



## **Tesis doctoral**

Una aportación a los sistemas de procesamiento de la información basados  
en modelos neuronales pulsantes

Elena Cerezuela Escudero  
Sevilla, mayo de 2015



Departamento de Arquitectura y Tecnología de Computadores  
Escuela Técnica Superior de Ingeniería Informática  
Universidad de Sevilla

**Una aportación a los sistemas de procesamiento de la  
información basados en modelos neuronales pulsantes**

por  
**Elena Cerezuela Escudero**

PROPUESTA DE TESIS DOCTORAL  
PARA LA OBTENCIÓN DEL GRADO DE  
**DOCTOR INGENIERO EN INFORMÁTICA**  
SEVILLA, MAYO DE 2015

Directores:  
**Dr. Ángel Jiménez Fernández**  
**Dr. Gabriel Jiménez Moreno**



# UNIVERSIDAD DE SEVILLA

Memoria presentada para optar al grado de Doctor Ingeniero en Informática por la Universidad de Sevilla.

Autor: Elena Cerezuela Escudero

**Título: Una aportación a los sistemas de procesamiento de la información basados en modelos neuronales pulsantes**

Departamento: **Arquitectura y Tecnología de Computadores**

Vº Bº Directores

---

**Dr. Ángel Jiménez Fernández**  
Profesor Ayudante Doctor

---

**Dr. Gabriel Jiménez Moreno**  
Profesor Titular de Universidad

La autora

---

**Elena Cerezuela Escudero**  
Ingeniero Informático



## Agradecimientos

Muchas gracias a los directores de este trabajo por sus incontables enseñanzas, dedicación y sabios consejos; Ángel y Gabriel, sin vosotros este trabajo no hubiera sido posible.

Muchas gracias Rafa por apoyarme y acompañarme en todo momento.

Muchas gracias a mi madre por educarme en la inquietud por el conocimiento.

Muchas gracias a mis sobrinos, a mi hermana y a mi abuela por recordarme las cosas importantes de la vida, gracias a ellos no me rindo. Gracias a mis padrinos y a mis primas por ser tan buena familia.

Muchas gracias a todos mis compañeros del grupo de investigación *Robótica y Tecnología de Computadores*, que me recibieron con los brazos abiertos y me han ayudado en mi carrera como investigador: gracias Manu por guiarme, gracias Miguel Ángel por regalarme tu sabiduría, gracias Lourdes por tus consejos en el campo auditivo, gracias Alejandro por orientarme en el campo neuromórfico, gracias Javier por apoyarme, gracias Antonio por acercarme al campo de la fusión sensorial, gracias Manuel Rivas por ser un gran compañero, gracias Paco por contagiarme tu ilusión por investigar y gracias Antón por tus tan acertados consejos.

Muchas gracias al grupo de investigación *Robótica Aplicada* de la Universidad de Cádiz, liderado por Arturo, por la formación que he adquirido con ellos.

Muchas gracias a mis amigas por su cariño y confianza en mí.





# Índice de Contenido

1. Introducción.....	1
1.1. Motivación.....	3
1.2. Objetivos.....	5
1.3. Estructura de la tesis.....	7
2. Los sistemas neuroinspirados, justificación y antecedentes.....	9
2.1. Ingeniería Neuromórfica .....	11
2.2. <i>Address-Event-Representation</i> .....	19
2.3. Implementaciones de sistemas neuro-inspirados.....	23
2.4. Grupos de investigación neuromórficos .....	28
3. El sistema auditivo: modelos e implementaciones.....	31
3.1. Oír: El sentido de la audición .....	32
3.1.1. Región periférica del sistema auditivo .....	33
3.1.2. Región central del sistema auditivo.....	43
3.2. Psicoacústica.....	45
3.2.1. Umbrales absolutos y umbrales diferenciales .....	46
3.2.2. Enmascaramientos y bandas críticas .....	49
3.2.3. Frecuencia, tono y timbre.....	51
3.3. Modelos del sistema auditivo .....	52
3.3.1. Modelo ERB (Equivalent Rectangular Bandwidth).....	52
3.3.2. Modelo de Lyon .....	53
3.3.3. Modelos de Lyon y Katsiamis.....	56

3.3.4. Modelo de las células ciliadas internas .....	57
3.4. Sistemas electrónicos auditivos bioinspirados.....	58
3.4.1. Cócleas analógicas con diseño en cascada .....	60
3.4.2. Cócleas analógicas con diseño paralelo .....	63
3.4.3. Cócleas analógicas bidimensionales (2-D).....	64
3.4.4. Cócleas digitales.....	65
3.4.5. Resumen de las características de las cócleas artificiales previas ...	67
4. Monitorización de spikes.....	69
4.1. Monitor de spikes masivo.....	71
4.2. Monitor de spikes distribuido .....	75
4.3. Escenario experimental .....	78
4.3.1. Placas usadas en el experimento: USB-AER y USB-AERmini2 ....	79
4.3.2. Excitación de los monitores y captura de los eventos generados ....	81
4.3.3. Generación de los spikes y procesamiento de la salida .....	82
4.4. Resultados experimentales .....	83
4.5. Consumo hardware .....	87
5. Sistema Neuromórfico de Audición .....	91
5.1. Arquitectura general del sistema neuromórfico de audición .....	93
5.2. Implementación del NAS estéreo con 64 canales.....	97
5.2.1. Sintonización del banco de filtros de spikes.....	97
5.2.1. Sintonización del generador de spikes reverse bit-wise .....	100
5.2.2. Sintonización del divisor de spikes basado en el método reverse bit-wise	104
5.2.3. Sintonización del monitor distribuido de spikes .....	105
5.2.4. Síntesis de NAS.....	106

5.3.	Escenario experimental del NAS.....	109
5.4.	Resultados experimentales del NAS implementado.....	111
5.4.1.	Respuesta temporal.....	111
5.4.2.	Características en frecuencia.....	113
5.4.3.	Rango dinámico.....	119
5.5.	Normalización de la primera versión del NAS.....	120
5.5.1.	Sintonización de los divisores de spikes.....	122
5.5.2.	Resultados experimentales del NAS normalizado.....	123
5.6.	Análisis de la tasa de eventos del NAS para diferentes configuraciones en el generador de spikes.....	129
5.7.	Comparativa del NAS con los sistemas previos.....	132
6.	Sistemas de reconocimiento automático de sonido.....	133
6.1.	Modelo estadístico: Modelo Oculto de Markov.....	134
6.2.	Redes neuronales artificiales.....	136
6.2.1.	Redes neuronales de información muestreada.....	138
6.2.2.	Redes neuronales pulsantes.....	145
6.2.3.	Redes Neuronales de Convolución.....	147
6.3.	Sistemas de reconocimiento de sonidos.....	151
7.	Reconocimiento de sonidos.....	155
7.1.	Sistema de identificación de la frecuencia de un motor.....	156
7.1.1.	Escenario del experimento para el reconocimiento del sonido generado por un motor.....	156
7.1.2.	Sistema de identificación de la frecuencia del motor.....	158
7.1.3.	Experimento y sus resultados.....	160
7.2.	Sistema de clasificación de sonidos mediante redes neuronales de tasas	161

7.2.1. Clasificación de tonos puros mediante red neuronal MLP .....	164
7.2.2. Identificación de notas musicales mediante red neuronal MLP ....	170
7.3. Sistema de reconocimiento de sonidos mediante redes neuronales de convolución .....	175
7.3.1. Arquitectura del sistema de convolución .....	178
7.3.2. Entrenamiento de la red de convolución .....	180
7.3.3. Síntesis del sistema de convolución .....	181
7.3.4. Escenario experimental .....	183
7.3.5. Resultado de los experimentos .....	188
7.4. Comparativa de los sistemas de reconocimiento expuestos .....	192
8. Conclusiones .....	195
9. Trabajo futuro .....	197
10. Bibliografía .....	201
11. Anexo: componentes hardwares y resultados .....	217
11.1. USB-AER .....	217
11.2. USB-AERmini2 .....	219
11.3. Virtex-5 FXT FPGA ML507 .....	222
11.4. Resultados de los experimentos de evaluación de los sistemas de reconocimiento para tonos puros .....	224

# Índice de Figuras

<i>Figura 2.1. Reproducción de una lámina ilustrada de Ramón y Cajal del cerebro de un ave</i>	13
<i>Figura 2.2. A) Neurona de los grabados de Ramón y Cajal en la que están indicadas las partes de la neurona y esquema en el que se representa el potencial de acción o spike. B) Señal transmitida desde la neurona j a la neurona i. La sinapsis está marcada por un círculo. Imagen tomada de (Gerstner et al. 2002).</i>	14
<i>Figura 2.3. Diagrama de un potencial de acción, o spike, generado por una neurona</i>	15
<i>Figura 2.4. Esquema de la transmisión de la información pulsante mediante el uso de la representación AER.</i>	20
<i>Figura 2.5. Especificación de pines de los conectores y cables AER del proyecto CAVIAR. Imagen tomada de (Häfliger 2007).</i>	22
<i>Figura 2.6. Fases del protocolo Handshake. Las restricciones de tiempo se muestran en la Tabla 2.2. Imagen tomada de (Häfliger 2007).</i>	23
<i>Figura 2.7. Retina artificial diferencial espacio-temporal. Imagen tomada de (Patrick Lichtsteiner et al. 2008).</i>	25
<i>Figura 2.8. (a) Placa de prototipo que incorpora el chip AEREAR2.(b) Cocleograma ante la señal de voz: " The quick red fox jumped over the lazy dog". Imagen tomada de (Liu et al. 2010).</i>	26
<i>Figura 3.1. Anatomía del oído. Imagen tomada de (McConnell &amp; Hull 2010).</i>	34
<i>Figura 3.2. A) Anatomía del oído medio e interno; B) sección transversal de la cóclea. Imagen tomada de (McConnell &amp; Hull 2010).</i>	37
<i>Figura 3.3. Estructura interna del órgano de Corti</i>	38
<i>Figura 3.4. Efecto de las ondas sonoras sobre las estructuras del oído. Imagen tomada de (Thibodeau &amp; Patton 2012)</i>	39
<i>Figura 3.5. Organización tonotópica de la cóclea. A) Distribución tonotópica de la cóclea. B) Localización de la respuesta coclear a altas frecuencias. C) Localización de la respuesta coclear a frecuencias medias. D) Localización de la respuesta coclear a bajas frecuencias.</i>	40
<i>Figura 3.6. Efectos de la no linealidad del comportamiento de la membrana basilar. (a) Respuesta de un punto de la membrana basilar son el efecto de los OHCs (Passive) y con</i>	

<i>OHCs (Active). CF indica la característica de frecuencia de la sección de la membrana basilar. (b) Nivel de respuesta en función del nivel de entrada de la frecuencia característica de la sección de la membrana basilar. (Hamilton 2008).</i> .....	42
<i>Figura 3.7. Vías Auditivas y corteza auditiva primaria. Imagen tomada de (McConnell &amp; Hull 2010)</i> .....	44
<i>Figura 3.8. Curva de Wegel. Curva en la que se representan los umbrales de audición respecto a la frecuencia e intensidad del sonido</i> .....	48
<i>Figura 3.9. Curvas de igual sonoridad de Fletcher-Munson</i> .....	49
<i>Figura 3.10. Representación de la medida ERB (Miró-Amarante 2013)</i> .....	53
<i>Figura 3.11. Diagrama de bloques de los filtros en el modelo de Lyon</i> .....	55
<i>Figura 3.12. Función de transferencia de los filtros usados en el banco de filtros. Imagen tomada de (Lyon 1982)</i> .....	55
<i>Figura 3.13. Respuesta en frecuencia del modelo de Lyon (64 secciones) para frecuencias características: 3.0, 2.0, 1.0, 0.6 y 0.3 KHz (Miró-Amarante 2013)</i> .....	56
<i>Figura 3.14. Representación gráfica del banco de filtros en arquitectura de Lyon- Katsiamis y en arquitectura en cascada. Las frecuencias de corte de las diferentes etapas (en el gráfico taps) están distribuidas exponencialmente. Imagen tomada de (Katsiamis et al. 2007).</i> .....	57
<i>Figura 3.15. Árbol de la evolución histórica de las cócleas artificiales. Figura tomada de (Hamilton 2008) y modificada en el contexto de este trabajo</i> .....	59
<i>Figura 3.16. Estructura en cascada de los filtros que forman a la cóclea artificial de Lyon y Mead de 100 etapas (Lyon &amp; Mead 1988)</i> .....	60
<i>Figura 3.17. Respuesta frecuencial de los filtros de la cóclea artificial de Lyon y en (b) la versión mejorada de Watts et.al. (Watts et al. 1992).</i> .....	61
<i>Figura 3.18. Distribución de las frecuencias de corte (Hz) para la cóclea artificial de Watts et. al. (izquierda) y de van Schaik (derecha) (van Schaik et al. 1996).</i> .....	62
<i>Figura 3.19. Respuesta de un filtro de segundo orden, en el modelo de cóclea paralelo (a) y en el modelo de cóclea en cascada (b)</i> .....	63
<i>Figura 3.20. Diagrama de bloques de la cóclea digital propuesta por Summerfield et.al. Imagen tomada de (Summerfield &amp; Lyon 1992)</i> .....	65
<i>Figura 3.21. Respuesta frecuencial de la implementación coclear de Leong et. al. Imagen tomada de (Leong et al. 2003).</i> .....	66
<i>Figura 3.22. Filtros en cascada con salida cada 4 secciones de la cóclea digital presentada en (Mugliette et al. 2011). Imagen tomada de (Mugliette et al. 2011)</i> .....	67

<i>Figura 4.1. Escenario típico de uso de un monitor de spikes .....</i>	<i>70</i>
<i>Figura 4.2. Arquitectura y funcionamiento del monitor masivo de spikes.....</i>	<i>73</i>
<i>Figura 4.3. Arquitectura y funcionamiento del monitor de spikes distribuido.....</i>	<i>77</i>
<i>Figura 4.4. Fases de los experimentos .....</i>	<i>79</i>
<i>Figura 4.5. Fotografía del escenario experimental .....</i>	<i>81</i>
<i>Figura 4.6. Componentes de la placa USB-AER para la evaluación de los monitores de spikes .....</i>	<i>82</i>
<i>Figura 4.7. Tasa de eventos de salida del monitor masivo (arriba) y del monitor distribuido (abajo).....</i>	<i>85</i>
<i>Figura 4.8. Porcentaje de spikes perdidos para el monitor masivo (arriba) y distribuido (abajo).....</i>	<i>86</i>
<i>Figura 4.9. Slices usados por cada monitor y recta de regresión para cada monitor .....</i>	<i>88</i>
<i>Figura 5.1. Forma de procesar la información de los distintos tipos de cócleas.....</i>	<i>92</i>
<i>Figura 5.2. Arquitectura del NAS .....</i>	<i>96</i>
<i>Figura 5.3. Gráfica que muestra la frecuencia central de cada banda deseada y obtenida .....</i>	<i>100</i>
<i>Figura 5.4. Diagrama de bloques del generador Reverse bit-wise .....</i>	<i>103</i>
<i>Figura 5.5. Arquitectura del divisor de spikes bit-wise .....</i>	<i>105</i>
<i>Figura 5.6. Estructura de las direcciones AER de salida del NAS .....</i>	<i>107</i>
<i>Figura 5.7. Relación entre los requisitos lógicos y el tamaño del banco de filtros.....</i>	<i>109</i>
<i>Figura 5.8. Esquema de conexión de los dispositivos del escenario.....</i>	<i>110</i>
<i>Figura 5.9. Fotografía del escenario experimental .....</i>	<i>111</i>
<i>Figura 5.10. Cocleograma del NAS de 64 canales en presencia de voz humana .....</i>	<i>112</i>
<i>Figura 5.11. Sonograma del NAS izquierdo de 64 canales ante voz habla humana .....</i>	<i>113</i>
<i>Figura 5.12. Diagrama de Bode del banco de filtros.....</i>	<i>114</i>
<i>Figura 5.13. Representación en superficie de la respuesta en frecuencia del NAS izquierdo .....</i>	<i>115</i>
<i>Figura 5.14. Frecuencia media y tasa máxima de los eventos de los 64 canales del NAS izquierdo (arriba) y derecho (abajo) .....</i>	<i>117</i>
<i>Figura 5.15. Factor de calidad de cada banda del NAS izquierdo.....</i>	<i>118</i>
<i>Figura 5.16. Ancho de banda de cada banda del NAS izquierdo .....</i>	<i>118</i>
<i>Figura 5.17. Tasa de eventos resultante ante la excitación del NAS para una secuencia de ruido blanco con diferente volumen .....</i>	<i>119</i>
<i>Figura 5.18. Diagrama de bloques de un filtro paso de banda del NAS.....</i>	<i>120</i>

<i>Figura 5.19. Valores a usar para normalizar las ganancias de las bandas .....</i>	<i>123</i>
<i>Figura 5.20. Cocleograma de un tono puro a 440Hz en NASv1 (arriba) y NASv2 (abajo)..</i>	<i>125</i>
<i>Figura 5.21. Diagrama de bode de NASv2 izquierdo (arriba) y derecho (abajo) .....</i>	<i>126</i>
<i>Figura 5.22. Tasa de eventos en representación en superficie .....</i>	<i>127</i>
<i>Figura 5.23. Ganancias máximas relativas de cada banda del NASv2 izquierdo.....</i>	<i>128</i>
<i>Figura 5.24. Rango dinámico del NASv2 .....</i>	<i>129</i>
<i>Figura 5.25. Diagrama de Bode de NAS con generador al 001Eh.....</i>	<i>131</i>
<i>Figura 5.26. Ganancias máximas de cada banda para NASv2 y NASv3 .....</i>	<i>131</i>
<i>Figura 6.1. Cadena de un ejemplo de HMM de 5 estados (etiquetados <math>S_1</math> a <math>S_5</math>) con las correspondientes transiciones entre los estados con sus probabilidades (<math>a_{ij}</math>, siendo <math>i</math> el estado de origen de la transición y <math>j</math> el estado destino de la transición). Imagen tomada de (Rabiner 1989). .....</i>	<i>135</i>
<i>Figura 6.2. Ejemplo de perceptrón Multicapa con una única capa oculta (<math>N_h</math>) .....</i>	<i>139</i>
<i>Figura 6.3. Esquema de neurona artificial del perceptrón.....</i>	<i>141</i>
<i>Figura 6.4. Esquema del modelo de retropropagación de errores .....</i>	<i>144</i>
<i>Figura 6.5. Ecuaciones que describen la operación de las neuronas del modelo Integra-y-dispara con fugas. Imagen tomada de (Cassidy et al. 2013). .....</i>	<i>147</i>
<i>Figura 6.6. Esquema de funcionamiento de la operación de convolución para frames digitales .....</i>	<i>150</i>
<i>Figura 7.1. Diagrama de bloques representativo del sistema de estimación de la frecuencia de un motor DC mediante los sensores NASv1 y DVS128 .....</i>	<i>157</i>
<i>Figura 7.2. Escenario del experimento de identificación de la frecuencia de un motor DC mediante NAS y DVS128 .....</i>	<i>158</i>
<i>Figura 7.3. Relación entre la frecuencia del motor DC y la tasa de eventos generada por NAS izquierdo y derecho, respectivamente.....</i>	<i>159</i>
<i>Figura 7.4. Componentes que intervienen en los experimentos de redes neuronales de tasas.....</i>	<i>163</i>
<i>Figura 7.5. Datos del entrenamiento de la red neuronal tradicional.....</i>	<i>166</i>
<i>Figura 7.6. Tasa de acierto del sistema de reconocimiento de tonos puros en presencia de ruido blanco, para el NASv1 (arriba) y el NASv2 (abajo) .....</i>	<i>169</i>
<i>Figura 7.7. Tasa de eventos para cada canal del NASv2 cada 20ms .....</i>	<i>171</i>
<i>Figura 7.8. Tasa de eventos para tres notas F3 del piano cada 0.4 segundos.....</i>	<i>173</i>
<i>Figura 7.9. Tasa de eventos para la nota F3 de la flauta virtual cada 20ms .....</i>	<i>174</i>
<i>Figura 7.10. Arquitectura de una neurona de la red de convolución.....</i>	<i>177</i>



<i>Figura 7.11. Arquitectura de la red neuronal de convolución</i> .....	179
<i>Figura 7.12. Diagrama de flujo de una neurona de convolución</i> .....	179
<i>Figura 7.13. Bits necesarios para codificar los spikes del NAS</i> .....	181
<i>Figura 7.14. Relación entre el número de neuronas de convolución y el número de slices necesarios</i> .....	183
<i>Figura 7.15: Fotografía del escenario experimental del sistema de reconocimientos de sonidos</i> .....	184
<i>Figura 7.16. Diagrama de bloques de la fase de entrenamiento del sistema de reconocimiento basado en una red de convolución</i> .....	185
<i>Figura 7.17. Diagrama de bloques de la fases de ejecución del sistema de reconocimiento basado en una red de convolución</i> .....	186
<i>Figura 7.18. Valores de los núcleos de convolución para cada tono puro respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2 (derecha)</i> .....	187
<i>Figura 7.19. Valores de los núcleos de convolución para cada nota de piano respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2</i> .....	188
<i>Figura 7.20. Valores de los núcleos de convolución para cada nota de piano respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2</i> .....	188
<i>Figura 7.21. Tasa de acierto de la ConvNet para el reconocimiento de tonos puros en presencia de ruido blanco, para el NASv1 (arriba) y el NASv2 (abajo)</i> .....	190
<i>Figura 11.1. Fotografía de la placa USB-AER</i> .....	218
<i>Figura 11.2. Fotografía de la tarjeta USBAERmini2, destacando los puertos que tiene</i> ....	220
<i>Figura 11.3. Diagrama de bloque de la USBAERMini2</i> .....	221
<i>Figura 11.4. Kit de evaluación ML507 de Xilinx con la FPGA Virtex-5</i> .....	223

# Índice de Ecuaciones

<i>Ecuación 4.1</i> .....	74
<i>Ecuación 4.2</i> .....	75
<i>Ecuación 4.3</i> .....	76
<i>Ecuación 4.4</i> .....	76
<i>Ecuación 4.5</i> .....	83
<i>Ecuación 4.6</i> .....	87
<i>Ecuación 5.1</i> .....	98
<i>Ecuación 5.2</i> .....	98
<i>Ecuación 5.3</i> .....	99
<i>Ecuación 5.4</i> .....	100
<i>Ecuación 5.5</i> .....	103
<i>Ecuación 5.6</i> .....	105
<i>Ecuación 5.7</i> .....	106
<i>Ecuación 5.8</i> .....	109
<i>Ecuación 5.9</i> .....	121
<i>Ecuación 5.10</i> .....	121
<i>Ecuación 5.11</i> .....	130
<i>Ecuación 6.1</i> .....	148
<i>Ecuación 6.2</i> .....	148
<i>Ecuación 6.3</i> .....	149
<i>Ecuación 7.1</i> .....	159
<i>Ecuación 7.2</i> .....	159
<i>Ecuación 7.3</i> .....	160
<i>Ecuación 7.4</i> .....	176
<i>Ecuación 7.5</i> .....	176
<i>Ecuación 7.6</i> .....	182

# Índice de Tablas

<i>Tabla 2.1. Comparativa cualitativa entre un computador y el sistema nervioso .....</i>	<i>18</i>
<i>Tabla 2.2. Requisitos de tiempo de las cuatro fases del protocolo Handshake de la Figura 2.6. Imagen tomada de (Häfliger 2007).....</i>	<i>23</i>
<i>Tabla 3.1. Resumen de características de cócleas analógicas .....</i>	<i>68</i>
<i>Tabla 3.2. Resumen de características de cócleas digitales.....</i>	<i>68</i>
<i>Tabla 4.1. Ejemplo de espacio de direcciones de los monitores .....</i>	<i>74</i>
<i>Tabla 4.2. Consumo hardware en slices del monitor masivo y distribuido .....</i>	<i>88</i>
<i>Tabla 5.1. Frecuencias centrales de cada banda del NAS .....</i>	<i>99</i>
<i>Tabla 5.2. Secuencia de emisión de spikes de un generador exhaustivo bit-wise de 3 bits</i>	<i>102</i>
<i>Tabla 5.3. Requisitos hardware para diferentes números de canales del NAS.....</i>	<i>108</i>
<i>Tabla 5.4. Resumen de las características del NAS.....</i>	<i>132</i>
<i>Tabla 6.1. Principales tipos de función de activación de las neuronas artificiales.....</i>	<i>140</i>
<i>Tabla 6.2. Resumen de las características de los sistemas de reconocimiento de sonidos expuestos.....</i>	<i>154</i>
<i>Tabla 7.1. Frecuencias caraterísticas de los tonos puros a identificar.....</i>	<i>164</i>
<i>Tabla 7.2. Resultados de la red neuronal de 10 neuronas en la capa oculta para reconocer 8 tonos puros usando el NASv1.....</i>	<i>166</i>
<i>Tabla 7.3. Resultados de la red neuronal de 10 neuronas en la capa oculta para reconocer 8 tonos puros usando el NASv2.....</i>	<i>167</i>
<i>Tabla 7.4. Resumen de la tasa de aciertos del sistema de reconocimiento de tonos puros usando el NASv1 .....</i>	<i>169</i>
<i>Tabla 7.5. Resumen de la tasa de aciertos del sistema de reconocimiento de tonos puros usando el NASv2 .....</i>	<i>170</i>
<i>Tabla 7.6. % de aciertos en la clasificación para la red de reconocimiento de 4 notas del piano electrónico, calculando las muestras cada 20ms.....</i>	<i>172</i>
<i>Tabla 7.7. % de fallos de clasificadas en la red de reconocimiento de patrones para las 4 notas del piano virtual, realizando el cálculo de las muestras en distintos periodos de tiempo.....</i>	<i>173</i>
<i>Tabla 7.8. % de fallos de clasificadas en la red de reconocimiento de patrones para las 4 notas de la flauta, usando distintos tamaños en la capa oculta .....</i>	<i>174</i>

<i>Tabla 7.9. Requisitos hardware de la red de convolución para diferentes números de neuronas .....</i>	<i>182</i>
<i>Tabla 7.10. Relación entre la frecuencia fundamental de los tonos a reconocer y las notas musicales .....</i>	<i>186</i>
<i>Tabla 7.11. Frecuencia fundamental de las notas musicales F3, F4, F5y F6 .....</i>	<i>187</i>
<i>Tabla 7.12. Resumen de la tasa de aciertos de la ConvNet para el reconocimiento de tonos puros usando el NASv1 .....</i>	<i>190</i>
<i>Tabla 7.13. Resumen de la tasa de aciertos de la ConvNet para el reconocimiento de tonos puros usando el NASv2 .....</i>	<i>191</i>
<i>Tabla 7.14. Porcentaje de aciertos del experimento de clasificación de notas musicales de piano y flauta.....</i>	<i>191</i>
<i>Tabla 7.15. Tasa de aciertos del experimento de clasificación de notas musicales de piano con dos capas de neuronas.....</i>	<i>192</i>
<i>Tabla 7.16. Resultados de la ConvNet y de la red MLP para el reconocimiento de tonos puros en presencia de ruido blanco .....</i>	<i>193</i>
<i>Tabla 7.17. Resultados de la ConvNet y de la red MLP para el reconocimiento de notas musicales .....</i>	<i>193</i>
<i>Tabla 11.1. Porcentaje de aciertos del sistema de reconocimiento de tonos puros mediante la red neuronal MLP usando el NASv1 .....</i>	<i>224</i>
<i>Tabla 11.2. Porcentaje de aciertos del sistema de reconocimiento de tonos puros mediante la red neuronal MLP usando el NASv2.....</i>	<i>225</i>
<i>Tabla 11.3. Tasa de aciertos (en tanto por 1) del sistema de reconocimiento de tonos puros ConvNet usando el NASv1.....</i>	<i>226</i>
<i>Tabla 11.4. Tasa de aciertos (en tanto por 1) del sistema de reconocimiento de tonos puros ConvNet usando el NASv2.....</i>	<i>227</i>

## Resumen

En este trabajo se propone e implementa un nuevo sistema de reconocimiento de sonidos basado en una cóclea artificial pulsante innovadora.

Inicialmente se estudian los mecanismos clásicos de procesamiento de audio para el reconocimiento de sonidos; así como del funcionamiento biológico del sistema auditivo humano en conjunción con los procesos neuronales del cerebro. A partir de dichos estudios, es como se proponen nuevos sistemas en lo que respecta al procesamiento de audio y reconocimiento de sonidos automático.

En trabajos de investigación recientes se han desarrollado una serie de elementos neuromórficos<sup>1</sup> hardware pulsantes basados en codificación AER, en esta tesis estos bloques sirven de punto de partida para la implementación de nuestro sistema de reconocimiento de sonidos.

Se proponen e implementan dos sistemas: por una parte, una cóclea artificial para la obtención de las componentes de frecuencia del sonido, imitando el funcionamiento del aparato auditivo. Y por otra, un sistema de reconocimiento de patrones sonoros, obtenidos a partir de la salida generada por la cóclea artificial, inspirado en el comportamiento de las neuronas y las conexiones entre ellas.

Por último, en este trabajo, se realiza un estudio exhaustivo para evaluar la eficiencia de los sistemas implementados y compararlos con los desarrollos previos.

---

<sup>1</sup> La Ingeniería neuromórfica es la disciplina que aplica los mecanismos del sistema nervioso para la resolución de los problemas (descrita en el apartado 2.1 de este documento)





# 1. Introducción

*“Si no conozco una cosa, la investigaré”, Louis Pasteur*

Uno de los mayores retos de la humanidad se ha centrado en conseguir solucionar artificialmente determinados problemas que estaban solventados en la naturaleza, con el fin de poder utilizar dichos mecanismos para fines propios o para facilitar determinadas tareas más pesadas. En el último escalón de esta evolución de aprendizaje se sitúa la capacidad de dotar de cierta autonomía a elementos artificiales, con el fin de automatizar procesos sin la necesidad de la intervención del ser humano. Este proceso de automatización conlleva la investigación en un amplio abanico de campos, a lo largo de diversas materias.

Los sistemas neuro-inspirados<sup>2</sup> en el mundo animal se caracterizan por emular propiedades básicas del procesado sensorial. Así tenemos sistemas de visión, auditivos, táctiles, entre otros, que realizan funciones básicas con propiedades muy parecidas a las que podemos encontrar en la naturaleza.

Este trabajo está enfocado al procesado neuro-inspirados de la señal acústica, con el objeto de obtener un sistema de reconocimiento de sonidos con un gran rango dinámico, eficiente, automático e inmune al ruido. Este tipo de procesamiento se suele realizar en dos etapas: la primera etapa es de extracción de parámetros característicos para describir la señal acústica y la segunda fase consiste en identificar los sonidos respecto a una medida de similitud entre las características obtenidas en la fase anterior y unos modelos de referencia. El mundo animal resuelve estas dos etapas del reconocimiento de sonidos en tiempo

---

<sup>2</sup> Sistemas que tratan de resolver problemas comunes en la ingeniería mediante el uso de sistemas basados en la manera que el sistema nervioso codifica y procesa la información



real, teniendo éxito incluso en entornos cambiantes y altamente ruidosos, por lo tanto, vamos a desarrollar los sistemas necesarios para cada fase inspirándonos en la solución aportada por la biología. Para resolver la primera fase nos hemos centrado en las investigaciones sobre el desarrollo de modelos computacionales de la fisiología auditiva aplicados a cócleas artificiales. De esta forma, hemos desarrollado un innovador sensor neuromórfico<sup>3</sup> de audio que imita la funcionalidad de la cóclea biológica y la representación de la información en el nervio auditivo. Este sistema procesa la información mediante pulsos estrechos, usando como mecanismo de modulación el tiempo entre pulsos de ancho constante (Modulación por Frecuencia de Pulsos). Para la etapa de identificación de sonidos, nos hemos basado en las investigaciones sobre el reconocimiento automático de patrones mediante diversos modelos de redes neuronales. Hemos desarrollado dos sistemas de reconocimiento de patrones, uno de ellos basado en las redes neuronales artificiales de tasas y otro en las redes neuronales pulsantes. Estos sistemas de clasificación buscan patrones sobre las características frecuenciales de la información aportada por el sensor neuromórfico.

Las tareas más frecuentes de los sistemas de reconocimiento de sonidos son las siguientes: localizar la fuente del sonido (Chan et al. 2007), (Van Schaik et al. 2009), (Chan et al. 2012), determinar la naturaleza del sonido (Nielsen et al. 2006), (Jackel et al. 2010), (Qian & Nian 2007) e interpretar el significado de los sonidos (Guerrero-turrubiates et al. 2014), (Barbancho et al. 2012), (Miró-Amarante 2013), (Kim et al. 2009).

En este trabajo presentamos un sistema de reconocimiento de sonidos mediante el tono o las características frecuenciales del sonido. La determinación del tono de un sonido es un proceso útil tanto para determinar la naturaleza del sonido como para interpretar el significado de los sonidos. En tareas de localización se puede usar la identificación del tono en los casos en los que se quiera localizar la fuente en movimiento de un sonido, basándose en el efecto Doppler<sup>4</sup>.

---

<sup>3</sup> La Ingeniería neuromórfica trata de resolver problemas comunes en la ingeniería mediante el uso de sistemas basados en la manera que el sistema nervioso codifica y procesa la información.

<sup>4</sup> El efecto Doppler, llamado así por el físico austriaco Christian Andreas Doppler, es el aparente cambio de frecuencia de una onda producido por el movimiento relativo de la fuente respecto a su observador.

## 1.1. Motivación

Una cuestión pendiente de resolver con respecto al sistema sensorial y nervioso de los seres vivos es cómo se consigue tal grado de eficiencia usando, en principio y casi exclusivamente, un modelo de la información basado en señales pulsantes<sup>5</sup>, de hecho, se desconoce el mecanismo de codificación utilizado. En este trabajo nos planteamos hacer uso de señales pulsantes bio-inspiradas basadas en la Modulación por Frecuencia de Pulsos (PFM), comúnmente utilizadas en la ingeniería neuromórfica, para resolver problemas en otros ámbitos de la ingeniería. Es evidente que no vamos a imitar de forma fidedigna al sistema nervioso de los seres vivos, pero el uso de sistemas neuronales pulsantes artificiales sí nos pueden ir mostrando ciertas características que nos irán acercando a dicho conocimiento de la naturaleza. Por otra parte, los nuevos mecanismos y sistemas que podamos obtener con la tecnología pulsante neuro-inspirada pueden representar ciertas ventajas tecnológicas en diversas aplicaciones prácticas. En definitiva esta es la doble motivación que ha guiado siempre la investigación científica: conocer la naturaleza y usar dicho conocimiento para el avance tecnológico.

La idea básica de los sistemas pulsantes es representar la información de forma pulsante y después actuar sobre un flujo de pulsos para, “simplemente quitando y/o poniendo pulsos”, procesar la información. El término “simplemente” debe entenderse en el sentido de que las operaciones a utilizar son sencillas: quitar y poner pulsos, pero no implica que sea evidente cuales, o cuantos, debemos quitar o poner.

Los sistemas de reconocimiento de sonidos son necesarios en una amplia variedad de aplicaciones como la transcripción musical, diferenciación entre instrumentos y la voz en canciones, catalogación, reconocimiento del habla, del hablante de idiomas, control de calidad. La determinación del tono de un sonido también tiene aplicaciones en los implantes cocleares.

---

<sup>5</sup> Potenciales de acción de las neuronas, se explica en detalle en el capítulo 2

El reconocimiento automático de sonidos es una tarea inherentemente difícil debido a la variabilidad de las señales acústicas. Si la señal se registra en condiciones favorables, se consiguen muy buenas prestaciones en el proceso de reconocimiento. Sin embargo, cuando el sistema funciona en situaciones reales se encuentra con condiciones adversas motivadas fundamentalmente por cambios en el entorno acústico (ruidos, reverberación y ecos) o eléctrico (ruido o distorsiones de la señal provocados por el micrófono o el canal de transmisión). Otro factor que hace el reconocimiento de sonidos una tarea compleja es que hace falta un amplio rango dinámico de sonoridad para separar los sonidos en sus características de frecuencia y su estructura temporal. Estas características forman tramas complejas, semejantes entre sí y con ruido, que son difíciles de categorizar. Las soluciones bio-inspiradas se enfrentan a estos problemas de forma exitosa, porque las cócleas biológicas aportan un amplio rango dinámico e inmunidad al ruido y las redes neuronales pulsantes tienen buenos resultados en reconocimiento de patrones entre tramas semejantes entre sí.

El reconocimiento de sonidos, evidentemente, puede ser resuelto con la tecnología actual de diversas formas, básicamente podemos utilizar tecnología digital en plataformas software o en hardware, analógica, o mezcla de ambas. En este trabajo se intentan resolver estos problemas llevando la utilización de los sistemas pulsantes al extremo máximo. Pero lo sorprendente es que los sistemas resultantes son relativamente simples, obteniéndose además una forma de proceder, o metodología, que puede ser de aplicación a otros problemas diferentes a los dos planteados.

La primera decisión a tomar para intentar resolver mediante el procesado de pulsos estos problemas es el modo de codificar la información. En diversos trabajos (Maass & Bishop 1999), (Thorpe et al. 2010) se recogen numerosas formas de codificar la información para su uso en sistemas neuro-inspirados. En este trabajo nos hemos decantado por el mecanismo clásico de PFM con pulsos estrechos, por ser el más simple y con mayores posibilidades de procesado. En dicho mecanismo de modulación se representa la información mediante el tiempo entre pulsos o frecuencia de los mismos, de esta forma una señal analógica se

puede representar, por ejemplo, asociando su amplitud en cada instante a una frecuencia de pulsos. Un aspecto importante es la fidelidad de la representación pulsante PFM con respecto a la señal origen analógica, respuesta a esto la podemos encontrar en (Morgado-Estevez 2004). En general tenemos que las componentes en frecuencia de la señal analógica que se pretenda representar y el índice de modulación son parámetros importantes en este contexto, las componentes de alta frecuencia de la señal analógica, que además sean de amplitud baja, serán difícilmente representables mediante pulsos. En (Morgado-Estevez 2004) podemos observar como la representación PFM no puede tener una frecuencia inferior a la de la señal que representa esto es similar al teorema del muestreo. En este trabajo se obvia estos problemas de representación y suponemos que siempre se tienen señales e índices de modulación correctos.

Para dotar al sistema de autonomía y que se pueda utilizar en plataformas robóticas, se decide trabajar con FPGAs para el desarrollo del sistema neuromórfico, tanto para los sistemas sensoriales como para los procesadores de la información.

## 1.2. Objetivos

Podemos señalar dos tipos de objetivos: unos generales, que pretenden avanzar en el estudio de los sistemas neuromórficos, y otros específicos, centrados en la resolución de problemas particulares y el diseño de sus correspondientes sistemas reales.

Objetivo General: Estudio y búsqueda de nuevas arquitecturas computacionales basadas en sistemas pulsantes, análogas a los sistemas neuronales, para la percepción y el procesamiento de la información auditiva.

- Estudiar como el sistema auditivo humano capta, analiza y codifica la información acústica en impulsos nerviosos para ser procesados por el cerebro.

- Adaptar los sistemas analógicos y digitales existentes de procesamiento de audio a sistemas basados en la representación pulsante. Como punto de partida se analizará las soluciones clásicas del campo auditivo.

Este objetivo general es común en los trabajos de ingeniería neuromórfica y, en realidad, es una doble vía: por una parte se intenta imitar al sistema nervioso para, sacar provecho de la evolución de la naturaleza, pero por otra parte, se pretende llegar a descubrir y entender nuevos aspectos del funcionamiento de los sistemas neuronales biológicos.

Para conseguir este objetivo general, tan amplio y ambicioso, se han fijado un conjunto de objetivos específicos concretos:

- Estudiar y diseñar un nuevo modelo de cóclea neuromórfica, basada en la cóclea biológica y en la representación pulsante de la información.
  - Estudio del sistema auditivo humano.
  - Análisis de los modelos físicos más relevantes el comportamiento de la cóclea humana.
- Estudiar el funcionamiento de las neuronas y la conexión entre ellas para analizar los diferentes mecanismos de codificación de la información de forma pulsante.
- Estudiar y desarrollar un sistema capaz de monitorizar la actividad de pulsos entre diferentes capas de neuronas. Este módulo permitirá la comunicación entre capas de neuronas artificiales y por lo tanto, entre los diferentes sistemas que se exponen en este trabajo.
- Estudiar y crear nuevos modelos neuronales artificiales, que a partir de la información en pulsos de una cóclea artificial pulsante, sea capaz de reconocer sonidos. Se realizará sobre dos tipos de redes neuronales artificiales:
  - Redes neuronales tradicionales.
  - Redes neuronales pulsantes.
- Implementar estos nuevos modelos sobre una plataforma hardware basada en una FPGA, con los siguientes requisitos:

- La implementación no debe incluir ningún computador convencional en el núcleo del procesado. No pretendemos ser fundamentalistas del procesado pulsante, pero sí llevar hasta donde sea posible el paradigma de los sistemas nerviosos naturales, en los que dichos elementos simplemente no existen.
- La implementación debe ser realista y realizable, modular y que permita demostrar empíricamente la viabilidad de la construcción de este nuevo sistema neuromórfico, que incluye tanto el sistema de sensado como el de procesado de audio.
- Caracterizar los nuevos modelos desarrollados a partir de pruebas y experimentos sobre estímulos reales. Para ello, se van a reconocer tres tipos de sonidos: tonos puros, notas musicales y el sonido que realiza un motor a diferentes revoluciones.

### **1.3. Estructura de la tesis**

Esta memoria se ha estructurado en 9 capítulos que se detallan a continuación:

- El capítulo 1 en el que se exponen las motivaciones, objetivos y estructura del documento.
- En el capítulo 2 se hace una exposición del estado de los sistemas neuromórficos actuales, basados en la representación AER.
- En el capítulo 3 se estudian el sistema auditivo, la psicoacústica y los modelos del sistema auditivo. También se hace un análisis de las cócleas artificiales implementadas para poder comparar sus características con las implementaciones desarrolladas en este trabajo.
- En el capítulo 4 se exponen dos sistemas cuya función es monitorizar la actividad de pulsos entre diferentes capas de neuronas, con la finalidad de permitir la comunicación entre capas de neuronas artificiales y por lo tanto, entre los diferentes sistemas que se exponen en este trabajo.
- En el capítulo 5 se exponen los sistemas neuromórficos de audición propuesto en este trabajo.

- En el capítulo 6 se exponen las diferentes metodologías utilizadas para el reconocimiento de sonidos y patrones. Además, se realiza un análisis de los sistemas de reconocimientos de sonidos y sus prestaciones para poder compararlos con los sistemas de reconocimientos desarrollados en este trabajo.
- En el capítulo 7 se exponen tres tipos de reconocimiento de sonidos mediante diferentes metodologías: método estadístico, redes neuronales de tasas, y por último, redes neuronales pulsantes.
- En el capítulo 8 se exponen las conclusiones y aportaciones logradas con este trabajo.
- Finalmente en el capítulo 9 se presentan las líneas de trabajo futuras.

Presentamos las referencias bibliográficas usadas en esta tesis agrupadas en el Capítulo 10, y en el Anexo se incluyen descripciones de las plataformas hardware usadas en este trabajo, y los resultados completos de los experimentos de reconocimiento de sonidos más relevantes.

## 2. Los sistemas neuroinspirados, justificación y antecedentes

*“La imitación es la más sincera forma de adecuación”,  
Charles Caleb Colton*

El comienzo de la vida en la tierra se data hace 4.400 millones de años, desde aquel momento los seres vivos comenzaron a colonizar todo el planeta. A lo largo y ancho de la tierra nos encontramos con una gran diversidad de condiciones medioambientales, desde ambientes tanto favorables como extremos para la vida. Sin embargo, los seres vivos han conseguido colonizar exitosamente estos entornos. Una de las claves más importantes para la expansión de la vida ha sido la capacidad de adaptación de los seres vivos, estando dotados por la naturaleza de las cualidades necesarias para poder sobrevivir a un determinado entorno. Además, dada la elevada diversidad de hábitats naturales, los seres vivos no han tenido más remedio que especializarse en la supervivencia en su entorno más próximo, y en consecuencia se ha producido una riquísima diversidad de especies. Las características particulares de cada especie están codificadas en el código genético de cada individuo, las cuales han sido moldeadas desde el origen de la vida gracias a la evolución natural. En 1859 Charles Darwin publicó su teoría de la evolución en el “Origen de las especies”. En ella proponía que mediante selección natural, los individuos de una especie mejor adaptados en su entorno, tenían mayores probabilidades de sobrevivir y perpetuar así su línea genética, es decir, los individuos más exitosos distribuían su código genético a una mayor cantidad de nuevos individuos. Sin embargo, la transmisión de la información genética de un individuo a otro no está libre de errores, sino que incluye un pequeño “ruido” o mutaciones. De manera que el código genético de cada individuo es prácticamente



único, presentando pequeñas variaciones que en una u otra medida afectarán a la manera en la que el individuo sobrevivirá en su entorno, siendo ésta la clave de la adaptación y la evolución de las especies. Gracias a la evolución, la naturaleza ha conseguido crear una gran diversidad de especies, logrando una infinidad de soluciones muy eficientes a los problemas de adaptación de los seres vivos en su entorno.

A lo largo de la historia en infinidad de ocasiones, los ingenieros se han inspirado en las soluciones alcanzadas por la naturaleza para resolver problemas en los más diversos campos, siendo éste el origen de los sistemas bio-inspirados, encontrándolos a nuestro alrededor cada vez con más frecuencia. Por ejemplo, alguno de los sistemas bio-inspirados, son los submarinos y los robots humanoides.

En las últimas décadas la industria ha sufrido una revolución gracias a la aparición de los sistemas computacionales y robóticos. En la industria se le han ido asignando diversas tareas a los robots, en especial tareas peligrosas, aquellas que requieren de una “fuerza sobrehumana”, o muy repetitivas y precisas. Sin embargo estos robots están programados para realizar un conjunto muy limitado de tareas, teniendo que estar localizados en un ambiente casi completamente controlado, con una capacidad de adaptación y aprendizaje prácticamente nula, y consumiendo unas cantidades ingentes de energía comparados con los seres vivos. Sin embargo este hecho contrasta frontalmente con las habilidades de los animales, y sobre todo con la extrema facilidad con la que se desenvuelven en su entorno, no sólo pueden navegar libre e inteligentemente (dentro de sus posibilidades) por su entorno, sino que son capaces de procurarse su propia energía, aprender, desarrollar actividades sociales, personalidades propias, comunicarse, organizarse para lograr fines comunes, etc... En otras palabras, el desarrollo de habilidades sociales, cognitivas y culturales.

En la actualidad la mayoría de los robots están gobernados por comportamientos algorítmicos procesados por sistemas basados en computadores. En los últimos años los computadores han evolucionado a un ritmo muy alto, alcanzando una capacidad de procesamiento elevadísima, tal y como fue predicho por la Ley de Moore (Moore 1998). Sin embargo las habilidades de los robots no

han aumentado en la misma medida, progresando a un ritmo muchísimo más lento. Caben plantearse dos preguntas, ¿se puede modelar con la suficiente fidelidad el comportamiento de un ser vivo para con posterioridad ser codificado algorítmicamente? Y en caso de que así fuera, ¿podría ejecutarse este algoritmo en tiempo real y con suficiente fiabilidad en un computador? Las respuestas a ambas preguntas son en cualquier caso debatibles, y por supuesto dependientes del animal a ser tomado como modelo, pero no parece muy arriesgado afirmar que el comportamiento del ser humano será difícilmente modelable en las próximas décadas. Las posibles respuestas a estas preguntas pueden dar lugar a muchas más preguntas, pero de entre ellas cabe plantearse si los sistemas basados en computador actuales son los sistemas más adecuados para proporcionar a los robots de habilidades cognitivas avanzadas (Penrose 1989).

La solución a todas estas cuestiones es actualmente desconocida, pero tal vez pase por imitar al “controlador” de los propios seres vivos, el sistema nervioso, realizando tareas de “ingeniería inversa”, surgiendo los sistemas neuro-inspirados. Estos sistemas son un subconjunto de los sistemas bio-inspirados, que tratan de resolver problemas comunes en la ingeniería mediante el uso de sistemas basados en la manera que el sistema nervioso codifica y procesa la información, siendo un campo en continuo desarrollo gracias al trabajo de los ingenieros neuromórficos.

En este capítulo, se estudian los principios por los que se rige la ingeniería neuromórfica; se verán diferentes sistemas sensoriales neuromórficos, muchos de ellos basados en la representación por direcciones de eventos, *AER*, también explicada en este capítulo.

## 2.1. Ingeniería Neuromórfica

El término ingeniería neuromórfica fue usado por primera vez por Carver Mead en el Instituto de Tecnología de California (CalTech) al final de los años 80, siendo su objetivo inicial imitar el comportamiento de las neuronas en el sistema nervioso mediante circuitos analógicos VLSI (aVLSI). Su aportación, ha consistido en comprender los sistemas neuronales biológicos por medio de su recreación en

silicio, lo que ha impulsado el campo del diseño de circuitos neuromórficos analógicos (Mead 1990). Sin embargo, el campo de estudio de los ingenieros neuromórficos se ha expandido en los últimos años, usando tanto circuitos analógicos, como digitales, o de señal mixta para implementar modelos neuronales.

Los sistemas biológicos sobrepasan a cualquier sistema de percepción hecho por el hombre respecto a su eficiencia, sus mecanismos de aprendizaje, robustez al ruido, y al poco consumo de potencia necesario para su funcionamiento. Es por esto, que el diseño de sistemas de procesamiento bio-inspirados es una alternativa en el que se puede lograr un mejor resultado en el consumo de potencia, la velocidad de procesamiento, los tiempos de respuesta y el área utilizada, en comparación con los sistemas y técnicas tradicionales.

La ingeniería neuromórfica se define como un campo de investigación multidisciplinar dedicado al diseño y a la fabricación de sistemas artificiales de computación, cuyas propiedades físicas, estructuras o representaciones de la información están basadas en el sistema nervioso biológico. Por lo tanto, la ingeniería neuromórfica agrupa a investigadores especialistas en los más diversos campos, como son matemáticos, físicos, biólogos, psicólogos e ingenieros de las más diversas ramas.

