Fundamentos en Robótica

Unidad 3.2

Teoría de Control

mina
network management | artificial intelligence

Temario

- Introducción al control
- Control lineal y PID
- Control no lineal

Composición de Controladores

Ejemplo: auto autónomo

- Acelerador y frenos
  - Mantener velocidad, distancia... (seleccionar cambios?)
- Volante
  - Dirección asistida
  - Mantenerse en la senda
- Los anteriores, controlados por *Mantenerse en el tráfico*
  - Reaccionar a otros autos, peatones
  - Respetar señales
  - Dirigirse al destino
  - ...

Control no lineal

Sistemas difíciles de controlar con lo visto:

- Dominios no continuos
- Sistemas con estados internos
- Dinámicas complejas, no modelables analíticamente
  - Sistemas descritos **no matemáticamente**
  - Sólo tenemos **ejemplos** de cómo reacciona
  - Sólo tenemos **métricas externas**

Control adaptativo

Ajustar sistema de control, p.ej PID, dinámicamente

- Selección de ganancia (*gain scheduling*)
  - Dividimos el rango de operación, y calculamos parámetros para cada subrango.
- Aprendizaje de parámetros
  - Vamos ajustando parámetros en tiempo de ejecución
  - Problema de aprendizaje, optimización (parámetros controlados por otro sistema de control)

Aprendizaje automático

Un sistema con AA mejora su rendimiento a medida que pasa el tiempo (experiencia)

- Área de investigación muy vasta
- Puede ser considerado un requisito para la inteligencia
- Se pueden aplicar a problemas de control
  - Como técnica en si
  - Para optimizar parámetros de otras técnicas

Aprendizaje automático

¿Donde se aplica en robótica?

- Conocimiento difícil de programar
  ej: manipulaciones complejas
- Información desconocida
  ej: orientarse en entornos sin mapear
- Ambientes cambiantes
  ej: adaptarse a fallos de componentes

Aprendizaje automático

- Exploración
  Recorrer el espacio de soluciones para ver si encuentro una mejor solución
- Explotación
  Usar la mejor solución disponible para mejorar el rendimiento del sistema

Aprendizaje automático

Algunas técnicas populares:

- Aprendizaje por refuerzo, Q-Learning
  - Encontrar estrategia que maximice premios
  - Se entrena con ejemplos o con un evaluador humano
- Redes Neuronales
  - Inspiración biológica, aproxima funciones arbitrarias
  - Se entrena con ejemplos
- Algoritmos Evolutivos
  - Inspiración biológica, explora espacio de soluciones
  - Problema de optimización de una métrica

Redes Neuronales

Perceptrón, modelo de neurona (McCulloch y Pitts, 1943)

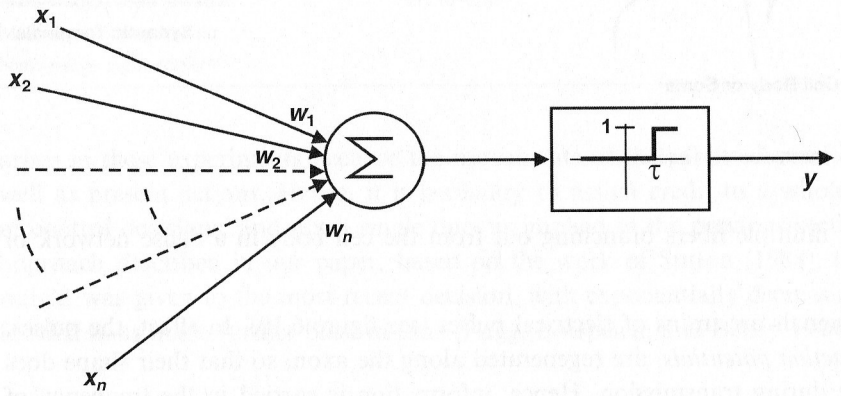
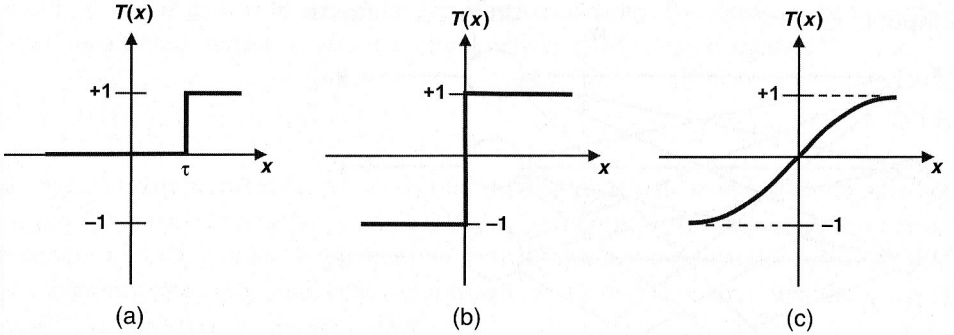


Figure 6.11
McCulloch-Pitts neuron



igure 6.12
ommon activation functions: (a) threshold, (b) step, and (c) sigmoid

$$y = T\left(\sum_{j=1}^{N} w_j x_j\right)$$

$$\textit{Threshold} \qquad T(x) = \begin{cases} 1 \; if \; x \geq \tau \\ 0 \; if \; x < \tau \end{cases}$$

$$\textit{Step} \qquad T(x) = \begin{cases} +1 \; if \; x \geq \tau \\ -1 \; if \; x < \tau \end{cases}$$

