

UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ÁREA DE ESTUDIOS DE POSTGRADO
MAESTRÍA EN MATEMÁTICA Y COMPUTACIÓN

**PARÁMETROS CARACTERÍSTICOS DE SEÑALES DE VOZ UTILIZANDO
WAVELET PACKET Y ACP PARA SU CLASIFICACIÓN MEDIANTE MÁQUINA
DE VECTORES DE SOPORTE**

Autor: Ing. Franklin J. Yusti R.
Tutor: Ing. MSc. Carlos Jiménez.

Bárbula, Julio de 2015

UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ÁREA DE ESTUDIOS DE POSTGRADO
MAESTRÍA EN MATEMÁTICA Y COMPUTACIÓN

**PARÁMETROS CARACTERÍSTICOS DE SEÑALES DE VOZ UTILIZANDO
WAVELET PACKET Y ACP PARA SU CLASIFICACIÓN MEDIANTE MÁQUINA
DE VECTORES DE SOPORTE**

Autor: Ing. Franklin J. Yusti R.

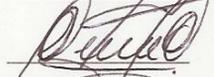
Trabajo presentado ante el Área de Estudios de Postgrado de la Universidad de Carabobo para optar al Título de Magíster en Matemática y Computación.

Bárbula, Julio 2015

UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ÁREA DE ESTUDIOS DE POSTGRADO
MAESTRÍA EN MATEMÁTICA Y COMPUTACIÓN

**PARÁMETROS CARACTERÍSTICOS DE SEÑALES DE VOZ UTILIZANDO
WAVELET PACKET Y ACP PARA SU CLASIFICACIÓN MEDIANTE MÁQUINA
DE VECTORES DE SOPORTE**

Aprobado en el Área de Estudios de Postgrado de la Universidad de Carabobo por los miembros de la Comisión Coordinadora del Programa:

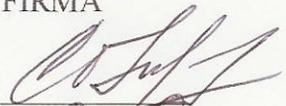
NOMBRE	APELLIDO	CI	FIRMA
<u>CÉSAR</u>	<u>SEIJAS</u>	<u>4.567.093</u>	
<u>Miguel</u>	<u>Fasanella</u>	<u>24554371</u>	
<u>Victor</u>	<u>Barrios</u>	<u>3896992</u>	

Bárbula, Noviembre 2015

UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ÁREA DE ESTUDIOS DE POSTGRADO
MAESTRÍA EN MATEMÁTICA Y COMPUTACIÓN

VEREDICTO

Nosotros, Miembros del Jurado designado para la evaluación del Trabajo Especial de Grado titulado: **PARÁMETROS CARACTERÍSTICOS DE SEÑALES DE VOZ UTILIZANDO WAVELET PACKET Y ACP PARA SU CLASIFICACIÓN MEDIANTE MÁQUINA DE VECTORES DE SOPORTE**, presentado por Franklin J. Yusti R. para optar al Título de **MAGÍSTER EN MATEMÁTICA Y COMPUTACIÓN**, estimamos que *el mismo reúne los requisitos para ser considerado como: APROBADO.*

NOMBRE	APELLIDO	CI	FIRMA
<u>CÉSAR</u>	<u>SEIJAS</u>	<u>4.567.093</u>	
<u>MIGUEL</u>	<u>FASANELLA</u>	<u>24554371</u>	
<u>VICTOR</u>	<u>Barrios</u>	<u>3896992</u>	

Bárbula, Noviembre 2015

AGRADECIMIENTOS

De especial manera y significativamente agradezco en primera instancia a Dios todopoderoso creador del universo y padre de nuestro señor Jesús el Cristo fabricante de yugos, sin su intervención divina no habría sido posible llegar a esta fase de la vida.

De igual modo agradezco a mis padres José Yusti y Zoraida Riera quienes siempre me han brindado su confianza y su apoyo incondicional en todas las etapas de mi vida y esta etapa siguiente no sería la excepción.

Al profesor Ing. Carlos Jiménez, por todo su valioso tiempo invertido al enseñarme desinteresadamente todos esos conocimientos necesarios para cumplir con cada uno de los objetivos de este trabajo de investigación, por ser un gran guía académico y un excelente tutor en este proyecto de investigación.

Al Centro de Análisis y Tratamiento de Señales (CATS) por su importante aporte en lo que respecta a las bases de datos de señales de voces utilizadas en este trabajo de investigación.

ÍNDICE GENERAL

	Pag.
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
RESUMEN	xiv
INTRODUCCIÓN	15
CAPITULO I	
EL PROBLEMA	
Planteamiento del Problema.....	17
Formulación del Problema.....	20
Objetivos de la Investigación.....	20
Objetivo General.....	20
Objetivos Específicos.....	20
Justificación de la Investigación.....	21
CAPITULO II	
MARCO TEÓRICO REFERENCIAL	
Antecedentes.....	24
Bases Teóricas.....	26
La Voz y el Habla.....	26
Aparato Fonador Humano.....	27
Modelo simplificado de las cavidades oral, labial y nasal.....	28
Forma de Onda de la Señal de Voz.....	28
Digitalización de la Señal de Voz.....	29
Características Fundamentales de la Señal de Voz.....	30
Energía y Cruces por Cero.....	30
Frecuencias Formantes.....	34

Tipos de Señales de Voz.....	35
Señal Sonora.....	36
Señal no Sonora.....	37
Señal Plosiva.....	37
Factores que Afectan la Señal de Voz.....	38
Transformada Wavelet.....	41
Función Escala.....	43
Función Wavelet.....	44
Transformada Discreta Wavelet (DWT).....	45
Reconstrucción De Señales (Síntesis).....	54
Descomposición (Análisis) Wavelet Packet.....	63
Análisis de Componentes Principales (ACP).....	65
Máquinas de Vectores de Soporte (SVM).....	66
Clasificación en dos Dimensiones.....	68
Errores de Entrenamiento (Soft Margin).....	69
Consideraciones Alrededor de las SVM.....	70
Función Kernel.....	71
Tipos de Funciones Kernel (Núcleo).....	72
SVM Multiclase.....	73
Caso no Linealmente Separable.....	74

CAPITULO III

MARCO METODOLÓGICO

Tipo y Modalidad de Investigación.....	76
Diseño de la Investigación.....	77
Población y Muestra.....	77
Técnicas e Instrumentos de Recolección de Datos.....	78
Selección de la Base de Datos.....	79
Procesamiento y Análisis de los Datos.....	82

Procedimientos de la Investigación.....	82
Etapa I: Selección y Lectura de las Muestras de Voz.....	83
Etapa II: Determinación de Parámetros.....	84
Etapa III: Análisis Estadístico.....	89
Etapa IV: Clasificación.....	97
CAPITULO IV	
PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS	
Clasificación de Muestras de Voz y Reducción de la Dimensión de Parámetros....	108
CAPITULO V	
CONCLUSIONES Y RECOMENDACIONES	
Conclusiones.....	137
Recomendaciones.....	139
REFERENCIAS BIBLIOGRÁFICAS.....	140
APÉNDICE A.....	143
APÉNDICE B.....	149
APÉNDICE C.....	151
APÉNDICE D.....	155
APÉNDICE E.....	157
APÉNDICE F.....	160
APÉNDICE G.....	165
APÉNDICE H.....	167
APÉNDICE I.....	170

ÍNDICE DE FIGURAS

Figura 2.1: Aparato fonador humano.....	27
Figura 2.2: Forma de onda de la palabra “Explorador”	29
Figura 2.3: Cruces por Cero. a) Señal Continua. b) Señal Discreta.....	31
Figura 2.4: Energía y cruces por cero de la palabra “seis”.....	32
Figura 2.5: Espectro de frecuencia de la palabra 'dos'.....	33
Figura 2.6: Envolvente espectral.....	34
Figura 2.7: Frecuencias formantes.....	35
Figura 2.8: Señal sonora.....	36
Figura 2.9: Señal no sonora.....	37
Figura 2.10: Señal sonora plosiva.....	38
Figura 2.11: Representación de los subespacios V_j , se muestra que están anidados.	45
Figura 2.12: Representa los subespacios W_j y se observa que la intersección entre ellos es vacía y que $V_2 = V_1 \oplus W_1$	47
Figura 2.13: Se muestra como una función wavelet ventaniza a una señal.....	48
Figura 2.14: Representa la grafica del sistema Dubechie8. Función wavelet y su representación como filtro pasaalto, gráficos superiores. Función escala y su representación como filtro pasabajo, gráficos inferiores.....	49
Figura 2.15: Diagrama algorítmico para calcular los coeficientes de aproximación y detalle de una señal cualquiera f que ha sido muestreada.....	52
Figura 2.16: Descomposición multinivel de una señal muestreada con 8 muestras. Se observa la formación del vector wavelet para cada nivel de descomposición, proceso denominado análisis.....	53
Figura 2.17: Se muestra el primer nivel de descomposición de una señal mediante el sistema Haar.....	56
Figura 2.18: Reconstrucción de una señal utilizando el sistema Haar, la señal ha sido descompuesta sólo hasta el primer nivel de descomposición.....	58

Figura 2.19: Descomposición multinivel (análisis) de una señal de $8=2^3$ muestras con sus respectivos vectores wavelet en cada nivel de descomposición, además se observa que $0 < k < 2^j$	61
Figura 2.20: Descomposición Wavelet (análisis multinivel).....	63
Figura 2.21: Descomposición Wavelet Packet.....	64
Figura 2.22: Árbol Binario de Descomposición Wavelet Packet.....	64
Figura 2.23: Clasificación SVM, caso linealmente separable.....	67
Figura 2.24: Clasificación en dos dimensiones, hiperplanos separadores. Hay infinitos hiperplanos posibles.....	68
Figura 2.25: H1 no separa las clases. H2 las separa, pero solo con un margen pequeño. H3 las separa con el margen máximo.....	69
Figura 2.26: Función kernel polinomial SVM 8 polinomial.....	72
Figura 2.27: Svm 10 perceptrón.....	72
Figura 2.28: Clasificación con función kernel de base radial Gaussiana.....	73
Figura 2.29: Conjunto no linealmente separable.....	74
Figura 3.1: Diagrama de bloques del procedimiento metodológico.....	83
Figura 3.2: Contracción Garrote.....	86
Figura 4.1: Clasificación de la señal EAL06AN.NSP, se observa su cercanía al hiperplano separador.....	136

ÍNDICE DE TABLAS

Tabla 1.1: Comparación de los tipos de micrófonos.....	40
Tabla 3.1: Estructura del encabezado de un archivo <i>.NSP</i>	79
Tabla 3.2: Base de datos utilizadas para voces sanas (50 individuos).....	80
Tabla 3.3: Base de datos utilizadas para voces con patologías (99 individuos).....	81
Tabla 3.4: Parámetros estadísticos calculados al PMR de las señales de voz por segmentos y sus respectivos comandos	87
Tabla 3.5: Estructura de los vectores wavelet packet de las señales de voz en los diferentes niveles de descomposición.....	88
Tabla 3.6: Parámetros estadísticos calculados al VWPT de las señales de voz y sus respectivos comandos	88
Tabla 3.7: Variables que contiene el vector de parámetros característicos.....	89
Tabla 3.8: Matriz “xtrainWPT” de parámetros característicos para 20 voces sanas y 20 voces con patologías, las muestras son tomadas de las tablas 3.2 y 3.3.....	91
Tabla 3.9: Matriz de componentes principales referidas a la matriz xtrainWPT.....	92
Tabla 3.10: Matriz de variables estandarizadas “xtrainWPTstd”.....	95
Tabla 3.11: Matriz de variables transformadas Ztrain.....	97
Tabla 3.12: Clasificación con las variables originales.....	100
Tabla 3.13: Clasificación con todas las variables originales estandarizadas.....	101
Tabla 3.14: Clasificación con todas las variables Z.....	101
Tabla 3.15: Determinación del porcentaje de varianza explicada.....	102
Tabla 3.16: Clasificación con las variables Z reducidas, las primeras 5.....	102
Tabla 3.17: Clasificación con las variables Z reducidas, las primeras 4.....	103
Tabla 3.18: Clasificación con las variables Z reducidas, las primeras 3.....	103
Tabla 3.19: Comparación de la clasificación.....	107
Tabla 4.1: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet en el nivel 0, (señal original, sin descomponer).....	109

Tabla 4.2: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 1.....	110
Tabla 4.3: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 1.....	111
Tabla 4.4: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 1.....	112
Tabla 4.5: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 1.....	113
Tabla 4.6: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 1.....	114
Tabla 4.7: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 1.....	115
Tabla 4.8: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 1.....	116
Tabla 4.9: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 1.....	117
Tabla 4.10: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 2.....	118
Tabla 4.11: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 2.....	119
Tabla 4.12: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 2.....	120
Tabla 4.13: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 2.....	121
Tabla 4.14: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 2.....	122
Tabla 4.15: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 2.....	123
Tabla 4.16: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 2.....	124

Tabla 4.17: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 2.....	124
Tabla 4.18: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 3.....	126
Tabla 4.19: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 3.....	127
Tabla 4.20: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 3.....	128
Tabla 4.21: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 3.....	129
Tabla 4.22: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 3.....	130
Tabla 4.23: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 3.....	131
Tabla 4.24: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 3.....	132
Tabla 4.25: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 3.....	133
Tabla 4.26: Comparación de la precisión de la clasificación tomando las señales originales (sin descomposición wavelet alguna).....	134
Tabla 4.27: Comparación de la precisión de la clasificación tomando el vector vwpt de la señal en el nivel 1 de descomposición para distintas familias wavelet.....	134
Tabla 4.28: Comparación de la precisión de la clasificación tomando el vector vwpt de la señal en el nivel 2 de descomposición para distintas familias wavelet.....	134
Tabla 4.29: Comparación de la precisión de la clasificación tomando el vector vwpt de la señal en el nivel 3 de descomposición para distintas familias wavelet.....	135

RESUMEN

PARÁMETROS CARACTERÍSTICOS DE SEÑALES DE VOZ UTILIZANDO WAVELET PACKET Y ACP PARA SU CLASIFICACIÓN MEDIANTE MÁQUINA DE VECTORES DE SOPORTE

Autor:

Ing. Franklin J. Yusti R.

Tutor:

Ing. MSc: Carlos Jiménez.

Bárbula, Julio 2015.

Analizar los sonidos emitidos por la voz humana es una tarea que ha mostrado interés por grupos de investigadores con distintos fines de estudio; se presenta en esta monografía una metodología que expone la reducción de dimensionalidad de un número de parámetros característicos de las señales de voz por medio del análisis de componentes principales (ACP) el cual es una técnica estadística de síntesis de información, al reducir el número de parámetros característicos mediante ACP se obtiene un conjunto de menor dimensión de variables donde cada variable es una combinación lineal de las variables originales y además estas nuevas variables son independientes entre sí. El objetivo fundamental de reducir el número de parámetros que caracterizan las señales de voz cobra importancia por el hecho de que estos nuevos parámetros servirán de entrenamiento para el diseño de una máquina de vectores de soporte que será capaz de clasificar entre señales de voces sanas y señales de voces con patologías. Se determinaron parámetros característicos de las señales de voz en el dominio de la frecuencia y en el dominio wavelet en el cual se utilizó la transformada wavelet packet. Se realizaron diferentes experimentos en los que los parámetros característicos de las señales de voz fueron determinados utilizando diferentes bases wavelet y distintos niveles de descomposición, lográndose conseguir una base wavelet y un nivel de descomposición en los que el conjunto de señales de voz aportado por el centro de análisis y tratamiento de señales (CATS) fueron clasificados con un 100% de precisión.

Palabras clave: Parámetros característicos de voz, transformada wavelet, componentes principales, máquina de vectores de soporte, SVM.

INTRODUCCIÓN

La señal de voz producida por los seres humanos con la intención principal de comunicarse, en algunos casos puede presentar patologías que dificultan su entendimiento las cuales se presentan en el aparato fonador de las personas; para la detección de la presencia de patologías en la señal de voz se han realizado estudios basados en procesamiento digital de señales para facilitar el diagnóstico sobre las anomalías presentes en el aparato fonador; debido a las características no estacionarias de la señal de voz se han hecho estudios de la misma basados en técnicas que usan la transformada de Fourier y la transformada wavelet por lo tanto siguiendo en este mismo ámbito de investigación se propuso utilizar la transformada wavelet packet ya que esta además de permitir analizar la señal en el dominio del tiempo y de la frecuencia simultáneamente permite realizar un proceso de filtrado sobre la señal de voz con mayor profundidad que las transformadas antes mencionadas.

Así mismo, para ayudar a los profesionales en esta área de estudio, se planteó utilizar la máquina de vectores de soporte (SVM) la cual fue la encargada de realizar la clasificación de señales entre patológicas y señal sana, debido a que esta máquina de soporte vectorial necesita un entrenamiento de aprendizaje, se necesitó una base de datos de señales de voz con patologías y señales de voz sanas con la clasificación de la misma conocida previamente.

Es por ello, como la máquina de vectores de soporte necesita de un entrenamiento previo para poder clasificar las señales y esta necesita de una cantidad de parámetros de entrada que caractericen la señal de voz se evidenció la necesidad de utilizar el análisis de componentes principales (ACP) el cual se encargó de seleccionar sólo aquellos parámetros característicos de la señal de voz que mejor describan a dicha señal y con esto se pretendió

reducir el número de parámetros de entrada que utilizará la máquina de vectores de soporte para realizar la posterior clasificación de señales desconocidas.

El trabajo se dividió en cinco capítulos. En el capítulo I se presenta tanto el planteamiento del problema así como los objetivos que se pretenden alcanzar y la justificación de dicho proyecto. El capítulo II presenta los antecedentes que fundamentan la realización de proyectos en esta misma área de investigación, también se presentan las bases teóricas y los términos básicos necesarios para comprender las ideas propuestas en esta investigación. El capítulo III se refiere al planteamiento de la metodología empleada, en él se describe el tipo, la modalidad y el diseño de la investigación así como las diferentes fases en la que se realizó este trabajo de investigación. El capítulo IV está referido a los resultados obtenidos en esta investigación finalmente en el capítulo V se presentan las conclusiones y recomendaciones propuestas tanto para este trabajo como para investigaciones futuras.

CAPITULO I

EL PROBLEMA

Planteamiento Del Problema

Las señales de voz son sonidos producidos por los seres humanos con la intención principal de comunicarse, estas señales son producidas por el aparato fonador, la presencia de patologías en el aparato fonador produce alteraciones en las señales de voz que ocasionan dificultad en la comunicación mediante el habla y esto conlleva a otra serie de problemas en la vida del individuo como baja en la autoestima, disminución en su calidad de vida al perder el interés en querer comunicarse, incluso puede incurrir en que el individuo tienda a apartarse de su entorno en el cual se desenvuelve.

En el estudio de las patologías presentes en las señales de voz, los investigadores en este campo han desarrollado técnicas mediante procesamiento digital de señales que han sido utilizadas para facilitar el diagnóstico de la presencia de patologías en las señales de voz, dichas técnicas se basan en la identificación de parámetros que puedan ser utilizados para caracterizar las señales, sin embargo, la cantidad de parámetros característicos que puedan extraerse de una señal puede ser numerosa y algunos aportaran mas información que otros por lo cual surge un problema de selección de parámetros característicos de la señal tal como lo establece Del Pino P., y otros. (2008:02). “Si se seleccionan estos parámetros convenientemente, se podrían utilizar para comparar voces provenientes de pacientes sanos y de pacientes con patologías de voz, para ayudar al especialista a diagnosticar en forma objetiva y

rápida el estado de la señal de voz”. Sin embargo esta selección conveniente de parámetros a la cual hace mención Del Pino no establece un número determinado en la cantidad de parámetros a considerar por lo cual podría quizás producirse que se estén calculando parámetros que ofrezcan información redundante, se evidencia aquí una interrogante respecto a si son necesarios todos los parámetros característicos que se extraigan de las señales de voz para ser comparadas y clasificadas entre normales o patológicas.

En lo que respecta a comparar voces provenientes de pacientes sanos y de pacientes con patologías es necesario crear una base de datos que contenga diferentes tipos de voces para extraer sus parámetros característicos, dicha base de datos se forma a partir de grabaciones realizadas mediante dispositivos electrónicos los cuales al grabar la señal de voz también graban cualquier otro sonido que se presente en el ambiente el cual se presenta como una fuente de ruido y además por ser de naturaleza electrónica también se añade ruido proveniente de la red de alimentación del sistema.

Con el propósito de realizar una reducción del ruido presente en la grabación de voz se emplean técnicas como las que usan la transformada wavelet ya que el análisis de las señales mediante la transformada wavelet es equivalente a un proceso de filtrado, donde se realiza una descomposición de la señal obteniéndose los coeficientes de aproximación asociados a las componentes de baja frecuencia y los coeficientes de detalle asociados a las componentes de alta frecuencia.

Así mismo, en el proceso de reducción de ruido en las señales de voz se han implementado técnicas que utilizan un valor de umbral para modificar los coeficientes de detalle provenientes de la descomposición wavelet. Por lo tanto los

problemas se centran en obtener una buena estimación de la señal de voz al reducirle el ruido, para ello hay que seleccionar un buen valor de umbral, un buen nivel de descomposición de la señal y una buena base wavelet, a partir de las estimaciones de la señal de voz se obtienen nuevos parámetros característicos que determinan la señal estudiada.

Por tales razones, es que se presenta en este trabajo de investigación como alternativa para lograr una mejora en la estimación de la señal de voz, utilizar la transformada wavelet empaquetada (wavelet packet) para descomponer la señal teniendo en cuenta que la wavelet packet ofrece una mejor descomposición de la señal debido a que esta descompone tanto los coeficientes de aproximación como los coeficientes de detalle; siguiendo en el mismo ámbito de mejorar la reducción del ruido presente en la señal de voz se propone utilizar una técnica para calcular el valor umbral como la propuesta por Vega, C. (2009) denominada umbral garrote y posteriormente obtener los diferentes parámetros que caractericen las señales de voz para que con el uso de la técnica de análisis de componentes principales reducir el número de parámetros que caractericen la voz y tener un número más pequeño de parámetros que permitan entrenar la máquina de vectores de soporte y proceder a clasificar señales desconocidas entre sanas o con patologías. Tomando en cuenta lo anteriormente expuesto se evidencia la necesidad y la ventaja de aplicar nuevas técnicas para determinar Parámetros Característicos de Señales de Voz Utilizando Wavelet Packet y ACP para su Clasificación Mediante Máquina de Vectores de Soporte.

Formulación del Problema

¿De qué manera se puede determinar un número reducido de parámetros característicos en las señales de voz para que puedan ser clasificadas por medio de una máquina de vectores de soporte entre normales y patológicas?

Objetivos de la Investigación

Objetivo General

Determinar parámetros característicos en las señales de voz utilizando transformada wavelet packet y análisis de componentes principales (ACP) para su clasificación mediante máquinas de vectores de soporte.

Objetivos Específicos

Recolectar señales voz para establecer una base de datos conformada por señales de voz normales y señales de voz con patologías.

Aplicar transformada wavelet packet a las señales de voz para obtener una nueva representación de las mismas en el dominio frecuencial.

Determinar parámetros característicos en las señales de voz tanto en el dominio del tiempo como en el dominio de la frecuencia para establecer indicadores representativos únicos para cada tipo de señal.

Utilizar el análisis de componentes principales (ACP) para reducir el número de parámetros característicos y minimizar el número de elementos de entrada a la máquina de vectores de soporte.

Realizar la clasificación de las señales mediante la máquina de vectores de soporte para diagnosticar si una señal de voz presenta patologías.

Justificación de la Investigación

Relevancia Tecnológica

Motivado en la búsqueda de mejoras en cuanto al diagnóstico de patologías presentes en las señales de voz, el siguiente trabajo de investigación propone un aporte en materia de estimación de señales mediante el uso de una novedosa herramienta como lo es la transformada wavelet empaquetada combinada con la técnica de contracción garrote para reducir el ruido presente en las señales de voz.

Así mismo, se pretende mejorar la rapidez del diagnóstico de la presencia de patologías al utilizar el análisis de componentes principales en la reducción del

número de parámetros característicos que serán utilizados para entrenar la máquina de vectores de soporte encargada de la clasificación de las señales.

Relevancia Social

El desarrollo de este trabajo podría aportar ventajas en el campo de la medicina como por ejemplo servir de apoyo al personal médico en entrenamiento para verificar o comparar sus diagnósticos con los resultados ofrecidos por esta herramienta y fortalecer su capacitación.

Relevancia Académica

Además, las técnicas presentadas en este trabajo de investigación pueden servir de provecho a los grupos de investigadores de la Facultad de Ingeniería de la Universidad de Carabobo que actualmente se encuentran desarrollando proyectos de investigación en análisis de voz para ser aplicados en robótica, inteligencia artificial y procesos de automatización.

Así mismo, este proyecto de investigación aportará una documentación bibliográfica que servirá como material de consulta en futuras investigaciones así como también nuevas herramientas metodológicas en el procesamiento digital de señales aplicado en el diagnóstico de patologías en señales de voz.

Relevancia Económica

Para la realización de este proyecto de investigación sólo se necesitan los recursos con los que cuenta actualmente la Facultad de Ingeniería de la Universidad de Carabobo, entre ellos, bibliotecas, laboratorios de computación, investigadores en el área y conexión a Internet, es decir, no se necesita de ningún tipo de financiamiento extra para gastos de realización del proyecto.

En cuanto al aporte en el área económica, por ser implementado en computadores personales es que puede ser utilizado como se mencionó anteriormente como una herramienta de entrenamiento de personal médico en capacitación con lo cual se puede prescindir del pago de personal de adiestramiento. Para la comunidad en general puede ser una herramienta que aporte un diagnóstico general de manera rápida en los casos en los cuales no se necesite de un examen médico riguroso dependiendo de las patologías presentadas.

CAPITULO II

MARCO TEÓRICO REFERENCIAL

En este capítulo se presentará siguiendo este orden, un resumen de las investigaciones previas relacionadas con este trabajo cuyas técnicas o teorías utilizadas serán tomadas en cuenta como antecedentes para esta investigación, seguidamente se presentarán las bases teóricas necesarias para entender y desarrollar las técnicas de procesamiento de señales de voz utilizadas en este trabajo de investigación y finalmente se detallará un resumen en cuanto a la definición de términos básicos involucrados para la buena comprensión del desarrollo teórico práctico de las técnicas aquí presentes.

Antecedentes

Jiménez C., Jiménez J. (2010). **Clasificación de señales de voz utilizando Transformada de Wavelet y Máquinas de Vectores de Soporte.** En este proyecto de investigación se diseñó un procedimiento destinado a la clasificación de señales de voz entre sanas y patológicas, dicho diseño consistió en procesar la señal de voz aplicándole la transformada wavelet para extraerle los parámetros característicos a la señal obtenida, luego, para realizar la clasificación de las señales de voz se aplicaron estos parámetros a una maquina de vectores de soporte. De este proyecto se toma como base la metodología aplicada en la extracción de los parámetros característicos de las señales de voz y la aplicación de las máquinas de vectores de soporte en el proceso de clasificación de señales.

Del Pino P., y otros. (2008). **Aplicación de la transformada de wavelet para el análisis de señales de voz normales y patológicas.** En este artículo se utilizó la transformada de wavelet db15 (Daubechies 15), con un nivel de descomposición de $J=10$, para realizar el proceso de filtrado de señales de voz y extraer parámetros que permitan determinar si la señal de voz pertenece a un paciente sano o con alguna patología. De este artículo se consideran los parámetros característicos obtenidos a partir del procesamiento de la señal de voz mediante la transformada wavelet.

Del Pino P. (2003). **Identificación de los parámetros espectrales que determinan la calidad de la voz.** En dicho trabajo de investigación se determinan parámetros que caracterizan las señales de voz obtenidos a partir del estudio de su espectro de frecuencias mediante el uso de la transformada de Fourier. De este trabajo se analizan el conjunto de parámetros identificados como calificadores de la calidad de las señales de voz así como también la metodología utilizada en la obtención de dichos parámetros.

Jiménez C. (2004). **Análisis de la relación señal a ruido de la voz, utilizando la transformada de Wavelet.** En esta investigación se utiliza la transformada de wavelet para filtrar las señales de voces tanto sanas como patológicas, para posteriormente obtener la relación señal a ruido. En el proceso se utilizan diferentes bases wavelet y distintos niveles de descomposición y finalmente se compara la relación señal a ruido para las dos clases de señales. De este proyecto se analizará la metodología de aplicación de la transformada de wavelet y la de obtención de la relación señal a ruido.

Vega C., (2009). **Un método automático para la elección de la base de ondícula y la selección del umbral para la estimación de señales.** La contribución de este trabajo es la creación y el desarrollo de un procedimiento novedoso para la estimación de señales, procedimiento LDM–G. El procedimiento consiste en la implementación de algoritmos para la estimación de señales mediante la transformada de ondícula, con el uso del principio de longitud de descripción mínima para la selección simultánea y automática de la base de ondícula y del valor del umbral. De este trabajo se considera el análisis de la aplicación de la técnica de contracción garrote en el proceso de estimación de señales.

Bases Teóricas

La Voz y el Habla

En términos poco complejos, la voz puede considerarse como el sonido que producen los seres humanos al hablar, se evidencia entonces una alta interrelación entre la voz y el habla, siendo el habla uno de los medios de comunicación natural que utiliza el hombre, sin esfuerzo alguno aparentemente. Esta capacidad de comunicarse mediante el habla diferencia al hombre de otros seres vivos y se considera una manifestación de inteligencia.

Así mismo, la voz es uno de los factores claves en la comunicación mediante el habla, la voz debe ser clara, debe producir sonidos que sean reconocibles por el receptor; una señal de voz patológica puede entonces dificultar el proceso de comunicación entre el emisor de la voz y el receptor de la misma.

Aparato Fonador Humano

El aparato fonador es el conjunto de órganos que tienen como función producir la voz humana, lo conforman los pulmones, los cuales producen un flujo de aire; la laringe, que contiene las cuerdas vocales, la faringe, las cavidades oral y nasal y una serie de elementos articulatorios como los labios, los dientes, el alvéolo, el paladar, el velo del paladar y la lengua tal como se muestra en la figura 2.1.

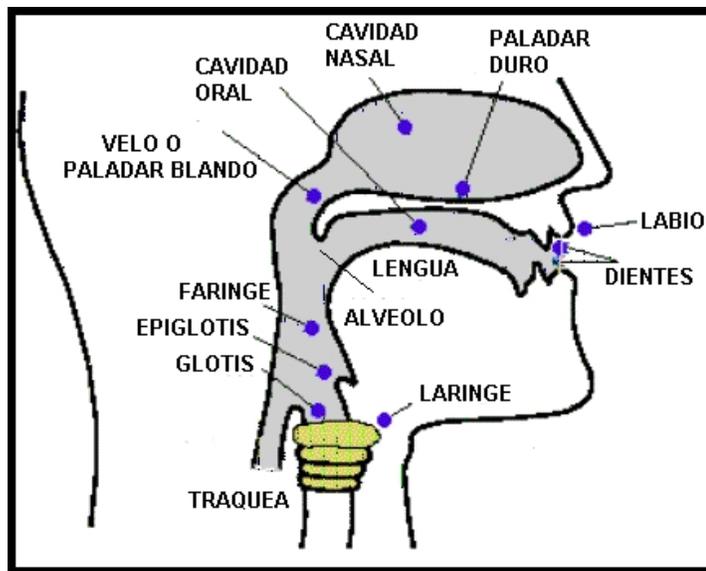


Figura 2.1: Aparato fonador humano.

Fuente: http://es.wikipedia.org/wiki/Aparato_fonador (2014).

