

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL IPN

**ROBOTS QUE
JUEGAN FOOTBALL**

Doctorado en Computación

Alumno: M. en C. Farid García Lamont

Asesor: Dr. José Matías Alvarado Mentado

Contenido

- Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV).
 - Departamento de Computación
- Universidades Politécnicas
 - Universidad Politécnica de Pachuca (UPP)

Contenido

→ Robótica

- Historia
- Definiciones
- Aplicaciones
 - ❖ Industriales
 - ❖ Terapéuticos
 - ❖ Medicina
 - ❖ Entretenimiento

Contenido

→ Torneo Robocup

- ¿Qué es RoboCup?
- Historia del RoboCup
- Objetivos del RoboCup
- Categorías de Robots
- Dr. Raúl Rojas (RR)
 - ❖ Fu-Fighters (Equipo de trabajo)
 - ❖ Campeonatos obtenidos

Contenido

- Funcionamiento
 - ❖ Robótica
 - ❖ Visión Artificial
 - ❖ Teoría de Control
 - ❖ Redes Neuronales (RN)
 - ❖ Computación
 - ❖ Electrónica

Contenido

→ Propuesta

- Limitaciones de robots de RR
- Tracking en superficies irregulares
- Ejemplos de visión artificial
- Modelo Basado en Apariencias (MBA)
 - ❖ Espacios propios (Algebra lineal)
 - ❖ Interpolaciones
 - ❖ Reconocimiento de objetos
 - ❖ RN para calculo de ruta optima (considerando la superficie)

CINVESTAV

El Cinvestav fue creado un por Decreto Presidencial que expidió el Lic. Adolfo López Mateos, el 17 de abril de 1961, que es un organismo descentralizado de interés público, con personalidad jurídica y patrimonio propios.

Los objetivos fundamentales son: preparar investigadores y profesores especializados que promuevan la superación de la enseñanza y generar condiciones para la realización de investigaciones originales en diversas áreas científicas y tecnológicas.

Sección de Computación

En 1983, Sección de Computación dentro del Departamento de Ingeniería Eléctrica. El propósito fue hacer frente a las necesidades crecientes que se tenían en aquel entonces y que se siguen teniendo en México, en las ramas de la computación, la informática y sus aplicaciones. El principal objetivo de la Sección fue el realizar investigaciones científicas y desarrollos tecnológicos en la disciplina de la Computación.

Universidades Politécnicas

El modelo académico de estas universidades está sustentado en programas educativos diseñados con base en competencias laborales y profesionales, los cuales son el resultado de un cuidadoso análisis de las necesidades y oportunidades de desarrollo regional.

Universidades Politécnicas

- Aguascalientes
- Baja California
- Morelos
- Hidalgo (Pachuca y Tulancingo)
- Puebla
- Querétaro
- San Luis Potosí
- Valle de México

Universidades Politécnicas

- Zacatecas
- Tlaxcala
- Sinaloa
- Chiapas
- Durango (Durango y Gómez Palacio)
- Guanajuato
- Jalisco

UNIVERSIDAD POLITECNICA DE PACHUCA



- Ingeniería en Mecatrónica
- Ingeniería en Telemática
- Biotecnología
- * Ingeniería en software
- * Terapia física
- * Finanzas
- * Bioingeniería

Robótica

Robot: Aparato automático capaz de manipular objetos o ejecutar una o varias operaciones según un programa establecido.

Robótica: Conjunto de técnicas utilizadas para el diseño y construcción de robots y la puesta en práctica de sus aplicaciones.

Definiciones

Cibernética: Ciencia que estudia los mecanismos de comunicación y de control en las máquinas y los seres vivos.

Mecatrónica:

¿Qué es RoboCup?

La Robocup es un evento con diversas categorías donde el objetivo es desarrollar máquinas y programas "inteligentes" capaces de jugar fútbol sin que tengan que intervenir seres humanos durante el transcurso del partido.

Categorías

Liga de simulación (simulation league)



Categorías

Liga robots pequeños (small size league)



Categorías

Liga robots medianos (middle size league)



Categorías

Liga de cuatro patas (four legged league)



Categorías

Liga de humanoides (humanoid league)



Visión de robots futbolistas pequeños (small-size)



Desplazamiento en superficies lisas



Habilidades de robots pequeños

1. Libertad de dribbling
2. Angulo de dribbling
3. Distancia del dribbling a la meta
4. Espacio disponible
5. Angulo de pase
6. Distancia mínima tangencial
7. Tiempo de espera durante el pase
8. Libertad de dribbling para el receptor después del pase

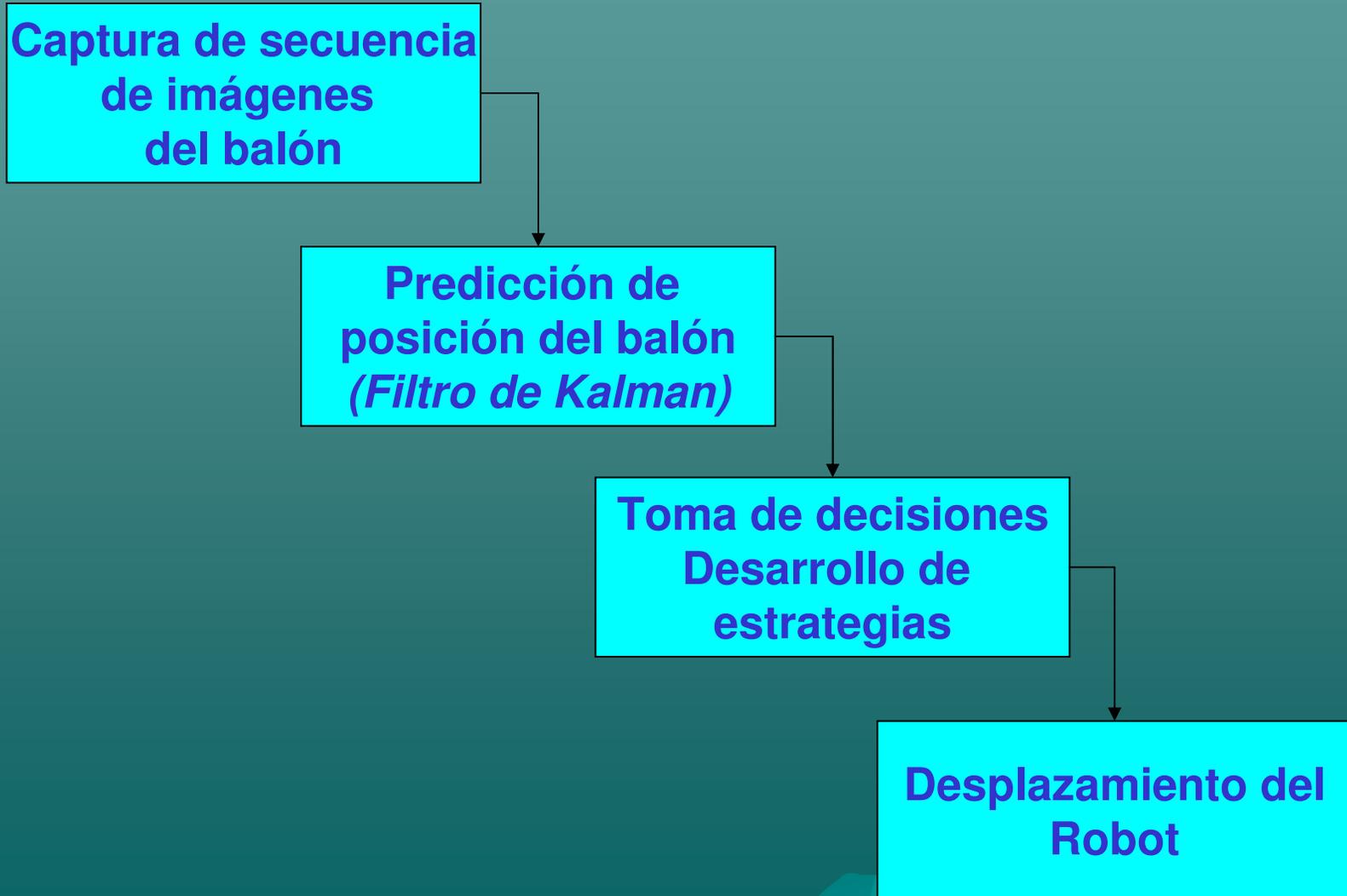
¿Cómo ejecutan esas habilidades?

Captura de secuencia
de imágenes
del balón

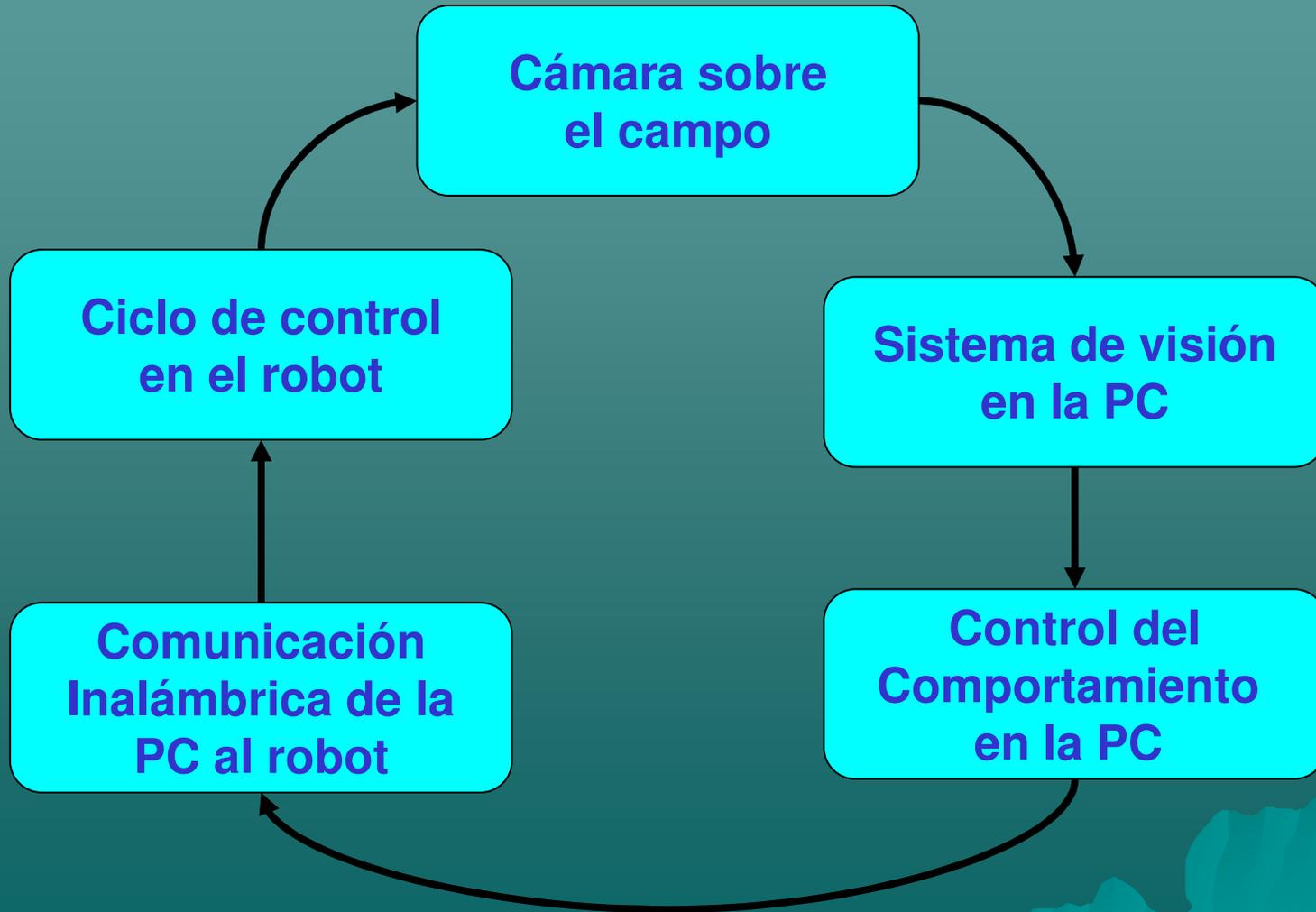
Predicción de
posición del balón
(*Filtro de Kalman*)