$$\textit{Sigmoid} \qquad T(x) = \frac{1}{1+e^{-x}}$$

Redes Neuronales

La red se entrena con ejemplos entrada-salida

- *Backpropagation*: ajusta los *w* observando el error

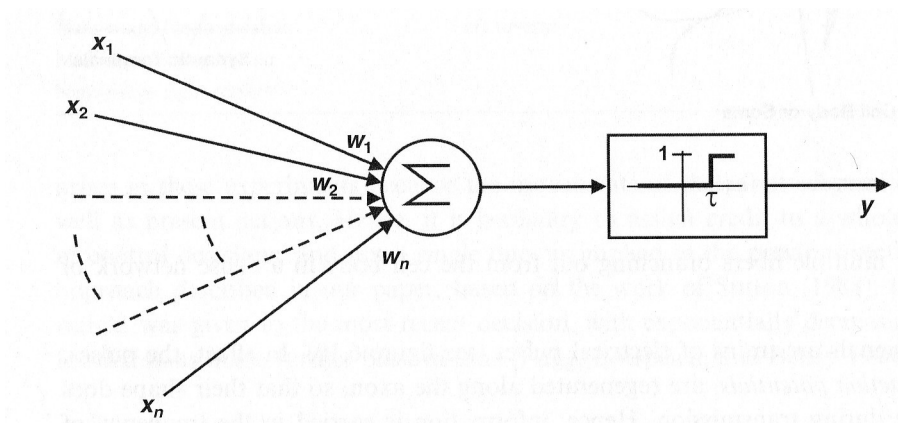
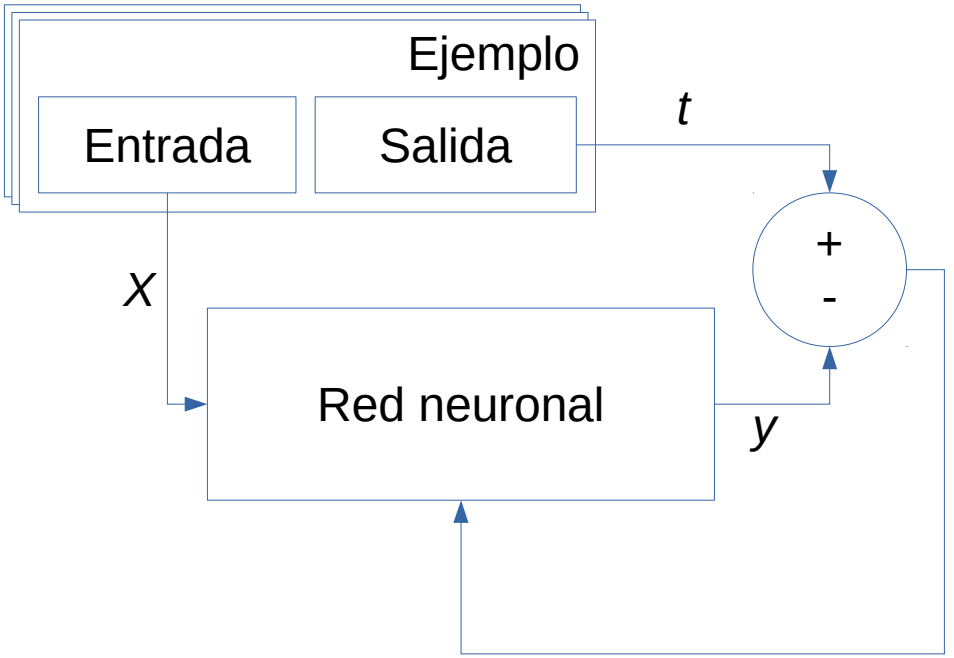


Figure 6.11
McCulloch-Pitts neuron

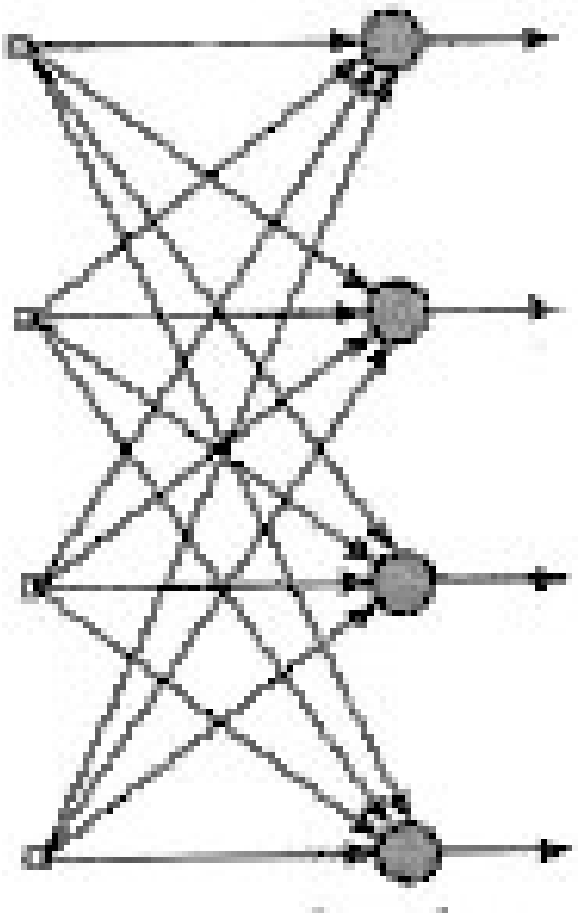
$$y = T\left(\sum_{j=1}^{N} w_j x_j\right)$$



$$w_i \leftarrow w_i + \Delta w_i$$
$$\Delta w_i = \eta (t - y) x_i$$

Redes Neuronales

- Capa única con *feed-forward*
  - Capa de entrada
  - Capa de salida
- Buenas para problemas de clasificación y filtrado de ruido.
- Convergencia rápida.



Redes Neuronales

Red *feed-forward*, una capa oculta

- Backpropagation se puede extender a múltiples capas
- “Aproximador universal”
- Entrenamiento lento
- Evaluación rápida

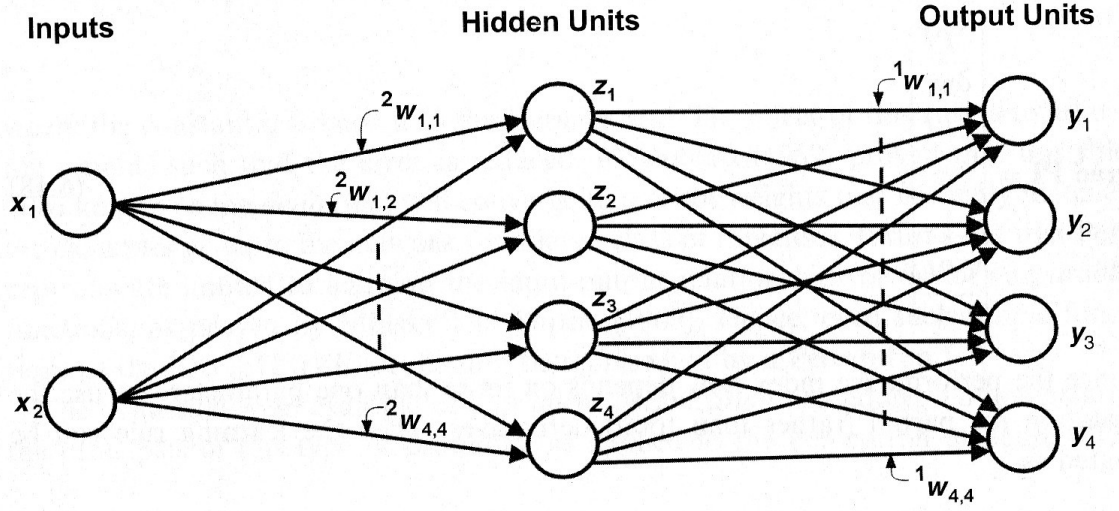
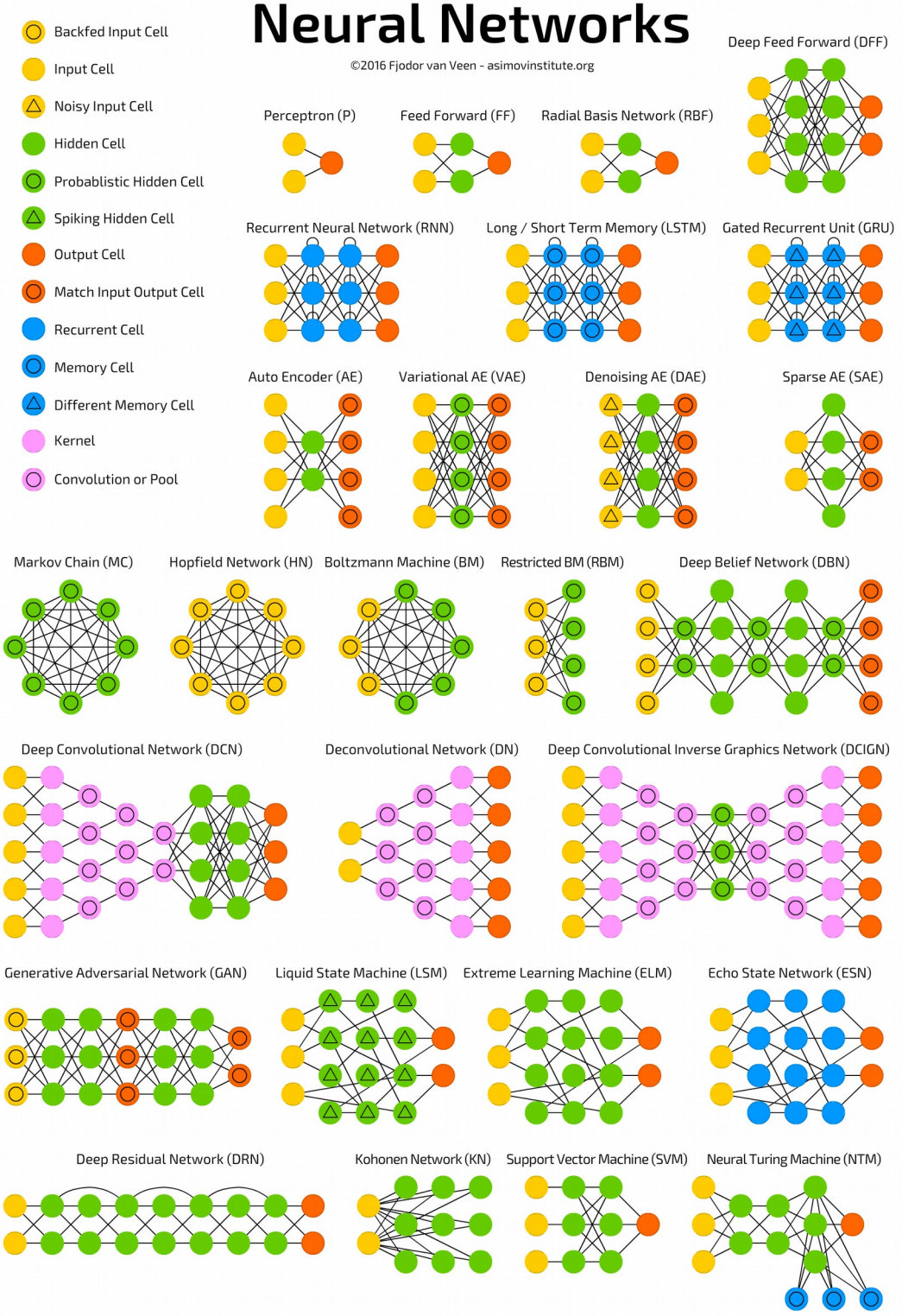