En el proceso de generación de la voz, el sonido inicial proviene de la vibración de las cuerdas vocales conocida como vibración glotal, es decir, el efecto sonoro se genera por la rápida apertura y cierre de las cuerdas vocales conjuntamente con el flujo de aire emitido desde los pulmones. Las cuerdas vocales son dos membranas ubicadas dentro de la laringe, la abertura entre ambas cuerdas se denomina glotis. Cuando la glotis comienza a cerrarse, el aire proveniente desde los pulmones experimenta una turbulencia, emitiéndose un ruido de origen aerodinámico.

Modelo simplificado de las cavidades oral, labial y nasal

Al cerrarse más las cuerdas vocales comienzan a vibrar a modo de lengüetas, produciéndose un sonido tonal, es decir periódico y cuya frecuencia varía en forma inversa al tamaño de las cuerdas. Este sonido es propio del hablante y es más agudo para el caso de mujeres y niños. Carece de información lingüística.

Luego de atravesar la glotis el sonido pasa a través de la cavidad supraglótica, que es la porción del aparato fonador que permite modificar el sonido dentro de márgenes muy amplios. Está conformado principalmente por tres cavidades, la cavidad oral, la cavidad labial y la cavidad nasal, correspondientes a la garganta, los labios y la nariz respectivamente. Estas cavidades constituyen resonadores acústicos, los cuales modifican los sonidos de acuerdo a la forma que adopten, la lengua y los labios permiten efectuar esta variación de manera voluntaria.

Forma de Onda de la Señal de Voz

La señal de voz está constituida por un conjunto de sonidos generados por el aparato fonador. Esta señal acústica puede ser transformada por un micrófono en una señal eléctrica. La señal de voz en el tiempo puede ser representada en un par de ejes cartesianos.

Como todos los sonidos, está formado esencialmente por curvas elementales (senos y cosenos) pero las posibles combinaciones de éstas pueden ser complejas. A manera de ejemplo, se muestra en la figura 2.2 la forma de onda de la palabra

“explorador”. La representación de la señal de voz en función del tiempo es importante puesto que brinda información sobre características importantes como la energía y los cruces por cero, las cuales facilitan su estudio y análisis.

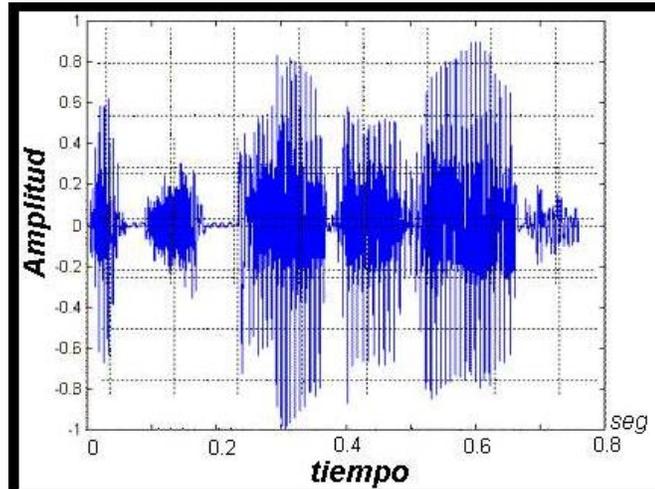


Figura 2.2: Forma de onda de la palabra “Explorador”.
Fuente: Peralta A., et all. (2002).

Digitalización de la Señal de Voz

Según lo expuesto por Capoccioni D. (2006:46). “La señal producida por el habla es analógica ya que el flujo de ondas de presión de aire, que al vibrar a determinadas frecuencias se hacen sonoras, es continuo y variado en el tiempo”.

Es por ello que Childers (1999) citado por Capoccioni D. (2006) establece que:

Para registrar el habla en cualquier aparato que pueda almacenar sonidos es necesario utilizar una interface entre el locutor y el aparato, esta interface es el micrófono el cual está constituido por un circuito eléctrico. Dependiendo del

principio de funcionamiento y por lo tanto del elemento variable sensible a la presión del aire, se tienen micrófonos de carbón que usan como resistencia variable conformada por gránulos de carbón, micrófonos de bobina móvil que usan una inductancia variable y micrófonos de condensador variable que tienen capacitancia variable incluida en su circuito eléctrico. Cuando en la entrada del micrófono se detecta, por medio de su elemento variable, los cambios de presión que se producen en el medio, cambia la impedancia del circuito que lo conforma y por lo tanto varía el voltaje en su salida produciendo voltajes en una cantidad de valores continuos en el tiempo llamada señal eléctrica analógica.

A su vez el micrófono está conectado a un computador, el cual está equipado con hardware y software adecuados para convertir las señales analógicas que llegan desde el micrófono en señal digital, eso es, detecta las amplitudes en tiempo discreto (señal con valores discretos) registrando un número determinado de veces por segundo (frecuencia, Hz) esos valores de la amplitud de voltaje y almacenándolos como un vector o arreglo unidimensional. (p: 49).

De esa manera la señal analógica obtenida se convierte en señal de formato digital y puede ser procesada mediante el muestreo y la cuantificación de la misma. Cada valor de presión muestreada, durante el habla, es redondeado o cuantificado al valor más cercano que se pueda expresar en un número dado de bits.

Características Fundamentales de la Señal de Voz

Energía y Cruces por Cero

La variación de energía en la señal de voz se debe a la variación de la presión subglotal y de la forma del tracto vocal. La Energía es útil para distinguir segmentos

sordos y sonoros en la señal de voz, debido a que los valores de esta característica aumentan en los sonidos sonoros respecto a los sordos.

Para el caso de las señales discretas donde N es el número de muestras de la señal, la energía se define por (ecuación 2.1):

$$E = \sum_{m=0}^{N-1} x(m)^2 \quad (2.1)$$

En la figura 2.2 se observa que los cruces por cero indican el número de veces que una señal continua toma el valor de cero. Para las señales discretas, un cruce por cero ocurre cuando dos muestras consecutivas difieren de signo, o bien una muestra toma el valor de cero.

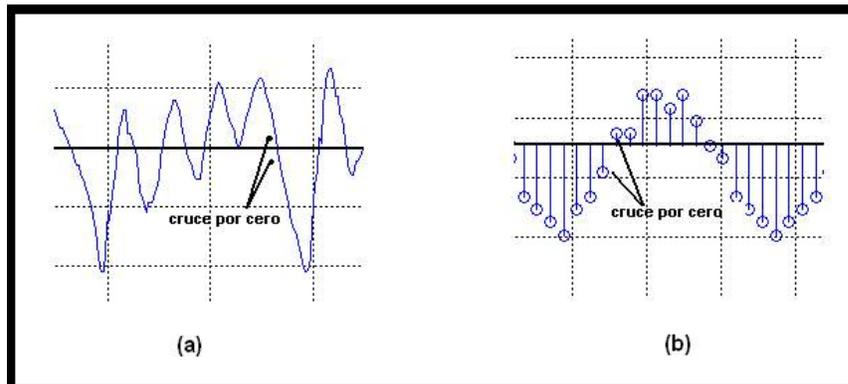


Figura 2.3: Cruces por Cero. a) Señal Continua. b) Señal Discreta.
Fuente: Peralta A, et all. (2002).

Consecuentemente, las señales con mayor frecuencia presentan un mayor valor de esta característica, el ruido también genera un gran número de cruces por cero.

La formulación matemática de la densidad de cruces por cero para señales discretas (ecuación 2.2), en la cual, sgn es la función signo y N es el número de muestras de la señal.

$$z = \sum_{m=0}^{N-1} |sgn[x(m)] - sgn[x(m - 1)]| \quad (2.2)$$

Se muestran las gráficas de energía y cruces por cero de la palabra 'seis' en la figura 2.4. Como se puede observar, el valor de la energía varía en relación directa con la amplitud de la señal. La función de densidad de cruces por cero alcanza sus valores más altos cuando se trata de sonidos tales como la 's', que son conocidos como sonidos fricativos.

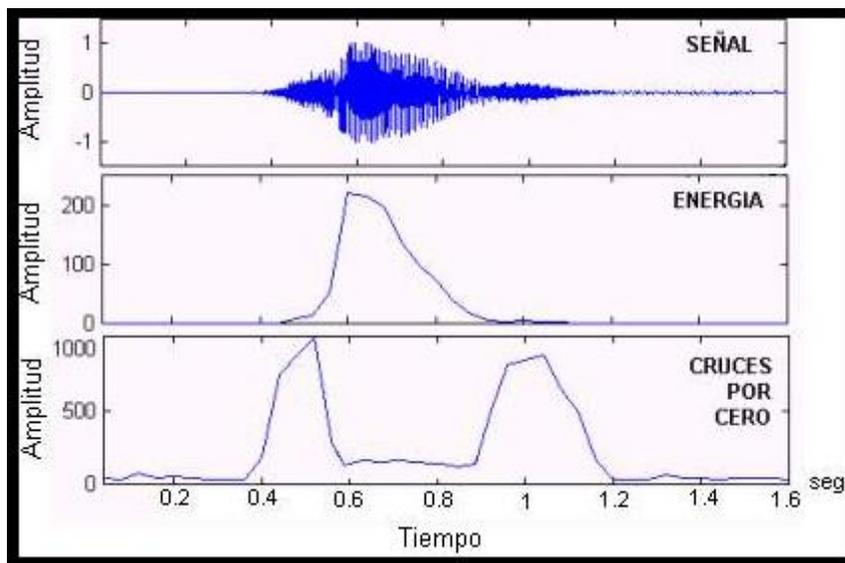


Figura 2.4: Energía y cruces por cero de la palabra “seis”.
Fuente: Peralta A, et all. (2002).

El estudio de la señal de voz se realiza en el dominio de la frecuencia, con la finalidad de conocer sus características espectrales. En la figura 2.5 se muestra el espectro de una señal de voz correspondiente a la palabra 'Dos'.

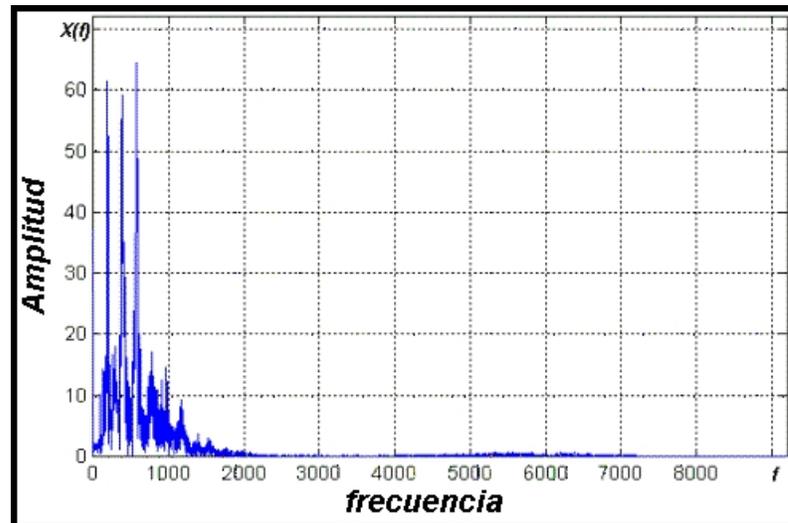


Figura 2.5: Espectro de frecuencia de la palabra 'dos'.
Fuentes: Flores A. (1993).

La frecuencia fundamental o también denominada pitch, brinda información sobre la velocidad a la que vibran las cuerdas vocales al producir un sonido, el cual es generado por la rápida apertura y cierre de las cuerdas vocales con pequeños soplos de aire, produciendo un espectro de frecuencia similar al mostrado en la figura 2.5. Este espectro podría ser obtenido si se colocara un micrófono de amplio rango directamente en la garganta, encima de las cuerdas vocales, pero debajo de las estructuras resonantes del tracto vocal.

Según lo expuesto por Flores A. (1993:71). “El espectro está conformado de armónicos de periodo pitch, el cual es el rango fundamental de frecuencia producidas por las cuerdas vocales”.

Si bien el espectro lleva un gran componente cerca de la frecuencia pitch (aproximadamente 50 Hz), tiene gran cantidad de armónicos, y así tiene componentes de frecuencia que se extiende hasta pasado los 5 KHz.

Otra característica importante es la envolvente espectral, un ejemplo de ella se muestra en la figura 2.6. Un análisis adecuado sobre esta característica permite obtener información sobre los diferentes tipos de sonido.

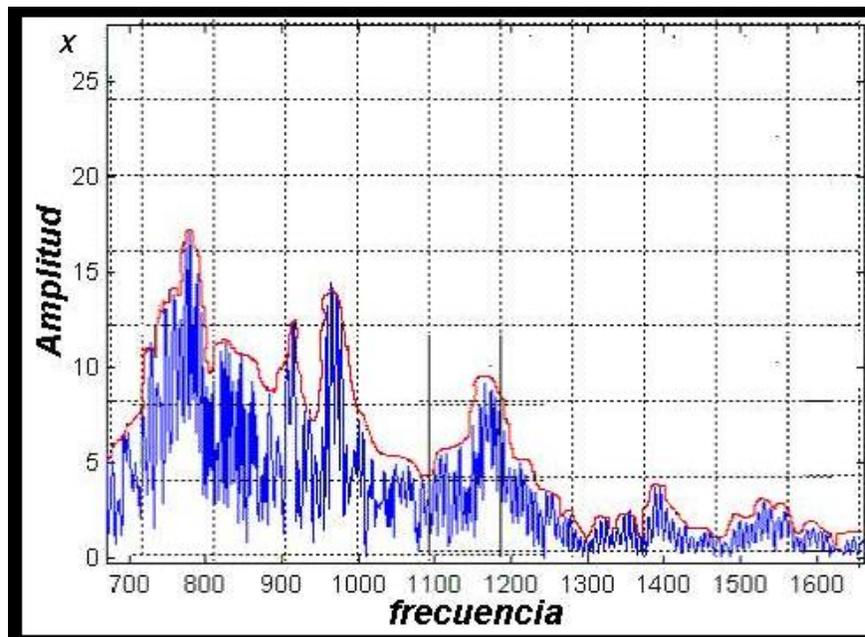


Figura 2.6: Envolvente espectral.
Fuente: Flores A. (1993).

Frecuencias Formantes

Las cavidades que conforman la cavidad supraglótica actúan como resonadores acústicos. Si se realiza un análisis espectral del sonido luego de haber atravesado estas cavidades, el efecto de la resonancia produciría un énfasis en determinadas

frecuencias del espectro obtenido, a las que se les denominara 'formantes'. Existen tantas formantes como resonadores posee el tracto vocal. Sin embargo se considera que sólo las tres primeras permiten diferenciar los distintos tipos de sonido. En la figura 2.7 se muestra el espectro de la palabra 'uno', y se denominan F1, F2 y F3 a sus tres principales frecuencias formantes. La amplificación de cada una de estas tres frecuencias depende del tamaño y forma que adopta la cavidad bucal y la cavidad oral, y si el aire pasa o no por la nariz.

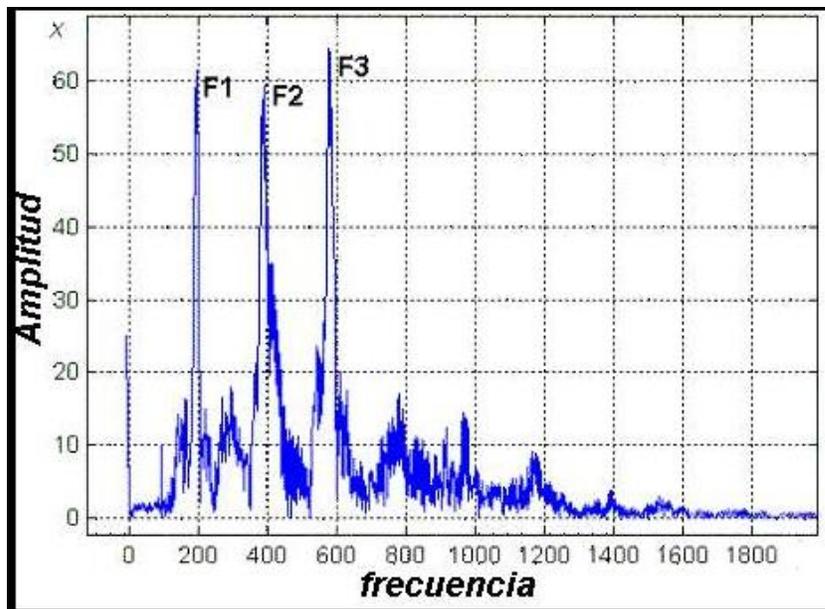


Figura 2.7: Frecuencias formantes.
Fuente: Flores A. (1993).

Tipos de Señales de Voz

De acuerdo a lo expuesto por Flores A. (1993:89). “Básicamente, la Señal de Voz puede clasificarse en los siguientes tipos, Sonora, No Sonora y Plosiva”.

Señal Sonora

La señal sonora se genera por la vibración de las cuerdas vocales manteniendo la glotis abierta, lo que permite que el aire fluya a través de ella. Estas señales se caracterizan por tener alta Energía y un contenido frecuencial en el rango de los 300Hz a 4000Hz presentando cierta periodicidad, es decir son de naturaleza cuasiperiódica. El tracto vocal actúa como una cavidad resonante reforzando la energía en torno a determinadas frecuencias (formantes). En la figura 2.8 se muestra el comportamiento de este tipo de señales en el tiempo. Todas las vocales se caracterizan por ser sonoras pero existen consonantes que también lo son, tales como, la 'b', 'd' y la 'm', entre otras.

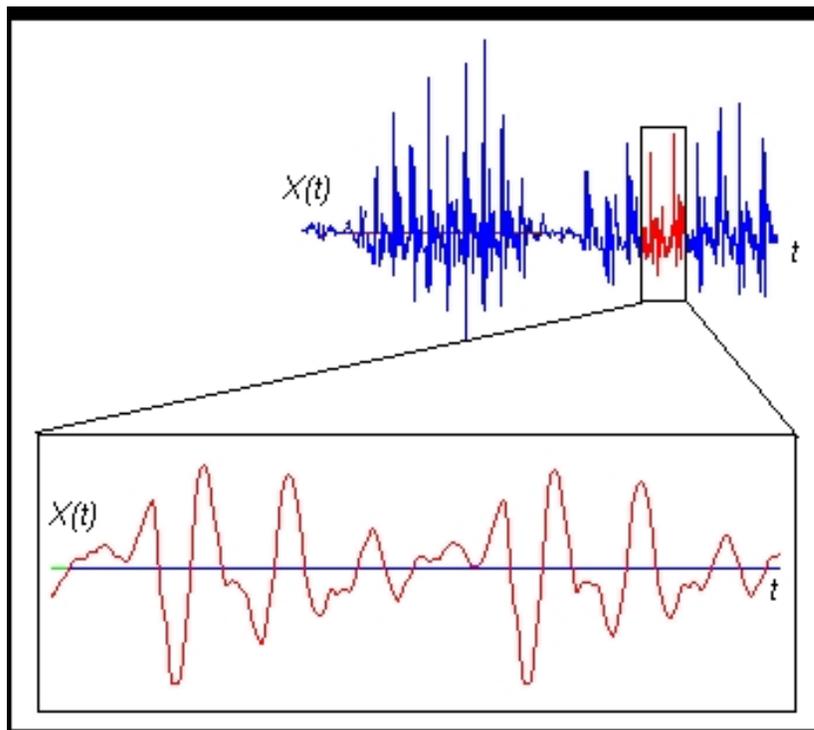


Figura 2.8: Señal sonora.
Fuente: www.esacademic.com (2014).

Señal no Sonora

A esta señal también se le conoce como señal fricativa o sorda, y se caracteriza por tener un comportamiento aleatorio en forma de ruido blanco. Tienen una alta densidad de Cruces por Cero y baja Energía comparadas con las señales de tipo sonora. Durante su producción no se genera vibración de las cuerdas vocales, ya que, el aire atraviesa un estrechamiento, y genera una turbulencia. Las consonantes que producen este tipo sonidos son la 's', la 'f' y la 'z' entre otras. La figura 2.9 muestra la forma de onda de una señal no sonora.

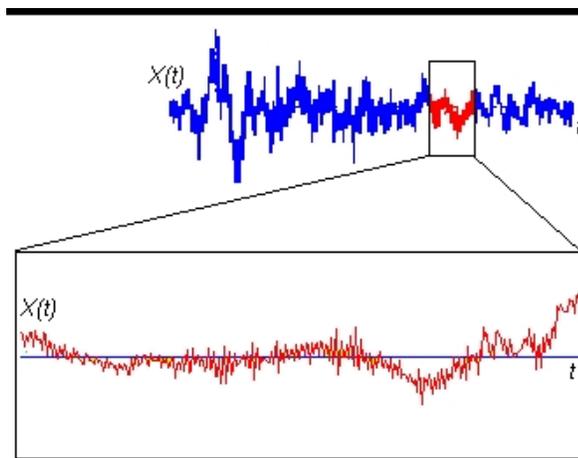


Figura 2.9: Señal no sonora.

Fuente: www.esacademic.com (2014).

Señal Plosiva

Esta señal se genera cuando el tracto vocal se cierra en algún punto, lo que causa que el aire se acumule para después salir expulsado repentinamente (explosión). Se caracterizan por que la expulsión de aire está precedida de un silencio. Estos sonidos se generan por ejemplo, cuando se pronuncia la palabra “campo”. La “p” es una

consonante de carácter plosivo, y existe un silencio entre las sílabas “cam” y “po”. Otras consonantes que presentan esta característica son “t”, y “k”, entre otras. La figura 2.10 muestra el comportamiento de este tipo de señal.

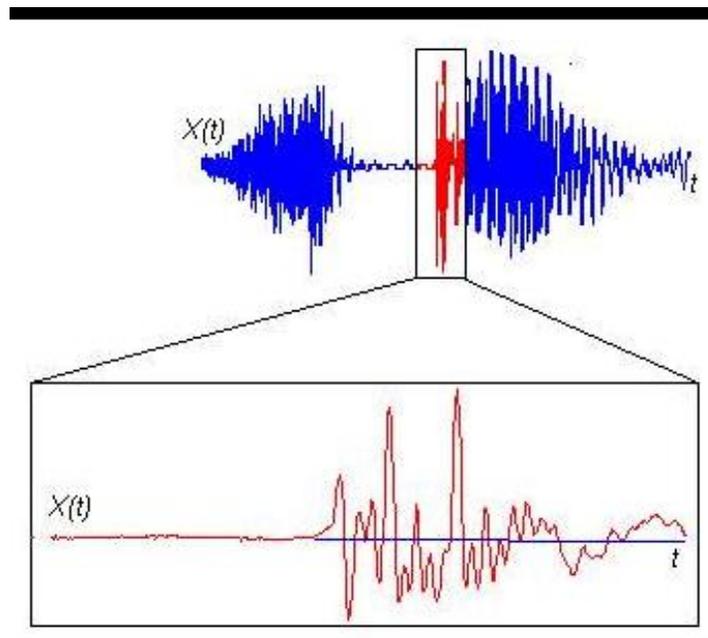


Figura 2.10: Señal sonora plosiva.
Fuente: www.esacademic.com (2014).

Factores que Afectan la Señal de Voz

Existen muchos factores que afectan la correcta percepción de las Señales de Voz, tales como el ruido, la acústica y la calidad del micrófono. El ruido, se define como aquellos sonidos aleatorios que de forma "oculta" transforman y enmascaran el sonido. Dado que, es poco probable encontrar un entorno de audio digital en perfecto silencio, es importante conocer la cantidad de ruido, en relación con la señal que se introduce en el equipo de sonido, especialmente en la tarjeta de sonido. La fuerza de cualquier sonido (hablar por ejemplo), comparada con la fuerza promedio del ruido,

se conoce como relación señal a ruido (**SNR** por sus siglas en inglés). A medida que aumenta la relación SNR, es mejor el trabajo realizado en grabación.

Para registrar el habla en cualquier aparato que pueda almacenar sonidos es necesario utilizar una interface entre el locutor y el aparato, esta interface es el micrófono el cual está constituido por un circuito eléctrico. Dependiendo del

De acuerdo con Cater, J (1984), la acústica de la habitación (ecos), puede crear cambios en el espectro de la Señal de Voz, debido a las resonancias de la habitación. Puesto que, cualquier ambiente cerrado tendría resonancias inherentes, su énfasis cuando interfiere con una señal de habla puede crear rangos anormales de frecuencias. Debido a esto, se producen dos cambios básicos en la acústica de una habitación, el primero es causado por el retardo en el tiempo del retorno de la señal original de una superficie reflectante, tal como una pared o una ventana. Cuando la onda es reflejada, regresa con mucha menor amplitud, y retardada en el tiempo, ésta interactúa con la forma de onda originalmente hablada para crear un nuevo espectro compuesto del habla. El segundo, está relacionado con la reflexión de una superficie rugosa de una pared, lo cual tiende a atenuar en altas frecuencias, pero a reforzar en el rango de bajas frecuencias.

El ruido del ambiente afecta si el usuario del sistema está operando el dispositivo en cualquier lugar que no sea una habitación tranquila, existe la posibilidad de la interferencia del ruido con las formas de onda. No obstante sin ruido externo, el sistema es susceptible de captar ruido a través del micrófono, y aunque suene extraño, muchas veces el ruido proviene desde la boca durante la pronunciación del mensaje.

En el caso de los sonidos plosivos, si el micrófono es ubicado directamente enfrente de la boca del hablante, entonces es muy susceptible de ser bombardeado por pequeñas ráfagas de aire ocasionadas por los sonidos plosivos. La mejor forma de tratar el problema es de rodear el micrófono con un material esponjoso transparente acústico, que rápidamente disipe la velocidad del viento de las pronunciaciones plosivas, permitiendo a las vibraciones acústicas normales pasar a través del micrófono. Otras fuentes de ruido externo, tal como los ventiladores en las computadoras, aire acondicionado, teléfonos, y otras personas hablando puede también causar problemas con la exactitud del sistema de reconocimiento. Otra técnica para cancelar el ruido externo es filtrar la señal de audio antes procesarla. Debido a que las frecuencias de voz que contienen información relevante están dentro de un rango relativamente estrecho desde 200Hz a 3000Hz, el espectro de audio puede ser filtrado a través de un filtro pasabanda para rechazar las señales acústicas fuera de ese rango de frecuencias. La calidad del micrófono probablemente es, el factor que más influye en la adquisición electrónica de señales del habla es el tipo de micrófono que se está usando. Existen, principalmente, cuatro tipos de micrófonos disponibles en el mercado, los cuales son el electreto, el dinámico, el de cristal y el de carbón. Para percibir fácilmente las diferencias entre estos tipos de micrófonos, sus características principales son comparadas en la tabla 1.1.

Tabla 1.1: Comparación de los tipos de micrófonos.

Parámetro	Tipo de micrófono			
	Electreto	Dinámico	Cristal	Carbón
Respuesta en Frecuencia	Excelente	Excelente	Bien	Regular
Distorsión	Muy bajo	Muy bajo	Bajo	Alto
Cancelación de ruido	Excelente	Bien	Regular	Regular
Tamaño	Pequeño	Medio	Grande	Grande
Peso	Bajo	Medio	Bajo	Medio
Costo	Alto	Alto	Medio	Bajo
Nivel de Salida	Bajo (voltaje)	Medio (voltaje)	Alto (voltaje)	Alto (Resistencia)
Impedancia	Alto	Bajo	Alto	Bajo

Fuente: Cater J (1984).

Transformada Wavelet

Faundez P., et all (2002:78) establecen que, si se tiene una función continua $f(t)$ en el tiempo la cual cumple ciertas condiciones, entonces, la transformada wavelet de $f(t)$ no es mas que expresar a $f(t)$ mediante una expansión de términos o coeficientes proporcionales al producto interno entre la función y diferentes versiones escaladas y trasladadas de una función prototipo $\psi(t)$ (wavelet madre). Por lo tanto se puede representar matemáticamente la transformada continua wavelet ($CWT(a,b)$) de una función $f(t)$ mediante:

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (2.3)$$

La variable a controla el ancho o soporte efectivo de la función $\psi(t)$ y la variable b provee la ubicación en el dominio del tiempo de la función $\psi(t)$. Algunos autores denominan a las variables a y b como variables de escala y traslación respectivamente.

Una de las propiedades de la función $\psi(t)$ es que cumple con la condición de admisibilidad, la cual indica que:

$$\Psi(0) = 0 \quad (2.4)$$

En esta ultima ecuación $\Psi = \Psi(\omega)$ es la transformada de Fourier de $\psi(t)$. El cumplimiento de esta condición implica que el valor promedio de $\psi(t) = 0$, lo que significa que $\psi(t)$ debe tener valores tanto positivos como negativos, es decir, $\psi(t)$ es una onda y está definida en un intervalo de tiempo finito dado por a alrededor de

un tiempo $t = b$, estas características son la que le dan a $\psi(t)$ la identidad de wavelet u ondícula.

Faundez, P., et all (2002:84) establecen que, una función wavelet madre $\psi(t)$ trae asociada consigo una función escala $\phi(t)$, con este par de funciones se puede aproximar cualquier función $f(t) \in L^2(R)$ mediante una de las funciones o mediante ambas. Así:

$$f(t) = \sum_k \sum_j c_{j,k} \phi(t) + \sum_k \sum_j d_{j,k} \psi(t) \quad (2.5)$$

$L^2(R)$ denota el espacio vectorial formado por el conjunto de *funciones cuadrado integrables*, es decir aquellas funciones que cumplen $\int_{-\infty}^{\infty} [f(t)]^2 dt < \infty$ también llamadas *señales de energía*.

En esta ultima ecuación los términos $c_{j,k}$ y $d_{j,k}$ (*coeficientes escala o de aproximación y coeficientes wavelet o de detalle*) se obtienen mediante el producto interno definido en las siguientes ecuaciones:

$$c_{j,k} = \frac{1}{\sqrt{j}} \int_{-\infty}^{\infty} f(t) \phi\left(\frac{t-k}{j}\right) dt \quad (2.6a)$$

$$d_{j,k} = \frac{1}{\sqrt{j}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-k}{j}\right) dt \quad (2.6b)$$

Función Escala

Es una función $\phi(t) \in L^2(\mathbb{R})$ que trasladada y escalada genera un conjunto de funciones $\{\phi_{j,k}(t) \mid j, k \in \mathbb{Z}\}$ definida como:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^{\frac{j}{2}} t - k) ; \quad j, k \in \mathbb{Z} \quad (2.7)$$

Ahora bien, se puede definir el siguiente subespacio vectorial dentro del conjunto $L^2(\mathbb{R})$:

$$V_j = L\{\phi_{j,k \in \mathbb{Z}}(t)\} \subseteq L^2(\mathbb{R})$$

Es decir, el conjunto $\{\phi_{j,k \in \mathbb{Z}}(t)\}$ es una base para el espacio vectorial V_j .

