

RAZONAMIENTO

INFORME DE PROYECTO

* BENITEZ, Emanuel

* BENITEZ, Lautaro

UNIVERSIDAD NACIONAL DE CUYO
FACULTAD DE INGENIERIA

2022

ÍNDICE

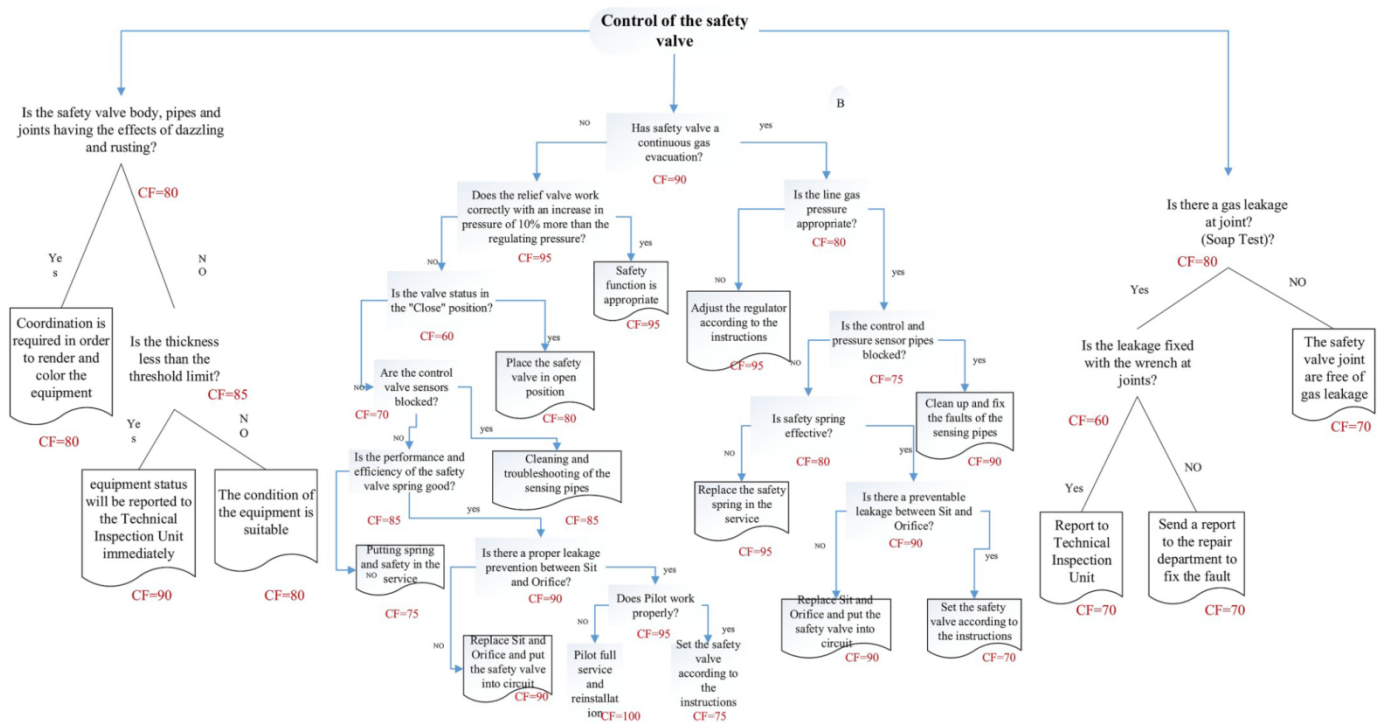
| | |
|---|-----------|
| INTRODUCCIÓN | 3 |
| AGENTES BASADOS EN EL CONOCIMIENTO | 3 |
| PLANNING | 4 |
| Carga Aérea | 4 |
| CAPP | 7 |
| LÓGICA DIFUSA | 9 |
| Problema | 9 |
| Borrosificación | 9 |
| Tabla FAM | 10 |
| Reglas de Inferencia de Mandani | 10 |
| Desborrosificación | 11 |
| Resultados | 12 |
| CONCLUSIÓN | 15 |

INTRODUCCIÓN

En este informe se abordará el desarrollo en softwares de los siguientes temas: “Agentes basados en el conocimiento”, “Planning” y “Lógica difusa”. A lo largo del mismo se exponen algunos de los detalles más importantes, a nuestro parecer, de cada uno de los temas mencionados, llevados a cabo a modo de ejercicios prácticos. Es necesario mencionar que en dicho trabajo se elaboró una de las posibles formas de resolver cada uno de los problemas planteados por la cátedra. Para cada uno de estos temas se utilizó el entorno VSC con algunas extensiones instaladas para el correcto funcionamiento de cada uno de los algoritmos.

AGENTES BASADOS EN EL CONOCIMIENTO

Para el primer ejercicio, se desarrolló una base de conocimiento en PROLOG de un sistema de evaluación y mantenimiento para una válvula de seguridad en una estación de reducción de presión de gas. Este proceso (simplificado) puede describirse mediante el árbol de decisión que se presenta a continuación:



El algoritmo se basa en la verificación del “estado” de los elementos que componen el sistema de seguridad de la central. Dicho estado responde a preguntas de tipo booleanas (Verdadero o Falso), en este caso, se asignó “Si” o “No” al valor de los estados, aunque existe la posibilidad de “Desconocido” en caso que no se posea información sobre el estado de ese componente en particular. En este último caso el sistema pedirá la verificación de los estados de los componentes dependientes según el diagrama.