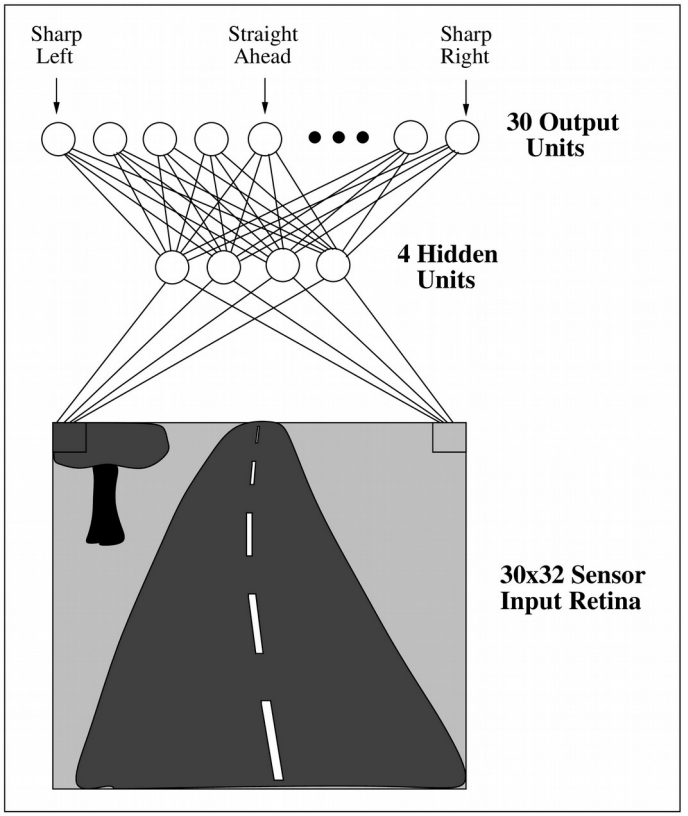
Figure 6.14
Three-layer feedfoward network showing hidden layer

A mostly complete chart of

Neural Networks



Redes Neuronales, ejemplo



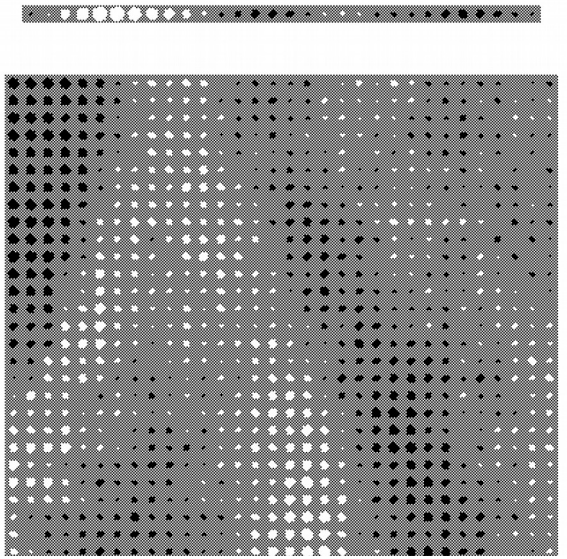


FIGURE 4.1
Neural network learning to steer an autonomous vehicle. The ALVINN system uses BACKPROPAGATION to learn to steer an autonomous vehicle (photo at top) driving at speeds up to 70 miles per hour. The diagram on the left shows how the image of a forward-mounted camera is mapped to 960 neural network inputs, which are fed forward to 4 hidden units, connected to 30 output units. Network outputs encode the commanded steering direction. The figure on the right shows weight values for one of the hidden units in this network. The 30×32 weights into the hidden unit are displayed in the large matrix, with white blocks indicating positive and black indicating negative weights. The weights from this hidden unit to the 30 output units are depicted by the smaller rectangular block directly above the large block. As can be seen from these output weights, activation of this particular hidden unit encourages a turn toward the left.

