



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERIA

**DISEÑO DEL CONTROLADOR DE VELOCIDAD
DE UN MOTOR DE DC UTILIZANDO
LOGICA DIFUSA**

T E S I S

**QUE PARA OBTENER EL TITULO DE
INGENIERO MECÁNICO ELECTRICISTA
P R E S E N T A :**

MARTIN ARNULFO GOMEZ MARIN



ASESOR: M.I. JORGE VALERIANO ASSEM

MEXICO, D.F.

2007



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos:

A Dios:

Por su amor y paciencia para conmigo, por permitirme ser y estar más cerca de Él.

A mi Madre:

Evangelina, por su dedicación, apoyo y comprensión desde siempre y en todos los momentos de mi vida.

A mi Padre:

Arnulfo, que en compañía de Dios esta, quien me enseñó el sentido de responsabilidad y deber.

A mi Hijo:

Jesús Martín Elí, a quien amo profundamente, por ser él, motivo y razón para concluir esta etapa.

A mis Hermanos:

Elda, Alejandro, Antonio, Erick y Evangelina, por su ejemplo y motivación.

A la Universidad Nacional Autónoma de México.

A la Facultad de Ingeniería:

Por la formación que me dio. Por ser el espacio donde conocí la ciencia y la expresión de nuestra cultura.

A mi Asesor:

M.I. Jorge Valeriano, por su apoyo, tiempo y consejos para guiarme en la realización de mi tesis.

ÍNDICE

INTRODUCCIÓN.

CAPÍTULO 1. LÓGICA DIFUSA.....	1
1.1 Historia de la Lógica Difusa.....	1
1.2 Definiciones y conceptos básicos.....	3
1.2.1 Variables lingüísticas.....	5
1.2.2 Conjuntos difusos.....	7
1.2.3 Operaciones básicas entre conjuntos difusos.....	10
1.2.4 Propiedades de los conjuntos difusos.....	13
1.2.5 Características de la función de membresía.....	15
1.2.6 Relaciones difusas.....	17
1.2.7 Reglas difusas.....	19
1.2.8 Desdifusión.....	22
1.2.9 Aplicación de control con Lógica Difusa.....	25
1.3 Diferencia entre la Lógica Difusa y los métodos convencionales de control.....	26
1.4 Ventajas de la utilización de la Lógica Difusa sobre los métodos convencionales de control.....	28
1.5 El uso de la Lógica Difusa.....	33
1.5.1 Ciclo de análisis y diseño usando Lógica Difusa.....	34
1.5.2 Sistema de decodificación, concisión o desdifusión.....	34
1.5.3 Ejemplo: El péndulo invertido.....	35
CAPÍTULO 2. MOTORES DE CORRIENTE DIRECTA.....	41
2.1 Conceptos básicos de los motores de CD.....	42
2.1.1 Relación entre el par y la potencia.....	43
2.1.2 Producción del par.....	44
2.1.3 Fuerza contraelectromotriz en un motor.....	45
2.1.4 Potencia desarrollada por la armadura.....	46
2.1.5 Velocidad de equilibrio del motor.....	47
2.1.6 Tipo de excitación en motores de CD.....	48
2.2 Control de velocidad en motores de CD.....	49
2.2.1 Causas básicas y límites de velocidad.....	49
2.2.2 Mecanismos que se usan en el control de velocidad.....	51
2.2.3 Control de velocidad de estado sólido en motores de CD.....	51
CAPÍTULO 3. SOFTWARE PARA SISTEMAS DIFUSOS.....	54
3.1 FIDE, Fuzzy Inference Development Environment (herramienta de desarrollo de Inferencia difusa).....	54
3.2 FuzzyTech.....	59
3.3 MATLAB.....	63
3.3.1 Herramienta de lógica difusa.....	63
3.3.2 Proceso de Inferencia Difusa.....	64
3.3.3 Ejemplo de lógica difusa usando MATLAB.....	66

CAPÍTULO 4. DISEÑO DEL CONTROLADOR DE VELOCIDAD DEL MOTOR DE CD.....	75
4.1 Acciones de control.....	75
4.1.1 Acción de control de dos posiciones, encendido y apagado (on-off).....	75
4.1.2 Acción de control proporcional.....	76
4.1.3 Acción de control integral.....	76
4.1.4 Acción de control proporcional-integral.....	76
4.1.5 Acción de control proporcional-derivativa.....	77
4.1.6 Acción de control proporcional-integral-derivativa.....	77
4.2 El control difuso.....	78
4.2.1 Difusión.....	78
4.2.2 Reglas difusas.....	79
4.2.3 Inferencia difusa.....	79
4.2.4 Desdifusión.....	80
4.3 Controlador Mamdani difuso PID, PD y PI.....	80
4.4 Simulación del sistema de control difuso.....	82
4.4.1 Controlador del motor usando SimPowerSystem, (Simulador de sistemas de potencia) de MATLAB.....	82
4.4.2 El controlador discreto PI.....	85
4.4.3 Voltaje promedio $V(\text{avg})$ y corriente promedio $I(\text{avg})$	88
4.4.4 El controlador difuso PI Mamdani.....	91
4.4.5 Primer controlador (49 reglas difusas).....	92
4.4.6 Segundo controlador (4 reglas difusas), conjuntos difusos de salida triangulares.....	99
4.4.7 Tercer controlador (4 reglas difusas), conjuntos difusos de salida unitarios (singleton).....	106
4.4.8 Cuarto controlador (4 reglas difusas), conjuntos difusos de salida trapezoidales.....	112
CONCLUSIONES.....	116
APÉNDICE. Programación de un bloque controlador de lógica difusa.....	125
BIBLIOGRAFÍA.....	131

INTRODUCCIÓN

En los procesos de control se ve el control difuso como una forma de control “inteligente” y es considerado con ventajas en algunos aspectos con respecto al control clásico. Para aplicar el control clásico se requiere del modelo matemático del sistema, por el contrario, el control difuso no necesita un modelo explícito del sistema. El modelo del sistema es implícito en la construcción de las reglas difusas, las operaciones difusas y los conjuntos difusos.

La intención es aplicar el control difuso para controlar la velocidad de un motor de corriente directa. Para ello haremos uso del programa MATLAB, el cual nos permite la simulación del motor y los componentes eléctricos y electrónicos interactuando con la herramienta de lógica difusa donde se diseñara el controlador difuso.

En el Primer capítulo, Lógica difusa, hablamos de su historia, las definiciones y conceptos básicos de las operaciones con lógica difusa, los métodos de desdifusión y las ventajas de la aplicación de la lógica difusa en el control de sistemas con respecto al control clásico.

El Segundo capítulo, Motores de corriente directa, esta enfocado a los conceptos y consideraciones necesarias de este tipo de motores, que nos permitan comprender su funcionamiento y las variables que debemos considerar, de acuerdo al tipo de excitación, con el fin de controlar su velocidad aun cuando tengamos variaciones en cuanto a la velocidad y la carga.

En el Tercer capítulo, Software para Sistemas Difusos, damos información de algunos paquetes de software que se usan en el diseño de sistemas difusos, hacemos referencia a FIDE, FuzzyTech y MATLAB. Los dos primeros son paquetes que suponen grandes ventajas, pero que solo algunos centros de investigación los tienen debido a su alto costo y dificultad para obtenerlos. En cambio MATLAB es una paquetería que es de bajo costo y cumple con los requerimientos necesarios para nuestro objetivo. En este capítulo mencionamos las características de MATLAB enfocadas a nuestra aplicación, el proceso de Inferencia Difusa y el enlace con su herramienta de simulación.

En el Cuarto capítulo, Diseño del controlador de velocidad del motor de CD, vemos las acciones de control clásico así como su relación con la lógica difusa para aplicar el control difuso. Usamos un controlador Mamdani PI (Proporcional Integral) para controlar la velocidad del motor de corriente directa. Diseñamos varios bloques difusos en el sistema de control que nos permiten ver en la simulación la respuesta del motor en cuanto a la velocidad y su comportamiento cuando varía la carga del motor.

CAPÍTULO 1. LÓGICA DIFUSA

1.1 Historia de la Lógica Difusa.

El cerebro no razona como lo hace una computadora. La computadora razona en términos absolutos, es blanco o negro, estos signos son 0s y 1s. Nosotros razonamos en términos vagos de acuerdo al sentido común, como “El aire está frío” o “La velocidad es rápida”. Estos hechos difusos son verdad solo en algún grado entre 0 y 1, y son falsos en cierto grado. El cerebro trabaja con estos patrones difusos con facilidad y las computadoras no. La lógica difusa y la ingeniería tratan de cambiar esto.

La idea clave de situaciones difusas viene de la lógica multivaluada de 1920s: *Todo es materia de graduar*. Una declaración de un hecho como “El cielo es azul” o “El ángulo es pequeño” no tiene un valor binario verdadero. Esto tiene un vago o difuso valor verdadero entre 0 y 1. Esto también tiene su negación, “El cielo no es azul”. Por lo tanto el cielo es ambos, azul y no azul en cierto grado. Este simple punto del hecho viola las principales leyes de la lógica, tan extendidas desde la primera lógica formal de los antiguos griegos hasta la fundación de las modernas matemáticas y ciencias.

La ciencia moderna describe esto con planteamientos y ecuaciones binarias, lo mejor que podemos hacer es llevarlo a unos pocos lugares decimales. No podemos llevarlo esto a un infinito decimal que la verdad binaria requiera.

La lógica difusa comienza como situaciones vagas. El filósofo pragmático Charles Sanders Peirce presentó la primera lógica para tratar con las vaguedades [Pierce, C. S., *Collected Papers of Charles Sanders Peirce*, C. Hartshorne and P. Weiss, editors, Harvard University Press, 1931]: “Vaguedad no es mas que el ser de un hecho contenido en el mundo de la lógica como la fricción es a la mecánica”. Un concepto es vago solo en el caso de que sus límites sean borrosos. El concepto montaña es vago porque no sabemos donde termina una montaña y comienza una colina.

El lógico Bertrand Russell en 1923 fue el primero en identificar vaguedad como nivel de lógica simbólica [Russell, B., “*Vagueness*”, *Australian Journal of Philosophy*, vol.1, 1923]. El concepto A es vago si y solo si este rompe la ley de Aristóteles de la exclusión del punto medio, esto si y solo si A y NO-A fallan a este dominio. Enunciados de lógica o matemáticas obedecen las leyes de Aristóteles: $1+1=2$ es 100% verdad y 0% falso. $1+1=3$ es 0% verdad y 100% falso. Pero enunciados de hechos son vagos y tienen valores de verdad entre los extremos binarios. Russell revisó la paradoja del filósofo griego Zenón donde demostró que la inducción no puede ser binaria: “¿Soy calvo?, No. Arranco unos cabellos. ¿Soy ahora calvo?, No. Puedes seguir quitando y preguntando. Por lo tanto se es calvo hasta que arranques todos los mas de 100 000 cabellos de tu cabeza. Cada cabello quitado ligeramente incrementa el grado de calvicie y decrementa la fortaleza de un cambio inductivo [Gaines, B. R., “*Foundations of Fuzzy Reasoning*,” *International Journal of Man-Machine Studies*, vol. 8, 623-688, 1976]. La vaga y final inferencia de no calvicie tienen

valores de verdad cerca del cero, estos provienen del largo número de valores entre 0 y 1.

Russell encontró la respuesta a paradojas en trabajos matemáticos que realizó junto con Alfred North Whitehead en los volúmenes "Principia Mathematica" [Whitehead, A. N., and Russell, B., *Principia Matemática*, vol. I, University of Cambridge Press, 1910]. Él trató la antigua paradoja del mentiroso de Creta. El Cretense dice que todos los Cretenses mienten. ¿Entonces él miente o dice la verdad? Si él miente entonces dice la verdad y por lo tanto no miente. Si él no esta mintiendo entonces él dice la verdad y por lo tanto miente. Ambos casos conllevan a la contradicción A y NO-A.

Russell encontró la misma paradoja en un conjunto de teorías. El conjunto de todos los conjuntos es un conjunto. Por lo tanto éste es un miembro de él mismo. Pero muchos conjuntos no son miembros de ellos mismos. El conjunto de manzanas no es un miembro de él mismo debido a que sus miembros son manzanas y no conjuntos. Pero, ¿Qué, acerca del conjunto de todos los conjuntos que no son miembros de ellos mismos?, ¿Es éste un miembro de él mismo? Si éste es entonces no es. Y si éste no es entonces es. Aquí A y NO-A no pertenecen al dominio del claro-oscuro dominio de los pensamientos sino al sistema formal de las matemáticas binarias.

Russell puso como principio su *teoría de los tipos* para rechazar tales paradojas, pero las paradojas todavía emergen en otras formas para desacreditar su rechazo. Esto es algo intensamente contra intuitivo, como la negación que el conjunto de todos los conjuntos no es el mismo un conjunto como muchos lo son por llamarse una "clase". Russell dijo en su artículo "Vaguedad" de 1923 que podemos relajar esta situación si no rechazamos la ley de Aristóteles de la exclusión del justo medio para permitir el trato con las paradojas y la relación con la vaguedad de los enunciados reales. La lógica difusa tuvo su comienzo formal con este artículo de 1923.

Las matemáticas muestran que la aplicación de la lógica difusa resuelve la paradoja del mentiroso con el punto medio de valor verdadero de $\frac{1}{2}$ ó 50%. Las paradojas son medias verdades. Los Cretenses mienten el 50% y no mienten el 50%.

Las paradojas motivaron muchos de los trabajos en vaguedad o lógica difusa. El lógico Polaco Jan Lukasiewicz hizo el mejor avance después de Russell. En la década de 1920 Lukasiewicz sacó su primer trabajo de lógica difusa o multivaluada [Lukasiewicz, J., "*Philisophical Remarks on Many-Valued Systems of Propositional Logic*," *Selected Works*, Borkowski, editor, *Studies in Logic and the Foundations of Mathematics*, 153-179, North Holland, 1970]. Los problemas cotidianos presentan esta lógica y sus operaciones básicas de mínimos, máximos y su negación. En 1937 un artículo de "Philosophy of Science" el filósofo cuántico Max Black aplicó lógica multivaluada para catalogar o hacer conjuntos de objetos e hizo el primer conjunto de curvas difusas [Black, M., "*Vagueness: An Exercise in Logical Análisis*," *Philosophy of Science*, vol. 4, 427-455, 1937 y "*Reasoning with Loose Concepts*," *Dialogue*, vol. 2, 1-12, 1963]. Estos conjuntos A son tales que cada objeto x obedece o pertenece a A

y NO-A en algún grado y por lo tanto son propiamente vagos o difusos. Black siguió el uso de los términos de Russell y llamó a los conjuntos “vagos”. Kaplan y Schott [Kaplan, A., and Schott, H. A., “*A Calculus for Empirical Classes,*” *Methodos III*, vol.11, 165-188, 1951] propusieron las operaciones para los mínimos y máximos para definir el álgebra de conjuntos difusos como lo hicieron otros lógicos en los 1950s.

En 1965 Lotfi Zadeh de la Universidad de Berkeley en California publicó su trabajo llamado “Conjuntos Difusos” [Zadeh, L. A., “*Fuzzy Sets,*” *Information and Control*, vol. 8, 338-353, 1965 y “*Fuzzy Sets and Applications: Selected Papers,*” R. R. Yager, S. Ovchinnikov, R.M. Tong, and H. T. Nguyen, editors, Wiley-Interscience, 1987]. Este trabajo fue el primero en usar la palabra Difuso (Fuzzy) para significar Vago (Vague) en la literatura técnica. El término difuso no es el único que persiste pero es más usado que el término vago. El trabajo de Zadeh de 1965 aplica la lógica de Lukasiewicz a cada objeto en el conjunto para hacer un trabajo completo de álgebra difusa de conjuntos y para extender el teorema de la separación convexa del reconocimiento del patrón. Existe cierta controversia aquí [Kosko, B., *Fuzzy Thinking*, Hyperion, New York, 1993 y McNeill, D., and Freiberger, P., *Fuzzy Logic*, Simon and Schuster, 1993], porque Zadeh no se refiere a los trabajos de Lukasiewicz o a ningún otro de los trabajos de los lógicos de multivaluación, que desde mucho tiempo atrás definen los conceptos de vago y la puntualización de los operadores (min., máx., y $1 - x$) en la teoría de los conjuntos difusos. Esto permite la percepción que la lógica multivaluada comienza con los conjuntos difusos (Fuzzy sets) dejando atrás medio siglo de trabajos anteriores. Por lo tanto, Zadeh pertenece a la segunda generación de investigadores de lógica multivaluada bajo las tendencias y el lenguaje de la lógica difusa. La IEEE lo premió con la medalla de honor en 1995 por su trabajo de Conjuntos Difusos.

A mediados de 1970s Ebrahim H. Mamdani del Queen Mary Collage en Londres fue el primero en aplicar conjuntos difusos a los sistemas en la forma de reglas-base en sistemas difusos [Mamdani, E. H., “*Application of Fuzzy Logic to Approximate Reasoning Using Lingustic Synthesis,*” *IEEE Transactions on Computers*, vol. C-26, no. 12, 1182-1191, December 1977]. Mamdani diseñó un sistema difuso para controlar una máquina de vapor, éste sistema difiere solo en detalles de los sistemas difusos usados ahora. El trabajo de Mamdani marca el comienzo de la Ingeniería Difusa.

1.2 Definiciones y conceptos básicos.

Las lógicas difusas, pues de hecho hay que hablar de ellas en plural, son esencialmente lógicas multivaluadas que extienden a las lógicas clásicas. Estas últimas imponen a sus enunciados únicamente valores falso o verdadero. Bien que éstas han modelado satisfactoriamente a una gran parte del razonamiento “natural”, es cierto que el razonamiento humano utiliza valores de verdad que no necesariamente son “tan deterministas”. Las lógicas difusas procuran crear aproximaciones matemáticas en la resolución de ciertos tipos de problemas. Pretenden producir resultados exactos a partir de datos imprecisos, por lo cual son particularmente útiles en aplicaciones electrónicas o

computacionales. El adjetivo “difuso” aplicado a ellas se debe a que los valores de verdad no deterministas utilizados en ellas tienen, por lo general, una connotación de incertidumbre. Un vaso medio lleno, independientemente de que también esté medio vacío, no está lleno completamente ni está vacío completamente. Qué tan lleno puede estar es un elemento de incertidumbre, es decir, de difusidad, entendida esta última como una propiedad de indeterminismo. Ahora bien, los valores de verdad asumidos por enunciados aunque no son deterministas, no necesariamente son desconocidos. Por otra parte, desde un punto de vista optimista, lo difuso puede entenderse como la posibilidad de asignar más valores de verdad a los enunciados que los clásicos “falso” o “verdadero”. Así pues, reiteramos, las lógicas difusas son tipos especiales de lógicas multivaluadas. Las lógicas difusas han tenido aplicaciones de suma relevancia en el procesamiento electrónico de datos. En determinadas áreas de conocimiento, a sus enunciados se les asocia valores de verdad que son grados de veracidad de los enunciados de salida conforme se refinan los de los de entrada. Por estas propiedades es que ciertos sistemas de lógica difusa aparentan una labor de aprendizaje, y son excelentes mecanismos de control de procesos. Desde el punto de vista tecnológico, las lógicas difusas se encuadran en el área de la llamada Inteligencia Artificial y han dado origen a sistemas expertos de tipo difuso y a sistemas de control automático.

La lógica difusa es la manera en como el cerebro humano trabaja, y nosotros podemos imitar esto en las máquinas de manera tal que puedan desempeñarse como humanos (no hay que confundirlo con Inteligencia Artificial, donde el objetivo es que las máquinas se desempeñen exactamente como los humanos). Los sistemas de análisis y control con lógica difusa pueden ser electromecánicos por naturaleza o relacionado solamente con datos, por ejemplo datos económicos, en todos los casos guiados por reglas “if-then” (si-entonces) establecidos en el lenguaje humano.

El análisis de lógica difusa y los métodos de control son, entonces:

1. La recepción de uno o un gran número de mediciones o algunos otros tipos de factores existentes en el sistema que queremos analizar o controlar.
2. El procesamiento de éstas entradas de acuerdo a las bases humanas, reglas difusas “if-then”, que pueden ser expresadas en palabras de lenguaje, en combinación con procesadores tradicionales no-difusos.
3. Promediado y pesando de los resultados de salida desde todas las reglas individuales dentro de una simple decisión de salida que decide que hacer o decir a un sistema de control para actuar. La señal de salida llega eventualmente y se convierte en un valor “crisp” (absoluto).

La lógica difusa es una metodología de sistema de control para resolver problemas que permiten por sí mismos la implementación de sistemas en rangos desde simples y pequeños microcontroladores hasta grandes PCs multicanal y en red o estaciones de trabajo para adquisición de datos y

sistemas de control. Puede ser implementado en hardware, software o una combinación de los dos.

Debido al carácter multivaluado de las lógicas difusas, es importante introducir la noción de variables lingüísticas así como de conjunto difuso y las operaciones usuales en ese tipo de conjuntos.

1.2.1 Variables lingüísticas.

Una Variable Lingüística es aquella variable cuyos valores son palabras o sentencias que van a enmarcarse en un lenguaje predeterminado. Para estas variables lingüísticas se utilizará un nombre y un valor lingüístico sobre un Universo de Discurso. Además, podrán dar lugar a sentencias generadas por reglas sintácticas, a las que se les podrá dar un significado mediante distintas reglas semánticas.

En 1973, el Profesor Lofti Zadeh propuso un concepto de variables lingüísticas o difusas. Se piensa en ellas como los objetos lingüísticos o palabras, más que en números. La entrada del sensor es un sustantivo, por ejemplo, "temperatura", "velocidad", "flujo", "presión", etc. Dado que un error es solo la diferencia de valores, no se puede pensar en el de la misma manera. Las mismas variables lingüísticas son adjetivos que modifican las variables (por ejemplo: error "grande y positivo", error "cero", error "pequeño y negativo" y error "grande y negativo"). Como mínimo, uno podría simplemente tener variables "positivas", "cero" y "negativas" para cada uno de los parámetros. Rangos adicionales como "muy grandes" y "muy pequeñas" podrían también ser agregadas para extender la respuesta a condiciones excepcionales y no lineales, pero no son necesarias en sistemas básicos.

El centro de las técnicas de modelado difuso es la idea de variable lingüística. Desde su raíz, una variable lingüística es el nombre de un conjunto difuso. Si tenemos un conjunto difuso llamado "largo" éste es una simple variable lingüística y puede ser empleada como una regla-base en un sistema basado en la longitud de un proyecto en particular. Si la duración-proyecto es larga entonces la terminación de tareas es DECRECIENTE. Una variable lingüística encapsula las propiedades de aproximación o conceptos de imprecisión en un sistema y da una forma de computar adecuada. Esto reduce la aparente complejidad de describir un sistema que debe concordar con su semántica. Una variable lingüística siempre representa un espacio difuso.

Lo importante del concepto de variable lingüística es su estimación de variable de alto orden más que una variable difusa. En el sentido de que una variable lingüística toma variables difusas como sus valores. En el campo de la semántica difusa cuantitativa al significado de un término "x" se le representa como un conjunto difuso $M(x)$ del universo de discusión. Desde este punto de vista, uno de los problemas básicos en semántica es que se desea calcular el significado de un término compuesto.

La idea básica sugerida por Zadeh es que una etiqueta lingüística tal como "muy", "más o menos", "ligeramente", etc... puede considerarse como un operador que actúa sobre un conjunto difuso asociado al significado de su operando. Por ejemplo en el caso de un término compuesto "muy alto", el operador "muy" actúa en el conjunto difuso asociado al significado del operando "alto". Una representación aproximada para una etiqueta lingüística se puede lograr en términos de combinaciones o composiciones de las operaciones básicas. Es importante aclarar que se hará mayor énfasis en que estas representaciones se proponen principalmente para ilustrar el enfoque, más que para proporcionar una definición exacta de las etiquetas lingüísticas. Zadeh también considera que las etiquetas lingüísticas pueden clasificarse en dos categorías que informalmente se definen como sigue:

- Tipo I: las que pueden representarse como operadores que actúan en un conjunto difuso: "muy", "más o menos", "mucho", "ligeramente", "altamente", "bastante", etc. y,
- Tipo II: las que requieren una descripción de cómo actúan en los componentes del conjunto difuso (operando): "esencialmente", "técnicamente", "estrictamente", "prácticamente", "virtualmente", etc...

En otras palabras, las variables lingüísticas pueden ser caracterizadas como operadores más que construcciones complicadas sobre las operaciones primitivas de conjuntos difusos.

Ejemplos de etiquetas tipo I.

De acuerdo a éste punto de vista y sabiendo que el lenguaje natural es muy rico y complejo, tomamos el operador "muy" que podemos caracterizar con un significado de que aún cuando no tenga validez universal sea sólo una aproximación. Asumimos que si el significado de un término x es un conjunto difuso A , entonces el significado de $\text{muy } x$ pertenece a este conjunto difuso A .

Más y menos

Se pueden definir etiquetas lingüísticas artificiales, por ejemplo: más, menos, que son instancias de lo que puede llamarse acentuador y desacentuador respectivamente, cuya función es proporcionar ligeras variantes de la concentración y la dilatación.

Los exponentes se eligen de modo que se de la igualdad aproximada: $\text{mas mas } x = \text{menos muy } x$, y que, además, se pueden utilizar para definir etiquetas lingüísticas cuyo significado difiere ligeramente de otras, ejemplo:

Más o menos

Otra etiqueta lingüística interesante es "más o menos" que en sus usos más comunes como "más o menos inteligente", "más o menos rectangular" etc, juega el papel de desdifusor.

Ligeramente

Su efecto es dependiente de la definición de proximidad u ordenamientos en el dominio del operando. Existen casos, sin embargo, en los que su significado puede definirse en términos de etiquetas lingüísticas tipo I, bajo la suposición de que el dominio del operando es un conjunto ordenado linealmente.

Clase de

Es una etiqueta lingüística que tiene el efecto de reducir el grado de pertenencia de los elementos que están en el "centro" (grados de pertenencia grandes) de una clase x e incrementa el de aquellos que están en su periferia (grados de pertenencia pequeños).

Regular

Es una etiqueta que tiene el efecto de reducir el grado de pertenencia de aquellos elementos que tienen tanto un alto grado de pertenencia al conjunto como de aquellos que lo tienen pequeño, y sólo aumenta el grado de pertenencia de aquellos elementos que tienen un grado de pertenencia cercano.

Etiquetas tipo II.

Su caracterización envuelve una descripción de forma que afectan a los componentes del operando, y por lo tanto es más compleja que las del tipo I. En general, la definición de una etiqueta de este tipo debe formularse como un algoritmo difuso que envuelve etiquetas tipo I. Su efecto puede describirse aproximadamente como una modificación de los coeficientes de ponderación de una combinación convexa. Como la magnitud de las ponderaciones es una medida del atributo asociado, intuitivamente una etiqueta de este tipo tiene el efecto de aumentar las ponderaciones de los atributos importantes y disminuir los que relativamente no lo son.

1.2.2 Conjuntos difusos.

La Teoría de conjuntos difusos puede utilizarse para representar expresiones lingüísticas que se utilizan para describir conjuntos o algoritmos. Los conjuntos difusos son capaces de captar por sí mismos la vaguedad lingüística de palabras y frases comúnmente aceptadas, como "gato pardo" o "ligero cambio". La habilidad humana de comunicarse mediante definiciones vagas o inciertas es un atributo importante de la inteligencia.

De manera intuitiva se tiene el concepto de conjunto como una colección bien definida de elementos, en la que es posible determinar para un objeto cualquiera, en un universo dado, si acaso éste pertenece o no al conjunto. La decisión, naturalmente, es “si pertenece” o bien “no pertenece”.

El primer ejemplo utilizado por Lofti A. Zadeh, para ilustrar el concepto de conjunto difuso, fue el conjunto “hombres altos”. Según la teoría de la lógica clásica el conjunto “hombres altos” es un conjunto al que pertenecieran los hombres con una estatura mayor a un cierto valor, que podemos establecer en 1.80 metros, por ejemplo, y todos los hombres con una altura inferior a este valor quedarían fuera del conjunto. Así tendríamos que un hombre que mide 1.81 metros de estatura pertenecería al conjunto hombres altos, y en cambio un hombre que mida 1.79 metros de altura ya no pertenecería a ese conjunto. Sin embargo no parece muy lógico decir que un hombre es alto y otro no lo es cuando su altura difiere en dos centímetros. El enfoque de la lógica difusa considera que el conjunto “hombres altos” es un conjunto que no tiene una frontera clara para pertenecer o no pertenecer a él: mediante una función que define la transición de “alto” o “no alto” se asigna a cada valor de altura un grado de pertenencia al conjunto, entre 0 y 1. Así por ejemplo, un hombre que mida 1.79 podría pertenecer al conjunto difuso “hombres altos” con un grado 0.8 de pertenencia, uno que mida 1.81 con un grado 0.85, y uno que mida 1.50 m con un grado 0.1. Visto desde esta perspectiva se puede considerar que la lógica clásica es un caso límite de la lógica difusa en el que se asigna un grado de pertenencia 1 a los hombres una altura mayor o igual a 1.80 y un grado de pertenencia 0 a los que tienen una altura menor.

Así pues, los conjuntos difusos pueden ser considerados como una generalización de los conjuntos clásicos: la teoría clásica de conjuntos solo contempla la pertenencia o no pertenencia de un elemento a un conjunto, sin embargo la teoría de conjuntos difusos contempla la pertenencia parcial de un elemento a un conjunto, es decir, cada elemento presenta un grado de pertenencia a un conjunto difuso que puede tomar cualquier valor entre 0 y 1. Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso: para cada valor que pueda tomar un elemento o variable de entrada x la función característica $\mu_A(x)$ proporciona el grado de pertenencia de este valor de x al conjunto difuso A .

Formalmente, un conjunto clásico A , en un universo de discurso U , se puede definir de varias formas: enumerando los elementos que pertenecen al conjunto, especificando las propiedades que deben cumplir los elementos que pertenecen a ese conjunto o, en términos de la función de pertenencia $\mu_A(x)$:

$$\begin{aligned}\mu_A(x) &= 1 \text{ si } x \in A \\ \mu_A(x) &= 0 \text{ si } x \notin A\end{aligned}$$

Podemos además decir que el conjunto A es matemáticamente equivalente a su función de pertenencia o característica $\mu_A(x)$, ya que conocer $\mu_A(x)$ es lo mismo que conocer A .

Un conjunto difuso en el universo de discurso U se caracteriza por una función de pertenencia $\mu_A(x)$ que toma valores en el intervalo $[0,1]$ y puede

representarse como un conjunto de pares ordenados de un elemento x y su valor de pertenencia al conjunto:

$$A = \{ (x, \mu_A(x)) \mid x \in U \}$$

Muchos conceptos de teoría clásica de conjuntos se pueden hacer extensivos a los conjuntos difusos, otros son exclusivos e inherentes a la teoría de conjuntos difusos. Algunos de los más utilizados son los siguientes:

- El soporte de un conjunto difuso A en el universo de discurso U es un conjunto “crisp” (absoluto) numérico que contiene todos los elementos de U que tienen un valor de pertenencia distinto de cero en A , esto es, $\text{sop}(A) = \{ x \in U \mid \mu_A(x) > 0 \}$

Si el soporte de un conjunto difuso no contiene ningún elemento tendremos un conjunto difuso vacío. Si el soporte de un conjunto difuso es un solo punto tendremos lo que se conoce como “singleton” (unitario) difuso.

- El punto de cruce de un conjunto difuso es el punto de U cuyo valor de pertenencia al conjunto es igual a 0.5
- Dos conjuntos difusos A y B son iguales si y sólo si sus funciones características $\mu_A(x)$ y $\mu_B(x)$ son iguales
- El conjunto difuso B contiene al conjunto difuso A , esto es $A \subset B$, si y sólo si $\mu_A(x) \leq \mu_B(x)$ para todo $x \in U$

La función característica proporciona una medida del grado de similaridad de un elemento de U con el conjunto difuso. La forma de la función característica utilizada, depende del criterio aplicado en la resolución de cada problema y variará en función de la cultura, geografía, época o punto de vista del usuario. La única condición que debe cumplir una función característica es que tome valores entre 0 y 1, con continuidad. Las funciones características más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son: triangular, trapezoidal, gaussiana, sigmoideal, gamma, pi, campana, etc. Conceptualmente existen dos aproximaciones para determinar la función característica asociada a un conjunto: la primera aproximación está basada en el conocimiento humano de los expertos y la segunda aproximación es utilizar una colección de datos para diseñar la función.

El número de funciones características asociadas a una misma variable es elegido por el experto: a mayor número de funciones características tendremos mayor resolución pero también mayor complejidad computacional; además estas funciones pueden estar solapadas o no, el hecho de estar solapadas pone de manifiesto un aspecto clave de la lógica difusa: una variable puede pertenecer con diferentes grados a varios conjuntos difusos a la vez.

Un conjunto difuso es un conjunto que contiene elementos los cuales tienen variedad de grado de membresía en el mismo. Esta idea contrasta con la teoría clásica o absoluta (crisp), donde los miembros de un conjunto absoluto no lo son hasta que su membresía no sea total o completa en dicho conjunto (la membresía tenga asignado el valor de 1). Los elementos en un conjunto difuso,

donde sus valores de membresía pueden tener valores diferentes del absoluto o completo, pueden por tanto ser miembros de otros conjuntos difusos en el mismo universo.

Elementos de un conjunto difuso son mapeados a un universo de “valores de membresía” usando la forma de la función-teorética. Los conjuntos difusos serán denotados por un conjunto de símbolos con una línea ondulada debajo, $\underline{\hat{A}}$ el cual va a ser el conjunto difuso A. Esta función mapea elementos de un conjunto difuso $\underline{\hat{A}}$ a valores reales en el intervalo 0 a 1. Si un elemento en el universo, llamado x, es un miembro de un conjunto difuso $\underline{\hat{A}}$ entonces su mapeo es dado por:

$$\mu_{\underline{\hat{A}}}(x) \in [0,1]$$

$$\underline{\hat{A}} = (x, \mu_{\underline{\hat{A}}}(x) | x \in X)$$

Una notación convencional para conjuntos difusos cuando el universo del discurso, X, es discreto y finito es dada por:

$$\underline{\hat{A}} = \frac{\mu_{\underline{\hat{A}}}(x_1)}{x_1} + \frac{\mu_{\underline{\hat{A}}}(x_2)}{x_2} + \dots = \sum_i \frac{\mu_{\underline{\hat{A}}}(x_i)}{x_i}$$

y cuando el universo X, es continuo e infinito, tenemos:

$$\underline{\hat{A}} = \int \frac{\mu_{\underline{\hat{A}}}(x)}{x}$$

En ambas notaciones, el numerador en cada expresión individual es el valor de membresía en el conjunto $\underline{\hat{A}}$ asociado con el elemento en el universo indicado en el denominador de cada expresión. En la primera notación, el símbolo de sumatoria no es para una suma algebraica, mas bien tiende a denotar una unión difusa, por lo tanto los signos “+” en la primera notación no son una “suma” algebraica. En la segunda notación el signo de integral no es una integral algebraica, pero delimita lo que es una notación para la unión de conjuntos para variables continuas.

1.2.3 Operaciones básicas entre conjuntos difusos.

Definimos tres conjuntos difusos $\underline{\hat{A}}, \underline{\hat{B}}, \underline{\hat{C}}$ en el universo X. Para un elemento x en el universo, las siguientes funciones unión, intersección y complemento son definidas como operaciones básicas entre conjuntos difusos:

$$\underline{\hat{A}}, \underline{\hat{B}}, \underline{\hat{C}} \text{ en } X$$

Unión: $\mu_{\underline{\hat{A}} \cup \underline{\hat{B}}}(x) = \mu_{\underline{\hat{A}}}(x) \vee \mu_{\underline{\hat{B}}}(x)$

Intersección: $\mu_{\underline{\hat{A}} \cap \underline{\hat{B}}}(x) = \mu_{\underline{\hat{A}}}(x) \wedge \mu_{\underline{\hat{B}}}(x)$

Complemento: $\mu_{\underline{\hat{A}}^c}(x) = 1 - \mu_{\underline{\hat{A}}}(x)$

Los diagramas de Venn de estas operaciones, considerando conjuntos difusos se muestran en las figuras 1.1 a 1.3.

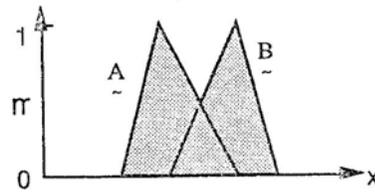


Figura 1.1 Unión de conjuntos difusos.

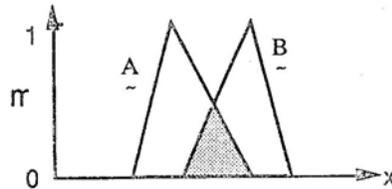


Figura 1.2 Intersección de conjuntos difusos.

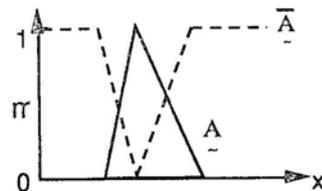


Figura 1.3 Complemento de un conjunto difuso.

Estas tres operaciones definidas para conjuntos difusos cumplen, al igual que en la teoría clásica de conjuntos con asociatividad, conmutatividad y distributividad así como las leyes de Morgan.

Por definición, como en los conjuntos clásicos, el valor de membresía de cualquier elemento x en el conjunto \emptyset es 0, y el valor de membresía de un elemento x en el conjunto completo es 1, observe que el conjunto nulo y el completo no son conjuntos difusos en este contexto.

La notación apropiada para estas ideas es:

$$A \subseteq X \rightarrow \mu_A(x) \leq \mu_X(x)$$

$$\forall x \in X, \mu_{\emptyset}(x) = 0$$

$$\forall x \in X, \mu_X(x) = 1$$

La colección o grupo de todos los conjuntos difusos y subconjuntos difusos en X se denota como el conjunto difuso de poder $P(X)$. Se observa, basando en el hecho de que todos los conjuntos difusos pueden superponerse o trasladarse, que la cardinalidad del conjunto difuso de poder es infinita:

$$\text{Cardinality of } P(X) \rightarrow n_{P(X)} = \infty$$

Las leyes de DeMorgan para conjuntos clásicos pueden aplicarse para conjuntos difusos, denotándose con las expresiones siguientes:

$$\overline{(A \cap B)} = \bar{A} \cup \bar{B}$$

$$\overline{(A \cup B)} = \bar{A} \cap \bar{B}$$

Algunas otras operaciones pueden aplicarse para conjuntos difusos, menos las leyes de la exclusión del medio. Debido a que los conjuntos difusos se traslapan, un conjunto difuso y su complemento se traslapan, esto se expresa por:

$$A \cup \bar{A} \neq X$$

$$A \cap \bar{A} \neq \emptyset$$

Se muestran los diagramas de Venn para estas situaciones:

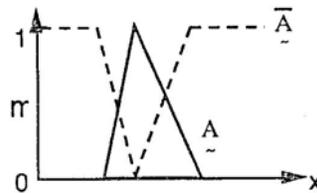


Figura 1.4 Leyes de exclusión de la mitad de un Conjunto difuso y su complemento.

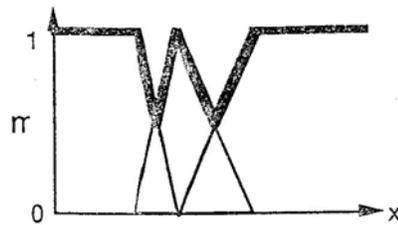


Figura 1.5 Leyes de exclusión de la mitad para $A \cup \bar{A} = X$.

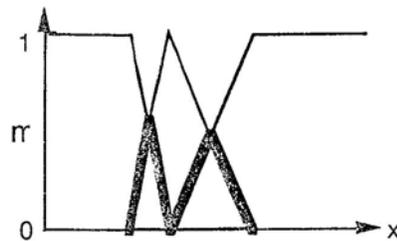


Figura 1.6 Leyes de exclusión de la mitad para $A \cap \bar{A} \neq \emptyset$.

1.2.4 Propiedades de los conjuntos difusos.

Las propiedades de los conjuntos difusos son las siguientes:

Commutativa:
$$\begin{aligned} \underline{\underline{A}} \cup \underline{\underline{B}} &= \underline{\underline{B}} \cup \underline{\underline{A}} \\ \underline{\underline{A}} \cap \underline{\underline{B}} &= \underline{\underline{B}} \cap \underline{\underline{A}} \end{aligned}$$

Asociativa:
$$\underline{\underline{A}} \cup (\underline{\underline{B}} \cup \underline{\underline{C}}) = (\underline{\underline{A}} \cup \underline{\underline{B}}) \cup \underline{\underline{C}} \text{ and } \underline{\underline{A}} \cap (\underline{\underline{B}} \cap \underline{\underline{C}}) = (\underline{\underline{A}} \cap \underline{\underline{B}}) \cap \underline{\underline{C}}$$

Distributiva:
$$\underline{\underline{A}} \cup (\underline{\underline{B}} \cap \underline{\underline{C}}) = (\underline{\underline{A}} \cup \underline{\underline{B}}) \cap (\underline{\underline{A}} \cup \underline{\underline{C}}) \text{ and } \underline{\underline{A}} \cap (\underline{\underline{B}} \cup \underline{\underline{C}}) = (\underline{\underline{A}} \cap \underline{\underline{B}}) \cup (\underline{\underline{A}} \cap \underline{\underline{C}})$$

Idempotencia:
$$\underline{\underline{A}} \cup \underline{\underline{A}} = \underline{\underline{A}} \text{ and } \underline{\underline{A}} \cap \underline{\underline{A}} = \underline{\underline{A}}$$

Identidad:
$$\underline{\underline{A}} \cup \emptyset = \underline{\underline{A}} \text{ and } \underline{\underline{A}} \cap X = \underline{\underline{A}} \text{ and } \underline{\underline{A}} \cap \emptyset = \emptyset \text{ and } \underline{\underline{A}} \cup X = X$$

Transitiva:
$$\text{If } \underline{\underline{A}} \subseteq \underline{\underline{B}} \subseteq \underline{\underline{C}}, \text{ then } \underline{\underline{A}} \subseteq \underline{\underline{C}}$$

Involución:
$$\overline{\underline{\underline{A}}} = \underline{\underline{A}}$$

Sin embargo, también hay que destacar que existen dos leyes fundamentales de la teoría clásica de conjuntos como son el Principio de contradicción: $a \cup A = U$ y el Principio de exclusión: $a - A = 0$ que no se cumplen en la teoría de conjuntos difusos; de hecho una de las formas para describir en qué se diferencia la teoría clásica de conjuntos de la teoría difusa es explicar que estas dos leyes en términos de Lógica Difusa no se cumplen. En consecuencia, algunas de las teorías derivadas de la teoría de conjuntos como por ejemplo la de la probabilidad será muy diferente planteada en términos difusos.

➤ Principio de extensión.

Suponemos un mapeo entre elementos, u , de un universo, U , en los elementos, v , de un universo, V , a través de la función f . El principio de extensión descrito por Zadeh (1975) y después elaborado por Yager (1986) nos permite extender el dominio de la función en conjuntos difusos. Este mapeo es descrito por:

$f: u \rightarrow v$, definimos $\underline{\underline{A}}$ como un conjunto difuso en el universo U , tal que $\underline{\underline{A}} \subseteq U$, y

$$\underline{\underline{A}} = \frac{\mu_1}{u_1} + \frac{\mu_2}{u_2} + \dots + \frac{\mu_n}{u_n}$$

entonces el principio de extensión permite que para una función f la cual mapea un elemento en el universo U en un elemento en el universo V , tengamos:

$$f(\underline{A}) = f\left(\frac{\mu_1}{u_1} + \frac{\mu_2}{u_2} + \dots + \frac{\mu_n}{u_n}\right)$$

$$= \frac{\mu_1}{f(u_1)} + \frac{\mu_2}{f(u_2)} + \dots + \frac{\mu_n}{f(u_n)}$$

Este mapeo es llamado "uno-a-uno".

Ejemplo:

Tenemos un conjunto difuso \underline{A} que esta definido en el universo $U=\{1,2,3\}$. Buscamos mapear elementos de este conjunto difuso en otro universo, V , bajo la función $v=f(u) = 2u-1$.

Los elementos de $V=\{1,3,5\}$ y suponemos que el conjunto difuso esta dado por:

$$\underline{A} = \frac{0.6}{1} + \frac{1}{2} + \frac{0.8}{3},$$

entonces la función difusa de membresía para $v=f(u)=2u-1$ será:

$$f(u_1)=2(1)-1=1$$

$$f(u_2)=2(2)-1=3$$

$$f(u_3)=2(3)-1=5, \text{ quedando:}$$

$$f(\underline{A}) = \frac{0.6}{1} + \frac{1}{3} + \frac{0.8}{5}$$

Para casos donde este mapeo funcional f mapee productos de elementos de dos universos, llamados U_1 y U_2 , en otro universo V , y si definimos \underline{A} como un conjunto difuso en el espacio cartesiano $U_1 \times U_2$, tenemos:

$$f(\underline{A}) = \left\{ \frac{\min[\mu_1(i), \mu_2(j)]}{f(i, j)} \mid i \in U_1, j \in U_2 \right\}$$

Ejemplo:

Tenemos los enteros 1 a 10 de dos idénticos pero diferente universos, $U_1=U_2=\{1,2,3,\dots,10\}$, definimos dos conjuntos difusos en el universo U_1 y U_2 , respectivamente:

$$\underline{A} = \underline{2} = \text{"approximately 2"} = \frac{0.6}{1} + \frac{1}{2} + \frac{0.8}{3},$$

$$\underline{B} = \underline{6} = \text{"approximately 6"} = \frac{0.8}{5} + \frac{1}{6} + \frac{0.7}{7},$$

entonces el producto de ("approximately 2") x ("approximately 6") nos permite mapear un numero difuso "approximately 12", el cual es un conjunto difuso definido en un universo, llamado V , de enteros, $V=\{5,6,\dots,18,21\}$, que es determinado por el principio de extensión como se muestra a continuación:

$$\begin{aligned}
2 \times 6 &= \left(\frac{0.6}{1} + \frac{1}{2} + \frac{0.8}{3} \right) \left(\frac{0.8}{5} + \frac{1}{6} + \frac{0.7}{7} \right) \\
&= \frac{\min(0.6, 0.8)}{5} + \frac{\min(0.6, 1)}{6} + \dots + \frac{\min(0.8, 1)}{18} + \frac{\min(0.8, 0.7)}{21} \\
&= \frac{0.6}{5} + \frac{0.6}{6} + \frac{0.6}{7} + \frac{0.8}{10} + \frac{1}{12} + \frac{0.7}{14} + \frac{0.8}{15} + \frac{0.8}{18} + \frac{0.7}{21}
\end{aligned}$$

La complejidad del principio de extensión se incrementa cuando consideramos que la combinación de las variables de entrada, U1 y U2, son mapeadas a la misma variable en el espacio de salida, V. En este caso tomaremos los grados de membresía máximos de las combinaciones mapeando a la misma variable de salida, donde tenemos:

$$\begin{aligned}
\mu_A(u_1, u_2) &= \max\{\min\{\mu_1(u_1), \mu_2(u_2)\}\} \\
v &= f(u_1, u_2)
\end{aligned}$$

Ejemplo:

Tenemos dos conjuntos difusos, cada uno definido en su propio universo:

$$a = \frac{0.2}{1} + \frac{1}{2} + \frac{0.7}{4} \quad \text{y} \quad b = \frac{0.5}{1} + \frac{1}{2}$$

para determinar los valores de membresía para el mapeo tenemos $f(a, b) = a \times b$

$$\begin{aligned}
&= \frac{\min(0.2, 0.5)}{1} + \frac{\max[\min(0.2, 1), \min(0.5, 1)]}{2} + \frac{\max[\min(0.7, 0.5), \min(1, 1)]}{4} + \frac{\min(0.7, 1)}{8} \\
&= \frac{0.2}{1} + \frac{0.5}{2} + \frac{1}{4} + \frac{0.7}{8}
\end{aligned}$$

en este caso el mapeo envuelve dos maneras de obtener a 2 (1x2 y 2x1) y dos para obtener 4 (4x1 y 2x2).

1.2.5 Características de la función de membresía.

Toda la información contenida en un conjunto difuso se describe como su función de membresía está es útil para describir en términos de lenguaje varias condiciones especiales de esta función, como se muestra en la figura 1.7.

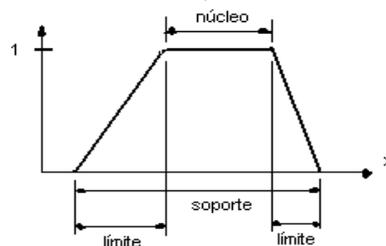


Figura 1.7 Núcleo, soporte y límite de un conjunto difuso.

El núcleo comprende todos los elementos del universo, x, donde $\mu_A(x) = 1$.

El soporte es comprendido por todos aquellos elementos en el universo, x ,

donde $\mu_{\tilde{A}}(x) \neq 0$.

Los límites son comprendidos por todos aquellos elementos en el universo, x ,

donde $0 < \mu_{\tilde{A}}(x) < 1$.

Un conjunto difuso normal es aquel que tiene al menos un elemento en el universo, x , cuya función de membresía valga 1, este elemento es referido como el "prototipo" del conjunto o el elemento prototípico.

Un conjunto difuso convexo se describe como una función de membresía cuyos valores de membresía son estricta y monotómicamente incrementados o cuyos valores de membresía son estricta y monotómicamente decrecientes.

Las funciones que definen la unión y la intersección de conjuntos difusos pueden generalizarse, a condición de cumplir ciertas restricciones. Las funciones que cumplen estas condiciones se conocen como Conorma Triangular (T – Conorma) y Norma Triangular (T – Norma). Los principales operadores que cumplen las condiciones para ser t-conormas son el operador máximo y la suma algebraica $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x)$ y los principales operadores que cumplen las condiciones para ser t-normas son el operador mínimo y el producto algebraico $\mu_{A \cap B}(x) = \mu_A(x) \mu_B(x)$. En la mayoría de las aplicaciones a la ingeniería de la lógica difusa se usan como t-conorma el operador máximo y como t-norma los operadores mínimo o producto.

Los conjuntos intuitivos pueden combinarse mediante las operaciones, llamadas booleanas, de complemento, unión e intersección: el complemento de un conjunto está formado por los elementos del universo que no están en él, la unión de dos conjuntos la forman los elementos que están en uno o en otro conjunto y la intersección la conforman los elementos en ambos conjuntos. Si nos referimos a funciones características del conjunto, la característica de la unión de dos conjuntos vale uno si al menos una de las características de los conjuntos vale uno, y la característica de la intersección vale uno si las características de ambos conjuntos valen uno. Esto puede servir de motivación para definir operadores de composición de conjuntos difusos. De hecho, para cada una de las interpretaciones descritas arriba se puede introducir una colección particular de operadores.

A partir de operaciones de complemento, unión e intersección, se obtienen conjuntos difusos "más complejos" como resultado de aplicar sucesivamente estos operadores partiendo de una colección de conjuntos difusos, digamos, "primitivos". Para hablar con un poco de más precisión: Si A_1, \dots, A_n son conjuntos difusos primitivos, para operadores de complemento, de intersección y de unión fijos la clase de conjuntos definibles, partiendo de los conjuntos primitivos, son los que se obtienen mediante las reglas siguientes:

- 1.- Todo conjunto primitivo es definible.
- 2.- El complemento de todo definible es, a su vez, definible.
- 3.- La intersección y la unión de dos conjuntos definibles, son, a su vez, definibles también.

1.2.6 Relaciones difusas.

Una relación difusa representa el grado de presencia o ausencia de asociación, interacción o interconexión entre elementos de dos o más conjuntos difusos. Las relaciones difusas nos permiten mapear elementos de un universo, llamando X, a aquellos de otro universo, llamado Y, a través del producto Cartesiano de los dos universos. Por lo tanto la "fuerza" de la relación entre pares ordenados de dos universos no es medida con la función característica, pero es medida o graduada con la función de membresía expresando varios "grados" de la fuerza de la relación en la unidad de intervalo [0,1]. Entonces una relación difusa

\tilde{R} , es un mapeo del espacio Cartesiano $X \times Y$ en el intervalo [0,1], donde la fuerza del mapeo es expresada por la función de membresía de la relación $\mu_{\tilde{R}}(x,y)$ para los pares ordenados de los dos universos \tilde{R} .

➤ Cardinalidad en relaciones difusas.

Como la cardinalidad de conjuntos difusos en cualquier universo es infinita, entonces la cardinalidad de una relación difusa entre dos o mas universos es por lo tanto infinita.

➤ Operaciones en relaciones difusas.

Sean \tilde{R} , \tilde{S} , y \tilde{T} relaciones difusas en el espacio Cartesiano $X \times Y$. Entonces las siguientes operaciones aplican:

Unión	$\mu_{\tilde{R} \cup \tilde{S}}(x,y) = \max(\mu_{\tilde{R}}(x,y), \mu_{\tilde{S}}(x,y))$
Intersección	$\mu_{\tilde{R} \cap \tilde{S}}(x,y) = \min(\mu_{\tilde{R}}(x,y), \mu_{\tilde{S}}(x,y))$
Complemento	$\mu_{\tilde{R}^c}(x,y) = 1 - \mu_{\tilde{R}}(x,y)$
Contenido	$\tilde{R} \subset \tilde{S} \Rightarrow \mu_{\tilde{R}}(x,y) \leq \mu_{\tilde{S}}(x,y)$

Como en las relaciones absolutas (crisp), las operaciones de conmutativa, asociativa, distributiva, involución e ídem potencia son permitidas para las relaciones difusas. Las leyes de DeMorgan aplican para relaciones difusas, la relación nula, 0, y la relación completa, E, son análogos al conjunto nulo y el conjunto completo o absoluto en la teoría de conjuntos. Las operaciones que no son permitidas para relaciones difusas, como en el caso de los conjuntos difusos, son las leyes de exclusión de la mitad. Como una relación difusa \tilde{R} es después de todo un conjunto difuso, existe una sobreposición entre una relación y su complemento:

$$\tilde{R} \cup \tilde{R}^c \neq E$$

$$\tilde{R} \cap \tilde{R}^c \neq O$$

como vemos las leyes de exclusión de la mitad para relaciones no resultan en una relación nula, 0, o en una relación completa, E.

Porque las relaciones difusas, en general, son conjuntos difusos, podemos definir el producto Cartesiano entre dos conjuntos difusos. Sea \underline{A} un conjunto difuso en el universo X, sea \underline{B} un conjunto difuso en el universo Y, entonces el producto Cartesiano entre conjuntos difusos \underline{A} y \underline{B} resulta en una relación difusa \underline{R} , o

$$\underline{A} \times \underline{B} = \underline{R} \subset X \times Y$$

con función de membresía,

$$\mu_{\underline{R}}(x,y) = \mu_{\underline{A} \times \underline{B}}(x,y) = \min(\mu_{\underline{A}}(x), \mu_{\underline{B}}(y))$$

Ejemplo:

Supongamos que tenemos dos conjuntos difusos en un universo, \underline{A} y \underline{B} , y buscamos encontrar el producto Cartesiano difuso entre ellos, tenemos:

$$\underline{A} = \frac{0.2}{x_1} + \frac{0.5}{x_2} + \frac{1}{x_3} \quad \text{y} \quad \underline{B} = \frac{0.3}{y_1} + \frac{0.9}{y_2}$$

por lo tanto el producto Cartesiano difuso es:

$$\underline{A} \times \underline{B} = \underline{R} = \begin{matrix} & \begin{matrix} Y1 & Y2 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 0.2 & 0.2 \\ 0.3 & 0.5 \\ 0.3 & 0.9 \end{bmatrix} \end{matrix}$$

➤ Composición difusa.

Suponemos que \underline{R} es una relación difusa en el espacio Cartesiano X x Y, \underline{S} es una relación en Y x Z, y \underline{T} es una relación difusa en X x Z, entonces la composición difusa max-min es definida como:

$$\underline{T} = \underline{R} \circ \underline{S}$$

$$\mu_{\underline{T}}(x,z) = \bigvee_{y \in Y} \left(\mu_{\underline{R}}(x,y) \wedge \mu_{\underline{S}}(y,z) \right)$$

y la composición difusa producto-max es definida por:

$$\mu_{\underline{T}}(x,z) = \bigvee_{y \in Y} \left(\mu_{\underline{R}}(x,y) \ast \mu_{\underline{S}}(y,z) \right)$$

Ejemplo:

Tenemos las relaciones difusas entre los universos X-Y y los universos Y-Z, denotados por la siguiente relaciones difusas respectivamente:

$$R = \begin{bmatrix} 0.7 & 0.5 \\ 0.8 & 0.4 \end{bmatrix} \quad S = \begin{bmatrix} 0.9 & 0.6 & 0.2 \\ 0.1 & 0.7 & 0.5 \end{bmatrix}$$

entonces la relación difusa resultante, T la cual relaciona elementos de un universo X a elementos de un universo Z, por la composición difusa max-min:

$$T = \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.8 & 0.6 & 0.4 \end{bmatrix}$$

La composición difusa inversa no implica igualdad:

$$R \circ S \neq S \circ R$$

1.2.7 Reglas difusas.

Las reglas son postulados en lenguaje ordinario que describen las acciones que un humano experto tomaría para resolver el problema de la aplicación. Las reglas y funciones de pertenencia pueden ser reducidas a estructuras de datos relativamente simples (la base de conocimiento) almacenados en memoria no volátil. Un núcleo de inferencia difusa puede ser escrito por un programador quien no conoce como trabaja el sistema de la aplicación. La única cosa que el programador necesita hacer con la información de la base de conocimiento es almacenarla en localidades de memoria usadas por el núcleo.

Un pase de ejecución del núcleo de inferencia difusa genera señales de salida del sistema en respuesta a condiciones actuales de entrada. El núcleo es ejecutado tan a menudo como sea necesario para mantener el control. Si el núcleo es ejecutado más a menudo de lo necesario, el ancho de banda y la potencia del procesador se desperdician; tardándose mucho entre pases puede causar que el sistema se salga de control. La elección de la tasa periódica para un sistema de control difuso es lo mismo que sería para un sistema de control convencional.

Un enunciado lingüístico que tiende a expresar ideas subjetivas y el cual puede ser interpretado ligeramente diferente por varios individuos envuelve proposiciones difusas. El valor de verdad asignado a P puede ser cualquier valor en el intervalo $[0, 1]$. La asignación del valor de verdad de una proposición es un mapeo desde el intervalo $[0, 1]$ a el universo U de valores de verdad, T .

$$T: U \rightarrow [0,1]$$

Como en la lógica binaria clásica, asignamos una proposición lógica a un conjunto en el universo de discusión. Proposiciones difusas son asignadas a

conjuntos difusos. Supongamos que la proposición \underline{P} es asignada al conjunto difuso \underline{A} , entonces el valor de verdad de la proposición denotada por $T(\underline{P})$ es dado por:

$$T(\underline{P}) = \mu_{\underline{A}}(x) \text{ where } 0 \leq \mu_{\underline{A}} \leq 1$$

La graduación de verdad de la proposición $\underline{P}: x \in \underline{A}$ es igual al grado de membresía de x en \underline{A} .

Los conectivos lógicos de negación, disyunción, conjunción e implicación son definidas en seguida para la lógica difusa. Estos conectivos son dados por medio de dos simples proposiciones: \underline{P} definido en un conjunto difuso \underline{A} y \underline{Q} definido en un conjunto difuso \underline{B} .

Negación:
$$T(\underline{\bar{P}}) = 1 - T(\underline{P})$$

Disyunción:
$$\underline{P} \vee \underline{Q} \Rightarrow x \text{ is } \underline{A} \text{ or } \underline{B}$$

$$T(\underline{P} \vee \underline{Q}) = \max(T(\underline{P}), T(\underline{Q}))$$

Conjunción:
$$\underline{P} \wedge \underline{Q} \Rightarrow x \text{ is } \underline{A} \text{ and } \underline{B}$$

$$T(\underline{P} \wedge \underline{Q}) = \min(T(\underline{P}), T(\underline{Q}))$$

Implicación:
$$\underline{P} \rightarrow \underline{Q} \Rightarrow x \text{ is } \underline{A}, \text{ then } x \text{ is } \underline{B}$$

$$T(\underline{P} \rightarrow \underline{Q}) = T(\underline{\bar{P}} \vee \underline{Q}) = \max(T(\underline{\bar{P}}), T(\underline{Q}))$$

Como en la lógica binaria, el conectivo de implicación puede ser modelado en una forma de regla base:

$$\underline{P} \rightarrow \underline{Q} \quad \text{is: IF } x \text{ is } \underline{A}, \text{ THEN } y \text{ is } \underline{B}$$

y esta es equivalente a la siguiente relación difusa, \underline{R} :

$$\underline{R} = (\underline{A} \times \underline{B}) \cup (\bar{\underline{A}} \times \underline{Y})$$

cuya función de membresía es expresada por la siguiente formula:

$$\mu_{\underline{R}}(x, y) = \max\left[\left(\mu_{\underline{A}}(x) \wedge \mu_{\underline{B}}(y)\right), \left(1 - \mu_{\underline{A}}(x)\right)\right]$$

Cuando la implicación de la condicional lógica es de la forma compuesta:

$$\text{IF } x \text{ is } \underline{A}, \text{ THEN } y \text{ is } \underline{B}, \text{ ELSE } y \text{ is } \underline{C}$$

entonces la relación difusa equivalente \underline{R} , es expresada como:

$$\underline{R} = (\underline{A} \times \underline{B}) \cup (\bar{\underline{A}} \times \underline{C})$$

cuya función de membresía es expresada por la siguiente formula:

$$\mu_{\underline{R}}(x,y) = \max\left[\left(\mu_{\underline{A}}(x) \wedge \mu_{\underline{B}}(y)\right), \left(1 - \mu_{\underline{A}}(x) \wedge \mu_{\underline{C}}(y)\right)\right]$$

➤ **Razonamiento aproximado.**

Esto es formar los fundamentos teóricos para el razonamiento sobre proposiciones imprecisas, tal razonamiento es referido como razonamiento aproximado. El razonamiento aproximado es análogo a la inferencia lógica para el razonamiento con proposiciones precisas, y por lo tanto es una extensión del calculo clásico proposicional que trata con verdades parciales. Supongamos que tenemos un formato de regla base para representar información difusa. Estas reglas son expresadas en forma convencional, antecedente-consecuencia:

Regla 1: $\text{IF } x \text{ is } \underline{A}, \text{ THEN } y \text{ is } \underline{B}$, donde \underline{A} y \underline{B} representan proposiciones difusas (conjuntos).

Ahora suponemos que introducimos un nuevo antecedente, llamado $\underline{A'}$, y consideramos la siguiente regla:

Regla 2: $\text{IF } x \text{ is } \underline{A'}, \text{ THEN } y \text{ is } \underline{B'}$

De la información derivada de la regla 1, ¿es posible derivar la cosecuente $\underline{B'}$ en la regla 2?

Si, y el procedimiento es una composición difusa. La consecuenta $\underline{B'}$ puede ser encontrada de la operación de composición:

$$\underline{B'} = \underline{A'} \circ \underline{R}$$

La última aplicación de la lógica difusa es proveer fundamentos para el razonamiento aproximado con proposiciones imprecisas usando esta teoría de conjuntos difusos. Se usan los siguientes principios fundamentales en lógica difusa:

Generalized Modus Ponens.

Esta regla de inferencia da dos proposiciones difusas, $x \text{ is } \underline{A'}$ ($x \text{ es } \underline{A}$) y $\text{IF } x \text{ is } \underline{A} \text{ THEN } y \text{ is } \underline{B}$ ($\text{SI } x \text{ es } \underline{A} \text{ ENTONCES } y \text{ es } \underline{B}$), donde podemos inferir una nueva proposición difusa, $y \text{ is } \underline{B'}$ ($y \text{ es } \underline{B}$).

- Premisa 1. $x \text{ is } \underline{A'}$
- Premisa 2. $\text{IF } x \text{ is } \underline{A} \text{ THEN } y \text{ is } \underline{B}$
- Conclusión. $y \text{ is } \underline{B'}$

Generalized Modus Tollens.

- Premisa 1. $y \text{ is } \underline{B'}$
- Premisa 2. $\text{IF } x \text{ is } \underline{A} \text{ THEN } y \text{ is } \underline{B}$
- Conclusión. $x \text{ is } \underline{A'}$

Generalized Hypothetical Syllogism.

Premisa 1. IF x is A THEN y is B
Premisa 2. IF y is B' THEN z is C
Conclusión. IF x is A THEN z is C'

Para determinar la función de membresía de las proposiciones difusas en las conclusiones dadas por las premisas, usamos las reglas de inferencia composicional.

1.2.8 Desdifusión.

La entrada para el proceso de desdifusión es un conjunto difuso (la salida de la etapa de agregación) y la salida es un simple número concreto. Los sistemas del mundo real requieren un resultado numérico. Es un proceso en el que se determina un valor definido z^* , que representa a los conjuntos difusos de salida.

Existen varios métodos para realizar la desdifusión:

Método del centroide. Consiste en determinar el punto de equilibrio de la figura formada por los conjuntos difusos de salida, con la expresión:

$$z^* = \frac{\int x\mu(x)dx}{\int \mu(x)dx}$$

El punto de equilibrio obtenido se utiliza para representar a los conjuntos difusos de salida.

Método del centro de gravedad. Este método determina el centro del área perteneciente a la combinada función de membresía. Esta operación es computacionalmente compleja y por lo tanto resultan ciclos lentos de inferencia. El punto representativo de los conjuntos de salida se determina con la expresión:

$$z^* = \frac{\sum z_i \mu(z_i)}{\sum \mu(z_i)}$$

Método de altura máxima. Consiste en realizar un barrido de todos los elementos de los conjuntos difusos de salida, para localizar el elemento con mayor grado de pertenencia. El elemento obtenido se utiliza para representar los conjuntos difusos de salida. Al realizar un barrido de todos los elementos de los conjuntos difusos de la figura 1.8 se determina que el elemento con mayor grado de pertenencia es z^* .

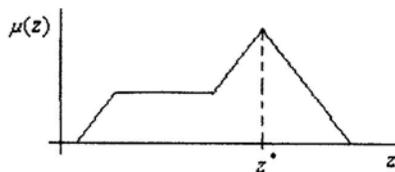


Figura 1.8 Desdifusión con el método de altura máxima.

Método de primero de los máximos. Se realiza un barrido de izquierda a derecha, de los elementos de los conjuntos difusos, hasta obtener el primer elemento con valor de pertenencia máximo, el cual se utiliza para representar a los conjuntos difusos de salida. Al realizar un barrido de los elementos de los conjuntos difusos de salida en la figura 1.9, se determina que el primer valor de pertenencia máximo se encuentra en el elemento z^* .

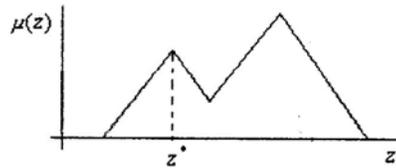


Figura 1.9 Desdifusión con el método de primero de los máximos.

Método de último de los máximos. Se realiza un barrido de izquierda a derecha, de los elementos de los conjuntos difusos hasta obtener el último valor de pertenencia máximo, el cual se utiliza para representar a los conjuntos difusos de salida. Al realizar un barrido de los elementos de los conjuntos difusos de salida de la figura 1.10, se determina que el último valor de pertenencia máximo se encuentra en el elemento z^* .

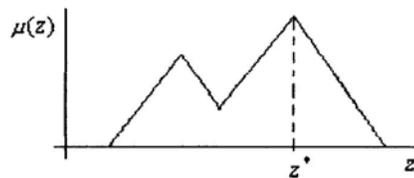


Figura 1.10 Desdifusión con el método de último de los máximos.

Método de promedio pesado. Se aplica cuando los conjuntos difusos de salida están formados por figuras simétricas. Para aplicarlo se siguen los siguientes pasos:

1. Se identifican las figuras simétricas.
2. Se obtiene el punto $(z_i, \mu(z_i))$, localizado sobre el eje de simetría de cada figura.
3. Se aplica la expresión:

$$z^* = \frac{\sum z_i \mu(z_i)}{\sum \mu(z_i)}$$

En los conjuntos difusos de salida de la figura 1.11 se tienen dos figuras simétricas, las cuales se representan con los puntos $(z_1, \mu(z_1))$ y $(z_2, \mu(z_2))$ ubicados en los ejes de simetría de cada una. El valor representativo de los conjuntos difusos se calcula como

$$z^* = \frac{z_1 \mu(z_1) + z_2 \mu(z_2)}{\mu(z_1) + \mu(z_2)}$$

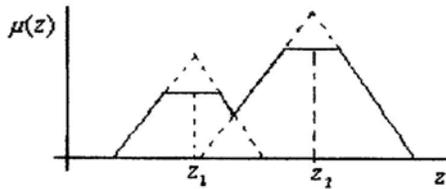


Figura 1.11 Desdifusión con el método de promedio pesado.

Método de centro de sumas. Para aplicar este método se siguen los siguientes pasos:

1. Se determina un punto representativo z_i para cada conjunto difuso.
2. Se calcula el área A_i de cada conjunto difuso.
3. Se obtiene un punto representativo de todos los conjuntos difusos con la expresión

$$z = \frac{\sum_{i=1}^n z_i A_i}{\sum_{i=0}^n A_i}$$

En la figura 1.12 se tiene que z_1 y z_2 son los puntos representativos de los conjuntos difusos, donde A_1 y A_2 representan las áreas de cada conjunto, al aplicar el método de centro de sumas se obtiene que

$$z^* = \frac{z_1 A_1 + z_2 A_2}{A_1 + A_2}$$

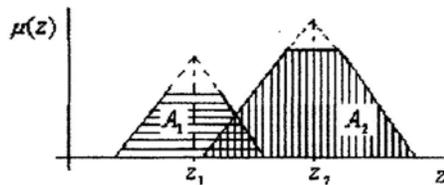


Figura 1.12 Desdifusión con el método de centro de sumas.

Método de promedio de máximos. Consiste en realizar un barrido de todos los elementos de los conjuntos difusos para localizar el primero y el último elemento con mayor grado de pertenencia, representados como z_1 y z_2 respectivamente. El valor representativo z^* de los conjuntos difusos se calcula con la expresión

$$z^* = \frac{z_1 + z_2}{2}$$

Al realizar un barrido de todos los elementos del conjunto difuso de la figura 1.13, se encontró el primer valor máximo de pertenencia en el elemento z_1 y el último valor máximo de pertenencia en el elemento z_2 . El valor representativo de los conjuntos se calcula como el valor promedio de estos dos elementos.

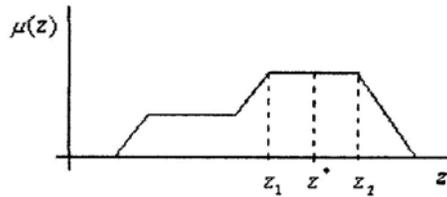


Figura 1.13 Desdifusión con el método de promedio de máximos.

Método de centro de área mayor. Para aplicar este método se siguen los siguientes pasos:

1. Se identifican los conjuntos convexos que forma la figura.
2. Se calcula el área de cada conjunto convexo.
3. Se comparan las áreas.
4. Se obtiene el centroide del área mayor.

En la figura 1.14 se tienen los conjuntos convexos A y B , cuyas áreas son A_1 y A_2 respectivamente. Si al comparar las áreas se obtiene que $A_1 > A_2$, entonces el punto representativo de los conjuntos difusos se obtiene con el centroide del conjunto convexo de área mayor con la expresión

$$z^* = \frac{\int z_{A1} \mu_{A1}(z) dz}{\int \mu(z)_{A1} dz}$$

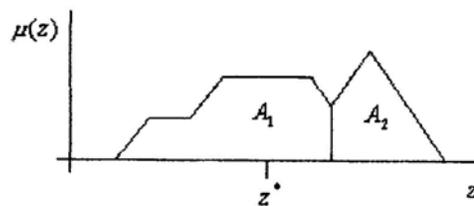


Figura 1.14 Desdifusión con el método de centro de área mayor.

1.2.9 Aplicación de control con Lógica Difusa.

En general, la lógica difusa proporciona definiciones de conjuntos que tienen límites difusos (fuzzy) en lugar de límites nítidos (crisp) de la lógica de Aristóteles. Estos conjuntos pueden traslaparse de modo que, para un valor específico de entrada, uno o más conjuntos asociados con etiquetas lingüísticas puede ser verdad hasta cierto grado al mismo tiempo. Al variar la entrada desde el rango de un conjunto dentro del rango de un conjunto adyacente, el primer conjunto llega a ser progresivamente menos verdadero mientras que el segundo conjunto llega a ser progresivamente más verdadero.

La lógica difusa tiene funciones de pertenencia que emulan conceptos humanos como "la temperatura es tibia"; esto es, condiciones que son percibidas con límites graduales. Este concepto parece ser un elemento clave

de la habilidad humana para resolver ciertos tipos de problemas complejos que han eludido a los métodos tradicionales de control.

Los conjuntos difusos brindan un medio para usar expresiones lingüísticas como “la temperatura es tibia” en reglas que pueden entonces ser evaluadas con un alto grado de precisión numérica y repetidamente. Esto contradice directamente la percepción errónea tan común de que la lógica difusa produce resultados aproximados; un conjunto específico de condiciones de entrada siempre produce el mismo resultado, justamente como un sistema de control convencional lo hace.

Un sistema de control de lógica difusa basado en microcontrolador tiene dos partes. La primera parte es un núcleo de inferencia difusa el cual es ejecutado periódicamente para determinar las salidas del sistema basados en las entradas actuales del sistema. La segunda parte del sistema es una base de conocimiento la que contiene funciones de pertenencia y Reglas. La figura 1.15 es un diagrama a bloques de esta clase de sistema de lógica difusa.

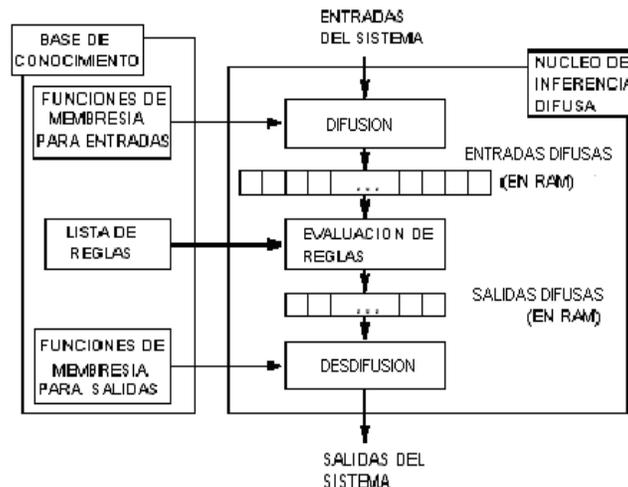


Figura 1.15 Diagrama a Bloques de un Sistema de Lógica Difusa.

La base de conocimiento puede ser desarrollada por un experto en la aplicación sin ninguna experiencia en programación de microcontroladores. Las funciones de pertenencia son simplemente expresiones del entendimiento del experto de los términos lingüísticos que describen al sistema a ser controlado.

1.3 Diferencias entre la Lógica Difusa y los métodos convencionales de control.

La Lógica Difusa es un paradigma para una metodología de diseño alternativa, la cual puede ser aplicada en el desarrollo de sistemas lineales y no-lineales de control incrustado. Usando la Lógica Difusa, los diseñadores pueden obtener menores costos de desarrollo, mayores características y un mejor desempeño del producto final. Además, los productos pueden ser comercializados más rápido y con un costo efectivo.

Una metodología de diseño alternativa, más simple y rápida.

Para apreciar porque una metodología de diseño del tipo Difusa es muy atractiva para las aplicaciones de control incrustado, examinemos el flujo de diseño típico. La figura 1.16 ilustra la secuencia de pasos de diseño requeridos para el desarrollo de un controlador utilizando una perspectiva convencional y una Difusa.

METODOLOGIA DEL DISEÑO CONVENCIONAL	METODOLOGIA DEL DISEÑO BASADO EN LOGICA DIFUSA
ENTENDER EL SISTEMA FISICO Y LOS REQUERIMIENTOS DE CONTROL	ENTENDER EL SISTEMA FISICO Y LOS REQUERIMIENTOS DE CONTROL
DISEÑAR UN MODELO LINEAL DE PLANTA, SENSORES Y ACTUADORES	DISEÑAR EL CONTROLADOR USANDO REGLAS DIFUSAS
DETERMINAR UN CONTROLADOR SIMPLIFICADO DE LA TEORIA DE CONTROL	
DESARROLLAR UN ALGORITMO PARA EL CONTROLADOR	
SIMULAR, DEPURAR E IMPLEMENTAR EL DISEÑO	SIMULAR, DEPURAR E IMPLEMENTAR EL DISEÑO

Figura 1.16 Diseño Convencional y Diseño Difuso.

Usando la perspectiva convencional, nuestro primer paso es entender el sistema físico y sus requerimientos de control. Con este conocimiento, el segundo paso es desarrollar un modelo que incluya la planta, los sensores y los actuadores. El tercer paso es usar la teoría de control lineal para obtener una versión simplificada del controlador, como los parámetros de un controlador PID. El cuarto paso es desarrollar un algoritmo para el controlador simplificado. El último paso es simular el diseño incluyendo los efectos de no-linealidad, ruido, y variación en los parámetros. Si el desempeño no es satisfactorio, es necesario modificar el modelado del sistema, rediseñar el controlador, re-escribir el algoritmo y volver a probar.

Con la Lógica Difusa el primer paso es entender y caracterizar el comportamiento del sistema usando nuestro conocimiento y experiencia. El segundo paso es directamente diseñar el algoritmo de control usando reglas difusas, las cuales describen los principios de regulación del controlador en términos de la relación entre sus entradas y salidas. El último paso es simular y depurar el diseño. Si el desempeño no es satisfactorio, solo es necesario modificar algunas reglas difusas y volver a probar.

Aunque las dos metodologías son similares, la basada en Lógica Difusa substancialmente simplifica el ciclo de diseño. Esto produce algunos beneficios, como reducción del tiempo de desarrollo, diseño simple y un tiempo de producción mas corto.

➤ **La Lógica Difusa reduce el ciclo del desarrollo del diseño.**

Con una metodología de diseño con lógica difusa, algunos pasos que consumen mucho tiempo son eliminados. Es más, durante el ciclo de depuración y puesta a punto puede cambiar sus sistemas simplemente modificando las reglas, en lugar de rediseñar el controlador. Además, dado que la lógica difusa está basada en reglas, no necesita ser un experto en un lenguaje de alto o bajo nivel, lo cual le permite enfocarse más en su aplicación en lugar de la programación. Como resultado, la Lógica Difusa reduce substancialmente el ciclo de desarrollo.

➤ **La Lógica Difusa simplifica la complejidad del diseño.**

La Lógica Difusa le permite describir sistemas complejos usando su conocimiento y experiencia, para producir reglas sencillas. No requiere ningún sistema de modelado o ecuaciones matemáticas complejas gobernando la relación entre las entradas y las salidas. Las reglas Difusas son muy sencillas de aprender y usar, aun para no expertos. Típicamente se requieren solo unas pocas reglas para describir sistemas que pueden requerir varias líneas de software convencional. Como resultado, la Lógica Difusa simplifica significativamente la complejidad del diseño.

➤ **La Lógica Difusa mejora el tiempo de comercialización.**

Las aplicaciones comerciales de control incrustado requieren un esfuerzo de desarrollo significativo mucho del cual es gastado en la porción de software del proyecto. El tiempo de desarrollo es una función de la complejidad del diseño y del número de iteraciones requeridas en los ciclos de depuración y puesta a punto. Como se explicó anteriormente, una metodología de diseño usando Lógica Difusa resuelve ambas situaciones de manera efectiva. Es más, dada su simplicidad, la descripción de un controlador difuso no solo es transportable a través de equipos de diseño, sino que proporciona un medio excelente para preservar, mantener y actualizar propiedad intelectual. Como resultado, la Lógica Difusa puede dramáticamente mejorar el tiempo para la comercialización del producto.

1.4 Ventajas de la utilización de la Lógica Difusa sobre los métodos convencionales de control.

La Lógica Difusa incorpora una perspectiva simple, basada en reglas del tipo IF X AND Y THEN Z para resolver problemas de control, más que intentar modelar un sistema matemáticamente. El modelo de la Lógica Difusa es empíricamente construido, confiando en la experiencia del operador más que en su conocimiento técnico del sistema. Por ejemplo, más que tratar con la temperatura de control en términos de "SP=500F", "T < 1000F", o "210C < TEMP < 220C", términos como "IF (el proceso es demasiado frío) AND (el proceso se está enfriando) THEN (agrega calor al proceso)" o "IF (el proceso es demasiado caliente) AND (el proceso se está calentando rápidamente) THEN (enfriá el proceso rápidamente)" son usados. Estos términos son

imprecisos y aun así muy descriptivos de que es lo que debe hacerse. Considere lo que usted realiza en la regadera si la temperatura es muy baja: usted hará del agua agradable muy rápidamente sin problema. La Lógica Difusa es capaz de imitar este comportamiento pero a una alta velocidad.

La mayoría de los sistemas físicos reales son en realidad sistemas no lineales. Las perspectivas de diseño convencional usan diferentes métodos de aproximaciones para manejar la no-linealidad. Algunas opciones típicas son las aproximaciones lineales, fragmentos de linealidad y tablas de búsqueda para compensar los factores de complejidad, costo y desempeño del sistema.

Una técnica de aproximación lineal es relativamente simple, sin embargo tiende a limitar el desempeño del control y puede ser costoso para ser implementado en ciertas aplicaciones. La técnica de fragmentos de linealidad funciona mejor, aunque es tediosa para ser implementada porque a menudo requiere del diseño de varios controladores lineales. La técnica de la tabla de búsqueda puede ayudar a mejorar el desempeño del control, pero es difícil de depurar y poner a punto. Además en sistemas complejos donde múltiples entradas existen, una tabla de búsqueda puede ser impráctica o muy costosa de implementar debido a sus requerimientos de gran cantidad de memoria. La Lógica Difusa proporciona una solución alternativa para el control no lineal porque se acerca más al mundo real. La no linealidad es manejada por reglas, funciones de membresía y los procesos de inferencia, los cuales producen un desempeño mejorado, una implementación sencilla y reduce los costos de diseño.

➤ **La Lógica Difusa mejora el desempeño del control.**

En muchas aplicaciones la Lógica Difusa puede resultar en mejor desempeño del control que las técnicas de linealidad, fragmentos de linealidad o tablas de búsqueda. Por ejemplo, un problema típico asociado con las técnicas tradicionales es compensar el tiempo de respuesta del controlador contra el sobre-control (overshoot). Por ejemplo un controlador de temperatura de una entrada ejemplifica esto y se ilustra en la figura 1.17.

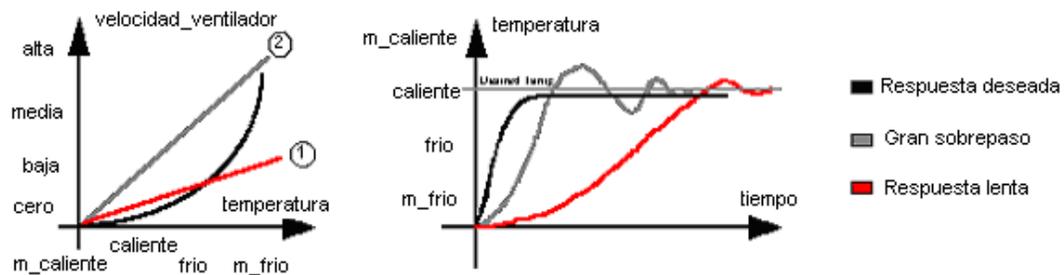


Figura 1.17 Sobrepaso contra tiempo de respuesta.

La primera aproximación lineal para la curva deseada genera una respuesta de salida lenta sin sobre-control ni subsecuentes fluctuaciones, lo cual implica que la temperatura será confortable por un periodo de tiempo.

Con la Lógica Difusa podemos usar reglas y funciones de membresía para aproximar cualquier función continua a cualquier grado de precisión. La figura 1.18 ilustra como podemos aproximar la curva de control deseada para nuestro controlador de temperatura usando cuatro puntos (o reglas). También podemos agregar mas reglas para incrementar la exactitud de la aproximación (similar a una transformada de Fourier), lo cual apunta para un mejor desempeño del control. Las reglas son mucho más sencillas de implementar y mucho más fáciles de depurar y poner a punto que las técnicas de fragmentos de linealidad y tablas de búsqueda.

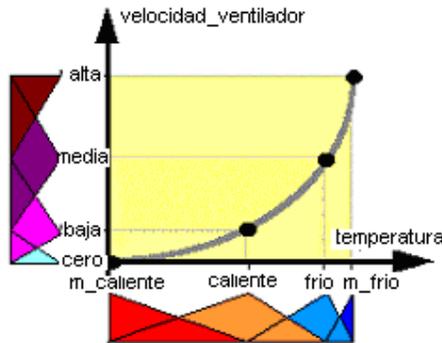


Figura 1.18 Reglas y funciones de aproximación de una función no-lineal.

- IF *temperatura* IS *m_frio* THEN fuerza IS *alta*
- IF *temperatura* IS *frio* THEN fuerza IS *media*
- IF *temperatura* IS *caliente* THEN fuerza IS *baja*
- IF *temperatura* IS *m_caliente* THEN fuerza IS *cero*

Las reglas no son como una tabla de búsqueda porque la aritmética difusa interpola la forma de la función no lineal. La memoria combinada requerida para las etiquetas y la inferencia difusa es substancialmente menor que la requerida por una tabla de búsqueda, especialmente cuando se trata de sistemas de múltiples entradas. Como resultado, la velocidad de procesamiento puede ser mejorada también.

Otro ejemplo de un control sólido que puede ser conseguido con la Lógica Difusa es el problema clásico del péndulo invertido. Un controlador convencional para el péndulo depende en parámetros de sistema tales como longitud, peso, y masa. Si los parámetros cambian, entonces necesitamos rediseñar nuestro controlador. Se ha demostrado prácticamente con dispositivos reales donde se puede variar el peso o la longitud del péndulo y el sistema continua estable usando el mismo conjunto de reglas originales.

Usando una perspectiva más natural basada en reglas, la cual es mas cercana al mundo real, el control Difuso puede ofrecer un mejor desempeño y una mejor compensación entre la solidez del sistema y su sensibilidad, lo cual resulta en que el control para sistemas no lineales, mejor que los métodos convencionales.

➤ **La Lógica Difusa simplifica la implementación.**

El controlador de una entrada presentado hasta el momento ha contribuido a ilustrar algunos conceptos fundamentales, sin embargo el control en la vida real es mucho más complejo. La mayoría de las aplicaciones de control tienen múltiples entradas y requiere modelado y puesta a punto de un gran número de parámetros lo cual hace a la implementación muy tediosa y consumidora de tiempo. Las reglas Difusas pueden ayudar a simplificar la implementación combinando múltiples entradas en una sola sentencia del tipo if-then y conservar la no-linealidad.

Considere una versión modificada del controlador de temperatura del ejemplo, con dos entradas, la temperatura y la humedad y la misma salida, la velocidad de ventilador, figura 1.19. Este ejemplo puede ser descrito con un pequeño conjunto de reglas:

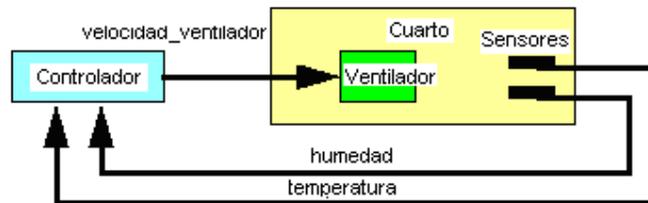


Figura 1.19 Controlador de Temperatura modificado.

- IF temperatura IS m _ fría AND humedad IS alta THEN vel_vent IS alta
- IF temperatura IS fría AND humedad IS alta THEN vel_vent IS media
- IF temperatura IS caliente AND humedad IS alta THEN vel_vent IS baja
- IF temperatura IS m_caliente AND humedad IS alta THEN vel_vent IS cero
- IF temperatura IS m _ fría AND humedad IS media THEN vel_vent IS media
- IF temperatura IS fría AND humedad IS media THEN vel_vent IS baja
- IF temperatura IS caliente AND humedad IS media THEN vel_vent IS cero
- IF temperatura IS m_caliente AND humedad IS media THEN vel_vent IS cero
- IF temperatura IS m _ fría AND humedad IS baja THEN vel_vent IS media
- IF temperatura IS fría AND humedad IS baja THEN vel_vent IS baja
- IF temperatura IS caliente AND humedad IS baja THEN vel_vent IS cero
- IF temperatura IS m_caliente AND humedad IS baja THEN vel_vent IS cero

Una aproximación lineal requiere el manejo de cada entrada de manera separada, lo cual multiplica el esfuerzo del diseño. De manera similar, la perspectiva de fragmentos de linealidad requiere de varios controladores y es costoso de implementar. Una tabla de búsqueda parece mas apropiado para este problema pero requiere tiempo para ser desarrollado, depurado y puesto a punto. Por ejemplo, si asumimos que cada entrada requiere 8 bits, una tabla de búsqueda requiere 64k entradas, lo cual vuelve a la implementación muy costosa en tiempo, figura 1.20.

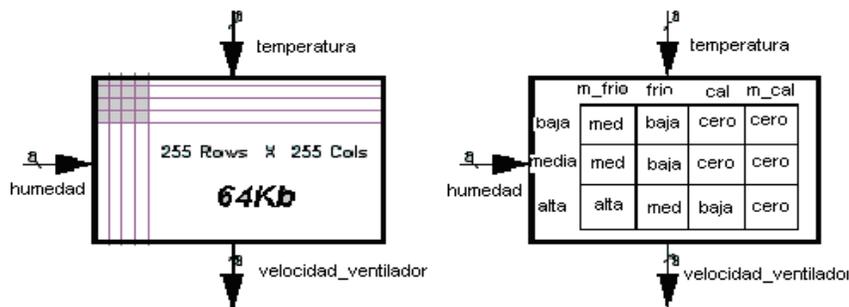


Figura 1.20 Tabla de localización contra reglas y funciones de membresía.

Usando la Lógica Difusa podemos describir la salida como una función de dos o más entradas relacionadas con operadores lógicos como AND. Esta relación también puede ser representada en forma de una tabla. La perspectiva difusa requiere significativamente menos entradas que la búsqueda en una tabla dependiendo del número de etiquetas para cada variable de entrada. Las reglas son mucho más fáciles de desarrollar, más simples de depurar y poner a punto comparados contra la búsqueda en una tabla.

Otro ejemplo de simplicidad es el problema clásico de control del péndulo invertido de dos etapas. Usando programación convencional, este problema es extremadamente difícil o incluso imposible de implementar. Algunas empresas (como Apronix) ha mostrado un modelo físico del péndulo invertido de 2 etapas, el cual se consiguió con solo 30 reglas. La porción del software del proyecto tomó solo dos días en desarrollar.

➤ **La Lógica Difusa reduce el costo del hardware.**

Usando una tabla de búsqueda, el controlador de dos entradas requiere de 63k de memoria, mientras la solución Difusa es conseguida con menos de 0.5Kb de memoria combinada para las etiquetas y el código. Esta diferencia en ahorro de memoria implica una implementación en hardware más barata. Además, las técnicas convencionales en la mayoría de las aplicaciones para el mundo real requieren de análisis y modelado matemático complejo, algoritmos de punto flotante y complejas estructuras de control de flujo (branching). Esto conduce a un gran tamaño de código objeto que requiere de coprocesadores matemáticos para correr. La Lógica Difusa posibilita el uso de una perspectiva basada en reglas la cual ofrece ahorro de costos, tanto en memoria como en clase de procesador.

Como ejemplo, considere otra vez el péndulo invertido de 2 etapas, cuyo modelo es equivalente a una ecuación diferencial de segundo grado. Con la perspectiva tradicional, el modelo requiere de estaciones de trabajo de alto perfil para ser desarrollado, y el controlador es extremadamente difícil o muy costoso para ser implementado. Por ejemplo, una ecuación no-lineal requeriría un procesador de alto desempeño costoso para atender la intensidad en los cálculos, mientras que una tabla de búsqueda requeriría una gran cantidad de memoria. Con la Lógica Difusa el péndulo de 2 etapas ha sido desarrollado en una PC, corriendo en un micro-controlador de 8 bits de bajo costo y requiere de menos de 1Kb de memoria. Este logro demuestra que podemos usar hardware de bajo perfil y costo para atacar problemas de control complejos.

1.5 El uso de la Lógica Difusa.

- 1) Definir los criterios y objetivos de sistema de control: ¿Que estoy intentando controlar? ¿Que tengo que hacer para controlar el sistema? ¿Cuáles son las posibles (probables) modos de falla del sistema?
- 2) Determinar las relaciones de entrada y salida y elegir un numero de variables mínimas para la entrada a el motor de Lógica Difusa (típicamente el error y la rapidez en la variación del error).
- 3) Usando la estructura basada en reglas de la Lógica Difusa, romper el problema de control en una serie de reglas del tipo IF X AND Y THEN Z que definan la respuesta de salida deseada del sistema para determinadas condiciones de salida del sistema. El numero y complejidad de las reglas dependen de el numero de parámetros de entrada que van a ser procesadas y el numero de variables difusas asociadas con cada parámetro. Si es posible, use al menos una variable y su derivada en el tiempo. Aunque es posible usar solo un parámetro de error con una lectura instantánea sin conocer su velocidad de cambio, esto le quita al sistema su habilidad de minimizar el sobre control para alguna entradas.
- 4) Crear las funciones de membresía que definan el significado (valores) para los términos de entrada y salida usados en las reglas.
- 5) Crear las rutinas de procesamiento previas y posteriores si se implementa en software, en caso contrario, programar las reglas en el hardware usado.
- 6) Probar el sistema, evaluar los resultados, afinar las reglas y funciones de membresías y volver a probar hasta que resultados satisfactorios sean obtenidos.

Un sistema experto puede hacer uso de la Lógica Difusa. Los sistemas expertos son usados, por ejemplo, para diagnósticos médicos. Los sistemas expertos tradicionales usan valores concretos de valores lógicos para realizar un diagnostico o una recomendación basada en algunas evidencias. En muchos sentidos, esta es una forma elegante de aplicar el conocimiento de un experto. Por otro lado, la mayoría de las decisiones de los expertos no son blancas o negras. Un experto al que se le presenta un paciente con un grupo de síntomas no es capaz de proporcionar un diagnostico con absoluta certeza pero tendrá un fuerte presentimiento a cerca del diagnostico, basado en el peso de la evidencia.

Por eso, aplicar la Lógica difusa a estas reglas, parece ser una manera natural de avanzar.

Los sistemas expertos pueden ser construidos eligiendo un conjunto de variables lingüísticas apropiadas para el problema de definir funciones de membresía para esas variables. Las reglas son entonces generadas basadas en el conocimiento del experto y usando variables lingüísticas. Las reglas difusas pueden entonces ser aplicadas.

1.5.1 Ciclo de análisis y diseño usando Lógica Difusa.

➤ **Análisis**

1. Definir los Objetivos, las Restricciones y el Comportamiento del Sistema.
2. Identificar las variables de Entrada y Salida: Variables lingüísticas y sus etiquetas lingüísticas.
3. Definir procedimientos y criterios para la prueba y validación.

➤ **Diseño**

4. Definir los conjuntos difusos de cada etiqueta lingüística.
5. Definir el marco de representación de la solución (representación del conocimiento, reglas...).
6. Definir el marco de los procedimientos empleados (difusión, Inferencia y agregación).
7. Especificar la forma requerida de la salida del sistema (decodificación o desdifusión).
8. Ejecutar pruebas para validar el sistema.
9. Verificar si la solución es compatible con el paso 1:
 - Refinar el Diseño: Ir al paso 4.
 - Necesidad de Mejora: Ir al paso 1.

1.5.2 Sistema de Decodificación, Concisión o *Desdifusión*.

- Convierte los valores difusos de las variables de salida en valores concretos dentro del universo de discurso correspondiente.
- Genera una acción no difusa a partir de la acción difusa resultante del sistema de inferencia.
- Existen diversos métodos, estudiados anteriormente (*defuzzification methods*):
- Usando la agregación de todos los conjuntos difusos resultantes de todas las reglas: Centro de Gravedad (CoG), Punto de Máximo Criterio (PMCo Media de Máximos), Centro de Area...
- Usando los conjuntos difusos resultantes de cada regla individualmente: CoG (o PMC) ponderado por el área o por la altura, Media de PMC, Media del mínimo y máximo PMC, CoG (o PMC) del de mayor área o del de mayor altura...

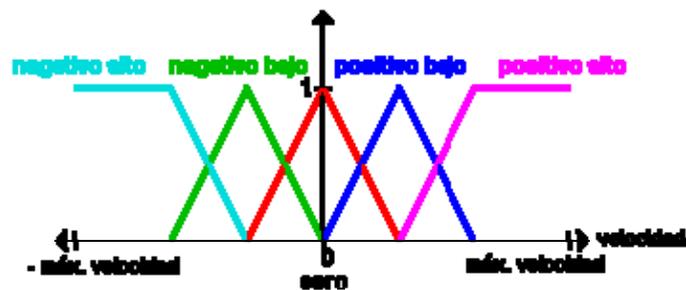
Los controladores difusos son las aplicaciones más importantes de la teoría difusa. Ellos trabajan de una forma bastante diferente a los controladores convencionales; el conocimiento experto se usa en vez de ecuaciones

diferenciales para describir un sistema. Este conocimiento puede expresarse de una manera muy natural, empleando las variables lingüísticas que son descritas mediante conjuntos difusos.

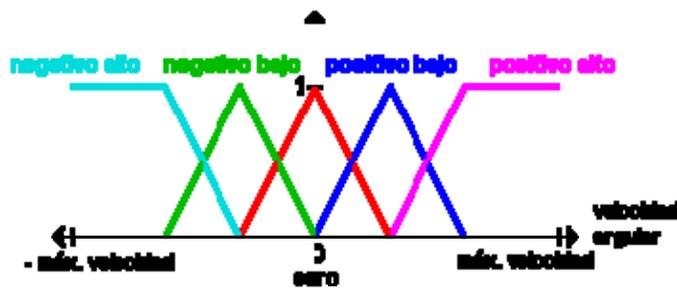
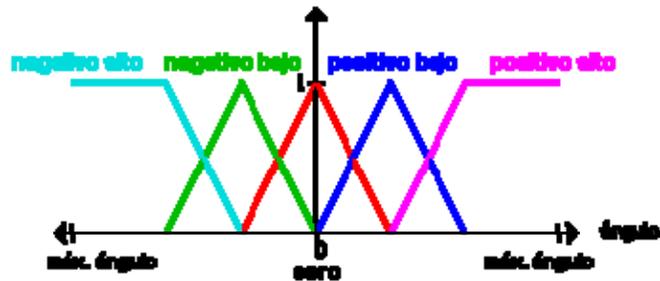
1.5.3 Ejemplo: El péndulo invertido.

El problema está en equilibrar una pértiga sobre una plataforma móvil que puede moverse en dos únicas direcciones, a la izquierda o a la derecha. Ante todo, nosotros tenemos que definir (subjetivamente) cual es la velocidad del andén: alta, baja, etc. Esto se hace para especificar las funciones pertenecientes al conjunto difuso:

- negativo alto (celeste)
- negativo bajo (verde)
- cero (rojo)
- positivo bajo (azul)
- positivo alto (morado)



Lo mismo se hace para el ángulo entre la plataforma y la pértiga, además de para la velocidad angular de este ángulo:



Apréciase que, para hacerlo más fácil, suponemos que al principio la pértiga está en una posición cercana a la central para que un ángulo mayor de, digamos, 45 grados en cualquier dirección no pueda, por definición, ocurrir.

Ahora daremos varias reglas que dicen qué hacer en situaciones concretas: Considere por ejemplo que la pértiga está en la posición central (el ángulo es cero) y no se mueve (la velocidad angular es cero). Obviamente esta es la situación deseada, y por lo tanto no tenemos que hacer nada (la velocidad es cero).

Consideremos otro caso: el polo está en la posición central como antes, pero está en movimiento a baja velocidad en la dirección positiva. Naturalmente nosotros tendríamos que compensar el movimiento de la pértiga moviendo la plataforma en la misma dirección a baja velocidad.

De esta forma hemos constituido dos reglas que pueden ponerse en una forma más formalizada como esta:

- Si el ángulo es cero y la velocidad angular es cero entonces la velocidad será cero.
- Si el ángulo es cero y la velocidad angular es positiva baja entonces la velocidad será positiva baja.

Podemos resumir todas las reglas aplicables en una tabla:

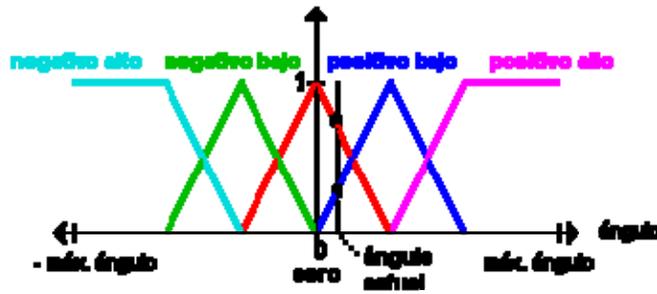
	ángulo				
veloc	NA	NB	C	PB	PA
v	NA		NA		
.	NB		NB	C	
a	C	NA	NB	C	PB PA
n	PB		C	PB	
g	PA		PA		

donde NA es una usual abreviatura para negativa alta, NB para negativa baja, etc.

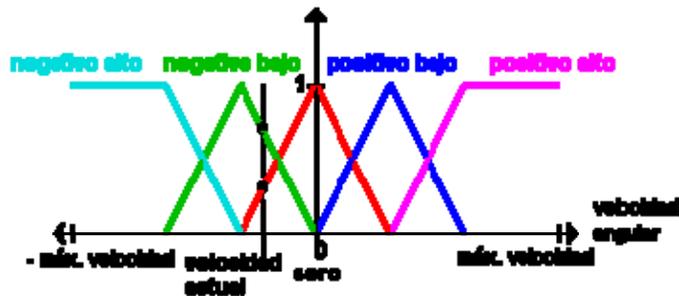
A continuación mostraremos como estas reglas pueden aplicarse con valores concretos para el ángulo y velocidad angular. Para ello vamos a definir dos valores explícitos para el ángulo y la velocidad angular para operar con ellos.

Consideremos la situación siguiente:

Un valor actual para el ángulo:



Un valor actual para la velocidad angular:

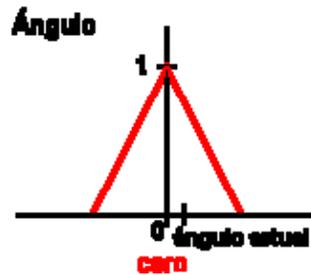


Ahora mostraremos como aplicar nuestras reglas a esta situación real. Veamos como aplicar la regla.

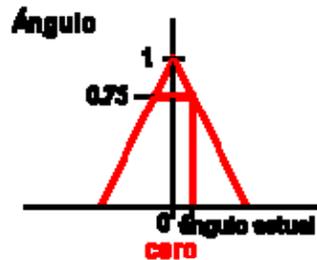
Si el ángulo es cero y la velocidad angular es cero entonces la velocidad será cero.

A los valores que hemos definido.

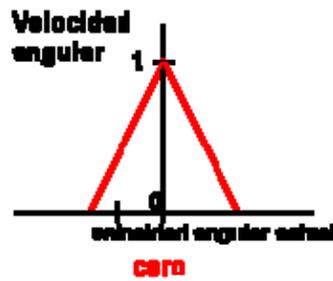
Esta es la variable lingüística "ángulo" donde nos centramos en el conjunto "cero" y el ángulo actual:



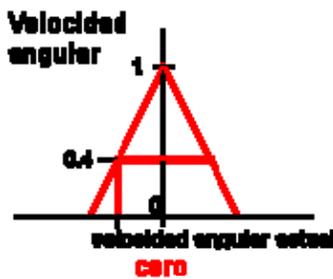
Nos damos cuenta que nuestro valor real pertenece al conjunto difuso "cero" en un grado de 0.75:



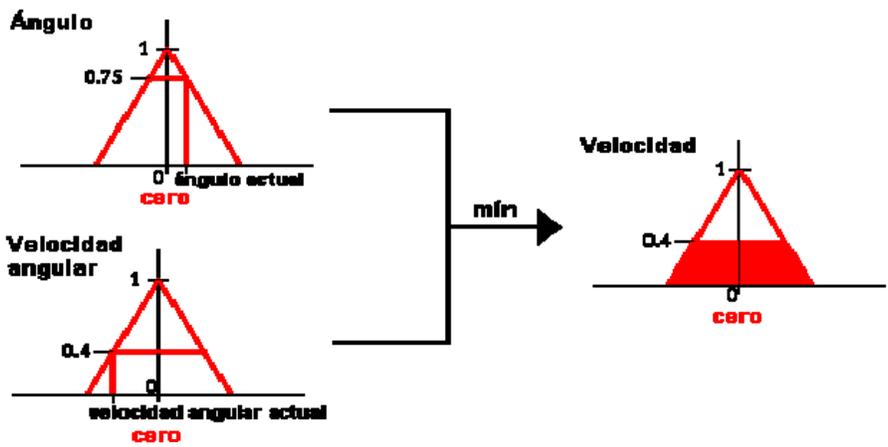
Ahora mostramos la variable lingüística "velocidad angular" donde nos centramos en el conjunto difuso "cero" y el valor actual de velocidad angular:



Nos damos cuenta que nuestro valor real pertenece al conjunto difuso "cero" en un grado de 0.4:



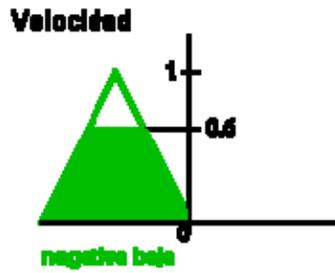
Como las dos partes de la condición de nuestra regla están unidas por una Y (operación lógica AND) calculamos el $\min(0.75, 0.4) = 0.4$ y cortamos el conjunto difuso "cero" de la variable "velocidad" a este nivel (según nuestra regla):



Por su parte, el resultado de la regla:

Si el ángulo es cero y la velocidad angular es negativa baja entonces la velocidad será negativa baja.

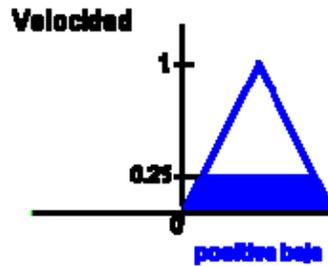
Es:



El resultado de la regla:

Si el ángulo es cero y la velocidad angular es positiva baja entonces la velocidad será positiva baja.

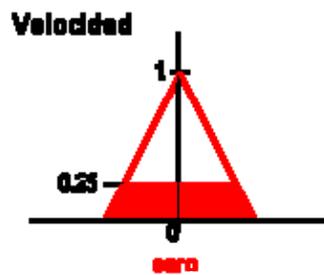
Es:



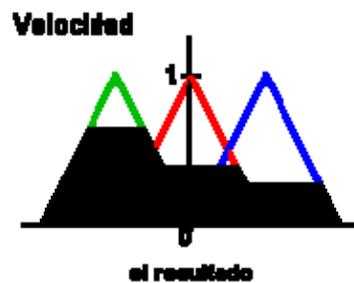
El resultado de la regla:

Si el ángulo es positivo bajo y la velocidad angular es negativa baja entonces la velocidad será cero.

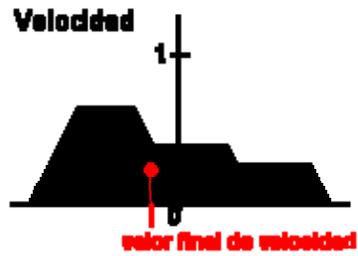
Es:



Estas cuatro reglas solapadas desembocan en un resultado único:



El resultado del controlador difuso es un conjunto difuso (de velocidad), así que tenemos que escoger un valor representativo como salida final. Hay varios métodos heurísticos (métodos de claridad o desfusión), uno de ellos es tomar el centro de gravedad del conjunto difuso:



El procedimiento completo se denomina controlador Mamdani.

CAPÍTULO 2. MOTORES DE CORRIENTE DIRECTA

El motor mismo es el fundamento de toda industria y sus principios básicos nos acercan al origen de todo movimiento, fuerza y velocidad. Un electromotor transforma la energía eléctrica en energía mecánica, éste es el concepto básico de los equipos que en este proyecto trataremos de desarrollar.

Los motores de corriente directa se utilizan en casos en los que es de importancia el poder regular continuamente la velocidad del eje y en aquellos casos en los que se necesita de un toque de arranque elevado, estos también se utilizan cuando es necesario cambiar el sentido de giro. Además, se utilizan en aquellos casos en los que es imprescindible utilizar corriente continua, como es el caso de trenes y automóviles eléctricos, motores para utilizar en el arranque y en los controles de automóviles, motores accionados a pilas o baterías, etc. Para funcionar, el motor de corriente continua o directa precisa de dos circuitos eléctricos distintos: el circuito de campo magnético y el circuito de la armadura.

Este tipo de motores eléctricos basa su funcionamiento en la reversibilidad de un generador de CC. (Dinamo). El movimiento de un conductor o espira dentro de un campo magnético engendra en él una corriente inducida, cuyo sentido depende del que rija el movimiento de la espira. Esto se consigue haciendo girar mecánicamente un campo magnético. Si por el contrario aportamos una corriente continua a un conductor o espira inmersa en un campo magnético, nace en él un movimiento cuyo sentido depende también del sentido del campo y del sentido de la corriente que atraviesa el conductor.

De este principio básico se deduce que si a un generador de CC, le aplicamos una fuerza mecánica (Rotatoria), obtendremos energía eléctrica. Si por el contrario la aplicamos al mismo generador una CC, obtendremos energía mecánica.

El campo (básicamente un imán o un electroimán) permite la transformación de energía eléctrica recibida por la armadura en energía mecánica entregada a través del eje. La energía eléctrica que recibe el campo se consume totalmente en la resistencia externa con la cual se regula la corriente del campo magnético. Es decir, ninguna parte de la energía eléctrica recibida por el circuito del campo, es transformada en energía mecánica. El campo magnético actúa como una especie de catalizador que permite la transformación de energía en la armadura.

La armadura consiste en un grupo de bobinados alojados en el rotor y en un ingenioso dispositivo denominado colector mediante el cual se recibe corriente continua desde una fuente exterior y se convierte la correspondiente energía eléctrica en energía mecánica que se entrega a través del eje del motor. En la transformación se pierde un pequeño porcentaje de energía en los carbones del colector, en el cobre de los bobinados, en el hierro (por corrientes parásitas e histéresis), en los rodamientos del eje y la fricción del rotor por el aire.

El motor de CD es básicamente un transductor de par que convierte la energía eléctrica en energía mecánica. El par desarrollado por el eje del motor es directamente proporcional al flujo en el campo y a la corriente en la armadura. Un conductor que lleva corriente está colocado en un campo magnético con flujo ϕ , a una distancia r del centro de rotación. La relación entre el par desarrollado, el flujo ϕ y la corriente i_a es:

$$T_m = K_m \phi i_a$$

En donde T_m es el par del motor (N-m, lb-pie, u oz.-plg.), ϕ es el flujo magnético (webers), i_a es la corriente de armadura (amperes), y K_m es la constante de proporcionalidad.

Además del par desarrollado por el arreglo del motor, cuando el conductor se mueve en el campo magnético se genera un voltaje entre sus terminales. Este voltaje da la fuerza contraelectromotriz, la cual es proporcional a la velocidad del eje, tiende a oponerse al flujo de la corriente. La relación entre la fuerza contraelectromotriz y la velocidad del eje es:

$$E_c = K_m \phi W_m$$

En donde E_c denota la fuerza contraelectromotriz (volts), y W_m es la velocidad del eje (rad/s) del motor. Estas dos ecuaciones forman la base de la operación del motor de CD.

2.1 Conceptos básicos de los motores de CD.

Se toman en consideración conceptos básicos para la comprensión del funcionamiento del motor de cd, y como se obtienen las dos ecuaciones anteriores, de acuerdo al tipo de motor.

Fuerza. La fuerza es una tendencia a causar movimiento, como la presión sobre una superficie, la tensión de un cable o la atracción de un imán. No es necesario que genere movimiento para hacer fuerza. Se mide en dinas en el sistema cgs, en libras en sistema inglés y en Newtons en el SI.

$$\begin{aligned} 1 \text{ libra (lb)} &= 4.4482 \text{ Newtons} \\ 0.22481 \text{ Lb} &= 1 \text{ Newton (N)} \end{aligned}$$

Trabajo. El trabajo es una fuerza ejercida a lo largo de una distancia. Al considerar mecánicamente el trabajo debe ocurrir un movimiento, como en un motor eléctrico. La unidad cgs del trabajo es el ergio o dina-centímetro. En el sistema inglés se emplea la libra-pie (lb-ft) y en el SI se utiliza el newton-metro (N m), que también recibe el nombre de Joule (J).

$$\begin{aligned} 1 \text{ ft-lb} &= 1.3558 \text{ newtons-metro (N m)} \\ 0.73755 \text{ ft-lb} &= 1 \text{ (N m)} \end{aligned}$$

Potencia. El concepto de potencia es la rapidez con la que se hace el trabajo. La unidad original de potencia se obtuvo a partir del cálculo del caballo de fuerza (hp) hecho por James Watt. La unidad básica de potencia eléctrica recibió el nombre de Watt, quien definió 1 hp como 550 lb-ft/s o 33 000 lb-ft/min.

$$\begin{array}{rclclcl}
 1 \text{ ft-lb/s} & = & 1.8182 \cdot 10^{-3} \text{ hp} & = & 1.3558 \text{ N m/s} & = & 1.3558 \text{ W} \\
 550 \text{ ft-lb/s} & = & 1 \text{ hp} & = & 745.70 \text{ N m/s} & = & 745.70 \text{ W} \\
 0.73755 \text{ ft-lb/s} & = & 1.341 \cdot 10^{-3} \text{ hp} & = & 1 \text{ N m/s} & = & 1 \text{ W} \\
 737.55 \text{ ft-lb/s} & = & 1.341 \text{ hp} & = & 1000 \text{ N m/s} & = & 1 \text{ kW}
 \end{array}$$

2.1.1 Relación entre el par y la potencia.

Al medir la potencia en hp o en kW, tenemos que el par o efecto rotatorio, es una fuerza que actúa en ángulo recto a una distancia radial de un centro de rotación. Una libra que actúa a un brazo de palanca de 1 ft produce un par de 1 lb-ft, 1 N que actúa con un brazo de palanca de 1 m produce un par de 1 N m.

Cualquiera de estas fuerzas de par, al girar, deben moverse 2π veces el radio en una vuelta completa. Por lo tanto, 1 hp de potencia de rotación se puede deducir como:

F = fuerza en libras

D = distancia radial, en pies, de la acción de la fuerza

S = razón del movimiento de rotación en revoluciones/minuto

2π = factor para obtener radianes por revolución, aunque el radián no es una dimensión física

por lo tanto: 2π radianes/revolución

Por definición: 1 caballo de fuerza = 33 000 ft-lbs/min

Esto puede escribirse como:

$$1 \text{ hp min} / 33\,000 \text{ ft-lbs}$$

$$(F [\text{lbs}] D [\text{ft}] S [\text{rev}] 2\pi [\text{rad}] [\text{min}]) / ([\text{min}] [\text{rev}] 33\,000 [\text{ft-lbs}]) = 2\pi FDS / 33\,000 = \text{hp}$$

El empleo de radianes permite que el movimiento rotacional con respecto al tiempo sea equivalente al movimiento lineal con respecto al tiempo. Como el término radián es adimensional, desaparece.

Lo anterior se simplifica a $(FDS/5252.1) =$ caballo de fuerza, o bien, si el par T en libras-pie se sustituye en vez de fuerza F por distancia D , la fórmula se convierten

$$TS / 5252.1 = \text{caballos de fuerza}$$

Para el SI tenemos que la potencia de rotación en watts o en kilowatts, es

f = fuerza en newtons

d = distancia radial (brazo de palanca) de la acción de la fuerza en metros

w = rapidez de rotación en radianes por segundo, donde el radián es adimensional: radianes/segundo

Como la velocidad angular se da en radianes por unidad de tiempo, en el SI no se requiere el factor de conversión 2π . Por lo tanto, al combinar y eliminar términos se tiene

$$f \text{ [newtons]} \cdot d \text{ [metros]} \cdot w \text{ [radianes/segundo]} = f \cdot d \cdot w = \text{watts}$$

ya que $1 \text{ N} \cdot \text{m} / \text{s} = 1 \text{ W}$, y el término radian se elimina. Esto se convierte en

$$fdw \cdot 10^{-3} = \text{kilowatts}$$

si: $f \cdot d = t$ en newtons-metro

$$tw \cdot 10^{-3} = \text{kilowatts}$$

2.1.2 Producción del par.

Un motor eléctrico o cualquier otro dispositivo de producción de energía rotatoria, es una fuente de par rotatorio continuo, como se pueden poner en la ecuación el par y la velocidad angular para obtener potencia, es necesario saber cómo se produce el par y cómo se mide.

Las siguientes ecuaciones indican la fuerza sobre un conductor inmerso en un campo magnético.

$$F = (B I L \cdot 10^{-7}) / (1.14 \text{ lb})$$

$$F = B I \ell \text{ Newton}$$

Donde L se da en pulgadas y ℓ en metros.

Estas ecuaciones se relacionan con el par en una situación motriz efectuando las multiplicaciones siguientes:

- 1) por el número de conductores Z que hay en la armadura.
- 2) por el equivalente decimal del porcentaje de la cobertura efectiva del arco polar para hallar el número de conductores que están dentro del campo magnético, y por tanto que están funcionando.
- 3) después se multiplica por el radio efectivo D de los conductores en pies o d en metros, para convertir la fuerza en un par.
- 4) convirtiendo la corriente I , en amperes en la I_a corriente total de la armadura, y dividiendo entre a el número de trayectorias paralelas del devanado, lo que da la corriente real por conductor.

$$T = (B I_a L Z (\% \text{ cob.}) D \cdot 10^{-7}) / (1.13 a) \quad \text{ft-lb}$$

$$T = (B I_a I Z (\% \text{ cob.}) d) / a \quad \text{N m}$$

Estos valores del par se sustituyen en las ecuaciones se sustituyen en las ecuaciones para obtener la potencia bruta desarrollada.

2.1.3 Fuerza contraelectromotriz en un motor.

Cuando la armadura de un motor gira a consecuencia del par que se produce por la acción del motor, actúa como generador al mismo tiempo. Como debe haber un campo magnético desde los polos de campo a fin de que se produzca la acción del motor, ese mismo campo genera un voltaje en cualquier conductor de la armadura que pase ahí. El voltaje generado se opone a la corriente así producida por el voltaje aplicado que causa la acción del motor. A este voltaje generado que se opone al voltaje de línea aplicado se le conoce como contrafem o fuerza contraelectromotriz, E_c . La magnitud de la contrafem es función lineal de la velocidad de rotación si el flujo se mantiene constante. Conforme un motor aumenta su velocidad, su contrafem aumenta hasta ser una parte sustancial del voltaje de línea aplicado. La contrafem es el efecto benéfico necesario que regula la corriente de la armadura absorbida de las líneas.

El par que produce un motor es resultado lineal de la corriente de armadura que fluye si el flujo del campo se mantiene constante. La corriente I_a está en un principio limitada sólo por la resistencia del circuito de la armadura. Esta R_a es tan pequeña como se pueda hacer. En consecuencia, la corriente inicial que toma un motor de cd típico sería inaceptablemente grande si no se limitara por un voltaje de arranque pequeño o por una resistencia de arranque adicional.

La corriente de armadura de un motor inicialmente alta causa un par tan grande, debido a la acción del motor, que la armadura se acelera. Conforme la velocidad angular del motor aumenta, su contrafem también lo hace. En breve se llega a un punto en el que la diferencia entre la contrafem y el voltaje de línea se hace pequeña, reduciéndose la corriente de armadura. El equilibrio final se alcanza, por lo regular, en un lapso de unos pocos segundos, cuando la diferencia entre ambos voltajes permite que circule apenas la corriente necesaria para satisfacer el requerimiento del par la velocidad angular (en rpm o rad/s) alcanza un valor estable.

Por lo tanto, el voltaje del circuito de la armadura V_a menos la cotrafem E_c iguala exactamente la caída IR entre las terminales de la armadura, según la ley de ohm

$$V_a - E_c = I_a R_a$$

Como E_c se opone a la corriente de la armadura, tiene la misma polaridad que el voltaje de línea V . Si el motor girase tan rápido que su E_c fuera igual al voltaje de línea V , no fluiría corriente. Entonces el motor no tendría par y flotaría en la línea. V es probablemente igual a V_a , a menos que haya una caída de voltaje en la resistencia de arranque.

Otra relación útil al multiplicar la ecuación anterior por la corriente de equilibrio de la armadura es:

$$V_a I_a - E_c I_a = I_a^2 R_a$$

O, su equivalente
$$E_c I_a = V_a I_a - I_a^2 R_a$$

2.1.4 Potencia desarrollada por la armadura.

La ecuación anterior muestra que la potencia de la armadura desarrollada en el motor P_d , es igual a $E_c I_a$. Esto es así porque $V_a I_a$ es la potencia total dada por la armadura (voltaje por corriente), e $(I_a I_a R_a)$ son las pérdidas en el cobre del circuito de armadura. La resistencia efectiva R_a incluye todo en el circuito de la armadura: armadura, escobillas, campos de conmutación, campos en serie y devanado de compensación, si lo hay, así como cualquier resistencia de arranque o de control.

Cuanto menor sea R_a , menores serán las pérdidas de potencia.

La velocidad de rotación de un motor se autoajusta hasta que fluye la corriente suficiente para provocar un par. Si el par de la carga aumenta, el motor se frenará tanto que la caída resultante en E_c permitirá que la aumente hasta llevar la carga. Contrariamente si la carga disminuye, el motor se acelera de modo que la E_c resultante (mayor) limita la I_a a un valor menor, que apenas satisface los requerimientos. Cuando un motor está sin carga, la velocidad aumenta para permitir apenas la corriente suficiente que compense las diversas pérdidas internas del par. Sin embargo, la carga se puede hacer tan grande que la corriente no es suficiente para satisfacer las exigencias de los pares, y el motor se detiene. La corriente de detención, por cierto, es igual a la corriente de arranque a voltaje de línea pleno y es muy grande.

Un voltaje multiplicado por una corriente, que es igual a potencia eléctrica en watts, aparece como potencia mecánica en watts en un motor.

El término $E_c I_a$ es potencia mecánica desarrollada:

$$E_c I_a = P_d$$

El término $(I_a I_a R_a)$ es una pérdida de potencia que se manifiesta directamente en forma de calor en los diversos elementos involucrados de la armadura.

Si suponemos un motor que con su carga normal toma 22.5 A a 125 V de la línea y tiene una resistencia del circuito de la armadura de 0.45 ohms, tendríamos:

La contrafem que se genera

$$V_a - E_c = I_a R_a, \quad \text{despejando } E_c = V_a - I_a R_a$$

$$E_c = 125 \text{ V} - (22.5 \text{ A}) * 0.45 \text{ ohms} = 114.9 \text{ V}$$

y la potencia mecánica bruta desarrollada

$$E_c I_a = P_d$$

$$114.9 \text{ V} * 22.5 \text{ A} = 2585 \text{ W}$$

$$P_d = 2.585 \text{ Kw}$$

como hay 746 W/hp, equivale a

$$2585 / 746 = P_d = 3.465 \text{ hp}$$

La potencia útil neta es un poco menor.

2.1.5 Velocidad de equilibrio del motor.

La velocidad de rotación de un motor de cd es el resultado en equilibrio de la contrafem debida a la velocidad que permite apenas la corriente precisa para satisfacer los requerimientos del par bruto. La potencia bruta desarrollada es el resultado en equilibrio de la potencia bruta del circuito de la armadura proveniente de la línea menos las pérdidas del tipo (I^2R) en el cobre del circuito de la armadura. Cualquier potencia requerida del campo en derivación también es una pérdida, lo mismo que algunas otras de tipo mecánico y magnético.

La contrafem desarrollada es un factor principal que determina el desempeño del motor. Las ecuaciones para calcular esta contrafem son las relaciones básicas de un generador. Sin embargo, los términos de flujo de campo utilizados dependen de la forma en que se exciten las diversas bobinas de campo en serie y/o en derivación. Los distintos tipos de motores dependen fundamentalmente entonces de cómo se dispongan los circuitos de sus devanados de campo.

Puede emplearse la ecuación $V_a - E_c = I_a R_a$ y despejarla en la forma $E_c = V_a - I_a R_a$. Por lo tanto, para un motor dado, en el que la disposición mecánica es fija, puede utilizarse la siguiente modificación:

$$E_c = \frac{\Phi ZSP}{60a} \times 10^{-8} \text{ V}$$

Una vez construido el motor, Z , P y a son cantidades invariantes.

Igualmente, 10^{-8} y 60 son constantes propias de las unidades en el sistema inglés. Por lo tanto se puede decir que:

$$E_c = K\Phi S$$

$$E_c = k\phi\omega$$

y

donde K y k son constantes fijas, igualando las dos ecuaciones anteriores con $E_c = V_a - I_a R_a$, tenemos:

$$K\Phi S = V_a - I_a R_a$$

$$k\phi\omega = V_a - I_a R_a$$

Esto da

$$\text{rpm} = S = \frac{V_a - I_a R_a}{K\Phi}$$

y

$$\text{radianes por segundo} = \omega = \frac{V_a - I_a R_a}{k\phi}$$

Estas dos ecuaciones muestran que la rapidez de un motor es directamente proporcional a la contrafem ($V_a - I_a R_a$), e inversamente proporcional al valor de K (o k) y al flujo de campo Φ (o ϕ). La carga de un motor hace que la (I_a) y Φ (o ϕ), cambien en distintas formas, dependiendo de cómo se conectan los circuitos de campo. Los tipos de motores de cd que hay son de excitación independiente, en derivación, en serie y compuesta.

2.1.6 Tipo de excitación en motores de CD.

De excitación Independiente. Son aquellos que obtienen la alimentación del rotor y del estator de dos fuentes de tensión independientes. Con ello, el campo del estator es constante al no depender de la carga del motor, y el par de fuerza es entonces prácticamente constante. Las variaciones de velocidad al aumentar la carga se deberán sólo a la disminución de la fuerza electromotriz por aumentar la caída de tensión en el rotor. Este sistema de excitación no se suele utilizar debido al inconveniente que presenta el tener que utilizar una fuente exterior de corriente.

De excitación en Derivación. Los devanados del inducido e inductor están conectados en paralelo y alimentados por una fuente común. También se denominan máquinas *shunt*, y en ellas un aumento de la tensión en el inducido hace aumentar la velocidad de la máquina.

De excitación en Serie. Los devanados de inducido y el inductor están colocados en serie y alimentados por una misma fuente de tensión. En este tipo de motores existe dependencia entre el par y la velocidad; son motores en los que, al aumentar la corriente de excitación, se hace disminuir la velocidad, con un aumento del par.

De excitación Compuesta. También llamados *compound*, en este caso el devanado de excitación tiene una parte de él en serie con el inducido y otra parte en paralelo. El arrollamiento en serie con el inducido está constituido por pocas espiras de gran sección, mientras que el otro está formado por un gran número de espiras de pequeña sección. Permite obtener por tanto un motor con las ventajas del motor serie, pero sin sus inconvenientes. Sus curvas características serán intermedias entre las que se obtienen con excitación serie y con excitación en derivación. Existen dos tipos de excitación compuesta. En la llamada *compuesta acumulativa* el sentido de la corriente que recorre los arrollamientos serie y paralelo es el mismo, por lo que sus efectos se suman, el campo serie apoya al campo en derivación, a diferencia de la *compuesta diferencial*, donde el sentido de la corriente que recorre los arrollamientos tiene sentido contrario y por lo tanto los efectos de ambos devanados se restan, el campo serie se conecta para oponerse al campo en derivación.

2.2 Control de velocidad en motores de CD.

Una vez que el motor de cd se pone en movimiento y ha concluido su aceleración temporizada o relacionada con la carga, se define que tan lento o rápido se desea que trabaje. Muchos sistemas trabajan a una velocidad fija, sin embargo, muchos sistemas requieren que se modifique la velocidad o se mantenga cuando varíe la carga.

Existen cuatro medios básicos de controlar la velocidad de un motor de cd, y cada uno tiene un rango aplicable distinto de efectividad.

- 1) Control de campo, más específicamente, control de flujo magnético de campo.
- 2) Control de resistencia de armadura o control del voltaje de armadura disponible por resistencia en serie.
- 3) Control de resistencia de armadura en serie y en derivación, que utiliza resistencia tanto en serie como en derivación con la armadura. Hay complicación adicional y mayores pérdidas, pero tiene ciertas características deseables.
- 4) Control de voltaje de armadura, el cual, utiliza una fuente de voltaje controlado para la armadura. Esto puede ser en un motor en derivación excitado por separado, independiente o en un motor en serie. Por lo general, una alimentación colectiva tiene un voltaje relativamente fijo, pero cuando se ha proyectado para ser variable tiene muchas ventajas.

2.2.1 Causas básicas y límites de la velocidad.

Estos cuatro tipos afectan a cada tipo de motor de manera diferente. Es necesario analizar las causas básicas y las limitaciones de la velocidad de rotación. Un motor de cd de cualquier tipo alcanza una velocidad de equilibrio donde la corriente que se toma de las líneas suministra justo el par suficiente para las condiciones. La corriente se regula por la diferencia entre el voltaje de línea y la fuerza contraelectromotriz, o contravoltaje, que se genera dentro de la armadura en rotación. La fuerza contraelectromotriz a su vez, obedece exactamente la misma fórmula de generación de voltaje que controla a un generador. El voltaje que se genera es:

$$E_c = \frac{\Phi Z S P}{60 a} \times 10^{-8} \text{ V}$$

para unidades del SI

$$E_c = \frac{\phi Z \omega P}{a} \text{ volts}$$

En ambos casos, Z es el número de conductores por armadura, P el número de polos y a el número de trayectos paralelos en el devanado. Todos son fijados en la construcción de la maquina y no son ajustables en servicio.

-8

Las cantidades en la ecuación del sistema inglés 10 y 60 son constantes de proporcionalidad necesarias y no se pueden modificar. Las dos fórmulas de generación se transforman en:

$$E_c = K\Phi S \quad \text{y} \quad E_c = k\phi\omega$$

Cualquier máquina cuya disposición de devanados y construcción física sean fijos genera un voltaje que es directamente proporcional a su flujo magnético total y a su velocidad de rotación. Este voltaje E se considera que es la fuerza contraelectromotriz o contravoltaje, y se identificará como E_c , de modo que $E = E_c$.

Durante la acción del motor, E_c debe ser menor que el voltaje de terminales. Se ajusta en forma automática. Esto se puede ver en las ecuaciones:

$$E_c = V_t - I_a R_a = K\Phi S$$

y

$$E_c = V_t - I_a R_a = k\phi\omega$$

Si estas ecuaciones se transponen y se despeja el término de velocidad, encontramos que:

$$S = \frac{E_c}{K\Phi} = \frac{V_t - I_a R_a}{K\Phi}$$

y

$$\omega = \frac{E_c}{k\phi} = \frac{V_t - I_a R_a}{k\phi}$$

donde $V_t = V_l$.

Con base en estas relaciones podemos concluir que la velocidad de rotación de estado estable se puede cambiar modificando ya sea el contravoltaje E_c o el flujo Φ (o ϕ). El flujo se puede cambiar dentro de los límites de la máquina particular como límite alto. El límite bajo de flujo es en realidad el magnetismo residual que la estructura retiene. Sin embargo, si el campo se debilita demasiado, la conmutación se ve afectada y el motor se vuelve inestable. El término de contravoltaje se controla tanto por el voltaje de línea V_l como por la corriente que toma el motor. La corriente que se requiere está casi directamente relacionada con el par de carga necesario.

No siempre es necesario evaluar en forma efectiva los términos Φ (o ϕ). Por lo común es posible, trabajar a partir de una situación donde se conocen las corrientes, voltajes y velocidades, predecir el efecto de un cambio causado por un circuito de armadura o una resistencia de circuito de campo sin determinar el flujo.

Si el ajuste de velocidad se hace al modificar sólo el voltaje a través de las terminales del circuito de armadura, y si la carga es de tal naturaleza que la corriente de armadura permanece constante, el par se conserva constante. Al

mismo tiempo, la potencia bruta de armadura varía casi en forma directa con la velocidad.

La potencia bruta del motor es el producto del contravoltaje que se desarrolla y la corriente de armadura. Esto es cierto por la ley básica de potencia $P = I E$, pero en este caso $Pd = Ia Ec$.

La potencia real de la flecha que se entrega a la carga es esta potencia bruta de armadura menos la pérdida por rotación en las condiciones de velocidad y flujo. Esto es:

$$P_{out} = Ia Ec - P_{rot}$$

2.2.2 Mecanismos que se usan en el control de velocidad.

Se emplean muchos métodos para modificar las resistencias que se usan en el control de velocidad de motores. El control por resistencia de armadura en serie, suele ser parte de las resistencias limitadoras de corriente de aceleración.

Las resistencias de armadura en derivación, son unidades de alta potencia y se controlan en etapas por contactores que ponen en cortocircuito porciones de la resistencia. Estas son parte de un sistema de freno dinámico.

Las resistencias de campo en derivación, son por lo general reóstatos, pues transportan sólo las corrientes relativamente más pequeñas de campo en derivación. Cualquier resistencia de circuito de armadura que se use para el control de la velocidad representa una pérdida considerable. Una resistencia de circuito de armadura en serie que haga caer el voltaje de terminales del motor a la mitad del voltaje de línea consume tanta energía como el motor mismo. Un motor con una eficiencia de 80% que trabaja a la mitad del voltaje tiene una eficiencia mala de $80 * 0.5 = 40\%$. Además, en estas condiciones una resistencia de circuito de armadura debe ser mucho más grande de lo que tiene que ser para el servicio de arranque.

2.2.3 Control de velocidad de estado sólido en motores de CD.

El control de velocidad de motores es una parte muy necesaria e importante en muchos sistemas. Se usan motores de cd para manejar un gran número de estos sistemas porque es relativamente fácil controlar su velocidad.

Con el desarrollo y las mejoras en la tecnología de semiconductores de potencia, los motores de inducción de ca se usan cada vez más en aplicaciones que requieren velocidad variable. El uso de motores de cd, está decayendo porque tienen escobillas y requieren un mantenimiento más frecuente y costoso. Sin embargo, los motores de cd son en general menos costosos en el arranque y tienen características de desempeño extremadamente buenas.

Como ya se ha señalado, la velocidad de un motor de cd se puede modificar al variar el voltaje hacia los devanados de campo o el voltaje hacia la armadura. Al cambiar el voltaje hacia el campo mientras se alimenta voltaje constante a la armadura, la velocidad sólo se puede hacer variar arriba de la velocidad nominal. Esto menoscaba el par de salida haciendo que disminuya conforme aumenta la velocidad. Sin embargo, cuando se mantiene constante el voltaje (y la corriente) de campo y se modifica el voltaje de armadura, la velocidad del motor de cd se puede hacer variar desde cierto valor fijo arriba de cero hasta la velocidad nominal con par constante.

En los métodos convencionales de control de velocidad se consigue al modificar los resistores de potencia, los cuales hacen variar el voltaje de campo, el voltaje de armadura o ambos. El control de estado sólido logra el mismo objetivo electrónicamente y no tiene resistores de gran disipación que dispersen y malgasten energía. Las pérdidas de energía externas al motor mismo se restringen a las pérdidas en los circuitos de estado sólido y periféricos de soporte, es decir, los circuitos de activación y de conmutación.

El motor de cd en serie, con su elevado par de arranque, se usa en general con cargas inerciales altas, pero en si tiene mala regulación de velocidad. El motor en derivación tiene excelente regulación de velocidad y a veces se le describe como una maquina de velocidad constante, aunque su velocidad disminuye ligeramente con carga.

El motor en derivación tiene por lo general su devanado de campo conectado en paralelo con la armadura, pero puede alimentarse desde una fuente totalmente independiente, en cuyo caso la disposición completa se conoce como un motor de cd en derivación excitado por separado.

En el circuito de la figura 2.1, se representa un motor de este tipo. El circuito de armadura se muestra con la resistencia R_a , la inductancia de armadura L_a , en serie con un inductor filtrado L_x . Se muestra un voltaje de generador E_c , en serie con R_a , L_a y L_x , para representar el contravoltaje que se genera dentro de la armadura del motor conforme el mismo gira. El devanado de campo también está compuesto de la resistencia de campo R_f , y la inductancia de campo L_f .

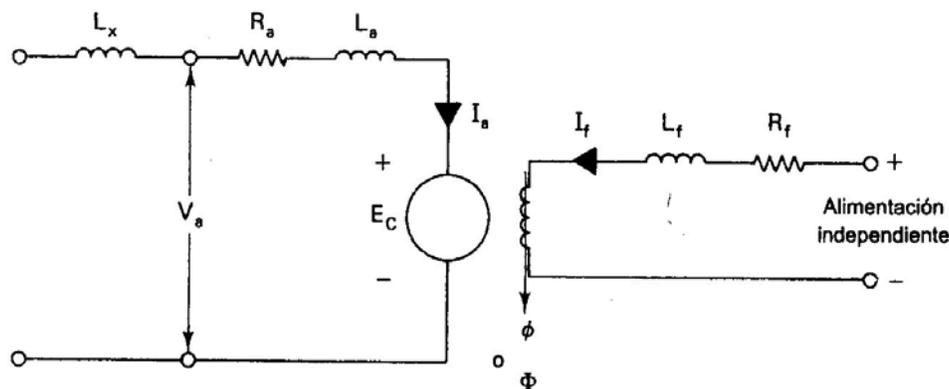


Figura 2.1 Motor de cd en derivación excitado por separado

De las ecuaciones:

$$\text{rpm} = S = \frac{V_a - I_a R_a}{K\Phi} = \frac{E_c}{K\Phi}$$

$$\text{radianes por segundo} = \omega = \frac{V_a - I_a R_a}{k\phi} = \frac{E_c}{k\phi}$$

$$T = C \Phi I_a$$

$$t = c \phi I_a$$

Se pueden hacer ciertas observaciones de cómo se puede controlar la velocidad o el par en una máquina de cd.

El par de un motor de cd en derivación es directamente proporcional a dos variables que son el flujo de campo y la corriente de armadura. C o c en las ecuaciones anteriores es una constante. Si el flujo de campo se conserva constante (manteniendo una corriente de campo constante I_f), entonces el par que se produce dependerá del valor promedio de la corriente de armadura I_a . Si tanto el flujo de campo como la corriente de armadura se controlan cada uno a un valor constante, el par será constante.

Un Modo de operación común ocurre cuando la máquina trabaja con velocidad constante y par variable, como se usa en general con bombas y bandas transportadoras. El flujo de campo se conserva constante manteniendo la corriente de campo nominal con una fuente por separado. Puesto que C o c y Φ (ϕ) se mantienen constantes, la única manera de hacer variar el par es modificar la corriente de armadura I_a . Para mantener velocidad constante con las constantes dadas, se debe conservar constante el contravoltaje E_c . Puesto que $E_c = V_a - I_a R_a$, si I_a aumenta debido a que se carga el motor, V_a se debe incrementar de inmediato en una cantidad adecuada para conservar E_c constante. Esto a su vez, mantiene una velocidad constante al nuevo par de salida.

Por último, se puede dar cabida a las aplicaciones donde se requiere velocidad constante y par variable a través de variaciones simultáneas controladas del voltaje de armadura V_a , que se aplica y de la corriente de armadura I_a .

CAPÍTULO 3. SOFTWARE PARA SISTEMAS DE DIFUSOS

Una gran variedad de paquetes de software para sistemas difusos han sido desarrollados para facilitar al diseñador la prueba e implementación de sus máquinas de inferencia difusa. En este capítulo daremos referencias de tres de ellos, FIDE, FuzzyTech y MATLAB.

FIDE y FuzzyTech son paquetes de software difíciles de obtener por su alto costo, además de que en forma particular, vía Internet, no se obtuvo respuesta de cómo obtenerlo.

3.1 FIDE, Fuzzy Inference Development Environment (herramienta de desarrollo de Inferencia Difusa).

Con FIDE se puede hacer el diseño en tres pasos:

Paso #1. Describe el problema con reglas y Funciones de membresía.

Puede describir el controlador con Apronix's FIL, usando texto en la herramienta del editor. Se puede especificar las variables entrada/salida, las funciones de membresía asociadas y las reglas describiendo la operación del controlador. FIL permite usar diferentes operadores, métodos de inferencia y desfusión, también el número de bits para el microcontrolador.

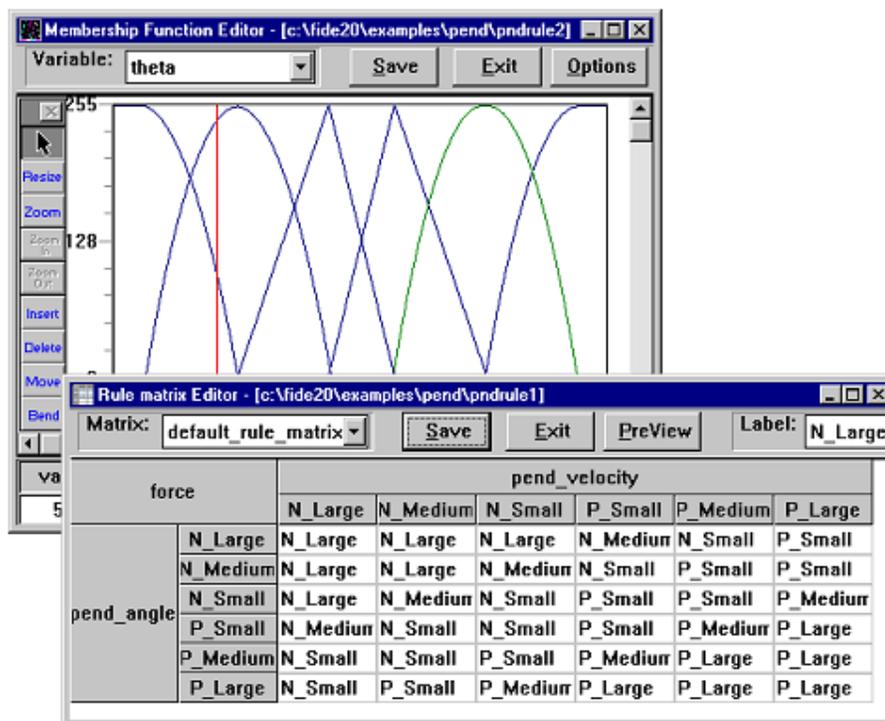


Figura 3.1 Editor de funciones de membresía y reglas.

En FIL se puede crear o modificar las funciones de membresía describiendo los rangos de entrada y salida usando la herramienta MF-Editor grafico. Esta soporta curvas triangulares o más complejas. Tiene la capacidad de agregar, borrar, modificar puntos individuales. El Editor de Reglas provee una opción para la construcción de reglas difusas. Una vez terminado, se puede compilar para trasladar la fuente en estructuras de datos internos Apronix.

Paso #2. Depuración y sintonización del controlador

Herramientas convencionales incorporan depuradores, estos explotan la naturaleza secuencial del código ejecutado. Las herramientas de depuradores fueron diseñadas para tomar ventaja del hecho de que las reglas difusas se ejecutan en paralelo. FIDE provee una elegante forma de sintonización para el controlador examinando y analizando sus cualidades desde múltiples perspectivas vía tres poderosas herramientas de debug (depuración), el trazador, el analizador y el simulador.

Tracer (trazador):

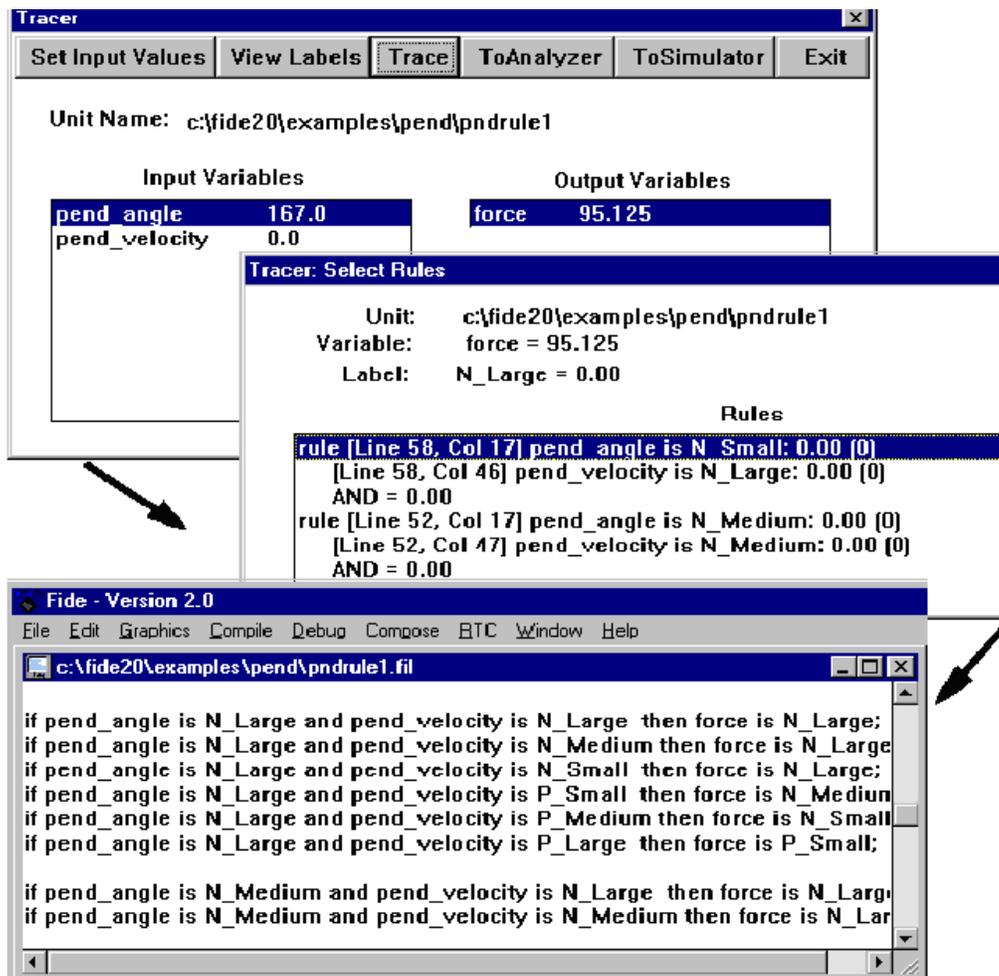


Figura 3.2 Trazador.

Una de las características inherentes de la Inferencia difusa es que el proceso es transparente, esto es, podemos trazar una salida volviendo a la entrada específica que causo esta. El trazador de FIDE toma ventaja de esto, provee una capacidad en línea de trazar paso por paso, a través del proceso de Inferencia difusa. Por ejemplo, se puede poner un número en la entrada y observar el valor resultante de salida. Si un resultado no deseado ocurre se puede identificar fácilmente la fuente del problema siguiendo hacia atrás paso a paso desde la salida todo el camino de una regla específica que tuvo la mayor contribución al resultado. Entonces se puede modificar alguna función de membresía y volver a correrlo teniendo una recompilación del código FIL.

Ejecución del sistema por el Analizador (Display 3-D Transfiere la Función):

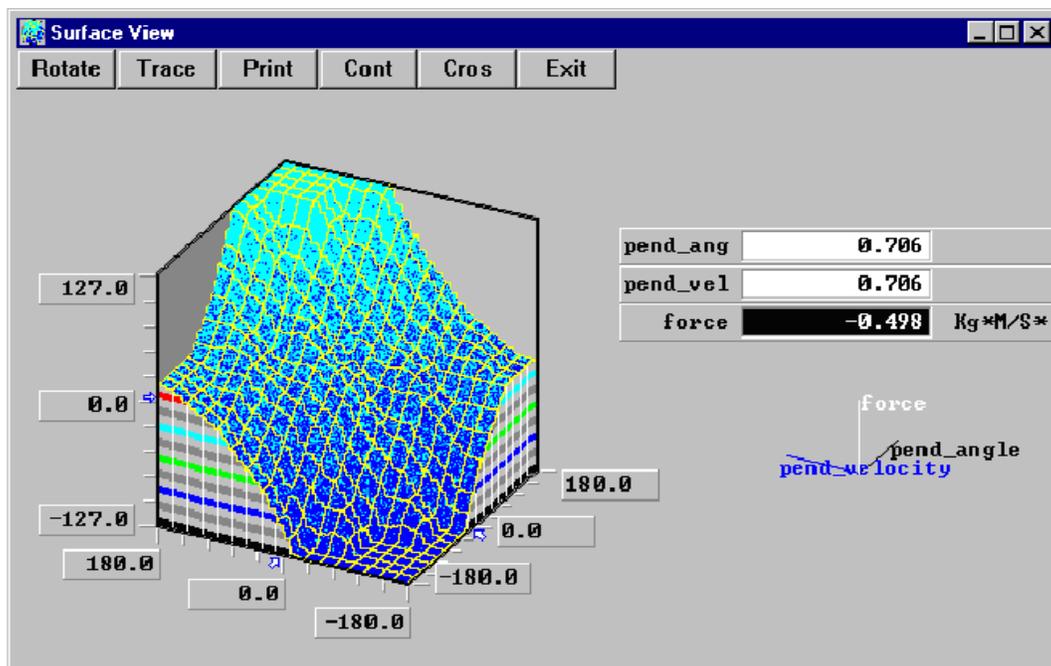


Figura 3.3 Analizador.

El Analizador despliega una vista global de la respuesta de la función de transferencia. Aquí la relación entrada/salida se ve en tres dimensiones. Graficas avanzadas e interactivas le permiten revisar la función en detalle, examinar la superficie desde una variedad de perspectivas y aislar alguna respuesta indeseable por medio de flechas movibles. Apronix tiene avances importantes en la tecnología de debugging (depuradores), provee un link hot (enlace-caliente), entre el analizador y el trazador. Esto permite seguir cualquier problema mostrado en la superficie 3-D directamente en la regla que pudo haber contribuido a ello.

Corre simulaciones con entradas controladas vía el Simulador:

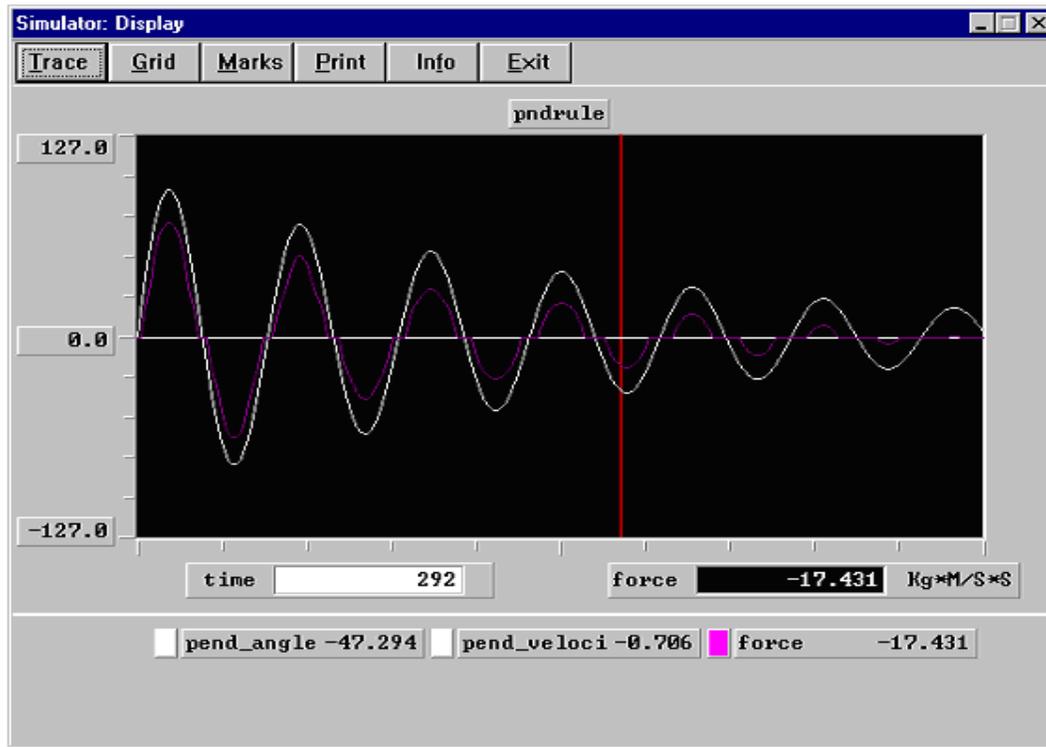


Figura 3.4 Simulador.

El simulador le permite observar la dinámica de su controlador. La entrada al simulador es un archivo el cual contiene un conjunto de valores a ser aplicados a las entradas del controlador. Entonces puede correr el simulador para desplegar la curva de salida generada desde cada uno de los valores aplicados. La línea vertical movable nos permite los valore de entrada y salida en cualquier paso. El simulador usa el link-hot al trazador el cual provee una capacidad superior de depuración permitiéndonos aislar ejecuciones no deseadas en todo el camino desde la fuente.

Paso #3. Generación de Código en Tiempo-Real. C, Ensamblador, Matlab o Java.

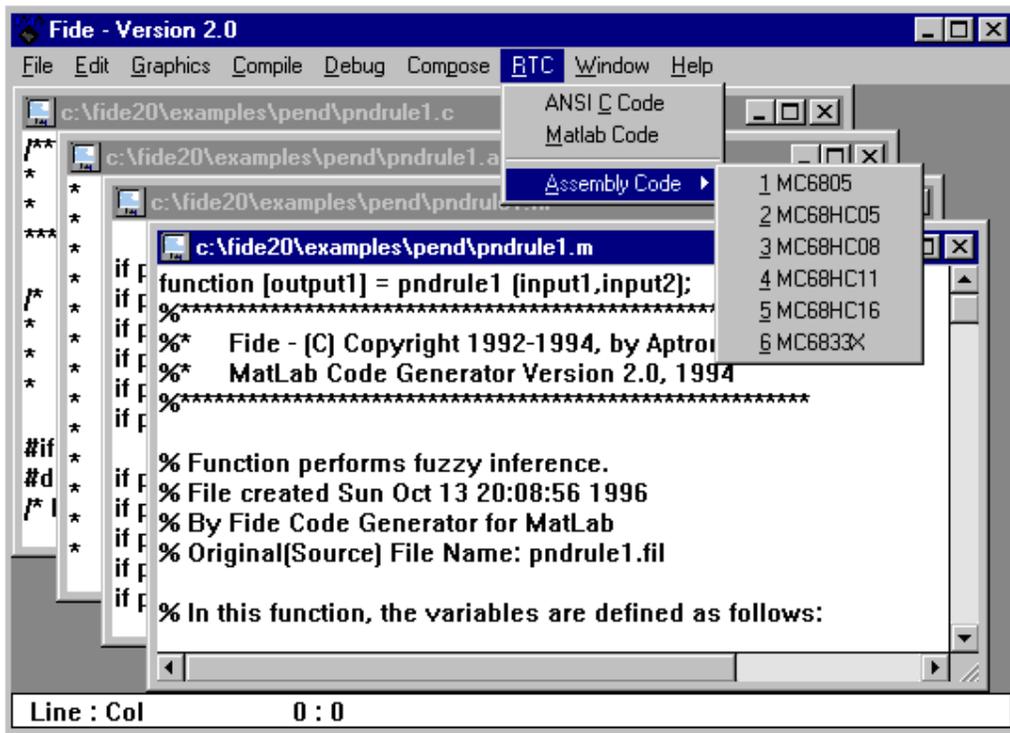


Figura 3.5 Generador de código en tiempo real.

Quando se ha elegido el microcontrolador para la implementación, el RTC (Real Time Code Generator) nos da un programa en lenguaje ensamblador haciendo clic en el menú. Apronix ha demostrado que el RTC de FIDE genera un código compacto ahorrando memoria y ayudando a la aplicación a correr rápido. Si la aplicación es diseñada en software, el RTC puede generar código ANSI C y archivos con código Matlab M.

Las reglas de la lógica difusa se determinan de acuerdo a como el sistema reacciona a las condiciones de la entrada. Aquí es donde se pone en ejecución las estrategias del control y de optimización para su solución. Para responder lo mejor posible a las necesidades de los diversos usos, el fuzzyTECH proporciona tres redactores de reglas de bloque. El redactor de la regla de la hoja de balance sigue el estilo familiar de una tabla. Se define las bases completas de la regla por el mouse (ratón), la consistencia se hace cumplir automáticamente. Muchos usuarios experimentados prefieren el redactor de la regla de la matriz para los bloques complejos de la regla. Incorporar reglas en formato del FTL (Lenguaje de tecnología difusa) le permite mecanografiar dentro de las reglas de manera más flexible. De cualquier manera si se modificaron las reglas, se puede utilizar siempre cualquier otro redactor para procesarlos o para analizar aun más.

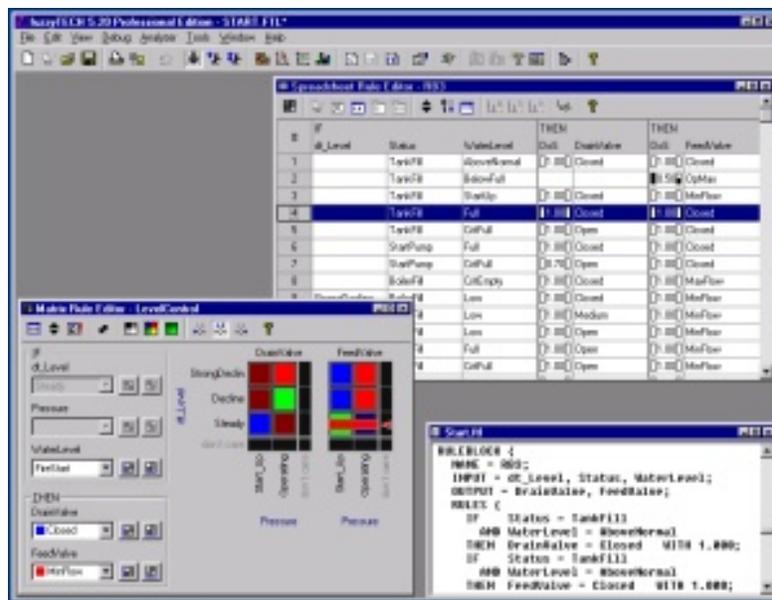


Figura 3.8 Editor de reglas.

El conjunto de herramientas de análisis de fuzzyTech, ha fijado estándares en el software visual de simulación. Las imágenes en 2D y 3D le permiten analizar las características dinámicas de un sistema de lógica difusa, así como sus componentes. Cualquier modificación al sistema se refleja inmediatamente en todos los analizadores. Este desarrollo interactivo facilita la elaboración de la solución. Los analizadores de tiempo le permiten entender y optimizar el sistema de lógica difusa siendo controlado.

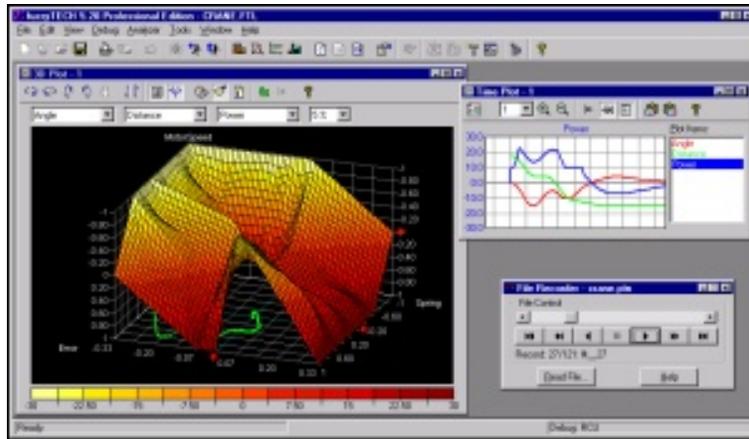


Figura 3.9 Analizadores.

Mientras que fuzzyTECH abarca ya todas las características de la simulación, para el sistema de lógica difusa bajo diseño, se puede también utilizar fuzzyTECH con los paquetes de software estándares de la simulación tales como Matlab/Simulink™, MatrixX™, y otros. Durante la simulación, se pueden utilizar todos los redactores y los analizadores del fuzzyTECH y de se pueden modificar el sistema corriente “en marcha”. Todas las ediciones de fuzzyTECH, incluyendo el software de la demostración, incluyen simulaciones animadas completas de procesos del mundo real.

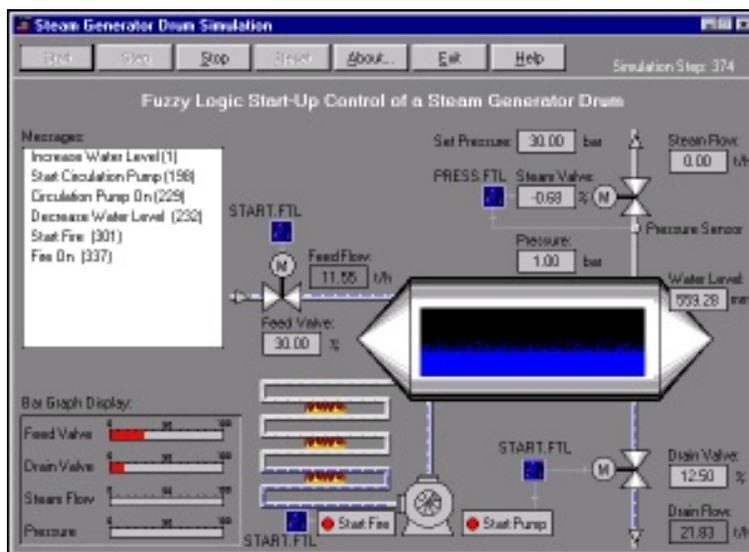


Figura 3.10 Simulador de procesos.

La perspectiva estructurada del diseño de fuzzyTECH, es complementado por un generador de documentación incorporado y un sistema de control de la revisión. El generador de la documentación produce la documentación completa para el desarrollo de acuerdo a estándares ISO 9000.

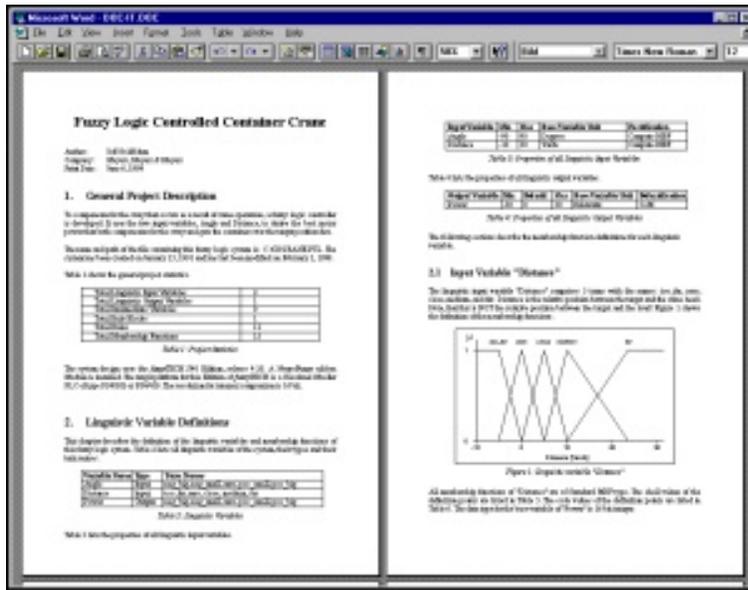


Figura 3.11 Generador de documentación.

3.3 MATLAB.

MATLAB es un lenguaje técnico de computación de alto nivel y un ambiente interactivo para el desarrollo de algoritmos, visualización de los datos, análisis de datos, y cómputo numérico. Usando MATLAB, se pueden solucionar problemas de cómputo técnicos más rápidamente que con lenguajes de programación tradicionales, por ejemplo C, C++, y FORTRAN.

Se puede utilizar MATLAB en una amplia gama de aplicaciones, incluyendo el procesamiento de señales e imágenes, comunicaciones, control, diseño, prueba y mediciones, el modelamiento y análisis financieros y biología. Las herramientas adicionales (colecciones de funciones de propósito especial de MATLAB), amplía por separado el ambiente de MATLAB para solucionar clases particulares de problemas en estas áreas de aplicación.

MATLAB proporciona un número de características para documentar y compartir su trabajo. Se puede integrar su código de MATLAB con otros idiomas y usos, y distribuya sus algoritmos y los usos de MATLAB.

Características principales.

- Idioma de alto nivel para cómputo técnico.
- Ambiente de desarrollo para el código de manejo, archivos y datos.
- Herramientas interactivas para la exploración iterativa en diseño y solución de problemas.
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtración, optimización e integración numérica.
- Gráficas 2D y 3D para la visualización de datos.
- Herramientas para construir interfaces utilizando gráficos personalizados.
- Las funciones que integran MATLAB utilizan algoritmos con usos en lenguajes externos, por ejemplo C, C++, Fortran.

3.3.1 Herramienta de lógica difusa.

El paquete de herramientas (Toolbox) de lógica difusa extiende el ambiente de cómputo técnico de MATLAB con las herramientas para el diseño de los sistemas basados en lógica difusa. Posee una interfase gráfica de usuario (GUIs) que permite realizar el desarrollo de un sistema difuso clásico, así como el reconocimiento de patrones. Las funciones se proporcionan para muchos métodos comunes de la lógica difusa, incluyendo conjuntos.

La herramienta le permite modelar comportamientos del sistema complejo usando reglas de lógica simples y después poner estas reglas en ejecución en un sistema. Se puede utilizar esta herramienta como máquina de inferencia difusa independiente.

Como todos los paquetes de herramientas de MATLAB, la caja de herramientas de la lógica difusa puede ser modificada para requisitos particulares. Se puede examinar algoritmos, modificar el código fuente y agregar sus propias funciones de membresía o técnicas de desdifusión.

Características especiales.

- GUIs especializado para la construcción de los sistemas de inferencia y el análisis de resultados difusos.
- Funciones de membresía para crear sistemas de inferencia difusos.
- Ayuda para la definición de reglas del tipo AND, OR, y NOT definidas por el usuario.
- Funciones de membresía automatizadas formadas a partir de técnicas de aprendizaje neuro-adaptivo y difuso.
- Capacidad de incrustar un sistema de inferencia difuso en un modelo de Simulink.
- Capacidad de generar código incrustado de C o las maquinas de inferencia difusa ejecutables independientes.

3.3.2 Proceso de Inferencia Difusa.

En la herramienta de lógica difusa se tienen 5 partes en el proceso de inferencia difusa, difusión de las variables de entrada, aplicación de los operadores difusos (AND y OR) en el antecedente, implicación del antecedente en el consecuente, agregación de los consecuentes a través de las reglas y la desfusión.

Paso 1. El primer paso es tomar las entradas y determinar el grado en el cual estas pertenecen a cada uno de los conjuntos difusos vía las funciones de membresía. Las entradas son por lo regular valores numéricos limitados al universo de discurso de las variables de entrada y la salida es un grado difuso de membresía de alguno de sus conjuntos difusos.

Paso 2. Al aplicar el operador difuso después de la difusión de las entradas, conocemos el grado en el que cada parte del antecedente es satisfactorio para cada regla. En el antecedente de cada regla podemos tener más de una parte, el operador difuso es aplicado para obtener un valor que represente el resultado del antecedente de esa regla. Este valor puede ser entonces aplicado a la función de salida. Las entradas del operador difuso son dos o mas valores de membresía de las variables de entrada difusas. La salida es un valor real simple. Se tienen varios métodos que pueden ser usados para las operaciones AND y OR. En la herramienta de lógica difusa se tienen dos métodos, para AND soporta min (mínimo) y prod (producto). Para OR se tienen dos métodos, max (máximo) y probor (probabilístico), también conocido como suma algebraica, este es calculado de acuerdo a $\text{probor}(a,b) = a + b - ab$.

Paso 3. Método de Implicación. Antes de aplicar el método de implicación, se toma cuidado del peso de las reglas. Cada regla tiene un peso (un numero entre 0 y 1), el cual es aplicado al numero dado por el antecedente. Generalmente este peso es 1 y por lo tanto este no tiene efecto en todo el proceso de implicación. De tiempo en tiempo se puede buscar evaluar un peso relativo de una regla con respecto a otras cambiando su valor de peso a un valor diferente de 1. Una vez que un valor de peso apropiado ha sido asignado a cada una de las reglas, el método de implicación es implementado. La consecuencia es un conjunto difuso representado por una función de membresía, el cual pesa apropiadamente las características lingüísticas que

son atribuidas a este. El consecuente es reformado usando la función asociada con el antecedente (un único número). La entrada para el proceso de implicación es un número único dado por el antecedente y la salida es un conjunto difuso. La implicación es implementada para cada regla. Se tienen dos métodos, y estos son las mismas funciones que son usadas por el método AND, estos son min (mínimo), el cual trunca el conjunto difuso de salida, y prod (producto), el cual escala el conjunto difuso de salida.

Paso 4. Agregación de todas las salidas. Agregación es el proceso por el cual los conjuntos difusos que representan las salidas de cada regla son combinados en un solo conjunto difuso. La agregación solo ocurre una vez para cada variable de salida, justo antes del quinto y último paso, desdifusión. La entrada del proceso de agregación es la lista de las funciones de salida truncadas, producidas por el proceso de implicación para cada regla. La salida del proceso de agregación es un conjunto difuso para cada variable de salida. Este método de agregación es conmutativo, por lo tanto, el orden en el cual las reglas son ejecutadas no tiene importancia. Este soporta tres métodos, max (máximo), probor (OR probabilística), y sum (suma, la suma simple de cada conjunto de salida de las reglas). Las tres reglas han sido puestas juntas para mostrar como la salida de cada regla es combinada, o agregada en un solo conjunto difuso cuya función de membresía asigna un peso para cada valor de salida. En la figura 3.12 se pueden ver representados estos cuatro pasos.

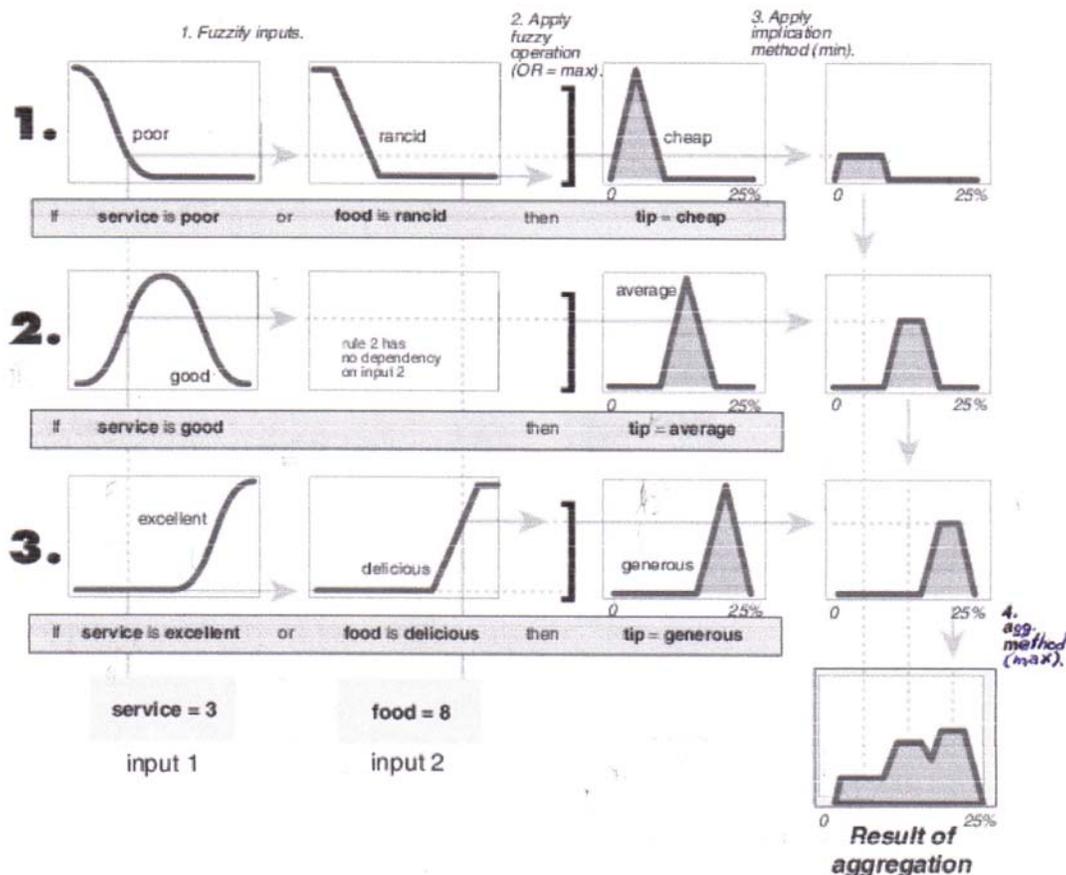


Figura 3.12 Difusión, operadores lógicos, implicación y agregación.

Paso 5. Desdifusión. La entrada del proceso de desdifusión es un conjunto difuso (el agregado conjunto difuso de salida) y la salida es un simple número. La salida final deseada para cada variable es generalmente un número único. Por lo tanto, el agregado de un conjunto difuso esta acompañado de un rango de valores de salida, y por esto debe ser desdifusionado en orden de obtener un valor único de salida del conjunto difuso. El método más popular de desdifusión es el cálculo del centroide, el cual obtiene el centro de área bajo la curva. La desdifusión soporta cinco métodos, centroid (centroide), bisector, mom, middle of maximum (medio de los máximos, el valor medio de los máximos del conjunto de salida), lom, largest of maximum (método del primero de los máximos), y som, smallest of maximum (método del último de los máximos).

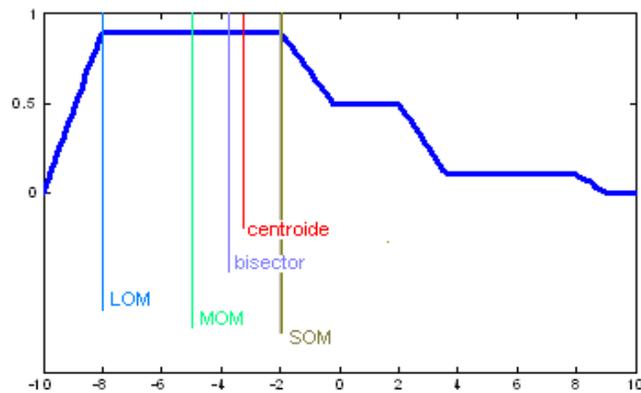


Figura 3.13 Métodos de desdifusión.

3.3.3 Ejemplo de lógica difusa usando MATLAB.

En este ejemplo se dan las instrucciones para trabajar con la herramienta de lógica difusa y se muestran las gráficas y comandos que se necesitan para elaborar un sistema difuso, se conservan los nombres de las variables en ingles con el fin de facilitar la comprensión a quien lo realice y el seguimiento tal como aparece en el programa de MATLAB.

Con este sistema de lógica difusa se pretende evaluar en este ejemplo el porcentaje de propina (variable tip) que recibiría un mesero de acuerdo a la calidad de la comida (variable food) y el servicio al cliente (variable service).

En la herramienta de lógica difusa se ejecuta el siguiente comando:

```
>>fuzzy
```

Aparece una ventana llamada FIS Editor, esta tiene una entrada y una salida, para agregar una entrada se elige en menú Edit, Add Variable e Input.

Con el mouse se hace clic en input 1 se pone en Name el nombre de la variable de entrada, en este caso es "service".

En input 2, se hace lo mismo, se pone el nombre de la segunda variable de entrada, "food".

En output1, que es la variable de salida se cambia el nombre por "tip".
 Del menú File seleccionamos Export y To Workspace, aparece una ventana en que se asigna en Workspace variable, un nombre, en este caso "tipper", se presiona ok.

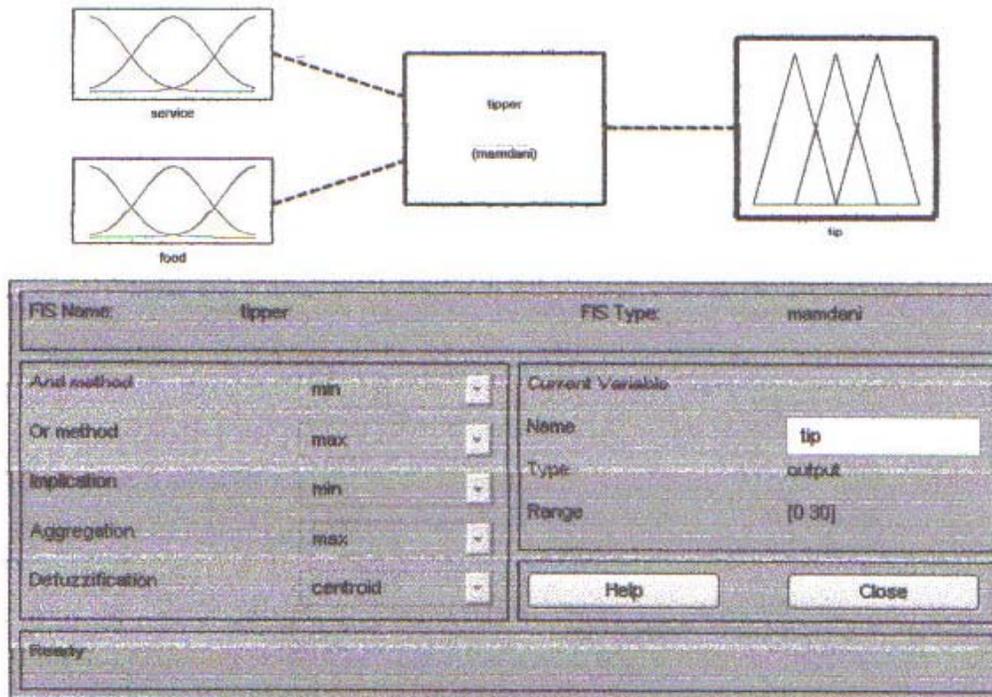


Figura 3.14 Ventana de Editor FIS.

Con el mouse, se da doble clic en el cuadro de la variable de salida "tip", apareciendo la ventana Membership Function Editor.

Selecciona la variable de entrada "service", se da doble clic, se pone Range y Display Range con el vector de valores del universo de discurso, [0 10].

Selecciona Edit y Add MFs, aparece una ventana, elige MF type gaussmf y Number of MFs 3. Si aparecen el número de curvas necesarias para las etiquetas lingüísticas de la variable, no se adicionan y si ocupamos mas se anexan las que falten.

Clic en la primera curva, asigna la etiqueta lingüística, en este caso, el nombre de "poor", Params con [1.5 0].

La curva siguiente, de en medio Name "good" y la tercera "excellent".

Selecciona la variable de entrada "food", Range [0 10].

En el menú de selecciona Edit, Add Mfs, MF type trapmf y Number of MFs 2.

Con un clic en la primer curva asigna, la elegimos y ponemos el nombre de "rancid", Params [0 0 1 3].

La segunda curva Name "delicious".

Lo siguiente es crear la MF para la variable de salida "tip", se elige tipo triangular, Type trimf, Range [0 30].

Se asignan tres curvas, Cheap [0 5 10], average [10 15 20] y generous [20 25 30].

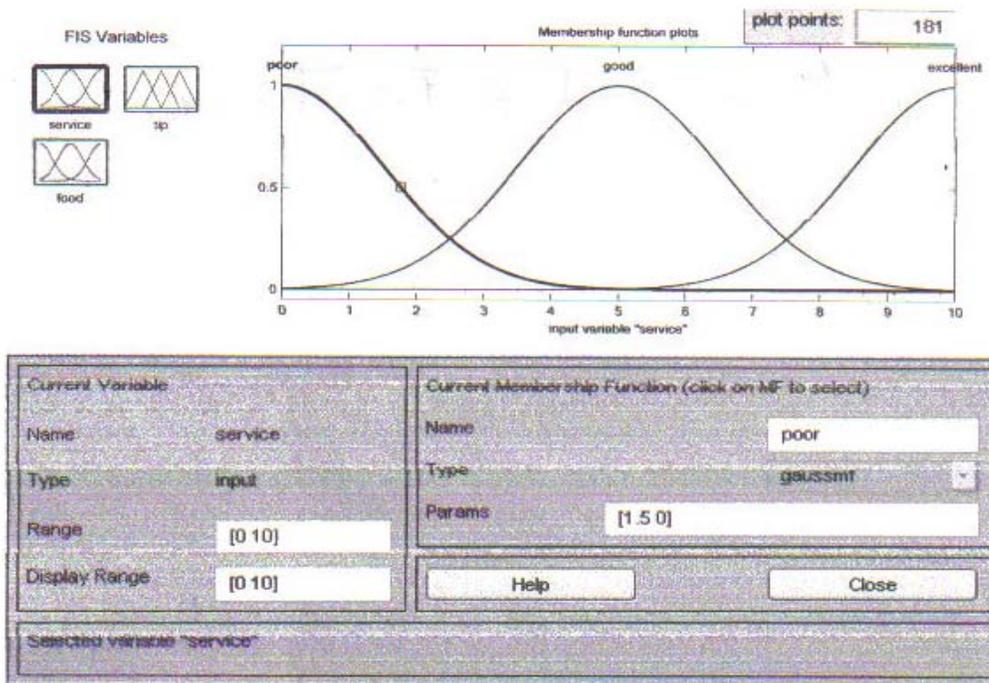


Figura 3.15 Ventana de Editor de funciones de membresía, primer variable de entrada.

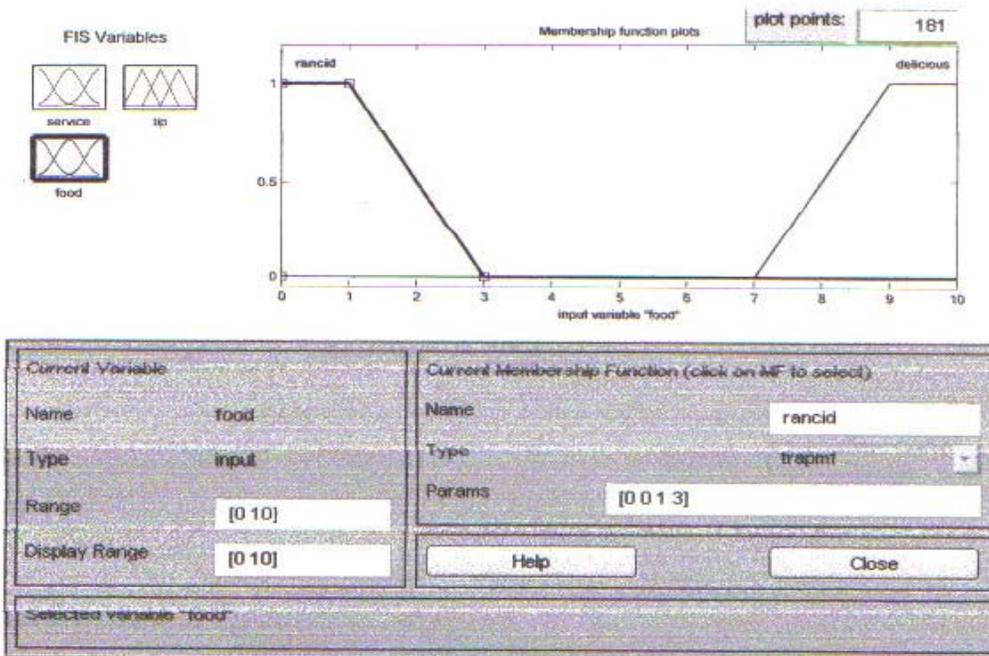


Figura 3.16 Ventana de Editor de funciones de membresía, segunda variable de entrada.

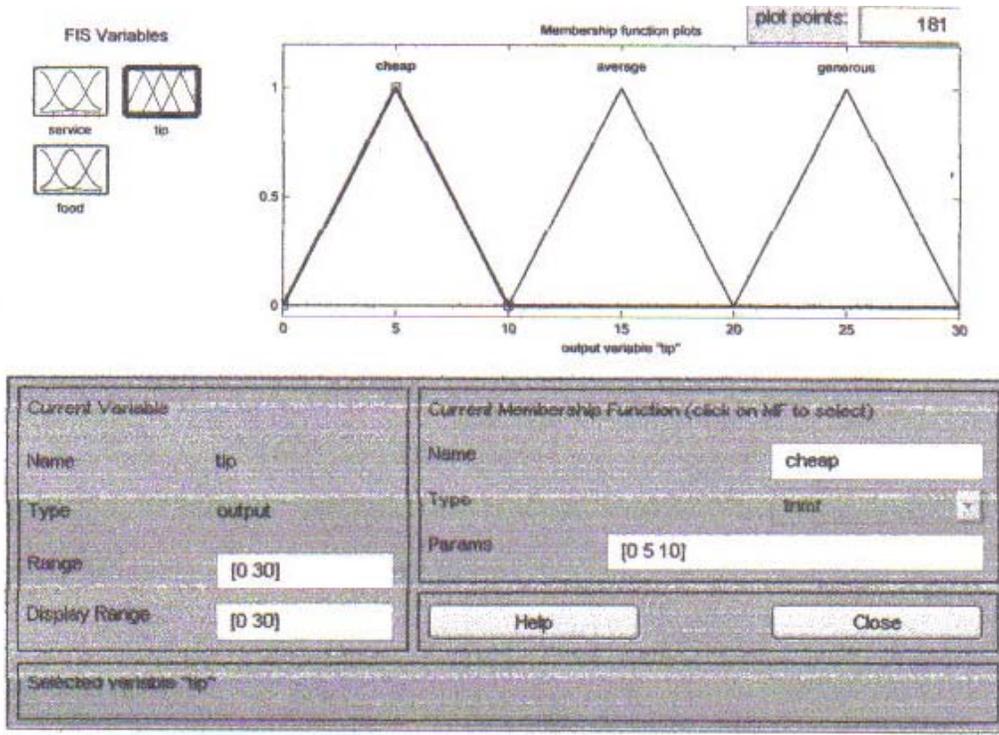


Figura 3.17 Ventana de Editor de funciones de membresía, variable de salida.

Se continúa en menú Edit y Rules y aparece la ventana Rule Editor: tipper. En ella se construyen con variables lingüísticas las reglas difusas IF-THEN.

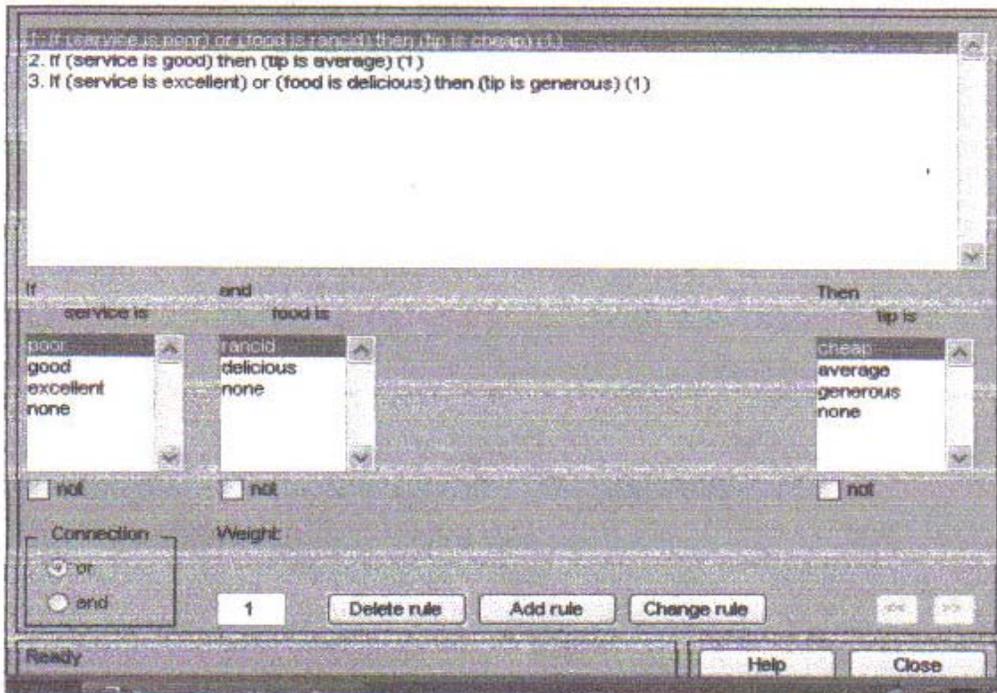


Figura 3.18 Ventana de Editor de reglas.

Después se selecciona View, Rules y aparece la venta Rule Viewer: tipper.

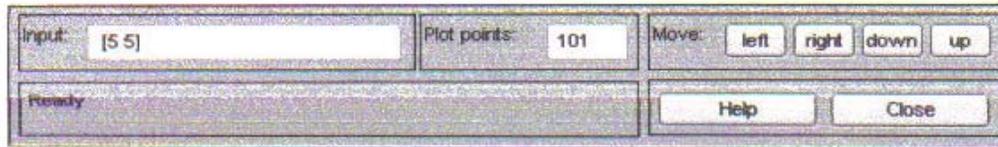
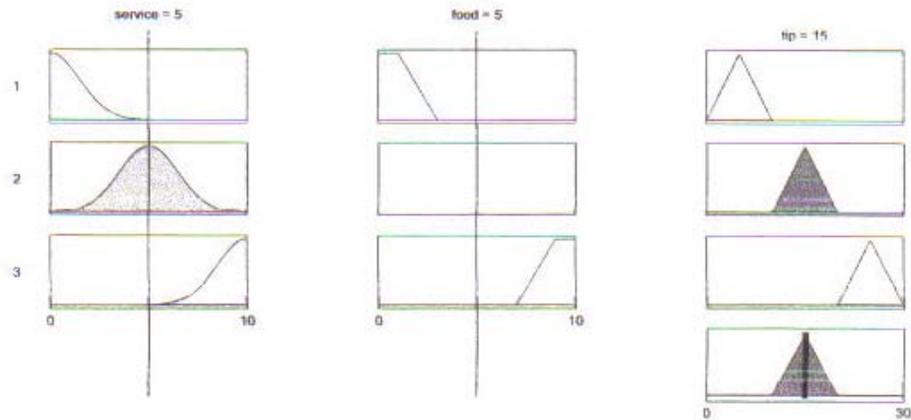


Figura 3.19 Ventana de observación gráfica de reglas.

Para terminar se selecciona View, surface y aparece la venta Surface Viewer: tipper.

Salvar en menú con Edit, export, to disc.

Los siguientes comandos nos permiten visualizar el sistema difuso y evaluarlo.

```
>> readfis('tipper')
```

```
ans =
```

```

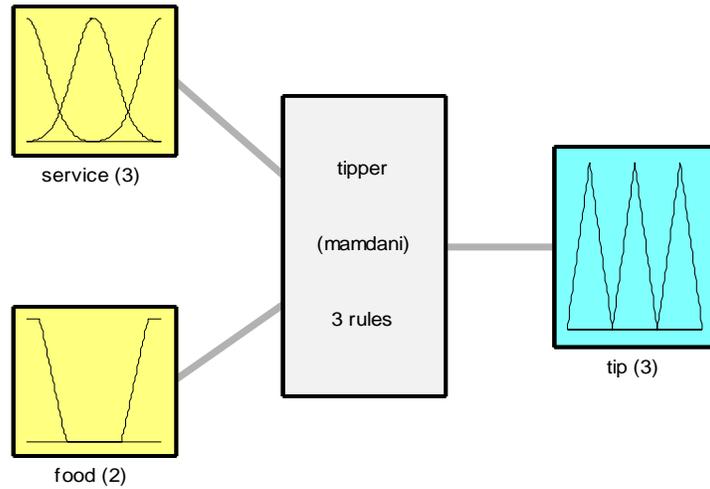
name: 'tipper'
type: 'mamdani'
andMethod: 'min'
orMethod: 'max'
defuzzMethod: 'centroid'
impMethod: 'min'
aggMethod: 'max'
input: [1x2 struct]
output: [1x1 struct]
rule: [1x3 struct]

```

```
>> a=readfis('tipper');
```

(Con los siguientes comandos aparecen ventanas en donde se muestran las figuras).

>> plotfis(a)



System tipper: 2 inputs, 1 outputs, 3 rules

Fig. 3.20 Características del sistema difuso tipper.

>> plotmf(a,'input',1)

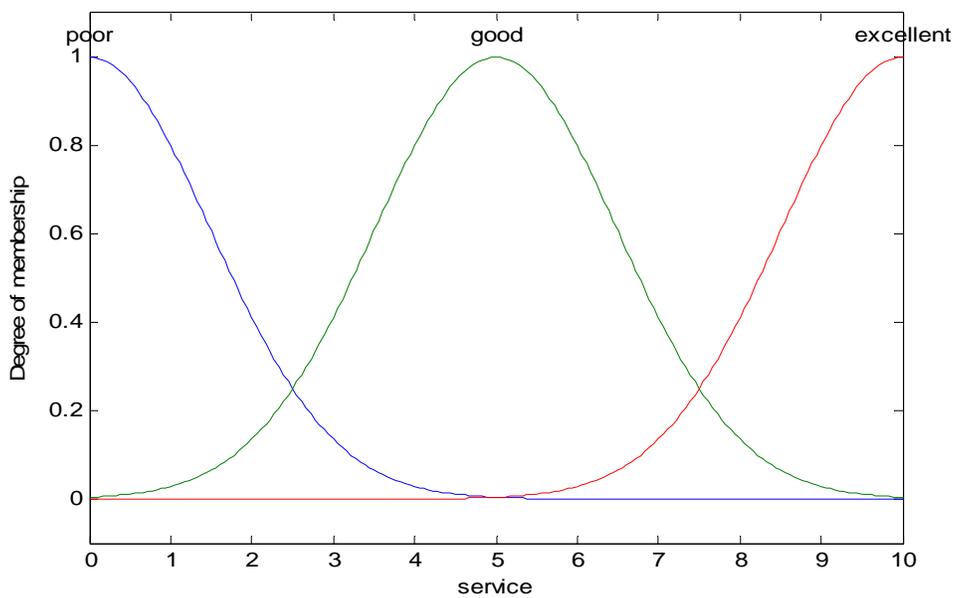


Fig. 3.21 Funciones de membresía para la variable de entrada service.

```
>> plotmf(a,'input',2)
```

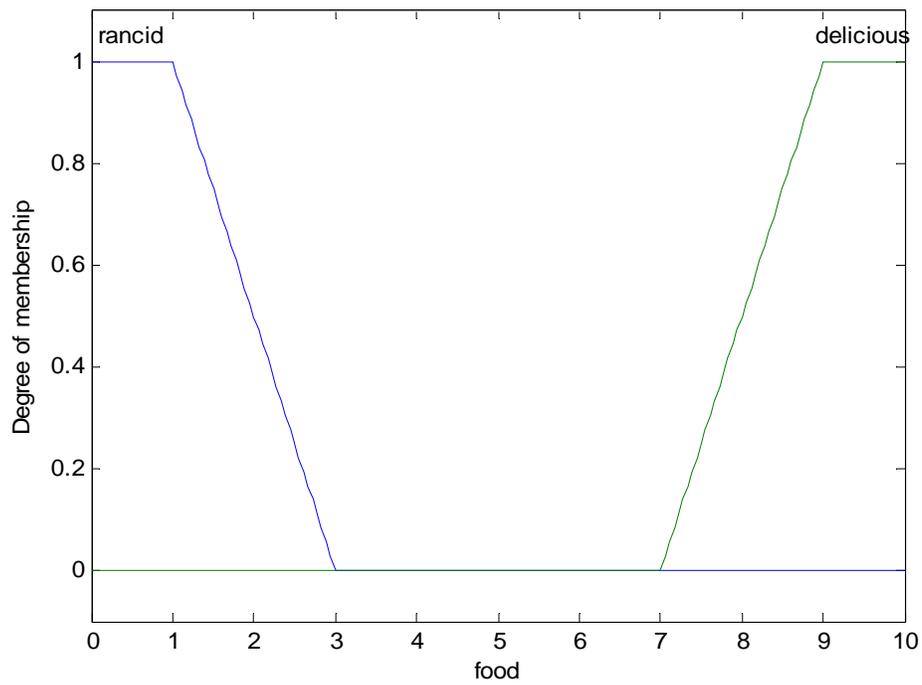


Fig. 3.22 Funciones de membresía para la variable de entrada food.

```
>> plotmf(a,'output',1)
```

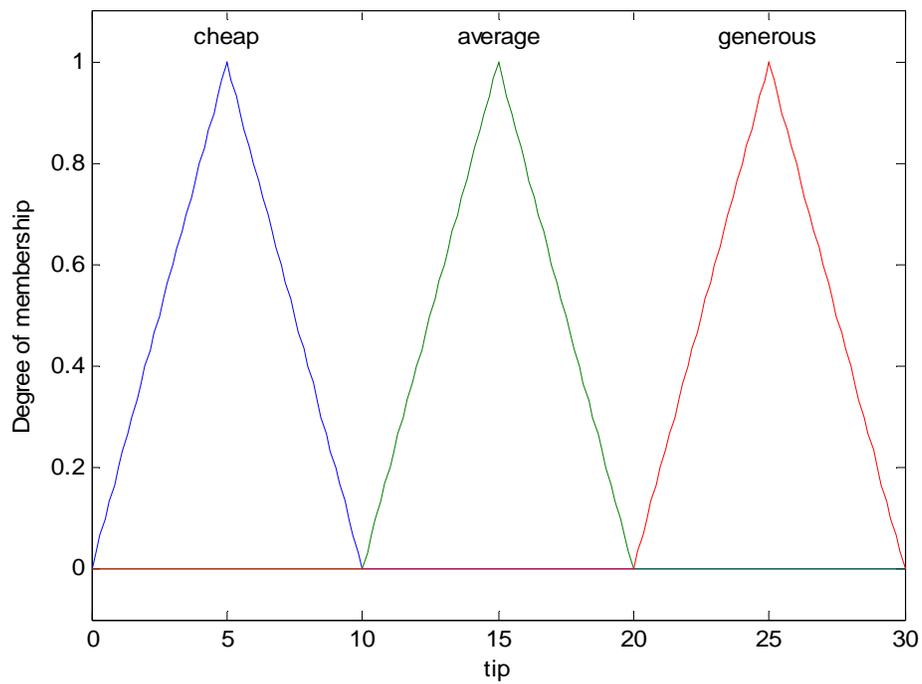


Fig. 3.23 Funciones de membresía para la variable de salida tip.

```
>> gensurf(a)
```

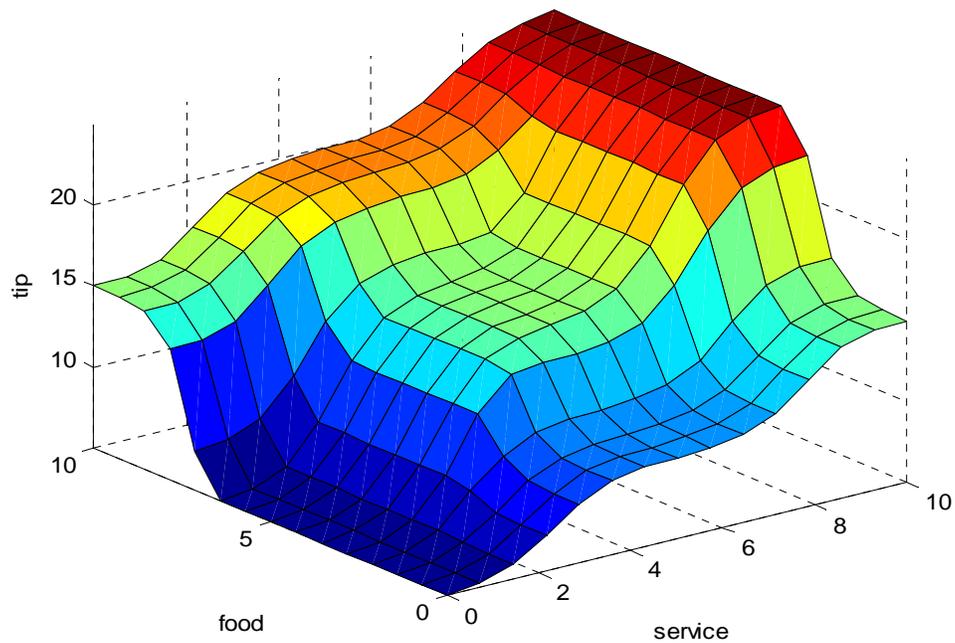


Fig. 3.24 Superficie de control generada.

(El comando `evalfis` nos permite dar valores a las variables de entrada y evaluar nuestro sistema difuso `a`).

```
>> evalfis([5 5], a)
```

```
ans =
```

```
15.0000
```

(Evaluación: Si la variable difusa `food` (comida) tiene un valor de 5, en el rango de 0 a 10 y si la variable difusa `service` (servicio) tiene un valor de 5, en un rango de 0 a 10, entonces el resultado, la variable de salida `tip` (propina) es 15, es decir 15 %).

```
>> evalfis([2 2], a)
```

```
ans =
```

```
7.5201
```

```
>> evalfis([0 10], a)
```

```
ans =
```

15.0000

>> evalfis([10 0], a)

ans =

15.0000

>> evalfis([10 10], a)

ans =

24.9220

>> evalfis([8 10], a)

ans =

22.9831

>> evalfis([4 10], a)

ans =

19.6810

>> evalfis([10 3], a)

ans =

24.9220

CAPÍTULO 4. DISEÑO DEL CONTROLADOR DE VELOCIDAD DEL MOTOR DE CD

4.1 Acciones de control.

En los sistemas de control retroalimentados, una señal actuante, se obtiene de una comparación de la señal de referencia y la señal retroalimentada. La señal actuante se usa para hacer cualquier corrección necesaria. El término acción de control se refiere a la manera en que la señal actuante es empleada por las porciones de control del sistema (los elementos de control) para lograr la corrección. Las acciones de control básicas que se utilizan son:

4.1.1 Acción de control de dos posiciones, encendido y apagado (on-off).

El elemento de actuación solo tiene dos posiciones fijas, en muchos casos, solo encendido y apagado. Suponemos la señal de salida $u(t)$ y la señal de error $e(t)$. La señal $u(t)$ permanece en un valor ya sea máximo o mínimo, dependiendo de si la señal de error es positiva o negativa, tenemos:

$$\begin{aligned} u(t) &= U_1, & \text{para } e(t) > 0 \\ u(t) &= U_2, & \text{para } e(t) < 0 \end{aligned}$$

En donde U_1 y U_2 son constantes. El valor mínimo de U_2 es cero o $-U_1$. Es común que los controladores de dos posiciones sean dispositivos eléctricos, como una válvula operada por un solenoide. Las figuras 4.1 (a) y (b) muestran los diagramas de bloques para dos controladores de dos posiciones. En la figura 4.1(b) se muestra una brecha diferencial. La brecha provoca que la salida del controlador $u(t)$ conserve su valor presente hasta que la señal de error se haya desplazado ligeramente más allá de cero. En algunos casos se provoca para evitar una operación demasiado frecuente del mecanismo de encendido apagado.

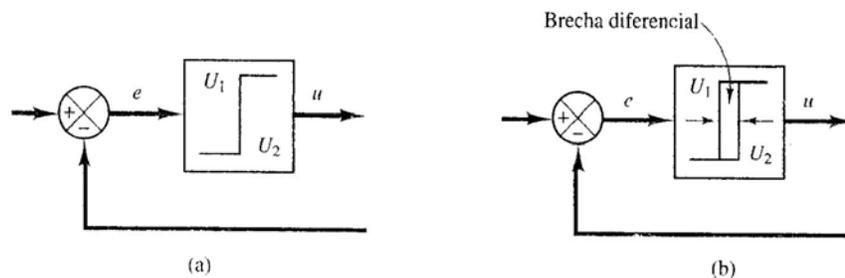


Figura 4.1 (a) Controlador de encendido y apagado; (b) Controlador de encendido y apagado con una brecha diferencial.

4.1.2 Acción de control proporcional.

Para un controlador con acción de control proporcional, la relación entre la salida del controlador $u(t)$ y la señal de error $e(t)$ es:

$$u(t) = K_p e(t)$$

o bien

$$\frac{U(s)}{E(s)} = K_p$$

en donde K_p se considera la ganancia proporcional. Cualquiera que sea el mecanismo real y la forma de la potencia de operación, el controlador proporcional es, en esencia, un amplificador con una ganancia ajustable.

4.1.3 Acción de control integral.

El valor de salida del controlador $u(t)$ se cambia a una razón proporcional a la señal de error $e(t)$. Es decir,

$$\frac{du(t)}{dt} = K_i e(t)$$

o bien

$$u(t) = K_i \int_0^t e(t) dt$$

en donde K_i es una constante ajustable. La función de transferencia del controlador integral es

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

Si se duplica el valor de $e(t)$, el valor de $u(t)$ varía dos veces más rápido. Para un error de cero, el valor de $u(t)$ permanece estacionario. En ocasiones la acción de control integral se denomina control de reajuste (reset).

4.1.4 Acción de control proporcional-integral.

Un controlador PI se define

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

o la función de transferencia es

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

en donde K_p es la ganancia proporcional y T_i se denomina tiempo integral. Tanto K_p como T_i son ajustables. El tiempo integral ajusta la acción de control integral, mientras que un cambio en el valor de K_p afecta las partes integral y

proporcional de la acción de control. El inverso del tiempo integral T_i se denomina velocidad de reajuste. La velocidad de reajuste es la cantidad de veces por minuto que se duplica la parte proporcional de la acción de control.

4.1.5 Acción de control proporcional-derivativa.

La acción de control de un controlador PD se define mediante:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

y la función de transferencia es

$$\frac{U(s)}{E(s)} = K_p(1 + T_d s)$$

en donde K_p es la ganancia proporcional y T_d es una constante denominada tiempo derivativo. Tanto K_p como T_d son ajustables. La acción de control derivativa, en ocasiones denominada control de velocidad, ocurre donde la magnitud de la salida del controlador es proporcional a la velocidad de cambio de la señal de error. El tiempo derivativo T_d es el intervalo de tiempo durante el cual la acción de la velocidad hace avanzar el efecto de la acción de control proporcional. Aunque la acción de control derivativo tiene la ventaja de ser de previsión, tiene las desventajas de que amplifica las señales de ruido y puede provocar un efecto de saturación en el actuador. La acción de control derivativa no se usa nunca sola, debido a que sólo es eficaz durante periodos transitorios.

4.1.6 Acción de control proporcional-integral-derivativa.

Esta acción PID tiene las ventajas de que cada una de las tres acciones de control individual. La ecuación se obtiene mediante

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

la función de transferencia es

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

en donde K_p es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo.

4.2 El control difuso.

Las consideraciones de las ventajas del control difuso son, primero que no es necesario un modelo matemático del sistema a controlar y segundo que puede ser desarrollado un controlador no-lineal satisfactorio.

Los controladores Mamdani SISO (Single Input-Single Output) una entrada una salida, son suficientes para nuestro propósito, controlar la velocidad de un motor de cd utilizando un sistema de control realimentado, son utilizados para controlar sistemas que pueden ser lineales o no lineales y su modelo puede ser conocido o no. En la práctica los controladores difusos usan computadoras o procesadores digitales para su implementación, por lo tanto los controladores difusos son controladores discretos.

Los componentes principales de un controlador difuso son Difusión (fuzzification), base de reglas difusas (fuzzy rule base), inferencia difusa (fuzzy inference) y desdifusión (defuzzification).

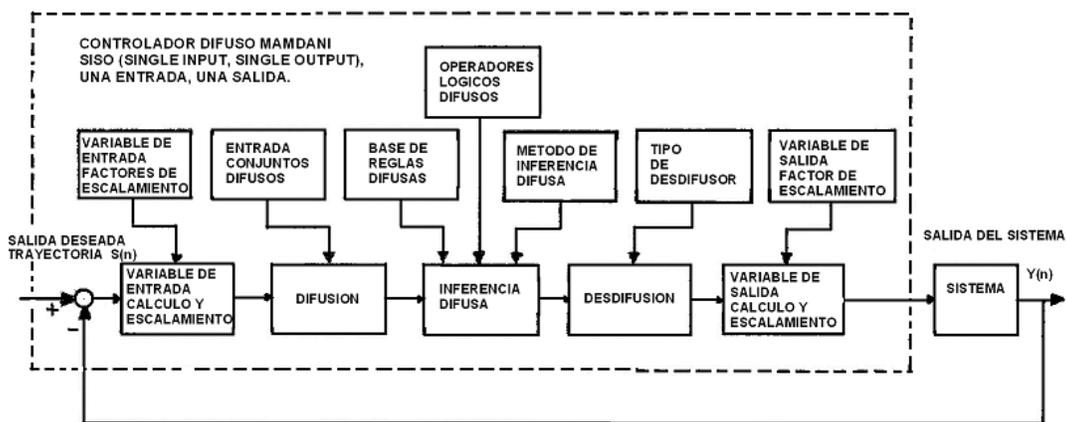


Figura 4.2 Sistema realimentado de control difuso Mamdani.

4.2.1 Difusión.

La salida del sistema se designa con $y(n)$, el muestreo en el tiempo es nT , donde T es el periodo de muestreo, la referencia es $S(n)$, la cual puede ser constante o variante en el tiempo.

El error y el cambio en el error, también llamado velocidad de cambio en el error (rate), son usados como variables de entrada, estas son

$$e(n) = S(n) - y(n)$$

$$r(n) = e(n) - e(n - 1) = y(n - 1) - y(n)$$

$e(n)$ y $r(n)$ tienen sus rangos, $[a1, b1]$ y $[a2, b2]$ los cuales corresponden a sus respectivos universos de discurso. Estos son llamados factores de escalamiento de las entradas, los llamaremos K_e y K_r , tenemos

$$E(n) = K_e e(n)$$

$$R(n) = K_r r(n)$$

Dos arreglos de conjuntos difusos son necesarios para ambos, estos son seleccionados de los diferentes tipos, ya sea triangular, trapezoidal, gaussiana o campana. El uso de Positivo o Negativo en las variables lingüísticas es necesario ya que $e(n)$ y $r(n)$ pueden ser positivos o negativos.

La difusión puede ser formulada matemáticamente, esto es las variables lingüísticas pueden ser remplazadas por un sistema de índices numéricos, donde podemos usar, por ejemplo:

$A_i, i = -2, -1, 1, 2$, los conjuntos difusos para $E(n)$ serian $\{A_{-2}, A_{-1}, A_1, A_2\}$ que simbolizan {Negativo Grande, Negativo Pequeño, Positivo Pequeño y Positivo Grande}, para $R(n)$ los conjuntos difusos serian $\{B_{-2}, B_{-1}, B_0, B_1, B_2\}$.

El numero de conjuntos difusos de las variables lingüísticas y sus etiquetas lingüísticas son parámetros determinados por el diseñador del controlador difuso. Existen en la literatura reglas y sugerencias de conocimientos empíricos acumulados por la práctica y el estudio.

Generalmente de 2 a 13 conjuntos difusos son usados para cada variable de entrada. Los conjuntos difusos deben cubrir por completo la escala del universo de discurso de tal manera que en cualquier valor de las variables de entrada se produzca un valor de membresía diferente de cero.

$$\mu_{A_2}(e(n^*)) = 0.6 \quad \mu_{A_1}(e(n^*)) = 0.2 \quad \mu_{A_{-1}}(e(n^*)) = 0 \quad \mu_{A_{-2}}(e(n^*)) = 0$$

4.2.2 Reglas Difusas.

La variable de salida puede ser una salida difusa de control $u(n)$ o un incremento en la salida difusa $\Delta u(n)$. El número de conjuntos difusos de salida esta en relación con el número de conjuntos difusos de entrada, si tenemos N_1 y N_2 conjuntos de entrada para $E(n)$ y $R(n)$

respectivamente tendremos $N_1 * N_2$ conjuntos de salida. En la práctica pueden ser menos de este máximo. La regla general difusa puede ser expresada como

$$\text{IF } E(n) \text{ is } A_i \text{ AND } R(n) \text{ is } B_j \text{ THEN } u(n) \text{ (o } \Delta u(n)) \text{ is } W_k$$

W_k es el conjunto difuso de salida. En el control difuso los operadores AND mas usados son el operador Zadeh AND y el operador producto AND. Teóricamente una combinación de diferentes operadores difusos pueden ser usados en diferentes reglas difusas and/or si tenemos mas de tres variables de entrada.

4.2.3 Inferencia Difusa.

Cuando una información especifica es asignada a las variables de entrada en el antecedente de la regla, la inferencia difusa es necesaria para calcular el valor para las variables de salida en el consecuente de la regla (THEN $u(n)$ (o $\Delta u(n)$) is W_k). En muchos casos los controladores difusos Mamdani usan en el consecuente de la regla conjuntos difusos singleton.

Los valores de membresía obtenidos por la difusión, primero son combinados por la operación lógica difusa AND u OR en el antecedente de la regla. El resultado es entonces relacionado al conjunto difuso de salida por la inferencia

difusa. El método más usado en control difuso es la inferencia Mamdani del mínimo (Mamdani minimum inference method).

Los operadores lógicos difusos más usados son:

Operador lógico difuso Zadeh AND: $\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu_B(x))$
 Operador lógico producto AND: $\mu_{A \wedge B}(x) = (\mu_A(x) \times \mu_B(x))$
 Operador lógico difuso Zadeh OR: $\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x))$
 Operador lógico difuso Luckasiewicz OR: $\mu_{A \vee B}(x) = \min(\mu_A(x) + \mu_B(x), 1)$

Los métodos de Inferencia difusa mas usados son:

Mamdani minimum inference, RM: $\min(\mu, \mu_W(z))$, para toda z
 Larsen product inference, RL: $\mu * \mu_W(z)$, para toda z
 Drastic product inference, RDP: μ , para $\mu_W(z) = 1$
 $\mu_W(z)$, para $\mu = 1$
 0, para $\mu < 1$ y $\mu_W(z) < 1$
 Bounded product inference, RBD: $\max(\mu + \mu_W(z) - 1, 0)$

$\mu_W(z)$ es la función de membresía del conjunto difuso W que representa W_k en el consecuente de la regla, μ es la membresía final obtenida por los operadores lógicos difusos AND en el antecedente de la regla.

4.2.4 Desdifusión.

Es el proceso matemático para convertir un conjunto difuso o conjuntos difusos en un número real. Los valores de membresía generados por la inferencia difusa son combinados matemáticamente para obtener un solo número como la salida del controlador difuso. Los actuadores del sistema de control aceptan un solo valor como sus señales de entrada.

$U(n^*)$ es la salida del controlador difuso en el tiempo n^* . Si la desdifusión resultante es $\Delta U(n^*)$

La salida del controlador difuso sería

$$U(n^*) = U(n^* - 1) + \Delta U(n^*)$$

Las operaciones de difusión, inferencia difusa y desdifusión son repetidas en cada periodo de muestreo.

4.3 Controlador Mamdani difuso PID, PD Y PI.

El controlador lineal PID continuo en el tiempo es el siguiente

$$U(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

donde $e(t)$ es el error, K es la ganancia, T_i es el tiempo de integración y T_d es el tiempo derivativo.

La forma discreta en el tiempo es

$$U(n) = K \left(e(n) + \frac{T}{T_i} \sum_{i=0}^n e(i) + \frac{T_d}{T} r(n) \right) = K e(n) + \frac{KT}{T_i} \sum_{i=0}^n e(i) + \frac{KT_d}{T} r(n)$$

$$= K_p e(n) + K_i \sum_{i=0}^n e(i) + K_d r(n)$$

donde $r(n)$ es la velocidad de cambio del error, ya definida, y T es el periodo de muestreo. K_p , K_d y K_i , son la ganancia proporcional, la ganancia integral y la ganancia derivativa del controlador PID. Las tres ganancias son constantes para un controlador lineal PID, si el valor de al menos una de las ganancias varía con el tiempo, el controlador PID sería no lineal. Los algoritmos del controlador PID son en forma posicional porque ellos calculan directamente la salida del controlador. El controlador PID es también usado en forma incremental, este calcula cambios en la salida del controlador. En el tiempo $(n - 1)$ tenemos

$$U(n - 1) = K_p e(n - 1) + K_i \sum_{i=0}^{n-1} e(i) + K_d r(n - 1).$$

por lo tanto la forma incremental del controlador PID correspondiente es

$$\Delta U(n) = U(n) - U(n - 1) = K_p r(n) + K_i e(n) + K_d d(n)$$

donde

$$d(n) = r(n) - r(n - 1)$$

En la práctica un controlador PID algunas veces no es conveniente. Por lo tanto, un control parcial PID, en forma de control PI o PD es más efectivo y apropiado. Esto es porque el término derivativo tiende a amplificar el ruido. Por otro lado, el término integral puede causar una respuesta lenta del sistema y una respuesta mayor a la referencia del sistema (overshoot).

Cuando K_d es cero, el controlador PID será un controlador PI en forma incremental

$$\Delta U(n) = K_p r(n) + K_i e(n)$$

si $K_i = 0$, el controlador PID se reduce a un controlador PD en forma incremental

$$\Delta U(n) = K_p r(n) + K_d d(n)$$

Un controlador PI en forma incremental es relacionado con un controlador PD en forma posicional. Si hacemos $K_i = 0$ en la ecuación del control PID en forma posicional, obtenemos un controlador PD en forma posicional.

$$U(n) = K_p e(n) + K_d r(n)$$

Comparando las ecuaciones podemos ver que un controlador PD en forma posicional deviene en un controlador PI en forma incremental si, $e(n)$ y $r(n)$ intercambian posiciones, K_d es remplazada por K_i y $U(n)$ es remplazada por $\Delta U(n)$. También podemos observar que un controlador PI en forma incremental deviene en un controlador PD en forma incremental si $e(n)$ es remplazada por $d(n)$ y K_i es remplazada por K_d .

4.4 Simulación del sistema de control difuso.

Para la simulación del control de velocidad de un motor de DC usamos las herramientas propuestas en el capítulo anterior, que son la herramienta de lógica difusa y la herramienta simulink de Matlab.

Las consideraciones para nuestro sistema de control realimentado son:

- 1) El arranque del motor, en el cual se debe considerar el flujo de corriente para que no se dañe el motor.
- 2) Alcanzar la velocidad deseada en el menor tiempo posible con una carga inicial.
- 3) Mantener la velocidad sin fluctuaciones con respecto a nuestro punto de referencia.
- 4) Al variar la velocidad, es decir el punto de referencia obtener la respuesta deseada.
- 5) Si incrementamos la carga que el disturbio en la velocidad sea lo mas corto posible con respecto al tiempo y que mantenga el punto de referencia en la velocidad.

Ponemos como referencia el controlador de un motor en el cual usamos la acción de control proporcional integral (PI).

4.4.1 Controlador del motor usando SimPowerSystem (simulador de sistemas de potencia) de Matlab.

El diagrama del manejador del sistema es construido con bloques de la librería powerlib combinada con bloques de Simulink. La operación del motor de cd es manejada mediante el control del voltaje de armadura con tristor chopper GTO.

Un diagrama simplificado del sistema es mostrado en la figura 4.3. El motor es alimentado por una fuente de CD a través de un chopper (recortador) que consiste de un tristor GTO $Th1$ y un diodo de transporte libre (free-wheeling) $D1$. El motor de CD maneja una carga mecánica la cual es caracterizada por la inercia J , el coeficiente de fricción B y la carga torque T_L .

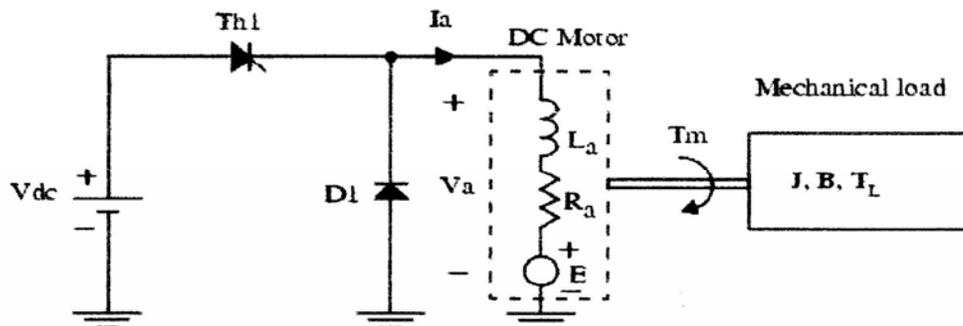


Fig. 4.3 Sistema de control del motor de CD.

En el diagrama anterior el motor es representado por su circuito equivalente que consiste de un inductor L_a y una resistencia R_a en serie con la fuerza contraelectromotriz (emf) E .

La emf es proporcional a la velocidad del motor

$$E = K_E w$$

donde K_E es una constante del voltaje del motor y w es la velocidad del motor. En una maquina de CD excitada por separado, la constante del voltaje del motor K_E es proporcional a la corriente de campo I_f

$$K_E = L_{af} I_f$$

L_{af} es la inductancia mutua del campo y la armadura.

El torque desarrollado por el motor es proporcional a la corriente de armadura I_a

$$T_m = K_T I_a$$

K_T es la constante del torque del motor. La constante del torque del motor es igual a la constante del voltaje

$$K_T = K_E$$

el tristor $Th1$ es disparado por una señal PWM modulación de ancho de pulso (pulse-width-modulated) para controlar el promedio del voltaje del motor. La operación es mostrada en la figura 4.4.

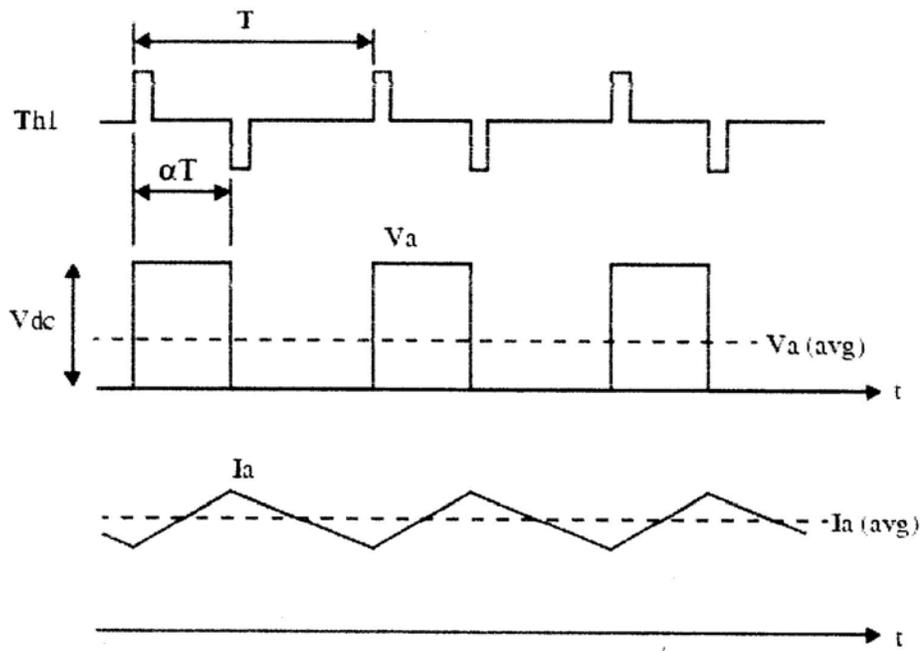


Fig. 4.4 Operación del chopper (pulsador).

El voltaje promedio de armadura es una función directa del ciclo del chopper α .

$$V_{a(\text{avg})} = \alpha V_{\text{dc}}$$

Esta relación es válida solo cuando la corriente de armadura es continua. En estado estable, el promedio del voltaje de armadura es igual a

$$I_{a(\text{avg})} = (V_{a(\text{avg})} - E) / R_a$$

La figura 4.5 ilustra el controlador.

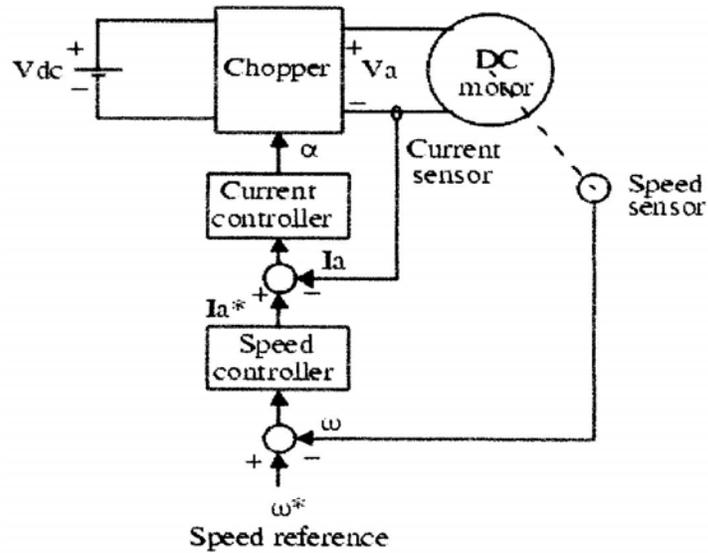


Fig. 4.5 Controlador de velocidad variable de un motor de CD.

El torque del motor es controlado por la corriente de armadura I_a , la cual es regulada por un loop de control de corriente. La velocidad del motor es controlada por un loop externo, el cual provee la corriente de referencia I_a^* para el loop de control de corriente.

4.4.2 El controlador discreto PI.

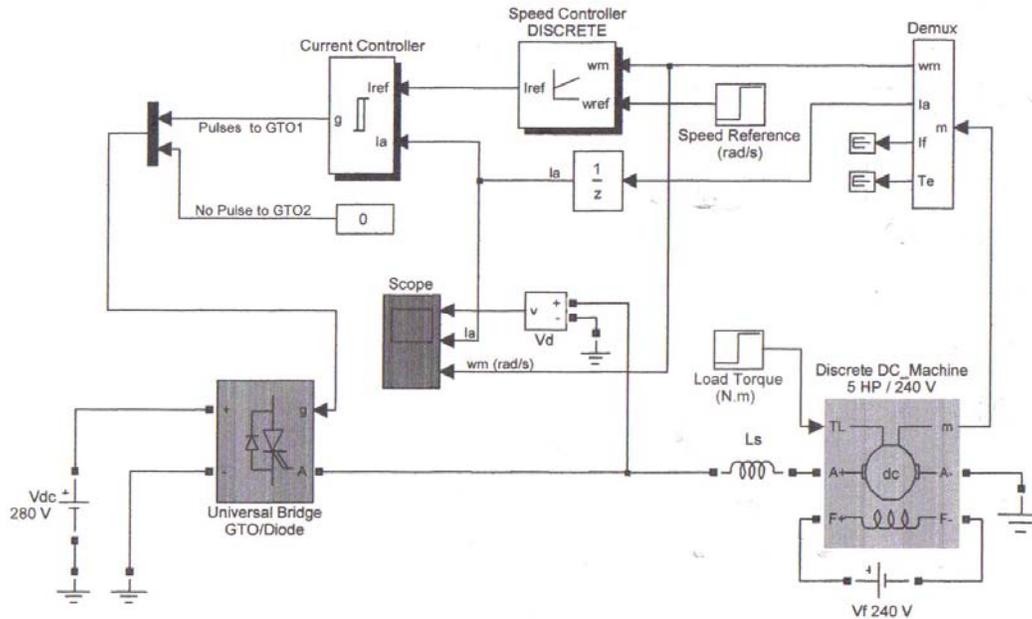
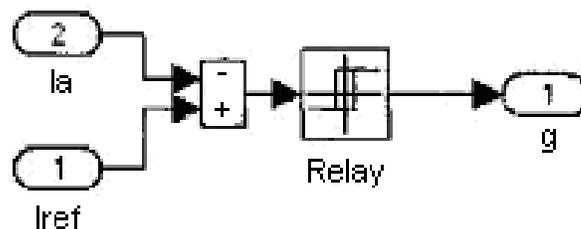


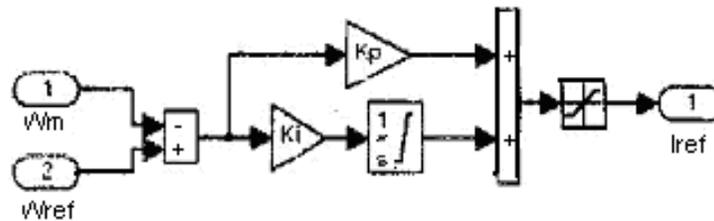
Fig. 4.6 Sistema de control para un motor de CD con controlador clásico PI.

El motor usado es una máquina de cd con excitación separada, de 5 HP/240V con los siguientes parámetros, $R_a = 0.5$ omhs, $L_a = 10$ mH, $K_E = 1.23$ V / (rad/s), $K_T = 1.23$ N m / A.

Un inductor $L_s = 10$ mH es conectado en serie con el motor para suavizar la salida de la corriente de armadura. La excitación constante es implementada por la conexión de un bloque de una fuente de CD al enrollamiento de campo. La señal requerida de disparo para el tristor GTO es generada por un controlador de corriente de histéresis, el cual fuerza la corriente del motor a seguir la referencia entre los limites $+ h/2$ y $- h/2$, donde h es la banda de histéresis. El controlador de corriente es un bloque enmascarado que contiene



El loop de control de la velocidad usa un controlador PI, proporcional integral, el cual es implementado por los siguientes bloques



Debido a que no se pueden discretizar dispositivos electrónicos que usan conmutación forzada como los GTO cuando estos son simulados por bloques individuales, usamos el bloque Universal Bridge, para implementar el GTO y el diodo. El sistema de control y la maquina de cd son discretizados usando el mismo periodo de muestreo, $T_s = 10 \cdot 10^{-6} = 10$ microsegundos. Los parámetros del controlador PI son $K_p = 1.6$, $K_i = 16$ y el bloque tiene un limitador de corriente de 30 A.

El motor arranca con una carga que requiere un Torque de 5 Nm, la velocidad de referencia a la que debe llegar y mantener es de 120 rad/s. A los 5 segundos la referencia en la velocidad cambia de 120 a 160 rad/s y a los 7 seg. la carga varía de 5 a 25 Nm. En la grafica del comportamiento del sistema, figura 4.7, vemos el voltaje de armadura V_a (V), la corriente de armadura I_a (A) y la velocidad angular ω_m (rad/s).

La velocidad de 120 rad/s la alcanza en 0.58 seg en los que vemos que la corriente de armadura (I_a), es la máxima permitida, debido al limitador de corriente implementado en el bloque de control, al llegar al estado estable en la velocidad mantiene una corriente promedio de 5.5 A, al cambiar el punto de referencia en la velocidad de 120 a 160 rad/s la corriente aumenta nuevamente hasta el máximo de 30 A y alcanza la velocidad en 0.49 seg. manteniendo una corriente promedio de 6.5 A y por ultimo, al variar la carga tenemos un disturbio en la velocidad con un decremento en ella, para luego volver al punto de referencia 160 rad/s, este dura 0.21 seg. y la corriente promedio es de 24 A. Estos datos los manejaremos en la siguiente tabla como referencia para el controlador con lógica difusa que implementaremos.

	Tiempo de respuesta y tiempo de disturbio	Corriente de armadura	
		I_a máx.(avg)	I_a (avg)
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	30 A	5.37 A
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	30 A	6.75 A
Cambio en la carga (5 a 25 Nm)	0.25 seg.	25 A	23.83 A

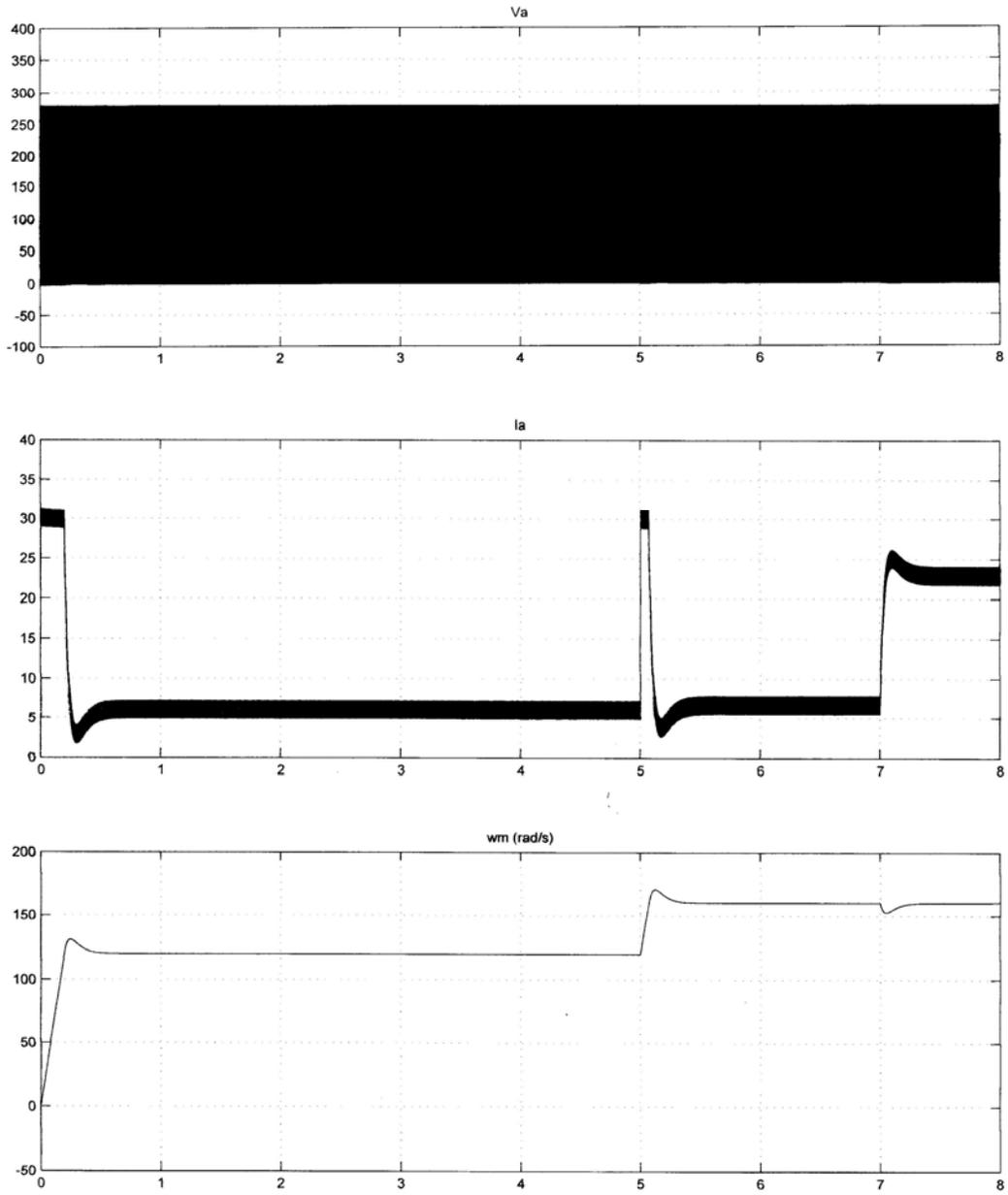


Fig. 4.7 Respuesta del controlador clásico PI.

4.4.3 Voltaje promedio $V(\text{avg})$ y corriente promedio $I(\text{avg})$.

La velocidad en rpm = $(\text{rad/s} * 60 \text{ s}) / (2 * \pi)$, por lo tanto la velocidad en revoluciones por minuto, para:

$$(120 \text{ rad/s} * 60 \text{ s}) / (2 * \pi) = 1145.913 \text{ rpm}$$

y para:

$$(160 \text{ rad/s} * 60 \text{ s}) / (2 * \pi) = 1527.884 \text{ rpm}$$

La fuerza contraelectromotriz (EMF), la calculamos para una velocidad angular de 120 rad/s:

$$E = K_E \omega = 1.23 (\text{ V / (rad/s) }) * 120 (\text{ rad/s}) = 147.6 \text{ V}$$

y para 160 rad/s:

$$E = K_E \omega = 1.23 (\text{ V / (rad/s) }) * 160 (\text{ rad/s}) = 196.8 \text{ V}$$

Como el torque es proporcional a la corriente de armadura:

$$T_m = K_T I_a$$

Obtenemos la corriente de armadura para un torque de 5 N m:

$$I_a = T_m / K_T = 5 (\text{ N m}) / 1.23 (\text{ N m / A}) = 4.065 \text{ A}$$

Y la corriente de armadura para un torque de 25 N m:

$$I_a = T_m / K_T = 25 (\text{ N m}) / 1.23 (\text{ N m / A}) = 20.325 \text{ A}$$

Para calcular el voltaje promedio y la corriente promedio, obtenemos de las gráficas de voltaje y corriente, el valor del periodo (T) y del ciclo de operación del chopper (α).

Para una velocidad de 120 rad/s y una carga de 5 N m, tenemos de la figura 4.8:

$$T = 0.667 \text{ ms}; \quad \alpha T = 0.358 \text{ ms};$$

$$\text{por lo tanto} \quad \alpha = 0.358 \text{ ms} / 0.667 \text{ ms} = 0.5367316$$

El voltaje promedio de armadura es:

$$V_{a(\text{avg})} = \alpha V_{dc} = 0.5567316 * 280 \text{ V} = 150.285 \text{ V}$$

En estado estable, el promedio de la corriente de armadura es igual a:

$$I_{a(\text{avg})} = (V_{a(\text{avg})} - E) / R_a = (150.285 \text{ V} - 147.6 \text{ V}) / 0.5 \text{ ohms} = 5.37 \text{ A}$$

Para una velocidad de 160 rad/s y una carga de 25 N m, tenemos de la figura 4.9:

$$T = 0.8944 \text{ ms}; \quad \alpha T = 0.6667 \text{ ms};$$

por lo tanto $\alpha = 0.6667 \text{ ms} / 0.8944 \text{ ms} = 0.7454159$

El voltaje promedio de armadura es:

$$V_{a(\text{avg})} = \alpha V_{\text{dc}} = 0.7454159 * 280 \text{ V} = 208.716 \text{ V}$$

y el promedio de la corriente de armadura es igual a:

$$I_{a(\text{avg})} = (V_{a(\text{avg})} - E) / R_a = (208.716 \text{ V} - 196.8 \text{ V}) / 0.5 \text{ ohms} = 23.83 \text{ A}$$

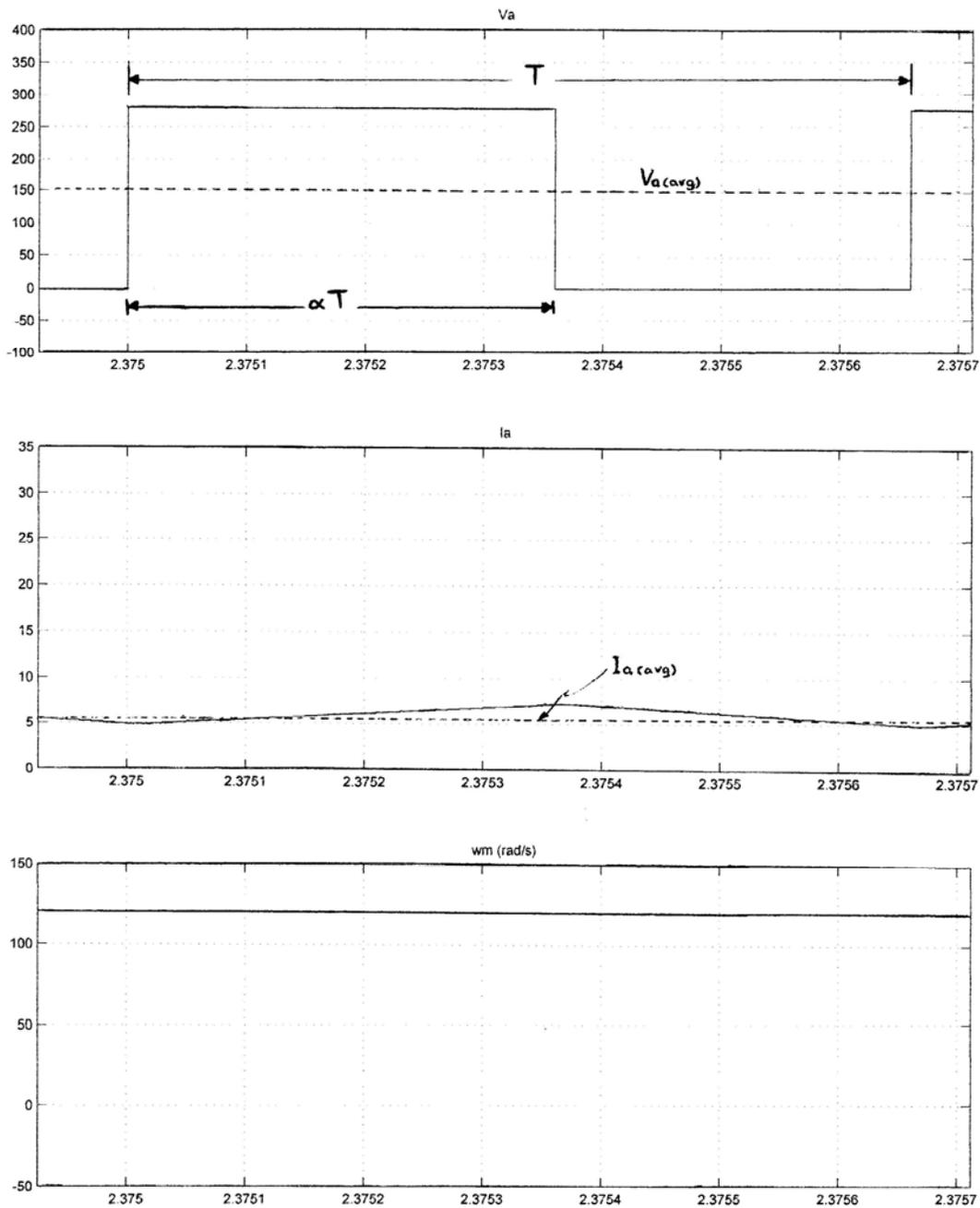


Fig. 4.8 Voltaje promedio $V_{a(\text{avg})}$ y corriente promedio $I_{a(\text{avg})}$ del controlador PI a 120 rad/s y 5 N m.

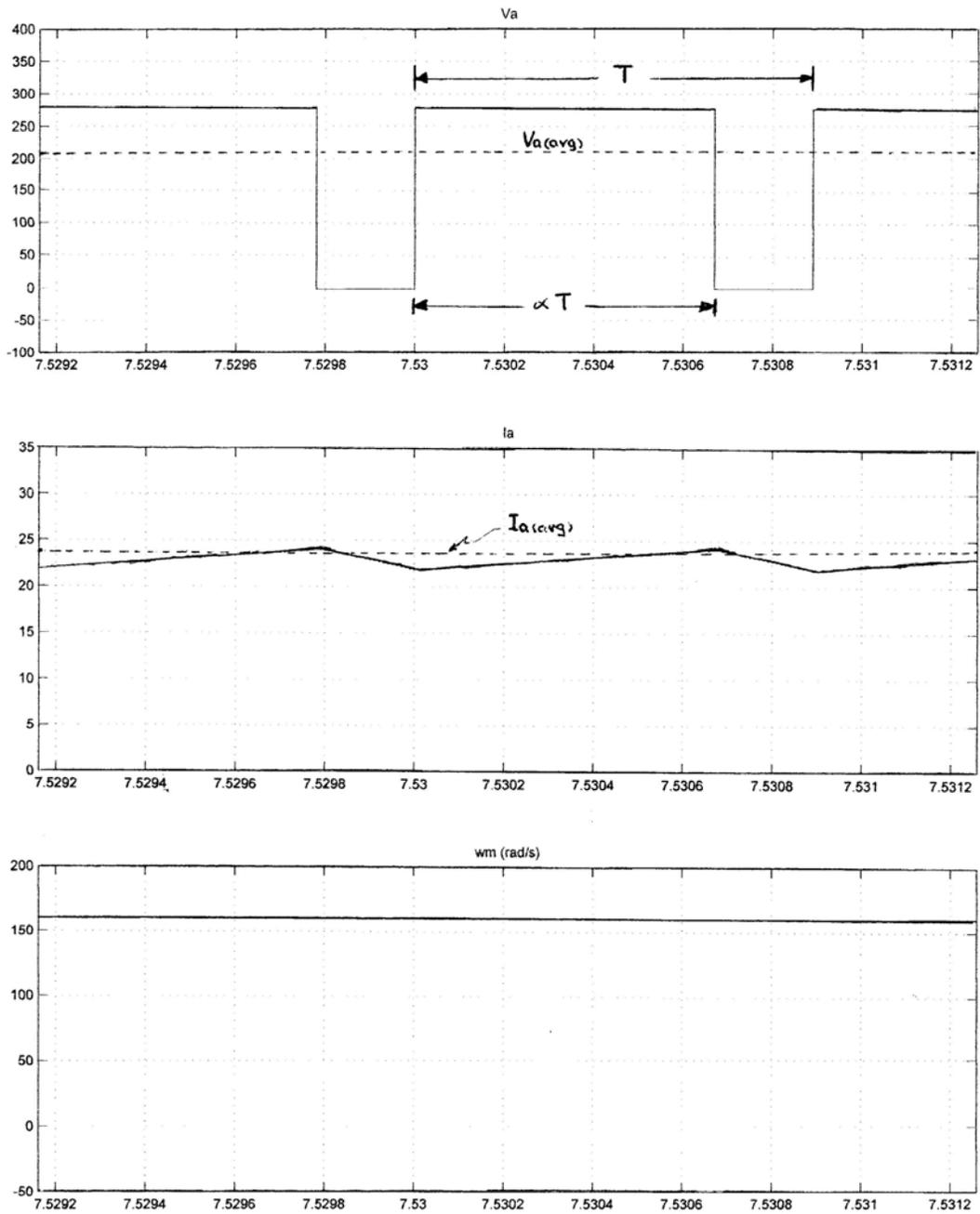


Fig. 4.9 Voltaje promedio $V_a(\text{avg})$ y corriente promedio $I_a(\text{avg})$ del controlador PI a 160 rad/s y 25 N m.

4.4.4 El controlador difuso PI Mamdani.

Usamos el mismo sistema de control descrito anteriormente, únicamente sustituiremos el bloque de control proporcional integral por un bloque de control con lógica difusa. Para el bloque de lógica difusa tenemos como variables de entrada

$$e(n) = S(n) - y(n)$$

$$r(n) = e(n) - e(n - 1)$$

La señal de error $e(n)$, la implementamos en un bloque donde obtenemos la diferencia de la velocidad de referencia $S(n)$ menos la velocidad real del motor $y(n)$.

La velocidad de cambio del error (ratio) $r(n)$, la obtenemos al procesar la señal de error, tenemos dos bloques, uno es una memoria con valor inicial 0 que almacena el valor $e(n)$ durante un periodo de muestreo para obtener $e(n - 1)$, estas dos señales las pasamos por un segundo bloque donde restamos del error el valor de la memoria 1, es decir $e(n - 1)$.

Estas dos señales, $e(n)$ y $r(n)$ entran al bloque de control de lógica difusa a través de un multiplexor 1. La salida del bloque de control difuso es en forma incremental $\Delta U(n)$, esto implica que la señal de control será

$$U(n) = U(n - 1) + \Delta U(n)$$

Por lo tanto, ponemos una memoria 2 con las mismas características, con un valor inicial de 0, a la salida de la señal de control para obtener $U(n - 1)$ y la sumamos con la salida del bloque de control de lógica difusa $\Delta U(n)$, para obtener la señal de control $U(n)$. En la figura 4.10 vemos el sistema para el control del motor.

La implementación del bloque de control difuso se realiza por separado en la herramienta de lógica difusa y luego en Simulink se abre el bloque y se pone el nombre del archivo que se implemento.

Implementación del bloque: controlador de lógica difusa.

De acuerdo a la literatura se pueden usar para las variables de entrada $e(n)$ y $r(n)$ de 2 a N conjuntos difusos y para la variable de salida de 3 a N. Para realizar los controladores se toma en cuenta que la velocidad máxima a controlar sería aproximadamente de 200 rad/s y que el controlador difuso PI es en forma incremental, por lo tanto calculara la variación en la señal de salida del controlador. Para no generar confusiones, usaremos los valores reales del universo de discurso para las variables tanto de entrada como de salida, ya que se tendrá que probar con variaciones en el rango de la variable de salida, debido a esto supondremos los valores de escalamiento de $e(n)$, $r(n)$ y $\Delta U(n)$ con el valor de 1. Estos controladores son los mejores que se obtuvieron después de haberlos sintonizado.

4.4.5 Primer controlador (49 reglas difusas).

Usamos 7 conjuntos para las variables de entrada, tanto para $e(n)$ como para $r(n)$, y 7 conjuntos para la variable de salida.

Para $e(n)$, $r(n)$ y $\Delta U(n)$ las etiquetas lingüísticas serán NB (Negativo Grande), NM (Negativo Medio), NS (Negativo Pequeño), ZO (Tiende a Cero), PS (Positivo Pequeño), PM (Positivo Medio), PB (Positivo Grande). De acuerdo a la tabla de la figura 4.11 se generan las reglas difusas, que en este caso serían $7 \times 7 = 49$.

Las características del bloque de control difuso elaborado con la herramienta de lógica difusa se muestran en las figuras 4.12 a 4.16.

Las 49 reglas del archivo mamdani7 que generamos en el Editor son las siguientes:

1. IF (e is NB) AND (Ae is nb) THEN (Au is ANB)
2. IF (e is NB) AND (Ae is nm) THEN (Au is ANB)
3. IF (e is NB) AND (Ae is ns) THEN (Au is ANB)
4. IF (e is NB) AND (Ae is zo) THEN (Au is ANB)
5. IF (e is NB) AND (Ae is ps) THEN (Au is ANM)
6. IF (e is NB) AND (Ae is pm) THEN (Au is ANS)
7. IF (e is NB) AND (Ae is pb) THEN (Au is AZO)

8. IF (e is NM) AND (Ae is nb) THEN (Au is ANB)
9. IF (e is NM) AND (Ae is nm) THEN (Au is ANB)
10. IF (e is NM) AND (Ae is ns) THEN (Au is ANB)
11. IF (e is NM) AND (Ae is zo) THEN (Au is ANM)
12. IF (e is NM) AND (Ae is ps) THEN (Au is ANS)
13. IF (e is NM) AND (Ae is pm) THEN (Au is AZO)
14. IF (e is NM) AND (Ae is pb) THEN (Au is APS)

15. IF (e is NS) AND (Ae is nb) THEN (Au is ANB)
16. IF (e is NS) AND (Ae is nm) THEN (Au is ANB)
17. IF (e is NS) AND (Ae is ns) THEN (Au is ANM)
18. IF (e is NS) AND (Ae is zo) THEN (Au is ANS)
19. IF (e is NS) AND (Ae is ps) THEN (Au is AZO)
20. IF (e is NS) AND (Ae is pm) THEN (Au is APS)
21. IF (e is NS) AND (Ae is pb) THEN (Au is APM)

22. IF (e is ZO) AND (Ae is nb) THEN (Au is ANB)
23. IF (e is ZO) AND (Ae is nm) THEN (Au is ANM)
24. IF (e is ZO) AND (Ae is ns) THEN (Au is ANS)
25. IF (e is ZO) AND (Ae is zo) THEN (Au is AZO)
26. IF (e is ZO) AND (Ae is ps) THEN (Au is APS)
27. IF (e is ZO) AND (Ae is pm) THEN (Au is APM)
28. IF (e is ZO) AND (Ae is pb) THEN (Au is APB)

29. IF (e is PS) AND (Ae is nb) THEN (Au is ANM)
30. IF (e is PS) AND (Ae is nm) THEN (Au is ANS)
31. IF (e is PS) AND (Ae is ns) THEN (Au is AZO)
32. IF (e is PS) AND (Ae is zo) THEN (Au is APS)
33. IF (e is PS) AND (Ae is ps) THEN (Au is APM)
34. IF (e is PS) AND (Ae is pm) THEN (Au is APB)
35. IF (e is PS) AND (Ae is pb) THEN (Au is APB)

36. IF (e is PM) AND (Ae is nb) THEN (Au is ANS)
37. IF (e is PM) AND (Ae is nm) THEN (Au is AZO)
38. IF (e is PM) AND (Ae is ns) THEN (Au is APS)
39. IF (e is PM) AND (Ae is zo) THEN (Au is APM)
40. IF (e is PM) AND (Ae is ps) THEN (Au is APB)
41. IF (e is PM) AND (Ae is pm) THEN (Au is APB)
42. IF (e is PM) AND (Ae is pb) THEN (Au is APB)

43. IF (e is PB) AND (Ae is nb) THEN (Au is AZO)
44. IF (e is PB) AND (Ae is nm) THEN (Au is APS)
45. IF (e is PB) AND (Ae is ns) THEN (Au is APM)
46. IF (e is PB) AND (Ae is zo) THEN (Au is APB)
47. IF (e is PB) AND (Ae is ps) THEN (Au is APB)
48. IF (e is PB) AND (Ae is pm) THEN (Au is APB)
49. IF (e is PB) AND (Ae is pb) THEN (Au is APB)

Controlador de un motor de DC (Discreto)

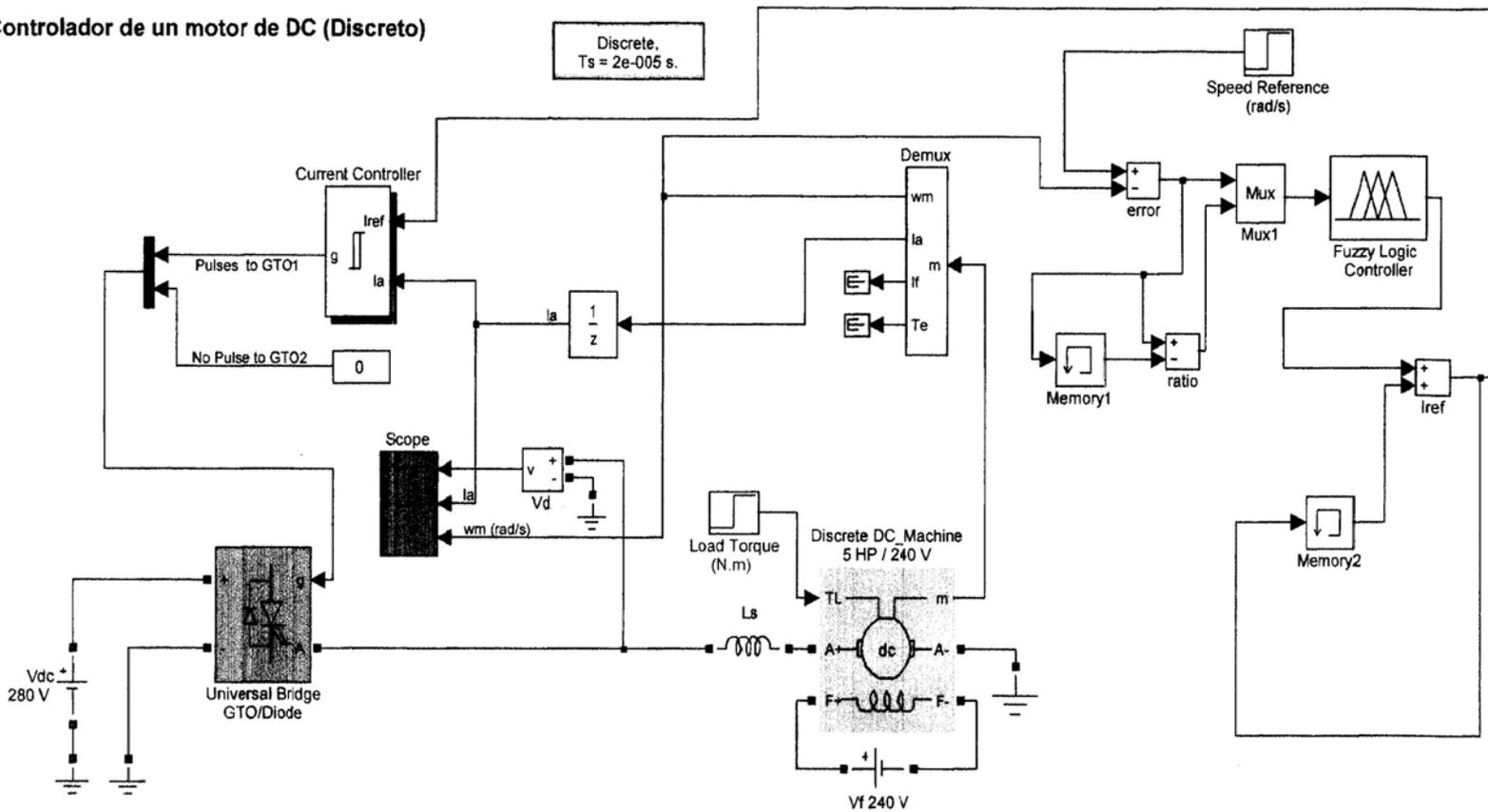
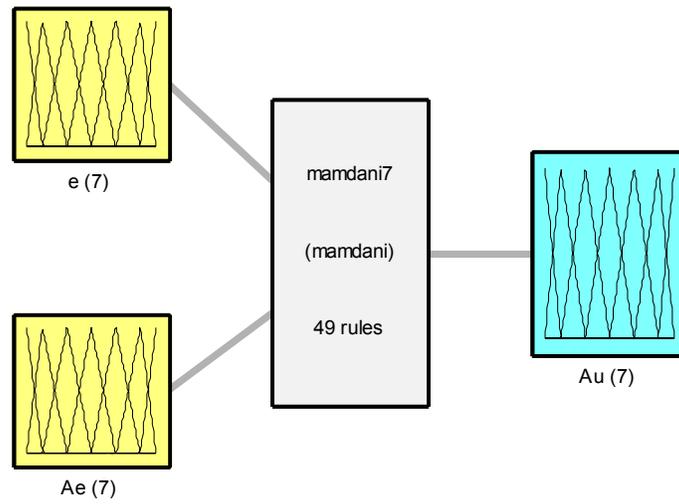


Fig. 4.10 Sistema de control para un motor de cd con lógica difusa.

Δe	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PS	PB	PB	PB

Fig. 4.11 Base de reglas difusas en forma tabular.



System mamdani7: 2 inputs, 1 outputs, 49 rules

Fig. 4.12 Características del controlador difuso mamdani7.

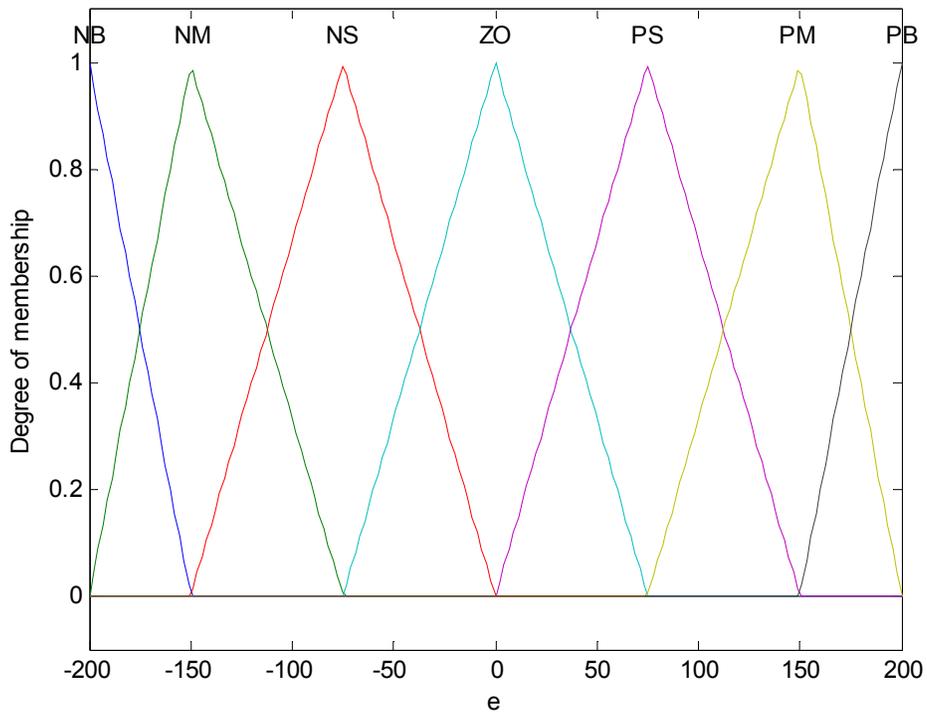


Fig. 4.13 Funciones de membresía para $e(n)$.

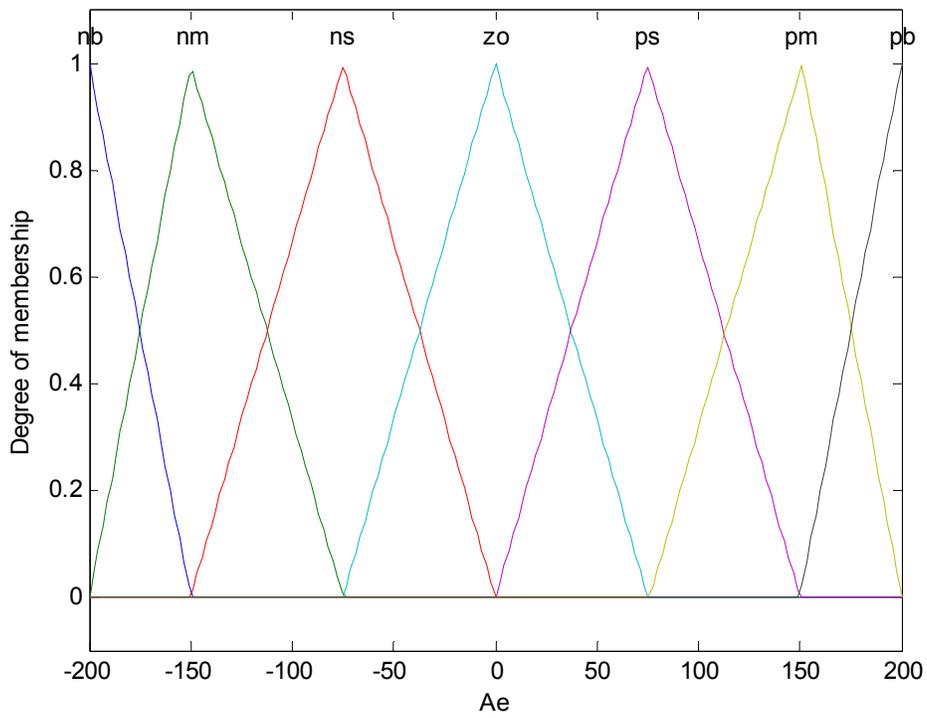


Fig. 4.14 Funciones de membresía para $r(n)$.

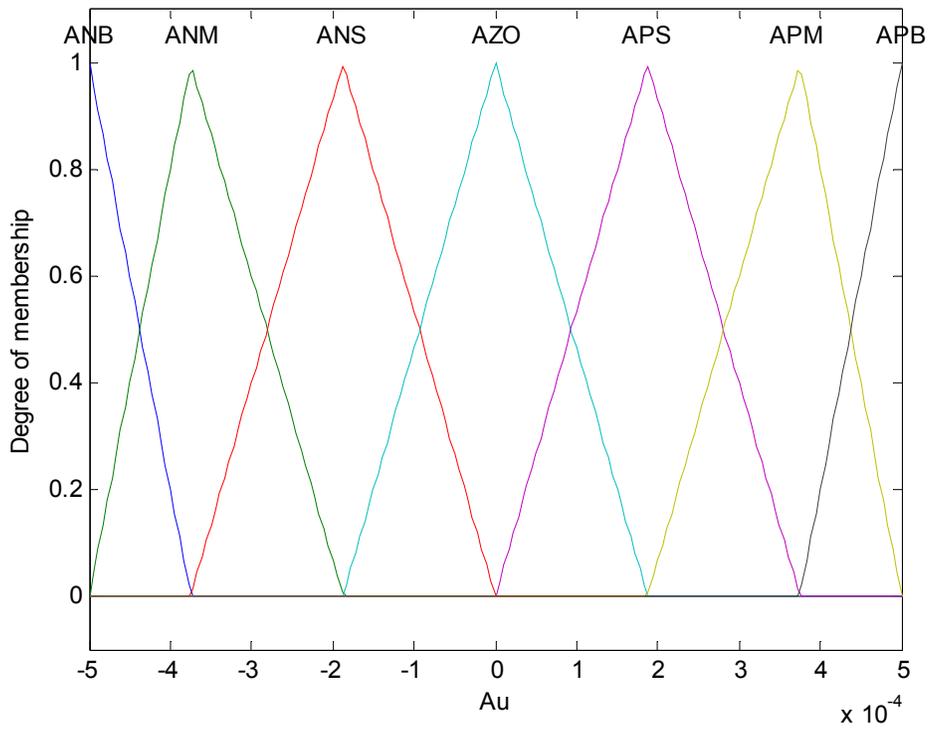


Fig. 4.15 Funciones de membresía para $\Delta U(n)$.

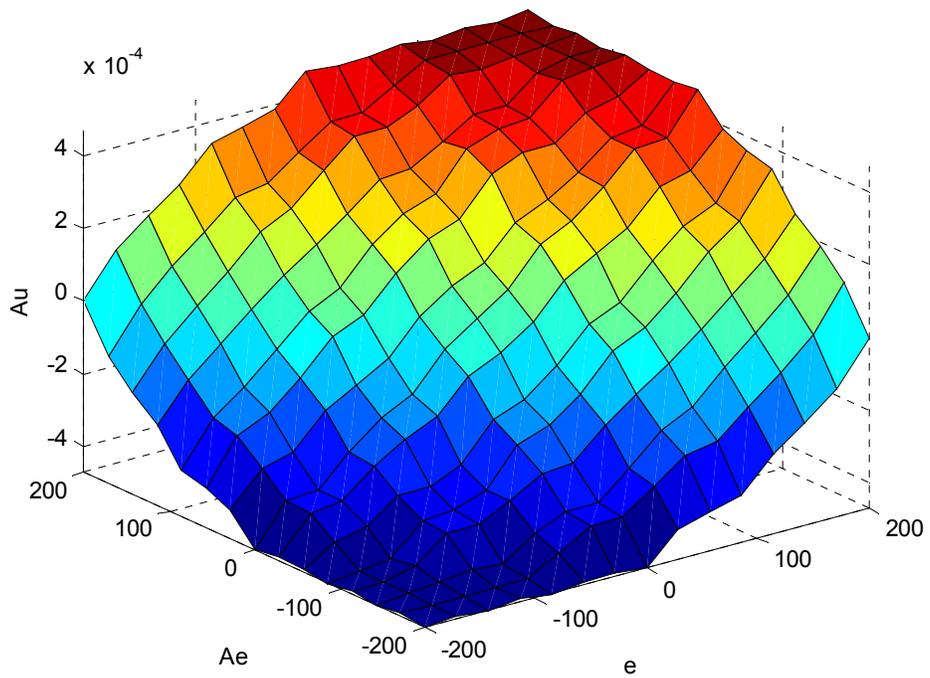


Fig. 4.16 Superficie de control generada.

La respuesta del sistema de control difuso PI la vemos en la figura 4.17.

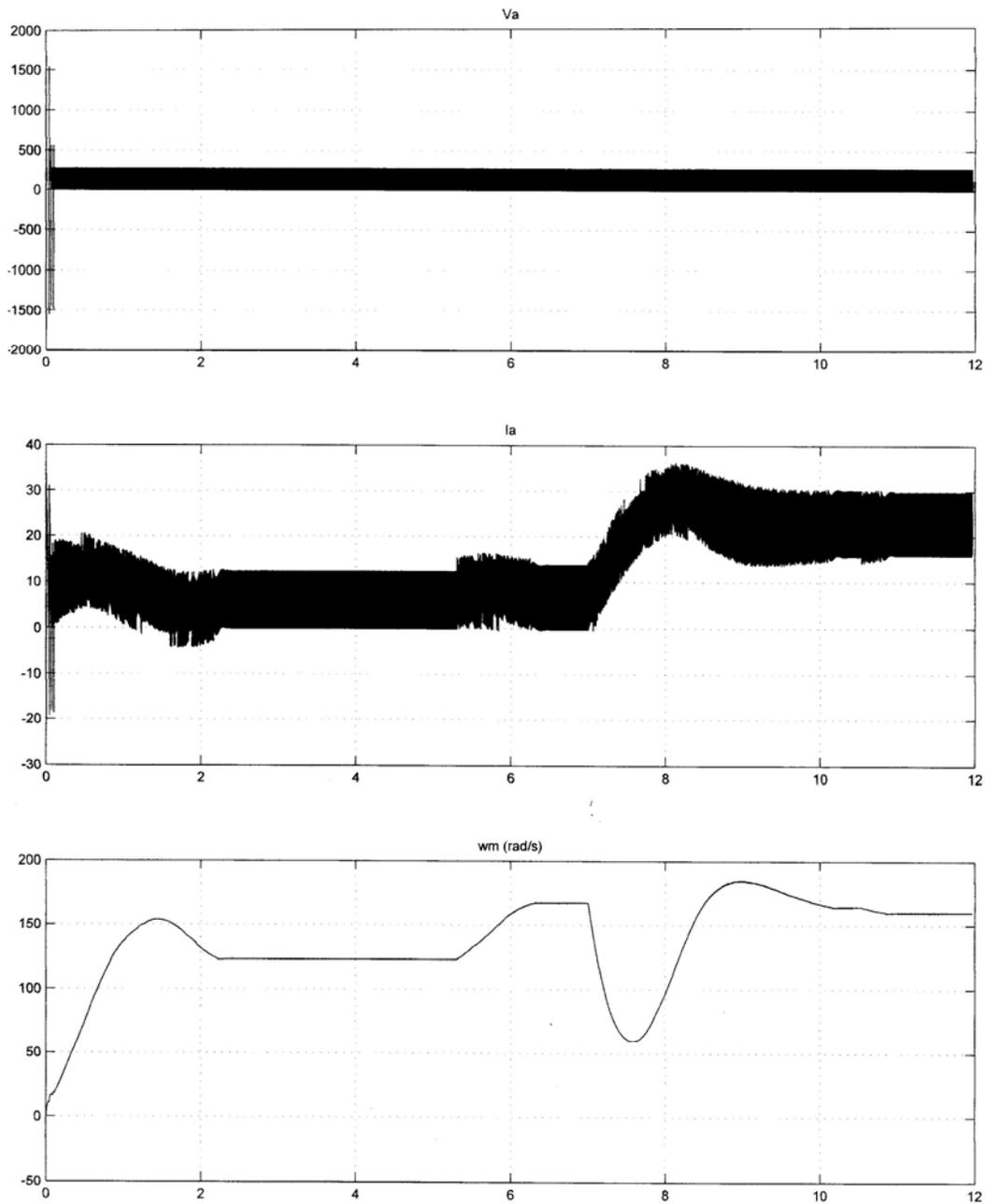


Fig. 4.17 Respuesta del controlador difuso PI, mamdani7.

La variación en los operadores lógicos y en la inferencia no determina mejoras en el comportamiento, estas fueron más sensibles al método de desdifusión. En este controlador se usó bisección, la variación en los conjuntos difusos no implicó mejoras, la sintonización se logró con el periodo de muestreo $T_s = 0.0005 \text{ s} = 0.5 \text{ ms}$ y el método de desdifusión. Como podemos observar, aun cuando el tiempo de respuesta en alcanzar la primer velocidad de referencia en un estado estable es de 2.32 seg., esta tiene una diferencia de 3 rad/s (offset),

por lo que la velocidad es de 123 rad/s, en cuanto al segundo punto de referencia tarda 1.3 seg. y mantiene una diferencia de 7 rad/s mas, esto es, tiene una velocidad de 167 rad/s.

Al cambiar la carga el disturbio provoca que la velocidad disminuya a 60 rad/s y luego sube hasta 185 rad/s, para luego llegar a un estado estable en su velocidad de referencia después de 3.9 seg.

	Tiempo de respuesta y tiempo de disturbio	Corriente de armadura	
		$I_{a(máx.(avg))}$	$I_{a(avg)}$
Velocidad de Ref. 1 (120 rad/s)	2.32 seg.	12 A	6.5 A
Velocidad de Ref. 2 (160 rad/s)	1.30 seg.	9 A	7.5 A
Cambio en la carga (5 a 25 Nm)	3.90 seg.	28 A	23 A

Se hicieron variaciones en cuanto al número de conjuntos de las variables de entrada y de las variables de salida, así como en el número de reglas sin obtener un cambio sensible en el comportamiento, por lo cual se procedió a implementar el controlador sugerido en la referencia [Fuzzy Control and Modeling, Hao Ying].

4.4.6 Segundo controlador (4 reglas difusas), conjuntos difusos de salida triangulares.

El controlador difuso usa 2 conjuntos difusos idénticos, llamados positivo y negativo, para las variables de entrada escaladas $E(n)$ y $R(n)$. Las funciones de membresía para los conjuntos difusos son, para $E(n)$

$$\mu_P(e) = \begin{cases} 0, & E(n) < -L \\ (K_e e(n) + L) / (2L), & -L \leq E(n) \leq L \\ 1, & E(n) > L \end{cases}$$

$$\mu_N(e) = \begin{cases} 1, & E(n) < -L \\ (-K_e e(n) + L) / (2L), & -L \leq E(n) \leq L \\ 0, & E(n) > L \end{cases}$$

y las funciones de membresía para $R(n)$ son

$$\mu_P(r) = \begin{cases} 0, & R(n) < -L \\ (K_r r(n) + L) / (2L), & -L \leq R(n) \leq L \\ 1, & R(n) > L \end{cases}$$

$$\mu_N(r) = \begin{cases} 1, & R(n) < -L \\ (-K_r r(n) + L) / (2L), & -L \leq R(n) \leq L \\ 0, & R(n) > L \end{cases}$$

para la salida usamos tres conjuntos difusos triangulares, negativo, cero y positivo. Usando el mismo criterio, ponemos el valor de los factores de escalamiento unitario.

Este controlador difuso PI, usa las siguientes cuatro reglas difusas

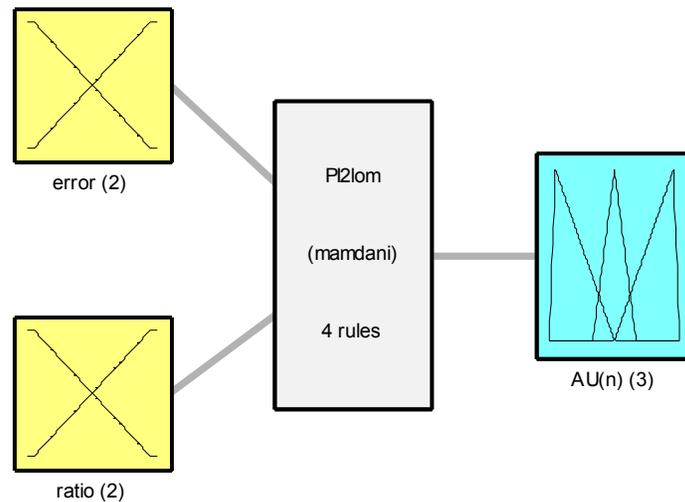
IF E(n) is Positivo AND R(n) is Positivo THEN $\Delta U(n)$ is Positivo

IF E(n) is Positivo AND R(n) is Negativo THEN $\Delta U(n)$ is Cero

IF E(n) is Negativo AND R(n) is Positivo THEN $\Delta U(n)$ is Cero

IF E(n) is Negativo AND R(n) is Negativo THEN $\Delta U(n)$ is Negativo

Las características del control difuso las vemos en las figuras 4.18 a 4.22.



System PI2lom: 2 inputs, 1 outputs, 4 rules

Fig. 4.18 Características del controlador difuso PI2lom.

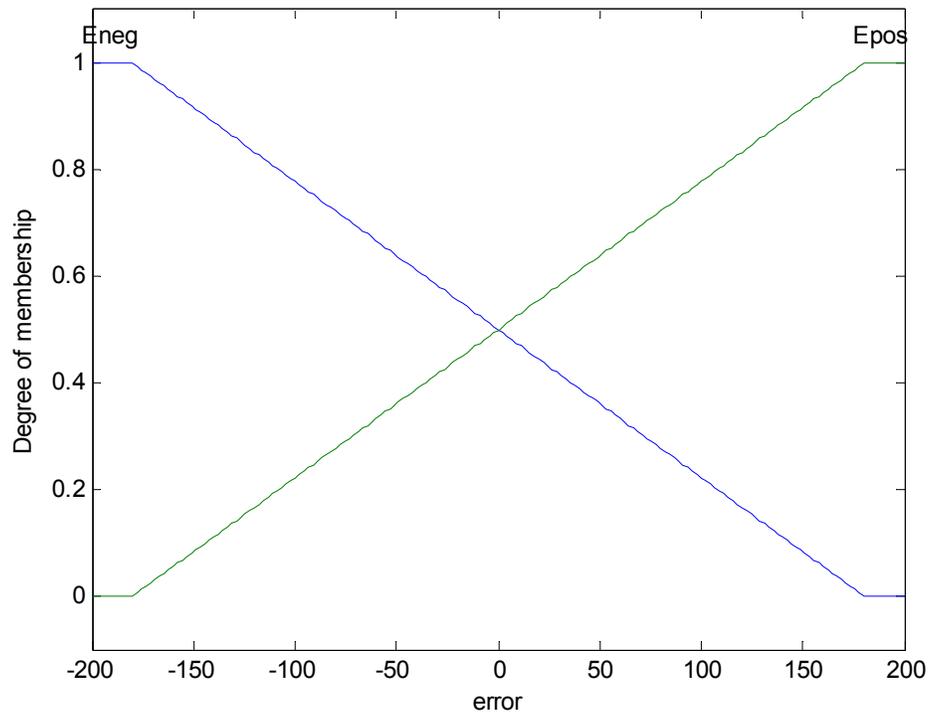


Fig. 4.19 Funciones de membresía para $e(n)$.

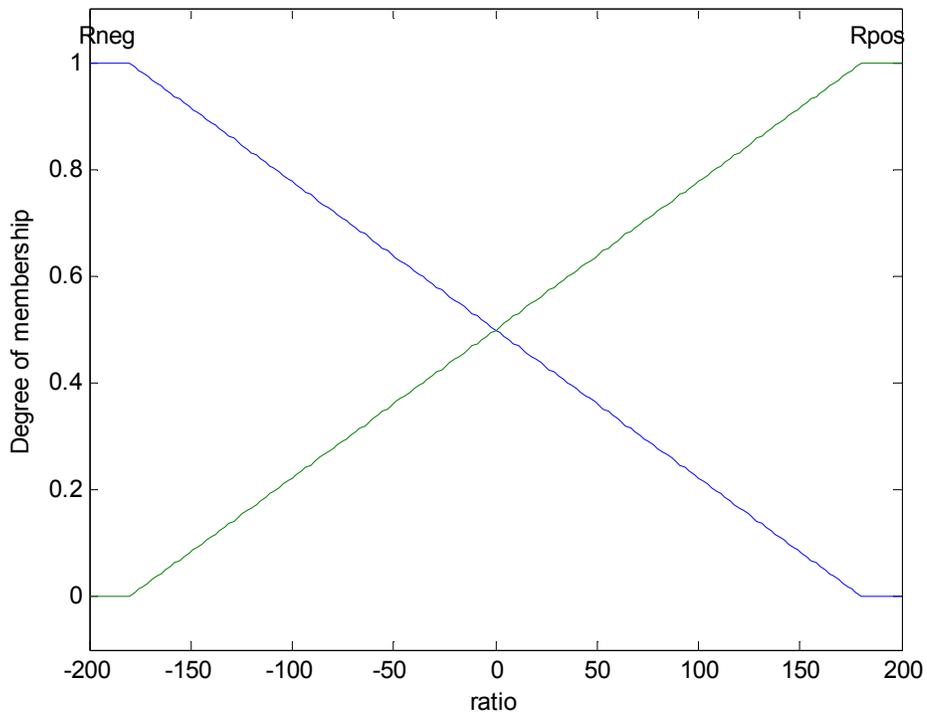


Fig. 4.20 Funciones de membresía para $r(n)$.

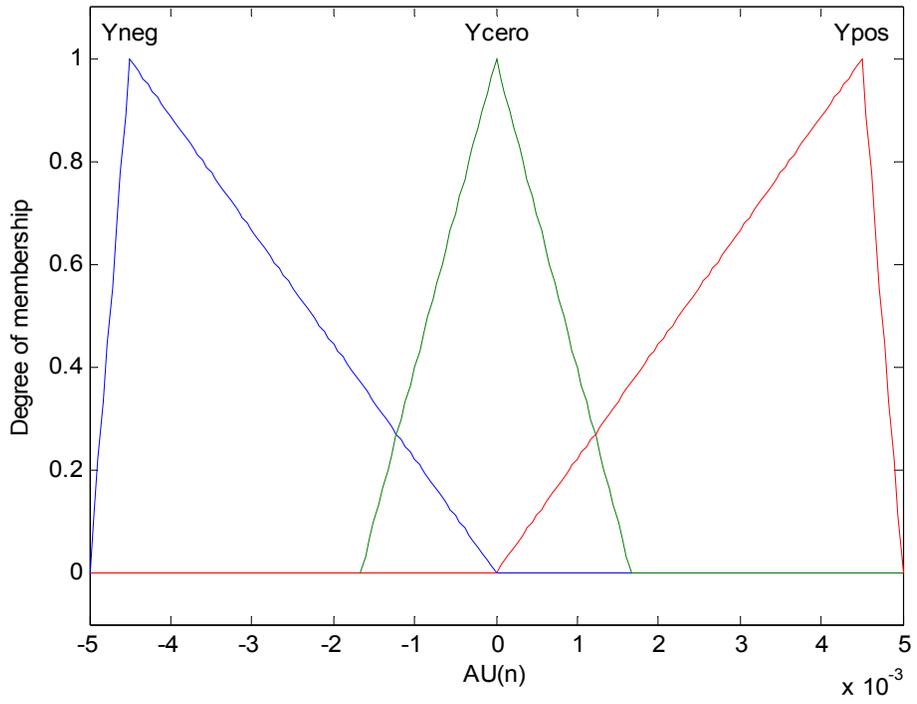


Fig. 4.21 Funciones de membresía para $\Delta U(n)$.

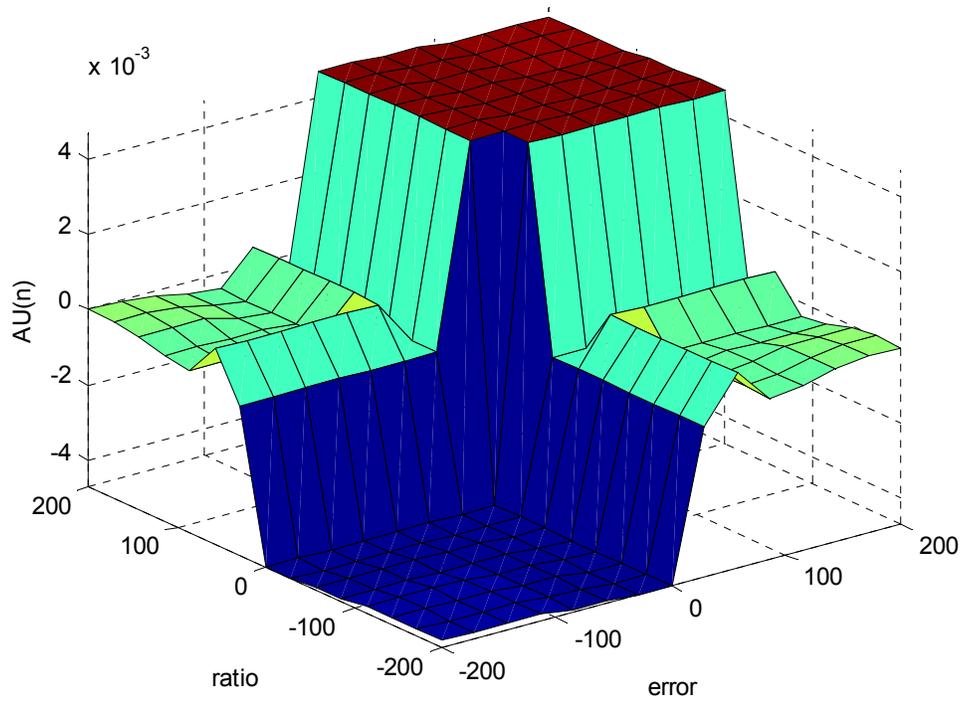


Fig. 4.22 Superficie de control generada.

Después de sintonizarlo, como podemos ver la respuesta del sistema es muy buena. Esta sintonización se logra mediante el uso del método de desdifusión lom, el periodo de muestreo $T_s = 0.00001 \text{ s} = 10 \mu\text{s}$, el rango del universo de discurso o incremento de la salida $\Delta U(n)$ que es de $[-0.005 \ 0.005]$, de acuerdo a la referencia [Fuzzy Control and Modeling, Hao Ying] la variación en la forma de los conjuntos triangulares y en menor grado la variación en la pendiente de las entradas, la cual comienza mas cerca del limite del universo de discurso, esto redujo el tiempo de respuesta.

	Tiempo de respuesta y tiempo de disturbio control PI de referencia	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	1.31 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	0.56 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.19 seg.

	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	32.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	22.5 A
Cambio en la carga (5 a 25 Nm)	27 A	29.5 A

En la figura 4.23 vemos la respuesta del motor de cd con el controlador PI difuso.

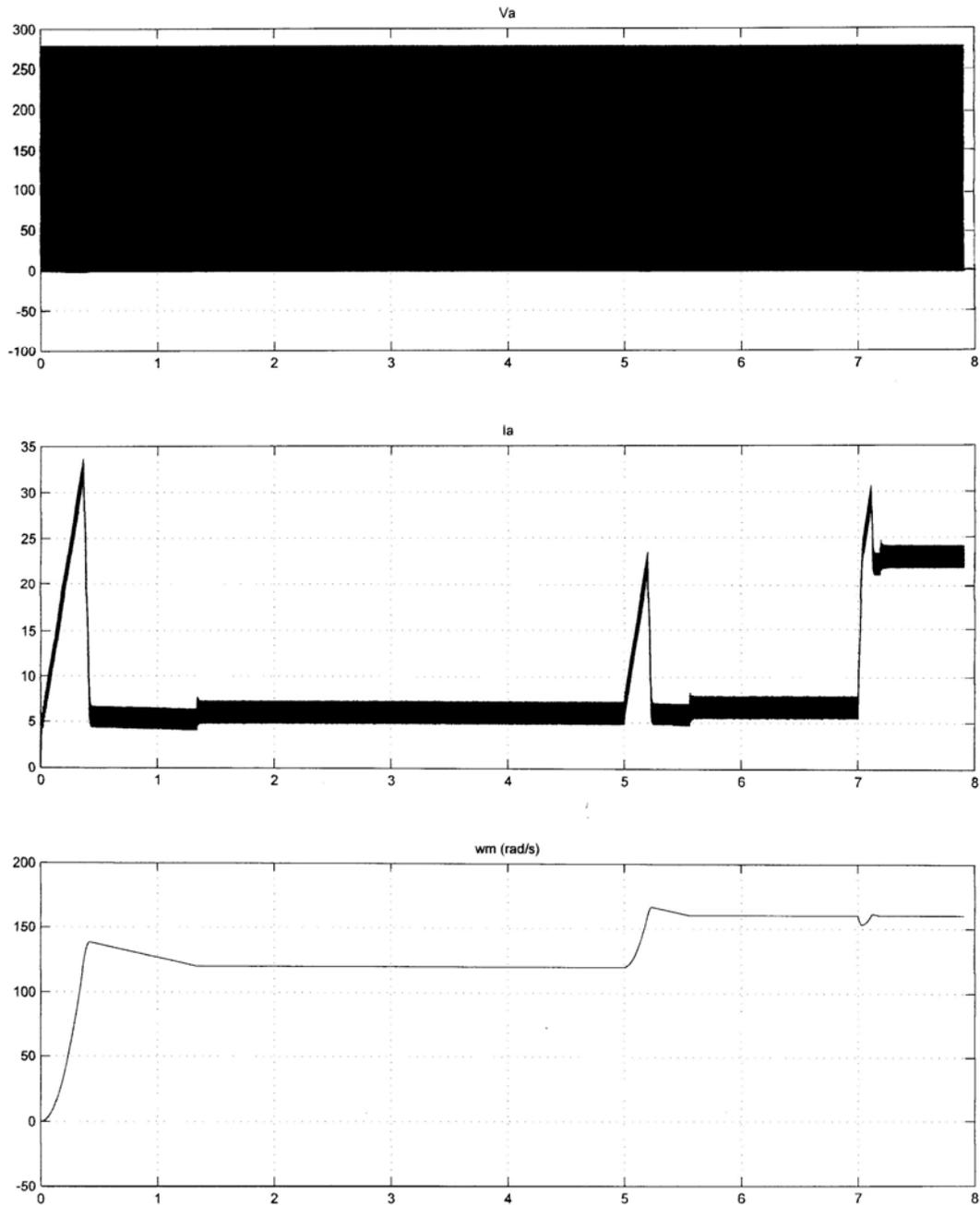


Fig. 4.23 Respuesta del controlador difuso PI, PI2lom.

Otra consideración en nuestro sistema de control es la exactitud en la referencia de velocidad deseada, puesto que presenta un rango de oscilación, que es natural en un sistema dinámico, pero que tomamos en cuenta, de acuerdo al requerimiento de la aplicación específica.

Tenemos en la figura 4.24 el rango de variación de nuestra referencia, el control PI y en la figura 4.25 la variación de este sistema de control PI difuso, como podemos observar la respuesta en ambos es otro de los parámetros que se consideraron para establecer las características del controlador, ya que el sistema difuso podemos variar el periodo de muestreo, reduciéndolo, para lograr una respuesta mas rápida, pero esto amplía el rango de variación en la

exactitud, por lo tanto, aun cuando responde el control difuso mas lento, logra la misma exactitud.

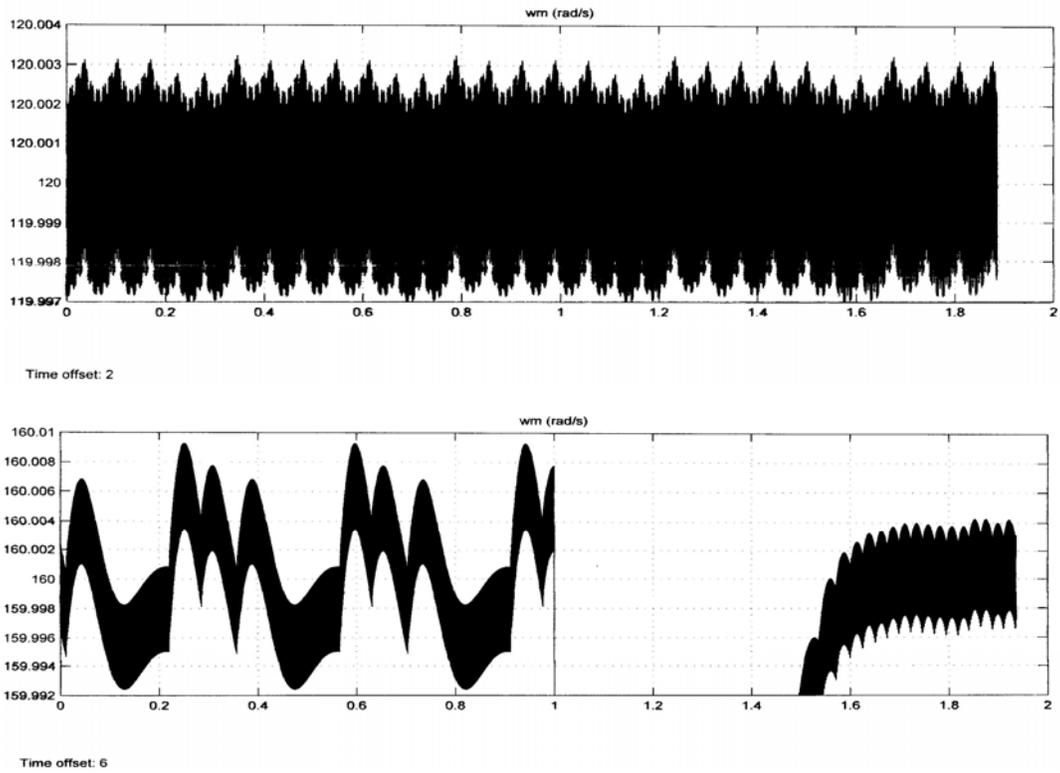


Fig. 4.24 Rango de variación a 120 rad/s y 160 rad/s del control clásico PI.

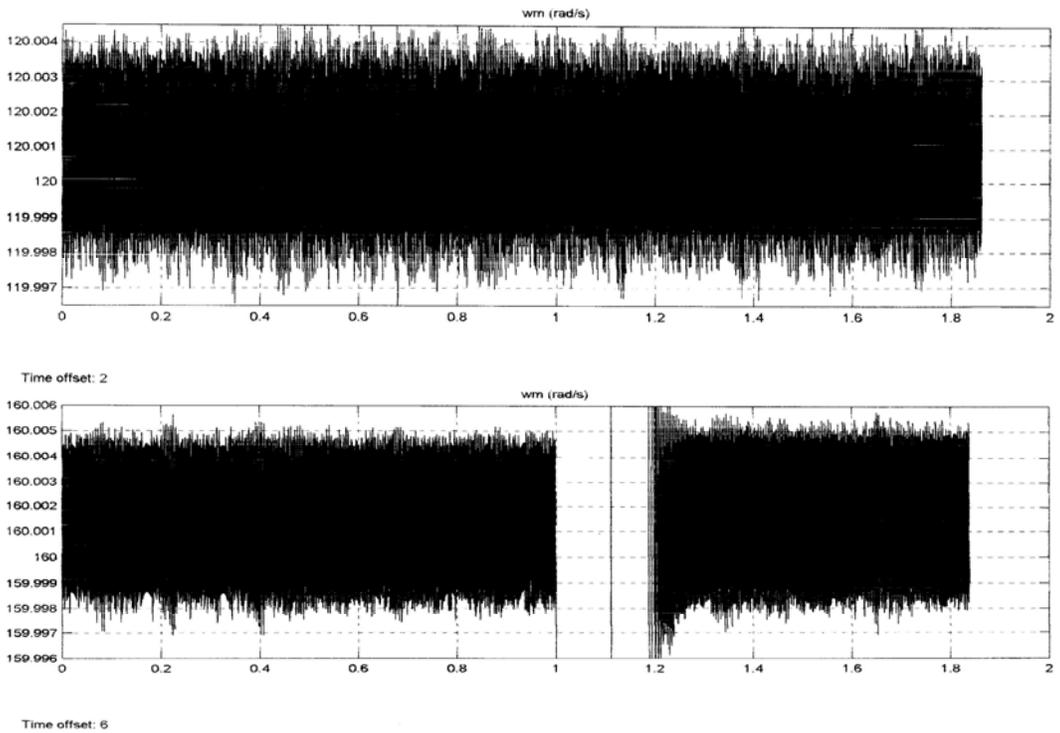


Fig. 4.25 Rango de variación a 120 rad/s y 160 rad/s del control PI difuso, PI2lom.

4.4.7 Tercer controlador (4 reglas difusas), conjuntos difusos de salida unitarios (singleton).

Este controlador difuso usa 2 conjuntos difusos idénticos, llamados positivo y negativo, al igual que el anterior, para las variables de entrada $E(n)$ y $R(n)$.

Para la salida usamos tres conjuntos difusos unitarios, negativo, cero y positivo.

El valor de los factores de escalamiento es de uno.

Este controlador difuso PI, usa las siguientes cuatro reglas difusas

IF $E(n)$ is Positivo AND $R(n)$ is Positivo THEN $\Delta U(n)$ is Positivo

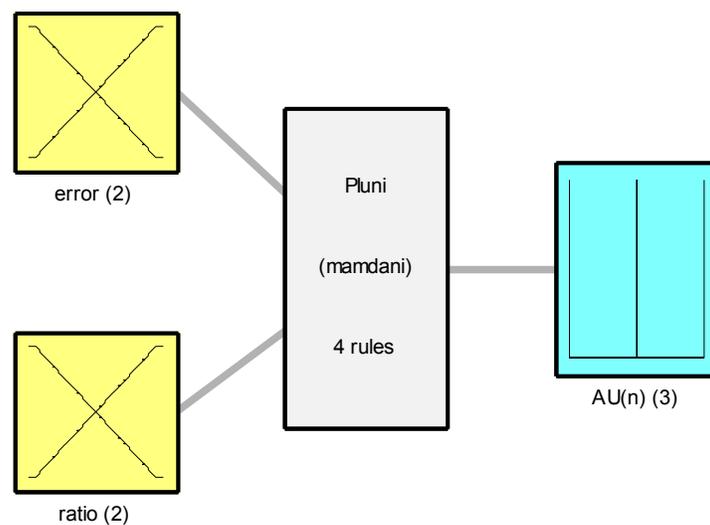
IF $E(n)$ is Positivo AND $R(n)$ is Negativo THEN $\Delta U(n)$ is Cero

IF $E(n)$ is Negativo AND $R(n)$ is Positivo THEN $\Delta U(n)$ is Cero

IF $E(n)$ is Negativo AND $R(n)$ is Negativo THEN $\Delta U(n)$ is Negativo

En la sintonización se vario el rango del universo de discurso de $\Delta U(n)$, quedando en $[-0.01 \ 0.01]$, la pendiente en los conjuntos de las entradas no varia significativamente la respuesta, con el uso de conjuntos unitarios en la salida la respuesta del sistema es prácticamente la misma usando para la desdifusión los métodos mom, lom y som. Este controlador es mucho más fácil de sintonizar, ya que después de las consideraciones anteriores el único parámetro a variar es el periodo de muestreo.

Las características del control difuso con conjuntos de salida unitarios (singleton) son las siguientes, figuras 4.26 a 4.30.



System Pluni: 2 inputs, 1 outputs, 4 rules

Fig. 4.26 Características del controlador difuso Pluni.

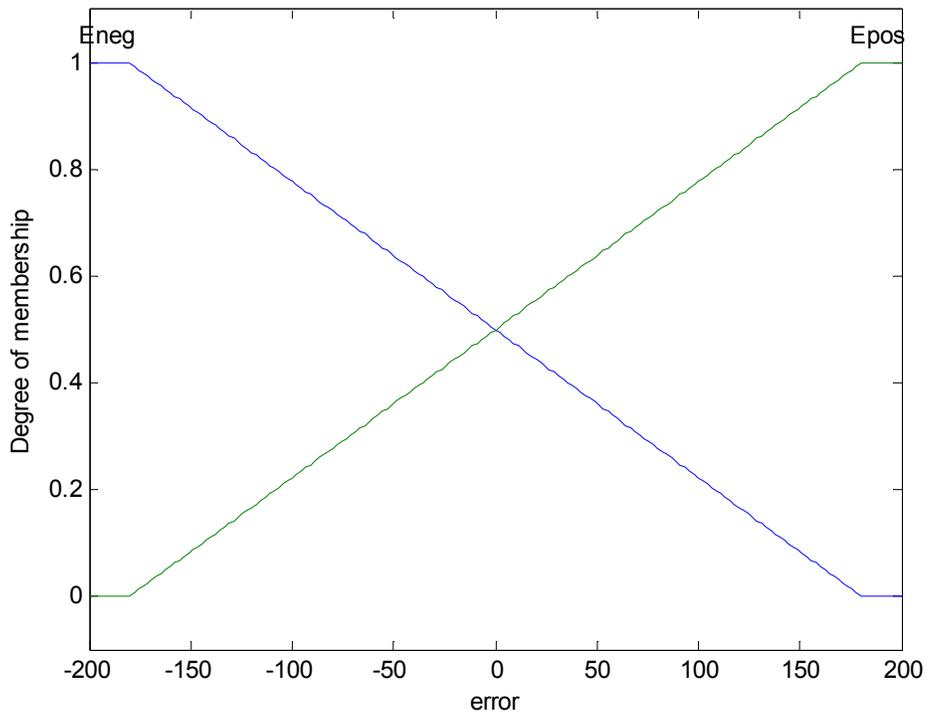


Fig. 4.27 Funciones de membresía para $e(n)$.

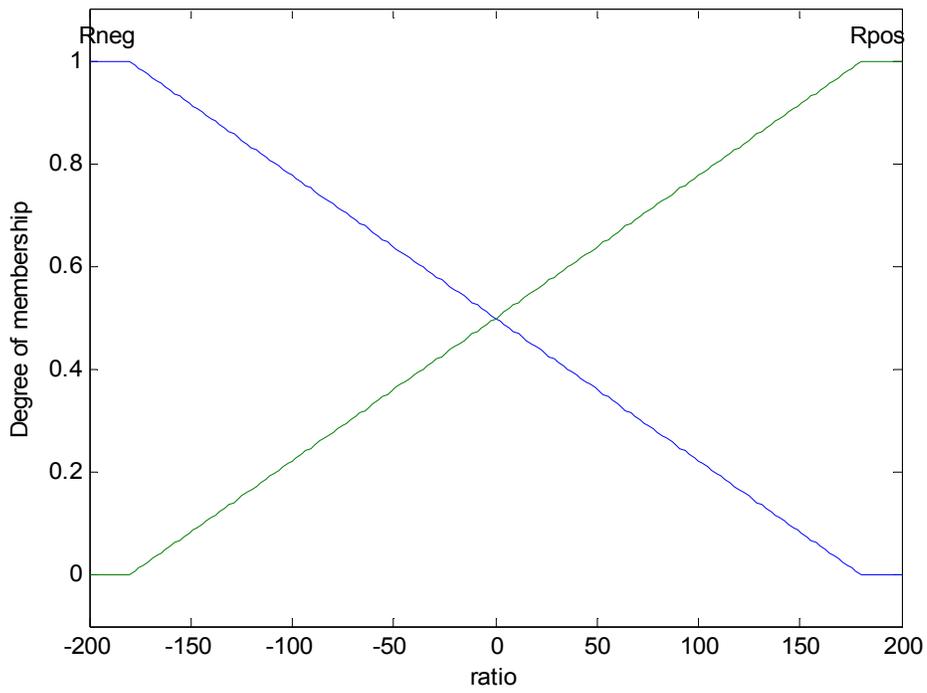


Fig. 4.28 Funciones de membresía para $r(n)$.

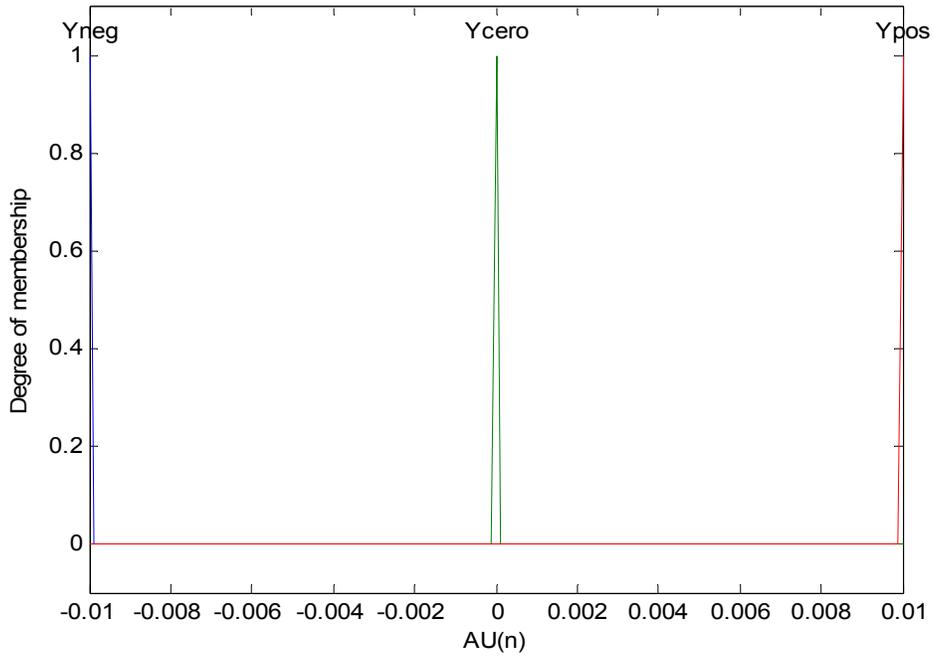


Fig. 4.29 Funciones de membresía para $\Delta U(n)$.

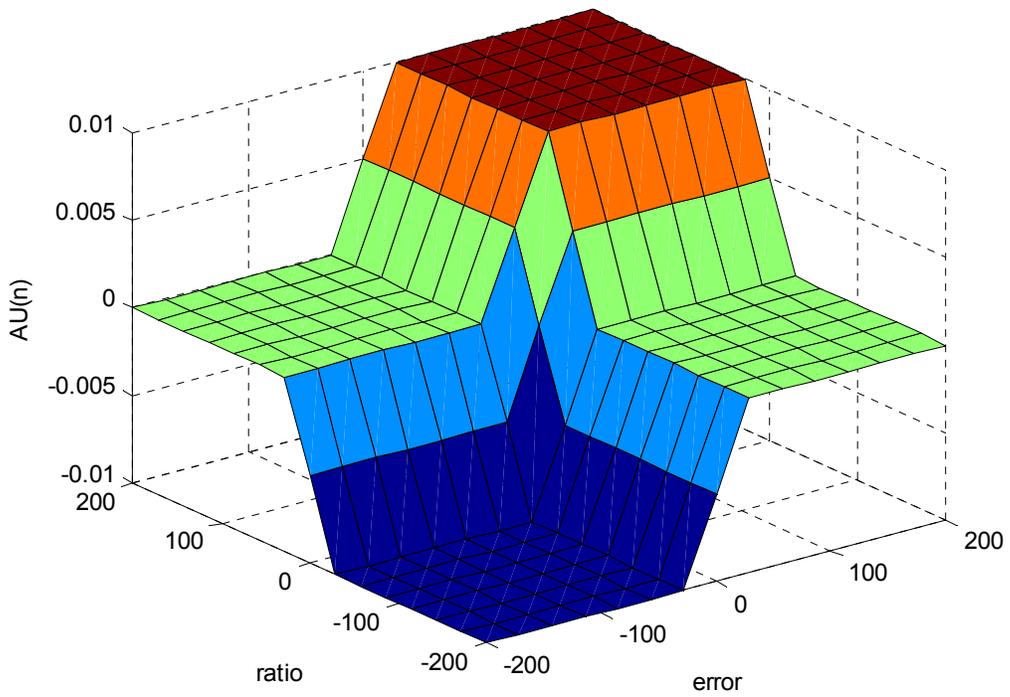


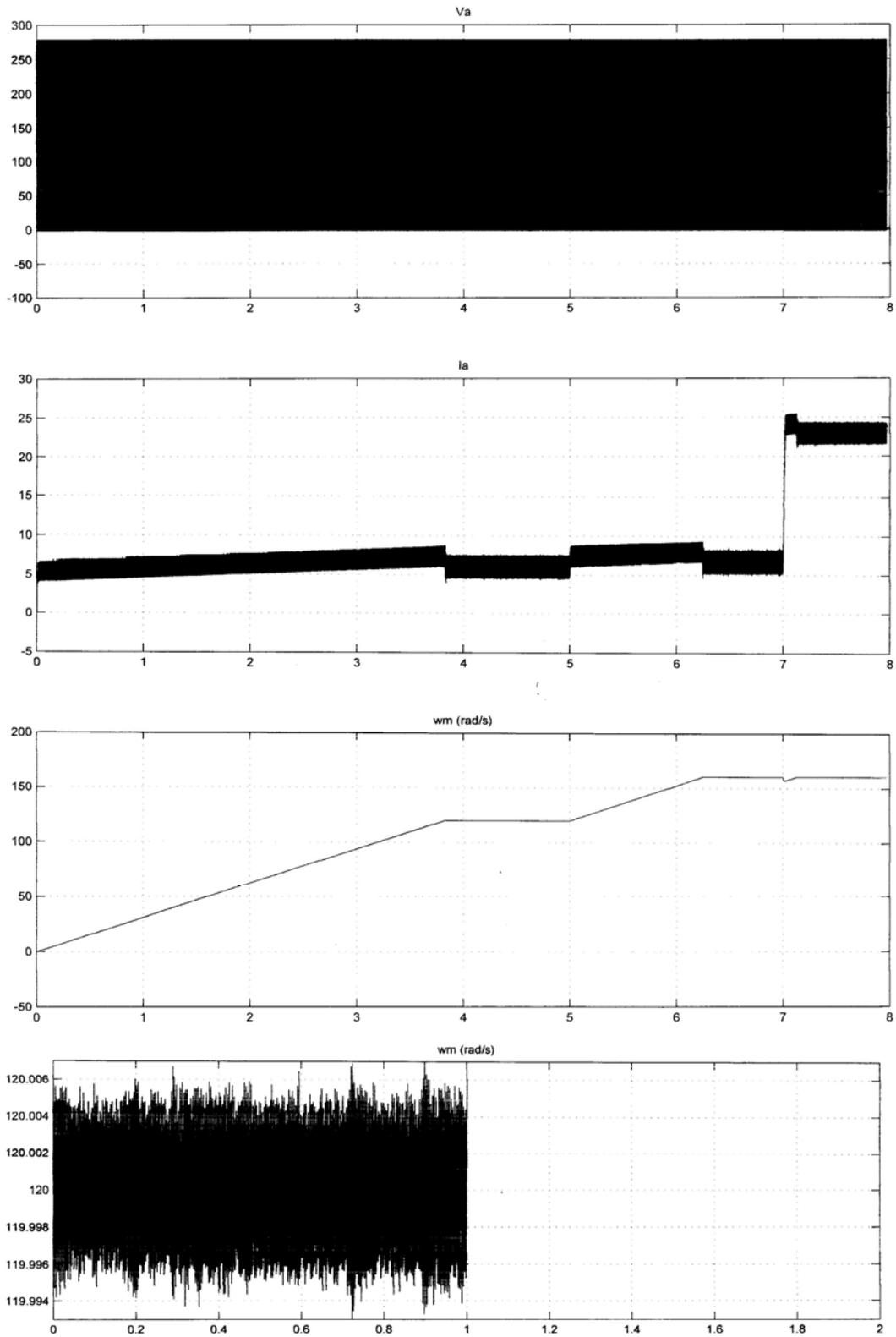
Fig. 4.30 Superficie de control generada.

Usando este controlador difuso, tenemos en la figura 4.31 la respuesta del sistema de control en el motor de cd, el periodo de muestreo $T_s = 0.00002 \text{ s} = 20 \mu\text{s}$, nos da una variación en la exactitud de la velocidad de referencia en estado estable, 120 rad/s, como la del control PI clásico, aunque el tiempo de respuesta para alcanzar esta velocidad es mucho mas grande.

	Tiempo de respuesta y tiempo de disturbio control PI de referencia.	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	3.78 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	1.25 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.12 seg.
	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	7.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	8.0 A
Cambio en la carga (5 a 25 Nm)	27 A	24.5 A

En la figura 4.32 vemos la respuesta del motor de cd usando el mismo control difuso, únicamente variando en nuestro sistema el periodo de muestreo, en el rango mas bajo que podemos usar sin que se pierda la estabilidad en el control de la velocidad y la corriente que no debe sobrepasar los 30 A. Este periodo es de $T_s = 0.00015 \text{ s} = 0.15 \text{ ms}$. La comparación de este sistema es la siguiente.

	Tiempo de respuesta y tiempo de disturbio control PI de referencia	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	1.62 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	0.53 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.07 seg.
	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	9.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	10.0 A
Cambio en la carga (5 a 25 Nm)	27 A	27.5 A



Time offset: 4

Fig. 4.31 Respuesta del controlador difuso Pluni con $T_s = 20 \mu s$.

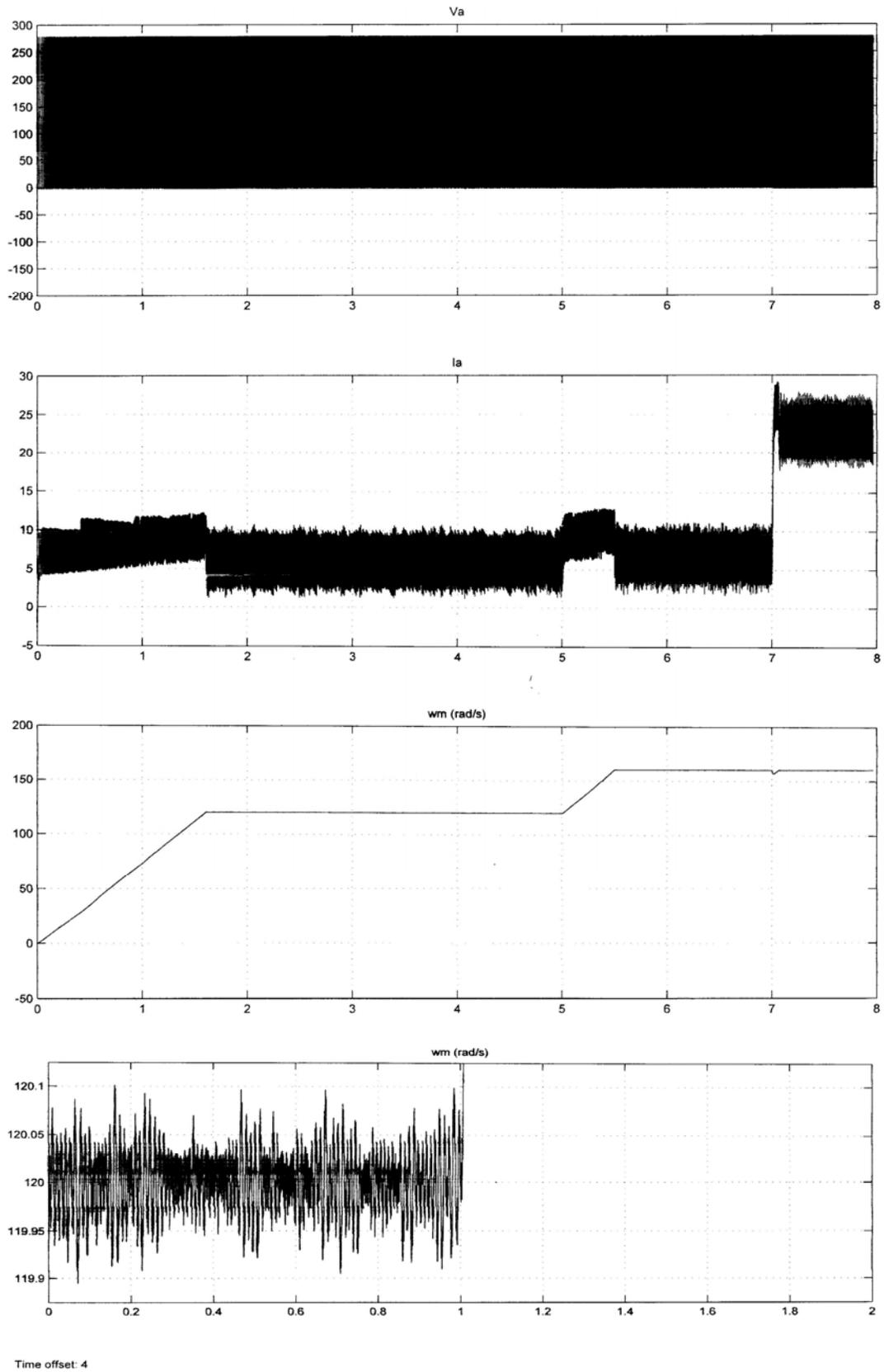
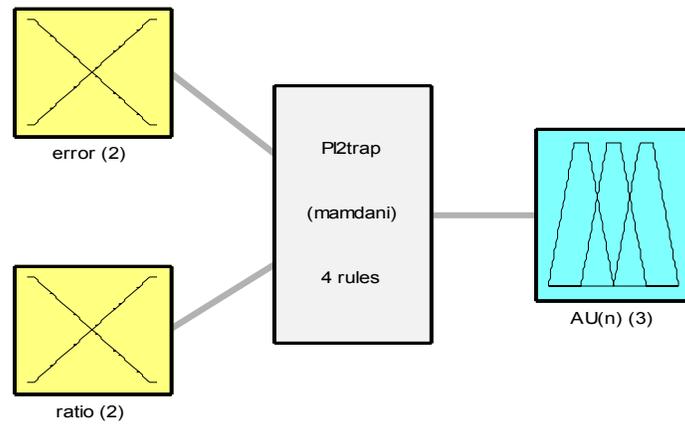


Fig. 4.32 Respuesta del controlador difuso Pluni con $T_s = 0.15$ ms.

4.4.8 Cuarto controlador (4 reglas difusas), conjuntos difusos de salida trapezoidales.

Las variables de entrada $E(n)$ y $R(n)$ son iguales que los anteriores controles difusos.

En la salida tenemos tres conjuntos difusos trapezoidales, negativo, cero y positivo. El valor de los factores de escalamiento es de uno. Las reglas difusas son las mismas. El bloque de control difuso con conjuntos trapezoidales en la salida se muestra en las figuras 4.33 a 4.37.



System PI2trap: 2 inputs, 1 outputs, 4 rules

Fig. 4.33 Características del controlador difuso PI2trap.

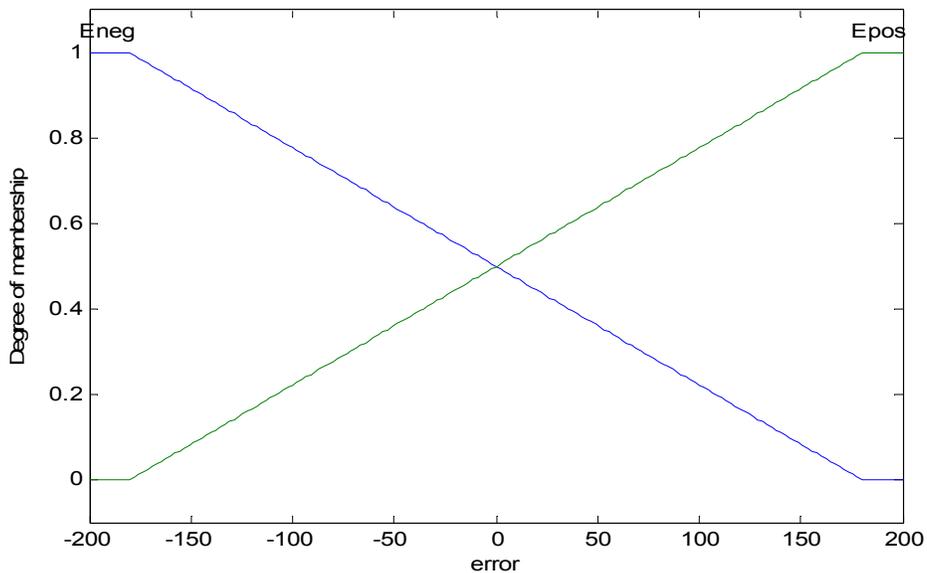


Fig. 4.34 Funciones de membresía para $e(n)$.

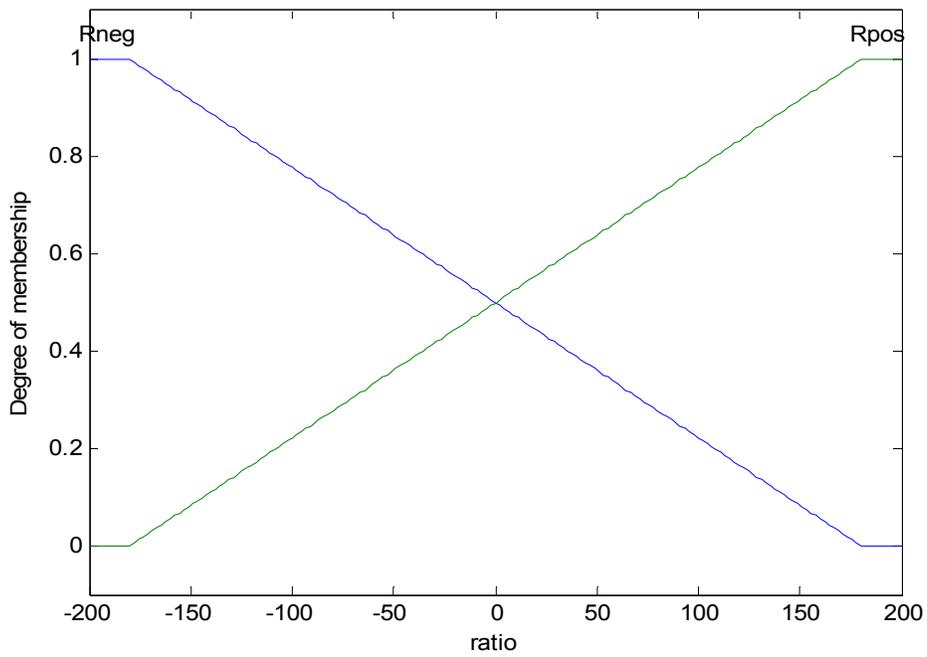


Fig. 4.35 Funciones de membresía para $r(n)$.

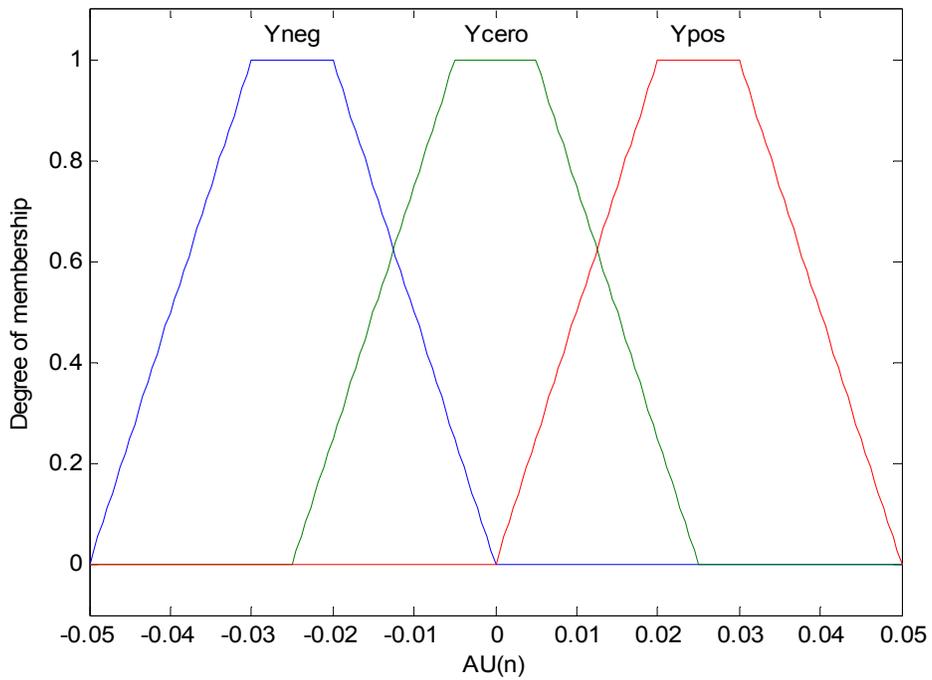


Fig. 4.36 Funciones de membresía para $\Delta U(n)$.

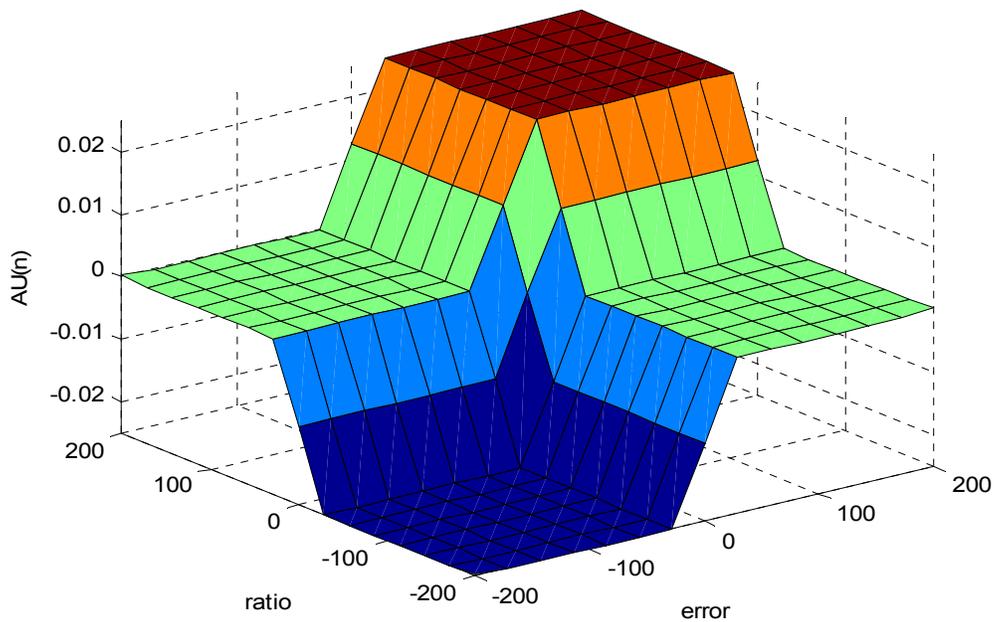


Fig. 4.37 Superficie de control generada.

Aun cuando en la literatura se plantea una mejor en cuanto a la respuesta del sistema, esta es muy parecida al controlador difuso con conjuntos unitarios (singleton), la variación en los conjuntos difusos trapezoidales o en los conjuntos difusos de entrada no implica una disminución del tiempo de respuesta. Este bloque de control difuso utiliza el método de desdifusión mom, la respuesta con el método som es cercana y con el método lom tarda casi el doble de tiempo en llegar a la velocidad de referencia. Con un periodo de muestreo de $T_s = 0.00002 \text{ s} = 20 \mu\text{s}$, en la figura 4.38 tenemos la siguiente respuesta.

	Tiempo de respuesta y tiempo de disturbio control PI de referencia.	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	3.73 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	1.20 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.05 seg.

	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	7.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	8.0 A
Cambio en la carga 5 a 25 Nm	27 A	24.5 A

La calidad de este controlador difuso es que el tiempo del disturbio cuando se incrementa la carga es muy pequeño (0.05 s), por lo que en estado estable, las variaciones en la carga no implican una variación sensible en la velocidad del motor de cd.

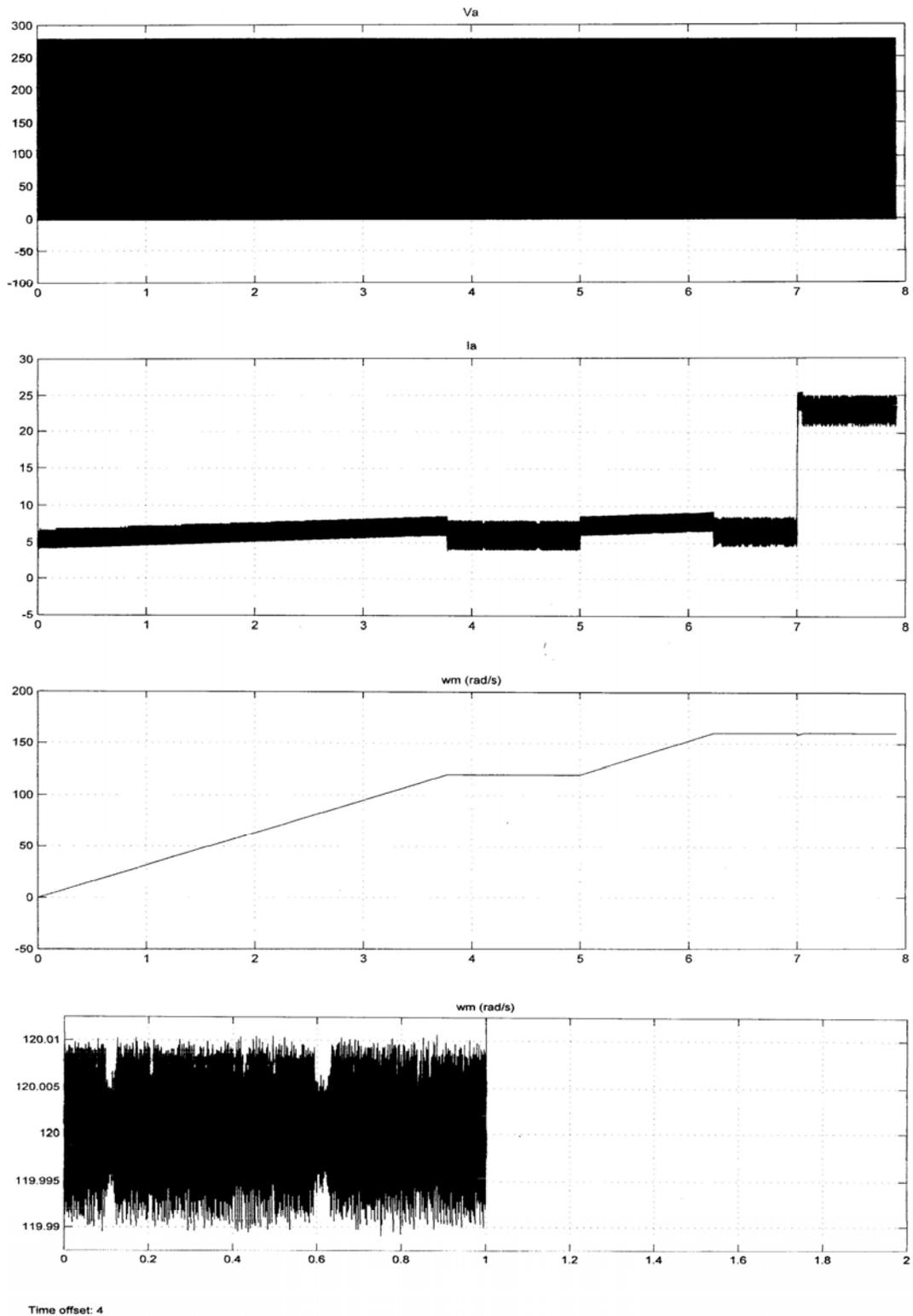


Fig. 4.38 Respuesta del controlador difuso PI2tap con $T_s = 20 \mu s$.

CONCLUSIONES

La respuesta de los controladores PI difusos después de haberlos sintonizado, paso de subamortiguada a críticamente amortiguada, este tipo de respuesta es mas sensible en cuanto a su respuesta, en los controladores con un mayor numero de reglas difusas, como es el caso del primer controlador, el cual tiende a una respuesta subamortiguada, es decir oscila y por lo tanto toma mayor tiempo alcanzar la velocidad de referencia.