Los axiomas del dominio del problema, se construyeron respetando el funcionamiento del algoritmo de búsqueda de PROLOG. En este caso es una búsqueda en profundidad hacia atrás, es decir que partiendo de una pregunta es posible verificar los predicados (estados) anteriores, en otras palabras es como leer el diagrama de árbol de abajo hacia arriba. Cuando se logra encontrar un estado coincidente, se toma una decisión, que podría terminar el proceso de búsqueda, es decir, que ningún paso anterior posea un estado “Desconocido”.

Es posible acceder a un menú de opciones de diagnóstico con la instancia “inicio.” que desplegará las siguientes opciones:

- 1 . Control del espesor y oxidación de la válvula
- 2 . Control del funcionamiento del piloto
- 3 . Control de la fuga de gas en la unión asiento-orificio
- 4 . Fin del programa

Ingresando la opción “1.” se puede acceder a la verificación del estado más relevante de la rama izquierda del diagrama, que en este caso es el “control del espesor y oxidación de la válvula”. Ingresando la opción “2.” se puede acceder a la verificación del estado más relevante de la rama central del diagrama, que aunque se bifurca en dos situaciones posibles existe una condición de una de ellas que es la más relevante de todas “control del funcionamiento del piloto”. En caso de ingresar la opción “3.” se verifica el estado de la rama derecha del diagrama cuya condición más relevante es el “control de la fuga de gas en la unión asiento-orificio”. Finalmente por comodidad, se dejó la opción “4.” que finaliza el programa, para que de una forma más práctica y rápida, libere la memoria y así poder probar posibles modificaciones.

PLANNING

Para el presente tema, se propone abordar dos problemas de planificación tales como el transporte aéreo de cargas y la planificación de procesos asistida por computadora (CAPP). El proceso de modelado, tanto los esquemas del dominio como los problemas, se diseñan en lenguaje PDDL haciendo uso de una versión online de planificador llamada LCAS-Lincoln Centre for Autonomous Systems.

Carga Aérea

Se toma como base el archivo del modelo planteado por la cátedra y se trabaja sobre el problema dejando sin cambios el dominio. Luego de varias modificaciones del problema se logró entender, además del ejercicio en sí, la lógica que conlleva la planificación al momento de encontrar una solución. De esta manera, el método de planning genera internamente un grafo de búsqueda donde analiza varios caminos en paralelo y luego de cierto punto, donde ya no puede seguir expandiéndose, ya sea porque encontró una solución, aplica el algoritmo A* para encontrar el mejor camino de vuelta. Otra situación por la que el grafo no puede seguir expandiéndose, se debe a que los estados sucesivos se repiten y no cubre la totalidad de los objetivos, en cuyo caso el algoritmo no encuentra una solución y falla.

Cabe considerar que el esquema del dominio planteado se maneja con costos unitarios por cada acción, es decir que no existe relevancia entre las acciones, por lo tanto el análisis no es completo. Teniendo en cuenta esto, se decidió seguir con el ejercicio y dejar planteado el análisis de costos para futuros trabajos.

En principio planteamos un escenario del problema con una cantidad de 4 aviones, aeropuertos y cargas debido a que una cantidad mayor de los mismos requería un costo computacional alto y que para fines prácticos y de entendimiento, no se consideró relevante.

A continuación se plantean dos situaciones de trabajo para un mismo estado inicial:

Estado inicial → (en LA01 AER)
 (en AA02 BAR)
 (en FB03 LIM)

(en GO01 MEX)

(en CARGA_ANIMALES_VIVOS AER)

(en CARGA_PERECIBLE LIM)

(en CARGA_GENERAL AER)

(en CARGA_FRAGIL BAR)

(en CARGA_MERCANCIAS_RESTRINGIDAS MEX)

| Estado objetivo 1 | Solucion 1 |
|--|--|
| (en CARGA_ANIMALES_VIVOS MEX) (en CARGA_PERECIBLE MEX) (en CARGA_GENERAL BAR) (en CARGA_FRAGIL AER) (en CARGA_MERCANCIAS_RESTRINGIDAS LIM) | (cargar carga_general la01 aer) (volar la01 aer bar) (descargar carga_general la01 bar) (cargar carga_fragil la01 bar) (volar la01 bar aer) (descargar carga_fragil la01 aer) (cargar carga_animales_vivos la01 aer) (volar la01 aer lim) (cargar carga_perecible la01 lim) (volar la01 lim mex) (descargar carga_perecible la01 mex) (cargar carga_mercancias_restringidas la01 mex) (descargar carga_animales_vivos la01 mex) (volar la01 mex lim) (descargar carga_mercancias_restringidas la01 lim) ; cost = 15 (unit cost) |

| Estado objetivo 2 | Solucion 2 |
|---|---|
| (en CARGA_ANIMALES_VIVOS MEX) (en CARGA_PERECIBLE MEX) (en CARGA_GENERAL BAR) | (cargar carga_general la01 aer) (volar la01 aer bar) (descargar carga_general la01 bar) |