Tom M. Mitchell. Machine Learning. McGraw Hill. 1997

Caja negra vs blanca

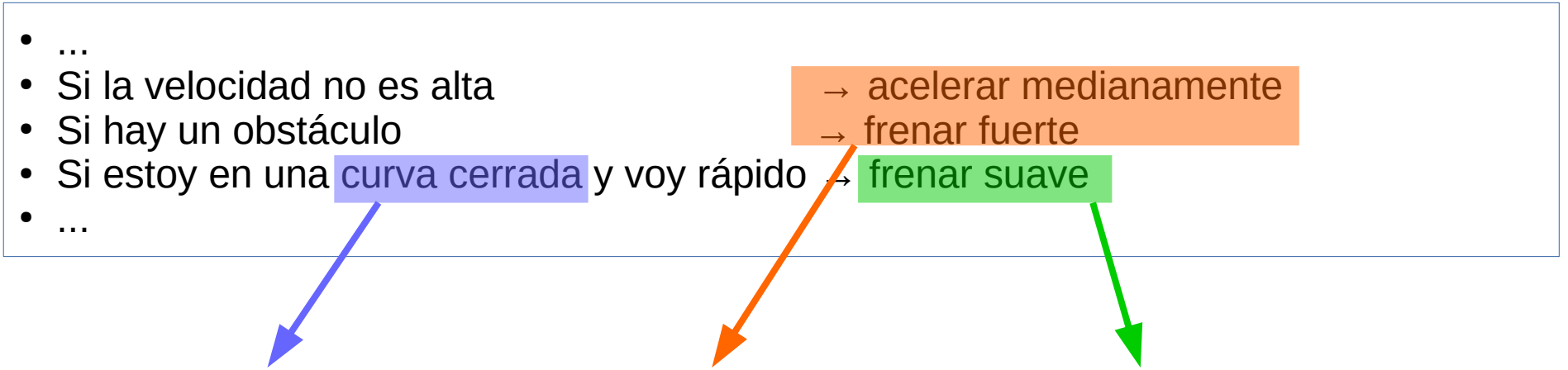
- Redes neuronales
  - Se entrenan con ejemplos
  - Los parámetros aprendidos son imposibles de interpretar
  - Conocemos el entorno
- Reglas
  - Se construyen por el programador
  - El controlador “tiene sentido” y puede ser razonado
  - Sabemos qué queremos

Motores de reglas

Ejemplo: seleccionar velocidad de movimiento

- Si la velocidad no es alta → acelerar medianamente
- Si estoy cerca de la velocidad máxima → no acelerar
- Si supero velocidad máxima → frenar suavemente
- Si está vibrando mucho → frenar suavemente
- Si hay un obstáculo → frenar fuerte
- Si estoy en una curva no muy suave → no acelerar
- Si estoy en una curva cerrada y voy rápido → frenar suave
- ...

Motores de reglas



Entrada

Correspondencia entre valores medidos y criterios de las reglas

Resolución de conflictos

Puede haber varias reglas que aplican simultáneamente

Salida

Correspondencia entre acciones y valores para los actuadores

Motores de reglas

- Entrada y salida: *Fuzzyfication* y *Defuzzyfication*
  - Umbrales, funciones D:{true, false}
  - Lógica difusa, funciones de membresía D:{0..1}
- Conflictos
  - Prioridades explícitas: cada regla tiene una prioridad
  - Combinación aritmética
  - Votación

Controlador fuzzy

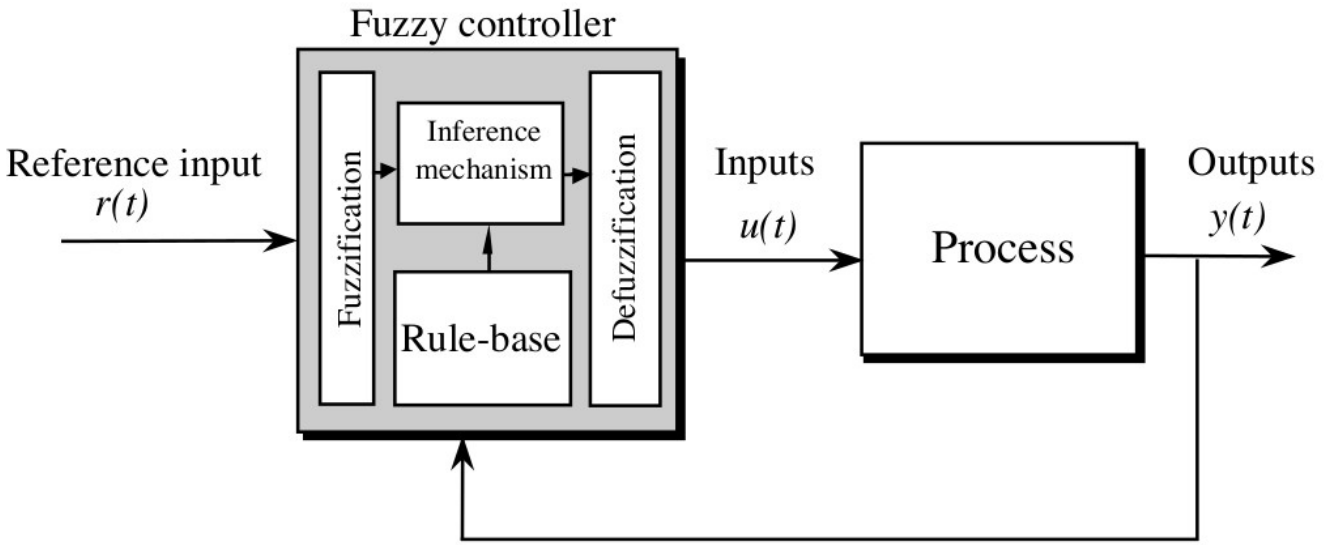


FIGURE 2.1 Fuzzy controller.

Controlador fuzzy

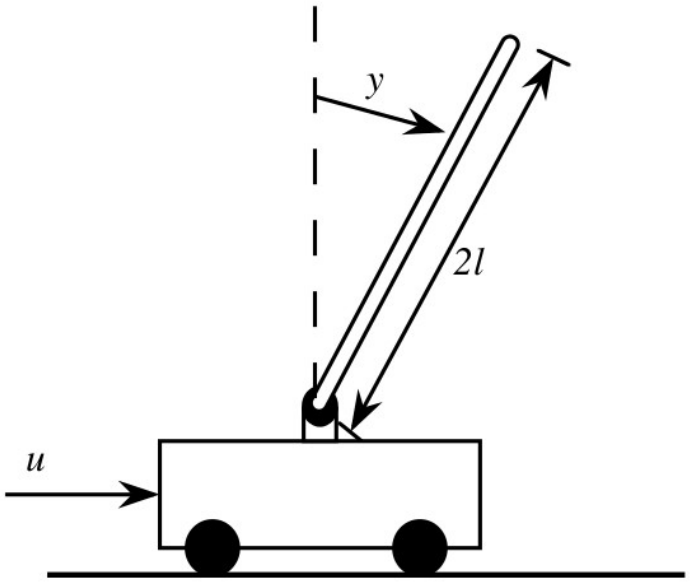


FIGURE 2.2 Inverted pendulum on a cart.

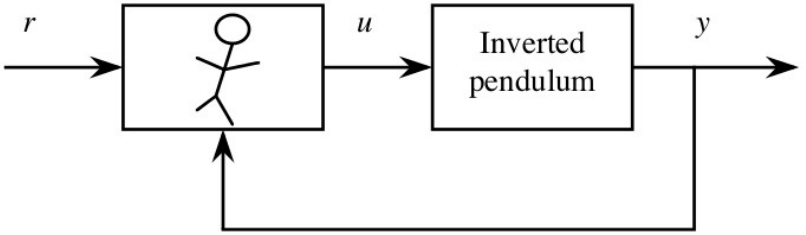


FIGURE 2.3 Human controlling an inverted pendulum on a cart.

