



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# **DISEÑO Y DESARROLLO DE UN SISTEMA BARRA-BOLA**

**Alumno: Montoro Lendínez, Alicia**

Tutor: Prof. D. Elisabet Estevez Estevez

Cotutor: Prof. D. Ildefonso Ruano Ruano

Dpto: Electrónica y Automática Industrial

**Junio,2020**





Universidad de Jaén  
Escuela Politécnica Superior de Jaén

Doña ELISABET ESTEVEZ ESTEVEZ y Don ILDEFONSO RUANO RUANO, tutores del Proyecto Fin de Carrera titulado: DISEÑO Y DESARROLLO DE UN SISTEMA BARRA-BOLA, que presenta ALICIA MONTORO LENDÍNEZ, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2020

El alumno:

Los tutores:

ALICIA MONTORO LENDÍNEZ

ELISABET ESTEVEZ ESTEVEZ

ILDEFONSO RUANO RUANO

## RESUMEN

En el presente trabajo, se realiza un estudio sobre el control y diseño de un laboratorio virtual del sistema barra-bola mediante la herramienta Easy Java Simulations para su utilización en asignaturas. Previamente se realiza un modelado del sistema. Este sistema ha sido modificado de tal manera que la bola no se encuentra sobre la barra sino que se encuentra dentro del seno de un fluido contenido en un tubo cilíndrico. Esta modificación se ha realizado debido a que el sistema tenía una dinámica muy rápida e imposibilitaba su implementación como laboratorio remoto, por ello, para que el movimiento de la bola fuese más lento se introdujo en un fluido. El trabajo se divide principalmente en tres partes:

- Modelado del sistema barra-bola.
- Diseño del control del sistema barra-bola. Realizando pruebas para concluir que estrategia de control es mejor para el sistema.
- Implementación del control y diseño de simulaciones en varios laboratorios virtuales donde se lleva a cabo la práctica de todo.

## ABSTRACT

In this work, an analysis about the control and design of the system bar and ball virtual laboratory is done by the Easy Java Simulations tool in order to use in subjects. First, a model system is done. This system have modified with a view to the ball is not on the bar. However, the ball is inside a fluid which is contained in a cylindrical tube. This modification is done due to the system had quick dynamics and the implementation was impossible at remote laboratory, therefore, the movement of the ball is slower, the ball is inside a fluid. This works has three main parts:

- System bar and ball model
- System bar and ball design. Making tests to conclude which control strategy is the best for the system.
- Control implementation and design of simulations in several virtual laboratories where every practice is done

**ÍNDICE DE FIGURAS**

Figura 3-1. Esquema de perfil del sistema barra-bola.....	3
Figura 3-2. Esquema de perfil de la barra y el motor .....	8
Figura 4-1. Dinámica del sistema .....	9
Figura 4-2. Bloque del motor .....	11
Figura 4-3. Ensayo del motor .....	11
Figura 4-4. Gráfica del ensayo del motor.....	12
Figura 4-5. Gráfica 2 del ensayo del motor.....	12
Figura 4-6. Gráfica 3 del ensayo del motor.....	13
Figura 4-7. Dinámica del sistema con las funciones de transferencia.....	14
Figura 4-8. Dinámica del sistema simplificado .....	15
Figura 4-9. Control en cascada en lazo cerrado. ....	19
Figura 4-10. Lazo de control esclavo.....	19
Figura 4-11. Lazo de control maestro .....	20
Figura 4-12. Lazo cerrado de control del sistema simplificado.....	22
Figura 5-1. Variables del sistema .....	22
Figura 5-2. Diagrama de bloque del motor .....	23
Figura 5-3. Gráfica del motor ante un escalón.....	23
Figura 5-4. Lazo cerrado del control del actuador.....	24
Figura 5-5. Controlador esclavo PD diseñado en PID tuner.....	24
Figura 5-6. Respuesta del lazo cerrado del motor ante una entrada escalón. ....	25
Figura 5-7. Diagrama de bloques del control del motor .....	25
Figura 5-8. Respuesta del lazo cerrado del motor ante una entrada escalón sin PID tuner. .....	26
Figura 5-9. Diagrama de bloques del sistema barra-bola .....	26
Figura 5-10. Diagrama de bloque de todo el control .....	27
Figura 5-11. Controlador maestro PD diseñado en PID tuner.....	27
Figura 5-12. Respuesta del lazo cerrado ante una entrada escalón. ....	28
Figura 5-13. Función de transferencia .....	28
Figura 5-14. Lugar de las raíces.....	29
Figura 5-15. Diseño controlador lazo cerrado .....	30
Figura 5-16. Posición estable de la bola .....	30
Figura 5-17. Zoom posición estable de la bola .....	31
Figura 6-1. Variables RTE .....	33
Figura 6-2. Inicialización de getVersion, getLearnerId y getLearnerName .....	34
Figura 6-3. Vista HTML del panel superior .....	34
Figura 6-4. Ventana emergente apartado 1 .....	35
Figura 6-5. Datos de seguimiento de la puntuación .....	35
Figura 6-6. NavRequest y terminate .....	36
Figura 6-7. Utilización setLearnerPreferenceLanguage .....	36
Figura 6-8. Inicialización getLearnerPreferenceLanguage.....	37
Figura 6-9. Datos seguimiento idioma .....	37
Figura 6-10. setCompletionStatus y setSucessStatus .....	37
Figura 6-11. Dato seguimiento del tiempo .....	38
Figura 6-12. Inicialización de los setInteractions.....	38
Figura 6-13. Ejecución de la ventana emergente apartado 2.....	39
Figura 6-14. Ventana emergente apartado 2 .....	39

Figura 6-15. Datos seguimientos de las interacciones.....	39
Figura 6-16. Empaquetar en SCORM.....	40
Figura 6-17. Versión SCORM.....	40
Figura 6-18. Añadir módulo de aprendizaje SCORM.....	41
Figura 6-19. Opciones configuración SCORM.....	41
Figura 6-20. Variables significativas .....	43
Figura 6-21. Variables del control.....	43
Figura 6-22. Variables de puntuación .....	44
Figura 6-23. Variables para el cambio de idioma.....	44
Figura 6-24. Evolución del motor.....	45
Figura 6-25. Evolución del control práctica 1 .....	45
Figura 6-26. Relaciones fijas de la simulación1. Inglés.....	46
Figura 6-27. Relaciones fijas simulación 1. Español.....	46
Figura 6-28. Relaciones fijas simulación 1. N.....	46
Figura 6-29. HTML View simulación 1 .....	47
Figura 6-30. Interfaz del apartado 1 de la simulación 1 .....	48
Figura 6-31. Lazo de control apartado 1 .....	49
Figura 6-32. Lugar de las raíces $G_{ba}$ (práctica 1, apartado 1) .....	49
Figura 6-33. Lazo de control apartado 2.....	51
Figura 6-34. Lugar de las raíces $G_{ba}$ (práctica 1, apartado 2) .....	52
Figura 6-35. Variables significativas .....	55
Figura 6-36. Variables del control.....	55
Figura 6-37. Variables de puntuación .....	56
Figura 6-38. Variables para el cambio de idioma.....	56
Figura 6-39. Evolución del motor.....	57
Figura 6-40. Evolución del control práctica 2 .....	57
Figura 6-41. Relaciones fijas de la simulación2. Inglés.....	58
Figura 6-42. Relaciones fijas simulación 2. Español.....	58
Figura 6-43. Relaciones fijas simulación 2. N.....	58
Figura 6-44. HTML View simulación 2 .....	59
Figura 6-45. Interfaz del apartado 2 de la simulación 2 .....	60
Figura 6-46. Lazo de control apartado 1 .....	61
Figura 6-47. Lugar de las raíces $G_{ba}$ (práctica 2, apartado 1) .....	62
Figura 6-48. Lazo de control apartado 2.....	63
Figura 6-49. Lugar de las raíces $G_{ba}$ (práctica 2, apartado 2).....	64
Figura 6-50. Variables significativas .....	67
Figura 6-51. Constantes simulación 3 .....	68
Figura 6-52. Variables del motor .....	68
Figura 6-53. Variables del control.....	68
Figura 6-54. Variables de puntuación .....	69
Figura 6-55. Variables para el cambio de idioma.....	69
Figura 6-56. Evolución del motor.....	70
Figura 6-57. Evolución de la posición de la bola con linealizado .....	71
Figura 6-58. Evolución de la posición de la bola sin linealizado.....	71
Figura 6-59. Evolución del control en cascada .....	72
Figura 6-60. Relaciones fijas de la simulación3. Inglés.....	72
Figura 6-61. Relaciones fijas simulación 3. Español.....	72
Figura 6-62. Relaciones fijas simulación 3. N.....	73

Figura 6-63. HTML View simulación 3 .....	73
Figura 6-64. Vista previa de la simulación 3 .....	74
Figura 6-65. Lugar de las raíces de $G_{ba1}$ .....	75
Figura 6-66. Lugar de las raíces $G_{ba2}$ .....	77
Figura 10-1. Descripción. Introducción .....	84
Figura 10-2. Descripción. Actividades .....	85
Figura 10-3. Variables dinámicas .....	85
Figura 10-4. Evolución. Ecuaciones .....	86
Figura 10-5. Relaciones Fijas. Fórmulas energías.....	86
Figura 10-6. HTML View.....	87
Figura 10-7. Vista previa MassAndSpring.....	87

**ÍNDICE DE CONTENIDOS**

1.	Introducción.....	1
2.	Objetivos .....	2
3.	Modelado matemático de la barra y bola.....	3
4.	Dinámica del sistema. ....	9
4.1.	Dinámica del actuador: motor.....	9
4.2.	Dinámica de la planta: barra y bola. ....	14
4.3.	Dinámica del sistema barra y bola.....	14
4.4.	Dinámica del sensor.....	15
4.5.	Diseño del control.....	16
5.	Simulación del lazo de control diseñado en Simulink .....	22
6.	Integración del laboratorio virtual con LMS a través del estándar SCORM. ....	32
6.1.	Herramientas utilizadas.....	32
6.1.1.	Easy Java Simulations .....	32
6.1.2.	SCORM.....	32
6.2.	Control en posición del motor.....	42
6.2.1.	Modelado matemático.....	42
6.2.2.	Implementación en EJS .....	42
6.2.3.	Solución de la práctica. ....	48
6.3.	Control en velocidad del motor.....	54
6.3.1.	Modelado matemático.....	54
6.3.2.	Implementación en EJS .....	54
6.3.3.	Solución de la práctica. ....	60
6.4.	Control del sistema barra-bola.....	66
6.4.1.	Modelado matemático.....	66
6.4.2.	Implementación en EJS .....	67
6.4.3.	Solución de la práctica. ....	74
7.	Conclusión .....	80
8.	Referencias.....	82
9.	Bibliografía .....	83
10.	ANEXO.....	84
10.1.	Ejemplo de utilización del EJS .....	84
10.2.	Prácticas diseñadas.....	87

10.3. Solucionario de las prácticas diseñadas. ....88



## 1. Introducción

La iniciativa para realizar este proyecto surge de la necesidad de efectuar este sistema barra-bola en un laboratorio virtual para el alumnado del grado de ingeniería electrónica industrial en asignaturas como ingeniería de control y para el alumnado del grado de ingeniería eléctrica en asignaturas como regulación de automática. De esta manera asignaturas con gran contenido teórico pasarían a tener contenido práctico de una gran importancia donde el alumnado de éstas podrá afianzar los conocimientos teóricos con problemas prácticos como podría ser el control de este sistema barra-bola.

Además, ofrece la oportunidad de trabajar online en el problema, presentando ventajas como no tener que requerir del apoyo de varios equipos físicos, ser más económicos, pudiéndose emplear a cualquier hora y no siendo necesario desplazarse hasta el lugar donde esté ubicado. Este planteamiento se podría aprovechar en situaciones como la que se está viviendo actualmente donde se han tenido que transformar por completo las clases presenciales por clases totalmente virtuales, así el alumnado continuaría realizando sus problemas prácticos desde casa.

## 2. Objetivos

El objetivo principal es el estudio de la evolución de aquellas variables que influyen y son significativas en el control del sistema barra-bola para efectuar un control más preciso y viable a la vez que se aborda el control de un sistema más complejo. Todo ello, gracias a la modificación del sistema barra-bola para ralentizar la dinámica del proceso.

Con el propósito de alcanzar el objetivo principal se han definido una serie de objetivos específicos:

- Realizar el modelado del sistema barra-bola.
- Diseñar un control eficaz y preciso de la posición de la bola en la barra.
- Implementar el control del sistema barra-bola en un laboratorio virtual.

También se intentaba implementar una maqueta para un laboratorio remoto pero debido a las circunstancias no ha sido posible ejecutarlo. Por ello, se ha hecho más hincapié en el diseño 3D del laboratorio virtual y en las posibles prácticas que se podrían llevar a cabo con éste.

### 3. Modelado matemático de la barra y bola.

El sistema barra-bola está compuesto por tres piezas principales: el motor, la barra y la bola. Como aparece en la Figura 3-1.

La bola se encuentra situada encima de la barra. La función de este sistema es mover la bola sobre la barra, para ello, el motor rota un ángulo,  $\Theta$ , y, por consecuencia, la barra se inclina un ángulo,  $\alpha$ , que termina generando el movimiento de la bola sobre la barra.

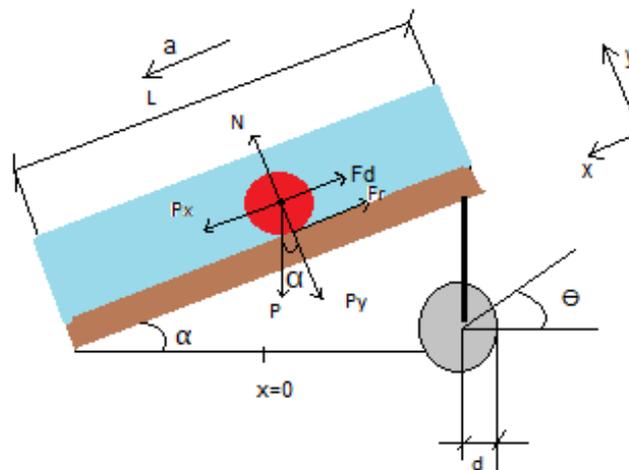


Figura 3-1. Esquema de perfil del sistema barra-bola

En primer lugar, se obtienen las ecuaciones que rigen el movimiento de la bola si la barra se encuentra inclinada como la Figura 3-1, para ello, se utiliza la segunda ley de Newton

$$\sum \vec{F}(t) = m * \vec{a}(t) \quad (1)$$

Se supone que la aceleración tiene un sentido positivo de derecha a izquierda por lo que todas las fuerzas que se opongan a la aceleración serán negativas y todas las que favorezcan a la aceleración serán positivas.

Las fuerzas que actúan sobre la bola serán:

- $P(t)$ , peso de la bola. Se descompone según el eje X en  $P_x(t)$  y según el eje Y en  $P_y(t)$ .
- $N(t)$ , fuerza normal.
- $F_r(t)$ , fuerza de rozamiento.
- $F_d(t)$ , fuerza de arrastre del fluido.

Esta última fuerza de arrastre del fluido aparece porque el sistema ha sido modificado debido a la rápida dinámica que hacía imposible su implementación en un laboratorio remoto. La bola se encuentra sumergida en un fluido dentro de un tubo cilíndrico que hace la función de barra. Esta modificación no altera el funcionamiento anteriormente explicado del sistema barra-bola.

Se descompone y se utiliza la segunda ley de Newton según los ejes X e Y.

Primero se hace el sumatorio de fuerzas en el eje Y que será igual a cero porque en el eje Y la aceleración es nula,  $\sum \vec{F}_y(t) = 0$

$$N(t) - P_y(t) = 0 \quad (2)$$

$$P_y(t) = m * g * \cos\alpha(t) \quad (3)$$

$$N(t) - m * g * \cos\alpha(t) = 0 \quad (4)$$

Siendo,

- $g$ , la constante de la gravedad.
- $m$ , masa de la bola.
- $\alpha$ , ángulo de inclinación de la barra.

Ahora se aplica en el eje X,  $\sum \vec{F}_x(t) = m * a(t)$

$$-Fr(t) - Fd(t) + P_x(t) = m * a(t) \quad (5)$$

$$P_x(t) = m * g * \text{sen}\alpha(t) \quad (6)$$

$$-Fr(t) - Fd(t) + m * g * \text{sen}\alpha(t) = m * a(t) \quad (7)$$

$Fr(t)$ , es la fuerza de rozamiento que hace que ruede la bola sobre la barra sin deslizar. Esto se conoce como la condición de rodadura que relaciona la aceleración con la que se traslada el centro de masas (CM) y la aceleración angular de rotación del sólido.

Puesto que la variación del ángulo girado es la aceleración angular de rotación de la bola, se obtiene la siguiente ecuación:

$$a(t) = R * a_\alpha(t) \quad (8)$$

Siendo,

- $a_\alpha$ , la aceleración angular de la bola
- $R$ , radio de giro de la bola
- $a(t)$ , aceleración lineal de la bola.

Aplicando el movimiento de rotación alrededor de un eje que pasa por el CM

$$Fr(t) * R = J_{bola} * a_\alpha(t) \quad (9)$$

Siendo,

- $J_{bola}$ , el momento de inercia de una esfera sólida.

$$J_{bola} = \frac{2}{5} * m * R^2 \quad (10)$$

Por lo que si se sustituye en la ecuación ( 7 ):

$$Fr(t) * R = \frac{2}{5} * m * R^2 * a_{\alpha}(t); Fr(t) = \frac{2}{5} * m * R * a_{\alpha}(t) \quad ( 11 )$$

Sustituyendo la ecuación ( 8 ) en la ( 11 ):

$$Fr(t) = \frac{2}{5} * m * a(t) \quad ( 12 )$$

Sustituyendo la ecuación ( 12 ) en la ( 7 )

$$-\frac{2}{5} * m * a(t) - Fd(t) + m * g * sen\alpha(t) = m * a(t) \quad ( 13 )$$

$Fd(t)$  es la fuerza de arrastre del fluido en el régimen laminar que es proporcional a la velocidad procedente de la ley de Stokes [1], su ecuación es:

$$Fd(t) = 6 * \eta * \pi * R * v_x(t) \quad ( 14 )$$

Siendo

- $\eta$ , la viscosidad del fluido.
- $R$ , radio de giro de la bola.
- $v_x(t)$ , velocidad de la bola.

El fluido que se utiliza es glicerina rebajada con agua por lo que esa viscosidad de mezcla de fluido,  $\eta$ , dependerá del porcentaje de agua y glicerina. Este porcentaje variará en las simulaciones para cada alumno según el último dígito de usuario de ILIAS. La ecuación que la rige la viscosidad de la mezcla:

$$\eta = x * \eta_{glicerina} + (1 - x) * \eta_{agua} \quad ( 15 )$$

Siendo,

- $x$ , el porcentaje de glicerina de la mezcla.
- $\eta_{\text{agua}}$ , la viscosidad del agua.
- $\eta_{\text{glicerina}}$ , la viscosidad de la glicerina.

Si se sustituye la ecuación ( 14 ) en la ( 13 ):

$$-\frac{2}{5} * m * a(t) - 6 * \eta * \pi * R * v_x(t) + m * g * \text{sen}\alpha(t) = m * a(t) \quad ( 16 )$$

Como el ángulo de inclinación de la barra será muy pequeño, se realiza la simplificación donde se considera que  $\text{sen}\alpha=\alpha$ . Esta simplificación se debe a que el ángulo es tan pequeño que tiende a cero.

Esta cuestión se pondrá en práctica en la última simulación que se realiza. Reagrupando y simplificando la ecuación ( 15 ):

$$-\frac{2}{5} * m * a(t) - 6 * \eta * \pi * R * v_x(t) + m * g * \alpha(t) = m * a(t) \quad ( 17 )$$

La Figura 3-2 muestra un esquema de perfil de la barra y el motor. Observando la relación que existe entre el ángulo de rotación del motor y el ángulo de inclinación de la barra, se obtiene:

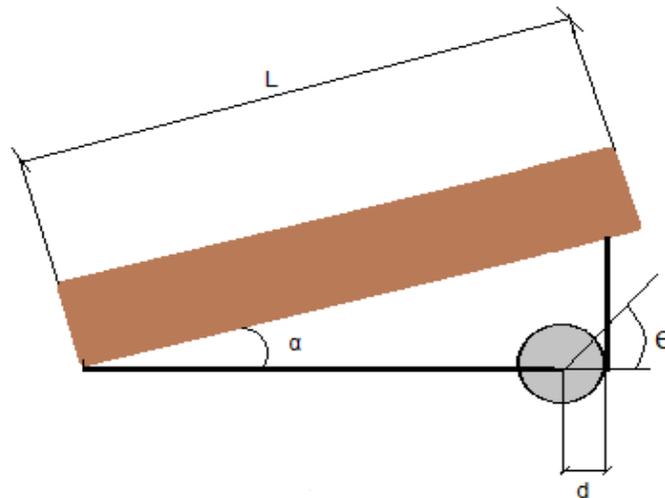


Figura 3-2. Esquema de perfil de la barra y el motor

$$\alpha = \frac{d}{L} \theta \quad (18)$$

Siendo,

- $d$ , offset del brazo en el motor
- $L$ , longitud de la barra
- $\theta$ , ángulo de rotación del motor.

Concluyendo, la función de transferencia del ángulo de inclinación del motor con respecto al ángulo de la barra será solamente una ganancia. De tal manera que:

$$H(s) = \frac{d}{L} \quad (19)$$

## 4. Dinámica del sistema.

En este apartado se obtendrá la dinámica de cada una de las partes del sistema barra-bola con el fin de obtener la dinámica de la planta completa. El objetivo es implementar un controlador adecuado.

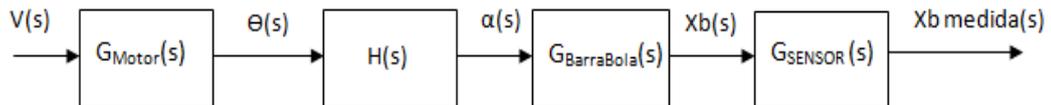


Figura 4-1. Dinámica del sistema

En la Figura 4-1 están indicadas las siguientes variables significativas:

- Variable controlada: Ángulo de giro del motor ( $\Theta$ ).
- Variable manipulada: Posición de la bola en el eje X ( $X_b$ ).
- Variable de proceso manipulada: Ángulo de inclinación de la barra en el eje X ( $\alpha$ ).
- Perturbaciones. Puede tener posibles perturbaciones que producen oscilaciones en la barra que modificarán la posición de la bola.

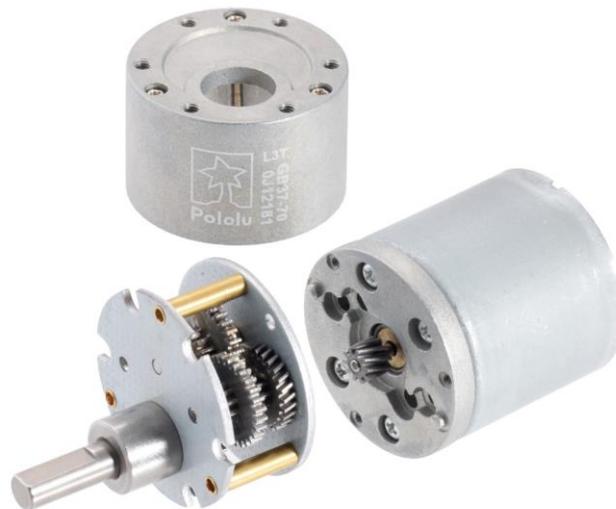
### 4.1. Dinámica del actuador: motor

El actuador elegido para este sistema barra-bola es un motorreductor.



**Ilustración 4-1: Actuador del sistema.**

Este motorreductor es un motor CC con escobillas capaz de funcionar a 12V que contiene una caja de engranajes con relación 70:1. Esta caja de engranajes (ilustración 3.3) está compuesta sobre todo de engranajes rectos pero en la primera etapa cuenta con engranajes helicoidales que reduce el ruido y lo vuelve más eficiente.