Cuando se desarrollan sistemas de computación bio-inspirados es importante identificar las características de los sistemas biológicos en las que se basa su capacidad de computación y que pueden ser adaptadas a la tecnología de implementación mediante circuitos electrónicos relativamente sencillos. Además, debe aprovechar características inherentes a los circuitos electrónicos como su mayor ancho de banda, velocidad de respuesta, etc.; y utilizar esquemas como multiplexación temporal, comunicación binaria, comunicación mediante direcciones, etc. La combinación óptima de principios bio-inspirados y características explotables de los circuitos electrónicos constituye la esencia de la ingeniería neuromórfica.

En 1906 el científico español Santiago Ramón y Cajal recibió el premio Nóbel de medicina por sus pioneras investigaciones sobre la estructura microscópica del

cerebro, descubriendo que el cerebro estaba formado por un conjunto de células independientes conectadas entre sí, las *neuronas*. Sus estudios fueron posibles gracias a los avances en los métodos de tinción y de tecnología de los microscopios, que permitieron plasmar en láminas ilustradas sus observaciones, pudiéndose encontrar en la Figura 2.1 una reproducción de una de las láminas originales.

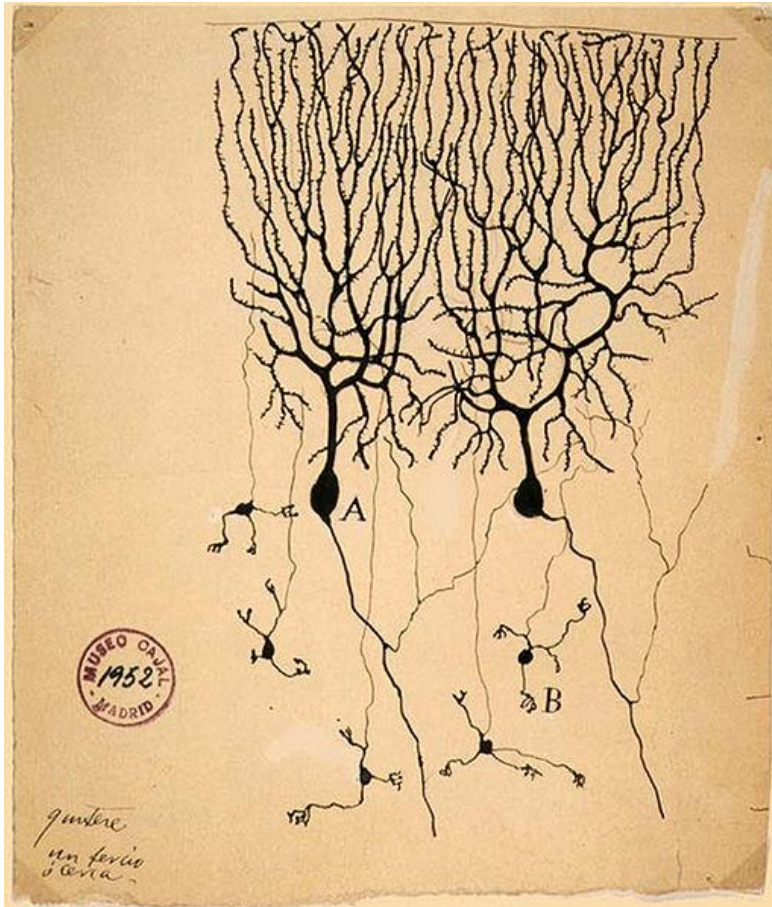


Figura 2.1. Reproducción de una lámina ilustrada de Ramón y Cajal del cerebro de un ave

Las neuronas son un tipo especial de células que transmiten su actividad en forma de impulsos eléctricos o potenciales de acción, y aunque son diversas, la mayoría están compuestas por tres elementos, el soma, o núcleo de la neurona, las

dendritas, compuestas por filamentos de la membrana celular sensible a estímulos externos, y el axón, el cual representa el canal a través del que “viajan” los estímulos generados por una neurona a otras neuronas a las que están conectadas mediante sus dendritas. El fenómeno que se produce al transmitirse el estímulo generado por una neurona a otra mediante el axón de la emisora y la dendrita de la neurona receptora es conocido como la *sinapsis* (Johnston & Wu 1995). Las neuronas tienen características fisiológicas muy complejas, en 1939 Hodgkin y Huxley (HODGKIN & HUXLEY 1939) analizaron el comportamiento electrónico de una neurona aislada, estudiando el comportamiento de los canales de sodio y potasio, recibiendo gracias a su estudio el premio Nóbel de medicina en 1963; demostrando que las neuronas representaban, comunicaban y procesaban la información mediante pequeños pulsos electrónicos en el tiempo, conocidos como potenciales de acción, *action potentials* o *spikes*. En su modelo describen el comportamiento de una neurona como una serie de ecuaciones diferenciales no lineales (Lamberti & Rodríguez 2007). En la Figura 2.2 se muestra una neurona con sus elementos, una representación del potencial de acción o spike y la sinapsis entre dos neuronas conectadas.

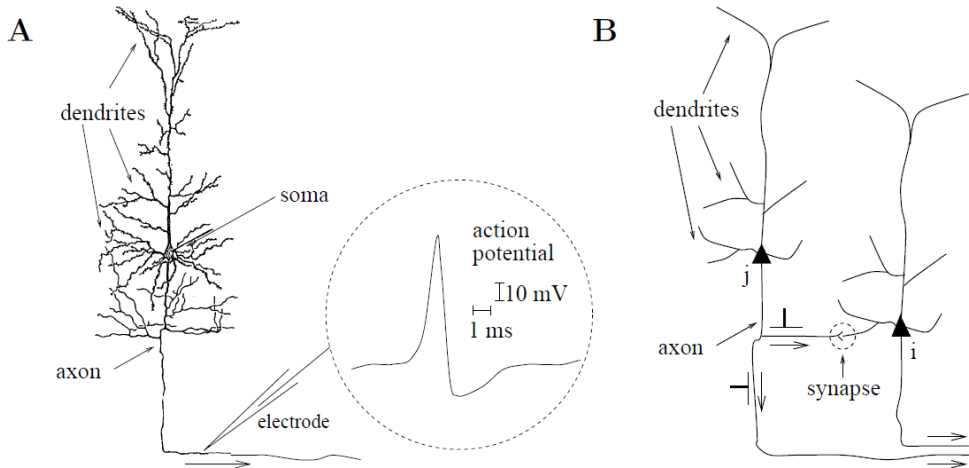


Figura 2.2. A) Neurona de los grabados de Ramón y Cajal en la que están indicadas las partes de la neurona y esquema en el que se representa el potencial de acción o spike. B) Señal transmitida desde la neurona *j* a la neurona *i*. La sinapsis está marcada por un círculo. Imagen tomada de (Gerstner et al. 2002).

Un *spike* consiste en una alteración del voltaje entre las membranas de las dendritas pertenecientes a las neuronas. De esta forma, se generan nuevos *spikes*, en base a los recibidos por sus dendritas, estando toda la información codificada en los *spikes* disparados<sup>6</sup> por las diversas neuronas (Shepherd 2004; Maass & Bishop 1999).

La Figura 2.3 muestra las características temporales de un *spike* al ser disparado por una neurona; al principio los canales de la membrana de la neurona se encuentran polarizados con un potencial inferior a un umbral (*Threshold*), no emitiendo, y por tanto, no realizando ninguna actividad mientras este umbral no se supere. En algún momento llegarán a la neurona una serie de estímulos externos, de tal manera que si el potencial de la membrana de la neurona alcanza el umbral, despolarizándose bruscamente, se genera un spike y se transmite un impulso nervioso, conteniendo en él la información procesada. Un breve instante después a la despolarización de la neurona, se polarizará de nuevo hasta “hiper-polarizarse”, no siendo capaz de transmitir un nuevo spike hasta transcurrido un período de tiempo conocido como período refractario.

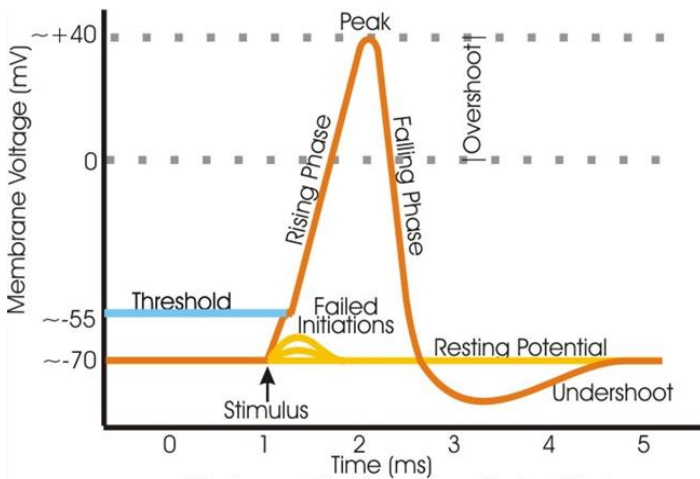


Figura 2.3. Diagrama de un potencial de acción, o spike, generado por una neurona

Uno de los puntos clave de los mecanismos de procesamiento neuronal es la hipótesis de la representación de la información por parte de las neuronas. Mucho

<sup>6</sup> En el campo de la ingeniería neuromórfica, disparar un spike se usa como sinónimo a emitir un spike

se ha debatido sobre cómo codificar la información en spikes, Horace Barlow en 1961 propuso varios modelos (Barlow 1961) siendo un modelo muy aceptado en el campo de la ingeniería neuromórfica el que propone la codificación de la información en la frecuencia de los spikes, siguiendo una modulación en frecuencia de pulsos, spikes, o PFM (Westerman et al. 1997; Maass & Bishop 1999). De esta manera la información puede ser codificada de forma continua, sin necesidad de realizar una discretización temporal de la información (Fujii et al. 1996; Hynna & Boahen 2001). Otras formas de codificar la información es mediante el intervalo de tiempo entre los spikes (Giacomo Indiveri et al. 2006), o mediante el tiempo desde el reset, donde los eventos más importantes son los que se han lanzado primero (con mayor prioridad) (Thorpe et al. 2010).

La representación pulsante es muy eficiente desde varios puntos de vista:

- **Simplicidad:** reduce la cantidad de canales de comunicación necesarios para transmitir los pulsos.
- **Continuidad:** información continua en el tiempo, en ningún caso discreta.

Ambos hechos tienen dos consecuencias:

- **Minimización de los canales de comunicación:** permitiendo una alta tasa de conectividad entre las neuronas.
- **Evita la transmisión de información redundante:** al no estar la información muestreada, se transmiten los pulsos únicamente cuando son necesarios, evitando saturar los canales de comunicación innecesariamente.

Esta representación no sólo es eficiente desde el punto de vista de la comunicación y conectividad, sino que también es robusta frente al ruido, ya que la información está codificada en el tiempo entre pulsos; siendo de importancia la existencia (o no) de pulsos; mientras que las señales analógicas clásicas están indefensas ante las perturbaciones externas.

El cerebro es el elemento central del sistema nervioso de todos los vertebrados y la mayoría de los invertebrados, localizándose en la cabeza protegido por el cráneo y próximo a los órganos sensitivos más importantes, como son: la vista, el oído, el equilibrio, el gusto y el olfato. El cerebro humano es extremadamente complejo, se

estima que contiene entre 15 y 33 billones de neuronas, pudiendo estar cada una de ellas conectadas con otras 10 mil. Las neuronas se estructuran en capas, las cuales están especializadas en procesar una parte de la información, y tienen una funcionalidad más o menos definida (Rakic 1988; Shadlen & Newsome 1994). Para realizar el procesamiento de la información, cada neurona está conectada a un campo proyectivo de neuronas a lo largo de diversas capas, fluyendo la información entre capas, y procesándose en este mismo flujo. Por ejemplo, se piensa que el córtex cerebral humano, o neo-córtex<sup>7</sup>, es la parte más “evolucionada” de nuestro cerebro, siendo el responsable de la memoria, la atención, el pensamiento, el lenguaje y la consciencia.

En la Tabla 2.1 se muestra una comparativa cualitativa desde un punto de vista muy general acerca de la manera en que funciona un computador y el sistema nervioso de los seres vivos. La primera diferencia que encontramos es que un computador está sincronizado por una señal de reloj global, que le hace reaccionar continuamente cada ciclo de reloj, sin embargo, las neuronas se comportan de manera completamente asíncrona, no existiendo ningún mecanismo explícito de sincronización entre ellas. Además el computador es un elemento completamente determinista, que dictamina en todo momento tras una sucesión de operaciones aritméticas y lógicas en qué estado debe encontrarse, en yuxtaposición, las neuronas responden a un modelo estocástico, dependiendo su reacción de modelos probabilísticos dinámicos.

Los sistemas actuales computacionales tienen una alta resolución de la información que manejan, muestreada a un ritmo constante, una vez más, el sistema nervioso es completamente opuesto, la resolución de la información no es tan elevada, pero es capaz de adaptarse a las características de la información para mejorar su representación. En los sistemas computacionales actuales el procesamiento en sí de la información está muy centralizado, en el caso de ordenador personal, o levemente distribuido, como en un clúster, comparado con la manera en que las neuronas procesan la información, ya que cada neurona procesa de manera muy simple una pequeña parte de la información, no dependiendo de

---

<sup>7</sup> Áreas más evolucionadas del córtex, también llamado isocórtex.



otras neuronas, e implementando de esta manera un modelo de procesamiento de la información masivamente paralelo. Para usar los computadores actuales resulta imperativo proveerles de dispositivos de memoria tanto para almacenar los algoritmos a ejecutar, así como los datos iniciales, intermedios y finales, por contra, los sistemas nerviosos no necesitan memoria para ninguno de estos fines, ya que por un lado el “algoritmo neuronal” que ejecutan las neuronas está modelado mediante las características fisiológicas de cada neurona y la topología con la que se conecta con otras neuronas, y por otro lado, toda la información relativa al procesamiento simplemente fluye, no se deja almacenada en ninguna posición de memoria, sino que parte desde los órganos sensitivos y va siendo procesada a medida que va atravesando capas neuronales. En consecuencia, desde el punto de vista de los sistemas computacionales, la solución alcanzada por la naturaleza se presenta completamente revolucionaria en muchísimos aspectos, además de los ya comentados, representando el desarrollo de elementos computacionales neuro-inspirados una actividad muy innovadora y prometedora.

Tabla 2.1. Comparativa cualitativa entre un computador y el sistema nervioso

<b>Computador</b>	<b>Sistema nervioso</b>
Síncrono, con reloj centralizado de alta velocidad	Asíncrono, sin ninguna señal global de reloj
Comportamiento determinista: dictamina el estado lógico en que debe encontrarse	Las neuronas se comportan de forma estocástica, respondiendo a modelos probabilísticos dinámicos
Alta resolución de la información con una tasa de muestreo constante	De baja resolución, pero adaptativo. Sin período de muestreo, estando la información contenida en los spikes.
La computación está centralizada o levemente distribuida	Computación distribuida y masivamente paralela: cada neurona procesa una pequeña parte de la información
La memoria está muy “lejos” del computador, necesitando memoria tanto para el algoritmo como para almacenar los datos	Las características morfológicas y las interconexiones de cada neurona son el algoritmo en sí, la información está contenida en el fluir de los spikes

La mayoría de los sistemas neuromórficos están formados por uno o más sensores neuromórficos y una red de neuronas artificiales pulsantes, que tratan de imitar la interconexión de las neuronas biológicas. Desafortunadamente, la extensiva conectividad del cerebro, donde cada neurona puede estar conectada con otras 10 mil, es imposible implementarla directamente en sistemas *VLSI* debido a las limitaciones físicas de conectividad dentro y entre microchips. Sin embargo, las neuronas presentan tiempos de respuesta del orden de milisegundos, mientras que los tiempos de los circuitos electrónicos actuales están en el orden de los nanosegundos, es decir, un millón de veces más rápido que las neuronas biológicas. Basados en esta diferencia temporal y en un alto ancho de banda de los sistemas *VLSI*, se propone, como solución al problema de la conectividad, un sistema de multiplexación en el tiempo sobre un mismo canal, a través del cual se transmite un identificador para cada neurona que emite. A este esquema se le denomina *Address-Event Representation*, (*AER*) (Silviotti 1991), (Mahowald 1992), (K. A. Boahen 2000), (K. a. Boahen, 2000).

### ***2.2. Address-Event-Representation***

La Representación *Address Event* (*AER*) es un protocolo de comunicación conducido por eventos (*event-driven*) usado originalmente en implementaciones *VLSI* de redes neuronales para transferir pulsos<sup>8</sup> (*action potentials*) entre neuronas (Mahowald 1992).

La representación *AER* fue usada primero como un acercamiento a la masiva conectividad de las redes neuronales biológicas, aunque, en general es adecuada para transportar una gran cantidad de información, codificada en frecuencia de eventos, a través de un canal de menor capacidad (bus digital asíncrono). Es por tanto una técnica de multiplexación digital asíncrona. En principio, no especifica la naturaleza de las unidades involucradas: ellas pueden ser circuitos integrados neuromórficos, circuitos digitales, emulaciones software de sistemas neuronales, neuronas biológicas o software algorítmico común.

---

<sup>8</sup> Se usan las palabras pulso y spike como sinónimos de potencial de acción

En la Figura 2.4 se muestra de manera esquemática la realización de una transmisión de información mediante la representación AER entre dos chips neuroinspirados. La representación AER propone el uso de un bus común multiplexado de alta velocidad para la comunicación de los pulsos disparados por las neuronas de un chip: el bus AER. La idea es asignar una dirección (*address*), a cada neurona, de manera que cada vez que una neurona dispare un pulso (*event*) en el chip transmisor, un arbitrador provocará que aparezca en el bus AER la dirección de la neurona que ha producido el pulso (*spike*), denominado evento AER, pudiendo ser transmitido de diversas maneras, como podrá comprobarse posteriormente. Una vez recibido un evento AER en el chip receptor, éste es decodificado, enviando el pulso original a una serie de neuronas receptoras. De esta forma, las neuronas de cada chip se encuentran virtualmente conectadas, fluyendo la información entre ellas, ya que el acceso al canal común se encuentra multiplexado. Las neuronas más activas accederán a él de manera más eficiente (Boahen 1998; K. Boahen 2000; Zaghoul & Boahen 2004a; Zaghoul & Boahen 2004b).

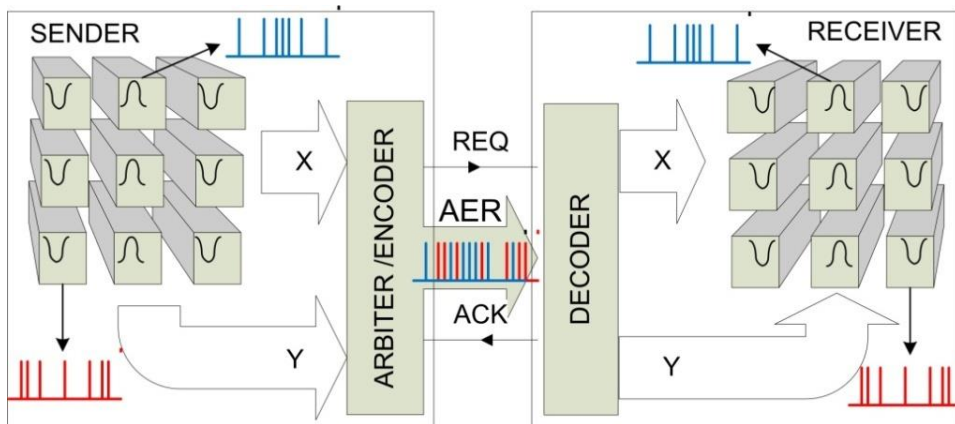


Figura 2.4. Esquema de la transmisión de la información pulsante mediante el uso de la representación AER

Se han propuesto diversas maneras de implementar el protocolo de comunicación de eventos AER. Estos protocolos de comunicación se pueden clasificar en dos grandes grupos: los protocolos paralelos (Linares-Barranco 2003; Häfliger 2004; Boahen 1998; K. Boahen 2000) y los protocolos serie, serial AER o

SAER, (Miró-Amarante et al. 2006; Miró-Amarante et al. 2007; Berge & Häfliger 2007; Fasnacht et al. 2008; Zamarreño-Ramos et al. 2008). En general los protocolos más extendidos son los protocolos paralelos, en particular, se tomará como referencia para el desarrollo de este trabajo, el protocolo utilizado durante el proyecto europeo CAVIAR (R. Serrano-Gotarredona et al. 2009), especificado formalmente en (Häfliger 2004). En la actualidad, los protocolos AER serie están en fase de desarrollo y testeo, siendo pocos los dispositivos hardware neuromórfico que lo implementan. Aun así, no cabe duda que dichos protocolos remplazarán paulatinamente a los paralelos, tal y como ha ocurrido en general con los sistemas digitales actuales de altas prestaciones.

El protocolo AER es un protocolo *handshake* de cuatro fases, entre el emisor y receptor que garantiza la sincronización entre ambos chips; las líneas de datos comunican la dirección del nodo emisor que solicita la petición al chip receptor. El protocolo permite de forma eficiente que un nodo emisor de un chip comunique pulsos digitales a un nodo receptor. Un chip, el emisor, inicia el proceso con una petición (*request*). El segundo chip, el receptor, debe contestar a la petición del emisor con un asentimiento (*acknowledge*). Para completar la transmisión, el emisor elimina la petición y el receptor el asentimiento. El sistema vuelve así a su estado inicial. Ambas partes están inactivas hasta que algún proceso del emisor inicia otra petición. Así, se dice que el protocolo *adres-event* está conducido por los datos (*data driven*) porque el inicio de la transmisión depende de los nodos neuronales del emisor que tratan de transmitir un evento. Puesto que las peticiones pueden aparecer en cualquier tiempo desde cualquier nodo, es necesario utilizar un esquema de arbitraje para serializar las operaciones del protocolo lo más rápido posible, del orden de nanosegundos.

En esta tesis, el protocolo de comunicación AER constituye la base para la transmisión y el procesado de la información auditiva. El protocolo AER, usado en este trabajo y en la mayoría de los desarrollos AER, fue especificado en el proyecto europeo CAVIAR: Convolution AER Vision Architecture for Real-Time (IST-2001-34124) (R. Serrano-Gotarredona et al. 2009). En el documento de especificaciones del AER (Häfliger, 2007) se describen las características del

conector y el cable (ATA/133 de 40 pines). El significado de cada pin se ilustra en la Figura 2.5. En la Figura 2.6 y Tabla 2.2 se muestran los parámetros y requisitos temporales del protocolo AER.

Header (on PCB, front view)		Connector (on Cable, front view)	
GND	39	Reserved	1
Reserved	37	Reserved	2
Reserved	35	Reserved	3
Reserved	33	Reserved	4
Reserved	31	Reserved	5
Reserved	29	Reserved	6
GND	27	/ACK	7
Reserved	25	Reserved	8
GND	23	Reserved	9
GND	21	Reserved	10
key pin pin missing	19	/REQ	11
AE[15]	18	GND	12
AE[14]	16	GND	13
AE[13]	14	GND	14
AE[12]	12	Reserved	15
AE[11]	10	Reserved	16
AE[10]	9	Reserved	17
AE[9]	7	Reserved	18
AE[8]	5	Reserved	19
GND	3	Reserved	20
	2	Reserved	21
	1	Reserved	22
	0	Reserved	23
	-1	Reserved	24
	-2	Reserved	25
	-3	Reserved	26
	-4	Reserved	27
	-5	Reserved	28
	-6	Reserved	29
	-7	Reserved	30
	-8	Reserved	31
	-9	Reserved	32
	-10	Reserved	33
	-11	Reserved	34
	-12	Reserved	35
	-13	Reserved	36
	-14	Reserved	37
	-15	Reserved	38
	-16	Reserved	39
	-17	Reserved	40

Figura 2.5. Especificación de pines de los conectores y cables AER del proyecto CAVIAR. Imagen tomada de (Häfliger 2007).

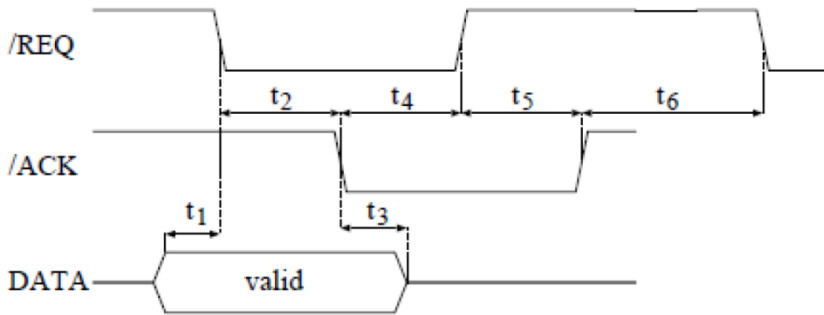


Figura 2.6. Fases del protocolo Handshake. Las restricciones de tiempo se muestran en la Tabla 2.2. Imagen tomada de (Häfliger 2007)

Tabla 2.2. Requisitos de tiempo de las cuatro fases del protocolo Handshake de la Figura 2.6. Imagen tomada de (Häfliger 2007)

	min	max	avg
$t_1$	0s	$\infty$	
$t_2$	0s	$\infty$	$\leq 700\text{ns}$
$t_3$	0s	$\infty$	
$t_4$	0s	100ns	
$t_5$	0s	100ns	
$t_6$	0s	$\infty$	

### 2.3. Implementaciones de sistemas neuro-inspirados

Actualmente existen una gran variedad de implementaciones hardware de sistemas neuro-inspirados que utilizan la representación AER, todos ellos elementos desarrollados en los diversos centros de investigación. Pero antes de poder procesar la información pulsante codificada según la representación AER, resulta imperativa su obtención, para ello encontramos sobre todo una gran variedad de sensores neuro-inspirados.

