



Universidad Nacional de Rosario

FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA

A3 - Sistemas y señales I

Trabajo práctico N°2:

SIMULACIÓN DIGITAL DE SISTEMAS DINÁMICOS

Asistencia al laboratorio: *Lunes 8 de Mayo 2023*

Fecha de entrega: *Lunes 12 de Junio 2023*

Federico, Scheytt S-5268/1
Joaquín, Gonzalez Targon G-5767/3

Índice

Introducción	2
1. Sistema mecánico rotacional	3
1.1. Sistema de ecuaciones que describe la dinámica del sistema	3
1.2. Interpretación física de la respuesta del sistema	5
1.3. Función transferencia y estabilidad del sistema	7
1.4. Diagrama de Bode	8
1.5. Polos y ceros de la función transferencia	9
1.6. Comparación con datos obtenidos experimentalmente	9
2. Sistema de levitación magnética	11
2.1. Primer modelo del sistema	11
2.1.1. Ecuaciones diferenciales que gobiernan la dinámica del sistema . . .	11
2.1.2. Diagrama de bloques del sistema	12
2.1.3. Valores de equilibrio	12
2.1.4. Implementación del diagrama de bloques en Simulink	13
2.1.5. Gráficas de las variables $u(t)$, $i(t)$ y $y(t)$	14
2.2. Modelo incremental lineal del sistema	15
2.2.1. Implementación del diagrama de bloques en Simulink	16
2.2.2. Gráficas de las variables $u(t)$, $i(t)$ y $y(t)$ del modelo linealizado . . .	16
2.2.3. Comparación entre los resultados de los dos modelos	18
2.2.4. Función transferencia entre la entrada y la salida incremental	20
2.2.5. Diagrama de Bode	22
2.3. Implementación de control proporcional P	23
2.3.1. Implementación del sistema de lazo cerrado en Simulink	24
Conclusión	26

Introducción

En este trabajo práctico se plantean dos problemas cuya resolución y planteamiento se dan a partir de conceptos teóricos adquiridos en el cursado de la asignatura y mediante el uso de herramientas de cómputo numérico como lo es Matlab (Matrix Laboratory). El programa Matlab (correspondiente a la abreviación de Matrix Laboratory) es una potente herramienta de cálculo numérico y visualización de uso muy difundido a nivel mundial en los ambientes industrial y académico en tareas de investigación, desarrollo y diseño en diversas áreas tales como procesamiento de señales, control y comunicaciones.

En el primer problema se nos presenta un sistema mecánico rotacional que permite varias configuraciones a partir de la conexión, mediante resortes de torsión de distintas masas, donde además se cuenta con un servo motor que permite simular diferentes torques aplicados al sistema. Para este sistema realizaremos un modelo teórico de su comportamiento, interpretaremos los resultados obtenidos y los compararemos con datos obtenidos experimentalmente bajo las mismas condiciones en las cuales realizaremos los cálculos.

En el segundo problema tenemos un sistema de suspensión magnético de una esfera diamagnética la cual debe de quedar suspendida a una determinada distancia del dispositivo, esto se logra a través de una fuerza magnética proporcional a la corriente del circuito eléctrico del sistema. Para este problema plantearemos dos modelos, uno no lineal y uno lineal, evaluaremos cuales son sus correspondientes respuestas, realizaremos simulaciones mediante el uso de la herramienta Simulink y finalmente implementaremos un control proporcional al mismo.

1. Sistema mecánico rotacional

El modelo 205a de ECP que se encuentra en el Laboratorio de Automatización y Control (LAC), y que se muestra en la Fig. 1(a), es un sistema mecánico rotacional que permite varias configuraciones a partir de la conexión, mediante resortes de torsión, de hasta 3 discos (cuyas masas pueden variarse) y un servo motor que permite simular diferentes torques aplicados al sistema.

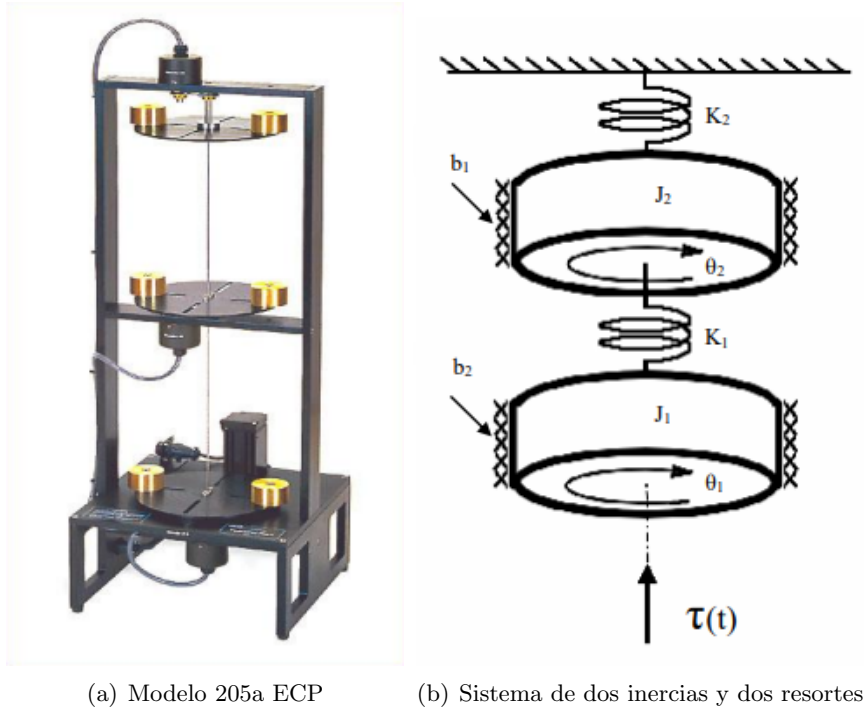


Figura 1: Sistema mecánico rotacional

Si consideramos la configuración particular del sistema mecánico rotacional cuyo diagrama se muestra en la Fig. 1(b) vemos que el mismo consta de dos inercias y dos resortes, con rozamiento de tipo viscoso en cada una de ellas.

1.1. Sistema de ecuaciones que describe la dinámica del sistema

Para poder encontrar las ecuaciones físicas que gobiernan el comportamiento del sistema realizamos la sumatoria de torques utilizando la segunda ley de Newton para mecánicas rotacionales.

$$\sum \tau_1 = \tau(t) - b_2 \dot{\theta}_1(t) - k_1(\theta_1(t) - \theta_2(t)) = J_1 \ddot{\theta}_1(t)$$

$$\ddot{\theta}_1(t) = \frac{1}{J_1} \left(\tau(t) - b_2 \dot{\theta}_1(t) - k_1(\theta_1(t) - \theta_2(t)) \right)$$

$$\sum \tau_2 = k_1(\theta_1(t) - \theta_2(t)) - b_1 \dot{\theta}_2(t) - k_2 \theta_2(t) = J_2 \ddot{\theta}_2(t)$$

$$\ddot{\theta}_2(t) = \frac{1}{J_2} \left(k_1(\theta_1(t) - \theta_2(t)) - b_1 \dot{\theta}_2(t) - k_2 \theta_2(t) \right)$$

Ahora escribiendo estas ecuaciones diferenciales de segundo orden como un conjunto de ecuaciones diferenciales de primer orden equivalentes de la forma:

$$\begin{cases} \dot{x}_1(t) = \theta_1(t) \\ \dot{x}_2(t) = \dot{x}_1(t) \\ \dot{x}_3(t) = \theta_2(t) \\ \dot{x}_4(t) = \dot{x}_3(t) \end{cases}$$

Por lo tanto, obtenemos:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{1}{J_1} (\tau(t) - b_2 x_2(t) - k_1(x_1(t) - x_3(t))) \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = \frac{1}{J_2} (k_1(x_1(t) - x_3(t)) - b_1 x_4(t) - k_2 x_3(t)) \end{cases} \quad (1)$$

Para obtener la representación en espacio de estados de un sistema es necesario obtener las siguientes dos ecuaciones matriciales:

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

donde la matriz A es la llamada matriz de transición, la matriz B es la matriz de entradas, la matriz C es la matriz de salida y la matriz D es la matriz de transferencia directa.

Ahora podemos reordenar el sistema de ecuaciones 1 de la siguiente forma:

$$\begin{cases} \dot{x}_1(t) = 0 \cdot x_1(t) + 1 \cdot x_2(t) + 0 \cdot x_3(t) + 0 \cdot x_4(t) + 0 \cdot \tau(t) \\ \dot{x}_2(t) = (\frac{-k_1}{J_1}) \cdot x_1(t) + (\frac{-b_2}{J_1}) \cdot x_2(t) + (\frac{k_1}{J_1}) \cdot x_3(t) + 0 \cdot x_4(t) + (\frac{1}{J_1}) \cdot \tau(t) \\ \dot{x}_3(t) = 0 \cdot x_1(t) + 0 \cdot x_2(t) + 0 \cdot x_3(t) + 1 \cdot x_4(t) + 0 \cdot \tau(t) \\ \dot{x}_4(t) = (\frac{k_1}{J_2}) \cdot x_1(t) + 0 \cdot x_2(t) + (\frac{-k_1-k_2}{J_2}) \cdot x_3(t) + (\frac{-b_1}{J_2}) \cdot x_4(t) + 0 \cdot \tau(t) \end{cases}$$

Donde ahora dicho sistema puede representarse como:

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_1}{J_1} & \frac{-b_2}{J_1} & \frac{k_1}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_1}{J_2} & 0 & \frac{-k_1-k_2}{J_2} & \frac{-b_1}{J_2} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \end{bmatrix}}_B \underbrace{\tau(t)}_{u(t)}$$

Podemos ver que además tomando como salida $y(t) = x_1(t)$, la misma puede representarse como:

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_D \underbrace{\tau(t)}_{u(t)}$$

Por lo tanto vemos que así quedan definidas las matrices A , B , C y D , con sus correspondientes dimensiones.

