Escuela Politécnica Superior de Jaén

Grado en Ingeniería Electrónica Industrial

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior

Trabajo Fin de Grado

DESARROLLO E IMPLANTACIÓN DE PLANTAS VIRTUALES EN PLATAFORMAS DE VIRTUALIZACIÓN DE BAJO COSTE

Alumno: José Álvaro Molina Lorite

Tutores: Prof. D. Pablo Cano Marchal
Prof. D. Diego Martínez Gila

Dpto: Ingeniería Electrónica y Automática

Noviembre, 2017

Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Ingeniería Electrónica y Automática

Don Pablo Cano Marchal y Don Diego Martínez Gila, Los tutores del Proyecto Fin de Carrera titulado: Desarrollo e implementación de plantas virtuales en plataforma de virtualización de bajo coste, que presenta José Álvaro Molina Lorite, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Noviembre de 2017

El alumno:

José Álvaro Molina Lorite

Los tutores:

Pablo Cano Marchal

Diego Martínez Gila

Índice

Índice de Figuras

Índice de tablas

1. INTRODUCCIÓN

El presente trabajo final de grado trata sobre el desarrollo e implementación de las plantas virtuales: Levitador Magnético y el Péndulo Invertido para plataformas de virtualización de bajo coste. En particular, la tecnología utilizada es Easy Java Simulations (abreviado: EjsS) que dispone de lenguaje de programación JavaScript por lo que es necesario un dispositivo hardware que disponga de navegador web para la implementación final del sistema.

1.1. Motivación

La principal motivación que lleva a la realización de este trabajo final de grado es el interés por aprender en este campo de la ingeniería como es la ingeniería de control y las ventajas que aporta trabajar con plataformas virtuales de bajo costo.

Por un lado, resulta interesante este trabajo porque permite al alumno emplear conocimientos obtenidos durante sus estudios universitarios, como el desarrollo del modelo matemático y dada su complejidad, ser capaz de observar la necesidad del controlador adecuado para cada caso utilizando tanto métodos matemáticos como software de diseño.

Asimismo, resulta interesante la visualización en plataformas virtuales ya que estas son muy útiles para fines educativos permitiendo así, una mayor comprensión de los modelos desarrollados.

1.2. Objetivos

El objetivo principal de este proyecto es la implementación de plantas virtuales en plataformas de virtualización. Para cumplir el objetivo principal se desarrolla el modelo matemático de los sistemas dinámicos, se obtiene sus funciones de transferencia y comprueba su estabilidad, para el caso en que los sistemas sean inestables se procede a la búsqueda de un controlador adecuado.

Para los sistemas inestables es necesario la búsqueda de un controlador adecuado para su estabilización, en este punto es necesario el uso de un software específico para su diseño. El software a utilizar es Matlab que sirve como apoyo para la validación del controlador.

Asimismo, para facilitar la implementación de las ecuaciones de los sistemas en EjsS se recurre al modelo en espacio de estados de los sistemas.

1.3. Metodología

El primer punto llevado a cabo ha sido el desarrollo del modelado no lineal de los sistemas dinámicos, realizando un estudio de las fuerzas implicadas en el sistema.

Los modelos no lineales son inestables y no permiten una manipulación sencilla de las ecuaciones, por lo tanto, se ha realizado la linealización en torno a un punto de operación dando como resultado los modelos lineales.

Los modelos lineales del levitador magnético y el péndulo invertido son inestables por lo que ha sido necesario el diseño de un controlador que proporcione a la salida de los sistemas una respuesta estable. Para el estudio de un controlador adecuado para los sistemas resulta complejo solo con el uso de EjsS por lo que se ha servido del apoyo de un programa más potente para este aspecto, Matlab.

EjsS no permite incluir funciones de transferencia, pero dispone de páginas ODEs que permite insertar ecuaciones diferenciales. Las ecuaciones diferenciales de los modelos se han obtenido desarrollando el modelo en espacio de estados.

Por último, se han desarrollado una interface en EjsS para cada modelo y se han implementado sus ecuaciones de estados y controladores.

1.4. Organización de la memoria

Este proyecto está dividido en 7 capítulos, de la siguiente manera:

- Capítulo 1. Introducción. En este capítulo se realiza una introducción a la temática de este proyecto, donde se habla de la motivación que ha llevado a su realización, el objetivo final por que se ha llevado a cabo, así como la metodología aplicada durante el desarrollo del proyecto.
- Capítulo 2. Easy Java Simulations. Aquí se comenta las partes en las que está estructurado el programa, el funcionamiento que desempeña las opciones principales, así como las condiciones de uso que dispone el programa.
- Capítulo 3. Fundamentos del modelado de sistemas. En este capítulo se dan nociones teóricas sobre los distintos métodos que se llevan a cabo para la obtención de los modelos.
- Capítulo 4. Péndulo Invertido. En este capítulo se desarrolla el modelado matemático del péndulo invertido, diseño del controlador e implementación en EjsS.
- Capítulo 5. Levitador magnético. En este capítulo se desarrolla el modelo matemático del levitador magnético, diseño del controlador e implementación en EjsS.
- Capítulo 6. Anexos. Dentro este capítulo se encuentra material de apoyo al capítulo 5 acerca transformar una función de transferencia en modelo de espacio de estados y al capítulo 4 donde se visualiza el bloque del péndulo invertido no lineal en Simulink.
- Capítulo 7. Bibliografía. Contiene las referencias y bibliografías utilizadas durante el desarrollo del proyecto.

2. EASY JAVA SIMULATIONS

Easy java / Javascript simulations (abreviado: EjsS) es una herramienta de simulación diseñada y desarrollado para el aprendizaje o enseñanza de aquellas personas que no tenga conocimientos avanzados en programación para que así sean capaces de crear sus propias simulaciones interactivas. EjsS ha sido creado por el profesor Francisco Esquembre, Doctor en Matemáticas de la Universidad de Murcia, España.

El entorno de simulación EjsS, así como su documentación y algunos ejemplos diseñados en este entorno de estudio, puede ser descargado gratuitamente de siguiente sitio web: *http://fem.um.es/Ejs*.

2.1. Licencia y condiciones de uso de EjsS

Las condiciones de uso de EjsS y el copyright se encuentran disponibles en el sitio web: *http://fem.um.es/Ejs*. Sus principales características son:

El uso de las simulaciones creadas en EjsS pueden ser usadas solo bajo el acuerdo de licencia correspondiente.

Las simulaciones de Easy Java (EjsS) pueden distribuirse bajo los términos de uso no comercial.

Si la creación de simulaciones en Easy Java Simulations es destinado a fines comerciales, es necesario ponerse en contacto con el siguiente correo electrónico: *ejs@um.es*, para llegar a un acuerdo de uso comercial.

El uso no comercial de EjsS está destinado enfocado al ámbito educativo sin intención de obtener ningún beneficio comercial, orientado principalmente a estudiantes y profesores para favorecer su enseñanza o sus estudios académicos.

2.2. Estructura de Easy Java Simulations

El contenido de los siguientes apartados está adaptado a partir del *Ejswiki* contenido disponible en *http://www.um.es/fem/EjsWiki/.*

La estructura de la que dispone la aplicación Easy Java Simulations es la siguiente:

2.2.1. Consola EjsS:

Tras iniciar, el EjsS lo primero en mostrar es una consola encargada de ejecutar la interface EjsS donde se puede proceder al desarrollo e implementación del modelo, por otra parte, en la consola muestra cualquier tipo de mensaje de salida o mensaje de error que puede producirse durante la simulación.

La consola aparece en una ventana separada de la interface EjsS. Cuando la consola arranca, ejecuta una copia de la interface EjsS aunque permite ejecutar varias simultáneamente.

Figura 2.1: Consola EJS

2.2.2. Interfaz de usuario de EjsS:

Una vez abierta la interface de usuario de EjsS una ventana (Figura 2.2), donde a la derecha de la imagen dispone de un panel de información para la simulación permitiendo al usuario guardar el archivo, buscar código, configurar EjsS, empaquetar uno o más simulaciones en un archivo JAR, etc.

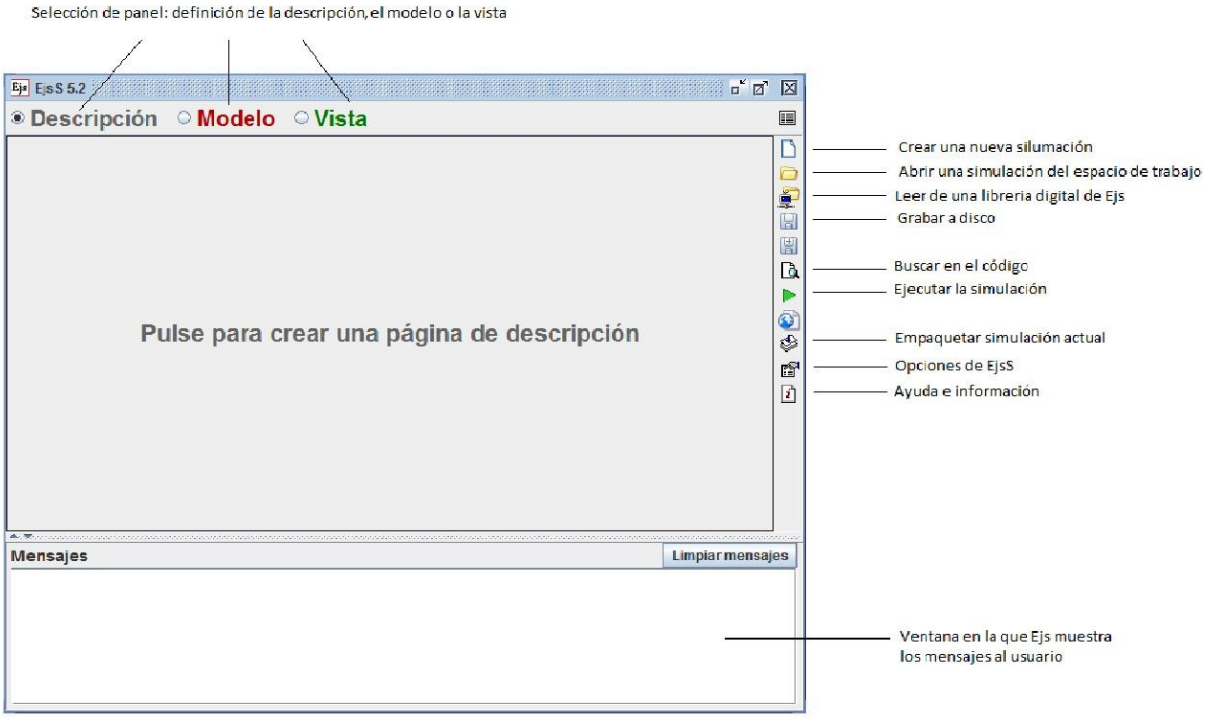


Figura 2.2: Interface de usuario EjsS

Por otro lado, en la parte superior de la interface de usuario existen tres paneles de trabajo:

i. **Descripción** este panel permite al usuario escribir textos introductorios donde narra la información cuando la simulación este en ejecución, este panel permite editar formato de fuente, insertar imágenes, tablas, etc. Por otra parte, si se prefiere un editor HTML el programa tiene la opción importar página HTML la cual incorpora un archivo HTML existente. También, dispone de la opción Enlace a página HTML externo dando opción a mostrar una página HTML existente.

ii. El segundo panel de trabajo, **Modelo**, dispone de seis pestañas:

- **Variables:** Esta pestaña permite declarar e inicializar variables del modelo, también da opción de elegir el tipo de variables: int, double, boolean, etc, para el caso de vectores o matrices permite determinar su dimensión.

Por otra parte, la pestaña variable permite hacer comentarios de cada variable y comentarios de página que favorecen la comprensión de la simulación.

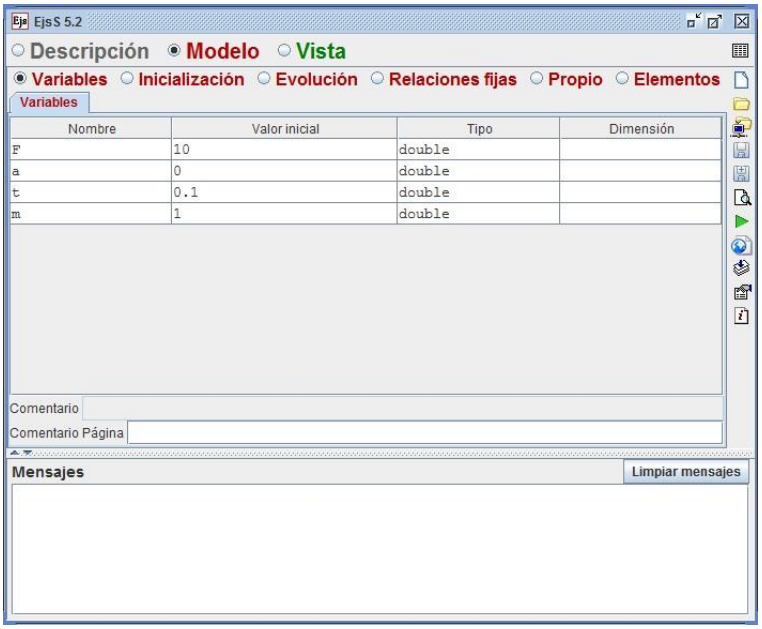


Figura 2.3: Pestaña de Variables

- **Inicialización:** Esta pestaña sirve como apoyo al panel de variable, es decir, se necesite algún cálculo previo al comienzo de la simulación.

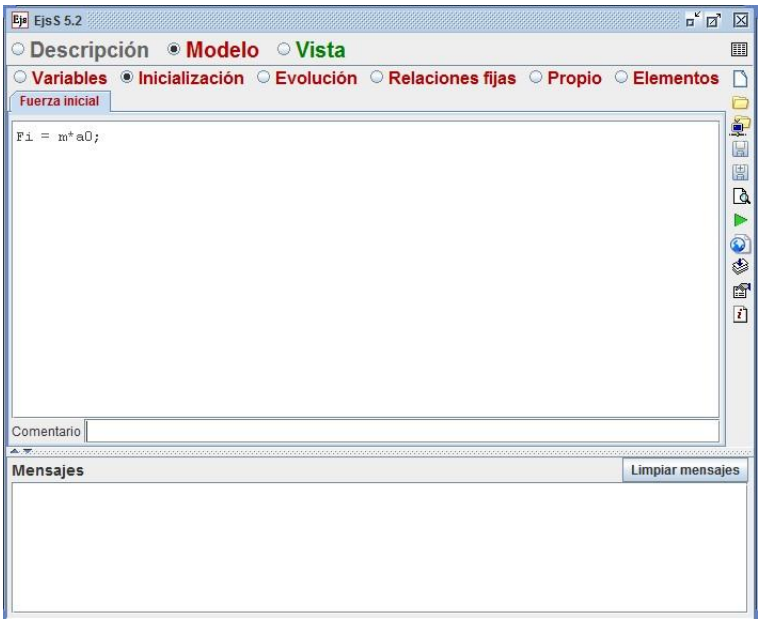


Figura 2.4: Pestaña Inicialización

- **Evolución:** Esta pestaña describe cómo se desarrolla el modelo respecto al tiempo. Junto a la pestaña de variables, son las pestañas mínimas para producir una simulación.
  El panel de evolución además permite al usuario crear una página de código o una página de EDO (ecuaciones diferenciales ordinarias).

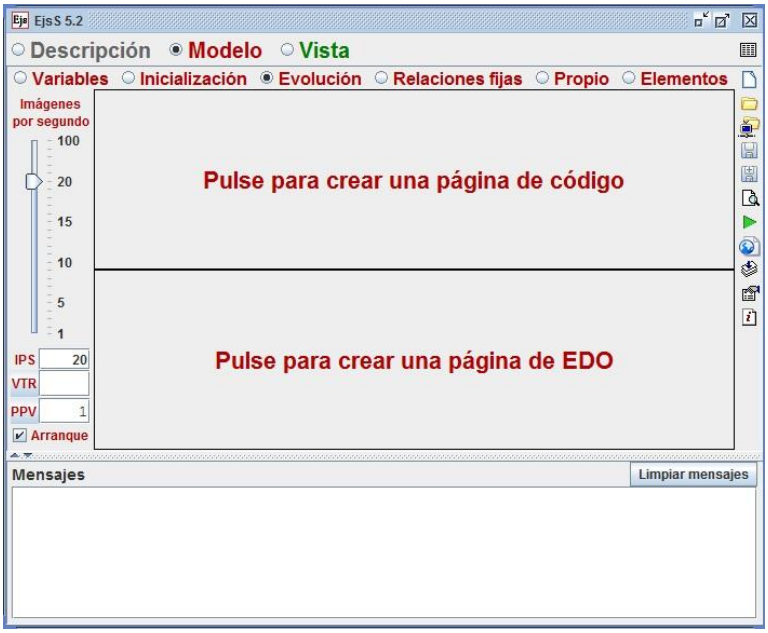


Figura 2.5: Pestaña de Evolución

- **Relaciones fijas:** Esta pestaña complementa al panel de evolución que se utiliza cuando es necesario realizar un cálculo adicional antes de cada paso de simulación.

- **Propio:** En esta pestaña se puede crear métodos adicionales, además de poder utilizar algunos métodos predefinidos de Easy Java Simulations.

- **Elementos:** Esta pestaña ofrece algunas herramientas predefinidas por EjsS.

iii. En tercer lugar, **Vista**, este panel de trabajo es muy importante porque permite al usuario desarrollar la interfaz gráfica de la simulación, dándole la posibilidad para controlar la simulación y mostrar su salida. La interfaz se construye mediante los elementos que se encuentra incluidos en la paleta y se agregan a la vista mediante árbol de elementos (Figura 2.6) y se visualiza la figura en el Frame.

