UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Laboratorio 2 - Modelos de Estado

Integrantes:	Vanina Correa Chávez
	John Serrano Carrasco
Curso:	Modelos y Simulación
Sección:	13321-0-A-1
Profesor:	Gonzalo Acuña Leiva
Ayudante:	Tómas López Aleuy

16 de Junio de 2023

Tabla de contenidos

1. Introducción

Los modelos fenomenológicos son aquellos modelos creados mediante una metodología teórica, es decir, son creados gracias a conocimientos científicos y leyes aceptadas en el dominio de lo que se quiere modelar. Debido a lo anterior, tienden a ser modelos comprensibles, argumentables, defendibles y por sobre todo, generalizable a una gran cantidad de situaciones.

Los modelos de estado surgen de una mentalidad de lograr un tratamiento computacionalmente más eficiente en la operación de los modelos fenomenológicos. En otras palabra, los modelos fenomenológicos, si bien son generalizables, tienden a ser complejos y requerir la ayuda de expertos para ser completamente entendidos, siendo ahí donde entran a jugar un rol clave los modelos de estado, los cuales simplifican un modelo para que este sea mas fácil y agradable de comprender.

Como objetivo principal, se tiene realizar el análisis de modelos de estado utilizando MATLAB, donde a lo largo de la experiencia se utilizarán distintas funciones, como step(), ss(), lsim(), entre otras.

Como estructura del informe, se tiene una definición de varios conceptos claves e importantes para la comprensión del texto, correspondiente a un Marco Teórico donde se definen conceptos como Modelo, Modelo de Estado, Modelo Fenomenológico, Variables de estado, entre otros. Luego se muestra el desarrollo de cada parte del laboratorio, indicando el paso a paso para obtener los distintos resultados junto con los gráficos correspondientes. Para finalizar, se realiza una conclusión de todo el trabajo realizado, teniendo en cuenta varios aspectos del informe como también de los códigos realizados en MATLAB.

2. Marco Teórico

2.1. Modelo

Un modelo es una representación de un sistema, el cual es una porción de la realidad la cual se quiere estudiar. Un modelo debe ser similar a una herramienta y como tal, debe ser lo mas simple posible y solo contener los elementos esenciales de un sistema. *Existen múltiples tipos de modelos, como por ejemplo, los modelos fenomenológicos y también existen múltiples representaciones de los distintos tipos de modelo, como por ejemplo, los modelos de estado.* (Ibáñez (2008))

2.2. Modelo Fenomenológico

Un modelo fenomenológico es aquel que es construido mediante una metodología fundamental o teórica. Estos modelos son deductivos y son construidos a través de conocimientos científicos y leyes científicas. *Son generalizables pero suelen requerir de un expertos para poder ser comprendidos en su totalidad. Son modelos complejos pero que pueden ser más fácil de entender si se representan mediante un modelo de estado.* (Acuña (2020a))

2.3. Modelo de Estado

Los modelos de estado son una representación simplificada de los modelos fenomenológicos, los cuales ser complicados e incómodos de trabajar. Los modelos de estado representan a un modelo mediante matrices y vectores, utilizando unas variables denominadas Variables de Estado. Para un Sistema Lineal Invariante en el tiempo, un modelo de estado tiene la siguiente forma:

$$X' = AX + BU \tag{1}$$

$$Y = CX + DU \tag{2}$$

Donde X' es la matriz correspondiente a las derivadas de primer orden de las variables de estado, A es la matriz de los coeficientes que acompañan a las variables de estado, X es la matriz de las variables de estado, B es la matriz correspondiente a los coeficientes que acompañan a las entradas, U es la matriz de variables de entradas, Y es la matriz de variables de salidas, C es la matriz que acompañan a los coeficientes de las variables de estado, D es la matriz correspondiente a los vectores que acompañan a las entradas. *Un sistema de ecuaciones correspondiente a un Modelo de Estado, tiene una solución.* (Ogata (2010))

2.4. Variables de Estado

Las Variables de estado son variables que permiten predecir el comportamiento futuro de un sistema para cierto punto en el tiempo. (Acuña (2020b)) Estas variables suelen estar asociadas a elementos acumuladores, como por ejemplo, una variable asociada al liquido actual de un estanque. Se puede dar el caso de que las variables de estado puedan ser también las variables de salida, pero no siempre es así.

2.5. Variables de entrada

Las Variables de Entrada de un sistema son aquellas variables que afectan al sistema pero el sistema no las afecta a ella. (Acuña (2020a)) Puede ser una o varias.