Toma de decisiones
Desarrollo de
estrategias

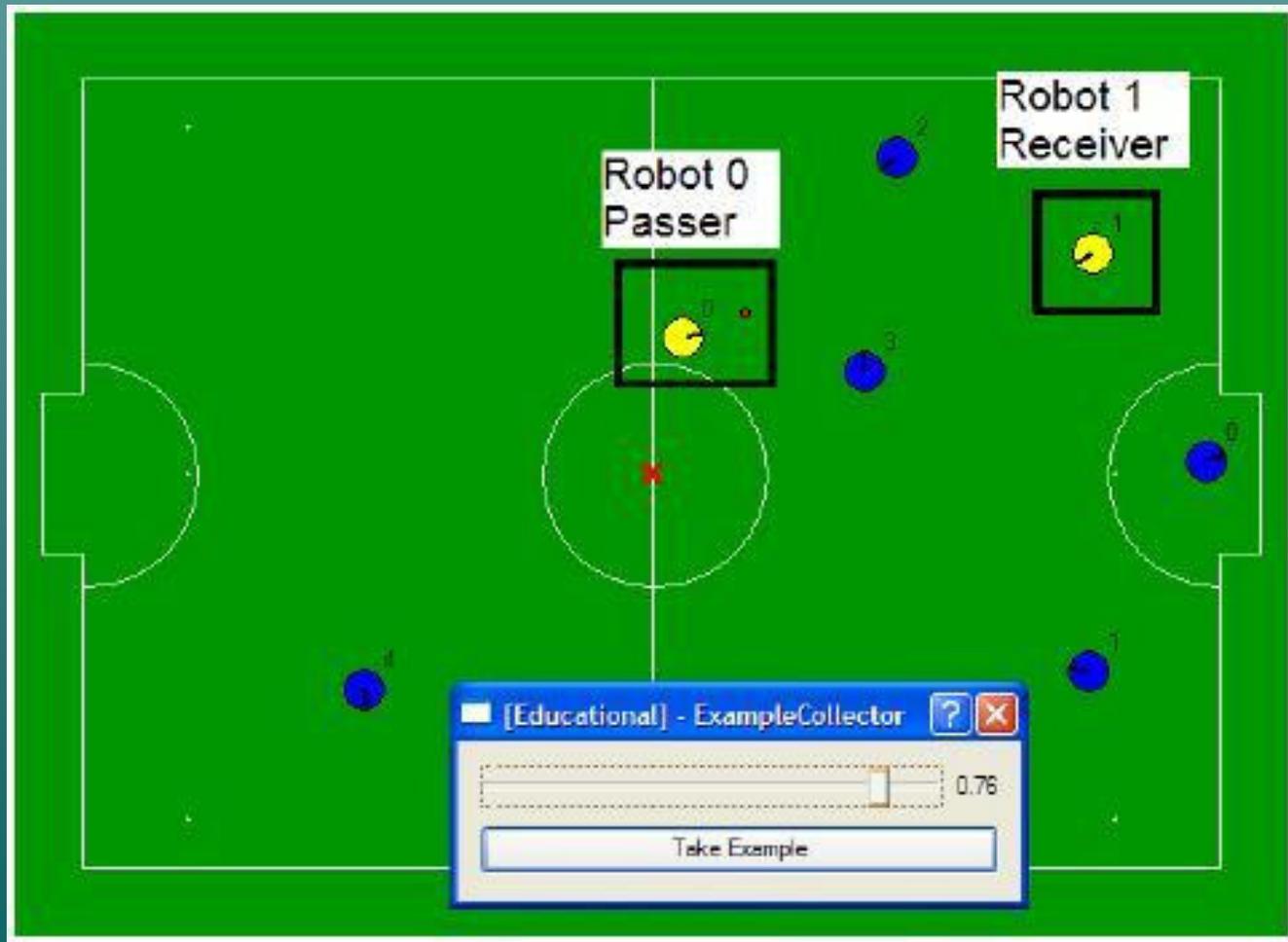
Desplazamiento del
Robot



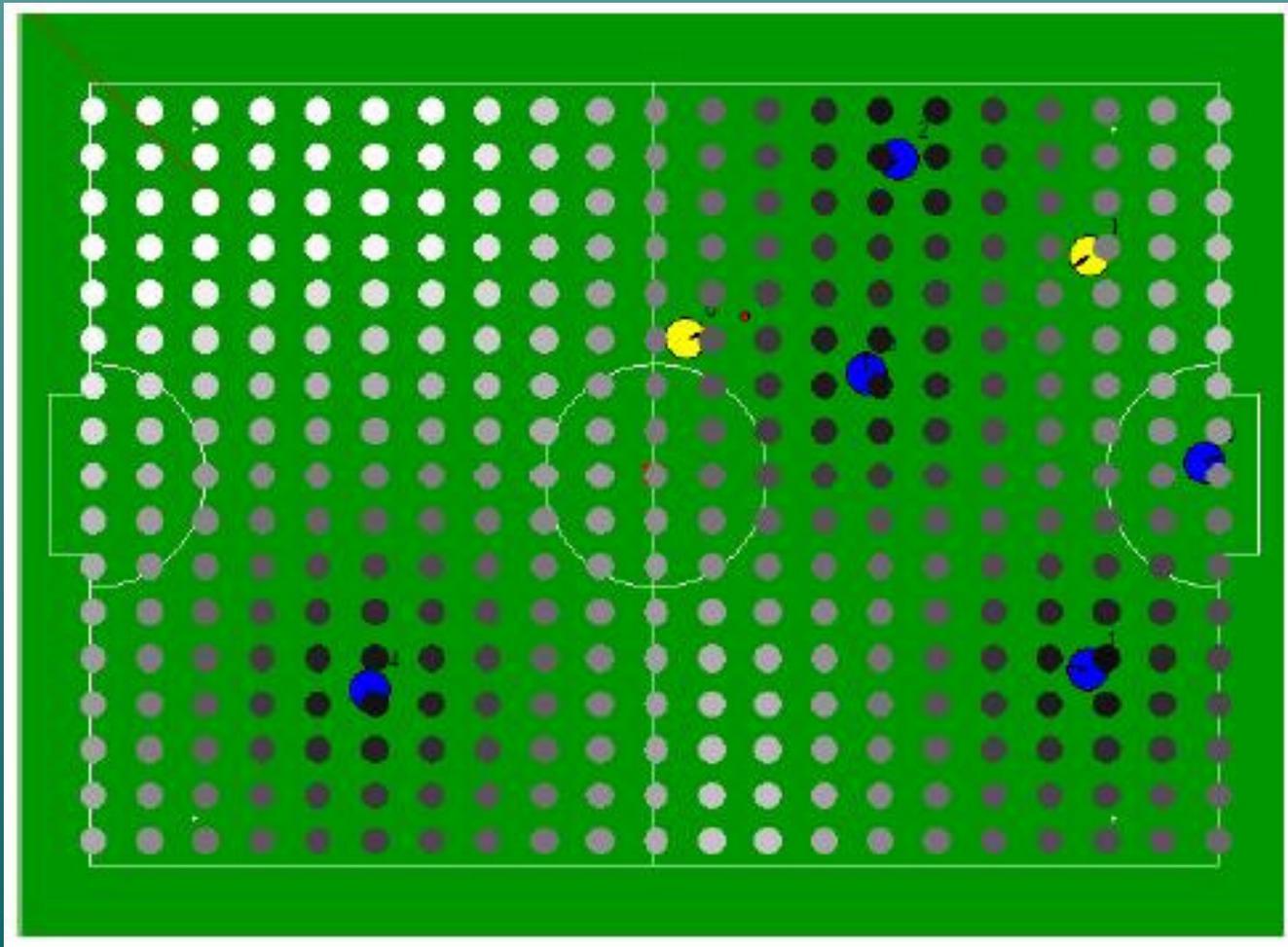
¿Cómo se ejecutan esas habilidades?