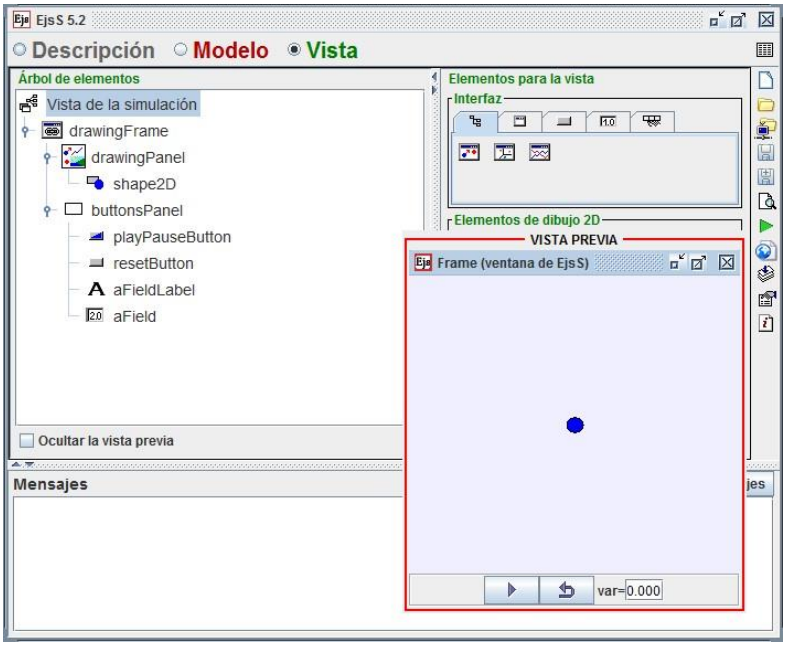


Figura 2.6: Vista

3. FUNDAMENTOS DE MODELADO DE SISTEMAS

Los sistemas dinámicos dan lugar a modelos matemáticos cuyas funciones representan de forma aproximada el comportamiento de las variables de los sistemas bajo estudio. Dichos modelos se caracterizan por producir conjuntos de ecuaciones diferenciales y algebraicas, normalmente no lineales que se obtienen a partir de un estudio analítico.

Los sistemas dinámicos tienen varias formas de representación y estas se pueden dividir en dos tipos principalmente: representación externa y representación interna.

Representación externa describe la conversión de una ecuación en el dominio del tiempo a una ecuación en el dominio de la transformada de Laplace, basado en el concepto de Función de Transferencia, que simplifica la resolución de las ecuaciones.

Representación interna o representación en espacio de estados describe un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales de primer orden.

Las ecuaciones diferenciales que detallan el funcionamiento dinámico de un sistema físico se desarrollan utilizando leyes físicas del proceso. Este método se aplica de igual manera a sistemas mecánicos, eléctricos, electromecánicos, hidráulicos, etc.

En la siguiente tabla viene dado algunas de estas ecuaciones diferenciales transformadas a Laplace:

Tipo de elemento	Elemento físico	Ecuación descriptiva	Transformada de Laplace
Sistema eléctrico	Inductancia eléctrica	$v_L(t) = L\frac{di(t)}{dt}$	$V_L(s) = LsI(s)$
	Tensión del condensador	$i_C(t) = C\frac{dv(t)}{dt}$	$I_C(s) = CsV(s)$
Sistema mecánico	Fuerza aplicada en la masa	$F(t) = m\frac{d^2x(t)}{dt^2}$	$F(s) = ms^2X(s)$
	Amortiguador traslacional	$F(t) = b\frac{dx(t)}{dt}$	$F(s) = bsX(s)$

Tabla 3.1: Resumen de ecuaciones diferenciales y su transformada

3.1. Linealización de sistemas

La linealización de un sistema se realiza para estudiar el comportamiento próximo a un punto de operación suponiendo las variaciones muy pequeñas cercanas a dicho punto, lo que da lugar a un modelo lineal:

$$y = y_0 + \left.\frac{df(x)}{dx}\right|_{x=x_0} * (x - x_0)$$

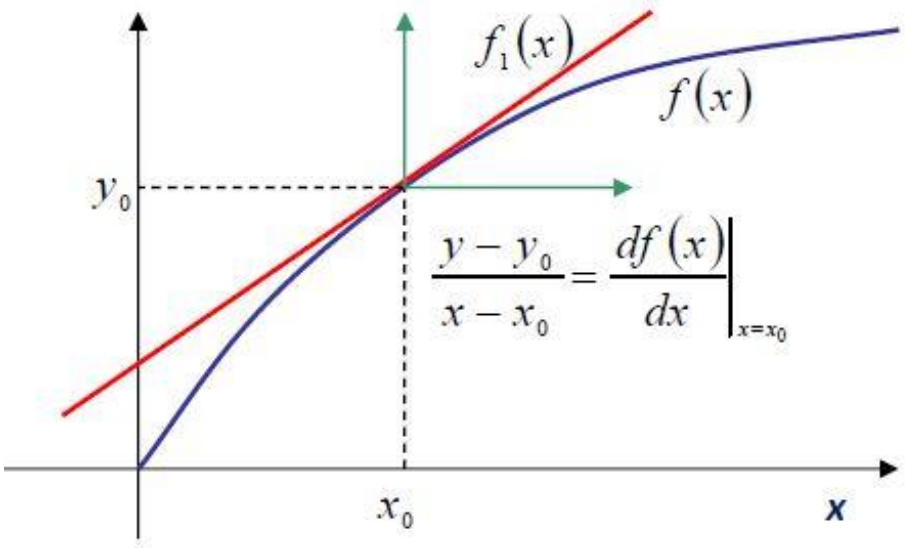


Figura 3.1: Gráfica de linealización

La linealización de los sistemas aportan algunas ventajas, dado que el uso de las ecuaciones no lineales resulta complicado, al realizar una aproximación lineal se obtienen unas ecuaciones que proporcionan una manipulación más sencilla.

Algunos de los inconvenientes que se pueden encontrar cuando se linealiza es que el rango de validez es más limitado. Además de debe de tener en cuenta la variable de desviación ya que esta representa el punto de operación en torno al cual se ha linealizado el sistema.

3.2. Función de transferencia

La función de transferencia de un sistema se define como la relación entre variable de salida y la de entrada en el dominio de Laplace, suponiendo que las condiciones iniciales son iguales a cero. Esta función puede expresarse como:

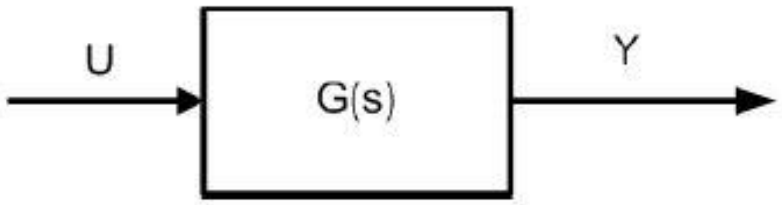


Figura 3.2: Función de transferencia

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + \cdots + b_1 \cdot s + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \cdots + a_1 \cdot s + a_0} \quad (3.1)$$

Donde, en el numerador se encuentra el número *m* de ceros y en el denominador el número *n* de polos. Para que el sistema sea realizable físicamente el grado del numerador debe ser menor o igual que el denominador.

La expresión de salida para una entrada sería:

$$Y(s) = U(s) \cdot G(s) \tag{3.2}$$

La solución correspondiente en el dominio del tiempo se obtiene al aplicar a Y(s) la transformada inversa de la Laplace, dando lugar a y(t).

3.3. Modelo en espacio de estados

El modelo en espacio de estados es también conocido como representación interna, esta representación es válido tanto para sistemas lineales como para sistemas no lineales y es el método más apropiado para los sistemas multivariables (SIMO, MISO y MIMO).

Los sistemas multivariables SIMO son sistemas que presenta una entrada y dos o más salidas, el sistema MISO son sistemas que presentan dos o más entradas y una sola salida y el sistema MIMO presentan dos o más entradas y salidas.

Los modelos en espacio de estados se representan en forma matricial de primer orden donde las variables de entrada, salida y estado son expresadas como vectores y las ecuaciones diferenciales se escriben de forma matricial cuando el sistema dinámico es lineal e invariante en el tiempo. Si se tiene un sistema de orden *n* se caracteriza por tener *n* variables, las cuales son denominadas variables de estado del sistema y son dependientes del tiempo, estas variables de estado del sistema son las que determinan las condiciones de la dinámica del sistema.

Donde:

u(t): Es la señal de entrada.

y(t): Es la señal de salida.

x(t): Vector de estados.

Las representaciones de las variables de estado pueden ser las siguientes:

$$u(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_e(t) \end{pmatrix};\ x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{pmatrix};\ y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_s(t) \end{pmatrix} \tag{3.3}$$

El sistema tiene *e* entradas, *s* salidas y *n* variables de estado.

Analizando la representación de la (Figura 3.3) debe predecirse la respuesta y(t) con una excitación provocada por la entrada u(t) conociendo los valores iniciales de x(0):

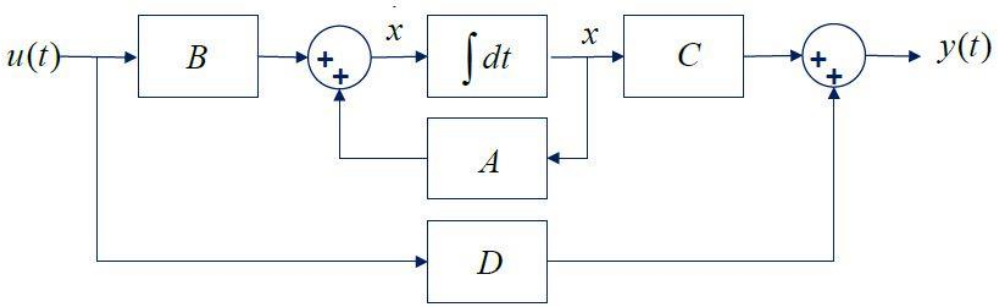


Figura 3.3: Diagrama de representación de modelo en espacio de estado

A partir del diagrama de la (Figura 3.3) se puede determinar el siguiente conjunto de ecuaciones diferenciales:

$$\begin{aligned} \dot{x}(t) &= A(t) \cdot x(t) + B(t) \cdot u(t) \\ y(t) &= C(t) \cdot x(t) + D(t) \cdot u(t) \end{aligned} \tag{3.4}$$

Donde:

ẋ(t): Ecuación diferencial de estado.

x(t): Vector de estado.

u(t): Señal de entrada.

y(t): Ecuación de salida.

Otra forma de representación del modelo en espacio de estados es en su forma canónica. Teniendo una función de transferencia de orden *n* de la siguiente forma:

$$G(s) = \frac{b_m \cdot s^m + b_{m-1} \cdot s^{m-1} + \cdots + b_1 \cdot s + b_0}{s^n + a_{n-1} \cdot s^{n-1} + a_{n-2} \cdot s^{n-2} + \cdots + a_1 \cdot s + a_0} \tag{3.5}$$

Cualquier función de transferencia con la forma que se muestra en la ecuación (3.5) puede ser escrita como un modelo en espacio de estado en su forma canónica. Los coeficientes del numerador y el denominador pueden ser insertado en la matriz del modelo en espacio de estado. La matriz de representación canónica tiene la siguiente forma:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 1 \\ -a_0 & -a_2 & -a_3 & \dots & -a_{n-1} \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \cdot u(t) \tag{3.6}$$

$$\begin{gathered} \dot{x}_1(t) = x_2(t)\ ;\ \dot{x}_2(t) = x_3(t); \dots;\ \dot{x}_n(t) = x_{n+1}(t) \\ \dot{x}_{n+1}(t) = -a_0 \cdot x_1(t) - a_1 \cdot x_2(t) - \cdots - a_{n-1} \cdot x_n(t) \cdot u(t) \end{gathered} \tag{3.7}$$

La variable de salida es:

$$y(t) = b_0 \cdot x_1(t) + b_1 \cdot x_2(t) + \cdots + a_m \cdot x_{m+1}(t) + u(t) \tag{3.8}$$

En el caso de los controladores, también se puede aplicar el método de espacio de estado en su forma canónica. Para el caso concreto en el que el controlador presenta un polo en el origen que se puede ver en la ecuación (3.9) se puede resolver de la siguiente manera:

$$\frac{Y(s)}{U(s)} = \frac{Z(s)}{Z(s)} \frac{(s + q_1) \cdot (s + q_2)}{s \cdot (s + p_1)} \tag{3.9}$$

Dónde:

c: Ceros de la función de transferencia.

p: Polos de la función de transferencia.

Despejado de la ecuación (3.9):

$$\begin{cases} u(t) = \ddot{z}(t) + p_1 \cdot \dot{z}(t) \\ y(t) = \ddot{z}(t) + (c_1 + c_2) \cdot \dot{z}(t) + c_1 c_2 \cdot z(t) \end{cases} \quad (3.10)$$

Haciendo la siguiente conversión de variables:

$$\begin{cases} z_1 = z \\ z_2 = \dot{z} = \dot{z}_1 \\ z_3 = \ddot{z} = \dot{z}_2 \end{cases} \quad (3.11)$$

Dando como resultado el siguiente modelo en espacio de estados en su forma canónica para el controlador:

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -p_1 \end{bmatrix} \cdot \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot u(t)$$

$$y(t) = [c_1 c_2 \quad c_1 + c_2 - p_1] \cdot \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + u(t) \quad (3.12)$$

4. PÉNDULO INVERTIDO

4.1. Introducción

El sistema del péndulo invertido es un problema clásico en ingeniería de control. Consta de una masa en uno de los extremos de una varilla que se mueve bidimensionalmente, el extremo contrario de la varilla se encuentra unido a un carro que puede moverse longitudinalmente.

4.2. Dinámica del péndulo invertido

El problema del péndulo invertido resulta interesante por su complejidad al ser un sistema inestable, ya que la varilla puede caer hacia cualquier dirección dentro del plano X-Y. Con el objetivo de evitar que el péndulo caiga, hay que diseñar un controlador que se encargue de mantener la varilla con la masa en posición vertical, para ello se tendrá el control de la fuerza aplicada sobre el carro, sabiendo en cada instante el ángulo del péndulo.

A continuación, se describe el modelo matemático que rige el funcionamiento del sistema, presentando la fuerzas que actúan y que mantienen en equilibrio el sistema, para luego desarrollar sus ecuaciones e implementar un controlador adecuado para la estabilización del sistema.

Durante el desarrollo del modelo matemático del péndulo invertido se va a despreciar algunas características del sistema. La inercia de la varilla (no la del peso), o la fricción (rozamiento) del carro, para simplificar el sistema.

4.2.1. Movimiento del carro

El carro permanece en reposo cuando no se le aplica ninguna fuerza externa cumpliendo así la primera ley de Newton. No obstante, cuando se le aplica una fuerza externa el carro se moverá con una aceleración, obedeciendo la segunda ley de Newton:

$$\sum F = m \cdot a \tag{4.1}$$

Donde:

F: Es la fuerza aplicada.

m: Es la masa del carro.

a: Es la aceleración.

4.2.2. Mecánica del péndulo

Las ecuaciones que rigen el movimiento del péndulo invertido con un extremo fijo a un pivote son similares a las de un péndulo simple. Una salvedad es que el péndulo invertido no sufre ninguna resistencia al movimiento. El péndulo invertido produce la ecuación para un oscilador armónico:

$$\ddot{\theta} + \frac{g}{l} \cdot \sin\theta = 0 \tag{4.2}$$

Donde:

$\ddot{\theta}$: Aceleración angular.

g: Constante de la gravedad.

l: Longitud de la varilla.

θ: Desplazamiento angular desde la posición de equilibrio.

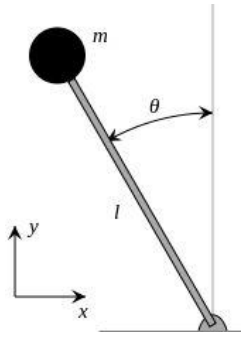


Figura 4.1: Péndulo invertido

4.3. Descripción del sistema

El péndulo invertido puede ser descrito por la siguiente figura:

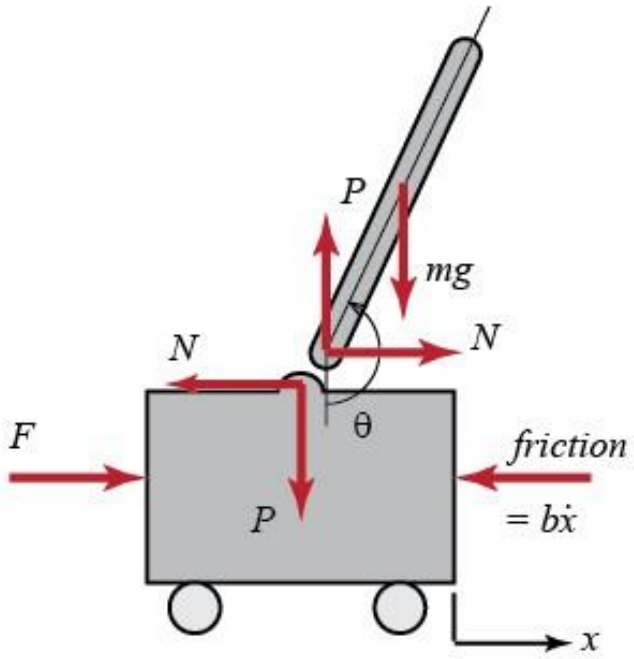


Figura 4.2: Descripción del péndulo invertido

Para afrontar el problema se tomará los siguientes valores:

Datos:	Símbolo	Unidad
Masa del carro	M	1 kg
Masa del péndulo	m	0,1 kg
Aceleración de la gravedad	g	9.8 m/s^2
Coeficiente de fricción del carro	b	-
Longitud del centro de péndulo	l	0,3 m
Momento de inercia del péndulo	I	-
La fuerza aplicada al carro	F	Entrada de control
Posición del carrito	x	salida
Angulo de péndulo desde la vertical	Ɵ	salida

Tabla 4.1: Datos de péndulo invertido

El contenido de los siguientes apartados está adaptado a partir de *Control Tutorials for Matlab & Simulink* contenido disponible en: *http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling*

Para el análisis de las fuerzas, sumando las fuerzas sobre el carro en la dirección horizontal del diagrama de cuerpo libre:

$$M\ddot{x} + b\dot{x} + N = F \tag{4.3}$$

De la misma manera sumando las fuerzas aplicadas horizontalmente sobre el péndulo, se obtiene la expresión para N:

$$N = m\ddot{x} - ml\dot{\Theta}^2 \sin\Theta + ml\ddot{\Theta}\cos\Theta \tag{4.4}$$

Sustituyendo la ecuación (4.4) en la ecuación (4.3) se obtiene:

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\Theta}\cos\Theta - ml\dot{\Theta}^2 \sin\Theta = F \tag{4.5}$$

Para la obtención de la segunda ecuación del movimiento es necesario sumar las fuerzas perpendiculares al péndulo:

$$Psin\,\Theta + Ncos\,\Theta + mgsin\,\Theta = ml\ddot{\Theta} + m\ddot{x}\cos\Theta \tag{4.6}$$

Para eliminar los termino P y N de la ecuación anterior, se suman los momentos alrededor del centroide del péndulo, dando como resultado la siguiente ecuación:

$$Psin\,\Theta + Ncos\,\Theta = -\frac{\ddot{\Theta}I}{l} \tag{4.7}$$

Sustituyendo la ecuación (4.7) en la ecuación (4.6) se obtiene la siguiente expresión:

$$(I + ml^2)\ddot{\Theta} - mglsin\,\Theta = -ml\ddot{x}\cos\Theta \tag{4.8}$$

Reorganizando las ecuaciones hasta el momento quedaría en las siguientes ecuaciones:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\Theta}\cos\Theta - ml\dot{\Theta}^2\sin\Theta = F \tag{4.9}$$

$$(I + ml^2)\ddot{\Theta} - mglsin\,\Theta = -ml\ddot{x}\cos\Theta \tag{4.10}$$

4.4. Linealización del sistema

Como las ecuaciones obtenidas del péndulo invertido son no-lineales y el sistema es inestable, lo que hace compleja la manipulación de las ecuaciones, se recurre a la linealización del sistema. La linealización de las ecuaciones se toma respecto a la posición equilibrio verticalmente hacia arriba, en $\Theta = 0$, y que el sistema se mantendrá dentro de una pequeña zona cercana al punto de equilibrio.

