



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO DE INSTRUMENTO VIRTUAL PARA EL BALANCEO DINÁMICO DE
LOS ROTORES DE MOTORES DE INDUCCIÓN BASADO EN UN SISTEMA
DIFUSO”**

TESIS DE POSTGRADO
Previo a la obtención del Título de:

MAGISTER EN AUTOMATIZACION Y CONTROL INDUSTRIAL

Presentada por:
ALEX CAZCO ARÍZAGA
Guayaquil – Ecuador

AÑO
2012

AGRADECIMIENTO

A la Escuela Superior Politécnica del Litoral y a todas las personas que de uno u otro modo colaboraron en la realización de este trabajo, especialmente al MSc. Arístides Reyes Bacardí por su total apoyo.

A mi familia por su ejemplo de vida.

Y por encima de todo a Dios.

Alex Cazco Arízaga

DEDICATORIA

A mi hijo Alex, mi esposa
Romira y a aquellos seres
que me dieron la vida, a
quienes siempre dedicaré
todos mis triunfos.

TRIBUNAL DE GRADUACIÓN

Dr. Boris Vintimilla Burgos

Presidente del Tribunal

MSc. Arístides Reyes B.

Director de Tesis

MSc. Efrén Herrera M.

Miembro Principal

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

Alex Cazco Arízaga

RESUMEN

El desbalance se puede definir de manera simple, como una distribución no uniforme de la masa de un rotor alrededor de su eje axial o de rotación, lo cual produce una desviación del centro de masa y/o del eje principal de inercia con respecto a su eje de rotación.

Una de las razones de importancia, por las cuales un rotor debe ser balanceado es que las fuerzas creadas por el desbalance son dañinas para la vida de la máquina.

El desbalance de un rotor no se puede determinar y corregir directamente. Esto significa que en la práctica, no es posible determinar la cantidad y ubicación del desbalance de un rotor para corregirlo en su origen. Pero sí es posible determinar su efecto, y calcular la cantidad y ubicación de las masas necesarias a agregar o quitar para reducir o controlar dicho efecto a niveles aceptables o tolerables. El problema es ¿cómo determinar la cantidad y ubicación de las masas de corrección?. Este es el problema científico que aborda esta Tesis.

En la presente tesis se trata sobre las máquinas balanceadoras MC-5 marca Hoffman, las cuales poseen un software al cual no puede acceder el cliente, cuando la misma requiera de mantenimiento programático.

Considerando esta problemática de diseño, queda abierta la investigación del método de cálculo de los valores de los parámetros de salida de la máquina de balanceo dinámico a partir de los datos recopilados durante el balanceo en taller.

Las nuevas técnicas de entrenamiento de sistemas difusos o neuronales o su conjunción, neurodifuso, han demostrado ser herramientas idóneas para aplicaciones similares.

Es por esto que el tema propuesto tiene como soporte las técnicas de los sistemas difusos, con el objetivo de disminuir el esfuerzo en la programación, donde será necesario definir el sistema (funciones de pertenencia, ubicación de las mismas, conjunto de reglas, tipo de implicación a emplear, etc.) y el método más adecuado para el entrenamiento dirigido a determinar los valores requeridos en el balanceo dinámico.

Es así que empleando la lógica difusa como técnica de modelado, y partiendo de los datos de experimentos se pretende desarrollar un sistema difuso, el cual es el soporte principal para el diseño de una aplicación de los logros alcanzados, como un instrumento virtual de balanceo dinámico de rotores de los motores eléctricos de inducción.

ÍNDICE

RESUMEN	VI
INDICE	VIII
INDICE DE FIGURAS	X
INDICE DE TABLAS	XIII
NOMENCLATURA	XIV
INTRODUCCION	15

CAPÍTULO I. ESTADO DEL ARTE DEL BALANCEO DINÁMICO DE MÁQUINAS ROTATORIAS.

1.1 INTRODUCCION	18
1.2 CLASIFICACIÓN DE LAS MÁQUINAS DE BALANCEO DINÁMICO Y SUS PROBLEMAS TRABAJO	24
1.3 ESTADO DEL ARTE EN EL BALANCEO DINÁMICO DE ROTORES DE LAS MÁQUINAS DE INDUCCIÓN	27
1.4 LOS SISTEMAS DIFUSOS COMO CALCULADORES DE PARÁMETROS	38

CAPÍTULO II. MODELADO DIFUSO DEL CALCULADOR DE PARÁMETROS

2.1 IDENTIFICACIÓN DEL MODELO DIFUSO	54
2.2 DETERMINACIÓN DE LAS FUNCIONES DE PERTENENCIA EN EL ANTECEDENTE Y EN EL CONSECUENTE, LA UBICACIÓN DE LOS CENTROS DE LAS FUNCIONES DE PERTENENCIA, DE LAS OPERACIONES DIFUSAS	86
2.3 ENTRENAMIENTO DEL SISTEMA DIFUSO CON LA BASE DE DATOS DE LA MÁQUINA MC-5	88
2.4 VALIDACIÓN DEL MODELO	90

CAPÍTULO III. DISEÑO DEL INSTRUMENTO VIRTUAL

3.1 GENERALIDADES DEL DISEÑO DEL INSTRUMENTO VIRTUAL CON LABVIEW DE NATIONAL INSTRUMENT EN SU VARIANTE DIFUSA	122
3.2 DISEÑO DE LOS BLOQUES Y SUBBLOQUES DE INSTRUMENTOS VIRTUALES REQUERIDOS PARA EL SISTEMA DIFUSO	126
3.3 DISEÑO DE LOS BLOQUES DE INSTRUMENTOS VIRTUALES DE ADQUISICIÓN DE DATOS Y MANEJO DE PUERTOS	133
3.4 DISEÑO DE BLOQUES DE INSTRUMENTOS VIRTUALES GRÁFICOS EN COORDENADAS POLARES	138
3.5 DISEÑO DE PRUEBAS	141

CAPÍTULO IV. ANÁLISIS DE LOS RESULTADOS EXPERIMENTALES

4.1 ANÁLISIS DE LOS RESULTADOS EXPERIMENTALES CON EL BALANCEADOR DIFUSO	143
4.2 RESULTADOS DE LAS PRUEBAS	153

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES	155
5.2 RECOMENDACIONES	157

BIBLIOGRAFÍA	158
ANEXOS	163

ÍNDICE DE FIGURAS

FIGURA 1.1: CLASIFICACIÓN DE LOS MÉTODOS ACTUALES DE BALANCEO DE ROTORES	27
FIGURA 1.2: FUNCIÓN QUE MAPEA CON TRES PARES DE DATOS DE ENTRADA SALIDA CONOCIDOS	48

FIGURA 2.1: VECTORES DE FUERZAS DE DESBALANCE Y APLICADOS SOBRE LOS APOYOS DE LOS PEDESTALES DE LA MÁQUINA DE BALANCEO DINÁMICO	60
FIGURA 2.2: RESULTADO DE LA SIMULACIÓN DE UNA CORRIDA CON UN ROTOR DE DIMENSIONES DADAS	61
FIGURA 2.3: RESULTADOS DEL RESTO DE LAS SIMULACIONES SEGÚN ORDEN DE LAS COLUMNAS “PRUEBA”	62
FIGURA 2.4: RESULTADOS DE LA PRUEBA 6 DE LA TABLA 2.3	64
FIGURA 2.5: RESULTADOS DE LAS PRUEBAS RESTANTES DE LA TABLA 2.2 (PRUEBAS 7, 8, 9 Y 10)	65
FIGURA 2.6: REPRESENTACIÓN DEL ÁNGULO 1 EN FUNCIÓN DE DOS VARIABLES, AMPLITUD DE VIBRACIÓN Y DESPLAZAMIENTO DEL MÁXIMO	67
FIGURA 2.7: REPRESENTACIÓN DEL PESO 1 EN FUNCIÓN DE DOS VARIABLES, AMPLITUD DE VIBRACIÓN Y DESPLAZAMIENTO DEL MÁXIMO	68
FIGURA 2.8: REPRESENTACIÓN DEL “PESO 2” EN FUNCIÓN DE DOS VARIABLES, AMPLITUD DE VIBRACIÓN Y DESPLAZAMIENTO DEL MÁXIMO	70
FIGURA 2.9: REPRESENTACIÓN DEL “ÁNGULO 2” EN FUNCIÓN DE DOS VARIABLES, AMPLITUD DE VIBRACIÓN Y DESPLAZAMIENTO DEL MÁXIMO	70
FIGURA 2.10: DIAGRAMA DE FLUJO DE ENTRENAMIENTO Y VALIDACIÓN	84

FIGURA 2.11: CÁLCULO DEL PESO CON 3X2 REGLAS	93
FIGURA 2.12: CÁLCULO DEL PESO CON 5X2 REGLAS	94
FIGURA 2.13: REPRESENTACIÓN EN 3D DEL CALCULO DEL PESO CON DOS ENTRADAS DE LA LÓGICA DIFUSA	95
FIGURA 2.14: CÁLCULO DEL PESO CON 7X2 REGLAS	95
FIGURA 2.15: CÁLCULO DEL PESO CON 9X2 REGLAS	96
FIGURA 2.16: CÁLCULO DEL PESO CON 9X3 REGLAS	97
FIGURA 2.17: CÁLCULO DEL ÁNGULO CON 3X7 REGLAS	100
FIGURA 2.18: CÁLCULO DEL ÁNGULO CON 3X7 REGLAS CON MEJOR ELECCIÓN DE VALORES DE INICIALIZACIÓN	101
FIGURA 2.19: EVOLUCIÓN DE LA VARIANZA CON EL NÚMERO DE ITERACIONES	105
FIGURA 2.20: DESVIACIÓN ESTÁNDAR PARA CADA ESTRUCTURA DE LÓGICA DIFUSA VALIDADA	108
FIGURA 2.21: SISTEMA DE LÓGICAS DIFUSAS PROPUESTO	110
FIGURA 3.1: PANEL FRONTAL EN LABVIEW	112
FIGURA 3.2: DIAGRAMA DE BLOQUES EN LABVIEW	113
FIGURA 3.3: DIAGRAMA DE BLOQUE MEDICIÓN DE AMPLITUD Y NIVEL	114
FIGURA 3.4: SEÑALES DEL SENSOR DE VIBRACIONES Y DEL SENSOR ESTROBOSCÓPICO DE LA MAQUINA DE BALANCEO	117

DINÁMICO	
FIGURA 3.5: DETECTOR DE PICO POR UMBRAL	119
FIGURA 3.6: SEÑAL DE MÚLTIPLES TONOS DESPUÉS DE SER EXAMINADA POR EL VI DETECTOR DE PICOS POR UMBRAL	121
FIGURA 3.7: VI DETECTOR DE PICO Y LAS ENTRADAS Y SALIDA DE VI	122
FIGURA 3.8: DISEÑADOR DEL SISTEMA PARA CREAR LAS VARIABLES LINGÜÍSTICAS	128
FIGURA 3.9: DETECTOR DE PICO DE FORMA DE ONDA	134
FIGURA 3.10: SELECCIÓN DE LA FORMA DE ONDA PROPUESTA	137
FIGURA 3.11: DIAGRAMA DE BLOQUE ELEMENTO GRAFICO PROPUESTO	138
FIGURA 4.1: SECCIÓN DE MEDICIÓN DE AMPLITUD DE LA VIBRACIÓN	144
FIGURA 4.2: PANEL FRONTAL DE MEDICIÓN DE LA AMPLITUD DE LA VIBRACIÓN PARA UN PLANO DE BALANCEO	145
FIGURA 4.3: MEDICIÓN DEL DESPLAZAMIENTO DEL MÁXIMO DE LA SEÑAL DE LOS SENSORES DE VIBRACIÓN	146
FIGURA 4.4: RELACIÓN ENTRE LA MEDICIÓN DE LA FASE DEL TONO SIMPLE EN GRADOS, Y EL DESPLAZAMIENTO DEL MÁXIMO EN GRADOS	148
FIGURA 4.5: MEDICIÓN DEL DESPLAZAMIENTO DEL MÁXIMO	149
FIGURA 4.6: PANEL FRONTAL DEL GRÁFICO POLAR	151

FIGURA 4.7: DIAGRAMA EN BLOQUES DEL INDICADOR DE ÁNGULO PROPUESTO PARA LA MÁQUINA DE BALANCEO DINÁMICO	151
FIGURA 4.8: VI DEL CONTADOR DIFUSO DEL TIPO MÚLTIPLES ENTRADAS UNA SALIDA (MISO)	152
FIGURA 4.9: INSTRUMENTO VIRTUAL PARA CARGAR LA LÓGICA DIFUSA	153

ÍNDICE DE TABLAS

TABLA 1.1: CRITERIO PRACTICO DEL TIPO DE BALANCEO	22
TABLA 2.1: DATOS DE EXPERIMENTOS	59
TABLA 2.2: RELACIÓN DE VALORES EMPLEADOS PARA LA INTERPOLACIÓN DE DATOS EN LAS PRUEBAS 1-5	61
TABLA 2.3: RELACIÓN DE VALORES EMPLEADOS PARA LA INTERPOLACIÓN DE DATOS EN LAS PRUEBAS 6-10	63
TABLA 2.4: RELACIÓN DE VALORES EMPLEADOS PARA LA INTERPOLACIÓN DE DATOS EN LAS PRUEBAS 11-15	66
TABLA 2.5: BASE DE DATOS	71
TABLA 2.6: CENTROS, ANCHOS Y VALORES CONSTANTES DEL CONSECUENTE PARA EL SIST. DIFUSO DE ESTIMACIÓN DE LOS	98

PESOS	
TABLA 2.7: CENTROS, ANCHOS Y VALORES CONSTANTES DEL CONSECUENTE PARA EL SISTEMA DIFUSO DE ESTIMACIÓN DE LOS ÁNGULOS DE INTRODUCCIÓN DEL PESO	100
TABLA 2.8: VALORES DE LAS DESVIACIONES ESTÁNDAR POR ESTRUCTURA DE LA LÓGICA DIFUSA VALIDADA	107
TABLA 2.9: ERRORES DE LA LÓGICA DIFUSA	107
TABLA 3.1: OPCIONES DEL BLOQUE “AMPLITUDE AND LEVEL MEASUREMENTS”	114
TABLA 3.2: ENTRADAS DEL BLOQUE “AMPLITUDE AND LEVEL MEASUREMENTS”	115
TABLA 3.3: SALIDAS BLOQUE “AMPLITUDE AND LEVEL MEASUREMENTS”	116

NOMENCLATURA

IEEE : The Institute of Electrical and Electronics Engineers

F : Amplitud de la vibración

Φ : Desplazamiento

Θ : Angulo en donde introducir el peso

P : Peso a introducir

U : Fuerzas de desbalance

A : Soporte del pedestal A

B : Soporte del pedestal B

$\mu_i(x)$: Sistema clásico difuso

c : Centro

σ : Ancho

b : Consecuente

x : Entradas

y : Respuesta de la lógica difusa de TS

INTRODUCCIÓN:

Las máquinas de balanceo dinámico de los rotores de los motores de inducción, tales como: MC-5, YYQ-160, YYW-1000, PHW-10000 ó la máquina HQ-1000, son máquinas de fabricación foráneas y contienen software a los cuales no puede acceder el cliente cuando el mismo requiera de mantenimiento programático.

Por otro lado, el diseño del software debe contemplar no solo el cálculo del peso a introducir y su ubicación sino que también debe proporcionar campos para la introducción de datos de los rotores a prueba y gráficos para la interacción con el usuario (un sistema HMI) por lo que un nuevo soporte programático debe cumplir con estas exigencias, además de ser amigable.

Las nuevas técnicas de entrenamiento de sistemas difusos o neuronales o su conjunción, neuro-difuso, han demostrado ser herramientas idóneas para aplicaciones similares.

Es por esto que el presente trabajo tiene como objetivo general el diseño de un instrumento virtual para el balanceo de rotores con base en un sistema difuso entrenado para proporcionar los datos requeridos para el balanceo dinámico de los rotores de motores de inducción, a partir de la elección de datos para la identificación

del modelo difuso, donde será necesario definir el sistema y el método más adecuado para el entrenamiento, dirigido a determinar los valores requeridos en el balanceo dinámico.

Así, empleando la lógica difusa como técnica de modelado y partiendo de los datos de experimentos se pretende desarrollar un sistema difuso, el cual es el soporte principal para el diseño del instrumento virtual de balanceo dinámico de rotores de motores eléctricos de inducción.

La estructura de la presente tesis es la siguiente: Introducción, cuatro capítulos, conclusiones, recomendaciones, anexos y referencias bibliográficas.

En el primer capítulo se resume el estudio de los artículos publicados referentes al balanceo de máquinas rotatorias en general y al balanceo dinámico de rotores de las máquinas de inducción, se presenta el enfoque de lógica difusa como calculador de parámetros.

En el segundo capítulo se selecciona el modelo difuso para lo cual se eligen el tipo de función de pertenencia en el antecedente y en el consecuente, la ubicación de los centros de estas funciones de pertenencia, se definen las operaciones difusas. Se desarrolla el entrenamiento del sistema difuso a partir de la base de datos que proporciona la máquina MC-5 de la firma Hofmann. Se valida el modelo difuso.

En el tercer capítulo diseña el instrumento virtual empleando el programa LabView en su variante difusa para el balanceo dinámico de rotores.

En el cuarto capítulo se resume el trabajo de investigación realizado, se dan los resultados de los experimentos de verificación del instrumento diseñado. Se exponen y discuten los resultados prácticos. Se dan las conclusiones y las propuestas que emanen del desarrollo de la tesis.

La hipótesis principal es: si los valores de los pesos y los ángulos de introducción de los mismos en los rotores a balancear resultan una función de los valores de las amplitudes y orientación espacial de las fuerzas que provocan las vibraciones, entonces existe una lógica difusa capaz de reproducir los algoritmos de cálculo de dichos parámetros del desbalance y balanceo dinámico en dos planos.

La hipótesis que se formula como de trabajo es: si el paquete de programas G proporciona funciones similares a las empleadas por la lógica difusa, entonces un instrumento virtual puede ser diseñado como soporte programático para el balanceo dinámico de rotores en dos planos de la maquina MC-5.

CAPÍTULO I:

ESTADO DEL ARTE DEL BALANCEO DINÁMICO DE MÁQUINAS ROTATORIAS.

1.1. Introducción

El desbalance se puede definir de manera simple, como una distribución no uniforme de la masa de un rotor alrededor de su eje axial o de rotación, lo cual produce una desviación del centro de masa y/o del eje principal de inercia con respecto a su eje de rotación.

El centro de masa de un cuerpo se define como el punto en el cual se supone concentrada la masa, de tal manera que su efecto sea el mismo que el de la masa distribuida cuando el cuerpo se encuentra en movimiento de traslación. De manera similar, el eje principal de inercia es el eje alrededor del cual la masa está uniformemente distribuida y es el eje alrededor del cual tiende a rotar un cuerpo libremente en el espacio. Adicionalmente, el eje principal de inercia de un cuerpo siempre pasa por su centro de masa.

De manera natural, solo existen dos tipos de desbalance: Estático y Dinámico. El desbalance estático se produce cuando el centro de masa del rotor está desviado de su eje de rotación, pero el eje principal de inercia se mantiene paralelo al eje de rotación. Mientras que el desbalance dinámico ocurre cuando el eje principal de inercia está desviado angularmente con respecto al eje de rotación y el centro de masa se mantiene sobre el eje de rotación.

Existen muchas razones por las cuales se puede presentar desbalance de un rotor, esto es, razones por las cuales la masa de un rotor puede no estar uniformemente distribuida alrededor de su eje axial. Las causas más comunes del desbalance son:

- a) Tolerancias de Fabricación y Ensamblaje
- b) Asimetría de diseño
- c) No-Homogeneidad del Material
- d) Distorsión en Servicio (por liberación de esfuerzos y/o distorsión térmica)
- e) Corrosión y desgaste
- f) Acumulación de depósitos
- g) Desbalance estático
- h) Desbalance dinámico

La respuesta de una máquina al desbalance de su rotor depende de las características dinámicas del rotor y de la misma máquina. La amplitud del movimiento vibratorio generado y el ángulo de fase de este movimiento, con respecto al desbalance, son una función principalmente de la masa y la elasticidad de la máquina, además la deflexión

máxima de un rotor flexible en cualquiera de sus modos es controlada por el amortiguamiento de la máquina.

Aunque la vibración puede ser causada por una variedad de condiciones como la inclinación de los ejes, no alineación de los acoplamientos, rotura del soporte, etc., el desbalance es la fuente más común de grandes vibraciones que afectan a las máquinas rotatorias.

Los sistemas con rotores complejos tienen amplias aplicaciones industriales, como por ejemplo en turbinas de gas y vapor, en turbogeneradores, en aeronaves, etc. Las vibraciones forzadas en estos sistemas normalmente son causadas por las fuerzas centrífugas debido al desequilibrio de la masa del rotor y a varias fuerzas de transmisión. Aun cuando el rotor está bien equilibrado, el equilibrio puede deteriorarse con el uso.

Para el balanceo de rotores, tanto rígidos como flexibles, es importante conocer cómo varían la amplitud y el ángulo de fase de la respuesta, principalmente, con la relación entre la velocidad de rotación y la velocidad crítica.

Existen dos razones importantes por las cuales un rotor debe ser balanceado. La primera es que las fuerzas creadas por el desbalance son dañinas para la vida de la máquina. La magnitud de la fuerza creada es proporcional a la cantidad de desbalance y al cuadrado de la velocidad de rotación, así un desbalance relativamente pequeño puede producir fuerzas de gran magnitud en la máquina de alta velocidad.

La otra razón importante es la vibración indeseada generada, la cual puede ser dañina además, para los operadores, etc. Por otro lado, la vibración puede traer problemas de mala calidad de productos, como en el caso de máquinas herramientas.

El balanceo de rotores se puede definir como el proceso por medio del cual se ajusta la distribución de masa de un rotor de tal manera de hacerla más concéntrica con su eje de rotación, con el fin de reducir o controlar la carga sobre los cojinetes y la vibración sincrónica.

El desbalance de un rotor no se puede determinar y corregir directamente. Esto significa que en la práctica, no es posible determinar la cantidad y ubicación del desbalance de un rotor para corregirlo en su origen. Pero sí es posible determinar su efecto y calcular la cantidad y ubicación de las masas necesarias a agregar o quitar para reducir o controlar dicho efecto a niveles aceptables o tolerables.

El proceso de balanceo de un rotor se puede dividir en dos etapas:

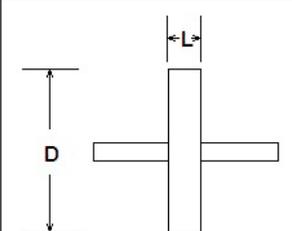
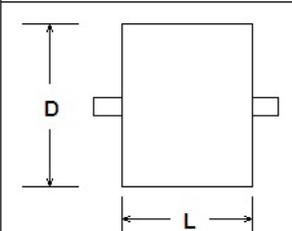
- La medición de los efectos del desbalance
- El cálculo de la cantidad y ubicación de las masas de corrección requeridas en cada plano de balanceo.

Entonces el problema es ¿cómo determinar la cantidad y ubicación de las masas de corrección?

Existe una variedad de técnicas o métodos de balanceo de rotores, pero el problema está en cual usar: ¿Balancear en Uno o Dos Planos?, ¿Balancear en Taller o en el Sitio? o ¿Balancear en Dos o en Múltiples Planos?

Un criterio práctico para seleccionar que tipo de balanceo se empleará puede reflejarse en la Tabla 1 y se fundamenta en la relación L/D y en el valor de la velocidad de rotación.

Tabla No. 1 Criterio práctico del tipo de balanceo

	RELACION L/D	PLANOS DE BALANCEO	
		UN PLANO	DOS PLANOS
	MENOR QUE 0.5	HASTA 1000 RPM	MAYOR QUE 1000 RPM
	MAYOR QUE 0.5	HASTA 150 RPM	MAYOR QUE 150 RPM

La guía mostrada en la Tabla No. 1 es solo una guía práctica, no se puede esperar que sea válida en todos los casos. El término “Balanceo en Taller” se debe a que el rotor debe ser desmontado de la máquina y transportado hasta el taller donde se encuentra la máquina balanceadora.

Una máquina balanceadora es una máquina especialmente diseñada para cumplir tres funciones fundamentales:

- Soportar el rotor a ser balanceado, con facilidad de montaje y desmontaje
- Hacer girar el rotor a una velocidad de balanceo pre establecida, y
- Medir el efecto dinámico del desbalance y calcular los pesos de corrección necesarios en cada plano de balanceo.

Las máquinas balanceadoras, sean de soportes rígidos o flexibles, miden la amplitud y el ángulo de fase de las fuerzas aplicadas a los cojinetes o del movimiento de los soportes, respectivamente. Luego, sobre la base de que el rotor se comporta de manera completamente rígida, calculan la cantidad y posición angular de las masas a agregar o quitar en cada plano de balanceo.

Las máquinas de balanceo dinámico de los rotores de los motores de inducción, tales como las máquinas MC-5, YYQ-160, YYW-1000, PHW-10000 o la máquina HQ-1000 son máquinas de fabricación foráneas y poseen software a los cuales no puede acceder el cliente, cuando la misma requiera de mantenimiento programático.

Por otro lado, un diseño casero del requerido software debe contemplar no solo el cálculo del peso a introducir y su ubicación, sino que también debe proporcionar campos para la introducción de datos de los rotores a prueba y gráficos para la interacción con el usuario (un sistema HMI), por lo que un nuevo soporte programático debe cumplir con estas exigencias, además de ser amigable.

Considerando esta problemática de diseño queda abierta la investigación del método de cálculo de los valores de los parámetros de salida de la máquina de balanceo dinámico a partir de los datos recopilados durante el balanceo en taller.

Las nuevas técnicas de entrenamiento de sistemas difusos o neuronales o su conjunción, neurodifuso, han demostrado ser herramientas idóneas para aplicaciones similares. Es por esto que el tema propuesto tiene como soporte las técnicas de los sistemas difusos, donde será necesario definir el sistema (funciones de pertenencia, ubicación de las mismas, conjunto de reglas, tipo de implicación a emplear, etc.) y el método más adecuado para el entrenamiento dirigido a determinar los valores requeridos en el balanceo dinámico.

Es así que empleando la lógica difusa como técnica de modelado, y partiendo de los datos de experimentos se pretende desarrollar un sistema difuso, el cual es el soporte principal para el diseño del instrumento virtual de balanceo dinámico de rotores de los motores eléctricos de inducción.

1.2 Clasificación de las máquinas de balanceo dinámico y sus problemas.

Las máquinas balanceadoras se pueden clasificar de acuerdo a su principio de operación en:

- Máquinas Estáticas o de Gravedad.
- Máquinas Dinámicas o Centrifugas.

El primer tipo corrige solo desbalance “estático”, determinando la posición angular del centro de masa (punto pesado) por efecto de “gravedad” sin poner el eje en rotación,

esto es, estáticamente. Mientras que el segundo tipo puede corregir desbalance estático y dinámico a través del efecto “dinámico” o “centrífugo” generado por el desbalance cuando el eje está rotando.

Una Máquina Balanceadora es una máquina especialmente diseñada y construida para el balanceo de rotores rígidos en taller. Todas las máquinas balanceadoras tienen la capacidad de determinar, por alguna técnica, la cantidad y posición angular del peso de corrección requeridos en cada plano de balanceo.

Considerando que son suficientes dos planos para balancear un rotor rígido, solo existen máquinas balanceadoras para uno y dos planos de balanceo. En algunos casos se balancean rotores flexibles de turbomáquinas en múltiples planos y a múltiples velocidades usando una máquina balanceadora de dos planos, pero esto equivale a realizar varios balanceos en dos planos redefiniendo los planos y cambiando la velocidad de balanceo entre una corrida y otra.

Las máquinas dinámicas se clasifican a su vez, en dos tipos:

- Máquinas Balanceadoras de Soporte Flexibles.
- Máquinas Balanceadoras de Soporte Rígido.

En las máquinas de soportes flexibles el efecto dinámico medido es el movimiento vibratorio de los soportes, mientras que en las de soporte rígido se mide la fuerza ejercida sobre los cojinetes. Estos dos tipos de máquinas dinámicas son capaces de procesar las señales medidas e indicar directamente, por técnicas diferentes, la

cantidad y posición angular de los pesos de corrección necesarios en uno o dos planos de balanceo.

Las máquinas balanceadoras dinámicas también se clasifican de acuerdo a su función en:

- Máquinas Balanceadoras de Mantenimiento.
- Máquinas Balanceadoras de Producción

Finalmente, las máquinas balanceadoras se pueden clasificar de acuerdo a su grado de automatización en:

- Máquinas Universales.
- Máquinas Semi-Automáticas.
- Máquinas Totalmente Automáticas.

Las máquinas universales son las típicamente usadas en mantenimiento, mientras que las totalmente automáticas se usan cuando el balanceo es una etapa más en el proceso de producción o fabricación de rotores.

Una clasificación de los métodos actuales en el balanceo de rotores tomado de Zhou and Shi, 2001 se muestra en la Figura 1.1

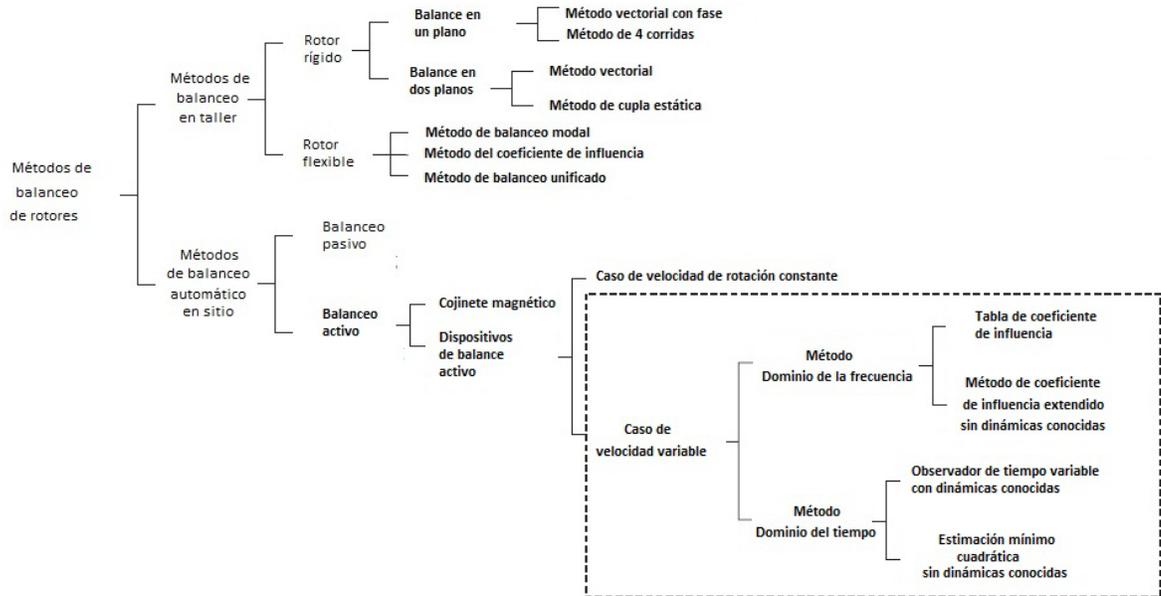


Figura 1.1 Clasificación de los Métodos actuales de balanceo de rotores.

1.3 Estado del Arte en el balanceo dinámico de rotores de las máquinas de inducción.

En la publicación de Knospe y Tamer, 1997, se estudia una serie de experimentos encaminados a determinar la robustez de la estabilidad y del funcionamiento de un algoritmo para la supresión de la vibración inducida por el desbalance de un rotor en cojinetes magnéticos. Los resultados se emplean para validar herramientas de análisis y síntesis.

Los resultados de la síntesis producen una matriz de ganancias del algoritmo de adaptación, la cual es robusta para la incertidumbre estructurada del rotor. Se sintetizan y prueban 3 matrices. Cada una de ellas proporcionan mayor robustez que las soluciones estándares del problema. Los resultados experimentales también

demuestran que el grado de estabilidad y la robustez del funcionamiento obtenido con las matrices de ganancias están muy cerca de los especificados en el procedimiento de diseño.

Las máquinas convencionales de balanceo utilizan dos planos separados para determinación del desbalance dinámico. Sin embargo las masas desbalanceadas de un eje de grúa no pueden ser corregidas mediante la medición en dos planos.

En el estudio presentado por Kang y otros, 2000, se ofrece la información de sus resultados en el empleo de un método modificado para el balanceo de tales ejes con máquinas de soporte suaves. Se probaron dos métodos, el Método del coeficiente de influencia modificado y el Método de Interacción, fundamentado en el Enfoque del Coeficiente de Influencia.

En el artículo de Pavlovskaja y otros, 2004, se desarrolla el modelo de dos grados de libertad de un rotor tipo Jeffcott sujeto a una excitación fuera de balance por un anillo amortiguador pre cargado. Se investigan las situaciones puras de impacto.

El rotor hace contactos intermitentes con el anillo amortiguador pre cargado y como consecuencia puede estar en uno de cinco diferentes regímenes de contacto, cuyos límites se han encontrado analíticamente.

La ubicación del anillo amortiguador ha sido determinada usando el principio de la mínima energía elástica en el anillo del amortiguador. Por consiguiente, se obtuvo y estudió numéricamente un sistema dinámico no lineal a trozos suaves. Los resultados,

en forma de diagramas de bifurcación, plano de fase y mapas de Poincare muestran diferencias significativas para los casos con y sin el anillo amortiguador pre cargado.

En el artículo de Lin y otros, 2005, se investiga la utilización de los Algoritmos Genéticos con selección elitista para investigar las influencias del número de condición del método HMIC en la exactitud del balanceo. Además, se establece el método para optimizar la posición de los planos de balanceo y/o los sensores de medición del balanceo del rotor.

Adicionalmente, se comparan la exactitud del balanceo y la velocidad de convergencia obtenidas con esta propuesta con otros planos de balanceo y sensores de medición. Entonces, se obtienen las ubicaciones óptimas de los planos de balanceo y sensores como la optimización definitiva. Se emplea además, el método de elementos finitos para determinar la respuesta en estado estacionario del sistema rotor-cojinete flexible.

La optimización mejora la exactitud de balanceo, la cual puede validarse por los experimentos de balanceo en un kit de rotor.

En el artículo del autor, Al-Taee, 2007, el mismo se enfoca en resolver el problema del desbalance de masas que afecta a los rotores en forma de disco tales como: las bombas centrífugas, los ventiladores, los motores, volantes de la turbina, etc., logrando el balanceo en un equipo de prueba que usa las mediciones de amplitud de la vibración tomadas de una corrida inicial y cuatro corridas con masas de ensayo.

