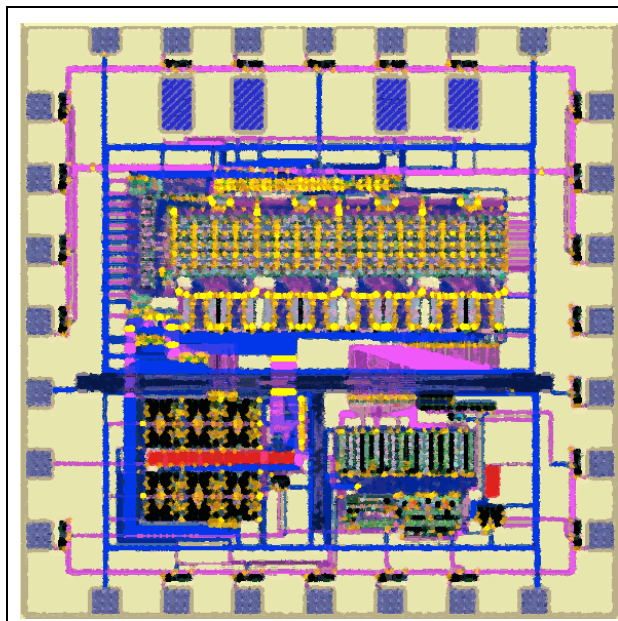


**DISEÑO Y APLICACIÓN DE CONTROLADORES  
NEURO-DIFUSOS INTEGRADOS EN MODO MIXTO  
DE ALTA COMPLEJIDAD**

**TESIS DOCTORAL**

**Rafael de Jesús Navas González**



*Departamento de Electrónica  
Universidad de Málaga*



**DISEÑO Y APLICACIÓN DE CONTROLADORES  
NEURO-DIFUSOS INTEGRADOS EN MODO MIXTO  
DE ALTA COMPLEJIDAD**

**TESIS DOCTORAL**

**Rafael de Jesús Navas González**

*Departamento de Electrónica  
Universidad de Málaga  
Málaga, Diciembre 2000*



**DISEÑO Y APLICACIÓN DE CONTROLADORES  
NEURO-DIFUSOS INTEGRADOS EN MODO MIXTO  
DE ALTA COMPLEJIDAD**

**TESIS DOCTORAL**

**Rafael de Jesús Navas González**

**Dirigida por Fernando Vidal Verdú**

*Departamento de Electrónica  
Universidad de Málaga  
Málaga, Diciembre 2000*

Agradecimientos:

A Fernando, por animarme y ofrecerme la oportunidad de seguir sus pasos, por su mimo y estrecha colaboración en la dirección de esta tesis, y sobre todo por su amistad tras las muchas horas de despacho compartido.

A Ángel Rodríguez Vázquez, por su inestimable apoyo y por seguir manteniendo los lazos que nos unen a pesar de la distancia.

Y a todos mis compañeros del departamento de Electrónica.

*A mis padres  
que todo lo dan  
y todo lo merecen.*

*A María Luisa  
y  
Carmen  
que con sus risas y llantos  
han contribuido  
a hacer inolvidables  
estos días.*

*Y a tí, Luisa,  
por tu fortaleza,  
tu paciencia y comprensión  
en este año,  
y por todos los que hemos  
de estar juntos, gracias.*

---

## Índice

---

|                                                                                                                       |           |
|-----------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Introducción</b>                                                                                                   | <b>1</b>  |
| <b>Capítulo 1: Aspectos Básicos de los Sistemas Difusos y Neuro-Difusos y su Implementación</b>                       | <b>9</b>  |
| 1. INTRODUCCIÓN .....                                                                                                 | 9         |
| 2. SISTEMAS DIFUSOS .....                                                                                             | 10        |
| 3. SISTEMAS NEURO-DIFUSOS .....                                                                                       | 14        |
| 4. DISEÑO DE CONTROLADORES DIFUSOS Y NEURO-DIFUSOS .....                                                              | 17        |
| 4.1. Aspectos generales de diseño.....                                                                                | 19        |
| 4.2. Partición del espacio de entrada .....                                                                           | 20        |
| 4.3. Sistema de inferencia Takagi-Sugeno con <i>singleton</i> en el consecuente .....                                 | 22        |
| 4.4. Arquitectura ANFIS .....                                                                                         | 26        |
| 4.5. Arquitecturas para controladores neuro-difusos basadas en ANFIS .....                                            | 28        |
| 4.5.1. <i>Arquitectura general</i> .....                                                                              | 28        |
| 4.5.2. <i>Arquitectura para particiones en rejilla del espacio de entrada</i> .....                                   | 30        |
| 5. TEORÍA DE LA APROXIMACIÓN EN LOS SISTEMAS DIFUSOS. PRECISIÓN DE LA APROXIMACIÓN Y PROPIEDAD DE DESCOMPOSICIÓN..... | 32        |
| 5.1. Teoría de la aproximación en los sistemas difusos .....                                                          | 33        |
| 5.2. Propiedades de aproximación de los sistemas difusos .....                                                        | 35        |
| 5.3. Propiedad de descomposición de los sistemas difusos .....                                                        | 41        |
| 5.4. Conclusiones .....                                                                                               | 43        |
| 6. REALIZACIONES DE CONTROLADORES DIFUSOS Y NEURO-DIFUSOS MONOLÍTICOS MEDIANTE HARDWARE ANALÓGICO Y MIXTO .....       | 44        |
| 6.1. Aspectos generales .....                                                                                         | 44        |
| 6.2. Controladores difusos y neuro-difusos monolíticos analógicos y mixtos .....                                      | 47        |
| <b>Capítulo 2: Propuesta de una Arquitectura Multiplexada para Controladores Difusos en Modo Mixto</b>                | <b>55</b> |
| 1. INTRODUCCIÓN .....                                                                                                 | 55        |
| 2. ARQUITECTURA MULTIPLEXADA .....                                                                                    | 57        |
| 2.1. Ejemplo bidimensional .....                                                                                      | 57        |
| 2.2. Caso multidimensional .....                                                                                      | 61        |
| 3. PROPUESTA DE ARQUITECTURA PARA SISTEMAS DIFUSOS CON NÚCLEO ANALÓGICO MULTIPLEXADO .....                            | 65        |
| 4. CAPACIDAD DE ADAPTACION DE LA ARQUITECTURA PARA SISTEMAS DIFUSOS CON NÚCLEO ANALÓGICO MULTIPLEXADO .....           | 69        |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 4.1. Programación directa de los valores <i>singleton</i> ..... | 70 |
| 4.2. Aprendizaje de los valores <i>singleton</i> .....          | 71 |

### Capítulo 3: Descripción Funcional y Diseño CMOS del Núcleo Analógico

75

|                                                                  |    |
|------------------------------------------------------------------|----|
| 1. INTRODUCCIÓN .....                                            | 75 |
| 2. NÚCLEO ANALÓGICO .....                                        | 75 |
| 2.1. Discusión de los requerimientos y objetivos de diseño ..... | 76 |
| 2.2. Propuesta de diseño .....                                   | 76 |
| 2.3. Diseño CMOS del Núcleo Analógico .....                      | 81 |
| 2.3.1. Diseño del Bloque de Entrada .....                        | 81 |
| 2.3.2. Diseño del Bloque de Regla .....                          | 90 |
| 2.3.3. Consideraciones globales de diseño .....                  | 99 |

### Capítulo 4: Descripción Funcional y Diseño CMOS de los Bloques de Multiplexado (I): Selector de Intervalo

109

|                                                                  |     |
|------------------------------------------------------------------|-----|
| 1. INTRODUCCIÓN .....                                            | 109 |
| 2. SELECTOR DE INTERVALO .....                                   | 111 |
| 2.1. Discusión de los requerimientos y objetivos de diseño ..... | 111 |
| 2.2. Propuesta de diseño .....                                   | 112 |
| 2.3. Diseño CMOS del bloque Selector de Intervalo .....          | 115 |
| 2.3.1. Array resistivo .....                                     | 115 |
| 2.3.2. Comparador analógico con histéresis .....                 | 119 |
| 2.3.3. Codificador Gray .....                                    | 138 |
| 2.3.4. Consideraciones globales de diseño .....                  | 140 |

### Capítulo 5: Descripción Funcional y Diseño CMOS de los Bloques de Multiplexado (II): Programador de Antecedente de Reglas y Programador de Consecuente de Reglas

145

|                                                                        |     |
|------------------------------------------------------------------------|-----|
| 1. INTRODUCCIÓN .....                                                  | 145 |
| 2. PROGRAMADOR DE ANTECEDENTE DE REGLAS .....                          | 145 |
| 2.1. Discusión de los requerimientos y objetivos de diseño .....       | 145 |
| 2.2. Propuesta de diseño .....                                         | 145 |
| 2.3. Diseño CMOS del bloque Programador de Antecedente de Reglas ..... | 146 |
| 2.3.1. Multiplexor analógico con control digital .....                 | 146 |
| 2.3.2. Control del multiplexor analógico .....                         | 150 |
| 2.3.3. Consideraciones globales de diseño .....                        | 152 |

|                                                                                                                                           |            |
|-------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 3. PROGRAMADOR DE CONSECUENTE DE REGLAS .....                                                                                             | 155        |
| 3.1. Discusión de los requerimientos y objetivos de diseño .....                                                                          | 155        |
| 3.1.1. <i>Discusión de los requerimientos de la arquitectura</i> .....                                                                    | 156        |
| 3.1.2. <i>Discusión de los requerimientos de temporización de los accesos</i> .....                                                       | 161        |
| 3.2. Propuesta de diseño .....                                                                                                            | 164        |
| 3.3. Diseño CMOS del bloque Programador de Consecuente de Reglas .....                                                                    | 168        |
| 3.3.1. <i>Organización general en el controlador ejemplo</i> .....                                                                        | 171        |
| 3.3.2. <i>Diseño del Bloque de Celdas de Memoria</i> .....                                                                                | 175        |
| 3.3.3. <i>Selector de Filas</i> .....                                                                                                     | 182        |
| 3.3.4. <i>Selector de Columnas para accesos externos</i> .....                                                                            | 186        |
| 3.3.5. <i>Selector de Columnas para accesos internos</i> .....                                                                            | 189        |
| 3.3.6. <i>Consideraciones globales de diseño</i> .....                                                                                    | 196        |
| <br><b>Capítulo 6 : Controlador Ejemplo: Integración y Resultados</b> .....                                                               | <b>205</b> |
| 1. INTRODUCCIÓN .....                                                                                                                     | 205        |
| 2. DESCRIPCIÓN DEL CHIP CONTROLADOR MFCON .....                                                                                           | 206        |
| 3. SISTEMA DE TEST .....                                                                                                                  | 211        |
| 3.1. Interfaz de programación y de adquisición de datos .....                                                                             | 211        |
| 3.2. Generación de las polarizaciones externas .....                                                                                      | 213        |
| 3.3. Circuitos de control y adquisición de datos.....                                                                                     | 214        |
| 3.3.1. <i>Familia de conversores digital a analógico de 8 bits</i><br><i>DAC0800/DAC0801/DAC0802</i> .....                                | 214        |
| 3.3.2. <i>Conversor Analógico digital de 8-bits compatible con</i><br><i>microprocesador ADC0803</i> .....                                | 216        |
| 3.4. Programas de pruebas del sistema de test .....                                                                                       | 218        |
| 4. RESULTADOS EXPERIMENTALES .....                                                                                                        | 219        |
| 4.1. Comportamiento dc del chip controlador difuso MFCON .....                                                                            | 219        |
| 4.1.1. <i>Medida de la salida para los diferentes valores singleton</i> .....                                                             | 219        |
| 4.1.2. <i>Ejemplos de Superficies de Control</i> .....                                                                                    | 223        |
| 4.2. Comportamiento dinámico de MFCON .....                                                                                               | 233        |
| <br><b>Capítulo 7: Aplicación a un caso práctico: Arranque suave</b><br><b>                    de motores de corriente continua</b> ..... | <b>235</b> |
| 1. INTRODUCCIÓN .....                                                                                                                     | 235        |
| 2. ELECTRÓNICA DE POTENCIA Y CONTROL DE MOVIMIENTO .....                                                                                  | 237        |
| 3. CONTROL DE VELOCIDAD Y ARRANQUE SUAVE DE MOTORES DE<br>CORRIENTE CONTINUA .....                                                        | 243        |
| 3.1. Problema del arranque .....                                                                                                          | 243        |



|                                                                                                                 |            |
|-----------------------------------------------------------------------------------------------------------------|------------|
| 3.2. Ecuaciones básicas                                                                                         | 244        |
| 3.2.1. Corriente de armadura durante del transitorio de arranque                                                | 249        |
| 3.3. Control de motores de corriente continua                                                                   | 250        |
| 3.3.1. Convertidor DC-DC o Troceador                                                                            | 250        |
| 3.3.2. Control en lazo abierto                                                                                  | 253        |
| 3.3.3. Control en lazo cerrado                                                                                  | 253        |
| 3.4. Estrategias para el arranque suave de motores de corriente continua                                        | 254        |
| 3.4.1. Control en lazo abierto                                                                                  | 254        |
| 3.4.2. Control en lazo cerrado                                                                                  | 257        |
| 3.5. Arranque suave de motores de corriente continua empleando el controlador difuso MFCN en el lazo de control | 258        |
| 3.5.1. Estrategia de control                                                                                    | 258        |
| 3.5.2. Descripción del sistema experimental                                                                     | 261        |
| 4. RESULTADOS DE LOS EXPERIMENTOS DE CONTROL DE ARRANQUE REALIZADOS CON MFCN EN EL BUCLE DE CONTROL             | 268        |
| 4.1. Determinación experimental de algunas relaciones y características del sistema de control                  | 268        |
| 4.2. Resultados experimentales                                                                                  | 271        |
| 4.2.1. Resultados experimentales en arranque directo                                                            | 271        |
| 4.2.2. Arranque suave. Ejemplo I                                                                                | 272        |
| 4.2.3. Arranque suave. Ejemplo II                                                                               | 277        |
| 4.2.4. Resumen de resultados                                                                                    | 280        |
| <b>Conclusiones</b>                                                                                             | <b>281</b> |
| <b>Apéndice</b>                                                                                                 | <b>283</b> |
| <b>Referencias</b>                                                                                              | <b>287</b> |

## Introducción

En el X Congreso Español sobre Tecnologías y Lógica Fuzzy de este año, se planteaba un debate acerca del futuro de la investigación en la materia. Como en el caso de las redes neuronales, la *Lógica Difusa* ha sido objeto de una gran concentración de esfuerzos en los últimos años. Sin embargo, se reconoce una falta de resultados o aportes tan significativos como los de Zadeh en 1965 [Zade65][Klir96], o los de Mamdani [Mamd74], Yamakawa [Yama80] [Yama93], o Watanabe [Wata90], en los terrenos del control y de la implementación *hardware* respectivamente. Aunque el interés en la lógica difusa nace en los años 80 como consecuencia de los éxitos alcanzados en su aplicación industrial, lo cierto es que las expectativas creadas por los sistemas neuronales y difusos han ido más allá de los avances conseguidos, y esto crea un clima de incertidumbre hacia las posibilidades reales de estas tecnologías, y cuestiona si se ha alcanzado ya un techo o queda aún más por aportar. La conclusión es que se debe incidir en resolver problemas reales y de complejidad suficiente para corroborar la vigencia de este interés en las tecnologías fuzzy. Esta Tesis Doctoral trata de avanzar en ese sentido, dando respuesta a la pregunta de cómo podemos aumentar la complejidad de los problemas abordables mediante controladores analógicos, al tiempo que mantenemos sus principales ventajas. Asimismo, los demostradores elegidos para las implementaciones *hardware* deben trabajar en condiciones suficientemente exigentes para probar la solvencia que supone su alta velocidad de operación, y ésto se pretende en esta Tesis con un demostrador que ha de operar en un bucle de control muy rápido, en concreto en el control del transitorio de arranque de un motor eléctrico de corriente continua.

El mencionado auge experimentado por las aplicaciones basadas en la lógica difusa se debe sin duda a sus éxitos prácticos en tareas de control, tanto en procesos industriales como en electrónica de consumo. A este éxito han contribuido en gran medida los desarrollos *hardware*, tanto específico como de propósito general, que han permitido acelerar el cómputo de los algoritmos de inferencia difusa e introducirlos en sistemas y aplicaciones en tiempo real, para los que, en general, se requiere una mejor razón velocidad/consumo de energía que la que proporcionan las implementaciones *software* [Boni95][Cost95][Kand98]. En muchas de esas aplicaciones prácticas de los *controladores difusos* tanto las entradas (representadas por un vector multidimensional  $\mathbf{x} = \{x_1, x_2, \dots, x_M\}^T$ ), como las salidas (representadas por un escalar  $y$ ), son señales analógicas. Este es el caso en aplicaciones de control en las que las entradas se obtienen a partir de medidas de sensores, y las salidas son empleadas para fijar el valor de alguna variable física a través de un transductor de salida o actuador [Mark94].

Hay dos aproximaciones básicas para realizar el *hardware* requerido en esas aplicaciones. Una emplea circuitos analógicos solamente en los interfaces, mientras que el procesamiento se realiza de forma digital, bien por medio de circuitos integrados de propósito

general, bien mediante circuitos de aplicación específica (ASICs) [Wata92][Eich95] [Mase95][Cost95]. Otra opción, utiliza circuitos analógicos para el procesamiento de la señal, mientras que la circuitería digital es empleada para dotar de programabilidad al sistema [Nish93]. En este último caso, el sistema analógico puede trabajar en tiempo discreto, sincronizado mediante una señal de reloj, de manera que el sistema sólo es capaz de responder a los cambios en las entradas en ciertos intervalos de tiempo denominados intervalos activos [Fatt94] [Ugur97] [Bour98]; o bien en tiempo continuo, esto es, de manera asíncrona, de forma que el sistema es capaz de responder a los cambios en la entrada en cualquier instante de tiempo [Guo96] [Batu97] [Fran98] [Rodr99]. Este último es el caso de los sistemas considerados en este trabajo.

En general las implementaciones digitales proporcionan una mayor flexibilidad, y se integran muy bien en entornos de procesamiento digital; permiten abordar problemas de mayor complejidad <sup>1†</sup> y consiguen mayores resoluciones que sus homólogas analógicas. Para una complejidad dada, el ciclo de diseño es mucho más corto para las implementaciones digitales, sobre todo a partir del importante desarrollo de las herramientas CAD de síntesis automática. Por otra parte, las técnicas de diseño “pipe-line” han permitido superar el límite en la velocidad de proceso que impone el retraso de la celda estándar en la tecnología digital usada. Finalmente los diferentes métodos de optimización desarrollados para la representación y almacenamiento de las funciones de pertenencia y base de reglas [Wata90], unidos a la explotación de las propiedades de localidad de los algoritmos difusos, con conceptos como los de regla activa [Ikeda91] [Asci95], han permitido reducir las necesidades de almacenamiento y procesamiento de datos de muchos de los diseños digitales, las cuales suponían una de sus principales fuentes de ocupación de área y consumo de potencia.

Por su parte, los controladores difusos analógicos ofrecen un interfaz directo a sensores y actuadores, lo que hace que el retraso entrada-salida, parámetro realmente importante en control, sea usualmente menor que el de sus homólogos digitales (considerando los convertidores a entrada y salida). En general, los controladores analógicos resultan ser ventajosos en aplicaciones de baja complejidad, que requieran compacidad, alta velocidad de operación y bajo consumo de potencia [Vitt90][Nish93]. Esto último se consigue, por un lado explotando el paralelismo de cómputo intrínseco de los algoritmos difusos, compatible con el propio paralelismo de procesamiento analógico; y por otro, aprovechando la versatilidad de los dispositivos electrónicos cuando se explotan sus diferentes regiones de operación para realizar de manera muy eficiente, y con circuitos muy simples, muchos de los operadores lineales y no lineales involucrados en el cálculo de estos algoritmos.

El inconveniente que supone una mayor duración del ciclo de diseño de los sistemas

---

<sup>1†</sup>. En este contexto la complejidad se mide en relación al número de entradas y número de reglas del sistema, siguiendo la propuesta de [Cost95].

analógicos, se debe principalmente al mayor nivel de complejidad que resulta de la gran variedad de primitivas analógicas, y el mayor grado de detalle en cuanto a elementos y parámetros de diseño que hay que manejar. Estas circunstancias, unidas a un menor desarrollo de las herramientas de diseño automático para este tipo de sistemas, hace que en muchos casos el diseño analógico esté reservado a diseñadores muy expertos. Sin embargo, el ciclo de diseño de un controlador analógico puede verse reducido si se establece una metodología del diseño, fundamentada sobre una arquitectura sencilla y cómoda para su implementación *hardware*, que sea modular, y desarrollada con bloques constructivos bien definidos y escalables que permita cierto grado de automatización [Lema94][Vida95a] [Mana96] [Batu97][Rodr99].

Otro de los inconvenientes de las implementaciones analógicas es su limitada precisión, aunque puede superarse con un modelado apropiado de las fuentes de imprecisión [Pelg89], y el uso de una adecuada estrategia de diseño puede conducir a precisiones de hasta el 1%, las cuales resultan más que suficientes para muchas de las aplicaciones prácticas del control difuso, en las que es tolerable hasta de un 10% de precisión [Yama93].

Por otra parte, la escasa capacidad de adaptación de las primeras generaciones de controladores puramente analógicos [Sasa92][Miki93], ha sido superada por una nueva generación de controladores analógicos programables digitalmente [Guo96] [Batu97] [Fran98], y controladores analógicos neuro-difusos que combinan las principales cualidades de los sistemas de inferencia difusa con la capacidad de aprendizaje de los sistemas neuronales [Jang93b] [Taka94] [Vida95a] [Batu98a].

A esta nueva generación de realizaciones *hardware* [Kand98], pertenecen los trabajos desarrollados en [Vida96b]. En ellos se toma como base la arquitectura neuro-difusa *ANFIS* (*Adaptive Network-Based Fuzzy Inference System*) propuesta en [Jang93b], para desarrollar una metodología de diseño de controladores neuro-difusos analógicos, con tecnologías CMOS estándar, para aplicaciones de baja y media precisión, que utiliza bloques funcionales programables. Consideraciones de simplificación del diseño *hardware* y facilidad para introducir programabilidad [Yama93] llevan a elegir el sistema de inferencia de Takagi-Sugeno, donde el consecuente de cada regla es un valor constante, o *singleton* en terminología de la lógica difusa [Taka89]. Este algoritmo puede plasmarse directamente sobre la topología neuro-difusa *ANFIS* [Jang95] en la que los diferentes operadores involucrados en dicho algoritmo son distribuidos en la estructura en capas de la red. Se distinguen cinco capas de procesamiento, tres de las cuales están formadas por nodos que realizan funciones fijas mientras que dos de ellas contienen nodos que realizan funciones parametrizables y por tanto susceptibles de ser diseñadas para ser programables y realizar chips adaptativos. En [Vida96b] se propone un conjunto de bloques constructivos CMOS para realizar cada una de las capas de la arquitectura *ANFIS*, tanto de las capas fijas como las programables, que permiten un diseño modular del sistema. Con estos elementos se diseñan diferentes chips controladores

reconfigurables, y se proponen algoritmos de aprendizaje compatibles con el *hardware*. Esta metodología de diseño cuida especialmente la simplicidad de los bloques constructivos a nivel de circuito, un prerequisite para poder incrementar la complejidad y la velocidad del controlador.

Sin embargo, uno de los principales obstáculos para incrementar la complejidad de los problemas de control abordables con los controladores difusos analógicos propuestos en [Vida96b] en particular, o con los diversos controladores difusos analógicos reportados en la literatura [Guo96] [Fran98] [Batu98] [Rodr99] en general, es la existencia de nudos de computación global propios del algoritmo difuso que realizan, (como los que aparecen al realizar el proceso de *defuzzificación* empleando el método del centro de gravedad). En estos nudos se suman también los condensadores parásitos, de modo que su velocidad de operación se degrada, así como los errores (sistemáticos o aleatorios) causados por los circuitos que realizan los antecedentes de regla, con lo que el error global final se hace mayor, lo que a su vez supone, entre otras cosas, una degradación en su precisión. Esta circunstancia es especialmente cierta en aplicaciones de control en las cuales es bastante usual emplear una partición en rejilla del universo de discurso [Brow94], lo que da lugar a un incremento exponencial de error global debido principalmente al crecimiento exponencial del número de reglas con el número de entradas. Así, para sistemas de complejidad media las implementaciones analógicas comienzan a tener problemas.

Ideas como la de *reglas activas* [Iked91] que explotan el carácter local de los sistemas de inferencia difusa, y que ya han sido usadas en controladores digitales [Asci95] [Mase95][Asci95] para reducir área y necesidades de cómputo, pueden ser también usadas en el terreno de los controladores analógicos. Por otra parte, propiedades de los sistemas difusos, como la propiedad de descomposición, enunciada en [Zeng96b], y que permite definir el concepto de *sistema difuso más simple posible*, proporcionan ideas que pueden ser aprovechadas para diseñar una nueva arquitectura que trate de paliar las dificultades de los controladores analógicos antes expuestas, y así aumentar la complejidad de los problemas abordables con este tipo de controladores. Este es el caso de la arquitectura de los sistemas propuestos en este trabajo de Tesis. Si, como es el caso en muchos sistemas de control, la salida para cada región del universo de discurso puede obtenerse de un conjunto reducido de reglas, aquellas que contribuyen con valor no nulo en dicha región (reglas activas), este cómputo podría realizarse con circuitería analógica, lo cual se ha demostrado ventajoso en cuanto a prestaciones, y con una precisión aceptable. A la vez, esta circuitería tendría que ser lo suficientemente flexible y rápida para poder ser reconfigurada dinámicamente y operar con las reglas activas en cualquier región del espacio de entrada. Así, esta técnica de multiplexado espacial puede permitir diseñar controladores difusos que preserven las ventajas de una implementación analógica (p.e. paralelismo computacional, compacidad, óptima eficiencia en

cuanto a la razón velocidad/consumo de potencia, etc.), a la vez que se consigue independizar la acumulación de error y de condensadores parásitos del número de reglas, lo que permite un incremento en la complejidad abordable con estos controladores. Para estas tareas de multiplexado y reconfiguración, el empleo de técnicas de diseño de señal mixta digitales-analógicas puede resultar bastante adecuado [Nava97].

Las principales aportaciones de esta Tesis Doctoral se centran en el campo del diseño y aplicación de circuitos integrados analógicos y mixtos de sistemas de inferencia neuro-difusos con tecnologías CMOS estándar, y tienen como objetivo extender el ámbito de aplicación de las realizaciones puramente analógicas a problemas de mayor complejidad preservando sus principales ventajas, explotando las ideas arriba expresadas. En este sentido, los trabajos aquí desarrollados tienen como antecedente directo y suponen una continuación de la línea iniciada en [Vida96b], aprovechan la experiencia de la metodología y los bloques funcionales allí definidos, y los extienden y adaptan a la necesidades de las nuevas ideas desarrolladas.

Más concretamente, en este trabajo de Tesis se presenta una nueva arquitectura para controladores neuro-difusos que desarrolla la idea de multiplexado antes mencionada

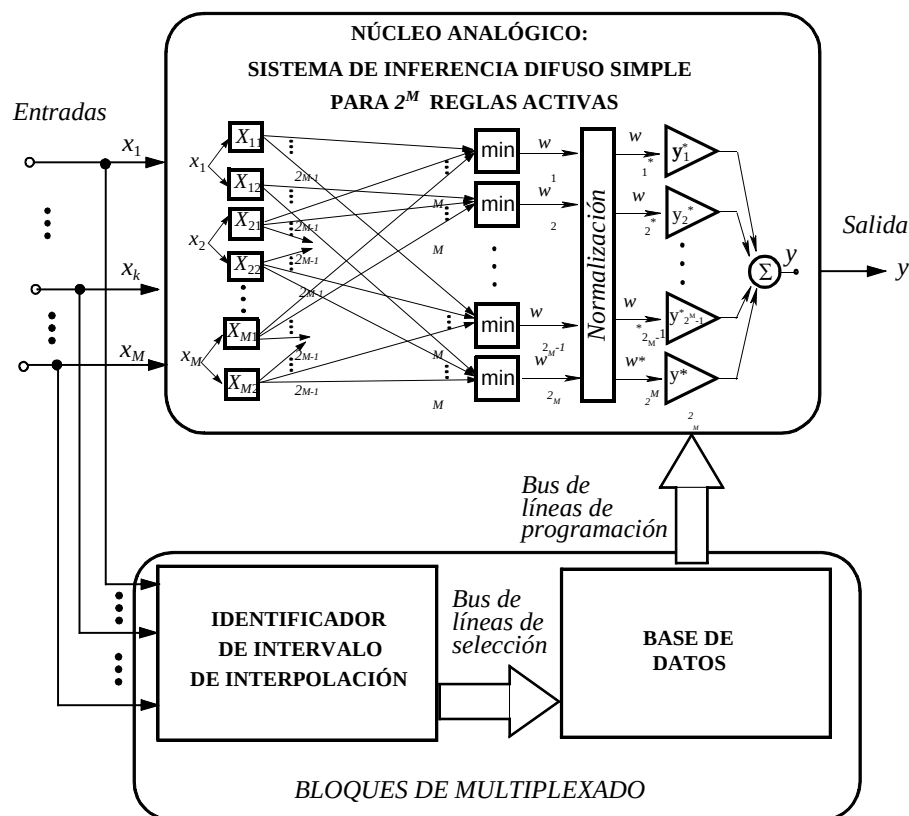


Fig.1 Diagrama de bloques de un sistema difuso de arquitectura multiplexada.

([Vida98a]) con un núcleo analógico basado en el controlador reportado en [Rodr99]. La Fig.1 sirve para ilustrar esta nueva arquitectura. Para ella se proponen realizaciones de cada bloque, que incluyen modificaciones de los bloques del Núcleo Analógico para los circuitos que realizan las funciones de pertenencia, la normalización y la ponderación *singleton*, a fin de optimizar su rendimiento dado el comportamiento dinámico que exige esta arquitectura, así como propuestas para los bloques que realizan la programación dinámica: el bloque Identificador de Intervalo de Interpolación y Base de Datos [Nava99] [Vida00].

Estas propuestas se avalan con los resultados obtenidos sobre un prototipo fabricado con tecnologías CMOS estándar de 0.7 micras con una capa de polisilicio y dos de metal. Este controlador, que ha sido denominado MFCON, posee dos entradas, ocho etiquetas por entrada, y sesenta y cuatro reglas, y resulta ser de mayor complejidad que los reportados hasta ahora en la literatura para este tipo de controladores [Nava00]. A modo de ilustración, la Fig.2(a) muestra una microfotografía del mismo, mientras que la Fig.2(b) muestra un ejemplo de superficie de control con él generada.

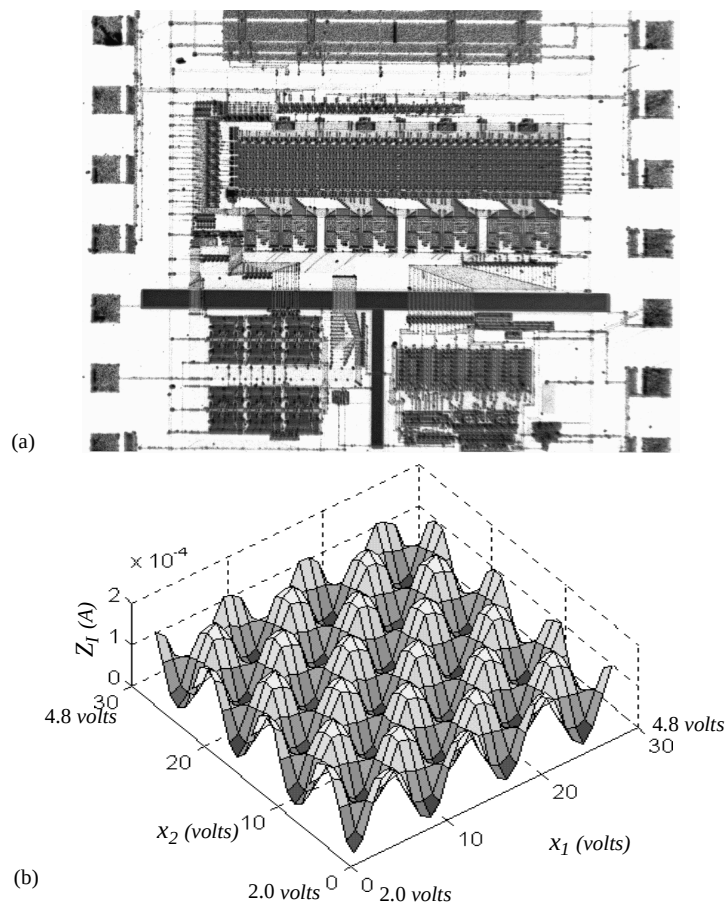


Fig.2 MFCON: Microfotografía (a) y superficie de control (b) .

Además, para ilustrar las prestaciones de MFCON se ha desarrollado un ejemplo de aplicación consistente en el control del arranque de un motor de corriente continua [Nava00a]. La Fig.3 muestra un esquema básico del sistema experimental desarrollado para esta aplicación. Este ejemplo resulta original en la literatura, y de complejidad adecuada para mostrar las cualidades de MFCON cuando funciona integrado en un bucle de control que necesita una alta velocidad de respuesta. Con él se pretende además adquirir una primera experiencia en el campo de la *Electrónica de Potencia* y el *Control de Movimiento*, disciplinas éstas para las cuales, según diversos autores [Bose94] [Hara94] [Boni95], la incorporación de las nuevas tecnologías de la inteligencia artificial, entre las que destaca la lógica difusa, supone un fuerte impacto. Esta experiencia va dirigida a iniciar una línea de trabajo en la búsqueda de aplicaciones de los controladores difusos analógicos y de señal mixta en general, y de los controladores aquí propuestos en particular, en las que sus principales cualidades: interfaz natural con sensores y actuadores, compacidad, buena eficiencia en la razón velocidad/consumo de potencia para resoluciones medias o bajas, muy pequeño retardo entrada-salida, flexibilidad de configuración, y capacidad de aprendizaje, sean explotadas.

En este sentido, el ejemplo de aplicación aquí presentado puede ser considerado como un caso particular del problema más general que constituye el control de velocidad y limitación de corriente en servomecanismos, o en sistemas con motores de velocidad variable. En ambos casos la exigencia en estabilidad y rapidez de respuesta a un cambio en los valores de referencia de posición o velocidad, y la reacción que esta acción provoca en el sistema, es la misma, de manera que el arranque constituye un caso extremo, y la experiencia adquirida puede ser posteriormente aplicada al caso más general. Los resultados experimentales obtenidos para distintos objetivos de arranque muestran la capacidad de control del chip diseñado.

Las ideas arriba expresadas se desarrollan en esta memoria en torno a los siguientes capítulos:

En el Capítulo 1 se repasan conceptos básicos del control difuso y neuro difuso, con especial atención a los sistemas de inferencia del tipo Takagi-Sugeno de orden cero y a su interpretación como algoritmos no lineales de interpolación de superficies de control, visión

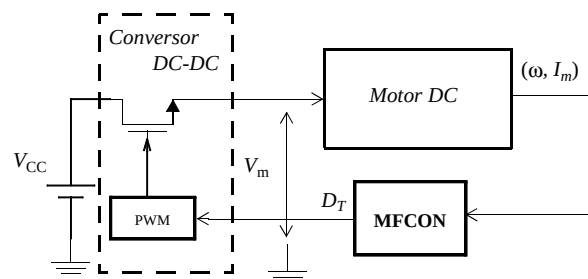


**Fig.3 Esquema básico del ejemplo de aplicación de arranque suave de motores de corriente continua.**



ésta que permite utilizar conceptos de la *Teoría de la Aproximación* para evaluar y cualificar su capacidad de interpolación, así como fundamentar sobre una base teórica la arquitectura que soporta los sistemas desarrollados en esta Tesis. Finalmente se discuten las características de algunas de las implementaciones de controladores analógicos monolíticos reportadas en la literatura y que sirven para destacar las principales aportaciones de este trabajo.

En el Capítulo 2 se desarrolla la propuesta de una arquitectura multiplexada para la realización de controladores neuro-difusos de señal mixta. Se realiza una propuesta de implementación basada en dos grandes bloques funcionales fundamentales: *Núcleo Analógico* y *Bloques de Multiplexado del Núcleo Analógico*, que son a su vez descritos y desarrollados. Finalmente se analiza la capacidad de adaptación de la propuesta.

En el Capítulo 3 se describe y analiza una propuesta de realización con circuitos analógicos CMOS del *Núcleo Analógico* de la arquitectura multiplexada.

En el Capítulo 4 se discuten en primer lugar y de forma global los principales requerimientos de diseño de los *Bloques de Multiplexado del Núcleo Analógico*. A continuación, se discuten los requerimientos y se analiza una propuesta de diseño con circuitos CMOS de uno de estos bloques: el *Bloque Selector de Intervalo*.

En el Capítulo 5 se discuten los requerimientos y se analizan las propuestas de realización con circuitos CMOS de los bloques *Programador de Antecedente de Reglas* y *Programador de Consecuente de Reglas*, lo que completa la descripción de los *Bloques de Multiplexado del Núcleo Analógico*.

En el Capítulo 6 se resumen en primer lugar las principales características del controlador monolítico MFCN, diseñado con tecnología CMOS 0.7  $\mu\text{m}$ , y fabricado para mostrar la viabilidad de la arquitectura propuesta. A continuación se describen los principales aspectos del sistema de test diseñado y empleado en las pruebas de los prototipos del controlador. Finalmente se muestran los resultados de diversas pruebas que ilustran las principales cualidades del comportamiento estático y dinámico del controlador.

En el Capítulo 7 se describe un ejemplo de aplicación de MFCN en el control en lazo cerrado del arranque de motores de corriente continua. En primer lugar se exponen las motivaciones para desarrollar este ejemplo. A continuación se describen los principales aspectos del problema de control considerado y las diversas opciones de control utilizadas. Se describe también el sistema experimental desarrollado para el ejemplo de aplicación y la estrategia de control que incorpora a MFCN en el lazo de control. Finalmente se exponen los resultados experimentales obtenidos en ejemplos en los que se han fijado diferentes objetivos de arranque.

---

## Capítulo I

---

### Aspectos Básicos de los Sistemas Difusos y Neuro-Difusos y su Implementación

#### 1. INTRODUCCIÓN

Los sistemas objeto de esta tesis son circuitos integrados electrónicos de señal mixta, analógica-digital, desarrollados para ser empleados en tareas de control en tiempo real en el marco del *control difuso*. Su diseño está orientado a realizar de manera eficiente y flexible, y mediante cómputo analógico el *sistema de inferencia difusa del tipo Tagaki-Sugeno de orden cero*, que resulta especialmente adecuado para las implementaciones *hardware* [Vitt90], sobre una *arquitectura Neuro-Difusa ANFIS (Adaptive Network-Based Fuzzy Inference System)* [Jang93b]. Estos controladores emplean una *partición en rejilla* de su espacio de entrada y explotan el *carácter local de las reglas difusas*, el concepto de *regla activa* y la *propiedad de descomposición de los sistemas difusos* para poder abordar problemas de alta complejidad (medida en términos de nº de entradas y nº de reglas) con una importante reducción *hardware* (medida en ocupación de área de silicio y consumo de potencia) y manteniendo la precisión dentro de los límites admisibles en la práctica en los problemas de control. La arquitectura resultante es altamente modular y escalable, programable en chip y compatible con *algoritmos de aprendizaje hardware* [Rodr99].

En este capítulo se comentan algunos de los conceptos y términos básicos que definen los principales rasgos estructurales de los sistemas de *control difuso* y *neuro-difuso* desarrollados en este trabajo. Dada la madurez alcanzada por la *Teoría de los Conjuntos Difusos*, la *Lógica Difusa* y los *Sistemas de Control Difuso y Neuro-Difuso*, gracias a las contribuciones de los muchos autores que, desde los trabajos inaugurales de L.A. Zadeh [Zade65] [Klir96], han dedicado su esfuerzo al desarrollo de esta rama, en la actualidad existe una amplia bibliografía que incluye tanto tutoriales [Lee90] [Mend95] como manuales para el desarrollo de cursos [Wang97] [Brow94] [Lin96], y que recogen de una manera sistemática y estructurada los conceptos básicos de la teoría, por lo que éstos no serán desarrollados en este trabajo. En este capítulo se ha prestado especial atención a la interpretación de los sistemas difusos como algoritmos no lineales de interpolación de superficies, visión ésta, que permite utilizar conceptos de la *Teoría de la Aproximación* para evaluar y cualificar su capacidad de interpolación, así como fundamentar sobre una base teórica la nueva arquitectura propuesta que soporta los sistemas desarrollados en esta tesis. Finalmente se discuten las características de algunas de las implementaciones de controladores analógicos monolíticos reportadas en la literatura y que sirven para destacar las principales aportaciones de este trabajo. Se hace

especial hincapié en las realizaciones propuestas en [Vida96b] por ser estas antecedente directo de los circuitos aquí desarrollados.

## 2. SISTEMAS DIFUSOS

Los Sistemas Difusos, proporcionan un marco de procesamiento numérico para la información propia de los sistemas basados en conocimiento, formulándola de una manera sistemática que intenta emular las propiedades del razonamiento humano. Esta técnica combina la *Teoría de los Conjuntos Difusos* y la *Lógica Difusa* [Zade65] y consigue reproducir comportamientos complejos, en general no lineales, enunciados en forma de relaciones cualitativas e imprecisas, propios del lenguaje natural, y expresarlos en términos numéricos, compatibles con los métodos usados en los sistemas de ingeniería [Zade73].

La estructura básica de un sistema difuso puro se muestra en la Fig.1.1 y consta de dos bloques fundamentales denominados:

- *Base de Reglas*, y
- *Máquina de Inferencia*.

El primero sistematiza y estructura los enunciados que constituyen el conocimiento que se posee sobre el problema en cuestión, en forma de sentencias condicionales del tipo

**"SI A (situación) ENTONCES B (acción)"**,

que resultan especialmente adecuadas para tareas de control.

Cada una de estas sentencias posee dos importantes características: En primer lugar, expresan un conocimiento cualitativo más que cuantitativo; y en segundo lugar, establecen una relación causa efecto que posee un carácter marcadamente local. Ambas características se formalizan y convierten a términos numéricos con la ayuda de los conceptos de la Teoría Difusa. Así su carácter cuantitativo se recoge en el concepto de *Conjunto Difuso*, mientras que la localidad de relación causal se corresponde con el concepto de *Relación Difusa*, o *Implicación Difusa*.

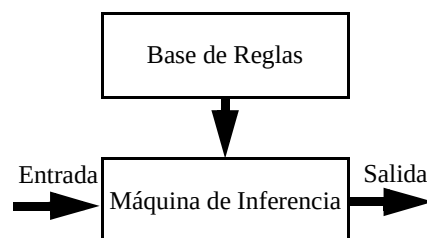


**Fig.1.1 Diagrama de bloques de un sistema de inferencia difuso puro.**

Por otra parte, la capacidad de generalización del experto humano se formaliza mediante la Máquina de Inferencia que combina la información almacenada en la Base de Reglas empleando el concepto de *Regla de Composición*. De este modo, para una entrada dada que expresa una *situación*, no necesariamente coincidente con alguna de las recogidas en la Base de Reglas, pero que presenta cierta semejanza con una o varias de ellas, el sistema proporciona a la salida una *acción* adecuada, no necesariamente coincidente con alguna de las recogidas en dicha base, aunque más próxima a aquella respuesta que proporcionaría la regla o reglas con la que presente mayor semejanza.

Aunque bastante adecuado para emular a un operador humano desde un punto de vista conceptual y cualitativo, el sistema difuso así definido no resulta útil en problemas prácticos de ingeniería y control automático, donde tanto las *situaciones*, (resultado de medidas de sensores), como las *acciones* de control, (señales de control sobre actuadores), vienen expresadas como variables reales. Por esta razón los *controladores difusos*, (esto es, los sistemas difusos empleados en tareas de control) incorporan dos elementos de interfaz adicionales al esquema básico del sistema difuso:

- Un sistema, "*fuzzificador*", que convierte un valor no difuso, "*crisp*", en un conjunto difuso; y
- Un sistema "*defuzzificador*", que convierte la salida del sistema difuso en un valor real, "*crisp*".

La Fig.1.2 muestra un esquema de la configuración básica de un sistema de control difuso, que incorpora los bloques de interfaz antes mencionados. Cada uno de los bloques que aparecen en la figura pueden ser caracterizados mediante fórmulas matemáticas precisas proporcionadas por la Teoría de los Conjuntos Difusos. De esta manera un controlador difuso con múltiples entradas y múltiples salidas realiza en general un mapeado no lineal de vectores del espacio de entrada en vectores del espacio de salida, por lo que, desde este punto de vista, puede ser considerado como un sistema *aproximador de funciones*. Este es un aspecto importante que diferencia a los Sistema Difusos de los Sistemas Expertos y de Inteligencia Artificial, y que los hace más atractivos que éstos para su implementación *software* o *hardware* en tareas de control automático.

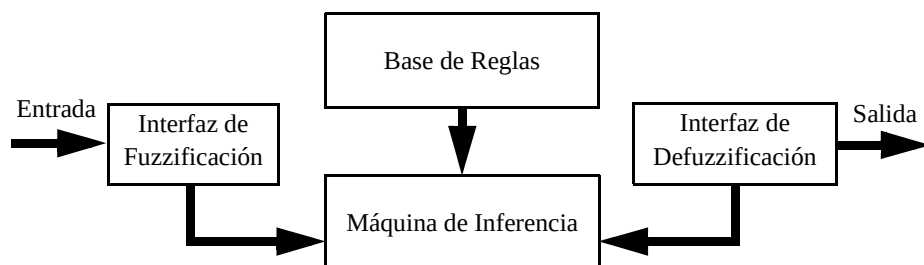


Fig.1.2 Diagrama de bloques de un sistema de inferencia difusa.

El procedimiento que realizan los controladores difusos, llamado *inferencia difusa* o *razonamiento aproximado* en general comprende los siguientes pasos [Jang93b]:

1. Evaluar el *grado de pertenencia* de cada entrada a cada conjunto difuso o *etiqueta lingüística* definida en su espacio de entrada, este proceso se denomina *fuzzificación*.
2. Combinar los grados de pertenencia de cada una de las entradas que constituyen el antecedente de regla para obtener *el peso de cada regla* en la salida global, por medio del *operador t-norma*.
3. Evaluar cada regla para obtener su *consecuente* a partir del peso dado por el antecedente y la función de pertenencia asociada al consecuente, este proceso se denomina *implicación difusa*.
4. Componer los consecuentes de cada regla para obtener una conclusión, este proceso se denomina *regla de composición*.
5. Obtener una acción de control no difusa, proceso denominado *defuzzificación*.

En la práctica se distinguen tres tipos fundamentales de sistemas de inferencia difusa que se diferencian básicamente en la representación de los consecuentes de las reglas difusas y en sus procedimientos de agregación y *defuzzificación* [Jang93b][Lin96]:

- **Tipo 1.** Es el caso más general. Emplea reglas enunciadas en términos de variables lingüísticas tanto para las entradas como para las salidas en la forma

$$\text{SI } x_1 \text{ es } A_i, \dots, x_n \text{ es } B_i, \text{ ENTONCES } y = C_i$$

La salida difusa de cada regla se obtiene combinando su función de pertenencia de salida, bien empleando un operador de mínimo (**modelo de Mamdani**), bien un operador producto (**modelo de Larsen**), con un factor obtenido de evaluar el antecedente de la regla para una entrada dada. La salida difusa del sistema se obtiene combinando con operador máximo las salidas de todas las reglas. Para obtener una salida no difusa del sistema se pueden emplear diferentes métodos, siendo el más usado el llamado *método del centroide*.

- **Tipo 2 (modelo de Tsukamoto):** Emplea reglas del mismo tipo que el anterior, salvo que la función de pertenencia de la salida  $C_i$  es una función monótona creciente. La salida final del sistema es la suma ponderada de cada salida de regla, dada por el peso de la regla resultante del antecedente y por la función monótona creciente del consecuente.
- **Tipo 3 (modelo de Takagi-Sugeno):** Emplea reglas enunciadas en términos de variables lingüísticas para las entradas, y el consecuente de cada regla se obtiene

como una combinación lineal de los valores de entrada más un término constante.

$$\text{SI } x_1 \text{ es } A_i, \dots, Y \text{ } x_n \text{ es } B_i, \text{ ENTONCES } y = f_i(x_1 \dots x_n)$$

La salida del sistema es una suma ponderada de las salidas de las reglas.

Los tres tipos de inferencia se ilustran en la Fig.1.3 [Jang93b], en la que por simplicidad se muestra el caso de un sistema con dos entradas.

A pesar de su popularidad, la implementación *hardware* de cualquiera de estos tres mecanismos de razonamiento en su forma más general resulta costosa en términos de área de silicio y consumo de potencia. Un caso particular, común a los tres tipos de inferencia difusa, es aquel en que las reglas difusas quedan enunciadas así:

$$\text{SI } x_1 \text{ es } A_i, \dots, Y \text{ } x_n \text{ es } B_i, \text{ ENTONCES } y = y^*$$

donde  $y^*$  es un valor constante no difuso o *singleton*.

Para el tipo de razonamiento 1, ya sea el modelo de *Mamdani*, o el modelo de *Larsen*, el *singleton* es un caso extremo de función de pertenencia correspondiente a un conjunto difuso normalizado cuyo *soporte* y *núcleo* coinciden en un único valor. En el tipo de razonamiento 2

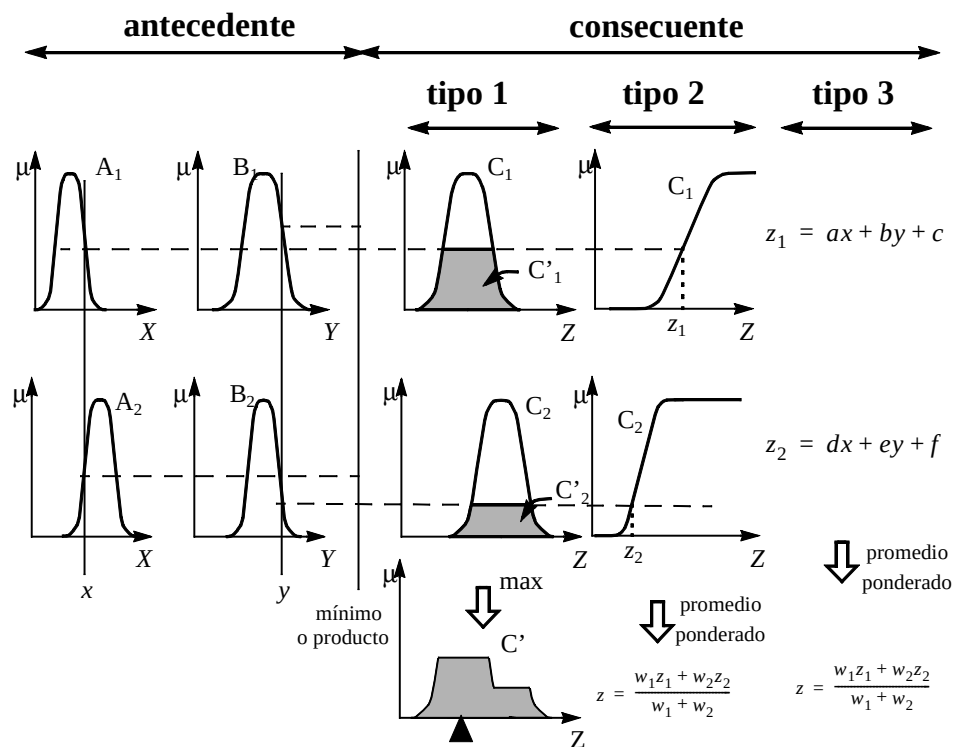


Fig.1.3 Mecanismos de razonamiento difuso más usuales.

de Tsukamoto, se encuentra el equivalente si se consideran funciones escalón en el consecuente, centradas en el valor del *singleton*. Finalmente, este razonamiento coincide también con el de tipo 3 o de *Takagi-Sugeno* cuando sólo el término independiente de la expresión lineal del consecuente es distinto de cero, lo que se conoce como *modelo de Sugeno de orden cero* [Jang95]. Se demuestra [Jang93b][Brow94] que, bajo ciertas condiciones, el sistema de inferencia así obtenido es capaz de aproximar cualquier función no lineal arbitraria con el grado de precisión que se desee. Debido a esto, y a su sencillez en el cálculo del consecuente este tipo de razonamiento es el más extendido entre las implementaciones *hardware* de los sistemas de inferencia difusos [Vitt90] .

En el diseño de sistemas difusos se explota su característica transparencia al conocimiento humano para definir su estructura, esto es, para determinar el número de entradas, número de etiquetas lingüísticas necesarias, forma y distribución de las funciones de pertenencia sobre el espacio de entrada, número de reglas, sistema de inferencia a emplear, etc. Sin embargo, no existe un método sistemático de diseño que pueda ser empleado de forma general y casi siempre, después de haber determinado su estructura se hace necesario un proceso de ajuste fino para conseguir que la relación entrada-salida deseada se mantenga dentro de un error aceptable.

### 3. SISTEMAS NEURO-DIFUSOS

Los sistemas *Neuro-Difusos* son resultado de la simbiosis de dos tecnologías: las *Redes Neuronales* y los *Sistemas Difusos*, las cuales compiten en el campo del diseño de sistemas inteligentes con las técnicas de la *Inteligencia Artificial* y las de la *Teoría de Control* convencional e incluso las mejoran, en el caso en que estos sistemas han de trabajar en entornos ruidosos, cambiantes y con cierto grado de imprecisión e incertidumbre.

Los sistemas *Neuro-Difusos* tratan de explotar la complementariedad existente entre ambas tecnologías, aprovechando los méritos, y eludiendo las lagunas de cada una de ellas por separado. No está en el propósito de este capítulo realizar una comparación exhaustiva de cada una de ellas, sin embargo, a fin de destacar los méritos y lagunas antes citados, y hacer patente su complementariedad en la Tabla 1.1 [Lin96] se resume el resultado de la comparación entre las distintas técnicas usadas en el diseño de sistemas inteligentes atendiendo a criterios diversos. Como puede observarse los *Sistemas Difusos* y las *Redes Neuronales* resultan superiores a la *Teoría de Control* convencional y a la *Inteligencia Artificial* en su manejo de la no linealidad y en cuanto a tolerancia a fallos e incertidumbre. Por otra parte, los *Sistemas Difusos* y la *Inteligencia Artificial* son superiores a las *Redes Neuronales* y a la *Teoría de Control* convencional en su transparencia para representar conocimiento estructurado o basado en la experiencia, mientras que las *Redes Neuronales* resultan superiores a las demás técnicas

en su capacidad de aprendizaje y optimización. Por último todas las técnicas resultan apropiadas para operaciones en tiempo real salvo la de la Inteligencia Artificial.

**Tabla 1.1:** Comparación entre los Sistema Difusos (SD), Redes Neuronales (RN), Control Convencional (CC) e Inteligencia Artificial (IA)

| Criterio                            | SD        | RN        | CC    | IA    |
|-------------------------------------|-----------|-----------|-------|-------|
| Modelado Matemático                 | Aceptable | Malo      | Bueno | Pobre |
| Aprendizaje                         | Malo      | Bueno     | Malo  | Malo  |
| Transparencia                       | Bueno     | Malo      | Pobre | Bueno |
| No linealidad                       | Bueno     | Bueno     | Malo  | Pobre |
| Optimización                        | Malo      | Aceptable | Pobre | Malo  |
| Tolerancia a fallos e incertidumbre | Bueno     | Bueno     | Malo  | Malo  |
| Operación en tiempo real            | Bueno     | Aceptable | Bueno | Malo  |

Las Redes Neuronales, debido principalmente a su capacidad de aprendizaje y adaptación, resultan apropiadas para sistemas que procesan y adquieren automáticamente información numérica proveniente de sensores; o que resuelven problemas de reconocimiento de patrones o de optimización. Sin embargo su modo de procesar la información resulta difícil de expresar en términos de lenguaje natural, (se dice que las Redes Neuronales no son transparentes al conocimiento estructurado), lo que muchas veces dificulta o impide explotar el conocimiento del experto en las tareas de diseño.

Por su parte los Sistemas Difusos, debido principalmente a su habilidad para incorporar a su estructura información lingüística tomada del experto humano (transparencia), son especialmente atractivos para automatizar sistemas que son controlados con éxito por operadores humanos y que bien por su complejidad, bien por su deficiente especificación, debida principalmente a las imprecisiones e incertidumbres del entorno en que se desarrollan, resultan difíciles o muy costosos de modelar con la simplicidad necesaria para ser tratados por otros métodos. Sin embargo su principal dificultad estriba su falta de adaptabilidad.

Los sistemas Neuro-Difusos, además de los rasgos que son comunes a ambas tecnologías, (mapeado no lineal, tolerancia a fallos, operación en tiempo real, etc), se benefician de:

- La transparencia propia de los Sistemas Difusos que mejora la comprensión de la Red Neuronal.
- La capacidad de aprender y adaptarse de las Redes Neuronales que aporta un mecanismo de sintonización automática, y adaptabilidad al Sistema Difuso.

Esta simbiosis resulta especialmente interesante en aplicaciones de control. En este



contexto ambos sistemas se muestran funcionalmente equivalentes cuando se interpretan como *aproximadores universales de funciones*<sup>1†</sup>[Lin96][Brow94].

Tanto las redes neuronales, como los sistemas de inferencia difusas son sistemas dinámicos que pueden ser considerados como aproximadores de funciones sin modelo, entendiendo por tal, sistemas que tienen la capacidad de estimar el valor la función que caracteriza su relación entrada-salida, para cualquier elemento de su soporte, sin necesidad de disponer de un modelo o descripción analítica de la misma. En ambos casos, esta estimación es realizada partir de información adquirida por medio de ejemplos proporcionados por el diseñador. Debido a su naturaleza, esencialmente numérica, ambos sistemas admiten un soporte matemático riguroso y una caracterización algorítmica que favorece el uso de herramientas de cálculo para su evaluación, y que resulta atractiva para su implementación mediante circuitos electrónicos integrados, debido principalmente a su marcado carácter de procesamiento paralelo.

Los sistemas Neuro-Difusos diseñados para aplicaciones de control reciben el nombre de controladores Neuro-Difusos. Estos sistemas se caracterizan por mantener los rasgos fundamentales de los sistemas de inferencia difusa, entre ellos su transparencia para incorporar el conocimiento de un experto humano, mientras que aprovechan la capacidad de aprendizaje de las Redes Neuronales para dotarse de la flexibilidad y la capacidad de adaptación necesaria para, bien ayudar en la tarea de automatizar su proceso de diseño, bien adaptar su comportamiento frente a cambios en el entorno.

En un contexto más amplio la integración de los sistemas difusos y las redes neuronales sigue tres líneas diferentes de trabajo que da lugar a tres tipos diferentes de sistemas [Lin96][Lin96][Lin96] [Nie95]:

- *Sistemas Neuro Difusos*: Usan las Redes Neuronales como herramienta para el diseño de Sistemas Difusos.
- *Redes Neuronales Difusas*: Traducen a términos de los Sistemas Difusos las Redes Neuronales tradicionales.
- *Sistemas híbridos Neuro-Difusos*: Incorporan ambas tecnologías en un nuevo sistema híbrido.

Los sistemas Neuro-Difusos, son sistemas difusos que aprovechan métodos de ajuste propios de las redes neuronales; por tanto en ellos se siguen identificando los bloques fundamentales del sistema difuso ( base de reglas, máquina de inferencia, e interfaces de

---

<sup>1†</sup>. Esta equivalencia funcional se convierte también en estructural en el caso concreto de las redes **RBFN** (Radial Basis Function Network) y de los sistemas de inferencia difusa que utilizan el operador producto como t-norma, funciones de pertenencia Gaussianas y reglas con valores *singleton* en el consecuente.

fuzzificación y defuzzificación). Las redes neuronales son utilizadas para aumentar y mejorar la capacidad de procesamiento del sistema difuso, en tareas como la definición y ajuste de las funciones de pertenencia y el diseño y ajuste de la base de reglas. Como estos sistemas siguen siendo sistemas difusos su principal área de aplicación sigue siendo la del control.

Las Redes Neuronales Difusas, mantienen las propiedades y estructura básicas de las redes neuronales e introducen conceptos de la Lógica Difusa en alguno de sus elementos. En algunos casos los conceptos difusos se introducen en la topología de la red. Así aparece la *neurona difusa* en la que su respuesta deja de ser una función de tipo sigmoideal, para ser sustituida por una relación difusa. En otros casos, la formalización del conocimiento en forma de conjuntos difusos es usada para mejorar las propiedades de los algoritmos de aprendizaje, el modo de actualización de pesos y la velocidad de convergencia, aumentando sus capacidades de interpretación. Debido a que estos sistemas son básicamente redes neuronales, su campo de aplicación sigue siendo fundamentalmente el de aplicaciones de reconocimiento de patrones.

En los sistemas Híbridos Neuro-Difusos, los sistemas difusos y las redes neuronales realizan su propio trabajo de forma independiente y complementaria y cooperan en las diferentes funciones para cumplir el objetivo común del sistema global. Son en general sistemas cuya arquitectura depende mucho del tipo de aplicación y resultan adecuados tanto para tareas de control como de reconocimiento de patrones.

Los sistemas objeto de esta tesis quedan encuadrados dentro del campo de los sistemas de control Neuro-Difuso por ser su estructura básica la de la red adaptativa *ANFIS* que permite trasladar los diversos algoritmos de inferencia difusa a una topología de red neuronal, incorporándoles la capacidad de aprendizaje. Las principales características de esta red serán descritas con posterioridad en este capítulo.

#### 4. DISEÑO DE CONTROLADORES DIFUSOS Y NEURO-DIFUSOS

Como ya se ha mencionado, los controladores difusos emplean el procedimiento llamado de *inferencia difusa* para modelar sistemas no lineales multidimensionales [Mend95]. Esto es equivalente a capturar el comportamiento del sistema por medio de una relación

$$y = f(\mathbf{x}) \quad (1.1)$$

donde  $y$  es la salida del sistema y  $\mathbf{x} = \{x_1, x_2, \dots, x_M\}^T$  es un vector de entrada <sup>2†</sup>. En aplicaciones de control las entradas son llamadas habitualmente *hechos*, las salidas *acciones* y la ley que los relaciona *superficie de control* [Taka94].

<sup>2†</sup>. Sin pérdida de generalidad asumiremos que vamos a trabajar con sistemas con múltiples entradas y una sola salida (MISO), debido a que un sistema difuso con múltiples entradas y múltiples salidas (MIMO) siempre puede ser descompuesto en varios sistemas con una única salida [Lee90].

Algunas de las características del modelado no lineal que realizan los sistemas difusos son [Brow94][Jang95]:

- El modelo global del comportamiento del sistema para cualquier condición de entrada, representado por la superficie de control, se obtiene como composición de funciones locales, que representan ese comportamiento en una región limitada del espacio de entrada.
- Esas funciones locales pueden ser descritas por medio sentencias enunciadas en lenguaje natural del tipo,

$R_k: \text{SI } x_1 \text{ es } A_{1k} \text{ Y } x_2 \text{ es } A_{2k} \text{ Y } \dots x_M \text{ es } A_{Mk} \text{ ENTONCES Acción en el Consecuente}$

que reciben el nombre de *reglas difusas*, y donde cada  $A_{ik}$  es *el conjunto difuso o etiqueta lingüística* correspondiente a la *variable lingüística difusa*  $A_i$  asociada a la entrada  $x_i$ , que interviene en la regla  $k$ -ésima, mientras que la *Acción en el Consecuente* contribuye a la salida  $y$ , con un peso que depende del resultado de la evaluación de los antecedentes de regla.

- El grado de cumplimiento del antecedente de la sentencia “Si  $x_i$  es  $A_{ik}$ ” se denomina *grado de pertenencia*, y se calcula evaluando una función no-lineal, continua, asociada a cada etiqueta  $A_{ik}$ , que toma valores en el intervalo  $[0, 1]$ , a la que se denomina *función de pertenencia*,  $s_{ik}(x_i)$ , la cual toma valores distintos de cero solamente dentro de un subintervalo del intervalo completo para el que está definida la entrada  $x_i$ .
- La salida  $y$  se obtiene agregando la *Acción en el Consecuente* de todas las reglas, cada una de ellas ponderada por un factor que depende del grado de cumplimiento de su respectivo antecedente.

Estas características garantizan, por una parte, una interpolación suave de la superficie de control a partir del conocimiento local expresado mediante las reglas difusas, y por otra, que cualquier cambio localizado, esto es, que se produzca en una región limitada del espacio de entradas, pueda ser fácilmente incorporado al modelo global modificando solamente aquellas reglas que se vean afectadas.

Por otra parte, como ya se ha mencionado anteriormente, reciben el nombre controladores neuro-difusos los sistemas neuro-difusos diseñados para aplicaciones de control. Estos se caracterizan por mantener los rasgos fundamentales y los componentes básicos de los sistemas de inferencia difusa, mientras que aprovechan la capacidad de aprendizaje de las redes neuronales para dotar al conjunto de la flexibilidad y la capacidad de adaptación necesaria que permita automatizar su proceso de diseño, o adaptar su comportamiento frente a cambios en el entorno. La Fig.1.4 muestra un modelo general de un controlador neuro-difuso.

#### 4.1. Aspectos generales de diseño

Diseñar un sistema de control difuso y/o neuro difuso supone :

- Determinar su estructura.
- Ajustar los parámetros que caracterizan a los elementos de esa estructura.
- Y, en su caso, seleccionar una estrategia de aprendizaje.

Definir la estructura del sistema consiste en resolver cuestiones como:

- Determinar el número de variables de entrada-salida, y, para cada una de ellas, el número y distribución de sus etiquetas lingüísticas, las cuales determinan una partición del universo de discurso.
- Determinar el tipo de sus funciones de pertenencia asociadas a cada etiqueta lingüística.
- Determinar las reglas que constituyen la base de reglas. Esto es, su número y su enunciado en terminos de antecedente y consecuente.
- Seleccionar una máquina de inferencia, eligiendo los operadores *t-norma* y *t-conorma* a emplear.
- Seleccionar un método de *defuzzificación*, etc.

Una vez definida la estructura se hace necesario ajustar los parámetros de los distintos elementos que la forman, cómo son los parámetros que definen la forma y localización de las funciones de pertenencia, o el peso de cada una de las reglas, etc. En los controladores difusos, esta tarea es llevada a cabo mediante métodos de ensayo y error, o mediante algún sistema de

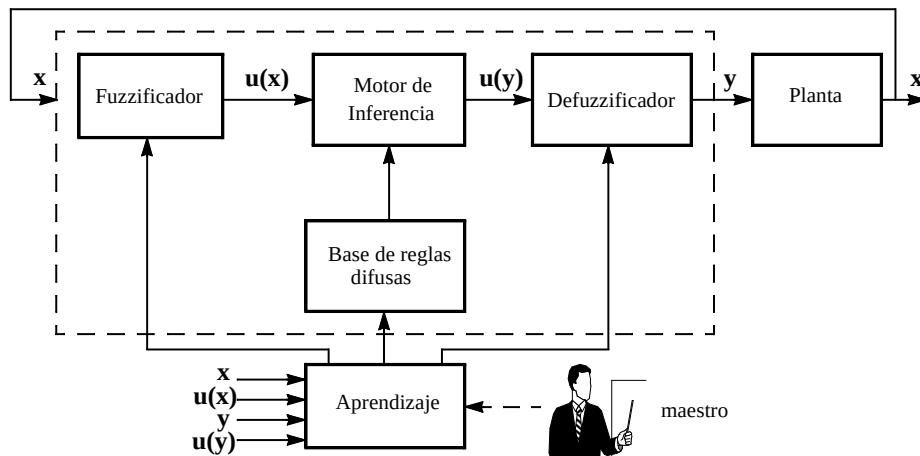


Fig.1.4 Modelo general de un controlador neuro-difuso.

simulación. Por su parte, en los controladores neuro-difusos, las técnicas de aprendizaje de las redes neuronales son empleadas para encontrar y ajustar los parámetros y/o la estructura de estos sistemas y cumplir de manera óptima el objetivo de control.

El objetivo de aprendizaje de la estructura y/o los parámetros sobre los distintos tipos de sistemas de inferencia difusa (*Mamdani*, *Tsukamoto* y *Takagi-Sugeno*) puede ser llevado a cabo de diferentes maneras según se emplee alguno de los diferentes esquemas de aprendizaje de las Redes Neuronales, lo que da lugar a distintas estrategias para el diseño de sistemas de control Neuro-Difusos.

Los tres principales esquemas de aprendizaje de las Redes Neuronales son:

- Aprendizaje no supervisado. En este esquema el modelo interno se construye capturando las regularidades que muestran los vectores de la entrada, sin necesidad de información adicional, como puede ser por ejemplo la respuesta esperada para una entrada dada. Resultan, por tanto, muy utilizados para tareas de clasificación o extracción de características.
- Aprendizaje supervisado. El modelo interno se construye a partir de parejas entrada/salida proporcionadas por un tutor (de ahí su nombre), el objetivo de aprendizaje se lleva a cabo minimizando una función de error.
- Aprendizaje con refuerzo. El modelo interno se construye a partir de parejas entrada/salida proporcionadas por un tutor, que además evalúa y orienta el proceso de aprendizaje según se avance hacia el objetivo de minimizar una función de error.

La primera estrategia resulta atractiva cuando el objetivo es el aprendizaje de la estructura y la identificación de reglas difusas; mientras que las dos últimas lo son adecuadas para el ajuste de parámetros.

#### 4.2. Partición del espacio de entrada

La partición del espacio de entrada es un aspecto importante para la determinación de la estructura de un controlador difuso, sea cual sea el sistema de inferencia que emplee, y está directamente relacionado con el número total de reglas, y con la forma que van a adoptar los antecedentes de regla.

En la Fig.1.5 se ilustran, para el caso bidimensional, diversas estrategias que son utilizadas habitualmente para definir una partición del espacio de entrada [Jang95]:

- En la Fig.1.5(a) se representa una partición en rejilla. En general, esta estrategia es la más simple porque requiere un número pequeño de funciones de pertenencia por entrada; nótese que en la Fig.1.5(a), la función de pertenencia  $s_{xA}(x)$  puede ser

compartida en las subregiones A, B y C. Además, este tipo de partición da lugar a reglas lingüísticas con una sintaxis uniforme, e interpretables en términos sencillos. Por ello es una de las más usadas en problemas de control donde se emplea como entrada un número pequeño de variables de estado y la estrategia de control se establece en base a la magnitud de la desviación de esas variables respecto su valor para la trayectoria objetivo, empleando, por ejemplo, etiquetas lingüísticas como "grande", "pequeño". Su principal inconveniente está en el crecimiento exponencial del número de reglas con el número de entradas, que puede llegar a resultar excesivo incluso para un número de entradas moderado.

- En la Fig.1.5(b) se muestra una partición en árbol. Esta estrategia salva el problema del incremento exponencial del número de reglas y define cada región del universo de discurso de manera unívoca utilizando un esquema de árbol de decisión. Sin embargo, se necesita especificar un mayor número de funciones de pertenencia por entrada para definir cada una de esas regiones, observe que en la Fig.1.5(b) se necesitan tres funciones de pertenencia distintas en la variable de entrada  $x$  para caracterizar las regiones A, B y C. Además las reglas así obtenidas, presentan una sintaxis más irregular y son más difíciles de interpretar en términos lingüísticos.
- En la Fig.1.5(c) se muestra una partición dispersa. Esta estrategia limita el número de reglas a un tamaño razonable. Pero no permite un tratamiento sistemático, y depende fuertemente del conocimiento que *a priori* se tenga del problema, o de la capacidad de extracción de este tipo de reglas a partir de datos.

En la práctica, cuando se posee cierto conocimiento sobre un problema enunciado en forma de reglas, éste se traslada en primer lugar a una matriz de reglas (que determinan una arquitectura, esto es, el número de entradas, número de funciones de pertenencia por entrada, etc) y posteriormente se utiliza algún mecanismo para ajustar los parámetros libres de la

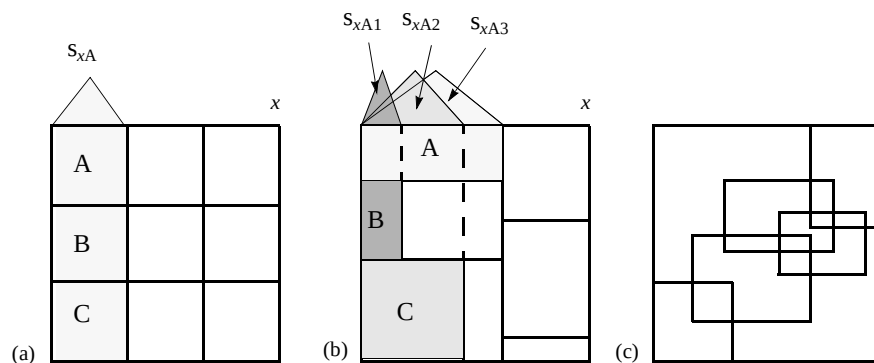


Fig.1.5 Diferentes alternativas para realizar una partición del espacio de entrada: partición en rejilla (a) ; partición en árbol(b); partición dispersa (c).

arquitectura hasta conseguir las prestaciones deseadas. Sin embargo, en otras ocasiones sólo están disponibles conjuntos de datos obtenidos a partir de medidas. En este caso se aborda el problema de procesar esos datos para extraer de ellos información sobre la estructura. Como se vera más adelante, en el diseño de controladores *hardware*, la opción es, o bien diseñar un sistema con el mayor grado posible de flexibilidad para realizar cualquier tipo de partición, lo que supone un importante gasto en área y consumo de potencia, así como cierta dificultad de manejo, debido principalmente al elevado número de parámetros a determinar y ajustar; o bien guiarse por un criterio de sencillez de uso y ahorro de área y consumo, llegando a un compromiso en cuanto a flexibilidad y capacidad de modelado.

#### 4.3. Sistema de Inferencia Takagi-Sugeno con *singleton* en el consecuente

De las diferentes maneras de realizar este algoritmo difuso que se proponen en la literatura [Mend95] [Jang95] y que se han mencionado con anterioridad, Fig.1.3, el *sistema de inferencia de Takagi-Sugeno* donde el consecuente de regla es un valor constante o *singleton* [Taka89], también llamado *sistema Takagi-Sugeno de orden cero*, es el más popular y más ampliamente utilizado en las realizaciones *hardware*. Esto es debido a que establece un muy buen compromiso entre la simplicidad, en cuanto diseño y circuitería necesarias para su realización, y la eficiencia en el control; como recogen las discusiones que a este respecto se realizan en [Yama93], [Miki95] y [Batu97], y su elección en los diferentes controladores reportados [Guo96] [Fran98] [Rodr99]. Por esta razón éste es el sistema de inferencia elegido para los sistemas objeto de este trabajo. A continuación se describen su principales características, tal y como es empleado en los sistemas objeto de este trabajo.

En el sistema de inferencia *Takagi-Sugeno de orden cero*, cada una de las reglas difusas toma la forma

$$\begin{aligned} \text{SI } (x_1 \text{ es } A_{1k}) \text{ Y } (x_2 \text{ es } A_{2k}) \text{ Y } \dots (x_M \text{ es } A_{Mk}) \\ \text{ENTONCES } y = y_k^* \quad (1 \leq k \leq N) \end{aligned} \quad (1.2)$$

donde  $y_k^*$  es el valor *singleton* asociado a la  $k$ -ésima regla, y  $N$  el número total de reglas del sistema. Esta elección resulta ventajosa para su realización *hardware* porque permite implementaciones más simples, con lo que se reduce la ocupación de área y el consumo de potencia, a la vez que facilita la incorporación de programabilidad, al mejorar la interpretación de las reglas [Yama93].

Además, se demuestra que el sistema difuso así obtenido se comporta como *aproximador universal de funciones*, siendo capaz de aproximar cualquier superficie de control con el grado de precisión que se desee [Wang92] [Zeng95] [Zeng96a] [Wang97].

La superficie de control se obtiene como una combinación lineal de *funciones base* según la expresión

$$y = f(\mathbf{x}) = \sum_{k=1, N} y_k^* w_k^*(\mathbf{x}) \quad (1.3)$$

donde  $w_k^*(\mathbf{x})$  es una función base multidimensional normalizada, correspondiente a la salida del antecedente de la  $k$ -ésima regla difusa, mientras que el coeficiente  $y_k^*$  es el valor *singleton* asociado a su consecuente. Con la normalización se garantiza que la salida del sistema en cualquier punto no toma valores mayores que el mayor de los valores *singleton*. En la expresión (1.3) los valores *singleton* son parámetros reales, mientras que las funciones base normalizadas se calculan según la expresión,

$$w_k^*(\mathbf{x}) = \frac{w_k(\mathbf{x})}{\sum_{k=1, N} w_k(\mathbf{x})} \quad (1.4)$$

donde  $w_k(\mathbf{x})$  corresponde con la salida del antecedente de la  $k$ -ésima regla difusa y se obtiene a partir de las entradas según la expresión,

$$w_k(\mathbf{x}) = \min\{s_{1k}(x_1), s_{2k}(x_2), \dots, s_{Mk}(x_M)\} \quad (1.5)$$

En esta expresión  $\min\{\bullet\}$  es el operador mínimo multidimensional, mientras que  $s_{ik}(x_i)$ , son las funciones de pertenencia sobre las que se evalúan el grado de coincidencia entre las entradas y las etiquetas lingüísticas difusas.

Estas funciones de pertenencia, junto con los valores *singleton*, que en el contexto de aproximación de funciones pueden ser considerados como muestras de la superficie de control deseada, constituyen los elementos primarios de información a partir de los cuales el sistema de inferencia difusa reconstruye la superficie de control completa. De manera que, como muestra la expresión (1.3), la salida  $y$  para un vector de entrada  $\mathbf{x}$ , se obtiene agregando de forma ponderada las muestras de la superficie de control deseada, siendo el factor de ponderación el resultado de la evaluación de cada función base  $w_k^*(\mathbf{x})$  para dicho vector de entrada.

En la Fig.1.6 se muestran diferentes funciones utilizadas para representar a las funciones de pertenencia unidimensionales, junto con su expresión analítica [Jang95]. Cada una de ellas queda perfectamente caracterizada a partir de tres parámetros:

- *anchura* ( $2\Delta$ ), que se define como la longitud del intervalo que determinan las abscisas en los puntos en que la función toma el valor 0.5;
- *centro* ( $E_C$ ), que se define como el punto medio de dicho intervalo; y



- *pendiente* ( $\zeta$ ), que se define como el valor absoluto de la pendiente de la función en los puntos de valor 0.5, o puntos de cruce.

Así, para aproximar cualquier función arbitraria, un primer paso consiste en determinar cuántas de estas funciones de pertenencia hay que emplear por cada entrada, y, para cada una de ellas, cuál es el valor que ha de tomar cada uno de estos tres parámetros.

En la Fig.1.7 se representa de forma cualitativa el procedimiento para obtener una función base bidimensional, que corresponde al antecedente de regla para un sistema de dos entradas, a partir de funciones de pertenencia unidimensionales del tipo campana  $s_{1k}(x_1)$  y  $s_{2k}(x_2)$ , en este caso se tiene que  $w_k(\mathbf{x}) = \min\{s_{1k}(x_1), s_{2k}(x_2)\}$ .

A fin de ilustrar el procedimiento de construcción de la superficie de control a partir de la información local representada por las funciones de pertenencia y los valores *singleton*, en la Fig.1.8 se muestra el ejemplo simple de un controlador unidimensional con cuatro reglas. Para este controlador cada una de las cuatro reglas se enuncia

$$\text{"SI } x \text{ es } A_k \text{ ENTONCES } y = y_k^*", \text{ con } 1 \leq k \leq 4$$

y está asociada a cada una de las cuatro etiquetas  $A_k$ , que se han definido en su espacio de entrada. En la parte derecha de la Fig.1.8(b) se muestran las cuatro funciones de pertenencia  $s_k(x)$ , que en este ejemplo aparecen auto-normalizadas [Wang97].

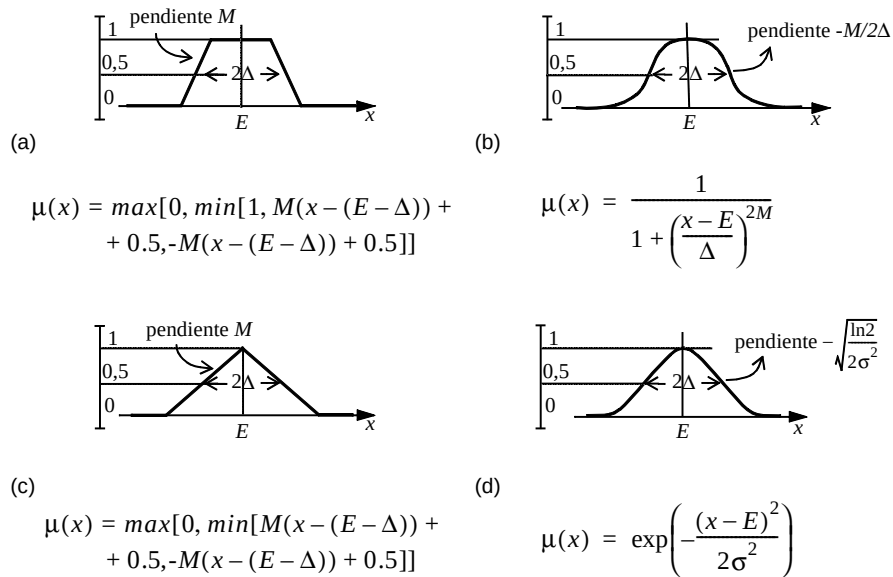


**Fig.1.6 Funciones de pertenencia típicas: trapezoidal (a); campana (b); triangular (c); gaussiana (d).**

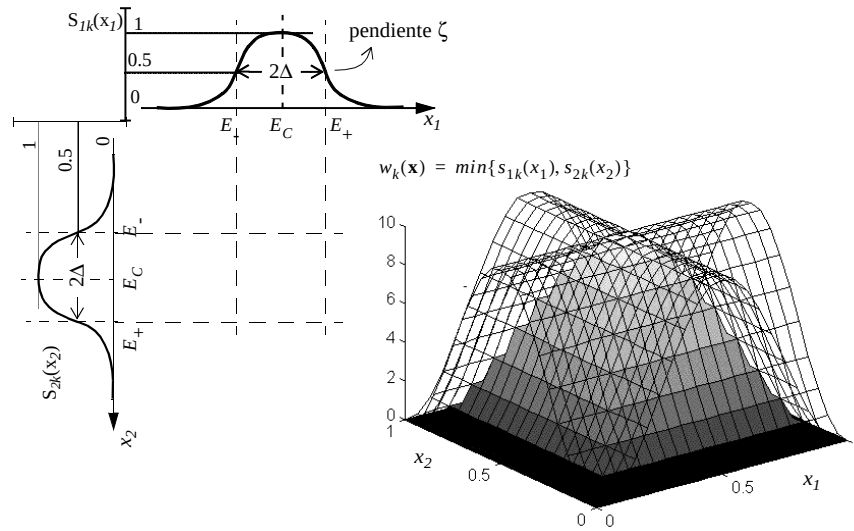


Fig.1.7 Obtención de una función de pertenencia bidimensional.

La validez de cada regla para un valor concreto de la variable de entrada  $x$  se obtiene evaluando la correspondiente función de pertenencia  $s_k(x)$  para esa entrada (Fig.1.8(a)). Si la entrada actual  $x$  corresponde con el centro del intervalo  $I_k$  para la  $k$ -ésima función de pertenencia, entonces  $s_k(x) = 1$  y la salida del sistema viene dada por el valor del  $k$ -ésimo *singleton*  $y = y_k^*$ . Para cualquier punto distinto de los centros de los intervalos de las funciones de pertenencia la salida no coincide con ninguno de los valores *singleton* y se obtiene usando la siguiente expresión,

$$y = y_1^* s_1^*(x) + y_2^* s_2^*(x) + y_3^* s_3^*(x) + y_4^* s_4^*(x) \quad (1.6)$$

donde  $s_k^* = s_k / \left[ \sum_{k=1,4} s_k \right]$ , es la función de pertenencia unidimensional normalizada.

En la Fig.1.8(b) se ilustran diferentes pasos para evaluar las expresión (1.6) y obtener la curva de salida  $y$ . En la parte izquierda se pueden ver las cuatro funciones de pertenencia  $s_k(x)$ , mientras que en la parte derecha se muestra el resultado de ponderar cada una de ellas con su correspondiente valor *singleton*  $s_k^*(x)y_k^*$ , y finalmente la agregación de todas para obtener  $y$ .

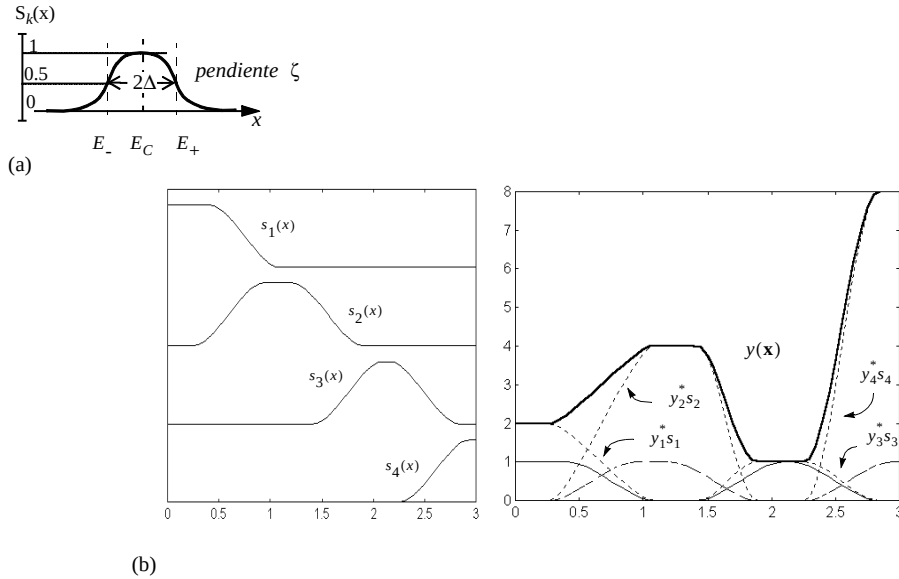


**Fig.1.8** Forma de la función de pertenencia unidimensional (a) ; Ilustración de la aproximación de una función para un controlador difuso unidimensional con *singleton* en el consecuente (b).

#### 4.4. Arquitectura ANFIS

En el contexto de esta tesis, resulta de especial importancia la representación como red adaptativa de un sistema de inferencia difusa que se lleva a cabo con la red denominada ANFIS (Adaptive-Networks Based Fuzzy Inference Systems) [Jang93b], que permite mapear sobre una estructura de red neuronal cualquiera de los tres principales tipos de sistemas de inferencia difusa (Mandani, Tsukamoto, Takagi-Sugeno) descritos en la Fig.1.3. En el caso particular del sistema de Takagi-Sugeno la arquitectura ANFIS se muestra en Fig.1.9 para dos entradas y dos reglas difusas,

SI  $x_1$  es  $A_1$  Y  $x_2$  es  $B_1$ , ENTONCES  $y_1^*$

SI  $x_1$  es  $A_2$  Y  $x_2$  es  $B_2$ , ENTONCES  $y_2^*$

La arquitectura ANFIS correspondiente a estas dos reglas consta de cinco capas, siendo la función asociada a cada nodo:

- **Capa 1:** Los nodos de esta capa tienen asociada la función de pertenencia correspondiente a la etiqueta lingüística del nodo,

$$o_{Ai}^1 = \mu_{A_i}(x_1) \quad , \quad o_{Bi}^1 = \mu_{B_i}(x_2) \quad i = 1, 2 \quad (1.7)$$

- **Capa 2:** La función de este nodo es fija y consiste en realizar la t-norma entre los grados de pertenencia dados por la *Capa 1*, dando el peso de la regla en el proceso de obtención del consecuente. En [Jang93b] se propone el operador producto para la realización de la t-norma, y como tal se asocia a este nodo. Sin embargo, por razones de conveniencia desde el punto de vista de la implementación *hardware*, el operador mínimo se escoge aquí para realizar la t-norma,

$$w_i = \min(\mu_{A_i}(x_1), \mu_{B_i}(x_2)) \quad i = 1, 2 \quad (1.8)$$

La falta de diferenciabilidad de este operador puede resultar un inconveniente en algunos algoritmos de aprendizaje, razón por la cual Jang escoge el producto, pero existen alternativas tanto en el sentido de sustituirlos por operadores pseudo-mínimos derivables [Yuan92], como en el de escoger algoritmos de aprendizaje que no exijan diferenciabilidad [Jabr92].

- **Capa 3:** Esta capa tiene asociada la función,

$$w_i^* = \frac{w_i}{w_1 + w_2} \quad i = 1, 2 \quad (1.9)$$

es decir, da el peso de la regla normalizado con respecto al peso de todas las reglas.

- **Capa 4:** Multiplica el peso normalizado del antecedente de la regla por el valor del singleton asociado al consecuente,

$$o_i^4 = w_i^* y_i^* \quad i = 1, 2 \quad (1.10)$$

- **Capa 5:** Contiene un sólo nodo que agrega los singletons ponderados para obtener la salida del sistema.

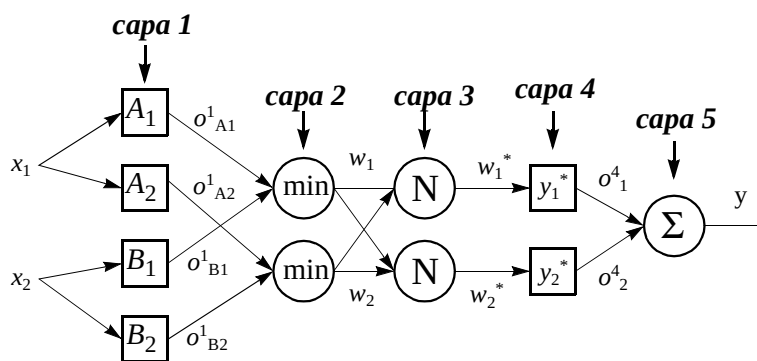


Fig.1.9 Arquitectura ANFIS para un controlador difuso.

A continuación se consideran diversas arquitecturas conceptuales basadas en ANFIS de especial interés para los sistemas objeto de este trabajo, y se analizan bajo el prisma de su posible implementación *hardware*.

#### 4.5. Arquitecturas para Controladores Neuro-Difusos basadas en ANFIS

Las arquitecturas neuro-difusas basadas en la red adaptativa ANFIS, propuesta en [Jang93b], resultan atractivas desde el punto de vista de las aplicaciones e implementaciones *hardware* [Boni95]. La nueva generación de controladores surgida de esta simbiosis [Russ98], combina las habilidades de inferencia de los sistemas difusos con la capacidad de aprender de las redes neuronales, y responde a la demanda de controladores difusos que sean flexibles y capaces de adaptarse a cambios producidos en su entorno de operación, bien por necesidades de ajuste debidos a imprecisiones, o situaciones no contempladas durante su fase de diseño, bien por su aplicación en entornos cambiantes y con cierto grado de incertidumbre [Kand98].

La distribución de las distintas operaciones implicadas en el algoritmo de inferencia difusa en capas y nodos, al estilo de las redes neuronales, algunas de ellas de funcionalidad fija y otros adaptables, propocionan una nueva perspectiva para desarrollar una aproximación de diseño modular, basada en bloques constructivos bien caracterizados, escalables, reconfigurables mediante parámetros controlados eléctricamente, y diseñados para realizar de manera óptima cada una de las tareas asignada a cada nodo de la red, (lo que resulta muy adecuado para la automatización del diseño de este tipo de controladores)[Vida95a].

Por otra parte la estructura de red neuronal permite definir o adaptar los algoritmos de aprendizaje ya existentes, de forma que resulten compatibles con el *hardware*, para el ajuste de esos mismos parámetros, mediante entrenamiento tanto *off-line*, como *on-line* [Vida95b].

A continuación vamos a explorar diferentes arquitecturas neuro-difusas desde el punto de vista de su implementación *hardware*.

##### 4.5.1. Arquitectura General

En la Fig.1.10 se muestra una arquitectura general basada en la arquitectura ANFIS para la realización del algoritmo de Takagi-Sugeno de orden cero recogido en la expresión (1.3). En ella cada una de las cinco capas ocultas realiza una operación del algoritmo difuso,

- evaluación de las funciones de pertenencia,
- calculo de la T-norma multidimensional,
- normalización de reglas,
- ponderación de las reglas, según los valores *singleton*, y
- agregación de reglas ya ponderadas para obtener la salida.

De las cinco capas, tres de ellas, segunda, tercera y quinta, realizan funciones fijas, (operador mínimo, normalización y agregación respectivamente); mientras que la primera, que representa a las funciones de pertenencia, y la cuarta, que corresponde a la ponderación con valores *singleton*, son adaptativas, dependientes de parámetros, y por tanto susceptibles de ser programadas o usadas en tareas de entrenamiento. (Ambas aparecen sobreimpresas con una flecha en la Fig.1.10 para destacar esta cualidad). Como ya se ha visto anteriormente hace falta especificar tres parámetros, (ancho ( $2\Delta$ ), centro ( $E_C$ ) y pendiente ( $\zeta$ )) para caracterizar cada función de pertenencia y por tanto, a cada nodo de la primera capa; y un valor singleton ( $y_k^*$ ) para cada nodo de la cuarta capa.

La arquitectura de la Fig.1.10 es general en el sentido de que es capaz de representar a cualquier controlador neuro-difuso que realice el sistema de inferencia Takagi-Sugeno de orden cero, cualquiera que sea el número de entradas ( $M$ ), cualquiera que sea el número de reglas ( $N$ ), y cualquiera que sea la forma de los antecedentes de regla; y por tanto, cualquiera que sea la partición del universo de discurso sobre la que esté definido (ver Fig.1.5). Para ello basta con definir adecuadamente el conjunto completo de parámetros que caracterizan a los nodos adaptativos.

Para realizar un controlador genérico siguiendo esta arquitectura serán necesarios:

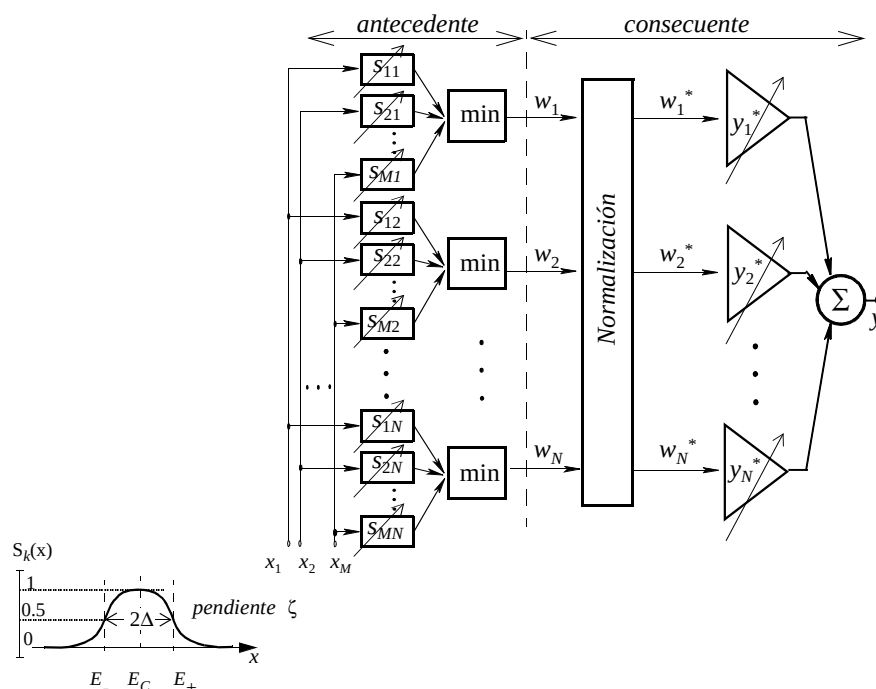


**Fig.1.10** Arquitectura ANFIS general para un controlador neuro-difuso que realiza el algoritmo de Takagi-Sugeno de orden cero.

- $M \times N$  bloques de función de pertenencia.
- $N$  bloques de operador mínimo de  $M$  entradas cada uno.
- Un bloque normalizador de  $N$  entradas.
- $N$  bloques de ponderación de *singleton*.
- Un bloque de agregación de  $N$  entradas.

Y será necesario almacenar, determinar y ajustar:

- $(M \times N) \times 3$  parámetros de funciones de pertenencia.
- $N$  valores *singleton*.

La realización de esta arquitectura en un circuito integrado puede resultar costosa en cuanto a ocupación de silicio y consumo de potencia, incluso para valores moderados de  $M$  y  $N$ , principalmente por el importante número de bloques de función de pertenencia a que da lugar. Además el controlador así obtenido puede resultar complejo a la hora de ser configurado, debido sobre todo al elevado número de parámetros que hay que determinar y ajustar.

#### 4.5.2. Arquitectura para Particiones en Rejilla del Espacio de Entrada.

En muchas de las aplicaciones de control es bastante habitual el empleo de particiones en rejilla [Brow94]. Este tipo de partición resulta simple porque requiere un número pequeño de funciones de pertenencia por entrada y da lugar a reglas lingüísticas interpretables en términos sencillos. Ambas cualidades resultan muy interesantes para los problemas de control donde se suele emplear como entradas un número pequeño de variables de estado y la estrategia de control se establece en base a reglas relativas a la acción a tomar en función de la magnitud de la desviación de esas variables respecto su valor para la trayectoria de control objetivo. Vamos a analizar a continuación la arquitectura que resulta de considerar esta situación.

En términos generales, una partición en rejilla tiene la potencial desventaja de que el número de reglas necesarias para realizar una buena aproximación de una superficie de control puede llegar a ser prohibitivamente grande, incluso para un número de entradas no muy elevado. Por otra parte, una dificultad adicional se deriva del hecho de que la granularidad<sup>3†</sup> necesaria para aproximar una función en una región del espacio de entradas, puede resultar inapropiada en otras. Sin embargo, la importante reducción en cuanto a número de bloques y parámetros de ajuste que supone esta elección, frente a la arquitectura general, la hace más atractiva desde la perspectiva de integración *hardware*, sobre todo para el diseño de sistemas en los que se requiera una complejidad media, o baja.

En la Fig.1.10 se muestra una arquitectura ANFIS para un controlador neuro-difuso que realiza el algoritmo de Takagi-Sugeno de orden cero según la expresión (1.3) sobre una

---

<sup>3†</sup>. En este contexto por granularidad, se entiende el tamaño de los intervalos que definen la partición en rejilla.

partición en rejilla del espacio de entrada. Por simplicidad aquí se ha considerado el caso en que para cada una de las  $M$  entradas se emplea el mismo número de funciones de pertenencia  $L^{4\ddagger}$ , y por tanto el número de reglas  $N$  resulta ser  $L^M$ . En esta arquitectura, la distribución de operaciones en las cinco capas es la misma que en el caso general. La diferencia con la arquitectura anterior aparece en la capa uno, en la que como consecuencia de la partición realizada, cada función de pertenencia es compartida por  $L^{M-1}$  reglas, lo que supone un importante ahorro en el número de bloques necesarios para su realización.

Para realizar un controlador genérico siguiendo esta arquitectura serán necesarios:

- $M \times L$  bloques de función de pertenencia.
- $N$  bloques de operador mínimo de  $M$  entradas cada uno.
- Un bloque normalizador de  $N$  entradas.
- $N$  bloques de ponderación de *singleton*.
- Un bloque de agregación de  $N$  entradas.

Y será necesario almacenar, determinar y ajustar:

- $(M \times L) \times 3$  parámetros de funciones de pertenencia.
- $N$  valores *singleton*.

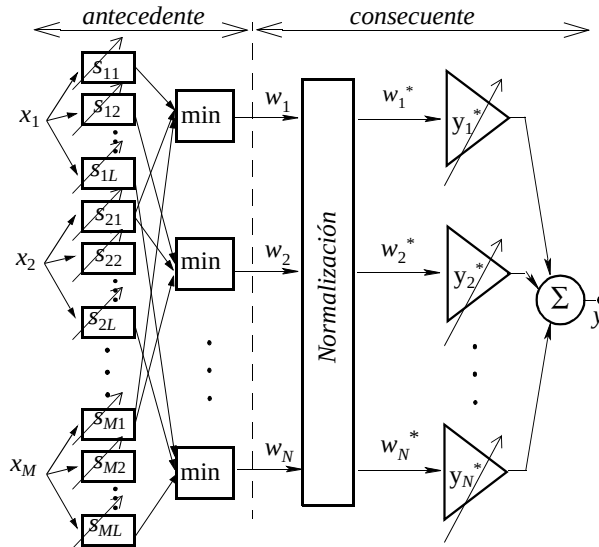


**Fig.1.11** Arquitectura ANFIS para un controlador neuro-difuso que realiza el algoritmo de Takagi-Sugeno de orden cero sobre una partición en rejilla del espacio de entrada.

<sup>4†</sup>. Un caso más general sería considerar que el número de etiquetas fuese distinto para cada entrada,

$$L_{x_k} \quad k = 1 \dots M \quad , \text{y por tanto } N = \prod_{k=1}^M L_{x_k}$$



La implementación de esta arquitectura resulta, en general, menos costosa que la anterior al reducirse de una manera importante en número de bloques de función de pertenencia a realizar, pues se pasa de las  $M \times N$  del caso general, a las  $M \times \log_M(N)$  de esta arquitectura. El número de parámetros a determinar y ajustar también se ve reducido de manera importante, al reducirse de  $(M \times N) \times 3$  a  $(M \times L) \times 3$ . Este número puede reducirse si se considera una partición en rejilla fija, en la que no se permite variar la posición de las funciones de pertenencia; e incluso más aún, si la partición es fija y uniforme, en la que además se considera que el ancho de todas las funciones de pertenencia es el mismo. En este caso solamente es necesario conocer un parámetro por entrada, para caracterizar a todas las funciones de pertenencia. Cualquiera de estas dos últimas opciones no limita la capacidad de aproximación del sistema de inferencia siempre que se aumente la granularidad de la partición, aumentando  $L$ , para un tamaño dado del espacio de entrada, [Wang97] [Zeng96a]. Sin embargo, un aumento del valor de  $L$  conlleva un incremento en el valor del número total de reglas  $N$ , por lo que se ha de establecer un compromiso entre complejidad del sistema y su capacidad de aproximación.

Por otra parte la complejidad abordable por una realización *hardware* particular también está limitada por factores tecnológicos y de diseño, como ocurre en el caso de los controladores analógicos. En este tipo de realizaciones de los algoritmos difusos aparecen nudos de computación global, (como los que aparecen al realizar el proceso de *defuzzificación* empleando el método del centro de gravedad), en los que se suman también los errores (sistemáticos o aleatorios) causados por los circuitos que realizan los antecedentes de regla, de modo que el error global final se hace mayor, lo que supone, entre otras cosas una degradación en su precisión. Además, en estos nudos también se acumulan las capacidades parásitas, lo que afecta negativamente a la velocidad de operación. Por tanto, un incremento en la necesidad de complejidad del sistema difuso, lleva a una degradación de las prestaciones, en cuanto a precisión, velocidad, consumo y ocupación de área de silicio, que ofrece su realización mediante sistemas analógicos.

## 5. TEORIA DE LA APROXIMACIÓN EN LOS SISTEMAS DIFUSOS. PRECISIÓN DE LA APROXIMACIÓN Y PROPIEDAD DE DESCOMPOSICIÓN

Una reflexión sobre el modo en que funciona el algoritmo de inferencia en el contexto de la aproximación de funciones proporciona un punto de vista diferente, con nuevas ideas y conceptos que permiten avanzar en la búsqueda de nuevas posibilidades de arquitectura para los controladores difusos. De especial importancia para la propuesta desarrollada en esta tesis resulta la *propiedad de descomposición de los sistemas difusos*, junto con la generalización del concepto de *sistema difuso más simple posible* propuestos en [Zeng96b]. Ambos son

enunciados y utilizados en varias aplicaciones por los autores en el contexto del análisis de sistema difusos complejos. En el siguiente capítulo se mostrará cómo estos resultados también pueden ser aprovechados en tareas de diseño para definir una nueva arquitectura dinámica que permite extender la complejidad abordable por los sistemas analógicos que realizan sistemas de control difuso definidos sobre particiones en rejilla, salvando sus principales dificultades. Ambas cuestiones, capacidad de aproximación y propiedad de descomposición de los sistemas difusos son abordadas de una forma general en este apartado.

### 5.1. Teoría de la aproximación en los sistemas difusos

Hay cierta clase de sistemas difusos que admiten ser descritos con fórmulas matemáticas detalladas. Entre ellos se encuentran aquellos que emplean en el proceso de *defuzzificación* el método del centroide. (Este es el caso del sistema de Takagi-Sugeno con *singleton* en el consecuente considerado en la expresión (1.3)). Para estos sistemas es posible escribir su salida como una combinación lineal de funciones base, en la forma

$$y = f(\mathbf{x}) = \sum_{k=1, N} \bar{y}_k^* B_k^*(\mathbf{x}) \quad (1.11)$$

donde  $f: U \subset R^M \rightarrow V \subset R$ ,  $(U = U_1 \times U_2 \times \dots \times U_M) \subset R^M$  es el espacio de entrada y  $V \subset R$  el espacio de salida, y  $\mathbf{x} = \{x_1, x_2, \dots, x_M\}^T$ , tal que  $\mathbf{x} \in U$  es un vector de entrada.

En la expresión (1.11), cada coeficiente  $\bar{y}_k^* \in V$ , es un valor real que corresponde al centro de la función de pertenencia asociada al consecuente de la  $k$ -ésima regla difusa, mientras que cada  $B_k^*(\mathbf{x})$  es una función base multidimensional normalizada, que se obtiene a partir del antecedente de regla, y se calcula según la expresión,

$$B_k^*(\mathbf{x}) = \frac{B_k(\mathbf{x})}{\sum_{k=1, N} B_k(\mathbf{x})} \quad (1.12)$$

donde  $B_k(\mathbf{x})$  corresponde con la salida del antecedente de la  $k$ -ésima regla difusa y se obtiene a partir de las entradas según la expresión,

$$B_k(\mathbf{x}) = \text{T-norm}\{s_{1k}(x_1), s_{2k}(x_2), \dots, s_{Mk}(x_M)\} \quad (1.13)$$

En esta expresión  $\text{T-norm}\{\bullet\}$  puede ser, bien el operador mínimo, bien el operador producto, mientras que  $s_{ik}(x_i)$ , son las funciones de pertenencia sobre las que se evalúan el grado de coincidencia entre las entradas y las etiquetas lingüísticas difusas.

El proceso de diseño de estos sistemas difusos consiste en determinar tanto los valores  $\bar{y}_k^*$ , como las características que definen a las funciones base  $B_k(\mathbf{x})$  para cumplir con el

objetivo de control. La expresión (1.11), que define la salida del sistema, es una fórmula detallada del mapeado no lineal que realiza el sistema difuso, por tanto, para este tipo de sistemas, el problema de diseño puede plantearse en general como un problema de aproximación de cierta superficie de control  $g(\mathbf{x})$ , definida tal que  $g : U \subset R^M \rightarrow V \subset R$ , con un grado de precisión dado  $\varepsilon \in R | \varepsilon > 0$ .

En este contexto el problema de diseño presenta diversos aspectos entre los que destacan:

- Determinar la capacidad de aproximación de los diferentes tipos de sistemas difusos.
- Proporcionar una medida de la bondad de la aproximación alcanzable con las diferentes implementaciones concretas de los diversos tipos de sistemas difusos.
- Proporcionar un método de diseño que permita encontrar los valores  $\bar{y}_k^*$  y las funciones base  $B_k(\mathbf{x})$  óptimos para alcanzar la precisión deseada en la aproximación.

Los dos primeros garantizan el éxito en la tarea de diseño, en la medida en que se demuestra que es posible diseñar un sistema difuso  $f(\mathbf{x})$ , capaz de aproximar una función arbitraria  $g(\mathbf{x})$ , con un grado de precisión también arbitrario, y en la de que es posible proporcionar una medida del error que se asume, y por tanto acotarlo, para una aproximación dada. Sin embargo, nada aportan sobre la manera de obtener dicho sistema. Encontrar el sistema óptimo resulta en general más difícil que demostrar su existencia, y los diferentes métodos para lograrlo dependerán de la información disponible sobre  $g(\mathbf{x})$ . En un problema práctico se presentan dos situaciones:

- Es posible generar arbitrariamente parejas de datos entrada salida  $((\mathbf{x}^k, g(\mathbf{x}^k)))$ . Bien porque se tiene una expresión analítica conocida para la función  $g(\mathbf{x})$ , o bien porque, aunque no se conoce la expresión analítica de  $g(\mathbf{x})$ , es posible tener acceso al sistema de forma que, para cualquier vector de entrada  $\mathbf{x}$ , es posible medir su respuesta.
- Sólo se conoce un número limitado de parejas de datos entrada salida  $((\mathbf{x}^k, g(\mathbf{x}^k)))$ , que no pueden ser escogidas arbitrariamente.

La primera de ellas, aunque bastante realista, no es el habitual en la mayoría de los casos prácticos. Sin embargo su análisis resulta de mucha utilidad porque permite definir una metodología precisa de diseño y determinar expresiones para caracterizar la precisión alcanzable con cada una de las diversas alternativas que resultan de considerar los diferentes tipos de sistemas difusos.

El segundo caso es el más habitual en la práctica, en la que, en la mayoría de las situaciones, sólo se posee información recolectada a partir de medidas, en general no

arbitrarias, sobre el proceso a controlar. En este caso, aunque la metodología anterior no sea aplicable directamente, puede ser empleada como punto de partida para un posterior reajuste; y siempre que el conjunto de parejas de datos resulte representativo del fenómeno global.

Los tres aspectos de diseño de sistemas difusos que pueden ser representados como una combinación lineal de funciones base, en el contexto de aproximación de funciones, arriba mencionados, son abordados en conjunto en los trabajos de [Zeng94] [Zeng95]. Tomando como punto de partida diversas propiedades de las funciones base, puestas de manifiesto como resultado de un análisis sistemático y detallado de las mismas, los autores obtienen las propiedades básicas de aproximación de los sistemas difusos, que revelan los fundamentos de ese mecanismo de aproximación, y formalizan unas cuantas ideas básicas que ya eran empleadas habitualmente, y de forma intuitiva, por los diseñadores.

Debido al carácter constructivo de sus demostraciones, también proporcionan un método para construir un sistema difuso, con el grado de precisión requerido, para el caso en que es posible generar arbitrariamente parejas de datos entrada salida  $(\mathbf{x}^k, g(\mathbf{x}^k))$ .

Por otra parte, en [Zeng96a] se analizan y proporcionan fórmulas que determinan los límites del error de aproximación para diferentes tipos de sistemas difusos, esto es, sistemas difusos generados a partir de diferentes métodos de inferencia o *defuzzificación*.

En relación con los trabajos antes citados y de especial importancia para el trabajo desarrollado en esta tesis resulta la *propiedad de descomposición de los sistemas difusos* propuesta en [Zeng96b]. En este trabajo los autores presentan dicha propiedad utilizando un procedimiento de descomposición simple y constructivo que permite dividir el espacio de entrada en diversos subespacios de entrada, y cualquier sistema difuso en subsistemas difusos del tipo más simple posible sobre cada subespacio de entrada. El concepto de *sistema difuso más simple posible* es definido por los propios autores como una extensión de la definición inicialmente propuesta en [Ying90][Ying93] para sistemas con dos entradas y una salida. Aunque esta propiedad de descomposición es enunciada y utilizada en varias aplicaciones por los autores en el contexto del análisis de sistemas difusos complejos, veremos como también puede ser aprovechada en tareas de diseño. A continuación se resumen los principales resultados.

## 5.2. Propiedades de aproximación de los sistemas difusos.

En primer lugar consideramos algunas definiciones y propiedades de las funciones de pertenencia de uso común en el diseño de sistemas difusos. Por simplicidad en la notación, en este apartado son introducidas para el caso de funciones de pertenencia unidimensionales, pero pueden ser enunciadas para el caso general multidimensional [Zeng96b].

