

Departamento de Control
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

(A3) Sistemas y Señales I

Trabajo Práctico N° 2 Simulación Digital de Sistemas Dinámicos

Autores: Cátedra SyS-I

Febrero de 2025

Trabajo Práctico N° 2

Simulación Digital de Sistemas Dinámicos

1. Objetivos

El principal objetivo de este Trabajo Práctico es el análisis de Sistemas Dinámicos en Tiempo Continuo a través del modelado y la simulación digital usando el programa **Matlab/Simulink**. Se considerarán distintos sistemas físicos reales y se estudiará su comportamiento mediante un análisis en los dominios temporal y frecuencial.

2. Introducción

En este Trabajo Práctico se analizará el comportamiento de distintos sistemas físicos a través del modelado y simulación digital usando el programa **Matlab/Simulink** [1]. Los sistemas que se considerarán en el Trabajo Práctico corresponden a algunos de los equipos para enseñanza de Control Automático que distribuye la firma Educational Control Products (ECP), de California, USA [3], de los cuales hay información disponible acerca de los valores numéricos de algunos de los parámetros que caracterizan a los sistemas.

Para los distintos sistemas que se considerarán en el práctico, se obtendrá un modelo en la forma de ecuaciones diferenciales (y/o diagrama de bloques) basándose en leyes fundamentales (usando la metodología desarrollada durante el curso). Las ecuaciones diferenciales que describen el comportamiento del sistema podrán entonces integrarse numéricamente en una forma muy simple utilizando alguna de las rutinas para integración numérica disponibles en **Matlab** (como por ejemplo las funciones `ode45`, `ode23`, `ode113`, `ode15s` y `ode23s`). Estas rutinas permitirán analizar el comportamiento del sistema ante diferentes señales de excitación, y diferentes condiciones iniciales.

Para aquellos sistemas que son lineales, o que admiten un modelo lineal válido en cierto rango de operación, la respuesta del sistema a entradas arbitrarias se podrá computar usando la función `lsim` del *Control System Toolbox* [2] de **Matlab**. Para estos sistemas se analizará además su respuesta en frecuencia usando alguna de las funciones que el mencionado *Control System Toolbox* tiene disponibles para tal fin, como ser la rutina `bode`.

En algunos casos se plantearán además problemas simples de diseño de controladores, para los cuales se podrán usar algunas de las funciones disponibles en el *Control System Toolbox*.

3. Desarrollo del Trabajo Práctico

3.1. Integración numérica de Ecuaciones Diferenciales Ordinarias (EDOs) usando Matlab

En esta Subsección se ilustrará, a través de ejemplos, como puede usarse **Matlab** para resolver problemas de valor inicial de ecuaciones diferenciales ordinarias. Se discutirá como representar los problemas de valor inicial en **Matlab**, y como aplicar los comandos de integración numérica de **Matlab** para resolver estos problemas.

Supongamos que se quiere resolver (numéricamente) el problema de valor inicial (PVI) dado por una ecuación diferencial ordinaria de orden n con condiciones iniciales, de la forma

$$\begin{cases} y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}) \\ y(0) = y_0, \quad y'(0) = y'_0, \quad \dots, \quad y^{(n-1)}(0) = y_0^{(n-1)} \end{cases}$$

donde t es la variable independiente tiempo, e $y^{(k)}$ denota la k -ésima derivada de la variable dependiente y con respecto al tiempo. Para representar este PVI en **Matlab**, se deben seguir los siguientes pasos.

- i. Escribir la EDO de orden n como un conjunto de n ecuaciones diferenciales de primer orden. Esto puede lograrse definiendo, por ejemplo, las variables (de estado, en la terminología de Teoría de Sistemas)

$$\begin{cases} x_1 = y \\ x_2 = y' \\ \vdots \\ x_n = y^{(n-1)} \end{cases}$$

con lo que se obtiene un sistema de ecuaciones diferenciales de primer orden equivalente de la forma

$$\begin{cases} x'_1 = x_2 \\ x'_2 = x_3 \\ \vdots \\ x'_{n-1} = x_n \\ x'_n = f(t, x_1, x_2, \dots, x_n) \end{cases},$$

con condiciones iniciales $x_1(0) = y_0$, $x_2(0) = y'_0$, \dots , $x_n(0) = y_0^{(n-1)}$. Definiendo el vector de estados

$$x = [x_1, x_2, \dots, x_n]^T$$

el sistema de ecuaciones de primer orden puede escribirse en forma vectorial como

$$x' = F(t, x),$$

con condiciones iniciales $x(0) = [y_0, y'_0, \dots, y_0^{(n-1)}]^T$.

- ii. Se debe crear un archivo .m del tipo *function file* (que de ahora en más denominaremos ODE-file), que tenga dos argumentos de entrada, t y x , y un argumento de salida, de la forma

```
function dx=sistema(t,x)
dx=[x(2); x(3); ...; x(n); f(t,x(1),x(2),...,x(n))];
```

Ambos argumentos de entrada t y x , deben estar presentes, aún cuando la función no haga uso de ellos. En este caso, el argumento de salida dx es un vector columna.