Presenta, además, un nuevo análisis teórico para desarrollar las ecuaciones matemáticas y se implementa un programa de cómputo específicamente escrito por el autor.

En Lu y otros, 2008, se estudia un balanceador automático del tipo bola, el cual consiste en varias bolas rotando, que puede reducir de manera efectiva las vibraciones no balanceadas del accionamiento del disco óptico. Bajo condiciones apropiadas de trabajo las bolas pueden balancear el desbalance de un disco mediante la ubicación de ángulos relativos apropiados con el centro de masas del disco. A esta posición particular se refieren como las posiciones perfectas de balanceo.

Una gran cantidad de trabajo se ha realizado para determinar la estabilidad de las posiciones de equilibrio de forma numérica para algunos valores específicos de los parámetros del sistema. En contraste con esto, en este trabajo se estudia de manera analítica las características de las posiciones de equilibrio bajo ciertas condiciones.

Las ecuaciones de gobierno del modelo teórico se obtienen empleando las ecuaciones de Lagrange. Se presentan las ecuaciones en forma cerrada y las condiciones de existencia para las posiciones de equilibrio. El criterio de Routh es el empleado para determinar las posiciones de equilibrio y finalmente se proponen las guías sobre la estabilidad de las posiciones de equilibrio.

En la publicación de Manka y Uhl, 2008, se presentan los resultados teóricos y el marco de los experimentos de investigación de un sistema de detección y aislamiento de fallos

y restauración (FDIR en idioma inglés) para una máquina rotatoria creada en el Departamento de Robótica y Mecatrónica de la Universidad de Ciencias y Tecnología AGH en Cracovia.

La solución presentada concierne al problema del balanceo automático de los sistemas rotatorios mientras están en operación. Los resultados experimentales y las simulaciones proporcionaron que las aplicaciones de la solución encontrada conducen a la reducción en las vibraciones causadas por el desbalance, sin impedir el servicio. El enfoque mecatrónico al proceso de diseño hace que este proceso sea más efectivo.

En Mahfoud y otros, 2009, se presenta un método para controlar y monitorear las máquinas rotatorias. Se fundamenta en la idea de representar la máquina rotatoria como un modelo equivalente reducido obtenido por identificación de manera experimental. La identificación se ha realizado empleando un número limitado de fuerzas excitadoras y sensores de desplazamiento ubicados a lo largo del eje. Para identificar el modelo se empleó la técnica del mínimo cuadrado. El orden del modelo reducido se corresponde con la cantidad de sensores empleados. Una vez que se ha identificado la matriz de espacio de estados las fuerzas de excitación externas pueden ser determinadas en tiempo real por el rango de frecuencias considerado.

Ya que la determinación de las fuerzas externas es muy importante en el estudio de el comportamiento dinámico de las máquinas rotatorias, su modelo inverso puede ser fácilmente adaptado para el monitoreo de las aplicaciones.

En el artículo de Pandey y Nakra, 2011, se plantea que el monitoreo de la vibración es una técnica útil para la aplicación en las máquinas rotatorias y proporciona una valiosa información con respecto a los síntomas de fallos de la máquina que en la práctica pueden evitar las averías costosas.

Este artículo involucra el diseño y la fabricación de un kit del rotor y las investigaciones de las vibraciones en los cojinetes del equipo debido al efecto de fallas simuladas, el equipo de prueba se diseña en el caso de carga dinámica y también incluye el empleo de varios componentes y la fabricación del kit del rotor. Fueron simuladas fallas tales como las no alineaciones paralelas, no alineaciones angulares, la combinación de ambas y los desbalances.

Para estudiar los efectos de las fallas simuladas, se monitoreó el valor medio cuadrático (RMS) de la aceleración de las vibraciones en las direcciones vertical, horizontal y axial de los soportes flexibles tipo cojinetes empleando acelerómetros piezoeléctricos.

Se ha llevado a cabo la revisión de la literatura en un periodo grande de tiempo, desde 1997 hasta 2011. Los resultados demuestran el interés de la comunidad científica internacional por el problema del desbalance de los rotores de las máquinas rotatorias, los esfuerzos realizados en estos años por determinar modelos matemáticos confiables y estables a partir de las mediciones a diferentes variables, como son: el desplazamiento de los soportes o apoyos, la velocidad de desplazamiento, su aceleración y por tanto las fuerzas en los apoyos con procesamiento directo de estas

magnitudes o mediante análisis espectral o empleando la transformada de Fourier de tales señales.

Se destacan los métodos aplicados para mejorar la exactitud en la estimación del desbalance dinámico por disímiles métodos y en su compensación, así como también en la aplicación de los conocimientos adquiridos para descubrir la influencia del desbalance en los fallos de las máquinas rotatorias. Se pone de manifiesto, además, los esfuerzos y recursos aplicados en mejorar cada día las máquinas para el monitoreo en sitio y en talleres.

1.3.1 Método de balanceo por los coeficientes de influencia.

El método de los coeficientes de influencia, expuesto en un gran número de textos de balanceo dinámico de rotores rígidos, entre los cuales se mencionan (Neville F. Rieger, 1986), se fundamenta en la existencia de dos fuerzas de desbalance aplicadas sobre el rotor en rotación, las cuales se pueden calcular como:

$$F = \frac{W_1 a_1 \omega^2}{g} = \frac{\omega^2}{g} U_1; \text{ con } U_1 = W_1 a_1 \quad (1.1)$$

$$R = \frac{W_2 a_2 \omega^2}{g} = \frac{\omega^2}{g} U_2; \text{ con } U_2 = W_2 a_2 \quad (1.2)$$

Siendo W_1 y W_2 los pesos de dos discos equivalentes a la mitad del peso del rotor, cada uno, y a_1 y a_2 son las excentricidades respectivas.

Las ecuaciones de equilibrio para las reacciones de los cojinetes R y S son:

$$\bar{\alpha}_{11}F + \bar{\alpha}_{12}\bar{P} - \bar{R} = 0 \quad (1.3)$$

$$\bar{\alpha}_{21}F + \bar{\alpha}_{22}\bar{P} - \bar{R} = 0 \quad (1.4)$$

Donde: $\bar{\alpha}_{11}$, $\bar{\alpha}_{12}$, $\bar{\alpha}_{21}$ y $\bar{\alpha}_{22}$ son los coeficientes de influencia del vector que relacionan las fuerzas de desbalance con las reacciones de los cojinetes. Ellos se consideran magnitudes complejas. Mediante la sustitución de las mismas en las ecuaciones (1.3) y (1.4) y separando la parte real y la imaginaria queda formada la ecuación matricial compleja siguiente:

$$[A][G] = [H] \quad (1.5)$$

Siendo, entonces:

$$[A] = \begin{bmatrix} \alpha_{11}^r & -\alpha_{11}^i & \alpha_{12}^r & -\alpha_{12}^i \\ \alpha_{11}^i & \alpha_{11}^r & \alpha_{12}^i & \alpha_{12}^r \\ \alpha_{21}^r & -\alpha_{21}^i & \alpha_{22}^r & -\alpha_{22}^i \\ \alpha_{21}^i & \alpha_{21}^r & \alpha_{22}^i & \alpha_{22}^r \end{bmatrix} \quad (1.6)$$

$$[G] = \begin{bmatrix} F^r \\ F^i \\ P^r \\ P^i \end{bmatrix} \quad (1.7)$$

$$[H] = \begin{bmatrix} R^r \\ R^i \\ S^r \\ S^i \end{bmatrix} \quad (1.8)$$

Donde los supra índices indican la parte real (r) y la parte imaginaria (i) de los coeficientes y fuerzas.

Los coeficientes de influencia se determinan ubicando un peso balanceador conocido de prueba B en cada plano de balanceo por orden, lo que proporciona una fuerza de desbalance:

$$\bar{T}_r = \frac{\omega^2}{g} \bar{B}_r, r = 1,2 \quad (1.9)$$

Considerando esta nueva fuerza en cada plano de balanceo y desarrollando las ecuaciones de balance de manera similar al desarrollo anterior, en parte real e imaginaria y restando las ecuaciones de balance mecánico desarrolladas también en parte real e imaginaria respectivamente se tienen las siguientes 8 expresiones para los coeficientes de influencia:

$$\alpha_{11}^r = \alpha_{11} \cos \zeta_{11} = \frac{R_1}{T_1} \cos(\zeta_{R_1} - \zeta_{T_1}) - \frac{R}{T_1} \cos(\zeta_R - \zeta_{T_1}) \quad (1.10)$$

$$\alpha_{11}^i = \alpha_{11} \operatorname{sen} \zeta_{11} = \frac{R_1}{T_1} \operatorname{sen}(\zeta_{R_1} - \zeta_{T_1}) - \frac{R}{T_1} \operatorname{sen}(\zeta_R - \zeta_{T_1}) \quad (1.11)$$

$$\alpha_{21}^r = \alpha_{21} \cos \zeta_{21} = \frac{S_1}{T_1} \cos(\zeta_{S_1} - \zeta_{T_1}) - \frac{S}{T_1} \cos(\zeta_S - \zeta_{T_1}) \quad (1.12)$$

$$\alpha_{21}^i = \alpha_{21} \operatorname{sen} \zeta_{21} = \frac{S_1}{T_1} \operatorname{sen}(\zeta_{S_1} - \zeta_{T_1}) - \frac{S}{T_1} \operatorname{sen}(\zeta_S - \zeta_{T_1}) \quad (1.13)$$

$$\alpha_{12}^r = \alpha_{12} \cos \zeta_{12} = \frac{R_2}{T_2} \cos(\zeta_{R_2} - \zeta_{T_2}) - \frac{R}{T_2} \cos(\zeta_R - \zeta_{T_2}) \quad (1.14)$$

$$\alpha_{12}^i = \alpha_{12} \operatorname{sen} \zeta_{12} = \frac{R_2}{T_2} \operatorname{sen}(\zeta_{R_2} - \zeta_{T_2}) - \frac{R}{T_2} \operatorname{sen}(\zeta_R - \zeta_{T_2}) \quad (1.15)$$

$$\alpha_{22}^r = \alpha_{22} \cos \zeta_{22} = \frac{S_2}{T_2} \cos(\zeta_{S_2} - \zeta_{T_2}) - \frac{S}{T_2} \cos(\zeta_S - \zeta_{T_2}) \quad (1.16)$$

$$\alpha_{22}^i = \alpha_{22} \operatorname{sen} \zeta_{22} = \frac{S_2}{T_2} \operatorname{sen}(\zeta_{S_2} - \zeta_{T_2}) - \frac{S}{T_2} \operatorname{sen}(\zeta_S - \zeta_{T_2}) \quad (1.17)$$

Cuando estas expresiones son evaluadas a partir de los datos obtenidos con los pesos de prueba, se determinan los coeficientes. Queda determinada, así, la matriz de los coeficientes de influencia. Calculando la inversa de esta matriz quedara determinada la matriz columna G por la expresión:

$$[G] = [A]^{-1}[H] \quad (1.18)$$

Los pesos requeridos para balancear el rotor desbalanceado se determinan como sigue:

$$F_1 = \sqrt{(F_1^r)^2 + (F_1^i)^2} \quad (1.19)$$

$$C_1 = U_1 = \left(\frac{g}{\omega^2}\right) F_1 \quad (1.20)$$

$$W_{C1} = \frac{U_1}{r_1} \quad (1.21)$$

$$P_1 = \sqrt{(P_1^r)^2 + (P_1^i)^2} \quad (1.22)$$

$$C_2 = U_2 = \left(\frac{g}{\omega^2}\right) P_1 \quad (1.23)$$

$$W_{C2} = \frac{U_2}{r_2} \quad (1.24)$$

Con r1 y r2 radios en los planos de corrección respectivamente. Los ángulos de orientación, donde deben ser ubicados los pesos calculados se determinan por las expresiones siguientes:

$$\zeta_F^C = \arctan\left(\frac{F_1^i}{F_1^r}\right) + 180^\circ \quad (1.25)$$

$$\zeta_P^C = \arctan\left(\frac{P_1^i}{P_1^r}\right) + 180^\circ \quad (1.26)$$

Por tanto, el desbalance se corrige ubicando ambos pesos correctores en los planos de corrección en los ángulos determinados por las expresiones (1.25) y (1.26) ya que el primer sumando de esas expresiones es la ubicación angular del desbalance original.

Para llevar a cabo los cálculos por este método será necesario determinar los valores máximos de las reacciones a partir de las mediciones realizadas en los pedestales de la instalación mecánica de la maquina balanceadora. Además, se requiere determinar, cada vez, la posición angular de estos valores máximos con respecto a la señal de referencia tomada desde el transductor estroboscópico.

Luego de realizadas estas mediciones y convertidos estos valores en señales digitales, los resultados se aplican al programa de cálculo de los pesos y magnitudes angulares para su ubicación. Este programa de cálculo se desarrolla formando un algoritmo de cómputo desde la expresión (1.6) hasta la (1.26).

El algoritmo programado requiere:

- Los pesos de prueba $W1$ y $W2$,
- Amplitudes de los sensores en los pedestales $R, R1, R2, S, S1$ y $S2$,
- Medición de fases $\zeta_R, \zeta_{R1}, \zeta_{R2}, \zeta_S, \zeta_{S1}, \zeta_{S2}$ con respecto a la magnitud de referencia del sensor estroboscópico,
- Cálculo de los coeficientes alfa,
- Cálculo de la matriz inversa A^{-1}
- Cálculo de la matriz G , que involucra el producto de dos matrices,
- Cálculo de $F1, C1, W1$ y ζ_F^C
- Cálculo de $P1, C2, W2$ y ζ_P^C

Como se puede apreciar, estas operaciones consumen un gran recurso de memoria y programático. Debe considerarse además, que las operaciones indicadas conllevan a realizar diferentes corridas o pruebas para determinar el desbalance inicial y el desbalance con los pesos de prueba en cada plano de balanceo y la comprobación final del desbalance residual.

Si el desbalance residual supera los valores permitidos de desbalance para cada rotor de motor, las operaciones indicadas deben realizarse nuevamente, lo cual consume un gran recurso de tiempo y energía eléctrica.

1.4 Los Sistemas difusos como calculadores de parámetros.

Un gran número de trabajos publicados se han dedicado al problema de Aproximación de Funciones con los sistemas difusos o lógica difusa. Se resumen a continuación los trabajos que han sido relevantes en esta esfera.

La importancia del artículo de [Takagi, Sugeno, 1985] radica en que sus autores son los creadores del modelo llamado más tarde “de Takagi-Sugeno”. En él se presenta una herramienta matemática para construir un modelo difuso de un sistema donde se emplean las implicaciones y el razonamiento.

Demuestran que la premisa de una implicación es la descripción del sub espacio de las entradas y sus consecuentes son una relación lineal entre la entrada y la salida. Se muestra el método de identificación de un sistema empleando sus datos entrada-salida.

Se discuten además dos aplicaciones del método para los procesos industriales: un proceso de purificar el agua y un convertidor en un proceso de fabricación de acero.

Ya en ese tiempo destacaban que “el modelado difuso basado en las implicaciones difusas y el razonamiento puede ser uno de los campos de mayor importancia en la teoría de los sistemas difusos”.

En Kosco, 1994 se plantea que “un sistema borroso aditivo puede aproximar cualquier función realmente continua, de manera uniforme, en un dominio compacto con cualquier grado de precisión.

Más adelante plantea que este sistema aproxima la función cubriendo su gráfico con parches borrosos en el espacio de estados entrada-salida y promediando los parches que se solapan. Si se ven los conjuntos borrosos como conjuntos casuales entonces el sistema borroso calcula un valor esperado condicional.

Cada regla borrosa define un parche borroso y conecta el conocimiento del sentido común con el espacio geométrico de estados.

Mendel y Mouzouris, 1997, declaran que los Sistemas Lógicos Borrosos son sistemas no lineales capaces de inferir las relaciones no lineales complejas entre las variables de entrada y salida. La propiedad no lineal es particularmente importante cuando el mecanismo físico subyacente a ser “mapeado” es inherentemente no lineal.

El sistema puede “aprender” la no linealidad si se presenta una sucesión de señales de entrada y la respuesta deseada correspondiente usando además un algoritmo de

optimización para determinar los valores de los parámetros del sistema. Éste es uno de los paradigmas de aprendizaje normalmente usados, llamado Aprendizaje Dirigido.

Más adelante, en la misma fuente se plantea: “Aun cuando el proceso a ser mapeado es no estacionario, el sistema puede actualizarse para que refleje las estadísticas cambiantes del proceso”.

En el artículo de [Ying, 1998], se prueba constructivamente una clase general de modelo de sistema difuso de múltiples entradas y una sola salida de Takagi-Sugeno (TS) como aproximador universal. Los sistemas usan cualquier tipo de conjuntos difusos continuos, cualquier tipo de lógica difusa “AND”, de reglas difusas y consecuente lineal y un dispositivo generalizado que realiza la acción contraria al difusificador.

Se demuestra primero que los sistemas difusos TS pueden aproximar uniformemente y arbitrariamente bien cualquier polinomio multivariable, y se demuestra que ellos pueden aproximar, arbitrariamente bien, cualquier función uniformemente continua multivariable.

Se obtuvo por los autores una fórmula para calcular los límites superiores mínimos en el número de conjuntos difusos y las reglas difusas necesarias para lograr la exactitud especificada con antelación de la aproximación para cualquier función dada de dos variables. Se presenta un ejemplo numérico. Los resultados obtenidos proporcionan una base teórica sólida para las aplicaciones del sistema difuso, particularmente como controladores y modelos difusos.

Otro de los artículos publicados al finalizar el siglo pasado (Yen y otros, 1998) promueve la idea de que el sistema de inferencia difuso propuesto por Takagi, Sugeno y Kang, conocido en la literatura de Sistemas Difusos como el Modelo TSK, proporciona una herramienta poderosa para modelar los sistemas no lineales complejos.

Al contrario del modelado convencional, donde se usa un solo modelo para describir la conducta global de un sistema, el modelo TSK es esencialmente un enfoque de modelo múltiple en que los sub modelos simples (modelos típicamente lineales) se combinan para describir la conducta global del sistema. Se destaca que la mayoría de los algoritmos de aprendizaje existentes para identificar al modelo de TSK se basan en minimizar el cuadrado del residuo entre las salidas globales del sistema real y las salidas del modelo identificado.

Se plantea que aunque estos algoritmos pueden generar un modelo de TSK con buena actuación global (es decir, el modelo es capaz de aproximar el sistema dado con exactitud arbitraria, con tal de que se usen las reglas y los suficientes datos de entrenamiento que estén disponibles), ellos no pueden garantizar que el modelo resultante tenga una buena actuación local.

Destacan que a menudo estos sub modelos en el modelo de TSK presentan una conducta errática difícil de interpretar. Ya que una de las motivaciones importantes de usar el modelo TSK (también otros modelos difusos) es ganar visión en el modelo, investigan la interpretabilidad del modelo TSK. En este artículo, se propone un nuevo

algoritmo de aprendizaje que integra el aprendizaje global y el aprendizaje local en un solo algorítmico.

Este algoritmo usa la idea de regresión ponderada local y la aproximación local en la estadística no paramétrica, pero mantiene el componente de ajuste global en los algoritmos existentes de aprendizaje. El algoritmo es capaz de ajustar sus parámetros basado en la preferencia del usuario, los modelos generadores con buen intercambio en términos referidos al ajuste global y la interpretación local. Se ilustra la actuación del algoritmo propuesto empleando un ejemplo de modelado de motocicleta.

En la monografía clásica de (Passino y Yurkovich, 1998) se fundamenta la construcción de los sistemas borrosos destinados a la estimación de parámetros.

En el Capítulo 5, numeral 5.2 de la mencionada monografía se dice “Comenzamos esta sección precisamente definiendo el problema de aproximación de funciones, en la cual se sintetiza una función para aproximar otra función que está representada mediante un número finito de asociaciones entrada salida (i.e., solo se conoce como la función mapea un número finito de puntos desde su dominio a su rango)”.

Para hacer esto, se necesitan datos que muestren aproximadamente cómo debe comportarse el mapeo entrada-salida del estimador (es decir, cómo debe estimar).

Una manera de generar estos datos se expone también en (Passino y Yurkovich, 1998), donde reflejan “empezar estableciendo una prueba de simulación para la planta para la

cual debe realizarse la estimación del parámetro. Entonces se lleva a cabo un conjunto de simulaciones, cada uno con un valor diferente para el parámetro a ser estimado”.

Acoplando las condiciones de prueba y los datos simulados-generados con el valor del parámetro, se recolectan los pares de datos apropiados que permiten construir un estimador borroso. Se indica, además, que para algunas plantas puede ser posible realizar este procedimiento con los datos experimentales reales (ajustando físicamente el parámetro a ser estimado).

En Guillaume, 2001, se exponen los resultados del amplio uso de los Sistemas de Inferencia Borrosos (FIS). Se reafirma el hecho que tales sistemas o pueden diseñarse partiendo del conocimiento del especialista o pueden diseñarse partiendo de los datos experimentales. Se destaca el problema que en los sistemas complejos, los sistemas FIS, basados sólo en el conocimiento del especialista, pueden padecer de una pérdida de exactitud.

Éste resultado es el incentivo principal para el empleo de las reglas borrosas inferidas de los datos. Se propone la descomposición en dos fases de los diseños de los FIS partiendo de los datos: la generación automática de las reglas y la optimización del sistema.

Declaran que la generación de las reglas conduce a un sistema básico con una partición dada del espacio y el conjunto correspondiente de reglas. Afirman que la optimización del Sistema Borroso puede hacerse en varios niveles.

Por último determinan que:

1. La selección de la variable puede ser por una selección global o puede manejarse regla a regla.
2. La optimización de la base de reglas apunta a seleccionar las reglas más útiles y perfeccionar las conclusiones de la regla.
3. La partición del espacio puede mejorarse agregando o quitando los conjuntos difusos y poniendo a punto los parámetros de las funciones de pertenencia.
4. La optimización de la estructura es de una importancia mayor para lo cual seleccionan las variables, reducen la base de reglas y perfeccionan el número de conjuntos borrosos.

A través de los años, muchos métodos de diseño de los FIS a partir de los datos se han puesto a disposición de los ingenieros e investigadores. Su eficiencia, usualmente, se caracteriza por el índice de funcionamiento numérico. Sin embargo, para la cooperación hombre-máquina se necesita otro criterio: la interpretabilidad de las reglas.

Un supuesto implícito establece que las reglas borrosas son por naturaleza interpretables. Esto podría no ser así al tratar con sistemas multivariables complejos o cuando la partición generada no tiene sentido para los expertos.

En este artículo, además, se definen tres condiciones para que un conjunto de reglas sea interpretable.

En (Chen and Linkes, 2004) se presentan los resultados obtenidos al aplicar la autoextracción de la base de reglas y el método de simplificación para establecer modelos difusos interpretables a partir de datos numéricos.

El modelado difuso llevado a cabo por los datos numéricos se ha usado en una gran variedad de aplicaciones. Sin embargo, en los modelos basados en reglas difusas por los datos numéricos adquiridos, a menudo existe la redundancia en forma de reglas redundantes o de conjuntos difusos similares. Esto produce una innecesaria complejidad estructural y disminuye la interpretabilidad del sistema.

En este artículo se usa una técnica de agrupamiento (cluster) difusa asociada con el índice de validez de la partición difusa propuesto para, inicialmente extraer la base de reglas difusa y encontrar el número óptimo de reglas difusas. Para reducir la complejidad de los modelos difusos, manteniendo una buena exactitud del modelo, se presentan algunas medidas de similitud aproximadas y se introduce el mecanismo de ajuste fino del parámetro para mejorar la exactitud del modelo simplificado. Usando las mediciones de similitud propuestas, se eliminan las reglas difusas redundantes y los conjuntos difusos alejados y se crea un conjunto difuso común en la base de reglas. La base de reglas simplificada es computacionalmente eficiente y lingüísticamente interpretable.

El enfoque se ha aplicado con éxito al modelado difuso para la aproximación de funciones no lineales y en el modelado de sistemas dinámicos.

Los artículos referenciados demuestran las líneas de desarrollo seguidas por la comunidad internacional en el modelado difuso y la aproximación de funciones, principalmente no lineales.

Se destaca el interés por encontrar modelos interpretables y no redundantes, estables y de rápida convergencia al ajuste fino de las funciones. De igual forma se han estudiado las potencialidades de los métodos para el entrenamiento de los modelos difusos de manera similar a como se entrenan las redes neuronales artificiales, siendo los métodos más empleados el Método de los Mínimos Cuadrados en todas sus variantes y el de Descenso por el Gradiente con buenos resultados.

1.4.1 Identificación y estimación con sistemas difusos

El problema básico que subyace en esta tesis es cómo construir un sistema difuso a partir de los datos numéricos. El marco teórico para la identificación y estimación de parámetros se formula como sigue. Dada alguna función:

$$g: \bar{X} \rightarrow \bar{Y} \tag{1.27}$$

Donde

$$\bar{X} \subset \mathfrak{R}^n \text{ y } \bar{Y} \subset \mathfrak{R}$$

Se desea construir un sistema difuso

$$f: X \rightarrow Y \tag{1.28}$$

Dónde:

$X \subset \bar{X}$ y $Y \subset \bar{Y}$ son ciertos dominios y rangos de interés.

Escogiendo un vector de parámetros θ (el cual puede incluir a las funciones de pertenencia, centros de las mismas, anchos, etc.), de tal forma que:

$$g(x) = f(x|\theta) + e(x) \quad (1.29)$$

$$\forall x = [x_1, x_2, \dots, x_n]^T \in X \quad (1.30)$$

donde el error de aproximación $e(x)$ es tan pequeño como sea posible.

Supóngase que todo lo que se conoce para escoger los parámetros θ del sistema difuso $f(x|\theta)$ es alguna parte de la función g en forma de un conjunto finito de pares de datos de entrada-salida (es decir, el mapeo funcional implementado por g es esencialmente desconocido).

El r -ésimo par de datos de entrada-salida desde el sistema g se denota por (x^r, y^r) donde $x^r \in X$, $y^r \in Y$, y $y^r = g(x^r)$.

Sea el par de datos (x^r, y^r) la representación del vector de entrada. Ya que cada dato de entrada es n -dimensional, se tiene:

$$x^r = [x_1^r, x_2^r, \dots, x_n^r]^T \quad (1.31)$$

Que representa al vector de entrada para el r -ésimo par de entrada. Por tanto, x_s^r es el elemento s -ésimo del r -ésimo vector de datos (tiene un valor específico y no es una variable). El conjunto de pares de datos de entrada salida se denomina “conjunto de datos de entrenamiento” y se denota por

$$G = \{(x^1, y^1), \dots, (x^M, y^M)\} \subset X \times Y \quad (1.32)$$

donde M denota el número de pares de datos de entrada salida contenido en G

con $r=[1, 2, \dots, M]$

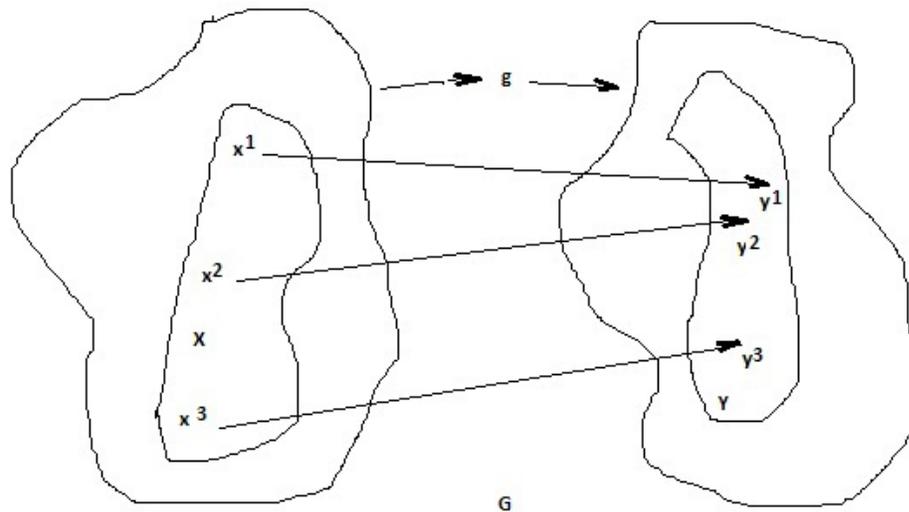


Figura 1.2. Función que mapea con tres pares de datos de entrada salida conocidos.

Un cuadro gráfico del problema de la Aproximación de Función se proporciona en la Figura 1.2. Ella muestra claramente cuál es el desafío; puede ser ciertamente difícil proponer una buena función f que iguale el mapeo g si sólo se sabe algo sobre la asociación entre X y Y en la forma de pares de datos G . Más aun, puede ser difícil saber cuándo se tiene una buena aproximación, esto es ¿cuándo f aproxima a g en todo el espacio de las entradas X ?

Entonces, el problema de la aproximación de una función se concentra en encontrar una función $f(x|\theta)$ mediante la manipulación del vector θ , de tal forma que aproxime a g tan cerca como sea posible.

La pregunta ahora es: ¿Cómo se evalúa un sistema difuso $f(x|\theta)$ que aproxime estrechamente a la función $g(x)$ para todo $x \in X$ dado un vector de parámetros θ ?

Supóngase que existe la cota:

$$\sup_{x \in X} \{|g(x) - f(x|\theta)|\} \quad (1.33)$$

Que es un límite en el error de aproximación (si existe). Pero especificar tal límite requiere conocer la función g completamente; sin embargo, sólo conocemos una parte de g dada por G . Por tanto, sólo se podrá evaluar la exactitud de aproximación evaluando el error entre $g(x)$ y $f(x|\theta)$ en cierto punto $x \in X$ dado por los datos disponibles de entrada-salida. Este conjunto de datos entrada-salida se denominará conjunto de prueba y se denota (Passino and Yurkovich, 1998) como Γ , dónde

$$\Gamma = \{(x^1, y^1), \dots, (x^{M_\Gamma}, y^{M_\Gamma})\} \subset X \times Y \quad (1.34)$$

Siendo M_Γ el número de pares de datos conocidos de entrada-salida contenido en el conjunto de prueba.

La evaluación del error en la aproximación entre g y el sistema difuso $f(x|\theta)$ basado en un conjunto de prueba Γ puede o no, (Passino and Yurkovich, 1998), ser una verdadera medida del error entre g y f para cada $x \in X$, pero es la única evaluación que se puede hacer basada en la información conocida. Por tanto, se pueden usar mediciones parecidas a estas para medir el error de la aproximación.

La aproximación exacta de la función requiere que alguna expresión de esta naturaleza sea pequeña; sin embargo, esto no garantiza una representación perfecta de g con f ya que con frecuencia no se puede probar que esa función f iguale a g en todos los posibles puntos de la entrada.

La elección del vector del parámetro depende, por ejemplo, de cuántas funciones de pertenencia y reglas se usen. Generalmente, se quiere tener un número grande de funciones de pertenencia y de reglas para conseguir una buena exactitud, pero no demasiadas ya que si la función está “sobre parametrizada” esto puede realmente degradar la exactitud de la aproximación.

A menudo, es mejor si la estructura del aproximador se basa en un poco de conocimiento físico del sistema.

Finalmente, en la presente Tesis se enfoca, principalmente, a los sistemas difusos aunque a veces puede ser beneficioso usar otra aproximación estructural como las redes neurales, polinomios, wavelets, o splines.

Suponga que el sistema difuso construido de los datos numéricos se da por las expresiones (1.35) y (1.36):

$$\mathbf{f}(\mathbf{x}) = \frac{\sum_{i=1}^R \mathbf{b}_i \mu_i(\mathbf{x})}{\sum_{i=1}^R \mu_i(\mathbf{x})} \quad (1.35)$$

Donde,

$$\mu_i(\mathbf{x}) = \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j - c_j^i}{\sigma_j^i}\right)^2\right) \quad (1.36)$$

Este modelo usa la fusificación simple, las funciones de pertenencia del tipo Gauss, el producto para la premisa y la implicación, y la des fusificación mediante el promedio de los centros. Tiene R reglas, los centros de las funciones de pertenencia de salida en b_i , los centros de las funciones de pertenencia de entrada en c_j^i , y un ancho de las funciones de pertenencia de entrada σ_j^i

1.4.2 Ajuste de los Sistemas Difusos

Es posible usar los métodos de los mínimos cuadrados descritos en [Ljung, 1999] para o poner a punto los sistemas difusos por lote o en el modo de tiempo real. En (Passino and Yurkovich, 1998) se explica cómo poner a punto los sistemas difusos estándares y los de Takagi-Sugeno con muchas entradas y sólo una salida. Para entrenar los sistemas difusos con muchas salidas, simplemente se repite el procedimiento descrito para cada salida.

Sistemas Difusos estándares

Un sistema difuso estándar puede ser descrito como sigue:

$$y = f(x|\theta) = \frac{\sum_{i=1}^R b_i \mu_i(x)}{\sum_{i=1}^R \mu_i} \quad (1.37)$$

donde $x = [x_1, x_2, \dots, x_n]^T$ y $\mu_i(x)$ definido como la certeza de la premisa de la i -ésima regla (especificado mediante las funciones de pertenencia en el universo de discurso de la entrada junto con la elección del método a usar en la norma triangular para

representar la conjunción en la premisa). Los valores b_i , con $i = 1, 2, \dots, R$ son los centros de las funciones de pertenencia de la salida.

Descomponiendo (1.37) en suma de fracciones con igual denominador, se tiene:

$$y = f(x|\theta) = \frac{b_1\mu_1(x)}{\sum_{i=1}^R \mu_i} + \frac{b_2\mu_2(x)}{\sum_{i=1}^R \mu_i} + \dots + \frac{b_R\mu_R(x)}{\sum_{i=1}^R \mu_i} \quad (1.38)$$

Con

$$\xi_i(x) = \frac{\mu_i(x)}{\sum_{i=1}^R \mu_i} \quad (1.39)$$

Se replantea (1.38) como sigue:

$$y = f(x|\theta) = b_1\xi_1(x) + b_2\xi_2(x) + \dots + b_R\xi_R(x) \quad (1.40)$$

Si se definen los vectores

$$\xi(x) = [\xi_1, \xi_2, \dots, \xi_R]^T, \text{ y } \theta = [\theta_1, \theta_2, \dots, \theta_R]^T \text{ entonces (1.39) queda en forma compacta:}$$

$$y = f(x|\theta) = \theta^T \xi(x) \quad (1.41)$$

Como se observa, la forma del modelo que debe ser identificado es solo ligeramente diferente de la forma estándar del caso mínimo cuadrado, de hecho si los valores μ_i están dados, entonces está dado $\xi(x)$ de manera exacta a como se formula el caso de mínimo cuadrado estándar, ya que se puede comprender el vector $\xi(x)$ como un vector conocido de regresión.