2.6. Variables de salida

Las variables de salida de un sistema son aquellas cuyo comportamiento se quiere estudiar y conocer. (Acuña (2020a)) Es el diseñador quien las escoge.

2.7. Función de Transferencia

Una función de Transferencia, la cual comúnmente es denominada como $H(s)$, corresponde a una herramienta muy esencial a la hora de analizar la respuesta a un sistema. En palabras simples, *las Funciones de Transferencia suelen ser ecuaciones matemáticas que describen como un sistema se comporta ante una cierta entrada.* (Castaño (2020)) Es posible llevar un modelo de estado a una función de transferencia y obtener un modelo de estado a través de una función de transferencia.

3. Desarrollo

3.1. Primera parte

Se presenta el diagrama de bloques mostrado en la Figura 1 . Se solicita escribir una función $bam()$ que reciba como parámetro los valores a, b, c, d, e y f y que retorne las matrices del modelo de estado, además escribir una función mab cuya entrada sea la salida de la función anterior y retorne la función de transferencia $H(s)$ del sistema. Luego, se debe graficar el resultado de la función $mab()$ y compararlo con la función $step()$ de Matlab para verificar los resultados.

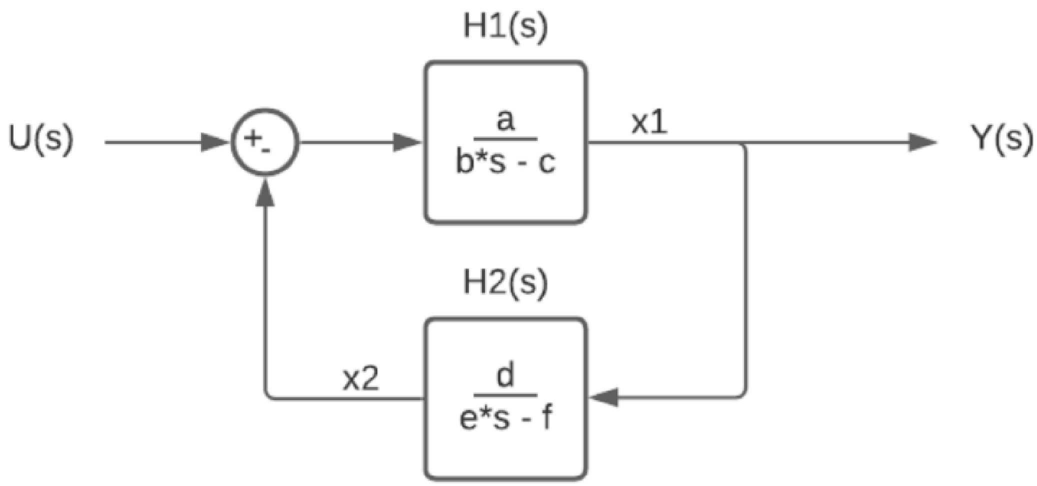

Figura 1: Diagrama de bloques.

De manera adicional, se pide el desarrollo algebraico para ambas funciones. Entonces, se identifica como variable de entrada la función $U(s)$, la variable de salida $Y(s)$ y dado que las funciones de transferencia del sistema son de primer orden, se definen como variables de estado las salidas de las funciones de transferencia X_1 y X_2. Utilizando la ecuación $Y(s) = H(s) \cdot U(s)$ se tiene:

$$X_1 = \frac{a}{bs - c} \cdot (U - X_2)$$

$$X_2 = \frac{d}{es - f} \cdot X_1$$

Se distribuyen ambas ecuaciones para despejar sX_1 y sX_2:

$$sX_1 = \frac{a}{b}U + \frac{c}{b}X_1 - \frac{a}{b}X_2$$

$$sX_2 = \frac{d}{e}X_1 + \frac{f}{e}X_2$$

Al volver al dominio del tiempo:

$$x_1' = \frac{a}{b}u + \frac{c}{b}x_1 - \frac{a}{b}x_2$$

$$x_2' = \frac{d}{e}x_1 + \frac{f}{e}x_2$$

Luego, se construye el modelo de estado matricial, considerando la salida $y = x_1$.

