

Autor: Pablo Santervás Blanco

Tutor: Juan Manuel Escaño González

Trabajo Fin de Grado

en Ingeniería Electrónica, Robótica y Mecatrónica

Control inteligente de grúas industriales: aplicación de lógica borrosa en la automatización de tareas de carga

Dpto. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2024

Trabajo Fin de Grado

en Ingeniería Electrónica, Robótica y Mecatrónica

**Control inteligente de grúas industriales: aplicación de lógica borrosa en la automatización de tareas de carga**

Autor:

Pablo Santervás Blanco

Tutor:

Juan Manuel Escaño González

Profesor titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2024

Trabajo Fin de Grado: Control inteligente de grúas industriales: aplicación de lógica borrosa en la automatización de tareas de carga

|  |  |
| --- | --- |
| Autor: | Pablo Santervás Blanco |
| Tutor: | Juan Manuel Escaño González |

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

Agradecimientos

Quiero expresar mi más profundo agradecimiento a mi familia, que ha estado siempre a mi lado a lo largo de este camino, brindándome apoyo, confianza y palabras de aliento, especialmente en los momentos en los que las cosas no salían como esperaba.

También agradezco a todos los compañeros que he tenido la suerte de conocer durante estos años. Han sido personas extraordinarias que no solo me han acompañado, sino que también me han ofrecido su ayuda, tanto en lo personal como en lo académico, siempre con generosidad y disposición. Del mismo modo, quiero recordar a mis viejos amigos, quienes, a pesar del paso del tiempo, han continuado apoyándome y animándome en cada etapa de este proceso.

Por último, mi gratitud se extiende a todos los profesores que han sido parte de mi formación, por compartir sus conocimientos y guiarme durante esta etapa. De manera especial, quiero destacar a mi tutor, un gran profesional que me ha orientado con paciencia y creatividad, aportándome ideas clave cuando enfrentaba dificultades en este proyecto.

Pablo Santervás Blanco

Sevilla, 2024

Resumen

El presente trabajo aborda el desarrollo de un sistema automático para la carga de materiales pesados, con el objetivo de reducir los riesgos para los operarios y aumentar la productividad en el entorno industrial. Inicialmente, el manejo manual de las cargas implicaba altos riesgos de accidentes y limitaciones en la eficiencia operativa. Como solución, se planteó la automatización del proceso mediante una grúa controlada por un PLC, una opción que destaca por su capacidad para cumplir con los requisitos de seguridad y eficiencia, además de ser una tecnología ampliamente utilizada y probada en aplicaciones industriales.

El sistema de control implementado se basa en la lógica difusa, un enfoque reconocido por su capacidad para gestionar sistemas complejos con incertidumbre. La elección de esta metodología se fundamentó en la revisión de literatura científica que aborda problemáticas similares, aportando así una base sólida para la propuesta planteada.

Para evaluar el desempeño del sistema, se realizaron comparaciones con el método clásico de control basado en controladores PID. Los resultados evidenciaron la superioridad del controlador difuso, destacando no solo por ofrecer una mejor respuesta dinámica, sino también por su mayor adaptabilidad a cambios en las condiciones de operación, como el peso de la carga. Estas evaluaciones se llevaron a cabo en el entorno de simulación Simulink (MATLAB), empleando tanto los controladores de la herramienta Fuzzy Logic Toolbox como el modelo de controlador TM221 de Schneider Electric, disponible en el entorno Machine Expert Basic.

Cabe destacar que el desarrollo de este trabajo se basó en un caso real dentro de una fábrica donde se manipulan lotes de chapas de diferentes formatos y pesos. Sin embargo, la solución planteada puede ser aplicada a cualquier tipo de carga, no solo al caso particular en el que se basa este proyecto.

En conclusión, este proyecto demuestra la eficacia de soluciones tecnológicas consolidadas en la industria, contribuyendo a mejorar tanto la seguridad como la productividad en el manejo de materiales.

Abstract

This project focuses on the development of an automated system for handling heavy materials, aiming to reduce risks for operators and increase productivity in industrial environments. Initially, the manual handling of loads posed high accident risks and operational efficiency limitations. To address this issue, the automation of the process through a crane controlled by a PLC was proposed. This solution stands out for its ability to meet safety and efficiency requirements, as well as being a widely used and proven technology in industrial applications.

The implemented control system is based on fuzzy logic, a well-recognized approach for managing complex systems with uncertainty. The choice of this methodology was supported by a review of scientific literature addressing similar problems, providing a solid foundation for the proposed solution.

To evaluate the system's performance, comparisons were made with the classical control method using PID controllers. The results demonstrated the superiority of the fuzzy controller, excelling not only in delivering better dynamic response but also in greater adaptability to changes in operating conditions, such as load weight. These evaluations were conducted in the Simulink (MATLAB) simulation environment, using both the controllers provided by the Fuzzy Logic Toolbox and the TM221 controller model from Schneider Electric, available for simulation in the Machine Expert Basic platform.

It is important to note that this project was developed based on a real-world case in a factory that handles batches of metal sheets of varying sizes and weights. Nevertheless, the proposed solution can be applied to any type of load, not just the specific case on which this project is based.

In conclusion, this project confirms the effectiveness of well-established technological solutions in the industry, contributing to improved safety and productivity in material handling.

Índice

Agradecimientos vii

Resumen ix

Abstract xi

Índice xii

Índice de Tablas xv

Índice de Figuras xvii

Notación xix

1 Introducción 1

2 Viabilidad Técnica 12

2.1 Puente grúa automático 12

2.1.1 Funcionamiento 12

2.1.2 Ventajas: 12

2.1.3 Inconvenientes 12

2.2 Manipulador 12

2.2.1 Funcionamiento 12

2.2.2 Ventajas 12

2.2.3 Desventajas 13

2.3 AGV 13

2.3.1 Funcionamiento 13

2.3.2 Ventajas 13

2.3.3 Desventajas 13

2.4 Conclusiones 13

3 Simulación del sistema 16

3.1 Factory i/o 16

3.1.1 Introducción 16

3.1.2 Bloques usados 16

3.1.3 Modelo de planta 17

3.1.4 Simulaciones 18

3.2 Machine Expert Basic 19

3.2.1 Introducción 19

3.2.2 Programación y conexión con factory io 19

3.2.3 Programa 20

3.3 Conclusión y resumen 22

4 Control de la carga 24

4.1.1 Otra subsección 25

4.4 Otra sección 25

5 Lógica difusa 29

5.1 Estado del arte 29

5.2 Marco teórico 30

5.2.1 Conjuntos difusos 30

5.2.2 Variables lingüísticas 30

5.2.3 Funciones de membresía 30

5.2.4 Reglas difusas 31

Controlador difuso 33

5.2.5 Fuzzificación 33

5.2.6 Base de reglas 33

5.2.7 Mecanismos de inferencia 34

5.2.8 Defuzzificación 34

5.3 Controlador difuso en MATLAB 36

5.3.1 Creación del archive\*.fis 36

5.3.2 I/O y funciones de membresía 37

5.3.3 Reglas de inferencia 38

5.3.4 Visualizador de reglas 38

5.3.5 Implemetación del controlador en Simulink 40

6 Otro Capítulo 42

6.1 Estilos de un documento 42

6.2 Títulos y Referencias Cruzadas 43

6.3 Versiones y Sistemas Operativos 43

6.3.1 Macintosh 43

6.3.2 Linux 43

6.4 Texto en inglés 44

6.5 Elementos básicos de un libro 44

6.6 Símbolos y fórmulas 44

6.7 Ecuaciones y MathType® 44

6.7.1 Fuentes 45

6.7.2 Epígrafes o citas célebres 45

6.7.3 Figuras y tablas 46

6.7.4 Hiperenlaces 46

6.7.5 Tabla de contenido 47

6.7.6 Índice de figuras, tablas y otros elementos 47

6.7.7 Formatos de páginas 47

6.7.8 Teoremas y otros elementos similares 47

6.7.9 Ejemplos 48

6.7.10 Índices de palabras y glosarios 48

6.8 Antes del documento 48

6.9 Fuente del texto 49

6.10 Cubierta 49

Referencias 51

Índice de Conceptos 53

Glosario 55

# **Índice de Tablas**

Tabla 3‑1. Correspondecia entre input y outputs 20

Tabla 4‑1. Tipos de transmisión y frecuencia central 26

Tabla 5‑1. Operaciones lógicas del controlador 32

Tabla 5‑2. Ecuaciones del centroide 35

Tabla 6‑1 Tipos de transmisión y frecuencia central 46

# **Índice de Figuras**

Figura 1‑1. Tasa (%) de accidentes laborales en España a lo largo de la década (2009-18) [1] 1

Figura 4‑1. Esto es el pie de la figura. 25

Figura 5‑1. Ejemplo de funciones de membresía (MF) para el caso de la temperatura en una habitación 31