Básicamente, los datos de entrenamiento, x^i , se mapean dentro de $\xi(x^i)$ y el algoritmo de mínimo cuadrado produce un estimado de los mejores centros para los centros de las funciones de pertenencia de la salida b_i

Todo lo que se debe hacer es sustituir los valores por los valores en la formación del vector Φ para el mínimo cuadrado por lote, por ejemplo.

Ideas similares se encuentran en (Takagi and Sugeno, 1985) para los sistemas difusos del tipo Takagi-Sugeno,

$$\mathbf{y} = \frac{\sum_{i=1}^R \mathbf{g}_i(\mathbf{x}) \mu_i(\mathbf{x})}{\sum_{i=1}^R \mu_i} \quad (1.42)$$

$$\text{Con } \mathbf{g}_i(\mathbf{x}) = \mathbf{a}_{i,0} + \mathbf{a}_{i,1}\mathbf{x}_1 + \dots + \mathbf{a}_{i,n}\mathbf{x}_n, \quad (1.43)$$

Algunos investigadores, tales como (Horikawa, Furuhashi, Uchikawa, 1992) y (Yen, Wang, and Wayne, 1998), han probado con éxito estimar los sistemas difusos mediante enfoques basados en redes neuronales.

CAPÍTULO II:

MODELADO DIFUSO DEL CALCULADOR DE PARÁMETROS

2.1 Identificación del modelo difuso

En este capítulo se examina el empleo de los sistemas difusos para la estimación. El problema básico a desarrollar aquí es ¿cómo construir un sistema en lógica difusa a partir de datos numéricos?

Este enfoque contrasta de manera radical los enfoques clásicos de empleo de los sistemas difusos para diseñar un controlador, dónde se usa la lingüística como el punto de partida para especificar un sistema difuso.

Si los datos numéricos son obtenidos partiendo de un experimento realizado sobre la planta como un conjunto o par ordenado de datos entrada/salida, se podrá identificar un modelo del sistema difuso de la planta. Esta es la hipótesis principal del problema.

Este modelo identificado, puede ser útil para satisfacer los propósitos de simulación y a veces para ser usado en un controlador, sin embargo, sirve además para la estimación de valores de parámetros de salida.

Por tanto, un sistema de lógica difusa puede ser usado para proporcionar una función no lineal parametrizada que ajusta los datos empleando sus capacidades básicas de interpolación, o sea el modelo difuso se presenta como un Aproximador de Función Universal.

Para hacer esto, se necesitan datos que muestren aproximadamente cómo debe comportarse el estimador partiendo del mapeo entrada/salida.

Una manera de generar estos datos es empezar estableciendo una prueba por simulación para la planta para la cual debe realizarse la estimación del o los parámetros. Así, puede planificarse una serie de experimentos por simulación, cada uno de ellos con un valor diferente para el parámetro a ser estimado.

Coordinando las condiciones del experimento y los datos generados por simulación con el valor del parámetro, se pueden recoger los pares de datos apropiados que permiten la construcción de un estimador difuso.

En (Passino and Yurkovish, 1998) se plantea al respecto que, para algunas plantas, será posible realizar este procedimiento con los propios datos reales experimentales.

De forma similar se podrán construir predictores difusos.

Un sistema lógico difuso convierte las reglas proporcionadas por el usuario a una forma matemática correspondiente. Esto no sólo simplifica el trabajo sino que también produce resultados más exactos. Estos sistemas son simples, flexibles y ellos pueden modelar una función no lineal compleja aun cuando los datos disponibles sean inexactos e incompletos (Zadeh, 1984). El sistema difuso puede ser diseñado para que coincida con cualquier conjunto dado de datos entrada/salida.

En los sistemas difusos, se ajustan los parámetros de las funciones de pertenencia buscando aproximar una función minimizando la función objetivo. Hay varios métodos usados para ajustar los parámetros, por ejemplo, los algoritmos genéticos (Cordon, Del Jesus and Herrera,1998), (Lee and Shin, 2003) y (Sánchez and Otero, 2004), los métodos del gradiente (Shi, Mizumoto, Yubazaki and Otani, 1996) y los algoritmos de mínimo cuadrados, etc.

Los métodos del gradiente simplemente dependen de las derivadas de las funciones y las derivadas muestran la pendiente de una función. Así, si la pendiente de una función puede ser identificada, entonces la función puede minimizarse para que converja hacia el punto mínimo (Dan, 2002).

En el proceso de identificación estos métodos se usan para minimizar la función del error buscando una mejor aproximación. Un gran número de trabajos pueden ser encontrados sobre las aplicaciones de los métodos del gradiente al modelado de los sistemas difusos, como ejemplo se citan (Kim, Park and Park. 1997), (Dan, 2005),

(Hatanaka, Kawaguchi and Uosaki, 2004), (Mohamed, 2003), (Oh, Pedrycz and Park, 2003) y (Toivonen and Makilaf, 1987)

Se presenta, en el numeral 2.1.1 un resumen de las diferentes maneras de realizar el método del gradiente los cuales son empleados para minimizar la función objetivo mediante entrenamiento de los parámetros del sistema difuso, ajustados para entregar la mejor aproximación.

Simulación del comportamiento del desbalance en dos planos para rotores de motores de inducción.

Uno de los grandes problemas a que se enfrentan los investigadores es la experimentación con los procesos reales. Generalmente estos experimentos no pueden ser efectuados con la duración requerida ni las repeticiones necesarias, puesto que el proceso es un proceso productivo y no de laboratorio.

Debido a esto los experimentos llevados a cabo con la máquina de balanceo en estudio fueron solo dos, o sea solo se tenían a disposición de la investigación dos rotores uno de los cuales es el rotor de prueba de la propia maquina. Los rotores que se balancean en esta máquina son rotores de máquinas que están en uso y no pueden ser desmontadas para realizar pruebas de experimentación alguna. Así, para obtener datos reales de desbalance de rotores se desbalanceó el rotor patrón de la máquina. El segundo o es un rotor de mayor dimensión que el rotor de prueba. Los datos de los rotores que pueden ser instalados para el balaceo se proporcionan en el Anexo 1.

Otro de los problemas que limita la realización de los experimentos es la prohibición de mantener en funcionamiento continuo la instalación durante un tiempo mayor a 4 minutos, esta prohibición es impuesta por el fabricante, lo que conlleva a que se realicen pruebas reales de forma intermitente, permitiendo así la conservación de la instalación.

Se dedica este acápite a la generación de datos por simulación para interpolar los pares de datos con los pares de datos obtenidos mediante experimentación y que permitan la construcción de un estimador difuso conociendo datos de experimentos siguiendo la manera de obtener la base de datos expuesta en Guillaume, 2001 y Passino y Yurkovich, 1998.

Para obtener la base de datos de la maquina MC-5 se realizó un experimento, el cual consistió en desconectar el bloque de adecuación de señales de la CPU de la maquina analizada y conectar su salida al instrumento Tektronix TDS320 y realizar las mediciones de amplitud de las señales adecuadas por el bloque y su retardo respecto de la señal del sensor estroboscópico.

Además, fueron empleados dos rotores diferentes el primero el rotor de prueba y el segundo de un motor trifásico con una configuración geométrica similar a la cilíndrica y el valor de la velocidad se tomó el valor de 300 rpm, acostumbrado por los operadores de esta máquina. Los valores de los datos recolectados de amplitud y retardo se muestran en la Tabla 2.1:

Tabla 2.1 Datos de experimentos

Rotor	F1	F2	$\Phi 1$	$\Phi 2$	Peso 1	Peso 2	Angulo 1	Angulo 2
Rotor 1	5.5	4	135°	60°	0.02473	0.01779	139.8°	230.7°
Rotor 2	9.5	14	280°	360°	0.04285	0.06345	90°	184°

Como es evidente, estos datos de los experimentos son relativamente pocos, por lo cual empleando el Método de los Coeficientes de Influencia, mediante interpolación de valores se logró alcanzar una cantidad mayor de datos, los cuales serán tomados para el entrenamiento de la lógica difusa.

Empleando los algoritmos de cálculo del peso y ángulo para balancear los rotores en dos planos por el método arriba mencionado, se desarrollan una serie de simulaciones. Las simulaciones han sido realizadas con el Diagrama Simulink y graficados sus resultados. El Diagrama Simulink se muestra en el Anexo 2.

En ellas se ha tenido en cuenta:

1. La variación de la amplitud de las señales de los sensores de fuerzas instalados en los pedestales de la máquina de balanceo de la firma Hoffman, (**F1, F2**).
2. La variación de los desplazamientos en fase (**$\Phi 1, \Phi 2$**) de los máximos de las señales tomadas de los sensores de fuerza con respecto a la marca arbitraria en el rotor, dada por el sensor estroboscópico.
3. La variación de las dimensiones geométricas de los rotores (**a, b, c, r1, r2**) que pueden ser instalados en los pedestales de la máquina de balanceo dinámico, siendo r1 y r2 los radios del rotor en los planos de balanceo.

En la Figura 2.1 se muestran los vectores F_1 , F_2 y los vectores correspondientes a las fuerzas de desbalance U_1 y U_2 ; se muestran además los ángulos de orientación de dichos vectores en rotación.

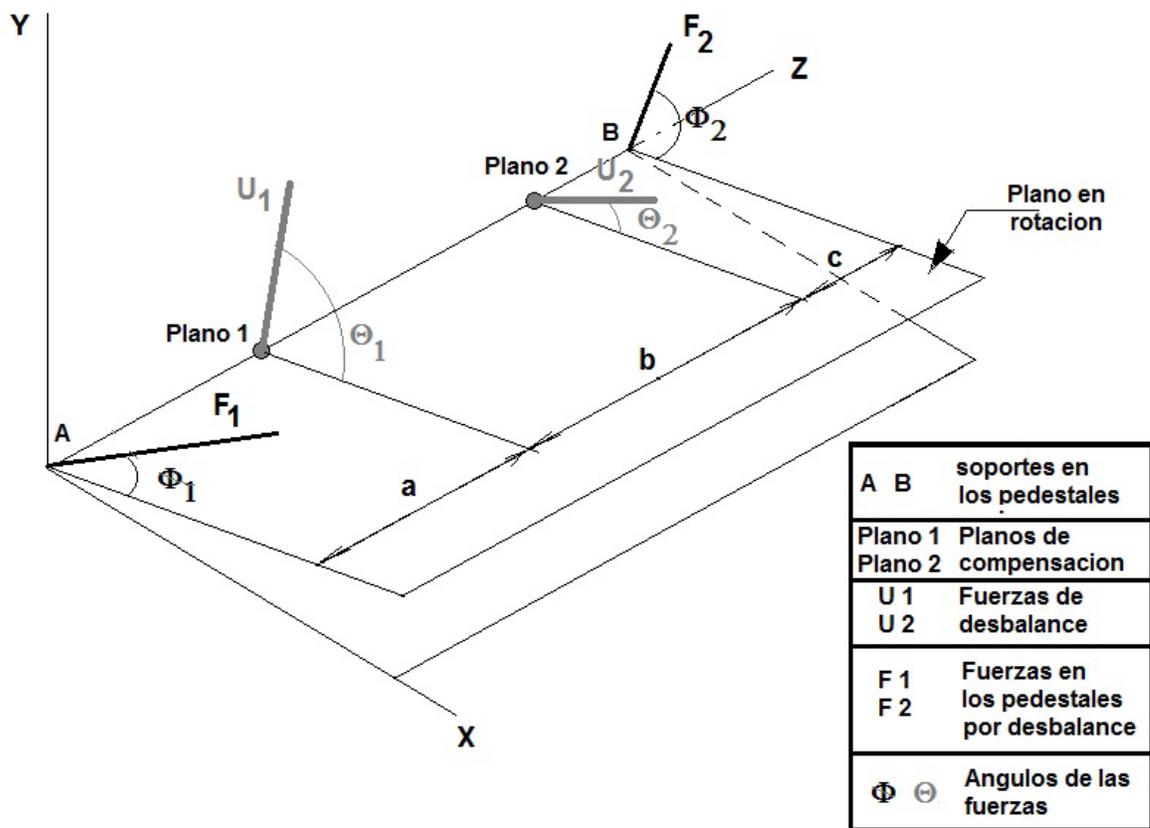


Figura 2.1 Vectores de fuerzas de desbalance y aplicados sobre los apoyos de los pedestales de la máquina de balanceo dinámico.

Considerando que, generalmente, los operadores de la máquina de balanceo establecen la velocidad de prueba como 300 rpm, para evitar coincidir con la velocidad crítica de los rotores en existencia, la misma se ha mantenido constante durante las simulaciones realizadas.

En la Figura 2.2 se muestra el resultado de la primera corrida, correspondiente a los valores de la columna **Prueba 1** de la Tabla 2.2.

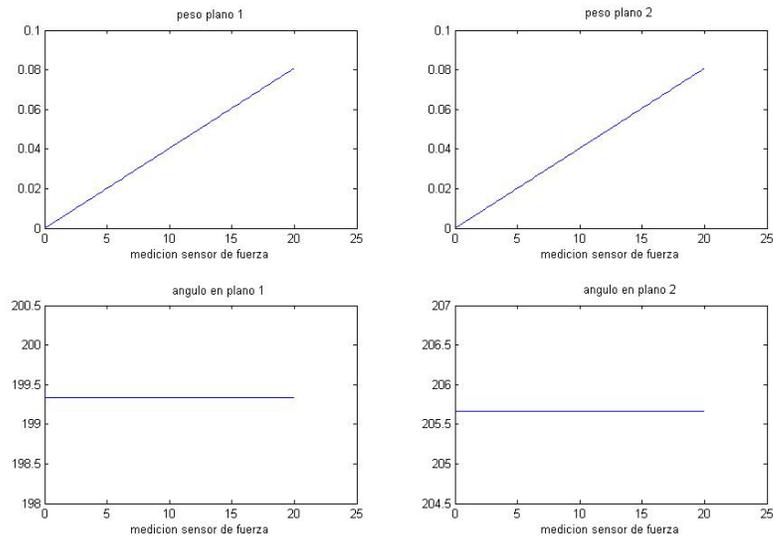


Figura 2.2. Resultado de la simulación de una corrida con un rotor de dimensiones dadas.

Los gráficos presentados reflejan el comportamiento del peso a introducir para balancear el rotor vs la amplitud de la vibración provocada por el desbalance y el ángulo donde debe ser introducido dicho peso vs la misma amplitud de la vibración.

Tabla 2.2 Relación de valores empleados para la interpolación de datos en las Pruebas 1 - 5

Señales	Valores por pruebas				
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
F₁	0.01-20	0.01-20	0.01-20	0.01-20	0.01-20
F₂	0.01-20	0.01-20	0.01-20	0.01-20	0.01-20
Φ₁	20°	30°	60°	45°	70°
Φ₂	25°	60°	30°	100°	220°
r₁	5.9	7.2	8	5.9	8
r₂	5.9	7.2	8	7.2	7.2
a	3.149	3.149	3.149	3.149	3.149
b	23.62	25.19	25.19	26.4	26.4
c	3.149	3.149	3.149	3.149	3.149

Los pesos resultantes a introducir están dados en gramos y presentan una dependencia ligeramente no lineal con respecto a las mediciones de las amplitudes de las vibraciones medidas.

La Figura 2.3 muestra los resultados superpuestos del resto de las pruebas 2, 3, 4 y 5. Una segunda corrida de simulaciones se realizó con los valores dados en la Tabla 2, donde lo más significativo es la variación de las estimaciones de los desplazamientos de los máximos de la señal del sensor de vibraciones respecto a la marca antes mencionada.

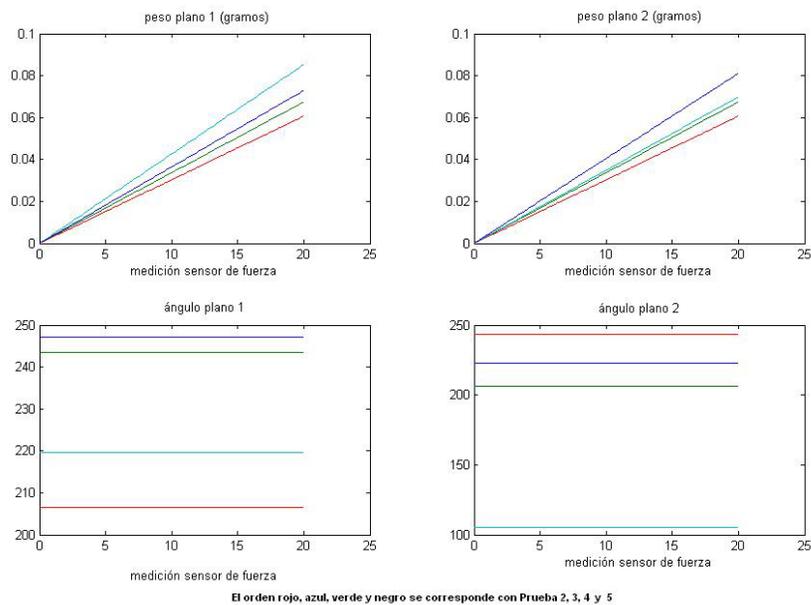


Figura 2.3 Resultados del resto de las simulaciones según orden de las columnas “Prueba”

Para ello se realizó un barrido entre los valores de 0° y 360° manteniendo siempre valores altos de los máximos de las señales de los sensores de vibraciones.

En las tres primeras pruebas (Pruebas 6, 7 y 8) se han supuesto rotores cuya geometría es axialmente uniforme y en las dos restantes, se ha variado un poco la forma del rotor a tipo cono truncado en los planos de corrección.

En la Figura 2.4 se muestran los resultados obtenidos con los datos de la Prueba 6. Se observa que para las variaciones de los ángulos de las masas desbalanceadoras existe un cambio brusco de los ángulos de ubicación de los pesos de corrección en función de tales variaciones.

Esto ocurre al sobre pasar, aquellos ángulos, el valor de 180° lo cual es evidente ya que los pesos correctores del desbalance deberán ser ubicados en el primer cuadrante.

Tabla 2.3 Relación de valores empleados para la interpolación de datos en las Pruebas 6 - 10

Señales	Valores por pruebas				
	Prueba 6	Prueba 7	Prueba 8	Prueba 9	Prueba 10
F₁	10	20	15	25	25
F₂	20	10	15	10	10
Φ₁	0° -360°	0° -360°	0° -360°	0° -360°	0° -360°
Φ₂	0° -360°	0° -360°	0° -360°	0° -360°	0° -360°
r₁	5.9	7.2	8	5.9	8
r₂	5.9	7.2	8	7.2	7.2
a	3.149	3.149	3.149	3.149	3.149
b	23.62	25.19	25.19	26.4	26.4
c	3.149	3.149	3.149	3.149	3.149

En la Figura 2.4, se muestran las simulaciones restantes realizadas:

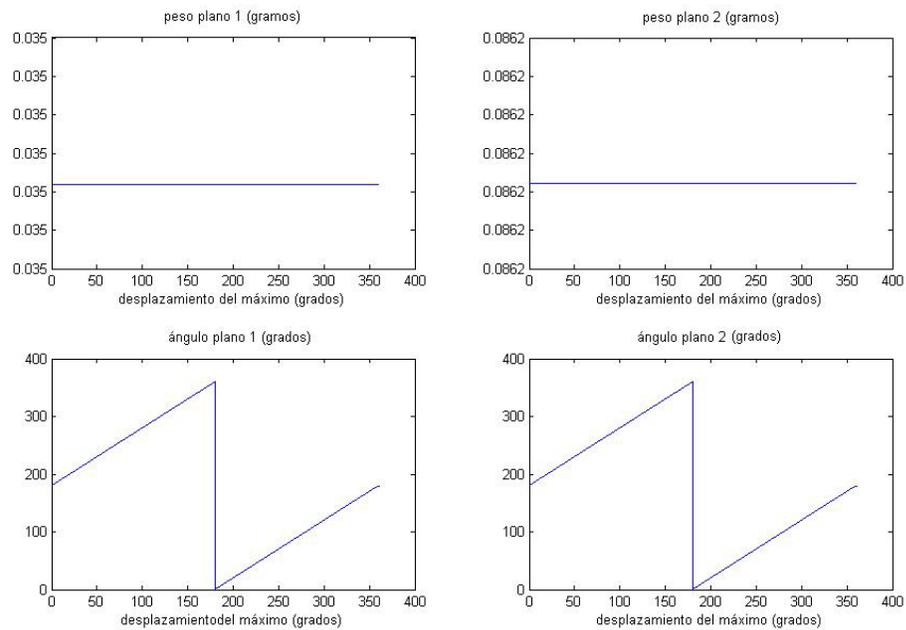


Figura 2.4. Resultados de la Prueba 6 de la Tabla 2.3.

En esta figura se muestra la variación de los pesos a introducir vs la variación del desplazamiento del máximo de las vibraciones respecto de la posición del pulso del sensor estroboscópico.

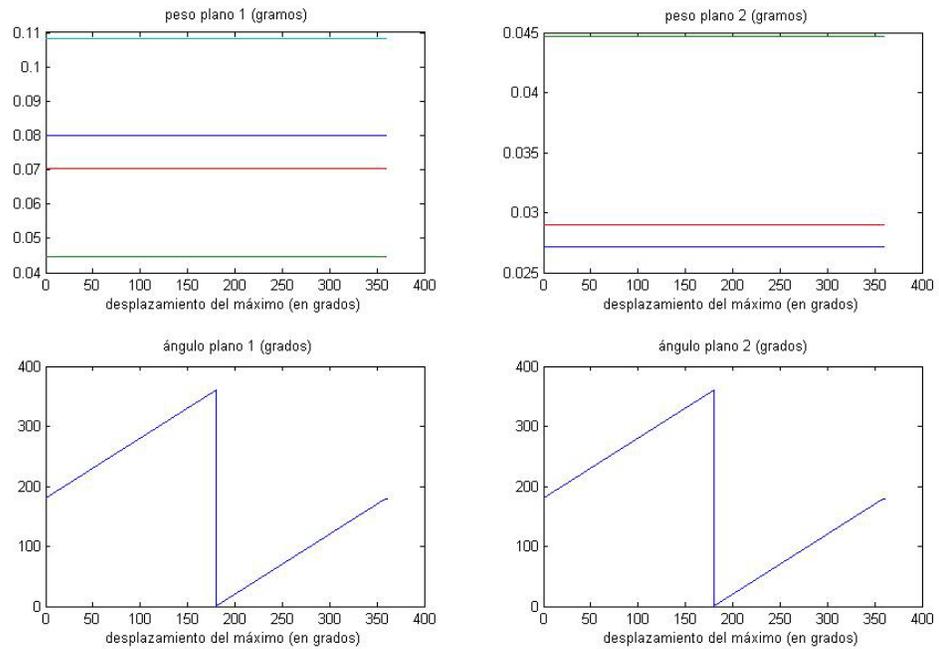


Figura 2.5. Resultados de las pruebas restantes de la Tabla 2.3 (Pruebas 7, 8, 9 y 10)

Se observa que:

- Las variaciones de los valores geométricos del rotor y de las fuerzas de desbalance prácticamente no determinan en la dependencia del ángulo de introducción de los pesos correctores, ya que todas las “curvas” de los ángulos en los planos 1 y 2 se encuentran solapadas unas sobre otra,
- Las variaciones de mayor influencia sobre los valores de los pesos a introducir se deben a las variaciones de los valores máximos de las vibraciones.

Una tercera corrida de simulaciones se realizó con los valores dados en la Tabla 2.4, donde lo más significativo corresponde a:

- Las variaciones de las estimaciones de los desplazamientos de los máximos de la señal del sensor de vibraciones respecto a la marca antes mencionada.

- Las variaciones de los valores máximos de la señal del sensor de vibraciones.

Para ello se realizó el mismo barrido por ángulo, entre los valores de 0° y 360° el mismo barrido por magnitud de las vibraciones empleado en la primera simulación. O sea, se pretende ahora determinar la influencia conjunta de ambas variaciones de las señales de interés para el balanceo dinámico de rotores.

En las tres primeras pruebas (Pruebas 6, 7 y 8) se han supuesto rotores cuya geometría es axialmente uniforme y en las dos restantes, se ha variado un poco la forma del rotor a tipo cono truncado en los planos de corrección.

Tabla 2.4 Relación de valores empleados para la interpolación de datos en las Pruebas 11-15

Señales	Valores por pruebas				
	Prueba 11	Prueba 12	Prueba 13	Prueba 14	Prueba 15
F₁	0.01-20	0.01-20	0.01-20	0.01-20	0.01-20
F₂	0.01-20	0.01-20	0.01-20	0.01-20	0.01-20
Φ₁	$0^\circ - 360^\circ$				
Φ₂	$0^\circ - 360^\circ$				
r₁	5.9	7.2	8	5.9	8
r₂	5.9	7.2	8	7.2	7.2
a	3.149	3.149	3.149	3.149	3.149
b	23.62	25.19	25.19	26.4	26.4
c	3.149	3.149	3.149	3.149	3.149

En la Figura 2.6 se muestran la influencia de las variaciones del valor máximo de la vibración en el pedestal 1 y de la posición angular de la masa desbalanceadora sobre la posición angular del peso de compensación del desbalance en el plano 1.

En la Figura 2.7 se muestran la influencia de las variaciones del valor máximo de la vibración en el pedestal 1 y de la posición angular de la masa desbalanceadora sobre el valor del peso de compensación del desbalance en el plano 1.

En la Figura 2.8 se muestran la influencia de las variaciones del valor máximo de la vibración en el pedestal 2 y de la posición angular de la masa desbalanceadora sobre la posición angular del peso de compensación del desbalance en el plano 2.

En la Figura 2.9 se muestran la influencia de las variaciones del valor máximo de la vibración en el pedestal 2 y de la posición angular de la masa desbalanceadora sobre el valor del peso de compensación del desbalance en el plano 2.

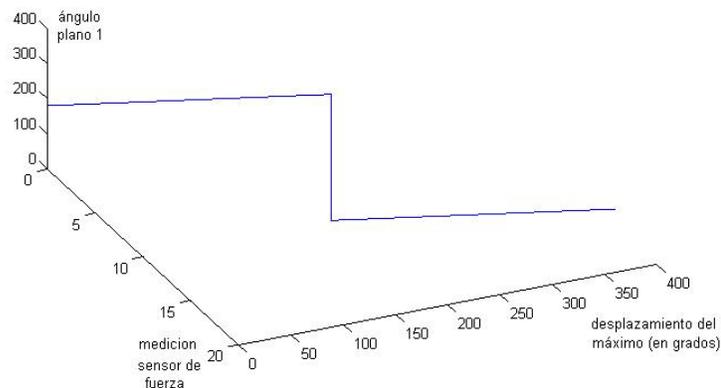


Figura 2.6 Representación del “ángulo 1” en función de dos variables, amplitud de la vibración y desplazamiento del máximo

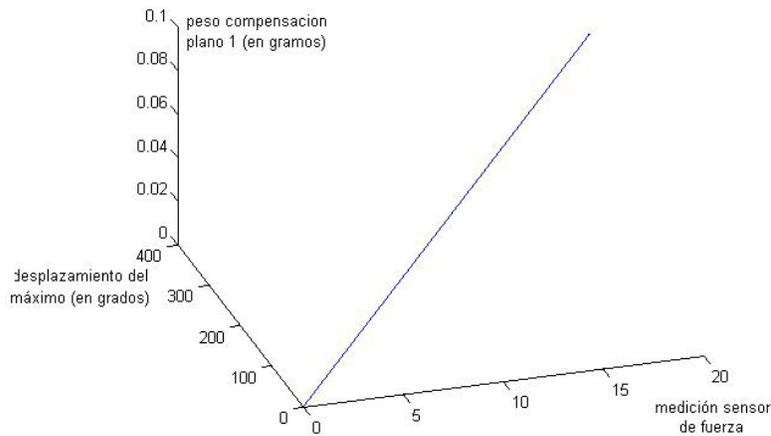


Figura 2.7 Representación del “peso 1” en función de dos variables, amplitud de la vibración y desplazamiento del máximo

De las Figuras 2.6 y 2.7 se observa que:

1. Los valores de los pesos a introducir en los planos de balanceo dinámico son funciones de la amplitud máxima de las vibraciones y del desplazamiento de estos valores máximos de la marca dada por el sensor estroboscópico,
2. Los valores de los ángulos donde deben ser introducidos estos pesos también son funciones, en este con discontinuidad en el valor 180° , de la amplitud máxima de las vibraciones y del desplazamiento de estos valores máximos de la marca dada por el sensor estroboscópico,
3. Si son funciones, entonces es posible diseñar un aproximador de funciones,
4. Para diseñar una lógica difusa que determine el ángulo y el peso a introducir en el plano de compensación del des balance será necesario dividir los rangos de variación del máximo de la vibración y del desplazamiento de dicho máximo, o sea, los universos de discurso, en valores lingüísticos.

Como se analizó en Estado del Arte, la cantidad de valores lingüísticos (partición del espacio, Guillaume, 2001) puede ser determinada quitando o agregando conjuntos difusos y poniendo a punto los parámetros de las funciones de pertenencia”

En el diseño de controladores difusos es una práctica aceptada comenzar el diseño partiendo de tres funciones de pertenencia o valores lingüísticos, como mínimo, para las variables. Sin embargo, para el diseño de Aproximadores de Funciones no se establecen límites y esto se deja a la determinación por programas de optimización del conjunto de reglas.

De las Figuras 2.8 y 2.9 se observa que una lógica difusa puede ser diseñada, de manera que determine el ángulo y el peso a introducir en el plano de compensación 2 dividiendo los rangos de variación del máximo de la vibración y del desplazamiento de dicho máximo, o sea, los universos de discurso, en valores lingüísticos similares al caso anterior.

Por lo cual, para el primer universo de discurso, medición del máximo mediante el sensor de vibraciones en el pedestal 2, se proponen comenzar el diseño a partir de la ubicación de los centros en los valores 5, 10 y 15. Para el segundo universo de discurso, desplazamiento del máximo de las vibraciones, se proponen los centros en los valores 90° y 270° .

Por último, la simulación empleando el Método de Coeficientes de Influencia permite no solo encontrar valores interpolados de los datos de experimentos, sino también verificar el carácter funcional de las dependencias peso balanceador/ amplitud de vibraciones y

desplazamiento del máximo; ángulo de introducción del peso/ amplitud de vibraciones y desplazamiento del máximo, lo cual es el fundamento metodológico para el diseño de los Aproximadores Universales de Funciones.

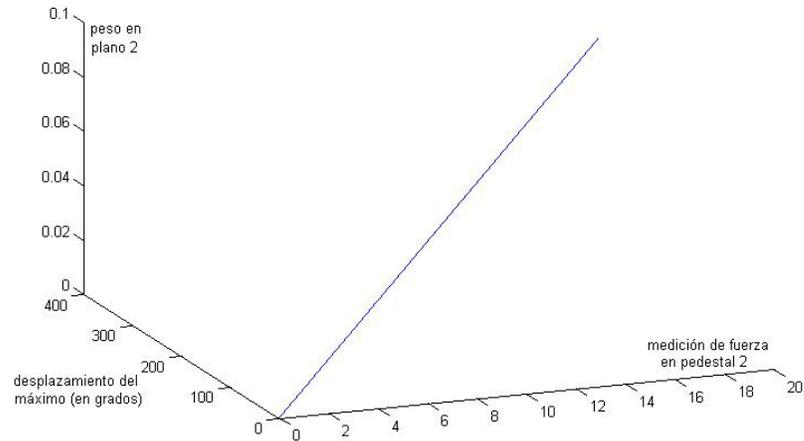


Figura 2.8 Representación del “peso 2” en función de dos variables, amplitud de la vibración y desplazamiento del máximo

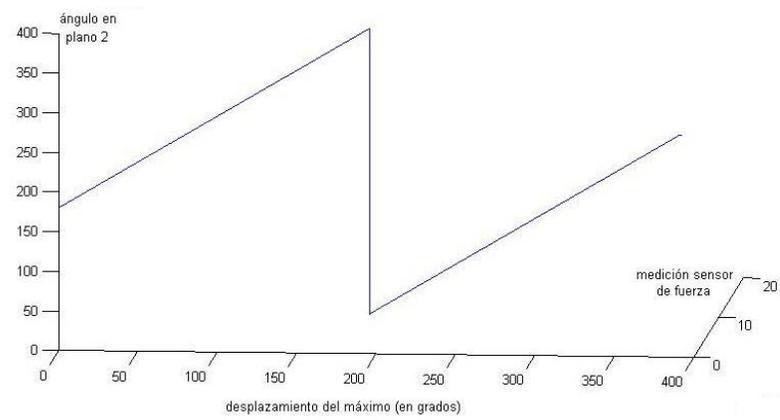


Figura 2.9 Representación del “ángulo 2” en función de dos variables, amplitud de la vibración y desplazamiento del máximo

Estos valores de los centros son valores iniciales, por lo cual deberán ser ajustados mediante entrenamiento o ajuste de la lógica difusa.

Como resultado de las mediciones realizadas con dos rotores diferentes, y la corrida del programa simulink, mostrado en el anexo 2, se obtuvo la base de datos que se muestra a continuación:

Tabla 2.5 Base de Datos

	Amp (F)	Desp (Φ)	Peso (P)	Angulo (Θ)
1	0.5	1	0.002	181
2	1	5	0.004042	185
3	2	30	0.008084	210
4	3	45	0.01213	225
5	4	60	0.01617	240
6	4.5	70	0.01819	250
7	5	90	0.02021	270
8	5.5	135	0.02223	315
9	6	180	0.02425	360
10	6.5	175	0.02627	355
11	7	200	0.02831	20
12	7.5	210	0.03032	30
13	8	250	0.03234	70
14	8.5	240	0.03436	60
15	9	300	0.03638	120
16	9.5	280	0.0384	100
17	10	310	0.04042	140
18	11	320	0.04446	130
19	15	360	0.060631	180
20	14	340	0.056591	160

Esta base de datos se dividió en dos partes: 1) Base de datos de entrenamiento, cuyos valores se muestran en color **rojo**, y 2) Base de datos de validación de la lógica difusa entrenada, cuyos valores se muestran en color **azul**.

2.1.1 Métodos del Gradiente para entrenamiento de sistemas difusos estándares

Los métodos del gradiente requieren del cálculo de la primera o segunda derivada de la función objetivo con respecto a aquellos parámetros que serán ajustados.

Los métodos que desarrollan las ideas del Método del gradiente son:

- Método del Descenso más Rápido.
- Método de Newton.
- Método Cuasi Newton.
- Método de Levenberg/Marquardt para el Método de Cuasi Newton.
- Método de Gauss/Newton.
- Método de Levenberg/Marquardt para el Método de Gauss/Newton.

Se supone en lo adelante que el sistema difuso está dado por la función:

$$f(x|\rho_j) = \frac{\sum_{i=1}^L \beta_i \mu_i(x)}{\sum_{i=1}^L \mu_i(x)} \quad (2.1)$$

Donde

$$\mu_i(x) = \prod_{j=1}^l \exp\left(-\frac{1}{2} \left(\frac{x_i - c_j^i}{\sigma_j^i}\right)^2\right) \quad (2.2)$$

O sea un sistema clásico difuso.