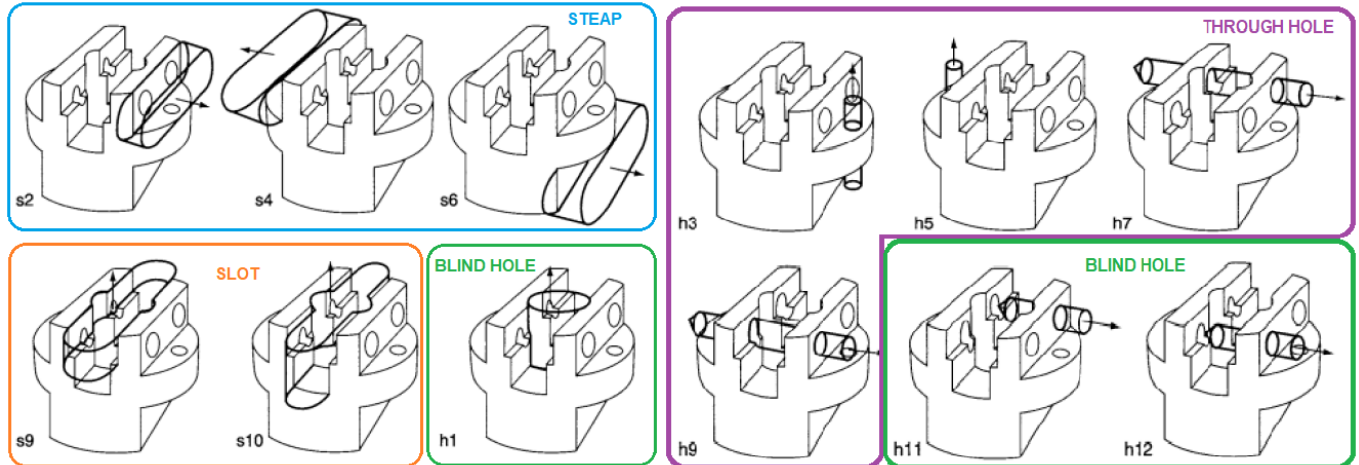
| | |
|--|--|
| (en CARGA_FRAGIL AER) | (cargar carga_mercancias_restringidas go01 mex) |
| (en CARGA_MERCANCIAS_RESTRINGIDAS LIM) | (cargar carga_perecible fb03 lim) |
| (en FB03 MEX) | (volar go01 mex lim) |
| (en AA02 MEX) | (cargar carga_fragil aa02 bar) |
| (en GO01 LIM) | (descargar carga_mercancias_restringidas go01 lim) |
| (en LA01 BAR) | (volar aa02 bar aer) |
| | (volar fb03 lim mex) |
| | (descargar carga_perecible fb03 mex) |
| | (descargar carga_fragil aa02 aer) |
| | (cargar carga_animales_vivos aa02 aer) |
| | (volar aa02 aer mex) |
| | (descargar carga_animales_vivos aa02 mex) |
| | ; cost = 15 (unit cost) |

| Parámetros del cálculo del OBJETIVO 1 | Parámetros del cálculo del OBJETIVO 2 |
|--|---|
| Plan length: 15 step(s). Plan cost: 15 Expanded 172 state(s). Reopened 0 state(s). Evaluated 1666 state(s). Evaluations: 1666 Generated 2712 state(s). Dead ends: 0 state(s). Search time: 0.0140361s Total time: 0.017285s Solution found. Peak memory: 34608 KB | Plan length: 15 step(s). Plan cost: 15 Expanded 539 state(s). Reopened 0 state(s). Evaluated 5851 state(s). Evaluations: 5851 Generated 9317 state(s). Dead ends: 0 state(s). Search time: 0.06495s Total time: 0.0705901s Solution found. Peak memory: 34776 KB |

En base a los resultados obtenidos, en el objetivo 1 al especificar únicamente la ubicación final de las cargas, el planificador automáticamente no toma en consideración la cantidad de aviones declarados en la inicialización y genera un plan que llega al objetivo, con un solo avión que recorre todos los aeropuertos transportando las diferentes cargas. Por otra parte en el objetivo 2, sí se especifica la ubicación final en la que se deben encontrar los diferentes aviones, luego el plan de acción encontrado difiere con el primer caso, pero a pesar de ello, el costo sigue siendo 15 unidades. Sin embargo en términos computacionales, los parámetros durante el cálculo difieren entre sí, demostrando que al agregar más estados objetivos el grafo se agranda y el tiempo de búsqueda como el recurso de memoria se incrementan.

CAPP

La CAPP (Planificación de procesos asistida por computadora) consiste en la definición de todos los procesos de fabricación para llegar desde la materia prima al producto terminado. En el presente ejercicio se trabajó sobre el dominio modelado por la cátedra, incluyendo nuevos setups y acciones para la fabricación de todas las features presentadas a continuación.



Para la realización del esquema del dominio se tomaron algunas suposiciones de trabajo, por ejemplo, se consideró trabajar con dos tipos de máquinas, la máquina de torno que realiza solo la operación torneado y la máquina de fresa que realiza la operación fresado y taladrado. Por lo tanto, se incluye un setup en el dominio denominado “cambio-máquina” como también “cambió-herramienta”, ya que la operación taladrado se ejecutará en la fresadora, y ésta debe tener la capacidad de intercambiar la herramienta de trabajo fresa por una broca. Otro punto a tener en cuenta es que durante toda la operación de maquinado, solo la pieza tiene la posibilidad de cambiar de posición, esto quiere decir que habrá un setup llamado “cambio-orientación-pieza” que ejecutará la acción, dejando fijos los instrumentos de corte y/o perforado de las máquinas. Por otra parte sumamos las operación propias del maquinado, como lo es el torneado, fresado y taladrado, destacando que para las dos últimas se tuvo que hacer una distinción con respecto a la posición que tomaría la pieza en relación con su herramienta de corte, ya sea longitudinal o transversal, para llegar a fabricar sus respectivos features. Es decir que tendremos dos acciones de fresado y taladrado, una por cada dirección.