Figura 5‑2. Operadores lógicos aplicados a conjuntos difusos [10]. 32

Figura 5‑3. Esquema de un controlador difuso [10]. 33

Figura 5‑4. Ejemplo de evaluación de reglas [11]. 34

Figura 5‑5. Aplicación del método del centroide [11]. 35

Figura 5‑6. Proceso completo de operaciones de un controlador fuzzy [13]. 36

Figura 5‑7. FLD 37

Figura 5‑8. Editor de MFs 37

Figura 5‑9. Editor de reglas de inferencia 38

Figura 5‑10. Rule Inference 39

Figura 5‑11. Control Surface 39

Figura 5‑12. FLC 40

Figura 6‑1. Pie de figura 46

Notación

|  |  |
| --- | --- |
| LD  SFC  IL  POU  FIS  FLD  MF  IR  FLC  COF  E-L  MIMO  SS | Ladder Diagram  Sequential Function Chart  Instruction List  Program Organization Unit  Fuzzy Inference System  Fuzzy Logic Designer  Membership Function  Inference Rule  Fuzzy Logic Controller  Coefficient Of Friction  Euler-Lagrange  Multiple Input Multiple Output  State Space |

# Introducción

E

n el ámbito industrial, los trabajadores enfrentan múltiples riesgos laborales que afectan su seguridad y salud. Según estadísticas recientes, sectores como la manufactura y la logística presentan tasas significativas de accidentes debido a factores como el uso de maquinaria pesada, el manejo manual de cargas y las condiciones de trabajo en entornos exigentes. Estos incidentes no solo comprometen la integridad física de los empleados, sino que también generan importantes pérdidas económicas y operativas para las empresas, acentuando la necesidad de adoptar medidas efectivas para prevenirlos.

|  |
| --- |
| Figura 1‑1. Tasa (%) de accidentes laborales en España a lo largo de la década (2009-18) [1] |

En este contexto, el presente trabajo se enfoca en una problemática específica: el manejo de cargas pesadas. Actualmente, estas operaciones suelen realizarse con equipos manuales como transpaletas, lo que expone a los operarios a riesgos significativos. El proceso de carga y transporte, debido al peso y volumen de las cargas, incrementa las probabilidades de accidentes graves, creando puntos críticos de peligrosidad dentro de la planta. Este proyecto busca contribuir a la reducción de estos riesgos mediante la implementación de un sistema automatizado que no solo mejore la seguridad, sino que también optimice la productividad de los procesos industriales. Teniendo todo lo anterior en cuenta, la solución propuesta pretende minimizar la intervención humana en áreas críticas, reduciendo la exposición a riesgos y garantizando condiciones de trabajo más seguras. Al mismo tiempo, se busca optimizar los procesos de carga mediante una operación más eficiente y consistente, promoviendo una mayor productividad y alineándose con los estándares modernos de automatización industrial.

El diseño del sistema de control se basó en lógica difusa, un método conocido por su capacidad de gestionar sistemas complejos y con incertidumbre. Para su desarrollo y simulación se utilizó MATLAB, empleando herramientas específicas como Simulink y el Fuzzy Logic Toolbox. Estas plataformas permitieron modelar el comportamiento del sistema y optimizar las reglas difusas antes de su implementación en un PLC. Una de las ventajas principales de este enfoque radica en su flexibilidad, ya que permite incluir diversas variables que influyen en la operación, como la posición, velocidad y peso de la carga.

El proyecto también incluye la comparación del controlador difuso con un control en cascada PID+PI convencional, realizado en el entorno de simulación de Simulink. Estas comparaciones demostraron que la lógica difusa no solo proporciona una mejor respuesta dinámica, sino que también se adapta más eficientemente a cambios en las condiciones de operación, como las variaciones en el peso de la carga.

Aunque el diseño del controlador difuso puede requerir un mayor tiempo de ajuste debido a la importancia de definir correctamente sus reglas, su flexibilidad y rendimiento justifican ampliamente su elección. Por otro lado, el uso de MATLAB y Simulink facilitó tanto la validación del modelo como la comparación con otras estrategias de control, destacando la versatilidad de estas herramientas para el desarrollo de proyectos industriales.

Adicionalmente, se desarrolló un modelo virtual de la planta utilizando el software Factory IO. Este modelo permitió recrear de manera precisa la casuística real del proceso de carga, incluyendo aspectos como la disposición de los elementos, el flujo de materiales y las restricciones operativas. Sobre esta base, se implementó la automatización del sistema mediante un PLC simulado en el entorno Machine Expert Basic, mostrando una prueba visual de la automatización del proceso. Esta metodología permitió validar la funcionalidad del sistema en un entorno seguro y controlado antes de su eventual aplicación en un entorno físico.

En conclusión, este trabajo no solo implementa una solución ampliamente validada en la industria, sino que también explora el uso de un enfoque de control intuitivo y adaptable para mejorar la seguridad y la productividad en el manejo de materiales pesados. Además, el desarrollo y la validación del sistema mediante simulaciones en Factory IO y Machine Expert Basic permitió replicar con precisión las condiciones reales de la planta, garantizando la fiabilidad y eficacia de la solución antes de su implementación física.

# Viabilidad Técnica

C

omo paso previo a la selección de una solución, es necesario plantear diferentes propuestas que puedan ser empleadas para abordar el problema propuesto. En este caso particular, se han considerado tres posibles planteamientos que utilizan diferentes tecnologías, cada una con sus propias ventajas e inconvenientes. Esta sección actúa como preámbulo y justificación de la solución expuesta en los apartados siguientes.

## Puente grúa automático

### Funcionamiento

El sistema propuesto utilizaría una grúa para realizar todas las operaciones. La secuencia se iniciaría cuando el operario seleccionase el formato y el proceso de transformación del material, o lo que es lo mismo, el depósito de la máquina que procesará el material. Tras ello, la grúa recogería un lote del formato seleccionado y lo trasladaría a la celda de operación manual, donde un operario cortaría las bridas metálicas que sujetan la carga el material al palé, desacoplándolos, para después asegurar la carga mediante cinchos. Toda la operación se haría con la grúa alejada de la celda y parada, esperando a la confirmación manual del operario de la finalización del proceso. Posteriormente a un tiempo de seguridad después de la confirmación, la grúa trasladaría la carga a la estación seleccionada, esperando en su posición home hasta que hubiese una nueva orden.

### Ventajas:

* Utilización de maquinaría genérica y simple.
* Número de sensores y actuadores reducido, como corolario de la ventaja anterior.
* Implementación de un modo manual directo y simple de aprender.
* Robustez ante fallos, con posibilidad de operación manual en caso de error en el modo automático.
* Gran durabilidad, acompañado de un mantenimiento simple.

### Inconvenientes

* La carga se maneja suspendida en el aire, por lo que el fallo puede ser altamente peligroso.

## Manipulador

### 2.2.1 Funcionamiento

El sistema constaría de sendos depósitos de máquinas, una estación de operación y un manipulador móvil. La descarga en la plataforma de operación se realizaría mediante el método original de descarga. Posteriormente, y al igual que en el caso de la grúa el operario cortaría las bridas de acople entre la carga y el palé. El robot, una vez confirmada la ausencia de presencia humana en el entorno de trabajo, trasladaría la carga por unidad de la estación de operación a la estación de almacenaje de la máquina seleccionada. Al finalizar, el robot volvería a la posición home, esperando nuevas órdenes.

### 2.2.2 Ventajas

* Alto nivel de seguridad al manejar cargas inferiores.
* Dimensiones reducidas.
* Adaptable a diferentes condiciones de operación.

### 2.2.3 Desventajas

* Sistema complejo con muchos sensores y actuadores.
* Alta utilización debido a la descarga por unidad.
* Instalación compleja y voluminosa.
* Mayor asiduidad en el mantenimiento.
* Dificultad para reconocer el origen de los fallos.

## AGV

### 2.3.1 Funcionamiento

El sistema constaría de sendos depósitos de máquinas, una estación de operación y un AGV móvil. La descarga en la plataforma de operación lo realizaría el AGV recogiendo la carga. Posteriormente, y al igual que en los anteriores casos, el operario cortaría las bridas de acople entre la carga y el palé. El AGV, una vez confirmada la ausencia de presencia humana en el entorno de trabajo, trasladaría el lote de la estación de operación a la estación de almacenaje de la máquina seleccionada. Al finalizar, el AGV volvería a la posición home, esperando nuevas órdenes.

### 2.3.2 Ventajas

* Alto nivel de seguridad al manejar los lotes a la altura del suelo de forma autónoma.
* Dimensiones reducidas.
* Modulable a diferentes tipos de carga.
* El sistema puede integrarse en entornos industriales más grandes o complejos con relativa facilidad.

### 2.3.3 Desventajas

* Sistema complejo con muchos sensores y actuadores.
* Instalación compleja y altamente dependiente de la presencia de otros objetos y personas en el entorno.
* Mantenimiento complejo.
* **Requerimiento de una infraestructura específica** para que funcionen correctamente.

## Conclusiones

Tras analizar las diferentes alternativas propuestas para la automatización del manejo de materiales pesados, se concluye que la opción del puente grúa automático es la más adecuada para esta aplicación específica. Esta elección se fundamenta en varios factores clave que hacen que esta solución sobresalga frente a las demás.

En primer lugar, es un sistema ampliamente validado por su uso en entornos como los puertos comerciales, donde se emplean para manejar cargas pesadas de forma eficiente y segura. Estos antecedentes demuestran la capacidad de esta tecnología para operar en condiciones diversas y su fiabilidad a lo largo del tiempo.

