

Universidad de Jaén

Utilización de Easy Java Simulations para laboratorio virtual docente

CONTROL DEL SISTEMA BARRA Y BOLA

ALICIA MONTORO LENDÍNEZ
24/06/2019

Tabla de contenido

- 1. OBJETIVO 3
- 2. MODELADO DEL SISTEMA 3
 - 2.1. Modelado del sistema barra y bola..... 3
 - 2.2. Modelado del motor 7
 - 2.2.1. Ensayo del servomotor..... 7
 - 2.3. Detalles de extremos y límites 14
 - 2.4. Relaciones Fijas 15
- 3. CONTROL DEL SISTEMA..... 17

1. OBJETIVO

El objeto del trabajo es realizar un laboratorio virtual docente para practicar el control proporcional, derivativo e integral en un sistema típico de una barra y una bola que se ha modelado.

2. MODELADO DEL SISTEMA

2.1. Modelado del sistema barra y bola

Para comenzar se tiene que obtener las ecuaciones que rigen el movimiento de la bola sobre la barra sin tener en cuenta el motor, es un primer sistema muy simplificado.

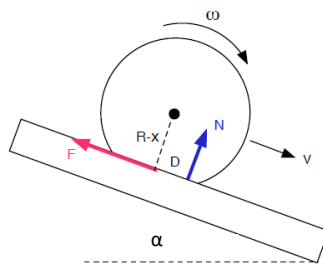


Ilustración 1: Bola sobre el plano inclinado

Una bola rodando por un plano inclinado un ángulo α , tendrá una fuerza normal N , una fuerza de fricción F , y una fuerza producida por el peso de la bola P que será descompuesto en dos fuerzas dependiendo del coseno y el seno del ángulo α .

$$F = ma$$

$$N = mg \cos \alpha$$

$$Fr = \mu N$$

Sustituyendo en

$$Fr - mg \sin \alpha = ma$$

Se obtiene

$$\mu mg \cos \alpha - mg \sin \alpha = ma$$

Simplificando

$$a = g(\mu \cos \alpha - \sin \alpha)$$

Siendo,

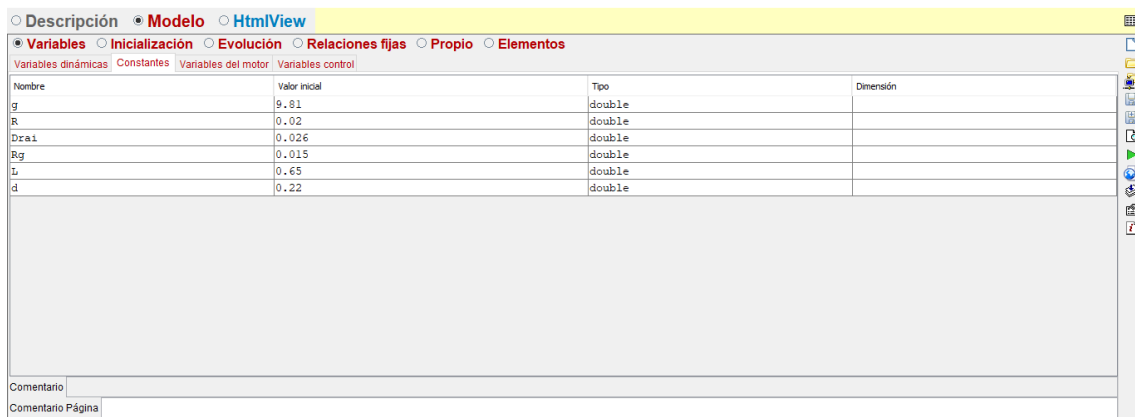
μ el coeficiente de rozamiento,

a la aceleración,

g la gravedad,

α el ángulo de inclinación de la barra.

Al utilizar Easy Java Simulations se cambia μ por el símbolo cr y, además, se añade un coeficiente $cract$ para activar y desactivar la fricción de manera que se pueda realizar dos simulaciones en una, teniendo en cuenta la fuerza de fricción de la bola sobre la barra y otra sin tenerla en cuenta.



Nombre	Valor inicial	Tipo	Dimensión
g	9.81	double	
R	0.02	double	
Dra1	0.026	double	
Rg	0.015	double	
L	0.65	double	
d	0.22	double	

Comentario

Comentario Página

Ilustración 2: Tabla de constantes

<input type="radio"/> Descripción <input checked="" type="radio"/> Modelo <input type="radio"/> HtmlView		
<input checked="" type="radio"/> Variables <input type="radio"/> Inicialización <input type="radio"/> Evolución <input type="radio"/> Relaciones fijas <input type="radio"/> Propio <input type="radio"/> Elementos		
Variables dinámicas <input checked="" type="radio"/> Constantes <input type="radio"/> Variables del motor		
Nombre	Valor inicial	Tipo
xref	0.000000000	double
x	0.17	double
y	0.0	double
vx	0.0	double
Croz	true	boolean
cr	0.005	double
cract	0	double
Alfa	0	double
t	0.0	double
dt	0.05	double
Fr	0.0	double

Ilustración 3: Tabla de variables dinámicas del movimiento de la bola

Para la evolución en Easy Java Simulations se tiene en cuenta que la derivada del espacio es la velocidad y la derivada de ésta última es la aceleración.