1.2. Interpretación física de la respuesta del sistema

Nos proponemos a continuación interpretar físicamente la respuesta del sistema a un pulso de amplitud $\tau = 0,5Nm$ y una duración $D = 4s$, suponiendo condiciones iniciales nulas, esto lo hacemos mediante la simulación de nuestro modelo utilizando rutinas ODE de Matlab.

Para resolver el sistema de ecuaciones diferenciales presentado anteriormente haremos uso de una rutina de integración numérica, en particular de `ode45()`. Para implementar este algoritmo creamos un ODE-file, tal como se muestra a continuación.

```
function dx = P1_function_ode_file(t,x)
    % P1_function_ode_file(t,x)
    % Author: Federico Scheytt - Joaquin Gonzalez Targon
    % Date: Junio 2023

    % Carga de datos
    P1_datos

    % Entrada
    if t < D
        T = 0.5;
    else
        T = 0;
    end

    % Ecuaciones de estado
    dx = [ x(2);
           (1/J1)*(T - b2*x(2) - k1*(x(1) - x(3)));
           x(4);
           (1/J2)*(-b1*x(4) - k1*(x(3) - x(1)) - k2*x(3));
        ];
```

Figura 2: ODE-file

Decidimos además crear un archivo que contenga los datos necesarios para la implementación de este sistema, el cual se muestra en la siguiente figura.

```
% Script: P1_datos.m
% Author: Federico, Scheytt - Joaquin, Gonzalez Targon
% Date: Junio 2023

% Datos
J1 = 0.009;
J2 = 0.009;
b1 = 0.02;
b2 = 0.02;
k1 = 2.7114;
k2 = 2.7114;

D = 4;
```

Figura 3: Datos del problema

Luego realizamos un script que implemente la rutina de integración numérica mencio-

nada anteriormente, y haga uso del ODE-file.

```
% Script: P1_script.m
% Author: Federico, Scheytt - Joaquin, Gonzalez Targon
% Date: Junio 2023

clear, clc, close all

% Carga de datos
P1_datos

% Inicializaciones
t = 10;
tspan = [0 t];
x0 = [0,0,0,0];

% Rutina de integracion numerica
[T,x] = ode45('P1_function_ode_file',tspan,x0);
```

Figura 4: Implementación del solver

Ahora para mostrar los resultados calculados, realizamos las gráficas correspondientes de la siguiente forma.

```
% Graficos
figure(1)
subplot(2,2,1), plot(T,x(:,1),'r'),
    title('Posicion de la masa 1 (\theta_1(t))')
    xlabel('Tiempo [seg]')
    ylabel('x_1(t)=\theta_1(t) [rad]')
    hold on, grid on
subplot(2,2,2), plot(T,x(:,2),'b')
    title('Velocidad de la masa 1 (\omega_1(t))')
    xlabel('Tiempo [seg]')
    ylabel('x_2(t)=\omega_1(t) [rad/s]')
    hold on, grid on
subplot(2,2,3), plot(T,x(:,3),'r')
    title('Posicion de la masa 2 (\theta_2(t))')
    xlabel('Tiempo [seg]')
    ylabel('x_3(t)=\theta_2(t) [rad]')
    hold on, grid on
subplot(2,2,4), plot(T,x(:,4),'b')
    title('Velocidad de la masa 2 (\omega_2(t))')
    xlabel('Tiempo [seg]')
    ylabel('x_4(t)=\omega_2(t) [rad/s]')
    hold on, grid on
```

Figura 5: Gráficos

Finalmente obtenemos como resultado:

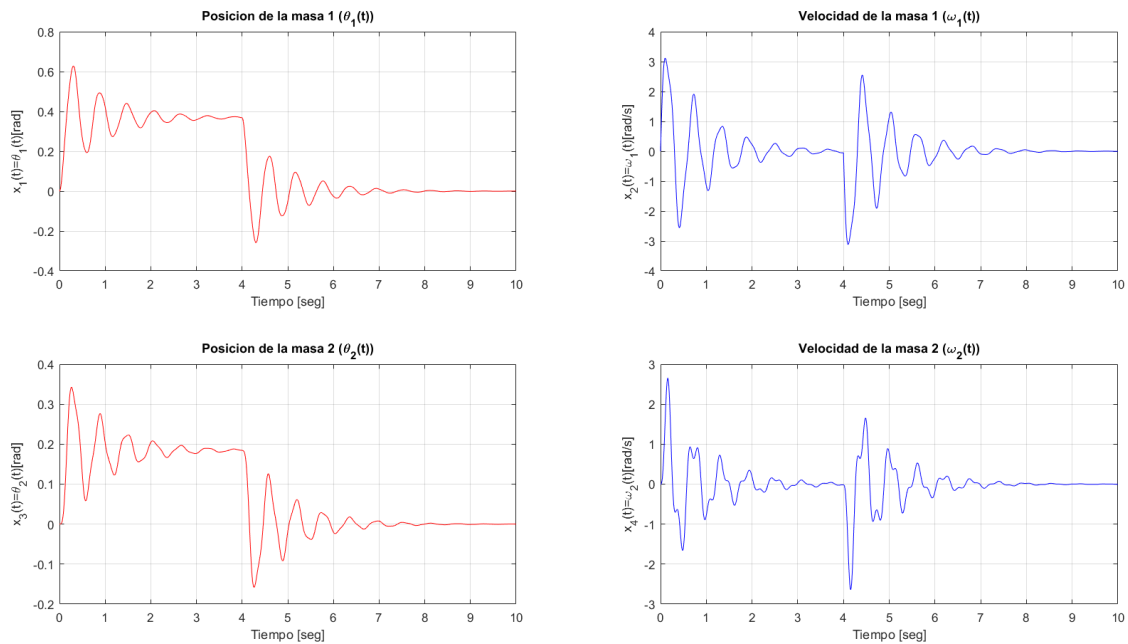


Figura 6: Gráficos del comportamiento del sistema

Vemos entonces que cuando se le aplica al sistema un pulso de duración finita con condiciones iniciales nulas, lo que ocurre es que el servo motor genera un desplazamiento de la masa 1 junto con una torsión del resorte que esta conectado a dicha masa, que a su vez esta conectado a la masa 2. Esto va a producir un desequilibrio en el sistema, donde las masas van a rotar de manera oscilatoria.

A medida que va evolucionando la posición de la masa 1 vemos que la misma tiende a estabilizarse en un valor concreto. Este comportamiento genera una torsión en el resorte que conecta con la masa 2, la cual hace que también gire, y que evolucione de una forma similar hasta lograr un equilibrio.

Una vez pasado los 4 segundos que dura el escalón, la fuerza externa desaparece y ambas masas tienden a la posición inicial, pero sus evoluciones se van a ver afectadas por las fuerzas de torsión de los resortes, como se puede ver ambas masas oscilan en la posición inicial y luego se estabilizan en cero.

1.3. Función transferencia y estabilidad del sistema

Para encontrar la función transferencia hacemos uso de las herramientas que nos brinda Matlab e implementamos el siguiente script para dicho fin.

```
% Matrices
A = [0,1,0,0; -k1/J1,-b2/J1,k1/J1,0; 0,0,0,1; k1/J2,0,(-k1-k2)/J2,-b1/J2];
B = [0;1/J1;0;0];
C = [1,0,0,0];
D = 0;

% Funcion transferencia
[num,den] = ss2tf(A,B,C,D);
G = tf(num,den);
```

Figura 7: Calculo de función transferencia

De lo cual obtenemos:

```
G =

      111.1 s^2 + 246.9 s + 6.695e04
-----
s^4 + 4.444 s^3 + 908.7 s^2 + 2008 s + 9.076e04

Continuous-time transfer function.
```

Figura 8: Función transferencia

Si analizamos las raíces del polinomio denominador encontraremos los polos de la función transferencia, para este caso dichas raíces son números complejos conjugados con parte real negativa con lo cual podemos concluir que el sistema es BIBO estable (Bounded Input – Bounded Output).

