



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Control difuso

© Fernando Berzal, berzal@acm.org

Control difuso



- Control
- Control clásico (PID)
- Control difuso
- Controladores tipo Mamdani vs. tipo Sugeno (TSK)
- Control neurodifuso



Control



¿Qué es el control?

Hacer que un proceso dinámico se comporte como nosotros queramos.

¿Qué necesitamos?

- Una forma de **influnciar** el proceso.
- Una forma de **observar** cómo se comporta el proceso.
- Una forma de **definir** cómo queremos que se comporte.



Control



Cómo definir el comportamiento del sistema

Usualmente, estableciendo el valor deseado para una variable de salida del sistema [**setpoint**].

EJEMPLO: Temperatura

Cómo influenciar el proceso

Mediante una entrada de control que puede crear cambios en el sistema bajo control (típicamente, distinta a la variable de salida que deseamos controlar).

EJEMPLO: Interruptor del aire acondicionado





Cómo observar el comportamiento del sistema

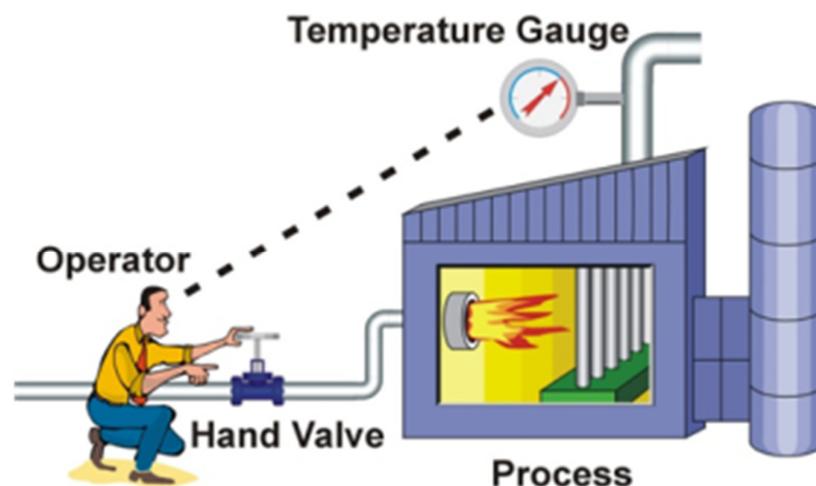
Si supiésemos con exactitud cómo funciona el proceso, sabríamos que entrada de control es necesaria para obtener la salida deseada...

... pero normalmente no es así, por lo que necesitamos medir el estado del proceso (usando variables de medida, también conocidas como variables de proceso).



Control con realimentación [feedback]

- Variables de medida: MV [measurement variables]
- Variables de control: SP [setpoint]
- Variables de salida: Señales de control



Control

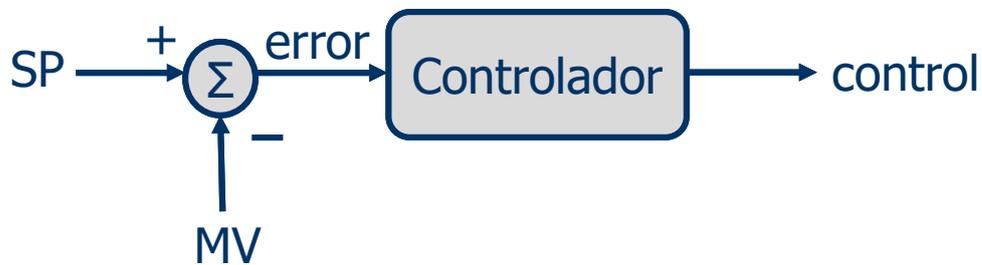


El controlador



Señal de error

Podemos interpretar que el controlador, en realidad, actúa sobre la diferencia entre el valor deseado [SP] y el valor medido [MV]:



Control



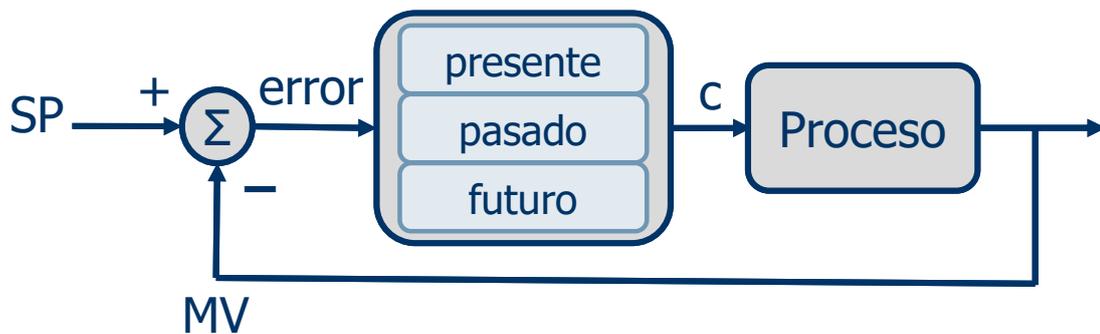
Control de bucle cerrado [closed loop]



Control clásico



Dado que el proceso es dinámico, depende de las entradas en distintos instantes de tiempo, el controlador también debería ser dinámico...



... depende del presente, del pasado y del futuro



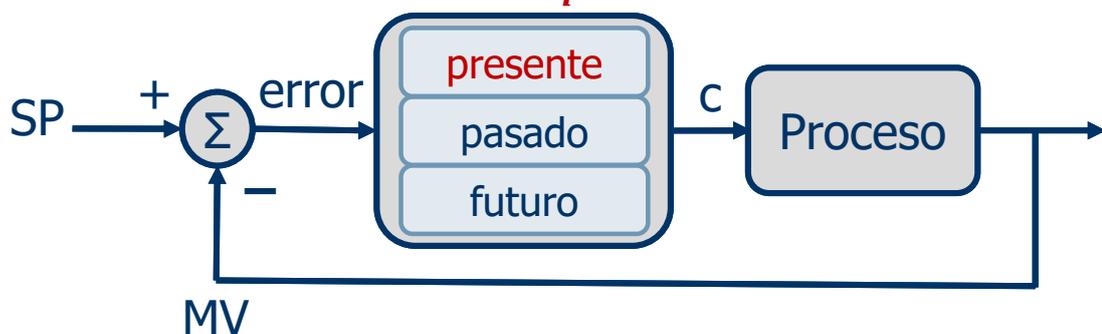
Control clásico



Control proporcional (P)

En el presente, la salida del controlador debería depender del error actual: señal de control proporcional al error.

$$\text{control} = k_p \times \text{error}$$



Primer parámetro del controlador:
Ganancia proporcional k_p





Control proporcional (P)

Problemas

- **Oscilaciones**

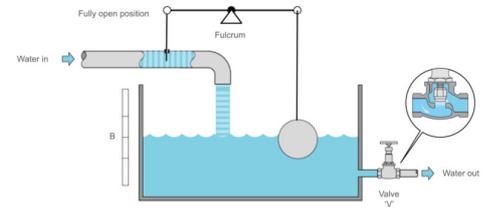
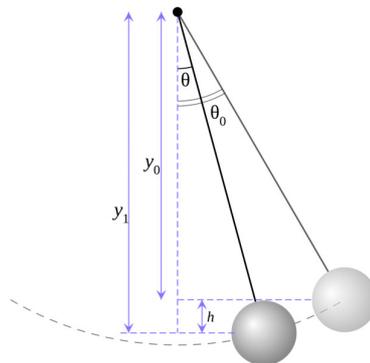


Fig. 5.2.6 Valve open

- **Error estacionario** [e_{ss} , steady state error]:

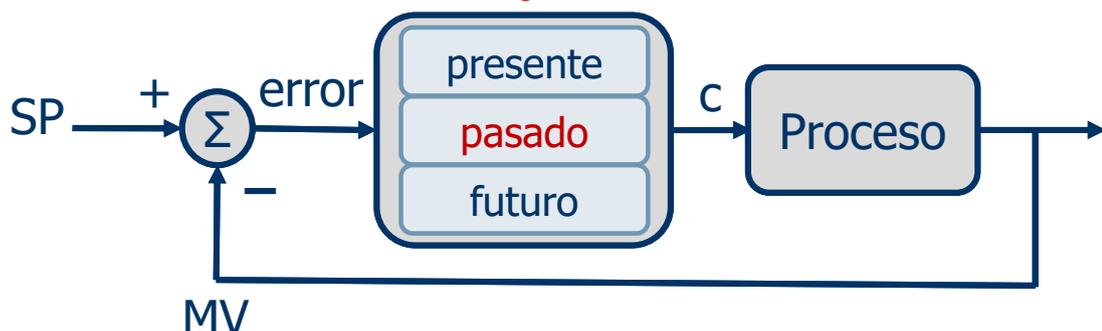
Si el error es cero, la salida del controlador también lo es (¿y si necesitamos que no lo sea?)



Control integral (I)

Para resolver los problemas del control proporcional, fijémonos en qué sucedió en el pasado...

$$\text{control} = k_i \int_0^t \text{error}(\tau) d\tau$$

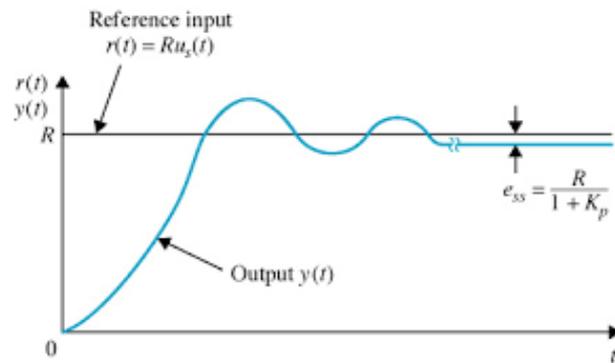


Ganancia integral k_i

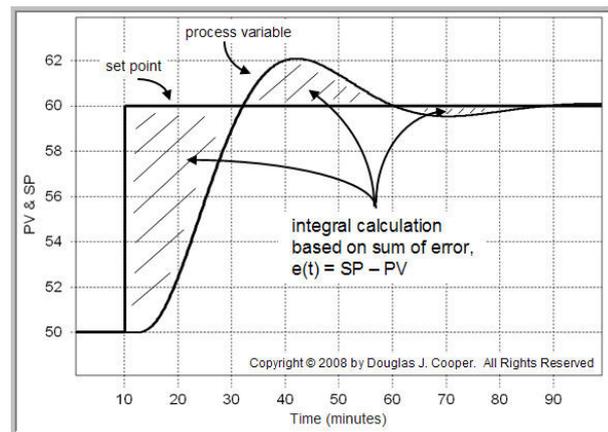




Control integral (I)

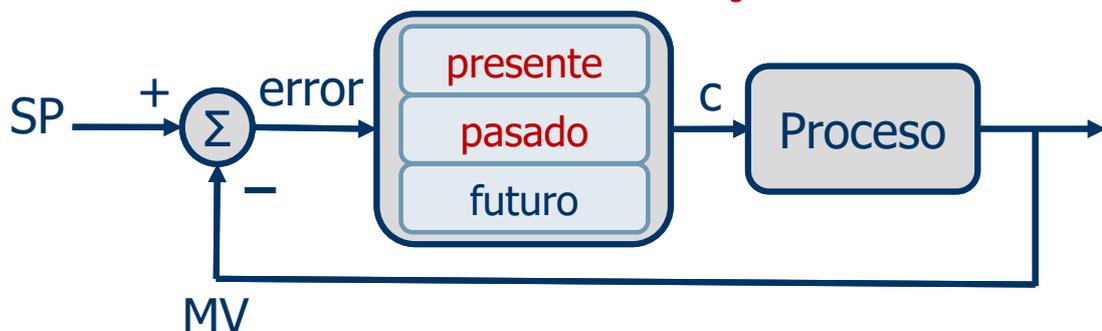


La ganancia integral convierte la suma de errores pasados en una señal de control.



Control PI [proporcional + integral]

$$c(t) = k_p \times e(t) + k_i \int_0^t e(\tau) d\tau$$



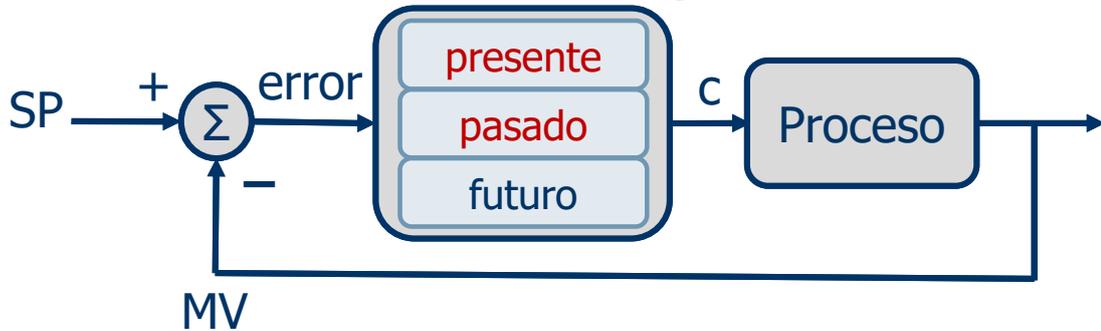
Ganancia integral k_i





Control PI [proporcional + integral]

$$c(t) = k_p \times e(t) + \frac{T_i}{k_p} \int_0^t e(\tau) d\tau$$



Tiempo integral T_i

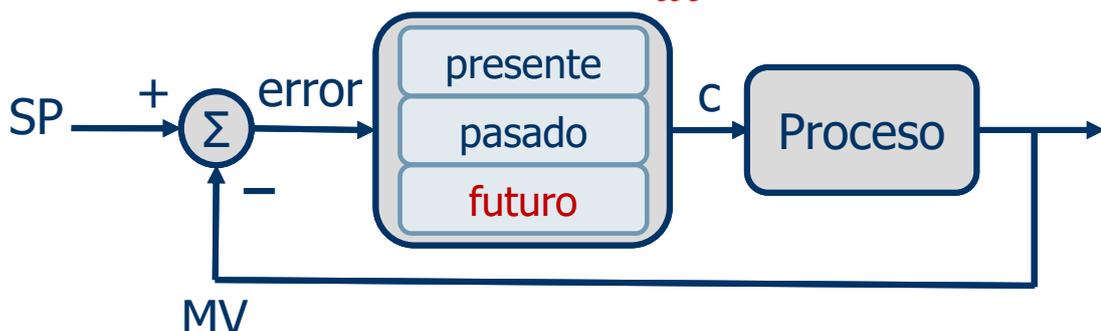
La ganancia integral k_i expresada en unidades de tiempo (cómo de rápido se corrige el error SSE)



Control derivativo (D)

Para resolver el problema de las oscilaciones, realicemos una predicción sobre el futuro...

$$\text{control} = k_d \frac{de(t)}{dt}$$



Ganancia derivativa k_d



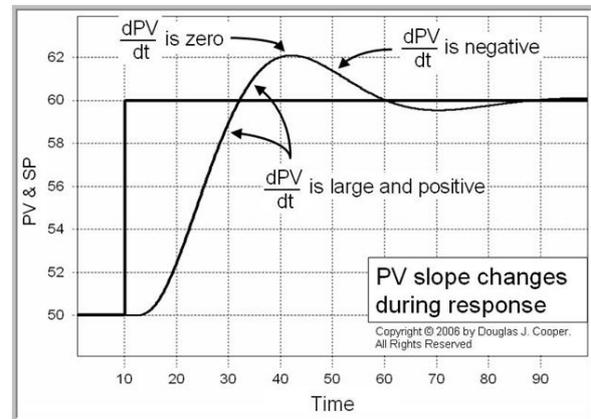
Control clásico



Control derivativo (D)

¿Cómo predecimos el error en el futuro?