- Sea  $[a, d] \subset R$ . Una función de pertenencia asociada a un conjunto difuso  $A$ , es de tipo *Pseudo-Trapezoidal*, si es una función continua de  $R$  y puede expresarse

$$s_A(x; a, b, c, d, H) = \begin{cases} I(x), & x \in [a, b) \\ H, & x \in [b, c] \\ D(x), & x \in (c, d] \\ 0, & x \in R - (a, d) \end{cases} \quad (1.14)$$

donde  $a \leq b \leq c \leq d$ ,  $a < d$ ,  $0 < H \leq 1$ , y donde  $0 \leq I(x) \leq 1$  es una función no decreciente en el intervalo semicerrado  $[a, b)$  y donde  $0 \leq D(x) \leq 1$  es una función no creciente en el intervalo semicerrado  $(c, d]$ . Cuando  $H = 1$  se dice que  $A$  está normalizado y entonces se escribe  $s_A(x; a, b, c, d)$ .

Todas las funciones de pertenencia de la Fig.1.6 son de este tipo, por lo que las funciones Pseudo-Trapezoidales constituyen una familia amplia y muy usada en aplicaciones de control.

- Dado un conjunto de conjuntos difusos  $A^1, A^2, \dots, A^L$  en  $W \subset R$ , se dice que es *Completo* en  $W$ , si para cada  $x \in W$ , existe un conjunto difuso  $A^j$  tal que  $s_{A^j}(x) > 0$ .
- Dado un conjunto de conjuntos difusos  $A^1, A^2, \dots, A^L$  en  $W \subset R$ , se dice que es *Consistente* en  $W$ , si  $\forall x \in W$  tal que  $s_{A^i}(x) = 1$  esto implica que  $s_{A^j}(x) = 0$ ,  $\forall i \neq j$ .
- Dados dos conjuntos difusos  $A^1, A^2$  en  $W \subset R$ , se dice que  $A^1 > A^2$ , si  $M(A^1) > M(A^2)$ , siendo  $M(A^i) = \{x \in W | s_{A^i}(x) = \sup_{x' \in W} [s_{A^i}(x')]\}$ . Para el caso en que  $s_{A^i}(x; a, b, c, d)$ , es pseudo-trapezoidal y está normalizada, se tiene que  $M(A^i) = [b, c]$ .
- Dado  $W \subset R$ , siempre es posible escoger un conjunto de conjuntos difusos  $A^1, A^2, \dots, A^L$  normalizado, consistente y completo, con funciones de pertenencia de tipo pseudo-trapezoidal  $s_{A^i}(x; a_i, b_i, c_i, d_i)$ , tal que  $A^1 < A^2 < \dots < A^L$ . Además se cumple que  $c_i \leq a_{i+1} \leq d_i \leq b_{i+1}$ ,  $\forall i = 1, 2, \dots, (L-1)$ . Si además  $c_i = a_{i+1}$  y  $d_i = b_{i+1}$  y para cada intervalo  $[c_i, b_{i+1}]$ ,  $\forall i = 1, 2, \dots, (L-1)$  se cumple  $s_{A^i}(x; a_i, b_i, c_i, d_i) = 1 - s_{A^{i+1}}(x; a_{i+1}, b_{i+1}, c_{i+1}, d_{i+1})$ , se dice que dicho conjunto está autonormalizado.

En la Fig.1.12 se ilustran las anteriores definiciones y propiedades con tres ejemplos en los que aparece un conjunto de cuatro conjuntos difusos que emplean funciones de pertenencia pseudotrapezoidales,  $s_{A^i}(x; a_i, b_i, c_i, d_i)$ , para  $i = 1, 2, 3, 4$  y  $x \in W$ , normalizados. En los tres ejemplos se cumple que  $A^1 < A^2 < A^3 < A^4$ . En la Fig.1.12(a) se muestra el ejemplo de un conjunto que no es completo, ni consistente. No es completo porque ninguna de las cuatro funciones de pertenencia permite evaluar a los puntos del intervalo abierto  $(d_2, a_3) \subset W$ . No es

consistente porque  $\forall x \in [a_2, c_1], s_{A^1}(x; a_1, b_1, c_1, d_1) = 1$  y además  $s_{A^2}(x; a_2, b_2, c_2, d_2) \neq 0$ ; lo mismo ocurre  $\forall x \in [b_2, d_1]$ , donde se tiene que  $s_{A^2}(x; a_2, b_2, c_2, d_2) = 1$  y además  $s_{A^1}(x; a_1, b_1, c_1, d_1) \neq 0$ . En el ejemplo de la Fig.1.12(b) el conjunto etiquetas considerado es completo y consistente, y es fácil verificar que se cumple que  $c_i \leq a_{i+1} \leq d_i \leq b_{i+1}$ , para  $i = 1, 2, 3, 4$ . Finalmente en la Fig.1.12(c) se muestra un conjunto consistente, completo y autonormalizado, donde  $c_i = a_{i+1}$  y  $d_i = b_{i+1}$  con  $i = 1, 2, 3, 4$ , y para cada intervalo  $[c_i, b_{i+1}]$ , se cumple  $s_{A^i}(x; a_i, b_i, c_i, d_i) = 1 - s_{A^{i+1}}(x; a_{i+1}, b_{i+1}, c_{i+1}, d_{i+1})$ .

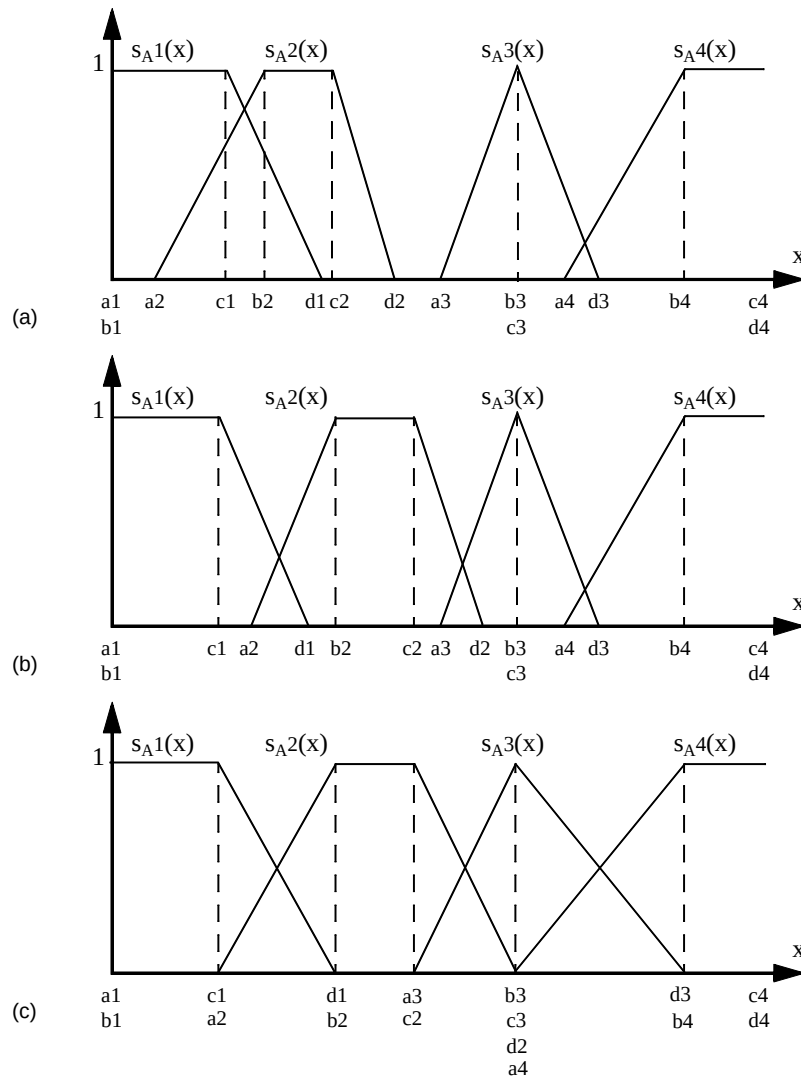


Fig.1.12 Ejemplos de conjuntos difusos con funciones de pertenencia pseudo-trapezoidales: (a) Conjunto no completo y no consistente; (b) conjunto completo y consistente; (c) conjunto completo, consistente y autonormalizado.

A continuación se resumen los principales resultados de la teoría de aproximación en los sistemas difusos, expuestos en [Zeng94] [Zeng95] y [Zeng96a] son los siguientes:

- a) *Capacidad de Aproximación.* Cualquier sistema difuso que pueda ser descrito según la expresión (1.11) puede ser considerado como un aproximador universal de funciones. Esto es, dada cualquier función continua  $g(\mathbf{x})$ , definida tal que  $g: U \subset R^M \rightarrow V \subset R$ , y cualquier número real  $\varepsilon > 0$ , existe un sistema difuso  $f(\mathbf{x})$ , expresado según (1.11), tal que  $\sup_{\mathbf{x} \in U} |g(\mathbf{x}) - f(\mathbf{x})| < \varepsilon$ .
- b) *Método de diseño.* El sistema difuso  $f(\mathbf{x})$  definido como  $f: U \subset R^M \rightarrow V \subset R$ , cuya existencia afirma el resultado a) puede ser construido de la siguiente manera, con la única condición de que sea posible generar arbitrariamente parejas de datos entrada salida  $((\mathbf{x}^k, g(\mathbf{x}^k)))$ :

- Por cada entrada  $x_i \in U_i \equiv [\alpha_i, \beta_i]$  se definen  $L_i$  conjuntos difusos  $A_i^1, A_i^2, \dots, A_i^{L_i}$ , que están normalizados, y son consistentes y completos, siendo  $s_{A_i^j}(x; a_i^j, b_i^j, c_i^j, d_i^j)$  su correspondiente función de pertenencia de tipo trapezoidal, con  $j = 1, 2, \dots, L_i$ . con  $a_i^1 = b_i^1 = \alpha_i$  y  $a_i^{L_i} = b_i^{L_i} = \beta_i$ . Además dicho conjunto se elige tal que  $A_i^1 < A_i^2 < \dots < A_i^{L_i}$ .
- Por cada entrada  $x_i \in U_i \equiv [\alpha_i, \beta_i]$  se definen también los puntos  $e_i^j$  con  $(j = 1, 2, \dots, L_i)$  de forma que  $e_i^1 = \alpha_i$ ,  $e_i^{L_i} = \beta_i$  y  $e_i^j = \frac{1}{2}(b_i^j + c_i^j)$ , para  $j = 2, \dots, (L_i - 1)$ .
- Las anteriores definiciones determinan una partición en rejilla  $U$ , y definen un sistema difuso de  $N = \prod_{i=1}^M L_i$  reglas que pueden ser enunciadas en la forma

$$R_{j_1 \dots j_M}: \text{SI } x_1 \text{ es } A_1^{j_1} \quad \text{Y} \dots \quad \text{Y } x_M \text{ es } A_M^{j_M} \quad \text{ENTONCES } y \text{ es } C_{j_1 \dots j_M} \quad (1.15)$$

donde  $(j_i = 1, 2, \dots, L_i)$  e  $\forall i = 1, 2, \dots, M$  y donde  $C_{j_1 \dots j_M}$  es la función de pertenencia asociada al consecuente de dicha regla y cuyo centro denotado por  $y_{j_1 \dots j_M}$  se escoge de manera que se cumpla

$$\bar{y}_{j_1 \dots j_M}^* = g(e_1^{j_1}, \dots, e_M^{j_M}). \quad (1.16)$$

- El sistema completo puede escribirse de forma análoga a la expresión (1.11)

$$y = f(\mathbf{x}) = \sum_{N \text{ términos}} \bar{y}_{j_1 \dots j_M}^* B_{j_1 \dots j_M}^*(\mathbf{x}) \quad (1.17)$$

$$\text{con } (j_i = 1, 2, \dots, L_i) e \quad i = 1, 2, \dots, M$$

donde  $\bar{y}_{j_1 \dots j_M}^*$  está definido en (1.16) y donde  $B_{j_1 \dots j_M}^*(\mathbf{x})$  denota cada una de las funciones base normalizada, que se obtienen según la expresión,

$$B_{j_1 \dots j_M}^*(\mathbf{x}) = \frac{B_{j_1 \dots j_M}(\mathbf{x})}{\sum_{N \text{ términos}} B_{j_1 \dots j_M}(\mathbf{x})} \quad (1.18)$$

$$\text{con } (j_i = 1, 2, \dots, L_i) e \quad i = 1, 2, \dots, M$$

y donde  $B_{j_1 \dots j_M}(\mathbf{x})$  se calcula a partir de las funciones de pertenencia  $s_{A_{j_i}}(x_i)$  según la expresión,

$$B_{j_1 \dots j_M}(\mathbf{x}) = \text{T-norm} \left\{ s_{A_{j_1}}(x_1), \dots, s_{A_{j_M}}(x_M) \right\} \quad (1.19)$$

En esta expresión  $\text{T-norm}\{ \bullet \}$  puede ser, bien el operador mínimo, bien el operador producto. Como los conjuntos difusos  $A_i^1, A_i^2, \dots, A_i^{L_i}$  son completos para cada vector  $\mathbf{x} \in U$  existirá un conjunto de índices  $j_1 \dots j_M$  tal que  $B_{j_1 \dots j_M}(\mathbf{x}) \neq 0$ , por lo que tanto las expresiones (1.18), como (1.17) estarán siempre bien definidas.

En la Fig.1.13 se muestra un ejemplo, para un caso bidimensional, de la partición que realiza el sistema difuso  $f(\mathbf{x})$  descrito en el apartado b), y que corresponde a una partición en rejilla.

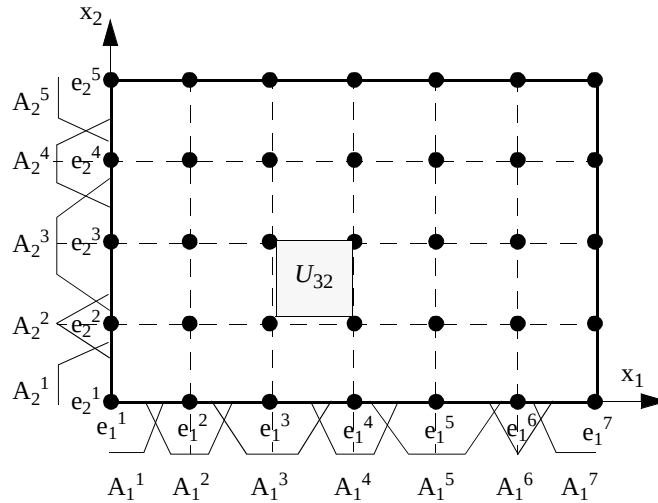


**Fig.1.13 Ejemplo de la partición del espacio de entrada que realiza el sistema difuso  $f(\mathbf{x})$ , para el caso bidimensional.**



- c) *Precisión en la aproximación, aproximador de primer orden.* Un sistema difuso genérico  $f(\mathbf{x})$ , como el definido en la expresión (1.17), es un aproximador de primer orden para la función  $g(\mathbf{x})$  del apartado a), y que aparece en (1.16), lo que puede expresarse así,

$$\|g - f\|_{\infty} \leq \sum_{i=1}^M \left\| \frac{\partial g}{\partial x_i} \right\|_{\infty} h_i \leq h \sum_{i=1}^M \left\| \frac{\partial g}{\partial x_i} \right\|_{\infty} \quad (1.20)$$

donde  $\|\bullet\|_{\infty}$  denota la norma del supremo definida como  $\|d(\mathbf{x})\|_{\infty} = \sup_{\mathbf{x} \in U} (|d(\mathbf{x})|)$ , mientras que  $h_i = \max_{1 \leq j \leq L_i-1} (|e_i^{j+1} - e_i^j|)$  y  $h = \sup(h_i)$  con  $i = 1, 2, \dots, M$ . Al número  $h$  se le denomina módulo de la partición realizada en  $U$ .

Este resultado es importante y permite sacar interesantes conclusiones:

- La expresión (1.20) verifica el resultado a). Debido a que  $\forall i = 1, 2, \dots, M$ ,  $\left\| \frac{\partial g}{\partial x_i} \right\|_{\infty}$  es un número finito, para cualquier  $\varepsilon > 0$  puede escogerse un  $h$  tal que se verifique la desigualdad  $h \sum_{i=1}^M \left\| \frac{\partial g}{\partial x_i} \right\|_{\infty} < \varepsilon$ .
  - De la expresión (1.20) y de la definición de  $h_i$  también se deduce que, mientras más conjuntos difusos se definan, (mayor sea la granularidad de la partición que se realice), mejor será la aproximación que realiza  $f(\mathbf{x})$ ; confirmando así la intuición de diseño.
  - Para diseñar un sistema con un grado de precisión deseado, según esta aproximación es necesario poder obtener los valores  $g(\mathbf{x}_{j_1 \dots j_M})$  para los vectores  $\mathbf{x}_{j_1 \dots j_M} = (e_1^{j_1}, \dots, e_M^{j_M})$ , con  $(j_i = 1, 2, \dots, L_i) e i = 1, 2, \dots, M$ , y además conocer una cota de cada una de las derivadas parciales  $\left\| \frac{\partial g}{\partial x_i} \right\|_{\infty}$ .
  - Para cualquier vector  $\mathbf{x}_{j_1 \dots j_M}$  se verifica la igualdad  $g(\mathbf{x}_{j_1 \dots j_M}) = f(\mathbf{x}_{j_1 \dots j_M})$ . Lo que corresponde con la idea intuitiva de que el sistema difuso, es un sistema que es capaz de interpolar los valores de  $g(\mathbf{x})$  para cualquier  $\mathbf{x} \in U$ , a partir de las muestras  $g(\mathbf{x}_{j_1 \dots j_M})$ .
  - El uso del criterio que establece la expresión (1.20) da lugar por lo general a sistemas que necesitan un número elevado de reglas, incluso para aproximar funciones relativamente simples, por lo que sería deseable mejorar ese resultado.
- d) *Precisión en la aproximación, aproximador de segundo orden.* Un sistema difuso  $f(\mathbf{x})$ , como el definido en la expresión (1.17), que emplee funciones de pertenencia

triangulares y utilice el operador producto como T-norma, se comporta como un aproximador de segundo orden para funciones del tipo  $g(\mathbf{x})$  del apartado a) que admitan segunda derivada. En este caso puede escribirse

$$\|g - f\|_{\infty} \leq \frac{1}{8} \sum_{i=1}^M \left\| \frac{\partial^2 g}{\partial x_i^2} \right\|_{\infty} h_i^2 \leq \frac{h}{8} \sum_{i=1}^M \left\| \frac{\partial^2 g}{\partial x_i^2} \right\|_{\infty} \quad (1.21)$$

donde  $\|\bullet\|_{\infty}$  denota la norma del supremo definida como  $\|d(\mathbf{x})\|_{\infty} = \sup_{\mathbf{x} \in U} (|d(\mathbf{x})|)$ , mientras que  $h_i = \max_{1 \leq j \leq L_i-1} (|e_i^{j+1} - e_i^j|)$  y  $h = \sup(h_i)$  con  $i = 1, 2, \dots, M$ . Al número  $h$  se le denomina módulo de la partición realizada en  $U$ . El uso de este criterio da lugar a sistemas que necesitan un menor número de reglas, que las que resultan de emplear el criterio dado por la expresión (1.20).

### 5.3. Propiedad de descomposición de los sistemas difusos.

El resultado fundamental presentado en [Zeng96b] es la *propiedad de descomposición de los sistemas difusos* que establece que dividiendo el espacio de entrada de forma apropiada en diferentes subespacios, cualquier sistema difuso que pueda expresarse como combinación lineal de funciones base, según la fórmula (1.11), puede ser descompuesto en varios subsistemas difusos, cada uno de los cuales constituye el *sistema difuso más simple posible* en cada uno de los subespacios de entrada. Dada la naturaleza constructiva de la demostración que realizan los autores, ésta proporciona el método para realizar la partición del espacio de entrada y la expresión de cada uno de los subsistemas difusos. A continuación se resumen las principales definiciones y resultados:

- a) *Sistema difuso más simple posible*. Un sistema difuso  $f^s(\mathbf{x})$  tal que  $f^s: U \subset R^M \rightarrow V \subset R$ , donde  $(U = U_1 \times U_2 \times \dots \times U_M) \subset R^M$  es el espacio de entrada y  $V \subset R$  el espacio de salida, es el sistema difuso más simple posible en  $U$  si para cada  $U_i$  (con  $i = 1, 2, \dots, M$ ) sólo están definidos dos conjuntos difusos  $A_i^1, A_i^2$ . Por tanto, para este sistema difuso se tienen  $N = 2^M$  reglas que pueden ser enunciadas en la forma

$$R_{j_1 \dots j_M}: \text{SI } x_1 \text{ es } A_1^{j_1} \quad \text{Y} \dots \quad \text{Y } x_M \text{ es } A_M^{j_M} \quad \text{ENTONCES } y \text{ es } C_{j_1 \dots j_M} \quad (1.22)$$

donde  $(j_i = 1, 2)$  e  $\forall i = 1, 2, \dots, M$  y donde  $C_{j_1 \dots j_M}$  es la función de pertenencia asociada al consecuente de dicha regla y cuyo centro es denotado por  $y_{j_1 \dots j_M}$ .

- b) Empleando el concepto de *Sistema difuso más simple posible* definido en a) y suponiendo que el sistema general considerado  $f(\mathbf{x})$  ha sido diseñado siguiendo el método descrito en 5.2.(b) y por tanto puede expresarse según (1.17), el procedimiento de descomposición se desarrolla según los siguientes pasos :

1) *Descomposición del espacio de entrada*. Dada la partición en rejilla empleada para el diseño de  $f(\mathbf{x})$ , el espacio de entrada  $U$  puede obtenerse según la expresión

$$U = \bigcup_{C \text{ términos}} U_{k_1 \dots k_M} \quad (1.23)$$

$$\text{con}(k_i = 1, 2, \dots, (L_i - 1)) e \quad i = 1, 2, \dots, M$$

donde  $C = \prod_{i=1}^M (L_i - 1)$  es el número total de subespacios de entrada  $U_{k_1 \dots k_M}$ , cada uno de los cuales queda definido según la expresión

$$U_{k_1 \dots k_M} = [e_1^{j_1}, e_1^{j_1+1}] \times [e_2^{j_2}, e_2^{j_2+1}] \times \dots \times [e_M^{j_M}, e_M^{j_M+1}] \quad (1.24)$$

donde  $e_i^{j_i}$  ( $\forall j_i = 1, 2, \dots, L_i; \forall i = 1, 2, \dots, M$ ) corresponde a cada uno de los puntos que definen la partición en el espacio de entrada de cada variable  $x_i$ , y donde para cada subíndice  $k_i$  en (1.24) se cumple  $k_i = j_i \quad \forall i = 1, 2, \dots, M$ . La Fig.1.13 sirve para ilustrar el resultado de la descomposición del espacio de entrada en un caso bidimensional; en ella, como ejemplo, aparece sombreado el subespacio de entrada  $U_{32} = [e_1^3, e_1^4] \times [e_2^2, e_2^3]$ .

2) *Descomposición del sistema difuso*. Sobre cada subespacio de entrada  $U_{k_1 \dots k_M}$  es posible definir un sistema difuso del tipo definido en el apartado a),  $f_{k_1 \dots k_M}^s(\mathbf{x})$ , tal que

$$f_{k_1 \dots k_M}^s(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \forall (\mathbf{x} \in U_{k_1 \dots k_M}) \\ 0 & \forall (\mathbf{x} \in U - U_{k_1 \dots k_M}) \end{cases} \quad (1.25)$$

y de forma que

$$f(\mathbf{x}) = \sum_{C \text{ términos}} f_{k_1 \dots k_M}^s(\mathbf{x}) \quad (1.26)$$

$$\text{con}(k_i = 1, 2, \dots, (L_i - 1)) e \quad i = 1, 2, \dots, M$$

Cada sistema difuso simple  $f_{k_1 \dots k_M}^s(\mathbf{x})$  puede a su vez expresarse como una combinación de funciones base según la expresión

$$f_{k_1 \dots k_M}^s(\mathbf{x}) = \sum_{2^M \text{ términos}} \overline{y}_{k_1 \dots k_M}^{l_1 \dots l_M} B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x}) \quad (1.27)$$

$$\text{con}(l_i = k_i, k_{i+1})e \quad i = 1, 2, \dots, M$$

donde cada coeficiente  $\overline{y}_{k_1 \dots k_M}^{l_1 \dots l_M}$  corresponde con el valor de  $f(\mathbf{x})$  en cada uno de los puntos que delimitan el subespacio de entrada  $U_{k_1 \dots k_M}$ ; de forma que se tiene que

$$\overline{y}_{k_1 \dots k_M}^{l_1 \dots l_M} = f(e_1^{l_1}, \dots, e_M^{l_M}) \quad (1.28)$$

$$\text{con}(l_i = k_i, k_{i+1})e \quad i = 1, 2, \dots, M$$

Cada función base normalizada  $B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x})$  se calcula según la expresión

$$B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x}) = \frac{B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x})}{\sum_{2^M \text{ términos}} B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x})} \quad \forall \mathbf{x} \in U_{k_1 \dots k_M} \quad (1.29)$$

$$\text{con}(l_i = k_i, k_{i+1})e \quad i = 1, 2, \dots, M$$

y donde  $B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x})$  se calcula a partir de las funciones de pertenencia  $s_{A_i}^{l_i}(x_i)$ ,  $\forall j_i = 1, 2, \dots, L; \forall i = 1, 2, \dots, M$  según la expresión,

$$B_{k_1 \dots k_M}^{l_1 \dots l_M}(\mathbf{x}) = \text{T-norm} \left\{ s_{A_1}^{l_1}(x_1), \dots, s_{A_M}^{l_M}(x_M) \right\} \quad \forall \mathbf{x} \in U_{k_1 \dots k_M} \quad (1.30)$$

donde  $(k_i = j_i \quad \forall i = 1, 2, \dots, M)$  y donde  $(l_i = k_i, k_{i+1})e \quad i = 1, 2, \dots, M$

En esta expresión  $\text{T-norm}\{ \bullet \}$  puede ser, bien el operador mínimo, bien el operador producto.

Así, las expresiones (1.23)-(1.30) proporciona el método para realizar la partición del espacio de entrada y determinar el sistema difuso más simple en cada uno de los subsistemas difusos.

#### 5.4. Conclusiones

De los resultados anteriores se pueden extraer las siguientes conclusiones:

- Una sistema de control difuso como el de la expresión (1.3), definido sobre una partición en rejilla fija de su espacio de entrada es capaz de aproximar cualquier superficie de control con el grado de precisión deseado. La expresión (1.20) caracteriza a dicho sistema como un aproximador de funciones de primer orden, y

permite calcular el valor del módulo de la partición del espacio de entrada para una superficie y una precisión dadas. Sin embargo, en la práctica surge la dificultad de que, para obtener una buena precisión, suele ser necesario realizar una partición bastante fina del espacio de entrada, la cual da lugar a un número elevado de reglas y por consiguiente a sistemas de una complejidad elevada. Por otra parte, nótese que el sistema aquí considerado emplea como T-norma el operador mínimo, por lo que en este caso no es aplicable el resultado (1.21), para aproximadores de segundo orden.

- La propiedad de descomposición de los sistemas difusos, resultado del carácter local del mecanismo de aproximación de estos sistemas, muestra que es posible reproducir el comportamiento de cualquier sistema difuso a partir de las contribuciones de sistemas difusos del tipo más simple posible, (de mínima complejidad para un número de entradas dado), definidos sobre cada una de las diferentes subregiones en las que, a su vez, puede ser descompuesto el espacio de entrada original.

Ambos resultados pueden combinarse y ser explotados en tareas de diseño de forma que sea posible aprovechar la simplicidad, tanto conceptual, como de implementación, que caracteriza a las particiones en rejilla, y eludir el problema de la elevada complejidad requerida. De esta idea surge la arquitectura de diseño que se propone en este trabajo de tesis y que se desarrolla en el siguiente capítulo.

## **6. REALIZACIONES DE CONTROLADORES DIFUSOS Y NEURO-DIFUSOS MONOLÍTICOS MEDIANTE *HARDWARE* ANALÓGICO Y MIXTO**

### **6.1. Aspectos generales**

Dado el carácter no lineal de la relación entrada/salida (1.1) que caracteriza a los sistemas difusos, y a su aplicación a tareas de control en sistemas donde tanto las entradas como las salidas son señales analógicas, resulta bastante natural pensar en una realización *hardware* con circuitos electrónicos analógicos no lineales, sobre todo cuando las necesidades de velocidad de operación, compacidad, bajo consumo y coste son requisitos importantes para dicha aplicación; éste es el caso de muchas aplicaciones de control en tiempo real (robótica, procesamiento de imágenes y voz, electrónica de potencia, etc.).

Comparadas con las realizaciones digitales, (competidoras y alternativas en el campo del diseño electrónico) las principales ventajas de las realizaciones analógicas residen por una parte en el hecho de que la tecnología analógica permite operar con señales en tiempo continuo, y evitan la necesidad de la conversión analógico-digital, digital-analógico en los interfaces de entrada-salida; y por otra en que en su diseño es posible aprovechar las características no

lineales de los dispositivos electrónicos para realizar los diferentes operadores no lineales que conforman los algoritmos difusos mediante estructuras simples y compactas, lo que se traduce en sistemas que ofrecen las buenas prestaciones en cuanto a velocidad de operación, compacidad, bajo consumo y coste antes mencionadas. Esto último es posible gracias a las peculiaridades propias de los circuitos analógicos necesarios para la realización de los controladores difusos, cuyos requerimientos de diseño resultan ser menos restrictivos que los aplicables a los circuitos analógicos de propósito general, o a los empleados en aplicaciones lineales. Estas peculiaridades son destacadas por Yamakawa, que es considerado el precursor de la implementación de controladores difusos con *hardware* analógico [Yama80] [Yama86], en [Yama93]. En ese trabajo destaca por ejemplo, que en los circuitos analógicos empleados en el diseño de controladores difusos no resulta necesario realizar diseños con una elevada exactitud en cuanto a linealidad, deriva térmica y *offset*, debido a que una resolución de un 10% es suficiente en muchas de las aplicaciones, lo que permite que sea posible liberar a estos sistemas de la necesidad de los circuitos de compensación, que en los circuitos analógicos lineales se emplean precisamente par mejorar dichos aspectos, lo que permite a su vez emplear arquitecturas más sencillas y que utilizan menos recursos.

Un aspecto importante en la realización de un sistema difuso en general, y de su realización *hardware* en particular, ya sea mediante técnicas analógicas o digitales, es la elección del sistema de inferencia difuso a implementar. Su importancia radica no sólo en el hecho de que éste condicione las prestaciones del sistema de difuso en cuanto a la capacidad de modelado de la superficie objetivo de control, cuestión ésta que ha de ser abordada desde la teoría; sino que, desde el punto de vista de la realización, esta elección determina el conjunto de operadores y funciones que han de ser diseñados y en definitiva en la variedad, complejidad y tamaño del *hardware* necesario, y por tanto en la duración del ciclo de diseño. Como ya se ha mencionado, el sistema de inferencia difusa de *Takagi-Sugeno de orden cero*, descrito en el apartado 4.3. de este capítulo, es el más popular y más ampliamente utilizado en las realizaciones *hardware*, debido a que establece un muy buen compromiso entre la simplicidad en cuanto diseño y circuitería necesarias para su realización, y la eficiencia en el control, como recogen las discusiones que a este respecto se realizan en [Yama93], [Miki95] y [Batu97]. Cabe recordar que este sistema de inferencia emplea el método del centro de gravedad para el proceso de *defuzzificación* que resulta ventajoso desde el punto de vista de la implementación y que proporciona muy buenas prestaciones desde el punto de vista de las aplicaciones de control [Miki95].

Las críticas a las primeras implementaciones analógicas [Yama86] [Sasa92] destacaban los principales aspectos que deberían ser mejorados en posteriores desarrollos. Entre ellas cabe mencionar:

- La *mayor duración del ciclo de diseño* de las implementaciones analógicas,

consecuencia de la gran variedad primitivas y el mayor grado de detalle en cuanto a elementos y parámetros de diseño que hay que manejar, unido a la escasa automatización alcanzada en el área del diseño analógico.

- La *pobre flexibilidad y capacidad de adaptación*, debido principalmente al carácter de sistema cableado que tenían estas primeras implementaciones.
- La *baja complejidad* de los problemas de control abordable por los controladores analógicos, debido principalmente a factores y limitaciones propios del diseño analógico y de la naturaleza de los algoritmos difusos, en los que aparecen nudos de computación global, en los que errores y las capacidades parásitas asociadas a los diferentes dispositivos se agregan.

Estos aspectos, junto con la ocupación de área de silicio y la velocidad de operación determinan a su vez que los principales criterios de comparación entre las diferentes implementaciones. Las soluciones aportadas a esas críticas han supuesto diferentes líneas de desarrollo de controladores difusos.

En general, una vez decidido el sistema de inferencia a realizar, las principales aportaciones van encaminadas a establecer una metodología, fundamentada sobre una arquitectura sencilla y cómoda para su implementación, modular y desarrollada a partir de bloques constructivos bien definidos y escalables, de forma que resulte posible su automatización, y así conseguir una importante reducción del ciclo de diseño de un controlador analógico. Además, si este diseño modular está basado en bloques parametrizables y programables, se proporciona también una solución al problema de la falta de adaptabilidad. En esta doble vertiente de aportaciones destacan los trabajos reportados en [Mana96] [Pete98] [Batu97] y [Rodr99], para el diseño de controladores difusos analógicos y mixtos que realizan el sistema de inferencia de *Takagi-Sugeno de orden cero*.

Tradicionalmente, la clase de los controladores analógicos se ha dividido en controladores que operan en modo corriente y controladores que operan en modo tensión. Esta división también puede establecerse para las realizaciones antes mencionadas, en cuanto al modo de operación de los circuitos analógicos que contienen, y que son los que realizan los elementos de cálculo del algoritmo, aunque todos ellos pueden ser considerados controladores en modo mixto ya que emplean técnicas digitales para sus propósitos de flexibilidad. Así los controladores reportados en [Mana96] [Batu97] y [Rodr99] trabajan en modo corriente, mientras que los reportados [Pete98] operan en modo tensión. A continuación se describen algunos aspectos de estos trabajos.

## 6.2. Controladores Difusos y Neuro-Difusos monolíticos analógicos y mixtos

En este apartado se consideran brevemente las diferentes realizaciones de controladores analógicos y mixtos monolíticos reportadas en [Mana96] [Guo96] [Batu97] y [Vida96b], prestando atención al hecho de que representan distintas estrategias y aproximaciones al diseño de controladores difusos, más que a la naturaleza de los circuitos empleados en la realización de los distintos bloques que definen la arquitectura propuesta en cada uno de los casos.

La Fig.1.14 muestra un diagrama que clasifica de las diferentes posibilidades que se han contemplado en la aproximación al diseño de controladores difusos analógicos flexibles y de los trabajos que sirven de referencia. En primer lugar cabe distinguir dos tipos de aproximaciones, según se aborde la cuestión de la flexibilidad, bien como cuestión previa a la fabricación, determinando y fijando los parámetros de configuración durante la fase de diseño, lo que supone a la vez un proceso de extracción o minimización automática del número de reglas, a fin de realizar el controlador con el menor número de elementos posible; o bien como una cuestión posterior a la fabricación, lo que supone la construcción de un controlador programable, que puede ser configurado por programación directa de dichos parámetros, o a partir de valores de los parámetros obtenidos mediante algoritmos de aprendizaje. En esta aproximación al diseño de controladores programables cabe a su vez distinguir dos opciones diferentes, que podrían denominarse, de flexibilidad completa, o bien de flexibilidad limitada. En la primera se plantea la posibilidad de configurar todos los elementos de la arquitectura del

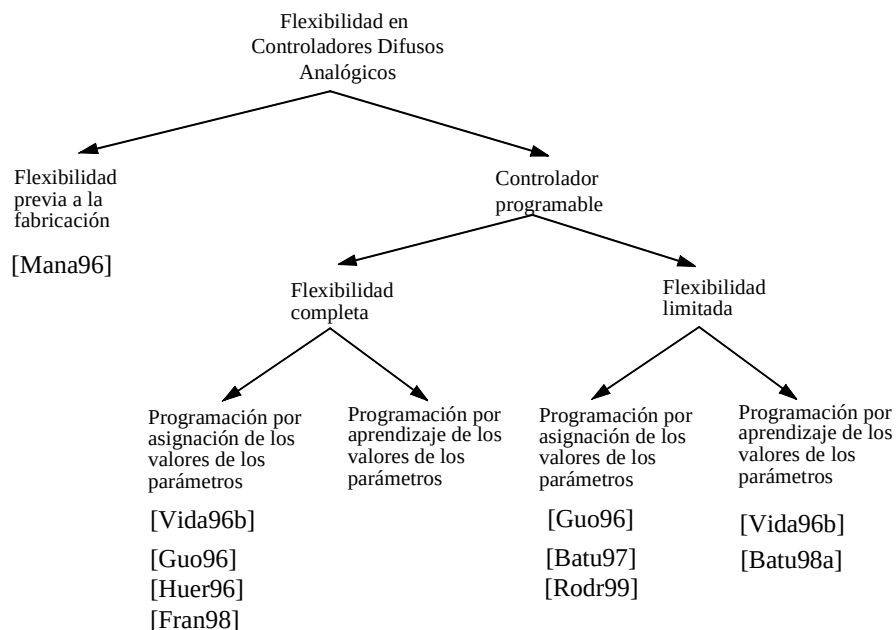


Fig.1.14 Diagrama de las diferentes aproximaciones al diseño de controladores difusos analógicos flexibles consideradas.



controlador, (forma y ubicación de las funciones de pertenencia, número reglas y composición de los antecedentes de cada regla, etc.) de modo que sea posible realizar cualquier partición del universo de discurso, y tan solo con la limitación de un número máximo de reglas. Esta opción supone un mayor gasto de área de silicio y un gran número de parámetros de configuración. La segunda opción restringe la flexibilidad en alguno de los aspectos de la arquitectura, (fijar la forma de las funciones de pertenencia, fijar el número de reglas, trabajar con una partición en rejilla del espacio de entrada, etc), lo que permite compartir recursos y aumentar la transparencia del sistema difuso. Esta opción supone un ahorro en ocupación de área de silicio, y la disminución del número de parámetros de configuración, lo que mejora de su transparencia facilita su programación. A continuación se describen brevemente los principales aspectos de cada una de las propuestas:

- En [Mana96] se presenta un compilador de silicio para la realización de controladores difusos analógicos. La herramienta *software* reportada acepta como entrada, bien una descripción clásica en forma de las reglas difusas, o bien ejemplos numéricos en forma de pares entrada-salida. El programa, que trabaja sobre una arquitectura modular basada en bloques asociados a reglas como ilustra la Fig.1.15, se encarga automáticamente de seleccionar el conjunto mínimo de reglas y de computar los parámetros de configuración de cada uno de esos bloques, que son necesarios para aproximar una superficie de control dada. El resultado es un conjunto de sentencias en el formato que sirve como directivas que son interpretadas por una herramienta comercial de generación automática de *layout*. La principal ventaja de esta aproximación recae en la importante reducción en el ciclo de diseño y en la optimización del área de silicio ocupada para la solución de un problema de control

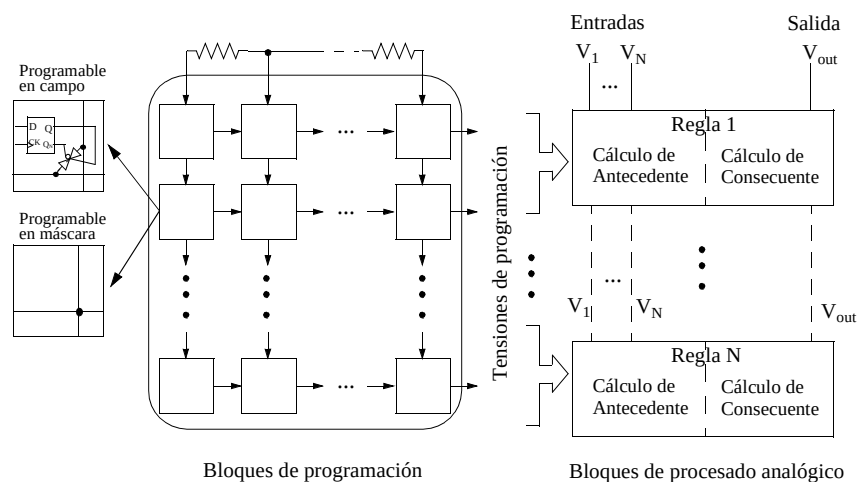


Fig.1.15 Arquitectura de los controladores reportados en [Mana96] y [Fran98].

particular. En su contra cabe destacar la imposibilidad de realizar modificaciones y/o ajustes posteriores en campo, por lo que el éxito en la aplicación recae en la bondad del trabajo realizado por los programas de minimización de reglas y cálculo de parámetros, y modelado de los circuitos que interviene en el diseño. Para eludir en cierta forma este inconveniente, los autores proponen en [Fran98] un controlador programable en campo, útil durante la fase de prototipo para una aplicación dada. En este caso, cada prototipo considerado presenta una arquitectura que contempla un número máximo de reglas a realizar, con los grados de libertad necesarios para ubicar las funciones de pertenencia en cualquier punto del universo de discurso y con un número elevado de parámetros de configuración que son gestionados por la herramienta software de diseño, lo que disminuye su transparencia. Aunque en este caso la ocupación de área puede resultar mayor que la de las otras realizaciones propuestas en la literatura, es de suponer que tras el ajuste en campo del prototipo, el controlador puede ser finalmente fabricado de forma óptima para una aplicación dada, aunque, claro está, a costa de perder su flexibilidad.

- En [Huer96] y [Batu97] se describen diferentes arquitecturas modulares de tipo matricial para controladores programables basadas en bloques funcionales programables digitalmente mediante conversores D/A y que realizan las distintas etapas del algoritmo difuso, (etapa de *fuzzificación*, etapa *conectiva* y etapa de *agregación*). En [Huer96] la arquitectura es completamente flexible permitiendo realizar cualquier tipo de partición del espacio de entrada, programando

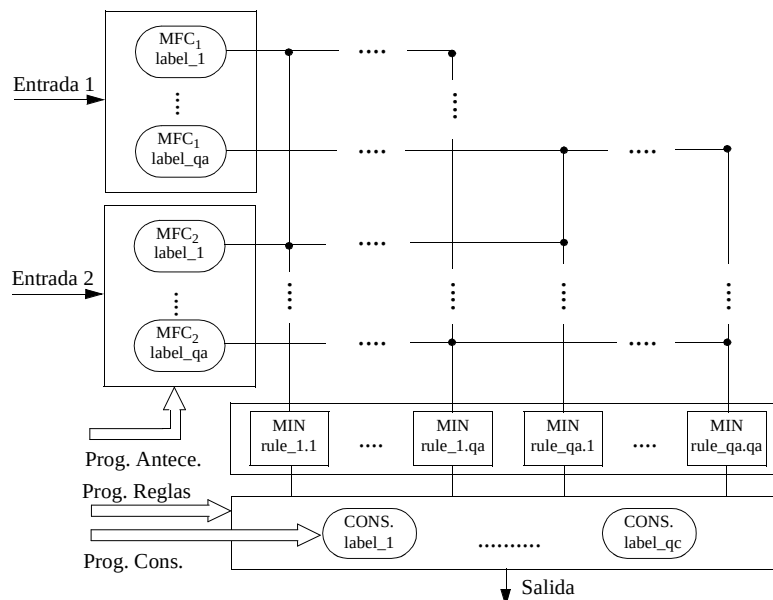


Fig.1.16 Arquitectura de los controladores reportados en [Batu97].

adecuadamente los diferentes bloques. Esta realización resulta costosa en área, sobre todo cuando se pretende diseñar un controlador programable de propósito general, y resulta compleja de manejar dado el elevado número de parámetros de programación a contemplar, por lo que se hace necesario establecer un compromiso. En base a este argumento en [Batu97] se propone una restricción en la flexibilidad reduciendo la partición del espacio de entrada a configuraciones en rejilla, esto supone compartir los bloques de funciones de pertenencia entre entradas y compartir los valores de los consecuentes de regla entre reglas. La Fig.1.16 muestra un esquema de la arquitectura de esta propuesta para un controlador con dos entradas, una salida,  $q_a$  etiquetas por entrada, y  $q_c$  *singleton*. En ella, sí se sigue manteniendo la programación de la forma de las pendientes del funciones de pertenencia triangulares utilizadas. En esta situación se consigue un considerable ahorro en área y en el número de parámetros de programación a manejar. En cuanto a la capacidad de aproximación de funciones, según se ha contemplado en el apartado 5.2. de este capítulo, ésta queda limitada por la granularidad de la partición efectuada. Las prestaciones de este tipo de controladores, debido a la partición en rejilla del espacio de entrada que realizan, se ven degradadas con el aumento del número de entradas y de reglas. Finalmente en [Batu98a] se reporta un ejemplo de la utilización de un controlador que sigue el modelo descrito en [Batu97] integrado en un lazo de aprendizaje de los valores *singleton* de programación.

- En [Guo96] se presenta un controlador programable que permite una reconfiguración de la base de reglas y los valores *singleton*, y que pertenece a una segunda generación de controladores difusos según se describe en [Pete98]. Esta segunda generación, constituye una etapa intermedia entre los controladores no flexibles de la primera generación y los controladores de una tercera generación denominada por los autores coprocesadores difusos que ofrecen una completa flexibilidad en la elección de la base de reglas y términos difusos que pueden ser seleccionados a través de un interfaz digital compatible con tarjetas de entrada/salida estándar para PC. En la Fig.1.17 se muestra la arquitectura de un controlador perteneciente a esta última generación, que posee tres entradas, cinco etiquetas por entrada, y siete *singleton*. En los controladores pertenecientes a la segunda y tercera generación la flexibilidad en la configuración de la base de reglas se consigue, de la misma manera que se hace en [Huer96], introduciendo multiplexores y demultiplexores analógicos, controlados digitalmente, en el camino de procesamiento de señal que conecta antecedentes y consecuentes de regla, lo que sin duda condiciona sus prestaciones, frente a una implementación puramente analógica. En el compromiso de diseño entre flexibilidad, compacidad y consumo, en esta propuesta se apuesta por la flexibilidad, como se deduce de los datos aportados en [Pete98].

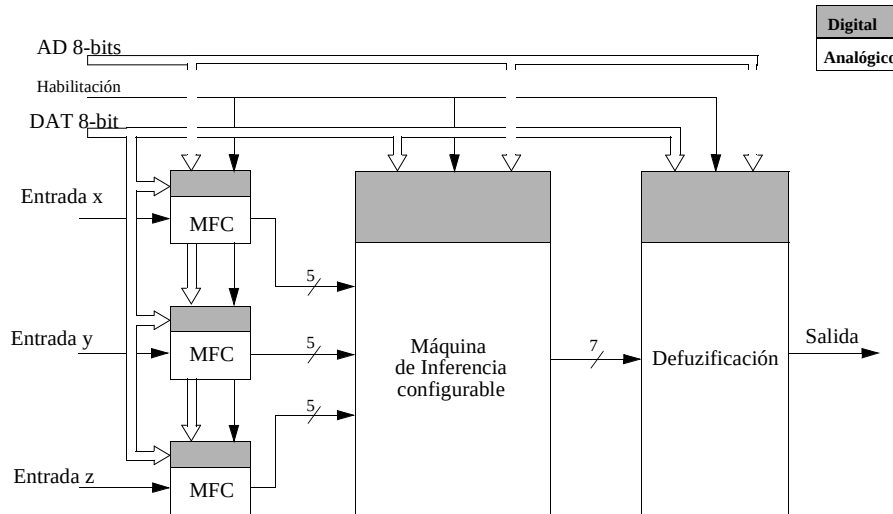


Fig.1.17 Arquitectura para la tercera generación de controladores reportados en [Pete98].

- En [Rodr99] se presenta un controlador difuso diseñado sobre una arquitectura modular basada en dos bloques de alto nivel parametrizados y programables, llamados *Bloque de Etiqueta* y *Bloque de Regla*, que permiten realizar mediante simple conexionado de los mismos controladores difusos analógicos sobre particiones en rejilla de su espacio de entrada. La Fig.1.18 recoge un diagrama de bloques para un controlador genérico que posea  $M$  entradas y  $L$  etiquetas por entrada y por tanto  $L^M$  reglas. Esta arquitectura resulta de reorganizar y redistribuir los diferentes circuitos que realizan los operadores que constituyen el algoritmo difuso según la arquitectura ANFIS para particiones en rejilla mostrada en la Fig.1.10, y discutida en el apartado 4.5.2. de este capítulo. Los bloques constructivos básicos para la realización de las funciones asociadas a los diferentes nodos de la red adaptativa ANFIS están desarrollados en [Vida96b], y constituyen una propuesta para el diseño con *hardware* analógico de sistemas neuro-difusos. Con esos bloques constructivos también es posible acometer el diseño de un controlador según la arquitectura general ANFIS mostrada en la Fig.1.10 y discutida en el apartado 4.5.1. de este capítulo. En [Vida96b] se muestran resultados de prototipos sencillos diseñados con esa organización. Sin embargo, y siguiendo la discusión que se hace en el apartado 4.5.1., un diseño de estas características para controladores de mayor complejidad resulta en general más costoso en cuanto a ocupación de área, consumo de potencia y complejidad de uso, de ahí la opción de partición en rejilla elegida en [Rodr99]. Por otra parte, dado el carácter neuro-difuso de la arquitectura conceptual que soporta a este controlador, en [Vida96b] también se propone una estrategia de

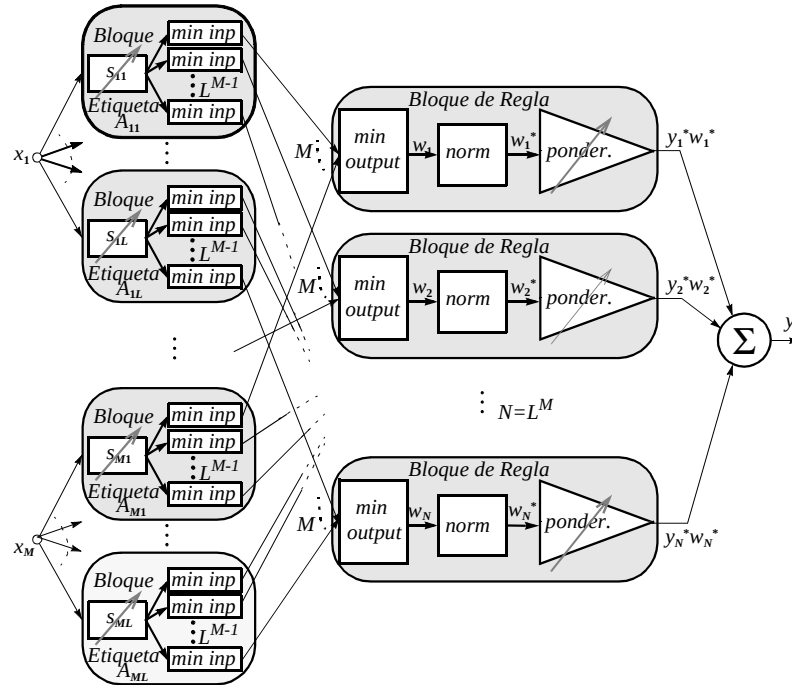


Fig.1.18 Arquitectura de los controladores reportados en [Rodr99].

de controladores. Esta propuesta adopta un algoritmo de aprendizaje supervisado híbrido, en el que para el aprendizaje de los parámetros que caracterizan a las funciones de pertenencia se utiliza el algoritmo *weight perturbation* [Jabr92] mientras que para el aprendizaje de los valores singleton se emplea el algoritmo *out-start*. La Fig.1.19 muestra un esquema del entrenamiento a partir de ejemplos numéricos. Finalmente, dada la partición en rejilla del espacio de entrada que realizan los controladores de esta propuesta, sus prestaciones se ven degradadas de una manera importante con el aumento del número de entradas y de reglas.

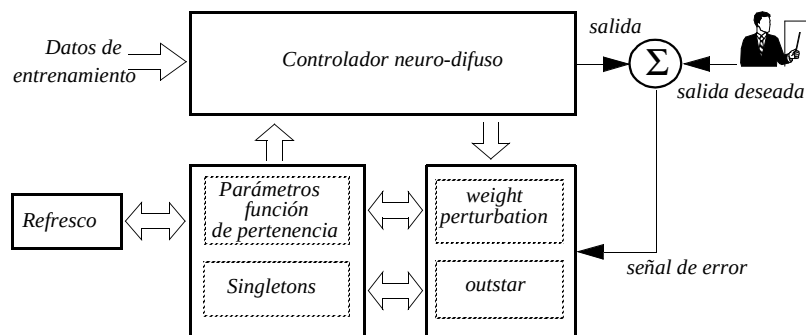


Fig.1.19 Esquema de aprendizaje propuesto en [Vida96b].

Como se ha mencionado con anterioridad, todos estos trabajos realizan aportaciones encaminadas a establecer una metodología de diseño, fundamentada sobre una arquitectura sencilla, modular y cómoda para su implementación, desarrollada a partir de bloques constructivos bien definidos, escalables, y parametrizados de forma que se facilite la implementación de controladores difusos flexibles y su automatización. Sin embargo, la cuestión de la complejidad abordable por las realizaciones mediante *hardware* analógico en general, o por las realizaciones descritas arriba en particular, así como su influencia en las principales prestaciones del sistema, es un aspecto que pocas veces es considerado, y cuando se aportan datos como en [Mana96], éstos ponen de manifiesto la importante dependencia de factores como el consumo total de potencia, o el retardo de propagación con el número entradas y el número de reglas que el controlador implementa. Además, en la mayoría de los casos, los controladores ejemplo que son reportados resultan ser controladores de baja complejidad, y no exceden las dieciséis reglas y tres entradas.

Los principales factores limitadores del incremento de la complejidad abordable mediante controladores realizados con circuitos analógicos son de índole tecnológica y de diseño. En todas las realizaciones analógicas de los algoritmos difusos que se han propuesto aparecen nudos de computación global, en los que se suman también los errores (sistemáticos o aleatorios) causados por los circuitos que proporcionan entradas a esos nudos, de modo que el error global final se hace mayor a medida que se incrementa el número de términos. Este es el caso de los nudos de computación que aparecen al realizar el proceso de *defuzzificación* empleando el método del centro de gravedad, en los que los errores procedentes de los circuitos que generan los antecedentes de reglas se suman. Esta acumulación del error supone, entre otras cosas, una degradación en la precisión del controlador. Además, en estos nudos también se acumulan los condensadores parásitos, lo que afecta negativamente a la velocidad de operación. Por tanto, un incremento en la complejidad del sistema difuso, lleva a una degradación de las prestaciones, en cuanto a precisión, velocidad, consumo y ocupación de área de silicio, que resultan ser las principales ventajas de las realizaciones mediante sistemas analógicos.

En este sentido el propósito de la arquitectura para controladores difusos propuesta en este trabajo, y que se presenta en el siguiente capítulo, va encaminada a permitir un sensible incremento en la complejidad de los controladores realizados con técnicas analógicas, manteniendo sus principales prestaciones equiparables a las que presentan los controladores de baja complejidad.



---

## Capítulo 2

---

### Propuesta de una arquitectura multiplexada para controladores difusos en modo mixto.

#### 1. INTRODUCCIÓN

Como se ha indicado en la Introducción, los controladores difusos analógicos resultan ser ventajosos en aplicaciones de baja complejidad. Muchos de los problemas de control en los que un ASIC puede ser empleado tienen un nivel de complejidad no superior a 100 reglas y 4 entradas [Cost95], sin embargo, los controladores difusos analógicos más complejos que aparecen en la literatura, entre los que están los reportados en [Guo96] con 13-reglas y 3 entradas, y en [Vida96a] con 16-reglas y 2 entradas, resultan de mucha menor complejidad que la anteriormente citada. Uno de los principales obstáculos para incrementar la complejidad de los controladores difusos analógicos es la existencia de nudos de computación global [Jang95][Brow94], (como los que aparecen al realizar el proceso de *defuzzificación* empleando el método del centro de gravedad). En estos nudos se suman los errores (sistemáticos o aleatorios) causados por los circuitos que realizan los antecedentes de regla, de modo que el error global final se hace mayor, lo que supone, entre otras cosas una degradación en su precisión. Por otra parte en estos nudos se suman también los condensadores parásitos asociados a los dispositivos electrónicos empleados en el diseño, lo que se traduce en una disminución en la velocidad de operación. Además, el aumento de complejidad, lleva a una degradación de las buenas prestaciones de los sistemas analógicos en cuanto a consumo y ocupación de área de silicio. Todas estas circunstancias son especialmente relevantes en aplicaciones de control, en las cuales es bastante usual emplear una partición en rejilla del universo de discurso [Brow94], y en las que hay un crecimiento exponencial del número de reglas con el número de entradas.

En este capítulo se presenta una nueva arquitectura conceptual para la implementación mediante circuitos integrados de señal mixta de controladores difusos, y que llamaremos *Sistema Neuro-Difuso de Arquitectura Multiplexada*. Con ella se preservan las ventajas de una implementación puramente analógica (manteniendo por ejemplo, el paralelismo computacional, la compacidad y una óptima eficiencia en cuanto al balance velocidad/consumo), a la vez que se independiza la acumulación del error del número de reglas, lo que permite un incremento en la complejidad de los problemas abordables con estos sistemas.

Por razones de simplicidad en la implementación *hardware*, esta arquitectura realiza el algoritmo difuso de *Takagi-Sugeno* con valores *singleton* en el consecuente [Taka89], que ya ha sido descrito en el capítulo anterior. Por la misma razón, genera únicamente particiones en



rejilla del espacio de entrada, y explota el carácter local de las reglas difusas a que da lugar este tipo de partición, de modo que, para cada región del el universo de discurso, la salida puede ser obtenida a partir de un conjunto parametrizado de reglas que contribuyen con un valor no nulo en esa región. Estas reglas son denominadas, reglas activas [Iked91]. Los valores de dichos parámetros son específicos de cada región del universo de discurso y por tanto dependen directamente de las entradas del sistema, lo que permite su identificación a partir de éstas. Dado que se trabaja con una partición en rejilla, el sistema difuso formado por las reglas activas, es también el sistema difuso más simple posible, según la definición de [Zeng96b], en la región considerada.

La arquitectura propuesta se organiza en torno a un bloque programable que realiza de forma genérica, y con técnicas analógicas, el sistema de inferencia difusa más simple posible para el espacio de entrada considerado. Este bloque realiza el cómputo de la salida en base a las reglas activas sobre una región dada. Una bloque de almacenamiento se utiliza para guardar el conjunto completo de parámetros que caracteriza al sistema global y que constituye la base de reglas. Finalmente un sistema de selección identifica las reglas activas a partir de las entradas actuales y permite seleccionar los valores adecuados de la base de reglas para reconfigurar dinámicamente sistema difuso programable. Así, la superficie de respuesta global se obtiene programando y multiplexando dinámicamente el bloque que realiza el sistema difuso más simple posible; de ahí el nombre de *Sistema Neuro-Difuso de Arquitectura Multiplexada*. La Fig.2.1 muestra un esquema genérico de esta arquitectura.

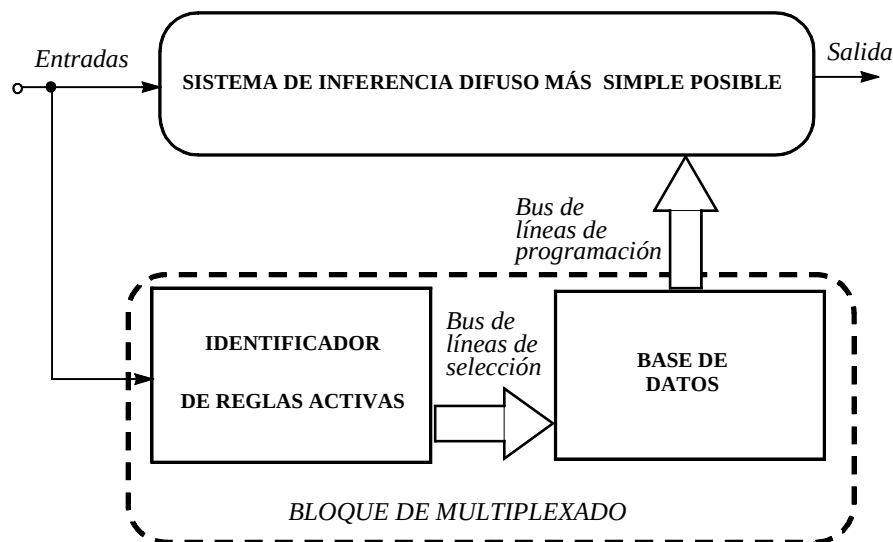


Fig.2.1 Diagrama de bloques genérico de un sistema difuso de arquitectura multiplexada

Es importante señalar que, aunque, al menos de forma teórica, tanto la capacidad de aproximación, como el grado de precisión que es posible alcanzar con el sistema propuesto quedan fijados por diseño, al especificar la estructura de la partición en rejilla considerada en el espacio de entrada, la nueva arquitectura aquí definida está dotada de la flexibilidad suficiente para permitir ser reconfigurada, e incluso entrenada, a partir de un conjunto de parejas de datos de entrada/salida, modificando los valores de los parámetros almacenados, bien por programación directa, o bien como resultado de su inclusión en un lazo de aprendizaje.

El presente capítulo se organiza de la siguiente manera. En primer lugar se definen los requerimientos básicos de la nueva arquitectura en base a conceptos como los de *sistema de inferencia más simple posible*, *regla activa*, e *intervalo de interpolación*, y propiedades como la *propiedad de descomposición* de los sistemas difusos, presentados en el Capítulo 1 que resultan claves para la arquitectura propuesta. Por razones de claridad, se explora en primer lugar un caso bidimensional, para abordar a continuación un caso general. Finalmente se describe a nivel de diagrama de bloques una propuesta de arquitectura que cumple con esas especificaciones. Queda para capítulos posteriores la descripción de una propuesta para su realización mediante circuitos CMOS estándar. Por último se discuten la capacidades de programación y aprendizaje del sistema a partir de una regla de aprendizaje simple, orientada al *hardware* y que no necesita el cómputo de derivadas.

## 2. ARQUITECTURA MULTIPLEXADA.

En este apartado se describen los aspectos básicos de definen la nueva arquitectura multiplexada para controladores difusos propuesta en esta tesis. Se considera en primer lugar el problema más sencillo de sistemas bidimensionales, para posteriormente generalizar para un sistema  $M$ -dimensional.

### 2.1. Ejemplo bidimensional

Consideremos en primer lugar el problema de aproximar una superficie bidimensional  $y = f(x_1, x_2)$ , tal que  $f: U \subset R^2 \rightarrow V \subset R$ , donde  $(U = U_1 \times U_2) \subset R^2$  es el espacio de entrada y  $V \subset R$  el espacio de salida, mediante un sistema difuso del tipo Takagi-Sugeno descrito en el apartado 4.3. del Capítulo 1, con dos entradas  $(x_1, x_2)$ , definidas sobre el universo de discurso  $U$ . Para cada una de ellas se define un conjunto de  $L$  etiquetas,  $X_1 = \{X_{11}, \dots, X_{1L}\}$  y  $X_2 = \{X_{21}, \dots, X_{2L}\}$ , respectivamente. Estos dos conjuntos se eligen normalizados, consistentes y completos [Zeng96a][Wang97]. Esta situación genera en el espacio de entrada una partición en rejilla bidimensional como la que se muestra en la Fig.2.2(a), donde cada región, como la señalada con  $R_{mn}$ , coincide con la región que define el ancho de una función de pertenencia

bidimensional normalizada  $w_{mn}^*$  ( $2\Delta \times 2\Delta$ ), como la mostrada en la Fig.2.2(c). Se tendrá, por tanto, un total de  $L^2$  regiones de este tipo, correspondientes a otras tantas reglas. Fijemos nuestra atención en el recuadro sombreado en oscuro en la Fig.2.2(a). Esa región del espacio de entrada corresponde a cuatro reglas genéricas adyacentes  $R_{m,n}$ ,  $R_{m+1,n}$ ,  $R_{m,n+1}$ ,  $R_{m+1,n+1}$  definidas en la forma,

$$\begin{aligned}
 R_{m,n}: & \text{ SI } x_1 \text{ es } X_{1m} \quad \text{ Y } x_2 \text{ es } X_{2n} \quad \text{ ENTONCES } y = y_{mn}^* \\
 R_{(m+1),n}: & \text{ SI } x_1 \text{ es } X_{1(m+1)} \quad \text{ Y } x_2 \text{ es } X_{2n} \quad \text{ ENTONCES } y = y_{(m+1)n}^* \\
 R_{m,(n+1)}: & \text{ SI } x_1 \text{ es } X_{1m} \quad \text{ Y } x_2 \text{ es } X_{2(n+1)} \quad \text{ ENTONCES } y = y_{m(n+1)}^* \\
 R_{(m+1),(n+1)}: & \text{ SI } x_1 \text{ es } X_{1(m+1)} \quad \text{ Y } x_2 \text{ es } X_{2(n+1)} \quad \text{ ENTONCES } y = y_{(m+1)(n+1)}^*
 \end{aligned} \tag{2.1}$$

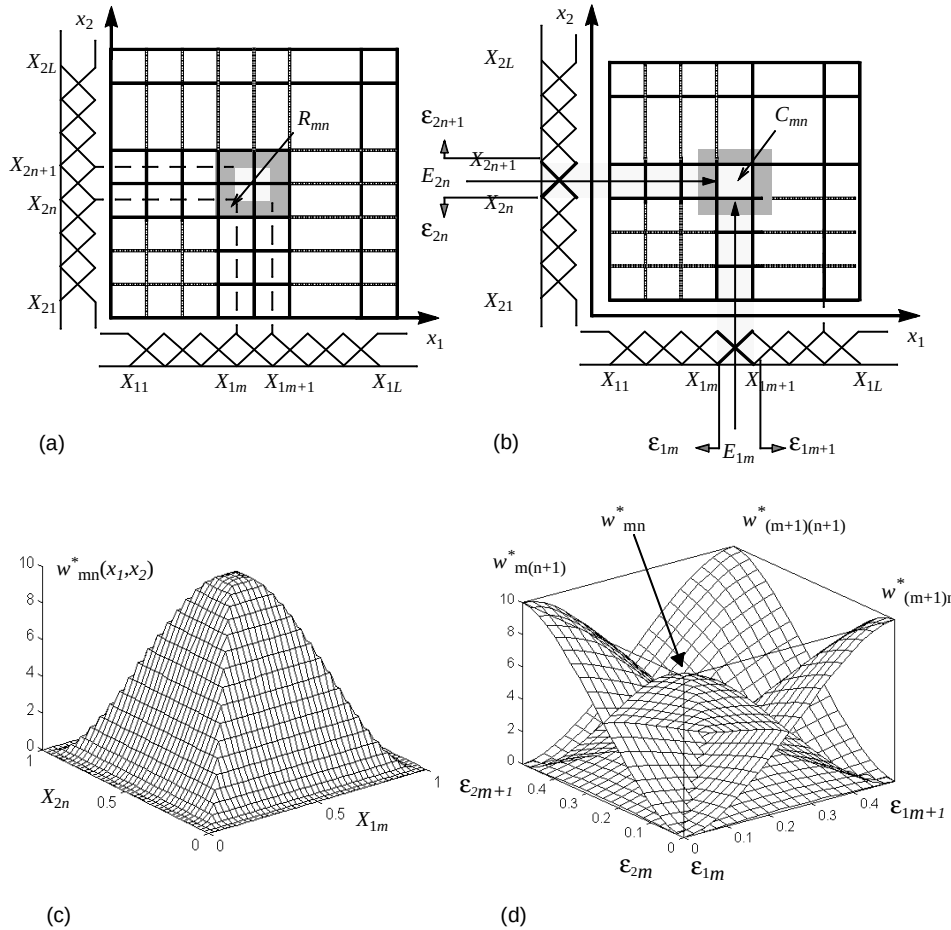


**Fig.2.2** Partición en rejilla bidimensional (a); partición en intervalos de interpolación (b); función de pertenencia multidimensional, antecedente de la regla genérica  $R_{mn}$  (c); antecedentes de las cuatro reglas activas en el intervalo de interpolación  $C_{mn}$  (d).

y donde  $y_{mn}^*, y_{m(n+1)}^*, y_{(m+1)n}^*$  e  $y_{(m+1)(n+1)}^*$  son los valores *singleton* que determinan el consecuente de cada regla, y que coinciden con el valor que toma la superficie de control que se quiere generar en el centro de cada una de las cuatro regiones ( por ejemplo, el punto de coordenadas  $\epsilon_{mn} = (\epsilon_{1m}, \epsilon_{2n})$  para  $R_{mn}$ , el la Fig.2.2(b)).

Para cualquier pareja de entradas  $(x_1, x_2)$  contenida en la zona central de esta región, en color más claro en la Fig.2.2(a), la salida del sistema se obtendrá a partir de las contribuciones de esas cuatro reglas, mientras que el resto de las reglas tendrán una contribución nula. Esta situación se ilustra en la Fig.2.2(d), donde aparecen los antecedentes de las reglas con contribución no nula en la subregión central considerada (  $w_{mn}^*, w_{m(n+1)}^*, w_{(m+1)n}^*$  y  $w_{(m+1)(n+1)}^*$  ). Llamaremos a esa subregión *Intervalo de Interpolación*, ( genéricamente  $C_{mn}$ ), y a las reglas que contribuyen de forma no nula a la salida en ese intervalo,  $(R_{m,n}, R_{m+1,n}, R_{m,n+1}, R_{m+1,n+1})$ , *reglas activas* [Ikeda91] [Mase95].

En base a estas definiciones podemos realizar una nueva partición del espacio de entrada en intervalos de interpolación, tal y como se muestra en la Fig.2.2(b). En ella, cada intervalo de interpolación genérico, que aparece señalado como  $C_{mn}$  en dicha figura, define una región de  $U$ ,  $C_{mn} = ([\epsilon_{1m}, \epsilon_{1m+1}[ \times [\epsilon_{2n}, \epsilon_{2n+1}[ )$ , en donde cada uno de los puntos,  $(\epsilon_{1m}, \epsilon_{2n})$ ,  $(\epsilon_{1m}, \epsilon_{2(n+1)})$ ,  $(\epsilon_{1(m+1)}, \epsilon_{2n})$ , y  $(\epsilon_{1(m+1)}, \epsilon_{2(n+1)})$  corresponde a uno de los centros de cada una de las cuatro funciones de pertenencia de las reglas activas en dicho intervalo. En total se tendrán,  $(L-1)^2$  intervalos de interpolación diferentes y disjuntos<sup>1†</sup>, cada uno de los cuales queda claramente identificado, para cada pareja de entrada  $(x_1, x_2)$ , de forma que el espacio de entrada completo  $U$  puede obtenerse como

$$U = \bigcup_{m,n=1}^{L-1} C_{mn} \quad (2.2)$$

El conjunto de reglas activas descrito en (2.1) constituye un sistema difuso simple, según la definición de [Zeng96b], sobre el subespacio de entrada determinado por  $C_{mn}$ . Este sistema presenta las siguientes características:

- Es un sistema de inferencia con cuatro reglas y definido sobre una partición en rejilla que emplea dos etiquetas por entrada.
- Cada una de las etiquetas se corresponde con un trozo de una etiqueta definida en el sistema global, aquel trozo que queda incluido en intervalo genérico  $C_{mn}$  que determina el espacio de entrada de este sistema simple, (véanse los trozos de las etiquetas genéricas  $X_{1m}$  y  $X_{1(m+1)}$ , y  $X_{2n}$  y  $X_{2(n+1)}$  marcados con trazo más grueso en la Fig.2.2(b) y los cuatro antecedentes de regla bidimensionales a que dan lugar en la Fig.2.2(d)).

<sup>1†</sup>. Nótese que  $C_{mn}$  se ha definido como el producto cartesiano de dos intervalos semicerrados.

- Cada pareja de funciones de pertenencia resulta ser una la complementaria de la otra, y por tanto, para cada entrada, ambas pueden especificarse a partir de un solo parámetro por entrada; por ejemplo a partir de los valores  $E_{1m}$  y  $E_{2n}$  que determina el punto de cruce de cada pareja de funciones de pertenencia, (Fig.2.2(b)) y ambos simultáneamente el centro del intervalo  $C_{mn}$ .

Por tanto, este sistema simple queda completamente especificado a partir de dos conjuntos de parámetros, los dos puntos  $E_{1m}$  y  $E_{2n}$ , y los valores *singleton*  $y_{mn}^*, y_{m(n+1)}^*, y_{(m+1)n}^*, y_{(m+1)(n+1)}^*$  que permiten configurar, respectivamente, los antecedentes y los consecuentes de las reglas activas en el intervalo  $C_{mn}$ .

Si llamamos  $y_{mn}$  a la salida proporcionada por el sistema simple en cualquier punto del intervalo  $C_{mn}$ , y dado el resultado que expresa (2.2), la salida del sistema global en cualquier punto del espacio de entrada completo  $U$ , puede obtenerse a partir de las contribuciones de cada sistema simple sobre su propio intervalo de definición según la expresión,

$$y = \sum_{m,n} y_{mn} \quad (1 \leq m \leq L-1) \text{ y } (1 \leq n \leq L-1) \quad (2.3)$$

donde

$$y_{mn} = y_{mn}^* w_{mn}^* \Big|_{mn} + y_{(m+1)n}^* w_{(m+1)n}^* \Big|_{mn} + y_{m(n+1)}^* w_{m(n+1)}^* \Big|_{mn} + y_{(m+1)(n+1)}^* w_{(m+1)(n+1)}^* \Big|_{mn} \quad (2.4)$$

y donde las expresiones  $w_{mn}^* \Big|_{mn}, w_{m(n+1)}^* \Big|_{mn}, w_{(m+1)n}^* \Big|_{mn}$  y  $w_{(m+1)(n+1)}^* \Big|_{mn}$  denotan las funciones base normalizadas, resultado de evaluar los antecedentes de reglas, con su soporte restringido al intervalo de interpolación  $C_{mn}$ . Nótese que cada  $y_{mn}$  contribuye con valor cero fuera de su intervalo de definición  $C_{mn}$ .

Bajo estas consideraciones, el problema de diseñar un sistema para aproximar la superficie de control en todo el espacio de entradas según la expresión (2.3), puede abordarse realizando un sistema difuso simple que trabaje con las reglas activas asociadas a cada intervalo de interpolación  $C_{mn}$ , que obtenga  $y_{mn}$  según la expresión (2.4), y que posea a la vez la suficiente flexibilidad para reconfigurarse dinámicamente y operar en cualquiera de los intervalos en los que se ha dividido el espacio de entrada. Llamaremos al sistema completo, *Sistema Difuso de Arquitectura Multiplexada*; y al sistema difuso simple, *Núcleo* del sistema difuso.

El sistema difuso de arquitectura multiplexada, ha de proporcionar la manera de almacenar o de generar los valores de los parámetros que caracterizan al sistema global:

- El conjunto de valores  $\{\epsilon_{1m}, \epsilon_{2n}\}, 1 \leq m \leq L, 1 \leq n \leq L$  que definen la partición en

rejilla del espacio de entrada global.

- El conjunto de valores  $\{E_{1m} E_{2n}\}$ ,  $1 \leq m \leq (L-1)$ ,  $1 \leq n \leq (L-1)$  que configuran los antecedentes de regla en cada intervalo.
- El conjunto de valores *singleton*  $\{y_{mn}^*\}$ ,  $1 \leq m \leq L$ ,  $1 \leq n \leq L$  que determinan el consecuente de todas las reglas del sistema completo.

Además ha de identificar, a partir de las entradas, cuál es el intervalo de interpolación actual  $C_{mn}$ , y seleccionar los parámetros adecuados  $E_{1m}$  y  $E_{2n}$ , y  $y_{mn}^*, y_{m(n+1)}^*, y_{(m+1)n}^*$  e  $y_{(m+1)(n+1)}^*$  que reconfiguran al núcleo para operar con las correspondientes reglas activas.

## 2.2. Caso multidimensional

Consideremos a continuación el problema general de aproximar una superficie  $M$ -dimensional  $y = f(x)$ , con  $f: U \subset R^M \rightarrow V \subset R$ , donde  $(U = U_1 \times U_2 \times \dots \times U_M) \subset R^M$  es el espacio de entrada y  $V \subset R$  el espacio de salida, mediante un sistema difuso del tipo Takagi-Sugeno descrito en el apartado 4.3. del Capítulo 1, con  $M$  entradas,  $L$  etiquetas por entrada, y, por tanto,  $L^M$  reglas. Para cada entrada  $x_i$ , con  $1 \leq i \leq M$ , se considera un conjunto de etiquetas  $X_i \equiv \{X_{i1}, X_{i2}, \dots, X_{iL}\}$ . A cada etiqueta  $X_{ij_i}$  se asocia una función de pertenencia  $s_{ij_i}(x)$  del tipo de las que aparecen en la Fig.1.8(b),(c) y (d) del Capítulo 1, y todos los  $X_i$  se suponen normalizados, consistentes y completos.

En esta situación se tendrán  $L^M$  funciones de pertenencia  $M$ -dimensionales  $w_{j_1 \dots j_M} = \min\{s_{1j_1}(x_1), s_{2j_2}(x_2), \dots, s_{Mj_M}(x_M)\}$  con cada  $j_i$  tal que  $1 \leq j_i \leq L$ .

Se tendrán además  $(L-1)^M$  intervalos de interpolación. Cada intervalo  $M$ -dimensional genérico  $C_{k_1 \dots k_M}$ , con  $1 \leq k_i \leq L-1$  se obtiene como

$$\prod_{i=1}^M [\epsilon_{ij_i}, \epsilon_{i(j_i+1)}] \quad , \text{ con } (j_i = k_i), \forall i \quad (2.5)$$

donde los puntos  $(\epsilon_{ij_i}, \epsilon_{i(j_i+1)})$  son los centros de los intervalos adyacentes sobre los que se define las etiquetas  $X_{ij_i}$  y  $X_{i(j_i+1)}$ , y de forma que  $U$  puede obtenerse como

$$U = \bigcup_{k_1, \dots, k_M=1}^{L-1} C_{k_1 \dots k_M}. \quad (2.6)$$

Además cada intervalo de interpolación  $C_{k_1 \dots k_M}$  tendrá asociadas  $2^M$  reglas activas

$R_{j_1 \dots j_M}, R_{(j_1+1) \dots j_M}, \dots, R_{j_1 \dots (j_M+1)}, R_{(j_1+1) \dots (j_M+1)}$ , de modo que se cumple la relación  $(j_1 = k_1), \dots, (j_M = k_M)$  entre los subíndices. Este conjunto de reglas están definidas en la forma,

$$\begin{aligned}
R_{j_1 \dots j_M} : & \text{SI } x_1 \text{ es } X_{1j_1} \quad Y \dots Y \quad x_M \text{ es } X_{Mj_M} \\
& \text{ENTONCES } y = y_{j_1 \dots j_M}^* \\
R_{(j_1+1) \dots j_M} : & \text{SI } x_1 \text{ es } X_{1(j_1+1)} \quad Y \dots Y \quad x_M \text{ es } X_{Mj_M} \\
& \text{ENTONCES } y = y_{(j_1+1) \dots j_M}^* \\
& \dots \\
R_{j_1 \dots (j_M+1)} : & \text{SI } x_1 \text{ es } X_{1j_1} \quad Y \dots Y \quad x_M \text{ es } X_{M(j_M+1)} \\
& \text{ENTONCES } y = y_{j_1 \dots (j_M+1)}^* \\
R_{(j_1+1) \dots (j_M+1)} : & \text{SI } x_1 \text{ es } X_{1(j_1+1)} \quad Y \dots Y \quad x_M \text{ es } X_{M(j_M+1)} \\
& \text{ENTONCES } y = y_{(j_1+1) \dots (j_M+1)}^*
\end{aligned} \tag{2.7}$$

donde  $y_{j_1 \dots j_M}^*, y_{(j_1+1) \dots j_M}^*, \dots, y_{j_1 \dots (j_M+1)}^*$  e  $y_{(j_1+1) \dots (j_M+1)}^*$  son los correspondientes valores *singleton*.

El conjunto de reglas descrito en (2.7) constituye un sistema difuso simple sobre el subespacio de entrada determinado por  $C_{k_1 \dots k_M}$ , con idénticas características a las descritas para el caso bidimensional. Si llamamos  $y_{k_1 \dots k_M}$  a la salida de ese sistema simple, la salida del sistema global  $y$ , en cualquier punto de  $U$  podrá obtenerse a partir de las contribuciones de cada uno de esos intervalos,

$$y = \sum_{(L-1)^M \text{ términos}} y_{k_1 \dots k_M} \quad (1 \leq k_i \leq L-1) \text{ y } (1 \leq i \leq M) \tag{2.8}$$

en donde

$$y_{k_1 \dots k_M} = \sum_{2^M \text{ términos}} y_{j_1 \dots j_M}^* \cdot \left( w_{j_1 \dots j_M}^* \Big|_{C_{k_1 \dots k_M}} \right) \quad (j_i \leq j_i \leq (j_i+1)) \text{ y } (1 \leq i \leq M) \quad , \tag{2.9}$$

y donde la expresión  $w_{j_1 \dots j_M}^* \Big|_{C_{k_1 \dots k_M}}$  con  $(j_i \leq j_i \leq (j_i+1)) \text{ y } (1 \leq i \leq M)$  denota

las funciones base normalizadas, resultado de evaluar los antecedentes de reglas, con su soporte restringido al intervalo de interpolación  $C_{k_1 \dots k_M}$ .

Finalmente, y generalizando a partir del caso bidimensional, para especificar completamente el sistema simple definido en (2.9), que actúa sobre cada intervalo  $C_{k_1 \dots k_M}$  es necesario proporcionar dos conjuntos de parámetros:

Por una parte, el conjunto de  $M$  valores  $PA_{C_{k_1 \dots k_M}} = \{E_{k_1}, \dots, E_{k_i}, \dots, E_{k_M}\}$  que determina los antecedentes de regla, y por otra el conjunto de  $2^M$  valores *singleton*  $PC_{C_{k_1 \dots k_M}} = \left\{ y_{j_1 \dots j_M}^*, y_{(j_1+1) \dots j_M}^*, \dots, y_{j_1 \dots (j_M+1)}^*, y_{(j_1+1) \dots (j_M+1)}^* \right\}$  que determina los

consecuentes de regla.

En la Fig.2.3 se ilustra la idea de sistema de arquitectura multiplexada, y el procedimiento descrito anteriormente, mediante un diagrama de bloques general. Nótese además, que en cada intervalo de interpolación se tiene a su vez una partición en rejilla, con dos etiquetas por entrada, por lo que puede ser utilizada una arquitectura ANFIS del tipo de las presentadas en el apartado 4.5.2. del Capítulo 1, para realizar el sistema de inferencia difuso simple, como se representa en la Fig.2.3.

Para realizar un controlador genérico siguiendo esta arquitectura serán necesarios:

- $M \times 2$  bloques de función de pertenencia.
- $2^M$  bloques de operador mínimo de  $M$  entradas cada uno.
- Un bloque normalizador de  $2^M$  entradas.
- $2^M$  bloques de ponderación de *singleton*.
- Un bloque de agregación de  $2^M$  entradas.
- Un bloque identificador de intervalo.

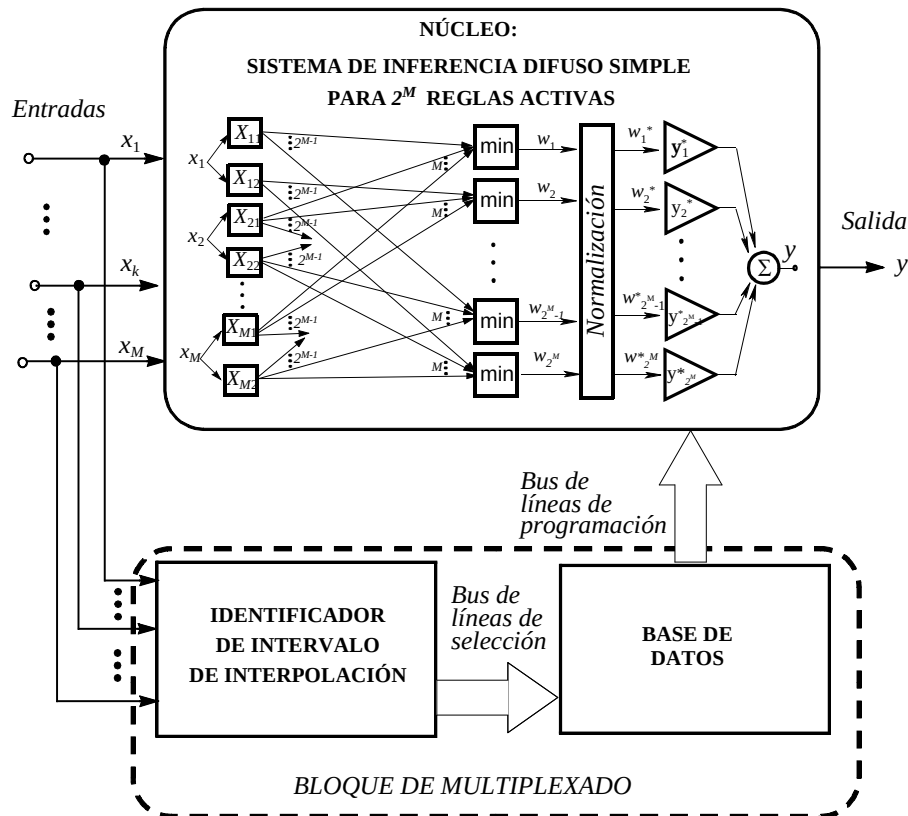


Fig.2.3 Diagrama de bloques de un sistema difuso de arquitectura multiplexada.



Y será necesario almacenar, determinar y ajustar:

- $(L - 1)^{2\dagger}$  parámetros de funciones de pertenencia.
- $L^M$  valores *singleton*.

Si en primera aproximación consideramos que tanto el tamaño, como el consumo de potencia del circuito que realiza el algoritmo difuso son función del número de reglas que se implementan físicamente, en el caso bidimensional, contemplado en el ejemplo que ha servido de base para el razonamiento, la ventaja de la propuesta es evidente. El sistema difuso ha de diseñarse para operar únicamente con cuatro reglas, frente a las  $L^2$  que hay que considerar para realizar un sistema que aborde el problema de forma global. Por su parte, en el caso general de un sistema con  $M$  entradas y  $L$  etiquetas por entrada, esto es  $N = L^M$  reglas, la ventaja resulta aún más evidente, pues el sistema de inferencia ha de trabajar solamente con  $2^M$ . Téngase en cuenta además que el tamaño de la *Base de Datos* que almacena los parámetros de programación ha de ser prácticamente la misma en cualquiera de los dos casos. Por su parte, por lo que respecta al bloque *Identificador de Intervalo*, su tamaño y consumo puede estimarse en función del número de bits necesarios para identificar a todos los intervalos de la partición, que es  $M \times i_s(\log_2 L)^{3\dagger}$ , por lo que su aportación al cómputo global resulta menos importante, cuanto mayor sea el valor de  $N$ .

En la Tabla 2.1 se resumen los datos sobre el número de bloques necesarios para la realización de cada una de las diferentes arquitecturas. Se evidencia así las ventajas de una arquitectura multiplexada.

**Tabla 2.1:** Número de bloques para diferentes arquitecturas

| Arquitectura | Funciones de Pertenencia | Operador $\min\{.\}$ /entradas | Entradas del Norma. | Agregación |
|--------------|--------------------------|--------------------------------|---------------------|------------|
| General      | $M \times N$             | $M \times N$                   | $N$                 | $N$        |
| En Rejilla   | $M \times L$             | $M \times N$                   | $N$                 | $N$        |
| Multiplexada | $M \times 2$             | $M \times 2^M$                 | $2^M$               | $2^M$      |

<sup>2</sup>†. Este valor resulta de considerar una partición en rejilla fija y uniforme, lo que significa que las particiones unidimensionales de cualquier entrada son idénticas.

<sup>3</sup>†.  $i_s(\log_2 L)$  denota al menor entero superior a  $\log_2 L$

### 3. PROPUESTA DE ARQUITECTURA PARA SISTEMAS DIFUSOS CON NÚCLEO ANALÓGICO MULTIPLEXADO

El empleo de núcleos analógicos en los co-procesadores difusos ya ha sido sugerido y utilizado con el fin de incrementar su ancho de banda con un menor consumo de área y potencia [Miki93]. Estas cualidades de las implementaciones difusas analógicas, junto a la ventaja que supone el ofrecer un interfaz directo a sensores y actuadores que ahorra el empleo de conversores A/D y D/A en las interfases, fueron destacadas en la Introducción.

Como también ya se ha señalado, la limitada precisión de las implementaciones analógicas, inferiores a nueve bits con tecnologías estándar [Pelg89], no supone un grave inconveniente para las aplicaciones prácticas del control difuso en las que es tolerable hasta de un 10% de precisión [Yama93]. Esa buena tolerancia lleva, en muchos casos, a no considerar esta cuestión en detalle durante la fase de diseño, argumentándose además que muchos de los errores, aleatorios o sistemáticos, que pueden aparecer tras el proceso de fabricación, pueden ser atenuados mediante el ajuste ulterior en campo de ciertos parámetros críticos, para los cuales se ha previsto puntos de acceso externo; o en el caso de controladores programables, mediante un ajuste guiado por algún algoritmo de aprendizaje. Un modelado apropiado de las fuentes de imprecisión [Pelg89] y el uso de una adecuada estrategia de diseño es siempre deseable, pues conduce, no solo a realizaciones con mejores precisiones, sino también al diseño de controladores que puedan ser programados de manera más robusta y transparente [Rodr99]. Sin embargo, a pesar de que cada uno de los diferentes bloques que forman el sistema global esté diseñado con el cuidado necesario para conseguir una muy buena precisión, el problema de la precisión del sistema completo puede no estar resuelto. Esta circunstancia es especialmente relevante en circuitos que realizan cálculos globales, como los que se han de realizar en el proceso de *defuzzificación* empleando el método del centro de gravedad en los controladores difusos. En estos circuitos, los parásitos y los errores (sistemáticos o aleatorios) causados por los circuitos que realizan los antecedentes de regla, se agregan, de modo que el error global final se hace mayor, lo que supone, entre otras cosas una degradación en su precisión. Este es uno de los principales obstáculos para incrementar la complejidad de los controladores difusos analógicos. Además, en general, el aumento de complejidad lleva aparejado una degradación de las buenas prestaciones de los sistemas analógicos en cuanto a consumo de potencia, velocidad de respuesta y ocupación de área de silicio. Esto es especialmente cierto cuando se emplea una partición en rejilla del universo de discurso, en las que, como ya hemos visto, hay una dependencia exponencial del número de reglas con el número de entradas.

Un esquema como el mostrado en la Fig.2.3 resulta bastante adecuado para desarrollar una arquitectura que permita cumplir con uno de los objetivos de este trabajo: el incremento del grado de complejidad de los problemas de control abordables con sistemas difusos

realizados con circuitos analógicos, preservando, en la medida de lo posible, las ventajas de dichas implementación. En dicho esquema, el *Núcleo* del sistema difuso puede ser realizado con circuitos analógicos, mientras que el *Bloque de Multiplexado* podría realizarse con técnicas de señal mixta.

El sistema de inferencia difuso que se emplea en el núcleo de la arquitectura multiplexada ha de ser diseñado para cumplir con la precisión requerida, pero resulta lo suficientemente simple, incluso para un número de entradas que puede ser considerado elevado desde el punto de vista de una implementación analógica, sobre todo teniendo en cuenta que el número total de reglas que podría manejar ese sistema. Como ejemplo numérico, supongamos un sistema con 6 entradas (pe. error en la posición, y su derivada, para un sistema con tres grados de libertad) y 5 etiquetas por entrada, (cinco es un número habitual en aplicaciones de control para expresar etiquetas lingüísticas como "Negativo Grande", "Negativo Pequeño", "Cero", "Positivo Pequeño", "Positivo Grande"). En este ejemplo, el sistema de inferencia difuso de la arquitectura multiplexada debe operar con  $2^6 = 64$  reglas, que aún puede resultar muy grande para las implementaciones analógicas, pero que resulta pequeño frente a las  $5^6 = 15625$  reglas que puede realizar el sistema global.

De esta forma, siguiendo el esquema de la Fig.2.3, para el ejemplo anterior, puede diseñarse un sistema difuso de alta complejidad, con las prestaciones y la precisión que pueda ser alcanzable en el diseño del *Núcleo Analógico*.

Dada la especial sensibilidad de los circuitos analógicos a la propagación y acumulación de errores resulta importante que, en el esquema planteado, la influencia de la arquitectura de multiplexado sea lo más pequeña posible sobre el procesador analógico, por ello se propone como requerimiento de diseño que, en la medida de lo posible y ante diversas alternativas, se opte por aquella solución que menos penalice el procesado realizado por el *Núcleo* del sistema difuso.

En la Fig.2.4 se muestra la arquitectura global propuesta en esta tesis y que es capaz de realizar el procedimiento de multiplexado descrito en el apartado 2.2., e ilustrado en la Fig.2.3, para un controlador con  $M$  entradas y  $L$  etiquetas por entrada (esto es, un total de  $L^M$  reglas). En esta propuesta se considera una partición en rejilla fija del espacio de entrada, por lo que, su capacidad de aproximación quedará determinada por las expresiones que hemos visto en el apartado 5.2. del Capítulo 1. Como también se ha visto en el apartado 4.5.2. de ese capítulo, esta opción resulta ventajosa desde el punto de vista de la implementación *hardware*, tanto por la simplificación en el número de bloques funcionales necesarios, como por la mayor simplicidad desde el punto de vista del usuario, debido a la reducción en el número de parámetros a considerar para configurar al sistema. Los diferentes bloques de alto nivel que en ella aparecen se describen brevemente a continuación:

- Núcleo Analógico:** Este es el bloque que realiza el procedimiento de inferencia difusa descrito en la expresión (1.3) del Capítulo 1, con las reglas activas en cada momento, y según la expresión (2.9). Recibe las entradas del sistema ( $x_1, \dots, x_k, \dots, x_M$ ) y genera la salida global  $y$ . Posee dos conjuntos de entradas de programación:  $PA_{C_{k_1 \dots k_M}} = \{E_1, \dots, E_k, \dots, E_M\}$  proporcionado por el bloque *Programador de Antecedente de Reglas* y  $PC_{C_{k_1 \dots k_M}} = \{y_1^*, \dots, y_l^*, \dots, y_{2^M}^*\}$  proporcionado por el bloque *Programador de Consecuente de Regla*, donde en ambos, el subíndice  $C_{k_1 \dots k_M}$  simboliza la dependencia de dicho conjunto del intervalo de interpolación considerado. Estos dos conjuntos de entradas configuran al *Núcleo Analógico* para trabajar con el conjunto de reglas activas. El primero configura las funciones de pertenencia asociadas a los antecedentes de regla, mientras que el segundo está constituido por los valores *singleton* asociados a los consecuentes de regla. El bloque se organiza en torno a dos bloques de diseño de alto nivel denominados *Bloque de Entrada* y *Bloque de Regla*, relacionado con los antecedentes y consecuentes de las reglas difusas, respectivamente. Además, la programación de las funciones de pertenencia se realiza mediante valores de tensión analógicos, mientras que la programación de los valores *singleton* lo es en forma digital. Nótese que en la Fig.2.4, cada  $y_l^*$  es una palabra de  $S$  bits.

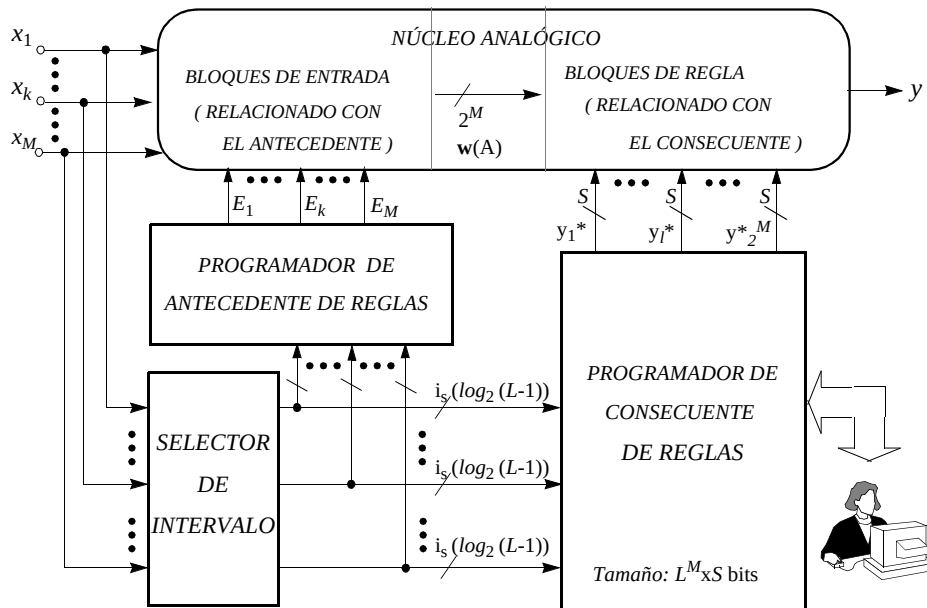


Fig.2.4 Arquitectura global de un Controlador difuso con Núcleo analógico Multiplexado.

- *Selector de Intervalo*: Este bloque establece la partición en rejilla del espacio de entrada, definida en (2.6), fijando por diseño el conjunto de valores  $\{\epsilon_{ij_i} | i = 1, 2, \dots, M, (j_i = 1, 2, \dots, L)\}$ , que determinan los límites de cada intervalo  $C_{k_1 \dots k_M}$  según (2.5). Proporciona, además la manera de identificar el intervalo de interpolación al que pertenece una entrada dada y así determinar el conjunto de reglas activas. Está formado por  $M$  convertidores analógico/digital, uno por entrada, con una resolución de  $i_s(\log_2(L-1))$  bits. En conjunto, proporcionan una palabra de  $M \times i_s(\log_2(L-1))$  bits, que codifica a cada uno de los intervalos de interpolación. Esta palabra digital es usada para seleccionar las funciones de pertenencia de los antecedentes de regla por el bloque *Programador de Antecedente de Reglas*, y para seleccionar los valores *singleton* apropiados en el bloque *Programador de Consecuente de Reglas*.
- *Programador de Antecedente de Reglas*: Este bloque selecciona, a partir de la palabra digital proporcionada por el bloque de *Selector de Intervalo*, el conjunto de valores analógicos de tensión  $PA_{C_{k_1 \dots k_M}} = \{E_1, \dots, E_k, \dots, E_M\}$  que configuran el *Bloque de Etiquetas del Núcleo Analógico* para operar con las funciones de pertenencia asociadas a las reglas activas en cada intervalo.
- *Programador de Consecuente de Reglas*: Este bloque está formado por una memoria digital que almacena los valores *singleton* asociados a los consecuentes de las reglas del sistema global. Proporciona, a partir de la palabra digital de selección generada por el bloque *Selector de Intervalo*, y de forma simultánea el conjunto de valores *singleton* de programación  $PC_{C_{k_1 \dots k_M}} = \{y_1^*, \dots, y_l^*, \dots, y_{2^M}^*\}$  que configuran el *Bloque de Reglas del Núcleo Analógico* para operar con los consecuentes de las reglas activas. Si cada valor *singleton* es representado por una palabra de  $S$  bits, el tamaño de la memoria será de  $L^M$  palabras de  $S$  bits. Cada lectura proporciona de forma simultánea y ordenada  $2^M$  palabras de  $S$  bits. Este bloque proporciona también un interfaz para accesos de lectura y escritura que constituye el interfaz externo de programación del sistema.

El sistema en conjunto funciona de manera asíncrona, de forma que los valores de parámetros de programación del *Núcleo Analógico*,  $(E_1, \dots, E_k, \dots, E_M)$  y  $(y_1^*, \dots, y_l^*, \dots, y_{2^M}^*)$  se obtienen de forma asíncrona una vez identificado el intervalo de interpolación correspondiente a un valor del vector de entrada  $(x_1, \dots, x_k, \dots, x_M)$  permaneciendo estables mientras el vector de entrada tome valores correspondientes a dicho intervalo; y cambian cada vez que un cambio en las entradas supone un cambio de intervalo de interpolación. Por otra parte, es un objetivo de diseño que el retardo entrada/salida del sistema se vea lo menos influenciado posible por el tiempo de procesado de la circuitería de multiplexado, y que coincida lo más posible con el retardo del algoritmo difuso por en *Núcleo Analógico*, para así cumplir con el objetivo de

preservar alta velocidad de funcionamiento de los controladores difusos puramente analógicos. Por tanto, uno de los principales requerimientos de diseño de la circuitería de multiplexado es la de realizar su función en un tiempo que resulte despreciable frente al tiempo de procesado en el *Núcleo Analógico*.

Es importante observar que la arquitectura mostrada en la Fig.2.4 resulta más ventajosa a medida que se incrementa el número de entradas y etiquetas. Por otra parte, el número de salidas del controlador podría ampliarse sin demasiado esfuerzo debido principalmente a que muchos de los bloques descritos podrían ser compartidos por la circuitería dedicada a generar cada salida. Más concretamente, cada salida adicional necesitaría un nuevo bloque *Programador de Consecuente de Reglas* y  $2^M$  *Bloques de Regla*, mientras que el resto de bloques no cambiaría.

En los siguientes capítulos se describirán con más detalle cada uno de los bloques funcionales que constituyen la arquitectura multiplexada, haciendo hincapié en las principales especificaciones de diseño, así como, la propuesta de diseño a nivel de circuito para tecnologías CMOS estándar, a partir de la cual ha sido diseñado un controlador ejemplo.

#### 4. CAPACIDAD DE ADAPTACION DE LA ARQUITECTURA PARA SISTEMAS DIFUSOS CON NUCLEO ANALÓGICO MULTIPLEXADO

En este apartado se discuten las diferentes posibilidades de reconfiguración y adaptabilidad de la arquitectura para sistemas difusos con núcleo analógico multiplexado.

Dado que, tal y como se ha definido esta propuesta en el apartado 3. de este capítulo, la partición en rejilla del espacio de entrada queda fijada por diseño, los únicos parámetros accesibles y que quedan por ser ajustados, y por tanto los que proporcionan flexibilidad y cierta capacidad de adaptación a esta arquitectura, son los correspondientes a los valores *singleton* que determinan los consecuentes de las reglas.

Esta restricción en la flexibilidad, frente a un caso más general que permitiera reconfigurar también los antecedentes de regla, se ve compensada por una mayor simplicidad y modularidad en cuanto a diseño y por una facilidad en cuanto a manejo desde el punto de vista de un posible usuario. Además, como ya se ha visto en el apartado 5.2. del Capítulo 1, su capacidad de aproximación no resulta perjudicada, siempre que se haya considerado un número suficiente de reglas que garantice la precisión requerida para un conjunto suficientemente amplio de superficies de control, según indica la expresión (1.20) del Capítulo 1. Por otra parte, al error en la aproximación, consecuencia del límite en la precisión que resulta de la partición realizada, hay que añadir el error de cuantización consecuencia del limitado número de bits empleado para representar a los valores *singleton*, ( $S$  en la Fig.2.4) .

La expresión (1.20) del Capítulo 1, también puede ser interpretada en el sentido de que, al fijar la partición del espacio de entrada, ésto es, dado el valor de  $h$ , también queda determinado que condiciones han de cumplir las funciones  $g(\mathbf{x})$  que pueden ser aproximadas por el sistema difuso  $f(\mathbf{x})$  para una precisión dada  $\varepsilon$ . Esta condición resulta ser

$$\sum_{i=1}^M \left\| \frac{\partial g}{\partial x_i} \right\|_{\infty} \leq \frac{\varepsilon}{h} \quad (2.10)$$

donde  $\| \bullet \|_{\infty}$  denota la norma del supremo como ya fue definida en el apartado 5.2. del Capítulo 1. Por tanto, bajo esta segunda interpretación, la tarea de configurar un sistema concreto para aproximar una superficie de control, que se sabe que puede ser aproximada por él con un grado de precisión también dado, consiste en proporcionar al sistema los valores *singleton* adecuados para conseguir que el sistema trabaje de forma óptima.

En general para esta tarea se presentan dos alternativas, dependiendo de la información disponible sobre la superficie de control que se desee aproximar: *Programación directa* y/o *Aprendizaje* de los parámetros que permiten configurar al sistema. Ambas opciones son posibles para esta arquitectura como veremos a continuación.

#### 4.1. Programación directa de los valores *singleton*

Mediante programación directa, el sistema es configurado para interpolar la superficie de control considerada, a partir de un conjunto de  $N$  muestras, donde  $N$  coincide con el número total de reglas de sistema difuso. Esta opción resulta adecuada cuando es posible generar arbitrariamente parejas de datos entrada salida  $(\mathbf{x}^k, f(\mathbf{x}^k))$ ; bien porque se tiene una expresión analítica conocida para la función  $f(\mathbf{x})$ , o bien porque, aunque no se conoce la expresión analítica de  $f(\mathbf{x})$ , es posible tener acceso al sistema de forma que, para cualquier vector de entrada  $\mathbf{x}^k$ , es posible medir su respuesta  $f(\mathbf{x}^k)$ .

Por tanto, en esta situación, es posible conocer el valor exacto que han de tomar todos los valores *singleton* a partir de muestras de la función a aproximar. Si, como ejemplo, consideramos el sistema difuso y definido en el apartado 2.2. como el sistema que ha de aproximar una función dada  $f(\mathbf{x})$ , se tendrá que cada valor *singleton*  $y_{j_1 \dots j_M}^*$  puede obtenerse según la expresión

$$y_{j_1 \dots j_M}^* = y(\mathbf{x}^{j_1 \dots j_M}) = f(\mathbf{x}^{j_1 \dots j_M}),$$

donde cada vector  $\mathbf{x}^{j_1 \dots j_M} = (e_1^{j_1}, \dots, e_M^{j_M})$ , representa al punto del espacio de entrada para el cual la función base de la regla correspondiente al índice  $j_1 \dots j_M$  toma su valor máximo, esto es, el punto para el cual se verifica que  $w_{j_1 \dots j_M}^*(e_1^{j_1}, \dots, e_M^{j_M}) = 1$ . Estos puntos son los que, para el sistema difuso considerado, definen la partición del espacio de entrada. Por su parte,

$f(\mathbf{x}^{j_1 \dots j_M})$  es una muestra de la superficie que se desea aproximar en punto  $\mathbf{x}^{j_1 \dots j_M}$  considerado.

#### 4.2. Aprendizaje de los valores *singleton*

Aun cuando la función a aproximar,  $f(\mathbf{x})$ , no sea conocida, sí es posible disponer de un número limitado de parejas de datos entrada salida  $((\mathbf{x}^k, f(\mathbf{x}^k)))$ , aunque éstas no puedan ser escogidas arbitrariamente, o incluso no correspondan a ninguno de los vectores  $\mathbf{x}^{j_1 \dots j_M} = (e_1^{j_1}, \dots, e_M^{j_M})$ , la arquitectura propuesta sigue manteniendo su validez y los valores singleton puede ser obtenidos empleando el procedimiento de aprendizaje propio de las redes neuronales que se describe a continuación.

Como ya se ha visto una de las características del sistema de inferencia difuso considerado es que obtiene la superficie de control como una combinación lineal de funciones base, (según la expresión (1.3) del Capítulo1) y que éstas coinciden con las funciones de pertenencia multidimensionales resultado de la composición de los antecedentes de regla. Esta circunstancia permite reconfigurar la arquitectura de la Fig.2.4 de una forma más general, como aparece en la Fig.2.5(a), donde las funciones de pertenencia se suponen normalizadas, (Fig.2.5(b)). Este punto de vista coincide con otros en el campo de la Inteligencia Artificial, donde se emplean reglas y algoritmos de aprendizaje. En particular se ha demostrado la equivalencia funcional entre las RBFN (Radial-Basis-Function-Network) y el algoritmo, aquí considerado [Jang93a] [Spec91].

El aprendizaje en estas redes se realiza en dos pasos [Hush93]:

- Aprendizaje de la primera capa (no lineal)
- Aprendizaje de la capa de salida (lineal)

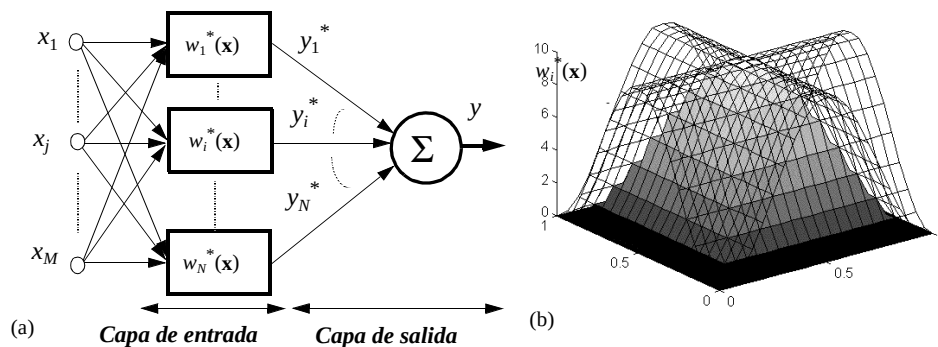


Fig.2.5 (a) Arquitectura en dos capas para un controlador difuso; (b) salida de antecedente de regla normalizada para un caso bidimensional.



Como en la arquitectura considerada (Fig.2.4) solo es posible ajustar los valores *singleton*, asociados a los consecuentes de regla, debido a que se ha considerado una partición fija del espacio de entrada, el único aprendizaje posible es el de la capa de salida.

Supongamos que se tiene un conjunto de  $K$  parejas de datos de entrenamiento  $\{\mathbf{x}^k, y_d(\mathbf{x}^k)\} (k = 1, \dots, K)$ , donde  $y_d(\mathbf{x}^k)$  es la salida proporcionada por el sistema difuso a la entrada de entrenamiento. Si se introducen esas parejas en la expresión (1.3) del Capítulo 1, se obtiene un conjunto de  $K$  ecuaciones con  $K$  incógnitas, las cuales se corresponden con los valores *singleton*. Por tanto, para el controlador de la Fig.2.4 puede escribirse

$$\mathbf{Y}_d = \mathbf{W}^* \mathbf{Y}^* \quad (2.11)$$

donde  $\mathbf{Y}_d$  es un vector columna de dimensión  $K$  que contiene las  $K$  salidas proporcionadas por las parejas de entrenamiento,  $\mathbf{W}^*$  es una matriz de dimensión  $K \times N$  que contiene las salidas de las  $N$  funciones de pertenencia multidimensionales para cada pareja de datos de entrenamiento, e  $\mathbf{Y}^*$  es un vector columna que contiene los valores *singleton* a determinar  $(y_1^*, \dots, y_i^* \dots y_N^*)$ .

La mejor solución para (2.11) se estima minimizando el error cuadrático medio (LMS)  $\|\mathbf{W}^* \mathbf{Y}^* - \mathbf{Y}_d\|^2$  y viene dada por,

$$\mathbf{Y}^* = (\mathbf{W}^{*T} \mathbf{W}^*)^{-1} \mathbf{W}^{*T} \mathbf{Y}_d \quad (2.12)$$

la cual es computada de forma recursiva como se hace en [Perk92] [Jang95].

El anterior algoritmo procesa los datos de entrenamiento por lotes, esto es, los parámetros (*singletons*) son actualizados en bloque, después de que hayan sido procesados todos los datos de entrenamiento en conjunto. Otra alternativa, más simple, es considerar una *regla de aprendizaje incremental* (LMS), donde los parámetros son actualizados cada vez que una pareja de datos de entrenamiento es procesada [Hush93], de forma que

$$y_{i \text{ new}}^* = y_{i \text{ old}}^* + \mu (f(\mathbf{x}^k) - y_d(\mathbf{x}^k)) w_i^*(\mathbf{x}^k) \quad i = 1, \dots, N \quad (2.13)$$

donde  $\mu$  es una constante. Es interesante observar que la expresión (2.13) también se puede obtener a partir de una *regla del gradiente* [Nie93]. Obsérvese que, en esta expresión, el incremento,  $f(\mathbf{x}^k) - y_d(\mathbf{x}^k)$ , que sufre el valor *singleton* que está siendo actualizado a partir de la entrada  $\mathbf{x}^k$ , está ponderado por el grado de pertenencia de dicha entrada, a su propio conjunto difuso, a través del término  $w_i^*(\mathbf{x}^k)$ . De esta manera, el valor *singleton* es ajustado en función de la influencia que la muestra  $(\mathbf{x}^k, y_d(\mathbf{x}^k))$  tenga en la salida.

