CINVESTAV-IPN
CENTRO DE INVESTIGACION Y ESTUDIOS AVANZADOS DEL IPN

HERRAMIENTAS COMPUTACIONALES PARA ANALISIS DE SISTEMAS EN CONTROL AUTOMATICO

TESIS QUE PRESENTA

ING. JUAN CARLOS GONZALEZ CASTOLO

PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS

EN LA ESPECIALIDAD DE

INGENIERIA ELECTRICA

Guadalajara. Jal., 1998

DIRECTOR DE TESIS

DRA. OFELIA BEGOVICH MENDOZA

Índice General

RESUMEN

En esta tesis se crea un paquete de herramientas computacionales para facilitar el análisis y diseño de sistemas de control automático. Las funciones integradas enriquecen a los programas comerciales de control existentes ("Control Toolboxes"). El software desarrollado utiliza un lenjuage simbólico sobre la plataforma de Matlab 5.0. Las funciones básicas que forman el paquete permiten: descomponer una matriz de transferencia en su parte estrictamente propia, factorizar una matriz de transferencia como una descripción en fracción matricial (MFD[1]), obtener realizaciones en variables de estado a partir de una MFD y encontrar los polos y ceros de sistemas multivariables. Un manual de usuario del paquete es incluido en este trabajo.

[1]MFD es una abreviación del ingles "Matrix-Fraction Descriptions" Se adopta está abreviatura debido a su extenso uso dentro del lenguaje que se usa en la literatura de control.

NOTACION

$\mathbb{R}^n$	Conjunto de vectores con elementos reales
$\mathbb{R}^p[s]$	Conjunto de vectores polinomiales con coeficientes reales
$\mathbb{R}^p(s)$	Conjunto de vectores racionales con coeficientes reales
$\mathbb{R}^{p\times m}$	Conjunto de matrices con elementos reales
$\mathbb{R}^{p\times m}[s]$	Conjunto de matrices polinomiales con coeficientes reales
$\mathbb{R}^{p\times m}(s)$	Conjunto de matrices racionales con coeficientes reales
Vi	Vector V de dimensión i
A_{ij}	Matriz A de dimensión $i \times j$
a_i	i-ésimo elemento del vector a
a_{ij}	Elemento de la i-ésima fila y la j-ésima columna de la matriz A
A^T	Matriz transpuesta de A
A^{-1}	Matriz inversa de A
I_n	Matriz identidad de dimensión $n \times n$
$det(A)$	Determinante de la matriz A
$deg(h)$	Grado del polinomio h
$\mathcal{C}\{A, B\}$	Matriz de controlabilidad del par $\{A, B\}$
$\mathcal{O}\{C, A\}$	Matriz de observabilidad del par $\{A, C\}$

Capítulo 1

Introducción

Este trabajo trata de la creación de herramientas computacionales para el análisis de sistemas de control lineal invariantes en el tiempo de múltiples entradas y múltiples salidas (o multivariables[1]). Se muestran algunas de las técnicas utilizadas para el análisis de estos sistemas, como por ejemplo: para obtener una realización en variables de estado a partir de una matriz de transferencia y que una vez obtenida se pueden analizar diferentes propiedades del sistema de ecuaciones, como lo son, la observabilidad y controlabilidad. Para efectuar el análisis, es necesario manejar la teoría de matrices polinomiales.

En el proceso de análisis y diseño de sistemas de control, los programas de cálculo son herramientas útiles; esto es debido a que el software agiliza y hace menos susceptible a errores el desarrollo de las operaciones. Actualmente existen muchas nuevas herramientas matemáticas y de manipulación de datos en áreas tan variadas como ingeniería eléctrica, economía y finanzas, ingeniería mecánica, matemáticas, ciencias médicas, entre otras. La evolución tan rápida de éstas herramientas, es debido en gran medida, a la aplicación de los ordenadores y programas de desarrollo, tales como Matlab [**18**], Maple [**23**] [**10**] [**20**], etc. Entre las innovaciones tenemos el uso de matemática simbólica, lo cual hace que muchos algoritmos[2] puedan programarse[3] en forma algebráica. Dentro del área de ingeniería eléctrica y concretamente en el campo de sistemas de control, se encuentran diversas universidades e investigadores desarrollando herramientas que ayuden en los métodos de análisis, de sistemas de control [**22**] [**13**]. El problema de programación de un algoritmo de control "X" es de sumo interés, ya que una vez solucionado es aplicado en la solución de problemas de ingeniería y, adicionalmente, puede ser utilizado en la educación como apoyo pedagógico para los profesores de éste campo [**26**] [**6**]; es decir, que los conceptos teóricos pueden ser introducidos, a los estudiantes, por medio del uso de la computadora.

Dentro de las técnicas de análisis antes mencionadas, se aplican algunos algoritmos que se encuentran en la literatura clásica de la ingeniería de control y que sin embargo, no están integrados aún como funciones dentro del software comercial, como Matlab [**18**], Maple [**23**],

[1] Los Sistemas de Control multivariables son comúnmente conocidos como sistemas MIMO ("Multi-Input Multi-Output")

[2] Un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema [**2**].

[3] Programar es plasmar una serie de instrucciones ordenadas lógicamente que resumen de forma óptima la secuencia necesaria para activar un proceso informático [**2**].

Mathematica [**31**]. Por ejemplo, la obtención de una realización en el espacio de estado de una matriz de transferencia, la obtención de una descripción en fracción matricial de una matriz de transferencia, la obtención de los polos y ceros de un sistema multivariable. En éste último ejemplo, cabe decir que se obtienen los ceros correspondientes a varias clasificaciones, como son: los ceros invariantes, los ceros de transmisión, los ceros de desacoplamiento de entrada, los ceros de desacoplamiento de salida y también el número de ceros al infinito.

Como una necesidad de efectuar el análisis de los sistemas lineales MIMO, de una manera rápida y evitando los errores de mano, se han desarrollado en base a estos algoritmos, una caja de herramientas integrada por una serie de funciones, las cuales operan bajo la plataforma de Matlab 5.0. Este trabajo se organiza de la siguiente forma.

El capítulo dos presenta un panorama general. Se parte de una descripción coloquial de un sistema monovariable y un sistema multivariable para de ahí exponer, por analogía, la complejidad de analizar los sistemas multivariables en comparación con los monovariables. También se da un marco teórico en el cual se ubica éste trabajo y se exponen algunos problemas donde resultaría práctico disponer de una biblioteca de programas para ayudar en su solución. Se determina que software utilizar y por qué inclinarse por programar algoritmos en computación simbólica.

En capítulo tres se presenta la teoría formal de algunos métodos de análisis de sistemas de control multivariables y se intercalan, a lo largo de esté, algoritmos descritos en seudo-código [**8**] [**29**].

En el cuarto capítulo se presenta un manual de usuario de los algoritmos programados. Estos algoritmos, integrados como funciones dentro de una caja de herramientas, están ordenados alfabeticamente. Para cada uno de ellos se da el propósito, se muestra la sintaxis y se describe brevemente; también, se da un ejemplo simple que ilustre la manera de hacer uso de la función y finalmente se da una lista de otras funciones con las que se relaciona. En este capítulo se presenta una tabla de referencia rápida y se enumeran los pasos que debe seguir el usuario para habilitar la caja de herramientas.

En el capítulo cinco se discuten las funciones presentadas en esta tesis con otros trabajos y se incluyen, también, ejemplos de aplicaciones para resaltar lo práctico que resulta hacer uso de las herramientas computacionales presentadas en está tesis.

Es importante decir que al inicio de éste proyecto no se tenía conocimiento de la existencia, en software comercial, de ninguna de las herramientas aquí elaboradas. Sin embargo, existen investigadores [**13**] que actualmente continúan desarrollando herramientas para el análisis de sistemas de control, por lo que es muy factible que a la fecha, algunas de las herramientas programadas existan o estén por existir en algún paquete comercial. En el capítulo cinco, se discuten brevemente las herramientas desarrolladas por el Dr. Kwakeernak y las presentadas en ésta tesis, haciendo énfasis en su diferencia.

Finalmente en el capítulo seis se dan las conclusiones y se muestra un panorama de los campos abiertos para trabajos futuros.

Capítulo 2

Panorama general

Imaginemos que una persona (operador) está encargada(o) de regular la válvula de gas de un quemador el cual calienta un tanque de agua. El problema del operador es mantener la temperatura del agua a un valor prefijado. Si pensamos que el operador no sólo se encarga de mantener la temperatura del tanque, sino que también debe mantener el nivel del agua del tanque, o peor aún, mantener la temperatura y el nivel del agua de un sistema de dos tanques acoplados, se puede intuir, que el problema del operador se complica conforme se incrementan las variables que debe controlar. En la primer caso, se presenta un sistema de control de una sola variable manipulada (flujo de gas) y una sola variable controlada (temperatura del tanque), es decir un sistema escalar[1]. En los siguientes casos se tiene un sistema de control con más de una variable controlada (temperatura, nivel) y más de una variable manipulada (flujo de gas, flujo de agua), es decir un sistema MIMO. Si bien es cierto que al operador no le es tan sencillo controlar los sistemas MIMO, también es cierto que el ingeniero de control deberá utilizar herramientas más sofisticadas para el análisis y diseño de estos sistemas; lo anterior comparativamente con el caso de los sistemas SISO.

2.1 Planteamiento del problema

Ahora bien, para facilitar la tarea del operador, el ingeniero de sistemas de control puede diseñar un sistema de control automático.

2.1.1 El proceso de diseño de sistemas de control

Este procedimiento debe seguir los siguientes pasos [**27**].

1.- Se estudia el sistema(planta) a ser controlado y se obtiene información inicial a cerca de los objetivos del control.
2.- Se obtiene el modelo del sistema y se simplifica el modelo, si es necesario.
3.- Se analiza el modelo; se determinan sus propiedades.
4.- Se decide cuales variables son controladas (salidas controladas).

[1] Los Sistemas de Control escalares son comúnmente conocidos como sistemas monovariables o como sistemas SISO ("Single-Input Single-Output")

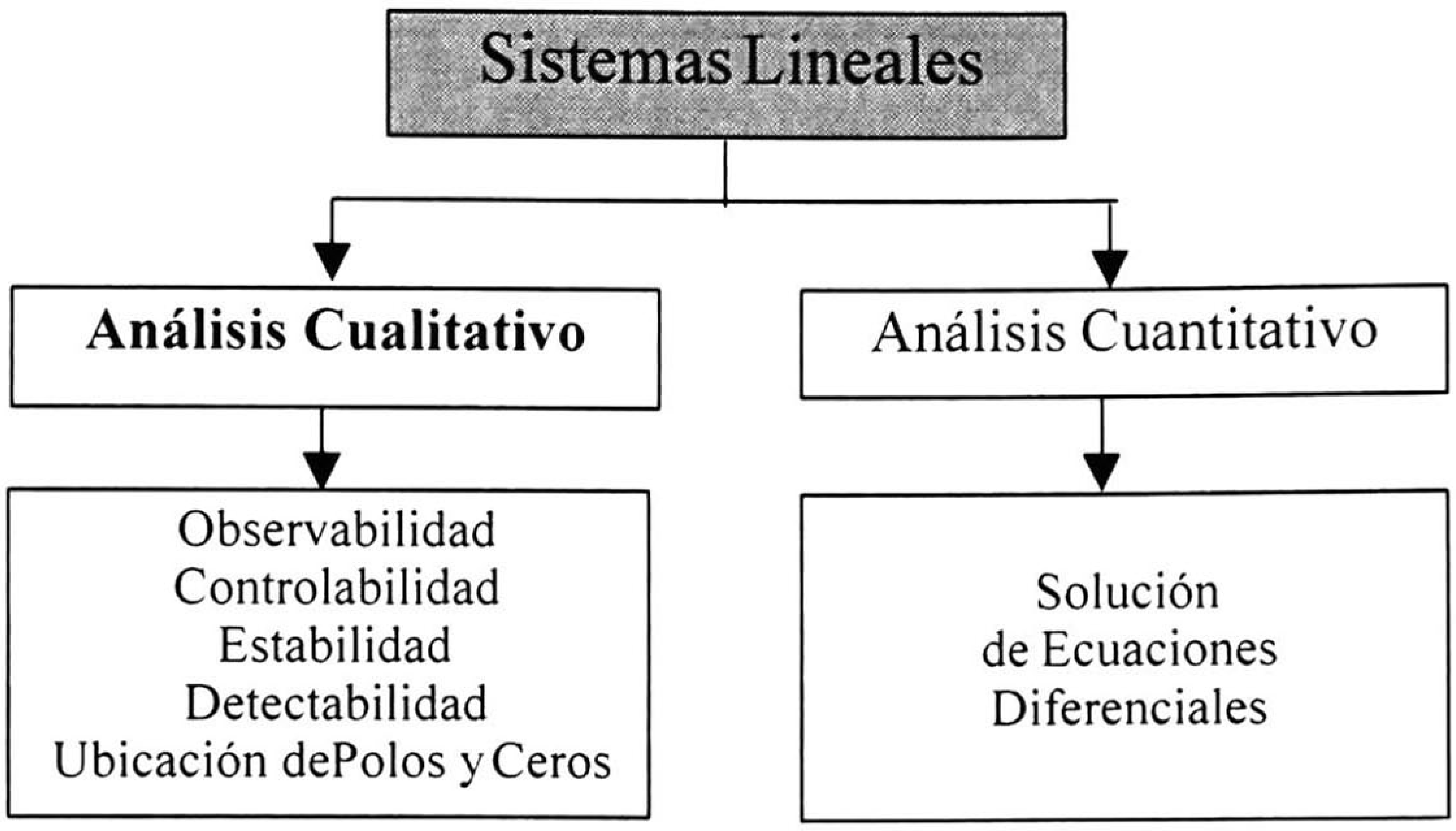

Figura 2.1: Tipos de análisis para sistemas lineales

5.- Se decide, en relación a las variables medidas y las variables manipuladas, que sensores y actuadores serán usados y donde se colocarán.

6.- Se selecciona la configuración del control.

7.– Se decide sobre un tipo de controlador a usar.

8.- Se decide sobre especificaciones de desempeño, basadas sobre las especificaciones totales de los objetivos de control.

9.- Se diseña el controlador.

10.- Se analizan los resultados del sistema controlado para ver si satisface las especificaciones. Si no se satisfacen es necesario modificar las especificaciones o el tipo de controlador.

11.- Se simulan los resultados del sistema controlado, utilizando una computadora o una planta piloto.

12.- Se repite desde el paso 2, si es necesario.

13.- Se elije el "hardware" y "software" y el controlador implementado.

14.- Se realiza un chequeo del sistema y se sintoniza el controlador en línea, si es necesario.

El proceso de análisis de los sistemas de control se enfoca comúnmente en los pasos 9, 10 y 11 del procedimiento antes mencionado; es decir, en métodos para diseño de controladores y análisis de sistemas de control.

2.1.2 Análisis de sistemas de control

Dentro de los sistemas de control lineal existen dos tipos de análisis que se llevan a cabo: el análisis cuantitativo y el análisis cualitativo. El primero consiste en resolver explícitamente un sistema de ecuaciones diferenciales lineales y obtener mediante esta forma, la respuesta dinámica de dicho sistema. El segundo consiste en estudiar propiedades de las ecuaciones diferenciales y algebráicas del sistema, como controlabilidad, observabilidad, estabilidad, etc. [**24**], Figura 2.1.

Para efectuar estos tipos de análisis se dispone de herramientas de análisis y de manipulación de datos, (Maple [**23**], Mathematica [**31**], Matlab [**18**]), así como de simulación, (VisSim [**28**], Simnon [**5**], Simulink).

2.1.3 Definiciones

Antes de seguir adelante se darán algunas definiciones de conceptos que se manejaran más adelante. Primeramente se definirá, para sistemas MIMO, la matriz de transferencia que en el caso de sistemas SISO se le conoce como función de transferencia.

Definición 1 *La matriz de transferencia de un sistema se define como la relación entre la transformada de Laplace de las variables de salida y la transformadas de Laplace de las variables de entrada, suponiendo que todas las condiciones iniciales se hacen iguales a cero. La matriz de transferencia de un sistema representa la relación que describe la dinámica del sistema. La matriz de transferencia es una descripción entradas-salidas del comportamiento de un sistema y por lo tanto la matriz de transferencia no incluye ninguna información concerniente a la estructura interna del sistema y a su comportamiento [**21**].*

Definición 2 *El estado de un sistema dinámico es el conjunto más pequeño de variables (denominadas variables de estado) tales que el conocimiento de esas variables en $t = t_0$, conjuntamente con el conocimiento de la entrada para $t \geqslant t_0$, determinan completamente el comportamiento del sistema en cualquier tiempo $t \geqslant t_0$ [**21**].*

Definición 3 *Un vector de estado es aquel que posee n componentes o variables de estado y en donde se tiene que esas n variables, pueden describir completamente el comportamiento de un sistema dado [**21**].*

Definición 4 *Ecuaciones del espacio de estado. En las ecuaciones del espacio de estado se manejan variables de salida "$y(t)$", variables de entrada "$u(t)$" y variables de estado "$x(t)$"*

Las ecuaciones del espacio de estado de sistemas lineales invariantes en el tiempo están dadas por

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

*donde $A \in R^{n\times n}$ se denomina matriz de estado, $B \in R^{n\times m}$ matriz de entrada, $C \in R^{p\times n}$ matriz de salida y $D \in R^{p\times m}$ matriz de transmisión directa [**21**].*

2.2 Problemas en el análisis de sistemas de control

En esta sección se expondrán algunos ejemplos donde, se pretende hacer notar que existen procedimientos, dentro de los métodos de análisis de un sistema de control MIMO, no integrados como funciones dentro del "software" comercial. En vista de ésto y para agilizar el proceso de los cálculos matemáticos, en esta tesis se ha desarrollado una serie de herramientas computacionales. A continuación se enlistan algunos problemas de control donde se ha detectado la necesidad de crear **herramientas computacionales** para facilitar el análisis.

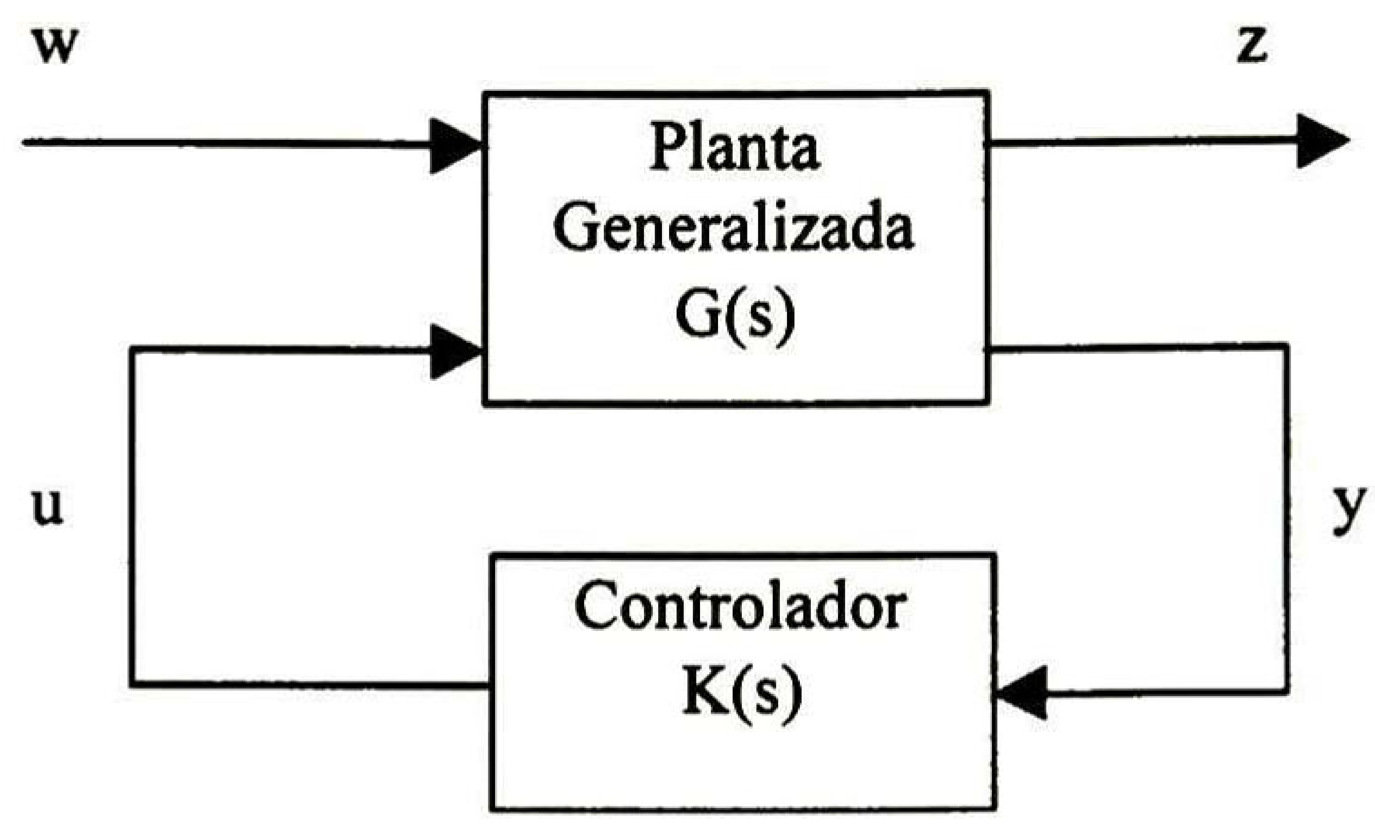

Figura 2.2: Esquema del problema estándar

2.2.1 Problema estándar

Muchos problemas de control pueden llevarse al esquema denominado esquema del problema estándar[2] [7], Figura 2.2, donde la planta generalizada contiene la planta a controlar y los modelos generadores de consignas, ruidos y perturbaciones. En general la planta generalizada en el dominio de la frecuencia está representada por una matriz de transferencia $G(s)$:

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \tag{2.1}$$

de la Figura 2.2, se tiene que

$$\begin{aligned} \begin{bmatrix} z(s) \\ y(s) \end{bmatrix} &= \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} w(s) \\ u(s) \end{bmatrix} \\ u(s) &= K(s)y(s) \end{aligned} \tag{2.2}$$

Por otro lado, en ciertos algoritmos para el diseño de controladores (por ejemplo: H_2, H_∞) [7] [4], es necesario obtener la realización en el espacio de estado la planta generalizada La pregunta, de rigor, es:

¿Cómo pasar una matriz de transferencia (en este caso la matriz de la planta Generalizada) al espacio de estado con herramientas computacionales?

Actualmente el control toolbox puede pasar fácilmente una función de transferencia a una realización en variables de estado, sin embargo, para realizar una operación similar con una matriz de transferencia no es tan sencillo; actualmente en el Control Toolbox de Matlab, se obtiene la representación en variables de estado, transformando cada uno de los elementos de la matriz de transferencia a su representación en variables de estado y se forma un sistema de ecuaciones de estado por bloques, con la posible desventaja de que las realizaciones así obtenidas generalmente no son mínimas. Lo anterior, sugiere la necesidad de programar algoritmos que permitan de pasar de forma más directa una matriz de transferencia a una realización en variables de estado mínima.

[2] El problema estándar también es conocido como "standard problem" o como problema normalizado.

Propiedad de controlabilidad y observabilidad También, en muchas ocasiones se hace necesario estudiar propiedades de los sistemas de ecuaciones en el espacio de estado, por ejemplo: una propiedad típica, estudiada en teoría de sistemas, es la controlabilidad y la observabilidad de un sistema. Esto sugiere la necesidad de tener **herramientas computacionales** que permitan obtener realizaciones en el espacio de estado donde la observabilidad y controlabilidad sean fácilmente detectadas.

2.2.2 Problema para encontrar la factorización espectral

Dentro de la teoría de control lineal con criterios de optimización de H_∞ se presenta, por ejemplo, un método para encontrar la factorización espectral de una matriz de transferencia $G(s)$ [**7**] tal que

$$G(s) = G_-^-(s)G_-(s)$$

$G(s)$ debe satisfacer que no tenga ni polos ni ceros sobre el eje imaginario, que tenga simetría de polos y ceros con respecto al eje imaginario y finalmente que sea bipropia ($G(s)$ evaluada en $s = \infty$ sea una constante positiva). La pregunta es:

¿Cómo obtener los valores de los polos y ceros de un sistema MIMO?

No se tienen actualmente **herramientas computacionales**, dentro del "software" comercial, para responder al problema planteado.

2.3 Herramientas computacionales

Los avances de "hardware" y "software", de programación y manipulación de datos, permiten elaborar, concretamente, herramientas para la investigación en áreas tan variadas como la Ingeniería Eléctrica, las Ciencias Médicas, la Geología,, la Astronomía, las Ciencias Sociales, etc. Estas herramientas son usadas diariamente por científicos, ingenieros, analistas, estudiantes, entre otros, alrededor del mundo. Es obvio suponer que en el terreno de la automatización y control automático se tenga un desarrollo similar de herramientas de análisis y síntesis, para sistemas de control. Lo anterior hace evidente que muchos de los desarrollos matemáticos involucrados, en esté análisis y síntesis de sistemas de control, se realicen en forma relativamente fácil.

Ahora, la pregunta que surge es: ¿En que lenguaje deben de programarse las herramientas computacionales, para resolver los problemas presentados en la sección anterior?. Como la información que se maneja en estas técnicas de análisis es algebráica, sugiere efectuar la programación en una **plataforma de computación simbólica.**

2.3.1 Programas de computación simbólica

Entre los más extensamente usados dentro de la teoría de control se encuentran Maple [**23**], Mathematica [**31**], Symbolic Math Toolbox de Matlab [**18**] [**19**]. Estos, en contraste con programas matemáticos convencionales, no requieren valores numéricos para todas las variables. Se pueden usar las capacidades simbólicas para obtener soluciones analíticas exactas a muchos problemas matemáticos incluyendo integrales, sistemas de ecuaciones, ecuaciones diferenciales y problemas del álgebra lineal. Complementan las operaciones simbólicas un largo conjunto de rutinas gráficas, para visualizar información matemática complicada, y un completo y comprensivo lenguaje de programación, para desarrollar funciones y aplicaciones.

En el aspecto educativo se observa que los profesores pueden hacer uso del "software" de computación simbólica para introducir conceptos teóricos en la formación de los alumnos, es decir que los estudiantes pueden concentrarse sobre conceptos importantes y dejar de lado la tediosa manipulación algebráica.

2.3.2 ¿Por qué elegir Matlab como plataforma de programación?

Los programas de computación simbólica como Maple, Mathematica y Symbolic Math Toolbox de Matlab, poseen una serie de librerías, integradas por funciones, las cuales ayudan en la solución de problemas generales, como obtener la inversa de una matriz, y problemas específicos, como obtener diagramas de bode. En el caso de control automático se hace uso de estas herramientas para efectuar análisis y síntesis de sistemas de control.