Las retinas<sup>9</sup> artificiales son sensores de visión neuro-inspirados, en las que cada pixel representa a una célula de una retina biológica, que emiten una serie de spikes cuya frecuencia dependerá de una determinada función de la iluminación a la que se ve sometida. Las retinas pulsantes poseen una particularidad que las diferencia del resto de cámaras: las imágenes que se obtienen no están discretizadas en fotogramas, sino que son continuas en el tiempo. Existen diversas implementaciones, ya sean en chips aVLSI (Mahowald 1992; Boahen 1999; Barbaro et al. 2002; Culurciello et al. 2003; Lichtsteiner & Delbruck 2005; Lichtsteiner et al. 2006; Patrick Lichtsteiner et al. 2008; Costas-Santos et al. 2007; Leñero-Bardallo et al. 2009) o retinas sintéticas basadas en FPGA<sup>10</sup> (Paz-Vicente et al. 2009). En la Figura 2.7 se muestra esquemáticamente la retina artificial de (Patrick Lichtsteiner et al. 2008). También se expone un ejemplo de fusión sensorial en (Jiménez-Fernández et al. 2010), en el que se combina una retina artificial y un acelerómetro para el diseño de un sistema de estabilización visual neuro-inspirada. En general, hasta la fecha, las retinas artificiales han sido los sensores más utilizados por los sistemas neuro-inspirados.

---

<sup>9</sup> Tejido sensible a la luz situado en la superficie interior del ojo

<sup>10</sup> Field-Programmable Gate Array

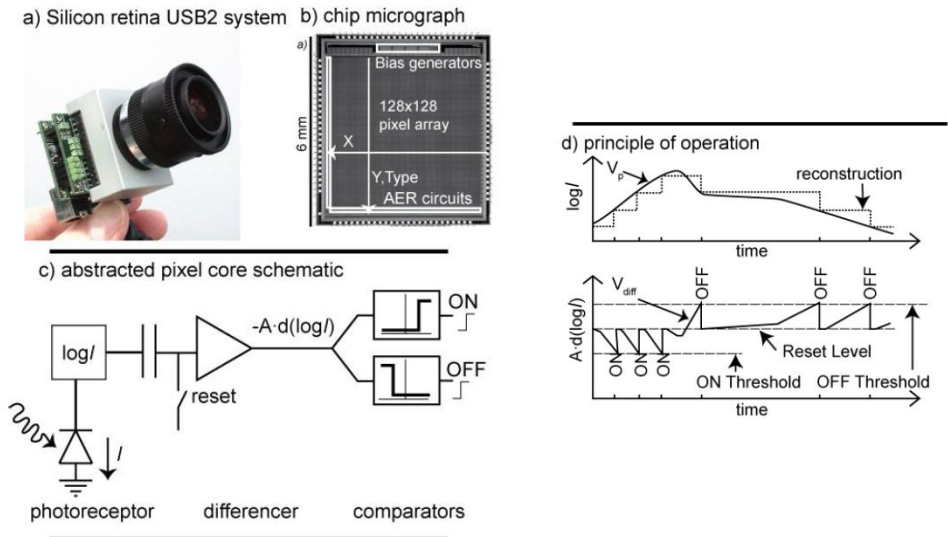


Figura 2.7. Retina artificial diferencial espacio-temporal. Imagen tomada de (Patrick Lichtsteiner et al. 2008).

Otros sensores algo menos extendidos son las cócleas artificiales. Generalizando, estos sensores son sensibles al sonido, descomponiéndolo en componentes frecuenciales. Algunas implementaciones son: (Lyon & Mead 1988), (Wen & Boahen 2009), (Hamilton et al. 2008); y que usen AER: (Lazzaro & Wawrzynek 1995), (Kumar et al. 1998) y (Liu et al. 2010). En la Figura 2.8 se muestra un ejemplo de implementación de una cóclea artificial analógica desarrollada en el Instituto de Neuro-Informática de la universidad ETH de Zürich (Liu et al. 2010), que está comercialmente disponible. En la placa de prototipo se destaca el chip que implementa el par de cócleas analógicas artificiales de 64 canales, la interfaz USB, y la interfaz con dos micrófonos, (a). A la derecha, se representa la salida del sistema en forma de cocleograma, como respuesta a la señal de voz “The quick red fox jumped over the lazy dog”. El eje y se corresponde con los 64 canales de cada cóclea artificial en función del tiempo (eje  $x$ ) (b). En el capítulo 3 del presente trabajo se exponen en mayor profundidad las cócleas artificiales ya implementadas, y en el capítulo 5 exponemos el modelo de una innovadora cóclea artificial, desarrollada en este trabajo.



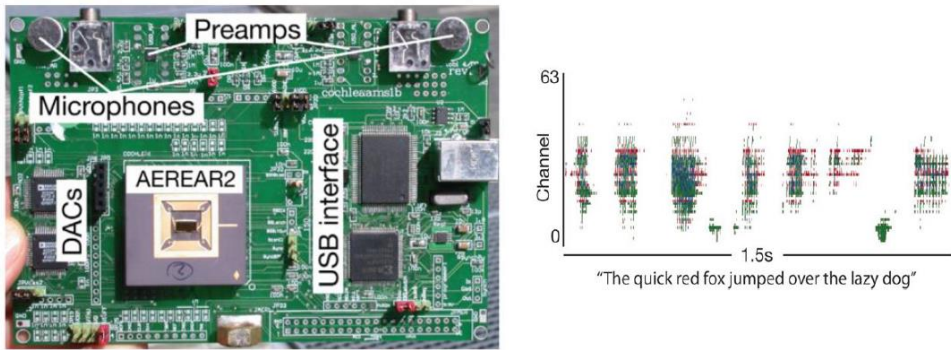


Figura 2.8. (a) Placa de prototipo que incorpora el chip AEREAR2.(b) Cocleograma ante la señal de voz: " The quick red fox jumped over the lazy dog". Imagen tomada de (Liu et al. 2010)

A partir del auge en los sensores neuro-inspirados, han surgido investigaciones sobre fusión sensorial, sistemas que integran varios sensores para aprovechar la colaboración entre los datos generados. Ejemplos de sistemas integran fusión sensorial audio-visual son: (O'Connor et al. 2013), (Rios-Navarro et al. 2014) y (Chan et al. 2012).

Para procesar la información AER existen diversos modelos computacionales y elementos que los implementan. Un modelo muy usado es el modelo *Integrate & Fire*<sup>11</sup>, del que podemos encontrar diversos chips neuromórficos que contienen capas de neuronas implementando este modelo (Liu & Douglas 2004; Goldberg et al. 2001). También se pueden encontrar sistemas que realizan convoluciones<sup>12</sup> basadas en pulsos o eventos AER, existiendo una gran variedad de convolucionadores AER, ya sean analógicos (Serrano-Gotarredona et al. 2005; Serrano-Gotarredona et al. 2007; Serrano-Gotarredona et al. 2008), o digitales (Paz-Vicente et al. 2008; Camuñas-Mesa et al. 2008), (Ríos et al. 2012). Otros sistemas son mixtos, incluyendo un computador empotrado para procesamiento AER (Luján-Martínez et al. 2007), aprovechando la potencia del co-diseño<sup>13</sup> hardware/software; así como propuestas de redes híbridas, en las que se utilizan

<sup>11</sup> Integrar y disparar

<sup>12</sup> Función que, de forma lineal y continua, transforma una señal de entrada en una nueva señal de salida.

<sup>13</sup> Diseñar el sistema hardware y software de un sistema mixto. El comportamiento está formado por la ejecución de ambos.

nuevas arquitecturas de redes celulares para integrarlas con sistemas AER (Linares-Barranco et al. 2008). Como aplicaciones particulares de los convolucionadores, se han presentado diversos dispositivos para el filtrado de visión AER (Serrano-Gotarredona et al. 1999; Gomez-Rodriguez, Linares-Barranco, Paz, et al. 2007), (Camuñas-Mesa et al. 2011). En este trabajo se ha desarrollado un sistema de clasificación de sonidos basado en redes neuronales de convolución (ConvNets), por lo tanto, en el capítulo 6 se describe en más detalle este tipo de procesamiento con spikes y en el capítulo 7 se expone el sistema desarrollado.

Otros modelos muy extendidos son, el de la *sinapsis probabilística* (Westerman et al. 1997), en el que los pulsos son procesados mediante el uso de técnicas probabilísticas (Goldberg et al. 2001); y el modelo *Winner-Take-All*<sup>14</sup>, en el que sólo la neurona más activa de toda una capa dispara pulsos, implementando de esta manera modelos atencionales (Indiveri 2000; Oster et al. 2007).

También existen implementaciones basadas en la representación AER pulsante capaces de ser entrenadas para aprender patrones, encontrando diversos modelos e implementaciones en circuitos aVLSI (G. Indiveri et al. 2006; Yang et al. 2006), así como propuestas para el aprendizaje de robots (Molina-Vilaplana et al. 2007). Estos sistemas se han utilizado para las más diversas aplicaciones como, por ejemplo, en el reconocimiento de voz (Chakrabartty & Liu 2010), la ecolocalización (Yu et al. 2009), etc.

Además, se han desarrollado sistemas inspirados en la capacidad de locomoción animal. Destacamos dos sistemas que realizan el control de actuadores robóticos mediante spikes: controlador PID de lazo cerrado basado en spikes (Jimenez-Fernandez et al. 2012) y generación dinámica de trayectorias (Perez-Peña et al. 2013). Siguiendo con esta filosofía de procesamiento de información neuroinspirada, en (Jimenez-Fernandez et al. 2010) se propone el desarrollo de sistemas análogos al procesamiento de señales clásico (DSP), pero sobre spikes, conocido como *Spikes Signal Processing* (SSP). Este sistema trata de implementar las

---

<sup>14</sup> El ganador se lo queda todo.

operaciones aritméticas básicas (suma, resta, integración, derivación...) pero sobre flujos de spikes haciendo uso de circuitos digitales dedicados en una FPGA.

Además, puede encontrarse una gran variedad de ejemplos en los que estos sistemas han sido integrados en plataformas robóticas (Lewis et al. 2003), (Lewis et al. 2001), (Vogelstein et al. 2008), (Vogelstein et al. 2006), (Gomez-Rodriguez, Linares-Barranco, Miro, et al. 2007), (Rios-Navarro et al. 2014), (Conde, Orbe, De Diego, et al. 2011), (Conde, Orbe, Diego, et al. 2011), (Chan et al. 2012).

El diseño, implementación, e integración de sistemas neuro-inspirados no sería posible si no existieran herramientas especializadas para la depuración y el desarrollo de sistemas AER. Las llamadas *AER-Tools* (Paz-Vicente et al. 2005; Gómez-Rodríguez et al. 2006), son una serie de elementos que permiten la depuración e interfaz con un ordenador personal de los sistemas AER (Paz-Vicente et al. 2006; Berner et al. 2007), para poder visualizar, reproducir, y analizar la información AER que circula por una cadena de multi-chips neuro-inspirados basado en la representación AER.

## 2.4. Grupos de investigación neuromórficos

A continuación se nombran diferentes grupos de investigación interesados en el AER:

- El grupo Brain in Silicon de la Universidad de Stanford, liderado por Kwabena Boahen (<http://www.stanford.edu/group/brainsinsilicon/>).
- El grupo Neuromórfico del IMSE-CNM-CSIC, liderado por Bernabé Linares Barranco (<http://www.imse-cnm.csic.es/>).
- The sensors research group del Instituto de Neuroinformática de la ETH de Zúrich, liderado por Tobias Delbruck (<http://sensors.ini.uzh.ch/>).
- The Neuromorphic Cognitive Systems group del Instituto de Neuroinformática de la ETH de Zúrich, liderado por Giacomo Indiveri (<http://ncs.ethz.ch/>).

- El e-Lab de la Universidad de Yale, liderado por Eugenio Culurciello (<https://engineering.purdue.edu/elab/blog/>).
- El Computational NeuroEngineering Lab (CNEL) de la Universidad de Florida, liderado por John G. Harris (<http://www.cnel.ufl.edu/>).
- El Computational Sensory-Motor Systems Lab de la Universidad Johns Hopkins, liderado por Ralph Etienne-Cummings (<http://etienne.ece.jhu.edu/>).
- El grupo de Robótica y Tecnología de Computadores de la Universidad de Sevilla, liderado por Antón Cívit Balcells ([www.rtc.us.es](http://www.rtc.us.es)).

En el campo de la ingeniería neuromórfica hay que destacar el Institute of Neuromorphic Engineering (INE, 2012), responsable de la organización de uno de los eventos anuales de esta disciplina más importantes, el Telluride Neuromorphic Cognition Engineering Workshop, que se celebra en Telluride en el estado de Colorado (Estados Unidos) desde mediados de los 90; y que tiene un reflejo en Europa en el Cappelletti Neuromorphic Cognition Engineering Workshop. En ambos eventos se dan cita los investigadores más importantes en la ingeniería neuromórfica, así como una gran cantidad de estudiantes que inician sus pasos en dicho campo científico-técnico. También, es interesante destacar una publicación especializada en este campo, *Frontiers in Neuromorphic Engineering*<sup>15</sup>.

El grupo de investigación Robótica y Tecnología de Computadores de la Universidad de Sevilla ha participado en los últimos años en varios proyectos de investigación (tanto regionales, como nacionales y europeos) para el desarrollo de sistemas pulsantes AER: (en orden cronológico) VICTOR (TIC2000-0406-P4-05), CAVIAR (R. Serrano-Gotarredona et al. 2009), SAMANTA I (TIC2003-08164-C03-02) y II (TEC2006-11730-C03-02) VULCANO (P050-12/E03) y BIOSENSE (TEC2012-37868-C04-02). Los proyectos mencionados pretendían en general la realización de demostradores que permitieran poner a prueba las posibilidades del procesado AER, en especial en lo referente a los campos sensorial y motor.

---

<sup>15</sup> Revista online de acceso libre sobre ingeniería neuromórfica:  
[http://www.frontiersin.org/neuromorphic\\_engineering](http://www.frontiersin.org/neuromorphic_engineering)



### 3. El sistema auditivo: modelos e implementaciones

*“Oír es precioso para el que escucha”, Proverbio egipcio*

Como se expuso en el capítulo anterior, los sistemas neuro-inspirados se caracterizan por emular las propiedades básicas del procesado sensorial y cerebral, con el objetivo de conseguir las ventajas que tienen los seres vivos. Para poder desarrollar este tipo de sistemas, es necesario conocer a fondo la base biológica de los procesamientos que se quieren emular. Por lo tanto, en este capítulo se exponen las características del sentido de la audición. Además, se realiza un análisis de los modelos y sistemas previos ya publicados, con el objetivo de conseguir un sistema eficiente y que presente ventajas y avances respecto a los trabajos previos.

En este capítulo, se expone en primer lugar, cómo el sistema auditivo capta, analiza y codifica la información acústica en impulsos nerviosos para ser procesados por el cerebro, en segundo lugar, se hace una introducción a los conceptos básicos de la psicoacústica los cuales van a permitir caracterizar la respuesta del sistema auditivo humano. Para terminar, se estudian los modelos computacionales que representan la propagación del sonido a lo largo del oído interno y la conversión de la energía acústica en impulsos nerviosos, así como algunas implementaciones de dichos modelos, compuesto por un estudio de las cócleas artificiales más relevantes.

Gracias al estudio de la información presente en este capítulo, implementamos nuestro sistema de forma que tenga el mismo comportamiento y las mismas propiedades que el sistema auditivo humano, y, además, que resuelva algunas de las desventajas de los sistemas previos. En el capítulo 5 se expone el sistema que

hemos desarrollado, realizando una comparativa respecto las cócleas artificiales expuestas en este capítulo.

### 3.1. Oír: El sentido de la audición

En esta sección se describe la anatomía y fisiología del aparato auditivo, haciendo énfasis en aquellas partes y estructuras más importantes para el desarrollo de modelos perceptuales.

Antes de empezar con la estructura física del oído, se exponen un breve repaso de los conceptos de percepción y sensación.

La percepción consiste en la detección de un estímulo por un receptor sensorial. Por ejemplo, los receptores sensoriales de nuestro cuerpo perciben continuamente cosas de las que no tenemos conocimiento consciente (p. ej., presión y pH arterial), ya que sus señales no son enviadas hacia la corteza cerebral para su integración en la consciencia. En otras circunstancias, la percepción lleva a la sensación, es decir, a la percepción consciente. En estos casos, los receptores sensoriales transmiten señales eléctricas a la corteza cerebral a través de los nervios que viajan hasta allí. En microsegundos, la corteza cerebral integra las señales en una sensación (McConnell & Hull 2010).

La audición es un sentido clasificado dentro del grupo de los sentidos especiales, que se caracterizan porque son sistemas complejos con aparatos de detección sofisticados, que ocupan una localización anatómica concreta y sus señales se integran en la consciencia. Por lo tanto, se puede definir la audición como la detección de las ondas sonoras y la integración en la consciencia de las señales sensoriales para obtener sensaciones (McConnell & Hull 2010).

Todas las sensaciones, incluyendo las generadas por el oído, son el resultado de la misma secuencia de acontecimientos:

1. Se produce un estímulo.

2. El receptor sensorial detecta el estímulo y lo convierte en una señal eléctrica.
3. La señal eléctrica es transmitida al cerebro, en el caso del oído, usando las vías auditivas.
4. El cerebro integra la señal en percepción consciente (sensación).

La captación, procesamiento y transducción de los estímulos sonoros se llevan a cabo en el oído propiamente dicho, mientras que la etapa de procesamiento neuronal, en el cual se producen las diversas sensaciones auditivas, se encuentra ubicada en el cerebro. Así pues, se distinguen dos regiones en el sistema auditivo: la **región periférica**, en la que los estímulos sonoros de ondas mecánicas se convierten en señales electroquímicas, y la **región central**, en la que se transforman dichas señales en sensaciones.

En la región central también intervienen procesos cognitivos, mediante los cuales se asigna un contexto y un significado a los sonidos; es decir, permiten reconocer una palabra o determinar si un sonido dado corresponde a un violín o a un piano.

### 3.1.1. Región periférica del sistema auditivo

La región periférica auditiva, conocida como oído, está formada por tres órganos interconectados: el *oído externo*, *oído medio* y *oído interno*. A través de estas zonas se propagan los estímulos sonoros que sufren diversas transformaciones hasta su conversión final en impulsos nerviosos. Tanto el procesamiento mecánico de las ondas sonoras como la conversión de éstas en señales electroquímicas son procesos no lineales (Fastl & Zwicker 2007), lo cual dificulta la caracterización y modelado de los fenómenos perceptuales.

A continuación, se describe la anatomía y funcionamiento de estas tres zonas del oído, así como la propagación y procesamiento del sonido a través de las mismas.

#### Oído externo



El oído externo comienza en su parte más externa con el pabellón auricular (también conocido como oreja). Su forma de embudo refleja su función, que es recoger las ondas sonoras y encauzarlas hacia el conducto auditivo externo. Además, interviene en la localización de sonidos debido a su forma y posición en la cabeza. El conducto auditivo externo, con cerca de 2,5 cm de largo, se extiende desde el pabellón hasta la membrana timpánica (tímpano) del oído medio. Su función es la de proteger la entrada al oído medio y mantener las estructuras del oído medio a una temperatura estable. Se pueden observar todos estos componentes nombrados previamente en la Figura 3.1 (McConnell & Hull 2010).

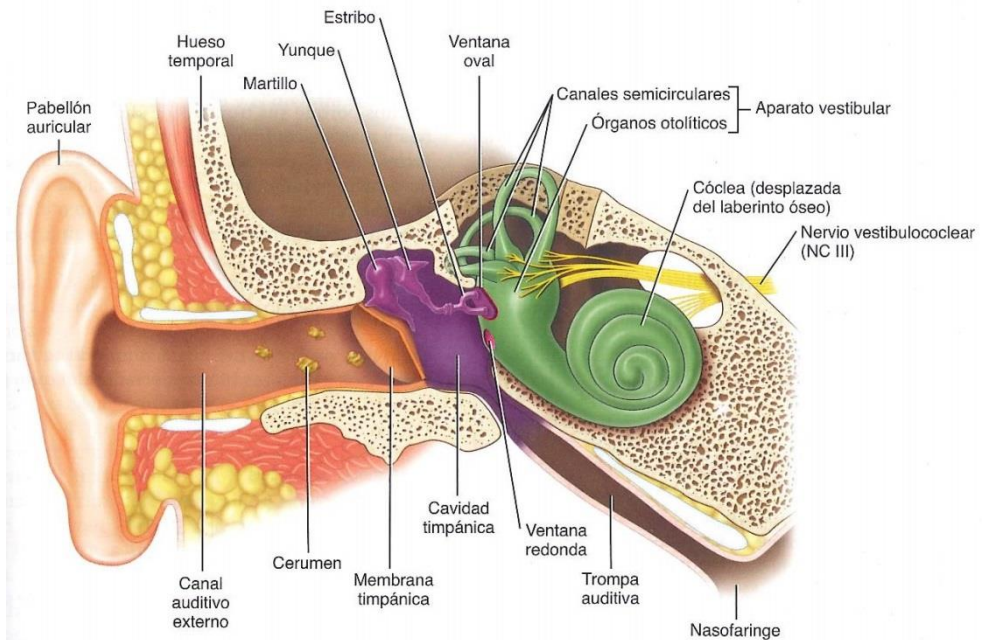


Figura 3.1. Anatomía del oído. Imagen tomada de (McConnell & Hull 2010).

La determinación de la dirección (localización) del sonido depende de la anatomía del oído externo. Excepto para los sonidos que vienen directamente de frente o desde atrás, desde arriba o desde abajo, el sonido alcanza un oído una fracción de segundos antes que el otro y es más fuerte en el oído más próximo que en el más lejano. Estas diferencias son detectadas por el cerebro e interpretadas direccionalmente. En los humanos, la diferencia mínima interaural de tiempo que

podemos percibir es de  $10\mu\text{s}$ . Si el sonido llega a nuestra cabeza desde un lado, dependiendo del ángulo, se puede producir el efecto de difracción para sonidos de baja frecuencia y para altas frecuencias se puede incluso llegar a bloquear el sonido que llega al oído opuesto.

### Oído medio

Excavada en el macizo del hueso temporal se encuentra la cámara de aire del oído medio llamada cavidad timpánica, Figura 3.1. Su pared lateral es la membrana timpánica, que lo separa del conducto auditivo externo. La membrana timpánica vibra cuando es golpeada por las ondas sonoras. Estas vibraciones marcan el paso inicial para la detección de las ondas de sonido. Dentro de la cavidad timpánica y abarcándola se encuentran tres pequeños huesos (martillo, yunque y estribo), que transfieren las ondas de sonido desde la membrana timpánica al oído interno. El estribo se encuentra en contacto con uno de los fluidos contenidos en el oído interno mediante la ventana oval; por tanto, el tímpano y la cadena de huesecillos actúan como un mecanismo para transformar las vibraciones del aire en vibraciones del fluido. Para conseguirlo, es importante que la presión del aire dentro del oído medio sea igual a la presión atmosférica; se consigue gracias a la trompa de Eustaquio, que lo provee de aire procedente de la faringe (McConnell & Hull 2010).

El oído medio tiene tres funciones. La primera consiste en aumentar la presión recibida del tímpano. Esto es importante porque la cóclea está llena de líquido, no de aire. Y la densidad y compresibilidad del líquido coclear es casi 4.000 veces menor que la del aire. Por tanto, si no dispusiéramos de un mecanismo para aumentar la presión, sólo llegaría al interior de la cóclea un 0,1% de la presión timpánica. La segunda función es la de proteger las estructuras del oído interno de ruidos excesivamente fuertes, gracias al estribo, que es capaz de contraerse de forma refleja cuando llega un sonido inferior a 1-2 kHz y con una intensidad superior a 85-90 dB. La tercera función es desarrollada por los músculos del oído medio. Se comportan como un filtro paso baja que reducen la transmisión de los sonidos de baja frecuencia, con una atenuación aproximada de 15 dB por octava en

la zona de 1000 Hz, disminuyendo así el enmascaramiento que éstos producirían sobre frecuencias más altas.

### Oído interno

En el oído interno encontramos, por un lado, el *aparato vestibular* encargado de controlar el equilibrio y la *cóclea*, el órgano de la audición por excelencia.

