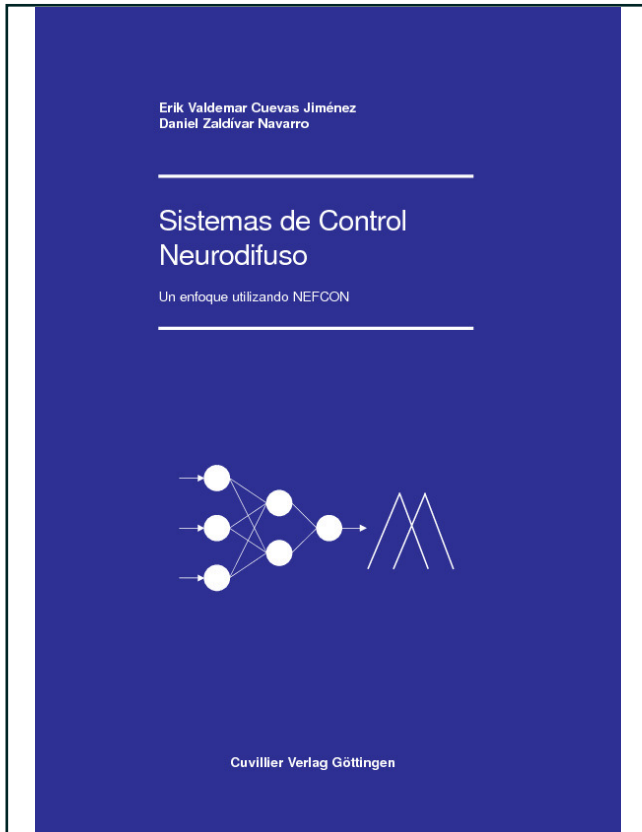




Erik Valdemar Cuevas Jiménez (Autor)
Daniel Zaldívar Navarro (Autor)
Sistemas de Control Neurodifuso
Un enfoque utilizando NEFCON



<https://cuvillier.de/de/shop/publications/2114>

Copyright:

Cuvillier Verlag, Inhaberin Annette Jentsch-Cuvillier, Nonnenstieg 8, 37075 Göttingen, Germany
Telefon: +49 (0)551 54724-0, E-Mail: info@cuvillier.de, Website: <https://cuvillier.de>

eslabones, uno de los cuales está en la base del sistema y su posición está determinada por un actuador, mientras que el segundo eslabón tiene movimiento libre. El Pendubot es parecido al péndulo invertido simple clásico que es utilizado comúnmente como plataforma de comparación en el desempeño de controladores. Las características dinámicas que se establecen con dos eslabones dan como resultado un sistema cuyas propiedades lo hacen más complicado y atractivo como planta a controlar.

El principal objetivo de este libro es el aplicar un esquema de control neurodifuso para controlar sistemas dinámicos atenuando el efecto de las perturbaciones externas que puedan mover al sistema fuera del punto de equilibrio.

Una de las principales aportaciones del libro es demostrar como puede utilizarse exitosamente un método de aprendizaje neurodifuso en la selección de reglas y optimización de funciones de membresía para la construcción de un controlador difuso para el control en tiempo real de sistemas dinámicos.

En este libro se persigue también el objetivo de profundizar en el conocimiento del modelo neurodifuso NEFCON, en particular por las características de interpretabilidad de los controladores generados como sistemas de reglas difusas. También se considera ventajoso conocer a fondo el paradigma, no sólo para estar en condiciones de utilizarlo adecuadamente, sino también para encontrar claves o pistas que en un momento dado permitan mejorar su desempeño.

El presente libro está organizado de la siguiente manera: el Capítulo 2, se presentan los conceptos fundamentales de los sistemas neurodifusos, para lo cual se analizan sus principales características. En dicho estudio se ponen de relieve las razones por las cuales es necesario la combinación de ambos paradigmas. El capítulo termina con la clasificación de los principales enfoques neurodifusos utilizados en inteligencia computacional.

En el Capítulo 3 se describe el modelo NEFCON, se establecen sus fundamentos, sus principales características y se hace una revisión de sus diferentes formas de aprendizaje. En este capítulo también se analiza la herramienta desarrollada para MATLAB/SIMULINK en la implementación de los controladores para sistemas dinámicos.

En el Capítulo 4 se presentan los detalles de la implementación del controlador neurodifuso para la estabilización del Pendubot utilizando el modelo NEFCON, se presentan los datos obtenidos de la simulación, así como los resultados obtenidos en laboratorio tras la implementación en tiempo real del sistema y su comportamiento ante perturbaciones mecánicas.

El Apéndice A contiene una guía para la instalación del sistema NEFCON para MATLAB/SIMULINK. En el Apéndice B se listan los programas creados para MATLAB y para el control en tiempo real del Pendubot. El Apéndice C presenta el desarrollo de las ecuaciones que describen la dinámica del Pendubot, así como las principales características y datos de identificación de este sistema electromecánico.