Entonces se dirá que una función $f(t)$ estará en V_j (esto es $f(t) \in V_j$) si puede expresarse como combinación lineal de los términos de la base del subespacio V_j .

$$f(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k}(t) \quad (2.8)$$

Para hallar los términos $c_{j,k}$ se realiza el producto interno de la ecuación (3.8) con la función con la función escala, obteniéndose la ecuación (3.9).

$$c_{j,k} = f(t) \cdot \phi(t) = \int_{-\infty}^{\infty} f(t) \phi(t) dt \quad (2.9)$$

Una función $\phi(t) \in L^2(\mathbb{R})$ se considera una buena función escala si cumple con las siguientes características:

- ✓ Para cada $j \in Z$ el conjunto $\{\phi_{j,k \in Z}(t)\}$ forma una base ortonormal para el subespacio $V_j \subseteq L^2(R)$.
- ✓ Los subespacios V_j están anidados, es decir, $\forall j \in Z, \{V_j \subset V_{j+1}\}$
- ✓ La función $\phi(t) \in L^2(R)$ tiene soporte compacto, $\exists \{x \in Z \mid \phi(x) \neq 0\}$

Función Wavelet

De la misma forma como se definió el subespacio V_j se puede definir otro subespacio llamado W_j con otro conjunto de funciones base:

$$W_j = L\{\psi_{j,k \in Z}(t)\} \quad (2.10)$$

Donde, $\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^{\frac{j}{2}} t - k) ; j, k \in Z$ (2.11)

Y cualquier función que esté en W_j se puede escribir:

$$f(t) = \sum_{k \in Z} d_{j,k} \psi_{j,k}(t) \quad (2.12)$$

Al conjunto W_j se le denomina el complemento ortogonal de V_j en V_{j+1} , es decir, todos los miembros de V_j son ortogonales a todos los miembros de W_j .

Matemáticamente se puede expresar mediante un producto interno que:

$$\langle \phi_{j,k}(t) | \psi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} \phi_{j,k}(t) \psi_{j,k}(t) dt = 0 \text{ para toda } j, k \in \mathbb{Z} \quad (2.13)$$

Se puede generalizar que:

$$V_{j+1} = V_j \oplus W_j \quad (2.14)$$

Transformada Discreta Wavelet (DWT)

Para este nuevo caso de estudio ahora se tendrá una discretización de la función $f(t)$ y de las funciones $\phi(t)$ y $\psi(t)$.

Ahora, si se tienen los siguientes espacios V_j representados en la figura 2.11.

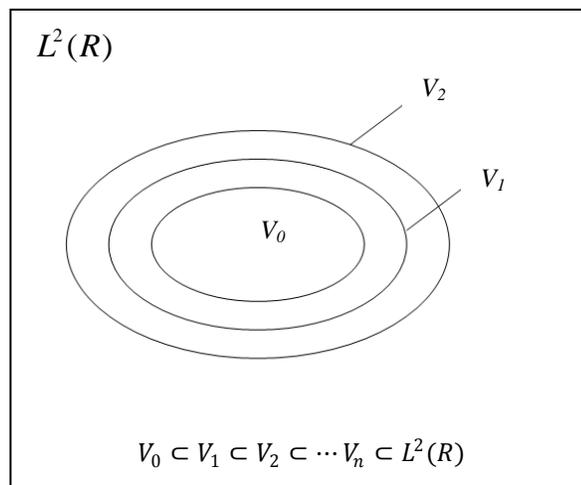


Figura 2.11: Representación de los subespacios V_j , se muestra que están anidados.
Fuente: Yusti F. (2010).

En la figura 2.11 se puede observar que el subespacio V_j cuya base es $\{\phi_{1,k \in \mathbb{Z}}(t)\}$ puede expandirse si se le completa la base hasta obtener una base para el subespacio V_2 . Este conjunto base sería de la forma $\{\phi_{2,k \in \mathbb{Z}}(t)\}$

Otra propiedad que presenta la función escala $\phi(t)$ es:

$$f(t) \in V_j \therefore f(2t) \in V_{j+1}$$

$$f(t) \in V_j \Leftrightarrow f(t) = \sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k}(t)$$

$$f(2t) = \sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k}(2t)$$

$$f(2t) = \sum_{k \in \mathbb{Z}} c_{j,k} 2^{\frac{j}{2}} \phi_{j,k}(2^{j+1}t - k)$$

$$f(2t) = \sum_{k \in \mathbb{Z}} \frac{c_{j,k}}{\sqrt{2}} 2^{\frac{j+1}{2}} \phi_{j,k}(2^{j+1}t - k)$$

$$f(2t) = \sum_{k \in \mathbb{Z}} \frac{c_{j,k}}{\sqrt{2}} 2^{\frac{j+1}{2}} \phi_{j+1,k}(t)$$

$$\therefore f(2t) \in V_{j+1}$$

Ahora se deduce que si una función $f(t)$ está en V_2 , entonces puede representarse como combinación lineal de los vectores de la base del subespacio V_2 .

$$f(t) \in V_2 \therefore f(t) = \sum_{k \in \mathbb{Z}} c_{j,k} 2^{\frac{j}{2}} \phi_{j,k}(2^{j+1}t - k) \mid j = 2 \quad (2.15)$$

Pero como la base del subespacio V_1 se puede completar mediante su complemento ortogonal W_1 se obtiene entonces una base para el subespacio V_2 .

Por lo tanto de la figura 2.12 se puede intuir que una función que esté en V_{j_0} se puede representar por la ecuación (2.16) la cual no es más que la ecuación (2.5) con una correcta notación para las sumatorias que dependen de los términos j y k :

$$f(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{N-1} \sum_{k=0}^{2^j-1} d_{j,k} \psi_{j,k}(t) \quad j, k \in \mathbb{Z}^+ \quad (2.16)$$

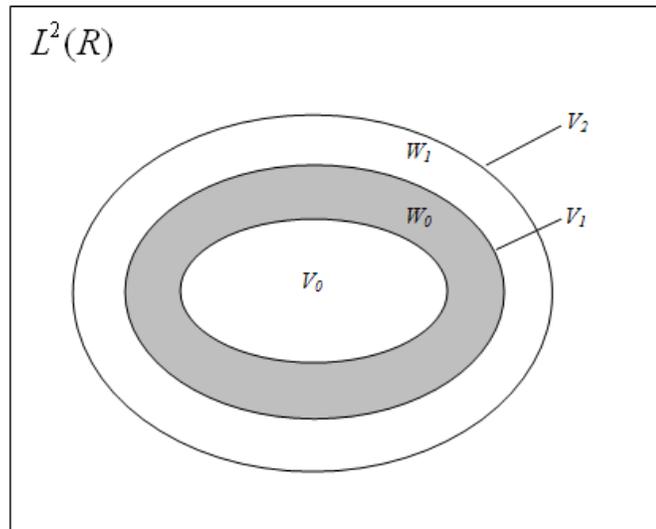


Figura 2.12: Representa los subespacios W_j y se observa que la intersección entre ellos es vacía y que $V_2 = V_1 \oplus W_1$

Fuente: Yusti F. (2010).

Quedando $f(t)$ aproximada por medio de la combinación lineal de funciones escala (*aproximaciones*) más funciones wavelet (*detalles*).

A cada uno de los niveles j en V_j y W_j se les denomina niveles de descomposición, así en la ecuación (2.16) se dice que $f(t)$ está en el subespacio V_{j_0} y por lo tanto se tienen j_0 niveles descomposición para $f(t)$.

Entonces para cada subespacio V_j se puede establecer:

$$V_2 = V_1 \oplus W_1$$

$$V_3 = V_2 \oplus W_2 = V_1 \oplus W_1 \oplus W_2$$

En general una función $f(t)$ (la cual es un vector) se puede aproximar mediante los vectores de la base de un subespacio generado por funciones escala más la sumatoria de los vectores de la base de subespacios ortogonales generados por funciones wavelet.

En términos prácticos, como se estableció en la definición de transformada continua wavelet, una función wavelet ventaniza a una función $f(t)$ (ver figura 2.13) y como la función wavelet tiene soporte compacto al hacer las múltiples traslaciones de la función wavelet y realizar el producto interno se obtienen los diferentes coeficientes wavelet:

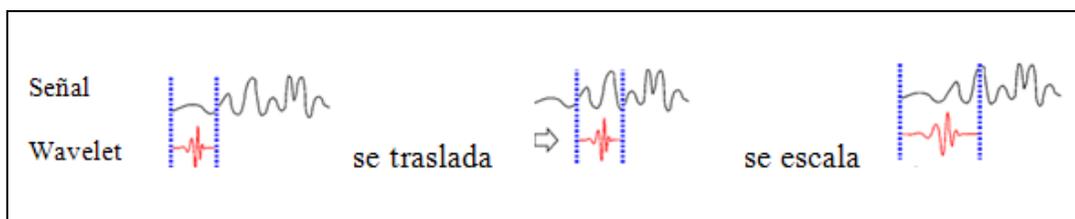


Figura 2.13: Se muestra como una función wavelet ventaniza a una señal.
Fuente: Yusti F. (2010).

La figura 2.13 indica como una wavelet ventaniza a una señal $f(t)$. El resultado del solapamiento entre estas dos señales da origen a los coeficientes wavelet de $f(t)$.

La figura 2.14 muestra un ejemplo de función wavelet y función escala (Daubechies8) y su representación como filtros pasaalto y pasabajo respectivamente.

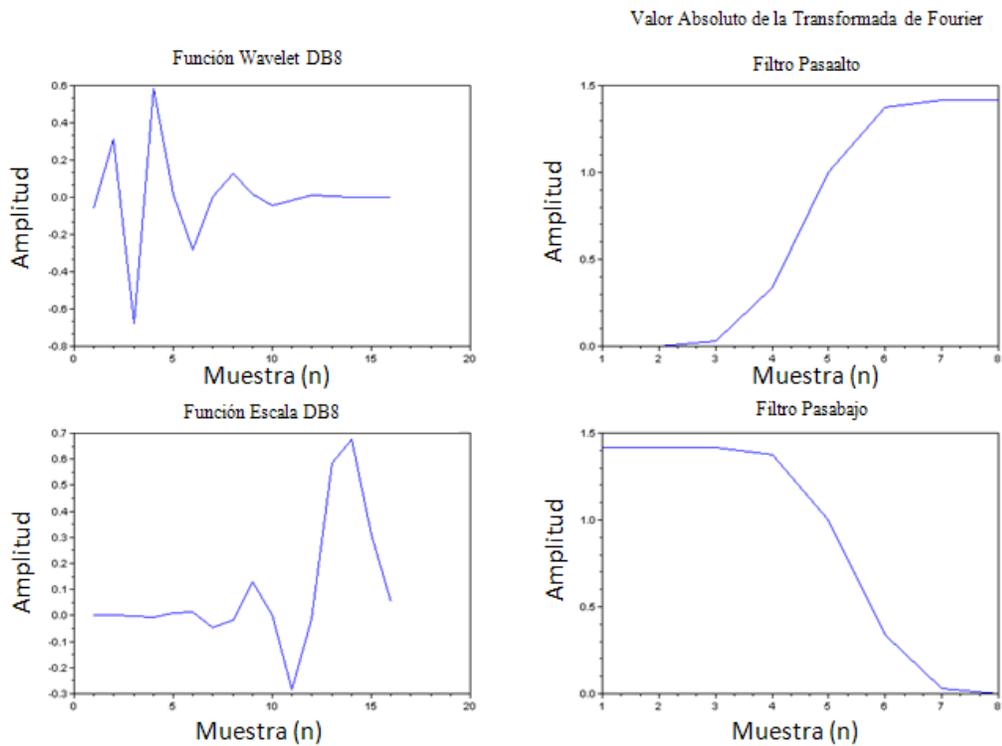


Figura 2.14: Representa la grafica del sistema Dubechie8. Función wavelet y su representación como filtro pasaalto, gráficos superiores. Función escala y su representación como filtro pasabajo, gráficos inferiores.

Fuente: Yusti F. (2010).

En tiempo discreto una función $f(t)$ se representa por medio de una secuencia o por medio de un vector cuyas entradas son las imágenes para distintos instantes de tiempo puntuales.

Por lo tanto el producto interno definido en (2.6a) y (2.6b) se puede definir en tiempo discreto como:

$$c_{[k]} = \sum_{j=1}^{j=k} f_{[j]} \phi_{[k+1-j]} \quad (2.17a)$$

$$d_{[k]} = \sum_{j=1}^{j=k} f_{[j]} \psi_{[k+1-j]} \quad (2.17b)$$

Las ecuaciones (2.17a) y (2.17b) definen el producto interno de dos funciones discretizadas, también son conocidas como la **convolución** entre dos secuencias. De forma más general:

Sean U y V vectores de longitud m y n respectivamente. Entonces W es un vector de longitud $m + n - 1$, cuyos k -ésimos elementos vienen dados por:

$$W_{[k]} = \sum_{j=1}^{j=k} U_{[j]} V_{[k+1-j]} \quad (2.18)$$

$$W_{[1]} = U_{[1]} V_{[1]}$$

$$W_{[2]} = U_{[1]} V_{[2]} + U_{[2]} V_{[1]}$$

$$W_{[3]} = U_{[1]} V_{[3]} + U_{[2]} V_{[2]} + U_{[3]} V_{[1]}$$

⋮

$$W_{[n]} = U_{[1]} V_{[n]} + U_{[2]} V_{[n-1]} + \dots + U_{[n]} V_{[1]}$$

Ejemplo 1

Supóngase que se tiene una señal $f(t)$ discretizada, cuyas muestras son:

$$f_{[n]} = [2, 4, 1, 6, 4, 5, 1, 3]$$

Determinar los coeficientes escala y los coeficientes wavelet en el primer nivel de descomposición usando el sistema Haar o Daubechies1.

Como los coeficientes escala (de *aproximación*) y los coeficientes wavelet (de *detalle*) se determinan mediante las ecuaciones (2.17a) y (1.17b) respectivamente, se necesita conocer los coeficientes del filtro del sistema Haar los cuales ya están tabulados, así que los filtros son:

$$\phi_{[n]} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \text{ Filtro escala o filtro pasabajo de descomposición}$$

$$\psi_{[n]} = \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \text{ Filtro wavelet o filtro pasaalto de descomposición}$$

En la figura 2.15 se muestra el proceso que se lleva a cabo en el cálculo de los coeficientes de aproximación y detalle.

Primero se realiza la convolución entre la señal $f_{[n]}$ y los filtros pasaalto y pasabajo, luego se realiza un submuestreo para eliminar muestras redundantes. Posteriormente se construye el vector wavelet.

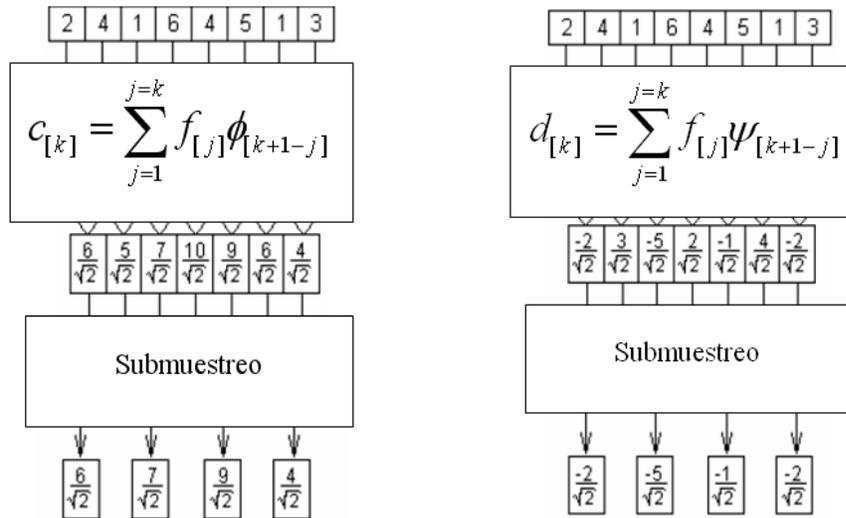


Figura 2.15: Diagrama algorítmico para calcular los coeficientes de aproximación y detalle de una señal cualquiera f que ha sido muestreada.

Fuente: Yusti F. (2010).

En este primer nivel de descomposición el vector wavelet $V_w[n]$ estará formado por los 4 coeficientes de aproximación seguidos de los 4 coeficientes de detalle.

$$V_w[n] = \left[\frac{6}{\sqrt{2}}, \frac{7}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{4}{\sqrt{2}}, \frac{-2}{\sqrt{2}}, \frac{-5}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-2}{\sqrt{2}} \right]$$

Ejemplo 2

Para la secuencia del ejemplo anterior, realizar mediante el sistema Haar todos los niveles de descomposición y determinar el vector wavelet en el último nivel de descomposición.

La secuencia del ejemplo anterior tiene 8 ($2^3 = 2^N$) muestras, por lo tanto N , que representa el número máximo de niveles de descomposición en este caso es $N = 3$.

En la figura 2.16 se muestra un esquema que describe como la descomposición multinivel sólo toma en cuenta la salida del filtro pasabajo para futuras descomposiciones, así finalmente el vector wavelet en el último nivel de descomposición estará formado por un solo coeficiente de aproximación y el resto de las componentes son coeficientes de detalles. Por lo tanto de la figura 2.16, se puede intuir que una función que esté en V_{j_0} se puede representar también por la ecuación (2.16) la cual no es más que la ecuación (2.5) con una correcta notación para las sumatorias que dependen de los términos j y k . La figura 2.16 representa una completa descomposición multinivel de una señal muestreada, proceso denominado *análisis*.

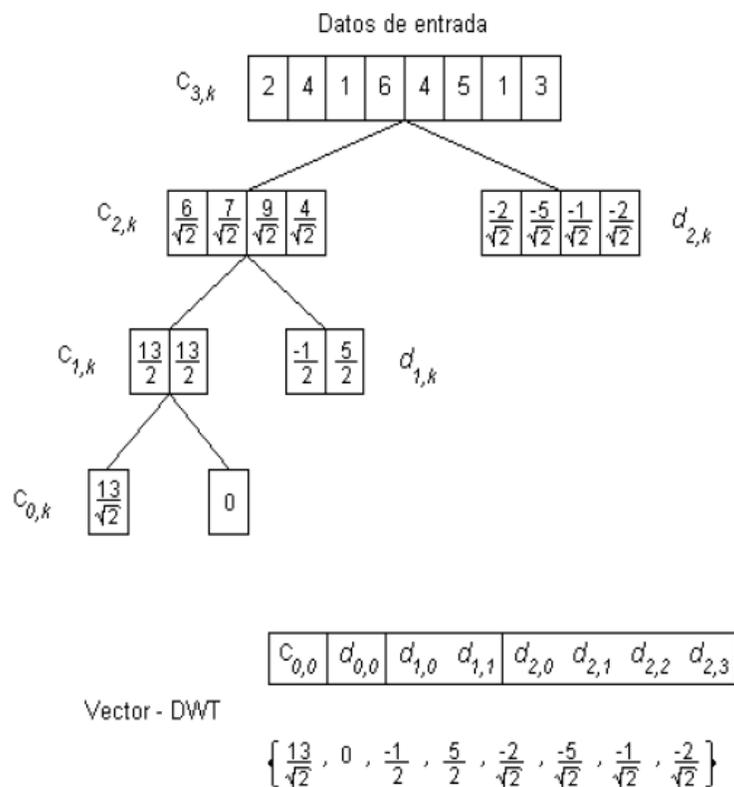


Figura 2.16: Descomposición multinivel de una señal muestreada con 8 muestras. Se observa la formación del vector wavelet para cada nivel de descomposición, proceso denominado análisis.

Fuente: Faundez P., et all (2002).

Reconstrucción De Señales (Síntesis)

Al proceso de reconstrucción de señales se le denomina síntesis y corresponde a la inversa de la transformada discreta wavelet (IDWT). Este proceso consiste en recuperar la señal original sin que se produzca pérdida de información a partir de las componentes obtenidas durante el proceso de análisis.

Por lo tanto, lo que se desea es lograr representar los coeficientes escala en un nivel de resolución más alto mediante una combinación de los coeficientes escala y wavelets en un nivel de resolución más bajo.

Los procesos de análisis y síntesis son procesos iterativos de tal manera que en teoría pueden repetirse en forma infinita con la salvedad que el proceso de síntesis depende del proceso de análisis. Sin embargo, en la práctica estos procesos no pueden repetirse de manera infinita, siendo el nivel de resolución de la señal original el que pone el límite.

Una explicación más detallada puede expresarse de la siguiente manera:

Supóngase una señal con una longitud $N=2^n$, en el análisis se divide la señal original en una aproximación y un detalle correspondientes al primer nivel de descomposición, luego la aproximación que queda de longitud igual a 2^{n-1} es nuevamente dividida obteniendo una nueva aproximación y detalle correspondientes a un segundo nivel de descomposición. Este procedimiento se vuelve a repetir hasta que la aproximación y el detalle estén representados por un solo coeficiente, es decir, tienen una longitud de $1=2^0$, lo que significa que el número de iteraciones posibles de

realizar es de $n=\log_2 N$. De esta forma se obtiene un vector de longitud N que contiene un sólo término encargado de representar la forma general de la señal (coeficiente escala) y todos los otros términos con información sobre el detalle obtenido en los diferentes niveles de descomposición (coeficientes wavelet), tal como se ilustró en la figura 2.16 donde aparece el vector al cual se le denomina vector-DWT.

La síntesis por su lado toma la aproximación y el detalle, aumenta su longitud al doble mediante el súper muestreo y realiza la convolución discreta con los respectivos filtros, obteniéndose como resultado una mejor aproximación a la señal correspondiente al primer nivel de reconstrucción.

Lógicamente, el número de veces que se realiza el proceso de reconstrucción hasta llegar a la señal original depende del grado de descomposición al que se llegó en el proceso de análisis.

Para explicar cómo se realiza una descomposición y una reconstrucción, se ilustrará el proceso utilizando la wavelet Haar y suponiendo una señal $F_{[n]}=[2,4,1,6,4,5,1,3]$, la descomposición se observa en la figura 2.17.

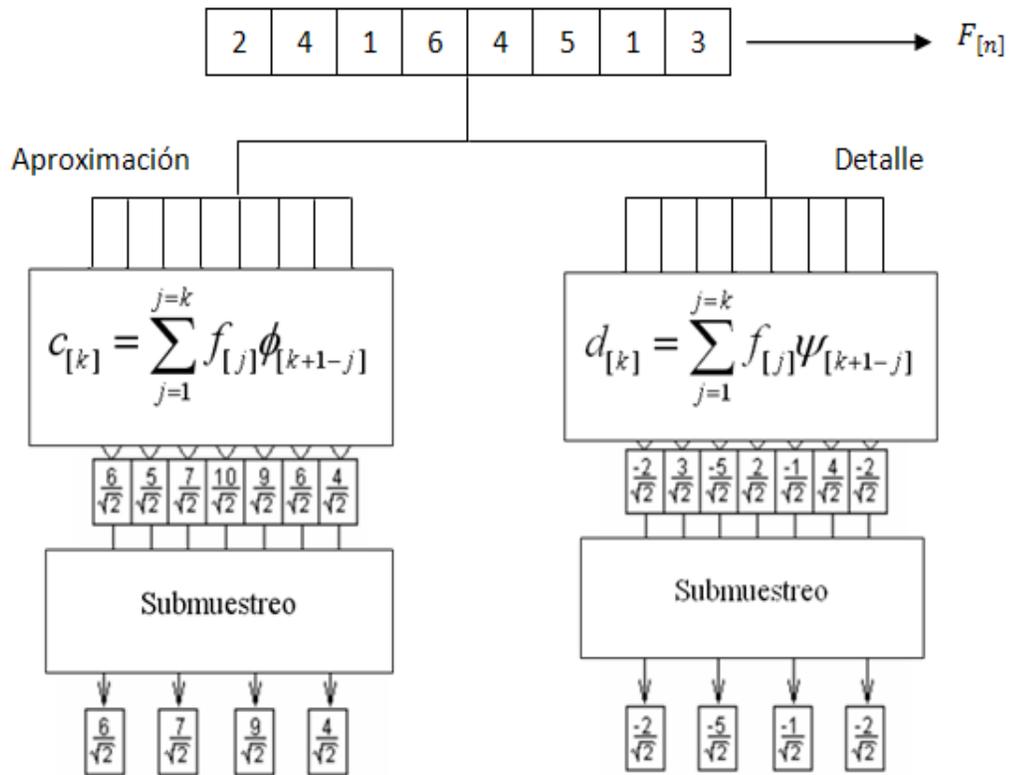


Figura 2.17: Se muestra el primer nivel de descomposición de una señal mediante el sistema Haar.

Fuente: Yusti F. (2010).

Como la aproximación está relacionada con un promedio y el detalle con diferencias, entonces, se puede obtener la primera descomposición de acuerdo a las ecuaciones (2.19a) y (2.19b) donde los c_j son los coeficientes correspondientes a la señal original, es decir los datos de entrada representados por $F_{[n]}$ y J es el nivel de resolución más alto con el que se puede trabajar, $N=2^J$ es la longitud de la señal original, para este caso $F_{[n]}$ es una señal de longitud $N = 8 = 2^3$.

$$c_{j-1,k} = \frac{1}{\sqrt{2}}(c_{j,2k} + c_{j,2k+1}) \quad (2.19a)$$

$$d_{j-1,k} = \frac{1}{\sqrt{2}}(c_{j,2k} - c_{j,2k+1}) \quad (2.19b)$$

Aplicando el algoritmo de descomposición, se puede observar en la figura 2.17 como se realiza la convolución discreta entre los filtros escalas y los filtros wavelet con la señal original, además se observa el posterior submuestreo de tal forma que se obtienen dos conjuntos de coeficientes, uno encargado de la aproximación a la señal original y el otro encargado del detalle, ambos de longitud $2^{3-1}=4$.

El paso siguiente es mantener el detalle y volver a aplicar el algoritmo a los coeficientes de aproximación dando como resultado una nueva aproximación mas general y un nuevo detalle, ambos de longitud $2^{3-2}=2$. Este proceso se podrá repetir una vez mas ya que tanto la longitud de la nueva aproximación como del nuevo detalle será igual a $2^{3-3}=1$.

Para reconstruir la señal a partir de los coeficientes escala y los coeficientes wavelet pertenecientes al primer nivel de descomposición, se observa que sumando y restando las ecuaciones (2.19a) y (2.19b) se obtienen:

$$c_{J,2k} = \frac{1}{\sqrt{2}}(c_{J-1,k} + d_{J-1,k}) \quad (2.20a)$$

$$c_{J,2k+1} = \frac{1}{\sqrt{2}}(c_{J-1,k} - d_{J-1,k}) \quad (2.20b)$$

Estas últimas ecuaciones (2.20a) y (2.20b) permiten realizar una reconstrucción perfecta de la señal. En la figura 2.18 se puede ver como se realiza el proceso de súper muestreo y posteriormente la convolución para reconstruir la señal en forma perfecta mediante la suma de los coeficientes de reconstrucción escala y wavelet

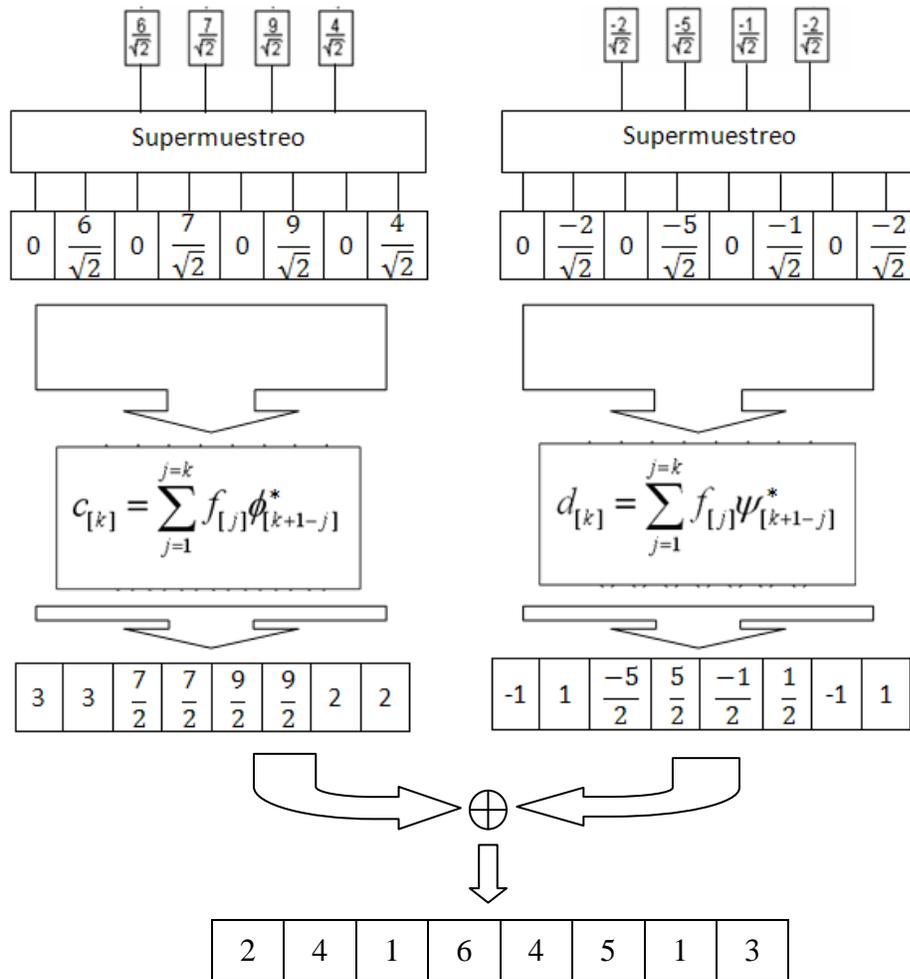


Figura 2.18: Reconstrucción de una señal utilizando el sistema Haar, la señal ha sido descompuesta sólo hasta el primer nivel de descomposición.

Fuente: Yusti F. (2010).

Se observa en esta última figura (figura 2.18) cómo se obtiene la reconstrucción de la señal representada por $f_{[n]}$ en la figura 2.17, además se puede ver que el filtro escala de reconstrucción y el filtro wavelet de reconstrucción se representan con un asterisco de supra índice ya que no son los mismos que fueron utilizados en la descomposición pero están altamente relacionados.

Considerando esta última parte en la que se ha advertido sobre la existencia de filtros de reconstrucción se procederá a mostrar en el sistema Haar los filtros de descomposición y los filtros de reconstrucción utilizados en los procesos descritos en las figuras 2.17 y 2.18.

Función escala (filtro pasabajo) y función wavelet (filtro pasa alto) discretizadas, sistema Haar

Filtro Pasabajo de Descomposición

$$f_e = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]$$

Filtro Pasa bajo de Reconstrucción

$$f_e^* = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]$$

Filtro Pasa alto de Descomposición

$$f_w = \left[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]$$

Filtro Pasa alto de Reconstrucción

$$f_w^* = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right]$$

Se puede notar la ortogonalidad entre las secuencias que describen a la función escala y las secuencias que describen a la función wavelet, esto quiere decir, tanto para los filtros de descomposición como para los filtros de reconstrucción se cumple que el producto interno (en este caso se trata de un producto interno usual en \mathbb{R}^n) $f_e \cdot f_w = 0$ o también $f_e^* \cdot f_w^* = 0$

Existen diferentes sistemas para realizar la transformada wavelet de una señal, entre ellos se pueden contabilizar, el sistema Haar (también conocido como Daubechies1), la familia Daubechies, la familia Biortogonal, la familia Symlets y la familia Coiflets entre otros.

Para observar la relación que existe entre los filtros escala y los filtros wavelet se utilizará el sistema Daubechies2 conocido como Db2, de allí se comprenderá como a partir del filtro escala se puede conocer el filtro wavelet.

$$f_e = \left[\frac{1 - \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{1 + \sqrt{3}}{4\sqrt{2}} \right]$$

$$f_w = \left[-\frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, -\frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}} \right]$$

Primero que nada se observa la ortogonalidad entre los filtros, esto es $f_e \cdot f_w = 0$, segundo, se puede ver como el filtro wavelet no es más que recorrer el filtro escala al revés y cambiarle el signo a los que queden en posición impar en el filtro wavelet.