Método del Descenso Más Rápido

Este es uno de los primeros métodos empleados para la minimización de la función de coste. Los métodos que emplean el gradiente se fundamentan en la asignación de un valor inicial para los parámetros buscados con posterior actualización del valor

asignado buscando minimizar la función mencionada. La formulación matemática de estos métodos es como sigue:

$$r(k+1)=r(k)-\gamma g(k) \quad (2.3)$$

Siendo γ la razón de aprendizaje, la cual, para una rápida convergencia toma valores positivos pequeños. $g(k)$ es el gradiente de la función de coste y k es el índice de iteración.

$$g(k) = \frac{\partial e_m}{\partial \rho} = [f(x^m|\rho) - \tau^m] * \frac{\partial e_m}{\partial \rho} \quad (2.4)$$

$$g(k) = \frac{\partial e_m}{\partial \rho} = -\varepsilon * \frac{\partial e_m}{\partial \rho} \quad (2.5)$$

Los sistemas difusos más empleados para la estimación de parámetros son aquellos que tienen tres parámetros, a saber, la posición de los centros de las funciones de pertenencia de salida, β_i ; el ancho, σ_j^i , y las posiciones de los centros, c_j^i de las funciones de pertenencia de entrada del sistema.

Empleando las expresiones (2.3) y (2.5) pueden ser actualizados todos estos parámetros:

La ley de actualización de los centros de las funciones de pertenencia de salida (β_i) es:

$$\beta_i(k + 1) = \beta_i(k) - \gamma \frac{\partial e_m}{\partial \beta_i} \quad (2.6)$$

$$\frac{\partial e_m}{\partial \beta_i} = \varepsilon * \frac{\prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_i - c_j^i}{\sigma_j^i}\right)^2\right)}{\sum_{i=1}^l \prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_i - c_j^i}{\sigma_j^i}\right)^2\right)} \quad (2.7)$$

Empleando (2.7) en (2.6), la ley final de actualización para las funciones de pertenencia de salida es:

$$\beta_i(k+1) = \beta_i(k) - \gamma \varepsilon * \frac{\prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_i - c_j^i}{\sigma_j^i}\right)^2\right)}{\sum_{i=1}^l \prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_i - c_j^i}{\sigma_j^i}\right)^2\right)} \quad (2.8)$$

La ley de actualización de los anchos (σ_j^i) de las funciones de pertenencia de entrada se encuentra como sigue:

$$\sigma_i(k+1) = \sigma_i(k) - \gamma \frac{\partial e_m}{\partial \sigma_i} \quad (2.9)$$

$$\frac{\partial e_m}{\partial \sigma_i} = \varepsilon * \frac{\partial}{\partial \sigma_i} \frac{\sum_{i=1}^l \beta_i \prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_j^m - c_j^i}{\sigma_j^i}\right)^2\right)}{\sum_{i=1}^l \prod_{j=1}^l \exp\left(-\frac{1}{2}\left(\frac{x_j^m - c_j^i}{\sigma_j^i}\right)^2\right)} \quad (2.10)$$

Diferenciando y simplificando (2.10):

$$\frac{\partial e_m}{\partial \sigma_i} = \varepsilon * \left[\frac{\sum_{i=1}^l \beta_i \frac{\sum_{i=1}^l \beta_i * \mu_i(x^m)}{\sum_{i=1}^l \mu_i(x^m)}}{\sum_{i=1}^l \mu_i(x^m)} \right] * \left[\frac{(x_j^m - c_j^i)^2}{(\sigma_j^i)^3} \right] * \mu_i(x^m) \quad (2.11)$$

Empleando (2.11) y (2.9):

$$\sigma_i(k+1) = \sigma_i(k) - \gamma \varepsilon * \left[\frac{\sum_{i=1}^L \beta_i - \frac{\sum_{i=1}^L \beta_i * \mu_i(x^m)}{\sum_{i=1}^L \mu_i(x^m)}}{\sum_{i=1}^L \mu_i(x^m)} \right] * \left[\frac{(x_j^m - c_j^i)^2}{(\sigma_j^i)^3} \right] * \mu_i(x^m) \quad (2.12)$$

Para actualizar los centros de las funciones de pertenencia de entrada:

$$c_i(k+1) = c_i(k) - \gamma \frac{\partial e_m}{\partial c_i}$$

$$\frac{\partial e_m}{\partial c_i} = \varepsilon * \frac{\partial}{\partial c_i} \frac{\sum_{i=1}^L \beta_i \prod_{j=1}^l \exp\left(-\frac{1}{2} \left(\frac{x_j^m - c_j^i}{\sigma_j^i}\right)^2\right)}{\sum_{i=1}^L \prod_{j=1}^l \exp\left(-\frac{1}{2} \left(\frac{x_j^m - c_j^i}{\sigma_j^i}\right)^2\right)} \quad (2.13)$$

Derivando y simplificando (2.13) se tiene:

$$\frac{\partial e_m}{\partial c_i} = \varepsilon * \left[\frac{\sum_{i=1}^L \beta_i - \frac{\sum_{i=1}^L \beta_i * \mu_i(x^m)}{\sum_{i=1}^L \mu_i(x^m)}}{\sum_{i=1}^L \mu_i(x^m)} \right] * \left[\frac{(x_j^m - c_j^i)}{(\sigma_j^i)^2} \right] * \mu_i(x^m) \quad (2.14)$$

Entonces, la relación para actualizar los centros de las funciones de pertenencia de entrada es:

$$c_i(k+1) = c_i(k) - \gamma \varepsilon * \left[\frac{\sum_{i=1}^L \beta_i - \frac{\sum_{i=1}^L \beta_i * \mu_i(x^m)}{\sum_{i=1}^L \mu_i(x^m)}}{\sum_{i=1}^L \mu_i(x^m)} \right] * \left[\frac{(x_j^m - c_j^i)}{(\sigma_j^i)^2} \right] * \mu_i(x^m) \quad (2.15)$$

Así, las ecuaciones (2.8), (2.12) y (2.15) son las ecuaciones para la actualización de los parámetros del sistema clásico difuso empleado como aproximador universal a partir de datos experimentales.

Estas ecuaciones, además, son las que se emplean para programar el entrenamiento del sistema lógico difuso, o sea, para buscar los parámetros del sistema difuso que mejor estiman los datos de entrenamiento y que podrán proporcionar estimaciones con el mínimo error dados otros datos.

Método de Newton

Este método se basa en la expansión de segundo orden de la serie de Taylor. La idea básica que subyace en este método de Newton es aproximar las funciones cuadráticas y procede para encontrar el mínimo de la función cuadrática en un paso.

Matemáticamente, la expresión para la actualización se representa como:

$$\rho(k + 1) = \rho(k) - H^{-1}g_k \quad (2.16)$$

Donde, H es la matriz Hessiana, la cual consiste en las derivadas parciales de segundo orden de la función de coste. En (Khan, L., Anjum, S. and Badar, R. Standard Fuzzy Model Identification using Gradient Methods. World Applied Sciences Journal 8. 2010. ISSN 1818-4952) se pone de manifiesto que si la función no es cuadrática este método es menos eficaz y no convergerá en un paso.

Subsecuentemente, este método depende completamente de la función y los supuestos iniciales de los parámetros, por lo que no hay ninguna seguridad de su convergencia. Es más, con el aumento del número de parámetros a identificar aumenta la dimensión de la matriz Hessiana y el cálculo se hace más complejo.

Para ajustar los parámetros, se calcula primeramente la matriz Hessiana H y después su inversa. En [Passino and Yurkovich, 1998] se desarrolla una manera de calcular la matriz Hessiana y las actualizaciones de los parámetros.

Método Cuasi Newton

El concepto de Cuasi-Newton surge para hacer más eficiente y rápido el método de Newton.

Una selección apropiada de la razón de aprendizaje η hace que la aproximación ocurra más rápido. Sin embargo no hay mucha diferencia entre los métodos de Newton y Cuasi-Newton, aunque este último busca la respuesta deseada de manera más rápida

Levenberg Marquart para el Método Cuasi Newton

Este método resuelve los serios problemas que tienen los anteriores que es que cuando la matriz es singular no es invertible. Un valor adecuado de λI agregado a la matriz Hessiana la convierte en no singular y por tanto invertible. Esto se escribe como:

$$\rho(k + 1) = \rho(k) - \gamma(H + \lambda I)^{-1}g_k \quad (2.17)$$

Método de Gauss Newton

Este método también recibe el nombre de Método de Linealización. Emplea la expansión en serie de Taylor para obtener un modelo lineal que estime al modelo no lineal original. Matemáticamente se expresa como sigue:

$$\rho(k + 1) = \rho(k) - \gamma(J^T J)^{-1} J \quad (2.18)$$

Siendo J la matriz Jacobiana conocida también como gradiente. Se debe calcular el gradiente de todos los parámetros para ser empleados en (2.18).

Levenberg Marquardt dieron una modificación de este método agregando un factor para cubrir el problema de la singularidad. En (Passino and Yurkovich, 1998) se proporcionan las pautas para desarrollar este método.

En [Khan, Anjum y Badar, 2010] se emplean diferentes métodos del gradiente para estimar las señales de referencia empleando el modelo difuso estándar. Ellos observaron que la eficacia de las técnicas se reduce con la complejidad. Mientras mayor sea la complejidad del algoritmo mejor será la estimación a costa de la velocidad. Constataron que los valores de los parámetros usados en los métodos del gradiente deben ser ajustados convenientemente para trabajar correctamente.

2.1.2 Cuestiones a considerar en el diseño del algoritmo de entrenamiento de los sistemas difusos

En esta sección se discuten varios problemas que se encontrarán al llevar a cabo el enfoque del gradiente para entrenar los sistemas difusos.

Problemas en el diseño del algoritmo

Hay varios problemas a resolver en el diseño del entrenamiento de un sistema difuso mediante el algoritmo del gradiente. Como siempre, la elección de los datos de entrenamiento G es crítica. Luego, note que se debe escoger n el número de entradas del sistema difuso a ser entrenado y el número de reglas R . ***El método no adiciona reglas, solo pone a punto las existentes.***

La elección del estimado inicial para los parámetros $\beta_i(\mathbf{0})$, $c_{ij}(\mathbf{0})$, y $\sigma_{ij}(\mathbf{0})$ puede ser importante. A veces escogiéndolos cerca de dónde ellos deben estar puede ayudar a la convergencia.

Nótese que puede no ser conveniente escoger $\beta_i = \mathbf{0}$ para todo $i = 1, 2, \dots, R$ ya que puede que el algoritmo para ajustar β_i se quede en el valor cero para todo $k \geq \mathbf{0}$.

El lenguaje de programación a emplear no permitirá escoger $\sigma_{ij}(\mathbf{0}) = \mathbf{0}$ ya que está la expresión de dividir por esta cantidad en el algoritmo de actualización del parámetro, como se indica en las expresiones (2.11), (2.12), (2.14) y (2.15).

Note que, generalmente, la elección de las funciones de pertenencia de entrada es libre ya que en el algoritmo de actualización no se realiza la división por ella.

Además, los algoritmos anteriores del gradiente son sólo para **un par de datos de entrenamiento**. Esto es, se puede ejecutar el algoritmo durante mucho tiempo (es decir, para muchos valores de k) para un sólo par de datos intentando entrenar el sistema difuso de tal forma que haga coincidir muy bien la respuesta del sistema difuso con ese par de datos.

Entonces se puede pasar al próximo par de datos del conjunto de datos de entrenamiento, empezando con los últimos valores calculados para β_i , c_{ij} , y σ_{ij} considerándolos como valores iniciales por el nuevo par de datos y ejecutar el algoritmo del gradiente para tantos pasos como se desee para este nuevo par de datos y así sucesivamente.

Alternativamente, se puede avanzar muchas veces a través de los datos de entrenamiento, tomando en cada paso el algoritmo del gradiente para cada par de datos.

Es difícil saber cuántos pasos de actualización de parámetros se requieren. Sin embargo, si se emplean algunos datos de entrada con más frecuencia que otros, entonces el sistema difuso entrenado tenderá a ser más exacto para los datos empleados con más frecuencia que para los datos que se usaron con menor frecuencia durante el entrenamiento.

Hay autores que prefieren recorrer todos los datos para que cada par de datos se visite el mismo número de veces y emplear pequeñas dimensiones del paso para que las actualizaciones no sean demasiado grandes en cualquier dirección.

Es claro que se debe tener cuidado en la elección del valor de λ_i , $i = 1, 2, 3$ ya que los valores demasiado grandes pueden producir un algoritmo inestable (es decir, los valores de los parámetros pueden oscilar o pueden hacerse ilimitados). Por otro lado, los valores demasiado pequeños pueden producir una convergencia muy lenta.

El problema principal, sin embargo, es que, ¡en el caso general, no hay ninguna garantía de que el algoritmo del gradiente converja en absoluto! Es más, se puede necesitar una cantidad grande de datos de entrenamiento y mucho tiempo de entrenamiento para lograr buenos resultados.

Generalmente, se pueden hacer algunas pruebas para ver qué bien se está construyendo el sistema difuso comparando la salida para los pares de datos con sus valores reales; sin embargo, aun cuando esta comparación aparente indica que el sistema difuso está trazando bien los datos, no hay ninguna garantía teórica de que “generalizará” (es decir, interpolará) para los datos que no estaban en el conjunto de datos de entrenamiento con el cual fue entrenado.

Para terminar el algoritmo del gradiente, se puede esperar a que todos los valores de los parámetros cesen de cambiar o a que el cambio sea muy pequeño de un paso a otro de actualización. Esto indica que los parámetros no están actualizándose y que los gradientes son pequeños, debiéndose estar en un mínimo de la superficie del error.

De otra manera, se puede esperar hasta que el error o su suma no varíen durante un número fijo de pasos. Esto indicará que aun cuando los valores del parámetro están cambiando, el valor del error no es decreciente, y que el algoritmo ha encontrado un mínimo y puede detenerse el entrenamiento.

2.1.3 Entrenamiento de los sistemas difusos de Takagi-Sugeno

Los sistemas difusos de Takagi-Sugeno, [Takagi and Sugeno, 1985.], toman la forma:

$$f(x|\theta(k)) = \frac{\sum_{i=1}^R g_i(x,k)\mu_i(x,k)}{\sum_{i=1}^R \mu_i(x,k)} \quad (2.19)$$

Donde $\mu_i(x, k)$ se define por la fórmula (2.20):

$$\mu_i(x^m, k) = \prod_{j=1}^n \exp\left(-\frac{1}{2}\left(\frac{x_j^m - c_j^i(k)}{\sigma_j^i(k)}\right)^2\right) \quad (2.20)$$

$$x = [x_1, x_2, \dots, x_n]^T \quad (2.21)$$

$$g_i(x, k) = a_{i,0}(k) + a_{i,1}(k)x_1 + a_{i,2}(k)x_2 + \dots + a_{i,n}(k)x_n \quad (2.22)$$

El sub índice k se ha agregado ya que, ahora, se actualizarán los coeficientes o parámetros $a_{i,j}$, c_j^i y σ_j^i

Fórmulas para actualizar los parámetros

Siguiendo el mismo enfoque presentado en 2.1.1 se procede a plantear las ecuaciones de actualización de los parámetros del modelo difuso de Takagi-Sugeno. De (2.20) y (2.22) se conoce que los parámetros a actualizar son;

- c_j^i y σ_j^i para adecuar la función $\mu_i(x^m, k)$
- $a_{i,j}$ para adecuar el modelo de Takagi-Sugeno

Si se observa detenidamente, se facilita el trabajo si se emplean fórmulas similares a las fórmulas (2.12) y (2.15). La manera de hacerlo y un ejemplo típico se ofrece en [Passino and Yurkovich, 1998].

Para actualizar los coeficientes de los polinomios de Takagi-Sugeno [Passino and Yurkovich, 1998] proponen las siguientes expresiones:

$$a_{ij}(k+1) = a_{ij}(k) - \lambda \frac{\partial e}{\partial a_{ij}} \quad (2.23)$$

$$\frac{\partial e}{\partial a_{ij}} = e_m(k) \frac{\partial f(x, \theta, k)}{\partial a_{ij}} \quad (2.24)$$

$$\frac{\partial f(x, \theta, k)}{\partial a_{ij}} = \frac{\mu_i(x^m, k)}{\sum_{i=1}^L \mu_i(x^m, k)} \frac{\partial g_i(x^m, k)}{\partial a_{ij}} \quad (2.25)$$

Con:

$$\frac{\partial g_i(x^m, k)}{\partial a_{i0}} = 1 \quad (2.26)$$

$$\frac{\partial g_i(x^m, k)}{\partial a_{ij}} = x_j \quad (2.27)$$

2.1.4 Programación en lenguaje MATLAB

Se pretende, con este desarrollo, adelantar algunas funciones para aterrizar la teoría expuesta con antelación.

En el entrenamiento de la lógica difusa se ha pensado como una actividad recurrente hasta encontrar los parámetros de las funciones de pertenencia de las entradas y las constantes de los consecuentes, siguiendo el diagrama de flujo mostrado en la figura 2.10.

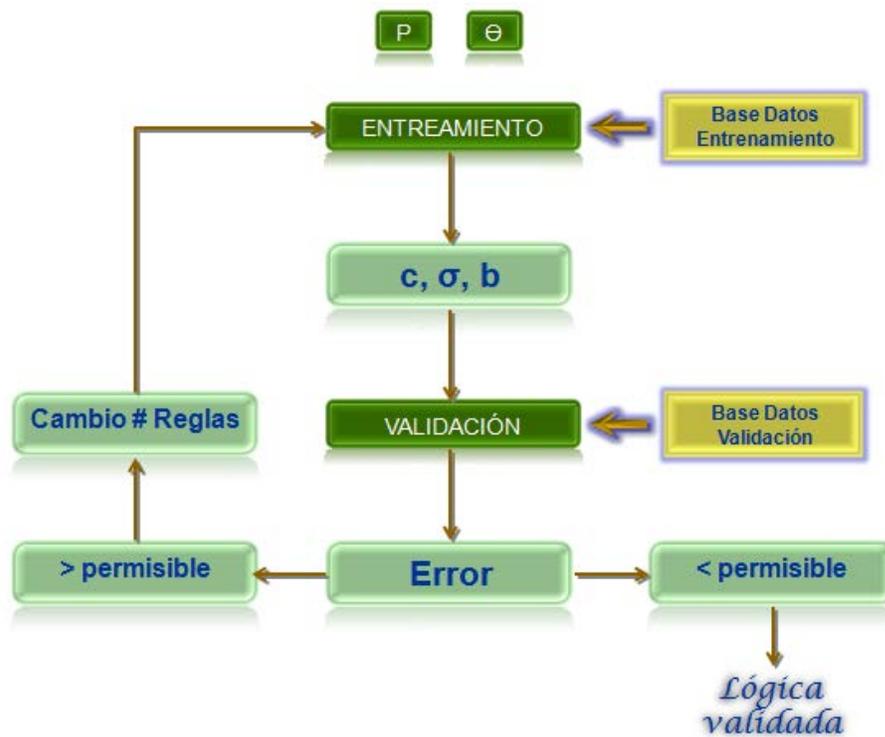


Figura 2.10 Diagrama de flujo del entrenamiento y validación

La figura 2.10 muestra además, que el proceso de repetición del entrenamiento sólo ocurre si el proceso de validación no satisface el error permisible. En ese caso, se propone una nueva estructura (aumento de funciones de pertenencia en los universos de discurso) para un nuevo entrenamiento. Si el error en la validación es menor que el permisible, se acepta lógica difusa, con lo cual se acepta la última estructura comprobada.

Para obtener los valores de la expresión (2.19) se crea una función de MATLAB (MATLAB function) denominada **sborr**.

```
function [y,b,zl]=sborr(u,y1,xl,s1)

l=max(size(y1));
zl=prod(exp(-((ones(1,1)*u'-xl)./s1).^2),2); % cálculo de la fórmula
%(2.20)
b=sum(zl);
y=y1'*zl/b; % cálculo de la fórmula (2.19)
```

Ella tiene como argumento de la función las variables de entrada:

u – entrada al modelo difuso (fuerza y ángulo)
y1 – vector del parámetro “consecuentes constantes de Sugeno”
xl – centros de las funciones de pertenencia de entrada
s1 – ancho de las funciones de pertenencia de entrada

Esta función devuelve, al programa principal, las variables de salida:

y – salida de la lógica difusa
zl – evaluación de los antecedentes
b - denominador

Estas variables serán empleadas para:

- Encontrar el error durante el entrenamiento de la lógica difusa
- En el cálculo de los gradientes para la actualización de los coeficientes del consecuente y los centros y anchos de las funciones de pertenencia de las dos entradas: fuerza y ángulo.

Después, en el mismo programa se emplean datos de validación del modelo identificado donde nuevamente se emplea la función **sborr** con los parámetros identificados y datos frescos para validar.

2.2 Determinación de las funciones de pertenencia en el antecedente y en el consecuente, de la ubicación de los centros de las funciones de pertenencia de las operaciones difusas

Las funciones de pertenencia en los sistemas de lógica difusa pueden tener diferentes formas. Se pueden emplear del tipo triangular, del tipo campana de Gauss, trapezoidales, truncadas, etc. No existe una guía específica para la elección de las funciones de pertenencia en la lógica difusa, generalmente las mismas se escogen en dependencia de la aplicación.

En la presente tesis se propone el empleo de las funciones de pertenencia del tipo campana de Gauss, ya que para el entrenamiento de la lógica difusa se requiere realizar la diferenciación de las funciones y esta función de Gauss es muy fácil de diferenciar, dada la presencia en ella de una función exponencial.

Los valores finales de los centros y anchos de dicha función se escogerán por entrenamiento, o sea, estos valores de los centros y anchos de las funciones de pertenencia de las variables de entrada se hará mediante el entrenamiento de la lógica difusa, de tal forma que el error de reproducción de los datos de salida del algoritmo de cálculo del método de los coeficientes sea mínimo, con lo cual se logra el algoritmo difuso óptimo por ubicación de los centros de las funciones y sus anchos. Esta elección de anchos y centros se realiza por programación empleando los datos calculados por el mencionado algoritmo.

Considerando la ausencia de publicaciones que referencien programas detallados sobre el entrenamiento de lógicas difusas, a continuación se expone, con ciertos detalles, la programación realizada para tales fines

El programa consta de varias partes o secciones. La primera involucra la declaración de los datos y la creación de los arreglos de los centros y anchos de las funciones de pertenencia de las variables de entrada. Esta parte se muestra a continuación.

```
%%%%%%%%% INICIALIZACION CENTROS, ANCHOS Y SALIDA %%%%%%%%%%%%%%
AW=[2 2 2];%3 anchos fp fuerza
CW=[2.5 7.5 12.5];%3 centros fp fuerza
AQ=[4 4];%2 anchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 6
%inicializacion valores de salida logica borrosa
Y1=[0.002 0.0162 0.03 0.032 0.04 0.06]';

%%%%%%%%%%%%%% SE FORMAN ARREGLOS CON CENTROS Y ANCHOS DE
FP%%%%%%%%%%%%%%

Centros=[];
c=1;
for i=1:3
    for j=1:2
        Centros(:,c) = [CW(i);CQ(j)];    %por columnas
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:3
    for j=1:2
        Anchos(:,c) = [AW(i);AQ(j)];    %por columnas
        c=c+1;
    end
end
```

Estos arreglos se requieren por lo siguiente. Como se sabe, la lógica difusa combina variables de entrada definidas en términos de conjuntos difusos, por medio de reglas que producen uno o varios valores de salida.

La máxima cantidad de combinaciones en los antecedentes de las reglas es el resultado de multiplicar entre si las cantidades de conjuntos difusos en cada universo de discurso de cada variable, o sea si se tienen dos variables, y una variable lingüística tiene 3 conjuntos difusos definidos y la otra solo 2 en su universo de discurso, entonces, la cantidad máxima de reglas a formar (de combinaciones distintas en el antecedente) es $3 \times 2 = 6$ reglas.

Si se definen los conjuntos mediante sus centros, cada centro (3 y 2 centros) de cada conjunto debe asociarse con los centros de conjuntos de la otra variable. Este es el número máximo posible de reglas.

Los datos mostrados se corresponden con la corrida inicial realizada para la identificación de una lógica difusa con 6 reglas (dadas por el producto de 3×2 variables lingüísticas supuestas en los universos de discurso de las variables de entrada)

2.3 Entrenamiento del sistema difuso con la base de datos de la máquina MC-5

Así, la segunda parte del programa principal contiene los datos empleados para el entrenamiento y para la validación del modelo difuso. La misma se muestra a continuación:

```

%%%%%%%%% DATOS DE ENTRENAMIENTO %%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
peso=[0.002;0.008084;0.01617;0.02021;0.02425;0.02831;0.03234;0.03
638;0.04446;0.06063];
%%%%%%%%%DATOSVALIDACION%%%%%%%%%
%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
peso_VAL=[0.004042;0.01213;0.01819;0.02223;0.02627;0.03032;0.0343
6;0.0384;0.04042;0.05659];

```

El cuerpo principal del programa se desarrolla en la tercera parte, denominada “**AJUSTE**”. En ella se realiza el ajuste de los centros y anchos de las funciones de pertenecía de entrada y así como el ajuste del valor de los consecuentes del modelo de Sugeno partiendo del error de modelado y empleando el método del gradiente para la actualización por lotes.

Al inicio, se asignan los valores a los factores gamma que aparecen en las expresiones de cálculo. Se inicializan las variables auxiliares y el número de iteraciones que será empleado durante el entrenamiento. El lazo anidado en **j** comienza con los valores inicialmente declarados y calcula la salida para la lógica difusa, el error para los valores de los centros, anchos y constantes de Sugeno iniciales. Con ellos se calculan los gradientes requeridos para la actualización, la cual se efectúa, como es habitual, en las últimas instrucciones del programa al salir del lazo interno **j**.

Los valores encontrados se aplican nuevamente en la iteración por **i**, donde se requiere inicializar las matrices de los gradientes, el coste y la sumatoria del error antes de entrar en el lazo interno anidado **j**. Las variables del argumento de la función **sborr** (Y1, Centros, Anchos) ahora son las últimas encontradas, repitiéndose el ciclo de cálculo de la salida de la lógica difusa, cálculo del error, **coste**, **gradientes**, etc. hasta obtener valores aún más actualizados. Esta operación se realiza tantas veces como indica la asignación a **NumIter**.

En el programa principal y en la función **sborr**, se han empleado de forma sistemática los conceptos del álgebra de matrices que proporciona MATLAB, en especial declaraciones de matrices vacías, transposiciones de matrices, operaciones con arreglos, la indexación y productos y potenciación, elemento por elemento, de matrices.

2.4 Validación del modelo

La cuarta parte o sección del programa principal se encarga de la validación con datos proporcionados para tal efecto. La misma ha sido denominada “VALIDACIÓN” y consiste en emplear los datos obtenidos con otros valores de fuerzas y ángulos de éstas y el record de los pesos de compensación del desbalance y el ángulo donde deben ser introducidos los mismos. Estos valores forman la base de datos de validación, empleada en esta sección del programa.


```

plot(peso_VAL, 'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1), Ybo2(:,2), Ybo2(:,3), 'r*');
hold on;
plot3(valida(:,1), valida(:,2), peso_VAL, 'b*');
grid on;

```

En las Figuras 2.11 y 2.12 se muestran los resultados encontrados para los pesos a introducir en los rotores de prueba. La grafica de color rojo pertenece a los datos que proporciona el modelo difuso y en color azul, a los datos reales. Como se observa el modelo difuso aun no ha capturado toda la información que encierran los datos reales.

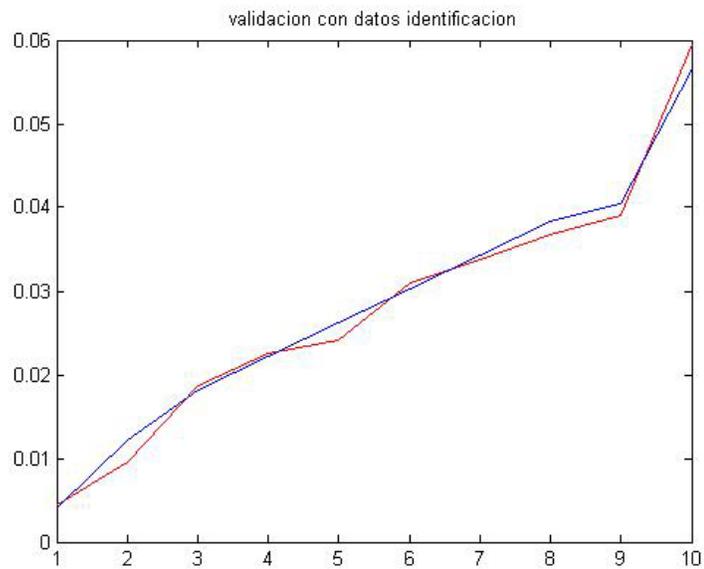


Figura 2.11. Cálculo del peso con 3x2 reglas

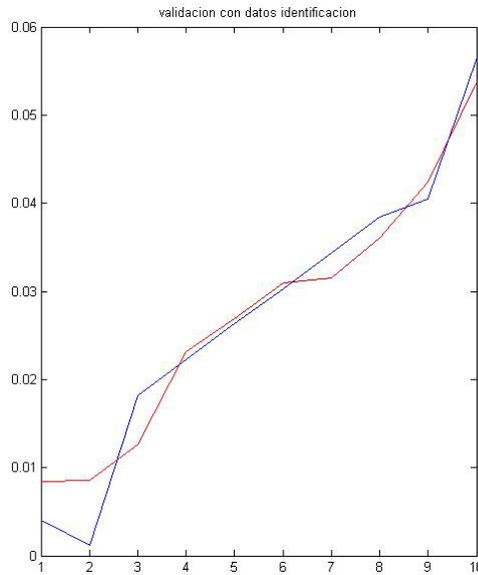


Figura 2.12 Cálculo del peso con 5x2 reglas

Las Figuras 2.13, 2.14, 2.15 y 2.16 muestran los resultados de la actualización de los modelos del algoritmo de cálculo del peso a introducir con diferentes cantidades de reglas.

Como se observa, el modelo difuso con 5x2 reglas va capturando paulatinamente los enlaces analíticos existentes en el algoritmo del método de los coeficientes.

El resto de los resultados se corresponden con 7x2, 9x2 y 9x3 reglas.

Obsérvese que la introducción de más funciones de pertenencia, en este caso de dos mas (7x2) en el universo de discurso de la variable fuerza y la variación de el ancho de la función de pertenencia central, proporciona un mejor ajuste de los pesos a introducir

en el rotor del motor comprobado al balanceo dinámico. Los datos introducidos se muestran a continuación:

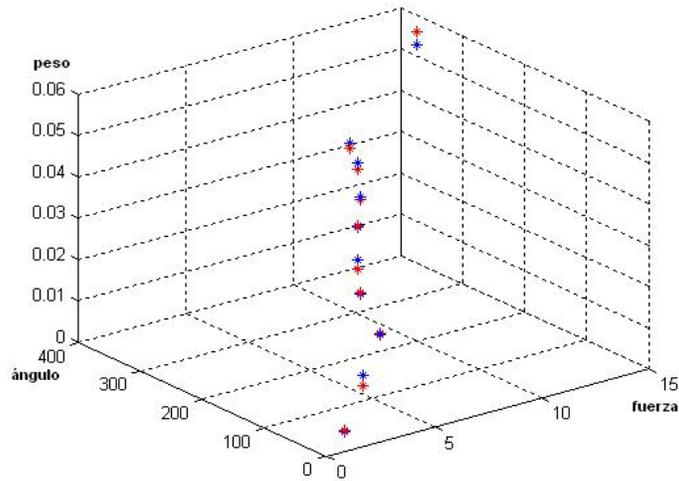


Figura 2.13 Representación en 3D del cálculo del peso con dos entradas de la lógica difusa.

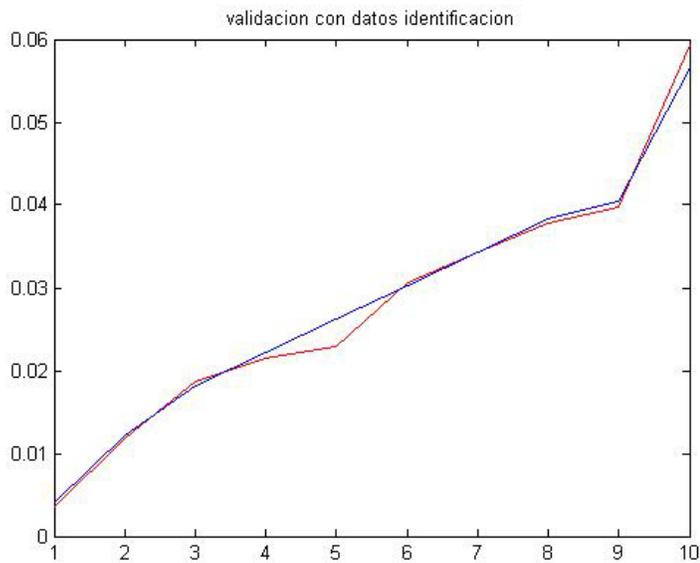


Figura 2.14 Cálculo del peso con 7x2 reglas

```
AW=[2 2 2 4 2 2 2];%7 anchos fp fuerza
CW=[2.5 4 5 7.5 10 11 12.5];%7 centros fp fuerza
```

```

AQ=[4 4];%2 anchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 14
%inicializacion valores de salida logica borrosa
Y1=[0.015 0.016 0.017 0.018 0.019 0.020 0.021 0.022 0.023 0.025
0.03 0.035 0.04 0.06]';

```

Compárense estos datos iniciales con los introducidos para el entrenamiento de la lógica difusa en el caso de 3x2 dado antes.

Las figuras 2.15 y 2.16 muestran el resultado de aumentar a 9 el número de funciones de pertenencia en las variables “fuerza”. Se observa mayor regularidad de los ajustes.