Mirando su gradiente...



El gradiente del error (su derivada en el tiempo) nos da una dirección en la que podemos reducir el error.

¡OJO! La derivada amplifica cambios rápidos (p.ej. ruido en la entrada), por lo que se suele añadir un filtro paso bajo.

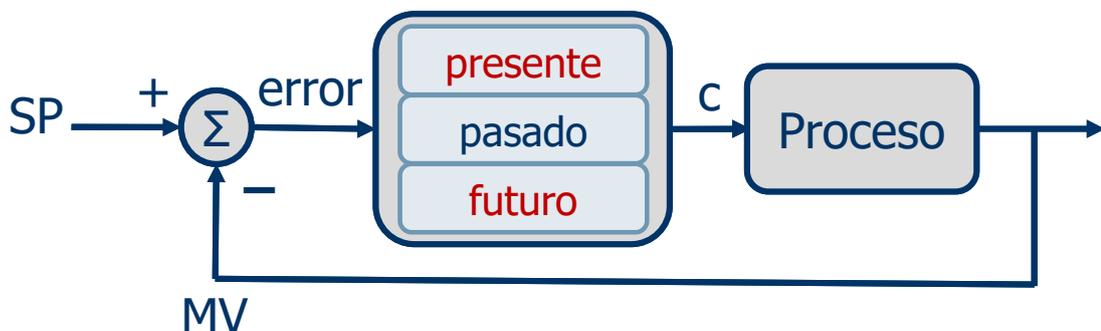


Control clásico



Control PD [proporcional + derivativo]

$$c(t) = k_p \times e(t) + k_d \frac{de(t)}{dt}$$



Ganancia derivativa k_d



Control clásico

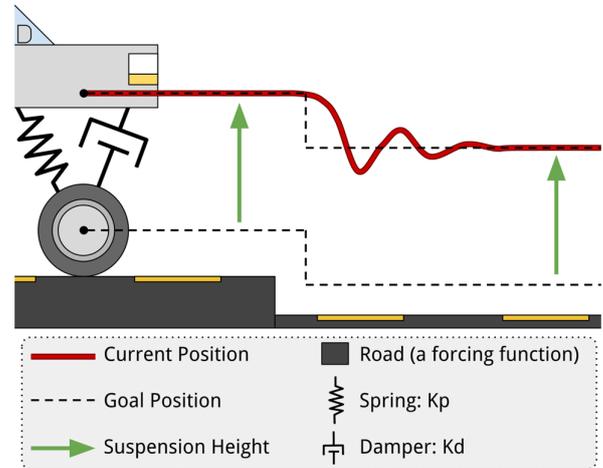


Control PD [proporcional + derivativo]

Efecto amortiguación

El muelle hace de control proporcional para la rueda.

El amortiguador se opone a sus cambios de posición.



Control clásico

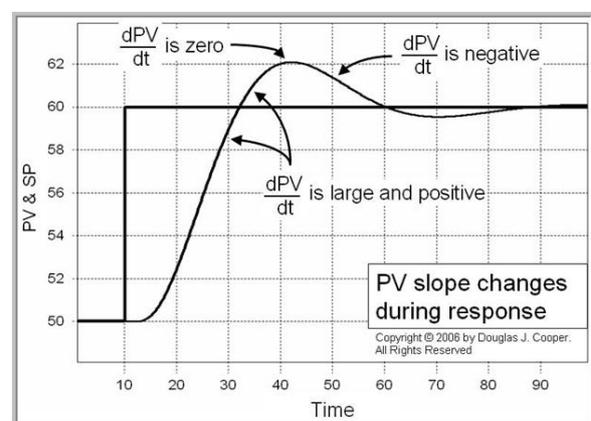


Control PD [proporcional + derivativo]

También podemos expresar la ganancia derivativa en términos de tiempo:

$$k_d = k_p T_d$$

T_d nos indica hasta dónde miramos en el futuro

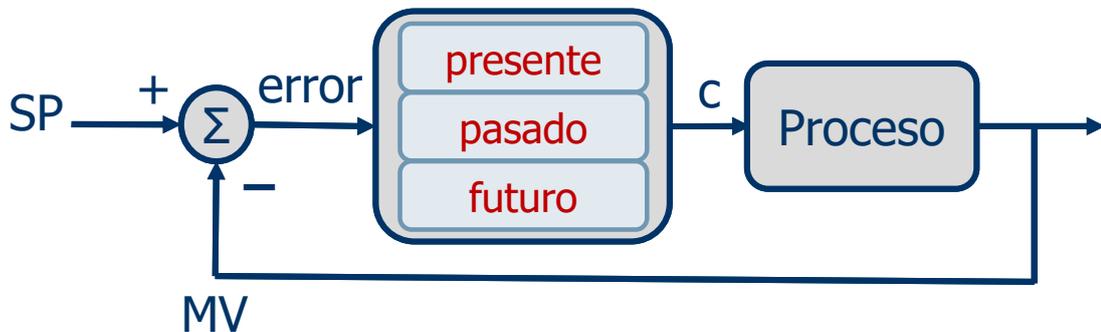


Control clásico



Control PID [proporcional + integral + derivativo]

$$c(t) = k_p \times e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$



Tres ganancias k_p , k_i , k_d

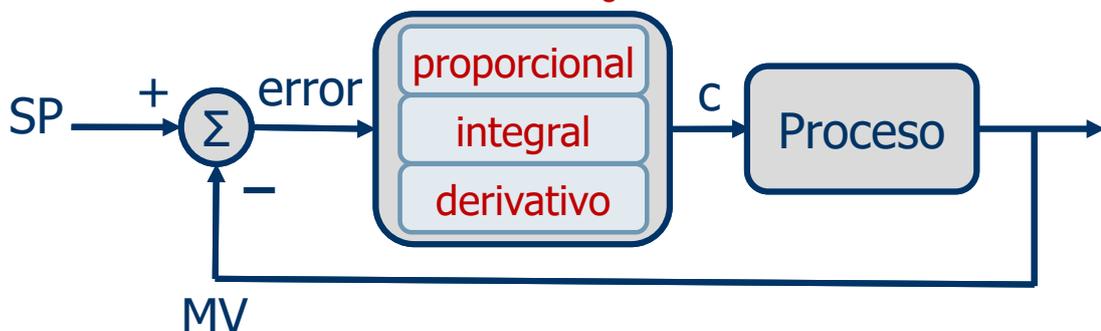


Control clásico



Control PID [proporcional + integral + derivativo]

$$c(t) = k_p \times e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$



Tres ganancias k_p , k_i , k_d

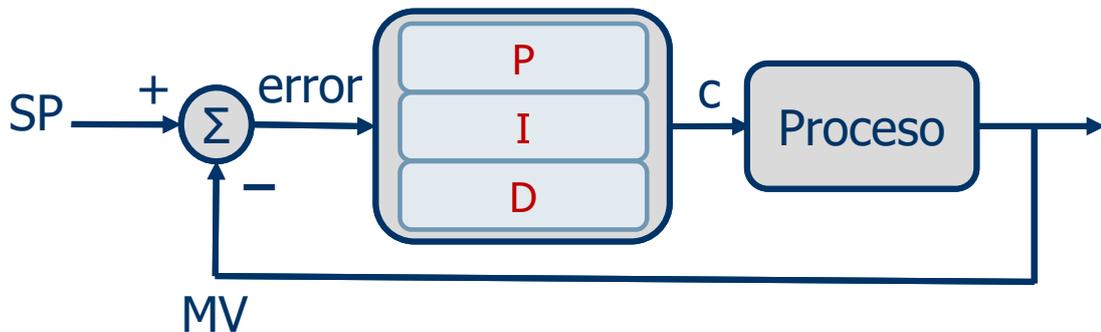


Control clásico



Control PID [proporcional + integral + derivativo]

$$c(t) = k_p \times e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$



Tres ganancias k_p , k_i , k_d

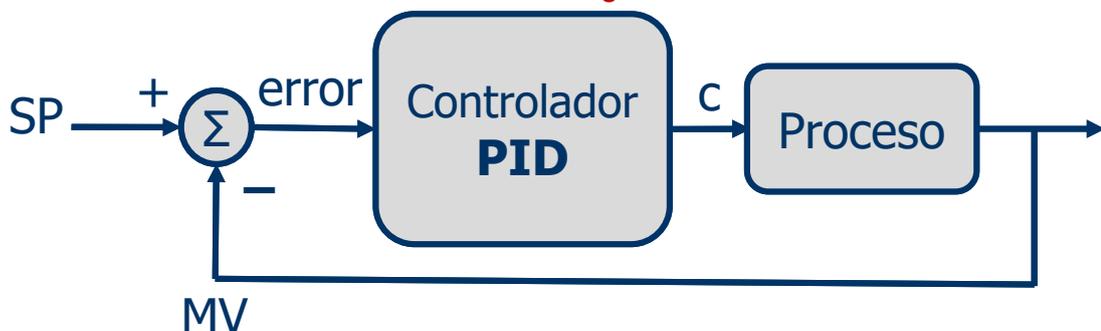


Control clásico



Control PID [proporcional + integral + derivativo]

$$c(t) = k_p \times e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$



Tres ganancias k_p , k_i , k_d





- Aplicación más notable de la lógica difusa.
- Un controlador difuso utiliza “reglas de control” expresadas en términos lingüísticos, tal como las formularía un experto.
- “Versión lingüística de los sistemas de control”:
Sistema de inferencia difusa
(i.e. sistema basado en reglas difusas)
con la particularidad de que sus entradas son señales no difusas [crisp].

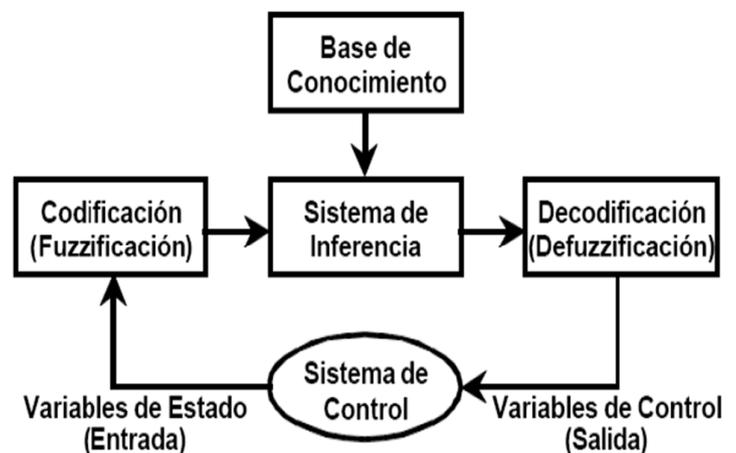


Controlador difuso



Estructura de un controlador difuso

[Lee 1990]

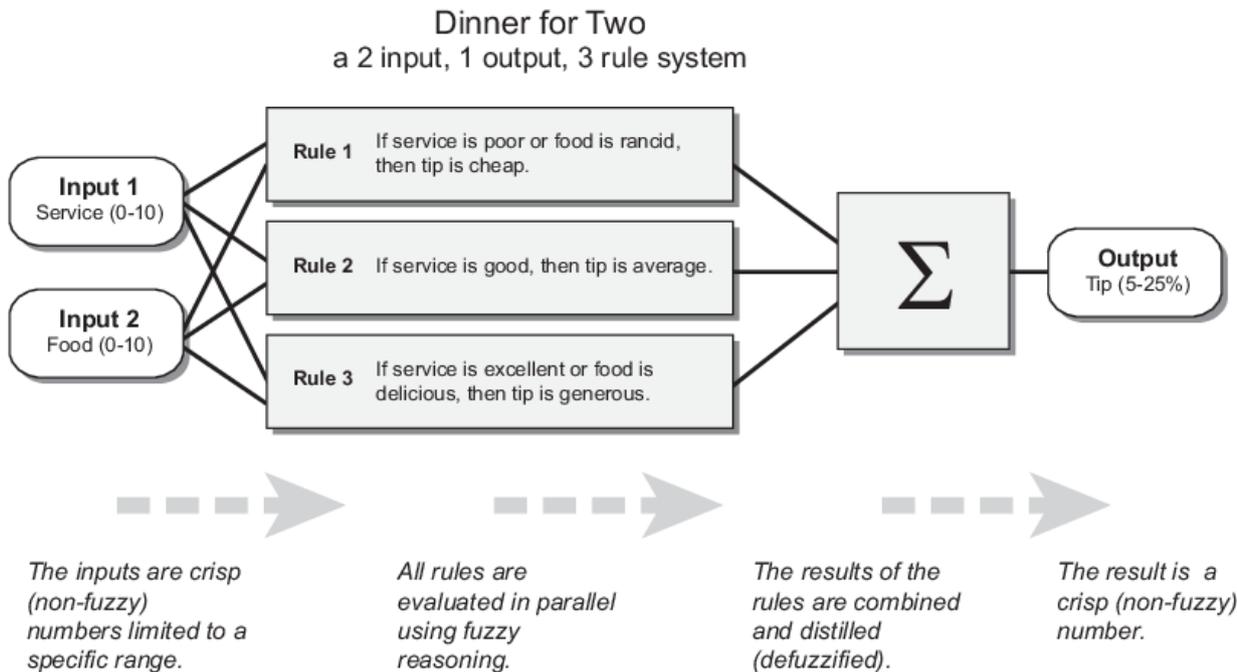


4 componentes:

- Codificador [fuzzifier]
- Base de conocimiento (reglas difusas)
- Sistema de inferencia
- Decodificador [defuzzifier]



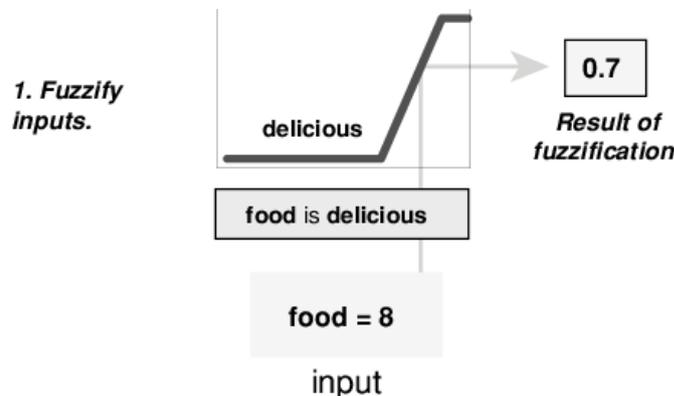
Controlador difuso



Controlador difuso



Codificación o "difuminación" [fuzzification]



Se convierten las variables de entrada del controlador [crisp] en conjuntos difusos, p.ej.