**Ilustración 4-2. Motorreductor con la caja de engranajes retirada.**

Al ser un motor con escobillas utiliza éstas conectadas al colector para realizar el cambio de polaridad en el rotor, por lo tanto, su función es conmutar mecánicamente la corriente de la bobinas al motor.

El bloque del motor tendrá una entrada u(s) y una salida y(s) como aparece en la Figura 4-2

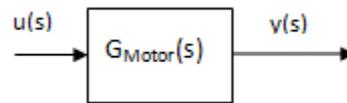


Figura 4-2. Bloque del motor

Por tanto

$$y(s) = G_{motor}(s) * u(s) \quad (20)$$

Realizando un ensayo (Figura 4-3) donde se aplica una entrada escalón de amplitud 100 ( $u(s) = \frac{100}{s}$ ) se consigue la velocidad del motor (en revoluciones por minuto) como se puede observar en la gráfica de la Figura 4-4.

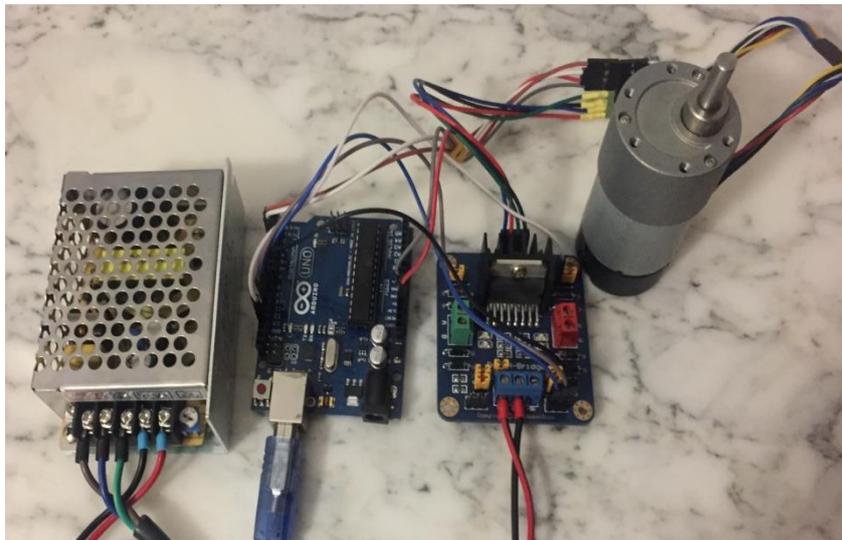


Figura 4-3. Ensayo del motor

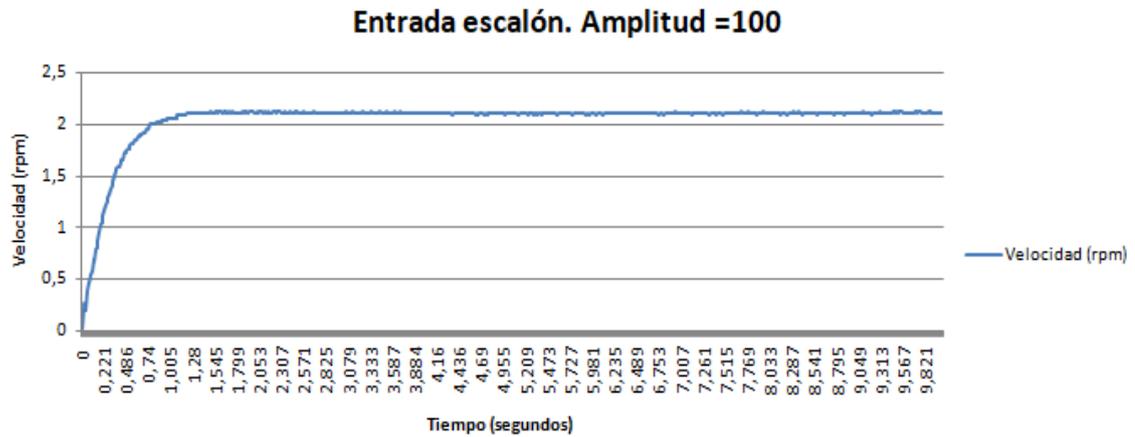


Figura 4-4. Gráfica del ensayo del motor

Al analizar la gráfica se puede ver que la velocidad se acerca a 2.1 rpm y se puede comenzar a definir el modelo del sistema.

Si se amplía la gráfica al transitorio como Figura 4-5 se puede observar que la respuesta del actuador tiene un comportamiento parecido a la respuesta de los sistemas de primer orden.

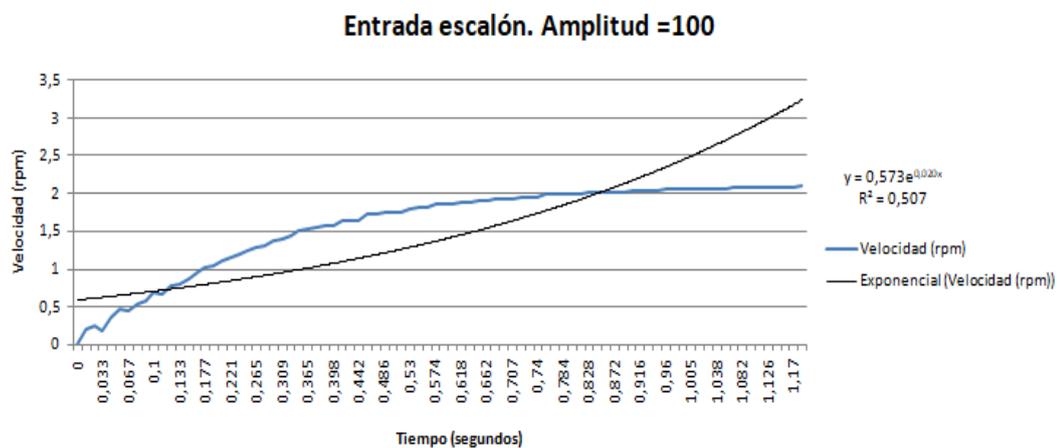


Figura 4-5. Gráfica 2 del ensayo del motor

Por tanto, la función de transferencia del actuador debe tener una estructura parecida a la ecuación ( 21 )

$$G_{motor}(s) = \frac{K}{\tau s + 1} \quad (21)$$

El cálculo del parámetro  $K$  y  $\tau$  se puede realizar utilizando las siguientes fórmulas para la identificación de los sistemas de primer orden.

$$K = \frac{\Delta y}{\Delta u} = \frac{2.1}{100} = 0.021 \quad (22)$$

$$y(\tau) = 0.632 * K * u = 0.632 * 0.021 * 100 = 1.3272 \quad (23)$$

Si se observa este valor de 1.3272 sobre el eje Y, se puede ver su equivalencia sobre el eje X que da el valor de  $\tau$  como aparece en la Figura 4-6

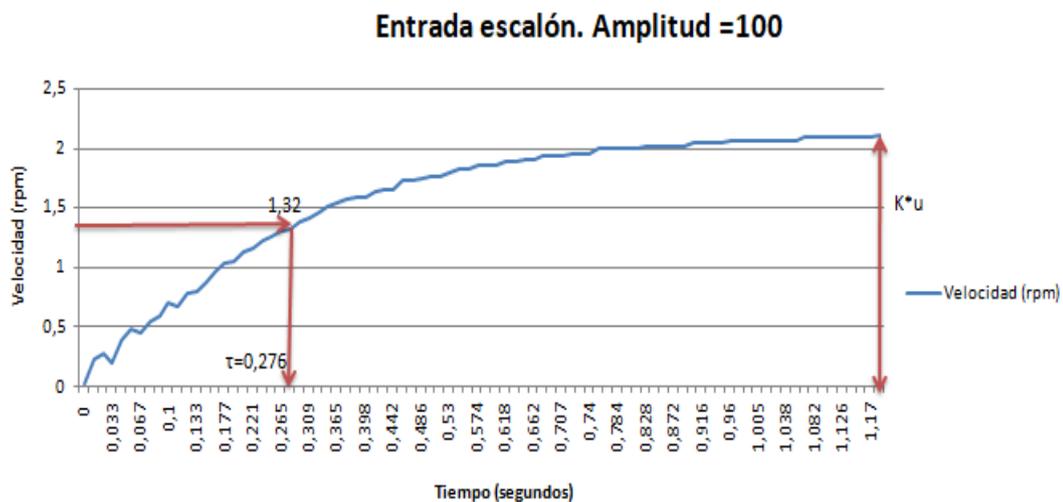


Figura 4-6. Gráfica 3 del ensayo del motor

Si se quiere la función de transferencia que relacione la entrada del motor con la salida del ángulo del motor simplemente se añade un integrador ( $1/s$ ) a la ecuación (21)

$$G_{motor}(s) = \frac{\theta(s)}{u(s)} = \frac{K}{\tau s + 1} * \frac{1}{s} = \frac{K}{s(\tau s + 1)} = \frac{\frac{K}{\tau}}{s(s + \frac{1}{\tau})} \quad (24)$$

#### 4.2. Dinámica de la planta: barra y bola.

En el modelado matemático del sistema se obtuvo la ecuación ( 17 ), si se aplica la transformada de Laplace se obtiene:

$$-\frac{2}{5}m * X(s) * s^2 - 6\eta\pi R * X(s) * s + mg * \alpha(s) = m * X(s) * s^2 \quad (25)$$

Reagrupando los términos:

$$mg * \alpha(s) = X(s) * \left(\frac{7}{5}m * s^2 + 6\eta\pi R * s\right) \quad (26)$$

Por lo tanto, la función de transferencia que relaciona la posición de la bola con el ángulo de la barra es

$$G_{\text{BarraBola}}(s) = \frac{X(s)}{\alpha(s)} = \frac{\frac{5g}{7}}{s\left(s + \frac{30\eta\pi R}{7m}\right)} \quad (27)$$

#### 4.3. Dinámica del sistema barra y bola.

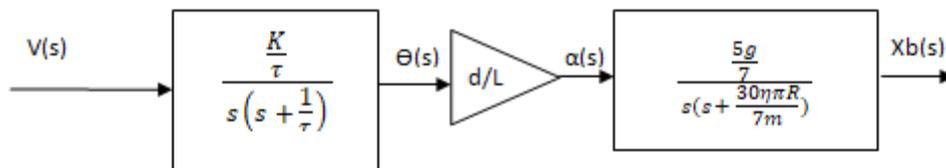


Figura 4-7. Dinámica del sistema con las funciones de transferencia

Agrupando todas las funciones de transferencia se obtiene la dinámica de la planta del sistema barra-bola definida en una sola función de transferencia en la que

se representará la posición de la bola en el eje X en función de la tensión a la que se alimenta el motor.

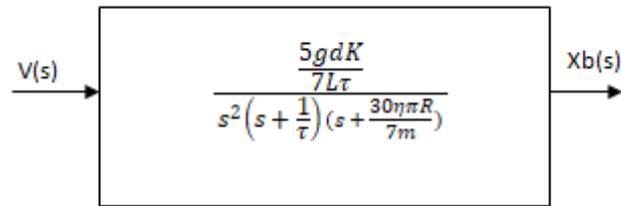


Figura 4-8. Dinámica del sistema simplificado

#### 4.4. Dinámica del sensor

El sensor elegido es un sensor software porque al tener la bola sumergida en el fluido se hace difícil utilizar un sensor láser.

La dinámica del sensor la realiza una aplicación que estaba ya creada cuya función es segmentar por color la bola y calcular la distancia euclídea entre el centro de la barra y el centroide de la bola.

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Esta distancia sirve para definir la distancia entre dos puntos en un espacio de tres dimensiones o en el plano, se obtiene en píxeles y se modifica a centímetros, ya que, se conoce la relación que hay entre el número de píxeles que contiene el área de la bola detectada y la dimensión real de la bola.

Teniendo la medida transformada en centímetros se puede realizar el control con ella.

$$G_{sensor} = D_{euclídea}(px) * \frac{A_{bolacm}}{A_{bolapx}} \quad (28)$$

Siendo

- $D_{euclídea}(px)$ , la distancia euclídea desde el centroide de la barra a la bola

- $A_{bolacm}$ , área de la bola en centímetros
- $A_{bolapx}$ , área de la bola en píxeles.

#### 4.5. Diseño del control

Los controladores PID incluyen tres acciones.

- El parámetro proporcional (P) depende del error actual. Da una salida del controlador que es proporcional al error.

$$u(t) = Kp * error(t)$$

$Kp$ , es la ganancia proporcional.

Este tipo de controlador se utiliza para aumentar la velocidad de respuesta.

Sin embargo, este valor  $Kp$  no puede aumentar infinitamente.  $Kp$  tiene un límite para que no comiencen a producirse sobreoscilaciones.

Además, del límite físico del actuador debido al cual se produce la saturación de la señal de control.

- El parámetro derivativo (D) no se usa de forma independiente

$$u(t) = Kp * Td * \frac{derror(t)}{dt}$$

Mejora el comportamiento, hace el proceso más rápido y con menos oscilaciones.

Evita las variaciones en el régimen transitorio.

- El parámetro integral (I) no se usa de forma independiente, se utiliza junto el proporcional.

$$u(t) = \frac{Kp}{Ti} * \int_0^t error(t) dt$$

Ayuda a disminuir y eliminar el error en estado estacionario, ya que, mientras existe error la acción integral evoluciona. Por lo tanto compensa las perturbaciones e intenta mantener la variable controlada en el punto de consigna.

Sin embargo, a partir de un valor mínimo produce oscilaciones.

Combinando proporcional, integral y derivativo, dará lugar a los siguientes controladores:

- Controlador proporcional-derivativo (PD):

$$u(t) = Kp * (error(t) + Td * \frac{derror(t)}{dt})$$

Td es el tiempo que se requiere para que la acción proporcional y derivativa que contribuyen a la salida del controlador lo hagan en la misma cantidad.

A mayor Td, mayor acción derivativa, y, a menor Td, menor acción derivativa.

Se utiliza porque al aumentar Kp, para aumentar la velocidad de respuesta, la acción derivativa ayuda a la estabilidad y evita que se produzcan oscilaciones.

- Controlador proporcional-integral (PI):

$$u(t) = Kp * (error(t) + \frac{1}{Ti} * \int_0^t error(t) dt)$$

$T_i$  es el tiempo que se requiere para que la acción proporcional e integral que contribuyen a la salida del controlador lo hagan en la misma cantidad.

A menor  $T_i$ , mayor acción integral, por lo que mayor velocidad de eliminación del error y viceversa.

- Controlador proporcional-integral-derivativo(PID):

$$u(t) = K_p * (error(t) + T_d * \frac{derror(t)}{dt} + \frac{1}{T_i} * \int_0^t error(t) dt)$$

Une las ventajas de cada una de las acciones de control en una sola acción de control.

La acción integral añade un polo al origen, cambiando el tipo del sistema.

La acción derivativa reduce las sobreoscilaciones en el régimen transitorio.

La acción proporcional reduce el tiempo de subida.

El sistema barra-bola ofrece la oportunidad de poder diseñar un control en cascada. Las situaciones cuando se debe utilizar este tipo de control es:

- El proceso que se quiere controlar tiene diferentes etapas
- Se tiene una variable manipulable y más de una variable medida.
- Las perturbaciones afectan directamente a la variable de proceso manipulada
- La variable controlada es mucho más lenta que las intermediarias.

El control en cascada tiene dos lazos. Un lazo primario con un controlador primario también llamado "maestro" y un lazo secundario con un controlador secundario también denominado "esclavo", siendo la salida del primario el punto de consigna del secundario. La salida del controlador secundario es la que actúa sobre el proceso.

Como se muestra en la Figura 4-9 en el sistema barra-bola el controlador maestro será el  $G_{c2}(s)$  y el controlador esclavo será el  $G_{c1}(s)$ . La salida del maestro es  $\Theta^*$ , que a su vez es la consigna del esclavo. La consigna del maestro será  $X_{b\_ref}$  y la salida del esclavo es  $\Theta$  que actúa sobre el proceso.

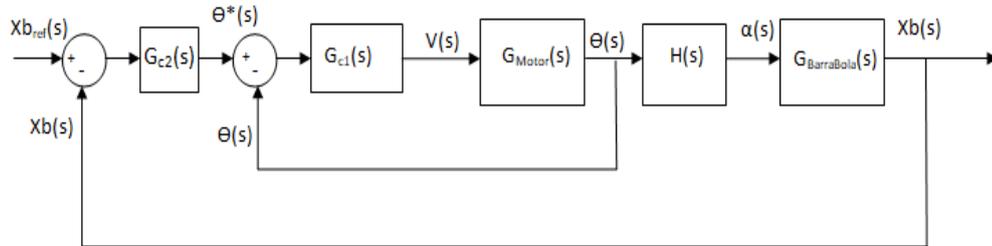


Figura 4-9. Control en cascada en lazo cerrado.

Por tanto, en el sistema barra-bola se reducen las posibles perturbaciones en el lazo secundario que tiene efecto sobre el lazo primario. Asimismo, reduce los efectos de retardo de tiempo porque al ser la dinámica interna muy rápida, no se siente en lazo primario [3].

Primero, se diseña el controlador esclavo,  $G_{c1}(s)$ , y después el controlador maestro,  $G_{c2}(s)$ .

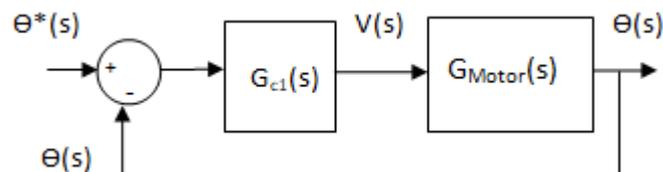


Figura 4-10. Lazo de control esclavo

Para realizar el control del primer bloque (Figura 4-10), se puede utilizar un controlador proporcional para evitar el error en régimen estacionario ( $e_{ss}=0$ ). Además, como se quiere adelantar la respuesta se añade la acción derivativa, por tanto, el resultado es un controlador proporcional-derivativo (PD).

$$G1(s) = G_{c1}(s) * G_{Motor}(s) = Kp1 * Td1 \left( s + \frac{1}{Td1} \right) * \left( \frac{\frac{Km}{\tau}}{s \left( s + \frac{1}{\tau} \right)} \right) \quad (29)$$

Para simplificar se puede definir  $Td1=\tau$ , utilizando la cancelación cero/polo.

$$G1(s) = Kp1 * \tau \left( s + \frac{1}{\tau} \right) * \left( \frac{\frac{Km}{\tau}}{s \left( s + \frac{1}{\tau} \right)} \right) = \frac{Kp1 * Km}{s} \quad (30)$$

De esta manera se reduce el polo no dominante más lejano del origen y al añadir un cero se contribuye a adelantar la respuesta.

$$G1_{bc}(s) = \frac{Kp1 * Km}{s + Kp1 * Km} \quad (31)$$

Creado el lazo del controlador esclavo del bloque correspondiente al motor, ahora, se realiza el controlador maestro. (Figura 4-11)

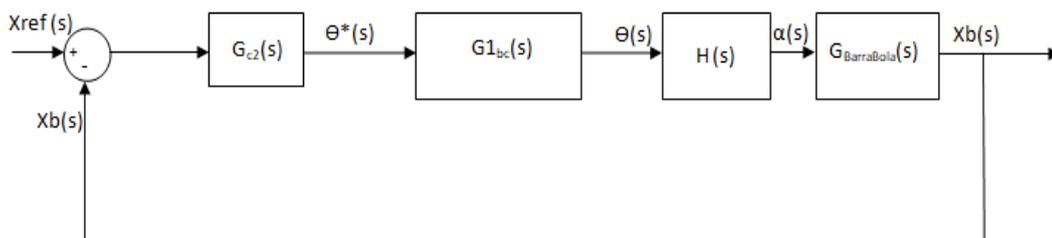


Figura 4-11. Lazo de control maestro

De nuevo, para realizar este controlador, se puede utilizar un controlador proporcional para evitar el error en régimen estacionario ( $e_{ss}=0$ ). Pero, además, como se quiere adelantar la respuesta se añade la acción derivativa, que añade un cero y contribuye a adelantar la respuesta. Por tanto, el resultado es un controlador proporcional-derivativo (PD).

$$G_{ba}(s) = G_{c2}(s) * G1_{bc}(s) * H(s) * G_{BarraBola}(s) \quad (32)$$

$$G_{ba}(s) = Kp2 * Td2 \left( s + \frac{1}{Td2} \right) * \frac{Kp1 * Km}{s + Kp1 * Km} * \frac{d}{L} * \frac{\frac{5g}{7}}{s \left( s + \frac{30\pi\eta R}{7m} \right)} \quad (33)$$

Ahora la función tiene un polo en el origen, un polo simple en  $-Kp1 * Km$ , un polo simple en  $-\frac{30\pi\eta R}{7m}$  y un cero simple en  $-\frac{1}{Td2}$ .

Si se consigue que el cero simple se encuentre cerca del polo en  $-\frac{30\pi\eta R}{7m}$  se producirá que este polo se cancela sin afectar al comportamiento del sistema, por lo tanto se fija  $Td2 = \frac{7m}{30\pi\eta R}$ .

$$G_{ba}(s) = \frac{\frac{5 * Kp1 * Kp2 * Td2 * Km * g * d}{7 * L}}{s(s + Kp1 * Km)} \quad (34)$$

Se agrupa en una sola variable todas las variables, quedando tal que así:

$$C = \frac{5 * g * d * Km * Kp1 * Td2}{7L} \quad (35)$$

Simplificando la ecuación (34):

$$G_{ba}(s) = \frac{C * Kp2}{s(s + Kp1 * Km)} \quad (36)$$

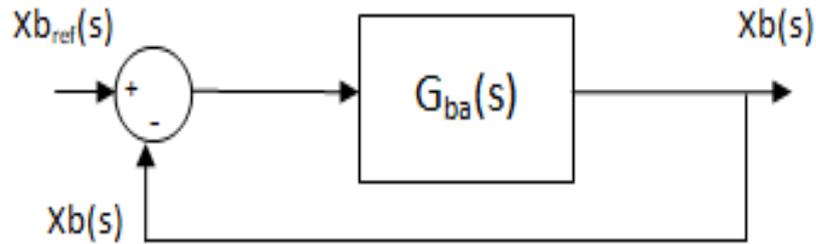


Figura 4-12. Lazo cerrado de control del sistema simplificado

Realizando la función de transferencia en bucle cerrado como se muestra en la Figura 4-12:

$$G_{bc}(s) = \frac{G_{ba}(s)}{1 + G_{ba}(s)} = \frac{C * Kp2}{s^2 + Kp1 * Kms + C * Kp2} \quad (37)$$

## 5. Simulación del lazo de control diseñado en Simulink

Se realiza un archivo en Matlab (Figura 5-1) con todas las constantes y parámetros que intervienen en los bloques del actuador la planta y el control. De esta manera si se decide cambiar algún parámetro simplemente se tendría que cambiar en este archivo.

```

Ra=0.025;%radio de la bola
m=0.05;%masa
nv=0.415;%viscosidad
g=9.81;
L=0.06;%longitud barra
d=0.022;%offset del brazo motor

Km=0.021;
Tm=0.276;
ts=40;
kpl=8/(ts*Km);
tdl=Tm;
td2=(7*m)/(30*pi*Ra*nv);
kp2=(7*kpl*Km*L)/(5*4*d*g*td2);

```

Figura 5-1. Variables del sistema

Para realizar el control en cascada primero se comienza por el lazo de control más interno que pertenece al lazo de control del motor como aparece en la Figura 5-2.