A continuación se mostrará la solución encontrada para un estado inicial y objetivo propuesto:

Estado inicial → (pieza parte-001)
(orientacion-pieza orientacion-x)
(en herramienta-buril maquina-torno)
(en herramienta-broca maquina-fresa)
(en parte-001 maquina-torno)

| Estado objetivo | Solucion |
|-----------------|--|
| (fabricada c1) | (cambio-orientacion-pieza orientacion-x orientacion+y) |
| (fabricada h5) | (operacion_5 torneado c1 cilindrado orientacion+y orientacion-x parte-001 maquina-torno herramienta-buril) |
| (fabricada h3) | (cambio-maquina maquina-torno maquina-fresa parte-001) |
| (fabricada h7) | (operacion_4 taladrado-dist-dir h7 through-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada s2) | (operacion_4 taladrado-dist-dir h9 through-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada s4) | (operacion_4 taladrado-dist-dir h9 through-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada s6) | (operacion_4 taladrado-dist-dir h9 through-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada s9) | (cambio-orientacion-pieza orientacion+y orientacion-z) |
| (fabricada s10) | (operacion_3 taladrado-igual-dir h3 through-hole-longitudinal orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada h1) | (operacion_3 taladrado-igual-dir h5 through-hole-longitudinal orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada h9) | (operacion_3 taladrado-igual-dir h5 through-hole-longitudinal orientacion-z parte-001 maquina-fresa herramienta-broca) |
| (fabricada h11) | (cambio-herramienta-en maquina-fresa herramienta-broca herramienta-fresa) |
| (fabricada h12) | (operacion_1 fresado-igual-dir h1 blind-hole-longitudinal orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_1 fresado-igual-dir s10 slot orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_1 fresado-igual-dir s9 slot orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (cambio-orientacion-pieza orientacion-z orientacion+y) |
| | (operacion_2 fresado-dist-dir h11 blind-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_2 fresado-dist-dir h12 blind-hole-transversal orientacion+y orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_2 fresado-dist-dir s2 step orientacion+y orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_2 fresado-dist-dir s4 step orientacion+y orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | (operacion_2 fresado-dist-dir s6 step orientacion+y orientacion-z parte-001 maquina-fresa herramienta-fresa) |
| | ; cost = 18 (unit cost) |

La solución encontrada a partir del estado inicial del problema, en principio genera un plan adecuado y coherente. Sin embargo es necesario resaltar que el esquema del dominio no tiene en cuenta restricciones de

precedencia. Por lo tanto, para dar una forma cilíndrica a la pieza en bruto, primero que nada debe hacerse un torneado de la misma. Es así que para lograr este cometido y evitar que el programa dé respuestas incongruentes, se empieza con un estado inicial tal que la pieza ya se encuentre en la máquina del torno, genere el cilindro y continúe con los demás features. En caso contrario, en el que la pieza se inicialice en la máquina fresadora, el programa empezará a ejecutar las features que corresponden a tales procesos, dejando para el último el cilindrado de la pieza.

LÓGICA DIFUSA

Problema

Se implementó un sistema de inferencia difusa para controlar un péndulo invertido montado sobre un carro que se mueve en una sola dirección; considerando que el mismo no tiene un espacio restringido de movimiento, y que responde al siguiente modelo planteado

Borrosificación

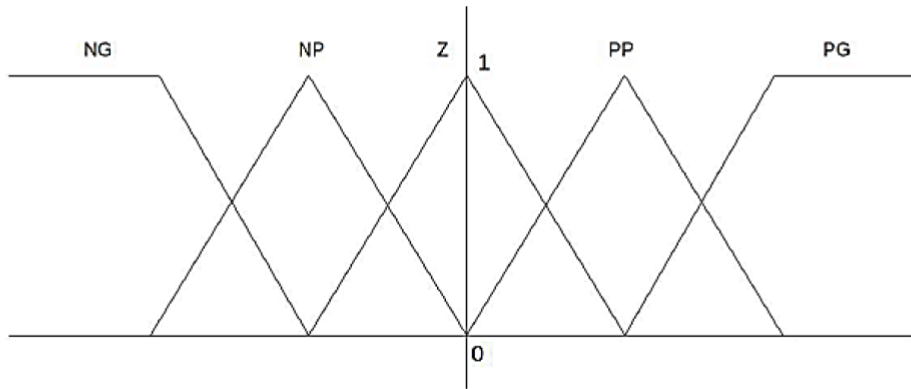


$$\ddot{\theta} = \frac{g \sin \theta + \cos \theta \left(\frac{-F - m l \dot{\theta}^2 \sin \theta}{M + m} \right)}{l \left(\frac{4}{3} - \frac{m \cos^2 \theta}{M + m} \right)}$$

$$\theta' = \theta' + \theta'' \Delta t$$

$$\theta = \theta + \theta' \Delta t + (\theta'' \Delta t^2) / 2$$

Todas las variables lingüísticas, tanto las de entrada como la de salida, tienen su partición borrosa dado por 5 conjuntos borrosos como se muestra a continuación.