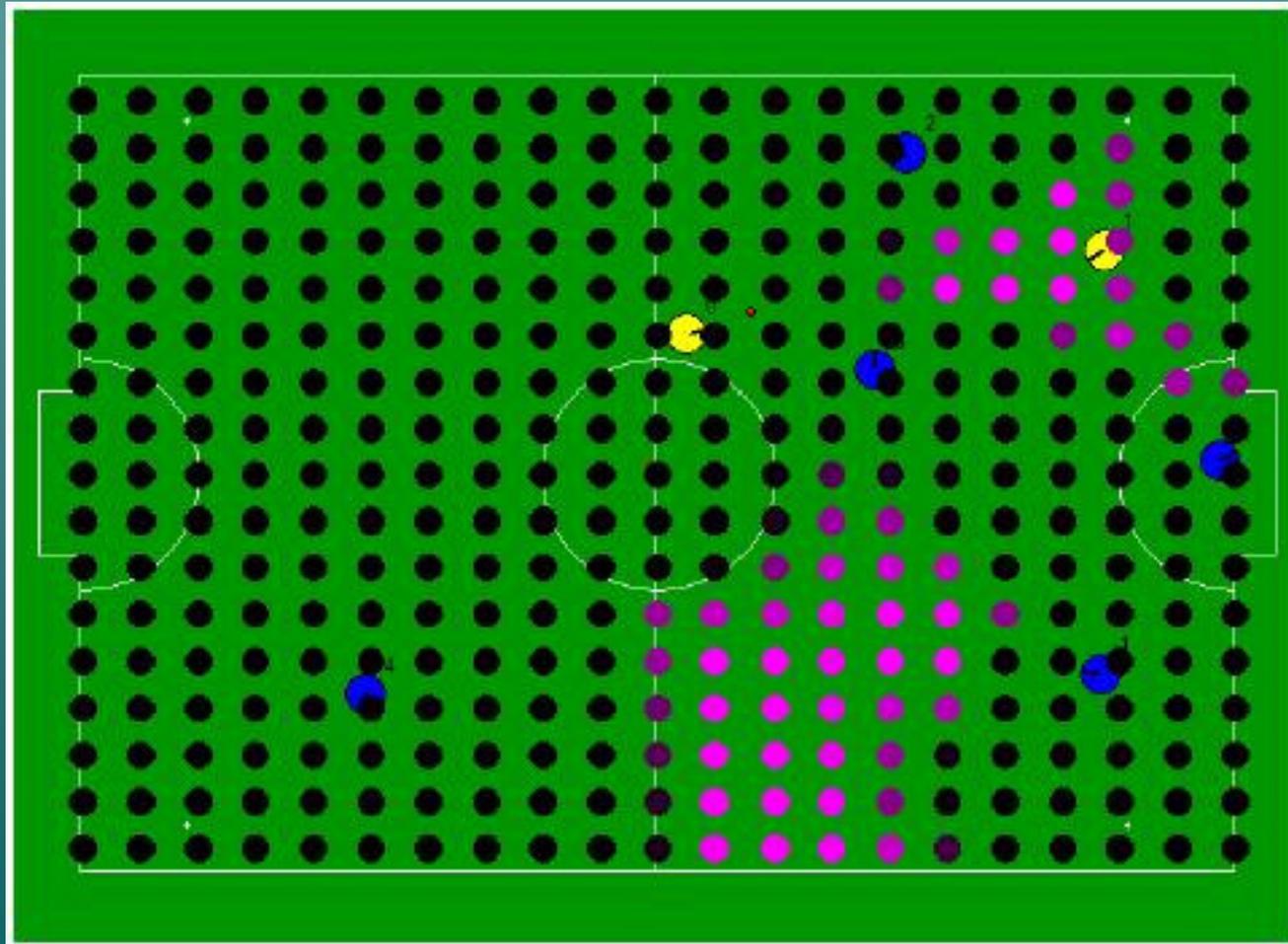
Desplazamiento y seguimiento con robots pequeños



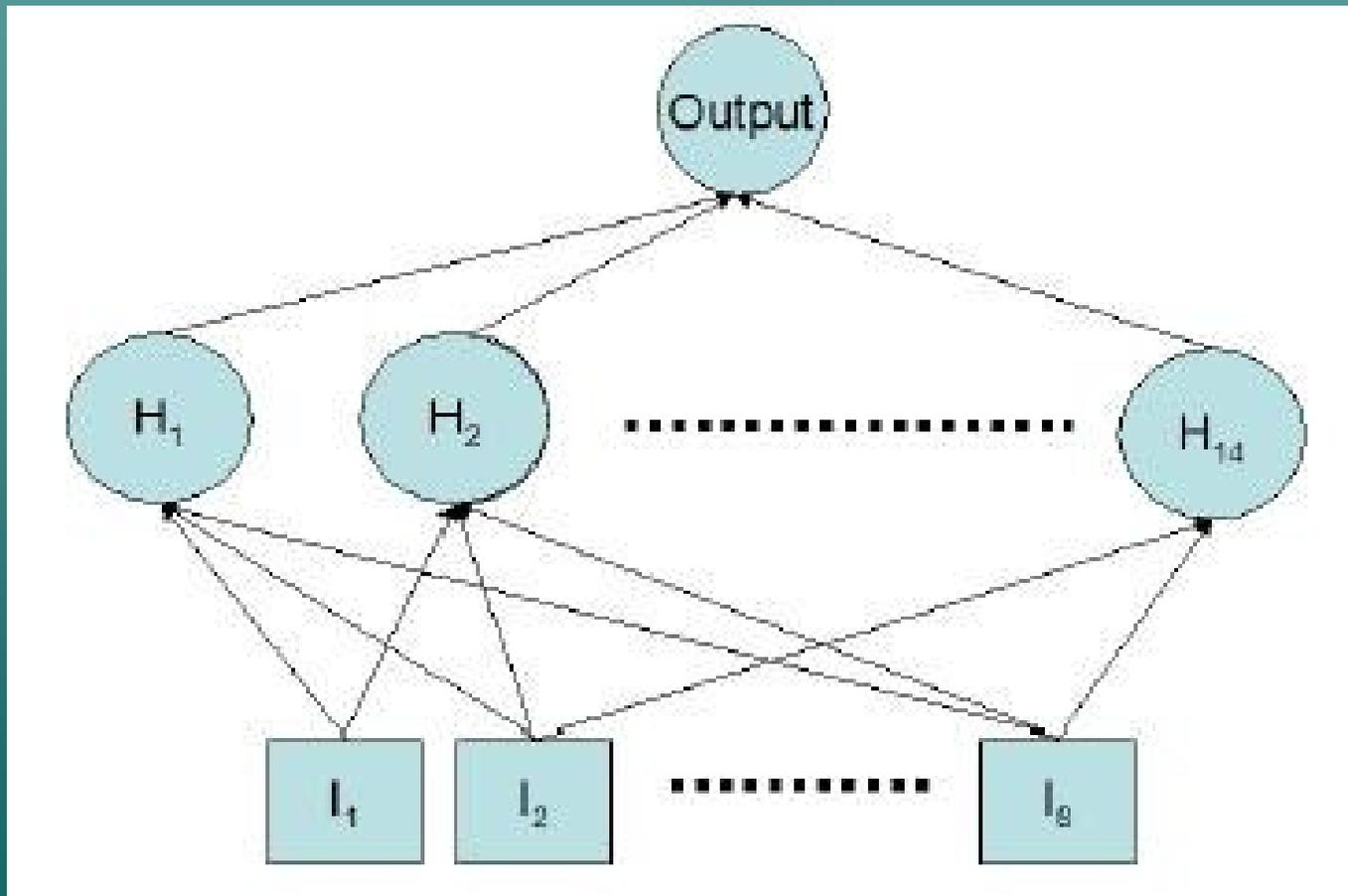
Desplazamiento y seguimiento con robots pequeños



Desplazamiento y seguimiento con robots pequeños



Desplazamiento y seguimiento en robots pequeños



Desplazamiento y seguimiento en robots pequeños

Kategorie	Desired Output	Actual Output	
<input checked="" type="checkbox"/> 5 Aus der Verteidigung raus			
<input checked="" type="checkbox"/> 5 Ballacks Passse			
NewDataSet	1	1.06651	■
NewDataSet	1	0.967294	■
NewDataSet	1	0.953524	■
NewDataSet	1	0.837924	■
NewDataSet	1	0.948919	■
NewDataSet	1	0.958833	■
NewDataSet	1	0.795119	■
NewDataSet	1	0.935119	■
NewDataSet	1	1.0593	■
NewDataSet	1	1.09876	■
NewDataSet	1	1.00938	■
NewDataSet	1	1.0217	■
NewDataSet	1	0.895954	■
NewDataSet	1	1.05437	■
<input checked="" type="checkbox"/> 5 Dribbling d. Mitte			
<input checked="" type="checkbox"/> 5 FluegelDribbel			
<input checked="" type="checkbox"/> 5 FluegelPass			
<input checked="" type="checkbox"/> 5 Gedeckt			
<input checked="" type="checkbox"/> 5 KeinKein			
<input checked="" type="checkbox"/> 5 Rueckpresse			
<input checked="" type="checkbox"/> 5 Seitenwechsel			
<input checked="" type="checkbox"/> 5 Solele			
<input checked="" type="checkbox"/> 7 Angriff zuruecklegen			
<input checked="" type="checkbox"/> General Example			
NewDataSet	1	0.696522	■
NewDataSet	-1	-1.03387	■
NewDataSet	-1	-1.60047	■

Example Management

Check All Remove

Load Default Load...

Save Default Save As

New Category

Neural Net

Init Train

Load Default Load...

Save Default Save As

Set Training Set

Training Cycles

Hidden Layer Nodes

Success Ratio 0.978723

Visualizer

PP-DribbelFreiheit

Input Visual

Output Visual

Teoría de Control

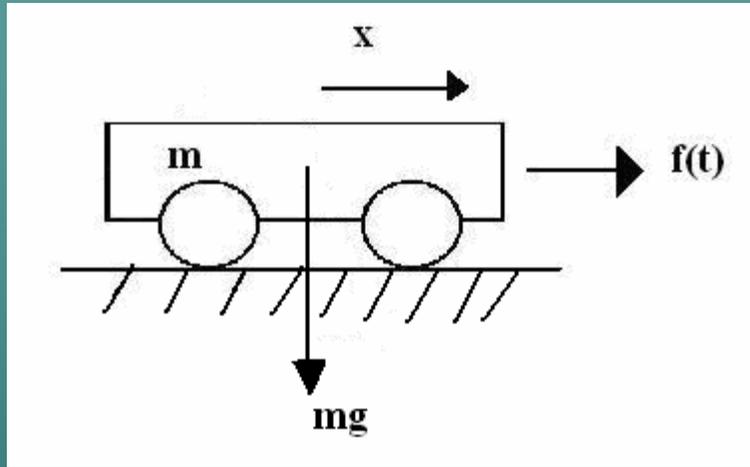


Diagrama de cuerpo libre del robot.