Para obtener los filtros de reconstrucción simplemente se recorren en orden inverso los filtros de descomposición.

$$f_e^* = \left[\frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}} \right]$$

$$f_w^* = \left[\frac{1 - \sqrt{3}}{4\sqrt{2}}, -\frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, -\frac{1 + \sqrt{3}}{4\sqrt{2}} \right]$$

Hasta el momento se ha realizado una reconstrucción para una señal que sólo ha sido descompuesta en su primer nivel de descomposición. Sin embargo el objetivo

principal de un análisis wavelet es obtener el vector-Dwt ya que este es el que contiene información útil para la aplicación de esta herramienta en diferentes campos del procesamiento de señales, entiéndase por estos, la reducción de ruido, la aproximación no lineal, la compresión de datos, la detección de singularidades, parámetros característicos, etc. En la figura 2.19 se muestra la descomposición completa (en todos sus niveles de descomposición de la señal $f_{[n]}$).

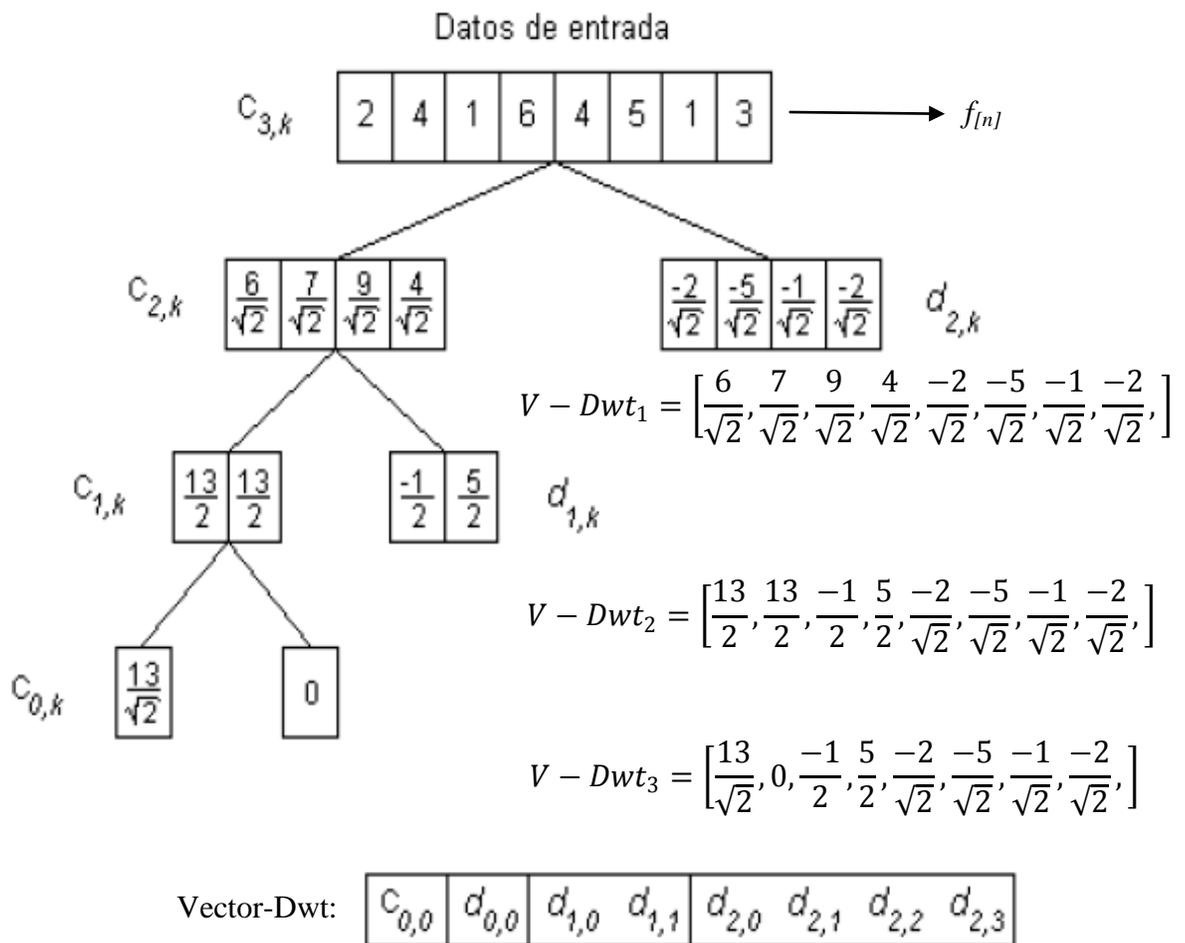


Figura 2.19: Descomposición multinivel (análisis) de una señal de $8=2^3$ muestras con sus respectivos vectores wavelet en cada nivel de descomposición, además se observa que $0 < k < 2^j$.

Fuente: Yusti F. (2010).

Ahora se procederá a ilustrar el proceso de reconstrucción total de la señal. Este proceso se observa resumidamente a continuación, es de hacer notar que se está realizando la reconstrucción de la señal descompuesta en la figura 2.19.

Reconstrucción de los coeficientes en el nivel $J=0$

$$c_{0,k} \quad \left[\frac{13}{\sqrt{2}} \right] \quad \text{supermuestrear} \quad * f_e^* = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \quad \left[\frac{13}{2}, \frac{13}{2} \right]$$

$$d_{0,k} \quad [0] \quad \text{supermuestrear} \quad * f_w^* = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \quad [0,0]$$

Los coeficientes de aproximación del nivel $J=1$ serán

$$\left[\frac{13}{2}, \frac{13}{2} \right] - [0,0] = \left[\frac{13}{2}, \frac{13}{2} \right]$$

Reconstrucción de los coeficientes en el nivel $J=1$

$$c_{1,k} \quad \left[\frac{13}{2}, \frac{13}{2} \right] \quad \text{supermuestrear} \quad * f_e^* = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \quad \left[\frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}} \right]$$

$$d_{1,k} \quad [0,0] \quad \text{supermuestrear} \quad * f_w^* = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \quad \left[\frac{-1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{5}{2\sqrt{2}}, \frac{-5}{2\sqrt{2}} \right]$$

Los coeficientes de aproximación del nivel $J=2$ serán

$$\left[\frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}}, \frac{13}{2\sqrt{2}} \right] + \left[\frac{-1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{5}{2\sqrt{2}}, \frac{-5}{2\sqrt{2}} \right] = \left[\frac{6}{\sqrt{2}}, \frac{7}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{4}{\sqrt{2}} \right]$$

Reconstrucción de los coeficientes en el nivel $J=2$

$$c_{2,k} \left[\frac{6}{\sqrt{2}}, \frac{7}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{4}{\sqrt{2}} \right] \text{ supermuestrear} \quad * f_e^* = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \quad \left[3, 3, \frac{7}{2}, \frac{7}{2}, \frac{9}{2}, \frac{9}{2}, 2, 2 \right]$$

$$d_{2,k} \left[\frac{-2}{\sqrt{2}}, \frac{-5}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-2}{\sqrt{2}} \right] \text{ supermuestrear} \quad * f_w^* = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \quad \left[-1, 1, \frac{-5}{2}, \frac{5}{2}, \frac{-1}{2}, \frac{1}{2}, -1, 1 \right]$$

$$f_{[n]} = \left[3, 3, \frac{7}{2}, \frac{7}{2}, \frac{9}{2}, \frac{9}{2}, 2, 2 \right] + \left[-1, 1, \frac{-5}{2}, \frac{5}{2}, \frac{-1}{2}, \frac{1}{2}, -1, 1 \right] = [2, 4, 1, 6, 4, 5, 1, 3]$$

Descomposición (Análisis) Wavelet Packet

El método wavelet packet es una generalización de la descomposición wavelet que ofrece un rango más amplio de posibilidades para el análisis de señales. En el análisis wavelet, una señal es dividida en dos partes, una aproximación y un detalle. La aproximación es luego dividida en un segundo nivel de aproximación y detalle, este proceso es repetido en forma sucesiva. Para un nivel n de descomposición hay n+1 formas posibles de descomponer o codificar la señal.

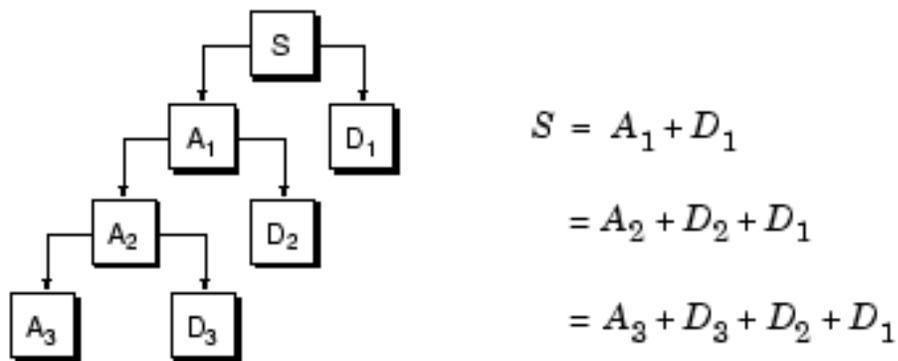


Figura 2.20: Descomposición Wavelet (análisis multinivel).
Fuente: Chui C. (1992).

En la descomposición (análisis) de una señal discreta usando *wavelet Packet* se realiza un análisis multi-resolución tanto en los coeficientes de aproximación como en los coeficientes de detalles. Es decir, para cada nivel de descomposición, las componentes de alta frecuencia (coeficientes de detalles) también se hacen pasar sucesivamente por filtros paso bajo y paso alto obteniéndose un árbol de descomposición como el mostrado en la figura 2.21.

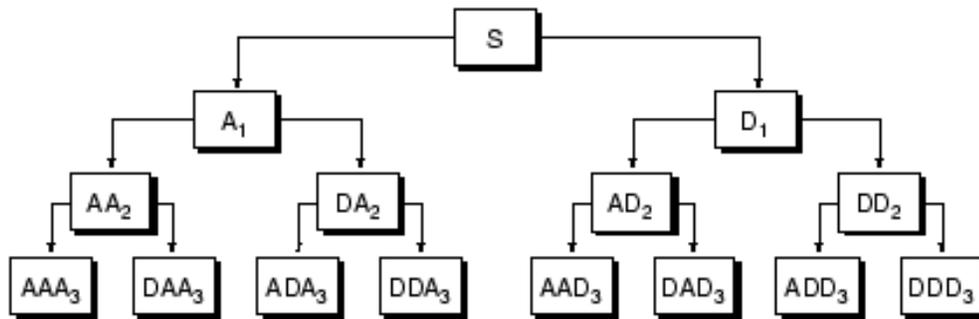


Figura 2.21: Descomposición Wavelet Packet.

Fuente: Chui C. (1992).

Así, en el j -ésimo nivel de descomposición usando *wavelet packet* se tendrán 2^j secuencias de salida conteniendo cada una 2^{N-j} coeficientes, siendo 2^N el número de muestras de la señal original.

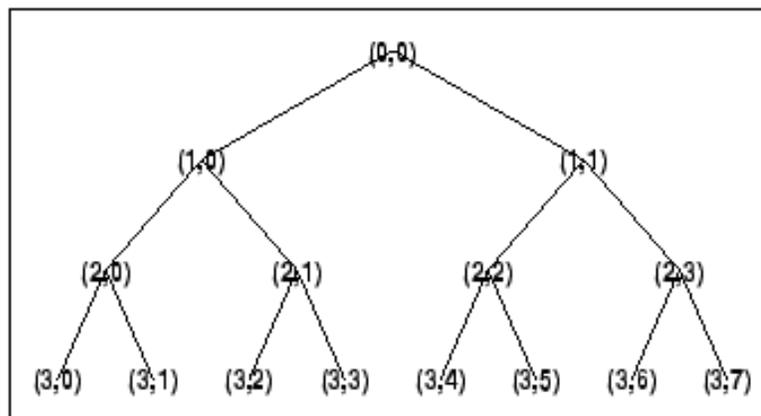


Figura 2.22: Árbol Binario de Descomposición Wavelet Packet.

Fuente: Kaiser G. (1994).

La figura 2.22 ilustra tres niveles de descomposición usando *wavelet packet*, donde los nodos del árbol representan los coeficientes *wavelet*, $d_{j,i}$, siendo j el nivel de descomposición e i la posición en el árbol a ese nivel. $d_{j,i}$ puede verse como un vector fila de 2^{N-j} componentes.

Análisis de Componentes Principales (ACP)

El Análisis de Componentes Principales (ACP) es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables). Es decir, ante un banco de datos con muchas variables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible. Los nuevos componentes principales o factores serán una combinación lineal de las variables originales, y además serán independientes entre sí.

Un análisis de componentes principales tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que existe información redundante y, por tanto, pocos factores explicarán gran parte de la variabilidad total.

La elección de los factores se realiza de tal forma que el primero recoja la mayor proporción posible de la variabilidad original; el segundo factor debe recoger la máxima variabilidad posible no recogida por el primero, y así sucesivamente. Del total de factores se elegirán aquéllos que recojan el porcentaje de variabilidad que se considere suficiente. A éstos se les denominará componentes principales.

Una vez seleccionados los componentes principales, se representan en forma de matriz. Cada elemento de ésta representa los coeficientes factoriales de las variables

(las correlaciones entre las variables y los componentes principales). La matriz tendrá tantas columnas como componentes principales y tantas filas como variables.

Para obtener las componentes principales, supóngase que se tiene una matriz X tal que cada una de sus filas indica el número de observaciones que se han realizado y en las columnas se indica el número de variables que se han observado. El primer paso es obtener la matriz de correlación S mediante:

$$S = \frac{1}{n-1} X' * X \quad (2.21)$$

El siguiente paso será calcular los eigenvalores y los eigenvectores de la matriz S . Cada eigenvalor representará la varianza explicada de cada variable observada; entonces para reducir el número de variables que ofrecen la misma información que el total de variables se puede utilizar el criterio de Kayser el cual establece que deben tomarse las variables cuyos eigenvalores arrojen un valor mayor que 1 o en su defecto puede utilizarse el criterio de Jolliffe I., (2002:72) el cual establece “seleccionar las variables cuyos eigenvalores arrojen valores mayores a 0.7”.

Máquinas de Vectores de Soporte (SVM)

Una Máquina de Vectores de Soporte (SVM) aprende la superficie de decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un Kernel Gaussiano u otro tipo de Kernel a un espacio de características en un espacio dimensional más alto, donde se busca la máxima

separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada puede separar los datos en todas las clases distintas, cada una formando un agrupamiento.

La teoría de las máquinas de vectores de soporte es una técnica de clasificación que ha sido aplicada en numerosas investigaciones en años recientes; esta teoría está basada en la idea de minimización de riesgo estructural (SRM). En muchas aplicaciones, las SVM han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación.

Asfur B., et all (2007:33) establece que “una SVM primero mapea los puntos de entrada a un espacio de característica de una dimensión mayor y encuentra un hiperplano que los separe y maximice el margen m entre las clases en este espacio”, en la figura 2.23 se muestra a grandes rasgos el proceso de clasificación entre dos clases de muestras para el caso linealmente separable.

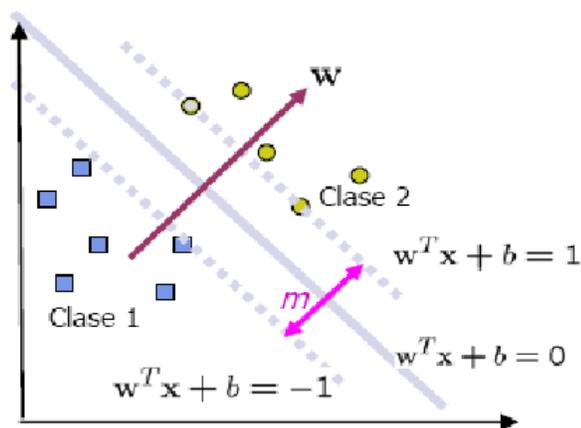


Figura 2.23: Clasificación SVM, caso linealmente separable.
Fuente: Asfur B, et all (2007).

La frontera de decisión debe estar tan lejos de los datos de ambas clases como sea posible. Maximizar el margen m es un problema de programación cuadrática y puede ser resuelto aplicando la técnica de multiplicadores de Lagrange. Asfur B., et al (2007:33) afirma que “la SVM encuentra el hiperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas Kernel”. El hiperplano óptimo solución puede ser escrito como la combinación lineal de unos pocos puntos de entrada que son llamados *Vectores de Soporte*.

Clasificación en dos Dimensiones

En el siguiente ejemplo idealizado para 2-dimensiones, la representación de los datos a clasificar se realiza en el plano x - y . El algoritmo SVM trata de encontrar un hiperplano 1-dimensional (en el ejemplo que nos ocupa es una línea) que une a las variables predictoras y constituye el límite que define si un elemento de entrada pertenece a una categoría o a la otra. Existe un número infinito de posibles hiperplanos (líneas) tal como se muestra en la figura 2.24 que realicen la clasificación pero, ¿cuál es la mejor y cómo la definimos?

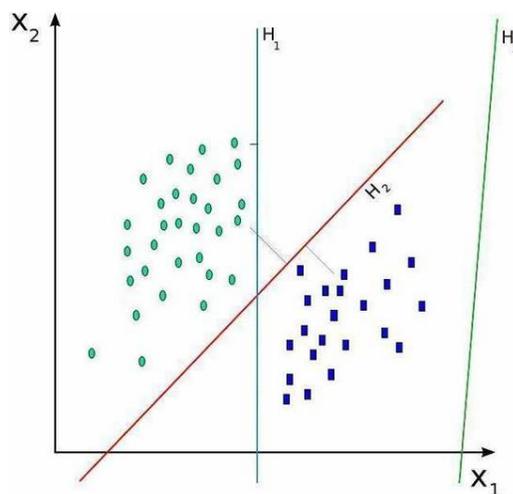


Figura 2.24: Clasificación en dos dimensiones, hiperplanos separadores. Hay infinitos hiperplanos posibles.

Fuente: <http://www.support-vector.net/> (2014).

La mejor solución es aquella que permita un margen máximo entre los elementos de las dos categorías.

Se denominan vectores de soporte (figura 2.25) a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible.

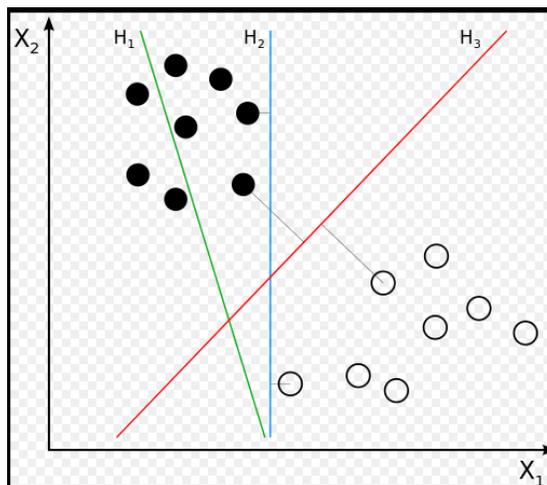


Figura 2.25: H_1 no separa las clases. H_2 las separa, pero solo con un margen pequeño. H_3 las separa con el margen máximo.

Fuente: <http://www.support-vector.net/> (2014).

Errores de Entrenamiento (Soft Margin)

Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (overfitting).

Con el fin de permitir cierta flexibilidad, las SVM manejan un parámetro C que controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando (soft margin) que permita algunos errores en la clasificación a la vez que los penaliza.

Consideraciones Alrededor de las SVM

Las SVM son básicamente clasificadoras para dos clases. Los datos son divididos inteligentemente en dos partes de diferentes formas y una SVM es entrenada para cada forma de división.

Fortalezas

El entrenamiento es relativamente fácil de realizar.

El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.

Datos no tradicionales como cadenas de caracteres y arboles pueden ser usados como entradas a la SVM en vez de vectores de características.

Debilidades

Se necesita una buena función Kernel, es decir, se necesitan metodologías eficientes para sintonizar los parámetros de inicialización de la SVM.

Función Kernel

La manera más simple de realizar la separación es mediante una línea recta, un plano recto o un hiperplano N-dimensional. Desafortunadamente los universos a estudiar no se suelen presentar en casos idílicos de dos dimensiones como en el ejemplo anterior, sino que un algoritmo SVM debe tratar con:

- a) más de dos variables predictoras.
- b) curvas no lineales de separación.
- c) casos donde los conjuntos de datos no pueden ser completamente separados.
- d) clasificaciones en más de dos categorías.

Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquina de aprendizaje lineal. Es decir, se mapea el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad (Hilbert):

$$F = \{\varphi(x) | x \in X\} \quad (2.22)$$

$$x = \{x_1, x_2, \dots, x_n\} \rightarrow \varphi(x) = \{\varphi(x)_1, \varphi(x)_2, \dots, \varphi(x)_n\} \quad (2.23)$$

Tipos de Funciones Kernel (Núcleo)

Polinomial-homogénea (figura2.26):

$$K(x_i, x_j) = (x_i \cdot x_j)^n \quad (2.24)$$

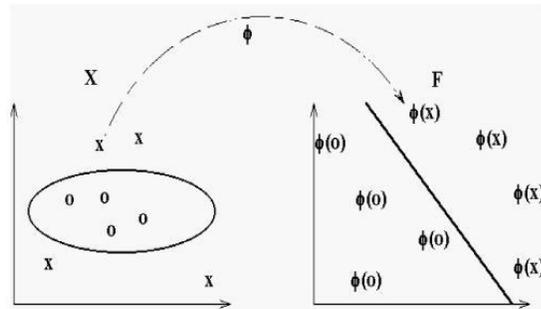


Figura 2.26: Función kernel polinomial SVM 8 polinomial.

Fuente: <http://www.support-vector.net/> (2014).

Perceptrón (figura2.27):

$$K(x_i, x_j) = \|x_i - x_j\| \quad (2.25)$$

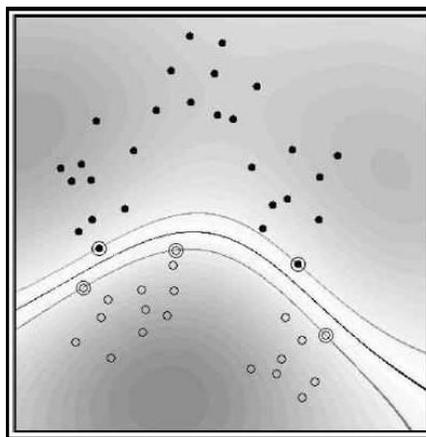


Figura 2.27: Svm 10 perceptrón.

Fuente: <http://www.support-vector.net/> (2014).

Función de Base Radial (rbf) Gaussiana (figura 2.28):

$$K(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}} \quad (2.26)$$

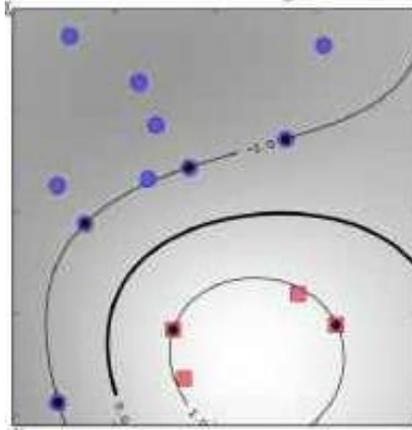


Figura 2.28: Clasificación con función kernel de base radial Gaussiana.

Fuente: <http://www.support-vector.net/> (2014).

Sigmoid:

$$K(x_i, x_j) = \tanh(x_i * x_j - \theta) \quad (2.27)$$

SVM Multiclase

Hay dos filosofías básicas para resolver el problema de querer clasificar los datos en más de dos categorías:

- a) cada categoría es dividida en otras y todas son combinadas.
- b) se construyen $k(k-1) / 2$ modelos donde k es el número de categorías.

Caso no Linealmente Separable

Si el conjunto S no es linealmente separable, tal como se muestra en la figura 2.29, violaciones a la clasificación deben ser permitidas en la formulación de la SVM

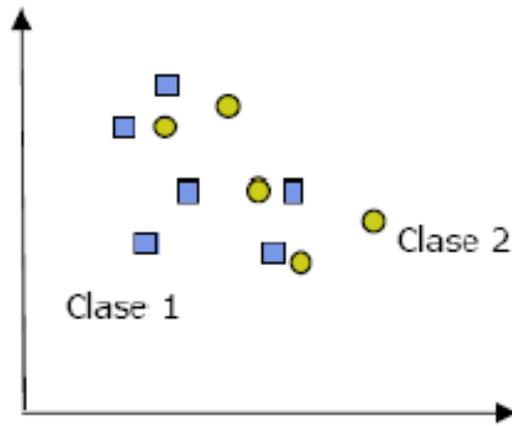


Figura 2.29: Conjunto no linealmente separable.
Fuente: Asfur B, et all (2007)

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no negativas $\xi \geq 0$ de tal modo que es modificado a:

$$y_i(w \cdot z_i + b) \geq 1 - \xi_i \quad i=1, \dots, l \quad (2.28)$$

El problema del hiperplano óptimo de separación es entonces redefinido como la solución al problema:

$$\min \left\{ \frac{1}{2} w \cdot w + C \sum_{i=1}^l \xi_i \right\} \quad (2.29)$$

sujeto a las condiciones: $y_i(w \cdot z_i + b) \geq 1 - \xi_i \quad i=1, \dots, l$
 $\xi_i \geq 0, \quad i = 1, \dots, l$

Donde C es una constante. El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de este parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación.

Buscar el hiperplano óptimo es un problema de programación cuadrática QP, que puede ser resuelto construyendo un Lagrangiano y transformándolo en el dual:

$$\begin{aligned}
 \text{Max } W(\alpha) &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j z_i \cdot z_j \\
 \text{sujeto a } \sum_{i=1}^l y_i \alpha_i &= 0 \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l
 \end{aligned} \tag{2.30}$$

Donde α es un vector de multiplicadores de Lagrange positivos asociados con las constantes en la ecuación anterior. Para una revisión con mayor profundidad sobre las SVM se puede citar Jiménez C., Jiménez J. (2010).

CAPITULO III

MARCO METODOLÓGICO

En este capítulo se presenta una descripción metodológica de este trabajo de investigación clasificando el mismo según su tipo, diseño, y modalidad. Además se presentan las técnicas utilizadas para la recolección de los datos y las fases en las que se ha dividido este proyecto para lograr el cumplimiento de los objetivos propuestos en esta investigación.

Tipo y modalidad de investigación

La investigación aquí propuesta, destinada a determinar parámetros característicos en las señales de voz utilizando transformada wavelet empaquetada y análisis de componentes principales (ACP) para su clasificación mediante maquinas de vectores de soporte presenta características de una investigación del tipo científicista, la cual según Orozco Labrador y Palencia (2002:7) “se presenta si el interés del investigador está dirigido a encontrar respuestas en relación con lo no conocido mediante evidencia empírica, a través del uso de técnicas de recolección y análisis cuantitativo de los datos”, en esta investigación, se pretende determinar los parámetros característicos de las voces mediante la puesta en marcha de procedimientos experimentales empíricos y además se persigue la búsqueda de nuevos parámetros a través de procedimientos prácticos mediante análisis estadístico de los resultados obtenidos lo que le da a esta investigación el enfoque cuantitativo respecto a la observación y análisis de los datos.

Así mismo, tomando en cuenta el nivel de conocimiento junto a los propósitos que se pretenden alcanzar en este trabajo de investigación, el mismo presenta características de una investigación enfocada a la modalidad descriptiva, la cual según Orozco, Labrador y Palencia (2002:8), es aquella que “tiene como propósito la descripción cuantitativa de un evento o fenómeno tal cual ocurre en la realidad. En este tipo de investigación no se formulan hipótesis”. En este trabajo de investigación se pretende caracterizar las señales de voz mediante la cuantificación de diferentes parámetros como por ejemplo la desviación estándar de las muestras, la relación señal a ruido y otros, es decir, se hará una descripción cuantitativa mediante el cálculo numérico de los parámetros obtenidos en cada muestra de señal de voz utilizada.

Diseño de la investigación

Tomando en consideración que el cálculo de los parámetros característicos de las señales de voz que se utilizaron en esta investigación se realizó una única vez por cada muestra de voz, es decir, el resultado observado se hizo en un momento único en el tiempo, y los mismos se analizaron tal cual como fueron obtenidos sus valores, se puede observar que el presente proyecto se enmarca bajo un diseño de investigación transeccional, tal como lo define Orozco, Labrador y Palencia (2002:9) “En los estudios transeccionales se toman los datos de una o varias muestras en una sola aplicación. Es decir, se describe en un momento único cualquier evento”.

Población y Muestra

Para este trabajo de investigación enfocado en la caracterización y clasificación de las señales de voz la población referida estará formada por todas las señales de voz

existentes a nivel mundial, este conjunto de señales de voz será tomado en consideración como población objetivo, según Lohr S. (2010:3) “La población objetivo es la colección completa de observaciones que deseamos estudiar”.

Así mismo, tomando en cuenta la imposibilidad de realizar pruebas sobre toda la población de señales de voz existentes a nivel mundial se procederá a delimitar el conjunto y trabajar sobre una muestra de la población total, una muestra de la población está definida según Orozco, Labrador y Palencia (2002:84) como sigue: “es un subconjunto de la población. Son los sujetos involucrados en el estudio. Es decir, es la unidad contextual que aporta la información”. Por lo tanto para cumplir con los objetivos de este trabajo de investigación se tomará como muestra un conjunto de señales de voz seleccionadas y clasificadas por un experto suministradas en formato digital provenientes de grabaciones de voz las cuales estarán almacenadas en un dispositivo de almacenamiento masivo ya sea CD, DVD o Memory Flash.

Técnicas e Instrumentos de Recolección de Datos

La técnica empleada para la recolección de las señales de voz será la selección de las mismas a través del análisis previo realizado por un experto en el estudio de voces, el cual suministrará la base datos a estudiar.

Los instrumentos utilizados para recolectar las voces serán los dispositivos de grabación de voz y los computadores destinados a digitalizar las mismas, son los encargados de convertir las señales analógicas de voz en señales digitales para que puedan ser almacenadas en los diferentes tipos de dispositivos de almacenamiento digital.

Selección de la Base de Datos

Para el desarrollo de este trabajo se seleccionó una base de datos formada por muestras de señales de voz de personas sanas y muestras de señales de voz de personas con alguna patología en el habla, es de hacer notar que dichas muestras de señales de voz son digitales, por lo tanto se pueden representar en el dominio del tiempo discreto. La base de datos seleccionada consta de 50 registros de voces de individuos sanos y 99 registros de voces de individuos con patologías; todos los registros de voz fueron almacenados en formato *.NSP*.

Un archivo con extensión *.NSP* según lo establece Kabal P. (2011:28) “es un archivo guardado en formato de audio creado por KayPENTAX (Kay Elemetrics) específicamente en el *Computerized Speech Lab* (CSL: Laboratorio Computarizado del Habla)”. El formato de audio *.NSP* está formado con la siguiente estructura en su encabezado, tal como se indica en la tabla 3.1:

Tabla 3.1: Estructura del encabezado de un archivo *.NSP*.

Offset	Length	Type	Contents
0	4	character	Identifier: "FORM"
4	4	character	Chunk identifier: "DS16"
8	4	integer	Chunk length
12	-	-	Chunk data

Fuente: Kabal P. (2011).

El “chunk identifier DS16” indica que la data está codificada con un entero de 16bits. Los datos de audio se encuentran a partir de la posición 12 en el offset del archivo.

En las tabla 3.2 y 3.3 están resumidas las señales de voz utilizadas indicando algunas de sus características.