La cóclea, mostrada en la Figura 3.2 A, es un tubo rígido en forma de espiral, de unos 32-35 mm de largo y un grosor que va desde 4 mm<sup>2</sup> en la base hasta 1 mm<sup>2</sup> en la punta o ápice, lleno con dos fluidos de distinta composición,. En la Figura 3.2 B se muestra una sección transversal de la cóclea en la que se observan tres espacios tubulares que se enrollan uno al lado del otro en torno a vueltas cocleares. El tubo central es el *conducto coclear* lleno de endolinfa. El segundo y tercer tubos, llamados *rampa vestibular* y *rampa timpánica*, contienen el mismo fluido, perilinfa, puesto que se interconectan por una pequeña abertura situada en el vértice del caracol. La base del estribo, a través de la *ventana oval*, está en contacto con el fluido de la rampa vestibular, mientras que la rampa timpánica desemboca en la cavidad timpánica a través de otra abertura, la *ventana redonda*, como se observa gráficamente en la Figura 3.2 A. La pared común entre el conducto coclear y el rampa vestibular se conoce como la *membrana vestibular o de Reissner*, mientras que la pared común entre el conducto coclear y el conducto timpánico de llama *membrana basilar*, Figura 3.2 B (McConnell & Hull 2010).

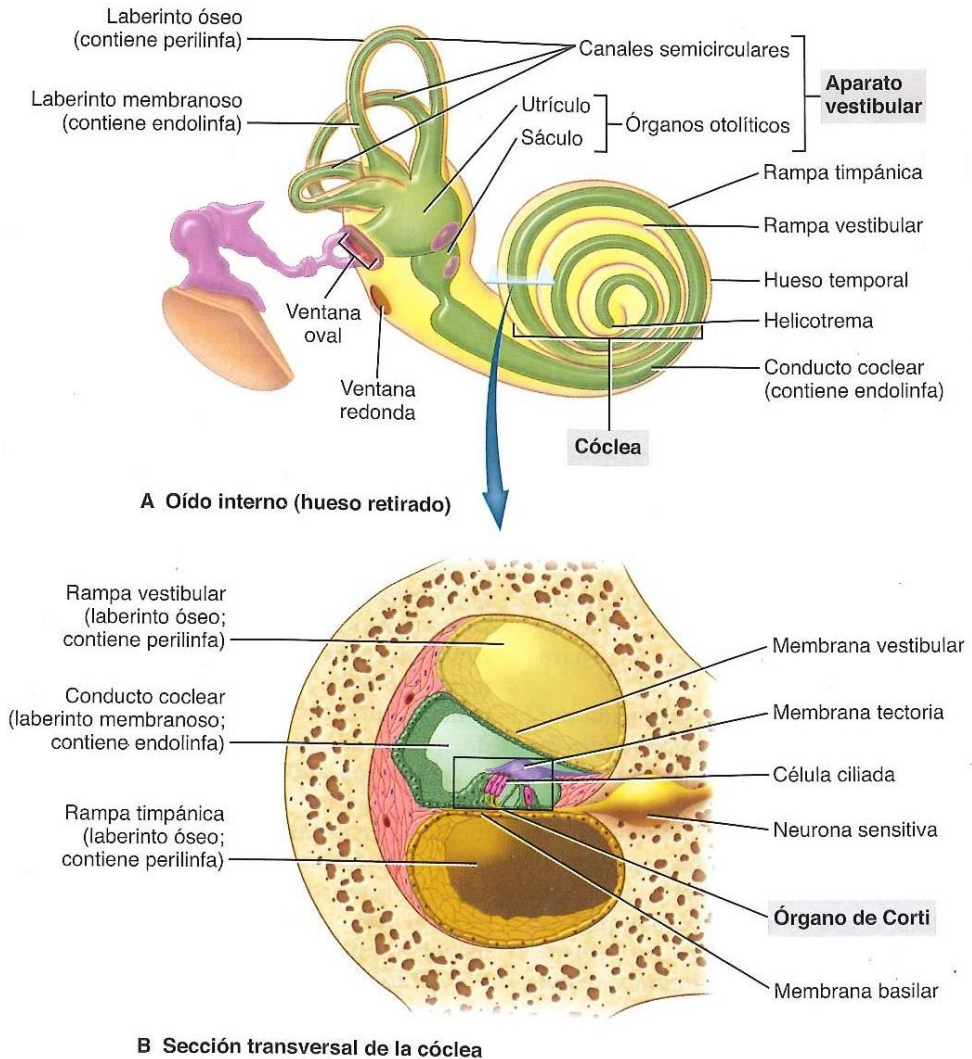


Figura 3.2. A) Anatomía del oído medio e interno; B) sección transversal de la cóclea. Imagen tomada de (McConnell & Hull 2010).

La membrana basilar es una estructura cuyo espesor y rigidez no es constante: cerca de la ventana oval, la membrana es gruesa y rígida, pero a medida que se acerca hacia el vértice de la cóclea se vuelve más delgada y flexible. La rigidez decae casi exponencialmente con la distancia a la ventana oval; esta variación de la rigidez en función de la posición afecta a la velocidad de propagación de las ondas

sonoras a lo largo de ella, y es responsable en gran medida de un fenómeno muy importante: la selectividad frecuencial del oído interno.

La membrana basilar es el soporte del *órgano de Corti*, el elemento más importante de la cóclea, encargado de convertir el movimiento en descargas que activen las fibras nerviosas, Figura 3.3. El *órgano de Corti* contiene entre 15.000 y 30.000 receptores del nervio auditivo: las células ciliadas, de las cuales salen los haces de fibras que componen el nervio auditivo o coclear. Cada célula tiene una serie de cilios con capacidad para producir descargas eléctricas al rozar la membrana superior, la *membrana tectorial*.

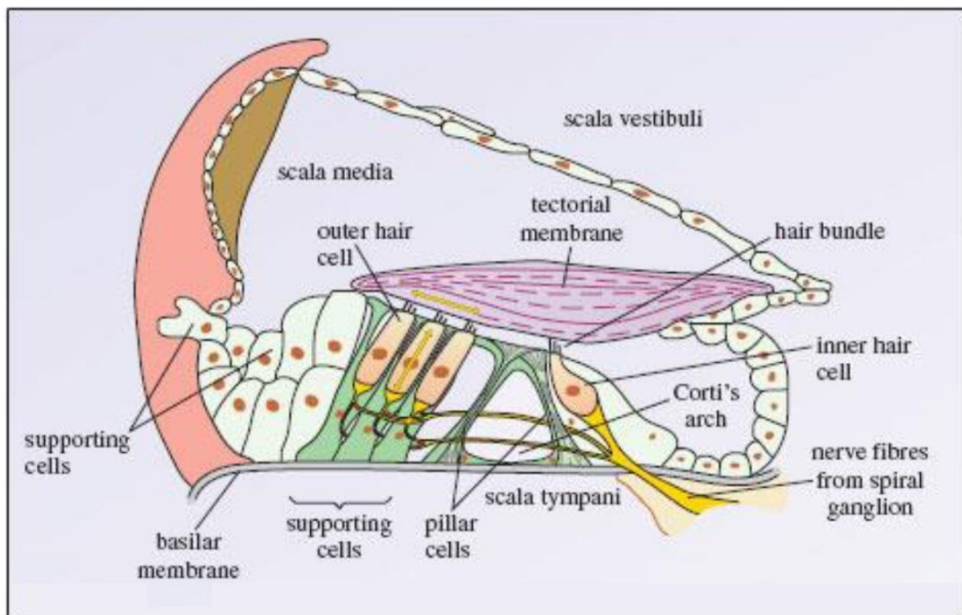


Figura 3.3. Estructura interna del órgano de Corti

Dependiendo de su ubicación en el *órgano de Corti*, se pueden distinguir dos tipos de células ciliadas: internas (*Inner hair cells, IHCs*) y externas (*Outer hair cells, OHCs*). Existen alrededor de 3.500 IHCs y unas 20.000 OHCs. Ambos tipos de células presentan conexiones o sinapsis con las fibras nerviosas aferentes (que aportan impulsos hacia el cerebro) y eferentes (que transportan impulsos provenientes del cerebro), las cuales conforman el nervio auditivo. Sin embargo, la

distribución de las fibras es muy desigual: más del 90% de las fibras aferentes inervan a las IHCs, mientras que la mayoría de las 500 fibras eferentes inervan a las OHCs. La función de cada tipo se expone a continuación.

El funcionamiento de la cóclea comienza porque las vibraciones del estribo provocan vibraciones en el fluido de la ramba vestibular. Las oscilaciones de la perilinfa de la ramba vestibular se transmiten a la endolinfa y de ésta a la membrana basilar; la membrana basilar a su vez, provoca oscilaciones en el fluido de la ramba timpánica, Figura 3.4. Es importante destacar que la amplitud y frecuencia de las vibraciones son directamente proporcionales a la amplitud y frecuencia de las ondas sonoras.

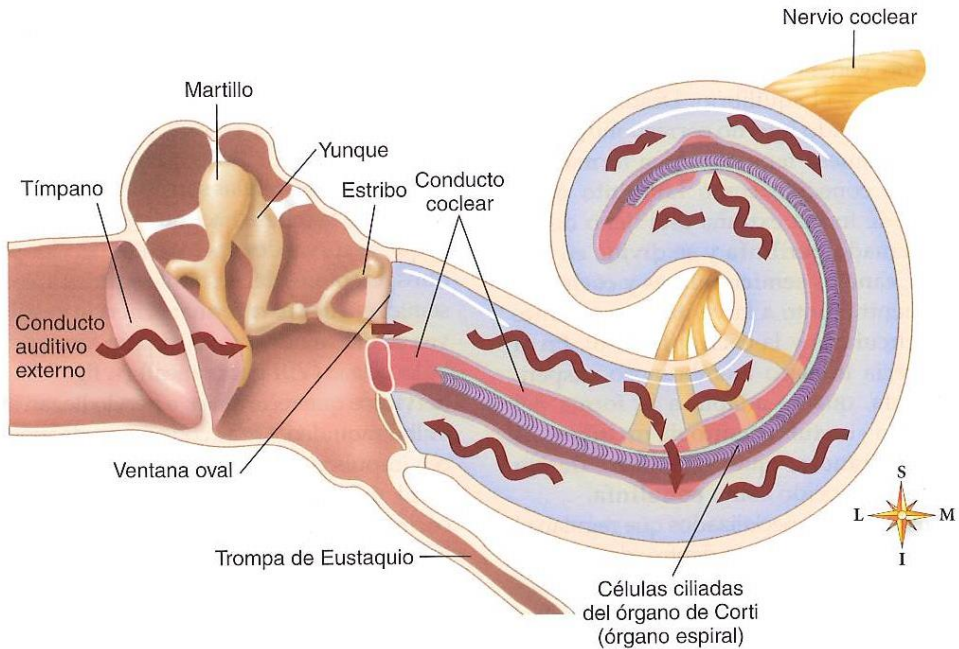


Figura 3.4. Efecto de las ondas sonoras sobre las estructuras del oído. Imagen tomada de (Thibodeau & Patton 2012)

La onda sonora que provoca estas oscilaciones tiene en una región específica de la cóclea un máximo en su amplitud que depende de la frecuencia del sonido y posteriormente tiende a disminuir rápidamente hacia el ápice de la cóclea. Mientras

menor es la frecuencia del tono, mayor es la distancia que viaja la onda a lo largo de la membrana antes de ser atenuada, y viceversa. De esta forma, la membrana basilar dispersa las distintas componentes de una señal de espectro complejo en posiciones bien definidas respecto a la ventana oval, como se muestra gráficamente en la Figura 3.5.

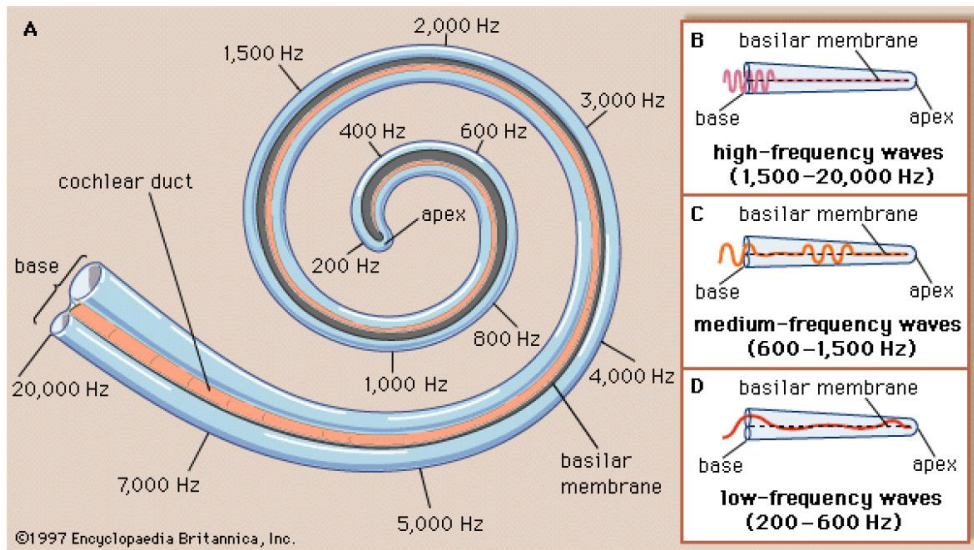


Figura 3.5. Organización tonotópica de la cóclea. A) Distribución tonotópica de la cóclea. B) Localización de la respuesta coclear a altas frecuencias. C) Localización de la respuesta coclear a frecuencias medias. D) Localización de la respuesta coclear a bajas frecuencias.

Lo importante de las ondas que se desplazan por la cóclea es el empuje que ejercen sobre el conducto coclear y por lo tanto, sobre el órgano de Corti. En el órgano de Corti, las células ciliadas descansan sobre la membrana basilar y los extremos de estas células están incluidos en la membrana tectorial, Figura 3.3. Ambas membranas presentan una flexibilidad algo diferente, por lo que cada una de ellas se mueve en relación con la otra cuando la onda de desplazamiento pasa a través de las mismas. Como resultado, los extremos de las células se flexionan de atrás hacia delante, mientras la membrana basilar se mueve en relación con la membrana tectorial. La flexión de los cilios cambia el potencial de membrana de las células ciliadas: la flexión en una dirección hiperpolariza la célula ciliada, mientras que la flexión en la otra dirección la despolariza. Estas alteraciones del

potencial producen cambios en la liberación de neurotransmisores de las células ciliadas en sus sinapsis con una neurona de primer orden. Las moléculas de neurotransmisores liberadas en la sinapsis cambian el potencial de la membrana de una neurona del nervio vestibulococlear, alterando la frecuencia de impulso de los potenciales de acción. Estos potenciales viajan a través del nervio vestibulococlear hasta el cerebro (McConnell & Hull 2010).

La intensidad de la estimulación auditiva depende del número de potenciales de acción por unidad de tiempo y del número de células estimuladas, mientras que la frecuencia percibida depende de la población específica de fibras nerviosas activadas. Existe una asociación específica entre la frecuencia del estímulo sonoro y la sección de la corteza cerebral estimulada. Cuanto menor es la frecuencia de vibración del sonido, más cerca del ápice se produce el máximo desplazamiento de la membrana basilar. Para frecuencias mayores, el máximo desplazamiento se localiza más cerca del ápice de la cóclea.

Dependiendo de la región de la membrana basilar que oscila con mayor amplitud, las células ciliadas de esa área se activan en mayor proporción que sus vecinas, excitando subsecuentemente a las neuronas aferentes que hacen sinapsis con ellas. Este proceso ha dado origen al concepto de frecuencia característica, para describir la forma en que las neuronas de la vía auditiva responden con un umbral especialmente bajo para los sonidos de cierta frecuencia, y tiene un papel fundamental en la discriminación de los tonos de un sonido. Ante un tono puro muy próximo a su frecuencia característica, un lugar concreto de la membrana basilar oscilará con una mayor amplitud. A medida que la frecuencia del tono puro se aleja de la frecuencia característica, la respuesta se debilitará. Se dice por tanto que cada zona de la membrana basilar actúa como un filtro auditivo, que responde ante un rango estrecho de frecuencias. Cuando cualquier tono se duplica en frecuencia, se desplaza una octava, la región que resuena de la cóclea se desplaza alrededor de 3.5 a 4 mm, sin importar la diferencia absoluta entre las frecuencias de la octava; siempre que la frecuencia se multiplica, la posición de resonancia en la cóclea no se multiplica, simplemente se desplaza una cierta distancia. Es decir, son las proporciones de frecuencia y no sus diferencias las que determinan el



desplazamiento de la región de resonancia de la cóclea. En consecuencia, no existe una relación lineal entre la distancia desde la ventana oval a un punto sobre la membrana basilar y la frecuencia de resonancia de ese punto, sino que la relación es exponencial.

Conforme un sonido incrementa su amplitud, aumenta la amplitud de la onda viajera en la membrana basilar incrementándose tanto el número de IHCs que excitan, como la cantidad de potenciales de acción que generan en la vía aferente. La diferencia entre las IHCs y OHCs radica en su función. Mientras las células ciliadas internas se encargan de transformar la amplitud de la onda en potenciales de acción tal y como se explica previamente, la función de las células ciliadas externas es control automático de ganancia. Por lo tanto, debido a las OHCs la respuesta de la cóclea es no lineal, Figura 3.6.

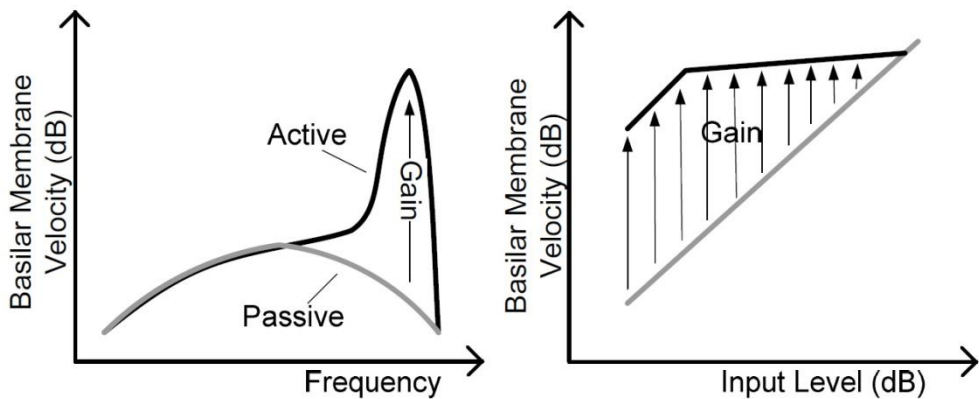


Figura 3.6. Efectos de la no linealidad del comportamiento de la membrana basilar. (a) Respuesta de un punto de la membrana basilar son el efecto de los OHCs (Passive) y con OHCs (Active). CF indica la característica de frecuencia de la sección de la membrana basilar. (b) Nivel de respuesta en función del nivel de entrada de la frecuencia característica de la sección de la membrana basilar. (Hamilton 2008).

Los centros cerebrales superiores categorizan los tonos según la región de la cóclea que se excita, y las amplitudes según el número de neuronas activas y la intensidad con que éstas descargan.

En resumen, en el nervio auditivo se pueden identificar básicamente dos tipos de representación de señales, la representación espectral y la temporal. Esta dualidad se debe principalmente al hecho de que las células de la cóclea, que presenta una organización tonotópica, dan una respuesta en función de la amplitud de la señal y de su envolvente temporal. Se ha descrito que el funcionamiento de la cóclea permite una representación espectral de la señal y los *IHCs* ofrecen una representación temporal. Por lo tanto, un tono a una frecuencia dada se representa en el nervio auditivo tanto por su posición, según la posición tonotópica que mejor responde a esa frecuencia, como por la periodicidad de las respuestas de todas las fibras que responden a ese estímulo (representación temporal).

### 3.1.2. Región central del sistema auditivo

El sistema nervioso auditivo central está formado por las vías auditivas y los sectores de nuestro cerebro dedicados a la audición.

Las señales generadas por el órgano de Corti de la cóclea son enviadas mediante la rama coclear del nervio vestibulococlear, Figura 3.7, a la *corteza auditiva primaria* en la superficie superior del lóbulo temporal, y luego a la *corteza de asociación auditiva próxima*. Las señales pueden tomar diferentes vías desde un oído a la corteza. El 70% de las vías cruzan a la corteza opuesta y el 30% restante va al mismo lado del cerebro que del oído, por lo tanto, cada hemisferio de la corteza cerebral recibe información auditiva de ambos oídos (McConnell & Hull 2010).

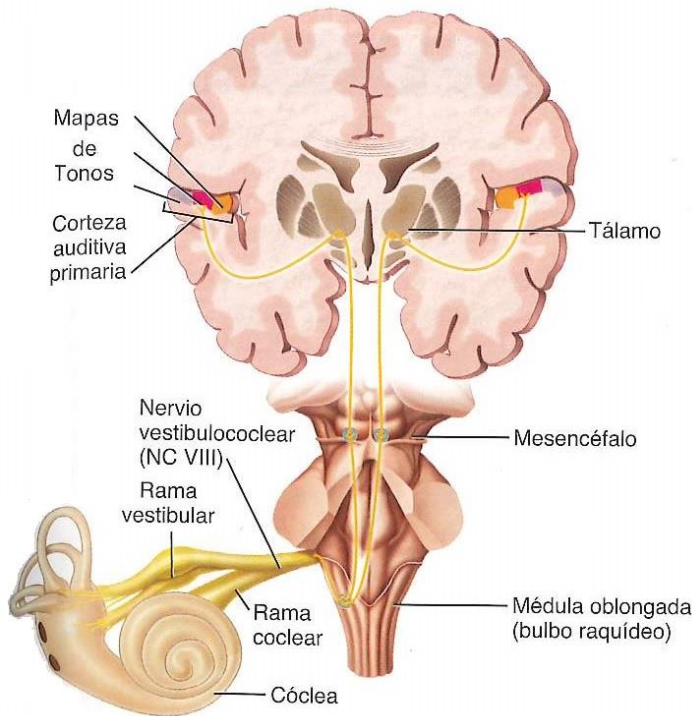


Figura 3.7. Vías Auditivas y corteza auditiva primaria. Imagen tomada de (McConnell & Hull 2010).

La corteza auditiva primaria se divide en *mapas de tonos*, regiones diferentes que son sensibles a un determinado intervalo de frecuencias tal como se muestra gráficamente en la Figura 3.7. Las señales de las células ciliadas de una zona determinada de la membrana basilar (sensibles a un intervalo concreto de frecuencia) se proyectan en la región del tono correspondiente de la corteza cerebral auditiva. El cerebro, por lo tanto, percibe el tono según el área de la corteza activada. La corteza auditiva primaria también puede dividirse en *mapas de volumen*, con algunas neuronas que responden mejor a los potenciales de acción provocados por sonidos fuertes, mientras que otras son más sensibles a los potenciales de acción menos frecuentes de los sonidos suaves.

Después de que nuestro aparato auditivo ha detectado las distintas cualidades del sonido y las trasmite a la corteza auditiva primaria, el cerebro aún debe dar

“sentido” a lo que, en caso contrario, sería un ruido sin sentido. Ésta es la función de la corteza de asociación auditiva próxima. Esta parte de la corteza distingue a los *patrones* de sonido de acuerdo con las complejas variaciones de tono, intensidad y dirección, atribuyéndoles un significado. La corteza de asociación auditiva también recibe información de otras regiones del cerebro, como las cortezas visual y somatosensorial, para colaborar en esta tarea (McConnell & Hull 2010).

### 3.2. Psicoacústica

La acústica es la rama de la física interdisciplinaria encargada de describir las características físicas del sonido mientras que la psicoacústica estudia cómo los humanos percibimos el sonido.

La mayoría de los conceptos de acústica son conocidos: el comportamiento ondulatorio del sonido, la propagación del sonido y los fenómenos que le afectan, los diferentes dominios en los que se puede estudiar el sonido (espacial, temporal y de frecuencia), el rango dinámico, la relación señal ruido, etcétera... En este apartado nos vamos a centrar en exponer los conceptos relacionados con la psicoacústica.

Debido a que la comprensión que se tiene acerca de lo que ocurre en las estructuras cerebrales es muy limitada, especialmente en lo relativo a los centros superiores del cerebro, es necesario recurrir a la descripción psicoacústica de los fenómenos perceptuales y de las sensaciones. A continuación se enumeran los objetivos generales de la psicoacústica:

- Caracterizar la respuesta de nuestro sistema auditivo.
- Obtener el umbral absoluto de la sensación.
- Obtener el umbral diferencial de determinados parámetros de los estímulos: la mínima variación y diferencia perceptibles.
- Comprender y obtener la capacidad de resolución del sistema auditivo, separando estímulos simultáneos para crear sensaciones.

- Entender la variación temporal de la sensación del estímulo.

En resumen, el estudio de la audición a través de las respuestas subjetivas a los estímulos acústicos, especialmente en tareas de detección y discriminación, es el objetivo de la psicoacústica o psicofísica auditiva.

Hay que destacar la diferencia entre los conceptos de *detección*, *discriminación* e *identificación*. La detección implica notar la presencia, o ausencia, de un estímulo, pero sin llegar a identificarlo; se pueden detectar estímulos en función de su duración, intensidad y frecuencia. En la discriminación se comparan y se buscan diferencias entre estímulos próximos. Finalmente en la identificación, se relaciona el estímulo que se presenta con una representación que tenemos en la memoria, a la cual corresponde una “etiqueta” determinada.