1.4. Diagrama de Bode

Realizamos además el diagrama de Bode correspondiente a nuestro sistema implementando el siguiente script para dicho fin.

```
% Diagrama de bode
figure(2)
bode(num,den), grid on
```

Figura 9: Diagrama de Bode

De lo cual obtenemos:

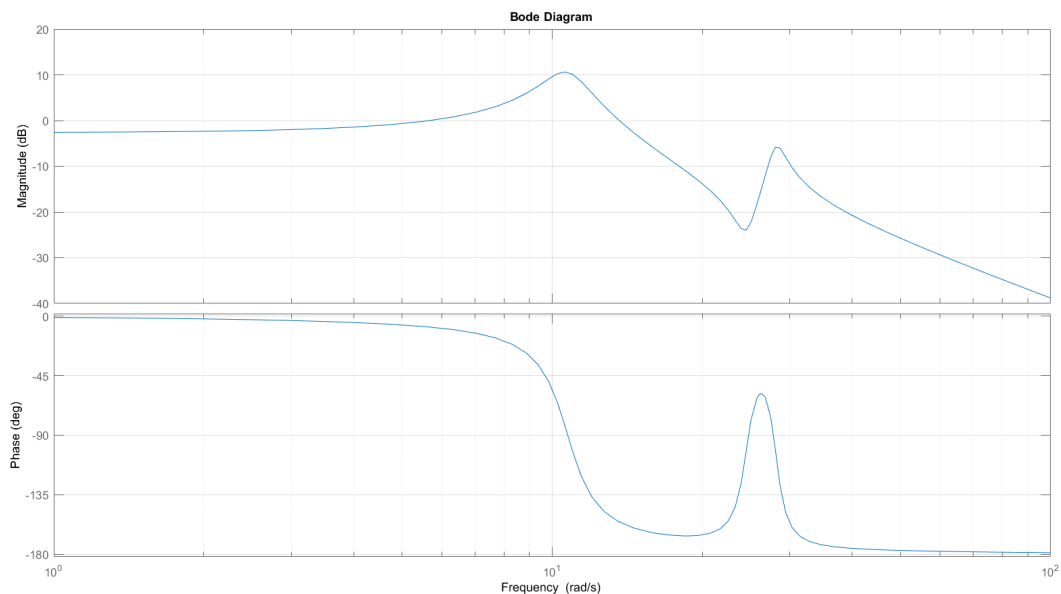


Figura 10: Diagrama de Bode

1.5. Polos y ceros de la función transferencia

Para calcular los polos y ceros de la función transferencia realizamos el siguiente script.

```
% Polos y ceros
polos = roots(den);
ceros = roots(num);
```

Figura 11: Calculo de polos y ceros

De lo cual obtenemos:

<pre>polos =</pre> <pre>-1.1111 +28.0623i</pre> <pre>-1.1111 -28.0623i</pre> <pre>-1.1111 +10.6695i</pre> <pre>-1.1111 -10.6695i</pre>	<pre>ceros =</pre> <pre>-1.1111 +24.5214i</pre> <pre>-1.1111 -24.5214i</pre>
(a) Polos	(b) Ceros

Figura 12: Polos y ceros de la función transferencia

Donde vemos que todos los polos poseen parte real negativa, y como dijimos anteriormente el sistema sera BIBO estable. Además podemos ver que los ceros de la función transferencia también presenta parte real negativa, por lo que el sistema también sera de mínima fase.

En el diagrama de Bode vemos que el mismo presenta picos y valles. El primer pico se da a la frecuencia $\omega_n = 10,7rad/s$, la cual se corresponde con nuestro segundo par de polos complejos conjugados. Luego, nos encontramos con un valle en $\omega_n = 24,5rad/s$ el cual se corresponde con el par de ceros complejos conjugados que obtuvimos. Finalmente nos encontramos con un ultimo pico que se da en $\omega_n = 28,1rad/s$ que corresponde con el primer par de polos complejos conjugados obtenidos.

Respecto a la fase, notamos que la misma va a tender a -180 en para el primer par de polos complejos pasando por -90 en $\omega_n = 10,7rad/s$. Luego el par de ceros complejos conjugados produce que la fase tienda de -180 a 0 pasando por -90 en $\omega_n = 24,5rad/s$. Finalmente el ultimo par de polos complejos produce una tendencia igual al primero pasando por -90 en $\omega_n = 28,1rad/s$.

1.6. Comparación con datos obtenidos experimentalmente

A partir de mediciones experimentales realizadas sobre el equipo de laboratorio, con las mismas condiciones iniciales con las cuales se realizaron los cálculos teóricos, proponemos compararlos con los resultados obtenidos mediante la simulación de la respuesta del sistema que obtuvimos con Matlab. A continuación se muestra el script que utilizamos para este fin.

```

% Carga de datos experimentales
Pl_datos_experimentales

% Graficos comparativos
figure(3)
subplot(2,1,1), plot(T,x(:,1),'r',evol(:,2), evol(:,4)/2544, 'b'),
    title('Posicion de la masa 1 (\theta_1(t))')
    legend('Teorica','Experimental')
    xlabel('Tiempo [seg]')
    ylabel('x_1(t)=\theta_1(t) [rad]')
    hold on, grid on
subplot(2,1,2), plot(T,x(:,3),'r',evol(:,2), evol(:,5)/2544, 'b')
    title('Posicion de la masa 2 (\theta_2(t))')
    legend('Teorica','Experimental')
    xlabel('Tiempo [seg]')
    ylabel('x_3(t)=\theta_2(t) [rad]')
    hold on, grid on

```

Figura 13: Comparación de resultados

De lo cual obtuvimos:

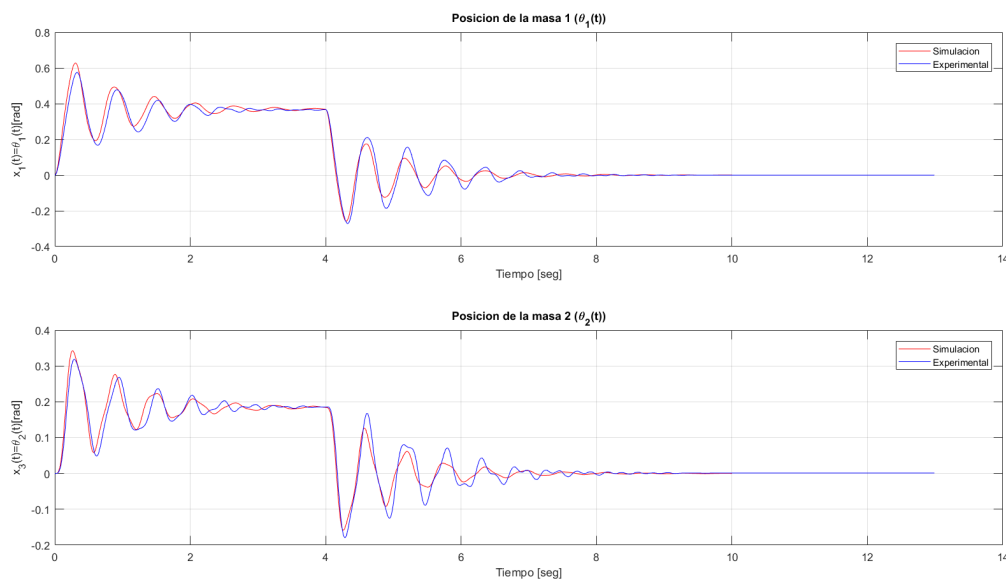


Figura 14: Comparación de resultados

Podemos notar que los resultados obtenidos son similares a las mediciones realizadas. Las discrepancias observadas entre ambas gráficas pueden estar relacionadas a que estamos utilizando un modelo del sistema, en el cual se realizan muchas simplificaciones y no se tienen en cuenta muchos factores que pueden afectar el comportamiento del mismo, como lo son la temperatura, las constantes relacionadas a los componentes, entre otras. Además mediante la implementación de una solución numérica se comenten errores numéricos, los cuales se acumulan en la integración que estamos realizando y pueden producir una discrepancia con el comportamiento real del sistema. Sin embargo, a pesar de todas estas consideraciones, podemos ver que el comportamiento general del dispositivo es muy similar en ambos casos y podemos decir que se trata de una buena aproximación del comportamiento real del mismo.

2. Sistema de levitación magnética

Para este problema contamos con un sistema de suspensión magnética de una esfera diamagnética. El objetivo de este sistema es lograr que la esfera diamagnética quede suspendida a una distancia $y(t)$ determinada del extremo del electroimán, a través de la regulación de la corriente $i(t)$ por el mismo mediante el control de la tensión de alimentación $u(t)$, como se muestra en la siguiente figura.

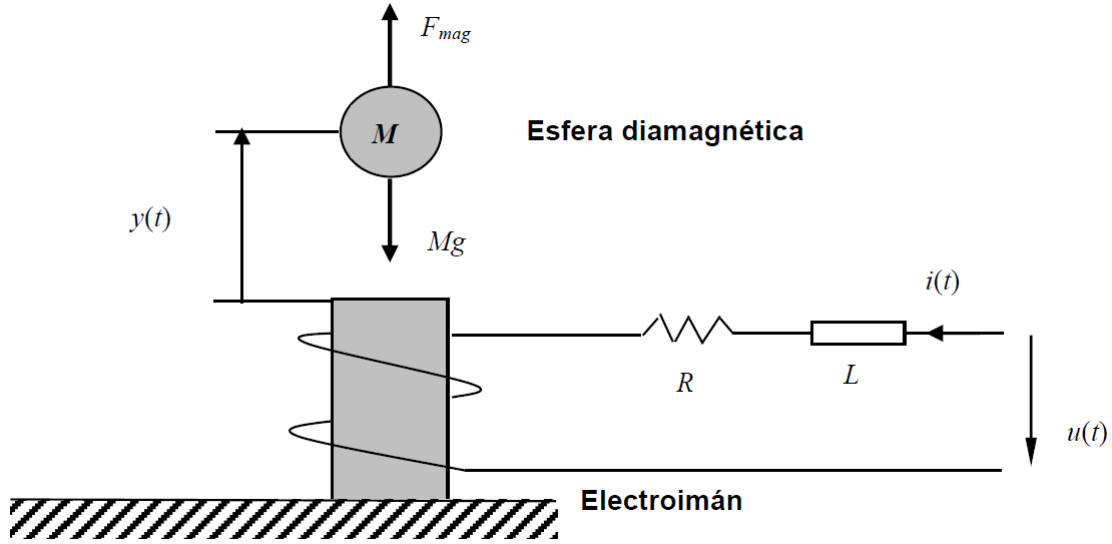


Figura 15: Sistema de suspensión magnética

La fuerza magnética de la esfera puede modelarse como:

$$F_{mag} = \frac{ki(t)^2}{y(t)}$$

2.1. Primer modelo del sistema

2.1.1. Ecuaciones diferenciales que gobiernan la dinámica del sistema

Debemos notar que este sistema está compuesto por dos subsistemas, los cuales se encuentran vinculados por la corriente $i(t)$.