Tabla 3.2: Base de datos utilizadas para voces sanas (50 individuos).

Muestra	Edad	Sexo	Diagnóstico	Muestra	Edad	Sexo	Diagnóstico
AXH1NAL.NSP	29	F	normal voice	LLA1NAL.NSP	30	F	normal voice
BJB1NAL.NSP	34	M	normal voice	LMV1NAL.NSP	43	F	normal voice
BJV1NAL.NSP	52	F	normal voice	LMW1NAL.NSP	45	F	normal voice
CAD1NAL.NSP	31	F	normal voice	MAS1NAL.NSP	37	M	normal voice
CEB1NAL.NSP	43	F	normal voice	MCB1NAL.NSP	28	F	normal voice
DAJ1NAL.NSP	26	F	normal voice	MFM1NAL.NSP	28	M	normal voice
DFP1NAL.NSP	34	F	normal voice	MJU1NAL.NSP	26	M	normal voice
DMA1NAL.NSP	24	F	normal voice	MXB1NAL.NSP	24	F	normal voice
DWS1NAL.NSP	32	M	normal voice	MXZ1NAL.NSP	28	F	normal voice
EDC1NAL.NSP	32	F	normal voice	NJS1NAL.NSP	39	F	normal voice
EJC1NAL.NSP	44	M	normal voice	OVK1NAL.NSP	29	M	normal voice
FMB1NAL.NSP	28	M	normal voice	PBD1NAL.NSP	40	F	normal voice
GPC1NAL.NSP	40	M	normal voice	PCA1NAL.NSP	36	M	normal voice
GZZ1NAL.NSP	47	M	normal voice	RHG1NAL.NSP	59	M	normal voice
HBL1NAL.NSP	25	F	normal voice	RHM1NAL.NSP	40	M	normal voice
JAF1NAL.NSP	31	F	normal voice	RJS1NAL.NSP	46	M	normal voice
JAN1NAL.NSP	30	F	normal voice	SCK1NAL.NSP	33	F	normal voice
JAP1NAL.NSP	40	F	normal voice	SCT1NAL.NSP	39	F	normal voice
JEG1NAL.NSP	26	F	normal voice	SEB1NAL.NSP	37	F	normal voice
JMC1NAL.NSP	45	M	normal voice	SIS1NAL.NSP	36	M	normal voice
JTH1NAL.NSP	31	F	normal voice	SLC1NAL.NSP	22	F	normal voice
JXC1NAL.NSP	43	F	normal voice	SXV1NAL.NSP	38	M	normal voice
KAN1NAL.NSP	55	M	normal voice	TXN1NAL.NSP	39	M	normal voice
LAD1NAL.NSP	40	F	normal voice	VMC1NAL.NSP	44	F	normal voice
LDP1NAL.NSP	22	F	normal voice	WDK1NAL.NSP	39	M	normal voice

Fuente: Yusti F. (2015).

Tabla 3.3: Base de datos utilizadas para voces con patologías (99 individuos).

Muestra	Diagnóstico	Muestra	Diagnóstico	Muestra	Diagnóstico
AAK02AN.NSP	patológica	CMR01AN.NSP	patológica	EGW23AN.NSP	patológica
AAS16AN.NSP	patológica	CMS25AN.NSP	patológica	EMD08AN.NSP	patológica
AAT30AN.NSP	patológica	CNP07AN.NSP	patológica	EOW04AN.NSP	patológica
AAT31AN.NSP	patológica	CNR01AN.NSP	patológica	ERS07AN.NSP	patológica
ADP02AN.NSP	patológica	CPK19AN.NSP	patológica	ESM05AN.NSP	patológica
ADP11AN.NSP	patológica	CPK21AN.NSP	patológica	ESP04AN.NSP	patológica
AFR17AN.NSP	patológica	CSJ16AN.NSP	patológica	EXL23AN.NSP	patológica
AHK02AN.NSP	patológica	CSY01AN.NSP	patológica	EXH21AN.NSP	patológica
AHS20AN.NSP	patológica	CXL08AN.NSP	patológica	FAH01AN.NSP	patológica
AJM29AN.NSP	patológica	CXM07AN.NSP	patológica	FJL23AN.NSP	patológica
AJP25AN.NSP	patológica	CXM14AN.NSP	patológica	FLW13AN.NSP	patológica
ALW27AN.NSP	patológica	CXN14AN.NSP	patológica	FMQ20AN.NSP	patológica
ALW28AN.NSP	patológica	CXP02AN.NSP	patológica	FRH18AN.NSP	patológica
AMJ23AN.NSP	patológica	CXT08AN.NSP	patológica	FSP13AN.NSP	patológica
AMP12AN.NSP	patológica	DAM08AN.NSP	patológica	FXE24AN.NSP	patológica
AMV23AN.NSP	patológica	DAS10AN.NSP	patológica	FXI23AN.NSP	patológica
ASR20AN.NSP	patológica	DAS24AN.NSP	patológica	GCU31AN.NSP	patológica
ASR23AN.NSP	patológica	DAS40AN.NSP	patológica	GEA24AN.NSP	patológica
BAS19AN.NSP	patológica	DBA02AN.NSP	patológica	GEK02AN.NSP	patológica
BAT19AN.NSP	patológica	DGL30AN.NSP	patológica	GJW09AN.NSP	patológica
BBR24AN.NSP	patológica	DGO03AN.NSP	patológica	GLB01AN.NSP	patológica
BGS05AN.NSP	patológica	DJM14AN.NSP	patológica	GLB22AN.NSP	patológica
BJH05AN.NSP	patológica	DJM28AN.NSP	patológica	GMS03AN.NSP	patológica
BJK29AN.NSP	patológica	DLL25AN.NSP	patológica	GMT01AN.NSP	patológica
BMK05AN.NSP	patológica	DLT09AN.NSP	patológica	GRS20AN.NSP	patológica
BMM09AN.NSP	patológica	DLW04AN.NSP	patológica	GSL04AN.NSP	patológica
CBR29AN.NSP	patológica	DMF11AN.NSP	patológica	HBS12AN.NSP	patológica
CCM15AN.NSP	patológica	DMR27AN.NSP	patológica	HLK01AN.NSP	patológica
CCP21AN.NSP	patológica	DMS01AN.NSP	patológica	HLK15AN.NSP	patológica
CCP29AN.NSP	patológica	DOA27AN.NSP	patológica	HML26AN.NSP	patológica
CDW03AN.NSP	patológica	EAL06AN.NSP	patológica	HXL28AN.NSP	patológica
CEN21AN.NSP	patológica	EAW21AN.NSP	patológica	HXR23AN.NSP	patológica
CFW04AN.NSP	patológica	EBJ03AN.NSP	patológica	JAJ10AN.NSP	patológica
CJP10AN.NSP	patológica	EGK30AN.NSP	patológica	JAJ22AN.NSP	patológica

Fuente: Yusti F. (2015).

Procesamiento y Análisis de los Datos

Una vez seleccionada la base de datos a utilizar se determinaron una serie de parámetros característicos que se consideraron necesarios para describir la señal, estos fueron calculados tanto en el dominio de la frecuencia como en el dominio wavelet para el cual se utilizó la transformada wavelet packet de la señal en estudio, en lo que respecta a la reducción de la presencia de ruido en las señales de voz se utilizó la técnica de contracción garrote la cual permitió obtener una nueva representación de las señales de voz y de este modo obtener una nueva caracterización de las señales..

En ese escenario de ideas, se aplicó el Análisis de Componentes Principales (ACP) al conjunto de parámetros característicos obtenidos para seleccionar sólo aquellos que ofrezcan mayor caracterización de la señal y reducir el número de parámetros de entrada a la máquina de vectores de soporte, la cual es utilizada en este trabajo de investigación para la clasificación de las señales de voz.

Para enfrentar esos retos de realizar los cálculos que implican cada una de las tareas antes mencionadas se utilizó herramientas computacionales de programación.

Procedimientos de la Investigación

Debido a que la base de datos de las señales de voz fue suministrada en formato digital, la herramienta fundamental para la lectura de la señal, el procesamiento de la misma, el cálculo de los parámetros y la clasificación fue el computador personal, teniendo en cuenta esto, el procedimiento metodológico que se utilizó para desarrollar

este trabajo de investigación se resume en el siguiente diagrama de bloques mostrado en la figura 3.1.

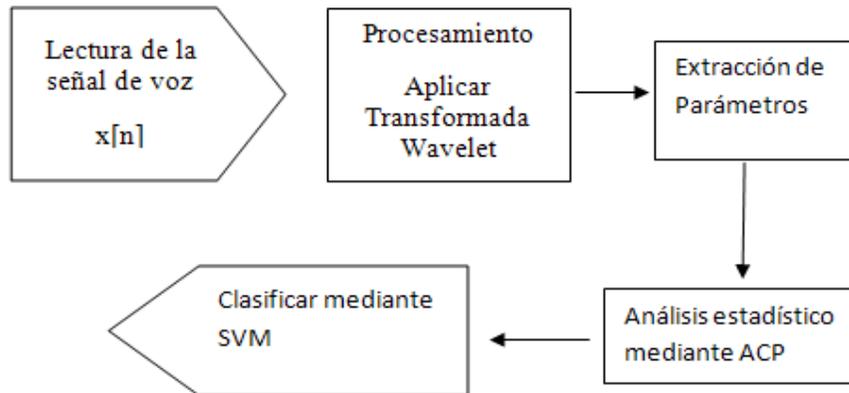


Figura 3.1: Diagrama de bloques del procedimiento metodológico.
Fuente: Yusti F. (2015).

Detallando a grandes rasgos el procedimiento metodológico a utilizar, en el mismo se desarrollaron las siguientes etapas:

Etapas I: Selección y Lectura de las Muestras de Voz.

Las señales de voz a procesar se tomaron de una base de datos suministrada por un experto en el área del tratamiento de voces, el procesamiento de los datos se realizó teniendo conocimiento previo de una clasificación inicial de las señales de voz en grupos de señales sanas y señales con patologías.

Para la lectura de las muestras de voz, cada señal de la base de datos es leída mediante las siguientes instrucciones:

```
fid = fopen('nombre de la muestra de voz .NSP');
fseek(fid, 'bof', 12)
Nx = fread(fid);
fclose(fid);
```

Se describe a continuación como operan los comandos antes indicados: Abrir cada uno de los archivos de la tabla 3.2 y tabla 3.3 con el comando fopen, este coloca el puntero en la posición inicial del archivo:

```
fid = fopen('nombre de la muestra de voz .NSP');
```

Colocar el puntero en la posición 12 (línea 12 del offset):

```
fseek(fid, 'bof', 12)
```

Leer los datos de la voz a partir de la posición indicada y finalmente cerrar el archivo:

```
Nx = fread(fid);
fclose(fid);
```

Etapa II: Determinación de Parámetros.

Para extraer los rasgos de la señal se procedió a determinar una serie de parámetros que son característicos de las señales de voz, estos se determinaron tanto en el dominio de la frecuencia como en el dominio wavelet. Los caracteres enumerados son los siguientes:

PMR: es la relación existente entre el valor pico del espectro de la señal de voz y su valor medio.

Se determinó el PMR aplicando los siguientes comandos a cada una de las muestras de voz:

```
k=abs(fft(SignalVoz,length(SignalVoz)));  
PMR=max(k)/mean(k);
```

PMRf: Es el PMR aplicado a la muestra de voz después de haber sido filtradas. En este trabajo de investigación se utilizó la técnica de contracción garrote para filtrar las señales de voz utilizadas.

Centrados en este enfoque, se utilizó la siguiente rutina para filtrar la señal de voz y obtener su PMRf.

```
xf=filtrarGarrote(SignalVoz,NivelDeDescomposicion);  
kf=abs(fft(xf));  
PMRf=max(kf)/mean(kf);
```

Debe señalarse que se construyó y validó la función denominada “filtrarGarrote” la cual se muestra en el apéndice A; dicha rutina recibe como parámetros de entrada la señal original y el nivel de descomposición wavelet deseado para producir la señal de voz filtrada de acuerdo a la metodología propuesta por Donoho y Johnstone. Dicha técnica consiste en estimar el nivel de ruido δ de la señal de entrada mediante el cálculo de la media absoluta de los coeficientes wavelet obtenidos en el primer nivel de descomposición ($cd_{j-1,k}$) dividida por 0.6745, esto es:

$$\delta = \frac{\sum_k \frac{|cd_{j-1,k}|}{n}}{0.6745} \quad (3.1)$$

Bajo este mismo enfoque, el umbral a seleccionar se determina según la teoría de Donoho, D (1995) con la ecuación 3.2, este umbral es denominado Umbral Universal, siendo N la longitud de datos de la señal de entrada:

$$thr = \delta \sqrt{2 \ln(N)} \quad (3.2)$$

La contracción garrote para el nivel de umbral thr fijado es dada según la ecuación 3.3, donde $cdn_{j-1,k}$ son los nuevos coeficientes de detalle umbralizados.

$$cdn_{j-1,k} = \begin{cases} 0, & \text{si } |cd_{j-1,k}| < thr \\ cd_{j-1,k} \left(1 - \left(\frac{thr}{|cd_{j-1,k}|} \right)^2 \right), & \text{si } |cd_{j-1,k}| \geq thr \end{cases} \quad (3.3)$$

En la figura 3.2 se muestran los resultados de la modificación de los coeficientes cuando se aplica contracción garrote.

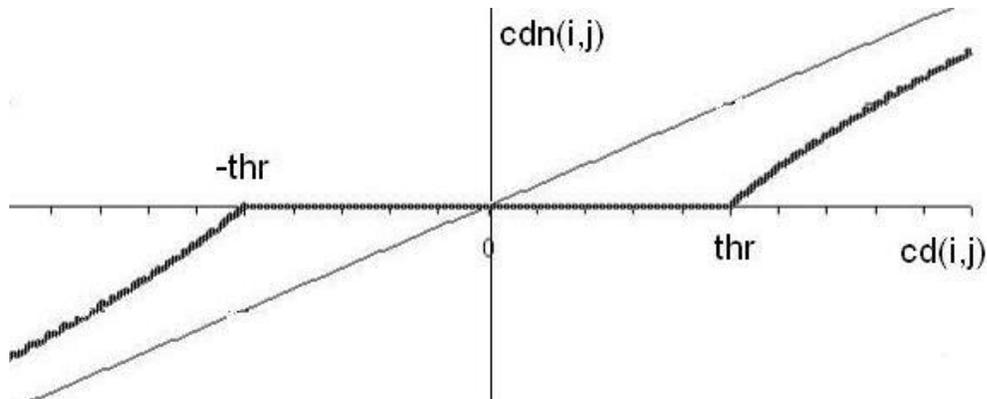


Figura 3.2: Contracción Garrote.
Fuente: Yusti F. (2015).

PMRseg: Es el PMR de la señal de voz aplicada por intervalos, en este trabajo de investigación se realizó la segmentación de la señal en 100 intervalos de igual longitud, para ello se utilizaron los siguientes comandos:

```

senal=SignalVoz;
x=linspace(1,length(senal),100);
fftSenal=abs(fft(senal));
for k=1:length(x)-1
    Nf=fftSenal(x(k):x(k+1));
    PMRseg(k) = max(Nf)/mean(Nf);
end

```

Ahora bien, se hace mención a que el cálculo del PMRseg se realizó en la señal original más no en la señal filtrada. En este punto, se obtuvo un vector que contiene el valor del PMR en cada segmento de la señal, a este vector denominado PMRseg se le calcularon los siguientes valores estadísticos para obtener nuevos parámetros que caractericen las señales de voz, dichos parámetros se resumen en la tabla 3.4:

Tabla 3.4: Parámetros estadísticos calculados al PMR de las señales de voz por segmentos y sus respectivos comandos.

Parámetro Estadístico del PMR	Sintaxis de Programación
Desviación Estándar	std(PMRseg)
Coficiente de Asimetría	skewness(PMRseg)
Kurtosis	kurtosis(PMRseg)
Varianza	var(PMRseg)
Coficiente de Variación (CVPMR)	std(PMRseg)/mean(PMRseg)

Fuente: Yusti F. (2015).

A título ilustrativo, se indica que en el cálculo de los parámetros estadísticos mostrados en la tabla 3.4, estos pueden ser normalizados por N o $N-1$ según sea el caso, en el apéndice B se muestran las fórmulas utilizadas para determinar dichos parámetros.

VWPT: Es el vector wavelet packet de la señal de voz, se trabajó desde el nivel 0 de descomposición hasta el nivel 3 de descomposición. De la figura 2.21 pueden obtenerse los siguientes vectores wavelet packet según sea el nivel de descomposición especificado, tal como se especifica en la tabla 3.5. Donde el nivel de descomposición 0 significa trabajar con la señal original.

Tabla 3.5: Estructura de los vectores wavelet packet de las señales de voz en los diferentes niveles de descomposición.

Nivel de descomposición	Vector wavelet packet
Nivel 0	<code>vwpt = señal de voz original</code>
Nivel 1	<code>vwpt = [A1 D1]</code>
Nivel 2	<code>vwpt = [AA2 DA2 AD2 DD2]</code>
Nivel 3	<code>vwpt = [AAA3 DAA3 ADA3 DDA3 AAD3 DAD3 ADD3 DDD3]</code>

Fuente: Yusti F. (2015).

En la perspectiva que aquí se adopta, para conseguir parámetros adicionales que sirvan para caracterizar las muestras de señales de voz, se procedió a calcular los cuatro momentos estadísticos al vector wavelet de la señal de voz junto con la varianza y el coeficiente de variación, los mismos son resumidos en la tabla 3.6.

Tabla 3.6: Parámetros estadísticos calculados al VWPT de las señales de voz y sus respectivos comandos.

Parámetro Estadístico del vwpt	Sintaxis de Programación
Valor Promedio	<code>mean (vwpt)</code>
Desviación Estándar	<code>std (vwpt)</code>
Coeficiente de Asimetría	<code>skewness (vwpt)</code>
Kurtosis	<code>kurtosis (vwpt)</code>
Varianza	<code>var (vwpt)</code>
Coeficiente de Variación (CVvwpt)	<code>std (vwpt) / mean (vwpt)</code>

Fuente: Yusti F. (2015).

Por último, es conveniente acotar que para caracterizar cada una de las señales de voz en estudio se construyeron vectores que contienen en sus componentes cada uno de los parámetros descritos anteriormente, para ello se implementó la función “CalcularVectorDeRasgosWPT” la cual puede revisarse en el apéndice C. Se hace necesario, por ello, construir matrices las cuales cada una de sus filas están formadas por los vectores de rasgos que arroja como salida la función implementada.

De este modo, si especificamos las siguientes variables X_i como se muestra en la tabla 3.7, el vector de rasgos queda formado por las siguientes componentes:

Tabla 3.7: Variables que contiene el vector de parámetros característicos.

X1: PMR	X2: PMRf	X3: std(PMRseg)	X4: skewness(PMRseg)
X5: kurtosis(PMRseg)	X6: var(PMRseg)	X7: CVPMR	X8: mean(vwpt)
X9: std(vwpt)	X10: skewness(vwpt)	X11: kurtosis(vwpt)	X12: var(vwpt):
X13: CVvwpt			
Vector de Rasgos = [X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13]			

Fuente: Yusti F. (2015).

Etapas III: Análisis Estadístico.

Una vez seleccionados y calculados los diferentes parámetros característicos, en este trabajo de investigación se tomaron en cuenta 13 parámetros, se construyen las matrices cuyas filas corresponden a cada una de las muestras de voz seleccionadas y

cada columna se corresponde con cada uno de los parámetros característicos calculados según el vector de rasgos mostrado en la tabla 3.7. Bajo esta misma perspectiva, se muestra en la tabla 3.8 la matriz 40x13 que contiene los parámetros característicos de las 20 primeras muestras de voz de la tabla 3.2 (correspondientes a voces sanas) en las filas desde la fila 1 hasta la fila 20 y en las filas desde la fila 21 hasta la fila 40 se encuentran los parámetros característicos de las primeras 20 muestras de voces con patologías mostradas en la tabla 3.3.

Tabla 3.8: Matriz “xtrainWPT” de parámetros característicos para 20 voces sanas y 20 voces con patologías, las muestras son tomadas de las tablas 3.2 y 3.3.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
1246,68	2227,00	50,68	9,50	93,03	2568,64	4,87	90,58	131,09	0,08	2,20	17184,11	1,45
1067,39	1920,34	43,71	9,75	96,43	1910,42	5,12	90,40	133,54	0,08	2,22	17832,37	1,48
1124,03	2012,06	50,88	9,59	94,29	2588,46	4,92	90,31	130,42	0,08	2,33	17008,29	1,44
1248,82	2228,37	56,94	9,49	92,95	3241,78	4,84	89,89	136,08	0,08	2,16	18518,60	1,51
1129,92	2024,55	37,02	9,65	95,00	1370,36	4,55	90,18	127,40	0,10	2,40	16229,73	1,41
1298,91	2321,65	55,20	9,51	93,24	3047,41	4,79	90,36	134,37	0,08	2,09	18054,41	1,49
1216,87	2175,86	46,32	9,65	94,99	2145,35	4,89	90,48	128,02	0,09	2,27	16388,12	1,41
1193,03	2140,01	41,96	9,56	93,78	1760,91	4,62	90,16	137,30	0,09	2,18	18850,50	1,52
1134,82	2028,52	54,13	9,56	93,88	2929,84	4,73	90,22	131,24	0,08	2,29	17223,55	1,45
1113,91	1993,50	51,61	9,67	95,36	2663,21	4,98	90,06	127,88	0,09	2,33	16354,49	1,42
1136,24	2035,57	47,78	9,55	93,61	2282,52	4,92	90,35	122,08	0,10	2,49	14903,12	1,35
1169,23	2094,29	44,14	9,41	91,67	1948,34	4,59	90,11	133,64	0,09	2,25	17859,38	1,48
1134,11	2035,35	55,40	9,69	95,59	3068,69	5,21	90,19	134,07	0,08	2,20	17974,21	1,49
1149,66	2057,96	58,05	9,69	95,56	3370,15	5,40	90,03	130,58	0,08	2,23	17051,40	1,45
1129,92	2028,69	49,68	9,55	93,76	2468,50	4,91	90,00	132,67	0,09	2,29	17601,40	1,47
1052,68	1888,68	35,74	9,66	95,11	1277,57	4,50	90,55	131,14	0,09	2,27	17197,44	1,45
881,04	1591,90	21,94	9,73	96,06	481,58	3,63	90,52	127,19	0,09	2,33	16176,62	1,41
1196,82	2138,92	67,38	9,70	95,78	4540,12	5,59	90,19	138,15	0,08	2,17	19086,78	1,53
1096,93	1963,68	47,21	9,51	93,17	2229,11	4,70	90,01	124,50	0,09	2,48	15501,11	1,38
1101,87	1973,09	47,29	9,67	95,33	2236,49	4,94	89,90	134,61	0,09	2,26	18119,73	1,50
545,14	965,88	19,56	8,60	79,95	382,77	3,04	89,75	135,29	0,09	2,09	18302,58	1,51
593,56	1050,64	18,96	8,84	83,66	359,48	3,18	90,32	129,71	0,11	2,12	16824,40	1,44
390,15	707,38	9,57	9,73	96,06	91,62	2,36	89,34	134,82	0,08	2,28	18175,88	1,51
407,61	739,70	10,09	9,76	96,45	101,85	2,46	89,51	134,71	0,07	2,27	18146,69	1,50
441,74	785,53	16,26	8,83	83,30	264,53	2,89	90,09	126,57	0,07	2,40	16021,20	1,41
423,25	753,53	12,85	8,79	82,66	165,12	2,60	89,49	120,73	0,11	2,68	14574,65	1,35

Tabla 3.8: (Continuación) Matriz de parámetros característicos...

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
544,41	965,40	21,25	9,13	87,94	451,47	3,14	90,09	145,23	0,08	1,86	21091,69	1,61
416,04	742,90	17,83	8,56	79,28	317,83	2,96	90,64	116,03	0,10	2,72	13463,79	1,28
371,61	661,50	16,25	9,66	95,19	263,98	3,26	90,17	130,61	0,04	2,37	17059,77	1,45
526,43	936,92	18,77	7,89	69,01	352,13	3,06	90,12	133,80	0,09	2,17	17902,84	1,48
417,52	739,26	17,50	9,03	86,80	306,34	2,64	89,98	125,08	0,07	2,48	15644,19	1,39
497,64	888,86	18,48	8,13	73,46	341,34	2,73	89,78	125,47	0,09	2,31	15742,03	1,40
399,67	709,81	16,31	8,94	85,40	266,06	2,67	90,35	115,96	0,08	2,74	13447,34	1,28
514,71	917,52	15,65	9,03	86,30	244,81	2,93	90,08	131,70	0,09	2,16	17345,05	1,46
414,91	744,15	13,58	9,09	86,93	184,32	2,85	90,51	130,69	0,07	2,34	17080,40	1,44
441,21	780,09	17,86	8,80	83,58	319,00	2,81	90,00	122,66	0,11	2,49	15044,57	1,36
376,81	677,30	13,85	9,24	89,31	191,93	2,78	89,67	115,88	0,09	2,82	13428,03	1,29
516,56	922,02	20,02	9,02	86,12	400,92	3,22	90,71	147,58	0,08	1,89	21779,83	1,63
323,29	574,84	10,82	9,52	93,28	117,12	2,47	90,55	132,72	0,07	2,21	17614,32	1,47
553,95	979,21	20,63	8,54	79,38	425,77	3,00	90,68	131,21	0,10	2,06	17215,69	1,45

Fuente: Yusti F. (2015).

Sin duda, la dimensión inicial es 13, podrían utilizarse estas 13 variables como parámetros de entrada para entrenar la máquina de vectores de soporte, sin embargo, la idea es estudiar si es posible describir este conjunto de datos utilizando un número menor de dimensiones aprovechando las interrelaciones entre las variables.

En este sentido, se planteó la utilización del método de las componentes principales como herramienta para la reducción de la dimensionalidad de las variables, para lo cual se procedió a cargar la matriz de la tabla 3.8 en la variable “xtrainWPT”.

De igual modo, una vez cargada la matriz xtrainWPT se procede a determinar la matriz de correlación de las variables involucradas en la matriz xtrainWPT, esta matriz de correlación es una matriz simétrica según lo indica la ecuación 2.21. Por

otro lado, utilizando las componentes principales se puede estimar los autovalores, autovectores y el porcentaje explicado de cada una de las nuevas variables cuando estas son rotadas a la nueva base vectorial. Para este fin se hace uso de las siguientes instrucciones, donde el comando “corrcoef” determina la matriz de correlación “R” de las variables de la matriz xtrainWPT:

```
R=corrcoef(xtrainWPT);
[cpwpt, lambdawpt, explicacionwpt]=pcacov(R);
```

En lo que respecta a la función pcacov, se observa que “cpwpt” es la matriz 13x13 de componentes principales, o sea, los autovectores de la matriz R, dicha matriz se muestra en la tabla 3.9. Así mismo “lambdawpt” es la matriz que contiene los autovalores de R y “explicacionwpt” representa el porcentaje explicado de la varianza ofrecida por cada una de las componentes principales.

Tabla 3.9: Matriz cpwpt, componentes principales de la matriz xtrainWPT.

0,362	-0,183	0,143	-0,038	0,008	-0,479	0,078	-0,263	-0,046	0,026	-0,086	0,702	-0,073
0,362	-0,183	0,139	-0,039	0,013	-0,484	0,089	-0,259	-0,101	0,039	0,039	-0,698	0,071
0,360	-0,182	0,127	-0,063	-0,268	0,238	-0,069	-0,030	0,761	-0,202	0,253	-0,011	0,015
0,285	-0,100	-0,481	0,059	0,392	0,089	-0,065	0,023	-0,193	0,003	0,681	0,073	0,030
0,283	-0,102	-0,483	0,062	0,398	0,104	-0,097	-0,025	0,239	-0,013	-0,655	-0,069	-0,028
0,354	-0,170	0,084	-0,101	-0,328	0,570	-0,209	-0,301	-0,481	0,109	-0,136	0,002	-0,007
0,365	-0,184	0,103	0,009	-0,124	-0,024	0,190	0,854	-0,173	0,054	-0,093	0,002	-0,007
0,080	-0,007	0,252	0,939	0,095	0,126	0,106	-0,082	-0,014	-0,045	0,003	-0,005	-0,048
0,220	0,447	0,031	-0,050	0,036	0,064	0,244	-0,043	-0,090	-0,397	-0,067	0,069	0,712
-0,033	-0,186	0,593	-0,268	0,687	0,258	0,047	0,006	0,008	-0,019	0,006	0,000	0,000
-0,201	-0,424	-0,209	-0,054	-0,089	0,180	0,813	-0,167	-0,007	-0,058	-0,013	0,004	-0,001
0,215	0,450	0,033	-0,046	0,040	0,116	0,315	-0,064	0,189	0,769	0,048	0,004	-0,021
0,215	0,448	0,013	-0,116	0,029	0,053	0,233	-0,039	-0,098	-0,433	0,006	-0,070	-0,691

Fuente: Yusti F. (2015).

Ahora bien, los autovalores presentes en la variable *lambdawpt* son los siguientes:

| 6.29 | 3.43 | 1.61 | 0.96 | 0.49 | 0.10 | 0.07 | 0.02 | 0.0028 | 0.0005 | 0.0002 | 0 | 0 |

Según el criterio de Kayser deben tomarse las variables cuyos autovalores arrojen un valor mayor que “1” lo que significa que deberían tomarse sólo las tres primeras variables transformadas; en su defecto puede utilizarse el criterio de Jolliffe quien sugiere que se tomen las variables cuyos autovalores arrojen valores mayores a 0.7, por lo que deberían tomarse las primeras 4 variables transformadas.

De allí la importancia que tienen los autovalores de la matriz de correlación, estas referencias deben tenerse presente en el momento de decidir qué tan eficiente se desea el modelo de predicción con las variables transformadas, en este trabajo de investigación se analizó la varianza explicada por cada componente principal, dicho porcentaje de varianza explicada es arrojada en la variable “*explicacionwpt*”:

| 48.39 | 74.81 | 87.21 | 94.62 | 98.4 | 99.2 | 99.76 | 99.97 | 99.99 | 99.99 | 99.99 | 99.99 | 100 |

De este resultado se puede observar que utilizando el criterio de Kayser, las variables seleccionadas explican el 87.21% de la varianza de los datos, ahora bien, si se asume el criterio de Jolliffe, las 4 primeras variables transformadas explicarían el 94.62% de la varianza total, puede concluirse que efectivamente se puede hacer una reducción en la dimensionalidad de las variables y si se quiere ser un tanto más preciso se pueden escoger las 5 primeras variables transformadas ya que estas recogen el 98.4% de la varianza total, por lo tanto a la entrada de la máquina de vectores de soporte se pueden llevar sólo 5 variables y no 13 como se tenían inicialmente.

En virtud de ello, se procede a la estandarización de la matriz de datos a los fines de poder rotarlos, se estandariza variable por variable en la matriz xtrainWPT, este proceso significa determinar la media y la desviación estándar de cada columna de la matriz xtrainWPT, a cada término de la columna se le resta la media y se divide entre la desviación estándar de su columna correspondiente, para ello se utilizó el comando “zscore” de la siguiente forma:

`xtrainWPTstd=zscore(xtrainWPT)`

Este comando realiza la estandarización en las columnas de una matriz “x” bajo la expresión $(x - \text{mean}(x)) / \text{std}(x)$, permite obtener datos con media cero y desviación estándar uno, así se tiene ahora una matriz 40x13 denominada “xtrainWPTstd” la cual es la matriz xtrainWPT estandarizada, la misma se muestra en la tabla 3.10.

