



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



**SISTEMAS NO LINEALES - MUAR 2023/2024**

**DISEÑO DE UN MODELO FÍSICO Y  
DOS TIPOS DE CONTROLADORES NO LINEALES  
PARA EL JUEGO MONZA**

David Redondo Quintero - Pablo Vela Silva

Universidad Politécnica de Madrid

24-01-2024

# ÍNDICE

1. INTRODUCCIÓN Y OBJETIVOS
2. DISEÑO DEL MODELO PROPIO
  - 2.1. ECUACIONES
  - 2.2. SIMULADOR Y DIAGRAMA DE BLOQUES
  - 2.3. FUNCIONAMIENTO Y COMPARATIVA CON EL MODELO PROPORCIONADO POR EL PROFESOR
3. ALTERNATIVAS DE CONTROL
  - 3.1. CONTROL DIFUSO
  - 3.2. GAIN SCHEDULING
  - 3.3. CONTROL ADAPTATIVO
  - 3.4. CONTROL INTELIGENTE
4. IMPLEMENTACIÓN DEL CONTROLADOR BORROSO
  - 4.1. VARIABLES DE ENTRADA Y SALIDA. REGLAS
  - 4.1. DIAGRAMA DE BLOQUES EN SIMULINK
  - 4.3. RESULTADOS EN MODELO DADO. CONCLUSIONES
  - 4.4. RESULTADOS EN MODELO PROPIO. CONCLUSIONES
5. IMPLEMENTACIÓN DEL CONTROLADOR ADAPTATIVO
  - 5.1. DIAGRAMA EN SIMULINK
  - 5.2. MODIFICACIÓN
  - 5.3. RESULTADOS EN MODELO DADO. CONCLUSIONES
  - 5.4. RESULTADOS EN MODELO PROPIO. CONCLUSIONES

# 1. INTRODUCCIÓN Y OBJETIVOS

El proyecto planteado en esta asignatura consiste en desarrollar una simulación que consiga completar el juego ‘Monza’.

Su funcionamiento es sencillo: se introduce una moneda por una rendija en la parte superior y, mediante un volante, se controla el giro del compartimento con la idea de conseguir que caiga hacia las pistas inferiores sin salirse del recorrido.

Los distintos carriles paralelos tienen forma de parábola, de modo que si la moneda cae por el extremo correcto puede continuar con su recorrido al carril inferior, pero si cae por el extremo contrario la moneda se pierde y el juego termina.

El objetivo es desarrollar dos tipos de controladores capaces de resolver correctamente cada uno de los niveles de dificultad que presenta el juego, de forma que, a mayor dificultad mayor longitud tendrán las pistas y más complicado será realizar un control que evite que la moneda salga del recorrido.

Al tratarse de un modelo no lineal, se deberá recurrir a técnicas de control no lineal, dejando de lado controladores sencillos ampliamente conocidos como el PID.

Para la resolución del problema se proporciona una planta en Matlab y Simulink con el modelo físico ya construido, de forma que simplemente el desarrollo del controlador permita automatizar la tarea. Sin embargo, la idea es que éste modelo ya construido sirva como referencia, como si se tratase del modelo real, de forma que cada grupo deberá crear su propio modelo, en base al cálculo e implementación de las ecuaciones físicas y a la construcción del mismo en Simulink.



*Ilustración 1. Monza*

## 2. DISEÑO DEL MODELO PROPIO

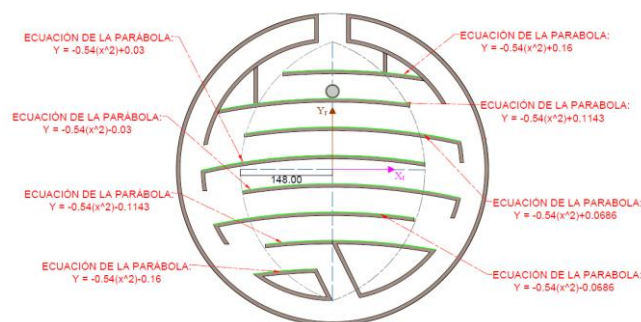
### 2.1. ECUACIONES

El primer paso a la hora de implementar el modelo es el cálculo de las ecuaciones físicas que determinan el comportamiento del sistema.

Existen dos fases bien diferenciadas a la hora de determinar la mecánica del sistema:

#### 2.1.1. MOVIMIENTO SOBRE LAS PISTAS

Para poder definir el comportamiento de la moneda sobre cada carril es necesario disponer de las ecuaciones que definen cada parábola.



*Ilustración 2. Ecuaciones de las pistas*

Como se puede observar, la ecuación es la misma para todos los casos, variando únicamente el valor del término independiente. Por tanto, la ecuación de las parábolas queda definida de la siguiente forma:

$$y = -0.54 \cdot x^2 + b; (1)$$

$b$ : altura de cada pista tomando como referencia el punto central del disco giratorio.

En el instante inicial, al introducir la moneda por la rendija superior estando el volante completamente recto, existe la posibilidad de que ésta quede en un estado de completo reposo. Esto se debe a la existencia de situaciones de equilibrio, en las que la moneda pierde toda su energía cinética. Sin embargo, este estado de equilibrio es tremendamente difícil de alcanzar. Se deben dar unas condiciones muy preciosas para conseguir mantener la moneda quieta, lo que hace estos puntos de equilibrio sean altamente inestables, ya que cualquier ligera perturbación sacará a la moneda de su estado de reposo, haciendo que ruede por el carril en el que se encuentre.

Sin embargo, el objetivo del juego es precisamente ese: sacar a la moneda de su estado de reposo o intentar estabilizarla en función de la velocidad a la que se desee que ruede. La principal perturbación en este sistema es el giro controlado del disco móvil, que permite variar el ángulo de inclinación de las pistas. De este modo, por acción de la propia gravedad, la moneda rodará de un carril a otro.

Este movimiento de caída se asemeja al clásico problema de un cuerpo rígido cayendo por un plano inclinado. Sin embargo, la particularidad es que en nuestro plano inclinado la pendiente

varía según cambia la distancia en el eje horizontal. En una rampa típica los objetos caen a velocidad constante, puesto que el ángulo de inclinación no varía. Sin embargo, si el plano es una parábola, la moneda caerá cada vez con más velocidad según se aleje del punto más alto ya que el ángulo de inclinación cada vez es mayor.

De este modo, partiendo de la ecuación anterior se puede definir el ángulo de inclinación  $\alpha$  en cada punto de la parábola de la siguiente forma:

PENDIENTE DE LA PARÁBOLA EN CADA PUNTO:  $\frac{dF}{dx} = -1.08 \cdot x$  (2)

ANGULO DE LA PARÁBOLA EN CADA PUNTO:  $\alpha = \text{atan}(-1.08 \cdot x)$  (3)

Otro concepto importante son los sistemas de referencia utilizados. Se pueden distinguir tres sistemas distintos:

- Sistema inercial. Sistema del mundo. En otras palabras, es el sistema de referencia fijo, cuyos ejes se mantienen inmóviles, aunque el disco gire.
- Sistema referencial. Este sistema gira de forma solidaria al disco, en función del ángulo  $\theta$ .
- Sistema de la moneda. Desde lejos podría parecer que el sistema referencial y el de la moneda son el mismo, pero al hacer zoom se puede apreciar que existe cierta diferencia entre ambos, la cual se debe precisamente al ángulo de inclinación variable de las parábolas, al cual hemos denominado  $\alpha$ . El punto superior más alto de cada parábola será el único en el que estos dos sistemas sean coincidentes.

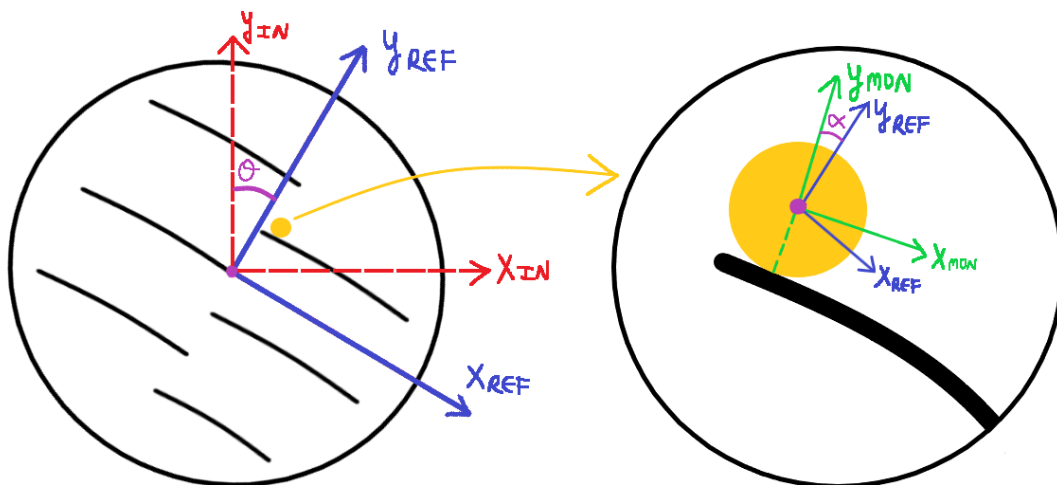


Ilustración 3. Sistemas de referencia

Tras analizar los tres sistemas de referencia existentes se pueden realizar distintos enfoques para obtener las fórmulas del modelo a implementar, dependiendo del sistema que se quiera usar como base.

Todos los enfoques son similares, puesto que las fórmulas físicas no varían, pero si la forma en la que se opera con los ángulos  $\theta$  y  $\alpha$ .

En este caso se van a plantear las ecuaciones desde el punto de vista de la moneda.

Como se ha comentado anteriormente, las pistas son planos inclinados en los que varía la inclinación según aumenta la coordenada 'x'. Sin embargo, al usar como referencia el sistema de la moneda, se puede hacer la siguiente suposición: la moneda rueda sobre un plano recto sin inclinación y es la dirección de la gravedad la que varía en función del ángulo de inclinación del disco y de las parábolas.

De este modo, desarrollando la segunda ley de Newton, las fórmulas quedarían de la siguiente manera:

$$F = M \cdot a \quad F_G + F_R = M \cdot a_{moneda} \quad (4)$$

$$F_G = M_{moneda} \cdot g \cdot \sin(\Phi) \quad (5)$$

$$F_R = \mu \cdot V_{moneda} \quad (6)$$

Siendo  $\Phi$  el ángulo total:  $\Phi = \theta + \alpha$  (El signo de este ángulo dependerá del sentido de giro del disco móvil)

Despejando la aceleración de la moneda, se obtiene:

$$a_{moneda} = g \cdot \sin(\Phi) - \frac{\mu \cdot V_{moneda}}{M_{moneda}} \quad (7)$$

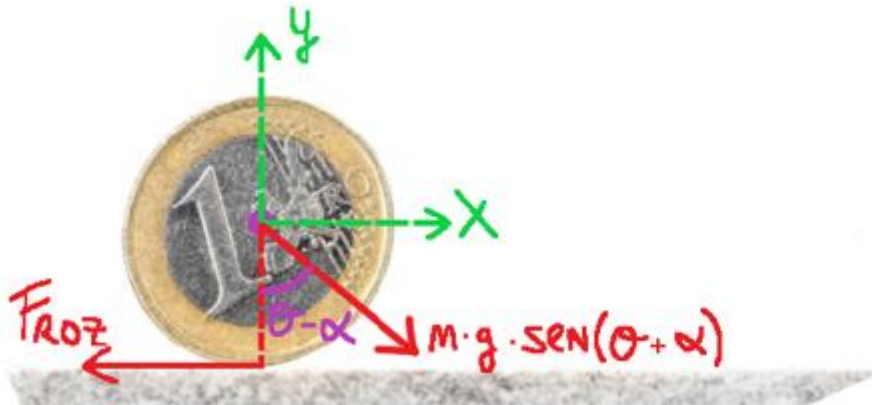


Ilustración 4. Fuerzas durante el movimiento por el carril

Una vez obtenida la aceleración de la moneda se utiliza un bloque de Simulink para integrar y obtener tanto velocidad como posición.

