

REPORTE TAREA #6

INTELIGENCIA

COMPUTACIONAL

línea horizontal

Jesús Rivera Flores - A00820643

Diego Arnoldo Azuela Rosas - A01208345



**CONTENIDO**

# 

[**INSTRUCCIONES**](#_cixyks6g2g16) **3**

[NOMBRE](#_49k4nmgv0c6q) 3

[OBJETIVO](#_hs0x2q2hce2) 3

[DEFINICIÓN DEL PROBLEMA](#_9sqi88wywaha) 3

[SOLUCIÓN](#_fvocu6wckn95) 3

[**DESCRIPCIÓN DEL PROBLEMA**](#_kyv7po6fj888) **3**

[DESRIPCIÓN](#_8rly1yf7t58s) 3

[**DESCRIPCIÓN DEL SISTEMA DIFUSO**](#_xo9m9qrh7icf) **3**

[INSTRUCCIONES](#_ef7md1uaciz6) 4

[INTERPRETACIÓN](#_kobyc1nr8ba) 4

[**GRÁFICAS**](#_hts6begpcy6a) **5**

[GRÁFICAS DE VARIABLES](#_piiftdbcx1me) 5

[**CONCLUSIÓN Y RETOS ENCONTRADOS**](#_erdvdg66zgyw) **7**

[**BIBLIOGRAFÍAS**](#_m0fr61w9wsj4) **9**

# 

# 

# INSTRUCCIONES

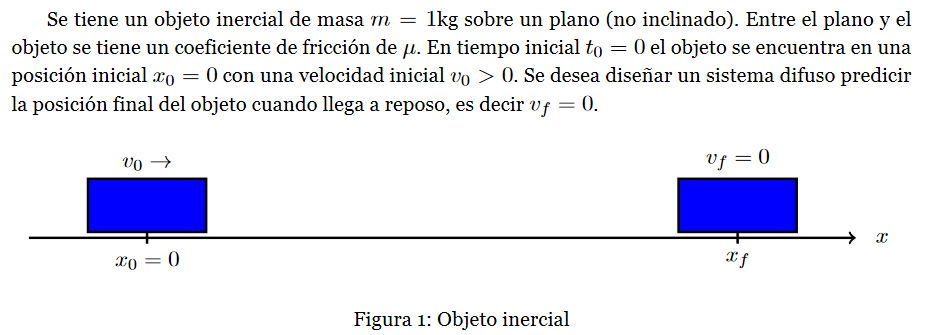
## NOMBRE

* Actividad: Control difuso

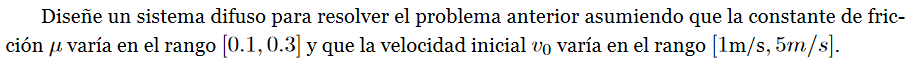
## OBJETIVO

* En este proyecto se desarrollará y se optimizará un sistema difuso para predecir el movimiento de un objeto inercial en un plano con fricción.

## DEFINICIÓN DEL PROBLEMA



## SOLUCIÓN



# DESCRIPCIÓN DEL PROBLEMA

## DESRIPCIÓN

* El problema planteado consta de tener una masa de velocidad “x” en un plano. La masa empieza en “X0” y llegará a “Xf”, se sabe que existen parámetros limitantes para el sistema. No se puede esperar que el objeto de masa “m” siga su trayectoria de manera infinita, esto dado que existe la variable de coeficiente de fricción. De esta manera sabemos que existe un límite para la distancia que puede recorrer el objeto, de la misma manera se sabe que el objeto tendrá un límite inferior. Dado que se tiene una velocidad mayor a 0, el objeto no retrocederá del punto de partida.
* Estos son los parámetros que le dicen al usuario que existe un “piso” y un “techo” para la posición posible del objeto. A partir de ahí el usuario debe de tomar en cuenta el resto de las variables para definir de manera acertada el rango en donde puede estar el objeto al final de su trayectoria. Se tienen diferentes variables y todas ellas tienen un “rango” como posible respuesta. Por lo mismo, se tendrá un “rango” de posibles respuestas. Es aquí en donde entra la lógica difusa, resolviendo problemáticas en donde la solución depende de variables/rangos varios.

# DESCRIPCIÓN DEL SISTEMA DIFUSO

## INSTRUCCIONES

## 

## INTERPRETACIÓN

* De acuerdo a lo establecido por Eduardo Estrada en su investigación “*SISTEMAS DE CONTROL DIFUSO”* un sistema de control difuso es una de las aplicaciones más interesantes de la lógica difusa. Los sistemas de control difuso, como el presente en esta actividad, se componen por entradas asociadas a parámetros vagos (conjuntos difusos), bases de datos que gobiernan el comportamiento del sistema y un compendio de reglas que serán evaluadas por un motor de inferencia para, a partir de los valores de entrada, generar los valores de salida pertinentes.
* En resumen, un sistema de control difuso se encuentra compuesto por 3 parámetros, todos de suma importancia a la hora de computar las respuestas. Es necesario que cada una de las reglas se exprese de manera resumida, dado que se pueden tomar 2 reglas y combinar para evitar redundancias. En este problema en particular las entradas son la ***velocidad inicial*** y el ***coeficiente de fricción***. La ***masa*** es una constante no importante. De la misma manera, entre más reglas simplificadas se tengan en el sistema, la salida tendrá un mejor comportamiento.

