$ es la función delta.

El problema de control LQG es encontrar el control optimal $u(t)$ el cual minimiza

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

Donde Q y R son las matrices constantes de pesos (parámetros de diseño) tales que $Q = Q^T \geq 0$ y $R = R^T > 0$. El nombre LQG viene del uso del modelo Lineal, una función de costo integral Cuadrática y ruidos blancos Gaussianos de proceso para modelar las señales y el ruido de perturbación. Primero se calcula el control optimal para un regulador LQR, es decir, sin ω_d y ω_n . Lo cual se escribe en términos de la ley de realimentación de estados

$$u(t) = -K_r x(t)$$

Donde K_r es una matriz constante la cual es independiente de W y V , las propiedades estadísticas del ruido de la planta. Se conocen todos los estados, y dada una condición inicial diferente de cero, encontrar la señal $u(t)$ que lleve el sistema al estado cero de forma óptima, es decir, minimizando el costo determinístico

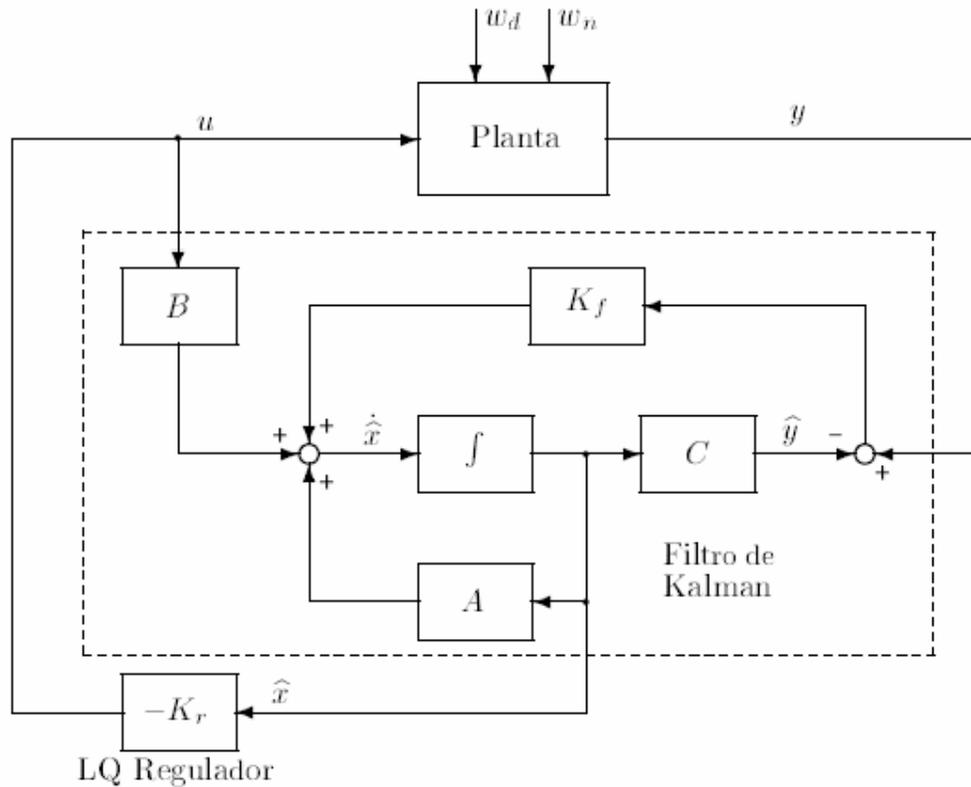
$$J_r = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

La solución optimal está dada para un estado inicial $u(t) = -K_r x(t)$ donde

$$K_r = R^{-1} B^T X$$

Y $X = X^T \geq 0$ es la única solución positiva de la ecuación algebraica de Riccati

$$A^T X + X A - X B R^{-1} B^T X + Q = 0$$



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 9: Controlador LQG y planta con ruido

El filtro de Kalman tiene la estructura de un estimador de estados ordinario, como se ve en la Figura 9, con

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

La elección de K_f , la cual minimiza $E\left\{\begin{bmatrix} x - \hat{x} \\ x - \hat{x} \end{bmatrix}^T \begin{bmatrix} x - \hat{x} \\ x - \hat{x} \end{bmatrix}\right\}$ está dada por

$$K_f = YC^T V^{-1}$$

Donde $Y = Y^T \geq 0$ es la única solución positiva semidefinitiva de la ecuación algebraica de Riccati

$$YA^T + AY - YC^T V^{-1} CY + W = 0$$

La función de transferencia de la combinación del estimador de estados optimal y la realimentación de estados optimal es desde y hasta u es

$$K_{LQG} = \begin{bmatrix} A - BK_r - K_f C & K_f \\ -K_r & 0 \end{bmatrix}$$

$$= \begin{bmatrix} A - BR^{-1}B^T X - YC^T V^{-1}C & YC^T V^{-1} \\ -R^{-1}B^T X & 0 \end{bmatrix}$$

Para desarrollar este control en MATLAB se utilizó el siguiente código

```
G = ss(a,b,c,d); % Funcion de transferencia en ss
[n,m] = size(b); % Se halla el vector de estado y los tamaños de
%control
P = size(c,1); % Se halla el tamaño del vector de salida
% Primero se diseña el regulador de Realimentacion de estados
A = [a zeros(n,m) ; -c zeros(m,m)]; B = [b ; -d]; % Planta aumentada
%con integradores
Q = [zeros(n,n) zeros(n,m); zeros(m,n) eye(m,m)]; % Peso en el
%error integrado
R = 1*eye(m); % Peso de entrada - R pequeña de respuesta mas rapida
Kr = lqr(A,B,Q,R); % Se diseña el regulador optimo de realimentacion
% de estados
Krp = Kr(1:m,1:n); Kri = Kr(1:m,n+1:n+m); % Se extrae la
realimentacion de los estados y el integrador
% Ahora se diseña el filtro de Kalman para la planta
Bnoise = eye(n); % Ruido del proceso (disturbance) q entra
directamente a la planta
W = eye(n); % Peso del ruido del proceso
V = 1*eye(m);
% Peso de la medicion del ruido - disminuirlo genera una respuesta mas
%rapida
Ke = lqe(a,Bnoise,c,W,V);
% No hay necesidad de estimar los estados del integrador
% Se genera el controlador de 2DOF desde [r y]' a u con integradores
%incluidos
Ac = [ zeros(m,m) zeros(m,n); -b*Kri a-b*Krp-Ke*c];
Bc = [eye(m) -eye(m); zeros(n,m) +Ke];
Cc = [-Kri -Krp]; Dc = [zeros(m,m) zeros(m,m)];
K = ss(Ac,Bc,Cc,Dc);
Gsim = ss(a,b,[c;c],[d;d]);
disp(' ');
disp('Controlador LQG: ');
Ksim = ss(Ac,Bc,[Cc;Cc],[Dc;Dc]);
sys = lft(Gsim,-Ksim,1,1);
Bc1 = Bc(:,1:m); Bc2=Bc(:,m+1:m+m);

Acl = [a b*Cc ; Bc2*c Ac+Bc2*d*Cc ];
Bc1 = [zeros(n,m) ; Bc1 ];
Cc1 = [c d*Cc ; zeros(m,n) Cc];
% Aumentado para incluir la señal de control u
Dc1 = [zeros(m,m); zeros(m,m)];
T=ss(Acl,Bc1,Cc1,Dc1);
```

Para el diseño del controlador LQG los pesos del ruido y del filtro se dejaron de valor unitario para ser utilizados de forma general, por medio del conocimiento de la planta y de la experiencia de la interacción de la planta con el controlador se pueden hacer ajustes que lleven a una respuesta de desempeño óptimo.

3.3 ROBUSTEZ

La robustez se hace necesaria cuando los procesos no son definidos exactamente (planta), debido a esto siempre hay una cantidad de incertidumbre en el modelo del sistema.

La incertidumbre en una planta tiene varios orígenes:

1. Siempre hay parámetros en el modelo lineal en el cual son conocidos aproximadamente o son de error simple.
2. Los parámetros del modelo lineal pueden variar debido a las no linealidades o cambios en las condiciones de operación.
3. Los dispositivos de medición tienen imperfectos. Esto puede aumentar la incertidumbre en las entradas manipuladas, cuando la entrada actual a menudo es medida y ajustada en serie.
4. A altas frecuencias, la estructura y el orden del modelo son desconocidos, y la incertidumbre excede el 100% en alguna frecuencia.
5. Inclusive cuando un modelo muy detallado se tiene, se escoge trabajar con un modelo nominal más simple (de orden menor).
6. Finalmente, el controlador implementado puede ser diferente al obtenido en el diseño. En cuyo caso se pueden incluir incertidumbres para permitir reducción en el orden del controlador e inexactitudes en la implementación.

Las fuentes de incertidumbre pueden ser definidas en dos grupos:

1. Incertidumbre paramétrica. En este grupo, el modelo de la estructura es conocido, pero algunos parámetros no lo son.
2. Incertidumbre por dinámicas desconocidas o sin modelar. En este grupo, el modelo tiene error debido a las dinámicas desconocidas, usualmente a alta frecuencia, que se deben ya sea a desconocimiento deliberado o a la carencia del entendimiento del proceso físico. Cualquier modelo de un proceso real tendrá esta fuente de incertidumbre.

Se considerará la incertidumbre de la ganancia de entrada, ϵ , entonces se tiene

$$u' = \epsilon u$$

Se nota que la incertidumbre esta en el cambio de las entradas, y no en sus valores absolutos. Un 20% de error es típico para aplicaciones de procesos de control. La incertidumbre definida previamente no causa por si misma inestabilidad.

Para un proceso con incertidumbre en la ganancia de sus entradas

$$L'(s) = K_{inv} G' = K_{inv} G \begin{bmatrix} 1+\epsilon_1 & 0 & 0 & 0 \\ 0 & 1+\epsilon_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1+\epsilon_n \end{bmatrix}$$

Donde K_{inv} controlador inverso, el cual puede ser visto como un controlador PI

$$K_{inv} = \frac{k}{s} G^{-1}(s)$$

Para el desarrollo de un controlador robusto se utilizo el siguiente código de MATLAB

```
G=ss(a,b,c,d);
G0=dcgain(G);

dyn = tf(1,[1 1]); % Planta con integrador
G=dyn*G0;

% Controlador Inverso
dynk = tf([1 1],[1 1.e-6]);
disp(' ');
disp('Controlador Robusto: ');
Kinv = (dynk*inv(G0));
ss(Kinv)
I2=eye(length(G*Kinv));

% Planta Nominal
S=minreal(inv(I2+G*Kinv));
T=minreal(I2-S);

% Con 20% de Incertidumbre a la entrada
Unc = [1.2 0; 0 0.8];
Lu=G*Unc*Kinv;
Su=minreal(inv(I2+Lu));
Tu=minreal(I2-Su);

% Simulacion en el tiempo
Kr=tf(1,[5 1]);
Tr=T*Kr; % Sensibilidad complementaria sin incertidumbre
Tru=Tu*Kr; % Sensibilidad complementaria con incertidumbre
```

3.4 DISEÑO DE SENSIBILIDAD MIXTA (H_∞)

Sensibilidad mixta se utiliza en problemas donde se modela el lazo de la función de transferencia, en la cual la función de sensibilidad $S = (I + GK)^{-1}$ es modelada por una o más funciones de transferencia de lazo cerrado tales como KS o la sensibilidad complementaria $T = I - S$.