- Singleton (opción más sencilla y utilizada)
- Conjunto difuso triangular centrado en el valor crisp y con soporte acorde a la incertidumbre de la medición.





Base de conocimiento

- Conocimiento asociado al dominio de aplicación y sus objetivos de control.
- Reglas difusas de control, con las variables de estado en el antecedente y las variables de control en el consecuente.
- Especificación en términos de etiquetas lingüísticas.



Base de conocimiento

FUNCIONES

- Almacenar las definiciones necesarias para determinar las reglas lingüísticas de control y la manipulación de los datos difusos del controlador (esto es, reglas difusas y funciones de pertenencia de las etiquetas lingüísticas).
- Almacenar los objetivos y la política de control como un experto humano en el dominio.



Controlador difuso



Sistema de inferencia

- Núcleo del controlador difuso.
- Infiere las acciones de control simulando el proceso de decisión humano (mediante las reglas de inferencia de la lógica difusa).
- Utiliza las técnicas de los sistemas basados en reglas [difusas].

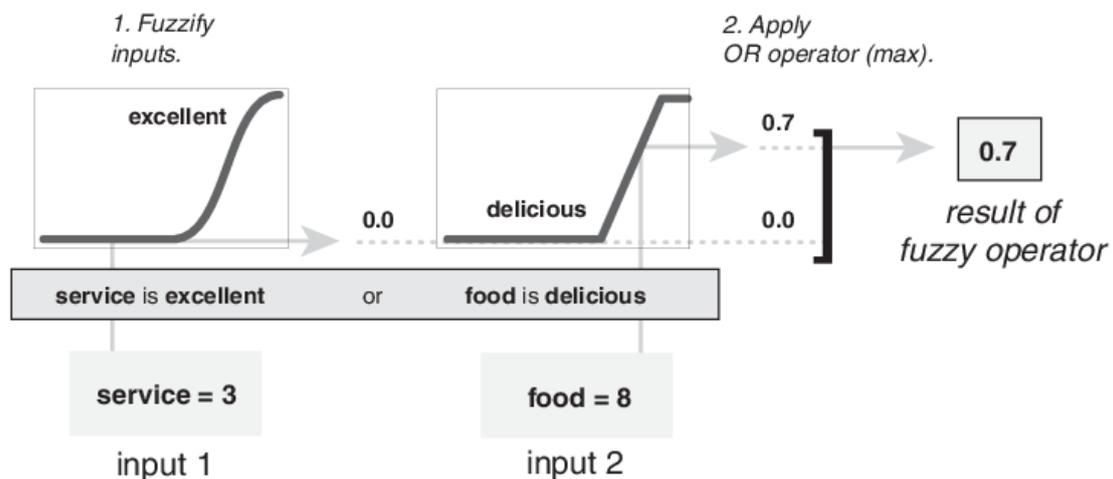


Controlador difuso



Sistema de inferencia

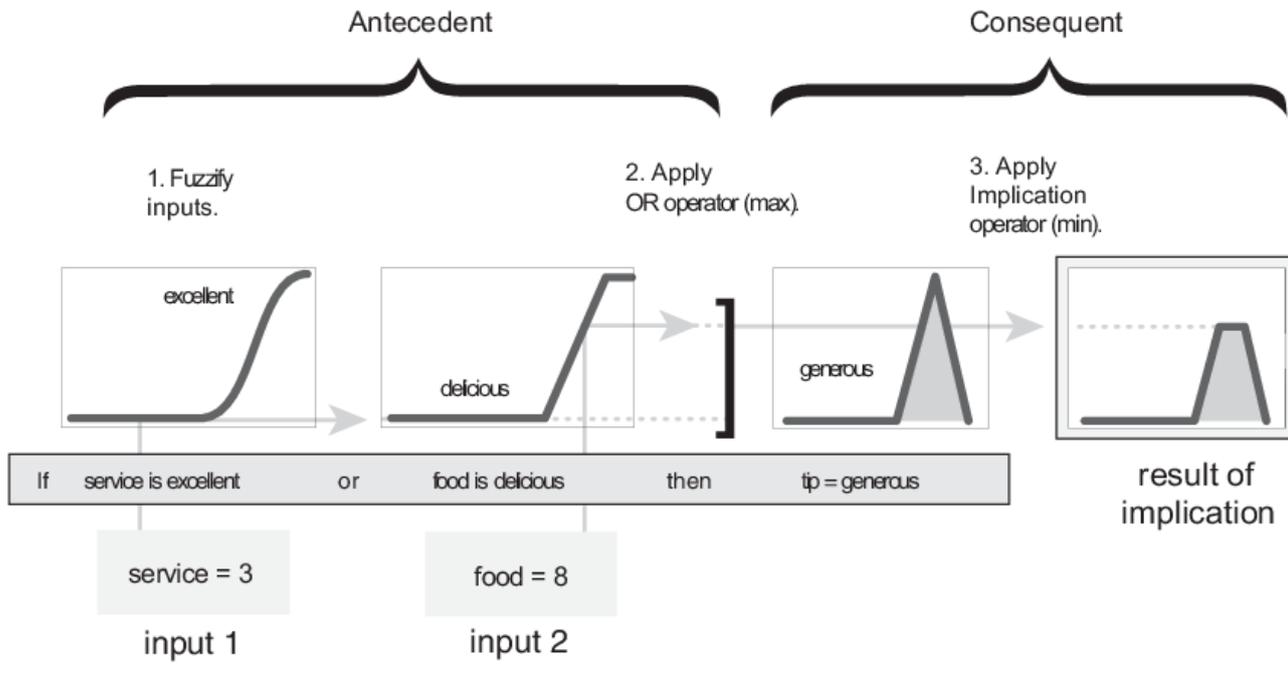
Grado de cumplimiento de los antecedentes



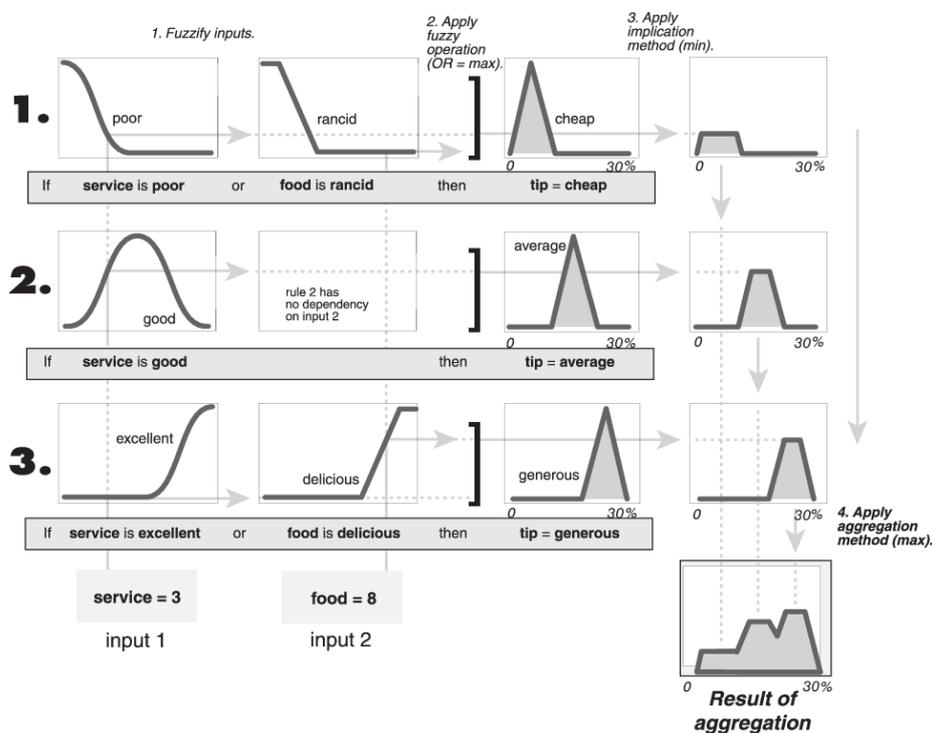


Sistema de inferencia

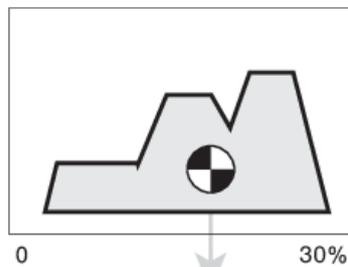
Implicación difusa



Sistema de inferencia: Agregación



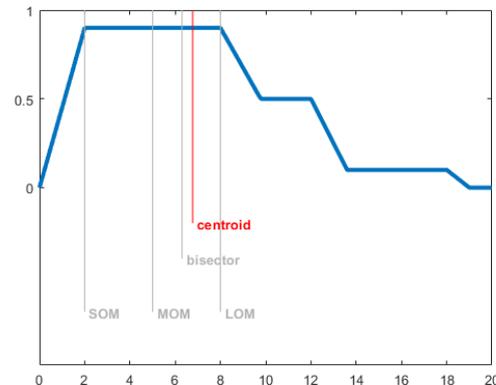
Decodificación o "concisión" [defuzzification]



5. Defuzzify the aggregate output (centroid).

tip = 16.7%

Result of defuzzification



Se convierten los valores difusos de las variables de salida del controlador en valores crisp para generar una señal de control concreta.

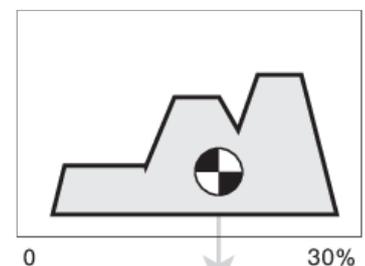


Decodificación o "concisión" [defuzzification]

¿Cuál es el valor crisp que mejor representa a un conjunto difuso?

- Centro de gravedad (centroide del área)
- Bisector del área
- Menor de los máximos [SOM]
- Media de los máximos [MOM]
- Mayor de los máximos [LOM]

...

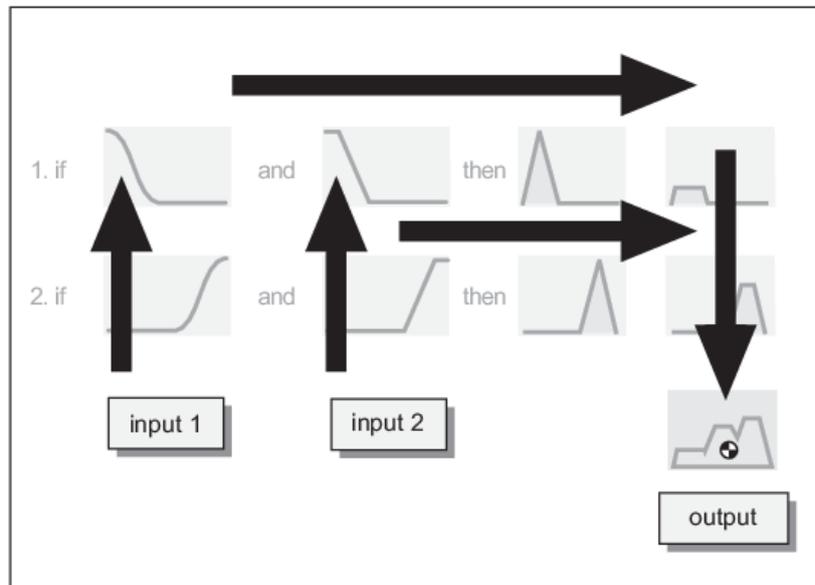


tip = 16.7%

Result of defuzzification



Interpreting the fuzzy inference diagram

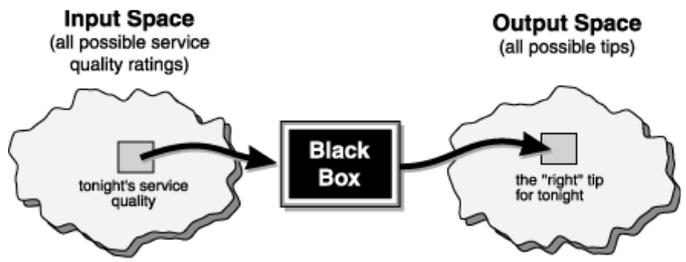


Ventajas de la difuminación [fuzzification]

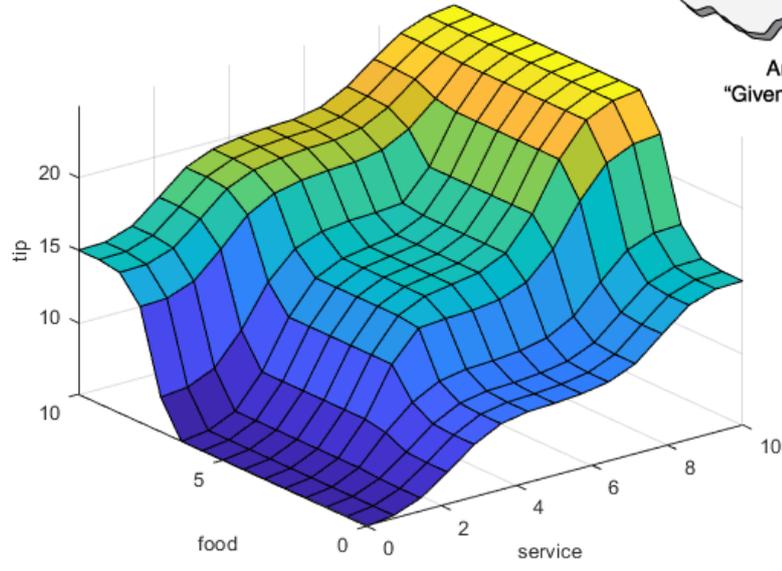
- Minimiza cambios ligeros en las variables de entrada.
- Minimiza posibles errores en la toma de datos.
- Suaviza el comportamiento del sistema.



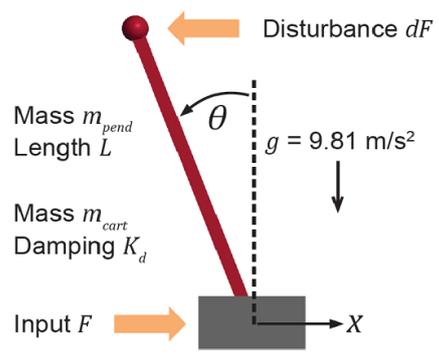
Controlador difuso



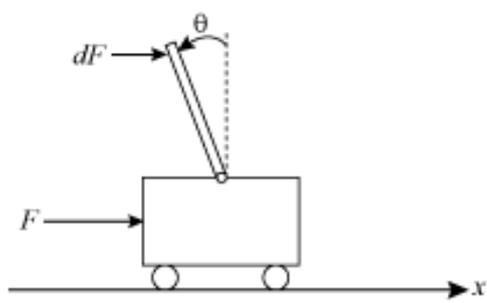
An input/output map for the tipping problem:
 "Given the quality of service, how much should I tip?"