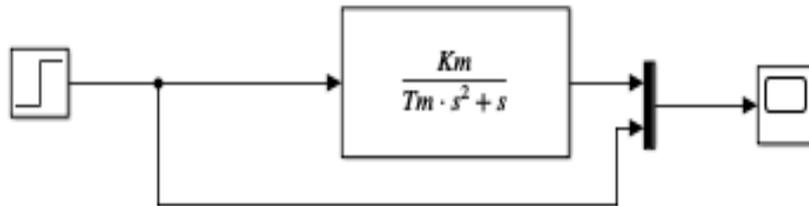


Figura 5-2. Diagrama de bloque del motor

Ante una entrada escalón de 1 se observa como la señal del motor tiende a crecer infinitamente.

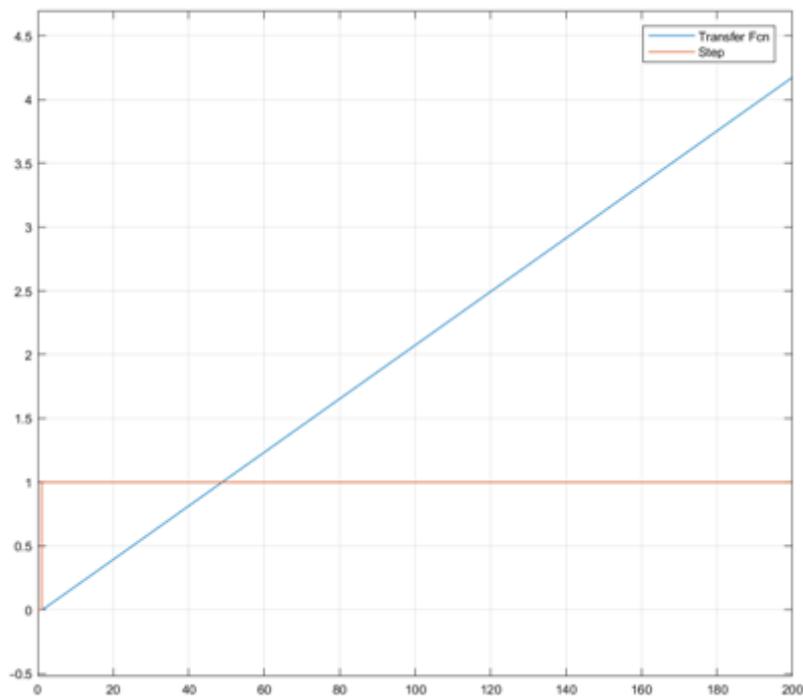


Figura 5-3. Gráfica del motor ante un escalón.

Como se trabajó anteriormente en el control del bloque del motor se añade un PD para ello utilizamos la herramienta de PID Tuner de Simulink.

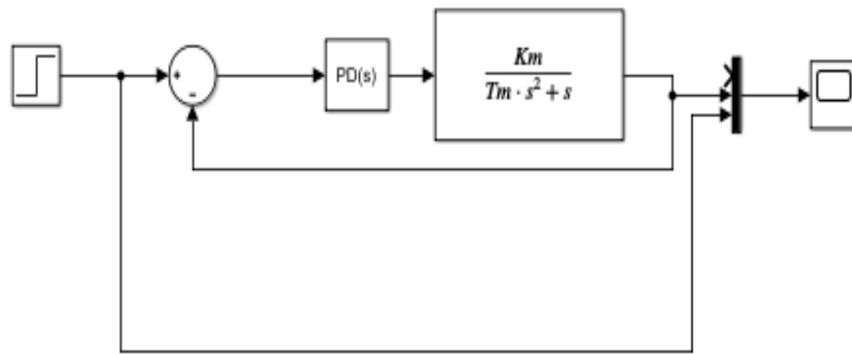


Figura 5-4. Lazo cerrado del control del actuador.

Este bloque tendrá una parte proporcional y otra derivativa a la cual se dará los valores antes obtenidos, también tiene una parte integral que no se utiliza en este caso.

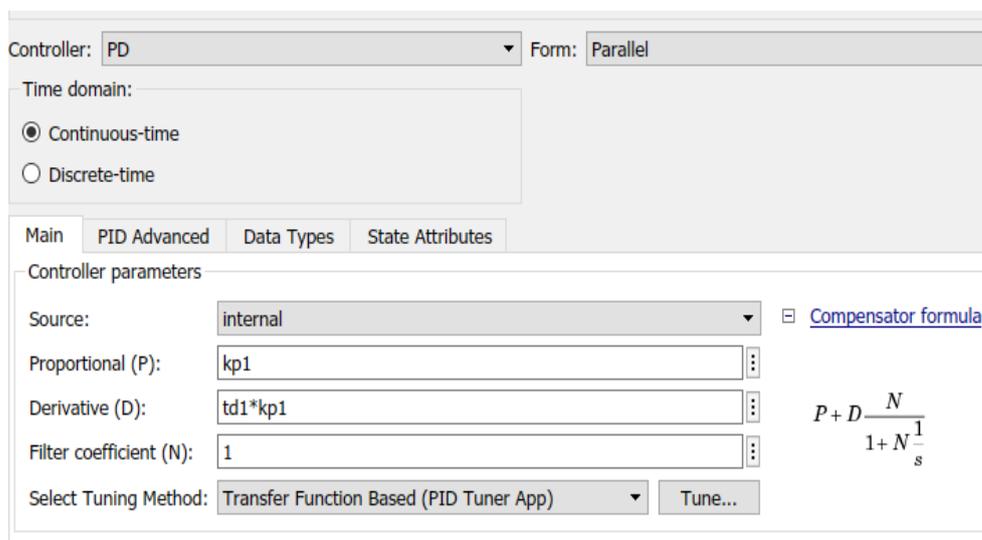


Figura 5-5. Controlador esclavo PD diseñado en PID tuner

Tras introducir los parámetros se aplican los cambios y se realiza la simulación con el controlador ahora diseñado (Figura 5-5). Para esta simulación se aplica una entrada de valor unitario y el objetivo es que el motor se mantenga estable en posición con respecto al valor de referencia con el mínimo amortiguamiento posible.

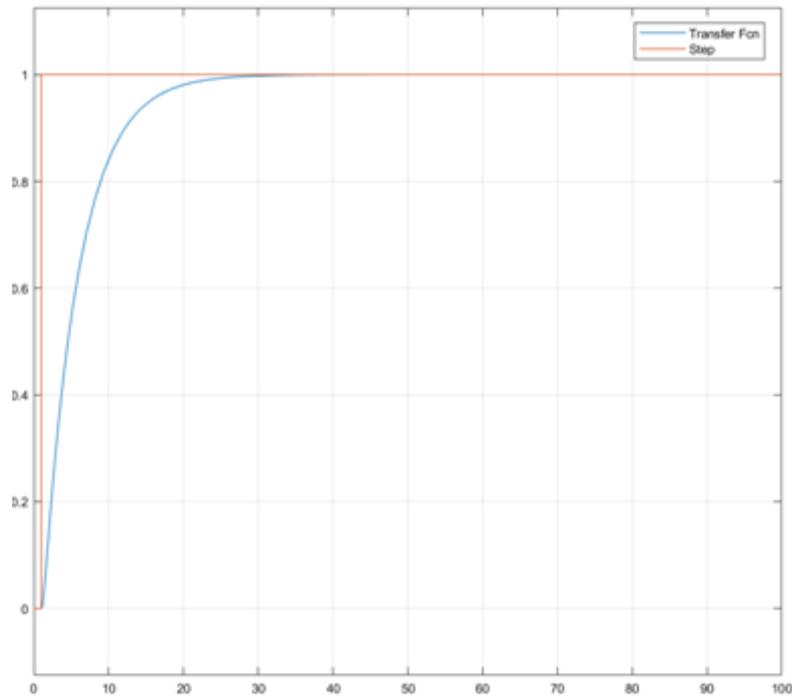


Figura 5-6. Respuesta del lazo cerrado del motor ante una entrada escalón.

Como se observa en la gráfica obtenida en la simulación (Figura 5-6) se cumple la condición de que el motor se mantenga estable en la referencia inicial.

Otra forma de diseñar el controlador es diseñar el propio bloque del controlador de la manera que se muestra en la Figura 5-7.

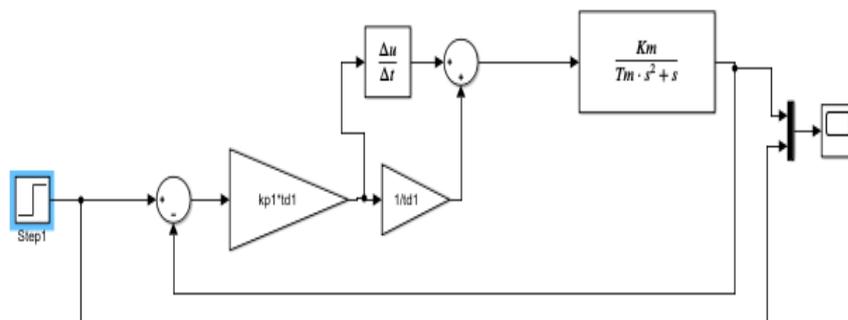


Figura 5-7. Diagrama de bloques del control del motor

De nuevo ante una referencia de una entrada inicial de valor unitario la respuesta debe de ser igual que anteriormente que la posición del motor se mantenga estable en la referencia inicial.

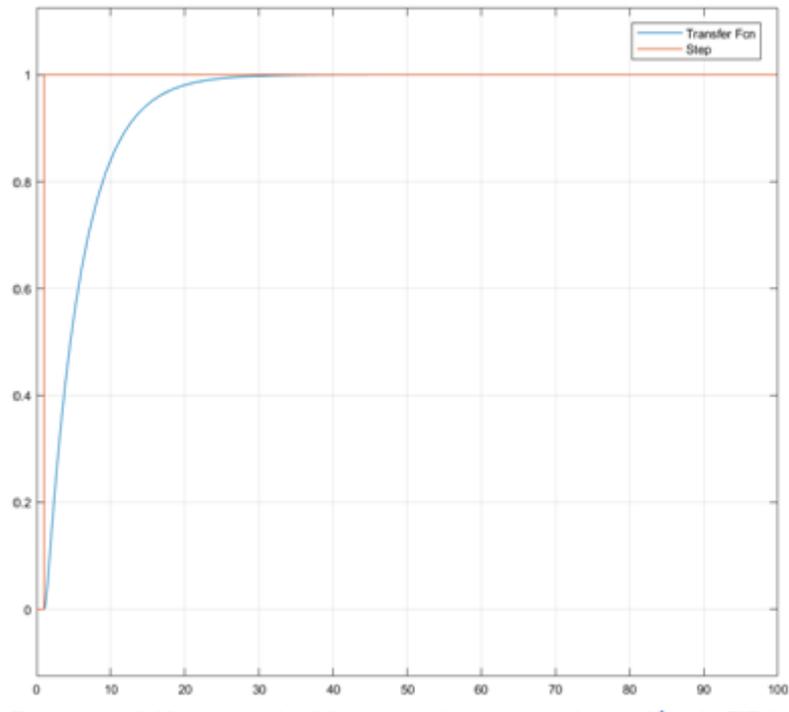


Figura 5-8. Respuesta del lazo cerrado del motor ante una entrada escalón sin PID tuner.

Realizado el control del lazo más interno se realizará el segundo controlador que pertenece al lazo más externo (Figura 5-9)

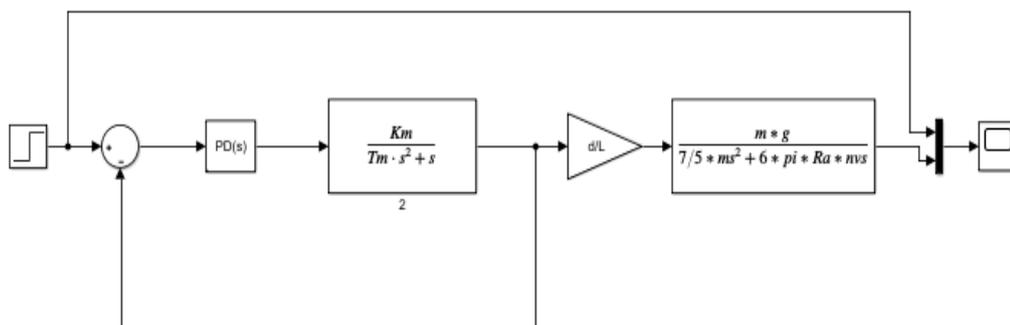


Figura 5-9. Diagrama de bloques del sistema barra-bola

Se añade el bloque del segundo controlador que volverá a ser un PD y se agrupa todo en un solo bloque

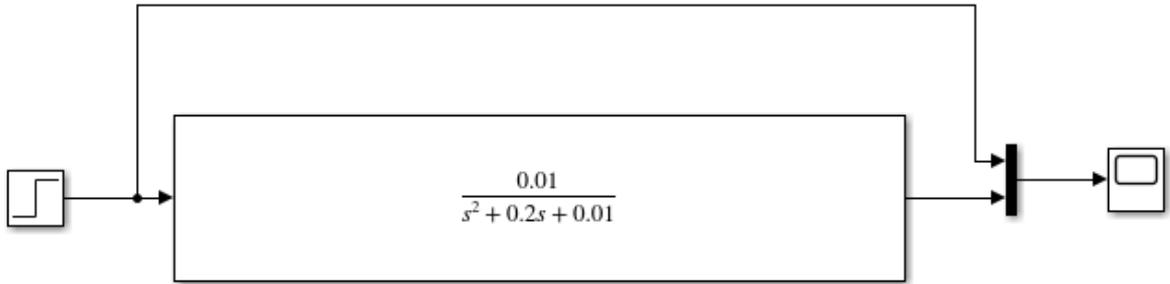


Figura 5-10. Diagrama de bloque de todo el control

Tras introducir los parámetros en el PD controller se aplican los cambios y se realiza la simulación con el control por completo diseñado.

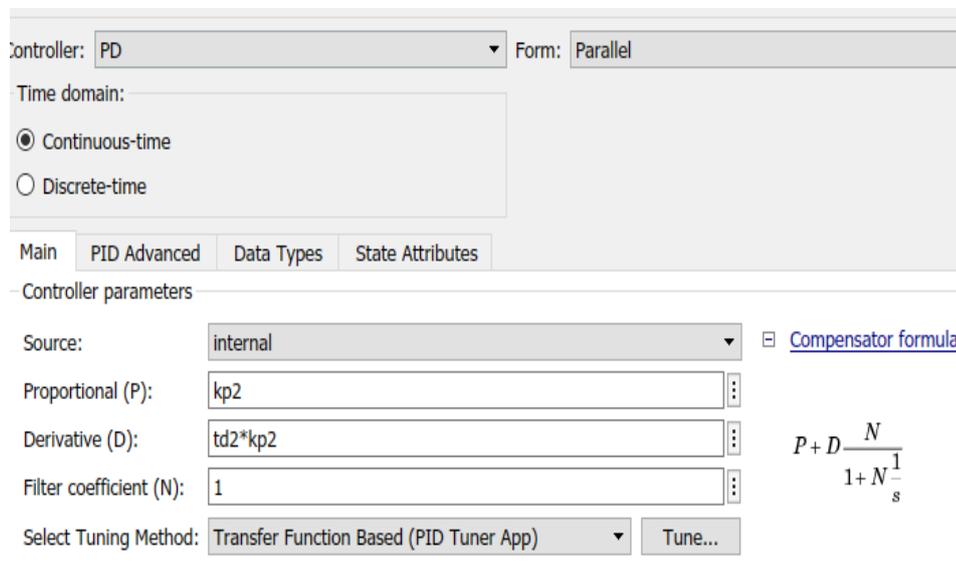


Figura 5-11. Controlador maestro PD diseñado en PID tuner

Para esta simulación se aplica una entrada de valor unitario y el objetivo es que la bola se mantenga estable en posición con respecto al valor de referencia con el mínimo amortiguamiento posible.

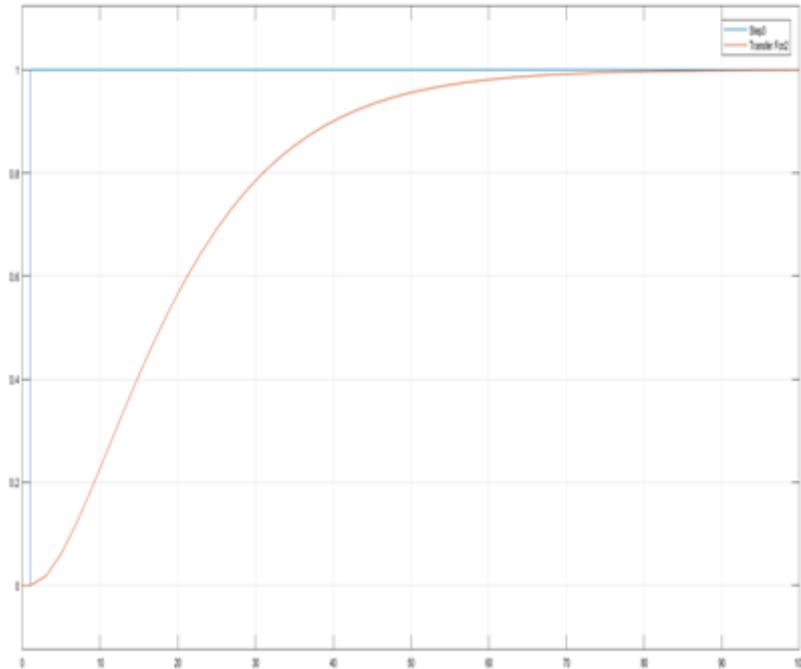


Figura 5-12. Respuesta del lazo cerrado ante una entrada escalón.

Se observa en la gráfica (Figura 5-12) como la respuesta de la posición de bola se mantiene estable ante una entrada de escalón de referencia unitaria que sería la referencia.

Otra manera de abordar la realización del control sin tener que ser un control en cascada, es realizando un controlador que controle todo el lazo del sistema. Para ello se obtiene la función de transferencia en lazo abierto (Figura 5-13) y se calcula el lugar de las raíces de esta función de transferencia para conseguir los polos y ceros que contiene. Se utiliza el comando de MATLAB `rlocus()` para mostrar el lugar de las raíces como aparece en la Figura 5-14

$$gplanta = \frac{0.05396}{s^4 + 6.417 s^3 + 10.12 s^2}$$

Figura 5-13. Función de transferencia

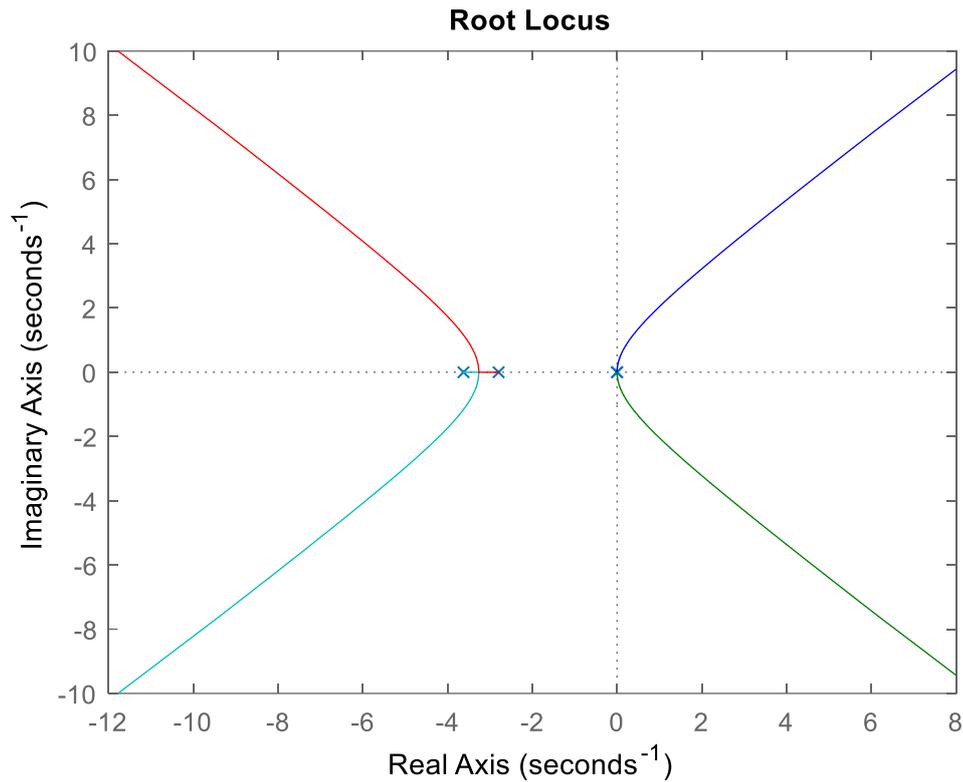


Figura 5-14. Lugar de las raíces

Se puede observar dos polos en el origen, un polo en -2.791 y otro en -3.626, por lo que se necesita un control PD porque un PID añadiría otro polo más. Para diseñar este controlador se van a añadir un cero en el origen y añadiendo una acción derivativa que añada otro cero en -2.791 y otro cero más entre 0 y -3.626 para conseguir que el sistema sea estable.

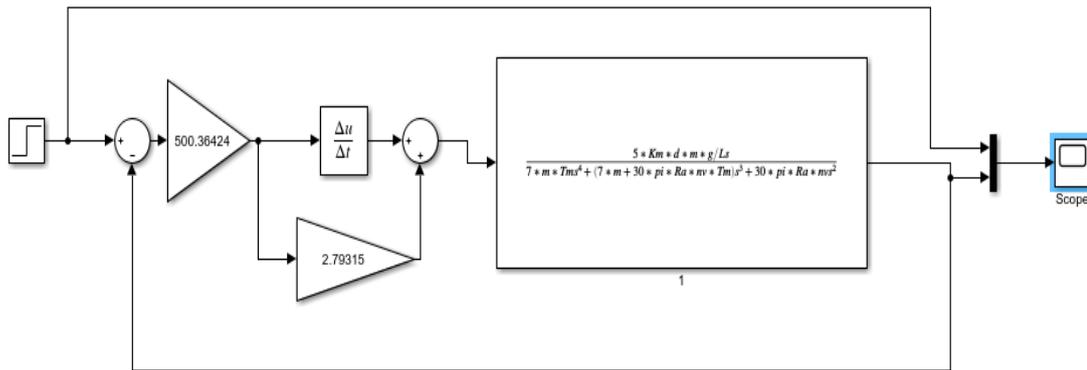


Figura 5-15. Diseño controlador lazo cerrado

El objetivo de realizar este controlador tiene similitud con el control en cascada se debe estabilizar la posición de bola ante la posición de referencia que se ha establecido en la entrada. En este apartado se aplica una entrada en escalón de valor unitario para establecer la referencia. Obteniendo la siguiente gráfica (Figura 5-16)

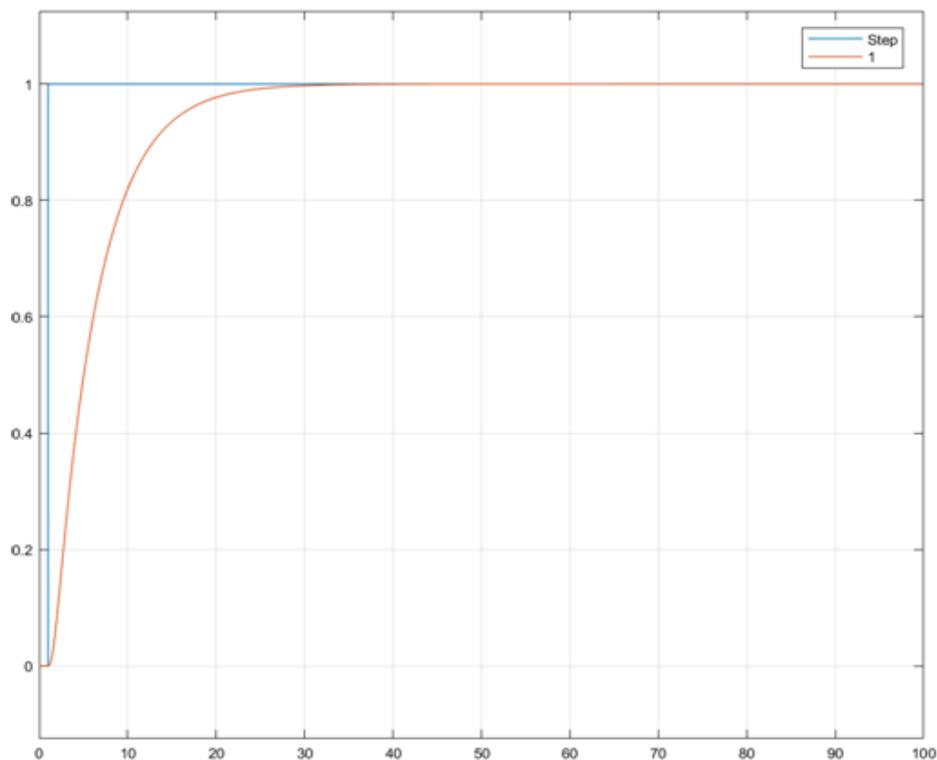


Figura 5-16. Posición estable de la bola

Parece ser que es un buen resultado el obtenido, sin embargo, si se acerca la gráfica se puede ver que la posición de la bola supera la referencia marcada, el error es muy pequeño pero existe, además de unas pequeñas sobreoscilaciones.