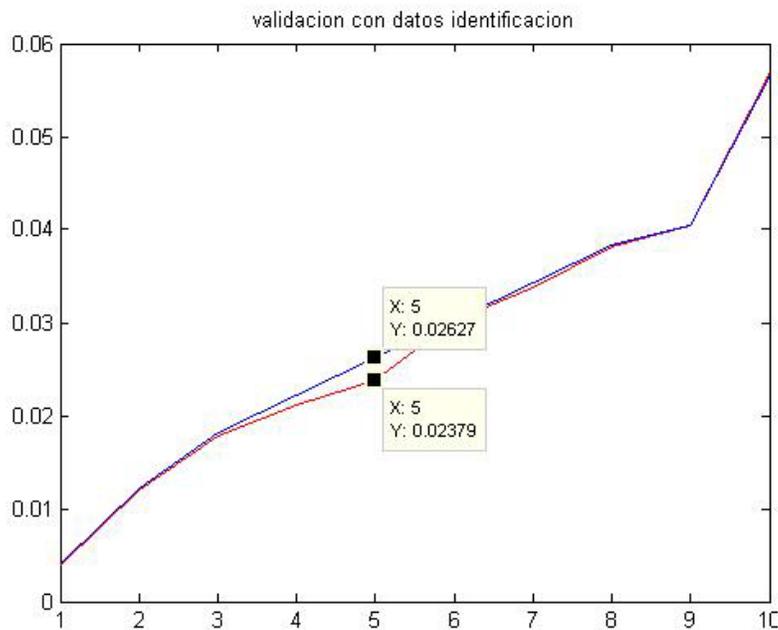


Figura 2.15 Cálculo del peso con 9x2 reglas

Las últimas Figuras 2.15 y 2.16 se deben a la introducción de una función de pertenencia más en la variable de entrada “ángulo”, o sea, ahora se trata de aumentar la exactitud empleando mas funciones de pertenencia en la segunda variable. Durante la selección del número de funciones de pertenencia y entrenamiento con varias

combinaciones se determinó que un aumento posterior del número de funciones de pertenencia en ambas variables no proporcionaba un aumento significativo de la exactitud de reproducción de los pesos. Los resultados, sin embargo, son aceptables, sobre lo cual se expondrá posteriormente.

Los datos introducidos en el programa para el entrenamiento e inicialización de los centros y anchos se ofrecen a continuación:

```
AW=[3.5 3.5 3.5 3.5 4 4 4 4 4];%9 anchos fp fuerza
CW=[2.5 3 4.5 5 6 7.5 10 11 12];%9 centros fp fuerza
AQ=[4 4 4];%3 anchos fp angulo
CQ=[90 180 270];%3 centros fp angulo
%numero maximo de reglas 27
%inicializacion valores de 27 salida logica borrosa
Y1=[0.015 0.0155 0.016 0.0165 0.017 0.0175 0.018 0.0185 0.019
0.0195 0.020 0.0205 0.0210 0.0215 0.022 0.0225 0.023 0.0235 0.024
0.0245 0.025 0.0255 0.03 0.035 0.04 0.045 0.06]';
```

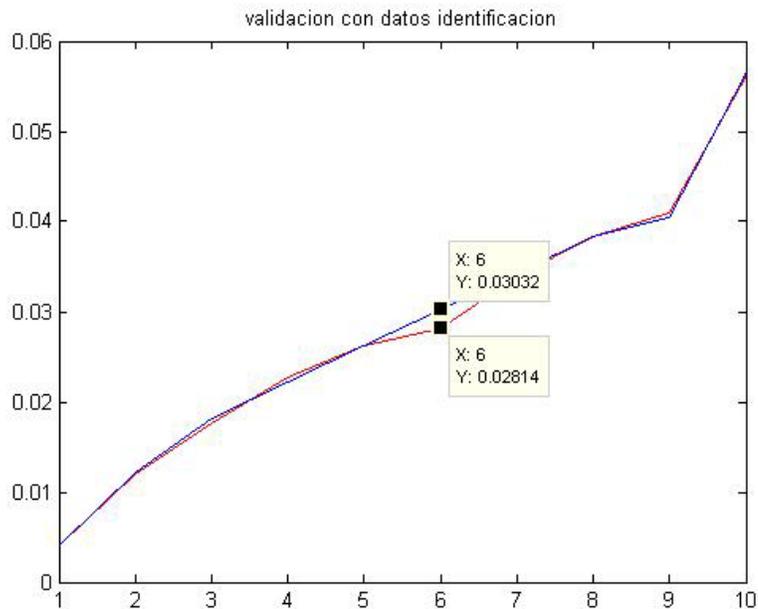


Figura 2.16 Calculo del peso con 9x3 reglas

Como puede calcularse de los datos ofrecidos en las Figuras 2.15 y 2.16 los errores de validación de la lógica difusa propuesta para el dato de salida “peso” son:

9x2: 0.0944

7x3: 0.0718

Nuevos aumentos en las cantidades de las funciones de pertenencia no proporcionaron aumento de la exactitud o disminución del error de reproducción del valor del peso.

Como resultado del entrenamiento se determinó el modelo final difuso para el cálculo de los pesos. Los centros, anchos y valores constantes del consecuente, por planos, se muestran en la Tabla 2.6:

Tabla 2.6. Centros, anchos y valores constantes del consecuente para el sistema difuso de estimación de los pesos

Variables fuerzas (amplitudes medidas)									
Centros	2.5	3	4.5	5	6	7.5	10	11	12
Anchos	3.5012	3.5	3.4996	3.5	3.9996	3.9999	3.9995	4	4.0012
Variables ángulos (retardos medidos)									
Centros	90			180			270		
Anchos	4.0012			3.9996			3.9994		
Constantes del consecuente de cada regla de Sugeno									
Consecuentes	-0.0108	0.0162	0.0164	0.0008	0.0182	0.0181	0.0223	0.0218	0.0206
	0.0275	0.0241	0.0226	0.0296	0.0275	0.0250	0.0312	0.0298	0.0269
	0.0270	0.0288	0.0299	0.0270	0.0329	0.0429	0.0407	0.0467	0.0762

El programa de entrenamiento de un sistema difuso para la determinación de los valores de los ángulos de introducción del peso en cada plano es similar al desarrollado y explicado con anterioridad. La diferencia radica en que para entrenar y determinar los valores de anchos, centros y consecuentes constantes se da al programa los datos estimados de ángulos según el plano de balanceo y por supuesto los valores iniciales

de los consecuentes constantes de Sugeno deben corresponderse con los resultados experimentales, o sea, se deben proporcionar estimaciones iniciales en el universo de discurso de la variable de salida que en este caso es “ángulo”. Se probaron variantes con diferentes cantidades de reglas. La variante más acertada es la mostrada en el programa último, con 3x7 reglas.

Estos programas se proporcionan en el Anexo 3. Se proporciona a continuación el programa último, mediante el cual se realizó el ajuste fino de los valores identificados mediante entrenamiento y en la Figura 2.17 se muestran los errores que se comenten empleando los valores de centros, anchos y constantes del consecuente ajustados. Siendo el mayor error de estimación del ángulo para ubicar los pesos es de 18.9°.

Como resultado del ajuste por entrenamiento y el ajuste fino posterior se determina que los centros, anchos y valores constantes del consecuente para el sistema difuso de estimación de los ángulos de introducción del peso, por planos, deben ser los mostrados en la Tabla 2.7, cuyo programa se presenta en el anexo 3.

Tabla 2.7. Centros, anchos y valores constantes del consecuente para el sistema difuso de estimación de los ángulos de introducción del peso.

Entrada de fuerzas (amplitudes medidas)									
Centros	2.6			6.5			12.5		
Anchos	2			2			2		
Entrada de ángulos (retardos medidos)									
Centros	5	90	139	179	218	260.38	340		
Anchos	40	40	18	15	15	15	15		
Constantes del consecuente de cada regla									
Consecuentes	180	250	270	100	90	110	179	225	270
	310	355	40	52	34	60	120	130	140
	150	140	150	-----	-----	-----	-----	-----	-----

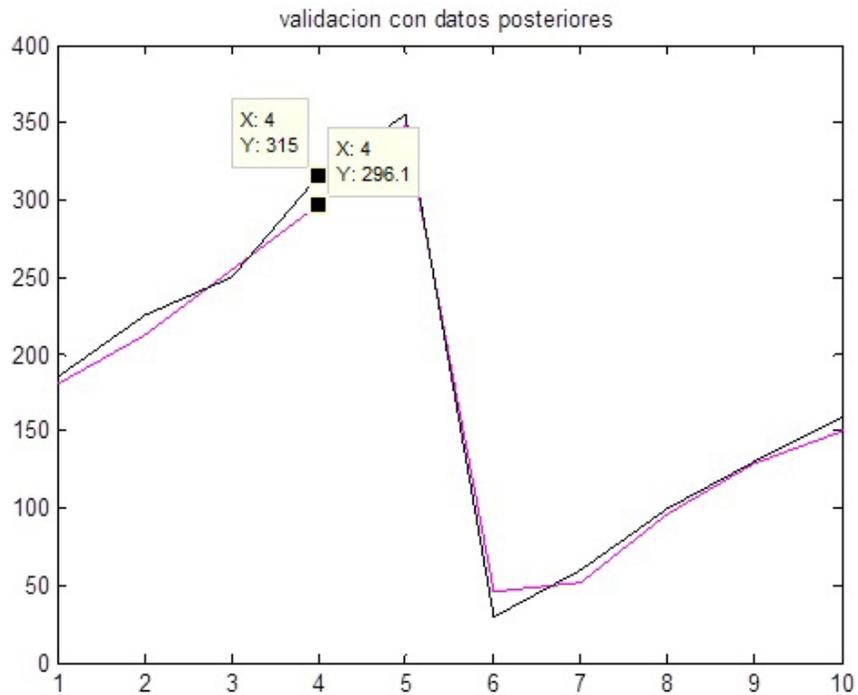


Figura 2.17 Cálculo del ángulo a introducir con 3x7 reglas.

Un posterior ajuste de los datos iniciales para los parámetros de las funciones de pertenencia del antecedente y las constantes del consecuente, permitió disminuir los errores de cálculo de la lógica difusa validada para la estimación de dicho ángulo.

En la Figura 2.18 se muestra el resultado de aplicar los datos de validación a la lógica entrenada. Se observa que el error máximo ha disminuido casi en un 50% del anterior.

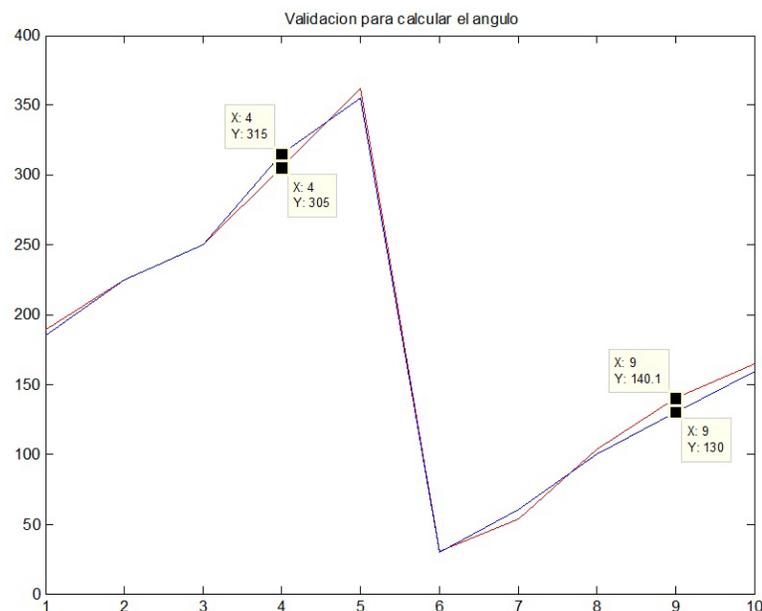


Figura 2.18 Cálculo del ángulo a introducir con 3x7 con mejor elección de valores de inicialización.

El cálculo de la desviación estándar de esta validación se muestra a continuación, el mismo que fue calculado mediante programación en el programa de entrenamiento y validación (Anexo 3).

varianza = 0.0021°

desv_stand = 0.0460°

Angulo de validación/error

185°	-4.7705°
225°	0.1176°
250°	-0.4256°
315°	9.9660°
355°	-6.9061°
30°	-0.6106°
60°	6.2921°
100°	-3.0611°
130°	-10.0583°
160°	-4.4778°

Los sistemas difusos definidos generalmente en la literatura son llamados sistemas difusos estándares. En la presente tesis se trata sobre un modelo o lógica difusa del tipo funcional. El sistema o lógica difusa del tipo Takagi-Sugeno es un caso especial de tales sistemas. Para esta lógica, en los consecuentes se emplean las funciones de pertenencia llamadas singleton por su denominación en ingles.

En la lógica aplicada a la determinación de los pesos y el ángulo de introducción se tienen R reglas TISO, o sea de dos entradas y una salida para cada par de variables de salida “peso-ángulo” por cada plano.

Existen muchas elecciones para la descripción lingüística de las variables. Algunos autores prefieren la descripción en correspondencia con los propósitos de la lógica, otros prefieren emplear notaciones lo más cortas posibles.

Se siguen las propuestas del bloque de Lógica Difusa de Simulink y se enumeran los conjuntos difusos partiendo de izquierda a derecha, desde el valor uno (1).

Así, las reglas resultantes pueden ser declaradas como sigue:

Se definen nueve conjuntos difusos para la variable difusa “amplitud de la vibración” con valores lingüísticos 1, 2, 3, 4, 5, 6, 7, 8 y 9.

Por brevedad se le llamara “amplitud”. Para la variable “desplazamiento del máximo” se obtuvieron 3 valores lingüísticos que se denominan como 1, 2 y 3. De igual manera se le llamara “desplazamiento”. La ubicación, anchos y consecuentes de estas reglas se muestran a continuación:

Las reglas para la lógica difusa que determina el peso son:

Regla	Antecedentes	Consecuentes
1	Si “Amplitud” es 1 y “Desplazamiento” es 1 entonces	-0.0108
2	Si “Amplitud” es 1 y “Desplazamiento” es 2 entonces	0.0162
3	Si “Amplitud” es 1 y “Desplazamiento” es 3 entonces	0.0164
4	Si “Amplitud” es 1 y “Desplazamiento” es 4 entonces	0.0008
5	Si “Amplitud” es 1 y “Desplazamiento” es 5 entonces	0.0182
6	Si “Amplitud” es 1 y “Desplazamiento” es 6 entonces	0.0181
7	Si “Amplitud” es 1 y “Desplazamiento” es 7 entonces	0.0223
8	Si “Amplitud” es 1 y “Desplazamiento” es 8 entonces	0.0218
9	Si “Amplitud” es 1 y “Desplazamiento” es 9 entonces	0.0206
10	Si “Amplitud” es 2 y “Desplazamiento” es 1 entonces	0.0275
11	Si “Amplitud” es 2 y “Desplazamiento” es 2 entonces	0.0241
12	Si “Amplitud” es 2 y “Desplazamiento” es 3 entonces	0.0226
13	Si “Amplitud” es 2 y “Desplazamiento” es 4 entonces	0.0296
14	Si “Amplitud” es 2 y “Desplazamiento” es 5 entonces	0.0275
15	Si “Amplitud” es 2 y “Desplazamiento” es 6 entonces	0.0250
16	Si “Amplitud” es 2 y “Desplazamiento” es 7 entonces	0.0312
17	Si “Amplitud” es 2 y “Desplazamiento” es 8 entonces	0.0298
18	Si “Amplitud” es 2 y “Desplazamiento” es 9 entonces	0.0269
19	Si “Amplitud” es 3 y “Desplazamiento” es 1 entonces	0.0270
20	Si “Amplitud” es 3 y “Desplazamiento” es 2 entonces	0.0288
21	Si “Amplitud” es 3 y “Desplazamiento” es 3 entonces	0.0299
22	Si “Amplitud” es 3 y “Desplazamiento” es 4 entonces	0.0270
23	Si “Amplitud” es 3 y “Desplazamiento” es 5 entonces	0.0329
24	Si “Amplitud” es 3 y “Desplazamiento” es 6 entonces	0.0429
25	Si “Amplitud” es 3 y “Desplazamiento” es 7 entonces	0.0407
26	Si “Amplitud” es 3 y “Desplazamiento” es 8 entonces	0.0467
27	Si “Amplitud” es 3 y “Desplazamiento” es 9 entonces	0.0762

Las reglas para la lógica difusa que determina el ángulo de introducción del peso son:

Regla	Antecedentes	Consecuentes
1	Si "Amplitud" es 1 y "Desplazamiento" es 1 entonces	180
2	Si "Amplitud" es 1 y "Desplazamiento" es 2 entonces	250
3	Si "Amplitud" es 1 y "Desplazamiento" es 3 entonces	270
4	Si "Amplitud" es 1 y "Desplazamiento" es 4 entonces	100
5	Si "Amplitud" es 1 y "Desplazamiento" es 5 entonces	90
6	Si "Amplitud" es 1 y "Desplazamiento" es 6 entonces	110
7	Si "Amplitud" es 1 y "Desplazamiento" es 7 entonces	179
8	Si "Amplitud" es 2 y "Desplazamiento" es 1 entonces	225
9	Si "Amplitud" es 2 y "Desplazamiento" es 2 entonces	270
10	Si "Amplitud" es 2 y "Desplazamiento" es 3 entonces	310
11	Si "Amplitud" es 2 y "Desplazamiento" es 4 entonces	355
12	Si "Amplitud" es 2 y "Desplazamiento" es 5 entonces	40
13	Si "Amplitud" es 2 y "Desplazamiento" es 6 entonces	52
14	Si "Amplitud" es 2 y "Desplazamiento" es 7 entonces	34
15	Si "Amplitud" es 3 y "Desplazamiento" es 1 entonces	60
16	Si "Amplitud" es 3 y "Desplazamiento" es 2 entonces	120
17	Si "Amplitud" es 3 y "Desplazamiento" es 3 entonces	130
18	Si "Amplitud" es 3 y "Desplazamiento" es 4 entonces	140
19	Si "Amplitud" es 3 y "Desplazamiento" es 5 entonces	150
20	Si "Amplitud" es 3 y "Desplazamiento" es 6 entonces	140
21	Si "Amplitud" es 3 y "Desplazamiento" es 7 entonces	150

Los siguientes comandos de MATLAB se introducen para graficar la varianza:

```
i=1:500;
```

```
plot(i,varianza)
```

```
title('Comportamiento del criterio')
```

```
ylabel('Varianza')
```

```
xlabel('NumIter')
```

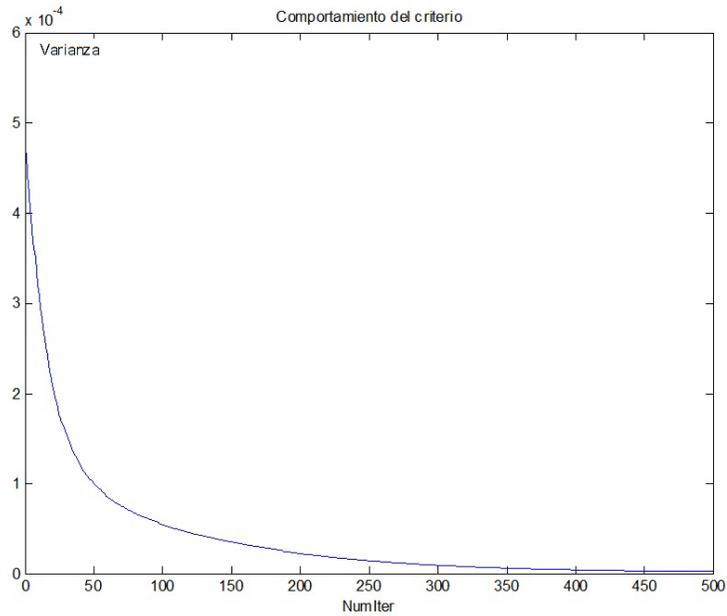


Figura 2.19 Evolución de la varianza con el numero de iteraciones.

De la Figura 2.19 se observa que la varianza obtenida partiendo de la minimización de la variable “coste” reduce significativamente su valor en 50 iteraciones y es prácticamente cero a partir de la iteración 350. Este valor determina el número de iteraciones a partir del cual puede detenerse el entrenamiento.

Los criterios de validación de modelos existentes se ajustan a procesos temporales, [Ljung, 1999]. Estos se fundamentan en la prueba de blancura de los residuos, validación cruzada, etc.

En la presente tesis se trata con un aproximador de funciones, para el cual los métodos de validación son líneas de investigación abiertas. No obstante, se sigue el mismo

criterio de evaluar los residuos, pero en este caso se evalúan los residuos relativos dado que los valores obtenidos de residuos se corresponden con diferentes rotores a balancear.

Así, se propone validar el aproximador universal de funciones encontrado para la estimación del peso a introducir en un rotor desbalanceado empleando la siguiente expresión para la estimación de la varianza:

$$\bar{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (\text{dif})^2$$

Donde:

dif=(peso_VAL)'-Valyb – diferencia entre el peso de validación (peso_VAL) y la salida de la lógica difusa para cada par de datos de validación (Valyb);

$\bar{\sigma}^2$ - promedio de la dispersión en la estimación del valor del peso a introducir.

N=10

En las expresiones anteriores se han empleado los nombres de las variables de los programas elaborados.

El resultado de aplicar este segmento de programa luego de obtener los resultados para las diferentes lógicas probadas de 3x2, 5x2, 7x2, 9x2 y 9x3 se muestra en la Tabla 2.8.

Tabla 2.8 Valores de las desviaciones estándar por estructura de la lógica difusa validada

Estructuras (modelos)	3x2	5x2	7x2	9x2	9x3
desv_estand	0.0030 gr	0.0016 gr	0.0014 gr	0.0009 gr	0.0008 gr

Los errores de la lógica difusa en cada valor de peso de validación en la estructura de 9x3 se muestran a continuación en la Tabla 2.9:

Tabla 2.9 Errores de la lógica difusa

peso-VAL	0.004	0.0121	0.0182	0.0222	0.0263	0.0303	0.0344	0.0384	0.0404	0.0566
error	0.00	0.0001	0.0005	-0.0004	0.00	0.0022	0.0004	0.0001	-0.0006	0.0003

Los resultados mostrados en las Figuras 2.11, 2.12, 2.14, 2.15 y 2.16, así como los mostrados en la Tabla 2.8, al emplear este criterio, evidencian que el modelo de lógica difusa con 27 reglas presenta la menor varianza.

Considerando ahora como criterio de comparación de la exactitud de la lógica difusa diseñada el error porcentual de la pesa empleada para pesar el peso de los pernos introducidos para balancear los rotores el cual es de 0.01%, tendremos que el modelo o estructura de la lógica difusa que mejor reproduce el peso a introducir es el quinto modelo, o sea, la estructura de la lógica difusa que tiene 27 reglas con 9 funciones de pertenencia para la variable de entrada “amplitud de la vibración” y 3 para la variable de entrada “desplazamiento del máximo”. Esta será la estructura de la lógica difusa elegida.

Un grafico que muestra los valores calculados se presenta en la Figura 2.20.

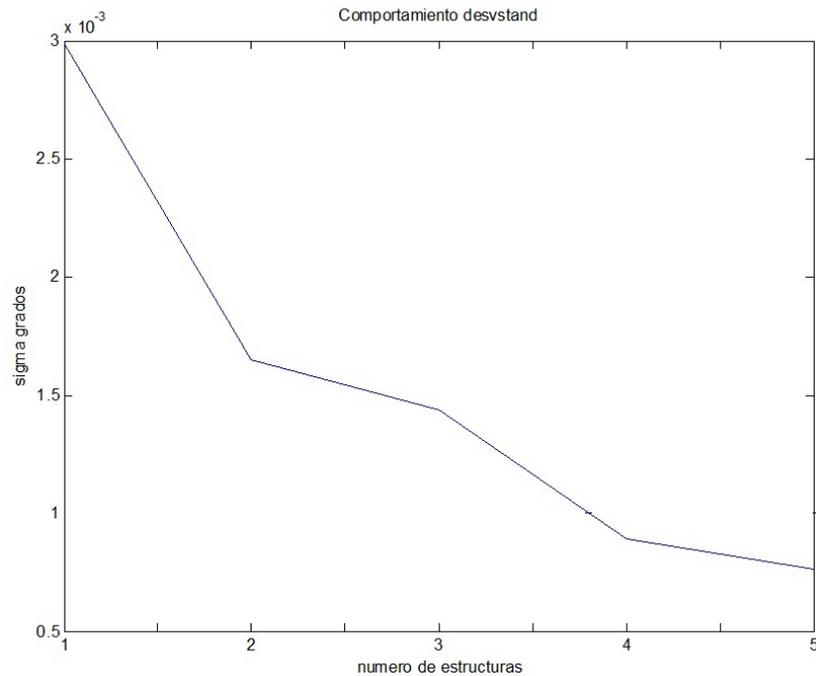


Figura 2.20 Desviación estándar para cada estructura de lógica difusa validada

Un análisis similar se realizó para el cálculo del ángulo de introducción del peso calculado mediante la lógica difusa validada.

El segmento de programa que ejecuta estas operaciones, luego de ejecutar el guion principal, dado en los Anexos de programación, se muestra a continuación:

```
dif=(peso_VAL)'-Valyb;  
disp=(dif.*dif);  
unos=ones(10,1);  
var=0.1*(disp*unos)  
desv_stand=sqrt(var)  
matriz_error=[ang_VAL dif']
```

2.4.1 Conclusiones del capítulo

Se demostró que existe una dependencia funcional entre los pesos a introducir para el balanceo dinámico de los rotores y la amplitud de las vibraciones medidas y otra relación funcional, diferente, entre el desplazamiento del máximo de estas oscilaciones respecto de la marca dada por el sensor estroboscópico.

Tomando una base de datos para entrenamiento y otra para la validación se encontró el Aproximador de Funciones o lógica difusa requerida.

La determinación del número de funciones de pertenencia por variables de entrada a la lógica difusa se realizó considerando la existencia de indicaciones para el diseño de la lógica difusa para control, no obligatorias para los aproximadores de funciones y la selección por prueba y error de este número.

Se propone un programa de entrenamiento o ajuste de los parámetros de la lógica difusa por el método del gradiente, que puede, además, servir de base para otras aplicaciones.

La lógica difusa que se propone se compone de funciones de pertenencia de tipo gaussianas, con centros y anchos definidos mediante entrenamientos por lotes. De igual manera, se determinaron los valores de los consecuentes para una proposición de Takagi-Sugeno con constantes en el consecuente.

El sistema de lógicas difusas propuesto se puede representar como se muestra en la figura, o sea para cada plano de balanceo se deben programar dos lógicas difusas,

cada una de las cuales tendrá su distribución de conjuntos difusos los cuales componen las bases de las funciones de pertenencia de gauss.

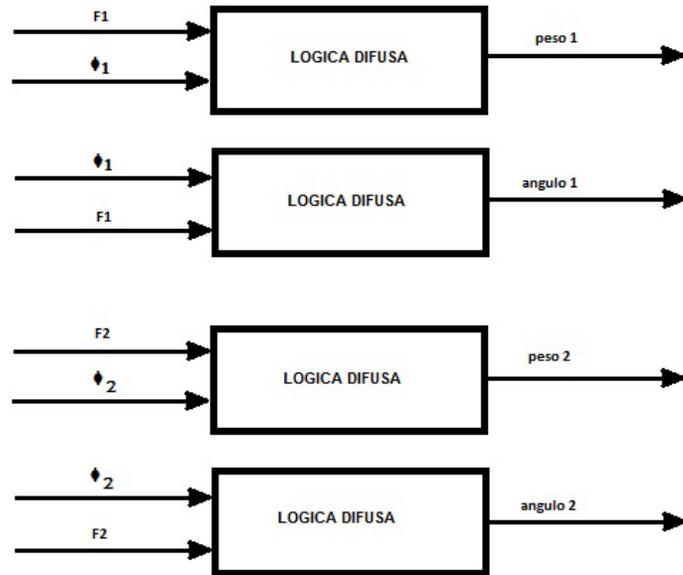


Figura 2.21 Sistema de lógicas difusas propuesto.

CAPÍTULO III:

DISEÑO DEL INSTRUMENTO VIRTUAL.

El entorno de programación escogido es Labview (Laboratory Virtual Instrument Engineering Workbench).

Labview es un entorno gráfico de programación orientado al control de instrumentos y adquisición de datos mediante un PC. Labview permite crear de una manera sencilla instrumentos virtuales.

Este entorno fue creado en 1976 por National Instruments para funcionar únicamente sobre máquinas MAC. Posteriormente, se incorporó su uso a las plataformas Windows, Unix y Linux.

Una definición completa sería que: se trata de una herramienta gráfica de test, control y diseño mediante la programación. Al lenguaje que Labview utiliza para programar se le denomina, Lenguaje G, debido a que es una programación gráfica.

A los programas que se crean con este entorno se les denomina VI (Virtual Instrument), lo que hace intuir que su principal aplicación es la creación de instrumentos de manera virtual.

La característica que lo distingue del resto de entornos de programación, es que, un usuario principiante en programación, es capaz de realizar programas con un cierto grado de complejidad que le sería imposible realizar en otros entornos, debido a que es una programación gráfica lo que hace que la creación de programas sea mucho más intuitiva. En este sentido tiene similitud con la herramienta Simulink de MATLAB.

Un VI o un programa de Labview, se divide en dos partes:

Panel Frontal o Front Panel

En esta parte, el usuario diseña el programa estéticamente a su gusto y necesidades, es decir, esta parte será la que se visualiza al ejecutar el programa, por lo tanto, contendrá todos los botones, gráficos, tablas, indicadores, etc. necesarios para la interacción entre el usuario, la maquina de comprobación y compensación del desbalance de los rotores a prueba y el computador o PC.

Un ejemplo de panel frontal simple sería el siguiente:



Figura 3.1. Panel Frontal en Labview

Diagrama de bloques o Block Diagram

En esta parte, el usuario ha de desarrollar toda la programación, es decir, ya aparecen todos los controles creados en el Panel Frontal y ahora se trata de hacer el diseño colocando cada control donde se necesite.

Un ejemplo de Diagrama de bloques sería el siguiente:

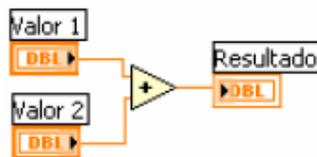


Figura 3.2. Diagrama de Bloques en Labview

Como se observa en las Figuras 3.1 y 3.2, a simple vista, no resulta ser un lenguaje de programación muy complicado. En el ejemplo, en el Panel Frontal, se crearon 3 variables, las cuales aparecen automáticamente en el Diagrama de Bloques y solo se tendrá que unir las variables con la operación deseada, en este caso la suma.

LabView proporciona diferentes bloques con los cuales se puede realizar la medición del valor máximo de una señal sinusoidal.

Uno de estos bloques es:

“AMPLITUDE AND LEVEL MEASUREMENTS”

Las opciones que ofrece el Cuadro de Dialogo de este bloque permiten realizar diferentes funciones, por selección, sobre los parámetros de amplitud de una señal sinusoidal. Estas opciones se muestran en la Tabla 3.1. En la Figura 3.3 se muestra el Diagrama de Bloque mencionado.

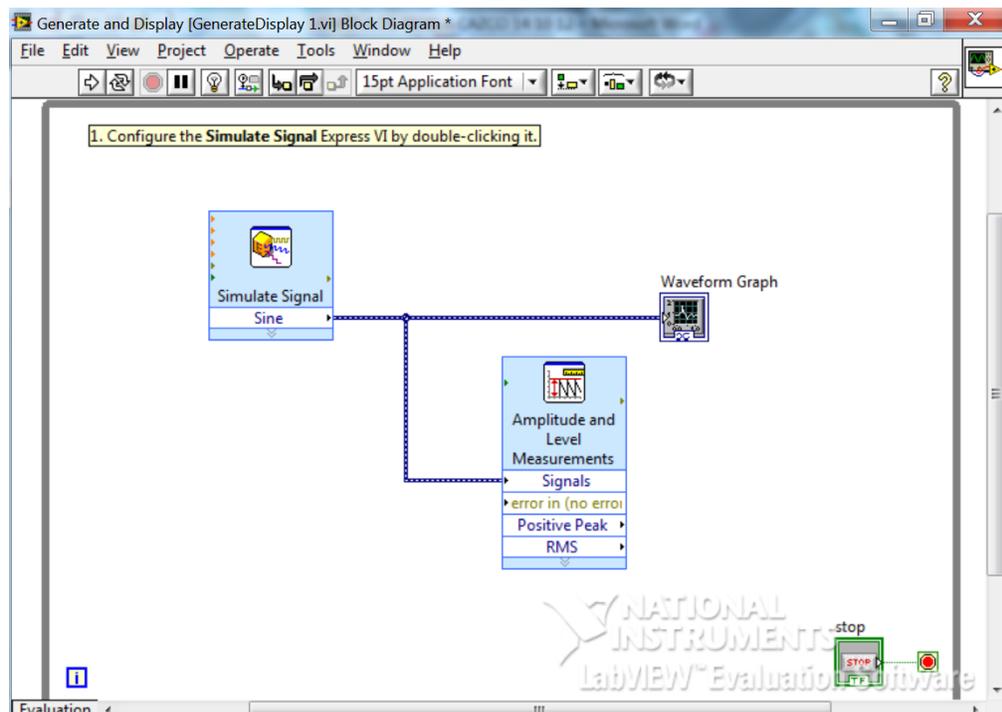


Figura 3.3. Diagrama de Bloque Medición de Amplitud y Nivel.

Tabla 3.1. Opciones del bloque “Amplitude and level measurements”

Parámetros	Descripción
Medición de Amplitud	<p>Contiene las siguientes opciones:</p> <ul style="list-style-type: none"> "DC"- adquiere la medición de una señal DC, "RMS"-calcula la raíz cuadrada del valor medio de la señal, "Ventana de Aplicaciones - Esta opción sólo está disponible cuando se pone un checkmark en la casilla de verificación DC

	<p>o RMS.</p> <p>"Pico Maximo" – realiza la medición del valor mas positivo en la señal"</p> <p>"Picos Minimo" – realiza la medición del valor mas negativo en la señal.</p> <p>"Valor Pico a Pico"- realiza la medición entre el valor mas positivo y el valor mas negativo en los Signos.</p> <p>"Promedio en un ciclo – realiza la medición del nivel medio en un periodo completo de la señal periodica.</p> <p>"Valor RMS"- calcula la raíz cuadrada del valor medio de la señal en un período completo de una señal periódica de entrada.</p>
--	---

Las entradas de este diagrama en bloque y su descripción se muestran en la Tabla 3.2.

Tabla 3.2. Entradas del bloque "Amplitude and level measurements"

Parámetro	Descripción
Signals	Entrada de aplicación de la señal o señales de entrada
Restart averagung	Especifica el proceso de promediación seleccionado cuando se reinicia. El valor por defecto es FALSO. Cuando se llama a este VI por primera vez, el proceso de promediación se inicia automáticamente. Esta entrada sólo aparece si se pone una marca de chequeo en la casilla de verificación Averaging .
Error in	Describe las condiciones de error que ocurren antes de que este VI o function comience a correr.