### 3.2.1. Umbrales absolutos y umbrales diferenciales

El oído presenta unos límites en la identificación de las tres cualidades del sonido: frecuencia, intensidad y duración. Cuando se estudia la percepción humana además de conocer los valores de los umbrales absolutos, también son importantes las Diferencias Mínimas Perceptibles (DMP) porque muestran la capacidad de resolución del oído y los límites de la audición. Las DMP consisten en los valores menores para que un cambio en un estímulo sea perceptible.

En cuanto a las frecuencias, generalizando, podemos oír los sonidos cuya frecuencia pertenece al rango 20 a 20000 Hz, pero somos especialmente sensibles a los que se sitúan entre 2500 y 5000 Hz. El rango varía de unas personas a otras, disminuyendo a lo largo del tiempo, en particular la habilidad de escuchar las altas frecuencias se pierde con la edad.

La resolución frecuencial del oído depende de la intensidad y de la frecuencia de dichos sonidos. Para tonos sobre los 200Hz, podemos discriminar sonidos con diferencias mínimas de 1Hz. Para frecuencias superiores, la resolución frecuencial aumenta, por ejemplo, para un tono de 10000Hz, la resolución frecuencial es de 200Hz. Esto se debe a que el sistema auditivo actúa como un conjunto de filtros

superpuestos en los que estos filtros son más estrechos en frecuencias graves y más anchos en frecuencias agudas, y van a definir las llamadas bandas críticas (descritas en el siguiente apartado) (Scott 2004).

Respecto a las intensidades, en un ambiente silencioso, los humanos podemos oír desde 0 dB-SPL (*Sound Pressure Level*), que se considera la intensidad mínima para distinguir un sonido del silencio. Pero como es complicado encontrar entornos silenciosos, se suele proponer como límite inferior de audición el ruido ambiente del entorno determinado. La mayor intensidad que podemos oír es del orden de 120 a 130dB-SPL. Los sonidos con intensidades mayores provocan dolor y pueden llegar a dañar el oído. Respecto al umbral diferencial de la intensidad entre dos sonidos, la diferencia mínima perceptible es conocida como JND (*Just Noticeable Difference*) cuyo valor es sobre 1dB (Scott 2004).

Sobre la duración sólo existe un límite inferior. El sonido más breve perceptible puede oscilar entre 10 y 40 ms y la mayor sensibilidad natural aparece en el rango de 40 a 60 ms (López Bascuas 1997). Por lo tanto, la resolución temporal del oído es buena para estímulos entre 10 y 100 ms. 20 ms es el tiempo característico de integración en el procesamiento auditivo. Para esa trama temporal, la señal es cuasi-estacionaria y se puede analizar como tal.

Hay que considerar lo que ocurre cuando se entrecruzan estas tres categorías, y especialmente las dos primeras, porque no todas las frecuencias requieren la misma intensidad para ser percibidas. La Figura 3.8, conocida como curva de Wegel, muestra los umbrales de la audición humana respecto la frecuencia y la intensidad y, dentro de ellos, los márgenes utilizados habitualmente por la música y el lenguaje articulado.

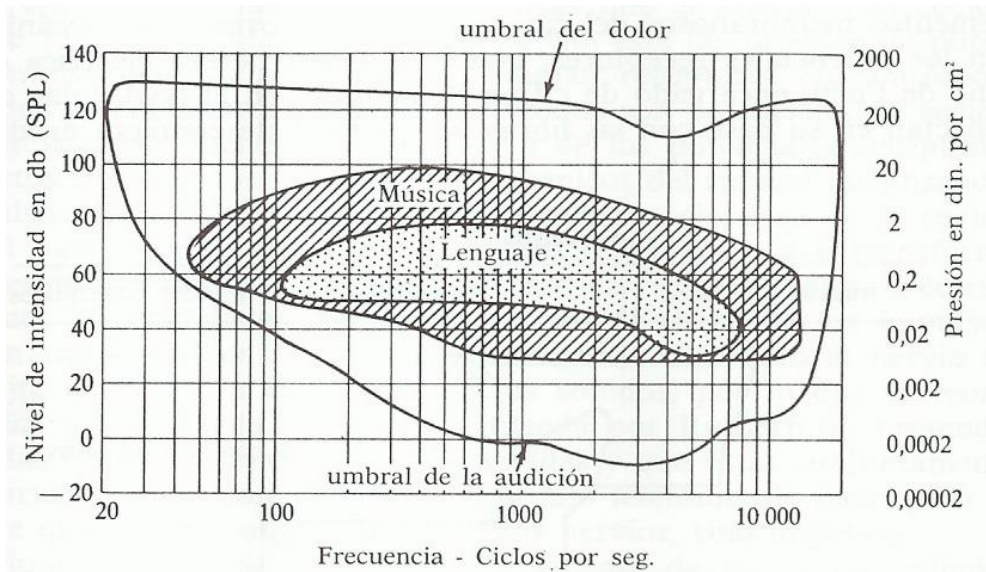


Figura 3.8. Curva de Wegel. Curva en la que se representan los umbrales de audición respecto a la frecuencia e intensidad del sonido

En la psicoacústica se estudia otra característica de intensidad, la intensidad con la que percibimos los sonidos, llamada sonoridad (*loudness*) (Scott 2004). Ésta varía según la persona y no es directamente medible, y además la sonoridad es dependiente de la frecuencia. El Instituto Nacional Americano de Estándares (*American National Standards Institute, ANSI*) ha definido la sonoridad como: “El atributo de la sensación auditiva en términos los que sonidos pueden ser ordenados en una escala de silencio a alto”.

La métrica más famosa de la sonoridad fue desarrollada por Fletcher y Munson en su trabajo (Fletcher & Munson 1933). Definieron esta métrica a partir de la siguiente afirmación: cualquier sonido de 1000Hz de frecuencia y 40dB-SPL de intensidad tiene una sonoridad de 40phons. Para representar de forma gráfica la sonoridad respecto a la frecuencia e intensidad de los tonos se usa unas curvas de igual sonoridad, denominadas curvas isofónicas de Fletcher-Munson. Estas curvas calculan la relación existente entre la frecuencia y la intensidad (en decibelios) de dos sonidos, para que éstos sean percibidos como igual de fuertes por el oído, de

manera que todos los puntos sobre una misma curva isofónica tienen la misma sonoridad (Scott 2004).

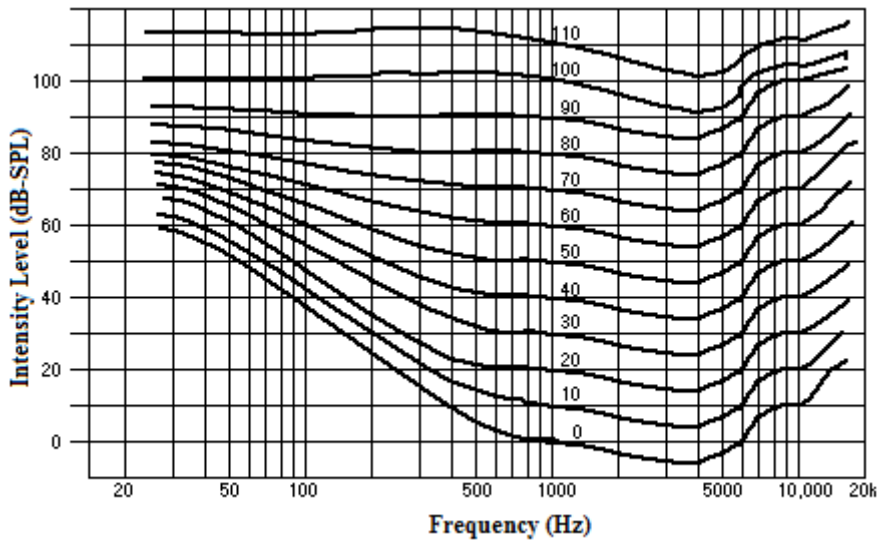


Figura 3.9. Curvas de igual sonoridad de Fletcher-Munson

También existe una relación entre la duración y la sonoridad de un sonido. Un sonido con una intensidad constante disminuye en sonoridad en un 25% en el transcurso de minutos (Scott 2004).

### 3.2.2. Enmascaramientos y bandas críticas

El enmascaramiento es el efecto por el que un sonido es ocultado por otro. ANSI en 1960 definió este fenómeno como el proceso por el cual el umbral de audibilidad de un sonido aumenta debido a la presencia de otro sonido, al que se le denomina máscara. Siendo este efecto una respuesta no lineal del sistema auditivo.

Se ha descrito en este capítulo la capacidad del sistema auditivo, concretamente la cóclea, para discriminar frecuencias de los sonidos. Una característica importante del sistema auditivo es el efecto de enmascaramiento (O'Shaughnessy 2000), según el cual la percepción de los sonidos se ve oscurecida o impedida por



la presencia de otros sonidos. Ya se ha hablado previamente de la resolución frecuencial, que implica que cuando dos sonidos tienen sus frecuencias por debajo de la resolución frecuencial, sólo escuchamos un único sonido. A este fenómeno se le denomina enmascaramiento frecuencial, en el que el sonido de menor frecuencia enmascara el sonido de mayor frecuencia. Y por otro lado, está el enmascaramiento temporal, si los dos sonidos se producen con un cierto retraso. El efecto de enmascaramiento es importante por su influencia en la no linealidad del sistema auditivo y perceptivo humano.

Las investigaciones del enmascaramiento simultáneo guió a Fletcher a proponer la hipótesis de que una porción del sistema auditivo humano se comporta como un conjunto continuo y solapados de filtros paso de banda (Fletcher 1940). Cada uno de estos filtros debe corresponderse con una zona localizada en la membrana basilar. Aproximadamente, cada sección se corresponde con un largo de 0.9 mm de la membrana basilar. El rango de estos filtros que estimulan la misma porción de la membrana basilar se denomina *banda crítica* y al ancho de cada banda crítica se le conoce como *ancho de banda crítica*.

Este banco de filtros no sigue una configuración lineal, el ancho de banda y morfología de cada filtro depende de su frecuencia central. Los filtros correspondientes al extremo más próximo a la ventana oval y al tímpano responden a altas frecuencias, ya que la membrana es rígida y ligera. Por el contrario, en el extremo más distante, la membrana basilar es pesada y suave, por lo que los filtros correspondientes responden a las bajas frecuencias.

Aunque algunos estudios describen un número explícito de bandas críticas con unos rangos bien definidos, como la escala de *Bark* (Zwicker 1961), la escala de *Mel* (Stevens 1937) y escala *ERB* (*Equivalent Rectangular Bandwidth*) (Glasberg & Moore 1990), otros estudios han sugerido que la cóclea es un conjunto de filtros continuos y que es apropiado situar las bandas críticas en cualquier frecuencia audible.

Gracias a los datos obtenidos por este estudio psicoacústico de las bandas críticas en conjunción con el comportamiento de descomposición frecuencial de la

cóclea biológica, decidimos optar en nuestra implementación por generar tantas bandas críticas como el hardware que tenemos disponible nos lo permita y distribuidas logarítmicamente.

### 3.2.3. Frecuencia, tono y timbre

De la misma forma que se diferencia entre la intensidad de un sonido y la sonoridad, es decir, la intensidad con la que percibimos los sonidos, también se puede diferenciar entre el concepto de frecuencia del sonido y la frecuencia con la que se percibe. Las componentes frecuenciales de un sonido es información objetiva, en cambio, la interpretación subjetiva de la frecuencia de un sonido se denomina *tono*. El tono ha sido definido por ANSI como “el atributo de la sensación auditiva en términos de que el sonido puede ser ordenado en una escala de agudo a grave”. Para sonidos complejos como los producidos por notas musicales, que están formados por la frecuencia fundamental y armónicos, el tono queda determinado por la frecuencia fundamental.

En el presente trabajo se exponen sistemas de reconocimiento del tono de un sonido tanto para tonos puros como notas musicales. En los tonos puros, vamos a usar como sinónimo frecuencia y tono. Para las notas musicales usaremos el concepto de tono para definir la frecuencia fundamental de la nota.

El timbre es definido por ANSI como “el atributo de la sensación auditiva que nos permite juzgar si dos sonidos con la misma sonoridad y tono son diferentes”. Por lo tanto, el timbre es la cualidad que nos permite distinguir entre dos instrumentos musicales que generan la misma nota. Aunque las dos notas tengan la misma frecuencia fundamental, el espectro es diferente debido a los armónicos.

Mientras que los sonidos se pueden ordenar respecto el tono y la sonoridad, el timbre no es un concepto de ordenación.

### 3.3. Modelos del sistema auditivo

Tal como se ha expuesto en los apartados anteriores, una porción del sistema auditivo humano se comporta como un conjunto continuo y solapado de filtros paso de banda, que se corresponden con una zona localizada en la membrana basilar. En este apartado se exponen los modelos físicos más relevantes sobre dicho comportamiento de la cóclea. Algunos modelos, además de modelar el comportamiento de la cóclea, también modelan otras partes del oído.

Se pueden diferenciar entre dos tipos de modelos auditivos dependiendo de la estructura de conexión que tienen los filtros: paralela (filtros independientes) o en cascada (conjunto de filtros acoplados).

En la respuesta de las bandas críticas de la membrana basilar se observa un pico muy pronunciado que coincide con la frecuencia característica y una caída en la respuesta para frecuencias por encima y por debajo de la frecuencia característica, siendo más pronunciada la pendiente en las frecuencias superiores. Este efecto se consigue fundamentalmente con un banco de filtros en cascada. Un banco de filtros paralelos, necesita filtros de un orden mayor.

#### 3.3.1. Modelo ERB (Equivalent Rectangular Bandwidth)

El modelo ERB o modelo Patterson-Holdsworth (Slaney 1993), está basado en los trabajos de Patterson y Holdsworth sobre la cóclea (Patterson et al. 1992). Consiste en un conjunto de filtros paso banda en estructura paralela o independiente. Cada uno de ellos sintonizados a una frecuencia diferente. En este modelo el ancho de banda de cada filtro coclear está descrito por un ancho de banda rectangular equivalente (Equivalent Rectangular Bandwidth, ERB). La idea de esta medida es aproximar las bandas críticas de la cóclea mediante filtros paso de banda por un rectángulo equivalente cuya altura es el máximo de la respuesta en magnitud del filtro y cuya área es igual a la respuesta de dicho filtro, de manera que permite en su interior la misma cantidad de energía. Esta relación se muestra

en la Figura 3.10. Un filtro de bandas críticas o ERB modela la señal que está presente en una única célula del nervio auditivo.

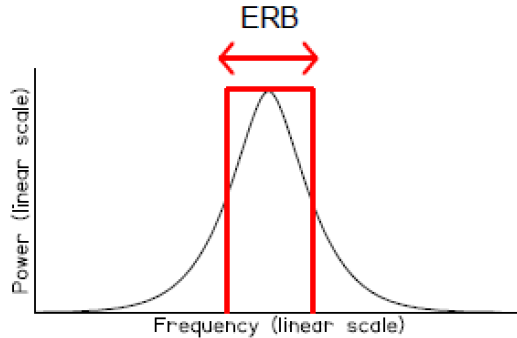


Figura 3.10. Representación de la medida ERB (Miró-Amarante 2013)

Como la arquitectura es paralela, no existe ninguna dependencia entre filtros sucesivos como ocurre en un banco de filtros en cascada.

Este modelo está basado en filtros *gammatone* con respuestas al impulso. La importancia de estos filtros para la audición reside en que pueden generar una respuesta en frecuencia muy parecida a la de los filtros auditivos humanos, obtenidos de forma perceptual por Patterson. Es más, son capaces de indicar como se mueve la membrana basilar frente a un estímulo dado. Un posible inconveniente de este modelo basado en filtros *gammatone* es que las respuestas son muy simétricas, es decir, no existe diferencias entre las pendientes de atenuación ascendente y descendente respecto de cada frecuencia característica.

En este modelo no se especifica la separación frecuencial entre canales. Ni se realiza control de ganancia ni adaptación.

### 3.3.2. Modelo de Lyon

Richard F. Lyon desarrolló un modelo coclear basado en el conocimiento del funcionamiento de la cóclea (Lyon 1982). Este modelo describe la propagación del

sonido en el oído interno y la conversión de la energía acústica en representaciones neuronales.

El modelo coclear descrito por Lyon combina una serie de filtros que modelan la onda de presión viajera, rectificadores de media onda, *HWR* (*Half Wave Rectifiers*) que detectan la energía de la señal actuando como las *IHCs* y distintas etapas de control automático de ganancia, *AGC* (*Automatic Gain Control*), modelando el comportamiento de las *OHCs*. Debido a que la frecuencia característica de la membrana basilar decrece exponencialmente de la base al ápice, se divide la membrana basilar en segmentos de la misma longitud para obtener la distribución frecuencial de los filtros.

En cada punto de la cóclea la onda acústica es filtrada por un filtro *notch*<sup>16</sup>. Cada filtro notch opera en una frecuencia satisfactoriamente baja, de manera que el efecto global es un filtrado paso baja gradual. Un resonador adicional (filtro paso banda) deja pasar una pequeña parte de la energía de la onda viajera y modela la conversión del movimiento de la membrana basilar que es detectado por las *IHCs*. El diagrama de bloques que representa esta arquitectura en cascada se muestra en la

Figura 3.11. De forma que las componentes de la señal de alta frecuencia son filtradas mientras la señal viaja a través de la cascada de filtros. La eliminación de las componentes de alta frecuencia da como resultado que la respuesta frecuencial de los filtros tenga una pendiente empinada en el rechazo de las frecuencias bajas. En la cóclea biológica también está presente esa pendiente en su respuesta frecuencial. Por lo tanto, este modelo provee una buena aproximación al procesamiento que se realiza en la cóclea biológica. En la Figura 3.12 se observa la función de transferencia del filtro notch, del resonador y de la combinación de ambos.

---

<sup>16</sup> El filtro notch, también conocido como “filtro elimina banda”, “filtro trampa” o “filtro de rechazo de banda” es un filtro electrónico que no permite el paso de señales cuyas frecuencias se encuentran comprendidas entre las frecuencias de corte superior e inferior.

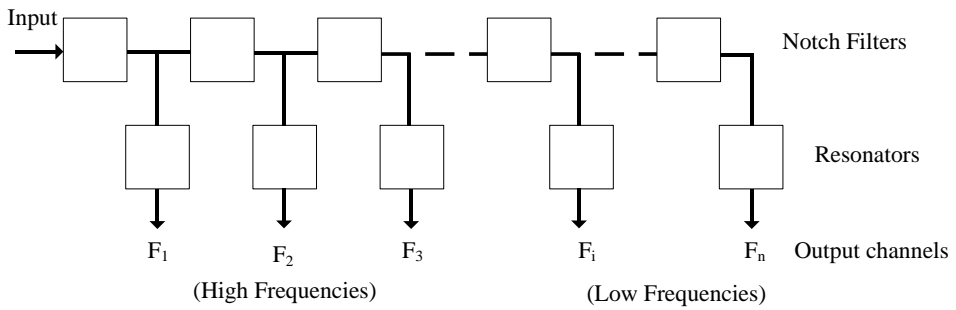


Figura 3.11. Diagrama de bloques de los filtros en el modelo de Lyon

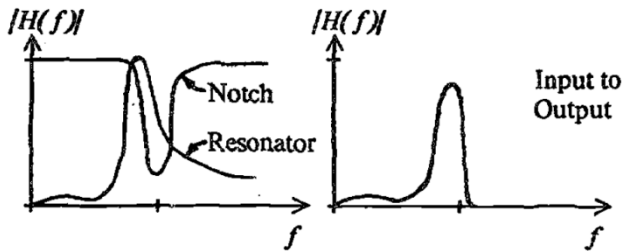


Figura 3.12. Función de transferencia de los filtros usados en el banco de filtros. Imagen tomada de (Lyon 1982)

En la Figura 3.13 se presenta la respuesta de este modelo de 64 secciones para una frecuencia de muestreo de 8 kHz. Es interesante resaltar como existe una diferencia clara entre las pendientes ascendentes y descendentes respecto la frecuencia característica. También se observa que las pendientes de las curvas después de la frecuencia fundamental son mayores mientras mayor es la frecuencia característica, esto se debe a que la eliminación de las componentes de altas frecuencias se produce conforme la señal recorre los filtros. La atenuación de la señal a bajas frecuencias se debe a incluir un filtro de preénfasis. Este filtro de preénfasis es un filtro paso de alta que modela una aproximación a la respuesta en frecuencia del oído externo y medio.

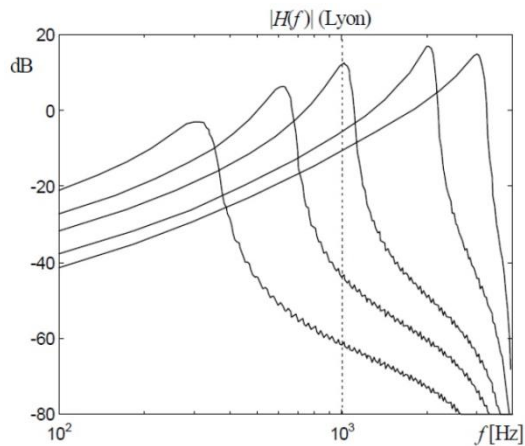


Figura 3.13. Respuesta en frecuencia del modelo de Lyon (64 secciones) para frecuencias características: 3.0, 2.0, 1.0, 0.6 y 0.3 KHz (Miró-Amarante 2013)

### 3.3.3. Modelos de Lyon y Katsiamis

Richard Lyon junto con Andreas Katsiamis y Emmanuel Drakakis publican un artículo de investigación en el 2007, donde presentan funciones de transferencia en el dominio continuo diseñadas a partir del filtro gammatone para el procesamiento auditivo (Katsiamis et al. 2007). En él se muestra el diseño de dos tipos de filtros, el filtro gammatone todos-polos diferencial, DAPGF (Differentiated All-Pole Gammatone Filter) y el Filtro Gammatone Un-Cero, OZGF (One-Zerod gammatone Filter). Estos dos diseños se caracterizan por presentar una arquitectura orientada a la implementación hardware, que cuenta con las mismas propiedades de operación de la cóclea y supera algunas limitaciones del uso de filtros gammatone, como son la simetría de su respuesta en frecuencia y su complejidad en la descripción en el dominio de la frecuencia (Miró-Amarante 2013).

A diferencia de la arquitectura en cascada del modelo de Lyon, este modelo se basa en un banco de filtros compuesto por etapas en paralelo, las cuales se componen de bloques conectados en cascada. En la Figura 3.14 se muestra como la membrana basilar puede ser modelada tanto con una arquitectura paralela como con una arquitectura en cascada.

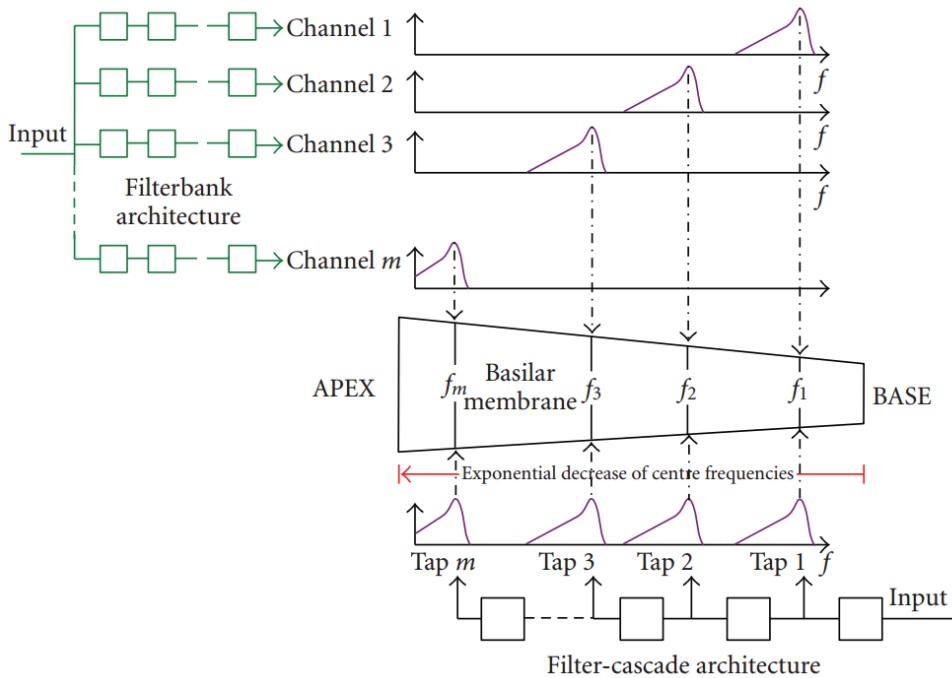


Figura 3.14. Representación gráfica del banco de filtros en arquitectura de Lyon- Katsiamis y en arquitectura en cascada. Las frecuencias de corte de las diferentes etapas (en el gráfico *taps*) están distribuidas exponencialmente. Imagen tomada de (Katsiamis et al. 2007).

### 3.3.4. Modelo de las células ciliadas internas

Para crear un modelo neuromórfico completo de la cóclea, se han añadido elementos que imitarán el comportamiento de las células ciliadas internas (IHCs) al modelo básico de la cóclea, encargadas de transformar la vibración mecánica de la membrana basilar en señales eléctricas (proceso descrito en el apartado 3.1.1).