Matlab es un ambiente de computación numérica el cual posee una serie de herramientas integrables mejor conocidas como toolboxes. Entre las herramientas de Matlab se encuentra el Symbolic Math Toolbox, el cual habilita a Matlab para trabajar expresiones con símbolos, es decir que pasa de ser un programa de computación numérica a ser un programa de computación simbólica. Una vez habilitado Matlab como "software" de computación simbólica puede accesar el núcleo del Maple; es decir que muchas de las funciones de Maple pueden ser llamadas durante el desarrollo de un problema matemático en el entorno de Matlab.

Matlab tiene una ventaja sobre Maple y ésta es que Matlab posee herramientas para el estudio de problemas concernientes a control automático; por ejemplo: Robust Control Toolbox [**3**], Control Systems Toolbox [**14**], Optimization Toolbox, Nonlinear Control Design Blockset, Neural Network Toolbox, etc. En Maple no es posible obtener una gráfica de Bode o un diagrama de Nyquist, entre otras cosas. Matlab es una plataforma en la que investigadores como el Dr. Kwakernaak, en Holanda, desarrollan funciones de análisis para control automático.

Mathematica, si bien es cierto que tiene herramientas para análisis de sistemas de control, no posee aún la diversidad y abundancia de herramientas que ofrece Matlab.

Matlab

Matlab puede considerarse como un lenguaje de programación. Una característica de los números en Matlab es que no hay distinción entre reales, complejos, enteros, de precisión sencilla y de doble precisión. En Matlab, todos estos números están conectados continuamente. Esto significa que en Matlab cualquier variable puede contener números de cualquier tipo sin una declaración especial durante la programación, con lo cual esta última se hace más rápida y productiva. En Fortran se requiere una subrutina distinta para cada variable sencilla o doble, real o compleja, o entera, mientras que en Matlab no hay necesidad de separarlas.

La biblioteca matemática de Matlab facilita los análisis matemáticos. Además, el usuario puede crear rutinas matemáticas adicionales gracias a la continuidad entre las variables reales y complejas. Entre las numerosas funciones matemáticas, los toolboxes desempeñan un papel crucial; de hecho, es precisamente uno de ellos Symbolic Math Toolbox que habilita a Matlab a trabajar como un programa de **computación simbólica**.

2.4 Computación simbólica

La **computación simbólica**, también llamada **álgebra computacional**, sustituye al lápiz y papel por el teclado y la pantalla en los cálculos matemáticos algebráicos. Los programas computacionales interactivos, los cuales son llamados sistemas de álgebra computacional, permiten a los usuarios realizar los cálculos no solamente con números, sino también con símbolos, fórmulas, ecuaciones, etc. Muchos de los cálculos matemáticos, tales como integración, diferenciación, expansión de funciones en series, inversión de matrices con literales, pueden ser llevadas a cabo con suma rapidez, enfatizando en la exactitud del resultado, y sin mucho esfuerzo humano [**10**].

Los sistemas de computación simbólica se vuelven cada vez más importantes en la educación y en la investigación pura y aplicada. En el caso de control automático, la computación simbólica, favorece que muchos algoritmos de análisis y síntesis de sistemas, que eran desarrollados únicamente de forma analítica (lápiz y papel), puedan ser implementados en computadora con el consiguiente efecto de ahorro de tiempo y reducción de esfuerzo humano.

2.4.1 ¿Qué es la computación simbólica?

Históricamente el verbo "Calcular" ha sido utilizado en el sentido de "cálculo con números" La computación numérica involucra que las operaciones aritméticas son llevadas a cabo sobre números y sólo por números. Esto causa que en la mayoría de los casos los cálculos numéricos no sean exactos, por la particularidad intrínseca del manejo de los números de punto flotante en una computadora (errores de redondeo).

Pero la computación tiene otra componente importante que el llamada computación simbólica (álgebra computacional). Esta puede definirse como el cálculo con símbolos que representan objetos matemáticos. El adjetivo simbólico enfatiza que en muchos casos la meta última de la solución del problema matemático es expresar la respuesta en una fórmula cerrada o en una aproximación simbólica. Por algebráico se entiende que los cálculos son llevados a cabo con exactitud, de acuerdo con las reglas del álgebra, en vez de utilizar aritmética de punto flotante, que solo proporciona resultados aproximados.

Los últimos 25 años han sido de un gran progreso en la investigación teórica de algoritmos algebráicos. Esto ha dado al nacimiento de una nueva disciplina comúnmente referida como álgebra computacional. A la par, ha habido un desarrollo de herramientas conocidas como sistemas de álgebra computacional, Mathematica, Maple, Symbolic Math Toolbox [**9**].

2.4.2 Algunas propiedades de los sistemas de computación simbólica

Los sistemas de álgebra computacional son programas interactivos que, a diferencia con los programas computacionales numéricos, permiten cálculos matemáticos con símbolos.

Un segundo aspecto de los sistemas de álgebra computacional es el énfasis que hacen en la exactitud aritmética. Así, es posible obtener un resultado exacto, como $\frac{1}{4}\pi$, en vez de su valor aproximado, que es 0.785398; o, en su caso, obtener un valor numérico aproximado con una precisión definida por el usuario.

Los sistemas de álgebra computacional contienen una cantidad sustancial de conocimiento matemático, lo cual los hace buenos asistentes matemáticos. Con las herramientas de cálculo disponibles, es fácil explotar las funciones matemáticas. Por lo que, en ocasiones, se les conoce como sistemas matemáticos expertos.

En añadidura a sus funciones de calculadores simbólicos y algebráicos, la mayoría de los sistemas de álgebra computacional pueden ser usados como lenguajes de programación para construir nuevos algoritmos matemáticos [**20**].

2.4.3 Ventajas de la computación simbólica

La meta última del álgebra computacional es automatizar tanto como sea posible el proceso de resolución del problema matemático en su forma analítica. Esto hace a las herramientas útiles a la investigación y enseñanza.

La principal ventaja de un sistema de álgebra computacional es su capacidad para ejecutar grandes cálculos algebráicos que de otra forma, ejecutándolos a lápiz y papel, llevarían una gran cantidad de tiempo y esfuerzo.

Puesto que la computación simbólica y algebráica proceden al cálculo numérico, es conveniente que un sistema de álgebra computacional provea una buena interface entre estos dos tipos de computación. P. ej., Maple puede generar expresiones en Fortran y en C de expresiones de Maple.

Usando un sistema de álgebra computacional, es posible centrarse en al análisis matemático del problema, dejando a la computadora los detalles del cálculo rutinario. Los sistemas de álgebra computacional invitan a hacer "experimentos matemáticos" Ello hace fácil probar conjetura matemáticas y proponer nueva suposiciones basadas en los cálculos realizados [**9**].

2.4.4 Limitaciones de la computación simbólica

Los sistemas de álgebra computacional frecuentemente demandan mucho espacio de memoria y tiempo de ejecución en las computadoras. El precio que se paga por la exactitud aritmética tiene una relación que, generalmente, es exponencial al incremento en tamaño de las expresiones.

Un segundo problema en el uso de sistemas de álgebra computacional es de orden psicológico: ¿cuantas líneas en pantalla es posible retener? Y cuando es expresado un resultado en una larga expresión, ¿ puede el resultado simplificarse?. Además, se debe tener un conocimiento profundo del problema a solucionar, para así utilizar convenientemente los recursos del sistema y no llegar a obtener resultados inesperados o simplemente no obtener resultados por un mal manejo de los recursos (memoria y tiempo de ejecución de la computadora) [**9**].

Capítulo 3

Teoría para el análisis de sistemas MIMO

En este capítulo se presenta la teoría formal de los métodos para el análisis de sistemas de control multivariables (MIMO) [**12**]. Se presentan, también, los algoritmos correspondientes a los procedimientos requeridos en este análisis. Para un rápido y sencillo entendimiento de los algoritmos, éstos son descritos en seudo-código [**8**] [**29**] [**11**].

Puesto que las demostraciones no se presentan, si se desean más detalles, se recomienda revisar la bibliografía dada como referencia.

3.1 Conceptos para sistemas lineales MIMO

Definición 5 *Mínimo común múltiplo de dos o más expresiones algebraicas es la expresión algebraica de menor coeficiente numérico y de menor grado, que es divisible exactamente por cada una de las expresiones dadas [1].*

Definición 6 *Máximo común divisor de dos o más expresiones algebraicas es la expresión algebraica de mayor coeficiente numérico y de mayor grado, que está contenida exactamente en cada una de ellas [1].*

Definición 7 *Un polinomio mónico es aquel en donde el coeficiente líder es igual a uno.*

Definición 8 *El rango de una matriz es el número de columnas (filas) linealmente independientes de la matriz.*

Definición 9 *Rango pleno por filas (columnas). Se dice que una matriz tiene rango pleno por filas (columnas) si el rango de la matriz es igual al número de filas (columnas) de la matriz.*

Lema 10 *Una matriz cuadrada es invertible si, y sólo si, la matriz es de rango pleno.*

Definición 11 *Término es una expresión algebraica que consta de un solo símbolo o de varios símbolos no separados entre sí por el signo $+$ o $-$ [1].*

Definición 12 *Polinomio es una expresión algebraica que consta de más de un solo término [1].*

Definición 13 *Una función es polinomial si tiene la forma $g(s) = a(s)$, donde $a(s)$ es un polinomio.*

Definición 14 *Una función es racional,en las variables, si tiene la forma $h(s) = \frac{b(s)}{a(s)}$, donde $a(s)$ y $b(s)$ son polinomios.*

Definición 15 *Una matriz es llamada polinomial si todos sus elementos son funciones polinomiales.*

Definición 16 *Una matriz es llamada racional si todos sus elementos son funciones racionales.*

ISPOLNAL.M

A continuación se describe el algoritmo, ISPOLNAL, cuya función es determinar si una matriz es polinomial.

Algoritmo

```
DADA UNA MATRIZ X(S) SE OBTIENE SU DIMENSIÓN
[renx,colx]←—dimensión_de X;
INICIO DE PROCESO DE IDENTIFICACIÓN Y SOLUCIÓN
for i=1:renx
   for j=1:colx
EVALÚA CADA CELDA DE X(S)
      if deg(denominador_de x_ij) > 0
         no_es_una_matriz_polinomial; termina;
      end
   end
end
SE CONCLUYE QUE
es_una_matriz_polinomial;
```

■

Sea la ecuación dinámica

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.1}$$

donde $A \in \mathbb{R}^{n \times n}$; $B \in \mathbb{R}^{n \times m}$; $C \in \mathbb{R}^{p \times n}$

Definición 17 *Una realización $\{A, B, C\}$ es llamada observable si se puede determinar el estado $x(t)$ de la ecuación dinámica (Ec.3.1), conociendo $\{A, B, C\}$ y $\{y(t), u(t), t \geq 0\}$*

La matriz

$$\mathcal{O}^T = \begin{bmatrix} C^T & A^T C^T & \dots & (A^T)^{n-1} C^T \end{bmatrix} \tag{3.2}$$

es la matriz de observabilidad, comúnmente referenciada como $\mathcal{O}(C, A)$ o simplemente $\mathcal{O}$.

Lema 18 *Una realización $\{A, B, C\}$ es observable si, y sólo si, la matriz de observabilidad $\mathcal{O}(C, A) \in \mathbb{R}^{np \times n}$ tiene rango n.*

Definición 19 *Una realización $\{A, B, C\}$ sera llamada controlable si para cualquier tiempo t_0, cada estado inicial $x(t_0)$,de (Ec.3.1), puede transferirse a cualquier estado final $x(t_f)$ en un tiempo finito $t_f > t_0$ por medio de un vector de entrada $u(t)$ $t \in [t_0, t_f]$.*

La matriz

$$\mathcal{C} = \begin{bmatrix} B & AB & \dots & A^{n-1} B \end{bmatrix} \tag{3.3}$$

es la matriz de controlabilidad, comúnmente referenciada como $\mathcal{C}(A, B)$ o simplemente $\mathcal{C}$.

Lema 20 *Una realización $\{A, B, C\}$ es controlable si, y sólo si, la matriz de controlabilidad $\mathcal{C}(A, B) \in \mathbb{R}^{n \times nm}$ tiene rango n.*

Definición 21 *Dos realizaciones $\{A, B, C\}$ y $\{\bar{A}, \bar{B}, \bar{C}\}$son similares si existe una matriz invertible T tal que*

$$\begin{aligned} \bar{A} &= T^{-1} A T \\ \bar{B} &= T^{-1} B \\ \bar{C} &= C T \end{aligned} \tag{3.4}$$

Se puede ver que los estados de las dos realizaciones están relacionados como

$$x(t) = T \bar{x}(t) \tag{3.5}$$

pero la matriz de transferencia no es afectada, esto es, sí

$$\begin{aligned} G(s) &= C(sI - A)^{-1} B \\ \overline{G}(s) &= \bar{C}(sI - \bar{A})^{-1} \bar{B} \end{aligned} \tag{3.6}$$

se tiene que

$$G(s) = \overline{G}(s) \tag{3.7}$$

La relación entre la matriz de observabilidad y controlabilidad del sistema $\{A, B, C\}$ y $\{\bar{A}, \bar{B}, \bar{C}\}$ está dada por

$$\bar{\mathcal{O}} = \mathcal{O} T \tag{3.8}$$

$$\bar{\mathcal{C}} = T^{-1} \mathcal{C} \tag{3.9}$$

donde

$$\begin{array}{l} \bar{\mathcal{O}} \text{ es la matriz de observabilidad del par } \{\bar{A}, \bar{C}\} \\ \bar{\mathcal{C}} \text{ es la matriz de controlabilidad del par } \{\bar{A}, \bar{B}\} \end{array}$$

Puesto que la función de transferencia no es afectada (Ec.3.7) se tiene que los rangos de las matrices de controlabilidad y observabilidad son preservados bajo transformaciones de similaridad.

Lema 22 *Estados controlables y no controlables. Suponga una realización* $\{A, B, C\}$ *donde*

$$\mathcal{C}(A, B) \text{ tiene rango } r < n$$

Entonces es posible encontrar una transformación de similaridad T *tal que*

$$\bar{A} = T^{-1}AT = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{\bar{c}} \end{bmatrix}, \bar{B} = T^{-1}B = \begin{bmatrix} \bar{B}_c \\ 0 \end{bmatrix}, \bar{C}^T = (CT)^T = \begin{bmatrix} \bar{C}_c^T \\ \bar{C}_{\bar{c}}^T \end{bmatrix}$$

con las siguientes propiedades
1.- $\{\bar{A}_c, \bar{B}_c\}$ *es controlable*
2.- Los estados de $\{\bar{A}, \bar{B}, \bar{C}\}$ *pueden ser separados en estados controlables y no controlables.*
3.- $\bar{C}_c(sI - \bar{A}_c)^{-1}\bar{B}_c = C(sI - A)^{-1}B$

Lema 23 *Estados observables y no observables. Suponga una realización* $\{A, B, C\}$ *donde*

$$\mathcal{O}(C, A) \text{ tiene rango } r < n$$

Entonces es posible encontrar una transformación de similaridad T *tal que*

$$\bar{A} = T^{-1}AT = \begin{bmatrix} \bar{A}_o & 0 \\ \bar{A}_{21} & \bar{A}_{\bar{o}} \end{bmatrix}, \bar{B} = T^{-1}B = \begin{bmatrix} \bar{B}_o \\ \bar{B}_{\bar{o}} \end{bmatrix}, \bar{C}^T = (CT)^T = \begin{bmatrix} \bar{C}_o^T \\ 0 \end{bmatrix}$$

con las siguientes propiedades
1.- $\{\bar{C}_o, \bar{A}_o\}$ *es observable*
2.- Los estados de $\{\bar{A}, \bar{B}, \bar{C}\}$ *pueden se separados en estados observables y no observables.*
3.- $\bar{C}_o(sI - \bar{A}_o)^{-1}\bar{B}_o = C(sI - A)^{-1}B$

Definición 24 *Realización mínima es una que tiene la dimensión mas pequeña de la matriz* A*, para todas las tripletas* $\{A, B, C\}$ *que satisfagan*

$$C(sI - A)^{-1}B = H(s)$$

donde $H(s)$ *es una función de transferencia dada.*

Lema 25 *Una realización* $\{A, B, C\}$ *es mínima si, y sólo si, es controlable y observable.*

Lema 26 *Si* $\{A_i, B_i, C_i, i = 1, 2\}$ *son dos realizaciones mínimas de una función de transferencia, entonces existe una única matriz invertible* T *tal que*

$$\begin{aligned} A_2 &= T^{-1}A_1T \\ B_2 &= T^{-1}B_1 \\ C_2 &= C_1T \end{aligned} \tag{3.10}$$

3.2 Descripción en fracción matricial (MFD)

Definición 27 *Se dice que las matrices, $N(s)$, $D(s)$, forman una descripción en fracción matricial derecha, RMFD, de una matriz $H(s)$, si*
i) $D(s)$ es invertible
ii) $H(s) = N(s)D^{-1}(s)$
donde $N(s) \in \mathbb{R}^{p\times m}[s]$ y $D(s) \in \mathbb{R}^{m\times m}[s]$

A continuación se describe un caso particular para obtener una descripción en fracción matricial derecha, RMFD. Dada una matriz, $H(s)$, es posible descomponerla en

$$H(s) = N_R(s)D_R^{-1}(s) \tag{3.11}$$

donde

$$D_R(s) = d(s)I_m \tag{3.12}$$
$$N_R(s) = N(s) \tag{3.13}$$

y

$$H(s) = \frac{N(s)}{d(s)} \tag{3.14}$$

donde $d(s)$ es el mínimo común múltiplo de los denominadores de los elementos h_{ij} de $H(s)$.

Lema 28 *Dada una MFD derecha de $H(s)$, siempre es posible obtener una realización en espacio de estados controlable $\{A, B, C\}$ de orden*

$$n = deg\_det D_R(s) \triangleq \textit{el grado de la MFD}$$

Nota 29 *Frecuentemente se omitirá el sub-índice $_R$, restaurándolo solamente cuando sea necesario por claridad de la exposición.*

Definición 30 *Se dice que las matrices, $N_L(s)$, $D_L(s)$, forman una descripción en fracción matricial izquierda, LMFD, de una matriz $H(s)$, si*
i) $D_L(s)$ es invertible
ii) $H(s) = D_L^{-1}(s)N_L(s)$
donde $N_L(s) \in \mathbb{R}^{p\times m}[s]$ y $D_L(s) \in \mathbb{R}^{p\times p}[s]$

A continuación se describe un caso particular para obtener una descripción en fracción de matrices izquierda, LMFD. Dada una matriz, $H(s)$, es posible descomponerla en

$$H(s) = D_L^{-1}(s)N_L(s) \tag{3.15}$$

donde

$$D_L(s) = d(s)I_p \tag{3.16}$$
$$N_L(s) = N(s) \tag{3.17}$$

y

$$H(s) = \frac{N(s)}{d(s)} \tag{3.18}$$

donde $d(s)$ es el mínimo común múltiplo de los denominadores de los elementos h_{ij} de $H(s)$.

Lema 31 *Dada una MFD izquierda de $H(s)$, siempre es posible obtener una realización en espacio de estados observable $\{A, B, C\}$ de orden*

$$n = deg\_det D_L(s) \triangleq \text{el grado de la MFD}$$

3.2.1 Máximo común divisor y coprimicidad

Si se tiene

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) & (3.19)\\ &= \bar{N}(s)\bar{D}^{-1}(s) & (3.20)\end{aligned}$$

donde

$$\begin{aligned} D(s) &= \bar{D}(s)W(s) & (3.21)\\ N(s) &= \bar{N}(s)W(s) & (3.22)\end{aligned}$$

se puede llamar a $W(s) \in \mathbb{R}^{p\times m}[s]$ un divisor derecho de $N(s)$ y $D(s)$, por lo tanto el grado de la MFD, (esto es el grado del determinante de la matriz denominador), puede ser reducido quitando los divisores derechos comunes de las matrices numerador y denominador. Luego entonces, es posible obtener una MFD de grado mínimo por la extracción de un máximo común divisor derecho (gcrd[1]) de $N(s)$ y $D(s)$.

Definición 32 *Una matriz $W(s)$ es llamada unimodular si, y sólo si, $W(s)$ tiene la siguiente propiedad*

$$det\ W(s) = \text{una constante diferente de cero} \tag{3.23}$$

Lema 33 *Es posible determinar un gcrd de $N(s)$ y $D(s)$ si, y sólo si,*

$$deg\_det D(s) = deg\_det \bar{D}(s)$$

para todos los divisores derechos no singulares, $W(s)$, de $N(s)$ y $D(s)$ se tiene que la matriz $W(s)$ es unimodular.

Lema 34 *Una matriz polinomial, $W(s)$, es unimodular si, y sólo si, tiene inversa, $W^{-1}(s)$, también polinomial.*

Definición 35 *Una MFD $H(s) = N(s)D^{-1}(s)$ será llamada irreducible si el gcrd de $N(s)$ y $D(s)$ es una matriz unimodular.*

Lema 36 *Una MFD irreducible no es única. Dada una factorización derecha, $N(s)D^{-1}(s)$, irreducible, entonces, es posible obtener otra factorización derecha irreducible, esto es*

$$N(s)W(s)[D(s)W(s)]^{-1}$$

para una $W(s)$ unimodular. Como $W(s)$ no es única entonces una MFD irreducible no es única tampoco y por lo tanto los gcrds no son únicos.

[1] gcrd es un abreviación del ingles greatest common right divisor. Se adopta astá abreviatura debido a su extenso uso dentro del lenguaje que maneja la literatura de control.

GCRDMFD.M/GCLDMFD.M

A continuación se presenta una breve introducción donde se describe como obtener el máximo común divisor derecho de una MFD derecha. Luego se describe el algoritmo, GCRDMFD, cuya función es obtener el máximo común divisor derecho de las matrices de una MFD derecha y además las matrices correspondientes a una MFD derecha irreducible.

Un máximo común divisor derecho (gcrd), de dos matrices $\{N(s), D(s)\}$, es una matriz $R(s) \in \mathbb{R}^{m\times m}[s]$ con las siguientes propiedades

1.- $R(s)$ es un divisor derecho de $N(s)$ y $D(s)$; esto es, existen matrices polinomiales $\bar{N}(s) \in \mathbb{R}^{p\times m}[s]$ y $\bar{D}(s) \in \mathbb{R}^{m\times m}[s]$ tal que

$$N(s) = \bar{N}(s)R(s) \tag{3.24}$$
$$D(s) = \bar{D}(s)R(s) \tag{3.25}$$

2.- Si $R_1(s) \in \mathbb{R}^{m\times m}[s]$ es cualquier otro divisor derecho de $N(s)$ y $D(s)$, entonces $R_1(s)$ es un divisor derecho de $R(s)$; esto es existe una matriz polinomial $W(s)$ tal que $R(s) = W(s)R_1(s)$.

Dada una MFD con $D(s) \in \mathbb{R}^{m\times m}[s]$ y $N(s) \in \mathbb{R}^{p\times m}[s]$, es posible determinar una matriz unimodular $U(s)$ tal que

$$\begin{bmatrix} U_{11}(s) & U_{12}(s) \\ U_{21}(s) & U_{22}(s) \end{bmatrix} \begin{bmatrix} D(s) \\ N(s) \end{bmatrix} = \begin{bmatrix} R(s) \\ 0 \end{bmatrix} \tag{3.26}$$

donde $R(s)$ será un gcrd de $D(s)$ y $N(s)$.

Una manera sencilla de obtener $R(s)$, (Ec.3.26), es a partir de la forma de hermite de la matriz expandida, $[D(s); N(s)]$. Una vez obtenido el gcrd, $R(s)$, se extrae de cada una de las matrices de la MFD, (Ec.3.21, Ec.3.22), para así obtener una nueva MFD la cual es de grado mínimo. Es posible que $R(s)$ sea una matriz unimodular, (Ec.3.23), lo cual significa que $N(s)$ y $D(s)$ son irreducibles.

Algoritmo

LA FORMA DE HERMITE PERMITE OBTENER R(S)
[R(s);0]⟵matriz_de_hermite_de_[D;N];
CON LO QUE SE OBTINEN LAS NUEVAS N(S) Y D(S)
D_irreducible⟵D*R^{-1};
N_irreducible⟵N*R^{-1};
■

Para desarrollar el algoritmo GCLDMFD.M, (cuya función es obtener el máximo común divisor izquierdo de las matrices de una MFD izquierda y además las matrices correspondientes a una MFD izquierda irreducible), se aplica un razonamiento análogo.

Definición 37 *Dos matrices polinomiales $N(s)$ y $D(s)$ con el mismo número de columnas son llamadas primas relativas derechas (o coprimas derechas) si, y sólo si, tienen divisores unimodulares derechos comúnes.*

Definición 38 *Una MFD $H(s) = N(s)D^{-1}(s)$ será llamada irreducible si $N(s)$ y $D(s)$ son coprimas derechas.*

ISRCOPRI.M/ISLCOPRI.M

A continuación se describe el algoritmo ISRCOPRI, el cual a partir de una MFD derecha, $H(s) = N(s)D^{-1}(s)$, determina si las matrices $N(s)$ y $D(s)$ son coprimas derechas. Estas son matrices coprimas derechas, si el máximo común divisor derecho de $N(s)$ y $D(s)$ es una matriz unimodular $W(s)$, (Ec.3.23).

Algoritmo

```
SE OBTIENE EL MÁXIMO COMÚN DIVISOR DERECHO
W←—gcrdmfd(N,D);
EVALUACIÓN DE W
if det(W)=constante_diferente_de_cero
   son_matrices_coprimas_derechas_N_D;
else
   no_son_matrices_coprimas_derechas_N_D;
end
```

■

Para desarrollar el algoritmo ISLCOPRI.M, (el cual a partir de una MFD izquierda determina si las matrices $N(s)$ y $D(s)$ son coprimas izquierdas), se aplica un razonamiento análogo.

3.2.2 Matrices reducidas por columnas y/o filas

Definición 39 *Una matriz de transferencia, $H(s)$, es llamada propia si*

$$\lim_{s\to\infty} H(s) < \infty$$

y estrictamente propia si

$$\lim_{s\to\infty} H(s) = 0$$

Lema 40 *Si $H(s)$ es una matriz de transferencia estrictamente propia (propia) tal que*

$$H(s) = N(s)D^{-1}(s) \tag{3.27}$$

entonces cada columna de $N(s)$ tiene grado estrictamente menor que (menor o igual que) la columna correspondiente de $D(s)$.