Variables de decisión:

$$e(t) = r(t) - y(t)$$

$$\frac{d}{dt} e(t)$$

Controlador fuzzy

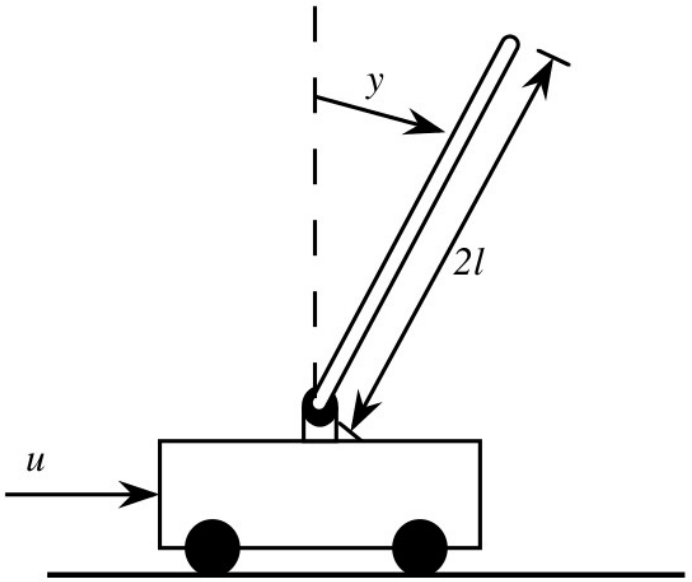


FIGURE 2.2 Inverted pendulum on a cart.

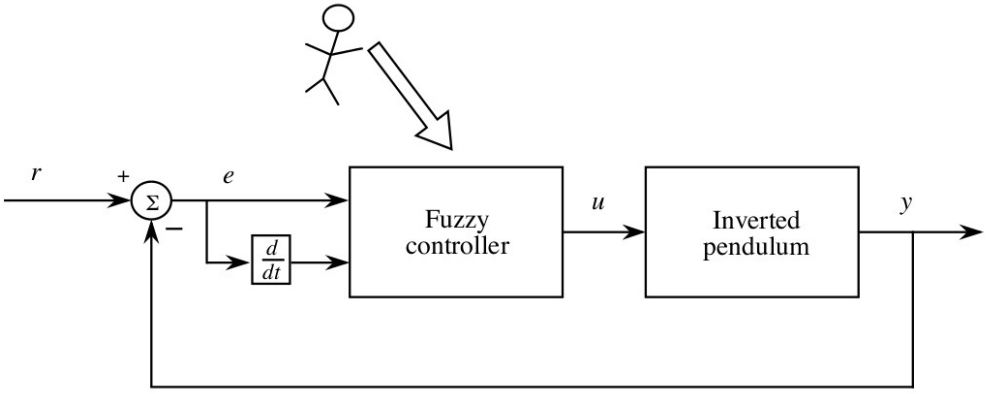


FIGURE 2.4 Fuzzy controller for an inverted pendulum on a cart.

Controlador fuzzy

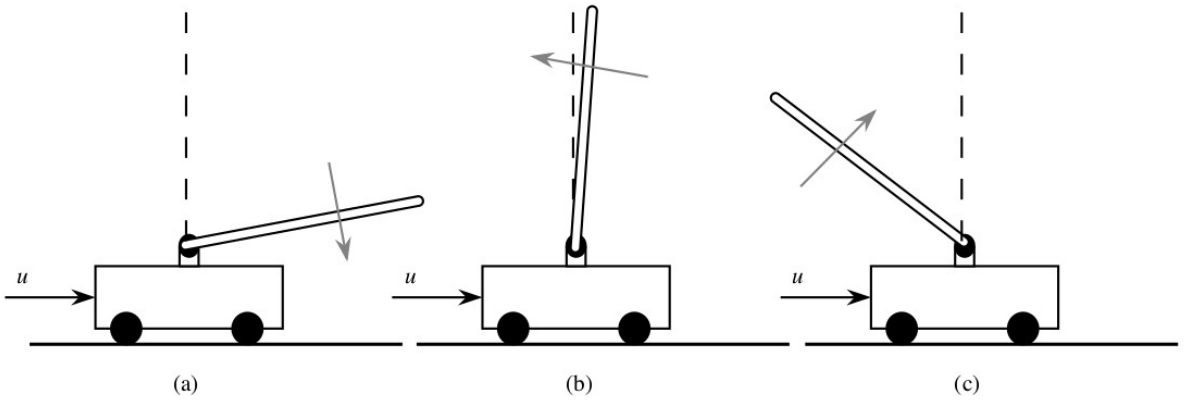


FIGURE 2.5 Inverted pendulum in various positions.

(a) *If error is neglarge and change-in-error is neglarge*
→ *Then force is poslarge*

(b) *If error is zero and change-in-error is possmall*
→ *Then force is negsmall*

(c) *If error is poslarge and change-in-error is negsmall*
→ *Then force is negsmall*

Controlador fuzzy

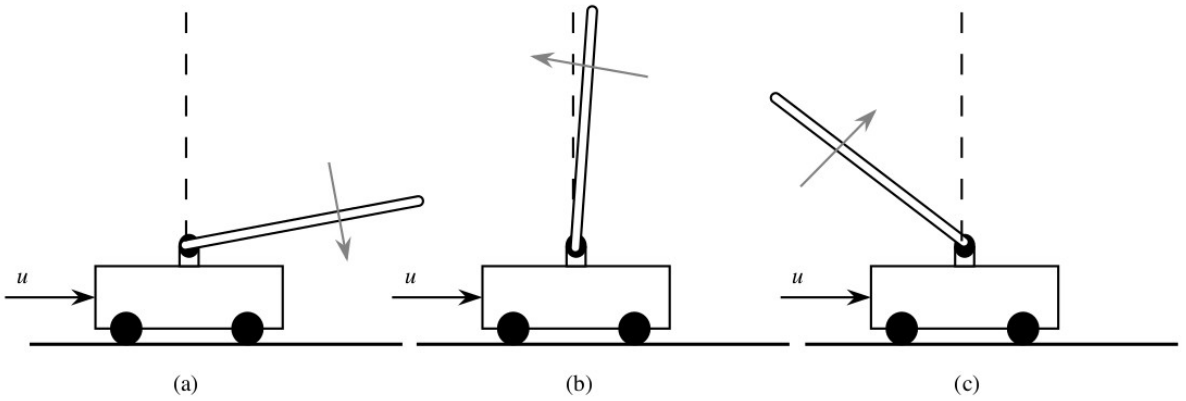


FIGURE 2.5 Inverted pendulum in various positions.