El siguiente paso es transformar las coordenadas de la moneda a coordenadas referenciales, para poder operar correctamente en el modelo simulado. El primer paso para ello es aplicar la siguiente ecuación:

$$x_{REF} = x_{MONEDA} \cdot \cos(\alpha) - y_{MONEDA} \cdot \sin(\alpha) \quad (8)$$

Para obtener  $y_{MON}$  se deben utilizar las coordenadas del sistema referencial calculadas a partir de las coordenadas en el sistema de la moneda:

$$y_{MONEDA} = x_{REF} \cdot \sin(1.08 \cdot x_{REF}) + y_{REF} \cdot \cos(1.08 \cdot x_{REF}) \quad (9)$$

El valor de  $y_{REF}$  se calcula sustituyendo  $x_{REF}$  en la ecuación de la parábola sobre la cual se encuentra la moneda, (ecuación (1)). Todo estas transformaciones y cambios de ejes generan un lazo algebraico que hay que romper en Simulink.

### 2.1.2. CAIDA A LA PISTA INFERIOR Y RECEPCIÓN

El siguiente paso es calcular las ecuaciones que rigen el comportamiento de la moneda desde que llega al final de un carril hasta que aterriza en el siguiente.

Antes de comenzar con el desarrollo, se van a suponer dos premisas:

- No existe rozamiento de la moneda con el aire.
- No existe rebote de la moneda al aterrizar en la pista inferior.

Cuando la moneda llega al final de la pista, su velocidad se divide en dos componentes, vertical y horizontal (en caso de que el ángulo de inclinación combinado sea 0 entonces solo habrá componente horizontal). En el momento exacto en que la moneda deje de estar en contacto con el carril se inicia el proceso de caída.

Las ecuaciones diferenciales que rigen su comportamiento durante esta fase (en el eje referencial) son:

$$acel\ x_{REF} = -\frac{d^2x}{dt^2} \cdot \sin(\theta) \cdot g \quad (11)$$

$$acel\ y_{REF} = -\frac{d^2y}{dt^2} \cdot \cos(\theta) \cdot g \quad (12)$$

Integrando respecto al tiempo:

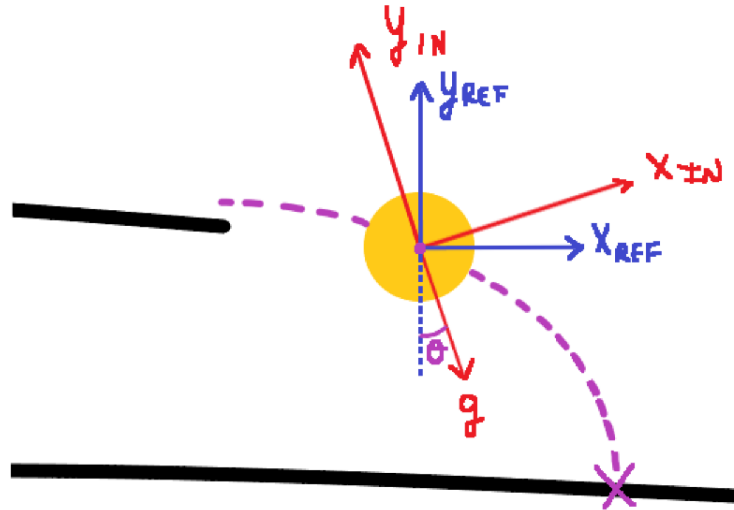
$$x_{REF}(t) = x_{REF\ INICIAL} + vel\ x_{REF} \cdot t - \frac{1}{2}g \cdot \sin(\theta) \cdot t^2 \quad (13)$$

$$y_{REF}(t) = y_{REF\ INICIAL} + vel\ y_{REF} \cdot t - \frac{1}{2}g \cdot \cos(\theta) \cdot t^2 \quad (14)$$

Para calcular el punto de recepción de la moneda en el nuevo carril, se determina el punto de corte de la trayectoria de la caída en función del tiempo con la ecuación de la parábola de dicho carril. Como se pudo apreciar en la ecuación (15), se construye la función de corte entre la pista y la trayectoria de la moneda. La raíz de esta función auxiliar es el instante de tiempo en el que se produce el impacto.

$$f(t) = y_{REF}(t) - y_{PARABOLA}$$

$$f(t) = y_{REF}(t) - (-0.54x_{REF}(t)^2 + b) \quad (15)$$



*Ilustración 5. Esquema gráfico de la caída de la moneda.*

Para calcular la raíz se utiliza la función ‘fzero’ de Matlab. El cálculo de la nueva posición es inmediato, basta con sustituir dicho instante de tiempo en las ecuaciones (13) y (14).

Este proceso se lleva a cabo en cada instante de tiempo de la simulación, de forma que si el disco gira mientras la moneda está en caída, su trayectoria se verá alterada en función del ángulo de giro  $\theta$ .

Una vez en la nueva pista, se deberá volver a utilizar las ecuaciones desarrolladas en el apartado anterior, puesto que la dinámica de moneda habrá cambiado de nuevo a la fase de rodadura.

Para ello se incluye un bloque ‘switch’, que es capaz de cambiar entre ambas dinámicas, como se podrá observar en el siguiente apartado.

## 2.2. SIMULADOR Y DIAGRAMA DE BLOQUES

El modelo propio ha sido construido en Simulink apoyándonos en la estructura del simulador proporcionado. A partir de éste se han implementado las ecuaciones comentadas en el inicio.

El objetivo es construir un simulador en el que hacer las pruebas de los controladores, por lo que cuanto más se parezca al modelo real (modelo del profesor) mejor será el desempeño de estos.

En las siguientes imágenes se detallan los bloques del diagrama:



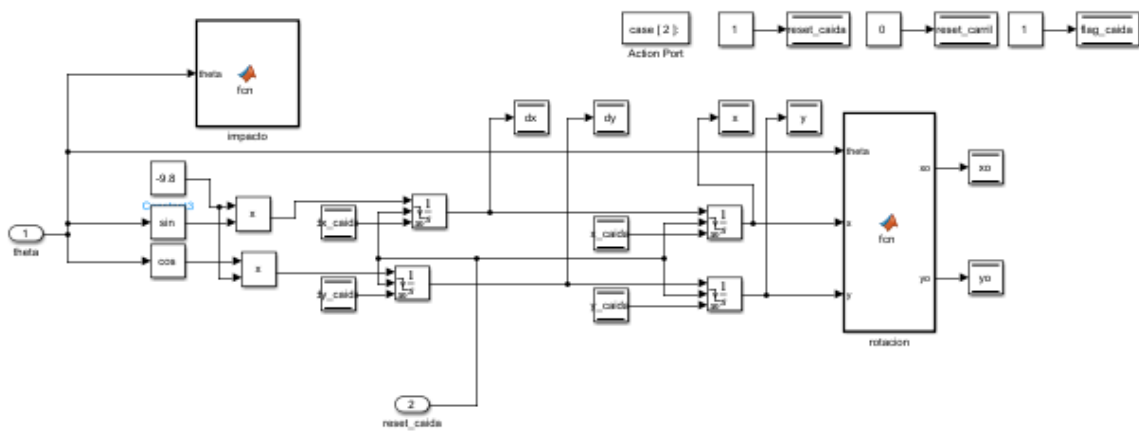
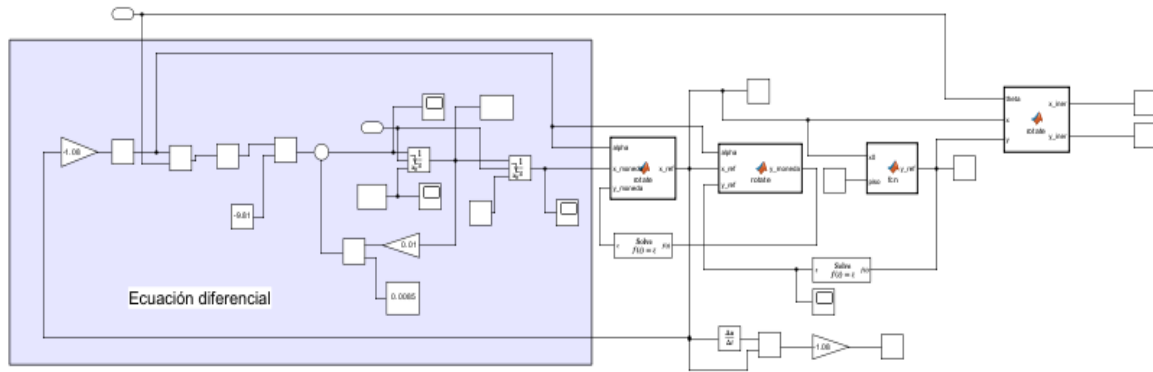


Ilustración 6A y 6B. Diagrama de bloques del simulador construido.

En la ilustración 6A se puede apreciar el diagrama de bloques referente al movimiento por las pistas, mientras que en la ilustración 6B se observa el diagrama correspondiente a la dinámica de la moneda durante el proceso de caída.

## 2.3. COMPARATIVA CON MODELO DADO

Primero se pone a prueba el modelo real con el propio inclinando el Monza 0.2 radianes hacia la derecha y evaluando el movimiento de la moneda en el eje referencial.

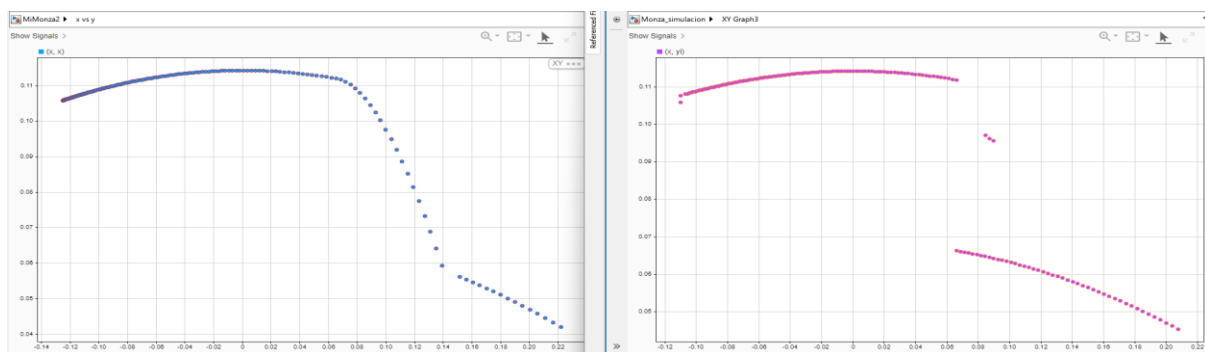
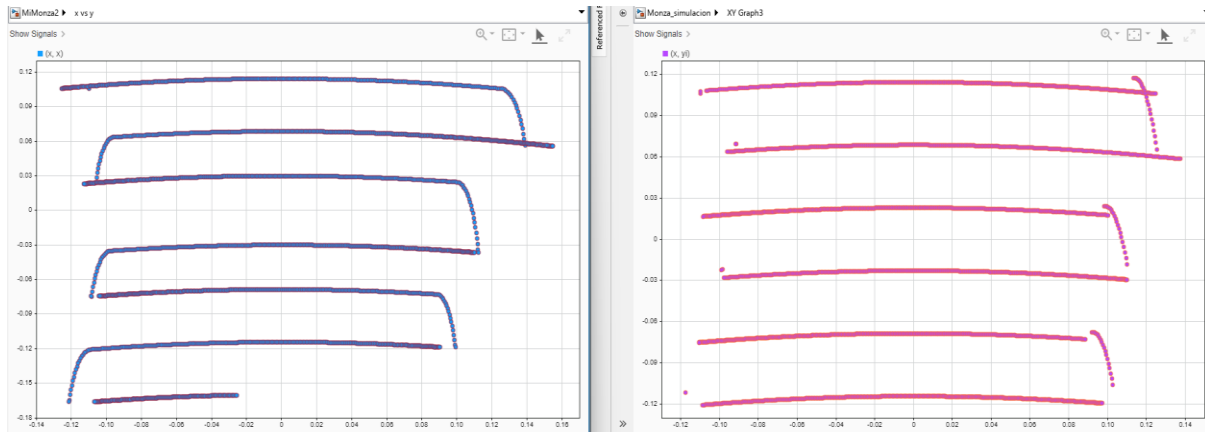


Ilustración 7. Movimiento de caída en el modelo propio (izquierda) y en el dado (derecha)

Como se puede observar, comportamiento es muy parecido, sobre todo en el carril. Por otro lado, en el modelo real no se genera bien la parábola de caída, por lo que no es posible compararlas.