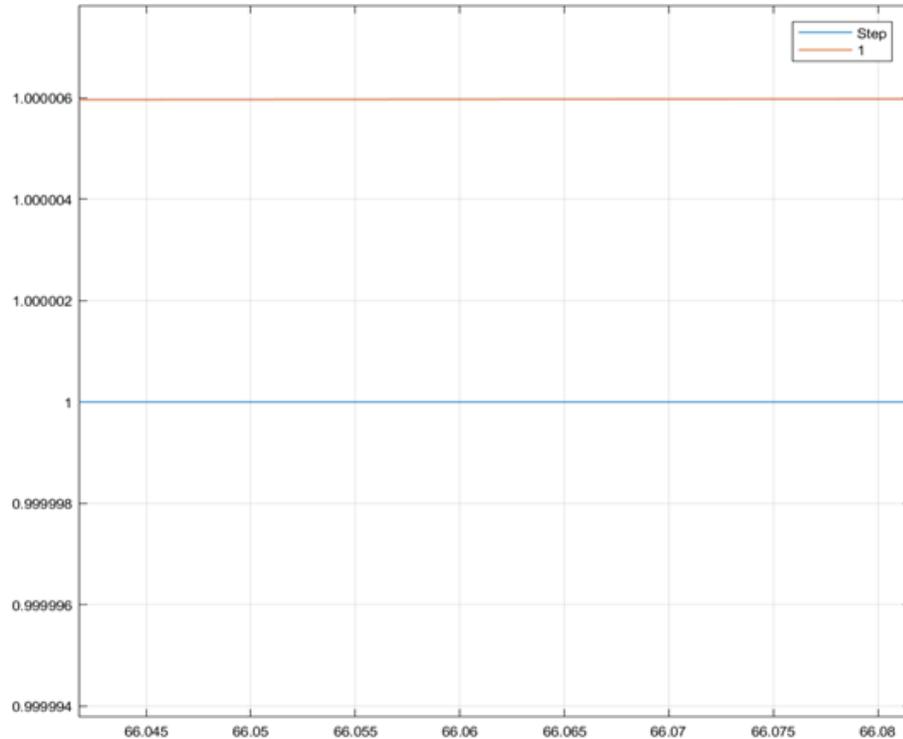


Figura 5-17. Zoom posición estable de la bola

En conclusión, el control en cascada es mejor porque no se produce este error entre la posición que se quiere alcanzar y la que realmente alcanza el sistema, siendo el control en cascada más preciso. Además, de la desventaja de que esta segunda forma de realizar el control es que se debe ajustar, sin embargo, el método de control en cascada es más preciso y real ya que el controlador se diseña acorde a los parámetros y variables del sistema barra-bola.

## **6. Integración del laboratorio virtual con LMS a través del estándar SCORM.**

En este apartado se desarrolla los pasos llevados a cabo para implementar los laboratorios virtuales diseñados con el LMS de la Universidad de Jaén a través del estándar SCORM.

### **6.1. Herramientas utilizadas.**

A continuación, se explica que es Easy Java Simulations, donde se han creado las simulaciones, y el estándar SCORM para integrar las simulaciones del laboratorio virtual en el LMS de la Universidad de Jaén, plataforma ILIAS.

#### **6.1.1. Easy Java Simulations**

En este proyecto se ha utilizado la aplicación informática Easy Java Simulations (EJS), creando simulaciones que pueden sustituir o complementar ejercicios de prácticas con unas ventajas que benefician a la formación del alumnado.

La elección de utilizar EJS para diseñar el control del laboratorio virtual del sistema barra-bola es debido a la eficacia y flexibilidad de esta herramienta, asimismo, por ser bastante empleada en el campo técnico para diseñar simulaciones por ordenador. Otras ventajas que han motivado su elección son:

- Accesibilidad desde ordenadores personales alejados del equipo que alberga la aplicación.
- Facilidad de uso.
- Es fácilmente compatible con las diferentes configuraciones que los otros ordenadores personales puedan poseer.

#### **6.1.2. SCORM**

El estándar utilizado para llevar a cabo la implementación es el estándar SCORM. SCORM define una manera específica de construir un sistema de gestión de aprendizaje, capacitación y contenido de cursos de manera que se pueda ser compartido con otros sistemas que utilicen SCORM.

La versión que se utiliza es la SCORM 2004 es la versión más compleja de SCORM y contempla una secuenciación de contenido con diferentes caminos que el usuario puede tomar. [2]

Permite empaquetar su contenido creando un archivo XML que contiene toda la información para ponerlo en marcha.

En EJS es necesario introducir unas funciones para así tener la posibilidad de guardar ciertos resultados, puntuaciones del alumno o información personal entre otros.

Se ha utilizado la librería RTE Javascript (uja\_scorm\_rte.js) creada por Ildfonso Ruano Ruano. [4]

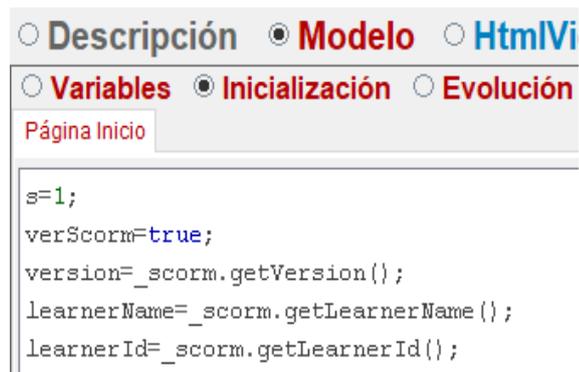
Primero, se crea una página de variables Runtime Environment, RTE, (Figura 6-1) para utilizar con las funciones que se detallan en los siguientes apartados y que son utilizadas en las tres simulaciones que posteriormente se explican.

<input type="radio"/> Descripción <input checked="" type="radio"/> <b>Modelo</b> <input type="radio"/> HtmlView		
<input checked="" type="radio"/> <b>Variables</b> <input type="radio"/> Inicialización <input type="radio"/> Evolución <input type="radio"/> Relaciones fijas <input type="radio"/> Propio <input type="radio"/> Elementos		
<input type="checkbox"/> Significativas <input type="checkbox"/> Control <input type="checkbox"/> Visual <input type="checkbox"/> puntuacion <input type="checkbox"/> english/spanish <input checked="" type="checkbox"/> RTE		
Nombre	Valor inicial	Tipo
learnerName	""	String
learnerId	""	String
version	""	String
verScorm	true	boolean
scoreRaw	""	String
scoreMax	""	String
scoreMin	""	String
comments	"Hola"	String
interacion	0	int
peso	50	double
completionStatus	"unknown"	String
sucessStatus	"unknown"	String
navReq	"continue"	String
esp	"ES"	String
eng	"EN"	String
idioma	""	String
		String
Comentario		

Figura 6-1. Variables RTE

#### 6.1.2.1 getVersion, getLearnerId y getLearnerName.

Estas tres funciones dan la posibilidad de obtener la versión del archivo, el Id de ILIAS del usuario que está realizando la simulación y nombre y apellidos. Todas ellas se utilizan en la página de inicialización.



```

 Descripción
 Modelo
 HtmlVi

 Variables
 Inicialización
 Evolución

Página Inicio

s=1;
verScorm=true;
version=_scorm.getVersion();
learnerName=_scorm.getLearnerName();
learnerId=_scorm.getLearnerId();

```

Figura 6-2. Inicialización de getVersion, getLearnerId y getLearnerName

Y se muestran en nuestra vista HTML en el panel superior.



Figura 6-3. Vista HTML del panel superior

#### 6.1.2.2 `getScoreMax`, `getScoreMin`, `getScoreRaw`, `setScoreMax`, `setScoreMin` y `setScoreRaw`.

Estas seis funciones permiten según si son las de tipo set, darle un valor a la puntuación máxima (`ScoreMax`), mínima (`ScoreMin`) que tenga asignado la simulación en total y la del momento que haya conseguido el usuario (`ScoreRaw`) y si son las de tipo get, obtener el valor de puntuación máxima, puntuación mínima y la puntuación que en ese preciso instante haya conseguido el usuario.

Al principio de la simulación, se inician estos valores y al final de cada apartado correspondiente se realiza un `setScoreRaw` para recoger la puntuación obtenida al igual que `setScoreMax` y `setScoreMin`.

```

finalpoints=score;
_scorm.setScoreRaw(finalpoints.toString());
_scorm.setScoreMin("0");
_scorm.setScoreMax("20");

```

Ilustración 6-1. Utilización de `setScoreRaw`, `setScoreMax` y `setScoreMin` al final de cada apartado

En función de si el alumno ha realizado la simulación anteriormente o no lo haya hecho le da la posibilidad de restablecer esa puntuación a cero o simplemente poder guardar la puntuación que tenía anteriormente y cerrar la simulación.

En esta primera imagen se muestra como se ha codificado en inicialización esta posibilidad.

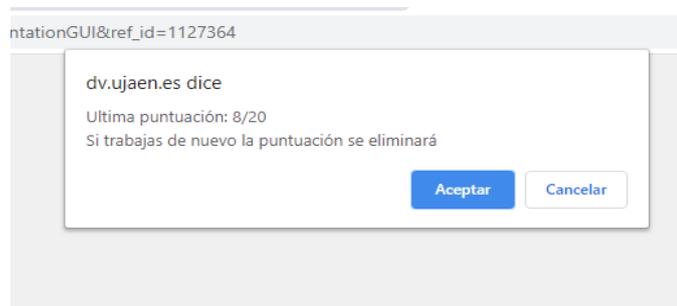
```

if ((scoreRaw > 0) && {scoreRaw<21}) {
    var n = confirm("Ultima puntuación: " + scoreRaw + "/20\nSi trabajas de nuevo la puntuación se eliminará");
} else {
    _scorm.setScoreRaw("0");
    scoreRaw=_scorm.getScoreRaw();
    var n = confirm("Ultima puntuación: " + scoreRaw + "/20\nNo has trabajado con esta simulación o no conseguistes puntos");
}
if(n) {
    _scorm.setScoreRaw("0");
    _scorm.setScoreMax("20");
    _scorm.setScoreMin("0");
} else {
    navReq="exitAll";
    _scorm.setNavRequest(navReq);
    _scorm.terminate();
}

```

**Ilustración 6-2. Inicialización de ScoreMax, ScoreMin y ScoreRaw**

En esta Figura 6-4 se contempla como sería el código antes implementado de manera visual.



**Figura 6-4. Ventana emergente apartado 1**

En esta Figura 6-5 se muestra los datos de seguimiento guardados en ILIAS.

Id del módulo de aprendizaje	Título del módulo de aprendizaje	Id del capítulo (SCO)	relevante para el progreso de aprendizaje	título del capítulo (SCO)	Usuario	cmi.score.raw: achieved score
1461012	315719	no	practica1scorm1versionbuena1	Montoro Lendínez, Alicia	10	0.97

**Figura 6-5. Datos de seguimiento de la puntuación**

Cuando se acepta la simulación comienza de nuevo totalmente y cuando se cancela se cierra, para realizar esta acción se utiliza dos funciones que se explican en el siguiente apartado.

### 6.1.2.3 setNavRequest y terminate.

La ejecución de ambas funciones permite el cierre de la simulación y el fin de la sesión. A parte de ser utilizadas en la ventana emergente de los apartados (cancelar), también son ejecutadas en el final de la simulación cuando se ha realizado correctamente todos los apartados.

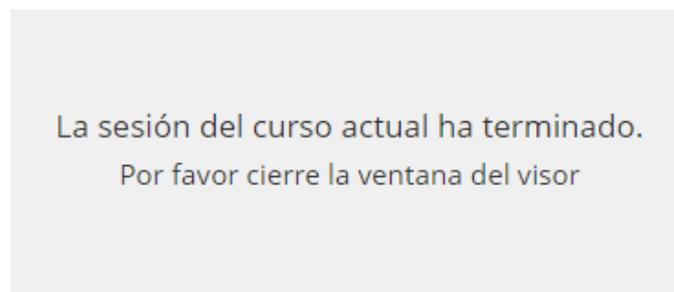


Figura 6-6. NavRequest y terminate

### 6.1.2.4 getLearnerPreferenceLanguage y setLearnerPreferenceLanguage

Guarda la preferencia de idioma, o bien español, o bien inglés que el alumno en la utilización de la simulación anteriormente ha predefinido. Para guardarlo se ejecuta en la parte del código del botón comprobar que sirve para comprobar si el resultado que el alumno ha escrito es el correcto.

```
if(espanol) {  
  _scorm.setLearnerPreferenceLanguage(esp);  
}else{  
  _scorm.setLearnerPreferenceLanguage(eng);  
}
```

Figura 6-7. Utilización setLearnerPreferenceLanguage

Y la siguiente vez que ejecute esta simulación recogerá ese dato para ejecutar la simulación en el idioma establecido anteriormente.

```

idioma=_scorm.getLearnerPreferenceLanguage();
if (idioma=='EM'){
    espanol=false;
}else{
    espanol=true;
}

```

Figura 6-8. Inicialización getLearnerPreferenceLanguage

Como se muestra en ILIAS la elección guardada de en este caso inglés (ENG). (Figura 6-9)

Id del módulo de aprendizaje	Título del módulo de aprendizaje	Id del capítulo (SCO)	relevante para el progreso de aprendizaje	título del capítulo (SCO)	Usuario	cmi.learner_preference.language
1461029	315767	no	practica2scorm1versionbuena	Montoro Lendinez, Alicia	EN	20

Figura 6-9. Datos seguimiento idioma

#### 6.1.2.5 setCompletionStatus y setSuccessStatus

Ambas funciones indican al final de la simulación cual es el estado de la simulación. La función setCompletionStatus guarda si el estado de la simulación es completo o incompleto y setSuccessStatus guarda si se ha pasado exitosamente o si por el contrario no se ha aprobado la simulación.

Título del módulo de aprendizaje	Usuario	estado total
Montoro Lendinez, Alicia	incomplete	notSatisfied

Figura 6-10. setCompletionStatus y setSuccessStatus

#### 6.1.2.6 setSessionTime

Indica el tiempo que el alumno ha necesitado en realizar la simulación. Se puede acceder a este dato en los datos de seguimiento del módulo de aprendizaje SCORM. (Figura 6-11)

(1 - 1 de 1)

Id del módulo de aprendizaje	Título del módulo de aprendizaje	Id del capítulo (SCO)	relevante para el progreso de aprendizaje	título del capítulo (SCO)	Usuario	cmi.session_time: time in the last session
1461029	315767	no	practica2scorm1versionbuena	Montoro Lendínez, Alicia	PT00H00M22.64S	23

Figura 6-11. Dato seguimiento del tiempo

### 6.1.2.7 setInteractionsId, setInteractionsType, setInteractionsTsAut, setInteractionsWeight, setInteractionsDescription, setInteractionsResponse, setInteractionsResult y getInteractionsResult

Todas estas funciones permiten interactuar con los apartados definidos en las simulaciones.

- setInteractionsId: Guarda el id que se define para esa interacción.
- setInteractionsType: Guarda el tipo de interacción es, en estos casos las respuestas siempre serán de tipo numérico pero también pueden ser, por ejemplo, de tipo true/false o múltiple respuesta.
- setInteractionsTsAut: Guarda el tiempo que tarda en realizar cada interacción el alumno, es decir, cada apartado por separado.
- setInteractionsWeight: Guarda el peso que tendrá cada apartado con respecto al total.
- setInteractionsDescription: Guarda la descripción de cada apartado.
- setInteractionsResponse: Guarda la respuesta del alumno.
- setInteractionsResult: Guarda la puntuación que ha obtenido el alumno con respecto a la respuesta que ha dado.
- getInteractionsResult: Recoge esa puntuación guardada anteriormente.

Todos los de tipo set serán inicializados y ejecutados en cada apartado para guardar cada correspondiente interacción según el correspondiente apartado.(Figura 6-12)

```
_scorm.setInteractionsId(interacion, "Apartado1");
_scorm.setInteractionsType(interacion, "numeric");
_scorm.setInteractionsTsAut(interacion);
_scorm.setInteractionsWeight(interacion, peso);
_scorm.setInteractionsDescription(interacion, "Apartado1. Esquema de control P
```

Figura 6-12. Inicialización de los setInteractions

Solo el de tipo get se ejecuta cuando en el segundo apartado se quiere sobrescribir una puntuación que había anteriormente o simplemente guardar la puntuación de ese apartado segundo por primera vez. Como resultado, aparece una ventana emergente dando al alumno la posibilidad de sobrescribir y continuar la simulación o cancelar y finalizar la sesión con la puntuación anteriormente guardada. (Figura 6-14)

```

if ({score2a}>=5) && {score2b}>=5} {
  p2OK=true;
  resp=resp+"- P2 OK (" +score2+"");
  var e = confirm("Ultima puntuación del apartado2a: " + _scorm.getInteractionsResult(1) + "/10\
if(e){
  finalpoints=score+score2;
}else{
  navReq="exitAll";
  _scorm.setNavRequest(navReq);
_scorm.terminate();
}
}

```

Figura 6-13. Ejecución de la ventana emergente apartado 2

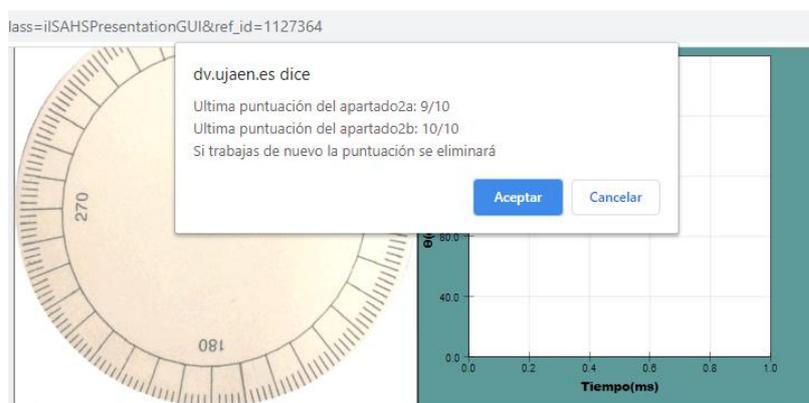


Figura 6-14. Ventana emergente apartado 2

Finalmente, todos estos datos pueden ser vistos desde la pestaña de datos de seguimiento del módulo de aprendizaje SCORM en ILIAS. (Figura 6-15)

título del capítulo (SCO)	Usuario	Id	cmi.interactions.n.weighting	cmi.interactions.n.type	cmi.interactions.n.result	Último acceso	cmi.interactions.n.learner_response: Response
practica1scorm1versionbuena1	Montoro Lendínez, Alicia	Apartado1	50	numeric	10	2020-06-10T11:42:03	1168
practica1scorm1versionbuena1	Montoro Lendínez, Alicia	Apartado2a	25	numeric	9	2020-06-10T11:42:13	140
practica1scorm1versionbuena1	Montoro Lendínez, Alicia	Apartado2b	25	numeric	10	2020-06-10T11:42:13	0.29

Figura 6-15. Datos seguimientos de las interacciones

### 6.1.2.8 Simulación en ILIAS

Para poder mostrar las simulaciones en ILIAS se empaqueta el archivo EJS como SCORM siguiendo estos pasos:

Se hace clic en el botón derecho sobre el icono  que se encuentra en el panel vertical del lado derecho del EJS.

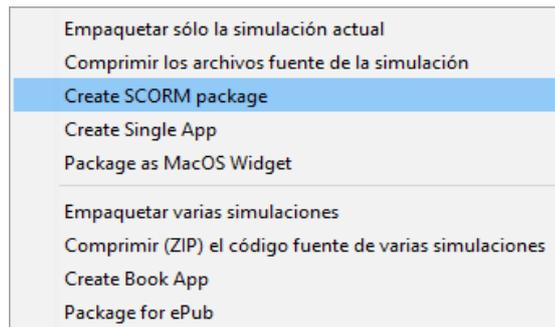


Figura 6-16. Empaquetar en SCORM

Y a continuación se elige la versión y si se desea añadir páginas de descripción.

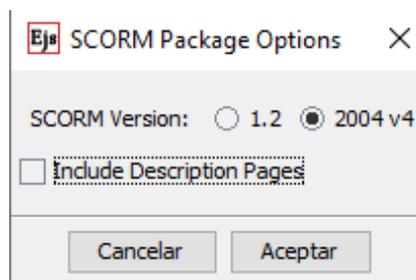


Figura 6-17. Versión SCORM

Finalmente, se guarda el archivo.

Para añadir el SCORM a nuestra plataforma LMS que es el sistema de gestión de aprendizaje que se va a utilizar para dejar el contenido a todos los usuarios, se sigue los siguientes pasos:

Se abre el espacio virtual donde se desea añadir el archivo. Se busca añadir nuevo elemento y se selecciona el módulo de aprendizaje SCORM.

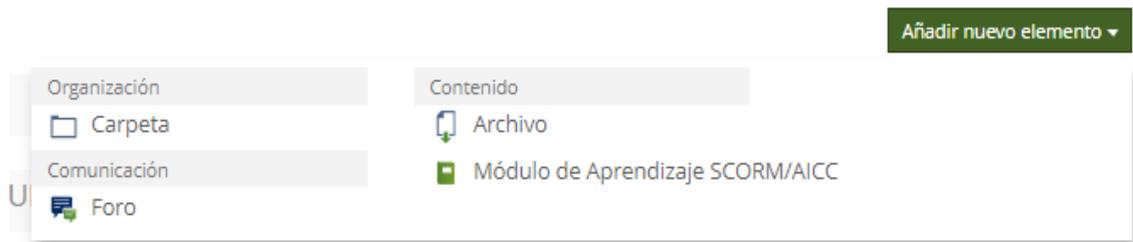


Figura 6-18. Añadir módulo de aprendizaje SCORM

Se importa el archivo que antes se había empaquetado como SCORM en EJS y se selecciona la configuración, por ejemplo, si se quiere que aparezca el nombre primero y después los apellidos o el menú de depuración o el estado activo de este módulo. En la siguiente imagen se muestran algunas de las opciones.