La pequeña desviación en torno al punto de equilibrio vendrá dada por la variable φ, es decir, la variación en torno al punto de equilibrio quedará de la siguiente forma: $\Theta = 0 + \varphi$. A continuación, hay que calcular las derivadas parciales para la linealización del sistema:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\Theta}\cos\Theta - ml\dot{\Theta}^2\sin\Theta - F = 0 \tag{4.11}$$

$$f1\left(\ddot{x}, \dot{x}, \ddot{\Theta}, \dot{\Theta}, \Theta, F\right) = 0 \tag{4.12}$$

$$\left.\frac{\partial f1}{\partial \ddot{x}}\right|_0 \Delta\ddot{x} + \left.\frac{\partial f1}{\partial \dot{x}}\right|_0 \Delta\dot{x} + \left.\frac{\partial f1}{\partial \ddot{\Theta}}\right|_0 \Delta\ddot{\Theta} + \left.\frac{\partial f1}{\partial \dot{\Theta}}\right|_0 \Delta\dot{\Theta} + \left.\frac{\partial f1}{\partial \Theta}\right|_0 \Delta\Theta + \left.\frac{\partial f1}{\partial F}\right|_0 \Delta F = 0$$

$$\left.\frac{\partial f1}{\partial \ddot{x}}\right|_0 \Delta\ddot{x} = (M+m);$$

$$\left.\frac{\partial f1}{\partial \dot{x}}\right|_0 \Delta\dot{x} = b;$$

$$\left.\frac{\partial f1}{\partial \ddot{\Theta}}\right|_0 \Delta\ddot{\Theta} = ml;$$

$$\left.\frac{\partial f1}{\partial \dot{\Theta}}\right|_0 \Delta\dot{\Theta} = 0;$$

$$\left.\frac{\partial f1}{\partial \Theta}\right|_0 \Delta\Theta = 0;$$

$$\left.\frac{\partial f1}{\partial F}\right|_0 \Delta F = 1;$$

La ecuación linealizada tiene la siguiente forma:

$$(M+m)\ddot{x} + b\dot{x} + ml\,\ddot{\varphi} = F \qquad (4.13)$$

De la misma forma, las derivadas parciales de la segunda ecuación para su linealización son las siguientes:

$$(I+ml^2)\ddot{\Theta} + mglsin\,\Theta = \; ml\ddot{x}\cos\Theta \qquad (4.14)$$

$$f2(\ddot{x}, \ddot{\Theta}, \Theta) = 0 \tag{4.15}$$

$$\left.\frac{\partial f2}{\partial \ddot{x}}\right|_0 \Delta\ddot{x} + \left.\frac{\partial f2}{\partial \ddot{\Theta}}\right|_0 \Delta\ddot{\Theta} + \left.\frac{\partial f2}{\partial \Theta}\right|_0 \Delta\Theta = 0$$

$$\left.\frac{\partial f2}{\partial \ddot{x}}\right|_0 \Delta\ddot{x} = ml;$$

$$\left.\frac{\partial f2}{\partial \ddot{\Theta}}\right|_0 \Delta\ddot{\Theta} = (I + ml^2);$$

$$\left.\frac{\partial f2}{\partial \Theta}\right|_0 \Delta\Theta = -mgl;$$

$$(I + ml^2)\,\ddot{\varphi} - mgl\,\varphi = ml\ddot{x} \tag{4.16}$$

Finalmente, como resultado queda las siguientes ecuaciones linealizadas:

$$(M + m)\ddot{x} + b\dot{x} + ml\,\ddot{\varphi} = F \tag{4.17}$$

$$(I + ml^2)\,\ddot{\varphi} - mgl\,\varphi + ml\ddot{x} = 0 \tag{4.18}$$

Es interesante transformar las ecuaciones anteriores a representación interna ya que se adaptan mejor para el trabajo en Matlab.

4.4.1 Representación en espacio de estados

De las ecuaciones (4.17) y (4.18) se sabe que el valor de fricción y de inercia son nulos, por lo que haciendo las conversiones adecuadas y tomando el vector de estados como: $(x_1, x_2, x_3, x_4) = (x, \dot{x}, \varphi, \dot{\varphi})$, da como resultado las siguientes ecuaciones:

$$\dot{x}_1 = x_2 \; ; \; \dot{x}_2 = -\frac{mg}{M} \cdot x_3 + \frac{1}{M} \cdot u$$
$$\dot{x}_3 = x_4 \; ; \; \dot{x}_4 = \frac{g}{l} \cdot x_3 - \frac{1}{M \cdot l} \cdot u \tag{4.19}$$

Por lo tanto, la matriz en espacio de estados del sistema es:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{l} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} u \tag{4.21}$$

La matriz de salida:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{4.22}$$

A partir de las ecuaciones anteriores, compilando el archivo de Matlab P_I_Lineal.m, devuelve la función de transferencia del péndulo y el carro. Por consiguiente, con la ecuación (4.23) se obtiene el mismo resultado:

$$G(s) = C(sI - A)^{-1}B \tag{4.23}$$

Aplicando la ecuación anterior se obtiene como resultado las siguientes ecuaciones:

$$G_{\Theta}(s) = \frac{-3{,}33}{s^2 - 32.66} \tag{4.24}$$

$$G_x(s) = \frac{s^2 - 35{,}92}{s^4 - 32.66s^2} \tag{4.25}$$

Ambas ecuaciones tienen una respuesta inestable por lo que es necesario la búsqueda de un controlador. Se va a diseñar un controlador que estabilice el péndulo aplicando el lugar de las raíces. Hay que acentuar que al ser un sistema SIMO (simple-input Multipleoutput) es muy complicado encontrar un controlador para el péndulo que sea capaz de controlar el carro este proceso requiere de sistemas más complejos como el LQR.

4.5. Diseño del controlador

Para el diseño del controlador se ha usado una herramienta de Matlab, Sisotool que se utiliza para comprobar la respuesta de la función del péndulo que se observar en la (Figura 4.3), se comprueba que la respuesta es inestable en lazo abierto:

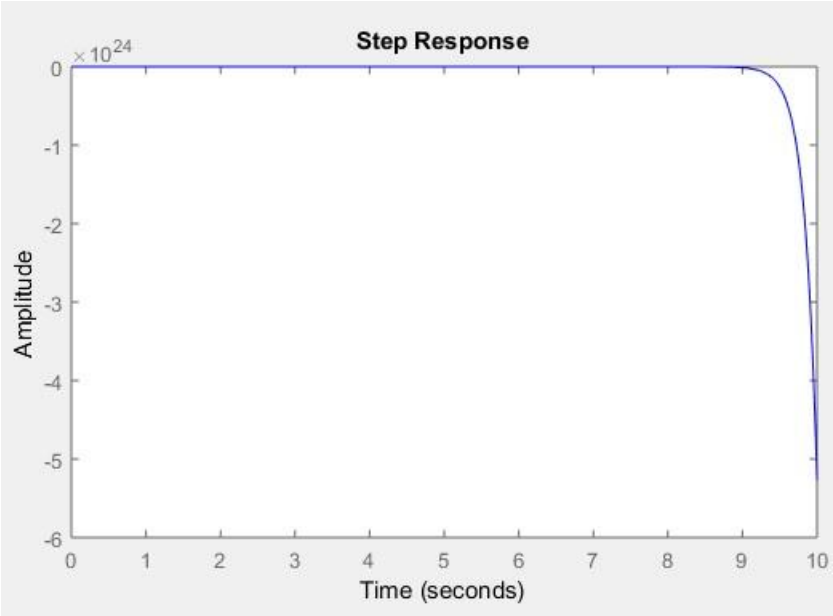


Figura 4.3: Respuesta en bucle abierto

Para evitar la inestabilidad es necesario un controlador que estabilice el sistema. De nuevo con la ayuda de la herramienta de Matlab, Sisotool, por medio de la representación del lugar de las raíces de la función de transferencia se obtiene un controlador adecuado (Figura 4.4), la función del controlador se puede observar en la ecuación (4.26):

$$C(s) = \frac{-121.4(s+7.5)(s+2)}{s(s+30)} \tag{4.26}$$

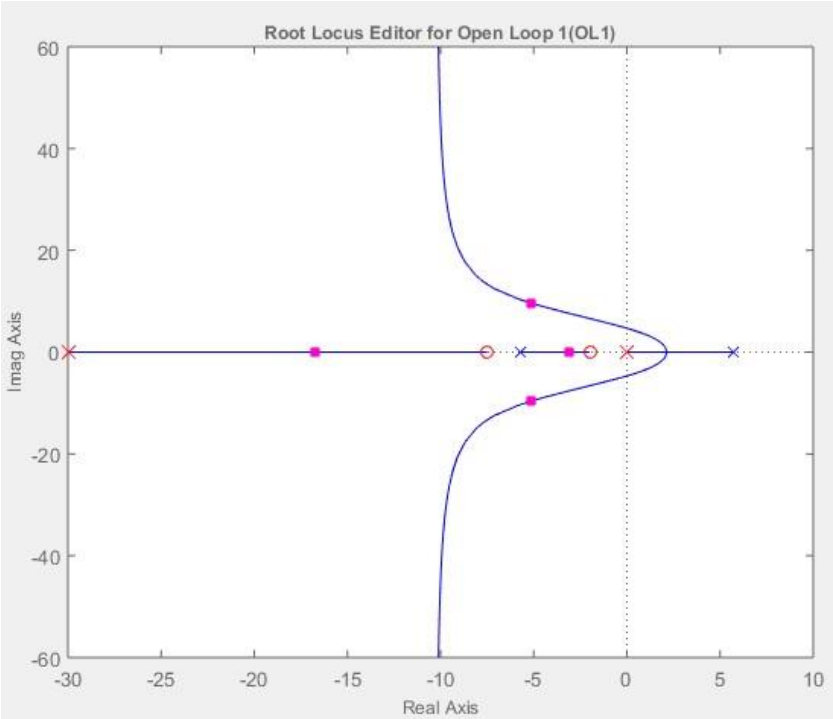


Figura 4.4: Representación de función de transferencia y controlador en Sisotool.

Una vez diseñado el controlador se procede a su implementación en el modelo linear y el modelo no-lineal para comprobar que es el adecuado, para ellos se utiliza la herramienta Simulink que se encuentra también en Matlab.

El bloque C viene dado por el compesador exportado desde Sisotool donde ya contiene la ganancia negativa del sistema, por otra parte, en el (Anexo I) viene la estructura del bloque péndulo invertido no lineal.

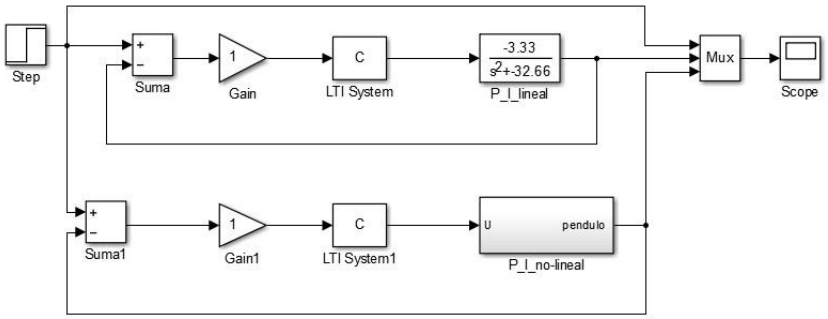


Figura 4.5: Péndulo invertido lineal y no-lineal en Simulink

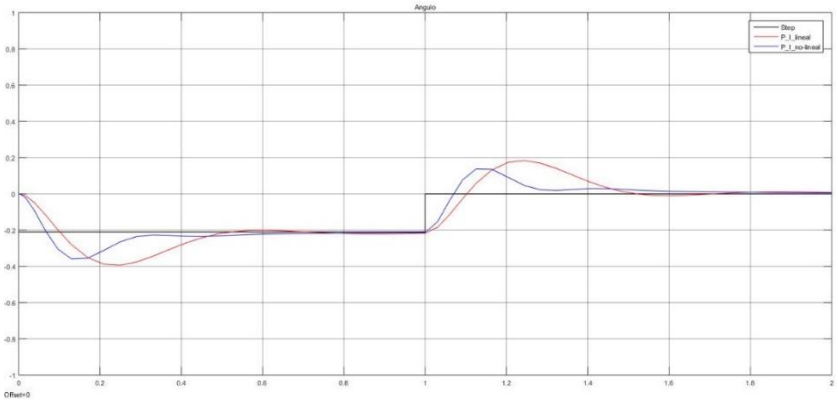


Figura 4.6: Respuesta del modelo lineal y no-lineal (Simulink)

Se observa que la respuesta del modelo lineal y el modelo no lineal son parecidas, por lo que se procede a su implementación en EjsS.

4.6. Implementación en EjsS

En este punto se desarrolla la interface del péndulo invertido. En el panel *Descripción* del programa viene una introducción del sistema lineal y no lineal:

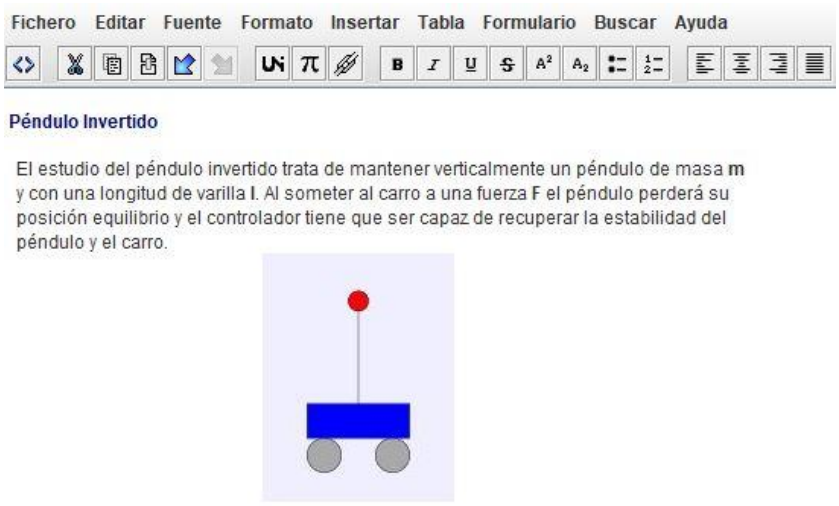
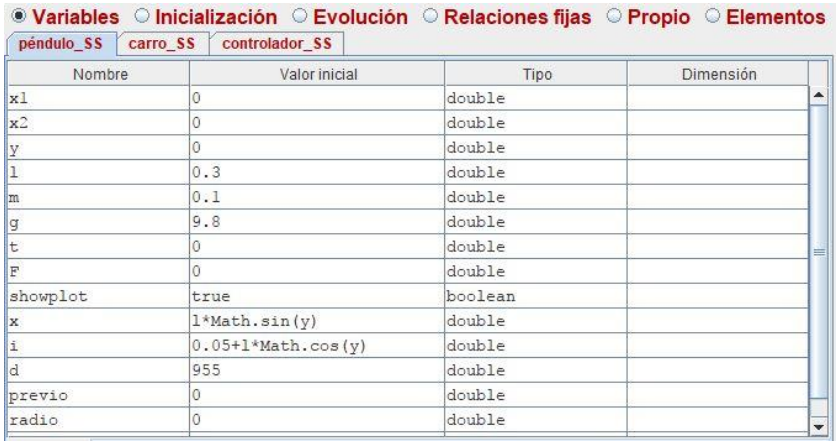


Figura 4.7: Descripción del péndulo invertido

4.6.1. Modelo lineal

En el panel *Modelo* se desarrolla el cuerpo de la simulación y presenta de seis pestañas. La pestaña *Variables* dispone de tres ventanas: variables del péndulo, carro y controlador.