TABLE 2.1 Rule Table for the Inverted Pendulum

"force" u		"change-in-error" $\dot{e}$				
		-2	-1	0	1	2
"error" e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

Controlador fuzzy

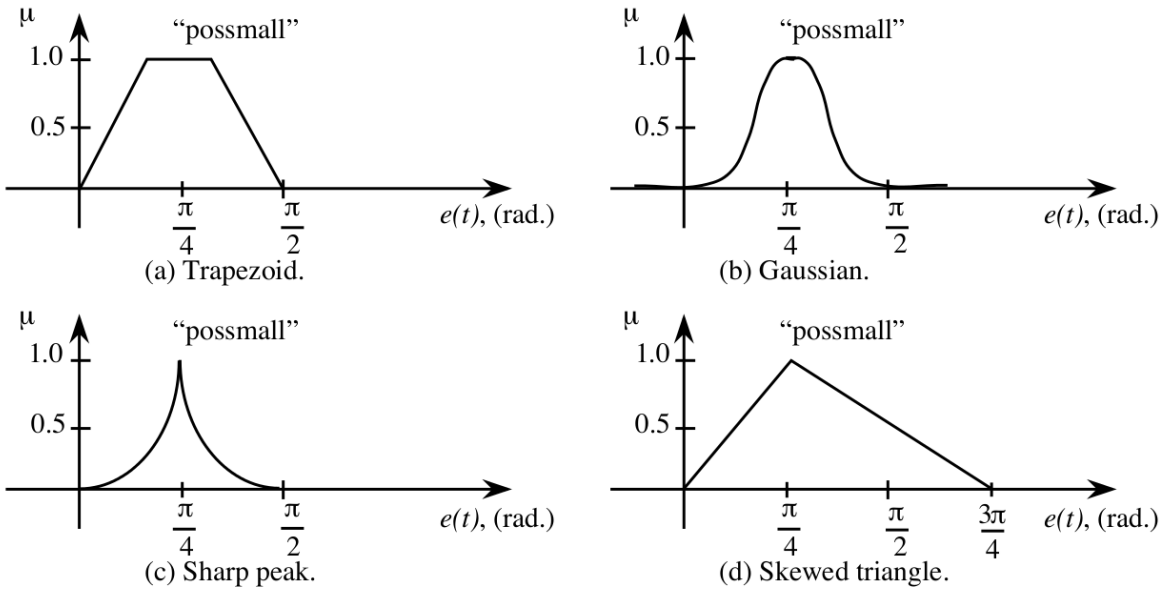


FIGURE 2.7 A few membership function choices for representing "error is possmall."

Controlador fuzzy

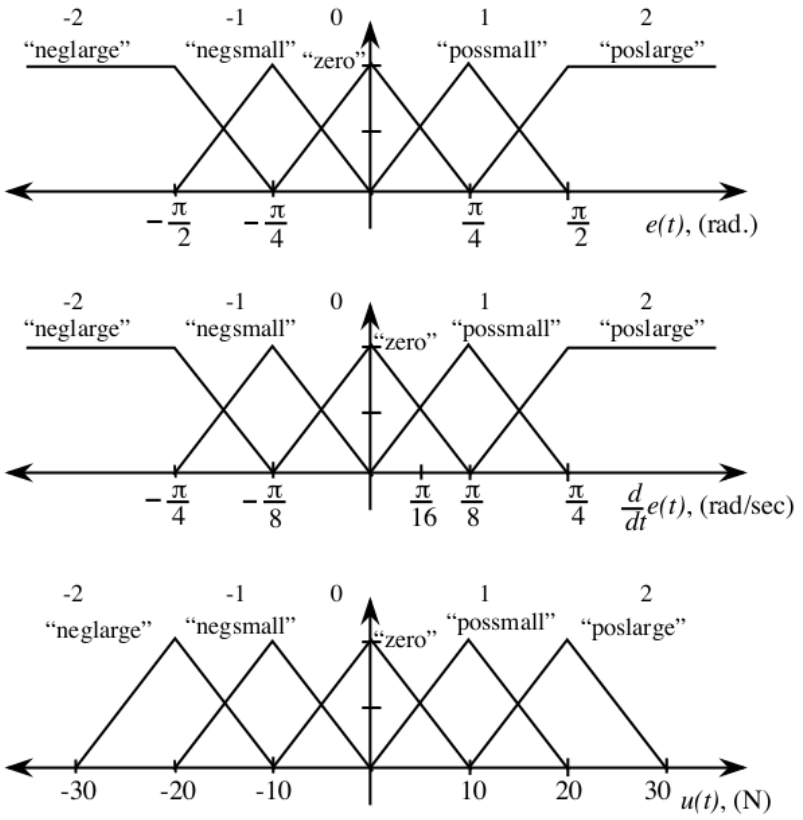


FIGURE 2.9 Membership functions for an inverted pendulum on a cart.

Controlador fuzzy

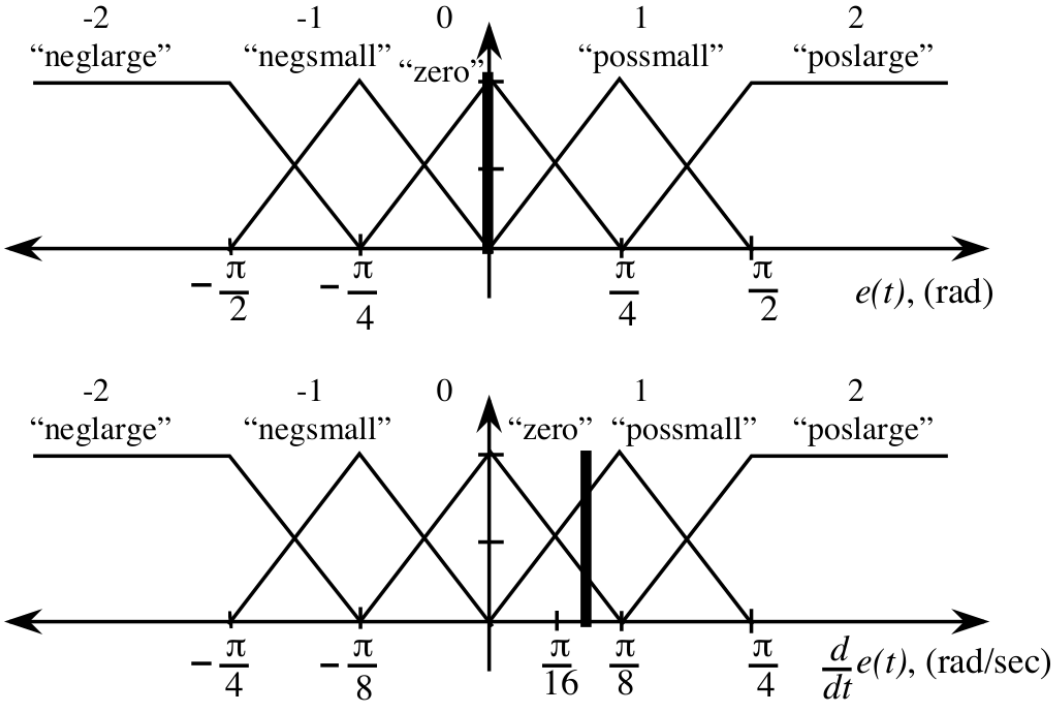


FIGURE 2.12 Input membership functions with input values.

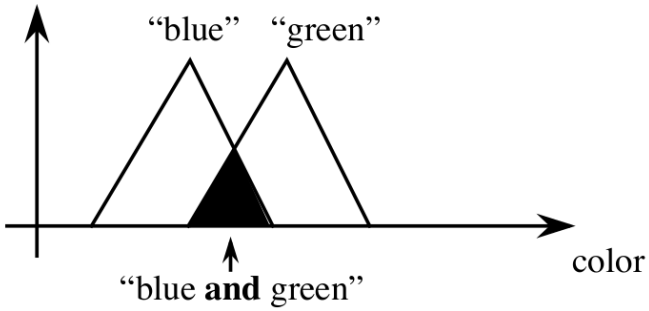


FIGURE 2.22 A membership function for the "and" of two membership functions.