<input type="radio"/> Descripción <input checked="" type="radio"/> Modelo <input type="radio"/> HtmlView		
<input type="radio"/> Variables <input type="radio"/> Inicialización <input checked="" type="radio"/> Evolución <input type="radio"/> Relaciones fijas <input type="radio"/> Propio <input type="radio"/> Elementos		
Imágenes por segundo: <input type="text" value="20"/> EvoluciónX Topes: EvoluciónMotor Limite Control		
Var. Indep.	t	Incremento dt
Estado	Derivada	
$\frac{dx}{dt} =$	vx	
$\frac{dvx}{dt} =$	$g * (-\text{Math.sin(Alfa)} + \text{cract} * \text{cr} * \text{Math.cos(Alfa)})$	
<div> <div>IPS: 20</div> <div>ppv: 1</div> <div>Método: Runge-Kutta 4</div> <div>Tol: 0.00001</div> <div>Eventos: 0</div> </div>		

Ilustración 4: Ecuaciones de la evolución de la bola

Ahora se tiene que realizar la interfaz para el usuario, para ello se va añadiendo paneles de dibujos, objetos y gráficas y todo lo que sea útil en el apartado de HTML View.

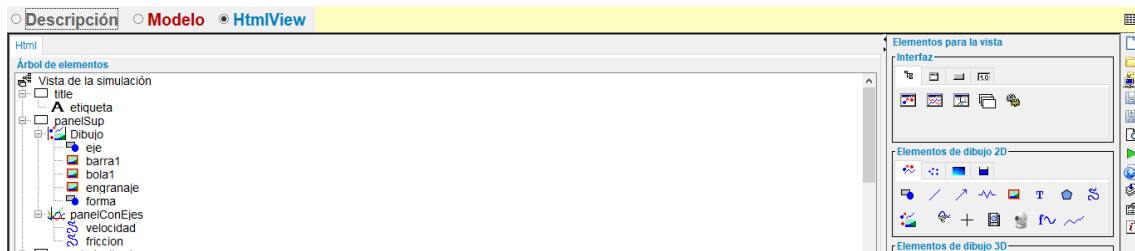


Ilustración 5: Panel Html

En la sección de la bola se tiene que añadir en *Rotate* la ecuación que rige la rotación de la bola para que así el efecto visual del movimiento en la simulación sea más real.

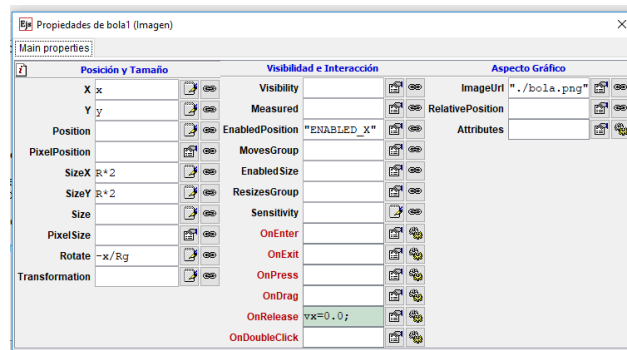


Ilustración 6: Propiedades de la bola

2.2. Modelado del motor

El motor que se utiliza para la maqueta física sobre la que se basa el laboratorio virtual es un servomotor Futaba S3003.



Ilustración 7: Servomotor

Modulation	Analog
Torque	4.8V: 44.00 oz-in (3.17 kg-cm) 6.0V: 57.00 oz-in (4.10 kg-cm)
Speed	4.8V: 0.23 sec/60° 6.0V: 0.19 sec/60°
current	7.2 mA – 8 mA
Weight	1.31 oz (37.0 g)

Tabla 1: Características del servomotor

2.2.1. Ensayo del servomotor

Para averiguar cuál es el orden del sistema del servomotor se realiza un pequeño ensayo para obtener la respuesta del servomotor ante una entrada de escalón.

Para ello se crean unas pequeñas piezas en Autodesk Inventor que después se fabrican en una impresora 3D.

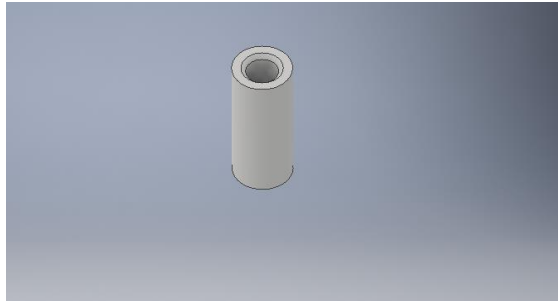


Ilustración 8: Modelo 3D de la pieza de acople

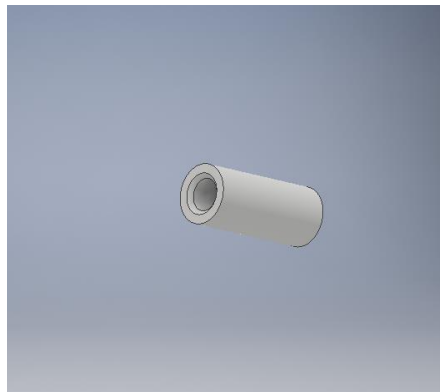


Ilustración 9: Modelado 3D de la pieza de acople

Para realizar el ensayo también se utiliza una placa Arduino Uno para crear un código donde se obtienen los valores de la respuesta ante la entrada de escalón.

```
#include <Servo.h>

// Declaramos la variable para controlar el servo
Servo servoMotor;
int i=0;
int valorADC=0;
int valorADC2=0;
void setup()
{
  Serial.begin(115200); //Inicialización y configuración del protocolo serial (Velocidad= 115200 baudios)
  // Iniciamos el servo para que empiece a trabajar con el pin 10
  servoMotor.attach(10);
  // Inicializamos al ángulo 0 el servomotor
  servoMotor.write(0);
  Serial.println("Start ciclo");
  delay(3000);
  servoMotor.write(180);

}

void loop()
{
  Serial.println(analogRead(A0)); //Se toma el dato del canal analogo A0
  delay(10); // Retardo
  while(micros()>100000000);
}
```