Además, el sistema de grúa presenta una complejidad significativamente menor en comparación con las otras alternativas, como el manipulador o el AGV. Su reducido número de sensores y actuadores no solo simplifica su diseño e instalación, sino que también minimiza los puntos de fallo potenciales, lo que incrementa la robustez del sistema. En caso de un fallo en el modo automático, la posibilidad de operar manualmente de manera directa permite garantizar la continuidad de las operaciones sin grandes interrupciones.

Por otro lado, la grúa destaca por su mantenimiento sencillo y durabilidad, factores que reducen los costos operativos a largo plazo. A diferencia del manipulador o el AGV, que requieren instalaciones más complejas y un mantenimiento más frecuente, la grúa ofrece un ciclo de vida más prolongado con menores intervenciones técnicas.

Finalmente, si bien el manejo de cargas suspendidas representa un riesgo en caso de fallos graves, este aspecto puede mitigarse mediante un diseño adecuado de los sistemas de seguridad, como sensores redundantes y procedimientos estrictos de operación. La experiencia acumulada en su uso en otros sectores respalda esta afirmación.

En resumen, el puente grúa automático se posiciona como la mejor opción por su simplicidad, confiabilidad, menor coste operativo y respaldo en aplicaciones similares, ofreciendo una solución robusta y eficiente para la automatización del manejo de materiales en este proyecto.

# Simulación del sistema

L

a simulación de sistemas automáticos es una herramienta esencial en el desarrollo y validación de soluciones industriales. Antes de implementar un sistema en un entorno físico, es fundamental evaluar su comportamiento en condiciones virtuales, lo que permite identificar posibles fallos, optimizar el diseño y garantizar su funcionalidad. Este enfoque reduce los riesgos asociados a la puesta en marcha y minimiza los costos de ajustes posteriores.

En este proyecto, se emplearon dos herramientas clave para la simulación: Factory IO, para modelar el entorno físico y las interacciones del sistema, y Machine Expert Basic, para simular la lógica de control del PLC. La integración de ambas herramientas permitió realizar pruebas exhaustivas del sistema propuesto, asegurando su correcto funcionamiento y fiabilidad antes de su implementación final.

En el caso particular en el que se ha basado el trabajo, es necesario realizar una operación manual sobre la carga previamente para poder introducirla en los depósitos de las máquinas, lo que lo hace más complejo que un caso en el que simplemente sea necesario trasladar la carga entre dos puntos deseados. Es por ello que cobra todavía más relevancia la simulación, pues es una situación de colaboración hombre-máquina en la que se debe de asegurar la integridad del operario.

## Factory i/o

### Introducción

Factory IO es una plataforma de simulación diseñada para crear entornos virtuales que emulan procesos industriales reales. Utiliza un motor gráfico basado en Unity 3D, para ofrecer una representación visual detallada y realista de los sistemas modelados. Este software está orientado tanto a estudiantes como a profesionales en el ámbito de la automatización industrial, brindando una herramienta versátil para el diseño, la simulación y la enseñanza de sistemas automatizados.

Entre sus funcionalidades básicas, Factory IO permite crear modelos de plantas industriales mediante una biblioteca extensa de componentes predefinidos, que incluye sensores, actuadores, transportadores, robots y otros elementos típicos de entornos automatizados. Estos componentes se pueden configurar y conectar para reproducir flujos de trabajo, probar secuencias de operación y validar soluciones de control. Además, el software es compatible con diversos controladores lógicos programables (PLC), sistemas SCADA y software de programación, lo que facilita la integración de simulaciones con entornos de control reales o virtuales.

Otra característica destacada es su capacidad para simular fallos en los componentes, lo que resulta útil para evaluar la robustez de los sistemas de control y preparar estrategias de respuesta ante eventos inesperados. Asimismo, permite realizar mediciones en tiempo real, como el conteo de piezas, tiempos de ciclo y eficiencia del sistema, proporcionando datos valiosos para optimizar procesos.

Esto lo ha convertido en una herramienta clave en el proyecto para diseñar y validar sistemas de automatización sin necesidad de contar con una instalación física, reduciendo costos y riesgos. Su enfoque intuitivo y su versatilidad han reducido el problema de diseño del entorno a mínimos, propiciando un mejor desarrollo del automatismo, sin necesidad de preocuparse de problemas relacionados con el modelo empleado.

### 3.1.2 Bloques usados

En el diseño del modelo se emplearon diversos bloques disponibles en Factory IO, cada uno desempeñando funciones específicas para emular las operaciones del sistema automático. A continuación, se describen los bloques utilizados y su papel dentro de la simulación:

1. [**Emitter**](https://docs.factoryio.com/manual/parts/emitter/) **&** [**Remover**](https://docs.factoryio.com/manual/parts/remover/)**:**

El bloque Emitter genera los lotes de material necesarios para el sistema, representados como [raw metal](https://docs.factoryio.com/manual/parts/items/#raw-material) y [square pallet](https://docs.factoryio.com/manual/parts/items/#pallets). El bloque Remover, por su parte, elimina los lotes de la simulación una vez completadas las operaciones asociadas, manteniendo el flujo de trabajo dentro del modelo.

1. [**Pick and Place**](https://docs.factoryio.com/manual/parts/stations/#pick-place) **(Analog):**

Este bloque emula el funcionamiento del puente grúa mediante movimientos controlados en los ejes X, Y y Z, además de la acción de "coger" (grab) el material. Debido a las limitaciones de tamaño en Factory IO, se utilizaron dos bloques de Pick and Place: uno para la recogida del material y otro para la descarga en el depósito de la máquina seleccionada.

1. [**Safety Door**](https://docs.factoryio.com/manual/parts/walkways/#safety-door) **&** [**Safeguards**](https://docs.factoryio.com/manual/parts/walkways/#safeguard)**:**

Estos bloques emulan la celda de trabajo donde la puerta, equipada con un sensor, detecta si hay alguien dentro o no. Se utiliza para garantizar la seguridad en la estación de operación, emulando el control de acceso del operario durante el proceso de corte.

1. [**Diffuser Sensor**](https://docs.factoryio.com/manual/parts/sensors/#diffuse-sensor) **&** [**Belt Conveyor**](https://docs.factoryio.com/manual/parts/light-load/#belt-conveyors) **(2m):**

Estos bloques se añadieron como elementos extra para trasladar la carga desde el Emitter al Remover dentro de la estación de operación, en el sistema real estos o existirían.

1. [**Warning Devices**](https://docs.factoryio.com/manual/parts/warning-devices/)**:**

Este conjunto de bloques incluye tres elementos:

* Warning Light, que se activa para avisar de la próxima activación del pick and place para la recogida del material de la estación de trabajo.
* Stack Light, que indica el estado del sistema mediante colores: rojo para emergencia, amarillo para espera o parada, y verde para funcionamiento normal.
* Alarm Siren, que solo se activa si el botón de parada de emergencia está presionado.

1. [**Panel de Control**](https://docs.factoryio.com/manual/parts/operators/)**:**

Este bloque centraliza todos los botones, selectores y contadores necesarios para interactuar con la simulación. Permite al operario seleccionar formatos, activar procesos y supervisar el estado del sistema.

1. [**Handrails**](https://docs.factoryio.com/manual/parts/walkways/#handrails) **&** [**platforms**](https://docs.factoryio.com/manual/parts/walkways/#platforms)**:**

Los primeros sirven para delimitar la zona de operación, al igual que se haría en una planta real y los segundos sirven para permitir establecer los emitters a una altura suficiente para que el bloque pick and place pueda recoger la carga.

Estos bloques permiten replicar de manera funcional las operaciones y la lógica del sistema descrito, garantizando un entorno de simulación realista y eficiente para probar y ajustar el control del PLC antes de su implementación.

### 3.1.3 Modelo de planta

Debido a las limitaciones de Factory IO, que no permite modificar las dimensiones de sus elementos, se ha diseñado una escena que, aunque no es idéntica a la planta descrita en el sistema físico, resulta funcionalmente equivalente desde la perspectiva del PLC. Este modelo incluye tres secciones principales que emulan las operaciones fundamentales del sistema: la recogida del material, la estación intermedia y la zona de descarga. A continuación, se describen en detalle estas secciones y la funcionalidad que cumple el PLC en cada una de ellas.

* **Zona de recogida del material:** Esta sección simula el almacenamiento de los lotes de material, representados por tres formatos distintos para simplificar el modelo. El sistema utiliza un pick and place que sustituye al puente grúa, encargado de recoger el formato solicitado por el operario a través de los selectores del cuadro de control. Una vez seleccionado el formato, el pick and place traslada el lote a la estación intermedia, donde continuará el proceso.
* **Estación intermedia:** En esta sección, se simula el proceso manual de corte de bridas mediante la apertura de una puerta con detector. La puerta permanece abierta durante todo el proceso de corte, y el operario debe pulsar el botón de "finish" para confirmar la finalización de la tarea. Tras esta acción, se activa una luz rotatoria y se inicia un temporizador de seguridad. Al finalizar este tiempo, el sistema activa la recogida del lote por el pick and place para continuar con el traslado. Si al finalizar el tiempo de seguridad la puerta sigue abierta, el sistema asume que hay presencia humana en la celda, y será necesario confirmar nuevamente la finalización de la tarea manual para garantizar la seguridad.
* **Zona de descarga:** Esta sección es funcionalmente análoga a la zona de recogida, pero en este caso, el pick and place traslada el lote desde la estación intermedia hasta el depósito de la máquina seleccionada por el operario mediante los selectores del cuadro de control. El sistema finaliza el proceso cuando el lote es colocado correctamente en el destino seleccionado.