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \frac{c}{b} & -\frac{a}{b} \\ \frac{d}{e} & \frac{f}{e} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{a}{b} \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u$$

Para obtener la función de transferencia, se reemplazan las matrices en la ecuación $H = C(sI - A)^{-1}B + D$, considerando que:

$$A = \begin{bmatrix} \frac{c}{b} & -\frac{a}{b} \\ \frac{d}{e} & \frac{f}{e} \end{bmatrix} B = \begin{bmatrix} \frac{a}{b} \\ 0 \end{bmatrix} C = \begin{bmatrix} 1 & 0 \end{bmatrix} D = 0$$

$$sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} \frac{c}{b} & -\frac{a}{b} \\ \frac{d}{e} & \frac{f}{e} \end{bmatrix} = \begin{bmatrix} s - \frac{c}{b} & -\frac{a}{b} \\ \frac{d}{e} & s - \frac{f}{e} \end{bmatrix}$$

$$(sI - A)^{-1} = \frac{1}{s^2 - s(\frac{f}{e} + \frac{c}{b}) + \frac{ad+cf}{be}} \begin{bmatrix} s - \frac{f}{e} & \frac{a}{b} \\ -\frac{d}{e} & s - \frac{c}{b} \end{bmatrix}$$

$$C(sI - A)^{-1} = \frac{1}{s^2 - s(\frac{f}{e} + \frac{c}{b}) + \frac{ad+cf}{be}} \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} s - \frac{f}{e} & \frac{a}{b} \\ -\frac{d}{e} & s - \frac{c}{b} \end{bmatrix} = \frac{1}{s^2 - s(\frac{f}{e} + \frac{c}{b}) + \frac{ad+cf}{be}} \begin{bmatrix} s - \frac{f}{e} & \frac{a}{b} \end{bmatrix}$$

$$C(sI-A)^{-1}B+D = \frac{1}{s^2 - s(\frac{f}{e}+\frac{c}{b}) + \frac{ad+cf}{be}} \begin{bmatrix} s-\frac{f}{e} & \frac{a}{b} \end{bmatrix} \begin{bmatrix} \frac{a}{b} \\ 0 \end{bmatrix} + 0 = \frac{\frac{sa}{b} - \frac{af}{be}}{s^2 - s(\frac{f}{e}+\frac{c}{b}) + \frac{ad+cf}{be}}$$

Finalmente, la función de transferencia es:

$$H(s) = \frac{a(es-f)}{bes^2 - s(bf+ce) + cf + ad}$$

En MATLAB, se define la función *bam*() en MATLAB para calcular el modelo de estado matricial del diagrama de bloque. Recibe como entrada los valores a, b, c, d, e y f, luego retorna las matrices del modelo de estado.

```
function [A,B,C,D] = bam(a,b,c,d,e,f)
% Calcula las matrices del modelo de estado según resultado algebraico
% presentado en el informe.
A = [c/b -a/b ; d/e f/e];
B = [a/b ; 0];
C = [1 0];
D = 0;
end
```

Figura 2: Definición de función bam() en MATLAB.

De manera similar, se define la función *mab*() en MATLAB que recibe como entrada la salida de la función *bam*(), es decir las matrices del modelo de estado, y entrega la función de transferencia del sistema.

```
function H = mab(A,B,C,D)
% Matriz identidad 2x2.
I = eye(2);
% Se define el modelo para las funciones de transferencia.
s = tf('s');
% Utiliza fórmula para calcular la función de transferencia.
sI_A = s*I - A;
inversa = inv(sI_A);
H = C*inversa*B + D;
end
```

Figura 3: Definición de función mab en MATLAB.

Para corroborar el correcto funcionamiento de las funciones anteriores, se definen los valores de las variables a, b, c, d, e y f como 1 y se calcula el modelo de estado y la función de transferencia.

```
% Calcula modelo matricial utilizando la función bam.
[A,B,C,D] = bam(1,1,1,1,1,1)

% Calcula la función de transferencia utilizando la función mab.
H_mab = mab(A,B,C,D)
```

Figura 4: Cálculo de modelo de estado y función de transferencia con bam() y mab().

Luego, se obtienen los siguientes resultados:

```
A = 2×2
      1    -1
      1     1

B = 2×1
      1
      0

C = 1×2
      1     0

D = 0
H_mab =

      s - 1
  -------------
  s^2 - 2 s + 2
```

Figura 5: Resultado del modelo de estado y función de transferencia con bam() y mab().

Si bien, observando el desarrollo algebraico anterior se puede notar que el resultado es correcto, adicionalmente se calcula la función de transferencia utilizando la función $feedback()$ de MATLAB.

```
% Funciones de transferencia de los sistemas del diagrama de bloque.
H1 = a/(b*s - c);
H2 = d/(e*s - f);

% Calcula la función de transferencia utilizando funciones de Matlab para
% verificar el resultado anterior
H_feedback = feedback(H1, H2)
```

Figura 6: Código para calcular la función de transferencia con $feedback()$.

