
Problema del péndulo invertido utilizando Lógica Borrosa

Universidad de Murcia
4º Grado en Ingeniería Informática
Asignatura: Desarrollo De Sistemas Inteligentes
Curso 2019/2020

Proyecto realizado por Nicolás Enrique Linares La Barba.

Índice

Introducción.....	3
Implementación.....	4
Resultados.....	5
Bibliografía.....	8

Introducción

La estabilización de un péndulo invertido colocado en un carrito que se puede mover hacia adelante y hacia atrás es una analogía del problema de estabilizar un palo colocado sobre la punta de un dedo. El sistema formado por el péndulo invertido es inherentemente inestable que requiere de la aplicación de técnicas de control y ha sido utilizado en muchos trabajos para evaluarlas.

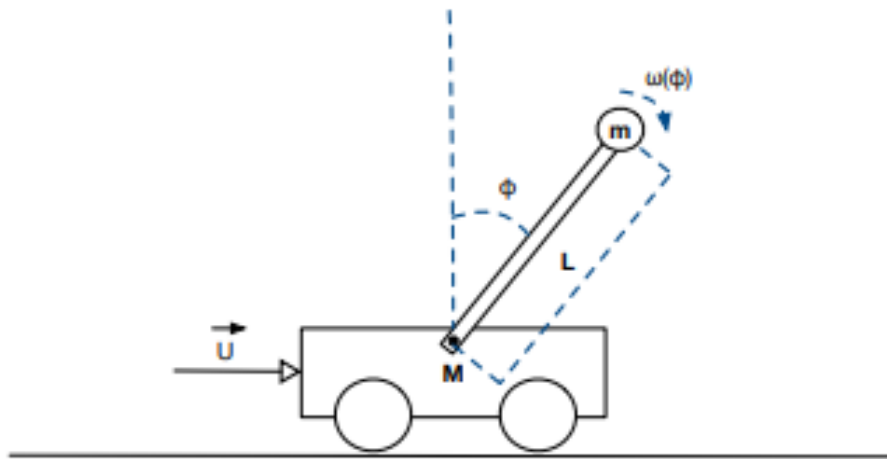


Figura 1. Diagrama del péndulo invertido sobre un carrito.

En la figura 1 se puede apreciar una aproximación al problema. Básicamente, se trata de conseguir estabilizar el péndulo en posición vertical ($\phi = 0^\circ$) o, al menos, que no caiga. Para ello, se parte de la información proporcionada por los sensores que nos indican en todo momento cuál es el ángulo en el que se encuentra el péndulo (medidos en grados en nuestro caso) y su velocidad angular (medida en rad/seg).

Con esta información, el sistema de control debe calcular la fuerza que hay que aplicar al carrito (en la figura 1 una fuerza positiva movería el carrito a la derecha y una fuerza negativa a la izquierda) para conseguir el objetivo final.

Implementación

Se ha implementado el sistema descrito con JFuzzyLogic y donde se mostrarán los resultados de la estabilización del mismo a través de gráficas que actualizan en tiempo real el ángulo del péndulo y la velocidad angular.

En el fichero **tipper.fcl** se han añadido todos los datos presentes en el enunciado para la implementación del lenguaje del sistema de inferencia lógica de este problema.

La implementación que se encarga de evaluar continuamente los datos del estado del péndulo se encuentra en el fichero **TestTipper.java**. Se ha incluido un apartado donde se indican los valores que se pueden modificar del programa para observar distintas ejecuciones o comportamientos del péndulo:

```
//#####  
// Valores modificables del programa  
  
//Parámetros del péndulo  
double angulo_barra_grados = 10;  
double velocidad_angular_barra_grados = 30;  
double angulo_estabilizado = 1;  
  
// Parámetros de la base  
double posicionX_base = 0;  
double velocidad_base = 0;  
  
//#####
```

Es importante destacar dos variables que se han añadido en el estado del péndulo y una función que se encarga de comprobar cuando el péndulo se considera estabilizado.