Para gestionar las operaciones de traslado entre secciones, así como la creación y eliminación de lotes dentro de la simulación, se utilizaron los bloques emitters y removers de Factory IO. El PLC, por su parte, supervisa y controla toda la lógica del sistema, incluyendo la selección de formatos, la activación de los elementos mecánicos y las comprobaciones de seguridad, garantizando el correcto funcionamiento de todo el proceso simulado.

### Simulaciones

#### Funcionamiento en Automático

[](https://www.youtube.com/embed/fXPZ8zIJSvA?feature=oembed)

#### Reset & Stop

[](https://www.youtube.com/embed/9RZSlg7Hhos?feature=oembed)

#### Emergency

[](https://www.youtube.com/embed/ubquyquheo0?feature=oembed)

## Machine Expert Basic

### Introducción

Machine Expert Basic es un software desarrollado por Schneider Electric para la programación y configuración de sus controladores lógicos programables (PLCs), específicamente de la serie Modicon TM221. Este entorno está diseñado para aplicaciones industriales de automatización, ofreciendo una interfaz intuitiva y herramientas versátiles que facilitan la programación y simulación de sistemas automatizados.

Uno de los aspectos más destacados de Machine Expert Basic es su compatibilidad con diversos lenguajes de programación estándar definidos en la norma IEC 61131-3. Entre los lenguajes soportados se encuentran:

* **LD:** Ideal para usuarios con experiencia en diagramas de escalera, común en sistemas eléctricos.
* **IL:** Un lenguaje basado en instrucciones, compacto y eficiente, similar al ensamblador.
* **SFC:** Diseñado para programar procesos secuenciales, permite organizar el control en pasos y transiciones, facilitando la visualización y estructuración lógica de sistemas complejos.

El software también incluye herramientas básicas de simulación que permiten probar la lógica de programación sin necesidad de hardware físico, reduciendo así el tiempo y los costos asociados al desarrollo.

Entre sus funcionalidades adicionales destacan:

* **Diagnóstico y depuración:** Permite identificar errores en tiempo real y realizar ajustes sobre la marcha.
* **Bibliotecas predefinidas:** Incluye bloques de funciones y plantillas que facilitan la programación de tareas comunes.
* **Compatibilidad con hardware de Schneider Electric:** Garantiza una integración fluida con dispositivos como sensores, actuadores y módulos de expansión.

En resumen, Machine Expert Basic está diseñado para ser una herramienta accesible pero poderosa, adecuada para profesionales de la automatización industrial que necesitan desarrollar y mantener sistemas de control eficientes y seguros. Su enfoque en la simplicidad y la compatibilidad lo convierte en una opción popular en entornos donde la optimización de tiempo y recursos es esencial.

### Programación y conexión con factory io

Factory IO se conecta con Machine Expert Basic a través del protocolo Modbus TCP/IP, uno de los métodos más comunes y eficientes para la comunicación entre dispositivos en sistemas de automatización industrial. Este protocolo permite el intercambio de datos en tiempo real, asegurando que las señales generadas en la simulación de Factory IO sean correctamente interpretadas y gestionadas por el PLC configurado en Machine Expert Basic.

#### Asignación de registros modbus

La conexión se establece asignando las variables de Factory IO a registros específicos del PLC mediante las siguientes convenciones:

**Señales digitales:**

Las entradas y salidas digitales de Factory IO se asignan a los registros del tipo %Mx en Modbus. Estos registros se relacionan directamente con:

* Entradas digitales (%Ix), para señales provenientes de sensores u otros dispositivos de entrada.
* Salidas digitales (%Qx), para señales destinadas a actuadores como luces o motores.

**Señales analógicas:**

Las entradas y salidas analógicas de Factory IO se asignan a registros del tipo %MWx en Modbus. Estos registros se relacionan con:

* Entradas analógicas (%IWx), para señales de sensores como potenciómetros o medidores.
* Salidas analógicas (%QWx), utilizadas para controlar dispositivos como servomotores o variadores de frecuencia.

Tabla 3‑1. Correspondecia entre input y outputs

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Output | Register |
| Digital | %Ix | %Qx | %Mx |
| Analog | %IWx | %QWx | %MWx |

#### Configuración de la Conexión

**En Factory IO:**

* Habilitar el protocolo Modbus TCP/IP desde el panel de conectividad.
* Definir la dirección IP del PLC o simulador (127.0.0.1).
* Asignar los registros Modbus correspondientes a cada entrada y salida digital o analógica.

**En Machine Expert Basic:**

* Declarar las variables de entrada y salida en el programa (%Mx y %MWx).
* Configurar la tabla de asignación para asegurar la correcta correspondencia con los dispositivos simulados.

Este enfoque no solo simplifica la integración, sino que también permite probar y depurar la lógica del sistema de manera eficiente antes de llevarla al entorno físico. La equivalencia en los registros asegura que la simulación sea una réplica fiel de las operaciones reales, optimizando el desarrollo y reduciendo posibles errores.

### Programa

#### Estructuración

El programa desarrollado está dividido en varios POUs para estructurar y organizar mejor la lógica de control. Cada uno de estos módulos tiene una función específica, lo que facilita la comprensión, el mantenimiento y la ampliación del sistema. Los POUs son los siguientes:

* **Grafcet:** Este programa, implementado en SFC contiene toda la lógica secuencial del sistema. Define los diferentes estados de la máquina (asociados a los registros %Xi del PLC) y las transiciones entre ellos, organizando el flujo del proceso de manera clara y estructurada.
* **Actions:** Un programa en LD que gestiona las condiciones necesarias para la activación de los actuadores del sistema. Aquí se define cuándo deben activarse elementos como motores, puertas o luces, dependiendo de las señales de entrada y las condiciones del sistema.
* **Init Cond:** También implementado en LD, este programa asegura que todos los actuadores del sistema se mantengan en su estado de reposo inicial cuando las condiciones lo requieran, garantizando la seguridad y estabilidad del sistema antes de iniciar cualquier operación.
* **Buttons:** Este programa en LD contiene la lógica asociada al panel de control. Gestiona las señales de los botones, selectores y demás dispositivos de entrada del operador, permitiendo la interacción directa con el sistema.
* **Lights:** Otro programa en LD dedicado a la gestión de los dispositivos de advertencia visual, como luces de advertencia y estados (warning lights y stack lights). Aquí se controla la activación de las luces dependiendo de las condiciones del sistema, indicando estados como emergencia, stand-by o funcionamiento normal.
* **Counters and Timers:** Este programa en LD incluye todos los contadores y temporizadores utilizados en el sistema. Su función es medir tiempos de espera, gestionar retardos y contar eventos, elementos críticos para el correcto funcionamiento del proceso secuencial.

Esta división modular permite abordar cada parte del sistema de manera independiente, lo que facilita la depuración y mejora la claridad del programa.

#### Programa principal

El GRAFCET diseñado, que comprende los POUs de ‘grafcet’, ‘actions’ e ‘init cond’ antes mencionados, sigue una secuencia lógica para garantizar el manejo adecuado del material mediante el sistema automatizado. Por comodidad se ha dividido en tres secciones que son las siguientes:

##### Movimiento de recogida

1. **Estado inicial (Reposo):** En este estado, todos los actuadores digitales están reseteados, asegurando que no haya actividad accidental. Mientras que los actuadores analógicos (ejes X, Y, y Z) están posicionados en Home, lo que garantiza un punto de referencia conocido y seguro.
2. **Selección del formato:** Según la selección realizada por el operario, el sistema determina el tipo de formato (1, 2, o 3) y calcula la posición correspondiente en el plano XY. Esto activa la transición al estado siguiente.
3. **Movimiento al punto de recogida:** El sistema mueve el eje X e Y a la posición designada según el formato seleccionado. La transición ocurre únicamente cuando ambos ejes alcanzan su posición objetivo, confirmada por sus sensores de posición.
4. **Descenso para recogida:** Una vez alcanzada la posición en XY, el eje Z desciende al nivel de recogida de la carga. En este estado, el sistema activa el actuador de agarre (set ‘grab’) para sujetar el material.
5. **Confirmación de agarre:** Un sensor en el mecanismo de agarre verifica que el material ha sido correctamente sujetado. Al recibir esta confirmación, el sistema procede a elevar el eje Z a la altura previa, garantizando un traslado seguro.
6. **Traslado a estación de trabajo:** Tras alcanzar la altura inicial, el sistema traslado la carga a la posición de la estación de trabajo en el plano XY.
7. **Depositado:** Análogo al movimiento **4** tras alcanzar la posición designada para los ejes X e Y, se procede a depositar la carga en la estación intermedia.