Nombre	Valor inicial	Tipo	Dimensión
x1	0	double	
x2	0	double	
y	0	double	
l	0.3	double	
m	0.1	double	
g	9.8	double	
t	0	double	
F	0	double	
showplot	true	boolean	
x	l*Math.sin(y)	double	
i	0.05+l*Math.cos(y)	double	
d	955	double	
previo	0	double	
radio	0	double	

Figura 4.8: Variables del péndulo invertido (lineal)

Declaración de variables del péndulo invertido

x1, x2: Variables de estado del sistema.

y: Valor del ángulo del péndulo.

l: Longitud de la varilla.

m: Masa del péndulo.

t: Tiempo de simulación.

F: Fuerza externa aplicada en al carro.

Showplot: Variable que permite al usuario visualizar la gráfica de control.

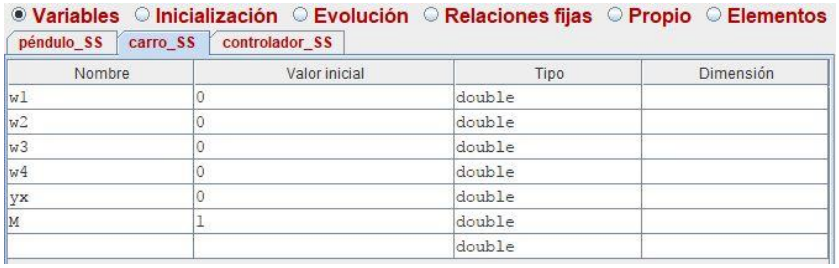
x: Variable para el movimiento en X del péndulo.

i: Variable para el movimiento en Y del péndulo.

d: Densidad de la esfera.

previo: Cálculo intermedio para la obtención del radio de la esfera.

radio: Cálculo del radio de la esfera.



Variables | Inicialización | Evolución | Relaciones fijas | Propio | Elementos

péndulo_SS | carro_SS | controlador_SS

Nombre	Valor inicial	Tipo	Dimensión
w1	0	double	
w2	0	double	
w3	0	double	
w4	0	double	
yx	0	double	
M	1	double	
		double	

Figura 4.9: Variables del carro (lineal)

Declaración de variables del carro

w1, w2, w3, w4: Variables de estado del sistema.

yx: Posición del carro.

M: Masa del carro.

Variables | Inicialización | Evolución | Relaciones fijas | Propio | Elementos

péndulo_SS | carro_SS | controlador_SS

Nombre	Valor inicial	Tipo	Dimensión
z1	0	double	
z2	0	double	
u	0	double	
K	-121.4	double	
c1	7.5	double	
c2	2	double	
p1	30	double	
r	0	double	

Figura 4.10: Variables del controlador (lineal)

Declaración de variables del controlador

z1, z2: Variables de estado del controlador.

u: Variable de salida del controlador.

K: Ganancia del sistema y el controlador.

c1, c2: Ceros del controlador.

p1: Polo del controlador.

r: Referencia del sistema.

En la pestaña *Inicialización* se realizan unos cálculos iniciales antes del comienzo de la simulación:

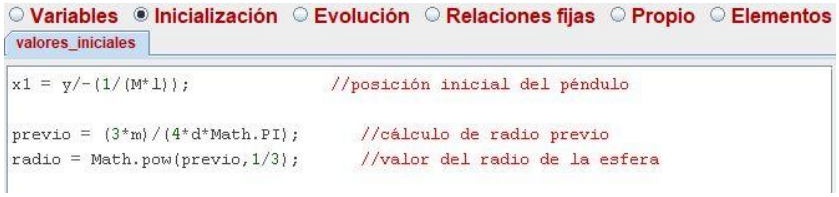


Figura 4.11: Inicialización (lineal)

El primer cálculo que presenta la pestaña de *Inicialización* es la posición inicial del péndulo. Además del cálculo del radio de la esfera del péndulo, este cálculo se usa también en otras pestañas.

La pestañ *Evolución* permite implementar las ecuaciones de estado en páginas EDOs, al tener las ecuaciones diferenciales de primer orden obtenidas por el método en espacio de estados en su forma canónica el uso de páginas EDOs es lo más conveniente.

El programa permite crear simultáneamente varias páginas EDOs pero no es posible utilizar la misma variable independiente en ambas páginas EDOs.

Para cualquier tipo de simulación en EjsS se debe prestar mucha atención al paso de integración que se está utilizando porque puede provocar que la simulación no funcione en tiempo real. El problema se soluciona ajustando el IPS (Imágenes por segundo) y el PPV (Paso por visualización).

Las IPS son el número de fotogramas que un dispositivo muestra por segundo y el PPV hace referencia al número de paso que avance antes de producirse la visualización.

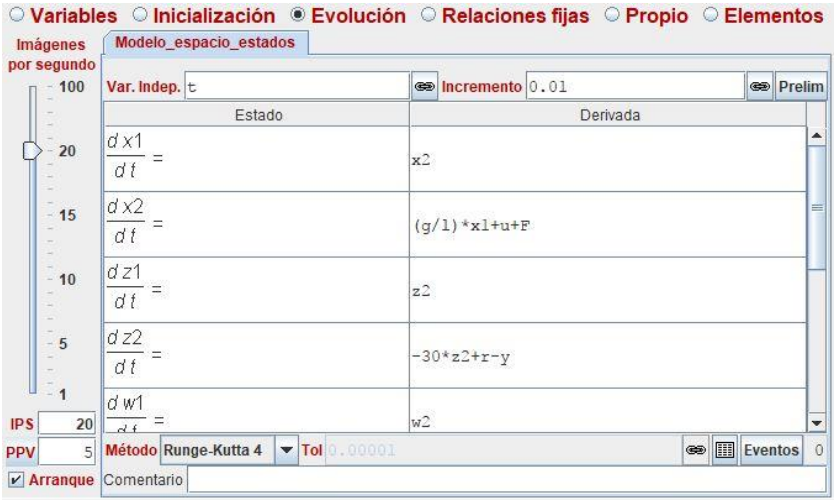


Figura 4.12: Evolución (lineal)

Para que la simulación con un incremento de 0.05 simule (aproximadamente) en tiempo real los IPS puede tener un valor de 20 y PPV de 1, pero en esta simulación utilizar un valor 0.01, es decir, 5 veces menor por lo que solo hace falta modificar PPV a 5.

La pestaña *Relaciones fijas* sirve como apoyo a la pestaña *Evolución.* Aquí se utilizan tres ventanas donde se realizan los cálculos de salida del péndulo, carro y controlador:

Figura 4.13: Relaciones fijas salida del péndulo (lineal)

En esta ventana de salida del péndulo se está recalculando la posición del péndulo mediante las variables *x* e *i.* A partir de la variable *y* se obtiene el valor de salida del péndulo. Las variables *previo, radio* sirve para el cálculo el radio de la esfera en el caso de que este se modifique durante la simulación.

Figura 4.14: Relaciones fijas salida del carro (lineal)

En esta ventana se incluye *yx* que se encarga de dar la posición del carro en todo momento y además se condiciona el reinicio de la simulación cuando el péndulo invertido sale de la pantalla.

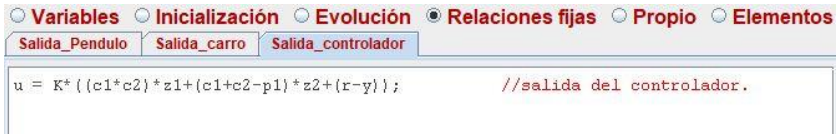


Figura 4.15: Relaciones fijas salida del controlador (lineal)

En la última ventana que se dispone en *Relaciones fijas* se encuentra el cálculo valor de salida del controlador.

Por ultimo antes de pasar al diseño gráfico de la simulación se encuentra la pestaña, *Propio,* donde se crea un método para poder interactuar el usuario y poder modificar la posición del péndulo con el puntero.

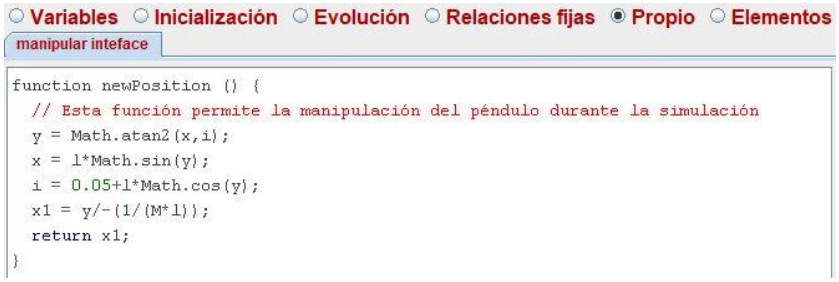


Figura 4.16: Método propio para interactuar con la simulación (lineal)

La ultima pestaña *HtmlView* se encuentra el diseño gráfico de la interface:

Figura 4.17: Diseño gráfico del modelo lineal

Finalmente, el aspecto de la página web donde se produce la simulación se presenta de la siguiente forma:

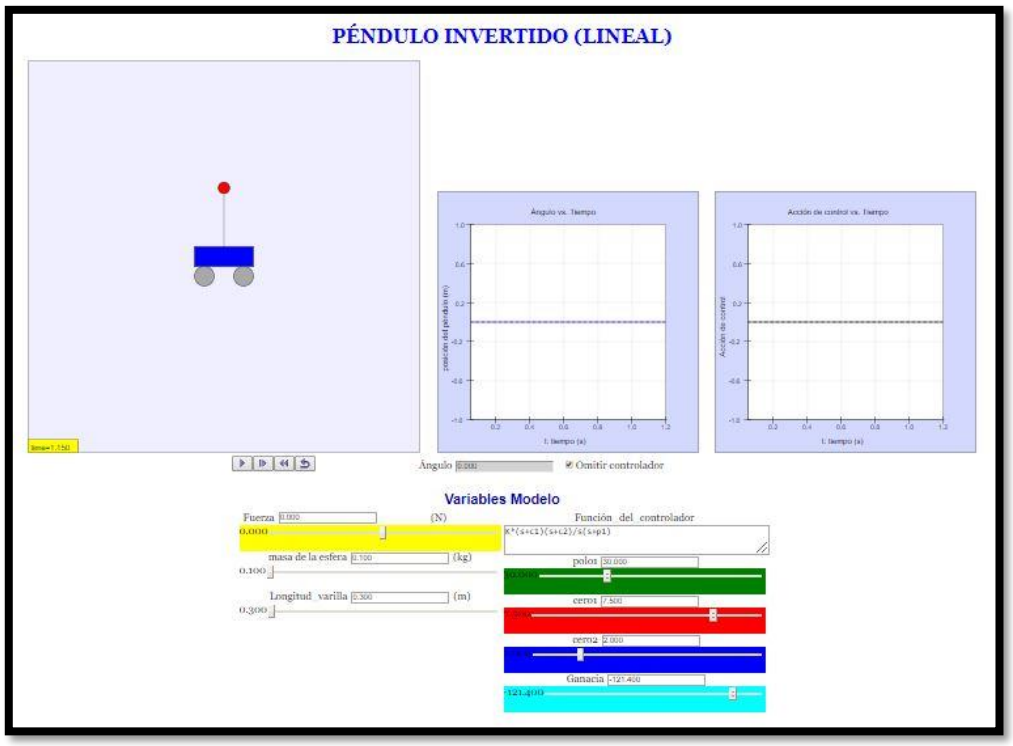


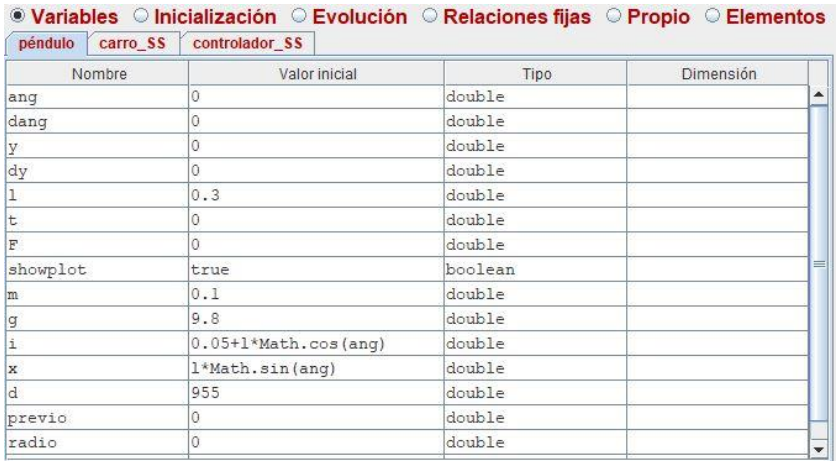
Figura 4.18: Visualización final de la interface del modelo lineal

La simulación permite al usuario modificar el tamaño de la esfera y la longitud de la varilla, así como los valores del controlador permitiendo observar el resultado en las gráficas.

Las gráficas muestran el ángulo del péndulo y la respuesta del controlador permitiendo al usuario poder ocultar la gráfica del controlador.

4.6.2. Modelo no lineal

La pestaña *Variable* presenta tres ventanas: variables del péndulo, carro y controlador.



Nombre	Valor inicial	Tipo	Dimensión
ang	0	double	
dang	0	double	
y	0	double	
dy	0	double	
l	0.3	double	
t	0	double	
F	0	double	
showplot	true	boolean	
m	0.1	double	
g	9.8	double	
i	0.05+l*Math.cos(ang)	double	
x	l*Math.sin(ang)	double	
d	955	double	
previo	0	double	
radio	0	double	

Figura 4.19: Variables del péndulo (no lineal)

Declaración de variables de péndulo invertido

ang: Ángulo del péndulo.

dang: Velocidad angular.

y: Posición del carro.

dy: Velocidad del carro.

l: Longitud de la varilla.

t: Tiempo de simulación.

F: Fuerza externa aplicada al carro.

Showplot: Variable que permite al usuario visualizar la gráfica de control.

m: Masa del péndulo.

g: Constante de la gravedad.

x: Variable para el movimiento en X del péndulo.

i: Variable para el movimiento en Y del péndulo.

d: Densidad de la esfera.

previo: Cálculo intermedio para la obtención del radio de la esfera.

radio: Cálculo del radio de la esfera.

Nombre	Valor inicial	Tipo	Dimensión
w1	0	double	
w2	0	double	
w3	0	double	
w4	0	double	
yx	0	double	
M	1	double	
		double	

Figura 4.20: Variables del carro (no lineal)

Declaración de variables del carro

w1, w2, w3, w4: variables de estado del sistema.

yx: posición del carro.

M: Masa del carro.

Nombre	Valor inicial	Tipo	Dimensión
z1	0	double	
z2	0	double	
u	0	double	
c1	7.5	double	
c2	2	double	
p1	30	double	
K	-121.4	double	
r	0	double	

Figura 4.21: Variables del controlador (no lineal)

Declaración de variables del controlador

z1, z2: Variables de estado del controlador.

u: Variable de salida del controlador.

K: Ganancia del sistema y el controlador.

c1, c2: Ceros del controlador.

p1: Polo del controlador.

r: Referencia del sistema.

En la pestaña *Evolución* vienen recogidas las ecuaciones no lineales del péndulo invertido y del controlador. En cada estado utilizado en *Evolución* se deriva los métodos definidos en la pestaña *Propio* donde se encuentra las ecuaciones del modelo. Esta pestaña también incluye la evolución del carro en variables de estado en su forma canónica.

En la parte superior de la ventana se puede observar la variable independiente t y el paso de integración 0.01 es igual al tomado en el modelo lineal. Con el paso de integración de 0.01 para que se produzca la simulación en tiempo real hace falta IPS de 20 y PPV de 5.

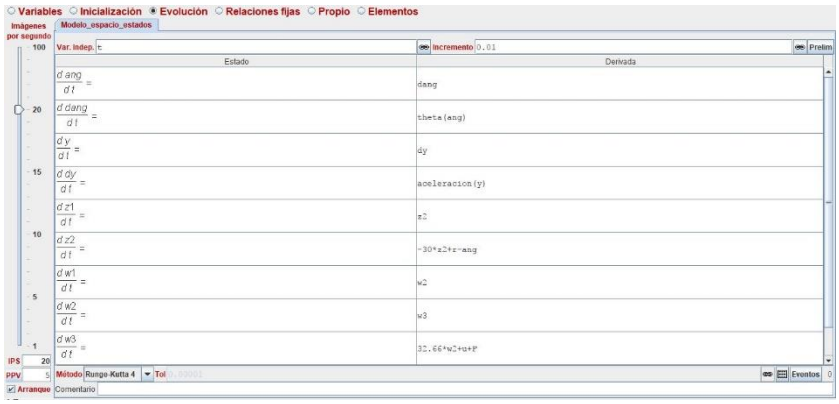


Figura 4.22: Evolución (no lineal)

En la pestaña *Relaciones fijas* se produce el cálculo la salida del controlador. La posición del carro y del ángulo se calculan en *Evolución*.

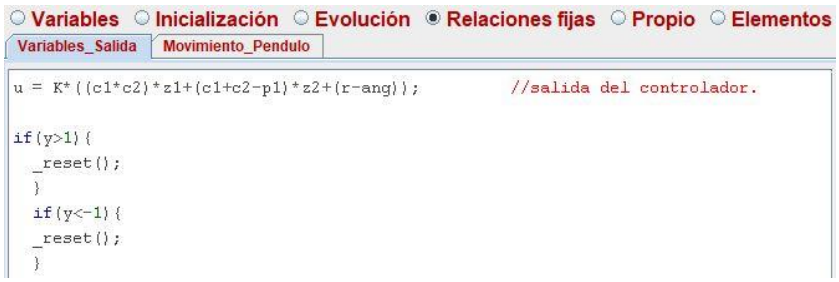


Figura 4.23: Relaciones fijas salida del controlador (no lineal)

En la (Figura 4.23) se realizan los cálculos del valor de salida del controlador como se ha comentado anteriormente, así como un límite establecido para el que cuando el carro salga de la pantalla de visualización gráfica se reinicie la simulación.