Las salidas de este diagrama en bloque pueden ser varias. Las mismas se listan y describen en la Tabla 3.3:

Tabla 3.3. Salidas del bloque “Amplitude and level measurements”

Parámetros	Descripción
Promedio en un ciclo	Devuelve el nivel medio en un periodo completo de la señal de entrada. El promedio en un ciclo de una onda sinusoidal perfecta es cero, pero el nivel promedio de una señal complete puede no ser cero como resultado de los periodos parciales en los límites de la señal.
Valor RMS en un ciclo	Devuelve la raíz cuadrada del valor medio cuadrático de un periodo completo de una señal periódica.
Valor medio (DC)	Devuelve el valor de directa medido en la entrada “Señal”.
RMS	Devuelve el valor RMS calculado de los valores en la entrada “Señal”.
Pico negativo	Devuelve una medición del pico mas negativo en la entrada “Señal”.
Pico positivo	Devuelve una medición del pico mas positivo en la entrada “Señal”.
Error out	Contiene la informacion de error. Si “ error in ” indica que, antes de que corra este VI ha ocurrido un error, entonces en esta salida estará el mismo error. De otra forma, describe el estado de error que este VI o función produzca.
Pico a pico	Devuelve una medicion desde el pico mas positive hasta el pico mas negativo en la entrada “Señal”.

La descripción detallada del diagrama en bloques, de sus entradas y salidas proporcionan la suficiente información para decidir si se puede emplear el mismo en la aplicación que se requiere para la máquina de balanceo dinámico analizada.

Efectivamente, para que la máquina en cuestión y la aplicación en Lógica Difusa funcionen correctamente, se necesita que el diagrama de bloque proporcione el valor máximo de la señal de los sensores.

La salida “Pico Positivo” no devuelve la amplitud de la señal sino el valor máximo partiendo desde cero ya que los voltajes aplicados a la PC desde el bloque de

adecuación de señales existentes en la maquina se alimenta entre cero y 15VDC, por lo que esta salida no puede ser empleada. Sin embargo, una solución algo más complicada pudiera ser emplear la salida que proporciona el valor “pico a pico” y determinar la mitad de este valor, pero para ello se requiere de hacer algo más compleja la manera de obtener el resultado.

Un análisis más detallado de las ideas que se proponen sigue a continuación. Para ello se empleara la Figura 3.4, en la cual se muestran las señales sinusoidal filtrada (del sensor de vibraciones en un pedestal) y el pulso del sensor estroboscópico.

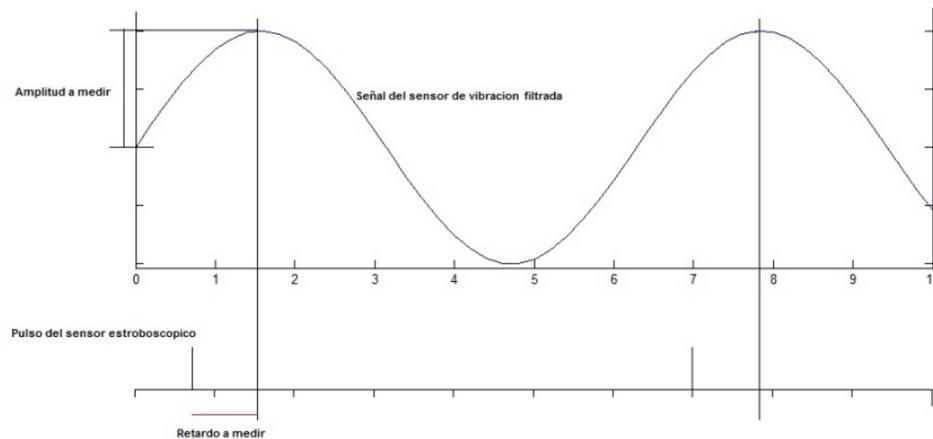


Figura 3.4. Señales del sensor de vibraciones y del sensor estroboscópico de la Máquina de Balanceo Dinámico.

Como se observa el problema no es solo determinar el valor máximo de la amplitud de la señal sinusoidal, sino que también se requiere determinar el retardo en tiempo y por tanto la posición angular, de este valor máximo respecto de la posición del pulso (ángulo cero) del sensor estroboscópico.

Luego, de lo que se trata es que el diagrama de bloque que se emplee sea capaz de realizar estas operaciones con el menor número de bloques y en el menor tiempo computacional.

Detectores de pico en LabView

La detección de picos es una de las funciones en el dominio del tiempo realizadas en el monitoreo de señales.

La detección de picos es el proceso de encontrar las ubicaciones y las amplitudes de los máximos y mínimos locales en una señal que satisface ciertas propiedades. Estas propiedades pueden ser simples o complejas. Por ejemplo, si se requiere que un pico sea aquel valor de la señal que exceda cierto valor umbral entonces es una propiedad simple. Sin embargo, si se requiere que la forma del pico se parezca a cierta forma prototipo, entonces es una propiedad compleja.

La detección de picos es importante en muchas aplicaciones, tales como en química, biología, y música. Los científicos e ingenieros que a menudo usan las técnicas de análisis tales como la espectroscopia, la cromatografía y la detección del tono usan los métodos de detección de máximos específico para aquellas técnicas de análisis.

Ahora se describe un método general que se aplica a una variedad de tipos de señales. Éste es el método usado en LabVIEW para las funciones de detección de máximos.

En algunas aplicaciones no se necesita saber con exactitud los valores máximos y sus posiciones, más bien se necesita saber el número o las situaciones generales de los

picos. En este caso se usa la función de detección por umbral de los máximos, tal como el VI Detector de Pico por Umbral de LabVIEW.

La Figura 3.5 muestra el VI Detector de Pico por Umbral y las entradas y salidas del VI.

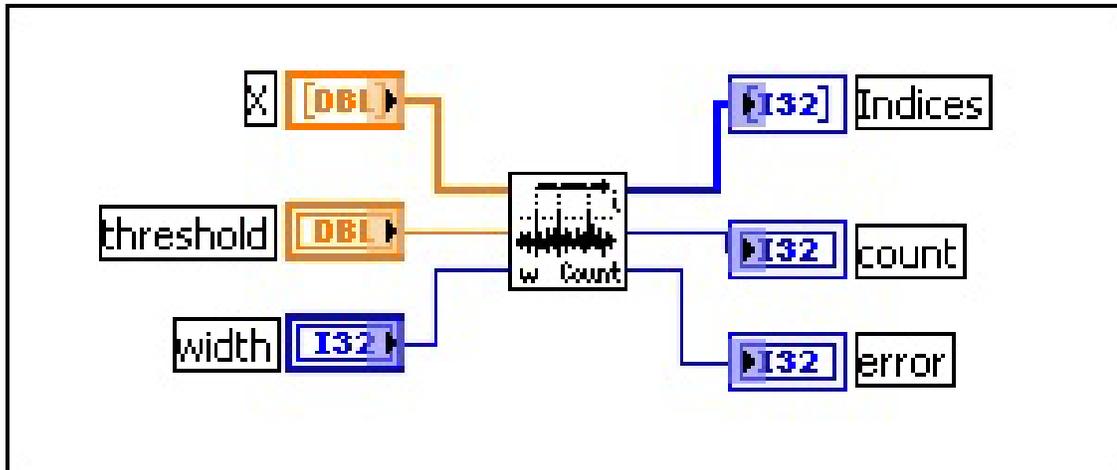


Figura 3.5. Detector de pico por umbral.

El VI examina la secuencia de entrada X, busca los picos válidos, y guarda la huella de los índices de los picos y el número total de picos encontrados. Un pico es válido si tiene las características siguientes:

- Los elementos de X que empiezan por debajo del umbral, exceden el umbral en algún índice, y retorna a un valor por debajo del umbral
- El número de elementos sucesivos que exceden el umbral es mayor que o igual al ancho.

Este VI no identifica las posiciones ni las amplitudes de los picos con gran exactitud, pero da una idea de dónde y cuan a menudo ocurre un cruce de la señal sobre un cierto valor umbral.

El gráfico de la Figura 3.6 muestra una señal de múltiples tonos después de ser examinada por el VI Detector de Picos por Umbral. Los parámetros de entrada son: umbral = 1.00 y ancho = 10. El VI identifica dos picos, localizados a aproximadamente en los instantes de tiempo 15 y 47. Los instantes en que ellos cruzan el umbral se marcan por los puntos negros en la Figura 3.2.

El VI, sin embargo, no identifica el tercer pico, que cruza el umbral aproximadamente en el instante de tiempo 132 porque no tiene un ancho de por lo menos 10 puntos. Este VI tiene algunas aplicaciones limitadas pero importantes. Es importante entender la diferencia entre este VI y el VI Detector de Máximos VI descrito más abajo.

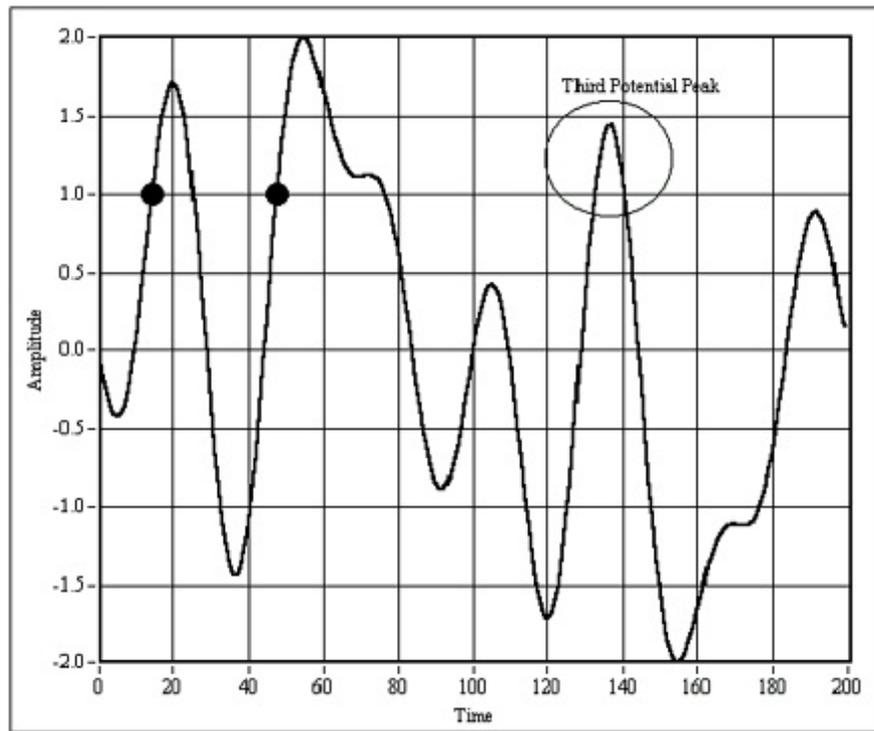


Figura 3.6. Señal de múltiples tonos después de ser examinada por el VI Detector de Picos por Umbral.

Detector de Pico Avanzado

Algunas aplicaciones requieren algoritmos de detección de máximos más robustos y exactos. Se enfocan ahora los usos de las funciones de detección avanzada del máximo, en especial se enfocan principalmente los picos. La misma información puede usarse para encontrar los mínimos locales.

La Figura 3.7 muestra el VI Detector de Pico y las entradas y salidas del VI. Obsérvese que el mismo toma los arreglos de las ubicaciones, de las amplitudes y de los datos de la segunda derivada.

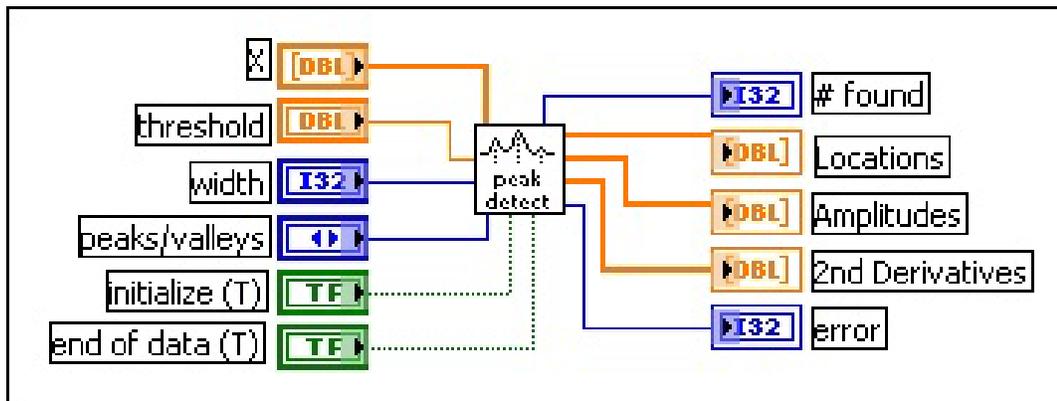


Figura 3.7. VI Detector de Pico y las entradas y salidas del VI

Otro de los diagramas de bloque que proporciona LabView es el diagrama “Waveform Peak Detector”, el cual se analizara en el numeral 3.3.

3.1 Generalidades del diseño del instrumento virtual con LabView de National Instrument en su variante difusa.

La lógica difusa es un método de decisión basado en reglas usado por los expertos en el control de sistemas y procesos que emulan la regla de la experiencia usada por los seres humanos. La lógica difusa puede usarse para controlar un proceso que generalmente se controla de forma manual por una persona con experiencia en dicho control manual. Las reglas de control lingüísticas, que un experto humano puede describir de manera general e intuitiva, pueden ser trasladadas directamente en una base de reglas para un controlador basado en la lógica difusa.

Con la Lógica Difusa para la herramienta G, se puede diseñar un controlador de la lógica difusa o un sistema experto para tomar una decisión en aplicaciones de la herramienta G.

El número de funciones de pertenencia y la base de reglas del controlador quedan definidos con el Diseñador de Instrumentos Virtuales de Lógica Difusa. El Instrumento Virtual de Diseño del Controlador es un Instrumento Virtual autónomo con una Interface Gráfica de Usuario que define completamente los componentes del sistema experto o controlador.

Todos los parámetros definidos para el controlador o sistema experto se guardan en un archivo de datos del controlador. Para llevar a cabo el controlador difuso en su aplicación G se usan dos Instrumentos Virtuales adicionales.

El Instrumento Virtual del Controlador Difuso guardado se usa para cargar todos los parámetros del controlador difuso en un archivo de datos mediante el Instrumento Virtual de Diseño del Controlador (Controller Design VI).

Estos datos se vinculan al Instrumento Virtual de Diseño del Controlador, los cuales se emplean en el funcionamiento de la Máquina de Inferencias de la lógica difusa.

Los parámetros del proceso (o entradas del controlador) se vinculan a las entradas del Instrumento Virtual del Controlador Difuso, y las salidas del controlador son las salidas del instrumento virtual.

Mediante la vinculación de los datos adquiridos por el hardware de adquisición de datos al controlador difuso, se puede llevar a cabo la toma de decisión en tiempo real con la cual se puede controlar un sistema físico. Adicionalmente, las salidas del controlador

difuso pueden ser usadas por un hardware de adquisición de datos (DAQ) de salidas analógicas para llevar a cabo el control del proceso en tiempo real.

El bloque primario del sistema de la lógica difusa es el bloque de variable lingüística. La variable lingüística se usa para combinar varias categorías subjetivas que describen el mismo contexto. Cada término lingüístico es representado por un conjunto difuso definido por una función de pertenencia.

Cuando se usan las variables lingüísticas para modelar los sistemas de ingeniería, son importantes las regiones solapadas de los términos lingüísticos vecinos.

Otro concepto básico de la lógica difusa involucra los procesos de toma de decisiones basada en reglas.

No siempre es necesaria una descripción matemática detallada para el funcionamiento óptimo de un proceso de ingeniería. En otras palabras, los operadores humanos, a menudo, son capaces de manejar situaciones complejas en una planta sin saber nada o casi nada sobre ecuaciones diferenciales.

Su conocimiento de la ingeniería está quizás disponible en una forma lingüística tan simple como

"Si la temperatura del líquido es correcta Y el valor del pH es demasiado alto, ENTONCES, ajustar el nivel de agua alimentada a un nivel más alto".

A menudo es imposible desarrollar un modelo matemático del sistema debido a las no linealidades inherentes, a los parámetros distribuidos y a las constantes de tiempo, que a veces son difíciles de determinar para un ingeniero de control.

Con la lógica difusa, la representación lingüística se usa para llevar a cabo el diseño del conocimiento de un experto o diseñar una estrategia de control.

Después de convertir los valores físicos de entrada en valores lingüísticos, se identifican todas las reglas de la base de reglas para aplicar la regla correcta a la situación. Estas reglas se identifican para calcular los valores de la variable lingüística de salida. La inferencia difusa consiste en dos componentes:

- Agregación – que no es más que la evaluación del antecedente (la condición) de cada regla
- Composición - que no es más que la evaluación del consecuente (la conclusión) de cada regla

Ya que las condiciones lingüísticas representan condiciones que son parcialmente verdad, la lógica dual convencional booleana AND no satisface el modelado difuso de la palabra AND (Y). Es por esto que se definen nuevos operadores en LabView para representar las conectividades lógicas AND, OR y NOT.

Los tres operadores usados en la mayoría de las aplicaciones de la lógica difusa se definen en un gran número de monografías, como por ejemplo en Passino, por lo que no se trataran de explicar en esta tesis.

3.2 Diseño de los Bloques y subbloques de Instrumentos Virtuales requeridos para el sistema difuso.

Para diseñar un sistema difuso se puede usar LabVIEW PID and Fuzzy Logic User Manual en dos maneras: usando al Diseñador de Sistema Difuso o usando Fuzzy Logic VI.

Para emplear Fuzzy System Designer se selecciona la Herramienta "Control Design and Simulation Fuzzy System Designer" para activar al Diseñador del Sistema Difuso.

El primer paso en el diseño de un sistema difuso con este Diseñador del Sistema es crear las variables lingüísticas de entrada y salida para el sistema.

Para crear las variables deben realizarse los siguientes pasos:

1. Seleccionar la Herramientas Control Design and Simulation Fuzzy System Designer para activar al Diseñador del Sistema Difuso. Este diseñador se despliega mostrando, por defecto, la página de variables **Variables**.
2. Haciendo click en botón Add Input Variable a la derecha de la lista de variables de entrada para activar la caja de dialogo Edit Variable.
3. En la caja de texto Name se debe escribir el nombre de la variable.
4. Se deben indicar los valores mínimo y máximo.
5. Hacer clic en el botón Add Membership Function para crear una nueva función de pertenencia para la variable lingüística.
6. En la caja de texto Name se debe escribir el nombre que se le asignara a esta función de pertenencia.

7. En el menú desplegable Shape se selecciona el tipo de función de pertenencia que determinara el grado de pertenencia de la variable lingüística.
8. Seleccionar el color que se quiere usar para la función de pertenencia en el gráfico desde la paleta de colores.
9. En el arreglo de puntos (Points Array) se declaran los puntos bases y altura de la función de pertenencia elegida.
10. Repetir los pasos desde el 5 hasta el 9 para cada una de las funciones de pertenencia.
11. Pulsar el botón OK para guardar los cambios. En la lista de las variables debe aparecer el nombre asignado a la variable y las correspondientes funciones de pertenencia en el grafico.

El procedimiento anterior se repite para crear el resto de las variables lingüísticas requeridas de entrada y salida.

Después de que se definen las variables lingüísticas de entrada y salida para el sistema difuso, se usaran estas variables lingüísticas para crear la Base de Reglas para el sistema.

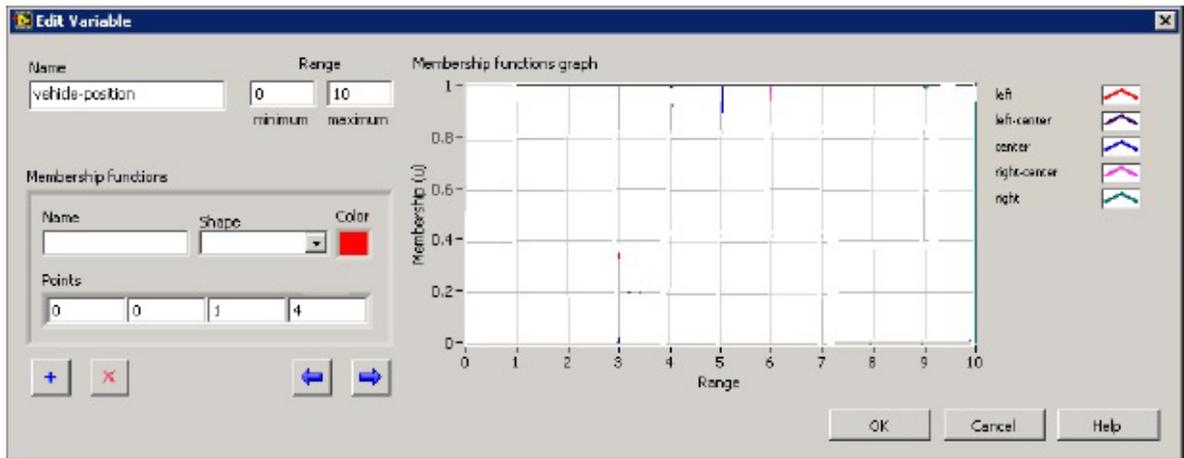


Figura 3.8 Diseñador del Sistema para crear las variables lingüísticas

Las reglas describen, en palabras, las relaciones entre las variables lingüísticas de entrada y salida con base en sus condiciones lingüísticas. La Base de Reglas de un sistema difuso determina los valores de la salida del sistema difuso con base en los valores de la entrada.

Las posibles combinaciones serán, como ocurre en cualquier sistema difuso el producto de la cantidad de funciones de pertenencia de cada variable lingüística declarada.

Para crear la Base de Reglas se realizan los siguientes pasos.

1. Hacer click en la tabla de reglas (Rules tab) del Diseñador del Sistema adifuso para desplegar la página de las Reglas.
2. Seleccione Operate >> Pre-Generate Rules para desplegar la caja de diálogo de Reglas Pre Generadas.

3. Seleccione **And** (el **Mínimo**) del menú desplegable **Antecedent connective** (Conjuntivo de Antecedente). Esta opción especifica que el grado más pequeño de la función de pertenencia de los antecedentes determinara el valor verdadero del antecedente de la regla agregada para cada regla.
4. Entre el **Grado de apoyo deseado**. Esta opción especifica el peso que usted quiere aplicar a cada regla. El peso final de la regla, para cada regla, es igual al **Grado de apoyo** multiplicado por el valor verdadero del antecedente de la regla agregada.
5. Seleccione **Mínimo** desde el menú desplegable **Consequent Implication** (la Implicación del Consecuente). Esta opción especifica que la lógica difusa trunca las funciones de pertenencia de salida en el valor de los pesos de la regla correspondiente antes de realizar la conversión del valor difuso al valor numérico (des fusificación).
6. Hacer click en el botón **OK** para poblar la lista de Reglas con todas las combinaciones posibles de condiciones lingüísticas de las variables de entrada.

Cada regla usa el mismo antecedente conjuntivo, el grado de apoyo, y el método de implicación especificado en la caja de diálogo de Reglas Pre Generadas.

En el caso que se requiera modificar las reglas pre-generadas con los consecuentes correctos, se deben realizar los pasos siguientes:

1. Seleccionar la Regla en la lista de las Reglas.
2. En la columna **THEN** para la Regla seleccionada, se selecciona el término lingüístico correcto para la variable lingüística de salida.

3. Repetir los pasos 1 y 2 para cada regla.
4. Seleccione **File >> Save** para guardar el sistema difuso.

Después de esto, el sistema difuso tiene una base de reglas lingüística completa con la cual podrá analizar los datos de entrada y producir los datos de salida. Sin embargo, la base de reglas aun devuelve datos de salida como términos lingüísticos. O sea, el sistema difuso debe convertir los datos para que la salida de la base de reglas antes especificada pueda ser aplicada a una estructura de control, si es el caso, o pueda ser interpretada como decisión del sistema experto.

Entonces es necesario especificar un método de conversión de variables lingüísticas a numéricas.

Después de crear la base de reglas para el sistema difuso, se debe especificar cómo la lógica difusa realiza la conversión. La conversión o des fusificación es el proceso de convertir los grados de pertenencia de las variables lingüísticas de salida en valores numéricos.

LabView proporciona la posibilidad de escoger cualquiera de los siguientes métodos de transformación, Centro de los Máximos, Centro del Área o Centro de las Sumas.

En la página de las Reglas del Diseñador del Sistema Difuso, del menú desplegable **Defuzzification method** se selecciona uno de los métodos de conversión mencionados. Entonces seleccionar **File >> Save** para guardar la lógica difusa diseñada.

Como se indico al inicio de este capítulo, LabView proporciona otra posibilidad de crear la Lógica Difusa, esta es mediante la programación del Instrumento Virtual (VI). Sin embargo, en la mayoría de los casos, LabView recomienda usar el Diseñador del Sistema Difuso para diseñar primero la lógica difusa y después usar la programación del Instrumento Virtual para hacer las modificaciones a la lógica mediante la programación. Para ello se procede de la manera siguiente:

Antes de modificar la lógica difusa diseñada, se debe abrir primero la misma y observar la lógica diseñada en el Diseñador del Sistema Difuso.

Para abrir y observar la Lógica Difusa en el Diseñador del Sistema Difuso. Se deben ejecutar los pasos siguientes:

1. Seleccionar **Tools >> Control Design and Simulation >> Fuzzy system Designer** para desplegar el Diseñador del Sistema Difuso.
2. Seleccionar **File >> Open**.
3. En la caja de diálogo del archivo que aparece, navegue al directorio labview\... y seleccione el archivo nombre_archivo.fs.
4. Pulse el botón OK. El archivo nombre_archivo.fs del sistema difuso aparece en el Diseñador del Sistema Difuso.
5. Pulse la etiqueta Rules del Diseñador del Sistema Difuso para desplegar la página de las Reglas.

6. Para ver el Antecedente conjuntivo, el Grado de apoyo y la implicación del Consecuente para una regla respectivamente se debe hacer click sobre la misma en la lista de las Reglas.

7. Para cerrar el Diseñador del Sistema Difuso se debe pulsar el botón Close.

En se describen la manera de modificar el sistema usando los Instrumentos Virtuales (VI). Los tópicos que son tratados en ese documento son:

- Como cargar el Sistema Difuso
- Como modificando una Variable Lingüística
- Como modificar las Funciones de pertenencia
- Como modificar una Regla
- Como crear los Antecedentes
- Como crear los consecuentes
- Como combinar los Antecedentes y Consecuencias para una Regla
- Como guardar los cambios realizados a la Lógica Difusa

El desarrollo anterior demuestra las facilidades que proporciona el paquete de programas de LabView para el diseño de un sistema de toma de decisiones mediante un sistema experto o un controlador difuso.

La propuesta que se presenta en esta tesis contiene dos módulos principales, que son:

- a) Módulo de inicio o interfaz principal

Este módulo contiene el programa que controla todos los subprogramas y es la interfaz gráfica principal del instrumento.

b) Módulo de balanceo dinámico en dos planos

Este módulo permite balancear rotores rígidos que, debido a sus dimensiones y/o velocidad de funcionamiento, se deben balancear en dos planos de corrección.

3.3 Diseño de los Bloques de Instrumentos Virtuales de adquisición de datos y manejo de puertos.

Como quedó definido, para determinar el peso y la posición donde hay que introducirlo para balancear el rotor a prueba se requiere conocer la amplitud de la señal de vibración y el retardo que tiene el valor máximo de esta señal respecto de la posición de un pulso, dado por un sensor estroboscópico. Las señales de los sensores de vibración son sinusoidales con algo de ruido, o sea, son señales “débilmente ruidosas”.

Ambas señales de los sensores, la sinusoidal y el pulso estroboscópico, tienen la misma frecuencia de repetición, dada por la velocidad seleccionada por un operador que generalmente se toma entre 100 y 300 rpm e introducida en la máquina como dato, pero el valor máximo de la primera está retardado respecto de la posición del pulso (desfasaje) según el desbalance del rotor. De igual manera la magnitud de ese valor máximo depende del desbalance del rotor a prueba.

Como se balancea en dos planos, se tienen dos sensores de vibraciones en los pedestales de la instalación donde se ubica el rotor a comprobar por lo que tanto el

hardware como el software se componen de dos “canales” iguales y al diseñar uno queda diseñado el otro “canal”.

En la introducción a este capítulo se analizaron diferentes diagramas de bloques que tiene LabView en sus bibliotecas de bloques denominados como detectores de pico. Los que fueron analizados, de alguna manera, no cumplen las expectativas de poder medir el valor máximo de las señales sinusoidales generadas por los sensores de vibraciones y, a la vez, entregar la posición de este valor máximo con respecto de la posición del pulso del sensor estroboscópico. Estas operaciones pudieran realizarse con ellos solo si se complica la solución.

LabView proporciona otro diagrama de bloque llamado “Waveform peak detector” cuya configuración se muestra En la Figura 3.9.

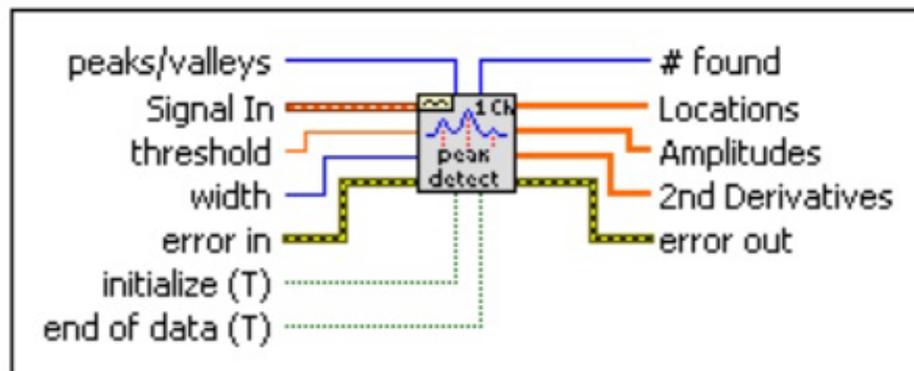


Figura 3.9 Detector de pico de forma de onda.

Las señales de entrada y salida de este diagrama de bloque se explican a continuación. La entrada “peaks/valleys” indica al bloque que se desea encontrar los picos o encontrar los valles o valores mínimos. Si se desea encontrar los picos o máximos esta

entrada debe ponerse en nivel “bajo” o cero (0). Si se desea determinar los valores mínimos se aplica el valor “alto” o uno (1).

Ya que el interés es por los valores máximos, en lo adelante solo se tratan las posibilidades del bloque para tales valores.

La entrada “Signal in” debe contener la forma de onda en la cual deben encontrarse los picos.

La entrada “Threshold” (umbral) instruye al instrumento VI sobre cuándo debe ignorar los picos (máximos) que sean muy pequeños. Se ignoran cuando la amplitud detectada es menor que el umbral.

La entrada “width” (ancho) especifica el número de puntos de datos consecutivos a emplear en el ajuste de curvas por mínimos cuadrados.

La entrada “erro in” describe las condiciones de error que ocurren antes de que comience a funcionar el VI. Por defecto es “no error”.

La entrada “initialize” cuando esta en TRUE (valor predeterminado), provoca que el VI procese el primer bloque de datos. Los VI requieren algún arreglo interno que deba hacerse al inicio para el funcionamiento apropiado.

La entrada “end of data” (T), cuando esta en TRUE (valor predeterminado), provoca que los VI procesen el último bloque de datos. Los VI limpian los datos internos después de que el último bloque se ha procesado.

La salida “#found” es el número de picos encontrados en el bloque de datos que fue procesado y define la dimensión de los arreglos que están en las salidas Locations, Amplitudes, y 2ª Derivada.

La salida “Locations” contiene los índices de todas las ubicaciones de los picos descubiertos en el bloque actual de datos. Ya que el algoritmo de detección del máximo usa el ajuste de curvas “mínimo cuadrático” para encontrar los picos, lo que realmente hace es interpolar entre los puntos de los datos. Por consiguiente, los índices no son enteros. De otra manera, los picos encontrados no son puntos necesariamente reales en los datos de entrada por lo que pueden estar en fracciones de un índice y con amplitudes no encontradas en el arreglo de datos de entrada.

La salida “Amplitudes” contiene las amplitudes de los picos encontrados en el bloque actual de datos.

La salida “2nd Derivatives” proporciona la medición de la segunda derivada de la amplitud en cada uno de los picos encontrados en el bloque actual de datos. La segunda derivada proporciona una medida de cuan agudo es el pico. En este caso los valores son todos negativos.

Las ideas para implementar el dispositivo en LabView son las siguientes:

1. Aplicar la señal del sensor de vibración filtrada y seleccionada o seccionada por bloques como se muestra en la Figura 3.8.

2. Emplear “Waveform Peak Detector for 1 Chan” para estimar la amplitud y el retardo.
3. Seleccionar “peaks”
4. Aplicar por “initialize (T)” la señal adecuada sincronizada con el cambio a “uno” del pulso rectangular mostrado
5. Aplicar por “EndOfData (T)” la señal adecuada “cero”

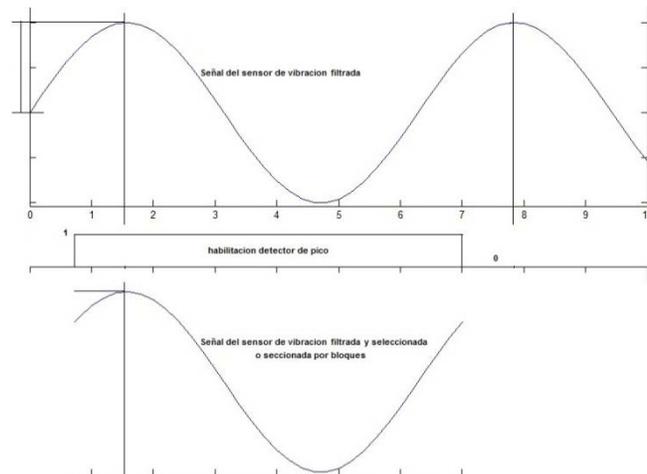


Figura 3.10 Selección de la forma de onda propuesta

La Figura 3.10 muestra de manera gráfica las ideas antes expuestas. En el primer gráfico se muestra la señal aplicada desde los sensores de vibración, filtrada. En el segundo gráfico se muestra un pulso rectangular que debe formarse con los pulsos del sensor estroboscópico, o sea de duración igual al periodo de estos impulsos y que se corresponde con la velocidad seleccionada por el operador. De esta manera el bloque de señal que será procesado solo contendrá un máximo. El tercer gráfico muestra la señal que debe ser aplicada a la entrada “Signal in”.

Se requiere el muestreo simultáneo para las entradas analógicas desde una tarjeta de adquisición de datos.

3.4 Diseño de los Bloques de Instrumentos Virtuales gráficos en coordenadas polares.

LabVIEW proporciona la posibilidad de emplear gráficos en coordenadas polares. Se analiza ahora las prestaciones disponibles en tal diagrama de bloque.

La Figura 3.11 muestra el diagrama de bloque y los requerimientos para su funcionamiento.