##### Operación manual

1. **Retorno a posición home:** Una vez depositada la carga en la estación intermedia, el sistema se mueve a la posición Home, alejándose completamente de la zona de trabajo para garantizar la seguridad del operario.
2. **Operación manual del operario:** El operario entra en la celda de trabajo y mantiene la puerta abierta mientras realiza la operación necesaria, verbigracia, cortar las bridas y asegurar la carga, ahora libre, con cinchos. Al finalizar, cierra la puerta y confirma manualmente en el panel de control que la tarea ha concluido.
3. **Activación del tiempo de seguridad:** Se enciende la luz rotatoria como advertencia de que la grúa se moverá tras el tiempo de seguridad (TOF).Si al finalizar el tiempo la puerta está abierta, el sistema asume la presencia del operario y solicita una nueva confirmación tras cerrar la puerta.En cambio, si la puerta está cerrada, se confirma la transición al siguiente estado automáticamente.

##### Movimiento de depositado

1. **Traslado a la estación de trabajo:** El sistema se mueve en el plano XY hacia la posición correspondiente a la celda de trabajo. Pasando al siguiente estado cuando llegue a la posición objetivo.
2. **Recogida de la carga:** El sistema desciende y activa el sistema de agarre (set ‘grab’) para sujetar la carga. Un sensor confirma que la carga está asegurada, permitiendo que la grúa eleve el material nuevamente a la posición Z previa.
3. **Traslado al depósito de la máquina seleccionada:** Según la selección realizada en el panel de control, sistema se desplaza en el plano XY hacia el depósito de la máquina de procesado correspondiente.
4. **Depósito de la carga:** La grúa desciende y deposita la carga en el almacén de la máquina seleccionada.Una vez alcanzada la altura de depósito, el sistema de agarre se desactiva (reset ‘grab’) para liberar el material.
5. **Retorno a posición home:** La grúa vuelve a la posición Home, quedando lista para recibir nuevas órdenes, completando así el ciclo.

## Conclusión y resumen

A lo largo de este apartado, se ha presentado el desarrollo de un sistema automatizado mediante las herramientas Factory IO y Machine Expert Basic, destacando sus características principales y su papel complementario en el diseño, simulación y validación de sistemas industriales.

Factory IO ha permitido crear un entorno virtual en el que se emula de forma realista el comportamiento de una planta industrial. Aunque cuenta con limitaciones en cuanto a la personalización de algunos elementos, ofrece una plataforma flexible e intuitiva para experimentar con distintos escenarios y realizar pruebas antes de implementar la solución en un entorno físico. Su integración con protocolos estándar como Modbus TCP/IP facilita la comunicación con controladores reales o simulados.

Por su parte, Machine Expert Basic ha proporcionado el entorno ideal para el diseño y programación del controlador lógico programable (PLC). Con su compatibilidad con lenguajes de programación estándar como LD y SFC, ha sido posible implementar la lógica del sistema de manera estructurada, utilizando diversas POU para organizar las funciones clave.

La combinación de ambas herramientas ha permitido modelar, probar y ajustar un sistema automatizado completo, desde la lógica de control hasta la simulación del comportamiento físico, asegurando la viabilidad y la fiabilidad del diseño. Esto destaca la importancia de utilizar herramientas integradas en la industria para optimizar tanto los procesos de desarrollo como los recursos empleados en la implementación de soluciones reales.

# Lógica difusa

E

n el entorno que nos rodea, existen numerosos sistemas que son o pueden ser controlados mediante diferentes métodos. Un ejemplo común de ello es el control en lazo cerrado usando un controlador PID. Con el paso del tiempo, se ha incrementado el número de sistemas y aplicaciones que los seres humanos buscan automatizar para lograr mayores niveles de eficiencia y optimización. Además de esto, la automatización ofrece la ventaja de reducir costos y aumentar la producción, ya que los sistemas pueden realizar tareas de forma continua, sin necesidad de descansos, a diferencia de las personas, que necesitan reponer su fuerza de trabajo. Esta capacidad de operar durante largos períodos hace que la automatización sea cada vez más utilizada en diferentes sectores, como en fábricas de alimentos procesados o en sistemas logísticos de distribución. Sin embargo, surgen dificultades cuando se tratan de sistemas reales cuya modelización mediante una función de transferencia resulta compleja o incluso imposible, ya sea porque presentan comportamientos no lineales, variables a lo largo del tiempo, o son demasiado complejos para ser descritos con una función matemática precisa. Por esta razón, se han desarrollado métodos alternativos para abordar estos casos. Uno de ellos es el control mediante lógica difusa, o controlador difuso, que será el foco de este estudio.

Este tipo de lógica se acerca más al razonamiento humano que la lógica tradicional binaria, en la que los resultados son siempre claros y definidos. En la vida cotidiana, muchas situaciones no están fácilmente determinadas. Un buen ejemplo de esto es la temperatura en una habitación, donde no hay un valor exacto que se pueda clasificar como fría, templada o cálida. Cada persona puede tener una sensación distinta sobre qué temperatura corresponde a cada categoría. Sin embargo, lo que sí se puede hacer es establecer rangos para clasificar las temperaturas, como considerar debajo de 18°C como fría, entre 18°C y 22°C como templada, y por encima de 22°C como cálida. Estos rangos, por supuesto, no tienen límites estrictos y pueden variar según la percepción individual. Si la temperatura es de 21°C, siguiendo los criterios tradicionales, se clasificaría como templada, ya que cae dentro del rango estipulado. No obstante, la diferencia con una temperatura de 22°C es mínima, lo que hace que la clasificación no sea tan clara. La lógica difusa, en lugar de dar una respuesta absoluta, ofrece un grado de pertenencia a cada categoría, algo que se abordará en las siguientes secciones.

Este apartado introduce y explica el método de control que será empleado, destacando su relevancia frente a los métodos tradicionales, especialmente en escenarios donde la precisión matemática no es suficiente o no es viable debido a la naturaleza de los sistemas a controlar. La lógica difusa se presenta como una solución adaptada a la complejidad y a la flexibilidad requeridas en este tipo de casos.

## Estado del arte

Las bases del concepto de lógica difusa (fuzzy logic, en inglés) o lógica borrosa, tal como lo conocemos hoy en día, fueron establecidas en 1965 por el profesor Lotfi Asker Zadeh de la Universidad de California en Berkeley, en su artículo “Fuzzy Sets” [13]. De dicho artículo se puede inferir que Zadeh discrepaba del paradigma de la lógica clásica, lo que lo llevó a proponer un nuevo concepto: los conjuntos difusos. En esta teoría, un elemento puede pertenecer parcialmente a un conjunto, asignándole un valor numérico denominado grado de pertenencia, que va de 0 a 1 [2].

Años después, Zadeh profundizó aún más en su teoría con la publicación de otros dos artículos [3], [4], en los cuales introdujo conceptos adicionales como las variables lingüísticas y las reglas del tipo "If-then". Estos conceptos forman parte fundamental de la lógica difusa actual y serán explicados en detalle en el siguiente apartado, para su correcta comprensión.

No sería hasta 1974 cuando Ebrahim H. Mamdani diseñó el primer controlador difuso, cuyo propósito era controlar una máquina de vapor [5]. Este trabajo sentó las bases de uno de los métodos de inferencia más utilizados en la lógica difusa en la actualidad, además del más simple de entender. En este proyecto, el modelo usado es de este tipo.

Una década más tarde, en 1985, Tomohiro Takagi y Michio Sugeno propusieron un método alternativo de control difuso al desarrollado por Mamdani [6], siendo más eficiente computacionalmente pero más complejo de entender y usar. Aunque en este trabajo no se use un controlador de tipo Sugeno, sí que se le hará una mención en este apartado.

Desde entonces, se han realizado numerosas investigaciones y trabajos en los que se ha empleado el control difuso, allí donde le control clásico no consigue llegar.

## Marco teórico

Haciendo un repaso de los conceptos mencionados previamente, se procederá a explicar en detalle cada uno de ellos y el funcionamiento de la lógica difusa. Como se ha señalado, esta lógica es una herramienta que permite resolver problemas donde el objeto de estudio no puede describirse de forma precisa mediante una función de transferencia, pues es muy no lineal; o presenta una gran complejidad. Esto se logra utilizando un lenguaje más cercano al razonamiento humano, que no asigna valores absolutos en situaciones determinadas, como el ejemplo de la temperatura en la habitación. Por esta razón, se denomina lógica difusa o borrosa, ya que existen elementos que no pueden clasificarse dentro de los conjuntos clásicos debido a la falta de límites bien definidos. Este desafío fue abordado mediante la creación de los conjuntos difusos, la base de la lógica que abordaremos.

### Conjuntos difusos

Los conjuntos difusos se caracterizan por permitir que un elemento pertenezca a ellos en cierto grado, definido por un valor llamado grado de pertenencia (μ). A diferencia de los conjuntos clásicos, que tienen límites estrictos y solo aceptan pertenencia total (1) o ausencia total (0), los conjuntos difusos asignan a cada elemento un valor numérico dentro del intervalo [0,1]. Este valor indica el nivel de pertenencia del elemento al conjunto: un grado de 1 implica pertenencia completa, mientras que un grado de 0 indica que no pertenece en absoluto. Esta flexibilidad es la principal diferencia entre los conjuntos clásicos y los difusos, ya que estos últimos permiten describir situaciones más complejas donde cada elemento puede tener un determinado grado parcial de pertenencia a varios conjunto [7].