Luego, el resultado de la función de transferencia coincide, por lo que se confirma que el procedimiento es correcto. Lo anterior se puede comprobar en el gráfico a continuación, donde se gráfica la respuesta de ambas funciones de transferencia frente a un escalón.

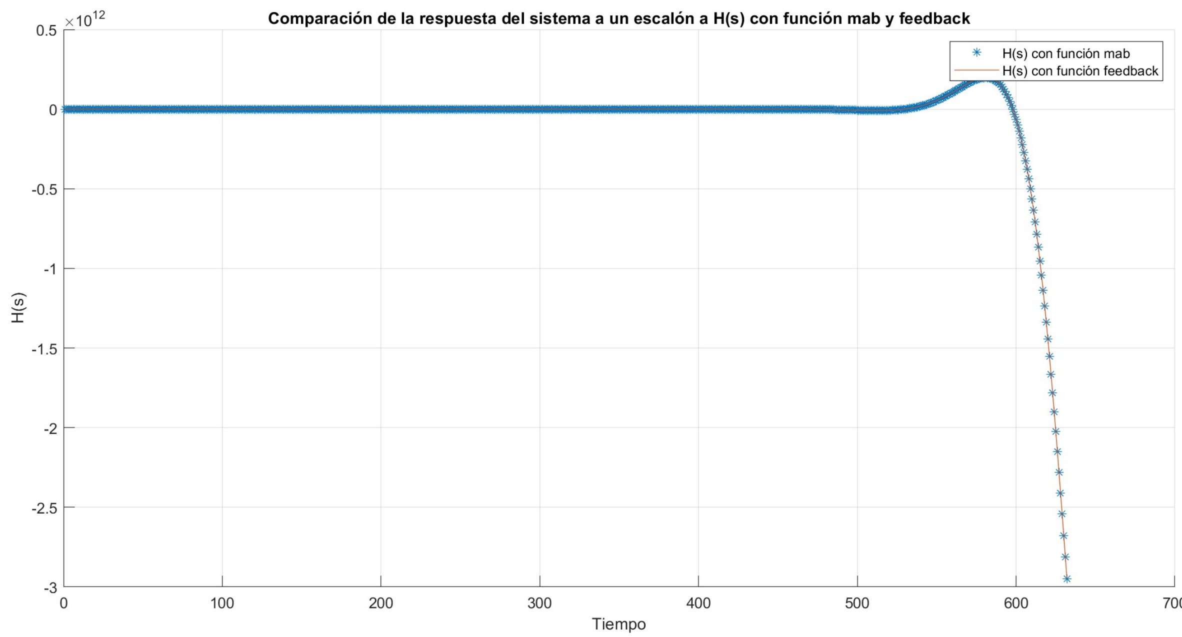

Figura 7: Gráfica de la respuesta de ambas funciones de transferencia frente al escalón.

3.2. Segunda parte

Dado el sistema mostrado en la Figura 8, se solicita obtener el modelo de estado algebraico y graficar el resultado del sistema ante un impulso, un escalón y una función $u(t)$. A continuación se entregan las siguientes ecuaciones cuando $A_1 = 2m^2$, $A_2 = 4m^2$, $R_{i1} = 0,25\frac{s}{m^2}$, $R_{i2} = 0,0625\frac{s}{m^2}$, $R_{s1} = 0,1\frac{s}{m^2}$ y $R_{s2} = 0,1\frac{s}{m^2}$

$$F_{i1} = \frac{h_1 - h_2}{R_{i1}}$$
$$F_{i2} = \frac{h_2 - h_1}{R_{i2}}$$
$$F_{s1} = \frac{h_1}{R_{s1}}$$
$$F_{s2} = \frac{h_2}{R_{s2}}$$