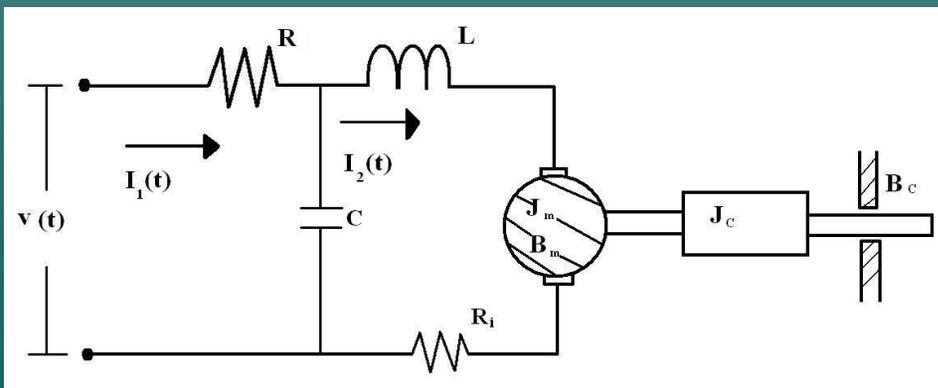


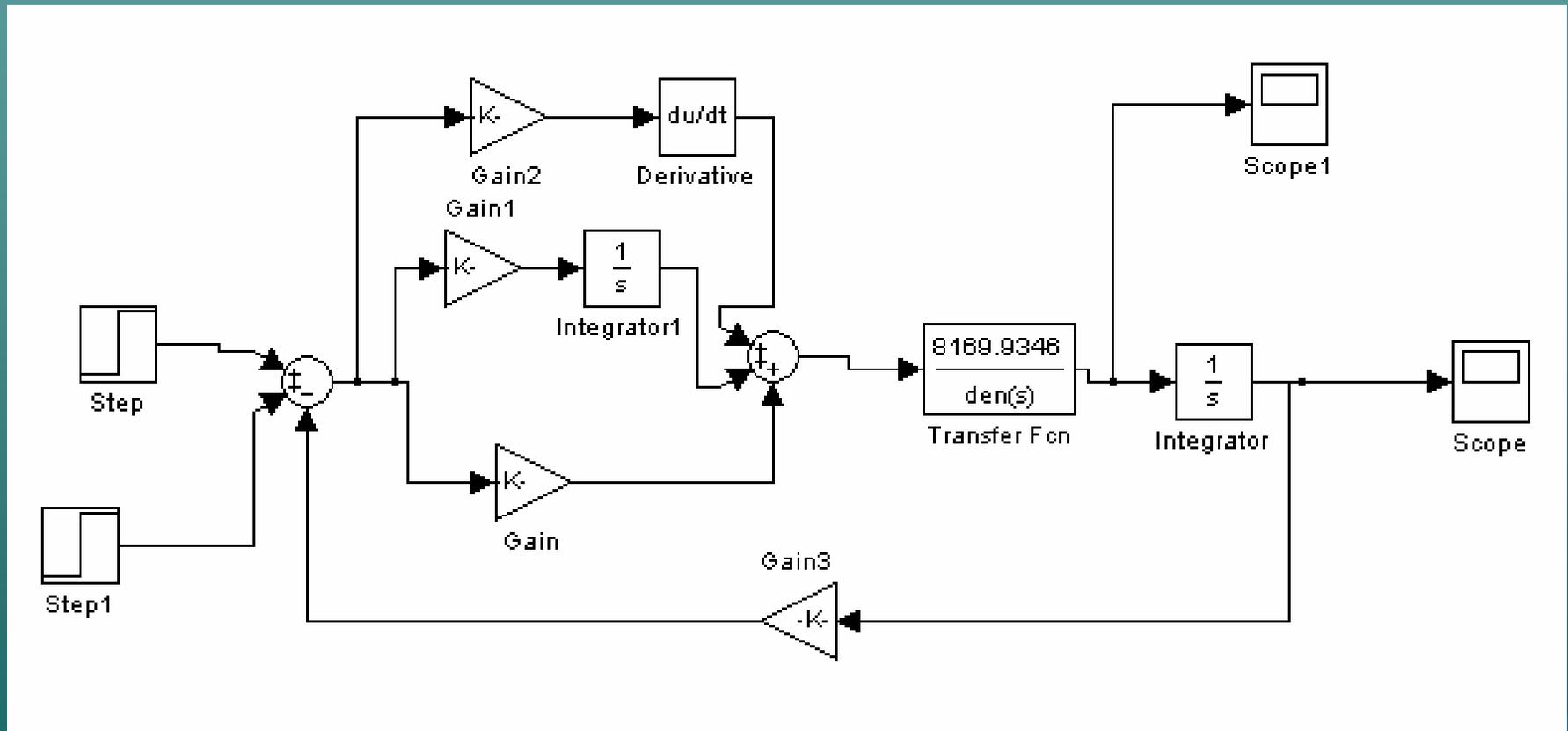
Diagrama eléctrico de motores.

Modelo Matemático

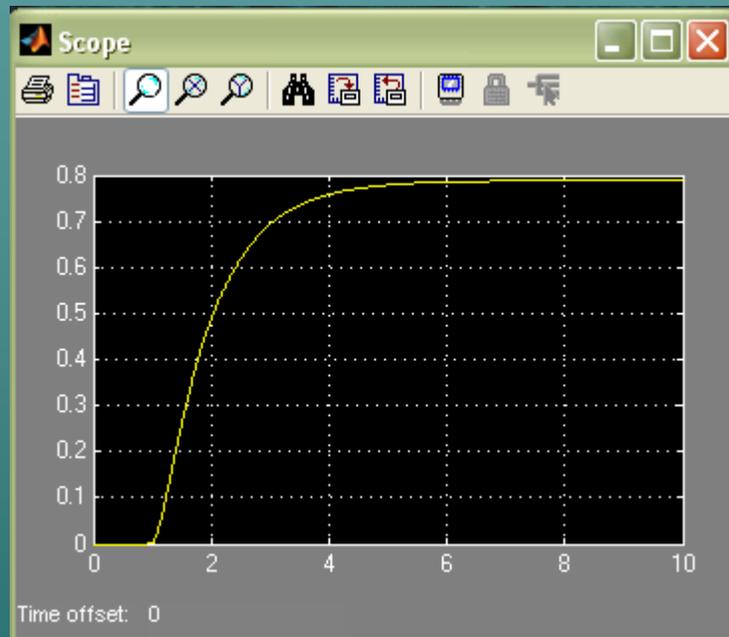
$$\begin{bmatrix} \dot{V}_C \\ \dot{I}_2 \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} & -\frac{1}{C} & 0 & 0 \\ \frac{1}{L} & -\frac{R_i}{L} & 0 & -\frac{k_b}{L} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{k_p}{J+mr^2} & 0 & -\frac{B+br^2}{J+mr^2} \end{bmatrix} \begin{bmatrix} V_C \\ I_2 \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{RC} \\ 0 \\ 0 \\ 0 \end{bmatrix} v(t)$$