El primer subsistema se encuentra representado por la esfera que se quiere suspender magnéticamente, el cual está regido por las leyes de Newton, por lo cual lo modelamos como se muestra a continuación.

$$\sum F(t) = F_{mag}(t) - F_p(t) - F_b(t) = M\ddot{y}(t)$$

$$\sum F(t) = \frac{ki(t)^2}{y(t)} - Mg - b\dot{y}(t) = M\ddot{y}(t)$$

Por lo tanto:

$$\ddot{y}(t) = \frac{1}{M} \left(\frac{ki(t)^2}{y(t)} - Mg - b\dot{y}(t) \right) \quad (2)$$

El segundo subsistema se encuentra representado por la parte eléctrica del sistema, el cual se encuentra regido por las leyes de Kirchhoff, por lo cual lo modelamos como se muestra a continuación.

$$\sum U(t) = u_L(t) + u_R(t) - u(t) = 0$$

$$\sum U(t) = L\dot{i}(t) + Ri(t) - u(t) = 0$$

Por lo tanto:

$$\dot{i}(t) = \frac{1}{L} (u(t) - Ri(t)) \quad (3)$$

Por lo tanto las ecuaciones 2 y 3 representan el comportamiento del sistema.

2.1.2. Diagrama de bloques del sistema

A partir de las ecuaciones 2 y 3 realizamos el siguiente diagrama de bloques del sistema, considerando como entrada la tensión $u(t)$ y como salida la posición $y(t)$.

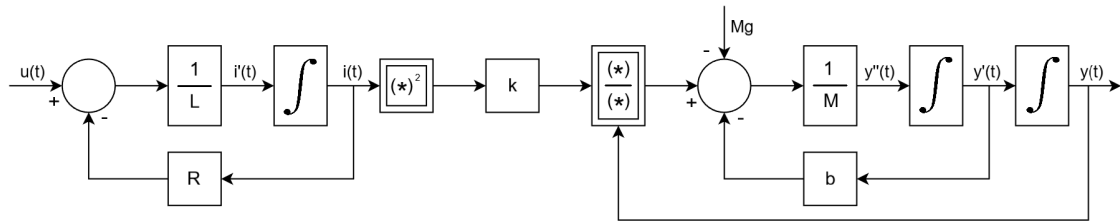


Figura 16: Diagrama de bloques

Podemos ver que en el diagrama de bloques aparecen dos bloques no lineales.

2.1.3. Valores de equilibrio

Ahora nos proponemos calcular los valores de equilibrio \bar{i} y \bar{y} de las variables $i(t)$ e $y(t)$ respectivamente, correspondientes a una tensión de alimentación constante $u(t) = \bar{u} = 10V$.

Vemos entonces que en el equilibrio la tensión es constante, por lo que además la masa M permanecerá inmóvil, por lo que tanto su aceleración como su velocidad serán nulas.

$$\sum F_{eq}(t) = F_{mag}(t) - F_p(t) - \cancel{F_b(t)} = M \cancel{\ddot{y}(t)} \rightarrow 0$$

$$\sum F_{eq}(t) = \frac{k\bar{i}^2}{\bar{y}} - Mg = 0$$

Por lo tanto:

$$\bar{y} = \frac{k\bar{i}^2}{Mg} \quad (4)$$

Desde el punto de vista eléctrico podemos ver que la corriente que circula por el circuito es constante, por lo tanto no se produce una caída de tensión en el inductor, es decir se comporta como un cortocircuito.

$$\sum U_{eq}(t) = \cancel{u_L(t)} + u_R(t) - \bar{u} = 0$$

$$\sum U_{eq}(t) = R\bar{i} - \bar{u} = 0$$

Por lo tanto:

$$\bar{i} = \frac{\bar{u}(t)}{R} \quad (5)$$

Ahora remplazando en las ecuaciones 4 y 5 por los datos correspondientes obtenemos los valores de equilibrio del sistema para la tensión dada.

$$\begin{cases} \bar{i} = 0,5A \\ \bar{y} = 0,2551m \end{cases}$$

2.1.4. Implementación del diagrama de bloques en Simulink

Primero comenzamos realizando el diagrama de bloques en el entorno de trabajo de Simulink, como se muestra en la siguiente figura.

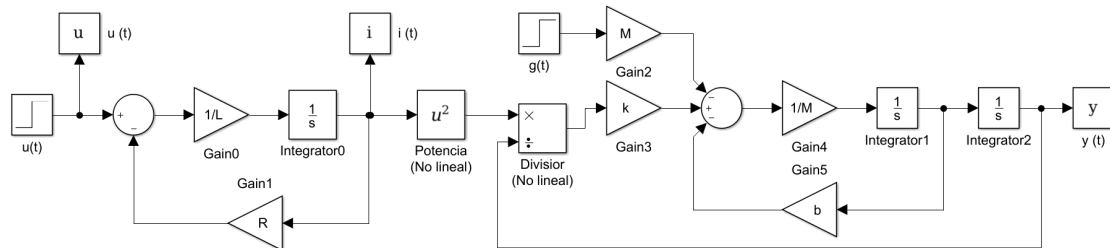


Figura 17: Diagrama de bloques en Simulink

Para implementar nuestro diagrama de bloques en Simulink además creamos un script que almacene las constantes necesarias para la simulación.

```
% Script: P2_datos.m
% Author: Federico, Scheytt - Joaquin, Gonzalez Targon
% Date: Junio 2023

% Datos
R = 20;
b = 0.5;
L = 0.01;
M = 0.2;
g = 9.8;
k = 2;
```

Figura 18: Datos del problema

2.1.5. Gráficas de las variables $u(t)$, $i(t)$ y $y(t)$

Para realizar los gráficos correspondientes realizamos el siguiente script.

```
% Script: P2_script.m
% Author: Federico, Scheytt - Joaquin, Gonzalez Targon
% Date: Junio 2023

clc, close all

% Carga de datos
P2_datos

% Implementacion del diagrama de bloques no lineal
% Graficas
figure(1)
subplot(3,1,1), plot(y.time,y.data,'b'), grid on, hold on;
title('Posicion de la masa M (y(t))');
xlabel('Tiempo [seg]')
ylabel('y(t)')
subplot(3,1,2), plot(i.time,i.data,'b'), grid on, hold on;
title('Corriente (i(t))')
xlabel('Tiempo [seg]')
ylabel('i(t)')
subplot(3,1,3), plot(u.time,u.data,'b'), grid on, hold on;
title('Tension (u(t))')
xlabel('Tiempo [seg]')
ylabel('u(t)')
```

Figura 19: Gráficos

De la simulación del diagrama de bloques antes mencionado obtenemos los siguientes resultados.

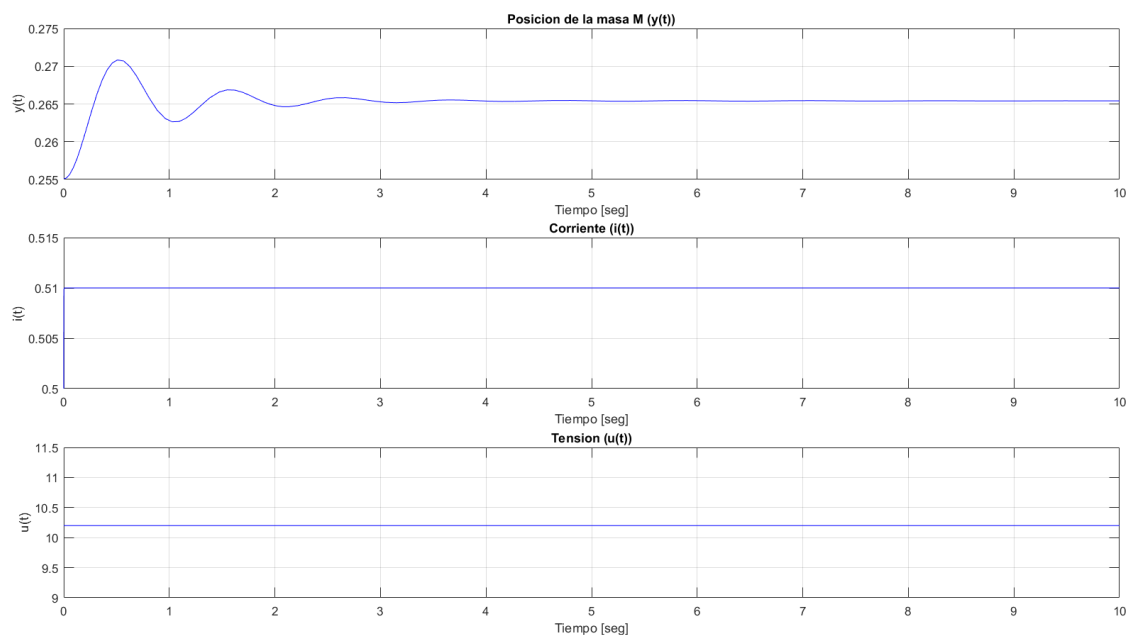


Figura 20: Gráficos para una entrada de tensión de 0,2V a partir del valor de equilibrio