Mediante la variable modificable “angulo_estabilizado” que se ve en el código anterior, se indica cuando el péndulo se considera estabilizado, en ese caso, cuando el ángulo máximo de oscilación es de 1 grado. Esto se ha añadido para evitar una ejecución infinita del programa, ya que continúa oscilando en valores muy pequeños.

En cuanto a la función que comprueba este valor de estabilización, se ha implementado en la clase Estado.java, siguiendo la siguiente idea: cuando se alcance el ángulo máximo de oscilación se comprobará si ese valor es menor a un valor (variable “angulo_estabilizado”), por ejemplo, si oscila solo un grado podemos considerar que se ha estabilizado.

Se ha hecho uso de la librería open source XChart para poder realizar los gráficos en tiempo real para analizar el comportamiento de las variables.

Resultados

En cuanto a la observación de resultados, se puede analizar el comportamiento del péndulo a través de las dos gráficas que se actualizan en tiempo de ejecución para el ángulo y para la velocidad angular del péndulo. Por ejemplo, para el caso anterior indicado en el código, se obtiene las gráficas que podemos observar en la figura 2 y 3.

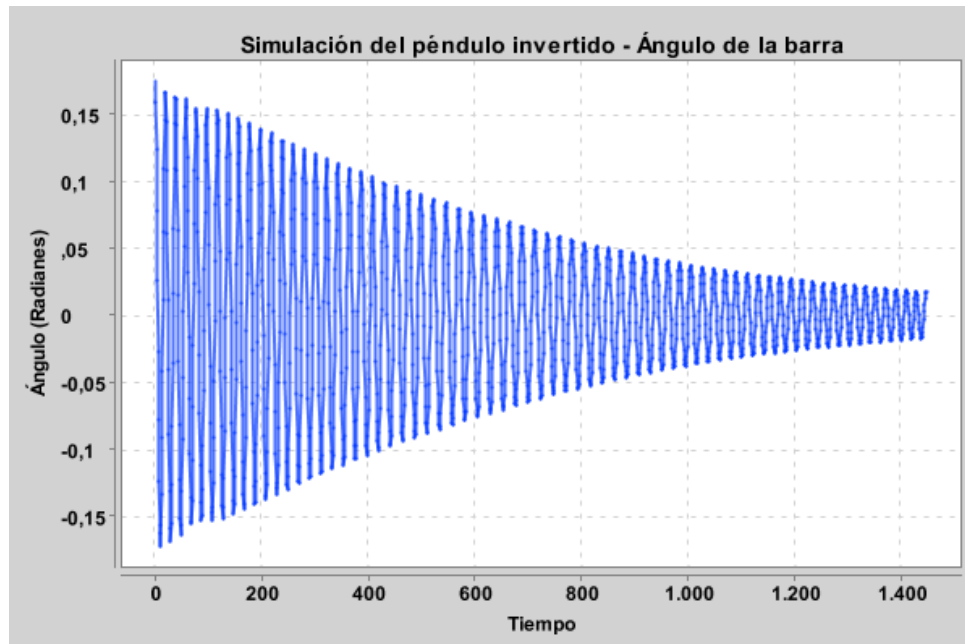


Figura 2. Simulación del péndulo (ángulo de la barra).