Definición 41 *Una matriz de coeficientes líderes por filas, $Q \in \mathbb{R}^{p\times m}$, de una matriz, $X(s) \in \mathbb{R}^{p\times m}[s]$, es aquella cuyos elementos, q_{ij}, contienen los coeficientes de los términos de más alta potencia de los polinomios de mayor grado de cada fila ($l_i, i = 1, ..., p$) de $X(s)$; por ejemplo, sea*

$$X(s) = \begin{bmatrix} s^3 & 1 & 5s^5 \\ s^3 & 3s^3 & 1 \end{bmatrix}$$

donde

$$l_1 = 5; \; l_2 = 3$$

entonces

$$Q = \begin{bmatrix} 0 & 0 & 5 \\ 1 & 3 & 0 \end{bmatrix}$$

Definición 42 *Una matriz de coeficientes líderes por columnas, $Q \in \mathbb{R}^{p\times m}$, de una matriz, $X(s) \in \mathbb{R}^{p\times m}[s]$, es aquella cuyos elementos, q_{ij}, contienen los coeficientes de los términos de más alta potencia de los polinomios de mayor grado de cada columna ($k_j, j = 1, ..., m$) de $X(s)$; por ejemplo, sea*

$$X(s) = \begin{bmatrix} s^3 & 1 & 5s^5 \\ s^3 & 3s^3 & 1 \end{bmatrix}$$

donde

$$k_1 = 3; \; k_2 = 3; \; k_3 = 5$$

entonces

$$Q = \begin{bmatrix} 1 & 0 & 5 \\ 1 & 3 & 0 \end{bmatrix}$$

Definición 43 *Matriz reducida por filas (columnas). Dada una matriz polinomial $X(s)$, con rango pleno por filas (columnas), "p"("m"), será reducida por filas (columnas) si, y sólo si, el rango de la matriz de coeficientes líderes, Q, es igual a "p"("m").*

Lema 44 *Si $D(s)$ es reducida por columnas, entonces $H(s) = N(s)D^{-1}(s)$ es estrictamente propia (propia) si, y sólo si, cada columna de $N(s)$ tiene un grado menor que (menor que o igual a) el grado de la columna correspondiente de $D(s)$.*

Lema 45 *Una matriz invertible, $D(s) \in \mathbb{R}^{m\times m}[s]$, siempre es posible transformarla en reducida por columnas (filas) mediante operaciones fila o columna*

$$D_{reducida} = D(s)V(s) \tag{3.28}$$
$$D_{reducida} = U(s)D(s) \tag{3.29}$$

donde $\{U(s), V(s)\} \in \mathbb{R}^{m\times m}(s)$

ISROWRED.M/ISCOLRED.M

A continuación se describe el algoritmo ISROWRED, cuya función es determinar si una matriz es reducida por filas; se retorna además la matriz de coeficientes líderes, (Definición 41).

Algoritmo .

```
SE OBTIENE LA DIMENSIÓN DE D(S)
[rend,cold]←—dimensión_de_D;
INICIO DE PROCESO
for i=1:rend
SOBRE UNA FILA I SE LOCALIZA LA CELDA Dij EN LA CUAL SE ENCUENTRA EL POLINOMIO DE MAYOR GRADO.
  k←—1;
  for j=2:cold
    if grado_de dik < grado_de dij
      k←—j;
    end
  end
SE CONSTRUYE UNA MATRIZ DE COEFICIENTES LÍDERES
  qik ←—coeff(dik,deg(dik))
  for w=k+1:cold
    if deg(dik)=deg(diw)
      qiw ←—coeff(diw,deg(diw));
    end
  end
end
SE ANALIZA LA MATRIZ DE COEFICIENTES LÍDERES, Q
if rank(Q)≠rend
  no_es_reducida_por_filas;
else
  es_reducida_por_filas;
end
```

■

Para desarrollar el algoritmo ISCOLRED.M, (cuya función es determinar si una matriz es reducida por columnas y retornar además la matriz de coeficientes líderes), se aplica un razonamiento análogo.

Nota 46 *Nuestra discusión acerca de matrices reducidas por columnas, grados de las columnas de las matrices, etc. tienen obviamente una contraparte para la cual podemos llamar matrices reducidas por filas, grados de las filas de las matrices, etc. Se decide no hacer la repetición de éstos conceptos para no confundir al lector.*

MPROPSYM.M

A continuación se presenta el procedimiento bajo la cual opera el algoritmo MPROPSYM cuya función es descomponer una matriz racional, $H(s) \in \mathbb{R}^{p\times m}(s)$, en una matriz racional estrictamente propia, $H_p(s) \in \mathbb{R}^{p\times m}(s)$, más una matriz polinomial residual, $Dft(s) \in \mathbb{R}^{p\times m}[s]$

$$H(s) = Hp(s) + Dft(s) \tag{3.30}$$

Dadas la matrices $\{N(s), D(s)\}$ es posible encontrar matrices $Q(s)$ y $R(s)$ tal que

$$\begin{aligned} N(s) &= Q(s)D(s) + R(s) \\ \text{con} \quad & R(s)D^{-1}(s) \text{ estrictamente propia} \end{aligned} \tag{3.31}$$

Un camino para hacer ésto, es descomponer la matriz racional $H(s) = N(s)D^{-1}(s)$ como

$$H(s) = N(s)D^{-1}(s) \tag{3.32}$$
$$= Hp(s) + Dft(s) \tag{3.33}$$

donde $Dft(s)$ es una matriz polinomial y $Hp(s)$ es estrictamente propia. Teniendo en cuenta

$$H(s)D(s) = N(s) = Dft(s)D(s) + R(s) \tag{3.34}$$

donde

$$R(s) = Hp(s)D(s) \tag{3.35}$$

se concluye que $R(s)D^{-1}(s)$ es estrictamente propia.

Algoritmo

SE OBTIENE LA DIMENSIÓN DE H(S)
[renh,colh]⟵dimensión_de H;
INICIO DE PROCESO
for i=1:renh
    for j=1:colh
        [cociente,residuo]⟵división_de h_{ij}
        if cociente_es_diferente_de_cero
            $d_{ft[ij]}$ ⟵cociente;
            $h_{p[ij]}$ ⟵residuo_entre_el_denominador_de h_{ij};
        else
            $d_{ft[ij]}$ ⟵cero;
        end
    end
end
■

3.2.3 MFD irreducibles y realizaciones mínimas

Lema 47 *Cualquier realización en espacio de estado de una MFD, con orden igual al grado del determinante de la matriz denominador, es una realización mínima (equivalente, a controlable y observable) si y sólo si la MFD es irreducible.*

Lema 48 *Si existe una realización en espacio de estado controlable y observable para $N(s)$ $D^{-1}(s)$, con orden $n = deg\_detD(s)$, entonces todas las realizaciones de orden n serán también controlables y observables.*

Lema 49 *Una realización eb espacio de estado en forma controlador para $N(s)D^{-1}(s)$, de orden igual al deg_det$D(s)$, es también observable (y de aquí en adelante mínima) si y sólo si la MFD es irreducible.*

Lema 50 *Suponiendo que $\{N(s)D^{-1}(s), i = 1, 2\}$ son dos MFDs irreducibles. Entonces existe una matriz unimodular $U(s)$ tal que*

$$D_1(s) = D_2(s)U(s) \tag{3.36}$$
$$N_1(s) = N_2(s)U(s) \tag{3.37}$$

Lema 51 *Si $\{N(s), D(s)\}$ es una MFD de $H(s)$ y $\{\bar{N}(s), \bar{D}(s)\}$ es una MFD irreducible de $H(s)$, entonces existe una matriz polinomial $R(s)$, no necesariamente unimodular, tal que*

$$D(s) = \bar{D}(s)R(s) \tag{3.38}$$
$$N(s) = \bar{N}(s)R(s) \tag{3.39}$$

Lema 52 *El grado del determinante de una MFD derecha irreducible de $H(s)$ es igual al grado determinado de una MFD izquierda irreducible de $H(s)$.*

Lema 53 *Si $H(s) = N(s)D^{-1}(s)$ es propia entonces $\begin{bmatrix} D^T(s) & N^T(s) \end{bmatrix}^T$ es reducida por columnas si y sólo si $D(s)$ es reducida por columnas.*

Teorema 1 *Dada una matriz de transferencia, $G(s)$, es posible obtener una factorización izquierda, $D_L^{-1}(s)N_L(s)$, a partir de una factorización derecha, $N(s)D^{-1}(s)$, de la transpuesta de $G(s)$.*

Prueba: *Dada una matriz $G(s) \in \mathbb{R}^{p\times m}(s)$, se define*

$$H(s) = G^T(s) \tag{3.40}$$

por otro lado, se sabe que

$$H(s) = N_R(s)D_R^{-1}(s) \tag{3.41}$$
$$= D_L^{-1}(s)N_L(s) \tag{3.42}$$

y además

$$H(s) = \left(\left(D_R^{-1}(s)\right)^T \left(N_R(s)\right)^T\right)^T \tag{3.43}$$

entonces

$$(H(s))^T = \left(D_R^{-1}(s)\right)^T \left(N_R(s)\right)^T \tag{3.44}$$

ahora haciendo

$$\bar{H}(s) = (H(s))^T \tag{3.45}$$

se obtiene

$$\begin{aligned}\bar{H}(s) &= \left(D_R^{-1}(s)\right)^T (N_R(s))^T && (3.46)\\ &= D_L^{-1}(s)N_L(s) && (3.47)\end{aligned}$$

como

$$\begin{aligned}H(s) &= G^T(s) \Longrightarrow \bar{H}(s) = (H(s))^T && (3.48)\\ &\Longrightarrow \bar{H}(s) = \left(G^T(s)\right)^T && (3.49)\\ \bar{H}(s) &= G(s) \Longrightarrow G(s) = D_L^{-1}(s)N_L(s)\,.. && (3.50)\end{aligned}$$

MFD.M

A continuación se describe el algoritmo, MFD, cuya función es obtener una MFD derecha mínima, $H(s) = N(s)D^{-1}(s)$; para esto se obtiene una $D(s)$ la cual es construida colocando en su diagonal, d_{ii}, el máximo común divisor de la columna i de $H(s)$. Dado que $D(s)$ es invertible es fácil obtener $N(s)$.

$$N(s) = H(s)D(s) \tag{3.51}$$

Hecho lo anterior, se obtiene el gcrd de $N(s)$ y $D(s)$ para así llegar a una MFD mínima, (Ec.3.21, Ec.3.22).

Algoritmo

PARA OBTENER UNA MFD DERECHA, H(S)=N_R(S)D_R^{-1}(S)

SE OBTIENE LA DIMENSIÓN DE H(S)

[renh,colh]⟵dimensión_de H;

COLOCA EN LA DIAGONAL DE D_R(S) EL MÍNIMO COMÚN MULTIPLO DE LOS DENOMINADORES DE CADA UNO DE LOS ELEMENTOS DE LA COLUMNA CORRESPONDIENTE DE H(S).

```
for k=1:colh
  d_kk ⟵denominador_de h_1k;
  for i=2:renh
    d_kk ⟵mínimo_común_múltiplo_de d_kk y_el denominador_de h_ik;
  end
end
```

ES FÁCIL AHORA DEDUCIR N_R(S) PARTIENDO DE $H(s) = N_R(s)D_R^{-1}(s) \Longrightarrow N_R(s) = H(s)D_R(s)$

```
for j=1:colh
  for i=1:renh
    n_ij ⟵h_ij*d_jj;
  end
end
```

FINALMENTE SE EXTRAE EL MÁXIMO COMÚN DIVISOR DERECHO DE N_R(S) Y D_R(S) PARA QUE POR ENDE SEAN ESTAS COPRIMAS DERECHAS

[R,N,D]⟵gcrdmfd(N,D);

■

Para obtener una MFD izquierda, $H(s) = D_L^{-1}(s)N_L(s)$, se aplica Teorema 1.

3.2.4 MFD izquierda a partir de una MFD derecha y viceversa

Lema 54 *Dada una factorización izquierda (derecha) siempre es posible obtener una factorización derecha (izquierda) por medio de operaciones elementales de filas y columnas.*

Procedimiento 55 *Dada una MFD izquierda*

$$D_L^{-1}(s)N_L(s) \tag{3.52}$$

se construye la matriz

$$\begin{aligned} M &= \begin{bmatrix} E \\ F_{(d+n)} \end{bmatrix} && (3.53) \\ &= \begin{bmatrix} \begin{bmatrix} e_{11} & e_{12} \end{bmatrix} \\ \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \end{bmatrix} && (3.54) \\ &= \begin{bmatrix} \begin{bmatrix} D_L^{-1}(s) & N_L(s) \end{bmatrix} \\ \begin{bmatrix} I_d & 0 \\ 0 & I_n \end{bmatrix} \end{bmatrix} && (3.55) \end{aligned}$$

donde $F_{(d+n)}$, es una matriz identidad, con "d" igual a la dimensión de $D_L(s)$ y "n" igual al número de columnas de $N(s)$.

Haciendo operaciones sobre filas y columnas, se obtiene

$$M = \begin{bmatrix} \begin{bmatrix} I_d & 0 \end{bmatrix} \\ \begin{bmatrix} * & * \\ N_R(s) & D_R^{-1}(s) \end{bmatrix} \end{bmatrix} \tag{3.56}$$

donde finalmente se tiene una MFD derecha, $N_R(s)D_R^{-1}$ Cabe decir que $N_R(s)$ y $D_R(s)$ son coprimas derechas, pero, $D_R(s)$ no es reducida por columnas.

Nota 56 *Un procedimiento análogo, al anterior, se aplica para obtener una MFD izquierda a partir de una derecha.*

MFDIZ2DE.M/MFDDE2IZ.M

A continuación se describe el algoritmo, MFDIZ2DE, cuya función es obtener una MFD derecha, $N_R(s)D_R^{-1}(s)$, a partir de una MFD izquierda, $D_L^{-1}(s)N_L(s)$ según el procedimiento anteriormente descrito.

Algoritmo

```
INICIO
[renh,colh]←—dimensión_de N_L;
E←—[D_L,N_L];
MATRIZ IDENTIDAD PARA F
F←—Identidad_(renh+colh);
INICIO DE PROCESO
for m=1:renh % mueve por renglones
   k←—renh;
   while k < colh+renh % mueve por columnas diagonales
      k←—k+1;
      if E_mk diferente_de_cero
         q=cociente_de E_mm÷ E_mk;
         if q diferente_de_cero
PROCEDER A SIMPLIFICAR E(*,M)
            for i=m:renh
               E_im ←—E_im-q*E_ik;
            end
            for i=1:renh+colh
               F_im ←—F_im-q*F_ik;
            end
         end
         if E_mm==0
            E←—intercambia_las_columnas_m & k de_E;
            F←—intercambia_las_columnas_m & k de_F;
         end
         q←—cociente_de E_mk ÷ E_mm;
HACER CEROS EN LA COLUMNA (K), Y RENGLÓN (M) DE E(S)
         for i=m:renh
            E_ik ←—E_ik-q*E_im;
         end
         for i=1:renh+colh
            F_ik ←—F_ik-q*F_im;
         end
SWAM SI DIFERENTE DE CERO
         if E_mk_diferente_de_cero
            E←—intercambia_las_columnas_m & k de_E;
            F←—intercambia_las_columnas_m & k de_F;
            k←—k-1;
         end
      end
   end
end
SELECCIÓN DE DATOS PARA N
for i=1:renh
```

```
    for j=1:colh
      N_ij ←—F_i,j+renh;
    end
end
SELECCIÓN DE DATOS PARA D
for i=1:colh
    for j=1:colh
      D_ij ←—F_i+renh,j+renh;
    end
end
```

■

Para desarrollar el algoritmo MFDDE2IZ.M, (cuya función es obtener una MFD izquierda a partir de una MFD derecha), se aplica un razonamiento análogo.

3.3 Realizaciones en espacio de estados

A continuación se describen, brevemente, algunos procedimientos para obtener realizaciones en espacio de estados a partir de una MFD. Es pertinente señalar que si la MFD, trabajada en cada uno de los algoritmos, es mínima entonces la realización también lo será.

3.3.1 Forma Controlador

Es posible obtener una realización en forma controlador, $\{A_c, B_c, C_c\}$, a partir de una factorización derecha de una matriz de transferencia , $H(s) = N(s)D^{-1}(s)$. Es necesario que $D(s)$ sea reducida por columnas, de lo contrario se aplica (Ec.3.28[2]). La realización es de orden, n, igual al grado del determinante de $D(s)$

$$n = deg\_detD(s) = \sum_{1}^{m} k_i \tag{3.57}$$

donde

$$k_1 \leq k_2 \leq ... \leq k_m$$

$k_i, i = 1, ..., m$ es el grado de la i-ésima columna de $D(s)$.
Se tiene

$$H(s) = C_c(SI - A_c)^{-1}B_c \tag{3.58}$$
$$= N(s)D^{-1}(s) \tag{3.59}$$

donde $H(s)$ es una matriz estrictamente propia. La matriz $H(s)$ es de dimensión $p \times m$, esto es

p es el número de filas de $H(s)$

m es el número de columnas de $H(s)$

[2]Si se ha modificado, con ciertas operaciones, la matriz $D(s)$ para hacerla reducida por columnas, de igual manera se debe modificar $N(s)$.

La matriz $D(s)$ se descompone como

$$D(s) = D_{hc}S(s) + D_{lc}\Psi(s) \tag{3.60}$$

donde

D_{hc} es la matriz de coeficientes líderes de $D(s)$ (Definición 42).

D_{lc} es una matriz con los coeficientes de los términos de $D(s)$ no incluidos en D_{hc}.

$$\begin{aligned} S(s) &= diag\{s^{k_i}, i = 1, ...m\} && (3.61) \\ \Psi^T(s) &= block\_diag\{[s^{k_i-1}, ..., s, 1], i = 1, ..., m\} && (3.62) \end{aligned}$$

$N(s)$, de la (Ec.3.59), se factoriza como

$$N(s) = N_{lc}\Psi(s) \tag{3.63}$$

en la cual N_{lc} es una matriz de coeficientes de $N(s)$.

Para poder hacer el cálculo correspondiente de las matrices de la realización, controlador, se construyen las siguientes matrices

$$\begin{aligned} A_c^0 &= block\_diag\left\{\begin{bmatrix} 0 & & & 0 & 0 \\ 1 & & & & 0 \\ . & & & & \\ 0 & & & & \\ 0 & 0 & & 1 & 0 \end{bmatrix}, k_i \times k_i, i = 1, ..., m\right\} \\ [B_c^0]^T &= block\_diag\left\{\begin{bmatrix} 1 & 0 & ... & 0 & 0 \end{bmatrix}, 1 \times k_i, i = 1, ..., m\right\} \\ C_c^0 &= I_n \end{aligned}$$

finalmente se tiene

$$\begin{aligned} A_c &= A_c^0 - B_c^0 D_{hc}^{-1} D_{lc} && (3.64) \\ B_c &= B_c^0 D_{hc}^{-1} && (3.65) \\ C_c &= N_{lc} && (3.66) \end{aligned}$$

que son las matrices de la realización en forma controlador $\{A_c, B_c, C_c\}$.

La forma controlador puede ser obtenida, por dualidad, a partir de la forma observador

$$\begin{aligned} [A_x, B_x, C_x] &= fobmfd(N_R^T, D_R^T) \\ A_c &= A_x^T \\ B_c &= C_x^T \\ C_c &= B_x^T \end{aligned}$$

donde "$fobmfd$" es la función para obtener la forma observador.

FCOMFD.M

A continuación se describe el algoritmo, FCOMFD, cuya función es obtener una realización en forma controlador, $\{A_c, B_c, C_c\}$, a partir de una factorización derecha de una matriz de transferencia, $H(s) = N(s)D^{-1}(s)$.

Algoritmo

```
SOLUCIÓN DE DHC
[NY,Dhc]←—iscolred(D);
SE OBTIENE LA DIMENSIÓN DE N(S) Ó H(S)
[renh,colh]←—dimensión_de N;
INICIO DE PROCESO
m1 ←—1;
m←—1;
for j=1:colh
   k←—1;
   for i=2:colh
      if grado_de d_kj < grado_de d_ij
         k←—i;
      end
   end
OBTENER LOS K_i (GRADOS POR COLUMNA DE D(S))
   ki_j ←—deg(d_kj);
PARA SOLUCIONAR Y(S)
   for h=deg(d_kj)-1:-1:0
      m←—m+1;
   end
PARA SOLUCIONAR DE DLC
   for f=1:colh
      q←—1;
      for g=m1:m-1
         d_lc[fg] ←—coeff(d_fj,deg(d_kj)-q);
         q←—q+1;
      end
   end
PARA SOLUCIONAR NLC
   for f=1:renh
      q←—1;
      for g=m1:m-1
         n_lc[fg] ←—coeff(n_fj,deg(d_kj)-q);
         q←—q+1;
      end
   end
   m1 ←—m;
end
CREACIÓN DE AOC
dor←—0;
for i=1:colh
   for j=2:ki_i
      Aoc_(j+dor)(j-1+dor) ←—1;
   end
```

```
   dor⟵dor+ki_i;
end
CREACIÓN DE BOC
dor⟵1;
for i=1:colh
   Boc(dor)(i) ⟵1;
   dor⟵dor+ki_i;
end
PARA SOLUCIONAR AC=AOC-BOC*DHC^-1*DLC
Ac⟵Aoc-Boc*inversa(Dhc)*Dlc;
PARA SOLUCIONAR BC=BOC*DHC^-1
Bc⟵Boc*inversa(Dhc);
PARA SOLUCIONAR CC=NLC
Cc⟵Nlc;
```

■

3.3.2 Forma Observador

Es posible obtener una realización en forma observador, $\{A_o, B_o, C_o\}$, a partir de una factorización izquierda de una matriz de transferencia, $H(s) = D_L^{-1}(s)N_L(s)$. Es necesario que $D_L(s)$ sea reducida por filas, de lo contrario se aplica (Ec.3.29[3]). La realización es de orden, n, igual al grado del determinante de $D_L(s)$

$$n = deg\_detD_L(s) = \sum_{1}^{p} l_i \tag{3.67}$$

donde

$l_1 \leq l_2 \leq ... \leq l_p$

$l_i, i = 1, ..., p$ es el grado de la i-ésima fila de $D_L(s)$.

Se tiene

$$\begin{aligned} H(s) &= C_o(SI - A_o)^{-1}B_o & (3.68)\\ &= D_L^{-1}(s)N_L(s) & (3.69) \end{aligned}$$

donde $H(s)$ es una matriz estrictamente propia. La matriz $H(s)$ es de dimensión $p \times m$, esto es

p es el número de filas de $H(s)$

m es el número de columnas de $H(s)$

[3]Si se ha modificado, con ciertas operaciones, la matriz $D(s)$ para hacerla reducida por filas, de igual manera se debe modificar $N(s)$.

La matriz $D(s)$ se descompone como

$$D_L(s) = S_L(s)D_{hr} + \Psi_L(s)D_{lr} \tag{3.70}$$

donde

D_{hr} es la matriz de coeficientes líderes de $D_L(s)$ (Definición 41).

D_{lr} es una matriz con los coeficientes de los términos de $D_L(s)$ no incluidos en D_{hr}.

$$\begin{aligned} S_L(s) &= diag\{s^{l_i}, i = 1, ..., p\} && (3.71) \\ \Psi_L(s) &= block\_diag\{[s^{l_i-1}, ..., s, 1], ..., i = 1, ..., p\} && (3.72) \end{aligned}$$

$N_L(s)$, de la (Ec.3.69), se factoriza como

$$N_L(s) = \Psi_L(s)N_{lr} \tag{3.73}$$

en la cual N_{lr} es una matriz de coeficientes de $N(s)$.

Para poder hacer el cálculo correspondiente de las matrices de la realización observador se construyen las siguientes matrices

$$\begin{aligned} A_o^0 &= block\_diag\left\{\begin{bmatrix} 0 & 1 & & 0 & 0 \\ & & & & 0 \\ & & & & \\ 0 & & & & 1 \\ 0 & 0 & & & 0 \end{bmatrix}, l_i \times l_i, i = 1, ..., p\right\} \\ B_o^0 &= I_n \\ C_o^0 &= block\_diag\left\{\begin{bmatrix} 1 & 0 & ... & 0 & 0 \end{bmatrix}, 1 \times l_i, i = 1, ..., p\right\} \end{aligned}$$

finalmente se tiene

$$\begin{aligned} A_o &= A_o^0 - D_{lr}D_{hr}^{-1}C_o^0 && (3.74) \\ B_o &= N_{lr} && (3.75) \\ C_o &= D_{hr}^{-1}C_o^0 && (3.76) \end{aligned}$$

que son las matrices de la realización en forma observador $\{A_o, B_o, C_o\}$.

La forma observador puede ser obtenida, por dualidad, a partir de la forma controlador

$$\begin{aligned} [A_x, B_x, C_x] &= fcomfd(N_L^T, D_L^T) \\ A_o &= A_x^T \\ B_o &= C_x^T \\ C_o &= B_x^T \end{aligned}$$

donde "$fcomfd$" es la función para obtener la forma controlador.

FOBMFD.M

A continuación se describe el algoritmo, FOBMFD, cuya función es obtener una realización en forma observador, $\{A_o, B_o, C_o\}$, a partir de una factorización izquierda de una matriz de transferencia, $H(s) = D_L^{-1}(s)N_L(s)$.

Algoritmo .

SE TRANSPONEN LAS MATRICES DE ENTRADA

NT⟵transpuesta_de N;

DT⟵transpuesta_de D;

SE LLAMA A LA FUNCION CONTROLADOR

[A,B,C]⟵fcomfd(NT,DT);

SE OBTIENEN LAS MATRICES DE LA REALIZACIÓN EN FORMA OBSERVADOR

Ao⟵transpuesta_de A;

Bo⟵transpuesta_de B;

Co⟵transpuesta_de C;

■

3.3.3 Forma Controlabilidad

Es posible obtener una realización en forma controlabilidad, $\{A_{co}, B_{co}, C_{co}\}$, a partir de una factorización derecha de una matriz de transferencia, $H(s) = N(s)D^{-1}(s)$. Es necesario que $D(s)$ sea reducida por filas, de lo contrario se aplica (Ec.3.29[4]). La realización es de orden, n, igual al grado del determinante de $D(s)$

$$n = deg\_detD(s) = \sum_{1}^{p} l_i \tag{3.77}$$

donde

$l_1 \leq l_2 \leq ... \leq l_p$

$l_i, i = 1, ..., p$ es el grado de la i-ésima fila de $D(s)$.

Se tiene

$$\begin{aligned} H(s) &= C_{co}(SI - A_{co})^{-1}B_{co} && (3.78)\\ &= N(s)D(s)^{-1} && (3.79) \end{aligned}$$

donde $H(s)$ es una matriz estrictamente propia. La matriz $H(s)$ es de dimensión $p \times m$, esto es

p es el número de filas de $H(s)$

m es el número de columnas de $H(s)$

La matriz $D(s)$ se descompone como

$$D(s) = S(s)D_{hr} + \Psi(s)D_{lr} \tag{3.80}$$

en la cual

D_{hr} es la matriz de coeficientes líderes de $D(s)$ (Definición 41).

[4]Si se ha modificado, con ciertas operaciones, la matriz $D(s)$ para hacerla reducida por filas, de igual manera se debe modificar $N(s)$.