DISPONIBILIDAD	
Activo	<input checked="" type="checkbox"/> Establecer el módulo de aprendizaje SCORM como activo para hacerlo visible y que esté disponible para los usuarios. En otro caso, solo los usuarios con permisos de 'Edición' tendrán acceso.
Permitir el modo sin conexión	<input type="checkbox"/> En el modo fuera de línea, los módulos de aprendizaje sólo pueden procesarse con el SCORM desconectado. Los datos de seguimiento se transfieren si el alumno hace clic en "terminar el modo fuera de línea" en ILIAS.
PRESENTACIÓN	
Mostrar	<input checked="" type="radio"/> En pestaña / nueva ventana con enlace <input type="radio"/> Abrir en la misma ventana mostrando el menú principal de ILIAS <input type="radio"/> Abrir definitivamente en una ventana nueva sin el menú principal de ILIAS
Desactivar Barra de navegación superior	<input checked="" type="checkbox"/> La barra de navegación superior puede ocultarse si el módulo de aprendizaje SCORM dispone de suficientes opciones para pararse, finalizar y navegar por su contenido.
Esconder el árbol de navegación.	<input checked="" type="checkbox"/> El árbol de navegación izquierdo puede ocultarse si el módulo de aprendizaje SCORM dispone de las opciones suficientes para navegar por el contenido o si se muestra la barra de navegación superior.
Cargar el último capítulo visitado (SCO/Asset)	<input checked="" type="checkbox"/> Abre el último SCO/Asset visitado cuando el usuario regresa al módulo de aprendizaje.
OPCIONES SCORM	
Modo de lección por defecto	<input checked="" type="radio"/> Normal

Figura 6-19. Opciones configuración SCORM

## 6.2. Control en posición del motor

### **Enunciado.**

Se desea realizar el control en posición de un motor en CC cuyas constantes son  $K$  y  $\tau_m$ , estas constantes irán cambiando en función del número de usuario del alumno,  $N$ .

Apartado 1. ¿Qué controlador es necesario sintonizar para que se encuentre el sistema críticamente amortiguado?

Apartado 2. Si se requiere un tiempo de establecimiento  $t_{s5}$  con el criterio del 5%, ¿sería posible con un controlador proporcional? Si la respuesta es negativa, ¿qué tipo de controlador se debería sintonizar? Realice ese tipo de controlador.

### **6.2.1. Modelado matemático.**

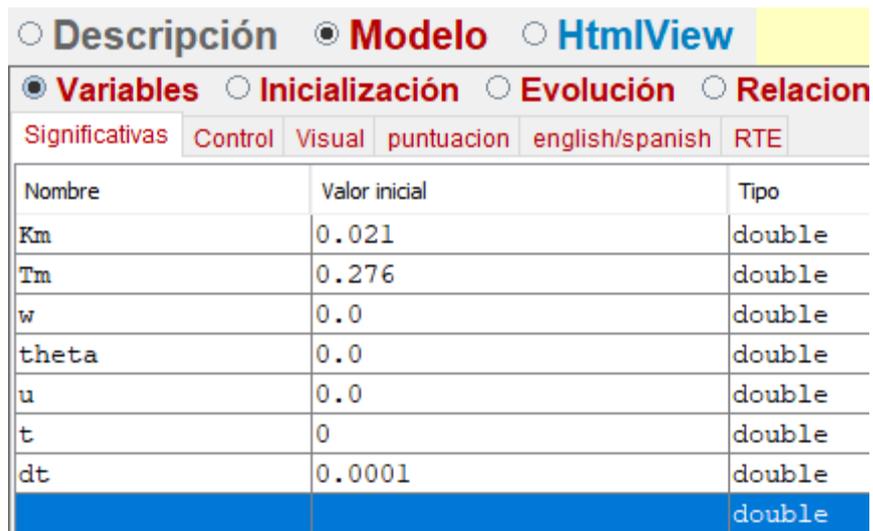
Para la función de transferencia del motor se realiza los mismos pasos que en el apartado 4.1 y, por tanto, se utiliza la ecuación ( 24 )

$$G_{motor}(s) = \frac{\theta(s)}{e(s)} = \frac{K}{s(\tau_m s + 1)}$$

### **6.2.2. Implementación en EJS**

Para su implementación en EJS, primero se completa el apartado de Modelo.

En variables se añade una pestaña con las variables significativas del motor. Por ejemplo, el ángulo inicial del motor será cero por eso theta es igual a cero. (Figura 6-20).



Nombre	Valor inicial	Tipo
Km	0.021	double
Tm	0.276	double
w	0.0	double
theta	0.0	double
u	0.0	double
t	0	double
dt	0.0001	double
		double

Figura 6-20. Variables significativas

Otra pestaña con las variables que se utilizan para diseñar el control en posición. Por ejemplo, derror es la derivada del error. (Figura 6-21)



Nombre	Valor inicial	Tipo
error	0.0	double
derror	0.0	double
ierror	0.0	double
thetaref	2.5	double
kp	1	double
td	0	double
Kd	0	double
ti	0.0	double
tpass	0.0	double
elapsedt	0.0	double
lasterror	0.0	double

Figura 6-21. Variables del control

Otra pestaña con las variables que se utilizan para recoger las puntuaciones obtenidas en los apartados. Por ejemplo, finalpoints es el total de puntos recogidos en todos los apartados por el alumno (Figura 6-22)

○ Descripción ● <b>Modelo</b> ○ HtmlView		
● <b>Variables</b> ○ Inicialización ○ Evolución ○ Relaciones fijas ○ Propio ○ Elementos		
Significativas Control Visual puntuacion english/spanish RTE		
Nombre	Valor inicial	Tipo
s	1	double
intento	0	double
intento2	0	double
espanol	true	boolean
p1OK	false	boolean
score	0	double
finalpoints	0	double
objetivo	2	double
score2a	0	double
score2b	0	double
score2	0	double
ts_objetivo	0	double
p2OK	false	boolean
N	0	int
points	0	int

Figura 6-22. Variables de puntuación

Otra pestaña con las variables que se utilizan para cambiar el texto del inglés al español y viceversa, ya que, se ha implementado la opción de cambio de idioma. (Figura 6-23)

○ Descripción ● <b>Modelo</b> ○ HtmlView		
● <b>Variables</b> ○ Inicialización ○ Evolución ○ Relaciones fijas ○ Propio ○ Elementos		
Significativas Control Visual puntuacion english/spanish RTE		
Nombre	Valor inicial	Tipo
axis	"Ejes"	String
clean_graph	"Borrar gráfica"	String
check	"Comprobar"	String
resp	"No ha respondido nada"	String
ap	"Apartado 1. Esquema de control P de posici..."	String
apartado	"Apartado 1"	String
color	"White"	String
go	"Ir a P2"	String
clean	"Limpiar"	String
title	"Control en posición del motor"	String
titleX	"Tiempo (ms) "	String
titleY	"θ (en grados) "	String
titlegraph	"ángulo de giro del motor"	String
puntuacion	"Puntos"	String
constantes	"Constantes"	String
user	"Usuario"	String
continua	"Continuar"	String

Figura 6-23. Variables para el cambio de idioma

A continuación, en evolución se añade las ecuaciones antes obtenidas del motor. Para ello, se utiliza la función de transferencia ( 24 ) volviendo como ecuación diferencial y despejando la aceleración angular.

$$\theta(s) * (s * (\tau_m s + 1)) = K * e(s) \quad (38)$$

$$\theta(s) * \tau_m * s^2 + \theta(s) * s = K * e(s) \quad (39)$$

$$\tau_m * \frac{d\omega(t)}{dt} + \omega(t) = K * e(t) \quad (40)$$

$$\frac{d\omega(t)}{dt} = \frac{K * e(t) - \omega(t)}{\tau_m} \quad (41)$$

Esta ecuación ( 41 ) es la que se añade a la página de código de evolución.(Figura 6-24)



Figura 6-24. Evolución del motor

En la siguiente página de código de evolución se implementa el control PD que controla la posición del motor de la simulación. Se tiene en cuenta que el error va a ser entre el ángulo de referencia (thetaref) y el ángulo actual del motor (theta). Se puede ver como quedaría en la Figura 6-25.

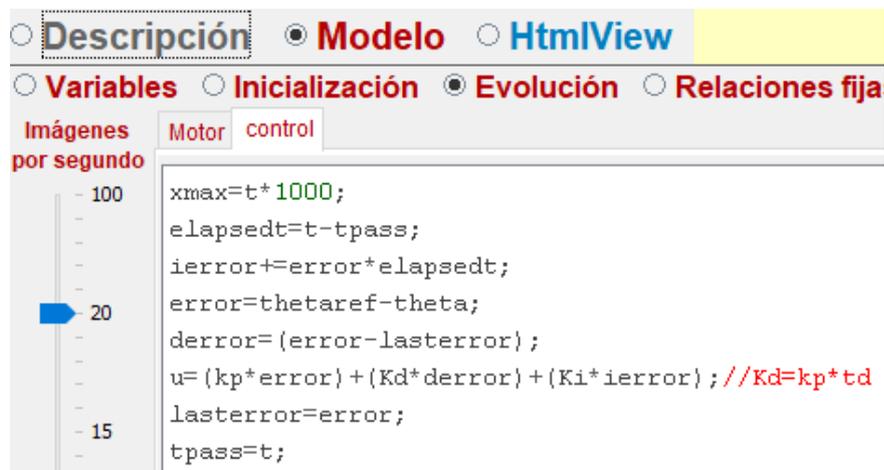


Figura 6-25. Evolución del control práctica 1

En el apartado de relaciones fijas, se realiza el diseño del código que permite el cambio de texto de un idioma a otro, según el elegido por el alumno. (Figura 6-26 y Figura 6-27) y se substraee cual es el N del alumno para elegir un tipo de motor y un tiempo de establecimiento como objetivo. (Figura 6-28)

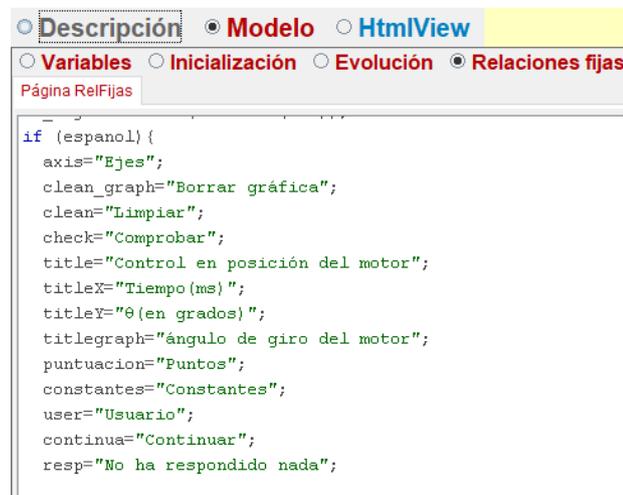


```

else {
    axis="Axis";
    clean_graph="Clean graph";
    clean="Clean";
    check="Check";
    title="Engine position control";
    titleX="Time (ms) ";
    titleY="θ (degrees) ";
    titlegraph="Engine turning angle";
    puntuacion="Score";
    constantes="Constants";
    user="User";
    continua="Continue";
    resp="You haven't answered anything yet";
}

```

Figura 6-26. Relaciones fijas de la simulación1. Inglés



```

if (español) {
    axis="Ejes";
    clean_graph="Borrar gráfica";
    clean="Limpiar";
    check="Comprobar";
    title="Control en posición del motor";
    titleX="Tiempo (ms) ";
    titleY="θ (en grados) ";
    titlegraph="ángulo de giro del motor";
    puntuacion="Puntos";
    constantes="Constantes";
    user="Usuario";
    continua="Continuar";
    resp="No ha respondido nada";
}

```

Figura 6-27. Relaciones fijas simulación 1. Español

```

N=learnerId.substr(learnerId.length-1,1);
if (N%2==0) {
    objetivo=2;
} else {
    objetivo=5;
}
Km=0.021*(1+0.035*(N-4));
Tm=0.276*(1+0.035*(N-4));
ts_objetivo=40*(1+0.035*(N-4));

```

Figura 6-28. Relaciones fijas simulación 1. N

En el apartado de HTMLView se implementa la vista de la simulación se puede observar que se ha generado un primer panel donde se muestra el título de la simulación, después un segundo panel que comparte la imagen del motor con la gráfica, donde se muestra la posición actual del motor y la referencia que se quiere alcanzar y, finalmente, los paneles de los botones para iniciar y parar la simulación,

limpiar la gráfica, resetear, mostrar datos importantes, el enunciado de los apartados etc.(Figura 6-29)

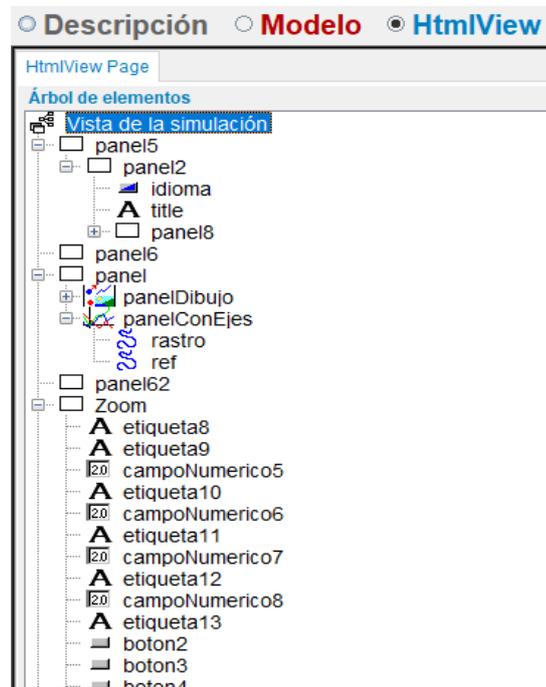
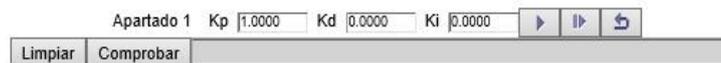


Figura 6-29. HTML View simulación 1

Finalmente, la vista previa quedaría como la Figura 6-30 .



Apartado 1. ¿Qué controlador es necesario sintonizar para que se encuentre el sistema críticamente amortiguado?



Nota:  $K_d = K_p \cdot T_d$ ;  $K_i = K_p / T_i$

Figura 6-30. Interfaz del apartado 1 de la simulación 1

### 6.2.3. Solución de la práctica.

En el apartado 1 se pide sintonizar el tipo de controlador que es necesario para que sea el sistema críticamente amortiguado.

Se sabe la función de transferencia del motor tiene un polo en el origen y otro en  $-\frac{1}{\tau_m}$ . Añadiendo un controlador proporcional, cuya ganancia proporcional se puede modelar, se consigue que el sistema sea críticamente amortiguado.

$$G_{motor}(s) = \frac{K}{s(s\tau_m + 1)} \quad (42)$$

$$G_{controlador}(s) = K_p \quad (43)$$

Quedando el lazo de control como en la Figura 6-31.

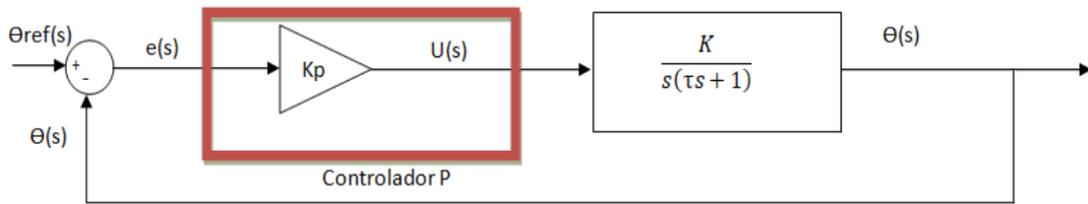


Figura 6-31. Lazo de control apartado 1

Haciendo la función en bucle abierto.

$$G_{ba}(s) = Kp * \frac{\frac{K}{\tau_m}}{s \left( s + \frac{1}{\tau_m} \right)} \quad (44)$$

En la

Figura 6-32 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$  y  $\tau_m=0.002$

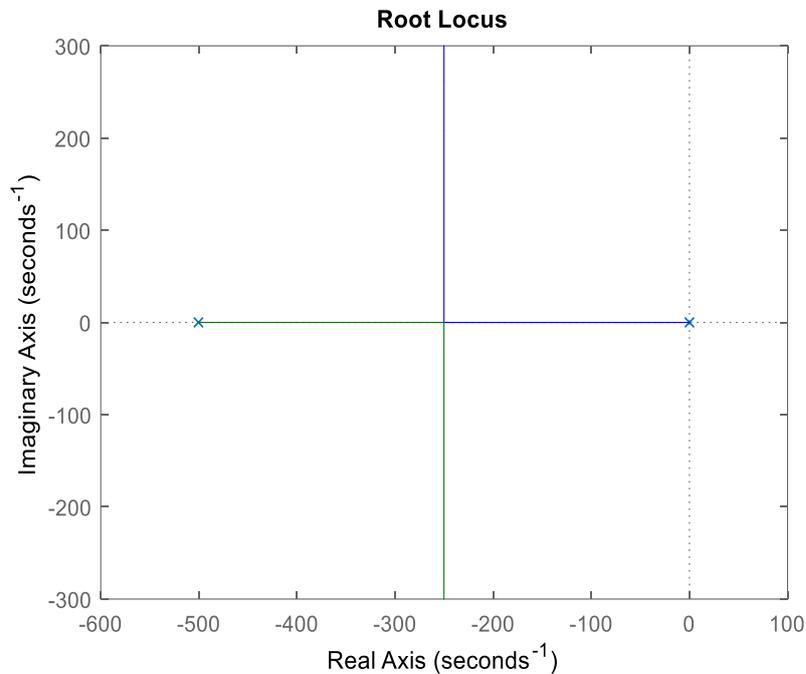


Figura 6-32. Lugar de las raíces  $G_{ba}$  (práctica 1, apartado 1)

Y continuando con la función de bucle cerrado.

$$G_{bc}(s) = \frac{G_{ba}(s)}{1 + G_{ba}(s)} = \frac{\frac{Kp * K}{\tau_m}}{s^2 + \frac{1}{\tau_m}s + \frac{Kp * K}{\tau_m}} \quad (45)$$

Si se quiere cumplir la especificación de  $\delta=1$

$$G_{bc}(s) = \frac{\frac{Kp * K}{\tau_m}}{s^2 + \frac{1}{\tau_m}s + \frac{Kp * K}{\tau_m}} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \quad (46)$$

Igualando los términos segundo y tercero.

$$2\delta\omega_n = \frac{1}{\tau_m}; \quad \omega_n = \frac{1}{2\tau_m} \quad (47)$$

$$\frac{Kp * K}{\tau_m} = \omega_n^2; \quad \frac{Kp * K}{\tau_m} = \left(\frac{1}{2\tau_m}\right)^2; \quad Kp * K = \frac{1}{4\tau_m} \quad (48)$$

Despejando  $Kp$ ,

$$Kp = \frac{1}{4K\tau_m} \quad (49)$$

En el apartado 2 se pide sintonizar el controlador para que se cumpla un tiempo de establecimiento  $t_{s5}$  con el criterio del 5%, para ello, se necesita adelantar la respuesta del sistema. Será necesario, a parte de la acción proporcional, también la acción derivativa que contribuye a adelantar la respuesta del sistema.

$$G_{motor}(s) = \frac{K}{s(s\tau_m + 1)} \quad (50)$$

$$G_{controlador}(s) = Kp * Td * \left(s + \frac{1}{Td}\right) \quad (51)$$

Quedando el lazo de control como en la Figura 6-33.

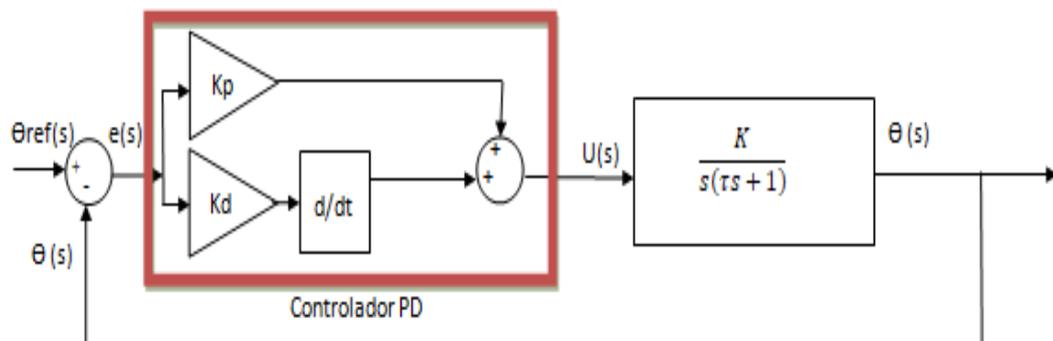


Figura 6-33. Lazo de control apartado 2

Haciendo la función en bucle abierto.

$$G_{ba}(s) = Kp * Td * \frac{K}{\tau_m} * \frac{\left(s + \frac{1}{Td}\right)}{s \left(s + \frac{1}{\tau_m}\right)} \quad (52)$$

Utilizando el criterio de cancelación cero/polo se supone que  $Td = \tau_m$

$$G_{ba}(s) = \frac{Kp * Td * \frac{K}{\tau_m}}{s} \quad (53)$$

En la Figura 6-34 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$ ,  $\tau_m=0.002$  y  $ts_5=0.2s$

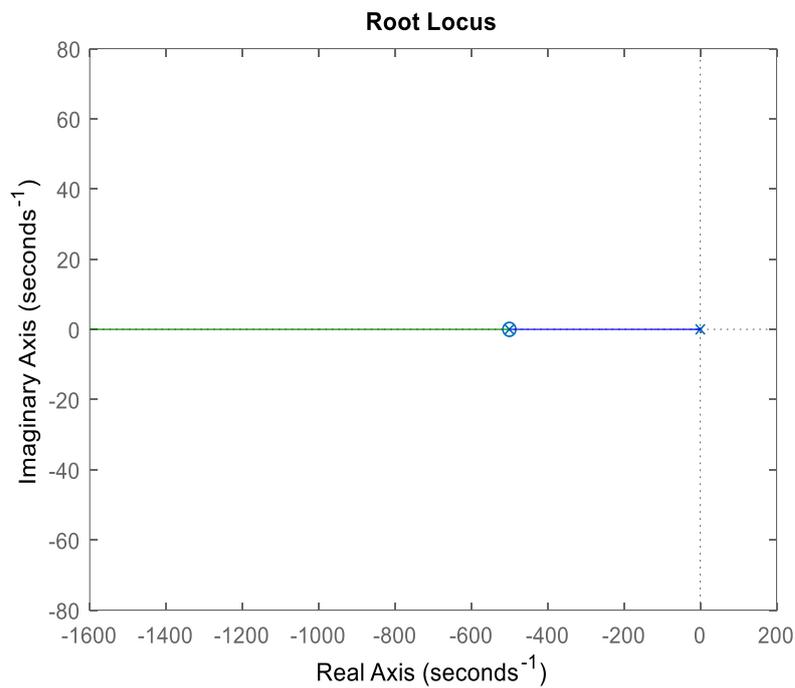


Figura 6-34. Lugar de las raíces  $G_{ba}$  (práctica 1, apartado 2)

Y continuando con la función de bucle cerrado.

$$G_{bc}(s) = \frac{G_{ba}(s)}{1 + G_{ba}(s)} = \frac{Kp * K}{s + Kp * K} \quad (54)$$

$$G_{bc}(s) = \frac{Kp * K}{s + Kp * K} = \frac{\frac{1}{\tau}}{s + \frac{1}{\tau}} \quad (55)$$

Igualando se obtiene,

$$\tau = \frac{1}{Kp * K} \quad (56)$$

Si la especificación es  $ts_5$ .

$$ts_5 = 3\tau; \tau = \frac{ts_5}{3} \quad (57)$$

$$\frac{ts_5}{3} = \frac{1}{Kp * K} \quad (58)$$

Despejando  $Kp$ ,

$$Kp = \frac{3}{ts_5 * K} \quad (59)$$

Y si  $Kd = Kp * Td$ , se obtiene

$$Kd = Kp * Td = Kp * \tau_m = \frac{3 * \tau_m}{ts_5 * K} \quad (60)$$

### 6.3. Control en velocidad del motor

**Enunciado.**

Se desea realizar el control en velocidad de un motor en CC cuyas constantes son K y  $\tau_m$ , estas constantes irán cambiando en función del número de usuario del alumno, N.

Apartado 1. ¿Qué tipo de controlador se necesita para un error en régimen permanente ante una referencia escalón unitario menor de  $\epsilon_{sp}$ ?

Apartado 2. ¿Qué tipo de controlador se debe sintonizar para un error en régimen permanente nulo y un tiempo de establecimiento  $t_{s5}$  con la condición del 5%?