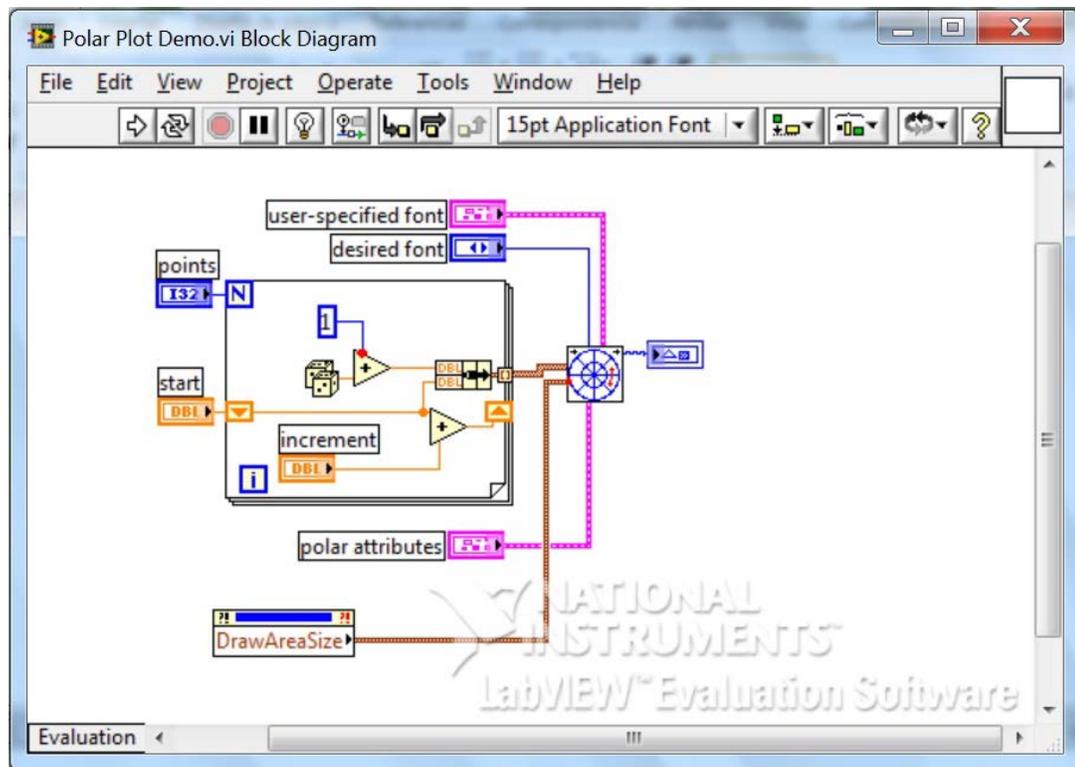


Figura 3.11 Diagrama de Bloque del elemento grafico propuesto

Las entradas al diagrama son las siguientes:

user-specific font - especifica las características de la fuente para el texto en el dibujo.

Los VI ignoran esta entrada a menos que se especifique por el usuario.

Font name - especifica la fuente a usar para el texto. Si se escribe mal el nombre de la fuente o se especifica una fuente que no se instala en la computadora, el sistema operativo selecciona una fuente.

Size - especifica el tamaño de la fuente en los puntos

Strikeout? - especifica si el texto aparece en el ~~strikeout~~. Si Strikeout? es TRUE, el texto aparece en el ~~strikeout~~

Italic? especifica si el texto aparece en *itálica*. Si italic? Es TRUE, entonces el texto aparecerá en este tipo de fuente

El bloque proporciona otras facilidades para la escritura de textos, como son subrayar el texto, sombreado, mayúsculas, etc.

Desired Font – especifica la fuente deseada del texto

0 Fuente especificada por el usuario

1 Fuente de la aplicación (valor predeterminado)

2 Fuente del sistema

3 Fuente del diálogo

Picture – es el dibujo que se desea introducir para la escala. Por defecto es un cuadro vacío

Data Array – es un arreglo de puntos, donde cada punto es un agrupamiento que contiene la magnitud y la fase en grados que especifican el formato y la precisión para la escala.

Magnitude - es la distancia de la línea recta desde el punto al centro del círculo.

Phase - es el ángulo (medido en grados) entre el semi eje positivo x y una línea recta desde el centro del círculo al punto.

Dimensions - especifica el punto izquierdo superior del mapa de bits.

Width - especifica la coordenada horizontal que aumenta hacia la derecha.

Height - especifica la coordenada vertical que aumenta hacia abajo.

Top left point - especifica en las coordenadas dónde poner la esquina superior izquierda de la imagen en el nuevo cuadro.

x es la coordenada horizontal, la cual se incrementa hacia la derecha.

y es la coordenada vertical la cual se incrementa hacia abajo.

Polar attributes – describe el formato para la rejilla y para la escala del gráfico polar e indica como el gráfico maneja los datos negativos, si los mantiene o los dibuja.

Maximum - especifica el valor máximo para la escala.

Mínimum - especifica el valor mínimo para la escala.

Presenta además otras posibilidades para escala logarítmica, color de la rejilla, el color del grafico, la precisión para la escala, formato numérico, etc.

Otro de los parámetros de interés es:

Visible section – este parámetro especifica que cuadrante o cuadrantes se quieren ver, para lo cual se debe elegir entre:

- 0 cuadrante superior derecho
- 1 cuadrante inferior derecho
- 2 cuadrante inferior izquierdo
- 3 cuadrante superior izquierdo
- 4 cuadrantes superiores derecho o izquierdo
- 5 cuadrantes inferiores derecho o izquierdo
- 6 cuadrantes superior e inferior derechos
- 7 cuadrantes superior e inferior izquierdos
- 8 todos los cuadrantes

La salida “new picture” es el dibujo que contiene al gráfico y puede ser aplicada a otra entrada para adicionarle más instrucciones al dibujo.

3.5 Diseño de pruebas

En los numerales anteriores se han analizado los diagramas en bloques que presenta LabView y que permiten implementar la aplicación principal de determinación de los pesos y ángulos donde se deben introducir los mismos para llevar a vías de hecho el balanceo dinámico en dos planos mediante la adquisición de datos de sensores analógicos, procesamiento de los datos de amplitud y fase de las señales de los sensores de vibración, tratamiento mediante lógica difusa y la representación gráfica de los resultados.

Para verificar el funcionamiento del diseño propuesto se diseña la siguiente Prueba de Funcionamiento.

1. crear señales sinusoidales y rectangulares de frecuencias entre 100 y 300 rpm desplazadas una de otras de tal manera que puedan ser establecidos los

parámetros de amplitud y el retardo de tiempo entre la señal sinusoidal y la señal rectangular.

2. aplicar las señales de prueba al diagrama de bloque “detector de pico de forma de onda”
3. aplicar los datos de salida del “detector de pico de forma de onda” a la lógica difusa
4. graficar los resultados.

CAPÍTULO IV:

ANÁLISIS DE LOS RESULTADOS EXPERIMENTALES.

4.1. Análisis de los resultados experimentales con el balanceador difuso

Después de realizar un análisis minucioso de las funciones que brindan los Instrumentos Virtuales (en inglés VI) de LabView quedo determinado la composición de la sección de Medición de Amplitud de la vibración. La composición de esta sección se muestra en la Figura 4.1 y se explica a continuación.

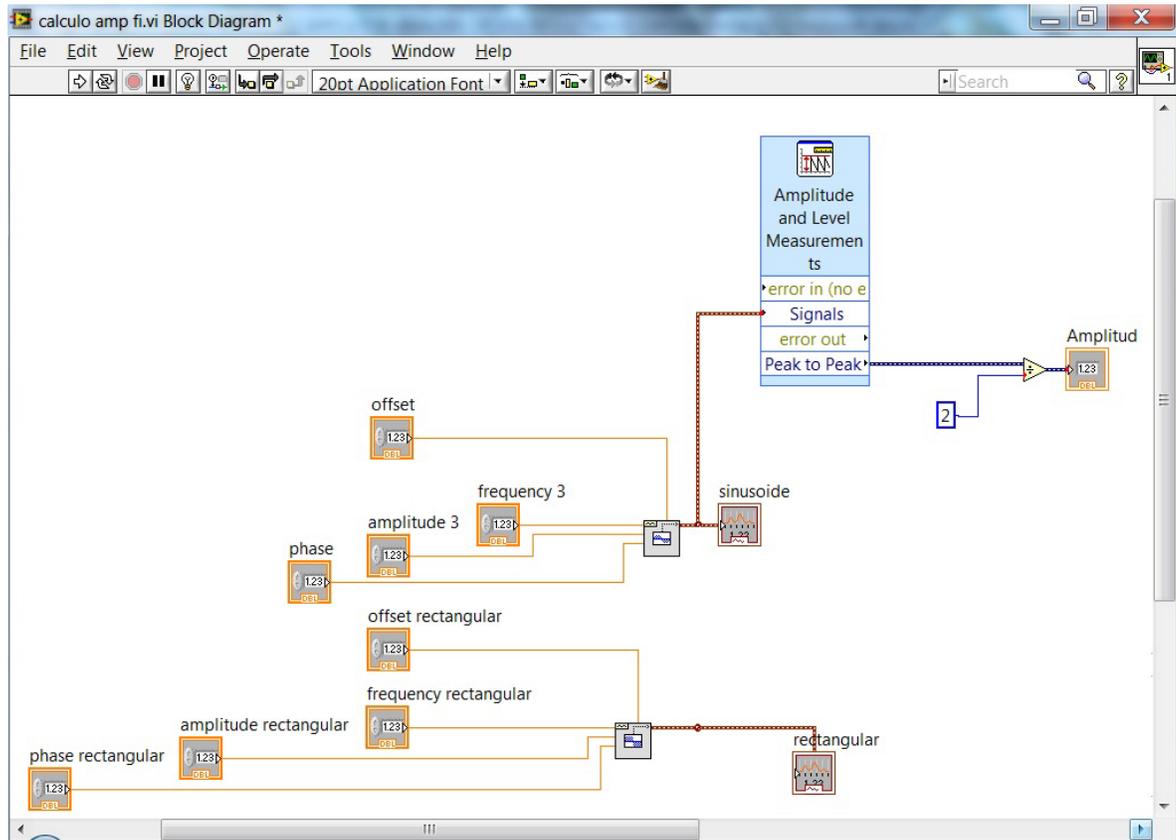


Figura 4.1 Sección de Medición de la Amplitud de la Vibración.

Para determinar la amplitud de la vibración se emplea el VI denominado “Amplitude and Level Measurement” (Medición de Amplitud y Nivel).

Este bloque tiene la entrada “Signals”, a la cual se aplica la señal filtrada de los sensores de vibración. En el bloque, además se selecciona la opción de medición pico a pico, con la cual se obtiene un valor igual al doble de la amplitud.

Para determinar el valor de la amplitud, será necesario introducir un VI de la Paleta de Funciones (Functions Palet) en los VI Express y a su vez dentro de los VI Arithmetic & Comparison encontrado en Express Numeric y llamado “DIVIDE”.

En la Figura 4.2 se muestra el Panel Frontal donde se ofrece la medición de la amplitud en el VI “Indicador” numérico.

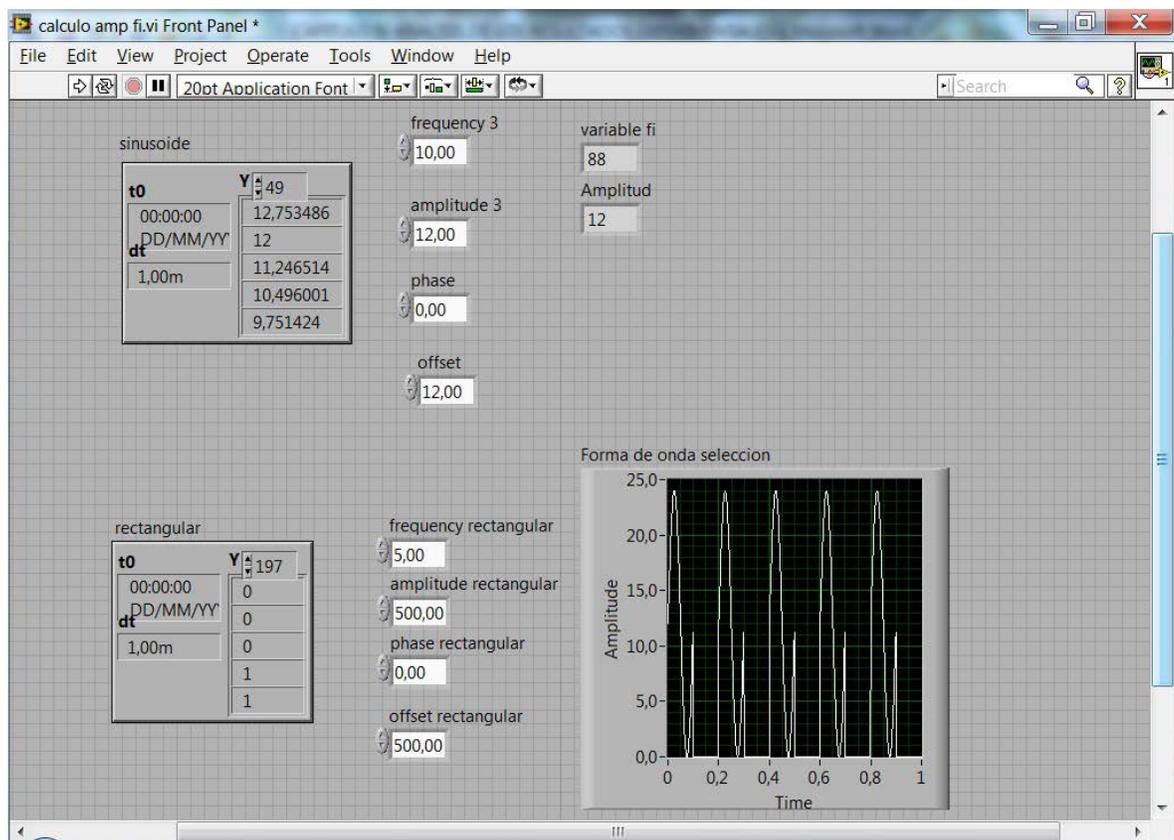


Figura 4.2 Panel Frontal de medición de la amplitud de la vibración para un plano de balanceo.

Como se observa, se ha establecido la amplitud de la sinusoide de prueba en 12 y la medición de la misma en el indicador numérico es también de 12.

Para la medición del desplazamiento del máximo de la vibración respecto del cambio a uno de la señal rectangular se ha diseñado la estructura que se muestra en la Figura 4.3 y explicada a continuación:

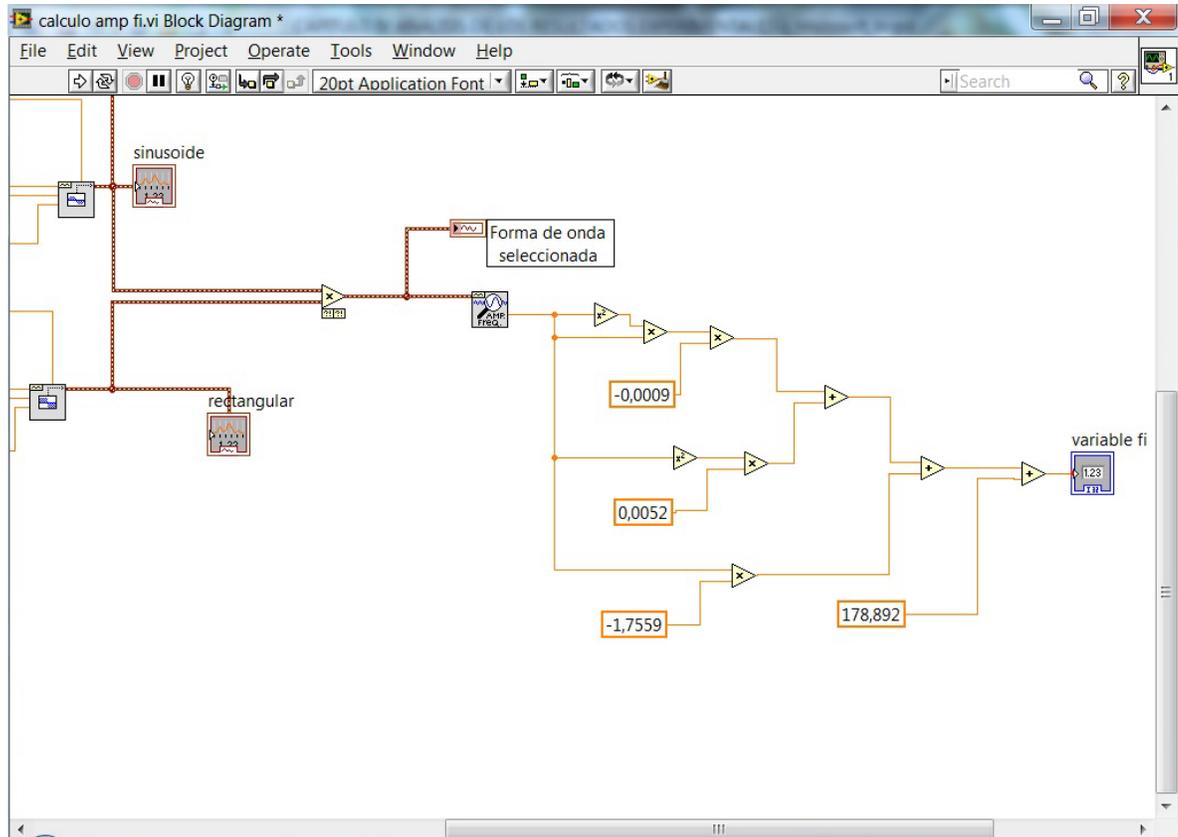


Figura 4.3 Medición del desplazamiento del máximo de la señal de los sensores de vibración.

Como se observa, la señal imitada de los sensores de vibración y la señal, denominada en lo adelante “Señal Rectangular 0/1” se aplican a un bloque multiplicador. Este bloque multiplicador se encuentra también ubicado en la Paleta de Funciones (Functions Palet) en los VI Express y a su vez dentro de los VI Arithmetic & Comparison encontrado en Express Numeric y llamado “Multiply”. La idea que se maneja con esta

configuración es crear un bloque selector de una porción de la señal filtrada de los sensores de vibración donde se ubica el máximo de la señal de los sensores.

La selección se logra haciendo variar la “Señal Rectangular 0/1” entre los valores cero y uno de manera que mientras esta señal tenga el valor cero no se tendrá valores de señal alguno a la salida del VI “Multiply”. Mientras que cuando esta señal tenga el valor uno se deja pasar a la salida de dicho VI la parte de señal de interés para la medición del desplazamiento del máximo respecto del instante de tiempo en que pasa a “1” la señal filtrada de los sensores de vibración.

Esta complejidad surge porque Labview no proporciona ningún VI capaz de calcular el desfase entre el valor máximo de una señal sinusoidal y el cruce por cero de una señal de referencia, solo proporciona la medición de fases clásica entre los cruces por cero de dos señales y este no es el caso en la aplicación que se diseña.

Para determinar la fase de la señal seleccionada se ha empleado el Instrumento Virtual denominado “Extract Single Tone Information VI” el cual toma la señal de entrada y encuentra el tono simple con la mayor amplitud y devuelve la fase de esta señal, entre otros parámetros. Este valor de fase está en correspondencia con el desplazamiento del máximo de la señal de los sensores.

Para determinar con mayor exactitud el desplazamiento del máximo se ha realizado un ajuste de curva. Este ajuste de curva proporciona una desviación estándar de 3°

Los bloques involucrados en el ajuste de curvas se muestran también en la Figura 4.3. El resultado de ajustar la curva se muestra en la Figura 4.4 y obedece a la relación polinomial siguiente:

$$P = -0.0009 * (\text{medicion})^3 + 0.0052 * (\text{medicion})^2 - 1.7559 * (\text{medicion}) + 178.8920$$

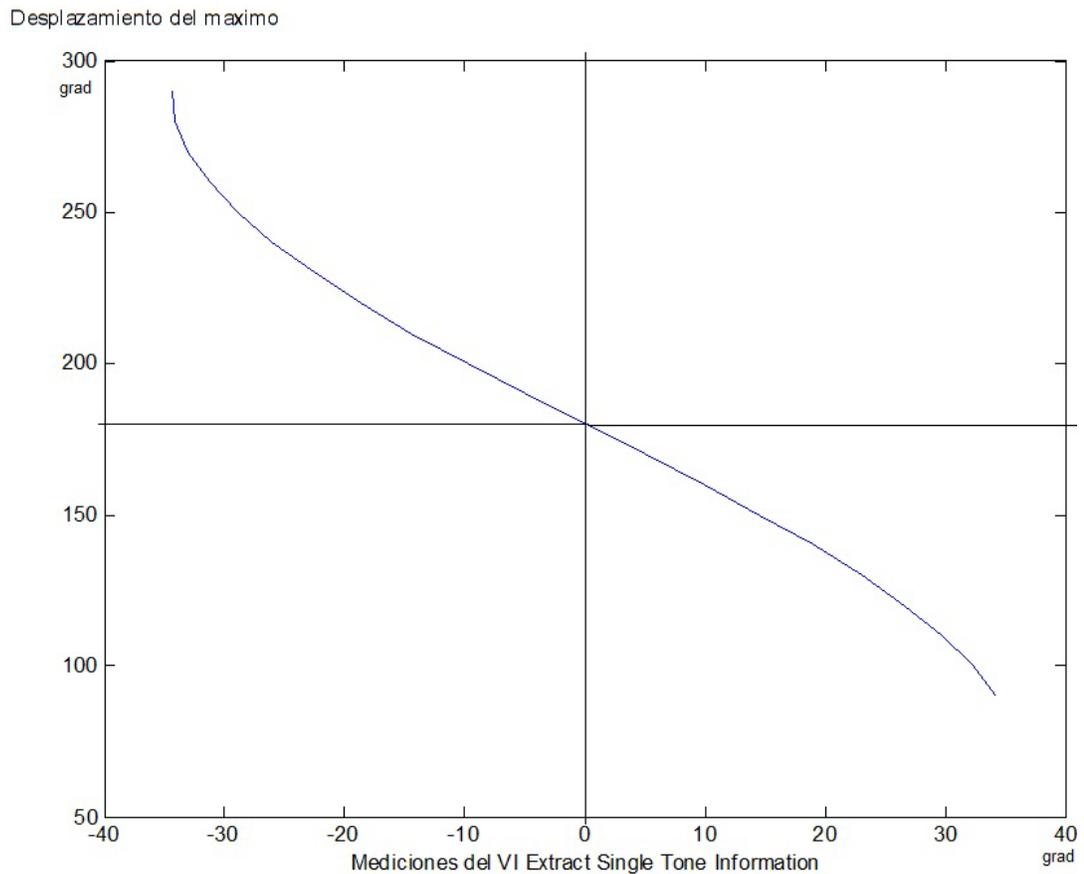


Figura 4.4 Relación entre la medición la fase del tono simple, en grados, y el desplazamiento del máximo también en grados.

En la Figura 4.2 se mostraron los resultados numéricos (valor fi) del desplazamiento del máximo de la señal de los sensores cuando el cruce por cero de esta señal se encuentra en fase con el cambio a uno de la “Señal Rectangular 0/1”.

En la Figura 4.5 se muestra el resultado numérico (valor fi) del desplazamiento del máximo de la señal de los sensores cuando el cruce por cero de esta señal está desfasado en -30° respecto del cruce por cero de la “Señal Rectangular 0/1”.

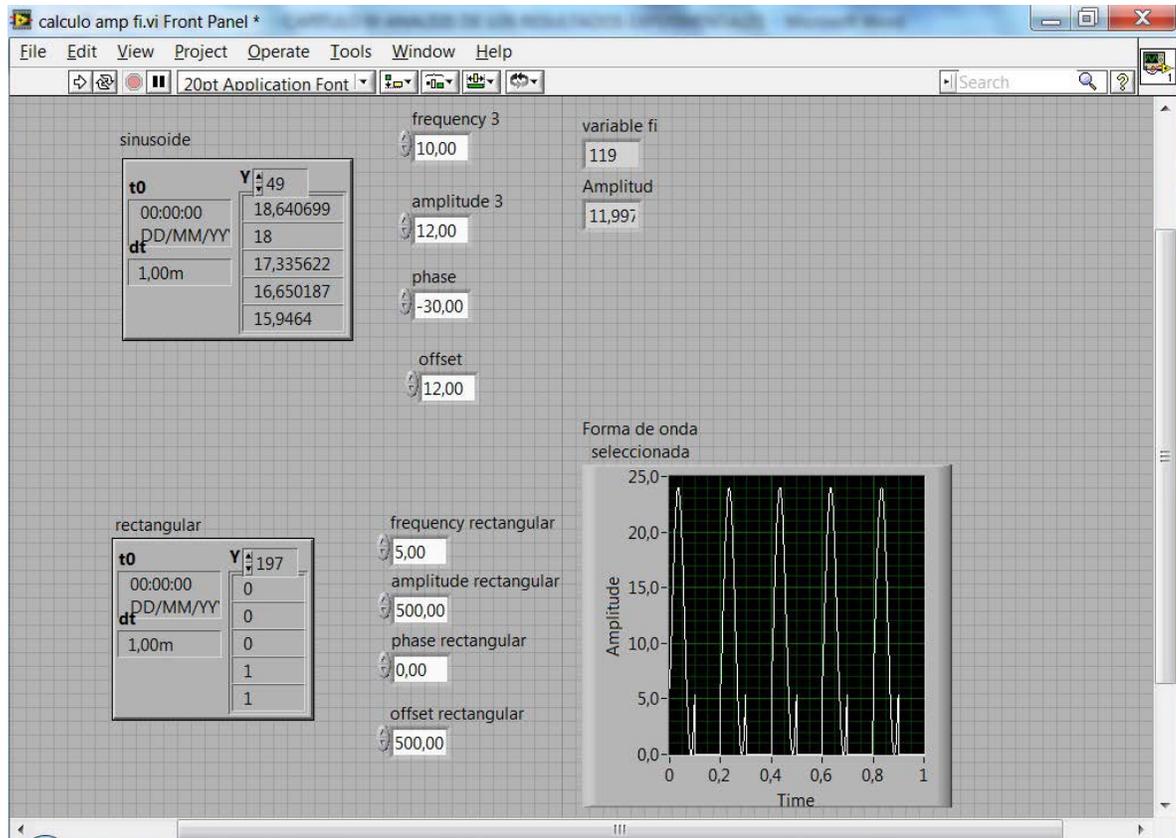
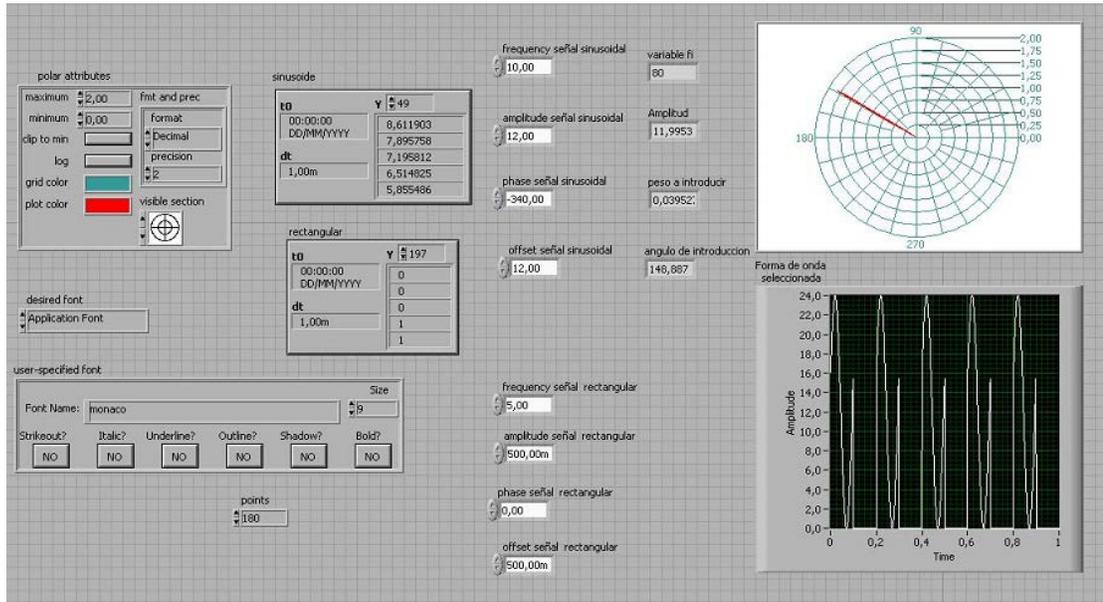


Figura 4.5 Medición del desplazamiento del máximo

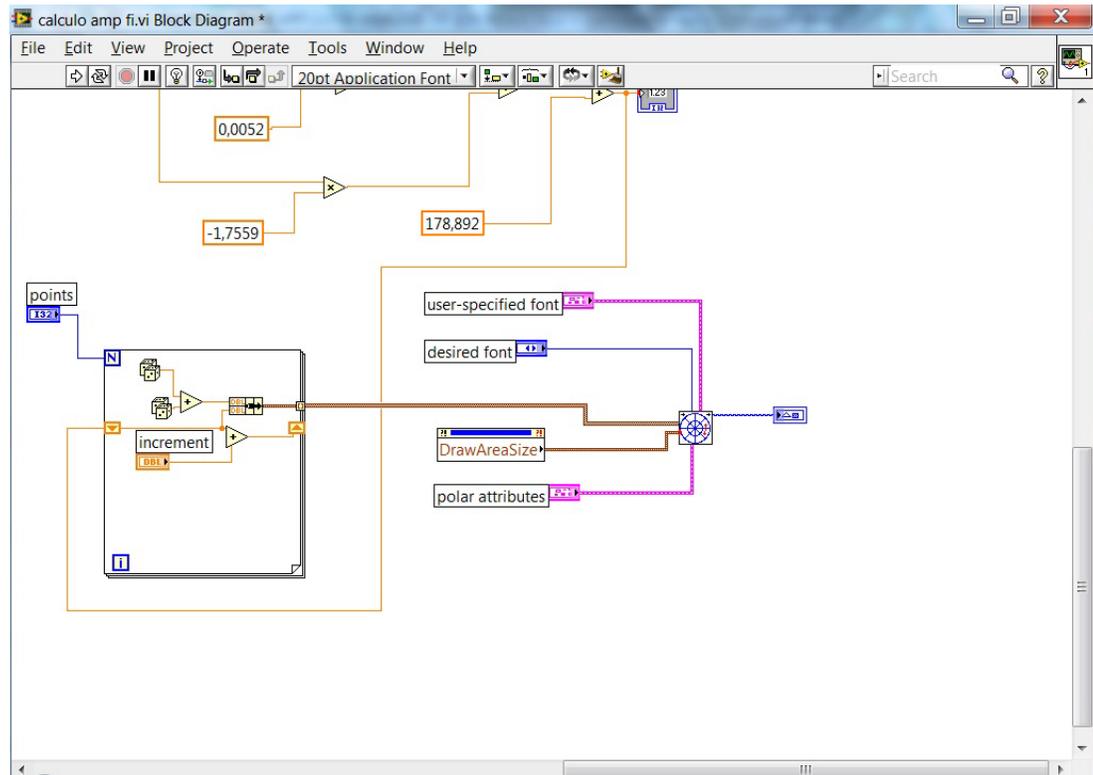
Para la representación del valor del peso a introducir para el balanceo dinámico en dos planos y del ángulo donde debe introducirse este peso se realizan las pruebas mostradas en las Figuras 4.6 y 4.7, donde se muestra la representación de los ángulos mediante una línea recta de color rojo en diferentes magnitudes de ángulos.

La composición del VI requerido para tal representación se muestra en la Figura 4.8. Los elementos que conforman esta sección de representación de los valores de ángulos son:

- Un lazo for (for loop) el cual se ejecuta n veces donde n es el numero asignado al contador ubicado en la esquina superior izquierda y nombrado como N.
- El lazo tiene en su interior un VI de números aleatorios para formar un segmento de recta de puntos rojos en un radio ubicado en el ángulo indicado por el valor numérico aplicado a la entrada del lazo for.
- La entrada del lazo for es un registro de desplazamiento, con el cual se mantiene invariable el valor del ángulo inicial dado por la lógica difusa.
- Dos generadores de números aleatorios con los cuales se forma la línea radial desde el valor cero hasta el valor 2.
- Dos VI sumadores.
- Una función para formar un paquete, la cual ensambla un agrupamiento (cluster) a partir de elementos individuales, que en este caso son números.
- Un VI de grafico polar con su indicador respectivo.



Figuras 4.6 Panel Frontal del Grafico Polar



Figuras 4.7 Diagrama en Bloques del Indicador de ángulo propuesto para la máquina de balanceo dinámico.

El diseño de la lógica difusa se lleva a cabo según lo explicado en el Capítulo 3. A continuación se muestran las pantallas del diagrama en bloques que muestran los bloques empleados para la lógica difusa.

En la Figura 4.8 se muestran los bloques que realizan los cálculos de la lógica diseñada. En la Figura 4.9 se muestran los bloques que cargan la estructura de la lógica diseñada en el VI. Estos VI se encuentran fuera del lazo while.