D_{lr} es una matriz con los coeficientes de los términos de $D(s)$ no contemplados en D_{hr}.

$$\begin{aligned} S(s) &= diag\{s^{l_i}, i = 1, ..., p\} && (3.81) \\ \Psi(s) &= block\_diag\{[s^{l_i-1}, ..., s, 1], i = 1, ..., p\} && (3.82) \end{aligned}$$

Se construyen las siguientes matrices

$$\begin{aligned} A_o^0 &= block\_diag\left\{\begin{bmatrix} 0 & 1 & & 0 & 0 \\ & & & & 0 \\ & & . & & \\ 0 & & & & 1 \\ 0 & 0 & . & . & 0 \end{bmatrix}, l_i \times l_i, i = 1, ..., p\right\} \\ C_o^0 &= block\_diag\left\{\begin{bmatrix} 1 & 0 & & 0 \end{bmatrix}, 1 \times l_i, i = 1, ..., p\right\} \\ \Upsilon &= \begin{bmatrix} 0 & 0 & & 1 \\ 0 & & . & \\ . & . & & \\ & 1 & & 0 \\ 1 & & 0 & 0 \end{bmatrix} \\ T &= block\_diag\left\{\Upsilon_{l_i}, i = 1, ..., p\right\} \end{aligned}$$

$N(s)$, de la (Ec.3.79), se factoriza como

$$\tilde{N}(s) = N(s)\,[C_oT] \qquad (3.83)$$

donde

$$C_oT = D_{hr}^{-1}C_o^0 \qquad (3.84)$$

Para poder hacer el cálculo correspondiente de las matrices de la realización controlabilidad es necesario obtener las matrices de la forma observador

$$\begin{aligned} A_o &= A_o^0 - D_{lr}D_{hr}^{-1}C_o^0 && (3.85) \\ C_o &= D_{hr}^{-1}C_o^0 && (3.86) \end{aligned}$$

finalmente se tiene

$$\begin{aligned} A_{co} &= TA_oT && (3.87) \\ [B_{co}]^T &= block\_diag\left\{\begin{bmatrix} 1 & 0 & ... & 0 & 0 \end{bmatrix}, 1 \times l_i, i = 1, ..., p\right\} && (3.88) \\ C_{co} &= \tilde{N}(A_{co}) && (3.89) \end{aligned}$$

que son las matrices de la realización en forma controlabilidad $\{A_{co}, B_{co}, C_{co}\}$.

La forma controlabilidad puede ser obtenida, por dualidad, a partir de la forma observabilidad

$$\begin{aligned} [A_x, B_x, C_x] &= fodadmfd(N_R^T, D_R^T) \\ A_{co} &= A_x^T \\ B_{co} &= C_x^T \\ C_{co} &= B_x^T \end{aligned}$$

donde "$fodadmfd$" es la función para obtener la forma observabilidad.

FCDADMFD.M

A continuación se describe el algoritmo, FCDADMFD, cuya función es obtener una realización en forma controlabilidad, $\{A_{co}, B_{co}, C_{co}\}$, a partir de una factorización derecha de una matriz de transferencia, $H(s) = N(s)D^{-1}(s)$.

Algoritmo

```
SOLUCIÓN DE DHR
[siono,Dhr]←—isrowred(D);
DE UNA FACTORIZACIÓN DERECHA SE PROCESA PARCIALMENTE LA FORMA
OBSERVADOR Y SE CONCLUYE LA FORMA CONTROLABILIDAD.
CÁLCULO DE MATRICES OBSERVADOR POR MFDS.
SE OBTIENE LA DIMENSIÓN DE N(S) Ó H(S)
[renh,colh]←—dimensión_de N;
INICIO DE PROCESO
m1 ←—1;
m←—1;
for j=1:colh
   k←—1;
   for i=2:colh
      if deg(d_jk) < deg(d_ji)
         k←—i;
      end
   end
OBTENER LOS KI(GRADOS POR FILA DE D(S))
   ki_j ←—deg(d_jk);
PARA SOLUCIONAR COO & BCO
   ban1←—0;
   for i=deg(d_jk)-1:-1:0
      if ban1==0
         c_oo[jm] ←—1;
         ban1←—1;
      end
      m←—m+1;
   end
PARA SOLUCIÓN DE DLR
   for f=1:colh
      q←—1;
      for i=m1:m-1
         d_lr[if]=coeff(d_jf,deg(d_jk)-q);
         q←—q+1;
      end
   end
   m1 ←—m;
```

```
end
CREACIÓN DE AOO
dor⟵0;
for i=1:renh
   for j=2:ki_i
      a_oo[(j-1+dor)(j+dor)] ⟵1;
   end
   dor⟵dor+ki_i;
end
PARA SOLUCIONAR AO=AOO-DLR*DHR^-1*COO
Ao⟵Aoo-Dlr*inversa(Dhr)*Coo;
PARA SOLUCIONAR CO=DHR^-1*COO
Co⟵inversa(Dhr)*Coo;
VARIABLE MAX
max⟵0;
for j=1:renh
   for i=1:colh
      if deg(n_ji) > max
         max⟵deg(n_ji);
      end
   end
end
EXPANSIÓN DEL PROGRAMA PARA CONTROLABILIDAD
CREACIÓN DE T
k⟵0;
dor⟵1;
for i=1:colh
   m⟵0;
   k⟵k+ki_i;
   for j=dor:k
      T[(k-m)j] ⟵1;
      m⟵m+1;
   end
   dor⟵dor+ki_i;
end
PARA SOLUCIONAR ACO=T*AO*T
Aco⟵T*Ao*T;
PARA SOLUCIONAR BCO=BLOCK_DIAG{(1 0 ... 0), 1XLI,I=1,...,P}
Bco⟵transpuesta_de Coo;
PARA SOLUCIONAR CCO
NN⟵N*Co*T;
Ad⟵Aco;
for m=0:max
  for i=1:renh
    for j=1:deg(determinante_de D)
```

SE OBTIENE EL ORDEN DE LA REALIZACIÓN

DeG$\longleftarrow$deg(determinante_de D);

D_{lc} es una matriz con los coeficientes de los términos de $D_L(s)$ no contemplados en D_{hc}.

$$\begin{aligned} S(s) &= diag\{s^{k_i}, i = 1, ...m\} && (3.94)\\ \Psi^T(s) &= block\_diag\{[s^{k_i-1}, ..., s, 1], i = 1, ..., m\} && (3.95) \end{aligned}$$

Se construyen las siguientes matrices

$$\begin{aligned} A_c^0 &= block\_diag\left\{\begin{bmatrix} 0 & & 0 & 0\\ 1 & & & 0\\ & & & \\ 0 & & & \\ 0 & 0 & 1 & 0 \end{bmatrix}, k_i \times k_i, i = 1, ..., m\right\}\\ [B_c^0]^T &= block\_diag\left\{\begin{bmatrix} 1 & 0 & & 0 \end{bmatrix}, 1 \times k_i, i = 1, ..., m\right\}\\ \Upsilon &= \begin{bmatrix} 0 & 0 & & 1\\ 0 & & & \\ & & & \\ & 1 & & 0\\ 1 & & 0 & 0 \end{bmatrix}\\ T &= block\_diag\left\{\Upsilon_{k_i}, i = 1, ..., m\right\} \end{aligned}$$

$N_L(s)$, de la (Ec.3.92), se factoriza como

$$\tilde{N}(s) = [TB_o]\, N_L(s) \qquad (3.96)$$

donde

$$[TB_o] = B_o^0 D_{hc}^{-1} \qquad (3.97)$$

Para poder hacer el cálculo correspondiente de las matrices de la realización observabilidad es necesario obtener las matrices de la forma controlador

$$\begin{aligned} A_c &= A_c^0 - B_c^0 D_{hc}^{-1} D_{lc} && (3.98)\\ B_c &= B_c^0 D_{hc}^{-1} && (3.99) \end{aligned}$$

finalmente se tiene

$$\begin{aligned} A_{ob} &= TA_cT && (3.100)\\ B_{ob} &= A_{ob}\tilde{N} && (3.101)\\ C_{ob} &= block\_diag\left\{\begin{bmatrix} 1 & 0 & & 0 \end{bmatrix}, 1 \times k_i, 1 = 1, ..., m\right\} && (3.102) \end{aligned}$$

que son las matrices de la realización en forma observabilidad $\{A_{ob}, B_{ob}, C_{ob}\}$.

La forma observabilidad puede ser obtenida, por dualidad, a partir de la forma controlabilidad

$$\begin{aligned} [A_x, B_x, C_x] &= fcdadmfd(N_L^T, D_L^T)\\ A_{ob} &= A_x^T\\ B_{ob} &= C_x^T\\ C_{ob} &= B_x^T \end{aligned}$$

donde "$fcdadmfd$" es la función para obtener la forma controlabilidad.

FODADMFD.M

A continuación se describe el algoritmo, FODADMFD, cuya función es obtener una realización en forma observabilidad, $\{A_{ob}, B_{ob}, C_{ob}\}$, a partir de una factorización izquierda de una matriz de transferencia, $H(s) = D_L^{-1}(s)N_L(s)$.

Algoritmo

SE TRANSPONEN LAS MATRICES DE ENTRADA
NT⟵transpuesta_de N;
DT⟵transpuesta_de D;
SE LLAMA A LA FUNCION CONTROLABILIDAD
[A,B,C]⟵fcdadmfd(NT,DT);
SE OBTIENEN LAS MATRICES DE LA REALIZACIÓN EN FORMA OBSERVABILIDAD
Aob⟵transpuesta_de A;
Bob⟵transpuesta_de B;
Cob⟵transpuesta_de C;
■

3.3.5 Forma Canónica Controlabilidad

Es posible obtener una realización en forma canónica controlabilidad, $\{A_{ccc}, B_{ccc}, C_{ccc}\}$, a partir de una factorización derecha de una matriz de transferencia, $H(s) = N(s)D^{-1}(s)$. Es necesario que $D(s)$ sea reducida por columnas (Definición 42), de lo contrario se aplica (Ec.3.28[6]). La realización es de orden, n, igual al grado del determinante de $D(s)$

$$n = deg\_detD(s) = \sum_{1}^{m} k_i \tag{3.103}$$

Se tiene

$$\begin{aligned} H(s) &= C_{ccc}(SI - A_{ccc})^{-1}B_{ccc} && (3.104)\\ &= N(s)D^{-1}(s) && (3.105) \end{aligned}$$

donde $H(s)$ es una matriz estrictamente propia. La matriz $H(s)$ es de dimensión $p \times m$, esto es

p es el número de filas de $H(s)$
m es el número de columnas de $H(s)$

[6] Si se ha modificado, con ciertas operaciones, la matriz $D(s)$ para hacerla reducida por columnas, de igual manera se debe modificar $N(s)$.

Se hace uso del algoritmo descrito antes, para obtener las matrices de la forma controlador

$$\begin{aligned} A_c &= A_c^0 - B_c^0 D_{hc}^{-1} D_{lc} \\ B_c &= B_c^0 D_{hc}^{-1} \\ C_c &= N_{lc} \end{aligned} \tag{3.106}$$

En el cálculo de la realización, en forma canónica controlabilidad, se requiere de una matriz de transformación, la cual se obtiene utilizando el modelo I de CRATE [Kail80]

$$T = \{\mathrm{b}_1, \mathrm{Ab}_1, ..., \mathrm{A}^{r_1-1}\mathrm{b}_1, \mathrm{b}_2, \mathrm{Ab}_2, ..., \mathrm{A}^{r_2-1}\mathrm{b}_2, ..., \mathrm{b}_m, \mathrm{Ab}_m, ..., \mathrm{A}^{r_m-1}\mathrm{b}_m\} \tag{3.107}$$

donde

$$\sum r_i = n$$

finalmente se tiene

$$A_{ccc} = T^{-1} A_c T \tag{3.108}$$
$$B_{ccc} = T^{-1} B_c \tag{3.109}$$
$$C_{ccc} = C_c T \tag{3.110}$$

que son las matrices de la realización en forma canónica controlabilidad $\{A_{ccc}, B_{ccc}, C_{ccc}\}$.

La forma canónica controlabilidad puede ser obtenida, por dualidad, a partir de la forma canónica observabilidad

$$\begin{aligned} [A_x, B_x, C_x] &= fccodad(N_R^T, D_R^T) \\ A_{ccc} &= A_x^T \\ B_{ccc} &= C_x^T \\ C_{ccc} &= B_x^T \end{aligned}$$

donde "$fccodad$" es la función para obtener la forma canónica observabilidad.

FCCCDAD.M

A continuación se describe el algoritmo, FCCCDAD, cuya función es obtener una realización en forma canónica controlabilidad, $\{A_{ccc}, B_{ccc}, C_{ccc}\}$, a partir de una factorización derecha de una matriz de transferencia, $H(s) = N(s)D^{-1}(s)$.

Algoritmo

SE OBTIENEN LAS MATRICES DE LA FORMA CONTROLADOR
[Ac,Bc,Cc]⟵fcomfd(N,D);
SE OBTIENE EL ORDEN DE LA REALIZACIÓN
DeG⟵deg(determinante_de D);

```
EXPANSIÓN PARA FORMA CANÓNICA DE CONTROLABILIDAD
SE USA EL MODELO I DE CRATE [12]
for i=1:número_de_columnas_de N
   b:i ←— Bc:i;
end
dor←—1;
pot←—0;
j←—0;
while j≠DeG
   j←—j+1;
   for i=1:DeG
      r←—Ac^pot*b:dor;
      tij ←—ri;
   end
   if rango_de T ≠ j
      dor←—dor+1;
      j←—j-1;
      pot←—-1;
   end
   pot←—pot+1;
end
A CANÓNICA CONTROLABILIDAD
Aco←—inversa(T)*Ac*T;
B CANÓNICA CONTROLABILIDAD
Bco←—inversa(T)*Bc;
C CANÓNICA CONTROLABILIDAD
Cco←—Cc*T;
```

■

3.3.6 Forma Canónica Observabilidad

Es posible obtener una realización en forma canónica observabilidad, $\{A_{cco}, B_{cco}, C_{cco}\}$, a partir de una factorización izquierda de una matriz de transferencia, $H(s) = D_L^{-1}(s)N_L(s)$. Es necesario que la matriz $D_L(s)$ sea reducida por filas (Definición 41), de lo contrario se aplica (Ec.3.29[7]). La realización es de orden, n, igual al grado del determinante de $D_L(s)$

$$n = deg\_detD_L(s) = \sum_{1}^{p} l_i \tag{3.111}$$

Se tiene

$$\begin{aligned} H(s) &= C_{cco}(SI - A_{cco})^{-1}B_{cco} &(3.112)\\ &= D_L^{-1}(s)N_L(s) &(3.113)\end{aligned}$$

[7] Si se ha modificado, con ciertas operaciones, la matriz $D(s)$ para hacerla reducida por filas, de igual manera se debe modificar $N(s)$.

donde $H(s)$ es una matriz estrictamente propia. Los índices se identifican como

$$\begin{array}{l} p \text{ es el número de filas de } H(s) \\ m \text{ es el número de columnas de } H(s) \end{array}$$

Se apela a la dualidad para obtener ésta realización.

La forma canónica observabilidad puede ser obtenida, por dualidad, a partir de la forma canónica controlabilidad

$$\begin{aligned} [A_x, B_x, C_x] &= fcccdad(N_L^T, D_L^T) \\ A_{cco} &= A_x^T \\ B_{cco} &= C_x^T \\ C_{cco} &= B_x^T \end{aligned}$$

donde "$fcccdad$" es la función para obtener la forma canónica controlabilidad.

FCCODAD.M

A continuación se describe el algoritmo, FCCODAD, cuya función es obtener una realización en forma canónica observabilidad, $\{A_{cco}, B_{cco}, C_{cco}\}$, a partir de una factorización izquierda de una matriz de transferencia, $H(s) = D_L^{-1}(s)N_L(s)$.

Algoritmo .

SE TRANSPONEN LAS MATRICES DE ENTRADA

NT⟵transpuesta_de N;

DT⟵transpuesta_de D;

SE LLAMA A LA FUNCION CANÓNICA CONTROLABILIDAD

[A,B,C]⟵fcccdad(NT,DT);

SE OBTIENEN LAS MATRICES DE LA REALIZACIÓN EN FORMA CANÓNICA OBSERVABILIDAD

Acco⟵transpuesta_de A;

Bcco⟵transpuesta_de B;

Ccco⟵transpuesta_de C;

■

En los algoritmos que se acaban de describir,"fcomfd",...,"fccodad", se muestra como obtener una realización en variables de estado a partir de una MFD. Se sabé, también, que una MFD es obtenida a partir de una matriz de transferencia. Para revertir la operación anterior, existe la instrucción **"sys2tm"** la cual a partir de un sistema de ecuaciones, $\{A, B, C, D\}$, obtiene la matriz de transferencia correspondiente.

3.4 Forma Smith-McMillan

Una forma canónica, para matrices racionales, es la forma Smith-McMillan. Una matriz de transferencia puede ser escrita como

$$H(s) = \frac{1}{d(s)}N(s) \tag{3.114}$$

donde $N(s)$ es una matriz polinomial y $d(s)$ es el mínimo común múltiplo de los denominadores de $H(s)$, (mónico). $N(s)$ puede ser escrita como

$$N(s) = H(s)d(s) \tag{3.115}$$

Si $N(s)$ tiene la forma Smith $S(s)$, entonces

$$N(s) = U(s)S(s)V(s) \tag{3.116}$$

donde $U(s)$, $V(s)$, son matrices unimodulares apropiadas. La forma Smith-McMillan, $M(s)$, está definida como

$$M(s) = \frac{1}{d(s)}S(s) \tag{3.117}$$

donde

$$M(s) = \begin{bmatrix} M^*(s) & 0 \\ 0 & 0 \end{bmatrix} \tag{3.118}$$

$$M^*(s) = diag\left\{\frac{\lambda_i(s)}{d(s)}\right\} \tag{3.119}$$

$$= diag\left\{\frac{\epsilon_i(s)}{\psi_i(s)}\right\} \tag{3.120}$$

donde

$$\begin{aligned} \{\epsilon_i(s), \psi_i(s) \text{ son polinomios coprimos}\}, i &= 1, ..., r \\ r &= \text{rango de } H(s). \end{aligned}$$

ahora, entonces

$$H(s) = U(s)M(s)V(s) \tag{3.121}$$

donde $M(s)$ es la forma de Smith-McMillan.

Lema 57 *La forma de Smith MacMillan ($M(s)$) de una matriz racional, $H(s)$, es única.*

SMITHMAC.M

A continuación se describe el algoritmo, SMITHMAC.M, cuya función es obtener la matriz Smith MacMillan, $M(s)$, de una matriz $H(s)$.

Algoritmo
INICIO
D⟵1;
[renh,colh]⟵dimensión_de H;
OBTENER EL MÍNIMO COMÚN MÚLTIPLO DE LOS DENOMINADORES DE H(S)
for i=1:renh

```
    for j=1:colh
        D⟵mínimo_común_múltiplo_entre D y_el_denominador_de H_ij;
    end
end
OBTENER LA MATRIZ NUMERADOR
for i=1:renh
    for j=1:colh
        N_ij ⟵H_ij*D;
    end
end
OBTENER LA MATRIZ DE SMITH
S⟵matriz_de_Smith_de N;
OBTENER LA MATRIZ SMITH MACMILLAN
for i=1:rango_de H
    M_ii ⟵S_ii÷D;
end
```

■

3.5 Polos y ceros de sistemas multivariables

La generalización del concepto de polos y ceros de una función de transferencia escalar al caso general de una matriz de transferencia, se hace a través de la forma Smith-McMillan **[15]**.

Definición 58 *Los ceros de transmisión son las raíces de los polinomios* $\{\epsilon_i(s)\}$ *(Ec.3.120), y los polos del sistema son las raíces de los polinomios* $\{\psi_i(s)\}$ *(Ec.3.120).*

Nota 59 *En sistemas multivariables un mismo valor de frecuencia puede ser un polo y un cero al mismo tiempo, a diferencia del caso escalar.*

Lema 60 *El orden de una realización mínima,* $\{A, B, C\}$*, de la matriz de transferencia* $H(s)$ *es igual a la suma de los grados de* $\psi_i(s)$ *(grado de MacMillan de* $H(s)$*), donde* $\psi_i(s)$ *son los polinomios de los denominadores en la forma de Smith-McMillan de* $H(s)$*.*

Si se supone que se tiene una MFD derecha irreducible

$$H(s) = N(s)D^{-1}(s) \tag{3.122}$$

los polos de $H(s)$ son las raíces del $det(D(s)) = 0$, donde $D(s)$ es una matriz denominador de una MFD irreducible de $H(s)$. Si $H(s)$ es cuadrada y no singular, entonces los ceros (de transmisión), de $H(s)$, son las raíces del $det(N(s)) = 0$, donde $N(s)$ es una matriz numerador de una MFD irreducible de $H(s)$.

Los ceros de transmisión coinciden con las frecuencias a las cuales el rango, normal, de $N(s)$ disminuye.

Lema 61 *Todos los numeradores izquierdos y derechos de factorizaciones irreducibles de $H(s)$ tienen la misma forma de Smith.*

Lema 62 *Todos los denominadores izquierdos (derechos) de factorizaciones irreducibles de $H(s)$ tienen los mismos polinomios invariantes, $\psi_i(s)$, y por lo tanto el mismo determinante.*

Entonces:

Los polos de $H(s)$ son las raíces del determinante de $D(s)$, donde $D(s)$ es el denominador de cualquier factorización irreducible (izquierda, derecha) de $H(s)$.

Los ceros (de transmisión) de $H(s)$ son las raíces de los polinomios invariantes de $N(s)$, donde $N(s)$ es el numerador de cualquier factorización irreducible (izquierda, derecha) de $H(s)$.

Nota 63 *El conjunto de polos del sistema es un subconjunto del conjunto de los valores propios de la matriz A.*

3.5.1 Otras clasificaciones de ceros

Definición 64 *Si se tiene que $\{A, B, C, D\}$ es una realización, se define la matriz del sistema, $P(s)$, como*

$$P(s) = \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix} \tag{3.123}$$

Definición 65 *Los ceros de la matriz $P(s)$ (Ec.3.123) son conocidos como ceros invariantes del sistema. Los ceros de transmisión son un subconjunto de los ceros invariantes, y si la realización es mínima (controlable y observable) ambos conjuntos son iguales.*

Los ceros invariantes que no son ceros de transmisión, pertenecen a un conjunto conocido como ceros de desacoplamiento. Estos están asociados a una evolución de tipo exponencial de las variables de estado, la cual se encuentra desacoplada de la entrada o de la salida. Los ceros invariantes son asociados con la solución del problema de seleccionar las entradas y los estados, asociados a un valor de frecuencia compleja, en la cual la salida del sistema es cero.

Definición 66 *Los ceros de desacoplamiento de salida son las frecuencias a las cuales la matriz*

$$P_o(s) = \begin{bmatrix} sI - A \\ C \end{bmatrix} \tag{3.124}$$

pierde rango. Si un sistema tiene más salidas que entradas, entonces algunos ceros invariantes serán también ceros de desacoplamiento de salida. Estos ceros son asociados con los modos del sistema que no son observables.

Definición 67 *Los ceros de desacoplamiento de entrada son las frecuencias a las cuales la matriz*

$$P_i(s) = \begin{bmatrix} sI - A & \vdots & -B \end{bmatrix} \tag{3.125}$$

pierde rango. Si un sistema tiene más entradas que salidas, entonces algunos ceros invariantes serán también ceros de desacoplamiento de entrada. Estos ceros son asociados con los modos del sistema que no son controlables.

Definición 68 *Si $P_o(s)$ y $P_i(s)$ pierden rango a una frecuencia particular, $s = z$, entonces z es llamado cero de desacoplamiento entrada - salida.*

Definición 69 *Se definen los ceros del sistema como*

$$Csist = \{CdesEnt + CdesSal + Ctrans\} - \{Cdesentsal\}$$

donde
$Csist$.- *ceros del sistema*
$CdesEnt$.- *ceros de desacoplamiento de entrada*
$CdesSal$.- *ceros de desacoplamiento de salida*
$Ctrans$.- *ceros de transmisión*
$Cdesentsal$.- *ceros de desacoplamiento de entrada salida*

CEROS AL INFINITO DE SISTEMAS MULTIVARIABLES.

Otra clase de ceros, son los ceros al infinito. Estos proveen información que es importante en el estudio de desacoplamiento, rechazo a perturbaciones, etc., de los sistemas lineales.

Una forma de estudiar la estructura al infinito de $H(s)$, es hacer el cambio de variable $s = \frac{1}{\lambda}$ y obtener la forma de Smith MacMillan de $H(\frac{1}{\lambda})$. Entonces, la información de $H(s)$ en $s = \infty$, corresponde a la información de $H(\frac{1}{\lambda})$ en $\lambda = 0$.

La matriz de Smith MacMillan de $H(\frac{1}{\lambda})$, $M_\infty(s)$, es conocida como la forma de Smith MacMillan al infinito de $H(s)$.

PZMIMO.M

A continuación se describe el algoritmo, PZMIMO.M, cuya función es obtener los polos y ceros de una matriz $H(s) \in \mathbb{R}^{p\times m}(s)$.