Ejemplo Péndulo invertido



Control clásico



Control difuso

PROBLEMA

Equilibrar una pértiga sobre una plataforma móvil que puede moverse en dos únicas direcciones, a la izquierda o a la derecha.

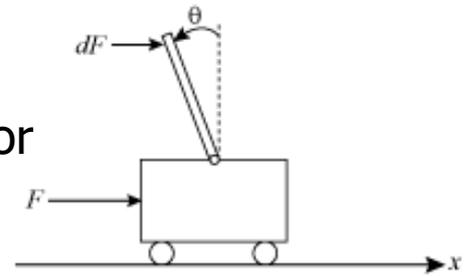


Ejemplo Péndulo invertido



Entradas

- Ángulo: Error
- Velocidad angular: Cambio en el error



Reglas

Si definimos N etiquetas lingüísticas para cada variable de entrada, tendremos hasta N^2 reglas del tipo

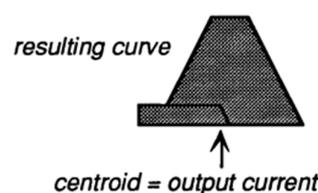
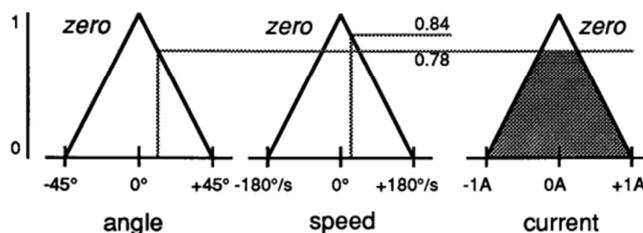
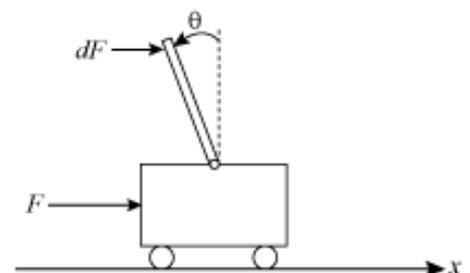
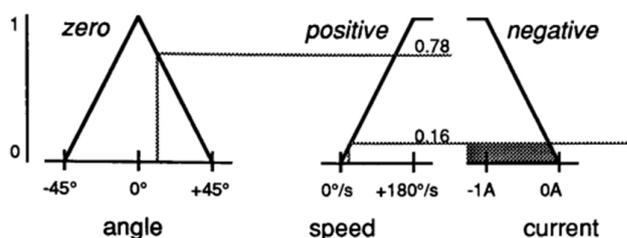
if error is zero and speed is negative
then output should be zero



Ejemplo Péndulo invertido



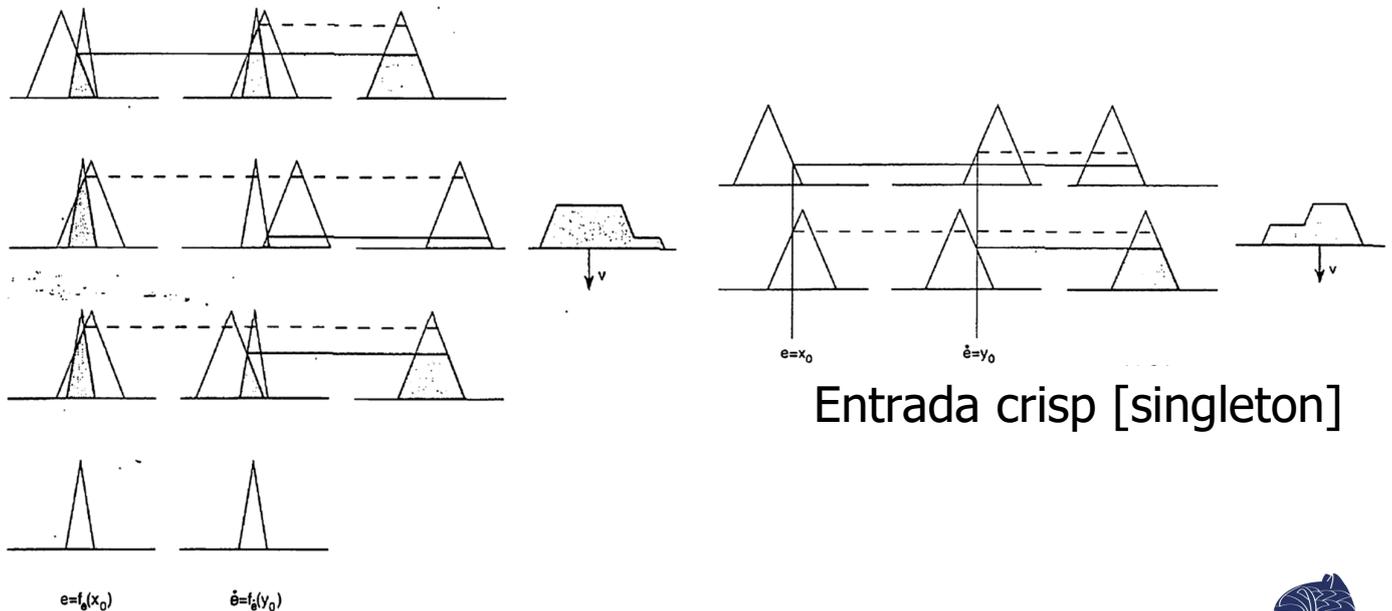
Aplicación de las reglas



Ejemplo Péndulo invertido



Aplicación de las reglas: Medidas de entrada



Entrada difusa [triangular]

Entrada crisp [singleton]

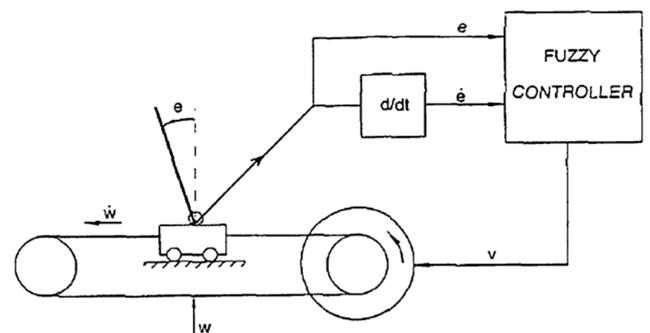


Ejemplo Péndulo invertido



Control clásico

$$\begin{aligned}
 I\ddot{e} &= VL \sin e - HL \cos e, \\
 V - mg &= -mL(\ddot{e} \sin e + \dot{e}^2 \cos e), \\
 H &= m\ddot{w} + mL(\ddot{e} \cos e - \dot{e}^2 \sin e), \\
 U - H &= M\ddot{w},
 \end{aligned}$$



Control difuso

Conjunto mínimo de 7 reglas lingüísticas

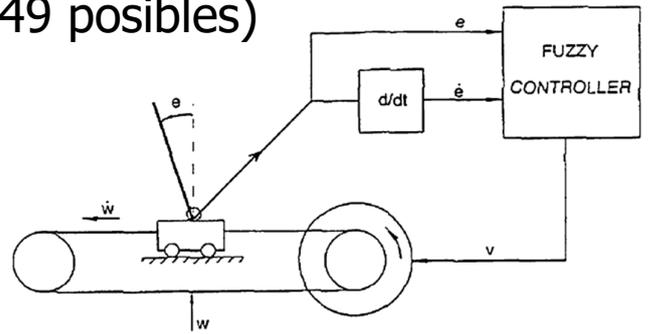
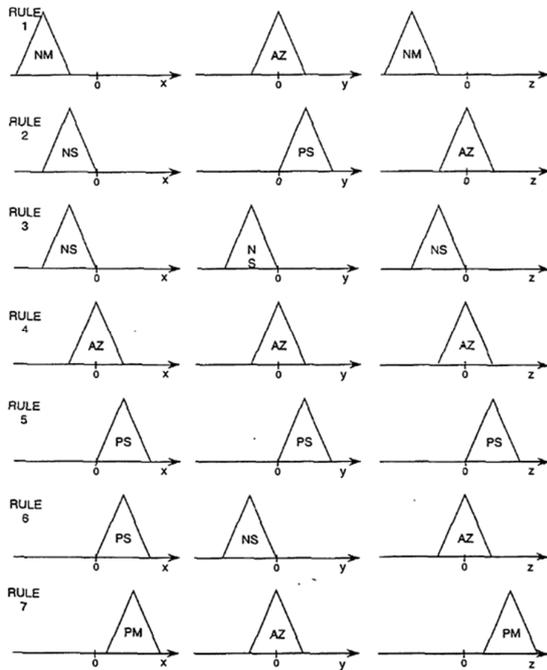
$e \backslash \dot{e}$	NM	NS	AZ	PS	PM
NS		NS		AZ	
AZ	NM		AZ		PM
PS		AZ		PS	



Ejemplo Péndulo invertido



Control difuso: 7 reglas (de 49 posibles)



Fuzzy Sets and Systems
Volume 32, Issue 2, 11 September 1989, Pages 161-180



Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system

Takeshi Yamakawa *

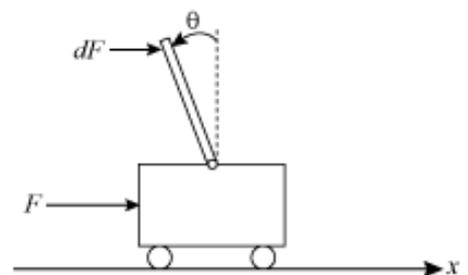


Ejemplo Péndulo invertido

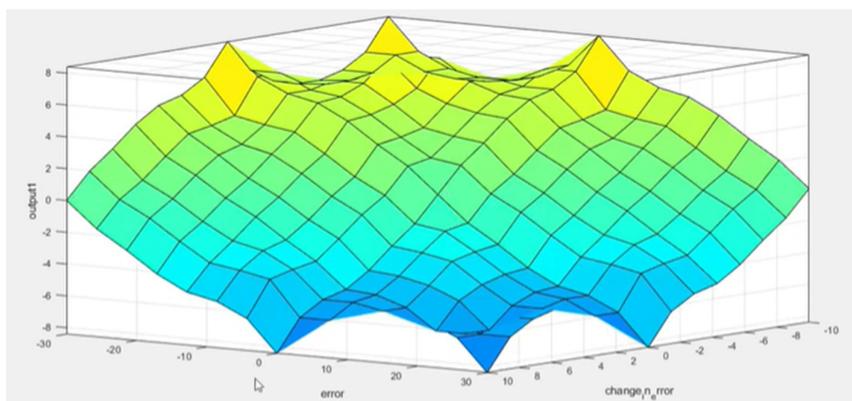


Entradas

- Error
- Cambio en el error



Salida



Ejemplo Péndulo invertido



APLICACIÓN "PRÁCTICA": Paliar el efecto de fuertes vientos o terremotos en un edificio.

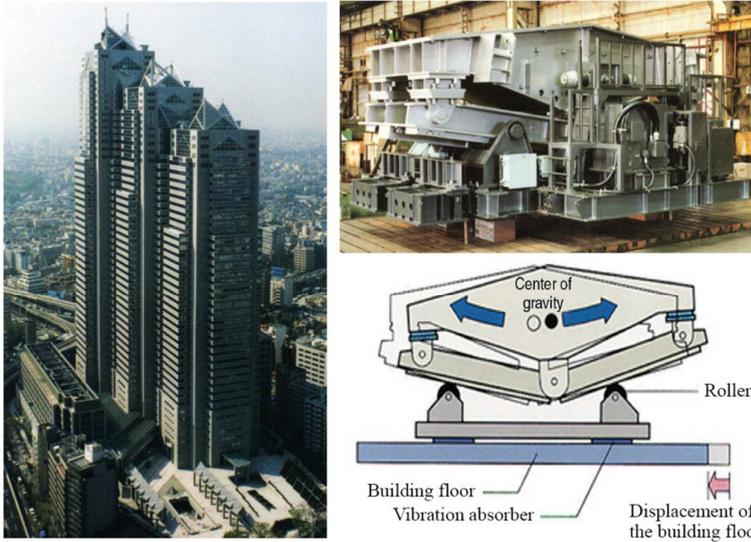


Figure 1.4: Shinsuku Park Tower (1993) installed with V-shaped HMDs.

Objetivo: Amortiguar las vibraciones de la estructura.

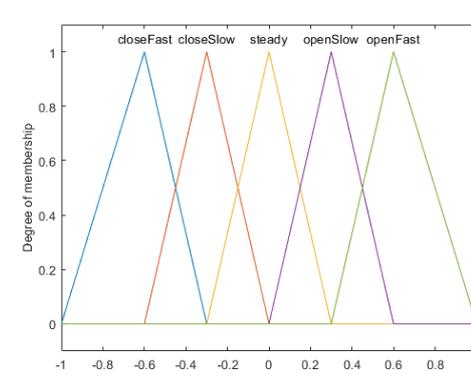
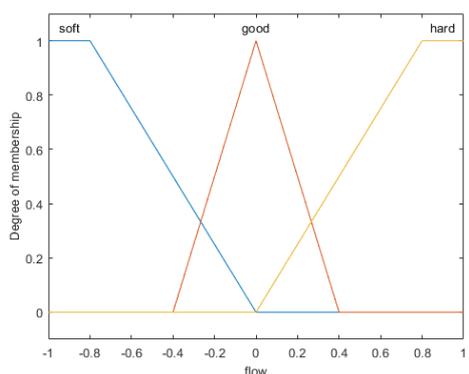
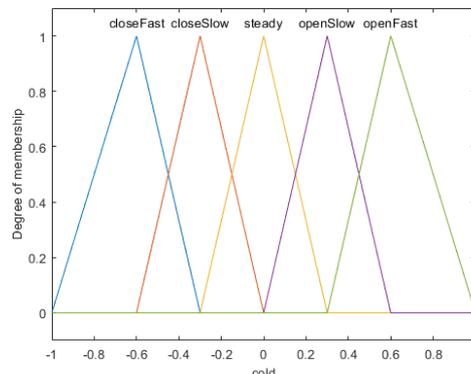
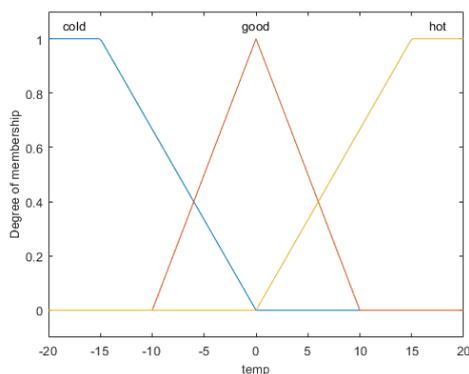