Suponiendo, que se tiene un problema de regulación en el cual se desea rechazar una perturbación d que entra en la salida de la planta y se asume que el ruido de medición es relativamente insignificante. Realizar seguimiento (Tracking) no es un tema y por lo tanto para este problema, tiene sentido formar el lazo de la función de transferencia del lazo cerrado S y KS . Es importante notar que incluir KS como un mecanismo para limitar el tamaño del ancho de banda del controlador, y por lo tanto el control a la energía utilizada. El tamaño de KS también es importante para la estabilidad robusta con respecto a la incertidumbre modelada como perturbaciones aditivas de la planta.

La señal de perturbación d es típicamente una señal de baja frecuencia, y por lo tanto será rechazada si el máximo valor singular de S es pequeño para las mismas bajas frecuencias. Para esto se implementa un filtro escalar $w_1(s)$ con un ancho de banda igual al de la perturbación, y luego encontrar un controlador que minimice $\|w_1 S\|_\infty$. Esta función de costo por sí misma no es muy práctica. Solo se enfoca en una función de transferencia de lazo cerrado y para polos en el semiplano derecho el controlador optimal tiene ganancias infinitas. Por lo tanto la función a minimizar es

$$\left\| \begin{array}{l} w_1 S \\ w_2 KS \end{array} \right\|_\infty$$

Donde $w_2(s)$ es un filtro pasa alto escalar con una frecuencia de cruce por cero aproximadamente igual al ancho de banda deseado de la función de lazo cerrado.

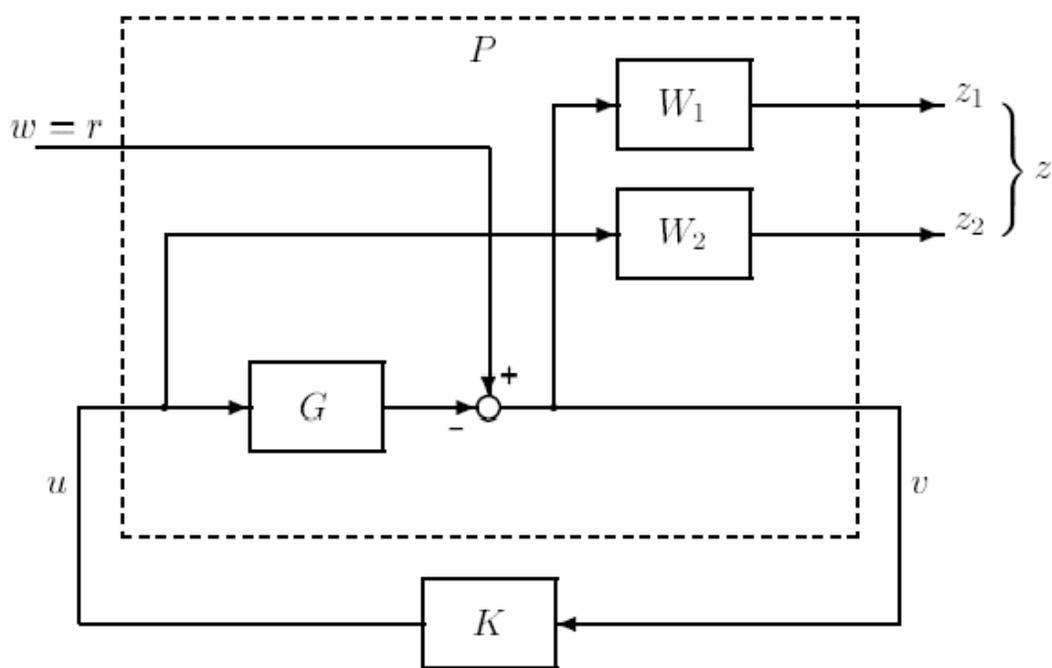
Las perturbaciones d se consideran como una sola entrada exógena y se define una señal de error igual a $z = [z_1^T \quad z_2^T]^T$, donde $z_1 = W_1 e$ y $z_2 = W_2 u$.

De la Figura 10 se puede ver que $z_1 = W_1 S \omega$ y $z_2 = W_2 K S \omega$ se requieren para determinar los elementos de la planta generalizada

Donde la planta generalizada es

$$P_{11} = \begin{bmatrix} W_1 \\ 0 \end{bmatrix} \quad P_{12} = \begin{bmatrix} W_1 G \\ -W_2 \end{bmatrix}$$

$$P_{21} = -I \quad P_{22} = -G$$



Fuente: SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

Figura 10: Forma estándar de la optimización de sensibilidad mixta

Donde

$$\begin{bmatrix} z_1 \\ z_2 \\ \dots \\ v \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}$$

La habilidad para darle forma a T es deseable para problema de perturbaciones y para lograr Tracking. También es importante para la estabilidad robusta con respecto a las perturbaciones multiplicativas a la salida de la planta.

Para la implementación del regulador en MATLAB se definen los pesos de desempeño como $W_p = \text{diag}\{w_{pi}\}$ con

$$w_{pi} = \frac{s/M_i + \omega_{Bi}^*}{s + \omega_{Bi}^* A_i}$$

Donde $A_i \gg 1$ asegura acción integral para frecuencias cercanas a cero. Se selecciona M_i igual a dos para cada salida, donde ω_{Bi}^* puede ser diferente para cada salida. Un valor grande para ω_{Bi}^* produce una respuesta mas rápida para la salida i .

Donde el código de MATLAB es el siguiente

```
G=ss(a,b,c,d);
% Diseño S-KS H-INFINITO
% Se requiere wb, pendiente l y pico menor q M
Mi=1.5; wbi=25; Ai=1e-4;
wpi = tf([1/Mi wbi], [1 wbi*Ai]);
Wp=[wpi 0;0 wpi]; Wu=1.0*eye(2);
[khinf1 hinfl gopt1] = mixsyn(G,Wp,Wu,[]);
disp(' ');
disp('Controlador de sensibilidad Mixta Hinf: ');
K=khinf1
S=inv(eye(2)+G*K);
T=eye(2)-S;
```

4. APLICACIÓN MATLAB

El desarrollo de la profundización de la teoría de control multivariable (MIMO) es realizado con el software MATLAB, el cual posee una gran versatilidad en la implementación de teorías y matemática para la teoría de control, en nuestro caso en particular se hace uso de la herramienta Graphical User Interface (GUI), para así, desarrollar un entorno más amigable al usuario.

La aplicación de MATLAB llamada “profundiMIMO” consta de un software que se encarga de desarrollar varios tipos de controladores a procesos multivariables, MIMO. Los procesos deben ser introducidos al programa en su forma de matrices de ecuaciones de estados

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

Con un número de estados n , un número de entradas l , un número de salidas m donde los tamaños de las matrices correspondientes son

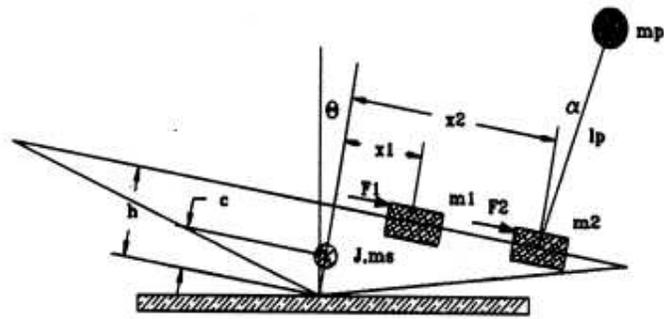
$$\begin{matrix} A_{n \times n} & B_{l \times n} \\ C_{m \times n} & D_{l \times m} \end{matrix}$$

Además se debe tener en cuenta que la planta debe tener estados observables y controlables.

Para la explicación de la aplicación se hace uso del modelo del péndulo invertido sobre un columpio, el cual posee ocho estados. Las variables de control son los voltajes de control de los motores DC. Las variables de salida son el ángulo de del péndulo invertido (α) y la posición del carro del péndulo

(x_1) y la entrada de control $u = [V_1 \quad V_2]^T$.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} x_1 \\ \theta \\ x_2 \\ \alpha \\ \dot{x}_1 \\ \dot{\theta} \\ x_2 \\ \alpha \end{bmatrix}$$



Fuente: Subbotin, Maxim. Balancing an Inverted pendulum on a seesaw
Figura 11: Configuración del péndulo invertido sobre un columpio

El modelo a utilizar está compuesto por las siguientes matrices

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0; \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0; \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0; \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000; \\ -1.3093 & 9.1850 & -1.0684 & 0 & -9.6021 & 0 & -0.1763 & 0; \\ 9.3525 & 4.3928 & -7.6311 & 0 & 1.2593 & 0 & 1.2593 & 0; \\ -1.3093 & 9.1850 & -1.0684 & -4.5231 & -0.1763 & 0 & -7.7622 & 0; \\ -9.3525 & -4.3928 & -7.6311 & 46.9609 & -1.2593 & 0 & 54.0963 & 0; \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0; \\ 0 & 0; \\ 0 & 0; \\ 0 & 0; \\ 1.2499 & 0.0230; \\ -0.1639 & -0.1639; \\ 0.0230 & 1.0104; \\ 0.1639 & -7.0420; \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0; \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0; \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0; \\ 0 & 0; \end{bmatrix}$$

Para ejecutar el aplicativo, se debe agregar al “path” de MATLAB la carpeta en donde se guardan los archivos `profundiMIMO.m` y `profundiMIMO.fig`.