Algoritmo

POLOS Y CEROS DE TRANSMISIÓN

SE HABILITA UNA VARIABLE
syms s;
SE OBTIENE LA MATRIZ DE TRANSFERENCIA
H⟵C*(sI-A)$^{-1}$*B+D;
SE OBTIENE LA MATRIZ SMITH MACMILLAN
[Mh]⟵Smith_MacMillan_de H;
INICIALIZAR VARIABLES
pnumerMh⟵1;
pdenomMh⟵1;
SE EXTRAEN LOS POLINOMIOS DE LOS NUMERADORES Y DENOMINADORES DE TAL FORMA QUE LAS RAÍCES DEL POLINOMIO NUMERADOR, SON LOS CEROS DE TRANSMISIÓN DEL SISTEMA Y LAS RAÍCES DEL POLINOMIO DEL DENOMINADOR, SON LOS POLOS DEL SISTEMA.
for j=1:rango_de Mh
    pnumerMh⟵pnumerMh*numerador_de $m_{h[jj]}$;

```
   pdenomMh←—pdenomMh*denominador_de m_h[jj];
end
SE OBTIENEN LAS RAÍCES DE LOS POLINOMIOS
ceros_de_transmisión←—raíces_de pnumerMh;
polos_del_sistema←—raíces_de pdenomMh;
CEROS AL INFINITO
———————————————————————————————
SE HACE UN CAMBIO DE VARIABLE S POR 1/W; PARA EL PROG. S=>1/S
G←—H(1/s);
SE OBTIENE LA MATRIZ SMITH MACMILLAN
[Mg]←—Smith_MacMillan_de G;
SE EXTRAE EL NUMERO DE RAICES IGUALES A CERO DE CADA NUMERADOR Y SE
FORMA UN VECTOR CON LOS CEROS AL INFINITO DEL SISTEMA.
for j=1:rango_de Mg
   raícesN←—raíces_del_ numerador_de m_g[jj];
   toto←—0;
   for i=1:size(raícesN,1)
      if raícesN_i1=0
         toto←—toto+1;
      end
   end
ceros_al_infinito←—[ceros_al_infinito, toto];
end
CEROS DE DESACOPLAMIENTO DE SALIDA
———————————————————————————————
SE CONSTRUYE LA MATRIZ [sI-A; C]'
matriz←—[sI-A; C];
SE OBTIENE LA MATRIZ SMITH MACMILLAN
[SMac]←—Smith_MacMillan_de matriz;
SE OBTIENEN LAS RAÍCES DEL DETERMINANTE DE LA MATRIZ DE S.MACMILLAN
ceros_de_desacoplamiento_de_salida←—raíces_de det(SMac);
CEROS DE DESACOPLAMIENTO DE ENTRADA
———————————————————————————————
SE CONSTRUYE LA MATRIZ [sI-A B]
matriz←—[sI-A B];
SE OBTIENE LA MATRIZ SMITH MACMILLAN
[SMac]←—Smith_MacMillan_de matriz;
SE OBTIENEN LAS RAÍCES DEL DETERMINANTE DE LA MATRIZ DE S.MACMILLAN
ceros_de_desacoplamiento_de_entrada←—raíces_de det(SMac);
CEROS INVARIANTES
———————————————————————————————
SE CONSTRUYE LA MATRIZ DEL SISTEMA (Ec.3.123)
matriz←—[sI-A -B; C D];
SE OBTIENE LA MATRIZ SMITH MACMILLAN
[SMac]←—Smith_MacMillan_de matriz;
```

SE OBTIENEN LAS RAÍCES DEL DETERMINANTE DE LA MATRIZ DE S.MACMILLAN
ceros_invariantes ⟵ raíces_de det(SMac);
■

Capítulo 4

Manual de usuario

4.1 El programa MATLAB

Matlab [**18**] es el acrónimo de **Matrix Lab**oratory y es un producto de MathWorks Inc. que apareció comercialmente en 1984. Inicialmente fue escrito en lenguaje Fortran; actualmente está escrito en lenguaje C [**17**].

Matlab es un ambiente de programación matemático interactivo donde se solucionan problemas en forma numérica. Utilizando la herramienta, **Symbolic Math Toolbox** [**19**], también permite la resolución de problemas en forma simbólica. **Matlab** es también un lenguaje de programación [**16**].

Matlab, como programa, tiene un diseño modular que consiste en tres partes: la interface de usuario llamada *Iris*, el núcleo algebráico básico y la *biblioteca* externa.

El Iris y el núcleo forman la parte más pequeña del sistema; están escritos en lenguaje C y son cargados al inicio de una sesión de **Matlab**. Iris cumple con las funciones de *parser* de entrada-salida, al realizar las tareas de despliegue de resultados, presentación de gráficos y las comunicaciones del sistema con el usuario.

El núcleo interpreta las instrucciones y lleva a cabo las operaciones aritméticas básicas. Contiene ciertas rutinas algebráicas, de uso frecuente, dispuestas en lenguaje de bajo nivel por razones de eficiencia.

Una característica de **Matlab** es que está formado de un pequeño núcleo y una gran biblioteca. El núcleo de **Matlab** contiene, entre otras cosas, los comandos del lenguaje de programación y algunas operaciones sencillas. El núcleo no puede ser accesado por el usuario.

La biblioteca de **Matlab** contiene miles de funciones, las cuales están escritas en el lenguaje de programación **Matlab**. Es decir, la mayoría del conocimiento matemático de **Matlab** ha sido codificado en lenguaje de programación **Matlab** y reside en forma de funciones en la biblioteca externa. Las herramientas complementarias de **Matlab**, tales como los Toolbox, son una colección de funciones. **Matlab** es un sistema compacto pues no satura la memoria de la computadora con su colección de funciones. Además, el usuario tiene acceso a tales funciones, es decir que puede copiarlas y modificarlas. El usuario también puede escribir sus propias librerías y anexarlas a la biblioteca del **Matlab**.

Se requiere, para utilizar las funciones presentadas en este manual, tener el software de **Matlab 5.x** [18] y **Symbolic Math Toolbox** [19]. Cabe decir que, el usuario puede solicitar ayuda en línea igual que con las funciones del **Matlab**.

Una caracteristica de éstas funciones es que manejan datos algebráicos con más de una literal y por tal razón, es menester definir la **variable de trabajo**; como por ejemplo: dada la expresión

$$(4)(g)(t)$$

1.- El coeficiente es $(4)(t)$ si la variable de trabajo es g
2.- El coeficiente es $(4)(g)$ si la variable de trabajo es t

Nota 70 *La variable de trabajo debe ser una literal.*

4.2 ¿Como habilitar las funciones?

Se adjunta a este escrito un disquete que contiene la caja de herramientas computacionales presentadas en este trabajo. Para hacer la instalación de está caja de herramientas se deben seguir los pasos siguientes:

1.0.- Dentro del subdirectorio "Toolbox" ubicado dentro del directorio de Matlab se debe crear una nueva carpeta, de nombre, por ejemplo "newtools"

c:\matlab\toolbox>*md newtools*

2.0.- Se debe transferir los archivos del disquete, adjunto a este escrito, al directorio "newtools"

3.0.- Para que el programa Matlab reconozca nueva ruta de búsqueda se presiona el icono "Path Browser", ubicado dentro del entorno de Matlab, Figura 4.1.

3.1.- Dentro del la ventana que se presenta, Figura 4.2, se presiona el icono "Add to Path" con lo que aparece una ventana, Figura 4.3, en la cual se busca el directorio "newtools" Presione el icono "ok" y se retornará a la ventana, Figura 4.2.

4.0.- Finalmente, se presiona el icono "Save Settings" para que se guarden las modificaciones. realizadas.

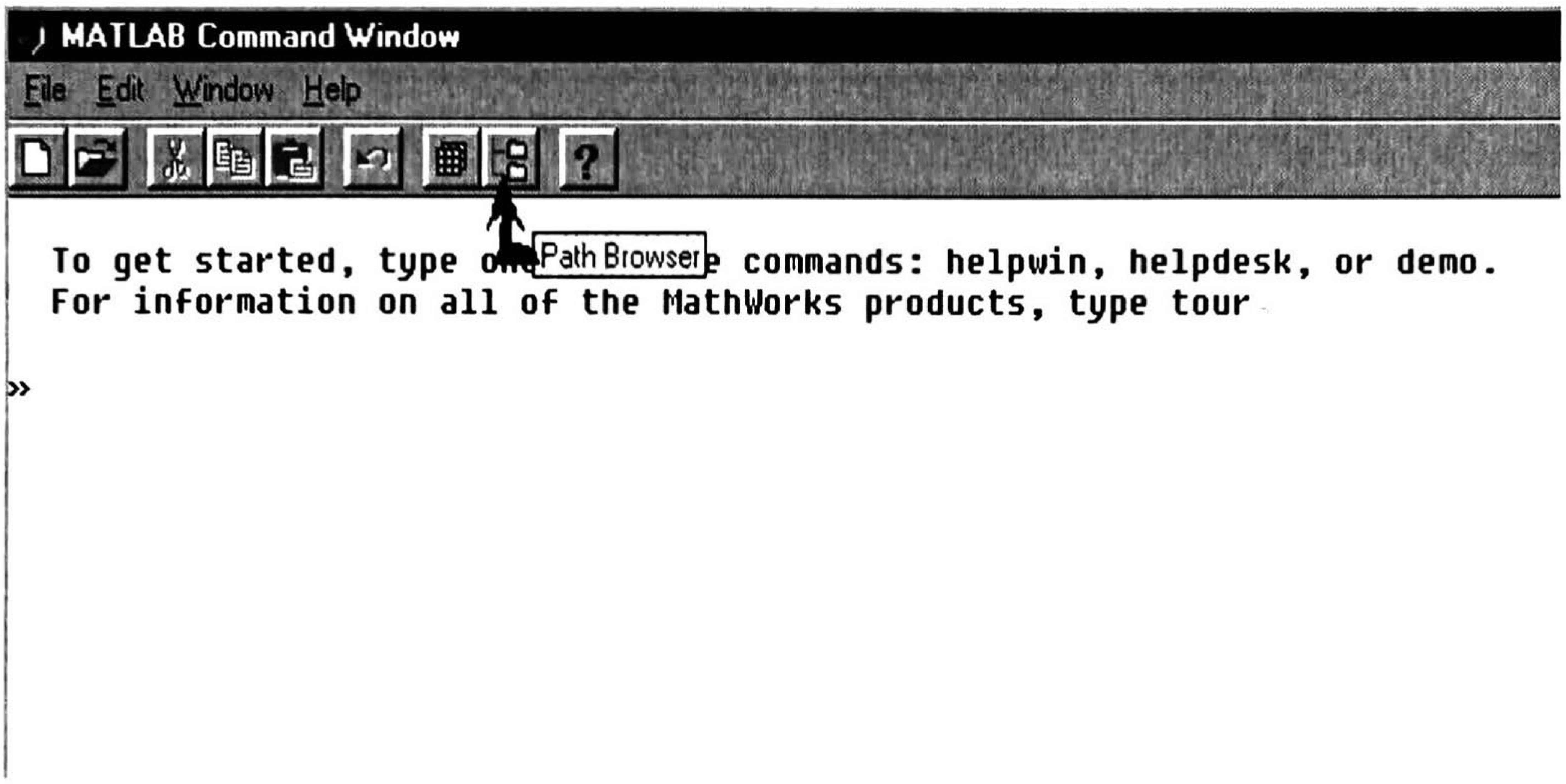

Figura 4.1: Icono de Path Browse

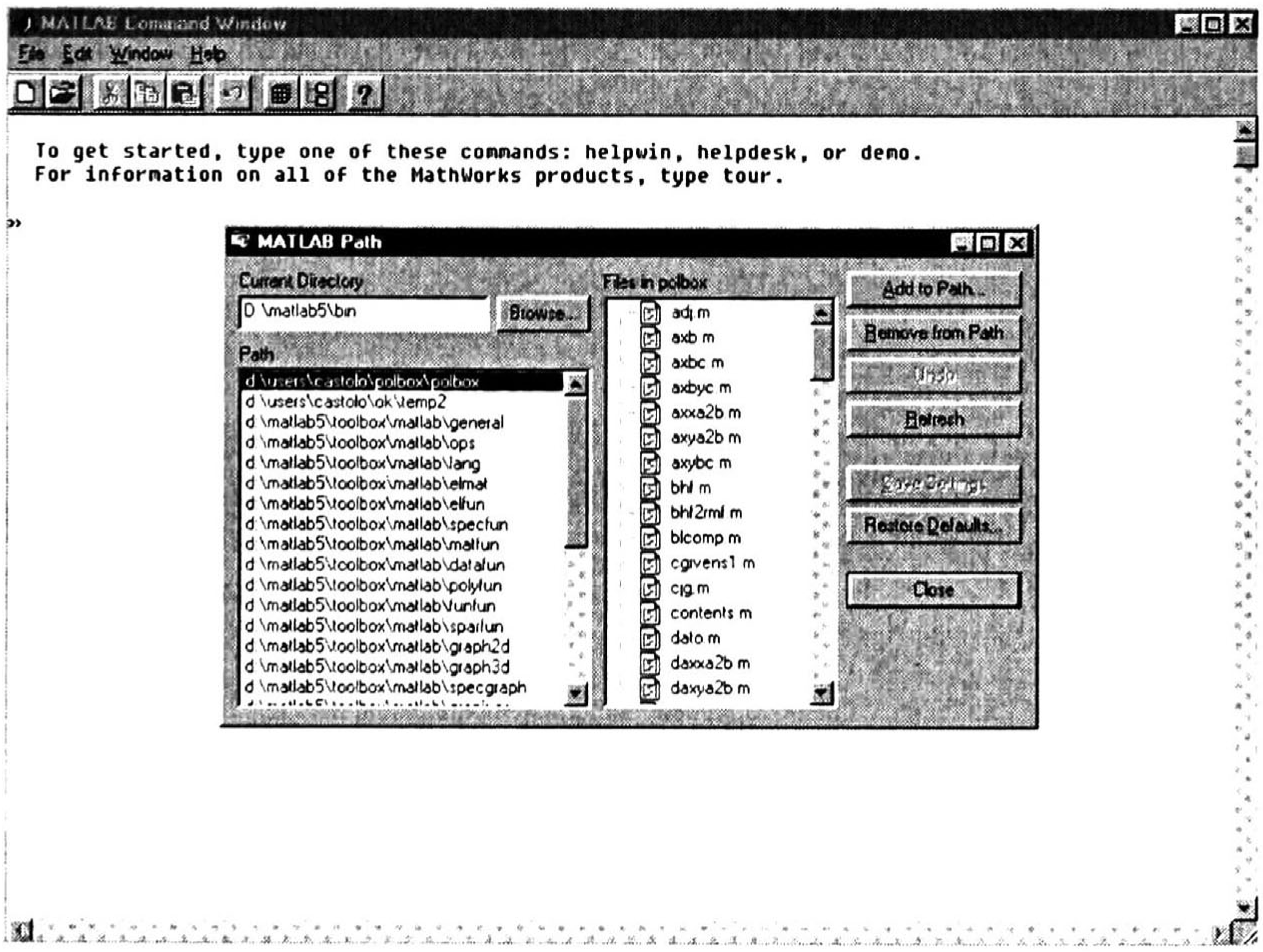

Figura 4.2: Ventana para adicionar rutas de busqueda

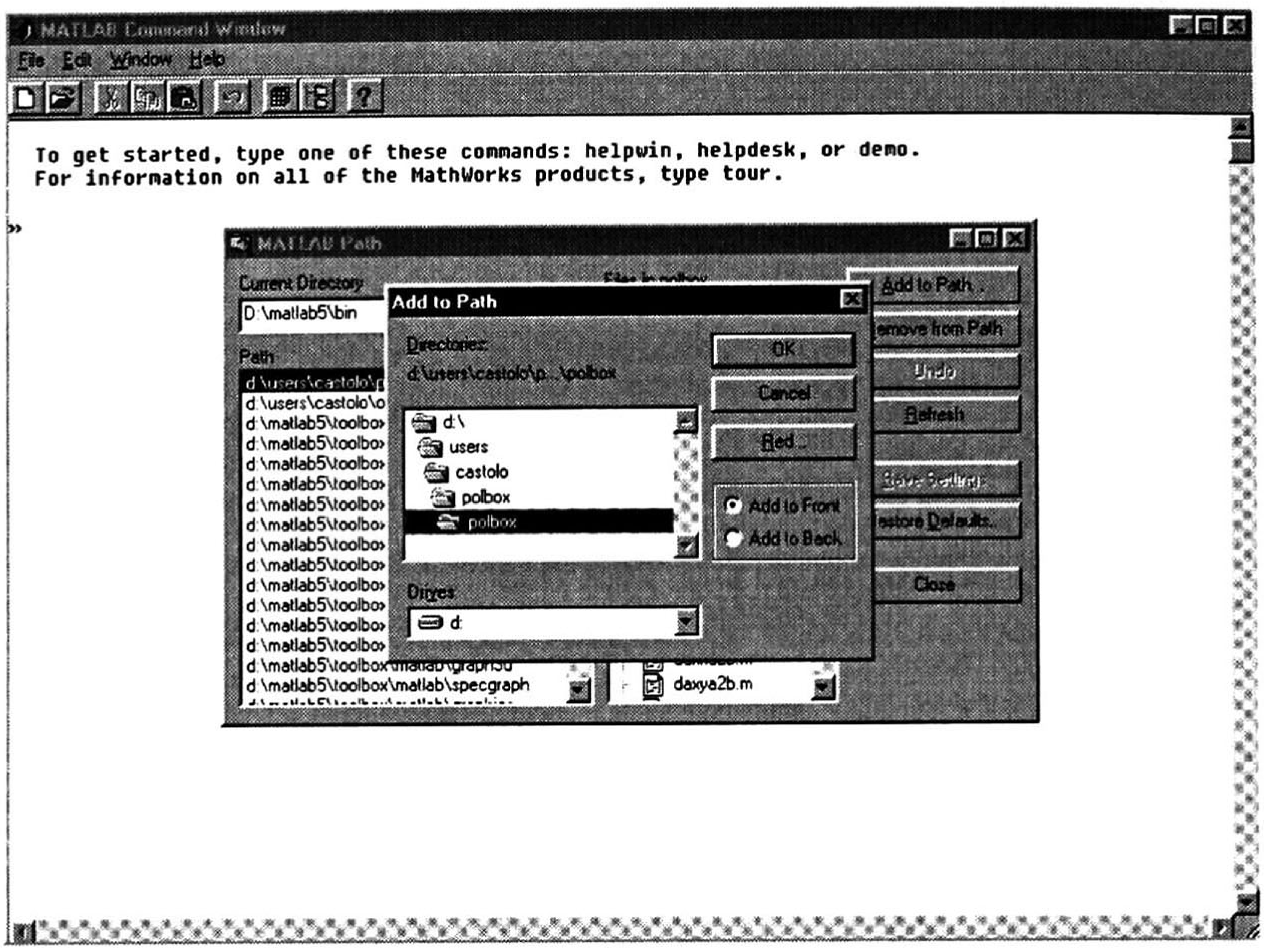

Figura 4.3: Busqueda del directorio "newtools"

TABLAS DE REFERENCIA RAPIDA

Factorizaciones	
Descripción en fracción matricial	mfd
Conversión de modelo	
Pasa una MFD izquierda a derecha	mfdiz2de
Pasa una MFD derecha a izquierda	mfdde2iz
Un sistema a matriz de transferencia	sys2tm
Realizaciones	
Forma canónica controlabilidad	fcccdad
Forma canónica observabilidad	fccodad
Forma controlabilidad	fcdadmfd
Forma controlador	fcomfd
Forma observador	fobmfd
Forma observabilidad	fodadmfd
Descomposiciones	
Una matriz de transferencia no estrictamente propia en una matriz estrictamente propia más una matriz residual	mpropsym
Utilerias	
Es polinomial	ispolnal
Son coprimos derechos	isrcopri
Son coprimos izquierdos	islcopri
Es reducida por filas	isrowred
Es reducida por columnas	iscolred
Propiedades del modelo	
Obtiene los polos y los ceros de un sistema MIMO	pzmimo
Gcrd de una MFD	gcrdmfd
Gcrld de una MFD	gcldmfd
Matrices	
Matriz identidad simbólica	eyesym
Forma de Smith MacMillan	smithmac
Funciones para matrices	
Obtiene el lcm de los den. por filas	rlcmsym
Obtiene el lcm de los den. por columnas	clcmsym

Figura 4.4: Tablas de referencia rápida

4.3 Funciones

4.3.1 fcccdad.m (Forma Canónica Controlabilidad)

Propósito

Obtener la realización en forma canónica controlabilidad a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD derecha; esto es $H(s) = N(s)D^{-1}(s)$.

Sintaxis

$$[A_{ccc}, B_{ccc}, C_{ccc}] = fcccdad(N, D, Variable)$$

donde las entradas son

$N = N(s) \in \mathbb{R}^{p\times m}[s]$
$D = D(s) \in \mathbb{R}^{m\times m}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_{ccc} \in \mathbb{R}^{n\times n}$
$B_{ccc} \in \mathbb{R}^{n\times m}$
$C_{ccc} \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización controlabilidad, que se obtiene a través de "fcccdad", es igual al grado del determinante de $D(s)$. Se requiere que $D(s)$ sea reducida por columnas. Se pide también que ningún k_i sea menor que uno, donde k_i, $i = 1, ..., m$ es el grado de la i-ésima columna de $D(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fcccdad"), respectivamente

$$H(s) = C_{ccc}(SI - A_{ccc})^{-1}B_{ccc} \tag{4.1}$$
$$= N(s)D^{-1}(s) \tag{4.2}$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Si H(s) es multiplicada por cualquier matriz unimodular y se hace uso de está función, "fcccdad", se obtendrá siempre la misma forma controlabilidad. Lo anterior no se aplica para la función "fcdadmfd"

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+2)^2(s+1)^2} & \frac{s}{(s+2)^2} \\ -\frac{s}{(s+2)^2} & -\frac{s}{(s+2)^2} \end{bmatrix}$$

donde

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) \\ D(s) &= \begin{bmatrix} 0 & -s^3-4s^2-5s-2 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} s & 0 \\ -s & -s^2 \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[0 -s^3-4*s^2-5*s-2; (s+2)^2 s+2];
>>N=[s 0; -s s^2];
>>[Accc,Bccc,Cccc]=fcccdad(N,D)

Accc =

   0,   0,   0,  -4,   2
   1,   0,   0, -12,   5
   0,   1,   0, -13,   4
   0,   0,   1,  -6,   1
   0,   0,   0,   0,  -2

Bccc =

 1, 0
 0, 0
 0, 0
 0, 0
 0, 1

Cccc =

   0,   0,   1,  -6,   1
  -1,   4, -12,  32,  -1
```

Diagnósticos

La función "fcccdad" regresa un mensaje de error

“Las_matrices_de_entrada_no_deben_estar_vacías”
“La_matriz_“D”_no_es_cuadrada”
“Incompatibilidad_de_matrices_de_entrada”
“Las_matrices_de_entrada_son_iguales”
“No_es_estrictamente_propia,_(N*invD)”
• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.
“Columna_en_“D”_con_grado_menor_que_uno”
• Alguna de las columnas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.
“La_matriz_“D”_no_es_reducida_por_columnas”

Ver También

mfd, fcomfd, fobmfd, fcdadmfd, fodadmfd, fccodad

4.3.2 fccodad.m (Forma Canónica Observabilidad)

Propósito

Obtener la realización en forma canónica observabilidad a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD izquierda; esto es $H(s) = D^{-1}(s)N(s)$.

Sintaxis

$$[A_{cco}, B_{cco}, C_{cco}] = fccodad(N, D, Variable)$$

donde las entradas son

$N = N(s) \in \mathbb{R}^{p\times m}[s]$
$D = D(s) \in \mathbb{R}^{p\times p}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_{cco} \in \mathbb{R}^{n\times n}$
$B_{cco} \in \mathbb{R}^{n\times m}$
$C_{cco} \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización observabilidad, que se obtiene a través de "fccodad", es igual al grado del determinante de $D(s)$. Se requiere que $D(s)$ sea reducida por filas. Se pide también que ningún l_i sea menor que uno, donde l_i, $i = 1, ..., p$ es el grado de la i-ésima fila de $D(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fccodad"), respectivamente

$$H(s) = C_{cco}(SI - A_{cco})^{-1}B_{cco} \tag{4.3}$$
$$= D^{-1}(s)N(s) \tag{4.4}$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Si H(s) es multiplicada por cualquier matriz unimodular y se hace uso de está función, "fccodad", se obtendrá siempre la misma forma observabilidad. Lo anterior no se aplica para la función "fodadmfd"

Ejemplo

Considere

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

donde

$$\begin{aligned} H(s) &= D^{-1}(s)N(s) \\ D(s) &= \begin{bmatrix} 4+12s+9s^2+2s^3 & -s^3-2s^2 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} -s^2(s+2)+s^3 & -4(s+1)^3-s^2(-4-4s) \\ -s & -4-4s \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[4+12*s+9*s^2+2*s^3 -s^3-2*s^2; s^2+4*s+4 s+2];
>>N=[-s^2*(s+2)+s^3 -4*(s+1)^3-s^2*(-4-4*s); -s -4-4*s];
>>[Acco,Bcco,Ccco]=fccodad(N,D)

Acco =

   0,   1,   0,   0,   0
   0,   0,   1,   0,   0
   0,   0,   0,   1,   0
  -4, -12, -13,  -6,   0
  -4,  -4,  -1,   0,  -2

Bcco =

  -1,  -4
   4,  12
 -11, -32
  26,  80
   0,   0

Ccco =

 1, 0, 0, 0, 0
 0, 0, 0, 0, 1
```

Diagnósticos

La función “fccodad” regresa un mensaje de error

"Las_matrices_de_entrada_no_deben_estar_vacías"
"La_matriz_"D"_no_es_cuadrada"
"Incompatibilidad_de_matrices_de_entrada"
"Las_matrices_de_entrada_son_iguales"
"No_es_estrictamente_propia,_(invD*N)"

• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.

"Fila_en_"D"_con_grado_menor_que_uno"

• Alguna de las filas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.

"La_matriz_"D"_no_es_reducida_por_filas"

Ver También

mfd, fcomfd, fobmfd, fcdadmfd, fodadmfd, fcccdad

4.3.3 fcdadmfd.m (Forma Controlabilidad a partir de una MFD)

Propósito

Obtener la realización en forma controlabilidad a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD derecha; esto es $H(s) = N(s)D^{-1}(s)$.