Ilustración 10: Código para el escalón hasta 180°

```

#include <Servo.h>

// Declaramos la variable para controlar el servo
Servo servoMotor;
int i=0;
int valorADC=0;
int valorADC2=0;
void setup()
{
  Serial.begin(115200); //Inicialización y configuración del protocolo serial (Velocidad= 115200 baudios)
  // Iniciamos el servo para que empiece a trabajar con el pin 10
  servoMotor.attach(10);
  // Inicializamos al ángulo 0 el servomotor
  servoMotor.write(0);
  Serial.println("Start ciclo");
  delay(3000);
  servoMotor.write(90);

}

void loop()
{
  Serial.println(analogRead(A0)); //Se toma el dato del canal analogo A0
  delay(10); // Retardo
  while(micros()>100000000);
}

```

Ilustración 11: Código para el escalón hasta 90°

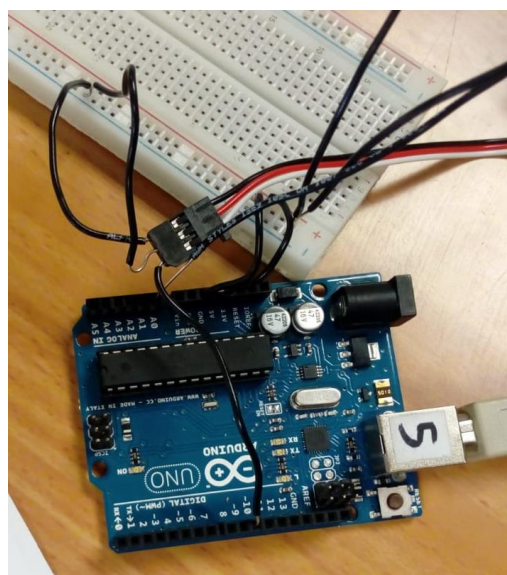


Ilustración 12: Arduino UNO



Ilustración 13: Potenciómetro acoplado al servomotor

Los datos obtenidos se grafican en una hoja Excel, tanto para el ensayo de 180° como para el ensayo de 90° se realizan varias pruebas para tomar unos valores correctos de las constantes que caracterizan al sistema, en este caso, al tratarse de un sistema de primer orden se obtendrá K y τ .