- iii. Una vez que el ODE-file ha sido creado se debe llamar a alguna de las rutinas de integración numérica de **Matlab** (es decir las funciones `ode45`, `ode23`, `ode113`, `ode15s` y `ode23s`), para que resuelva el PVI. La forma general de la sintaxis para invocar estas funciones es

```
[T,x]=solver('sistema',tspan,x0)
```

donde `solver` es alguna de las rutinas `ode45`, `ode23`, `ode113`, `ode15s`, u `ode23s`, y los argumentos de entrada son

```
sistema          nombre del ODE-file,
```

`tspan` vector de la forma `[t0 tfinal]` que hace que la rutina integre entre los tiempos `t0` y `tfinal`,
`x0` vector columna con las condiciones iniciales en el tiempo `t0`.

Cada elemento en el vector columna `x` es el valor de la variable dependiente en un tiempo dado por el correspondiente elemento del vector `T`.

Consideraremos ahora algunos ejemplos que ilustran el procedimiento descrito.

Ejemplo 1: Ecuación de Lienard

Consideremos el sistema (no lineal) descrito por el sistema de ecuaciones de primer orden siguiente

$$\begin{cases} \dot{x}_1 = x_2 - F(x_1) \\ \dot{x}_2 = -g(x_1) \end{cases} \quad (1)$$

Este sistema fue originalmente estudiado por el físico francés A. Lienard en 1928, quien estableció las condiciones sobre las funciones $F(x_1)$ y $g(x_1)$ que aseguraban la existencia de una solución periódica aislada (es decir un **ciclo límite**) y estable, del sistema de ecuaciones (1) (ver [4]).

En particular, las funciones $F(x_1) = \frac{x_1^3 - x_1}{x_1^2 + 1}$, y $g(x_1) = x_1$, verifican las condiciones establecidas

por Lienard, y por lo tanto aseguran la existencia de un único ciclo límite estable, solución del sistema de ecuaciones (1).

Nuestro interés es simular el Sistema de Lienard, a partir de distintas condiciones iniciales, usando **Matlab**. El ODE-file que representa al sistema, tomaría la forma

```
function dx=lienard(t,x)
dx=[x(2)-(x(1)^3 - x(1))/(x(1)^2 +1);-x(1)];
```

Los comandos **Matlab** para la integración numérica y la visualización de los resultados serían

```
[t,x]=ode45('lienard',[0 30],[0.01 0.01]);
subplot(211), plot(t,x(:,1),'k-',t,x(:,2),'k--');
xlabel('Tiempo [s]'); title('Solucion de la Ecuacion de Lienard');
grid; legend('x1','x2');
subplot(212), plot(x(:,1),x(:,2));
xlabel('x1'); ylabel('x2'); title('Plano de Fase'); grid;
```

La salida gráfica correspondiente se representa en la Figura 3.1.1.

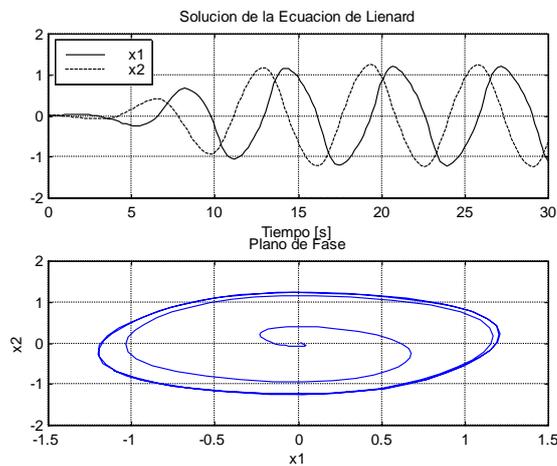


Figura 3.1.1: Salida gráfica correspondiente al Ejemplo 1.

□

Ejemplo 2: Sistema de dos masas y resorte

El modelo 210a de ECP que se encuentra en el Laboratorio de Control, y que se muestra en la Figura 3.1.2, es un sistema mecánico traslacional que permite varias configuraciones a partir de la conexión de hasta 3 carritos (cuya masa puede variarse) mediante resortes de diferentes constantes elásticas y un servo motor que permite simular diferentes fuerzas aplicadas al sistema.