Sintaxis

$$[A_{co}, B_{co}, C_{co}] = fcdadmfd(N, D, Variable)$$

donde las entradas son

$N = N(s) \in \mathbb{R}^{p\times m}[s]$
$D = D(s) \in \mathbb{R}^{m\times m}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_{co} \in \mathbb{R}^{n\times n}$
$B_{co} \in \mathbb{R}^{n\times m}$
$C_{co} \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización controlabilidad, que se obtiene a través de "fcdadmfd", es igual al grado del determinante de $D(s)$. Se requiere que $D(s)$ sea reducida por filas. Se pide también que ningún li, sea menor que uno, donde li, $i = 1, ..., p$ es el grado de la i-ésima fila de $D(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fcdadmfd"), respectivamente

$$\begin{aligned} H(s) &= C_{co}(SI - A_{co})^{-1}B_{co} && (4.5) \\ &= N(s)D(s)^{-1} && (4.6) \end{aligned}$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+2)^2(s+1)^2} & \frac{s}{(s+2)^2} \\ -\frac{s}{(s+2)^2} & -\frac{s}{(s+2)^2} \end{bmatrix}$$

donde

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) \\ D(s) &= \begin{bmatrix} 0 & -s^3-4s^2-5s-2 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} s & 0 \\ -s & s^2 \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[0 -s^3-4*s^2-5*s-2; (s+2)^2 s+2];
>>N=[s 0; -s s^2];
>>[Aco,Bco,Cco]=fcdadmfd(N,D)

Aco =

  0,  0, -2,  0,  0
  1,  0, -5,  0,  0
  0,  1, -4,  0,  0
  0,  0,  2,  0, -4
  0,  0,  1,  1, -4

Bco =

  1,  0
  0,  0
  0,  0
  0,  1
  0,  0

Cco =

   0,    0,    1,    1,  -4
  -1,    4,  -12,   -1,   4
```

Diagnósticos

La función "fcdadmfd" regresa un mensaje de error

“Las_matrices_de_entrada_no_deben_estar_vacías”
“La_matriz_“D”_no_es_cuadrada”
“Incompatibilidad_de_matrices_de_entrada”
“Las_matrices_de_entrada_son_iguales”
“No_es_estrictamente_propia,_(N*invD)”

• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.

“Fila_en_“D”_con_grado_menor_que_uno”

• Alguna de las filas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.

“La_matriz_“D”_no_es_reducida_por_filas”

Ver También

mfd, fcomfd, fobmfd, fodadmfd, fcccdad, fccodad

4.3.4 fcomfd.m (Forma Controlador a partir de una MFD)

Propósito

Obtener la realización en forma controlador a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD derecha; esto es $H(s) = N(s)D^{-1}(s)$.

Sintaxis

$$[A_c, B_c, C_c] = fcomfd(N, D, Variable)$$

donde las entradas son

$N = N(s) \in \mathbb{R}^{p\times m}[s]$
$D = D(s) \in \mathbb{R}^{m\times m}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_c \in \mathbb{R}^{n\times n}$
$B_c \in \mathbb{R}^{n\times m}$
$C_c \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización controlador, que se obtiene a través de "fcomfd", es igual al grado del determinante de $D(s)$. Se requiere que $D(s)$ sea reducida por columnas. Se pide también que ningún k_i sea menor que uno, donde k_i, $i = 1, ..., m$ es el grado de la i-ésima columna de $D(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fcomfd"), respectivamente

$$H(s) = C_c(SI - A_c)^{-1}B_c \tag{4.7}$$
$$= N(s)D^{-1}(s) \tag{4.8}$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+2)^2(s+1)^2} & \frac{s}{(s+2)^2} \\ -\frac{s}{(s+2)^2} & -\frac{s}{(s+2)^2} \end{bmatrix}$$

donde

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) \\ D(s) &= \begin{bmatrix} 0 & -s^3 - 4s^2 - 5s - 3 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} s & 0 \\ -s & -s^2 \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[0 -s^3-4*s^2-5*s-2; (s+2)^2 s+2];
>>N=[s 0; -s s^2];
>>[Ac,Bc,Cc]=fcomfd(N,D)

Ac =

 -4, -4,  0, -1, -2
  1,  0,  0,  0,  0
  0,  0, -4, -5, -2
  0,  0,  1,  0,  0
  0,  0,  0,  1,  0

Bc =

  0,  1
  0,  0
 -1,  0
  0,  0
  0,  0

Cc =

  1,  0,  0,  0,  0
 -1,  0,  1,  0,  0
```

Diagnósticos

La función "fcomfd" regresa un mensaje de error

"Las_matrices_de_entrada_no_deben_estar_vacías"
"La_matriz_"D"_no_es_cuadrada"
"Incompatibilidad_de_matrices_de_entrada"
"Las_matrices_de_entrada_son_iguales"
"No_es_estrictamente_propia,_(N*invD)"
• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.
"Columna_en_"D"_con_grado_menor_que_uno"
• Alguna de las columnas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.
"La_matriz_"D"_no_es_reducida_por_columnas"

Ver También

mfd, fobmfd, fcdadmfd, fodadmfd, fcccdad, fccodad

4.3.5 fobmfd.m (Forma Observador a partir de una MFD)

Propósito

Obtener la realización en forma observador a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD izquierda; esto es $H(s) = D_L^{-1}(s)N_L(s)$.

Sintaxis

$$[A_o, B_o, C_o] = fobmfd(N, D, Variable)$$

donde las entradas son

$N = N_L(s) \in \mathbb{R}^{p\times m}[s]$
$D = D_L(s) \in \mathbb{R}^{p\times p}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_o \in \mathbb{R}^{n\times n}$
$B_o \in \mathbb{R}^{n\times m}$
$C_o \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización observador, que se obtiene a través de "fobmfd", es igual al grado del determinante de $D_L(s)$. Se requiere que $D_L(s)$ sea reducida por filas. Se pide también que ningún l_i sea menor que uno, donde l_i, $i = 1, ..., p$ es el grado de la i-ésima fila de $D_L(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fobmfd"), respectivamente

$$\begin{aligned} H(s) &= C_o(SI - A_o)^{-1}B_o & (4.9)\\ &= D_L^{-1}(s)N_L(s) & (4.10)\end{aligned}$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+2)^2(s+1)^2} & \frac{s}{(s+2)^2} \\ -\frac{s}{(s+2)^2} & -\frac{s}{(s+2)^2} \end{bmatrix}$$

donde

$$\begin{aligned} H(s) &= D_L^{-1}(s)N_L(s) \\ D_L(s) &= \begin{bmatrix} (s+1)^2(s+2) & (s+2) \\ 0 & (s+2)^2 \end{bmatrix} \\ N_L(s) &= \begin{bmatrix} 0 & s^2 \\ -s & -s \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[(s+1)^2*(s+2) s+2; 0 (s+2)^2];
>>N=[0 s^2; -s -s];
>>[Ao,Bo,Co]=fobmfd(N,D)

Ao =

 -4,  1,  0,  0,  0
 -5,  0,  1, -1,  0
 -2,  0,  0, -2,  0
  0,  0,  0, -4,  1
  0,  0,  0, -4,  0

Bo =

  0,  1
  0,  0
  0,  0
 -1, -1
  0,  0

Co =

 1, 0, 0, 0, 0
 0, 0, 0, 1, 0
```

Diagnósticos

La función "fobmfd" regresa un mensaje de error

"Las_matrices_de_entrada_no_deben_estar_vacías"
"La_matriz_"D"_no_es_cuadrada"
"Incompatibilidad_de_matrices_de_entrada"
"Las_matrices_de_entrada_son_iguales"
"No_es_estrictamente_propia,_(invD*N)"

• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.

"Fila_en_"D"_con_grado_menor_que_uno"

• Alguna de las filas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.

"La_matriz_"D"_no_es_reducida_por_filas"

Ver También

mfd, fcomfd, fcdadmfd, fodadmfd, fcccdad, fccodad

4.3.6 fodadmfd.m (Forma Observabilidad a partir de una MFD)

Propósito

Obtener la realización en forma observabilidad a partir de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$, la cual es expresada como una MFD izquierda; esto es $H(s) = D_L^{-1}(s)N_L(s)$.

Sintaxis

$$[A_{ob}, B_{ob}, C_{ob}] = fodadmfd(N, D, Variable)$$

donde las entradas son

$N = N_L(s) \in \mathbb{R}^{p\times m}[s]$
$D = D_L(s) \in \mathbb{R}^{p\times p}[s]$
$Variable.-$ variable de trabajo

y las salidas

$A_{ob} \in \mathbb{R}^{n\times n}$
$B_{ob} \in \mathbb{R}^{n\times m}$
$C_{ob} \in \mathbb{R}^{p\times n}$

con 'n' la dimensión del estado.

Las matrices de entrada están en formato simbólico; las de salida están en formato numérico.

Descripción

El orden de la realización observabilidad, que se obtiene a través de "fodadmfd", es igual al grado del determinante de $D_L(s)$. Se requiere que $D_L(s)$ sea reducida por columnas. Se pide también que ningún k_i sea menor que uno, donde k_i, $i = 1, ..., m$ es el grado de la i-ésima columna de $D_L(s)$.

Las ecuaciones siguientes ilustran el significado de las matrices de salida y de entrada, (de "fodadmfd"), respectivamente

$$H(s) = C_{ob}(SI - A_{ob})^{-1}B_{ob} \quad (4.11)$$
$$= D_L(s)^{-1}N_L(s) \quad (4.12)$$

donde $H(s) \in \mathbb{R}^{p\times m}(s)$.

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+2)^2(s+1)^2} & \frac{s}{(s+2)^2} \\ -\frac{s}{(s+2)^2} & -\frac{s}{(s+2)^2} \end{bmatrix}$$

donde

$$\begin{aligned} H(s) &= D^{-1}(s)N(s) \\ D(s) &= \begin{bmatrix} (s+1)^2(s+2) & (s+2) \\ 0 & (s+2)^2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} 0 & s^2 \\ -s & -s \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>>syms s;
>>D=[(s+1)^2*(s+2) (s+2); 0 (s+2)^2];
>>N=[0 s^2; -s -s];
>>[Aob,Bob,Cob]=fodadmfd(N,D)

Aob =

  0,  1,  0,  0,  0
  0,  0,  1,  0,  0
 -2, -5, -4, -2, -1
  0,  0,  0,  0,  1
  0,  0,  0, -4, -4

Bob =

  0,  1
  0, -4
  1, 12
 -1, -1
  4,  4

Cob =

  1,  0,  0,  0,  0
  0,  0,  0,  1,  0
```

Diagnósticos

La función “fodadmfd” regresa un mensaje de error

"Las_matrices_de_entrada_no_deben_estar_vacías"
"La_matriz_"D"_no_es_cuadrada"
"Incompatibilidad_de_matrices_de_entrada"
"Las_matrices_de_entrada_son_iguales"
"No_es_estrictamente_propia,_(invD*N)"

• Alguna de las celdas de la matriz de transferencia, $H(s)$, no es estrictamente propia.

"Fila_en_"D"_con_grado_menor_que_uno"

• Alguna de las filas de D tiene grado menor que uno; si este es el caso se recomienda utilizar otra MFD de $H(s)$.

"La_matriz_"D"_no_es_reducida_por_columnas"

Ver También

mfd, fcomfd, fobmfd, fcdadmfd, fcccdad, fccodad

4.3.7 gcrdmfd.m (greatest common right divisor of the MFD), gcldmfd.m (greatest common left divisor of the MFD)

Propósito

Obtener de una MFD derecha (izquierda) el máximo común divisor derecho (izquierdo) y además una MFD derecha (izquierda) irreducible.

Sintaxis

$$\begin{aligned}[R, N_{Ri}, D_{Ri}] &= gcrdmfd(N_R, D_R, Variable)\\ [R, N_{Li}, D_{Li}] &= gcldmfd(N_L, D_L, Variable)\end{aligned}$$

donde las entradas son

para "gcrdmfd"	para "gcldmfd"
$N_R = N_R(s) \in \mathbb{R}^{p\times m}[s]$	$N_L = N_L(s) \in \mathbb{R}^{p\times m}[s]$
$D_R = D_R(s) \in \mathbb{R}^{m\times m}[s]$	$D_L = D_L(s) \in \mathbb{R}^{p\times p}[s]$

$Variable.$ – variable de trabajo

y las salidas

para "gcrdmfd"	para "gcldmfd"
$\{N_{Ri}(s) \in \mathbb{R}^{p\times m}[s]\} = N_{Ri}$	$\{N_{Li}(s) \in \mathbb{R}^{p\times m}[s]\} = N_{Li}$
$\{D_{Ri}(s) \in \mathbb{R}^{m\times m}[s]\} = D_{Ri}$	$\{D_{Li}(s) \in \mathbb{R}^{p\times p}[s]\} = D_{Li}$

$\{R(s) \in \mathbb{R}^{p\times m}[s]\} = R$ máximo común divisor derecho (izquierdo) de la MFD.

$\{N_{Ri}, D_{Ri}\}$ son matrices coprimas derechas.

$\{N_{Li}, D_{Li}\}$ son matrices coprimas izquierdas.

Las matrices N, D, Ni, Di y R están en formato simbólico.

Descripción

La función "gcrdmfd" ("gcldmfd") encuentra el máximo común divisor derecho (izquierdo), $R(s)$, de una MFD derecha (izquierda) y además arroja una MFD derecha (izquierda) la cual es irreducible. Lo anterior se ilustra en las siguientes ecuaciones

Para "gcrdmfd"

$$\begin{aligned}H(s) &= N_R(s)D_R^{-1}(s) &(4.13)\\ &= N_{Ri}(s)D_{Ri}^{-1}(s) &(4.14)\end{aligned}$$

donde

$$\begin{aligned}N_{Ri}(s) &= N_R(s)R(s)^{-1} &(4.15)\\ D_{Ri}(s) &= D_R(s)R(s)^{-1} &(4.16)\end{aligned}$$

Para "gcldmfd"

$$H(s) = D_L^{-1}(s)N_L(s) \tag{4.17}$$
$$= D_{Li}^{-1}(s)N_{Li}(s) \tag{4.18}$$

donde

$$N_{Li}(s) = R(s)^{-1}N_L(s) \tag{4.19}$$
$$D_{Li}(s) = R(s)^{-1}D_L(s) \tag{4.20}$$

Ejemplos

•Ejemplo 1.- Para "gcrdmfd" considere

$$\begin{aligned} H(s) &= N_R(s)D_R^{-1}(s) \\ &= \begin{bmatrix} s & s(s+1)^2 \\ -s(s+1)^2 & -s(s+1)^2 \end{bmatrix} \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ 0 & (s+1)^2(s+2)^2 \end{bmatrix}^{-1} \end{aligned}$$

Instrucciones en Matlab

```
>> syms s;
>> NR=[s s*(s+1)^2; -s*(s+1)^2 -s*(s+1)^2];
>> DR=[(s+1)^2*(s+2)^2 0; 0 (s+1)^2*(s+2)^2];
>> [R,NRi,DRi]=gcrdmfd(NR,DR)

R =

[                  1,        1+s^2+2*s]
[                  0, s^3+4*s^2+5*s+2]

NRi =

[             s,               0]
[ -s^3-2*s^2-s,             s^2]

DRi =

[ s^4+6*s^3+13*s^2+12*s+4,          -4*s^2-5*s-s^3-2]
[                       0,                       s+2]
```

•Ejemplo 2.- Para "gcldmfd" considere

$$\begin{aligned} H(s) &= D_L^{-1}(s)N_L(s) \\ &= \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ 0 & (s+2)^2 \end{bmatrix}^{-1} \begin{bmatrix} s & s(s+1)^2 \\ -s & -s \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>> syms s;
>> NL=[s s*(s+1)^2; -s -s];
>> DL=[(s+1)^2*(s+2)^2 0; 0 (s+2)^2];
>> [R,NLi,DLi]=gcldmfd(NL,DL)

R =

[   1,    0]
[  -1, s+2]

NLi =

[          s, s*(s+1)^2]
[          0,       s^2]

DLi =

[ (s+1)^2*(s+2)^2,                    0]
[    (s+2)*(s+1)^2,                 s+2]
```

Diagnósticos

La función "gcrdmfd" regresa un mensaje de error

> **"La_matrices_(N,D)_no_deben_estar_vacías."**
> **"La_matriz_"D"_no_es_cuadrada."**
> **"Las_dimensiones_de_las_matrices_de_entrada_son_incompatibles."**
> **"La_matriz_"D"_no_es_invertible."**
> El programa abortará automáticamente en caso de que alguna de las matrices, N y/o D, no sea simbólica.

Ver También

mpropsym, mfd, isrcopri

4.3.8 ispolnal.m (is polynomial)

Propósito

Determinar si una matriz, $X(s)$, es polinomial.

Sintaxis

$$[NY] = ispolnal(X)$$

donde las entradas son

$X = X(s)$

y las salidas

NY con valor de: 0 si la matriz no es polinomial
1 si la matriz es polinomial

Las matriz de entrada, X, debe estar en formato simbólico.

Descripción

Esta función, "ispolnal", evalúa cada elemento de una matriz, $X(s)$, y regresa un valor de cero si detecta que alguna celda tiene una función racional, en caso contrario regresa un valor de uno.

Ejemplo

Considere

$$X(s) = \begin{bmatrix} \frac{(s+1)^2(s+2)^2}{s} & 0 \\ 0 & (s+1)^2(s+2)^2 \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> X=[(s+1)^2*(s+2)^2/s 0; 0 (s+1)^2*(s+2)^2];
>> [NY]=ispolnal(X)

NY=

0
```

Diagnósticos

La función "ispolnal" no regresa mensajes de error.

Ver También

isrowred, mfd, isrcopri, gcrdmfd

4.3.9 isrcopri.m (is right coprime), islcopri.m (is left coprime)

Propósito

Determinar si las matrices de una MFD derecha (izquierda) son coprimas derechas (izquierdas).

Sintaxis

$$
\begin{aligned}
[NY] &= isrcopri(N_R, D_R, Variable) \\
[NY] &= islcopri(N_L, D_L, Variable)
\end{aligned}
$$

donde las entradas son

para "isrcopri"	para "islcopri"
$N_R = N_R(s) \in \mathbb{R}^{p\times m}[s]$	$N_L = N_L(s) \in \mathbb{R}^{p\times m}[s]$
$D_R = D_R(s) \in \mathbb{R}^{m\times m}[s]$	$D_L = D_L(s) \in \mathbb{R}^{p\times p}[s]$

$Variable.$ – variable de trabajo

y las salidas

NY con valor de:
0 si las matrices no son coprimas
1 si las matrices son coprimas

Las matrices de entrada N, D deben ser polinomiales y además deben estar en formato simbólico.

Descripción

Para la función, "isrcopri", se hace uso de la función "gcrdmfd"

$$N(s) = \bar{N}(s) * W(s) \tag{4.21}$$
$$D(s) = \bar{D}(s) * W(s) \tag{4.22}$$

donde $W(s)$ es un divisor derecho de $N(s)$ y $D(s)$. Si $W(s)$ es unimodular entonces $N(s)$ y $D(s)$ son coprimas derechas.

Para la función, "islcopri", se hace uso de la función "gcldmfd"

$$N(s) = W(s) * \bar{N}(s) \tag{4.23}$$
$$D(s) = W(s) * \bar{D}(s) \tag{4.24}$$

donde $W(s)$ es un divisor izquierdo de $N(s)$ y $D(s)$. Si $W(s)$ es unimodular entonces $N(s)$ y $D(s)$ son coprimas izquierdas.

Ejemplos

•Ejemplo 1.- Para "isrcopri" considere

$$N(s) = \begin{bmatrix} s & s(s+1)^2 \\ -s(s+1)^2 & -s(s+1)^2 \end{bmatrix}$$
$$D(s) = \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ 0 & (s+1)^2(s+2)^2 \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> N=[s s*(s+1)^2; -s*(s+1)^2 -s*(s+1)^2];
>> D=[(s+1)^2*(s+2)^2 0; 0 (s+1)^2*(s+2)^2];
>> isrcopri(N,D)

NY=

0
```

•Ejemplo 2.- Para "islcopri" considere

$$N(s) = \begin{bmatrix} s+1 & s+1 & 0 \\ (s+1)s^4 & 0 & 1 \end{bmatrix}$$
$$D(s) = \begin{bmatrix} s^2 & 0 \\ 0 & s^7 \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> N=[s+1, s+1, 0; (s+1)*s^4, 0, 1];
>> D=[s^2, 0; 0, s^7];
>> islcopri(N,D)

NY=

1
```

Diagnósticos

La función "isrcopri" regresa un mensaje de error

"La_matrices_(N,D)_no_deben_estar_vacías."
"La_matriz_"D"_no_es_cuadrada."
"Las_dimensiones_de_las_matrices_de_entrada_son_incompatibles."
"La_matriz_"D"_no_es_invertible."

El programa abortará automáticamente en caso de que alguna de las matrices, N y/o D, no sea simbólica.

Ver También

islcopri, mfd, gcrdmfd

4.3.10 isrowred.m (is row reduced), iscolred.m (is column reduced)

Propósito

Determinar si una matriz, polinomial, es reducida por filas (columnas), además regresa, también, la matriz de coeficientes líderes, $Q \in \mathbb{R}^{p\times m}$

Sintaxis

$$\begin{aligned} [NY, Q] &= isrowred(X, Variable) \\ [NY, Q] &= iscolred(X, Variable) \end{aligned}$$

donde las entradas son

$X = X(s) \in \mathbb{R}^{p\times m}[s]$
$Variable.-$ variable de trabajo

y las salidas

NY con valor de: 0 si la matriz no es reducida por filas (columnas)
1 si la matriz es reducida por filas (columnas)
$Q \in \mathbb{R}^{p\times m}$ es la matriz de coeficientes líderes.

Las matriz de entrada, 'X', debe ser polinomial y además debe estar en formato simbólico.

Descripción

Esta función, "isrowred", obtiene de una matriz, X, una matriz de coeficientes líderes, Q. Si Q es no singular entonces la matriz X será reducida por filas y la función regresará un valor de uno, en caso contrario regresará un valor de cero.

Ejemplos

•Ejemplo 1.- Para isrowred considere

$$X(s) = \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ 0 & (s+1)^2(s+2)^2 \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> X=[(s+1)^2*(s+2)^2 0; 0 (s+1)^2*(s+2)^2];
>> [NY,Q]=isrowred(X)

NY=
```

```
1

Q=

[1  0]
[0  1]
```

•Ejemplo 2.- Para iscolred considere

$$D(s) = \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ 0 & (s+1)^2(s+2)^2 \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> D=[(s+1)^2*(s+2)^2 0; 0 (s+1)^2*(s+2)^2];
>> [NY]=iscolred(D)

NY=

1

Q=

[1  0]
[0  1]
```

Diagnósticos

Las funciones "isrowred" no regresan mensajes de error.

Ver También

mfd, isrcopri, gcrdmfd

4.3.11 mfd.m (Matrix-Fraction Descriptions)

Propósito

Descomponer una matriz, $H(s) \in \mathbb{R}^{p\times m}(s)$ no necesariamente cuadrada, en una matriz estrictamente propia expresada como una factorización MFD derecha o izquierda mas una matriz residual, $Dft(s) \in \mathbb{R}^{p\times m}[s]$.

Sintaxis

$$[N, D, Dft] = mfd(H, tipo, Variable)$$

donde las entradas son

$$\begin{array}{l} H = H(s) \in \mathbb{R}^{p\times m}(s) \\ tipo = \left\{ \begin{array}{ll} 0 & \text{MFD izquierda} \\ 1 & \text{MFD derecha, por default} \end{array} \right\} \\ Variable.- \text{ variable de trabajo} \end{array}$$

y las salidas

$$\begin{array}{l} \text{si } tipo = 0 \left\{ \begin{array}{l} \{N_L(s) \in \mathbb{R}^{p\times m}[s]\} = N \\ \{D_L(s) \in \mathbb{R}^{p\times p}[s]\} = D \end{array}, \text{MFD izquierda} \right\} \\ \text{si } tipo = 1 \left\{ \begin{array}{l} \{N_R(s) \in \mathbb{R}^{p\times m}[s]\} = N \\ \{D_R(s) \in \mathbb{R}^{m\times m}[s]\} = D \end{array}, \text{MFD derecha} \right\} \\ \{Dft(s) \in \mathbb{R}^{p\times m}[s]\} = Dft \end{array}$$

Las matrices, H, N, D y Dft son de tipo simbólico y el modificador *tipo* es de tipo numérico.

Descripción

La función "mfd" realiza una descomposición de $H(s)$ como

$$H(s) = \bar{H}(s) + Dft(s) \tag{4.25}$$

En la ecuación anterior $Dft(s)$ será diferente de cero si $H(s)$ no es estrictamente propia. La matriz $\bar{H}(s)$ es una matriz estrictamente propia. La función "mfd" encuentra una factorización MFD, de $\bar{H}(s)$, derecha o izquierda según sea el valor del modificador *tipo* introducido. Si *tipo* es omitido "mfd" arrojará una factorización derecha.

Las ecuaciones siguientes

$$\bar{H}(s) = N_R(s)D_R^{-1}(s) \tag{4.26}$$

$$\bar{H}(s) = D_L^{-1}(s)N_L(s) \tag{4.27}$$

son unas factorizaciones, MFDs, derecha e izquierda respectivamente. La (Ec.4.26) es una factorización coprima derecha donde D_R es una matriz triangular superior no necesariamente reducida por columnas. La (Ec.4.27) es una factorización coprima izquierda donde $D_L(s)$ es una matriz triangular inferior no necesariamente reducida por filas.

Las ecuaciones siguientes ilustran el significado de las matrices de entrada y salida

$$H(s) = N_R(s)D_R^{-1}(s) + Dft(s) \quad (4.28)$$
$$H(s) = D_L^{-1}(s)N_L(s) + Dft(s) \quad (4.29)$$

Ejemplo

Ejemplo 1.- Para encontrar una MFD derecha considere

$$H(s) = \begin{bmatrix} \frac{s+1}{s^2} & \frac{s+1}{s^2} & s^2 \\ \frac{s+1}{s^3} & 0 & \frac{1}{s^7} \end{bmatrix}$$

con $H(s)$ expresada

$$H(s) = \bar{H}(s) + Dft(s)$$
$$H(s) = N_R(s)D_R^{-1}(s) + Dft(s)$$

de donde se encuentra una factorización derecha

$$\bar{H}(s) = N_R(s)D_R^{-1}(s)$$

Instrucciones en Matlab

```
>> syms s;
>> H=[(s+1)/s^2 (s+1)/s^2 s^2; (s+1)/s^3 0 1/s^7];
>> [N,D,Dft]=mfd(H)
''tipo''_por_default_igual_a_1,(derecha).

N =

[ s*(s+1),     s+1,    -1-s]
[     s+1,       0,      -1]


D =

[       s^3,          0, -1-s^2+s]
[         0,        s^2,      1-s]
[         0,          0,      s^4]


Dft =

[   0,    0, s^2]
[   0,    0,    0]
```

Ejemplo 2.- Para encontrar una MFD izquierda considere

$$H(s) = \begin{bmatrix} \frac{s+1}{s^2} & \frac{s+1}{s^2} & s^2 \\ \frac{s+1}{s^3} & 0 & \frac{1}{s^7} \end{bmatrix}$$

con $H(s)$ expresada

$$\begin{aligned} H(s) &= \bar{H}(s) + Dft(s) \\ H(s) &= D_L^{-1}(s)N_L(s) + Dft(s) \end{aligned}$$

donde se encuentra una factorización izquierda

$$\bar{H}(s) = D_L^{-1}(s)N_L(s)$$

Instrucciones en Matlab

```
>> syms s;
>> H=[(s+1)/s^2 (s+1)/s^2 s^2; (s+1)/s^3 0 1/s^7];
>> [N,D,Dft]=mfd(H,0)

N =

[        s+1,         s+1,           0]
[ (s+1)*s^4,           0,           1]

D =

[ s^2,   0]
[   0, s^7]

Dft =

[   0,   0, s^2]
[   0,   0,   0]
```

Diagnósticos

La función "mfd" regresa un mensaje de error

> **"La_matriz_de_entrada_no_debe_estar_vacía"**
> **"La_matriz_de_entrada_es_numérica"**
> **"'tipo'_debe_ser_numérico"**
> **"'tipo'_no_valido"**

Ver También

gcdrmfd, isrcopri, mpropsym, fcomfd

4.3.12 mfdiz2de.m (mfd izquierda a(2) derecha), mfdde2iz.m (mfd derecha a(2) izquierda)

Propósito

Obtener una MFD derecha (izquierda) a partir de una MFD izquierda (derecha).