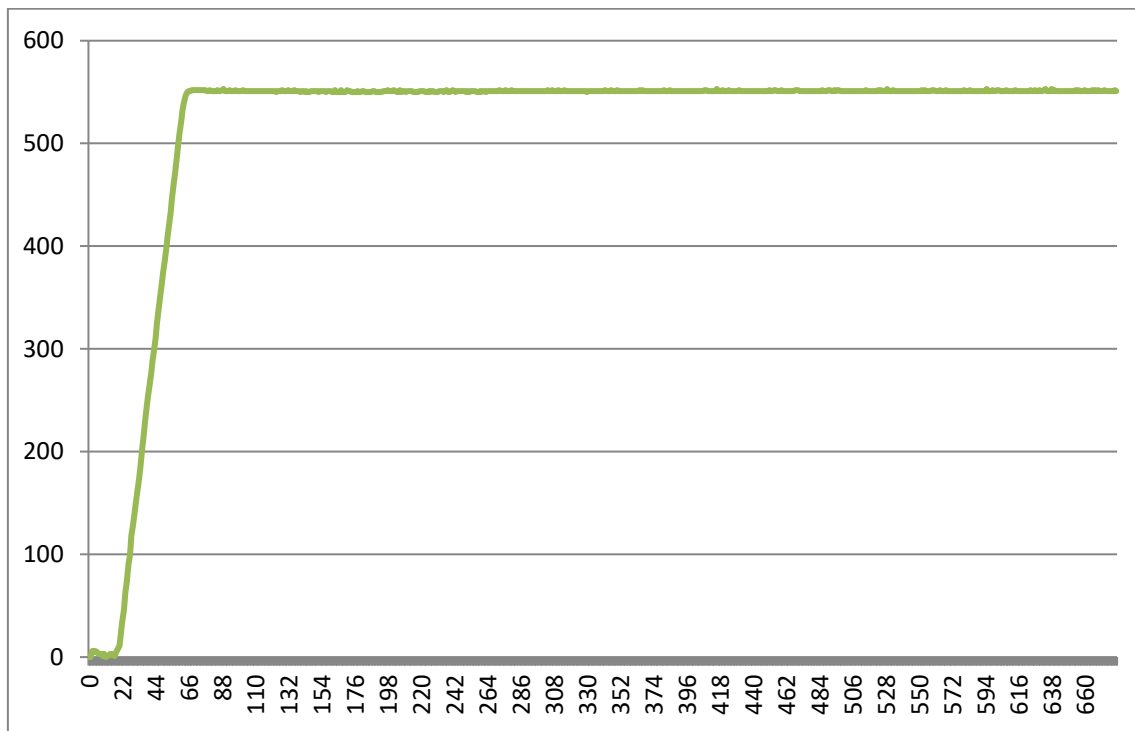


Tabla 2: Gráfica del ensayo a 180°

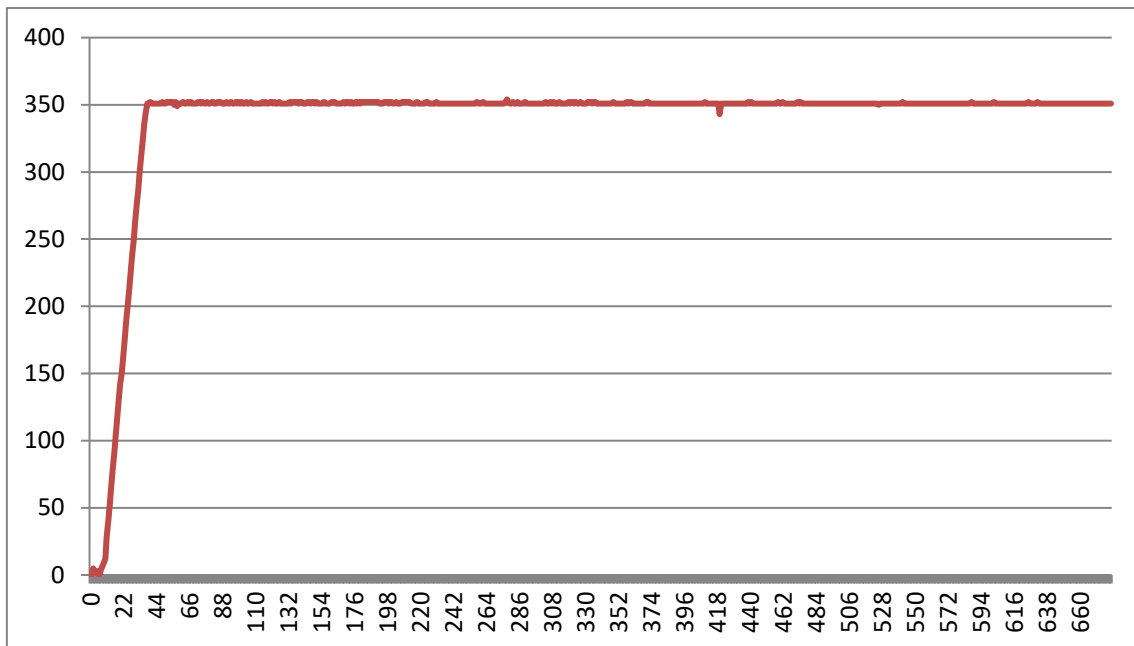


Tabla 3: Gráfica del ensayo de 90°

Por lo tanto la ecuación del servomotor es

$$\omega = \frac{K * u - \Theta}{\tau}$$

Siendo,

ω velocidad angular del motor,

K ganancia en estado estable,

u tensión de entrada,

τ constante de tiempo,

Θ ángulo de rotación del servomotor.

Para la evolución en Easy Java Simulations, ω es la derivada del ángulo de rotación del servomotor y τ y K , respectivamente, se nombraran como T_m y K_m .

☐ Descripción
 ☒ **Modelo**
☐ HtmlView

☒ **Variables**
☐ Inicialización
 ☐ Evolución
 ☐ Relaciones fijas
 ☐ Propio
 ☐ Elementos

Variables dinámicas
 ☐ Constantes
 ☐ Variables del motor
 ☒ Variables control

Nombre	Valor inicial	Tipo	Dimensión
\dot{m}	30	double	
Km	561	double	
theta	0.0000000	double	
u	0.0	double	
thetaref	0.0	double	

Comentario

Comentario Página

Ilustración 14: Tabla de variables del moto

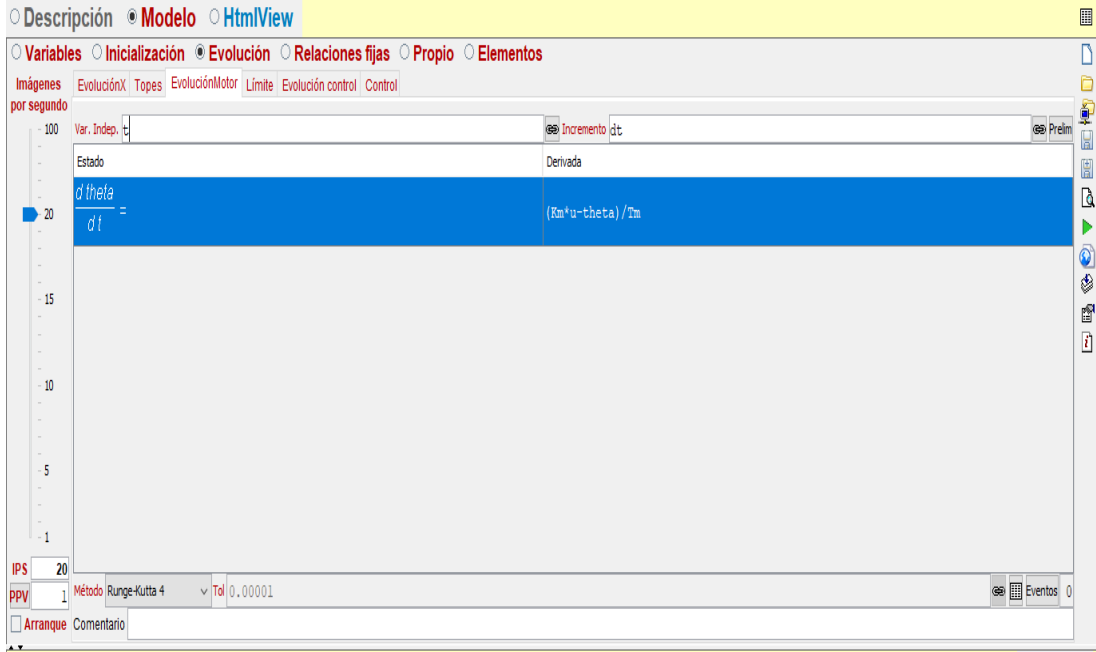


Ilustración 15: Ecuación de la evolución del motor

Se añade al panel de HTMLView los objetos, gráficas y variables que sean necesarias. Para el objeto que representa al motor dentro del apartado de *Transformation* se relaciona que sea de acuerdo a Theta.