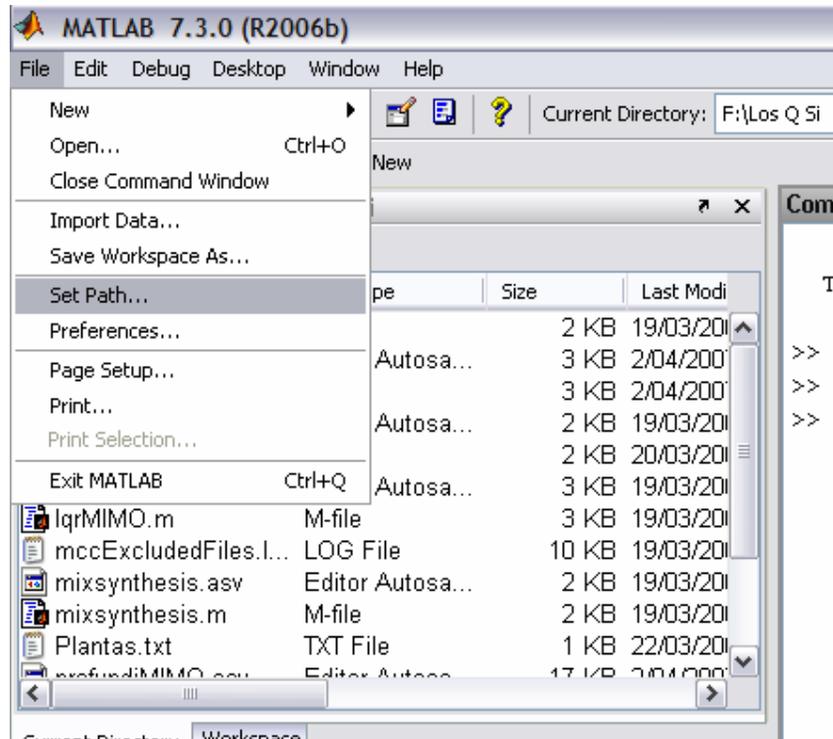


Figura 12: Forma para establecer la carpeta con los archivos en el “Path”

O también, se pueden copiar estos archivos en el directorio “WORK” que se encuentra dentro del directorio de archivos de MATLAB y luego se escribe en el command window de MATLAB “`profundiMIMO`”. Una ventana como la que se muestra en la figura siguiente debe aparecer



Figura 13: Vista del aplicativo de MATLAB

Como podemos observar, el aplicativo tiene cuatro casillas para introducir cada una de las matrices de estados (A, B, C, D), las cuales deben poseer las dimensiones apropiadas, las cuales fueron mencionadas previamente.

También posee un selector que permite escoger el tipo de controlador a diseñar, las posibles opciones son Diseño del lazo, Diseño LQG, Diseño de Control Robusto y Diseño de Sensibilidad Mixta con norma H_{∞} . Además se tiene el botón “CALCULAR CONTROLADOR”, el cual debe ser presionado una vez se introduce la planta y se determine el tipo de control a realizar.

El controlador es entregado en forma de matrices de estado, correspondientes a la documentación teórica previamente presentada. Las matrices de estado son entregadas en el “Command Window” de MATLAB. Para graficar la respuesta paso de la planta se procede a presionar el botón “Graficar” que se encuentra debajo del título “Respuesta Paso” y para graficar la descomposición de valores singulares se pulsa el botón “Graficar” que se encuentra debajo del título “SVD”.

Para finalizar, en el momento de salir solo es necesario presionar el botón “SALIR” que se encuentra en la esquina inferior derecha de la aplicación.

A continuación se mostrará el funcionamiento del aplicativo para la planta del péndulo invertido en un columpio con sus diferentes tipos de diseño.

4.1 DISEÑO DEL LAZO

Como Primer paso, se procede a introducir las matrices de estado de la planta en el aplicativo.



Figura 14: Pasos a seguir para manejo del aplicativo

En el segundo paso se selecciona la opción de “Diseño por Lazo” y como tercer paso se da clic en el botón “CALCULAR CONTROLADOR”.

Para lo cual se obtienen las siguientes matrices de estado que representan el controlador diseñado

Controlador de Diseño por lazo:

Controlador de Diseño por lazo:

a =

	x1	x2	x3	x4	x5	
x1	-1.302e-009	-1.804e-009	1.546e-011	1.164e-010		1
x2	4.547e-012	-1.601e-010	1.41e-010	2.183e-011	-3.524e-012	

x3	-2.183e-011	-1.455e-011	-3.638e-012	1.455e-011	-1.364e-012
x4	-1.237e-010	-1.164e-010	6.094e-011	-1.164e-010	-1.319e-011
x5	-2.466e+007	-5.033e+007	2.883e+007	5.284e+006	-2.176e+006
x6	3.27e+006	6.468e+006	-3.873e+006	-1.087e+006	2.885e+005
x7	-6.816e+005	-1.021e+005	1.117e+006	2.568e+006	-6.056e+004
x8	-1.645e+006	-1.235e+007	-3.09e+005	-1.654e+007	-1.421e+005
x9	0	0	0	0	0
x10	0	0	0	0	0
x11	0	0	0	0	0
x12	0	0	0	0	0
x13	0	0	0	0	0
x14	0	0	0	0	0
x15	0	0	0	0	0
x16	0	0	0	0	0
x17	0	0	0	0	0
x18	0	0	0	0	0
x19	0	0	0	0	0
x20	0	0	0	0	0

	x6	x7	x8	x9	x10
x1	-5.894e-010	-7.276e-012	1e-011	-0.4514	-2.024
x2	1	-3.638e-012	1.592e-012	-2.978	0.8928
x3	-7.276e-012	1	4.547e-013	1.708	-6.32
x4	-8.004e-011	-2.91e-011	1	7.756	51.82
x5	-1.628e+007	-4.386e+005	3.089e+005	8.384e+007	1.925e+006
x6	2.139e+006	-1.587e+004	-5.13e+004	-1.12e+007	-2.176e+006
x7	-3.237e+005	4.526e+005	7.345e+004	2.85e+006	1.211e+007
x8	-1.967e+006	-3.271e+006	-4.322e+005	1.879e+006	-8.397e+007
x9	0	0	0	0	0
x10	0	0	0	0	0
x11	0	0	0	0	0
x12	0	0	0	0	0
x13	0	0	0	0	0
x14	0	0	0	0	0
x15	0	0	0	0	0
x16	0	0	0	0	0
x17	0	0	0	0	0
x18	0	0	0	0	0
x19	0	0	0	0	0
x20	0	0	0	0	0

	x11	x12	x13	x14	x15
x1	-2.232e-010	3.297e-010	1.413e-010	-4.372e-012	-1.425e-011
x2	3.005e-011	-4.34e-011	-1.882e-011	6.051e-013	1.918e-012
x3	-8.693e-012	7.34e-012	4.62e-012	-2.798e-013	-5.72e-013
x4	3.268e-012	3.477e-011	4.738e-012	8.198e-013	4.583e-013
x5	1.761e-006	-3.371e-006	-1.476e-006	3.104e-008	-3.917e-008
x6	-2.34e-007	4.395e-007	1.956e-007	-4.329e-009	4.053e-009

x7	4.919e-008	-4.501e-008	-3.628e-008	2.292e-009	6.468e-009
x8	1.143e-007	-5.582e-007	-1.289e-007	-7.807e-009	-5.541e-008
x9	-1057	654.8	298.1	-12.4	-62.35
x10	-798.2	566.1	215.1	-8.288	-49.87
x11	-1372	944.9	381.5	-15.06	-84.96
x12	-1273	760.7	361.6	-15.54	-72.6
x13	-188.8	135.7	46.21	-1.362	-11.6
x14	4.464	-1.583	-1.86	-0.06489	-0.7265
x15	41.93	-29.8	-9.8	1.76	1.385
x16	-51.24	20.11	19.65	1.965	-6.338
x17	-1593	1022	424.5	-17.45	-64.05
x18	-45.73	-46.3	47.9	13.44	-34.01
x19	615.2	-392.7	-164.4	4.115	30.83
x20	-22.97	44.22	-8.2	-4.664	18.1

	x16	x17	x18	x19	x20
x1	-4.15e-011	1.294e-010	-5.689e-010	1.296e-011	-2.319e-010
x2	5.429e-012	-1.726e-011	7.606e-011	-1.695e-012	3.092e-011
x3	-4.05e-013	3.517e-012	-2.171e-011	1.537e-013	-6.253e-012
x4	-8.356e-012	8.981e-012	2.274e-012	2.141e-012	-1.592e-011
x5	-3.679e-008	-5.895e-007	-7.451e-008	-1.887e-007	-1.565e-006
x6	5.704e-009	7.835e-008	9.313e-010	2.519e-008	2.123e-007
x7	-5.239e-009	-1.508e-008	5.96e-008	-4.791e-009	-5.402e-008
x8	2.561e-008	-4.68e-008	-4.359e-007	-1.621e-008	-3.725e-008
x9	-250	450.1	-1625	129.7	-645.5
x10	-197.4	267.7	-1261	69.17	-498.8
x11	-337.6	509.6	-2170	137.8	-859.6
x12	-292.7	585	-1918	173.6	-763.3
x13	-44.68	34.67	-278.3	5.549	-108.9
x14	0.4025	11.7	2.158	4.625	3.005
x15	3.127	-36.78	26.16	-7.684	5.155
x16	-52.19	-53.06	-360.1	-45.51	-237.8
x17	-417.4	-1639	-2459	-1659	-849.8
x18	-298.9	99.88	-2272	-88.35	-1842
x19	207.4	1590	1070	-5999	377
x20	198.2	-145.4	1777	-8.044	-5949

b =

	u1	u2
x1	-1.628e-013	-1.312e-013
x2	-1.42e-013	-3.467e-013
x3	-2.115e-013	-4.987e-013
x4	-7.668e-014	-5.661e-013
x5	1.308e-007	6.277e-007
x6	-6.381e-009	-1.254e-007
x7	9.12e-009	-9.548e-009
x8	1.673e-009	7.669e-008
x9	-136.6	-328.4

x10	-119.5	-244.4
x11	-198.1	-422.8
x12	-158	-395.1
x13	-28.59	-56.55
x14	0.3684	1.468
x15	6.26	12.66
x16	-4.578	-16.74
x17	-217.3	-489.8
x18	8.078	-19.9
x19	83.5	189.3
x20	-8.734	-4.805

c =

	x1	x2	x3	x4	x5	
y1	1.973e+007	4.028e+007	-2.305e+007	-4.182e+006	1.74e+006	
y2	2.255e+005	-8.158e+005	-5.804e+005	-2.447e+006	2.032e+004	

	x6	x7	x8	x9	x10	
y1	1.303e+007	3.593e+005	-2.459e+005	-6.706e+007	-1.32e+006	
y2	2.386e+004	-4.561e+005	-6.709e+004	-1.294e+006	-1.196e+007	

	x11	x12	x13	x14	x15	
y1	-1.406e-006	2.695e-006	1.178e-006	-2.484e-008	3.225e-008	
y2	-1.625e-008	-1.671e-008	8.986e-009	-1.683e-009	-7.16e-009	

	x16	x17	x18	x19	x20	
y1	2.887e-008	4.745e-007	7.451e-008	1.499e-007	1.263e-006	
y2	4.191e-009	4.657e-009	-6.054e-008	1.08e-009	2.421e-008	

d =

	u1	u2
y1	-0.0004077	-0.0009567
y2	-4.121e-006	-9.089e-006

Continuous-time model.