Otro modo de considerar esa dependencia consiste en actualizar solamente el valor *singleton* cuyo antecedente de regla sea máximo para la entrada  $\mathbf{x}^k$  considerada, de forma que

$$y_{inew}^* = y_{iold}^* + \mu (f(\mathbf{x}^k) - y_d(\mathbf{x}^k)) \quad (2.14)$$

donde  $i$  es el índice de la regla que cumple que  $w_i^*(\mathbf{x}^k) = \max(w_1^*(\mathbf{x}^k), w_2^*(\mathbf{x}^k), \dots, w_N^*(\mathbf{x}^k))$ . Esta expresión coincide con la de la regla incremental denominada *out-star*, que es empleada para entrenar la capa de Grossberg en la red denominada *counterpropagation*. Aunque esta última es utilizada principalmente en problemas de *clustering*, también puede ser utilizada para aproximar funciones [Zura92]. El principio es el mismo que en el caso de la arquitectura general de la Fig.2.5(a), aunque en esta red, los conjunto de la primera capa (denominada capa de Kohonen) son valores nítidos (*crisp*) en lugar de conjuntos difusos. La simplicidad de la regla (2.14) la convierte en un buen candidato para ser implementada en silicio y resulta también muy adecuada para entrenar un sistema como el propuesto en la Fig.2.4 [Vida96b].

El parámetro  $\mu$  in (2.14) determina la velocidad del aprendizaje. Dado que para la actualización de los valores *singletons* se tendrá un incremento mínimo, que podrá ser expresado como  $\Delta y_{imin}^* = (y_{inew}^* - y_{iold}^*)_{min}$ , debido principalmente a la limitación en la precisión en la representación de los propios valores *singleton*, la constante  $\mu$  debe ser seleccionada de forma que

$$\mu \geq \frac{\Delta y_{imin}^*}{\varepsilon_{ymin}} \quad (2.15)$$

donde  $\varepsilon_{ymin} = |f(\mathbf{x}^k) - y_d(\mathbf{x}^k)|_{min} \forall k = 1, \dots, K$  es la mínima desviación posible a la salida entre el valor deseado  $f(\mathbf{x}^k)$  y el obtenido  $y_d(\mathbf{x}^k)$ , para cualquier entrada  $\mathbf{x}^k$ , que puede ser trasladada a un incremento en el valor *singleton*. En otras palabras, el proceso de aprendizaje se detiene y los parámetros (*singletons*) dejan de ser actualizados, cuando este valor mínimo es alcanzado.

Para ilustrar el modo de funcionamiento de este esquema de aprendizaje, se presentan a continuación los resultados del entrenamiento de un controlador (modelado por *software*) con una partición en rejilla del espacio de entrada, y seis etiquetas por entrada uniformemente distribuidas en el espacio de entrada. El conjunto de parejas de entrada/salida que sirven para el entrenamiento se han obtenido a partir de la función  $f(x_1, x_2) = 2 + \sin \pi x_1 \sin \pi x_2$ . Se han considerado un total de  $11 \times 11$  muestras para cubrir un intervalo de entrada  $[0, 2] \times [0, 2]$ . El sistema fue entrenado utilizando la regla *out-star*, descrita en (2.14), de forma que los valores *singleton* fueron actualizados a partir de un valor inicial de 1 para todos ellos. Los resultados se muestran en la Fig.2.6.

La Fig.2.6(a) muestra la función objetivo que ha de ser aproximada a partir de la matriz de datos muestreados, mientras que la Fig.2.6(b) muestra la salida del controlador después de 500 épocas, esto es, después de que el conjunto completo de muestras haya sido empleado para entrenar al sistema 500 veces. El error cuadrático medio final (RMSE) fue 0.067 y su evolución se muestra en la Fig.2.6(d), mientras que el error final,  $f(\mathbf{x}^k) - y_d(\mathbf{x}^k)$  aparece en la

Fig.2.6(c). Finalmente, para considerar la limitada resolución en los valores *singleton*, estos fueron truncados para obtener una precisión de cuatro bits <sup>4†</sup>. En este caso el RMSE obtenido fue 0.078 y la superficie de error es la que se muestra en la Fig.2.6(e).

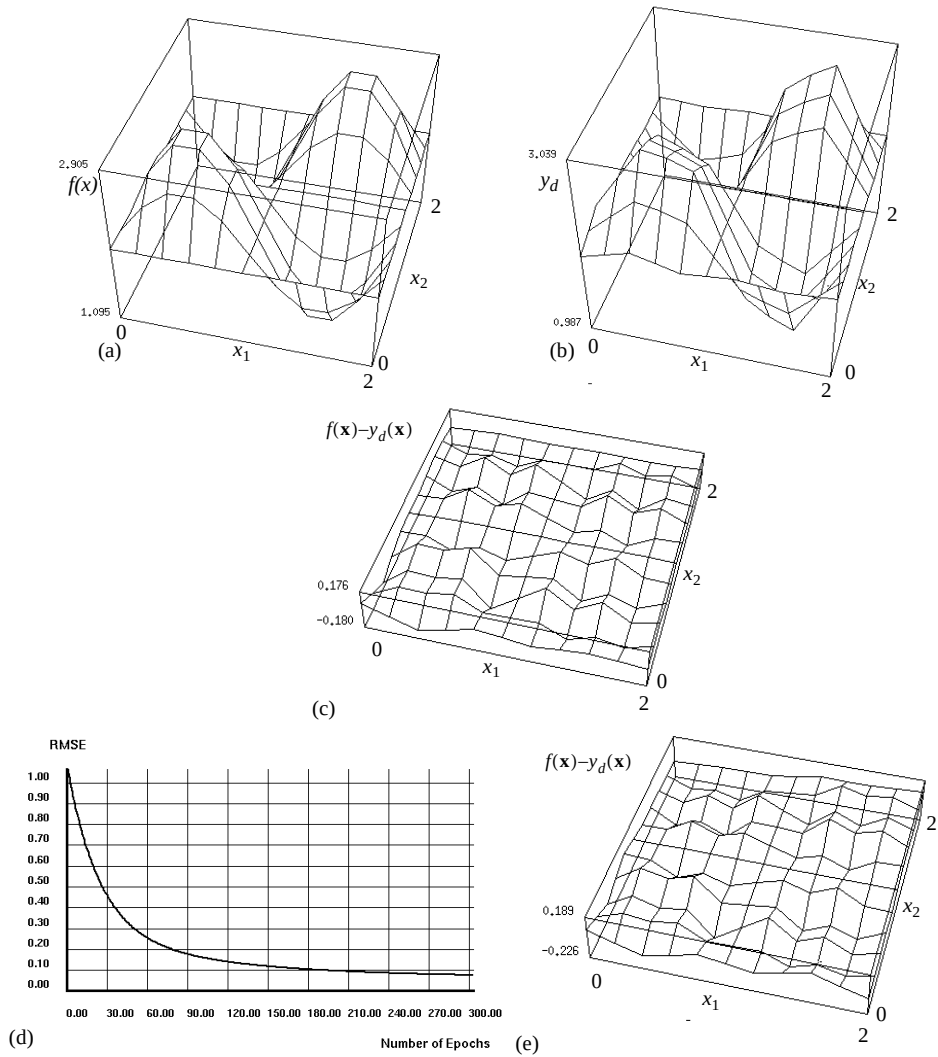


**Fig.2.6 Ejemplo de aprendizaje de valores *singleton*:** (a) Superficie a aproximar a partir de  $11 \times 11$  muestras; (b) superficie aprendida con la regla outstar a partir de una partición en rejilla y seis etiquetas por entrada; (c) superficie de error; (d) Curva de evolución del RMSE; (e) superficie de error con truncamiento de valores *singleton* (cuatro bits).

<sup>4†</sup>. Cuatro bits es la precisión considerada en el chip que ha sido implementado en esta tesis como ejemplo de viabilidad de la arquitectura multiplexada.

## Capítulo 3

### Descripción Funcional y Diseño CMOS del Núcleo Analógico

#### 1. INTRODUCCIÓN

Los capítulos 3, 4 y 5 están dedicados a realizar una descripción más detallada de los diferentes bloques que constituyen la Arquitectura para Controladores Difusos con Núcleo Analógico Multiplexado propuesta en el Capítulo 2, y que ha sido esbozada de forma general en el apartado 3. de ese capítulo. Para cada uno de ellos se realiza en primer lugar una descripción funcional en la que se destacan y discuten sus principales requerimientos y especificaciones, y a continuación se describen y analizan los circuitos empleados en el diseño del controlador difuso ejemplo MFCON, que ha sido diseñado para mostrar la viabilidad de la propuesta. Este controlador posee dos entradas, una salida y sesenta y cuatro reglas, y ha sido fabricado con tecnología CMOS estándar de  $0.7\mu\text{m}$ .

Este capítulo está dedicado al bloque denominado *Núcleo Analógico*, que es el encargado de realizar, mediante técnicas analógicas el algoritmo de inferencia difusa en cada región del espacio de entrada definida por la partición en rejilla que se ha considerado.

#### 2. NÚCLEO ANALÓGICO

Tal y como se indica en el apartado 3. del Capítulo 2, el *Núcleo Analógico* es el bloque encargado de realizar el sistema de inferencia difusa de Takagi-Sugeno descrito en la expresión (1.3) del Capítulo 1, con las  $2^M$  reglas, que constituyen el *sistema de inferencia más simple posible* en cada intervalo de interpolación  $C_{k_1 \dots k_M}$  en que se divide el espacio de entrada del sistema global. Este sistema difuso queda definido según la expresión (2.9) del Capítulo 2. Por tanto, este bloque recibe como entradas, el vector de entrada del sistema global  $(x_1, \dots, x_k, \dots, x_M)$  y genera una salida  $y_{k_1 \dots k_M}$  que a la postre constituye la salida del sistema global y, según recoge la expresión (2.8) del Capítulo 2. Este bloque posee además dos conjuntos de entradas de programación:  $PA_{C_{k_1 \dots k_M}} = \{E_1, \dots, E_k, \dots, E_M\}$  proporcionado por el bloque *Programador de Antecedente de Reglas* y  $PC_{C_{k_1 \dots k_M}} = \{y_1^*, \dots, y_l^*, \dots, y_{2^M}^*\}$  proporcionado por el bloque *Programador de Consecuente de Regla*; donde en ambos, el subíndice  $C_{k_1 \dots k_M}$  simboliza la dependencia de dicho conjunto del intervalo de interpolación considerado. Estos dos conjuntos de entradas proporcionan los valores de los parámetros que configuran al *Núcleo Analógico* para trabajar con el conjunto de reglas activas en cada intervalo

$C_{k_1 \dots k_M}$ . El primero configura las funciones de pertenencia asociadas a los antecedentes de regla, mientras que el segundo está constituido por los valores *singleton* asociados a los consecuentes de regla. La Fig.3.1 resalta el bloque *Núcleo Analógico* en el esquema general del *Controlador Difuso con Núcleo Analógico Multiplexado*, e ilustra además las señales, antes mencionadas, que constituyen su interfaz con el resto de bloques.

## 2.1. Discusión de los requerimientos y objetivos de diseño.

Dado que el sistema de inferencia difusa que realiza el *Núcleo Analógico* ha de poder operar en cualquiera de los intervalos en los que se ha dividido el espacio de entrada, su principal requerimiento de diseño recae en la elección para su realización de una arquitectura que sea lo suficientemente flexible para ser reconfigurada dinámicamente, conforme cambian las entradas y lo más rápidamente posible, para seguir así su evolución.

## 2.2. Propuesta de diseño

Para cumplir con los objetivos anteriormente mencionados, se propone diseñar el *Núcleo Analógico* siguiendo el modelo de arquitectura modular programable para controladores neuro-difusos propuesta en [Vida96b].

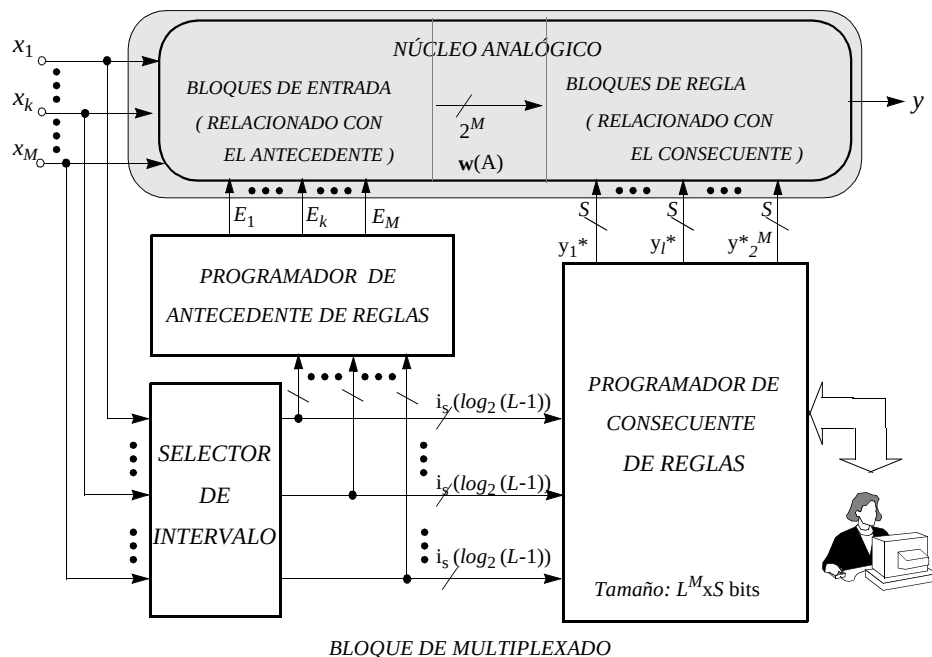


Fig.3.1 Núcleo Analógico en la arquitectura global del controlador difuso con núcleo analógico multiplexado

En la Fig.3.2 se muestra la arquitectura conceptual neuro-difusa ANFIS del *Núcleo Analógico* para un sistema genérico con  $M$  entradas, y por tanto,  $2^M$  reglas activas. En ella aparecen sobreimpresos con una flecha aquellos nodos que son reconfigurables y que proporcionan al sistema la flexibilidad requerida, y que son los nodos de la capa que realiza las funciones de pertenencia, y los nodos de la capa en la que se realiza la ponderación con valores *singleton*.

Siguiendo el modelo del controlador presentado en [Rodr99], resulta muy conveniente organizar la arquitectura de diseño de este sistema en torno a dos bloques de alto nivel, denominados *Bloque de Etiqueta* y *Bloque de Regla*, sobre los que se redistribuyen y agrupan los nodos de la estructura en capas de la Fig.3.2, a fin de explotar algunas características propias de los circuitos empleados en su realización consiguiendo así un diseño muy modular.

En la organización propuesta en [Rodr99], reproducida en la Fig.3.3 como referencia para el caso del *Núcleo Analógico* aquí considerado, a cada entrada  $x_i$  con ( $1 \leq i \leq M$ ) le corresponden dos etiquetas difusas  $X_{i1}$ ,  $X_{i2}$ , cada una de las cuales lleva asociada un *Bloque de Etiqueta*. En cada uno de estos bloques se realizan las siguientes operaciones:

- Se evalúa la correspondiente función de pertenencia,  $s_{i1}(x_i)$  o  $s_{i2}(x_i)$ .
- Se genera  $2^{M-1}$  réplicas del resultado.
- Cada una de estas réplicas se procesa en un bloque "min-inp", generando una salida.

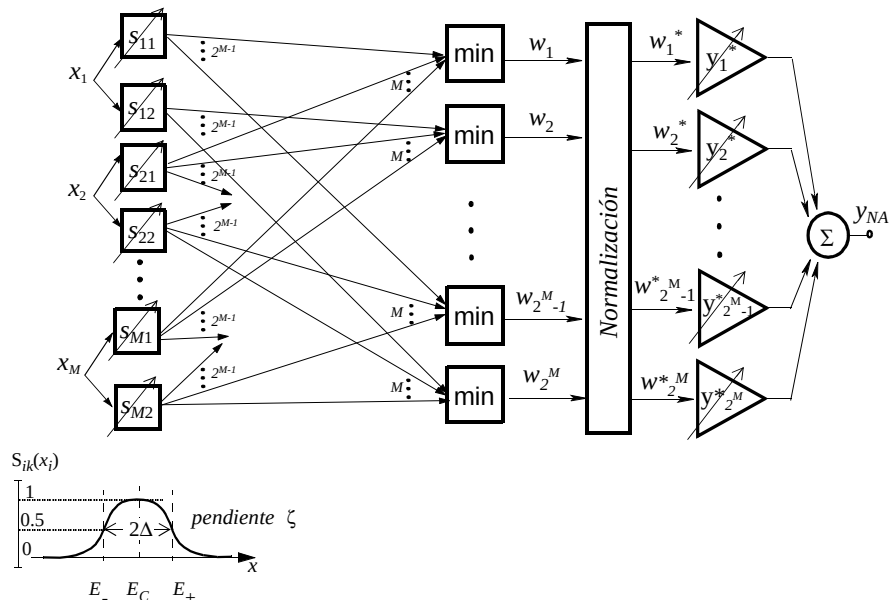


Fig.3.2 Arquitectura conceptual ANFIS del Núcleo Analógico para  $2^M$  reglas activas.

Cada bloque "*min-inp*" constituye la etapa de entrada de un circuito que realiza el operador de mínimo y todos realizan la misma operación. Hay  $2^{M-1}$  bloques de este tipo en cada *Bloque de Etiqueta*. En el caso general se tiene  $M \times 2$  *Bloques de Etiqueta*, cada uno de ellos con su salida replicada  $2^{M-1}$  veces.

Por cada una de las  $2^M$  reglas activas hay un *Bloque de Regla*, cada uno de los cuales recibe  $M$  entradas. Cada una de estas  $M$  entradas proviene a su vez de un *Bloque de Etiqueta* asociado a una entrada  $x_i$  diferente. En cada *Bloque de Regla* se realizan las siguientes operaciones:

- Se completa el operador mínimo de  $M$  entradas en el bloque "*min output*", evaluando la función de pertenencia  $m$ -dimensional  $w_k(\mathbf{x})$ .
- Se evalúa su valor normalizado  $w_k^*(\mathbf{x})$  en el bloque "*norm*".
- Se realiza su ponderación con los valores *singleton* para obtener  $y_k^* w_k^*(\mathbf{x})$  en el bloque "*ponderación*".

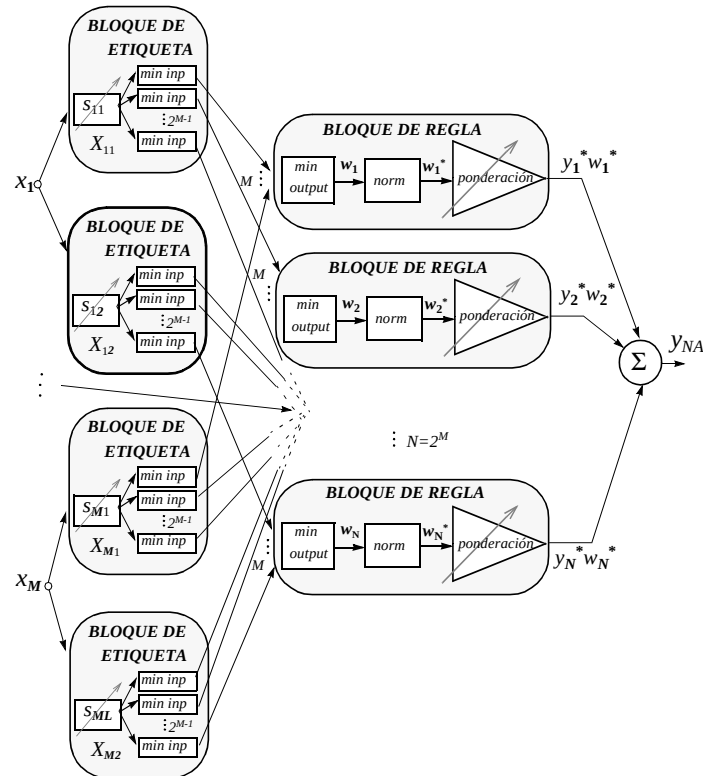


Fig.3.3 Arquitectura del Núcleo Analógico siguiendo el modelo de la propuesta [Rodr99]

Finalmente, la agregación de la salidas de los *Bloques de Regla* se realiza en el nodo de salida obteniendo,

$$y_{NA} = \sum_{k=1, 2^M} y_k^* w_k^*(\mathbf{x}) \quad (3.1)$$

La programación del *ancho* ( $2\Delta$ ), y el *centro* ( $E_C$ ) de cada función de pertenencia se realiza mediante valores de tensión analógicos, mientras que la *pendiente* ( $\zeta$ ) es fijada por diseño. Por su parte, la programación de los valores *singleton* se hace de forma digital. Para el caso del sistema del *Núcleo Analógico*, ( $2\Delta$ ) se considera un valor constante, al considerarse para el sistema global una partición en rejilla fija de módulo  $h^{-1\ddagger}$ , por lo que sólo se necesita programar el parámetro ( $E_C$ ) de cada *Bloque de Etiqueta* para cada intervalo de interpolación.

La traslación de la arquitectura de la Fig.3.3 al diseño del *Núcleo Analógico* no se ha realizado de forma directa en esta propuesta, sino que se han transformado algunos aspectos de los bloques básicos, para permitir, por una parte, que el *Núcleo Analógico* pueda ser utilizado en todo el rango de entrada sin introducir error, y por otra sacar partido de las características particulares de esta implementación a fin de simplificar al máximo el diseño.

Dentro de la primera categoría de transformaciones se encuentra la modificación del circuito de normalización. En la propuesta de [Vida96b] este circuito permitía el paso de una intensidad no nula entre la salida de los antecedentes de regla y entrada del circuito de normalización  $w_k(\mathbf{x})$ , aún cuando ésta debía ser mínima. Esta forma de operar no tiene mayor consecuencia que la de tener que retirar una intensidad de desplazamiento a la salida global del sistema. Sin embargo, esta misma estrategia empleada en el *Núcleo Analógico* da como resultado una intensidad de desplazamiento diferente para cada intervalo de interpolación  $C_{k_1 \dots k_M}$ , lo que la hace inviable. El circuito de normalización se diseña aquí para proporcionar salida nula a las reglas con contribución mínima, lo que evita el problema.

En cuanto a aquellas modificaciones tendentes a aprovechar las características particulares del nuevo enfoque, la principal es aquella que saca ventaja de la forma de las funciones de pertenencia del *sistema difuso más simple posible* que realiza el *Núcleo Analógico*, para simplificar el diseño del *Bloque de Etiqueta*. A modo de ilustración considérese el ejemplo de un sistema global bidimensional, cuya partición del espacio de entrada en intervalos de interpolación se muestra en la Fig.3.4(a). La Fig.3.4(b) muestra las cuatro etiquetas,  $X_{1a}$  y  $X_{1b}$ , y  $X_{2a}$  y  $X_{2b}$ , dos por entrada,  $x_1$  y  $x_2$ , del *sistema de inferencia difuso más simple* definido sobre el intervalo de interpolación genérico  $C_{mn}$  y sus correspondientes reglas activas  $R_1$ ,  $R_2$ ,  $R_3$ , y  $R_4$ . Cada una de estas etiquetas se obtienen a partir de las dos etiquetas definidas para cada una de las entradas en el intervalo  $C_{mn}$  en el sistema global, véanse los trozos de las etiquetas genéricas  $X_{1m}$  y  $X_{1(m+1)}$ , y  $X_{2n}$  y  $X_{2(n+1)}$  marcados en grueso en la

1 $\ddagger$ . Según ha sido definido en apartado 5.2. del Capítulo 1.



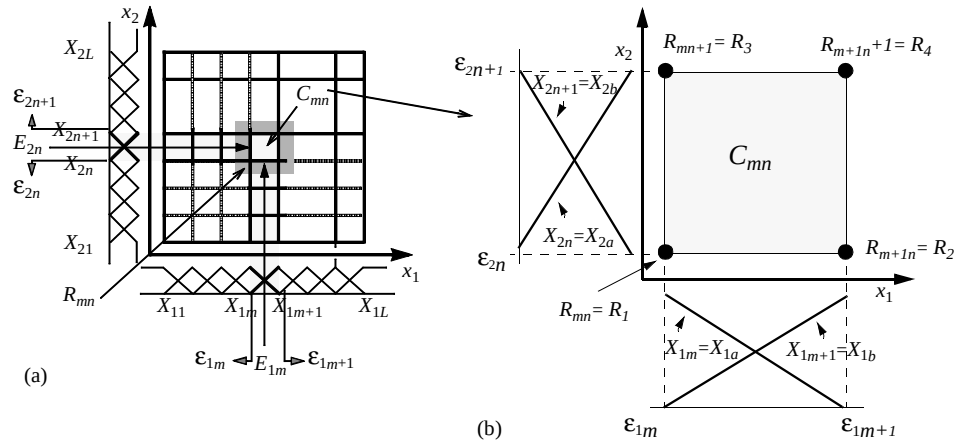


**Fig.3.4** Partición en intervalos de interpolación del universo de discurso del sistema global (a); universo de discurso y funciones de pertenencia del sistema de inferencia difuso más simple posible sobre un intervalo genérico  $C_{mn}$  (b).

Fig.3.4(a) cuyo soporte queda incluido en el intervalo genérico  $C_{mn}$ . Además a cada etiqueta le corresponde una función de pertenencia  $s_{ia}(x_i)$  y  $s_{ib}(x_i)$ , las cuales resultan ser la una la complementaria de la otra para cada una de las entradas, esto es, se verifica que  $s_{ia}(x_i) = \bar{s}_{ib}(x_i)$  y viceversa. Por tanto, el diseño del *Bloque de Etiqueta* puede replantearse de forma que, en lugar de generar y evaluar una única función de pertenencia unidimensional  $s_{i1}(x_i)$  o  $s_{i2}(x_i)$ , genere y evalúe simultáneamente los dos trozos complementarios correspondientes a cada una de las dos etiquetas lingüísticas de cada entrada en el intervalo considerado. Esta modificación permite reducir el número de *Bloques de Etiqueta* a uno por entrada. En esta propuesta a este nuevo bloque se le ha denominado *Bloque de Entrada*. Además, como se verá con más detalle más adelante, esta circunstancia permite utilizar un simple par diferencial como núcleo del circuito que genera las funciones de pertenencia, lo que supone una importante reducción en la complejidad y extensión de este bloque. La realización de este nuevo *Bloque de Entrada*, así como el análisis de las ventajas que ello supone a nivel de circuito serán consideradas en este capítulo, en los apartados dedicados al diseño de cada uno de los bloques que constituyen el *Núcleo Analógico*.

Finalmente, por lo que respecta al *Bloque de Regla*, desde el punto de vista funcional coincide con el de la propuesta de la Fig.3.3. Las principales modificaciones introducidas lo son a nivel de circuito, están encaminadas a mejorar la respuesta dinámica del circuito de ponderación, y son consecuencia del carácter dinámico de la programación de los valores *singleton*. Estas modificaciones serán consideradas también más adelante en este capítulo.

### 2.3. Diseño CMOS del Núcleo Analógico

En general para construir un *Núcleo Analógico* que realice un sistema difuso de  $M$  entradas y  $2^M$  reglas activas será necesario emplear  $M$  *Bloques de Entrada* y  $2^M$  *Bloques de Regla*. Dado que la mayoría de los circuitos básicos que conforman los *Bloques de Entrada* y los *Bloques de Regla* han sido bien descritos y caracterizados en [Vida96b], y dado que las aportaciones aquí presentadas se aprecian mejor en un nivel algo superior, en lo que sigue, el *Núcleo Analógico* se va a describir en términos de estos dos bloques de alto nivel, en lugar de realizar una descripción de los circuitos que realizan cada uno de los operadores que intervienen en el algoritmo de inferencia difusa. En el controlador ejemplo MFCON el *Núcleo Analógico* se ha construido a partir de estos bloques tal y como muestra la Fig.3.5. Se necesita un *Bloque de Entrada* por entrada en el sistema y un *Bloque de Regla* por regla activa a implementar, en este caso cuatro reglas activas. A partir de estos bloques de alto nivel, el núcleo se obtiene por simple cableado, lo que da idea de su modularidad. A continuación se describen los principales aspectos de diseño de ambos bloques por separado, y por último se abordan las consideraciones de diseño que les son comunes para la realización del *Núcleo Analógico* completo.

#### 2.3.1. Diseño del Bloque de Entrada

La Fig.3.6 muestra un esquema de las dos etapas fundamentales que constituyen el *Bloque de Entrada* del *Núcleo Analógico*. Estos son: el *Circuito de Funciones de Pertenencia*, (CFP), y las *Celdas de entrada del circuito de mínimo*, "min-inp". Como muestra dicha figura, este bloque posee dos entradas, una de ellas es una de las entradas externas del controlador,  $x_i$ ; y la otra la correspondiente línea de programación  $E_i$ , proporcionada por el bloque *Programador de Antecedente de Reglas*. Como líneas de salida proporciona dos conjuntos de  $2^{M-1}$  líneas cada uno, cada una de las cuales es a su vez salida de una de las *Celdas de entrada*

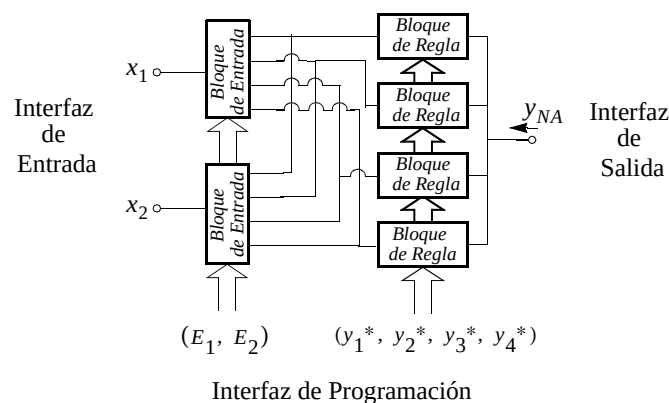


Fig.3.5 Construcción del núcleo analógico a partir de los bloques de alto nivel *Bloque de Entrada* y *Bloque de Regla* en el controlador ejemplo MFCON.

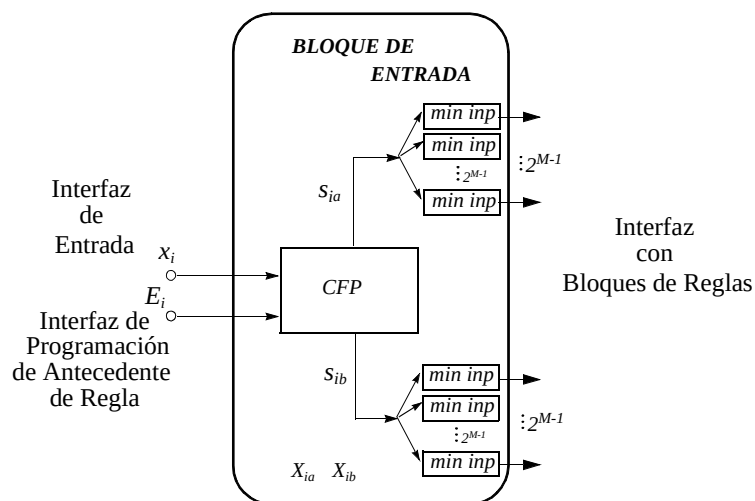


**Fig.3.6 Bloque de Entrada del Núcleo Analógico: Etapas constitutivas.**

del circuito de mínimo, en una situación idéntica a la descrita para el *Bloque de Etiqueta* de la arquitectura de la Fig.3.3. Para entender el diseño del *Bloque de Entrada*, y especialmente de su etapa *CFP*, recordemos que, como se ha mencionado más arriba, el *sistema difuso más simple* definido en cada intervalo de interpolación genérico  $C_{mn}$ , posee dos etiquetas por entrada, que coinciden con la parte de las etiquetas del sistema global marcada en trazo grueso en Fig.3.4(a), las cuales están asociadas a sendas funciones de pertenencia, que a su vez resultan ser una la complementaria de la otra. Por tanto, el circuito que ha de realizar las funciones de pertenencia en el *Núcleo Analógico* debe ser capaz de generar dos tramos con pendientes de signo contrario, como muestra claramente la Fig.3.4(b). Esto se consigue de la forma más simple mediante un par diferencial, como muestra la Fig.3.7(a). Además, el par diferencial proporciona dos curvas complementarias, lo cual se puede explotar para ahorrar la implementación explícita de un circuito que realice el operador complemento, si se implementa el operador mínimo para realizar la T-norma en base a circuitería de máximo-más-complemento aprovechando la ley de De Morgan [Wang97][Vida95a], como ya se hace en la arquitectura de la Fig.3.3 que sirve de referencia.

Desde el punto de vista de la programación, el par diferencial permite situar la función de pertenencia en distintos puntos del espacio de entrada sin más que cambiar la tensión asociada a la puerta de uno de los transistores del par,  $E_p$  en la Fig.3.7, y que es proporcionada por la circuitería de multiplexado, concretamente por el bloque *Programador de Antecedente de Reglas* como se muestra en el esquema de la Fig.3.1. Según este esquema cada una de las líneas  $\{E_1, \dots, E_k, \dots, E_M\}$  se conecta a una entrada  $E_p$  perteneciente al par diferencial correspondiente al *Bloque de Entrada* asociado a la entrada  $x_i$  del controlador.

La Fig.3.7(b) muestra el esquema de los circuitos que constituyen la implementación final del *Bloque de Entrada*. En él se observa que las entradas, tanto la del controlador  $x_i$ , como la de programación  $E_p$ , se proporcionan en forma de tensiones, mientras que el procesamiento posterior se hace en modo de corriente. Las salidas de intensidad del par diferencial  $\bar{s}_{ia}$  y  $\bar{s}_{ib}$  se replican para generar las  $2^M$  salidas necesarias para implementar los antecedentes de las  $2^M$  reglas que determinan la salida dentro de un determinado intervalo, lo que ahorra área y reduce el consumo de potencia. Estas intensidades son convertidas en las tensiones  $V_{Giak}$  y  $V_{Gibk}$ , en la *celda de entrada del circuito de mínimo*, que aparece sombreada en Fig.3.7(b). Las  $2^M$  salidas de tensión así generadas se cablean con otras proporcionadas por los restantes *Bloques de Entrada*, de forma que se obtienen los  $2^M$  antecedentes de regla que conforman el *Núcleo Analógico*. La Fig.3.8 ilustra cómo se puede obtener la salida de un antecedente de regla, concretamente el de la regla  $R_1$  del *Núcleo Analógico* (ver Fig.3.4(b)) en un caso bidimensional, a partir del cableado de las salidas de las *celdas de entrada del circuito de mínimo* en el *Bloque de Entrada* (Fig.3.7(b)). En este ejemplo, cuya partición del espacio de entrada se ilustra en la Fig.3.4(a), y para el que el *Núcleo Analógico* se organiza según el esquema de la Fig.3.5, se necesitan dos *Bloques de Entrada*, uno por entrada del sistema, y cada uno de ellos proporciona cuatro tensiones de salida  $V_{Gia1}$ ,  $V_{Gia2}$ ,  $V_{Gib1}$  y  $V_{Gib2}$  según la notación de la Fig.3.7(b).

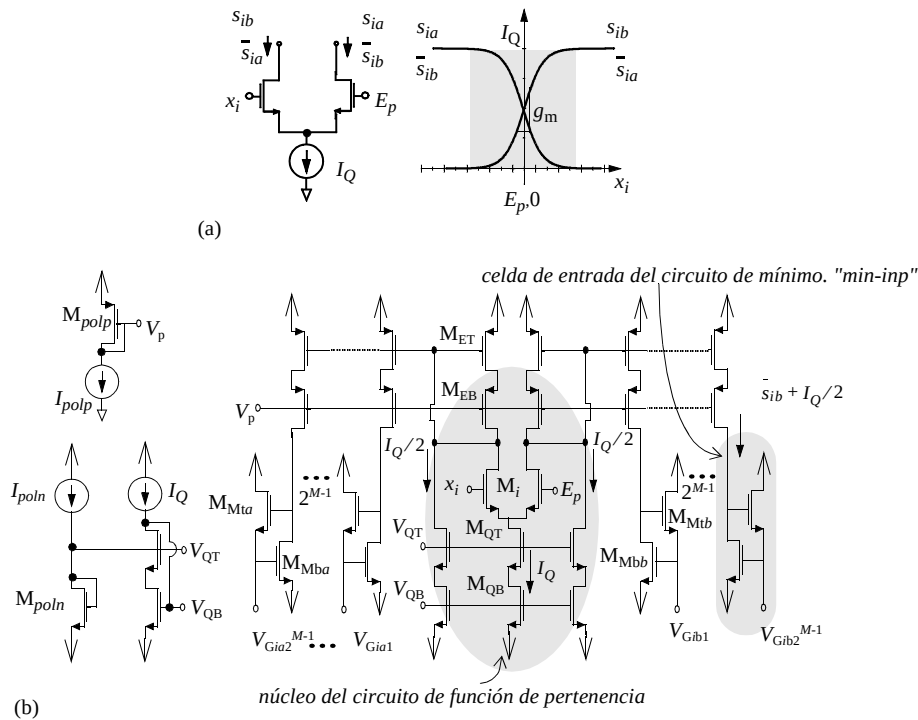


Fig.3.7 Par diferencial como generador de la función de pertenencia (a) y bloque de entrada (b).

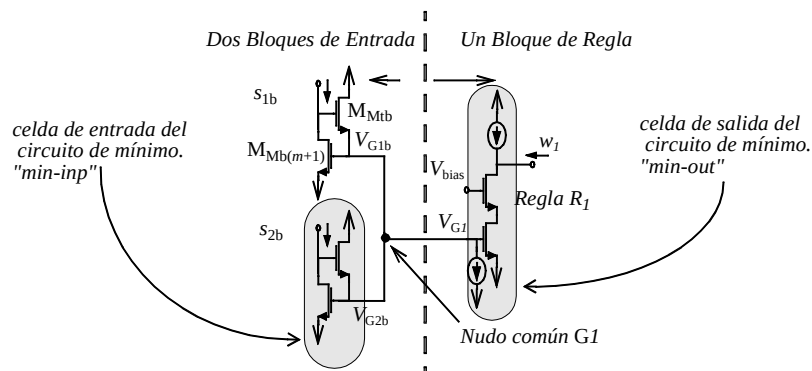


**Fig.3.8 Interfaz entre los bloques de entrada y regla para la obtención de la regla  $R_1$ , en el ejemplo bidimensional de la Fig.3.4.**

Para construir la regla  $R_1$ , cuyo antecedente queda enunciado en términos lingüísticos según la expresión

$$\text{SI } x_1 \text{ ES } X_{1a} \text{ Y } x_2 \text{ ES } X_{2a} \text{ ENTONCES ...},$$

las salidas  $V_{G1b1}$  y  $V_{G2b1}$ , que corresponden a las tensiones obtenidas a partir de las funciones de pertenencia complementarias a las asociadas a las etiquetas  $X_{1a}$  y  $X_{2a}$  en la Fig.3.4(b), se conectan en el nudo común  $G_1$  a la celda de salida del circuito de mínimo, (a la derecha en la Fig.3.8) que completa el procesamiento necesario para generar el antecedente de la regla  $R_1$ , según la expresión (1.5) del Capítulo 1. Como veremos en la próxima sección, este celda de salida constituye a su vez la etapa de entrada del *Bloque de Regla*. La Fig.3.8 es de hecho el circuito de mínimo propuesto en [Vida96b], donde los complementos a la entrada se ahorran porque se proporcionan de forma directa. Recuerdese que  $s_{ia}(x_i) = \bar{s}_{ib}(x_i)$ , por lo que  $s_{1b}$  y  $s_{2b}$  son los complementos de  $s_{1a}$  y  $s_{2a}$  respectivamente en el intervalo de interpolación  $C_{mn}$  según la discusión a propósito de la Fig.3.4(b).

### 2.3.1.1. Consideraciones de diseño del Bloque de Entrada

El par diferencial y los espejos de corriente empleados como bloques funcionales en Fig.3.7 son bien conocidos en diseño analógico, y el circuito de mínimo empleado se estudia en profundidad en [Vida96b]. Por tanto, vamos a obviar una descripción detallada de los mismos al objeto de centrarnos en los aspectos más específicos de la implementación propuesta.

#### Rango de entrada y forma de la función de pertenencia

Un primer parámetro de diseño importante a nivel de sistema es el rango de entrada del controlador, que define los límites del espacio de entrada del sistema difuso global. En esta

propuesta, su valor está determinado por el rango de entrada en modo común del par diferencial que constituye la etapa de entrada del *Bloque de Entrada*, Fig.3.7. De esta forma este circuito constituye a su vez el interfaz analógico externo de entrada del controlador. Para obtener las expresiones que determinan este rango de entrada, suponemos que la entrada diferencial del par en Fig.3.7(b) es nula, y que aumentamos progresivamente la entrada en modo común, hasta que los transistores del espejo de corriente superior salen de la región de saturación. La tensión en este punto nos da el límite superior del rango, mientras que el mismo procedimiento en sentido descendente nos da el límite inferior. Utilizando modelos de primer nivel del transistor MOS, estos cálculos dan el rango de entrada siguiente,

$$V_{QT} + \sqrt{\frac{I_Q}{2\beta_{M_i}}} \leq x_i \leq V_p + V_{T0n} + V_{T0p} \quad (3.2)$$

Donde  $V_p$  y  $V_{QT}$  deben ser tomadas a su vez para mantener a los transistores  $M_{ET}$  y  $M_{EB}$  del espejo superior, y  $M_{QT}$  y  $M_{QB}$  del espejo inferior en saturación, lo que implica

$$V_p \leq V_{DD} - (V_{T0p} + \sqrt{I_Q/\beta_{M_{ET}}} + \sqrt{I_Q/\beta_{M_{EB}}}) , y \quad (3.3)$$

$$V_{QT} \geq V_{T0n} + \sqrt{I_Q/\beta_{M_{QB}}} + \sqrt{I_Q/\beta_{M_{QT}}} . \quad (3.4)$$

Estas tensiones de polarización se generan internamente a partir de intensidades de polarización derivadas de una misma externa, y deben ser tales que

$$I_{polp}/\beta_{polp} = (V_{DD} - V_p - V_{T0p})^2 , y \quad (3.5)$$

$$I_{poln}/\beta_{poln} = (V_{QT} - V_{T0n})^2 . \quad (3.6)$$

Por otra parte, otros parámetros de interés para el diseñador de sistemas son la pendiente en los puntos de cruce  $x_i = E_p$ ,

$$g_m = \sqrt{\frac{I_Q \beta_{M_i}}{2}} \quad (3.7)$$

y el soporte de la función de pertenencia implementada <sup>2†</sup>. En este punto es conveniente hacer una discusión acerca de las consideraciones de diseño que influyen en la definición de este soporte.

En primer lugar, es bien conocido que los sistemas difusos se construyen con funciones de pertenencia de diversas formas, no necesariamente lineales. En el apartado 5.2. del Capítulo1 se establece una cota de aproximación que no presupone ninguna condición sobre la linealidad de las funciones de pertenencia, sólo se exige para éstas una forma

2†. El soporte de un conjunto difuso  $F$  es el conjunto nítido de elementos que tienen grado de pertenencia estrictamente mayor que 0,  $\text{Soporte}(F) = \{x \in X \mid \mu_F(x) > 0\}$  , donde  $X$  es un universo de discurso o dominio de  $x$ .

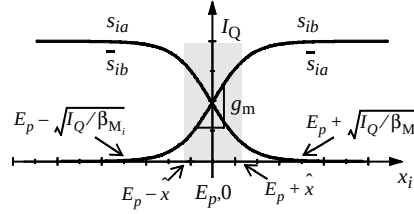


**Fig.3.9 Soporte de la función de pertenencia implementada.**

pseudotrapezoidal. Sin embargo, formas muy no lineales llevan a superficies de control poco suaves, cuando se sigue el procedimiento de construcción del sistema difuso descrito en esa misma sección, y sólo con un número mayor de reglas y con menor transparencia se podrían conseguir resultados similares a los conseguidos con funciones de pertenencia lineales. Es pues conveniente partir de funciones de pertenencia lo más suaves posible.

Por otra parte, como veremos en la sección siguiente, el *Núcleo Analógico* utiliza un circuito de normalización que implementa una transformación no lineal sobre la salida de la circuitería asociada al antecedente de regla,  $w_i$  en Fig.3.8, y  $w_k$  en general, para una regla genérica. El efecto conjunto produce una función de pertenencia normalizada que puede ser excesivamente no lineal. Por tanto, al objeto de conseguir una interpolación más suave, podemos diseñar el *Bloque de Entrada*, y en general el controlador, de manera que únicamente una porción de la curva de intensidad de salida del par diferencial en torno a  $x_i = E_p$  se utilice para generar  $w_k$ , evitando las regiones próximas a los puntos de saturación de la curva  $x_i = E_p \pm \sqrt{I_Q/\beta_{M_i}}$  donde la no linealidad es más acentuada, Fig.3.9.

La función de pertenencia así generada viene caracterizada por la ecuación,

$$s_{ib}(x_i) = I_Q/2 + \sqrt{I_Q\beta_{M_i}/2}(x_i - E_p)\sqrt{1 - (x_i - E_p)^2\beta_{M_i}/2I_Q} \quad , \quad |x_i - E_p| \leq \hat{x} \quad (3.8)$$

donde  $\hat{x} \leq \sqrt{I_Q/\beta_{M_i}}$  se elige para obtener la suavidad deseada, que es tanto mayor cuanto menor sea el valor de  $x$ .

A modo de ilustración de este comportamiento, en la Fig.3.10 se recoge el resultado de dos simulaciones HSPICE en la que se muestra la respuesta transitoria del controlador cuando una sus entradas es excitada por una señal en rampa que recorre por completo su universo de discurso, mientras la otra permanece a un valor constante, y éste se ha configurado con valores *singleton* que alternan sucesivamente valores máximos y mínimos. En estas circunstancias, la señal de salida que recoge la Fig.3.10(a) corresponde al caso en el que el *Núcleo Analógico* se diseña permitiendo la saturación del par diferencial que genera las funciones de pertenencia, mientras que en la señal de salida de la Fig.3.10(b), corresponde al caso en el que el *Núcleo Analógico* se diseña evitando las regiones próximas a los puntos de saturación de la curva del par diferencial. De la comparación de ambas curvas resulta clara la mayor suavidad que se

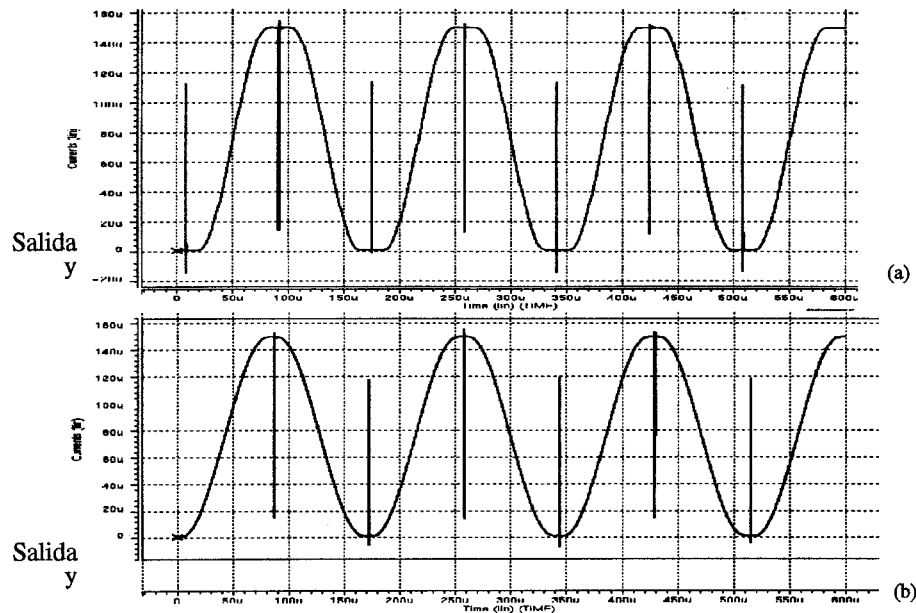


**Fig.3.10 Simulación HSPICE que ilustra la respuesta del controlador cuando se diseña el Núcleo Analógico sin evitar la zona de saturación del par diferencial que realiza las funciones de pertenencia (a); y evitando la zona de saturación (b).**

obtiene en el segundo caso. Las líneas verticales que cruzan las curvas en ambas gráficas corresponden al transitorio de reprogramación del *Núcleo Analógico* como consecuencia de un cambio de intervalo de interpolación.

Por su parte, la Fig.3.11 ilustra la suavidad de la interpolación conseguida cuando se intenta aproximar el plano de Fig.3.11(a) en un intervalo de interpolación. Una implementación

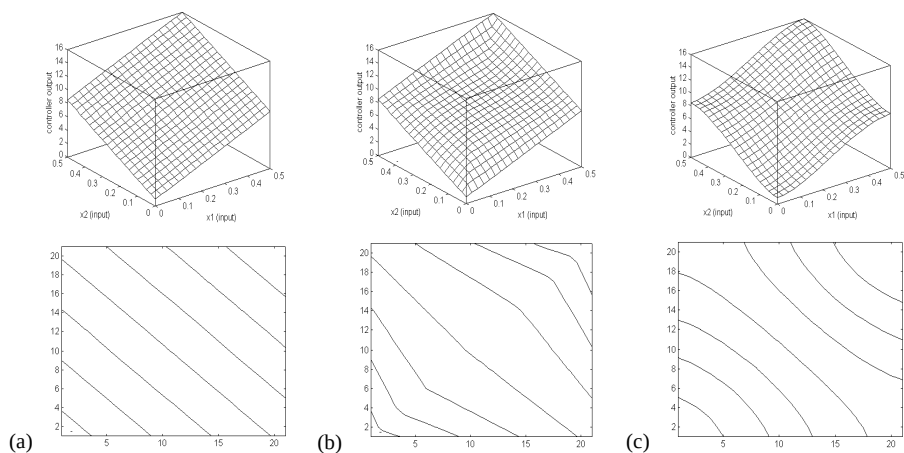


**Fig.3.11 Interpolación de un plano (a) mediante el algoritmo difuso con funciones de pertenencia triangulares y centro de área con división (b) y el núcleo analógico (c).**



del algoritmo difuso con funciones de pertenencia triangulares, t-norma mediante el operador mínimo, y *defuzzificación* mediante centro de área (con división explícita) da como resultado la superficie de interpolación en Fig.3.11(b), mientras que el *Núcleo Analógico* modelado con ecuaciones de primer nivel para los transistores MOS proporciona la superficie de interpolación en Fig.3.11(c). Se puede observar que la suavidad de ambas superficies es comparable, lo que se confirma con los diagramas de contorno en la parte inferior de cada una de las figuras.

#### *Respuesta dinámica y consumo*

La respuesta dinámica del circuito viene determinada por las respuestas de los elementos o bloques básicos que lo componen, en concreto el par diferencial, los espejos de corriente y el circuito de mínimo. El par diferencial y el espejo de corriente son bien conocidos en diseño analógico, mientras que el circuito de mínimo se analiza cuidadosamente en [Vida96b]. La conclusión sobre el comportamiento dinámico es general, y se resume en un mejor comportamiento para grandes intensidades, grandes razones  $W/L$  de los transistores, y pequeños condensadores parásitos, es decir pequeñas áreas de los transistores, o productos  $WL$ .

Estas directrices se tienen en cuenta en el diseño del *Bloque de Entrada* de la Fig.3.7, donde la estrategia de diseño a nivel de circuito para mejorar la respuesta dinámica se traduce en añadir las dos ramas laterales a ambos lados del par diferencial en la Fig.3.7(b), y que suman una intensidad de desplazamiento de valor  $I_Q/2$  a ambas salidas del par, y por tanto a la entrada de la *celda del circuito de mínimo*. Esta intensidad evita que los transistores  $M_{ET}$  y  $M_{EB}$  del espejo de arriba y del circuito de mínimo entren en la región de inversión débil, donde la respuesta dinámica es muy pobre.

Esta aproximación lleva a compromisos de diseño con el rango de entrada (obsérvese cómo disminuye el rango si aumentamos  $I_Q$  en (3.2)), y con el consumo, que para el bloque de entrada en Fig.3.7 es,

$$P_{bloque\_entrada} = 2I_Q(2^{2M-1} + 1)V_{DD} \quad (3.9)$$

y por tanto es directamente proporcional al valor de  $I_Q$ .

Por otra parte, el aumento de la razón  $W/L$  para mejorar la transconductancia de los transistores, y así la respuesta dinámica, también perjudica al rango de entrada; y la disminución del área, como veremos más adelante, aumenta el desapareamiento entre los transistores, y por tanto el error, por lo que todos estos factores también han de ser tenidos en cuenta en el diseño.

### Errores

Los errores de origen sistemático en los circuitos de la Fig.3.7 se han reducido en primer lugar buscando una topología simétrica, que implica cargas similares en ambos nudos de salida del par diferencial. Esta simetría minimiza la componente de error sistemático en el *offset* a la salida del par diferencial, porque las tensiones en los nudos de salida del bloque son iguales si atendemos únicamente a fuentes de error de origen sistemático.

Por otra parte, los espejos de corriente se han escogido en configuraciones cascode, y entre éstas, una configuración que no degrada en exceso el rango de entrada. La configuración escogida iguala las tensiones puerta-fuente y drenador-fuente en los transistores no cascode a la entrada y salida de los espejos, con lo que el error sistemático es muy pequeño.

El error más trascendente a nivel de sistema es aquel que produce el cambio de la tensión de fuente común de los transistores del par diferencial, y que cambia ligeramente la intensidad de polarización debido a la resistencia finita de la fuente de intensidad implementada. Sin embargo, debido a la implementación cascode esta resistencia es muy alta, y el error introducido es de unos pocos nanoamperios, despreciable frente a errores debidos al desapareamiento.

La fuente de error más importante está en el desapareamiento de los transistores debido a las no idealidades en el proceso de fabricación. Este desapareamiento local, dentro de un mismo circuito, da lugar a que parámetros como el factor de ganancia beta o la tensión umbral difieran de un transistor a otro, aún siendo teóricamente iguales. En el diseño del *Bloque de Entrada* en el controlador ejemplo los transistores se han tomado de tamaño no mínimo para reducir el desapareamiento. Una elección más depurada no se ha estimado necesaria por razones que se explican en el apartado 2.3.3. dedicado a consideraciones globales de diseño.

En la Tabla 3.1 se resumen los tamaños de los transistores y los valores de las fuentes de polarización de los circuitos del *Bloque de Entrada* según la notación de la Fig.3.7 empleados en el controlador ejemplo que ha sido fabricado.

**Tabla 3.1:** Tamaños de los transistores e intensidades en el *Bloque de Entrada*.

| $M_i (W/L)$                      | $M_{ET} (W/L)$                | $M_{EB} (W/L)$                  | $M_{QT} (W/L)$                  | $M_{QB} (W/L)$                | $M_{Mt} (W/L)$                 |
|----------------------------------|-------------------------------|---------------------------------|---------------------------------|-------------------------------|--------------------------------|
| 21.6 $\mu\text{m}/12\mu\text{m}$ | 24 $\mu\text{m}/4\mu\text{m}$ | 24 $\mu\text{m}/0.8\mu\text{m}$ | 12 $\mu\text{m}/0.8\mu\text{m}$ | 12 $\mu\text{m}/6\mu\text{m}$ | 9 $\mu\text{m}/4.2\mu\text{m}$ |
| $M_{Mb} (W/L)$                   | $M_{poln} (W/L)$              | $M_{polp} (W/L)$                | $I_{polp}$                      | $I_{poln}$                    | $I_Q$                          |
| 9 $\mu\text{m}/4.2\mu\text{m}$   | 5 $\mu\text{m}/20\mu\text{m}$ | 5 $\mu\text{m}/18\mu\text{m}$   | 5 $\mu\text{A}$                 | 5 $\mu\text{A}$               | 20 $\mu\text{A}$               |

Por su parte, en la Fig.3.12 se muestra un detalle del *layout* correspondiente a uno de estos *Bloques de Entrada* (núcleo del circuito de función de pertenencia, espejos de réplica y celdas de entrada del circuito de mínimo) en la que se aprecia la disposición en centroide común de

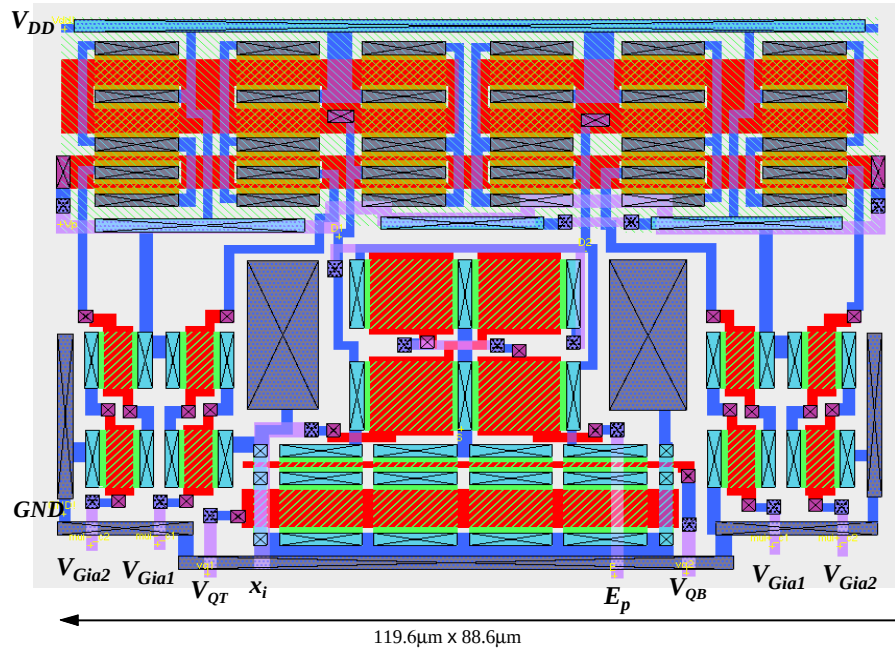


Fig.3.12 Detalle del layout de uno de los dos Bloques de Entrada empleado en el diseño del controlador ejemplo.

los transistores que forman el par diferencial, así como la simetría y regularidad en la disposición de los diferentes elementos, a fin de minimizar los efectos debidos al desapareamiento entre transistores y ahorrar área.

### 2.3.2. Diseño del Bloque de Regla

Como ya se ha mencionado, en la realización del *Núcleo Analógico* se emplea un *Bloque de Regla* por regla activa en el sistema. La Fig.3.13 muestra un esquema de las diferentes etapas que conforman cada uno de esos bloques. La etapa de entrada está formada por la *celda de*

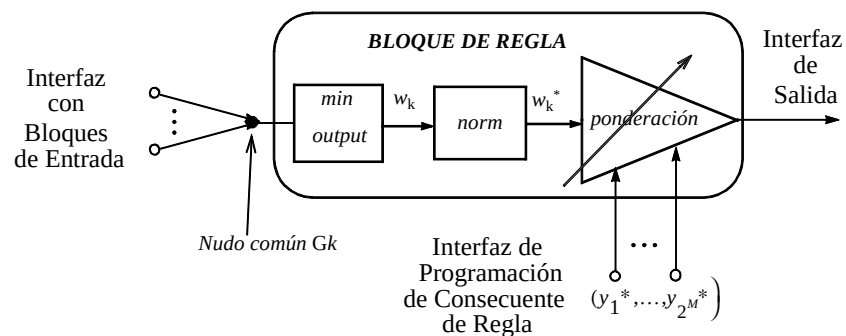


Fig.3.13 Bloque de Regla del Núcleo Analógico: Etapas constitutivas

salida del circuito de mínimo, "min-output", cuya salida proporciona el resultado de la evaluación del antecedente de regla  $w_k$ . A esta etapa le sigue una *etapa de normalización* que permite obtener el antecedente de regla normalizado  $w_k^*$ . Finalmente la *etapa de ponderación* con los valores *singleton*  $(y_1^*, \dots, y_{2^M}^*)$ , proporciona la salida del *Bloque de Regla*. La Fig.3.14 muestra el esquema de los circuitos que constituyen la implementación final del *Bloque de Regla*, que resultan ser muy similares a los empleados en el bloque de regla utilizado en la implementación descrita en [Vida96b], aunque hay diferencias que serán aquí destacadas.

El circuito que realiza la etapa de entrada del bloque en Fig.3.14(a) se corresponde con la celda de salida del circuito de mínimo en Fig.3.8. Su función es transformar la tensión de

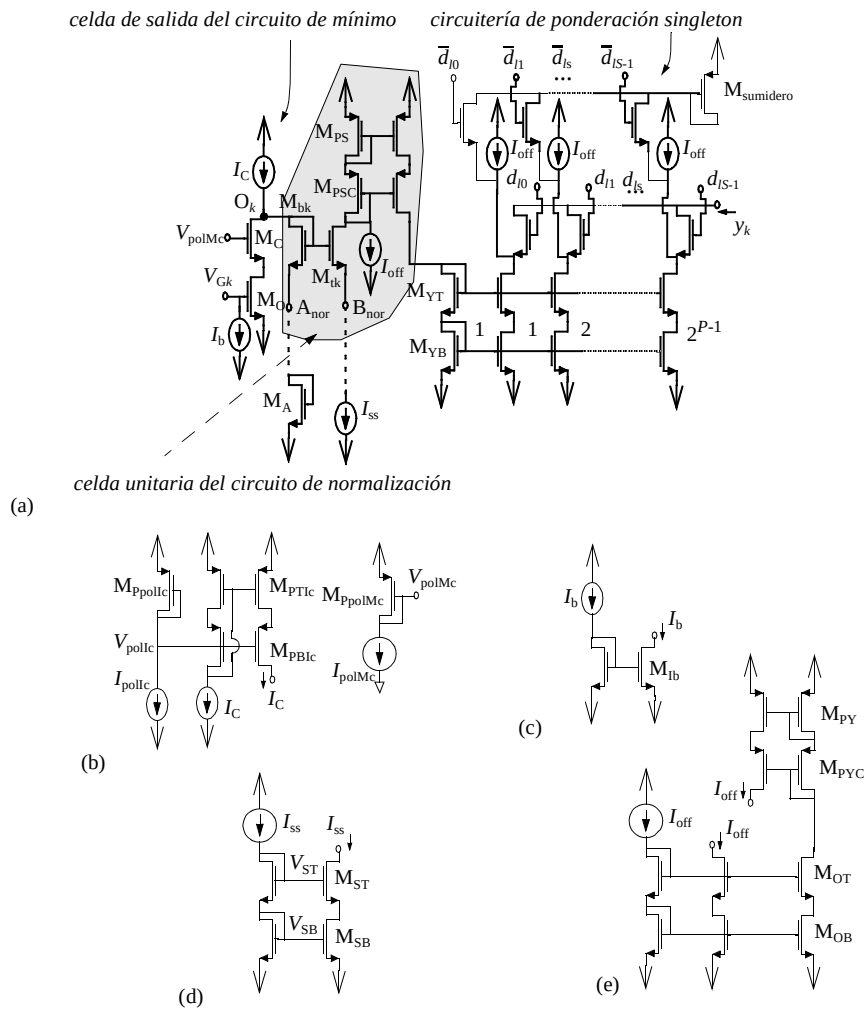


Fig.3.14 Bloque de Regla (a); generación polarizaciones: fuente  $I_C$  (b); fuente  $I_b$  (c); fuente  $I_{ss}$  (d) y fuente  $I_{off}$  (e).

salida proporcionada por las *celdas de entrada del circuito de mínimo*, ubicadas en el *Bloque de Entrada*, en una intensidad, y realizar la operación de complemento a la salida, necesaria debido a que el operador mínimo se realiza explotando la ley de De Morgan, es decir complementando a entrada y salida de un circuito de máximo. La fuente  $I_C$  es utilizada para realizar este complemento, así como para añadir una cierta intensidad de desplazamiento a la entrada del circuito de normalización, que es la etapa que sigue, siendo el resultado,

$$\begin{aligned} w_k &= I_C - I_Q/2 - \max\{\overline{s_{1k_j}(x_1)}, \overline{s_{2k_j}(x_2)}, \dots, \overline{s_{Mk_j}(x_M)}\} = \\ &= I_{OS} + \min\{s_{1k_j}(x_1), s_{2k_j}(x_2), \dots, s_{Mk_j}(x_M)\} \end{aligned} \quad (3.10)$$

donde  $k_j = a, b$  y donde  $s_{ik_j}(x_i)$  viene dado por (3.8). La fuente de intensidad  $I_b$  es necesaria para proporcionar un camino de descarga del nudo común  $Gk$  en Fig.3.8.

La siguiente etapa del bloque en Fig.3.14(a) es la *celda de entrada al circuito de normalización*, el cual aparece completo en Fig.3.15(a). Este bloque garantiza que la suma de todas las intensidades de salida del circuito es constante e igual a  $I_{ss}$ . La transformación que realiza sobre una entrada  $w_k$  es,

$$w_k^* = \frac{\beta_{N4t}}{\beta_{N4b}} w_k \left[ 1 + \frac{\eta(w)}{\sqrt{w_k}} \right]^2 \quad (3.11)$$

donde  $\eta(w)$  es,

$$\begin{aligned} \eta(w) &= \frac{\sum_k \sqrt{w_k}}{N} \left( \sqrt{1 + \frac{NI_{SS}' - \sum_k w_k}{\sum_k \sqrt{w_k}^2} - 1} \right) \\ &\quad \left( I_{SS}' = \frac{\beta_{Nb}}{\beta_{Nr}} I_{SS} \right) \end{aligned} \quad (3.12)$$

La modificación de la estrategia de diseño del circuito con respecto a la reportada en [Vida96b] radica en la geometría de los transistores, que se escoge para obtener una característica de transferencia en (3.11) coherente con la estrategia de multiplexado. En la implementación previa, con objeto de mejorar la respuesta dinámica de las etapas subsiguientes de la manera más sencilla posible, se permitía que todas las salidas del circuito de normalización tuvieran un valor no nulo, es decir había un desplazamiento de intensidad sumado a todas ellas. Por ello, era necesario retirar este desplazamiento de corriente, lo que se hacía en el nudo de salida mediante una fuente de intensidad. Sin embargo, esta intensidad a restar depende de los valores *singleton* asociados a las reglas implementadas, y como el *Núcleo Analógico* se multiplexa, el

conjunto de valores *singleton*  $y_1^* \dots y_l^* \dots y_{2^M}^*$  cambia de un intervalo de interpolación al siguiente, por tanto la intensidad de desplazamiento a retirar cambia de un intervalo de interpolación a otro, es decir depende de la entrada. Podría resolverse esta diferencia mediante una fuente de intensidad programable que retirase una intensidad a la salida dependiente del intervalo de interpolación. Sin embargo, una solución más simple se obtiene diseñando de manera que cuando una salida del circuito de normalización es máxima, las restantes sean nulas, por lo que no queda ningún desplazamiento residual. Esto se consigue diseñando el circuito de normalización de manera que salga del límite marcado por el rango diferencial asociado a una entrada. Si una de las entradas en Fig.3.15(a) crece mientras que las otras permanecen a un valor constante, el transistor de salida de la intensidad que cambia conducirá eventualmente toda la intensidad  $I_{SS}$ , y los otros transistores de salida permanecerán en corte. El rango diferencial de entrada definido por este hecho es,

$$0 \leq w_k \leq \beta_{Nb} \left( \sqrt{\frac{I_{OS}}{\beta_{Nb}}} + \sqrt{\frac{I_{SS}}{\beta_{Nt}}} \right)^2 \quad (3.13)$$

donde se ha explotado que los conjuntos difusos sobre los que se definen la reglas difusas en el sistema global son *consistentes*, (ver Capítulo 1, apartado 5.2.), de forma que cuando una regla tiene salida máxima el resto se anula, e  $I_{OS}$  es una intensidad de desplazamiento a la entrada que viene definida por (3.10).

La salida del circuito de normalización se refleja en un espejo de corriente y entra en el circuito de ponderación *singleton* de Fig.3.15(b), que es un espejo de corriente programable digitalmente por la palabra  $d_{l0} \dots d_{lS-1}$ , Fig.3.14(a), donde el índice  $l$ ,  $1 \leq l \leq 2^M$ , está asociado al índice de los valores *singleton*  $y_1^* \dots y_l^* \dots y_{2^M}^*$ , y  $S$  es su número de bits. Este circuito es empleado también en [Vida96b], aunque en el *Núcleo Analógico* multiplexado se ha modificado ligeramente para mejorar el rendimiento. En este caso la necesidad o conveniencia de modificar el circuito viene de observar que la programación dinámica del espejo de corriente producía grandes picos de corriente a la salida del controlador (ver Fig.3.16). La causa principal de estos picos es la corriente demandada por las ramas del espejo que no contribuyen

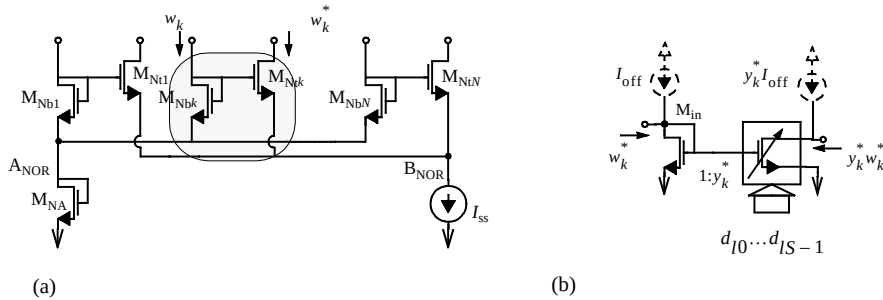


Fig.3.15 Circuito de normalización (a), espejo de ponderación por el valor *singleton* (b).

a la corriente de salida, y cuyos transistores trabajan en la región óhmica, cuando se seleccionan de nuevo para proporcionar corriente a la salida, debido a la carga necesaria para hacer la transición de óhmica a saturación. La solución propuesta en este trabajo e implementada, mantiene a todos los transistores en saturación proporcionando un camino alternativo de corriente a través de llaves de corriente controladas por las señales de control complementarias  $\bar{d}_{l0} \dots \bar{d}_{lS-1}$  que aparecen en los circuitos de la Fig.3.14(a). De esta manera, cuando una rama genérica deja de proporcionar corriente a la salida, su llave de corriente asociada controlada por  $d_{lS}$  se cierra, pero la llave controlada por  $\bar{d}_{lS}$  se abre y deja pasar corriente procedente del transistor  $M_{\text{sumidero}}$ , por lo que el transistor no sale de saturación. El precio a pagar por esta mejora en el comportamiento dinámico, como en otros muchos casos, es un aumento del consumo de potencia. La Fig.3.16 ilustra esta circunstancia del circuito de ponderación mediante el resultado de una simulación HSPICE, en la que se muestra la respuesta de un espejo de ponderación, sin camino de corriente alternativo (curva inferior) y con camino de corriente alternativo (curva central), a un transitorio de las líneas de control de programación  $y_1, y_2, y_3, y_4$  (curva superior). En ella se aprecia cómo en el primer caso, como consecuencia de la activación de todas las líneas de control, el espejo de corriente demanda inicialmente una corriente con un valor de pico de 1.2mA, antes de alcanzar el valor final programado de 150μA. Esta circunstancia desaparece en el segundo caso, en el que se produce una transición más rápida y suave.

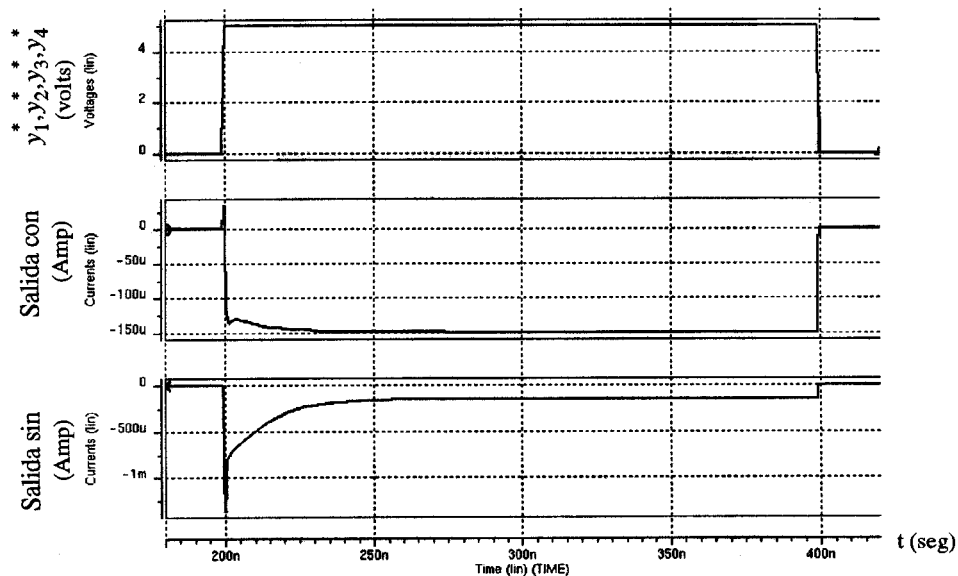


Fig.3.16 Simulación HSPICE del comportamiento del circuito de ponderación con camino de corriente alternativo (centro) y sin camino de corriente alternativo (abajo).

### 2.3.2.1. Consideraciones de diseño del Bloque de Regla

Como ya se ha indicado más arriba, el *Bloque de Regla* mostrado en la Fig.3.14 consta de tres etapas básicas: *celda de salida del circuito de mínimo*, *celda de normalización* y *espejo programable* para la ponderación por el valor *singleton*. Estudios detallados de estas etapas, salvo por las modificaciones introducidas, se encuentran en [Vida96b], aquí se recordarán los aspectos más relevantes que influyen en las decisiones de diseño.

#### *Celda salida del circuito de mínimo*

La *celda de salida del circuito de mínimo* tiene tres elementos que deben añadirse al transistor de salida del circuito de máximo multientrada. Estos tres elementos son la fuente de intensidad  $I_b$ , la fuente de intensidad  $I_C$  y el transistor cascode  $M_C$ . La fuente de intensidad  $I_b$  sirve para implementar un camino de descarga del nudo común en Fig.3.8. La magnitud de la intensidad  $I_b$  influye de dos formas, por una parte su valor ha de ser pequeño, porque la cota de error de peor caso es,

$$\varepsilon = \lambda \sqrt{\frac{I_b}{\beta_{MMt}}} \left(1 - \frac{1}{\sqrt{M}}\right) \quad (3.14)$$

donde  $M$  es el número de *Bloques de Entrada* conectados a un *Bloque de Regla* tal y como ilustra Fig.3.8, y que coincide con el número de entradas del controlador. Por otra parte, la discriminación teórica del circuito, es decir la diferencia mínima entre sus intensidades de entrada que es capaz de discernir es,

$$\Delta I = \lambda \beta_{MMb} (V_{Gk} - V_{T0n})^2 \sqrt{\frac{I_b}{\beta_{MMt}}} \quad (3.15)$$

Así pues, cuanto menor sea  $I_b$ , menor es  $\varepsilon$  en (3.14), y  $\Delta I$  en (3.22). El precio a pagar es un peor comportamiento dinámico, en el sentido de que la intensidad de entrada seleccionada se propaga a la salida más lentamente que si  $I_b$  fuera mayor. Sin embargo, el comportamiento dinámico del circuito de mínimo viene condicionado más fuertemente por la operación de conmutación, es decir cuando la entrada cambia y otra de las intensidades de entrada pasa a ser la mínima, con lo que otro de los bloques de entrada en Fig.3.8 determina la intensidad en la etapa de entrada del *Bloque de Regla*. La mencionada operación de conmutación tiene como compromisos de diseño los clásicos en diseño analógico, y su comportamiento mejora con menores tamaños y mayores razones  $W/L$  para los transistores, y mayores valores de las intensidades. Para este diseño se ha tomado  $I_b = 0.5\mu A$ , lo que coloca el error sistemático muy por debajo del 1% del fondo de escala, y la respuesta dinámica dentro de los objetivos de diseño. Nótese que el valor de  $I_b$  no es crítico, mientras sea pequeño, por lo que un espejo de corriente simple se utiliza para generar esta fuente, como se muestra en la Fig.3.14(c), (en cualquier caso, al ser  $G_k$  un nudo de baja impedancia, sus variaciones serán mínimas).



El segundo elemento en la etapa de entrada es la fuente de intensidad  $I_C$ . Su implementación ha de ser cuidadosa porque su error influye directamente en la salida de esta etapa, es decir en la salida del antecedente de regla. Como muestra la Fig.3.14(b), se implementa con un espejo cascode, que debe ser de alto rango de salida si las excursiones de tensión del nudo de salida  $O_k$  son suficientemente grandes. Esta es la estrategia en Fig.3.14(b), donde se debe cumplir,

$$\begin{aligned} V_{polMc} - V_{TMc} &\leq V_{TMbk} + \sqrt{\frac{I_{OS}}{\beta_{MC}}} + V_{T0n} + \sqrt{\frac{3I_{OS} + w_k}{\beta_{MA}}} \leq \\ &\leq V_{TMbk} + \sqrt{\frac{I_{OS}}{\beta_{MC}}} + V_{T0n} + \sqrt{\frac{3I_{OS} + w_k}{\beta_{MA}}} \leq V_{polIc} + V_{TPBIc} \end{aligned} \quad (3.16)$$

que resulta de imponer que las excursiones del nudo de entrada en el circuito de normalización  $O_k$  están dentro del rango de salida del circuito de mínimo. Por otra parte, al ser  $I_C$  una fuente estática, los transistores pueden ser grandes sin penalizar la respuesta dinámica, con lo que se mejora el apareamiento de los transistores.

Finalmente, el transistor cascode  $M_C$  y la tensión  $V_{polMc}$  se toman para mejorar el apareamiento entre los nudos de drenador del transistor  $M_O$  y los transistores  $M_{Mb(a,b)}$  del *Bloque de Entrada* que conducen la intensidad propagada a la salida de forma que se cumpla,

$$V_{polMc} = V_{T0n} + V_{TM_{Mb(a,b)}} + V_{TM_C} + \sqrt{\frac{I_Q}{\beta_{M_C}}} + \sqrt{\frac{I_Q}{\beta_{M_{Mb(a,b)}}}} + \sqrt{\frac{I_b}{\beta_{M_{Mb(a,b)}}}} \quad (3.17)$$

que se puede obtener eligiendo  $I_{polMc}$  y  $\beta_{polMc}$  en el circuito de polarización de la Fig.3.14(b) de forma que,

$$V_{polMc} = V_{T0p} + \sqrt{\frac{I_{polMc}}{\beta_{polMc}}} \quad (3.18)$$

### Circuito de Normalización

La siguiente etapa en Fig.3.14(a) es la correspondiente al *circuito de normalización*. Ya se ha dicho arriba, y se volverá a incidir en ello, que este circuito se diseña para operar fuera de su rango en modo diferencial en (3.13). Asimismo, su rango en modo común, que viene impuesto por que los transistores que generan la intensidad de normalización  $I_{ss}$  y los que replican la salida (espejo p de salida formado por los transistores  $M_{PS}$  y  $M_{PSC}$ ) trabajen en zona óhmica, está dado por la expresión,

$$\left[ \frac{V_{Bnor} - V_{T0n} + \sqrt{\frac{I_{ss}}{N\beta_{tk}}}}{\sqrt{\frac{1}{\beta_{bk}}} + \sqrt{\frac{N}{\beta_A}}} \right]^2 \leq w_k \leq \left[ \frac{V_{DD} - 2V_{T0p} - V_{T0n} - \sqrt{\frac{I_{ss}}{N\beta_{PS}}} - \sqrt{\frac{I_{ss}}{N\beta_{PSC}}}}{\sqrt{\frac{1}{\beta_{bk}}} + \sqrt{\frac{N}{\beta_A}}} \right]^2 \quad (3.19)$$

donde  $V_{Bnor}$  es la tensión mínima que mantiene a los transistores del espejo que proporciona  $I_{ss}$  en saturación y  $N$  es el número de reglas activas.

Las fuentes principales de error de origen sistemático en la etapa de normalización están en la impedancia finita de la fuente  $I_{ss}$ , ver Fig.3.14(d), que hace que cambie esta intensidad si cambia la tensión del nudo Bnor, y los errores introducidos por el espejo de salida de la celda. Ambos errores se minimizan utilizando configuraciones cascode.

Por último, el comportamiento dinámico se ha mejorado introduciendo una fuente de polarización a la entrada del espejo de salida  $I_{off}$ , para evitar que los transistores de este espejo trabajen en inversión débil. Esta intensidad de desplazamiento añadida se retira más adelante, en la etapa siguiente. Su generación, tanto a la entrada como a la salida, se hace tal y como se muestra en la Fig.3.14(e).

#### Circuito de ponderación

Finalmente la etapa de ponderación, Fig.3.14(a), la constituye un espejo de salida programable digitalmente, como ya se ha dicho. Al objeto de mejorar el apareamiento, los transistores que lo forman se construyen utilizando celdas de igual tamaño, e iguales a los transistores de entrada. Por su parte, la configuración cascode disminuye los errores de origen sistemático. En cuanto al comportamiento dinámico, éste es mejorado permitiendo a la entrada una intensidad de desplazamiento  $I_{off}$  (la misma que fue sumada a la entrada del espejo p en la etapa de salida de la celda de normalización), a la salida de cada rama. Por lo demás, se implementa la estrategia ya descrita destinada a disminuir los picos de corriente a la salida cuando el espejo es reprogramado.

El *Bloque de Regla* tiene un consumo estático que depende del grado en que la regla se verifica para una determinada entrada. Su valor máximo, cuando la salida de la celda de normalización conduce toda la intensidad de normalización es,

$$P_{max\_bloque\_regla} = [I_b + I_C + 2(y^* + 2)I_{ss}]V_{DD} \quad (3.20)$$

mientras que su valor mínimo se da cuando la regla no contribuye a la salida, por tanto la salida de la celda de normalización es mínima y el consumo es,

$$P_{min\_bloque\_regla} = [I_b + I_C + (y^* + 2)I_{ss}]V_{DD} \quad (3.21)$$

En ambos casos se ha considerado el consumo debido a la corriente a la salida, que se ha supuesto suministrada por la fuente de alimentación  $V_{DD}$ .

En la Fig.3.17 se muestra un detalle del *layout* correspondiente a uno de los *Bloques de Regla*, (circuitos de la Fig.3.14(a) excepto llaves de corriente y polarizaciones) realizados en el controlador ejemplo que ha sido fabricado. En ella se aprecia la simetría y regularidad en la disposición de los diferentes elementos, a fin de minimizar los efectos debidos al desapareamiento entre transistores, y ahorrar área; mientras que en la Tabla 3.2 se resumen los tamaños de los transistores y los valores de las fuentes de polarización del *Bloque de Regla* según la notación de la Fig.3.14.

**Tabla 3.2:** Tamaños de los transistores e intensidades en el Bloque de Regla.

| $M_o(W/L)$                        | $M_C(W/L)$                     | $M_{bk}(W/L)$                   | $M_{ik}(W/L)$                   | $M_{pSC}(W/L)$                  | $M_{pS}(W/L)$                 |
|-----------------------------------|--------------------------------|---------------------------------|---------------------------------|---------------------------------|-------------------------------|
| 9 $\mu\text{m}/4.4\mu\text{m}$    | 9 $\mu\text{m}/0.8\mu\text{m}$ | 13 $\mu\text{m}/5\mu\text{m}$   | 5 $\mu\text{m}/13\mu\text{m}$   | 25 $\mu\text{m}/0.8\mu\text{m}$ | 25 $\mu\text{m}/4\mu\text{m}$ |
| $M_A(W/L)$                        | $M_{YB}(W/L)$                  | $M_{YT}(W/L)$                   | $I_{ss}$                        | $I_{off}$                       | $M_{Ppolic}(W/L)$             |
| 40 $\mu\text{m}/10\mu\text{m}$    | 10 $\mu\text{m}/4\mu\text{m}$  | 10 $\mu\text{m}/0.8\mu\text{m}$ | 10 $\mu\text{A}$                | 10 $\mu\text{A}$                | 5 $\mu\text{m}/18\mu\text{m}$ |
| $I_{polic}$                       | $M_{PTI}(W/L)$                 | $M_{PBic}(W/L)$                 | $I_C$                           | $M_{PpolMc}(W/L)$               | $I_{polMc}$                   |
| 5 $\mu\text{A}$                   | 16 $\mu\text{m}/10\mu\text{m}$ | 16 $\mu\text{m}/0.8\mu\text{m}$ | 30 $\mu\text{A}$                | 10 $\mu\text{m}/8\mu\text{m}$   | 5 $\mu\text{A}$               |
| $M_{ib}(W/L)$                     | $I_b(W/L)$                     | $M_{pY}(W/L)$                   | $M_{pYC}(W/L)$                  | $M_{ST}(W/L)$                   | $M_{SB}(W/L)$                 |
| 1.2 $\mu\text{m}/13.6\mu\text{m}$ | 0.5 $\mu\text{A}$              | 20 $\mu\text{m}/4\mu\text{m}$   | 20 $\mu\text{m}/0.8\mu\text{m}$ | 20 $\mu\text{m}/0.8\mu\text{m}$ | 20 $\mu\text{m}/4\mu\text{m}$ |
| $M_{OT}(W/L)$                     | $M_{OB}(W/L)$                  |                                 |                                 |                                 |                               |
| 12 $\mu\text{m}/0.8\mu\text{m}$   | 12 $\mu\text{m}/6\mu\text{m}$  |                                 |                                 |                                 |                               |

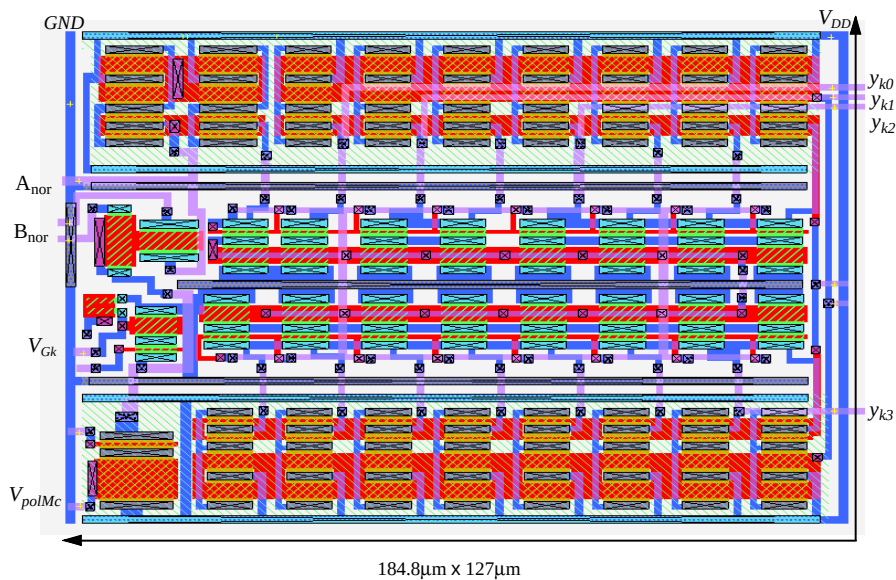


**Fig.3.17** Detalle de *layout* de uno de los dos *Bloques de Regla* empleados en el diseño del controlador ejemplo

### 2.3.3. Consideraciones Globales de Diseño

En este apartado, se pretende abordar algunas cuestiones de diseño que atañen a ambos bloques de alto nivel, tanto al de entrada como al de regla. Principalmente cómo se obtiene la salida global, y cómo se estima el error global del bloque *Núcleo Analógico*. Se incluyen además resultados de simulaciones HSPICE que ilustran el funcionamiento de este bloque integrado en el controlador ejemplo.

#### 2.3.3.1. Generación de la salida del Núcleo Analógico

La salida global del *Núcleo Analógico* se obtiene de agregar las salidas de los  $2^M$  *Bloques de Regla*,

$$y_{NA} = \sum_{k=1 \dots 2^M} y_k \quad (3.22)$$

lo que se hace por simple cableado, ya que las salidas de los *Bloques de Regla* son intensidades. Así pues, concluimos que a partir de los *Bloques de Entrada* y los *Bloques de Regla* se construye el *Núcleo Analógico* completo por simple cableado, obteniendo una salida de intensidad que se demanda en el nudo de salida. Esta circunstancia se ilustra en la Fig.3.18.

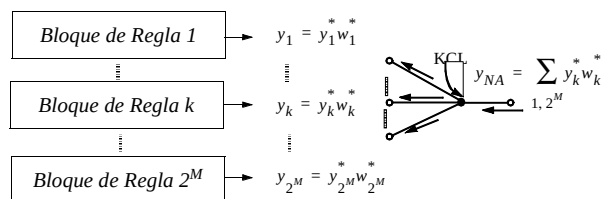


**Fig.3.18 Generación de la salida del Núcleo Analógico por agregación en el nudo de salida de la corriente de salida de todos los Bloques de Regla.**

#### 2.3.3.2. Generación de la salida del Sistema Difuso con Núcleo Analógico Multiplexado

Según la arquitectura propuesta en el Capítulo 2 para el Sistema Difuso con Núcleo Analógico Multiplexado que se recoge en la Fig.3.1, la salida global de este sistema y coincide con la salida  $y_{NA}$  del *Núcleo Analógico* en cualquiera de los intervalos de interpolación en los que este último trabaje, por lo que la salida  $y_{NA}$  en la Fig.3.18 podría ser considerada también como la salida  $y$  que constituye el interfaz analógico externo de salida del controlador. Como se ha indicado más arriba, y también se ilustra en la Fig.3.18, la salida  $y_{NA}$  viene dada como una señal en modo corriente, y de forma que ésta es una corriente demandada por el *Núcleo Analógico*. Sin embargo, en el controlador ejemplo MFCN que ha sido fabricado se ha preferido que esta corriente sea proporcionada por el controlador, por lo que la salida  $y$  del controlador MFCN se ha realizado replicando la corriente en el nudo  $y_{NA}$  mediante el espejo cascode a la salida que muestra la Fig.3.19.

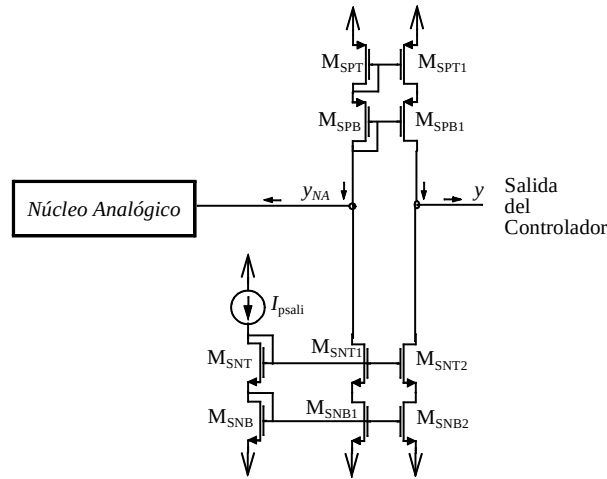


Fig.3.19 Salida en modo corriente del Controlador Difuso MFCON

En la Tabla 3.3 se resumen los tamaños de los transistores e intensidades empleados en el diseño del controlador ejemplo según la notación de esta figura.

Tabla 3.3 : Tamaño de los transistores e intensidades en el circuito de salida.

| $M_{SPT}, M_{SPT1} (W/L)$           | $M_{SPB}, M_{SPB1} (W/L)$           | $I_{psali}$      |
|-------------------------------------|-------------------------------------|------------------|
| 100 $\mu\text{m}/25\mu\text{m}$     | 5 $\mu\text{m}/5\mu\text{m}$        | 10 $\mu\text{A}$ |
| $M_{SNT}, M_{SNT1}, M_{SNT2} (W/L)$ | $M_{SNB}, M_{SNB1}, M_{SNB2} (W/L)$ |                  |
| 60 $\mu\text{m}/4\mu\text{m}$       | 60 $\mu\text{m}/20\mu\text{m}$      |                  |

### 2.3.3.3. Simulaciones HSPICE del Núcleo Analógico integrado en el controlador MFCON

A fin de ilustrar el comportamiento del *Núcleo Analógico* dentro en el controlador MFCON, a continuación se muestra el resultado de diversas simulaciones HSPICE. La Fig.3.21 recoge el resultado de un análisis transitorio en el que la entrada  $x_1$  describe una rampa a través del espacio de entrada, mientras  $x_2$  permanece constante. La curva resultante muestra una sección de la superficie de control resultado de programar el controlador con valores *singleton* que alternan entre su valor mínimo y máximo sucesivamente a lo largo del espacio de entrada bidimensional. En ella se aprecia el efecto que produce en la salida y del controlador la transición entre intervalos de interpolación. Este efecto queda más patente en la simulación que se muestra en la Fig.3.20, donde se muestra la respuesta y del controlador, (curva inferior en la Fig.3.20(a)), a una secuencia de entrada  $x_1$  (en la parte superior de la Fig.3.20(a)), mientras la entrada  $x_2$  permanece a un valor constante. En esta secuencia la entrada evoluciona

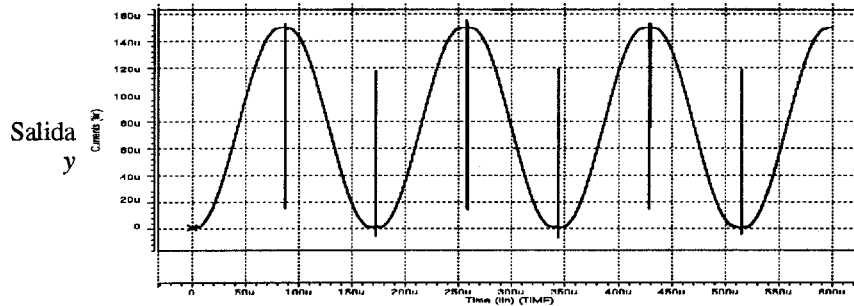


**Fig.3.21 Respuesta transitoria a una entrada en rampa que recorre todo el espacio de entrada sobre una superficie de control** resultado de programar el controlador con valores *singleton* que alternan entre su valor mínimo y máximo sucesivamente a lo largo del espacio de entrada bidimensional

a través de diferentes intervalos de interpolación de manera que el *Núcleo Analógico* ha de ser reprogramado, según la evolución de los valores *singleton* que muestra la Fig.3.20(b). En esta gráfica se observa cómo en la simulación se han considerado diferentes posibilidades, de forma que algunas de las transiciones de la entrada  $x_1$  entre intervalos diferentes suponen una reprogramación pero no un cambio en el valor *singleton* programado, por ejemplo la transición de la entrada entre los valores 2.45volts. y 3.25volts o bien la transición entre los valores 4.35 volts y 2.75volts; mientras que en otras transiciones de entrada, por ejemplo la transición entre

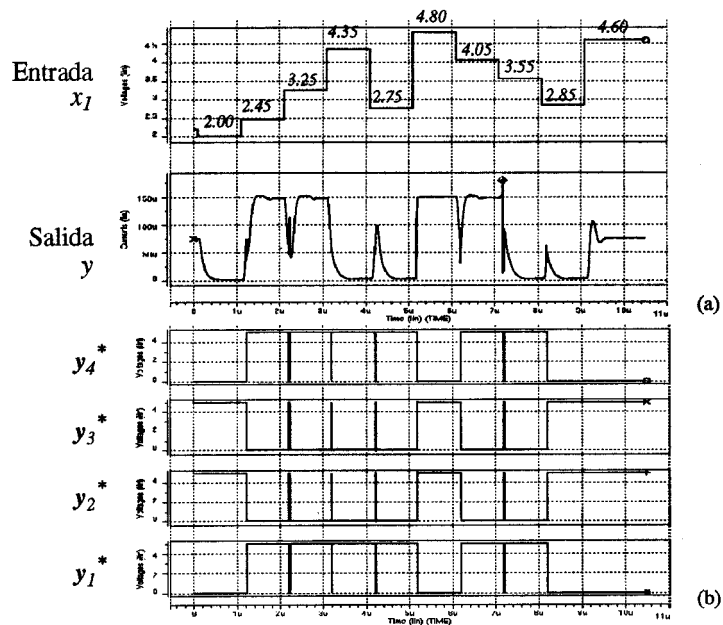


**Fig.3.20 Simulación HSPICE que ilustra el funcionamiento del Núcleo Analógico para una secuencia de entrada: Entrada  $x_1$  y salida  $y$  (a); líneas de programación *singleton* (b).**

los valores 2.00 volts y 2.45 volts, o bien entre los valores 3.25 volts y 4.35 volts, la reprogramación supone además un cambio en los valores *singleton* programados. En estos últimos casos el transitorio entre intervalos queda prácticamente enmascarado por el transitorio en la salida del controlador. Por su parte, el efecto de la transición entre intervalos se hace más patente en aquellos casos en los que el valor de la salida ha de ser el mismo en el intervalo de origen como en el de destino. La respuesta que en estos casos refleja la Fig.3.20 está justificada por el hecho de que en cada uno de esos intervalos la salida ha de ser proporcionada por una regla distinta, por lo que el efecto que se observa es justamente la desactivación de las reglas en el intervalo de origen y la activación de las reglas en el intervalo destino. En cualquiera de los casos el valor de la señal queda restablecido antes de los 500ns, según ha sido medido sobre estas simulaciones. Este tiempo prácticamente coincide con el tiempo de retardo medido para los casos en los que la señal de salida ha de transitar entre sus valores extremos. Por ello se estima este tiempo como el tiempo de retardo del controlador, lo que ha sido confirmado por las medidas experimentales sobre el prototipo fabricado, como se muestra en el Capítulo 6.

#### 2.3.3.4. Estimación del Error

En primer lugar, antes de estimar el error global, cabe detenerse aquí en el diseño para conseguir que el circuito de normalización trabaje fuera de su rango diferencial. Para ello, se obtienen las entradas máxima  $w_{kmax}$  y mínima  $I_{OS}$  de corriente a la celda del circuito de normalización en los dos casos extremos que reporta la tecnología, el "FAST" y el "SLOW". Obsérvese que la condición (3.13) puede escribirse también como

$$\frac{(W/L)_{Nb}}{(W/L)_{Nt}} \geq \left( \frac{\sqrt{w_k} - \sqrt{I_{OS}}}{\sqrt{I_{ss}}} \right)^2 \quad (3.23)$$

que es independiente de la tecnología. La ecuación (3.23) se impone ahora para los valores obtenidos de  $w_k$  (máximo valor) e  $I_{OS}$ , y se toma la razón  $\frac{(W/L)_{Nb}}{(W/L)_{Nt}}$  que verifica la misma tanto para el caso "FAST" como para el "SLOW".

El hecho de que se garantice que cuando una entrada sea máxima y el resto mínimas sólo la salida máxima del circuito de normalización conduzca corriente, y ésta sea la intensidad de normalización  $I_{ss}$ , simplifica la estimación del error a la salida. Esto es así porque estas condiciones se dan justamente en los núcleos de los conjuntos difusos, que coinciden con los puntos o áreas del espacio de entrada para los que se conocen los valores deseados de la función que se desea aproximar, y que según la discusión del apartado 4.1. del Capítulo 2, coinciden con los valores de programación *singleton*. Son por tanto estos puntos los más sensibles o adecuados para estimar el error, por cuanto el resto de los puntos son puntos de interpolación que únicamente han de cumplir con una condición de suavidad y monotonía.

En este sentido, para estimar el error debemos tener en cuenta los errores sistemáticos introducidos, que son básicamente los debidos a los distintos espejos de corriente en Fig.3.14(a), así como los errores con origen en el desapareamiento de transistores. Comoquiera que se han utilizado configuraciones cascode para implementar los espejos de corriente, los errores sistemáticos están muy por debajo del 1% del fondo de escala, por lo que la cota de error va a venir claramente impuesta por los errores debidos al desapareamiento de los transistores.

El desapareamiento de los transistores tiene origen en las no idealidades del proceso de producción, que se traducen en una variación de los parámetros que rigen el comportamiento de un transistor, principalmente del factor de transconductancia  $\beta$ , la tensión umbral sin polarizar,  $V_{T0}$ , y el parámetro de efecto sustrato  $\gamma$ . Así, dado un transistor con parámetros  $\beta$ ,  $V_{T0}$  y  $\gamma$ , otro transistor de igual geometría tiene en realidad parámetros distintos  $\beta'$ ,  $V_{T0}'$  y  $\gamma'$ , de forma que,

$$\begin{aligned}\beta' &= \beta + \Delta\beta \\ V_{T0}' &= V_{T0} + \Delta V_{T0} \\ \gamma' &= \gamma + \Delta\gamma\end{aligned}\tag{3.24}$$

Los incrementos  $\Delta\beta$ ,  $\Delta V_{T0}$  y  $\Delta\gamma$  se pueden modelar como variables estadísticas con valor medio cero [Pelg89], cuyas desviaciones estándar dependen del área del canal y de la distancia  $D$  entre los dos dispositivos como sigue,

$$\begin{aligned}\sigma^2(\Delta V_{T0}) &= \frac{A^2_{V_{T0}}}{WL} + S_{V_{T0}}^2 D^2 \\ \sigma^2(\Delta\gamma) &= \frac{A^2_{\gamma}}{WL} + S_{\gamma}^2 D^2 \\ \sigma^2\left(\frac{\Delta\beta}{\beta}\right) &= \frac{A^2_{\beta}}{WL} + S_{\beta}^2 D^2\end{aligned}\tag{3.25}$$

Donde  $A_p$  y  $S_p$  ( $P$  es cualquier parámetro) son constantes dependientes del proceso. Obsérvese que cada una de las expresiones en (3.25) tiene dos sumandos, el primero depende en forma inversamente proporcional al área del canal, mientras que el segundo depende de la distancia a la que se encuentren los dispositivos. Según lo reportado en [Pelg89], el término dependiente de la distancia se puede despreciar en transistores menores de  $100\mu\text{m}^2$ . Para nuestro análisis, supondremos que los transistores utilizados tienen áreas menores. En la práctica, los transistores de áreas grandes son divididos en varios transistores conectados en paralelo, con lo que esta aproximación sigue siendo válida.

Para la determinación de los parámetros  $A_p$  utilizamos los datos dados en [Pelg89] para una tecnología n-Well de 50nm de espesor del óxido de puerta y  $2.5\mu\text{m}$ , y otra de 25nm de espesor de puerta y  $1.6\mu\text{m}$ , basándonos en la afirmación de que estos datos no varían



significativamente de un proceso de fabricación a otro, en tecnologías de similares características [Pelg89]. El prototipo presentado en este trabajo ha sido realizado con una tecnología n-Well de 15nm de óxido de puerta y 0.7μm respectivamente. Dado que los datos en [Pelg89] para la tecnología de 1.6μm no están completos, y no se da ninguno para una tecnología de 0.7μm, se hace necesario extrapolar los presentados para la tecnología de 2.5μm. Esto es lo que se hace en [Espe94] para completar los datos de una tecnología de 2.5μm, siguiendo básicamente los resultados en [Pelg89] según los cuales las constantes  $A_{VT0n,p}$  y  $A_{\gamma n,p}$  dependen en forma lineal del espesor del óxido de puerta, mientras que  $A_{\beta n,p}$  permanece aproximadamente constante. Siguiendo estos criterios, para un proceso de fabricación de escritura directa en la oblea (DWW) se utilizan los valores que se muestran en la Tabla 3.4.

**Tabla 3.4:** Valores de los parámetros utilizados para la estimación de los errores por desapareamiento de transistores según [Pelg89].

| $A_{VT0n}$ | $A_{VT0p}$ | $A_{\beta n}$ | $A_{\beta p}$ | $A_{\gamma n}$                    | $A_{\gamma p}$                    |
|------------|------------|---------------|---------------|-----------------------------------|-----------------------------------|
| 9mVμm      | 10.8mVμm   | 3.3%μm        | 4.5%μm        | $4.8 \cdot 10^{-3} V^{0.5} \mu m$ | $3.6 \cdot 10^{-3} V^{0.5} \mu m$ |

Para estudiar la influencia del desapareamiento entre transistores en los circuitos propuestos, debemos tener en cuenta que los coeficientes en la Tabla 3.4 se han obtenido según la definición en (3.25), es decir se corresponden con la varianza de la diferencia de un determinado parámetro medido en dos transistores. Sin embargo, como hemos despreciado el factor dependiente de la distancia, las dos variables aleatorias relativas al valor del parámetro en ambos transistores son estadísticamente independientes, por lo que,

$$\sigma^2(\Delta P) = 2\sigma^2(P) \quad (3.26)$$

y por tanto podemos modelar los parámetros de cada transistor como variables independientes de varianzas dadas por,

$$\begin{aligned} \sigma^2(V_{T0}) &= \frac{A_{VT0}^2}{WL} \\ \sigma^2(\gamma) &= \frac{A_{\gamma}^2}{WL} \\ \frac{\sigma^2(\beta)}{\beta^2} &= \frac{A_{\beta}^2}{WL} \end{aligned} \quad (3.27)$$

Para estimar el error cometido por desapareamiento en un núcleo de conjunto difuso, suponemos que una de las celdas del circuito de normalización conduce toda la intensidad de normalización. Comoquiera que los transistores cascode influyen muy poco en el error por

desapareamiento [Wang90], podemos utilizar el esquemático en Fig.3.23, donde se indican los valores medios de las intensidades, para calcular el error. Nótese que el error se calcula básicamente sumando los errores introducidos por los sucesivos espejos de corriente.

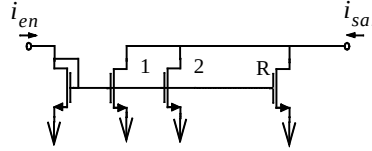


Fig.3.22 Espejo de corriente con R ramas de salida iguales.

En Fig.3.22 se ilustra un espejo de corriente con R ramas de salida, como el que implementa la ponderación por el valor *singleton*. El error relativo a la salida viene caracterizado por la expresión [Vida96b],

$$\frac{\sigma^2(i_{sal})}{i_{sal}^2} = \frac{\sigma^2(i_{en})}{i_{en}^2} + \left[ \frac{\sigma^2(\beta)}{\beta^2} + \frac{4\sigma^2(V_{T0})}{(V_G - V_{T0})^2} \right] \left( 1 + \frac{1}{R} \right) \quad (3.28)$$

de donde se deduce que el error es máximo para  $R=1$ , es decir cuando el valor *singleton*, o la ganancia del espejo, es la unidad. Esta es la razón por la que en Fig.3.23 se ha utilizado ganancia uno en los espejos de corriente asociados a la regla que aporta la intensidad  $I_{ss}$ . Considerando los parámetros de los transistores como estadísticamente independientes, y la expresión (3.28) para caracterizar el error introducido por cada espejo, el error debido a esta regla viene caracterizado por la expresión,

$$\begin{aligned} \sigma^2(y_{I_{ss}}) = & I_{ss}^2 \frac{\sigma^2(\beta_{SB})}{\beta_{SB}^2} + I_{ss} 4\beta_{SB} \sigma^2(V_{TSB}) + \\ & + 2I_{off}^2 \frac{\sigma^2(\beta_{OB})}{\beta_{OB}^2} + I_{off} 8\beta_{OB} \sigma^2(V_{TOB}) + \\ & + 2(I_{off} + I_{ss})^2 \frac{\sigma^2(\beta_{PS})}{\beta_{PS}^2} + 8(I_{off} + I_{ss})\beta_{PS} \sigma^2(V_{TPS}) + \\ & + 2(I_{off} + I_{ss})^2 \frac{\sigma^2(\beta_{YB})}{\beta_{YB}^2} + 8(I_{off} + I_{ss})\beta_{YB} \sigma^2(V_{TYS}) \\ & + 2I_{off}^2 \frac{\sigma^2(\beta_{PY})}{\beta_{PY}^2} + 8I_{off}\beta_{PY} \sigma^2(V_{TPY}) \end{aligned} \quad (3.29)$$

Sin embargo, aunque la salida de las restantes reglas sea nominalmente cero, y ya que este valor no es el resultado de una rectificación que suprima completamente el paso de corriente, sino el resultado de restar dos cantidades teóricamente iguales, éstas reglas aportan realmente un error a la salida, y este error puede crecer con la ganancia del espejo, puesto que una de sus componentes es precisamente el error a la entrada del espejo amplificado por la ganancia del mismo. Así, en Fig.3.23 se considera que estos espejos tienen ganancia  $y^*$ , en general distinta de uno, y el error a la salida aportado por estas reglas viene caracterizado por,

$$\begin{aligned} \sigma^2(y_0) = & 2(y^*I_{off})^2 \left[ \frac{\sigma^2(\beta_{OB})}{\beta_{OB}^2} + \frac{4\beta_{SB}}{I_{off}} \sigma^2(V_{TOB}) \right] + \\ & + 2(y^*I_{off})^2 \left[ \frac{\sigma^2(\beta_{PS})}{\beta_{PS}^2} + \frac{4\beta_{PS}}{I_{off}} \sigma^2(V_{TPS}) \right] + \\ & + \left(1 + \frac{1}{y^*}\right) (y^*I_{off})^2 \left[ \frac{\sigma^2(\beta_{YB})}{\beta_{YB}^2} + \frac{4\beta_{YB}}{I_{off}} \sigma^2(V_{TYB}) \right] + \\ & + \left(1 + \frac{1}{y^*}\right) (y^*I_{off})^2 \left[ \frac{\sigma^2(\beta_{PY})}{\beta_{PY}^2} + \frac{4\beta_{PY}}{I_{off}} \sigma^2(V_{TPY}) \right] \end{aligned} \quad (3.30)$$

siendo el error global a la salida determinado por,

$$\sigma^2(y) = \sigma^2(y_{I_{ss}}) + 3\sigma^2(y_0) \quad (3.31)$$

Para una determinada cota de error deseada, asimilable a una desviación estándar o varianza, la ecuación (3.31) se ajusta para cumplir esta cota. Evidentemente, esta ecuación tiene demasiados parámetros para ser ajustada a mano, por lo que es aconsejable utilizar un asistente matemático como MATLAB®, y ajustar los parámetros tras diversas iteraciones.

El procedimiento de diseño puede ser ahora poco conservador, y ajustar los parámetros para que el valor de  $3\sigma$  sea menor al 10% del fondo de escala, o bien más conservador, y buscar un error relativo menor. En el controlador ejemplo de esta tesis se siguió la primera aproximación, donde se hizo que  $\sigma(y) = 3.9\mu A$ , y el error medido en el laboratorio fue aproximadamente del 10% del fondo de escala ( $150\mu A$ ). La estrategia de multiplexado hace que en este controlador no sea posible reducir el error haciendo una corrección en el nudo de salida, puesto que este error depende en realidad del intervalo de interpolación en el que estemos, es decir del valor de la entrada. Un ajuste más conservador puede conseguirse con bastante facilidad reduciendo la intensidad  $I_{off}$ , que en el caso del controlador ejemplo vale  $10\mu A$ , y que influye mucho en el error global, como se puede observar en las expresiones (3.29)

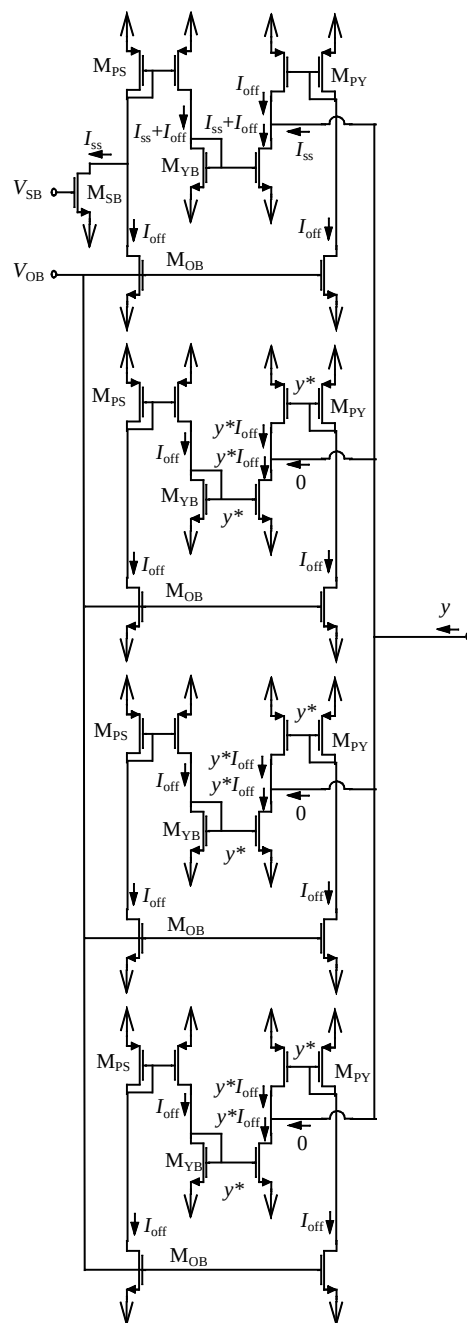


Fig.3.23 Circuito simplificado para la estimación del error a la salida.

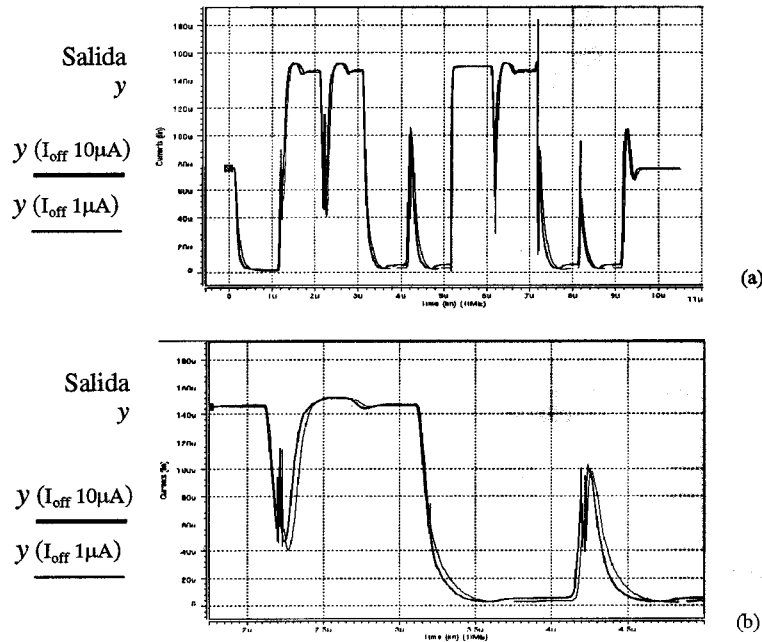


Fig.3.24 Comparación de la respuesta dinámica del controlador para diferentes valores de la corriente de polarización  $I_{off}$ : Gráfica general (a); detalle (b)

y (3.30). Un valor de  $1\mu A$  para esta intensidad hace que  $\sigma(y) = 1.1\mu A$ , aunque esto supone un cierto deterioro de la respuesta dinámica del espejo programable encargado de la ponderación por el valor *singleton*. Este deterioro no es sin embargo excesivo, como se puede observar en la Fig.3.24, donde la respuesta dinámica se ha simulado para  $I_{off} = 10\mu A$  y para  $I_{off} = 1\mu A$ .

## Capítulo 4

### Descripción Funcional y Diseño CMOS de los Bloques de Multiplexado (I): Selector de Intervalo.

#### 1. INTRODUCCIÓN

Considerados en conjunto, los bloques *Selector de Intervalo*, *Programador de Antecedente de Reglas* y *Programador de Consecuente de Reglas* constituyen el *Bloque de Multiplexado* del *Núcleo Analógico*, el cual permite reconfigurar el sistema de inferencia difuso simple, que este último realiza, para trabajar en cualquier región del espacio de entrada y proporcionar la salida del sistema difuso global. La arquitectura que sirve de soporte a este sistema se caracteriza por su capacidad de procesamiento paralelo, así como por su funcionamiento asíncrono, con respuesta en tiempo continuo, como queda puesto de manifiesto en la Fig.4.1 en la que no aparece ninguna señal de reloj que sincronice su funcionamiento. Por su parte, el tiempo de respuesta del sistema, y por tanto su velocidad de operación, está limitada

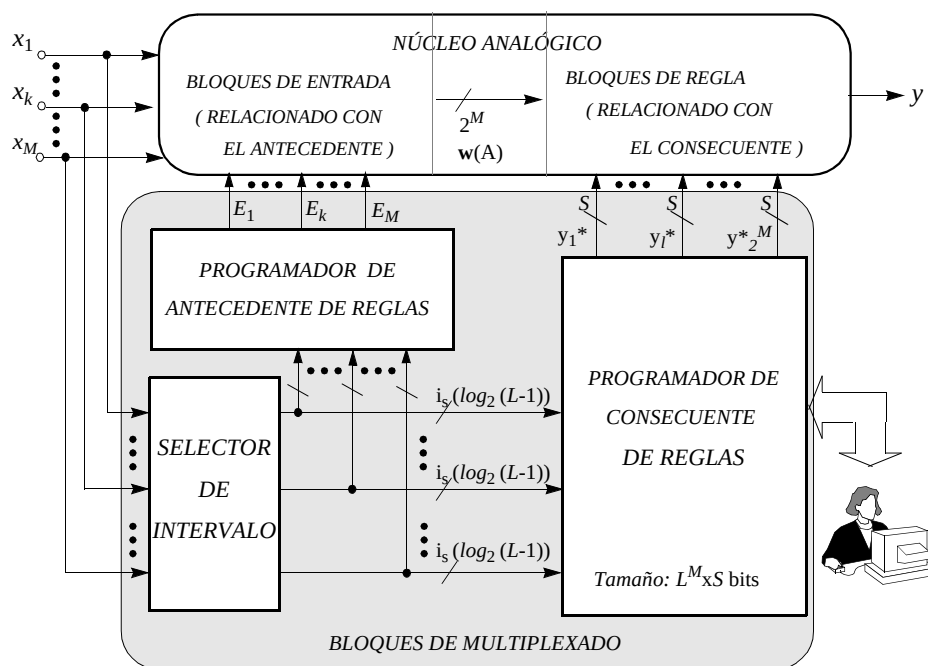


Fig.4.1 Arquitectura global de un Controlador difuso con Núcleo analógico Multiplexado.

únicamente por el tiempo de respuesta de los circuitos que lo conforman. En el caso de que el vector de entrada permanezca siempre dentro de los valores correspondientes a un mismo intervalo de interpolación, el tiempo de respuesta del sistema viene determinado por el tiempo de procesamiento propio del *Núcleo Analógico*, ya que en esta situación, el conjunto de parámetros que proporcionan los *Bloques de Multiplexado* permanece inalterado. Sin embargo, en el caso de que un cambio en el vector de entrada suponga un cambio en el intervalo de interpolación, el *Núcleo Analógico* tiene que ser reconfigurado, por lo que ahora el tiempo de respuesta del sistema estará determinado por tres componentes, y al retardo propio de procesamiento de éste habrá que añadir el retardo debido a su reconfiguración, y el retardo de operación de los *Bloques de Multiplexado*. Claramente, es este segundo caso el que resulta más crítico a la hora de determinar un valor que caracterice el tiempo de respuesta del sistema global. Por tanto, y con el fin de penalizar lo menos posible las buenas prestaciones de velocidad del *Núcleo Analógico*, uno de los principales requisitos que se han de contemplar en el diseño de estos *Bloques de Multiplexado* es la necesidad de una alta velocidad de operación, en comparación con la velocidad de procesamiento del *Núcleo Analógico*. Sobre todo, teniendo en cuenta que, como ya se ha mencionado, el retardo en la identificación del intervalo de interpolación y en la programación del núcleo, afecta directamente al retardo global del sistema, puesto que el *Núcleo Analógico* no proporcionará una salida válida hasta que no esté configurado correctamente.

Este requerimiento de velocidad es sin duda más relevante para el bloque *Selector de Intervalo*, ya que es éste el encargado de generar la información necesaria que permite a los demás bloques seleccionar los valores adecuados de los parámetros de programación. Por otra parte, y una vez configurado el *antecedente de regla*, el procesamiento del *Núcleo Analógico* progresa en paralelo con el proceso de obtención de los valores *singleton* en el bloque *Programador de Consecuente de Reglas*, por lo que el requisito de velocidad para este último, aunque necesario, resulta menos crítico y se hace relativo a la velocidad de procesado del *Bloque de Etiquetas* del *Núcleo Analógico*, de forma que no suponga una acumulación de retardo adicional.

En el presente capítulo, se contempla la descripción del bloque *Selector de Intervalo*, mientras que los bloques *Programador de Antecedente de Reglas*, y *Programador de Consecuente de Reglas* serán abordados en el capítulo siguiente. Para cada uno de ellos se realiza en primer lugar una descripción funcional destacando sus principales requerimientos y especificaciones de diseño, y a continuación se describen y analizan los circuitos empleados en el diseño del controlador ejemplo MFCON que posee dos entradas, una salida y sesenta y cuatro reglas, y que ha sido fabricado con tecnología CMOS estándar de  $0.7\mu\text{m}$ .

## 2. SELECTOR DE INTERVALO

La principal misión de este bloque es identificar el intervalo de interpolación  $C_{k_1 \dots k_M}$  al que pertenece un vector de entrada  $\mathbf{x} = \{x_1, x_2, \dots, x_M\}^T$  dado, y proporcionar a los correspondientes bloques de programación las adecuadas señales de control que les permitan poder seleccionar, de entre el conjunto completo de parámetros que definen al sistema global, el conjunto de valores de los parámetros que configuran a las reglas activas,

$$\begin{aligned} PA_{C_{k_1 \dots k_M}} &= \{E_{k_1^*}, \dots, E_{k_l^*}, \dots, E_{k_M^*}\} \text{ para los antecedentes de regla y} \\ PC_{C_{k_1 \dots k_M}} &= \{y_1, \dots, y_l, \dots, y_{2^M}\} \text{ para sus consecuentes, (ver Fig.4.1).} \end{aligned}$$

### 2.1. Discusión de los requerimientos y objetivos de diseño.

Para llevar a cabo su tarea, en este bloque se realizan dos funciones básicas:

- Generar la partición en rejilla del universo de discurso.
- Codificar cada uno de los intervalos de interpolación para su identificación.

Ambas resultan estar interrelacionadas.

Una forma cómoda de identificar cada uno de los intervalos es codificarlos en formato digital. Como el número total de intervalos de interpolación es  $(L-1)^M$ , para poder codificarlos se necesitará una palabra,  $AC_{k_1 \dots k_M}$ , de al menos  $M \times i_s(\log_2(L-1))$  bits. Esta palabra digital será usada para seleccionar los valores de los parámetros que permitirán al bloque *Programador de Antecedente de Reglas* configurar los antecedentes de regla en los *Bloques de Entrada del Núcleo Analógico*; y para acceder a la memoria del bloque *Programador de Consecuente de Reglas* para obtener los valores *singleton* de las reglas activas. Como con la notación  $C_{k_1 \dots k_M}$  cada intervalo  $M$ -dimensional queda identificado por  $M$  subíndices  $k_i$  donde cada uno de ellos toma valores comprendidos en el intervalo  $1 \leq k_i \leq L-1$ , cada uno de estos subíndices puede ser expresado en binario mediante una palabra  $A_{ib}$  de  $i_s(\log_2(L-1))$  bits, y consecuentemente resulta cómodo construir  $AC_{k_1 \dots k_M} = A_{1b} \dots A_{Mb}$ . Dado que, en la mayoría de las aplicaciones de control, el número de  $L$  de etiquetas empleado es pequeño, (entre cinco y siete etiquetas por entrada), lo será también el número de bit de  $A_{ib}$ , y por tanto las necesidades de resolución en la identificación de los intervalos.

Para realizar la partición en rejilla hay que generar los valores de tensión correspondientes a los puntos del espacio de entrada  $\varepsilon_{ij_i}$ , con  $(1 \leq j_i \leq L)$  y  $(1 \leq i \leq M)$ , que la definen y que también determinan cada intervalo  $M$ -dimensional  $C_{k_1 \dots k_M}$  según la expresión (2.5) del Capítulo 2.

Como ya se ha señalado en la introducción de este capítulo, el principal requerimiento de diseño del *Selector de Intervalo* apunta a su velocidad de operación. El retardo en la



identificación del intervalo de interpolación afecta directamente al retardo global del sistema, puesto que el *Núcleo Analógico* no estará programado correctamente, y por tanto su salida no será válida hasta que el intervalo de interpolación no haya sido identificado, por lo que al retardo propio del procesamiento analógico que el núcleo realiza habrá que sumar el tiempo empleado en su correcta configuración.

Por otra parte, una correcta identificación del intervalo sobre el que se ha de trabajar es crítico para el buen funcionamiento del sistema. En este sentido puede resultar especialmente problemático el caso en que las señales de entrada tomen valores próximos a aquellos que definen los límites de la partición del sistema en intervalos de interpolación, sobre todo para prevenir el caso en el que estas señales de entrada lleven asociado cierto nivel de ruido y presenten una dinámica de evolución lenta. (Este puede ser un caso típico de aplicaciones que se desarrollen en entornos ruidosos y con cierta incertidumbre, en el que los sistemas difusos se han mostrado bastante útiles.) En esta situación el *Núcleo Analógico* puede ser reprogramado innecesariamente por los *Bloques de Multiplexado*, y puede verse obligado a cambiar frecuentemente, y a la vez erróneamente, entre intervalos adyacentes, de forma que, aunque la salida del sistema no tuviese que cambiar de valor, o cambiase de forma suave, sin embargo sufriría los efectos de la conmutación de intervalo y por tanto cierta inestabilidad. Por tanto, otro requerimiento importante de diseño es que la identificación del intervalo de interpolación se haga con cierta inmunidad al ruido.

## 2.2. Propuesta de diseño

Para el diseño de este bloque se propone utilizar  $M$  convertidores analógico/digital, uno por entrada, con una resolución de  $i_s(\log_2(L-1))$  bits cada uno, como muestra la Fig.4.2(a). En conjunto proporcionan la palabra  $AC_{k_1 \dots k_M} = A_{1b} \dots A_{Mb}$  de  $M \times i_s(\log_2(L-1))$  bits que identifica a cada uno de los intervalos de interpolación.

En la literatura se pueden encontrar diferentes implementaciones de convertidores A/D [Chen95] [Alle87]. En esta propuesta, debido principalmente al funcionamiento asíncrono, y a la necesidad de una alta velocidad de operación y la baja resolución necesaria en la identificación de los intervalos, se ha optado por emplear convertidores del tipo paralelo o "flash", trabajando en tiempo continuo. La Fig.4.2(b) muestra la arquitectura genérica de un convertidor A/D paralelo de  $p$  bits de resolución empleado en este diseño. Este convertidor emplea  $2^p - 1$  comparadores analógicos que deciden si un valor de entrada  $x_i$  supera los diferentes niveles de referencia  $\epsilon_{ij}$ , que definen los  $2^p$  intervalos de cuantización posibles. Los valores de tensión que sirven de referencia son generados mediante un *array* resistivo, mientras que el código digital,  $A_{ib}[0:p]$ , es obtenido a partir de la salida de los comparadores,  $D_{ij}$  con  $(1 \leq j \leq L-1)$ , mediante un bloque digital codificador. En este sistema, para cada

entrada  $x_i$  se tiene el correspondiente código tras un tiempo de retardo, resultado de la suma del tiempo de retardo del bloque de comparadores, considerando como tal el retardo de propagación del comparador más lento, más el tiempo de propagación del codificador. Por otra parte, y dado el funcionamiento asíncrono, el código de salida permanecerá estable mientras la entrada  $x_i$  permanezca tomando valores dentro de un mismo intervalo. Nótese que en este diseño se ha prescindido del circuito de retención y muestreo, debido a su modo de operación en tiempo continuo.

Frente a su alta velocidad de operación, la principal desventaja de este tipo de convertidores A/D es su elevado consumo de área debido a que el número de comparadores que son necesarios crece con la resolución  $p$  del convertor de modo exponencial, lo que hace que, en la práctica, este tipo de convertidores sólo se utilicen cuando se requieren resoluciones pequeñas, en general inferiores a los ocho bits. Por otra parte, la generación de los valores de referencia mediante un *array* resistivo puede resultar también costosa, tanto en área, como en consumo de potencia.

Sin embargo, como ya se ha indicado, para este tipo de aplicaciones la resolución requerida,  $p = i_s(\log_2(L-1))$  bits, es en general baja,  $L$  raras veces suele ser mayor que siete u ocho. Por otra parte, una resolución de ocho bits supone un sistema con  $L = 256$ , que resulta muy elevada para muchos de los problemas prácticos de control. Por tanto, en este aspecto, la elección de un convertor A/D paralelo resulta adecuada.

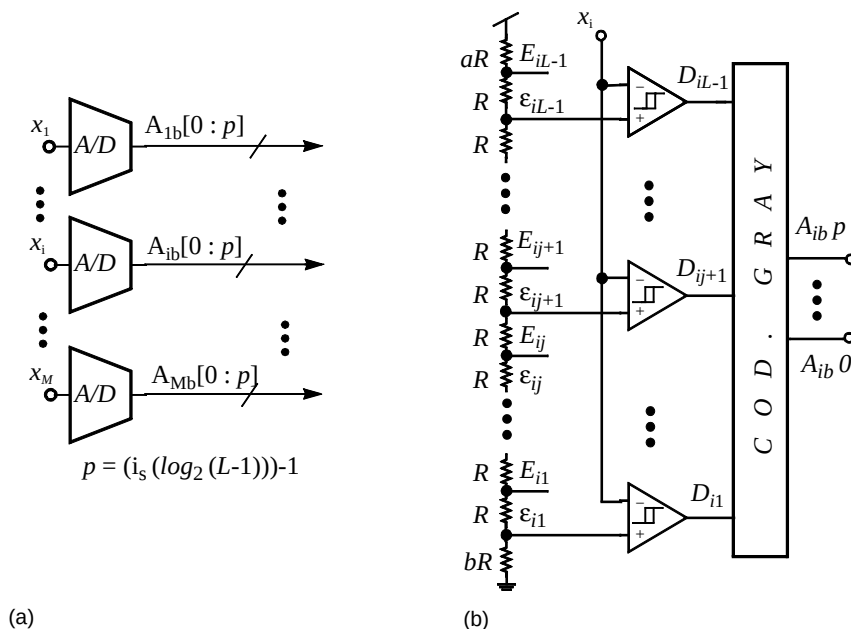


Fig.4.2 (a) Organización interna del bloque Selector de Intervalo ; (b) Convertidor genérico A/D de tipo "Flash" con histéresis .

Para proporcionar cierto grado de inmunidad frente al ruido que puede acompañar a las señales de entrada del sistema, se propone diseñar los comparadores que han de ser empleados en los conversores con cierto grado de histéresis, y además, utilizar una codificación del tipo Gray para convertir la escala en termómetro, formada por las salidas de dichos comparadores  $D_{ij}$  con  $(1 \leq j \leq L-1)$ , en un código Gray. Esta codificación de las salidas  $A_{ib}[0:p]$  permite minimizar el número de líneas que conmutan simultáneamente, sobre todo para transiciones entre intervalos adyacentes, debidas a cambios graduales en las variables de control, que es de suponer serán las más frecuentes, lo que permitirá reducir los riesgos lógicos o "azares" debidos a carreras y minimiza el ruido que puede ser inyectado por estas líneas en el resto del sistema.

La partición en rejilla, definida según la expresión (2.5) en el Capítulo 2 y los valores de tensión  $\varepsilon_{ij_i}$  que definen los límites de los diferentes intervalos de interpolación, se fijan por diseño al determinar la serie de resistencias que emplea cada convertidor A/D. Simultáneamente, dicha serie de resistencias puede proporcionar también los valores  $E_{ij_i}$  que constituyen los diferentes valores de tensión que sirven de programación de los antecedentes de regla de los *Bloques de Entrada* en el *Núcleo Analógico*, de manera que, puede decirse que

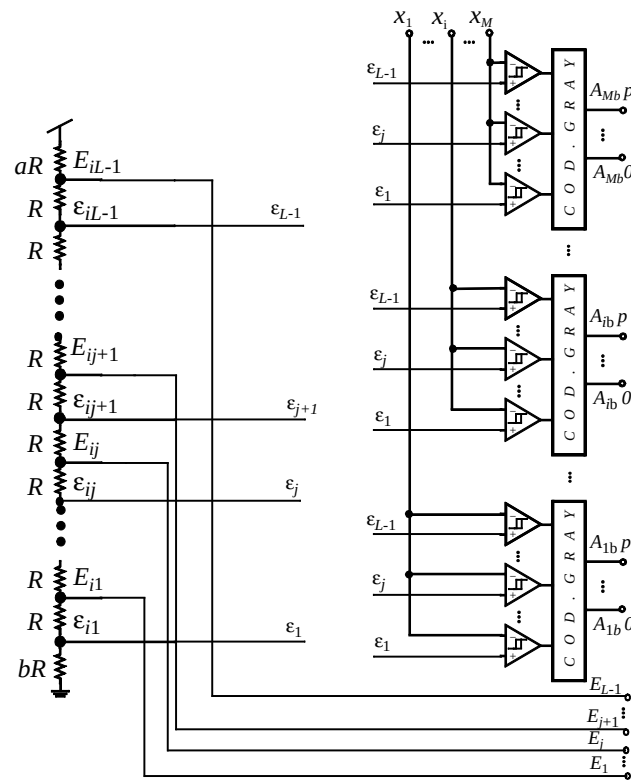


Fig.4.3 Organización interna del bloque Selector de Intervalo, con serie de resistencias compartida.

en conjunto, estos *arrays* resistivos, constituyen una especie de memoria analógica de solo lectura (ver Fig.4.2(b)).

Por otra parte, si se considera una partición en rejilla uniforme para todas las entradas, puede compartirse una única serie de resistores lineales para generar todos los valores de referencia (véase  $\varepsilon_j$  y  $E_j$  con  $(1 \leq j \leq L-1)$  en la Fig.4.3, en donde se ha eliminado el subíndice  $i$  al considerar ya un único valor, que es compartido por todas las entradas), con el consiguiente ahorro en área y consumo de potencia.

En esta situación el módulo de dicha partición  $h$  es constante y puede obtenerse como la diferencia entre dos valores de referencia contiguos cualesquiera  $h = \varepsilon_{j+1} - \varepsilon_j$ , además se tiene que  $E_j = \varepsilon_j + \frac{h}{2}$ , y que  $\varepsilon_{j+1} = E_j + \frac{h}{2}$ , y por tanto se verifica que  $h = E_{j+1} - E_j$ . Esta circunstancia permite un diseño modular de dicho array resistivo como muestran la Fig.4.2(b) y la Fig.4.3, a partir de un valor unitario  $R$ .

### 2.3. Diseño CMOS del bloque Selector de Intervalo.

A continuación se analizan y caracterizan los circuitos y elementos empleados para realizar el diseño propuesto en el apartado anterior para tecnologías CMOS estándar.

#### 2.3.1. Array resistivo.

Como ya se ha mencionado, los valores de tensión  $\varepsilon_j$  que determinan la partición en rejilla del espacio de entrada, y por tanto sirven de referencia para los comparadores de todos los conversores A/D; así como los valores de tensión  $E_j$  que sirven de programación de los antecedentes de regla del *Núcleo Analógico*, serán generados mediante un array resistivo (ver Fig.4.3). En los circuitos integrados los resistores pasivos se realizan a partir de tiras de capas resistivas, las cuales que se caracterizan por presentar una resistencia por cuadrado  $R_c$ , medida en ohmios/cuadrado  $\Omega/sq$ , denominada *sheet resistance* en terminología inglesa. El valor del resistor  $R$  se obtiene entonces como el producto de  $R_c$  y la razón  $L_R/W_R$ , longitud/anchura, de la tira empleada en su realización

$$R = R_c \cdot \frac{L_R}{W_R} \quad (4.1)$$

Debido a que, para la mayoría de las capas resistivas, el valor de  $R_c$ , es pequeño, para realizar resistencias de valores elevados es necesario emplear valores elevados de la razón  $L_R/W_R$ . Por ello, para optimizar la ocupación de área, es de uso común el empleo de estructuras en serpentina de anchura fija como la mostrada en la Fig.4.4(a). De las diferentes capas con las que se realiza un resistor en tecnologías CMOS, en este diseño se ha empleado una tira de polisilicio. Esta opción es la más habitual para esta tecnología, debido principalmente a la

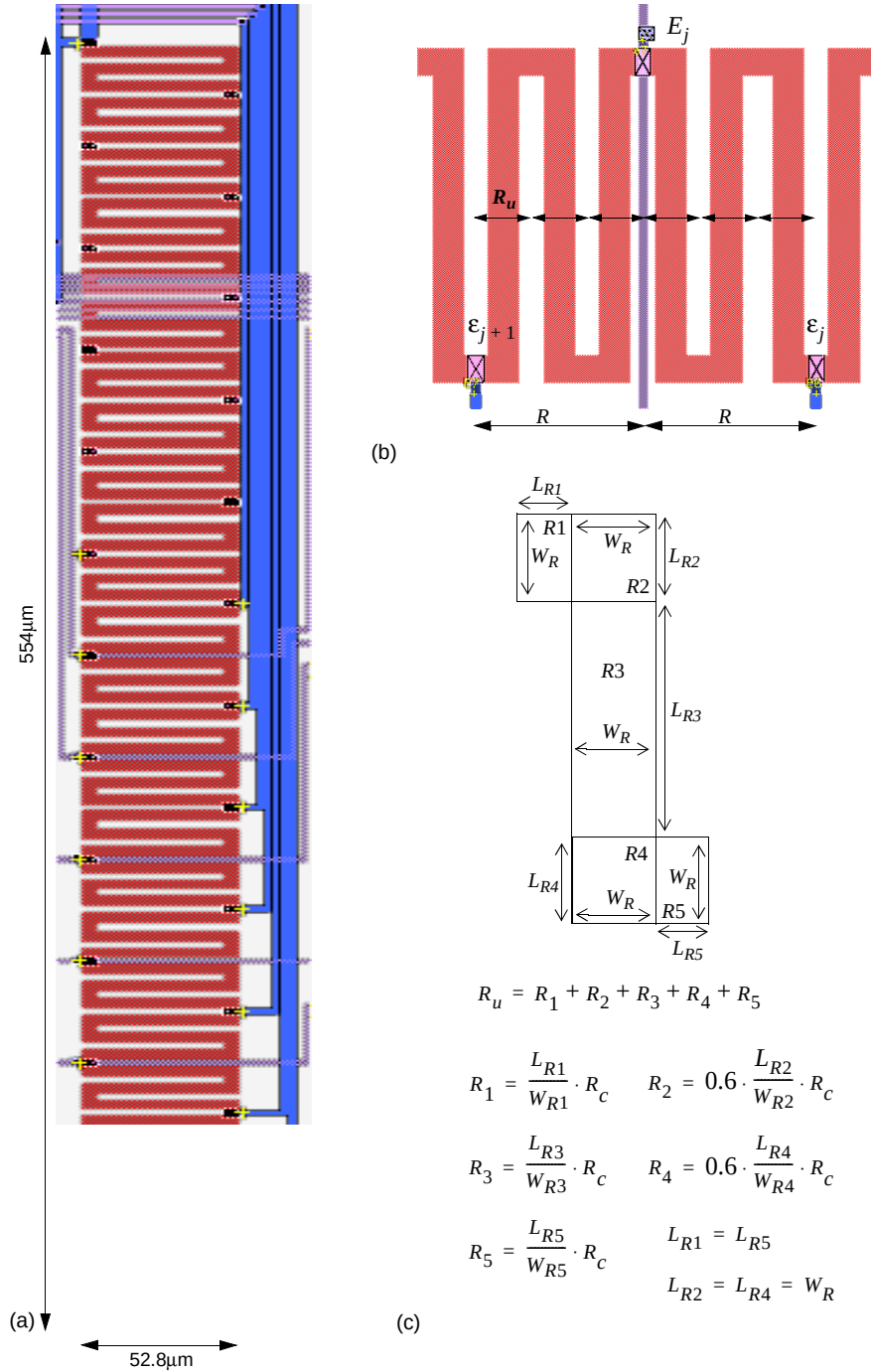


Fig.4.4 Layout del array resistivo (a); detalle de dos resistores  $R$  (b); diseño de  $R$  a partir de tramos de resistencia  $R_u$  (c).

mayor resistividad por cuadrado unidad de este material. La Fig.4.4(a) muestra un detalle del *layout* del *array* resistivo empleado para generar los valores  $\epsilon_j$  y  $E_j$ , en el circuito integrado realizado como ejemplo en esta tesis.

La estrategia empleada para su diseño se ilustra también en la Fig.4.4. La Fig.4.4(b) muestra un detalle del *array* resistivo en el que aparecen dos resistores unitarios  $R$  completos, cada uno de los cuales se obtiene enlazando convenientemente tres tramos de polisilicio como el mostrado en la Fig.4.4(c) de resistencia  $R_u$ , cuyo valor se obtiene sumando la contribución de las diferentes subregiones, según las expresiones que se muestran en dicha figura. El valor de la resistencia total del *array*,  $R_T$ , se obtiene a partir de la contribución de  $n_R$  resistencias de valor  $R$ .

El valor de  $R_T$  se elige de forma que  $R_T = \frac{V_{DD}^2}{P_R}$ , donde  $V_{DD}$  es la tensión de alimentación y  $P_R$  es la potencia que va a consumir el *array* resistivo. Fijada  $P_R < P_{Rmax}$ , como opción de diseño, donde  $P_{Rmax}$  es la potencia máxima que un *array* resistivo puede disipar según la tecnología empleada, queda también fijada la corriente  $I_R$  que va a circular por  $R_T$ .

El valor de  $R$  se elige ahora de forma que la tensión que caiga en ella,  $V_R$ , cumpla la relación  $V_R = I_R \cdot R = h$  siendo  $h$ , el módulo de la partición definido con anterioridad. Teniendo esto en cuenta los valores  $\epsilon_j$  y  $E_j$  para  $(1 \leq j \leq L-1)$  pueden obtenerse según las expresiones

$$\begin{aligned}\epsilon_j &= (b + 2(j-1)) \cdot R \cdot I_R \\ E_j &= (b + 2j-1) \cdot R \cdot I_R\end{aligned}\tag{4.2}$$

donde  $b$  es un factor que determina el número de resistencias  $R$  necesarias para fijar el valor de tensión correspondiente al límite inferior del espacio de entrada sobre el que se ha definido la partición (ver Fig.4.3, en ella  $a$  es un factor que indica el número de resistencias  $R$  necesarias hasta completar  $R_T$ ). Se tiene por tanto, que el número de resistencias  $n_R$  que componen  $R_T$  ha de ser

$$n_R = \frac{R_T}{R} = a + b + 2(L-1) .\tag{4.3}$$

Finalmente cabe destacar que el diseño del *array* resistivo completo a partir de resistencias unitarias  $R$  busca garantizar la uniformidad del valor de tensión  $h$  que define el módulo de la partición en rejilla, frente a variaciones en el valor de  $R_c$  debidas al proceso de fabricación, dado que de esta manera cada uno de los tramos resistivos está construido a partir de resistores idénticos desde el punto de vistas de su geometría.

Para el controlador ejemplo MFCON que ha sido fabricado, se tiene una partición en rejilla bidimensional, como la que se muestra en la Fig.4.5(a), resultado de considerar ocho etiquetas,  $X_{ij}$  para  $(1 \leq j \leq 8)$  por entrada y por tanto con sesenta y cuatro reglas. Para

determinar esta partición se necesita especificar, por una parte el rango de valores de tensión que delimitan el universo de discurso unidimensional; y por otra el módulo  $h$  de dicha partición. El rango del universo de discurso se ha establecido en base a consideraciones de diseño sobre el rango de entrada de los circuitos que realizan las funciones de pertenencia, según se ha discutido en el apartado 2.3.1. del Capítulo 3, y resulta estar comprendido en el intervalo  $[2,4.8]$  volts. Dado este rango y el número de etiquetas considerado se tiene que el valor del módulo de la partición  $h$  ha de ser 0.4 volts. A partir de estos valores y las expresiones de (4.2) es posible calcular los valores de tensión  $E_i$  para  $(1 \leq i \leq 7)$  que determinan la partición, y que son necesarios para programar el antecedente de reglas, así como los valores de tensión  $\varepsilon_i$  para  $(1 \leq i \leq 7)$  que delimitan los intervalos de interpolación. La Fig.4.5(b), muestra la distribución de estos valores sobre el universo de discurso de una entrada genérica  $x_i$ .

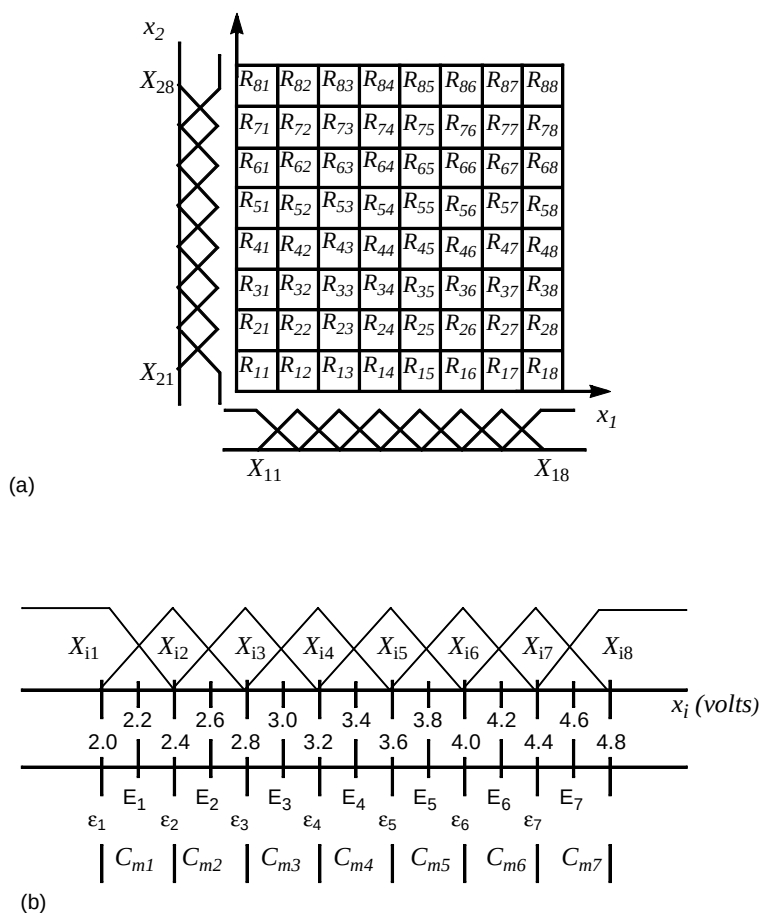


Fig.4.5 Partición en rejilla para el controlador ejemplo (a); valores que determinan la partición en rejilla en el espacio de entrada unidimensional (b).

En la Tabla 4.1 y en la Tabla 4.2 se recogen respectivamente, los valores nominales de los puntos que definen la partición en rejilla del espacio de entrada  $\varepsilon_i$  para  $(1 \leq i \leq 7)$ , así como los valores de los parámetros de programación de los antecedentes de regla del *Núcleo Analógico*  $E_i$  para  $(1 \leq i \leq 7)$ , por su parte en la Tabla 4.3 se recogen los valores nominales de los parámetros empleados en el diseño del array resistivo en el controlador ejemplo.

**Tabla 4.1:** Valores de tensión nominales que definen la partición en rejilla del espacio de entrada en el controlador ejemplo.

| $\varepsilon_1(\text{Volt})$ | $\varepsilon_2(\text{Volt})$ | $\varepsilon_3(\text{Volt})$ | $\varepsilon_4(\text{Volt})$ | $\varepsilon_5(\text{Volt})$ | $\varepsilon_6(\text{Volt})$ | $\varepsilon_7(\text{Volt})$ |
|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| 2.00                         | 2.40                         | 2.80                         | 3.20                         | 3.60                         | 4.00                         | 4.40                         |

**Tabla 4.2:** Valores de tensión nominales de los parámetros de programación del bloque de antecedentes de regla en el controlador ejemplo.

| $E_1(\text{Volt})$ | $E_2(\text{Volt})$ | $E_3(\text{Volt})$ | $E_4(\text{Volt})$ | $E_5(\text{Volt})$ | $E_6(\text{Volt})$ | $E_7(\text{Volt})$ |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| 2.20               | 2.60               | 3.00               | 3.40               | 3.80               | 4.20               | 4.60               |

**Tabla 4.3:** Valores nominales empleados en el diseño del array resistivo en controlador ejemplo.

| $V_{DD}(\text{Volt})$ | $P_R(\text{mW})$      | $R_T(\text{Kohm})$    | $R(\text{Kohm})$      | $V_R(\text{Volt})$    | $n_R$                 | $h(\text{Volt})$     |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------|
| 5.0                   | 1                     | 25                    | 1.0                   | 0.2                   | 25                    | 0.4                  |
| $W_R(\mu\text{m})$    | $L_{R1}(\mu\text{m})$ | $L_{R2}(\mu\text{m})$ | $L_{R3}(\mu\text{m})$ | $L_{R4}(\mu\text{m})$ | $L_{R5}(\mu\text{m})$ | $R_c(\text{ohm/sq})$ |
| 4                     | 1.6                   | 4                     | 40.8                  | 4                     | 1.6                   | 28                   |

### 2.3.2. Comparador Analógico con Histéresis

Los comparadores son bloques constructivos básicos de los convertidores analógico-digital, y por tanto para toda una nueva generación de circuitos integrados de señal mixta. También son ampliamente utilizados en aplicaciones tan diversas como las de generación de señales y funciones, las comunicaciones digitales, o las redes neuronales, entre otras. De ahí su importancia y la amplia variedad de diseños que pueden encontrarse. Un detallado resumen de las principales características y propiedades de estos circuitos puede encontrarse en [Rodr99a].

En general los comparadores analógicos son usados para detectar el signo de la diferencia entre dos señales analógicas,  $x_+(t)$  y  $x_-(t)$ , y proporcionar una salida digital,  $y_{out}$ , que codifica el resultado de dicha operación. Esta funcionalidad puede ser expresada como

$$y_{out} = \begin{cases} \geq V_{OH} & \text{sí} & x_+(t) > x_-(t) \\ \leq V_{OL} & \text{sí} & x_+(t) < x_-(t) \end{cases} \quad (4.4)$$



donde  $V_{OH}$  y  $V_{OL}$ , son niveles de tensión asociados al 1-lógico y al 0-lógico respectivamente, de forma que se garantice una correcta interpretación de la salida  $y$  como señal digital. En muchas ocasiones interesa expresar esta relación en función de una sola variable de entrada, y se toma la variable diferencial  $v_d = x_+(t) - x_-(t)$  de forma que (4.4) puede escribirse ahora como

$$y_{out} = \begin{cases} \geq V_{OH} & \text{sí} & v_d > 0 \\ \leq V_{OL} & \text{sí} & v_d < 0 \end{cases} \quad (4.5)$$

En esta aplicación el comparador es empleado como bloque constructivo de un conversor A/D, por lo que una de las entradas, por ejemplo  $x_-(t)$ , es un valor constante de referencia  $V_{REF}$ , y la función del comparador es detectar si la otra señal  $v_{IN}$ , aplicada como  $x_+(t)$ , es mayor o menor que dicha referencia. La Fig.4.6(a) muestra el símbolo más usual para representar a un comparador, mientras que la Fig.4.6(b) muestra la curva de transferencia del comparador ideal que expresa (4.4), donde  $V_{SH} > V_{OH}$  y  $V_{SL} > V_{OL}$  representan los niveles de saturación de la salida  $y$ , renombrada como  $v_{OUT}$  en dicha figura. Un comparador ideal se caracteriza por presentar una ganancia en tensión  $k_S$  infinitamente grande, una tensión *offset* de entrada  $V_{off}$  nula, un tiempo de respuesta  $T_D$  también nulo, además de permitir un rango de entrada en modo común CMR ilimitado, poseer una impedancia de entrada  $Z_{in}$  infinitamente grande y una ilimitada capacidad de conducción en su nudo de salida.

Cualquier comparador real presenta desviaciones respecto a este comportamiento que se refleja en los parámetros antes citados, y cuyo valor ha de ser tenido en cuenta en el proceso de diseño para cumplir con los requerimientos de cada aplicación en concreto. La Fig.4.6(c) muestra esquemáticamente la curva de transferencia de un comparador real, sobre la que se ilustran algunos de estos parámetros. Los principales factores de caracterización que han de

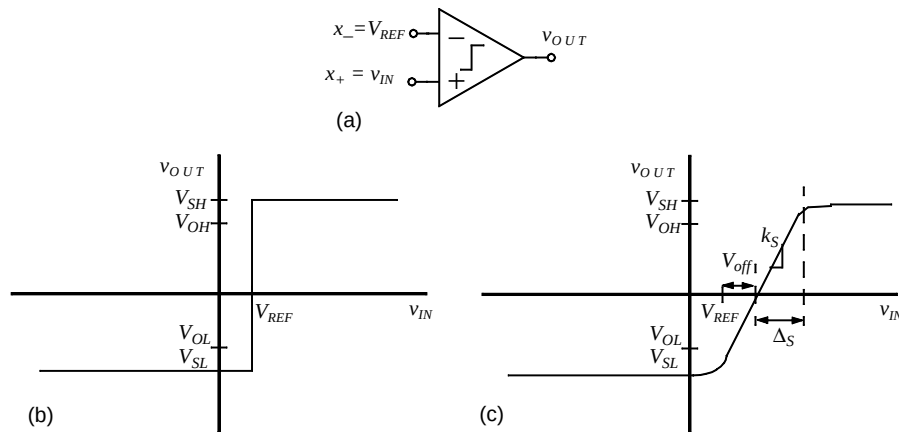


**Fig.4.6 Símbolo y curvas de trasferencia ideales: Comparador : Símbolo (a); curva de transferencia ideal (b); curva de transferencia real (c)**

ser tenidos en cuenta para este diseño, son la resolución estática y el tiempo de respuesta.

La resolución del comparador en condiciones estáticas es un factor que depende de dos parámetros: Por una parte, la de tensión *offset* de entrada  $V_{off}$  y por otra de la sensibilidad incremental  $\Delta_S$ , definida como el incremento (decremento) mínimo de la tensión de entrada, respecto a la tensión de referencia  $V_{ref}$  (en caso de  $V_{off}$  nulo), necesario para provocar la correspondiente transición en la salida del comparador; que a su vez depende en relación inversa con la ganancia en tensión  $k_S$ , Fig.4.6(c). En función de estos parámetros la resolución se expresa

$$\xi_S = |V_{off}| + \Delta_S \quad (4.6)$$

donde el valor absoluto expresa el carácter esencialmente aleatorio del factor  $V_{off}$  que, en general, es debido a factores del proceso fabricación y desapareamiento entre transistores. Para comparadores con una elevada ganancia  $k_S > 10^3$ , la resolución coincide prácticamente con el valor de  $V_{off}$ . Para cualquier valor de entrada comprendido en el intervalo  $[-\xi, \xi]$  la salida del comparador no está definida. Por su parte, cualquier valor de entrada externo a dicho intervalo es denominado *overdrive* y hace que la salida del comparador alcance uno de sus dos estados.

El tiempo de respuesta  $T_D$  mide el intervalo de tiempo que transcurre entre el instante en el que la señal de entrada transita por el nivel de referencia y el instante para el que la salida alcanza el correspondiente nivel lógico de salida (habitualmente medido cuando la señal de salida alcanza el 90% de  $V_{OH}$  o  $V_{OL}$  según el sentido de la transición). La Fig.4.7 ilustra este parámetro, cuyo valor, en general, depende del valor final alcanzado por la tensión de entrada, esto es de la amplitud del *overdrive*, de forma que a mayor amplitud, menor es el tiempo de

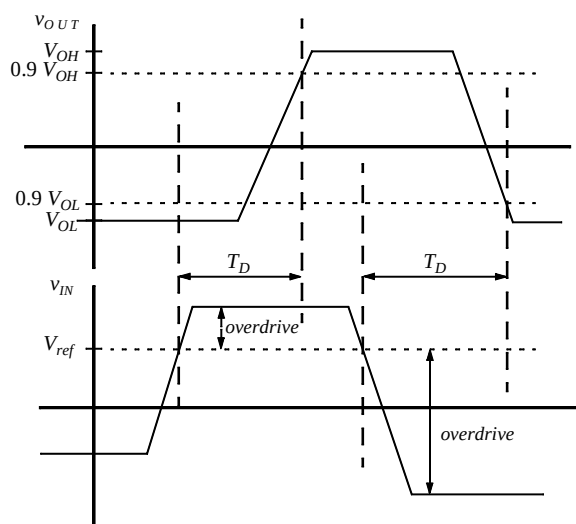


Fig.4.7 Ilustración del tiempo de respuesta  $T_D$  de un comparador real.

respuesta, aunque existe un límite a partir del cual no es posible obtener un tiempo de respuesta inferior.

Los comparadores pueden ser diseñados para trabajar tanto en tiempo continuo, como en tiempo discreto. Para esta aplicación, es clara la elección de un comparador en tiempo continuo.

Finalmente, y dada la inmunidad al ruido requerida en esta aplicación en la identificación de los intervalos de interpolación, el comparador que se propone para este diseño ha de presentar la propiedad de la histéresis, que en general se relaciona con sistemas que poseen cierta capacidad de memoria.

En este contexto la histéresis es una cualidad del comparador por la cual el valor umbral a la entrada, esto es, el nivel que define cuándo ha de producirse un cambio de estado en la salida del comparador, es función del propio estado de la salida. Para ilustrar este concepto, la Fig.4.8 muestra esquemáticamente las curvas de transferencia entrada-salida de dos comparadores ideales, con y sin la propiedad de histéresis. Para el comparador ideal sin histéresis de la Fig.4.8(a),  $V_{REF}$  es el nivel de tensión umbral, como muestra su curva de transferencia. Para todo valor de entrada  $v_{IN} > V_{REF}$  la salida toma el valor  $v_{OUT} = V_{SH}$ , mientras que para  $v_{IN} < V_{REF}$ ,  $v_{OUT} = V_{SL}$ . Sin embargo, para un comparador con histéresis, como se muestra en la Fig.4.8(b) se tienen dos puntos de transición o umbral,  $V_{TH-}$  y  $V_{TH+}$ , los cuales actúan como tales según la dirección de la transición de la salida. Esto es, si suponemos que inicialmente la salida del comparador es tal que  $v_{OUT} = V_{SL}$ , y la entrada  $v_{IN} < V_{TH-}$ ; y hacemos que  $v_{IN}$  empiece a crecer, la salida no transita hasta  $v_{OUT} = V_{SH}$  hasta que  $v_{IN}$  no haya superado

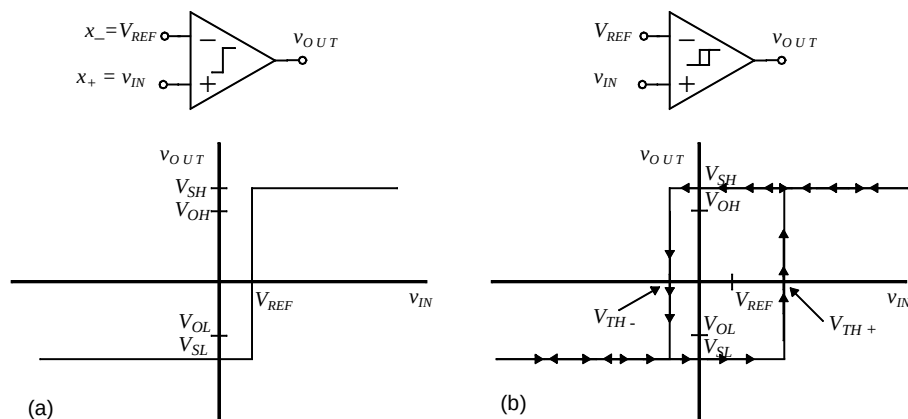


Fig.4.8 Símbolo y curvas de transferencia ideales: Comparador ideal sin histéresis (a); comparador ideal con histéresis (b)

el umbral  $V_{TH+}$ ; si a continuación  $v_{IN}$  comienza a decrecer, la transición de la salida  $v_{OUT}$  al valor  $V_{SL}$  no se producirá ahora hasta que  $v_{IN}$  no haya descendido por debajo del umbral  $V_{TH-}$ .

Esta propiedad puede ser aprovechada cuando el comparador ha de trabajar en entornos ruidosos, en los cuales deba detectar transiciones en señales próximas al nivel de referencia. En estas circunstancias si el comparador es suficientemente rápido, y la amplitud del ruido lo suficientemente elevada, la salida del comparador también resultará ruidosa en el sentido de que aparecerán transiciones no deseadas, es decir no provocadas por transiciones reales en la señal que se quiere detectar. Un comparador con un intervalo de histéresis mayor o igual a la amplitud de la señal de ruido, actuará de filtro, evitando las transiciones no deseadas.

La Fig.4.9 ilustra esta circunstancia representando el comportamiento ideal de los dos comparadores de la Fig.4.8, para el caso de una entrada diferencial  $v_d = v_{IN} - V_{REF}$ , a la que se le superpone cierto nivel de ruido. En ella se observa como en el caso del comparador sin histéresis, Fig.4.9(a), la salida  $v_{OUT}$  oscila entorno al punto de referencia antes de alcanzar su estado final. Esta situación desaparece en el caso del comparador con histéresis mostrado en la Fig.4.9(b).

Hay muchas maneras de dotar de histéresis a un comparador. Todas ellas se basan en el empleo de cierta cantidad de realimentación positiva [Rodr99a][Alle87]. La Fig.4.10(b) muestra un esquema conceptual del comparador con histéresis multietapa empleado en este diseño. La primera etapa esta constituida por un amplificador diferencial, que permite fijar el

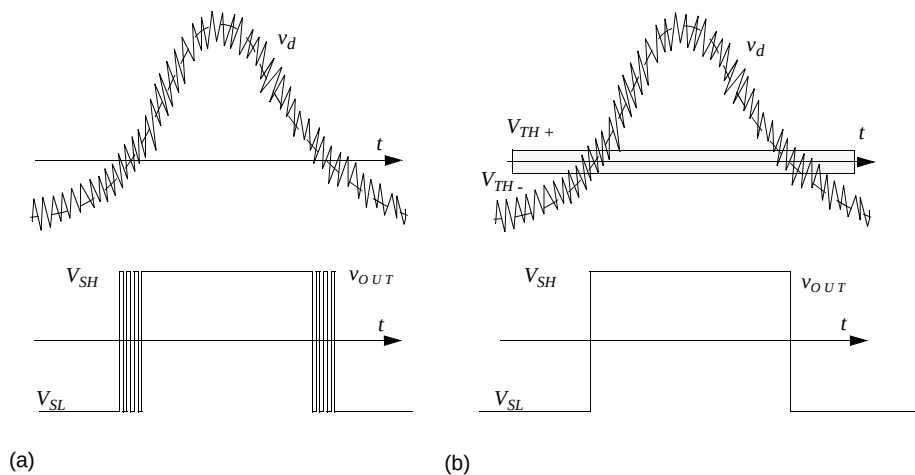


**Fig.4.9 Respuesta de los comparadores a entradas con ruido: comparador ideal sin histéresis (a); comparador ideal con histéresis (b).**

punto de referencia del comparador, proporcionando una alta impedancia de entrada, de forma que  $V_{REF}$  pueda ser obtenida directamente del *array* resistivo.

La segunda etapa proporciona la realimentación positiva que produce la histéresis, cuya magnitud es controlada mediante las fuentes de corriente  $I_{PH}$ . Esta etapa está constituida por un inversor lógico que actúa como comparador y etapa de ganancia, y dos llaves de corriente complementarias PMOS y NMOS, controladas por la salida  $v_{AO}$  del inversor, de forma que la fuente  $I_{PH}$  aporta o retira corriente del nudo de salida  $v_A$  de la primera etapa haciendo que el umbral de transición del comparador se desplace a  $V_{TH-}$  o  $V_{TH+}$ .

La tercera etapa es la etapa de salida, la cual genera el terminal de salida  $v_{OUT}$ , y proporciona una ganancia adicional, que acelera la respuesta del sistema, y un aislamiento del nodo de salida  $v_{OA}$  de la etapa de histéresis, respecto a la circuitería digital posterior.

A continuación se analiza con más detalle algunos aspectos del diseño de cada una de las etapas que lo constituyen.

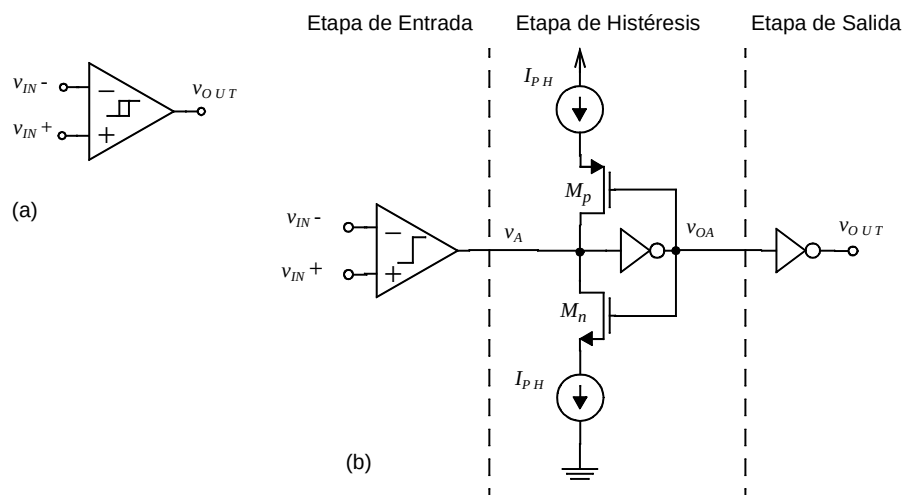


**Fig.4.10 Comparador con histéresis: Símbolo (a); esquema conceptual del comparador multietapa con histéresis empleado en este diseño (b).**

## 2.3.2.1. Circuito para la etapa de entrada.

La Fig.4.11(a) muestra el esquema del circuito que se ha empleado para realizar la etapa de entrada del comparador según el esquema de la Fig.4.10(b). Dicho circuito es una de las diferentes posibilidades de realizar la arquitectura más simple posible para este tipo de sistemas siguiendo el modelo de comportamiento de primer orden que se muestra en la Fig.4.11(c) [Lake94]. Este modelo consiste en la conexión de un transconductor no lineal  $G_m(v_d)$ , realizado mediante un par diferencial, (sombreado en la Fig.4.11(a)) y un resistor no lineal  $G_o(v_A)$  (resto de circuitería en la Fig.4.11(a)), además de un capacitor  $C_o$  que representa a la carga y al conjunto de capacidades parásitas y modela la respuesta dinámica. La Fig.4.11(d) ilustra la curva de transferencia del sistema que aproxima la curva ideal. Para pequeñas variaciones de  $v_d$  en torno al punto de equilibrio  $v_d = 0$ , la salida  $v_A$  se obtiene como  $v_A = V_A + k_s \cdot v_d$ , donde  $V_A$  es el valor de  $v_A$  en el equilibrio, mientras que  $k_s$ , (la pendiente de dicha curva en la zona central) es la ganancia diferencial en tensión, que se obtiene como el

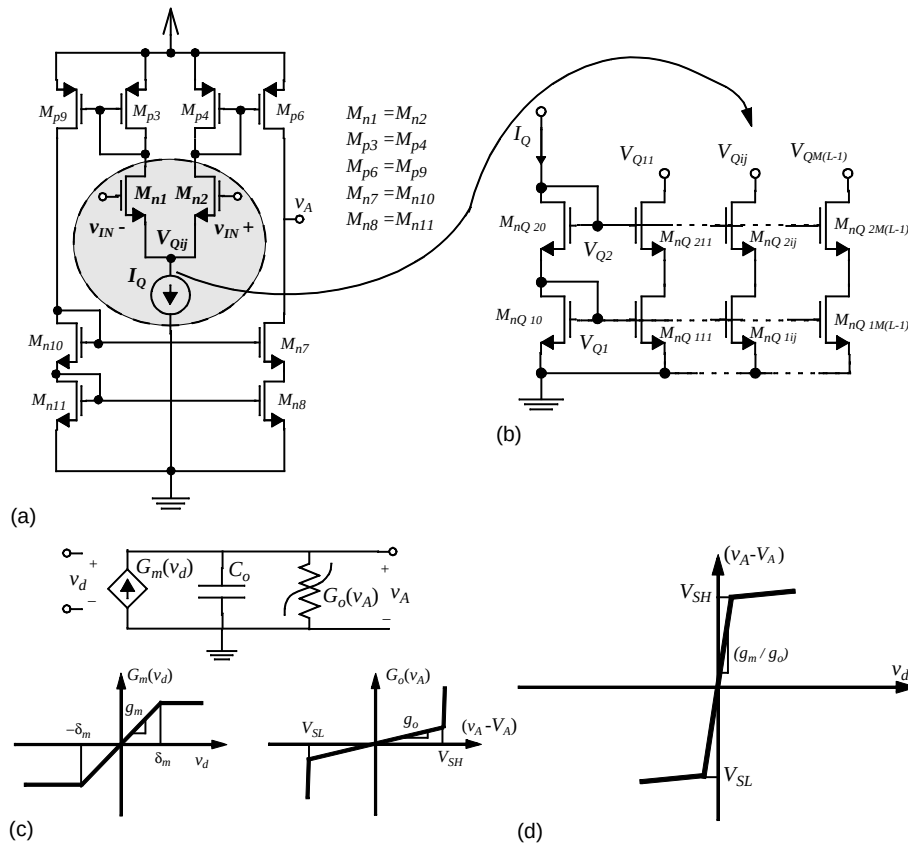


Fig.4.11 Etapa de entrada: esquema del circuito (a); circuito para la generación de la fuente de polarización  $I_Q$  (b); modelo de comportamiento de primer orden (c); curva de transferencia (d).

producto de la transconductancia  $g_m$  por la resistencia de salida  $r_o = g_o^{-1}$ , esto es  $k_s = g_m r_o = \frac{g_m}{g_o}$ . Para valores grandes de  $v_d$ , la salida  $v_A$  se satura como muestra la Fig.4.11(d). Cuanto mayor sea el valor de  $k_s$ , mayor será la semejanza de dicha curva con la del comportamiento del comparador ideal. Sin embargo  $k_s$  no puede ser escogido arbitrariamente grande, sin degradar otras características igualmente importantes, como son el tiempo de respuesta, o el rango en modo común, CMR.

**Tabla 4.4:** Parámetros del modelo del circuito de la etapa de entrada.

| Parámetro  | Expresión                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $g_m$      | $\sqrt{2\beta_{0n}\left(\frac{W}{L}\right)_{M_{n1}} I_Q}$                                                                                                                                                                                                                                                                                                                                                                     |
| $\delta_m$ | $\sqrt{\frac{I_Q}{2\beta_{0n}\left(\frac{W}{L}\right)_{M_{n1}}}}$                                                                                                                                                                                                                                                                                                                                                             |
| $g_o$      | $I_a \left( \frac{\lambda_{M_{n7}}^2}{2\lambda_{M_{n7}} + \sqrt{\frac{2\beta_{0n}\left(\frac{W}{L}\right)_{M_{n7}}}{I_a} (1 + \delta_{M_{n7}})}} + \lambda_{M_{p6}} \right)$ <p style="text-align: center;">con <math>I_a = \frac{\left(\frac{W}{L}\right)_{M_{p6}}}{\left(\frac{W}{L}\right)_{M_{n4}}} \frac{I_Q}{2}</math></p>                                                                                              |
| $C_o$      | $C_L + W_{M_{n7}} C_{GD0M_{n7}} + W_{M_{p6}} C_{GD0M_{p6}}$                                                                                                                                                                                                                                                                                                                                                                   |
| CMR        | $V_{in(max)} = V_{DD} - \sqrt{\frac{I_Q}{2\beta_{0n}\left(\frac{W}{L}\right)_{M_{n3}}}} - V_{T_{0p}(max)} + V_{T_{M_{n1}}(min)}$ $V_{in(min)} = \sqrt{\frac{I_Q}{2\beta_{0n}\left(\frac{W}{L}\right)_{M_{n1}}}} + \sqrt{\frac{I_Q}{\beta_{0n}\left(\frac{W}{L}\right)_{M_{nQ20}}}} + \sqrt{\frac{I_Q}{\beta_{0n}\left(\frac{W}{L}\right)_{M_{nQ10}}}} +$ $+ V_{T_{0n}(max)} + V_{T_{M_{nQ20}}(max)} - V_{T_{M_{nQ2ij}}(min)}$ |

En la Tabla 4.4 se resumen las expresiones de primer orden de los principales parámetros de diseño que caracterizan el modelo de comportamiento del circuito de la Fig.4.11(a), en función del tamaño de los transistores. En ella además de las expresiones para  $g_m$ ,  $g_o$ ,  $C_o$ , se incluyen las de el parámetro  $\delta_m$ , ancho de la zona de comportamiento lineal para el transconductor  $G_m(V_d)$ , y las de su CMR, rango de entrada en modo común. Estas expresiones se obtienen teniendo en cuenta un perfecto apareamiento de los transistores que forma en circuito según aparece en la Fig.4.11(a). De ellas se deduce que la ganancia puede ser controlada escogiendo apropiadamente el valor de la fuente de polarización  $I_Q$  y los tamaños de los transistores  $M_{n1}$ ,  $M_{n3}$ ,  $M_{n6}$  y  $M_{n7}$ , y teniendo en cuenta que una elección que lleva a incrementar la ganancia  $k_s$ , implica simultáneamente un incremento en  $C_o$  y una disminución en CMR; y viceversa.

Finalmente en la Fig.4.11(b) se muestra el espejo de corriente cascode empleado para replicar la fuente de intensidad de polarización  $I_Q$  de la Fig.4.11(a) para las etapas de entrada de todos los comparadores del bloque *Selector de Intervalo*. Nótese la conexión a través de los nudos nombrados  $V_{Qij}$ . En el controlador ejemplo que ha sido fabricado, se ha escogido para  $I_Q$  un valor de  $40\mu A$ , con lo que, teniendo en cuenta los tamaños de los transistores de los circuitos de la Fig.4.11 empleados en este diseño, y que se resumen en la Tabla 4.5; y los valores de los parámetros de la tecnología empleada (ver Apéndice), se obtienen los valores que se recogen en la Tabla 4.6 para los parámetros del comparador descritos en la Tabla 4.4.

**Tabla 4.5:** Tamaño de los transistores empleados en el diseño de los circuitos que se muestran en la Fig.4.11(a) y la Fig.4.11(b) en el controlador ejemplo.

| $M_{n1}=M_{n2} (W/L)$  | $M_{p3}=M_{p4} (W/L)$ | $M_{p6}=M_{p9} (W/L)$ | $M_{n7}=M_{n10} (W/L)$ |
|------------------------|-----------------------|-----------------------|------------------------|
| 150 $\mu m$ /2 $\mu m$ | 30 $\mu m$ /2 $\mu m$ | 60 $\mu m$ /2 $\mu m$ | 20 $\mu m$ /2 $\mu m$  |
| $M_{n8}=M_{n11} (W/L)$ | $M_{nQ2ij} (W/L)$     | $M_{nQ1ij} (W/L)$     |                        |
| 20 $\mu m$ /2 $\mu m$  | 10 $\mu m$ /2 $\mu m$ | 20 $\mu m$ /5 $\mu m$ |                        |

**Tabla 4.6:** Valores de los parámetros que definen el modelo de la etapa de entrada del comparador mostrado en la Fig.4.11(c), calculados a partir de las expresiones de la Tabla 4, los valores de la Tabla 5 y los parámetros de la tecnología para el controlador ejemplo.

| $g_m (\mu A V^{-1})$ | $\delta_m (mV)$                          | $g_o (\mu A V^{-1})$    |
|----------------------|------------------------------------------|-------------------------|
| 612.3                | 65.3                                     | 2.0                     |
| $C_o (pF)$           | CMR<br>$V_{in(max)}/V_{in(min)} (Volts)$ | $k_s = \frac{g_m}{g_o}$ |
| $C_L + 0.015$        | 4.45/1.67                                | 306.15                  |



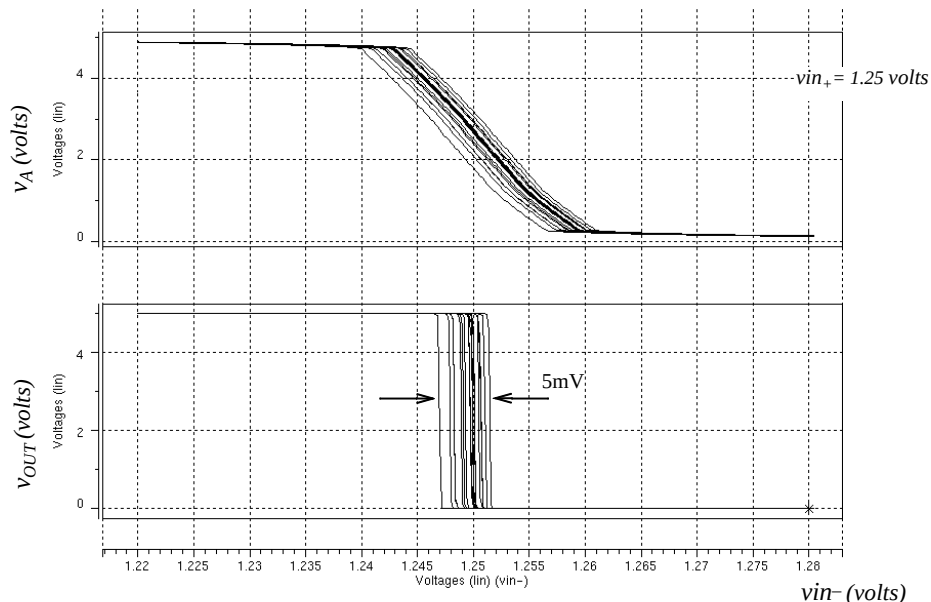


**Fig.4.12** Curva de transferencia de la etapa de entrada del comparador obtenida con HSPICE mediante un análisis de montecarlo de 20 curvas

El tamaño de los transistores en el controlador ejemplo se ha escogido suficientemente grande, y al dictado de la experiencia de diseño y las simulaciones efectuadas, para minimizar la influencia del desapareamiento entre transistores en las prestaciones del circuito, sin entrar en los detalle de caracterización de esa influencia, por considerar que no es especialmente crítico para el propósito de este diseño. A modo de ilustración, la Fig.4.12 muestra la curva de transferencia de esta etapa obtenida mediante simulación con HSPICE efectuando un análisis de montecarlo para 20 curvas. En esta simulación se ha fijado  $v_{in+}$  a 1.25 volts, mientras que para  $v_{in-}$  se ha efectuado un barrido entre 1.22volts y 1.28 volts. La pendiente medida en el punto medio de la curva de transferencia de  $v_A$  (ver Fig.4.17) coincide prácticamente con el valor para la ganancia  $k_s = \frac{g_m}{g_o}$  calculado que muestra la Tabla 4.6. Por su parte, la desviación de las 20 curvas no supera 2.5 mVolts en valor absoluto, en torno al valor de referencia. Este valor puede ser tomado a efectos prácticos como la tensión de *off set* del comparador.

### 2.3.2.2. Circuito para las etapas de histéresis y de salida.

La Fig.4.13(a) muestra el circuito que realiza las etapas de histéresis y de salida del comparador, correspondientes al diagrama de la Fig.4.10(b), salvo las fuentes de corriente  $I_{PH}$  que se generan replicando una fuente de polarización externa  $I_{POLH}$  con un doble sistema de espejos de corriente según se muestra en la Fig.4.13(b), en la que aparece el circuito completo utilizado para proporcionar la corriente de polarización de todas las etapas de salida de todos los comparadores empleados en el diseño de este bloque.

La realimentación positiva que proporciona la histéresis, y que es llevada a cabo por los transistores  $M_p$  y  $M_n$ , y las fuentes  $I_{PH}$  en la Fig.4.13(a), puede ser modelada en primer orden mediante un resistor no lineal  $G_{mo}(v_A)$ , cuya curva característica corriente-tensión se muestra en la Fig.4.14(a). En esa figura  $V_{TRINV}$  representa al valor de tensión umbral que sirve de referencia para las transiciones del comparador que constituye en este caso el inversor CMOS formado por los transistores  $M_{p1}$  y  $M_{n1}$  en la Fig.4.13(a), (ver Tabla 4.7.)

Para valores de  $v_A$  tales que  $v_A - V_{TRINV} \leq 0$  el transistor  $M_n$  conduce en su región óhmica, mientras que  $M_p$  entra en su región de corte, de forma que el resistor se comporta prácticamente como una fuente de corriente de valor  $I_{PHn}$ , que retira corriente al nudo  $v_A$ . La Fig.4.14(b) muestra dos circuitos equivalentes en esta región, donde  $r_{DSnON}$  modela la pequeña resistencia de conducción del transistor  $M_n$ , mientras que  $r_{onPH}$  modela la resistencia de salida del circuito que realiza la fuente  $I_{PH}$  con transistores de tipo **n**.

Para valores de  $v_A$  tales que  $v_A - V_{TRINV} \geq 0$  el transistor  $M_p$  conduce en su región óhmica, mientras que el transistor  $M_n$  entra en su región de corte, de forma que el resistor se comporta como una fuente de corriente de valor  $-I_{PHp}$ , que aporta corriente al nudo  $v_A$ . La Fig.4.14(c) muestra dos circuitos equivalentes en esta región, donde  $r_{DSpON}$  modela la pequeña

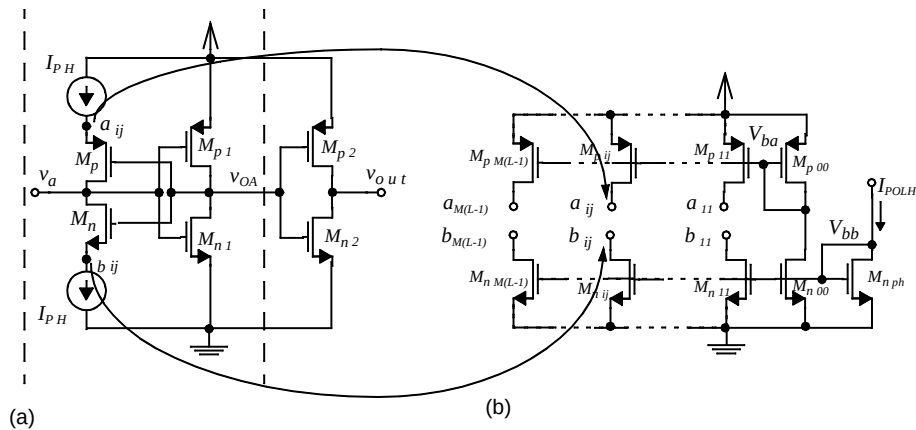


**Fig.4.13** Esquema del circuito que realiza las etapas de histéresis y de salida del comparador (a); circuito que realiza la fuente de polarización de la etapa de histéresis (b)

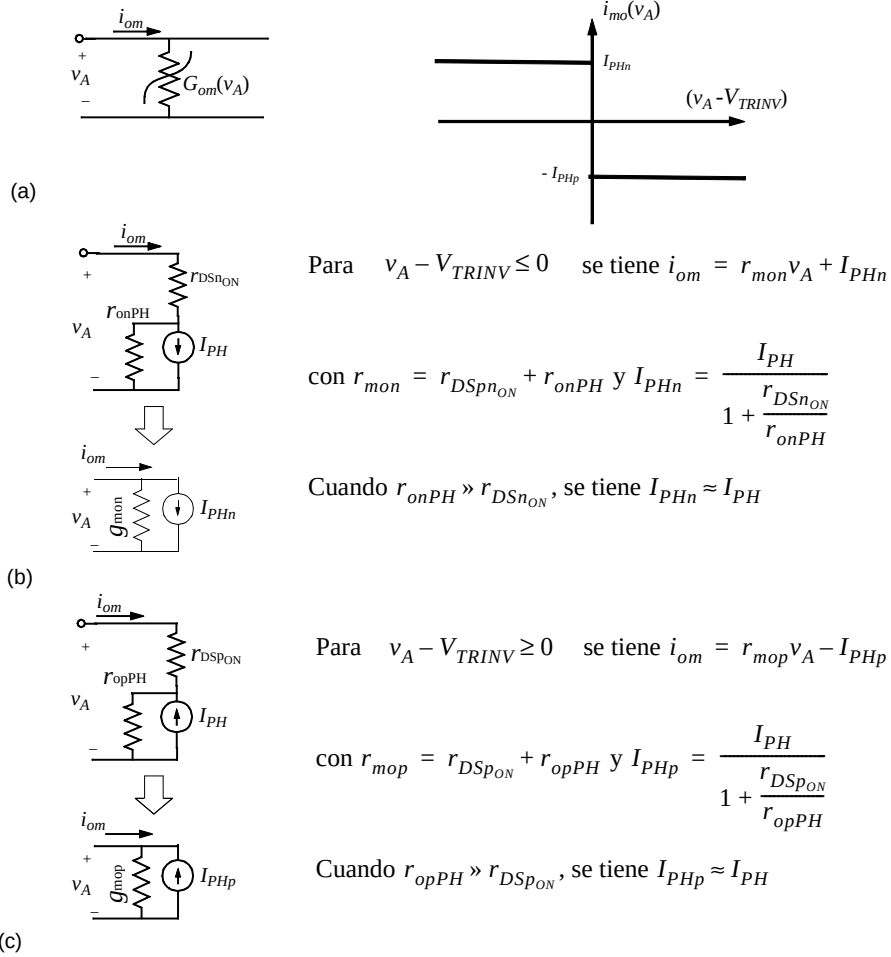


**Fig.4.14** Curvas corriente-tensión del resistor no lineal que modela la etapa de histéresis (a); circuito y expresiones del modelo en la región negativa de la curva (b); circuito y expresiones del modelo en la región positiva de la curva (c).

resistencia de conducción del transistor  $M_p$ , mientras que  $r_{opPH}$  modela la resistencia de salida del circuito que genera la fuente  $I_{PH}$  con transistores de tipo **p**.

Para obtener los valores de tensión  $V_{TH}^-$  y  $V_{TH}^+$  que determinan el ancho de la región de histéresis consideraremos en conjunto los modelos de los circuitos de la etapa de entrada y de histéresis, como muestra la Fig.4.15, para las dos situaciones que se corresponden con los dos tipos de transiciones posibles y analizaremos su comportamiento.

En primer lugar consideremos la situación en la que  $v_A - V_{TRINV} \leq 0$ , con  $v_d < 0$  pero próximo al punto de equilibrio. En este caso la etapa de entrada estará trabajando en su zona lineal, por lo que el sistema formado por esta etapa y la de histéresis puede ser modelado como

se muestra en la Fig.4.15(a). En esta situación se tiene que  $v_{OA} \geq V_{OH}$  y además

$$v_A = V_A + \frac{g_m v_d - I_{PHn}}{g_o + g_{mon}} \quad (4.7)$$

Esta expresión es válida incluso cuando  $v_d > 0$  siempre que la etapa de entrada siga trabajando en su zona lineal. Sin embargo en esta situación no se producirá la transición en la salida  $v_{OA}$  a  $v_{OA} \leq V_{OL}$  dado que para esto ocurra se tiene que cumplir que  $v_A \geq V_{TRINV}$ , pero para que esta situación se produzca  $v_d$  ha de tomar valores tales que

$$v_d \geq \frac{(V_{TRINV} - V_A)(g_o + g_{mon})}{g_m} + \frac{I_{PHn}}{g_m} \quad (4.8)$$

y por tanto  $v_{in} \geq V_{REF} + V_{TH}^+$  de donde se tiene que

$$V_{TH}^+ = \frac{(V_{TRINV} - V_A)(g_o + g_{mon})}{g_m} + \frac{I_{PHn}}{g_m} \quad (4.9)$$

Consideremos a continuación la situación inicial en la que  $v_A - V_{TRINV} \geq 0$ , con  $v_d > 0$  pero próximo al punto de equilibrio. En este caso la etapa de entrada estará trabajando en su zona lineal, por lo que el sistema formado por esta etapa y la de histéresis puede ser modelado como se muestra en la Fig.4.15(b). En esta situación se tiene que  $v_{OA} \leq V_{OL}$  y además

$$v_A = V_A + \frac{g_m v_d + I_{PHp}}{g_o + g_{mop}} \quad (4.10)$$

Esta expresión es válida incluso cuando  $v_d < 0$  siempre que la etapa de entrada siga trabajando en su zona lineal. Sin embargo en esta situación no se producirá la transición en la salida  $v_{OA}$  a  $v_{OA} \geq V_{OH}$  dado que para que esto ocurra se tiene que cumplir que  $v_A \leq V_{TRINV}$ , pero para que esta situación se produzca  $v_d$  ha de tomar valores tales que

$$v_d \leq \frac{(V_{TRINV} - V_A)(g_o + g_{mop})}{g_m} - \frac{I_{PHp}}{g_m} \quad (4.11)$$

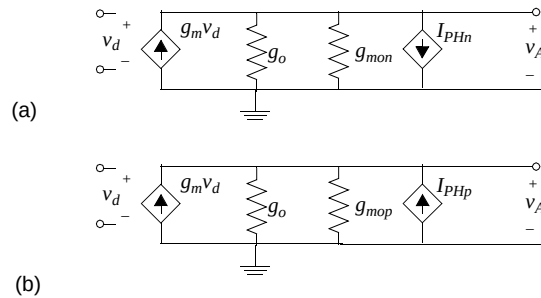


**Fig.4.15 Etapas de entrada y de histéresis: modelo para obtener  $V_{TH}^+$  (a); modelo para obtener  $V_{TH}^-$  (b).**

y por tanto  $v_{in} \leq V_{REF} + V_{TH^-}$  de donde se tiene que

$$V_{TH^-} = \frac{(V_{TRINV} - V_A)(g_o + g_{mop})}{g_m} - \frac{I_{PHp}}{g_m} \quad (4.12)$$

Las anteriores expresiones son válidas siempre que la etapa de entrada trabaje en su zona lineal  $[-\delta_m, \delta_m]$  y que se verifique además que  $V_{TRINV} \in [V_{SL}, V_{SH}]$ .

Por otra parte, en las expresiones (4.9) y (4.12) el primer término, que es común a ambas, puede ser considerado un término de *off-set* cuyo efecto puede ser minimizado por diseño si se hace  $V_{TRINV} \equiv V_A$ , o bien  $g_m \gg g_o + g_{mop}$ , por lo que puede ser despreciado en primera aproximación, con lo que se tienen las siguientes expresiones para  $V_{TH^+}$  y  $V_{TH^-}$

$$V_{TH^+} = \frac{I_{PHn}}{g_m} \quad V_{TH^-} = -\frac{I_{PHp}}{g_m} \quad (4.13)$$

que ponen de manifiesto que la anchura de la zona de histéresis puede ser controlada a partir de la fuente de corriente de polarización  $I_{POLH}$ , Fig.4.13(b).

En la Tabla 4.7 se muestran las expresiones de algunos de los parámetros empleados en los modelos de la Fig.4.14 y la Fig.4.15.

**Tabla 4.7:** Expresiones de algunos parámetros de los modelos empleados

| Parámetro   | Expresión                                                                                                                                                                                          | Parámetro                        | Expresión                                                       |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
| $V_{TRINV}$ | $\frac{V_{DD} - V_{T0p} + V_{T0n} \sqrt{\frac{2\beta_{0n}(\frac{W}{L})_{Mn1}}{2\beta_{0p}(\frac{W}{L})_{Mp1}}}}{1 + \sqrt{\frac{2\beta_{0n}(\frac{W}{L})_{Mn1}}{2\beta_{0p}(\frac{W}{L})_{Mp1}}}}$ | $r_{DSx_{ON}}$<br>con $x = n, p$ | $\frac{1}{2\beta_{0x}(\frac{W}{L})_{Mx} (v_{GS} -  V_T )_{Mx}}$ |
|             |                                                                                                                                                                                                    | $r_{oxPH}$<br>con $x = n, p$     | $\frac{1}{\lambda_{Mx_{ij}} I_{PH}}$                            |

Por su parte, y para completar el modelado de la etapa de histéresis, cabe señalar que el inversor CMOS formado por los transistores  $M_{p1}$  y  $M_{n1}$  en la Fig.4.13(a), que funciona en este caso como una etapa de comparación, también puede ser modelado en primer orden según el esquema que se ha discutido a propósito de las Fig.4.11(c) y Fig.4.11(d). La Fig.4.16(a) muestra el circuito que modela su comportamiento para este caso particular, mientras que en la Fig.4.16(b) se esquematiza su curva de transferencia. En este caso, la tensión de referencia para la comparación es el parámetro  $V_{TRINV}$  cuya expresión aparece en la Tabla 4.7. El valor de  $V_{TRINV}$  se fija por diseño, dada su dependencia de la geometría de los transistores  $M_{p1}$  y  $M_{n1}$ .

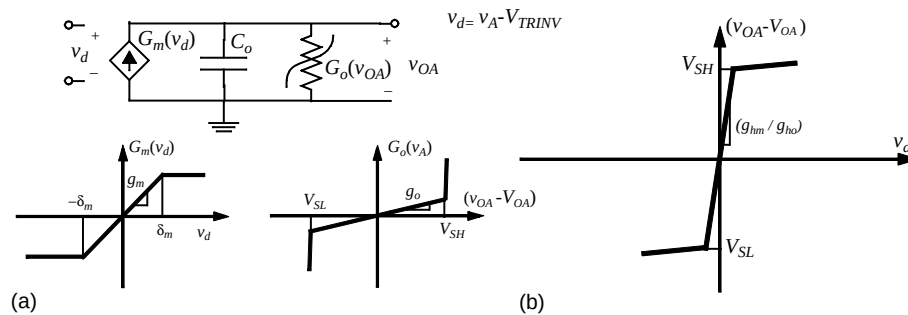


**Fig.4.16 Comparador inversor CMOS: modelo de comportamiento de primer orden (a); curva de transferencia (b).**

En la Tabla 4.8 se resumen las expresiones de los parámetros de primer orden que definen el modelo de la Fig.4.16(a).

**Tabla 4.8:** Parámetros del modelo de comportamiento del circuito de la Fig.4.16.

| Parámetro  | Expresión                                                                                                           |
|------------|---------------------------------------------------------------------------------------------------------------------|
| $g_m$      | $2\sqrt{\beta_{0n}\left(\frac{W}{L}\right)_{M_{n1}}} I_D + 2\sqrt{\beta_{0p}\left(\frac{W}{L}\right)_{M_{p1}}} I_D$ |
| $\delta_m$ | No aplicable                                                                                                        |
| $g_o$      | $I_D(\lambda_{M_{n1}} + \lambda_{M_{p1}})$                                                                          |
| $C_o$      | $C_L + W_{M_{n1}} C_{GD0M_{n1}} + W_{M_{p1}} C_{GD0M_{p1}}$                                                         |

Finalmente, la etapa de salida, está constituida por el inversor CMOS formado por los transistores  $M_{p2}$  y  $M_{n2}$  en la Fig.4.13(a). Por ello las expresiones de la Tabla 4.8 son igualmente aplicables a la etapa de salida. En conjunto ambos inversores CMOS, aportan una ganancia adicional a la ganancia de la etapa de entrada, lo que reduce el tiempo de respuesta del comparador. Este hecho se pone de manifiesto en las simulaciones que se recogen en la Fig.4.17. Además, la etapa de salida proporciona aislamiento para el nudo  $v_{OA}$  con respecto a la circuitería posterior.

La Tabla 4.9 resume los tamaños de los transistores que aparecen en los circuitos de la Fig.4.13 empleados en el diseño de estas etapas del comparador diseñado para el controlador ejemplo. En la Tabla 4.10 se muestran a modo de ilustración los valores de los parámetros recogidos en la Tabla 4.7, y calculados a partir de los valores de la Tabla 4.9, y de los propios

de la tecnología (ver Apéndice). Para esta cálculo se ha supuesto además que  $I_{PH} = 20\mu A$  y que se los transistores  $M_{n\ ij}$  y  $M_{p\ ij}$  del los espejos que proporcionan dicha corriente (ver Fig.4.13) permanecen siempre en saturación de forma que mientras los transistores  $M_{n1}$  y  $M_{p2}$  trabajen en su zona óhmica se cumple que  $v_{GS_{Mn1}} - V_{TMn1} \leq 3.087\ volt$  y que  $v_{SG_{Mp1}} - V_{TMP1} \leq 2.858\ volt$ .

**Tabla 4.9:** Tamaño de los transistores empleados en el diseño de los circuitos de la Fig.4.13

| $M_n$ (W/L)       | $M_p$ (W/L)       | $M_{n1}$ (W/L)    | $M_{p1}$ (W/L)    | $M_{n2}$ (W/L)    | $M_{p2}$ (W/L)    |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| $4\mu m/6\mu m$   | $4\mu m/2\mu m$   | $2\mu m/2.4\mu m$ | $3\mu m/0.8\mu m$ | $2\mu m/0.8\mu m$ | $4\mu m/0.8\mu m$ |
| $M_{n\ ph}$ (W/L) | $M_{p\ 00}$ (W/L) | $M_{n\ 00}$ (W/L) | $M_{p\ ij}$ (W/L) | $M_{n\ ij}$ (W/L) |                   |
| $15\mu m/3\mu m$  | $60\mu m/3\mu m$  | $15\mu m/3\mu m$  | $60\mu m/3\mu m$  | $15\mu m/3\mu m$  |                   |

**Tabla 4.10:** Valores de indicativos de los parámetros de la Tabla 4.7

| $V_{TRINV}$ (volts) | $r_{DS_{nON}}$ Kohm | $r_{DS_{pON}}$ Kohm | $r_{onPH}$ Kohm | $r_{opPH}$ Kohm |
|---------------------|---------------------|---------------------|-----------------|-----------------|
| 2.65                | > 3.9               | > 3.5               | 533             | 666             |

En la Fig.4.17 se muestran las curvas de transferencia del comparador con histéresis diseñado para el controlador ejemplo, resultado de diferentes simulaciones con HSPICE que ilustran su comportamiento para diferentes valores de la fuente de corriente de polarización

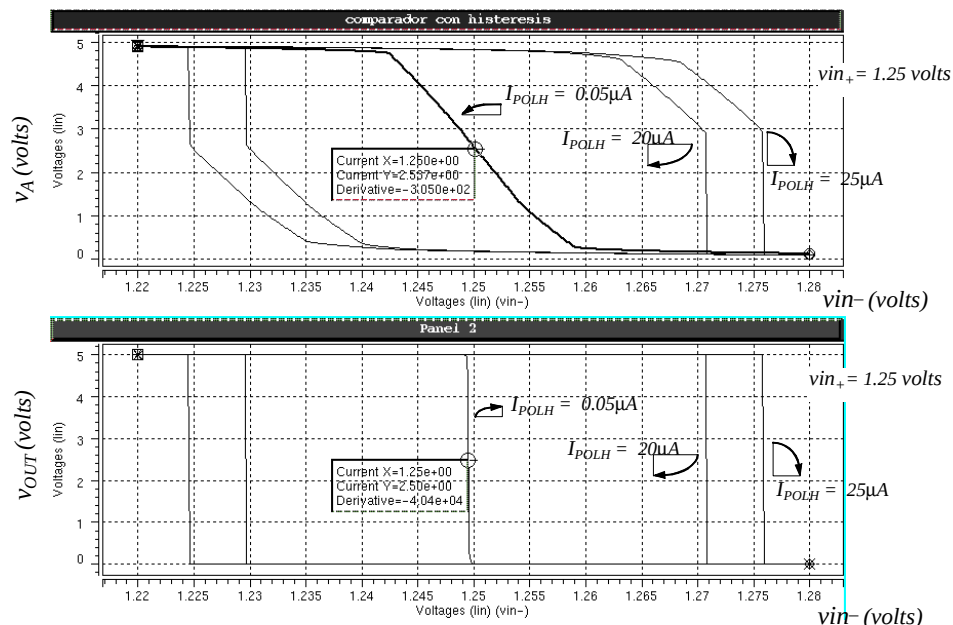


**Fig.4.17** Resultado de simulación con HSPICE de las curvas de transferencia del comparador con histéresis: curvas para el nudo de salida de la etapa de entrada  $v_A$  (arriba); curvas para el nudo de salida del comparador  $v_{OUT}$  (abajo).

$I_{POLH}$ , en Fig.4.13(b), y que muestran como varía, en función de ésta, el ancho de la región de histéresis. En estas simulaciones se ha fijado  $vin_+$  a 1.25 volts. mientras que para  $vin_-$  se han efectuado barridos ascendentes y descendentes entre 1.22volts y 1.28 volts.  $I_{POLH}$  toma los valores 0.05 $\mu$ A, 20 $\mu$ A y 25 $\mu$ A. Los valores de  $V_{TH}^+$  y  $V_{TH}^-$  medidos sobre estas curvas se ajustan a los calculados a partir de la expresión (4.13). En la parte superior de la Fig.4.17 se muestran las curvas para el nudo de salida de la etapa de entrada  $v_A$ , mientras que en la parte inferior se tiene la salida  $v_{OUT}$  de comparador completo. La medida de la pendiente en el punto medio de la curva de transferencia de  $v_A$ , coincide prácticamente con el valor para la ganancia  $k_s = \frac{g_m}{g_o}$  calculado que muestra la Tabla 6. Mientras que la medida de la pendiente en el punto medio de la curva de transferencia de  $v_{OUT}$  puede ser considerada como la ganancia del comparador completo. De la comparación entre ambas se pone de manifiesto claramente el incremento de ganancia que proporcionan los inversores de la etapa de histéresis y de salida, y por tanto su efecto de aceleración del tiempo de respuesta del comparador.

La Fig.4.18 muestra una simulación HSPICE que ilustra el comportamiento dinámico del comparador. La señal de referencia  $vin_+$  se ha fijado a 4.25 volts, mientras que la entrada  $vin_-$  realiza la secuencia de transiciones de tipo escalón de valores (4, 4.75, 4, 4.32, 4, 4.55, 4.20, 4.30, 4.20 volts) que supone probar diferentes amplitudes de *overdrive* de entrada.  $I_{POLH}$  se ha fijado a 17 $\mu$ A, con lo que según la expresión (4.13) se tiene un ancho de la región de histéresis de 50mV.

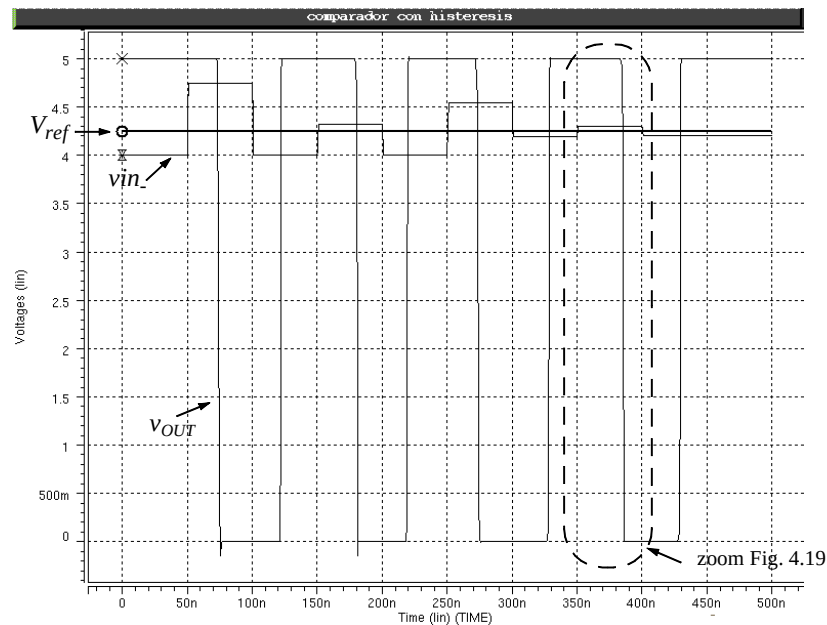


**Fig.4.18 Simulación HSPICE de la respuesta transitoria del comparador con histéresis del controlador ejemplo, para una secuencia con diferentes amplitudes de overdrive de entrada.**



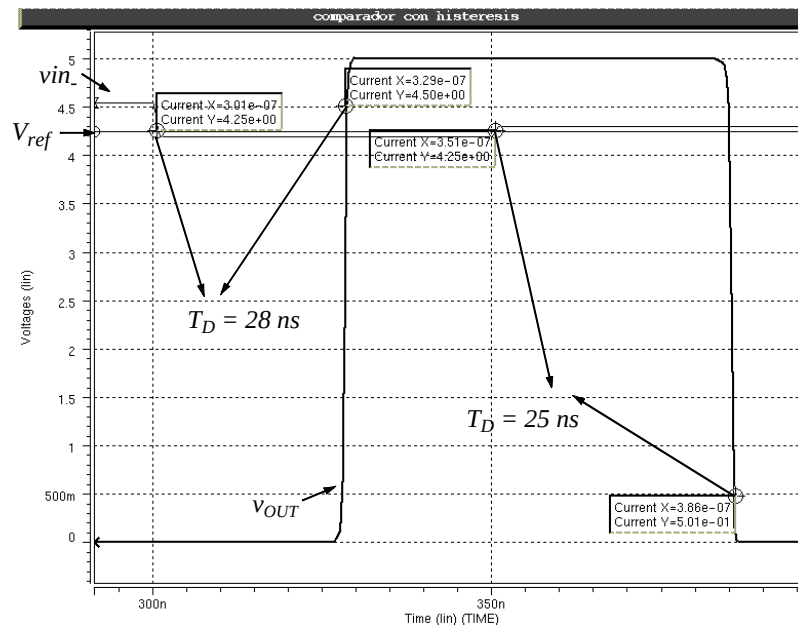


**Fig.4.19 Simulación HSPICE de la respuesta transitoria del comparador con histéresis del controlador ejemplo: Medida del tiempo de respuesta para una amplitud de overdrive de entrada de 50mV.**

El caso más crítico se muestra ampliado en la Fig.4.19 y corresponde a transición de menor amplitud de *overdrive* de entrada, (4.20, 4.30, 4.20). El tiempo de respuesta medido en este caso para una amplitud de *overdrive* de la entrada de 50mV sobre la referencia  $vin_+$  es de 28ns, como muestra dicha figura.

En la Fig.4.20 se muestra el *layout* completo de una celda comparador. La parte derecha corresponde al amplificador diferencial que constituye la etapa de entrada, mientras que en la parte izquierda están la etapa de histéresis y de salida. Dentro de la misma celda se han integrado los transistores  $M_{nQ\ 2ij}$  y  $M_{nQ\ 1ij}$  de la Fig.4.11(b) que forman el espejo de corriente que proporcionan las corrientes de polarización  $I_Q$  de cada etapa de entrada, por lo que el interfaz se establece a través de los nudos  $V_{Q1}$  y  $V_{Q2}$ . También se han incorporado a esta celda los transistores  $M_{p\ ij}$  y  $M_{n\ ij}$  de la Fig.4.13(b), que forman los espejos de corriente que proporcionan las corrientes de polarización  $I_{PH}$  de cada etapa de histéresis, por lo que el interfaz se establece a través de los nudos  $V_{ba}$  y  $V_{bb}$ . Los transistores  $M_{n1}$  y  $M_{n2}$  de la Fig.4.11(a), que forman el par diferencial se han dispuesto en la celda empleando una estructura de *layout* en centroide común, recomendada a fin de minimizar el *off set* debido al desapareamiento de dichos transistores.

La Fig.4.21 muestra un detalle de *layout* en el que aparecen conectados los seis comparadores que se emplean en cada convertor A/D, Fig.4.2(b). En el diseño final se ha prescindido del comparador que tiene como referencia el nivel inferior  $\varepsilon_1$ , con el consiguiente

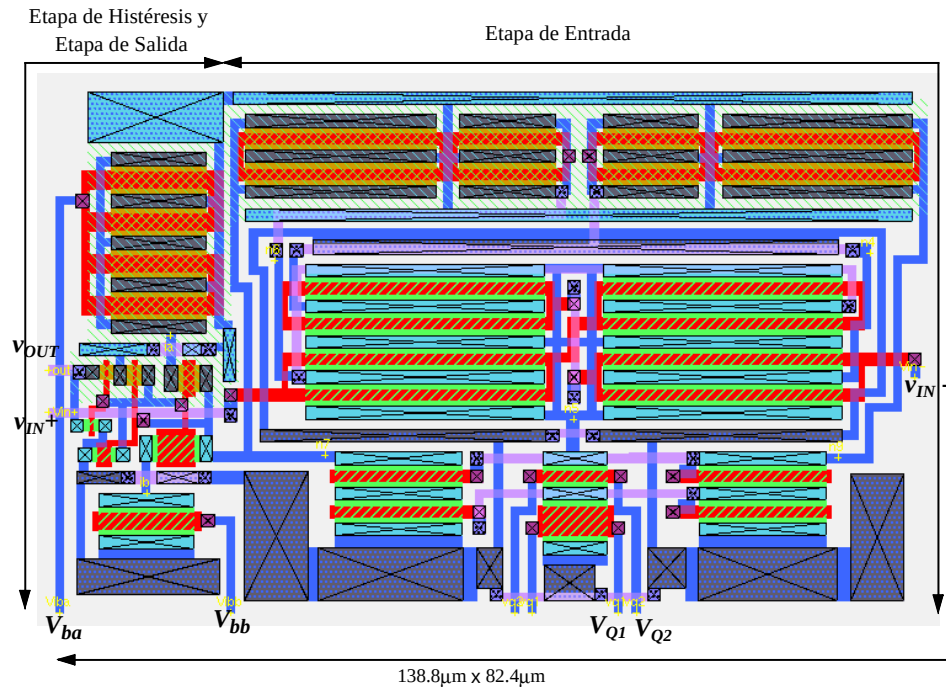


Fig.4.20 Layout completo de una de las celdas comparador con histéresis empleados en el diseño del controlador ejemplo.

ahorro en área. Esta opción permite definir una dirección de intervalo por defecto, lo que permite seleccionar un conjunto de valores de programación por defecto y asignar al Núcleo Analógico un estado conocido, incluso en el caso de que alguna de las señal de entrada estuviese fuera de su límite inferior en rango de valores permitidos.

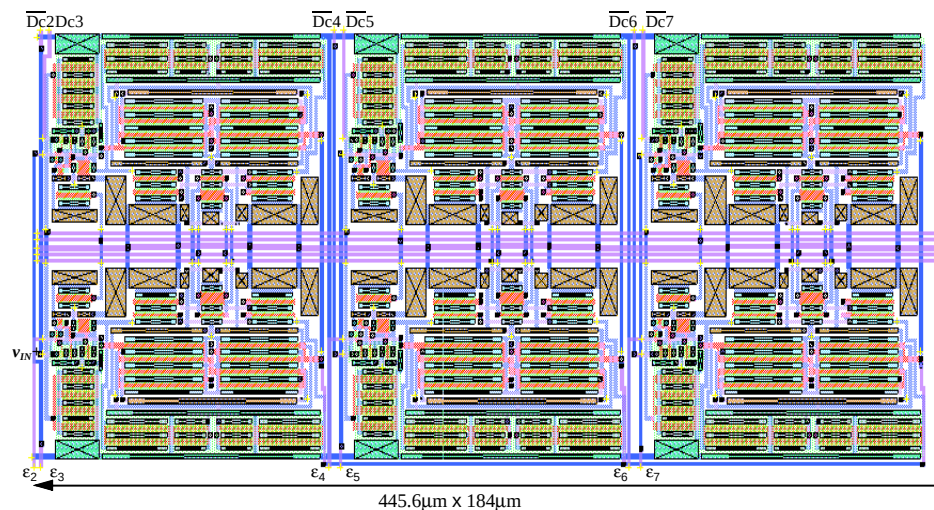
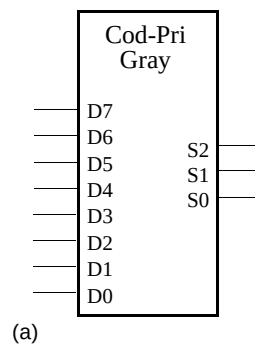


Fig.4.21 Layout del conjunto de comparadores con histéresis que forman cada uno de los conversores A/D.

## 2.3.3. Codificador Gray

Los códigos Gray son códigos binarios cíclicos con distancia unidad, esto es, la primera y la última palabra del código son consecutivas, y además, dos palabras de código consecutivas se diferencian únicamente en un bit. Esta última propiedad resulta importante porque minimiza el número líneas de datos que han de conmutar simultáneamente cuando se produce un cambio entre palabras de código adyacentes.

Un *Codificador Gray* de  $N$  bits es un circuito digital combinacional con  $2^N$  entradas y  $N$  salidas. Su función consiste en generar a la salida la palabra de código Gray asociada a la entrada que este activa en cada momento. Si se permite que puedan estar activas varias entradas a la vez, el código que se muestra a la salida es el de la entrada que sea considerada más prioritaria. Este tipo de codificadores se denominan *Codificadores con Prioridad*. Este es el caso del codificador necesario en esta aplicación, en el que se ha de codificar la escala termómetro que constituyen las salidas de los comparadores del conversor A/D y por lo tanto lo habitual es que varias entradas estén activas simultáneamente.



(a)

Tabla de verdad

| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | S2 | S1 | S0 | $C_{mn}$        |
|----|----|----|----|----|----|----|----|----|----|----|-----------------|
| x  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |                 |
| x  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | $C_{m1} C_{1n}$ |
| x  | x  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | $C_{m2} C_{2n}$ |
| x  | x  | x  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | $C_{m3} C_{3n}$ |
| x  | x  | x  | x  | 1  | 0  | 0  | 0  | 1  | 1  | 0  | $C_{m4} C_{4n}$ |
| x  | x  | x  | x  | x  | 1  | 0  | 0  | 1  | 1  | 1  | $C_{m5} C_{5n}$ |
| x  | x  | x  | x  | x  | x  | 1  | 0  | 1  | 0  | 1  | $C_{m6} C_{6n}$ |
| x  | x  | x  | x  | x  | x  | x  | 1  | 1  | 0  | 0  | $C_{m7} C_{7n}$ |

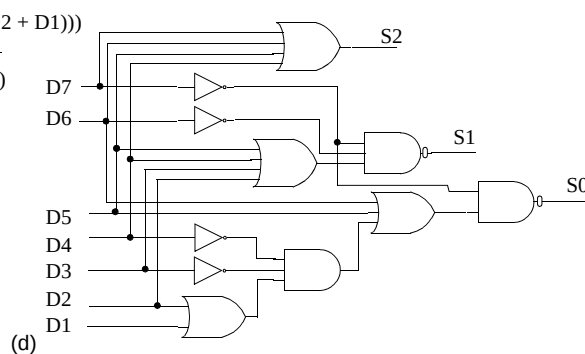
(b)

$$S0 = D7 * (D6 + D5 + (D4 * D3 * (D2 + D1)))$$

$$S1 = D7 * D6 * (D5 + D4 + D3 + D2)$$

$$S2 = D4 + D5 + D6 + D7$$

(c)



(d)

Fig.4.22 Codificador con prioridad ascendente en código Gray de tres bits: Símbolo (a); Tabla de Verdad (b); expresiones de las funciones booleanas de salida (c); diagrama lógico (d).

La Fig.4.22 muestra el símbolo, la tabla de verdad y las expresiones minimizadas de las funciones booleanas de cada una de sus salidas, correspondientes a un Codificador Gray de 3-bits con prioridad ascendente, que es el empleado en el circuito integrado MFCON desarrollado como ejemplo. En este caso se ha supuesto que las entradas son activas a nivel alto, esto es, 1-lógico. También se muestra en esa figura el correspondiente diagrama lógico. En el controlador ejemplo se han de codificar siete intervalos de interpolación,  $C_{mn}$  por entrada, como muestra la Fig.4.5(b). En el controlador ejemplo la primera palabra del código Gray que muestra la tabla de la Fig.4.22 no es asignada a ninguno de esos intervalos, el resto de palabras de código se han asignado según el orden creciente del subíndice correspondiente a cada entrada.

En el diseño ejemplo, las funciones booleanas S0 S1 y S2, han sido realizadas directamente como funciones lógicas CMOS, a partir de las expresiones

$$\begin{aligned} S2 &= \overline{D4} \cdot \overline{D5} \cdot \overline{D6} \cdot \overline{D7} \\ S1 &= \overline{D7} \cdot \overline{D6} \cdot (D5 + D4 + D3 + D2) \\ S0 &= \overline{D7} \cdot (D6 + D5 + (\overline{D4} \cdot \overline{D3} \cdot (D2 + D1))) \end{aligned} \quad (4.14)$$

que son equivalentes a las que aparecen en la Fig.4.22. La Fig.4.23 muestra el esquema de los circuitos CMOS empleados en su realización. Las líneas S[0:2] se hacen corresponder con las líneas  $A_{1b}[0:2]$  para el convertor asociado a la entrada  $x_1$  y  $A_{2b}[0:2]$  para el convertor asociado a la entrada  $x_2$ , según la notación adoptada en la Fig.4.2.

Por razones de simplicidad y ahorro de área, y por no constituir éste un aspecto crítico en el diseño, en el controlador ejemplo todos los transistores utilizados en el diseño del

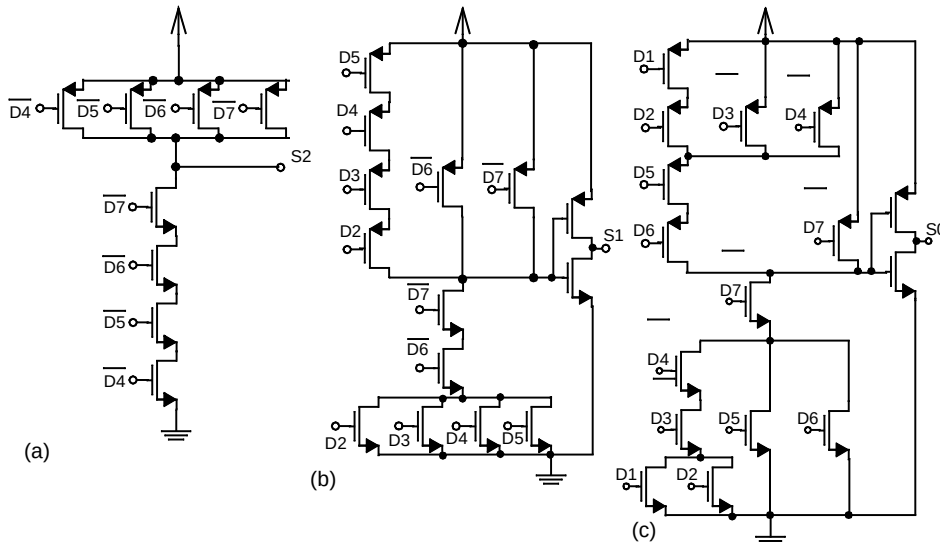


Fig.4.23 Realización mediante lógica CMOS de las funciones booleanas de salida del codificador Gray: S2 (a); S1 (b) y S0 (c).

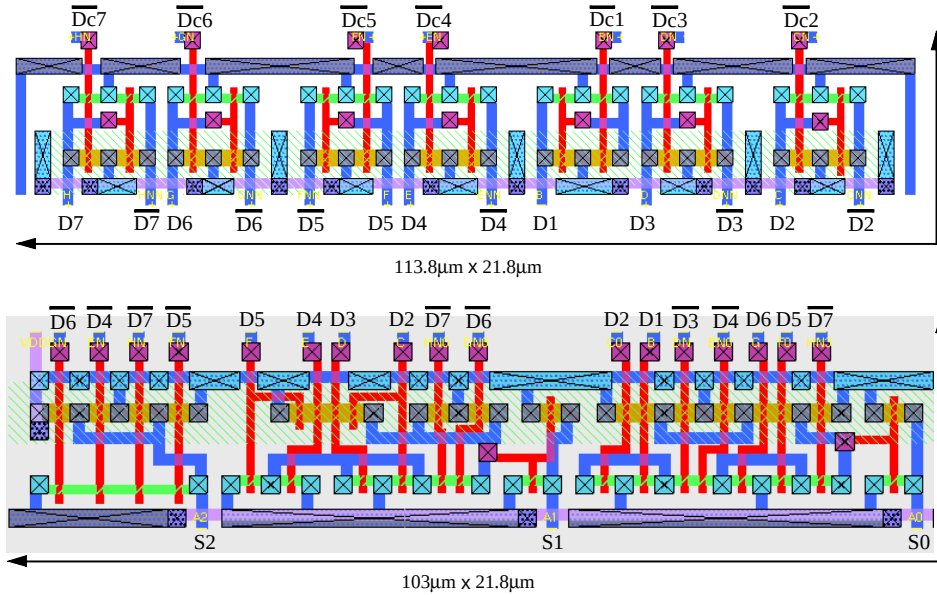


**Fig.4.24 Detalles de layout: Interfaz bloque de comparadores (a); codificador Gray diseñado según el esquema de la Fig.4.23, (b).**

Codificador Gray tienen longitud de canal mínima, ( $L = 0.8\mu\text{m}$ ), mientras que para el ancho de los transistores  $n$  y  $p$ ,  $W_n$  y  $W_p$ , respectivamente, se ha escogido  $W_n$  también de tamaño mínimo ( $W_n = 1\mu\text{m}$ ) y  $W_p$  tal que  $W_p = 2W_n$ , de forma que se igualen las transconductancias  $\beta_n$  y  $\beta_p$  de ambos tipos de transistores.

La Fig.4.24(b) muestra el *layout* del bloque codificador Gray según el esquema de la Fig.4.23. El interfaz entre este bloque y el bloque de comparadores se realiza a través del conjunto de inversores, cuyo *layout* se muestra en la Fig.4.24(a), y que proporcionan las entradas  $D_i$  y sus complementarias, a partir de las salidas de los comparadores denominadas  $\overline{Dc_i}$ , en la Fig.4.21.

#### 2.3.4. Consideraciones Globales de Diseño

Para ilustrar el funcionamiento del *Bloque Selector de Intervalo* que se ha diseñado para el controlador ejemplo MFCN, en la Fig.4.25 y en la Fig.4.26 se recoge el resultado de dos simulaciones HSPICE realizada a partir del fichero *netlist* extraído del *layout* final de dicho controlador. En la Fig.4.25 se muestra el correcto funcionamiento de los dos conversores A/D empleados en el diseño, al hacer que sus líneas de salida ( $S[2:0]$  en la Fig.4.24) recorran todos los valores posibles, al aplicar en cada una de las entradas,  $x_1$  y  $x_2$ , sendas señales en rampa que recorren todo el espacio de entrada en ambos sentidos, ascendente y descendente. Nótese como para la gráfica correspondiente a la entrada  $x_1$ , Fig.4.25(a), las señales  $S[2:0]$  evolucionan según la tabla de verdad de la Fig.4.22(b).

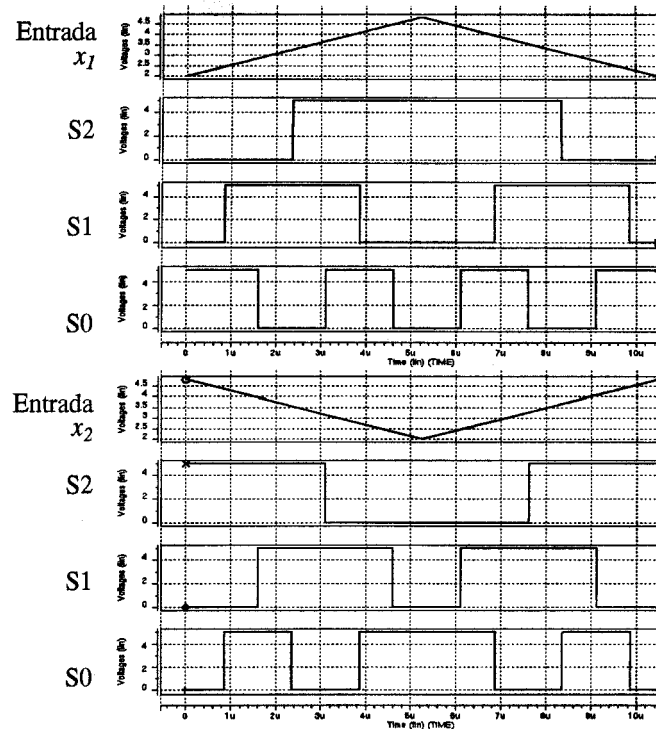


**Fig.4.25 Simulación HSPICE que ilustra el funcionamiento del Bloque Selector de Intervalo: Entrada  $x_1$  y salidas S[2:0] de convertor A/D asociado (a); entrada  $x_2$  y salidas S[2:0] de convertor A/D asociado(b).**

Por su parte, en la Fig.4.26 se muestra la evolución de las líneas de salida de los comparadores que forman el convertor A/D, Fig.4.26(b) (líneas D[2:7], en la Fig.4.24), así como las líneas de salida del Codificador GRAY, Fig.4.26(c) (S[2:0] en la Fig.4.24), y que constituyen la salida del bloque *Selector de Intervalo*, cuando la entrada  $x_1$  del controlador evoluciona según la secuencia de valores que muestra la Fig.4.26(a), mientras la entrada  $x_2$  permanece a un valor fijo de 2.4 volts. Dicha secuencia supone transitar desde el intervalo inicial  $C_{m1}$ , a través de los intervalos  $C_{m2}$ ,  $C_{m4}$ ,  $C_{m6}$ ,  $C_{m2}$ ,  $C_{m7}$ ,  $C_{m6}$ ,  $C_{m4}$ ,  $C_{m3}$  hasta el intervalo final  $C_{m7}$ , como puede apreciarse siguiendo la evolución de las líneas en gráficas de las Fig.4.26(b) y Fig.4.26(c).

Finalmente, para ilustrar el retardo que introduce el *Bloque Selector de Intervalo*, en la Fig.4.27 se muestra un detalle ampliado de las transiciones entre los intervalos  $C_{m6}$  a  $C_{m2}$ , recogida en la Fig.4.26, y que resulta ser el caso más lento. En ella se aprecia como, tras la transición en la entrada  $x_1$ , las líneas de salida de los comparadores D[6:3] se desactivan. La línea D3 resulta ser la más lenta en desactivarse. Esta circunstancia es debida a que esta salida

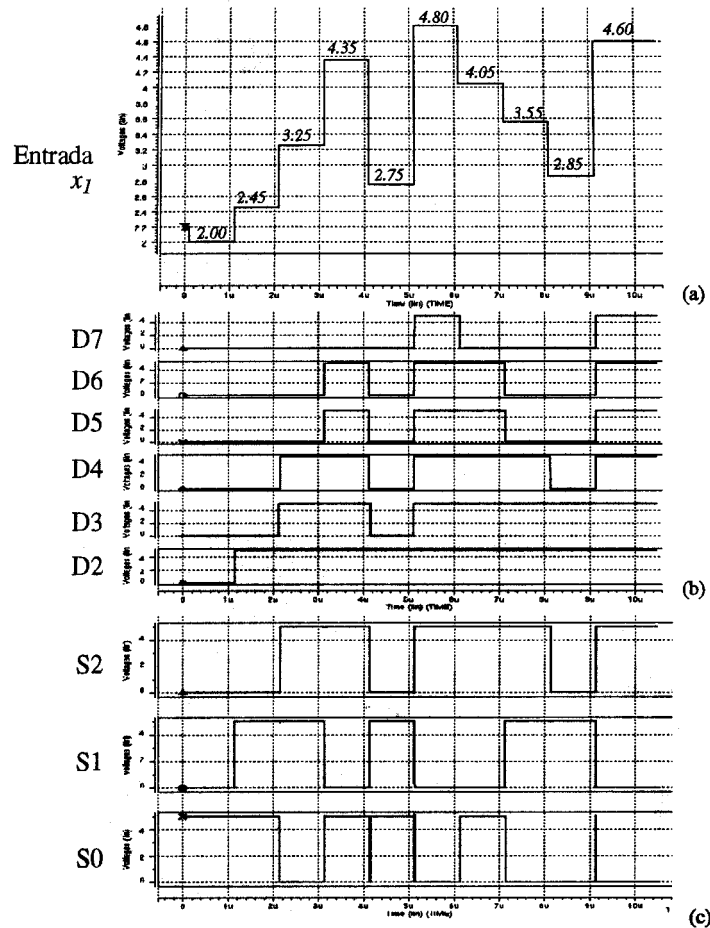


Fig.4.26 Simulación HSPICE que ilustra el funcionamiento del Bloque Selector de Intervalo para una secuencia de entrada: Entrada  $x_1$  (a); salidas de los comparadores (b); líneas de salida del codificador (c).

corresponde al comparador para el que la transición de la entrada  $x_1$  presenta un menor *overdrive*, dado que la entrada  $x_1$  toma un valor final de 2.75 volts, que resulta ser inferior en 50mV, respecto al valor de referencia de 2.80 volts, correspondiente a dicho comparador. Este mayor retardo produce una transición espuria en la línea S0 de salida del codificador Gray antes de alcanzar finalmente su valor estable. El tiempo de retardo medido, como muestra la gráfica de la Fig.4.27 es de aproximadamente 60ns.



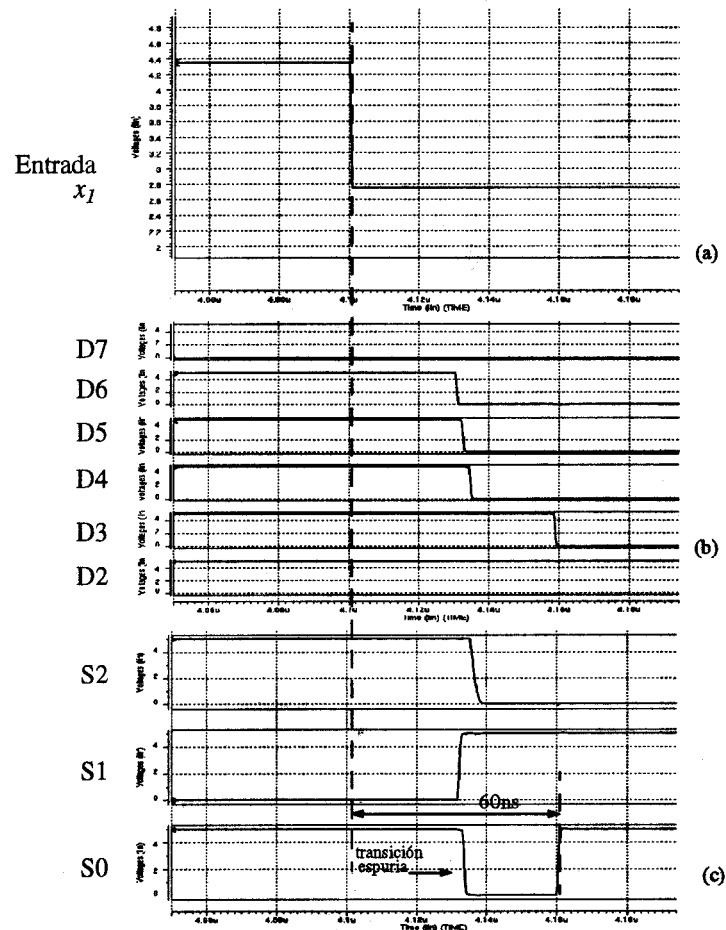


Fig.4.27 Simulación HSPICE. Detalle para ilustrar el retardo del Bloque Selector de Intervalo: Entrada  $x_1$  (a); salidas de los comparadores (b); líneas de salida del codificador (c)





## Capítulo 5

### Descripción Funcional y Diseño CMOS de los *Bloques de Multiplexado (II)*: Programador de Antecedente de Reglas y Programador de Consecuente de Reglas

#### 1. INTRODUCCIÓN

En este capítulo se completa la descripción de los *Bloques de Multiplexado del Núcleo Analógico*, iniciada en el capítulo anterior, con la descripción de los bloques *Programador de Antecedente de Reglas* y *Programador de Consecuente de Reglas*. En primer lugar se destacan sus principales requerimientos y especificaciones de diseño y a continuación se describen y analizan los circuitos empleados en el diseño de un controlador ejemplo que posee dos entradas, una salida y sesenta y cuatro reglas, y que ha sido fabricado con tecnología CMOS estándar de  $0.7\mu\text{m}$ .

#### 2. PROGRAMADOR DE ANTECEDENTE DE REGLAS

La función principal de este bloque es proporcionar el conjunto de valores analógicos de programación  $PA_{C_{k_1 \dots k_M}} = \{E_1, \dots, E_k, \dots, E_M\}$ , que permiten configurar cada uno de los  $M$  *Bloques de Etiquetas del Núcleo Analógico* (ver Fig.4.1) para trabajar en el intervalo de interpolación  $C_{k_1 \dots k_M}$  seleccionado.

##### 2.1. Discusión de los requerimientos y objetivos de diseño.

Su principal requerimiento de diseño es realizar su función, de la forma más rápida, fiable, simple y compacta posible.

##### 2.2. Propuesta de diseño

Para cumplir con este requerimiento se propone diseñar este bloque de forma modular, a partir de  $M$  celdas idénticas, una por cada entrada del sistema global, distribuidas como se muestra en la Fig.5.1(a). Cada una de estas celdas constituye un multiplexor analógico controlado digitalmente [Pall93][Yama93]. En conjunto, sus  $M$  salidas forman un bus de programación analógico de  $M$  líneas,  $(E_1 \dots E_M)$  en la Fig.4.1.

El valor de cada una de las líneas  $E_i$  es seleccionado, mediante un control digital, de entre los  $L$  valores de referencia ( $E_{i1} \dots E_{iL}$ ) que definen los centros de los intervalos de interpolación en el espacio de cada entrada, a partir del código generado por el bloque *Selector de Intervalo*. Como se indica en el apartado dedicado al bloque *Selector de Intervalo*, los valores ( $E_{i1} \dots E_{iL}$ ) son generados en la serie de resistores lineales allí definida, y pueden ser compartidos por todas las entradas en el caso de una partición uniforme (ver Fig.4.3).

En la Fig.5.1(b) se muestra una de las  $M$  celdas que constituyen este bloque. A partir de la salida  $A_i[0:p]$  de cada uno de los convertidores A/D del bloque *Selector de intervalo*, un decodificador Gray proporciona las señales de control que actúan sobre cada interruptor, de los que conforman el multiplexor analógico, permitiendo que la salida  $E_i$  tome el valor adecuado de entre los posibles valores ( $E_{i1} \dots E_{iL}$ ).

### 2.3. Diseño CMOS del bloque Programador de Antecedente de Reglas.

A continuación se analizan y caracterizan los circuitos y elementos empleados para realizar el diseño propuesto en el apartado anterior para tecnologías CMOS estándar.

#### 2.3.1. Multiplexor analógico con control digital

Un multiplexor es un sistema de encaminamiento de información que permite seleccionar y transmitir en un instante dado una de entre un conjunto de fuentes de información. Si la fuente

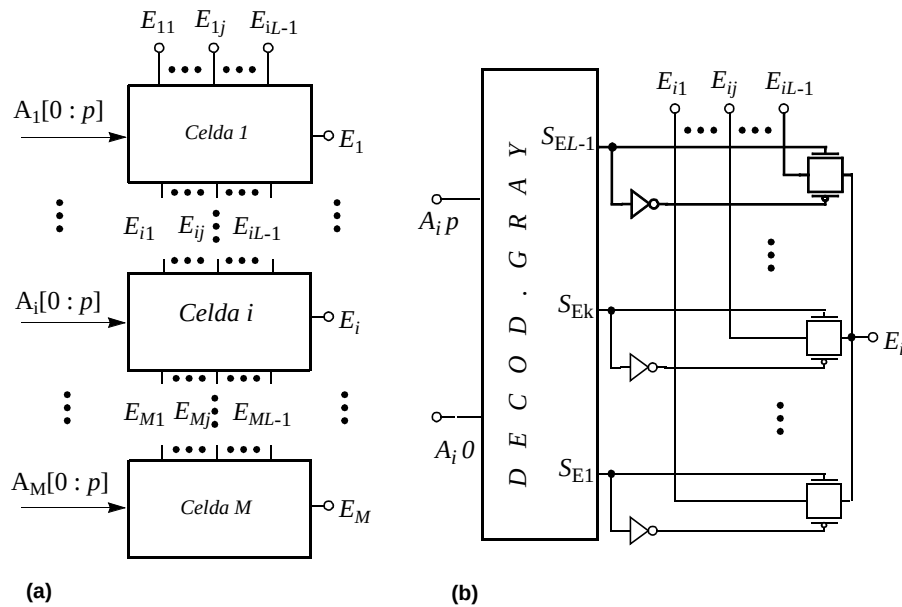


Fig.5.1 (a) Bloque Programador de Antecedente de Reglas; (b) Celda  $i$ -ésima .

de información es de tipo analógico, se habla entonces de multiplexor analógico. Si el control de la selección se realiza de forma digital, el mejor elemento para realizar este multiplexor es un *switch* analógico.

Un *switch* ideal, como el que es mostrado en la Fig.5.2(a), es un elemento de circuito que posee dos estados de funcionamiento, ON y OFF y tres terminales  $A, B$  y  $C$ . El terminal  $C$  es considerado el terminal de control que permite seleccionar el estado del elemento. En el estado ON el elemento se comporta como un cortocircuito entre los terminales  $A$  y  $B$ ; mientras que en el estado OFF el elemento se comporta como un circuito abierto entre dichos terminales. En la práctica los *switches* realizables tienen un comportamiento no ideal que puede modelarse como se indica en la Fig.5.2(b). En ese modelo se supone que el estado del *switch* se controla mediante una señal digital  $v_C$  que se aplica en el terminal  $C$ . Las características más importantes de este modelo son su resistencia en ON,  $R_{ON}$  y su resistencia en OFF  $R_{OFF}$ . Idealmente  $R_{ON}$  se supone nula mientras que  $R_{OFF}$  se supone infinita. La fuente de tensión  $V_{OS}$  modela la pequeña diferencia de potencial que puede aparecer entre los terminales  $A$  y  $B$ , incluso en el caso de que la corriente en ON sea nula; mientras que la fuente de corriente  $I_{OFF}$  representa la corriente de fuga que puede fluir entre los terminales  $A$  y  $B$ , en el estado OFF. En la Fig.5.2(b) las polaridades de dichas fuentes se han tomado de forma arbitraria puesto que en la práctica estas resultan desconocidas *a priori*. Finalmente  $R_A$  y  $R_B$  representan las resistencias vistas desde cada uno de los terminales y tierra; mientras que los condensadores  $C_A$ ,  $C_B$ ,  $C_{AB}$ ,  $C_{AC}$  y  $C_{BC}$  representan las capacidades parásitas entre los terminales y tierra y entre los propios terminales, y modelan la respuesta en frecuencias del elemento real [Alle87].

Por sí mismos, tanto los transistores NMOS como los PMOS pueden ser empleados como *switches* dadas sus características de comportamiento en gran señal. Sin embargo, en

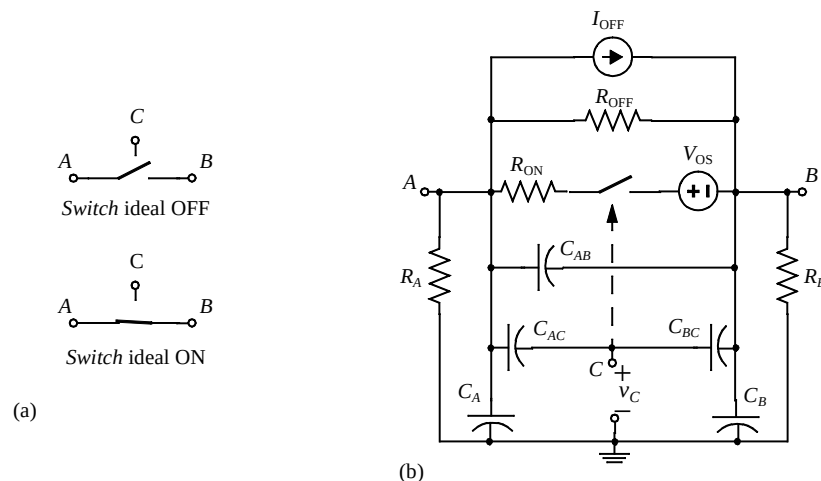


Fig.5.2 *Switch* analógico: modelo ideal (a); modelo no ideal (b).

tecnologías CMOS, un *switch* que reproduce un comportamiento bastante próximo al ideal, con un mejor rango dinámico en su estado ON, se construye conectando en paralelo los terminales fuente y el drenador de un transistor del canal **p** y otro de canal **n**, ambos de enriquecimiento, y de forma que sus terminales de puerta están conectados a señales digitales de control complementarias,  $\phi$  y  $\bar{\phi}$ , como muestra la Fig.5.3(a) <sup>1+</sup>. (Este sistema también se denomina puerta de transmisión en terminología digital).

Considerando solamente su comportamiento estático, cuando  $\phi = 0$  y por tanto  $\bar{\phi} = V_{DD}$ , se tiene que  $-V_A < V_{Tn}$  y que  $V_A - V_{DD} < V_{Tp}$ , (donde  $V_A$  es la tensión en el terminal A), en esta situación ambos transistores están cortados con lo que los terminales A y B del switch están prácticamente en circuito abierto, (la corriente de fuga está en el rango de los picoamperios), como se representa en la Fig.5.3(b).

Cuando  $\phi = V_{DD}$ , ( $\bar{\phi} = 0$ ), entre los terminales A y B se establece un camino de baja impedancia correspondiente a la resistencia en ON de uno o ambos transistores, dependiendo del valor de tensión analógico  $V_A$  a transmitir al terminal B. Si, como se simboliza en la Fig.5.3(c),  $V_{DD} - V_A \leq V_{Tn}$  y  $V_A \geq V_{Tp}$ , el transistor **n** está cortado, mientras que el transistor **p** conduce. Si, como se simboliza en la Fig.5.3(d),  $V_{DD} - V_A \geq V_{Tn}$  y  $V_A \leq V_{Tp}$ , es el transistor **p** el que ahora está cortado, mientras que el transistor **n** conduce. Finalmente, si, como se

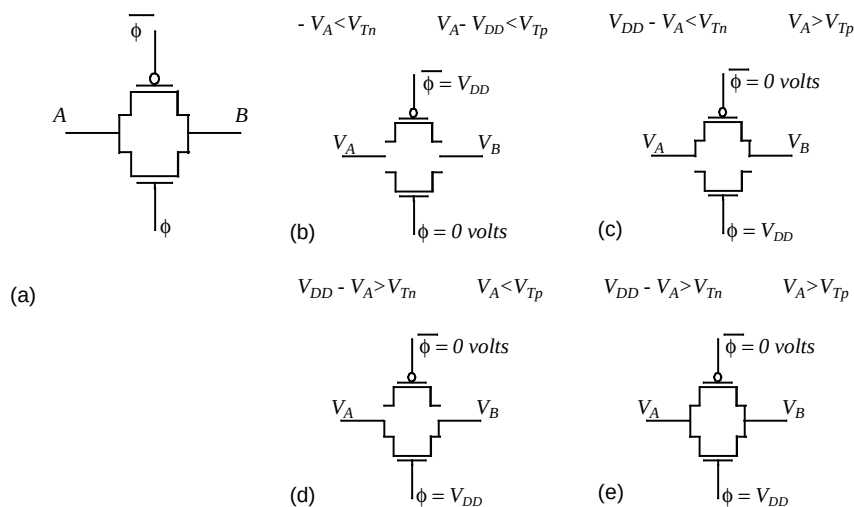


**Fig.5.3 Switch CMOS (a); diferentes estados: no conducción (b); conducción p (c); conducción n (d); conducción p y n (e).**

1+. En esta figura los dos tipos de transistores n y p son representados con la simbología empleada habitualmente en contextos digitales, y que en este caso pone de manifiesto la bidireccionalidad del sistema representado y la intercambiabilidad práctica de los terminales fuente y drenador de los transistores MOS. En esta representación el transistor **p** se distingue del **n** por el circulito que se añade a su terminal de puerta.

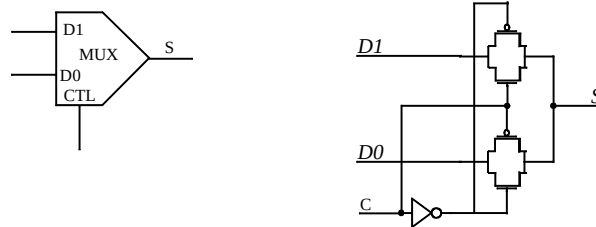


Fig.5.4 Multiplexor de dos entradas y una salida realizado con *switches* CMOS.

simboliza en la Fig.5.3(d),  $V_{DD} - V_A \geq V_{Tn}$  y  $V_A \geq V_{Tp}$ , ambos transistores conducen. En cualquiera de los tres casos se tiene un camino de baja impedancia entre los terminales A y B que considerando su comportamiento en estático puede ser considerado prácticamente como un cortocircuito.

El comportamiento dinámico del *switch* viene determinado por la capacidad de carga vista en el terminal B y las capacidades parásitas de ambos transistores. En el contexto de este diseño el retardo introducido es despreciable por lo que su análisis no es aquí considerado.

La Fig.5.4 muestra a modo de ejemplo el circuito que realiza un multiplexor simple con dos entradas y una señal de control realizado con este tipo de elementos.

En el controlador ejemplo MFCON que ha sido fabricado, el conjunto de parámetros de programación consta de dos elementos  $PA_{C_{mn}} = \{E_1, E_2\}$ , cada uno de los cuales ha de ser seleccionado de entre el conjunto de siete valores analógicos definido en la Tabla 4.2. La Fig.5.5 muestra un detalle de *layout* de uno de los multiplexores de siete entradas y una salida empleados en el diseño del controlador ejemplo. En él se han utilizado transistores de tamaño mínimo ( $L_n = L_p = 0.8\mu\text{m}$ ,  $W_n = W_p = 1\mu\text{m}$ ), para minimizar las capacidades parásitas y consumo de área. Por otra parte hay que destacar que dado que el terminal de salida E es un nudo de alta impedancia, en condiciones estáticas prácticamente no circula corriente a través del multiplexor, por lo que cualquier efecto debido a la resistencia en ON de los transistores resulta despreciable.

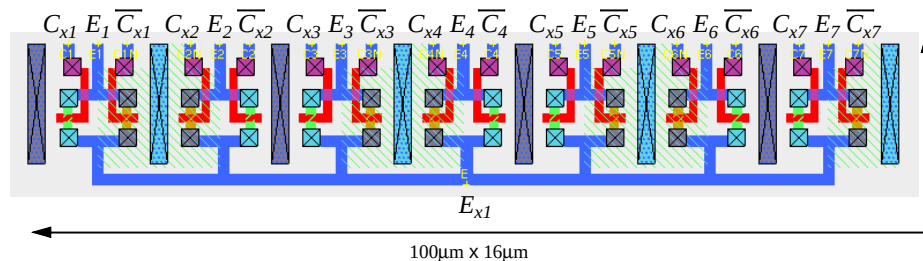


Fig.5.5 Detalle de *layout* de uno de los multiplexores empleados en el controlador ejemplo.

## 2.3.2. Control del multiplexor analógico

En la sección 2.2. se propone emplear un *Decodificador Gray* para generar las señales de control del multiplexor analógico, como muestra la Fig.5.1(b). Un *Decodificador Gray* de  $N$  bits es un circuito digital combinacional con  $N$  entradas y  $2^N$  salidas. Su función consiste activar la línea de salida asociada a la palabra en código Gray que se pone a su entrada en cada momento. Básicamente cada una de las salidas del decodificador corresponde a un minitérmino de una función de  $N$  variables, por lo que el diseño de un decodificador es directo.

La Fig.5.6 muestra el símbolo, la tabla de verdad y las expresiones minimizadas de las funciones booleanas de cada una de sus salidas, correspondientes a un Decodificador Gray de 3-bits, junto a su correspondiente diagrama lógico.

Para mayor simplicidad, y dado que el diseño *full-custom* aquí desarrollado así lo permite, en el controlador ejemplo, las funciones booleanas  $S[1 : 7]$  y sus complementarias

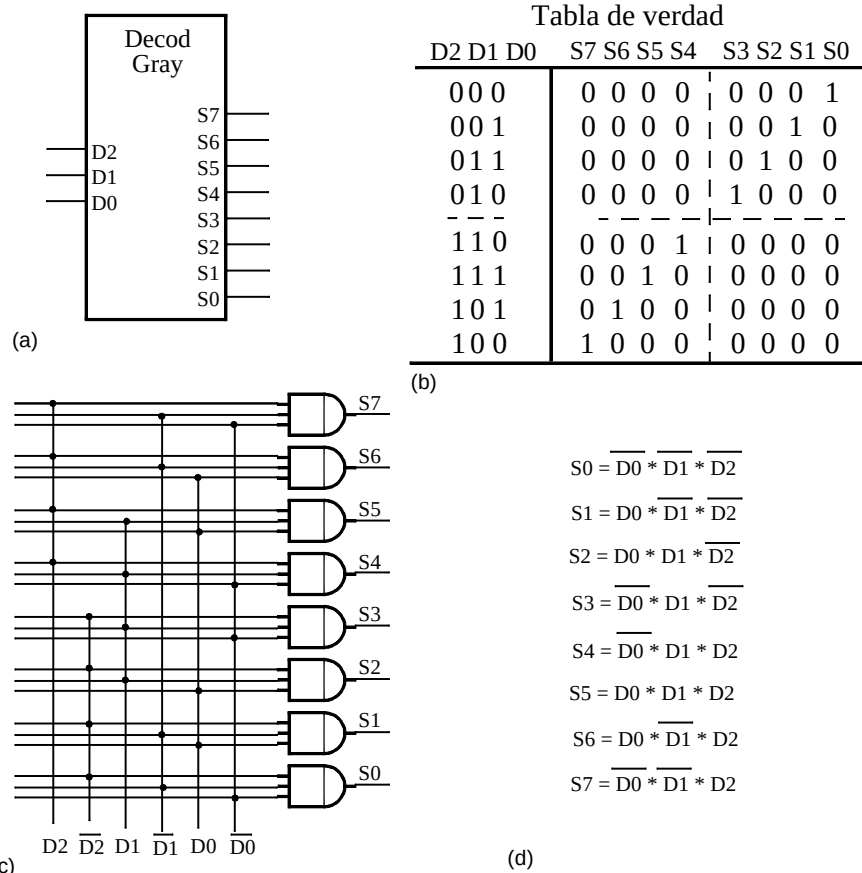


Fig.5.6 Decodificador código Gray de tres bits: Símbolo (a); Tabla de Verdad (b); diagrama lógico (c); expresiones de las funciones booleanas de salida (d).

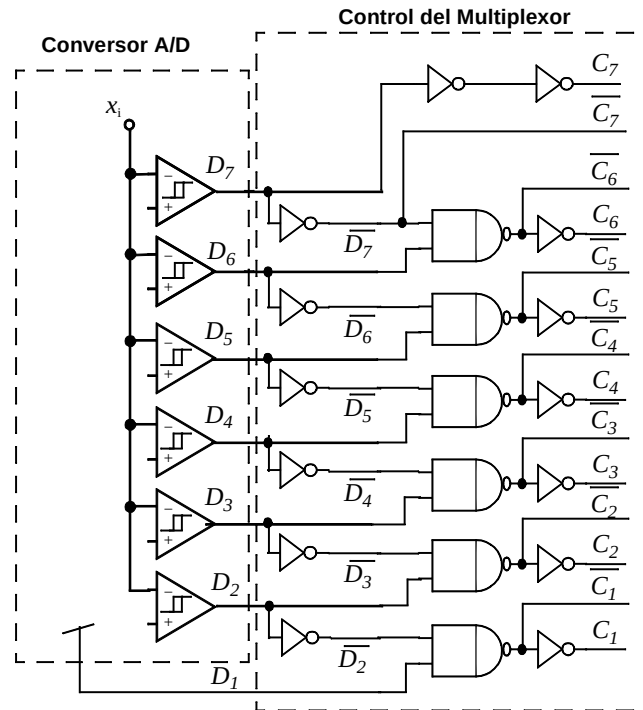


**Fig.5.7 Generación de las señales de control de los multiplexores analógicos a partir de las salidas de los comparadores del convertor A/D, en el controlador ejemplo.**

$\overline{S[1:7]}$ , que son empleadas para controlar el multiplexor analógico, (nombradas como  $C[1:7]$  en la Fig.5.5, Fig.5.7 y Fig.5.8), han sido realizadas directamente a partir de las salidas de los comparadores del convertor A/D, y no a partir de la decodificación del código Gray por éste generado, como se ha descrito con anterioridad, aprovechando la particularidad del carácter de escala termómetro de estas señales para simplificar las funciones booleanas del bloque de control del multiplexor. En la Fig.5.7 se muestra el diagrama lógico del sistema de control del multiplexor. En ella se muestra cómo se utilizan las salidas  $D_2$  a  $D_7$  de los seis comparadores que forman cada convertor A/D, además de una entrada adicional  $D_1$  que toma

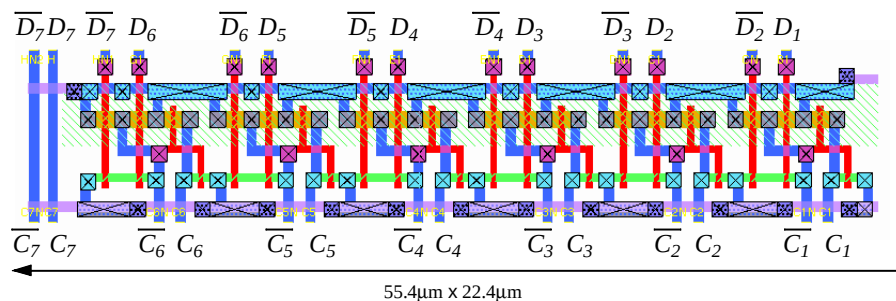


**Fig.5.8 Detalle de layout del control del multiplexor analógico en el controlador ejemplo.**



siempre el valor lógico 1 y que corresponde al comparador de la referencia del límite inferior del espacio de entrada que, como se indicó con anterioridad, ha sido eliminado en el controlador ejemplo. La Fig.5.8 muestra un detalle de *layout* del circuito que realiza el control. Todos los transistores utilizados en su diseño tienen longitud de canal mínima, ( $L = 0.8\mu\text{m}$ ), mientras que para el ancho de los transistores **n** y **p**,  $W_n$  y  $W_p$ , respectivamente, se ha escogido  $W_n$  también de tamaño mínimo ( $W_n = 1\mu\text{m}$ ) y  $W_p$  tal que  $W_p = 2W_n$ , de forma que se igualen las transconductancias  $\beta_n$  y  $\beta_p$  de ambos tipos de transistores.

### 2.3.3. Consideraciones Globales de Diseño

Para ilustrar el funcionamiento del *Bloque Programador de Antecedente de Reglas* diseñado en el controlador ejemplo, en la Fig.5.9 y Fig.5.10 se recoge el resultado de dos simulaciones HSPICE realizadas a partir del fichero *netlist* extraído del *layout* final de MFCON.

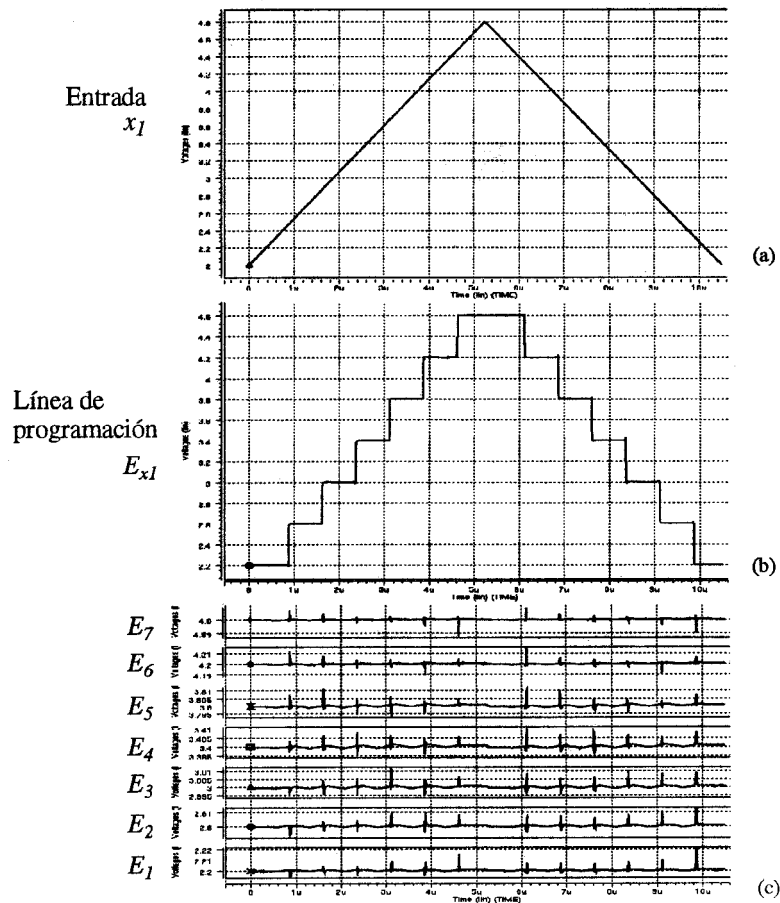


Fig.5.9 Simulación HSPICE que ilustra el funcionamiento del Bloque Programador de Antecedente de Reglas : Entrada  $x_1$  (a); línea de Programación (b); líneas de entrada del multiplexor analógico (c)

Las gráficas de la Fig.5.9 corresponden a la misma simulación que se presenta en las Fig.4.25, en la que ambas entradas del controlador se hacen evolucionar siguiendo una rampa que recorre por completo su correspondiente espacio de entrada como recuerda la Fig.5.9(a). La Fig.5.9(b) muestra línea de salida  $E_{x1}$  del multiplexor analógico asociado a la entrada  $x_1$ , en ella se puede apreciar como dicha línea evoluciona tomando en cada momento los valores de programación adecuados, según el intervalo de interpolación en el que en cada momento se encuentra la señal de entrada. Por su parte la Fig.5.9(c) muestra las líneas de entrada del multiplexor analógico, que representan los valores de tensión de entre los cuales es seleccionado el valor de salida  $E_{x1}$ .

Por su parte, en las gráficas de la Fig.5.10, que corresponde a la simulación que se presenta en la Fig.4.26, se muestra la evolución de la línea de salida del multiplexor analógico

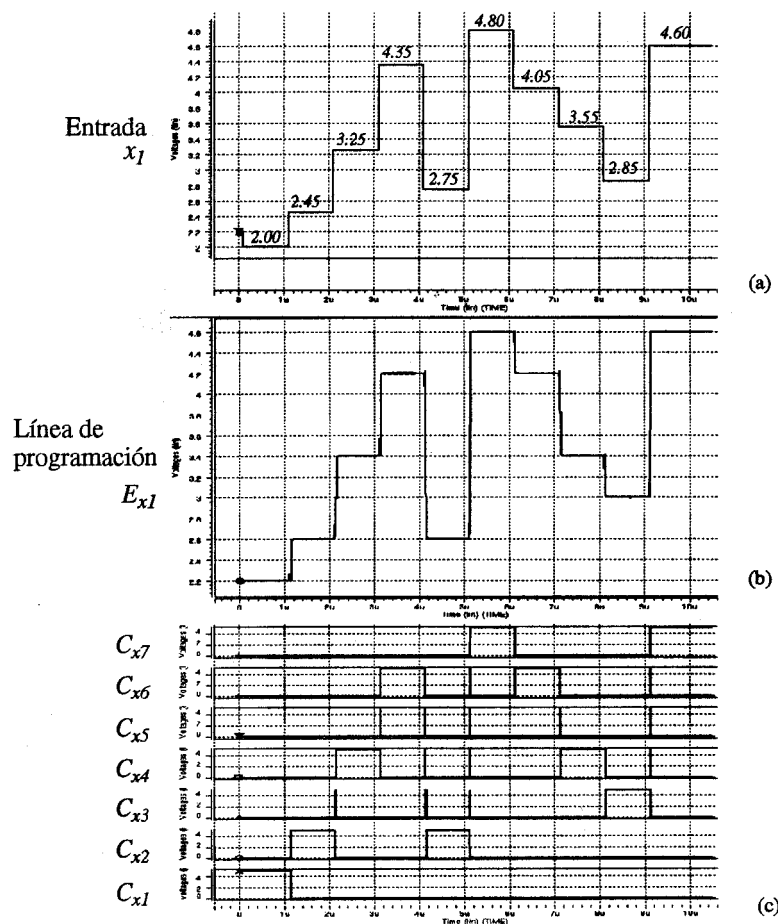


Fig.5.10 Simulación HSPICE que ilustra el funcionamiento del Bloque Programador de Antecedente de Reglas para una secuencia de entrada: Entrada  $x_1$  (a); línea de Programación  $E_{x1}$  (b); líneas de control del multiplexor analógico (c).

$E_{x1}$ , Fig.5.10(b), así como las líneas de control del multiplexor analógico, Fig.5.10(c) (líneas  $E$  y  $C_{x[1:7]}$  respectivamente en la Fig.5.5), cuando la entrada  $x_1$  del controlador evoluciona según la secuencia de valores que muestra la Fig.5.10(a), mientras la entrada  $x_2$  permanece a un valor fijo de 2.4 volts. Dicha secuencia supone transitar desde el intervalo inicial  $C_{m1}$ , a través de los intervalos  $C_{m2}$ ,  $C_{m4}$ ,  $C_{m6}$ ,  $C_{m2}$ ,  $C_{m7}$ ,  $C_{m6}$ ,  $C_{m4}$ ,  $C_{m3}$  hasta el intervalo final  $C_{m7}$ , según se ha discutido en el ejemplo presentado en la Fig.4.26 y que puede seguirse a partir de la activación de las líneas de control de la Fig.5.10(c). Como ilustra la Fig.5.10(b) la línea  $E_{x1}$  va tomando los valores de programación adecuados según el intervalo seleccionado siguiendo el esquema presentado en la Fig.4.5(b).

Finalmente, para ilustrar el retardo en la programación de los *Bloques de Regla* del *Núcleo Analógico*, desde que se produce una transición en la entrada que da lugar a un cambio

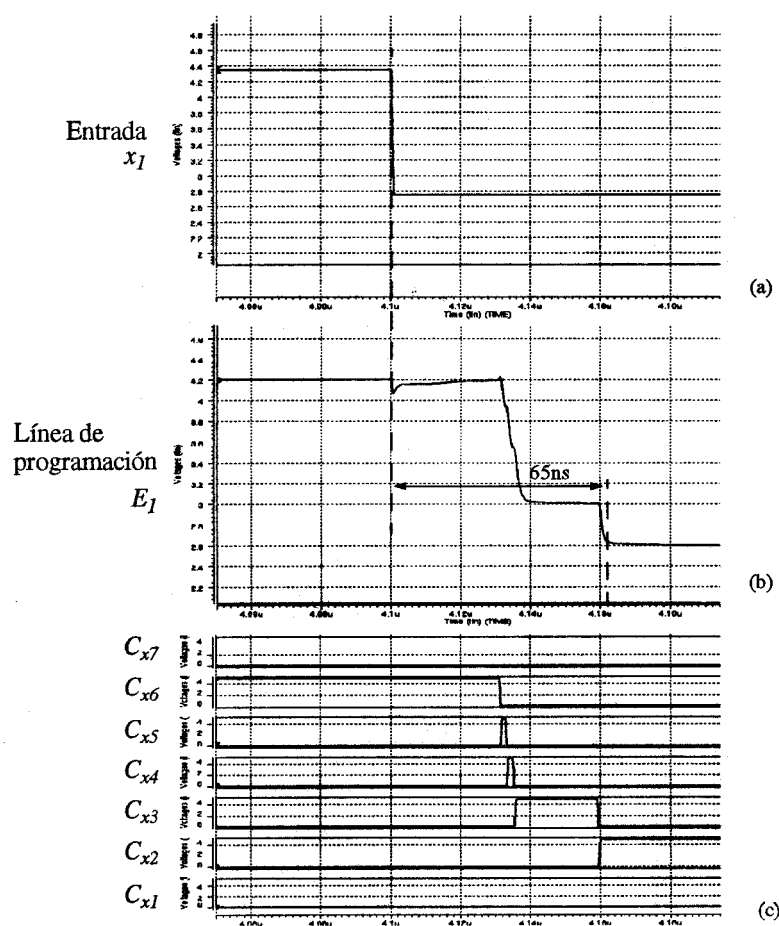


Fig.5.11 Detalle de una simulación HSPICE que ilustra el retardo en la programación del antecedente de reglas: Entrada  $x_1$  (a); línea de Programación (b); líneas de control del multiplexor analógico (c)

de intervalo de interpolación, e ilustrar que el retardo que introduce el bloque *Programador de Antecedente de Reglas* es despreciable frente al retardo debido al bloque *Selector de Intervalo*, en la Fig.5.11 se muestra un detalle ampliado de la transiciones entre los intervalos  $C_{m6}$  a  $C_{m2}$ , recogida en la Fig.5.10, y que resulta ser el caso más lento. En ella se aprecia como, tras la transición en la entrada  $x_1$ , Fig.5.11(a) la línea de control del multiplexor analógico  $C_{x6}$  se desactiva, y finalmente se activa la línea de control  $C_{x2}$ ; también se aprecia que durante el transitorio se producen *glitches* en las líneas  $C_{x5}$ ,  $C_{x4}$ , y  $C_{x3}$ , siendo este último el de mayor duración. Este comportamiento se justifica según el mismo argumento discutido en el ejemplo de la Fig.4.27, en donde se ilustra como la línea D3 resulta ser la más lenta en desactivarse, debido a que esta línea se activa como consecuencia de la desactivación de la línea de salida del comparador para el que la transición de la entrada  $x_1$  presenta un menor *overdrive*, dado que la entrada  $x_1$  toma un valor final de 2.75 volts, que resulta ser inferior en 50mV, respecto al valor de referencia de 2.80 volts, correspondiente a dicho comparador. A su vez este comportamiento justifica la evolución de la línea  $E_{x1}$  que muestra la Fig.5.11(b), en la que se aprecia como ésta evoluciona desde un valor inicial de 4.2 volts. correspondiente al intervalo  $C_{m6}$  hasta alcanzar el valor 2.8 volts, correspondiente al intervalo  $C_{m2}$ . El tiempo de retardo medido, como muestra la gráfica de la Fig.5.11 es de aproximadamente 65ns.

### 3. PROGRAMADOR DE CONSECUENTE DE REGLAS

La principal función de este bloque es proporcionar para cada intervalo de interpolación  $C_{k_1 \dots k_M}$  el conjunto de valores *singleton*,  $PC_{C_{k_1 \dots k_M}} = \left\{ y_1^*, \dots, y_l^*, \dots, y_{2^M}^* \right\}$  correspondientes a los consecuentes de las reglas activas, y que programan el *Bloque de Reglas* del *Núcleo Analógico*. Estos valores son seleccionados de entre el conjunto de valores *singleton*,  $y_{j_1 \dots j_M}^*$  ( $1 \leq j_i \leq L$ ) y ( $1 \leq i \leq M$ ), que constituyen la base de reglas del sistema completo.

Por tanto este bloque ha de proporcionar el soporte de almacenamiento apropiado, en cuanto a tamaño y formato, para albergar dicha información; además del interfaz adecuado para acceder a ella. (ver la Fig.4.1).

#### 3.1. Discusión de los requerimientos y objetivos de diseño.

Desde el punto de vista de su funcionamiento cabe distinguir dos fases, en principio, claramente diferenciadas, de manera que, en general, no coinciden en el tiempo y que imponen diferentes requerimientos al diseño de este bloque:

- Durante la **fase de configuración**, por parte de un procesador externo, son necesarios

accesos tanto de lectura como de escritura sobre una estructura de almacenamiento de información de tamaño y formato adecuado. Dado que el modelo elegido para realizar el *Núcleo Analógico* emplea para la etapa de ponderación con valores *singleton* datos en formato digital, de manera que cada uno de los valores *singleton*  $y_i^*$  es representado mediante una palabra de  $S$  bits, el sistema de almacenamiento más adecuado resulta ser una memoria digital. Por otra parte, el acceso a esta memoria necesita un interfaz de comunicación con el procesador externo que debe resultar lo más simple posible.

- Durante la **fase de actividad**, dentro de un bucle de control, son necesarios accesos de lectura internos, ésto es, iniciados desde el bloque *Selector de Intervalo*, que permitan obtener el conjunto de valores *singleton* necesarios para programar el consecuente de reglas dinámicamente y en tiempo continuo. Estos accesos han de cumplir con el protocolo que ha de ser definido para el interfaz entre dicho bloque y el *Bloque Consecuente de Reglas*, y que necesariamente ha de contemplar el carácter asíncrono que exige la consideración de funcionamiento en tiempo continuo, expresada como requerimiento de funcionamiento del sistema global.

En base a estas consideraciones, los principales requerimientos de diseño pueden concretarse en los siguientes puntos:

- Almacenar de forma eficiente el conjunto completo ( $L^M \times S$  bits) de valores *singleton* que constituyen los consecuentes del total de reglas del sistema global.
- Proporcionar accesos asíncronos de lectura que permitan seleccionar y obtener de forma rápida y simultánea los  $2^M$  elementos del conjunto  $PC_{C_{k_1 \dots k_M}}$ , necesarios para configurar en cada momento el *Núcleo Analógico*.
- Proporcionar accesos lectura y escritura externos para programación y actualización de los valores *singleton*.

Estos requerimientos condicionan tanto la arquitectura del bloque como temporización de los accesos en la forma que se concreta en los siguientes apartados.

### 3.1.1. Discusión de los requerimientos de la arquitectura.

Desde el punto de vista de la arquitectura, ésto es, por lo que respecta a cuestiones como el tamaño, la distribución de datos y la forma de acceso al sistema de almacenamiento de los valores *singleton*, así como al interfaz entre este bloque y el *Núcleo Analógico*, el principal condicionamiento de diseño está relacionado con conjugar, por una parte, que los  $2^M$  elementos del conjunto  $PC_{C_{k_1 \dots k_M}}$  han de ser obtenidos simultáneamente, y por otra, que, como consecuencia de la partición en rejilla escogida para el espacio de entrada, cada regla difusa genérica puede pertenecer a distintos conjuntos de reglas activas (Capítulo 2), por lo que cada

valor *singleton* genérico  $y_{j_1 \dots j_M}^*$  puede aparecer en la programación en distintos intervalos de interpolación, ésto es, formando parte de diferentes conjuntos  $PC_{C_{k_1 \dots k_M}}$ , o lo que es lo mismo, cada valor *singleton* genérico  $y_{j_1 \dots j_M}^*$  puede ser seleccionado por direcciones de selección  $AC_{k_1 \dots k_M}$  pertenecientes a diferentes intervalos  $C_{k_1 \dots k_M}$  y aparecer asociado a diferente  $y_l^*$  en cada uno de ellos.

Para ilustrar estas cuestiones, y sus consecuencias para el diseño, volvamos sobre el ejemplo de un sistema bidimensional como el empleado en la sección 2.1. del Capítulo 2.

En la Fig.5.12 se muestran cuatro intervalos de interpolación adyacentes,  $(C_{mn}, C_{m(n-1)}, C_{(m-1)n}, C_{(m-1)(n-1)})$  pertenecientes a una partición en rejilla realizada sobre un espacio de entrada bidimensional, (región punteada en la Fig.5.12(a)). Como se destaca en la Fig.5.12(b), la regla  $R_{mn}$ , que corresponde al centro de la citada región, será regla activa en cada uno de esos cuatro intervalos adyacentes, y por tanto su valor *singleton* asociado,  $y_{mn}^*$ , ha de poder ser direccionado desde cualquiera las cuatro direcciones  $(AC_{mn}, AC_{m(n-1)}, AC_{(m-1)n}, AC_{(m-1)(n-1)})$ .

Por otra parte, si en cada intervalo de interpolación genérico consideramos una ordenación relativa de las reglas activas en ese intervalo, tal y como se muestra en el ejemplo de la Fig.5.12(c), la regla que es común a los cuatro intervalos, ocupará una posición relativa distinta en cada uno de ellos. Así, por ejemplo, utilizando la ordenación de la Fig.5.12(c), la

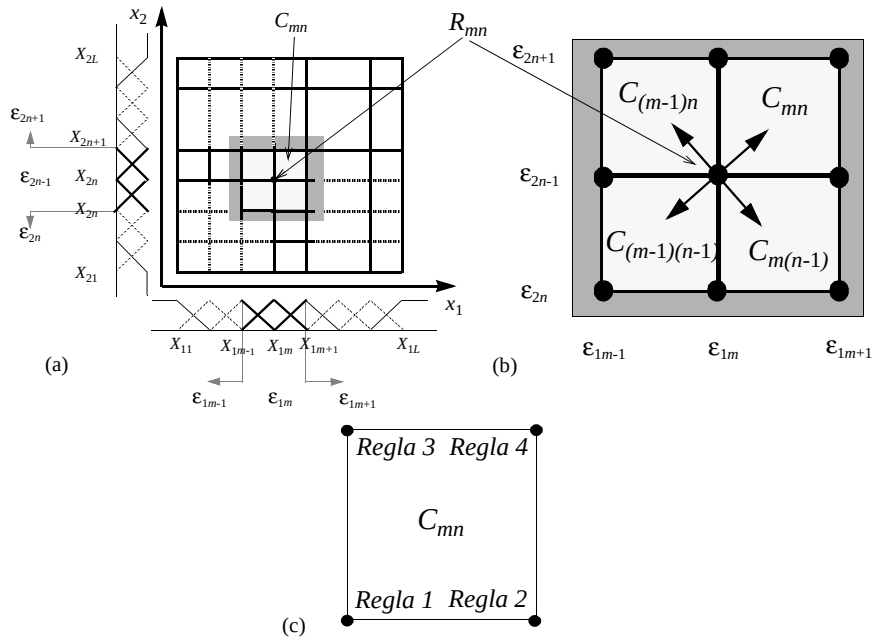


Fig.5.12 (a) Partición en intervalos de interpolación; (b)  $R_{mn}$  es regla activa en cuatro intervalos adyacentes; Ordenación de reglas relativa al intervalo  $C_{mn}$ . (c).

regla  $R_{mn}$  es la *Regla 1* en el intervalo  $C_{mn}$ , pero es la *Regla 2* en el intervalo  $C_{(m-1)n}$ , la *Regla 3* en el intervalo  $C_{m(n-1)}$  y la *Regla 4* en el intervalo  $C_{(m-1)(n-1)}$ . En cada uno de estos casos, el valor *singleton* asociado al consecuente de esa regla,  $y_{mn}^*$ , aparecerá en cada uno de los cuatro conjuntos de parámetros de programación  $PC_{C_{mn}}$ ,  $PC_{C_{(m-1)n}}$ ,  $PC_{C_{m(n-1)}}$  y  $PC_{C_{(m-1)(n-1)}}$ , asociado a un  $y_l^*$  (con  $l = 1, \dots, 4$ ) diferente.

En general, para un sistema de  $M$  entradas y  $L$  etiquetas por entrada, cada regla podrá ser regla activa en  $2^M$  intervalos adyacentes. Además, para el caso general, donde se tienen  $2^M$  intervalos adyacentes, dada una ordenación de las  $2^M$  reglas activas de los intervalos, la regla común ocupará una posición relativa distinta en cada uno de ellos.

Estas circunstancias han de ser tenidas en cuenta tanto a la hora de definir la organización de los datos, a fin de facilitar su accesibilidad, y evitar la redundancia en su almacenamiento; como a la hora de determinar la organización del interfaz entre el bloque *Programador de Consecuente de Reglas* y el *Núcleo Analógico*.

Consideraremos en primer lugar la cuestión del interfaz entre el bloque *Programador de Consecuente de Reglas* y el *Núcleo Analógico*.

#### 3.1.1.1. Interfaz de programación del Núcleo Analógico.

A la hora de definir el interfaz entre el bloque *Programador de Consecuente de Reglas* y el *Núcleo Analógico* se presentan dos alternativas:

- Hacer que cada valor *singleton* genérico  $y_{j_1 \dots j_M}^*$  aparezca siempre en una misma posición  $y_l^*$  en todos los conjuntos  $PC_{C_{k_1 \dots k_M}}$ , manteniendo fijo el interfaz del bloque *Programador de Consecuente de Reglas*. Esto supone permitir una reconfiguración dinámica, no sólo de los *Bloques de Etiqueta* y *Bloques de Regla*, como ya se contempla en la arquitectura del *Núcleo Analógico* mostrada en la Fig.3.3 del Capítulo 3, sino también de las interconexiones entre estos bloques, a fin de reordenar las reglas en el *Núcleo Analógico* según el intervalo de interpolación en que se esté trabajando.
- Hacer que cada valor *singleton*  $y_{j_1 \dots j_M}^*$  aparezca en una posición  $y_l^*$  diferente en cada uno de los conjuntos  $PC_{C_{k_1 \dots k_M}}$ , reordenando los valores *singleton*, y reorganizando así el interfaz del bloque *Programador de Consecuente de Reglas*, manteniendo el orden relativo de las reglas activas en cada intervalo. En este caso sólo se permite una reconfiguración dinámica de los *Bloques de Etiquetas* y *Bloques de Reglas*, manteniendo fija la interconexión entre estos bloques; ésto es, fijando la arquitectura y el interfaz del *Núcleo Analógico*.

A modo de ilustración consideremos cómo resulta el interfaz entre el bloque *Programador del consecuente de Reglas* y el *Bloque de Reglas del Núcleo Analógico* para cada una de las dos opciones antes mencionadas.

En la Fig.5.13 y la Fig.5.14 se muestra, para un caso bidimensional, y para dos intervalos de interpolación adyacentes  $C_{(m-1)n}$  y  $C_{mn}$  (Fig.5.13(a) y Fig.5.14(a)), cuáles son los conjuntos de parámetros de programación  $PC_{C_{(m-1)n}}$  y  $PC_{C_{mn}}$  que debe proporcionar el bloque *Programador de Consecuente de Reglas*, y el interfaz resultante en cada caso, Fig.5.13(c) y (d) y Fig.5.14(c). Como puede observarse, ambos intervalos comparten las reglas  $R_{mn}$  y  $R_{m(n+1)}$  y en consecuencia los dos valores *singleton* ( $y^*_{mn}$  e  $y^*_{m(n+1)}$ ). Precisamente el seguimiento de estos valores permite ilustrar las diferencias entre ambas alternativas.

En el ejemplo de la Fig.5.13, que coincide con la alternativa a), antes mencionada, lo que

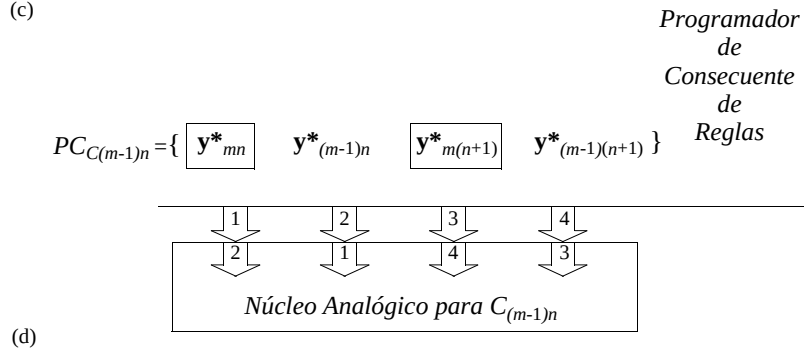
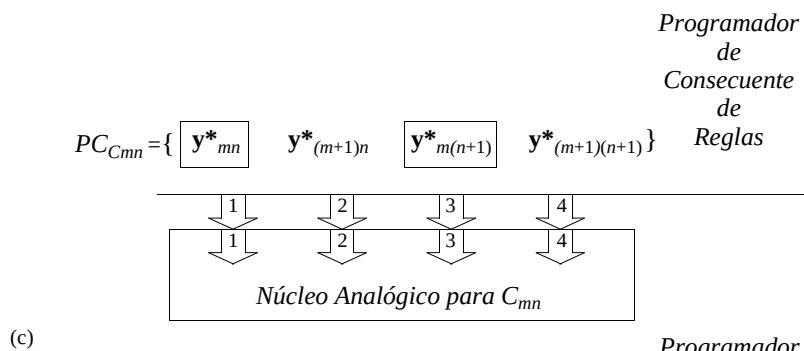
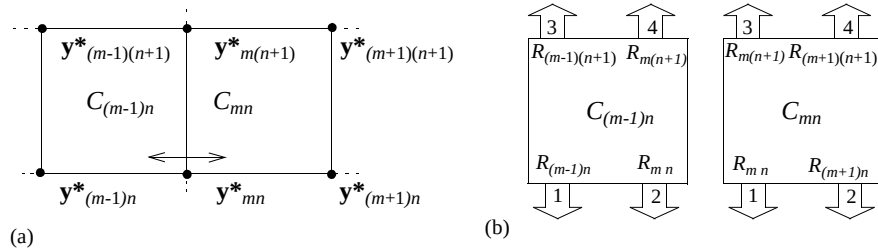


Fig.5.13 Ejemplo Interfaz Núcleo Analógico- Programador de Consecuente de Reglas



se quiere es mantener fija la posición de los *singleton* ( $y^*_{mn}$  and  $y^*_{m(n+1)}$ ) en cada uno de los conjuntos  $PC_{C_{mn}}$  y  $PC_{C_{(m-1)n}}$ ; ésto es, asignarles una posición fija en los conjuntos  $PC_{C_{mn}}$  y  $PC_{C_{(m-1)n}}$  atendiendo al orden absoluto de las reglas. Por tanto se ha de reorganizar el *Núcleo Analógico*, como se muestra en la Fig.5.13(c) y Fig.5.13(d) para hacer corresponder cada uno de los valores de programación con la regla adecuada según el orden que se indica en la Fig.5.13(b). Como puede observarse, mientras que para el intervalo  $C_{mn}$  el orden coincide, no ocurre así para el intervalo  $C_{(m-1)n}$ , donde el orden entre las reglas 1 y 2 por un lado y 3 y 4 por otro se invierte en el *Núcleo Analógico*.

Si por el contrario, como ocurre en el ejemplo de la Fig.5.14, que coincide con la alternativa b), en todos los intervalos de interpolación se mantiene el orden relativo de reglas como se indica en la Fig.5.14(b), lo que supone no alterar la estructura del *Núcleo Analógico*; entonces, ambos valores *singleton* deben aparecer en diferente orden en cada uno de los conjuntos  $PC_{C_{(m-1)n}}$  y  $PC_{C_{mn}}$  para que éste se programe correctamente, como muestra la Fig.5.14(c).

La primera alternativa presenta la ventaja de simplificar el diseño del bloque *Programador del Consecuente de Reglas*, y la organización del los datos a almacenar. Sin embargo, se penaliza la ruta de proceso del *Núcleo Analógico*, al tener que introducir la correspondiente circuitería de multiplexado para la reordenación de las reglas, que añade fuentes adicionales de error. Una propuesta de este tipo aparece en [Batu98].

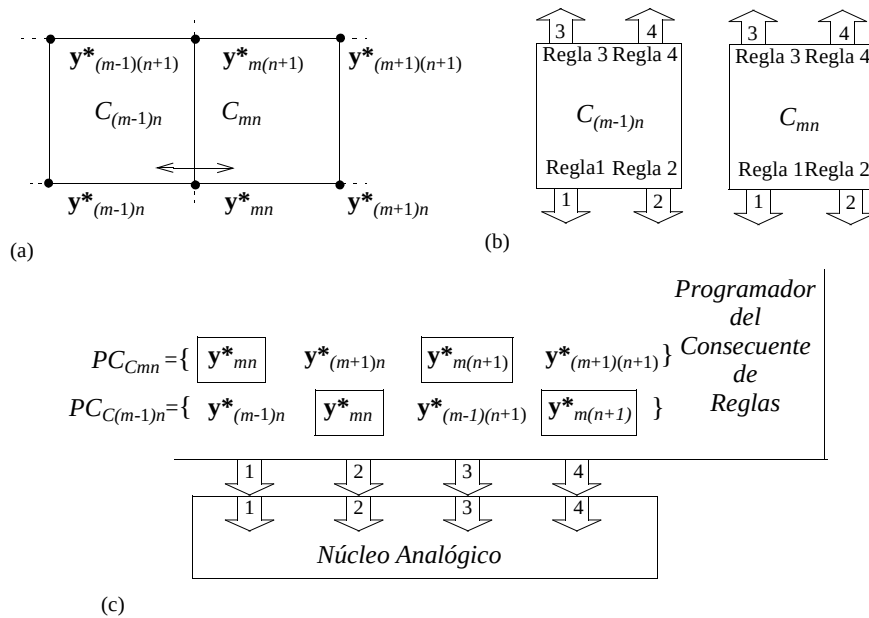


Fig.5.14 Ejemplo de interfaz Núcleo Analógico - Programador del Consecuente de Reglas.

Por su parte, la segunda opción mantiene intacta la estructura del *Núcleo Analógico*, aunque exige cierta elaboración en el diseño del bloque *Programador de Consecuente de Reglas*, para reordenar los valores *singleton*.

Dado que preservar el procesamiento del *Núcleo Analógico* es uno de los requerimientos globales de diseño, esta segunda opción es la adoptada en esta propuesta y la desarrollada en el diseño del controlador ejemplo.

### 3.1.1.2. Tamaño, distribución de datos y forma de acceso

Consideremos a continuación, y de forma conjunta, las cuestiones del tamaño, la distribución de datos y la forma de acceso al sistema de almacenamiento de los valores *singleton*.

En el ejemplo que se muestra en la Fig.5.14 se ha resaltado el hecho de que los dos intervalos adyacentes considerados,  $C_{(m-1)n}$  y  $C_{mn}$ , comparten los dos valores *singleton* ( $\mathbf{y}^*_{mn}$  e  $\mathbf{y}^*_{m(n+1)}$ ), de forma que esos dos datos han de poder ser seleccionados simultáneamente por las dos direcciones  $AC_{(m-1)n}$  y  $AC_{mn}$  correspondientes a cada uno de dichos los intervalos. Este resultado puede generalizarse para un sistema de  $M$  entradas y  $L$  etiquetas por entrada, de forma que cada pareja de intervalos adyacentes comparte  $2^{M-1}$  valores *singleton*, que han de poder ser seleccionados simultáneamente a partir de las direcciones de selección correspondientes.

Por tanto, para los accesos de lectura internos el bloque de memoria debe estar organizado de forma que esta circunstancia sea posible, adoptando una distribución de datos y proporcionando un sistema de acceso adecuados, evitando la redundancia en el almacenamiento de información, a fin de mantener el tamaño de la memoria en sus requerimientos mínimos, esto es  $(L^M \times S \text{ bits})$ .

Por otra parte, a fin de que los accesos externos se realicen de la forma más simple posible, esta organización y distribución de datos debe ser transparente en dicho interfaz.

### 3.1.2. Discusión de los requerimientos de temporización de los accesos.

En este apartado analizaremos las señales puestas en juego en cada uno de los dos tipos de accesos, externo e interno, a los valores almacenados en la memoria de este bloque, así como su temporización y las restricciones que plantea.

Desde el punto de vista de los accesos externos el interfaz más sencillo lo constituye un interfaz típico de periférico asíncrono que ha de trabajar en un sistema basado en microprocesador. Este interfaz es el más flexible y permite al controlador ser integrado en

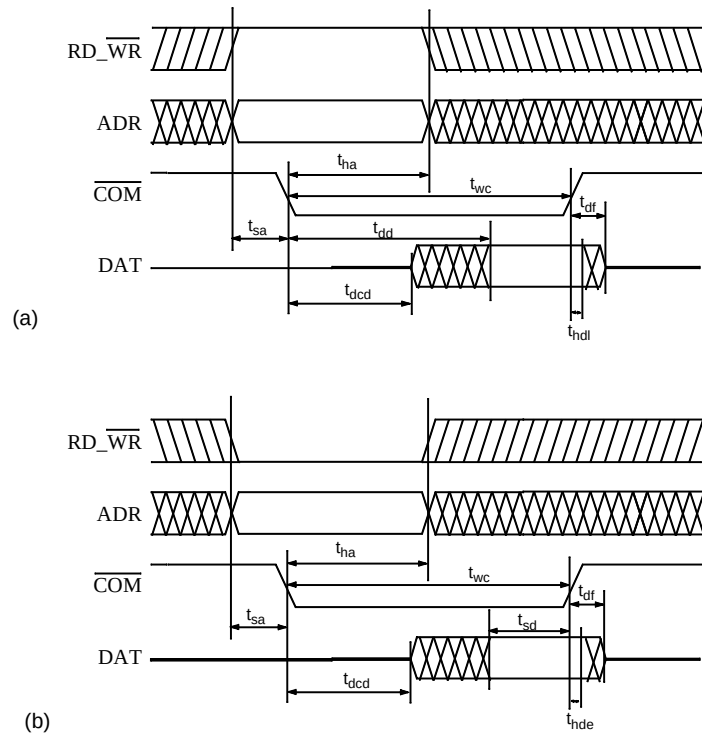


**Fig.5.15 Cronogramas del Interfaz para accesos externos: acceso de lectura (a); acceso de escritura (b);**

diferentes entornos de trabajo, tanto síncronos como asíncronos. La Fig.5.15 muestra un cronograma conceptual de los diferentes accesos tanto de lectura Fig.5.15(a), como escritura Fig.5.15(b). En él aparecen las diferentes señales que intervienen en el interfaz externo, mostrándose además los principales parámetros temporales.

Como puede observarse las señales que componen dicho interfaz son:

- Un bus de direcciones (ADR), con el número de líneas adecuado.
- Una señal de tipo de acceso, (RD\_W $\overline{R}$ ). (1-lógico lectura, 0-lógico escritura)
- Una señal de comando, ( $\overline{COM}$ ). Activa a nivel bajo.
- Y un bus de datos bidireccional (DAT) de S bits.

El significado de los diferentes parámetros temporales que aparecen en la Fig.5.15 se muestran a continuación en la Tabla 5.1. Sobre su valor no hay, en principio requerimientos especiales, forzados por el sistema.

Por lo que respecta al interfaz para accesos de lectura internos éste debe ser un interfaz asíncrono, tal que la memoria se comporte como una tabla de consulta, de forma que por cada

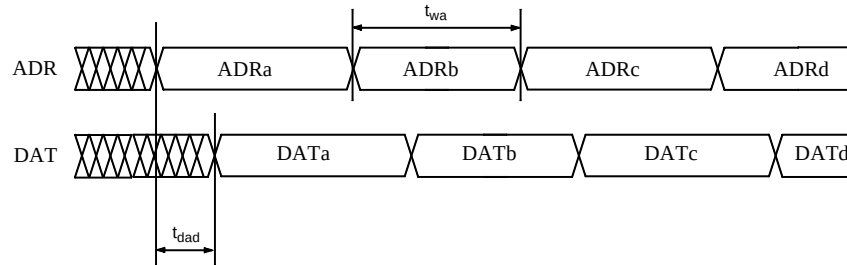


Fig.5.16 Cronograma del Interfaz para accesos internos de lectura.

dirección que aparezca en su línea de selección el sistema proporcione el correspondiente dato a la salida, y de forma que a un cambio de direcciones siga un cambio en el dato.

La Fig.5.16 muestra el cronograma que refleja este comportamiento, junto a las señales puestas en juego. En él se muestran también los principales parámetros temporales a considerar en este tipo de accesos. En la Fig.5.16 con la notación ADR se hace referencia al bus de direcciones  $AC_{k_1 \dots k_M}$  generado en el bloque *Selector de Intervalo*; mientras que DAT representa al bus asociado al conjunto de valores de programación  $PC_{C_{k_1 \dots k_M}} = \{y_1^*, \dots, y_l^*, \dots, y_{2^M}^*\}$ . El parámetro  $t_{dad}$  representa el tiempo de retardo máximo de datos válidos desde un cambio en el bus de direcciones; mientras que el parámetro  $t_{wa}$  representa la duración mínima de una nueva dirección para que pueda ser obtenido un dato válido. El principal requerimientos para estos parámetros temporales es que su valor sea pequeño frente al tiempo de respuesta global del *Núcleo Analógico*.

Tabla 5.1: Descripción de los parámetros temporales del cronograma de la Fig.5.15

| Parámetro | Descripción                                                      |
|-----------|------------------------------------------------------------------|
| $t_{sa}$  | Tiempo de <i>set-up</i> de ADR y RD_WR a comando activo          |
| $t_{ha}$  | Tiempo de <i>hold</i> de ADR y RD_WR desde comando activo        |
| $t_{wc}$  | Duración mínima de comando activo                                |
| $t_{dd}$  | Retardo de DAT valido desde comando activo para lectura          |
| $t_{dcd}$ | Retardo de paso a conducción de DAT desde comando activo         |
| $t_{hdl}$ | Tiempo de <i>hold</i> de DAT desde comando inactivo en lectura   |
| $t_{df}$  | Retardo paso a triestado de DAT desde comando inactivo           |
| $t_{sd}$  | Tiempo de <i>set-up</i> de DAT a comando inactivo                |
| $t_{hde}$ | Tiempo de <i>hold</i> de DAT desde comando inactivo en escritura |

### 3.2. Propuesta de diseño

Para cumplir con todos los requerimientos enunciados con anterioridad, el bloque *Programador de Consecuente de Reglas* se puede diseñar tomando como base diferentes bloques de memoria RAM, a los cuales es necesario proporcionar dos interfases, o puertos de acceso, y dos esquemas diferentes de organización, según la fase de operación del sistema global. Dicha memoria RAM debe poder almacenar el conjunto completo de valores *singleton*, por tanto su tamaño mínimo debe ser de  $L^M$  palabras de  $S$  bits.

Uno de esos dos puertos de acceso se configura para lectura y escritura externas, con su correspondiente línea de direcciones y datos, como la de cualquier periférico asíncrono de un sistema basado en microprocesador, siguiendo los requerimientos descritos en el apartado 3.1.2. de este capítulo y el cronograma de la Fig.5.15.

El otro de los puertos se utiliza para el acceso de lectura interna que ha de proporcionar los valores *singleton* ( $y_1^*, \dots, y_l^*, \dots, y_{2^M}^*$ ) adecuados según el intervalo seleccionado. Para los accesos a este puerto se emplean como señales de selección la palabra de  $M \times i_s(\log_2(L-1))$  bits,  $A_i[0:p]$  con  $(1 \leq i \leq M)$ , proporcionada por el bloque *Selector de Intervalo* y que identifica a cada intervalo  $C_{k_1 \dots k_M}$  en el espacio de entrada. Cada acceso de este tipo proporciona un conjunto de parámetros de programación  $PC_{C_{k_1 \dots k_M}}$  en forma de una palabra ordenada de  $2^M \times S$  bits, donde cada grupo de  $S$  bits corresponde a un valor *singleton* ( $y_1^*, \dots, y_l^*, \dots, y_{2^M}^*$ ); siguiendo los requerimientos descritos en el apartado 3.1.2. de este capítulo y el cronograma de la Fig.5.16.

Conviene resaltar de nuevo que la posición que ocupan los valores *singleton* en esa palabra viene impuesta por los requerimientos de programación del *Núcleo Analógico*, como consecuencia del objetivo global de diseño de penalizar lo menos posible el procesado que en él se realiza, según se ha discutido en la sección 3.1.1. y siguiendo el ejemplo de la Fig.5.14, y que hace que la acción de reordenamiento de reglas recaiga en el bloque *Programador de Consecuente de Reglas*.

Una posible estrategia para realizar el diseño de este bloque y obtener simultáneamente todos los valores, consiste en almacenar una palabra de  $2^M \times S$  bits que es necesario proporcionar para cada intervalo con los valores *singleton* en su posición correcta. Sin embargo, esto implica replicar cada valor *singleton* tantas veces como intervalos de los que forme parte haya, esto es  $2^M$ . El tamaño de la memoria requerida en este caso será de  $(L-1)^M$  palabras de  $2^M \times S$  bits, esto es  $2^M \cdot (1-1/L)^M$  veces mayor que la mínima necesaria, y que resulta tanto mayor cuanto mayores sean los valores de  $M$  y  $L$ . Nótese que para el controlador ejemplo, donde  $M = 2$  y  $L = 8$ , el tamaño de esta memoria resulta entre tres o cuatro veces el tamaño mínimo necesario. Por tanto, un almacenamiento eficiente de los valores *singleton* exige que estos no sean almacenados de forma redundante, por lo que el

diseño del sistema debe contemplar esta situación.

Otra alternativa consiste en realizar, de forma secuencial, diferentes accesos consecutivos, hasta formar la palabra de programación. Esta opción tiene que descartarse porque se ha planteado un diseño con funcionamiento en tiempo continuo, por lo que el acceso de lectura ha de proporcionar simultáneamente todos los elementos del conjunto  $PC_{C_{k_1 \dots k_M}}$ , siguiendo los cambios de intervalo y por tanto de forma asíncrona.

Una mejor estrategia consiste en organizar apropiadamente los valores *singleton* distribuyéndolos en distintos bloques de memoria, de manera que no coincidan en el mismo bloque datos que tengan que ser utilizados simultáneamente; o lo que es lo mismo, que para cada dirección correspondiente a un intervalo  $C_{k_1 \dots k_M}$  cada uno de los  $2^M$  valores del conjunto  $PC_{C_{k_1 \dots k_M}}$  tenga que ser leído de un bloque distinto de memoria. Un multiplexor a la salida, adecuadamente controlado, puede reordenar los datos obtenidos de cada bloque para formar el correspondiente conjunto de programación ordenado. Esta es la estrategia propuesta para el diseño de este bloque.

La Fig.5.17 muestra una arquitectura genérica para el bloque *Programador de Consecuente de Reglas*, que cumple con los requerimientos de funcionamiento interno indicados con anterioridad. En ella se muestra como los  $L^M$  valores *singleton* son distribuidos en  $2^M$  bloques de memoria, tantos como reglas activas hay en cada intervalo, de forma que cada bloque almacena en promedio  $L^M/2^M$  valores que no necesitan ser direccionados simultáneamente.

El bloque *Generador de Direcciones* ha de ser un sistema combinacional que proporcione, para cada dirección  $A_i[0:p] \text{ con } (1 \leq i \leq M)$  del intervalo  $C_{k_1 \dots k_M}$ , los  $2^M$  conjuntos de direcciones de  $i_s(\log_2(L^M/2^M))$  bits que permiten localizar a cada uno de los elementos del conjunto  $PC_{C_{k_1 \dots k_M}}$  en su correspondiente bloque de memoria. Al mismo tiempo el bloque *Control del Multiplexor*, que también ha de ser combinacional, permitirá colocar cada uno de los datos seleccionados en la posición correcta para formar la palabra de  $2^M \times S$  bits de salida, y todo ello en un sólo paso.

Nótese que el acceso de lectura debe ser completamente asíncrono, y que el nuevo dato aparece a la salida como consecuencia de un cambio en las líneas de direcciones, siguiendo el cronograma de la Fig.5.16, de esta manera se tendrá el comportamiento en tiempo continuo deseado para el sistema global. Este requerimiento condiciona el tipo de memoria RAM que ha de ser empleado.

Por otra parte, la distribución de los datos en los bloques de memoria puede realizarse de distintas maneras para cumplir con la condición de que no coincidan en el mismo bloque datos que tengan que ser utilizados simultáneamente. Una buena distribución de datos debe optimizar el tamaño y la ocupación de los diferentes bloques de memoria y tener en cuenta las direcciones

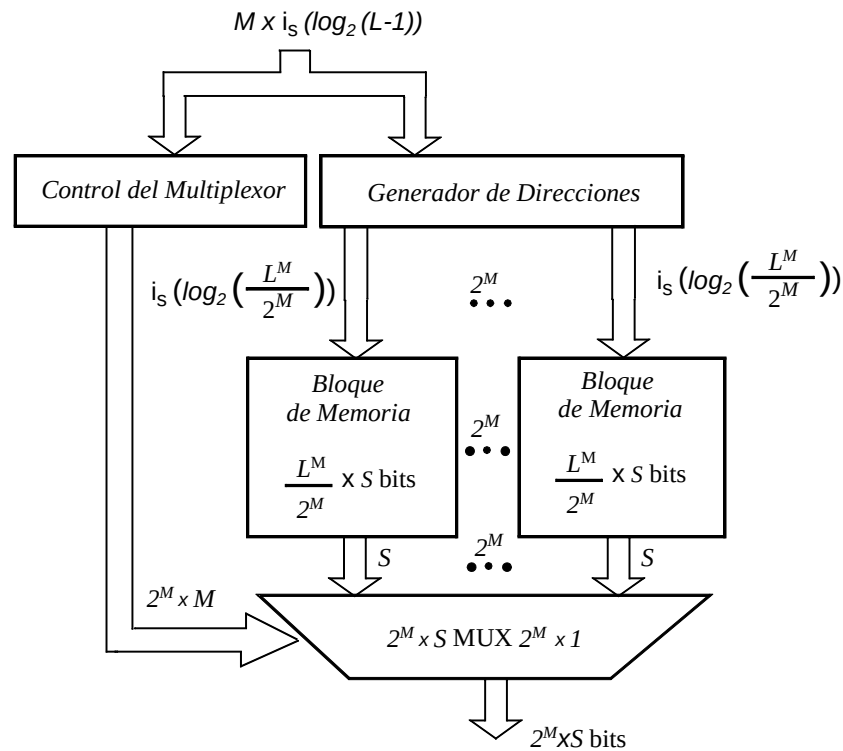


**Fig.5.17** Arquitectura genérica para el bloque Programador de Consecuente de Reglas.

asignadas a los datos en cada bloque con el fin de simplificar el diseño de los bloques combinacionales *Generación de Direcciones* y *Control del Multiplexor*. En general la distribución dependerá de los valores de  $L$  y  $M$ , resultando distribuciones más uniformes para el caso en que  $M$  sea par y  $L$  sea potencia de dos.

Para ilustrar la distribución de datos en los diferentes bloques de memoria, que resulta ventajoso en el caso especial antes mencionado, consideremos el ejemplo de cuatro intervalos adyacentes en un sistema bidimensional como los que se muestran en la Fig.5.18. En este caso se tienen cuatro reglas activas en cada intervalo, y por lo tanto se han de emplear cuatro bloques de memoria. En la Fig.5.18 cada regla es representada por el valor *singleton* de su consecuente  $y_{mn}^*$ , donde los índices  $m$  y  $n$ , toman valores enteros comprendidos entre 1 y  $L$ .

Según el carácter par o impar de cada uno de esos dos índices se tendrá que cada una de esas reglas puede ser clasificada en uno de los siguientes cuatro grupos:

- $m$  par y  $n$  par.
- $m$  impar y  $n$  par.
- $m$  par y  $n$  impar.
- $m$  impar y  $n$  impar.

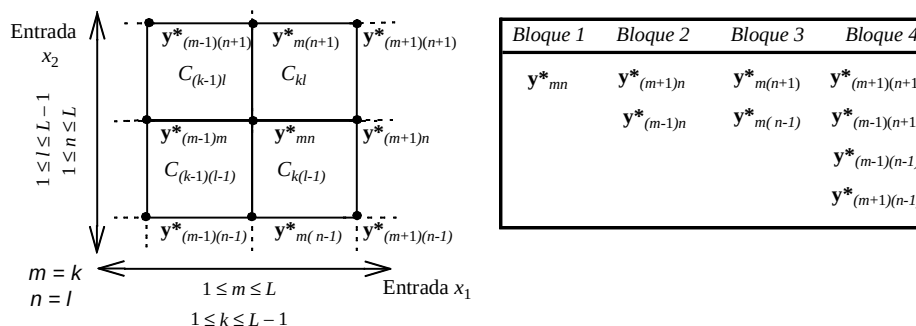


**Fig.5.18 Ejemplo de distribución de datos entre los diferentes bloques de memoria.**

Este criterio puede ser adoptado también para distribuir los correspondientes valores *singleton* en los bloques de memoria. Obsérvese que ninguna de las reglas activas en el intervalo  $C_{mn}$ , ( $k = m$  y  $l = n$ ) puede pertenecer a dos de los citados grupos. Por ejemplo si  $y^*_{mn}$  pertenece al grupo a),  $y^*_{(m+1)n}$  pertenecerá al b),  $y^*_{m(n+1)}$  al c) y  $y^*_{(m+1)(n+1)}$  al d).

Esta forma de distribución puede extenderse al caso general de  $M$  entradas. En este caso cada regla y por tanto cada *singleton* de la forma  $y^*_{j_1 \dots j_M}$  podrá ser clasificado en uno de los  $2^M$  grupos que pueden establecerse atendiendo a la paridad de los  $M$  subíndices.

Por otra parte, nótese que en el esquema de la Fig.5.17 no se ha contemplado el interfaz externo que ha de estar presente para cumplir con el requerimiento de programación y que por ser éste dependiente de la realización particular que se lleve a cabo, será considerado en el apartado dedicado al diseño concreto del controlador ejemplo.

Finalmente, conviene observar que la necesidad de un doble interfaz de acceso, externo e interno, no implica la necesidad de utilizar bloques de memoria de tipo multipuerto, ya que, en general, las distintas fases del funcionamiento del sistema global no van a ser simultáneas. Sin embargo, una estructura multipuerto podría ser necesaria pensando en un sistema neuro-difuso, que fuese capaz de adaptar su configuración sin interrumpir su acción de control, ejecutando *on-line* algún algoritmo de aprendizaje compatible con el *hardware*. En este caso es posible pensar en la necesidad de accesos simultáneos a memoria desde diferentes puertos.



### 3.3. Diseño CMOS del bloque Programador de Consecuente de Reglas

En el controlador ejemplo que ha sido fabricado se ha seguido una organización un poco distinta a la discutida anteriormente, y ha sido diseñado como un bloque de memoria único y a medida, que proporciona la capacidad de almacenamiento de datos necesaria y los sistemas de acceso apropiados. Sin tratar de ser un nuevo modelo o propuesta en el competitivo campo del diseño de memorias, sí proporciona ciertas simplificaciones en cuanto al control en la accesibilidad de los datos almacenados, y supone además una experiencia de diseño *full-custom*. Por ello, aunque en primer lugar se describe su arquitectura básica de la forma más general posible, a modo de propuesta genérica alternativa a la del apartado anterior, las posteriores descripciones del diseño de los diferentes elementos que intervienen en dicha arquitectura son abordadas desde un punto de vista más particular y haciendo siempre referencia al mencionado controlador ejemplo.

En la Fig.5.19 se muestra la arquitectura genérica para el bloque *Programador del Consecuente de Regla* que ha servido de base para este diseño. En ella se muestra como los  $L^M$  valores *singleton* son distribuidos en  $2^{M-1}$  bloques de celdas de memoria, de  $L^{M-1}/2^{M-1}$

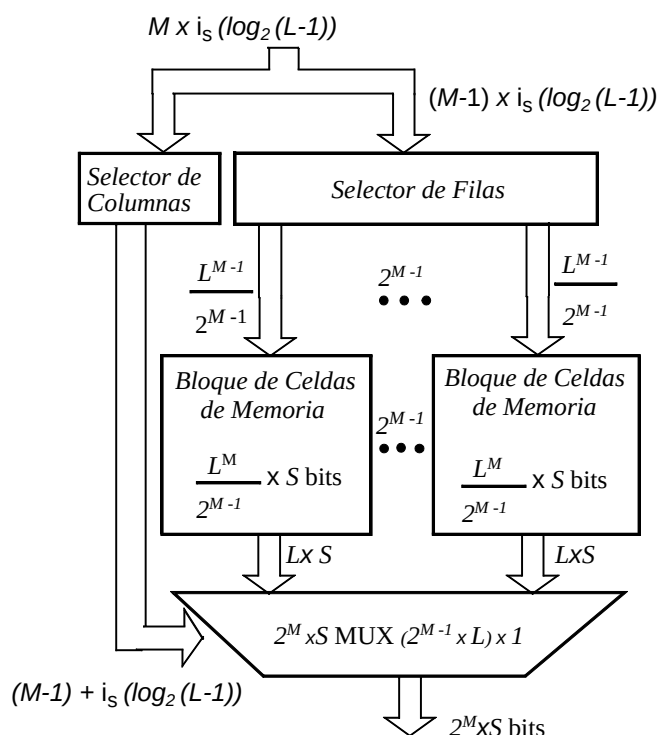


Fig.5.19 Arquitectura genérica para el bloque Programador de Consecuente de Reglas realizada en el controlador ejemplo.

filas y  $L$  columnas. En cada uno de estos bloques se almacenan parejas de valores *singleton* de manera que no coincidan en el mismo bloque parejas que han de ser direccionadas simultáneamente. Durante un acceso de lectura interna el *Selector de Filas* selecciona  $2^{M-1}$  filas simultáneamente, una por cada bloque de celdas de memoria, para obtener así un conjunto de  $L \times 2^{M-1}$  valores *singleton*. El *Selector de Columnas* selecciona los  $2^M$  valores *singleton* correctos a partir de este conjunto, y controla el multiplexor para colocarlos en la posición adecuada para formar la palabra de  $2^M \times S$  bits de salida, todo en un sólo paso.

El bloque *Selector de Fila* emplea  $(M-1) \times i_s(\log_2(L-1))$  bits de la dirección  $A_i[0:p]$  con  $(1 \leq i \leq M)$  del intervalo  $C_{k_1 \dots k_M}$ , para realizar su trabajo de selección generando  $2^{M-1}$  conjuntos de  $L^{M-1}/2^{M-1}$  líneas de selección cada uno; mientras que los  $M \times i_s(\log_2(L-1))$  bits restantes son utilizados por el *Selector de Columna* para generar las  $(M-1) + i_s(\log_2(L-1))$  líneas que controlan el multiplexor de salida.

Para accesos de programación externos, todos los bloques de celdas de memoria son reconfigurados de manera que en conjunto se comportan como una memoria RAM convencional. Esta situación será descrita más adelante cuando se estudie el caso concreto del controlador ejemplo y no se muestra en la Fig.5.19.

La principal ventaja de esta arquitectura, frente a la propuesta genérica que se muestra en la Fig.5.17 es que, como se verá en el ejemplo de diseño, permite una fácil reorganización de los datos almacenados, según el origen externo o interno de los accesos, que da lugar a una estructura de almacenamiento muy compacta y que ayuda a simplificar el diseño de los bloques de selección: *Selector de Columnas*, y *Selector de filas*.

Para ilustrar la distribución de datos en los diferentes bloques de *Celdas de Memoria* y seguir el funcionamiento en conjunto del bloque *Programador del Consecuente de Regla* durante un acceso a varias direcciones genéricas adyacentes podemos analizar la situación para un caso bidimensional, ( $M = 2$ ), como se muestra en la Fig.5.20.

En la Fig.5.20(a) se muestran cuatro intervalos genéricos adyacentes  $(C_{mn}, C_{m(n-1)}, C_{(m-1)n}, C_{(m-1)(n-1)})$  con sus correspondientes valores *singleton* asociados. Esos datos se han distribuido en  $2^{M-1} = 2$  bloques de *Celdas de Memoria* como muestra la Fig.5.20(b).

El bloque de *Celdas de Memoria 1* contiene las filas con el índice  $n$  impar, mientras que el bloque de *Celdas de Memoria 2* contiene las filas con índice  $n$  par. Cada uno de los bloques tiene  $L$  columnas. En la Fig.5.20(c), se muestra el resultado de la selección realizada por el *Selector de Filas* para cada uno de los cuatro intervalos de la Fig.5.20(a). Como puede observarse, para cada dirección asociada a un intervalo, son seleccionadas dos filas simultáneamente, una en cada uno de los bloques de *Celdas de Memoria*; así para los intervalos de índice  $n$ ,  $(C_{mn}, C_{(m-1)n})$ , son seleccionadas las filas  $n$  y  $n+1$ , mientras que para los

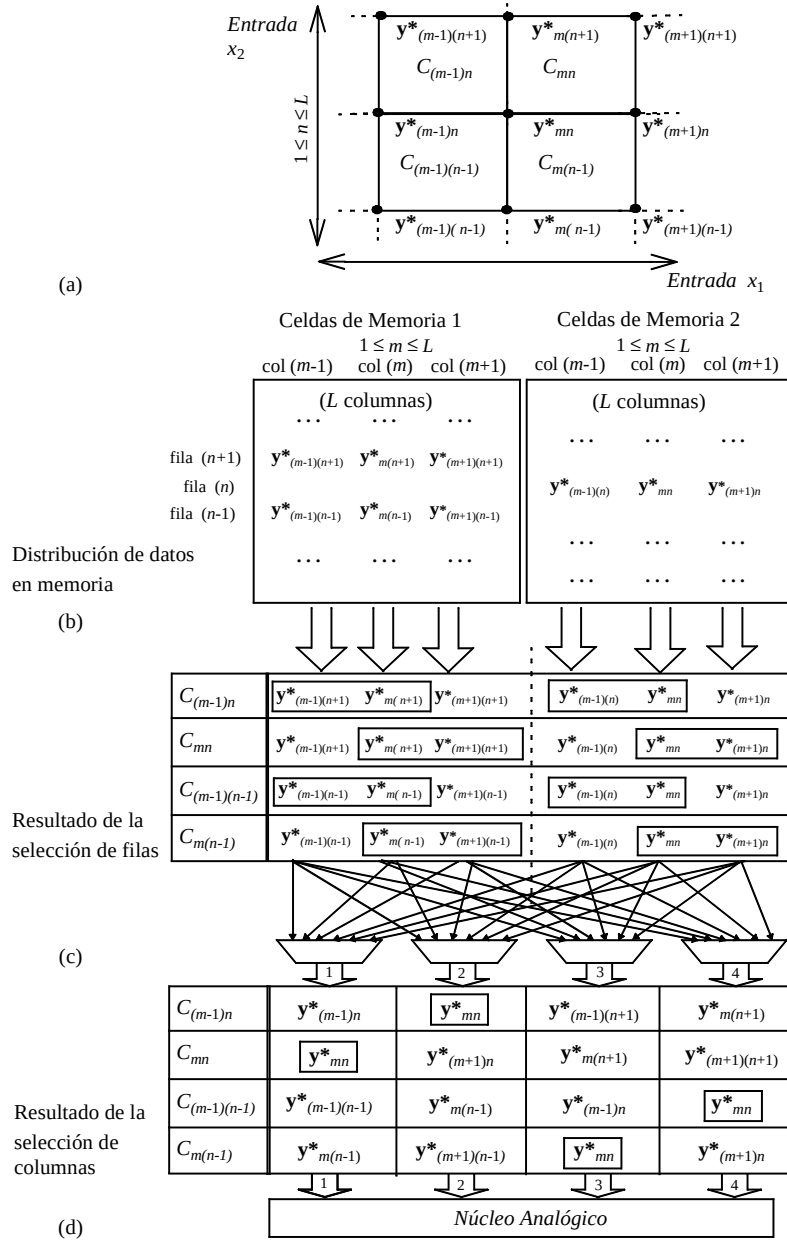


**Fig.5.20 Ejemplo de operación del Bloque de Programador de Consecuente de Reglas en el controlador ejemplo:** Esquema de cuatro intervalos genéricos adyacentes (a); esquema de la distribución de valores *singleton* en celdas de memoria (b); resultado de la selección de filas para cada uno de los cuatro intervalos(c); resultado de la selección de columnas para cada uno de los cuatro intervalos (d).

intervalos de índice  $n-1$ ,  $(C_{m(n-1)}, C_{(m-1)(n-1)})$ , son seleccionadas las filas  $n-1$  y  $n$ . En ambos casos todos los valores *singleton* necesarios para programar el *Núcleo Analógico* (marcados con un recuadro en la figura) están en alguna de las líneas de columna como resultado de la selección por filas.

Por último el *Selector de Columnas*, selecciona de entre ese conjunto, los valores correctos en cada caso y los coloca adecuadamente en el bus de salida, como puede verse en la Fig.5.20(c). Obsérvese que cada valor *singleton* aparece en su posición correcta como exige el interfaz con el *Núcleo Analógico*.

A continuación se analizan y caracterizan los diferentes elementos que constituyen este bloque y su realización mediante tecnologías CMOS estándar, siguiendo la organización de memoria aquí presentada y particularizando para el caso del controlador ejemplo.

### 3.3.1. Organización general del Bloque Programador de Consecuente de Reglas en el controlador ejemplo.

En la Tabla 5.2 se resumen las principales cifras que resultan de particularizar la arquitectura mostrada en la Fig.5.19 para el controlador ejemplo, y que son empleadas en las descripciones de los bloques en las siguientes secciones.

**Tabla 5.2:** Cifras de la arquitectura del bloque Programador de Consecuente de Reglas

| Nº de Entradas                                                                                    | Nº de Etiquetas por entrada                             | Nº total de Reglas                | Nº de bits por <i>singleton</i>                                                   | Nº de Reglas Activas por intervalo                                  |
|---------------------------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------|
| $M = 2$                                                                                           | $L = 8$                                                 | $L^M = 64$                        | $S = 4$                                                                           | $2^M = 4$                                                           |
| Nº de bloques de celdas de memoria                                                                | Tamaño de cada bloque de celdas                         | Tamaño de la palabra de selección |                                                                                   | Selector de columnas Entradas/Salidas                               |
| $2^{M-1} = 2$                                                                                     | $\frac{L^{M-1}}{2^{M-1}} = 4$ filas<br>$L = 8$ columnas | $M \times i_s (\log_2(L-1)) = 6$  |                                                                                   | $M \times i_s (\log_2(L-1)) = 6$<br>$(M-1) + i_s (\log_2(L-1)) = 4$ |
| Selector de filas Entradas/Salidas                                                                |                                                         |                                   | Tamaño del multiplexor de columnas                                                |                                                                     |
| $(M-1) \times i_s (\log_2(L-1)) = 3$<br>$2^{M-1} = 2$ grupos $\frac{L^{M-1}}{2^{M-1}} = 4$ líneas |                                                         |                                   | $2^M \times S = 16$ Multiplexores<br>$2^{M-1} \times L = 16$ entradas<br>1 salida |                                                                     |

Por su parte, en la Fig.5.21 se muestra un esquema de cómo son vistos los valores *singleton* almacenados en los bloques de *Celdas de memoria*, según el origen, externo o interno de los accesos en el controlador ejemplo. Para accesos internos estos datos son vistos como si estuviesen distribuidos en dos bloques de *Celdas de Memoria* de 32 palabras de 4 bits cada una, y organizados según una matriz de celdas básicas de 4 filas y 8 columnas, según muestra la Fig.5.21(b), que resulta más cómodo para facilitar el diseño de las funciones de decodificación internas. Sin embargo, para accesos externos esos dos bloques se reorganizan y son vistos como un bloque formado por una matriz de 8 filas por 8 columnas, según muestra la Fig.5.21(a), y que resulta más adecuada para facilitar la programación desde el punto de vista de los accesos de un usuario externo.

Distribución de Datos para accesos externos

|        | Col.1      | Col.2      | Col.3      | Col.4      | Col.5      | Col.6      | Col.7      | Col.8      |
|--------|------------|------------|------------|------------|------------|------------|------------|------------|
| Fila 1 | $Y_{11}^*$ | $Y_{21}^*$ | $Y_{31}^*$ | $Y_{41}^*$ | $Y_{51}^*$ | $Y_{61}^*$ | $Y_{71}^*$ | $Y_{81}^*$ |
| Fila2  | $Y_{12}^*$ | $Y_{22}^*$ | $Y_{32}^*$ | $Y_{42}^*$ | $Y_{52}^*$ | $Y_{62}^*$ | $Y_{72}^*$ | $Y_{82}^*$ |
| Fila3  | $Y_{13}^*$ | $Y_{23}^*$ | $Y_{33}^*$ | $Y_{43}^*$ | $Y_{53}^*$ | $Y_{63}^*$ | $Y_{73}^*$ | $Y_{83}^*$ |
| Fila4  | $Y_{14}^*$ | $Y_{24}^*$ | $Y_{34}^*$ | $Y_{44}^*$ | $Y_{54}^*$ | $Y_{64}^*$ | $Y_{74}^*$ | $Y_{84}^*$ |
| Fila5  | $Y_{15}^*$ | $Y_{25}^*$ | $Y_{35}^*$ | $Y_{45}^*$ | $Y_{55}^*$ | $Y_{65}^*$ | $Y_{75}^*$ | $Y_{85}^*$ |
| Fila6  | $Y_{16}^*$ | $Y_{26}^*$ | $Y_{36}^*$ | $Y_{46}^*$ | $Y_{56}^*$ | $Y_{66}^*$ | $Y_{76}^*$ | $Y_{86}^*$ |
| Fila7  | $Y_{17}^*$ | $Y_{27}^*$ | $Y_{37}^*$ | $Y_{47}^*$ | $Y_{57}^*$ | $Y_{67}^*$ | $Y_{77}^*$ | $Y_{87}^*$ |
| Fila8  | $Y_{18}^*$ | $Y_{28}^*$ | $Y_{38}^*$ | $Y_{48}^*$ | $Y_{58}^*$ | $Y_{68}^*$ | $Y_{78}^*$ | $Y_{88}^*$ |

(a)

Distribución de Datos para accesos internos

|        | Col.1      | Col.2      | Col.3      | Col.4      | Col.5      | Col.6      | Col.7      | Col.8      | Col.9      | Col.10     | Col.11     | Col.12     | Col.13     | Col.14     | Col.15     | Col.16     |
|--------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Fila 1 | $Y_{11}^*$ | $Y_{21}^*$ | $Y_{31}^*$ | $Y_{41}^*$ | $Y_{51}^*$ | $Y_{61}^*$ | $Y_{71}^*$ | $Y_{81}^*$ | $Y_{12}^*$ | $Y_{22}^*$ | $Y_{32}^*$ | $Y_{42}^*$ | $Y_{52}^*$ | $Y_{62}^*$ | $Y_{72}^*$ | $Y_{82}^*$ |
| Fila2  |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Fila3  | $Y_{13}^*$ | $Y_{23}^*$ | $Y_{33}^*$ | $Y_{43}^*$ | $Y_{53}^*$ | $Y_{63}^*$ | $Y_{73}^*$ | $Y_{83}^*$ | $Y_{14}^*$ | $Y_{24}^*$ | $Y_{34}^*$ | $Y_{44}^*$ | $Y_{54}^*$ | $Y_{64}^*$ | $Y_{74}^*$ | $Y_{84}^*$ |
| Fila4  |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Fila5  | $Y_{15}^*$ | $Y_{25}^*$ | $Y_{35}^*$ | $Y_{45}^*$ | $Y_{55}^*$ | $Y_{65}^*$ | $Y_{75}^*$ | $Y_{85}^*$ | $Y_{16}^*$ | $Y_{26}^*$ | $Y_{36}^*$ | $Y_{46}^*$ | $Y_{56}^*$ | $Y_{66}^*$ | $Y_{76}^*$ | $Y_{86}^*$ |
| Fila6  |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |
| Fila7  | $Y_{17}^*$ | $Y_{27}^*$ | $Y_{37}^*$ | $Y_{47}^*$ | $Y_{57}^*$ | $Y_{67}^*$ | $Y_{77}^*$ | $Y_{87}^*$ | $Y_{18}^*$ | $Y_{28}^*$ | $Y_{38}^*$ | $Y_{48}^*$ | $Y_{58}^*$ | $Y_{68}^*$ | $Y_{78}^*$ | $Y_{88}^*$ |
| Fila8  |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |

Bloque de Celdas de Memoria 1

Bloque de Celdas de Memoria 2

(b)

Fig.5.21 Distribución de datos en el controlador ejemplo, según es vista desde los diferentes orígenes de acceso: accesos externos (a); e internos (b).

En la Fig.5.22 se muestra un esquema en el que aparecen los distintos bloques de diseño que constituyen el bloque *Programador de Consecuente de Reglas*, realizado en el controlador ejemplo. Estos bloques son:

- Bloque de Celdas de Memoria
- Selector de Filas
- Selector de Columnas para accesos externos

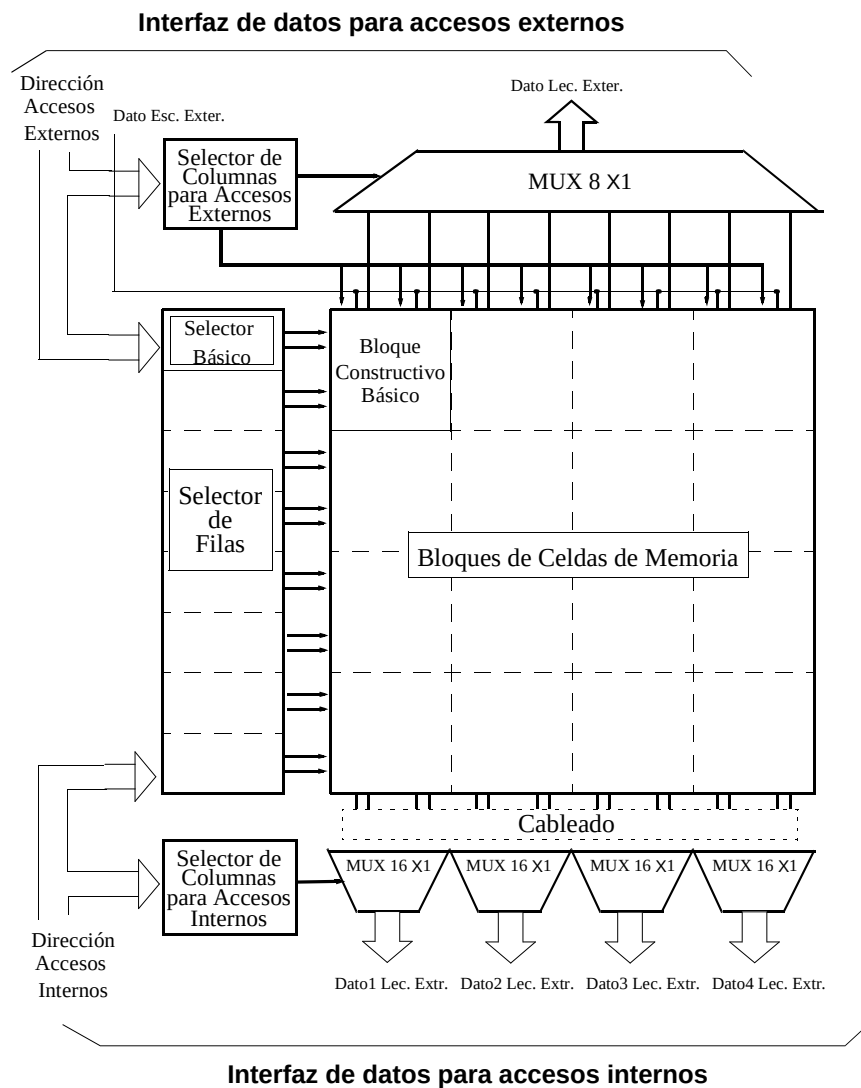


Fig.5.22 Esquema de los bloques constitutivos del *Bloque Programador de Consecuente de Reglas*, indicativo del *floorplan* realizado.

- Selector de Columnas para accesos internos

La disposición de los diferentes bloques en ese esquema guarda relación directa con el *floorplan* y el *layout* realizado. En la Fig.5.23 se muestra un detalle de ese *layout*, en el que se recogen las celdas de memoria correspondientes a uno de los bits de cada valor *singleton*, junto a la lógica de selección de fila y multiplexación de columnas en ambas interfases de acceso.

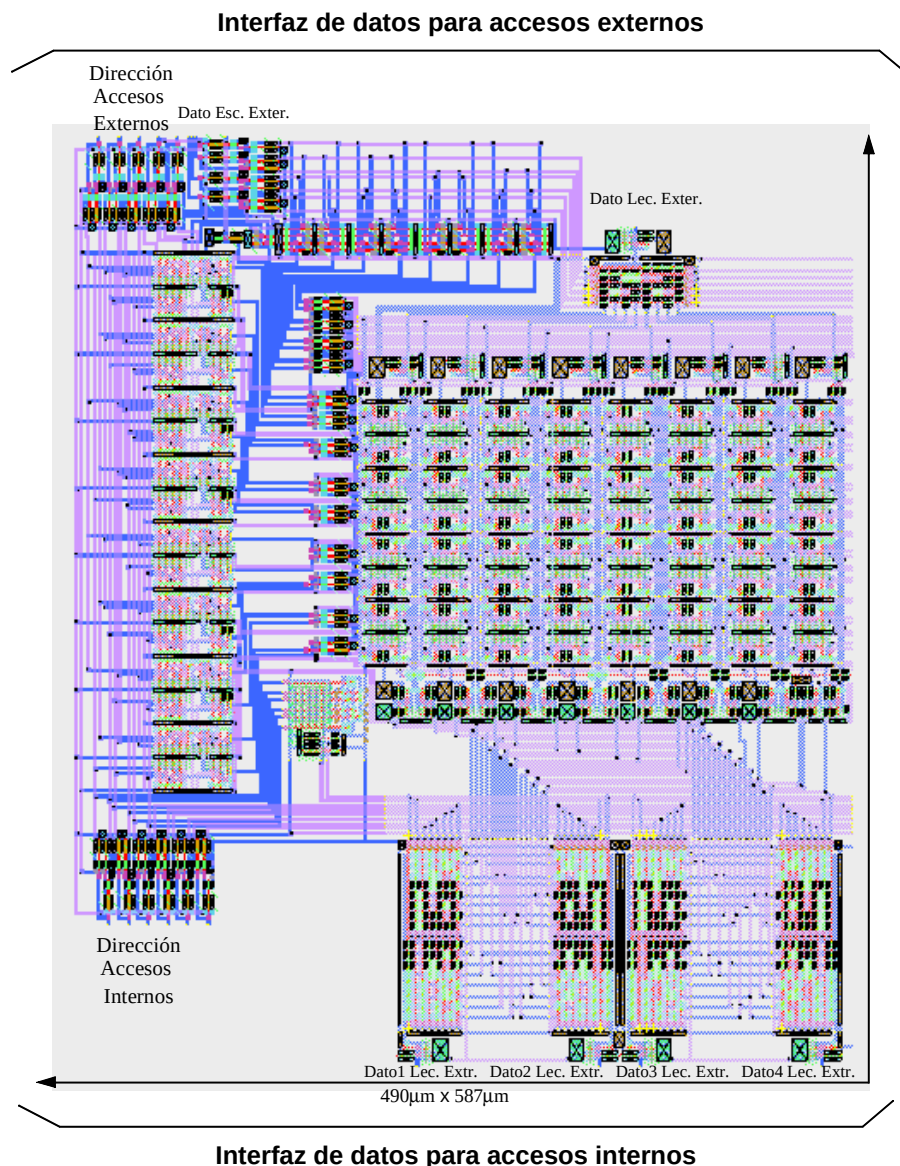


Fig.5.23 Detalle de *layout*: Celdas de memoria para un bit del *singleton* y circuitería de selección.

Los tres bits adicionales que constituyen un valor *singleton* completo se realizan replicando el bloque de celdas de memoria y los multiplexores de columnas, todos ellos controlados a partir de las mismas líneas de selección y control. A continuación se describe el diseño de cada uno de los bloques que aparecen en la Fig.5.22.

### 3.3.2. Diseño del Bloque de Celdas de Memoria

Como ponen de manifiesto las cifras de la Tabla 5.2, el tamaño global de memoria necesario en este diseño es de sesenta y cuatro palabras de cuatro bits. Para accesos internos estos datos son distribuidos en dos bloques de *Celdas de Memoria* de treinta y dos palabras de cuatro bits cada una, y organizados según una matriz de celdas básicas de cuatro filas y ocho columnas. Sin embargo, para accesos externos esos dos bloques se reorganizan y son vistos como una matriz de ocho filas por ocho columnas, que resulta más adecuada para facilitar el diseño de las funciones de decodificación. Esta organización es la que se muestra en la Fig.5.21.

El diseño del los *Bloques de Celdas de Memoria* se ha realizado a partir del bloque constructivo básico que se describe a continuación.

#### 3.3.2.1. Bloque constructivo básico

El bloque constructivo empleado para realizar la matriz de celdas de memoria, agrupa cuatro celdas básicas para un bit de memoria, según muestra la Fig.5.24. Esta agrupación facilita la reorganización de la matriz de celdas de memoria en función del origen y tipo de los accesos, favorece el empaquetado de celdas, y simplifica el *layout* y el *routing*; con el consiguiente ahorro, tanto en área, como en simplicidad del diseño de la lógica de selección.

Siguiendo el esquema de la Fig.5.20, en la Fig.5.24 la fila  $n$ , formada por las celdas  $(m,n)$  y  $(m+1,n)$ , pertenece al *Bloque de Celdas de Memoria 2*, mientras que la fila  $n+1$ , formada por las celdas  $(m,n+1)$  y  $(m+1,n+1)$ , pertenece al *Bloque de Celdas de Memoria 1*.

Durante la **fase de actividad**, cada una de estas celdas aparece como perteneciente a columnas de lectura independientes en el **Interfaz para Accesos Internos** como se muestra en la parte inferior de la Fig.5.24, lo que con la adecuada selección de filas permite acceder simultáneamente a las cuatro celdas.

Durante la **fase de configuración** las líneas de lectura de las celdas  $(m,n)$  y  $(m,n+1)$ , por un lado y las de las celdas  $(m+1,n)$  y  $(m+1,n+1)$  por otro, son conectadas cerrando los *switches*  $s_m$  y  $s_{m+1}$  respectivamente, pasando a formar parte de una misma columna, columnas  $m$  y  $m+1$  respectivamente, para el **Interfaz de Accesos Externos**, como ya lo son para la línea de escritura, como se muestra en la parte superior de la Fig.5.24.

La Fig.5.25 muestra un detalle del layout de este bloque constructivo que se corresponde



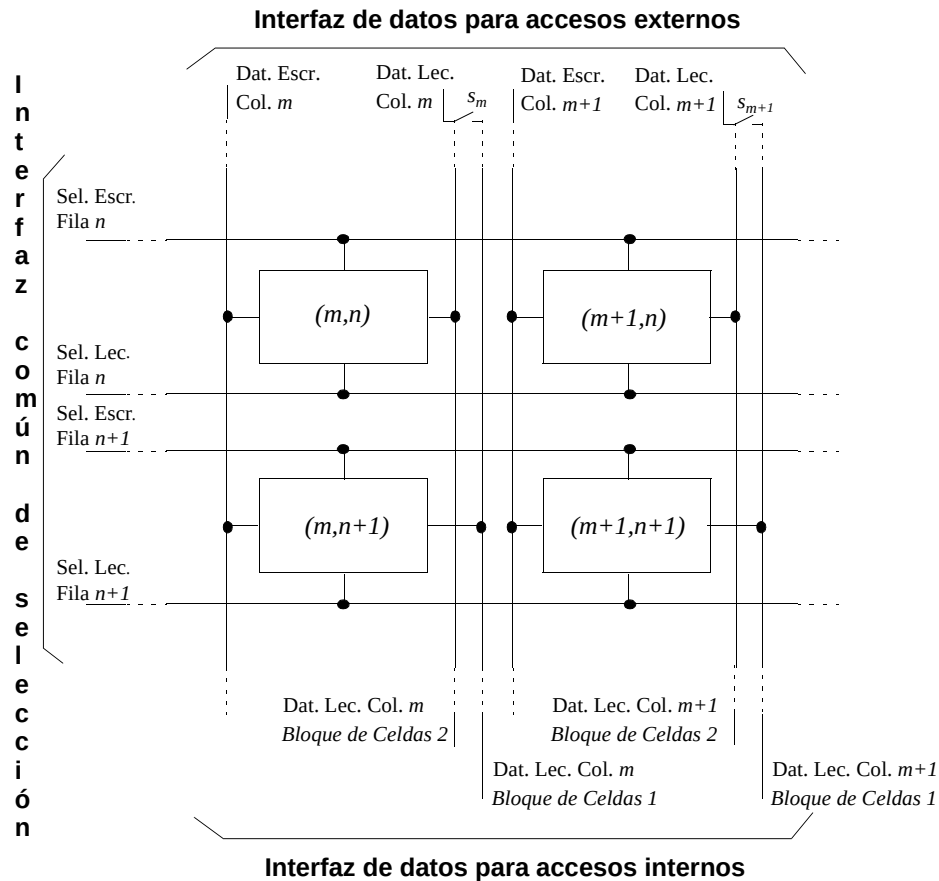


**Fig.5.24** Bloque constructivo del los Bloques de Celdas de Memoria uno y dos.

perfectamente con el esquema de la Fig.5.24.