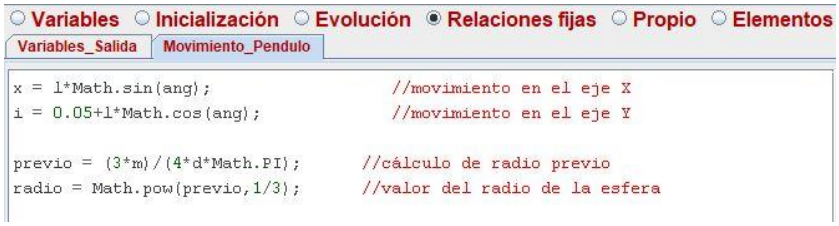


Figura 4.24: Relaciones fijas cálculos para el péndulo (no lineal)

En la pestaña *Propio* se crean tres métodos: uno para el cálculo del carro y otro para cálculo del ángulo del péndulo invertido como se puede observar en las (Figuras 4.25 y 4.26).

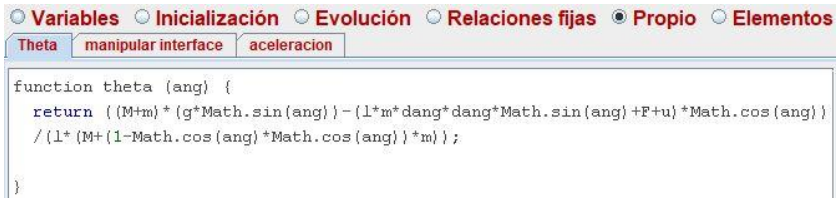


Figura 4.25: Propio función ángulo (no lineal)

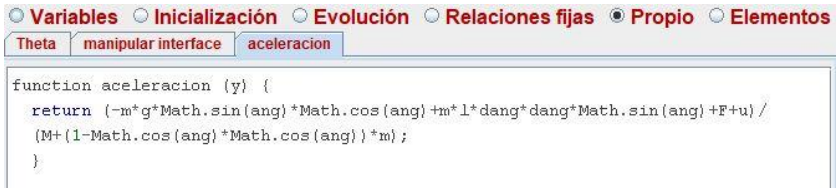


Figura 4.26: Propio función movimiento del carro (no lineal)

Por último, el método propio restante al igual que en el péndulo invertido lineal sirve para que el usuario pueda interactuar con la interface gráfica y poder modificar la posición del péndulo con el puntero.

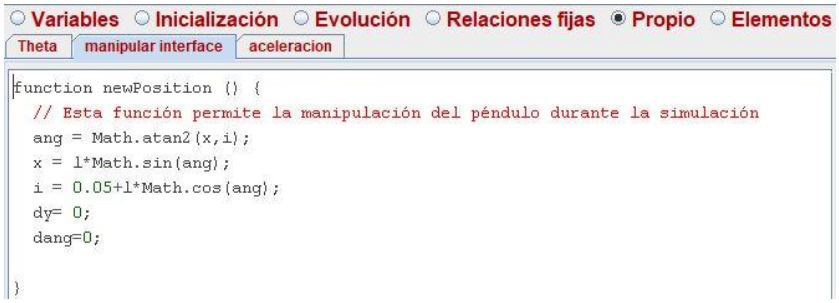


Figura 4.27: Método propio para interactuar con la simulación (no lineal)

En la (Figura 4.27) las variables de *x* e *i* se encargan de calcular la posición del péndulo. La variable *ang* se calcula mediante el comando Math.atan2(x,i) el cual se encarga de convertida las variables *x* e *i* de coordenadas rectangulares a coordenadas polares para facilitar el movimiento del péndulo.

La ultima pestaña *HtmlView* se encuentra el diseño gráfico de la interface:

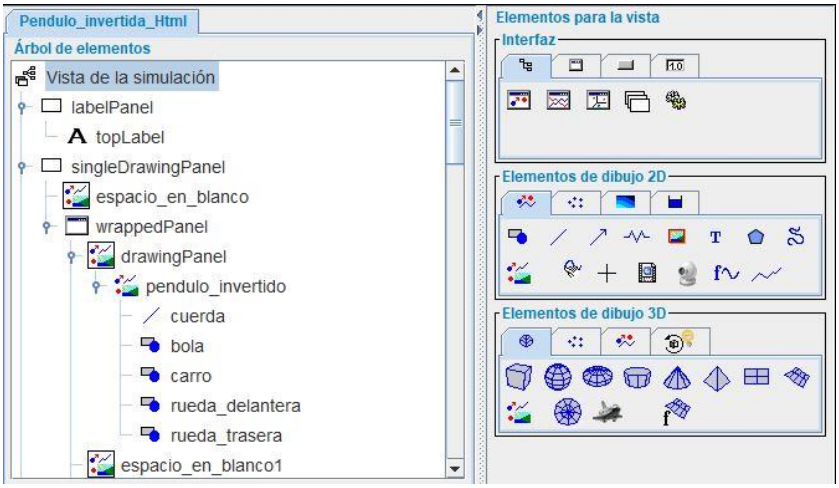


Figura 4.28: Diseño gráfico del modelo no lineal

Por último, la visualización de la página web donde se produce la simulación tiene el siguiente aspecto:

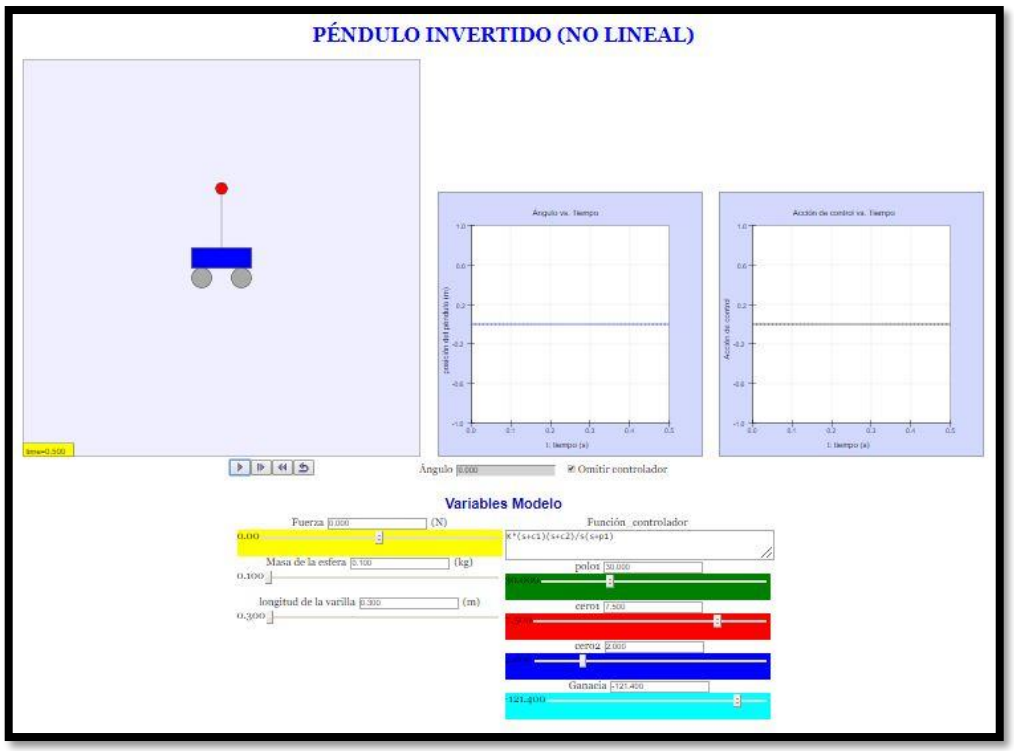


Figura 4.29: Visualización final de la interface del modelo no lineal

La simulación permite al usuario modificar el tamaño de la esfera y la longitud de la varilla, así como los valores del controlador permitiendo observar el resultado en las gráficas.

Las gráficas muestran el ángulo del péndulo y la respuesta del controlador permitiendo al usuario poder ocultar la gráfica del controlador.

5. LEVITADOR MAGNÉTICO

5.1. Introducción

El estudio del levitador magnético resulta un problema complejo, donde están implicadas diversas fuerzas que hay que tener presentes en el modelo matemático, las fuerzas que afectan al sistema se pueden citar como: fuerza mecánicas, eléctricas y magnéticas que se van analizar a lo largo del capítulo.

5.2. Dinámica del levitador magnético

El levitador magnético consiste en mantener un objeto suspendido en el aire sin ningún tipo de contacto mecánico. La fuerza que permite la suspensión del objeto es la fuerza electromagnética que se opone a la fuerza peso manteniendo el objeto suspendido en una posición de equilibrio, la fuerza electromagnética depende de la inducción del electroimán producido por el paso de la intensidad que le aporta el circuito eléctrico.

5.2.1. Modelo eléctrico

El comportamiento que modela el circuito eléctrico del levitador magnético es un circuito RL. Aplicando al circuito la ley de Kirchoff se obtiene la siguiente relación de tensión e intensidad:

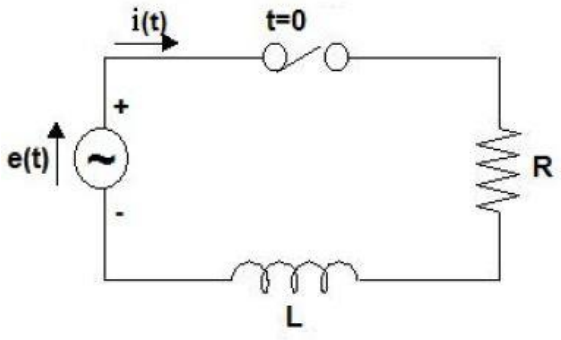


Figura 5.1: Circuito RL

$$e(t) = R \cdot i(t) + L(x) \cdot \frac{di(t)}{dt} \tag{5.1}$$

Donde *e(t)* es la tensión a la que se somete el circuito, R y L el valor de resistencia y bobina respectivamente, y *i(t)* es la intensidad que circula por el circuito.

5.2.2. Fuerza electromagnética

La fuerza magnética se produce por el movimiento de partículas cargadas, en el caso del levitador magnético debido al paso de la corriente eléctrica por la bobina se crea un campo magnético. La ecuación de la energía en un condensador que viene dada aplicando las leyes de Ampere y Faraday es la siguiente:

$$\overrightarrow{W_m} = \frac{1}{2}\vec{L} \cdot i(t)^2 \tag{5.2}$$

Por lo que la fuerza electromagnética se obtiene del gradiente de energía magnética en las tres direcciones espaciales:

$$\vec{f}_{mag} = \Delta\overrightarrow{W_m} = (\frac{d\overrightarrow{W_{mx}}}{dx}, \frac{d\overrightarrow{W_{my}}}{dy}, \frac{d\overrightarrow{W_{mz}}}{dz}) \tag{5.3}$$

En el caso del levitador magnetico de este modelo solo se estudiará el caso en una dirección:

$$\vec{f}_{mag} = \frac{i(t)^2}{2}\frac{dL(x)}{dt} \tag{5.4}$$

El valor de L(x) viene dado por $L_m + L_s$, siendo Lm la inductancia asociada a la bobina y L_s la asociada a la esfera, cuando el valor de *x* tiene a infinito el termino L_s se aproxima a un valor nulo.

La curva caracteristica del sistema es:

$$L(x) = L_m + \frac{x_0 \cdot L_s}{x} \tag{5.5}$$

Asumiendo que el valor de *x* se encuentra en el punto de equilibrio x_0 y que L_s es mucho menor que L_m, se puede hacer la siguiente aproximación para el valor de inductancia:

$$L(x) \approx L_m \tag{5.6}$$

Por tanto, la ecuacion que determina la fuerza electromagnética tiene la siguiente forma:

$$\vec{f}_{mag} = -\frac{L_s x_0}{2}\frac{i^2}{x^2} = -K\frac{i^2}{x^2} \tag{5.7}$$

Siendo $K = \frac{L_s x_0}{2}$.

Donde, *K* es la constante de acoplo magnético, *i(t)* es la intensidad que circula por el electroimán, y *x(t)* es la posicion del objeto.

5.2.3. Mecánica

La mecánica es la rama de la física que se encarga del estudio del movimiento de un sistema y las fuerza a las que esta sometido.

El estudio del levitador magnético se basa en la aplicación de la segunda ley de Newton que nos indica las fuerza a las que esta sometido el objeto que se encuentra suspendido en el aire. Esta ecuacion cumple el siguiente equilibrio de fuerzas:

$$\sum \vec{F} = m \cdot \vec{a} \tag{5.8}$$

Como el modelo solo tiene un grado de libertad y que solamente se estudia la variación de la altura, se obtiene la siguiente expresión:

$$m\frac{d^2h}{dt^2} = m \cdot g - f_{mag} \tag{5.9}$$

Donde *h* indica la altura del objeto, f_{mag} es la fuerza electromagnética generada por el electroimán, y donde se toma el peso $(m \cdot g)$ como positivo, dado que la

referencia se toma en la base del electroimán y el eje positivo en la dirección descendente.

5.3. Descripción del modelo

El levitador magnético viene representado por la siguiente figura:

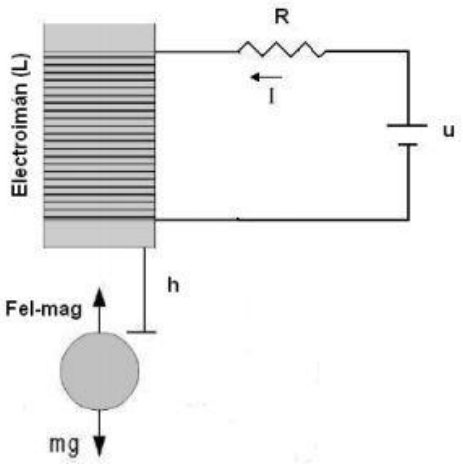


Figura 5.2: Levitador Magnético

Dados los apartados anteriores, se puede determinar las ecuaciones de las que depende el modelo del levitador magnético:

$$\frac{d^2h}{dt^2} = g - \frac{k}{m} \cdot \left(\frac{i}{h}\right)^2 \tag{5.10}$$

$$v = R \cdot i + L\frac{di}{dt} \tag{5.11}$$

Para un mejor desarrollo del modelado del sistemas se procede a la representacion de las ecuaciones en espacio de estados.

5.4. Linealización del sistema

El uso de la linealización es necesario para trabajar con las ecuaciones de una forma más sencilla porque permite trabajar en torno a un punto en el que el sistema se encuentra en equilibrio. Derivando parcialmente las ecuaciones (5.10) (función f) y (5.11) (función g) se obtiene:

$$\begin{aligned} f\left(i, i, h, \dot{h}, u\right) &= 0 \\ g\left(i, i, h, \dot{h}, u\right) &= 0 \end{aligned} \tag{5.12}$$

Evaluado en los puntos de operación: $i = i^*,\ i = 0,\ h = h^*,\ \dot{h} = 0,\ v = v^*$.

$$\left.\frac{\partial f}{\partial i}\right|_{op} \Delta i + \left.\frac{\partial f}{\partial i}\right|_{op} \Delta i + \left.\frac{\partial f}{\partial h}\right|_{op} \Delta h + \left.\frac{\partial f}{\partial \dot{h}}\right|_{op} \Delta \dot{h} = 0$$

$$\left.\frac{\partial f}{\partial i}\right|_{i^*} \Delta i = -\frac{2k}{m}\frac{i}{h^2};$$

$$\left.\frac{\partial f}{\partial h}\right|_{h^*} \Delta h = \frac{2k}{m}\frac{i^2}{h^3};$$

$$\left.\frac{\partial f}{\partial v}\right|_{v^*} \Delta v = 0;$$

$$\left.\frac{\partial g}{\partial i}\right|_{op} \Delta i + \left.\frac{\partial g}{\partial i}\right|_{op} \Delta i + \left.\frac{\partial g}{\partial h}\right|_{op} \Delta h + \left.\frac{\partial g}{\partial \dot{h}}\right|_{op} \Delta \dot{h} = 0$$

$$\left.\frac{\partial g}{\partial i}\right|_{i^*} \Delta i = -\frac{R}{L};$$

$$\left.\frac{\partial g}{\partial v}\right|_{v^*} \Delta v = \frac{1}{L};$$

Finalmente, las ecuaciones linealizadas presentan la siguiente forma:

$$\begin{cases} \dot{h} = -\dfrac{2k}{m}\dfrac{i^*}{h^{2*}} + \dfrac{2k}{m}\dfrac{i^{2*}}{h^{3*}} \\ i^* = \dfrac{v}{L} - \dfrac{R}{L}i^* \end{cases} \tag{5.13}$$

El valor de las variables evaluadas en el punto de operación se calcula con las siguientes ecuaciones:

$$\dot{h}^* = 0 \ \rightarrow 0 = g - \frac{k}{m}\left(\frac{i^*}{h^*}\right)^2 \rightarrow h^* = \sqrt{\frac{k}{gm}}\, i^* \tag{5.14}$$

$$v = v^* \rightarrow i^* = \frac{v^*}{R}$$

5.4.1. Representación en espacio de estados

Con el desarrollo de las ecuaciones lineales que modelan la dinámica del levitador magnético se procede a la transformación en espacio de estado, lo cual es necesario conocer cuáles son las variables de estado, las variables de entrada y las variables de salida.

El levitador magnético tiene únicamente una variable de entrada y otra salida (sistema SISO), son *v* y *h* respectivamente.