Como cuarto y último paso, se obtienen la respuesta paso de la planta controlada y la gráfica de descomposiciones de valores singulares. En la grafica de la respuesta paso se observa la respuesta de las salidas ante una entrada paso en una sola entrada por vez.

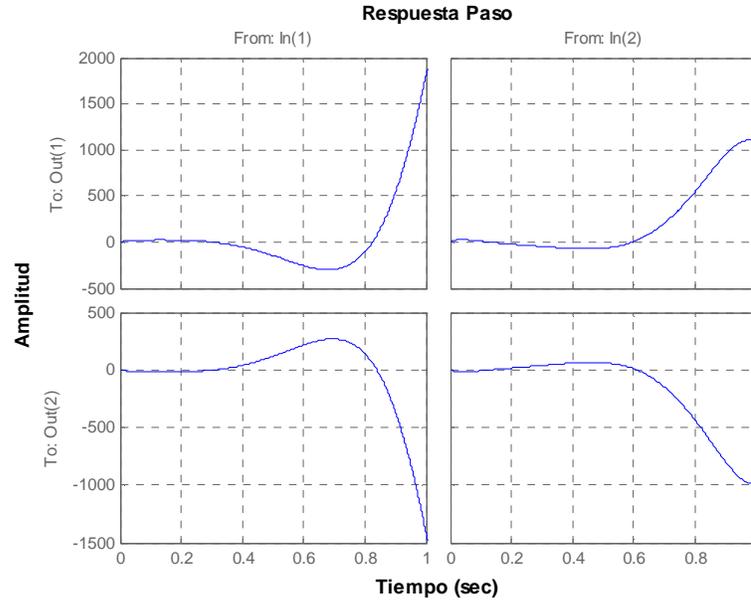


Figura 15: Respuesta Paso de la planta controlada (Diseño del Lazo)

Como se puede observar en la Figura 16, la descomposición de valores singulares (SVD) nos muestra las curvas para el desempeño, la robustez y las SVD para lazo abierto y cerrado.

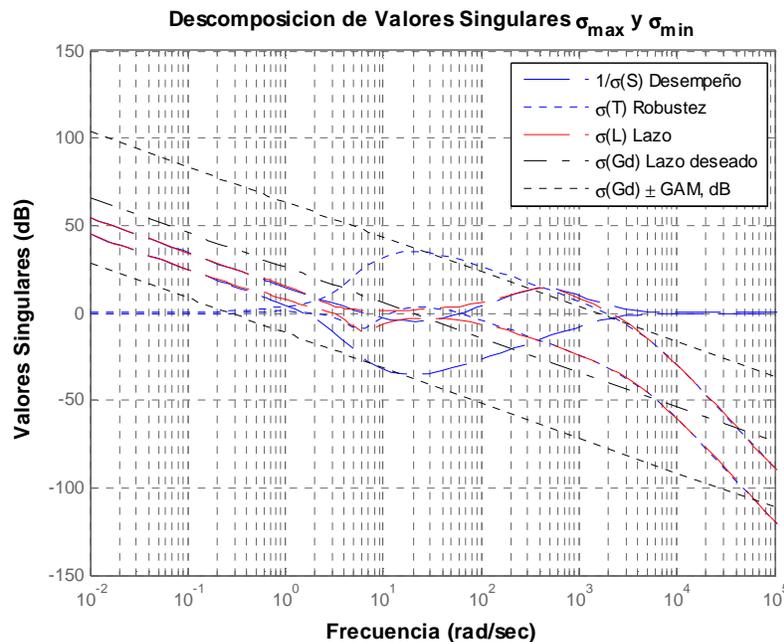


Figura 16: SVD de la planta controlada (Diseño del Lazo)

4.2 LQG

Para diseñar el controlador LQG, se introducen las variables de estado de la planta (igual para todos los tipos de diseño) y se selecciona la opción "LQG".



Figura 17: Pasos a seguir para diseño del controlador LQG

Después de presionar el botón "CALCULAR CONTROLADOR", el controlador diseñado es el siguiente

Controlador LQG:

a =

	x1	x2	x3	x4	x5	x6	x7
x1	0	0	0	0	0	0	0
x2	0	0	0	0	0	0	0
x3	0	0	-1.882	0	0	-1.844	1
x4	0	0	-4.588	0	0	-11.43	0
x5	0	0	-1.498	0	0	-0.5456	0
x6	0	0	-1.844	0	0	-17.52	0
x7	1.197	-0.3611	-100.8	-47.26	69.02	8.528	-20.53
x8	-0.1133	0.2022	10.94	8.273	-16.06	-38.58	2.918
x9	-0.2518	-0.9788	-19.08	30.25	-4.551	11.28	-1.798

x10 2.067 6.734 38.39 -166 34.85 -250 7.218

	x8	x9	x10
x1	0	0	0
x2	0	0	0
x3	0	0	0
x4	1	0	0
x5	0	1	0
x6	0	0	1
x7	-27.27	17.64	3.398
x8	3.166	-3.701	-1.071
x9	2.068	9.045	3.989
x10	-21.5	-58.52	-26.94

b =

	u1	u2	u3	u4
x1	1	0	-1	0
x2	0	1	0	-1
x3	0	0	1.882	1.844
x4	0	0	4.588	11.43
x5	0	0	1.498	0.5456
x6	0	0	1.844	17.52
x7	0	0	2.972	8.028
x8	0	0	13.08	33.11
x9	0	0	3.356	5.217
x10	0	0	27.74	154.6

c =

	x1	x2	x3	x4	x5	x6	x7
y1	0.9625	-0.2712	-76.96	-45.56	56.16	12.87	-8.716
y2	-0.2712	-0.9625	-12.51	21.89	-4.726	20.52	-1.407
y3	0.9625	-0.2712	-76.96	-45.56	56.16	12.87	-8.716
y4	-0.2712	-0.9625	-12.51	21.89	-4.726	20.52	-1.407

	x8	x9	x10
y1	-21.86	13.95	2.647
y2	2.544	16.32	3.887
y3	-21.86	13.95	2.647
y4	2.544	16.32	3.887

d =

	u1	u2	u3	u4
y1	0	0	0	0
y2	0	0	0	0
y3	0	0	0	0
y4	0	0	0	0

Continuous-time model.

Y las graficas de la respuesta paso y SVD son

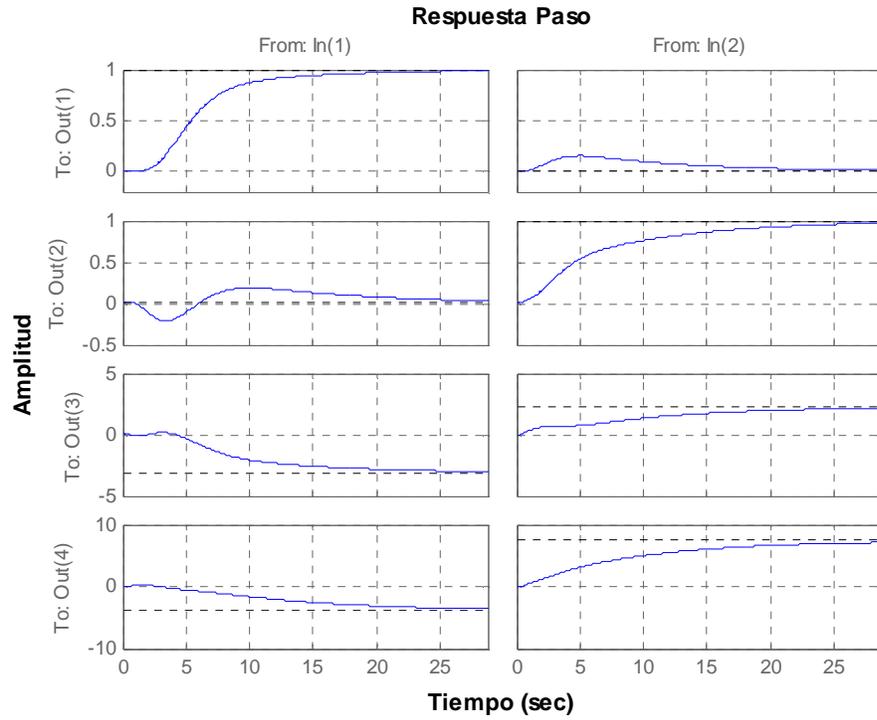


Figura 18: Respuesta Paso de la planta controlada (LQG)

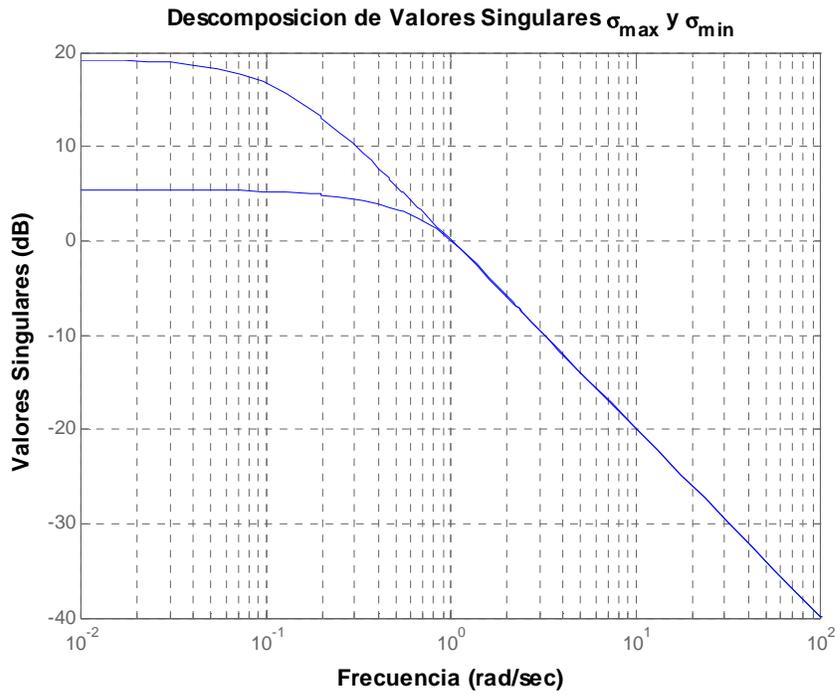


Figura 19: SVD de la planta controlada (LQG)

En la Figura 18 se puede observar la respuesta paso de la planta controlada, junto con las salidas del controlador optimal y en la Figura 19 se observa la SVD máxima y mínima para la planta controlada.