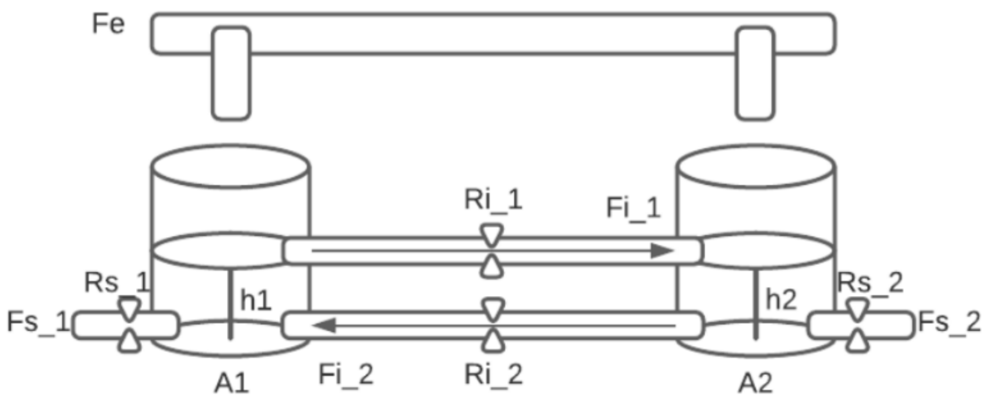

Figura 8: Diagrama de vasos comunicantes.

En primer lugar, se identifican las variables que participan en el modelo fenomenológico. Se tiene que la variable de entrada es F_0, las variables de salida son h_1 y h_2, dado que se desea controlar el nivel de los estanques y de igual manera, las variables de estado corresponden a h_1 y h_2.

Por lo tanto, a través de la variación del volumen de los vasos se tienen las siguientes ecuaciones diferenciales:

$$\frac{dV_1}{dt} = F_e - F_{i1} + F_{i2} - F_{s1}$$

$$\frac{dV_2}{dt} = F_e + F_{i1} - F_{i2} - F_{s2}$$

Considerando que $V_i = h_i \cdot A_i$, donde A_i es una constante:

$$\frac{dh_1}{dt} = \frac{F_e}{A_1} - \frac{F_{i1}}{A_1} + \frac{F_{i2}}{A_1} - \frac{F_{s1}}{A_1}$$

$$\frac{dh_2}{dt} = \frac{F_e}{A_2} + \frac{F_{i1}}{A_2} - \frac{F_{i2}}{A_2} - \frac{F_{s2}}{A_2}$$

Reemplazando las ecuaciones dadas en el enunciado:

$$\frac{dh_1}{dt} = \frac{F_e}{A_1} - (\frac{1}{A_1R_{i1}} + \frac{1}{A_1R_{i2}} + \frac{1}{A_1R_{s1}})h_1 + (\frac{1}{A_1R_{i1}} + \frac{1}{A_1R_{i2}})h_2$$

$$\frac{dh_2}{dt} = \frac{F_e}{A_2} + (\frac{1}{A_2R_{i1}} + \frac{1}{A_2R_{i2}})h_1 - \frac{1}{A_2R_{i1}} + \frac{1}{A_2R_{i2}} + \frac{1}{A_2R_{s2}})h_2$$

Finalmente, reemplazando los valores de las constantes:

$$\frac{dh_1}{dt} = \frac{F_e}{2} - 15h_1 + 10h_2$$

$$\frac{dh_2}{dt} = \frac{F_e}{4} + 5h_1 - \frac{15}{2}h_2$$

Por otro lado, se tiene que las salidas del sistema corresponden a $y_1 = h_1$ y $y_2 = h_2$

Entonces, utilizando la forma matricial se construye el modelo de estado del sistema.

$$\begin{bmatrix} h_1' \\ h_2' \end{bmatrix} = \begin{bmatrix} -15 & 10 \\ 5 & -\frac{15}{2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} F_e$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F_e$$

La información anterior se utiliza para reconstruir el modelo de estado en MATLAB y así poder obtener la función de transferencia $H(s)$ del sistema. Para hacer esto último se hace uso de la función $mab()$ definida en la parte 1.

```matlab
% Se limpian las variables guardadas en el Workspace.
clearvars;

% Se definen las matrices A, B, C y D
A = [-15 10; 5 -7.5];
B = [0.5; 0.25];
C = [1 0; 0 1];
D = 0;

% Se obtiene el modelo de estado del sistema
H_S = ss(A,B,C,D)

% Se utiliza la función mab() desarrollada en la parte 1 para
% obtener la función de transferencia H(s) del sistema.
H = mab(A,B,C,D)
```

Figura 9: Código de MATLAB para reconstruir el modelo de estado y obtener $H(s)$.

Se obtienen dos funciones de transferencia, $H_1(s)$ y $H_2(s)$, debido a que se tienen dos variables de salida asociadas al modelo. Los valores de $H_1(s)$ y $H_2(s)$ son los siguientes:

```
H =

  From input to output...
        0.5 s^3 + 17.5 s^2 + 171.9 s + 390.6
   1:  ---------------------------------------
       s^4 + 45 s^3 + 631.2 s^2 + 2812 s + 3906

        0.25 s^3 + 11.87 s^2 + 156.2 s + 390.6
   2:  ---------------------------------------
       s^4 + 45 s^3 + 631.2 s^2 + 2812 s + 3906

Continuous-time transfer function.
```

Figura 10: Valores de las funciones de transferencia del sistema $H_1(s)$ y $H_2(s)$.