A partir de las ecuaciones lineales del sistema la representación en espacio de estados permite expresar las ecuaciones de forma matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \dfrac{df_1}{dx_1} & \dfrac{df_1}{dx_2} & \dfrac{df_1}{dx_3} \\ \dfrac{df_2}{dx_1} & \dfrac{df_2}{dx_2} & \dfrac{df_2}{dx_3} \\ \dfrac{df_3}{dx_1} & \dfrac{df_3}{dx_2} & \dfrac{df_3}{dx_3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \dfrac{df_1}{du} \\ \dfrac{df_2}{du} \\ \dfrac{df_3}{du} \end{bmatrix} u \tag{5.15}$$

Realizando el siguiente cambio de variables:

$$\begin{cases} x_1 = h \\ x_2 = \dfrac{dh}{dt} \\ x_3 = i \end{cases} \tag{5.16}$$

Da lugar a las siguientes matrices de espacio de estados:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{2k}{m}\dfrac{x_3^{2*}}{x_1^{3*}} & 0 & -\dfrac{2k}{m}\dfrac{x_3^*}{x_1^{2*}} \\ 0 & 0 & -\dfrac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{1}{L} \end{bmatrix} u \tag{5.17}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Las variables evaluadas en el punto de operación se calculan con las siguientes ecuaciones:

$$x_2^* = 0 \ \rightarrow 0 = g - \frac{k}{m}\left(\frac{x_3^*}{x_1^*}\right)^2 \rightarrow \ x_1^* = \sqrt{\frac{k}{gm}}\ x_3^* \tag{5.18}$$

$$u = \ u^* \rightarrow x_3^* = \ \frac{u^*}{R}$$

En la tabla 5.1 viene los valores que se han tomado:

Datos:	Símbolo	Unidad
Resistencia eléctrica	R	2 Ω
Inductancia de la bobina	L	0,0713 H
Aceleración de la gravedad	g	9,8 m/s^2
Masa del objeto	m	0,05 kg
Intensidad en el punto de operación	x_3^*	7.5 A
Tensión en el punto de operación	u^*	15 V
Altura en el punto de operación	x_1^*	0,5357 m
Velocidad en el punto de operación	x_2^*	0 m/s
Constante de acoplo magnético	k	0.0025 N*m^2/A^2

Tabla 5.1: Parámetros del sistema de levitación

La matriz de estados queda de la siguiente forma:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 36.5857 & 0 & -2.6133\\ 0 & 0 & -28.0505\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 14.0252\end{bmatrix}u$$
$$y(t) = [1 \quad 0 \quad 0]\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{5.19}$$

Simulando el archivo de Matlab: L_M_lineal.m, se obtener la función de transferencia de matriz anterior:

$$G(s) = \frac{-36.65}{s^3 + 28.05s^2 - 36.59s - 1026} \tag{5.20}$$

Valiéndose de Simulink se compara el comportamiento del modelo lineal y el modelo no lineal. La (Figura 5.3) muestra el esquema de bloques en Simulink de ambos sistemas:

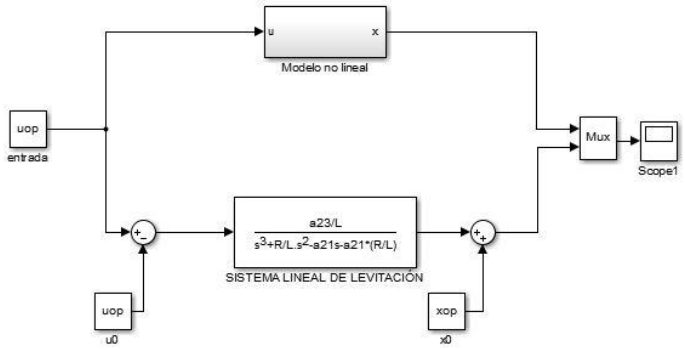


Figura 5.3: Esquema de los modelo lineal y no lineal en Simulink

En la (Figura 5.4) se puede observar el valor de la altura en el punto de operación. También se puede ver que los dos modelos toman los mismos valores a lo largo del tiempo, por lo tanto, la linealización se ha realizado correctamente:

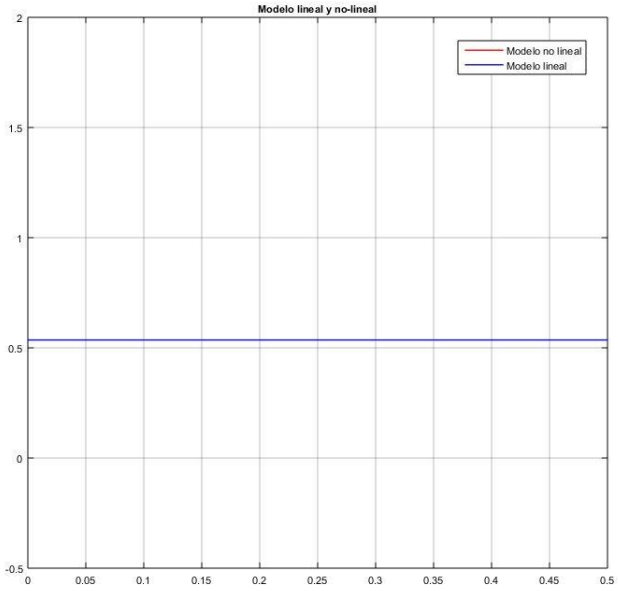


Figura 5.4:Comparación modelo lineal y no-lineal.

5.5. Diseño del controlador

La búsqueda de un controlador adecuado se lleva a cabo mediante el estudio del lugar de las raíces en Sisotool para comprobar su estabilidad.

La siguiente imagen presenta el lugar de las raíces de la función de transferencia (5.20) y se observa que para cualquier valor de ganancia el sistema es inestable.

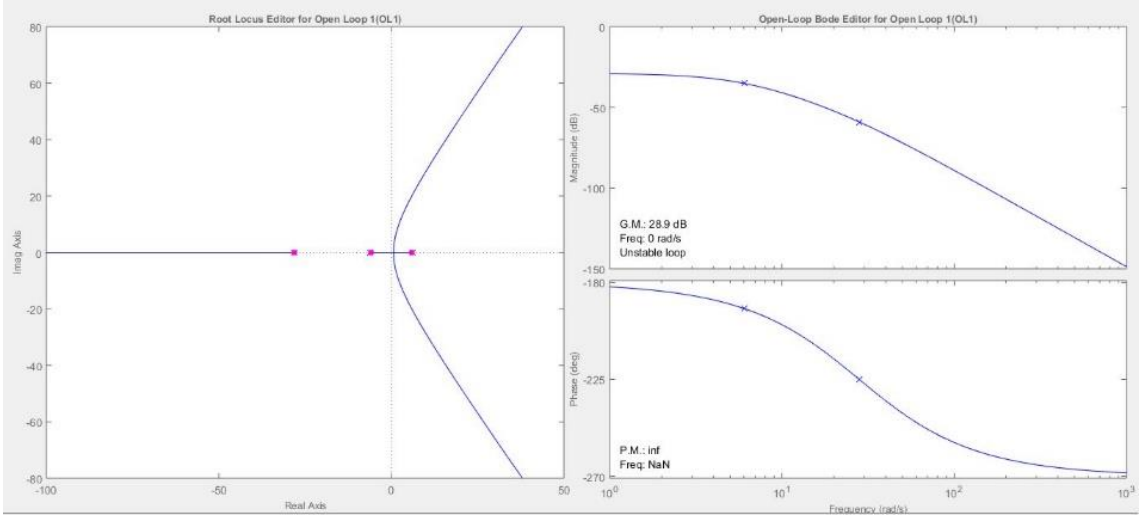


Figura 5.5: Lugar de las raíces del levitador magnético

La respuesta antes una entrada escalón es la siguiente:

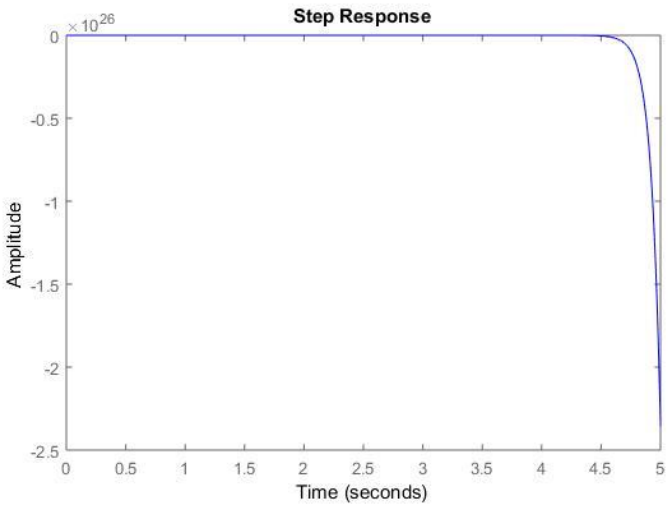


Figura 5.6: inestabilidad ante una respuesta escalón

En la (Figura 5.6) se comprueba que la respuesta es inestable en bucle cerrado. Con la ayuda de Sisotool se realiza la búsqueda del controlador indicado que proporcione a la salida una respuesta estable.

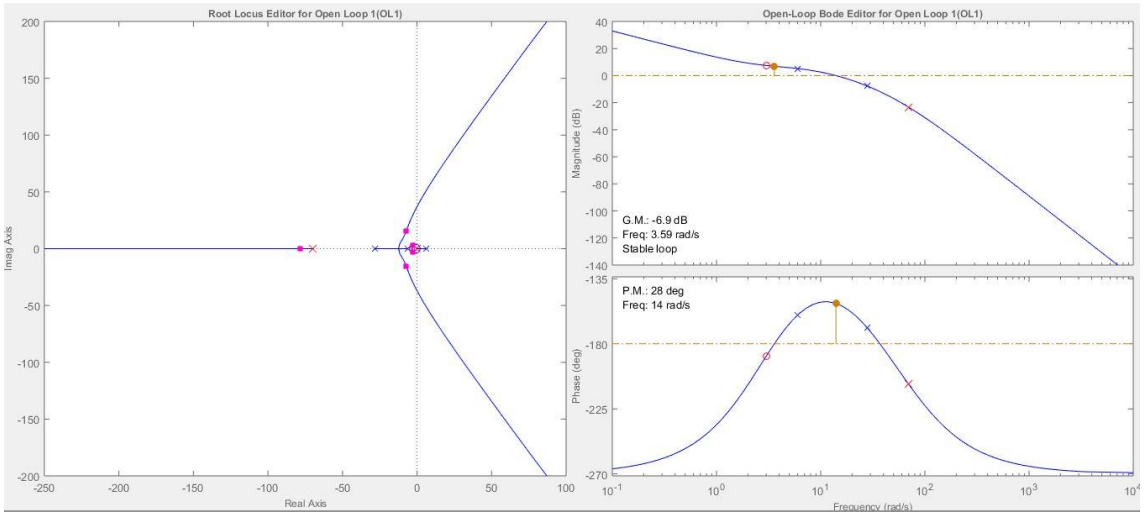


Figura 5.7: Lugar de las raíces insertando el controlador

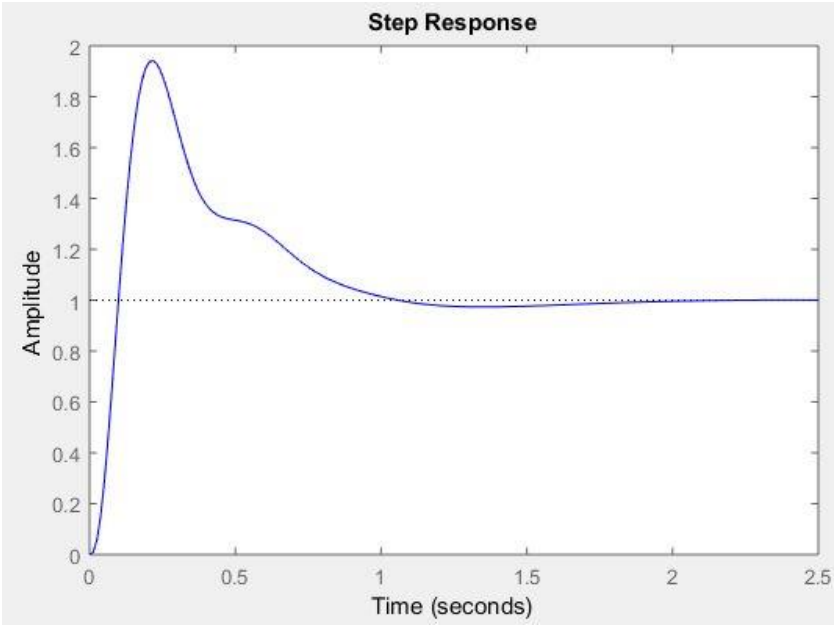


Figura 5.8: Respuesta G(s) y C(s) ante una entrada escalón

En la (Figura 5.8) se puede observar que la respuesta del sistema es estable en lazo abierto. La función del controlador se puede observar en la ecuación (5.21):

$$G_C(s) = \frac{-970.93 \cdot (s+3)^2}{s \cdot (s+70)} \qquad (5.21)$$

Para una posterior implementación en EjsS es necesario la transformación de la función $G_c(s)$ de la ecuación (5.21) y la G(s) de la ecuación (5.20) usando el método en espacio de estados en su forma canónica. Comenzando con la función G(s) se obtiene las siguientes matrices de estado en forma canónica:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1026 & 36.59 & -28.05\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u \tag{5.22}$$

$$y = [-36.65 \quad 0 \quad 0]\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{5.23}$$

Dando como resultado las siguientes ecuaciones lineales de primer orden que interpreta la evolución de levitador magnético en EjsS:

$$\begin{cases}\dot{x}_1 = x_2\\ \dot{x}_2 = x_3\\ \dot{x}_3 = 1026 \cdot x_1 + 35.59 \cdot x_2 - 28.05 \cdot x_3 + u\\ y = -36.65 \cdot x_1\end{cases} \tag{5.24}$$

De la misma manera se aplica para la función de controlador:

$$\begin{bmatrix}\dot{Z}_1\\ \dot{Z}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -70\end{bmatrix}\begin{bmatrix}Z_1\\ Z_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u \tag{5.26}$$

$$y(t) = [9 \quad -64]\begin{bmatrix}Z_1\\ Z_2\end{bmatrix} + u \tag{5.27}$$

Como el controlador presenta un polo en el origen se resuelve de la siguiente manera:

$$\begin{cases} \dot{Z}_1 = Z_2 \\ \dot{Z}_2 = -70 \cdot Z_2 + u \\ u = -970.93 \cdot (9 \cdot Z_1 - 64 \cdot Z_2 + u) \end{cases} \quad (5.28)$$

Una vez obtenidas estas ecuaciones se puede proceder al desarrollo de la interface en EjsS.

5.6. Implementación en EjsS

El desarrollo de la interface comienza con un resumen del funcionamiento del levitador magnético donde se expone las ecuaciones del modelo. Esta descripción aparece tanto en el modelo lineal como en el no lineal en la pestaña *Descripción* de la aplicación:

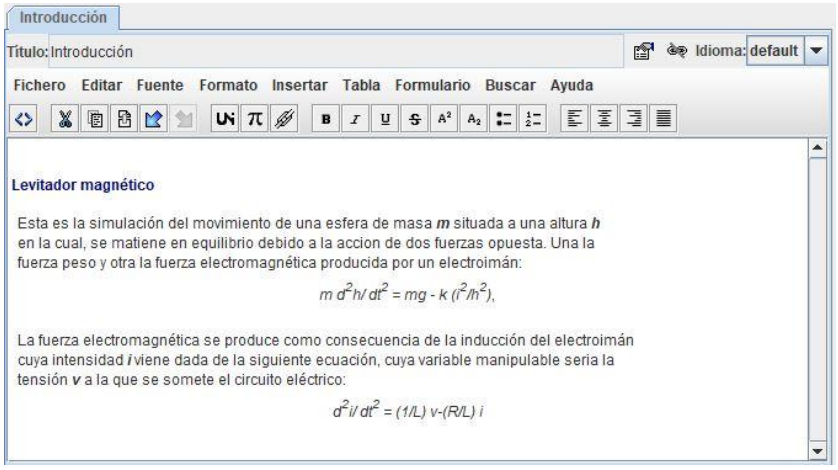
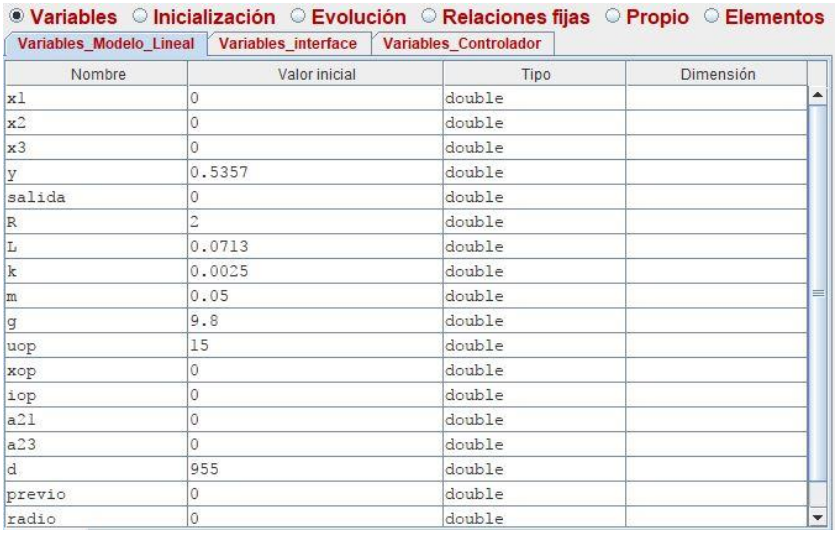


Figura 5.9: Descripción del levitador magnético

5.6.1. Modelo lineal

En la pestaña *Variable* del panel *Modelo* se declaran las variables que se utilizan en la simulación. Se utilizan tres ventanas: variables del modelo lineal, variables de la interface y variables del controlador:



Nombre	Valor inicial	Tipo	Dimensión
x1	0	double	
x2	0	double	
x3	0	double	
y	0.5357	double	
salida	0	double	
R	2	double	
L	0.0713	double	
k	0.0025	double	
m	0.05	double	
g	9.8	double	
uop	15	double	
xop	0	double	
iop	0	double	
a21	0	double	
a23	0	double	
d	955	double	
previo	0	double	
radio	0	double	

Figura 5.10: Variables modelo (lineal)

Declaración de variables modelo lineal:

x1, x2, x3: Variables de estado del sistema.

y: Altura de la masa suspendida.

salida: Respuesta que proporciona el sistema.

i: Intensidad eléctrica del circuito.