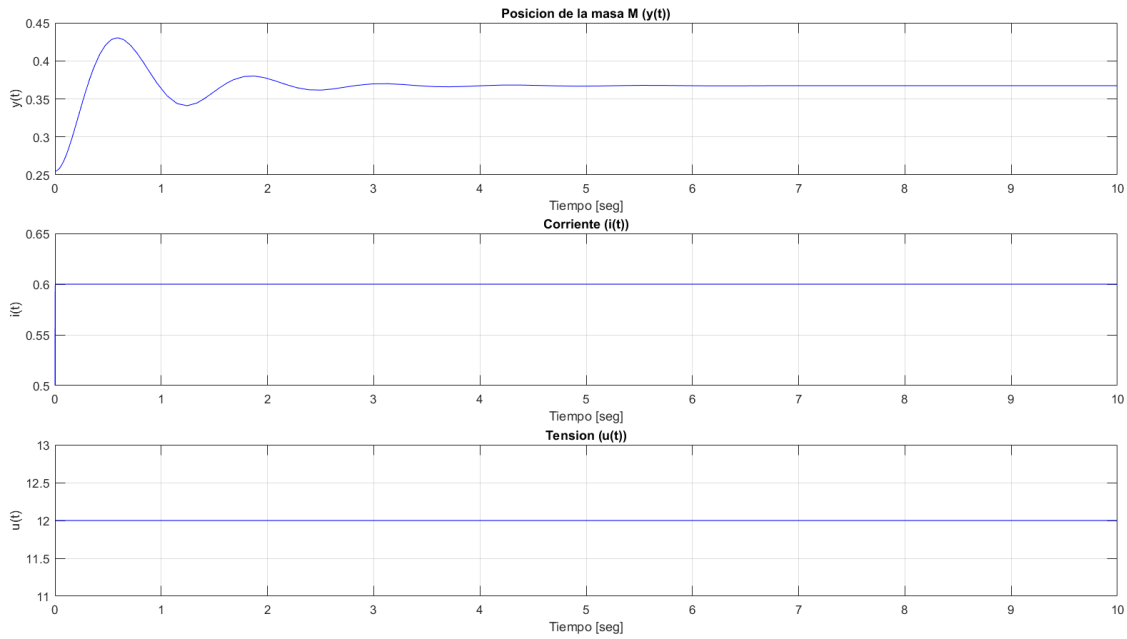


Figura 21: Gráficos para una entrada de tensión de 2V a partir del valor de equilibrio

Podemos ver entonces que el sistema presenta una evolución temporal del tipo subamortiguado, donde vemos que nuestra salida oscila en el valor de equilibrio, hasta que para un determinado tiempo se vuelve constante. Con la diferencia de que dicho valor de equilibrio varía según la entrada que apliquemos.

2.2. Modelo incremental lineal del sistema

Dado a que la relación que vincula la fuerza magnética con la corriente y la posición de la esfera produce un sistema no lineal, mediante la siguiente aproximación podemos aproximar dicha ecuación para poder producir una relación lineal del mismo.

$$F_{mag} \approx \bar{F}_{mag} + k_i \Delta i(t) + k_y \Delta y(t)$$

donde:

$$\bar{F}_{mag} = \frac{k \bar{i}^2}{\bar{y}}, \quad k_i = \left. \frac{\partial F_{mag}(t)}{\partial i} \right|_{i(t)=\bar{i}; y(t)=\bar{y}}, \quad k_y = \left. \frac{\partial F_{mag}(t)}{\partial y} \right|_{i(t)=\bar{i}; y(t)=\bar{y}},$$

$$\Delta i(t) = i(t) - \bar{i}, \quad \Delta y(t) = y(t) - \bar{y}$$

Por lo que mediante dichas ecuaciones calculamos los correspondientes valores de las constantes k_i y k_y .

$$k_i = \left. \frac{\partial}{\partial i} (F_{mag}(t)) \right|_{i(t)=\bar{i}; y(t)=\bar{y}} = \left. \frac{\partial}{\partial i} \left(\frac{k i(t)^2}{y(t)} \right) \right|_{i(t)=\bar{i}; y(t)=\bar{y}}$$

$$k_i = \left. \frac{2k i(t)}{y(t)} \right|_{i(t)=\bar{i}; y(t)=\bar{y}} = \frac{2k \bar{i}}{\bar{y}}$$

Remplazando los valores de equilibrio antes calculados obtenemos:

$$k_y = 7,84N/A$$

Luego,

$$k_y = \frac{\partial}{\partial y}(F_{mag}(t)) \Big|_{i(t)=\bar{i}; y(t)=\bar{y}} = \frac{\partial}{\partial y} \left(\frac{ki(t)^2}{y(t)} \right) \Big|_{i(t)=\bar{i}; y(t)=\bar{y}}$$

$$k_y = -k \left(\frac{i(t)}{y(t)} \right)^2 \Big|_{i(t)=\bar{i}; y(t)=\bar{y}} = -k \left(\frac{\bar{i}}{\bar{y}} \right)^2$$

Remplazando los valores de equilibrio antes calculados obtenemos:

$$k_i = -7,683N/m$$

2.2.1. Implementación del diagrama de bloques en Simulink

Ahora procedemos a realizar el diagrama de bloques en Simulink, donde remplazamos los bloques no lineales por las ganancias calculadas y pasivando las entradas correspondientes a los valores de equilibrio.

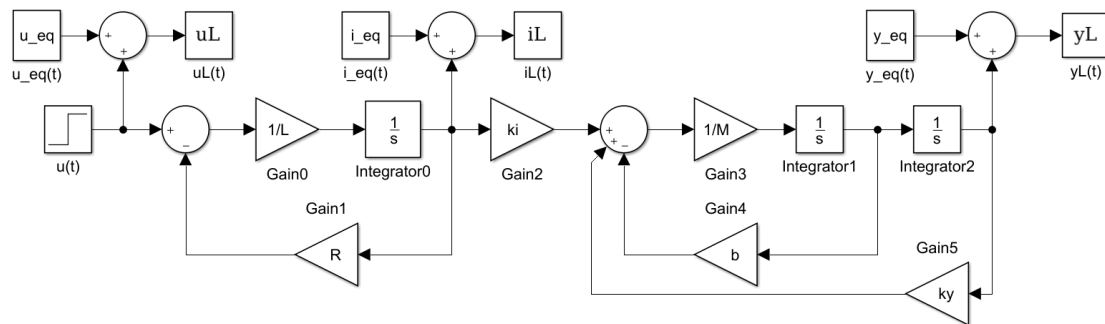


Figura 22: Diagrama de bloques linealizado en Simulink

2.2.2. Gráficas de las variables $u(t)$, $i(t)$ y $y(t)$ del modelo linealizado

Una vez realizada la simulación procedemos a graficar los resultados obtenidos mediante el siguiente script, donde mantenemos las constantes tenidas en cuenta en el primer modelo no lineal.

```
% Implementacion del diagrama de bloques linealizado
% Valores de equilibrio
u_eq = 10;
i_eq = 0.5;
y_eq = 0.2551;

% Condiciones iniciales Ki y Ky
ki = (2*k*i_eq)/(y_eq);
ky = -(k*(i_eq)^2)/((y_eq)^2);

% Graficas
figure(2)
subplot(3,1,1), plot(yL.time,yL.data,'r'), grid on, hold on;
title('Posicion de la masa M (y(t))');
xlabel('Tiempo [seg]')
ylabel('y(t)')
subplot(3,1,2), plot(iL.time,iL.data,'r'), grid on, hold on;
title('Corriente (i(t))');
xlabel('Tiempo [seg]')
ylabel('i(t)')
subplot(3,1,3), plot(uL.time,uL.data,'r'), grid on, hold on;
title('Tension (u(t))');
xlabel('Tiempo [seg]')
ylabel('u(t)')
```