Tabla 3.10: Matriz de variables estandarizadas “xtrainWPTstd”.

X1std	X2std	X3std	X4std	X5std	X6std	X7std	X8std	X9std	X10std	X11std	X12std	X13std
1,258	1,234	1,046	0,463	0,469	0,989	0,969	1,263	0,115	-0,052	-0,475	0,090	0,026
0,755	0,815	0,651	0,990	0,964	0,458	1,203	0,721	0,476	-0,169	-0,366	0,457	0,426
0,914	0,937	1,057	0,656	0,652	1,005	1,022	0,473	0,016	-0,565	0,163	-0,010	-0,018
1,264	1,255	1,401	0,448	0,458	1,531	0,943	-0,819	0,852	-0,469	-0,680	0,846	0,915
0,931	0,908	0,272	0,766	0,755	0,023	0,667	0,063	-0,430	0,828	0,517	-0,451	-0,435
1,405	1,382	1,303	0,490	0,499	1,374	0,898	0,614	0,598	-0,254	-1,032	0,583	0,556
1,175	1,152	0,799	0,764	0,754	0,647	0,990	0,979	-0,338	0,656	-0,156	-0,361	-0,407
1,108	1,092	0,552	0,577	0,578	0,338	0,730	-0,007	1,031	0,071	-0,594	1,035	1,034
0,944	0,962	1,242	0,589	0,592	1,280	0,839	0,201	0,137	-0,290	-0,039	0,112	0,123
0,886	0,912	1,099	0,820	0,808	1,065	1,079	-0,303	-0,358	0,041	0,133	-0,380	-0,338
0,948	0,933	0,882	0,557	0,554	0,758	1,022	0,587	-1,214	0,927	0,957	-1,203	-1,256
1,041	1,032	0,675	0,261	0,270	0,489	0,703	-0,160	0,491	0,399	-0,251	0,473	0,504
0,942	0,984	1,314	0,857	0,841	1,392	1,297	0,086	0,554	-0,250	-0,481	0,538	0,550
0,986	1,012	1,464	0,852	0,837	1,635	1,468	-0,374	0,040	-0,095	-0,315	0,015	0,067
0,931	0,935	0,990	0,573	0,575	0,908	1,012	-0,476	0,348	0,161	-0,063	0,326	0,383
0,714	0,705	0,199	0,783	0,772	-0,052	0,618	1,186	0,122	0,368	-0,117	0,097	0,039
0,232	0,261	-0,583	0,931	0,910	-0,693	-0,202	1,105	-0,461	0,327	0,179	-0,481	-0,538
1,118	1,118	1,993	0,887	0,869	2,577	1,652	0,084	1,157	-0,123	-0,621	1,169	1,154

Tabla 3.10: (Continuación) Matriz de variables estandarizadas...

X1std	X2std	X3std	X4std	X5std	X6std	X7std	X8std	X9std	X10std	X11std	X12std	X13std
0,838	0,834	0,850	0,486	0,488	0,715	0,813	-0,459	-0,856	0,302	0,888	-0,864	-0,828
0,852	0,849	0,854	0,817	0,804	0,721	1,033	-0,782	0,634	0,553	-0,163	0,620	0,693
-0,711	-0,750	-0,718	-1,427	-1,438	-0,773	-0,760	-1,244	0,734	0,047	-1,002	0,724	0,828
-0,575	-0,631	-0,752	-0,925	-0,897	-0,792	-0,623	0,487	-0,089	1,863	-0,880	-0,114	-0,123
-1,146	-1,129	-1,285	0,932	0,910	-1,008	-1,405	-2,473	0,665	-0,670	-0,102	0,652	0,849
-1,097	-1,075	-1,255	0,994	0,968	-0,999	-1,310	-1,950	0,649	-1,213	-0,122	0,636	0,794
-1,002	-0,995	-0,905	-0,949	-0,950	-0,868	-0,906	-0,220	-0,551	-1,430	0,514	-0,569	-0,537
-1,053	-1,073	-1,099	-1,020	-1,043	-0,948	-1,180	-2,027	-1,414	1,577	1,892	-1,389	-1,284
-0,713	-0,719	-0,623	-0,306	-0,274	-0,718	-0,667	-0,200	2,201	-0,229	-2,138	2,305	2,223
-1,074	-1,076	-0,817	-1,511	-1,537	-0,825	-0,834	1,460	-2,106	0,753	2,045	-2,019	-2,203
-1,198	-1,137	-0,906	0,799	0,784	-0,869	-0,556	0,046	0,045	-3,842	0,329	0,019	0,042
-0,764	-0,799	-0,763	-2,909	-3,034	-0,798	-0,740	-0,115	0,515	0,734	-0,602	0,497	0,525
-1,070	-1,063	-0,835	-0,529	-0,439	-0,835	-1,143	-0,526	-0,772	-1,171	0,897	-0,783	-0,738
-0,845	-0,858	-0,780	-2,400	-2,384	-0,806	-1,058	-1,147	-0,714	0,639	0,077	-0,728	-0,638
-1,120	-1,108	-0,903	-0,712	-0,644	-0,867	-1,108	0,585	-2,116	-0,057	2,182	-2,028	-2,159
-0,797	-0,820	-0,940	-0,523	-0,512	-0,884	-0,860	-0,228	0,205	0,334	-0,675	0,181	0,222
-1,077	-1,062	-1,058	-0,408	-0,420	-0,933	-0,942	1,069	0,056	-1,095	0,223	0,031	-0,019
-1,003	-1,013	-0,815	-0,997	-0,909	-0,824	-0,974	-0,469	-1,129	1,540	0,932	-1,123	-1,101
-1,184	-1,180	-1,042	-0,083	-0,074	-0,927	-1,004	-1,482	-2,128	0,368	2,553	-2,039	-2,041
-0,791	-0,791	-0,692	-0,550	-0,539	-0,758	-0,594	1,672	2,547	-0,548	-1,994	2,695	2,422
-1,334	-1,308	-1,214	0,499	0,505	-0,987	-1,296	1,199	0,355	-1,333	-0,455	0,334	0,271
-0,687	-0,726	-0,657	-1,541	-1,522	-0,738	-0,797	1,580	0,133	1,365	-1,159	0,108	0,021

Fuente: Yusti F. (2015).

En este sentido, se procede a calcular las variables transformadas mediante la rotación de sus ejes a los nuevos ejes definidos por las componentes principales; es por ello que se ejecuta la siguiente instrucción:

$$Z_{train} = x_{train} WPT_{std} * cpwpt$$

Se produce la matriz de variables transformadas Z_{train} 40x13 mediante el producto matricial antes indicado, dicha matriz se muestra en la tabla 3.11; se observa

además, que cada variable Z_i de la matriz Z_{train} es una combinación lineal de las variables originales dadas en la matriz $x_{trainWPTstd}$ según la expresión:

$$Z_{ij} = Xstd_{i1}cwpt_{1j} + \dots + Xstd_{ik}cwpt_{kj} = \sum_{s=1}^k Xstd_{is}cwpt_{sj} \quad (3.4)$$

El análisis precedente sustenta que cada columna de variables absorbe un porcentaje de la varianza total de la variabilidad de los datos, dicha fracción de porcentaje se encuentra como se mencionó anteriormente en la variable “*explicacionwpt*” la cual se utilizó para seleccionar cuántas de las variables transformadas “ Z_i ” se pueden utilizar para explicar en un porcentaje razonable el total de la variabilidad de los datos.

Tabla 3.11: Matriz de variables transformadas Z_{train} .

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13
2,50	-0,78	0,62	1,02	-0,19	-0,25	-0,15	-0,19	-0,02	0,01	-0,01	0,00	0,00
2,37	-0,10	-0,29	0,63	0,35	-0,07	0,19	0,41	-0,01	-0,01	0,00	-0,01	0,00
2,16	-0,99	-0,30	0,44	-0,56	-0,01	0,11	-0,01	0,03	-0,02	0,01	0,00	0,00
3,21	0,31	0,05	-1,07	-0,86	-0,09	-0,08	-0,30	0,01	0,01	0,00	0,00	0,00
1,04	-1,63	0,00	-0,09	0,94	-0,48	0,25	0,04	-0,03	0,00	-0,01	0,01	0,00
3,22	0,01	0,56	0,32	-0,45	-0,22	-0,38	-0,38	-0,01	0,03	0,00	0,00	0,00
2,00	-1,58	0,50	0,73	0,59	-0,27	-0,21	0,02	-0,02	0,02	0,00	0,01	0,00
2,51	0,81	0,17	-0,29	0,34	-0,50	0,46	-0,13	0,00	0,00	-0,02	0,00	0,00
2,34	-0,83	-0,05	0,04	-0,53	0,18	-0,06	-0,23	0,07	-0,03	0,03	0,00	0,00
1,98	-1,61	-0,30	-0,37	-0,17	0,06	-0,29	0,16	0,04	0,00	0,02	0,00	0,00
0,99	-3,17	0,42	0,38	0,33	0,01	-0,04	0,10	0,01	0,02	0,01	0,00	0,00
1,92	-0,08	0,52	-0,48	0,14	-0,38	0,30	-0,12	0,02	-0,02	-0,01	0,00	0,00
3,07	-0,25	-0,12	-0,12	-0,34	0,31	-0,08	0,15	0,00	-0,01	0,01	-0,01	0,00
2,90	-1,14	-0,14	-0,54	-0,49	0,34	-0,41	0,28	-0,04	0,02	0,00	0,00	0,00
2,25	-0,50	0,03	-0,71	-0,11	-0,01	0,18	0,07	0,03	-0,02	0,00	-0,01	0,00
1,40	-0,47	0,09	1,04	0,90	-0,29	0,23	0,07	-0,01	-0,02	0,00	0,00	0,00
-0,11	-0,82	-0,57	1,25	1,40	-0,44	-0,07	-0,12	-0,09	0,02	0,00	-0,01	0,00
4,43	0,16	0,28	-0,45	-0,77	1,14	0,10	-0,08	-0,13	0,01	-0,03	0,00	0,00
0,96	-2,40	-0,17	-0,51	-0,16	-0,14	0,02	0,03	0,06	0,00	0,01	0,00	0,00
2,39	-0,10	-0,05	-1,10	0,45	0,09	0,32	0,19	0,04	-0,03	0,01	0,00	0,00

Tabla 3.11: (Continuación) Matriz de variables transformadas Ztrain.

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13
-1,55	2,40	0,92	-1,29	-0,53	-0,33	-0,17	0,14	0,01	0,01	0,00	0,00	0,00
-1,64	0,66	1,89	0,09	1,20	0,13	-0,51	0,17	-0,02	0,00	-0,02	0,00	0,00
-1,31	2,05	-2,55	-1,93	0,94	0,09	-0,12	-0,15	-0,03	-0,01	0,02	0,00	0,00
-1,15	2,06	-2,77	-1,28	0,64	-0,02	-0,11	-0,14	-0,05	0,00	0,02	0,00	0,00
-2,66	0,34	-0,70	0,36	-1,24	-0,34	-0,05	0,04	-0,03	0,00	-0,01	0,00	0,00
-3,99	-1,74	0,28	-2,03	0,49	0,22	0,36	-0,08	-0,06	0,00	0,00	0,01	0,01
0,47	4,65	0,30	-0,36	0,49	0,15	0,00	0,08	0,12	0,04	0,01	0,00	0,00
-4,22	-2,71	1,14	1,54	-0,38	0,36	0,33	-0,03	-0,01	0,01	0,03	0,00	-0,01
-1,17	1,31	-3,68	1,38	-1,46	-0,31	-0,07	0,35	-0,01	0,00	0,00	0,01	0,00
-2,65	2,10	2,98	-0,56	-1,21	-0,27	0,37	0,08	-0,11	-0,02	0,03	0,00	0,00
-2,74	-0,19	-1,20	0,08	-0,78	-0,09	-0,07	-0,14	0,10	-0,03	-0,02	0,00	0,00
-3,50	0,19	1,81	-1,21	-1,05	-0,40	-0,31	-0,01	0,00	0,02	-0,01	-0,01	0,00
-3,99	-2,68	-0,47	1,03	-0,29	0,24	0,13	-0,18	0,06	0,03	-0,01	0,00	0,00
-1,60	1,37	0,29	-0,17	0,51	-0,06	-0,36	0,08	-0,04	0,00	-0,01	0,00	0,00
-1,97	1,13	-0,64	1,47	-0,31	0,09	0,24	-0,07	-0,08	-0,03	-0,01	0,00	0,00
-3,21	-1,15	0,88	-0,58	0,65	0,39	-0,08	0,02	0,08	-0,04	-0,03	0,00	0,00
-3,96	-2,94	-1,42	-0,96	0,29	0,26	0,19	0,07	0,01	0,05	0,00	0,00	0,01
0,60	5,13	0,80	1,40	0,29	0,39	0,61	-0,03	0,06	0,04	0,00	0,00	0,01
-1,50	1,88	-1,60	1,77	0,45	0,36	-0,31	-0,15	-0,02	-0,03	0,01	0,00	0,00
-1,79	1,30	2,51	1,14	0,49	0,18	-0,46	-0,01	0,03	-0,03	0,01	0,00	0,00

Fuente: Yusti F. (2015).

Etapas IV: Clasificación.

Se puede resumir a continuación, que luego de obtener un número reducido de parámetros característicos de las señales de voz, analizadas en la etapa anterior, estos mismos serán utilizados como parámetros de entrada a la máquina de vectores de soporte para realizar su clasificación entre señal de voz sana o señal de voz con patología.

Se plantea entonces el problema de la reducción del número de parámetros de entrada para entrenar la máquina de vectores de soporte, en virtud de ello, se procedió a realizar los siguientes experimentos, el primero de ellos consistió en entrenar la máquina con las variables originales, es decir, con todas las variables desde X1 hasta X13, luego se procedió a entrenar la máquina con todas las variables estandarizadas desde X1std hasta X13std, después se entrenó la máquina con todas las variables transformadas “Z”, es decir desde Z1 hasta Z13 y finalmente se realizó el entrenamiento reduciendo el número de variables transformadas “Z” considerando el porcentaje de varianza explicada presente en la variable “*explicacionwpt*”. Cabe señalar que después de cada entrenamiento, se realizó un entrenamiento supervisado introduciéndole a la máquina de vectores de soporte la misma matriz de entrenamiento para verificar si la máquina aprendió a clasificar; en segundo lugar se procedió a suministrarle a la máquina, los parámetros característicos de un conjunto de 109 señales distintas a las del entrenamiento (las cuales fueron mostradas en las tablas 3.2 y 3.3) divididas en 30 señales sanas y 79 señales con patologías, todo ello con la intención de dar validación al aprendizaje de la máquina.

En función de lo anteriormente expuesto se describe a continuación cada uno de los experimentos realizados en el diseño del clasificador de voces mediante la SVM. Este mismo consta básicamente de los siguientes pasos:

Generación de las señales de entrenamiento: para el diseño de cada máquina se utilizaron como entrenamiento, las matrices $x_{trainWPT}$, $x_{trainWPTstd}$ y Z_{train} , de la matriz de entrenamiento Z_{train} se fueron eliminando columnas con el fin de ir reduciendo la dimensión del conjunto de parámetros, es decir, se diseñaron como mínimo 4 máquinas de vectores de soporte para ir comparando el funcionamiento de las mismas.

Entrenamiento propio de la SVM: para este fin se utilizó la librería “libsvm-mat-2.86” disponible en <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, la cual es una librería de funciones cuya sintaxis puede revisarse a plenitud en el apéndice D, en este trabajo de investigación se utilizó de la siguiente forma:

```
model= svmtrain(Vector de grupo,Matriz de Entrenamiento, 'Opciones')
```

El vector de grupo es un vector columna (vector de etiquetas) que contiene “1” o “-1” en correspondencia con la matriz de entrenamiento para indicar de que clase es cada fila de la matriz, se estableció la etiqueta “1” para señales sanas y “-1” para señales con patologías.

En lo que respecta a la entrada “*opciones*”, allí se especifica entre otras cosas, el tipo de SVM, el tipo de función kernel a utilizar, en este caso se utilizó la función “rbf” por lo que hubo que suministrar el valor de σ (ecuación 2.26) y el valor de la constante de penalización C (missclassification tolerance parameter).

Supervisión del aprendizaje: en este punto, se procede a utilizar el comando “svmpredict” para realizar la clasificación de las señales de voz, se le denomina supervisado porque se van a clasificar las mismas señales del entrenamiento para verificar el funcionamiento de la máquina creada en el paso anterior, en caso de no obtener una precisión aceptable (lo más cercana posible al 100%) se modifican los valores de σ Y C del paso anterior de manera empírica, este proceso es conocido como entrenamiento supervisado.

```
[EtiquetaPredecida, precisión, valor estimado]= svmpredict(Vector de  
grupo de prueba,Matriz de prueba,model)
```

Clasificación de las señales de voz: de este modo se procede a crear las matrices que contienen los parámetros característicos de las señales a clasificar, se crearon las matrices SxtrainWPT la cual contiene las 30 últimas muestras de señales de la tabla 3.2 y las 79 últimas muestras de señales de la tabla 3.3, lo que significa que es una matriz de 109x13. En este mismo orden de ideas se estandarizó la matriz SxtrainWPT para crear la matriz SxtrainWPTstd y finalmente se crearon las variables transformadas a partir de la matriz SxtrainWPTstd para crear la matriz “Ztrains” (se diferencia de Ztrain porque termina en s y además es de 109x13). Visto de esta forma, ya que fueron creadas las matrices con los parámetros de las señales a clasificar, se utilizó la siguiente secuencia de comandos obteniéndose resultados, los cuales se muestran a continuación.

Se realizó el entrenamiento con todas las variables originales, observándose en la tabla 3.12 que se obtuvo un 100% de precisión en el entrenamiento supervisado y un 72.47% de precisión en la clasificación de los datos, sólo fue posible acertar 79 de las 109 señales introducidas.

Tabla 3.12: Clasificación con las variables originales.

Ejecución:	
<pre> grupo=[ones(20,1);-1*ones(20,1)]; Sgrupo=[ones(30,1);-1*ones(79,1)]; model= svmtrain(grupo,xtrainWPT,'-s 0 -t 2 -g 0.1 -c 0.01'); [salida, pre, ppp]= svmpredict(grupo,xtrainWPT,model); [salida, pre, ppp]= svmpredict(Sgrupo,SxtrainWPT,model); </pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 72.4771% (79/109) (classification)

Fuente: Yusti F. (2015).

Se repite el experimento anterior pero con las variables estandarizadas, se observa en la tabla 3.13 una excelente mejoría en la clasificación de los datos, 100% de precisión acertando 109 de las 109 señales estudiadas.

Tabla 3.13: Clasificación con todas las variables originales estandarizadas.

Ejecución:	
<pre> grupo=[ones(20,1);-1*ones(20,1)]; Sgrupo=[ones(30,1);-1*ones(79,1)]; model= svmtrain(grupo,xtrainWPTstd(:,2:4),'-s 0 -t 2 -g 1 -c 1'); [salida, pre, ppp]= svmpredict(grupo,xtrainWPTstd,model); [salida, pre, ppp]= svmpredict(Sgrupo,SxtrainWPTstd,model); </pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 100% (109/109) (classification)

Fuente: Yusti F. (2015).

Se repete el experimento pero ahora se utilizarán las variables transformadas Zi, se observa en la tabla 3.14 los resultados obtenidos.

Tabla 3.14: Clasificación con todas las variables Z.

Ejecución:	
<pre> grupo=[ones(20,1);-1*ones(20,1)]; Sgrupo=[ones(30,1);-1*ones(79,1)]; model= svmtrain(grupo,Ztrain,'-s 0 -t 2 -g 0.1 -c 0.01'); [salida, pre, ppp]= svmpredict(grupo,Ztrain,model); [salida, pre, ppp]= svmpredict(Sgrupo,Ztrains,model); </pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 99.0826% (108/109) (classification)

Fuente: Yusti F. (2015).

En atención a la problemática de reducir la dimensionalidad del conjunto de datos, se determinó la varianza explicada por cada una de las variables transformadas Zi con la siguiente secuencia de instrucciones mostradas en la tabla 3.15:

Tabla 3.15: Determinación del porcentaje de varianza explicada.

<pre>R=corrcoef(xtrainWPT); [cpwpt, lambdawpt, explicacionwpt]=pcacov(R); explicacionwpt=explicacionwpt'; explicacionwptacumulada=cumsum(explicacionwpt)</pre>												
48.39	74.81	87.21	94.62	98.4	99.2	99.76	99.97	99.99	99.99	99.99	99.99	100

Fuente: Yusti F. (2015).

Se realizaron los ensayos reduciendo la dimensionalidad del conjunto de datos considerando el porcentaje de varianza explicada mostrado en la tabla 3.15, se tomaron las 5 primeras variables transformadas (recogen el 98.4% de varianza explicada), luego las 4 primeras (recogen el 94.62% de la varianza total de los datos) y finalmente las 3 primeras las cuales acumulan el 87.21% del total de variabilidad de los datos, los resultados se muestran en las tablas 3.16, 3.17 y 3.18.

Tabla 3.16: Clasificación con las variables Z reducidas, las primeras 5.

Ejecución:	
<pre>model= svmtrain(grupo, Ztrain(:,1:5), '-s 0 -t 2 -g 0.1 -c 0.01'); [salida, pre, ppp]= svmpredict(grupo, Ztrain(:,1:5), model); [salida, pre, ppp]= svmpredict(Sgrupo, Ztrains(:,1:5), model);</pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 99.0826% (108/109) (classification)

Fuente: Yusti F. (2015).

Tabla 3.17: Clasificación con las variables Z reducidas, las primeras 4.

Ejecución:	
<pre> model= svmtrain(grupo,Ztrain(:,1:4),'-s 0 -t 2 -g 0.1 -c 0.01'); [salida, pre,ppp]= svmpredict(grupo,Ztrain(:,1:4),model); [salida, pre,ppp]= svmpredict(Sgrupo,Ztrains(:,1:4),model); </pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 99.0826% (108/109) (classification)

Fuente: Yusti F. (2015).

Lo señalado anteriormente cobra importancia en vista que se ha atacado la problemática de reducción de la dimensionalidad del conjunto de datos que se introducen a la máquina de vectores de soporte, se observa en la tabla 3.18 que aún cuando se redujo la dimensión de parámetros de entrada de 13 a sólo 3, los resultados obtenidos en la clasificación mantienen una precisión cercana al 100%, de 109 señales estudiadas se acertaron 108 de ellas.

Tabla 3.18: Clasificación con las variables Z reducidas, las primeras 3.

Ejecución:	
<pre> model= svmtrain(grupo,Ztrain(:,1:3),'-s 0 -t 2 -g 0.1 -c 0.01'); [salida, pre,ppp]= svmpredict(grupo,Ztrain(:,1:3),model); [salida, pre,ppp]= svmpredict(Sgrupo,Ztrains(:,1:3),model); </pre>	
Resultado:	
Señales de Entrenamiento	Accuracy = 100% (40/40) (classification)
Señales Desconocidas	Accuracy = 99.0826% (108/109) (classification)

Fuente: Yusti F. (2015).

Se hace necesario por ello, detectar la señal que es clasificada erróneamente por la máquina de vectores de soporte diseñada, para ello comparamos la variable “*salida*” (esta contiene la etiqueta predicha) con la variable “*Sgrupo*”, se hace la resta entre los dos vectores, entonces los resultados que den distinto de cero serán las señales

mal clasificadas, en la tabla 3.19 se muestran los resultados de la clasificación hecha en la tabla 3.18.

Tabla 3.19: Comparación de la clasificación.

Señal N°	<i>Sgrupo</i>	<i>salida</i>	<i>Sgrupo - salida</i>
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0
10	1	1	0
11	1	1	0
12	1	1	0
13	1	1	0
14	1	1	0
15	1	1	0
16	1	1	0
17	1	1	0
18	1	1	0
19	1	1	0
20	1	1	0
21	1	1	0
22	1	1	0
23	1	1	0
24	1	1	0
25	1	1	0
26	1	1	0
27	1	1	0
28	1	1	0
29	1	1	0
30	1	1	0
31	-1	-1	0
32	-1	-1	0
33	-1	-1	0

Tabla 3.19: (Continuación) Comparación...

Señal N°	<i>Sgrupo</i>	<i>salida</i>	<i>Sgrupo - salida</i>
34	-1	-1	0
35	-1	-1	0
36	-1	-1	0
37	-1	-1	0
38	-1	-1	0
39	-1	-1	0
40	-1	-1	0
41	-1	-1	0
42	-1	-1	0
43	-1	-1	0
44	-1	-1	0
45	-1	-1	0
46	-1	-1	0
47	-1	-1	0
48	-1	-1	0
49	-1	-1	0
50	-1	-1	0
51	-1	-1	0
52	-1	-1	0
53	-1	-1	0
54	-1	-1	0
55	-1	-1	0
56	-1	-1	0
57	-1	-1	0
58	-1	-1	0
59	-1	-1	0
60	-1	-1	0
61	-1	-1	0
62	-1	-1	0
63	-1	-1	0
64	-1	-1	0
65	-1	-1	0
66	-1	-1	0
67	-1	-1	0
68	-1	-1	0
69	-1	-1	0
70	-1	-1	0

Tabla 3.19: (Continuación) Comparación...

Señal N°	<i>Sgrupo</i>	<i>salida</i>	<i>Sgrupo - salida</i>
71	-1	-1	0
72	-1	-1	0
73	-1	-1	0
74	-1	1	-2
75	-1	-1	0
76	-1	-1	0
77	-1	-1	0
78	-1	-1	0
79	-1	-1	0
80	-1	-1	0
81	-1	-1	0
82	-1	-1	0
83	-1	-1	0
84	-1	-1	0
85	-1	-1	0
86	-1	-1	0
87	-1	-1	0
88	-1	-1	0
89	-1	-1	0
90	-1	-1	0
91	-1	-1	0
92	-1	-1	0
93	-1	-1	0
94	-1	-1	0
95	-1	-1	0
96	-1	-1	0
97	-1	-1	0
98	-1	-1	0
99	-1	-1	0
100	-1	-1	0
101	-1	-1	0
102	-1	-1	0
103	-1	-1	0
104	-1	-1	0
105	-1	-1	0
106	-1	-1	0
107	-1	-1	0

Tabla 3.19: (Continuación) Comparación...

Señal N°	<i>Sgrupo</i>	<i>salida</i>	<i>Sgrupo - salida</i>
108	-1	-1	0
109	-1	-1	0

Fuente: Yusti F. (2015).

De este modo se observa que la señal en la posición número 74 “EAL06AN.NSP” clasificada previamente como patológica, aparentemente tiene un comportamiento similar al de las señales de voz clasificadas como sanas para lo cual se sugiere a los expertos en el área someter dicha señal a pruebas más rigurosas.

CAPITULO IV

PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

En este capítulo se presentan los resultados de los experimentos en los que se determinaron parámetros característicos de señales de voces sanas y con patologías, entre los parámetros característicos señalados se hace mención a los que involucran al vector wavelet de las señales, dicho vector wavelet se trabajó para distintos niveles de descomposición desde el nivel 1 hasta el nivel 3 y se utilizaron las bases de la familia Daubechies desde db1 hasta db8. Se realizó el entrenamiento de las máquinas de vectores de soporte con las variables originales, con las variables estandarizadas y con las variables transformadas además se fue reduciendo el número de variables transformadas de entrada a la máquina de vectores de soporte y se comparó la precisión de la clasificación entre las máquinas de vectores de soporte diseñadas.

Clasificación de Muestras de Voz y Reducción de la Dimensión de Parámetros.

Tal y como se comentó anteriormente, se procedió a realizar la clasificación de las muestras de voces trabajando en primera instancia las señales originales, las mismas fueron cargadas según las rutinas mostradas en los apéndices E y F, esto quiere decir, que como vector wavelet se utilizó el vector categorizado en el Nivel 0 según lo indica la tabla 3.5, significa que no se hizo descomposición wavelet alguna a las muestras de voz, es decir, se trabajó en el dominio del tiempo, por ende, los parámetros característicos que aporta este vector al vector de rasgo son la desviación estándar, la asimetría, la kurtosis, la varianza y el coeficiente de variación de la señal original (en el dominio del tiempo); siendo más específicos, el vector de rasgos se

calculó según el “caso 0” como se observa en la rutina implementada mostrada en el apéndice C. Los resultados del experimento se resumen en la tabla 4.1.

Tabla 4.1: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet en el nivel 0, (señal original, sin descomponer).

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 97.2447% (106/109) (classification)			
Explicación (%)	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.2'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 97.2447% (106/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.2'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 97.2477% (106/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 97.2477% (106/109) (classification)		
	Observación: error en clasificación		
Nº	Etiqueta	Etiqueta Predicha	Señal
2	1	-1	JMC1NAL.NSP
25	1	-1	SCK1NAL.NSP
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Siguiendo este mismo orden de ideas, se dispuso a descomponer las señales de voz comenzando con el Nivel 1 y utilizando la familia wavelet Db1, Tabla 4.2.

Tabla 4.2: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.2'		
Señales de Entrenamiento: Accuracy = 100% (39/40) (classification) Señales Desconocidas: Accuracy = 95.4128% (104/109) (classification)				
Explicación (%)	Entrenamiento y Clasificación	'libsvm_options'		
49.1180 75.6817 86.4089 93.3103 98.3034 99.3516 99.7661 99.9707 99.9923 99.9975 99.9992 99.9999 100.0000	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 0.2'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 94.4954% (103/109) (classification)			
	Entrenamiento y Clasificación	'libsvm_options'		
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 0.2'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 91.7431% (100/109) (classification)			
	Entrenamiento y Clasificación	'libsvm_options'		
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.1 -c 0.1'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 92.6606% (101/109) (classification)			
	Observación: error en clasificación usando 5 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
2	1	-1	JMC1NAL.NSP	
25	1	-1	SCK1NAL.NSP	
61	-1	1	DAS24AN.NSP	
74	-1	1	EAL06AN.NSP	
86	-1	1	DJM28AN.NSP	

Fuente: Yusti F. (2015).

El vector de rasgos se calculó ejecutando el “caso 1” de acuerdo a la rutina implementada mostrada en el apéndice C, en este mismo punto se procedió a ir variando la base wavelet desde Db1 hasta Db8, los resultados se ilustran en las tablas desde la 4.2 hasta la 4.9.

Tabla 4.3: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación (%)	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.4: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)				
Explicación	Entrenamiento y Clasificación	'libsvm_options'		
48.7725 74.6702 87.0272 94.2604 98.5866 99.3771 99.7770 99.9740 99.9934 99.9975 99.9992 99.9999 100.0000	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 0.1'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación	'libsvm_options'		
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 0.1'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación	'libsvm_options'		
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.1 -c 0.1'		
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP	

Fuente: Yusti F. (2015).

Tabla 4.5: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.2'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.2'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.2'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.6: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 10'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.7: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 10'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.8: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)		
Entrenamiento y Clasificación		'libsvm_options'
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 10'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)		
Entrenamiento y Clasificación		'libsvm_options'
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 1'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
Explicación	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 1'
48.7497 75.4514 85.5311 93.5179 98.2292	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)	
99.2335 99.7657 99.9710 99.9929 99.9976 99.9992 99.9999 100.0000	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)	
	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.1 -c 1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)	
Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha
74	-1	1
		Señal
		EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.9: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 1.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.2'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.2'
49.1087 75.1029 86.7256 93.4871 98.4006 99.3830 99.7718 99.9714 99.9923 99.9976 99.9992 99.9999 100.0000	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.2'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
Observación: error en clasificación usando 3 variables transformadas			
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Bajo este mismo enfoque se procedió a variar el nivel descomposición al Nivel 2, el vector de rasgos se calculó ejecutando el “caso 2” de acuerdo a la rutina implementada mostrada en el apéndice C. Se resumen los resultados en las tablas 4.10 hasta la 4.17.