Continuando, es necesario definir una entrada $u(t)$ para ver la respuesta de $H_1(s)$ y $H_2(s)$ ante esta entrada. En MATLAB, la entrada $u(t)$ se define de la siguiente manera:

```matlab
% Se define el intervalo de tiempo a utilizar para u(t)
t = linspace(0, 12*pi, 5000);
% Se define u(t)
u = 100*sin(t/4);
u(u<0) = 0.;
```

Figura 11: Código de la definición de la entrada $u(t)$ en MATLAB.

Ya con la entrada, se puede realizar un gráfico de la respuesta de las funciones de transferencia ante la entrada u(t), utilizando la función $lsim()$ de MATLAB. El gráfico resultante es el siguiente:

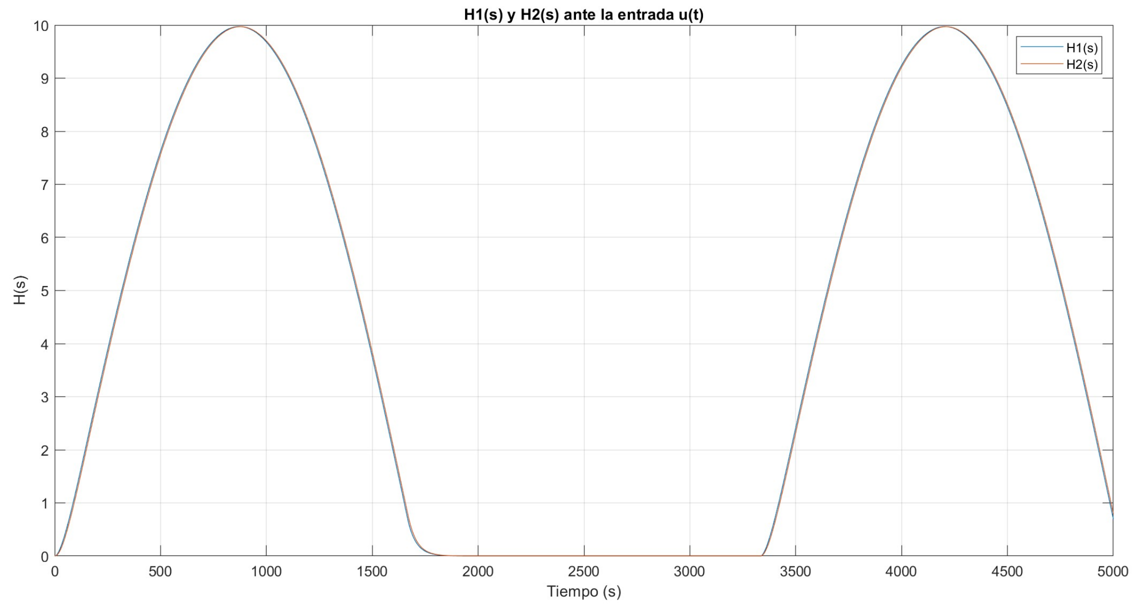

Figura 12: Gráfico de $H_1(s)$ *y* $H_2(s)$ *ante* $u(t)$*.*

El código de MATLAB relacionado al gráfico es el siguiente:

```
% Gráfico de H1(s) y H2(s) ante u(t)
figure;
plot(lsim(H, u, t));
title("H1(s) y H2(s) ante la entrada u(t)");
ylabel("H(s)");
xlabel("Tiempo (s)");A
legend("H1(s)", "H2(s)")
grid on;
```

Figura 13: Código de MATLAB del gráfico de $H_1(s)$ *y* $H_2(s)$ *ante* $u(t)$*.*

Como se puede ver en el gráfico, tanto $H_1(s)$ como $H_2(s)$ siguen la misma trayectoria y se superponen una sobra otra. A medida que avanza el tiempo, estas funciones llegan a valores alto y luego bajan hasta llegar a y = 0. Se mantienen constantes por un periodo de tiempo hasta que vuelven a subir y repetir el proceso. Se puede notar que la forma del gráfico tiene dos puntos extremos y es similar a una forma ondulatoria.