- Borrosificación "Singleton"
- Solapamiento 50%
- Cant de conjuntos 5

Tabla FAM

| THETA'/ THETA | NG | NP | Z | PP | PG |
|---------------|----|----|----|----|----|
| NG | PG | PG | PP | PP | Z |
| NP | PG | PP | PP | Z | NP |
| Z | PP | PP | Z | NP | NP |
| PP | Z | NP | NP | NP | NG |
| PG | Z | NP | NP | NG | NG |

Reglas de Inferencia de Mandani

- R1: If theta is NG **and** theta_prima is NG THEN Fuerza is PG
- R2: If theta is NG **and** theta_prima is NP THEN Fuerza is PG
- R3: If theta is NG **and** theta_prima is Z THEN Fuerza is PP
- R4: If theta is NG **and** theta_prima is PP THEN Fuerza is Z
- R5: If theta is NG **and** theta_prima is PG THEN Fuerza is Z

- R6: If theta is NP **and** theta_prima is NG THEN Fuerza is PG
- R7: If theta is NP **and** theta_prima is NP THEN Fuerza is PP
- R8: If theta is NP **and** theta_prima is Z THEN Fuerza is PP
- R9: If theta is NP **and** theta_prima is PP THEN Fuerza is NP
- R10: If theta is NP **and** theta_prima is PG THEN Fuerza is NP

- R11: If theta is Z **and** theta_prima is NG THEN Fuerza is PP
- R12: If theta is Z **and** theta_prima is NP THEN Fuerza is PP
- R13: If theta is Z **and** theta_prima is Z THEN Fuerza is Z
- R14: If theta is Z **and** theta_prima is PP THEN Fuerza is NP
- R15: If theta is Z **and** theta_prima is PG THEN Fuerza is NP

- R16: If theta is PP **and** theta_prima is NG THEN Fuerza is PP

- R17: If theta is PP **and** theta_prima is NP THEN Fuerza is Z
- R18: If theta is PP **and** theta_prima is Z THEN Fuerza is NP
- R19: If theta is PP **and** theta_prima is PP THEN Fuerza is NP
- R20: If theta is PP **and** theta_prima is PG THEN Fuerza is NG
- R21: If theta is PG **and** theta_prima is NG THEN Fuerza is Z
- R22: If theta is PG **and** theta_prima is NP THEN Fuerza is NP
- R23: If theta is PG **and** theta_prima is Z THEN Fuerza is NP
- R24: If theta is PG **and** theta_prima is PP THEN Fuerza is NG
- R25: If theta is PG **and** theta_prima is PG THEN Fuerza is NG

Desborrosificación

Se adopta como método para desborrosificar la media por centro, dado que es muy efectivo a un bajo costo computacional.

$$y = \frac{\sum_{l=1}^{l=M} \bar{y}^l \mu_{B'}(\bar{y}^l)}{\sum \mu_{B'}(\bar{y}^l)} \quad \text{donde } \bar{y}^l \text{ es el punto máximo de } B', \text{ y } M \text{ es la cantidad de conjuntos borrosos de salida}$$

Una vez en el algoritmo, primero se ingresan los valores de θ y θ' para ser borrosificados y obtener sus respectivos valores de pertenencia, en el caso en cuestión se tuvo en cuenta un solapamiento del 50% que en muchos de los casos las variables iban a tener valor de pertenencia para dos conjuntos difusos. Esto implica que cada vez que se ingrese un valor de θ y θ' , íbamos a tener que aplicar de 1 a 4 de las 25 reglas. Un caso de una regla sería tener valores extremos en los hombros, que darían valores de pertenencia a un solo conjunto borroso, tanto para θ y θ' .

En estas condiciones se pueden dar reglas con el mismo o distinto consecuente. Si tuvieran el mismo consecuente de fuerza, se calcula el máximo valor de pertenencia entre ellas para obtener un solo consecuente, de lo contrario si se tuvieran más reglas que estas dos, cuando calculemos la media de centros el valor resultante al desborrosificar sería el correspondiente al centro de ese conjunto.

En caso de que se tengan solamente reglas con distinto consecuente lo que se hace es la disyunción entre ambos conjuntos borrosos para obtener un solo conjunto borroso con los máximos de pertenencia de cada uno, para luego aplicar la media de centro.

Estos dos casos anteriores indican que aplicando lógica a las 25 reglas se pueden agrupar todas las que tuvieran el mismo consecuente en solo 5 reglas. En el caso del algoritmo que se diseñó, este trabaja con las 25 reglas y a medida que se calculan, se aplica la lógica en el caso de que haya reglas con el mismo consecuente.

Resultados

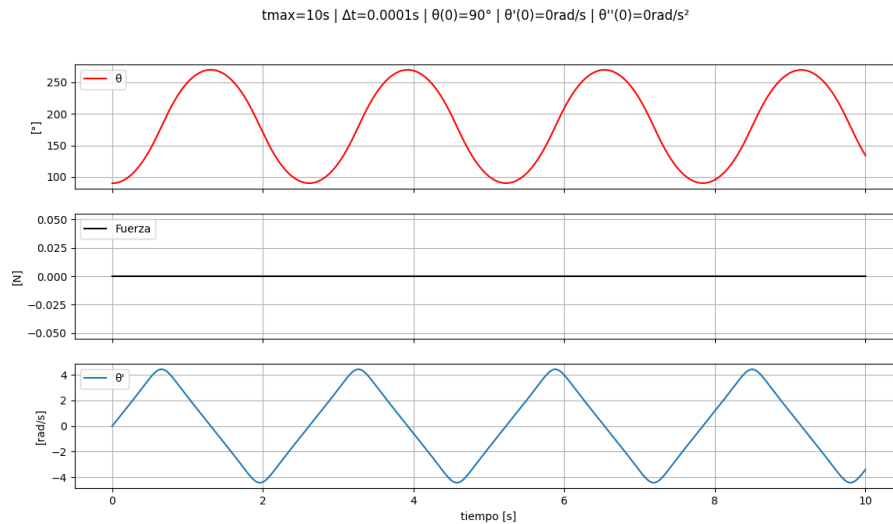
Para garantizar que el ejercicio tuviera validez en el razonamiento generado, se ha ido buscando primero de forma conveniente el dominio para cada una de las variables de entrada. En el caso de la velocidad, para el análisis, se observó entre qué valores oscila el péndulo en vacío con lo que se obtuvo el siguiente dominio (-6,6).