Tabla 4.10: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 98.1651% (107/109) (classification)			
Explicación	Entrenamiento y Clasificación	'libsvm_options'	
47.8118 73.4917 85.5642 94.2328 97.8326 99.1032 99.7505 99.9754 99.9925 99.9968 99.9987 99.9999 100.0000	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 98.1651% (107/109) (classification)		
	Entrenamiento y Clasificación	'libsvm_options'	
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación	'libsvm_options'	
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
	Nº	Etiqueta	Etiqueta Predicha
74	-1	1	
		Señal	
		EAL06AN.NSP	

Fuente: Yusti F. (2015).

Tabla 4.11: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)		
Entrenamiento y Clasificación		'libsvm_options'
variables originales estandarizadas		'-s 0 -t 2 -g 0.1 -c 10'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)		
Entrenamiento y Clasificación		'libsvm_options'
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 1'
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 98.1651% (107/109) (classification)		
Explicación	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 1'
48.0786 73.7139 85.2156 94.1257 97.8486 99.0897 99.7532 99.9728 99.9926 99.9969 99.9987 100.0000 100.0000	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 98.1651% (107/109) (classification)	
	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)	
	Entrenamiento y Clasificación	'libsvm_options'
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)	
Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha
74	-1	1
		Señal
		EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.12: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.13: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.14: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'
48.3695	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
74.2398			
86.7260			
93.9574			
97.9421	Entrenamiento y Clasificación		'libsvm_options'
99.0849	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'
99.7549	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
99.9741			
99.9925			
99.9970			
99.9988	Entrenamiento y Clasificación		'libsvm_options'
99.9999	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'
100.0000	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
Observación: error en clasificación usando 3 variables transformadas			
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.15: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.16: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.17: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 2.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Finalmente se cambió el nivel de descomposición al Nivel 3, el vector de rasgos se calculó ejecutando el “caso 3” de acuerdo a la rutina implementada mostrada en el apéndice C. Los resultados se muestran en las tablas 4.18 hasta la .25.

Tabla 4.18: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db1 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Explicación	Entrenamiento y Clasificación	'libsvm_options'	
47.0144 70.3828 84.5602 93.2914 98.0115 99.1556 99.7703 99.9819 99.9949 99.9983 99.9999 100.0000 100.0000	con las variables Z reducidas, las primeras 5	'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)		
	Entrenamiento y Clasificación	'libsvm_options'	
	con las variables Z reducidas, las primeras 4	'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación	'libsvm_options'	
	con las variables Z reducidas, las primeras 3	'-s 0 -t 2 -g 0.01 -c 1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
	N°	Etiqueta	Etiqueta Predicha
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.19: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db2 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.20: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db3 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.21: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db4 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.22: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db5 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.01'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.01 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
				Todas bien clasificadas

Fuente: Yusti F. (2015).

Tabla 4.23: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db6 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.24: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db7 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Entrenamiento y Clasificación		'libsvm_options'		
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'		
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)				
Explicación	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Entrenamiento y Clasificación		'libsvm_options'	
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'	
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
	Observación: error en clasificación usando 3 variables transformadas			
	Nº	Etiqueta	Etiqueta Predicha	Señal
	74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

Tabla 4.25: Entrenamiento y clasificación de las señales de las tablas 3.2 y 3.3 utilizando en el vector wavelet db8 y nivel 3.

Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables originales		'-s 0 -t 2 -g 0.1 -c 0.01'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 72.4771% (79/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
variables originales estandarizadas		'-s 0 -t 2 -g 0.01 -c 1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 100% (109/109) (classification)			
Entrenamiento y Clasificación		'libsvm_options'	
con todas las variables Z		'-s 0 -t 2 -g 0.1 -c 0.1'	
Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)			
Explicación	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 5		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 4		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Entrenamiento y Clasificación		'libsvm_options'
	con las variables Z reducidas, las primeras 3		'-s 0 -t 2 -g 0.1 -c 0.1'
	Señales de Entrenamiento: Accuracy = 100% (40/40) (classification) Señales Desconocidas: Accuracy = 99.0826% (108/109) (classification)		
	Observación: error en clasificación usando 3 variables transformadas		
Nº	Etiqueta	Etiqueta Predicha	Señal
74	-1	1	EAL06AN.NSP

Fuente: Yusti F. (2015).

A todas luces, se observa cómo varía el porcentaje de precisión en la clasificación de las señales estudiadas, en las tablas 4.26 hasta la 4.29 se resumen los resultados obtenidos en la precisión de la clasificación mediante las máquinas diseñadas.

Tabla 4.26: Comparación de la precisión de la clasificación tomando las señales originales (sin descomposición wavelet alguna). Nivel 0.

	Variables			
Nivel 0	Originales	Estandarizadas	Transformadas	Reducidas a 3
Señal Original	72.4771% (79/109)	100% (109/109)	97.2447% (106/109)	97.2477% (106/109)

Fuente: Yusti F. (2015).

Tabla 4.27: Comparación de la precisión de la clasificación tomando el vector vwpt de la señal en el nivel 1 de descomposición para distintas familias wavelet.

	Variables			
Nivel 1	Originales	Estandarizadas	Transformadas	Reducidas a 3
Db1	72.4771% (79/109)	100% (109/109)	95.4128% (104/109)	92.6606% (101/109)
Db2	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db3	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db4	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db5	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db6	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db7	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db8	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)

Fuente: Yusti F. (2015).

Tabla 4.28: Comparación de la precisión de la clasificación tomando el vector vwpt de la señal en el nivel 2 de descomposición para distintas familias wavelet.

	Variables			
Nivel 2	Originales	Estandarizadas	Transformadas	Reducidas a 3
Db1	72.4771% (79/109)	100% (109/109)	98.1651% (107/109)	99.0826% (108/109)
Db2	72.4771% (79/109)	100% (109/109)	98.1651% (107/109)	99.0826% (108/109)
Db3	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db4	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db5	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db6	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db7	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db8	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)

Fuente: Yusti F. (2015).

Tabla 4.29: Comparación de la precisión de la clasificación tomando el vector $wvpt$ de la señal en el nivel 3 de descomposición para distintas familias wavelet.

Nivel 3	Variables			
	Originales	Estandarizadas	Transformadas	Reducidas a 3
Db1	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db2	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db3	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db4	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)
Db5	72.4771% (79/109)	100% (109/109)	100% (109/109)	100% (109/109)
Db6	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db7	72.4771% (79/109)	100% (109/109)	100% (109/109)	99.0826% (108/109)
Db8	72.4771% (79/109)	100% (109/109)	99.0826% (108/109)	99.0826% (108/109)

Fuente: Yusti F. (2015).

Así se ha verificado que aún reduciendo la dimensionalidad del conjunto de variables que describen las señales de voz ha sido posible encontrar un nivel de descomposición y una base wavelet en la que el conjunto total de datos sean bien clasificados, está a ojos de vista en la tabla 4.29 que el nivel 3 de descomposición junto a la función wavelet Db5 permiten obtener un 100% de precisión en la clasificación de las señales seleccionadas para el análisis, en la tabla 4.22 se observa que este conjunto de variables reducidas ofrecía un 84.7796% de explicación de la variabilidad total de los datos.

Cabe considerar por otra parte que la señal EAL06AN.NSP fue la más complicada de clasificar en todos los experimentos realizados, en la base de datos suministrada por los expertos en el área dicha señal fue clasificada como patológica sin embargo con las señales de entrenamiento seleccionadas y las máquinas de vectores de soporte diseñadas las cuales funcionaron correctamente con las variables estandarizadas, dicha señal en algunos casos fue clasificada como sana, en la figura 4.1 se observa como dicha señal se encuentra cerca del hiperplano separador de las clases, la figura fue elaborada utilizando los dos primeros parámetros característicos de la matriz de variables transformadas Z_i construida para un nivel 3 de

descomposición y wavelet Db5, recogiendo un 71.1461% de varianza explicada, en la cual se utilizaron los siguientes comandos pertenecientes a la librería “Bioinformatics Toolbox” de la siguiente forma:

```
SVMStruct = svmtrain(Ztrain(:,1:2),grupo,'kernel_function','rbf','rbf_sigma',0.23...
    ..., 'method','QP','boxconstraint',10,'showplot',true)
```

```
Group = svmclassify(SVMStruct,Ztrains(74,1:2),'Showplot',true)
```

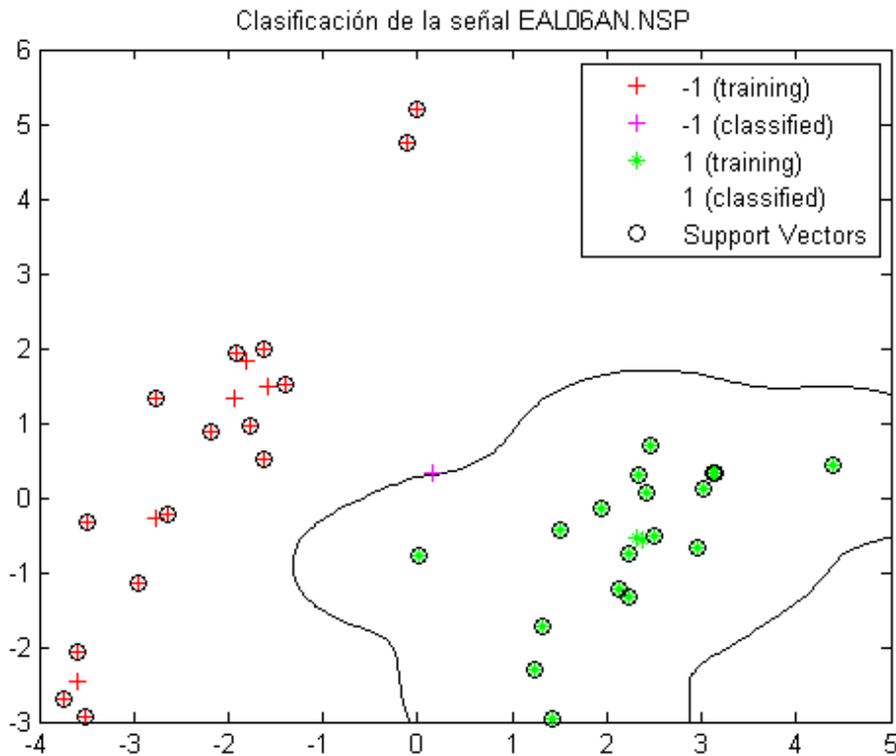


Figura 4.1: Clasificación de la señal EAL06AN.NSP, se observa su cercanía al hiperplano separador.

Fuente: Yusti F. (2015).

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

Desde la perspectiva más general basados en las generalizaciones anteriores, toda reflexión conlleva a las siguientes conclusiones y recomendaciones.

Conclusiones

En atención a la problemática expuesta, referida a la disminución de la dimensionalidad del número de parámetros característicos de las señales de voz necesarios para realizar una clasificación mediante maquinas de vectores de soporte, se evidenció que el análisis mediante componentes principales permite obtener un espacio de variables transformadas las cuales recogen información de las variables originales mediante combinaciones lineales de las mismas y por ende conseguir un espacio de variables de menor dimensión al inicial, por supuesto con pérdidas de información pero puede escogerse la precisión de la estimación seleccionando una cantidad de variables que expliquen en un porcentaje aceptable la totalidad de la varianza de los datos.

En lo que respecta al uso de la técnica de contracción garrote para el filtrado de las señales de voz, se analizó y se puso a prueba su principio de funcionamiento, entregando resultados satisfactorios en la estimación del nivel de ruido presente en las señales de voz utilizando la base wavelet Db4 en los niveles de descomposición 1, 2 y 3.

El análisis de componentes principales resulta útil al momento de reducir la dimensionalidad del conjunto en vista que cuando se tienen las variables originales se hace complicado o exhaustivo reducir el número de parámetros por cuanto resultan varias combinaciones de escogencia, es decir, por ejemplo se podrían tomar las 3 primeras variables o las tres últimas, o las 5 primeras o las que estén al centro, en fin si se tienen N variables se podrían tomar N combinaciones de grupos de M variables, mientras que con el análisis de componentes principales, las variables se van ordenando en función de la varianza explicada y siempre se escogen en el orden desde la primera hasta la última.

Las señales de voz fueron descompuestas utilizando la transformada wavelet packet en los niveles de descomposición 1, 2 y 3, para cada nivel de descomposición se obtuvo el vector wavelet de la señal al cual se le calcularon los cuatro momentos estadísticos así como la varianza y el coeficiente de variación para obtener de esta manera un conjunto de rasgos o parámetros que sean característicos de cada grupo de señales y poder utilizar dichos parámetros como entrada a una máquina de vectores de soporte.

En este mismo orden de ideas también se determinó un vector que contenía los valores de PMR por cada segmento en que la señal fue dividida, a dicho vector se le determinó los tres últimos momentos estadísticos junto con la varianza y el coeficiente de variación, obteniendo así otro conjunto de rasgos que se utilizaron para caracterizar cada grupo de señales.

En el proceso de clasificación, se realizaron ensayos llevando a la máquina de vectores de soporte, las variables originales obteniéndose un 72.4771% de precisión en la clasificación, también se ensayó con las variables estandarizadas obteniéndose

en todos los casos un 100% de precisión en la clasificación y finalmente se llevaron a la entrada de la máquina de vectores de soporte un conjunto reducido de variables transformadas, obteniéndose un 100% de precisión en la clasificación cuando se utilizó en el vector wavelet de la señal el nivel 3 de descomposición y la base wavelet Db5, se redujo de un conjunto de 13 variables de entrada a un conjunto de sólo 3 variables de entrada; es decir, se redujo del 100% al 20% del total de variables de entrada a la máquina de vectores de soporte.

Recomendaciones

Se recomienda realizar el estudio incrementando el número de descomposiciones en el vector wavelet de la señal así como utilizar otras familias wavelet; además de determinar el vector wavelet de la señal según criterios de mejor árbol de descomposición de la señal.

En todo caso, se recomienda incrementar la base de datos de señales de voz sanas y patológicas a un número de muestras bastantes significativas para dar mayor solidez a los resultados y para lograr un entrenamiento de la máquina de vectores de soporte con un número mayor de vectores a la entrada.

Basado en las generalizaciones anteriores, se recomienda el uso de esta metodología para ser aplicadas en otro tipo de señales.

REFERENCIAS BIBLIOGRÁFICAS.

Arias, Fidas (1999). **El Proyecto de Investigación: Guía para su elaboración.** Editorial Episteme C.A./ Oriol Ediciones. Caracas Venezuela.

Capoccioni, Dianora (2006). **Procesamiento Digital de Señal Fonética para Establecer Patrones de Pronunciación Correcta que Permitan Contrastar Dislalia en Castellano.** Tesis de Maestría. Universidad de Carabobo.

Cater J. (1984). **Electronically Hearing: Computer Speech Recognition,** 1st Edition. Howard W. Sams & Co. Inc.

Chui C.K. (1992). **Wavelets: a tutorial in theory and applications,** Academic Press.

Del Pino, Paulino (2003). **Identificación de los parámetros espectrales que determinan la calidad de la voz.** Tesis de Maestría. Universidad de Carabobo Venezuela.

Del Pino, P., Jiménez, C., Díaz, J., y Rothman, H. (2008). **Aplicación de la transformada de wavelet para el análisis de señales de voz normales y patológicas.** Artículo, Revista Ingeniería UC. Vol. 15. No. 1, Abril, Venezuela.

Faundez, P., Fuentes, A. (2002). **Procesamiento Digital de Señales Acústicas Utilizando Wavelets.** Tesis de Maestría Instituto de Matemática UACH.

Flores A. (1993). **Reconocimiento de Palabras Aisladas en Castellano**. Inictel. Dirección de Investigación y Desarrollo.

Jiménez, Carlos (2004). **Análisis de la relación señal a ruido de la voz, utilizando la transformada de Wavelet**. Tesis de Maestría. Universidad de Carabobo Venezuela.

Jiménez C., Jiménez J. (2010). **Clasificación de señales de voz utilizando Transformada de Wavelet y Máquinas de Vectores de Soporte**. Trabajo de ascenso. Facultad de Ingeniería. Universidad de Carabobo.

Jolliffe I. T. (2002). **Principal Component Analysis**, 2nd edition, Springer.

Kabal P. (2012). **MMSP Lab, ECE**, McGill University: Last update: 2011-01-12(with contributions from R. Roark).

Kaiser G. (1994). **A friendly guide to wavelets**, Birkhäuser.

Lohr, S. (2010). **Muestreo: Diseño y análisis**. Thompson Editores. Arizona State University.

Orozco, C., Labrador, M., y Palencia de M., A. (2002). **Metodología: Manual Teórico Práctico de Metodología para Tesistas, Asesores, Tutores y Jurados de Trabajos de investigación y Ascenso**. Valencia: Ofimax de Venezuela, C.A.

Palella y Martins (2006). **Métodos de la Investigación**. Editorial Trilla. México.

Peralta F., Cotrina A. (2002). **Reconocedor y analizador de voz**. Universidad Nacional Mayor de San Marcos.

Vega, C. (2003). **Aplicación de técnicas wavelet al análisis de series temporales**. Tesis doctoral no publicada. Universidad de Granada, Facultad de Ciencias, Departamento de Estadística e Investigación Operativa.

Vega, C. (2009). **Un método automático para la elección de la base de ondícula y la selección del umbral para la estimación de señales**. Artículo, Revista Ingeniería UC, Volumen 16, N°1, Abril, Venezuela.

APÉNDICE A

Función que aplica la técnica de contracción GARROTE para filtrar una señal.

```

function signalf=filtrarGarrote(signal,nivel);

xnoise=signal;

N=nivel; %nivel de umbralizado
wavelet='db4'; %tipo de wavelet madre

[thr,sorh,keepapp]=ddencomp('den','wv',xnoise);

%%filtrado con un umbral garrote%%
%descomposicion de la señal(nivel N)%
[C,L]=wavedec(xnoise,N,wavelet);

cd= C(L(1)+1:L(1)+L(2));

%umbralizado
n=length(cd);
%thr = thselect(xnoise,'sqrtwolog');
%delta=sum(abs(cd))/(0.6745*n);
Umbral=thr;%*delta; %esto es landa para la contraccion garrote%

%%contraccion garrote%%
for k=1:n
    if (abs(cd(k))-Umbral)>0
        cdu(k,1)=cd(k)*(1-(Umbral/cd(k))^2);%COEF. DE DETALLE UMBRALIZADOS%
    else
        cdu(k,1)=0;
    end
end

C(L(1)+1:L(1)+L(2))=cdu;
signalf = waverec(C,L,wavelet);
end

```

Validación de la función:

Para tal fin, se procedió a crear una señal a la que se le añade ruido blanco con una relación señal a ruido conocida. Para lograr esto se utilizó el comando:

$$[x,xn]=wnoise(Num,N,snrat)$$

Con el comando “wnoise” se le añade a una señal (x) ruido blanco con una relación señal a ruido de $(snrat)^2$ para generar la señal con ruido (xn). Aclaremos que $snrat$: es la raíz cuadrada del valor de la relación señal a ruido deseada. Además, x es la señal que depende del valor de Num , reescalada de tal modo que su desviación estándar ($srtcd$) sea igual al valor de la relación señal a ruido deseada, es decir $snrat$.

$$x=x*(snrat)/srtcd(x)$$

Finalmente, xn será la señal con ruido “blanco gaussiano”, que se genera con una relación señal a ruido $(snrat)^2$, mediante:

$$xn = x + wn$$

Se observa entonces que wn es el ruido blanco gaussiano con desviación estándar igual a uno y valor medio cero y N es el número de muestras de las señales generadas (x, xn) igual a 2^N .

Se realizó la siguiente prueba, introduciendo diferentes valores de snr y variando la función wavelet así como el nivel de descomposición:

```

%%programa para limpiar la señal de prueba cambiando el nivel de
descomposición%%

snrdb=30;%%relación señal a ruido deseada en decibeles%%
snr=10^(snrdb/10);%%relación señal a ruido deseada%%
snr=sqrt(snr);%%se aplica la raíz cuadrada porque la función wnoise
trabaja con el cuadrado de SNR%%

[xsruido,xnoise]=wnoise(3,12,snr,2055415866);%%esta funcion produce una
señal con ruido con snr conocido

xf = filtrarGarrote(xnoise,1);

subplot(211)
plot(xnoise,'k')
ylabel('Amplitud')
title('Señal con ruido gaussiano')
subplot(212)
plot(xf)
ylabel('Amplitud')
title('Señal filtrada')

% calculo de la relacion señal a ruido
SNR=10*log10(sum(xnoise.^2)/sum((xf-xnoise).^2))

```

Ensayo con una snr de 30db, wavelet db6 y nivel 1 de descomposición, en la figura A1 se observa la señal con ruido y la señal filtrada así como el snr estimado, el cual es bastante aproximado al snr que se añadió intencionalmente.

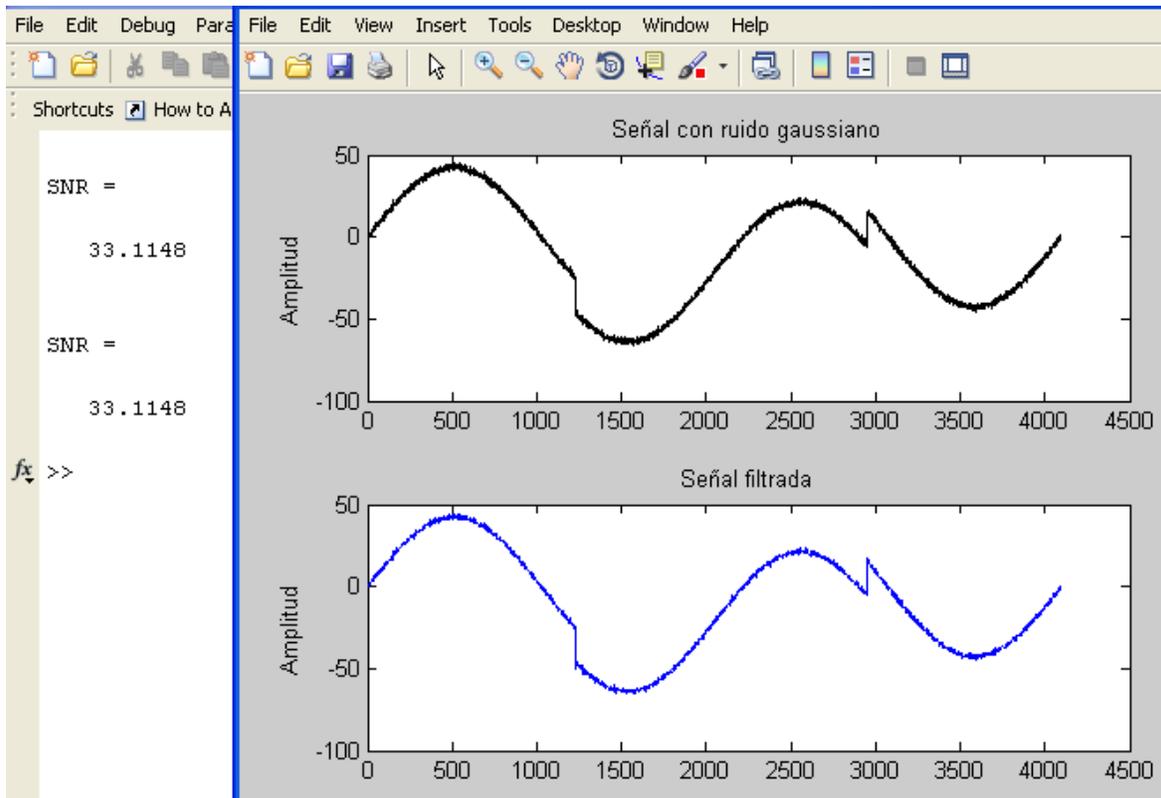


Figura A1: Filtrado de una señal con ruido cuya relación señal a ruido es conocida, $snr=30db$.

Ensayo con una snr de 50db (en este caso sería una señal prácticamente libre de ruido), wavelet db6 y nivel 1 de descomposición, en la figura A2 se observa la señal con ruido y la señal filtrada así como el snr estimado, el cual es bastante aproximado al snr que se añadió intencionalmente. Con esta relación señal a ruido bastante alta se observa que la rutina implementada no realiza sobre estimación de la señal a filtrar.

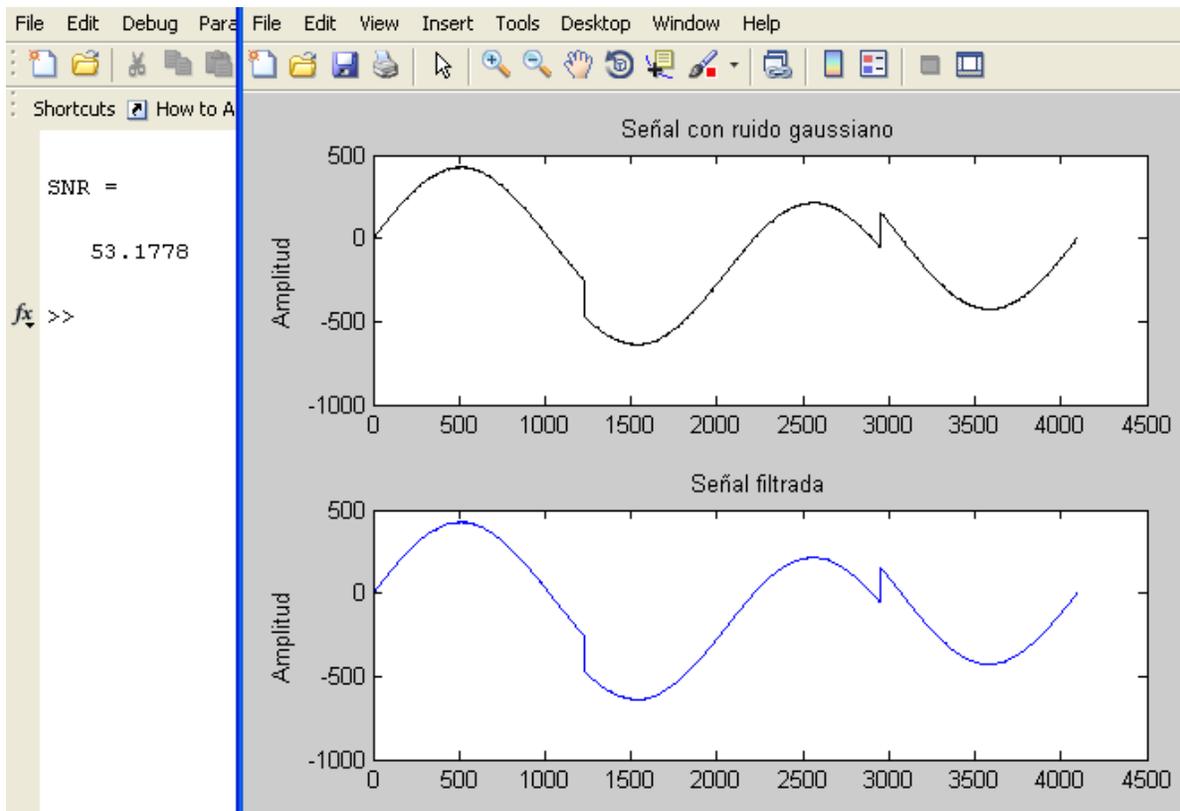


Figura A2: Filtrado de una señal con ruido cuya relación señal a ruido es conocida, $\text{snr}=50\text{db}$.

Los mejores resultados de estimación de ruido y filtrado de la señal se obtuvieron con la wavelet db6 y los niveles de descomposición desde el 1 hasta el 3.

Ensayo con una snr de 10db (en este caso sería una señal con una cantidad considerable de ruido), wavelet db6 y nivel 1 de descomposición, en la figura A3 se observa la señal con ruido y la señal filtrada así como el snr estimado, el cual es bastante aproximado al snr que se añadió intencionalmente. Con esta relación señal a ruido bastante baja se observa que la rutina implementada realiza un suavizado en la señal a filtrar.

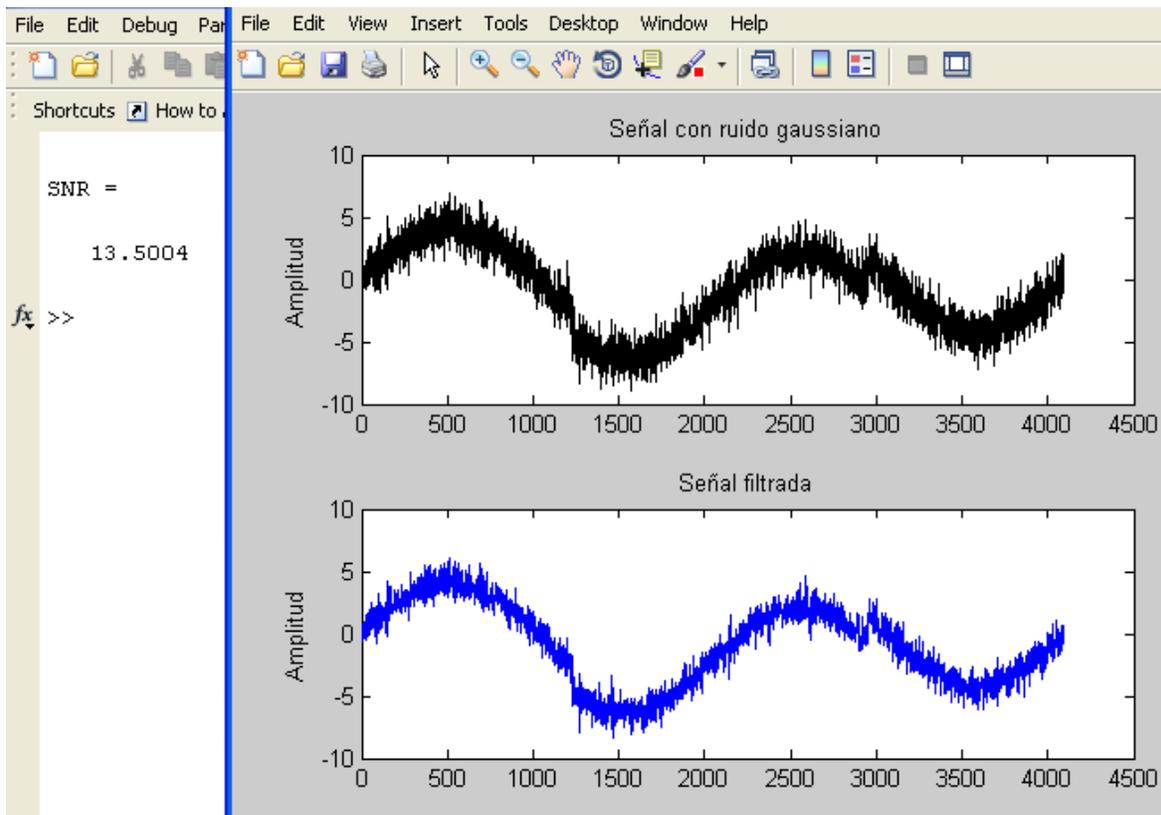


Figura A3: Filtrado de una señal con ruido cuya relación señal a ruido es conocida, $\text{snr}=10\text{db}$.

Los mejores resultados de estimación de ruido y filtrado de la señal se obtuvieron con la wavelet db6 y los niveles de descomposición desde el 1 hasta el 3.

APÉNDICE B

Fórmulas de las variables estadísticas utilizadas.