Ahora se realiza un gráfico de la respuesta de $H_1(s)$ y $H_2(s)$ ante un impulso, utilizando la función $impulse()$ de MATLAB.

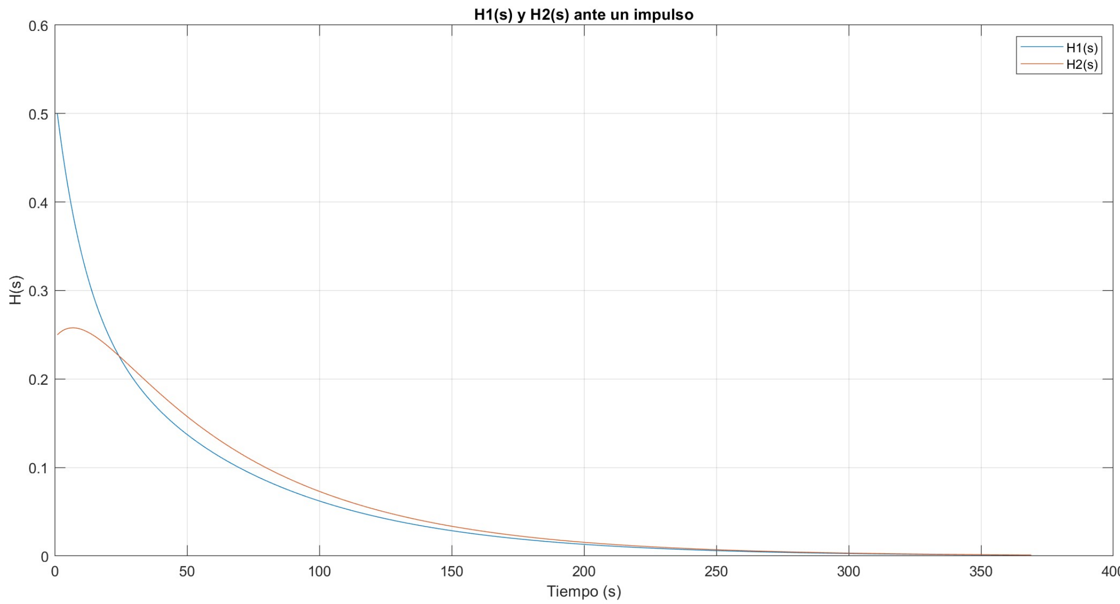

Figura 14: Gráfico de $H_1(s)$ y $H_2(s)$ ante un impulso.

El código de MATLAB relacionado al gráfico es el siguiente:

```
% Gráfico de H1(s) y H2(s) ante un impulso
figure;
plot(impulse(H));
title("H1(s) y H2(s) ante un impulso");
ylabel("H(s)");
xlabel("Tiempo (s)");
legend("H1(s)", "H2(s)")
grid on;
```

Figura 15: Código de MATLAB del gráfico de $H_1(s)$ y $H_2(s)$ ante un impulso.

Se puede notar que ambas funciones comienzan desde un punto elevado y a medida que avanzan el tiempo se acercan a y = 0. $H_1(s)$ Comienza desde y = 0.5, en cambio, $H_2(s)$

comienza cercano a y = 0.25. Esta última función sube por un corto periodo de tiempo y luego comienza a detener, donde se intersecta con $H_1(s)$ en un punto especifico y continua descendiendo hasta intersectarse completamente cuando ambas funciones están muy cercanas a y = 0.

Por último, se realiza un gráfico de la respuesta de $H_1(s)$ y $H_2(s)$ ante un escalón, utilizando la función *step*() de MATLAB.