Para la comparativa final se ha seleccionado el escenario de dificultad 3 y se muestra el movimiento de la moneda visto desde el eje referencial.



*Ilustración 8. Trayectoria de la moneda al ejecutar la dificultad 3 en el modelo propio (izquierda) y en el dado (derecha)*

Se pueden observar una clara diferencia en la distancia que se mueve la moneda en los carriles, siendo mayor en el simulador diseñado

Otro detalle que llama mucho la atención es que, en nuestro simulador, las caídas no están bien implementadas del todo. En el cambio de modo es posible que existe algún error con la nueva posición de la moneda tras la caída. Sin embargo, en el modelo proporcionado no ocurre este problema.

Más allá de estos detalles, parece que el simulador propio se ajusta bastante bien a la dinámica de la planta real, por lo que se considera adecuado para experimentar con él.

### 3. ALTERNATIVAS DE CONTROL

Una alternativa frecuentemente utilizada para el control de sistemas no lineales es recurrir a la linealización. Este método implica encontrar la aproximación lineal a una función en un punto dado y estudiar su estabilidad. En caso de que el sistema se aleje de dicho punto de trabajo, el controlador lineal será incapaz de controlar correctamente el sistema.

Por lo tanto, debido a estos inconvenientes y a la elevada no linealidad del sistema, es necesario realizar un estudio de las posibles estrategias de control para sistemas no lineales, proponiéndose a continuación varios tipos de controladores.

### 3.1. CONTROL DIFUSO

El control difuso permite digitalizar el conocimiento experto en un conjunto de reglas que se emplean para determinar que acciones y con que intensidad se realizan. La lógica borrosa resulta adecuada en situaciones de incertidumbre, con información cualitativa o en la resolución de problemas que un experto humano puede abordar con facilidad. Existen múltiples razones por las que este tipo de controladores se adaptan muy bien a este tipo de problemas:

- **Manejo de Incertidumbre:** Los sistemas de control borroso son robustos frente a la incertidumbre y variabilidad en el sistema. Dado que el juego Monza involucra factores impredecibles como la velocidad inicial de la moneda y pequeñas variaciones en la inclinación de las pistas, un controlador borroso puede adaptarse bien a estas condiciones cambiantes.
- **Modelado No Lineal.** Como ya es sabido, la dinámica del Monza es no lineal debido a la forma de las pistas en parábolas y la aceleración variable de la moneda. Los controladores borrosos son especialmente eficaces en sistemas no lineales, ya que no requieren una representación precisa del modelo matemático.
- **Facilidad de Implementación:** Los sistemas de lógica difusa son intuitivos y se pueden implementar de manera sencilla, gracias a interfaces como la toolbox de Matlab.
- **Toma de Decisiones Bajo Condiciones Incompletas:** En situaciones donde la información del sistema es incompleta, un controlador borroso puede tomar decisiones razonables basándose en reglas lingüísticas, haciendo frente a la falta de información precisa.

### 3.2. GAIN SCHEDULING

La técnica del Gain scheduling consiste en diseñar un controlador para cada punto de funcionamiento del sistema. Este diseño amplía los márgenes de funcionamiento de controladores lineales como el PID en entornos no lineales. No obstante, puede ser una tarea muy tediosa. Sus principales ventajas dentro de este problema son:

- **Adaptabilidad:** Permite adaptarse a los cambios de inclinación de las pistas.
- **Estabilidad mejorada:** Al ajustar los parámetros según las circunstancias, se puede mantener la estabilidad del sistema tanto en las situaciones de movimiento de carril como en la de recepción de la moneda tras la caída.

No obstante, implementar este controlador implicaría diseñar varios PID's según la cantidad de puntos de trabajos definidos. Para esta aplicación con dos controladores sería suficiente; un controlador suave para el movimiento a lo largo de la pista, y uno más agresivo para la recepción de la moneda en el nuevo carril.

La principal complicación que puede surgir sería la implementación y ajuste de ganancias variables ya que requiere un conocimiento profundo del sistema. Además, hay que tener cuidado al realizar el 'switching' entre controladores, ya que una mala implementación puede provocar que el sistema se vuelva inestable.

### **3.3. CONTROL ADAPTATIVO**

La idea principal detrás del control adaptativo es permitir que el sistema de control se adapte y modifique sus parámetros de acuerdo con las variaciones o cambios en las características del sistema que está controlando.

- **Ajuste en Tiempo Real.** El controlador adapta sus parámetros de manera dinámica mientras el sistema está en funcionamiento.
- **Autoajuste.** El control adaptativo busca ajustar sus propios parámetros automáticamente sin intervención externa.
- **Compensación ante variaciones.** Es capaz de compensar perturbaciones no modeladas o variaciones en los parámetros del sistema que pueden surgir durante la operación.

Aunque el control adaptativo ofrece beneficios significativos en términos de ajuste dinámico a cambios en las condiciones del sistema, también presenta desventajas y desafíos. Algunas de las limitaciones y desventajas del control adaptativo incluyen:

- **Complejidad del Diseño.** La implementación de sistemas de control adaptativo puede ser más compleja en comparación con sistemas de control convencionales. El diseño y ajuste de algoritmos adaptativos pueden requerir un conocimiento especializado.
- **Requerimientos de Identificación del Sistema.** El éxito del control adaptativo a menudo depende de la capacidad para identificar con precisión las características y parámetros del sistema. Si la identificación del sistema es incorrecta, el rendimiento adaptativo puede verse comprometido.

### **3.4. CONTROL INTELIGENTE**

El control inteligente se basa en redes neuronales. La información de entrada pasa a través de la red de neuronas y se genera un valor de salida en función de los valores de peso entre las neuronas y el umbral de cada una. Estos sistemas aprenden y se entrenan por sí mismos en lugar de ser programados explícitamente. La idea principal sería que los ‘inputs’ de la red fueran la posición referencial de la moneda y su velocidad, mientras que el output sería el giro del disco.

La principal ventaja de este tipo de control es que no requiere programación definida. En definitiva, la red neuronal se entrena a sí misma, ajustando pesos y umbrales hasta alcanzar objetivos definidos.

Una vez examinadas estas alternativas, se opta por usar un controlador difuso y un control adaptativo, puesto que creemos que son los métodos más sencillos de implementar. Además, son técnicas que ya hemos explorado en el máster, por lo que consideramos que ponerlas a prueba en este trabajo puede ayudarnos a reforzar los conocimientos adquiridos.

## 4. IMPLEMENTACIÓN DEL CONTROLADOR BORROSO

### 4.1. VARIABLES DE ENTRADA Y SALIDA. REGLAS

El primer paso a la hora de diseñar un controlador borroso es decidir cuáles van a ser sus entradas y salidas, conocidas normalmente como antecedentes y consecuentes:

- La salida es más evidente, y será el ángulo de giro (en radianes) con el que es necesario girar el disco en cada momento.
- En cuanto a las entradas, la primera se trata de la velocidad de la moneda. Es importante tener en cuenta este parámetro, ya que velocidades elevadas puede provocar la salida de la moneda del circuito durante el paso de una pista a otra.

Por otra parte, es necesario conocer la posición en la moneda en cada carril. Podría parecer que utilizar la posición referencial en 'x' que proporciona la planta como salida sería adecuado, sin embargo, en este caso se ha decidido que, en vez de eso, lo mejor es utilizar como entrada la distancia que le queda a la moneda para caer de cada carril. Esta distancia la denominaremos 'Error', y es la resta entre la coordenada x del punto final de cada carril y la posición referencial en x de la moneda en cada instante. En el siguiente apartado se explicará con más detalle en bloque diseñado para el cálculo de estos parámetros.

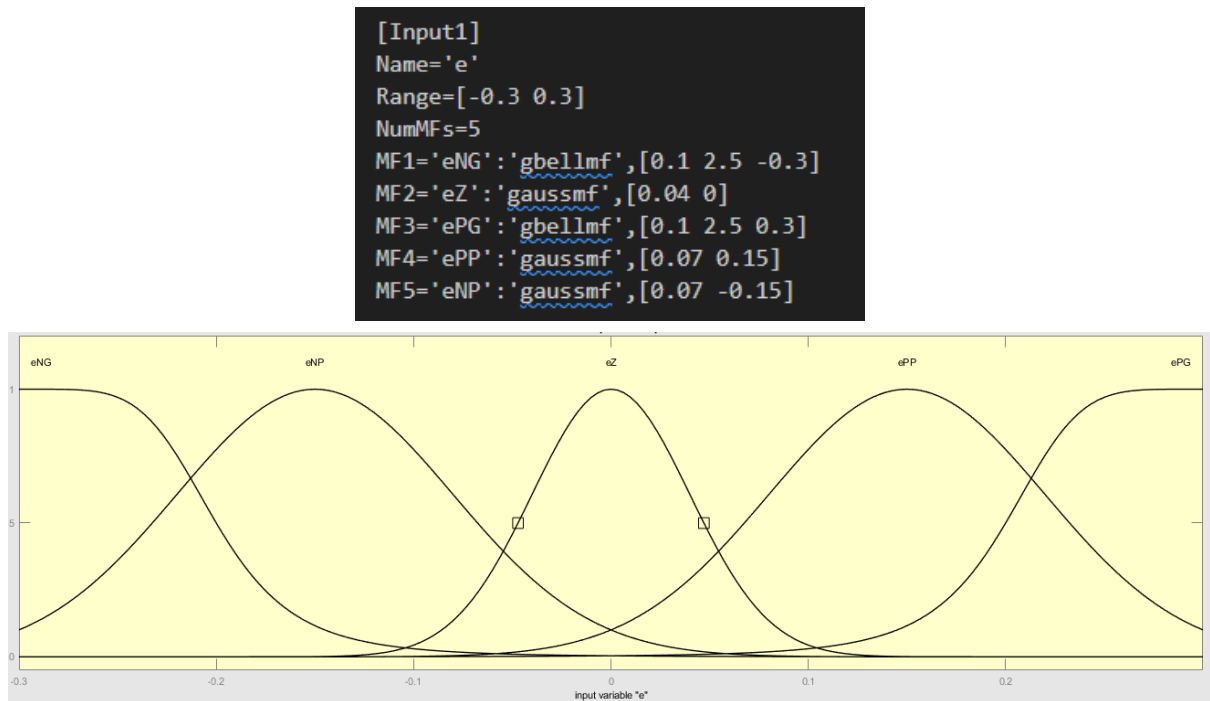
Una vez definidas las entradas y salidas del sistema se proceder a realizar el diseño del controlador. Para ello se ha utilizado la toolbox "fuzzy" de Matlab, que permite generar archivos de tipo '.fis' que posteriormente se pueden introducir en el bloque de control difuso de Simulink.

El primer paso es definir los antecedentes, sus valores lingüísticos y los 'fuzzy sets' o conjuntos difusos.

La primera variable lingüística será el Error 'e'. Se define su rango, que en este caso nunca tomará valores mayores de 0.3, ni menores de -0.3. En cuanto a los valores lingüísticos, se ha considerado crear 5 estados distintos: Error Negativo Grande (eNG), Error Negativo Pequeño (eNP), Error Cero (eZ), Error Positivo Pequeño (ePP) y Error Positivo Grande (ePG).

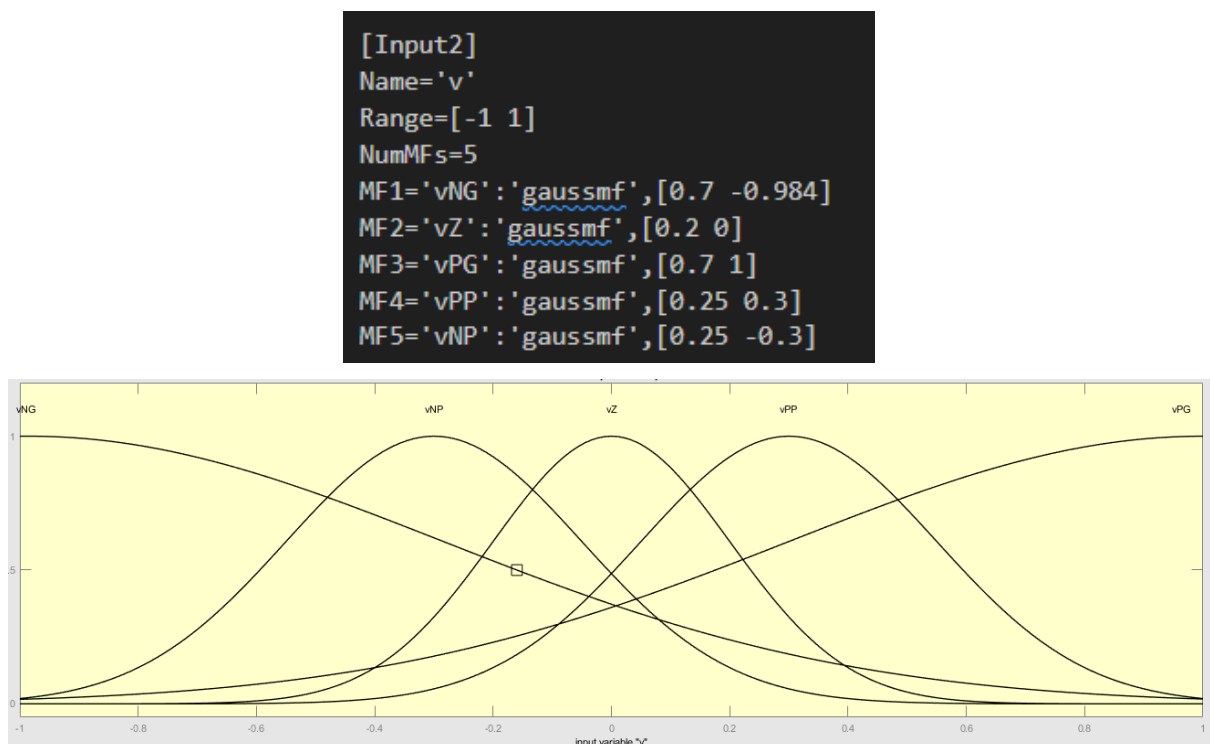
Como se puede observar, los conjuntos borrosos son de tipo gaussiano. No existe una causa específica por la que se haya escogido este tipo de geometría. Simplemente se probó a realizar el diseño de esta forma y se logró controlar el sistema, por lo que se ha mantenido así, aunque somos conscientes de que los conjuntos triangulares y trapezoidales son los más comunes y también se habría conseguido completar el juego en caso de haberlos utilizado.

Del mismo modo, los puntos iniciales y finales de cada conjunto se han ido ajustando de forma iterativa. No existe ninguna regla o fórmula que determine en que punto debe empezar el conjunto difuso asociado a cada valor lingüístico; se trata de un proceso de prueba y error, aunque es importante realizar un primer diseño acertado con valores lógicos y cercanos a los que hacen factible la solución.



*Ilustración 9. Vvalores lingüísticos y fuzzy sets asociados a la entrada Error*

Para la segunda variable Velocidad 'v' el proceso es el mismo. Su rango en este caso está definido entre -1 y 1. Sus valores lingüísticos son: Velocidad Negativa Grande (vNG), Velocidad Negativa Pequeña (vNP), Velocidad Cero (vZ), Velocidad Positiva Pequeña (vPP) y Velocidad Positiva Grande (vPG).



*Ilustración 10. Valores lingüísticos y fuzzy sets asociados a la entrada Velocidad*

En cuanto al consecuente (salida), simplemente es necesario definir los valores del giro del disco: Negativo Grande (NG), Negativo Pequeño (NP), Cero (Z), Positivo Pequeño (PP) y Positivo Grande (PG).

```
[Output1]
Name='output1'
Range=[-1 1]
NumMFs=5
MF1='NP': 'constant', [-0.1]
MF2='NG': 'constant', [-0.3]
MF3='Z': 'constant', [0]
MF4='PP': 'constant', [0.1]
MF5='PG': 'constant', [0.3]
```

*Ilustración 11. Valores lingüísticos y fuzzy sets asociados a la salida Giro*

En este caso los conjuntos difusos son valores constantes. Esto se debe a que se ha utilizado el método de Sugeno frente al de Mamdani, lo que presenta ciertas ventajas:

- Salida con Valores Constantes. El método de Sugeno permite asociar reglas borrosas con funciones lineales o constantes en lugar de funciones lingüísticas difusas. En el caso del Monza, al controlar el giro del disco, tener salidas que son valores constantes puede ser más natural y fácil de interpretar.
- Eficiencia Computacional. En algunos casos este método puede ser más eficiente desde el punto de vista computacional, ya que no requiere etapa adicional de defuzzificación. Esto puede ser beneficioso en sistemas en tiempo real o donde se busca una implementación más eficiente, como es el caso.

De todas formas, el método de Mamdani habría sido igual de válido para la implementación del controlador.

Por último, queda construir las reglas difusas para que el mecanismo de inferencia sea capaz de controlar el giro en función de las entradas que le lleguen.

- |   |   |
|---|---|
| 1. If (e is eZ) and (v is vZ) then (output1 is Z) (1)     | 13. If (e is ePP) and (v is vPG) then (output1 is PP) (1) |
| 2. If (e is eZ) and (v is vPP) then (output1 is Z) (1)    | 14. If (e is ePP) and (v is vPP) then (output1 is PP) (1) |
| 3. If (e is eZ) and (v is vNP) then (output1 is Z) (1)    | 15. If (e is ePP) and (v is vNP) then (output1 is NP) (1) |
| 4. If (e is eZ) and (v is vPG) then (output1 is Z) (1)    | 16. If (e is eNP) and (v is vNG) then (output1 is NP) (1) |
| 5. If (e is eZ) and (v is vNG) then (output1 is Z) (1)    | 17. If (e is eNP) and (v is vZ) then (output1 is PP) (1)  |
| 6. If (e is ePG) and (v is vPG) then (output1 is NG) (1)  | 18. If (e is eNP) and (v is vPG) then (output1 is PP) (1) |
| 7. If (e is ePG) and (v is vPP) then (output1 is NG) (1)  | 19. If (e is eNP) and (v is vPP) then (output1 is PP) (1) |
| 8. If (e is ePG) and (v is vNP) then (output1 is NG) (1)  | 20. If (e is eNP) and (v is vNP) then (output1 is NP) (1) |
| 9. If (e is ePG) and (v is vZ) then (output1 is NG) (1)   | 21. If (e is eNG) and (v is vPG) then (output1 is PG) (1) |
| 10. If (e is ePG) and (v is vNG) then (output1 is NG) (1) | 22. If (e is eNG) and (v is vPP) then (output1 is PG) (1) |
| 11. If (e is ePP) and (v is vNG) then (output1 is NP) (1) | 23. If (e is eNG) and (v is vNP) then (output1 is PG) (1) |
| 12. If (e is ePP) and (v is vZ) then (output1 is NP) (1)  | 24. If (e is eNG) and (v is vNG) then (output1 is PG) (1) |
| 13. If (e is ePP) and (v is vPG) then (output1 is PP) (1) | 25. If (e is eNG) and (v is vZ) then (output1 is PG) (1)  |

*Ilustración 12. Reglas del controlador difuso*

## 4.2. DIAGRAMA DE BLOQUES EN SIMULINK

Una vez definido el controlador, queda diseñar el diagrama de bloques en Simulink, el cual queda de la siguiente forma:

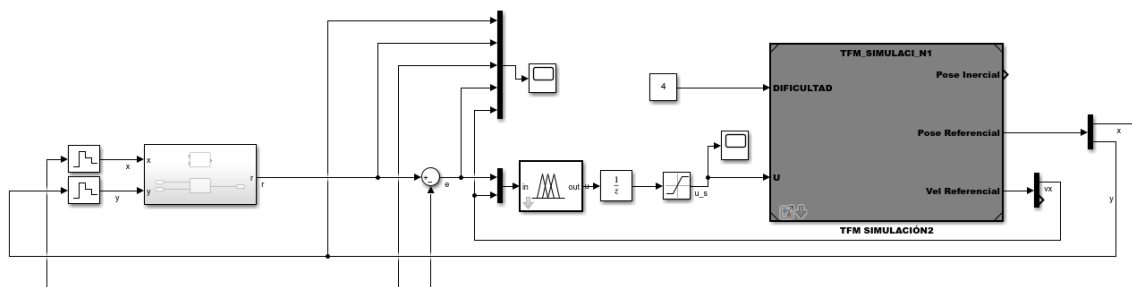


Ilustración 13. Diagrama de bloques del controlador difuso

Como se puede apreciar, se saca como salida de la planta tanto la posición referencial de la moneda en X e Y, como su velocidad.

Como se comentaba anteriormente, lo que nos interesa es realimentar dichos parámetros para que el controlador difuso pueda recibirlos como entrada.

En cuanto a la velocidad de la moneda, es posible realimentarla directamente como entrada del controlador.

Sin embargo, la posición referencial se introduce al bloque de la izquierda para hacer un tratamiento previo. En función de los valores de 'x' e 'y' entre los que se encuentre la moneda, la función incluida en este bloque es capaz de detectar la pista sobre la que se está moviendo, y modifica la variable 'r' con el valor de la coordenada x del punto final de dicho carril.

La salida de este bloque (r) se lleva a un sumador, en el cual se calcula el error 'e' como la resta entre 'r' y la posición referencial en 'x'.

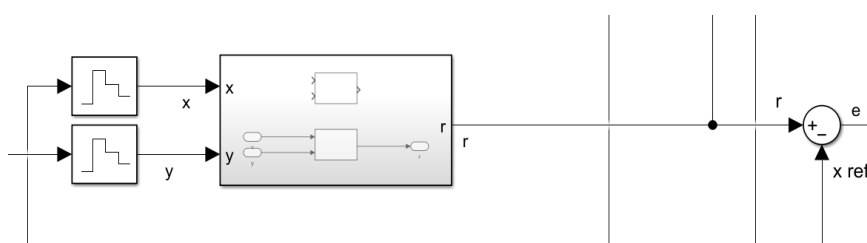
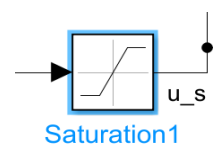


Ilustración 14. Bloque para el cálculo de 'r'

```
function r = fcn(x,y)
%#codegen
r=0;
if(x>-0.0256 && y>=-0.161 )
    r=-0.0256;
end
if(x>-0.11 && y>=-0.13 )
    r=-0.108;
end
if(x<0.132 && y>=-0.09 )
    r=0.1318;
end
if(x>-0.149 && y>=-0.045 )
    r=-0.14851;
end
if(x<0.1425 && y>=0.005 )
    r=0.14224;
end
if(x>-0.1425 && y>=0.05 )
    r=-0.14224;
end
if(x<0.12394 && y>=0.10)
    r=0.124;
end
```



Cabe destacar la presencia de un bloque de saturación, el cual evita que los valores giro del disco sean demasiado grandes. En principio el propio diseño del controlador difuso debería limitar los valores máximos que puede alcanzar el giro, pero nunca está de más añadir un bloque de saturación. Es una buena técnica que permite limitar la acción de control, sobre todo de cara a sobrepasar límites mecánicos en un supuesto modelo real.