### 3.3.2.2. Celda básica de memoria

Como celda básica de memoria se ha empleado una de celda de memoria estática, del tipo de las denominadas *single-ended bit-line* [Shin91], que, a diferencia de las celdas de memoria estática CMOS más usuales, posee líneas de bit de un solo terminal. Esta celda es empleada en el diseño de *Register Files*. Este último término hace referencia a estructuras de memoria RAM de muy corto tiempo de acceso y con múltiples puertos de lectura y escritura. Las celdas básicas de memoria reportadas para estos *Register Files* proporcionan suficiente flexibilidad para realizar muy diferentes configuraciones de sistemas multipuerto, tanto en la organización del

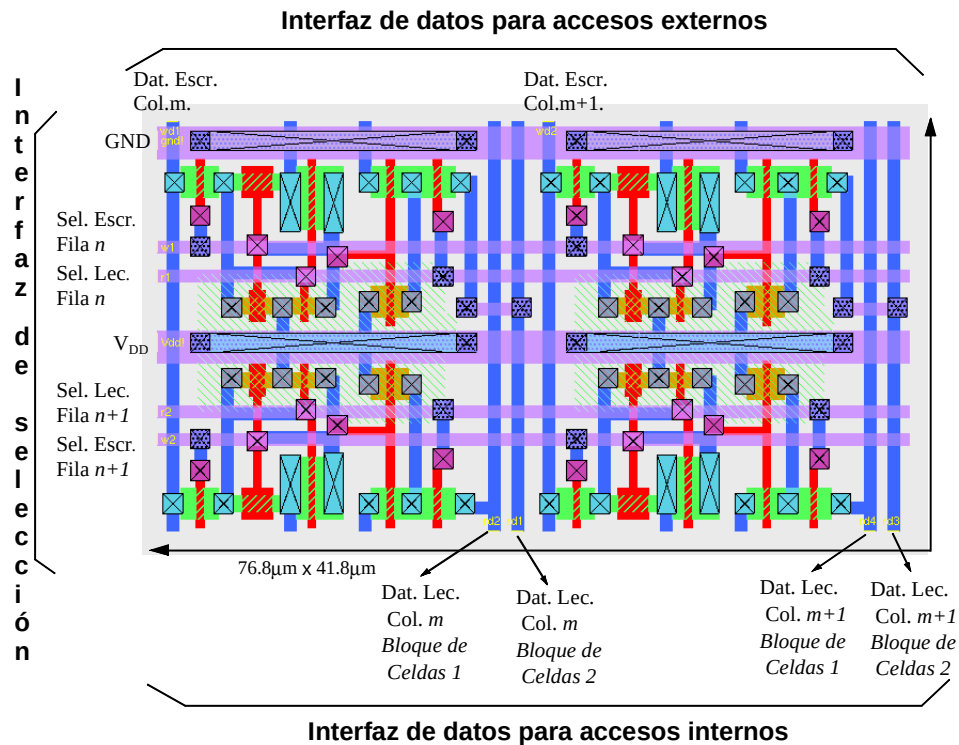


**Fig.5.25 Detalle de layout del bloque constructivo básico de celdas de memoria que incluye cuatro palabras de un bit.**

layout como en la organización de los puertos de acceso, además de ciertas ventajas adicionales de robustez y muy bajo consumo de potencia en dc [West94].

En la Fig.5.26(a) se muestra el esquema de la celda básica de memoria estática de un solo terminal por línea de bit empleada en este diseño; mientras que la Fig.5.26(b) muestra el esquema de la celda de memoria estática de seis transistores, habitual en el diseño de memorias CMOS, que emplea dos líneas de bit complementarias. Como puede suponerse de la comparación de la estructura de cada celda según se muestra en dicha figura, la celda de memoria seleccionada ocupará, en general, mayor área que la celda estándar CMOS, pero, dado el pequeño tamaño de la memoria necesaria en el controlador ejemplo, esta desventaja queda compensada con la simplicidad del funcionamiento de la primera, totalmente asíncrono, y la mayor flexibilidad y robustez que aporta a la hora de reconfigurar la organización de la memoria según el esquema antes descrito para los dos tipos de accesos de lectura, dado que es posible prescindir de los circuitos periféricos de memoria que han de acompañar a la celda estándar, y que son necesarios para su correcto funcionamiento.

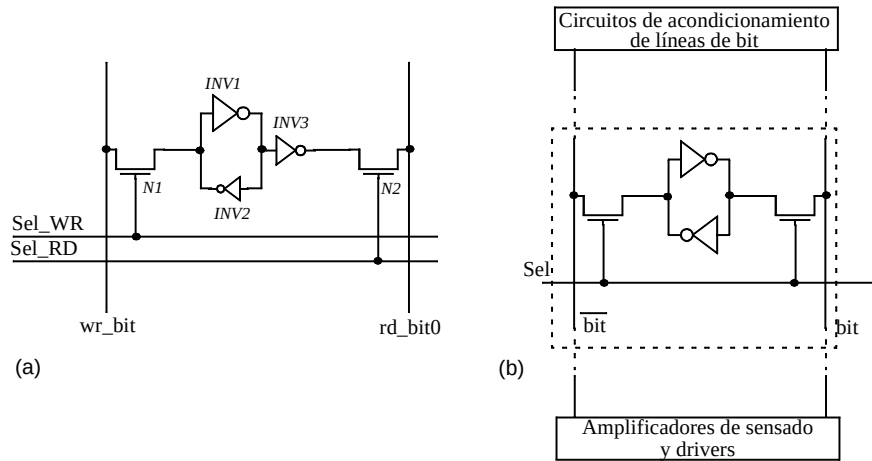


**Fig.5.26 Celdas estáticas de memoria: Celda de memoria de un solo terminal por línea de bit (a); celda estándar CMOS de seis transistores.**

La celda estándar CMOS está optimizada para ser empleada en el diseño de memorias de gran capacidad de almacenamiento, en las que aparece íntimamente ligada, tanto en su funcionamiento, como en sus prestaciones finales y diseño, a los llamados circuitos periféricos de memoria: *circuitos de acondicionamiento de línea de bit* y *amplificadores de sensado y drivers*, Fig.5.26(b). En general, estos circuitos son usados para acelerar el tiempo de acceso, y reducir el tamaño de los transistores que forman el *latch* de almacenamiento, además de hacer a la celda independiente de la tecnología y más segura frente al ruido u otros efectos que resultan difíciles de estimar antes de la fabricación. Para cumplir su objetivo todo el sistema debe ser diseñado de forma cuidadosa. Los *circuitos de acondicionamiento de línea de bit* son

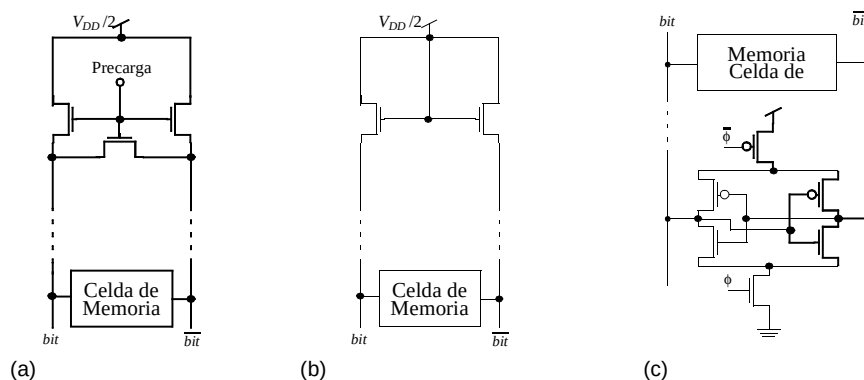


**Fig.5.27 Circuitos periféricos de memoria: Circuito de ecualización y precarga de línea dinámico (a); circuito de precarga estático (b); amplificador diferencial de sensado dinámico (c).**

empleados durante los accesos de lectura para precargar dichas líneas, habitualmente a un valor  $V_{DD}/2$ , de forma que, cuando la línea selección de fila es activada, el bit almacenado en la celda de memoria es sentido como una pequeña diferencia de potencial entre las dos líneas de bit. El *amplificador de sentido* se encarga de proporcionar a la salida la excursión lógica necesaria para interpretar el valor del bit almacenado. La pequeña diferencia de potencial necesaria a la entrada del amplificador de sentido para identificar el bit, permite diseñar una celda de memoria con transistores de menor tamaño y por tanto más pequeña. Téngase en cuenta que de estos circuitos periféricos sólo existe uno por columna en la memoria completa. Como circuitos de acondicionamiento de línea de bit suele emplearse dos tipos: *Circuitos de precarga y ecualización dinámicos*, que son controlados por una señal de forma que son activados únicamente en ciertas fase durante los accesos de lectura; o bien *circuitos de precarga estáticos*, que permanecen siempre activos. El primero de ellos exige una cuidadosa temporización de todas las señales de control para un buen funcionamiento. En el segundo caso, el diseño del sistema para los accesos de escritura ha de tener en cuenta el efecto de carga sobre la línea de bit de dichos circuitos, a la hora de forzar el valor adecuado que se desea escribir en el *latch* que forma la celda de memoria, misión ésta en la que intervienen los *drivers*. La Fig.5.27 muestra un ejemplo de los circuitos periféricos utilizados habitualmente. La presencia de estos elementos no resulta adecuada cuando se pretende realizar un bloque de memoria que funcione de forma asíncrona, y cuya organización tiene que ser reconfigurada según el origen y tipo de acceso, como es el caso del bloque constructivo de celdas de memoria propuesto en la Fig.5.24.

Por su parte, la celda de memoria de la Fig.5.26(a) está concebida para ser usada en *Register Files*, dada su flexibilidad para realizar muy diferentes configuraciones de sistemas multipuerto. En la Fig.5.28(a) se muestra a modo de ejemplo el caso de un sistema con una línea de escritura y dos de lectura.

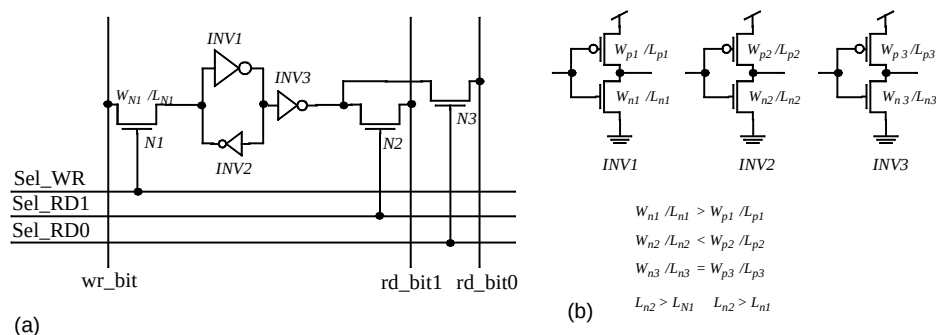


**Fig.5.28 Celda de memoria estática para Register Files con un puerto de escritura y dos de lectura (a); relación de aspectos entre los elementos de la celda (b) .**

En esta celda el par de inversores ( $INV1$  e  $INV2$ ) que forman el sistema de almacenamiento están desequilibrados de manera que el inversor  $INV2$  está diseñado para presentar un camino de realimentación débil entre la salida del inversor  $INV1$ , y su entrada. Este nudo está conectado a la línea de escritura a través del transistor de paso  $N1$ . Los transistores que forman el inversor  $INV2$  emplean un ancho de canal más pequeño, a la vez que una longitud de canal mayor, que dicho transistor de paso  $N1$ . Además el inversor  $INV1$  se dimensiona de forma que tiene su umbral de transición desplazado hacia el nivel de tierra; esto se consigue incrementando la relación de aspectos su transistor  $n$  respecto de su transistor  $p$ , esto es,  $W_{n1}/L_{n1} \geq W_{p1}/L_{p1}$ . Esta relación de aspectos, (ver Fig.5.28(b)) favorece el proceso de escritura del 1-lógico en la celda de memoria que es el más crítico cuando se emplean como transistores de paso elementos de canal  $n$  como es el caso del transistor  $N1$ . Por otra parte la salida del inversor  $INV1$  está conectada a otro inversor,  $INV3$ , que sirve de *buffer* para las diferentes líneas de lectura a las cuales se conecta a través de sus respectivos transistores de paso, ( $N2$  en la Fig.5.26(a) y  $N2$  y  $N3$  en la Fig.5.28(a)).

Estas características hacen que esta celda resulte ser muy robusta, y permita la flexibilidad requerida para la reorganización del bloque de *celdas de memoria* propuesto; dado que, sea cual sea la carga que aparezca en las líneas de salida del inversor  $INV3$ , el estado de la celda de memoria no corre el riesgo de ser alterado durante el proceso de lectura, como ocurre en las celdas convencionales si en su diseño no se pone especial cuidado. Además dicho inversor es capaz de forzar tanto en 1-lógico, como el 0-lógico en la línea de salida seleccionada, con lo que se evita la necesidad de los habituales dispositivos de *pull-up* que requieren consumo de potencia en dc, así como las señales de sincronismo y precarga. Por otra parte, su funcionamiento resulta ser completamente asíncrono.

Por todo esto, en el diseño del bloque constructivo básico a partir del cual se construye el *Bloque de Celdas de Memoria* se ha optado por emplear este tipo del celda. El precio a pagar, sin embargo, es la necesidad de un mayor número de transistores por celda. A pesar de todo, éste no es un precio elevado, dado el pequeño tamaño de la memoria necesaria en el controlador ejemplo y la importancia de aprovechar las ventajas de esta celda (empleo de una línea por puerto de acceso, su robustez ante las variaciones de carga en la línea de lectura, y su funcionamiento asíncrono).

El empleo de esta celda permite prescindir en el diseño del *Bloque de Celdas de Memoria* de los circuitos de acondicionamiento de línea de bit, ya que no son necesarios, al disponer cada celda de un inversor conectado a la línea de lectura. Por su parte, como amplificador de sentido se ha empleado un *latch* por cada línea de bit de lectura, en cada una de las dos interfases, externa e interna.

La Fig.5.29 muestra el esquema simplificado de una de las columnas de celdas de memoria junto con los elementos de reconfiguración y de sentido.

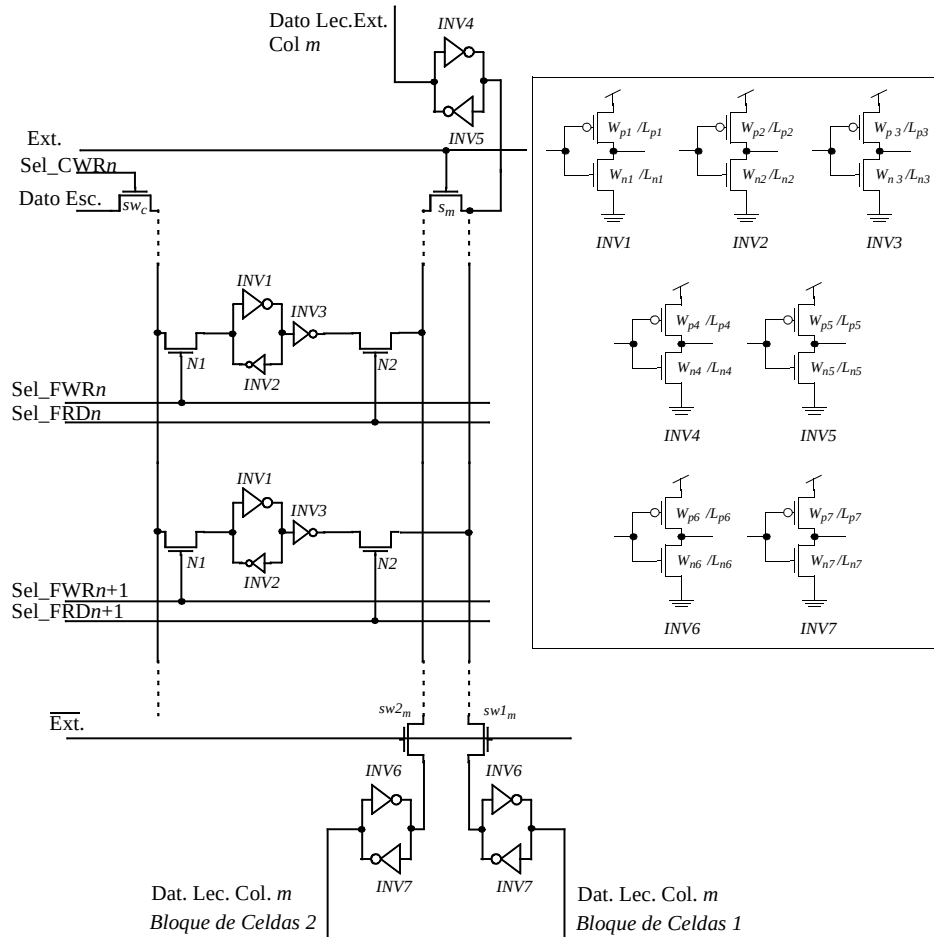


Fig.5.29 Dispositivos que intervienen en una columna del bloque de memoria

Finalmente, en la Tabla 5.3 se detallan la relación de aspectos de los transistores empleados el diseño del bloque de Celdas de Memoria, referidos al valor unitario ( $L=0.8\mu\text{m}$ ) según la denominación de la Fig.5.29.

**Tabla 5.3:** Relación aspectos de los transistores mostrados en esquema de la Fig.5.29

| INV1                                                                             | INV2                                                                            | INV3                                                                               | N1                                                                              | N2                                            | $s_m$                                         |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| $\frac{W_{p1}}{L_{p1}} = \frac{2}{1}$<br>$\frac{W_{n1}}{L_{n1}} = \frac{8}{1}$   | $\frac{W_{p2}}{L_{p2}} = \frac{2}{2}$<br>$\frac{W_{n2}}{L_{n2}} = \frac{2}{4}$  | $\frac{W_{p3}}{L_{p3}} = \frac{4}{1}$<br>$\frac{W_{n3}}{L_{n3}} = \frac{4}{1}$     | $\frac{W_{N1}}{L_{N1}} = \frac{4}{1}$                                           | $\frac{W_{N2}}{L_{N2}} = \frac{4}{1}$         | $\frac{W_{ns_m}}{L_{ns_m}} = \frac{5}{1}$     |
| INV4                                                                             | INV5                                                                            | INV6                                                                               | INV7                                                                            | $swc_n$                                       | $sw1_m = sw2_m$                               |
| $\frac{W_{p4}}{L_{p4}} = \frac{2}{2}$<br>$\frac{W_{n4}}{L_{n4}} = \frac{3}{1.5}$ | $\frac{W_{p5}}{L_{p5}} = \frac{5}{1}$<br>$\frac{W_{n5}}{L_{n5}} = \frac{10}{1}$ | $\frac{W_{p6}}{L_{p6}} = \frac{2.5}{2}$<br>$\frac{W_{n6}}{L_{n6}} = \frac{2.5}{3}$ | $\frac{W_{p7}}{L_{p7}} = \frac{5}{1}$<br>$\frac{W_{n7}}{L_{n7}} = \frac{10}{1}$ | $\frac{W_{nswc_n}}{L_{nswc_n}} = \frac{5}{1}$ | $\frac{W_{nsw1_m}}{L_{nsw1_m}} = \frac{5}{1}$ |

### 3.3.3. Selector de Filas

Según se muestra en la Fig.5.22, este bloque se encarga de decodificar las líneas de dirección (tanto externas, como internas), y activar las líneas de selección de fila adecuadas; todo ello dependiendo del origen y tipo de acceso.

La Fig.5.30 muestra un diagrama de bloques conceptual de este bloque, que sirve para explicar su funcionamiento, así como para definir el conjunto de señales de interfaz con el resto de bloques. Su diseño, que será descrito un poco después, se ha organizado de forma modular empleando un bloque básico que es repetido tantas veces como filas hay en el bloque de memoria.

El origen del acceso, externo o interno, lo indica la señal **Ext**, (**Ext** = 1 acceso externo, **Ext** = 0 acceso interno, según el esquema de la Fig.5.30) Esta señal es utilizada para habilitar o inhibir, según el caso, el decodificador binario de direcciones externas, o el decodificador Gray de direcciones internas.

Para los accesos externos (**Ext** = 1), como muestra la Fig.5.21(a), la memoria es vista como si estuviera organizada según una matriz de 8 filas y 8 columnas. Por ello para la decodificación de la fila seleccionada se han empleado los 3 bit menos significativos de los 6 bits de la dirección externa de acceso **AExt[5:0]**, según el esquema de la Fig.5.30, (los otros

tres bits se emplean en el control del multiplexor para la selección de la columna). Por cada dirección de acceso sólo se activa una línea de selección de fila, **Sel\_Fi**. Dada la celda de memoria utilizada, es necesario conocer además si el acceso es de escritura o lectura, lo que es indicado por la señal **Esc\_Lec**, a fin de activar la línea de selección de fila correspondiente, **SEL\_FWRi** o **SEL\_FRDi**. El diseño del bloque decodificador binario es directo ya que cada una de sus salidas corresponde con el minitérmino de la función de tres variables que indica su ordinal. Además cada uno de esos minitérminos está validado (AND lógica), por la entrada de habilitación **Ena**.

Por su parte, los accesos internos (**Ext** = 0), sólo pueden ser de lectura, por lo que en este caso las líneas de selección de fila para escritura, **SEL\_FWRi**, permanecerán siempre desactivadas. Como muestra la Fig.5.21(b), para este tipo de accesos la memoria es vista como dos bloques de 4 filas y 8 columnas, de forma que uno de ellos contiene las filas pares y otro las impares. Por cada dirección de acceso, **AInt[2:0]** en el diagrama de la Fig.5.30, el bloque decodificador Gray activa dos líneas de selección de fila **Seli\_Fi** y **Seli\_Fi+1**. La primera corresponde a la fila que indica la dirección de acceso, y la segunda corresponde a la dirección siguiente, por tanto, cada una de las filas seleccionadas pertenece a un bloque de celdas diferente. Como consecuencia, para accesos internos, siempre aparecen dos líneas de selección de fila consecutivas activas simultáneamente, **Sel\_FRDi** y **SEL\_FRDi+1**.

En el controlador ejemplo se tienen cuarenta y nueve intervalos  $C_{mn}$ , con  $m, n = 1, 2, \dots, 7$ , cada uno de los cuales está identificado por una palabra de seis bits que se ha simbolizado con anterioridad como  $AC_{mn} = A_{mb}A_{nb}$ , donde  $A_{mb}$ , y  $A_{nb}$  representan palabra  $m$ -ésima y  $n$ -ésima, respectivamente, del código Gray de tres bit generado a partir de

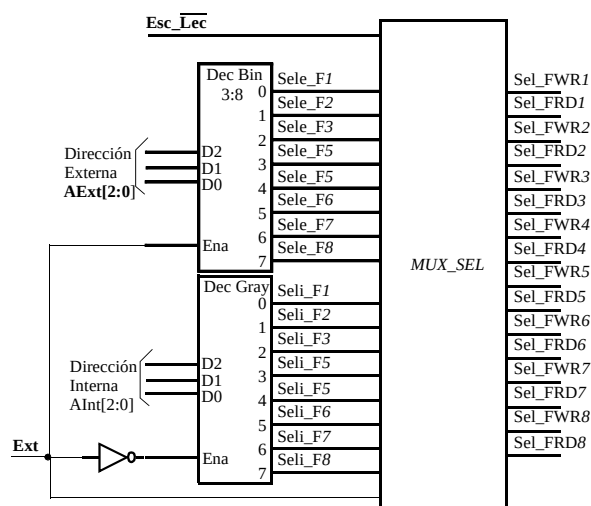


Fig.5.30 Diagrama de bloques conceptual del Selector de Filas.



Tabla de verdad

| Ena | D2 | D1 | D0 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |                                                   |
|-----|----|----|----|----|----|----|----|----|----|----|----|---------------------------------------------------|
| 0   | x  | x  | x  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | $S7 = \text{Ena} * \overline{D1} * \overline{D0}$ |
| 1   | 0  | 0  | 0  | x  | x  | x  | x  | x  | x  | x  | x  | $S6 = \text{Ena} * D2 * \overline{D1}$            |
| 1   | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | $S5 = \text{Ena} * D2 * D0$                       |
| 1   | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | $S4 = \text{Ena} * D2 * D1$                       |
| 1   | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | $S3 = \text{Ena} * D1 * \overline{D0}$            |
| 1   | 1  | 1  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | $S2 = \text{Ena} * \overline{D2} * D1$            |
| 1   | 1  | 1  | 1  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | $S1 = \text{Ena} * \overline{D2} * D0$            |
| 1   | 1  | 0  | 1  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | $S0 = \text{Ena} * \overline{D2} * \overline{D1}$ |
| 1   | 1  | 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |                                                   |

(a)

(b)

Fig.5.31 Tabla de verdad (a); y expresiones booleanas de las funciones de salida (b); del bloque de decodificación Gray.

la identificación del intervalo a que pertenecen cada una de las entradas. Si la distribución de datos en las celdas de memoria se ha realizado como se ha mostrado en el ejemplo de la Fig.5.20, se pueden utilizar las líneas que corresponden a los tres bits  $A_{nb}$  como entradas del bloque decodificación Gray que aparece en la Fig.5.30,  $A_n[2:0]$ , y generar la selección de filas adecuada. La Fig.5.31(a) muestra la Tabla de Verdad de dicho bloque combinacional, donde  $D[2:0]$  se hace corresponder con  $A_n[2:0]$ , esto es, los tres bit de la dirección interna. Como muestra dicha tabla la primera palabra del código no es utilizada. En ella se incluye además la entrada de habilitación del decodificador, denominada Ena en la Fig.5.30. Este bloque realiza una decodificación de palabras en código Gray, pero no se corresponde exactamente con el decodificador Gray de tres bits descrito en la Fig.5.6, ya que, como se ha mencionado, el bloque que aquí se emplea ha de ser capaz de activar dos de sus líneas de salida simultáneamente. En la Fig.5.31(b) se muestran las expresiones booleanas de cada una de las líneas de selección,  $S[7:0]$ , que se corresponden con las líneas **Seli\_F[8:1]** de la Fig.5.30.

Tabla de verdad

| Ext | Esc_Lec | Sel_FWRi | Sel_FRDi |
|-----|---------|----------|----------|
| 0   | 0       | 0        | Seli_Fi  |
| 0   | 1       | 0        | Seli_Fi  |
| 1   | 0       | 0        | Sele_Fi  |
| 1   | 1       | Sele_Fi  | 0        |

$$\text{Sel\_FWRi} = \text{Ext} * \text{Esc\_Lec} * \text{Sele\_Fi}$$

$$\text{Sel\_FRDi} = (\text{Ext} * \text{Esc\_Lec} * \text{Sele\_Fi}) + (\overline{\text{Ext}} * \text{Seli\_Fi})$$

Fig.5.32 Comportamiento del bloque MUX\_SEL: Tabla de verdad y expresiones booleanas.

Por lo que respecta al bloque denominado *MUX\_SEL* en la Fig.5.30, su funcionamiento puede resumirse como indica la tabla y las expresiones booleanas de la Fig.5.32. Para accesos internos, **Ext** = 0, todas las líneas de selección de fila para escritura, toman el valor lógico cero; mientras que las líneas de selección de fila para lectura toman el valor que corresponda a la salida del decodificador Gray. Para accesos externos, **Ext** = 1, si el acceso es de escritura, todas las líneas de selección de fila para lectura toman el valor lógico cero, mientras que las líneas de selección para escritura toman el valor correspondiente de la salida del decodificador binario. El caso contrario ocurre para accesos externos de lectura, donde todas las líneas de selección de fila para escritura toman el valor lógico cero, mientras que las líneas de selección para lectura toman el valor correspondiente de la salida del decodificador binario. Dado que, tanto las líneas **Sele\_Fi**, como las **Seli\_Fi** dependen solamente de las líneas de dirección correspondientes, cada una de las líneas **Sel\_FRDi** y **Sel\_FWRi** de la Fig.5.30 puede obtenerse a partir de bloques básicos idénticos, como el que se muestra en la Fig.5.33; por esta razón, el bloque selector de intervalo se ha diseñado empleando ocho de estos bloques básicos, uno para cada pareja de líneas de selección de fila.

La Fig.5.33(a) muestra el interfaz de ese bloque básico, mientras que la Fig.5.33(b) muestra su diagrama lógico. La Fig.5.34 muestra un detalle de su *layout*. Todas la puertas

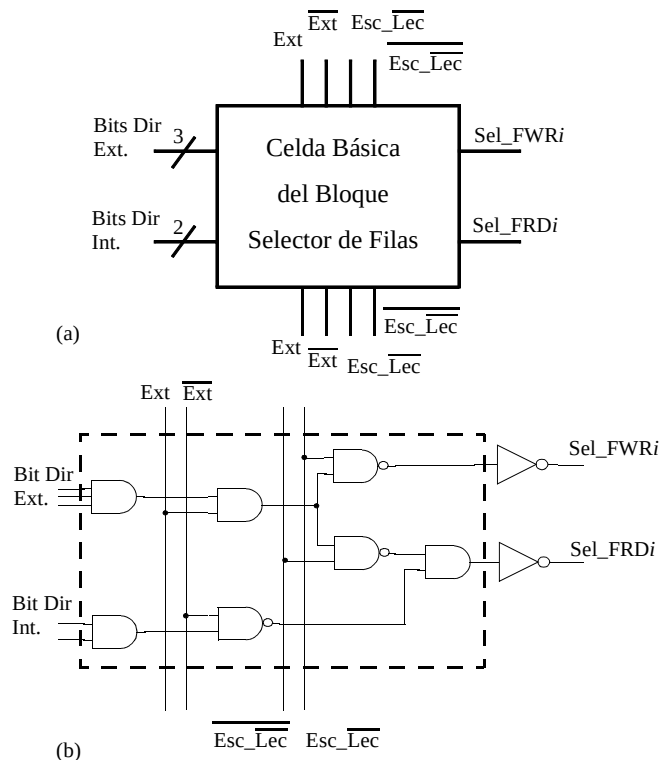


Fig.5.33 Celda básica del bloque selector de filas: Interfaz (a); diagrama lógico (b).

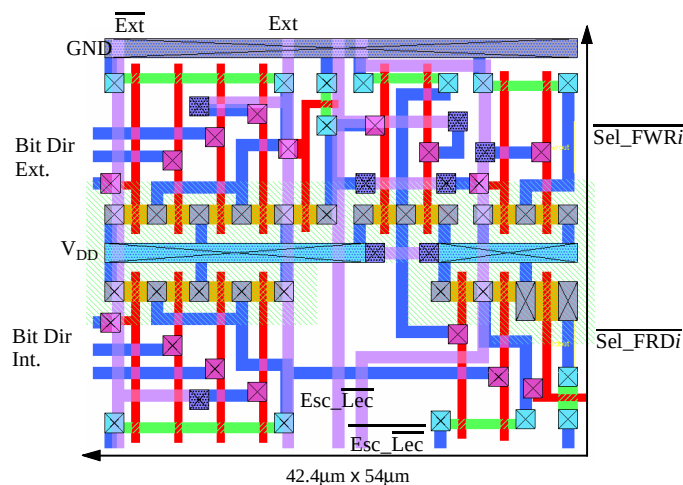


**Fig.5.34** Detalle del layout de la celda básica del bloque selector de filas

lógicas que aparecen en ese diagrama, han sido realizadas con lógica CMOS, y todas, salvo el inversor que genera la salida  $\overline{\text{Sel\_FRDi}}$ , emplean transistores **n** de relación de aspectos mínima y transistores **p** con anchura doble de la mínima. Los inversores que aparecen en la Fig.5.33(b), y que no aparecen en el *layout* de la Fig.5.34, han sido diseñados con una relación de aspectos mayor, de forma que actúen como *buffers*, para conducir las líneas de selección de fila.

### 3.3.4. Selector de Columnas para accesos externos

Como se muestra en la Fig.5.22, en la selección de columnas para accesos externos intervienen dos elementos: un bloque de *Control de Selección de Columnas* y un multiplexor digital. El primero proporciona dos conjuntos de señales de control: uno que permite seleccionar la columna adecuada del bloque de celdas de memoria para los accesos de escritura, mientras que el otro permite controlar el multiplexor digital, para palabras de cuatro bits, que permite seleccionar la palabra de salida para accesos de lectura externos, de entre las ocho columnas en las que, para este tipo de accesos, se reconfigura el bloque de *Celdas de Memoria*. La Fig.5.35 muestra un diagrama en el que aparecen estos elementos, y que sirve para explicar su funcionamiento, así como para definir el conjunto de señales de interfaz con el resto de bloques.

El bloque *Control de Selección de Columnas* utiliza como entrada los tres bits más significativos **AExt[5:3]** de la dirección externa de acceso **AExt[5:0]**, (los otros tres bits se emplean para la selección de la fila como se ha visto en el apartado anterior). Estas tres líneas son empleadas directamente como líneas de control del multiplexor **C[2:0]** en la Fig.5.35(b).

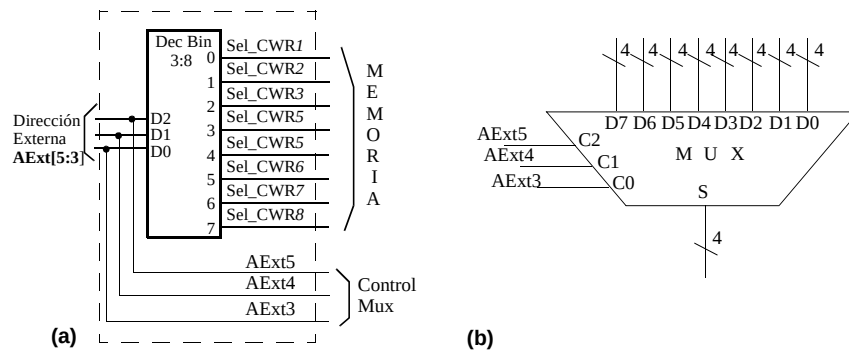


**Fig. 5.35 Elementos básicos para selección de columnas en el interfaz de accesos externos: Control de Selector de columnas (a); Símbolo del multiplexor 8x1 empleado como selector de columna de datos en lectura (b).**

Por su parte, las líneas de selección de columna para accesos de escritura **Sel\_CWR**, que constituyen su interfaz con el bloque de *Celdas de Memoria* se obtienen como ilustra la Fig. 5.35(a) a partir de la decodificación de las líneas de entrada por medio de un decodificador binario convencional.

En la Fig. 5.35(b) se muestra símbolo utilizado habitualmente para representar a un multiplexor digital, donde el ordinal de cada entrada se corresponde con el código binario natural de la palabra de control que la selecciona. Por su parte, la Fig. 5.36(a) muestra el esquema del circuito que realiza dicho multiplexor para palabras de un bit en el controlador ejemplo. Por simplicidad y compacidad este multiplexor ha sido diseñado empleando transistores NMOS, todos del mismo tamaño. El multiplexor completo se realiza empleando cuatro bloques como el de la Fig. 5.36(a) controlados en paralelo. En el controlador ejemplo, a

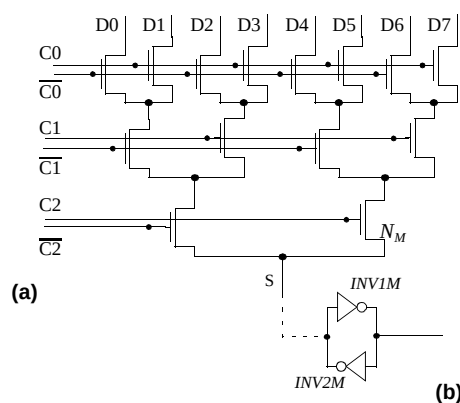


**Fig. 5.36 Multiplexor de columnas para accesos externos: esquema del multiplexor realizado con transistores NMOS (a). Latch empleado para la regeneración de los niveles lógicos (c).**

la salida de cada multiplexor se ha añadido un *latch* CMOS (INV1M e INV2M en la Fig.5.36(b)) que ha sido dimensionado para regenerar los niveles lógicos, especialmente el 1-lógico, que es el que resulta ser el más degradado en su transmisión a través de transistores NMOS.

La Fig.5.37 muestra un detalle de layout del multiplexor para un bit que sigue el esquema de la Fig.5.36(b), mientras la Tabla 5.4, recoge los aspectos de los transistores empleados en este diseño referidos al valor unitario ( $L=0.8\mu\text{m}$ ), y según se nombran en la Fig.5.36.

**Tabla 5.4:** Tamaño los transistores empleados en el diseño del controlador ejemplo

| INV1M                                      | INV2M                                   | $N_M$                                     |
|--------------------------------------------|-----------------------------------------|-------------------------------------------|
| $\frac{W_{p1M}}{L_{p1M}} = \frac{4}{1}$    | $\frac{W_{p2M}}{L_{p2M}} = \frac{2}{2}$ | $\frac{W_{nN_M}}{L_{nN_M}} = \frac{6}{1}$ |
| $\frac{W_{n1M}}{L_{n1M}} = \frac{12.5}{1}$ | $\frac{W_{n2M}}{L_{n2M}} = \frac{2}{3}$ |                                           |

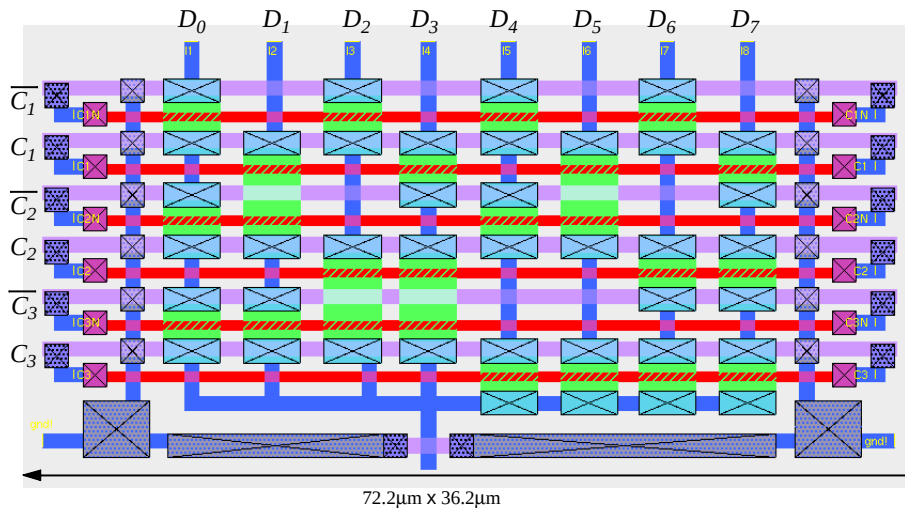


**Fig.5.37** Detalle de layout del multiplexor 8x1 realizado con transistores NMOS

## 3.3.5. Selector de Columnas para accesos internos

Como muestra la Fig.5.22, en la selección de columnas para accesos externos intervienen dos elementos: Un bloque de *Control de Selección de Columnas* y un conjunto de cuatro multiplexores digitales, para palabras de cuatro bits, que funcionan simultáneamente. Estos multiplexores permiten seleccionar cuatro palabras por cada acceso de lectura interno definido por la dirección  $AC_{mn} = A_{mb}A_{nb}$  generada por el bloque *Selector de Intervalo*, ( $A_{mb} = A_m[2:0]$  y  $A_{nb} = A_n[2:0]$ ). Esta selección se realiza de entre las 16 columnas, 8 por cada uno de los dos bloques de celdas de memoria, en las que, para este tipo de accesos, se reconfiguran los bloques de *Celdas de Memoria* del bloque *Programador de Consecuente de Reglas*, ver Fig.5.21(b). Recuérdese que 4 es el número de reglas activas, y por tanto 4 el número de valores *singleton* que este bloque ha de proporcionar al *Núcleo Analógico*.

La Fig.5.38 muestra un esquema conceptual de la organización de este bloque, que ilustra su funcionamiento, y sirve para definir el conjunto de señales de interfaz con el resto de bloques. En el interfaz con los bloques de *Celdas de Memoria* aparecen las 16 líneas de datos de columna. En el interfaz con el *Núcleo Analógico* aparecen los cuatro valores *singleton* de programación. Finalmente en el interfaz con el bloque *Selector de Intervalo* aparecen las líneas de dirección para accesos internos. El bloque de conexionado representa de modo genérico como se conectan las líneas de columna a los diferentes multiplexores.

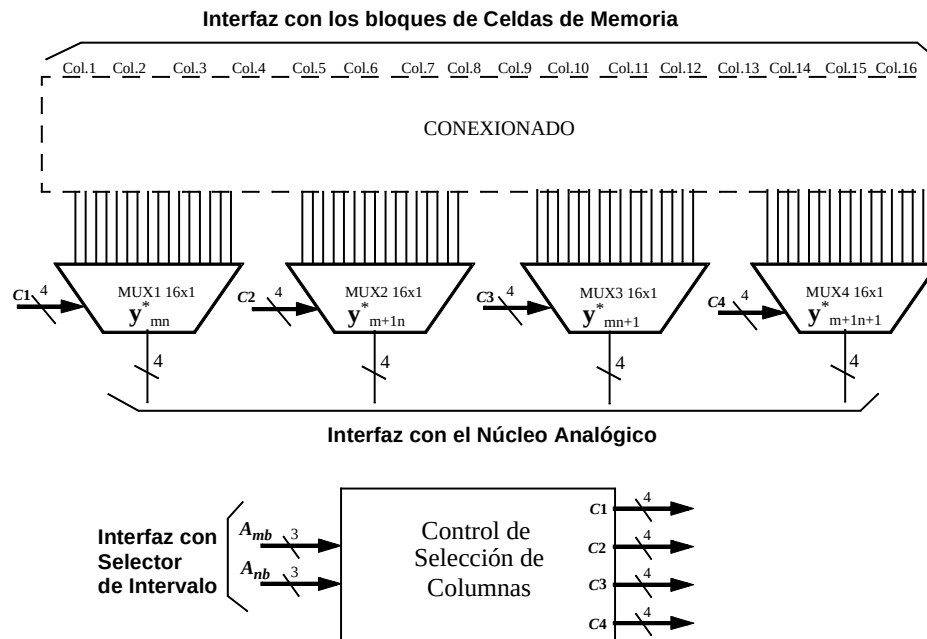


Fig.5.38 Diagrama de bloques conceptual del Selector de Columnas para accesos internos.

En la Fig.5.39(a) se muestra símbolo utilizado habitualmente para representar a un multiplexor digital 16x1, como los empleados en este diseño, donde el ordinal de cada entrada se corresponde con el código binario natural de la palabra de control que la selecciona. Por su parte, en la Fig.5.39(b) se muestra el esquema del circuito que realiza dicho multiplexor para palabras de un bit. Al igual que el multiplexor descrito en el apartado anterior, este multiplexor también ha sido diseñado empleando únicamente transistores NMOS, todos del mismo tamaño. Un multiplexor completo, para palabras de cuatro bits se realiza empleando cuatro de bloques como el de la Fig.5.39(b) controlados en paralelo. A su vez, los cuatro multiplexores de palabras de cuatro bit que forman el bloque completo son idénticos a éste. De la misma manera, y por la razón expuesta en el apartado anterior, a la salida de cada multiplexor para palabras de un bit se ha añadido un *latch* CMOS (INV1M e INV2M en la Fig.5.39(c)) que ha sido dimensionado para regenerar los niveles lógicos.

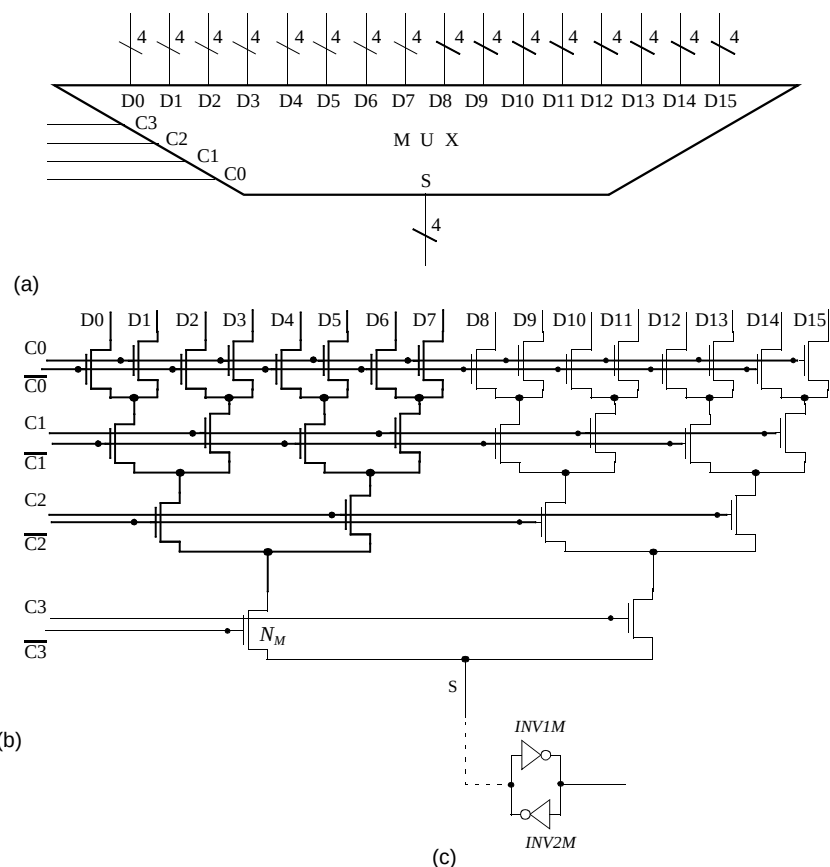


Fig.5.39 Multiplexor de columnas para accesos internos: Símbolo de un multiplexor 16x1 (a); esquema del multiplexor realizado con transistores NMOS (b). *latch* empleado en la regeneración de los niveles lógicos (c).

La Fig.5.40 muestra un detalle de *layout* del multiplexor para un bit que sigue el esquema de la Fig.5.39(b). La relación de aspectos de los transistores empleados en este diseño se muestran en la Tabla 5.6, referidos al valor unitario ( $L=0.8\mu\text{m}$ ) según la denominación de la Fig.5.39(b).

**Tabla 5.5:** Tamaño de los transistores empleados en el diseño del Selector de Columnas en el controlador ejemplo

| $INV1M$                                    | $INV2M$                                 | $N_M$                                     |
|--------------------------------------------|-----------------------------------------|-------------------------------------------|
| $\frac{W_{p1M}}{L_{p1M}} = \frac{4}{1}$    | $\frac{W_{p2M}}{L_{p2M}} = \frac{2}{2}$ | $\frac{W_{nN_M}}{L_{nN_M}} = \frac{6}{1}$ |
| $\frac{W_{n1M}}{L_{n1M}} = \frac{12.5}{1}$ | $\frac{W_{n2M}}{L_{n2M}} = \frac{2}{3}$ |                                           |

Por lo que respecta al diseño del bloque de *Control de Selección de Columnas* que aparece en la Fig.5.38, con un conexionado adecuado entre las líneas de datos de columna y las entradas de dato de los multiplexores puede conseguirse por una parte, unificar los cuatro buses de control,  $C_i$  con  $i = 1, \dots, 4$ , de forma que los cuatro multiplexores sean controlados por las mismas señales de control, y por otra que las funciones booleanas que generan dichas señales de control a partir de las líneas de dirección interna sean lo más sencillas posibles. En la Fig.5.41 se ilustra el conexionado realizado en el controlador ejemplo. Con él se consigue que las tres señales de control de menor peso,  $C[2:0]$  se correspondan directamente con las tres líneas de dirección interna  $A_{mb} = A_m[2:0]$  de la palabra  $AC_{mn} = A_{mb}A_{nb}$  generada por el bloque *Selector de Intervalo*; mientras que la línea de control de más peso,  $C3$  se obtiene como resultado de evaluar la paridad de las otras tres líneas  $A_{nb} = A_n[2:0]$ . Por tanto las funciones que generan las líneas  $C[2:0]$  son las más simples posibles, mientras que  $C3$  se genera a partir de una puerta XOR de tres entradas.

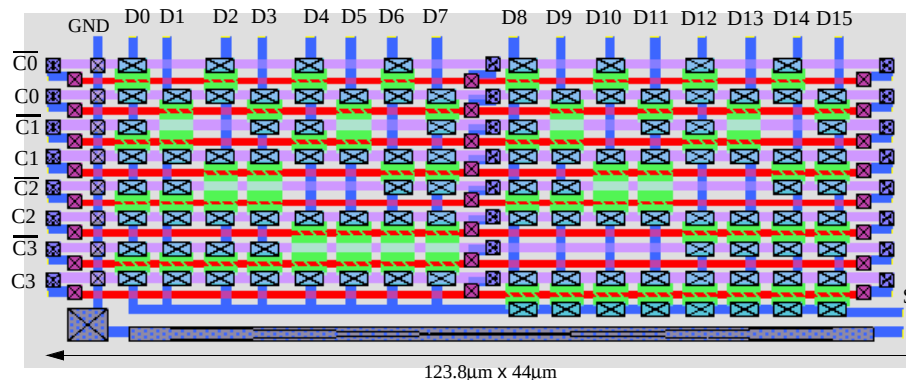


**Fig.5.40** Detalle de *layout* del multiplexor 16x1 realizado con transistores NMOS



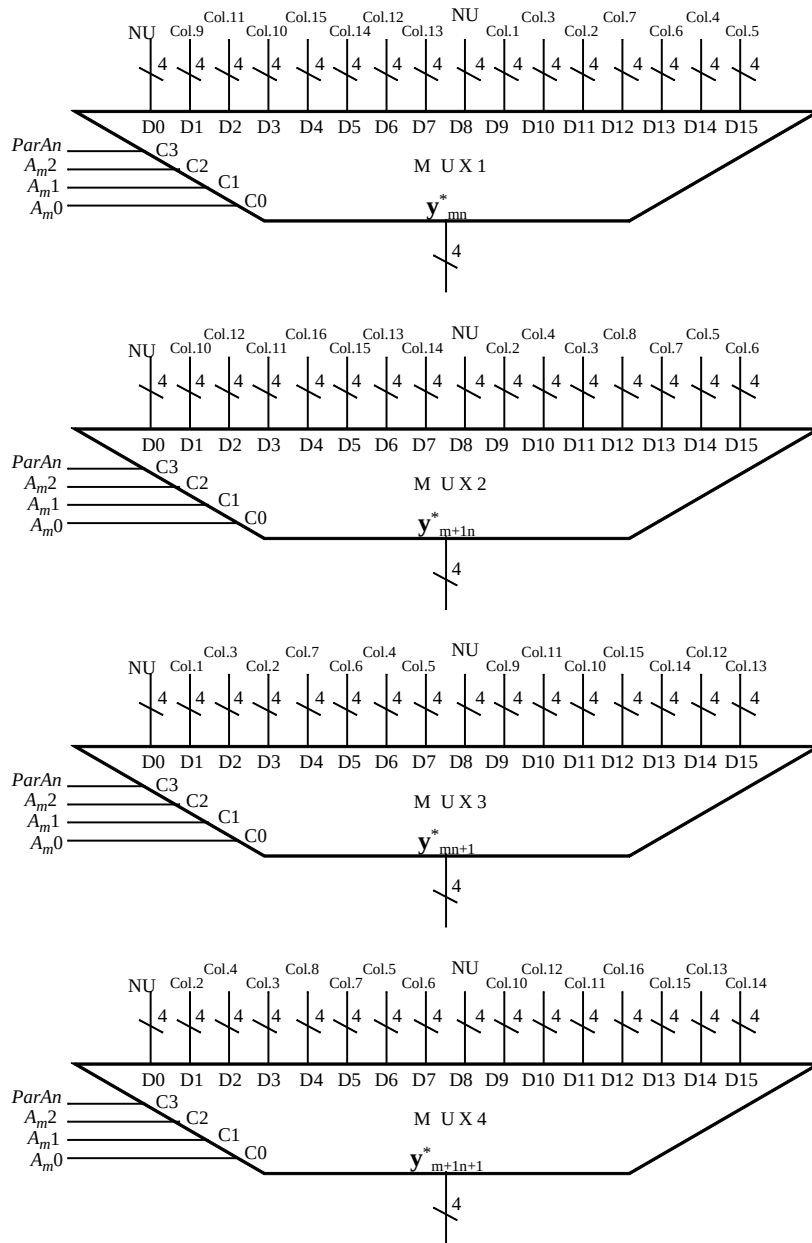


Fig.5.41 Esquema del conexionado entre las líneas de dato de columna y cada uno de los cuatro multiplexores.

La Tabla de Conexionado que muestra la Fig.5.42 resume el esquema de selección de columnas que da lugar al conexionado de la Fig.5.41. En esa tabla la entrada  $ParA_n$  toma valor lógico cero cuando  $A_{nb}$  posee un número par de unos, mientras que toma el valor lógico 1 cuando  $A_{nb}$  posee un número impar de unos.  $A_m[2:0]=C[2:0]$  corresponde con las líneas de control de los multiplexores. Como muestra la tabla, de las ocho direcciones que representa  $A_m[2:0]$ , la combinación  $A_{mb} = 000$  no se usa. Por último, las cifras 1,2,3,4 que aparecen rellenando la tabla representan a cada uno de los cuatro multiplexores.

Para ilustrar el significado de la Tabla de Conexionado de la Fig.5.42(b) consideremos los siguientes ejemplos:

- Supongamos que la dirección de acceso es  $A_{2b}A_{1b} = 011\ 001$ . Dado que  $A_{1b} = 001$  posee un número impar de unos se tiene que  $ParA_n = 1$ . Por su parte  $A_{2b} = 011$  indica la entrada de dato que es seleccionada en todos los multiplexores, en este caso  $D3$ . La información que proporciona la tabla debe interpretarse como que la columna 2 ha de ser conectada a la entrada  $D3$  del multiplexor 1, que la columna 3 ha de ser conectada a la entrada  $D3$  del multiplexor 2, que la columna 10 ha de ser conectada a la entrada  $D3$  del multiplexor 3 y que la columna 11 ha de ser conectada a la entrada  $D3$  del multiplexor 4. Dado que la dirección  $A_{nb} = 001$  selecciona las filas 1 y filas 2 del bloque de celdas de memoria, los valores *singleton* que aparecen a la salida son  $y_{21}^*$  y  $y_{31}^*$  y  $y_{22}^*$  y  $y_{32}^*$ , en los multiplexores 1,2 3 4 respectivamente.
- Supongamos que la dirección de acceso es  $A_{3b}A_{2b} = 010\ 011$ . Dado que  $A_{2b} = 011$  posee un número par de unos se tiene que  $ParA_n = 0$ . Por su parte  $A_{3b} = 010$  indica la entrada de dato que es seleccionada en todos los multiplexores, en este caso  $D2$ . La información que proporciona la tabla debe interpretarse como que la columna 11 ha de ser conectada a la entrada  $D2$  del multiplexor 1, que la columna 12 ha de ser conectada a la entrada  $D2$  del multiplexor 2, que la columna 3 ha de ser conectada a la entrada  $D2$  del multiplexor 3 y que la columna 4 ha de ser conectada a la entrada  $D2$  del multiplexor 4. Dado que la dirección  $A_{nb} = 011$  selecciona las filas 2 y filas 3 del bloque de celdas de memoria los valores *singleton* que aparecen a la salida son  $y_{32}^*$  y  $y_{42}^*$  y  $y_{33}^*$  y  $y_{43}^*$ , en los multiplexores 1,2 3 4 respectivamente.

Para completar la descripción del diseño del bloque *Control de Selección de Columnas* de la Fig.5.38, la Fig.5.43 muestra el diagrama lógico, Fig.5.43(a), y un circuito para tecnologías CMOS, Fig.5.43(b), que realizar la puerta XOR de tres entradas que se emplea para generar la señal de control  $C3 = ParA_n$  de los multiplexores en función de los bit de dirección interna  $A_n[2:0]$ . En el diseño se ha incluido también la generación de la señal complementaria  $\overline{C3}$ . La Fig.5.44 muestra un detalle del *layout* de este circuito. Como puede observarse todos los transistores  $P_M$  son idénticos y todos los transistores  $N_M$  también. Los inversores de salida han sido dimensionados para conducir adecuadamente las líneas de control de todos los

## Distribución de Datos para accesos internos

|        | Bloque de Celdas de Memoria 1 |            |            |            |            |            |            |            | Bloque de Celdas de Memoria 2 |            |            |            |            |            |            |            |
|--------|-------------------------------|------------|------------|------------|------------|------------|------------|------------|-------------------------------|------------|------------|------------|------------|------------|------------|------------|
|        | Col.1                         | Col.2      | Col.3      | Col.4      | Col.5      | Col.6      | Col.7      | Col.8      | Col.9                         | Col.10     | Col.11     | Col.12     | Col.13     | Col.14     | Col.15     | Col.16     |
| Fila 1 | $y_{11}^*$                    | $y_{21}^*$ | $y_{31}^*$ | $y_{41}^*$ | $y_{51}^*$ | $y_{61}^*$ | $y_{71}^*$ | $y_{81}^*$ | $y_{12}^*$                    | $y_{22}^*$ | $y_{32}^*$ | $y_{42}^*$ | $y_{52}^*$ | $y_{62}^*$ | $y_{72}^*$ | $y_{82}^*$ |
| Fila 2 |                               |            |            |            |            |            |            |            |                               |            |            |            |            |            |            |            |
| Fila 3 | $y_{13}^*$                    | $y_{23}^*$ | $y_{33}^*$ | $y_{43}^*$ | $y_{53}^*$ | $y_{63}^*$ | $y_{73}^*$ | $y_{83}^*$ | $y_{14}^*$                    | $y_{24}^*$ | $y_{34}^*$ | $y_{44}^*$ | $y_{54}^*$ | $y_{64}^*$ | $y_{74}^*$ | $y_{84}^*$ |
| Fila 4 |                               |            |            |            |            |            |            |            |                               |            |            |            |            |            |            |            |
| Fila 5 | $y_{15}^*$                    | $y_{25}^*$ | $y_{35}^*$ | $y_{45}^*$ | $y_{55}^*$ | $y_{65}^*$ | $y_{75}^*$ | $y_{85}^*$ | $y_{16}^*$                    | $y_{26}^*$ | $y_{36}^*$ | $y_{46}^*$ | $y_{56}^*$ | $y_{66}^*$ | $y_{76}^*$ | $y_{86}^*$ |
| Fila 6 |                               |            |            |            |            |            |            |            |                               |            |            |            |            |            |            |            |
| Fila 7 | $y_{17}^*$                    | $y_{27}^*$ | $y_{37}^*$ | $y_{47}^*$ | $y_{57}^*$ | $y_{67}^*$ | $y_{77}^*$ | $y_{87}^*$ | $y_{18}^*$                    | $y_{28}^*$ | $y_{38}^*$ | $y_{48}^*$ | $y_{58}^*$ | $y_{68}^*$ | $y_{78}^*$ | $y_{88}^*$ |
| Fila 8 |                               |            |            |            |            |            |            |            |                               |            |            |            |            |            |            |            |

(a)

Tabla de Conexionado

| Par | $A_n$ | $A_{m2}$ | $A_{m1}$ | $A_{m0}$ | Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 | Col.9 | Col.10 | Col.11 | Col.12 | Col.13 | Col.14 | Col.15 | Col.16 |
|-----|-------|----------|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|
| 0   | 0     | 0        | 0        | 0        | -     | -     | -     | -     | -     | -     | -     | -     | -     | -      | -      | -      | -      | -      | -      | -      |
|     | 0     | 0        | 1        |          | 3     | 4     | -     | -     | -     | -     | -     | -     | 1     | 2      | -      | -      | -      | -      | -      | -      |
|     | 0     | 1        | 1        |          | -     | 3     | 4     | -     | -     | -     | -     | -     | 1     | 2      | -      | -      | -      | -      | -      | -      |
|     | 0     | 1        | 0        |          | -     | -     | 3     | 4     | -     | -     | -     | -     | -     | 1      | 2      | -      | -      | -      | -      | -      |
|     | 0     | 1        | 0        |          | -     | -     | -     | 3     | 4     | -     | -     | -     | -     | -      | 1      | 2      | -      | -      | -      | -      |
|     | 1     | 1        | 0        |          | -     | -     | -     | -     | 3     | 4     | -     | -     | -     | -      | -      | 1      | 2      | -      | -      | -      |
|     | 1     | 1        | 1        |          | -     | -     | -     | -     | -     | 3     | 4     | -     | -     | -      | -      | -      | 1      | 2      | -      | -      |
|     | 1     | 0        | 1        |          | -     | -     | -     | -     | -     | -     | 3     | 4     | -     | -      | -      | -      | -      | 1      | 2      | -      |
| 1   | 0     | 0        | 0        |          | -     | -     | -     | -     | -     | -     | -     | -     | -     | -      | -      | -      | -      | -      | -      | -      |
|     | 0     | 0        | 1        |          | 1     | 2     | -     | -     | -     | -     | -     | -     | 3     | 4      | -      | -      | -      | -      | -      | -      |
|     | 0     | 1        | 1        |          | -     | 1     | 2     | -     | -     | -     | -     | -     | 3     | 4      | -      | -      | -      | -      | -      | -      |
|     | 0     | 1        | 0        |          | -     | -     | 1     | 2     | -     | -     | -     | -     | -     | 3      | 4      | -      | -      | -      | -      | -      |
|     | 1     | 1        | 0        |          | -     | -     | -     | 1     | 2     | -     | -     | -     | -     | -      | 3      | 4      | -      | -      | -      | -      |
|     | 1     | 1        | 1        |          | -     | -     | -     | -     | 1     | 2     | -     | -     | -     | -      | -      | 3      | 4      | -      | -      | -      |
|     | 1     | 0        | 1        |          | -     | -     | -     | -     | -     | 1     | 2     | -     | -     | -      | -      | -      | 3      | 4      | -      | -      |
|     | 1     | 0        | 0        |          | -     | -     | -     | -     | -     | -     | 1     | 2     | -     | -      | -      | -      | -      | 3      | 4      | -      |

(b)

Fig.5.42 Distribución de datos en memoria (a); Tabla de Conexionado entre columnas del bloque de celdas de memoria y entrada de dato de cada uno de los cuatro multiplexores (b).

multiplexores. La Tabla 5.6 resume la relación de aspectos de los transistores empleados en este diseño, referidos al valor unitario ( $L=0.8\mu\text{m}$ ) según la denominación de la Fig.5.43(b).

Tabla 5.6: Tamaño de los transistores empleados en el diseño del controlador ejemplo

| INV1                                   | INV2                                   | $N_M$                                     | $P_M$                                     |
|----------------------------------------|----------------------------------------|-------------------------------------------|-------------------------------------------|
| $\frac{W_{p1}}{L_{p1}} = \frac{10}{1}$ | $\frac{W_{p2}}{L_{p2}} = \frac{10}{1}$ | $\frac{W_{nN_M}}{L_{nN_M}} = \frac{1}{1}$ | $\frac{W_{nN_M}}{L_{nN_M}} = \frac{2}{1}$ |
| $\frac{W_{n1}}{L_{n1}} = \frac{5}{1}$  | $\frac{W_{n2}}{L_{n2}} = \frac{5}{1}$  |                                           |                                           |

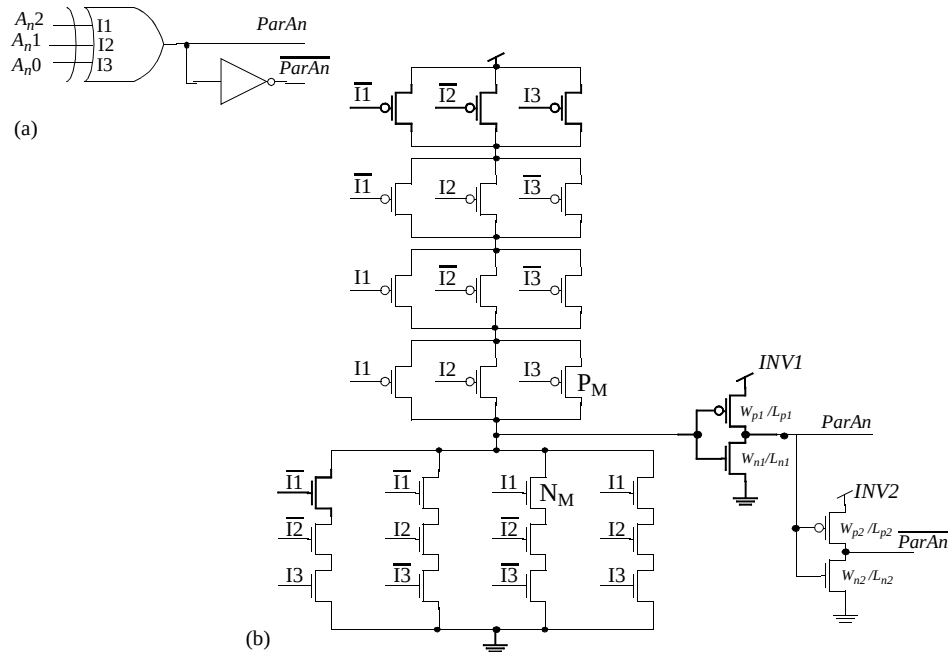


Fig.5.43 Puerta lógica XOR de tres entradas: Símbolo (a); circuito para su realización con tecnologías CMOS. (b)

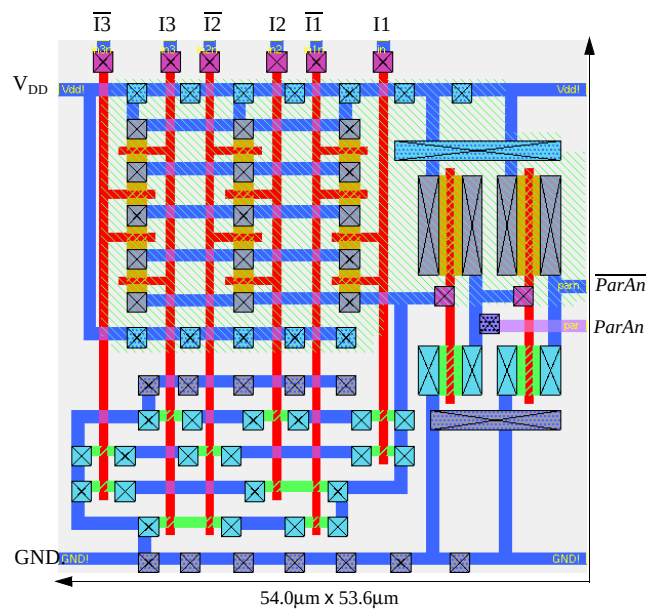


Fig.5.44 Detalle de layout del circuito XOR realizado con lógica CMOS

## 3.3.6. Consideraciones Globales de Diseño

El diseño del *Bloque de Programador de Consecuente de Reglas* ha sido verificado mediante la simulación con HSPICE del fichero **netlist** extraído del fichero de *layout*. A continuación se muestra, a modo de ilustración del correcto funcionamiento de este bloque, el resultado de alguna de las simulaciones realizadas.

En el ejemplo de simulación que aquí se presenta se incluyen todos los elementos que aparecen en el *layout* de la Fig.5.23. esto es, los bloques: *Celdas de Memoria*, *Selector de Filas*, *Selector de Columnas* para accesos externos y *Selector de Columnas* para accesos internos, según aparecen en la Fig.5.22. Sin pérdida de generalidad, para el propósito de esta simulación, el bloque de *Celdas de Memoria* que aquí se ha considerado sólo incluye las celdas correspondientes a uno de los cuatro bits que forman el valor *singleton*. En esta prueba, inicialmente se escriben, desde el interfaz externo las posiciones de memoria correspondientes a diez valores *singleton* que constituyen los consecuentes de otras tantas reglas y que pertenecen a tres intervalos de interpolación. A continuación se realizan tres accesos desde el interfaz interno, con las direcciones correspondientes a cada uno de estos tres intervalos, para leer los valores *singleton* correspondientes y en el orden establecido. Finalmente se realizan accesos desde el interfaz externo para leer los datos que se han escrito previamente y verificar la integridad de la memoria. La Fig.5.45 ilustra, sobre el esquema de la distribución de datos

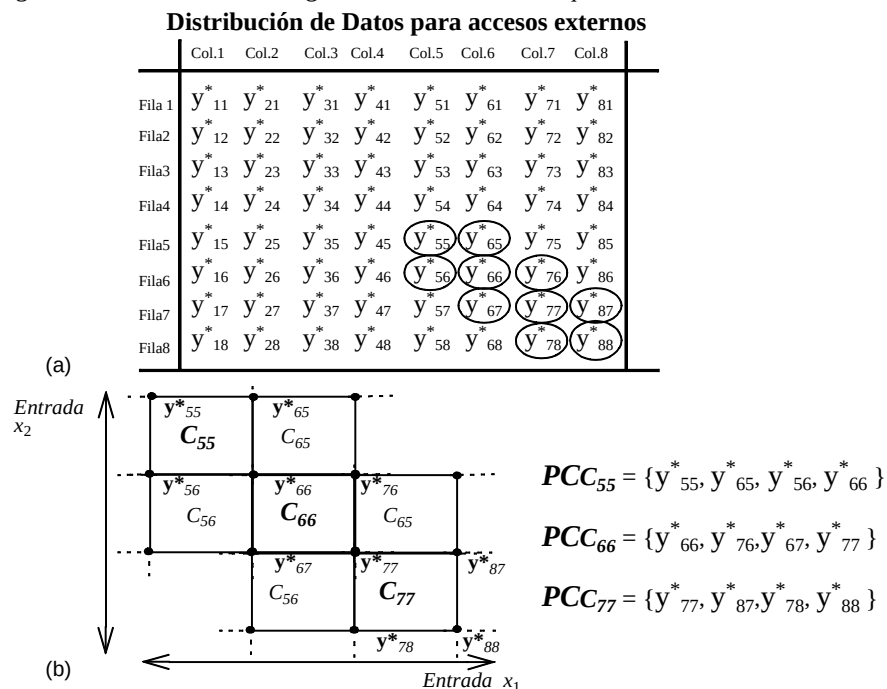


**Fig.5.45 Ilustración de las reglas e intervalos puestos en juego en el ejemplo de simulación que ilustra el comportamiento del bloque Programador de Consecuente de Reglas:**  
Localización de valores singleton en memoria (a); Intervalos de interpolación (b).

de la Fig.5.18, los valores *singleton* de los consecuentes de reglas puestos en juego y los intervalos de interpolación involucrados en este ejemplo. Las figuras Fig.5.46, Fig.5.47 y Fig.5.48 muestran la evolución de las principales señales de control y datos para la secuencia de operación antes descrita. Las diferentes señales puestas en funcionamiento en cada uno de los accesos, así como sus valores, tal y como evolucionan según las gráficas de esas figuras se resumen en la Tabla 5.7.

**Tabla 5.7:** Resumen de líneas y datos que aparecen en las gráficas de las figuras.

| Grupos de Señales                                                                          | Valores                                                                                                                                           |
|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Secuencia de dirección externa (AExt[5:0] (Hexa))                                          | 44H -> 45H -> 54H -> 55H -> 56H -><br>-> 65H -> 66H -> 67H -> 76H -> 77H                                                                          |
| Secuencia de datos escritura (DIN)                                                         | 1 -> 0 -> 1 -> 0 -> 0 -><br>-> 1 -> 1 -> 1 -> 0 -> 0                                                                                              |
| Secuencia de líneas de Selección de Columna para Escritura activas Sel_CWRn                | Sel_CWR5 -> Sel_CWR5 -> Sel_CWR6 -> Sel_CWR6 -> Sel_CWR6 -><br>-> Sel_CWR7 -> Sel_CWR7 -> Sel_CWR7 -> Sel_CWR8 -> Sel_CWR8                        |
| Secuencia de líneas de Selección de Fila para Escritura activas Sel_FWRn                   | Sel_FWR5 -> Sel_FWR5 -> Sel_FWR6 -> Sel_FWR6 -> Sel_FWR6 -><br>-> Sel_FWR7 -> Sel_FWR7 -> Sel_FWR7 -> Sel_FWR8 -> Sel_FWR8                        |
| Secuencia de líneas de Selección de Fila para Lectura externa activas Sel_FRDn             | Sel_FRD5 -> Sel_FRD5 -> Sel_FRD6 -> Sel_FRD6 -> Sel_FRD6 -><br>-> Sel_FRD7 -> Sel_FRD7 -> Sel_FRD7 -> Sel_FRD8 -> Sel_FRD8                        |
| Secuencia de líneas de control de multiplexor para lectura externa (Cl[2:0] (Hexa))        | 4H -> 4H -> 4H -> 5H -> 5H -><br>-> 6H -> 6H -> 6H -> 7H -> 7H                                                                                    |
| Secuencia de datos lectura (DOUT)                                                          | 1 -> 0 -> 1 -> 0 -> 0 -><br>-> 1 -> 1 -> 1 -> 0 -> 0                                                                                              |
| Serie de Valores <i>singleton</i> empleados                                                | $y_{55}^* = 1, y_{56}^* = 0, y_{65}^* = 1, y_{66}^* = 0, y_{67}^* = 0,$<br>$y_{76}^* = 1, y_{77}^* = 1, y_{78}^* = 1, y_{87}^* = 0, y_{88}^* = 0$ |
| Intervalos de interpolación empleados                                                      | $C_{55}, C_{66}, C_{77}$                                                                                                                          |
| Secuencia de dirección interna correspondientes a los intervalos (AInt[5:0](Hexa))         | 77H -><br>-> 55H -><br>-> 44H                                                                                                                     |
| Secuencia de parejas de líneas de Selección de Fila para Lectura internas activas Sel_FRDn | (Sel_FRD5, Sel_FRD6) -><br>-> (Sel_FRD6, Sel_FRD7) -><br>-> (Sel_FRD7, Sel_FRD8)                                                                  |

**Tabla 5.7:** Resumen de líneas y datos que aparecen en las gráficas de las figuras.

| Grupos de Señales                                                                                                | Valores                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Secuencia de líneas de control de multiplexor para lectura externa (C[3:0] (Hexa))                               | FH -><br>-> 5H -><br>-> CH                                                                                                                                                             |
| Secuencia de cuádruplas de valores <i>singleton</i> ordenadas según el intervalo de acceso considerado (BD[1:4]) | $(y_{55}^* = 1, y_{65}^* = 1, y_{56}^* = 0, y_{66}^* = 0)$<br>$(y_{66}^* = 0, y_{76}^* = 1, y_{67}^* = 0, y_{77}^* = 1)$<br>$(y_{77}^* = 1, y_{87}^* = 0, y_{78}^* = 1, y_{88}^* = 0)$ |

Finalmente, hay que señalar que la temporización impuesta en la simulación que aquí se muestra aprovecha al límite el tiempo de respuesta de este bloque para efectuar todas las acciones planeadas en el menor tiempo posible; así se pone de manifiesto su velocidad de operación y a la vez se consigue reducir, tanto el tiempo máquina de simulación, como el tamaño de los ficheros HSPICE de datos a manejar.

En las Fig.5.46(a) y Fig.5.46(b) se muestran las señales del interfaz para accesos externos, Fig.5.22, líneas de datos de entrada (DIN) y salida (DOUT), ciclo de escritura (ESC), ciclo de lectura (LEC), acceso externo (EXT), y líneas de dirección externa AExt[5:0]. Sobre estas gráficas puede observarse la secuencia completa de la simulación: acceso de escritura externo inicial, acceso interno intercalado, (según puede seguirse en la Fig.5.48) y finalmente acceso externo de lectura. Los valores de las secuencias de diferentes señales que intervienen se resumen en la Tabla 5.7.

La Fig.5.46(a) muestra cómo la secuencia DOUT obtenida durante los ciclos de lectura coincide con la secuencia DIN que había sido almacenada en memoria durante los ciclos de escritura, lo que pone de manifiesto el correcto funcionamiento durante el ciclo de escritura, y la integridad de la memoria, tras un acceso de lectura interno. El resto de gráficas de la Fig.5.46 muestran la evolución, descrita en la Tabla 5.7, de las principales señales de control puestas en juego durante los accesos de escritura externa. La Fig.5.46(c) muestra las líneas de selección de columna Sel\_CWRn, mientras que la Fig.5.46(d) muestra las líneas de selección de fila Sel\_FWRn, que son activadas para cada dirección de acceso. Véase la distribución de valores *singleton* en memoria que muestra la Fig.5.45. Aunque, como aparece en la Fig.5.46(c), las líneas de selección de columna se activan también durante los ciclos de lectura, el contenido de las celdas de memoria no es alterado puesto que durante estos ciclos no se activan las líneas de selección de fila, como queda manifiesto en la Fig.5.46(d).

En la Fig.5.47 se muestra la evolución de las principales señales puestas en juego durante los accesos de lectura externa. Las figuras Fig.5.47(a) y Fig.5.47(b) coinciden con las figuras Fig.5.46(a) y Fig.5.46(b), y son repetidas aquí para que sirvan de referencia. En la Fig.5.47(c) se muestran las líneas de selección de fila para accesos de lectura, Sel\_FRDn. La activación de

estas líneas que aparece en la zona central de la gráfica corresponde a los accesos de lectura internos, que serán comentados más adelante, como muestra que la señal EXT, Fig.5.47(a), está desactivada durante esos instantes. La activación de estas líneas como consecuencia del ciclo de lectura externa aparece en la derecha de esa gráfica, delimitada por la señal LEC de la Fig.5.47(a). Por su parte la Fig.5.47(d) muestra las líneas de control CI[2:0] del multiplexor de columnas para accesos externos, Fig.5.22. Nótese la coincidencia en aspecto de ambas secuencias con las correspondientes a las líneas de selección de fila y columnas durante el ciclo de escritura, dado que la secuencia de direcciones externas es la misma en ambos casos.

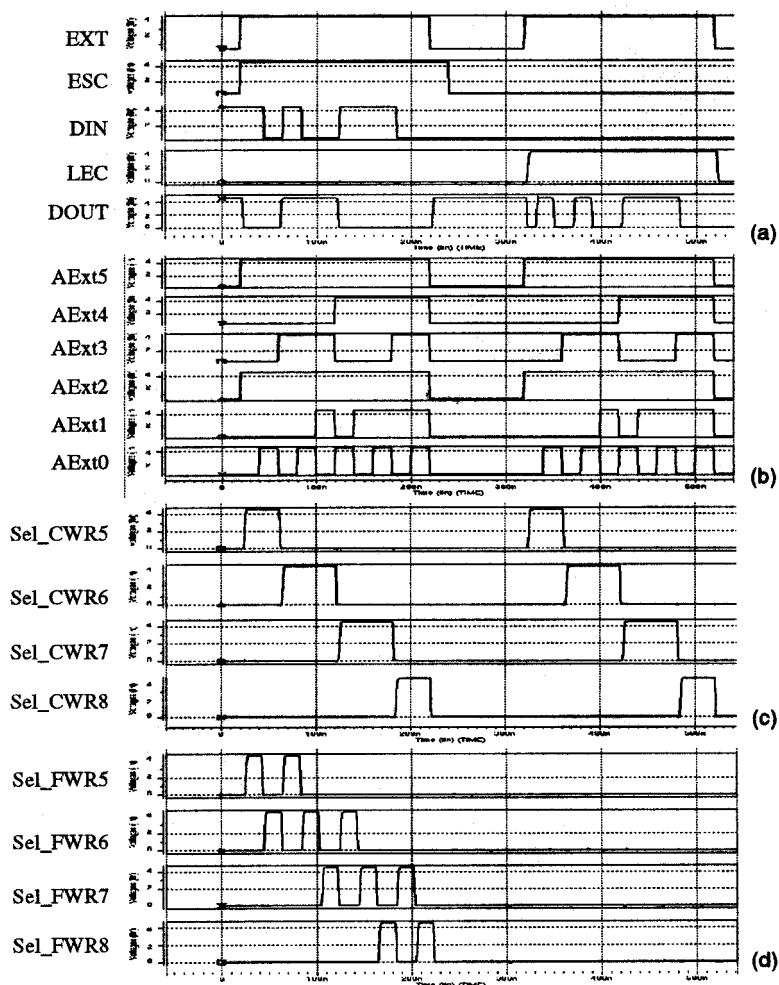


Fig.5.46 Simulación HSPICE para mostrar el funcionamiento del Bloque Programador del Consecuente de Reglas. Accesos externos de escritura: Comandos y líneas de datos en acceso externo (a); líneas de dirección de acceso externo (b); líneas de Selección de Columna en escritura(c); líneas de Selección de Fila en escritura (d).



La Fig.5.48 recoge la evolución de las principales señales que intervienen en los accesos desde el interfaz interno Fig.5.22. La Fig.5.48(a) muestra líneas de dirección interna, AInt[5:0] que evolucionan según el intervalo de interpolación seleccionado en cada instante. En la Fig.5.48(b) se muestra la evolución de las líneas de selección de fila para accesos de lectura Sel\_FRD. La activación de las líneas debido al acceso interno aparece en la parte central de esta gráfica. Nótese que, como se indicó en la descripción del bloque selector de fila, para cada ciclo de acceso interno son activadas dos líneas de selección de fila consecutivas simultáneamente, a diferencia de lo que ocurre durante los ciclos de lectura externos, en los que la selección de filas se realiza de forma individual, como puede verse en la activación de esas

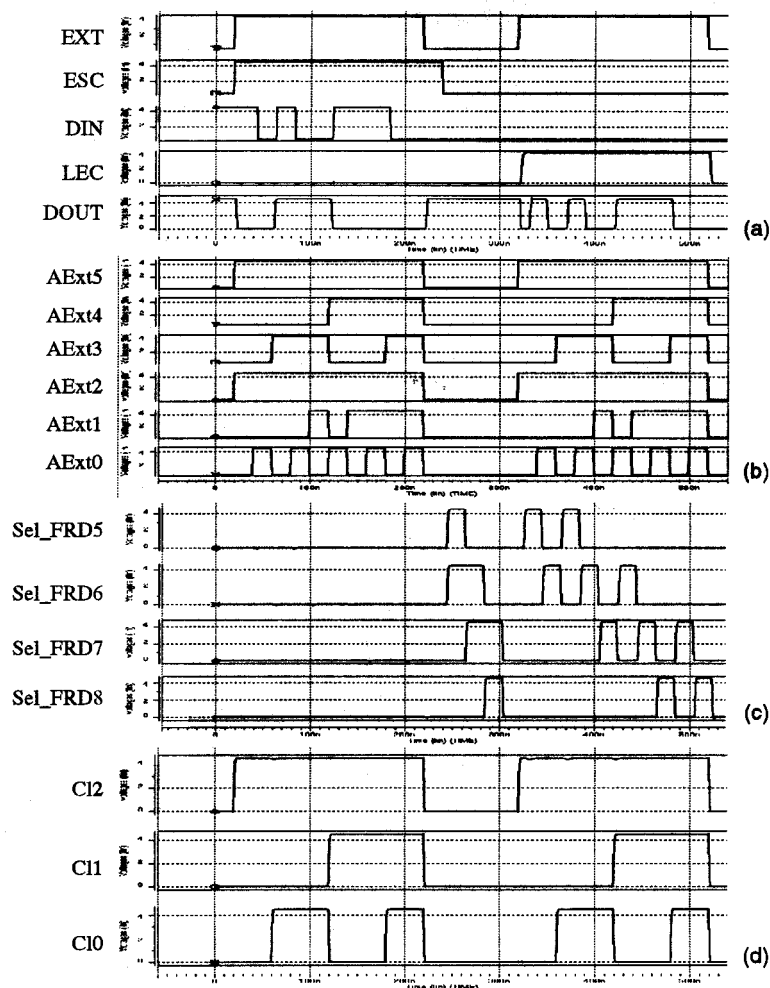


Fig.5.47 Simulación HSPICE para mostrar el funcionamiento del Bloque Programador del Consecuente de Reglas. Accesos externos de lectura: Comandos y líneas de datos en acceso externo (a); líneas de dirección de acceso externo (b); líneas de Selección de Fila en lectura (c); líneas de Selección de Columna en lectura (d).

líneas en la parte derecha de esta gráfica. La Fig.5.48(c) muestra la evolución de las líneas de control C[3:0] de los multiplexores del interfaz interno, Fig.5.22. La secuencia de valores que toman estas líneas puede seguirse con ayuda de la Tabla 5.7 y las figuras Fig.5.41 y Fig.5.42. Finalmente la Fig.5.48(d) muestra la evolución de las señales que representan a los valores *singleton* que son leídos para cada intervalo de interpolación y para cada una de las cuatro reglas que participan en cada intervalo y que constituyen los tres conjuntos de valores de programación del consecuente de reglas  $PC_{C_{55}}$ ,  $PC_{C_{66}}$  y  $PC_{C_{77}}$ , que se recogen en la Fig.5.45(b). En la Fig.5.48(d) se aprecian unas transiciones espurias, *glitches*, que no afectan correcto funcionamiento del sistema y que son consecuencia de la diferencia de retardo en la

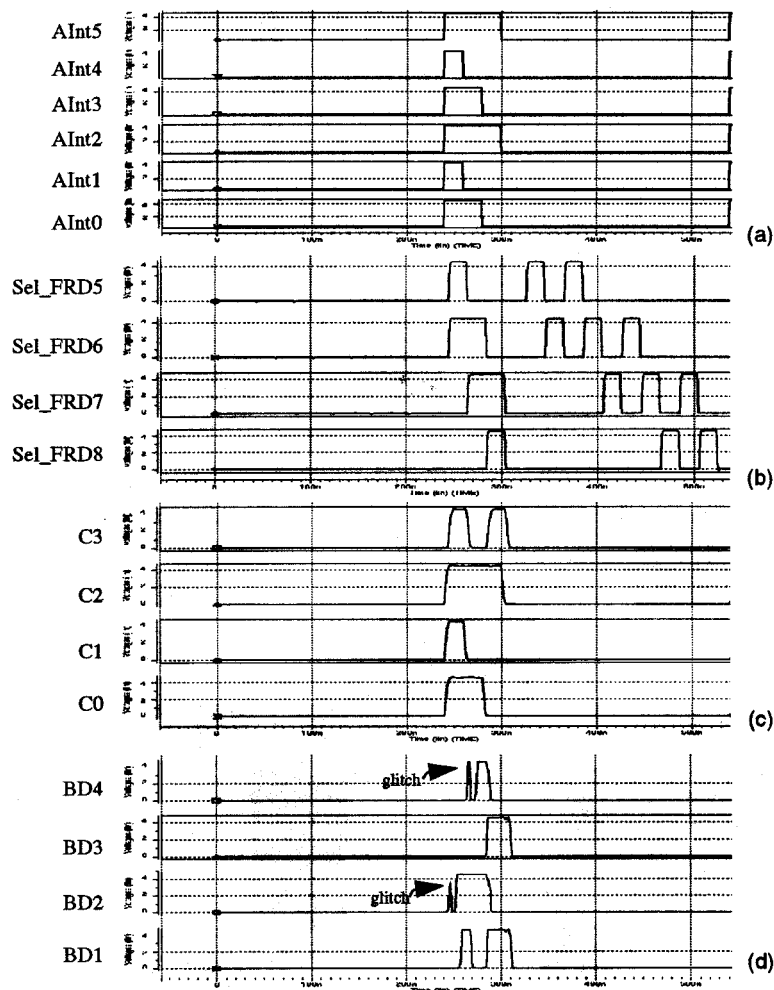


Fig.5.48 Simulación HSPICE para mostrar el funcionamiento del Bloque Programador del Consecuente de Reglas. Accesos internos de lectura: Líneas de dirección de acceso interno (a); líneas de Selección de Fila en lectura (b); líneas de Selección de Columna en lectura (c); líneas de valor *singleton* (d).

evolución de las líneas de control del multiplexor C[3:0], como puede apreciarse con más claridad en el detalle ampliado de esta gráfica que se muestra en la Fig.5.49. En esta figura se observa como la línea de control C3 presenta una evolución un poco más lenta que las demás líneas de control, la razón es clara si se tiene en cuenta que C[2:0] se obtienen directamente de las líneas AInt[2:0] y que C3 se obtiene como una función XOR de las líneas AInt[5:3], como se indicó en la Fig.5.43. En ningún caso esta circunstancia afecta más allá del propio transitorio, al condicionar el tiempo de respuesta de este bloque, dado que finalmente dichas

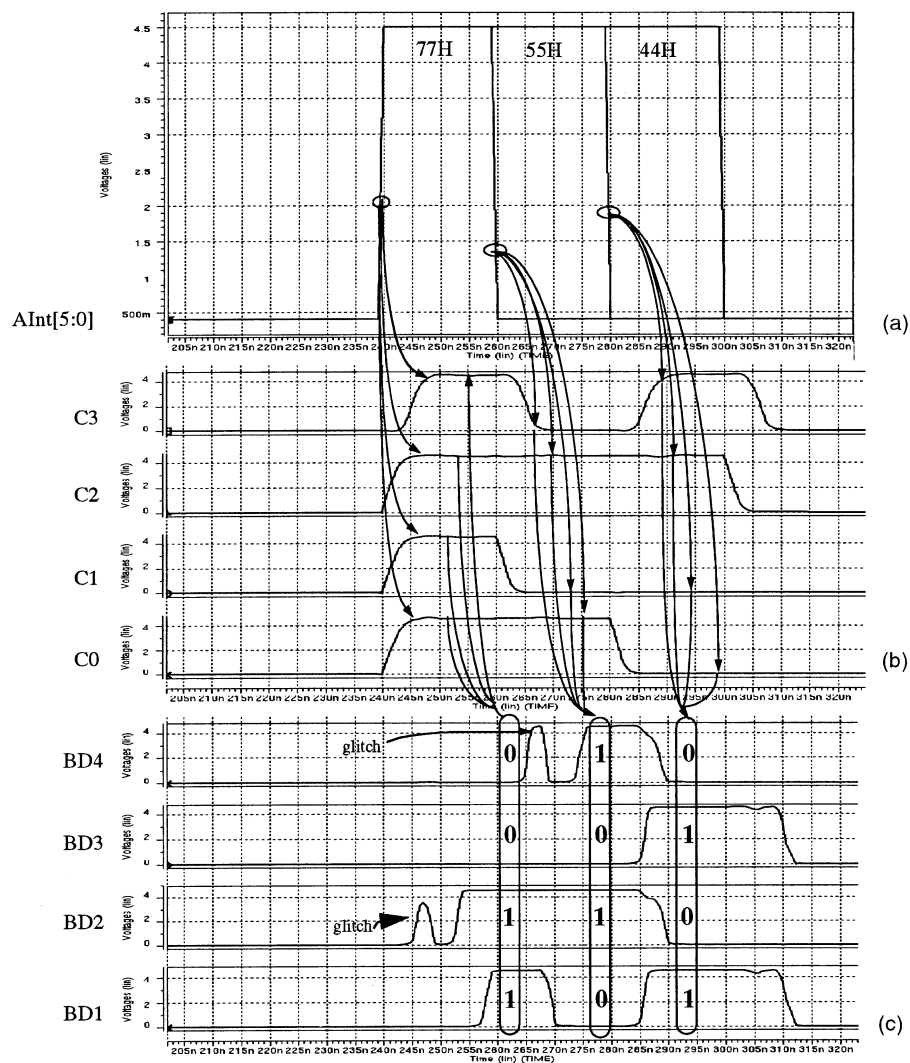


Fig.5.49 Simulación HSPICE para mostrar el funcionamiento del Bloque Programador del Consecuente de Reglas. Detalle de los accesos internos de lectura: Líneas de dirección de acceso interno (a); líneas de Selección de Columna en lectura (b); líneas de valor singleton (c).

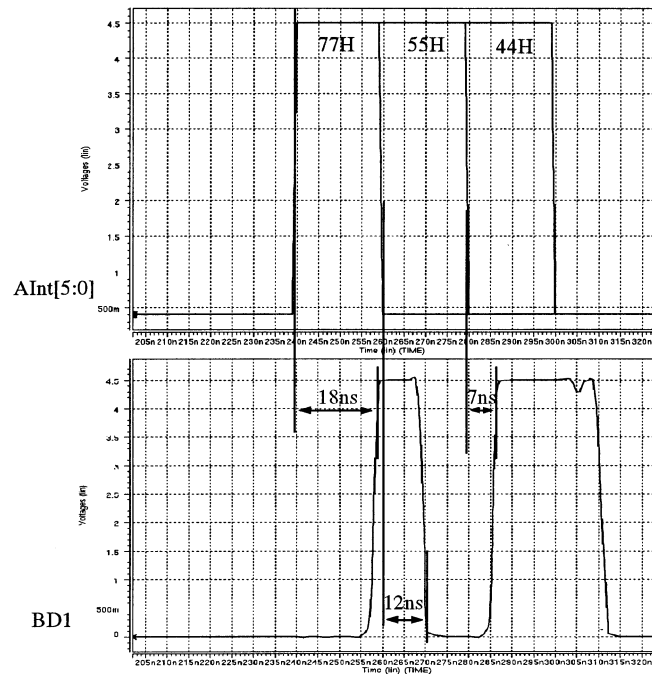


**Fig.5.50 Detalle de la simulación HSPICE: Medida del tiempo de acceso a memoria para lecturas internas.**

señales alcanzan un valor estable como claramente se aprecia en la figura. Para ilustrar el tiempo de respuesta del sistema en el caso más importante para el funcionamiento del controlador que es el de los accesos internos de lectura interna, en la Fig.5.50 se muestra un detalle de simulación en el retardo en la línea de salida de dato *singleton* BD1, tomando como referencia los cambios en las líneas de dirección AInt[5:0]. El caso peor es de 18ns y corresponde al caso en el que el valor de BD1 sigue a la activación de la línea de control C3, como puede observarse en las figuras Fig.5.49(b) y Fig.5.49(c). El tiempo de respuesta del bloque Programador de Consecuente de Reglas completo, esto es incluyendo las celdas de memoria correspondientes a todos los bits que forman un singleton, se ha estimado en 20ns medidos en la simulación HSPICE del fichero *netlist* extraído del fichero de *layout* bloque completo, considerando el caso peor lento especificado por el fabricante para los modelos de los transistores.



## Capítulo 6

### Controlador ejemplo: Integración y Resultados.

#### 1. INTRODUCCIÓN

Para evaluar la viabilidad y la prestaciones de la arquitectura multiplexada propuesta en los capítulos anteriores se ha diseñado y fabricado un chip controlador, que de aquí en adelante será denominado MFCON, (Controlador Fuzzy de arquitectura Multiplexada) con las siguientes características:

- Dos entradas.
- Una salida.
- Sesenta y cuatro reglas.
- Partición en rejilla fija del espacio de entrada.
- Programabilidad de los valores *singleton* que constituyen los consecuentes de las reglas mediante un interfaz compatible con un microprocesador.

Su estructura resulta de particularizar el esquema de la Fig.6.1 para los valores  $M = 2$ ,  $L = 8$  y  $S = 4$  y ha sido realizado con técnicas *full-custom* con la tecnología CMOS estándar de  $0.7\mu\text{m}$ , n-well, un polisilicio y dos metales de ES2. Como herramienta de diseño de *layout* se ha empleado el programa de libre distribución MAGIC sobre una plataforma LINUX. Como herramienta de interfaz con el fabricante el entorno de desarrollo Design Frame

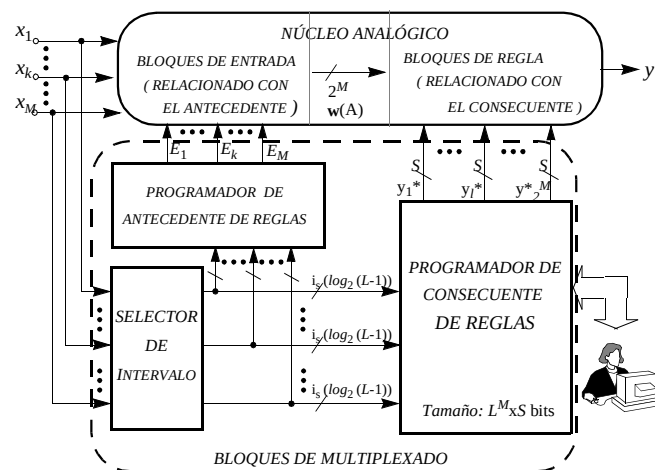


Fig.6.1 Arquitectura global de un Controlador difuso con Núcleo analógico Multiplexado.

Work II, sobre una estación de trabajo Unix. Las simulaciones previas y las posteriores simulaciones de validación *post-layout* se realizaron con la herramienta HSPICE. Las diferentes pruebas realizadas sobre las muestras fabricadas que se describen en este capítulo, están encaminadas a evaluar tanto su respuesta estática, esto es, su capacidad como interpolador de superficies de control tanto lineales como no lineales; como su respuesta dinámica, esto es, su velocidad de operación.

Como sistema de test se ha empleado un puesto de trabajo típico de un laboratorio de electrónica que incluye fuentes de alimentación (TEKTRONIX CPS250, HEWLETT PACKARD 6033A y PHILIPS PM2811), un multímetro digital (HP 3440/A), un generador de señales (HP 33120/A), un osciloscopio digital (TEKTRONIX TDS 520) y un ordenador personal equipado con una tarjeta (DC&C 8255/8253 I/O Card) de entrada-salida. También se ha diseñado y construido un placa de test que incluye, además del controlador difuso y la circuitería de apoyo, los elementos básicos de un sistema de adquisición de datos. Esta placa se conecta a través de la tarjeta de entrada salida instalada en el PC, desde el que se puede controlar y adquirir la respuesta del chip prototipo a las diferentes pruebas de evaluación. Con este sistema se han obtenido los datos para generar las diferentes superficies de control que se presentan en este capítulo para ilustrar la capacidad del controlador como interpolador de superficies de control. Los diferentes programas de test se han desarrollado en lenguaje TURBO C. Para efectuar las medidas de la respuesta dinámica se ha utilizado el osciloscopio digital. Dado que en los capítulos anteriores se ha ido describiendo el diseño de cada uno de los bloques funcionales que constituyen su arquitectura, y se ha particularizado para este controlador ejemplo cuando se han mostrado los esquemas de los circuitos, en este capítulo solamente se describen las principales características del interfaz externo y algunos de los aspectos de la integración del controlador monolítico fabricado. Se describe también el sistema de test y se muestran algunos de los resultados experimentales obtenidos en esas pruebas que ilustran sus principales cualidades.

## 2. DESCRIPCIÓN DEL CHIP CONTROLADOR MFCON.

El controlador que ha sido fabricado se presenta en un encapsulado cerámico **dual-in-line** de 28 pines con la distribución de señales que se muestra en la Fig.6.2(a). Para una mejor comprensión de su funcionalidad en la Fig.6.2(b) se muestran estas señales agrupadas en los dos tipos de interfaces que presenta el chip y, en cada uno de ellos, agrupadas en los correspondientes buses funcionales:

- El **interfaz digital** se corresponde con un interfaz típico de periférico asíncrono diseñado para trabajar en un sistema basado en microprocesador y es empleado para programar y reconfigurar el controlador desde un sistema externo. Este interfaz está

formado por tres buses:

- Un bus de direcciones (señales A(0:5)),
  - Un bus de datos (señales D(0:3))
  - y un bus de control (señales COM, W\_R).
- El **interfaz analógico** agrupa por un lado a las señales propias del sistema de control y por otro a ciertas señales de soporte y test:
    - Las entradas del controlador son  $x_1$  y  $x_2$ , ambas son señales en modo tensión cuyo rango de entrada nominal está comprendido entre los 2 volts. y los 4.8 volts.
    - La salida del controlador es  $z_I$ , que es una señal en modo corriente que proporciona un valor nominal de intensidad comprendido en el intervalo de 0  $\mu A$  a 150  $\mu A$ .
    - El bus denominado POLARIZACIÓN en la Fig.6.2(b) agrupa a cinco señales: IPH, IPN, IPG, IPV y VREF. Las cuatro primeras son fuentes independientes de corriente, mientras que la última es una fuente independiente de tensión. Todas ellas actúan como fuentes de polarización que es necesario proporcionar para el correcto funcionamiento de la circuitería analógica. Al tratarse de un prototipo experimental, en este controlador se ha optado por generar estas fuentes externamente. En una versión comercial podrían ser integradas dentro del propio sistema.
    - La salida  $z_V$  es una señal en modo tensión que resulta de convertir internamente la salida en modo corriente en una señal de tensión empleando un resistor de polisilicio. Su propósito es el de facilitar el test del prototipo permitiendo tomar medidas de la salida del controlador empleando directamente un osciloscopio. No se usa durante el funcionamiento del controlador integrado en un sistema de control.

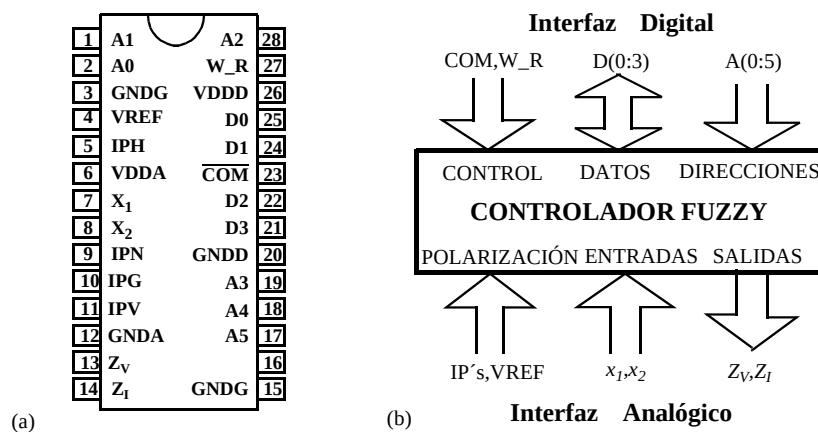


Fig.6.2 MFCON: Pin-out (a) e interfaz agrupado en buses (b).

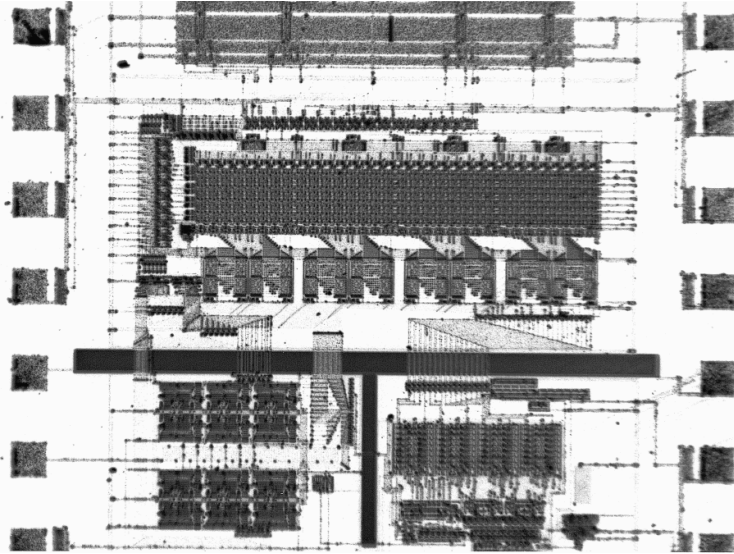


En la Tabla 6.1 se reúnen diferentes datos de cada una de las señales que constituyen el **pin-out** del controlador según se nombran en la Fig.6.2(a).

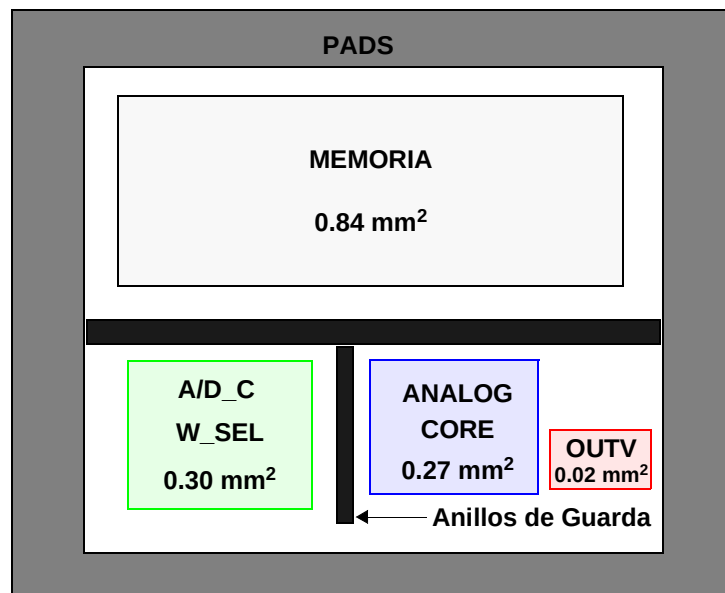
**Tabla 6.1:** Descripción del pin-out del chip MFCN

| Nombre                  | Nº PIN                | Tipo          | Descripción                                                                                    | Valores Nominales                              |
|-------------------------|-----------------------|---------------|------------------------------------------------------------------------------------------------|------------------------------------------------|
| VDDD                    | 26                    | Entrada       | Alimentación digital                                                                           | 5 (volts)                                      |
| GNDD                    | 20                    | Entrada       | Tierra digital                                                                                 |                                                |
| VDDA                    | 6                     | Entrada       | Alimentación analógica                                                                         | 5 (volts)                                      |
| GNDA                    | 12                    | Entrada       | Tierra analógica                                                                               |                                                |
| GNDG                    | 3, 15                 | Entrada       | Conexión a los anillos de guarda                                                               |                                                |
| A[5:0]                  | 17,18,<br>19, 28 ,2,1 | Entrada       | Bus de direcciones de acceso a la memoria que almacena los valores <i>singleton</i>            | Niveles digitales CMOS                         |
| D[3:0]                  | 21, 22,<br>24, 24     | Bidireccional | Bus de datos de acceso a la memoria que almacena los valores <i>singleton</i>                  | Niveles digitales TTL                          |
| W_R                     | 27                    | Entrada       | Indicación de escritura (1-lógico), lectura (0-lógico).                                        | Niveles digitales CMOS                         |
| $\overline{\text{COM}}$ | 23                    | Entrada       | Señal de inicio/fin de ciclo de acceso. (activa a nivel bajo).                                 | Niveles digitales CMOS                         |
| $x_1, x_2$              | 7,8                   | Entrada       | Entradas del controlador difuso. Señal en modo tensión.                                        | Señal analógica<br>Rango [2,4.8] volts         |
| $Z_I$                   | 14                    | Salida        | Salida del controlador difuso. Señal en modo corriente                                         | Señal analógica<br>Rango [0,150] $\mu\text{A}$ |
| $Z_V$                   | 13                    | Salida        | Salida del controlador para test de medida. Señal en modo tensión                              | Señal analógica<br>Rango relativo 1 volt       |
| VREF                    | 4                     | Entrada       | Tensión de alimentación del array resistivo que determina la partición en rejilla.             | 5 (volts)                                      |
| IPH                     | 5                     | Entrada       | Corriente de polarización que determina el ancho de histéresis de las entradas del controlador | Señal analógica<br>20 $\mu\text{A}$            |
| IPN                     | 9                     | Entrada       | Corriente de polarización del circuito normalizador                                            | Señal analógica<br>10 $\mu\text{A}$            |
| IPG                     | 10                    | Entrada       | Corriente de polarización general de la circuitería analógica                                  | Señal analógica<br>-5 $\mu\text{A}$            |
| IPV                     | 11                    | Entrada       | Corriente que controla el nivel de salida del pin $z_v$ .                                      | Señal analógica<br>-30 $\mu\text{A}$           |

La Fig.6.3(a) muestra una microfotografía del chip. El tamaño del dado es de  $5\text{mm}^2$ , tamaño mínimo exigido por el fabricante; sin embargo el consumo de área de los diferentes bloques funcionales, incluidos los canales de rutado, podría ser inferior a este valor. Como dato a tener en cuenta, la suma del tamaño de los diferentes bloques constructivos, excluido el área que ocupa el conexionado entre bloques, está por debajo de  $1\text{mm}^2$ .



(a)



(b)

Fig.6.3 MFCN: Microfotografía (a) y Floor plan (b) .

La Fig.6.3(b) muestra esquemáticamente el *floor plan* del chip MFCON con los diferentes bloques y sus tamaños. Comparando este esquema con la microfotografía se observa que no se ha tenido especial cuidado en optimizar el área ocupada por las líneas de rutado. Además, esta no restricción en cuanto a ocupación de área se ha aprovechado para ubicar los diferentes bloques de forma que estén suficientemente separadas la circuitería digital de la analógica, así como para rellenar ese espacio mediante grandes anillos de guarda, como queda de manifiesto en la Fig.6.3. Estas consideraciones, junto con la utilización de pines y líneas diferentes para las alimentaciones y tierras analógica y digital van encaminadas a minimizar el efecto de las interferencias eléctricas propias de los sistemas en modo mixto, siguiendo las habituales recomendaciones para este tipo de diseños [Inge97].

Las principales características del controlador se resumen en la Tabla 6.2

**Tabla 6.2:** Principales características del controlador MFCON

| CARACTERISTICA              | DATO                             |
|-----------------------------|----------------------------------|
| COMPLEJIDAD                 | 64reglas @ 2 entradas@ 1 salidas |
| TENOLOGÍA                   | 0.7 $\mu$ m CMOS                 |
| CONSUMO DE POTENCIA         | 16mW@5V                          |
| RETARDO ENTRADA/SALIDA      | 500 ns                           |
| AREA                        | 2.65mm <sup>2</sup> (sin pads)   |
| INTERFAZ (entradas@salidas) | tensión@corriente                |
| ALIMENTACIÓN                | 5V                               |
| PRECISION                   | 7.8 % (3 $\sigma$ )              |

El consumo de potencia se ha obtenido sensando las corrientes de entrada de cada uno de los terminales de alimentación analógico y digital. Por su parte, por lo que respecta a la precisión, ésta viene determinada por la precisión con que ha sido diseñado el *Núcleo Analógico*. Este aspecto ya ha sido discutido en el Capítulo 3. Como allí se señala, el controlador ejemplo ha sido diseñado para tener en cuenta un desapareamiento entre transistores con una desviación estándar ( $\sigma$ ) asociada a la salida que equivale a un error del 2.6% del fondo de escala.

Resulta ilustrativo comparar los datos de velocidad, área y consumo de potencia de MFCON con los datos del controlador puramente analógico reportado en [Vida96a]. Mientras que para ese controlador, (diseñado con la misma arquitectura que la del *Núcleo Analógico* de MFCON y con una tecnología CMOS estándar de 1 $\mu$ m), que posee dos entradas y que trabaja con 16 reglas, se tiene un retardo de 470ns, un consumo de 8.6mW, y ocupa un área de 1.6mm<sup>2</sup>, MFCON trabaja con 64 reglas, con prácticamente el mismo retardo entrada/salida, un consumo algo inferior al doble, 16mW, y un tamaño de 2.65mm<sup>2</sup>.

### 3. SISTEMA DE TEST.

Las Fig.6.5 muestra el diagrama de bloques de la placa que ha sido construida para realizar las pruebas del chip controlador. Esta placa sirve de soporte a dicho chip y contiene la circuitería que genera la fuentes de polarización necesarias para su buen funcionamiento. Mediante dos conectores de cable plano de 40 pines y la circuitería necesaria, la placa proporciona un interfaz con una tarjeta de entrada salida (DC&C 8255/8253 I/O Card) instalada en un PC. Este interfaz con el PC permite por una parte accesos de lectura y escritura para programar la memoria del chip controlador. También proporciona accesos de escritura sobre dos chip conversores D/A (DAC0800) que permiten excitar las entradas de tensión del controlador con valores conocidos, establecidos desde el programa de control. Proporciona además accesos de lectura sobre un chip conversor A/D (ADC0803) de forma que es posible adquirir los valores de salida del controlador. De esta forma la placa contiene los dispositivos básicos para poder trabajar como un sistema simple de adquisición de datos.

La Fig.6.4(a) muestra una fotografía de esta placa. Junto a ella, la Fig.6.4(b) muestra un diagrama de bloques que sirve para identificar cada uno de los componentes que la constituyen. A continuación se detallan algunos de los aspectos más destacables de esta placa de test.

#### 3.1. Interfaz de programación y de adquisición de datos

Como muestra la Fig.6.4, la placa de test posee dos conectores de cable plano de 40 pines a través de los cuales se establece la conexión física con la tarjeta de I/O instalada en el PC, y que conjuntamente soportan el interfaz de programación y adquisición de datos. Entre otras prestaciones, que pueden ser consultadas en el manual de usuario, la tarjeta de I/O proporciona seis puertos de ocho bit, tres por cada conector, (1PA[0:7], 1PB[0:7], 1PC[0:7], pines 3 al 27 sucesivamente en el conector 1; 2PA[0:7], 2PB[0:7], 2PC[0:7], pines 3 al 27 sucesivamente en el conector 2, configurables por *software* en tres modos básicos que ofrecen una amplia flexibilidad de funcionamiento. Para esta aplicación se ha seleccionado el **modo 0** o **modo básico de entrada salida**, el cual permite configurar cada uno de los puertos como puertos de entrada o salida, mediante el acceso de programación adecuado a sendos registros de control.