R: Valor de resistencia eléctrica.

L: Valor de inductancia de la bobina.

k: Constante de acoplo magnético.

m: Masa de la esfera.

g: Aceleración de la gravedad.

uop: Tensión eléctrica del circuito donde la esfera se encuentra en equilibrio.

xop: Valor de altura de la esfera en el punto de operación.

iop: Valor de intensidad de la esfera en el punto de operación.

a21: Cálculo intermedio ((2*k)/m)*((iop*iop)/(xop*xop*xop)).

a23: Cálculo intermedio -((2*k)/m)*((iop)/(xop*xop));.

d: Densidad de la esfera.

previo: Cálculo intermedio para la obtención del radio de la esfera.

radio: Cálculo del radio de la esfera.


Variables Inicialización Evolución Relaciones fijas Propio Elementos

Variables_Modelo_Lineal | Variables_interface | Variables_Controlador

Nombre	Valor inicial	Tipo	Dimensión
t	0	double	
showreference	true	boolean	
showplot	true	boolean	
dt	0.005	double	

Figura 5.11: Variables de la interface (lineal)

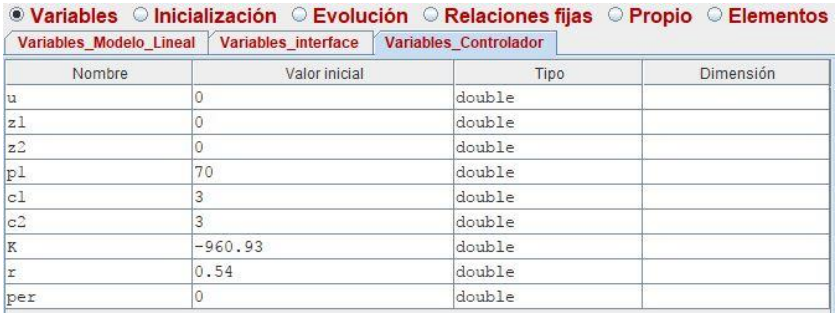
Declaración de variables de la interface:

t: Tiempo de simulación.

dt: Incremento de tiempo de cada paso de simulación.

showreference: Variable que permite al usuario visualizar la referencia.

showplot: Variable que permite al usuario visualizar la gráfica de control.


Variables Inicialización Evolución Relaciones fijas Propio Elementos

Variables_Modelo_Lineal | Variables_interface | Variables_Controlador

Nombre	Valor inicial	Tipo	Dimensión
u	0	double	
z1	0	double	
z2	0	double	
p1	70	double	
c1	3	double	
c2	3	double	
K	-960.93	double	
r	0.54	double	
per	0	double	

Figura 5.12: Variables del controlador (lineal)

Declaración de variables del controlador:

u: Variable de salida del controlador.

z1, z2: Variables de estado del controlador.

p1: Polo del controlador.

c1, c2: Ceros del controlador.

K: Ganancia del sistema y el controlador.

r: Referencia del sistema.

per: Fuerza o perturbación externa aplicada a la esfera.

Para el punto de *Inicialización* del *modelo* se recogen algunos cálculos previos antes de comenzar la evolución de la simulación como el valor de altura e intensidad en el punto de equilibrio y se le asigna inicialmente la posición de equilibrio, así como unos cálculos necesarios para facilitar la modificación de las variables durante la simulación como son *a21, a23, previo* y *radio*. Este último permite la modificación del tamaño de la esfera al igual que en el péndulo invertido.

Por otro lado, se usa la variable salida para un cálculo intermedio que se realiza en propio permitiendo la manipulación de la altura de la esfera durante la simulación.

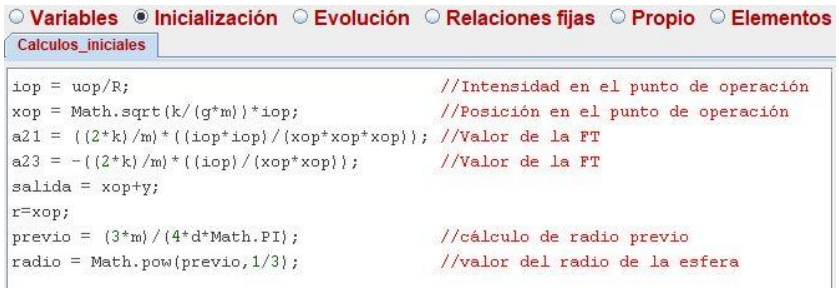


Figura 5.13: Inicialización del modelo (lineal)

En la *Evolución* se introduce las ecuaciones diferenciales del modelo lineal y las del controlador. Como se observa se introduce las ecuaciones de primer orden obtenidas por el método en espacio de estados en su forma canónica.

Como el incremento que se ha tomado para la simulación ha sido de un valor de 0.005 se ha tenido que ajusta las imágenes por segundo y los pasos por visualización para que simule en tiempo real, para ellos se ha tomado un IPS de 20 y un PPV de 10.

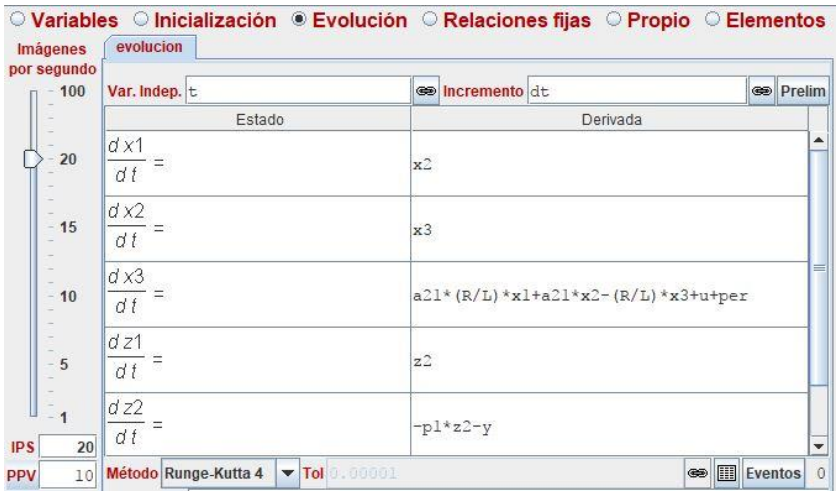


Figura 5.14: Evolución del modelo (lineal)

En el panel *Relaciones fijas* se crean dos ventanas una para los cálculos del modelo y otra para el valor de salida del controlador:

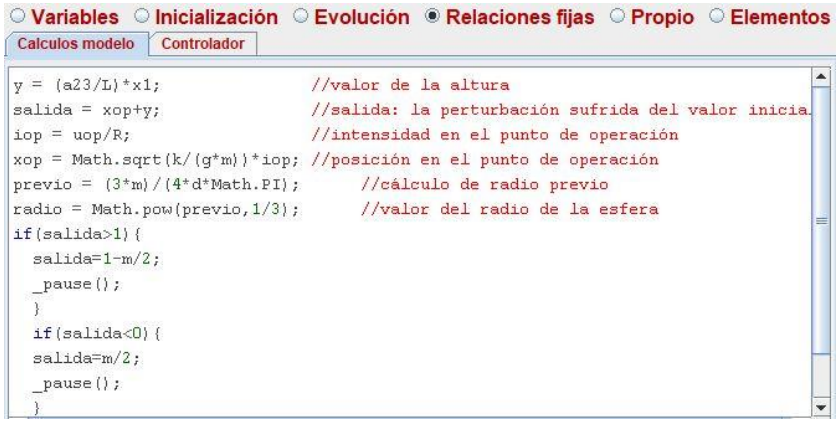


Figura 5.15: Relaciones fijas cálculos del modelo (lineal)

En la (Figura 5.15) se recalculan por cada paso de simulación los valores *iop, xop, previo* y *radio* para permitir al usuario modificar dichos valores durante la simulación.

Por otra parte, se realiza un cálculo adicional con variable *salida* permitiendo posteriormente modificar la posición de la esfera durante la simulación. También se condiciona el rango de movimiento del que dispone la esfera siendo el valor 1 la base del electroimán y 0 el suelo.

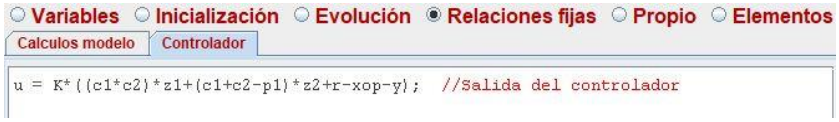


Figura 5.16: Relaciones fijas salida del controlador (lineal)

En la (Figura 5.16) se obtiene la salida del controlador.

En la pestaña *Propio* se crea un método que permita al usuario modificar la posición de la esfera. La variable que se modifica es el valor x1 a que se encuentra en la pestaña *Evolución*, además es necesario utilizar la variable adicional salida y restarle la posición en el punto de equilibrio ya que si se utilizará la variable *y* directamente no produciría ninguna modificación:

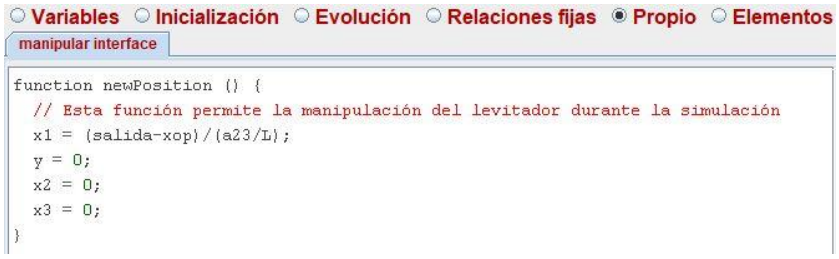


Figura 5.17: Método propio para interactuar con la simulación (no lineal)

Para la finalización de la simulación cabe destacar el último punto donde viene el desarrollo de la interface gráfica en la pestaña *HtmlView*:

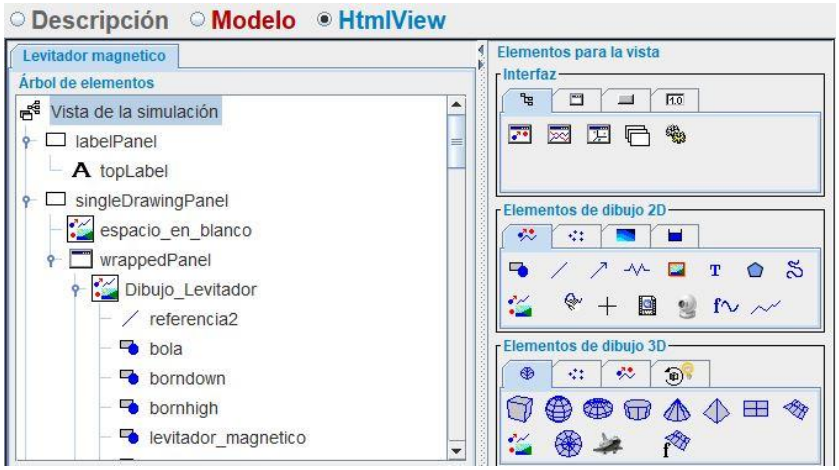


Figura 5.18: Diseño gráfico del modelo lineal

Finalmente, al ser el lenguaje de programación JavaScript al realizar la simulación se abre una página web, donde se encuentra todo el desarrollo realizado en la ventana *HtmlView.*

Además, esta página permite interactuar al usuario dándole la posibilidad de poder parar, reiniciar o reanudar la simulación, así como aplicarle una perturbación al sistema para ver cómo se comporta el sistema y el controlador diseñado.

También permite al usuario omitir la gráfica del controlador, el valor de referencia. Es posible modificar cualquier variable del modelo y observar un comportamiento distinto dentro de los límites que el controlador permita, si se supera la respuesta del modelo se hará inestable:

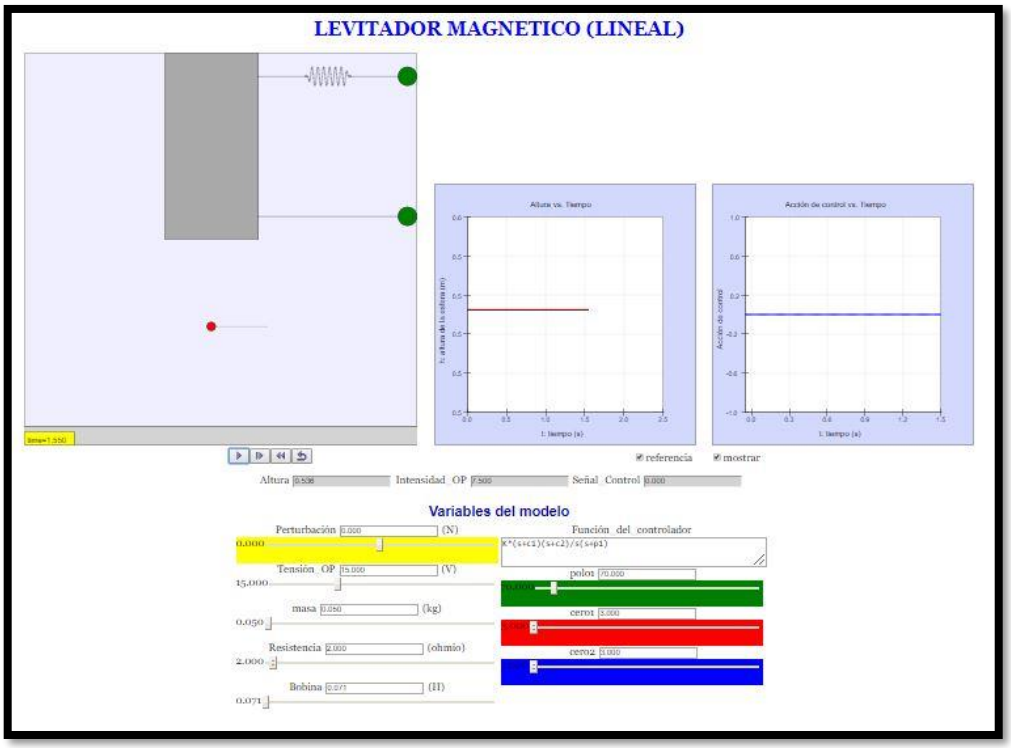


Figura 5.19: Visualización final de la interface del modelo lineal

5.6.2. Modelo no lineal

En primer lugar, se declaran las variables del modelo no lineal. Se usan tres pestañas de *Variables*: variables de modelo no lineal, variables de la interface y variables del controlador:

Variables | Inicialización | Evolución | Relaciones fijas | Propio | Elementos

Modelo_No_Lineal | Variable_interface | Variables_de_control

Nombre	Valor inicial	Tipo	Dimensión
h	0.5	double	
vel	0	double	
v	15	double	
i	0	double	
k	0.0025	double	
g	9.8	double	
L	0.0713	double	
R	2	double	
m	0.05	double	
xop	0	double	
iop	0	double	
d	955	double	
previo	0	double	
radio	0	double	

Figura 5.20: Variables del levitador magnético (no lineal)

Declaración de variables del modelo no lineal:

h: Altura de la masa.

vel: Velocidad de la masa.

v: Tensión eléctrica del circuito.

i: Intensidad eléctrica del circuito.

k: Constante de acoplo magnético.

g: Aceleración de la gravedad.

L: Valor de inductancia de la bobina.

R: Valor de resistencia eléctrica.

m: Masa de la esfera.

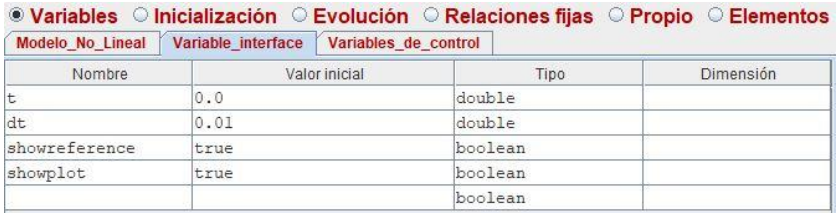
xop: Valor de altura de la esfera en el punto de operación.

iop: Valor de intensidad de la esfera en el punto de operación.

d: Densidad de la esfera.

previo: Cálculo intermedio para la obtención del radio de la esfera.

radio: Cálculo del radio de la esfera.



Nombre	Valor inicial	Tipo	Dimensión
t	0.0	double	
dt	0.01	double	
showreference	true	boolean	
showplot	true	boolean	
		boolean	

Figura 5.21: Variables de la interface (no lineal)

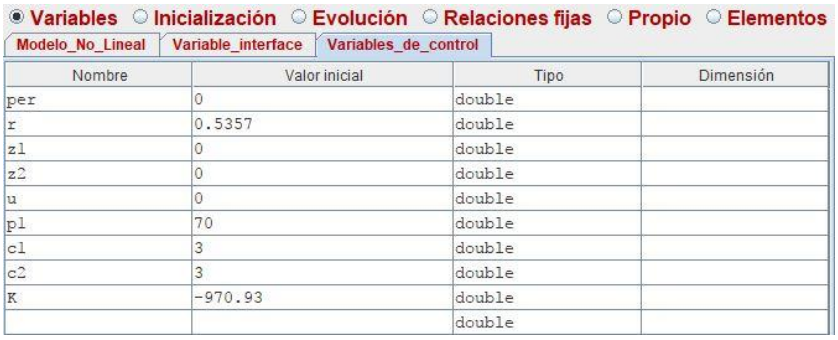
Declaración de variables de la interface:

t: Tiempo de simulación.

dt: Incremento de tiempo de cada paso de simulación.

showreference: Variable que permite al usuario visualizar la referencia.

showplot: Variable que permite al usuario visualizar la gráfica de control.