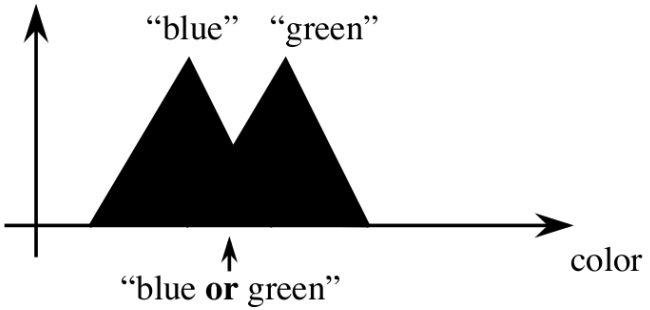


FIGURE 2.23 A membership function for the "or" of two membership functions.

Controlador fuzzy

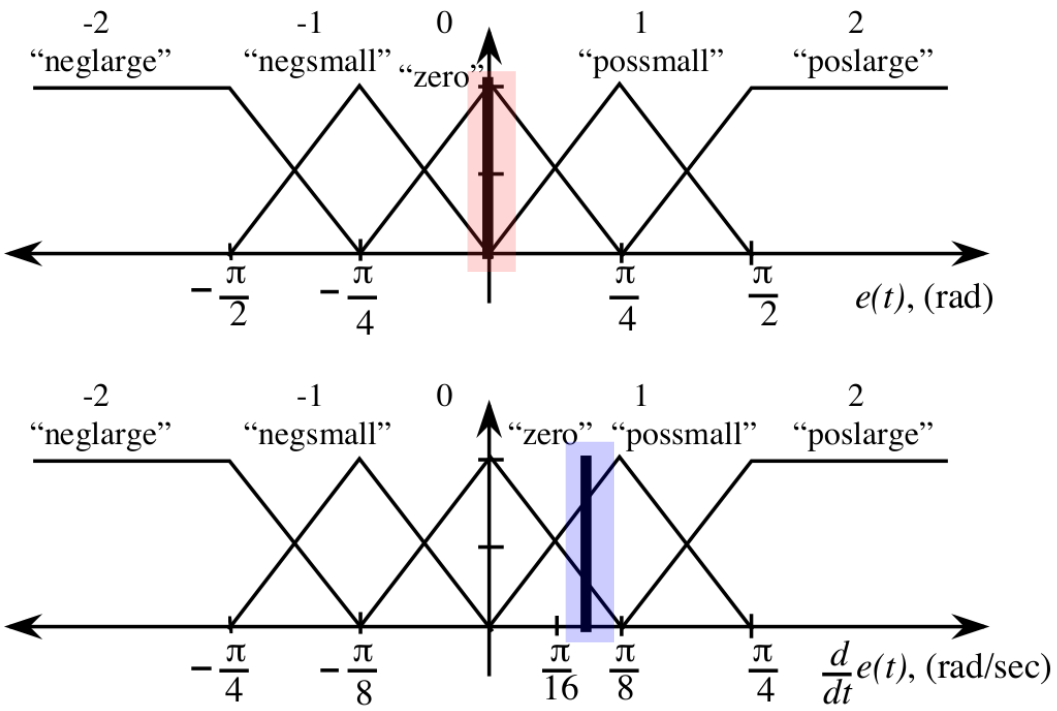


FIGURE 2.12 Input membership functions with input values.

TABLE 2.2 Rule Table for the Inverted Pendulum with Rules That Are "On" Highlighted.

"force" u		"change-in-error" $\dot{e}$				
		-2	-1	0	1	2
"error" e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

Controlador fuzzy

TABLE 2.2 Rule Table for the Inverted Pendulum with Rules That Are "On" Highlighted.

"force" u		"change-in-error" $\dot{e}$				
		-2	-1	0	1	2
"error" e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

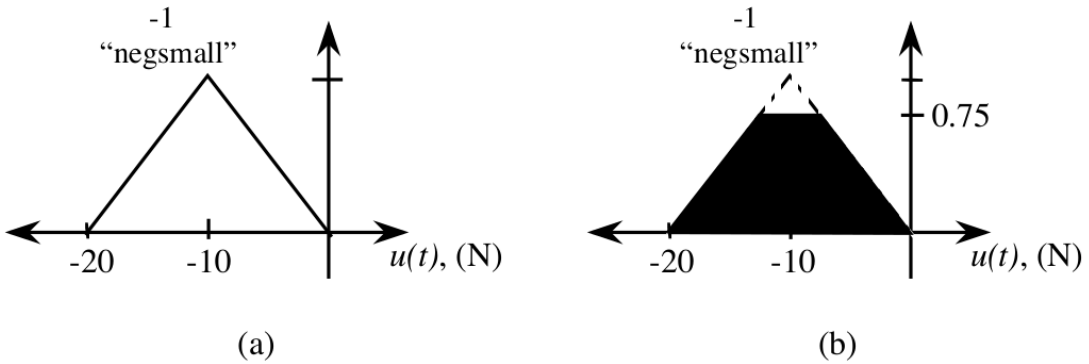


FIGURE 2.14 (a) Consequent membership function and (b) implied fuzzy set with membership function $\mu_{(2)}(u)$ for rule (2).

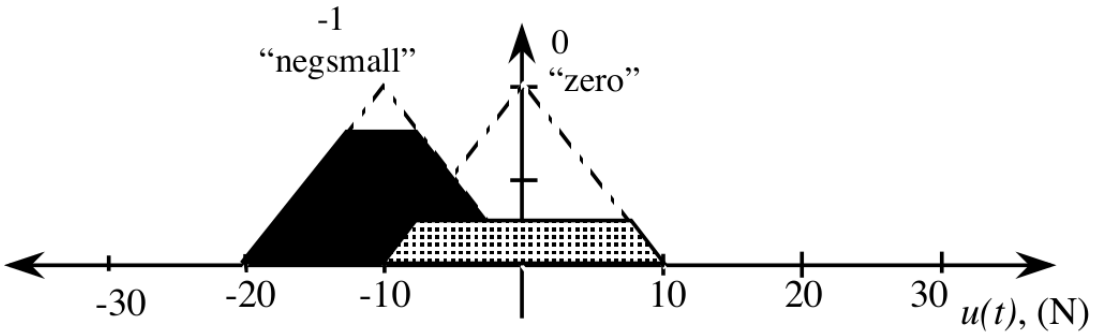


FIGURE 2.15 Implied fuzzy sets.