### Variables lingüísticas

Lotfi Zadeh definió las variables lingüísticas en uno de sus artículos [4] como: “A linguistic variable is defined as a variable whose values are sentences in a natural or artificial language. Thus, if tall, not tall, very tall, very very tall, etc. are values of height, then height is a linguistic variable” [Una variable lingüística se define como una variable cuyos valores son sentencias en un lenguaje natural o artificial. Así, si alto, no alto, muy alto, muy muy alto, etc. son valores de la altura, entonces la altura es una variable lingüística].

De esta definición se puede concluir que las variables lingüísticas son aquellas en las que los valores que pueden tomar no son numéricos, sino palabras o expresiones que representan un intervalo cuantitativo difuso, es decir, un conjunto difuso definido. Este enfoque se asemeja más al lenguaje utilizado por las personas para describir características específicas. Por ejemplo, recordando el caso de la temperatura en la habitación, la variable lingüística sería la temperatura, cuyos valores eran fría, templada o cálida.

### Funciones de membresía

Las funciones de pertenencia son las que definen a los conjuntos difusos, ya que establecen los límites imprecisos de estos conjuntos y determinan el grado de pertenencia de un elemento al mismo. Entre los diferentes tipos de funciones de pertenencia se encuentran la triangular, trapezoidal, gaussiana, sigmoidal y la de forma de campana. Las funciones triangulares y trapezoidales son las más utilizadas debido a su simplicidad y naturaleza lineal, aunque en ciertos casos es más adecuado recurrir a otras funciones, dependiendo de los requisitos específicos [8].

Para una variable lingüística, será necesario definir tantas funciones de pertenencia como conjuntos difusos sean necesarios para abarcar todos los posibles valores del universo de discurso de esa variable. Esto significa que cada función representará un rango de valores cuantitativos asociado a esa variable en el contexto del caso de estudio.

La mejor manera de mostrar la idea es mediante un ejemplo, por eso, haremos uso del caso de la temperatura en la habitación.

|  |
| --- |
| Gráfico, Gráfico de líneas  Figura 4‑1. Ejemplo de funciones de membresía (MF) para el caso de la temperatura en una habitación |

Como se puede observar, se han definido tres funciones de pertenencia que corresponden a los valores de la variable lingüística temperatura en el ejemplo. En este caso, se han utilizado funciones de pertenencia triangulares, aunque las funciones correspondientes a frío y cálido no presentan una arista definida en sus extremos. Esto se debe a que se ha querido otorgar el mayor grado de pertenencia a los valores más extremos: todas las temperaturas cercanas a 15°C son consideradas frías, mientras que las superiores a 30°C se clasifican como cálidas. También se puede apreciar cómo hay regiones en las que dos funciones de pertenencia se solapan, reflejando el hecho de que no existe un valor exacto que marque dónde termina un rango y comienza el otro.

Existen diferentes métodos para definir las funciones de pertenencia, siendo uno de ellos la intuición o el conocimiento de un experto en el sistema, siendo estos los más comunes. Otros métodos incluyen el uso de redes neuronales o algoritmos genéticos [9].

### Reglas difusas

Para diseñar un controlador difuso, es necesario definir un conjunto de reglas difusas basadas en el conocimiento experto sobre el ámbito del problema en estudio. Estas reglas conforman una base de conocimiento que guía el funcionamiento de la lógica difusa. Las reglas se expresan en formato IF-THEN, estableciendo relaciones entre los conjuntos difusos que aparecen en las premisas de la regla y el conjunto difuso que determina su conclusión [7]. La estructura general de estas reglas es la siguiente:

|  |  |
| --- | --- |
|  | (5-1) |

Las proposiciones difusas pueden clasificarse como atómicas o compuestas [10]. Una proposición atómica consiste en una asignación simple, como “x es A”, donde x representa una variable lingüística y A es un conjunto difuso definido mediante una función de pertenencia, por ejemplo, una función triangular. Por otro lado, las proposiciones difusas compuestas combinan varias proposiciones atómicas utilizando operadores lógicos como *and, or y not.* Un ejemplo de regla con una proposición compuesta podría ser: “Si X es A y Y es B, entonces Z es C”, donde X, Y y Z son variables lingüísticas (X e Y son entradas, mientras que Z es la salida), y A, B, C son conjuntos difusos definidos para dichas variables.

Estas reglas permiten establecer relaciones entre las entradas del controlador y las salidas que este genera, formando así la base de conocimiento mencionada previamente. Todas las reglas están formuladas lingüísticamente, sin emplear valores numéricos, lo que hace bastante simple plantearlas para una persona. Por ejemplo, si se busca ajustar la intensidad de la iluminación en una habitación dependiendo de la cantidad de luz natural, una posible regla sería: “Si Luz\_natural es Baja, entonces Intensidad\_luz es Alta”.

#### Operaciones lógicas

Como se ha mencionado, las proposiciones en este tipo de reglas pueden formarse mediante operaciones lógicas entre conjuntos difusos, por lo que es importante entender cómo funcionan dichas operaciones. Las tres operaciones lógicas más comunes son *and, or y not*. También se puede comprender el propio *then* (implicación) al que se le suelen aplicar métodos semejantes que a *and.*

Existen diferentes métodos que pueden emplearse para implementar las operaciones lógicas en lógica difusa, y su elección depende del caso de estudio o las especificaciones del sistema. Entre los métodos más comunes para la operación and se encuentran el mínimo (min) y el producto (prod), mientras que para or se utilizan el máximo (max) y la suma acotada (bsum). Por su parte, para la operación not se suele emplear el complemento estándar . En el caso de las reglas then, los métodos más utilizados son el producto y el mínimo, dependiendo del enfoque que se quiera dar al cálculo del grado de pertenencia en la inferencia difusa. En este proyecto, nos centraremos en los métodos específicos seleccionados para implementar estas operaciones, los cuales corresponden a la siguiente tabla:

Tabla 4‑1. Operaciones lógicas del controlador

|  |  |  |  |
| --- | --- | --- | --- |
| ***And*** | ***Or*** | ***Not*** | ***Then*** |
| ***Min*** | ***Max*** |  | ***Min*** |

Una vez aclarado esto, en la siguiente figura se explican cada una de ellas:

|  |
| --- |
| Figura 4‑2. Operadores lógicos aplicados a conjuntos difusos [11]. |

## Controlador difuso

Haciendo un repaso de los conceptos mencionados previamente, se procederá a explicar en detalle cada uno de ellos y el funcionamiento de la lógica difusa. Como se ha señalado, esta lógica es una herramienta que permite resolver problemas donde el objeto de estudio no puede describirse de forma precisa mediante una función de transferencia, pues es muy no lineal (como es nuestro caso); o presenta una gran complejidad. Esto se logra utilizando un lenguaje más cercano al razonamiento humano, que no asigna valores absolutos en situaciones determinadas, como el ejemplo de la temperatura en la habitación. Por esta razón, se denomina lógica difusa o borrosa, ya que existen elementos que no pueden clasificarse dentro de los conjuntos clásicos debido a la falta de límites bien definidos. Este desafío fue abordado mediante la creación de los conjuntos difusos, la base de la lógica que abordaremos.

|  |
| --- |
| Figura 4‑3. Esquema de un controlador difuso [12]. |

Para aclarar el funcionamiento de controlador se hablará de cada una de las etapas por las que se pasa en él expuesta en la figura.

### Fuzzificación

El proceso de fuzzificación consiste en transformar las señales de entrada, que son valores numéricos, en señales difusas. Esto se logra definiendo los conjuntos difusos asociados a las variables lingüísticas que representan las entradas del sistema. A través de las funciones de pertenencia correspondientes, se determina el grado de pertenencia de cada entrada a los conjuntos difusos, lo cual es fundamental para la evaluación de las reglas difusas en el sistema [13].

### Base de reglas

La base de reglas consiste en establecer un conjunto de reglas del tipo IF-THEN, las cuales, como se mencionó anteriormente, se construyen a partir del conocimiento o experiencia de un experto en el ámbito de estudio, utilizando razonamiento difuso.

Dependiendo de la forma de la proposición en el consecuente, las reglas pueden clasificarse en dos tipos principales. Si el consecuente utiliza conjuntos difusos, se trata de reglas del tipo **Mamdani**. Por otro lado, si el consecuente está definido mediante una ecuación lineal o una función matemática, corresponde a reglas del tipo **Takagi-Sugeno** [10].

### Mecanismos de inferencia

El mecanismo de inferencia es el proceso mediante el cual se evalúan las entradas del sistema en función de todas las reglas definidas, generando una salida en formato difuso.

Dentro de este proceso, la implicación de las reglas consiste en generar el consecuente de una regla a partir de su antecedente. Para ello, se utilizan distintos métodos de implicación (*then*), antes mencionados.

El método del mínimo, que es el empleado en el presente trabajo, recorta o trunca el conjunto difuso de salida, limitando los grados de pertenencia según el valor más bajo indicado por el antecedente. Por otro lado, el método del producto preserva la forma original de la función de pertenencia escalándola; esto se logra multiplicando cada grado de pertenencia de la función de salida por el grado de pertenencia indicado por el antecedente [13].

|  |
| --- |
| Figura 4‑4. Ejemplo de evaluación de reglas [13]. |

En la Figura 5-4 se pueden ver dos pasos:

* **Activación de la regla:** Es el paso ya mencionado previamente, en el cual, el consecuente es truncado con respecto al mínimo de los antecedentes.
* **Acumulación:** Este paso depende del método de agregación seleccionado, en el caso que nos concierne, se aplica el método **max**, que como se puede apreciar es la selección de la regla con mayor grado de pertenencia.