4.3 CONTROL ROBUSTO

Como en los pasos anteriores, se introduce las matrices de estado de la planta a controlar y se selecciona la opción "Control Robusto", tal como se ve en la figura siguiente



Figura 20: Pasos a seguir para diseño de Control Robusto

Ahora se procede a presionar el botón "CALCULAR CONTROLADOR" y se obtiene la siguiente planta en el Command Window

Controlador Robusto:

a =

	x1	x2
x1	-1e-006	0
x2	0	-1e-006

b =

	u1	u2
x1	2	0
x2	0	4

c =

	x1	x2
y1	-1.56	0.5714
y2	-1.939	1.855

d =

	u1	u2
y1	-3.12	2.286
y2	-3.877	7.421

Continuous-time model.

En las Figuras 21 y 22 se observa en control para una planta sin incertidumbre y para una planta con un 20% de incertidumbre

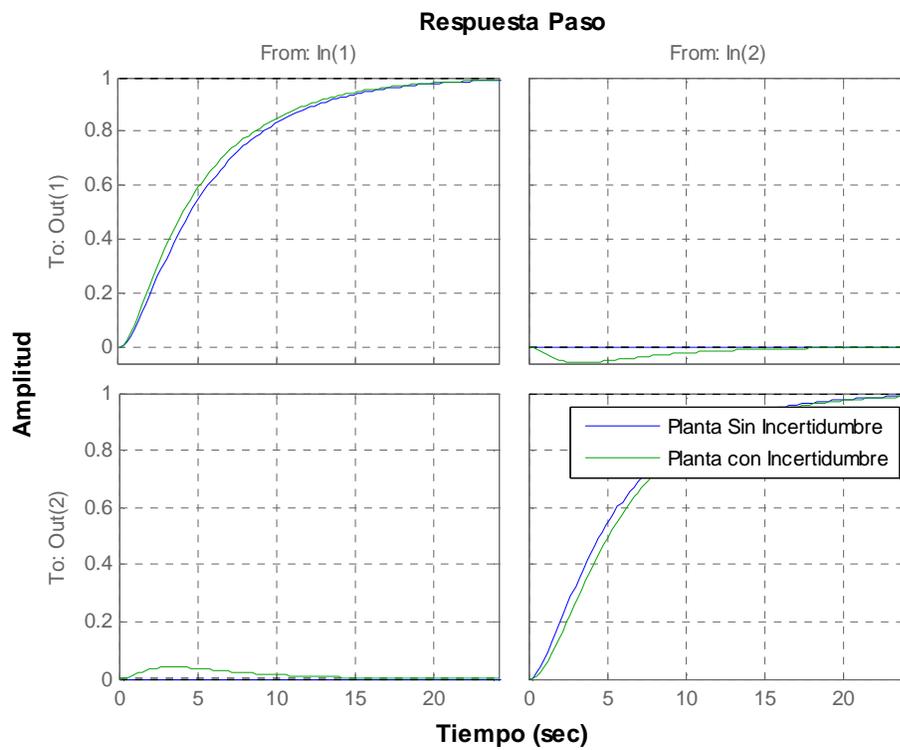


Figura 21: Respuesta Paso de la planta controlada (Control Robusto)

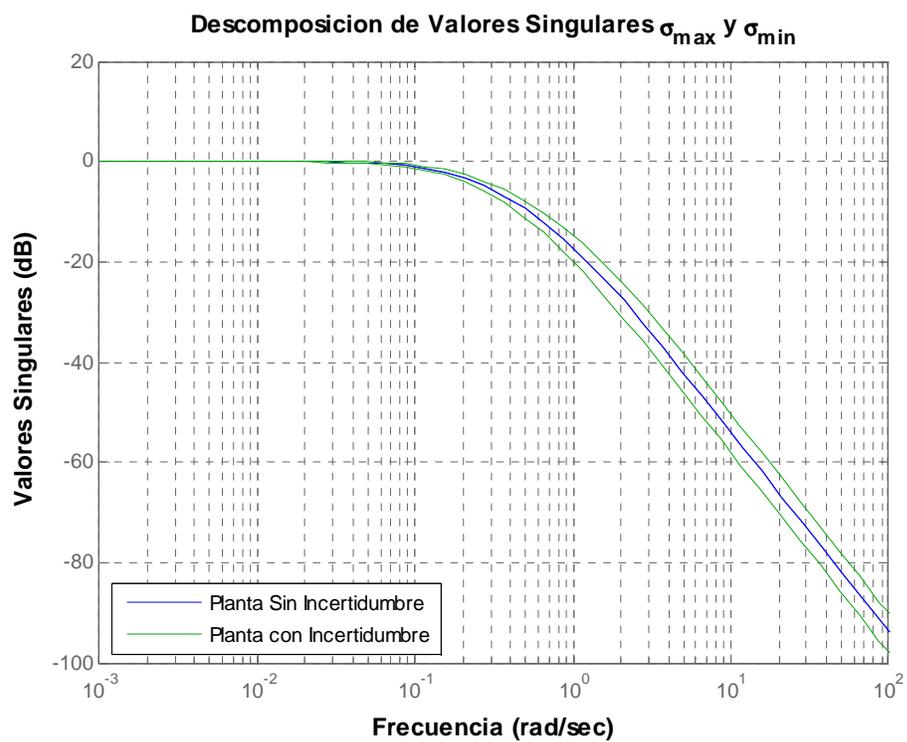


Figura 22: SVD de la planta controlada sin y con incertidumbre (Control Robusto)

4.4 SENSIBILIDAD MIXTA H_∞

Igual que en los procedimientos anteriores se procede a introducir las matrices de estado de la plana y se selecciona la opción “Sensibilidad Mixta Hinf” tal como se ve en la Figura 23.



Figura 23: Pasos a seguir para diseño de Sensibilidad Mixta Hinf

El controlador (matrices de estado) que se obtiene es el siguiente

Controlador de sensibilidad Mixta Hinf:

a =

	x1	x2	x3	x4	x5	
x1	-0.0025	-8.26e-037	8.882e-016	0	0	
x2	-3.304e-036	-0.0025	-3.009e-016	0	0	
x3	-1.6e-019	5.952e-020	-1.217	0	0	
x4	-2.753e-019	-2.976e-020	-2.805	0	0	
x5	-2.009e-019	1.934e-019	-1.889	0	0	
x6	-6.696e-020	-2.381e-019	-1.472	0	0	
x7	5.389e+005	-6.342e+005	-6.102e+006	-1.073e+007	5.507e+006	
x8	-4.487e+004	5.281e+004	5.081e+005	8.936e+005	-4.586e+005	

x9	-1.52e+005	1.789e+005	1.721e+006	3.027e+006	-1.553e+006
x10	1.2e+006	-1.412e+006	-1.359e+007	-2.39e+007	1.226e+007

	x6	x7	x8	x9	x10
x1	2.998e-016	0	0	0	0
x2	8.882e-016	0	0	0	0
x3	-1.472	1	0	0	0
x4	-7.788	0	1	0	0
x5	-1.213	0	0	1	0
x6	-16.06	0	0	0	1
x7	8.418e+005	-6.843e+005	-3.291e+006	-4.483e+005	-7127
x8	-7.012e+004	5.698e+004	2.74e+005	3.732e+004	592.4
x9	-2.374e+005	1.93e+005	9.282e+005	1.265e+005	2017
x10	1.874e+006	-1.524e+006	-7.329e+006	-9.984e+005	-1.592e+004

b =

	u1	u2
x1	3.919	2.937e-016
x2	-2.948e-016	3.919
x3	-1.192	-1.442
x4	-2.748	-7.63
x5	-1.85	-1.189
x6	-1.442	-15.73
x7	-1.787	-5.385
x8	-5.995	-20.27
x9	-2.01	-2.284
x10	-19.52	-127.3

c =

	x1	x2	x3	x4	x5
y1	4.431e+005	-5.215e+005	-5.017e+006	-8.824e+006	4.528e+006
y2	-1.637e+005	1.926e+005	1.853e+006	3.259e+006	-1.672e+006

	x6	x7	x8	x9	x10
y1	6.921e+005	-5.626e+005	-2.706e+006	-3.686e+005	-5860
y2	-2.556e+005	2.078e+005	9.993e+005	1.362e+005	2171

d =

	u1	u2
y1	0	0
y2	0	0

Continuous-time model.

La respuesta paso del correspondiente tipo de diseño se ve en la Figura 24 y la gráfica del SVD se ve en la Figura 25.

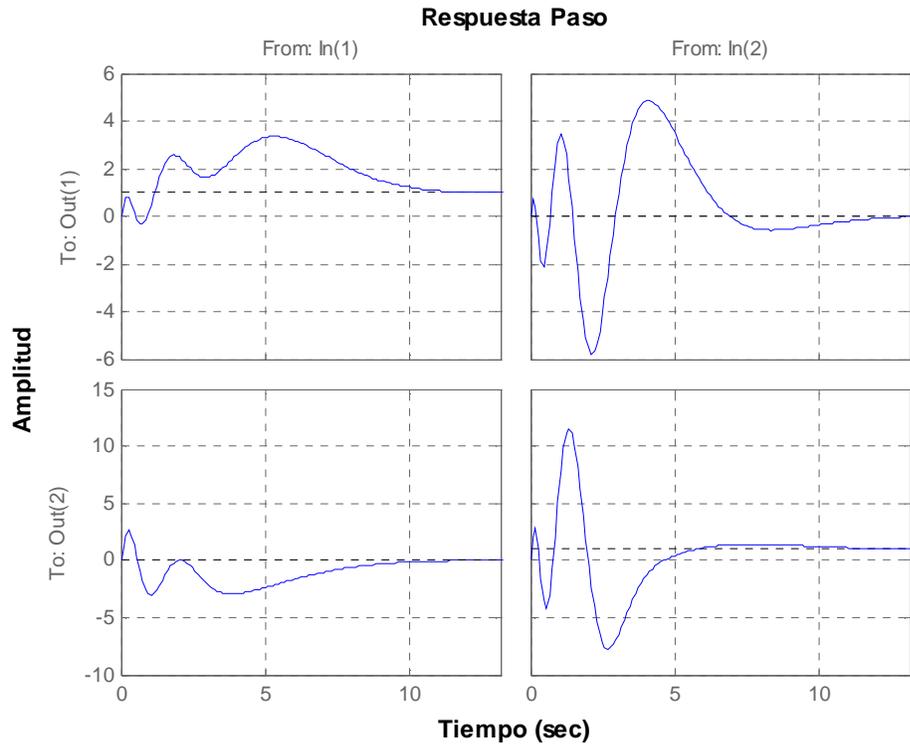


Figura 24: Respuesta Paso de la planta controlada (Sensibilidad Mixta Hinf)