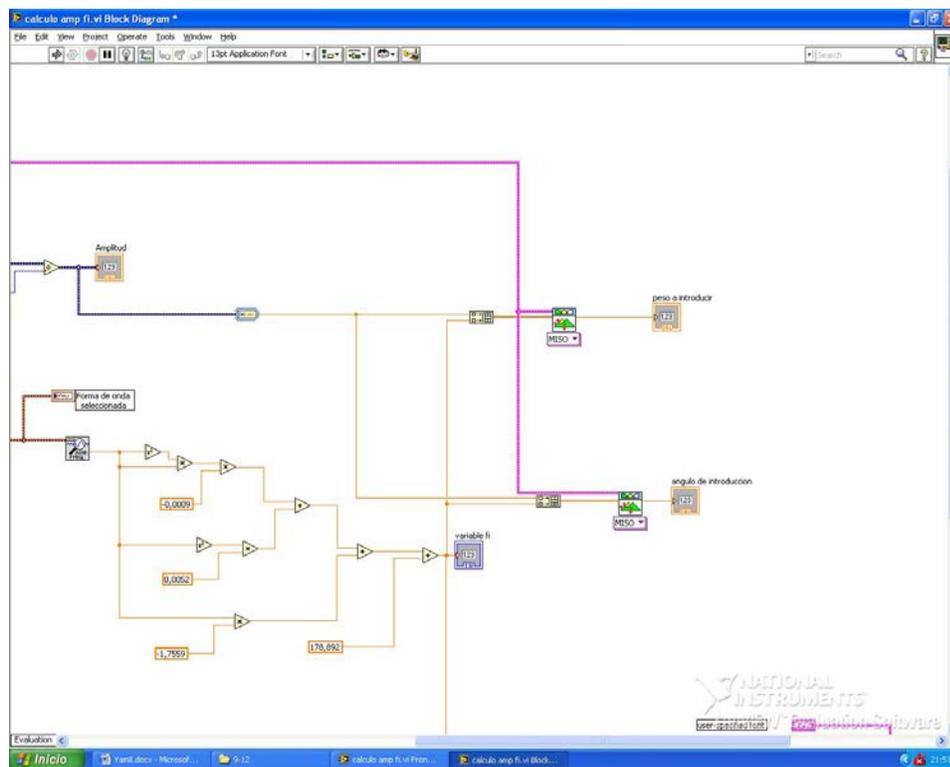


Figura 4.8 VI del controlador difuso del tipo Múltiples entradas una salida (MISO)

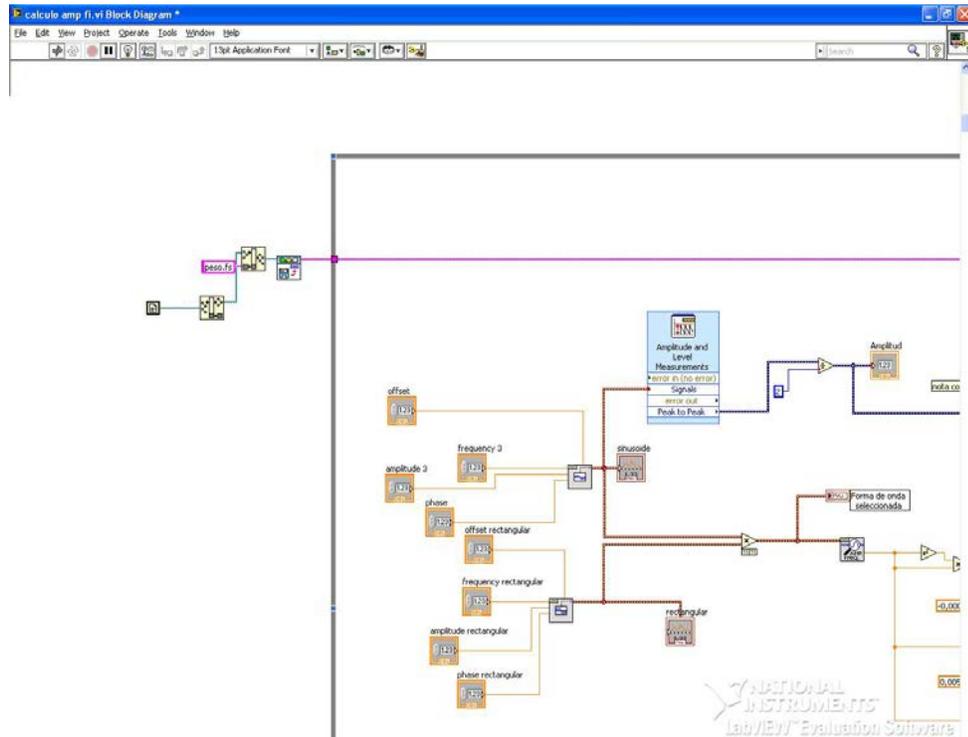


Figura 4.9 Instrumento Virtual para cargar la Lógica Difusa

4.2 Resultados de las pruebas

Las pruebas fueron realizadas mediante un VI en el cual se crean las señales de prueba del dispositivo para la determinación del valor del peso de compensación del desbalance y ángulo donde debe ser introducido dicho peso.

Los resultados muestran que la lógica difusa diseñada en el Capítulo 2 e introducida en un VI de LabView determina con un error de 0.0718 gramos en el valor del peso con un error menor que 10^0 en la determinación del ángulo donde debe ser introducido este peso.

Por otro lado, se requiere de la tarjeta de adquisición de datos para dar una evaluación definitiva del instrumento diseñado.

CONCLUSIONES Y RECOMENDACIONES:

5.1 Conclusiones

En la presente tesis se logró diseñar un instrumento virtual para el balanceo de rotores con base en un sistema difuso entrenado para proporcionar los datos requeridos para el balanceo dinámico de los rotores de motores de inducción a partir de la información recogida por sensores de la Máquina Balanceadora Horizontal MC-5 marca Hofmann variante HK11B.

Mediante experimentación y cálculo de los parámetros peso y ángulo de compensación del desbalance, se eligió la base de datos adecuada para la identificación del modelo difuso, lo cual permitió la elección del modelo difuso.

A través del entrenamiento del modelo se logró identificar los parámetros de la lógica difusa, que proporcionen los datos de: peso y ubicación angular del peso, en el rotor a balancear.

Se consiguió además diseñar la instrumentación virtual requerida para la adquisición de datos, identificación difusa, determinación del peso y ubicación del mismo en el rotor a balancear.

Los objetivos alcanzados evidenciaron la existencia de una lógica difusa en el espacio de lógicas del tipo Takagi-Sugeno, con funciones de pertenencia gaussianas, y funciones singleton en el consecuente, que realiza los cálculos del peso balanceador y el ángulo de introducción del mismo, determinando las relaciones funcionales entre los datos de entrada y salida.

Los resultados de la revisión bibliográfica efectuada demuestran el interés de la comunidad científica internacional para determinar modelos matemáticos confiables y estables a partir de las mediciones a diferentes variables, para abordar el problema del desbalance de los rotores de las máquinas rotatorias.

Se estudiaron los trabajos relevantes en el área dedicada a la lógica difusa como Aproximador Universal de Funciones. Se destacan los trabajos dedicados al entrenamiento de tales lógicas difusas para el ajuste de los parámetros que las definen.

Se demuestra la existencia de una relación funcional entre los valores de entrada y salida de la máquina de balanceo dado el Método de los Coeficientes de Influencia aplicado en la misma.

De acuerdo a los resultados de las pruebas realizadas se determinó que la lógica difusa más adecuada está compuesta de 9×3 reglas para la determinación del peso a

introducir y de 7x3 reglas para la determinación del lugar, en grados, donde debe ser introducido el peso.

5.2 Recomendaciones

Gestionar los recursos necesarios para la implementación de este diseño de instrumento virtual que permita el balanceo de rotores de motores de inducción mediante la aplicación de lógica difusa, a partir de la información recogida por sensores de la Máquina Balanceadora Horizontal MC-5 marca Hofmann variante HK11B.

Adquirir adicionalmente una tarjeta de adquisición de datos, que conectada al equipo de control, permita obtener con mayor facilidad los valores iniciales, para el cálculo del peso y ángulo de compensación del desbalance.

Continuar el desarrollo y la innovación de la máquina de balanceo dinámico, estudiando todas las prestaciones que proporcionan este tipo de máquinas.

Implementar mediante LabView un nuevo panel frontal que permita la interacción del hombre con la máquina de manera más amigable, y que proporcione la información que actualmente se maneja en los Talleres de Balanceo de Rotores más modernos, de tal manera que se pueda guardar la información en memorias portátiles mediante puertos USB.

BIBLIOGRAFÍA:

1. Al-Tae, M. Theoretical and Experimental Method of Rotating Discs Balancing. 2007
2. Chen, Min-You, Linkens, D.A. Rule-base self-generation and simplification for data-driven fuzzy models. Fuzzy sets and System. 142. 2004. www.elsevier.com/locate/fss
3. Guillaume, S. Designing Fuzzy Systems from Data: An Interpretability Oriented Review. IEEE Transaction on Fuzzy Systems , Vol. 9, No. 3, 2001
4. Kang, Y., Chang, Y.-P., Tseng, M.-H., Tang, P.-H. AND Chang, Y.-F. A modified approach based on influence coefficient method for balancing crank-shafts. Journal of Sound and Vibration. 2000.
5. Knospe, C. R., Tamer, S. M. Experiments in robust control of rotor unbalance response using magnetic bearings. Pergamon. Mechatronics Vol. 7, No. 3. 1997
6. Kosco, B. Fuzzy systems as Universal Approximators. IEEE Transactions on computers. Vol 43 No 11. 1994

7. Lin, T-W., Kang, Y., Wang, Ch-Ch., Chang, Ch-W., and Chiang, Ch-P. An investigation in optimal locations by Genetic Algorithm for balancing flexible rotors-bearing systems. Proceedings of GT2005 ASME Turbo Expo 2005. Nevada, USA. 2005
8. Lu, C-J., Wang, M-C., Huang, S-H. Analytical study of the stability of a two-ball automatic balancer. Mechanical Systems and Signal Processing. Elsevier. 2008.
9. Mahfoud, J., Hagopian, J. Der., N. Lévecque, V. Steffen Jr. Experimental model to control and monitor rotating machines. Mechanism and Machine Theory. Elsevier. Journal homepage: www.elsevier.com/locate/mechmt. 2009.
10. Manka, M. and Uhl, T. Mechatronic design of fault detection isolation and restoration systems for rotating machineries. Mechanism and Machine Theory. 2008 Elsevier Ltd
11. Mendel, J. M. and Mouzouris, G. C. Designing Fuzzy Logic Systems. IEEE Transaction on Circuits and Systems II: ANALOG AND DIGITAL SIGNAL PROCESSING, VOL. 44, NO. 11, 1997
12. Neville F. Rieger, Balancing of Rigid and Flexible Rotors. Stress Technology, Inc. United States Department of Defense. 1986.
13. Pandey, S. and Nakra, B. C. Vibration Monitoring of a Rotor System using RMS Accelerations (m2/s). International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 4 April 2011
14. Passino, K., y Yurkovich, S. Fuzzy control. Addison Wesley Longman, Inc. 1998

15. Pavlovskaja, E.E., Karpenko, E.V., Wiercigroch, M. Non-linear dynamic interactions of a Jeffcott rotor with preloaded snubber ring. Journal of Sound and Vibration 276, 361–379. 2004. Elsevier: www.sciencedirect.com
16. Takagi T., and Sugeno M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-15, No. 1, 1985.
17. Yen, J., Wang, L., and Wayne G. C. Improving the Interpretability of TSK Fuzzy Models by Combining Global Learning and Local Learning. IEEE Transactions on Fuzzy Systems, Vol. 6, No. 4, 1998.
18. Ying, H. Sufficient Conditions on Uniform Approximation of Multivariate Functions by General Takagi–Sugeno Fuzzy Systems with Linear Rule Consequent. IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Human, Vol. 28, No. 4, 1998
19. Zhou, S. and Shi, J. Active Balancing and Vibration Control of Rotating Machinery: A Survey. The Shock and vibration Digest, Vol 33, No. 4, 2001.
20. Horikawa, S., Furuhashi, T., Uchikawa, Y. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Trans. Neural Networks 3 (1992) 801–806.
21. Lin, C.T., Lee, C.S.G. Neural network based fuzzy logic control and decision systems, IEEE Trans. Comput. 40 (1990) 1320–1336.

Bibliografía del Capítulo II

1. Passino and Yurkovich. Fuzzy Control. Addison-Wesley. 1998
2. Zadeh, L.A. Making computer think like people. IEEE Spectrum, 1984.
3. Cordon, O., M.J. Del Jesus and F. Herrera, 1998. Genetic learning of fuzzy rulebased classification systems cooperating with fuzzy reasoning methods. Int. Jr. Intelligent Syst., 13: 1025-1053.
4. Lee, C.W. and Y.C. Shin, 2003. Construction of fuzzy systems using least-squares method and genetic algorithm. Elsevier, 137 (3): 297-323.
5. Sanchez, L. and J. Otero, 2004. A fast genetic method for inducing descriptive fuzzy models. Fuzzy Sets and Systems , 141 (1): 33-46.
6. Shi, Y., Mizumoto, M., Yubazaki, N, and Otani, M. 1996. A learning algorithm for tuning fuzzy rules based on gradient descent method. Proc. of 5th IEEE Int. Conf. Fuzzy Syst., New Orleans, Louisiana, pp: 55-61.
7. Dan, S. Sum normal optimization of fuzzy membership functions. Int. Jr. Uncertain Fuzziness Knowledge-Based Syst., 10 2002
8. Kim, E., M. Park and M. Park,. A new approach to fuzzy modeling. IEEE Trans. Fuzzy Syst, 5 1997.
9. Dan, S.,. H8 estimation for fuzzy membership function optimization. Int. Jr. Approx. Reasoning, 40. 2005.

10. Hatanaka, T., Y. Kawaguchi and K. Uosaki, Nonlinear system identification based on evolutionary fuzzy modeling. IEEE Congr. Evolutionary Computation, 1. 2004
11. Mohamed, I. Non-linear System Identification using Neural Networks Trained with Natural Gradient Decent. Eurasip Jr. Applied Signal Processing, 2003
12. Oh, S., W. Pedrycz and H. Park. Hybrid identification in fuzzy neural networks. Fuzzy sets and systems , 138 2003.
13. Toivonen, H.T. and P.M. Makilaf. Newton's method for solving parametric linear quadratic control problems. Int. Jr. Control, 46. 1987.
14. Khan, L., Anjum, S. and Badar, R. Standard Fuzzy Model Identification using Gradient Methods. World Applied Sciences Journal No. 8. 2010. ISSN 1818-4952
15. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Trans. Syst., Man, Cybern., vol. 15, pp. 116–132, 1985.

ANEXOS:

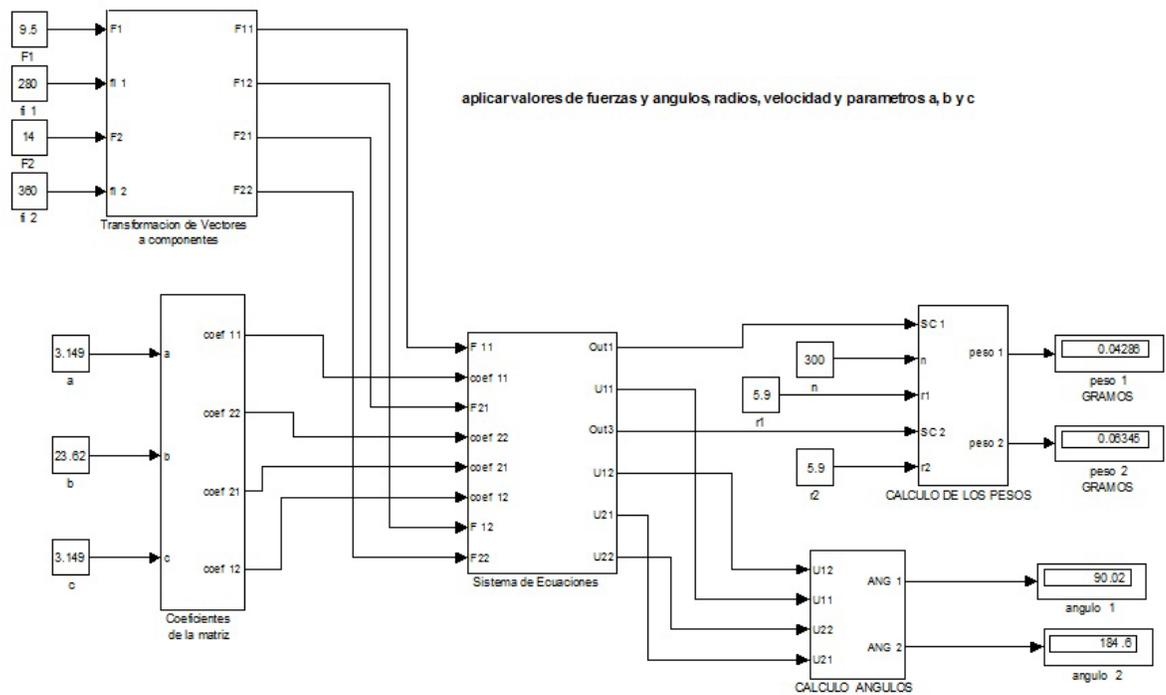
Anexo 1

DATOS TECNICOS DE LA MAQUINA BALANCEADORA HORIZONTAL MC 5 VERSION NHK

DESCRIPCION	MAGNITUD	UNIDADES
PESO MAXIMO DEL ROTOR PARA CARGA SIMETRICA	20	Kg
DIAMETRO MAXIMO DEL ROTOR SOBRE EL LECHO	210	mm
DIAMETRO DE LA PUNTA DE EJE PARA EL MANCAL	5-30	mm
EXTENSION DE LECHO DE LA MAQUINA	830	mm

Anexo 2

Diagrama Simulink



Anexo 3

Programacion en Matlab para el entrenamiento de la lógica difusa

Para el PESO

3x2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INICIALIZACION CENTROS, ANCHOS Y
SALIDA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AW=[2 2 2];%3 anchos fp fuerza
CW=[2.5 7.5 12.5];%3 centros fp fuerza
AQ=[4 4];%2 anchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 6
%inicializacion valores de salida logica borrosa
Y1=[0.002 0.0162 0.03 0.032 0.04 0.06]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SE FORMAN ARREGLOS CON CENTROS Y ANCHOS DE
FP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centros=[];
c=1;
for i=1:3
    for j=1:2
        Centros(:,c) = [CW(i);CQ(j)];    %per columnes
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:3
    for j=1:2
        Anchos(:,c) = [AW(i);AQ(j)];    %per columnes
        c=c+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS DE
ENTRENAMIENTO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
peso=[0.002;0.008084;0.01617;0.02021;0.02425;0.02831;0.03234;0.03638;0.
04446;0.06063];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS
VALIDACION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
```

```

peso_VAL=[0.004042;0.01213;0.01819;0.02223;0.02627;0.03032;0.03436;0.03
84;0.04042;0.05659];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%AJUSTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n1=0.05;
n2=[0.001 0.0009];
n3=[0.001 0.0009];
%yf=zeros(6,1);
aux1=ones(6,1);
aux2=ones(1,2);
NumIter=300;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(6,1);
    gradAnchos=zeros(6,2);
    gradCentros=zeros(6,2);
    for j=1:length(Entrena)
        [yb,b,zl]=sborr(Entrena(j,:)','Y1,Centros','Anchos');
        %siendo yb la salida borrosificada, b el sumatorio de las
        %funciones de pertenencia
        ek=peso(j)-yb;
        Sum_ek=Sum_ek+ek^2;
        coste=coste+(1/(2*j))*Sum_ek;
        grady=grady+(ek/b)*zl;
        gradCentros=gradCentros+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^2);
        gradAnchos=gradAnchos+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*((aux1*Entrena(j,)-Centros').^2)./(Anchos'.^3);
    end;
    %Ahora se realiza la actualizacion de los valores que se desean
ajustar
    Y1=Y1+n1*grady;
    Centros=Centros+aux2'*n2*gradCentros';
    Anchos=Anchos+aux2'*n3*gradAnchos';
    %para visualizar el sumatorio del error
    %i
    %coste
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VALIDACIÓN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Empleando valores facilitados para validar
Ybo=[];
Ybo2=[];
for i=1:length(valida)
    [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Y1,Centros','Anchos');
    Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
end

```

```

    %%%%% Graficos de la salida del peso dado por la maquina y la salida
del
    %modelo difuso identificado
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Valyb, 'm');
hold on;
plot(peso_VAL, 'k');
title('validacion con datos posteriores');
figure;
plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3), 'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL, 'b*');
grid on;
    for i=1:length(valida)

[Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:)','Y1,Centros','Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
    end
    %%%%% Graficos de los valores reales de compensacion y la salida del
modelo
    %difuso en la zona de
datos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb, 'r');
hold on;
plot(peso_VAL, 'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3), 'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL, 'b*');
grid on;

```

5x2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INICIALIZACION CENTROS,ANCHOS Y
SALIDA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AW=[2 2 2 2 2];%5 anchos fp fuerza
CW=[2.5 5 7.5 10 12.5];%5 centros fp fuerza
AQ=[4 4];%2 vanchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 10
%inicializacion valores de salida logica borrosa
Y1=[0.015 0.018 0.021 0.022 0.025 0.03 0.032 0.035 0.04 0.06]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SE FORMAN ARREGLOS CON CENTROS Y ANCHOS DE
FP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centros=[];
c=1;

for i=1:5
    for j=1:2
        Centros(:,c) = [CW(i);CQ(j)];    %per columnes
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:5
    for j=1:2
        Anchos(:,c) = [AW(i);AQ(j)];    %per columnes
        c=c+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS DE
ENTRENAMIENTO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
peso=[0.002;0.008084;0.01617;0.02021;0.02425;0.02831;0.03234;0.03638;0.
04446;0.06063];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS
VALIDACION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
peso_VAL=[0.004042;0.01213;0.01819;0.02223;0.02627;0.03032;0.03436;0.03
84;0.04042;0.05659];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%AJUSTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
n1=0.05;
n2=[0.001 0.0009];
n3=[0.001 0.0009];
aux1=ones(10,1);
```

```

aux2=ones(1,2);
NumIter=350;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(10,1);
    gradAnchos=zeros(10,2);
    gradCentros=zeros(10,2);
    for j=1:length(Entrena)
        [yb,b,zl]=sborr(Entrena(j,:)','Yl,Centros',Anchos');
        %siendo yb la salida borrosificada, b el sumatorio de las
        %funciones de pertenencia
        ek=peso(j)-yb;
        Sum_ek=Sum_ek+ek^2;
        coste=coste+(1/(2*j))*Sum_ek;
        grady=grady+(ek/b)*zl;
        gradCentros=gradCentros+2*(ek/b)*zl.*(Yl-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^2);
        gradAnchos=gradAnchos+2*(ek/b)*zl.*(Yl-
yb*aux1)*aux2.*((aux1*Entrena(j,)-Centros').^2)./(Anchos'.^3);
    end;
    %Ahora se realiza la actualizacion de los valores que se desean
ajustar
    Yl=Yl+n1*grady;
    Centros=Centros+aux2'*n2*gradCentros';
    Anchos=Anchos+aux2'*n3*gradAnchos';
    %para visualizar el sumatorio del error
    %i
    %coste
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VALIDACIÓN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Empleando valores facilitados para validar
Ybo=[];
Ybo2=[];
for i=1:length(valida)
    [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Yl,Centros',Anchos');
    Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
end
%%%%% Graficos de la salida del peso dado por la maquina y la salida
del
%modelo difuso identificado
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Valyb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos posteriores');
figure;

```

```

plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;
    for i=1:length(valida)

[Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:)','Y1,Centros',Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
    end
%%%%%Graficos de los valores reales de compensacion y la salida del
modelo
%difuso en la zona de
datos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;

```

7x2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INICIALIZACION CENTROS,ANCHOS Y
SALIDA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AW=[2 2 2 4 2 2 2];%7 anchos fp fuerza
CW=[2.5 4 5 7.5 10 11 12.5];%7 centros fp fuerza
AQ=[4 4];%2 anchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 14
%inicializacion valores de salida logica borrosa
Y1=[0.015 0.016 0.017 0.018 0.019 0.020 0.021 0.022 0.023 0.025 0.03
0.035 0.04 0.06]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SE FORMAN ARREGLOS CON CENTROS Y ANCHOS DE
FP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centros=[];
c=1;
for i=1:7
    for j=1:2
        Centros(:,c) = [CW(i);CQ(j)];    %per columnes
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:7
    for j=1:2
        Anchos(:,c) = [AW(i);AQ(j)];    %per columnes
        c=c+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS DE
ENTRENAMIENTO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
peso=[0.002;0.008084;0.01617;0.02021;0.02425;0.02831;0.03234;0.03638;0.
04446;0.06063];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS
VALIDACION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
peso_VAL=[0.004042;0.01213;0.01819;0.02223;0.02627;0.03032;0.03436;0.03
84;0.04042;0.05659];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%AJUSTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
n1=0.05;
n2=[0.001 0.0009];
n3=[0.001 0.0009];
aux1=ones(14,1);
```

```

aux2=ones(1,2);
NumIter=380;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(14,1);
    gradAnchos=zeros(14,2);
    gradCentros=zeros(14,2);
    for j=1:length(Entrena)
        [yb,b,zl]=sborr(Entrena(j,:)','Yl,Centros',Anchos');
        %siendo yb la salida borrosificada, b el sumatorio de las
        %funciones de pertenencia
        ek=peso(j)-yb;
        Sum_ek=Sum_ek+ek^2;
        coste=coste+(1/(2*j))*Sum_ek;
        grady=grady+(ek/b)*zl;
        gradCentros=gradCentros+2*(ek/b)*zl.*(Yl-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^2);
        gradAnchos=gradAnchos+2*(ek/b)*zl.*(Yl-
yb*aux1)*aux2.*((aux1*Entrena(j,)-Centros').^2)./(Anchos'.^3);
    end;
    %Ahora se realiza la actualizacion de los valores que se desean
ajustar
    Yl=Yl+n1*grady;
    Centros=Centros+aux2'*n2*gradCentros';
    Anchos=Anchos+aux2'*n3*gradAnchos';
    %para visualizar el sumatorio del error
    %i
    %coste
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VALIDACIÓN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Empleando valores facilitados para validar
Ybo=[];
Ybo2=[];
for i=1:length(valida)
    [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Yl,Centros',Anchos');
    Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
end
%%%% Graficos de la salida del peso dado por la maquina y la salida
del
%modelo difuso identificado
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Valyb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos posteriores');
figure;

```

```

plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;
    for i=1:length(valida)

[Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:)','Y1,Centros',Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
    end
%%%%%Graficos de los valores reales de compensacion y la salida del
modelo
%difuso en la zona de
datos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;

```

9x2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INICIALIZACION CENTROS,ANCHOS Y
SALIDA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AW=[3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5];%9 anchos fp fuerza
CW=[2.5 3 4.5 5 6 7.5 10 11 12.5];%9 centros fp fuerza
AQ=[4 4];%2 anchos fp angulo
CQ=[90 270];%2 centros fp angulo
%numero maximo de reglas 18
%inicializacion valores de salida logica borrosa
Y1=[0.015 0.016 0.017 0.0175 0.018 0.0185 0.019 0.0195 0.020 0.0205
0.0210 0.022 0.023 0.025 0.03 0.035 0.04 0.06]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SE FORMAN ARREGLOS CON CENTROS Y ANCHOS DE
FP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centros=[];
c=1;
for i=1:9
    for j=1:2
        Centros(:,c) = [CW(i);CQ(j)];    %per columnes
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:9
    for j=1:2
        Anchos(:,c) = [AW(i);AQ(j)];    %per columnes
        c=c+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS DE
ENTRENAMIENTO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
peso=[0.002;0.008084;0.01617;0.02021;0.02425;0.02831;0.03234;0.03638;0.
04446;0.06063];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS
VALIDACION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
peso_VAL=[0.004042;0.01213;0.01819;0.02223;0.02627;0.03032;0.03436;0.03
84;0.04042;0.05659];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%AJUSTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
n1=0.05;
n2=[0.001 0.0009];
n3=[0.001 0.0009];
aux1=ones(18,1);
```

```

aux2=ones(1,2);
NumIter=500;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(18,1);
    gradAnchos=zeros(18,2);
    gradCentros=zeros(18,2);
    for j=1:length(Entrena)
        [yb,b,zl]=sborr(Entrena(j,:)','Y1,Centros','Anchos');
        %siendo yb la salida borrosificada, b el sumatorio de las
        %funciones de pertenencia
        ek=peso(j)-yb;
        Sum_ek=Sum_ek+ek^2;
        coste=coste+(1/(2*j))*Sum_ek;
        grady=grady+(ek/b)*zl;
        gradCentros=gradCentros+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^2);
        gradAnchos=gradAnchos+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*((aux1*Entrena(j,)-Centros').^2)./(Anchos'.^3);
    end;
    %Ahora se realiza la actualizacion de los valores que se desean
ajustar
    Y1=Y1+n1*grady;
    Centros=Centros+aux2'*n2*gradCentros';
    Anchos=Anchos+aux2'*n3*gradAnchos';
    %para visualizar el sumatorio del error
    %i
    %coste
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VALIDACIÓN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Empleando valores facilitados para validar
Ybo=[];
Ybo2=[];
for i=1:length(valida)
    [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Y1,Centros','Anchos');
    Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
end
%%%% Graficos de la salida del peso dado por la maquina y la salida
del
%modelo difuso identificado
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Valyb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos posteriores');
figure;

```

```

plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;
    for i=1:length(valida)

[Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:)','Y1,Centros',Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
    end
%%%%%Graficos de los valores reales de compensacion y la salida del
modelo
%difuso en la zona de
datos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;

```



```

n3=[0.001 0.0009];
aux1=ones(27,1);
aux2=ones(1,2);
NumIter=500;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(27,1);
    gradAnchos=zeros(27,2);
    gradCentros=zeros(27,2);
    for j=1:length(Entrena)
        [yb,b,zl]=sborr(Entrena(j,:)','Y1,Centros',Anchos');
        %siendo yb la salida borrosificada, b el sumatorio de las
        %funciones de pertenencia
        ek=peso(j)-yb;
        Sum_ek=Sum_ek+ek^2;
        coste=coste+(1/(2*j))*Sum_ek;
        grady=grady+(ek/b)*zl;
        gradCentros=gradCentros+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^2);
        gradAnchos=gradAnchos+2*(ek/b)*zl.*(Y1-
yb*aux1)*aux2.*(aux1*Entrena(j,)-Centros')./(Anchos'.^3);
    end;
    %Ahora se realiza la actualizacion de los valores que se desean
ajustar
    Y1=Y1+n1*grady;
    Centros=Centros+aux2'*n2*gradCentros';
    Anchos=Anchos+aux2'*n3*gradAnchos';
    %para visualizar el sumatorio del error
    %i
    %coste
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VALIDACION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Empleando valores facilitados para validar
Ybo=[];
Ybo2=[];
    for i=1:length(valida)
        [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Y1,Centros',Anchos');
        Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
    end
    %%%%% Graficos de la salida del peso dado por la maquina y la salida
del
    %modelo difuso identificado
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Valyb,'r');
hold on;
plot(peso_VAL,'b');

```

```

title('validacion con datos posteriores');
figure;
plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;
    for i=1:length(valida)

[Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:)','Y1,Centros',Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
    end
%%%% Graficos de los valores reales de compensacion y la salida del
modelo
%difuso en la zona de datos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb,'r');
hold on;
plot(peso_VAL,'b');
title('validacion con datos identificacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),peso_VAL,'b*');
grid on;

```

Para el ÁNGULO

3x7

Programa de ajuste sistema difuso para estimar ángulos con instrucciones de graficar y para la desviación estándar.

```
%programa para estimar estructura de la logica difusa
%requerida para calcular valor de angulo de introduccion
%del peso corrector de desbalance
AW=[2 2 2];%3 anchos fp fuerza
CW=[2.5 7.5 12.5];%3 centros fp fuerza
AQ=[30 30 30 5 30 50 50];
%[20 10 3 3 3 10 20]7 anchos fp angulo
CQ=[30 90 135 180 225 270 315];
%[49 90 135 180 225 270 315]7 centros fp angulo
Y1=[30 45 60 80 90 110 179 225 280 310 335 9 12 10 15 18 25 30 31 32
40]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Centros=[];
c=1;
for i=1:3
    for j=1:7
        Centros(:,c) = [CW(i);CQ(j)];    %por columnas
        c=c+1;
    end
end
Anchos=[];
c=1;
for i=1:3
    for j=1:7
        Anchos(:,c) = [AW(i);AQ(j)];    %por columnas
        c=c+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%DATOS
ENTRENAMIENTO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza=[0.5;2;4;5;6;7;8;9;11;15];
angulo=[1;30;60;90;180;200;250;300;320;360];
Entrena=[fuerza,angulo];
ang=[181;210;240;270;360;20;70;120;140;180];%10 valores de angulos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DATOS VALIDACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fuerza_VAL=[1;3;4.5;5.5;6.5;7.5;8.5;9.5;10;14];
angulo_VAL=[5;45;70;135;175;210;240;280;310;340];
valida=[fuerza_VAL,angulo_VAL];
ang_VAL=[185;225;250;315;355;30;60;100;130;160];
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%AJUSTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
n1=0.05;
n2=[0.001 0.0009];
n3=[0.001 0.0009];
aux1=ones(21,1);
aux2=ones(1,2);
NumIter=500;
for i=1:NumIter
    coste=0;
    Sum_ek=0;
    grady=zeros(21,1);
    gradAnchos=zeros(21,2);
    gradCentros=zeros(21,2);
        for j=1:length(Entrena)
            [yb,b,zl]=sborr(Entrena(j,:)','Y1,Centros',Anchos');
%siendo yb la salida borrosificada, b el sumatorio de las
%funciones de pertenencia
            ek=ang(j)-yb;
            Sum_ek=Sum_ek+ek^2;
            coste=coste+(1/(2*j))*Sum_ek;
            grady=grady+(ek/b)*zl;
            gradCentros=gradCentros+2*(ek/b)*zl.*(Y1-
            yb*aux1)*aux2.*(aux1*Entrena(j,:)-Centros')./(Anchos'.^2);
            gradAnchos=gradAnchos+2*(ek/b)*zl.*(Y1-
            yb*aux1)*aux2.*((aux1*Entrena(j,:)-
            Centros').^2)./(Anchos'.^3);
        end;
%Ahora se realiza la actualizacion de los valores que se desean ajustar
        Y1=Y1+n1*grady;
        Centros=Centros+aux2'*n2*gradCentros';
        Anchos=Anchos+aux2'*n3*gradAnchos';
%para visualizar el sumatorio del error
        i;
        varianza(i)=coste;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VALIDACION
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Ybo=[];
    Ybo2=[];
    for i=1:length(valida)
        [Valyb(i),Valb,Valzl]=sborr(valida(i,:)','Y1,Centros',Anchos');
        Ybo=[Ybo ;[valida(i,:),Valyb(i)]];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%GRAFICOS SALIDA ANGULO MAQUINA Y MODELO DIFUSO%%
    figure;
    plot(Valyb,'m');
    hold on;
    plot(ang_VAL,'k');

```

```

title('validacion con datos posteriores');
figure;
plot3(Ybo(:,1),Ybo(:,2),Ybo(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),ang_VAL,'b*');
grid on;
for i=1:length(valida)
    [Val2yb(i),Val2b,Val2z1]=sborr(valida(i,:),'Yl,Centros',Anchos');
    Ybo2=[Ybo2 ;[valida(i,:),Val2yb(i)]];
end
%%%%% Graficos de los valores reales de compensacion y la salida del
%%% modelo difuso en la zona de datos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(Val2yb,'r');
hold on;
plot(ang_VAL,'b');
title('validacion');
figure;
plot3(Ybo2(:,1),Ybo2(:,2),Ybo2(:,3),'r*');
hold on;
plot3(valida(:,1),valida(:,2),ang_VAL,'b*');
grid on;
i=1:500;
figure(5)
plot(i,varianza)
title('Comportamiento del criterio')
ylabel('Varianza')
xlabel('NumIter')
dif=(ang_VAL)'-Valyb;
disp=(dif.*dif);
unos=ones(10,1);
varianza=0.1*(disp_rel*unos)
desv_stand=sqrt(varianza)
error=[ang_VAL dif']

```

Se probaron variantes con diferentes cantidades de reglas. La variante más acertada es la mostrada en el programa último, con 3x7 reglas. El mayor error de estimación del ángulo para ubicar los pesos es de 8°.