Figura 23: Representación de respuesta del sistema

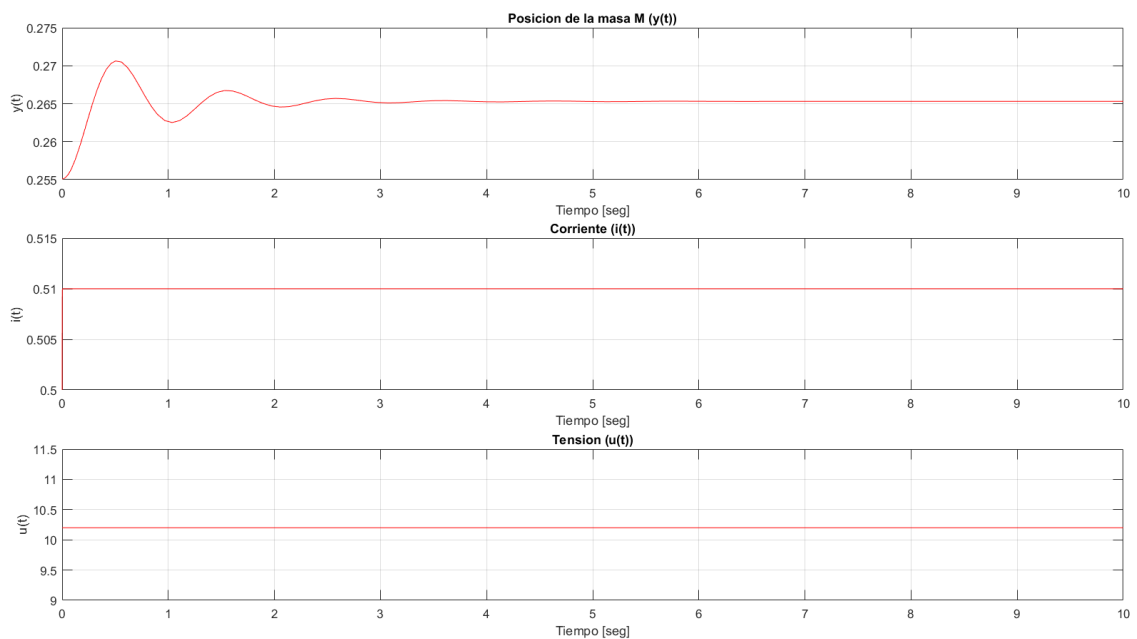


Figura 24: Gráficos para un aumento de tensión de 0,2V

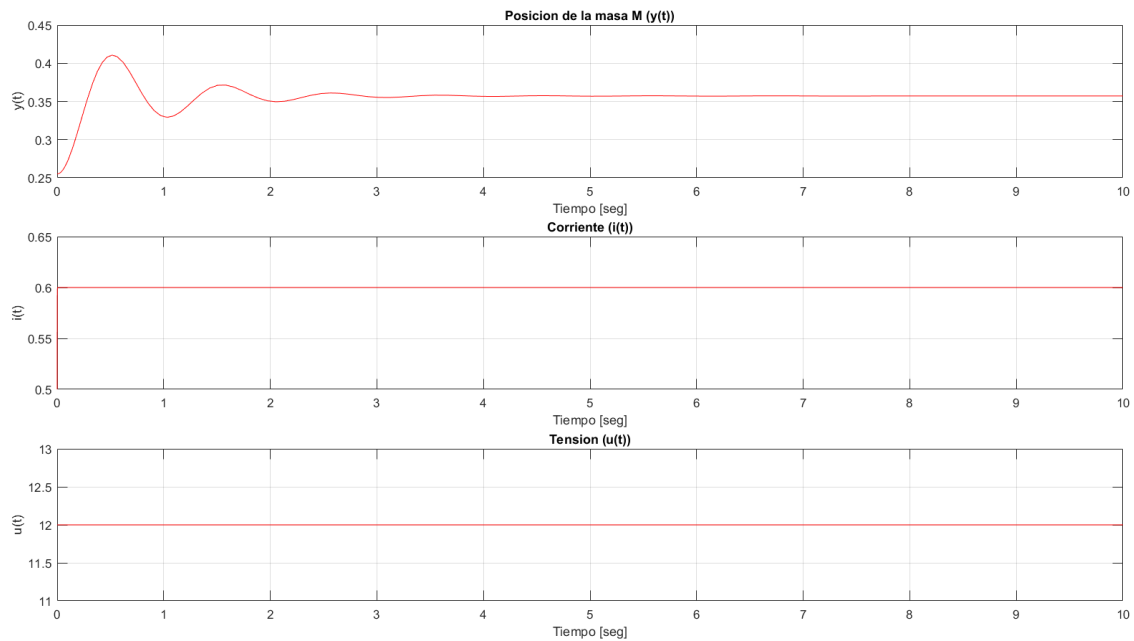


Figura 25: Gráficos para un aumento de tensión de 2V

Vemos entonces que para esta simulación del sistema linealizado, como es de esperarse, la respuesta resulta muy similar a la obtenida con el anterior modelo.

2.2.3. Comparación entre los resultados de los dos modelos

Para poder determinar de forma mas exacta cuales son las variaciones que se presentan en cada uno de los dos modelos, a continuación, realizamos en una misma gráfica las respuestas de las simulaciones.

```
% Comparacion de diagrama de bloques lineal y no lineal
% Graficas
figure(3)
subplot(3,1,1), plot(y.time,y.data,'b', yL.time,yL.data,'r'), grid on, hold on;
title('Comparacion de la posicion de la masa M (y(t))')
legend('No lineal','Lineal')
xlabel('Tiempo [seg]')
ylabel('y(t)')
hold on, grid on
subplot(3,1,2), plot(i.time,i.data,'b', iL.time,iL.data,'r'), grid on, hold on;
title('Comparacion de la corriente (i(t))')
legend('No lineal','Lineal')
xlabel('Tiempo [seg]')
ylabel('i(t)')
hold on, grid on
subplot(3,1,3), plot(u.time,u.data,'b', uL.time,uL.data,'r'), grid on, hold on;
title('Comparacion de la tension (u(t))')
legend('No lineal','Lineal')
xlabel('Tiempo [seg]')
ylabel('u(t)')
hold on, grid on
```

Figura 26: Representación de respuesta del sistema

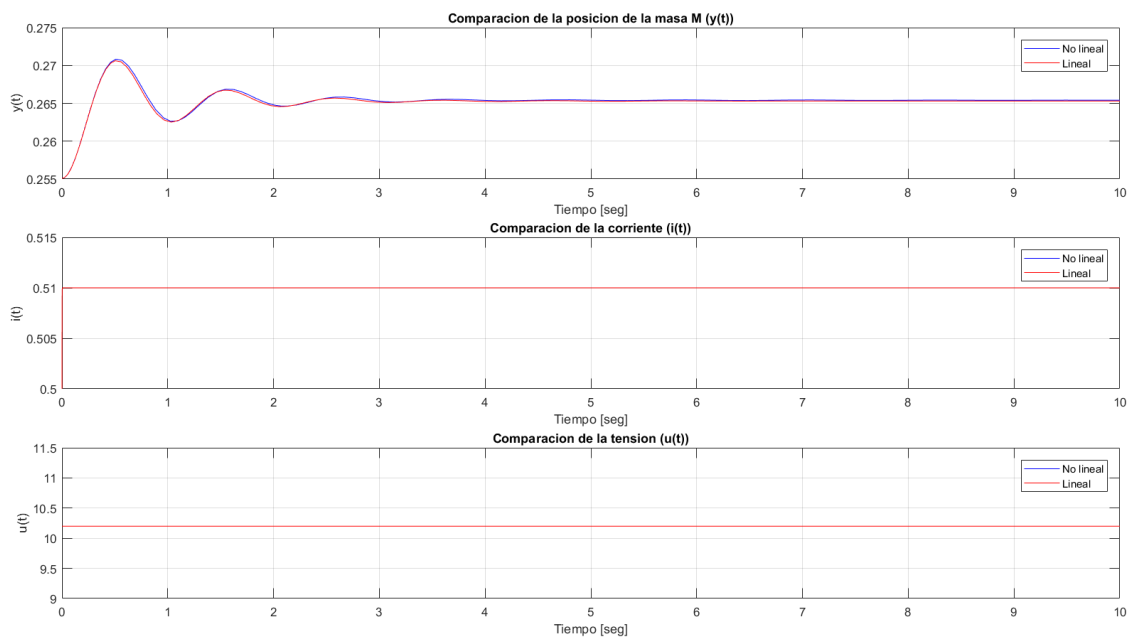


Figura 27: Comparación para un aumento de tensión de 0,2V

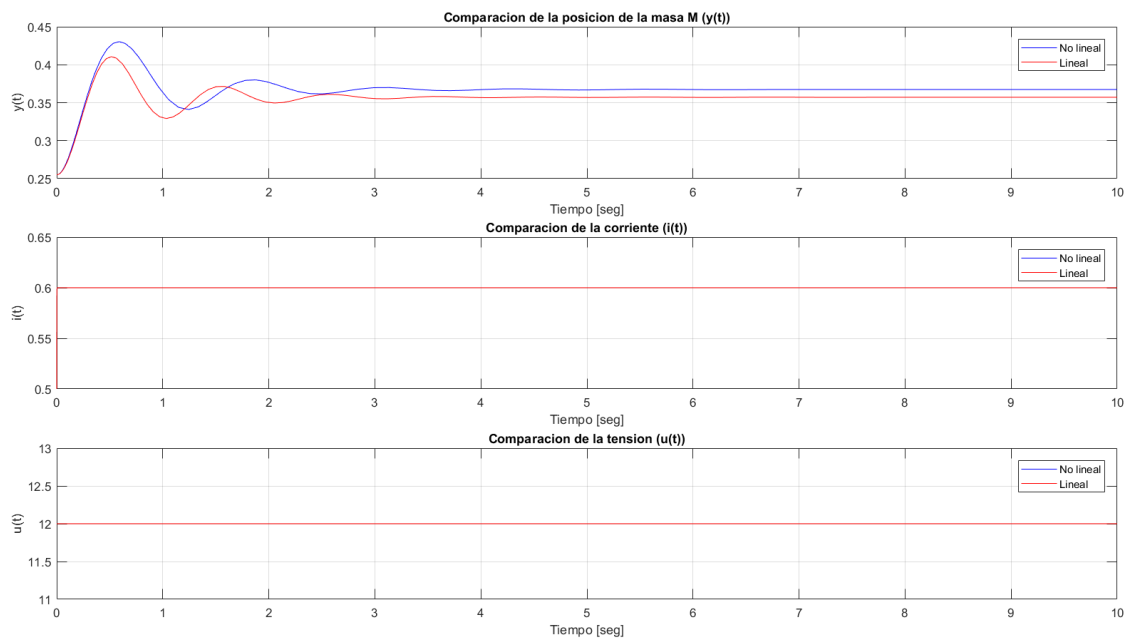


Figura 28: Comparación para un aumento de tensión de 2V

Para una pequeña variación de la entrada respecto al valor de equilibrio se puede observar una diferencia casi despreciable en cuanto a los resultados obtenidos, pero si en cambio aumentamos dicha variación vemos un cambio notable. Este fenómeno se debe a que el modelo incremental lineal resulta mas inexacto cuanto mas nos alejamos del valor

de equilibrio en el cual estamos calculando la respuesta del mismo, es decir, cuanto mas variamos el valor de la entrada.