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$

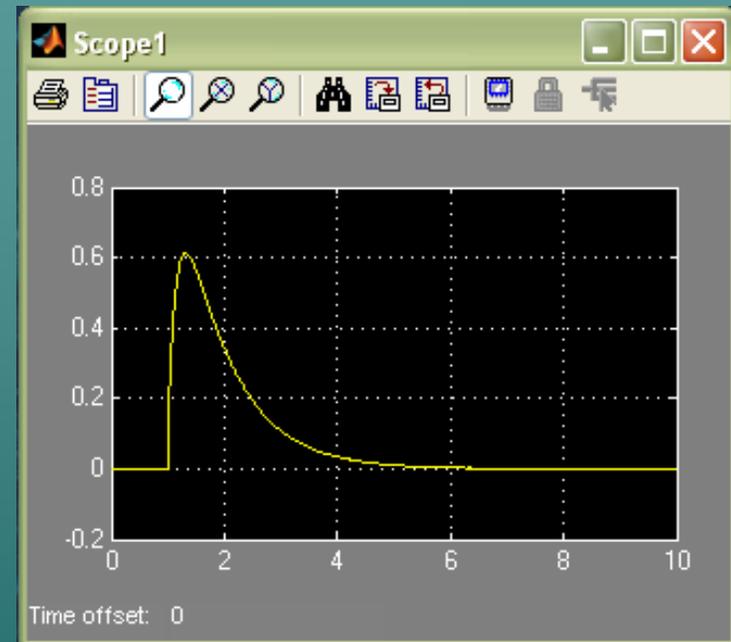
Control PID



Simulación



Posición
de motor



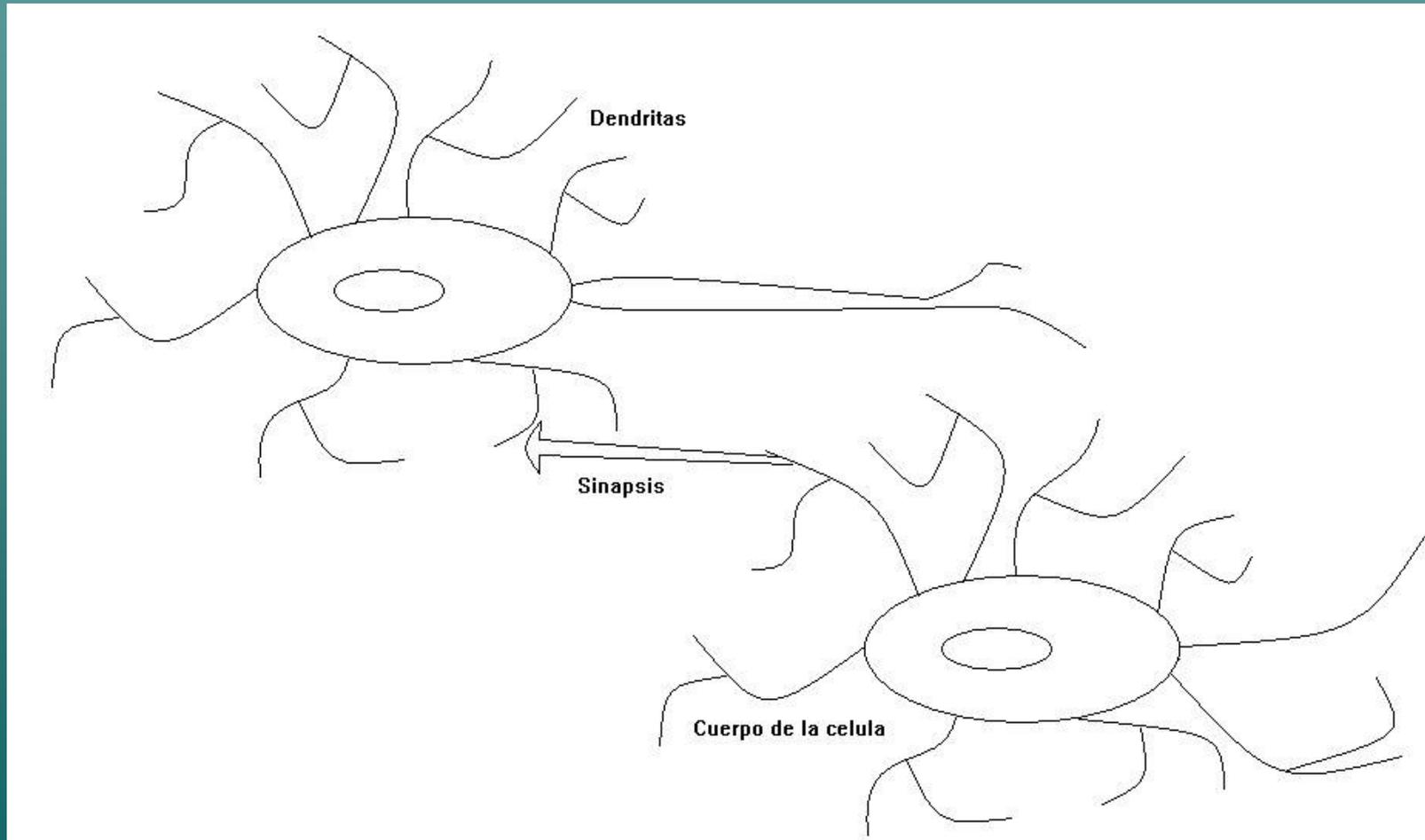
Velocidad
de motor

Redes Neuronales (RN)

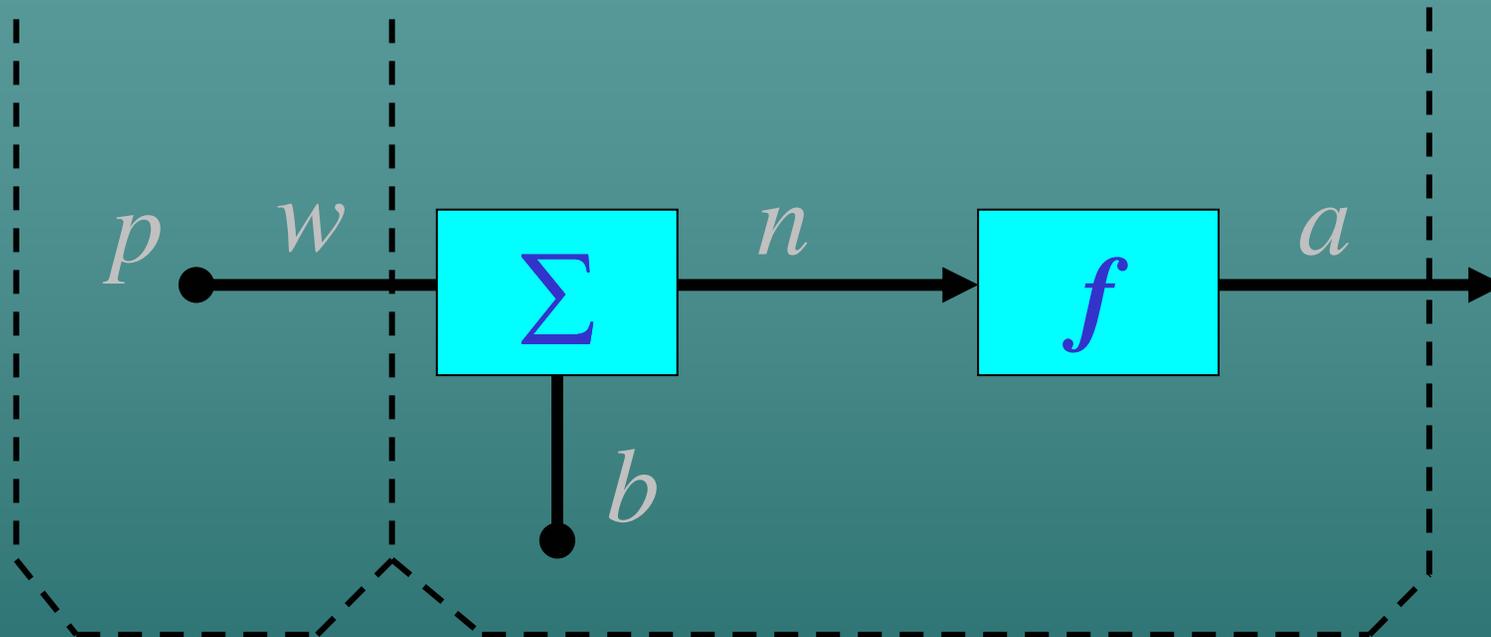
Abstracción extremadamente simple de neuronas biológicas.

Este tipo de neuronas no tienen ni una fracción del poder de las neuronas humanas pero pueden entrenarse para realizar funciones útiles.