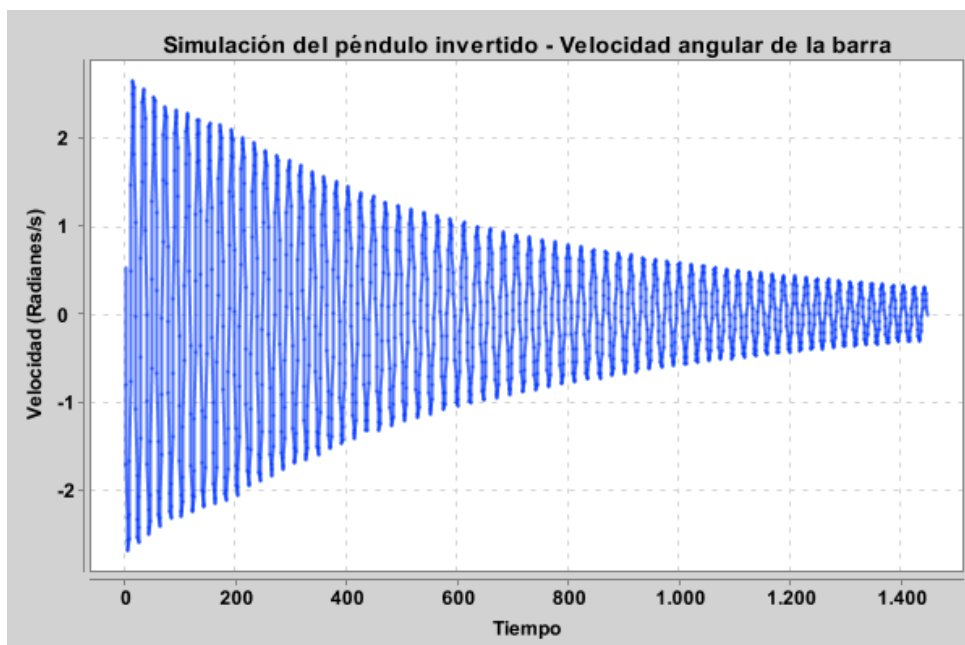


Figura 3. Simulación del péndulo (velocidad angular de la barra).

Como se aprecia en las gráficas los valores tienden a 0 a medida que avanza el tiempo, lo que indica una estabilización total del péndulo. La ejecución se ha parado según se ha explicado anteriormente.

Además, se indica en la terminal de Eclipse como se van actualizando las variables del SIB y seguidamente los valores del estado del péndulo, siendo esta la última salida:

```
#####
angulo :
    Value: 0.9835153178124078
    Term: pb      0.0    PieceWiseLinear : (5.0, 0.0) , (10.0, 1.0) , (70.0,
1.0) , (70.0, 0.0) ;
    Term: ps      0.19670306356248154 PieceWiseLinear : (0.0, 0.0) , (5.0,
1.0) , (10.0, 0.0) ;
    Term: nb      0.0    PieceWiseLinear : (-70.0, 0.0) , (-70.0, 1.0) , (-
10.0, 1.0) , (-5.0, 0.0) ;
    Term: ns      0.0    PieceWiseLinear : (-10.0, 0.0) , (-5.0, 1.0) ,
(0.0, 0.0) ;
    Term: z       0.8032969364375184 PieceWiseLinear : (-5.0, 0.0) , (0.0,
1.0) , (5.0, 0.0) ;
velocidad angular :
    Value: 0.07090381538686963
    Term: pb      0.0    PieceWiseLinear : (8.0, 0.0) , (200.0, 1.0) ,
(500.0, 1.0) , (500.0, 0.0) ;
    Term: ps      0.008862976923358704    PieceWiseLinear : (0.0, 0.0) ,
(8.0, 1.0) , (200.0, 0.0) ;
    Term: nb      0.0    PieceWiseLinear : (-500.0, 0.0) , (-500.0, 1.0) ,
(-200.0, 1.0) , (-8.0, 0.0) ;
    Term: ns      0.0    PieceWiseLinear : (-200.0, 0.0) , (-8.0, 1.0) ,
(0.0, 0.0) ;
    Term: z       0.9911370230766413 PieceWiseLinear : (-8.0, 0.0) , (0.0,
1.0) , (8.0, 0.0) ;
fuerza :
    Defuzzifier : CenterOfGravity
    Latest defuzzified value: 13.477110343863226
    Default value: 0.0
    Term: pb      0.0    PieceWiseLinear : (190.0, 0.0) , (200.0, 1.0) ,
(210.0, 0.0) ;
    Term: ps      0.0    PieceWiseLinear : (30.0, 0.0) , (50.0, 1.0) ,
(70.0, 0.0) ;
    Term: nb      0.0    PieceWiseLinear : (-210.0, 0.0) , (-200.0, 1.0) ,
(-190.0, 0.0) ;
    Term: ns      0.0    PieceWiseLinear : (-70.0, 0.0) , (-50.0, 1.0) , (-
30.0, 0.0) ;
    Term: z       0.3261444828068387 PieceWiseLinear : (-20.0, 0.0) , (0.0,
1.0) , (20.0, 0.0) ;
-----
Angulo (grados): 0.9622453950106249
Velocidad angular (rads/s): -0.01856150922680648
fuerza (N/m/s): 13.477110343863226
Posición de la base (metros): 11.05896411350304
Velocidad de la base (metros/s): 0.38879244716497857
#####
EL PÉNDULO SE HA ESTABILIZADO
Se considera estabilizado cuando el ángulo de oscilación es menor que 1.0
grado
```