Ejemplo Control de temperatura



Entradas: T & flujo

Salidas: Válvulas





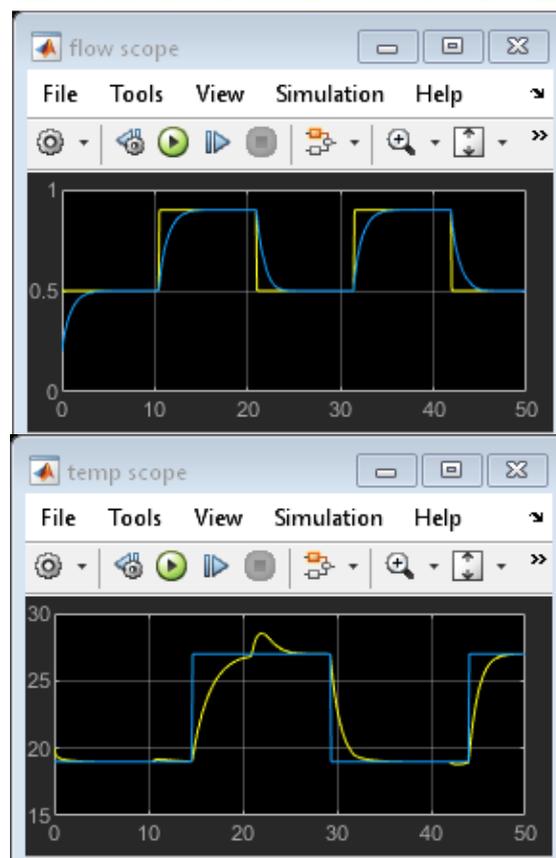
Reglas del sistema de control

Sintaxis de MATLAB

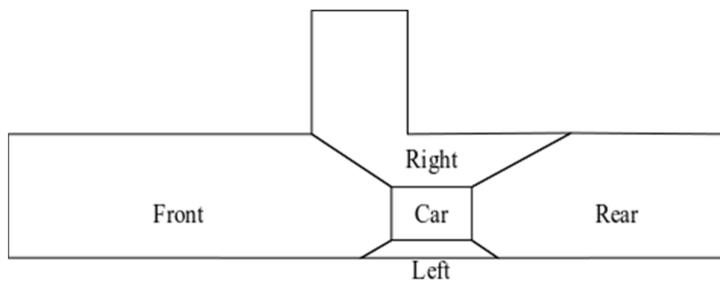
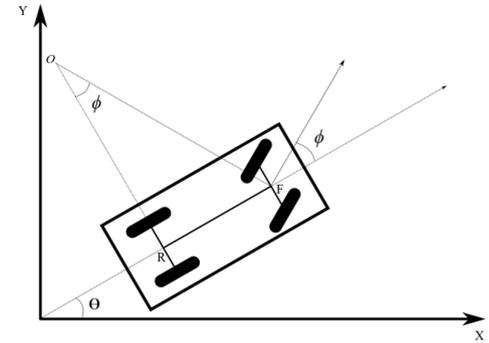
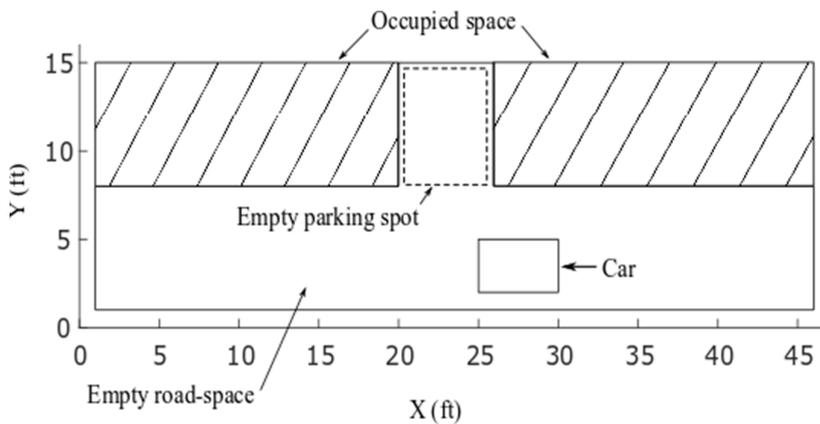
```
temp==cold & flow==soft => cold=openSlow, hot=openFast
temp==cold & flow==good => cold=closeSlow, hot=openSlow
temp==cold & flow==hard => cold=closeFast, hot=closeSlow
temp==good & flow==soft => cold=openSlow, hot=openSlow
temp==good & flow==good => cold=steady, hot=steady
temp==good & flow==hard => cold=closeSlow, hot=closeSlow
temp==hot & flow==soft => cold=openFast, hot=openSlow
temp==hot & flow==good => cold=openSlow, hot=closeSlow
temp==hot & flow==hard => cold=closeSlow, hot=closeFast
```



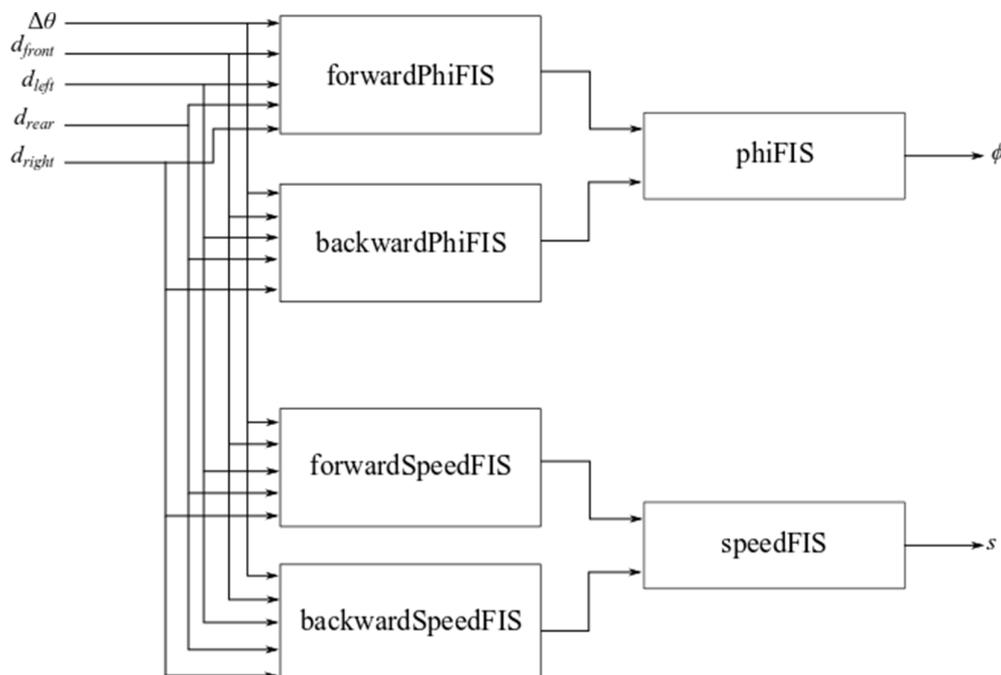
Simulación del sistema MATLAB



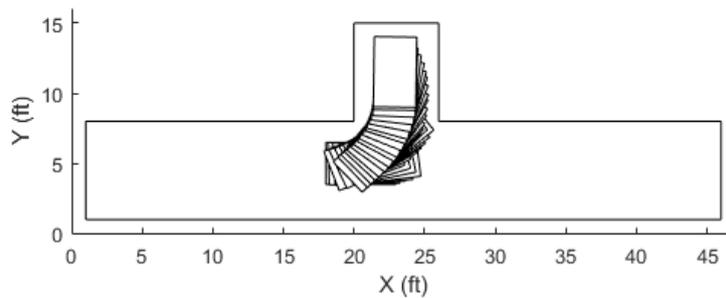
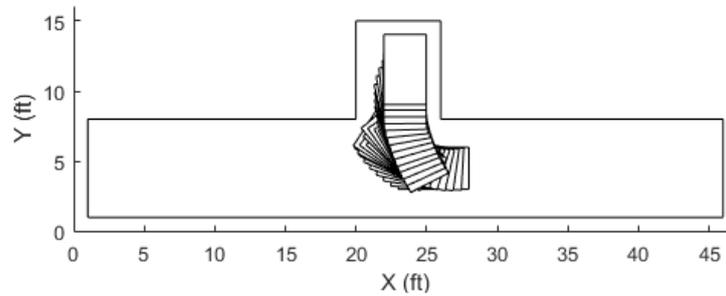
Ejemplo Aparcamiento automático



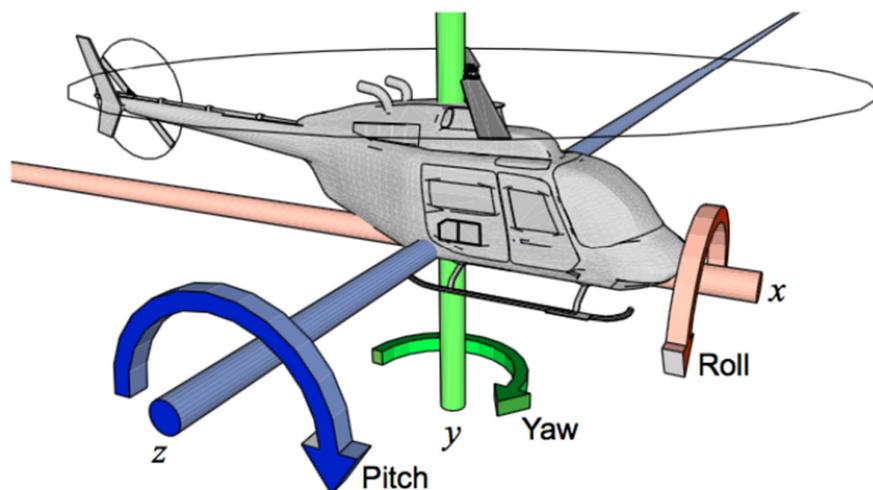
Ejemplo Aparcamiento automático



Ejemplo Aparcamiento automático



Ejemplo Helicóptero

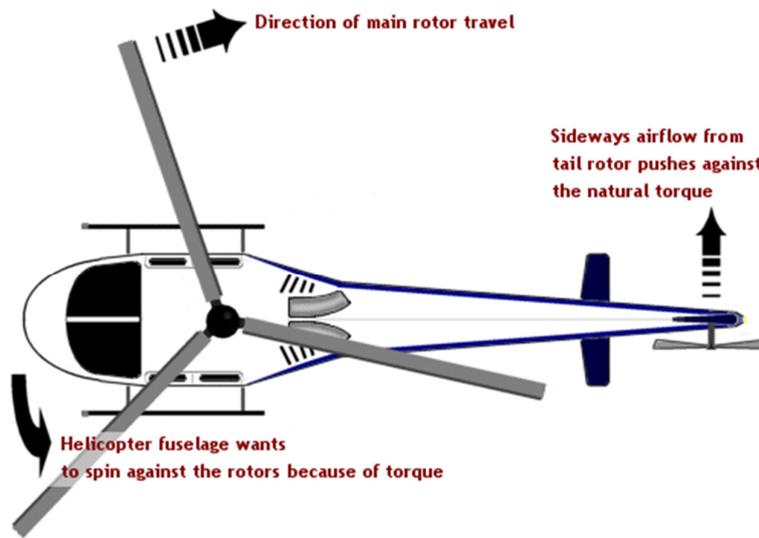


Seis grados de libertad para describir su posición:

- Coordenadas: x, y, z
- Ángulo: pitch, yaw, roll



Ejemplo Helicóptero



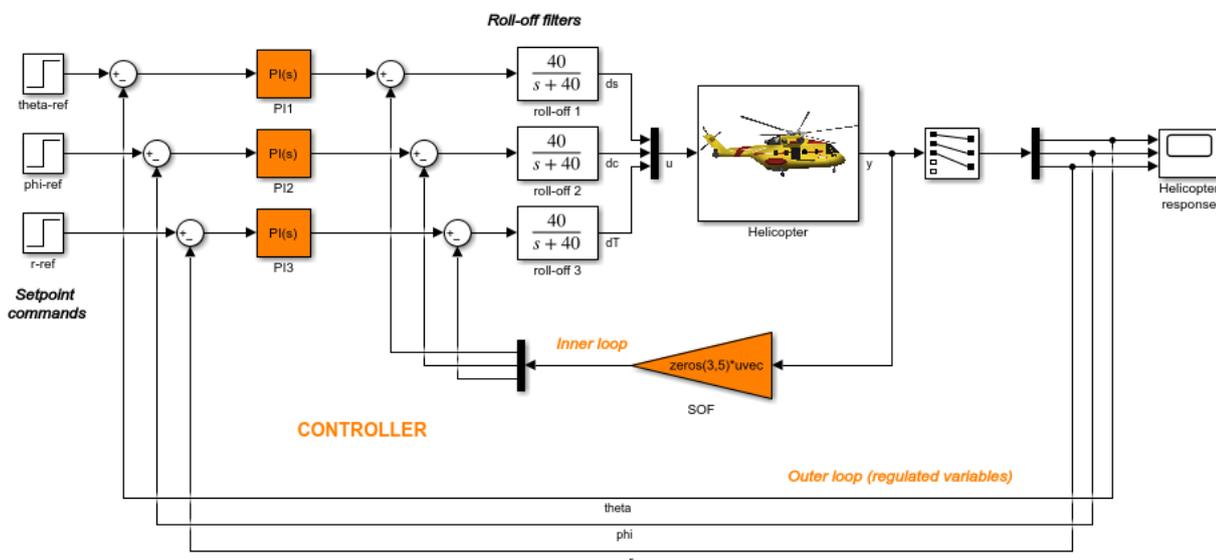
Dinámica compleja



Ejemplo Helicóptero



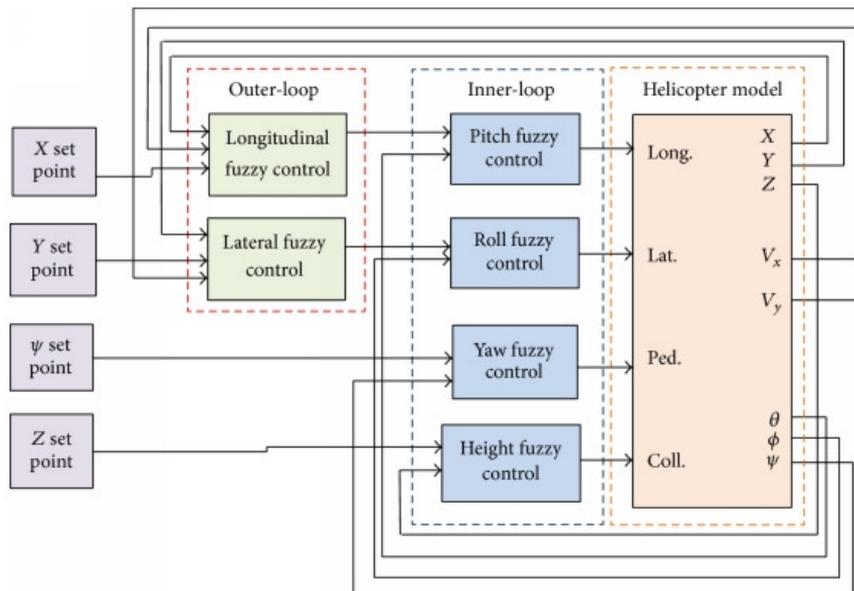
Controlador clásico (sólo hovering)