Nombre	Valor inicial	Tipo	Dimensión
per	0	double	
r	0.5357	double	
z1	0	double	
z2	0	double	
u	0	double	
p1	70	double	
c1	3	double	
c2	3	double	
K	-970.93	double	
		double	

Figura 5.22: Variables del controlador (no lineal)

Declaración de variables del controlador:

per: Fuerza o perturbación externa aplicada a la esfera.

r: Referencia del sistema.

z1, z2: Variables de estado del controlador.

u: Variable de salida del controlador.

K: Ganancia del sistema y el controlador.

c1, c2: Ceros del controlador.

p1: Polo del controlador.

Para el punto de *Inicialización* del *modelo* viene recogido algunos cálculos previos a la evolución de la simulación como el valor de altura e intensidad en el punto de equilibrio y se le asigna inicialmente la posición de equilibrio.

Por otro lado, se realiza un cálculo previo del tamaño de la esfera que posteriormente dicho cálculo se estará utilizando en otra ventana del programa para poder variar el tamaño de la esfera.

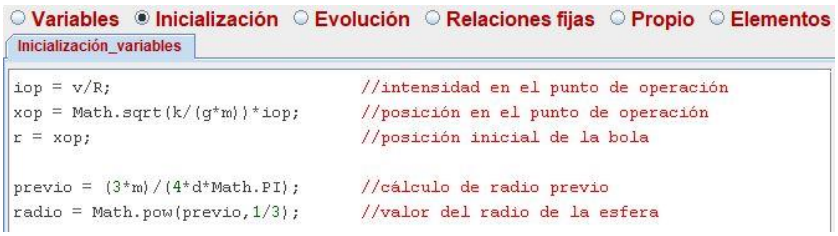


Figura 5.23: Inicialización del modelo (no lineal)

En la *Evolución* se introduce las ecuaciones diferenciales del modelo no lineal y las del controlador. Como el modelo tiene ecuaciones de segundo orden es más sencillo el uso de dos estados para derivar dos veces.

Como el incremento que se ha tomado para la simulación ha sido de un valor de 0.01 se ha tenido que ajusta las imágenes por segundo y los pasos por visualización para que simule en tiempo real, el ajuste que se ha producido es similar al del péndulo invertido.

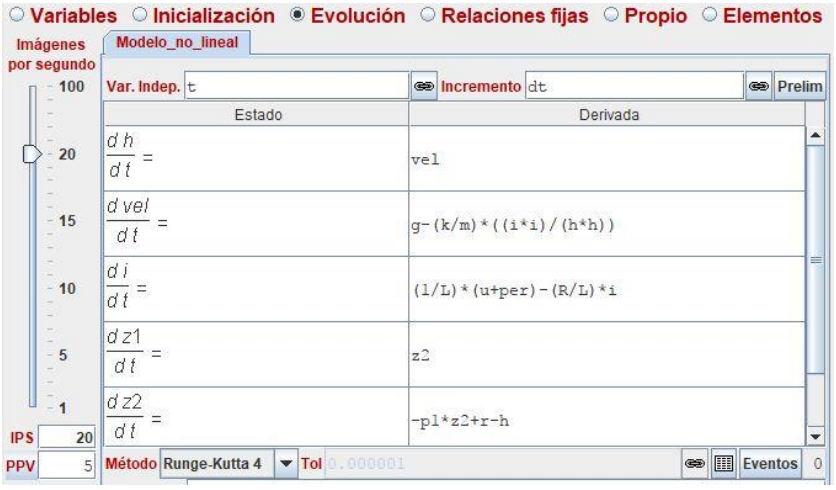


Figura 5.24: Evolución del modelo (no lineal)

En *Relaciones fijas* se implementa la ecuación que rige la salida del controlador que se recalcula con cada paso de simulación, también se utiliza una ventana adicional para realizar los mismos cálculos que en el panel *Inicialización* para que se recalculen durante la simulación si se produjera una variación:

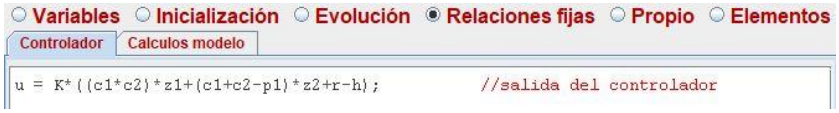


Figura 5.25: Relaciones fijas salida del controlador (no lineal)

En la ventana cálculos del modelo además del uso de las mismas ecuaciones del *Inicialización* se condiciona el modelo limitando su movimiento a través del eje Y dicha limitación seria 0 cuando se encuentra en el suelo y 1 cuando se encuentra en la base del levitador magnético.

Figura 5.26: Relaciones fijas variables del modelo (no lineal)

Por ultimo antes de pasar al diseño gráfico de la simulación se encuentra la ventana, *propio,* donde se crea un método para poder interactuar el usuario y poder modificar la posición de la esfera con el puntero.

A diferencia de los modelos anteriores este solo modifica su posición en el eje Y por lo que no requiere de métodos predefinidos en Java para que pueda realizar el movimiento el usuario solo es necesario reiniciar las variables:

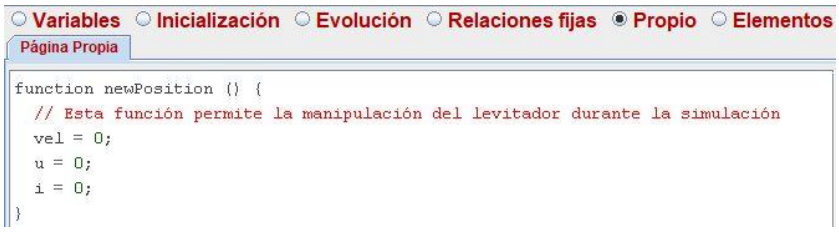


Figura 5.27: Método propio para interactuar con la simulación (no lineal)

Para la finalización de la simulación cabe destacar el último punto donde viene el desarrollo de la interface gráfica en la pestaña *HtmlView*:

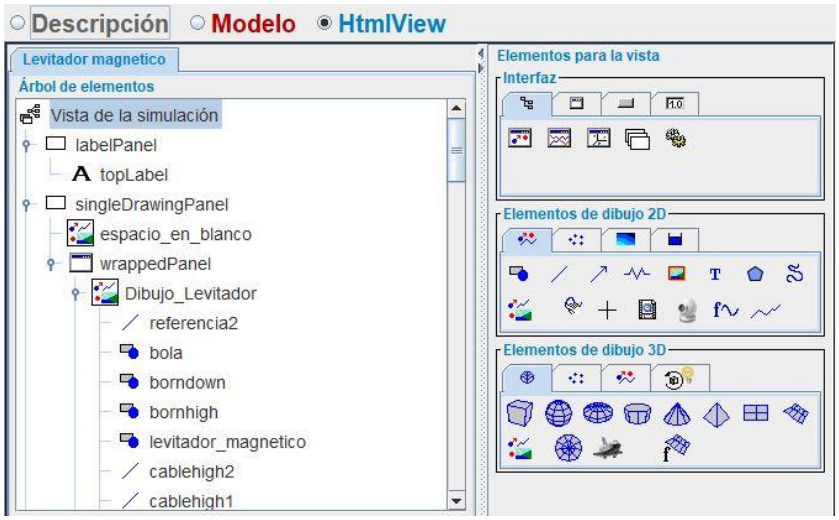


Figura 5.28: Diseño gráfico del modelo no lineal

Finalmente, al ser el lenguaje de programación JavaScript al correr la simulación este abre una página vez, donde se encuentra todo el desarrollo realizado en la ventana *HtmlView.*

Por otro lado, esta página permite interactuar al usuario permitiéndolo mediante acciones de control poder parar, reiniciar o reanudar la simulación, así como aplicarle una perturbación al sistema para ver cómo se comporta el sistema.

También permite al usuario omitir la gráfica del controlador el valor de referencia, la gráfica de control y de velocidad a elección del usuario. Es posible modificar los valores del controlador y del modelo y observar un comportamiento distinto dentro de los límites que el controlador permita si se supera la respuesta del modelo se hará inestable:

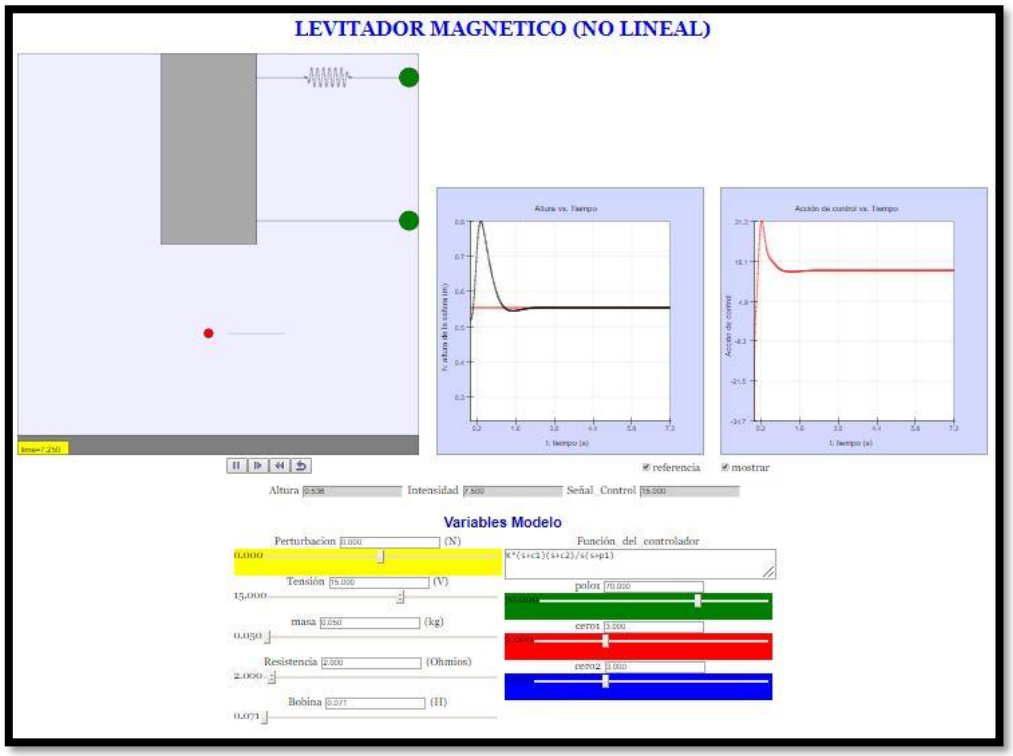


Figura 5.29: Visualización final de la interface del modelo no lineal

6. ANEXOS

6.1. Anexo I. Bloque del péndulo invertido no lineal en Simulink

Al abrir el bloque P_I_no-lineal se observa la siguiente imagen:

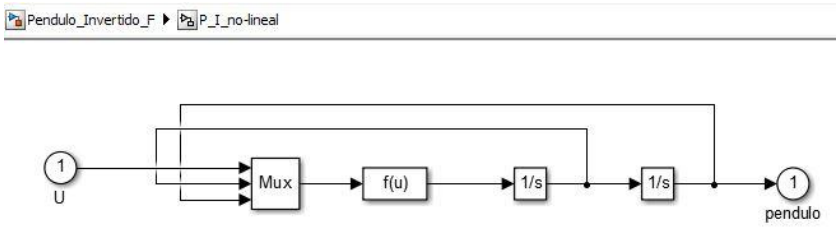


Figura 6.1: Bloque del péndulo invertido no lineal en Simulink

En la (Figura 6.1) representa el diagrama de bloques de la función del péndulo invertido. Dentro del bloque f(u) se encuentra la ecuación de evolución del péndulo invertido, haciendo click para acceder dentro del bloque se puede ver la expresión:

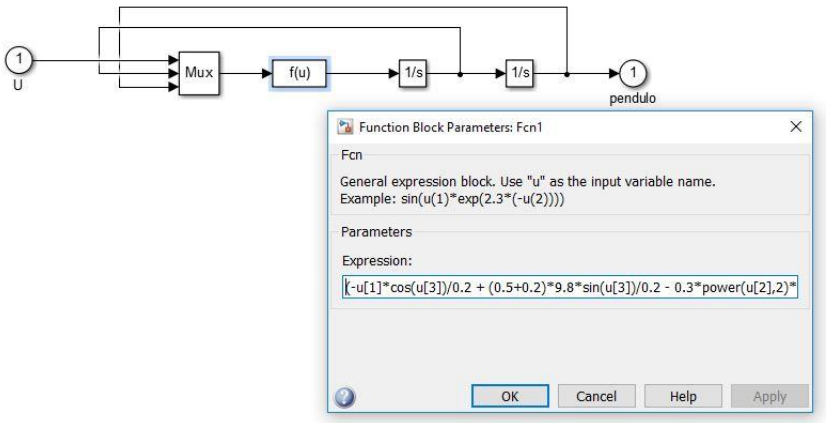


Figura 6.2: Expresión de la ecuación del péndulo invertido

6.2. Anexo II. Transformación SS a función de transferencia

Para trabajar con mayor sencillez y con el propósito de obtener cualquier tipo de función de transferencia mediante el programa desarrollado en Matlab, utilizando el archivo L_M_lineal.m, en este anexo se observa el desarrollo matemático que realiza el programa. Cumpliendo la siguiente ecuación se obtiene el resultado final:

$$G(s) = C \cdot (s \cdot I - A)^{-1} \cdot B \tag{6.1}$$

La matriz del sistema en espacio de estado es la siguiente:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m}\frac{x_3^{*2}}{x_1^{*3}} & 0 & -\frac{2k}{m}\frac{x_3^*}{x_1^{*2}} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u \tag{6.2}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{6.3}$$

Realizando los procedimientos matemáticos por partes resulta más sencillo, en primer lugar resolvemos $(s \cdot I - A)$:

$$(s \cdot I - A) = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m}\frac{x_3^{*2}}{x_1^{*3}} & 0 & -\frac{2k}{m}\frac{x_3^*}{x_1^{*2}} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \tag{6.4}$$

Obteniendo como resultado:

$$(s \cdot I - A) = \begin{bmatrix} s & 1 & 0 \\ -\frac{2k}{m}\frac{x_3^{*2}}{x_1^{*3}} & s & \frac{2k}{m}\frac{x_3^*}{x_1^{*2}} \\ 0 & 0 & s - \frac{R}{L} \end{bmatrix} \tag{6.5}$$

Para una manipulación más sencilla de las ecuaciones se hace la siguiente conversión de algunas ecuaciones de la matriz:

$$\begin{cases} a_{21} = \dfrac{2k}{m}\dfrac{x_3^2}{x_1^3} \\ a_{23} = -\dfrac{2k}{m}\dfrac{x_3}{x_1^2} \end{cases}$$

Llevado a cabo los cambios, se procede a realizar la inversa de la matriz. Como los cálculos al aplicar matriz inversa no es objetivo de este trabajo solo se representa el resultado:

$$(s \cdot I - A)^{-1} = \frac{\begin{bmatrix} s \cdot (s + \frac{R}{L}) & s + \frac{R}{L} & a_{23} \\ -(s + \frac{R}{L}) \cdot a_{21} & s \cdot (s + \frac{R}{L}) & -s \cdot a_{23} \\ 0 & 0 & s^2 - a_{21} \end{bmatrix}}{\det(sI - A)} \tag{6.6}$$

Y multiplicando por la matriz C y B se obtiene el siguiente resultado:

$$C \cdot (s \cdot I - A)^{-1} \cdot B = \frac{[1 \quad 0 \quad 0] \cdot (s \cdot I - A)^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}}{\det(sI - A)} \tag{6.7}$$

Dando como resultado la siguiente función de transferencia:

$$G(s) = \frac{\frac{a_{23}}{L}}{\mathrm{s}^3 + \frac{\mathrm{R}}{\mathrm{L}} \cdot s^2 - a_{21} \cdot \mathrm{s} - a_{21} \cdot (\frac{\mathrm{R}}{\mathrm{L}})} \tag{6.8}$$

Bibliografía y referencias

[1] Carl Dorf, Richard & Hamilton Bishop, Robert (2005). *Sistemas de control moderno*. Madrid, España: Editorial Pearson education.

[2] Esquembre, Francisco. *Creación de Simulaciones Interactivas en Java.* Madrid: Prentice-Hall, 2004.

[3] Ríos Ruiz, Juan David (2010). *Diseño y construcción de un sistema de levitación magnética controlador por un algoritmo PID*. Recuperado el 24 de mayo del 2017 de https://repository.eafit.edu.co/bitstream/handle/10784/2799/RiosRuiz_JuanDavid_2010.pdf.

[4] Antón Montero, Sara (2010). *Implementación de controladores PID con Easy Java Simulations para aplicaciones web docentes.* Recuperado 2 de mayo 2017 de https://e-archivo.uc3m.es/handle/10016/10682.

[5] Control Tutorials for Matlab & Simulink (2017). *Inverted Pendulum*. Disponible http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling. [Último acceso 25 de agosto de 2017].

[6] Garrido Bullon, Santiago. *Ilustración de un péndulo invertido*. Recuperado de https://www.researchgate.net/figure/263523094_fig1_Figure-1-Detalle-del-sistema-de-pendulo-invertido-La-se-nal-de-control-es-la-fuerza.

[7] Juan Carlos Milena Moreno. (2010). *Control lineal y no lineal de un Levitador magnético*. Barcelona, España. Recuperado de https://upcommons.upc.edu/bitstrea/handle/2099.1/9923Control%20lineal%/20y%20no%20lineal%20de%20un%20levitador%20magnetico.pdf.

[8] Benjamín C. Kuo (1996). *Sistemas de control automático.* Barcelona, España. México: Prentice Hall Hipanoamericana S.A.

[9] Sara Antón Moreno (2010). Implementación de controladores PID con Easy Java Simulations para aplicaciones web docentes. Madrid. Recuperado https://e-archivo.uc3m.es/handle/10016/10682.