2.2.4. Función transferencia entre la entrada y la salida incremental

Procedemos ahora a hallar la función transferencia $G(s)$ entre la entrada incremental Δu y la salida incremental Δy para los valores calculados de k_i y k_y .

Calcularemos la función transferencia a partir del diagrama de bloques correspondiente a nuestro sistema, como se muestra a continuación.

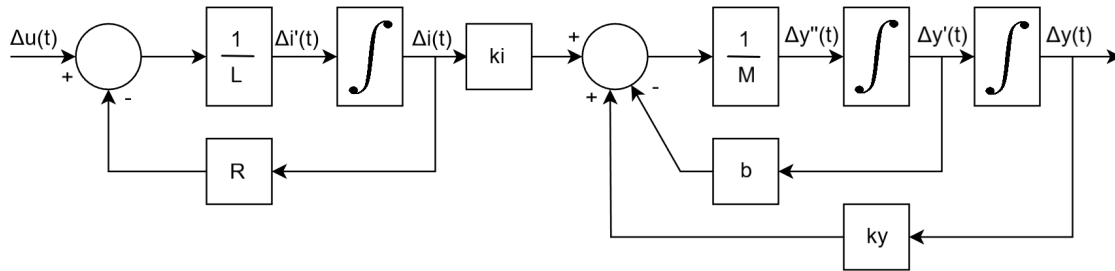


Figura 29: Diagrama de bloques sistema linealizado

Podemos ver que existen dos lazos simples de retroalimentación, por lo que comenzamos calculando sus equivalentes.

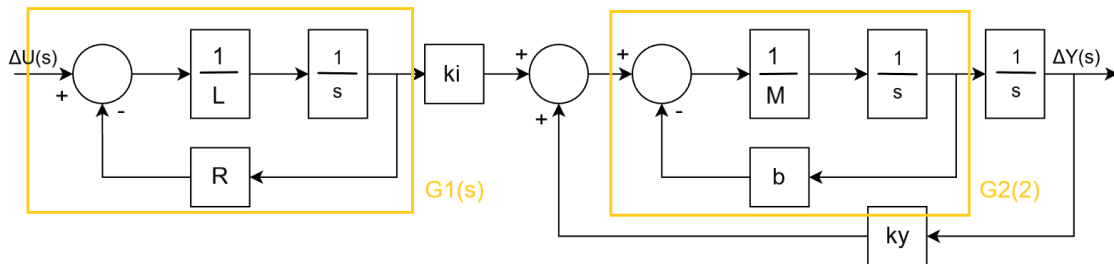


Figura 30: Diagrama de bloques sistema linealizado

Luego,

$$G_1(s) = \frac{\frac{1}{sL}}{1 + \frac{R}{sL}} = \frac{1}{sL + R}$$

$$G_2(s) = \frac{\frac{1}{sM}}{1 + \frac{b}{sM}} = \frac{1}{sM + b}$$

Por lo tanto obtenemos el siguiente diagrama simplificado, donde podemos ver que existe otro lazo simple de retroalimentación, por lo que continuamos calculando su equivalente.

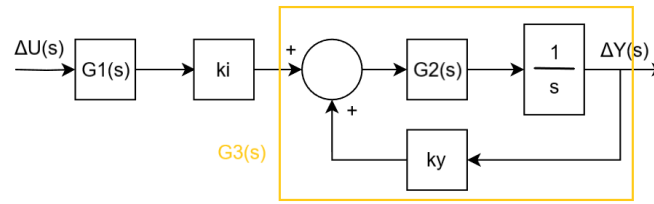


Figura 31: Diagrama de bloques sistema linealizado simplificado

$$G_3(s) = \frac{\frac{G_2(s)}{s}}{1 + \frac{k_y G_2(s)}{s}} = \frac{1}{Ms^2 + bs - k_y}$$

Finalmente obtenemos el siguiente diagrama de bloques, del cual podemos calcular simplemente la función transferencia del sistema.



Figura 32: Diagrama de bloques sistema linealizado simplificado

$$G(s) = G_1(s)k_i G_3(s) = \frac{k_i}{(sL + R)(Ms^2 + bs - k_y)}$$

Donde esquemáticamente el diagrama de bloques del sistema quedaría representado como se muestra en la siguiente figura.



Figura 33: Diagrama de bloques sistema linealizado simplificado

Por lo tanto la función transferencia del sistema queda representada por:

$$G(s) = \frac{k_i}{(ML)s^3 + (MR + bL)s^2 + (bR - k_y L)s - k_y R}$$

Mediante el siguiente script representamos la función transferencia en Matlab.

```
%Calculo de funcion transferencia
num = ki;
den = [M*L, b*L+R*M, R*b-ky*L, -ky*R];
% Funcion transferencia
G = tf(num, den);
```

Figura 34: Función transferencia

De lo cual obtenemos el siguiente resultado.

```
G =
      7.84
-----
0.002 s^3 + 4.005 s^2 + 10.08 s + 153.7
Continuous-time transfer function.
```

Figura 35: Función transferencia

Ahora calculamos los polos de la función transferencia.

```
% Polos y ceros
polos = roots(den);
ceros = roots(num);
```

Figura 36: Polos y ceros

De lo cual obtenemos.

```
polos =

      1.0e+03 *
-2.0000 + 0.0000i
-0.0013 + 0.0061i
-0.0013 - 0.0061i
```

Figura 37: Polos del sistema

Dado que todos los polos presentan parte real negativa, podemos decir que el sistema es BIBO estable.

2.2.5. Diagrama de Bode

Realizamos además el diagrama de Bode correspondiente a nuestro sistema implementando el siguiente script para dicho fin.

```
% Bode
figure(4)
bode(num,den), grid on
```

Figura 38: Diagrama de Bode

De lo cual obtenemos:

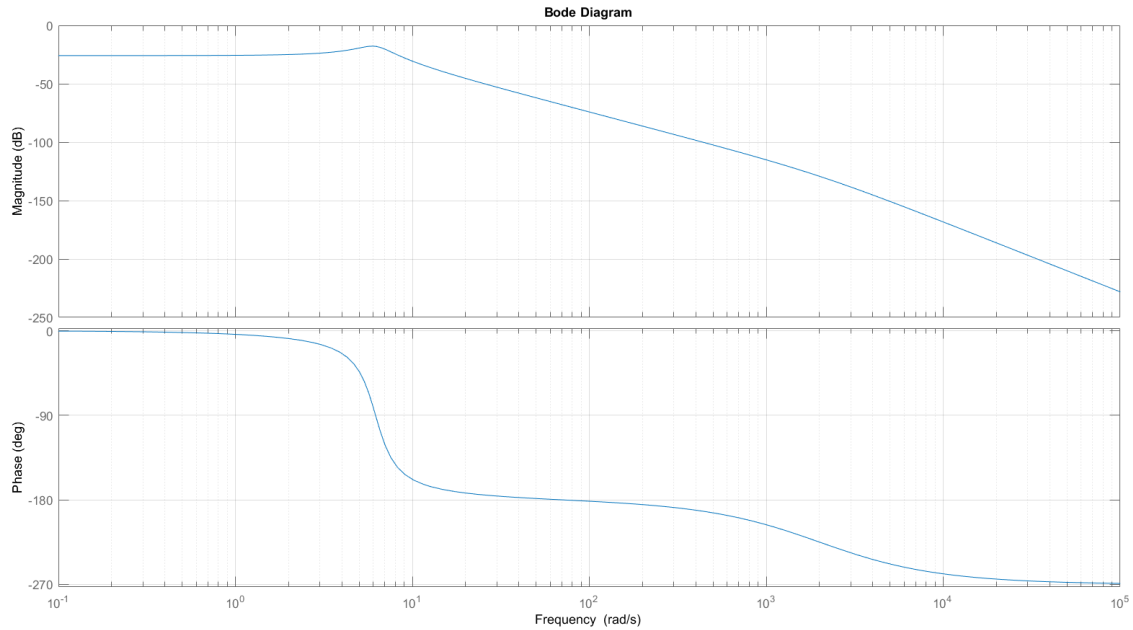


Figura 39: Diagrama de Bode

En el diagrama de Bode vemos que el mismo presenta un pico y luego una tendencia decreciente, esto debido a la presencia de un par de polos complejos conjugados y un polo real. El par de polos complejos produce un pico en $\omega_n = 6,2 \text{ rad/s}$, y luego el polo real produce a partir de $\omega_n = 2000 \text{ rad/s}$ una caída de 20 dB por década, lo cual coincide con lo visto gráficamente.