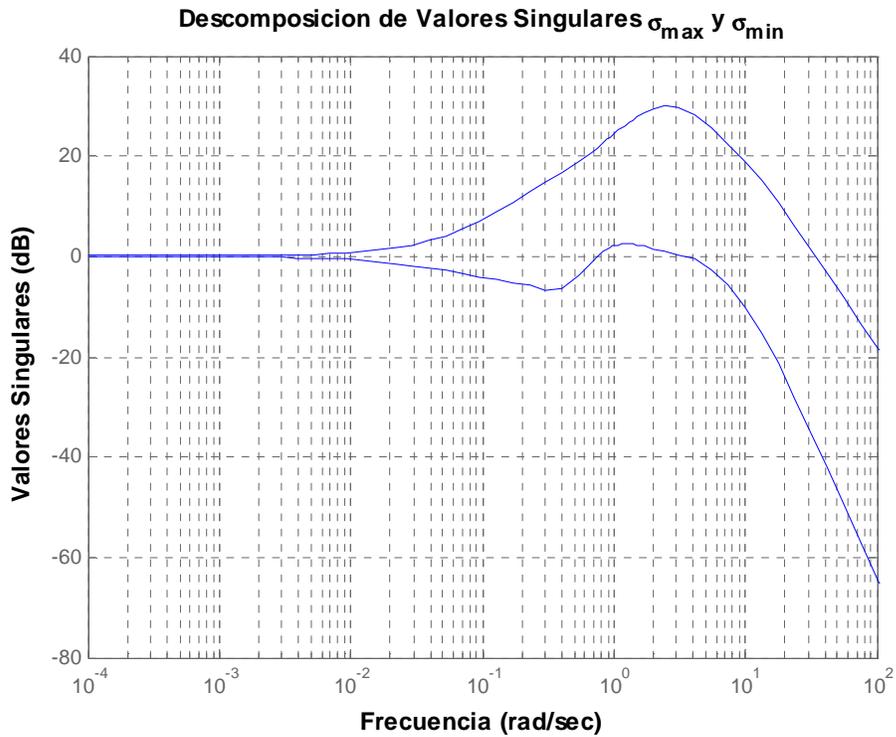


Figura 25: SVD de la planta controlada (Sensibilidad Mixta Hinf)

5. CÓDIGO DE MATLAB

El código implementado para el desarrollo de la aplicación del GUI de MATLAB es el siguiente, este tiene como nombre de archivo el de `profundiMIMO.m`.

```
function varargout = profundiMIMO(varargin)
% PROFUNDIMIMO M-file for profundiMIMO.fig
%   PROFUNDIMIMO, by itself, creates a new PROFUNDIMIMO or raises
%the existing
%   singleton*.
%
%   H = PROFUNDIMIMO returns the handle to a new PROFUNDIMIMO or
%the handle to
%   the existing singleton*.
%
%PROFUNDIMIMO('CALLBACK',hObject,eventData,handles,...) calls the
%local
%function named CALLBACK in PROFUNDIMIMO.M with the given input
%arguments.
%
%PROFUNDIMIMO('Property','Value',...) creates a new PROFUNDIMIMO or
% raises the existing singleton*. Starting from the left, property
%value pairs are
% applied to the GUI before profundiMIMO_OpeningFunction gets called.
% An unrecognized property name or invalid value makes property
%application
%stop. All inputs are passed to profundiMIMO_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help profundiMIMO

% Last Modified by GUIDE v2.5 02-Apr-2007 08:44:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @profundiMIMO_OpeningFcn, ...
                  'gui_OutputFcn',  @profundiMIMO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

% --- Executes just before profundimimo is made visible.
function profundimimo_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to profundimimo (see VARARGIN)

% Choose default command line output for profundimimo
handles.output = hObject;
handles.mydata.indice=1;
handles.mydata.calculado=0;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes profundimimo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = profundimimo_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function ABox_Callback(hObject, eventdata, handles)
% hObject    handle to ABox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
NewStrVal=get(hObject, 'String');          %Almacenar valor ingresado
handles.ABox=NewStrVal;                    %Almacenar en puntero
guidata(hObject,handles);
% Hints: get(hObject,'String') returns contents of ABox as text
%        str2double(get(hObject,'String')) returns contents of ABox as
%a double

% --- Executes during object creation, after setting all properties.
function ABox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ABox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%called

% Hint: edit controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Boton.
function Boton_Callback(hObject, eventdata, handles)
% hObject     handle to Boton (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

a=handles.ABox;
if isnumeric(a)==1
    errordlg('Introduzca la matriz A')
else
    a = str2num(a);
end

b=handles.BBox;
if isnumeric(b)==1
    errordlg('Introduzca la matriz B')
else
    b = str2num(b);
end

c=handles.CBox;
if isnumeric(c)==1
    errordlg('Introduzca la matriz C')
else
    c = str2num(c);
end

d=handles.DBox;
if isnumeric(d)==1
    errordlg('Introduzca la matriz D')
else
    d = str2num(d);
end
Ind=handles.mydata.indice;
G=ss(a,b,c,d);
[mattype,rowd,cold,num]=minfo(G);

if rowd==cold;
% G debe tener por lo menos tantas entradas como salidas

    if Ind==1

        s=zpk('s'); % Variable Laplace (s)
        Gd=20/s; % Curva del lazo deseado
        handles.mydata.Gd=Gd;
        % Se computa el controlador optimo de lazo (K)
        [K CL GAM]=loopsyn(G,Gd);
        handles.mydata.GAM=GAM;
        % LIMITACION: G debe ser estabilizable, detectable, y de
        % rango completo
    end
end

```

```

disp(' ');
disp('Controlador de Diseño por lazo: ');
K
L=G*K;          % Se halla el lazo abierto L
handles.mydata.L=L;
I=eye(size(L));
handles.mydata.I=I;
S=feedback(I,L); % Se halla la sensibilidad (S)
T=I-S;          % Se halla la sensibilidad complementaria (T)
handles.mydata.FT=T;
handles.mydata.calculado=1;
guidata(hObject,handles);

elseif Ind==2

[n,m]=size(b);
% Se halla el vector de estado y los tamaños de control
p=size(c,1); % Se halla el tamaño del vector de salida
% Primero se diseña el regulador de Realimentación de estados
A = [a zeros(n,m) ; -c zeros(m,m)]; B = [b ; -d];
% Planta aumentada con integradores
Q=[zeros(n,n) zeros(n,m); zeros(m,n) eye(m,m)];
% Peso en el error integrado
R=1*eye(m);
% Peso de entrada - R pequeña de respuesta mas rapida
Kr=lqr(A,B,Q,R);
% Se diseña el regulador optimo de realimentacion de estados
Krp = Kr(1:m,1:n); Kri = Kr(1:m,n+1:n+m);
% Se extrae la realimentacion de los estados y el integrador
% Ahora se diseña el filtro de Kalman para la planta
Bnoise = eye(n);
% Ruido del proceso (disturbance) q entra directamente a la
%planta
W = eye(n); % Peso del ruido del proceso
V = 1*eye(m); % Peso de la medicion del ruido
%disminuirlo genera una respuesta mas rapida
Ke = lqe(a,Bnoise,c,W,V); % No hay necesidad de estimar los
%estados del integrador
% Se genera el controlador de 2DOF desde [r y]' a u con
%integradores incluidos
Ac = [ zeros(m,m) zeros(m,n); -b*Kri a-b*Krp-Ke*c];
Bc = [eye(m) -eye(m); zeros(n,m) +Ke];
Cc = [-Kri -Krp]; Dc = [zeros(m,m) zeros(m,m)];
K = ss(Ac,Bc,Cc,Dc);
Gsim = ss(a,b,[c;c],[d;d]);
disp(' ');
disp('Controlador LQG: ');
Ksim = ss(Ac,Bc,[Cc;Cc],[Dc;Dc])
sys = lft(Gsim,-Ksim,1,1);

Bc1=Bc(:,1:m); Bc2=Bc(:,m+1:m+m);

Ac1 = [a b*Cc ; Bc2*c Ac+Bc2*d*Cc ];
Bc1 = [zeros(n,m) ; Bc1 ];
Cc1 = [c d*Cc ; zeros(m,n) Cc];
% Aumentado para incluir la señal de control u
Dc1 = [zeros(m,m); zeros(m,m)];
T=ss(Ac1,Bc1,Cc1,Dc1);

```

```

handles.mydata.FT=T;
handles.mydata.calculado=2;
guidata(hObject,handles);

elseif Ind==3

G0=dcgain(G);
dyn = tf(1,[1 1]); % Planta con integrador
G=dyn*G0;

% Controlador Inverso
dynk = tf([1 1],[1 1.e-6]);
disp(' ');
disp('Controlador Robusto: ');
Kinv = (dynk*inv(G0));
ss(Kinv)
I2=eye(length(G*Kinv));

% Planta Nominal
S=minreal(inv(I2+G*Kinv));
T=minreal(I2-S);

% Con 20% de Incertidumbre a la entrada
Unc = [1.2 0; 0 0.8];
Lu=G*Unc*Kinv;
Su=minreal(inv(I2+Lu));
Tu=minreal(I2-Su);

% Simulacion en el tiempo
Kr=tf(1,[5 1]);
Tr=T*Kr; % Sensibilidad complementaria sin incertidumbre
Tru=Tu*Kr; % Sensibilidad complementaria con incertidumbre
handles.mydata.FT=Tr;
handles.mydata.FT2=Tru;
handles.mydata.calculado=3;
guidata(hObject,handles);

elseif Ind==4

G=ss(a,b,c,d);

% Diseño S-KS H-INFINITO
% Se require wb, pendiente 1 y pico menor q M
Mi=1.5; wbi=25; Ai=1e-4;
wpi = tf([1/Mi wbi], [1 wbi*Ai]);
Wp=[wpi 0;0 wpi]; Wu=1.0*eye(2);
[khinf1 hinfl gopt1] = mixsyn(G,Wp,Wu,[]);
disp(' ');
disp('Controlador de sensibilidad Mixta Hinf: ');
K=khinf1

S=inv(eye(2)+G*K);
T=eye(2)-S;
handles.mydata.FT=T;
handles.mydata.calculado=4;
guidata(hObject,handles);
end