Controlador fuzzy

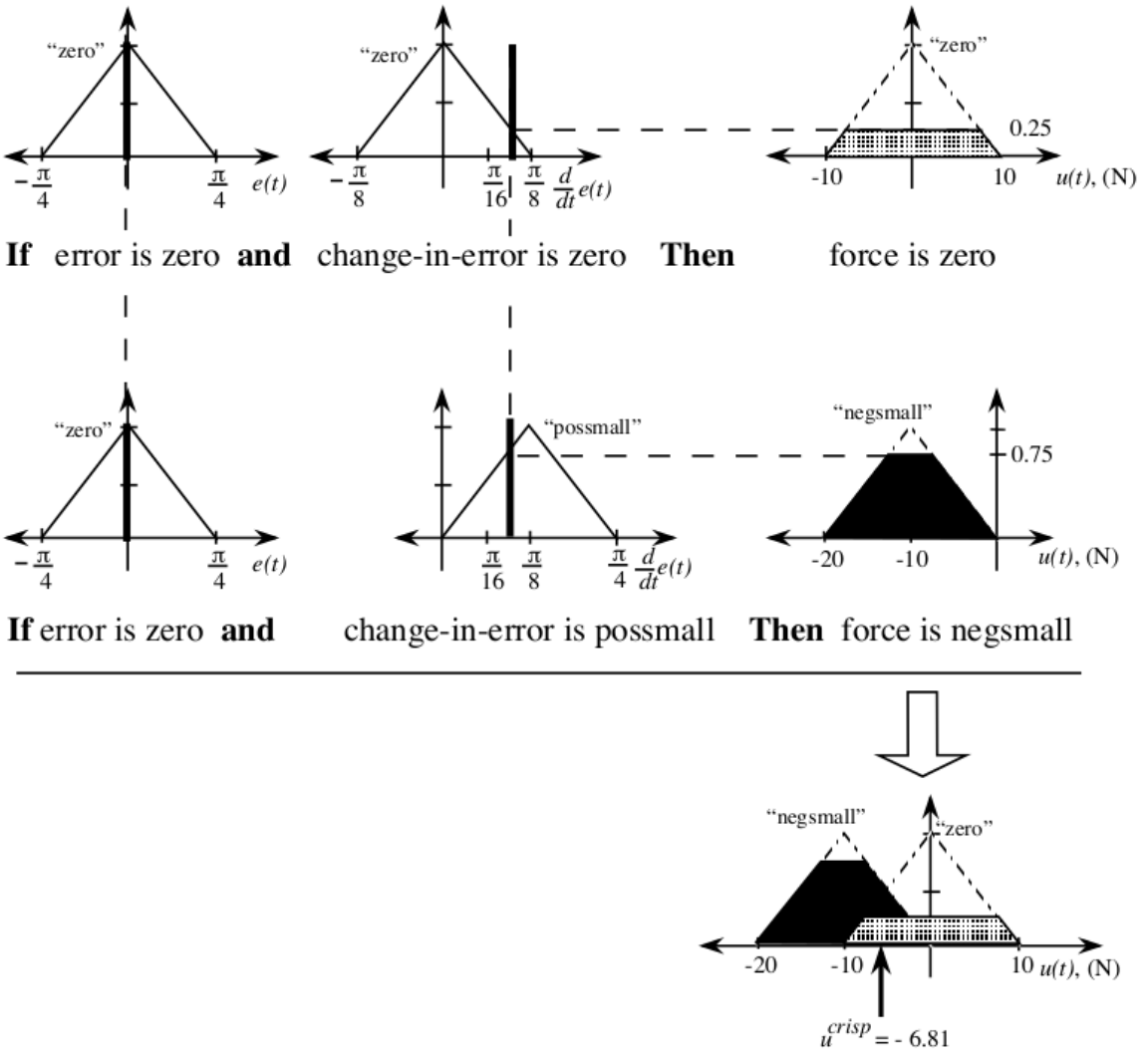
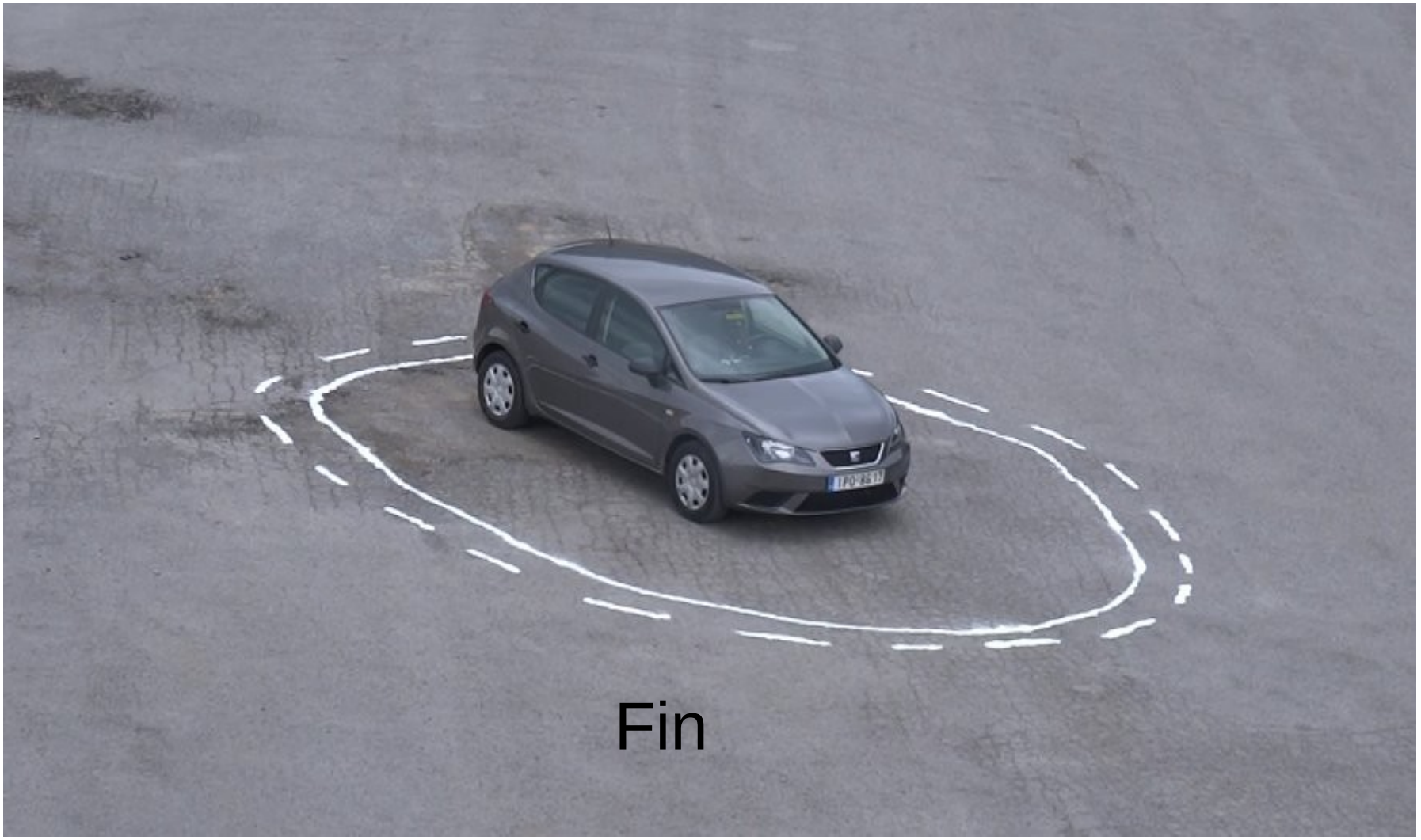


FIGURE 2.19 Graphical representation of fuzzy controller operations.


Fin

Referencias

- George A. Bekey, Autonomous Robots, MIT Press - 2005
- Hellerstein J.L., Diao Y., et al, Feedback Control of Computing Systems, Wiley – 2004
- Passino M. K., Yurkovich S., Fuzzy Control, Adison-Wesley – 1998