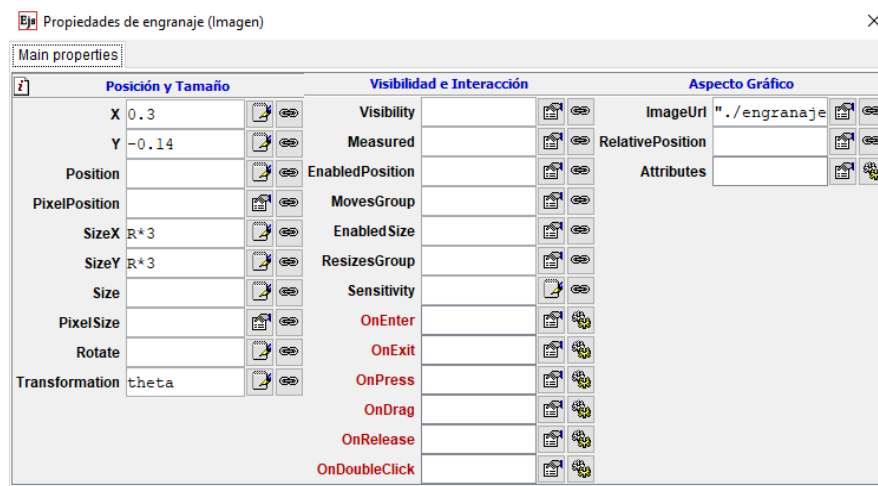


Ilustración 16: Propiedades del servomotor

2.3. Detalles de extremos y límites

Para el correcto funcionamiento de la simulación se debe de definir unos extremos de los cuales la bola vaya desde un extremo hacia otro y para más realidad de la simulación se define un rebote de ésta sobre ellos.



Ilustración 17: Código de los topes

Además se tiene en cuenta que la barra puede llegar hasta cierto ángulo de inclinación en la realidad aunque en la simulación pueda coger cualquier ángulo α se restringe el α en función del ángulo Θ del servomotor que de nuevo aunque en la simulación puede dar una vuelta completa en la realidad solo tiene un período de 0° a 180° . También se observa que a mayor α tendremos un mayor Θ por lo que no se necesita añadir ningún signo.

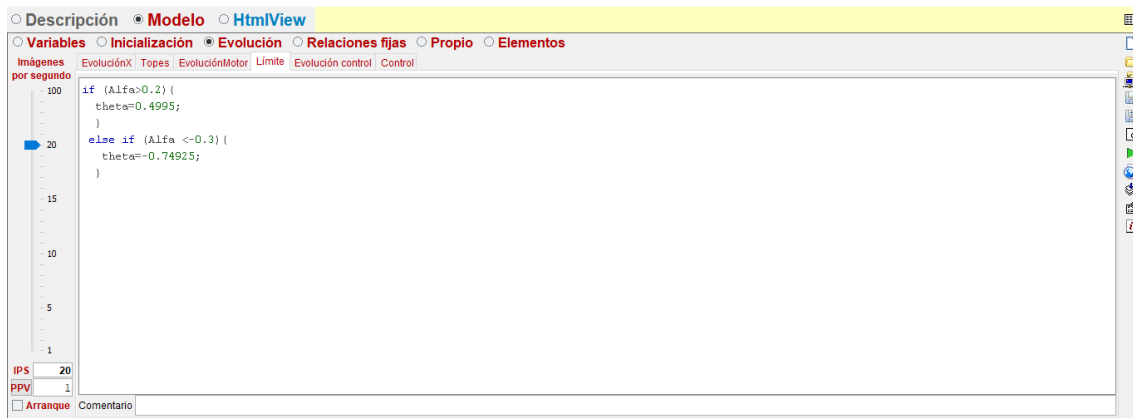


Ilustración 18: Código del límite entre ángulos

2.4. Relaciones Fijas

En este apartado se añade la relación que existe entre el ángulo de la barra α , y el ángulo del servomotor Θ . Se pone una relación para poder activar la fuerza de rozamiento F_r , que depende de si la bola se mueve desde el extremo derecho hacia el izquierdo o viceversa, si el movimiento es de derecha a izquierda la velocidad de la bola será negativa por lo que necesita que en F_r sea contraria y se añade un signo menos, en cambio, si el movimiento es de izquierda a derecha la velocidad de la bola es positiva por lo que no necesita ningún cambio de signo en F_r . La fuerza de rozamiento F_r , también se define en este apartado y se relaciona el movimiento de la bola en el eje X tiene con el eje Y.