Ejemplo Helicóptero



Controlador difuso



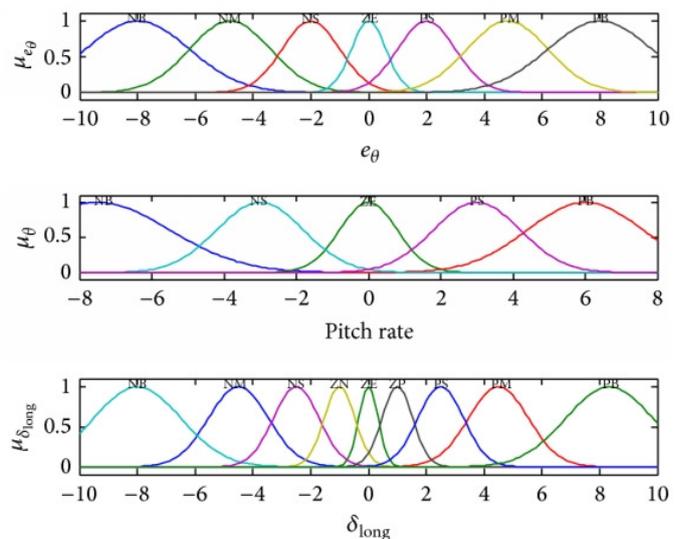
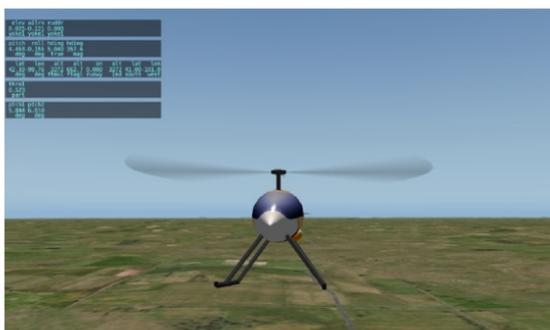
Como en los ejemplos anteriores, pero con más variables... 8



Ejemplo Helicóptero



Controlador difuso



Como en los ejemplos anteriores, pero con más variables...





Primer gran éxito de los controladores difusos

Sendai Subway 1000N series (仙台市交通局1000N系電車)

Sendai Subway Namboku Line @ Sendai, Japan.



- Desarrollado por Hitachi.
- Primer tren que utilizó un controlador difuso para controlar su velocidad: mayor suavidad al arrancar y al parar, 10% más eficiente que un conductor humano.



Anuncio de 1997

... hace lo más difícil:
ponértelo fácil.

Porque gracias al sistema
Fuzzy Logic, toma sus
propias decisiones.

Blanco y Negro, ABC
2 de noviembre de 1997
página 85

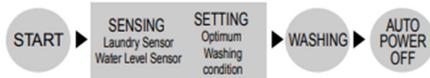


Ejemplo Lavadoras



FUZZY logic control

- A built-in load sensor automatically detects the laundry load and a microprocessor optimizes washing conditions such as ideal WATER LEVEL and washing time.



LG
Life's Good
**LAVADORA LG
FUZZY LOGIC
TURBODRUM 8.0KG
T7503TDFP1**



Ejemplo Lavadoras



NUEVO

WD14VVC456C

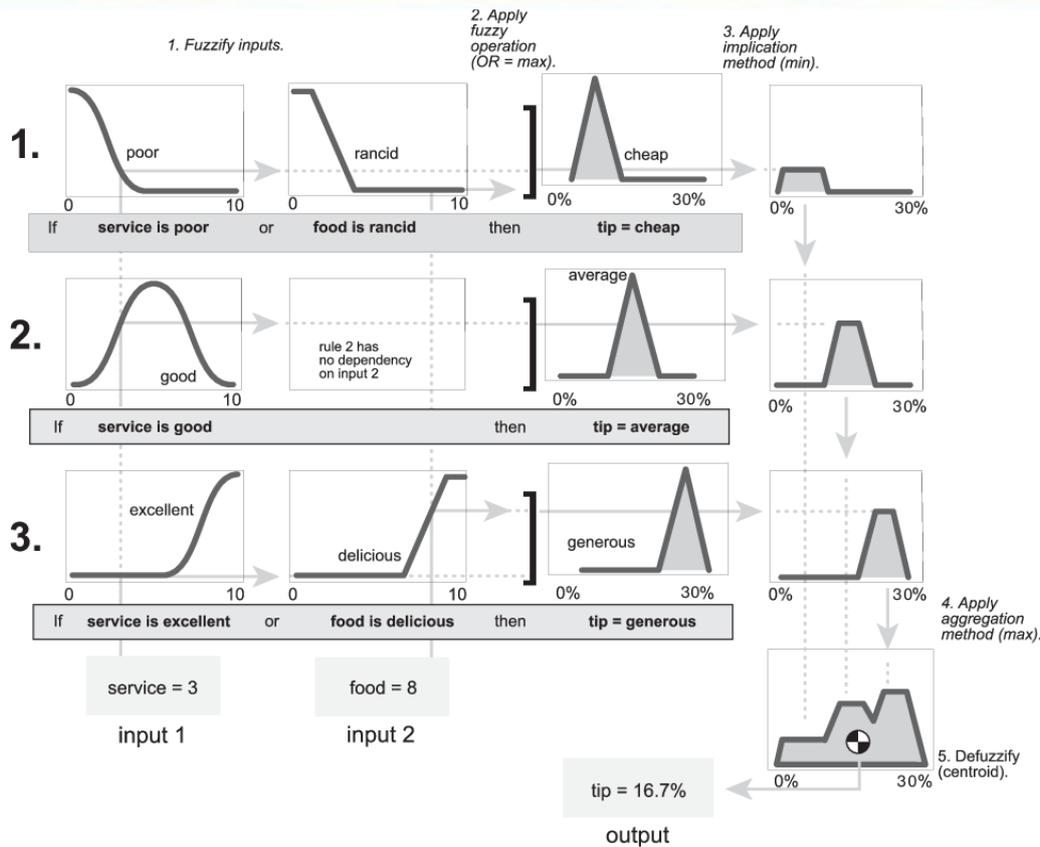
Lavadora-Secadora Carga Frontal con Inteligencia Artificial y capacidad total 14 kilos de lavado y secado

★★★★★ (0) | [Escriba una reseña](#)

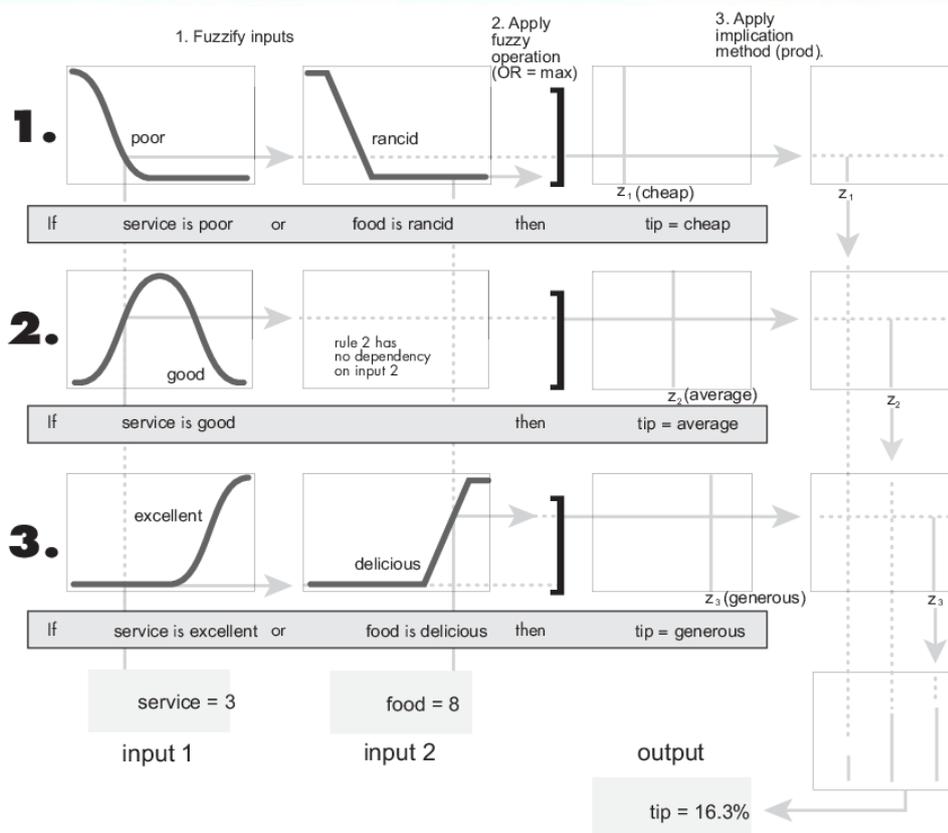
- AI DD™ / Inteligencia Artificial3
- Motor Inverter Direct Drive con 10 años de Garantía
- Tecnología Steam
- Smart Diagnosis y ThinQ - Wifi



Controlador tipo Mamdani



Controlador tipo Sugeno



Controlador tipo Sugeno



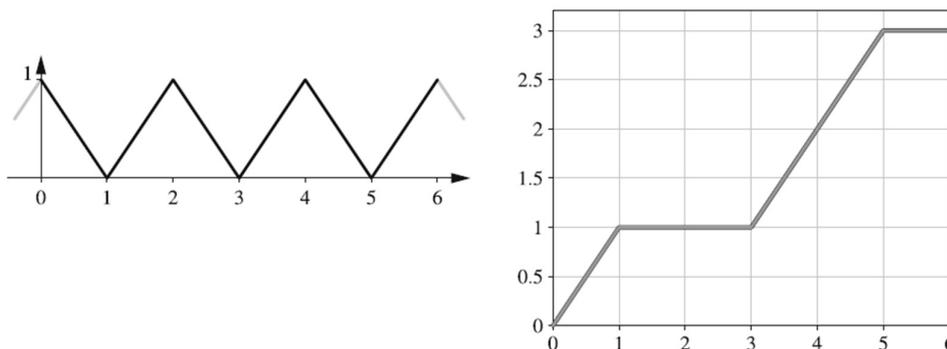
En los controladores de tipo Takagi-Sugeno-Kang (TSK), las reglas son de la forma

**if x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n
then output = $f(x_1, x_2, \dots, x_n)$**

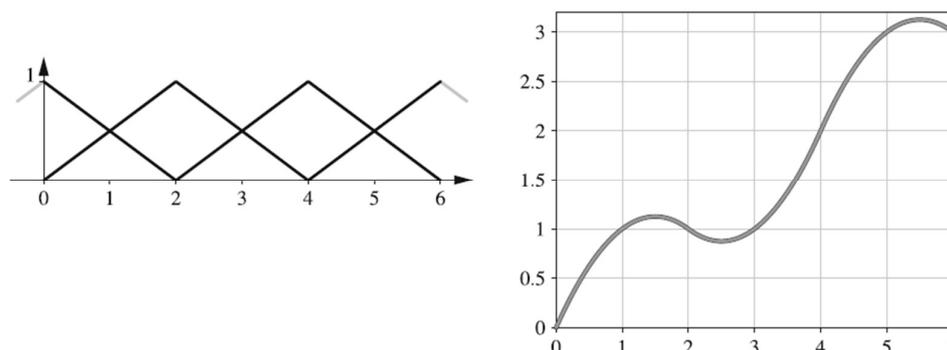
IDEA: La función de salida es una buena función de control local para el conjunto difuso descrito por el antecedente de la regla, p.ej. singleton o modelo lineal.



Controlador tipo Sugeno



Conjuntos difusos no solapados (modelos locales)



Conjuntos difusos solapados (mezcla de modelos)



Control neurodifuso



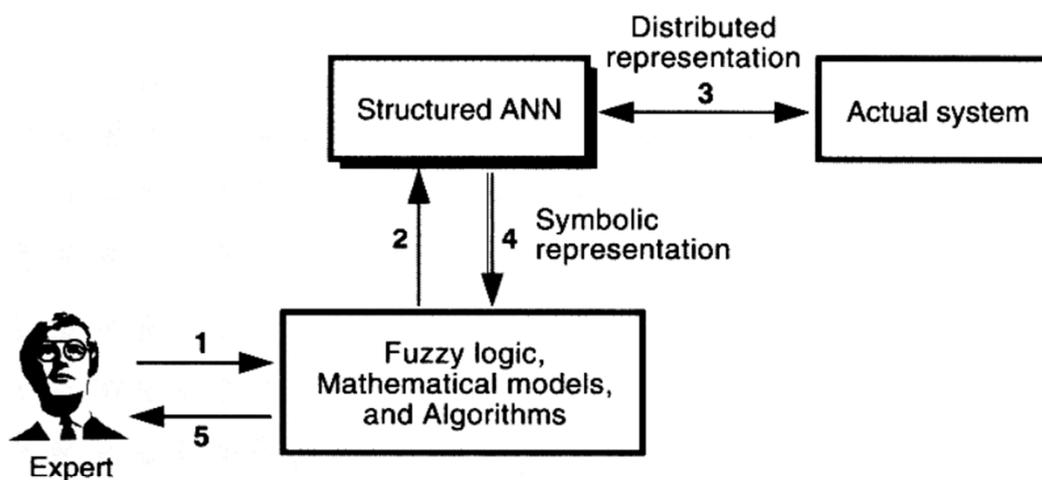
Se pueden combinar las ventajas de un controlador difuso (interpretabilidad lingüística) con la capacidad de aprendizaje de otras técnicas de Inteligencia Computacional (p.ej. redes neuronales).

A partir de un conjunto de datos, podemos aprender las reglas difusas para nuestro controlador...

- Los conjuntos difusos de los antecedentes.
- Los parámetros de la función de salida (modelo TSK).