```

```

else
    errordlg('La planta debe tener la misma ...
            cantidad de entradas y salidas');
end

function CBox_Callback(hObject, eventdata, handles)
% hObject    handle to CBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
NewStrVal=get(hObject,'String'); %Almacenar valor ingresado
handles.CBox=NewStrVal;         %Almacenar en puntero
guidata(hObject,handles);

% Hints: get(hObject,'String') returns contents of CBox as text
%        str2double(get(hObject,'String')) returns contents of CBox as
%a double

% --- Executes during object creation, after setting all properties.
function CBox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to CBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function BBox_Callback(hObject, eventdata, handles)
% hObject    handle to BBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
NewStrVal=get(hObject,'String'); %Almacenar valor ingresado
handles.BBox=NewStrVal;         %Almacenar en puntero
guidata(hObject,handles);

% Hints: get(hObject,'String') returns contents of BBox as text
%        str2double(get(hObject,'String')) returns contents of BBox as
%a double

% --- Executes during object creation, after setting all properties.
function BBox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to BBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function DBox_Callback(hObject, eventdata, handles)
% hObject    handle to DBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
NewStrVal=get(hObject,'String'); %Almacenar valor ingresado
handles.DBox=NewStrVal;          %Almacenar en puntero
guidata(hObject,handles);
% Hints: get(hObject,'String') returns contents of DBox as text
%       str2double(get(hObject,'String')) returns contents of DBox as
%a double

% --- Executes during object creation, after setting all properties.
function DBox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Salir.
function Salir_Callback(hObject, eventdata, handles)
close
% hObject    handle to Salir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function uipanel1_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to uipanel1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

switch get(hObject,'Tag') % Obtener etiqueta del objeto seleccionado
case 'LOOPTYNT'
    handles.mydata.indice=1;
    guidata(hObject,handles);
case 'LQG'
    handles.mydata.indice=2;
    guidata(hObject,handles);
case 'ROBUSTO'

```

```

        handles.mydata.indice=3;
        guidata(hObject,handles);
    case 'MIXTO'
        handles.mydata.indice=4;
        guidata(hObject,handles);
    otherwise
        handles.mydata.indice=0;
        guidata(hObject,handles);
end

% --- Executes on button press in STEPF.
function STEPF_Callback(hObject, eventdata, handles)
% hObject    handle to STEPF (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Ind=handles.mydata.indice;
if handles.mydata.indice==handles.mydata.calculado;
    figure(1)
    if Ind==3
        l=stepinfo(handles.mydata.FT);
        T=l.SettlingTime;
        step(handles.mydata.FT,1.2*T)
        hold on
        l=stepinfo(handles.mydata.FT2);
        T=l.SettlingTime;
        step(handles.mydata.FT2,1.2*T)
        hold off
        legend('Planta Sin Incertidumbre','Planta con Incertidumbre')
    else
        l=stepinfo(handles.mydata.FT);
        T=l.SettlingTime;
        step(handles.mydata.FT,1.2*T)
    end
    grid;
    title('Respuesta Paso','FontSize',10,'FontWeight','Bold');
    xlabel('Tiempo','FontSize',10,'FontWeight','Bold');
    ylabel('Amplitud','FontSize',10,'FontWeight','Bold');
else
    errordlg('No se ha calculado el controlador!')
end

% --- Executes on button press in SVDF.
function SVDF_Callback(hObject, eventdata, handles)
% hObject    handle to SVDF (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Ind=handles.mydata.indice;
if handles.mydata.indice==handles.mydata.calculado;
    if Ind==1
        I=handles.mydata.I;
        L=handles.mydata.L;
        T=handles.mydata.FT;
        Gd=handles.mydata.Gd;
        GAM=handles.mydata.GAM;
        figure(2)
        sigma(I+L,'--',T,':',L,'r--',Gd,'k-.',...

```

```

Gd/GAM, 'k:', Gd*GAM, 'k:');
legend('1/\sigma(S) Desempeño', ...
'\sigma(T) Robustez', ...
'\sigma(L) Lazo Abierto', ...
'\sigma(Gd) Lazo deseado', ...
'\sigma(Gd) \pm GAM, dB');

elseif Ind==3
    Tr=handles.mydata.FT;
    Tru=handles.mydata.FT2;
    figure(2)
    sigma(Tr,Tru)
    legend('Planta Sin Incertidumbre', 'Planta con
Incertidumbre', 'Location', 'SouthWest')
else
    figure(2)
    sigma(handles.mydata.FT);
end
grid;
title('Descomposicion de Valores Singulares ...
\sigma_{m_a_x} y \sigma_{m_i_n}', 'FontSize', 10, 'FontWeight', 'Bold')
xlabel('Frecuencia', 'FontSize', 10, 'FontWeight', 'Bold')
ylabel('Valores Singulares', 'FontSize', 10, 'FontWeight', 'Bold')
else
    errordlg('No se ha calculado el controlador!')
end

% PABLO JOSÉ PEÑUELA ROJAS --- UNIVERSIDAD SURCOLOMBIANA / 2007

```

6. CONCLUSIONES

Se realizó la aplicación de Control Multivariable en MATLAB, utilizando este trabajo como una aplicación introductoria a la teoría de Control Moderno, siendo más preciso, al Control Multivariable, pudiéndose convertir en una herramienta útil para el aprendizaje de dicho tema.

Se hace uso de las diferentes herramientas de control que posee MATLAB en sus "Toolbox", entendiéndose que este software puede ser utilizado como una herramienta especializada en el área de control, esta herramienta presenta una gran facilidad de uso debido a su naturaleza matricial, la cual corresponde a la naturaleza del modelo dinámico de los procesos, el cual es un conjunto de matrices de ecuaciones de estado.

Dentro de la extensa teoría de Control Multivariable se hizo hincapié en algunas de las técnicas de Control Centralizado, notando así, que para cada tipo de planta o proceso puede existir más de una configuración que se ajusta o no a las especificaciones del control.

Nos dimos cuenta que la Descomposición de Valores Singulares (SVD) y el Arreglo Relativo de Ganancia (RGA) son unas herramientas muy útiles porque nos permiten modelar el lazo abierto y así obtener un desempeño deseado. Lo cual nos lleva a técnicas como Diseño de lazo (Loop Shaping), donde el comportamiento de la planta se ciñe al comportamiento de otro proceso por medio de un controlador.

De las técnicas de diseño notamos que para plantas MIMO no es siempre fácil obtener un desempeño óptimo para todas las frecuencias y hay que entender que se hace necesario realizar "sacrificios" en cuanto al desempeño y por ende diseñar un controlador que se "ajuste" de mejor forma a nuestras exigencias.

Dimos uso de una de las técnicas más utilizadas en control, el control óptimo, que puede ser también implementado para plantas MIMO, aunque este tipo de controlador presenta problemas para plantas que poseen incertidumbre, es decir que no se conocen exactamente.

Se entendió que el concepto de incertidumbre no solo hace referencia a errores causados por desconocimiento de la totalidad de las características de la planta en el momento de modelarla, sino que también incluye todos esos errores debido a aproximaciones de parámetros y además de reducciones del orden de los modelos.

Se ha dado por entendido que el tema de Control Multivariable es un tema bastante extenso porque cubre la totalidad de temas del Control Clásico y además agrega otras teorías y complejidades que no se tienen en cuenta para teoría SISO. Lo hecho aquí se considera como un trabajo introductorio para el control MIMO y no pretende manejar la totalidad de los temas y complejidades que puede ofrecer el mundo de los modelos MIMO.

BIBLIOGRAFIA

SKOGESTAD, Sigurd. POSTLETHWAITE, Ian. Multivariable Feedback Control. Segunda edición. Inglaterra: Jhon Wiley & Sons, 1996. 595 p.

ALBERTOS, Pedro. SALA, Antonio. Multivariable Control Systems: An Engineering Approach. Primera Edición. Inglaterra: Springer-Verlag, 2004. 358p.

OGATA, Katsuhiko. Ingeniería de control moderna. Tercera edición. México: Prentice – Hall, 1994. 946 p.

The MATHWORKS, Inc. Robust Control Toolbox User's Guide. Versión 3.1.1. Estados Unidos de América, 2006. 655 p.

The MATHWORKS, Inc. Control Systems Control Toolbox User's Guide. Estados Unidos de América, 1998. 535 p.

The MATHWORKS, Inc. Creating Graphical Users Interface. Versión 7.3. Estados Unidos de América, 2006. 442 p.

The MATHWORKS, Inc. μ -Analysis and Synthesis Toolbox User's Guide. Versión 3.0.6. Estados Unidos de América, 2001. 734 p.

Curso de Sistemas Multivariable. Disponible en Internet:
<http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-245Spring2004/CourseHome/index.htm>

Revista Iberoamericana de Automática e Informática Industrial. Volumen 2. Número 5, Enero 2005, 4-18 p.

Subotin, Maxim. Balancing an inverted pendulum on a seesaw.

ANEXOS

A. Matriz identidad

En álgebra lineal, la matriz identidad es una matriz que cumple la propiedad de ser el elemento neutro del producto de matrices. Esto quiere decir que el producto de cualquier matriz por la matriz identidad (donde dicho producto esté definido) no tiene ningún efecto. La columna i -ésima de una matriz identidad es el vector unitario e_i de una base vectorial inmersa en un espacio Euclídeo de dimensión n .

Como el producto de matrices sólo tiene sentido si sus dimensiones son compatibles, existen infinitas matrices identidad dependiendo de las dimensiones. I_n , la matriz identidad de tamaño n , se define como la matriz diagonal que tiene 1 en cada una de las entradas de la diagonal principal, y 0 en el resto. Así,

$$I_1 = (1), I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \dots, I_n = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Empleando la notación que a veces se usa para describir concisamente las matrices diagonales, resulta:

$$I_n = \text{diag}(1, 1, \dots, 1)$$

Si el tamaño es inmaterial, o se puede deducir de forma trivial por el contexto, entonces se escribe simplemente como I .

También se puede escribir usando la notación delta de Kronecker:

$$I_{ij} = \delta_{ij}$$

o, de forma aún más sencilla,

$$I = (\delta_{ij})$$

La matriz identidad de orden n puede ser también considerada como la matriz permutación que es elemento neutro del grupo de matrices de permutación de orden $n!$.

B. Operador Norma

Sea V un espacio vectorial afín sobre un cuerpo \mathbb{K} y \vec{x} un vector del espacio. Se dice que $\|\cdot\| : V \rightarrow \mathbb{K}$ es un operador que define la norma de \vec{x} , y escribimos $\|\vec{x}\|$, si cumple:

1. Para todo \vec{x} de V su norma ha de ser positiva, y será cero si y sólo si \vec{x} es el vector cero: $0 < \|\vec{x}\|$ si $\vec{x} \neq \vec{0}$ y $\|\vec{x}\| = 0 \iff \vec{x} = \vec{0}$.
2. Para todo \vec{x} de V y para todo k de \mathbb{K} se satisface que $\|k\vec{x}\| = |k| \cdot \|\vec{x}\|$.
3. Para todos \vec{x} e \vec{y} de V se cumple que $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$ (desigualdad triangular).

Cualquier operador que cumpla estas tres condiciones, y en cualquier geometría, será un operador norma.