#### 6.3.1. Modelado matemático.

Para la función de transferencia del motor se realiza los mismos pasos que en el apartado 4.1 y, por tanto, se utiliza la ecuación ( 21 )

$$G_{motor}(s) = \frac{\omega(s)}{e(s)} = \frac{K}{(\tau_m s + 1)}$$

#### 6.3.2. Implementación en EJS

Para su implementación en EJS, primero se completa el apartado de Modelo.

En variables se añade una pestaña con las variables significativas del motor. Por ejemplo, u es la tensión de entrada del motor que comienza siendo cero porque inicialmente no tiene tensión de entrada. (Figura 6-35)

<input type="radio"/> Descripción <input checked="" type="radio"/> <b>Modelo</b> <input type="radio"/> HtmlView		
<input checked="" type="radio"/> <b>Variables</b> <input type="radio"/> Inicialización <input type="radio"/> Evolución <input type="radio"/> Relaciones		
Significativas <input type="radio"/> Control <input type="radio"/> Visual <input type="radio"/> puntuacion <input type="radio"/> english/spanish <input type="radio"/> RTE		
Nombre	Valor inicial	Tipo
Km	0.021	double
Tm	0.276	double
w	0.0	double
theta	0.0	double
R	0.0	double
t	0	double
dt	0.0001	double
thetaref	2.5	double
		double

Figura 6-35. Variables significativas

Otra pestaña con las variables que se utilizan para diseñar el control en velocidad. Por ejemplo, la variable ierror es la integral del error. (Figura 6-36)

<input type="radio"/> Descripción <input checked="" type="radio"/> <b>Modelo</b> <input type="radio"/> HtmlView		
<input checked="" type="radio"/> <b>Variables</b> <input type="radio"/> Inicialización <input type="radio"/> Evolución <input type="radio"/> Relaciones fijas <input type="radio"/> Propio <input type="radio"/> Elementos		
Significativas <input type="radio"/> Control <input type="radio"/> Visual <input type="radio"/> puntuacion <input type="radio"/> english/spanish <input type="radio"/> RTE		
Nombre	Valor inicial	Tipo
error	0.0	double
derror	0.0	double
ierror	0.0	double
wref	2	double
kp	1	double
td	0	double
Ki	0	double
ti	0.0	double
tpass	0.0	double
elapsedt	0.0	double
lasterror	0.0	double

Figura 6-36. Variables del control

Otra pestaña con las variables que se utilizan para recoger las puntuaciones obtenidas en los apartados. (Figura 6-37)

○ Descripción ● <b>Modelo</b> ○ HtmlView		
● <b>Variables</b> ○ Inicialización ○ Evolución ○ Relaciones fijas ○ Propio ○ Elementos		
Significativas Control Visual puntuacion english/spanish RTE		
Nombre	Valor inicial	Tipo
s	1	double
intento	0	double
intento2	0	double
espanol	true	boolean
p1OK	false	boolean
score	0	double
finalpoints	0	double
objetivo	2	double
score2a	0	double
score2b	0	double
score2	0	double
ts_objetivo	0	double
p2OK	false	boolean
N	0	int
points	0	int

Figura 6-37. Variables de puntuación

Otra pestaña con las variables que se utilizan para cambiar el texto del inglés al español y viceversa, ya que, se ha implementado la opción de cambio de idioma. (Figura 6-38)

○ Descripción ● <b>Modelo</b> ○ HtmlView		
● <b>Variables</b> ○ Inicialización ○ Evolución ○ Relaciones fijas ○ Propio ○ Elementos		
Significativas Control Visual puntuacion english/spanish RTE		
Nombre	Valor inicial	Tipo
axis	"Ejes"	String
clean_graph	"Borrar gráfica"	String
check	"Comprobar"	String
resp	"No ha respondido nada"	String
ap	"Apartado 1. Esquema de control P de posici..."	String
apartado	"Apartado 1"	String
color	"White"	String
go	"Ir a P2"	String
clean	"Limpiar"	String
title	"Control en posición del motor"	String
titleX	"Tiempo (ms)"	String
titleY	"θ (en grados)"	String
titlegraph	"ángulo de giro del motor"	String
puntuacion	"Puntos"	String
constantes	"Constantes"	String
user	"Usuario"	String
continua	"Continuar"	String

Figura 6-38. Variables para el cambio de idioma

A continuación, en evolución se añade las ecuaciones antes obtenidas del motor. Para ello, se utiliza la función de transferencia ( 21 ), volviendo como ecuación diferencial y despejando la aceleración angular.

$$\frac{\omega(s)}{e(s)} = \frac{K}{(\tau_m s + 1)} \quad (61)$$

$$\omega(s) * \tau_m * s + \omega(s) = K * e(s) \quad (62)$$

$$\tau_m * \frac{d\omega(t)}{dt} + \omega(t) = K * e(t) \quad (63)$$

$$\frac{d\omega(t)}{dt} = \frac{K * e(t) - \omega(t)}{\tau_m} \quad (64)$$

Esta ecuación ( 64 ) es la que se añade a la página de código de evolución.(Figura 6-39)



Figura 6-39. Evolución del motor

En la siguiente página de código de evolución se implementa el control PD que controla la velocidad del motor de la simulación. Se tiene en cuenta que el error va a ser entre el velocidad angular de referencia (wref) y el velocidad angular actual del motor (w). Se puede ver como quedaría en la Figura 6-40.

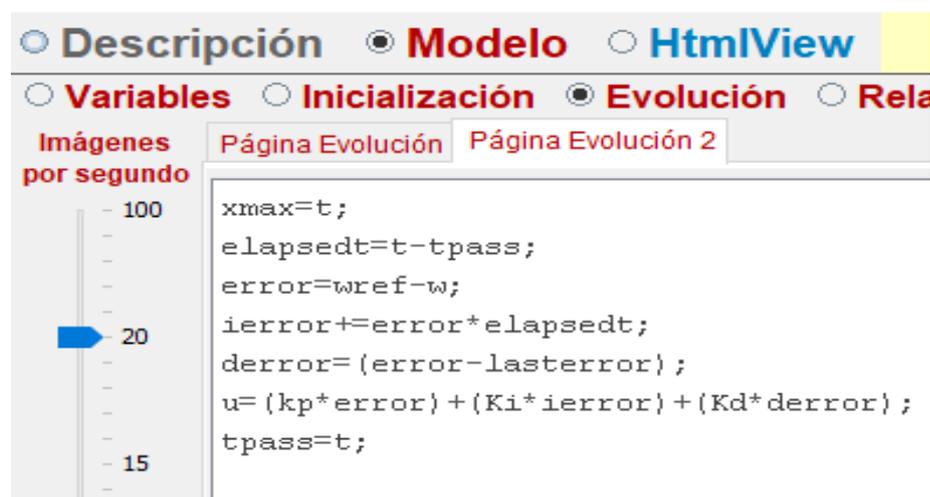
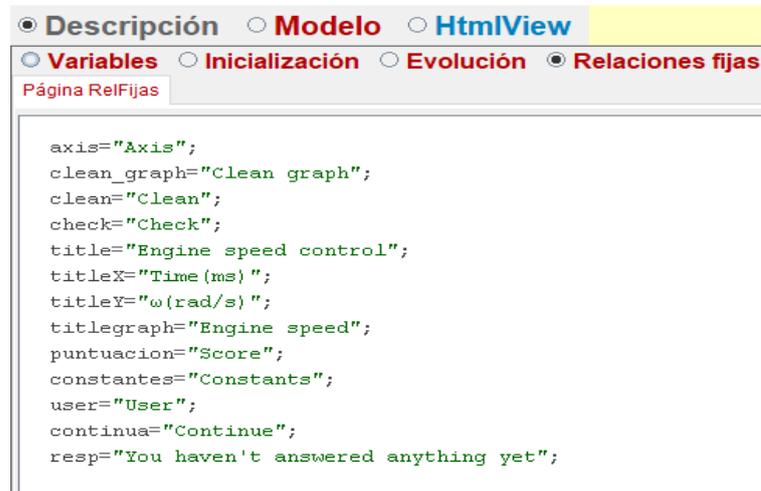


Figura 6-40. Evolución del control práctica 2

En el apartado de relaciones fijas, se realiza el diseño del código que permite el cambio de texto de un idioma a otro, según el elegido por el alumno. (Figura 6-41 y Figura 6-42) Y se subtrae cual es el N del alumno para elegir un tipo de motor y un

tiempo de establecimiento como objetivo y error en régimen permanente. (Figura 6-43)

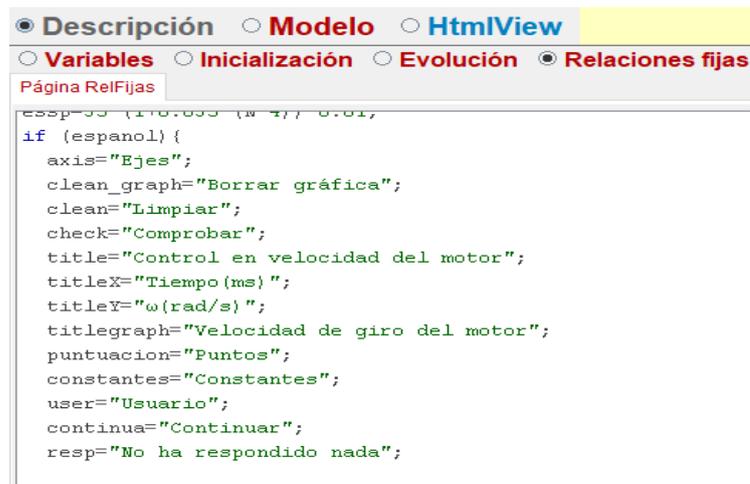


```

axis="Axis";
clean_graph="Clean graph";
clean="Clean";
check="Check";
title="Engine speed control";
titleX="Time (ms) ";
titleY="ω(rad/s) ";
titlegraph="Engine speed";
puntuacion="Score";
constantes="Constants";
user="User";
continua="Continue";
resp="You haven't answered anything yet";

```

Figura 6-41. Relaciones fijas de la simulación2. Inglés



```

esp=35 (1+0.035 (N-4)) *0.01;
if (espanol){
axis="Ejes";
clean_graph="Borrar gráfica";
clean="Limpiar";
check="Comprobar";
title="Control en velocidad del motor";
titleX="Tiempo (ms) ";
titleY="ω(rad/s) ";
titlegraph="Velocidad de giro del motor";
puntuacion="Puntos";
constantes="Constantes";
user="Usuario";
continua="Continuar";
resp="No ha respondido nada";

```

Figura 6-42. Relaciones fijas simulación 2. Español

```

N=learnerId.substr(learnerId.length-1,1);
if(N%2==0){
objetivo=2;
}else{
objetivo=5;
}
Km=0.021*(1+0.035*(N-4));
Tm=0.276*(1+0.035*(N-4));
ts_objetivo=5*(1+0.035*(N-4));
esp=35*(1+0.035*(N-4))*0.01;

```

Figura 6-43. Relaciones fijas simulación 2. N

En el apartado de HTMLView se implementa la vista de la simulación se puede observar que se ha generado un primer panel donde se muestra el título de la simulación, después un segundo panel que comparte la imagen del motor con la gráfica, donde se muestra la velocidad actual del motor y la referencia que se quiere alcanzar y, finalmente, los paneles de los botones para iniciar y parar la simulación, limpiar la gráfica, resetear, mostrar datos importantes, el enunciado de los apartados etc.(Figura 6-44)

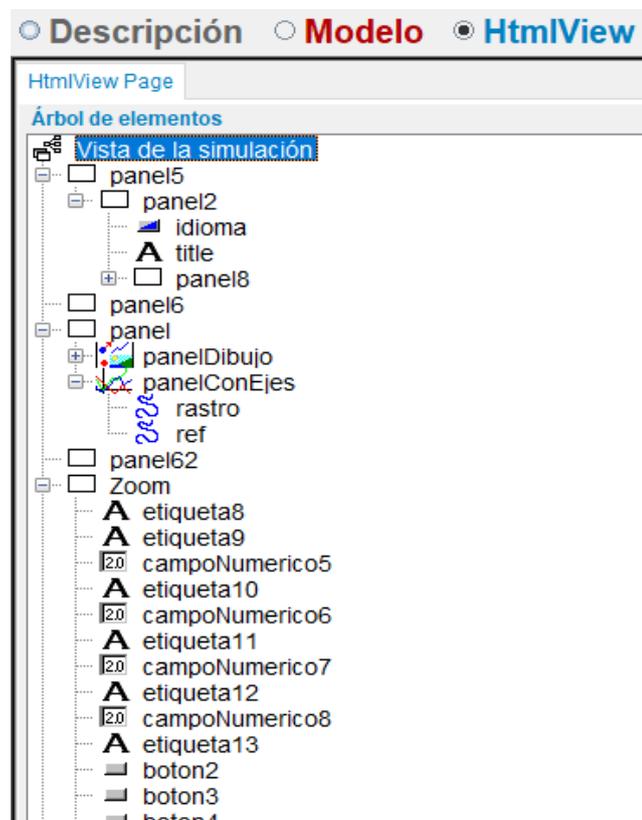


Figura 6-44. HTML View simulación 2

Finalmente, la vista previa quedaría como la Figura 6-45 .

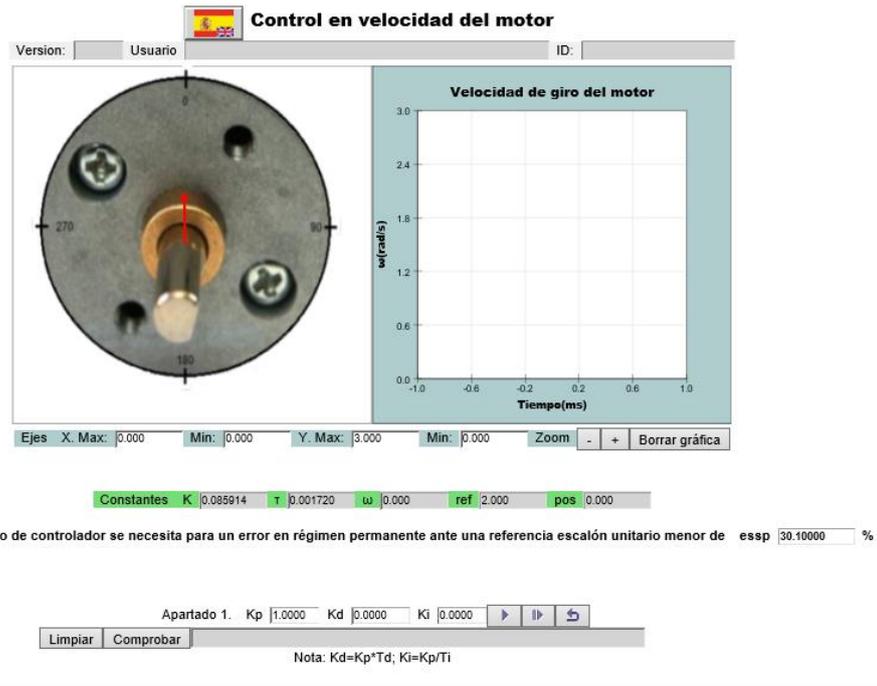


Figura 6-45. Interfaz del apartado 2 de la simulación 2

### 6.3.3. Solución de la práctica.

En el apartado 1 se pide sintonizar que controlador se necesita para un error en régimen permanente ante una referencia escalón unitario menor de  $\text{essp}$ .

Se sabe que para tener un error en régimen permanente y sin ninguna especificación de tiempo de establecimiento solo sería necesario con un controlador proporcional.

A continuación, la función de transferencia del motor y la función del controlador proporcional.

$$G_{motor}(s) = \frac{K}{(s\tau_m + 1)} \quad (65)$$

$$G_{controlador}(s) = Kp \quad (66)$$

$$R(s) = \frac{1}{s} \quad (67)$$

Quedando el lazo de control como en la Figura 6-46.

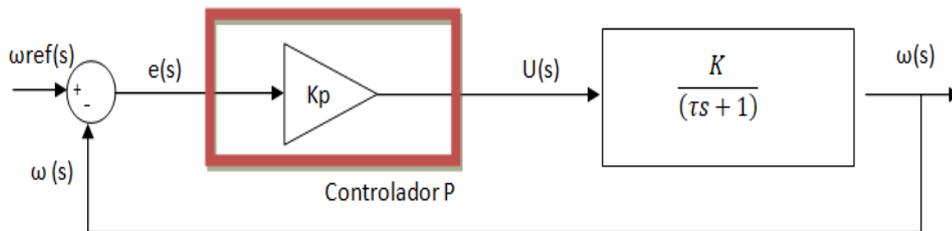


Figura 6-46. Lazo de control apartado 1

Haciendo la función en bucle abierto.

$$G_{ba}(s) = Kp * \frac{\frac{K}{\tau_m}}{\left(s + \frac{1}{\tau_m}\right)} \quad (68)$$

En la Figura 6-47 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$ ,  $\tau_m=0.002$  y  $essp=35\%$

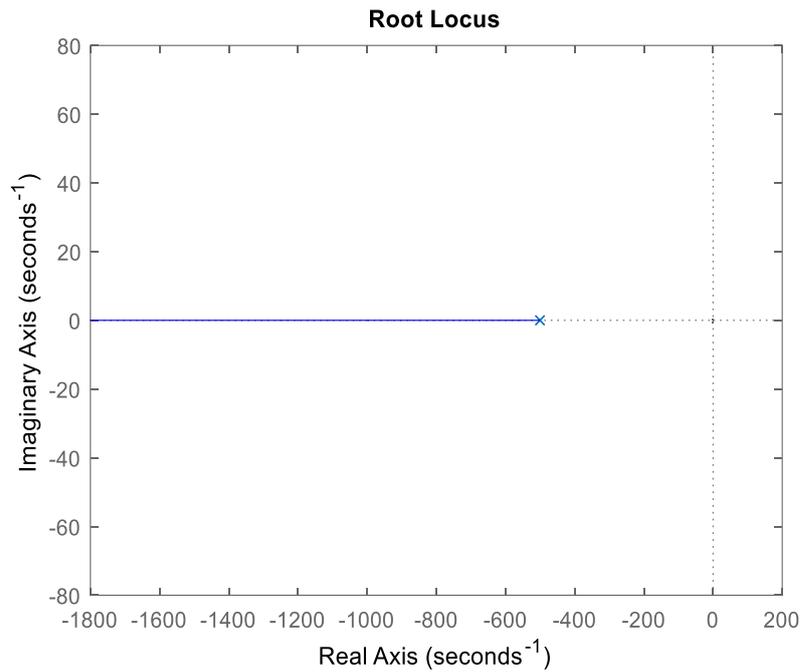


Figura 6-47. Lugar de las raíces  $G_{ba}$  (práctica 2, apartado 1)

Sabiendo que

$$essp = \lim_{s \rightarrow 0} s * R(s) * \frac{1}{1 + Gba} \quad (69)$$

$$essp = \lim_{s \rightarrow 0} s * \frac{1}{s} * \frac{s + \frac{1}{\tau_m}}{s + \frac{1}{\tau_m} + \frac{Kp * K}{\tau_m}} = \frac{\frac{1}{\tau_m}}{\frac{1}{\tau_m} + \frac{Kp * K}{\tau_m}} \quad (70)$$

$$essp = \frac{\frac{1}{\tau_m}}{\frac{1}{\tau_m} + \frac{Kp * K}{\tau_m}}; \quad 1 - essp = essp * Kp * K \quad (71)$$

Despejando  $Kp$ ,

$$Kp = \frac{1 - essp}{essp * K} \quad (72)$$

En el apartado 2 se pide sintonizar un controlador para que el error en régimen permanente sea nulo y un tiempo de establecimiento  $t_{s5}$  con la condición del 5%. Para cumplir que el sistema tenga un error en régimen permanente nulo, el controlador deberá ser un controlador proporcional-integral y se utilizará la condición del tiempo para definir la ganancia.

$$G_{motor}(s) = \frac{K}{s(s\tau_m + 1)} \quad (73)$$

$$G_{controlador}(s) = \frac{Kp\left(s + \frac{1}{Ti}\right)}{s} \quad (74)$$

Quedando el lazo de control como en la

Figura 6-48.

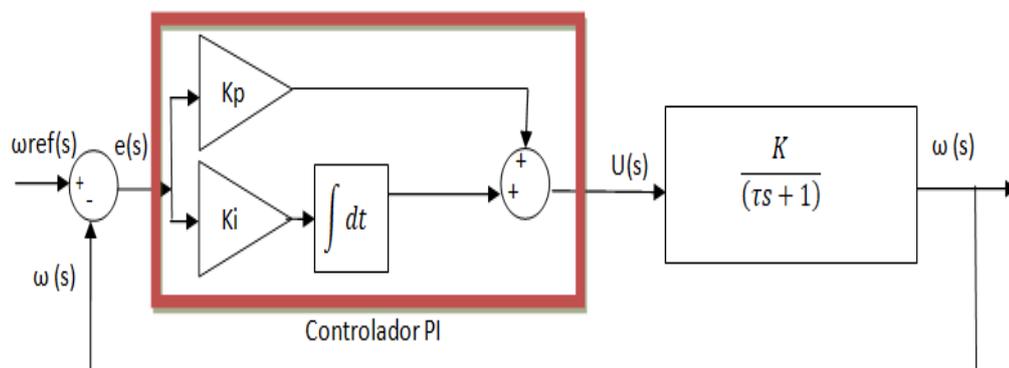


Figura 6-48. Lazo de control apartado 2

Haciendo la función en bucle abierto.

$$G_{ba}(s) = \frac{Kp * K}{\tau_m} \left( s + \frac{1}{Ti} \right) \quad (75)$$

$$s \left( s + \frac{1}{\tau_m} \right)$$

Utilizando el criterio de cancelación cero/polo se supone que  $Ti = \tau_m$

$$G_{ba}(s) = \frac{K * Kp}{s} \quad (76)$$

En la Figura 6-49 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$ ,  $\tau_m=0.002$  y  $t_{s5}=0.2s$

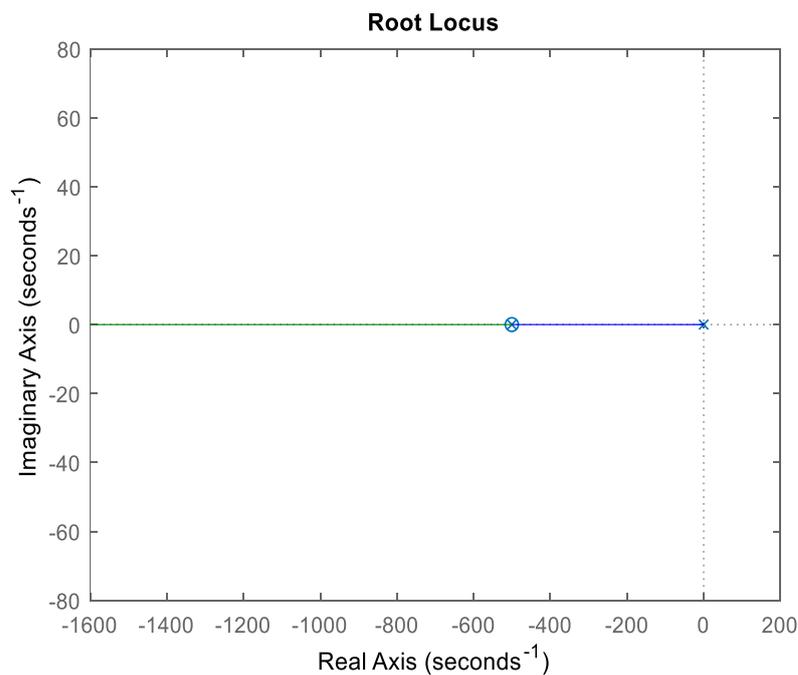


Figura 6-49. Lugar de las raíces  $G_{ba}$  (práctica 2, apartado 2)

Y continuando con la función de bucle cerrado.

$$G_{bc}(s) = \frac{G_{ba}(s)}{1 + G_{ba}(s)} = \frac{\frac{Kp * K}{\tau_m}}{s + \frac{Kp * K}{\tau_m}} \quad (77)$$

$$G_{bc}(s) = \frac{\frac{Kp * K}{\tau_m}}{s + \frac{Kp * K}{\tau_m}} = \frac{\frac{1}{\tau}}{s + \frac{1}{\tau}} \quad (78)$$

Igualando se obtiene,

$$\tau = \frac{\tau_m}{Kp * K} \quad (79)$$

Si la especificación es  $ts_5$ .

$$ts_5 = 3\tau; \tau = \frac{ts_5}{3} \quad (80)$$

$$\frac{ts_5}{3} = \frac{\tau_m}{Kp * K} \quad (81)$$

Despejando  $Kp$ ,

$$Kp = \frac{3\tau_m}{ts_5 * K} \quad (82)$$

Y si  $Ki$  se obtiene como

$$K_i = \frac{K_p}{T_i} = \frac{K_p}{\tau_m} = \frac{3\tau_m}{t_{s5} * K * \tau_m} = \frac{3}{t_{s5} * K} \quad (83)$$

## 6.4. Control del sistema barra-bola

### **Enunciado.**

Se desea realizar el control en cascada del sistema barra-bola. Este control en cascada consta de un controlador maestro,  $G_{c2}(s)$ , cuya entrada es la posición de referencia de la bola y su salida es el punto de consigna (ángulo de referencia del motor) para el controlador esclavo,  $G_{c1}(s)$ . Diseñar el controlador maestro ( $G_{c2}(s)$ ) y el controlador esclavo ( $G_{c1}(s)$ ) para que el sistema sea críticamente amortiguado y con un tiempo de establecimiento  $t_{s5}$  con la condición del 5%.