Desviación Estándar:

$$s = \text{std}(X)$$

$$s = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$$

Varianza:

$$V = \text{var}(X)$$

“The variance is the square of the standard deviation (STD)”.

Coefficiente de Asimetría:

$$y = \text{skewness}(X)$$

$$s_1 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \right)^3}$$

Kurtosis:

$$k = \text{kurtosis}(X)$$

$$k_1 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^2}$$

Coefficiente de Variación:

Representa la proporción entre la desviación estándar y la media de los datos:

$$CV = \frac{\text{std}(X)}{\bar{X}}$$

APÉNDICE C

Función que construye un vector con los parámetros característicos seleccionados de las señales de voz en estudio.

```

function VectorDeRasgos=CalcularVectorDeRasgosWPT (SignalVoz,nivel);
warning off
xx=SignalVoz;n=nivel;
wavelet='db6';

switch n

case 0
    wpt = wpdec(xx,n,wavelet);
    s=wpcoef(wpt,[0 0]);
    vwpt = s;

    k=abs(fft(xx,length(xx)));
    PMR=max(k)/mean(k);

    xf=filtrarGarrote(xx,1);
    kf=abs(fft(xf));
    PMRF=max(kf)/mean(kf);

    senal=xx;
    x=linspace(1,length(senal),100);
    fftSenal=abs(fft(senal));
    for k=1:length(x)-1
        Nf=fftSenal(x(k):x(k+1));
        PMRseg(k) = max(Nf)/mean(Nf);
    end

    CVPMR=std(PMRseg)/mean(PMRseg);
    CVvwpt=std(vwpt)/abs(mean(vwpt));
    VectorDeRasgos=[PMR PMRF std(PMRseg) skewness(PMRseg)...
        ...kurtosis(PMRseg) var(PMRseg) CVPMR mean(vwpt)...
        ... std(vwpt) skewness(vwpt) kurtosis(vwpt)...
        ... var(vwpt) CVvwpt];

case 1
    wpt = wpdec(xx,n,wavelet);
    A1=wpcoef(wpt,[1 0]);
    D1=wpcoef(wpt,[1 1]);
    vwpt = [A1' D1'];

    k=abs(fft(xx,length(xx)));
    PMR=max(k)/mean(k);

    xf=filtrarGarrote(xx,n);
    kf=abs(fft(xf));
    PMRF=max(kf)/mean(kf);

    senal=xx;
    x=linspace(1,length(senal),100);
    fftSenal=abs(fft(senal));
    for k=1:length(x)-1
        Nf=fftSenal(x(k):x(k+1));
        PMRseg(k) = max(Nf)/mean(Nf);
    end
end

```

```

CVPMR=std(PMRseg)/mean(PMRseg);
CVvwpt=std(vwpt)/abs(mean(vwpt));
VectorDeRasgos=[PMR PMRF std(PMRseg) skewness(PMRseg)...
...kurtosis(PMRseg) var(PMRseg) CVPMR mean(vwpt)...
... std(vwpt) skewness(vwpt) kurtosis(vwpt)...
... var(vwpt) CVvwpt];

```

case 2

```

wpt = wpdec(xx,n,wavelet);
AA2=wpcoef(wpt,[2 0]);
DA2=wpcoef(wpt,[2 1]);
AD2=wpcoef(wpt,[2 2]);
DD2=wpcoef(wpt,[2 3]);
vwpt = [AA2' DA2' AD2' DD2'];

```

```

k=abs(fft(xx,length(xx)));
PMR=max(k)/mean(k);

```

```

xf=filtrarGarrote(xx,n);
kf=abs(fft(xf));
PMRF=max(kf)/mean(kf);

```

```

senal=xx;
x=linspace(1,length(senal),100);
fftSenal=abs(fft(senal));
for k=1:length(x)-1
    Nf=fftSenal(x(k):x(k+1));
    PMRseg(k) = max(Nf)/mean(Nf);
end

```

end

```

CVPMR=std(PMRseg)/mean(PMRseg);
CVvwpt=std(vwpt)/abs(mean(vwpt));
VectorDeRasgos=[PMR PMRF std(PMRseg) skewness(PMRseg)...
...kurtosis(PMRseg) var(PMRseg) CVPMR mean(vwpt)...
... std(vwpt) skewness(vwpt) kurtosis(vwpt)...
... var(vwpt) CVvwpt];

```

case 3

```

wpt = wpdec(xx,n,wavelet);
AAA3=wpcoef(wpt,[3 0]);
DAA3=wpcoef(wpt,[3 1]);
ADA3=wpcoef(wpt,[3 2]);
DDA3=wpcoef(wpt,[3 3]);
AAD3=wpcoef(wpt,[3 4]);
DAD3=wpcoef(wpt,[3 5]);
ADD3=wpcoef(wpt,[3 6]);
DDD3=wpcoef(wpt,[3 7]);
vwpt = [AAA3' DAA3' ADA3' DDA3' AAD3' DAD3' ADD3' DDD3'];

```

```

k=abs(fft(xx,length(xx)));
PMR=max(k)/mean(k);

```

```

xf=filtrarGarrote(xx,n);
kf=abs(fft(xf));

```

```

PMRF=max(kf)/mean(kf);

senal=xx;
x=linspace(1,length(senal),100);
fftSenal=abs(fft(senal));
for k=1:length(x)-1
    Nf=fftSenal(x(k):x(k+1));
    PMRseg(k) = max(Nf)/mean(Nf);
end

CVPMR=std(PMRseg)/mean(PMRseg);
CVvwpt=std(vwpt)/abs(mean(vwpt));
VectorDeRasgos=[PMR PMRF std(PMRseg) skewness(PMRseg)...
    ...kurtosis(PMRseg) var(PMRseg) CVPMR mean(vwpt)...
    ... std(vwpt) skewness(vwpt) kurtosis(vwpt)...
    ... var(vwpt) CVvwpt];

otherwise
    disp('nivel debe ser 0, 1, 2, 3')
end
end

```

APÉNDICE D

Función “svmtrain”.de la librería libsvm-mat-2.86.

Usage: model = svmtrain(training_label_vector, training_instance_matrix, 'libsvm_options');

libsvm_options:

- s svm_type : set type of SVM (default 0)
 - 0 -- C-SVC
 - 1 -- nu-SVC
 - 2 -- one-class SVM
 - 3 -- epsilon-SVR
 - 4 -- nu-SVR
- t kernel_type : set type of kernel function (default 2)
 - 0 -- linear: $u \cdot v$
 - 1 -- polynomial: $(\gamma u \cdot v + \text{coef0})^{\text{degree}}$
 - 2 -- radial basis function: $\exp(-\gamma |u-v|^2)$
 - 3 -- sigmoid: $\tanh(\gamma u \cdot v + \text{coef0})$
 - 4 -- precomputed kernel (kernel values in training_instance_matrix)
- d degree : set degree in kernel function (default 3)
- g gamma : set gamma in kernel function (default 1/k)
- r coef0 : set coef0 in kernel function (default 0)
- c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
- n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
- p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
- m cachesize : set cache memory size in MB (default 100)
- e epsilon : set tolerance of termination criterion (default 0.001)
- h shrinking: whether to use the shrinking heuristics, 0 or 1 (default 1)
- b probability_estimates: whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
- wi weight: set the parameter C of class i to weight*C, for C-SVC (default 1)
- v n: n-fold cross validation mode

En este trabajo de investigación, se utilizó la sintaxis siguiente:

```
model= svmtrain(grupo,Ztrain(:,1:5),'-s 0 -t 2 -g 0.1 -c 0.01');
```

- s 0 para indicar que el tipo de máquina es un clasificador.
- t 2 para indicar que el tipo de función kernel es rbf (por lo tanto hay que suministrar el valor de σ (ecuación 2.26).
- g 0.1 para introducir el valor de σ (ecuación 2.26).
- c 0.01 constante de penalización C (missclassification tolerance parameter).

APÉNDICE E

Se cargan las señales clasificadas como sanas.

```
fid1 = fopen('AXH1NAL.NSP');
N01 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BJB1NAL.NSP');
N02 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BJV1NAL.NSP');
N03 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CAD1NAL.NSP');
N04 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CEB1NAL.NSP');
N05 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DAJ1NAL.NSP');
N06 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DFP1NAL.NSP');
N07 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DJG1NAL.NSP');
N08 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DMA1NAL.NSP');
N09 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DWS1NAL.NSP');
N10 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EDC1NAL.NSP');
N11 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EJC1NAL.NSP');
N12 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FMB1NAL.NSP');
N13 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('LAD1NAL.NSP');
N26 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('LDP1NAL.NSP');
N27= fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('LLA1NAL.NSP');
N28 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('LMV1NAL.NSP');
N29 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('LMW1NAL.NSP');
N30 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MAM1NAL.NSP');
N31 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MAS1NAL.NSP');
N32 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MCB1NAL.NSP');
N33 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MFM1NAL.NSP');
N34 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MJU1NAL.NSP');
N35 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MXB1NAL.NSP');
N36 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('MXZ1NAL.NSP');
N37 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('NJS1NAL.NSP');
N38 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GPC1NAL.NSP');
N14 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GZZ1NAL.NSP');
N15 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HBL1NAL.NSP');
N16 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JAF1NAL.NSP');
N17 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JAN1NAL.NSP');
N18 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JAP1NAL.NSP');
N19 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JEG1NAL.NSP');
N20 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JKR1NAL.NSP');
N21 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JMC1NAL.NSP');
N22 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JTH1NAL.NSP');
N23 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('JXC1NAL.NSP');
N24 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('KAN1NAL.NSP');
N25 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('OVK1NAL.NSP');
N39 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('PBD1NAL.NSP');
N40 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('PCA1NAL.NSP');
N41 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('RHG1NAL.NSP');
N42 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('RHM1NAL.NSP');
N43 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('RJS1NAL.NSP');
N44 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SCK1NAL.NSP');
N45 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SCT1NAL.NSP');
N46 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SEB1NAL.NSP');
N47 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SIS1NAL.NSP');
N48 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SLC1NAL.NSP');
N49 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('SXV1NAL.NSP');
N50 = fread(fid1);
fclose(fid1);
```

APÉNDICE F

Se cargan las señales clasificadas como patológicas.

```
fid1 = fopen('AAK02AN.NSP');
P01 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AAS16AN.NSP');
P02 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AAT30AN.NSP');
P03 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AAT31AN.NSP');
P04 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ADP02AN.NSP');
P05 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ADP11AN.NSP');
P06 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AFR17AN.NSP');
P07 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AHK02AN.NSP');
P08 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AHS20AN.NSP');
P09 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AJM29AN.NSP');
P10 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AJP25AN.NSP');
P11 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ALW27AN.NSP');
P12 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ALW28AN.NSP');
P13 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DAS24AN.NSP');
P51 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DBA02AN.NSP');
P52 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DGL30AN.NSP');
P53 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DGO03AN.NSP');
P54 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DJM14AN.NSP');
P55 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DJM28AN.NSP');
P56 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DLL25AN.NSP');
P57 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DLT09AN.NSP');
P58 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DLW04AN.NSP');
P59 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DMF11AN.NSP');
P60 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DMR27AN.NSP');
P61 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DMS01AN.NSP');
P62 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DOA27AN.NSP');
P63 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AMJ23AN.NSP');
P14 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AMP12AN.NSP');
P15 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('AMV23AN.NSP');
P16 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ASR20AN.NSP');
P17 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ASR23AN.NSP');
P18 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BAS19AN.NSP');
P19 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BAT19AN.NSP');
P20 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BBR24AN.NSP');
P21 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BGS05AN.NSP');
P22 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BJH05AN.NSP');
P23 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BJK29AN.NSP');
P24 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BMK05AN.NSP');
P25 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('BMM09AN.NSP');
P26 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CBR29AN.NSP');
P27 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EAL06AN.NSP');
P64 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EAW21AN.NSP');
P65 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EBJ03AN.NSP');
P66 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EGK30AN.NSP');
P67 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EGW23AN.NSP');
P68 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EMD08AN.NSP');
P69 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EOW04AN.NSP');
P70 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ERS07AN.NSP');
P71 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ESM05AN.NSP');
P72 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('ESP04AN.NSP');
P73 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EXH21AN.NSP');
P74 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('EXL23AN.NSP');
P75 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FAH01AN.NSP');
P76 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FJL23AN.NSP');
P77 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CCM15AN.NSP');
P28 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CCP21AN.NSP');
P29 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CCP29AN.NSP');
P30 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CDW03AN.NSP');
P31 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CEN21AN.NSP');
P32 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CFW04AN.NSP');
P33 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CJP10AN.NSP');
P34 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CMR01AN.NSP');
P35 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CMS25AN.NSP');
P36 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CNP07AN.NSP');
P37 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CNR01AN.NSP');
P38 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CPK19AN.NSP');
P39 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CPK21AN.NSP');
P40 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CSJ16AN.NSP');
P41 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FLW13AN.NSP');
P78 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FMQ20AN.NSP');
P79 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FRH18AN.NSP');
P80 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FSP13AN.NSP');
P81 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FXE24AN.NSP');
P82 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('FXI23AN.NSP');
P83 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GCU31AN.NSP');
P84 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GEA24AN.NSP');
P85 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GEK02AN.NSP');
P86 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GJW09AN.NSP');
P87 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GLB01AN.NSP');
P88 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GLB22AN.NSP');
P89 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GMS03AN.NSP');
P90 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GMT01AN.NSP');
P91 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CSY01AN.NSP');
P42 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXL08AN.NSP');
P43 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXM07AN.NSP');
P44 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXM14AN.NSP');
P45 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXN14AN.NSP');
P46 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXP02AN.NSP');
P47 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('CXT08AN.NSP');
P48 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DAM08AN.NSP');
P49 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('DAS10AN.NSP');
P50 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GRS20AN.NSP');
P92 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('GSL04AN.NSP');
P93 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HBS12AN.NSP');
P94 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HLK01AN.NSP');
P95 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HLK15AN.NSP');
P96 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HML26AN.NSP');
P97 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HXL28AN.NSP');
P98 = fread(fid1);
fclose(fid1);
```

```
fid1 = fopen('HXR23AN.NSP');
P99 = fread(fid1);
fclose(fid1);
```

APÉNDICE G

Creación de la matriz de entrenamiento “xtrainWPT”.

```

xtrainWPT(1,:)=CalcularVectorDeRasgosWPT(N01,nivel);
xtrainWPT(2,:)=CalcularVectorDeRasgosWPT(N02,nivel);
xtrainWPT(3,:)=CalcularVectorDeRasgosWPT(N03,nivel);
xtrainWPT(4,:)=CalcularVectorDeRasgosWPT(N04,nivel);
xtrainWPT(5,:)=CalcularVectorDeRasgosWPT(N05,nivel);
xtrainWPT(6,:)=CalcularVectorDeRasgosWPT(N06,nivel);
xtrainWPT(7,:)=CalcularVectorDeRasgosWPT(N07,nivel);
xtrainWPT(8,:)=CalcularVectorDeRasgosWPT(N08,nivel);
xtrainWPT(9,:)=CalcularVectorDeRasgosWPT(N09,nivel);
xtrainWPT(10,:)=CalcularVectorDeRasgosWPT(N10,nivel);
xtrainWPT(11,:)=CalcularVectorDeRasgosWPT(N11,nivel);
xtrainWPT(12,:)=CalcularVectorDeRasgosWPT(N12,nivel);
xtrainWPT(13,:)=CalcularVectorDeRasgosWPT(N13,nivel);
xtrainWPT(14,:)=CalcularVectorDeRasgosWPT(N14,nivel);
xtrainWPT(15,:)=CalcularVectorDeRasgosWPT(N15,nivel);
xtrainWPT(16,:)=CalcularVectorDeRasgosWPT(N16,nivel);
xtrainWPT(17,:)=CalcularVectorDeRasgosWPT(N17,nivel);
xtrainWPT(18,:)=CalcularVectorDeRasgosWPT(N18,nivel);
xtrainWPT(19,:)=CalcularVectorDeRasgosWPT(N19,nivel);
xtrainWPT(20,:)=CalcularVectorDeRasgosWPT(N20,nivel);

```

```

xtrainWPT(21,:)=CalcularVectorDeRasgosWPT(P01,nivel);
xtrainWPT(22,:)=CalcularVectorDeRasgosWPT(P02,nivel);
xtrainWPT(23,:)=CalcularVectorDeRasgosWPT(P03,nivel);
xtrainWPT(24,:)=CalcularVectorDeRasgosWPT(P04,nivel);
xtrainWPT(25,:)=CalcularVectorDeRasgosWPT(P05,nivel);
xtrainWPT(26,:)=CalcularVectorDeRasgosWPT(P06,nivel);
xtrainWPT(27,:)=CalcularVectorDeRasgosWPT(P07,nivel);
xtrainWPT(28,:)=CalcularVectorDeRasgosWPT(P08,nivel);
xtrainWPT(29,:)=CalcularVectorDeRasgosWPT(P09,nivel);
xtrainWPT(30,:)=CalcularVectorDeRasgosWPT(P10,nivel);
xtrainWPT(31,:)=CalcularVectorDeRasgosWPT(P11,nivel);
xtrainWPT(32,:)=CalcularVectorDeRasgosWPT(P12,nivel);
xtrainWPT(33,:)=CalcularVectorDeRasgosWPT(P13,nivel);
xtrainWPT(34,:)=CalcularVectorDeRasgosWPT(P14,nivel);
xtrainWPT(35,:)=CalcularVectorDeRasgosWPT(P15,nivel);
xtrainWPT(36,:)=CalcularVectorDeRasgosWPT(P16,nivel);
xtrainWPT(37,:)=CalcularVectorDeRasgosWPT(P17,nivel);
xtrainWPT(38,:)=CalcularVectorDeRasgosWPT(P18,nivel);
xtrainWPT(39,:)=CalcularVectorDeRasgosWPT(P19,nivel);
xtrainWPT(40,:)=CalcularVectorDeRasgosWPT(P20,nivel);

```

```

%Etiqueta de grupo, 1 para sanas, -1 para patológicas
grupo=[ones(20,1);-1*ones(20,1)];

```

APÉNDICE H

Creación de la matriz para clasificación “SxtrainWPT”.

```
SxtrainWPT(1,:)=CalcularVectorDeRasgosWPT(N21,nivel);
SxtrainWPT(2,:)=CalcularVectorDeRasgosWPT(N22,nivel);
SxtrainWPT(3,:)=CalcularVectorDeRasgosWPT(N23,nivel);
SxtrainWPT(4,:)=CalcularVectorDeRasgosWPT(N24,nivel);
SxtrainWPT(5,:)=CalcularVectorDeRasgosWPT(N25,nivel);
SxtrainWPT(6,:)=CalcularVectorDeRasgosWPT(N26,nivel);
SxtrainWPT(7,:)=CalcularVectorDeRasgosWPT(N27,nivel);
SxtrainWPT(8,:)=CalcularVectorDeRasgosWPT(N28,nivel);
SxtrainWPT(9,:)=CalcularVectorDeRasgosWPT(N29,nivel);
SxtrainWPT(10,:)=CalcularVectorDeRasgosWPT(N30,nivel);
SxtrainWPT(11,:)=CalcularVectorDeRasgosWPT(N31,nivel);
SxtrainWPT(12,:)=CalcularVectorDeRasgosWPT(N32,nivel);
SxtrainWPT(13,:)=CalcularVectorDeRasgosWPT(N33,nivel);
SxtrainWPT(14,:)=CalcularVectorDeRasgosWPT(N34,nivel);
SxtrainWPT(15,:)=CalcularVectorDeRasgosWPT(N35,nivel);
SxtrainWPT(16,:)=CalcularVectorDeRasgosWPT(N36,nivel);
SxtrainWPT(17,:)=CalcularVectorDeRasgosWPT(N37,nivel);
SxtrainWPT(18,:)=CalcularVectorDeRasgosWPT(N38,nivel);
SxtrainWPT(19,:)=CalcularVectorDeRasgosWPT(N39,nivel);
SxtrainWPT(20,:)=CalcularVectorDeRasgosWPT(N40,nivel);
SxtrainWPT(21,:)=CalcularVectorDeRasgosWPT(N41,nivel);
SxtrainWPT(22,:)=CalcularVectorDeRasgosWPT(N42,nivel);
SxtrainWPT(23,:)=CalcularVectorDeRasgosWPT(N43,nivel);
SxtrainWPT(24,:)=CalcularVectorDeRasgosWPT(N44,nivel);
SxtrainWPT(25,:)=CalcularVectorDeRasgosWPT(N45,nivel);
SxtrainWPT(26,:)=CalcularVectorDeRasgosWPT(N46,nivel);
SxtrainWPT(27,:)=CalcularVectorDeRasgosWPT(N47,nivel);
SxtrainWPT(28,:)=CalcularVectorDeRasgosWPT(N48,nivel);
SxtrainWPT(29,:)=CalcularVectorDeRasgosWPT(N49,nivel);
SxtrainWPT(30,:)=CalcularVectorDeRasgosWPT(N50,nivel);
SxtrainWPT(31,:)=CalcularVectorDeRasgosWPT(P21,nivel);
SxtrainWPT(32,:)=CalcularVectorDeRasgosWPT(P22,nivel);
SxtrainWPT(33,:)=CalcularVectorDeRasgosWPT(P23,nivel);
SxtrainWPT(34,:)=CalcularVectorDeRasgosWPT(P24,nivel);
SxtrainWPT(35,:)=CalcularVectorDeRasgosWPT(P25,nivel);
SxtrainWPT(36,:)=CalcularVectorDeRasgosWPT(P26,nivel);
SxtrainWPT(37,:)=CalcularVectorDeRasgosWPT(P27,nivel);
SxtrainWPT(38,:)=CalcularVectorDeRasgosWPT(P28,nivel);
SxtrainWPT(39,:)=CalcularVectorDeRasgosWPT(P29,nivel);
SxtrainWPT(40,:)=CalcularVectorDeRasgosWPT(P30,nivel);
SxtrainWPT(41,:)=CalcularVectorDeRasgosWPT(P31,nivel);
SxtrainWPT(42,:)=CalcularVectorDeRasgosWPT(P32,nivel);
SxtrainWPT(43,:)=CalcularVectorDeRasgosWPT(P33,nivel);
SxtrainWPT(44,:)=CalcularVectorDeRasgosWPT(P34,nivel);
SxtrainWPT(45,:)=CalcularVectorDeRasgosWPT(P35,nivel);
SxtrainWPT(46,:)=CalcularVectorDeRasgosWPT(P36,nivel);
SxtrainWPT(47,:)=CalcularVectorDeRasgosWPT(P37,nivel);
SxtrainWPT(48,:)=CalcularVectorDeRasgosWPT(P38,nivel);
SxtrainWPT(49,:)=CalcularVectorDeRasgosWPT(P39,nivel);
SxtrainWPT(50,:)=CalcularVectorDeRasgosWPT(P40,nivel);
SxtrainWPT(51,:)=CalcularVectorDeRasgosWPT(P41,nivel);
SxtrainWPT(52,:)=CalcularVectorDeRasgosWPT(P42,nivel);
SxtrainWPT(53,:)=CalcularVectorDeRasgosWPT(P43,nivel);
SxtrainWPT(54,:)=CalcularVectorDeRasgosWPT(P44,nivel);
SxtrainWPT(55,:)=CalcularVectorDeRasgosWPT(P45,nivel);
SxtrainWPT(56,:)=CalcularVectorDeRasgosWPT(P46,nivel);
```

```

SxtrainWPT(57,:)=CalcularVectorDeRasgosWPT(P47,nivel);
SxtrainWPT(58,:)=CalcularVectorDeRasgosWPT(P48,nivel);
SxtrainWPT(59,:)=CalcularVectorDeRasgosWPT(P49,nivel);
SxtrainWPT(60,:)=CalcularVectorDeRasgosWPT(P50,nivel);
SxtrainWPT(61,:)=CalcularVectorDeRasgosWPT(P51,nivel);
SxtrainWPT(62,:)=CalcularVectorDeRasgosWPT(P52,nivel);
SxtrainWPT(63,:)=CalcularVectorDeRasgosWPT(P53,nivel);
SxtrainWPT(64,:)=CalcularVectorDeRasgosWPT(P54,nivel);
SxtrainWPT(65,:)=CalcularVectorDeRasgosWPT(P55,nivel);
SxtrainWPT(66,:)=CalcularVectorDeRasgosWPT(P56,nivel);
SxtrainWPT(67,:)=CalcularVectorDeRasgosWPT(P57,nivel);
SxtrainWPT(68,:)=CalcularVectorDeRasgosWPT(P58,nivel);
SxtrainWPT(69,:)=CalcularVectorDeRasgosWPT(P59,nivel);
SxtrainWPT(70,:)=CalcularVectorDeRasgosWPT(P60,nivel);
SxtrainWPT(71,:)=CalcularVectorDeRasgosWPT(P61,nivel);
SxtrainWPT(72,:)=CalcularVectorDeRasgosWPT(P62,nivel);
SxtrainWPT(73,:)=CalcularVectorDeRasgosWPT(P63,nivel);
SxtrainWPT(74,:)=CalcularVectorDeRasgosWPT(P64,nivel);
SxtrainWPT(75,:)=CalcularVectorDeRasgosWPT(P65,nivel);
SxtrainWPT(76,:)=CalcularVectorDeRasgosWPT(P66,nivel);
SxtrainWPT(77,:)=CalcularVectorDeRasgosWPT(P67,nivel);
SxtrainWPT(78,:)=CalcularVectorDeRasgosWPT(P68,nivel);
SxtrainWPT(79,:)=CalcularVectorDeRasgosWPT(P69,nivel);
SxtrainWPT(80,:)=CalcularVectorDeRasgosWPT(P70,nivel);
SxtrainWPT(81,:)=CalcularVectorDeRasgosWPT(P71,nivel);
SxtrainWPT(82,:)=CalcularVectorDeRasgosWPT(P72,nivel);
SxtrainWPT(83,:)=CalcularVectorDeRasgosWPT(P73,nivel);
SxtrainWPT(84,:)=CalcularVectorDeRasgosWPT(P74,nivel);
SxtrainWPT(85,:)=CalcularVectorDeRasgosWPT(P75,nivel);
SxtrainWPT(86,:)=CalcularVectorDeRasgosWPT(P76,nivel);
SxtrainWPT(87,:)=CalcularVectorDeRasgosWPT(P77,nivel);
SxtrainWPT(88,:)=CalcularVectorDeRasgosWPT(P78,nivel);
SxtrainWPT(89,:)=CalcularVectorDeRasgosWPT(P79,nivel);
SxtrainWPT(90,:)=CalcularVectorDeRasgosWPT(P80,nivel);
SxtrainWPT(91,:)=CalcularVectorDeRasgosWPT(P81,nivel);
SxtrainWPT(92,:)=CalcularVectorDeRasgosWPT(P82,nivel);
SxtrainWPT(93,:)=CalcularVectorDeRasgosWPT(P83,nivel);
SxtrainWPT(94,:)=CalcularVectorDeRasgosWPT(P84,nivel);
SxtrainWPT(95,:)=CalcularVectorDeRasgosWPT(P85,nivel);
SxtrainWPT(96,:)=CalcularVectorDeRasgosWPT(P86,nivel);
SxtrainWPT(97,:)=CalcularVectorDeRasgosWPT(P87,nivel);
SxtrainWPT(98,:)=CalcularVectorDeRasgosWPT(P88,nivel);
SxtrainWPT(99,:)=CalcularVectorDeRasgosWPT(P89,nivel);
SxtrainWPT(100,:)=CalcularVectorDeRasgosWPT(P90,nivel);
SxtrainWPT(101,:)=CalcularVectorDeRasgosWPT(P91,nivel);
SxtrainWPT(102,:)=CalcularVectorDeRasgosWPT(P92,nivel);
SxtrainWPT(103,:)=CalcularVectorDeRasgosWPT(P93,nivel);
SxtrainWPT(104,:)=CalcularVectorDeRasgosWPT(P94,nivel);
SxtrainWPT(105,:)=CalcularVectorDeRasgosWPT(P95,nivel);
SxtrainWPT(106,:)=CalcularVectorDeRasgosWPT(P96,nivel);
SxtrainWPT(107,:)=CalcularVectorDeRasgosWPT(P97,nivel);
SxtrainWPT(108,:)=CalcularVectorDeRasgosWPT(P98,nivel);
SxtrainWPT(109,:)=CalcularVectorDeRasgosWPT(P99,nivel);
%Etiquetas para la clasificación
Sgrupo=[ones(30,1);-1*ones(79,1)];

```

APÉNDICE I

Construcción de las SVM para diferentes ensayos, con las variables originales, con las variables estandarizadas y con las variables transformadas.

```

clc
R=corrcoef(xtrainWPT);
[cpwpt,lambdawpt,explicacionwpt]=pcacov(R);

xtrainWPTstd=zscore(xtrainWPT);%matriz de entrenamiento estandarizada z =
(x-mean(x))./std(x)
Ztrain=xtrainWPTstd*cpwpt;%matriz de entrenamiento de variables
transformadas (rotadas)

explicacionwpt=explicacionwpt'
explicacionwptacumulada=cumsum(explicacionwpt)%. /sum(explicacionwpt)

disp(' ')
disp('con todas las variables originales')
disp(' ')
model= svmtrain(grupo,xtrainWPT,'-s 0 -t 2 -g 0.1 -c 0.01');
[salida, pre,ppp]= svmpredict(grupo,xtrainWPT,model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,SxtrainWPT,model);%supervision

Rs=corrcoef(SxtrainWPT);
[Scpwpt,Slambdawpt,Sexplicacionwpt]=pcacov(Rs);

SxtrainWPTstd=zscore(SxtrainWPT);%matriz de supervision estandarizada
Ztrains=SxtrainWPTstd*Scpwpt;%matriz de supervision de variables
transformadas (rotadas)

disp(' ')
disp('con todas las variables originales estandarizadas')
disp(' ')
model= svmtrain(grupo,xtrainWPTstd(:,2:4),'-s 0 -t 2 -g 0.01 -c 1');
[salida, pre,ppp]= svmpredict(grupo,xtrainWPTstd,model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,SxtrainWPTstd,model);%supervision

disp(' ')
disp('con todas las variables Z')
disp(' ')
model= svmtrain(grupo,Ztrain,'-s 0 -t 2 -g 0.1 -c 0.2'); % entrenamiento
[salida, pre,ppp]= svmpredict(grupo,Ztrain,model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,Ztrains,model);%supervision

disp(' ')
disp('con las variables Z reducidas, las primeras 5')
disp(' ')
model= svmtrain(grupo,Ztrain(:,1:5),'-s 0 -t 2 -g 0.1 -c 0.2')
[salida, pre,ppp]= svmpredict(grupo,Ztrain(:,1:5),model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,Ztrains(:,1:5),model);% supervision

```

```

disp(' ')
disp('con las variables Z reducidas, las primeras 4')
disp(' ')
model= svmtrain(grupo,Ztrain(:,1:4),'-s 0 -t 2 -g 0.1 -c 0.2'); %
entrenamiento
[salida, pre,ppp]= svmpredict(grupo,Ztrain(:,1:4),model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,Ztrains(:,1:4),model);% supervision

```

```

disp(' ')
disp('con las variables Z reducidas, las primeras 3')
disp(' ')
model= svmtrain(grupo,Ztrain(:,1:3),'-s 0 -t 2 -g 0.1 -c 0.1')
[salida, pre,ppp]= svmpredict(grupo,Ztrain(:,1:3),model);% validacion
[salida, pre,ppp]= svmpredict(Sgrupo,Ztrains(:,1:3),model);% supervision

```