Para el caso del desplazamiento, aclarando que en el algoritmo 0° es la posición de equilibrio, el dominio queda definido entre $(-90,90)$ siendo estos los casos en los que la pértiga quedaría horizontal o paralela a la dirección de movimiento del carro. A pesar de que el universo planteado permite como máximo dar dominio entre $(-180, 180)$, esto implicaría pasar por debajo de la horizontal donde en el caso real se encuentra el suelo.

Habiendo aclarado el dominio de las variables de entrada, se exponen a continuación una serie de gráficas que el algoritmo nos devuelve en base a una combinación de parámetros presentados, con las cuales se puede obtener un entendimiento global del funcionamiento del mismo y el problema real.

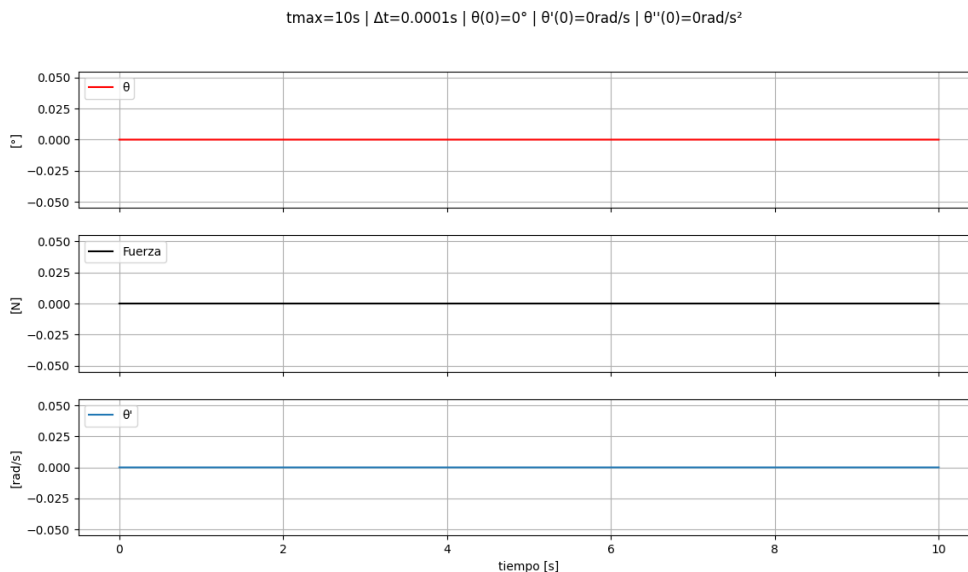
GRAFICA 1

En este caso se ha dejado el péndulo en uno de los límites del dominio del desplazamiento, sin velocidad inicial. Se puede ver como el péndulo oscila entre la máxima y la mínima posición ya que no tiene control de parte de la fuerza.



GRAFICA 2

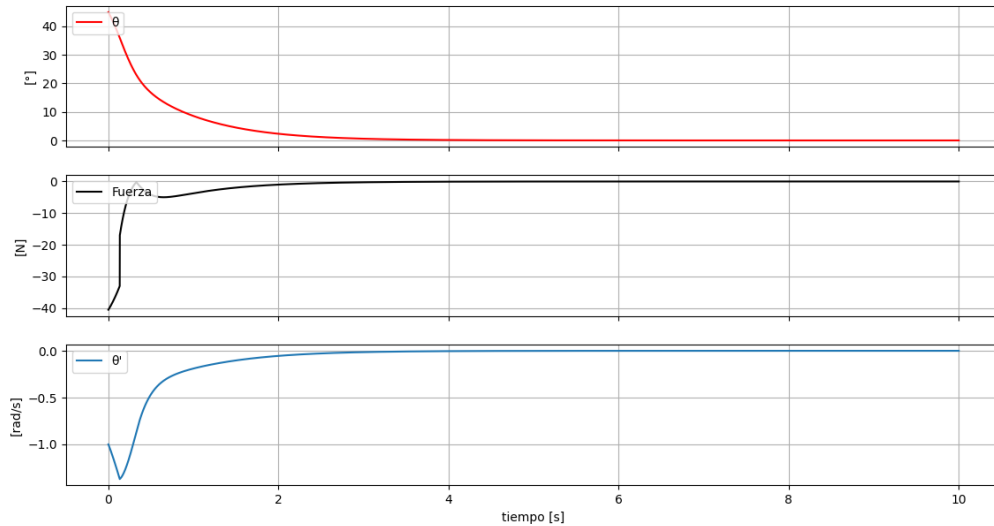
En este caso se generó la siguiente gráfica a modo de mostrar que el péndulo se encuentra en su estado de equilibrio sin ser afectado por ninguna velocidad, en cuyo caso todas las gráficas se mantienen invariables y por ende el sistema es capaz de no reaccionar en el estado de equilibrio.



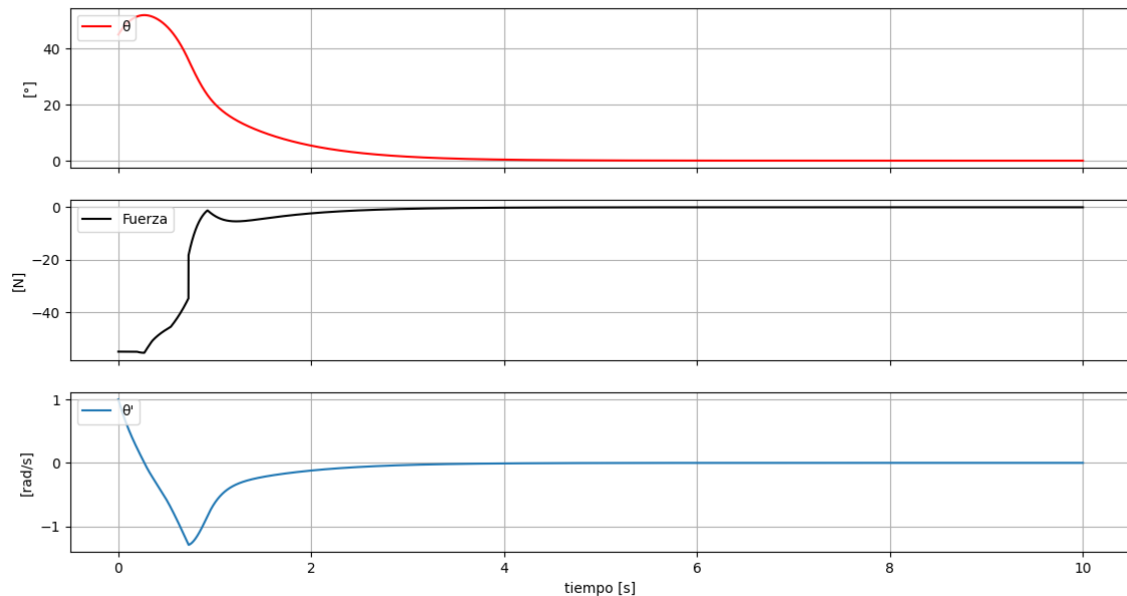
GRAFICA 3 y 4

Para poder determinar si en una situación intermedia entre el estado de equilibrio y el límite del dominio del desplazamiento el sistema controla correctamente el movimiento acorde a la lógica difusa planteada, se generaron estas dos posibilidades de donde a los 45° , en una se le aplica una velocidad pequeña negativa y en la otra el mismo módulo de velocidad pero positivo. Se puede ver claramente cómo afecta la velocidad en estas dos situaciones y en ambas la respuesta de la fuerza tiene un valor relativamente bajo.

$$t_{\max}=10s \mid \Delta t=0.0001s \mid \theta(0)=45^\circ \mid \theta'(0)=-1\text{rad/s} \mid \theta''(0)=0\text{rad/s}^2$$



$$t_{\max}=10s \mid \Delta t=0.0001s \mid \theta(0)=45^\circ \mid \theta'(0)=1\text{rad/s} \mid \theta''(0)=0\text{rad/s}^2$$