Existen varios modelos de las IHCs, pero, sin duda, el más conocido y utilizado es el propuesto por Meddis. En 1986, Meddis propone un modelo de simulación por computador de la IHC, (Meddis 1986), (Meddis 1988) (Meddis et al. 1990) y . Este modelo es ampliamente aceptado y se va a convertir en la base de posteriores modelos implementados en aVLSI.



### 3.4. Sistemas electrónicos auditivos bioinspirados

Las cócleas artificiales modelan la membrana basilar usando un conjunto de filtros o resonadores con frecuencias de corte/media (dependiendo que se implementen con filtros de paso de baja o filtros de paso de banda), que imitan la distribución de frecuencias a lo largo de la membrana basilar. Las características de flexibilidad y anchura dependiente de la posición en la membrana basilar se implementan con cambios en los parámetros de los filtros.

Como se ha presentado en los apartados anteriores, la cóclea biológica tiene una gran complejidad, complejidad que implica un alto coste de implementación. Este hecho ha dado lugar a que las cócleas artificiales modelen solo algunas de las características de la cóclea biológica, características seleccionadas dependiendo de la aplicación en la que se quiera integrar el dispositivo. Por lo tanto, la mayoría de las cócleas artificiales no permiten obtener unos resultados comparables con los de una cóclea biológica.

Desde el primer diseño de cóclea artificial, propuesto por Richard Lyon y Carver Mead en el año 1988 (Lyon & Mead 1988), ha aumentado ampliamente la actividad referida a la implementación de diferentes modelos matemáticos de la cóclea utilizando tecnología VLSI analógica, y en menor número, implementaciones digitales utilizando dispositivos lógicos reconfigurables.

Pero a pesar de las dos décadas de investigación, las cócleas artificiales están todavía muy lejos de compararse con la cóclea biológica, sobre todo en aspectos de consumo, rango de frecuencias, rango dinámico de entrada o inmunidad al ruido. Considerando que estamos pretendiendo construir un sistema que ha evolucionado durante millones de años, se han conseguido buenas aproximaciones.

En la Figura 3.15 se muestra gráficamente las diferentes implementaciones de cócleas artificiales y su evolución.

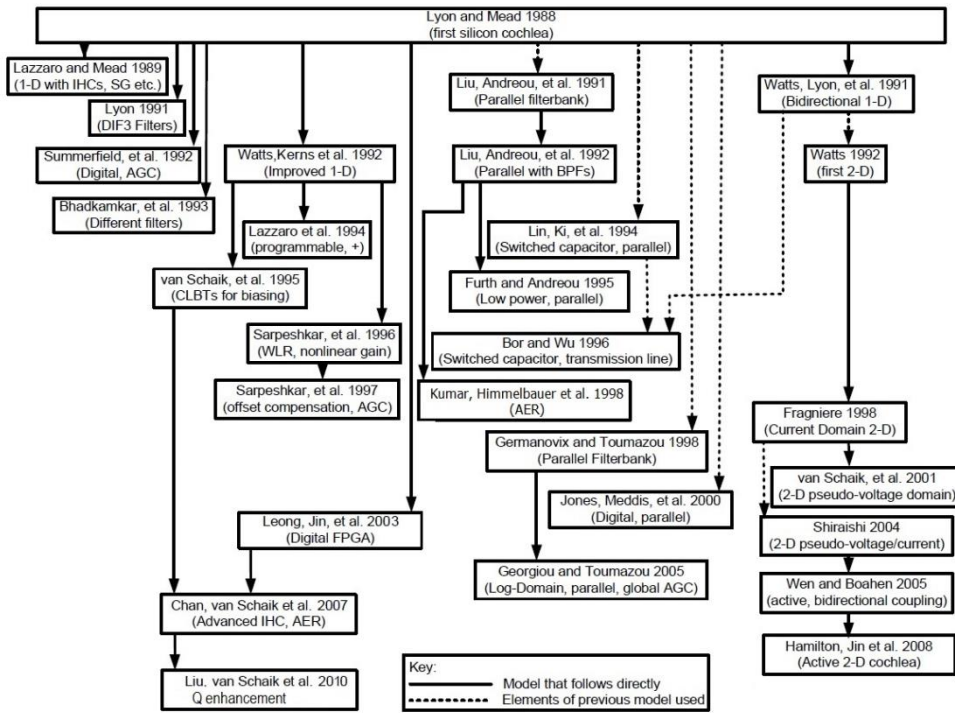


Figura 3.15. Árbol de la evolución histórica de las cócleas artificiales. Figura tomada de (Hamilton 2008) y modificada en el contexto de este trabajo.

Se pueden realizar muchas clasificaciones de las cócleas artificiales, pero a lo largo de la exposición de este estudio, se van a hacer referencia a las dos clasificaciones siguientes:

- Cócleas artificiales unidimensional (1-D, one-dimensional silicon cochlea) y bidimensionales (2-D, two-dimensional silicon cochlea). Las cócleas 1-D modela la propagación de la onda longitudinalmente, mientras que la cóclea 2-D modela el fluido dentro del conducto coclear, teniendo en cuenta también la propagación de la onda verticalmente.
- Cócleas artificiales activas o no activas, según la existencia, o no, de un control automático de ganancia, para permitir que los filtros se adapten dinámicamente a los cambios de intensidad de la entrada.

A continuación, resumimos las implementaciones más relevantes tanto de las cócleas analógicas como de las digitales.

### 3.4.1. Cócleas analógicas con diseño en cascada

El primer desarrollo de una cóclea artificial bioinspirada fue realizado por Lyon y Mead en 1988 (Lyon & Mead 1988). Está basada en el modelo de Lyon, descrito en el apartado 3.3.2 (Lyon 1982). Usa 480 secciones de filtros paso de baja de segundo orden en cascada, con frecuencias características distribuidas logarítmicamente para modelar la propagación de la onda y el análisis frecuencial asociado a la membrana basilar. La Figura 3.16 muestra un esquema de la estructura en cascada empleado por Lyon y Mead para una cóclea analógica de 100 etapas.

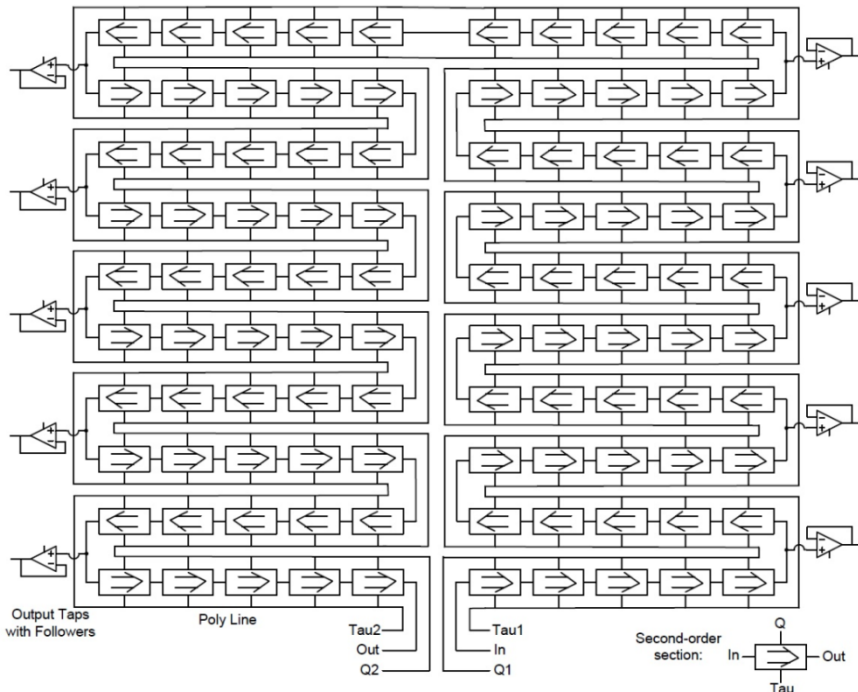


Figura 3.16. Estructura en cascada de los filtros que forman a la cóclea artificial de Lyon y Mead de 100 etapas (Lyon & Mead 1988)

John Lazzaro incluyó en esta cóclea circuitos que modelan los IHCs. Estos circuitos proveen de una codificación a los spikes de salida de la cóclea artificial (Lazzaro et al. 1989).

La cóclea artificial de Lyon y Mead ha sido un éxito en el modelado de diversas características de la cóclea biológica, y ha proporcionado un punto de partida para las investigaciones sobre las cócleas artificiales con circuitos neuromórficos. Hasta la fecha, hay muchas implementaciones basadas en este primer diseño planteado por Lyon.

Una de estas implementaciones es el trabajo de (Watts et al. 1992). Esta versión obtiene mejor distribución exponencial de las frecuencias características, que incrementa el rango lineal y elimina la gran inestabilidad en la señal. La Figura 3.17 muestra la respuesta frecuencial de cada una de las etapas de la cóclea antes de la mejora de los circuitos y después. Estas medidas indican que hay mayor uniformidad en la respuesta frecuencial en varias etapas de la cascada de filtros.

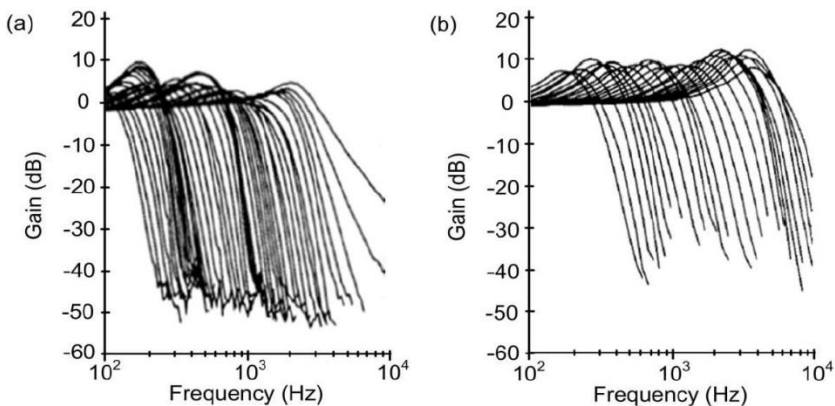


Figura 3.17. Respuesta frecuencial de los filtros de la cóclea artificial de Lyon y en (b) la versión mejorada de Watts et.al. (Watts et al. 1992).

En 1995, Lazzaro y Wawrzynek (Lazzaro & Wawrzynek 1995) proponen una evolución de la versión de Watts, añadiendo el protocolo de comunicación AER en la salida de la cóclea artificial. Es la primera cóclea artificial con este tipo de

comunicación, y en la actualidad, este protocolo está altamente extendido en los desarrollos neuromórficos.

En 1996, se realiza otra implementación de la cóclea artificial propuesta por Watts, expuesta en el trabajo (van Schaik et al. 1996). En la Figura 3.18 se muestra la distribución de las frecuencias de corte de esta implementación respecto a la distribución de las frecuencias de corte de la implementación de Watts. Se observa como en la implementación de van Schaik et al. existe uniformidad en la distribución exponencial de las frecuencias. Gracias a esta mejora, esta cóclea artificial permite procesamiento de sonidos biaurales, tal y como se muestra en los trabajos (Chan et al. 2007) y (Yu et al. 2009), en los que se usan dos cócleas artificiales de van Schaik, ampliadas con módulos más avanzados para gestionar los spikes de salida mediante el protocolo AER, para hacer experimentos de localización. La cóclea artificial expuesta en (Chan et al. 2007), también se ha usado en un sistema de clasificación entre dos tipos de sonidos: una palmada o bombo (Jackel et al. 2010). Esta cóclea ha sido ampliada en número de canales, tasa de eventos de salida y en el rango de frecuencias en el trabajo (Liu et al. 2010).

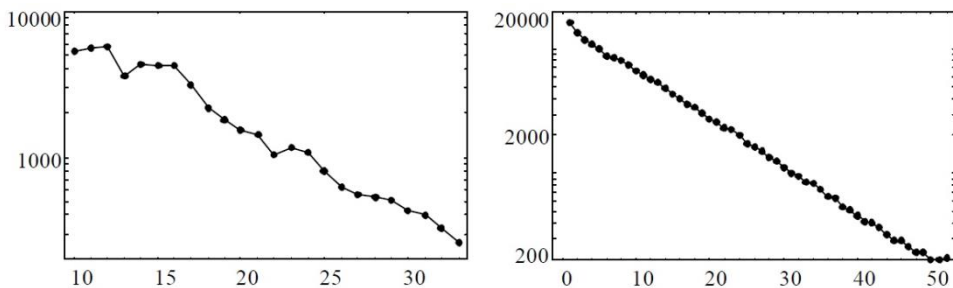


Figura 3.18. Distribución de las frecuencias de corte (Hz) para la cóclea artificial de Watts et. al. (izquierda) y de van Schaik (derecha) (van Schaik et al. 1996).

El principal inconveniente del diseño en cascada es su poca tolerancia a fallos ya que si un elemento falla, este error se propagará al resto de elementos posteriores. También, hay que destacar que cada segmento va a añadir un cierto retraso a la señal de entrada, que será inversamente proporcional a la frecuencia

central de cada filtro. Esto va a producir un retraso, sobre todo en los segmentos correspondientes a las bajas frecuencias, que son los que se encuentran al final de la cascada. Esto implica que el número de secciones (etapas) está limitado el tiempo de respuesta del sistema. Por otro lado, el ruido generado internamente por los filtros también se va acumulando a lo largo de la estructura, lo que provocará una reducción en el rango dinámico del sistema. Algunos de estos inconvenientes se resuelven con organización en paralelo o con estructura 2-D.

### 3.4.2. Cócleas analógicas con diseño paralelo

A menudo se elige la estructura paralela por su fácil implementación. Pero aunque este tipo de modelos no presenta los inconvenientes propios del modelo unidimensional en cascada, no es el preferido en los desarrollos de cócleas analógicas porque cada filtro actúa de modo independiente y para crear el mismo efecto de ‘pendiente pronunciada’ en las altas frecuencias, se necesitaría filtros de un orden mayor, lo cual implica un aumento considerable en el área sobre el silicio y consumo de potencia del sistema. En la siguiente figura se compara la salida de un único filtro de segundo orden tanto en el modelo 1-D paralelo, Figura 3.19 (a), como en el modelo 1-D en cascada, Figura 3.19 (b).

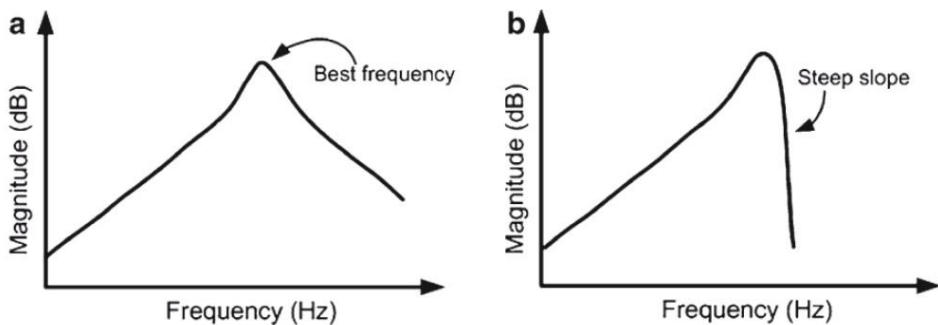


Figura 3.19. Respuesta de un filtro de segundo orden, en el modelo de cóclea paralelo (a) y en el modelo de cóclea en cascada (b)

La implementación expuesta en el trabajo (Liu et al. 1991) y su expansión (Liu et al. 1992) son ejemplos de arquitecturas en paralelo. Siguiendo la arquitectura

expuesta en estos trabajos, se desarrolló en 1998 otra cóclea artificial para su uso en la extracción de características para el reconocimiento del habla (Kumar et al. 1998).

### 3.4.3. Cócleas analógicas bidimensionales (2-D)

Las estructuras de los filtros en este tipo de cócleas artificiales se basan en el modelo Lyon y Katsiamis, explicado en el apartado 3.3.3 de este trabajo. Esta estructura combina las ventajas de las dos estructuras 1-D expuestas previamente: el acople de los filtros en paralelo permite generar una pendiente pronunciada en las altas frecuencias a pesar de seguir siendo filtros de segundo orden; además se mejora la tolerancia a fallos y se evita la acumulación de retrasos propio de la estructura en cascada. Las implementaciones de este modelo son: en 1992, Watts presenta una implementación con 50 etapas que mejora el rango dinámico, estabilidad y errores derivados de los transistores (Watts 1992), consiguiendo una uniformidad en la respuesta en frecuencia y unos buenos valores de los factores de calidad de los filtros (Q).

(Fragniere & Vittoz 1998) describe un modelo coclear 2-D en el cual la presión y el voltaje son matemáticamente análogos a la aceleración de la membrana basilar y los potenciales de acción. Las implementaciones recientes de este modelo son (van Schaik & Fragnière 2001), (Shiraishi 2003), (Hamilton et al. 2008) y (Wen & Boahen 2009).

Las cócleas artificiales de los últimos dos trabajos mencionados, (Hamilton et al. 2008) y (Wen & Boahen 2009), son activas. Estas implementaciones usan un control automático de ganancia (AGC, Automatic Gain Control). El AGC se utiliza para cambiar la ganancia dependiendo de los cambios de la señal de entrada. En estos casos, sin embargo, el AGC además controla el factor de calidad, Q de cada sección de la cóclea. De esta forma, no solo se cambia la dinámicamente la ganancia como respuesta a los cambios de las características del sonido de estímulo, sino que también se cambia el ancho de banda de cada filtro.

### 3.4.4. Cócleas digitales

La primera cóclea digital fue implementada en 1992 usando Circuito Integrado para Aplicaciones Específicas (ASIC) (Summerfield & Lyon 1992). Contiene 71 secciones de filtros en cascada siguiendo el modelo coclear de Lyon. La salida de cada filtro está conectada a un rectificador half-wave (HWR), que junto al bloque de control automático de ganancia (AGC) simulan la función de los OHCs. Por lo tanto, esta implementación incluye el control activo de la ganancia aunque el modelo en que se basa no lo contempla.

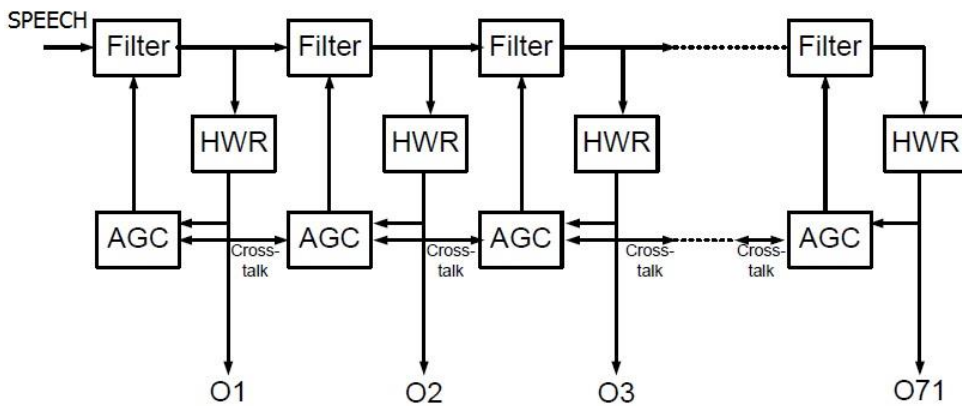


Figura 3.20. Diagrama de bloques de la cóclea digital propuesta por Summerfield et al. Imagen tomada de (Summerfield & Lyon 1992)

La cóclea digital del trabajo (Jones et al. 2000) implementa el banco de filtros paso de banda de segundo orden en una FPGA. El banco de filtros sigue la estructura paralela y fue específicamente construido para extraer el tono de sonidos complejos. Aunque la cóclea digital tiene una arquitectura sencilla, implementa en detalle el modelo de los IHCs y las vías auditivas.

En el trabajo (Leong et al. 2003) se presenta una implementación de 88 secciones en cascada mediante filtros IIR de segundo orden (Infinite Impulse Response). Este tipo de filtros son frecuentemente usados porque consiguen bandas más altas y estrechas con menor número de operaciones aritméticas. En la Figura



3.21 se observa la respuesta frecuencial de algunas de las secciones de la cóclea presentada en dicho trabajo.

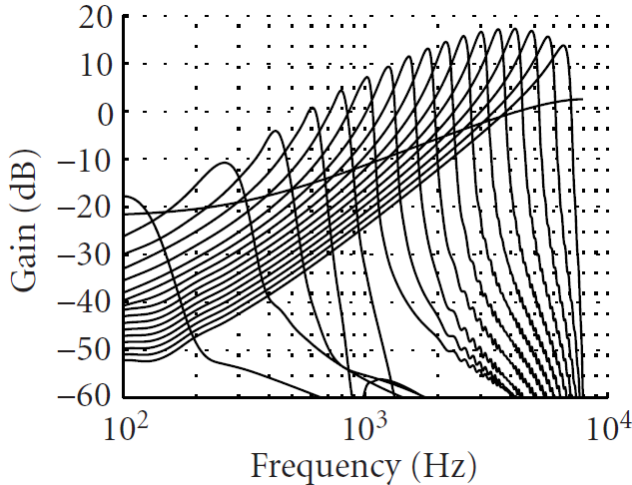


Figura 3.21. Respuesta frecuencial de la implementación coclear de Leong et. al. Imagen tomada de (Leong et al. 2003).

Aunque el procesado paralelo proporciona una salida a una mayor velocidad, existe una limitación en el tamaño de la cóclea puesto que los recursos necesarios crecen linealmente con el número de etapas de la cóclea para obtener la pendiente pronunciada típica de la cóclea biológica. En cambio, en la cóclea en cascada el límite del número de etapas suele venir determinado porque la velocidad disminuye linealmente con el número de etapas.

Las implementaciones más recientes de cócleas están basadas en filtros IIR, algunas usan la estructura en paralelo, como las presentadas en (Dundur et al. 2008) y en (Miró-Amarante 2013), y otras usan conexión de los filtros en cascada como (Gambin et al. 2010), (Mugliette et al. 2011) y (Thakur et al. 2014). El diseño presentado en (Dundur et al. 2008) es la base para la implementación de implante cocleares sobre FPGAs.

Los diseños (Gambin et al. 2010), (Mugliette et al. 2011) y (Thakur et al. 2014) usan la multiplexación en el tiempo para implementar la cadena de filtros con un

mayor número de secciones. Por ejemplo, la cóclea digital presentada en (Mugliette et al. 2011), está basada en un banco de 24 filtros IIR de segundo orden paso bajo en cascada y utiliza un esquema multiplexado en el tiempo, con el objetivo de sólo requerir 20 multiplicadores. En la Figura 3.22 se muestra el esquema de dicha implementación. Estos diseños basados en la multiplexación temporal ahorran recursos a costa de usar frecuencias de reloj muy elevadas, consumiendo una gran cantidad de potencia.

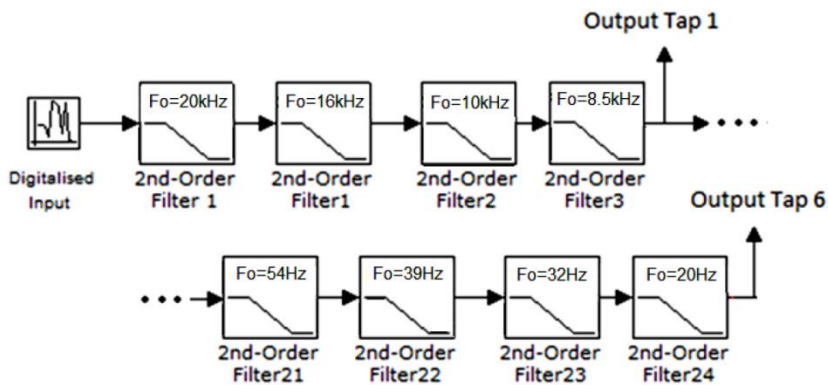


Figura 3.22. Filtros en cascada con salida cada 4 secciones de la cóclea digital presentada en (Mugliette et al. 2011). Imagen tomada de (Mugliette et al. 2011)

El uso de la tecnología FPGA presentan ventajas respecto los sistemas analógicos VLSI: un tiempo más breve de diseño, un tiempo de fabricación más rápido, ser más robusto respecto a cambios en la fuente de alimentación, de temperaturas y de errores en los transistores, un rango dinámico más amplio, un mayor SNR (Signal to noise ratio); mejor estabilidad; las placas se pueden reutilizar para diferentes aplicaciones y tiene una interfaz más simple con un PC.

### 3.4.5. Resumen de las características de las cócleas artificiales previas

En el capítulo 5 de este trabajo se expone la arquitectura y resultados característicos de la implementación de una cóclea artificial pulsante. Además, se presenta una comparativa con las implementaciones previas relevantes de las que

se conocen características medibles. Para realizar dicha comparativa, se resumen en las siguientes tablas, Tabla 3.1 y Tabla 3.2, los datos que se conocen de las cócleas artificiales más recientes.