Con esta definición matemática general y recurriendo a la definición de producto escalar, observamos que en un espacio euclídeo el operador norma habitual se puede definir:

$$\|\vec{x}\| = \sqrt{\langle \vec{x}, \vec{x} \rangle}$$

Esto permite ampliar el cálculo de una longitud a otras geometrías de forma que la operación, en su método básico habitual (el cuadrado escalar), no cambie

Operador norma

A continuación se muestran algunos otros ejemplos de posibles operadores norma (que satisfacen la definición matemática general):

- Para un vector $\vec{x} = (x_1, x_2, \dots, x_n)$ se define la norma p como:

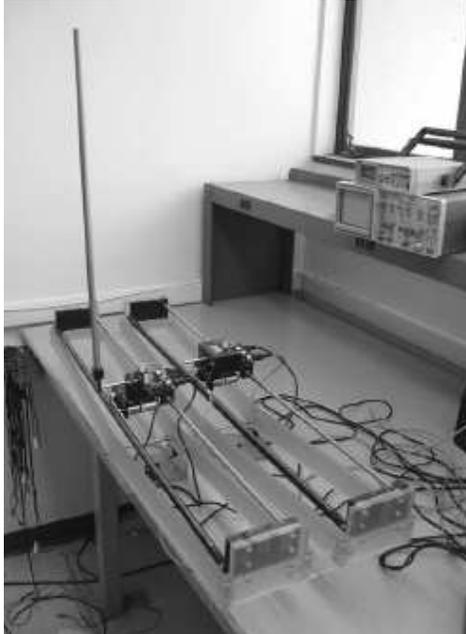
$$\|\vec{x}\|_p = \sqrt[p]{|x_1|^p + |x_2|^p + \dots + |x_n|^p}$$

Así, para el caso $p = 1$ se obtiene $\|\vec{x}\|_1 = |x_1| + |x_2| + \dots + |x_n|$, y para el caso $p = 2$ se obtiene la norma usual euclídea

- Otro operador norma sería $\|\vec{x}\| = \max(|x_1|, |x_2|, \dots, |x_n|)$, donde $\vec{x} = (x_1, x_2, \dots, x_n)$.

C. Péndulo Invertido en un columpio

Se tiene un péndulo invertido instalado en un carro que se mueve a lo largo de un riel que se encuentra sobre un columpio, además se instala un segundo carro en el columpio que se mueve en paralelo con el primer carro. El sistema resultante tiene dos entradas: una fuerza aplicada al carro con el péndulo instalado encima de él y una fuerza aplicada al carro de carga el cual trata de balancear la estructura del columpio:

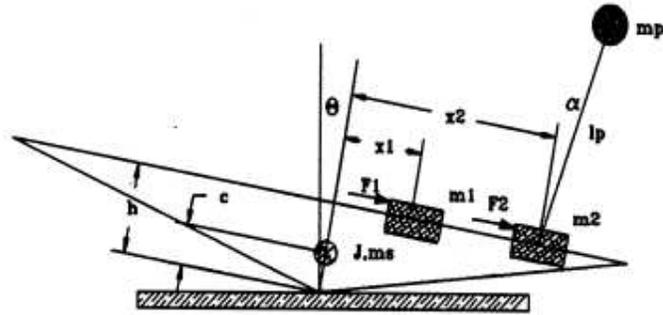


Fuente: Subbotin, Maxim. Balancing an Inverted pendulum on a seesaw

Con dos entradas y cuatro salidas disponibles, se define el sistema como un modelo inestable MIMO.

Para su modelado se hace uso de la aproximación de energía de LaGrange. Primero se determina el LaGrangiano para el cálculo de las energías potencial y cinética. Como el sistema posee cuatro grados de libertad se escogen cuatro coordenadas LaGrangianas: x_1 - posición del carro de carga, Θ - ángulo de la estructura del columpio con respecto al eje vertical, x_2 - posición del carro del péndulo y α - ángulo del péndulo con respecto al eje de simetría del columpio.

El diagrama simplificado del péndulo en un columpio es el siguiente:



Fuente: Subbotin, Maxim. Balancing an Inverted pendulum on a seesaw

El Lagrangiano del sistema está definido como

$$L = K - P,$$

Donde K y P son las energías totales cinéticas y potenciales del sistema respectivamente. Estas son calculadas como las sumas de las energías potenciales y cinéticas de los elementos correspondientes

$$K = K_p + K_{m_2} + K_{m_1} + K_s$$

$$P = P_p + P_{m_2} + P_{m_1} + P_s$$

Aquí las energías potenciales del péndulo, del carro del péndulo, del carro de carga y de la construcción del columpio son respectivamente

$$P_p = m_p g (h \cos \theta - x_2 \sin \theta + l_p \cos(\theta + \alpha));$$

$$P_{m_2} = m_2 g (h \cos \theta - x_2 \sin \theta);$$

$$P_{m_1} = m_1 g (h \cos \theta - x_1 \sin \theta);$$

$$P_s = m_s g c \cos \theta.$$

Y las energías cinéticas son

$$K_p = 0.5m_p \left\{ \left((h \cos \theta - x_2 \sin \theta + l_p \cos(\theta + \alpha)) \dot{\theta} + \dot{x}_2 \cos \theta + l_p \cos(\theta + \alpha) \right)^2 + \left((-h \sin \theta - x_2 \cos \theta - l_p \sin(\theta + \alpha)) \dot{\theta} - \dot{x}_2 \sin \theta - l_p \sin(\theta + \alpha) \dot{\alpha} \right)^2 \right\};$$

$$K_{m_2} = 0.5m_2 \left[\left(\dot{x}_2 + h \dot{\theta} \right)^2 + \left(x_2 \dot{\theta} \right)^2 \right];$$

$$K_{m_1} = 0.5m_2 \left[\left(\dot{x}_1 + h \dot{\theta} \right)^2 + \left(x_1 \dot{\theta} \right)^2 \right];$$

$$K_s = 0.5J \dot{\theta}^2.$$

En las ecuaciones anteriores los parámetros del sistema son definidas como sigue: $m_1 = 0.815[Kg]$ - masa del carro de la carga, $m_2 = 0.455[Kg]$ - masa del carro del péndulo, $m_p = 0.21[Kg]$ - masa del péndulo, $m_s = 6.6[Kg]$ - masa de la construcción del columpio. $J = 0.854[Kg \cdot m^2]$ - momento de inercia de las construcción del columpio alrededor de su centro de masa, $l_p = 0.305[m]$ - centro de masa del péndulo (mitad de la longitud), $h = 0.14[m]$ - altura del riel desde el eje del columpio, $c = 0.0585[m]$ - centro de masa de la construcción del columpio.

A partir del LaGrangiano el sistema puede ser definido de

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{x}_1} - \frac{\partial L}{\partial x_1} = F_1; \quad \frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{x}_2} - \frac{\partial L}{\partial x_2} = F_2;$$

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0; \quad \frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = 0.$$

Las fuerzas aplicadas a los carros están relacionadas con los voltajes de control de los motores DC por

$$F_i = \frac{K_m K_g}{Rr} V_i - \frac{K_m^2 K_g^2}{Rr^2} \dot{x}_i,$$

donde $K_m = 0.00767[Nm/A]$ - constante del motor, $K_g = 3.7[N/A]$ - radio del motoreductor, $R = 2.6[\Omega]$ - resistencia de la armadura del motor, $r = 0.00635[m]$ - radio del piñón del motor.

De lo anterior podemos determinar las ecuaciones no lineales que describen la

dinámica del sistema para $\ddot{x}_1, \ddot{x}_2, \ddot{\theta}, \ddot{\alpha}$. Una vez las ecuaciones han sido encontradas, se linealiza alrededor del punto de operación

$\dot{x}_1 = \dot{x}_2 = \dot{\theta} = \dot{\alpha} = x_1 = x_2 = \theta = \alpha = 0$, $V_1 = V_2 = 0$ y se obtiene un modelo de espacio de estados para el sistema con el vector de estados

$x = \begin{bmatrix} x_1 & \theta & x_2 & \alpha & \dot{x}_1 & \dot{\theta} & \dot{x}_2 & \dot{\alpha} \end{bmatrix}^T$, la entrada de control

$u = \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T$ y una salida $y = \begin{bmatrix} x_1 & \theta & x_2 & \alpha \end{bmatrix}^T$.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

Las matrices A , B y C fueron definidas como sigue

$$\begin{aligned} A &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}; \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}; \quad C = \begin{bmatrix} I^{4 \times 4} & 0^{4 \times 4} \end{bmatrix}; \\ A_{11} &= 0^{4 \times 4}; \quad A_{12} = I^4; \quad B_1 = 0^{4 \times 2}; \\ A_{21} &= \begin{bmatrix} -\frac{m_1 h g}{J} & -g \frac{h m_s c - J}{J} & -g h \frac{m_2 + m_p}{J} & 0 \\ \frac{m_1 g}{J} & \frac{m_s c g}{J} & g \frac{m_2 + m_p}{J} & 0 \\ -\frac{m_1 h g}{J} & -g \frac{h m_s c - J}{J} & -g h \frac{m_2 + m_p}{J} & -\frac{m_p g}{m_2} \\ -\frac{m_1 g}{J} & -\frac{m_s c g}{J} & -g \frac{m_2 + m_p}{J} & g \frac{m_2 + m_p}{m_2 l_p} \end{bmatrix}; \end{aligned}$$

$$A_{22} = \begin{bmatrix} -\frac{K_m^2 K_g^2}{r^2 R} \frac{J + m_1 h^2}{m_1 J} & 0 & -\frac{K_m^2 K_g^2}{r^2 R} \frac{h^2}{J} & 0 \\ \frac{K_m^2 K_g^2}{r^2 R} \frac{h}{J} & 0 & \frac{K_m^2 K_g^2}{r^2 R} \frac{h}{J} & 0 \\ -\frac{K_m^2 K_g^2}{r^2 R} \frac{h^2}{J} & 0 & -\frac{K_m^2 K_g^2}{r^2 R} \frac{J + m_2 h^2}{J} & 0 \\ -\frac{K_m^2 K_g^2}{r^2 R} \frac{h}{J} & 0 & -\frac{K_m^2 K_g^2}{r^2 R} \frac{l_p h m_2 - J}{m_2 J l_p} & 0 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} \frac{J + m_1 h^2}{m_1 J} & \frac{h^2}{J} \\ -\frac{h}{J} & -\frac{h}{J} \\ \frac{h^2}{J} & \frac{J + m_2 h^2}{J} \\ \frac{h}{J} & \frac{l_p h m_2 - J}{m_2 J l_p} \\ \frac{h}{J} & \frac{h^2}{J} \end{bmatrix}.$$