### Defuzzificación

La defuzzificación es el proceso inverso a la fuzzificación, en el cual la salida difusa generada por el mecanismo de inferencia se convierte en un valor numérico que representa la salida final del controlador (*crisp)*. Existen diversos métodos para llevar a cabo este proceso, entre los más destacados se encuentran los siguientes:

* **Centroide o centro de gravedad (center of areas):** Este método calcula el valor numérico de salida como el centroide del conjunto difuso resultante tras la agregación. Es el más utilizado en aplicaciones de control debido a su precisión y consistencia [13] [7]. Este método es el que ha sido empleado en el presente trabajo.
* **Bisectriz o áreas iguales (equal areas):** Determina el valor numérico como aquel que divide el conjunto difuso de salida en dos partes con áreas iguales [7]
* **Máximo inferior (lower maximum):** Se selecciona el primer valor del conjunto de salida en el que se alcanza el máximo grado de pertenencia, en caso de haber varios picos máximos [7]
* **Máximo superior (upper maximum):** Similar al anterior, pero se elige el último valor en el que se produce el máximo grado de pertenencia [7]
* **Media de máximos o centro de máximos (mean of maxima):** Este método toma como resultado el punto medio entre el máximo inferior y el máximo superior del conjunto de salida [7].

Cabe aclarar que independientemente del método empleado, el valor del **crisp s**e obtiene por una discretización de los puntos de la salida. El caso más simple de exponer es el del centroide:

Tabla 4‑2. Ecuaciones del centroide

|  |  |
| --- | --- |
|  | (5-3) |
|  |  |

|  |
| --- |
| Figura 4‑5. Aplicación del método del centroide [13]. |

Tras todo esto se muestra el resultado de aplicar todos los pasos en un caso particular, el del pago de una propina en base al servicio y la comida:

|  |
| --- |
| Figura 4‑6. Proceso completo de operaciones de un controlador fuzzy [14]. |

## Controlador difuso en MATLAB

### Creación del archive\*.fis

Tras haber explicado todos los conceptos necesarios para comprender esta metodología y sus aplicaciones actuales, es pertinente mencionar brevemente cómo trabajar con lógica difusa en MATLAB. El primer paso consiste en hacer uso de la app FLD. Este editor facilita la creación y gestión de sistemas difusos, permitiendo configurar entradas, salidas, reglas y funciones de pertenencia de forma visual.

Una vez completado el diseño, se puede guardar el sistema difuso como un archivo con extensión \*.fis en el menú, simplemente guardándolo como archivo dentro de la carpeta de trabajo. El archivo es en esencia un archivo de texto plano en el que se contiene toda la información del controlador, como el tipo, las reglas, las MFs, etc.

|  |
| --- |
| Figura 4‑7. FLD |

### I/O y funciones de membresía

El siguiente paso consiste en definir las variables de entrada (inputs) y salida (Output) del controlador difuso, así como sus correspondientes MFs. Esto se realiza seleccionando la variable deseada en el FLD, lo que abrirá automáticamente el editor de funciones de membresía (Figura 5-8).

En este editor, se puede configurar el tipo de función de membresía que mejor se adapte al comportamiento de las variables del sistema. Entre las opciones disponibles se encuentran las funciones triangular, trapezoidal, sigmoidal y gaussiana, entre otras. La elección de la función dependerá de las características y requisitos específicos del controlador que se esté diseñando.

|  |
| --- |
| Figura 4‑8. Editor de MFs |

### Reglas de inferencia

Una vez configuradas todas las variables del controlador difuso, el siguiente paso es definir las IRs que determinarán su comportamiento. Para ello, se debe hacer clic en el cuadro de “Rule Editor” de la ventana del FLD (Figura 3-8).

Como se mencionó en apartados anteriores, estas reglas siguen el formato "IF-THEN", y se pueden crear tantas como sean necesarias, considerando que el número máximo está limitado por la cantidad de variables y funciones de membresía definidas. Un aspecto relevante es la posibilidad de asignar un peso (*weight*) o importancia a cada regla. Este valor numérico debe estar entre 0 y 1, permitiendo que una regla tenga menos influencia que el resto si se asigna un peso inferior a 1. No obstante, es habitual que todas las reglas se configuren con el mismo nivel de importancia, siendo este el caso del controlador generado.

|  |
| --- |
| Figura 4‑9. Editor de reglas de inferencia |

### Visualizador de reglas

Después de definir las reglas lógicas y guardar el archivo, la configuración del controlador difuso queda completa y está listo para implementarse en Simulink. No obstante, antes de llevarlo a cabo, es posible evaluar el comportamiento de las reglas mediante dos visualizadores (“Rule inference” y “Control Surface”) integrados en el FLD. Estos permiten observar gráficamente cómo interactúan las variables de entrada con las reglas establecidas y cómo se genera la salida difusa, lo que resulta útil para validar el diseño antes de su implementación final.

Como se puede observar, el visualizador presenta varios gráficos en función de las entradas y salidas del controlador. En cada columna correspondiente a las entradas, se visualizarán tantos gráficos como reglas se hayan configurado. Además, en cada columna de salidas, habrá un gráfico adicional que muestra el valor resultante de la salida difusa, el cual estará representado por una línea roja más gruesa.

Esta herramienta es de gran utilidad, ya que permite verificar de forma aislada si el controlador funciona según lo esperado, sin interferencias de otros elementos del modelo en Simulink. Si el controlador no produce los resultados deseados al simularlo en el entorno de Simulink, el visualizador ayuda a determinar si el problema radica en la implementación técnica o en la validez de las reglas o funciones de membresía utilizadas. Esto facilita la corrección de errores en etapas tempranas del diseño y prueba del sistema.

|  |
| --- |
| Figura 4‑10. Rule Inference |
| Figura 4‑11. Control Surface |

### Implemetación del controlador en Simulink

Para integrar el archivo de controlador difuso que ha sido creado previamente en FLD, simplemente hay que utilizar el bloque FLC. En el interior de este bloque, se debe especificar el nombre del archivo con su extensión \*.fis entre comillas simples, incluyendo la ruta completa si es que no se encuentra en la misma carpeta que el modelo.

|  |
| --- |
| Figura 4‑12. FLC |

# Análisis del modelo e implementación

A

ntes de implementar un sistema de control en un entorno real, es crucial realizar simulaciones que permitan analizar su comportamiento y prever posibles problemas. Las simulaciones proporcionan un entorno seguro y controlado para probar estrategias de control sin arriesgar daños a los equipos ni comprometer la seguridad de las operaciones. Esto es especialmente importante cuando se trabaja con sistemas complejos, donde los errores pueden ser costosos o peligrosos.

La precisión en el modelado del sistema es un factor fundamental para garantizar que los resultados de las simulaciones sean representativos de la realidad. Un modelo que no capture correctamente las dinámicas del sistema puede llevar a conclusiones erróneas sobre su comportamiento y, por ende, a estrategias de control ineficaces. Por esta razón, es imprescindible dedicar tiempo y recursos a desarrollar un modelo que refleje lo más fielmente posible las características físicas y operativas del sistema. Además, un modelado preciso influye directamente en la elección de las estrategias de control. Los modelos permiten evaluar diferentes enfoques y determinar cuál se adapta mejor a las particularidades del sistema. Por ejemplo, en sistemas con comportamientos no lineales o dinámicas variables, métodos de control tradicionales como los PID pueden no ser suficientes, y es aquí donde estrategias como la lógica difusa cobran relevancia. El modelo actúa como un laboratorio virtual en el que se pueden ajustar parámetros, identificar limitaciones y optimizar el rendimiento del controlador antes de trasladarlo al entorno real.

En este contexto, se desarrollará un modelo detallado del sistema, el cual servirá como base para diseñar, ajustar y validar el controlador difuso propuesto en este estudio. A través de este modelo, se podrá evaluar el desempeño del controlador, asegurando que las soluciones implementadas sean efectivas y seguras para su aplicación final.

## Esquema del modelo

|  |
| --- |
| Figura 5‑1. Esquema de fuerzas [15] |

En la figura se muestra el esquema en el que se basa el modelo para, a través de las ecuaciones de Euler-Lagrange, establecer la dinámica del sistema a controlar. En el esquema se pueden diferenciar los siguientes elementos:

* **Trolley *(M)*:** Es el carro móvil de la grúa que se mueve para transportar la carga.
* **Rope *(l)*:** Es la cadena o cuerda que une la carga al carro. En este caso se asume inelástica.
* **Load *(m)*:** Es la carga a transporta de un punto *a* a otro punto *b* del eje *x*.
* **Position *(x)*:** Es la posición relativa con respecto al eje *x* del sistema de coordenadas.
* **Angle *(θ)*:** Es el ángulo que forma la carga con respecto al eje *y* del sistema de coordenadas.
* **Force *(F)*:** Es la fuerza que induce el movimiento sobre el carro. Esta fuerza sería la que ejercería el controlador.
* **COF *(μ)*:** Es el coeficiente de rozamiento dinámico existente entre el carro y la superficie por la que se desplaza.