Control neurodifuso



1. Translate expert's knowledge into a symbolic representation
2. Initialize ANN by the symbolic representation
3. Decrease errors between actual system and ANN by learning
4. Translate the distributed representation into a symbolic representation based upon the ANN structure
5. Acquire knowledge from the modified symbolic representation





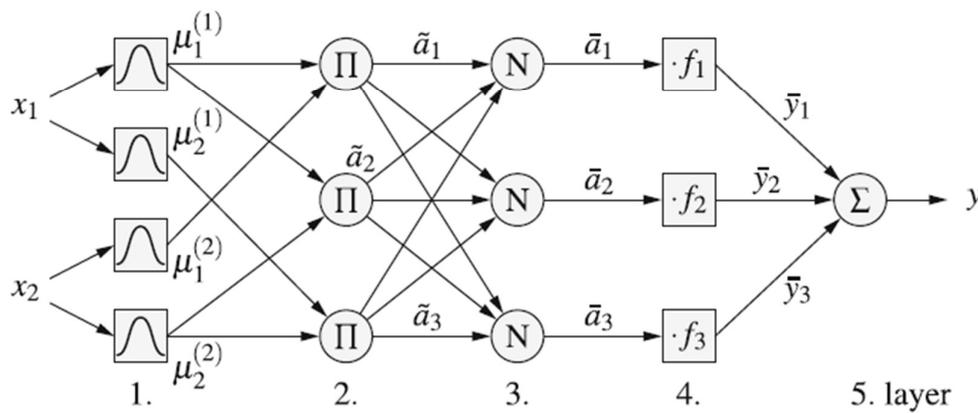
ANFIS

[Adaptive Network-based Fuzzy Inference System]

R_1 : If x_1 is A_1 and x_2 is B_1 then $y = f_1(x_1, x_2)$

R_2 : If x_1 is A_1 and x_2 is B_2 then $y = f_2(x_1, x_2)$

R_3 : If x_1 is A_2 and x_2 is B_2 then $y = f_3(x_1, x_2)$

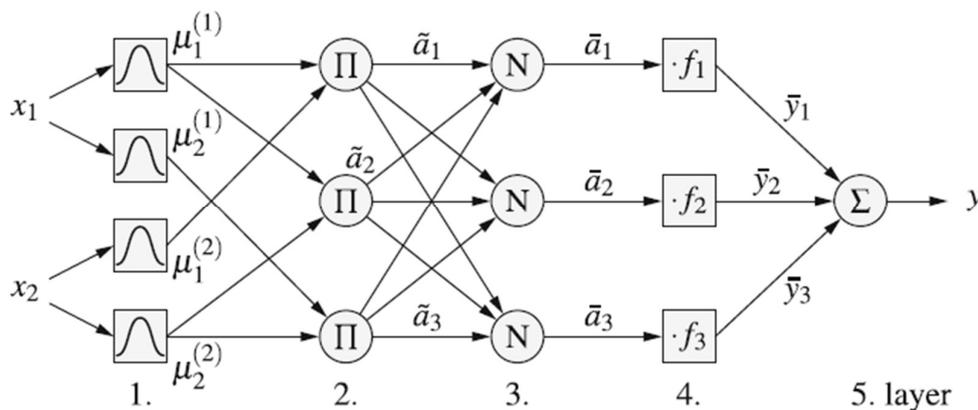


ANFIS

[Adaptive Network-based Fuzzy Inference System]

- Salida lineal

$$f_i = p_i x_1 + q_i x_2 + r_i.$$

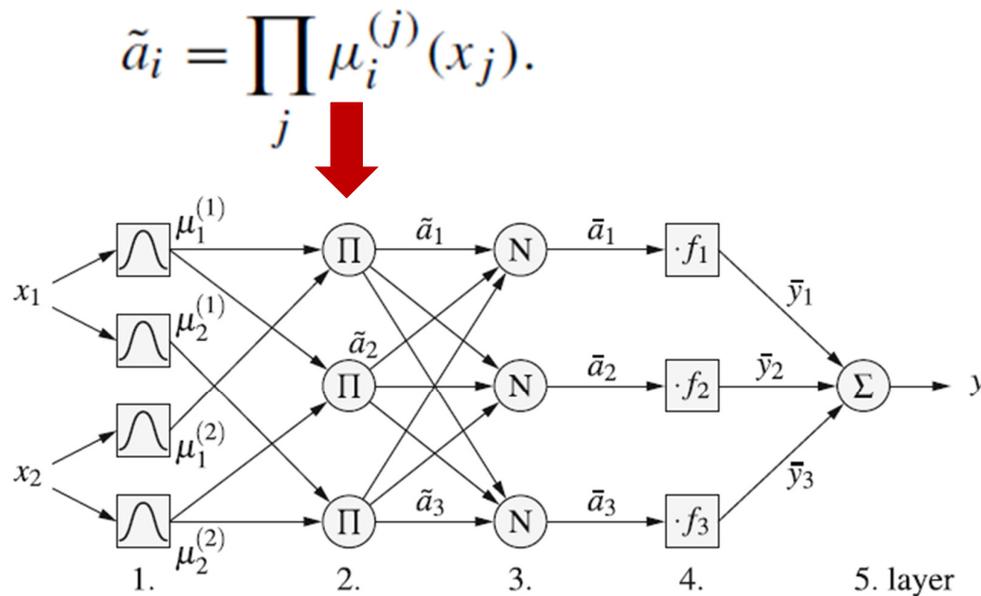




ANFIS

[Adaptive Network-based Fuzzy Inference System]

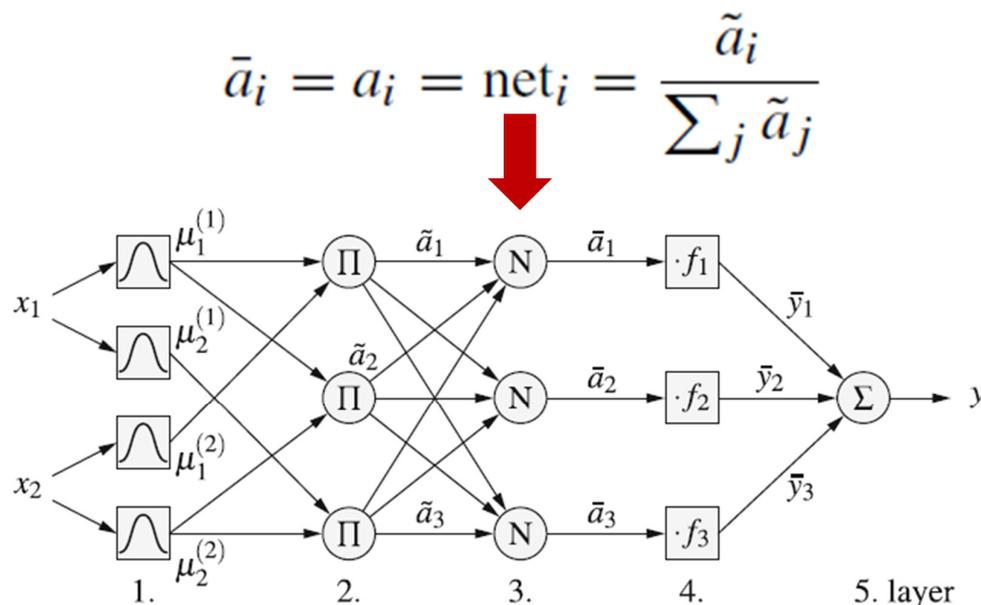
- Evaluación de los antecedentes (t-norma del producto)



ANFIS

[Adaptive Network-based Fuzzy Inference System]

- Contribución de cada regla a la salida

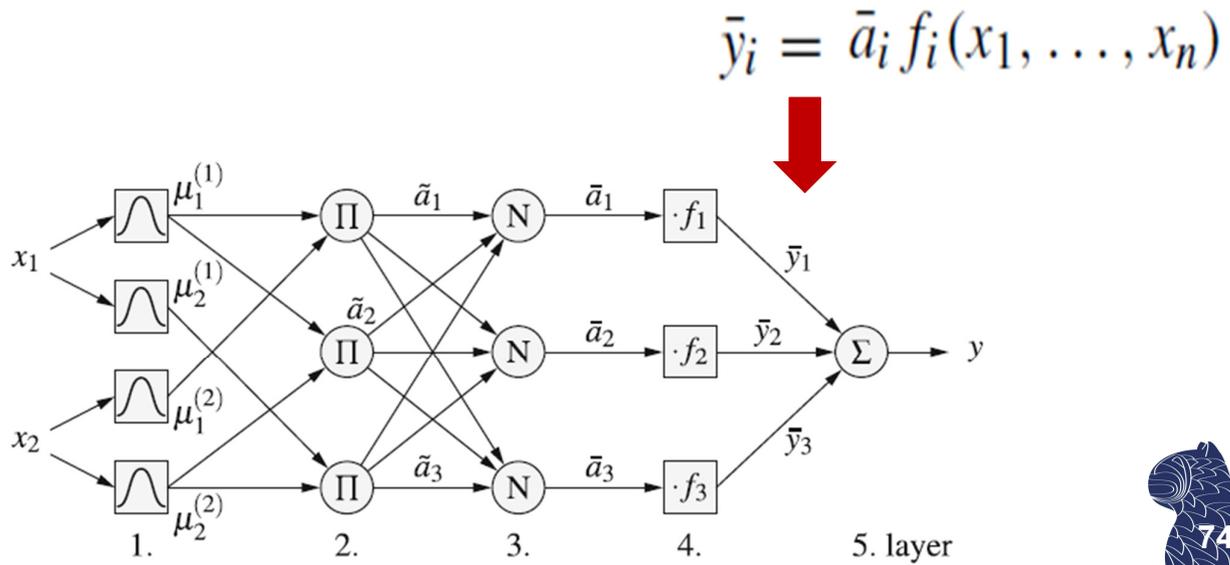




ANFIS

[Adaptive Network-based Fuzzy Inference System]

- Salidas de control ponderadas

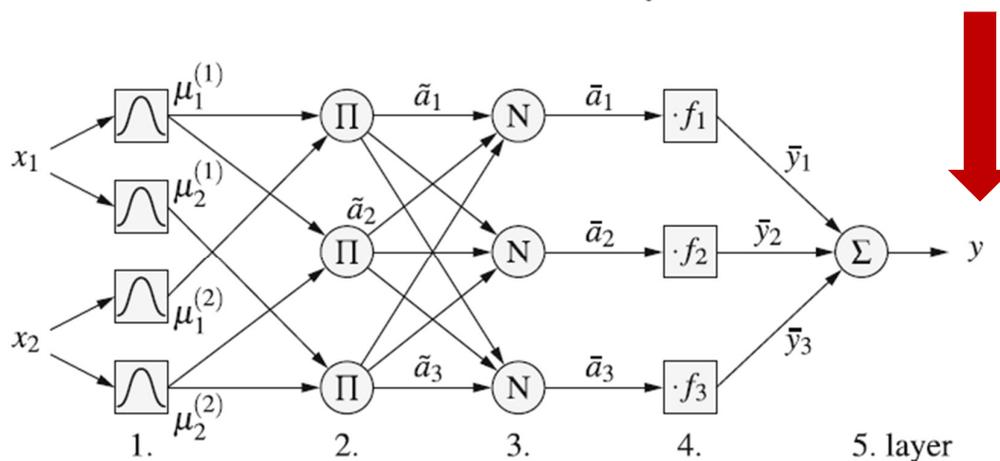


ANFIS

[Adaptive Network-based Fuzzy Inference System]

- Salida final del controlador

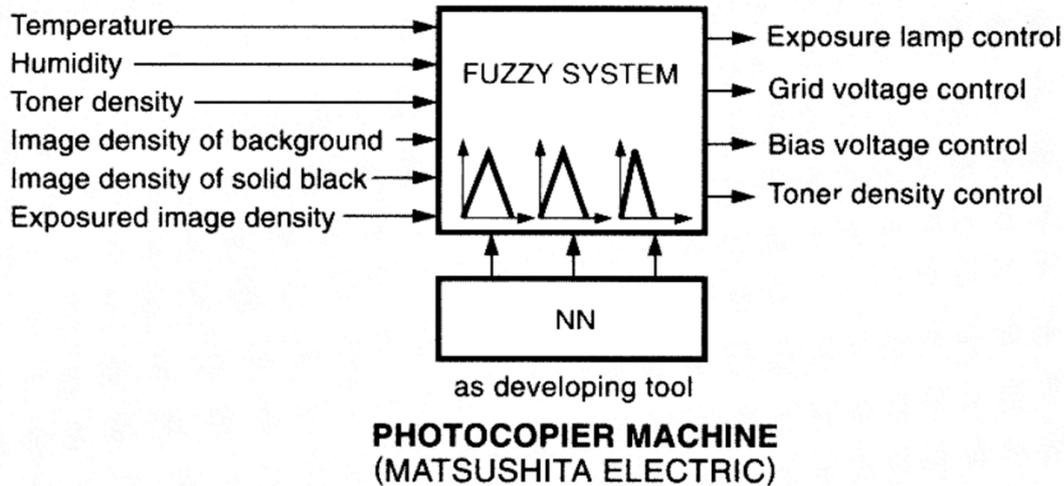
$$y = a_{\text{out}} = \text{net}_{\text{out}} = \sum_i \bar{y}_i = \frac{\sum_i \bar{a}_i f_i(x_1, \dots, x_n)}{\sum_i \bar{a}_i}$$



Control neurodifuso



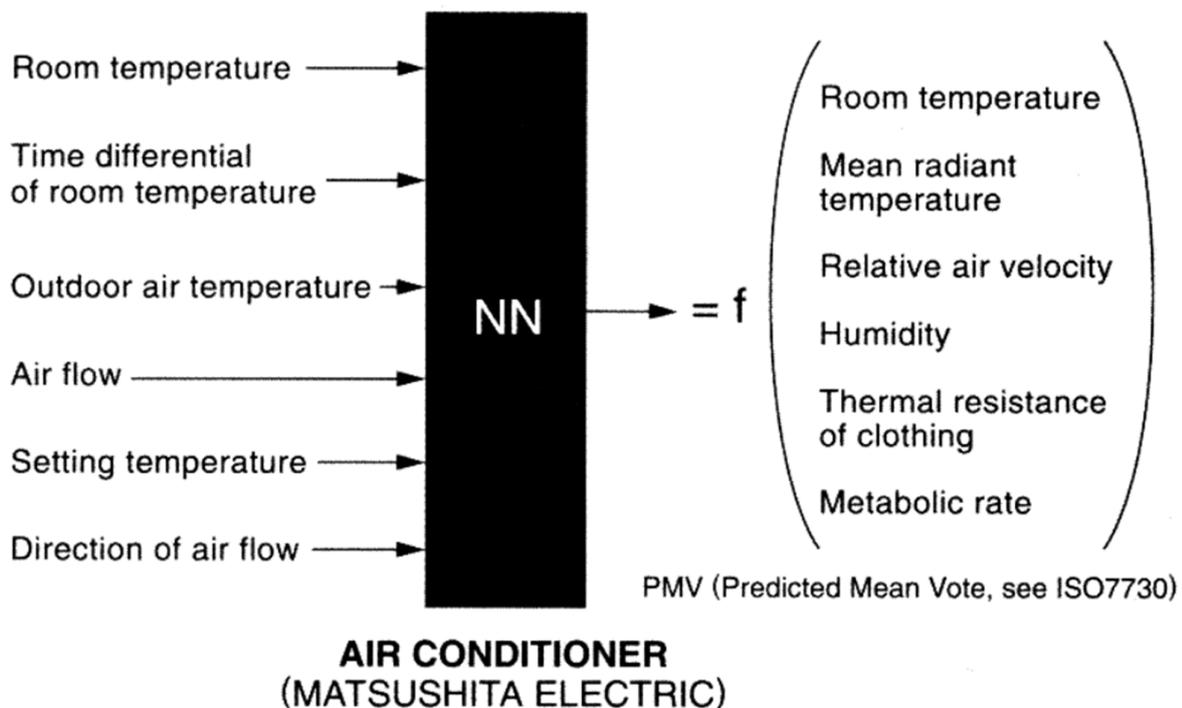
Aplicaciones industriales en Japón (años 90)...