Sintaxis

$$\begin{aligned}[N_R, D_R] &= mfdiz2de(N_L, D_L, Variable)\\ [N_L, D_L] &= mfdde2iz(N_R, D_R, Variable)\end{aligned}$$

donde las entradas son

para "mfdiz2de"	para "mfdde2iz"
$N_L = N_L(s) \in \mathbb{R}^{p\times m}[s]$	$N_R = N_R(s) \in \mathbb{R}^{p\times m}[s]$
$D_L = D_L(s) \in \mathbb{R}^{p\times p}[s]$	$D_R = D_R(s) \in \mathbb{R}^{m\times m}[s]$
$Variable.$ – variable de trabajo	

y las salidas

para "mfdiz2de"	para "mfdde2iz"
$\{N_R(s) \in \mathbb{R}^{p\times m}[s]\} = N_R$	$\{N_L(s) \in \mathbb{R}^{p\times m}[s]\} = N_L$
$\{D_R(s) \in \mathbb{R}^{m\times m}[s]\} = D_R$	$\{D_L(s) \in \mathbb{R}^{p\times p}[s]\} = D_L$

Tanto las matrices de entrada como de salida están en formato simbólico, N_L, N_R, D_L, D_R.

Descripción

Con éstas funciones se obtiene una factorización derecha (izquierda) a partir de una factorización izquierda (derecha) haciendo operaciones elementales sobre filas y columnas.

Una MFD derecha de una matriz, $H(s) \in \mathbb{R}^{p\times m}(s)$, esta dada por $N_R(s)$ y $D_R(s)$ tal que $H(s) = N_R(s)D_R^{-1}(s)$ y una MFD izquierda esta dada por $N_L(s)$ y $D_L(s)$ tal que $H(s) = D_L^{-1}(s)N(s)$.

Es importante puntualizar que éstas funciones no arrojan, necesariamente, la matriz denominador, $D_R(s)$ y $D_L(s)$, reducida por filas y/o columnas.

Ejemplos

•Ejemplo 1.- Para “mfdiz2de” considere

$$\begin{aligned} N_L(s) &= \begin{bmatrix} s & s(s+1)^2 \\ 0 & s^2 \end{bmatrix} \\ D_L(s) &= \begin{bmatrix} (s+1)^2(s+2)^2 & 0 \\ (s+1)^2(s+2) & s+2 \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>> syms s;
>> NL=[s s*(s+1)^2; 0 s^2];
>> DL=[(s+1)^2*(s+2)^2 0; (s+1)^2*(s+2) s+2]
>> [NR,DR]=mfdiz2de(NL,DL)

NR =

[          1/4*s,                        -1/16*(s+2)*s*(s+1)^2]
[-1/4*s*(s+1)^2, 1/16*s*(2+s^5+6*s^4+14*s^3+16*s^2+5*s)]

DR =

[1/4*(s+1)^2*(s+2)^2, -1/16*(s+2)*s*(s+1)^2*(s+3)*(s^2+3*s+4)]
[                  0,                                -1/2-1/4*s]
```

•Ejemplo 2.- Para “mfdde2iz” considere

$$\begin{aligned} N_R(s) &= \begin{bmatrix} s & 0 \\ -s(s+1)^2 & s^2 \end{bmatrix} \\ D_R(s) &= \begin{bmatrix} (s+1)^2(s+2)^2 & -(s+1)^2(s+2)^2 \\ 0 & s+2 \end{bmatrix} \end{aligned}$$

Instrucciones en Matlab

```
>> syms s;
>> NR=[s 0; -s*(s+1)^2 s^2];
>> DR=[(s+1)^2*(s+2)^2 -(s+1)^2*(s+2); 0 s+2]
>> [NL,DL]=mfdiz2de(NR,DR)

NL =

[                  1/4*s,                          1/4*s*(s+1)^2]
[1/16*(s+2)*s*(s+1)^2, 1/16*s*(2+s^5+6*s^4+14*s^3+16*s^2+5*s)]

DL =
```

```
[                    1/4*(s+1)^2*(s+2)^2,          0]
[1/16*(s+2)*s*(s+1)^2*(s+3)*(s^2+3*s+4), -1/2-1/4*s]
```

Diagnósticos

Las funciones "mfdiz2de" y "mfdde2iz" regresan un mensaje de error

> **"Las_matrices_de_entrada_no_deben_estar_vacías"**
> **"La_matriz_"D"_no_es_cuadrada"**
> **"Incompatibilidad_de_matrices_de_entrada"**
> **"Las_matrices_de_entrada_son_iguales"**
> **"Las_matrices_de_entrada_deben_ser_polinomiales"**

Ver También

isrowred, mfd, isrcopri, gcrdmfd

4.3.13 mpropsym.m (matrix proper symbolic)

Propósito

Descomponer una matriz, $H(s) \in \mathbb{R}^{p\times m}(s)$, en una matriz estrictamente propia, $Hp(s) \in \mathbb{R}^{p\times m}(s)$, y una matriz residual, $Dft(s) \in \mathbb{R}^{p\times m}[s]$, esto es $H(s) = Hp(s) + Dft(s)$.

Sintaxis

$$[Hp, Dft] = mpropsym(H, Variable)$$

donde las entradas son

$H = H(s) \in \mathbb{R}^{p\times m}(s)$
$Variable.-$ variable de trabajo

y las salidas

$\{Hp(s) \in \mathbb{R}^{p\times m}(s)\} = Hp$.-matriz estrictamente propia
$\{Dft(s) \in \mathbb{R}^{p\times m}[s]\} = Dft$.-matriz residual

Las matrices H, Hp y Dft están en formato simbólico.

Descripción

La función "mpropsym" realiza la descomposición de una matriz, $H(s)$, como sigue

$$H(s) = Hp(s) + Dft(s) \tag{4.30}$$

donde $Hp(s)$ es una matriz estrictamente propia. Definimos $H(s)$ como

$$H(s) = \begin{bmatrix} h_{11} & h_{12} & & h_{1n} \\ h_{21} & & & \\ & & & \\ & & & \\ h_{m1} & & & h_{mn} \end{bmatrix}. \tag{4.31}$$

entonces $Hp(s)$ y $Dft(s)$ se forman como

$$Hp(s)_{i,j} = \left\{ \frac{residuo\left(\frac{num(h_{i,j})}{denom(h_{i,j})}\right)}{denom\,(h_{i,j})}, i = 1, ..., m; j = 1, ..., n \right\} \tag{4.32}$$

$$Dft(s)_{i,j} = \left\{ cociente\left(\frac{num\,(h_{i,j})}{denom\,(h_{i,j})}\right), i = 1, ..., m; j = 1, ..., n \right\} \tag{4.33}$$

Ejemplo

Considere

$$H(s) = \begin{bmatrix} s & \frac{s}{s+1} \\ \frac{s^2}{s-1} & \frac{1}{s} \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> H=[s, s/(s+1); s^2/(s-1) 1/s];
>> [Hp,Dft]=mpropsym(H)

Hp =

[          0, -1/(s+1)]
[  1/(s-1),         1/s]

Dft =

[   s,    1]
[ s+1,    0]
```

Diagnósticos

La función "mpropsym" regresa un mensaje de error

"La_matriz_de_entrada_no_debe_estar_vacía"
"La_entrada_debe_ser_de_tipo_simbólica"

Ver También

gcrdmfd, rlcmsym, clcmsym

4.3.14 pzmimo.m (poles and zeros in mimo)

Propósito

Obtiene los polos y ceros de un sistema MIMO.

Sintaxis

$$[polos, ctransm, cinvar, cdesent, cdessal, cinf] = pzmimo(A, B, C, D)$$

donde las entradas son

$\{A, B, C, D\}$ realización en variables de estado

y las salidas

polos.- polos del sistema
ctransm.- ceros de transmisión
cinvar.- ceros invariantes
cdesent.- ceros de desacoplamiento de entrada
cdesal.- ceros de desacoplamiento de salida
cinf.- ceros al infinito

Tanto las matrices de entrada como los vectores de salida están en formato numérico.

Descripción

Existen dos clases más de ceros no contemplados por esta función, los cuales son: los ceros de desacoplamiento entrada-salida y los ceros del sistema, donde

$$\begin{aligned} cdesentsal &= \text{ceros de desacoplamiento entrada-salida} \\ cdesentsal &= cdessal \cap cdesent \\ \text{ceros del sistema} &= ctransm + cdessal + cdesent - cdesentsal \end{aligned}$$

Ejemplo

Considere

$$\begin{aligned} sistema &\equiv \{A, B, C\} \\ &\equiv \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^T \right\} \end{aligned}$$

Instrucciones en Matlab

```
>> a=[1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 3 0 0 0;
     0 0 0 -4 0 0; 0 0 0 0 -1 0; 0 0 0 0 0 3];
>> b=[0 -1; -1 0; 1 -1; 0 0; 0 1; -1 -1];
>> c=[1 0 0 1 0 0; 0 1 0 1 0 1; 0 0 1 0 0 1];
>> [polos,ctransm,cinvar,cdesent,cdessal,cinf]=pzmimo(a,b,c)

polos =

   3.0000 + 0.0000i
   3.0000 - 0.0000i
   1.0000 + 0.0000i
   1.0000 - 0.0000i

ctransm =

     2

cinvar =

     2
    -1

cdesent =

    -4

cdessal =

    -1

cinf =

     1
     0
```

Diagnósticos

La función "mpropsym" regresa un mensaje de error

> **"La␣matrices␣de␣entrada,␣(A,␣B,␣C),␣no␣deben␣estar␣vacías"**
> **"La␣matriz␣"D"␣no␣debe␣estar␣vacía"**
> **"Las␣matriz␣"A"␣no␣es␣cuadrada"**
> **"Las␣dimensiones␣de␣"A"␣y␣"B"␣no␣son␣compatibles"**
> **"Las␣dimensiones␣de␣"A"␣y␣"C"␣no␣son␣compatibles"**
> **"Las␣dimensiones␣de␣"D"␣no␣son␣compatibles"**

"Las_matrices_de_entrada,_(A,_B,_C),_no tienen_formato_numérico"
"La_matriz_"D"_debe_tener_formato_numérico"

Ver También

mfd, isrcopli, islcopri

4.3.15 smithmac.m (Smith-MacMillan)

Propósito

Obtener la matriz de Smith MacMillan, $M(s) \in \mathbb{R}^{p\times m}(s)$, de una matriz de transferencia, $H(s) \in \mathbb{R}^{p\times m}(s)$.

Sintaxis

$$[M] = smithmac(H)$$

donde las entradas son

$H = H(s) \in \mathbb{R}^{p\times m}(s)$

y las salidas

$M = M(s) \in \mathbb{R}^{p\times m}(s)$ que es la matriz de Smith MacMillan

Tanto la matriz de entrada, H, como la matriz de salida, M, están en formato simbólico.

Descripción

La función, "smithmac", obtiene por una matriz de transferencia, $H(s)$, la matriz de Smith MacMillan, $M(s)$.

$$H(s) = \frac{N(s)}{d(s)} \tag{4.34}$$

$$N(s) = U(s)S(s)V(s) \tag{4.35}$$

$$M(s) = \frac{S(s)}{d(s)} \tag{4.36}$$

donde $d(s)$ es el mínimo común múltiplo de los denominadores de $H(s)$.

El procedimiento de cálculo es: a partir de una matriz de transferencia se obtiene una matriz numerador, $N(s) \in \mathbb{R}^{p\times m}[s]$ (Ec.4.34); enseguida se obtiene su forma de Smith, $S(s)$ (Ec.4.35). $U(s)$ y $V(s)$ son matrices unimodulares. Finalmente la matriz de Smith MacMillan es obtenida al dividir la matriz de Smith, $S(s)$, entre $d(s)$ (Ec.4.36). La matriz $H(s)$ queda expresada como

$$H(s) = U(s)M(s)V(S) \tag{4.37}$$

Nota 71 *Las matrices $U(s)$ y $V(s)$ han sido omitidas por la función "smithmac"*

Ejemplo

Considere

$$H(s) = \begin{bmatrix} \frac{s}{(s+1)^2(s+2)^2} & \frac{s}{(s+2)^2} \\ \frac{-s}{(s+2)^2} & \frac{-s}{(s+2)^2} \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> H=[s/(s+1)^2/(s+2)^2 s/(s+2)^2; -s/(s+2)^2 -s/(s+2)^2];
>> M=smithmac(H)

M =

[ s/(s+1)^2/(s+2)^2,                0]
[                 0,         s^2/(s+2)]
```

Diagnósticos

La función "smithmac" regresa un mensaje de error

"La_matriz_de_entrada_no_debe_estar_vacía"

Ver También

mfd, isrcopri

4.3.16 sys2tm.m (system to transfer matrix)

Propósito

Encontrar la matriz de transferencia a partir de una realización en variables de estado

Sintaxis

$$[MT] = sys2tm(A, B, C, D)$$

donde
$\{A, B, C, D\}$ es la realización es espacio de estados
MT es la matriz de transferencia
Las matrices de entrada,$\{A, B, C, D\}$, deben estar en formato numérico.

Ejemplo

Considere

$$\begin{aligned} sistema &\equiv \{A, B, C\} \\ &\equiv \left\{ \begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ 1 & 0 & -5 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 \\ 0 & 0 & 2 & 0 & -4 \\ 0 & 0 & 1 & 1 & -4 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 0 & 4 \\ 1 & -12 \\ 1 & -1 \\ -4 & 4 \end{bmatrix}^T \right\} \end{aligned}$$

Instrucciones en Matlab

```
>> syms s;
>> A=[0 0 -2 0 0; 1 0 -5 0 0; 0 1 -4 0 0; 0 0 2 0 -4; 0 0 1 1 -4];
>> B=[1 0; 0 0; 0 0; 0 1; 0 0];
>> C=[0 0 1 1 -4; -1 4 -12 -1 4];
>> FT=sys2tm(A,B,C)

FT =

[ s/(s+2)^2/(s+1)^2,            s/(s+2)^2]
[          -s/(s+2)^2,          -s/(s+2)^2]
```

Diagnósticos

La función "smithmac" regresa un mensaje de error

"La_matrices_de_entrada,_(A,_B,_C),_no_deben_estar_vacías"
"La_matriz_"D"_no_debe_estar_vacía"
"Las_matriz_"A"_no_es_cuadrada"

"Las_dimensiones_de_"A"_y_"B"_no_son_compatibles"
"Las_dimensiones_de_"A"_y_"C"_no_son_compatibles"
"Las_dimensiones_de_"D"_no_son_compatibles"
"La_matrices_de_entrada,_(A,_B,_C),_no_son_numéricas"
"La_matriz_"D"_deben_tener_formato_numérico"

Ver también

mfd, fcomfd, fobmfd, fcdadmfd, fodadmfd, fcccdad, fccodad

4.4 Otras Funciones

Estas funciones, son auxiliares y, se encuentran integradas dentro de la programación de los algoritmos anteriores; por lo tanto, se ha considerado pertinente poner de manifiesto su existencia con el objeto de que los listados de los programas seán explícitos. Estas funciones entrañan algunos tips de programación y/o manejo de datos que podrán servir al usuario.

4.4.1 eyesym.m (matriz identidad simbólica)

Propósito

Construir la matriz identidad en formato simbólico

Sintaxis

$$[ident] = eyesym(renh, colh)$$

Descripción

$eyesym(n)$ es la matriz identidad de n-por-n.

$eyesym(m,n)$ o $eyesym([m,n])$ es una matriz de $m \times n$ con unos sobre la diagonal y ceros en el resto de las celdas.

$eyesym(size(A))$ es de la misma dimensión de A.

La salida es de tipo simbólico y las entradas de tipo numérico.

Ejemplo

Instrucciones en Matlab

```
>> (ident)=eyesym(3,4)

ident =

[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
```

4.4.2 mn2ms.m (matriz numérica a(2) matriz simbólica)

Propósito

Convertir una matriz numérica y formato numérico en una matriz numérica y formato simbólico.

Sintaxis

$$[ms] = mn2ms(mn)$$

mn es una matriz numérica en formato numérico
ms es una matriz numérica en formato simbólico

Descripción

Esta es una instrucción muy útil en operaciones entre elementos numéricos y simbólicos.

Ejemplo

Instrucciones en Matlab

```
>> Anumeric=[1 3 2; 0 0 1; 2 3 3]

Anumeric =

1  3  2
0  0  1
2  3  3

>> Asymbolic=mn2ms(A)

Asymbolic =

[1  3  2]
[0  0  1]
[2  3  3]
```

4.4.3 rlcmsym.m (row lcm symbolic), clcmsym.m (column lcm symbolic)

Propósito

Obtener el mínimo común múltiplo de los denominadores por filas (columnas) de una matriz, $H(s) \in \mathbb{R}^{p \times m}(s)$.

Sintaxis

$$
\begin{aligned}
[V] &= rlcmsym(H, Variable) \ \{\text{por filas}\} \\
[V] &= clcmsym(H, Variable) \ \{\text{por columnas}\}
\end{aligned}
$$

donde las entradas son

$H = H(s) \in \mathbb{R}^{p \times m}(s)$
$Variable.-$ variable de trabajo

y las salidas

$\{V(s) \in \mathbb{R}^{p}[s]\} = V$

La matriz H y el vector V están en formato simbólico.

Descripción

Sea la $H(s)$ una matriz y $V(s)$ un vector con el mínimo comúm múltiplo de los denominadores por filas (columnas) de $H(s)$; esto es si se define $H(s)$ como sigue

$$
H(s) = \begin{bmatrix} h_{11} & h_{12} & & h_{1n} \\ h_{21} & & & \\ . & & & \\ & & & \\ h_{m1} & & & h_{mn} \end{bmatrix} \tag{4.38}
$$

la función "rlcmsym" regresa un vector como

$$
\begin{aligned}
V(s) &= \begin{bmatrix} v_{11} & v_{21} & & v_{m1} \end{bmatrix}' && (4.39) \\
v_{i1} &= \{lcm\,(h_{ij})\,, i = 1, ..., m; j = 1, ..., n\} && (4.40)
\end{aligned}
$$

para "clcmsym" se regresa un vector como

$$
\begin{aligned}
V(s) &= \begin{bmatrix} v_{11} & v_{12} & & v_{1n} \end{bmatrix} && (4.41) \\
v_{1j} &= \{lcm\,(h_{ij})\,, i = 1, ..., m; j = 1, ..., n\} && (4.42)
\end{aligned}
$$

Ejemplos

•Ejemplo 1.- Para "rlcmsym" considere

$$H(s) = \begin{bmatrix} s & \frac{s}{s+1} \\ \frac{s^2}{s-1} & \frac{1}{s} \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> H=[s, s/(s+1); s^2/(s-1) 1/s];
>> [V]=rlcmsym(H)

V =

[    s+1]
[ s^2-s]
```

•Ejemplo 2.- Para "clcmsym" considere

$$H(s) = \begin{bmatrix} s & \frac{s}{s+1} \\ \frac{s^2}{s-1} & \frac{1}{s} \end{bmatrix}$$

Instrucciones en Matlab

```
>> syms s;
>> H=[s, s/(s+1); s^2/(s-1) 1/s];
>> [V]=clcmsym(H)

V =

[    s-1]
[ s^2+s]
```

4.4.4 sym2n.m (polinomio simbólico a(2) polinomio numérico)

Propósito

Convierte un polinomio simbólico en un polinomio numérico.

Sintaxis

$$[polix] = sym2n(P)$$

donde
$P = P(s)$ es el polinomio simbólico
polix es el vector del polinomio transformado (numérico).
El polinomio de entrada, "P", debe estar en formato simbólico. Es necesario que "P" tenga un grado mayor que cero.

Ejemplo

Considere

$$psym = 3s^3 + 4s^2 + 1$$

Instrucciones en Matlab

```
>> syms s;
>> psym=[3*s^3+4*s^2+1];

psym =

s*s^3+4*s^2+1

>> pnum=sym2n(psym)

pnum =

3  4  0  1
```

Capítulo 5

Discusión con otros trabajos y aplicaciones

5.1 Discusión con "Polynomial Toolbox"

Actualmente existe un equipo de trabajo integrado por diferentes investigadores entre los que se encuentran: Dr. Huibert Kwakernaak (Holanda), Dr. Rens C.W. (Holanda) y el Dr. Michael Sebek (República Checa). Este equipo de trabajo, dirigido por el Dr. Huibert Kwakernaak, se esta dado a la tarea de desarrollar herramientas para aplicarlas en polinomios, matrices polinomiales, control y procesamiento de señales. Las herramientas generadas por el Dr. Huibert Kwakernaak conforman la caja de herramientas **"Polynomial Toolbox"**, y ésta se habilita para ser utilizada bajo la plataforma de Matlab 5.x. Algunos de los algoritmos presentados, anteriormente, son similares a algunas de las herramientas de "Polynomial Toolbox". La similitud se da en las siguientes funciones:

Esta tesis Cálculo simbólico	**"Polynomial Toolbox"** Cálculo numérico	**Objetivo**
$fcomfd$	$rmf2ss$	Obtener la forma controlador
$fobmfd$	$lmf2ss$	Obtener la forma observador
$gcldmfd$	gld	Obtener el gcld de una MFD
$gcrdmfd$	grd	Obtener el gcrd de una MFD
$mfdde2iz$	rl	Pasar una MFD der. a una izq.
$mfdiz2de$	lr	Pasar una MFD izq. a una der.

A continuación se muestran un ejemplo donde primeramente se utiliza el **"Polynomial Toolbox"** y en seguida las herramientas presentadas en está tesis.

Ejemplo: Obtener la realización en espacio de estado en forma controlador de:

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) \\ D(s) &= \begin{bmatrix} 0 & -s^3-4s^2-5s-3 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} s & 0 \\ -s & -s^2 \end{bmatrix} \end{aligned}$$

a).- Utilizando "Polynomial Toolbox", b).- Utilizando las herramientas de esta tesis.

Solución:
a).- Utilizando "Polynomial Toolbox"
Las instrucciones en Matlab utilizando herramientas de "Polynomial Toolbox" son:

```
>> syms s;
>> D=[0 -2-5*s-4*s^2-s^3; 4+4*s+s^2 2+s];
>> N=[s 0; -s s^2];
```

% PASAR UNA MATRIZ SIMBOLICA A FORMATO NUMERICO

```
>> Dp=sym2p(D);
>> Np=sym2p(N);
>> [A,B,C,E]=rmf2ss(Np,Dp)

A =
 -4 -4  0 -1 -2
  1  0  0  0  0
  0  0 -4 -5 -2
  0  0  1  0  0
  0  0  0  1  0

B =
  0  1
  0  0
 -1  0
  0  0
  0  0

C =
  1  0  0  0  0
 -1  0  1  0  0

E =
  0  0
  0  0
```

b).- Utilizando las herramientas de esta tesis
Las instrucciones en Matlab utilizando herramientas de esta tesis son:

```
>> syms s;
>> D=[0 -2-5*s-4*s^2-s^3; 4+4*s+s^2 2+s];
>> N=[s 0; -s s^2];
>> [A,B,C]=fcomfd(N,D)
```

```
A =
[ -4, -4,  0, -1, -2]
[  1,  0,  0,  0,  0]
[  0,  0, -4, -5, -2]
[  0,  0,  1,  0,  0]
[  0,  0,  0,  1,  0]

B =

[  0,  1]
[  0,  0]
[ -1,  0]
[  0,  0]
[  0,  0]

C =

[  1,  0,  0,  0,  0]
[ -1,  0,  1,  0,  0]
```

5.1.1 ¿Qué es lo que hace la diferencia?

Polynomial Toolbox es un conjunto de herramientas cuyos algoritmos realizan cálculos con matrices polinomiales en forma numérica y no en forma simbólica como en este trabajo. El hecho de tener cálculos en forma simbólica da mas flexibilidad de resolver ciertos problemas. Para ilustrar lo anterior se muestra el siguiente ejemplo.

Ejemplo: Obtener la realización en espacio de estado en forma controlador de:

$$\begin{aligned} H(s) &= N(s)D^{-1}(s) \\ D(s) &= \begin{bmatrix} 0 & -s^3 - ks^2 - 5s - 3 \\ (s+2)^2 & s+2 \end{bmatrix} \\ N(s) &= \begin{bmatrix} s & 0 \\ -s & -s^2 \end{bmatrix} \end{aligned}$$

a) Utilizando "Polynomial Toolbox", b) Utilizando las herramientas de esta tesis.

Solución:

a).- Utilizando "Polynomial Toolbox"

No se puede resolver debido a que el ejemplo incluye una variable "k" en la matriz D.

b).- Utilizando las herramientas de esta tesis

Las instrucciones en Matlab utilizando herramientas de esta tesis son:

```
>> syms s k;
>> D=[0 -2-5*s-k*s^2-s^3; 4+4*s+s^2 2+s];
>> N=[s 0; -s s^2];
>> [A,B,C]=fcomfd(N,D,s)

A =

[ -4, -4,  0, -1, -2]
[  1,  0,  0,  0,  0]
[  0,  0, -k, -5, -2]
[  0,  0,  1,  0,  0]
[  0,  0,  0,  1,  0]

B =

[  0,  1]
[  0,  0]
[ -1,  0]
[  0,  0]
[  0,  0]

C =

[  1,  0,  0,  0,  0]
[ -1,  0,  1,  0,  0]
```

5.2 Aplicaciones

Para mostrar la utilidad de las herramientas presentadas en esta tesis, se presentan algunos ejemplos donde haciendo uso de éstas, el proceso de diseño y análisis de sistemas de control es más rápido y sencillo.