Neuronas Biológicas



Neurona Artificial

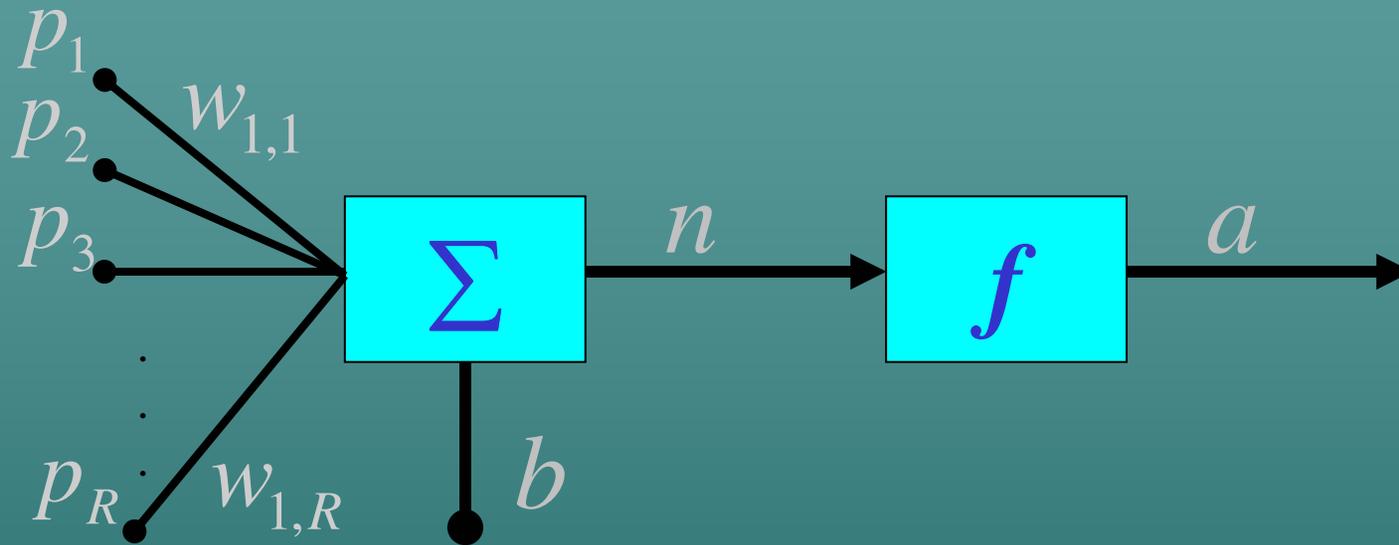


Entradas

Neurona General

$$a = f(wp + b)$$

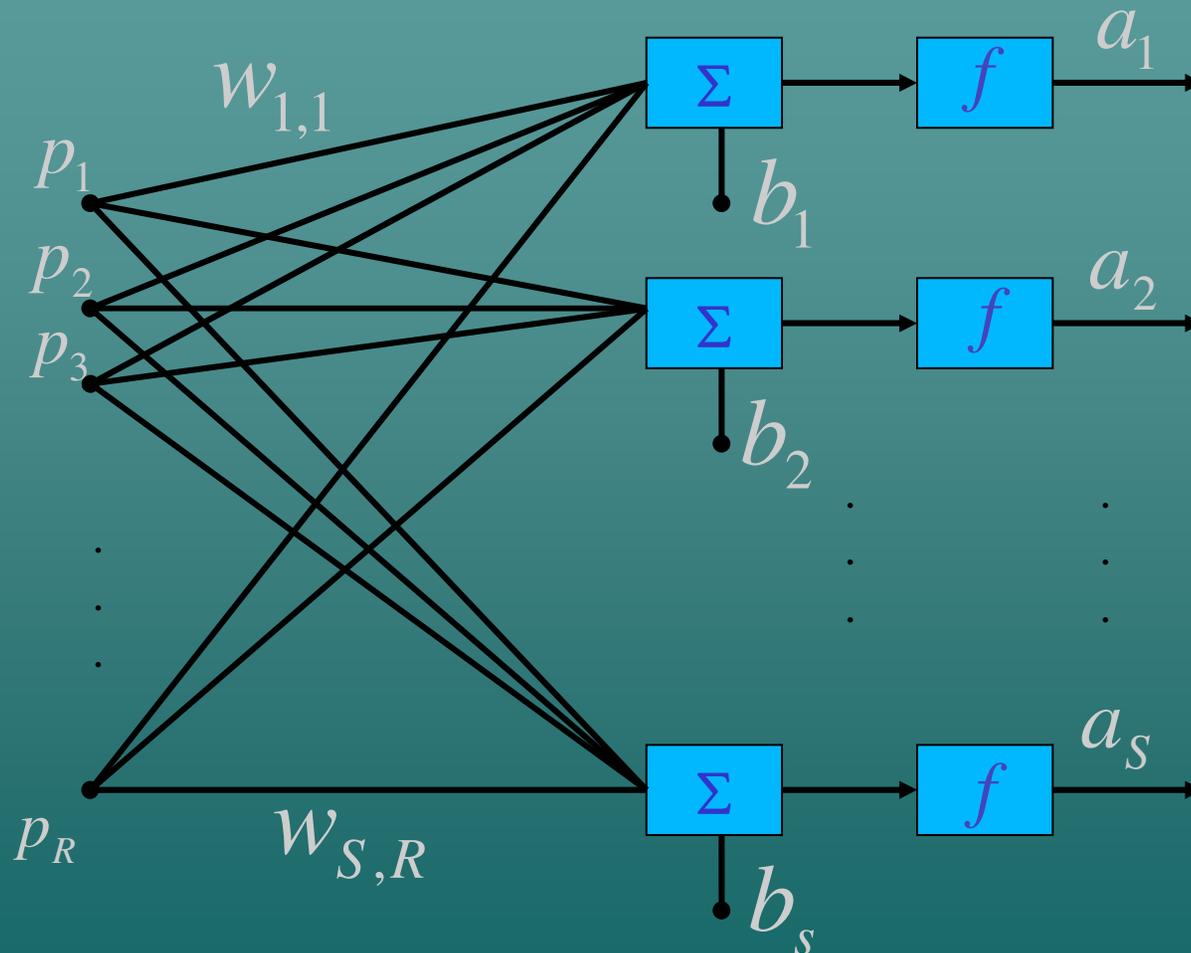
Neurona Multi-entrada



$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

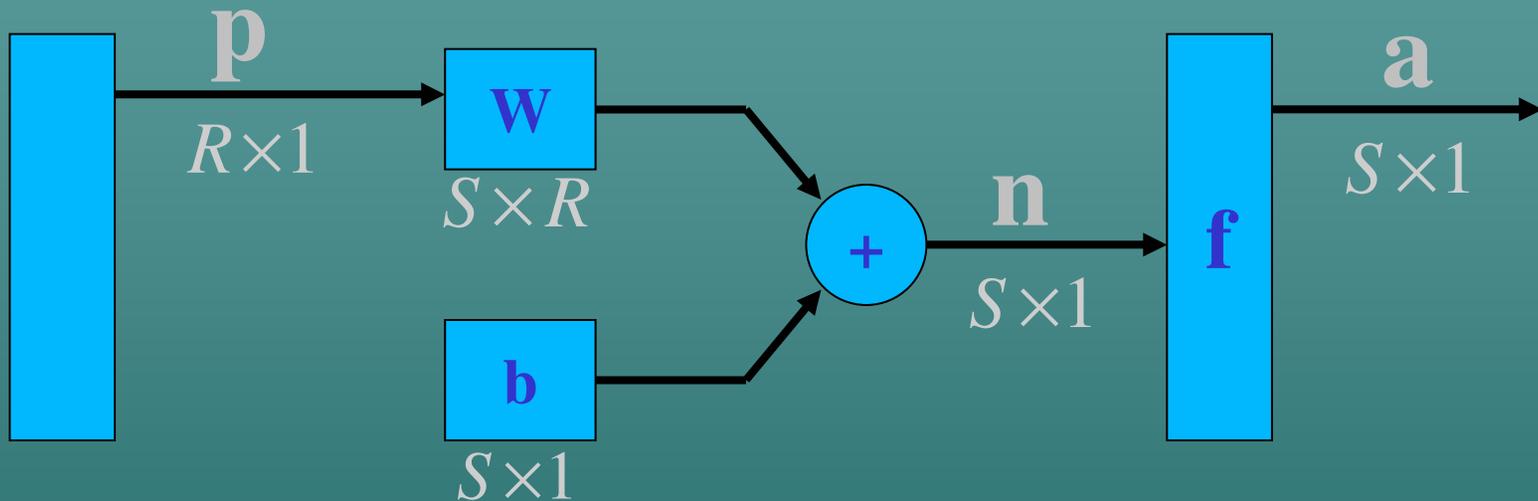
$$a = f(\mathbf{W}\mathbf{p} + b)$$

Multi-Neurona



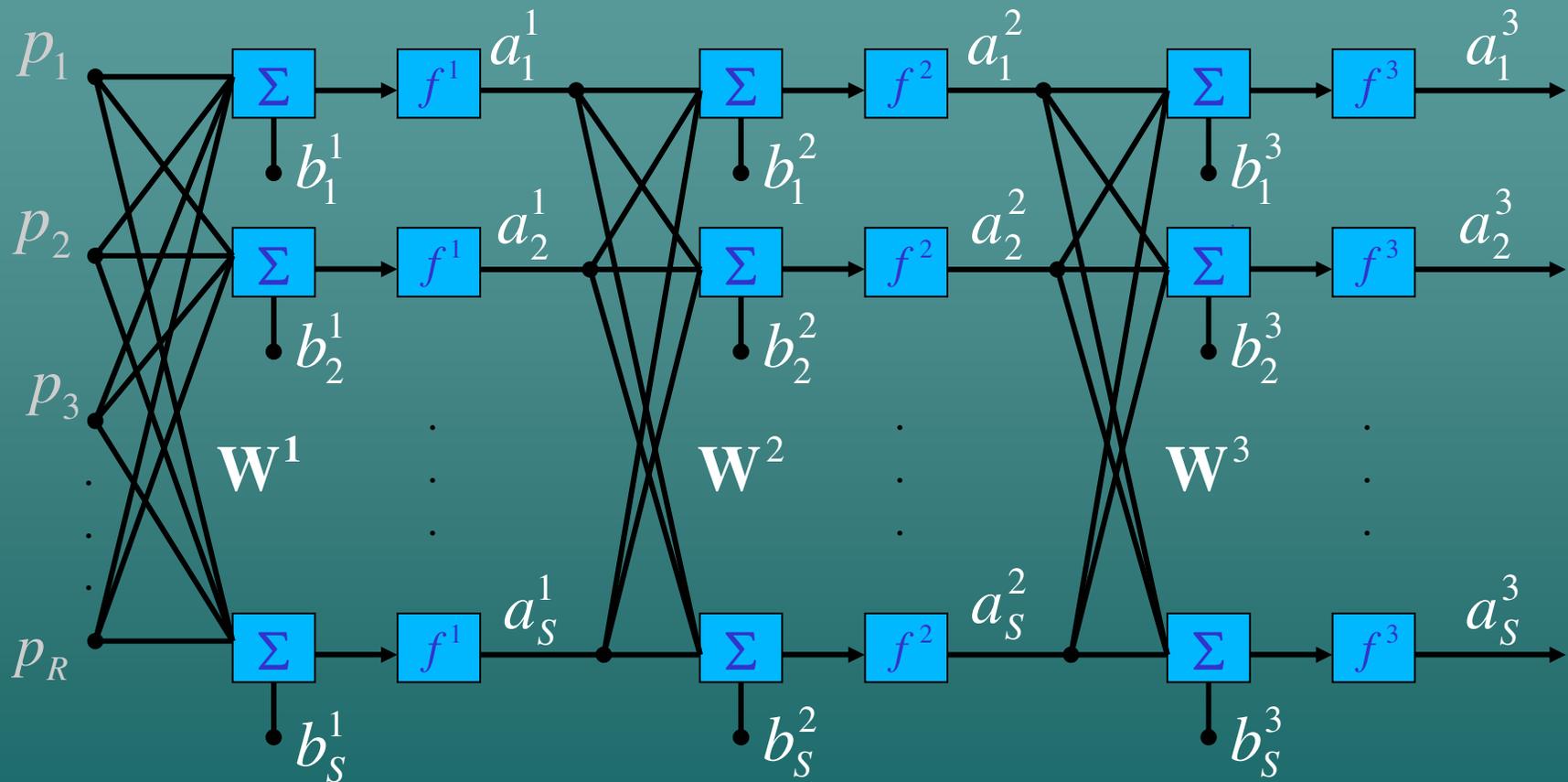
$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

Notación Abreviada



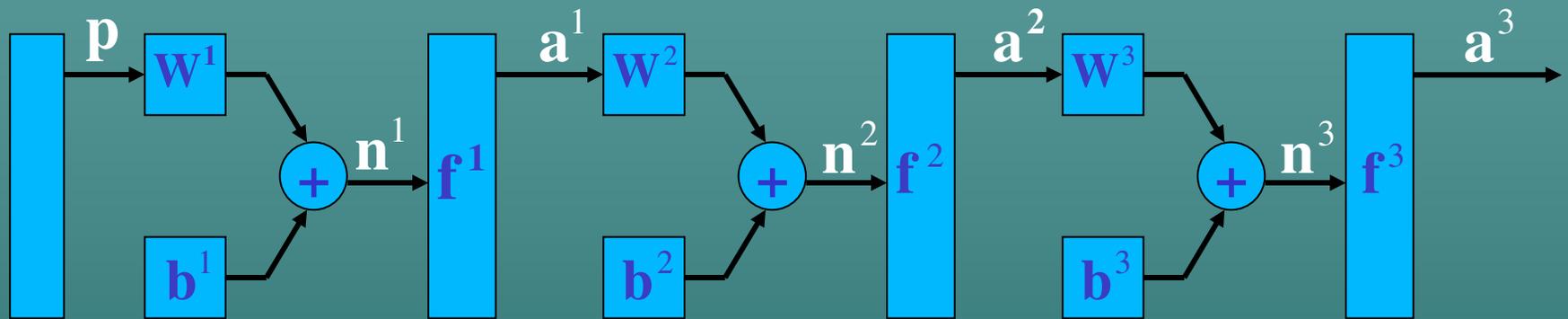
$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

Red Multi-Capa



$$\mathbf{a}^3 = \mathbf{f}^3 (\mathbf{W}^3 \mathbf{f}^2 (\mathbf{W}^2 \mathbf{f}^1 (\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Notación Abreviada



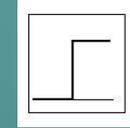
$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3)$$

Funciones de Transferencia

Hard Limit

(hardlim)

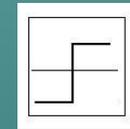
$$f(n) = \begin{cases} n < 0 & a = 0 \\ n \geq 0 & a = 1 \end{cases}$$



Symmetrical Hard Limit

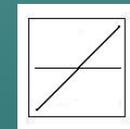
(hardlims)

$$f(n) = \begin{cases} n < 0 & a = -1 \\ n \geq 0 & a = +1 \end{cases}$$



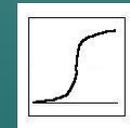
Linear (purelin)

$$f(n) = a$$



Log-Sigmoid (logsig)

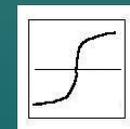
$$f(n) = \frac{1}{1 + e^{-n}}$$



Hyperbolic Tangent Sigmoid

(tansig)

$$f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$



Ejemplo de RN

Dada una neurona de dos entrada con los siguientes parametros: $b = 1.2$, $\mathbf{W} = \begin{bmatrix} 3 & 2 \end{bmatrix}$ y $\mathbf{p} = \begin{bmatrix} -5 & 6 \end{bmatrix}^T$, calcular la salida de la neurona con las siguientes funciones de transferencia:

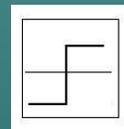
- Symmetrical hard limit
- Linear
- Hyperbolic tangent sigmoig

Ejemplo RN

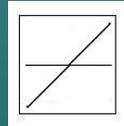
Primero se calcula la salida de la red n :

$$n = \mathbf{Wp} + b = \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} -5 \\ 6 \end{bmatrix} + 1.2 = -1.8$$

- $a = \text{hardlims}(-1.8) = -1$



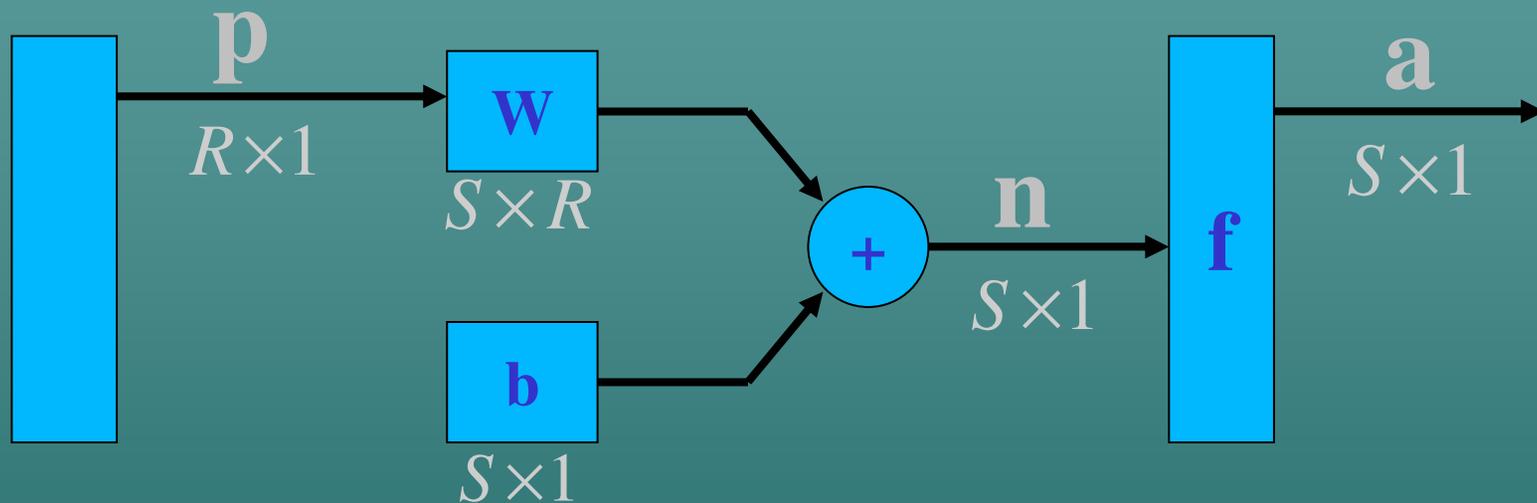
- $a = \text{purelin}(-1.8) = -1.8$



- $a = \text{tansig}(-1.8) = -0.9468$



Perceptron



$$\mathbf{a} = \text{hardlims}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

Ejemplo de RN

Una banda transporta fruta las cuales pasan por tres sensores que detectan la forma, la textura y el peso. El sensor de forma dará un 1 si tiene forma redonda y -1 si es elíptica. El sensor de la textura dará 1 si tiene superficie lisa y -1 si es rugosa. El sensor de peso dará 1 si pesa mas de 1lb y -1 si pesa menos de 1lb. Asíumase que solo existen dos tipos de frutas, naranjas y manzanas.

Ejemplo de RN

Prototipo de las naranjas $\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \mathbf{t}_1 = [0]$

$\mathbf{p} = \begin{bmatrix} \text{forma} \\ \text{textura} \\ \text{peso} \end{bmatrix}$

Prototipo de las manzanas $\mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_2 = [1]$

Entrenamiento de Perceptrones

$$\mathbf{W}^{nuevo} = \mathbf{W}^{anterior} + \mathbf{e}\mathbf{p}^T$$

$$\mathbf{b}^{nuevo} = \mathbf{b}^{anterior} + \mathbf{e}$$

$$\mathbf{e} = \mathbf{t} - \mathbf{a}$$

W: Matriz de pesos **b**: Vector de “bias”

p: Vector de entradas **t**: Vector de patrones

e: Vector de error **a**: Vector de salidas

Entrenamiento de RN

The image features a solid teal background. At the top, the text "Entrenamiento de RN" is written in a white, serif font. At the bottom, there is a silhouette of a mountain range in a lighter shade of teal.

Preliminares Matematicos

Un vector w se denomina *combinación lineal* de los vectores v_1, \dots, v_n si se puede expresar en la forma

$$w = c_1 v_1 + \dots + c_n v_n$$

donde c_1, \dots, c_n son escalares.

Preliminares Matematicos

Sean $\mathbf{v}_1, \dots, \mathbf{v}_n$ n vectores en un espacio V .
Entonces se dice que los vectores son
linealmente dependientes si existen n escalares
 c_1, \dots, c_n no todos cero, tales que

$$c_1 \mathbf{v}_1 + \dots + c_n \mathbf{v}_n = \mathbf{0}$$

Si los vectores no son linealmente dependientes,
entonces se dice que son *linealmente
independientes* (l.i.).

Ejemplos

Los vectores $\mathbf{v}_1 = \begin{bmatrix} 2 \\ -1 \\ 0 \\ 3 \end{bmatrix}$ y $\mathbf{v}_2 = \begin{bmatrix} -6 \\ 3 \\ 0 \\ -9 \end{bmatrix}$ son

linealmente dependientes ya que

$$3\mathbf{v}_1 + \mathbf{v}_2 = \mathbf{0}, \quad c_1 = 3 \neq 0, \quad c_2 = 1 \neq 0$$

Ejemplos

Los vectores $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$ y $\mathbf{v}_2 = \begin{bmatrix} 2 \\ 5 \\ -3 \end{bmatrix}$ son linealmente independientes ya que

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 = \mathbf{0}$$

si y solo si $c_1 = c_2 = 0$.

Preliminares Matematicos

Sean $\mathbf{v}_1, \dots, \mathbf{v}_n$ n vectores en un espacio vectorial V . El *espacio generado* por $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ es el conjunto de combinaciones lineales de $\mathbf{v}_1, \dots, \mathbf{v}_n$.

Preliminares Matematicos

Un conjunto de vectores $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ forma una *base* para V si

- $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ es linealmente independiente.
- $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ genera V .

Preliminares Matematicos

Si \mathbf{A} es una matriz $n \times n$, entonces un vector \mathbf{x} diferente de cero en R^n se denomina *eigenvector* de \mathbf{A} si \mathbf{Ax} es un múltiplo escalar de \mathbf{x} ; es decir,

$$\mathbf{Ax} = \lambda \mathbf{x}$$

para algún escalar λ . El escalar λ se denomina *eigenvalor* de \mathbf{A} , y se dice que \mathbf{x} es un eigen vector de \mathbf{A} correspondiente a λ .

Calculo de eigen-valores, vectores

- Hallar $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$.
- Hallar las raices $\lambda_1, \dots, \lambda_n$ de $p(\lambda) = 0$.
- Resolver el sistema homogeneo $(\mathbf{A} - \lambda_i\mathbf{I})\mathbf{v} = \mathbf{0}$ correspondiente a cada valor característico λ_i .

Ejemplo

Sea $\mathbf{A} = \begin{bmatrix} 4 & 2 \\ 3 & 3 \end{bmatrix}$, entonces:

$$\mathbf{A} - \lambda \mathbf{I} = \begin{bmatrix} 4 - \lambda & 2 \\ 3 & 3 - \lambda \end{bmatrix}$$

$$\det(\mathbf{A} - \lambda \mathbf{I}) = (4 - \lambda)(3 - \lambda) - 6 = 0$$

$$\det(\mathbf{A} - \lambda \mathbf{I}) = \lambda^2 - 7\lambda + 6 = 0$$

Ejemplo

Las raíces son $\lambda_1 = 1$ y $\lambda_2 = 6$.

Al sustituir los valores en los sistemas homogéneos se obtiene

$$\begin{bmatrix} 3 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad y \quad \begin{bmatrix} -2 & 2 \\ 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Ejemplo

Resolviendo los sistemas homogéneos se obtienen los siguientes eigenvectores

$$\mathbf{e}_1 = \begin{bmatrix} 2 \\ -3 \end{bmatrix} \quad \mathbf{e}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

MBA [2], [3], [4]

Sea $I = \{I_1, \dots, I_m\} \subset R^{n \times r}$ imágenes de un objeto tomadas en torno a él.

Los renglones de cada I_j son empilados en un vector columna i_j , y se obtiene el conjunto $i = \{i_1, \dots, i_m\} \subset R^{n \cdot r}$, $n \cdot r$ es el número de píxeles.

Para lograr invarianza en la escala, las imágenes son normalizadas:

$$\hat{i}_j = \frac{i_j}{\|i_j\|}$$

MBA

Se obtiene el conjunto de imágenes normalizadas:

$$\hat{i} = \{\hat{i}_1, \dots, \hat{i}_m\}$$

Vector $C \in R^{n \cdot r}$ de promedios de todas las imágenes.

C_h elemento de C es el promedio aritmético de los h -ésimos valores (píxeles) de cada imagen.

$$C_h = \frac{1}{m} \sum_{j=1}^m i_{hj} \quad 1 \leq h \leq n \cdot r$$

MBA

Se construye la matriz P para calcular el eigenespacio universal:

$$P = [\hat{i}_1 - C \cdots \hat{i}_m - C] \in R^{(n \cdot r) \times m}$$

Con P se asegura calcular el eigenvector con el mayor eigenvalor que representa la dimensión del subespacio.

Matriz de covarianza:

$$Q = P \cdot P^T \in R^{(n \cdot r) \times (n \cdot r)}$$

MBA

Calculo de valores y vectores propios de Q con:

$$\lambda_k e_k = Q e_k$$

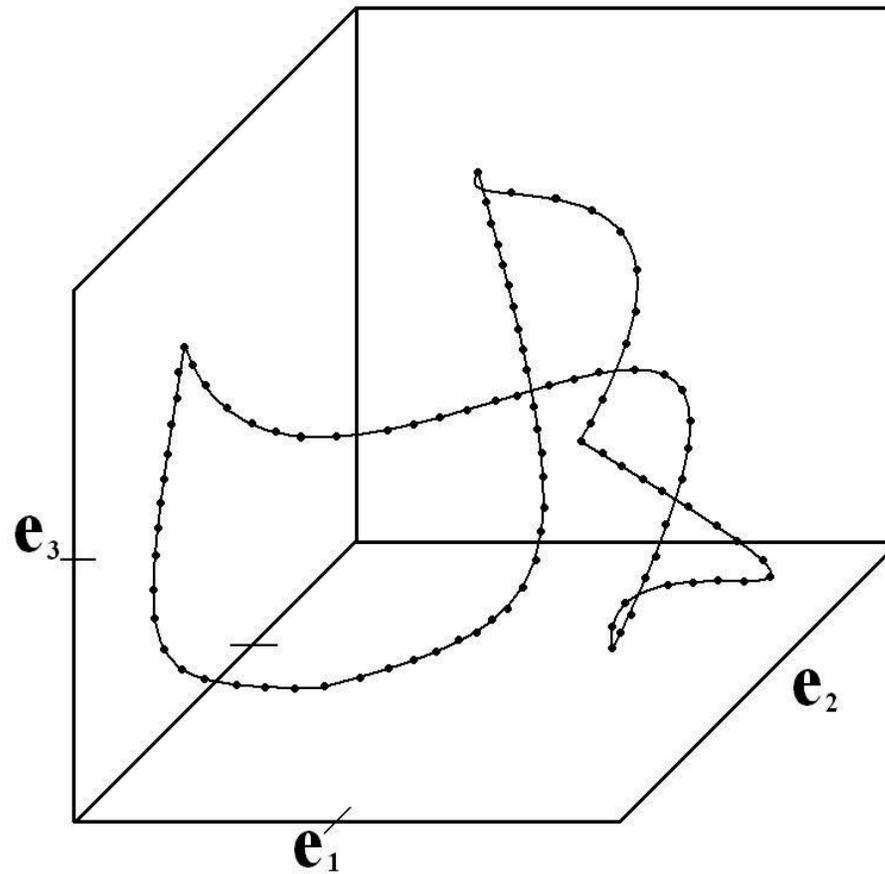
$\{\lambda_1, \dots, \lambda_k\}$ conjunto de eigenvalores,

$\{e_1, \dots, e_k\}$ conjunto de eigenvectores.

Donde $\{\lambda_1 \geq \dots \geq \lambda_k\}$ y $k \leq n \cdot r$

Cada imagen muestra \hat{i}_j (punto discreto) es representada en el eigenespacio.

Eigenespacio



MBA

*La representación de la imagen
(punto fijo) es:*

$$f_j = [e_1 \cdots e_k]^T \cdot (\hat{i}_j - C)$$

Los puntos discretos se interpolan para obtener una trayectoria, posible, que representa la apariencia del objeto, para las posiciones de la cámara alrededor del objeto.

La interpolación es realizada por medio de Splines.

Reconocimiento de Objetos

Se obtiene una nueva imagen f_c tomada en cualquier posición.

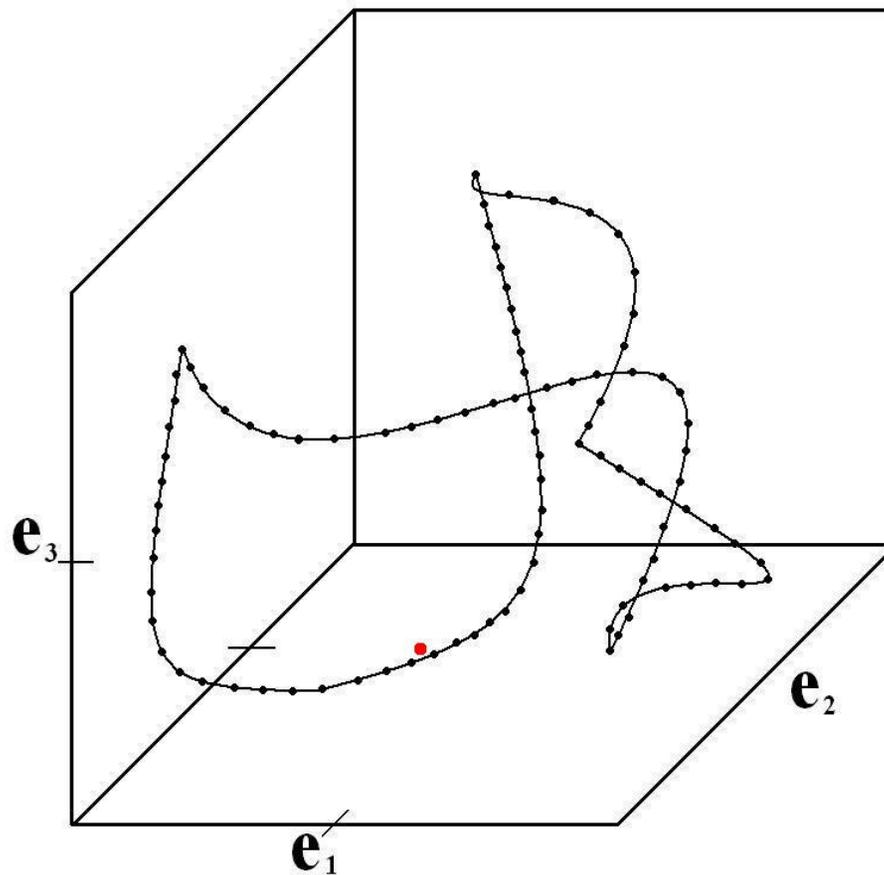
Se representa en el espacio con el punto:

$$f_c = [e_1 \cdots e_k]^T \cdot (\hat{i}_c - C)$$

El reconocimiento se reduce a localizar la trayectoria q que este mas cerca al punto

$$d = \min_q \| f_c - f(q) \|$$

Reconocimiento de Objetos



Ventajas y Desventajas

- La dimensión de los espacios es muy grande lo que implica un alto costo computacional.
- A mayor cantidad de imágenes muestra capturadas,
 - mayor será el costo computacional
 - mayor la precisión en el reconocimiento de objetos.

Adquisición Adaptativa [1]

Colocar objeto sobre
Mesa giratoria

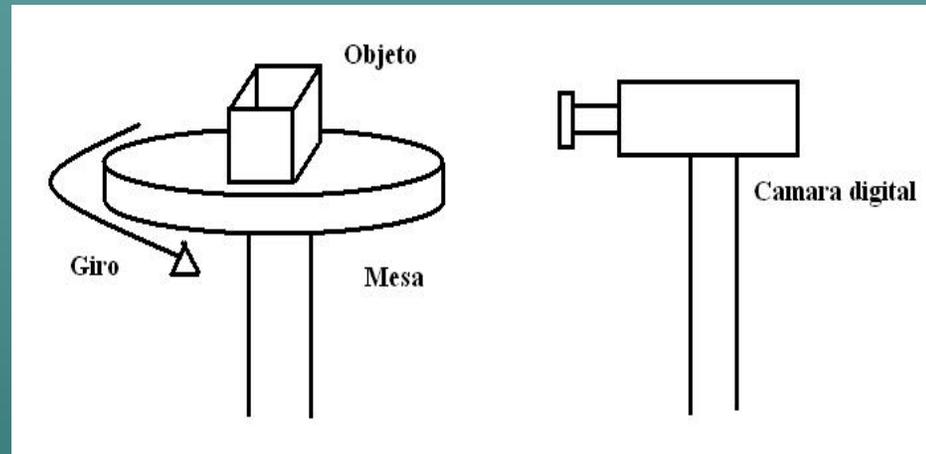
Girar la mesa un
Ángulo "x"

Realizar
toma digital del
objeto

Construcción
de imágenes
(Interpolación)

Guardar imágenes

Verificar precisión



Algoritmo Adaptativo

- 1) Asignar precisión deseada ε
- 2) $L_i=0$
- 3) $L_s=360$
- 4) Adquirir imágenes para las posiciones L_i y L_s
- 5) Si $(L_s-L_i)<2\theta$ hacer $Error=1$ e ir al paso 16
- 6) $L_m=(L_s+L_i)/2$
- 7) Adquirir la imagen correspondiente a la posición L_m

Algoritmo Adaptativo

- 8) Interpolar linealmente entre las imágenes L_i y L_s usando $g(\lambda) = \lambda X + (1 - \lambda)Y$, $0 \leq \lambda \leq 1$.
- 9) Si es posible interpolar la imagen L_m con las imágenes L_i y L_s dentro del margen de error ϵ con $g(0.5)$ y utilizando SSD como criterio de para comparar L_m y $g(0.5)$ esto es,
 $\|L_m - g(0.5)\|^2 \leq \epsilon$, entonces ir al paso 13.

Algoritmo Adaptativo

- 10) Almacenar en una pila el intervalo (L_m, L_s)
- 11) $L_s = L_m$
- 12) Ir al paso 5
- 13) Guardar las imágenes L_i y L_s como necesarias para construir el modelo del objeto

Algoritmo Adaptativo

- 14) Si la pila no esta vacía, extraer de ella una pareja de valores y asignarla a (L_i, L_m) , e ir al paso 5
- 15) $Error=0$
- 16) Fin

Algoritmo Adaptativo (ejemplo)



Imagen 1 (0°)



Imagen 2 (20°)



Interpolación (10°)



Imagen real (10°)

Referencias

1. Altamirano R., L.C., “*Metodologías para la reducción del número de imágenes requeridas para el reconocimiento de objetos articulados*”, CIC-IPN, Tesis doctoral, 2002.
2. Leonardis A., “*Robust recognition using eigenimages*”, *Computer Vision and Image Understanding*, 78, 2000, pp. 99-118.

Referencias

3. Nayar S., K., Nene S., A., Murase H., “*Subspace methods for robot vision*”, IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, Oct. 1996, pp. 750-758.
4. Murase H., Nayar S., K., “*Visual learning and recognition of 3-D objects from appearance*”, International Journal of Computer Vision, 14, 1995, pp. 5-24.

**GRACIAS POR SU
ATENCIÓN**

farid@computacion.cs.cinvestav.mx