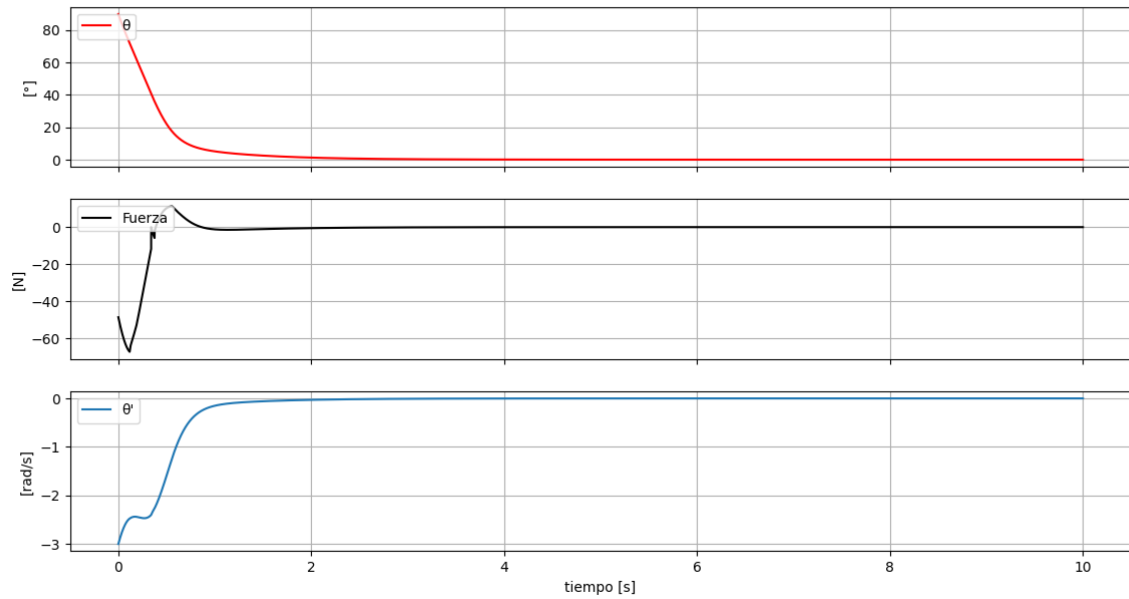


GRAFICA 5 y 6

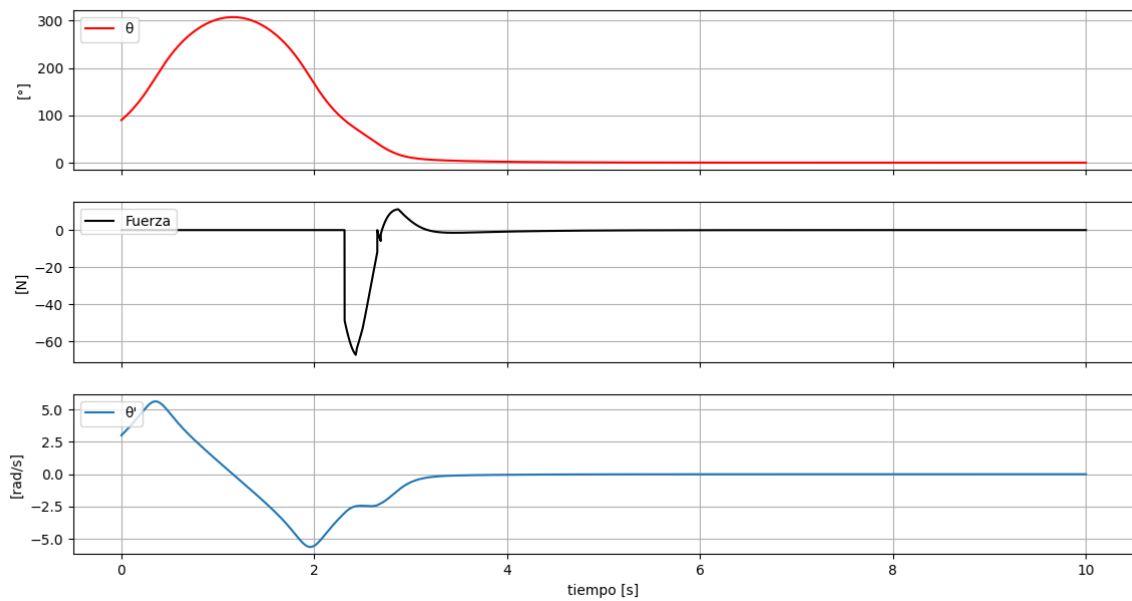
El mismo análisis anterior se ha aplicado en uno de los límites del dominio del desplazamiento (90°) para ver si es posible con una velocidad media y que valor de fuerza lleva el péndulo a su estado de equilibrio (0°). En uno de los casos, donde la velocidad es negativa, el sistema logra llegar al equilibrio con un valor de fuerza elevado pero no el máximo. Sin embargo, en la segunda imagen se puede ver claramente como el péndulo tiene un desplazamiento fuera de los límites del dominio y por ende la fuerza tiene un valor de 0N hasta que el mismo, por su peso vuelve, invirtiendo el sentido de la velocidad en cuyo caso la fuerza comienza a ofrecer su acción hasta la

estabilidad. Es necesario aclarar que dicho caso no podría pasar en la realidad ya que implicaría que el péndulo atravesase el piso.

$t_{\max}=10\text{s} \mid \Delta t=0.0001\text{s} \mid \theta(0)=90^\circ \mid \theta'(0)=-3\text{rad/s} \mid \theta''(0)=0\text{rad/s}^2$



$t_{\max}=10\text{s} \mid \Delta t=0.0001\text{s} \mid \theta(0)=90^\circ \mid \theta'(0)=3\text{rad/s} \mid \theta''(0)=0\text{rad/s}^2$



CONCLUSIÓN

Gracias a este informe se logró obtener un acercamiento a los conceptos de *Agentes basados en el conocimiento* y *Planning*, y una buena práctica de *Lógica Difusa* y por ende, de forma general, se pudo adquirir técnicas para el desarrollo de *Razonamiento* en IA.

En lo que respecta a *Agentes basados en el conocimiento*, hubiera sido una buena práctica la de obtener el estado de los componentes del sistema de forma externa al programa como por ejemplo, extraer los datos de un informe en formato “.txt”, que actualice los estados de la central o en su defecto la posibilidad de obtener estos datos directamente de los sensores vinculando el programa con otros en Arduino, Python, Raspberry Pi, o un PLC industrial.

En el caso de la planificación, como ya se comentó, no se tuvo en cuenta los costos de las acciones por lo tanto ninguna de ellas tenía relevancia sobre la otra. Sin embargo, en la realidad las acciones deben tener cierto nivel de importancia entre ellas, para lograr que el planning genere respuestas lógicas y eficientes con respecto a los problemas planteados.

Finalmente con *Lógica difusa*, la verdadera dificultad se encontró en lograr analizar el problema real, o sea, saber elegir bien qué criterios de borrosificación aplicar y generar las reglas de Mamdani adecuadas que reflejen el problema. Por otro lado, al momento de elegir los dominios de cada una de las variables, se logró dimensionar de alguna forma las limitaciones en la vida real, en otras palabras, el costo que implica cada resultado y los tiempos necesarios para estabilizar el sistema con los sensores y actuadores disponibles en el mercado acordes a un presupuesto lógico. El método para aplicar la fuerza puede limitar ciertos valores como así también la aplicación de esta podría comprometer la estructura. Como futuros ingenieros, el criterio, no solo implica reconocer las buenas aplicaciones de lógica difusa sino también saber minimizar los tiempos de implementación con un buen armado de base de conocimiento para luego maximizar los resultados adaptándose a la disponibilidad de recursos.