## Modelo dinámico en el dominio del tiempo

Como se ha comentado previamente el método empleado para obtener las ecuaciones del sistema físico, son las ecuaciones de Euler-Lagrange. Se escogió este método pues resulta más esquemático y fácil de aplicar en un sistema complejo como este que la aplicación directa de las leyes de newton sobre los diferentes cuerpos del sistema.

### Ecuaciones de Euler-Lagrange

En lugar de utilizar las fuerzas y leyes de Newton directamente, las ecuaciones de Lagrange trabajan con **energías**: la energía cinética *T* y la energía potencial *V*. Este enfoque es particularmente útil para sistemas con múltiples grados de libertad y restricciones porque permite describir el movimiento en términos de coordenadas generalizadas.

#### Principio fundamental

Las ecuaciones de Lagrange se derivan del **principio de Hamilton**, que establece que el sistema sigue una trayectoria que minimiza (o hace estacionaria) la acción *S*, definida como:

|  |  |
| --- | --- |
|  | (5-1) |

donde *L* es la **Lagrangiana**:

|  |  |
| --- | --- |
|  | (5-2) |

Donde *T* es la energía cinética y *V* es la energía potencial.

#### Fórmula general

Para un sistema con coordenadas generalizadas y sus derivadas temporales , las ecuaciones de Lagrange se escriben como:

|  |  |
| --- | --- |
|  | (5-3) |

Los términos son:

* : Momento generalizado asociado a .
* ​: Fuerza generalizada correspondiente.

#### Ventajas y razón de su uso

* **Flexibilidad**: Permite trabajar con coordenadas que no son necesariamente cartesianas, lo que es útil en sistemas con restricciones (por ejemplo, movimiento en superficies curvas).
* **Reducción de complejidad**: Evita calcular fuerzas de restricción explícitamente, pues estas se incorporan automáticamente en las ecuaciones.

### Ecuaciones del sistema

Lo primero es establecer las coordenadas en las que nos basaremos para exponer las ecuaciones. Las coordenadas se encuentran en la siguiente tabla:

Tabla 5‑1. Coordenadas de ecuaciones E-L

|  |  |
| --- | --- |
| *q1* | *q2* |
| x | θ |

Describimos la posición de ambos cuerpos:

|  |  |  |
| --- | --- | --- |
|  |  | (5-4) |
|  |  |  |

Establecemos la energía cinética y potencial del sistema para obtener la lagrangiana:

|  |  |
| --- | --- |
|  | (5-5) |
|  |  |
|  |  |

Aplicamos la fórmula de la lagrangiana para cada coordenada:

|  |  |  |
| --- | --- | --- |
|  |  | (5-6) |
|  |  |  |

Con esto ya poseemos las ecuaciones de la dinámica del sistema. Como se puede ver el sistema es muy no lineal (términos trigonométricos) y además existe acoplamiento, lo que lo hace más complejo de analizar. Además, es un sistema MIMO por lo que haciéndolo difícilmente representable a través de una función de transferencia *G(s)*. Por todo ello se ha elegido implementarlo como un SS.

## Espacio de estados

La representación mediante espacio de estados de un sistema posee las siguientes ecuaciones:

|  |  |
| --- | --- |
|  | (5-7) |

Pero esto solo es aplicable a sistemas linealizados, algo que en nuestro caso se puede hacer aplicando unas ciertas simplificaciones como las siguientes:

|  |  |
| --- | --- |
|  | (5-8) |
|  |  |
|  |  |

Estas aproximaciones surgen de la idea de que el ángulo en el sistema está limitado en un rango reducido pudiendo aproximarse los valores tal y como se puede ver en las ecuaciones (5-8). Esto dejaría un sistema como el siguiente:

|  |  |
| --- | --- |
|  | (5-9) |
|  |  |

De estas ecuaciones se obtiene fácilmente la expresión de espacio de estados de nuestro sistema:

|  |  |
| --- | --- |
|  | (5-10) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Este modelo es perfectamente válido ya que la idea subyacente del control es provocar precisamente que el ángulo de la carga este contenido en un rango reducido, pudiendo aplicar así la linealización expuesta. Aun así, el modelo empleado en Simulink no es el expuesto previamente. El modelo empleado mantiene las no linealidades, haciéndolo más preciso que modelo lineal expuesto. La razón por la cual se expone este modelo es porque fue el usado originalmente hasta que se dio con este otro que mantenía las no linealidades.

|  |
| --- |
| Figura 5‑2. Modelo empleado en Simulink [15]. |

## Conclusión

En resumen, hemos obtenido las ecuaciones de la dinámica del sistema en el dominio del tiempo utilizando las ecuaciones de Euler-Lagrange, lo que permitió describir el comportamiento físico del sistema de manera detallada. Posteriormente, representamos estas ecuaciones en el formato de espacio de estados, dado que el sistema analizado es de tipo MIMO (Multiple-Input Multiple-Output). Este paso fue esencial para preparar el modelo para su implementación en estrategias de control.

Para simplificar el análisis y diseño, se aplicaron diversas simplificaciones que nos permitieron linealizar el modelo. Esto resultó en la obtención de las matrices del sistema linealizado, que son cruciales para el diseño de controladores que asuman comportamientos lineales, como los clásicos PIDs o basados en el lugar de las raíces. Sin embargo, esta simplificación implica una pérdida de precisión en la descripción del comportamiento del sistema, especialmente en regímenes de operación alejados del punto de equilibrio considerado.

Finalmente, se recurrió a un modelo alternativo extraído de la literatura científica [15] que mantiene las no linealidades inherentes al sistema. Este modelo ofrece una representación más precisa y realista del comportamiento dinámico, especialmente útil para simulaciones y diseño de controladores que aprovechen características no lineales, como el caso de los controladores difusos *(FLC)*.

# Referencias

|  |  |
| --- | --- |
| [1] | M. d. Trabajo, «Estadísticas de accidentes de trabajo,» [En línea]. Available: https://www.mites.gob.es/es/estadisticas/condiciones\_trabajo\_relac\_laborales/EAT/welcome.htm. [Último acceso: 25 11 2024]. |
| [2] | L. A. Zaddeh, «Fuzzy sets,» *Information and control,* vol. 8, nº 3, pp. 338-353, 1965. |
| [3] | L. A. Zadeh, «Outline of a new approach to the analysis of complex systems and decision processes,» *IEEE Transactions on systems, Man, and Cybernetics,* vol. 3, nº 1, pp. 28-44, 1973. |
| [4] | L. A. Zadeh, «The concept of linguistic variable and its application to approximate reasoning,» *Information sciences,* vol. 8, nº 3, pp. 199-249, 1975. |
| [5] | E. H. Mamdani, «Application of fuzzy algorithms for control of simple dynamic plant,» *Proceedings of the institution of electrical engineers,* vol. 121, nº 12, pp. 1585-1588, 1974. |
| [6] | T. Takagi y M. Sugeno, «Fuzzy identification of systems and its applications to modeling and control,» *IEEE transactions on systems, man, and cybernetics,* vol. 15, nº 1, pp. 116-132, 1985. |
| [7] | I. H. Altas, Fuzzy Logic Control in Energy Systems: with Design Applications in MATLAB/Simulink, London: The Institution of Engineering and Technology, 2017. |
| [8] | M. H. Hazam, M. H. Hasan, S. Hassan y S. J. Abdulkadir, «Fuzzy Type-1 Triangular Membership Function Approximation Using Fuzzy C-Means,» de *International Conference on Computational Intelligence (ICCI)*, Malaysia, 2020. |
| [9] | P. N. Nwankwo, K. Akpado y C. C. Okezie, «Development of an Advanced Neuro-Fuzzy Algorithm for Intelligent Temperature Control System,» *International Journal of Advances in Engineering and Management (IJAEM) ,* vol. 6, nº 9, pp. 769-791, 2024. |
| [10] | J. C. G. Infante, *Sistemas con Lógica Difusa,* México: Instituto Politécnico Nacional, 2009. |
| [11] | MathWorks, «Foundations of Fuzzy Logic,» [En línea]. Available: https://es.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html. [Último acceso: 2 12 2024]. |
| [12] | D. Pacheco Bautista, E. Cortes Rico y F. Aguilar Acevedo, «Diseño de un controlador de carga de tres etapas para sistemas fotovoltaicos usando lógica difusa,» *Ingeniare,* vol. 27, nº 4, pp. 540-550, 2019. |
| [13] | J. Jaekel, R. Mikut y G. Bretthauer, «Fuzzy Control Systems,» de *Control Systems, Robotics, and Automation*, Karlsruhe, H. Unbehauen, 2004, pp. 1-30. |
| [14] | MathWorks, «Fuzzy Inference Process,» [En línea]. Available: https://es.mathworks.com/help/fuzzy/fuzzy-inference-process.html. [Último acceso: 2 12 2024]. |
| [15] | N. T. Nguyen, D. H. Nguyen, T. L. Nguyen y T. V. A. Nguyen, «Takagi-Sugeno Fuzzy Approach with Compressed Representation,» *Smart Systems and Devices,* vol. 32, nº 3, pp. 44-51, 2022. |

# **Índice de Conceptos**

conceptos 9

# **Glosario**

ISO: International Organization for Standardization 4

UNE: Una Norma Española 4