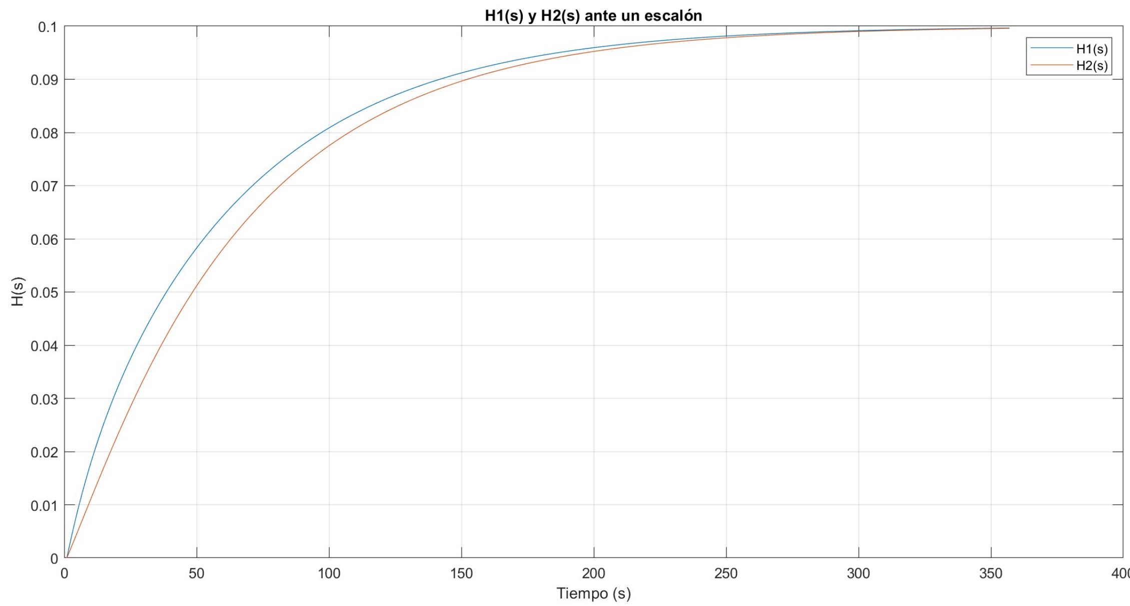

Figura 16: Gráfico de $H_1(s)$ y $H_2(s)$ ante un escalón.

El código de MATLAB relacionado al gráfico es el siguiente:

```
% Gráfico de H1(s) y H2(s) ante un escalón
figure;
plot(step(H));
title("H1(s) y H2(s) ante un escalón");
ylabel("H(s)");
xlabel("Tiempo (s)");
legend("H1(s)", "H2(s)")
grid on;
```

Figura 17: Código de MATLAB del gráfico de $H_1(s)$ y $H_2(s)$ ante un escalón.

Se puede ver del gráfico que ambas funciones comienzan desde el punto (0,0) y luego ascienden, donde cercano a y= 0,1 comienzan a intersectarse y cuando llegan a y =

0,1 se intersectan completamente y se estabilizan. Se puede observar que en ciertos intervalos $H_1(s)$ logra alcanzar amplitudes mayores que $H_2(s)$.

Por lo general y como se puede observar de los 3 gráficos anteriores, la respuesta del sistema va a variar dependiendo de cual sea la entrada, pero tanto $H_1(s)$ como $H_2(s)$ van a tener comportamientos muy similares entre si y en cierto punto se intersectarán.

4. Conclusión

En conclusión, se cumplió en su totalidad los objetivos del segundo laboratorio de Modelos y Simulación.

En primer lugar, se lograron crear las funciones $bam()$ y $mab()$ en MATLAB para obtener el modelo de estado matricial de un sistema y su función de transferencia, respectivamente, para luego comparar los resultados con la función $feedback()$ y graficar ambas respuestas del sistema frente a un escalón. Además, se incluye su desarrollo algebraico.

En la segunda parte, se logran identificar las variables que participan en el modelo fenomenológico, se encuentra su modelo de estado y la función de transferencia, para luego comparar gráficamente frente a un impulso, escalón y una entrada $u(t)$. Se comparan los resultados y se destacan ciertos aspectos de estos.

Se pudo comprobar que los Modelos de Estado si son una representación agradable de los modelos fenomenológicos que se pueden aplicar a una gran variedad de situaciones y que nos permiten llevar modelos complejos a una representación simple y fácil de trabajar.

Finalmente, se espera que los conocimientos obtenidos en esta actividad sean empleados en el resto de laboratorios, resultando beneficiosos al establecer una base sólida respecto a modelos fenomenológicos para comenzar con la segunda parte de los contenidos del ramo.

Bibliografía

Acuña, G. (2020a). *Modelación*. Universidad de Santiago de Chile.

Acuña, G. (2020b). *Modelos de estado*. Universidad de Santiago de Chile.

Castaño, S. (2020). Cero de una función de transferencia. https://controlautomaticoeducacion.com/analisis-de-sistemas/cero-de-una-funcion-de-transferencia/. Artículo Online. Recuperado el 10 de junio de 2023.

Ibáñez, J. (2008). Conceptos y tipos de modelos cientificos. https://www.madrimasd.org/blogs/universo/2008/05/10/91441. Artículo Online. Recuperado el 09 de junio de 2023.

Ogata, K. (2010). *Ingeniería de Control Moderna*. Pearson Prentice Hall, 3 edition.