Control neurodifuso



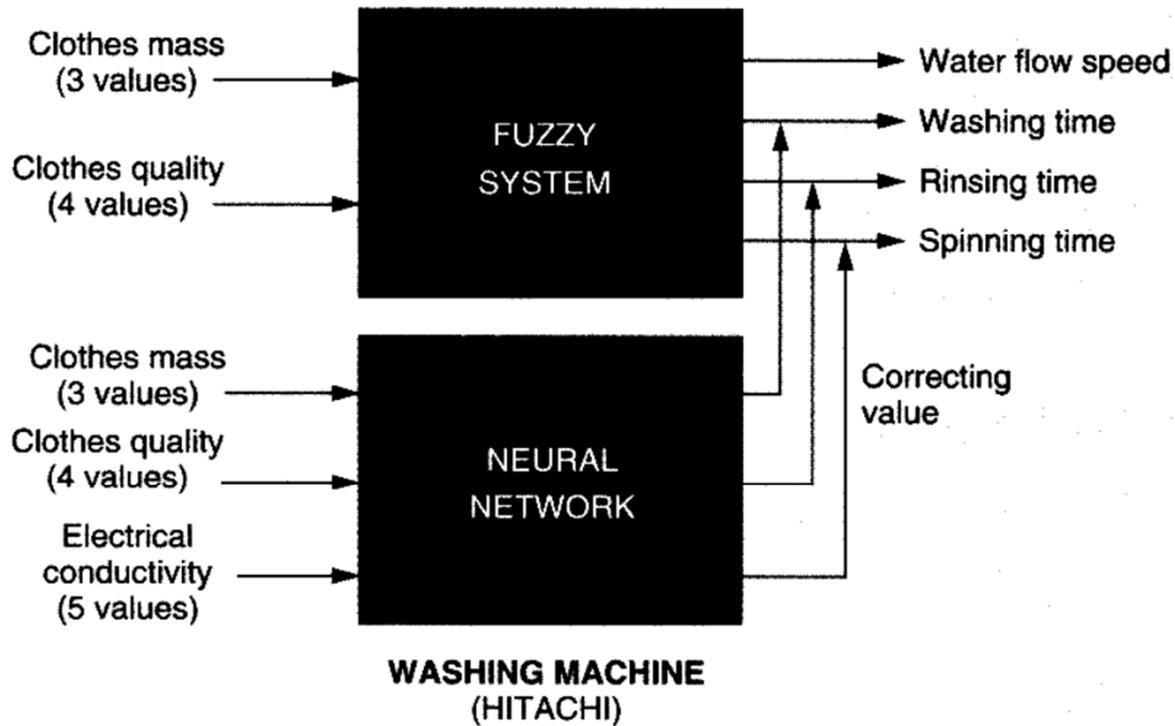
Aplicaciones industriales en Japón (años 90)...



Control neurodifuso



Aplicaciones industriales en Japón (años 90)...

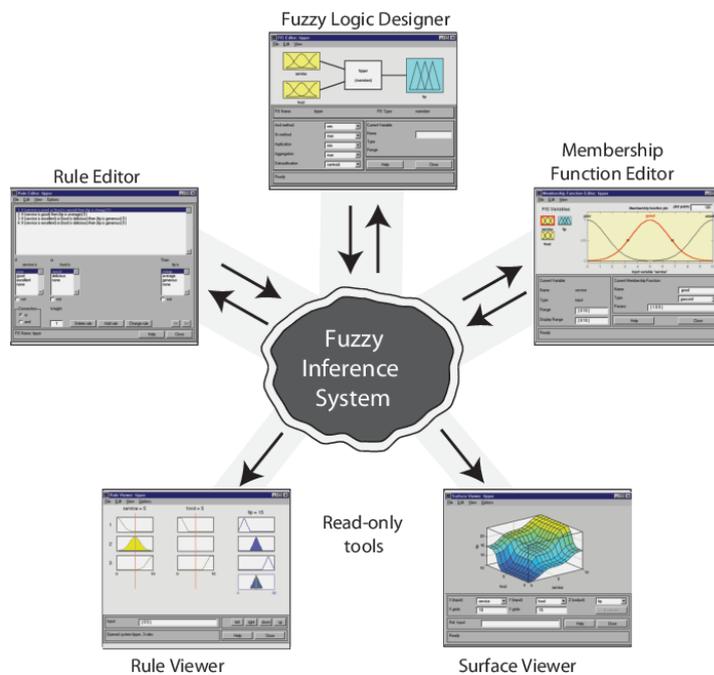
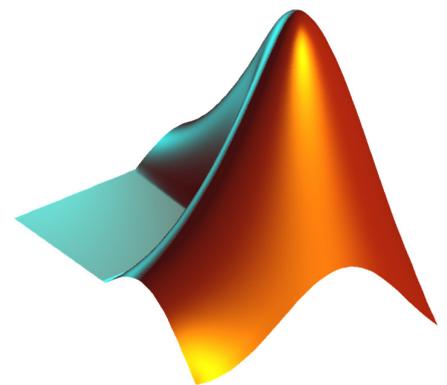


Herramientas



MATLAB Fuzzy Logic Toolbox

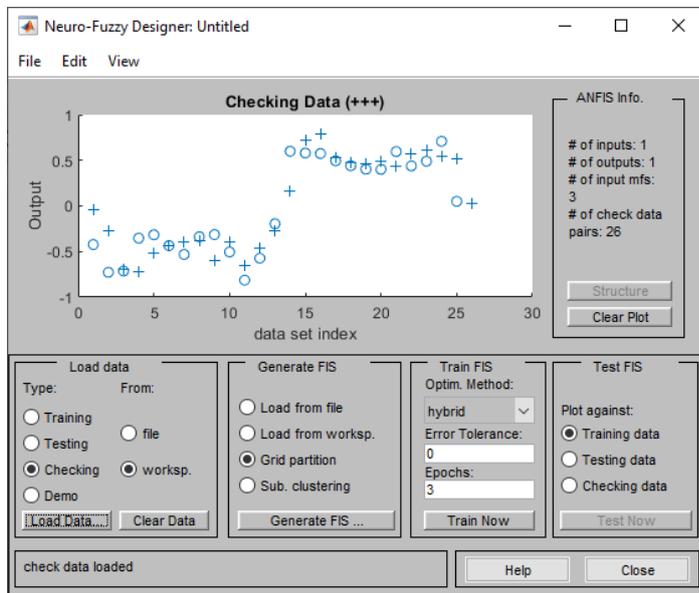
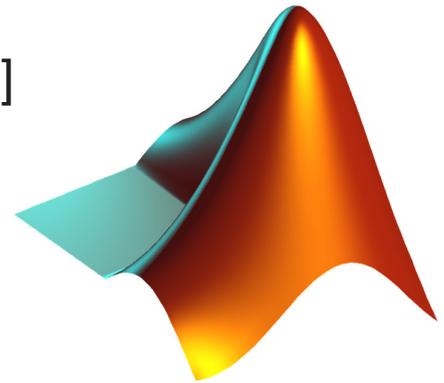
<https://es.mathworks.com/help/fuzzy/>





ANFIS

[adaptive neuro-fuzzy inference system]



Bibliografía



Miguel Delgado:

Apuntes de Inteligencia Computacional

Universidad de Granada, hasta el curso 2021/2022

Sesiones grabadas en vídeo, curso 2020/2021:

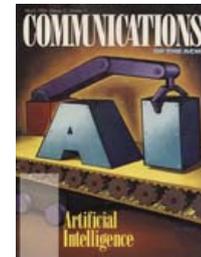
<https://elvex.ugr.es/decsai/computational-intelligence/video/fuzzy/>



Bibliografía



- Brad Schofield:
"Introduction to PID Control",
CERN, 2018
<https://indico.cern.ch/event/777969/>
- Kazuo Asakawa & Hideyuki Takagi:
"Neural Networks in Japan",
Communications of the ACM, March 1994
<https://doi.org/10.1145/175247.175258>
- Michio Sugeno:
"An Introductory Survey of Fuzzy Control,"
Information Sciences, 36(1-2):59-83, 1985
[https://doi.org/10.1016/0020-0255\(85\)90026-X](https://doi.org/10.1016/0020-0255(85)90026-X)



Bibliografía clásica



Concepto de controlador difuso (1975)

- E.H. Mamdani & S. Assilian: "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, 7(1):1-13, 1975

Primera aplicación industrial (1982)

- Lauritz P. Holmblad & Jens-Jørgen Østergaard: "Control of a Cement Kiln by Fuzzy Logic", Fuzzy Information and Decision Processes, 389-399, North-Holland, 1982

Primer gran éxito: Metro de Sendai (Hitachi, 1987)

- S. Yasunobu, S. Miyamoto & H. Ihara: "Fuzzy Control for Automatic Train Operation System," IFAC/IFIP/IFORS International Conference on Control in Transportation Systems, IFAC Proceedings 16(4):33-39, 1983.

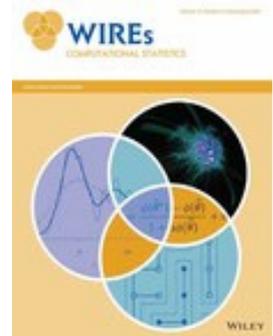


Bibliografía recomendada

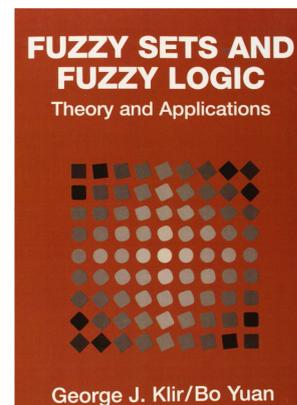


Lógica Difusa

- Hans-Jürgen Zimmermann:
Fuzzy Set Theory,
WIREs Computational Statistics,
John Wiley & Sons, 2:3, May/June 2010.
DOI 10.1002/wics.82



- George J. Klir & Bo Yuan:
**Fuzzy Sets and Fuzzy Logic:
Theory and Applications**,
1st edition, Prentice Hall, 1995.
ISBN 0131011715

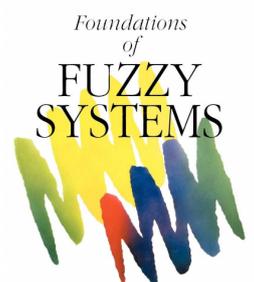


Bibliografía complementaria



Lógica y Sistemas Difusos

- Rudolf Kruse, Joan E. Gebhardt & Frank Klawonn:
Foundations of Fuzzy Systems.
John Wiley & Sons, 1994. ISBN 047194243X.
- Witold Pedrycz & Fernando Gomide:
An introduction to Fuzzy Sets: Analysis and Design.
MIT Press, 1998. ISBN 0262161710.
- Hans-Jürgen Zimmermann:
Fuzzy Set Theory and Its Applications,
Springer, 3rd edition, 1996. ISBN 0792396243
Springer, 4th edition, 2001. ISBN 9401038708.
- F. Martin McNeill & Ellen Thro:
Fuzzy Logic: A Practical Approach.
Morgan Kaufmann, 1994. ISBN 0124859658.



R. Kruse • J. Gebhardt • F. Klawonn

FUZZY LOGIC
A PRACTICAL APPROACH
F. MARTIN McNEILL • ELLEN THRO
Foreword by Ronald R. Yager

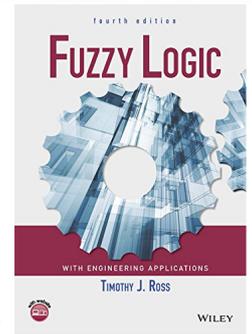


Bibliografía complementaria



Lógica y Sistemas Difusos

- Timothy J. Ross:
Fuzzy Logic with Engineering Applications,
4th edition, John Wiley & Sons, 2017. ISBN 1119235863.
- Lofti A. Zadeh: **Fuzzy Sets**.
Information and Control, volume 8, issue 3, pp. 338-353,
June 1965. DOI 10.1016/S0019-9958(65)90241-X
- James C. Bezdek: **Pattern Recognition with Fuzzy Objective
Function Algorithms**. Plenum Press, 1981. ISBN 0306406713.
- Bart Kosko: **Neural Networks and Fuzzy Systems: A
Dynamical Systems Approach to Machine Intelligence**.
Prentice Hall, 1992. ISBN 0136114350
- Mohammad Jamshidi, Nader Vadiee & Timothy Ross (editors):
**Fuzzy Logic and Control. Software and Hardware
Applications**. Prentice Hall, 1993. ISBN 0133342514.

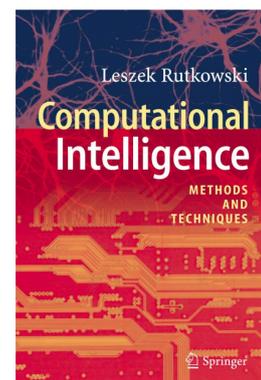
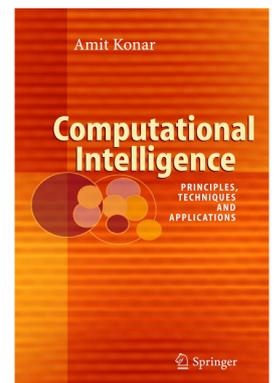
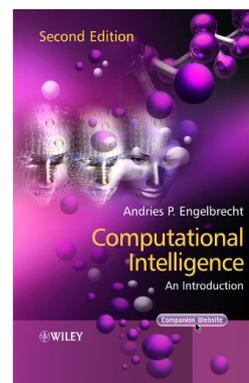


Bibliografía complementaria



Inteligencia Computacional

- Andries P. Engelbrecht:
**Computational Intelligence.
An Introduction**,
2nd edition, John Wiley, 2007.
ISBN 0470035617.
- Amit Konar:
**Computational Intelligence.
Principles, Techniques and Applications**,
Springer Verlag, 2005.
ISBN 3540208984.
- Leszek Rutkowski:
**Computational Intelligence.
Methods and Techniques**,
Springer Verlag, 2008.
ISBN 3540762876.





Inteligencia Computacional

- James M. Keller, Derong Liu & David B. Fogel:
Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation, Wiley - IEEE Press, 2016. ISBN 1119214343
- Rudolf Kruse, Christian Borgelt, Christian Braune, Sanaz Mostaghim, Matthias Steinbrecher, Frank Klawonn & Christian Moewes: **Computational Intelligence: A Methodological Introduction**. Springer, 2nd edition, 2016. ISBN 1447172949