La salida por terminal anterior muestra como el péndulo ya ha llegado a un ángulo de oscilación menor a 1 grado por lo que se termina la ejecución. Su velocidad angular también es casi nula, pero se sigue haciendo fuerza en la base ya que no está 100% estabilizado, por eso mismo también se encuentra moviéndose como vemos con la posición de la base y su velocidad.

Se grafica también las funciones de pertenencia para cada variable de entrada y de salida. Podemos verlo en la figura 4.

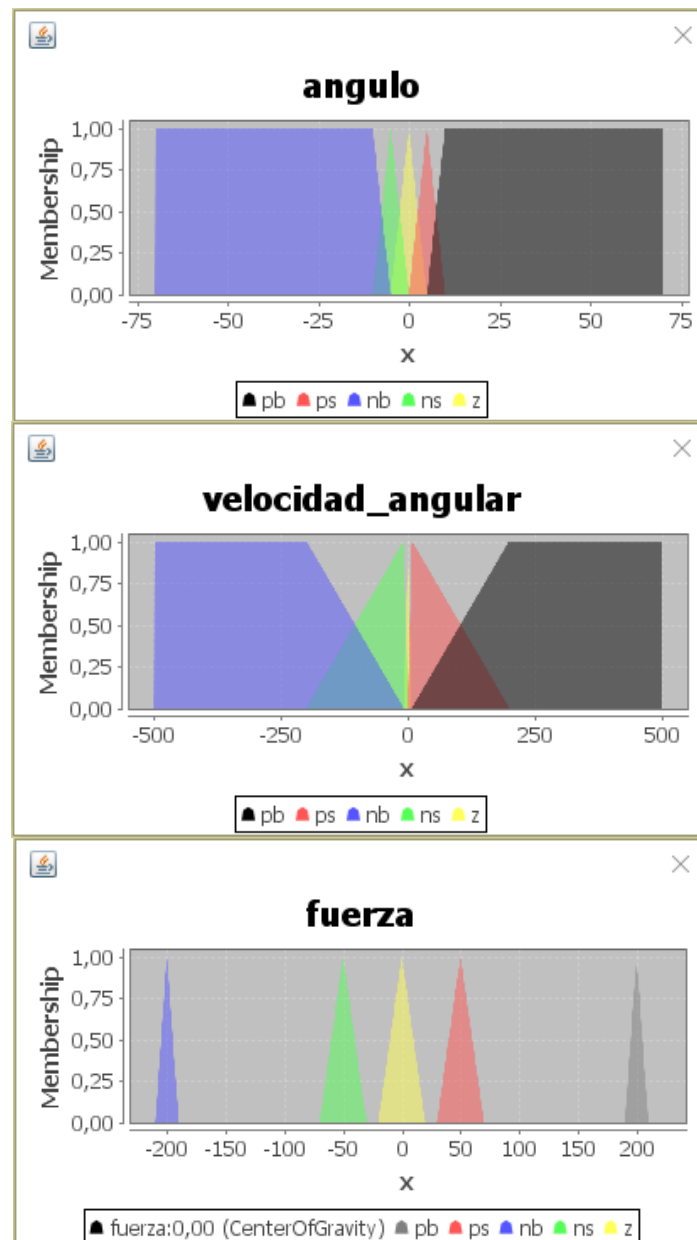


Figura 4. Funciones de pertenencia.

Bibliografía

- Inteligencia Artificial: Técnicas, métodos y aplicaciones. – José Tomás Palma Méndez y Roque Marín Morales.
- Manual JFuzzyLogic: <http://jfuzzylogic.sourceforge.net/html/manual.html>
- Manual XChart: <https://github.com/knowm/XChart>