### 6.4.1. Modelado matemático.

Para la función de transferencia del motor se realiza los mismos pasos que en el apartado 4.1 y, por tanto, se utiliza la ecuación ( 24 )

$$G_{motor}(s) = \frac{\theta(s)}{e(s)} = \frac{K}{s(\tau_m s + 1)}$$

Y el modelado de la barra y la bola se realiza como en el apartado 3 que como resultado se consigue la ecuación ( 17 )

$$-\frac{2}{5} * m * a(t) - 6 * \eta * \pi * R * v_x(t) + m * g * \alpha(t) = m * a(t)$$

Se realiza los pasos del apartado 4.2 y se alcanza la función de transferencia de la barra y la bola. (Ecuación( 27 ))

$$G_{\text{BarraBola}}(s) = \frac{x(s)}{\alpha(s)} = \frac{\frac{5g}{7}}{s\left(s + \frac{5*6*\eta*\pi*R}{7*m}\right)}$$

Además, se tiene en cuenta que  $H(s) = \frac{d}{L}$

#### 6.4.2. Implementación en EJS

Para su implementación en EJS, primero se completa el apartado de Modelo.

En variables se añade una pestaña con las variables significativas del sistema. Por ejemplo, x es la posición inicial de la bola que se encuentra a -0.25 m del eje central. (Figura 6-50)



Nombre	Valor inicial	Tipo
Alfa	0.00	double
xref	0	double
x	-0.25	double
y	$x * \text{Math.tan}(\text{Alfa})$	double
vx	0.0	double
t	0.0	double
dt	0.05	double
theta	0	double
thetaref	0	double
u	0	double
w	0	double
vxl	0	double
xl	-0.25	double

Figura 6-50. Variables significativas

Otra pestaña con las constantes que se utilizan. Por ejemplo, g es la constante de la gravedad, cuyo valor es 9.81 m/s<sup>2</sup> (Figura 6-51)

<input checked="" type="radio"/> Descripción <input 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		
Constantes         Variables significativas         Variables motor         puntuacion         visual         Controlador         english/spanish         RTE		
Nombre	Valor inicial	Tipo
g	9.81	double
R	0.025	double
L	0.6	double
Dc	R*3	double
d	0.022	double
n	0.83	double
fact	1/2	double
cract	1	double
Croz	true	boolean
pe	7750	double
m	0.05	double
		double

Figura 6-51. Constantes simulación 3

Otra pestaña con las constantes que definen el motor utilizado en el sistema. (Figura 6-52)

<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"/> Relacion		
Constantes         Variables significativas         Variables motor         puntuacion         visual		
Nombre	Valor inicial	Tipo
Km	0.021	double
Tm	0.276	double
		double

Figura 6-52. Variables del motor

Otra pestaña con las variables que se utilizan para diseñar el control. (Figura 6-53)

<input checked="" type="radio"/> Descripción <input 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		
Constantes         Variables significativas         Variables motor         puntuacion         visual         Controlador         english/spanish         RTE		
Nombre	Valor inicial	Tipo
kp	0	double
td	0	double
tdl	0	double
kpl	0	double
derrorl	0.0	double
errorl	0.0	double
lasterrorl	0.0	double
kd	0	double
kdl	0	double
error	0	double
derror	0	double
lasterror	0	double

Figura 6-53. Variables del control

Otra pestaña con las variables que se utilizan para recoger las puntuaciones obtenidas. (Figura 6-54)

Descripción		
Variables		
Nombre	Valor inicial	Tipo
s	1	double
intento	0	double
intento2	0	double
espanol	true	boolean
p1OK	false	boolean
score1	0	double
score1b	0	double
score1a	0	double
ts_objetivo	1	double
N	2	double
p2OK	false	boolean
score2	0	double
score2a	0	double
score2b	0	double
objetivo	2	double
finalpoints	0	double
points	0	int
		double

Figura 6-54. Variables de puntuación

Otra pestaña con las variables que se utilizan para cambiar el texto del inglés al español y viceversa, ya que, se ha implementado la opción de cambio de idioma. (Figura 6-55)

Descripción		
Variables		
Nombre	Valor inicial	Tipo
axis	"Ejes"	String
clean_graph	"Borrar grafica"	String
check	"Comprobar"	String
resp	"No ha respondido nada"	String
ap	"Apartado 1. Calcula Kp2 y Kd2 del controla..."	String
apartado	"Apartado 1"	String
color	"White"	String
go	"Ir a P2"	String
clean	"Limpiar"	String
title	"Control sistema barra-bola linealizado"	String
titleX	"Tiempo(ms) "	String
titleY	"Posicion de la bola (mm) "	String
titlegraph	"Posicion de la bola en la barra"	String
puntuacion	"Puntos"	String
constantes	"Constantes"	String
change	"Cambiar gráfica"	String
nota	"Nota"	String
user	"Usuario"	String
continua	"Continuar"	String

Figura 6-55. Variables para el cambio de idioma

A continuación, en evolución se añade las ecuaciones antes obtenidas del motor. Para ello, se utiliza la función de transferencia ( 24 ), volviendo como ecuación diferencial y despejando la aceleración angular.

$$\theta(s) * (s * (\tau_m s + 1)) = K * e(s) \quad (84)$$

$$\theta(s) * \tau_m * s^2 + \theta(s) * s = K * e(s) \quad (85)$$

$$\tau_m * \frac{d\omega(t)}{dt} + \omega(t) = K * e(t) \quad (86)$$

$$\frac{d\omega(t)}{dt} = \frac{K * e(t) - \omega(t)}{\tau_m} \quad (87)$$

Esta ecuación ( 87 ) es la que se añade a la página de código de evolución.(Figura 6-56)



Figura 6-56. Evolución del motor

En otra pestaña de evolución se añade las ecuaciones antes obtenidas de la barra y la bola. Para ello, se utiliza la función de transferencia ( 27 ),volviendo como ecuación diferencial y despejando la aceleración angular.

$$x(s) * s * \left( s + \frac{5 * 6 * \eta * \pi * R}{7 * m} \right) = \alpha(s) * \frac{5g}{7} \quad (88)$$

$$\frac{dv_x}{dt} + \frac{5 * 6 * \eta * \pi * R}{7 * m} * v_x = \alpha(t) * \frac{5g}{7} \quad (89)$$

$$\frac{dv_x}{dt} = \alpha(t) * \frac{5g}{7} - \frac{5 * 6 * \eta * \pi * R}{7 * m} * v_x \quad (90)$$

Esta ecuación ( 94 ) es la que se añade a la página de código de evolución.(Figura 6-57)

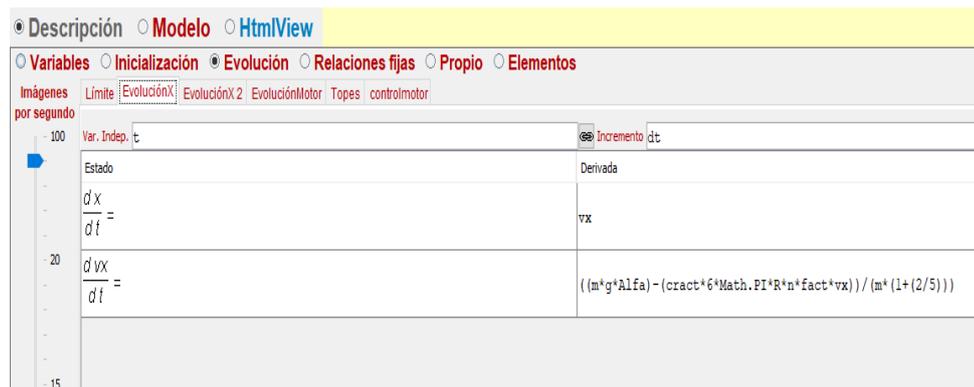


Figura 6-57. Evolución de la posición de la bola con linealizado

En esta práctica también se desea visualizar la evolución de la bola sobre la barra sin tener en cuenta la simplificación de seno ( $\alpha$ ) es igual a  $\alpha$ . En el siguiente pestaña se ejecutará el mismo código pero sin tener en cuenta esta simplificación. (Figura 6-58)

$$\frac{dv_x}{dt} = \sin(\alpha(t)) * \frac{5g}{7} - \frac{5 * 6 * \eta * \pi * R}{7 * m} * v_x \quad (91)$$

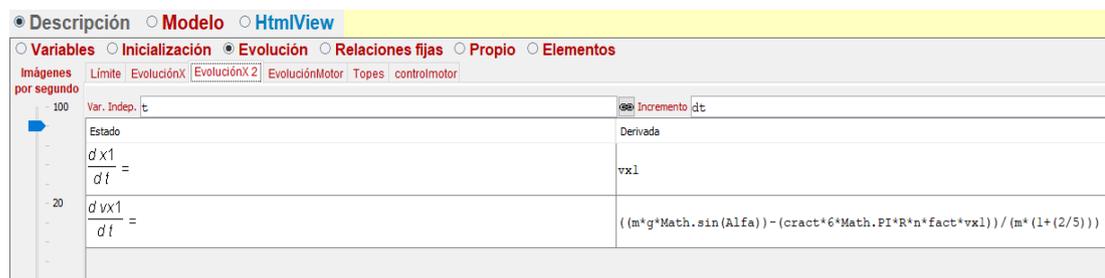
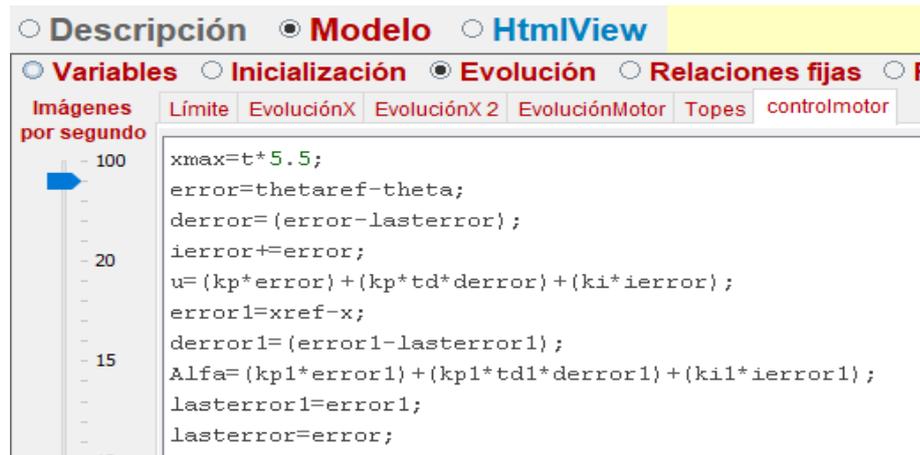


Figura 6-58. Evolución de la posición de la bola sin linealizado

En la siguiente página de código de evolución se implementa el control cascada con los dos controladores PD. Se puede ver como quedaría en la Figura 6-59.



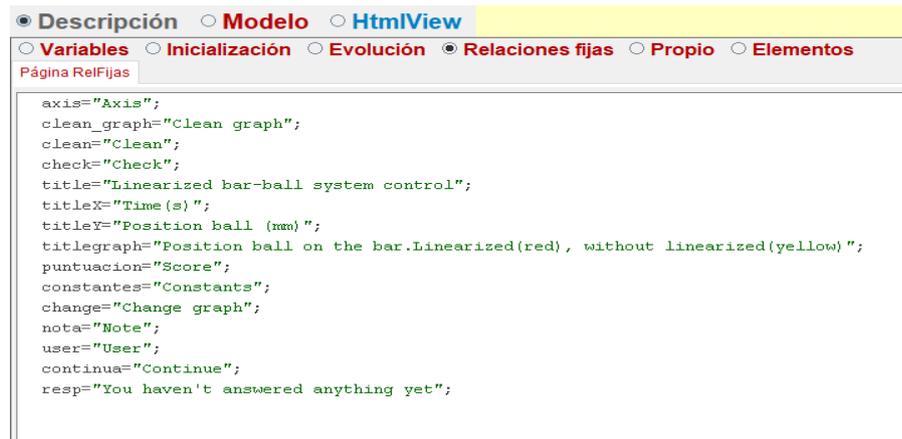
```

xmax=t*5.5;
error=thetaref-theta;
derror=(error-lasterror);
ierror+=error;
u=(kp*error)+(kp*td*derror)+(ki*ierror);
error1=xref-x;
derror1=(error1-lasterror1);
Alfa=(kp1*error1)+(kp1*td1*derror1)+(ki1*ierror1);
lasterror1=error1;
lasterror=error;

```

Figura 6-59. Evolución del control en cascada

En el apartado de relaciones fijas, se realiza el diseño del código que permite el cambio de texto de un idioma a otro, según el elegido por el alumno. (Figura 6-60 y Figura 6-61) y se substraen cual es el N del alumno para elegir un tipo de motor y un tiempo de establecimiento como objetivo. (Figura 6-62)

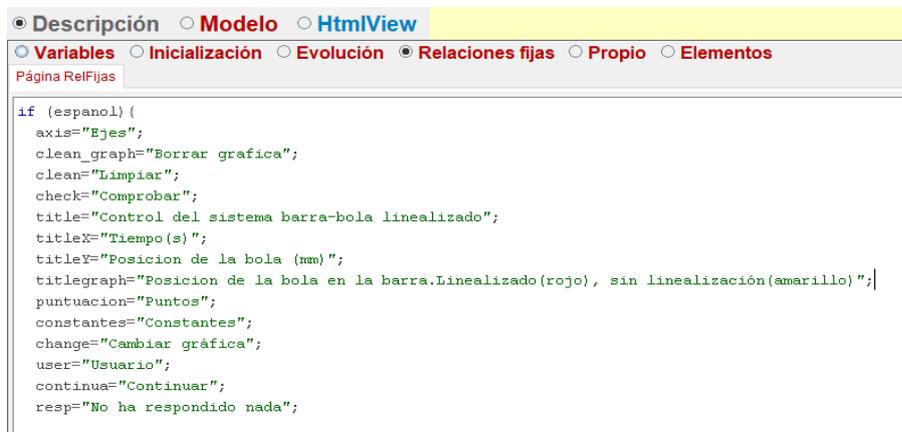


```

axis="Axis";
clean_graph="Clean graph";
clean="Clean";
check="Check";
title="Linearized bar-ball system control";
titleX="Time (s)";
titleY="Position ball (mm)";
titlegraph="Position ball on the bar.Linearized(red), without linearized(yellow)";
puntuacion="Score";
constantes="Constants";
change="Change graph";
nota="Note";
user="User";
continua="Continue";
resp="You haven't answered anything yet";

```

Figura 6-60. Relaciones fijas de la simulación3. Inglés



```

if (espanol){
axis="Ejes";
clean_graph="Borrar grafica";
clean="Limpiar";
check="Comprobar";
title="Control del sistema barra-bola linealizado";
titleX="Tiempo (s)";
titleY="Posicion de la bola (mm)";
titlegraph="Posicion de la bola en la barra.Linealizado(rojo), sin linealización(amarillo)";
puntuacion="Puntos";
constantes="Constantes";
change="Cambiar gráfica";
user="Usuario";
continua="Continuar";
resp="No ha respondido nada";
}

```

Figura 6-61. Relaciones fijas simulación 3. Español

```

N=learnerId.substr(learnerId.length-1,1);
thetaref=Alfa*L/d;
ts_objetivo=93.03*(1+0.035*(N-4));
Km=0.021*(1+0.035*(N-4));
Tm=0.276*(1+0.035*(N-4));

```

Figura 6-62. Relaciones fijas simulación 3. N

En este apartado se muestra la vista HTML de la simulación se puede observar que se ha generado un primer panel donde se muestra el título de la simulación, después un segundo panel donde comparte el dibujo 3D del sistema barra-bola con la gráfica donde se muestra la posición actual de la bola en la barra con aproximación y sin aproximación ( $\sin(\alpha)=\alpha$ ) y la referencia. Finalmente, los paneles de los botones para iniciar y parar la simulación, limpiar la gráfica, resetear, mostrar datos importantes, el enunciado del apartado etc.(Figura 6-63)

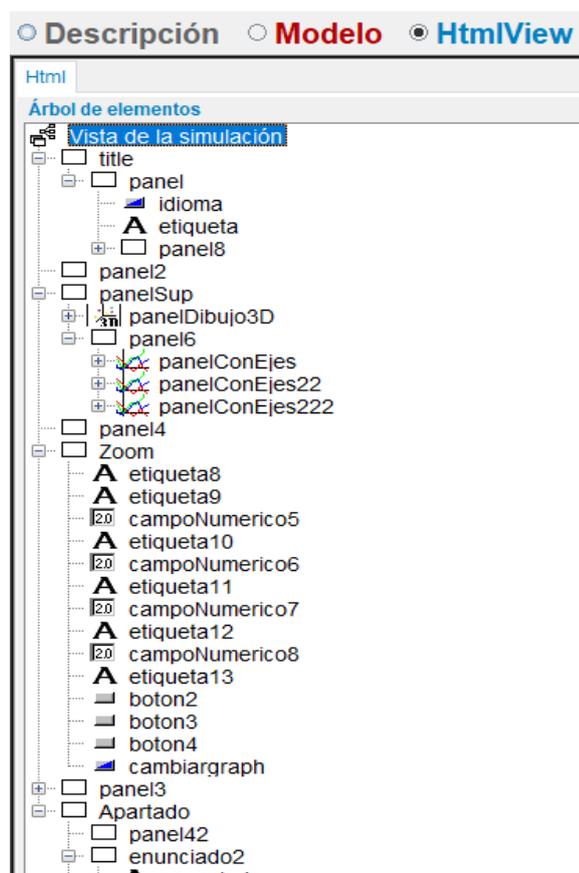


Figura 6-63. HTML View simulación 3

Finalmente, la vista previa quedaría como la Figura 6-64.

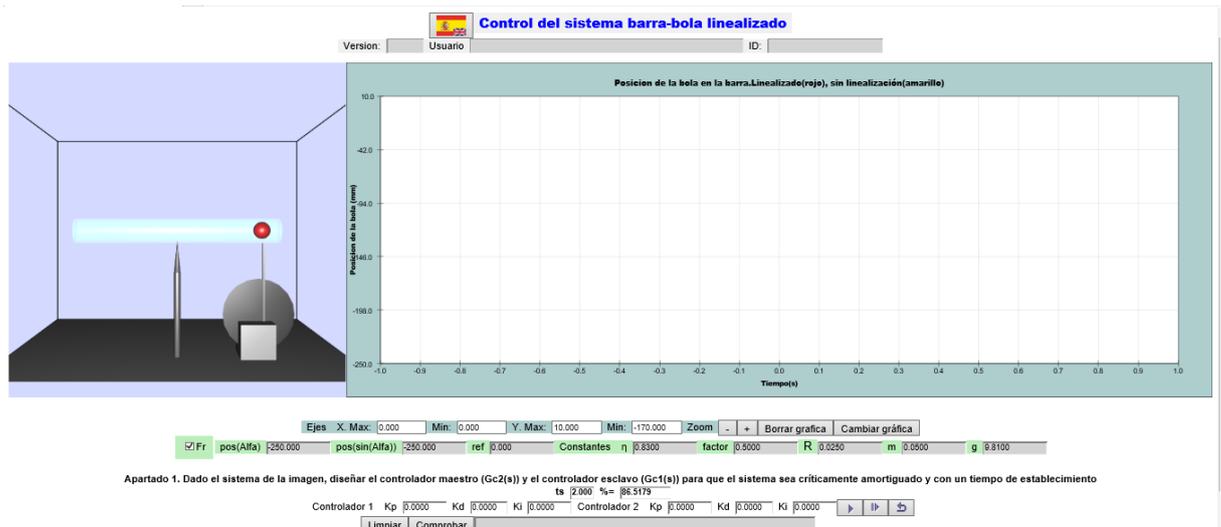


Figura 6-64. Vista previa de la simulación 3

### 6.4.3. Solución de la práctica.

En esta práctica, se pide diseñar el controlador maestro ( $G_{c2}(s)$ ) y el controlador esclavo ( $G_{c1}(s)$ ) para que el sistema sea críticamente amortiguado y con un tiempo de establecimiento  $t_{s5}$  con la condición del 5%, para que se cumpla estas especificaciones ambos controladores serán proporcionales-derivativo. De esta manera se ayuda a adelantar la velocidad de respuesta del lazo externo e interno y se consigue que el sistema sea críticamente amortiguado.

Se comienza realizando el controlador esclavo. Se sabe la función de transferencia del motor y el la función de transferencia de un controlador proporcional-derivativo.

$$G_{motor}(s) = \frac{K}{s(s\tau_m + 1)} \quad (92)$$

$$G_{controlador1}(s) = Kp1 * Td1 * \left(s + \frac{1}{Td1}\right) \quad (93)$$

Haciendo la función en bucle abierto.

$$G_{ba1}(s) = Kp1 * Td1 * \frac{K}{\tau_m} * \frac{s + \frac{1}{Td1}}{s \left( s + \frac{1}{\tau_m} \right)} \quad (94)$$

Utilizando el criterio de cancelación cero/polo se supone que  $Td1 = \tau_m$  así se cancela el polo más lejano con el cero que se añade la acción derivativa.

$$G_{ba1}(s) = \frac{K * Kp1}{s} \quad (95)$$

En la Figura 6-65 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$ ,  $\tau_m=0.002$ .

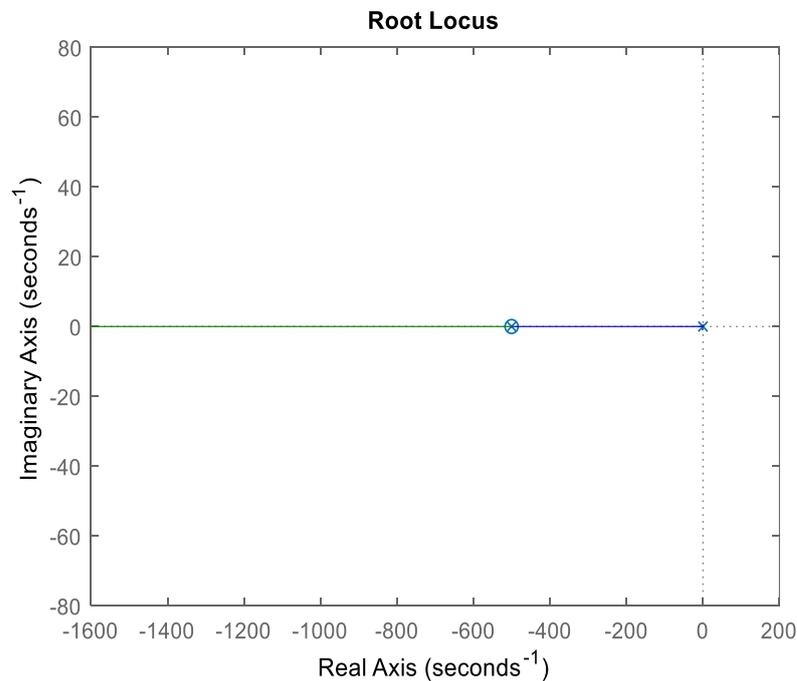


Figura 6-65. Lugar de las raíces de  $G_{ba1}$

Y continuando con la función de bucle cerrado, quedando controlado el motor

$$G_{bc1}(s) = \frac{G_{ba1}(s)}{1 + G_{ba1}(s)} = \frac{Kp * 1K}{s + Kp1 * K} \quad (96)$$

Ahora, se realiza el control del lazo más externo.

$$G_{ba2}(s) = G_{controlador2}(s) * G_{bc1}(s) * H(s) * G_{BarraBola}(s) \quad (97)$$

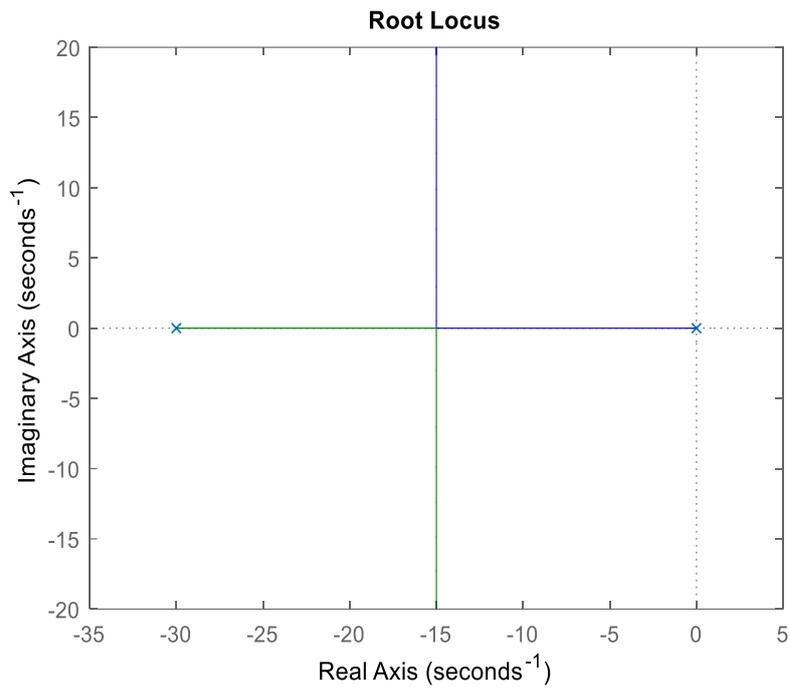
$$G_{ba2}(s) = Kp2 * Td2 * \left(s + \frac{1}{Td2}\right) * \frac{Kp1 * K}{s + Kp1 * K} * \frac{d}{L} * \frac{\frac{5}{7}g}{s \left(s + \frac{30*\pi*\eta*R}{7*m}\right)} \quad (98)$$

Si se supone que  $1/Td2 = \frac{30\pi\eta R * fact}{7m}$ , de esta manera se cancela un polo con el cero que añade la acción derivativa para contribuir a adelantar la respuesta.

$$G_{ba2}(s) = \frac{Kp2 * Td2 * Kp1 * K * \frac{5}{7} * g * d}{L} \quad (99)$$

$$G_{ba2}(s) = \frac{L}{s(s + Kp1 * K)}$$

En la Figura 6-66 se muestra el lugar de las raíces del bucle abierto, cogiendo los datos de ejemplo para  $K=0.0999$ ,  $\tau_m=0.002$ ,  $m=0.05$ ,  $n=0.83$ ,  $R=0.025$ ,  $g=9.81$ ,  $d=0.022$  y  $L=0.6$ .

Figura 6-66. Lugar de las raíces  $G_{ba2}$ 

Si se agrupa como  $A = \frac{Kp2 * Td2 * Kp1 * K * \frac{5}{7} * g * d}{L}$  y haciendo la función de bucle cerrado

$$G_{bc2}(s) = \frac{A}{s^2 + Kp1 * K * s + A} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \quad (100)$$

Igualando el términos segundo y como la especificación de que es críticamente amortiguado  $\delta=1$ , por lo tanto

$$2\delta\omega_n = Kp1 * K ; \omega_n = \frac{Kp1 * K}{2} \quad (101)$$

Si el tiempo de establecimiento es del 5%,

$$ts_5 = \frac{3}{\delta\omega_n}; \omega_n = \frac{3}{ts_5} \quad (102)$$

Se sustituye la ecuación ( 101 ) en la ecuación ( 102 ) y despejando Kp1

$$\frac{3}{ts_5} = \frac{Kp1 * K}{2}; Kp1 = \frac{6}{K * ts_5} \quad (103)$$

Ahora se iguala el tercer término

$$\omega_n^2 = A = Kp2 * Td2 * Kp1 * K * d * 5 * \frac{g}{7L} \quad (104)$$

Y sabiendo de la ecuación ( 101 ) si elevamos al cuadrado ambos términos que

$$\omega_n^2 = \left( \frac{Kp1 * K}{2} \right)^2 \quad (105)$$

Igualando la ecuación ( 105 ) y la ecuación ( 104 )

$$\left( \frac{Kp1K}{2} \right)^2 = Kp2 * Td2 * Kp1 * K * d * 5 * \frac{g}{7L} \quad (106)$$

Despejando  $Kp2$

$$Kp2 = \frac{Kp1 * K * 7L}{2^2 * 5 * g * Td2 * d} \quad ( 107 )$$

Finalmente se pregunta: ¿qué diferencia encuentras sin la aproximación por Taylor?

El alumno podrá observar en la gráfica, al poner en funcionamiento la simulación con el control adecuado, la diferencia entre utilizar simplemente  $\alpha$  o  $\text{seno}(\alpha)$ . El error que se comete es mínimo y, por lo tanto, se puede decir que la aproximación que se lleva a cabo es buena. La barra se mueve ángulos muy pequeños que prácticamente tienden a cero y se tiene una precisión aceptable.

## 7. Conclusión

Debido a la situación actual producida por el COVID-19, que imposibilita el acceso al campus de la universidad de Jaén, no se ha podido llevar a cabo la realización de objetivos como el desarrollo de la maqueta del sistema barra-bola y su posterior implementación en un laboratorio remoto.

Sin embargo, se puede concluir que en las tres simulaciones virtuales realizadas a modo de prácticas los alumnos pueden tener la ventaja de llevar a la práctica los conocimientos teóricos adquiridos sin necesidad de movilizarse al laboratorio, teniendo una visión bastante cercana a lo que podría ser una práctica de este tipo de forma física y real. También dando la posibilidad de realizar a los alumnos ejercicios prácticos en situaciones como la que se vive actualmente en la que no se permite el acceso a los laboratorios físicos.

Otro impedimento, fue la rápida dinámica del sistema barra-bola que dificultaba su implementación como laboratorio remoto. Por consiguiente, se cambió la barra por un tubo cilíndrico lleno de un fluido donde se introduce la bola. Como resultado, la bola se mueve por el tubo sumergida en el fluido que ralentiza el movimiento de ésta.

En las dos primeras simulaciones se trataba al motor como parte del proceso mientras que en la tercera es el actuador, de esta manera se muestra a los alumnos la versatilidad que puede tener un elemento. Según el tipo de proceso que se quiere controlar el mismo elemento puede desempeñar un papel diferente.

Por otra parte, se muestra la posibilidad de personalización de cada práctica. Cada alumno tendrá unos datos distintos, podrá elegir el idioma, cuando realizar el ejercicio y si quiere volver a repetirlo. Por ejemplo, se utiliza la viscosidad de la mezcla compuesta por un porcentaje de agua y glicerina, haciendo que dichos porcentajes varíen según los datos del alumno.

Asimismo, la recogida de datos de seguimiento como puntuación, tiempo de ejecución y respuestas dadas por el alumno serán inmediatas a la salida del alumno de la sesión del ejercicio.

A nivel económico se puede destacar el bajo coste que lleva la implementación de estas simulaciones y teniendo en cuenta que este tipo de laboratorio virtual podría realizarse para muchas más asignaturas beneficiándose de todas las ventajas anteriores.

## 8. Referencias

[1] [HTTP://WWW.SC.EHU.ES/SBWEB/FISICA/DINAMICA/STOKES/STOKES.HTML](http://www.sc.ehu.es/sbweb/fisica/dinamica/stokes/stokes.html)

[2] [HTTPS://WWW.EASY-LMS.COM/ES/CENTRO-DE-CONOCIMIENTO/CENTRO-DE-CONOCIMIENTO-LMS/QUE-ES-SCORM/ITEM10195#:~:TEXT=SCORM%20ES%20UNA%20ABREVIATURA%20Y,POR%20SU%20SIGLA%20EN%20INGL%C3%A9S\).&TEXT=SCORM%20TE%20PERMITE%20EMPAQUETAR%20TU%20CONTENIDO%20Y%20COMPARTIRLO%20CON%20OTROS%20SISTEMAS.](https://www.easy-lms.com/es/centro-de-conocimiento/centro-de-conocimiento-lms/que-es-scorm/item10195#:~:text=SCORM%20es%20una%20abreviatura%20y,por%20su%20sigla%20en%20ingl%C3%A9s).&text=SCORM%20te%20permite%20empaquetar%20tu%20contenido%20y%20compartirlo%20con%20otros%20sistemas.)

[3] [HTTPS://CONTROLAUTOMATICOEDUCACION.COM/CONTROL-REALIMENTADO/CONTROL-EN-CASCADA/](https://controlautomaticoeducacion.com/control-realimentado/control-en-cascada/)

[4] [HTTPS://DV.UJAEN.ES/GOTO\\_DOCENCIA\\_FILE\\_1108934\\_DOWNLOAD.HTML](https://dv.ujaen.es/goto_docencia_file_1108934_download.html)

## 9. Bibliografía

EJS Wiki: <https://www.um.es/fem/EjsWiki/Main/ElementsSoundOn>

Scorm 2004 4th edition RTE, Paul Jesukiewicz, Director Advanced Distributed Learning Initiative Office of Deputy Under Secretary of Defense for Readiness and Training, Policy and Programs.

## 10. ANEXO

### 10.1. Ejemplo de utilización del EJS

Para explicar cómo crear una simulación en EJS se utiliza el ejemplo MassAndSpring de los propios ejemplos de Javascript. En este ejemplo se estudia el comportamiento de una masa situada al final de un muelle que se puede estirar y comprimir. Cada proyecto de EJS tiene varias ventanas que se tienen que ir diseñando para crear el proyecto.

La primera ventana es la descripción donde se pondrá el enunciado de la práctica y las actividades a desarrollar como se muestra en Figura 10-1 y Figura 10-2

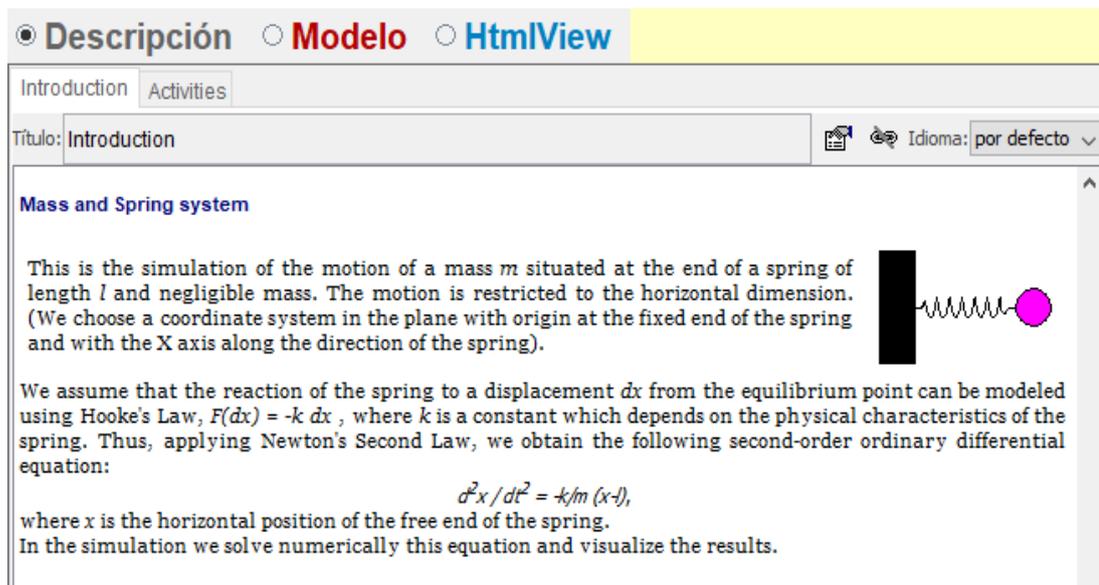


Figura 10-1. Descripción. Introducción

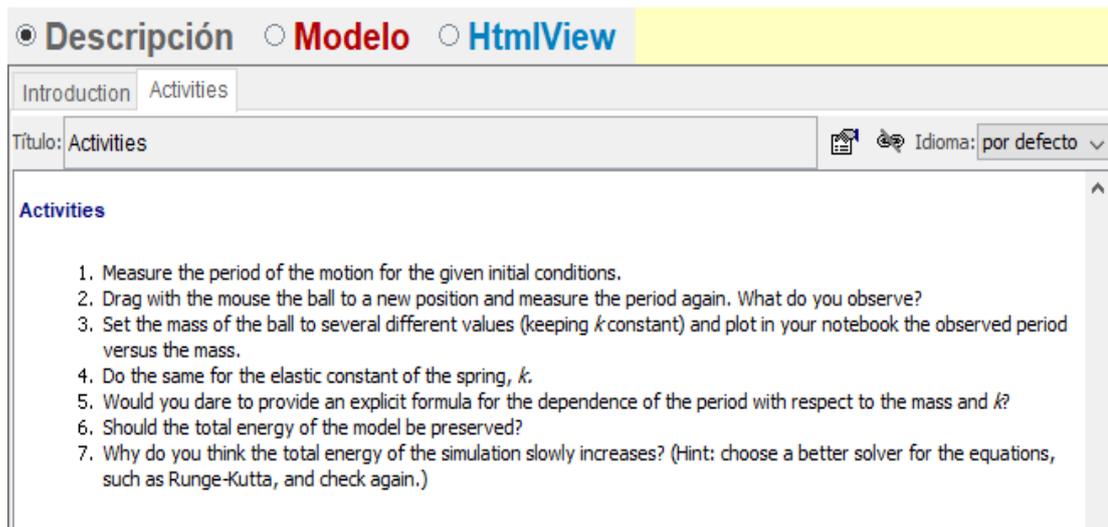


Figura 10-2. Descripción. Actividades

En el apartado de Modelo se incluye otras ventanas, la primera es variables donde se añaden las variables dinámicas, constantes, su tipo, u valor inicial y el nombre que se le da. Como el ejemplo que aparece en la Figura 10-3

Nombre	Valor inicial	Tipo
$x$	1.5	double
$v_x$	0.0	double
$t$	0.0	double
$dt$	0.05	double

Figura 10-3. Variables dinámicas

La segunda ventana es inicialización, en ella se añaden la parte de código que se quiere utilizar al principio del ejemplo y sólo se cargue al principio. En este ejemplo no hay código de ese tipo.

En la tercera ventana, llamada evolución, se añaden las ecuaciones que rigen el movimiento del muelle utilizando la segunda ley de Newton y expresándolo en forma de ecuación diferencial. Figura 10-4

Ecuación que rige el movimiento de un muelle con una masa al final de éste.

$$m * a(t) = -k * x(t); \quad a(t) = -k * \frac{x(t)}{m} ; \quad \frac{dv_x(t)}{dt} = -\frac{k}{m} * x(t)$$

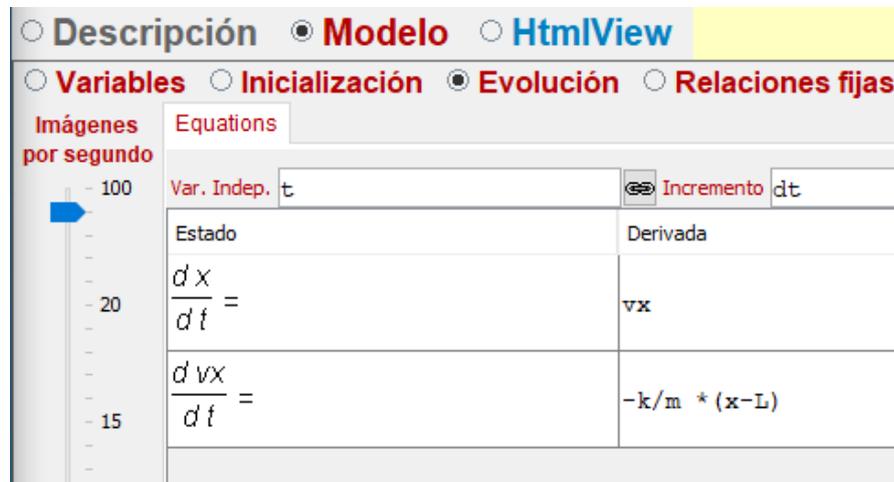


Figura 10-4. Evolución. Ecuaciones

La cuarta ventana es Relaciones Fijas y se utiliza para añadir esa parte de código que son fórmulas que no cambian, en el ejemplo aparecen las fórmulas de energía cinética, potencial y la suma de las dos (Figura 10-5).

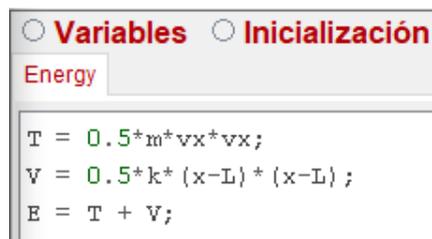


Figura 10-5. Relaciones Fijas. Fórmulas energías.

En la ventana de Propio se utiliza para escribir funciones que el propio desarrollador cree convenientes implementar porque se va utilizar en otras partes de código y así no tener que repetirlo. En este ejemplo no hay de ese tipo.

En el último apartado, HTMLView, se añade todos los elementos para crear la vista. Hay elementos para la interfaz, para dibujo 2D y dibujo 3D. Como aparece en la Figura 10-6, a la izquierda el árbol de elementos diseñado para la vista de este ejemplo y a la derecha los elementos que se encuentran disponibles para utilizarlos.

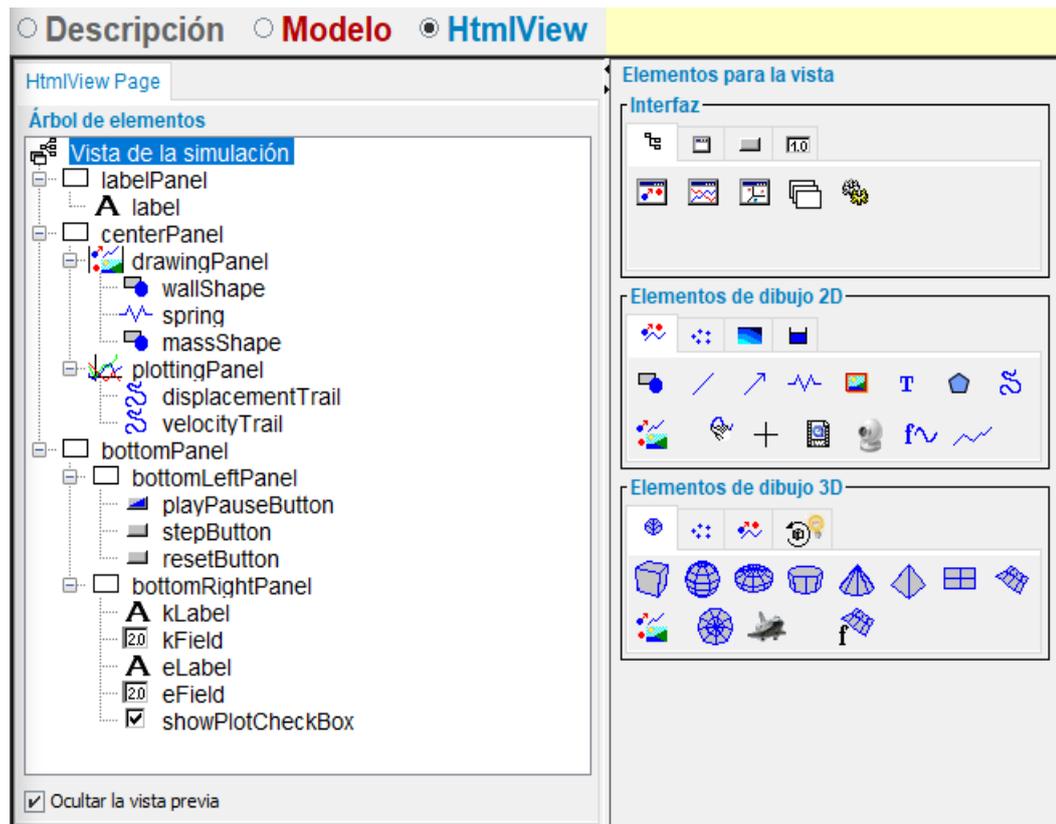


Figura 10-6. HTML View

Y en la Figura 10-7 se aprecia lo que sería la vista previa del ejemplo MassAndSpring

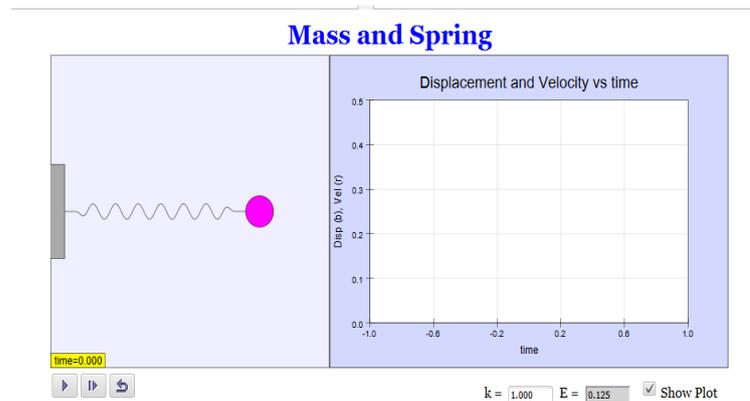


Figura 10-7. Vista previa MassAndSpring

## 10.2. Prácticas diseñadas

En el siguiente enlace se puede acceder a la carpeta donde se encuentran las tres simulaciones. (P1: Control en posición del motor, P2: Control en velocidad del motor y P3: Control del sistema barra-bola)

[https://dv.ujaen.es/ilias.php?ref\\_id=1108336&cmd=frameset&cmdClass=ilrepositorygui&cmdNode=st&baseClass=ilrepositorygui](https://dv.ujaen.es/ilias.php?ref_id=1108336&cmd=frameset&cmdClass=ilrepositorygui&cmdNode=st&baseClass=ilrepositorygui)

### **10.3. Solucionario de las prácticas diseñadas.**

En el siguiente enlace se puede acceder al solucionario en formato tabla de Excel de las tres prácticas diseñadas. Hay tres hojas cada hoja tiene las soluciones correspondientes a cada práctica y separadas por los apartados correspondientes de cada una de ella.

[https://dv.ujaen.es/ilias.php?ref\\_id=1128644&cmdClass=ilobjfilegui&cmdNode=st:ka&baseClass=ilRepositoryGUI](https://dv.ujaen.es/ilias.php?ref_id=1128644&cmdClass=ilobjfilegui&cmdNode=st:ka&baseClass=ilRepositoryGUI)