5.2.1 Problema de seguimiento de trayectoria

Primeramente se considera que se tiene un modelo como la configuración de la Figura 5.1 donde:

W es el modelo generador de la consigna w
w es una señal de consigna
r es la señal de referencia
C_1 es un precompensador
C_2 es el controlador del sistema en lazo cerrado
P es la planta

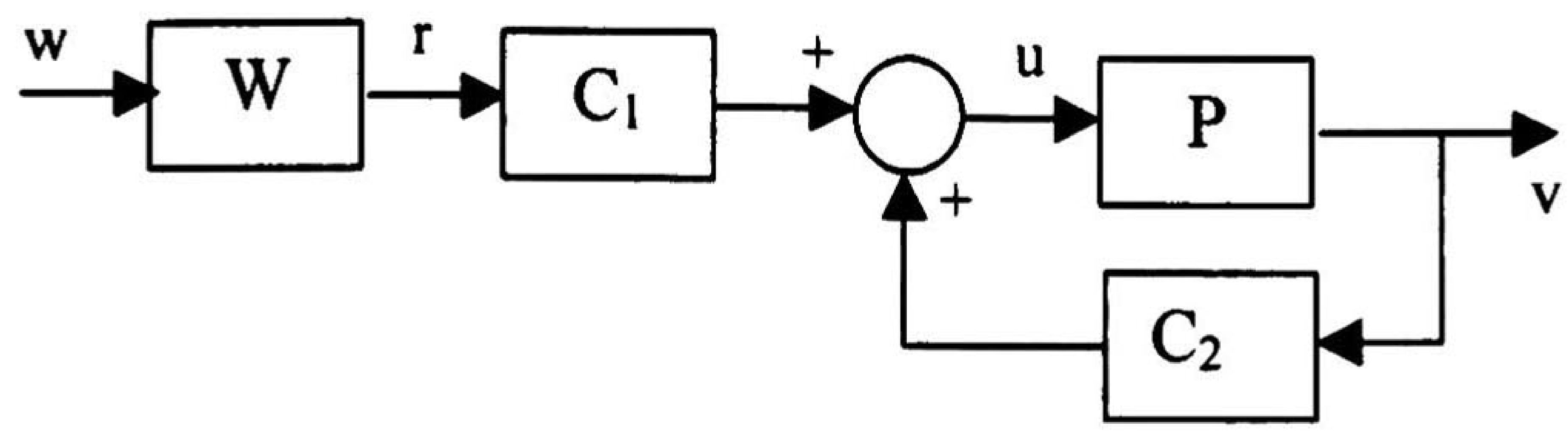

Figura 5.1: Esquema del problema de seguimiento

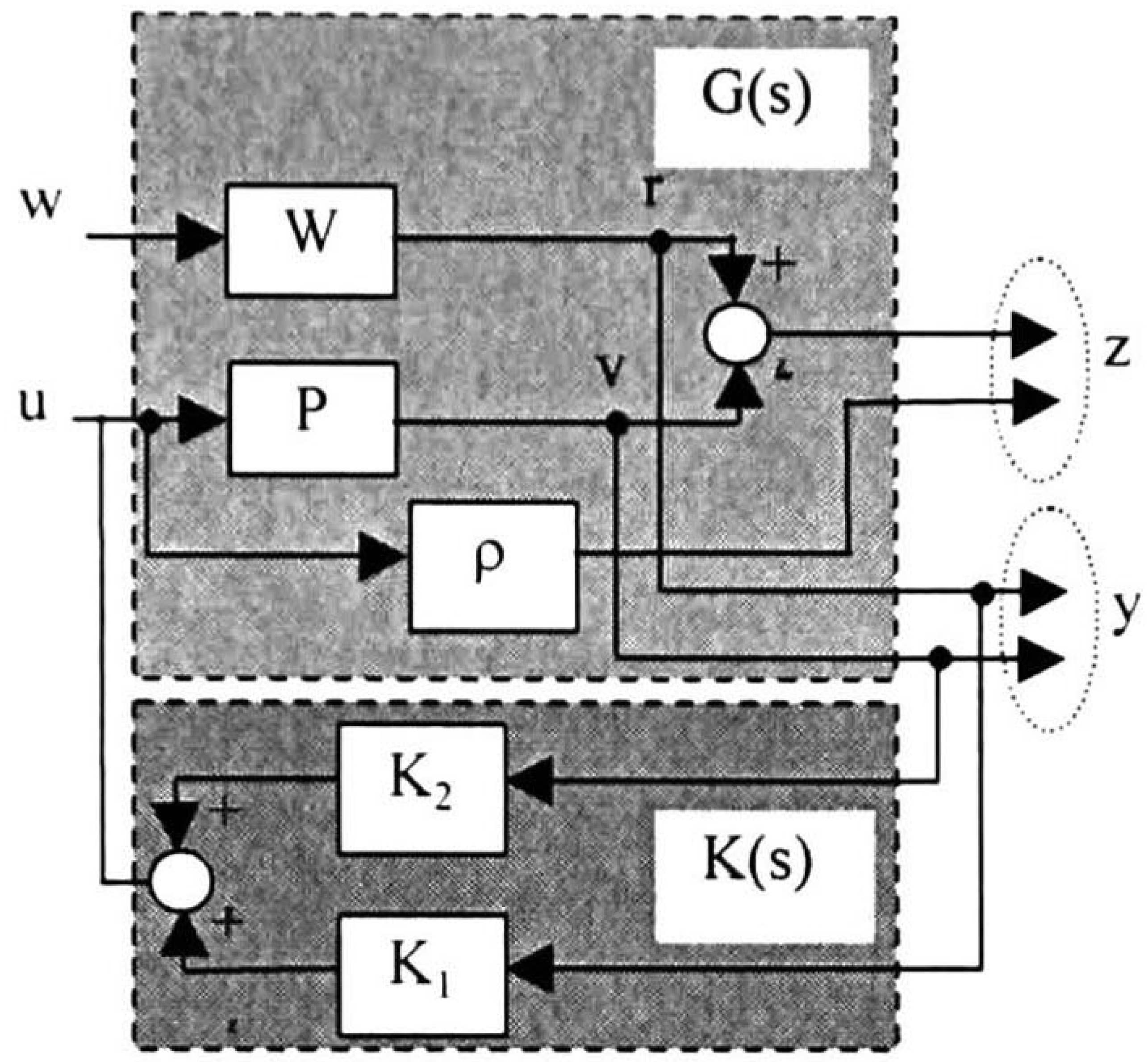

Figura 5.2: Esquema del problema estándar derivado

v es la señal de salida
u es la señal de control

El problema de seguimiento de trayectoria consiste en diseñar un controlador K formado por los bloques K_1 y K_2 en la Figura 5.1 que permita que la señal de salida v de la planta P se comporte como la señal de referencia r, la cual es generada por el sistema W bajo la señal de control w. Un objetivo de diseño es encontrar $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ tal que minimice la energía del error de seguimiento $(r - v)$ o aún más que también minimice la energía de control. Proponiendo, como variable a controlar

$$z = \begin{bmatrix} r - v \\ \rho u \end{bmatrix} \tag{5.1}$$

(donde ρ es un peso sobre u) se tiene que un criterio a minimizar puede ser expresado como

$$J = \|z\|_2 = \left\| \begin{bmatrix} r - v \\ \rho u \end{bmatrix} \right\|_2 = \left(\|r - v\|_2^2 + \|\rho u\|_2^2 \right)^{\frac{1}{2}} \tag{5.2}$$

Habiendo definido la variable a controlar y el criterio, el esquema del problema normal se obtiene redibujando la Figura 5.1 donde se incluye la definición de z, (Figura 5.2).

De aquí se obtiene

$$y = \begin{bmatrix} r \\ v \end{bmatrix}, \; w = w, \; u = u \tag{5.3}$$

por lo que

$$\begin{bmatrix} r-v \\ \rho u \\ \cdots \\ r \\ v \end{bmatrix} = \begin{bmatrix} W & \vdots & -P \\ 0 & \vdots & \rho I \\ \cdots & \cdots & \cdots \\ W & \vdots & 0 \\ 0 & \vdots & P \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix} \tag{5.4}$$

$$= \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{5.5}$$

si,

$$\begin{aligned} P(s) &= \frac{s-1}{s(s-2)} \\ W(s) &= \frac{s+1}{10s+1} \\ \rho &= 1 \end{aligned}$$

de (Ec.5.5) se tiene

$$\begin{aligned} G(s) &= \begin{bmatrix} G(s)_{11} & G(s)_{12} \\ G(s)_{21} & G(s)_{22} \end{bmatrix} \\ &= \begin{bmatrix} \frac{s+1}{10s+1} & \vdots & \frac{s-1}{s(s-2)} \\ 0 & & 1 \\ \cdots & \cdots & \cdots \\ \frac{s+1}{10s+1} & \vdots & 0 \\ 0 & & \frac{s-1}{s(s+2)} \end{bmatrix} \end{aligned}$$

El problema que se presenta ahora es obtener una realización mínima en variables de estado, $G(s) = [A, B, C, E]$

Instrucciones de Matlab para introducir la matriz $G(s)$

```
>> syms s

>> G=[(s+1)/(10*s+1) (s-1)/s/(s-2); 0 1; (s+1)/(10*s+1) 0; 0 (s-1)/s/(s-2)]

G =

[ (s+1)/(10*s+1), (s-1)/s/(s-2)]

[ 0 , 1 ]

[ (s+1)/(10*s+1), 0 ]

[ 0 , (s-1)/s/(s-2)]
```

Se aplica la función "mfd" (de esta tesis) para obtener una MFD derecha y la matriz E del sistema

```
>> [N,D,E]=mfd(G,1)

N -

[ 9, s-1]

[ 0, 0 ]
```

```
[ 9, 0 ]

[ 0, s-1]

D =

[ 100*s+10, 0 ]

[ 0 , s*(s-2)]

E =

[ 1/10, 0]

[ 0, 1]

[ 1/10, 0]

[ 0, 0]
```

Se aplica la función "fcomfd" (de esta tesis) para obtener una realización en forma controlador

```
>> [A,B,C]=fcomfd(N,D)

A =

-0.1000 0 0

0 2.0000 0

0 1.0000 0

B =

0.0100 0

0 1.0000

0 0

C =

9 1 -1

0 0 0

9 0 0

0 1 -1
```

pero

$$G(s) = C(sI - A)^{-1}B + E \tag{5.6}$$

$$= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + E \tag{5.7}$$

entonces se tiene

$$
\begin{aligned}
A &= \begin{bmatrix} -0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
B_1 &= \begin{bmatrix} 0.01 \\ 0 \\ 0 \end{bmatrix} \\
B_2 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\
C_1 &= \begin{bmatrix} 9 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \\
C_2 &= \begin{bmatrix} 9 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix}
\end{aligned}
$$

Lo que resta, ahora, es calcular el controlador del algoritmo presentado en [**7**]; el controlador que minimiza la (Ec.5.2) está dado por

$$K = Y_2 X_2^{-1}$$

donde

$$
\begin{aligned}
Y_2(s) &= \left[\begin{array}{c|c} A_F & -H \\ \hline F & 0 \end{array}\right] \\
X_2(s) &= \left[\begin{array}{c|c} A_F & -H \\ \hline C_2 & I \end{array}\right] \\
A_F &= A + B_2 F \\
A_H &= A + H C_2
\end{aligned}
$$

F es cualquier matriz que estabilize a A_F
H es cualquier matriz que estabilize a A_H

5.2.2 Problema de una columna de destilación

En el problema presentado a continuación se calcula un controlador LQ. En el método de calculo del controlador se muestra la utilidad de las herramientas presentadas en esta tesis. Se toma el modelo de una columna de destilación de [30]. El modelo esta dado por las siguientes ecuaciones

$$G_D(s, k_1, k_2, \tau_1, \tau_2) = \frac{1}{75s+1} \begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix} \begin{bmatrix} k_1 e^{-\tau_1 s} & 0 \\ 0 & k_2 e^{-\tau_2 s} \end{bmatrix}$$

donde $k_1 = k_2 = 1.0$ y $\tau_1 = \tau_2 = 0.5$.

Utilizando una aproximación de Padé de primer orden entonces

$$\begin{aligned} G_D(s) &= \frac{1}{75s+1}\begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}\begin{bmatrix} \frac{4-s}{4+s} & 0 \\ 0 & \frac{4-s}{4+s} \end{bmatrix} \\ &= \begin{bmatrix} -.878\frac{-4+s}{(75s+1)(4+s)} & .864\frac{-4+s}{(75s+1)(4+s)} \\ -1.082\frac{-4+s}{(75s+1)(4+s)} & 1.096\frac{-4+s}{(75s+1)(4+s)} \end{bmatrix} \end{aligned}$$

Instrucciones de Matlab para introducir $G_D(s)$

```
>> G=(1/(75*s+1))*[0.878 -0.864; 1.082 -1.096]*[(-s+4)/(s+4) 0; 0 (-s+4)/(s+4)]
G=
[ 439/500/(75*s+1)*(-s+4)/(s+4), -108/125/(75*s+1)*(-s+4)/(s+4)]
[ 541/500/(75*s+1)*(-s+4)/(s+4), -137/125/(75*s+1)*(-s+4)/(s+4)]
```

Se aplica la función "mfd" (de esta tesis) para obtener una MFD derecha

```
>>[N,D,E]=mfd(G)
"tipo"_por_default_igual_a_1,(derecha).
N=
[ -439*s+1756, 108*s-432]
[ -541*s+2164, 137*s-548]
D=
[ 500*(75*s+1)*(s+4), 0]
[ 0, 125*(75*s+1)*(s+4)]
E=
[ 0, 0]
[ 0, 0]
```

Se aplica la función "fcomfd" (de esta tesis) para obtener la realización en forma controlador

```
>>[A,B,C]=fcomfd(N,D)
A=
[ -301/75, -4/75, 0, 0]
[ 1, 0, 0, 0]
[ 0, 0, -301/75, -4/75]
[ 0, 0, 1, 0]
B=
[ 1/37500, 0]
[ 0, 0]
[ 0, 1/9375]
[ 0, 0]
C=
[ -439, 1756, 108, -432]
[ -541, 2164, 137, -548]
```

Con el fin de obtener más información del sistema se aplica la función "pzmimo" (de esta tesis). La función "pzmimo" obtiene los polos y los ceros del sistema.

```
>>[polos,ctransm,cinvar,cdesent,cdessal,cinf]=pzmimo(double(A),double(B),double(C),double(E))
polos=
```

```
-4.0000
-4.0000
-0.0133
-0.0133
ctransm=
4
4
cinvar=
4.0000
4.0000
cdesent=
Empty matrix: 0-by-1
cdessal=
Empty matrix: 0-by-1
cinf=
0
0
```

La información que tenemos ahora es:

a) El sistema es completamente controlable puesto que no hay ceros de desacoplamiento de entrada

b) El sistema es completamente observable puesto que no hay ceros de desacoplamiento de salida

- Si el sistema es completamente controlable y observable entonces el conjunto de ceros de transmisión (ctrans) coincide con el conjunto de ceros invariantes (cinvar), y esto se puede apreciar en los resultados obtenidos anteriormente.

c) El sistema es de fase no mínima puesto que se los ceros de transmisión están ubicados dentro del semiplano derecho del plano s; esto es debido a la aproximación de Padé

d) El sistema es estable puesto que sus polos están ubicados en el semiplano izquierdo del plano s

Calculo del controlador LQ:

Se definen las matrices de ponderación utilizadas en la función "lqr" (de Control Toolbox de Matlab)

```
q=eye(4)*10000;    % ponderación sobre los estados
r=eye(2);   % ponderación sobre las entradas
```

Se transforman las matrices del sistema a formato numérico

```
a=double(A);
b=double(B);
c=double(C);
e=double(E);
```

Se calcula el controlador LQ

```
klq=lqr(a,b,q,r)
```

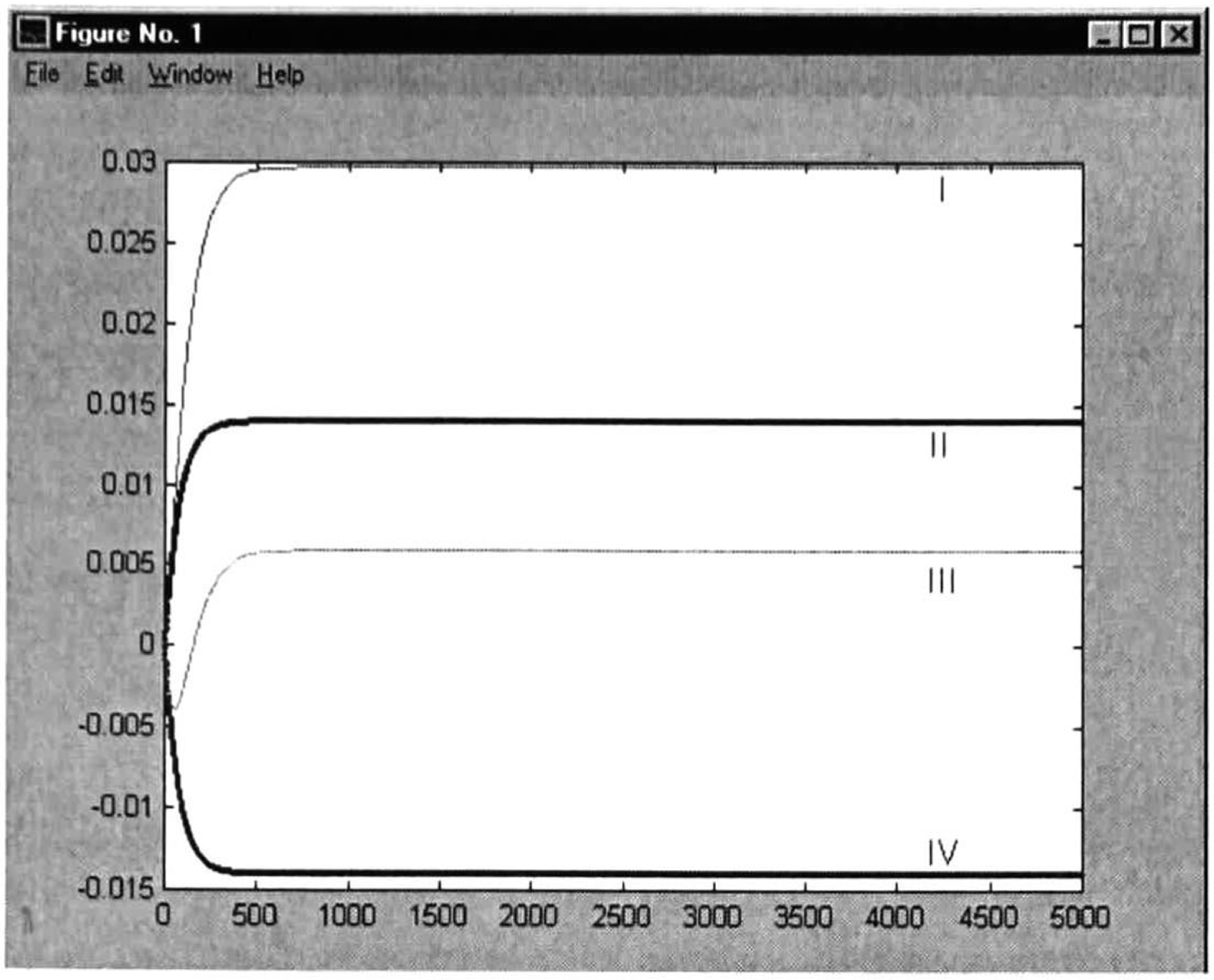

Figura 5.3: Respuestas de las salidas en lazo abierto y cerrado

```
klq =

 0.6558 2.4984 0.0000 0.0000

 0.0000 0.0000 2.6001 9.9020
```

Se simula la respuesta del sistema en lazo abierto a una entrada escalón

```
tf=5000;    % tiempo final de simulación

t=0:1:tf;   % duración y paso de la simulación

ref=[ones(size(t))' , ones(size(t))'];    % señal escalón

[yla,x]=lsim(a,b,c,e,ref,t);
```

Se simula la respuesta del sistema en lazo cerrado a una entrada escalón

```
af=a-b*klq;    %dinámica en lazo cerrado

yf=lsim(af,b,c,e,ref,t);
```

Se visualiza los resultados de la simulación (lazo abierto y lazo cerrado) del sistema, Figura 5.3

```
plot(t,yla,'.',t,yf);
```

En la Figura 5.3 se tiene que para lazo abierto $y1$ (II) y $y2$ (IV) y lazo cerrado $y1$ (I) y $y2$ (III). Al añadir el controlador al sistema de lazo abierto y cerrar el lazo se ha logrado que la respuesta del sistema sea positiva.

Con esto se muestra la sencillez de al elaborar un controlador a partir de la realización

Capítulo 6

Conclusiones

En el proceso de análisis de sistemas de control MIMO, para resolver ciertos problemas tales como el estudio de controlabilidad y observabilidad, diseño de leyes de control, etc., se detectó que no existen dentro del "software" comercial ciertas herramientas computacionales que agilicen la labor del cálculo matemático. En base a lo anterior, se planteó la necesidad de desarrollar herramientas computacionales; por lo cual se realizo un estudio de los métodos utilizados para analizar los sistemas MIMO con herramientas analíticas.

En este trabajo se han presentado una serie de algoritmos, los cuales han sido programados utilizando un lenguaje simbólico dentro de la plataforma de Matlab 5.x. Estos algoritmos programados son funciones integrables a las librerías de Matlab; es decir, que el usuario puede hacer uso de ellas como con cualquier otra función de Matlab e incluso obtener ayuda en línea si lo desea. Con la caja de herramientas presentada en este trabajo; básicamente, se ha resuelto el problema de pasar de una matriz de transferencia a un sistema de ecuaciones de estado. Para solucionar el problema, antes mencionado, se han desarrollado varias funciones con las cuales se pueden obtener diferentes formas de realización en el espacio de estado (controlador, observador, controlabilidad, observabilidad), también se han incluido dos funciones más que son para obtener las formas canónicas de controlabilidad y observabilidad. Se ha desarrollado, también, un programa para obtener una descripción en fracción matricial (MFD), izquierda y/o derecha, de una matriz de transferencia. Si ya se tiene una MFD izquierda y/o derecha existen dos funciones con las que se puede pasar de una a la otra respectivamente.

Un problema importante que se ha resuelto, es la obtención de los polos y ceros de sistemas MIMO. Cabe decir que se obtienen, con la función presentada en este trabajo "pzmimo", los valores de los ceros correspondientes a varias clasificaciones de ellos; como son, los ceros de transmisión, los ceros invariantes, los ceros de desacoplamiento de entrada, los ceros de desacoplamiento de salida y los ceros al infinito.

En torno a estos problemas, también, se han desarrollado herramientas de análisis como por ejemplo: "isponal" que evalúa si una matriz cualquiera es polinomial. Otras funciones lógicas son las que analizan si una matriz es reducida por filas o columnas.

El esfuerzo de este trabajo muestra la capacidad de crear herramientas computacionales lo cual acentúa que no es necesario comprar a otros países este tipo de tecnología (software).

Finalmente se puede concluir que el uso de herramientas computacionales es bueno pues el analista puede enfocarse más a problemas de control y no al cuidado de su desarrollo matemático. Al usuario le resulta cómodo, el uso de estas herramientas, ya que puede hacer cálculos rápidos y confiables lo cual acelera por ende la dinámica de resolución de problemas de diseño y/o análisis. En la enseñanza de teoría de sistemas, las herramientas computacionales, pueden ser usadas como apoyo pedagógico.

Finalmente queda pendiente como trabajos futuros seguir enriqueciendo esta caja de herramientas conforme se detecten inexistencia de funciones para cubrir necesidades de rapidez en el desarrollo de los análisis y diseños de sistemas de control. También, queda pendiente el estudio y mejora de la robustes numérica de los programas de esta caja de herramienta, así como mejorar su complejidad algorítmica.

Bibliografía

[1] A. Baldor, "Algebra", Publicaciones Cultural, 1995.

[2] Britannica, "Encyclopædia Britannica Publishers, Inc.", LEXIPEDIA, 1994-1995.

[3] R. Y Chiang, M. G. Safonov, "Robust Control Toolbox, MathWorks", 1988.

[4] J. C. Doyle, K. Glover,"State-Space Solutions to Standard H_2 and H_∞ Control Problems", IEEE Transactions on Automatic Control, Vol. 34, No 8, August 1989.

[5] H. Elmqvist, "SIMNON", 1972.

[6] E. Enander, A. Sjöberg, Bo Melin, and Pernilla Isaksson, "The MATLAB Handbook", Addison-Wesley, 1996.

[7] B. A. Francis, "A course in H_∞ Control Theory", Springer Verlag, 1987.

[8] Golub G.H. and Van Loan C.F., "Matrix Computations", John Hopkins University Press, 1983.

[9] L.O.González.C., "Una interface entre la entrada y salida de SPICE para el análisis de circuitos en el plano s", Tesis de Maestría, Instituto Nacional de Astrofísica Optica y Electrónica, 1997.

[10] A. Heck, "Introduction to MAPLE", Springer-Verlag, New York, 1993.

[11] Householder A.S., "The Theory of Matrices in Numerical Analisys", Dover, 1964.

[12] T. Kailath, "Linear System", Prentice Hall, 1980.

[13] H. Kwakernaak, M. Sebek, "Polinomial Toolbox", Versión pre-comercial disponible únicamente en Internet (http://www.math.utwente.nl/polbox), 1997.

[14] A. J. Laub, John N. Little, "Control System Toolbox", MathWorks, 1992.

[15] A. G. J. Macfarlane, N. Karcanias, "Poles ans Zeros of linear multivariable systems: a survey of the algebraic, geometric, and complex-variable theory", Int. J. Control, 1976.

[16] T. E. Marlin, "The Software Laboratory for Undergraduate Process Control Education", European Symposium on Computer Aided Process Engineering, 1996.

[17] The MathWorks Inc. "High-Performance Numeric Computation and Visualization Software", Matlab, 1993.

[18] Matlab, "MathWorks Inc.", 1996.

[19] C. Moler, P. J. Costa, "Symbolic Math Toolbox", MathWorks, 1997.

[20] M. B. Monagan, Et Al, "MAPLE V programming guide", Waterloo Maple, Inc., 1996.

[21] Ogata K., "Ingenieria de Control Moderna", Prentice-Hall, 1980.

[22] Ch. C. Paige, "Properties of Numerical Algorithms Related to Computing Controllability", IEEE Transactions on Automatic Control, February 1981.

[23] D. Redfern, "Maple V Release 5", Waterloo Maple, Inc., 1997.

[24] G. R. Romero Galvan, "Análisis y Síntesis de Sistemas de Control Lineal por Computadora", Tesis de Maestría, Universidad Autónoma de Nuevo León, 1993.

[25] M. Shacham, "Application of Feedback Control Principles for Solving Differential - Algebraic Systems of Equations in Process Control Education", European Symposium on Computer Aided Process Engineering, 1996.

[26] B. Shanian, M. Hassul, "Control system design using MATLAB", Prentice-Hall, 1995.

[27] S. Skogestad, I. Postlethwaite, "Multivariable Feedback Control", Wiley, 1996.

[28] VisSim, "VisSim", Visual Solutions Incorporated, 1989.

[29] D. S. Watkins, "Fundamentals of MATRIX Computations", John Wiley & Sons, Inc., 1991.

[30] J. F. Whidborne, I. Postlethwaite, and D.-W. Gu, "Robust Controller Design Using H_∞ Loop-Shaping and the Method of Inequalities", IEEE Transactions on Control Systems Technology, Vol. 2 No. 4, December 1994.

[31] Wolfram Research, "Mathematica 2.2", 1994.

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN
UNIDAD GUADALAJARA

El Jurado designado por el Laboratorio de Ingeniería Eléctrica y Ciencias de la Computación del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis: "Herramientas Computacionales para Análisis de Sistemas en Control Automático" el día 25 de septiembre de 1998.

Dr. Edgar Sánchez Camperos
Profesor Investigador 3B
CINVESTAV DEL IPN
Guadalajara.

Dra. Ofelia Begovich Mendoza
Profesor Investigador 3A
CINVESTAV DEL IPN
Guadalajara.

Dr. Antonio Ramírez Treviño
Profesor Investigador 2A
CINVESTAV DEL IPN
Guadalajara.

Dr. José Javier Ruiz León
Profesor Investigador 2A
CINVESTAV DEL IPN
Guadalajara.

M. C. Alejandro Justo Malo Tamayo
Profesor Titular 1B
CINVESTAV DEL IPN
Guadalajara.