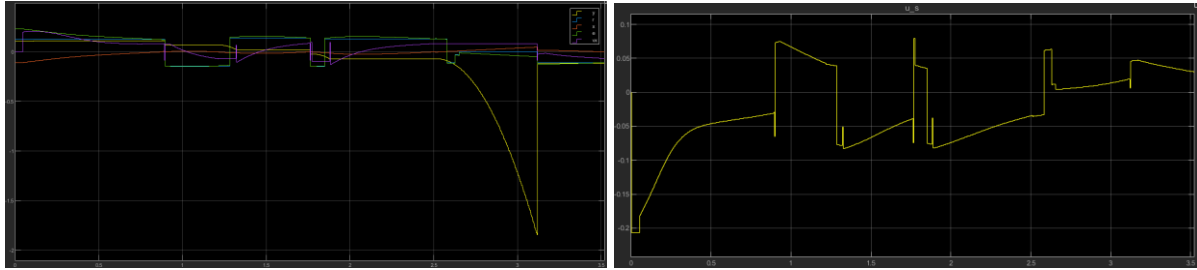


### 4.3. RESULTADOS EN MODELO DADO. CONCLUSIONES

#### DIFICULTAD 1

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



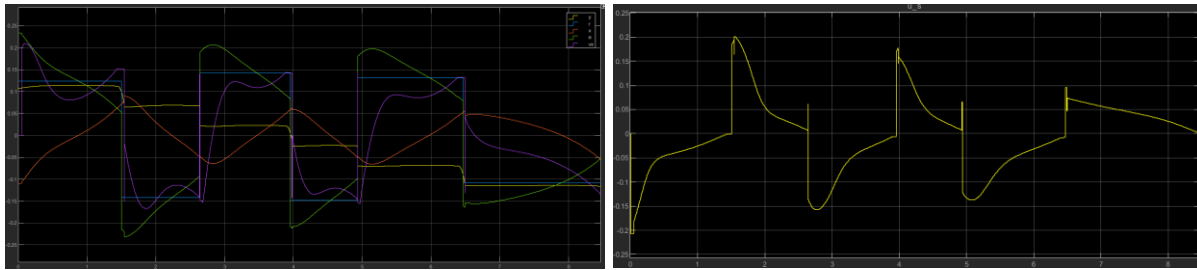
*Ilustración 15. Gráficas obtenidas para la dificultad 1 en el modelo proporcionado*

Tiempo total: **3.520 segundos**

#### DIFICULTAD 2

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



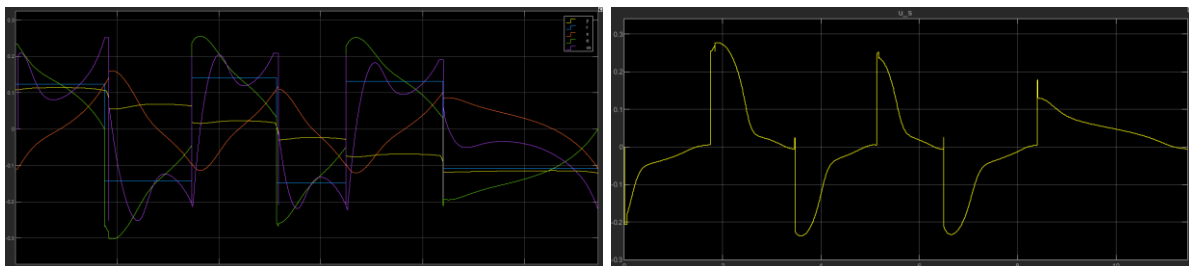
*Ilustración 16. Gráficas obtenidas para la dificultad 2 en el modelo proporcionado*

Tiempo total: **8.468 segundos**

#### DIFICULTAD 3

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



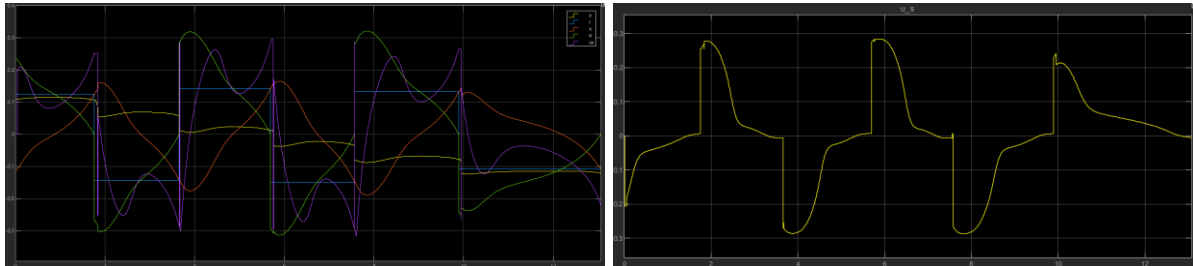
*Ilustración 17. Gráficas obtenidas para la dificultad 3 en el modelo proporcionado*

Tiempo total: **11.452 segundos**

## DIFICULTAD 4

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



*Ilustración 18. Gráficas obtenidas para la dificultad 4 en el modelo proporcionado*

Tiempo total: **13.69 segundos**

El primer detalle destacable es el pico que se produce en la gráfica de la dificultad 1. El controlador es capaz de superar el nivel haciendo que la moneda pase muy rápidamente de un carril a otro. Esto provoca que caiga desde la pista número 6 hasta la ranura final, pero sin llegar a tocar la última pista. La moneda cae, tal y como refleja ese pico descendente de color amarillo (disminución del valor de  $y$ ), pero no se reconoce que haya acabado el juego, por lo que tras un tiempo la moneda se recoloca en la última pista para finalmente volver a caer. Esta situación es llamativa pero poco preocupante, puesto que se produce por las condiciones y peculiaridades del código, mientras que en un entorno real la moneda llegaría a la ranura final sin problema.

Del mismo modo, la gráfica que refleja como varía el giro del disco en esta primera dificultad parece un poco brusca. Esto es lógico, puesto que la caída de la moneda es más precipitada al ser las pistas más cortas, y por tanto el controlador se ve obligado a dar “volantazos” de forma más agresiva.

Dejando de lado ese detalle, en las otras tres dificultades la señal de control es bastante similar. Se pueden apreciar dos fases por cada pista:

- En la primera el giro disminuye de forma suave según la moneda llega al final de la pista.
- Al llegar al final y producirse la caída, el giro aumenta repentinamente, ya sea en sentido positivo o negativo, para invertir el sentido de avance de la moneda. En ese momento se repite el ciclo y el giro va disminuyendo de nuevo poco a poco.

Este resultado obtenido para el giro  $u$  es totalmente coherente con la gráfica de la variable  $e$  (línea verde de la gráfica izquierda). Ambas gráficas tienen la misma forma, lo que deja claro que, según la moneda se acerca al fin de la pista, el controlador estabiliza el giro del disco para que la caída sea poco brusca.

En cuanto a la velocidad, se puede apreciar que evoluciona de forma bastante suave durante la parte media de los carriles, sin alcanzar picos muy grandes. Sin embargo, es cierto que alcanza valores altos cuando llega al final de cada pista, fruto de la inercia acumulada por el aumento de la inclinación de la parábola. Tal vez se puedan modificar las reglas difusas para que sean mas estrictas y rectifiquen aún más el giro, evitando que la bola se acelere al llegar al final del carril.

Del mismo modo, una posible mejora sería modificar las reglas para invertir la evolución de la velocidad, de forma que sea alta durante el tramo medio de las pistas, pero que se reduzca casi a cero al llegar al final de éstas. Este hecho supondría una reducción en el tiempo de resolución, y probablemente también se lograría un caída más suave y segura de unas pistas a otras. El punto negativo de dicho modo de control sería la necesidad de dar “volantazos” al disco, es decir, se producirían cambios bruscos en el ángulo de giro para frenar en seco a la moneda antes de que caiga, hecho que puede suponer una desventaja desde el punto de vista mecánico.

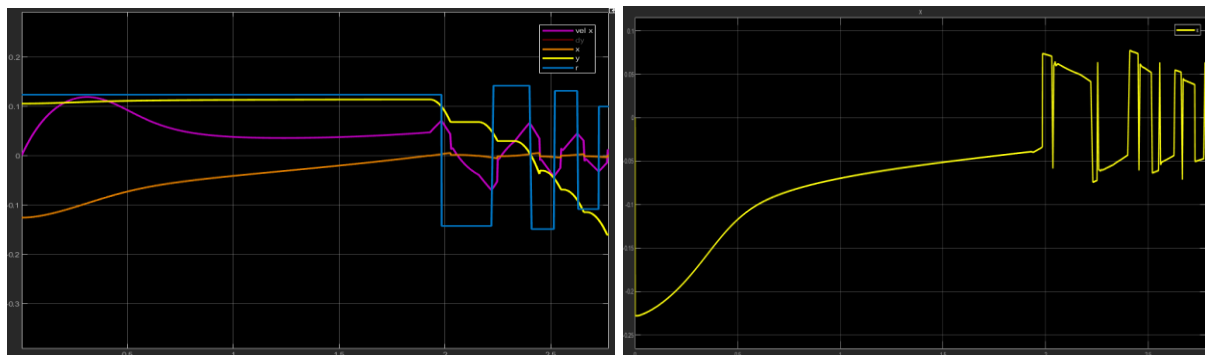
Por otra parte, como se puede observar en los videos, el sistema es bastante fiable, puesto que la moneda casi nunca se queda cerca de caer fuera del recorrido.

Consideramos que el tiempo de resolución es bastante bueno, con un tiempo máximo de 13.6 segundos para la última dificultad.