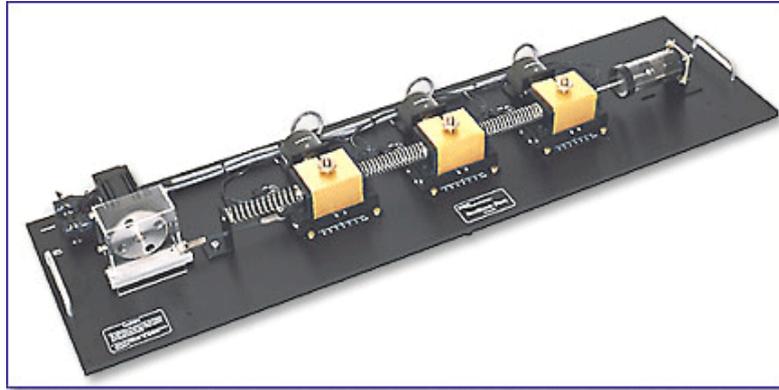


Figura 3.1.2: Modelo 210a ECP - Sistema Mecánico Traslacional.

La Figura 3.1.3, representa esquemáticamente una configuración del equipo anterior constituida por dos masas unidas por un resorte lineal. Se quiere estudiar la respuesta del sistema a un pulso de duración finita en la fuerza F . Se asume que no hay rozamiento.

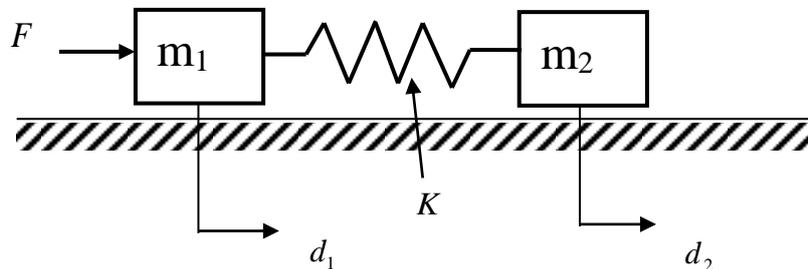


Figura 3.1.3: Sistema Mecánico Traslacional.

Basándose en la 2^{da} ley de Newton, es fácil ver que las ecuaciones que describen el comportamiento del sistema son

$$\begin{cases} m_1 d_1'' = F + K(d_2 - d_1) \\ m_2 d_2'' = -K(d_2 - d_1) \end{cases}$$

que pueden escribirse como un sistema de 4 ecuaciones de primer orden de la forma

$$\begin{cases} x_1' = x_2 \\ x_2' = \frac{F}{m_1} + \frac{K}{m_1}(x_3 - x_1) \\ x_3' = x_4 \\ x_4' = -\frac{K}{m_2}(x_3 - x_1) \end{cases}$$

donde se han definido las variables de estado $x_1 = d_1$, $x_2 = d_1'$, $x_3 = d_2$, $x_4 = d_2'$.

Asumiendo los siguientes valores de los parámetros $m_1 = 2$ kg, $m_2 = 2$ kg (provistos por el fabricante), $K = 273$ N/m (estimado a partir de ensayos), el ODE-file que representa al sistema será de la forma

```

function dx=mass_spring_mass(t,x)
k=273;
m1=2;
m2=2;
if t<0.5
    F=1;
else
    F=0;
end
dx=[x(2); (F/m1)+(k/m1)*(x(3)-x(1)); x(4); -(k/m2)*(x(3)-x(1))];

```

donde se ha asumido que la entrada $F(t)$ es un pulso de amplitud unitaria y duración $D = 0,5$ seg., aplicado en $t = 0$ seg. Los comandos **Matlab** para la integración numérica de las ecuaciones y la visualización de los resultados serían

```

[t,x]=ode45('mass_spring_mass',[0 6],[0;0;0;0]);
subplot(411), plot(t,x(:,1));xlabel('Tiempo [s]');
ylabel('x1 = d1');grid;
subplot(412), plot(t,x(:,2));xlabel('Tiempo [s]');
ylabel('x2 = v1');grid;
subplot(413), plot(t,x(:,3));xlabel('Tiempo [s]');
ylabel('x3 = d2');grid;
subplot(414), plot(t,x(:,4));xlabel('Tiempo [s]');
ylabel('x4 = v2');grid;

```

La salida gráfica correspondiente se representa en la Figura 3.1.4.