2. SISTEMAS NEURODIFUSOS

2.1 Redes neuronales

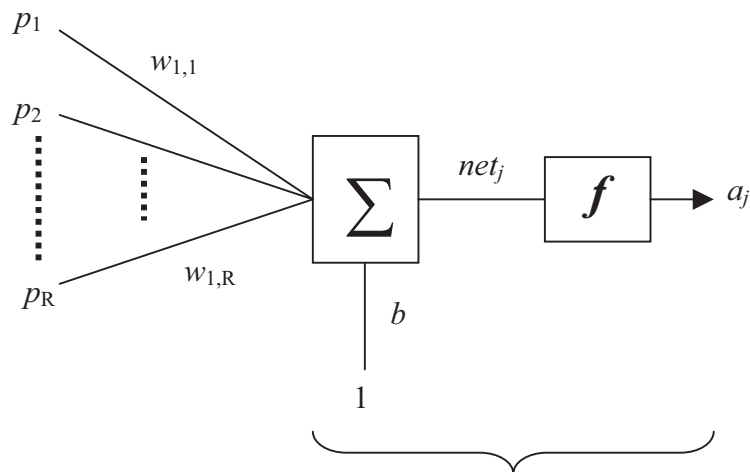
En esta sección se revisan los aspectos formales de las redes neuronales. Además se desarrolla un modelo genérico de red neuronal que es usado en el Capítulo 3.

2.1.1 Principales características de las redes neuronales

Las redes neuronales son modelos que tienden a imitar burdamente el funcionamiento del cerebro. En la Fig. 2.1 se puede apreciar una Red Neuronal Artificial (RNA en lo sucesivo) de una sola neurona. Cada entrada de la red (p) es multiplicada por el peso (w) que tiene la conexión. La suma de esas entradas multiplicadas por sus pesos respectivos constituyen la entrada a la neurona tal que

$$net_j = \sum_i w_{ij} p_i + b_j$$

la cual es pasada por una función de activación $f(net)$. Esta función permite que la salida de la red sea evaluada por funciones escalón, lineales, sigmoidales o gaussianas. El tipo de función utilizada dependerá del tipo de RNA. El elemento de tendencia o "bias" es un valor que se le suma al producto de los pesos y las entradas, esto ocasiona que la función de activación cambie su posición en el eje x; este elemento de tendencia generalmente tiene un valor de entrada de 1.



$$a_j = f\left(\sum_i w_{ij} p_i + b_j\right)$$

Fig. 2.1 Modelo de neurona.

Una RNA se construye a partir de la interconexión de varias neuronas, de tal forma que el número, su forma de interconexión y estructura define un tipo especial de red neuronal.

El aprendizaje de una RNA es simplemente el ajuste de los pesos de las conexiones de la red. Estos pesos simulan la sinapsis o conexiones entre las neuronas del cerebro. Las RNA aprenden de un conjunto de datos existentes sobre un proceso, para ello sólo es necesario entrenar a la red con los datos obtenidos de dicho proceso. Al modificar los pesos entre las conexiones modifican la salida respecto a la entrada.

Algunas ventajas de las RNA son:

- Aprenden de ejemplos.
- Procesan información en paralelo.
- Pueden representar relaciones no lineales.
- Son tolerantes a fallas.
- No necesitan un modelo matemático.
- Capacidad de generalización.
- Son relativamente fáciles de implementar.

Una desventaja de las RNA es que no se identifica claramente el mecanismo con el cual han resuelto un problema, es decir, una red neuronal se puede ver como una caja negra la cual tiene sus entradas y sus salidas, pero no se sabe del todo como trabaja lo que está dentro de la caja, se sabe que hay un algoritmo de entrenamiento que describe cómo se ajustan los pesos de la red, pero no como aprende. Otra desventaja es que no existe una metodología que indique que tipo de red debemos utilizar para resolver un problema específico, ni cuantas neuronas debemos utilizar en la capa escondida o si se tiene que utilizar más de una capa escondida.

Las RNA tienen muchas aplicaciones, tales como el reconocimiento de patrones (ya sea caracteres escritos a mano, firmas, voz e imágenes), compresión de datos, filtrado de señales, predicción del clima, estimación de variables, modelado y control de procesos.

Como se mencionó anteriormente las RNA aprenden mediante ejemplos aplicando para ello un algoritmo que hace posible la modificación de las conexiones en función del desempeño de la red. A este algoritmo se le llama algoritmo de entrenamiento. Por lo tanto el tipo de entrenamiento está en función de las características principales que manifiesta el algoritmo que será aplicado.

Los tipos de entrenamiento de las RNA se logran por medio de tres métodos fundamentalmente: