



**TECNOLÓGICO
DE MONTERREY.**



**TECNOLÓGICO
DE MONTERREY.**

BIBLIOTECA

Campus Ciudad de México

Instituto Tecnológico y de Estudios Superiores de Monterrey

**Campus Ciudad de México
División de Ingeniería y Arquitectura**

Ingeniería en Mecatrónica
Departamento de Ingeniería Eléctrica y Electrónica

Implementación de la Inteligencia Artificial en el Tratamiento Termoquímico de Borurización en Pasta

Autores: María del Carmen Islas Bravo
Enzo Arturo González y González



Asesores: Dr. Pedro Ponce. Dr. Iván Campos.

México D.F, a 13 de mayo de 2005.

Este proyecto es auspiciado por CONACYT con la clave 42883/A-1

ÍNDICE

Introducción.....	3
Objetivos	4
Estado del arte.....	5
Marco teórico.....	8
Borurización.....	8
Inteligencia Artificial.....	9
Método Mamdani.....	10
Método Takagi-Sugeno.....	14
Método de “clusters”.....	14
Redes Neuronales Artificiales.....	18
Resultados.....	23
Experimentación.....	23
Modelo matemático.....	26
Lógica Difusa.....	27
Método Mamdani.....	27
Método Takagi-Sugeno.....	32
Método de “clusters”.....	34
Redes Neuronales Artificiales.....	36
Evaluación del error.....	39
Contribución al estado del arte.....	40
Conclusiones.....	40
Trabajo a futuro.....	44
Agradecimientos.....	46
Referencias Bibliográficas.....	48
Anexo 1.....	50
Anexo 2.....	51
Anexo 3.....	53
Anexo 4.....	54
Anexo 5.....	56
Glosario.....	58

1. INTRODUCCIÓN

La inteligencia artificial incluye tecnologías como los sistemas expertos, las redes neuronales, los algoritmos genéticos, los sistemas difusos, los sistemas caóticos y los sistemas de anticipación, entre otros. Algunas de estas tecnologías tienen sus orígenes en fenómenos biológicos o de comportamiento relacionados con los humanos o animales, y otras son simplemente análogas. Para el caso que nos interesa desarrollar, el enfoque será principalmente en sistemas difusos y las redes neuronales artificiales.

El objetivo del procesamiento de información y de datos es entender los fenómenos que se ven involucrados así como evaluar de forma cualitativa los parámetros del proceso. Esto se logra mediante el modelado de los sistemas, ya sea de forma experimental o analítica. Una vez obtenido el modelo del sistema, se utilizan técnicas (análisis sensitivo, regresión estadística, etc.) para tener una mejor comprensión del mismo. Sin embargo existen situaciones donde los fenómenos involucrados son complejos y no son comprendidos adecuadamente, provocando que el modelado a partir de sus principios no sea posible. Más común es que las medidas físicas de las cantidades pertinentes del sistema son difíciles y caras de obtener. Estas complicaciones pueden ser resueltas con las técnicas antes citadas (sistemas difusos y redes neuronales artificiales) para obtener el modelo a partir de las mediciones experimentales.

En la historia de la ciencia y la tecnología, los nuevos desarrollos vienen comúnmente de observaciones realizadas con diferentes perspectivas. Entre estas observaciones está el análisis de las fluctuaciones en las variables de los procesos, las cuales proveen de información útil acerca de estos últimos. Estas fluctuaciones se manifiestan como incertidumbres en las variables medidas.

Las redes neuronales artificiales y los sistemas difusos son dos métodos distintos para tratar incertidumbre, la cual se presenta debido a la complejidad del sistema (no linealidad).

Las redes neuronales aproximan la representación del modelo usando entradas y salidas precisas para “entrenar” un modelo genérico, el cual cuenta con suficientes grados de libertad para formular una buena aproximación a la relación compleja entre las variables.

En el caso de los sistemas difusos, las variables de entrada y salida son codificadas en representaciones “difusas”, mientras que sus relaciones se rigen por reglas definidas de tipo “si... entonces...”. Zadeh (el creador de la lógica difusa) mostró con sus observaciones que la búsqueda de la precisión no solo puede ser innecesaria sino además una fuente de error.

Cada una de estas aproximaciones tiene sus propias ventajas y desventajas. Las redes neuronales artificiales pueden representar relaciones no lineales complejas, y son útiles para la clasificación de fenómenos en categorías preseleccionadas, utilizadas para el entrenamiento. Por otro lado la precisión de las salidas está limitada en ocasiones porque las variables son tratadas como análogas (incluso cuando son implementadas en computadoras digitales) y la “minimización del error

cuadrado” no significa “error cero”. Además, el tiempo de entrenamiento de una red neuronal artificial puede ser grande (puede tardar horas, incluso días). Una de las debilidades de las redes neuronales artificiales es la necesidad de datos substanciales que sean representativos y que cubran el rango entero sobre el cual se espera que las variables involucradas cambien.

Los sistemas difusos captan la imprecisión de las variables de entrada y de salida directamente al definir las en valores difusos que pueden ser expresados de forma lingüística (ejemplo: caliente, frío, tibio). Ofrecen flexibilidad a la hora de formular una descripción del sistema a un nivel de detalle apropiado. Con los sistemas basados en lógica difusa, el comportamiento de los procesos puede ser descrito en términos generales sin definir precisamente los fenómenos complejos involucrados (generalmente no lineales). Los sistemas difusos son fáciles de formular, modificar y tolerantes al cambio e inclusive a la falla.

Las redes neuronales artificiales y las tecnologías con lógica difusa son muy diferentes, y cada una cuenta con capacidades que son útiles en el procesamiento de información. Pueden ser usadas para generar los mismos resultados pero por caminos diferentes. Son utilizadas para controlar sistemas no lineales a un grado que no es posible alcanzar utilizando sistemas de control lineales convencionales. Realizan mapeos con cierto grado de imprecisión. Inclusive, pueden combinarse entre ellos y con otras tecnologías de inteligencia artificial para obtener sistemas mucho más confiables [1].

Las herramientas que utilizamos para el modelado de nuestros sistemas son la lógica difusa y las redes neuronales artificiales. Hoy en día existen varios sistemas que son regidos por la lógica difusa, sobretodo en el diseño de electrodomésticos y cámaras electrónicas. Para el caso de redes neuronales artificiales, existen sistemas de reconocimiento de voz, de imágenes y de caracteres que utilizan esta técnica. Por ello creemos que nuevos procesos pueden ser sometidos al modelado con lógica difusa y con redes neuronales artificiales, y entre ellos están los procesos térmicos como la borurización.

2. OBJETIVOS

2.1 Objetivo General

Modelar el tratamiento termoquímico de borurización, a partir de tres variables (potencial de boro, temperatura y tiempo) por medio de inteligencia artificial, aportando una nueva alternativa de modelo multivariable al área de investigación que estudia este proceso.

2.2 Objetivos Particulares

Los objetivos particulares de este proyecto son:

- Evaluar los resultados experimentales obtenidos de un proceso termoquímico de borurización: espesor de la capa del boruro Fe₂B formada.
- Clasificar la información del proceso de borurización en pasta en un acero 1045 a temperaturas de entre 920°-1000°C, con diferentes espesores de pasta en la superficie (1 – 5 mm). La experimentación fue realizada de enero – noviembre 2004.
- A partir de un modelo matemático ya planteado [2], obtener el crecimiento de capa y coeficientes de difusión del boruro Fe₂B en el sustrato.
- Familiarizarnos con temas relacionados a la inteligencia artificial, entre ellos lógica difusa y redes neuronales artificiales; así como con programas que nos ayuden a simularlas.
- Proponer sistemas de inteligencia artificial para modelar el proceso de borurización en un acero AISI 1045 y argumentar su viabilidad.
 - o Diseñar un sistema basado en lógica difusa.
 - o Diseñar un sistema basado en redes neuronales artificiales.
- Confrontación de resultados experimentales, del modelo matemático y del modelo a base de inteligencia artificial, comparando el error generado al calcular el espesor de capa.
- Encontrar las ventajas que ofrece un modelado por medio de inteligencia artificial ante un modelo matemático, de acuerdo al error obtenido en cada uno.
- Concluir sobre las ventajas y desventajas que presenta cada modelo al ser comparando con otros y con el proceso experimental.
- Publicar dos artículos sobre el proyecto.

3. ESTADO DEL ARTE

3.1 Redes Neuronales Artificiales

La respuesta de un acero AISI W1 a un proceso de borurización fue modelada por Kenan Genel por medio de redes neuronales artificiales. Este trabajo fue presentado en *Surface and Coatings Technology* en el año 2002. En éste, se presentan los resultados de la investigación realizada al buscar posibilidades para predecir la variación de la dureza y del espesor de capa utilizando redes neuronales artificiales. El algoritmo de aprendizaje utilizado fue *back-propagation*. Se establecieron relaciones empíricas entre parámetros del proceso y capa borurizada, así como microdureza [2].

A continuación se muestra la estructura de la red neuronal artificial utilizada en el trabajo mencionado:

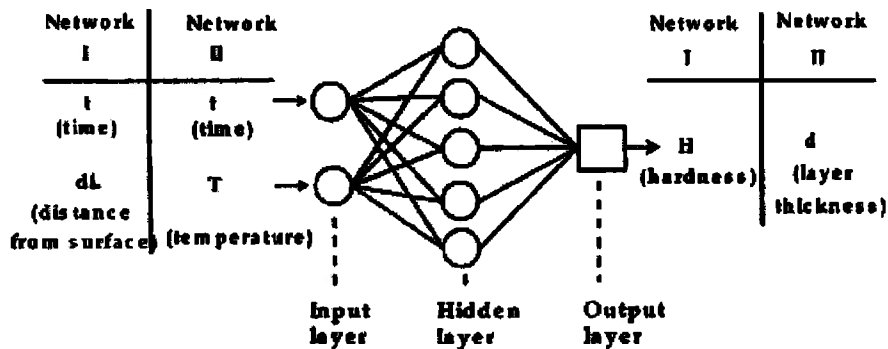


Figura 1. Estructura de red neuronal artificial para borurización utilizada por K. Genel et al. [2].

Como se puede observar, se utilizan dos estructuras iguales para poder obtener dos variables diferentes. En la primera, con las variables de entrada de *tiempo* y *distancia desde la superficie* se obtiene *dureza*; mientras que en la segunda al introducir el *tiempo* y *temperatura* se obtiene *espesor de la capa*.

El aprendizaje de la red neuronal artificial se realizó con resultados experimentales a determinados tiempos y temperaturas. La validación de la red neuronal artificial se obtuvo comparando resultados simulados contra experimentales con tiempos y temperaturas distintos a los valores utilizados para realizar el entrenamiento de la red. Los resultados obtenidos con la simulación del modelo a base de redes neuronales artificiales se acercaron a los valores reales con aproximadamente 3% de error.

3.2 Lógica Difusa

Por otro lado la relación entre temperatura, tiempo y dureza fue modelada de forma adaptativa para un ADI (*Austempered Ductil Iron*) por Labib Arafteh. La estructura del modelo difuso tipo Mamdani propuesto se muestra en la Figura 2.

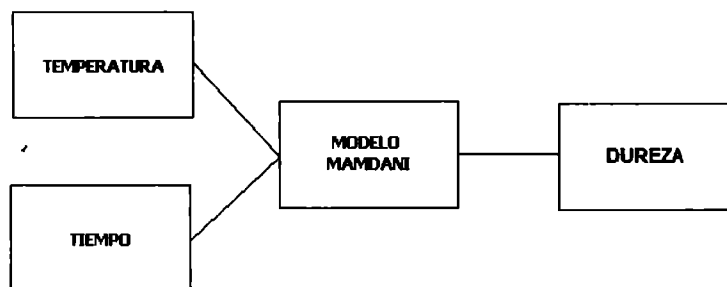


Figura 2. Diagrama de bloques del modelo Mamdani para un ADI. [3]

Los valores de dureza del hierro son afectados por la temperatura y tiempo de austenitización para composiciones químicas fijas. Por lo tanto, dentro del sistema difuso las entradas son *temperatura* y *tiempo* de austenitización, que generan como salida *el valor de dureza*.

El modelo general de las reglas lingüísticas usadas en este sistema es de la siguiente forma: Si “Temperatura de Austenitización” es “Temperatura” y “Tiempo de Austenitización” es “Tiempo”, entonces “Dureza” es “Dureza”.

El proceso para modelar el sistema difuso consistió en definir las entradas y la salida mencionadas anteriormente; definir rangos para las entradas y la salida; definir las funciones de membresía para cada una de las variables; establecer las relaciones entre variables mediante reglas lingüísticas; elegir el método de evaluación de las reglas y el método de obtención de las salidas nítidas; finalmente simular el sistema para generar resultados. Se utilizó el método de inferencia “min-max” para obtener la salida difusa, y el método de centro de gravedad para la obtención de las salidas nítidas.

Se probó el sistema y se sintonizaron las funciones de membresía hasta que la diferencia entre la experimentación y la modelación fuera razonable. En la Figura 3 se muestran las funciones de membresía para la entrada y la salida.

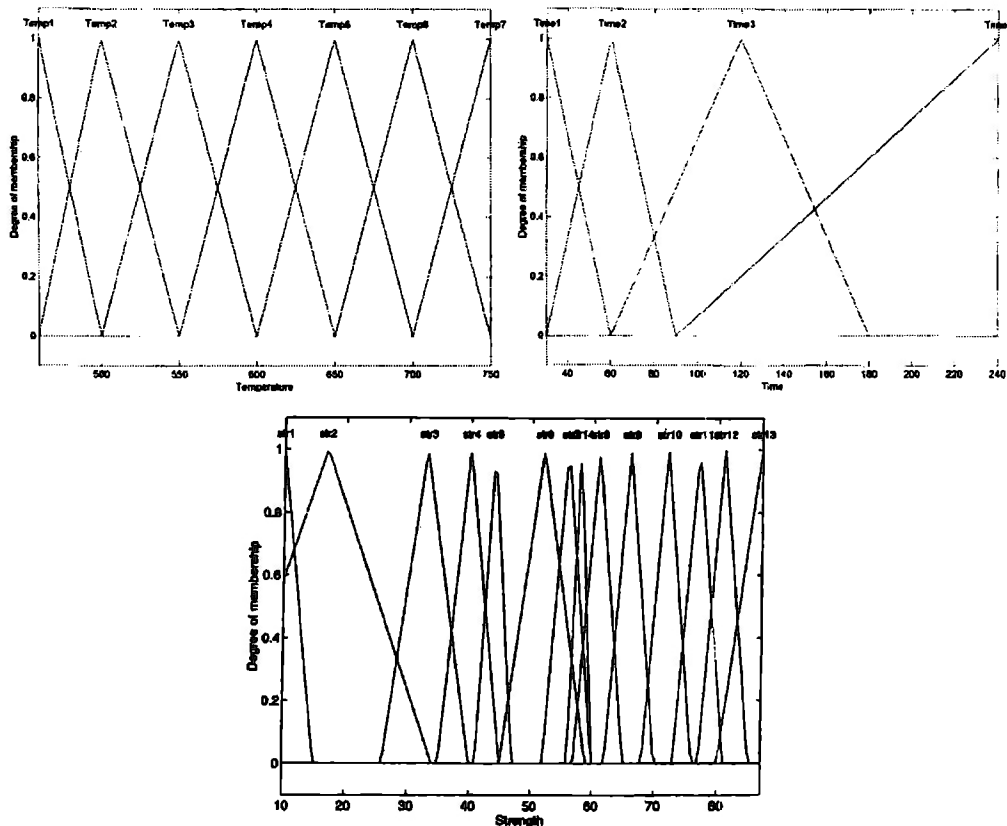


Figura 3. Funciones de membresía de las entradas y la salida para el ADI [3].

Como se puede observar las funciones de membresía no tienen un rango equivalente entre sí, sino que sus valores difieren. Esto se hizo para la sintonización de la entrada de tiempo y la salida; para así poder obtener un error razonable.

Para probar la confiabilidad del sistema difuso se utilizó el coeficiente de correlación, mostrado en la Ecuación 1.

$$r_{xy} = \sqrt{\frac{1 - \sum_i (y_i - f(x_i))^2}{\sum_i (y_i - \bar{y})^2}} \quad \dots\dots\dots (1)$$

Donde y_i es la i -ésimo dato actual, \bar{y} es el promedio de todos los datos actuales y $f(x_i)$ es el i -ésimo dato que se predice. Para el sistema difuso del ADI, se obtiene un coeficiente de correlación de 0.87 [3].

4. MARCO TEÓRICO

4.1 Borurización

La borurización es un proceso termoquímico superficial de endurecimiento que consiste en la saturación de boro en superficies de materiales ferrosos y no ferrosos. El proceso implica el calentamiento de un material limpio a un rango de temperaturas que van desde 700 hasta 1000° C con un tiempo de tratamiento que oscila entre 1 y 12 horas para métodos en polvo, y en el caso de borurización en pasta los tiempos de tratamiento fluctúan entre 1 y 8 horas. Este tipo de procesos se aplica para poder elevar la dureza en la superficie y así mejorar las propiedades mecánicas del material como puede ser la resistencia a la corrosión, desgaste, abrasión y fatiga.

Durante la borurización, al difundirse el boro en un material ferroso se espera la formación de boruros en dos fases conocidas como FeB y Fe₂B dependiendo del potencial de boro que rodea al sustrato. Dichas fases están constituidas por microestructuras cristalinas, ortorrómbica y tetragonal centrada en el cuerpo respectivamente. Los cristales tienen un crecimiento preferencial hacia donde encuentran menor resistencia de crecimiento, es decir, perpendicular al sustrato, produciendo una morfología aserrada en la capa borurizada particularmente en aceros de bajo a medio carbono. La morfología, crecimiento y composición de las fases de la capa borurizada puede ser influenciada por materiales aleantes presentes en el sustrato; por ejemplo, los dientes de boruros de la capa presente disminuyen conforme se incrementa el contenido de cromo y níquel presente en el acero. La microdureza de la capa también depende de la composición y estructura de la misma, así como de la composición del material base [4].

Los medios de difusión que se emplean son sólido, líquido y gaseoso (mezclas de polvos, sales y óxidos fundidos, medios gaseosos y a base de pastas). Dentro de la industria el medio sólido es el más utilizado por razones tanto tecnológicas, como económicas. En el medio sólido uno puede seleccionar entre la tecnología de pasta o de polvo. Se sugiere utilizar el medio a base de pasta cuando se requiere disminuir el trabajo manual utilizado en la borurización en polvo, esto ofrece ventajas al trabajar con altos volúmenes de piezas. El medio a base de pasta es recomendable para

tratamientos selectivos, además de que permite la obtención de capas superiores que en tratamientos similares de endurecimiento [5].

La pasta contiene aproximadamente 76% en peso de boro, con una composición de B_4C , la cual es disuelta en agua o criolita (Na_3AlF_6). Las relaciones de agua/pasta afectan drásticamente al potencial de boro necesario para la formación de las fases boruradas, disminuyendo conforme aumenta la relación mencionada anteriormente. Específicamente en trabajos anteriores, se ha verificado que el espesor de pasta sobre el substrato implica una dependencia en la movilidad de boro en las capas boruradas [9].

Una de las principales ventajas de las capas borurizadas es que pueden alcanzar grandes valores de dureza (entre 1450 – 5000 HV), mantenidos a altas temperaturas. La dureza de las capas borurizadas producidas en aceros al carbono es mucho mayor que las producidas por cualquier otro tratamiento superficial convencional (nitruración, carburización). La combinación de su alta dureza superficial y bajo coeficiente de fricción en la superficie de la capa, nos da contribuciones importantes para combatir la abrasión, oxidación, adhesión y fatiga.

Las desventajas principales son que debido a su gran dureza existe gran fragilidad, con el aumento de ésta disminuye la adherencia de las capas. El proceso es inflexible a nivel industrial, ofreciendo así, menor costo – beneficio que otros procesos convencionales [4].

A partir de lo anterior surge la necesidad de explorar nuevos campos y técnicas de modelado, tales como tratamientos termoquímicos multicomponenciales, análisis fractal e inteligencia artificial, con el objetivo de modelar el crecimiento de la capa borurizada.

4.2 Inteligencia Artificial

Durante el desarrollo del proyecto se utilizaron dos técnicas básicas de la inteligencia artificial: la lógica difusa y las redes neuronales artificiales. Como se describió en la introducción, cada una tiene ventajas y desventajas, sin embargo el interés específico es encontrar el sistema que obtenga el menor error al comparar los resultados simulados con los resultados experimentales. A continuación describiremos las bases de cada una de estas técnicas.

4.2.1 LÓGICA DIFUSA

Dentro del campo de la inteligencia artificial se encuentra la lógica difusa, que es una de las técnicas que usaremos para el diseño del sistema de predicción para el proceso de borurización. Debemos aclarar que no existe un verbo en español tal para las acciones de cada una de las etapas del diseño de un sistema basado en lógica difusa, por lo que las llamaremos por su nombre coloquial en español.

Los métodos de lógica difusa que se utilizaron para la borurización son el método Mamdani y el método Takagi-Sugeno, los cuales se explicarán a continuación.

4.2.1.1 Método Mamdani

Para diseñar un sistema tipo Mamdani con lógica difusa, se utiliza la siguiente metodología:

- "Fuzzificar"

Es la primera etapa, donde se convierten los valores nítidos de un sistema a valores difusos. Por ejemplo, para el caso de la borurización, una entrada nítida de 6 horas, será convertida a forma difusa en *mucho tiempo*. Para poder realizar esta transformación, se necesitan definir las funciones de membresía para las entradas.

Las funciones de membresía son mapeos de conjuntos de valores nítidos. Cada función de membresía está descrita por valores difusos llamados grados de pertenencia. A las funciones de membresía se les distingue por denominaciones lingüísticas, imitando la forma en que el ser humano cuantifica alguna variable de forma coloquial. Un ejemplo de funciones de membresía para la borurización está dado en la Figura 4.

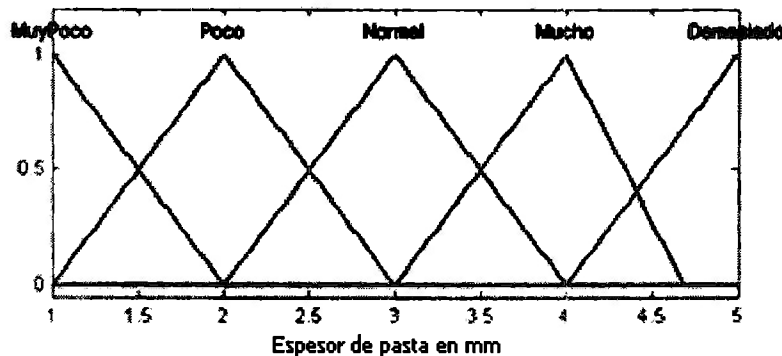


Figura 4. Ejemplos de funciones de membresía, obtenido mediante Matlab [6].

En esta figura podemos observar cinco funciones de membresía, las cuales están definidas para distintos rangos de valores nítidos, que en este caso representan el espesor. Es decir, para un rango de 1 a 3 mm en forma nítida, existe la función de membresía de *poco* espesor en forma difusa, pero también se observa la presencia de una parte de los conjuntos *muy poco* y *normal* espesor. En el eje perpendicular encontramos los grados de pertenencia, el cual va de 0 a 1. Es decir, para un valor nítido dado, se proyecta hacia las funciones de membresía y se observa a qué funciones de membresía puede pertenecer así como su grado de pertenencia a la misma.

Es importante hablar de la geometría de las funciones de membresía. La forma de la función de membresía afecta directamente a los requerimientos de tiempo y espacio para que se efectúe la "fuzzificación" y la "desfuzzificación". Las formas más comunes que se usan son la forma triangular y la forma trapezoidal, pero también están la forma de campana de Gauss así como la forma puntual, la cual es muy utilizada sobretodo para las funciones de membresía de la salida, ya que de esta forma es más fácil obtener el valor de la salida nítido, de lo cual hablaremos más adelante. Las siguientes ecuaciones de la Figura 5 representan las diferentes formas que puede tener una función de membresía.

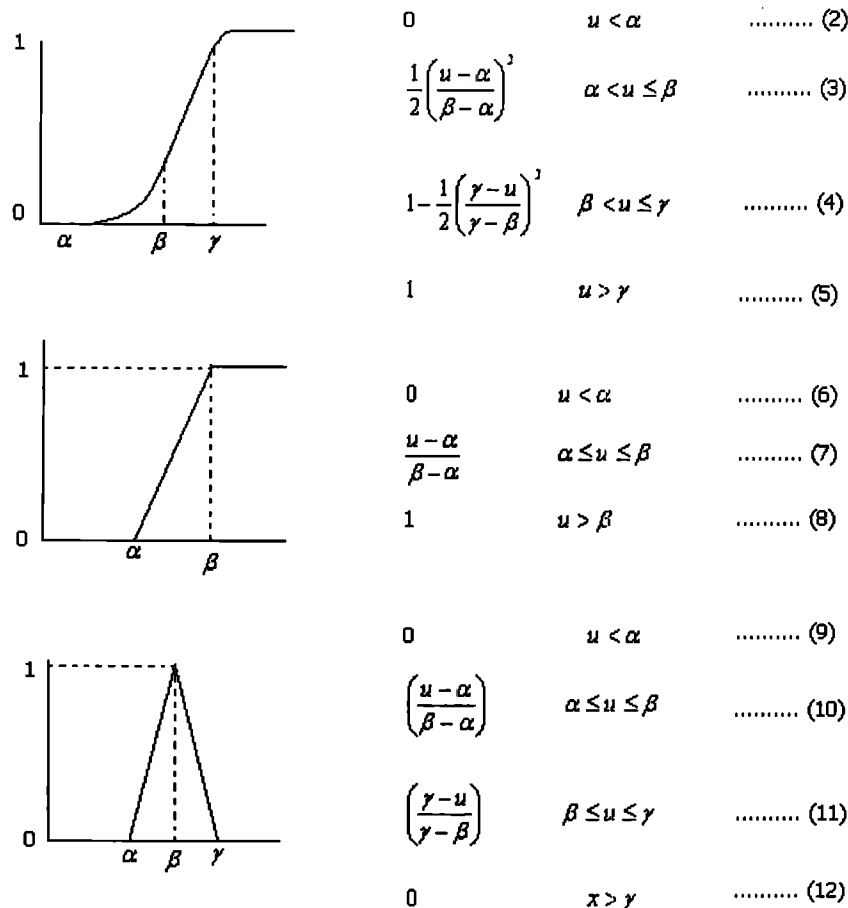


Figura 5. Geometría de las funciones de membresía [7].

De igual forma para la salida, se necesitan definir funciones de membresía para cada una de los rangos de la salida deseados.

- Evaluación de reglas

El siguiente paso en el diseño de un sistema con lógica difusa es la creación de las reglas lingüísticas. Las reglas lingüísticas son del tipo “si...entonces...”, es decir, son implicaciones, donde se evalúan los antecedentes para dar una consecuencia. Se combinan las funciones de membresía de la o las entradas entre sí para dar una salida difusa, la cual deberá estar dentro de las funciones de membresía de la salida anteriormente definidas. Un ejemplo de reglas lingüísticas para el caso de la borurización son las siguientes:

- Si el tiempo es *nada* entonces el crecimiento de capa es *cero*.
- Si el tiempo es *poco* entonces el crecimiento de capa es *poco*.
- Si el tiempo es *normal* entonces el crecimiento es *normal*.
- Si el tiempo es *mucho* entonces el crecimiento es *grande*.
- Si el tiempo es *demasiado* entonces el crecimiento es *muy grande*.

Cuando se tienen dos o más variables de entrada y/o de salida, las reglas lingüísticas utilizan la operación “booleana” AND en la mayoría de los casos, sin embargo también pueden ser utilizadas las operaciones OR y NOT [8].

Ya que las reglas lingüísticas están definidas, ahora lo que prosigue es evaluar la relevancia o grado de pertenencia de cada antecedente de la regla.

Por ejemplo, si la entrada nítida para el sistema difuso de borurización es de 5 hrs, podemos ver en la Figura 6 que ese valor proyectado pasa por 2 funciones de membresía, pero a un grado de pertenencia distinto.

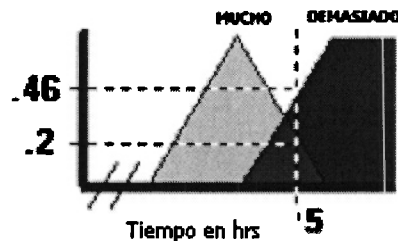


Figura 6. Grados de pertenencia [8].

Ya que la relevancia de cada antecedente ha sido determinada, el siguiente paso es encontrar el valor de verdad (la fuerza de la regla) para cada regla. Debido a que los antecedentes de las reglas están ligados con el operador AND, entonces se toma el valor mínimo de los antecedentes que forman la regla a evaluar. Este valor mínimo será el valor de verdad de la regla.

El siguiente paso es obtener todos los valores de verdad de todas las reglas, y hacer una comparación entre las reglas que tengan la misma consecuencia, para tomar el valor máximo de verdad que existe entre ellas. Este método se llama el método de composición de mínimos-máximos, ya que se toma el valor mínimo de los antecedentes de una regla para después tomar el valor máximo de las reglas que tengan la misma consecuencia. La regla que tenga el mayor valor de verdad es la que dominará en la salida [7-8].

- “Desfuzzificar”

En esta etapa todas las salidas difusas significativas se combinan para dar una salida nítida. Para ello se utiliza una técnica llamada Centro de Gravedad (CDG). En este método, cada función de membresía de la salida es acotada por el valor de la salida difusa, obtenido al evaluar las reglas lingüísticas. Este acotamiento se le llama corte lambda.

Al realizar el método min-max en la evaluación de las reglas lingüísticas, se obtuvieron valores de pertenencia para cada una de las funciones de membresía de la salida. Por ejemplo, en la Figura 7 se muestran funciones de membresía de salida con sus respectivos valores de pertenencia, para evaluar el crecimiento de capa dentro del proceso de borurización.

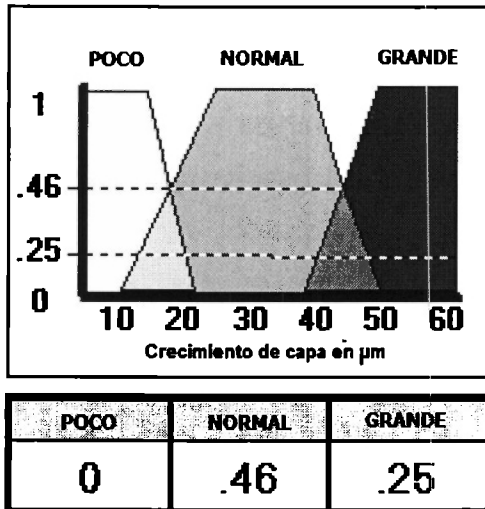


Figura 7. Grados de pertenencia de la salida [8].

Para este ejemplo, la salida será una combinación de *normal* y de *grande* con 0.46 y 0.25 de valor de pertenencia respectivamente. Se proyecta una sombra sobre las funciones de membresía que participan en la salida, teniendo como límites los valores de los grados de pertenencia. Esto se muestra en la Figura 8.

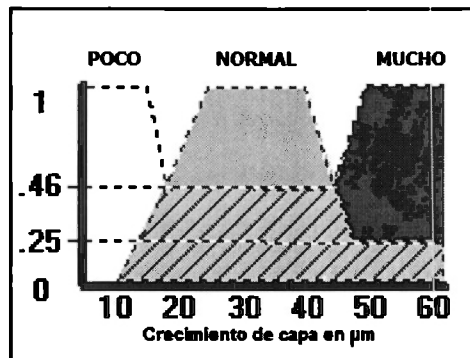


Figura 8. Sombra proyectada sobre las funciones de membresía de la salida [8].

Finalmente, el valor nítido se obtiene al encontrar el centro de gravedad de la sombra proyectada. Esto se logra mediante la siguiente ecuación.

$$Salida_real = \frac{\sum_{x=a}^b \mu(x)x}{\sum_{x=a}^b \mu(x)} \dots\dots\dots (13)$$

Donde $\mu(x)$ es el grado de pertenencia y x son los valores que están dentro de los rangos de las funciones de membresía de salida afectadas. Resolviendo obtenemos el valor nítido de salida.

Anteriormente habíamos mencionado que se sugiere que las funciones de membresía de la salida sean de la forma puntual. Esto es recomendable ya que sólo van a tener un valor y no un rango de valores, por lo que se simplifica el procedimiento para resolver la ecuación, ahorrando espacio y tiempo cuando un microcontrolador esté efectuando la operación [7].

4.2.1.2 Método Takagi – Sugeno

La variación que existe entre el método Mamdani y el método Takagi-Sugeno radica en la salida, ya que para Mamdani, las reglas que definen la salida están regidas por valores difusos, mientras que en el Sugeno están definidas por ecuaciones polinomiales.

Para el caso de Sugeno, la etapa de “fuzzificación” y la definición de reglas se realizan de igual modo que en el método de Mamdani. La diferencia se encuentra en la evaluación de reglas, las cuales son del siguiente tipo: Si tiempo *mucho* y espesor de pasta *normal* entonces crecimiento de capa es $X1+X2$.

Por lo tanto, en lugar de obtener un valor definido dentro de una función de membresía de salida, se evalúa una ecuación para determinados valores. El valor nítido de la salida se obtiene mediante la siguiente ecuación:

$$Salida_real = \frac{\sum |y = y^i| \cdot y^i}{\sum |y = y^i|} \dots\dots\dots (14)$$

Donde y^i son los resultados de la evaluación de las ecuaciones polinomiales de cada una de las reglas activadas y $|y = y^i|$ son los grados de pertenencia del mínimo de los antecedentes de cada una de las reglas activadas.

En realidad, el método Mamdani es un caso particular del método Takagi-Sugeno, ya que se pueden proponer ecuaciones que den los mismos valores que una función de membresía puntual. Cuando se utiliza el método Sugeno el sistema se está generalizado [7].

4.2.2 MÉTODO DE “CLUSTERS”

El método de “clusters” consiste en dividir una serie de datos en clases homogéneas o “clusters”, en donde los datos de la misma clase son lo más similar posible entre ellos, mientras que los datos de distintas clases son lo más diferente posible entre ellos [21].

Un ejemplo de “clusters” se aprecia en la siguiente Figura 9.

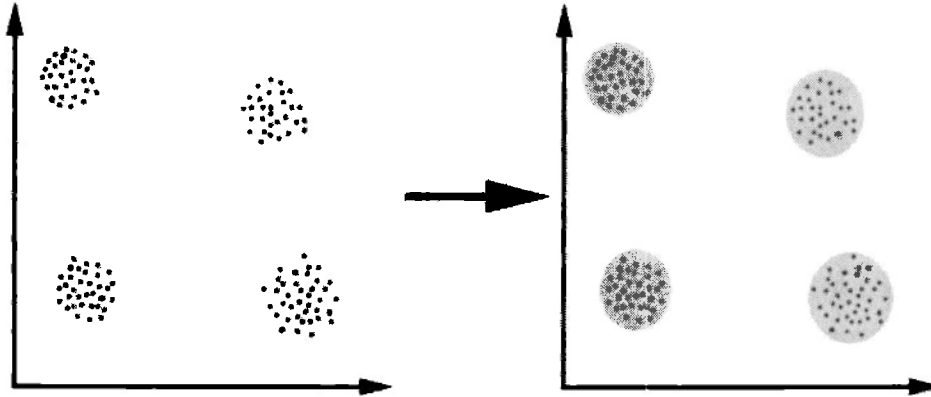


Figura 9. Identificación de posibles "clusters" en un plano.

Aquí se identifican claramente los cuatro posibles "clusters" en que los datos pueden ser divididos. El criterio de similitud es la distancia: dos o más objetos pertenecen al mismo "cluster" si son cercanos entre sí, de acuerdo a una distancia dada (distancia euclidiana).

Otro posible criterio de similitud es la descripción, es decir, dos o más objetos pertenecen al mismo "cluster" si éste tiene una característica descriptiva que es común a todos los elementos integrantes.

Para la formación de los "clusters", se utiliza un algoritmo. Existen varios tipos de algoritmos, siendo el algoritmo de traslape (también conocido como Fuzzy C-Means) el que nos interesa. Este algoritmo utiliza conjuntos difusos para agrupar los datos, por lo que cada punto puede pertenecer a dos o más "clusters" con diferentes grados de pertenencia [21]. El método está basado en la minimización de la siguiente ecuación:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty \dots\dots\dots (15)$$

Donde m es cualquier número mayor a 1, u_{ij} es el grado de pertenencia de x_i en el "cluster" j , x_i es el i -ésimo dato medido, c_j es el centro del "cluster" y $\|*\|$ es la norma que expresa la similitud entre cualquier dato medido y su centro.

La partición difusa se realiza a través de una optimización iterativa de la función que se muestra a continuación, con la actualización de la membresía u_{ij} y el centro del "cluster" c_j como sigue [21]:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \dots\dots\dots (16)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \dots\dots\dots (17)$$

Esta iteración se detendrá cuando $\max_{ij} \{u_{ij}^{(k+1)} - u_{ij}^{(k)}\} < \varepsilon \dots (18)$, donde ε es el criterio de finalización entre 0 y 1, y k son los pasos de iteración. Este procedimiento converge a un mínimo local de J_m .

En general, el algoritmo se compone de los siguientes pasos:

- Inicializar la matriz $U=[u_{ij}]$, matriz $U^{(0)}$.
- A un paso de k , calcular los centros de los vectores $C^{(k)}=[c_j]$ con $U^{(k)}$. En este caso se utiliza la Ecuación 17.
- Actualizar $U^{(k)}$, $U^{(k+1)}$, utilizando la Ecuación 16.
- Si $\|U^{(k+1)} - U^{(k)}\| < \varepsilon \dots (28)$ entonces se detiene el proceso, de otra forma se regresa al paso 2.

Los datos están ligados a cada “cluster” mediante la función de membresía, la cual representa el comportamiento difuso del algoritmo. Para lograr esto, se necesita construir una matriz aproximada U , cuyos factores sean números entre 0 y 1, y representen el grado de pertenencia entre los datos y los centros de los “clusters” [21].

Un ejemplo de esto es el siguiente. Dado un conjunto de datos, se representan de forma distribuida sobre un eje, como se muestra en la Figura 10.



Figura 10. Conjunto de datos para ser agrupados en “clusters”.

Al observar la figura, se pueden identificar dos “clusters”, a partir de las dos concentraciones de datos vistas. Se etiquetarán como A y B. Para el caso del algoritmo FCM, un dato dado no pertenece exclusivamente a un “cluster”. En este caso, la función de membresía cuenta con una línea suave para indicar que cada dato puede pertenecer a distintos “clusters” con diferentes grados de pertenencia. Esto se observa en la Figura 11.

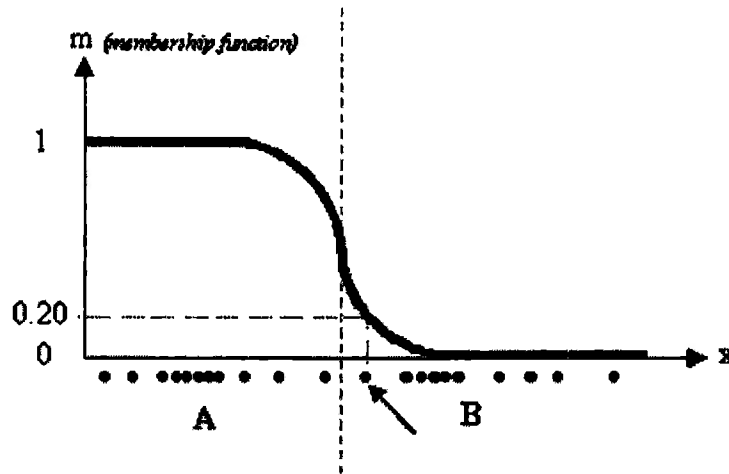


Figura 11. Datos con diferentes grados de pertenencia.

En la figura anterior, el dato señalado pertenece más al “cluster” B que al “cluster” A. El valor de 0.2 indica el grado de pertenencia en A para ese dato en particular. En lugar de usar una representación gráfica, se introduce entonces una matriz U cuyos factores son tomados de las funciones de membresía. La matriz se observa en la Figura 12.

$$U_{N \times C} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \dots & \dots \\ 0.9 & 0.1 \end{bmatrix}$$

Figura 12. Matriz de grados de pertenencia.

El número de filas y de columnas depende de cuántos datos y “clusters” se están considerando. Para este caso se tiene C = 2 columnas (2 “clusters”) y N filas, donde C es el número total de “clusters” y N es el número total de datos. El elemento genérico es indicado por u_{ij} [21].

Algunas propiedades importantes son las siguientes:

$$u_{ij} \in [0,1] \quad \forall \quad i, j \dots\dots\dots (19)$$

$$\sum_{j=1}^C u_{ij} = 1 \quad \forall \quad i \dots\dots\dots (20)$$

$$0 < \sum_{i=1}^N u_{ij} < N \quad \forall \quad j \dots\dots\dots (21)$$

4.2.3 REDES NEURONALES ARTIFICIALES

Una red neuronal artificial es una aproximación no lineal, y es comparada por muchos autores y expertos con las redes neuronales que conforman el funcionamiento del cerebro humano [7]. Esta comparación se debe a que de cierta forma existen similitudes entre las redes neuronales artificiales y las redes neuronales del cerebro, como las siguientes:

- Cuentan con la capacidad de aprendizaje a partir de la experiencia. Las redes neuronales pueden ser entrenadas para realizar tareas sin la necesidad de programar todas las alternativas.
- Cuentan con una alta velocidad de respuesta una vez entrenadas.
- Cuentan con un alto grado de robustez. El aprendizaje afecta a todas las neuronas, por lo que si algún sector de la red falla, ésta puede seguir dando resultados.

Una neurona artificial es un modelo cuyos componentes tienen analogías directas con una neurona natural. En la Figura 9 podemos observar la forma básica de una neurona artificial. Las señales de entrada se representan como $x_0, x_1, x_2, \dots, x_n$. Estas señales son variables continuas y no los pulsos eléctricos discretos que ocurren en el cerebro. Cada una de estas entradas es modificada por un peso, conocido como peso sináptico, cuya función es análoga a la unión sináptica que existe entre las neuronas naturales. Dichos pesos pueden ser positivos o negativos, correspondiendo a la aceleración o inhibición del flujo de las señales eléctricas en el cerebro. La neurona artificial consiste en dos partes: la primera, suma el producto resultante entre las entradas y sus respectivos pesos; la segunda, es un filtro no lineal denominado función de activación, por la que la señal combinada fluye [20].

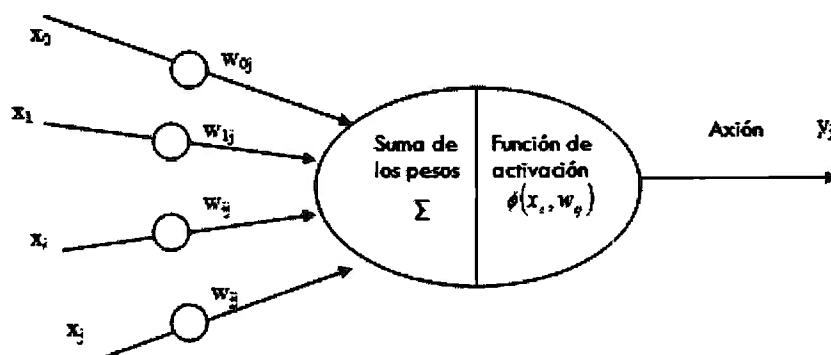


Figura 13. Representación esquemática de una neurona artificial [20].

Donde:

$I_j = \sum_{i=1}^n W_{ij} X_i \dots$ (22) es la suma de los pesos de la entrada.

$Y_j = \phi(I_j) \dots$ (23) es la función de activación.

Existen diversos tipos de funciones de activación. En la siguiente tabla se observa el tipo de funciones más comunes.

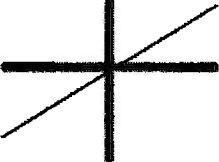

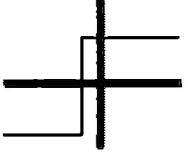
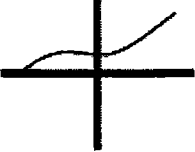
<p>Función lineal</p> <ul style="list-style-type: none"> - Tiene una zona lineal. - Se implementa fácilmente. 	<p>$f(x)=ax$ donde a es una constante</p>	
<p>Función no lineal</p> <ul style="list-style-type: none"> - Se usa para representar operaciones no lineales simples. 	<p>$f(x)=$ 1 si $x = 1$ x si $x < 1$ -1 si $x = -1$</p>	
<p>Función escalón</p> <ul style="list-style-type: none"> - Se implementa fácilmente. - Tiene un rango no lineal. - No puede imitar funciones suavemente. 	<p>$f(x)=$ 1 si $x > 0$ -1 en otro caso</p>	
<p>Función sigmoideal</p> <ul style="list-style-type: none"> - Es continuamente diferenciable. - Puede realizar decisiones difusas. - No es fácil de implementar. - La salida está limitada a valores positivos. 	<p>$f(x)= 1/(1+e^{-ax})$ donde a es un coeficiente de ajuste</p>	

Tabla 1. Tipos de funciones de activación [7].

Para entender la forma en que una red puede realizar su aprendizaje, observemos el diagrama de flujo de un programa de entrenamiento para un perceptrón (neurona conformada por 2 entradas y una salida).

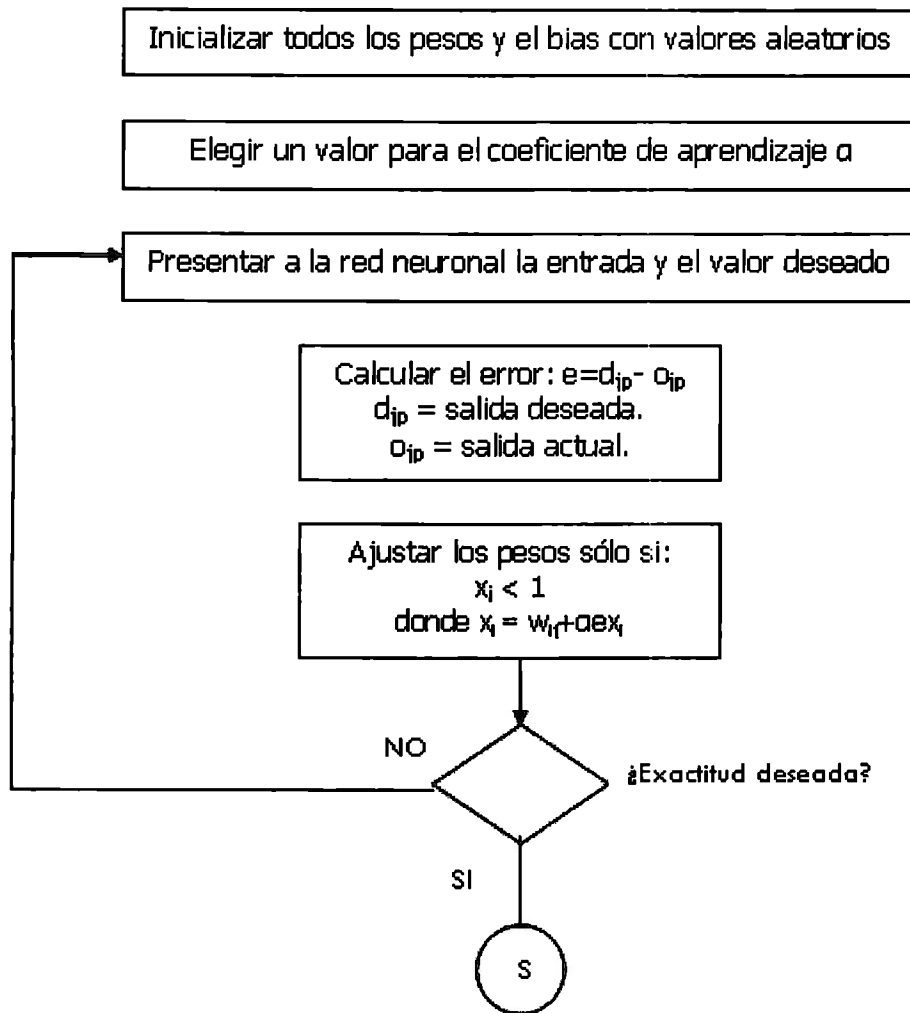


Figura 14. Diagrama de flujo para el entrenamiento de un perceptrón [7].

Entonces una red neuronal artificial está constituida por un largo número de elementos de procesamiento (neuronas artificiales) simples e interconectados entre sí, en una arquitectura inspirada por la estructura neuronal del cerebro. Estas neuronas artificiales están organizadas en capas con conexiones aleatorias o totales entre ellas. Regresando a la Figura 1, tenemos una representación de una red neuronal artificial para el caso de borurización, la cual está integrada por tres capas: una de entrada, una capa intermedia y una de salida. Cabe destacar que la capa de entrada no es una capa que compute, ya que no cuenta con pesos ni funciones de activación. La capa de salida nos dará la respuesta ante una entrada dada. Las capas intermedias son también llamadas capas ocultas ya que no tienen conexiones con elementos externos.

Durante el proceso de entrenamiento, los datos de entrada son presentados a la red y los pesos son ajustados dependiendo del algoritmo matemático que se esté utilizando. Los algoritmos más comunes son los descritos a continuación [8]:

- *Regla de Hebb*: Esta teoría está basada en la forma en que las neuronas naturales deben de aprender. Hebb utilizó la psicología y fisiología para sugerir que si dos neuronas en

ambos lados de la sinapsis disparan un pulso eléctrico, entonces la conexión se fortalece. Se le conoce como la regla Hebbiana, y es considerada como la base de muchas técnicas de aprendizaje dentro de la inteligencia artificial.

- *Regla Delta generalizada:* También conocida como la regla del mínimo error cuadrado medio (LMS por sus siglas en inglés). Esta regla consiste en modificar constantemente la fuerza de las conexiones para reducir la diferencia, o la delta, entre la salida deseada y la salida actual. La técnica de retropropagación del error es la generalización más común de la regla delta.
- *Ley de aprendizaje de Kohonen:* Se aplica solamente para aprendizajes no supervisados. Cada neurona artificial debe de competir por la oportunidad de aprender. La neurona artificial con mayor salida será la ganadora y puede tanto inhabilitar a sus competidoras como excitar a sus “vecinas”. Sólo la ganadora podrá tener una salida y sólo ella y sus vecinas podrán ajustar sus pesos. El tamaño del “vecindario” puede variar, empezando generalmente con un tamaño grande, para luego reducirlo mientras se desarrolla el aprendizaje.

El entrenamiento puede ser supervisado o no supervisado. Durante el entrenamiento supervisado, se presentan las entradas de la red así como las salidas deseadas. La presentación de esta información es continua, para que los valores de los pesos de las entradas se vayan modificando hasta que la diferencia entre la salida deseada y la salida real alcance algún valor predeterminado. Después de cada evaluación del algoritmo de aprendizaje, la diferencia entre la salida actual y la salida deseada es comparada con el valor del error permisible. Si la diferencia es igual o menor al valor del error, entonces el proceso se detiene y se considera que la red está entrenada.

Durante el entrenamiento no supervisado, la salida deseada para una serie de entradas no es presentada. La red neuronal construye modelos internos que capturan regularidades en los patrones de entrada. Es necesario proveer a la red de datos de entrenamiento que puedan cubrir la extensión del problema, y no caer en ambigüedades. El proceso de entrenamiento es repetido hasta que se alcance algún criterio específico.

La red de retropropagación del error, algunas veces llamada red de perceptrones multicapa, es uno de las estructuras de red más utilizadas. La retropropagación del error es una red multicapa en donde al menos dos de sus capas tienen pesos adaptables. La capa de entrada simplemente distribuye los valores a las capas ocultas. Los elementos de cada capa están conectados totalmente con los de capas anteriores y posteriores. En la siguiente figura podemos observar un tipo de red de retropropagación del error.

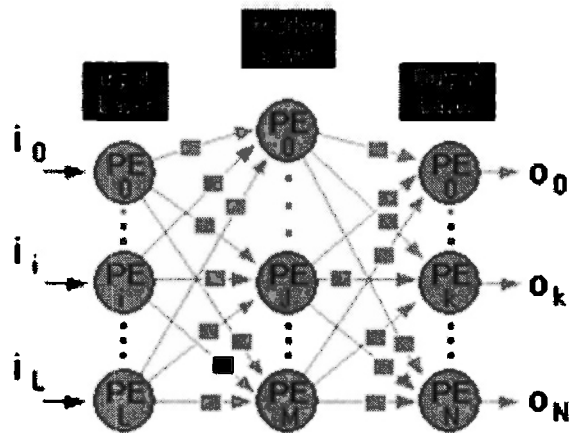


Figura 15. Diagrama de flujo para el entrenamiento de un perceptrón [8].

El entrenamiento de una red de retropropagación del error consiste en dos partes: una propagación hacia delante (también llamada fase de proceso) y una propagación hacia atrás (llamada fase de aprendizaje). Durante la fase de proceso un vector de entrada es presentada a la red. La capa de entrada simplemente pasa las entradas a la primera capa oculta. Cada neurona artificial de la capa oculta realiza la suma de los productos de las entradas por sus respectivos pesos, y pasa el resultado a la siguiente capa. Este proceso se repite hasta que la capa de salida produce un resultado. Durante la fase de aprendizaje, el error de salida es obtenido y propagado hacia atrás a través de las capas predecesoras. Los pesos se ajustan basados en el error calculado [8].

El algoritmo de una red de retropropagación del error es el siguiente:

- Se inicializan los pesos en valores aleatorios.
- Se presenta el vector de entrada.
- Se calcula la salida para cada neurona artificial de las capas ocultas con la Ecuación 24.

$$o_j = F\left(\sum_{i=0}^L w_{ij} \cdot i_i\right) \dots (24) \text{ donde } o_j \text{ es la salida de la neurona artificial de la capa oculta; } w_{ji} \text{ es}$$

el peso entre la conexión de una neurona de la capa de entrada con una neurona de la capa oculta; i_i es la entrada de la neurona de la capa de entrada.

- Se calcula la salida para cada neurona artificial de la capa de salida con la Ecuación 25.

$$o_k = F\left(\sum_{j=0}^M w_{kj} \cdot o_j\right) \dots (25) \text{ donde } o_k \text{ es la salida de la neurona artificial de la capa de salida;}$$

w_{kj} es el peso entre la conexión de una neurona de la capa oculta con una neurona de la capa de salida; o_j es la salida de la neurona de la capa oculta.

- Se calcula el error para cada neurona artificial en la capa de salida con la Ecuación 26.

$$e_k = (d_k - o_k) \cdot F'\left(\sum_{j=0}^M w_{kj} \cdot o_j\right) \dots (26) \text{ donde } e_k \text{ es el error en la capa de salida, } d_k \text{ es la salida}$$

deseada.

- Se calcula el error para cada neurona artificial de la capa oculta con la Ecuación 27.

$$e_j = F' \left(\sum_{j=0}^M w_{ji} \cdot i_i \right) \cdot \sum_{k=0}^N (e_k \cdot w_{kj}) \dots (27) \text{ donde } e_j \text{ es el error en la capa oculta.}$$

- Se actualizan los pesos tanto para la capa de salida como para la capa oculta con las siguientes dos ecuaciones.

$$w_{kj}(t+1) = w_{kj}(t) + \eta e_k o_j \dots (28) \text{ para la capa de salida.}$$

$$w_{ji}(t+1) = w_{ji}(t) + \eta e_j i_i \dots (29) \text{ para la capa oculta, donde en ambos casos } \eta \text{ es el coeficiente de aprendizaje.}$$

Cabe señalar que en ocasiones la red no converge, por lo que es necesario realizar cambios en los parámetros de diseño, como el número de capas, número de neuronas por capa, número de entradas y de salidas, conexiones entre capas, etc [8].

5 RESULTADOS

5.1 Experimentación

La borurización se realizó en un medio sólido de pasta a temperaturas entre 920° a 1000°C, en intervalos de tiempo de 2 a 6 hrs. El recubrimiento de pasta fue formado por una solución de B₄C (aproximadamente 76% peso de boro), disuelto en agua con una proporción de 5 gramos de polvo con 1 mililitro de agua. Este recubrimiento (espesor de capa) fue variando de 1 a 5 mm, cambiando así el potencial de boro existente sobre la superficie del sustrato. Se utilizó una atmósfera inerte de argón en una relación 95:5. El sustrato utilizado fue Acero AISI 1045. La forma de la muestra fue cilíndrica, y se observa a continuación.

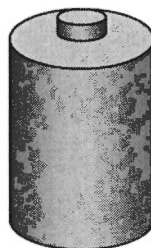


Figura 16. Forma de la muestra utilizada.

Para poder controlar el espesor de pasta del recubrimiento de las piezas cilíndricas se maquinaron moldes en acrílico en la fresadora de CNC. Las dimensiones de cada uno de los moldes fueron variando dependiendo del espesor deseado (1, 2, 3, 4 o 5mm). El molde se realizó con las siguientes especificaciones.

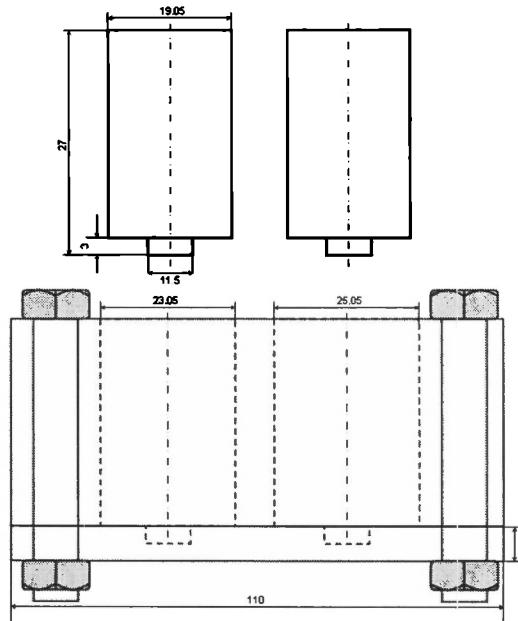


Figura 17. Molde de acrílico para controlar el espesor de pasta. Escala en mm.

La barra de acero 1045 (3/4 pulgada) fue cortada con segueta, y posteriormente torneada para que cumpliera con las especificaciones de dimensiones de acuerdo al molde de acrílico antes maquinado. Fue necesario lijar las muestras, para quitar daños externos físicos. Posteriormente se realizó la recristalización de las probetas, para normalizarlas y eliminar imperfecciones de su microestructura. La recristalización se realizó en el horno a 540°C. El mecanismo para recubrir de boro el acero, comenzó al introducir las piezas torneadas en el molde. Posteriormente se introdujo la pasta en las partes laterales de la muestra, después del tiempo necesario para que la pasta secase, y se extrajeron las piezas del molde con su recubrimiento. Finalmente se someten a un secado final a 100°C en el horno. Es necesario volver a realizar el recubrimiento y el secado, para asegurarse de eliminar los poros posibles y así garantizar la mejor difusión de boro. Posteriormente a estos pasos mencionados se realizó la borurización a temperaturas de 920, 950 y 1000°C.

Después de la borurización, se cortaron las piezas en dos pequeños cilindros, para poder hacer así la metalografía. Se utilizó baquelita para montar una de las mitades de las muestras. El montaje de la pieza en baquelita protege la zona superficial de difusión, y garantiza un lijado más plano, evitando así que la cara a observar se haga esférica. Después de realizar la metalografía se atacaron las probetas con una solución de alcohol y ácido nítrico, para poder ser observadas en el microscopio óptico y en el microscopio electrónico de barrido. Se tomaron fotografías de cada una de las probetas para poder estudiar posteriormente el crecimiento de capa borurizada. En nuestro caso, sólo se formó la capa Fe_2B ; ya que la capa FeB requiere de un alto potencial de boro, así como la influencia de elementos aleantes.

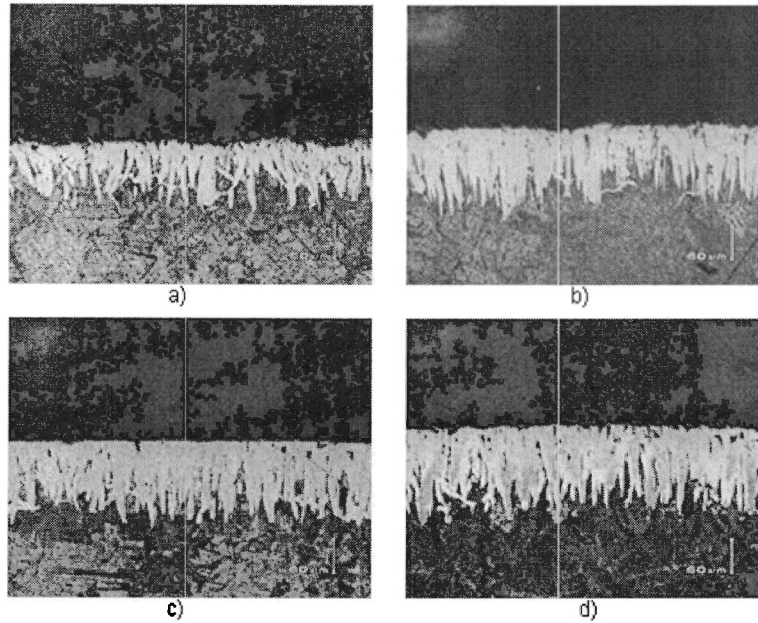


Figura 18. Acero 1045 sujeto a borurización a una temperatura de 1000°C, con 6hrs de tratamiento donde a) 2mm, b) 3mm, c) 4mm y d) 5mm de espesor de pasta de boro.

Con ayuda de software MSQ, se realizaron mediciones del espesor de capa para cada una de las probetas. Se realizaron 25 mediciones, los promedios obtenidos de éstas se muestran a continuación.

Espesor de pasta				
920°C				
Tiempo (hrs)	0hrs	2 hrs	4 hrs	6 hrs
1mm	0	0	0	38.2131
2mm	0	0	34.2543	44.7973
3mm	0	21.0900	48.2474	57.9381
4mm	0	42.0219	75.0859	97.6495
5mm	0	54.8214	81.0653	115.1890
950°C				
Tiempo (hrs)	0hrs	2 hrs	4 hrs	6 hrs
1mm	0	16.2199	19.0103	22.4605
2mm	0	37.7182	43.2440	49.2371
3mm	0	59.6564	65.1546	74.6529
4mm	0	70.5980	74.4193	98.5017
5mm	0	76.8935	94.2544	135.3401
1000°C				
Tiempo (hrs)	0hrs	2hrs	4 hrs	6 hrs
1mm	0	25.1546	40.0000	63.389
2mm	0	52.8660	76.5361	99.637
3mm	0	72.5773	96.3822	118.16
4mm	0	121.2097	140.7698	155.83
5mm	0	125.7457	165.3430	172.88

Tabla 2. Resultados experimentales de la borurización. Medida del espesor de la capa borurizada (um).

5.2 Modelo Matemático

Para la realización del modelo matemático se hacen las siguientes suposiciones:

- El movimiento entre las fases en una dimensión.
- Fe₂B empieza a formarse cuando t=0.
- Capa formada es delgada comparada con el espesor del sustrato.
- No hay efecto de porosidad en la superficie.
- El aumento de volumen por difusión de boro es despreciable.
- El crecimiento de capa es parabólico.
- El coeficiente de difusión es independiente de la concentración, se ajusta a la expresión de Arrhenius.

La ecuación de balance de masa para el crecimiento de Fe₂B, la cual es la base de la ecuación del modelo propuesto, es la que se presenta a continuación:

$$\left(C_{iFe_2B} - \beta C_0 \right) \frac{d\gamma}{dt} = D_{Fe_2B} \frac{\left(C_{eFe_2B} - C_{iFe_2B} \right)}{\gamma} \quad \dots\dots\dots (30)$$

En donde D_{Fe_2B} es el coeficiente de difusión del boro en la fase Fe₂B, β la relación del volumen específico por átomo de solvente del sustrato y la fase Fe₂B, y γ es el crecimiento de la capa borurizada. Los gradientes de concentración de boro para cada una de las interfases (interna o externa) son: $C_{eFe_2B} = 59.8 \times 10^3 \text{ mol m}^{-3}$ y $C_{iFe_2B} = (2/3) C_{eFe_2B}$.

La solución de la ecuación anterior obedece la ley de crecimiento parabólico de la forma siguiente:

$$\gamma^2 = Kt \quad \dots\dots\dots (31)$$

$$K = 2D_{Fe_2B} \frac{\left(C_{eFe_2B} - C_{iFe_2B} \right)}{\left(C_{iFe_2B} - \beta C_0 \right)} \quad \dots\dots\dots (32)$$

K es la constante de crecimiento parabólico de la fase Fe₂B.

A partir de los datos obtenidos experimentalmente, despejando la Ecuación 30 para el coeficiente de difusión y de la ecuación de Arrhenius se puede obtener una expresión para D_0 (coeficiente de difusión intrínseco), dependiendo de los diferentes espesores de pasta utilizados, así como de las diferentes temperaturas a las que se trabajó [9]. La ecuación de Arrhenius es la que se presenta a continuación:

$$D_{Fe_2B} = D_0 \exp \frac{Q}{RT} \dots\dots\dots (33)$$

en donde D_0 es el coeficiente de difusión intrínseco, Q es la energía de activación, R es la constante de los gases ideales (9.314J/mol K) y finalmente T es la temperatura en grados Kelvin.

Finalmente como presenta Campos [9], se obtiene que D_0 es lineal, y depende del espesor de pasta utilizado para realizar la borurización, de la siguiente forma:

$$D_{Fe_2B} = (3.946x - 0.006) \exp \left(-\frac{226.7kJ}{RT} \right) \left(\frac{m^2}{s} \right) \dots\dots\dots (34)$$

Las constantes de crecimiento parabólico encontradas por Campos [9], para el proceso de borurización en un medio de pasta de un acero AISI 1045, son las que se presentan en la Tabla 3.

	1193 K	1223 K	1273 K
2 mm	8.26E-14	1.19E-13	4.03E-13
3 mm	1.50E-13	2.84E-13	6.62E-13
4 mm	4.12E-13	4.24E-13	1.51E-12
5 mm	5.57E-13	7.67E-13	1.96E-12

Tabla 3. Valores de la constantes de crecimiento parabólico K para cada valor de temperatura y de espesor de pasta.

Utilizando el método de Runge-Kutta (Anexo 1) como una opción para solucionar la ecuación diferencial de balance de masa (30), se obtienen los siguientes crecimientos de la capa Fe_2B , para cada uno de los diferentes tiempos, temperaturas y espesores de pasta de boro. Las gráficas de los valores se pueden observar en el Anexo 2.

5.3 Lógica Difusa

Realizamos el modelo de sistemas difusos mediante las dos técnicas descritas anteriormente para las tres temperaturas de la experimentación. Posteriormente se utilizó el método de FCM para mejorar el sistema propuesto. A continuación presentamos los resultados.

5.3.1 Método Mamdani

El método de Mamdani se utilizó para modelar el crecimiento de capa Fe_2B con los valores de temperatura de 920°C, 950°C y 1000°C,. La estructura del sistema utilizado se muestra en la siguiente imagen, en donde se puede observar que se tomaron en cuenta dos entradas: *Tiempo* (hrs) y *Espesor de pasta* (mm):

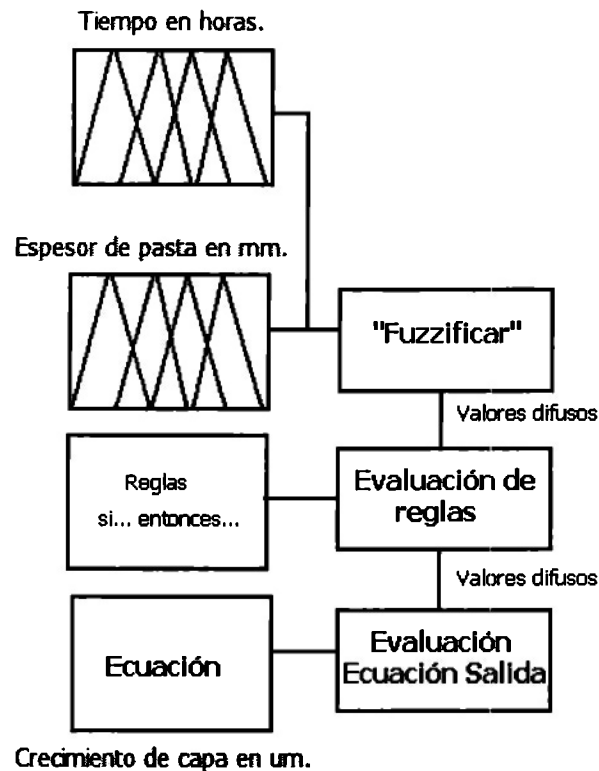


Figura 19. Diagrama Mamdani para borurización.

Las funciones de membresía para las entradas y la salida se muestran a continuación, en donde se puede observar que en el caso de la entrada de *Espesor de Pasta* existen 5 funciones de membresía. Las etiquetas corresponden como siguen:

Muy Poco – 1mm.

Poco – 2mm.

Normal – 3mm.

Mucho – 4mm.

Demasiado – 5mm.

En el caso de la entrada de *Tiempo*, se utilizan 4 funciones de membresía para cada uno de los tiempos, en donde las etiquetas corresponden a lo siguiente:

Nada – 0 hrs.

Poco – 2 hrs.

Normal – 4 hrs.

Mucho – 6 hrs.

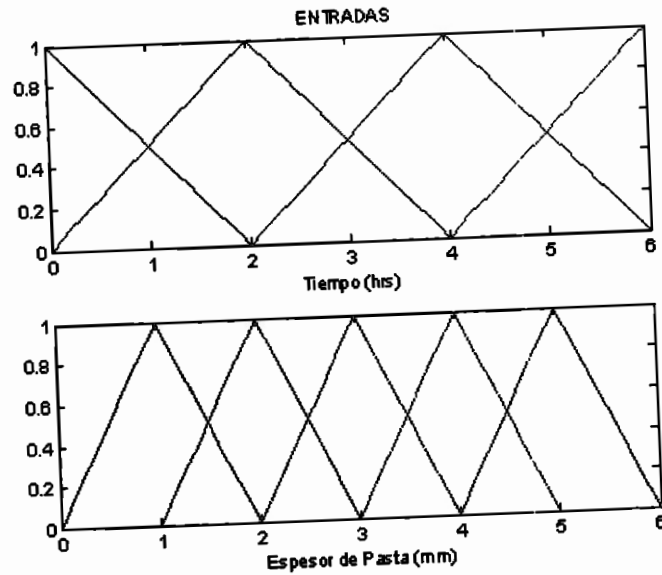


Figura 20. Entradas *Tiempo* y *Espesor de Pasta* del sistema Mamdani [6].

Finalmente para la salida se agrupo en rangos las posibles salidas existentes. Los rangos son de $5\ \mu\text{m}$ cada uno y van desde 0 hasta $180\ \mu\text{m}$, por lo tanto se obtienen 37 funciones de membresía triangulares para obtener las salidas. La asignación de los rangos y nombres para cada una de las funciones se presenta en el Anexo 3.

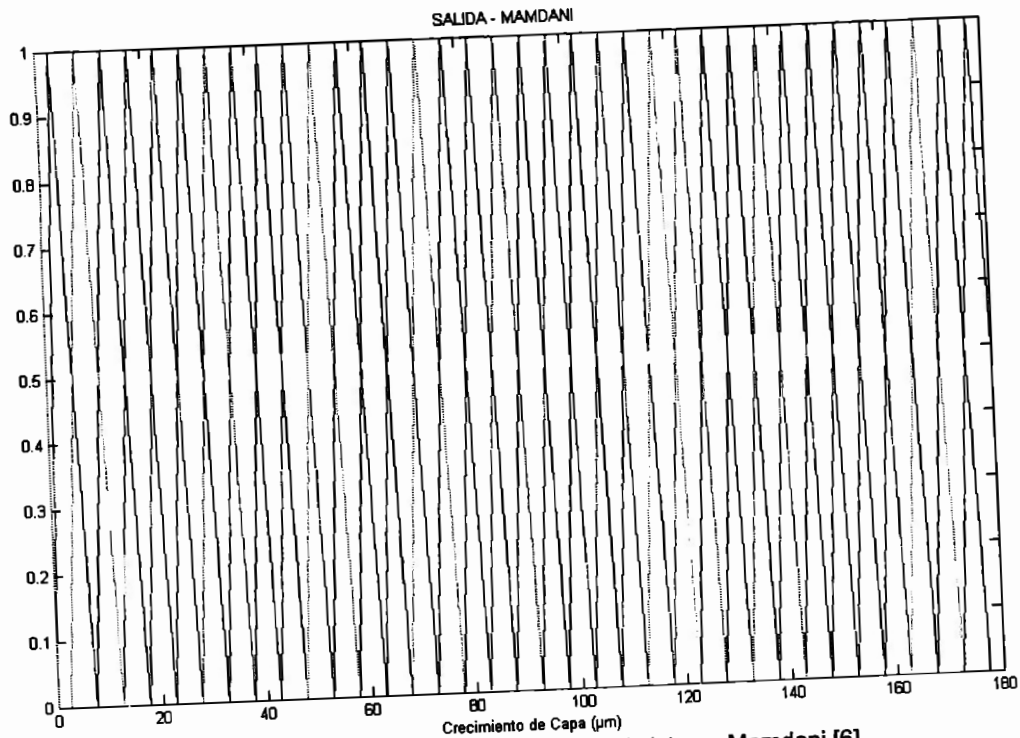


Figura 21. Salida *Crecimiento de Capa* del sistema Mamdani [6].

Después de determinar las funciones de membresía de entrada y de salida se prosiguió a dictar las reglas del modo “si... entonces...” que debe de seguir nuestro sistema. A continuación se presentan las reglas utilizadas para las distintas temperaturas en las siguientes tablas:

Espesor/Tiempo	Nada	Poco	Normal	Mucho
Muy Poco	0	0	0	8
Poco	0	0	7	9
Normal	0	5	10	12
Mucho	0	9	16	20
Demasiado	0	11	17	24

Tabla 4. Matriz de valores para 920°C.

Espesor/Tiempo	Nada	Poco	Normal	Mucho
Muy Poco	0	4	4	5
Poco	0	8	9	10
Normal	0	12	14	15
Mucho	0	15	15	20
Demasiado	0	16	19	28

Tabla 5. Matriz de valores para 950°C.

Espesor/Tiempo	Nada	Poco	Normal	Mucho
Muy Poco	0	6	8	13
Poco	0	11	16	20
Normal	0	15	20	24
Mucho	0	25	29	32
Demasiado	0	26	34	35

Tabla 6. Matriz de valores para 1000°C.

Las superficies generadas con las reglas presentadas anteriormente se muestran a continuación, para cada una de las temperaturas de trabajo. Es posible observar cómo para un tiempo de 6 horas con un espesor de pasta alto, se tiene un crecimiento de capa muy grande.

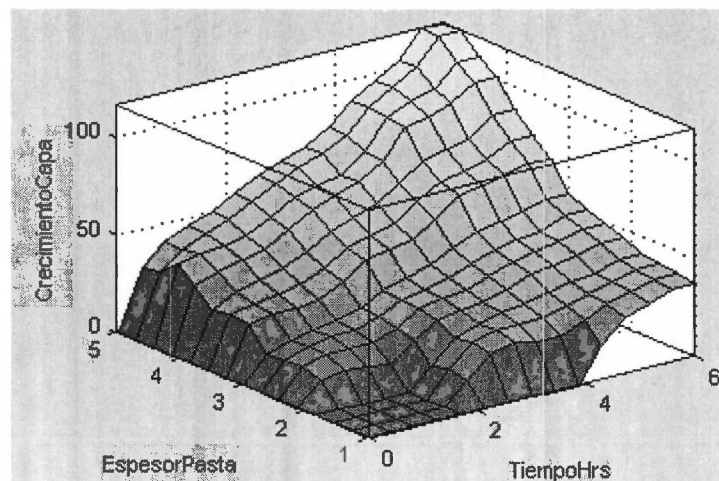


Figura 22. Superficie generada por el sistema Mamdani propuesto para 920°C, obtenida de Matlab [6].

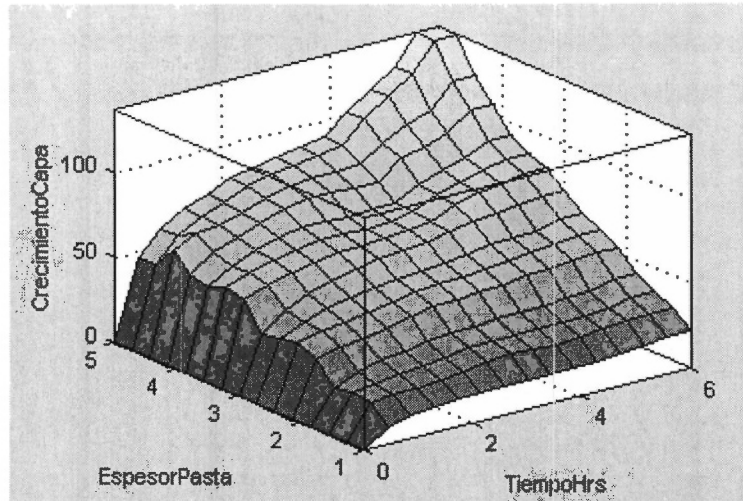


Figura 23. Superficie generada por el sistema Mamdani propuesto para 950°C, obtenida de Matlab [6].

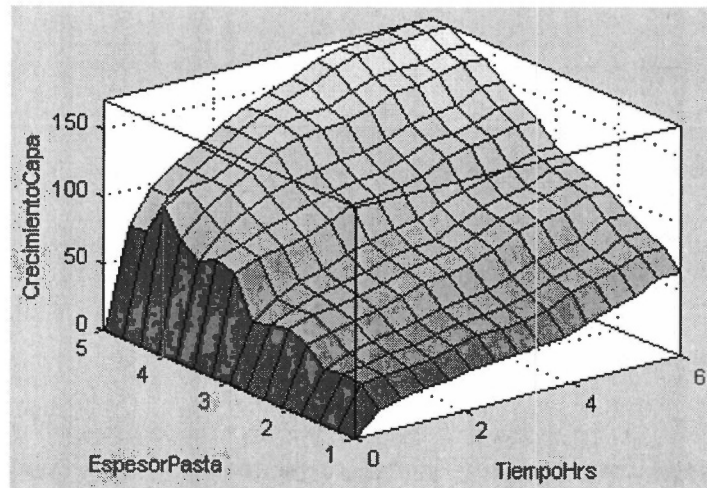


Figura 24. Superficie generada por el sistema Mamdani propuesto para 1000°C, obtenida de Matlab [6].

Finalmente los resultados obtenidos del sistema difuso para cada una de las temperaturas se observan en las siguientes tablas. Las discusiones acerca del error se realizarán en la siguiente sección.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	0	0	37.5
2mm	0	0	32.5	42.5
3mm	0	22.5	47.5	57.5
4mm	0	42.5	77.5	97.5
5mm	0	52.5	82.5	117

Tabla 7. Resultados del sistema difuso para 920°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	17.5	17.5	22.5
2mm	0	37.5	42.5	47.5
3mm	0	57.5	67.5	72.5
4mm	0	72.5	72.5	97.5
5mm	0	77.5	92.5	138

Tabla 8. Resultados del sistema difuso para 950°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	27.4	37.5	62.6
2mm	0	52.4	77.5	97.4
3mm	0	72.4	97.4	117
4mm	0	122	142	157
5mm	0	128	167	173

Tabla 9. Resultados del sistema difuso para 1000°C.

5.3.2 Método Takagi-Sugeno

De igual forma se utilizó el método de Takagi-Sugeno para modelar con la temperatura de 920°C, 950°C y 1000°C, el crecimiento de capa Fe_2B . La estructura del sistema utilizado se muestra en la siguiente imagen, en donde se puede observar su parecido con el de Mamdani. De nuevo se toman en cuenta dos entradas: *Tiempo (hrs)* y *Espesor de pasta (mm)*:

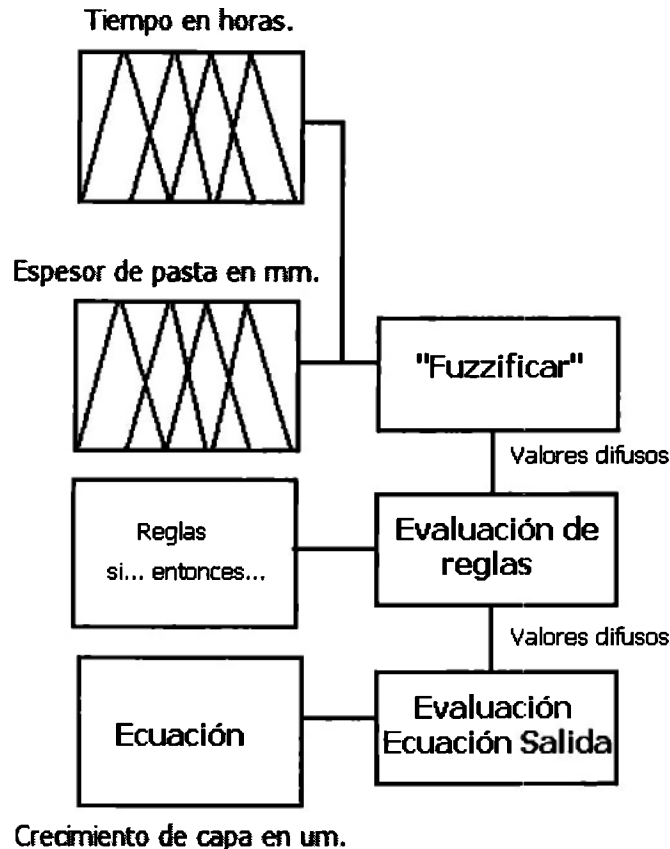


Figura 25. Diagrama Takagi-Sugeno para borurización.

Las funciones de membresía para las entradas son las mismas que utilizamos en el método Mamdani. Para la salida, se realizaron aproximaciones de diversas índoles a cada uno de los resultados experimentales con distintos espesores para las tres temperaturas. Así se obtuvo la ecuación polinomial, la cual es evaluada para obtener un valor difuso.

También las reglas tipo “si... entonces” son las mismas para las tres temperaturas con las que estamos trabajando, sólo cambian las ecuaciones polinomiales obtenidas de las aproximaciones. Con la ayuda de un software especializado en encontrar aproximaciones para una serie de datos localizado en el internet [19], encontramos las ecuaciones polinomiales adecuadas para cada uno de los casos, a distintas temperaturas, horas y espesores de pasta. Las aproximaciones utilizadas se muestran en el Anexo 4.

Como salida se obtiene una gráfica que se aproxima al comportamiento experimental. Las gráficas se pueden observar en las siguientes figuras:

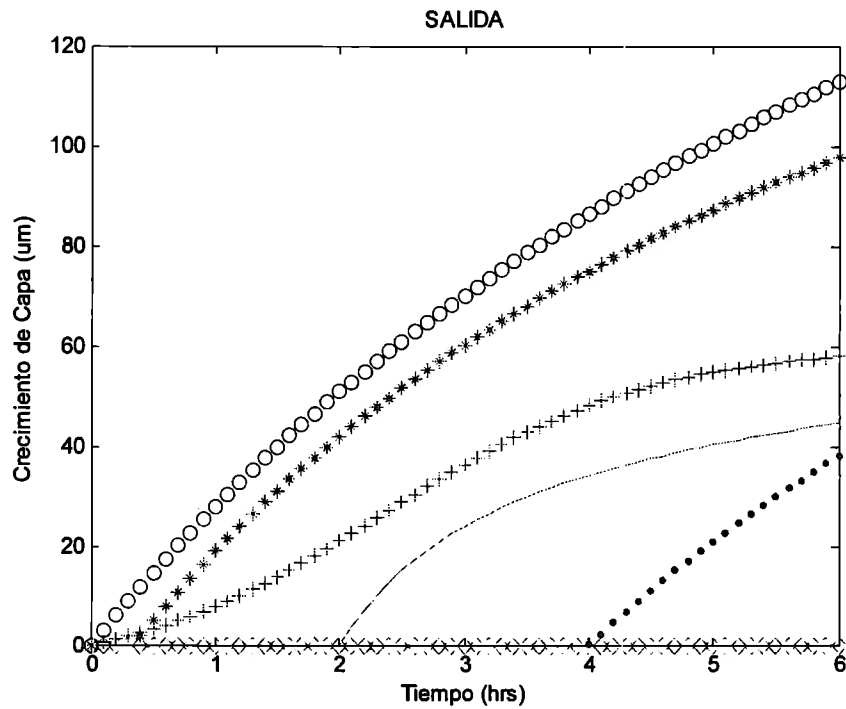


Figura 26. Gráfica de salida para 920°C mediante Sugeno [6].

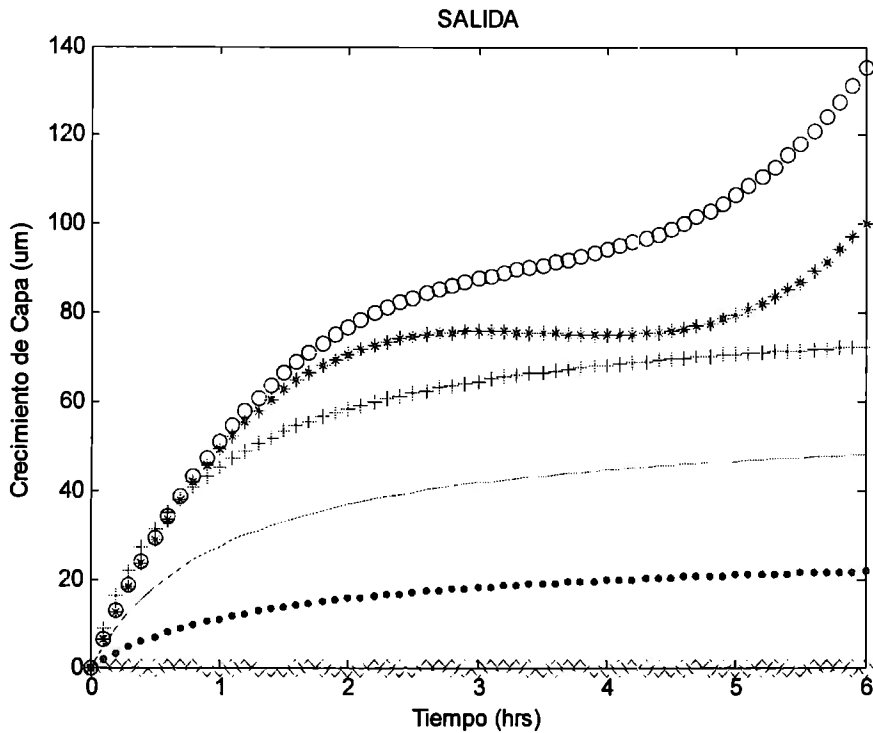


Figura 27. Gráfica de salida para 950°C mediante Sugeno [6].

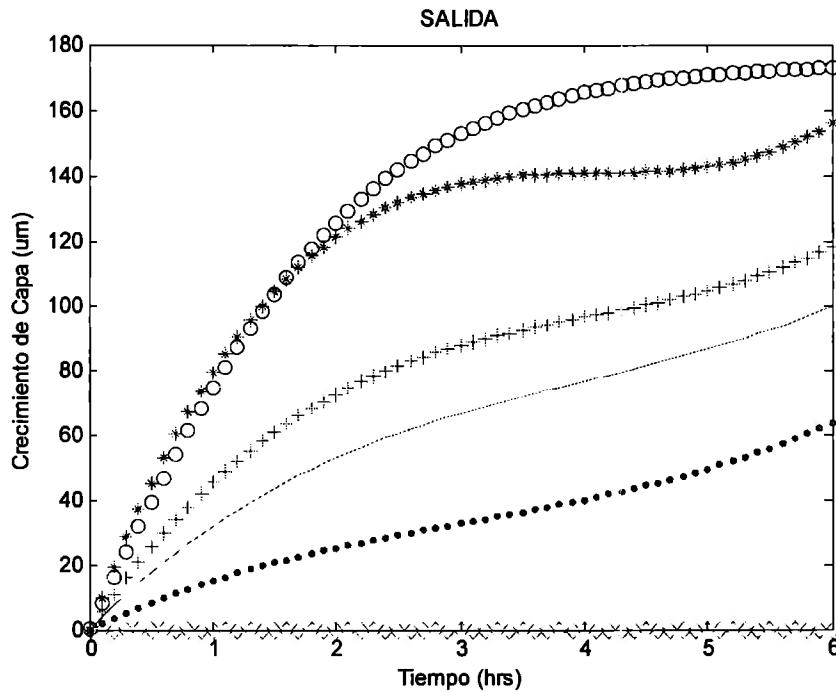


Figura 28. Gráfica de salida para 1000°C mediante Sugeno [6].

5.3.3 Método de "clusters"

Finalmente se diseñó un sistema más generalizado, con tres variables de entrada (espesor de pasta, temperatura y tiempo) para obtener el crecimiento de capa.

Para los datos de entrada se utilizaron las mismas funciones de membresía de los sistemas Mamdani y Takagi – Sugeno, y para la salida se aplicó el algoritmo FCM para obtener las nuevas formas de las funciones de membresía.

El número de “clusters” se definió en 24, generando 24 funciones de membresía de diversas formas. En la siguiente figura se aprecian las formas generadas.

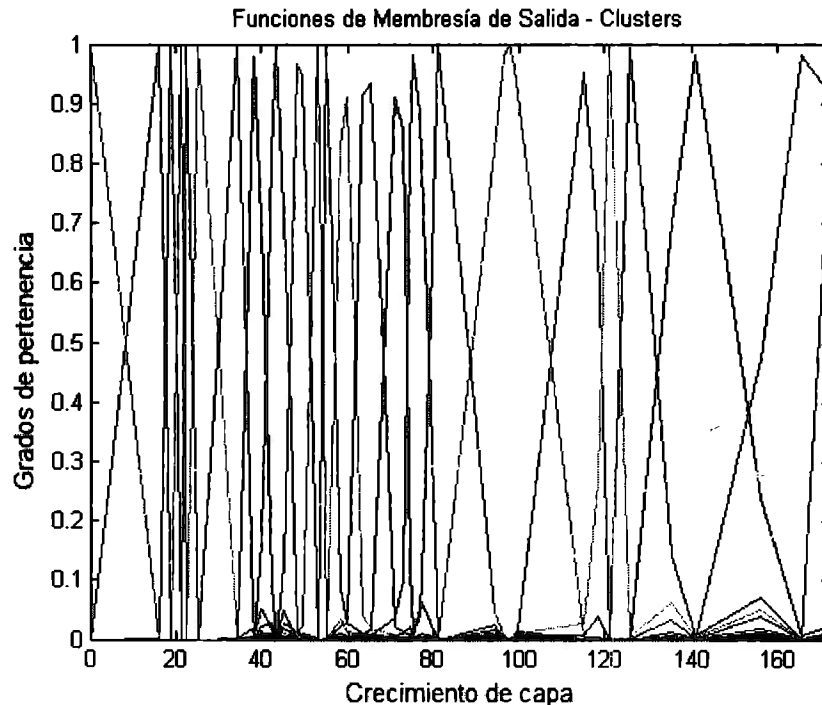


Figura 29. Formas de las funciones de membresía generadas por el método de “clusters”.

Se programó todo el sistema difuso (funciones de membresía para la entrada y la salida, reglas lingüísticas) y para la generación de la salida nítida se realizó un barrido en todas las funciones de membresía de la salida para observar cuál de ellas resultaba activa, ante una serie de datos específicos.

Los resultados generados para cada temperatura se presentan en las siguientes tablas.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	0	0	41.4697
2mm	0	0	34.2832	46.7494
3mm	0	24.3571	50.6736	62.3769
4mm	0	46.2515	79.3563	104.717
5mm	0	59.9634	86.0847	114.6837

Tabla 10. Resultados del sistema difuso para 920°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	12.5538	20.7624	25.2737
2mm	0	40.8832	46.2515	50.6736
3mm	0	61.9896	67.013	79.3563
4mm	0	74.1621	79.3563	104.7017
5mm	0	79.3563	104.7017	143.0802

Tabla 11. Resultados del sistema difuso para 950°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	28.7117	40.8834	67.013
2mm	0	56.0611	79.3563	104.7017
3mm	0	74.1621	104.7017	114.691
4mm	0	125.4508	143.0802	166.2055
5mm	0	130.9196	166.2055	173.0432

Tabla 12. Resultados del sistema difuso para 1000°C.

5.4 Redes Neuronales Artificiales

El siguiente método que se utilizó fue el de redes neuronales artificiales. Con la ayuda del Matlab se realizó un programa que simuló nuestra red neuronal. Principalmente se utilizó la topología de red presentada por Genel et al. [2], y posteriormente se redujeron el número de neuronas intermedias que presentaba la estructura. Los resultados que aquí se presentan son los obtenidos con la mayor reducción posible, buscando convergencia de la red al realizar el aprendizaje.

La propuesta de la red cuenta con tres capas. La primera capa corresponde a las entradas, donde se vaciaron los datos de espesor de pasta y de tiempo. La segunda capa corresponde a la capa intermedia, la cual se redujo a 4 neuronas. La tercera capa corresponde a la salida, donde se introdujeron los datos correspondientes a los valores de crecimiento de capa deseados. Se realizó una red para cada temperatura utilizada en la parte experimental; esto es 920°, 950° y 1000°C. Finalmente la estructura de la red propuesta es la que se observa a continuación:

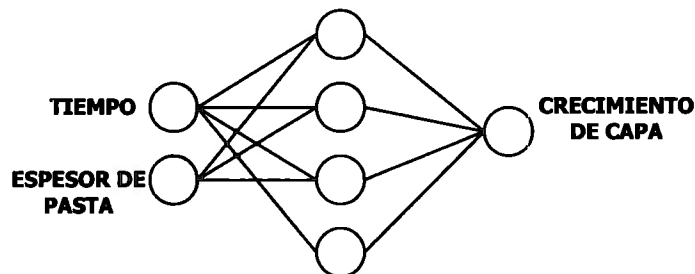


Figura 30. Estructura de red utilizada.

El método utilizado de aprendizaje para la red fue el de retro-propagación del error. En el entrenamiento de la red se buscó que el error existente entre los valores de salida que produce la red con los valores de salida reales convergiera a $1e^{-6}$. Para llegar a esta convergencia se determinó como máximo 1200 iteraciones del entrenamiento. Si la convergencia no se obtenía en menos de las

iteraciones deseadas, era necesario volver a reiniciar los valores de la red y así volver a realizar el entrenamiento.

Para introducir los datos experimentales de aprendizaje a la red fue necesario realizar normalización de los valores. Siguiendo la siguiente fórmula se normalizó:

$$x_{in}(k) = \frac{x_i(k) - \min_{j=1}^N(x_i(j))}{\max_{j=1}^N(x_i(j)) - \min_{j=1}^N(x_i(j))} \dots\dots\dots (35)$$

Los valores obtenidos se presentan a continuación:

- Tiempo

Para el tiempo: 0-6 hrs. El rango utilizado será de 0 a 1. Obteniendo los siguientes valores para cada uno de las entradas:

- 0 hrs. = 0
- 2 hrs. = 0.333
- 4 hrs. = 0.666
- 6 hrs. = 1

- Espesor de Pasta

El rango de espesor de pasta real está dado de 1 – 5 mm. El rango utilizado dentro de la red neuronal estará dado de 0 a 1; en donde 0 corresponderá a 0 mm y 1 a 5mm, se obtienen los siguientes valores:

- 1 mm = 0.2
- 2 mm = 0.4
- 3 mm = 0.6
- 4 mm = 0.8
- 5 mm = 1

- Crecimiento de capa.

Finalmente para el crecimiento de pasta obtenido se dará en mm, tomando como rango de normalización 200 – 0 μm.

Como función de activación para cada una de las capas de la red se eligió la función “logsig”, logarítmica sigmoide. La forma de la función se muestra en la siguiente figura. Está función utiliza el siguiente algoritmo para obtener la salida, en donde n es la entrada.

$$\text{log sig}(n) = \frac{1}{(1 + \exp(-n))} \dots\dots\dots(36)$$

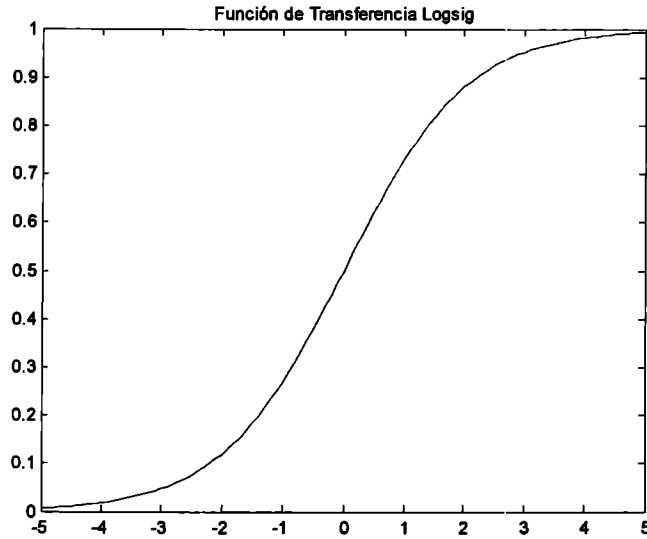


Figura 31. Función de activación [6].

Después de realizar el entrenamiento para cada una de las redes de acuerdo a las diferentes temperaturas y obtener los valores de crecimiento de capa resultantes. A continuación se presentan los valores obtenidos para las tres temperaturas de diseño.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0	0	0	38.2
2mm	0	0	34.3	44.8
3mm	0	21.1	48.2	57.9
4mm	0	42.0	75.1	97.6
5mm	0	54.8	81.1	115.2

Tabla 13. Resultados de la RNA para 920°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0.2	16.1	18.7	22.4
2mm	0.4	38.6	43.1	49.7
3mm	0.5	58.7	64.4	74.7
4mm	0.5	69.1	77.8	97.5
5mm	0.5	77.2	93.2	135.7

Tabla 14. Resultados de la RNA para 950°C.

Espesor/Tiempo	0hrs	2hrs	4hrs	6hrs
1mm	0.3	25.2	40.1	63.3
2mm	0	53	76.6	99.4
3mm	0	73.3	95.2	117.6
4mm	0	120.3	143.8	154.9
5mm	0.2	125.7	166.3	170.9

Tabla 15. Resultados de la RNA para 1000°C.

EVALUACIÓN DEL ERROR:

Ahora compararemos los resultados experimentales con los resultados del modelo matemático y con los resultados de los sistemas basados en inteligencia artificial, para comprobar el porcentaje de error que existe entre ellos y así decidir si este tipo de modelado es confiable o no.

Condensadamente, en la tabla siguiente se puede observar cada uno de los errores obtenidos para cada rango de temperatura utilizado.

	Mamdani	Sugeno	FCM	RNA	Matemático
920°C	2.8%	5.3%	4%	0.18%	13.97%
950°C	1.9%	1.8%	5.68%	0.24%	9.16%
1000°C	1.1%	0.05%	3.35%	0.67%	6.40%
Promedio	1.9%	2.4%	4.35%	0.36%	9.85%

Tabla 16. Comparación del error con todos los métodos.

Como se puede observar el modelo matemático descriptivo utilizado para comparar los resultados obtenidos de los métodos basados en inteligencia artificial, tiene un mayor error de aproximación comparado con los datos experimentales utilizados para alimentar tanto el modelo matemático descriptivo, como los modelos de inteligencia artificial. Podemos observar que los tres de los métodos utilizados obtuvieron un error menor al 3% presentado por Genel et al. [2], a excepción del método FCM que obtuvo un 4.35%. Esto puede ser debido a que el método de FCM es más general que los otros.

Finalmente se realizó la evaluación de cada uno de los modelos con la experimentación, para el caso de 5 hrs a 950°C, que es una serie de datos no utilizada para el diseño de los modelos de IA. La siguiente tabla muestra cada uno de los resultados obtenidos en cada uno de los casos, así como también los resultados experimentales.

Experimental	Mamdani	Sugeno	RNA	FCM	Matemático
20.7561	20.4	21.0567	20.8	22.5104	
44.5499	45.4	46.8124	42.4	47.6330	46.2817
71.0103	70.4	70.8005	66.2	72.6330	71.4983
81.4983	86.6	78.4868	83.2	90.4702	87.3613
115.3402	118.0	106.6215	112.4	123.2571	117.4989

Tabla 17. Evaluando en 5hrs para 950°C.

La siguiente tabla muestra el error existente entre el valor obtenido con el valor deseado (experimental) para los resultados presentados en la tabla anterior.

	Mamdani	Sugeno	RNA	FCM	Matemático
1mm	1.71%	1.44%	0.76%	8.45%	
2mm	1.91%	5.07%	4.76%	6.92%	3.93%
3mm	0.86%	0.30%	5.96%	2.28%	0.69%
4mm	6.26%	3.7%	2.16%	11%	7.24%
5mm	2.30%	7.56%	2.57%	6.86%	1.87%
PROMEDIO	2.61%	3.61%	3.26%	7.11%	3.43%

Tabla 18. Tabla de errores para 5hrs de 950°C.

CONTRIBUCIÓN AL ESTADO DEL ARTE

La principal contribución al estado del arte es la aplicación de técnicas de inteligencia artificial para el diseño de un proceso basado en la borurización, con el objetivo a futuro de que este proceso sea automatizado. La inteligencia artificial nos permite trabajar con sistemas multivariados, en donde no sea necesario un conocimiento basto sobre el proceso con el que se trabaja. En el caso de la borurización, variables como el tiempo, temperatura, espesor de pasta, crecimiento de boruros, coeficientes de difusión, microdureza, etc. es necesario conocerlas para realizar el tratamiento. Con un sistema de inteligencia artificial para la borurización se tiene la posibilidad de predecir el comportamiento de alguna variable específica. Se encontraron dos diseños específicos directamente relacionados con nuestro proyecto: el modelado de un sistema para la obtención de la dureza de un acero ADI mediante lógica difusa y una red neuronal artificial que modela la borurización, en el cual se obtiene un error promedio de 3% entre datos reales y datos obtenidos. Aunque la lógica difusa y las redes neuronales artificiales son dos métodos distintos, podemos decir que el modelo difuso Mamdani diseñado es una muy buena aproximación ya que cuenta con un error promedio de 2.6%.

CONCLUSIONES

Durante el desarrollo del proyecto, fueron utilizadas cuatro técnicas de modelado mediante inteligencia artificial. Como se ha podido observar en el presente documento, cada uno de los sistemas planteados modelan de una forma satisfactoria el proceso de borurización, lográndose obtener resultados confiables al realizar la simulación para un caso especial de borurización en el acero AISI 1045.

En el caso del método Mamdani es importante tener en cuenta que es sumamente útil para sistemas más simples. En el proceso de borurización, al aumentar las salidas el número de reglas aumenta y con esto la exactitud también.

Los resultados fueron disminuidos considerablemente al ir modificando cada uno de los métodos de lógica difusa utilizados. Es importante mencionar que el error presentado para el caso de Mamdani fue reducido desde 14% a 1.74% con la ayuda de un experto en borurización, que nos

asesoró para seleccionar las funciones de membresía de salida más adecuadas para nuestro caso. En el caso del método Tagagi-Sugeno las curvas de aproximación obtenidas y presentadas son las que más asemejaban el comportamiento parabólico que tiene el crecimiento de la capa de Fe_2B . Los resultados generados por este método también fueron reducidos a partir de una modificación que se realizó al sistema planteado en noviembre 2005. La reducción del error promedio fue de 5.9% a 1.90%, siendo este último el que se presenta en el trabajo.

El comportamiento parabólico, que se buscó seguir como referencia para plantear las funciones de salida del método Takagi-Sugeno, está dado por el modelo matemático descriptivo. Este comportamiento no puede generalizarse, ya que aumentando la temperatura y el tiempo de tratamiento el error puede llegar a minimizarse; mientras que para casos contrarios, la inexactitud del modelo es notoria y se puede observar en las tablas presentadas, con un error de hasta 12.03%.

El potencial de boro sobre la superficie es un factor importante para la formación de la capa Fe_2B . Al aumentar el potencial se generan capas más compactas y continuas así como también se alcanza de forma rápida la concentración en la superficie necesaria para que las capas comiencen a formarse. Como se puede observar el modelo matemático está limitado para obtener crecimientos de capa borurizada para potenciales de boro contenidos en pasta de espesores mayores a 2 mm. El bajo potencial de boro, presente en espesores menores a 2 mm., genera una baja concentración de boro en la superficie, que a bajas temperaturas no es suficiente para que las capas de Fe_2B se empiecen a formar.

Con ayuda de los métodos de inteligencia artificial se plantean métodos más generales que pueden ser utilizados para todos los espesores de pasta, y que no sólo se comportan adecuadamente para altos rangos de temperaturas y tiempos, sino que además disminuyen el error en la generalidad de casos a los que puede estar sometido el proceso. Esta generalización alcanza su mayor desarrollo en el caso de FCM; ya que a diferencia de los otros sistemas en donde se realizaba un sistema para cada temperatura, para el caso del sistema FCM se introdujo la temperatura como una tercera variable de entrada. Como resultado se obtuvo un sistema que abarca los rangos de todos los parámetros variables del proceso que se modificaron en la experimentación.

El método Mamdani es una opción eficiente de modelado para el proceso de borurización, en donde existe una gran disminución del error que se puede obtener; además de que el error de la simulación se ha reducido a más del 3%, propuesto por Genel et al. [2], para las diferentes temperaturas (920°C – 1.66%, 950°C – 2.14% y 1000°C – 1.41% de error).

En la Tabla 18, donde se evalúa cada uno de los modelos para un proceso de 5 horas a 950°C, se puede observar que entre el modelo difuso realizado con Mamdani y el modelado matemático descriptivo, el primero se acerca en un 80% más a los resultados que el segundo, con un error promedio del 2.61%. Con esto se concluye que el modelo a base del método Mamdani es más confiable en la mayoría de los casos. Aún cuando en el caso de 5mm la exactitud del modelo

matemático es mayor, el método Mamdani presenta un error de 2.3%, estando por lo tanto abajo del rango de error de 3%, presentado por Genel et al. [2].

Podemos asegurar que la lógica difusa, especialmente el método Mamdani, es un método alternativo eficiente para el modelado de procesos termoquímicos, borurización en nuestro caso. El modelo de lógica difusa por medio del método de Mamdani disminuye el error de aproximación considerablemente a comparación con el modelo matemático descriptivo, esto nos da una clara idea de que dependiendo del uso que se le quiere dar al modelado y de lo que se quiere saber, será el método a utilizar. Si lo que se desea es conocer con mayor precisión cuál es el espesor de la capa Fe_2B para diferentes temperaturas, tiempos y espesores, el método Mamdani de lógica difusa utilizado nos dará los mejores resultados.

En el caso del sistema basado en redes neuronales artificiales, se logró crear una red más simplificada que la presentada por Genel et al. [2], ya que la estructura propuesta cuenta con 2 neuronas de entrada, 4 neuronas intermedias y una de salida, para el caso específico de 920°C. Como se puede observar en la tabla de errores, la red neuronal artificial casi no cuenta con error al utilizar los datos de entrenamiento como entrada, ya que la red cuenta con un aprendizaje basado en los datos de entrada, por lo que “sabe” la forma en que debe responder ante ellos. El método de aprendizaje que fue utilizado fue el de retropropagación del error, con el cual se realizaron diversas pruebas ya que en ocasiones el sistema se alejaba bastante de los valores deseados o simplemente no convergía. Al presentar datos cuya característica fue la de no participar en el entrenamiento, la red entonces eleva su tasa de error hasta en un 3.3%, lo cual es ligeramente superior al error presentado por Genel et al. [2].

Una de las causas de lo elevado de este error es que probablemente la red neuronal esté sobre entrenada; es decir, que la red aprende de una forma excelente los valores con los que se le alimenta, pero al introducir un valor distinto que se quiera evaluar, no genera un resultado confiable. Como interrogante nos preguntamos si sería posible disminuir los datos de entrenamiento de la red; esto es, en vez de introducir los 20 datos para cada uno de los sistemas de cada temperatura, alimentar la red con sólo los datos más significativos. Podría ser el caso que se esté saturando a la red de información y con menos datos tendría el mismo desarrollo, y se obtendría los mismos resultados, o mejores. Al realizar la elección de topología de red se observó que para 6 neuronas en la capa intermedia, el comportamiento de la red al ser simulada para un caso en particular no mejoraba en comparación con el propuesto en el trabajo.

Finalmente el método FCM se presenta como un área de oportunidad del proyecto. Se puede observar que el error obtenido por el método es de 7.11%. Este método produce el mayor error a comparación de los otros métodos propuestos de inteligencia artificial; pero es importante reconocer que realmente los valores no pueden ser comparables, ya que como se mencionó anteriormente este caso es mucho más generalizado que los planteados anteriormente. Además de que existen

alternativas de ajuste para poder reducir aún más el error, entre ellas está modificar el número de “clusters”, hacer una revisión de las funciones de membresía propuestas, etc. Una de las ventajas de este método es que es mucho más general que los demás. Se puede plantear como una alternativa para desarrollar a futuro un sistema que pueda ser aplicable para distintos tipos de materiales, y también existe la posibilidad de generar un sistema auto-ajutable. Con esto no sólo se aplicaría a la borurización, sino que podría tener diversas aplicaciones para tratamientos termoquímicos.

Englobando las cuatro técnicas de inteligencia artificial usadas durante el desarrollo de este proyecto, tenemos dos herramientas distintas: lógica difusa y redes neuronales artificiales. Cada una cuenta con sus ventajas y sus desventajas, y estos aspectos deben de ser tomados en cuenta al momento de decidir qué camino tomar para la implementación.

La principal ventaja de la lógica difusa es que su diseño está basado en el conocimiento de un experto. En otras palabras, toda la experiencia de una persona, sobre un proceso determinado, puede ser “vacuada” en el sistema de lógica difusa, para que pueda reproducir resultados aceptables, comparados con el proceso original. Si un sistema de lógica difusa responde adecuadamente, entonces se puede seguir utilizando para nuevas variables, con la confianza de que los resultados que se obtendrán serán muy parecidos a los resultados obtenidos por el proceso original.

Es necesario tomar precauciones a la hora de diseñar el sistema difuso, ya que se puede caer en redundancias o malinterpretar la información, derivando en un sistema no apto para nuevos datos, y por ende, no confiable. La redundancia es provocada por el exceso de reglas lingüísticas, provocando que el sistema esté saturado, y al final resulte en un proceso mucho más complicado que el original. Por otro lado, cuando se diseña un sistema, se necesita de un experto y de un diseñador de sistemas. Cuando el experto es el mismo diseñador del sistema difuso, entonces está garantizado que todo el conocimiento será vaciado en el sistema; pero si el diseñador no es el experto, entonces se puede caer en malinterpretaciones y vacíos de información, provocando que el sistema difuso no responda adecuadamente a las expectativas del experto. Por ello es importante que el diseñador conozca el proceso original, y tenga un contacto estrecho con el experto, para modificar el sistema difuso en cuanto a número de reglas, número de funciones de membresía, tipo de funciones de membresía, variables de entrada y de salida, etc.

En el caso de las redes neuronales artificiales, su principal ventaja es que van “aprendiendo” cuando se les presenta la información. No es necesario conocer aspectos tan detallados del proceso original, ya que éstos se ven reflejados en los resultados, por ello cuando las variables de entrada y de salida son presentadas a la red neuronal artificial, ésta hace las inferencias y los ajustes pertinentes para reproducir de la manera más fiel posible, los resultados del proceso original. La capacidad de autoaprendizaje de la red neuronal es una gran ventaja, ya que dependiendo de los parámetros de proceso (como el número de iteraciones y el error al que se desea que converja la

red), la red es capaz de autoajustarse, modificando los pesos entre neuronas hasta que los resultados cuenten con el error deseado.

Sin embargo, si los parámetros de diseño no son los adecuados, la red puede saturarse y nunca converger. Además, no existe un método para la creación de una red neuronal artificial. El número de neuronas y el número de capas que debe tener una red para determinado proceso no está establecido por alguna metodología, sino que parte de la experiencia del diseñador, a partir del diseño de redes anteriores. Nadie garantiza que la topología utilizada para determinado sistema es la que arroja mejores resultados, ya que es posible que con una neurona o con una capa más, la red obtenga mejores resultados. Las redes neuronales un sistema de inteligencia artificial que debe ser probado numerosas veces para obtener un resultado aceptable.

Para la implementación, dos aspectos principales deben de ser tomados en cuenta: gasto computacional contra fidelidad del sistema. Entre más grande sea el sistema, mayor será el gasto computacional que requerirá para su operación. Es aquí donde se debe de evaluar qué tan conveniente es implementar un sistema mucho más grande que el otro, y que mejora el error en un bajo porcentaje, ya que la aplicación es totalmente industrial.

Es necesario realizar todas estas reflexiones, ya que la elección de qué sistema implementar se basa principalmente en ellas. Los sistemas de inteligencia artificial propuestos en este trabajo arrojan resultados aceptables, por lo que otros aspectos deben de ser tomados en cuenta para tomar la decisión final.

El uso de la inteligencia artificial en el proceso de borurización es justificable, ya que en este proceso existe la influencia de gran número de parámetros. Para poder estudiar la cinética del proceso, es necesario tomar en cuenta gran número de especificaciones, que puede llegar a complicar la realización de un modelo matemático descriptivo a detalle. En nuestro caso especial, podemos asegurar que para realizar un modelo que tome en cuenta todas las variables que afectan a la borurización, sin la necesidad de conocer las especificaciones del funcionamiento del tratamiento, la inteligencia artificial es una herramienta que proporciona resultados confiables y satisfactorios.

TRABAJO A FUTURO

A partir de los resultados obtenidos, observamos que hasta el momento el sistema más fiel es el de Mamdani, sin embargo existen muchas técnicas para el diseño de sistemas de inteligencia artificial, por lo que es posible encontrar un mejor sistema aplicando alguno de estos métodos.

Por otro lado también se puede trabajar por el diseño de un sistema mucho más general, es decir, en el cual se pueda incluir cualquier tipo de material, por ejemplo. Esto implica un diseño mucho más complejo, pero indudablemente resultará en un sistema mucho más interesante. Se propone la realización de un modelo que con sólo introducir los diferentes valores de entrada y la

matriz de salida esperada para cada combinación de entradas, el sistema realice todo lo realizado por nosotros a lo largo de este proyecto. El sistema propondría funciones de membresía de entrada, posteriormente se realizarían “clusters” para proponer funciones de membresía de salida, se propondría las reglas y se generarían resultados. Así, el sistema no estaría limitado a un solo material, o a un solo valor de temperatura, sino que se auto-propondría un modelo para cada caso particular, con las variables de entrada que le interesen al usuario.

Con esto el desarrollo de un sistema como el antes descrito, el trabajo que hemos desarrollado no quedaría limitado solamente a borurización y obsoleto para otros casos; sino que sería una de las bases fundamentales para poder proponer el mejor sistema con un área mucho mayor de aplicaciones.

REFERENCIAS BIBLIOGRÁFICAS

- [1] TSOUKALAS, Lefteri. *Fuzzy and neural approaches in engineering*. Ed. John Wiley & Sons. 1997.
- [2] GENEL K., Ozbek I., Kurf A., Bindal C. *Boriding response of AISI W1 steel and use of artificial neural network for prediction of borided layer properties*. Surface and Coatings Technology 135 (2000) 258-267.
- [3] ARAFEH, Labib; Singh, Harpreet; Putatunda, Susil. *A Neuro Fuzzy Logic Approach to Material Processing*. IEEE Transactions on system, man and cybernetics-Part C: Applications and Reviews. Vol. 29 No. 3, August 1999.
- [4] SINHA A. ASM Handbook, 4. ASM International. Materials Park, Ohio, USA. 1991. Pp. 437 – 447.
- [5] GALIBOIS A., BOUTENKO O., VOYZELLE B. *Mècanisme de formation des couches borurées sur les aciers a haut carbone – I. Technique des pates*. Acta Metallurgica Vol. 28, 1980. Pp 1753-1763.
- [6] Matlab versión 6.5, Matworks.
- [7] PONCE, Pedro. Apuntes de la clase de Sistemas Inteligentes.
- [8] Motorola's Fuzzy Logic Tutorial.
- [9] CAMPOS I., BAUTISTA O., RAMIREZ G., ISLAS M., DE LA PARRA J., ZUÑIGA P. *Dependence between the thickness of boron paste and the growth of the Fe₂B boride layer during the paste boriding process*. Applied Surface Science, 2005.
- [10] ZILL, Dennis G. *Ecuaciones diferenciales con aplicaciones de modelado*. Séptima Edición. Ed. Thomson Learning.
- [11] MATUSCHKA, A. Graf. Boronising. Carl Hanser Verlag. Munich 1980.
- [12] ASKELAND, Donald. *Ciencia e Ingeniería de los Materiales*. Internacional Thomson Editores. México, 1998.
- [13] WINSTON, Patrick H., *Inteligencia Artificial*. Addison-Wesley Iberoamericana, 3ª ed., 1994.
- [14] MALINOVA T., Malinov S., Pantev N. T.; et. Al. *Simulation of microhardness profiles for nitrocarburized surface layers by artificial neural network*. Surface and Coatings Technology 160 (2002) 38-43.
- [15] KOSKO, Bart. *Neural network and fuzzy systems: a dynamical systems approach to machine intelligence*. Ed. Prentice Hall.
- [16] FREEMAN, James. *Redes neuronales: algoritmos, aplicaciones y técnicas de Programación*. Ed. Addison-Wesley.
- [17] IGNIZIO, J.P. *Introduction to expert systems*. McGraw-Hill, 1991
- [18] LEONDES, Cornelius. *Intelligent systems: technology and applications*. Volumen I y II. Ed. CRC Press, 2000.

[19] Online Curve Fitting. <http://zunzun.com/>

[20] TSOUKALAS, Lefteri H. Fuzzy and Neural Approaches in Engineering. Wiley-Interscience Publication. 1997.

[21] DULYAKARAN Pornphan, RANGSANSER Yuttapong. Fuzzy C-Means Clustering Using Spatial Information. Noviembre, 2001.

ANEXO 1. MÉTODOS DE RUNGE-KUTTA

Para la solución del modelo matemático descriptivo propuesto se utilizó el método de Runge-Kutta. Éste es uno de los procedimientos más difundidos y a la vez más exactos para obtener soluciones aproximadas a problemas de valor inicial $y'=f(x,y)$, $y(x_0)=y_0$ sea el método de Runge-Kutta de cuarto orden. Como indica el nombre, hay métodos de Runge-Kutta de distintos órdenes, los cuales se deducen mediante desarrollos en serie de Taylor con residuo de la función $y(x)$.

Para el caso del modelo matemático para Borurización, se utilizó el método de cuarto orden, el cual se describirá a continuación.

El procedimiento de Runge-Kutta de cuarto orden consiste en determinar las constantes adecuadas para que la fórmula siguiente coincida con un polinomio de Taylor de cuarto grado.

$$y_{n+1} = y_n + ak_1 + bk_2 + ck_3 + dk_4 \quad \dots\dots\dots(A1-1)$$

Donde

$$k_1 = hf(x_n, y_n) \quad \dots\dots\dots (A1-2)$$

$$k_2 = hf(x_n + \alpha_1 h, y_n + \beta_1 k_1) \quad \dots\dots\dots (A1-3)$$

$$k_3 = hf(x_n + \alpha_2 h, y_n + \beta_2 k_1 + \beta_3 k_2) \quad \dots\dots\dots (A1-4)$$

$$k_4 = hf(x_n + \alpha_3 h, y_n + \beta_4 k_1 + \beta_5 k_2 + \beta_6 k_3) \quad \dots\dots\dots (A1-5)$$

Con lo anterior se obtienen 11 ecuaciones con 13 incógnitas. El conjunto de valores de las constantes que más se usa produce el siguiente resultado:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \dots\dots\dots (A1-6)$$

$$k_1 = hf(x_n, y_n) \quad \dots\dots\dots (A1-7)$$

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \quad \dots\dots\dots (A1-8)$$

$$k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \quad \dots\dots\dots (A1-9)$$

$$k_4 = hf(x_n + h, y_n + k_3) \quad \dots\dots\dots (A1-10)$$

Hay que notar que k_2 depende de k_1 , k_3 depende de k_2 y k_4 depende de k_3 . También, en k_2 y k_3 intervienen aproximaciones a la pendiente en el punto medio del intervalo entre x_n y x_{n+1} [10].

ANEXO 2. GRÁFICAS DE LOS RESULTADOS DEL MODELO MATEMÁTICO

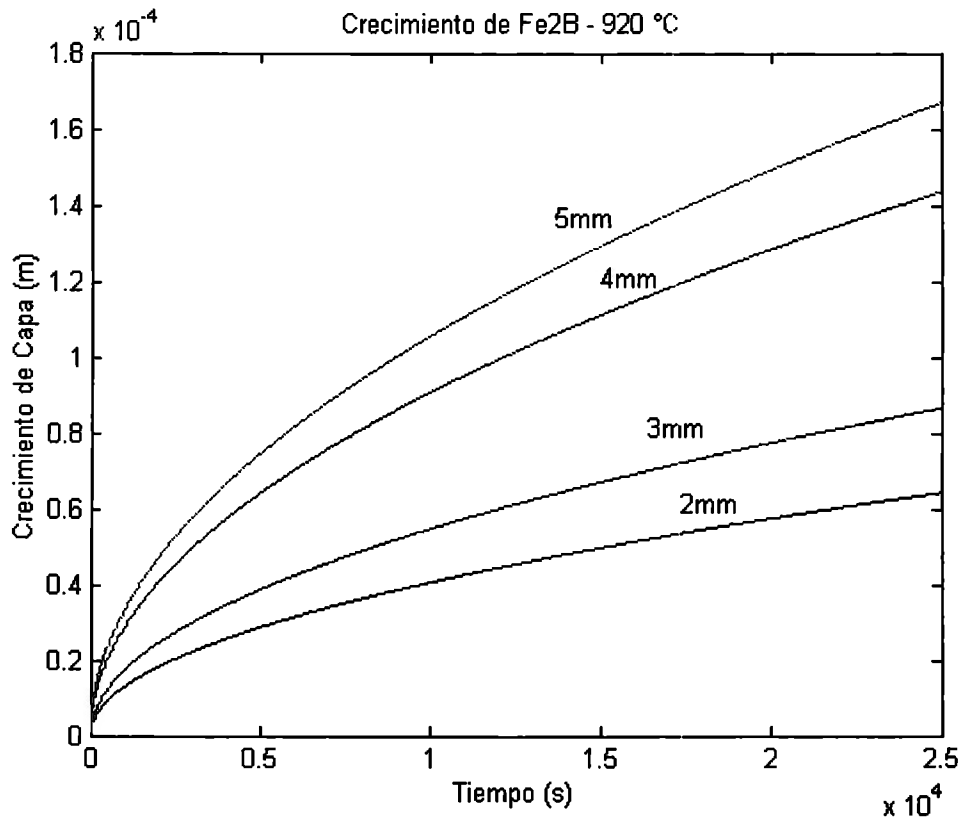


Figura A2-1. Gráfica de resultados del modelo matemático descriptivo para 920°C.

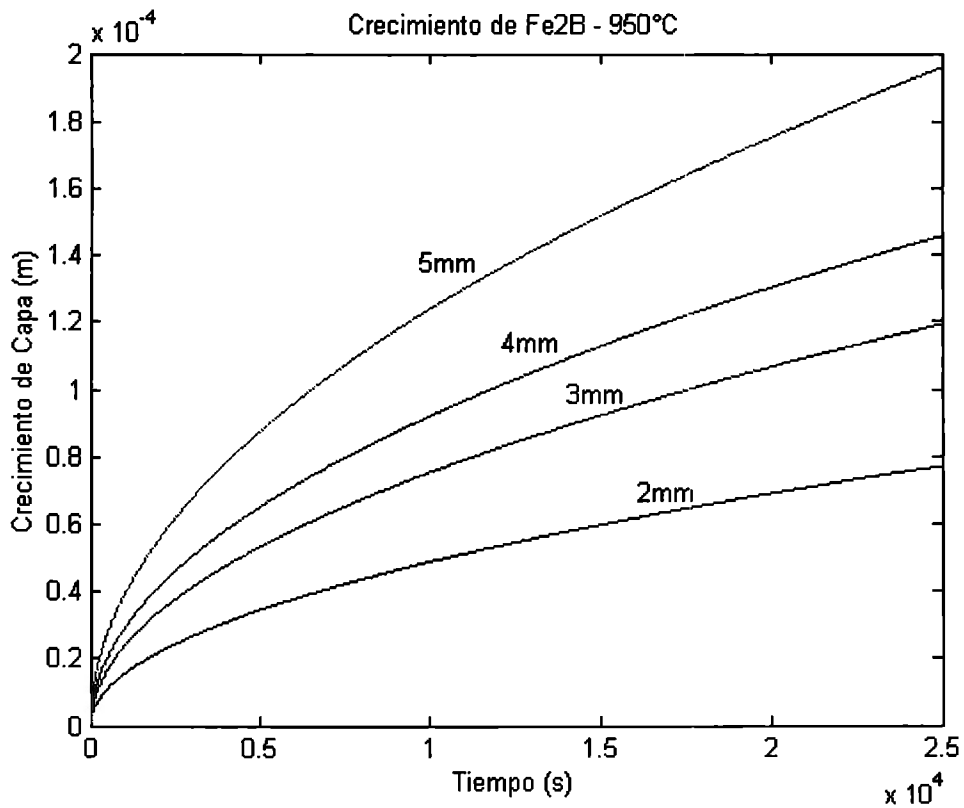


Figura A2-2. Gráfica de resultados del modelo matemático descriptivo para 950°C.

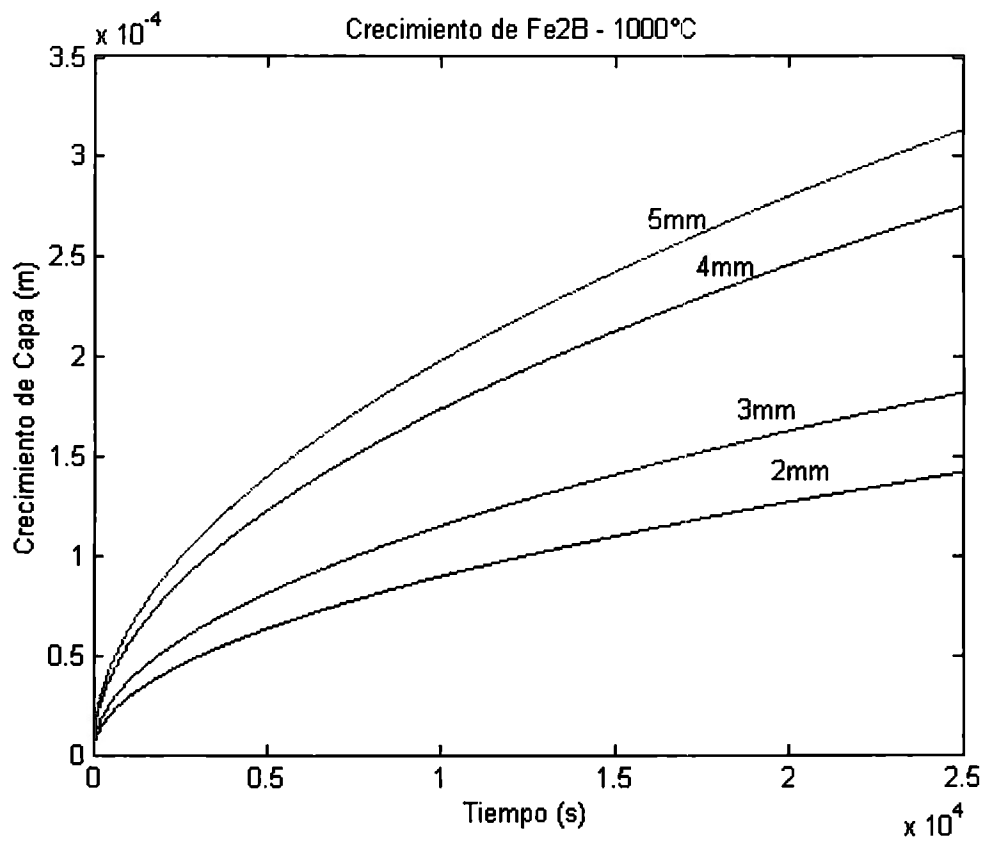


Figura A2-3. Gráfica de resultados del modelo matemático descriptivo para 1000°C.

ANEXO 3. SEPARACIÓN DE RANGOS PARA LA SALIDA DEL MÉTODO MAMDANI

Rango					Media	Función
0-5	0				2.50	1
5-10.					7.50	2
10-15.					12.50	3
15-20	16.2199	19.0103			17.50	4
20-25	22.4605	21.0900			22.50	5
25-30	25.1546				27.50	6
30-35	34.2543				32.50	7
35-40	37.7182	38.2131	40.0000		37.50	8
40-45	42.0219	44.7973	43.2440		42.50	9
45-50	49.2371	48.2474			47.50	10
50-55	54.8214	52.8660			52.50	11
55-60	57.9381	59.6564			57.50	12
60-65	63.389				62.50	13
65-70	65.1546				67.50	14
70-75	72.5773	70.5980	74.4193	74.6529	72.50	15
75-80	76.8935	76.5361	75.0859		77.50	16
80-85	81.4983				82.50	17
85-90					87.50	18
90-95	94.2544				92.50	19
95-100	97.6495	98.5017	96.3822	99.637	97.50	20
100-105					102.50	21
105-110					107.50	22
110-115					112.50	23
115-120	115.1890	118.16			117.50	24
120-125	121.2097				122.50	25
125-130	125.7457				127.50	26
130-135					132.50	27
135-140	135.3401				137.50	28
140-145	140.7698				142.50	29
145-150					147.50	30
150-155					152.50	31
155-160	155.83				157.50	32
160-165					162.50	33
165-170	165.343				167.50	34
170-175	172.88				172.50	35
175-180					177.50	36

Tabla A3-1. Rangos de valores para "defuzzificación".

ANEXO 4. APROXIMACIONES PARA EL MÉTODO TAKAGI-SUGENO

Se utilizó la herramienta "Online Curve Fitting" de la referencia [19] para realizar las aproximaciones a partir de la serie de datos experimentales y así obtener las ecuaciones polinomiales necesarias para el método de Takagi-Sugeno. Las aproximaciones de distinta índole. A continuación se presentan las ecuaciones polinomiales para cada caso.

Para 920°C

Thickness	Output Equation (um)
0 mm	$y_0 = 0$
1 mm	$y_1 = 0.8(t^{0.5}) - 1.123 \exp^4(t^{-0.5})$
2 mm	$y_2 = 5.4 \exp^{-4} t + 3.7(\ln(t)) - 1.9 \exp^9(t^{-2})$
3 mm	$y_3 = \frac{66.6}{\left(1 + e^{-\left(\frac{t-8524.4}{3832.1}\right)}\right)} - 6.5$
4 mm	$y_4 = \frac{217.2(t^2 - 1212.9t)}{(t^2 + 23743.2t + 974)}$
5 mm	$y_5 = \frac{8.6 \exp^{-3} t}{(1 + 2.9 \exp^{-5} t)}$

Para 950°C

Thickness	Output Equation (um)
0 mm	$y_0 = 0$
1 mm	$y_1 = \frac{5.3 \exp^{-3} t}{(1 + 19.7 \exp^{-5} t)}$
2 mm	$y_2 = \frac{14.9 \exp^{-3} t}{(1 + 26.5 \exp^{-5} t)}$
3 mm	$y_3 = \frac{28.2 \exp^{-3} t}{(1 + 34.3 \exp^{-5} t)}$
4 mm	$y_4 = -1.1 \exp^{-5} + 18.5 \exp^{-3} t - 1.5 \exp^{-6} t^2 + 3.9 \exp^{-11} t^3$
5 mm	$y_5 = -1.1 \exp^{-5} + 18.7 \exp^{-3} t - 1.4 \exp^{-6} t^2 + 3.7 \exp^{-11} t^3$

Para 1000°C

Thickness	Output Equation (um)
0 mm	$y_0 = 0$
1 mm	$y_1 = -3.1 \exp^{-6} + 5.1 \exp^{-3} t - 2.8 \exp^{-7} t^2 + 8.4 \exp^{-12} t^3$
2 mm	$y_2 = -6.6 \exp^{-6} + 10.7 \exp^{-3} t - 5.6 \exp^{-7} t^2 + 1.3 \exp^{-11} t^3$
3 mm	$y_3 = -9.6 \exp^{-6} + 15.6 \exp^{-3} t - 9.2 \exp^{-7} t^2 + 2.1 \exp^{-11} t^3$
4 mm	$y_4 = -1.7 \exp^{-5} + 28.4 \exp^{-3} t - 1.9 \exp^{-6} t^2 + 4.3 \exp^{-11} t^3$
5 mm	$y_5 = \frac{356.6}{\left(1 + e^{-\left(\frac{t+173.9}{3996.9}\right)}\right)} - 182.2$

ANEXO 5. ESPECIFICACIONES DE ACERO

ACERO AISI 1045

Acero fino al carbono de alta calidad, y fácil de templar. Al aumentar la dureza mediante tratamientos, se obtiene una mayor resistencia al desgaste. Al cromarse se fortalece a la resistencia a la corrosión y al desgaste, obteniendo valores de 69-71 RC de dureza. Sus componentes son los siguientes:

C: 0.43-0.50%.

Mn: 0.60-0.90%.

P: Max 0.04%,

S: Max 0.05%.

Si: 0.15-0.35%.

Aplicaciones: Utilizado en la construcción de maquinaria en general, para partes y piezas de mediana exigencia, como ejes, arboles, piezas prensadas o estampadas, chavetas y piezas expuestas al desgaste por fricción.

ACERO AISI W1

Acero tipo herramienta, de alto carbono. Se aumenta la dureza utilizando tratamientos térmicos para luego templarlo en agua. Acero de bajo costo con una aceptable resistencia al desgaste (dependiendo del contenido del carbono).

C: 0.60-1.40%

Mn: 0.10-0.40%

Si: 0.10-0.40%

Aplicaciones: Utilizado para diversas herramientas como taladros, dados, herramientas de corte, piezas de maquinaria, pernos, piñones, pivotes, cojinetes de rodillo; también para las barras de guía

y los pernos de ajuste para la fabricación de herramientas. Se utiliza también para instrumentos quirúrgicos y para la fabricación de ejes mecánicos de precisión.

HIERRO ADI (AUSTEMPERED DUCTIL IRON)

Hierro de bajo costo, flexibilidad en su diseño. Cuenta con buena maquinación, dureza, resistencia al desgaste y a la fatiga. Algunas propiedades son las siguientes:

Fuerza: Igual o mayor que el acero.

Dureza: Mejor que el hierro dúctil e igual o mejor que el acero forjado.

Densidad: Menor que la del acero.

GLOSARIO DE TÉRMINOS

Algoritmos genéticos: Técnica de la inteligencia artificial de optimización de sistemas, imitando la evolución natural de los sistemas biológicos.

Back propagation: Tipo de aprendizaje dentro de las redes neuronales artificiales, donde el error entre el valor deseado y el valor obtenido en la capa de salida, es propagado hacia atrás para disminuirlo durante el aprendizaje de la red.

Defuzzificar: Acción de pasar un valor difuso a un valor nítido.

Fuzzificar: Acción de pasar un valor nítido a un valor difuso.

Lógica difusa: Técnica de la inteligencia artificial que permite manejar información vaga o de difícil especificación para gobernar un sistema por medio de reglas de sentido común, definidas a partir de un experto, las cuales se refieren a cantidades indefinidas.

Minimización del error cuadrado: Regla basada en la búsqueda del mínimo de una expresión del error entre la salida deseada y la salida real, para redes neuronales artificiales.

Redes neuronales artificiales: Técnica de la inteligencia artificial basada en la sumatoria de funciones no lineales, donde la red "aprende" mediante la presentación de información fundamental y la obtención de los pesos entre sus conexiones, para la obtención de un resultado deseado. Varios autores manejan que están basadas en el sistema neuronal humano.

Sistemas expertos: Técnica de la inteligencia artificial basada en la metodología que sigue un especialista al resolver un sistema propio de su disciplina a partir de conocimientos teóricos y empíricos.

Tratamiento termoquímico: Tratamiento superficial de determinado material en donde se lleva a cabo una reacción química.

Valor difuso: Valor de una variable definida por una etiqueta lingüística de acuerdo a la lógica humana. Ejemplo: temperatura=muy caliente.

Valor nítido: Valor de una variable definida por un número concreto de acuerdo a las mediciones científicas. Ejemplo: temperatura=49°C.