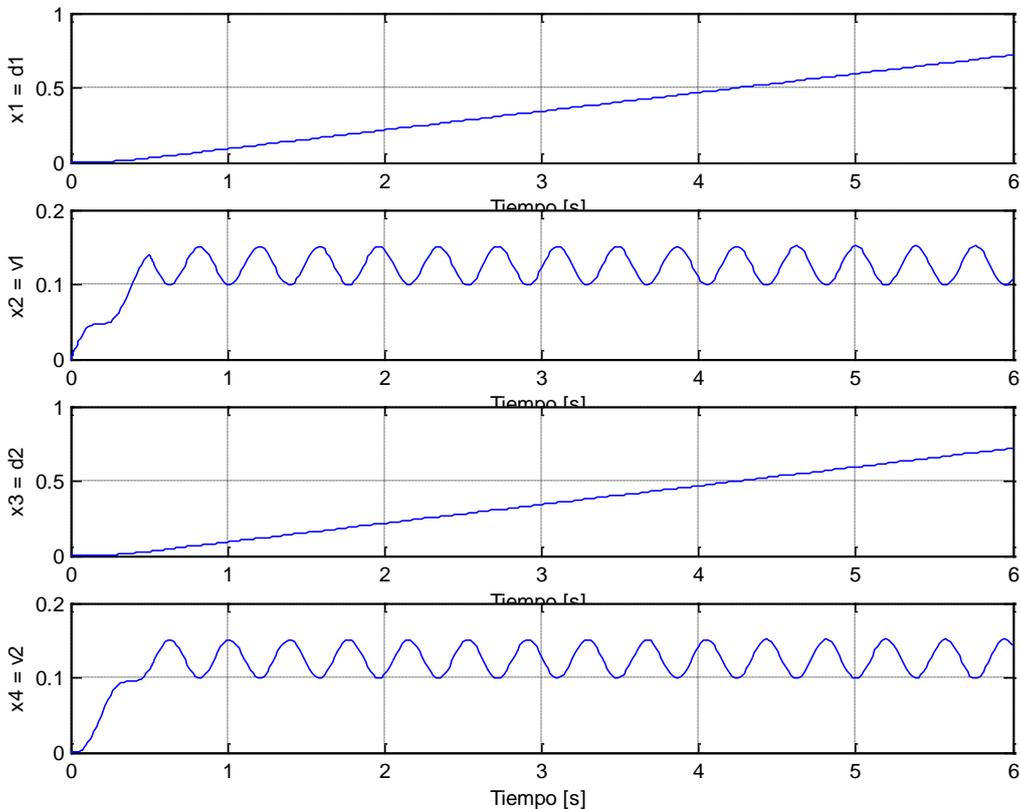


Figura 3.1.4: Salida Gráfica correspondiente al Ejemplo 2.

□

3.2. Representación de la Respuesta en Frecuencia de sistemas LE usando Matlab

En esta subsección, se describirán brevemente las rutinas **Matlab** que posibilitan la representación de sistemas lineales estacionarios, y el cómputo y la representación gráfica de la respuesta en frecuencia de los mismos. Estas funciones son parte del *Control System Toolbox* de **Matlab** [2]. El uso de los comandos se ilustrará con algunos ejemplos.

3.2.1. Representación de Sistemas Lineales Estacionarios en Matlab

El *Control System Toolbox* de **Matlab** ofrece básicamente tres formas distintas de especificar SLE. Estas formas son Función Transferencia, Modelo Cero-Polo-Ganancia, y Ecuaciones de Estado, que pueden ser generadas por las funciones `tf`, `zpk`, y `ss`, respectivamente. Estas funciones generan objetos que almacenan los datos en una única variable **Matlab**. Estos objetos pueden posteriormente ser manipulados por las otras funciones del *Control System Toolbox*. A continuación se describen las tres formas de representación mencionadas.

- **Función Transferencia (TF)**

Modelos de la forma

$$G(s) = \frac{N(s)}{D(s)}$$

donde $N(s)$ y $D(s)$ son polinomios en la variable s . Estos polinomios se representan en **Matlab** mediante vectores fila con los coeficientes del polinomio en orden decreciente de potencias de s . La función que permite generar estos modelos es `tf`. El siguiente ejemplo ilustra el uso de esta función.

Ejemplo 3: La siguiente secuencia de comandos

```
>> num=[1 0];  
>> den=[1 4 2];  
>> G=tf(num,den)
```

```
Transfer function:  
      s  
-----  
s^2 + 4s + 2
```

genera la función transferencia

$$G(s) = \frac{s}{s^2 + 4s + 2}$$

□

- **Modelos Cero-Polo-Ganancia (ZPK)**

Modelos de la forma

$$G(s) = k \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}$$

donde k es la ganancia, z_1, z_2, \dots, z_m son los ceros, y p_1, p_2, \dots, p_n son los polos de la función transferencia $G(s)$. La función que permite generar estos modelos es `zpk`. El siguiente ejemplo ilustra el uso de esta función.

Ejemplo 4: La siguiente secuencia de comandos

```
>> ceros=[0 -1];  
>> polos=[1-i 1+i 2];  
>> ganancia=4;  
>> G=zpk(ceros,polos,ganancia)
```

Zero/pole/gain:

4 s (s+1)

(s-2) (s^2 - 2s + 2)

genera la función transferencia

$$G(s) = \frac{4s(s+1)}{(s-2)(s^2 - 2s + 2)}$$

□

- **Modelos en Ecuaciones de Estado (SS)**

Modelos de la forma

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

donde x es el vector de estado, u es el vector de entradas, e y es el vector de salidas. La función que permite generar estos modelos es `ss`. El siguiente ejemplo ilustra el uso de esta función.

Ejemplo 5: Consideremos el sistema mecánico traslacional representado en la Figura 3.2.1. La ecuación diferencial que describe el comportamiento del sistema está dada por (el lector interesado puede fácilmente verificar esto)

$$x'' + \frac{b}{m}x' + \frac{K}{m}x = \frac{F}{m}$$

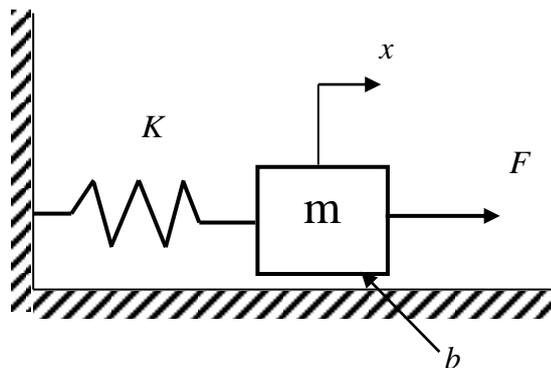


Figura 3.2.1: Sistema Mecánico Traslacional del Ejemplo 5.

Definiendo las variables de estado $x_1 = x$, $x_2 = x'$, la salida $y = x$, y la entrada $u = F$, las matrices A, B, C y D del modelo en Ecuaciones de Estado resultan

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{b}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = [1 \ 0], \quad D = 0$$

Asumiendo los valores $K = 2, b = 0.5, m = 1$, la siguiente secuencia de comandos

```
>> A=[0 1;-2 -0.5];  
>> B=[0;1];  
>> C=[1 0];  
>> D=0;  
>> G=ss(A,B,C,D)
```

```

a =
      x1      x2
x1      0      1.00000
x2     -2.00000 -0.50000
b =
      u1
x1      0
x2     1.00000
c =
      x1      x2
y1     1.00000  0
d =
      u1
y1      0
Continuous-time system.

```

genera el modelo en ecuaciones de estado del sistema.

□

3.2.2. Representación de la Respuesta en Frecuencia de SLE

El *Control System Toolbox* de **Matlab** presenta varias funciones para el cómputo y la visualización de la respuesta en frecuencia de Sistemas Lineales Estacionarios. Las funciones que se emplearán en el Trabajo Práctico son :

- `bode`, que permite la representación en forma de diagramas logarítmicos de amplitud y fase (Diagramas de Bode de Amplitud y Fase) de la respuesta en frecuencia $G(j\omega)$. Existen básicamente dos formas de invocar este comando.

- i. `[magnitud, fase, w]=bode(sistema)`
- ii. `[magnitud, fase]=bode(sistema, w)`

En ambos casos, la rutina devuelve a la salida la magnitud (en por unidad) y la fase (en grados) de la respuesta en frecuencia del sistema definido en `sistema` (con alguna de las rutinas descritas en la subsubsección 3.2.1.), calculada en las frecuencias contenidas en el vector `w`. En el caso **i.**, la rutina determina automáticamente el rango de frecuencias basándose en los polos y ceros del sistema. En el caso **ii.**, las frecuencias son especificadas por el usuario como un argumento de entrada. Si se omiten los argumentos de salida cuando se invoca la función, sólo se obtendrá la gráfica de los Diagramas de Bode de amplitud y fase.

- `nyquist`, que permite la representación en forma polar en el plano complejo ($\text{Re}\{G(s)\}, \text{Im}\{G(s)\}$) de la respuesta en frecuencia $G(j\omega)$. La forma de invocar esta función es completamente análoga a la correspondiente a la función `bode`.

A continuación se ilustrará el uso algunos de estos comandos a través de un ejemplo.

Ejemplo 6 La Figura 3.2.2 representa un circuito RLC donde la señal de entrada es senoidal de frecuencia variable ω_0 .

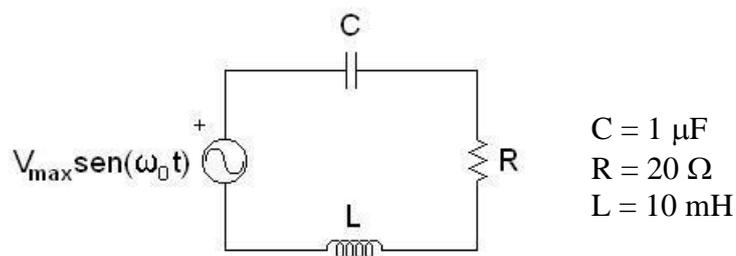


Figura 3.2.2: Circuito RLC

La función transferencia entre la tensión de entrada y la tensión en la resistencia resulta (verifique esta expresión):

$$H(s) = \frac{V_R(s)}{V(s)} = \frac{(R/L)s}{s^2 + (R/L)s + (1/LC)} = \frac{2\xi\omega_n s}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

donde:

$$\omega_n = \frac{1}{\sqrt{LC}} \quad \xi = \frac{R}{2} \sqrt{\frac{C}{L}}$$

La respuesta en frecuencia resulta (verifique esta expresión):

$$H(j\omega) = \frac{(R/L)j\omega}{\left(\frac{1}{LC} - \omega^2\right) + j(R/L)\omega}$$

La secuencia de comandos **Matlab** que permite computar y visualizar la respuesta en frecuencia $H(j\omega)$ del sistema es

```
>> num=2000;
>> den=[1 2000 1e8];
>> [mag, fase, w]=bode(num, den);
>> bode(num, den)
```

La salida gráfica correspondiente se representa en la Figura 3.2.3.

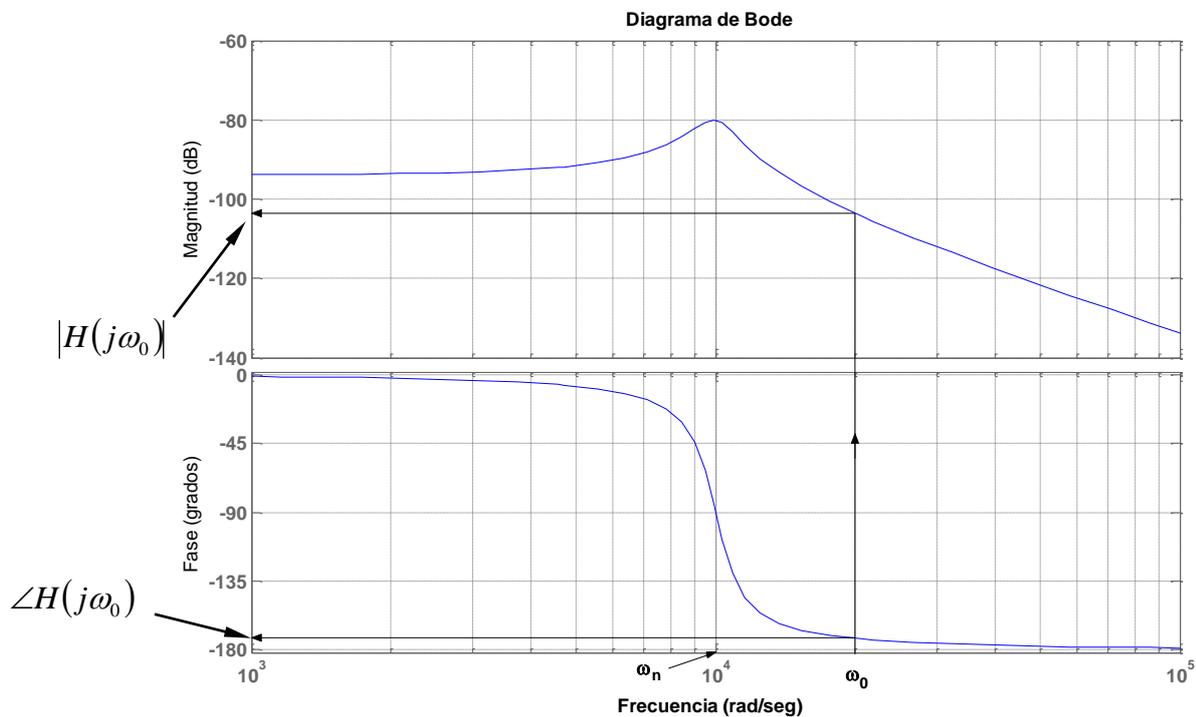


Figura 3.2.3: Diagramas de Bode de Amplitud y Fase correspondientes al Ejemplo 6.

Asumiendo que la fuente de Figura 3.2.2 tiene una frecuencia $\omega_0 = 2 \times 10^4$ rad/seg, como se indica en Figura 3.2.3, la respuesta del circuito en régimen permanente senoidal resulta

$$V_R^{RPS}(t) = |H(j\omega_0)| V_{\max} \text{sen}(\omega_0 t + \angle H(j\omega_0))$$

donde $|H(j\omega_0)|$ y $\angle H(j\omega_0)$ son los indicados en Figura 3.2.3.

3.3. Problemas Propuestos de Simulación durante la Sesión de Laboratorio

Problema 1 Sistema Mecánico Traslacional

Considere el sistema mecánico traslacional con dos masas y resorte presentado en el **Ejemplo 2**, pero ahora suponiendo que hay rozamiento de tipo viscoso (es decir con fuerza de rozamiento proporcional a la velocidad de la masa) en ambas masas, con constante de rozamiento $b = 8$ N/m (valor determinado experimentalmente).

- a. **Trabajo Previo:** Escriba el sistema de ecuaciones de primer orden que describe la dinámica del sistema en la forma matricial de ecuaciones de estado/ecuación de salida

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

donde $x = [d_1 \quad v_1 \quad d_2 \quad v_2]^T$ es el vector de estado, $u = F$ es la entrada, e $y = x_1$ es la salida del sistema, y donde A , B , C y D son las matrices del sistema de dimensiones adecuadas.