Aun cuando los controladores PI difusos no son tan rápidos como los controladores PI clásicos, su respuesta es muy buena y no es necesario usar limitadores de corriente, aunque en una implementación física es recomendable, como es el caso del control PI clásico que tenemos como referencia.

El segundo controlador donde usamos conjuntos difusos de salida triangulares tiene una respuesta muy buena, es casi tan rápido como el PI clásico, pero tiene el mismo inconveniente, para llegar a la velocidad de referencia de 120 rad/s, el consumo de corriente tiende a llegar rápidamente al limite y aun sobrepasar la corriente de 30 A, en este caso llega a 32 A, por lo que requeriría de igual forma de un limitador de corriente, esto obviamente es normal, si requerimos que la acción del controlador sea rápida, la corriente que demande el sistema en este estado transitorio será mayor, para luego en estado estable usar la corriente suficiente para mantener el régimen permanente de velocidad con la carga requerida al motor. El uso del motor de cd con excitación independiente nos permite variar la carga sin que la variación de velocidad sea significativa, solo presenta un pequeño disturbio, en el cual la variación de velocidad es muy pequeña, esto nos permite un control robusto en cuanto a la velocidad al presentarse variaciones en la carga.

El tercer controlador con conjuntos unitarios (singleton) en la salida, tiene una respuesta críticamente amortiguada, en donde, si bien tarda mas que el controlador PI clásico en alcanzar la velocidad de referencia, esto por el mismo sentido del tipo de respuesta que deseamos y por la dinámica del sistema al usar periodos de muestreo muy pequeños, consideramos que su respuesta es muy buena. Si requerimos que la exactitud en cuanto a la velocidad de referencia sea mayor, usaremos un periodo de muestreo menor, esto nos permite censar la actuación del sistema un mayor numero de veces y ejercer una acción de control, pero, por el mismo proceso y la restricción en cuanto a la amplitud de variación, es mas tardado que alcance la velocidad de referencia, aunque la ventaja es que esto también limita la corriente en este estado transitorio. Si requerimos mayor rapidez para alcanzar la velocidad de referencia, aumentamos el tiempo del periodo de muestreo, esto tiene la desventaja de disminuir la exactitud y permitir un aumento de la corriente en este estado transitorio. Podemos variar los parámetros de acuerdo a los requerimientos, en cuanto a exactitud en la velocidad de referencia y la rapidez en la respuesta cuando se cambie la velocidad requerida al motor.

El tiempo del disturbio con la variación de la carga es prácticamente el mismo.

En el cuarto controlador con conjuntos trapezoidales en la salida, aun cuando en la literatura se sugiere la obtención de una mejor respuesta con este tipo de

conjuntos, en el sistema que estamos controlando, el motor de cd, obtenemos prácticamente la misma respuesta que en el tercer controlador con conjuntos unitarios (singleton), por lo que el comportamiento es muy parecido, pero con la desventaja en el método de desdifusión, en los controladores con conjuntos unitarios es mas flexible, pues podemos hacer uso de tres métodos y con conjuntos trapezoidales la mejor respuesta es solo con uno.

Evaluando los controladores difusos, vemos que los mejores son los controladores difusos con conjuntos unitarios (singleton) en la salida. De acuerdo a las características de respuesta que pretendemos, críticamente amortiguado, y en un momento dado, si se implementara el controlador difuso, la membresía de estos conjuntos unitarios es 1 en un específico valor del universo de discurso determinado, lo que facilitaría su programación, esto aunado a la forma incremental del controlador difuso PI, nos permite tener una respuesta de control bastante buena, la cual nos limita la corriente, permite la protección del motor y donde los disturbios son muy pequeños con las variaciones en la carga.

CONTROL DIFUSO PI CON CONJUNTOS DIFUSOS UNITARIOS.

Se implementa un controlador difuso PI, con un sistema de control en el que aplicamos factores de escalamiento para las variables de entrada y la variable de salida, utilizamos en la salida conjuntos unitarios. En este controlador solo tendremos que manipular dos parámetros para su sintonización, el periodo de muestreo y el factor de escalamiento para la variable de salida del control difuso PI en forma incremental. Usaremos el sistema de control de la figura C1.

En el bloque del controlador con lógica difusa, usamos para las variables de entrada $E(n)$ y $R(n)$ un rango de -127 a 127, antes de entrar al bloque de control las variables $e(n)$ y $r(n)$ se multiplican con los bloques F. Esc. (factor de escalamiento), para que el universo de discurso este en el rango de velocidad que deseo controlar, de -200 a 200. Para obtener el valor de los factores de escalamiento tenemos que:

$$E(n) = K_e e(n); \quad \text{por lo tanto} \quad K_e = E(n) / e(n) = 127 / 200 = 0.635$$

$$R(n) = K_r r(n); \quad \text{por lo tanto} \quad K_r = R(n) / r(n) = 127 / 200 = 0.635$$

para la variable de salida del controlador en forma incremental usamos un rango de [-1 1], con la variación del factor de escalamiento de esta variable, y con el periodo de muestreo podemos sintonizar el controlador, esto nos permite trabajar en la sintonización sin modificar el bloque de control difuso.

El controlador usa las siguientes cuatro reglas difusas

IF $E(n)$ is Positivo AND $R(n)$ is Positivo THEN $\Delta U(n)$ is Positivo

IF $E(n)$ is Positivo AND $R(n)$ is Negativo THEN $\Delta U(n)$ is Cero

IF $E(n)$ is Negativo AND $R(n)$ is Positivo THEN $\Delta U(n)$ is Cero

IF $E(n)$ is Negativo AND $R(n)$ is Negativo THEN $\Delta U(n)$ is Negativo

Cotrolador de un Motor de CD (Discreto)

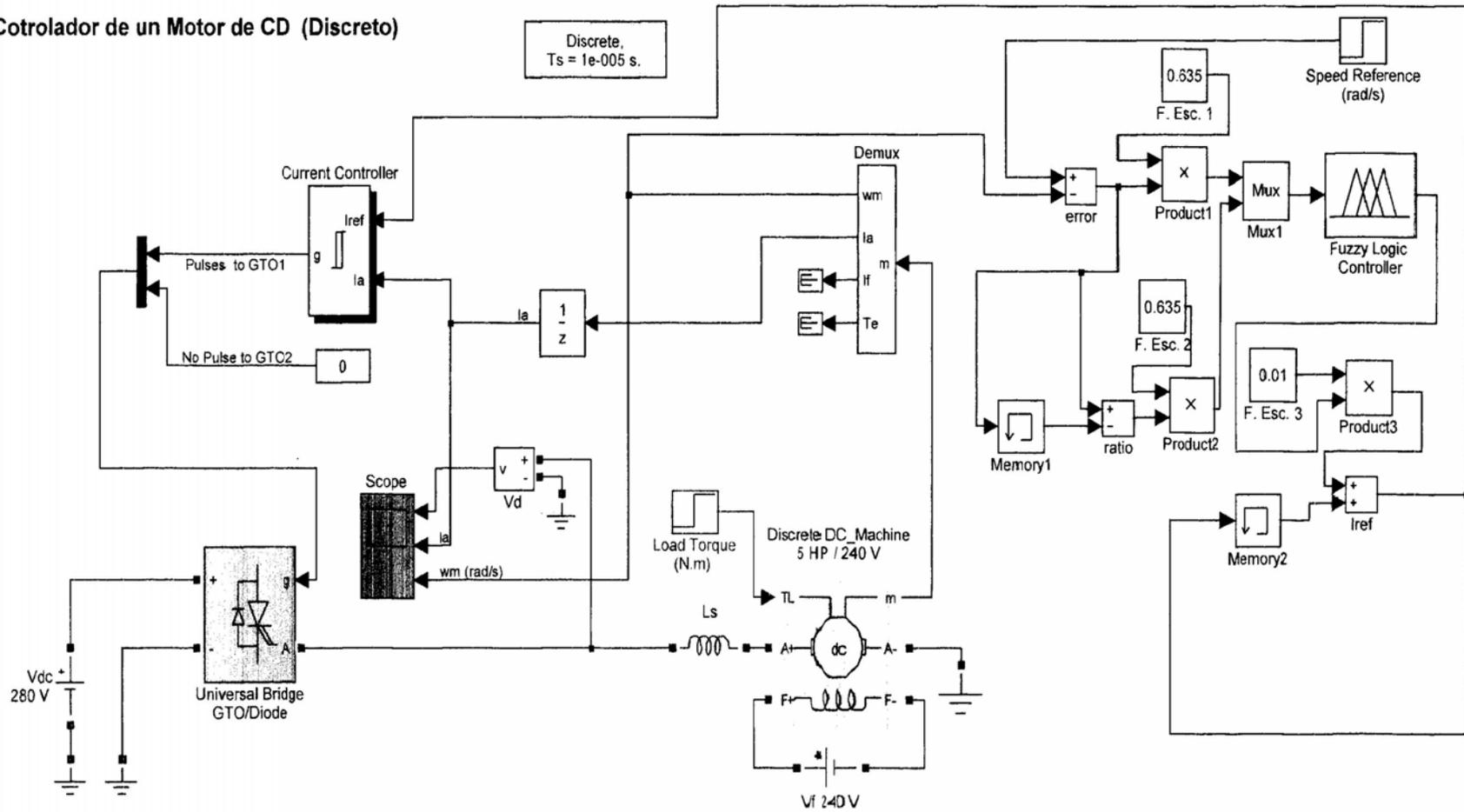
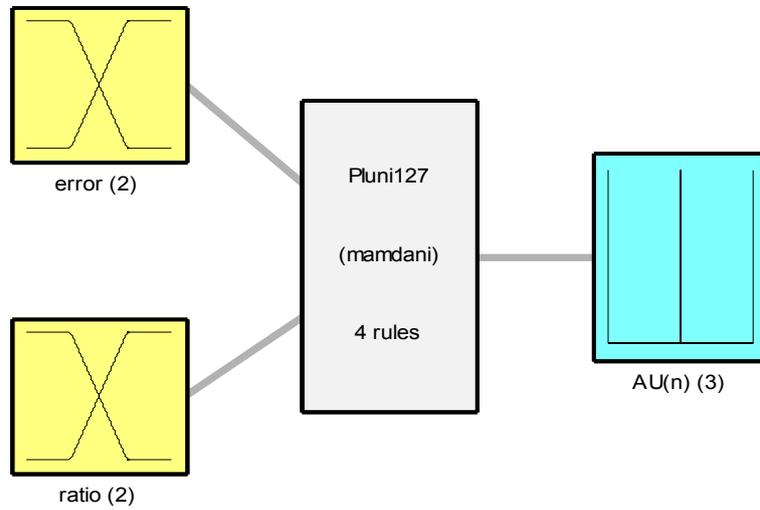


Fig. C1 Sistema de control difuso PI para un motor de CD.

Las características del bloque de control difuso las vemos en las figuras C2 a C6:



System Pluni127: 2 inputs, 1 outputs, 4 rules

Fig. C2 Características del controlador difuso Pluni127.

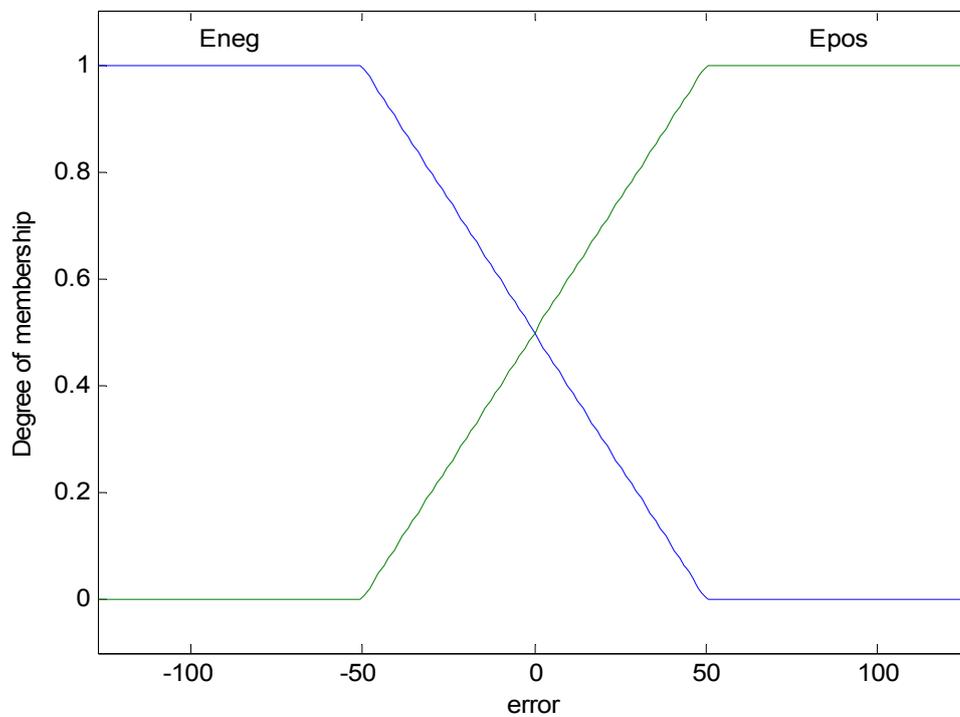


Fig. C3 Funciones de membresía para E(n).

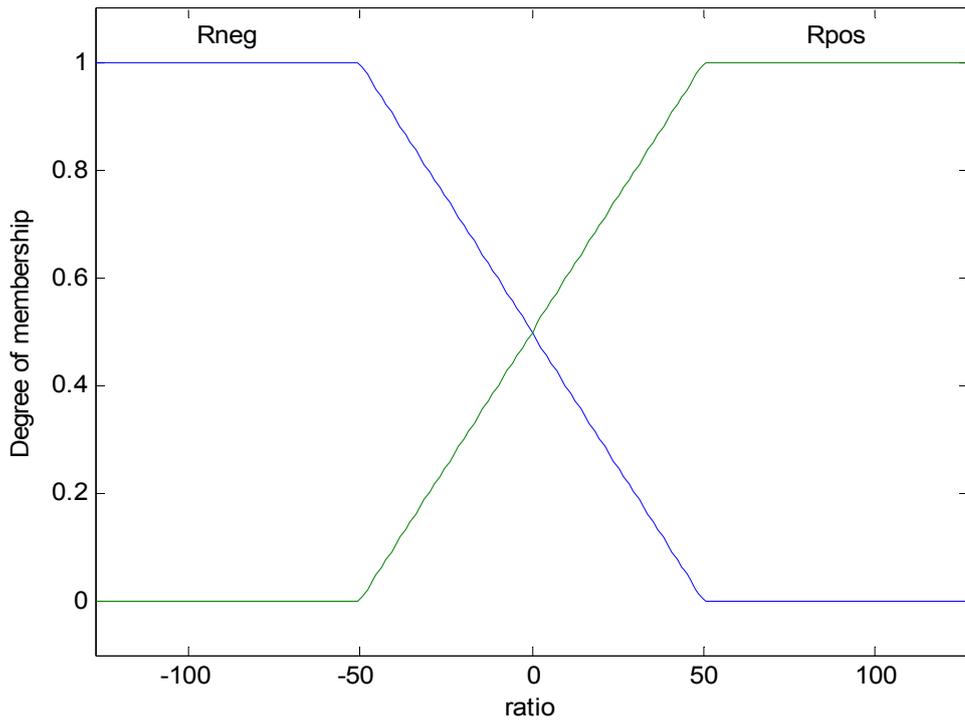


Fig. C4 Funciones de membresía para $R(n)$.

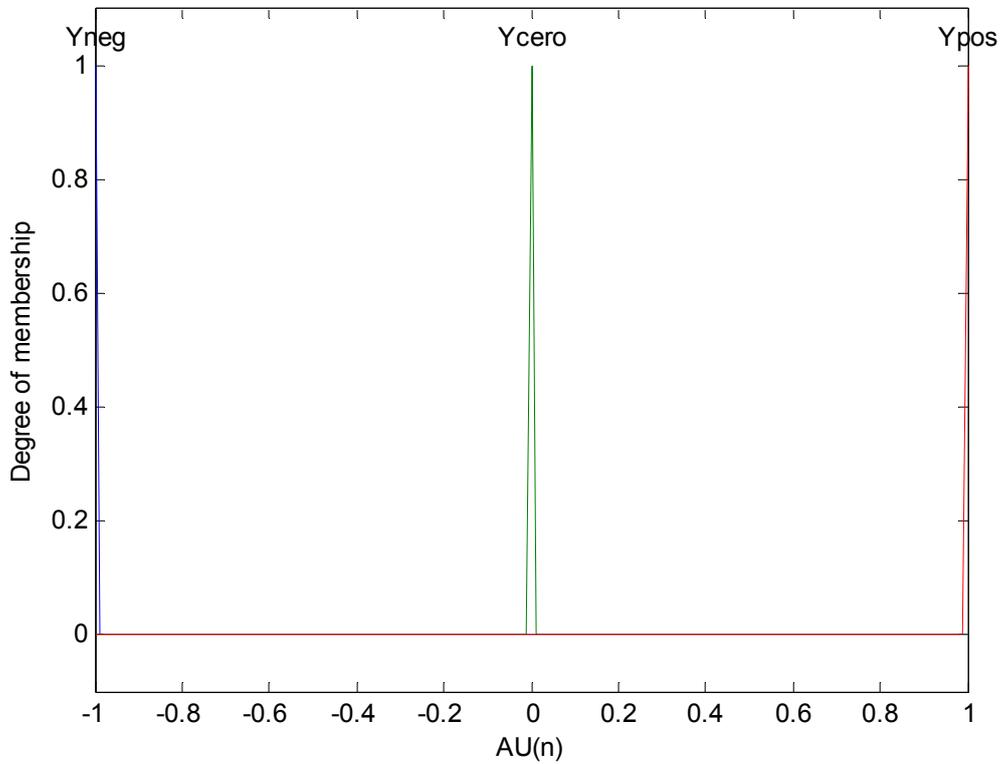


Fig. C5 Funciones de membresía para $\Delta U(n)$.

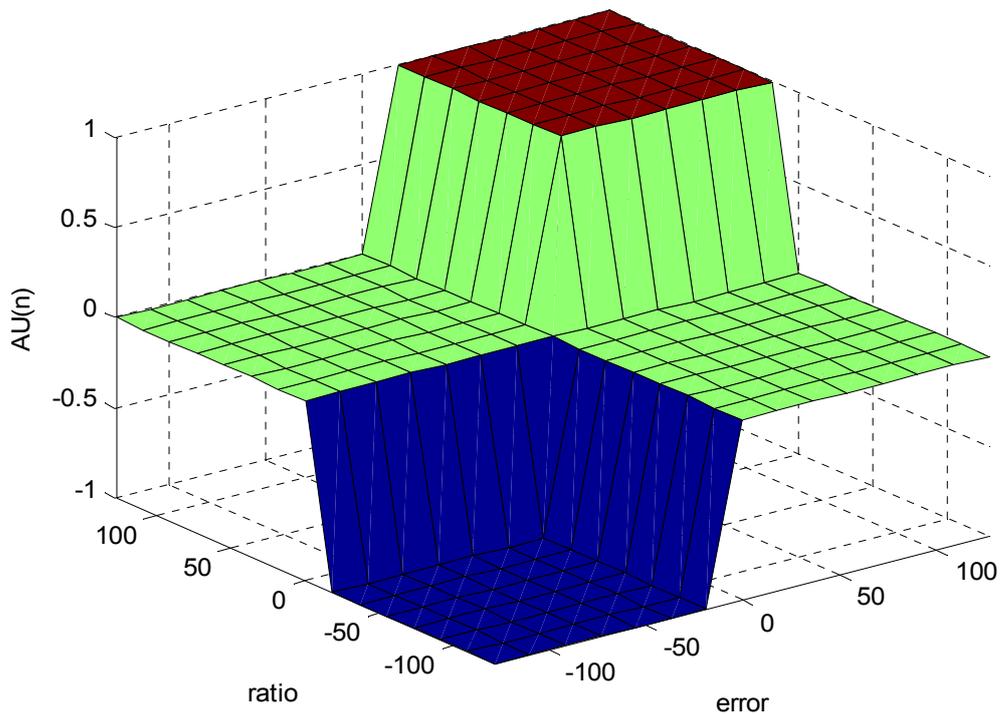


Fig. C6 Superficie de control generada.

Este bloque de control difuso usa el método de desdifusión som, la respuesta con los métodos mom y lom es prácticamente igual. Con un periodo de muestreo de $T_s = 0.00002 \text{ s} = 20 \mu\text{s}$, tenemos en la figura C7 la siguiente respuesta.

	Tiempo de respuesta y tiempo de disturbio control PI de referencia.	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	3.78 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	1.21 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.12 seg.
	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	7.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	8.0 A
Cambio en la carga (5 a 25 Nm)	27 A	24.5 A

El mismo controlador con un periodo de muestreo de $T_s = 0.00015 \text{ s} = 0.15 \text{ ms}$ nos permite disminuir el tiempo en alcanzar la velocidad de referencia, pero la exactitud en la referencia es menor como podemos ver en la figura C8, la respuesta es la siguiente:

	Tiempo de respuesta y tiempo de disturbio control PI de referencia.	Tiempo de respuesta y tiempo de disturbio control PI difuso.
Velocidad de Ref. 1 (120 rad/s)	0.58 seg.	1.63 seg.
Velocidad de Ref. 2 (160 rad/s)	0.49 seg.	0.52 seg.
Cambio en la carga (5 a 25 Nm)	0.25 seg.	0.08 seg.

	Corriente de armadura control PI de referencia. $I_a \text{ máx. (avg)}$	Corriente de armadura control PI difuso. $I_a \text{ máx. (avg)}$
Velocidad de Ref. 1 (120 rad/s)	30 A	9.5 A
Velocidad de Ref. 2 (160 rad/s)	30 A	10.0 A
Cambio en la carga (5 a 25 Nm)	27 A	27.5 A

La respuesta con este sistema de control difuso, usando factores de escalamiento diferentes de uno, es prácticamente la misma que la del tercer controlador, esto demuestra que este sistema nos permite trabajar un control difuso sin necesidad de hacer modificaciones en la herramienta de lógica difusa, únicamente modificando el factor de escalamiento de la salida, que es en forma incremental y el periodo de muestreo en simulink obtenemos la respuesta deseada.

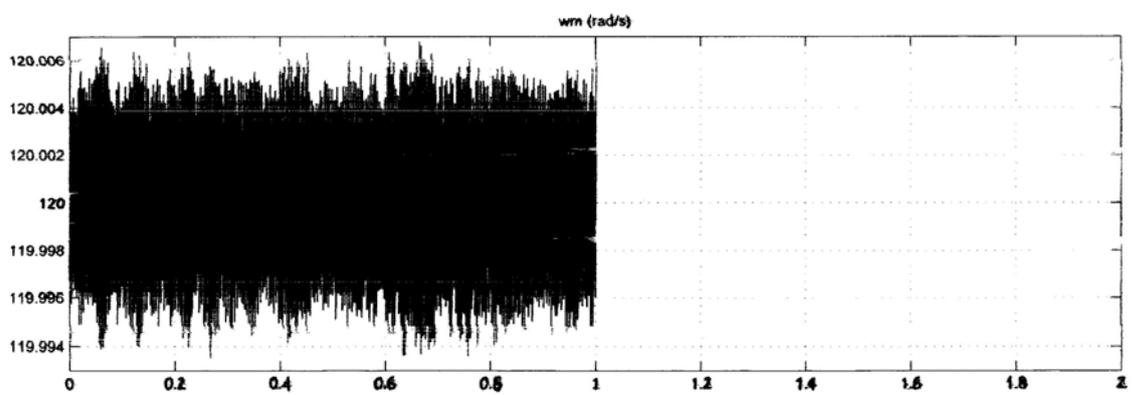
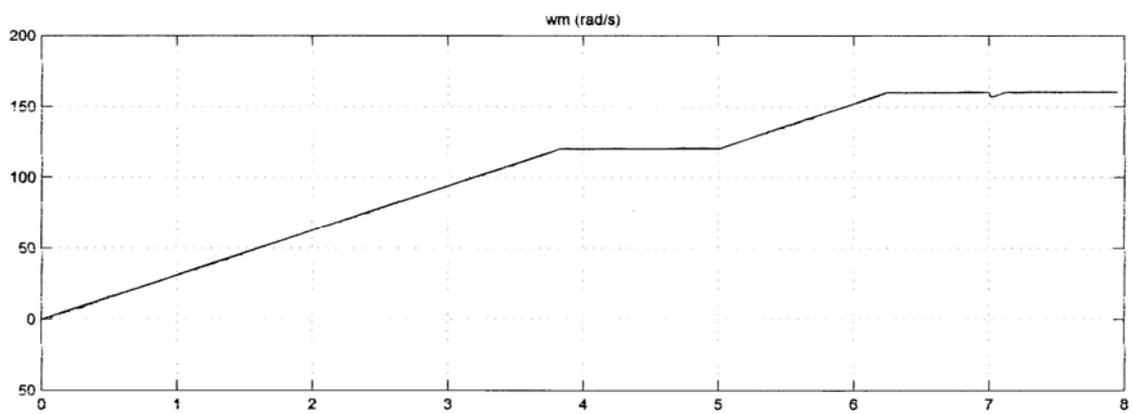
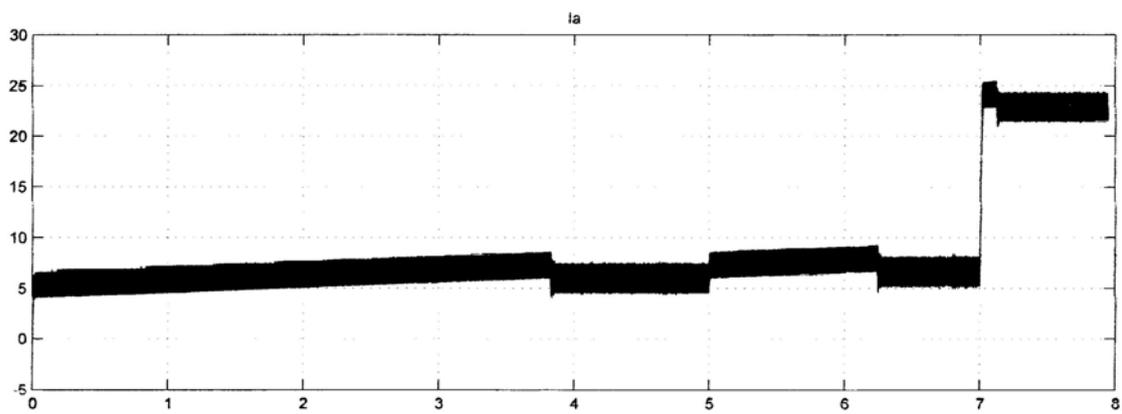
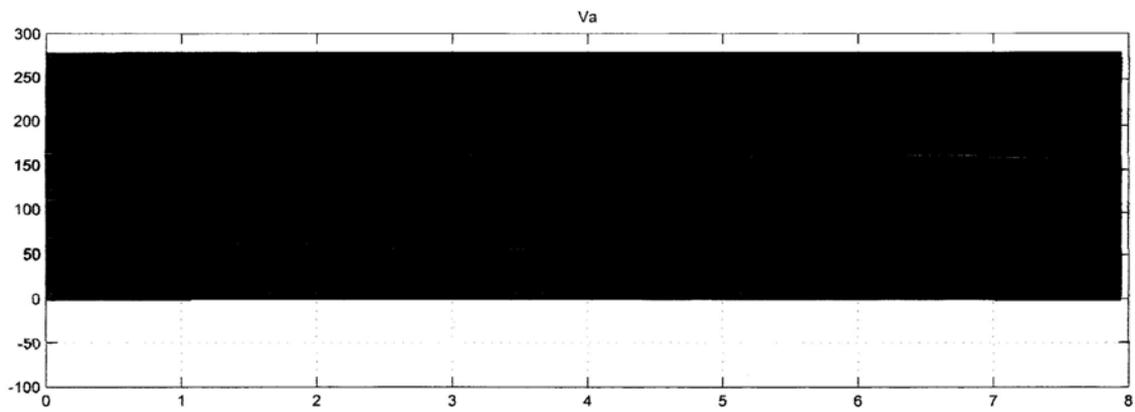


Fig. C7 Respuesta del controlador difuso Pluni127 con $T_s = 20 \mu s$.

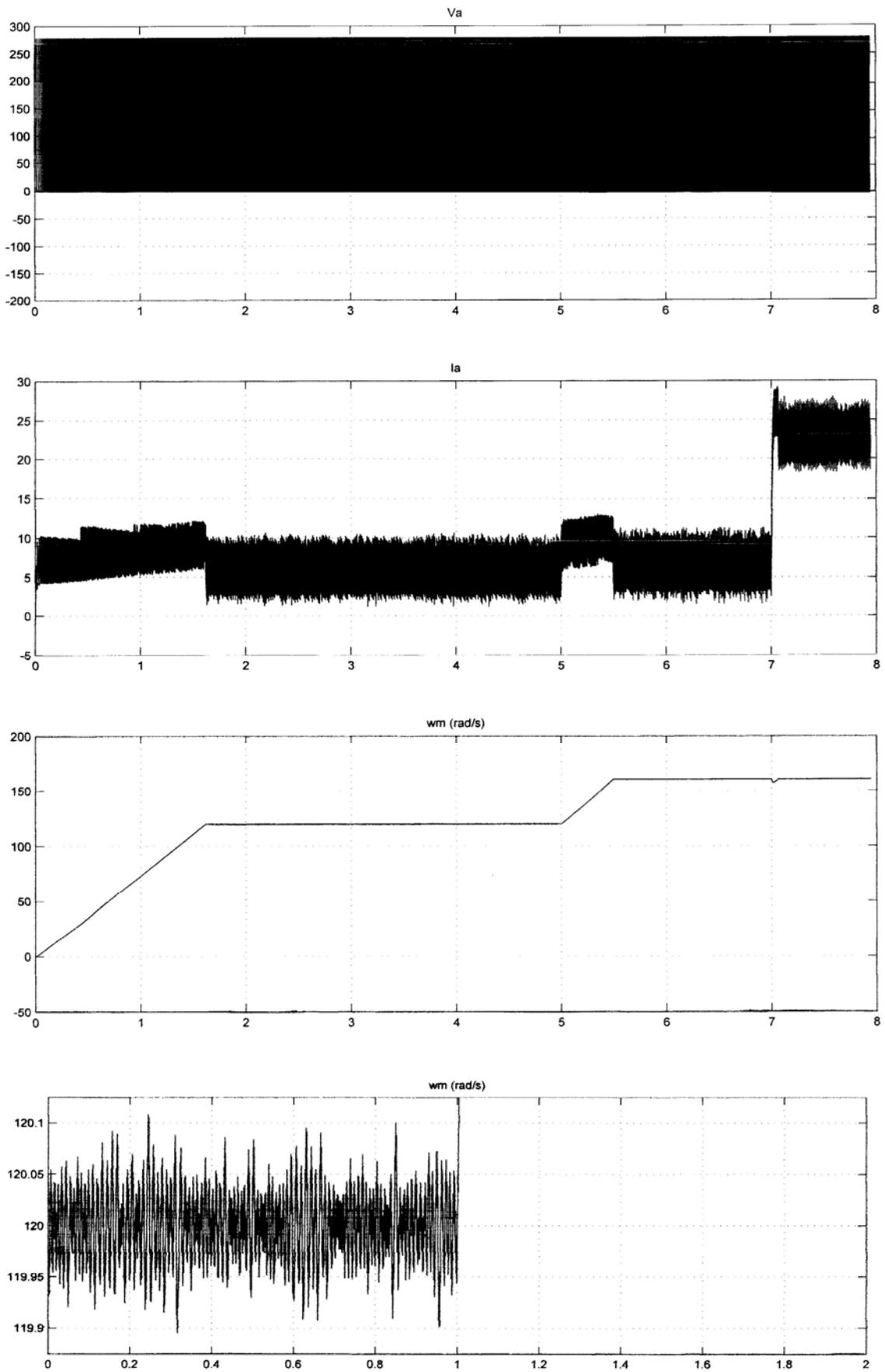


Fig. C8 Respuesta del controlador difuso Pluni127 con $T_s = 0.15$ ms.

APÉNDICE

PROGRAMACION DEL BLOQUE CONTROLADOR DE LÓGICA DIFUSA.

En el controlador difuso PI quitamos el bloque controlador de lógica difusa (Fuzzy Logic Controller) y lo sustituimos por el bloque de Inserción de funciones Matlab (Embedded Matlab Function) en el cual insertamos un programa que hace los mismos procesos que la herramienta de lógica difusa, estos son, difusión, implicación, agregación y desdifusión.

Las funciones de membresía para los conjuntos difusos son, para E(n):

$$\mu_P(e) = \begin{cases} 0, & E(n) < -L \\ (K_e e(n) + L) / (2L), & -L \leq E(n) \leq L \\ 1, & E(n) > L \end{cases}$$

$$\mu_N(e) = \begin{cases} 1, & E(n) < -L \\ (-K_e e(n) + L) / (2L), & -L \leq E(n) \leq L \\ 0, & E(n) > L \end{cases}$$

y las funciones de membresía para R(n) son

$$\mu_P(r) = \begin{cases} 0, & R(n) < -L \\ (K_r r(n) + L) / (2L), & -L \leq R(n) \leq L \\ 1, & R(n) > L \end{cases}$$

$$\mu_N(r) = \begin{cases} 1, & R(n) < -L \\ (-K_r r(n) + L) / (2L), & -L \leq R(n) \leq L \\ 0, & R(n) > L \end{cases}$$

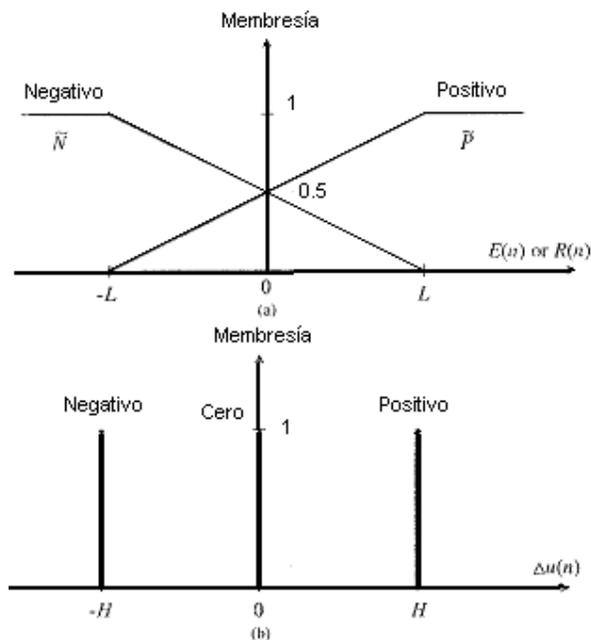


Figura A1 Definición gráfica de las entradas y la salida: (a) dos conjuntos difusos Positivo y Negativo para E(n) y R(n), (b) tres conjuntos de salida difusos unitarios (singleton) Positivo, Cero y Negativo.

Las reglas difusas son las mismas que usamos:

IF E(n) is Positivo AND R(n) is Positivo THEN $\Delta U(n)$ is Positivo
IF E(n) is Positivo AND R(n) is Negativo THEN $\Delta U(n)$ is Cero
IF E(n) is Negativo AND R(n) is Positivo THEN $\Delta U(n)$ is Cero
IF E(n) is Negativo AND R(n) is Negativo THEN $\Delta U(n)$ is Negativo

Ponemos los factores de escalamiento K_e y K_r igual a 1, por lo tanto $E(n) = e(n)$ y $R(n) = r(n)$,
El factor de escalamiento de la salida es también 1.

El programa es el siguiente:

```
function des = C9_prog(En, Rn)
```

```
% Las variables En y Rn representan E(n) y R(n), la variable des representa la  
% salida  $\Delta U(n)$ .
```

```
L=80;  
H=0.01;
```

```
%DIFUSION PARA En
```

```
if En < -L  
    mpe = 0;  
elseif En > L  
    mpe = 1;  
else mpe = (En + L)/(2*L);  
end  
if En < -L  
    mne = 1;  
elseif En > L  
    mne = 0;  
else mne = (-En + L)/(2*L);  
end
```

```
%DIFUSION PARA Rn
```

```
if Rn < -L  
    mpr = 0;  
elseif Rn > L  
    mpr = 1;  
else mpr = (Rn + L)/(2*L);  
end  
if Rn < -L  
    mnr = 1;  
elseif Rn > L  
    mnr = 0;  
else mnr = (-Rn + L)/(2*L);  
end
```

```

% Evaluación del de las 4 reglas con el operador lógico difuso Zadeh AND en
%el antecedente.
r1 = min(mpe,mpr);
r2 = min(mpe,mnr);
r3 = min(mne,mpr);
r4 = min(mne,mnr);

% IMPLICACION, usamos inferencia producto Larsen (Larsen product
% inference)  $\mu * \mu_w(z)$ .
lr1 = r1 * 1;
lr2 = r2 * 1;
lr3 = r3 * 1;
lr4 = r4 * 1;

% AGREGACION Y DESDIFUSION
% Para la agregación usamos el operador lógico difuso Zadeh OR en las reglas
% 2 y 3 debido a que usan el mismo conjunto difuso singleton de salida.
or23 = max(lr2,lr3);
%método som para la desdifusión.
if lr1 ~= 0
    if or23 ~= 0
        if lr4 ~= 0
            if or23 >= lr4
                if lr1 >= or23
                    des = lr1 * H;
                else des = or23 * 0; % el valor del conjunto difuso singl. "Cero" es 0.
                end
            elseif lr1 >= lr4
                des = lr1 * H;
            else des = lr4 * (-H);
            end
        else
            if lr1 >= or23
                des = lr1 * H;
            else des = or23 * 0;
            end
        end
    else
        if lr4 == 0
            des = lr1 * H;
        else
            if lr1 >= lr4
                des = lr1 * H;
            else des = lr4 * (-H);
            end
        end
    end
else
    if or23 == 0
        des = lr4 * (-H);
    end
end

```

```
elseif lr4 == 0
    des = or23 * 0;
else
    if or23 >= lr4
        des = or23 * 0;
    else des = lr4 * (-H);
    end
end
end
end
```

En la figura A2 vemos el sistema de control difuso PI con la sustitución de la herramienta de lógica difusa por el bloque donde está el programa de control difuso y en la figura A3 tenemos la respuesta del controlador que es prácticamente igual a la del controlador con la herramienta de lógica difusa, usando el método de desfusión som, smallest of maximum (último de los máximos).

Controlador de un motor de DC (Discreto)

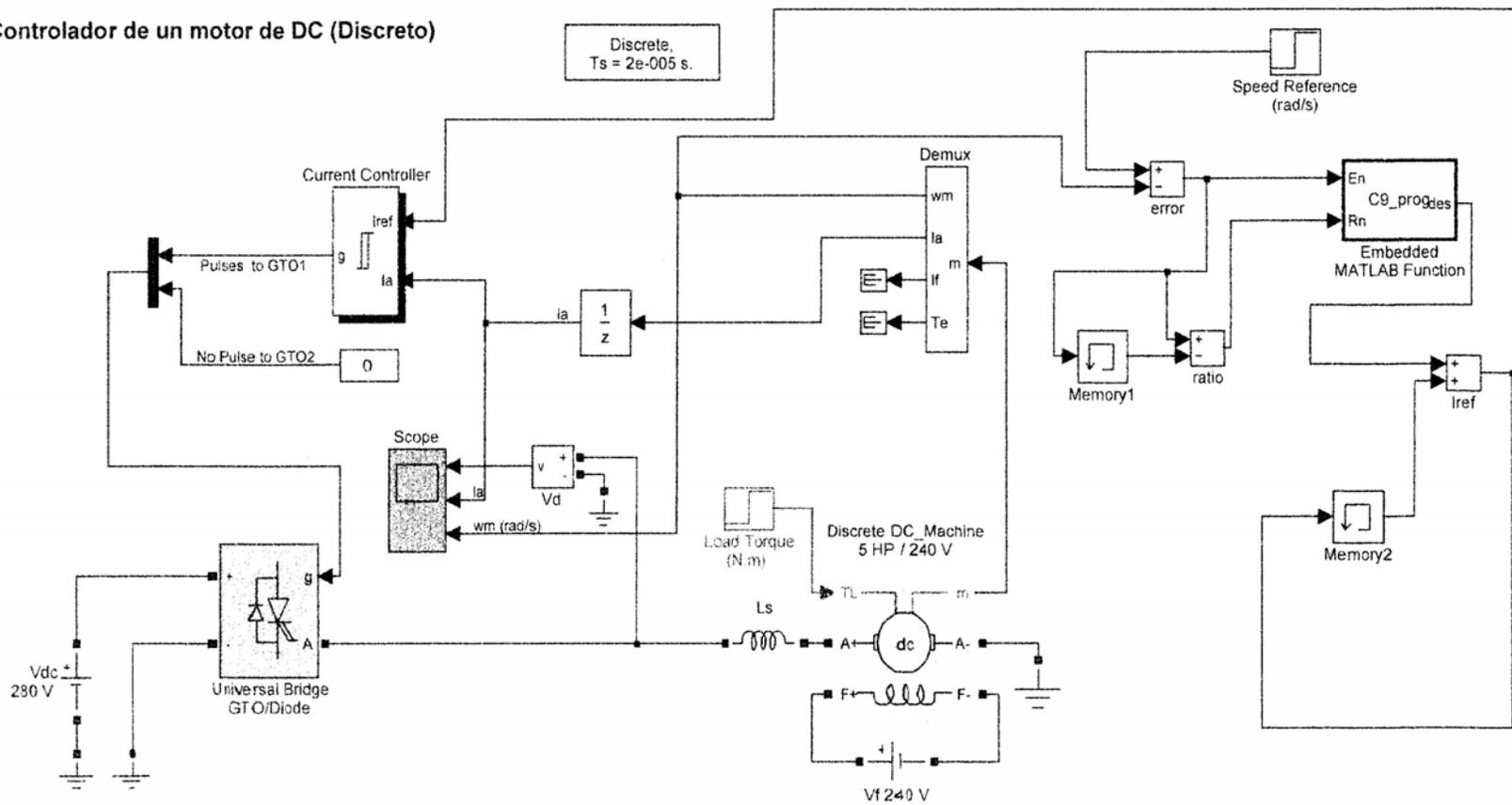


Fig. A2 Sistema de control difuso PI con el bloque para programación.

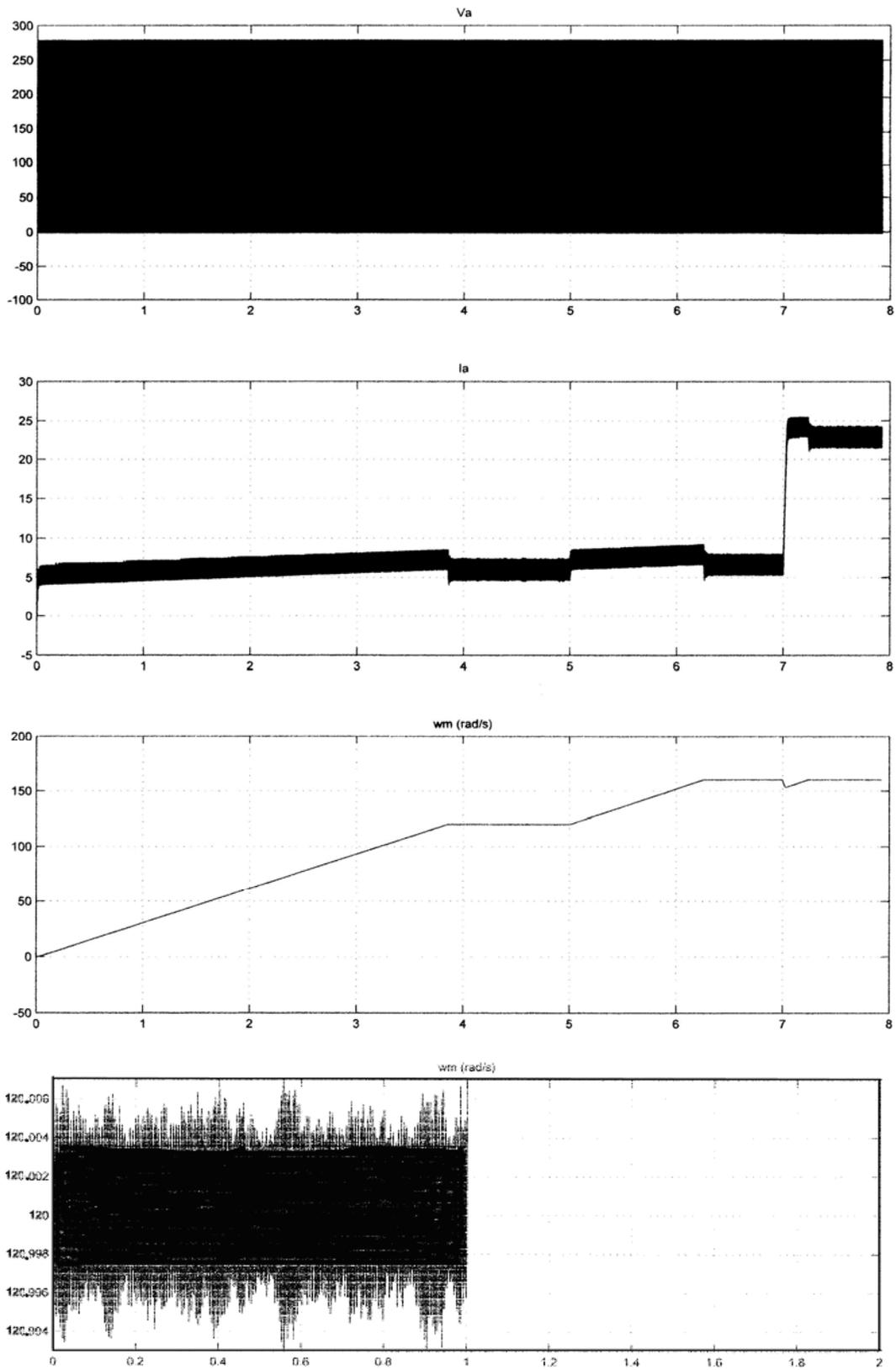


Fig. A3 Respuesta del controlador difuso programado con $T_s = 20 \mu s$.

BIBLIOGRAFÍA

- [1] Fuzzy Control and Modeling
Hao Ying
IEEE PRESS
2000.

- [2] A Course in Fuzzy Systems and Control
Li-Xin Wang
Prentice Hall Inc
1997.

- [3] An Introduction to Fuzzy Control
Dimitar Driandikov, Hans Hellendoorn y Michael Reinfrank
Springer-Verlag Berlin Heidelberg, New York
1996.

- [4] Ingeniería de Control Moderna
Katsuhiko Ogata
Prentice Hall Hispanoamericana S.A.,
1998.

- [5] Controles Automáticos
Howard I. Harrison y Jhon G. Bollinger
Editorial Trillas S.A.
1979.

- [6] Maquinas Eléctricas, Rotativas y Transformadores
Donald V. Richardson y Arthur J. Caisse
Prentice Hall Hispanoamérica S.A.
1997.

- [7] Maquinas Eléctricas
Stephen J. Chapman
Mc Graw-Hill Interamericana S.A.
2000

- [8] MATLAB 7.0.