En la Fig.6.6 se ilustra la configuración entrada o salida seleccionada para cada uno de los puertos, así como su utilización. Los puertos 1PA, 2PA y 2PB han sido configurados como puertos de salida, también se configuran como puerto de salida los cuatro bits menos significativos de los puertos 1PC y 2PC; mientras que el puerto 1PB y los cuatro bits más significativos de los puertos 1PC y 2PC son configurados como puertos de entrada. Por lo que respecta a su utilización, los puertos 1PA y 2PB se usan para escribir los valores las entradas  $x_1$  y  $x_2$  del chip controlador difuso que serán convertidos en valores analógicos por los

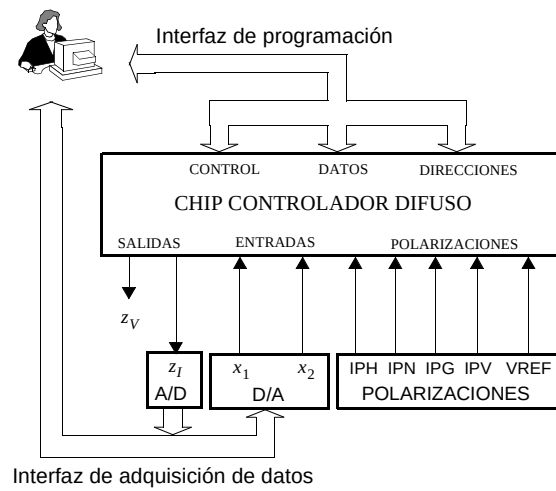


Fig.6.5 Diagrama de bloques de las placa de test.

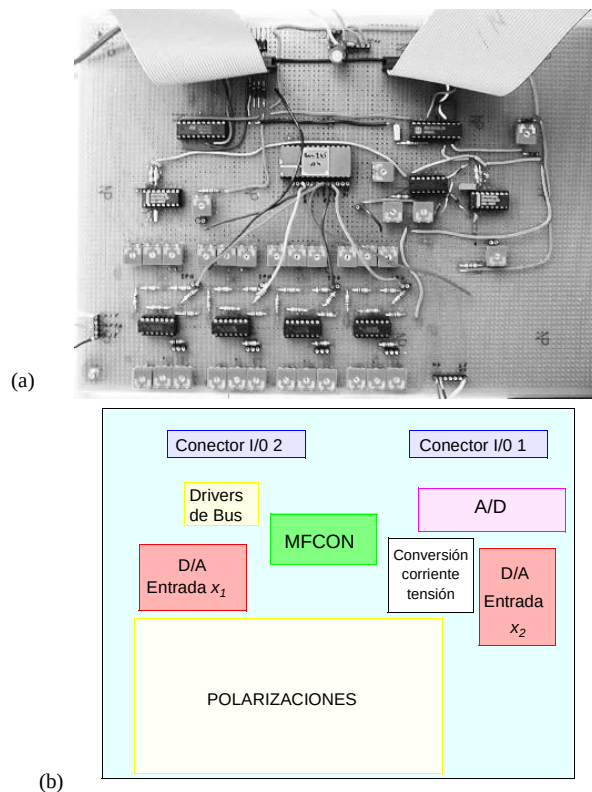


Fig.6.4 Placa de Test: Fotografía (a); esquema de componentes (b).

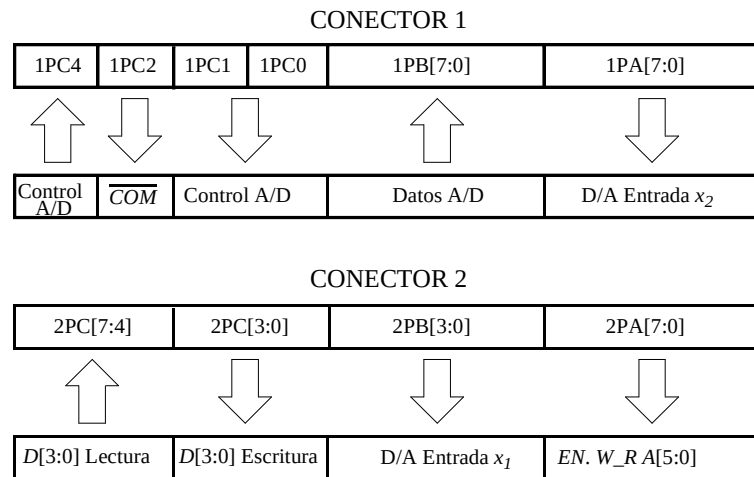


**Fig.6.6 Interfaz de programación y adquisición de datos: Configuración de puertos de I/O y asignación a los diferentes elementos de la placa de test.**

circuitos de conversión D/A, mientras que el puerto 1PB permite leer la salida que proporciona el controlador tras ser convertida a un valor digital por el circuito conversor A/D. Las líneas 1PC1 y 1PC0 se emplean para controlar las señales  $\overline{WR}$  y  $\overline{RD}$ , de escritura y lectura respectivamente, del circuito conversor A/D, mientras que 1PC4 se utiliza para sensar la respuesta  $\overline{INTR}$  de éste, todas ellas necesarias para efectuar la conversión A/D. El puerto 2PA se emplea para generar las direcciones A[5:0] del chip controlador difuso, así como la señal  $W\_R$ , y la señal  $EN$  que controla la dirección los *Drivers Triestado Bidireccionales* del Bus datos D[3:0]. La línea 1PC2 permite controlar la señal de ciclo de acceso  $\overline{COM}$  a la memoria del controlador. Por último el puerto 2PC se emplea para manejar los datos de programación del chip controlador, los bits 2PC[3:0] proporcionan los datos en los accesos de escritura, mientras que los bits 2PC[7:4] reciben los datos en los accesos de lectura.

### 3.2. Generación de las polarizaciones externas

Cada una de las cuatro fuentes de polarización de intensidad IPH, IPN, IPG, IPV se ha diseñado siguiendo el modelo del circuito de la Fig.6.7, donde se tiene que

$$I_{pol} = \left( -\frac{R_2}{R_1 R_5} \right) V_{pol} \quad (6.1)$$

siempre que las resistencias verifiquen la relación

$$\frac{R_2}{R_1} = \frac{R_4 + R_5}{R_3} \quad (6.2)$$

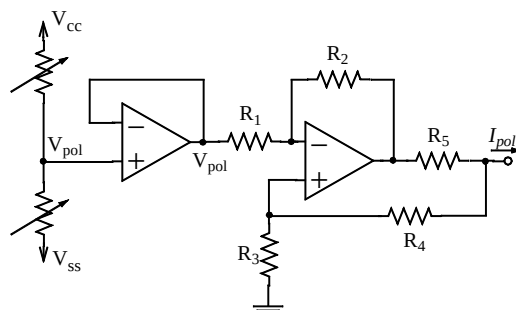


**Fig.6.7 Circuito para la generación de las fuentes de intensidad de polarización externas**

En la Tabla 6.3 se resumen los valores nominales de las resistencias empleadas en el diseño de cada una de las fuentes de polarización.

**Tabla 6.3:** Valores nominales de las resistencias empleadas en el diseño de las polarizaciones.

| Fuente/Valor    | $R_1(k\Omega)$ | $R_2(k\Omega)$ | $R_3(k\Omega)$ | $R_4(k\Omega)$ | $R_5(k\Omega)$ |
|-----------------|----------------|----------------|----------------|----------------|----------------|
| IPH/20 $\mu$ A  | 1              | 10             | 2x10           | 100            | 100            |
| IPN/10 $\mu$ A  | 1              | 10             | 2x100          | 1000           | 1000           |
| IPG/-5 $\mu$ A  | 1              | 10             | 2x100          | 1000           | 1000           |
| IPV/-30 $\mu$ A | 1              | 10             | 2x10           | 100            | 100            |

Por su parte, la fuente de tensión VREF se ha obtenido directamente de la fuente de alimentación analógica externa de 5 volts, la misma que se conecta a la entrada de alimentación de chip controlador VDDA.

### 3.3. Circuitos de control y adquisición de datos

A continuación se describen las principales características y el modo de empleo en la placa de test de los diferentes circuitos integrados relacionados con las funciones de control y adquisición de datos.

#### 3.3.1. Familia de conversores digital a analógico de 8 bits DAC0800/DAC0801/DAC0802.

En la placa de test se emplean dos de estos circuitos integrados para establecer valores analógicos en las entradas  $x_1$  y  $x_2$  del controlador difuso desde en programa de test. Como ya se ha mencionado, cada uno de ellos está asociado a un puerto de salida de ocho bits de la placa de I/O de forma que su función es convertir el valor que desde el programa de test se escribe en ese puerto en un valor analógico. Este valor permanece estable mientras el valor escrito en dicho puerto no sea alterado.

La familia DAC0800 de conversores [NS-DAC] proporciona una salida analógica en modo corriente que ha de ser convertida en una señal de tensión para ser conectada al controlador. Dado que las entradas del controlador son nudos de alta impedancia esta conversión se lleva a cabo de forma simple mediante una resistencia  $R_x$  de valor adecuado, en la placa esta resistencia resulta de la asociación en serie de un resistencia de valor constante y un potenciómetro de ajuste.

La Fig.6.8 muestra el esquema de conexión de este circuito según es utilizado en la placa de test. La salida analógica  $x_i$  se obtiene según la expresión

$$x_i = V_{DD} - (R_x \cdot I_o) \quad (6.3)$$

donde  $I_o$  depende de  $V_{ref}$ ,  $R_{ref}$

$$I_{FS} = I_o + \bar{I}_o$$

$$I_{FS} = \frac{V_{ref}}{R_{ref}} \times \frac{255}{256} \quad (6.4)$$

y del valor de la entrada digital D[1:8], según los datos de las especificaciones que aparecen en la documentación proporcionada por el fabricante.

Para  $V_{ref} = 10 \text{ volts}$  y  $R_{ref} = 5k\Omega$  se tiene que  $I_{FS} = 1.992 \text{ mA}$ . Por lo que  $I_o = 1.992 \text{ mA}$  para D[1:8] = FFh (valor de escala digital completa expresado en hexadecimal), mientras que se tiene  $I_o = 0 \text{ mA}$  para D[1:8] = 00h, (cero de la escala digital). Estos valores permiten ajustar el valor de la resistencias  $R_x$  para cubrir adecuadamente el rango de valores de tensión de las entradas del controlador.

Por lo que respecta a la sensibilidad, según la información de la hoja de especificaciones 1-bit supone un incremento  $\Delta I_o$  en la corriente de salida  $I_o$  de 8mA ; lo que supone un incremento en el valor de las entradas,  $\Delta x_i$  en la forma  $\Delta x_i = \Delta I_o \cdot R_x$ .

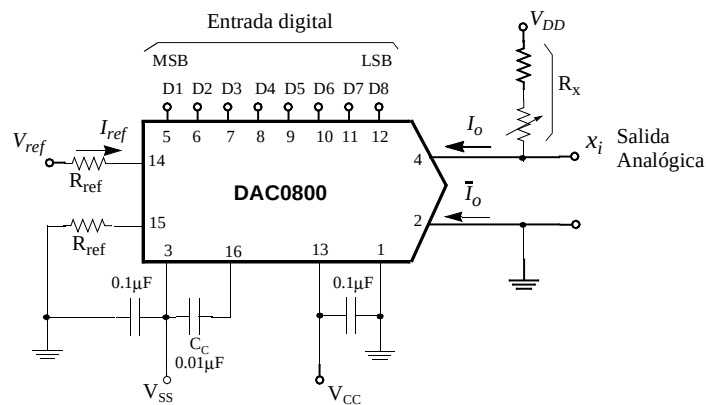


Fig.6.8 Circuito de conversión digital-analógico.



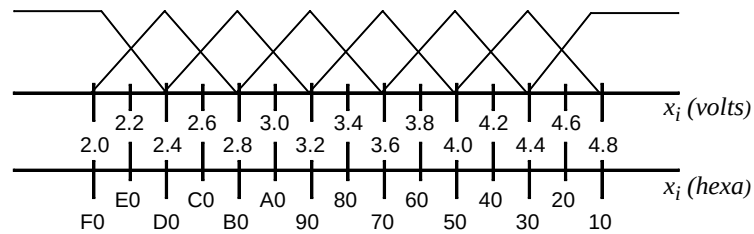


**Fig.6.9 Universo de discurso unidimensional de MFCON**

La Fig.6.9 muestra el rango de valores nominales de tensión (en voltios) y su correspondiente palabra digital (en formato hexadecimal), para el universo de discurso unidimensional de MFCON.

### 3.3.2. Conversor Analógico digital de 8-bits compatible con microprocesador ADC0803

En la placa de test se emplea este circuito integrado para transformar en formato digital el valor analógico de tensión al que es convertido la salida en corriente  $z_I$  de controlador difuso. El dato digital es leído desde el puerto de la placa de I/O configurado para tal efecto, controlando adecuadamente las señales del interfaz compatible con microprocesador que proporciona este circuito integrado [NS-ADC].

La conversión corriente-tensión de la salida  $z_I$  se realiza empleando la sonda de intensidad que muestra la Fig.6.10(a). Este circuito verifica que

$$\begin{aligned} V_+ &= V_{pol} \\ V_- &= V_{pol} - (R_I \cdot z_I) \end{aligned} \quad (6.5)$$

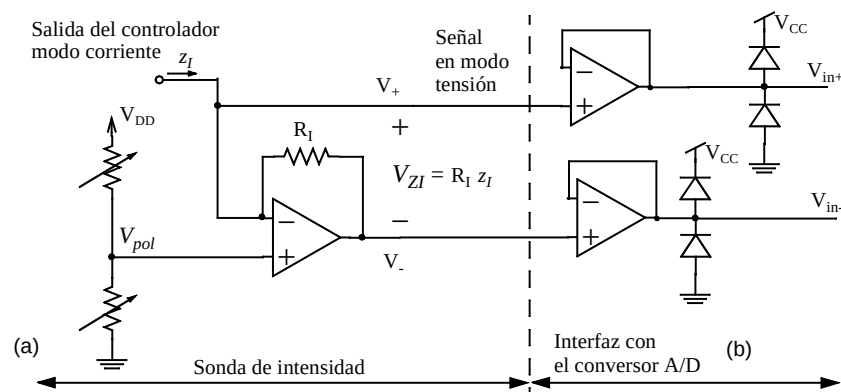


**Fig.6.10 Circuitos de interfaz entre el controlador y el conversor A/D: Sonda de corriente (a); seguidores y diodos de protección (b).**

de modo que la tensión  $V_{ZI}$ , esto es, la caída de tensión en la resistencia  $R_I$ , resulta ser proporcional a la corriente de salida del controlador  $Z_I$ .

$$V_{ZI} = V_+ - V_- = V_{pol} - (V_{pol} - R_I \cdot Z_I) = R_I \cdot Z_I \quad (6.6)$$

En la Fig.6.10(b) se muestran también los seguidores de tensión, y los diodos de protección que se emplean como interfaz con las entradas en modo diferencial del conversor A/D. De este modo se tiene que  $V_+ = V_{in+}$  y  $V_- = V_{in-}$ , donde  $V_{in+}$  y  $V_{in-}$  son las entradas en modo diferencial del conversor A/D.

Según la expresión (6.6), para el valor mínimo de  $Z_I$ , esto es  $0 \mu A$ ,  $V_{in+} = V_{pol}$  y  $V_{in-} = V_{pol}$  de modo que  $V_{ZI} = V_{pol} - V_{pol} = 0$ . Por otra parte, para el valor máximo de  $Z_I$ , esto es  $Z_{I(max)} = 150 \mu A$ ,  $V_{in+} = V_{pol}$  y  $V_{in-} = V_{pol} - (Z_{I(max)} \cdot R_I)$ , de modo que  $V_{ZI}$  toma el valor  $V_{ZI} = Z_{I(max)} \cdot R_I$ .

El rango en tensión para cada una de las entradas  $V_{in+}$  y  $V_{in-}$ , del conversor A/D está comprendido entre los 0 y 5 volts, para una tensión de alimentación  $V_{CC}$  de 5 volts. Dado que para que el conversor funcione correctamente cada una de estas entradas,  $V_{in+}$  y  $V_{in-}$ , han de ser tensiones positivas, se ha elegido para  $V_{pol}$  el valor de 2.5 volts, y  $R_I = V_{pol} / Z_{I(max)}$ . De esta forma el rango de valores de intensidad (0,150)  $\mu A$  es transformado en el rango de tensión diferencial [0,2.5] volts, con  $V_{in+} \geq 0$  y  $V_{in-} \geq 0$  para todos los posibles valores de  $Z_I$ . El valor escogido para  $V_{pol}$  resulta también adecuado para polarizar en nudo de salida  $Z_I$  del chip controlador. Con estos valores se tiene como valor nominal para  $R_I$  16.6k $\Omega$ .

La resolución del conversor ADC0803 para la escala completa es de 8 bits con un error de  $\pm 1/2$  LSB. Esto supone que LSB equivale aproximadamente a 20mV, y dado el valor de  $R_I$ , supone detectar un incremento de corriente de 1.2 $\mu A$ . Esta resolución resulta suficiente para la medida de los valores estáticos de salida del controlador cuya cota de error estimado es del 10% de su escala completa.

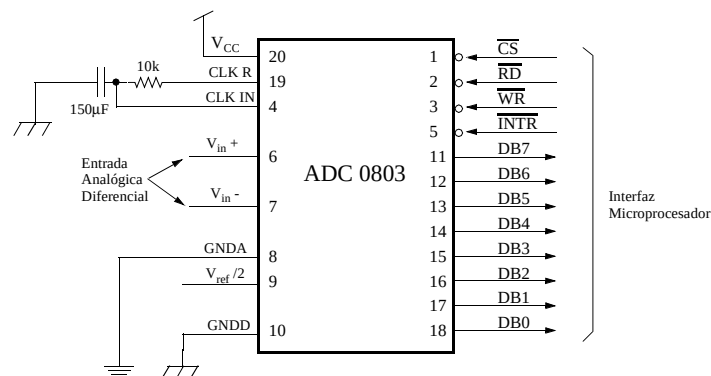


Fig.6.11 Circuito de conversión analógico-digital

La Fig.6.11 muestra el esquema de conexión de este circuito según es utilizado en la placa de test. En este diseño el funcionamiento del circuito ADC0803 es controlado desde el programa de pruebas a partir de las señales  $\overline{INTR}$ ,  $\overline{WR}$  y  $\overline{RD}$  de su interfaz con un microprocesador. El reloj de funcionamiento es generado dentro del propio circuito con la ayuda del circuito RC aparece en la Fig.6.11 conectado a los terminales CLK R y CLK IN.

### 3.4. Programas de pruebas del sistema de test.

Para controlar la tarjeta de I/O y la placa de test se han desarrollado un conjunto de rutinas de bajo nivel y programas básicos en el lenguaje de programación TURBO C. Las diferentes rutinas permiten por una parte, configurar adecuadamente la tarjeta de I/O, estableciendo su modo de operación y las direcciones asignadas a cada uno de los puertos, y por otra efectuar los accesos básicos a los diferentes puertos, descritos en el apartado 3.1., de una forma sencilla, de manera que resulte cómodo controlar los diferentes elementos de la placa de test. Empleando estas rutinas básicas se han realizado los programas que realizan las diferentes tareas que se reúnen en un menú principal. Estas tareas son:

- Reconfigurar y actualizar los parámetros básicos del sistema de adquisición de datos, que han de ser medidos y ajustados en la placa de test y deben ser conocidos por los programas para realizar cálculos. Estos datos deben ser ajustados al inicio, y/o pueden ser modificados, siempre que se necesite, durante cada sesión de trabajo.
- Reconfigurar, actualizar y validar el sistema de memoria que almacena los valores *singleton* en el chip controlador. El contenido de la memoria puede ser actualizado a partir de patrones fijos, generados por el propio programa; o bien a partir del contenido de un fichero de datos.
- Realizar accesos individuales al interfaz analógico de MFCON de forma que es posible tomar medidas de puntos individuales de la superficie de control que una determinada configuración del controlador genera.
- Adquisición de datos de la superficie de control. Se realiza un barrido unidimensional o bidimensional sobre el rango de valores de entrada y con el número de puntos que se indica. En este caso la información es almacenada en un fichero, para su posterior visualización.

En cualquier caso toda la sesión de trabajo queda registrada en un fichero para su posterior tratamiento.

#### 4. RESULTADOS EXPERIMENTALES

A continuación se muestran algunos de los resultados experimentales que ilustran el comportamiento y las principales características del controlador MFCON. Estos resultados se presentan organizados en dos apartados: En primer lugar se muestran resultados del comportamiento estático o dc; a continuación se muestran resultados que ilustran su comportamiento dinámico. En todos los experimentos el chip controlador funciona integrado en la placa de test que se ha descrito en los apartados anteriores. En cada uno de los casos que se presentan se indica cómo ha sido efectuada la medición.

##### 4.1. Comportamiento dc del chip controlador difuso MFCON

Con este tipo de pruebas se pretende evaluar el comportamiento estático del MFCON de forma que, para diferentes configuraciones, se obtenga la respuesta del circuito para todos los valores del espacio de las entradas. Las pruebas realizadas cubren dos aspectos básicos:

- Obtención de los valores de intensidad de corriente eléctrica correspondientes a cada uno de los valores *singleton*.
- Obtención de superficies de control que ilustren el comportamiento global de MFCON como sistema de interpolación.

Por una parte se han medido los valores de intensidad de corriente eléctrica que proporciona el controlador a su salida para cada uno de los valores *singleton* y se han comparado con los valores nominales. Esta comparación da el error de la implementación que ha de estar incluido dentro del margen de error asumido durante la fase de diseño, 10% sobre el fondo de escala. Por otra parte se han obtenido diferentes superficies de control correspondientes a diferentes configuraciones de MFCON efectuando barridos bidireccionales de sus variables de entrada  $x_1$  y  $x_2$ , empleando la placa y los programas de test descritos en apartados anteriores. Por último también se han obtenido medidas de secciones de superficies de control sensando la salida en tensión  $Z_V$  de MFCON con un osciloscopio digital. A continuación se muestran los resultados de estas pruebas.

##### 4.1.1. Medida de la salida para los diferentes valores *singleton*.

Para este tipo de pruebas se ha configurado MFCON para realizar superficies planas horizontales con los distintos valores *singleton* y se ha medido la corriente a la salida  $Z_I$  y la tensión de salida  $Z_V$  del controlador en diferentes puntos de la superficie empleando el multímetro digital para la primera, y el osciloscopio digital para la segunda. La precisión de cada medida corresponde a la del equipo de medida según la escala utilizada en cada caso para realizar dicha medida: 0.1 $\mu$ A para la medida de intensidad, 1mV para las medidas de tensión.

Como dato final se ha tomado el valor medio de las diferentes medidas. Estos datos se resumen en la Tabla 6.4 en la que se indica además su dispersión entre los diferentes puntos de la superficie.

**Tabla 6.4:** Valores medidos de  $Z_I$  y  $Z_V$  para cada valor *singleton*.

| <i>Singleton</i> | 0                 | 1                 | 2                 | 3                 |
|------------------|-------------------|-------------------|-------------------|-------------------|
| $Z_I$ ( $\mu$ A) | $0.0 \pm 0.1$     | $8.4 \pm 0.3$     | $16.9 \pm 0.5$    | $25.3 \pm 0.7$    |
| $Z_V$ (volts)    | $1.834 \pm 0.010$ | $1.882 \pm 0.010$ | $1.920 \pm 0.012$ | $1.966 \pm 0.010$ |
| <i>Singleton</i> | 4                 | 5                 | 6                 | 7                 |
| $Z_I$ ( $\mu$ A) | $35.1 \pm 0.4$    | $44.1 \pm 0.6$    | $52.0 \pm 0.5$    | $60.5 \pm 0.4$    |
| $Z_V$ (volts)    | $2.034 \pm 0.014$ | $2.068 \pm 0.012$ | $2.102 \pm 0.014$ | $2.154 \pm 0.014$ |
| <i>Singleton</i> | 8                 | 9                 | 10                | 11                |
| $Z_I$ ( $\mu$ A) | $96.2 \pm 0.5$    | $106.0 \pm 0.4$   | $114.7 \pm 0.5$   | $121.9 \pm 0.4$   |
| $Z_V$ (volts)    | $2.333 \pm 0.015$ | $2.390 \pm 0.014$ | $2.444 \pm 0.016$ | $2.474 \pm 0.016$ |
| <i>Singleton</i> | 12                | 13                | 14                | 15                |
| $Z_I$ ( $\mu$ A) | $131.7 \pm 0.6$   | $140.1 \pm 0.4$   | $148.6 \pm 0.5$   | $156.7 \pm 0.6$   |
| $Z_V$ (volts)    | $2.530 \pm 0.018$ | $2.566 \pm 0.018$ | $2.608 \pm 0.020$ | $2.640 \pm 0.020$ |

A modo de ejemplo, la Fig.6.12 muestra la superficie plana que corresponde al valor *singleton* 3, en ella se puede apreciar que la dispersión de valores a lo largo del espacio de entrada está entre los valores que se señalan en la Tabla 6.4 para este valor *singleton*.

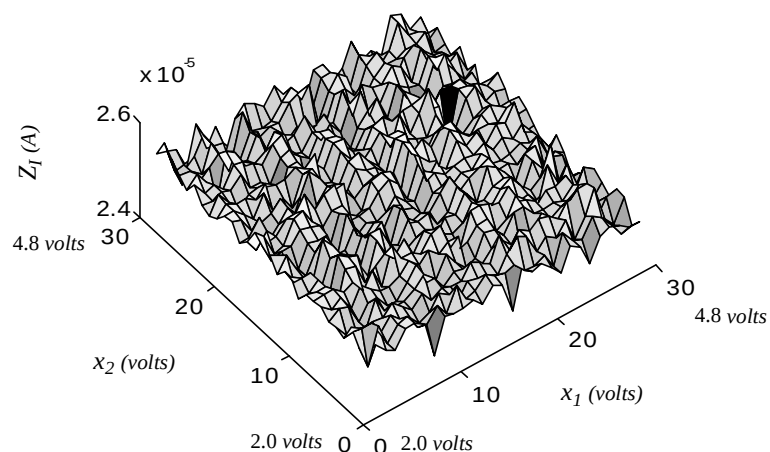


**Fig.6.12** Superficie de control plana de valor *singleton* 3

La curva de ajuste de mínimos cuadrados de estos valores *singleton* resulta

$$Z_I = 11.24 \cdot S - 5.67 \quad (\mu A) \quad (6.7)$$

donde la variable  $S$  toma valores en el rango de valores *singleton*.

En la Fig.6.13 se representan los valores  $Z_I$  recogidos en la Tabla 6.4, junto a la curva de ajuste de la expresión (6.7), frente a los valores *singleton*. En ella se aprecia cómo todos valores  $Z_I$  que proporciona MFCON están incluidos en la banda de desviación del 10% sobre el fondo de escala, ( $\pm 15\mu A$ ), que aparece delimitada por las rectas en trazo discontinuo que muestra la Fig.6.13.

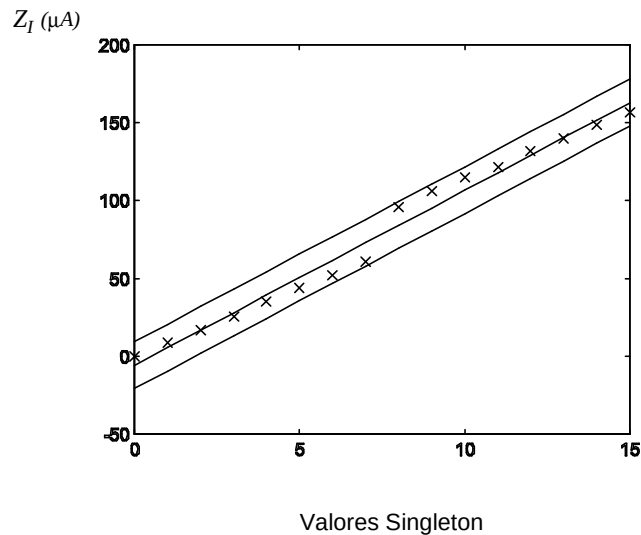


**Fig.6.13 Representación gráfica de  $Z_I$  frente a los valores *singleton*: + Valores nominales; x valores medidos en MFCON.**

En la Fig.6.14 se representan los valores de  $Z_V$  (volts) frente a los de  $Z_I$  ( $\mu A$ ), que aparecen en la Tabla 6.4, además de la curva de ajuste por mínimos cuadrados obtenida a partir de estos datos y cuya expresión resulta

$$Z_V = (0.0052 \cdot Z_I) + 1.878 \quad (volts) \quad (6.8)$$

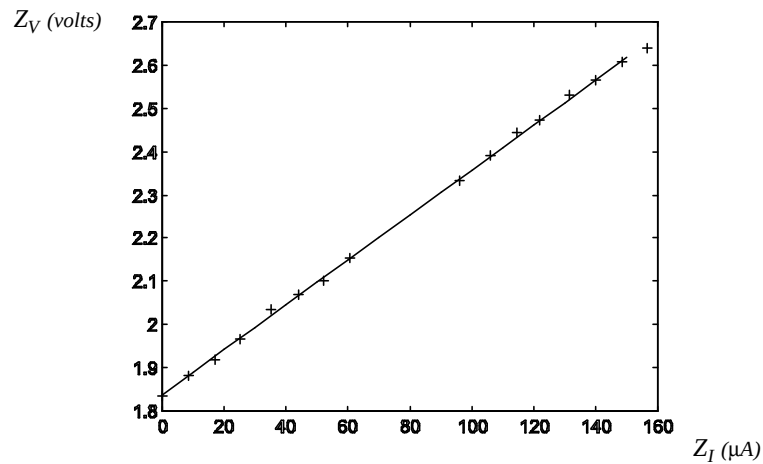


Fig.6.14 Representación gráfica de  $Z_V$  frente a  $Z_I$ : En trazo continuo curva ajustada; + valores medidos en MFCN.

Finalmente en la Fig.6.15 se representa  $Z_V$  (volts) frente a los valores *singleton*, junto a los valores que se obtienen a partir de la expresión (6.8).

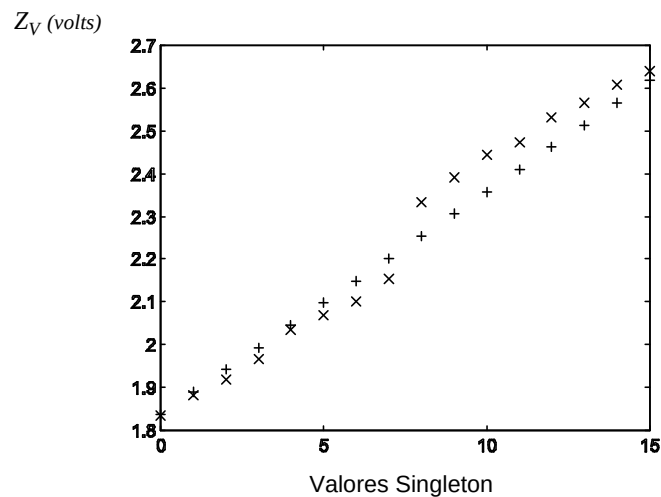


Fig.6.15 Representación gráfica de  $Z_V$  frente a los valores singleton: + Valores según la curva de ajuste; x valores medidos en MFCN.

## 4.1.2. Ejemplos de Superficies de Control

Para ilustrar el comportamiento de MFCON como interpolador de funciones, a continuación se muestran diferentes ejemplos de superficies de control lineales y no lineales generadas por este controlador cuando es configurado según la Matriz de Reglas difusas que acompaña a cada ejemplo. Esta matriz de reglas muestra los valores *singleton*, que en este contexto representan los puntos que sirven de base para la interpolación de la superficie completa.

En todos estos ejemplo las superficies han sido generadas con MATLAB®, a partir de los datos obtenidos mediante la placa y el correspondiente programa de test :

- **Ejemplo 1.** Superficies generadas a partir de una Matriz de Reglas que alterna valores *singleton* máximos y mínimos.

La Fig.6.16(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ve claramente como los valores *singleton* van alternando entre su valor mínimo (0) y su valor máximo (15). La superficie de control resultante, Fig.6.17(a), presenta un perfil de valles y picos alternados. La Fig.6.16(b) muestra una perspectiva vertical de esta superficie de control que ilustra el buen alineamiento de los puntos valle y pico. Aquí hay que recordar que MFCON genera esta superficie a partir de la multiplexación de un *Núcleo Analógico* que trabaja en cada instante en un intervalo de interpolación. La Fig.6.17(b) muestra un detalle de la superficie de control resultante en cada uno de dichos intervalos. Por tanto cada uno de los picos de esta superficie reproduce la forma de las funciones base bidimensional que emplea el algoritmo difuso y es el resultado de la contribución de esta a los cuatro intervalos adyacentes.

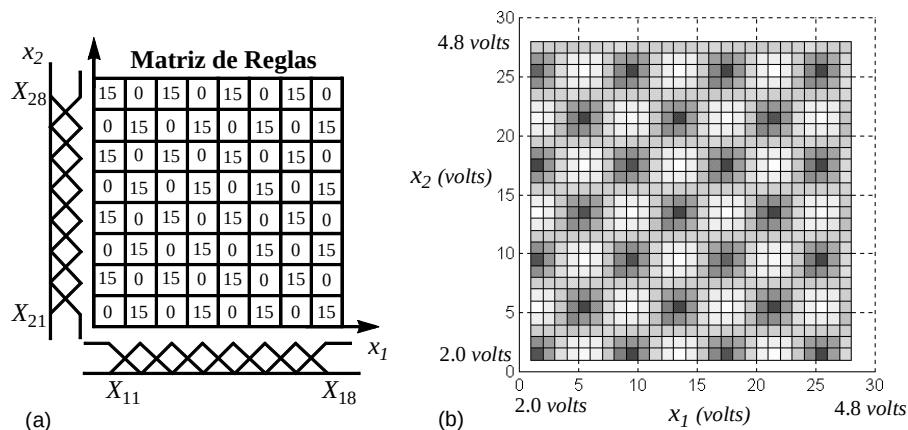


Fig.6.16 Resultados de MFCON. Ejemplo 1: Matriz de Reglas (a); Vista aérea de la superficie de control resultante (b).



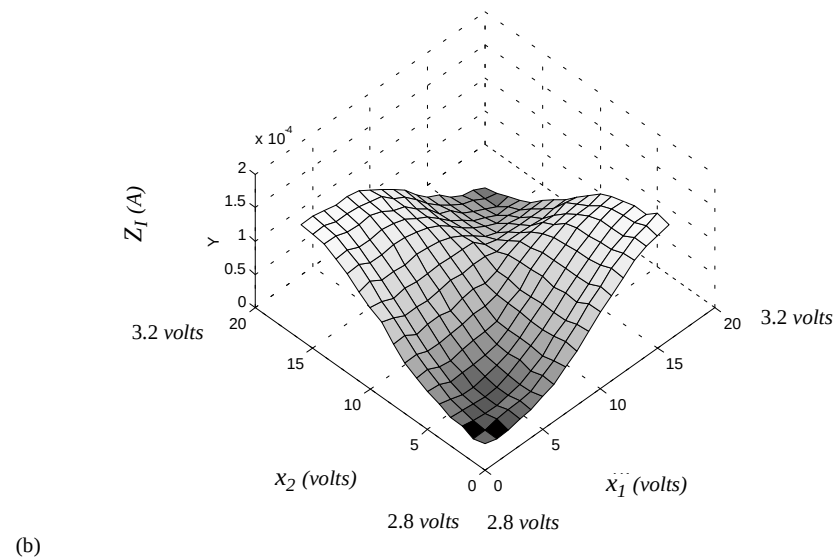
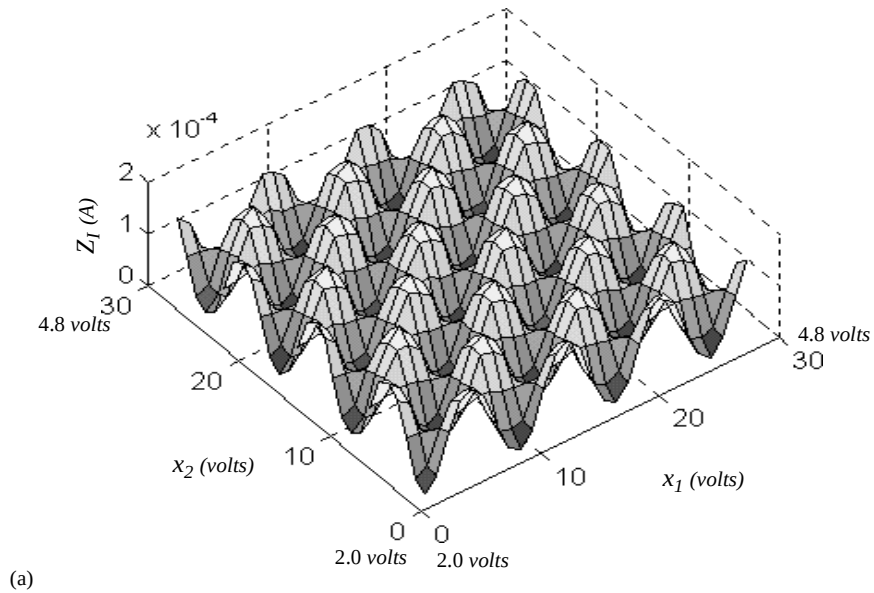


**Fig.6.17 Resultados de MFCON. Ejemplo 1: Superficie de control (a); Detalle de la superficie de control sobre un intervalo de interpolación (b).**

- **Ejemplo 2.** Plano inclinado en ambas variables de entrada.

La Fig.6.18(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ve claramente como los valores *singleton* van creciendo linealmente a lo largo del universo de discurso de las dos variables de entrada  $x_1$  y  $x_2$ , desde el valor mínimo (0) hasta el valor (14). La superficie de control resultante es el plano rampante que muestra la Fig.6.18. Este ejemplo ilustra las habilidades de MFCON en la interpolación de superficies lineales.

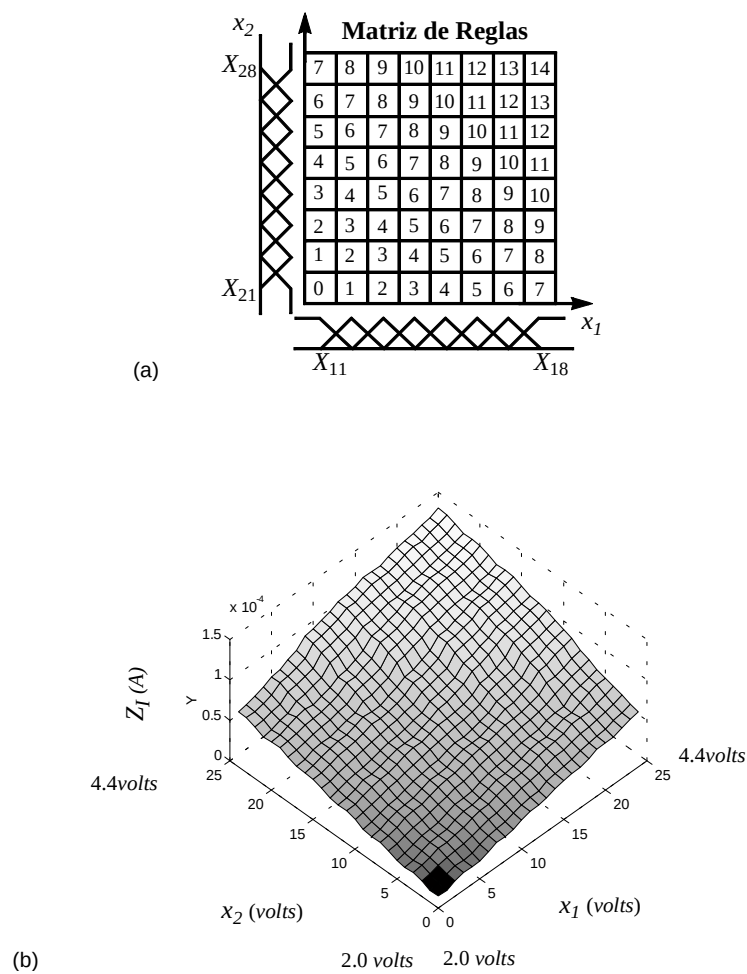


Fig.6.18 Resultados de MFCON. Ejemplo 2: Matriz de Reglas (a); superficie de control (b).

- **Ejemplo 3.** Plano inclinado en una de las variables de entrada.

La Fig.6.19 muestra la matriz de reglas con la que es configurada MFCON en este ejemplo. Con ella se genera una superficie lineal con forma de plano inclinado en la variable  $x_1$  que es empleada en este ejemplo para obtener el detalle del comportamiento como interpolador lineal que se observa en la Fig.6.20(a) y la Fig.6.20(b).

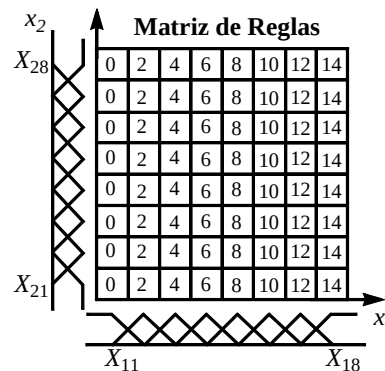
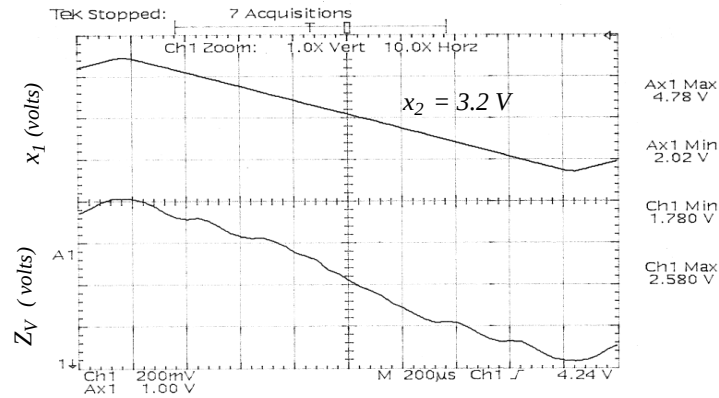


Fig.6.19 Resultados de MFCON. Ejemplo 3: Matriz de Reglas

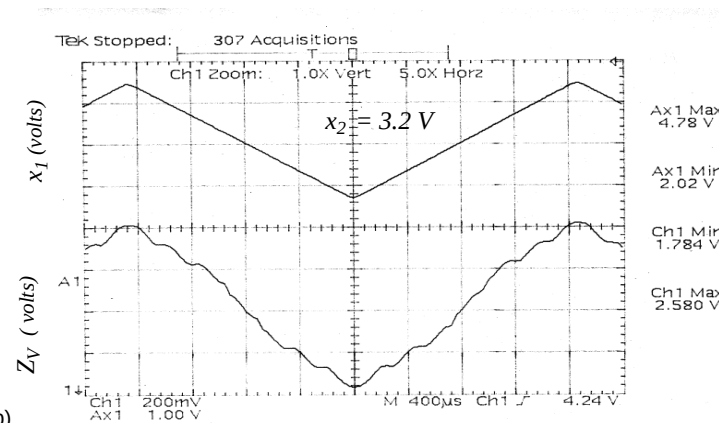
Ambas son imágenes obtenidas de la pantalla del osciloscopio digital con el que se ha sensado la salida en tensión  $Z_V$  de MFCON y muestran una sección transversal de la superficie de control. Para su obtención se ha fijado el valor de la entrada  $x_2$  a 3.2 volts y se ha efectuado un barrido sobre el rango de entrada de  $x_1$  aplicando una señal diente de sierra, Fig.6.20(a) y una señal triangular, Fig.6.20(b), ambas de frecuencia suficientemente baja para considerar que la respuesta que muestra el osciloscopio coincide prácticamente con la respuesta dc del sistema.

En ambas figuras se aprecia cierto rizado sobre el comportamiento lineal en la respuesta del controlador. Este efecto es consecuencia, por una parte, del multiplexado del *Núcleo Analógico* sobre los diferentes intervalos de interpolación; y por otra de la no linealidad propia de la respuesta del *Núcleo Analógico* y del algoritmo difuso que realiza. La Fig.6.21 muestra la superficie teórica que proporciona el controlador para aproximar un plano en un intervalo de interpolación obtenida por simulación del modelo matemático de los circuitos que realizan el *Núcleo Analógico*. En ella se muestra la no linealidad de dicha respuesta y la curvatura que da lugar al rizado de la Fig.6.20(a) y la Fig.6.20(b).

Por último, la pequeña asimetría respecto a la dirección de barrido, descendente o ascendente, que se aprecia en la Fig.6.20(b) es debida a la presencia de histéresis en las entradas de MFCON.



(a)



(b)

**Fig.6.20 Resultados de MFCON. Ejemplo 3: Sección transversal de la superficie obtenida mediante un barrido de  $x_1$  con una señal rampa (a); un barrido de  $x_1$  con una señal triangular (b). En ambos casos  $x_2$  permanece fija al valor de 3.2 volts.**

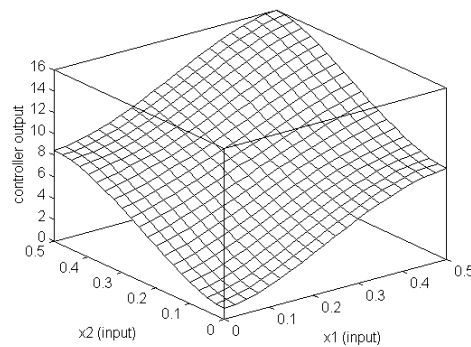


**Fig.6.21 Resultados de MFCON. Ejemplo 3: Superficie teórica generada por el Núcleo Analógico para aproximar un plano.**

- **Ejemplo 4.** Planos horizontales a diferente nivel.

La Fig.6.22(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ven claramente los cuatro planos horizontales definidos por los grupos de valores *singleton* 0, 5, 10 y 14. La superficie de control resultante se muestra en la Fig.6.22(b). Este es otro ejemplo que ilustra las habilidades de MFCON en la interpolación de superficies.

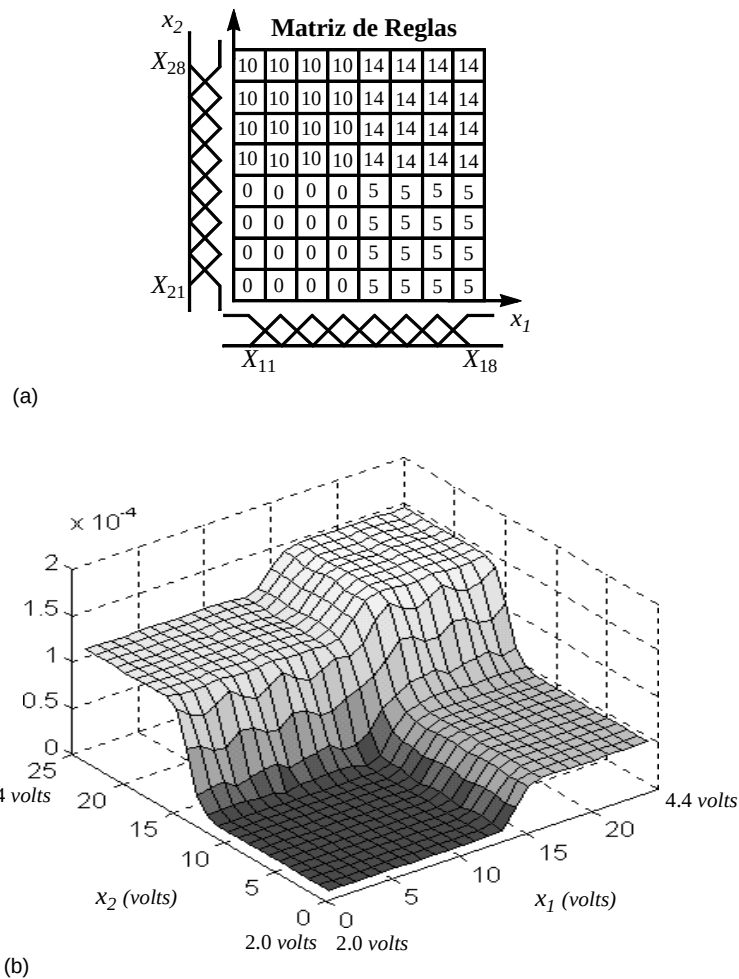


Fig.6.22 Resultados de MFCON. Ejemplo 4: Matriz de Reglas (a); superficie de control (b).

- **Ejemplo 5.** Pozos de planos horizontales a diferente nivel.

La Fig.6.23(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ven claramente los cuatro pozos planos horizontales definidos por los grupos de valores *singleton* 2, 5, 1 y 4. La superficie de control resultante se muestra en la Fig.6.23(b). Este es otro ejemplo que ilustra las habilidades de MFCON en la interpolación de superficies.

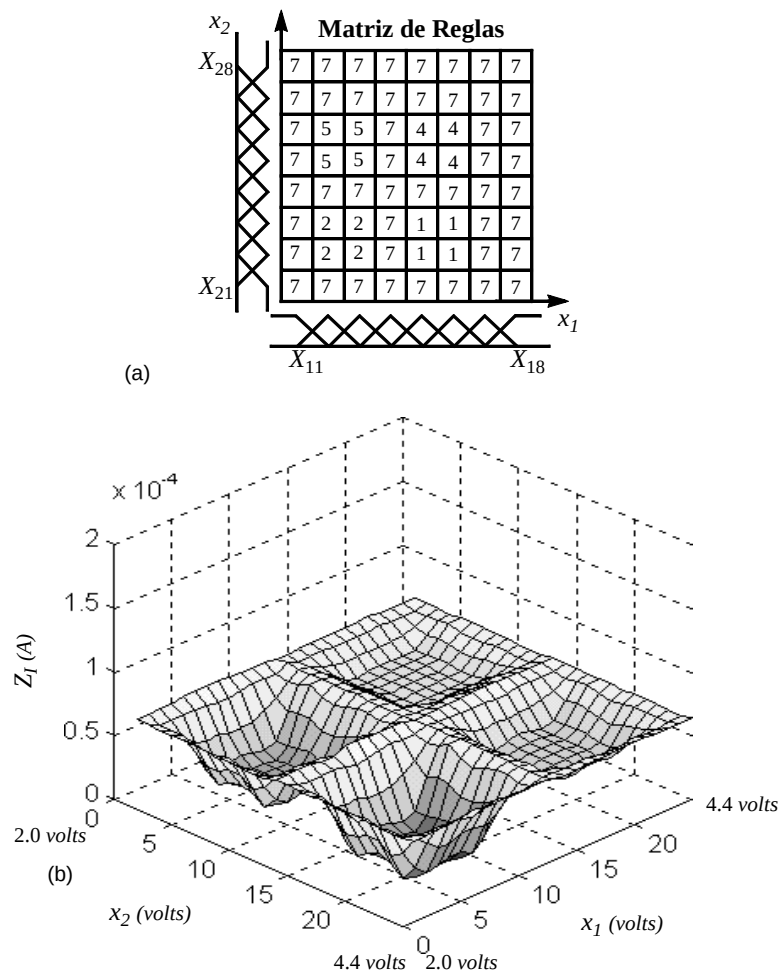


Fig.6.23 Resultados de MFCON. Ejemplo 5: Matriz de Reglas (a); superficie de control (b).

• **Ejemplo 6.** Superficie de control no lineal I.

La Fig.6.24(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ven los puntos que sirven de base para la interpolación de la superficie completa los cuales forman círculos concéntricos en torno al punto de valor máximo (15). La superficie de control resultante se muestra en la Fig.6.24(b). Este ejemplo ilustra las habilidades de MFCON en la interpolación de superficies no lineales.

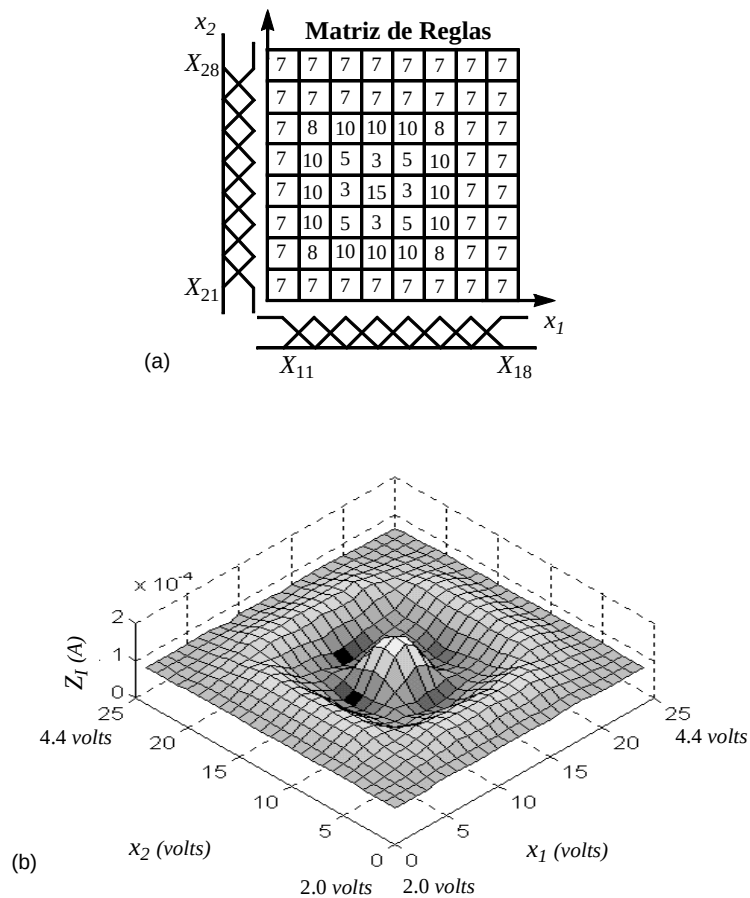


Fig.6.24 Resultados de MFCON. Ejemplo 6: Matriz de Reglas (a); superficie de control (b).

- **Ejemplo 7.** Superficie de control no lineal II.

La Fig.6.25(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. En ella se ven los puntos que sirven de base para la interpolación de la superficie completa los cuales forman círculos concéntricos en torno al punto de valor máximo (15). La superficie de control resultante se muestra en la Fig.6.25(b). Este ejemplo ilustra las habilidades de MFCON en la interpolación de superficies no lineales.

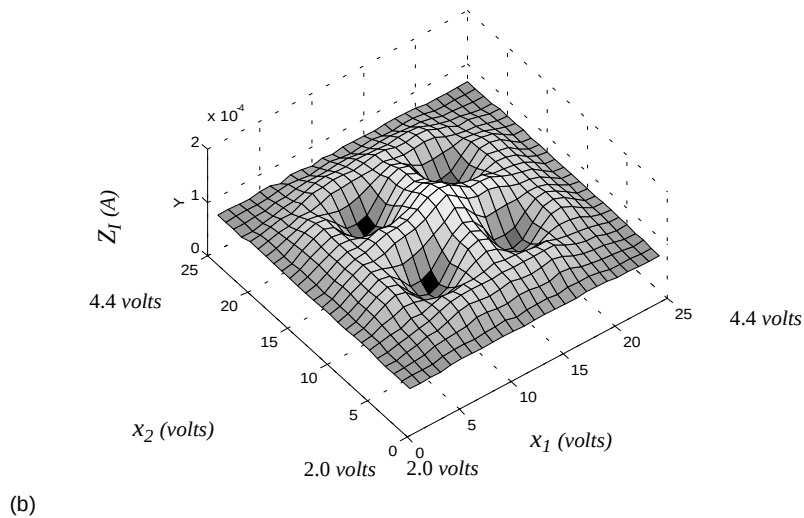
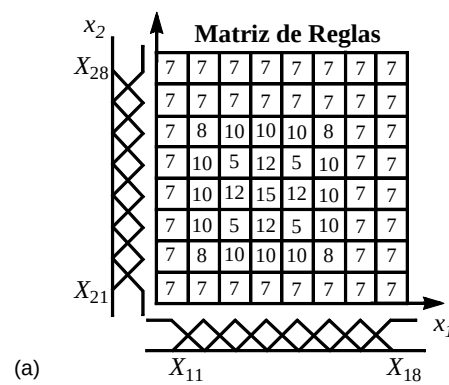


Fig.6.25 Resultados de MFCON. Ejemplo 7: Matriz de Reglas (a); superficie de control (b).



- **Ejemplo 8.** Superficie de control no lineal III.

La Fig.6.26(a) muestra la matriz de reglas con la que es configurado MFCON en este ejemplo. Con ella se genera la superficie cuya sección se muestra en la Fig.6.26(b) como ejemplo del comportamiento del controlador como interpolador no lineal.

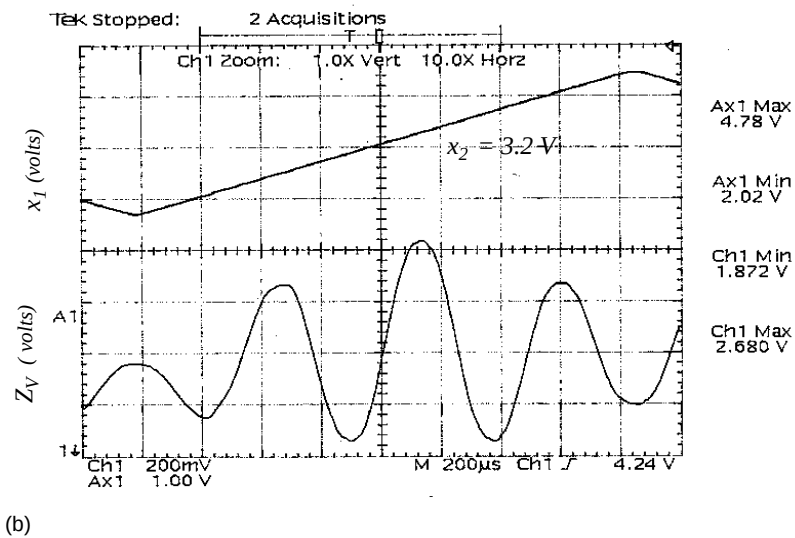
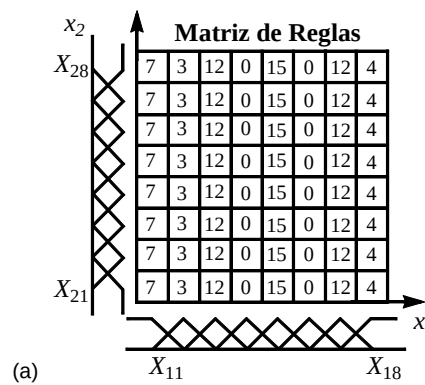


Fig.6.26 Resultados de MFCON. Ejemplo 8: Matriz de reglas (a); Sección transversal de la superficie obtenida mediante un barrido de  $x_1$  con una señal rampa, mientras  $x_2$  permanece fija al valor de 3.2 volts (b).

#### 4.2. Comportamiento dinámico del MFCÓN.

En este apartado se evalúa el comportamiento dinámico de MFCÓN. Las pruebas realizadas van encaminadas a determinar su tiempo de respuesta o retardo entrada salida en el peor de los casos.

Para ello se ha configurado MFCÓN con la matriz de reglas de la Fig.6.16(a) se ha fijado el valor de la entrada  $x_2$  a 2.8 volts y se ha aplicado una señal periódica de onda cuadrada 100kHz cuya amplitud cubre el rango de entrada de  $x_1$ . Con esta excitación se obliga al controlador a saltar entre intervalos de interpolación extremos en cada uno de los flancos de la señal de entrada, forzando una reconfiguración dinámica del *Núcleo Analógico* y un cambio a la salida entre los valores extremos de su rango. La Fig.6.27 muestra la respuesta del controlador a esta excitación.

Las medidas del tiempo de retardo han sido efectuadas con el osciloscopio digital y para ambas transiciones. Para la transición ascendente se han tomado como referencia de medida los puntos de 90% del valor final, mientras que para la transición descendente se ha tomado como referencia de medida los puntos de 10% del valor inicial.

La Fig.6.28 muestra el resultado de esta medida con los datos según aparecen en la pantalla del osciloscopio. Para la transición de subida el retardo medido es de 450ns, según muestra el resultado de la medida efectuada que aparece en el margen izquierdo de la Fig.6.28(a); mientras que para la transición descendente el retardo medido es 480ns, según muestra el resultado de la medida efectuada que aparece en el margen izquierdo de la Fig.6.28(b). Ambos datos resultan ser un promedio sobre varios ciclos de la señal de entrada.

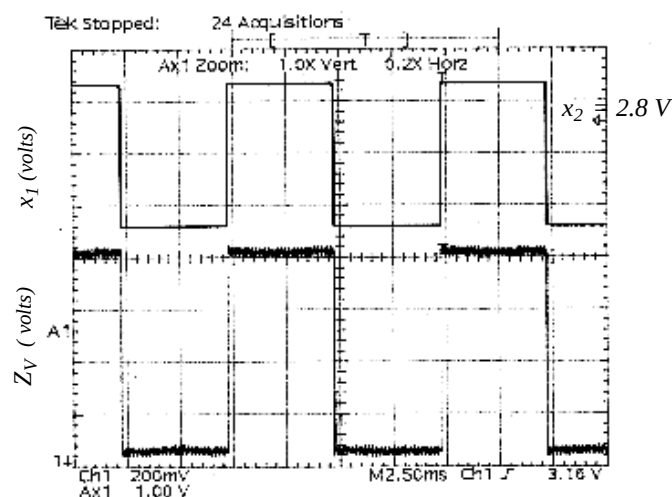
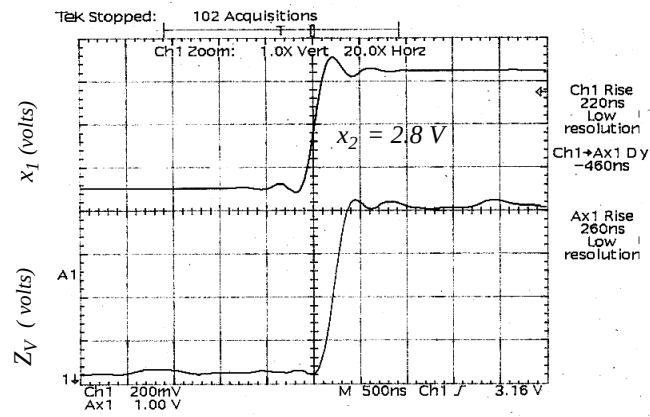
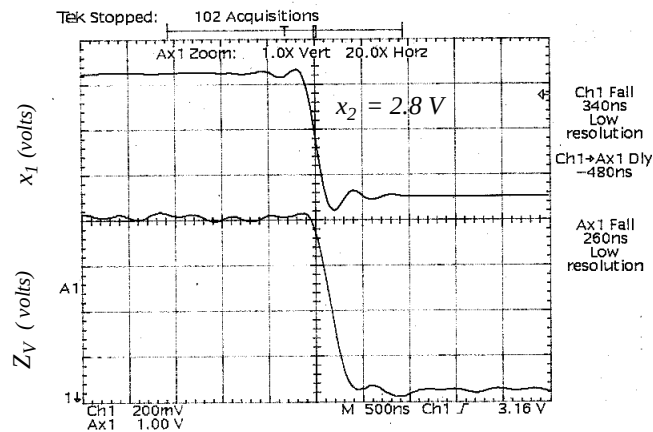


Fig.6.27 Respuesta dinámica de MFCÓN: Respuesta a una excitación periódica de onda cuadrada.



(a)



(b)

Fig.6.28 Respuesta dinámica de MFCON: Flanco de subida (a); flanco de bajada (b).

## Capítulo 7

### Aplicación a un caso práctico: Arranque suave de motores de corriente continua.

#### 1. INTRODUCCIÓN

En este capítulo se presenta un ejemplo de aplicación del controlador integrado MFCON, que introduce el control difuso en el problema del control del arranque de motores de corriente continua. En ella MFCON es utilizado para establecer el ciclo de trabajo (*duty cycle*) en régimen estacionario  $D_{TS}$ , de un convertidor conmutado de corriente continua a corriente continua (*DC-DC switching converter* o *chopper*) regulado desde un sistema PWM (modulación por anchura de pulso) que controla la transferencia de potencia, desde una fuente de alimentación continua,  $V_{CC}$ , a un motor de corriente continua  $M_{CC}$ , según muestra el esquema de la Fig.7.1. Como dispositivo de conmutación se utiliza un interruptor MOSFET. Este sistema permite controlar el valor medio de tensión,  $V_m$ , que se aplica en los terminales de armadura del motor, en función del ciclo de trabajo, y, en definitiva, determinar la velocidad de giro en estado estacionario  $\omega_s$ . El objetivo de control es llevar al motor desde el reposo hasta la velocidad en régimen estacionario fijada, en el menor tiempo posible y efectuando un arranque seguro, a par máximo y constante durante la fase de arranque. Para ello, el valor del ciclo de trabajo en cada instante  $D_T$  es actualizado para cada ciclo de la señal de referencia del sistema PWM, según la estrategia de control programada en MFCON. Como entradas del controlador se emplean las señales provenientes del sensado de la velocidad de giro del motor,  $\omega$ , y de la corriente que circula por su armadura  $I_m$ , señales éstas habitualmente utilizadas en los sistemas de control de velocidad y posición para este tipo de motores.

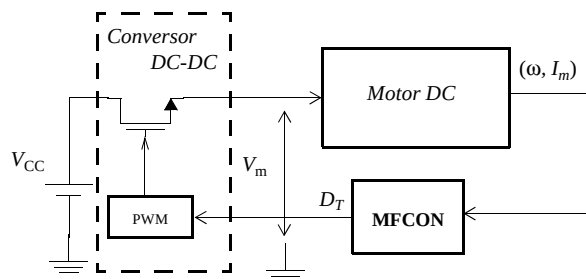


Fig.7.1 Esquema básico del ejemplo de aplicación de arranque suave de motores de corriente continua.

Con este ejemplo se persiguen dos objetivos:

En primer lugar, se trata de completar el conjunto de pruebas que sirven para ilustrar las prestaciones del prototipo que se ha desarrollado para verificar la viabilidad de la propuesta de controladores difusos con núcleo analógico multiplexado, y que se iniciaron en el capítulo anterior, con un ejemplo, original en la literatura y de complejidad adecuada, que permita mostrar sus cualidades cuando funciona integrado en un bucle de control que necesite una alta velocidad de respuesta.

En segundo lugar, y en igualdad de importancia que el anterior objetivo, se pretende adquirir una primera experiencia en el campo de la *Electrónica de Potencia* y el *Control de Movimiento*, disciplinas éstas para las cuales, según diversos autores [Bose94] [Hara94] [Boni95], la incorporación de las nuevas tecnologías de la inteligencia artificial, entre las que destaca la lógica difusa, está ejerciendo un fuerte impacto. Esta experiencia va dirigida a iniciar una línea de trabajo en la búsqueda de aplicaciones de los controladores difusos analógicos y de señal mixta en general, y de los controladores aquí propuestos en particular, en las que sus principales cualidades: interfaz natural con sensores y actuadores, compacidad, buena eficiencia en la razón velocidad/consumo de potencia para resoluciones medias o bajas, muy pequeño retardo entrada-salida, flexibilidad de configuración y capacidad de aprendizaje, sean explotadas.

En este sentido el ejemplo de aplicación aquí presentado puede ser considerado como un caso particular del problema de control más general que constituye el control de velocidad y limitación de corriente en servomecanismos o de sistemas con motores de velocidad variable. En ambos casos la exigencia en estabilidad y rapidez de respuesta a un cambio en los valores de referencia de posición o velocidad, y la reacción que esta acción provoca en el sistema, es la misma, de manera que el arranque constituye un caso extremo, y la experiencia adquirida puede ser posteriormente aplicada al caso más general.

En el siguiente apartado se resumen diferentes aspectos de la *Electrónica de Potencia* y el *Control de Movimiento* que justifican e ilustran el interés en el control difuso empleando sistemas analógicos. A continuación se define el problema del control del arranque de motores de corriente continua que se desarrolla como ejemplo, se describe la estrategia de control y el sistema diseñado para su realización y finalmente se exponen los resultados obtenidos en diversos ejemplos de control de arranque.

## 2. ELECTRÓNICA DE POTENCIA Y CONTROL DE MOVIMIENTO

La *Electrónica de Potencia* se ocupa básicamente de la transformación de la energía eléctrica y tiene como objetivo principal maximizar la eficiencia de dicho proceso de transformación, y en definitiva procurar el uso eficaz de la electricidad [Rash93] [Moha95]. La Fig.7.2 muestra el esquema básico de un sistema de electrónica de potencia, en el que además de la fuente de potencia y la carga, elementos entre los que se produce la transferencia de potencia, aparecen, como bloques principales, un bloque procesador de potencia y un bloque de control electrónico. El primero se construye a partir de bloques básicos denominados convertidores de potencia [Wyk94], contruidos con dispositivos electrónicos de potencia que funcionan en conmutación [Bali94]. Por su parte el bloque de control, ejecuta una estrategia de control a partir de la información de sensores y las señales de referencia y genera las señales de control que determinan el funcionamiento del procesador de potencia de la manera más eficiente posible.

Tras una primera etapa dominada por el empleo de los trisores y los correspondientes circuitos de control *ad-hoc*, y una segunda etapa impulsada por el empleo de los nuevos dispositivos semiconductores de alta velocidad de conmutación (GTO, MOSFET, IGBT, etc.) [Bali94] [Shep95], cuyo uso en procesadores de potencia tales como inversores, convertidores dc-dc, etc [Wyk94], se vio potenciado y favorecido con la incorporación de microcontroladores que permitieron introducir la moderna teoría de control en esta disciplina y mejorar así su objetivo de eficiencia [Bose87]; en la actualidad, la electrónica de potencia demanda una mayor frecuencia de conmutación que permita realizar sistemas de menor tamaño, además de aprovechar la velocidad y capacidad de computación que ofrecen los nuevos procesadores, para ejecutar los cada vez más sofisticados algoritmos de control, que inciden en el comportamiento no lineal de los sistemas a controlar. En este sentido los convertidores controlados con técnicas de modulación por anchura de pulsos (PWM) y los convertidores resonantes son los sistemas más empleados a altas frecuencias [Rash96].

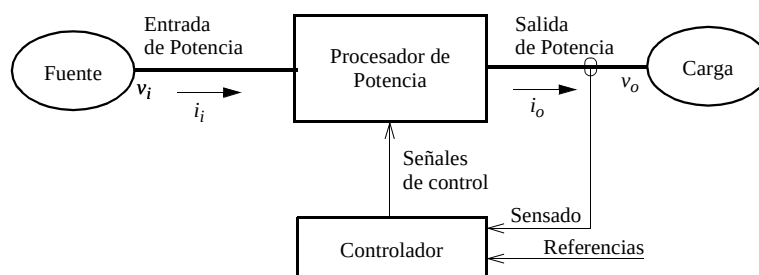


Fig.7.2 Diagrama de bloques de un sistema de electrónica de potencia.

Según muestra la Fig.7.3 [Hara94], el campo de aplicación de la electrónica de potencia se extiende desde las aplicaciones de muy alta potencia y baja frecuencia de operación, como son las líneas de alta tensión de corriente continua; hasta aplicaciones de baja potencia y elevada frecuencia, entre las que se encuentran los reguladores conmutados (*switching regulator*). Ocupando una zona intermedia aparece una amplia gama aplicaciones como las de control de máquinas eléctricas y por tanto con los sistemas de control de movimiento, la realización de fuentes de alimentación ininterrumpida (UPS), los coches eléctricos y los sistemas de calefacción por inducción, electrónica de consumo, entre otras. En todos estos ámbitos la lógica difusa es aplicada a muy diversas tareas, entre las que destacan las de control, diagnóstico, modelado y estimación de parámetros, etc. En muchas de ellas los controladores difusos son realizados utilizando sistemas basados microcontroladores de propósito general o DSP (*Digital Signal Processor*) [Mark94] [El-M96] [Hill96] [Guil96] [Peri94] [Lin97]. Como ejemplo de aplicaciones en las que intervienen controladores difusos analógicos se pueden citar [Pagn93] y [Fran98]. En ambos trabajos se aborda un problema de regulación de tensión empleando convertidores DC-DC conmutados. Dado que los sistemas objeto de esta tesis se encuentran en esta categoría, se consideraran brevemente algunos aspectos de estos ejemplos de aplicación.

En el primero, el uso de controladores difusos se justifica porque los sistemas de alta potencia, para los que la aplicación está orientada, requieren un control robusto, con una ley de control fuertemente dependiente de los parámetros del sistema, que se degrada ante variaciones éstos. Por otra parte, que el controlador sea analógico se justifica por la simplicidad de la estructura computacional del problema de control cuando se formula en términos de Lógica Difusa, lo que da lugar a una importante reducción del coste global, y por la velocidad de funcionamiento del regulador conmutado empleado (100kHz), lo que exige pequeño retardo en el lazo de control.

En el segundo trabajo, se plantea el control de un convertidor DC-DC conmutado, que es empleado en la regulación de las fuentes de alimentación de un moderno sistema digital basado

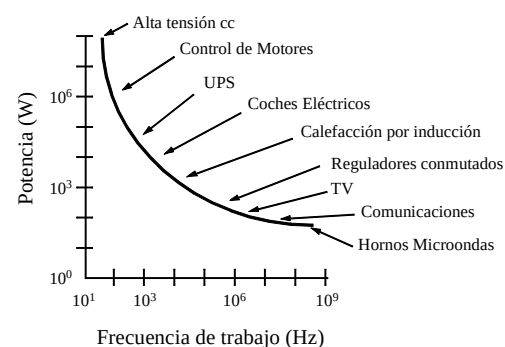


**Fig.7.3 Aplicaciones en electrónica de potencia representados en una gráfica potencia consumida frente a la frecuencia de trabajo.**

en microprocesador, en el que la alta frecuencia de trabajo da lugar a grandes picos de corriente, y en el que además es posible encontrar grandes variaciones en los valores medios de la corriente requerida en los estados activos y de reposo. En estos sistemas se hace necesaria una buena regulación del sistema de alimentación de manera que se garanticen pequeños errores en estado estacionario, se proporcione un rápido tiempo de establecimiento sin sobredisparos cuando se produzcan cambios en la impedancia de carga, además de una limitación en la corriente suministrada. En este ejemplo el controlador realiza una ley de control no lineal, por lo que se emplea un controlador difuso, visto aquí desde su perspectiva de sistema interpolador de superficies de control no lineales. La opción de un controlador analógico se justifica aquí por los elevados requerimientos de velocidad, la señal de conmutación del sistema PWM que controla el convertidor puede trabajar hasta 500kHz, además de la reducción de coste y tiempo de respuesta que supone la eliminación de conversores A/D y D/A en el lazo de control. Estos dos ejemplos van en la línea del ejemplo de aplicación aquí presentado.

Por su parte, la tecnología del *Control de Movimiento* se entiende en la actualidad como una materia multidisciplinar en la que se conjugan elementos de diversas disciplinas como son la electrónica de potencia, la ingeniería mecánica, la ingeniería de control y la ciencia de la computación. Su objetivo básico es el control eficiente de los elementos que producen el movimiento en sistemas tales como servosistemas y robots, de forma que puede decirse que esta disciplina abarca todas aquellas tecnologías relacionadas con el movimiento de objetos, desde los microsistemas realizados sobre silicio, hasta sistemas de gran tamaño [Hara94].

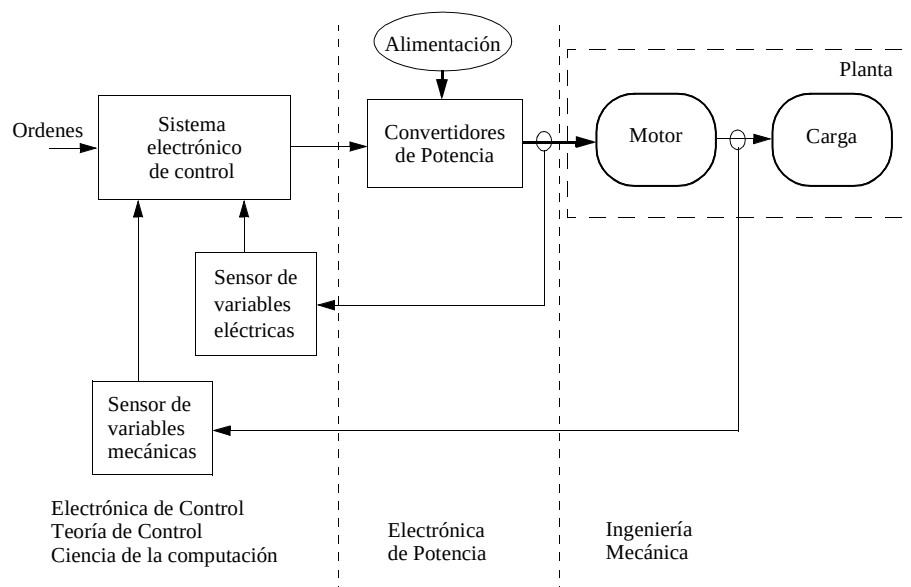


Fig.7.4 Esquema básico de un sistema de control de movimiento



Un sistema de control de movimiento, como por ejemplo un brazo manipulador o un sistema de posicionamiento, consiste en general en uno a varios actuadores eléctricos operando bajo el control de un ordenador central que coordina la acción global. El problema del control de un sistema de esa complejidad se divide habitualmente en diferentes niveles. A nivel alto se coordinan los diferentes actuadores que han de contribuir al movimiento de forma que se siga la trayectoria planeada. A un nivel más bajo se controla la acción que ha de ejecutar cada actuador eléctrico que generalmente se concreta en un control de la posición y la velocidad. La Fig.7.4 muestra un esquema básico de uno de estos sistemas. Las plantas a controlar son habitualmente máquinas eléctricas, (motores AC o DC) alimentadas a través de sistemas de potencia tales como inversores, convertidores DC-DC, etc [Wyk94], que son controlados a su vez por sistemas electrónicos que ejecutan la estrategia de control deseada.

Aunque tradicionalmente el control de los actuadores eléctricos fue materia de la electrónica analógica, esta disciplina se vio impulsada con la aparición de los primeros microprocesadores [Bose87] [Le-H94], los cuales en su evolución han ido también señalando el camino a su desarrollo posterior, en una doble vertiente: Por una parte la potencia de cálculo que estos ofrecen ha permitido aplicar en tiempo real, sofisticadas estrategias de control resultado de los métodos de las modernas teorías de control; y por otro el rápido incremento en su velocidad de operación ha permitido aprovechar los importantes avances que en el campo de la electrónica de potencia han supuesto las técnicas de conmutación (PWM) [Holt94], y los nuevos dispositivos de potencia como elementos de conmutación, (GTO, MOSFET, IGBT, etc.) [Bali94] [Shep95]. Por ello, en la actualidad, el caballo de batalla del control de movimiento, al nivel bajo mencionado con anterioridad, lo constituyen los microcontroladores, los cuales son dispositivos orientados a tareas de control que incorporan en un circuito integrado la mayoría de los elementos periféricos que necesita un sistema basado en microprocesador para efectuar tareas de control, como son: sistema de adquisición de datos provenientes de sensores, interfaz de comunicaciones con el ordenador central, unidad de procesamiento del algoritmo de control, interfaz con los circuitos de potencia para controlar la acción del actuador eléctrico, así como elementos de test y diagnóstico. Por su parte, el empleo de DSP (Digital Signal Processors), dispositivos inicialmente desarrollados para la computación digital en tiempo real en sistemas de procesamiento de señal, en sistemas de control de movimiento, ha permitido explotar su alta capacidad de cómputo para elevar la frecuencia de muestreo al ejecutar complejos algoritmos en tareas de control.

Según se señala en [Hara94], una de las cuestiones de actualidad en el campo de control de movimiento es la incorporación de "inteligencia" a los sistemas de control. En general, ésta se plantea en el sentido de realizar sistemas de control de manera que sean capaces de ajustarse a sí mismos para seguir los posibles cambios en su entorno o en los requerimientos en la tarea a desarrollar, lo que requiere la incorporación de técnicas que permitan el tratamiento de la

incertidumbre y el aprendizaje. Por otra parte, la mayoría de los sistemas de control están basados en la teoría de control lineal, mientras que los sistemas sobre los que se ha de actuar poseen no linealidades, cuyo efecto se hace más patente a medida que se desean mayores prestaciones. En este sentido, la incorporación de las técnicas de las redes neuronales y la lógica difusa, habituadas a tratar con estos problemas, resulta ser provechosa, como demuestra la gran cantidad de aplicaciones que se están desarrollando en estos últimos años y en las que los controladores difusos, realizados tanto con elementos *software* como *hardware*, son empleados en las más diversas tareas [Liaw91][Suyi93][Mark94]. Sin embargo, cuestiones a dilucidar como la dificultad de precisar en qué área o a qué nivel realmente es necesaria inteligencia, e incluso la propia definición del concepto de inteligencia en este contexto, hacen que la aplicación de estos métodos no resulte directa y exija aún esfuerzos de investigación.

La incorporación de estas nuevas técnicas precisa una creciente necesidad de potencia de cálculo de los sistemas de control a todos los niveles. Por ello, y dado que una estructura jerárquica de control resulta ser muy eficiente para la consecución del objetivo de control, siempre que se realice una distribución adecuada de tareas, a pesar de que la evolución de los microcontroladores sea capaz, en gran medida, de seguir esa demanda de potencia de cálculo, aun queda hueco en el sistema de control para la contribución de circuitos integrados de aplicación específica (ASIC), tanto digitales como analógicos, bien asumiendo la tarea del control de forma general, o bien participando en aspectos concretos que liberan al procesador de algunas tareas de cálculo demasiado específicas, de forma que ayuden a superar algunas de las limitaciones del control de sistemas de movimiento con elementos digitales, y que en el caso de los ASIC analógicos inciden en superar problemas como los inherentes de las propiedades de los sistemas discretos resultado del proceso de muestreo, cuantización y truncamiento, así como las limitaciones en el ancho de banda que debido al tiempo de computación pueden afectar a la estabilidad del sistema, esto es, principalmente cuestiones de interfaz y de velocidad de operación. Como ejemplo ilustrativo de este último tipo de colaboración puede citarse el que aparece en [Cerru92]. En él, como muestra la Fig.7.5, el controlador difuso de American

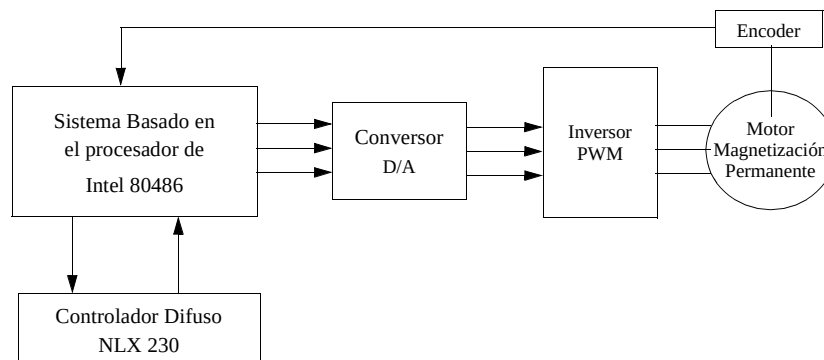


**Fig.7.5 Diagrama de bloques del sistema de control adaptativo de [Cerru92] en el que un controlador difuso funciona en colaboración con un sistema basado en microprocesador.**

Neuralogix NLX230 trabaja en colaboración con un sistema basado en el microprocesador 80486 de Intel en un sistema de control de posición y velocidad que emplea un algoritmo de control adaptativo que ejecuta el microprocesador. El controlador difuso es empleado para actualizar los parámetros de control una vez que han sido compensados para tener en cuenta posibles variaciones tanto en el par de carga como en el momento de inercia.

Como ejemplos adicionales del uso de controladores difusos aplicados al control de motores pueden citarse [Sous94] y [Sous93]. El primero es un ejemplo de control de velocidad y limitación de corriente de un motor DC. Un esquema del sistema de control empleado se muestra en la Fig.7.6. En este ejemplo el controlador es empleado para establecer cual ha de ser la corriente de referencia de un convertidor controlado por corriente, tomando como entrada el error en la velocidad y su derivada. El resultado es un control más rápido y robusto frente a perturbaciones en la carga y variación de parámetros que el que se consigue con un control tradicional PID. En el segundo ejemplo el controlador difuso es usado en la optimización del control de un motor de inducción, y es empleado para la estimación flujo magnético del punto de mínimo consumo de potencia. Con estos ejemplo se quiere destacar que el empleo de controladores difusos en el control de motores es un importante campo de trabajo, en el que, en la medida que sea necesaria una alta velocidad de operación, los controladores analógicos pueden aportar su buenas prestaciones.

Por otra parte el desarrollo de las técnicas de diseño de circuitos integrados de señal mixta favorecen la integración, a modo de bloques constructivos, de diferentes celdas analógicas o digitales, junto a sistemas basados en microcontrolador o DSP, lo que permite el desarrollo de sistemas de control de movimiento integrados optimizados. Esta alta integración da lugar a un reducción en cuanto a número de chips lo que conlleva una reducción en cuanto a costes de fabricación y supone una mejora en la fiabilidad del sistema. En este sentido la compacidad y buena eficiencia velocidad/consumo de potencia de las celdas analógicas resulta especialmente pertinente.

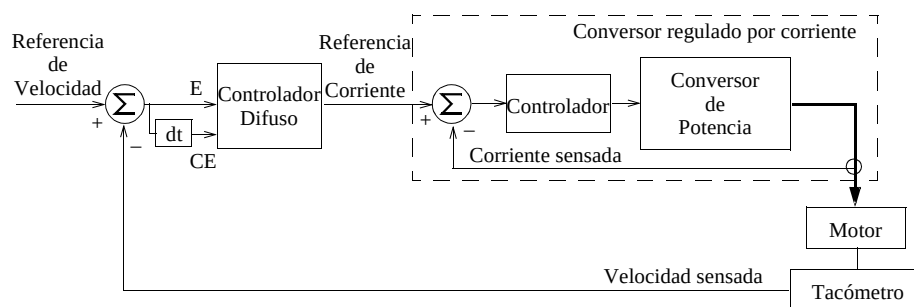


**Fig.7.6 Esquema de un sistema de control de velocidad de un motor DC que emplea un controlador difuso.**

### 3. CONTROL DE VELOCIDAD Y ARRANQUE SUAVE DE MOTORES DE CORRIENTE CONTINUA

#### 3.1. Problema del arranque de los motores de corriente continua.

Cuando un motor de corriente continua se conecta directamente a una fuente de alimentación para que comience a funcionar, durante el transitorio de arranque, esto es, mientras el motor se acelera hasta alcanzar su velocidad en régimen estacionario, la corriente que circula por el bobinado de su armadura excede el valor final que se alcanza en régimen estacionario. Esta corriente posee un acusado pico justo en el momento de arranque que puede llegar a presentar una magnitud muy superior a dicha corriente en régimen estacionario. La Fig.7.7 ilustra este comportamiento con los resultados obtenidos para el motor que es empleado en la realización ejemplo de aplicación que se describe en este capítulo. En ella se representan la evolución las variables resultado del sensado de la velocidad del motor y de su corriente durante el transitorio de arranque.

Este pico inicial de corriente produce sufrimiento en los materiales que forman el bobinado del motor y demás elementos del sistema, que puede evolucionar a un daño irreversible si la situación se repite frecuentemente a lo largo del tiempo, causando una reducción en la vida útil del motor en particular y del sistema en general. Por otra parte, estos picos de corriente provocan caídas en la salida de la fuente de tensión continua que alimenta al sistema, que afectan negativamente, tanto a la propia fuente, como al resto del elementos que estén siendo alimentados por ella. Estos efectos son especialmente perjudiciales en sistemas portátiles alimentados con baterías, en los que además provocan un exceso de consumo de potencia innecesario que reduce su carga y por tanto, el tiempo de autonomía del sistema.

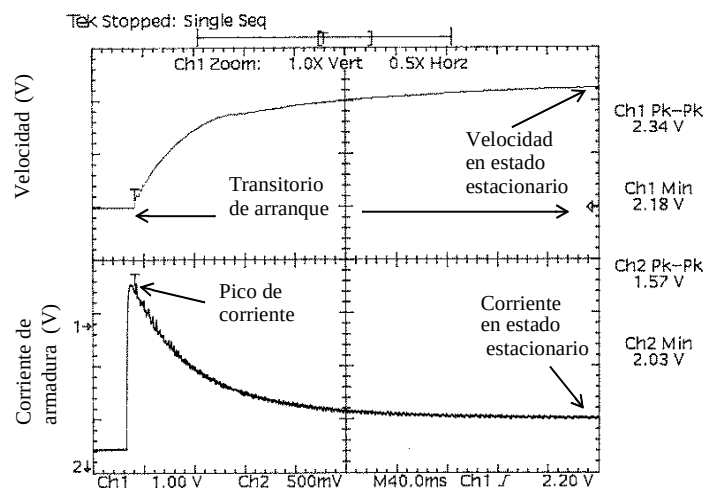


Fig.7.7 Transitorio de arranque directo de un motor de corriente continua

Además, todos estos efectos se ven acusados en motores que han de trabajar en sistemas con un régimen de funcionamiento que suponga un continuo encendido y apagado. Finalmente, como en estos sistemas el par desarrollado es proporcional a la corriente que circula por el motor, el pico inicial supone a la vez un pico también en el par de arranque, que en algunas circunstancias puede resultar perjudicial para la carga del motor.

### 3.2. Ecuaciones básicas de un motor de corriente continua.

Para entender el comportamiento que se ha descrito anteriormente, a continuación se repasan las principales expresiones del modelo más habitual empleado para caracterizar a este tipo de motores visto desde sus terminales [Shep95][Moha95][Rash93].

Un motor de corriente continua funciona bajo el principio básico de la interacción de un campo magnético con una espira de corriente. El campo magnético puede ser generado por un imán permanente o por una corriente eléctrica que circula por un bobinado, denominado por ello bobinado de campo. La espira de corriente se organiza entorno a un elemento móvil denominado rotor, que puede girar dentro del campo magnético, a este bobinado se le denomina bobinado de armadura. En la Fig.7.8 se representan esquemáticamente los dos bobinados de campo antes mencionados junto con los elementos y variables eléctricas que los modelan, y su circuito equivalente.

En el caso de la existencia de un bobinado de campo, la inyección de corriente continua en él, establece una corriente de excitación  $i_f$  que conforma un campo magnético en el entrehierro de flujo  $\phi_f$ .

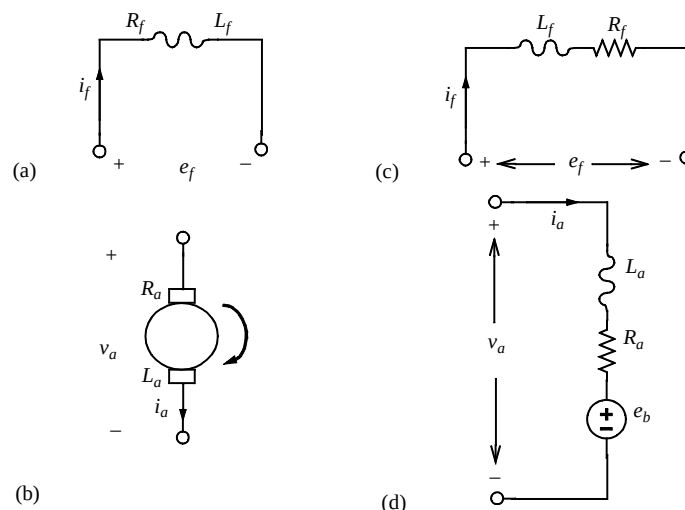


**Fig.7.8 Representación de los bobinados de campo (a); y armadura (b) en un motor de corriente continua. Circuitos equivalentes de los bobinados de campo (c); y armadura (d).**

En términos de las variables instantáneas se tiene

$$e_f = i_f \cdot R_f + L_f \cdot \frac{di_f}{dt} \quad (7.1)$$

$$\Phi_f = f(i_f) \quad (7.2)$$

Si el motor opera en la parte lineal de la curva de magnetización que representa la expresión (7.2) puede establecerse una relación lineal entre el flujo magnético  $\Phi_f$  y la corriente  $I_f$  del motor en el estado estacionario

$$\Phi_f = K_f \cdot I_f \quad (7.3)$$

En estado estacionario no hay variación con el tiempo en la corriente de campo por lo que el segundo término de la expresión (7.1) se anula y se tiene por tanto que

$$E_f = R_f \cdot I_f = \frac{R_f}{K_f} \cdot \Phi_f \quad (7.4)$$

El circuito de armadura y el circuito de campo, pueden ser interconectados según tres configuraciones básicas que se resumen en la Fig.7.9. De cada tipo de conexión resultan motores que presentan diferentes propiedades y que resultan adecuados para diferentes aplicaciones. Para la conexión en serie Fig.7.9(a), la corriente de armadura  $i_a$  y la corriente de campo  $i_f$  son idénticas, de modo que la curva de salida del motor (curva que representa el par frente a la velocidad) presenta una forma hiperbólica que se esquematiza en la Fig.7.10(a). Para las conexiones en paralelo Fig.7.9(b), y de excitación separada Fig.7.9(c), la curva de salida presenta la forma que se muestra en la Fig.7.10(b).

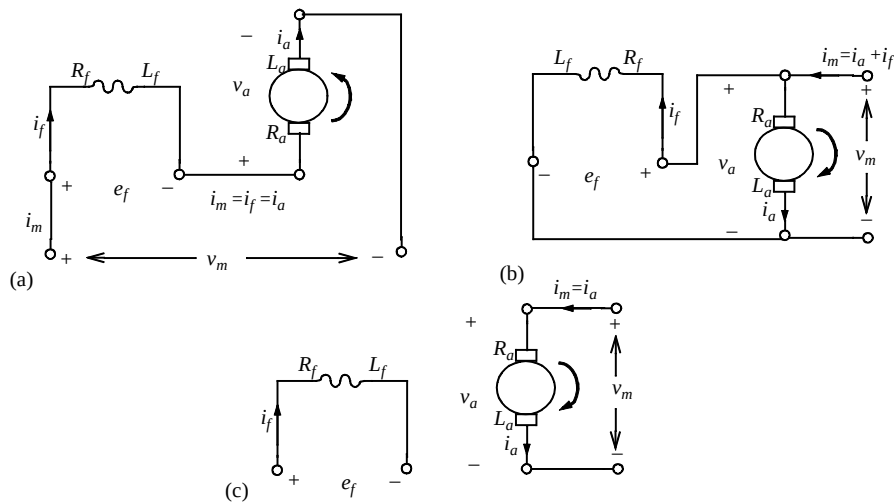


**Fig.7.9 Configuraciones básicas de motores de corriente continua: motor en serie (a); motor en paralelo (b); motor de excitación independiente (c).**

En cualquiera de los tres casos, cuando la corriente  $i_a$  circula por el bobinado de armadura, sobre él se ejerce una fuerza resultado de la interacción de esta corriente con el campo magnético de flujo  $\Phi$  en el entrehierro, (debido principalmente a  $\Phi_f$ ); como resultado, sobre este elemento se genera un par electromagnético instantáneo  $T_{em}(t)$  de valor dado por la expresión

$$T_{em}(t) = K_T \cdot \Phi \cdot i_a \quad (7.5)$$

que en términos de estado estacionario da lugar a un valor medio en el tiempo para  $T_{em}$ , dado por la expresión

$$T_{em} = K_T \cdot \Phi \cdot I_a, \quad (7.6)$$

que pone de manifiesto la relación de proporcionalidad entre el par medio  $T_{em}$  y el valor medio de la corriente de armadura  $I_a$  para el caso de  $\Phi$  constante.

A su vez, la rotación del conductor de armadura en el campo magnético induce en el circuito de armadura una fuerza electromotriz de polaridad inversa al flujo de corriente en la armadura, (de ahí el nombre de fuerza contraelectromotriz que usualmente recibe) y que para valores instantáneos puede expresarse como

$$e_b = K_E \cdot \Phi \cdot \omega_m \quad (7.7)$$

donde  $\omega_m$  denota la velocidad angular instantánea.

En términos de valores medios, válidos para operación en estado estacionario esta relación se expresa

$$E_b = K_E \cdot \Phi \cdot \Omega_m \quad (7.8)$$

En las anteriores expresiones las constantes  $K_T$  y  $K_E$  toman idéntico valor si se expresan en las unidades del sistema internacional, por lo que combinando las expresiones (7.6) y (7.8) se tiene

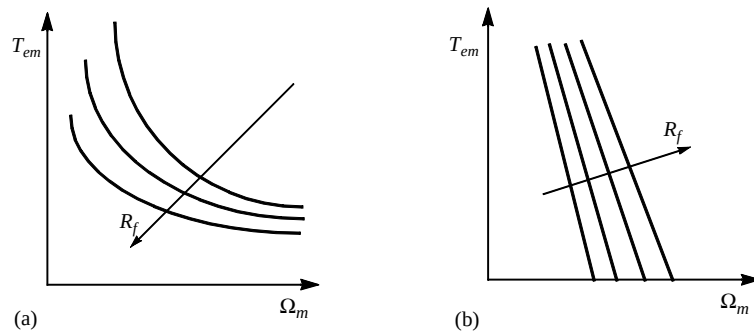


Fig.7.10 Características Par-Velocidad de un motor de corriente continua: Motor serie (a); motor paralelo y de excitación separada (b).

la potencia desarrollada por el motor según la expresión

$$P = T_{em} \cdot \Omega_m = \frac{K_T}{K_E} \cdot E_b \cdot I_a = E_b \cdot I_a \quad (7.9)$$

Todas las expresiones vistas, desde la (7.1) a la (7.9), son válidas para cualquiera de los tres tipos de motores que se muestran en la Fig.7.9. Éstos sin embargo difieren en las expresiones que definen las tensiones en sus terminales,  $v_m$ , en dicha figura.

Para un motor con conexión en paralelo o de excitación separada se tiene para la tensión instantánea en sus terminales la expresión

$$v_m = v_a = i_a \cdot R_a + L_a \cdot \frac{di_a}{dt} + e_b \quad (7.10)$$

mientras que para un motor conectado en serie se tiene

$$v_m = i_a \cdot (R_a + R_f) + (L_a + L_f) \cdot \frac{di_a}{dt} + e_b \quad (7.11)$$

ambas expresiones resultan de emplear el modelos de la Fig.7.8(b) para el circuito de bobinado de campo y Fig.7.8(d) para el circuito de armadura.

Si consideramos el sistema en estado estacionario los efectos inductivos son despreciables y las ecuaciones (7.10) y (7.11) pueden escribirse como

$$V_m = I_a \cdot R_a + E_b \quad (7.12)$$

para motores con conexión en paralelo o de excitación separada; y

$$V_m = I_a \cdot (R_a + R_f) + E_b \quad (7.13)$$

para motores con conexión en serie.

Todas las variables que intervienen en estas expresiones han de ser consideradas como valores medios. Combinando adecuadamente las anteriores expresiones se obtienen las expresiones de la velocidad  $\Omega_m$ , y el par motor  $T_{em}$ , en estado estacionario para los diferentes tipos de motores, las cuales se reúnen en la Tabla 7.1. Estas expresiones son válidas si el motor no opera en saturación profunda, situación para la cual la expresión (7.3) no es válida, y éstas han de ser modificadas.

Por su parte, en el caso de un motor cuyo campo magnético sea generado por un imán permanente, en las condiciones adecuadas, éste puede ser considerado constante, por tanto  $\Phi$  constante, por lo que para este motor se verifican expresiones similares a las que se han visto para un motor excitado de forma independiente, cambiando solamente el valor de las distintas constantes que aparecen en ellas.



**Tabla 7.1:** Expresiones de la velocidad y el par en régimen estacionario para diferentes motores de corriente continua

| Tipo de Motor         | Motor paralelo                                                                                                                              | Motor de excitación separada                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Velocidad de rotación | $\Omega_m = \frac{V_m - (I_a \cdot R_a)}{K_E \cdot \Phi} =$ $= \frac{V_m}{K_E \cdot \Phi} - \frac{R_a}{K_E \cdot K_T \cdot \Phi^2} T_{em}$  |                                                                                              |
| Par                   | $T_{em} = V_m^2 \frac{K_T K_f}{R_a R_f} \left( 1 - \frac{K_E K_f}{R_f} \Omega_m \right)$                                                    | $T_{em} = \frac{K_T K_f E_f}{R_a R_f} \left( V_m - \frac{K_E K_f E_f}{R_f} \Omega_m \right)$ |
| Tipo de Motor         | Motor serie                                                                                                                                 |                                                                                              |
| Velocidad de rotación | $\Omega_m = \frac{V_m - I_a \cdot (R_a + R_f)}{K_E \cdot \Phi} =$ $= \frac{V_m}{K_E \cdot K_f \cdot I_a} - \frac{R_a + R_f}{K_E \cdot K_f}$ |                                                                                              |
| Par                   | $T_{em} = V_m^2 \frac{K_T K_f^2}{[(R_a + R_f) + K_E K_f \Omega_m]^2}$                                                                       |                                                                                              |

Finalmente, el par  $T_{em}$  ha de cumplir la ecuación dinámica

$$T_{em} = K_i J \frac{d\Omega_m}{dt} + B\Omega_m + T_L \quad (7.14)$$

donde  $K_i$  es una constante,  $J \frac{d\Omega_m}{dt}$  es el par de inercia, el término  $B\Omega_m$  tiene en cuenta el par de fricción, y  $T_L$  es el par de carga.

## 3.2.1. Corriente de armadura durante del transitorio de arranque

Para analizar cuál es la situación en el instante inicial del transitorio de arranque, consideremos las expresiones (7.10) y (7.11), que pueden ser estudiadas conjuntamente si se considera la expresión

$$v_m = i_a \cdot R_m + L_m \cdot \frac{di_a}{dt} + e_b \quad (7.15)$$

donde  $R_m = R_a$  y  $L_m = L_a$  para motores en paralelo, de excitación separada o de imán permanente, mientras que  $R_m = R_a + R_f$  y  $L_m = L_a + L_f$  para motores conectados en serie.

Un arranque directo es equivalente a considerar que en los terminales del motor se aplica una entrada escalón en la forma

$$v_m = \begin{cases} 0 & \text{para } t < 0 \\ V_m & \text{para } t \geq 0 \end{cases}, \quad (7.16)$$

suponiendo que en el instante ( $t = 0$ ) la corriente de armadura toma el valor cero, ( $i_a(0) = 0$ ), y teniendo en cuenta que, como el motor está inicialmente parado ( $\omega = 0$ ), el término  $e_b$  que modela la fuerza electromotriz inducida en el bobinado de armadura es nulo, dada su dependencia con  $\omega$  que expresa (7.7), la solución de (7.15)

$$i_a(t) = \frac{V_m - e_b}{R_m} (1 - e^{-t(R_m/L_m)}) = \frac{V_m}{R_m} (1 - e^{-t(R_m/L_m)}) \quad (7.17)$$

proporciona una aproximación del comportamiento de dicha corriente de armadura en los instantes iniciales del transitorio de arranque. Dado que la constante de tiempo mecánica  $\tau_m^{1\ddagger}$ , que determina el tiempo que tarda la velocidad en alcanzar su valor estacionario, es mucho mayor que la constante de tiempo eléctrica  $\tau_e = L_m/R_m$ , que determina el tiempo de subida de la corriente de armadura, puede estimarse que en esos instantes iniciales  $i_a$  llegará a alcanzar un valor próximo a

$$I_{pk} = V_m/R_m, \quad (7.18)$$

el cual, dado el pequeño valor de  $R_m$ , resulta ser mucho mayor que el valor de la corriente en estado estacionario  $I_a$ . A medida que el motor se acelera, el término  $e_b$  crece de manera que  $i_a(t)$  decrece, al disminuir el factor  $V_m - e_b$ , hasta alcanzar el valor  $I_a$ . Dado el pequeño valor de  $L_m$  en los motores de corriente continua, que resulta además adecuado para reducir las pérdidas de potencia en el motor y garantizar una respuesta rápida a las demandas de el par de carga, el valor de  $\tau_e$  resulta ser pequeño en términos absolutos, en el rango de los microsegundos

<sup>1\ddagger</sup>. Esta contante de tiempo se determina resolviendo el sistema de ecuaciones que forman las expresiones (7.14) y (7.15) [Moha95].

[Guar97], por lo que cualquier acción de control a este nivel exigirá muy alta velocidad de respuesta.

### 3.3. Control de motores de corriente continua

Los aspectos más importantes a controlar en un motor de corriente continua son la velocidad y el par desarrollado.

En este tipo de motores, la variación de la velocidad se consigue habitualmente mediante dos procedimientos:

- a) variando la resistencia  $R_f$  del bobinado de campo, esto es, modificando el flujo magnético  $\Phi$ <sup>2†</sup>; o bien,
- b) variando la tensión de armadura  $V_m$ .

El método más eficiente y económico es el segundo, y una manera simple de conseguir una tensión  $V_m$  variable a partir de una fuente de tensión constante es mediante un convertidor continua-continua como el que se muestra en la Fig.7.1, y es por ello el procedimiento empleado en la aplicación ejemplo. Por otra parte, controlando la tensión de armadura también se controla la corriente de armadura  $I_a$ , y por tanto el par desarrollado por el motor.

A continuación se describen brevemente los principios básicos del funcionamiento y las principales características del convertidor continua-continua más simple, así como diferentes estrategias de control en lazo abierto y en lazo cerrado.

#### 3.3.1. Convertidor DC-DC o Troceador

El principio de funcionamiento de un convertidor de este tipo consiste en aplicar a la carga un tren de pulsos de tensión, según el esquema que muestra la Fig.7.11. En estos sistemas se denomina ciclo de trabajo, (*duty-cycle*) a la razón

$$D_T = \frac{T_{on}}{T_{on} + T_{off}} \quad (7.19)$$

donde  $T_{on}$  y  $T_{off}$  son los intervalos de tiempo señalados en la Fig.7.11(b). Controlando adecuadamente el tiempo de conducción  $T_{on}$ , esto es, el ancho del pulso de la señal  $v_L$ , es posible controlar el valor medio de la tensión que aparece en los terminales de la carga  $V_L$ , de forma que se tiene

$$V_L = \frac{T_{on}}{T_{on} + T_{off}} V_S = D_T V_S \quad (7.20)$$

---

<sup>2†</sup>. Obviamente esta opción no es válida en motores de imán permanente.

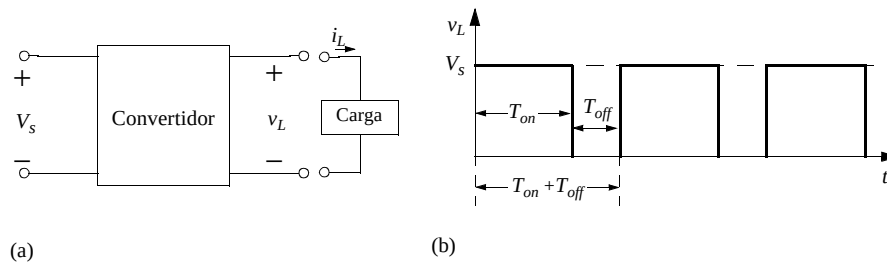


**Fig.7.11 Convertidor troceador: Conexión (a); y forma de onda aplicada a la carga.**

De esta manera, si se desea aplicar a la carga un valor medio de tensión dado  $V_L$ , distinto de  $V_S$ , basta con determinar  $D_T$  a partir de la expresión (7.20) y generar un tren de pulsos de periodo constante y  $D_T$  constante. Si se requiere poder variar el valor  $V_L$ , pero manteniendo fijo el periodo del tren de pulsos,  $T_{on} + T_{off}$ , se tiene que  $v_L$  es una señal modulada por anchura de pulso, (PWM). Por último si para conseguir  $V_L$  variable,  $T_{on}$  se mantiene fijo y se hace variar  $T_{off}$  se tiene que  $v_L$  es una señal modulada en frecuencias.

El convertidor más simple de este tipo está formado por un *switch* electrónico S, y un diodo rectificador D. En la Fig.7.12, además de estos elementos, se muestra un circuito de control de conmutación basado en un generador de señal PWM, además de una carga inductiva, en particular un motor de corriente continua, que ilustran un sistema simple de control de velocidad de motores. La forma de onda para  $v_L$  que se muestra en la Fig.7.11(b) es válida para este tipo de carga siempre que la corriente de carga  $i_L$  sea continua. A este tipo de convertidor se le conoce también como convertidor directo o *buck converter*, y se caracteriza porque el valor de salida  $V_L$  nunca excede al valor de entrada  $V_S$ . Cuando el *switch* S conduce, la tensión

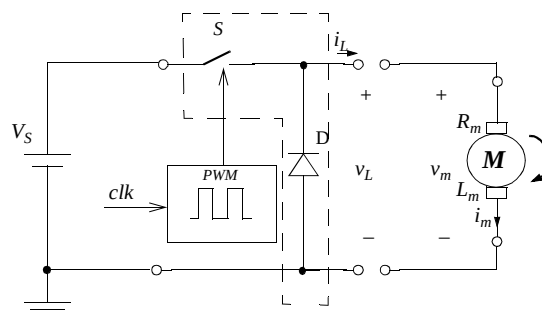


**Fig.7.12 Convertidor DC básico, controlado por un circuito PWM y conectado a un motor de corriente continua.**

de alimentación  $V_S$  es aplicada a la carga y el valor medio de la corriente se incrementa hasta alcanzar el valor  $I_{Lmax}$  justo en el momento en que S se abre. Cuando S está abierto la tensión  $v_L$  toma un valor próximo a cero, dado que, en el caso de que la corriente de carga continúe fluyendo, cosa que es cierta para cargas inductivas, el diodo D conduce. En este caso  $I_L$  empieza a decrecer hasta alcanzar el valor  $I_{Lmin}$ , justo en el momento en que S vuelve a estar cerrado. Dado que  $I_L$  nunca toma valores nulos, se dice que el sistema funciona con corriente de carga continua. La forma de onda típica para la tensión y la corriente de carga en sistemas como el de la Fig.7.12 se muestra en la Fig.7.13. En ella se aprecia que la corriente de carga presenta un rizado en torno a un valor medio  $I_L$ . La amplitud de este rizado se evalúa según la expresión

$$\Delta I = I_{Lmax} - I_{Lmin} = \frac{V_S D_T (1 - D_T)}{f L_m} \quad (7.21)$$

donde  $f$  es la frecuencia fundamental de la señal  $v_L$ , y  $L_m$  es la inductancia del motor. De esta expresión se deduce que, para conseguir un rizado pequeño, para cualquier valor del ciclo de trabajo  $D_T$ , se ha de tener  $f$  y  $L_m$  lo más grande posible.  $L_m$  suele ser pequeña, por lo que una opción habitualmente adoptada es incrementar su valor añadiendo un inductor externo. Otra opción es incrementar la frecuencia  $f$ ; esta opción resulta ventajosa siempre que ello no suponga un excesivo consumo de potencia debido al dispositivo de conmutación, cuyo consumo en general aumenta con la frecuencia de trabajo. Finalmente cabe señalar que este sistema proporciona una corriente unidireccional para cualquier tipo de carga y que su tensión no es reversible.

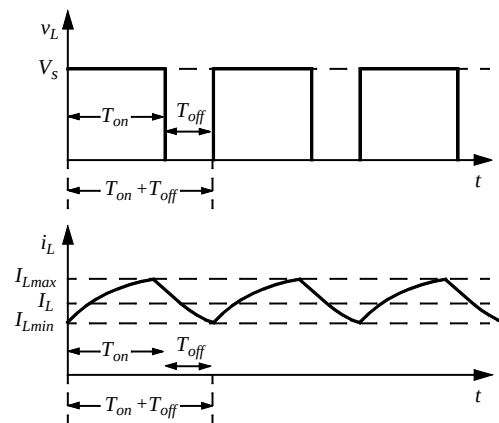


Fig.7.13 Formas de onda para la tensión y la corriente de carga para un convertidor troceador básico con carga inductiva.

### 3.3.2. Control en lazo abierto.

El control en lazo abierto es empleado cuando no es necesaria una gran precisión en la velocidad que alcanza el motor. En este caso la variable a controlar es la tensión en la armadura del motor. Aplicaciones de este tipo son sistemas de tracción, bombeo, ventilación, etc. También es empleado cuando se quiere controlar el par desarrollado, limitando la corriente, para proteger tanto la electrónica, como a la carga. En este caso la variable a controlar es la corriente de armadura. Entre las aplicaciones de este tipo se encuentran sistemas empleados en el tensado de bobinas, o de los cabezales de lectura de equipos de sonido o imagen.

Como ejemplo de este tipo de control resulta válido el sistema de la Fig.7.12, en el que la elección de un ciclo de trabajo adecuado permite determinar la tensión en la armadura y por tanto determinar, aunque de manera no muy precisa, la velocidad y el par desarrollado.

### 3.3.3. Control en lazo cerrado.

El control de motores en lazo cerrado es llevado a cabo en aplicaciones en las que se necesita velocidad variable, o ajuste de velocidad y posición, y limitación de corriente con bastante precisión.

En la Fig.7.14 se muestran diferentes esquemas de control en lazo cerrado empleados en estas aplicaciones. En el esquema de la Fig.7.14(a) se emplea un lazo interno de control para limitar la corriente en el propio convertidor empleando como variable de control la diferencia entre el valor de referencia y la corriente de armadura medida. De esta forma el convertidor electrónico de potencia opera como un convertidor regulado por corriente. Este lazo interno de control mejora el tiempo de respuesta del sistema. El control de velocidad se realiza mediante un lazo de control externo que, a partir de la medida de velocidad y su comparación con la velocidad de referencia, establece el par necesario, y a partir de él, la corriente que sirve de referencia al bucle interno.

En el esquema de la Fig.7.14(b) el control del convertidor se realiza a partir de la diferencia entre la velocidad de referencia y la velocidad medida. Esta acción de control es suspendida si la corriente medida excede el límite definido. En este caso se tiene un doble bucle de control que actúa simultáneamente. En ambos casos el límite de corriente puede ser a su vez dependiente de la velocidad.

Un ejemplo que incorpora un controlador difuso al lazo de control que se muestra en la Fig.7.14(a) es el de ejemplo ya citado [Sous94] y que se ilustra en la Fig.7.6.

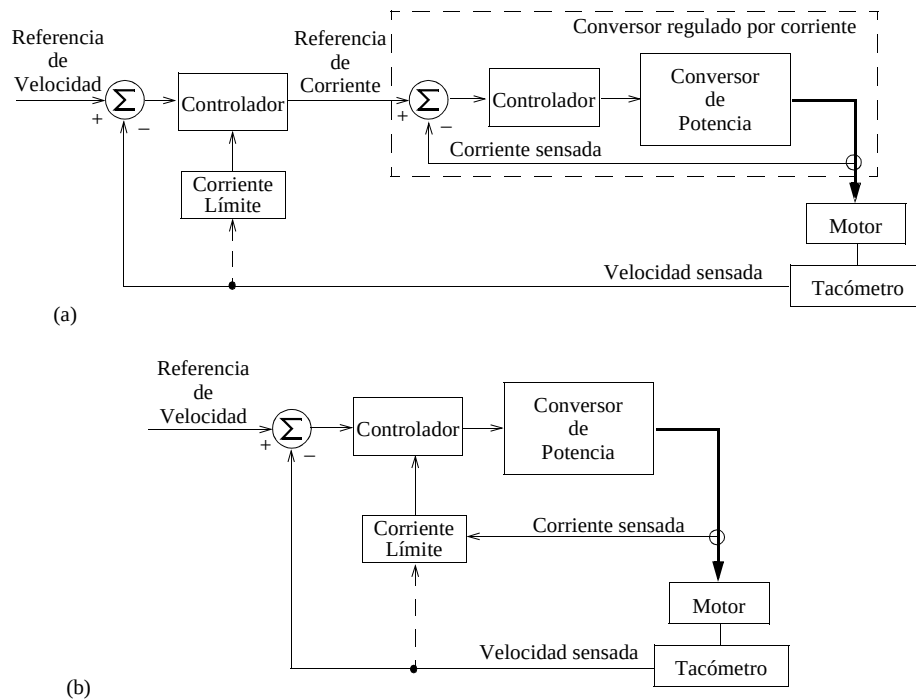


**Fig.7.14 Lazos de control en servosistemas y sistemas de velocidad variable: lazo de corriente interno (a); lazo de corriente externo.**

### 3.4. Estrategias para el arranque suave de motores de corriente continua.

#### 3.4.1. Control en lazo abierto

El procedimiento de arranque directo de un motor DC hace que éste desarrolle el mayor par de arranque posible, con lo que se consigue también el arranque más rápido posible, esto es, que el motor alcance su estado estacionario en el menor tiempo posible. Sin embargo, como se ha visto anteriormente, en estas circunstancias tanto el bobinado de armadura, como el sistema de alimentación se ve forzado a conducir una corriente muy superior a la corriente en régimen estacionario, que alcanzar un valor de pico dado por (7.18), y que puede dar lugar a efectos perjudiciales tanto para el propio motor como para el sistema de alimentación, como también ya se ha mencionado. Por ello es habitual emplear métodos que evite esta situación, sacrificando en cierta medida ambas prestaciones, par y tiempo de arranque. Estos procedimientos son conocidos como métodos de arranque suave (*soft-start*).

Las principales estrategias en lazo abierto para conseguir un arranque suave se basan en forzar un cambio gradual de alguna de las variables que aparecen en el lado derecho de la expresión (7.18), de forma que su evolución siga una rampa de pendiente suave, que evite que la corriente de armadura  $i_a(t)$  alcance valores no deseados. En general se tienen dos opciones:

- Forzar un cambio gradual en la variable  $V_m$ , comenzando por un valor pequeño, pero suficiente para que el motor empiece a girar, e ir incrementando este valor hasta alcanzar el valor final  $V_{mss}$  que define el estado estacionario.
- Forzar un cambio gradual en el valor de la resistencia del motor, que es vista desde la fuente de alimentación, añadiendo una serie de resistencias  $R_s$  de  $n$  tramos, en serie con  $R_m$ , de forma que, partiendo de un valor elevado, pero tal que el motor pueda comenzar a girar, el valor de la resistencia decrezca hasta que se alcance finalmente el valor  $R_m$  cuando el motor haya alcanzado en régimen estacionario.

En la Fig.7.15 se ilustran esquemáticamente ambos procedimientos. Lo habitual es que la evolución temporal de cada una de las magnitudes que determinan el arranque suave sea establecida por un temporizador que es ajustado en base a la experiencia y el tiempo de arranque deseado. La principal ventaja de estos procedimientos radica en su simplicidad, sin embargo ambos resultan ser poco flexibles ante variaciones del entorno de funcionamiento y

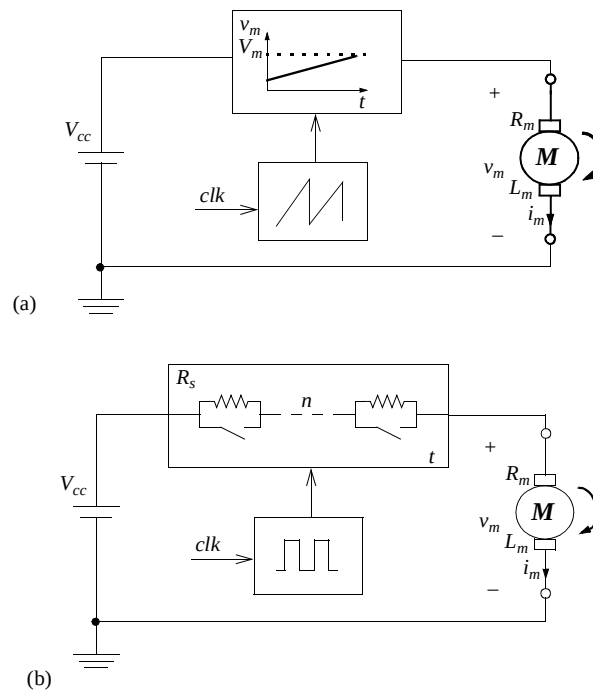


Fig.7.15 Estrategias de arranque suave: Rampa en la tensión de entrada (a); Resistencias en serie (b).



su eficiencia, en cuanto a par desarrollado y velocidad de arranque, e incluso consumo de potencia puede ser mejorada. Como ejemplo considérese el caso sencillo de arranque que se ilustra en la Fig.7.16, en la que se presenta el resultado de una simulación *MATLAB*®/*SIMULINK* del arranque suave de un motor empleando resistencias en serie con su resistencia de armadura, Fig.7.16(b), y se compara con el arranque directo de ese mismo motor, Fig.7.16(a), [Ong 98]. En este ejemplo la corriente de armadura, curva superior en ambas gráficas, queda limitada a un valor máximo de 100A, Fig.7.16(b), frente a los 500A que alcanza en el arranque directo, Fig.7.16(a). Este hecho determina el valor de  $R_s$ , el cual a su vez debe ser lo más pequeño posible para minimizar la potencia en ella disipada. Por su parte, como se aprecia claramente en la Fig.7.16(b) la corriente de armadura presenta un rizado que se produce cada vez que se cortocircuita un nuevo tramo de resistencias. Este rizado aparece también, en virtud de la expresión (7.6), en el par desarrollado. Para minimizar la amplitud de este rizado se hace necesario disponer de un número suficiente de tramos de resistencias, con el consiguiente aumento en la complejidad del control. Además, todos estos factores han de

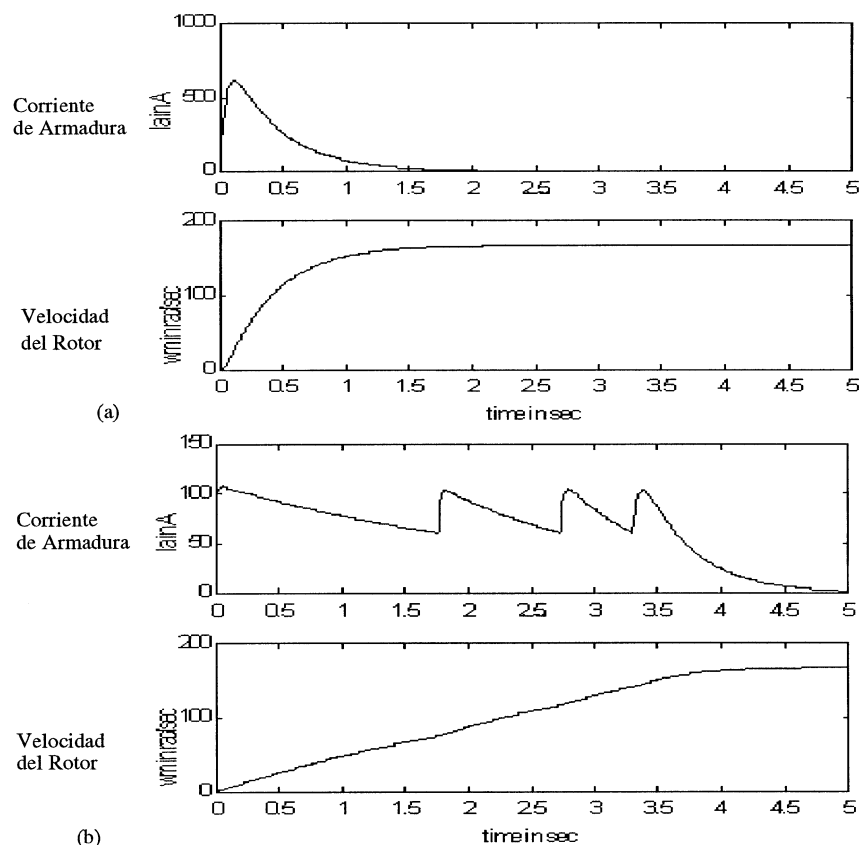


Fig.7.16 Ejemplo de simulación de arranque : Arranque directo(a); arranque suave con resistencias en serie (b).

quedar fijados al establecer la estrategia de arranque, lo que dificulta su posterior reajuste o capacidad de adaptación posterior.

Los dos procedimientos de arranque antes mencionados resultan ser equivalentes cuando se emplea un convertidor DC-DC como el descrito en el apartado 3.3.1. para alimentar el motor y proporcionar una tensión variables en sus terminales. Dado que, por la expresión (7.20) se tiene que  $V_m = D_T V_s$  y que para un convertidor ideal se tiene la siguiente relación de transformación de impedancias

$$R_s = \frac{R_m}{D_T^2} \quad (7.22)$$

donde  $R_s$  es la resistencia vista desde los terminales de la fuente de alimentación; forzando un incremento gradual del ciclo de trabajo del convertidor DC-DC,  $D_T$ , desde un valor mínimo  $D_{Tmin}$  hasta un valor dado  $D_{TS}$ , es posible realizar simultáneamente cualquiera de los procedimientos de arranque suave antes mencionados.

#### 3.4.2. Control en lazo cerrado

Dado que en los esquemas de control en lazo cerrado se emplean habitualmente como señales de entrada del controlador medidas de la velocidad de giro y de la corriente de armadura del motor, una estrategia de arranque, también en lazo cerrado, empleando estas mismas variables, que optimizase esta fase de operación, podría ser llevada a cabo sin demasiado coste adicional. En este tipo de sistemas lo habitual es que el control de todo el proceso recaiga en un microcontrolador, el cual establece los valores de referencia que rigen el comportamiento del sistema en cada instante. El proceso de arranque se realiza también bajo el control del microprocesador, que establece una rampa de valores de referencia para la velocidad como una sucesión de valores discretos de tensión de alimentación del motor durante esta fase. Lo habitual es que la temporización de esa sucesión de valores se establezca *a priori*, sin control realimentado, y por lo tanto de una forma no optimizada, dado el pequeño tiempo de respuesta que requeriría un control en lazo cerrado, sobre todo para seguir las variaciones de la variable corriente de armadura. Téngase en cuenta que al tiempo de ejecución del programa de control, hay que añadir el tiempo de adquisición y conversión de las medidas de velocidad y corriente, para evaluar el tiempo de respuesta del sistema de control.

Una estrategia de control en lazo cerrado, es la estrategia seguida en el ejemplo de aplicación del controlador ejemplo MFCON, la cual se detalla a continuación.

### 3.5. Arranque suave de motores de corriente continua empleando el controlador difuso MFCON en el lazo de control.

La Fig.7.17 muestra el esquema del sistema utilizado en la aplicación de arranque suave desarrollada que incorpora MFCON en el lazo de control. En ella aparece un motor de corriente continua alimentado por un convertidor DC-DC, que permite aplicar una tensión variable a sus terminales, cuyo valor es determinado, según la expresión (7.20), por el ciclo de trabajo  $D_T$  de la señal de control  $S_C$ , generada por un circuito PWM. El valor  $D_T$  es determinado a su vez en el lazo de control en el que interviene MFCON, a partir de las magnitudes velocidad  $\Omega_m$  y corriente de armadura  $I_m$  sensadas.  $D_T$  es forzado a cambiar de forma gradual, para conseguir un incremento gradual de la tensión de armadura  $V_m$  según la estrategia de control que se describe a continuación. Por otra parte la frecuencia de la señal de reloj  $clk$ , que sirve de base al circuito de PWM, se ha elegido de un valor elevado para forzar una acción de control rápida que ilustre las prestaciones de MFCON.

#### 3.5.1. Estrategia de control

Haciendo uso de las expresiones empleadas en el apartado 3.2. para modelar el motor de corriente continua en régimen estacionario, las expresiones (7.14) y (7.15) que describen su comportamiento dinámico pueden escribirse en la forma

$$\frac{dI_m}{dt} = \frac{1}{L_m}(V_s \cdot D_T - (R_m \cdot I_m + K_E \cdot \Phi \cdot \Omega_m)) \quad (7.23)$$

$$\frac{d\Omega_m}{dt} = \frac{1}{K_i J}(K_T \cdot \Phi \cdot I_m - T_L - B\Omega_m) \quad (7.24)$$

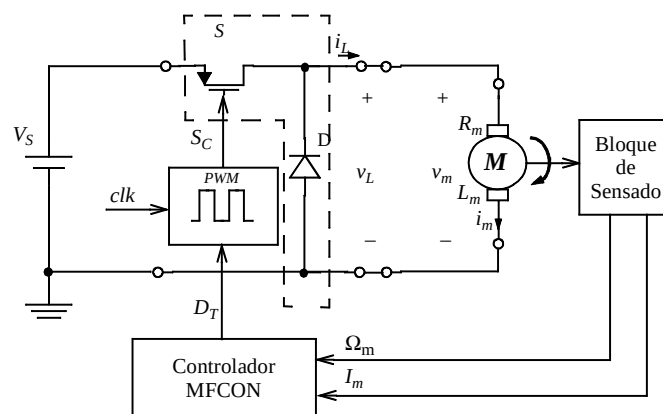


Fig.7.17 Esquema de control de la aplicación ejemplo

donde también se ha considerado la expresión (7.20) que relaciona el valor medio de la tensión en los terminales del motor  $V_m$  con la tensión continua de alimentación  $V_s$ .

Estas expresiones permiten escribir las ecuaciones del sistema de control en términos de variables de estado  $(\Omega_m(t), I_m(t))$  en la forma

$$\begin{aligned} I_m(t) &= \Phi_{I_m}(\Omega_m(t), I_m(t), D_T) \\ \dot{\Omega}_m(t) &= \Phi_{\Omega_m}(\Omega_m(t), I_m(t), D_T) \\ D_T &= g(\Omega_m(t), I_m(t)) \end{aligned} \quad (7.25)$$

donde las dos primeras expresiones coinciden con las expresiones (7.23) y (7.24) respectivamente, mientras que la tercera expresión representa a la función de control. Esta función, cuya expresión analítica es desconocida a priori, ha de ser escogida de forma que el estado del sistema se vea forzado a seguir la trayectoria  $(\Omega_{md}(t), I_{md}(t))$  dada por el objetivo de control.

En esta aplicación, la trayectoria de control está determinada por dos condiciones:

- El arranque de motor ha de ser seguro, esto es, se debe evitar que la corriente de armadura durante la fase de arranque sobrepase un nivel considerado aceptable.
- El arranque ha de ser lo más rápido posible.

Para cumplir estas dos condiciones se propone mantener la corriente de armadura lo más próxima posible al valor máximo permitido

$$I_{md}(t) = I_{mmax} \quad (7.26)$$

y durante el mayor tiempo posible. De esta forma el motor arrancará con el par máximo y constante durante el fase de arranque. Nótese que esta situación no es equivalente a realizar solamente una limitación de la corriente que circula por el motor, ya que en ese caso no se ejerce acción de control sobre el par, el cual decrece a medida que el motor se acelera. Este aspecto se verá posteriormente con más detalle al mostrar los resultados experimentales.

Asumiendo la expresión (7.26) como trayectoria objetivo para la variable de estado  $I_m$ , como trayectoria objetivo para la variable  $\Omega_m$  se ha escogido la expresión que determina la velocidad en estado estacionario que se recoge en la Tabla 7.1. Esta expresión es distinta según el tipo de motor considerado. Para un motor conectado en paralelo, excitado separadamente, o de imán permanente se tiene

$$\Omega_{md}(t) = \frac{D_T(t) \cdot V_s}{K_E \Phi} - \frac{I_{mmax} \cdot R_a}{K_E \Phi} \quad (7.27)$$

mientras que para un motor serie se tiene

$$\Omega_{md}(t) = \frac{D_T(t) \cdot V_s}{K_E K_f I_{mmax}} - \frac{R_a + R_f}{K_E K_f} \quad (7.28)$$

De esta manera la trayectoria de arranque se desarrolla a través de una sucesión de estados estacionarios entre los que se evoluciona de forma controlada.

En esta aplicación la función de control,  $D_T = g(\Omega_m(t), I_m(t))$ , necesaria para conseguir el objetivo, y cuya expresión analítica no esta completamente definida, es aproximada por el controlador difuso MFCON a partir de las siguientes reglas

$$\text{SI } I_m(t) \text{ ES MENOR QUE } I_{mmax} \text{ ENTONCES } D_T = D_{TS} \quad (7.29)$$

$$\text{SI } I_m(t) \text{ ES MAYOR QUE } I_{mmax} \text{ ENTONCES } D_T = D_{Tmin} \quad (7.30)$$

$$\text{SI } I_m(t) \text{ ES IGUAL A } I_{mmax} \text{ ENTONCES } D_T = D_{Tmd}(I_m, \Omega_m) \quad (7.31)$$

donde  $D_{TS}$  y  $D_{Tmin}$  son respectivamente los ciclos de trabajo objetivo y mínimo que pueden ser aplicados, mientras  $D_{Tmd}$  se obtiene, según el tipo de motor considerado, a partir de las expresiones (7.27) o (7.28). Para un motor conectado en paralelo, excitado separadamente, o de imán permanente se tiene

$$D_{Tmd} = \frac{I_m \cdot R_a}{V_s} + \frac{\Omega_m K_E \Phi}{V_s} \quad (7.32)$$

mientras que para un motor serie se tiene

$$D_{Tmd} = \frac{R_a + R_f + K_E K_f \Omega_m}{V_s} I_m \quad (7.33)$$

Las dos primeras reglas fuerzan a la corriente  $I_m$  a seguir a  $I_{mmax}$  como exige la expresión (7.26). La tercera regla fuerza a la velocidad a seguir la trayectoria dada por las expresiones (7.27) o (7.28) según sea el motor considerado, en la medida que  $I_m$  sea  $I_{mmax}$ . Finalmente indicar que la acción de control finaliza cuando la velocidad alcanza su valor estacionario  $\omega_s$ , y por tanto  $D_T$  alcanza el valor  $D_{TS}$ .

Hay que señalar que, aunque las expresiones (7.29)-(7.31) están enunciadas en términos de reglas SI-ENTONCES, éstas no son exactamente las reglas difusas definidas para MFCON. Las reglas difusas que constituyen la matriz de reglas que realiza MFCON son construidas a partir de las primeras como se mostrará más adelante en cada uno de los experimentos propuestos.

## 3.5.2. Descripción del sistema experimental

La Fig.7.18 muestra el diagrama de bloques del sistema experimental empleado en este ejemplo de aplicación. En ella aparecen como principales elementos:

- Bloque **MFCON** que contiene la placa de test, descrita en el anterior capítulo, configurada para trabajar en un lazo de control.
- Un **Motor** de imán permanente que incorpora un tacogenerador inductivo que proporciona a su salida una señal sinusoidal de amplitud proporcional a su velocidad de giro.
- Un resistor  $R_{sen}$ , de pequeño valor, conectada en serie con los terminales del motor, empleada para el sensado de la corriente en sus terminales.
- Un bloque de **Interfaz de Sensado** que acondiciona las señales de sensado a los requerimientos del interfaz de entrada de MFCON.
- Un bloque de **Interfaz PWM**, que acondiciona la salida de MFCON a las exigencias del circuito que genera la señal PWM.
- Un bloque **PWM** que genera la señal de control del switch MOSFET del convertidor DC-DC.
- Un bloque **Convertidor DC-DC** que alimenta al motor.

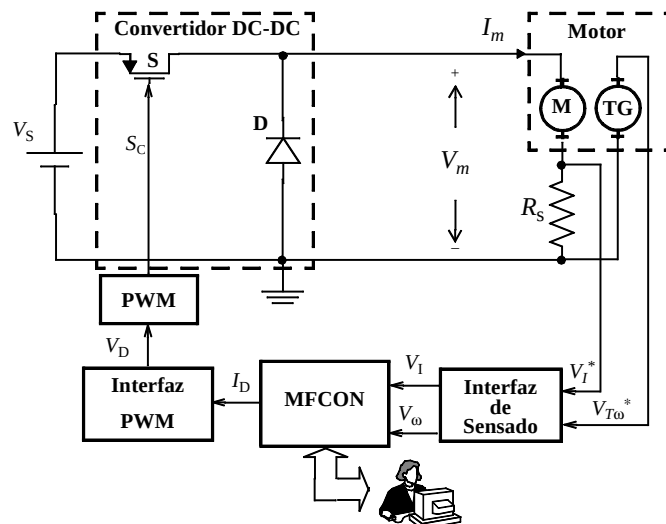


Fig.7.18 Diagrama de bloques del sistema experimental

La Fig.7.19 muestra varias fotografías del sistema experimental en el laboratorio. En ellas junto al PC y los instrumentos de medida, aparecen el motor, la *placa de test* que contiene al chip controlador, y una nueva placa que ha sido diseñada para albergar al resto de elementos que aparecen en la Fig.7.18. Esta placa que será llamada en adelante *placa de control de motor* y su fotografía se muestra en la Fig.7.20(a). La distribución de los diferentes elementos en esa placa se muestra en el esquema de la Fig.7.20(b).

A continuación se describen los principales aspectos de cada uno de los elementos que forman el sistema experimental. El principal objetivo de este diseño ha sido la sencillez en el cumplimiento de su función.

### 3.5.2.1. Bloque MFCON

Está constituido por la placa de test descrita en el capítulo anterior. Para su uso en el lazo de control se han aislado los terminales de entrada y salida del chip controlador del interfaz con el PC y se han conectado a la *placa de control de motor*. Se mantiene el interfaz de accesos a memoria para configurar y ajustar al chip MFCON según la superficie de control requerida en cada caso. Las variables de interfaz con la *placa de control de motor*,  $V_I$  y  $V_\omega$  que aparece en

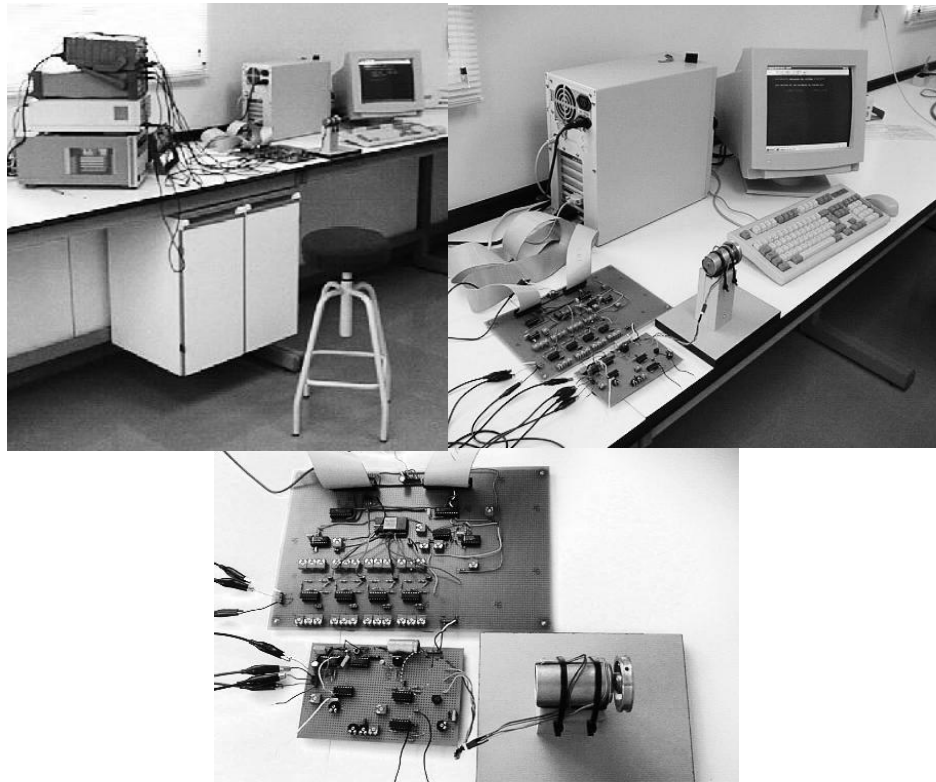
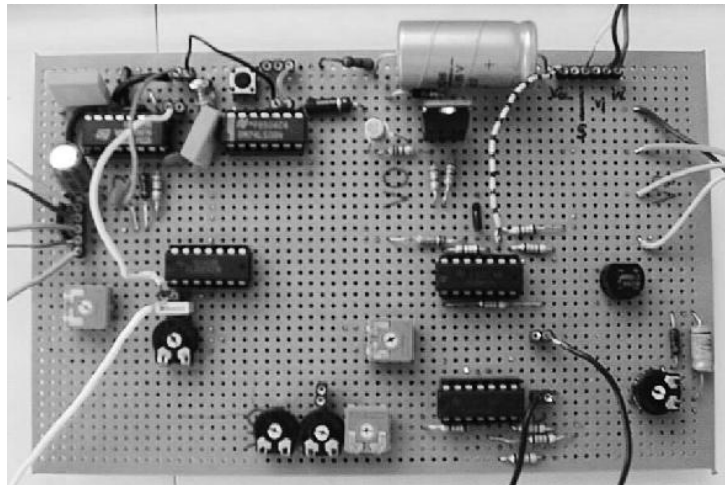


Fig.7.19 Fotografías del sistema experimental.

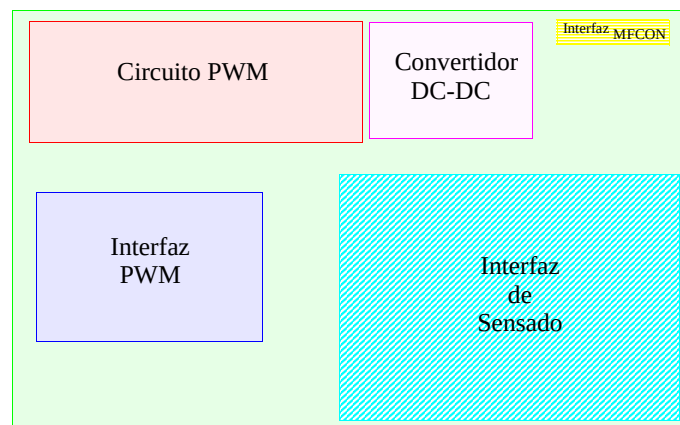
la Fig.7.18, son señales en tensión que representan, respectivamente, a la corriente y velocidad sensadas y acondicionadas para el rango de entrada de MFCON, mientras que  $I_D$  es la señal de salida en modo corriente de MFCON.

### 3.5.2.2. Motor de corriente continua de imán permanente.

Como ya se ha indicado, el motor empleado es un motor de imán permanente que incorpora un tacogenerador inductivo que proporciona a la salida una señal de tensión  $V_{T\omega}^*$  sinusoidal de amplitud proporcional a la velocidad de giro. El motor puede ser alimentado con una tensión continua no superior a 20 volts. Dado que no se ha dispuesto de la hoja de



(a)



(b)

Fig.7.20 Placa de control de motor: Fotografía (a); distribución de bloques funcionales (b).



características de este motor, los principales datos necesarios para los ejemplos realizados han sido adquiridos experimentalmente. Estos datos se recogen en el apartado dedicado a exponer los resultados de los ejemplos de aplicación y están especificados en términos de las variables que se manejan en los experimentos y que son las que aparecen en el esquema de la Fig.7.18.

### 3.5.2.3. Sensado de la corriente que circula por el motor

Existen diversos procedimientos para el sensado de la corriente  $I_m$ . En esta aplicación se ha optado por el procedimiento más sencillo que consiste en conectar una resistencia en serie  $R_{sen}$  entre uno de los terminales del motor y tierra, como aparece en la Fig.7.18. El valor de  $R_{sen}$  se escoge pequeño comparado con  $R_m$ , de forma que la caída de tensión en ella sea pequeña comparada con la tensión  $V_m$  aplicada en régimen estacionario. En el sistema experimental  $R_{sen}$  toma el valor de  $1\Omega$ . La caída de tensión en la resistencia de sensado se ha denominado  $V_I^*$  y, dado el valor escogido para la resistencia  $R_{sen}$ , su magnitud coincide numéricamente con la de la corriente que circula por ella.

### 3.5.2.4. Interfaz de Sensado

Este bloque se encarga de acondicionar las señales de sensado a los requerimientos del interfaz de entrada de MFCON. Hay dos señales que acondicionar:  $V_{T\omega}^*$  y  $V_I^*$ , que serán transformadas en  $V_\omega$  y  $V_I$  respectivamente, ambas en el rango de tensión de 2 a 4.8 volts.

$V_{T\omega}^*$  es una señal de tensión sinusoidal proveniente del tacogenerador de amplitud proporcional a la velocidad de giro del motor. Se ha verificado experimentalmente que su magnitud es próxima a la tensión continua que alimenta al motor para una velocidad dada cuando el motor funciona sin carga. La Fig.7.21 muestra el diagrama del circuito utilizado para

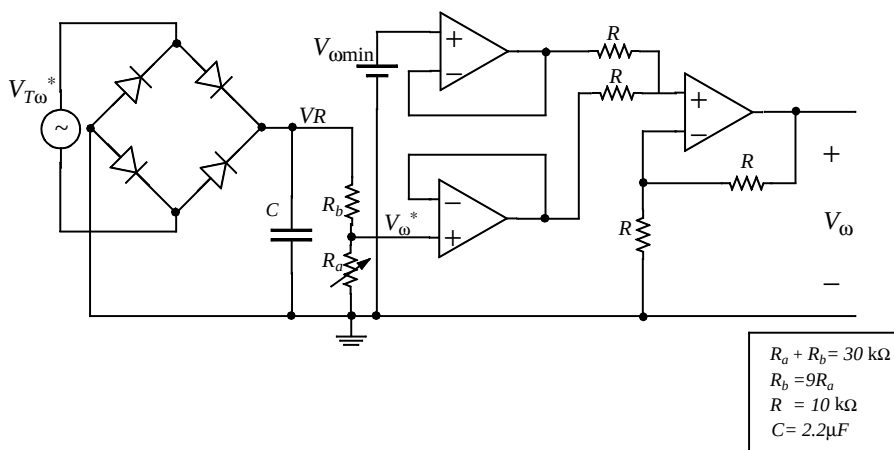


Fig.7.21 Circuito de acondicionamiento de la variable velocidad  $V_\omega$ .

su acondicionamiento. La señal del tacogenerador,  $V_{T\omega}^*$ , es rectificada mediante un puente de diodos, filtrada, dividida por un factor de diez para reducir su rango, dada la relación experimental antes mencionada, para obtener la señal  $V_\omega^*$ . Finalmente su nivel es desplazado un valor de  $V_{\omega min}$ , para que su rango coincida con el rango de entrada de MFCÓN y así obtener  $V_\omega$ . Los valores de  $R_a$ ,  $R_b$  y  $C$  han sido escogidos para tener un rizado pequeño en la señal rectificada, en todo el margen de velocidades, y  $R_a$  se realiza con un resistor variable para el ajuste del límite superior del rango de salida.

$V_I^*$  es la caída de tensión en la resistencia de sensado, experimentalmente se determina que su valor no supera en ningún caso los dos voltios, por lo que su acondicionamiento exige únicamente aislamiento, realizado por un seguidor de tensión y un desplazamiento de nivel, para que su rango coincida con el rango de entrada de MFCÓN. Estos dos requerimientos son realizados mediante el circuito que se muestra en la Fig.7.22.

### 3.5.2.5. Bloque de control PWM del convertidor DC-DC

Este bloque genera la señal PWM de control  $S_C$ , que se conecta a la puerta del switch MOSFET (**IRF9530**) del convertidor DC-DC. El ciclo de trabajo  $D_T$  de esta señal está determinado en cada instante por la tensión de la señal conectada al terminal de *Entrada* del circuito generador de señales PWM (**SG3525A**),  $V_D$ . Esta última se obtiene acondicionando la salida en modo corriente  $I_D$  de MFCÓN al rango en tensión exigido por el circuito de PWM. La Fig.7.23 muestra un diagrama en el que aparecen los elementos que componen este bloque y su conexionado.

El circuito integrado **SG3525A** es un circuito generador de señales PWM. La frecuencia de la señal de control queda fijada escogiendo adecuadamente el valor las resistencias  $R_T$  y  $R_D$  y del condensador  $C_T$  que aparecen en la Fig.7.23 según la expresión

$$f = \frac{1}{(0.7R_T + 3R_D)C_T} \quad (7.34)$$

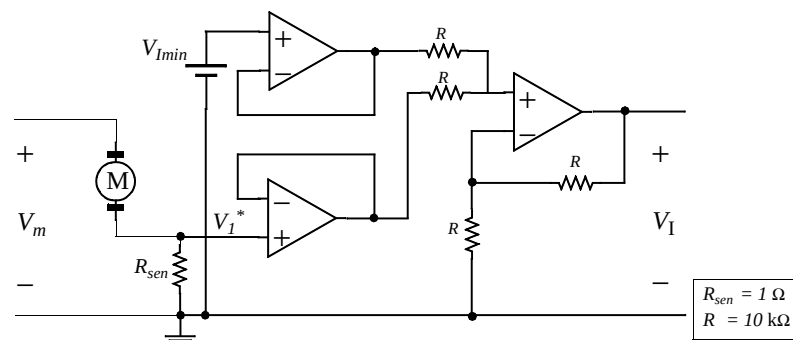


Fig.7.22 Circuito de acondicionamiento de la variable intensidad de corriente  $V_I$

Con los valores que aparecen recuadrados en la parte inferior de la Fig.7.23 se tiene una frecuencia  $f = 100.8 \text{ kHz}$ .

Para que el circuito **SG3525** funcione adecuadamente la tensión entre el terminal de *Entrada* ha de estar incluida en el rango de 0.9 a 3.6 volts. En el límite inferior se tiene una señal de ciclo de trabajo mínimo 0%, mientras que en límite superior se tiene una señal de ciclo de trabajo máximo 90%. La relación entre estas variables ha sido verificada experimentalmente y se ha obtenido

$$D_T(\%) = 29.8V_D - 19.7 \quad (7.35)$$

donde  $D_T$  se expresa en unidades de tanto por ciento, mientras que  $V_D$  se expresa en voltios. Esta expresión es válida únicamente en el rango de valores anteriormente indicado, y resulta importante a la hora de ajustar los rangos de valores de interfaz de este bloque y MFCON, como un primer paso para establecer una tabla de equivalencia valor *singleton*-ciclo de trabajo, que permita asignar valores *singleton* a la matriz de consecuentes de regla.

Para realizar los experimentos de arranque de forma controlada se ha incluido en el sistema un pulsador **P** que al estar pulsado habilita, mediante una puerta NAND la conexión de la señal de *Salida* del circuito **SG3525**, al circuito que actúa sobre el transistor MOSFET de tipo P que constituye el dispositivo de conmutación del convertidor DC-DC que alimenta al motor.

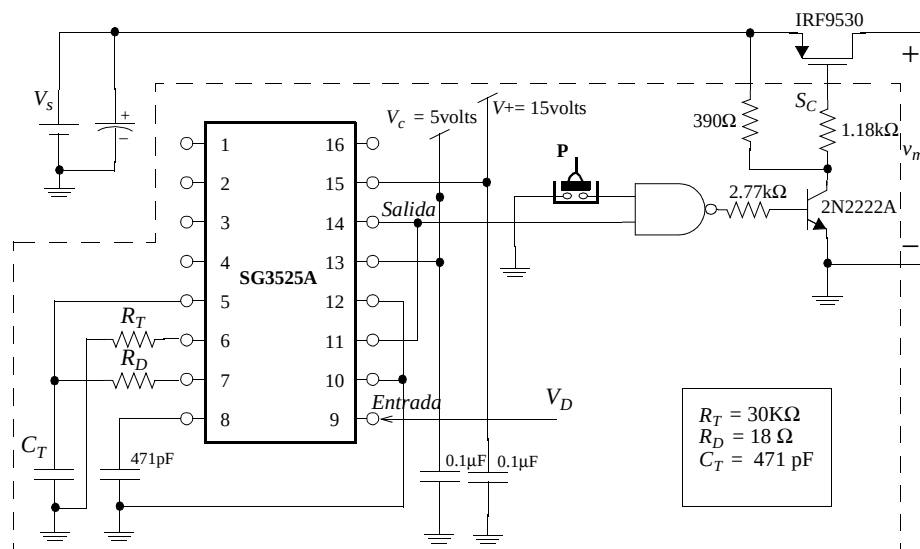


Fig.7.23 Circuito PWM para el control del Convertidor DC-DC

## 3.5.2.6. Bloque de interfaz entre MFCÓN y el circuito generador de PWM

Este bloque transforma la señal de salida en modo corriente  $I_D$  de MFCÓN, cuyo rango nominal va desde 0 a  $150\mu\text{A}$ , en la señal en modo tensión  $V_D$  que se conecta a la entrada del circuito generador de la señal PWM, y cuyo rango de valores va desde 1.0 a 3.3 voltios. Para realizar esta transformación se emplea el circuito de la Fig.7.24. Dado que la entrada a la que se conecta  $V_D$  es de alta impedancia, se verifica que

$$V_D = V_{Dmax} - I_D R_{var} \quad (7.36)$$

De esta forma, cuando  $I_D$  toma su valor mínimo,  $0\mu\text{A}$ , entonces  $V_D = V_{Dmax}$ , mientras que  $R_{var}$  se ajusta de forma que para  $I_D$  máximo,  $150\mu\text{A}$ ,  $V_D$  tome su valor mínimo  $V_D = V_{Dmin}$ .

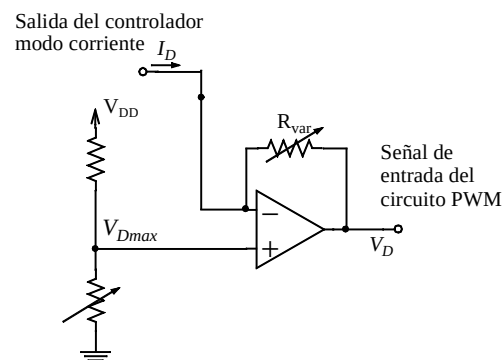


Fig.7.24 Circuito de Interfaz MFCÓN, bloque generador PWM.

#### 4. RESULTADOS DE LOS EXPERIMENTOS DE ARRANQUE SUAVE REALIZADOS CON MFCON EN EL BUCLE DE CONTROL

En este apartado se muestran los resultados de varios ejemplos del control del arranque empleando la estrategia y el sistema experimental descrito en el apartado anterior. En los diferentes ejemplos se exploran distintos objetivos y trayectorias de arranque.

##### 4.1. Determinación experimental de algunas relaciones y características del sistema de control.

Como se ha indicado anteriormente, dado que no se ha dispuesto de la hoja de características del motor de imán permanente, los principales datos necesarios para desarrollar e interpretar los resultados de los ejemplos realizados han sido adquiridos experimentalmente. A continuación se exponen éstos, en términos de las variables que se manejan en los experimentos y que son las que aparecen en el esquema de la Fig.7.18.

La tensión de alimentación  $V_S$  se fija a 20 volts. La resistencia  $R_m$  medida en los terminales del motor es  $7.7\Omega$ . La resistencia  $R_{sen}$  empleada para el sensado de la corriente del motor tiene el valor de  $1\Omega$ .

La Tabla 7.2 resume diferentes valores de las variables experimentales medidos en régimen estacionario, empleados para establecer diversas relaciones.

**Tabla 7.2:** Datos experimentales de las variables del sistema en régimen estacionario.

| $V_D$ (volts) | $D_T$ (%) | $V_\omega$ (volts) | $V_I$ (volts) |
|---------------|-----------|--------------------|---------------|
| < 0.9         | 0         | 2.061              | 2.014         |
| 1.0           | 10.2      | 2.093              | 2.115         |
| 1.1           | 12.7      | 2.204              | 2.122         |
| 1.2           | 15.9      | 2.370              | 2.134         |
| 1.3           | 18.8      | 2.522              | 2.146         |
| 1.7           | 31.2      | 3.133              | 2.180         |
| 2.0           | 40.6      | 3.470              | 2.207         |
| 2.3           | 48.9      | 3.750              | 2.232         |
| 2.7           | 61.8      | 4.080              | 2.265         |
| 3.0           | 71.0      | 4.260              | 2.300         |
| 3.3           | 77.1      | 4.320              | 2.340         |

Estos datos permiten establecer las siguientes relaciones entre las variables de sensado y el ciclo de trabajo  $D_T$

$$V_I = 0.0031D_T + 2.0824 \quad (7.37)$$

$$V_\omega = 0.0345D_T + 1.8797 \quad (7.38)$$

donde  $D_T$  se expresa en unidades de tanto por ciento, mientras que  $V_I$  y  $V_\omega$  se expresan en voltios. Dado el valor de  $V_S$ , a partir de (7.37) y (7.38) se obtiene las siguientes relaciones entre la tensión media aplicada a los terminales del motor  $V_m$  y las variables sensadas

$$V_I = 0.0157V_m + 2.0824 \quad (7.39)$$

$$V_\omega = 0.1727V_m + 1.8797 \quad (7.40)$$

Al no disponer de la relación de conversión  $V_\omega$  a unidades de velocidad de giro  $\Omega_m$ , en los experimentos la velocidad del motor es expresada voltios y en términos de la variable  $V_\omega$ . Por su parte la corriente de armadura en estado estacionario puede ser calculada según la expresión

$$I_m = \frac{V_I^*}{R_{sen}} = \frac{V_I - V_{Imin}}{R_{sen}} \quad (7.41)$$

También se ha determinado experimentalmente la relación entre los valores *singleton* y el ciclo de trabajo. Para ello se ha medido el valor del ciclo de trabajo de la señal de control  $S_C$ , cuando se programa el MFCON para generar superficies planas con cada uno de los valores *singleton*. La Tabla 7.3 muestra el resultado de esa medida.

**Tabla 7.3:** Relación valores *singleton*, ciclo de trabajo

|                  |      |      |      |      |      |      |      |      |
|------------------|------|------|------|------|------|------|------|------|
| <i>Singleton</i> | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
| $D_T$ (%)        | 79.0 | 72.8 | 67.6 | 62.4 | 57.2 | 52.0 | 46.8 | 41.6 |
| <i>Singleton</i> | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
| $D_T$ (%)        | 36.4 | 31.2 | 26.0 | 20.8 | 15.6 | 10.4 | 5.2  | 0    |

Finalmente, como ya se ha visto, la estrategia de control para el motor empleado en los experimentos se define según las expresiones (7.29) a (7.31) y (7.32), que se repiten a continuación.

**SI**  $I_m(t)$  **ES MENOR QUE**  $I_{mmax}$  **ENTONCES**  $D_T = D_{TS}$

**SI**  $I_m(t)$  **ES MAYOR QUE**  $I_{mmax}$  **ENTONCES**  $D_T = D_{Tmin}$

**SI**  $I_m(t)$  **ES IGUAL A**  $I_{mmax}$  **ENTONCES**  $D_T = D_{Tmd}(I_m, \Omega_m)$

$$\text{con } D_{Tmd} = \frac{I_m \cdot R_a}{V_s} + \frac{\Omega_m K_E \Phi}{V_s}.$$

Teniendo en cuenta la expresión (7.41) y suponiendo en primera aproximación una dependencia lineal en la relación entre  $V_\omega^*$  y  $\Omega_m$

$$V_\omega^* = K_\Omega \Omega_m \quad (7.42)$$

las reglas que definen el control pueden reescribirse en términos de las variables empleadas en el sistema experimental tal como sigue.

$$\text{SI } V_I^*(t) \text{ ES MENOR QUE } R_{sen} I_{mmax} \text{ ENTONCES } D_T = D_{TS} \quad (7.43)$$

$$\text{SI } V_I^*(t) \text{ ES MAYOR QUE } R_{sen} I_{mmax} \text{ ENTONCES } D_T = D_{Tmin} \quad (7.44)$$

$$\text{SI } V_I^*(t) \text{ ES IGUAL A } R_{sen} I_{mmax} \text{ ENTONCES}$$

$$D_T = D_{Tmd}(V_I^*, V_\omega^*) \quad (7.45)$$

$$\text{con } D_{Tmd} = \frac{V_I^* \cdot R_m}{R_{sen} V_s} + \frac{V_\omega^* K_E \Phi}{K_\Omega V_s} = K_I V_I^* + K_\omega V_\omega^* \quad (7.46)$$

En la expresión (7.46) la constante  $K_I$  vale  $0.385 \text{ (volts)}^{-1}$ , y se evalúa en términos de valores de elementos conocidos por ser fijados por diseño. Por su parte  $K_\omega$  puede obtenerse a partir de los datos experimentales de la Tabla 7.2, según la expresión

$$K_\omega = \frac{D_T - K_I V_I}{V_\omega} \quad (7.47)$$

obtenida despejando  $K_\omega$  de la expresión (7.46). Tomando el valor medio de los valores de  $K_\omega$  que se obtienen al sustituir en la expresión (7.47) todos los valores de la Tabla 7.2, se establece para esta constante el valor  $0.25 \text{ (volts)}^{-1}$ . Cabe recordar además que  $V_I^* = V_I - V_{Imin}$  y que  $V_\omega^* = V_\omega - V_{\omega min}$ .

En cada uno de los ejemplo de control la matriz de reglas difusas se construye a partir de estas expresiones y los datos de la tablas anteriores. Esas tablas permiten a la vez interpretar las medidas realizadas.

## 4.2. Resultados experimentales

### 4.2.1. Resultados experimentales en arranque directo.

En primer lugar se contempla el caso del arranque directo del motor en el sistema experimental. En este ejemplo el motor es arrancado forzando al sistema PWM a trabajar con un ciclo de trabajo constante de valor  $D_{TS} = 77.1\%$ . La imagen del osciloscopio que aparece en la Fig.7.25 muestra la evolución de las variables  $V_I$  y  $V_\omega$  durante el transitorio de arranque.

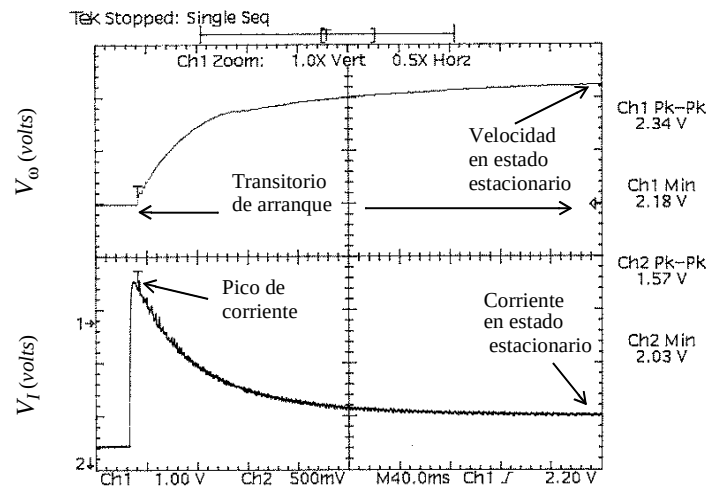


**Fig.7.25 Arranque directo del motor de corriente continua en el sistema experimental**

En ella se aprecia el pico inicial de la variable  $V_I$  de valor  $V_{Ip} = 3.6$  volts., mientras que el valor medido en estado estacionario es  $V_{Iss} = 2.36$  volts., En términos de corriente, según la expresión (7.41) y dado que, según se muestra en la Fig.7.25,  $V_{Imin} = 2.03$  volts., el valor del pico alcanzado es,  $I_{pk} = 1.57$  Amp., y la corriente en estado estacionario  $I_{ss} = 0.33$  Amp, por tanto el valor de pico es 4.75 veces mayor que el valor en estado estacionario. Por su parte, la velocidad en estado estacionario resulta  $V_{oss} = 4.53$  volts. como resulta de aplicar la expresión (7.38). Por último, como una medida de la duración del transitorio de arranque, se ha considerado el parámetro, tiempo de subida ( $tr$ ) medido con el osciloscopio sobre la variable  $V_\omega$  entre el 10% y 90% de su valor final. Para este ejemplo de arranque directo se ha obtenido  $tr = 212$ ms.



## 4.2.2. Arranque suave. Ejemplo I.

En este ejemplo se plantea un arranque suave para alcanzar la velocidad en régimen estacionario del ejemplo anterior, aunque fijando una corriente máxima,  $I_{max}$ , en el motor de 0.80 Amp., lo que supone 2.43 veces su valor en estado estacionario  $I_{ss}$ .

En términos de las variables experimentales estos objetivos se expresan

$$V_{\omega ss} = 4.53 \text{ volts}, V_{I_{max}} = 2.8 \text{ volts}, D_{TS} = 77.1 \%$$

Con estos valores, las reglas de control expresadas en (7.43)-(7.46) resultan

**SI**  $V_I(t)$  **ES MENOR QUE** 2.8 volts **ENTONCES**  $D_T = 77.1\%$

**SI**  $V_I(t)$  **ES MAYOR QUE** 2.8 volts **ENTONCES**  $D_T = 0\%$

**SI**  $V_I(t)$  **ES IGUAL A** 2.8 volts **ENTONCES**  $D_T = 0.385V_I^* + 0.25V_{\omega}^*$

con ellas y los datos de la Tabla 7.3 se construye la matriz de reglas que se muestra en la Fig.7.26.

Los valores que aparecen en esta matriz de reglas corresponden a los valores *singleton* que han de ser programados en MFCON. En la distribución de esos datos se aprecia claramente la aplicación de las dos primeras reglas de control antes mencionadas. La tercera regla da lugar a los valores *singleton* 9,7,5,3,1 que aparecen en la matriz cuando  $V_I = V_{I_{max}}$ , se han obtenido evaluando la expresión de  $D_T$  para los valores de  $V_{\omega}$  que definen los centros de las correspondientes funciones de pertenencia  $X_{\omega 1} - X_{\omega 6}$ . A partir de éstas  $D_T$  alcanza su valor en régimen estacionario,  $D_{TS}$ .

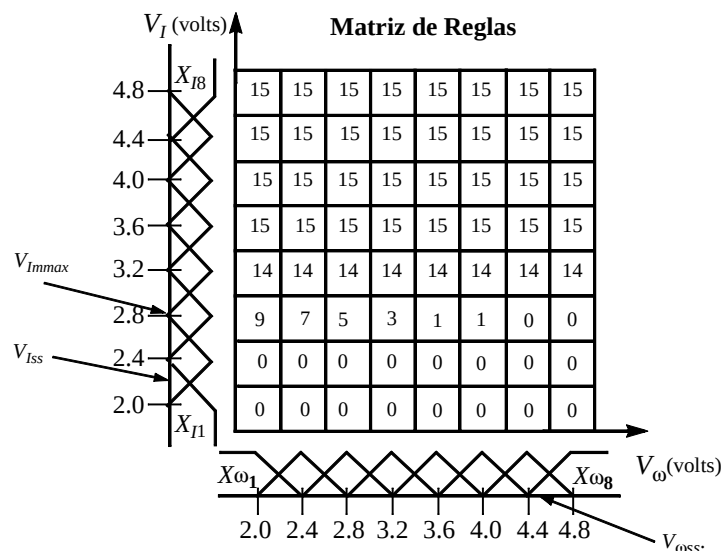


Fig.7.26 Matriz de reglas difusas para el ejemplo de control de arranque I.

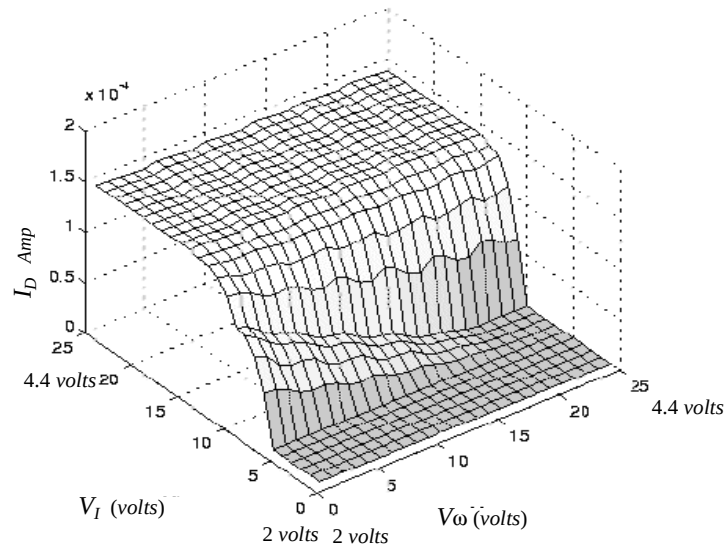
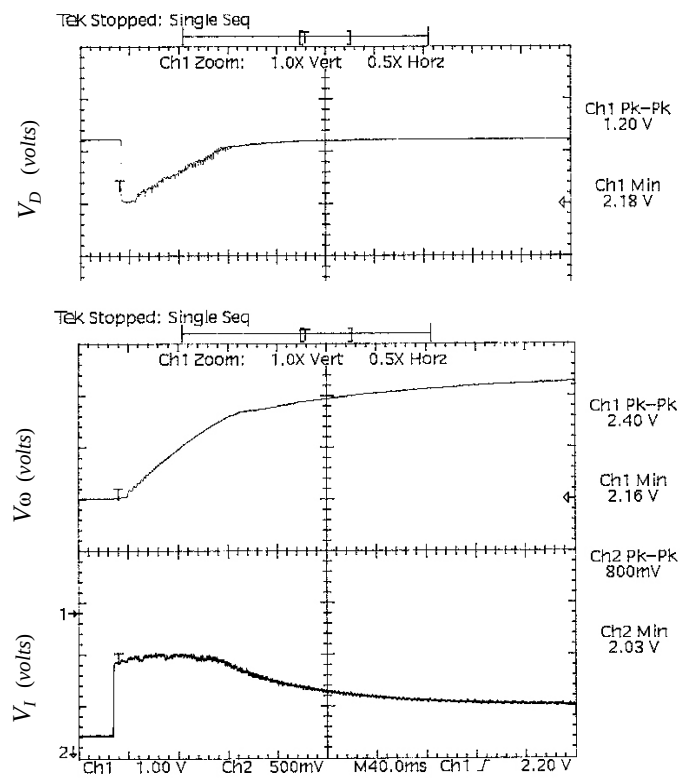


Fig.7.27 Superficie de control generada por MFCON según la matriz de reglas de

Fig.7.28 Arranque suave: ejemplo I: Acción de control  $V_D$  (arriba); velocidad  $V_\omega$  (centro) y corriente  $V_I$  (abajo).

La distribución final de valores *singleton* en la matriz de reglas de la Fig.7.26 es ajustada tras verificar experimentalmente su capacidad de control. La Fig.7.27 muestra la superficie de control generada por MFCON correspondiente a esa matriz de reglas.

La imagen del osciloscopio que aparece en la Fig.7.28 muestra la evolución de las variables de sensado  $V_I$  y  $V_\omega$  y de la acción de control  $V_D$  durante el transitorio de arranque. En ella se aprecia el pico inicial de la variable  $V_I$  de valor  $V_{Ipk} = 2.83$  volts., mientras que el valor medido en estado estacionario es  $V_{Iss} = 2.36$  volts. Expresando estos datos en términos de corriente, según la expresión (7.41) y dado que, según se muestra en la Fig.7.28,  $V_{Imin} = 2.03$  volts., el valor del pico de corriente es,  $I_{pk} = 0.8$  Amp., que corresponde con el valor  $I_{max}$  deseado. La corriente en estado estacionario corresponde a  $I_{ss} = 0.33$  Amp. En este ejemplo se tiene por tanto que el valor de pico es 2.43 veces mayor que el valor en estado estacionario. Por su parte, la medida de la velocidad en estado estacionario resulta ser  $V_{\omega ss} = 4.56$  volts. Por último, el tiempo de subida ( $tr$ ) medido con el osciloscopio sobre la variable  $V_\omega$  resulta ser en este ejemplo  $tr = 253$  ms., tan solo 1.19 veces el tiempo de arranque directo del ejemplo anterior. Este comportamiento es consecuencia de que la acción de control que se ejerce no se limita a impedir que la corriente de armadura sobrepase el valor máximo establecido, como haría un simple circuito limitador de corriente, sino que fuerza a ésta a mantener un valor próximo al máximo, durante el mayor tiempo posible en el transitorio de arranque, a fin de aprovechar el máximo par de arranque seguro y así acelerar el proceso para reducir el tiempo de arranque.

Esta circunstancia se pone de manifiesto si se comparan las gráficas  $V_I$  y  $V_\omega$  de la Fig.7.25 y de la Fig.7.28. Para visualizar el efecto de la acción de control antes descrito, en la Fig.7.29 se han superpuesto las curvas de ambas figuras.

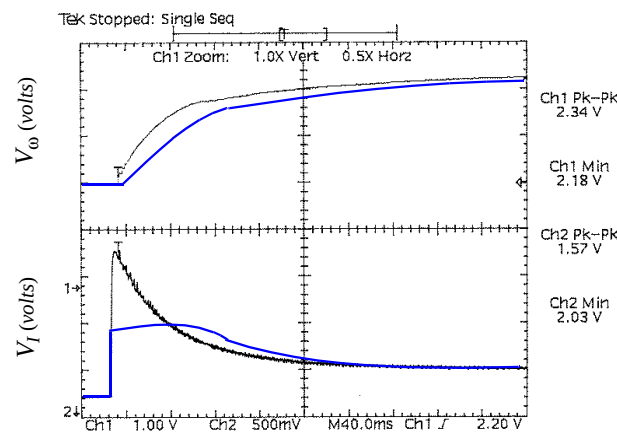


Fig.7.29 Superposición de las curvas  $V_I$  y  $V_\omega$  de las figuras Fig.7.25 y Fig.7.28 para destacar el efecto de la acción de control.

En la parte superior de la Fig.7.28 se muestra la evolución de la acción de control  $V_D$ . Inicialmente, cuando el motor está parado,  $V_D$  toma su valor máximo,  $V_{Dmax} = 3.38$  volts, según las medidas que aparecen en dicha figura. A continuación, y para frenar el rápido ascenso de la corriente,  $V_D$  toma su valor mínimo  $V_{Dmin} = 2.18$  volts., a partir del cual  $V_D$  inicia un suave ascenso hasta alcanzar de nuevo su valor máximo en régimen estacionario.

Los valores de  $V_D$  pueden ser expresados en términos de  $D_T$  a partir de la expresión (7.35). Así se tiene que el ciclo de trabajo aplicado inicialmente es  $D_{Tmax} = 81.0$  % correspondiente al valor *singleton* 0, según la Tabla 7.3; y que el ciclo de trabajo mínimo aplicado es  $D_{Tmin} = 43.9\%$  que corresponde a un valor *singleton* 7, según la Tabla 7.3.

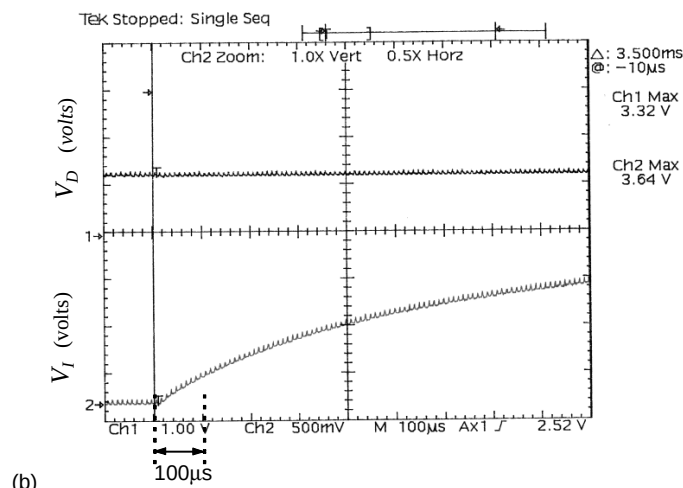
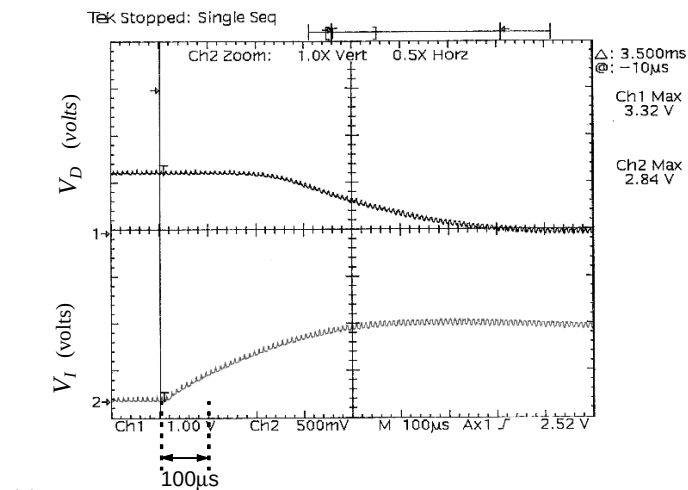


Fig.7.30 Detalle de arranque: Arranque suave ejemplo I (a); arranque directo (b).

A fin de ilustrar la velocidad de respuesta de MFCON y las prestaciones de su acción de control, en la Fig.7.30 se comparan dos detalles del transitorio de arranque, durante los 900 $\mu$ s iniciales, instante durante el cual se produce el pico de intensidad en el ejemplo de arranque directo. Esta circunstancia se pone de manifiesto en la Fig.7.30(b) donde se muestra la evolución de las variables  $V_D$  y  $V_I$ . Como puede apreciarse  $V_D$  permanece constante, puesto que no hay acción de control, mientras que  $V_I$  evoluciona hasta alcanzar su valor de pico. En la Fig.7.30(a) se muestra la evolución de las variables  $V_D$  y  $V_I$  en el ejemplo de arranque I. Como puede apreciarse  $V_D$  comienza con un valor máximo y a medida que  $V_I$  aumenta, la acción de control disminuye el valor de  $V_D$  para forzar a  $V_I$  a permanecer dentro de los límites impuestos.

Este detalle del transitorio de arranque pone de manifiesto la rapidez de respuesta del lazo de control, así como la suavidad de la acción de control que lleva a cabo MFCON. Estas características se deben a que MFCON es capaz de actuar sobre el ciclo de trabajo de la señal PWM, que controla el convertidor, y cuya frecuencia es 100kHz., ciclo a ciclo. Por su parte esta elevada frecuencia de conmutación permite reducir el rizado de la corriente de armadura, como también puede apreciarse en ambas figuras, sin la necesidad de elementos extra.

## 4.2.3. Arranque suave. Ejemplo II.

En este ejemplo se plantea un arranque suave para alcanzar la velocidad en régimen estacionario del ejemplo anterior, aunque fijando una corriente máxima,  $I_{max}$ , en el motor de 0.40 Amp., lo que supone 1.21 veces su valor en estado estacionario  $I_{ss}$ .

En términos de las variables experimentales estos objetivos se expresan

$$V_{\omega ss} = 4.53 \text{ volts}, V_{I_{max}} = 2.4 \text{ volts}, D_{TS} = 77.1 \%$$

Con estos valores, las reglas de control expresadas en (7.43)-(7.46) resultan

**SI**  $V_I(t)$  **ES MENOR QUE** 2.4 volts **ENTONCES**  $D_T = 77.1\%$

**SI**  $V_I(t)$  **ES MAYOR QUE** 2.4 volts **ENTONCES**  $D_T = 0\%$

**SI**  $V_I(t)$  **ES IGUAL A** 2.4 volts **ENTONCES**  $D_T = 0.385V_I^* + 0.25V_{\omega}^*$

con ellas y los datos de la Tabla 7.3 se construye la matriz de reglas que se muestra en la Fig.7.31.

Los valores que aparecen en esta matriz de reglas corresponden a los valores *singleton* que han de ser programados en MFCON. En la distribución de esos datos se aprecia claramente la aplicación de las dos primeras reglas de control antes mencionadas. La tercera regla da lugar a los valores *singleton* 11,9,8,6,2,1 que aparecen en la matriz cuando  $V_I = V_{I_{max}}$ , se han obtenido evaluando la expresión de  $D_T$  para los valores de  $V_{\omega}$  que definen los centros de las correspondientes funciones de pertenencia  $X_{\omega 1}$ - $X_{\omega 6}$ . A partir de ésta  $D_T$  alcanza su valor en régimen estacionario,  $D_{TS}$ .

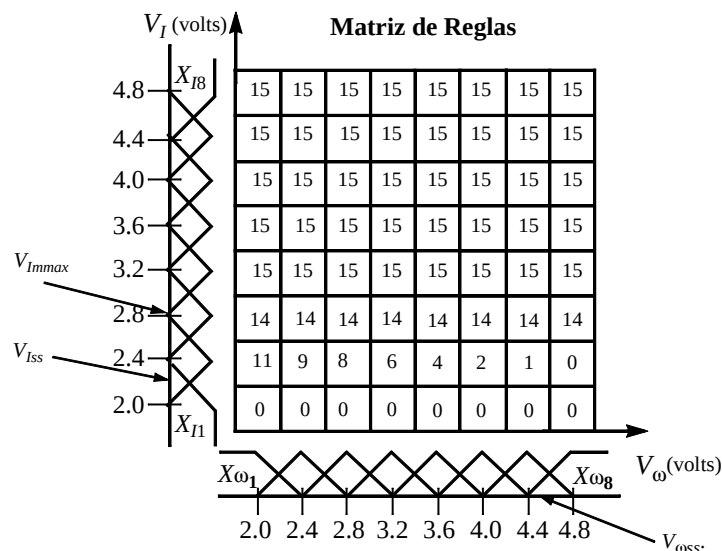


Fig.7.31 Matriz de reglas difusas para el ejemplo de control de arranque II.

La distribución final de valores *singleton* en la matriz de reglas de la Fig.7.31 es reajustada tras verificar experimentalmente su capacidad de control.

La imagen del osciloscopio que aparece en la Fig.7.32 muestra la evolución de las variables de sensado  $V_I$  y  $V_\omega$  y de la acción de control  $V_D$  durante el transitorio de arranque. En ella se aprecia cómo la variable  $V_I$  presenta un perfil prácticamente plano con un valor  $V_{Ipk} = 2.47$  volts., mientras que el valor medido en estado estacionario es  $V_{Iss} = 2.36$  volts. Expresando estos datos en términos de corriente, según la expresión (7.41) y dado que, según se muestra en la Fig.7.32,  $V_{Imin} = 2.03$  volts., el valor del pico de corriente es,  $I_{pk} = 0.44$  Amp., que coincide prácticamente con el valor  $I_{max}$  deseado. La corriente en estado estacionario corresponde a  $I_{ss} = 0.33$  Amp. En este ejemplo se tiene por tanto que el valor de pico es 1.33 veces mayor que el valor en estado estacionario, por lo que el transitorio de arranque se produce prácticamente a par constante. Por su parte, la medida de la velocidad en estado estacionario resulta ser la esperada,  $V_{\omega ss} = 4.52$  volts. Por último, el tiempo de subida ( $tr$ ) medido con el osciloscopio sobre la variable  $V_\omega$  resulta ser en este ejemplo  $tr = 406$  ms., 1.91 veces el tiempo de arranque directo, y 1.58 veces el tiempo de arranque del ejemplo anterior.

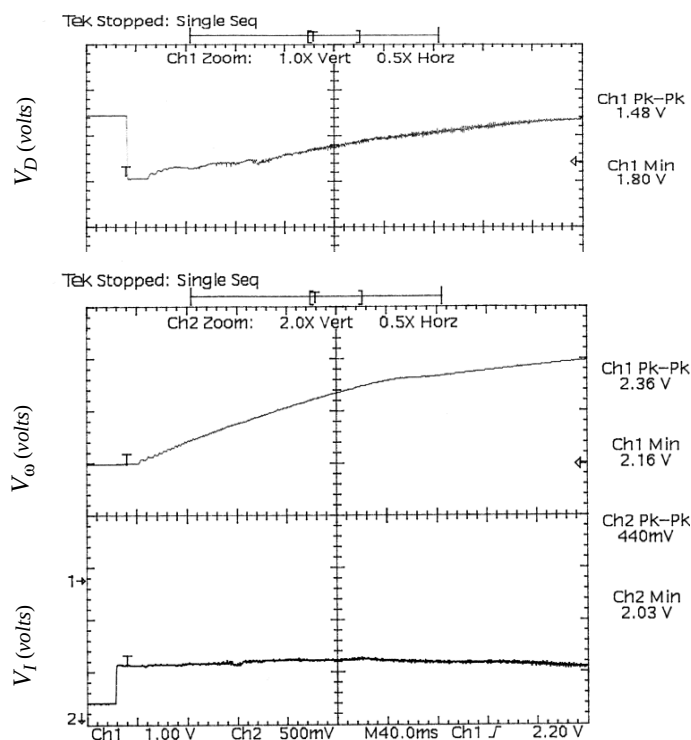


Fig.7.32 Arranque suave: ejemplo II: Acción de control  $V_D$  (arriba); velocidad  $V_\omega$  (centro) y corriente  $V_I$  (abajo).

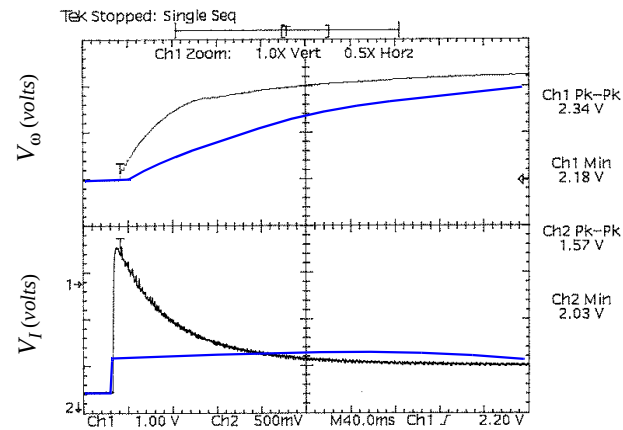


**Fig.7.33 Superposición de las curvas  $V_I$  y  $V_\omega$  de las figuras Fig.7.25 y Fig.7.32 para destacar el efecto de la acción de control.**

En este ejemplo también se pone de manifiesto cómo la acción de control no se limita a impedir que la corriente de armadura sobrepase el valor máximo establecido, sino que fuerza a ésta a mantener un valor próximo al máximo requerido, durante el mayor tiempo posible en el transitorio de arranque, a fin de aprovechar el máximo par de arranque seguro y así acelerar el proceso para reducir el tiempo de arranque. En la Fig.7.33 se visualiza este efecto superponiendo las gráficas de  $V_I$  y  $V_\omega$  de la Fig.7.25 y de la Fig.7.32.

En la parte superior de la Fig.7.32 se muestra la evolución de la acción de control  $V_D$ . Inicialmente, cuando el motor está parado,  $V_D$  toma su valor máximo,  $V_{Dmax} = 3.28$  volts, según las medidas que aparecen en dicha figura. A continuación, y para frenar el rápido ascenso de la corriente,  $V_D$  toma su valor mínimo  $V_{Dmin} = 1.80$  volts. A partir del cual  $V_D$  inicia un suave ascenso hasta alcanzar de nuevo su valor máximo en régimen estacionario.

Los valores de  $V_D$  pueden ser expresados en términos de  $D_T$  a partir de la expresión (7.35). Así se tiene que el ciclo de trabajo aplicado inicialmente es  $D_{Tmax} = 78.0\%$  correspondiente al valor *singleton* 0, según la Tabla 7.3; y que el ciclo de trabajo mínimo aplicado es  $D_{Tmin} = 33.9\%$  que corresponde a un valor *singleton* 9, según la Tabla 7.3.



#### 4.2.4. Resumen de resultados

En la Tabla 7.4 se resumen las principales cifras de los experimentos de arranque descritos en los apartados anteriores.

**Tabla 7.4:** Resumen de resultados de los ejemplos de arranque.

|                  | $I_{ss}$ (Amp) | $I_{max}$ (Amp) | $I_{pk}$ (Amp) | $I_{pk}/I_{ss}$ | $V_{\omega ss}$ (volts) | $tr$ (ms) |
|------------------|----------------|-----------------|----------------|-----------------|-------------------------|-----------|
| Arranque directo | 0.33           | no limitada     | 1.57           | 4.75            | 4.53                    | 212       |
| Ejemplo I        | 0.33           | 0.80            | 0.80           | 2.42            | 4.56                    | 256       |
| Ejemplo II       | 0.33           | 0.40            | 0.44           | 1.33            | 4.52                    | 406       |

Como ya se ha mencionado más arriba, en el Ejemplo I para una reducción del pico de corriente  $I_{pk}$  de aproximadamente un 50%, respecto al caso de arranque directo, el tiempo de arranque se incrementa sólo un 20%. Por su parte en el Ejemplo II se consigue un arranque prácticamente a par constante, y manteniendo la corriente próxima a su valor en régimen estacionario  $I_{ss}$ , con un tiempo de arranque apenas el doble del caso de arranque directo.

Finalmente, cabe destacar la rapidez y la suavidad de la acción de control que lleva a cabo MFCON, como pone de manifiesto el resultado y el comentario a propósito de la Fig.7.30(a).

## Conclusiones

- Esta tesis realiza las siguientes aportaciones:
  - Partiendo de conceptos como los de regla activa y la propiedad de descomposición de los sistemas difusos, se propone una nueva arquitectura para la realización de circuitos integrados en modo mixto de controladores neuro-difusos, que permite incrementar considerablemente la complejidad (estimada en términos de número de entradas y número de reglas) abordable con técnicas puramente analógicas, manteniendo las principales ventajas de estas implementaciones (paralelismo computacional, compacidad y óptima eficiencia en la razón velocidad/consumo de potencia).
  - Se discute la capacidad de aproximación de los controladores realizados siguiendo la arquitectura propuesta, que trabajan sobre particiones en rejilla del universo de discurso, las más empleadas en las realizaciones monolíticas y en aplicaciones de control.
  - Se propone una división del espacio de entrada en intervalos de interpolación. De este modo el conjunto de reglas activas constituye el sistema difuso más simple posible en cada intervalo de interpolación. La nueva arquitectura propone diseñar ese sistema difuso mediante un controlador analógico, denominado Núcleo Analógico, de manera que pueda ser programado dinámicamente, a fin de cubrir todo el espacio de entrada. Para las tareas de identificación del intervalo de interpolación y reconfiguración de dicho controlador, se propone una realización mediante técnicas mixtas analógico-digitales.
  - Se hace una propuesta de diseño de un controlador difuso monolítico para ser fabricado con tecnologías CMOS estándar. Para ello se proponen implementaciones de cada uno de los diferentes bloques funcionales de la arquitectura, y que básicamente se concretan en:
    - Implementación de un núcleo analógico que aprovecha las particularidades del sistema difuso reconfigurable que lo constituye.
    - Implementación de un bloque selector de intervalo, que identifica con una palabra digital a cada intervalo de interpolación.
    - Implementación de un bloque de programación del antecedente de reglas del núcleo analógico.
    - Implementación de un bloque programador del consecuente de reglas, que almacena los valores *singleton* de forma óptima, e intenta preservar el camino de

señal analógico.

- Los anteriores diseños y propuestas se han comprobado mediante cálculos a mano y simulaciones HSPICE, de las que se muestran resultados.
- Se valida la propuesta desarrollada mediante un controlador prototipo diseñado en *full-custom* y fabricado en tecnología CMOS de  $0.7\mu\text{m}$ , con dos capas de metal y una de polisilicio. Este chip posee dos entradas (modo tensión), una salida (modo corriente), implementa sesenta y cuatro reglas, y tiene programabilidad de los valores *singleton* mediante un interfaz compatible con microprocesadores. Su retraso entrada-salida es inferior a 500ns, con un consumo de 16mW y un área ocupada (sin pads) de  $2.65\text{mm}^2$ . Esto lo convierte en el controlador reportado que mejor rendimiento proporciona para su complejidad.
- Se construye un sistema de testado para el controlador que permite obtener diversas superficies de control y medidas de rendimiento.
- Se verifica su funcionamiento desarrollando una aplicación real de arranque suave de motores de corriente continua. Con este ejemplo de aplicación se han alcanzado los siguientes objetivos:
  - Validar el chip fabricado con un ejemplo de aplicación que resulta original en la literatura, y de complejidad adecuada, para mostrar sus cualidades cuando funciona integrado en un bucle de control que necesita una alta velocidad de respuesta.
  - Adquirir experiencia en el campo de la *Electrónica de Potencia* y el *Control de Movimiento*, dirigida a iniciar una línea de trabajo en la búsqueda de aplicaciones de los controladores difusos analógicos y de señal mixta, en las que sus principales cualidades sean explotadas. En este sentido, el ejemplo de aplicación aquí presentado puede ser considerado como un caso particular del problema de control más general que constituye el control de velocidad y limitación de corriente en servomecanismos o en sistemas con motores de velocidad variable.

## Apéndice

### Modelos del transistor y parámetros tecnológicos

Los diseños que se presentan en esta tesis han sido realizados con dos tecnologías CMOS estándar. En este apéndice se dan algunas ecuaciones de primer nivel y el modelo en pequeña señal que se utilizarán en las ecuaciones de diseño y evaluación. Asimismo, se darán los parámetros para estas ecuaciones de las dos tecnologías empleadas y se discutirá la influencia del proceso de fabricación sobre los mismos.

En Fig.A.1(a) se dan los símbolos utilizados del transistor MOS. En los diseños realizados, todos los terminales B de los transistores están conectados a un mismo sustrato, por lo que se utilizará el símbolo del transistor MOS como dispositivo de tres terminales que se muestra a la izquierda de Fig.A.1(a). El símbolo del transistor MOS como dispositivo de cuatro terminales se muestra a la derecha de Fig.A.1(a), y nos sirve de referencia para introducir el modelo en pequeña señal del transistor MOS, que se muestra en Fig.A.1(b) para el transistor NMOS. Los transistores de los circuitos que se presentan en este capítulo trabajan siempre en saturación cuando propagan la señal, siendo la ecuación de primer nivel que liga las magnitudes eléctricas del transistor,

$$I_D = \beta_n (V_{GS} - V_{Tn})^2 (1 + \lambda V_{DS}) \quad (A.1)$$

donde

$$V_{Tn} = V_{T0n} + \gamma [\sqrt{\phi_B + V_{SB}} - \sqrt{\phi_B}] \quad (A.2)$$

En las dos ecuaciones anteriores, los subíndices hacen referencia a los terminales en Fig.A.1(a). Similares ecuaciones rigen el comportamiento del transistor PMOS. En (A.1),  $\beta$  es el factor de transconductancia del transistor, y está relacionado con la geometría del mismo

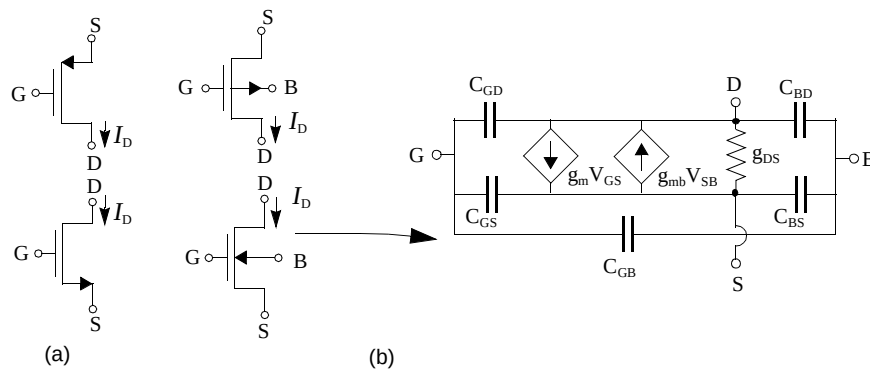


Fig.A.1 Símbolos del transistor MOS (a) ; modelo en pequeña señal (b) .

según la expresión,

$$\beta_n = \beta_{0n} \frac{W}{L} = \frac{\mu_{0n} C_{ox} W}{2} \frac{W}{L} \quad (\text{A.3})$$

donde  $\mu_{0n}$  modela la movilidad superficial del canal,  $C_{ox}$  es la capacidad por unidad de área del óxido de puerta y  $W$  y  $L$  la anchura y longitud efectivas del canal, respectivamente. El parámetro  $\lambda$  modela el cambio de la longitud del canal con la tensión entre el drenador y la fuente en (A.1), y depende de la longitud del canal. Una aproximación común consiste en tomar  $\lambda$  como,

$$\lambda = \frac{a}{L} \quad (\text{A.4})$$

donde  $a$  es una constante dependiente de la tecnología. Por otra parte,  $V_T$  en (A.1) y (A.2) modelan la tensión umbral del dispositivo. Si la fuente y el sustrato están a la misma tensión, el valor de la tensión umbral es  $V_{T0n}$ . Por el contrario, si la tensión fuente-sustrato es distinta de cero, la tensión umbral del dispositivo cambia según (A.2), donde  $\gamma$  es el parámetro de efecto sustrato y  $\phi_B = 2|\phi_F|$ , siendo  $\phi_F$  el potencial superficial en inversión fuerte.

En los diseños que se presentan en este capítulo se ha utilizado una tecnología CMOS n-Well de doble metal y un polisilicio. La longitud mínima del canal es  $0.7\mu\text{m}$  y el espesor del óxido de puerta  $15\text{nm}$ . Los parámetros de esta tecnología a que hacen referencia las ecuaciones anteriores se muestran en la Tabla A.1.

**Tabla A.1:** Parámetros tecnológicos para la tecnología de  $0.7\mu\text{m}$ .

| Parámetro | $\beta_0$                     | $ V_{T0} $              | $ \gamma $                 | $ \phi_B $       | $a$                        |
|-----------|-------------------------------|-------------------------|----------------------------|------------------|----------------------------|
| NMOS      | $62.50\mu\text{A}/\text{V}^2$ | $0.867\text{V}$         | $0.798\text{V}^{1/2}$      | $0.742\text{V}$  | $0.08\mu\text{A}/\text{V}$ |
| PMOS      | $24.85\mu\text{A}/\text{V}^2$ | $1.000\text{V}$         | $0.539\text{V}^{1/2}$      | $0.754\text{V}$  | $0.10\mu\text{A}/\text{V}$ |
| Parámetro | CJ                            | CJSW                    | $C_{ox}$                   | PB               | MJ                         |
| NMOS      | $486\mu\text{F}/\text{m}^2$   | $223\text{pF}/\text{m}$ | $2.30\text{mF}/\text{m}^2$ | $0.675$          | $0.43$                     |
| PMOS      | $600\mu\text{F}/\text{m}^2$   | $820\text{pF}/\text{m}$ | $2.30\text{mF}/\text{m}^2$ | $0.70$           | $0.51$                     |
| Parámetro | MJSW                          | LD                      | $\alpha_a$                 | $\alpha_m$       |                            |
| NMOS      | $0.43$                        | $100\text{nm}$          | $1\mu\text{m}$             | $0.5\mu\text{m}$ |                            |
| PMOS      | $0.51$                        | $80\text{nm}$           | $1\mu\text{m}$             | $0.5\mu\text{m}$ |                            |

Para los cálculos en pequeña señal, se utilizará el modelo en Fig.A.1(b). Sobre este modelo:

$$g_m = \left. \frac{\partial I_D}{\partial V_{GS}} \right|_Q \cong \sqrt{2\beta I_D} \quad (A.5)$$

$$g_{DS} = \left. \frac{\partial I_D}{\partial V_{DS}} \right|_Q \cong \lambda I_D \quad (A.6)$$

$$g_{mb} = \left. \frac{\partial I_D}{\partial V_{SB}} \right|_Q = \left. \frac{\partial I_D}{\partial V_T} \right|_Q \left. \frac{\partial V_T}{\partial V_{SB}} \right|_Q = g_m \delta \quad (A.7)$$

donde

$$\delta = \frac{\gamma}{2\sqrt{\phi_B + V_{SB}}} \quad (A.8)$$

Por otra parte, las capacidades de los condensadores en Fig.A.1(b) tienen los siguientes valores en la región de saturación,

$$C_{GB} \approx 0 \quad C_{GS} \approx C_1 + \frac{2}{3}C_2 \quad C_{GD} \approx C_3 \quad C_{BS} \approx C_{BS}^* + \frac{2}{3}\delta C_2 \quad C_{BD} \approx C_{BD}^* \quad (A.9)$$

con  $C_1$ ,  $C_2$ ,  $C_3$  y  $C_{BX}^*$  dados por,

$$C_1 = C_3 = C_{ox}WL_D \quad (A.10)$$

$$C_2 = C_{ox}WL \quad (A.11)$$

$$C_{BX}^* = \frac{C_JAX}{\left(1 - \frac{V_{BX}}{P_B}\right)^{MJ}} + \frac{C_{JSW}PX}{\left(1 - \frac{V_{BX}}{P_B}\right)^{MJSW}} \quad X = S, D \quad (A.12)$$

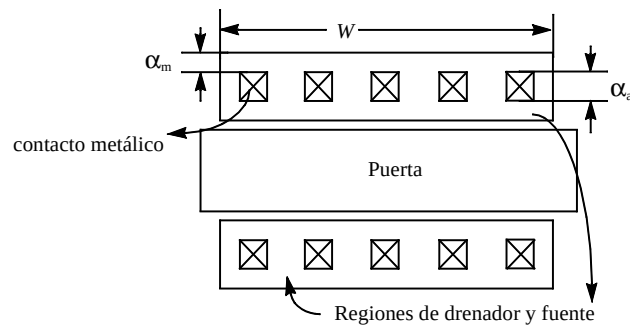


**Fig.A.2** Magnitudes involucradas en el cálculo del área y perímetro de las regiones de difusión de drenador y fuente.

donde  $AX$  y  $PX$  en (A.12)<sup>1†</sup> denotan el área y el perímetro respectivamente de las regiones de unión p-n en la fuente y el drenador, que se pueden aproximar mediante el cálculo del área y el perímetro de las regiones de difusión del drenador y la fuente. Un cálculo sencillo sobre Fig.A.2 da para éstos los siguientes valores,

$$\begin{aligned} AX &= (2\alpha_m + \alpha_a)W \\ PX &= 2W + 2(2\alpha_m + \alpha_a) \end{aligned} \tag{A.13}$$

donde  $\alpha_m$  y  $\alpha_a$  son parámetros que dependen de las reglas de diseño dadas por el fabricante. Los parámetros de las tecnologías empleadas relativos a las ecuaciones anteriores se dan en la Tabla A.1. Una descripción del significado físico de estos parámetros se puede encontrar en [Lake94], y se omitirá aquí.

---

<sup>1†</sup>. En esta ecuación se ha supuesto  $V_{BS} \leq 0$ .

## Referencias

- [Alle87] P.E.Allen, D.R.Holberg, CMOS Analog Circuit Design, Saunders College Publishing, 1987.
- [Asci95] G. Ascia, V. Catania, B. Giacalone, M. Russo, and L. Vita, "Designing for Parallel Fuzzy Computing", *IEEE MICRO*, vol. 15, pp. 62-74, Dec.1995.
- [Asci96] G. Ascia, V. Catania, M. Russo, and L. Vita, "Rule-Driven VLSI Fuzzy Processor", *IEEE MICRO*, vol. 16, pp. 62-74, June 1996.
- [Bali94] B.J. Baliga, "Power Semiconductor Devices for Variable Frequency Drives", *Proceedings of the IEEE*, vol.82, no.8,pp.1112-1122. August 1994
- [Batu97] I. Baturone, S. Sánchez-Solano, A. Barriga and J.L. Huertas, "Implementation of CMOS Fuzzy Controllers as Mixed-Signal Integrated Circuits", *IEEE Tran. on Fuzzy Systems*, vol.5, no.1, Feb.1997.
- [Batu98] I. Baturone, A. Barriga, S. Sánchez-Solano, and J.L. Huertas, "Mixed Signal Design of a Fully Parallel Fuzzy Processor", *Electronics Letter*, vol.34, no.5, pp.437-438, March.1998.
- [Batu98a] I. Baturone, S. Sánchez-Solano, and J.L. Huertas, "Towards the IC Implementation of Adaptive Fuzzy Systems", *IEICE Trans. on Fundamentals of Electronics. Communications and Computer Sciences*, vol.E81-A, no.9, pp.1877-1885, September 1998.
- [Boni95] P.P. Bonissone et al., "Industrial Applications of Fuzzy Logic at General Electric", *Proceedings of the IEEE*, Vol. 83, pp. 450-465, March 1995.
- [Bose87] B.K.Bose, *Microcomputer Control of Power Electronics and Drivers*, New York: IEEE Press, 1987.
- [Bose94] B.K.Bose "Expert Systems, Fuzzy Logic, and Neural Network Applications in Power Electronics and Motion Control". *Proceedings of the IEEE*, vol.82, no.8,pp.1303-1323. August 1994.
- [Bour98] S. Bouras, M. Krotonakis, k. Suyama and Y. Tsividis, Mixed Analog-Digital Fuzzy Logic Controller with Continuous-Amplitude Fuzzy Inferences and Defuzzification", *IEEE Trans. on Fuzzy Systems*, vol.6, no. 2, pp.205-215, May 1998.
- [Bove91] S.Boverie, B. Demaya and A. Titli, "Fuzzy Logic Control Compared with Other Automatic Control Approaches", in *Proc. 30th Conf. Decision Contr.*, Brighthon, U.K., Dec. 1991, pp 1212-1216.
- [Brow94] M. Brown and C. Harris, *Neuro Fuzzy Adaptive Modeling and Control*, Prentice-Hall International, 1994
- [Cost95] A. Costa, A. de Gloria, P. Faraboschi, A. Pagni and G. Rizzotto, "Hardware Solutions for Fuzzy Control". *Proceedings of the IEEE*, Vol. 83, pp. 422-434, March 1995.
- [Cerru92] E. Cerruto, A. Consoli, A. Raciti and A. Testa, "Adaptive Fuzzy Control of High Performance Motion System", *Proceeding IECON'92 Conf.*, pp.88-94, November 1992.
- [Chen95] W.Chen, *The Circuits and Filters Handbook*, CRC Press Inc. 1995.
- [Eich95] H. Eichfeld, M. Klimke, M. Menke, J. Nolles and T. Künemund, "A General-Purpose Fuzzy Inference Processor", *IEEE Micro*, Vol. 15, pp. 12-17, June 1995.
- [El-M96] K. A. El-Metwally, G. C. Hancock, "Implementation of a Fuzzy Logic PSS Using a Micro-controller and Experimental Results", *IEEE Trans. on Energy Conversión*, vol1, no.1 pp.91-96, March 1996.
- [Espe94] S. Espejo Meana, "Redes Neuronales Celulares: Modelado y Diseño Monolítico", Tesis Doctoral, Universidad de Sevilla, Sevilla 1994.
- [Fatt94] J.W. Fattaruso, S.S. Mahant-Shetti, and J.B.Barton, "A Fuzzy Logic Inference Processor", *Journal of Solid-State Circuits*, vol.29. no.4, pp.397-402, April 1994.



- [Fran98] E. Franchi, N. Manaresi, R. Rovatti, A. Bellini, and G. Baccarani, "Analog Synthesis of Nonlinear Functions Based on Fuzzy Logic", *IEEE Journal of Solid-State Circuits*, vol. 33, no. 6, June 1998.
- [Guil96] P. Guillemin, "Fuzzy Logic Applied to Motor Control", *IEEE Trans. on Industry Applications*, vol. 32, no.1, Jan./Feb. 1996.
- [Guar97] J.L. Guardado, K.J. Cornick, V. Venegas, J.L. Naredo, E. Melgoza. "A Three-Phase Model for Surge Distribution Studies in Electrical Machines". *IEEE Trans. on Energy Conversion*, Vol 12, Iss 1, pp.24-31, 1997.
- [Guo96] S.Guo, L. Peters, and Surmann, "Design and Application of an Analog Fuzzy Logic Controller". *IEEE Trans. on Fuzzy Systems*, Vol. 4, pp. 429-438, November 1996.
- [Hara94] F.Harashima, "Power Electronics and Motor Control.- A Future Perspective", *Proceedings of the IEEE*, vol.82, no.8,pp.1107-1111. August 1994.
- [Hill96] R. M. Hilloowala and A. M. Sharaf, "A Rule-Based Fuzzy Logic Controller for a PWM Inverter in a Stand Alone Wind Energy Conversion Scheme", *IEEE Trans. on Industrial Applications*, vol. 32, no.1, pp. 57-65, Jan./Feb.1996.
- [Holt94] J.Holtz. "Pulsewidth Modulation for Electronic Power Conversion", *Proceedings of the IEEE*, vol.82, no.8,pp.1194-1214. August 1994.
- [Huer96] J.L. Huertas, S.Sanchez Solano, I. Baturone, A. Barriga, "Integrated Circuits Implementation of Fuzzy Logic Controllers", *IEEE Journal of Solid-State Circuits*, vol.31, no.7, pp.1051-1058, Jul.1996.
- [Hush93] D. R. Hush and B. G. Horne, "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine*, pp. 8-39, January 1993.
- [Ikeda91] H. Ikeda, Y.Hiramoto, and N.Kisu, "A Fuzzy Inference Processor with an Active-Rule-Driven Architecture", *Symp. on VLSI Circuits*, Oiso, Japan, May 1991.
- [Inge97] M. Ingels and M.S.J.Steyaert, "Design Strategies and Decoupling Techniques for Reducing the Effects of Electrical Interference in Mixed-Mode IC's" *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, July 1997.
- [Jabr92] M. Jabri and B. Flower. "Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks", *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 154-157, Jan. 1992.
- [Jang93a] J.S.R. Jang and C. T. Sun, "Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems", *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, pp. 156-159, January 1993.
- [Jang93b] J.S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. on Syst., Man, Cybern.* vol. 23, no. 3, May/June 1993.
- [Jang95] J.S.R. Jang and C.T.Sun, "Neuro-Fuzzy Modeling and Control", *Proceedings of the IEEE*, vol. 83, pp.378-406, March 1995.
- [Jolly91] R.D.Jolly, "A 9-ns, 1.4-Gigabytes/s, 17-ported CMOS Register File" *IEEE Journal of Solid-State Circuits*, vol. 26, no. 10, pp. 1407-1412, Oct.1991.
- [Kand98] A.Kandel G.Langholz, *Fuzzy Hardware: Architectures and Applications*, Kluwer Academic Publishers 1998.
- [King77] P.J. King and E.H.Mamdani, "The Application of the Fuzzy Control Systems to Industrial Processes", *Automática*, vol.13, no.3, pp. 235-242, 1977.
- [Klir96] G.J. Klir, B. Yuan, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems, Selected Papers by Lofti A. Zadeh*, *Advances in Fuzzy Systems-Applications and Theory*, vol.6, World Scientific 1996.

- [Lake94] K.R.Laker and W.M.C. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, 1994.
- [Le-H94] H. Le-Huy, "Microprocessors and Digital IC's for Motion Control", *Proceedings of the IEEE*, vol.82, no.8, pp.1140-1163. August 1994.
- [Lee90] C.C.Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Control-Part I and II", *IEEE Trans. Syst. Man, Cybern.*, vol. 20, pp. 404-435, Mar./Apr. 1990.
- [Lema94] L. Lemaitre, M. Patyra and D. Mlynek, "Analysis and Design of CMOS Fuzzy Logic Controller in Current Mode", *IEEE Journal of Solid-State Circuits*, vol.29, no.3, pp.317-322, March 1994.
- [Liaw91] C-M Liaw and J-B Wang, "Design and Implementation of Fuzzy Controller for High Performance Induction Motor Drive", *Trans. on System, Man, and Cybernetic*, vol 21, no. 4, pp. 921-929, Jul./Aug. 1991.
- [Lin96] C.T.Lin, C.S. Lee, *Neural Fuzzy Systems*, Prentice-Hall PTR 1996.
- [Lin97] B.R.Lin, "Analysis of Neural and Fuzzy-Power Electronic Control", *IEE Proc.Sci. Meas. Technol.*, vol. 144, no. 1, Jan. 1997.
- [Mamd74] E.H. Mandani, "Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant", *Proceeding of The IEE*, vol.121, pp. 1585-1588, 1974.
- [Mana96] N. Manaresi, R. Rovatti, E. Franchi, R. Guerrieri, and G. Baccarani, "A Silicon Compiler of Analog Fuzzy Controllers: From Behavioral Specification to Layout", *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 4, Nov. 1996.
- [Mark94] R.J. Marks II (ed.): *"Fuzzy Logic Technologies and Applications"*. New-York: IEEE Press 1994.
- [Mase95] M. Masetti, E. Gandolfi, A. Gabrieli and F. Boschetti, "4 Input VLSI Fuzzy Chip Design able to process an Input Data Set every 320ns", *Proceedings of the JCIS'95*, Wrightsville Beach, North Carolina, USA, October 1995.
- [Mend95] J.M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial". *Proceedings of the IEEE*, vol.83, pp. 345-377, March 1995.
- [Miki93] T. Miki, H. Matsumoto, K. Ohto and T. Yamakawa, "Silicon implementation for a Novel High-Speed Fuzzy Inference Engine: Mega-Flips Analog Fuzzy Processor", *Journal of Intelligent and Fuzzy Systems*, vol.1 no. 1 pp 27-42, 1993.
- [Miki95] T. Miki, T. Yamakawa, "Fuzzy Inference on an Analog Fuzzy Chip", *IEEE Micro*, vol. 15, no. 4, pp. 58-66, 1995.
- [Moha95] Mohan, Underland, Robbins, *Power Electronics Converters, Applications, and Design*, Ed. Wiley 1995.
- [Nava97] R.Navas-González, F. Vidal-Verdú y A. Rodríguez Vázquez "Multiplexing Strategy for Mixed Signal CMOS Fuzzy Controllers" *Proc. of the XII Design of Circuits and Integrated Systems Conference*, pp. 401-405, Seville (Spain), Nov.1997.
- [Nava99] R. Navas-González, F. Vidal-Verdu, A. Rodriguez-Vazquez, "A Mixed-Signal Architecture for High Complexity CMOS Neuro-Fuzzy Controllers", *Mathware & Soft Computing*, vol VI, no.2-3, pp. 331-343, 1999.
- [Nava00] R. Navas-González, F. Vidal-Verdu, A. Rodriguez-Vazquez, "A Mixed-Signal Fuzzy Controller and its Application to Soft Start of DC Motors", *Ninth IEEE International Conference on Fuzzy Systems*, vol. 1, pp.128-133, San Antonio, Texas, US May 2000.
- [Nava00a] R. Navas-González, F. Vidal-Verdu, A. Rodriguez-Vazquez "Soft Start of DC Motors with a Mixed-Signal Fuzzy Controller", *Eigth International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems*, vol. 1, pp.30-35, Madrid July 2000.

- [Nie93] J. Nie and D. A. Linkens. "Learning Control Using Fuzzified Self-Organizing Radial Basis Function Network", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 4, pp. 280-287, November 1993.
- [Nie95] J. Nie & D. Linkens, *Fuzzy-Neural Control*, Prentice-Hall 1995.
- [Nish93] K.A. Nishimura, "Optimum Partitioning of Analog Mixed-Signal Circuits for Signal Processing". Memorandum No. ECB/ERL M93/67, University of California at Berkeley. 1993.
- [Ong 98] Ch. ONG, *Dinamic Simulation of Electric Machinery Using MATLAB®/SIMULINK*, Prentice-Hall inc. 1998.
- [Pagn93] A. Pagni, R. Poluzzi, G.Rizzoto, M. Lo Presti, "DC/DC Converters Fuzzy Control", pp.14-17,1993
- [Pall93] R. Pallás Areny, "Adquisición y Distribución de Señales", Ed. Marcombo.Boixareus, 1993.
- [Pelg89] M.J.M. Pelgrom et al. "Matching Properties of MOS Transistor", *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1433-1440, June 1989.
- [Peri94] L. Perier "Microcontrollers in Home Appliance: A soft Revolution", *SGS-THOMSON Application Note AN674*, 1994.
- [Perk92] J. E. Perkins, I. M. Y. Mareels and J. B. Moore, "Functional Learning in Signal Processing Via Least Squares", *International Journal of Adaptive Control and Signal Processing*, Vol. 6, pp. 481-498, 1992.
- [Pete98] L. Peters and S.Guo, "Tree Generations on Fuzzy Hardware" in *Fuzzy Hardware: Architectures and Applications*, A.Kandel G.Langholz (Eds.), Kluwer Academic Publishers 1998.
- [Rash93] M.H.Rashid, *Power Electronics, Cicuits, Devices, and Applications*, 2ed. Prentice-Hall International Editions 1993.
- [Rash96] M. H. Rashid, *Recent Developments in Power Electronics, Selected Readings*, IEEE 1996
- [Rodr99] A. Rodríguez-Vázquez, R.Navas, M.Delgado-Restituto, and F. Vidal-Verdú, "A Modular Programmable CMOS Analog Fuzzy Controller Chip", *IEEE Trans. on Circ. & Syst. II* vol. 46, no.3, pp. 251-265, March 1999.
- [Rodr99a] A.Rodríguez\_Vázquez et al. *Encyclopedia of Electrical and Electronics Engineering*, Wiley 1999.
- [Russ98] M. Russo, "Fuzzy Hardware Research From Historical Point of View" in *Fuzzy Hardware: Architectures and Applications*, A.Kandel G.Langholz (Eds.), Kluwer Academic Publishers 1998.
- [Sasa92] M. Sasaki et al.: "Current-Mode Analog Fuzzy Hardware with Voltage Input Interface and Normalization Locked Loop", *IEICE Trans. Fundamentals*, vol. E75-A, pp. 650-654, June 1992.
- [Shep95] W. Shepherd, L.N. Hulley and D.T. WLiag, *Power Electronics and Motor Control*, Cambridge University Press.1995.
- [Shin91] H.Shinohara, N. Matsumoto,K.Fujimori,Y.Tsujihashi,H.Nakao,S.Kato,Y.Horiba, and A.Tada "A Flexible Multiport RAM Compiler for Data Path", *IEEE Journal of Solid-State Circuits*, vol.26, no.3, pp. 343-349, March1991.
- [Spec91] D.F. Specht, "A General Regression Neural Networks", *IEEE Transactions on Neural Networks*, vol.2, no. 6, November 1991.
- [Sous93] G.C.D. Sousa, B.K. Bose, et al. "Fuzzy Logic Based on-line Efficiency Optimization of an Indirect Vector Controlled Induction Motor Drive", *.IEEE/IECON Conf. Procc.*, pp. 1168-1174, 1993.
- [Sous94] G.C.D. Sousa, B.K. Bose, "A Fuzzy Set Theory Based Control of a Phase-Controlled Converter DC Machine Drive", *Trans. on Industrial Applications*, vol. 30, pp. 34-44, Jan./Feb. 1994.
- [Suyi93] A. Suyitno, J. Fujikawa, H. Kobayashi, and Y. Dote, "Variable-Structured Robust Controller by Fuzzy Logic for Servomotors", *IEEE Trans. on Industrial Electronics*, vol.40, no.1, pp.80-88, Feb.1993.

- [Taka89] T.Takagi and Sugeno, "Derivation of Fuzzy Control Rules from Human Operator's Control Action", Proc. of the IFAC Symp. of Fuzzy Inf., Knowledge Representation and Decision Analysis, pp. 55-60, July 1989.
- [Taka94] H.Takagi, "Applications of Neural Network and Fuzzy Logic to Consumers Products", pp. 8-12 in *Fuzzy Logic Technologies and Applications*, New-York, IEEE Press 1994.
- [Ugur97] Ugur. Çilingiroglu, Banu Pamir, Z. Sezgin and F.Dülger, "Sampled-Analog Implementation of Application-Specific Fuzzy Controllers", *IEEE Trans. on Fuzzy Systems*, vol.5,no.3 pp.431-442, August 1997.
- [Vida95a] F. Vidal-Verdú and A. Rodríguez-Vázquez, "Using Building Blocks to Design Analog Neuro-Fuzzy Controllers " *IEEE Micro*, vol. 15, no. 14, pp. 49-57, August 1995.
- [Vida95b] F. Vidal-Verdú and A. Rodríguez-Vázquez, "Learning under hardware restrictions in CMOS Fuzzy Controllers Able to Extract Rules from Examples", *Proc. IFSA'95*, pp 189-192, Sao Paulo Brazil, July 1995.
- [Vida96a] F. Vidal-Verdú, R. Navas and A. Rodríguez-Vázquez, "A 16Rules@2.5 Mflips Mixed Signal Programmable Fuzzy Controller CMOS-1:μ Chip" *Proc. of the ESSCIR'96*, pp. 156-159, Sep.1996.
- [Vida96b] F.Vidal. "Una Aproximación al Diseño de Controladores Neuro-Difusos Usando Circuitos Integrados Analógicos", *Tesis Doctoral*. Universidad de Málaga 1996.
- [Vida98] F. Vidal, M. Delgado-Restituto, R. Navas and A. Rodríguez-Vázquez, "A Building Block Approach to the Design of Analog Neuro/Fuzzy Systems in CMOS Digital Technologies", in *Fuzzy Hardware: Architectures and Applications*. Editado por A. Kandel y G. Langholz, Kluwer Academic Publishers 1997.
- [Vida98a] F. Vidal-Verdú, R. Navas-González and A. Rodríguez-Vázquez, "Multiplexing architecture for mixed-signal CMOS fuzzy controllers", *Electronics Letters*, vol.34 no.14 pp. 1437-1438, July 1998.
- [Vida00] F. Vidal-Verdú, R. Navas-González, M. Delgado-Restituto and A. Rodríguez-Vázquez, "A Stored-Programmable Mixed-Signal Fuzzy Controller Chip with Supervised Learning Capabilities", in *Fuzzy Learning and Applications*, pp.51-92, Ed. M Russo, CRC Press, a publicar en el año 2000.
- [Vitt90] E. A. Vittoz, "The Future of Analog in VLSI Enviroment", *Procc. of the 1990 IEEE Int. Symp on Circuits and Systems*, pp. 1372-1375, 1990.
- [Wang90] Z. Wang, "Current-Mode Analog Integrated Circuits and Linearization Techniques in CMOS Technology", Ed. Hartung-Gorre and verlag Konstanz, Zürich 1990.
- [Wang92] L.X.Wang, "Fuzzy Systems are Universal Aproximators", *Proc. IEEE International Conf. on Fuzzy Systems*, San Diego, pp. 1163-1170, 1992.
- [Wang97] L.X.Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall PTR 1997.
- [Wata90] H.Watanabe, W.D.Detloff, and K.E.Yount, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture", *Journal of Solid State Circuits*, vol. 25, pp. 376-382, Apr. 1990.
- [Wata92] H.Watanabe, "RISC approach to design of fuzzy proccesor architecture", in *Procc. IEEE Int. Conf. Fuzzy Systems*, San Diego, CA Mar 1992.
- [West94] N.H.E.Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley 1994.
- [Wild97] T. Wildi, *Electrical Machines, Drivers, and Power Systems*, 3° ed. Prentice-Hall inc. 1997.
- [Wyk94] J.D. van Wyk, "Power Electric Converters for Motion Control", *Proceedings of the IEEE*, vol.82, no.8,pp.1164-1193. August 1994.
- [Yama80] T. Yamakawa et al. "Implementacion od Fuzzy Logic Hardware Systems", *Trans. IECE*, vol.63-C, no. 10., pp. 720.721, 1980.
- [Yama86] T. Yamakawa and T. Miki, "The Current Mode Fuzzy Logic Integrated Circuits Fabricated by the

- Standard CMOS Process", *IEEE Trans. on Computer*, vol C-35, no. 2, pp. 460-476, 1986.
- [Yama93] T. Yamakawa, "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control", *IEEE Trans. on Neural Networks*, vol.4, no.3, pp. 496-522, May 1993.
- [Ying90] H.Ying, W.Siler, and J.J.Buckley, "Fuzzy Control Theory: a Nonlinear Case", *Automatica*, vol. 26, pp.513-520, May 1990.
- [Ying93] H.Ying, "The simplest fuzzy controlles using different inference methods are different nonlinear proportional integral controllers with variable gains", *Automatica*, vol. 29 pp.1579-1589, Nov. 1993.
- [Ying94] H. Ying, "Sufficient conditions on general fuzzy systems as function approximations", *Automatica*, vol.30, no. 3, pp. 521-525, 1994.
- [Yuan92] F. Yuan, L.A. Feldkamp, L.I. Davis and G. V. Puskorius, "Training a Hybrid Neural-Fuzzy System", *International Joint Conference on Neural Networks*, vol. 4, pp. 739-744, Baltimor, Maryland, June 1992.
- [Zade65] L.A.Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.
- [Zade73] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" *IEEE Trans. Syst., Man, Cybern.*, vol SMC-3, no 1, pp. 28-44, Jan. 1973.
- [Zeng94] X.J.Zeng and M.G.Singh, "Aproximation Theory of Fuzzy Systems-SISO Case", *IEEE Tran. on Fuzzy Systems*, vol. 2, no. 2, pp. 162-176, May 1994.
- [Zeng95] X.J.Zeng and M.G.Singh, "Aproximation Theory of Fuzzy Systems-MIMO Case", *IEEE Tran. on Fuzzy Systems*, vol. 3, no. 2, pp. 219-235, May 1995.
- [Zeng96a] X.J.Zeng and M.G.Singh, "Aproximation Accuracy Analysis of Fuzzy Systems as Function Approximators", *IEEE Tran. on Fuzzy Systems*, vol. 4, no. 1, pp. 44-63, Feb.1996.
- [Zeng96b] X.J.Zeng and M.G.Singh, "Decomposition Property of Fuzzy Systems and Its Applications", *IEEE Tran. on Fuzzy Systems*, vol. 4, no. 2, pp. 149-165, May1996
- [Zura92] J. M. Zurada, "Introduction to Artificial Neural Systems". West Publishing 1992.
- [NS-ADC] National Semiconductor ADC0801/ADC0802/ADC0803/ADC0804 8-Bits mP Compatible A/D Converters. Data Sheet. December 1994.
- [NS-DAC] National Semiconductor DAC0800/DAC0801/DACC0803/ADC0802 8-Bit Digital to Analog Converters. Data Sheet. January 1995.