## 4.4. RESULTADOS EN MODELO PROPIO. CONCLUSIONES

### DIFICULTAD 1

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

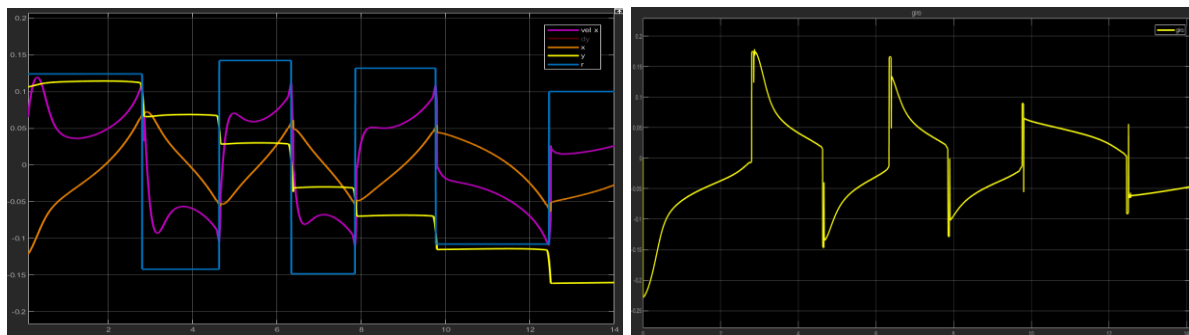


*Ilustración 19. Gráficas obtenidas para la dificultad 1 en el modelo propio*

Tiempo total: **2.78 segundos**

### DIFICULTAD 2

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

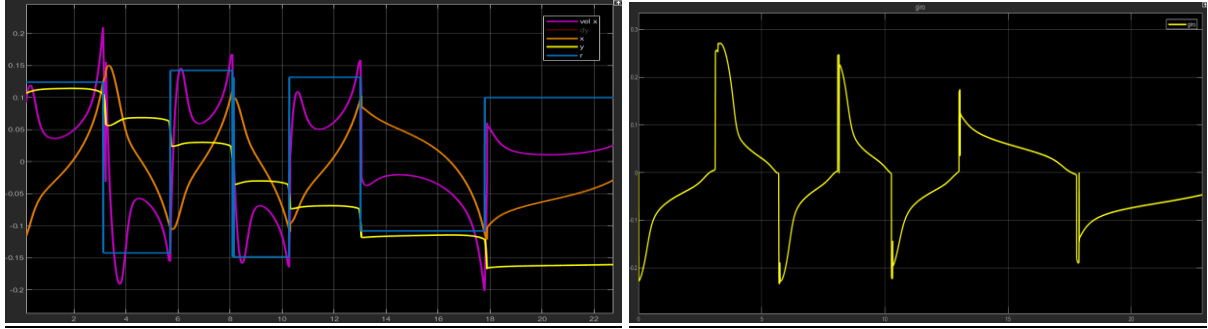


*Ilustración 20. Gráficas obtenidas para la dificultad 2 en el modelo propio*

Tiempo total: **14.100 segundos**

### DIFICULTAD 3

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

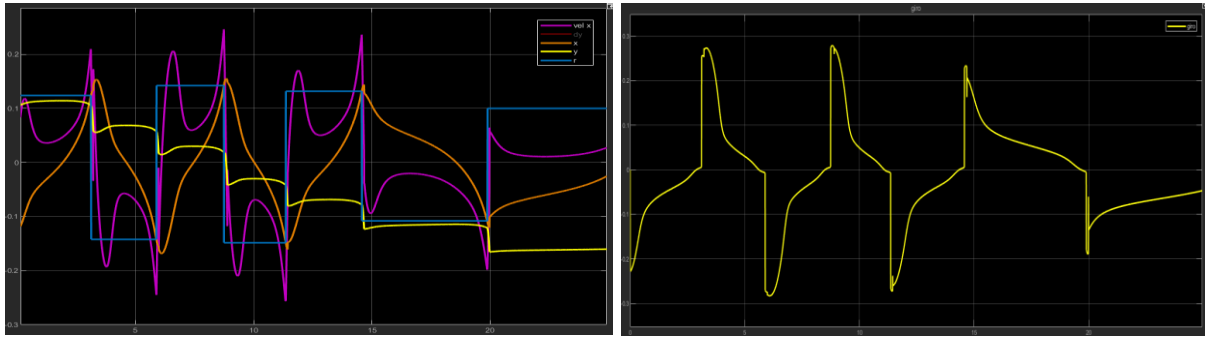


*Ilustración 21. Gráficas obtenidas para la dificultad 3 en el modelo propio*

Tiempo total: **22.89 segundos**

### DIFICULTAD 4

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



*Ilustración 22. Gráficas obtenidas para la dificultad 4 en el modelo propio*

Tiempo total: **24.95 segundos**

Con el modelo diseñado la ejecución es más lenta, pero igual de fiable y segura.

## **5. CONTROLADOR ADAPTATIVO**

Para el diseño del controlador adaptativo, el primer paso es linealizar las ecuaciones del movimiento que pueden ser controlables, es decir, el movimiento de la moneda por el carril. Según el análisis realizado al inicio, el movimiento del carril viene dado por la ecuación:

$$\ddot{x} = g \cdot \sin(\alpha + \theta) - \frac{\mu \cdot \dot{x}}{m} \quad (16)$$

Dado que el controlador solo va a tener en cuenta el desplazamiento en  $x$ , basta con linealizar la ecuación 16 en el punto  $x_0 = 0$  y  $\theta_0 = 0$ . Para simplificar el proceso, se aproxima el valor de  $x_{parabola}$  a  $x_{moneda} = x(t)$ .

$$\Delta \ddot{x} = \frac{\partial(g \cdot \sin(\alpha(x) + \theta))}{\partial x} \Delta x + \frac{\partial(g \cdot \sin(\alpha(x) + \theta))}{\partial \theta} \Delta \theta - \frac{\partial(\frac{\mu \cdot \dot{x}}{M})}{\partial \dot{x}} \Delta \dot{x} \quad (17)$$

$$\Delta \ddot{x} = A \Delta x + B \Delta \theta - C \Delta \dot{x} \quad (18)$$

$$G(s) = \frac{\Delta x}{\Delta \theta} = \frac{B}{s^2 + Cs - A} \quad (19)$$

La ecuación 19 es la transformada de Laplace de la ecuación 18. Suponiendo una masa de 7 gramos y un coeficiente de rozamiento de 0.01 donde cada termino es:

$$A = -\frac{1.08 \cdot g \cdot \cos(\theta_0 + \text{atan}(1.08 \cdot x_0))}{1.1664 \cdot x_0^2 + 1} = 10.58 \quad (20)$$

$$B = g \cdot \cos(\theta_0 + \text{atan}(1.08 \cdot x_0)) = -9.8 \quad (21)$$

$$C = \frac{\mu}{M} = \frac{0.01}{0.007} = 1.42 \quad (22)$$

El calculo de la derivada A se ha realizado mediante la herramienta Wólfram Alpha.

$$G(s) = \frac{\Delta x}{\Delta \theta} = \frac{B}{s^2 + Cs - A} = \frac{-9.8}{s^2 + 1.42s - 10.58} \quad (23)$$

Como se puede observar en la ecuación 23, la función de transferencia de la moneda cuando se desplaza por un carril es de segundo grado, por lo que resulta intuitivo usar un función de transferencia de segundo grado como modelo.

El siguiente paso es elegir un modelo de referencia  $G_m(s)$  que debe ser realizable por el Monza. Como modelo de referencia se ha escogido una planta de segundo orden subamortiguado con un factor de amortiguamiento relativo 0.8 y frecuencia angular natural rad/s 2.

Se ha diseñado un sistema subamortiguado para que la pelota caiga del carril. Por el contrario, los sistemas sobreamortiguados tienden a llegar a la referencia, pero la sobrepasan. Es decir, si la moneda sigue a la perfección el modelo de referencia, esta se quedaría atascada justo antes de caer al siguiente carril.

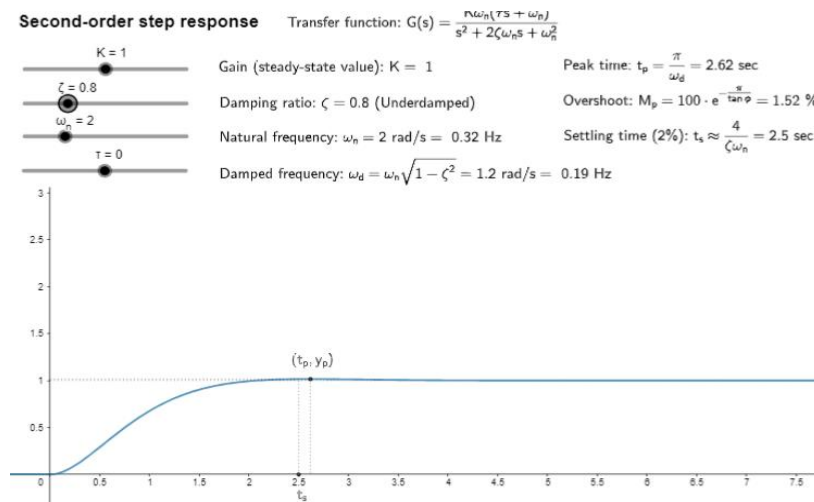


Ilustración 23. Sistema subamortiguado

$$G_m(s) = \frac{b_m}{s^2 + a_{1m}s + a_{2m}} = \frac{4}{s^2 + 3.2s + 4} \quad (24)$$

Para que el disco giratorio sea capaz de reproducir el modelo, se propone el controlador de la ecuación 24, donde  $\theta_1$  y  $\theta_2$  son los parámetros del controlador que serán optimizados usando la regla del MIT, de tal manera que el sistema reproduzca el modelo de referencia. La idea es que el valor de estos parámetros permita aproximar el modelo real al de referencia. Por último  $u_c$  es el setpoint.

$$u = \theta_1 u_c - \theta_2 y \quad (25)$$

La ley de adaptación de empleada es la MIT que tiene por objetivo ajustar los parámetros del controlador para minimizar una función de costo:

$$J(\theta) = \frac{1}{2} e^2(\theta) \rightarrow \frac{d\theta}{dt} = -\gamma \frac{dJ}{d\theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (26)$$

$$e = y - y_m = G(s)u - G_m(s)y \quad (27)$$

Donde  $\gamma$  es la ganancia de adaptación (positiva),  $e$  es el error de adaptación, “ $y$ ” es la salida de la planta, es decir, la posición referencial de la moneda en el Monza, y “ $y_m$ ” es la salida del modelo de referencia. Es importante indicar que esta regla no asegura ni la estabilidad ni la convergencia como si puede hacer un diseño usando la ley de Lyapunov. Es conveniente mantener una ganancia de adaptación baja para que el sistema no sea inestable debido a los sobrepicos que puede provocar el controlador. Dada esta sensibilidad del sistema al valor de la ganancia de adaptación, es crítico ajustarla correctamente para que el sistema sea estable. Este proceso se hará de forma experimental.

$$y = G(s)u = \frac{-9.8}{s^2 + 1.42s - 10.58} (\theta_1 u_c - \theta_2 y) \quad (28)$$

Despejando la “ $y$ ” de la ecuación 28 se obtiene la siguiente expresión:

$$y = G(s)u = \frac{-9.8 \cdot \theta_1}{s^2 + 1.42s - 10.58 - 9.8\theta_2} \cdot u_c \quad (29)$$

$$e = \frac{-9.8 \cdot \theta_1}{s^2 + 1.42s - 10.58 - 9.8\theta_2} \cdot u_c - G_m u_c \quad (30)$$

De esta expresión podemos obtener el error  $e$ , y derivándolo respecto a  $\theta_1$  y  $\theta_2$  se obtienen las reglas de adaptación:

$$\frac{\partial e}{\partial \theta_1} = \frac{-9.8}{s^2 + 1.42s - 10.58 - 9.8\theta_2} \quad (31)$$

$$\frac{d\theta_1}{dt} = -\gamma e \frac{\partial e}{\partial \theta_1} \quad (32)$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{-9.8 \cdot \theta_1}{s^2 + 1.42s - 10.58 - 9.8\theta_2} u_c = -\frac{-9.8 \cdot \theta_1}{s^2 + 1.42s - 10.58 - 9.8\theta_2} y \quad (33)$$

$$\frac{d\theta_2}{dt} = -\gamma e \frac{\partial e}{\partial \theta_2} \quad (34)$$

Suponiendo que el modelo real con la implementación de  $\theta_1$  y  $\theta_2$  es muy cercano al modelo de referencia, se puede hacer la siguiente aproximación:

$$s^2 + 1.42s - 10.58 - 9.8\theta_2 \sim s^2 + a1_ms + a2_m \quad (35)$$

$$\frac{\partial e}{\partial \theta_1} = \frac{a1_m + a2_m}{s^2 + a1_ms + a2_m} u_c \quad (36)$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{a1_m + a2_m}{s^2 + a1_ms + a2_m} y \quad (37)$$

Con esta suposición se puede construir el controlador con el diseño mostrado en el siguiente apartado.

## 5.1. DIAGRAMA EN SIMULINK

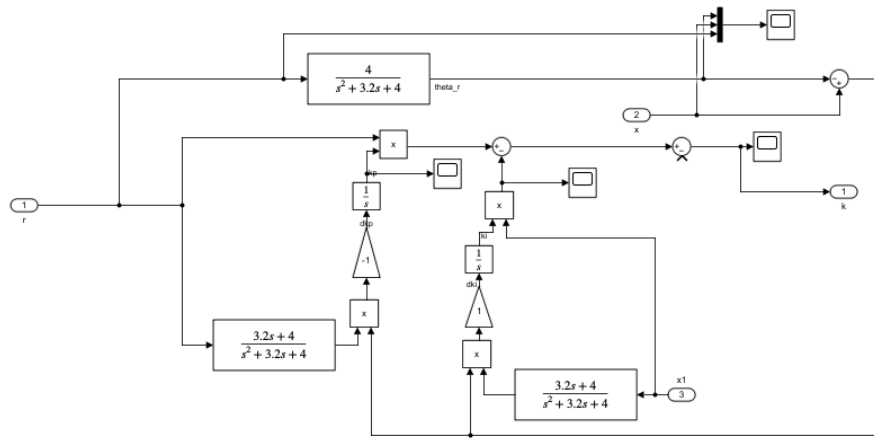


Ilustración 24. Diagrama de bloques en Simulink del controlador adaptativo

Ajustando la ganancia de adaptación a 1 y con valor inicial 1 para  $\theta_1$  y  $\theta_2$ , se obtiene un control muy agresivo. En la ilustración 25 se muestra la evolución de la moneda en el eje de coordenadas  $x$  referencial. Como se puede apreciar, su evolución llega a ser más agresiva incluso que el modelo de referencia.

Esta situación provoca que el controlador no sea capaz de controlar adecuadamente la velocidad de la misma, haciendo que el proceso de recepción sea menos robusto. En el caso de la ilustración 30 (dificultad 4) el controlador es incapaz de amortiguar la caída de la bola en el 4º carril.

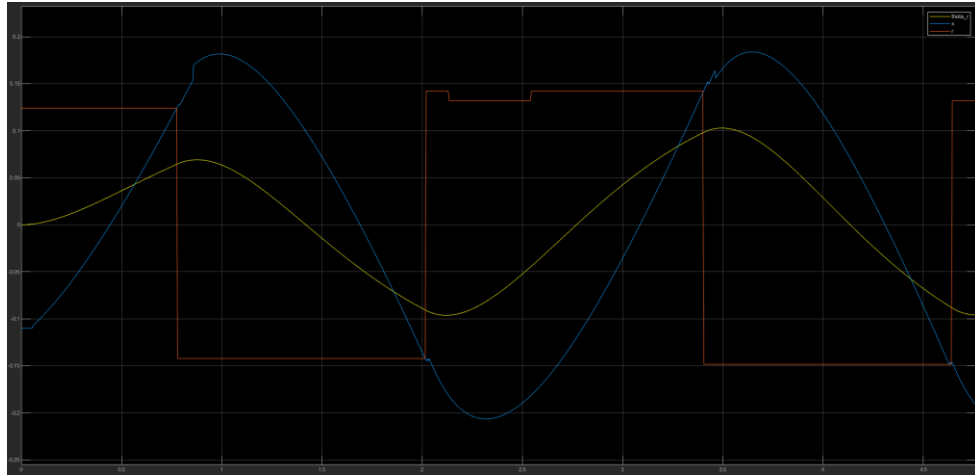


Ilustración 25

## 5.2. MODIFICACIÓN

Para solucionar el inconveniente de la velocidad, se propone añadir un control PD a la estructura de control. La función de este bloque es añadir esa componente derivativa que puede penalizar una evolución muy rápida del sistema. De esta forma la velocidad de la moneda a lo largo del carril será menor, por lo que la recepción será más suave y controlada.

Esta implementación provoca que el sistema sea más lento para completar el juego, pero también hace más robusto al controlador.

Tras varias pruebas experimentales, se llegó al siguiente ajuste:  $P = 0.2$ ;  $D = 0.05$ ;  $N = 10$

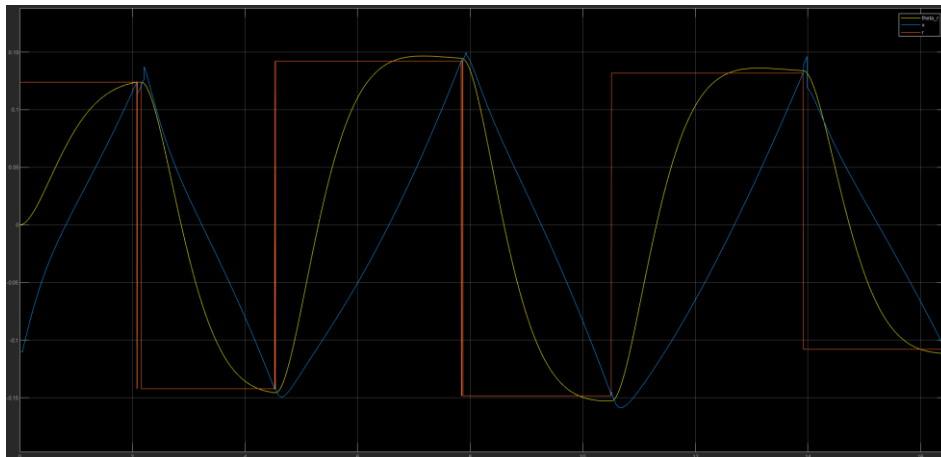


Ilustración 26.

En la ilustración 26 se muestra la evolución del sistema tras implementar el PD. Comparando con la ilustración 25 se observa lo comentado anteriormente. Ahora el sistema es más lento, pasa de tardar 4 segundos para llegar al 4º a tardar 12; es 3 veces más lento. Por otro lado, se observa que el movimiento de la moneda es más suave y lento. Mientras que en la ilustración 25 se ve que la bola llega hasta los 18 cm de recorrido, el nuevo controlador apenas supera los 16 cm.



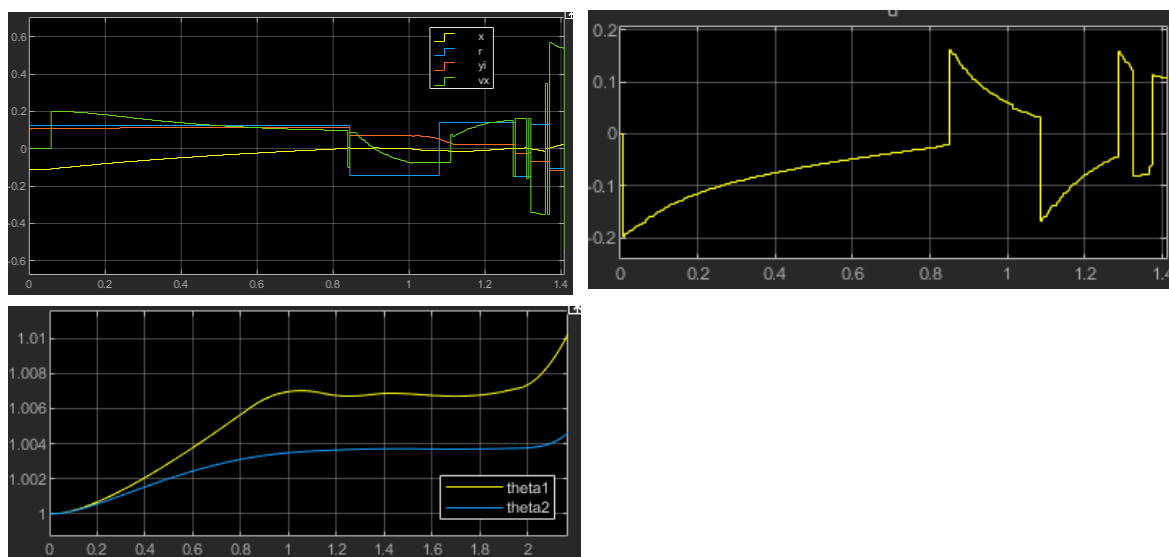
### 5.3. RESULTADOS EN MODELO DADO. CONCLUSIONES

Siguiendo la misma estructura que la planteada para el control difuso, se muestra la evolución de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

#### DIFICULTAD 1

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



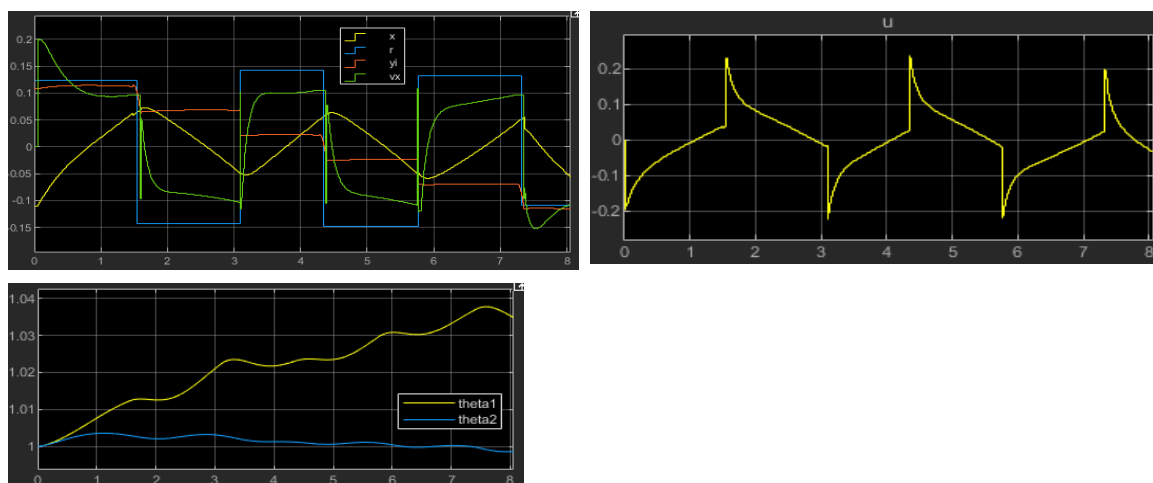
*Ilustración 27. Gráficas obtenidas para la dificultad 1 en el modelo proporcionado*

Tiempo total: **1.4 segundos**

#### DIFICULTAD 2

Video con la performance del controlador: [LINK AQUI](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



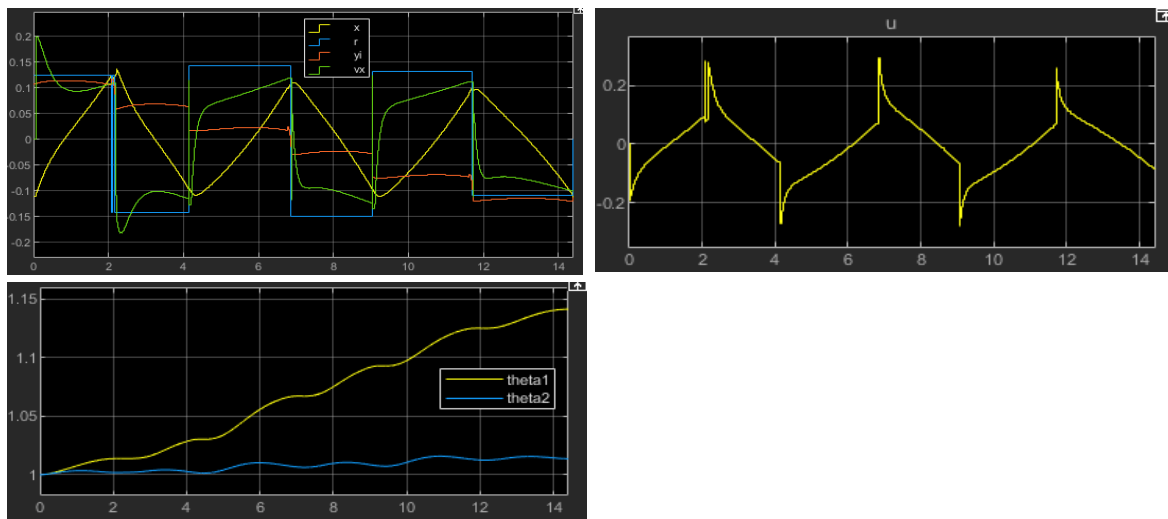
*Ilustración 28. Gráficas obtenidas para la dificultad 2 en el modelo proporcionado*

Tiempo total: **8 segundos**

### DIFICULTAD 3

Video con la performance del controlador: [LINK YOUTUBE](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



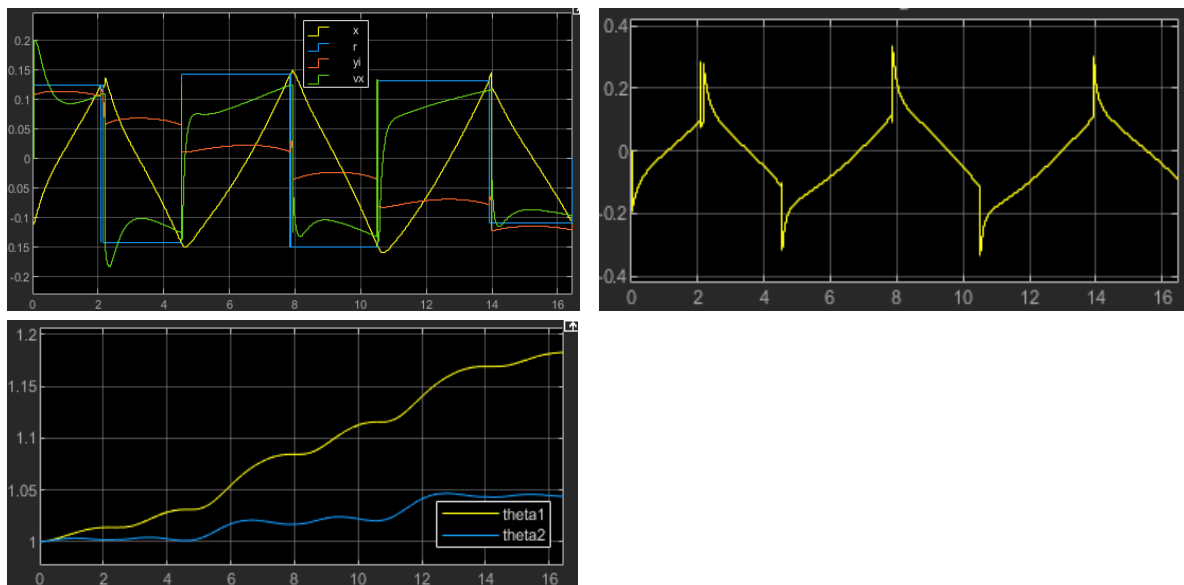
*Ilustración 29. Gráficas obtenidas para la dificultad 3 en el modelo proporcionado*

Tiempo total: **14.2 segundos**

### DIFICULTAD 4

Video con la performance del controlador: [LINK YOUTUBE](#)

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



*Ilustración 30. Gráficas obtenidas para la dificultad 4 en el modelo proporcionado*

Tiempo total: **16.3 segundos**

Como se puede apreciar, el control adaptativo con el PD es capaz de superar todas las dificultades. Observando las ilustraciones que muestran la velocidad y posición de la moneda durante el circuito y el input del controlador, se puede hacer una valoración de su funcionamiento.

En resumen, cada vez que la moneda cae a un nuevo carril, el disco da un “volantazo” rápido para recibir la moneda y agregar una velocidad inicial con la que moverse por la pista hasta el nuevo carril.

Este hecho se ve reflejado en la evolución de la velocidad de la moneda. Ésta llega a los finales de carriles con una velocidad que va creciendo con el tiempo, y al poco de caer cambia por completo su dirección. También se aprecia el breve recorrido que tiene que hacer la moneda hasta que cambia de dirección, claramente gracias a la lentitud con la que llega al final de la pista.

Respecto a la señal de control, se pueden diferenciar claramente dos fases: la fase de movimiento de la moneda por el carril y la fase de recibimiento. Mientras que en la primera fase el output del controlador es suave, en la segunda este se vuelve más agresivo para controlar rápidamente la moneda. Se observa que existe un escalón entre el ángulo del disco antes de la caída de la moneda y el ángulo cuando ésta ya está abajo.

Por lo general el escalón tiene un tamaño de 0,2 radianes, lo que equivale a inclinar el Monza 10° aproximadamente. Esta dinámica podría ser un problema para el motor que esté efectuando este movimiento. En nuestro caso, entendemos que dentro del dispositivo hay un pequeño servomotor que recibe la posición de referencia que debe alcanzar (output del controlador,  $u$ ). Dependiendo de las características de dicho motor, podría ser algo problemático.

En cualquier caso, creemos que no habría mucho problema en realizar dicho movimiento más allá de las propias limitaciones de movimiento del servomotor. También partimos de la suposición de que el disco es muy ligero y que el servo no tiene que aplicar un momento que le haga saturar.

En cuanto a los parámetros de adaptación, se ve una dinámica que tiende a crecer para los dos parámetros. En el caso de  $\theta_1$ , cuanto más aumenta, mayor es la ganancia del sistema. En cuanto a  $\theta_2$ , a mayor frecuencia natural del sistema, mayor es la rapidez con la que evoluciona el sistema. Llama la atención que la evolución de los parámetros no aproxima la función de la planta a la función modelo, sino que más bien se están alejando de ella.

Esto puede ser debido a una incorrecta linealización del sistema de ecuaciones.

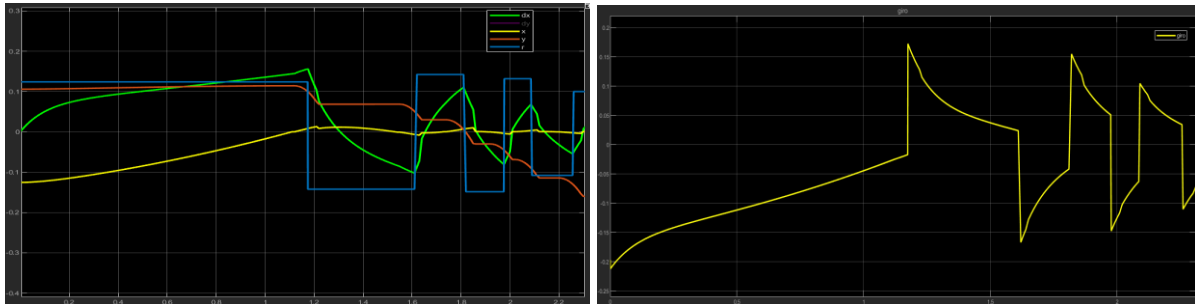
$$y = G(s)u = \frac{-9.8 \cdot \theta_1}{s^2 + 1.42s - 10.58 - 9.8\theta_2} \cdot u_c \quad (38)$$

A pesar de ello, el sistema es estable y logra resolver todos los niveles del juego.

## 5.4. RESULTADOS EN MODELO PROPIO. CONCLUSIONES

### DIFICULTAD 1

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

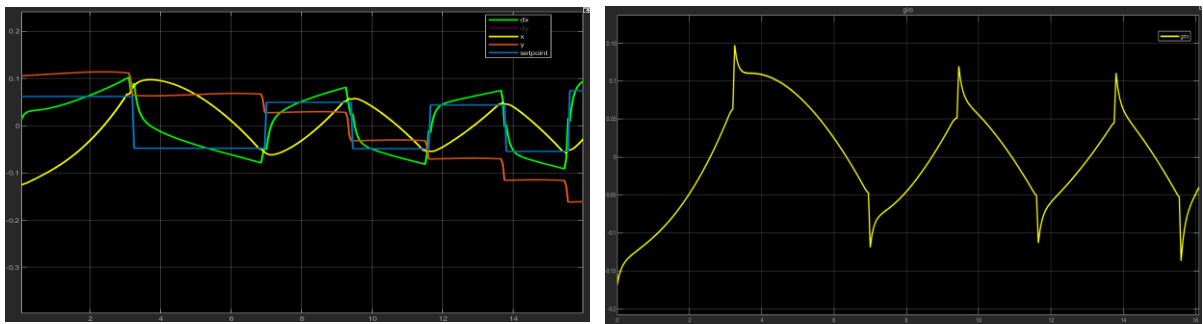


*Ilustración 31. Gráficas obtenidas para la dificultad 1 en el modelo propio*

Tiempo total: **2.31 segundos**

### DIFICULTAD 2

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

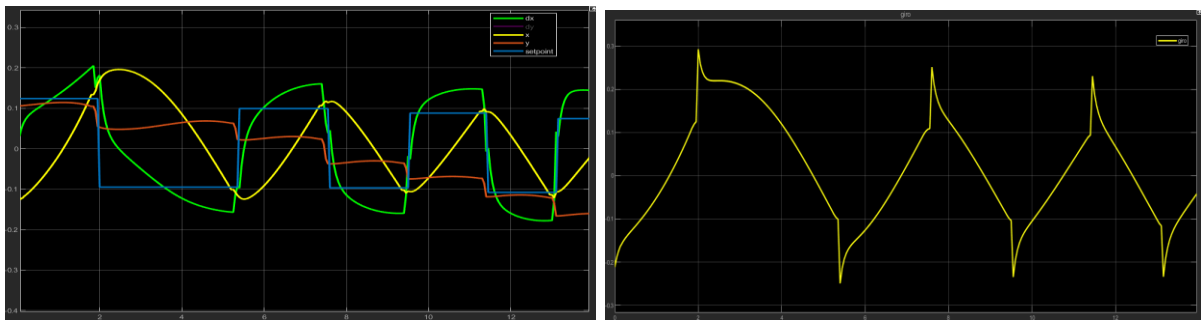


*Ilustración 32. Gráficas obtenidas para la dificultad 2 en el modelo propio*

Tiempo total: **7.95 segundos**

### DIFICULTAD 3

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :

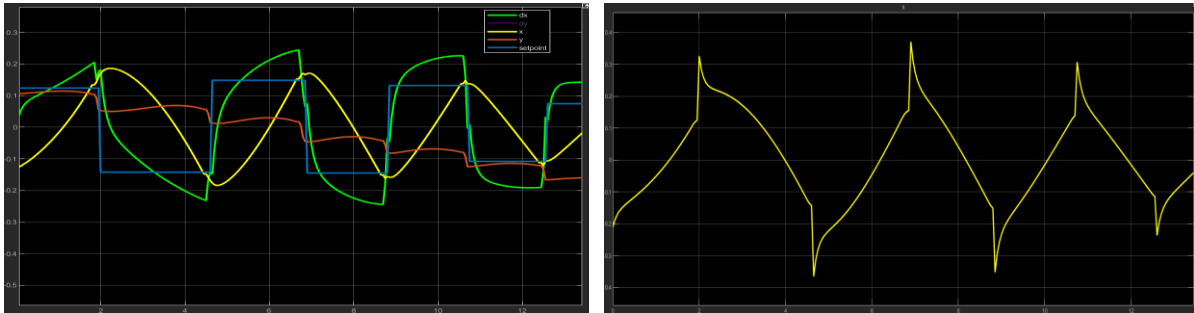


*Ilustración 33. Gráficas obtenidas para la dificultad 3 en el modelo propio*

Tiempo total: **11.29 segundos**

#### DIFICULTAD 4

Evolución temporal de las variables  $x$ ,  $y$ ,  $e$ ,  $r$ ,  $Velx$  y del ángulo de giro  $u$ :



*Ilustración 34. Gráficas obtenidas para la dificultad 4 en el modelo propio*

Tiempo total: **12.74 segundos**

Como ambos modelos son muy parecidos, los resultados son prácticamente iguales, y por tanto las conclusiones que sacamos son las mismas que en el apartado anterior.