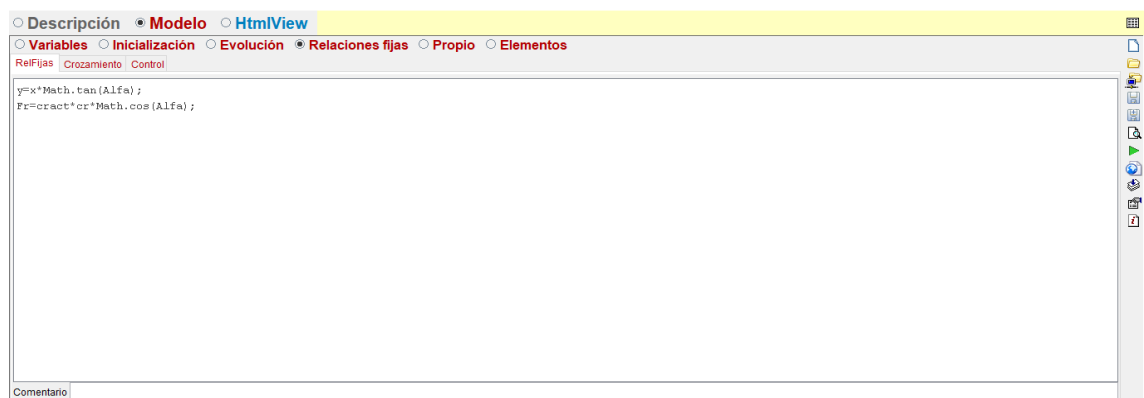


Ilustración 19: Definición de la F_r y relación eje X con eje Y

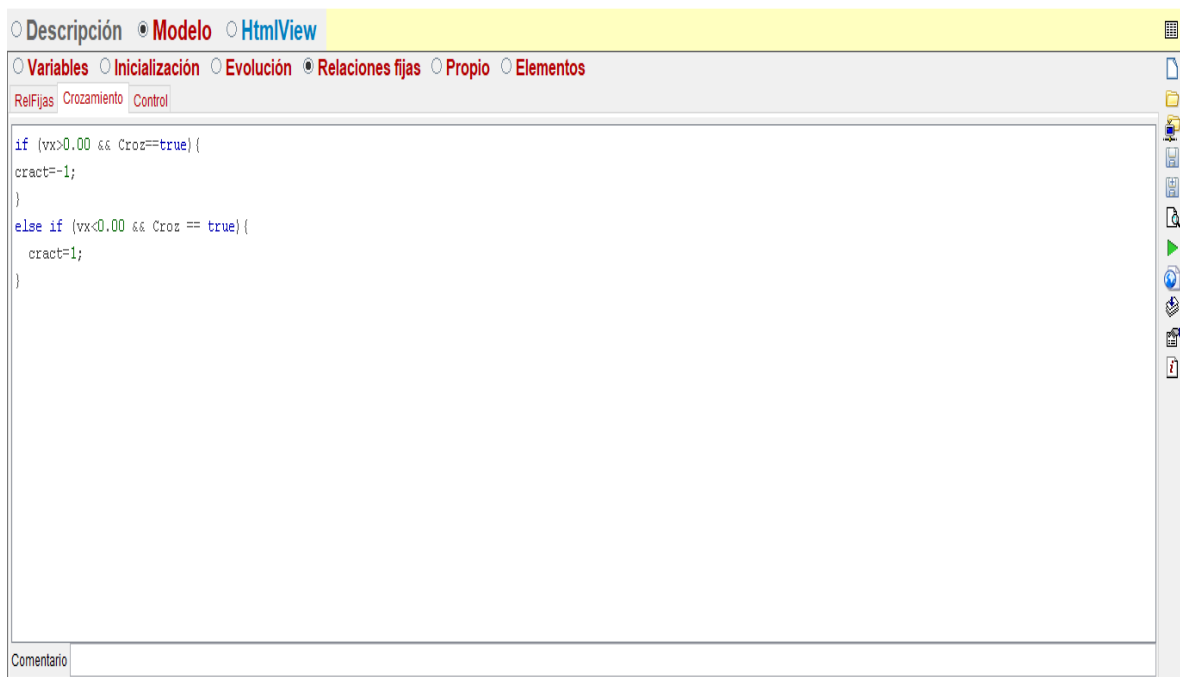


Ilustración 20: Activación del signo de F_r según la velocidad de la bola

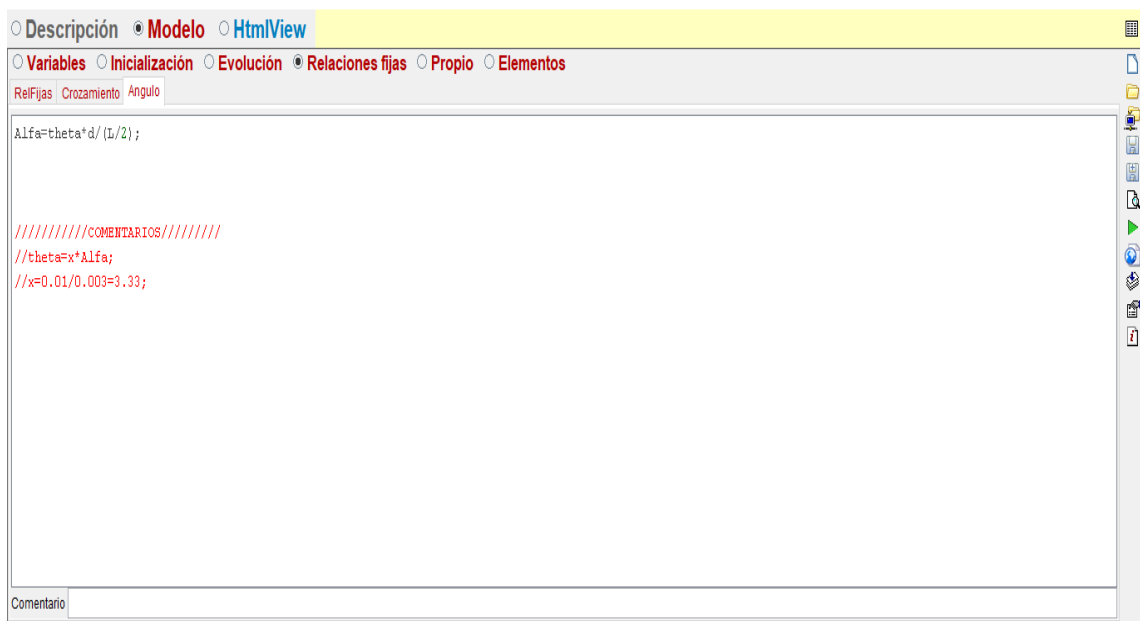
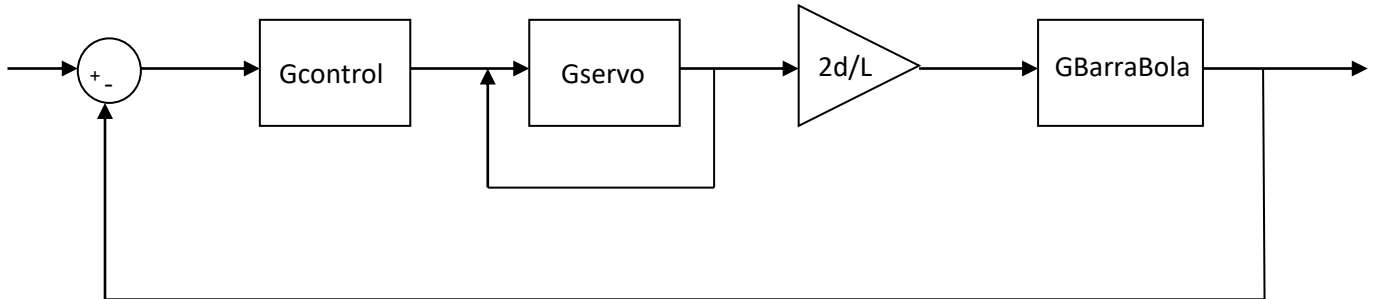