- b. Analice físicamente la respuesta del sistema a un pulso de amplitud $F = 1.48$ y duración $D = 0.2$ segundos, suponiendo condiciones iniciales nulas.
- c. Repita el apartado a., pero para condiciones iniciales $x_1 = 0, x_2 = 0.3$ m.
- d. Halle la función transferencia entre la entrada $u = F$ y la salida $y = x_1$. Analice la estabilidad entrada-salida del sistema.
- e. Obtenga una gráfica de los Diagramas de Bode de Amplitud y Fase para la transferencia hallada en d.
- f. A partir de una medición experimental realizada sobre el equipo de laboratorio ECP Model 210a con el mismo experimento del apartado b., se obtuvo la gráfica que se muestra en la Figura 3.3.1, para las posiciones de las masas m_1 y m_2 . Los datos de medición se encuentran en el archivo `datos_p1.m`. Tener en cuenta que la relación de cuentas es de 1cm=2266 cuentas.

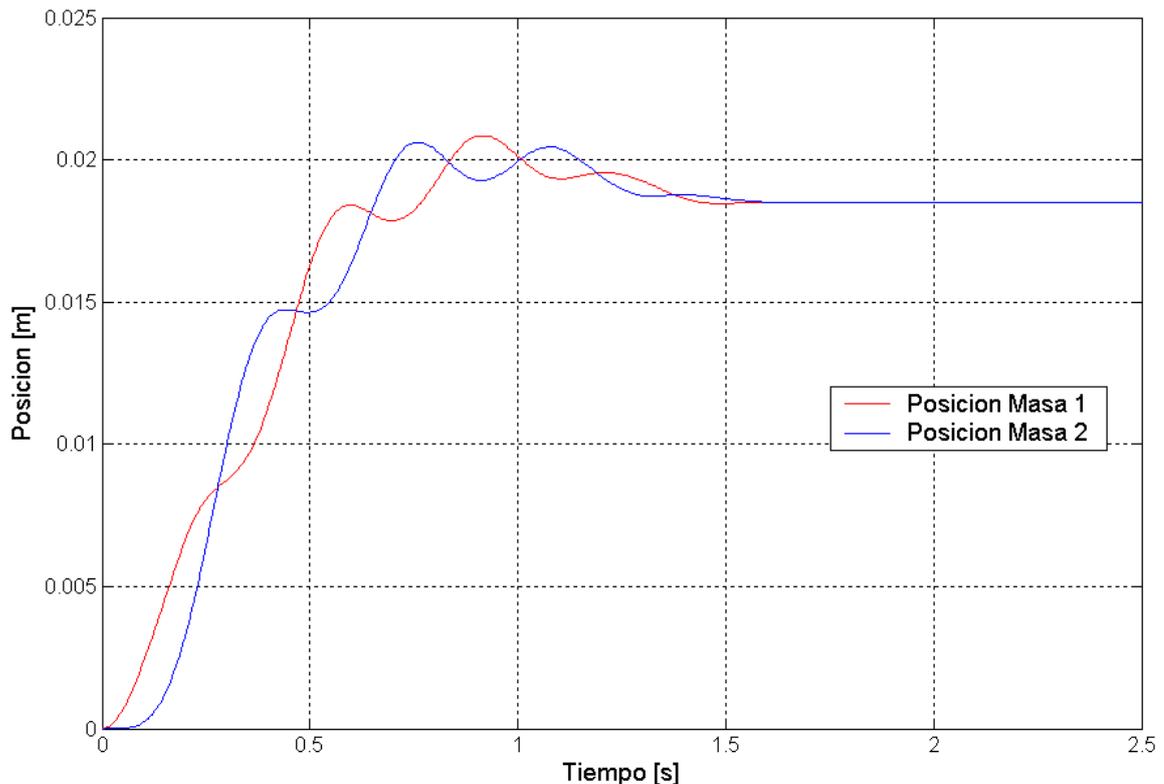


Figura 3.3.1: Gráfica de las posiciones de las masas obtenidas al experimentar con el equipo ECP modelo 210a (con $m_1 = 2$ kg, $m_2 = 2$ kg y tensión 0,25 V)

Compare los resultados experimentales con los obtenidos por simulación en el apartado b. Extraiga conclusiones respecto a posibles diferencias.

Problema 2: Carrito Minero

En la Figura 3.3.2 se muestra el diagrama esquemático de un carrito minero. El sistema consiste de un Motor de Corriente Continua cuyo eje se conecta a una carga mediante un sistema rueda-cremallera, y un cable y un sistema de poleas que modifican la dirección de la fuerza aplicada. La carga se encuentra apoyada sobre un plano inclinado a 30° . El motor provee la fuerza necesaria para compensar el peso de la carga y levantarla hasta una altura objetivo h . Se asume que el rozamiento en las poleas es nulo y que el cable no se deforma. El objetivo del experimento consiste en implementar un controlador que controle la altura h a la cual es movida la carga. El diagrama de cuerpo libre de la masa M se muestra en la Figura 3.3.3.