Respecto a la fase, notamos que la misma va a tender a -180 en para el par de polos complejos pasando por -90 en $\omega_n = 6,2 \text{ rad/s}$. Luego el polo real produce una caída de -90 pasando por -45 en $\omega_n = 2000 \text{ rad/s}$. Dado los aportes de los tres polos vemos que el desfase final es -270 .

2.3. Implementación de control proporcional P

Nos proponemos ahora implementar un control por retroalimentación de salida tipo P para el modelo incremental lineal, como se muestra en la siguiente figura.

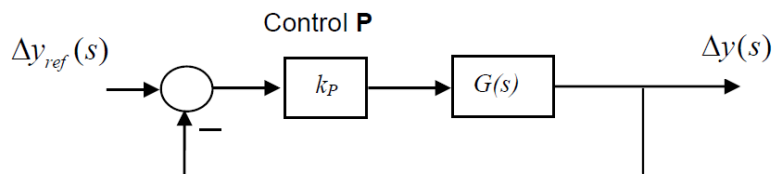


Figura 40: Control proporcional

Para dicho fin, debemos encontrar el valor de la constante k_P que nos asegure que el sistema de lazo cerrado sea BIBO estable.

La función transferencia del sistema incluyendo el lazo simple de retroalimentación

queda determinada de la siguiente manera.

$$G_P(s) = \frac{k_P G(s)}{1 + k_P G(s)}$$

$$G_P(s) = \frac{k_i k_P}{(ML)s^3 + (MR + bL)s^2 + (bR - k_y L)s + k_P k_i - k_y R}$$

Dado a que el denominador es un polinomio de tercer grado, no podemos garantizar la estabilidad del sistema a simple vista, para ello debemos recurrir al criterio de Routh, el cual nos da condiciones necesarias y suficientes para la BIBO estabilidad del sistema, para un rango de valores de la constante k_P .

$$\begin{array}{c|cc} s^3 & ML & bR - k_y L \\ s^2 & MR + bL & k_P k_i - k_y R \\ s^1 & \frac{ML(k_P k_i - k_y R) - (MR + bL)(bR - k_y L)}{MR + bL} & \\ s^0 & k_P k_i - k_y R & \end{array}$$

Por lo tanto para satisfacer el criterio de estabilidad se debe cumplir simultáneamente que:

$$\begin{cases} \frac{ML(k_P k_i - k_y R) - (MR + bL)(bR - k_y L)}{MR + bL} > 0 \\ k_P k_i - k_y R > 0 \end{cases}$$

Remplazando por los valores correspondientes obtenemos que:

$$\begin{cases} k_P < 2554,27 \\ k_P > -19,06 \end{cases}$$

Por lo tanto el valor de k_P debe mantenerse en ese rango de valores para poder garantizar la BIBO estabilidad del sistema.

2.3.1. Implementación del sistema de lazo cerrado en Simulink

Ahora procedemos a realizar el diagrama de bloques en Simulink, agregando el lazo cerrado de retroalimentación, para un valor de $k_P = 1000$.

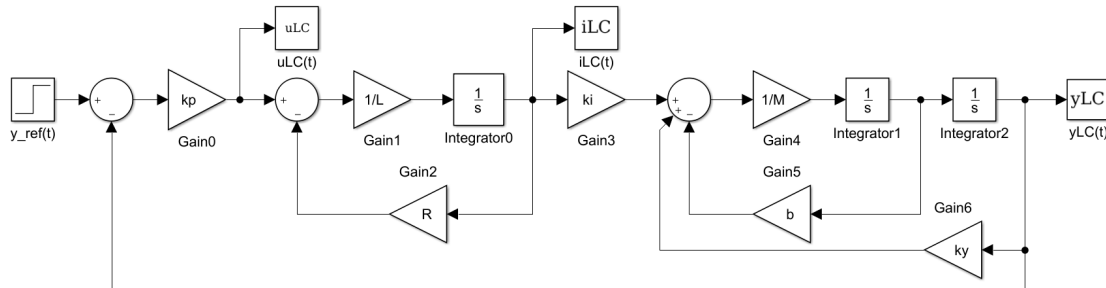


Figura 41: Diagrama de bloques del sistema con control P

Para obtener la respuesta de nuestro sistema realizamos el siguiente script.

```

% Constante de proporcionalidad
kp = 1000;
% Graficos
figure (5)
subplot(2,1,1), plot(yLC.time,yLC.data,'b'), grid on, hold on
title('Posicion de la masa M (y(t)) con control P');
xlabel('Tiempo [seg]')
ylabel('y(t)')
subplot(2,1,2), plot(iLC.time,iLC.data,'b'), grid on, hold on
title('Corriente (i(t)) con control P')
xlabel('Tiempo [seg]')
ylabel('i(t)')

```

Figura 42: Representación de respuesta del sistema

Donde los resultados obtenidos se pueden ver en la siguiente figura.

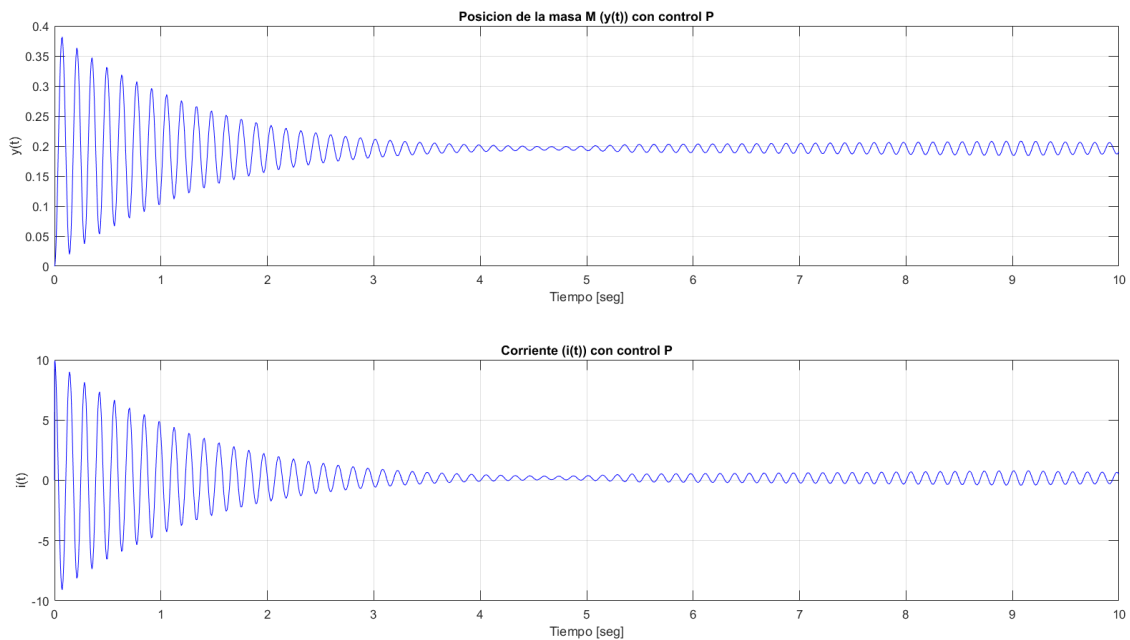


Figura 43: Respuesta del sistema

Podemos observar entonces que mediante el sistema de control de tipo P, se logra que la salida del sistema tienda al valor de referencia de la entrada. Esto lo hace con una respuesta subamortiguada, tendiendo hacia un valor final de manera oscilatoria respecto al valor de referencia que tiene nuestro controlador.

Conclusión

A modo de conclusión podemos destacar que ambos problemas requirieron tanto un desarrollo práctico como así el análisis de sus fundamentos teóricos a la hora de encontrar una solución a los mismos, como así también una correcta implementación numérica a la hora de trabajar con el software Matlab como así también la herramienta Simulink.

Para el primer caso, resultaron de gran utilidad las funciones de Matlab para poder calcular polos y ceros, como así también poder realizar el diagrama de Bode de manera muy sencilla. Además pudimos comparar los resultados obtenidos con datos experimentales extraídos de una simulación real del sistema y ver que nuestro modelo presenta un comportamiento muy similar al dispositivo real en funcionamiento.

Para el segundo caso analizamos un problema que pudimos modelar mediante un diagrama de bloques, de dos maneras diferentes. Obtuvimos una representación no lineal del mismo y una linealización, la cual luego implementamos en Simulink, lo que nos permitió ensayar ambos modelos y obtener importantes conclusiones acerca de su comportamiento. Por ejemplo la afectación de la respuesta del sistema linealizado ante un aumento de los parámetros de entrada. Además implementamos para uno de nuestros modelos un sistema de control de retroalimentación por lazo cerrado tipo P, el cual pudimos determinar el rango de valores de la constante de proporcionalidad del mismo, donde el mismo presenta características de BIBO estabilidad.