Ilustración 21: Relación de los ángulos Θ y α

3. CONTROL DEL SISTEMA

Para realizar el control del sistema primero se necesita definir cada bloque del lazo de control.



Se define el bloque del servomotor como G_{servo}

$$G_{servo} = \frac{\theta(s)}{u(s)} = \frac{Km}{\tau s + 1}$$

Utilizando los datos que se obtuvieron en el ensayo del motor

$$G_{servo} = \frac{\theta(s)}{u(s)} = \frac{551}{41s + 1}$$

Operando para ver claramente donde se encuentra el polo

$$G_{servo} = \frac{\frac{551}{41}}{s + \frac{1}{41}} = \frac{13.44}{s + 0.0244}$$

Haciendo la función de bucle cerrado

$$G_{servobc} = \frac{13.44}{s + 13.4644}$$

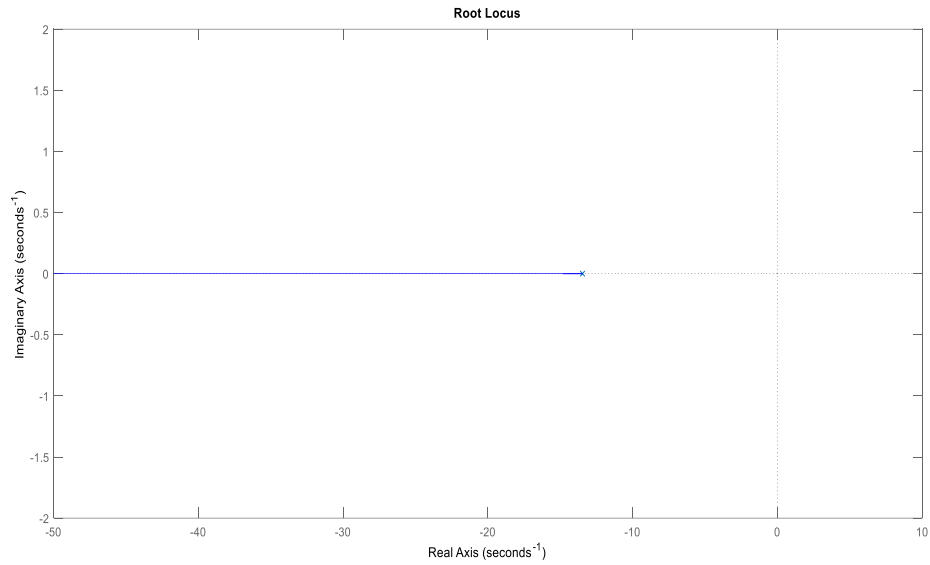


Ilustración 22: Lugar de las raíces del bucle cerrado de Gservo

Haciendo el lazo abierto de todo el sistema se obtiene

$$Gp = Gservobc * \frac{2d}{L} * GBarraBola$$

$$Gp = \frac{13.44}{s + 13.4644} * \frac{2d}{L} * \frac{g(cract * cr + 1)}{s^2}$$

Para hacer el control se tendrá que la fuerza de rozamiento es cero por esta condición se obtiene

$$Gp = \frac{13.44}{s + 13.4644} * \frac{2d}{L} * \frac{g}{s^2}$$

$$Gp = \frac{13.44}{s + 13.4644} * \frac{2 * 0.22}{0.65} * \frac{9.81}{s^2}$$

Agrupando los términos

$$Gp = \frac{89.25}{s^2(s + 13.4644)}$$

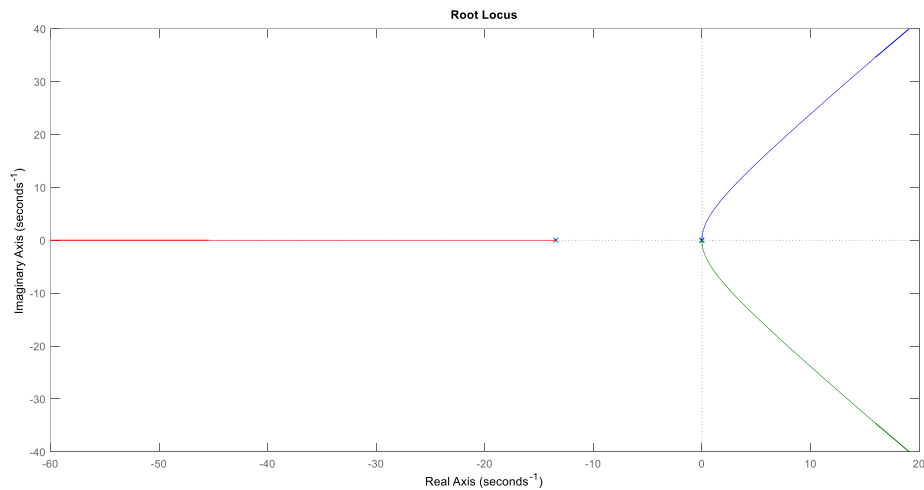


Ilustración 23: Lugar de las raíces de Gp

Se utiliza un control PD para intentar añadir un cero de esa manera las dos ramas de la derecha que hacen que el sistema sea inestable se vuelva estable.