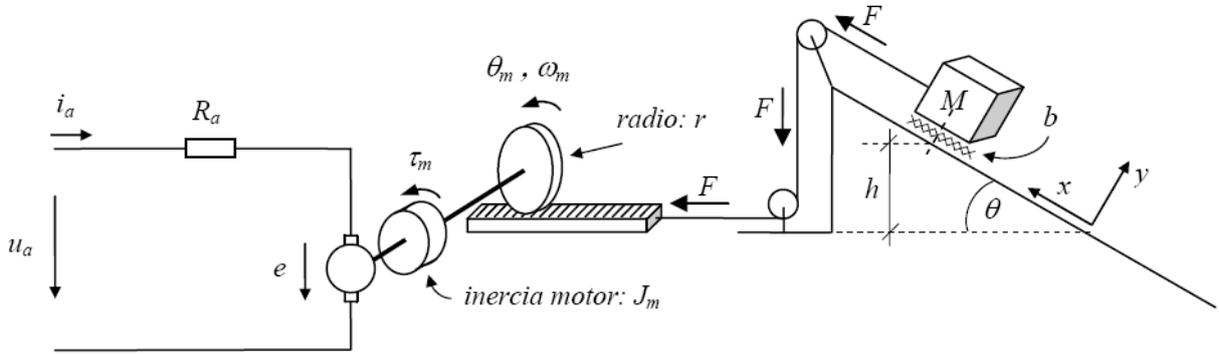


Figura 3.3.2: Diagrama Esquemático del Carrito Minero.

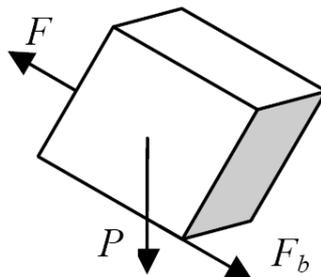


Figura 3.3.3: Diagrama de cuerpo libre de la masa M.

Análisis del sistema a lazo abierto

- a. Realice el Diagrama de Bloques tomando como entrada la tensión u_a del motor y la salida h , teniendo en cuenta que $g = 9.81 \text{ m/seg}^2$ es una entrada que corresponde a la aceleración de la gravedad.
- b. Calcule la función transferencia entre u_a y h . Observación: Para tal efecto pase la entrada g .
- c. Analice a partir de la función transferencia hallada el orden del sistema y la BIBO estabilidad.
- d. Asumiendo los siguientes valores de los parámetros, realice los Diagramas de Bode de Amplitud y Fase:

$\theta = 30^\circ$	$K_m = 0.767 \text{ Nm/A}$	$R = 2.6 \Omega$
$r = 0.01 \text{ m}$	$J_m = 3.87 \times 10^{-7} \text{ Kg m}^2$	$M = 3 \text{ Kg}$
$b = 8 \text{ N/m}$	$h_{max} = 10 \text{ m}$	

Realice el modelo en Simulink del Diagrama de Bloques del punto a. utilizando para la entrada de tensión un escalón de amplitud A y un escalón de amplitud g para la aceleración de la gravedad, aplicados ambos en el instante 0. Verifique que se cumple lo determinado en c. Interprete físicamente la salida gráfica obtenida. Observación: Considere $\text{sen}(\theta) = c$ (constante).

Análisis del sistema a lazo cerrado

e. Para que la altura h pueda seguir una referencia se propone implementar un control PD (Proporcional+Derivativo) de la forma:

$$u_a = K_p (h_{ref} - h) + K_d \dot{h}$$

Las ganancias K_p , K_d del controlador deben elegirse de manera que la transferencia en lazo cerrado entre la entrada altura de referencia y la salida altura de la carga h tenga un factor de amortiguamiento $\xi = 2$ y una frecuencia natural $\omega_n = 22.16$ rad/seg.

f. Si se quiere subir el carrito hasta el final de la rampa ($h_{max} = 10$ m), encuentre el rango de variación de ξ para evitar que el carrito se caiga. Justifique su respuesta.

g. Haga los Diagramas de Bode de Amplitud y Fase del sistema retroalimentado.

h. Compute y grafique la respuesta del sistema en lazo cerrado a un escalón unitario en la altura de referencia.

4. Referencias

- [1] The MathWorks, Inc. (1997). *Matlab, The Language of Technical Computing – Getting Started with Matlab, Version 5.*, 24 Prime Park Way, Natick, MA 01760-1500.
- [2] The MathWorks, Inc. (1996). *Matlab, Control System Toolbox – User’s Guide, Version 4*, 24 Prime Park Way, Natick, MA 01760-1500.
- [3] Park, T. (1991). *Manual for Model 210a: Rectilinear Control System*, ECP, CA, USA, www.ecpsystems.com.
- [4] Perko, L. (1991). *Differential Equations and Dynamical Systems*, Springer Verlag, Berlin.