| **VARIABLES DE ENTRADA** | # INPUT VARIABLES: Velocity, Coefficient of friction  velocidad = ctrl.Antecedent(np.arange(1, 6, 1), 'velocidad')  friccion = ctrl.Antecedent(np.arange(0.1, 0.4, 0.1), 'friccion')  posicion = ctrl.Consequent(np.array([0.1020408163265306, 0.12755102040816324, 0.17006802721088435, 0.2551020408163265, 0.4081632653061224, 0.510204081632653, 0.510204081632653, 0.6802721088435374, 0.9183673469387754, 1.020408163265306, 1.1479591836734693, 1.530612244897959, 1.6326530612244896, 2.040816326530612, 2.040816326530612, 2.2959183673469385, 2.5510204081632653, 2.7210884353741496, 3.1887755102040813, 4.081632653061224, 4.251700680272109, 4.591836734693877, 6.377551020408163, 8.163265306122447, 12.755102040816325]), 'posicion')  # Auto-membership function population is possible with .automf(3, 5, or 7)  velocidad.automf(5)  friccion.automf(5)  posicion.automf(5) |
| --- | --- |

* A continuación se presentarán las reglas con las cuales el usuario podrá determinar el rango de respuesta correspondiente.

| **REGLAS A IMPLEMENTAR** | #Set up the necessary rules  rule5 = ctrl.Rule(velocidad['poor'] , posicion['poor'])  rule6 = ctrl.Rule(velocidad['mediocre'] & friccion['poor'], posicion['mediocre'])  rule7 = ctrl.Rule(velocidad['mediocre'] & friccion['mediocre'], posicion['poor'])  rule8 = ctrl.Rule(velocidad['mediocre'] & friccion['average'], posicion['poor'])  rule9 = ctrl.Rule(velocidad['mediocre'] & friccion['decent'], posicion['poor'])  rule10 = ctrl.Rule(velocidad['mediocre'] & friccion['good'], posicion['poor'])  rule11 = ctrl.Rule(velocidad['average'] & friccion['poor'], posicion['mediocre'])  rule12 = ctrl.Rule(velocidad['average'] & friccion['mediocre'], posicion['mediocre'])  rule13 = ctrl.Rule(velocidad['average'] & friccion['average'], posicion['poor'])  rule14 = ctrl.Rule(velocidad['average'] & friccion['decent'], posicion['poor'])  rule15 = ctrl.Rule(velocidad['average'] & friccion['good'], posicion['poor'])  rule16 = ctrl.Rule(velocidad['decent'] & friccion['poor'], posicion['decent'])  rule17 = ctrl.Rule(velocidad['decent'] & friccion['mediocre'], posicion['mediocre'])  rule18 = ctrl.Rule(velocidad['decent'] & friccion['average'], posicion['mediocre'])  rule19 = ctrl.Rule(velocidad['decent'] & friccion['decent'], posicion['mediocre'])  rule20 = ctrl.Rule(velocidad['decent'] & friccion['good'], posicion['poor'])  rule21 = ctrl.Rule(velocidad['good'] & friccion['poor'], posicion['good'])  rule22 = ctrl.Rule(velocidad['good'] & friccion['mediocre'], posicion['average'])  rule23 = ctrl.Rule(velocidad['good'] & friccion['average'], posicion['mediocre'])  rule24 = ctrl.Rule(velocidad['good'] & friccion['decent'], posicion['mediocre'])  rule25 = ctrl.Rule(velocidad['good'] & friccion['good'], posicion['mediocre']) |
| --- | --- |
| **DISPLAY DE LAS REGLAS** | rule5.view() |

# 

# GRÁFICAS

## GRÁFICAS DE VARIABLES



| **GRÁFICAS** | velocidad['average'].view() |
| --- | --- |
| friccion.view() |
| posicion.view() |

# 

# CONCLUSIÓN Y RETOS ENCONTRADOS

* A continuación se despliega la porción inicial y final del código. Una de las partes más importantes del código es el desarrollo del mismo, una vez que se tienen todas las reglas declaradas el usuario las comprime dentro de “tripping\_ctrl” para hacer uso de ellas con la función incorporada de “ctrl.ControlSystemSimulation”.
* Esta práctica es de gran uso en la industria. Se tienen gran variedad de aplicaciones y le da al usuario una gran ayuda visual para resolver problemáticas multivariables que de otra manera pudiera no ser tan fácil resolver o de las cuales es complicado obtener un aproximado.
* Se cree que se obtuvieron resultados bastante favorables que demuestran el uso adecuado de los algoritmos.

| **DESARROLLO DEL CÓDIGO** | |
| --- | --- |
| **LIBRERÍAS** | # LIBRARIES NEEDED  import numpy as np  import skfuzzy as fuzz  from skfuzzy import control as ctrl |
| **DESARROLLO DE CÓDIGO** | # Passing on the control system to the function  tipping\_ctrl = ctrl.ControlSystem([rule5, rule6, rule7, rule8, rule9, rule10, rule11, rule12, rule13, rule14, rule15, rule16, rule17, rule18, rule19, rule20, rule21, rule22, rule23, rule24, rule25])  tipping = ctrl.ControlSystemSimulation(tipping\_ctrl) |
| # Pass inputs to the ControlSystem using Antecedent labels with Pythonic API  # Note: if you like passing many inputs all at once, use .inputs(dict\_of\_data)  tipping.input['velocidad'] = 5  tipping.input['friccion'] = .3  # Crunch the numbers  tipping.compute() |
| **RESULTADOS** | print (tipping.output['posicion'])  posicion.view(sim=tipping) |
| 3.2837447873503103 |

# 

# BIBLIOGRAFÍAS

* Estrada Kassir, Eduardo (s.f) SISTEMAS DE CONTROL DIFUSO. Recuperado el 29 de Mayo del 2021, de: http://www.unicauca.edu.co/matematicas/eventos/log%26co/MEMORIAS/SxCtrl\_Difuso.pdf