$$Gba = Gp * Gc$$

$$Gba = \frac{89.25}{s^2(s + 13.4464)} * Kp * Td * (s + \frac{1}{Td})$$

El valor idóneo para Td es 0.1

$$Gba = \frac{89.25}{s^2(s + 13.4464)} * Kp * 0.1 * (s + 10)$$

$$Gba = \frac{8.925 * Kp * (s + 10)}{s^2(s + 13.4464)}$$

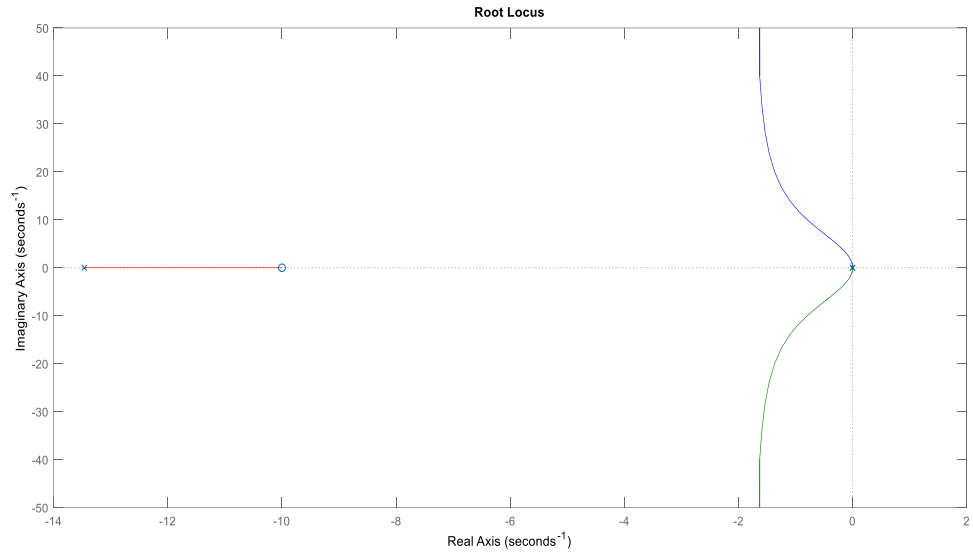


Ilustración 24: Lugar de las raíces para Gba

Para obtener Kp se realiza la función de bucle cerrado

$$Gbc = \frac{8.925 * Kp * (s + 10)}{8.925 * Kp * (s + 10) + s^2(s + 13.4464)}$$

El polinomio característico es

$$8.925 * Kp * (s + 10) + s^2(s + 13.4464) = s^3 + 13.4464s^2 + 8.925Kps + 89.25Kp$$

Utilizando la herramienta de Matlab se observa que un polo conjugado correcto sería

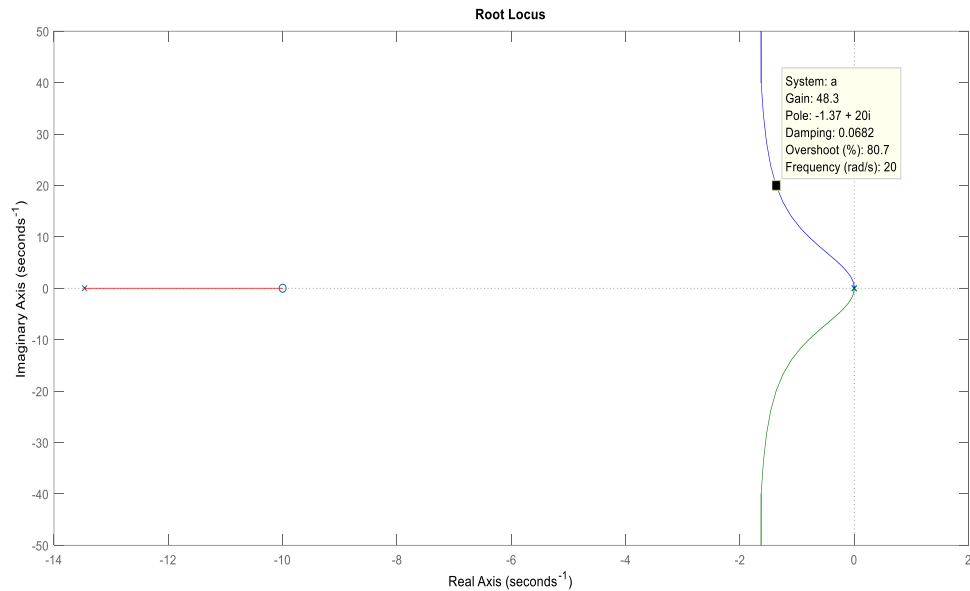


Ilustración 25: Polo conjugado del LR de Gba

$$s = -1.37 \pm 20i$$

$$(s^2 + 2.74s + 401.88)(s + a) = s^3 + 13.4464s^2 + 8.925Kps + 89.25Kp$$

De aquí se obtiene un sistema de ecuaciones del cual se resuelve la incógnita a y la Kp

$$2.74 + a = 13.4464 \quad (1)$$

$$401.88 + 2.74a = 8.925Kp \quad (2)$$

$$401.88a = 89.25Kp \quad (3)$$

De la ecuación 1 se obtiene

$$a = 13.4464 - 2.74 = 10.7064$$

Sustituyendo a en las ecuaciones 2 y 3 se obtiene los valores de Kp

$$Kp = \frac{401.88 + 2.74 * 10.7064}{13.4464} = 32.07$$

$$Kp = 401.88 * \frac{10.7064}{89.25} = 48.21$$