Tabla 3.1. Resumen de características de cócleas analógicas

Referencia de la cóclea artificial	Número de secciones	Rango de frecuencias	Rango dinámico	Tasa de eventos	Potencia de consumo
(Liu et al. 2010)	64x2	50Hz-50kHz (ajustable)	36dB	10MEvents/s.	18-26 mW
(Wen & Boahen 2009)	360	200Hz-20kHz	52dB	33kSpikes/s.	51,8 mW
(Hamilton et al. 2008)	64x2	200Hz-6,6kHz	46dB	No aporta	56,32 mW

Tabla 3.2. Resumen de características de cócleas digitales

Referencia de la cóclea artificial	Número de secciones	Rango de frecuencias	Reloj del sistema	Recursos hardware usados
(Thakur et al. 2014)	1224	20 Hz- 20,657 kHz	142MHz	113.760 slices registers 136.975 LUTs
(Dundur et al. 2008)	16	150Hz-3,4kHz	No aporta	11.048 slices 20.699 LUTs
(Leong et al. 2003)	88	1,006-7,630 kHz	56,42MHz (mínimo)- 63MHz (máximo)	5.770 slices (mínimo)- 10.771 slices (máximo)

## 4. Monitorización de spikes

*“Poca observación y muchas teorías llevan al error. Mucha observación y pocas teorías llevan a la verdad”, Alexis Carrel*

Como se ha explicado en el segundo capítulo de este trabajo, cuando tenemos un sistema de capas neuronales con cientos o miles de neuronas, resulta prácticamente imposible conectar punto a punto cada neurona implementada en dicho sistema. Una solución a este problema consiste en codificar cada spike según un espacio de direcciones AER (Boahen 1998), para ser transmitido a través del bus AER, bus digital, asíncrono y multiplexado en el tiempo. En este capítulo vamos a exponer y analizar dos componentes VHDL<sup>17</sup> cuya función es monitorizar la actividad interna de spikes lanzados por sistemas neuronales implementados en FPGAs. Estos circuitos se encargan de dar un código único (dirección) a cada neurona y cuando una neurona dispara un spike, el circuito debe tomar nota de ello, gestionar las posibles colisiones, neuronas que disparan un spike simultáneamente, y finalmente, codificar el evento con su dirección pre-asignada. Este evento será enviado a través del bus AER, el cual tiene las líneas de control de petición (REQ) y de respuesta (ACK) que implementa el protocolo hand-shake asíncrono de cuatro fases, establecido en el proyecto CAVIAR (Häfliger 2007), (Rafael Serrano-Gotarredona et al. 2009), explicado en detalle en el Capítulo 2. Las neuronas receptoras estarán escuchando el bus, buscando spikes que se han enviado para ellas, de esta forma se consigue conectar virtualmente neuronas mediante un flujo de eventos AER.

La principal dificultad del diseño de este tipo de circuitos está en el desarrollo de estrategias para minimizar posibles pérdidas de spikes derivadas de las

---

<sup>17</sup> VHSIC Hardware Description Language

colisiones temporales. Las colisiones temporales son aquellas situaciones donde dos o más spikes han sido disparados en el mismo instante, y tienen que ser enviados de forma multiplexada por un bus AER común. En un escenario ideal, donde los spikes son disparados de neurona a neurona de manera secuencial, sin colisiones temporales, la codificación de spikes como eventos AER sería automática usando un decodificador digital tradicional. Sin embargo, como las colisiones temporales son muy comunes, dos o más eventos AER deberían ser enviados simultáneamente por el bus AER, que es único y multiplexado en el tiempo. En consecuencia, los spikes que se han disparado en paralelo, deben ser transmitidos como eventos AER secuencialmente.

La aplicación típica de los monitores de spikes se muestra en la Figura 4.1, donde se encuentran circuitos que lanzan spikes (por ejemplo un conjunto de neuronas pulsantes) conectados al monitor de spikes, el cual codifica los spikes en direcciones AER y los envía por el bus AER a otra capa de procesamiento neuronal.

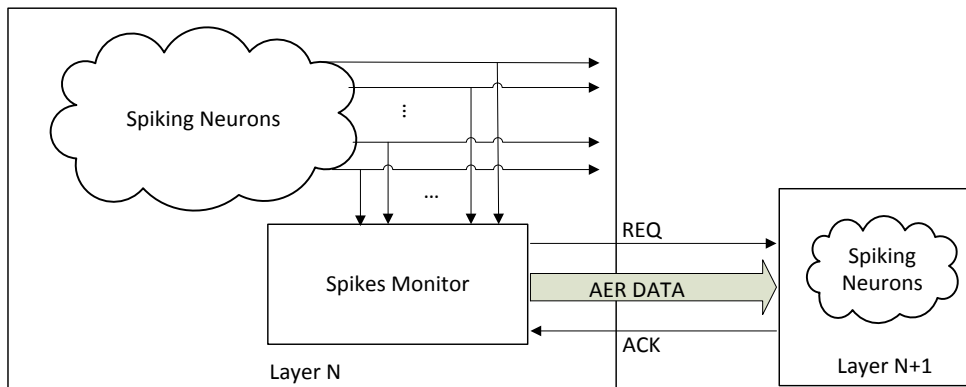


Figura 4.1. Escenario típico de uso de un monitor de spikes

En este capítulo se van a presentar y analizar dos componentes implementados en VHDL para la monitorización de la actividad interna de neuronas implementadas en FPGA, los cuales codifican cada spike acorde a la representación AER para mandarlos por el bus AER como eventos discretos,

usando diferentes estrategias para manejar las colisiones temporales. Ambos componentes están desarrollados en VHDL y son genéricos, es decir, escalables al número de neuronas que se necesite para una implementación concreta. Para analizar su comportamiento, hemos diseñado un escenario experimental donde diversos sistemas se han usado para estimular los monitores y se han recolectado los eventos AER de salida para su posterior análisis. En dicho análisis hemos medido diferentes características como es la máxima tasa de spikes y la proporción de eventos perdidos debido a las colisiones. Se presentarán los resultados de tasa de eventos máxima soportada y las pérdidas por colisiones para ambos monitores ante situaciones experimentales similares.

### 4.1. Monitor de spikes masivo

El monitor de spikes masivo, o Massive Spikes Monitor (MSM), fue desarrollado como una primera propuesta en el contexto del trabajo (Jimenez-Fernandez et al. 2009), con el objetivo de monitorizar la actividad de spikes en un sistema controlador de robots mediante spikes. A continuación presentamos un resumen de su arquitectura y comportamiento para poder compararlo con el monitor AER distribuido, desarrollado en el presente trabajo y que usaremos en el sistema de reconocimiento de audio presentado en el capítulo 5.

La arquitectura y comportamiento del MSM se muestran en la Figura 4.2. Este monitor se puede dividir en tres bloques: el bloque que toma una captura del estado de todas las señales que comunican los spikes en el caso de que alguno de ellos se haya disparado, el bloque que codifica los spikes disparados con su correspondiente dirección AER y el bloque que se encarga de enviar las direcciones AER mediante el protocolo de comunicación hand-shake asíncrono de cuatro fases que fue establecido en el proyecto CAVIAR (Häfliger 2007), (Rafael Serrano-Gotarredona et al. 2009). El primer bloque usa una memoria FIFO, denominada FIFO de spikes, con el objetivo de almacenar “fotografías” de los spikes disparados en un mismo instante de tiempo, de manera que los spikes son alineados como palabras de un tamaño de 2 bits por el número de spikes a monitorizar (hemos de tener en cuenta tanto los spikes positivos como los negativos), además se hace una operación OR entre ellos, cuya salida está

conectada a la señal de escritura (WR) de la FIFO de spikes. De tal manera, que en caso de que se dispare un solo spike, tomamos una fotografía de todos los spikes que se han disparado de manera simultánea. El segundo bloque tiene una máquina de estado finitos, llamada FSM Spikes2AER (situada debajo de la FIFO de los spikes en la Figura 4.2, y cuyo diagrama de estados se muestra en la Figura 4.2 a la izquierda), encargada de tomar una palabra de la FIFO de spikes, cargarla en un registro y recorrerlo bit a bit de forma secuencial, de tal manera que cada vez que se encuentre un '1' lógico en una posición determinada, significará que se ha disparado un spike de una determinada neurona, debiendo obtener la dirección AER asociada a la posición de ese spike de la memoria ROM de mapeo, y almacenarla en la FIFO de eventos AER. La FIFO de eventos AER es en la que se van almacenando los eventos AER que van a ser transmitidos a través del bus AER. Finalmente, el tercer bloque tiene una máquina de estados (a la derecha de la Figura 4.2) entre la FIFO de eventos AER y el puerto AER paralelo de salida; esta FSM se encarga de ir tomando los eventos de la FIFO AER, e ir transmitiéndolos usando el protocolo AER paralelo de hand-shake asíncrono. En resumen, los spikes que se disparan en paralelo son almacenados en la FIFO de los spikes, después son procesados secuencialmente por la FSM Spikes2AER, almacenando de manera secuencial su dirección AER en la FIFO AER, y finalmente comunicando los eventos AER uno a uno a través del puerto AER paralelo siguiendo el protocolo de comunicación AER asíncrono.

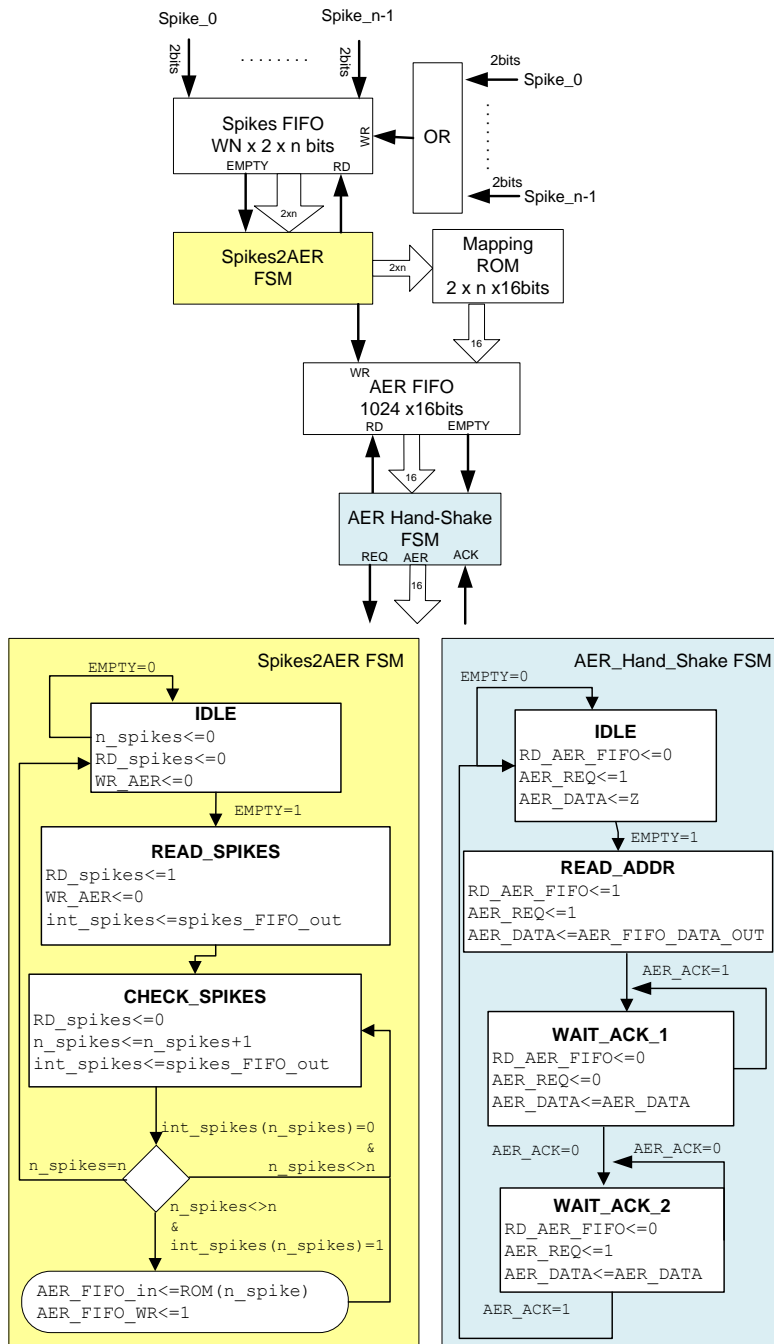


Figura 4.2. Arquitectura y funcionamiento del monitor masivo de spikes



Este monitor asigna direcciones AER a los spikes activos de las líneas de entrada según la Tabla 4.1. Por ejemplo, si está activo el spike de la posición 0 se le asigna la dirección AER 0, si está activo el spike de la posición 1 se le asigna la dirección AER 1 y así consecutivamente. Por lo tanto, el número de bits necesarios para codificar un spike de entrada en su correspondiente evento AER cumple la Ecuación 4.1, siendo  $n_{IN}$  el número de líneas de entrada del monitor y  $n_{AER}$  el número de bits necesarios para codificar el spike en su correspondiente evento AER.

$$n_{AER} \geq \log_2(n_{IN})$$

Ecuación 4.1

Tabla 4.1. Ejemplo de espacio de direcciones de los monitores

Spike activo	Dirección AER
00000...0001	0
00000...0010	1
00000...0100	2
00000...1000	3
...	...

El MSM presenta dos problemas: desaprovecha una cantidad considerable de memoria y demanda un elevado tiempo de procesamiento. El primer problema consiste en que se desperdicia memoria cuando se dispara un único spikes porque se hace una “fotografía” que ocupa todo el ancho de una palabra de spikes, por lo tanto se está desperdiciando tamaño de la FIFO de spikes de entrada. Para sistemas en los que se implementan pocas neuronas, este efecto es poco relevante, pero en sistemas con un elevado número de neuronas, tanto la anchura como profundidad de la FIFO de spikes necesaria consume una ingente cantidad de memoria. En conclusión, se necesita tener una FIFO de spikes grande con un ancho acorde con el tamaño de spikes de entrada que se va a desaprovechar porque en los sistemas de este tipo no suelen activarse muchos spikes de manera simultánea. El segundo problema consiste en que se desperdicia tiempo cuando se procesa un único spike

que se ha disparado sólo en ese instante de tiempo, porque el monitor tiene que recorrer el registro entero en busca del spike activo. Por ejemplo, si tenemos un MSM de 256 líneas de entrada, si se dispara un solo spike tenemos que despreciar 255 ciclos de reloj por un solo spike. Esto tiene dos consecuencias relacionadas entre sí, por un lado se introduce una latencia alta en la obtención de su correspondiente dirección y se ocupa una posición de memoria de la FIFO de spikes durante esa latencia.

En la Ecuación 4.2 se muestra la relación entre el tiempo que tarda el MSM en recorrer el registro donde se procesan los spikes para buscar los spikes activos y el número de líneas de entrada, donde  $n_{IN}$  es el número de líneas de entrada que tenga el monitor y  $CLK$  es la frecuencia de reloj.

$$t_{procesado} = n_{IN} * 1/CLK$$

Ecuación 4.2

## 4.2. Monitor de spikes distribuido

El propósito del monitor distribuido, o Distributed Spikes Monitor (DSM), es evitar los problemas del monitor masivo, distribuyendo el procesamiento de los spikes de entrada en cuatro circuitos idénticos que funcionan paralelamente. En la Figura 4.3 se observan estos cuatro circuitos con el nombre del *MODULE*. Cada módulo almacena un cuarto de los spikes en un registro y se procesan en la máquina de estados *SPIKES SCAN*. La función de dicha máquina de estados es recorrer el registro bit a bit, y en caso de que un spike se haya disparado se obtiene su correspondiente dirección AER parcial mediante la posición que ocupa en el registro. Esta dirección AER parcial, se almacena en una memoria FIFO, denominada en la Figura 4.3 *PARTIAL AER FIFO* cuyo ancho depende del ancho de los spikes de entrada según la relación de la Ecuación 4.3, dónde  $n_{AER\_parcial}$  es el ancho de la memoria FIFO para almacenar las direcciones AER parciales y  $n_{IN}$  es el número total de líneas de entrada del monitor.

$$n_{AER\_parcial} \geq \log_2(n_{IN}/4)$$

Ecuación 4.3

A continuación es necesario codificar cada dirección AER parcial con su dirección completa. De eso se encarga la máquina de estados *MERGE AER*, que le llegan las direcciones AER parciales y el estado de las memorias de tipo FIFO y con esa información obtiene la dirección AER completa. A partir del momento en que se obtiene la dirección AER completa, el procesamiento es similar al procesamiento del MSM, las direcciones AER se almacenan introduciéndola en la FIFO AER y finalmente, la máquina de estados AER Hand-Shake se encarga de ir transmitiendo los eventos AER de la FIFO AER a través del puerto AER mediante el protocolo de *hand-shake* asíncrono de 4 fases antes mencionado (Häfliger 2007). Como en el MSM, el DSM es adaptable a diferentes anchos de spikes de entrada y completamente configurable.

El DSM recorre los registros que contiene los spikes de entrada activos secuencialmente, pero el tiempo dedicado a recorrer estos registros en una cuarta parte que el tiempo que dedica el MSM porque puede realizar este recorrido en paralelo en los cuatro módulos, además, sólo tiene que recorrer los registros en cuya porción existan spikes activos. Es decir, si sólo se ha activado un spike, solamente el módulo que recibe dicha línea de entrada recorrerá el registro. En la Ecuación 4.4 se muestra la relación entre el tiempo que tarda el DSM en recorrer los registros donde se procesan los spikes para buscar los spikes activos y el número de líneas de entrada, donde  $n_{IN}$  es el número de líneas de entrada que tenga el monitor y  $CLK$  es la frecuencia de reloj.

$$t_{procesado} = n_{IN}/4 * 1/CLK$$

Ecuación 4.4

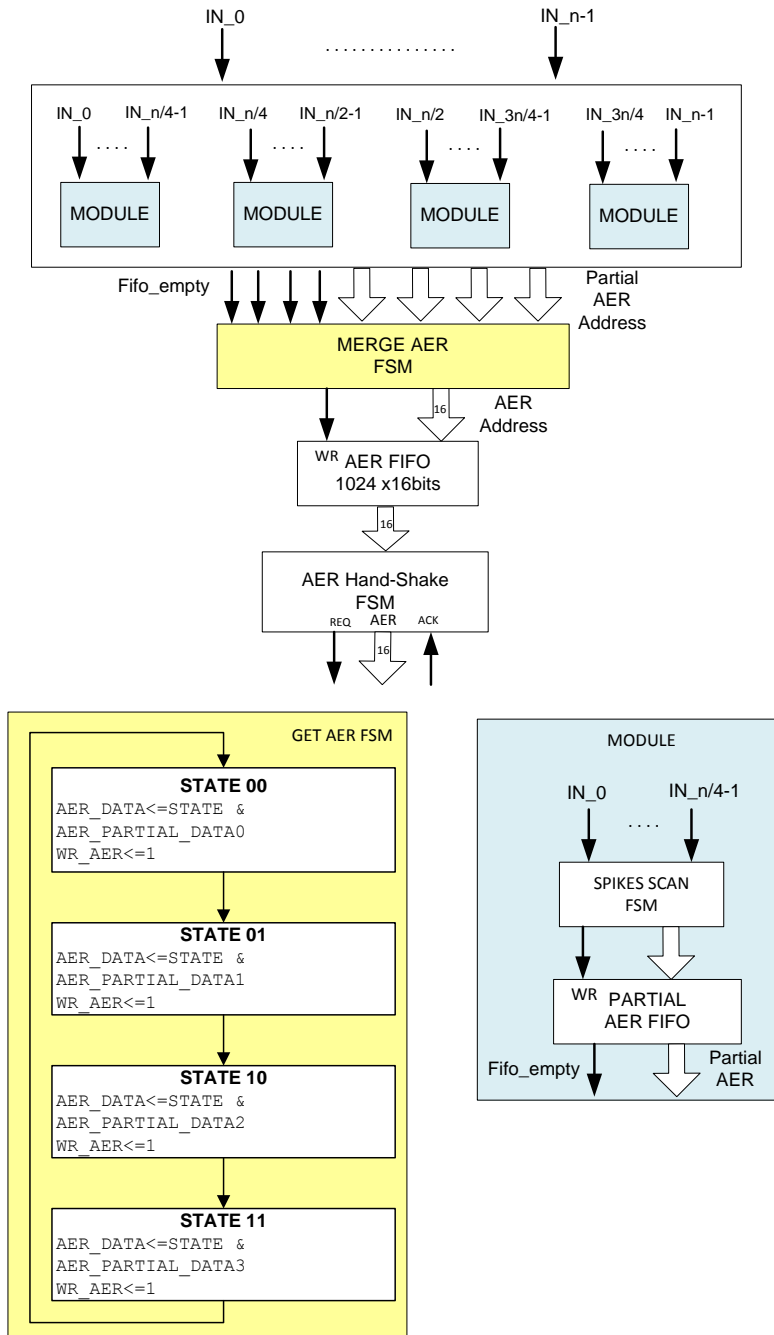


Figura 4.3. Arquitectura y funcionamiento del monitor de spikes distribuido

A continuación vamos a realizar una serie de experimentos para comprobar el funcionamiento de ambos monitores y calcular la tasa de eventos y el porcentaje de pérdidas debido a colisiones.

### 4.3. Escenario experimental

Como se comentó previamente, ambos monitores son escalables al número de neuronas que se necesite para una implementación concreta, es decir, al número de señales de entrada. Para realizar la comparativa de ambos monitores se ha fijado este número a 32 líneas de entrada, por lo tanto, los monitores tienen que codificar 32 direcciones AER distintas.

A continuación presentamos un análisis de la respuesta funcional de ambos monitores mediante medidas de rendimiento. Con la meta de obtener dichos resultados, hemos construido un escenario experimental, donde diversos sistemas AER han sido usados para estimular los monitores y recoger los eventos AER de salida para su posterior análisis. El diseño de los experimentos ha sido modificado a lo largo del proceso de desarrollo de ambos, pero finalmente las fases de ejecución de un experimento se pueden resumir como:

1. **Generación de spikes aleatorios:** en el PC, mediante un script de Matlab se generan spikes aleatorios, pudiendo configurar el tamaño de palabra de spikes, el cual debe ser igual al número de líneas que tiene el monitor de entrada, el porcentaje de carga de spikes y la probabilidad de que se active un spike.
2. **Almacenamiento de dichos spikes en la memoria RAM de la placa USB-AER:** mediante la conexión USB provista por un microcontrolador que contiene la placa y un componente VHDL que gestiona la memoria RAM sintetizado en una FPGA de la familia Spartan II, se escriben los spikes en la memoria RAM que contiene la USB-AER (Gomez-Rodriguez et al. 2006).
3. **Envío de spikes al monitor:** mediante un componente VHDL sintetizado en la FPGA de la placa USB-AER se leen los spikes de memoria y se envían al monitor.

4. **Monitorización de spikes:** el monitor procesa los spikes de entrada, genera sus correspondientes direcciones AER y manda dichas direcciones por el puerto AER de salida a la placa USB-AERmini2 (Berner et al. 2007).
5. **Colección y procesamiento de la salida:** la placa USB-AERmini2 envía los eventos AER al PC para su posterior análisis.

En la Figura 4.4 se pueden observar cada una de las fases de los experimentos, numeradas del 1 al 5, con los dispositivos que intervienen en cada fase.

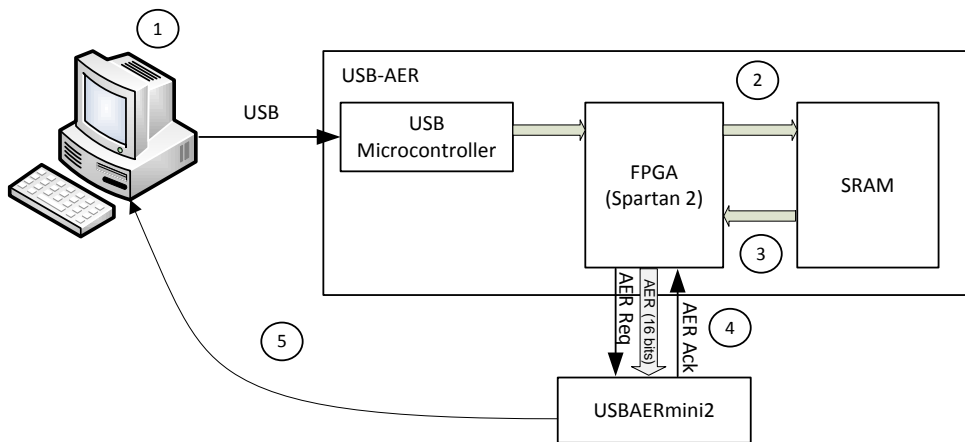


Figura 4.4. Fases de los experimentos

A continuación se van a exponer con más detalle cada una de estas fases, mediante la descripción de cada uno de los componentes que intervienen en el escenario experimental. Se van a introducir las placas que hemos usado en este trabajo. Se tiene disponible la descripción completa de estas placas en el Anexo de este trabajo y en (Gomez-Rodriguez et al. 2006), (Berner et al. 2007).

#### 4.3.1. Placas usadas en el experimento: USB-AER y USB-AERmini2

Ambos monitores van a ser programado en la placa USB-AER (Gomez-Rodriguez et al. 2006). Esta placa se basa en una FPGA de Xilinx de la familia Spartan II en la que vamos a programar los monitores AER junto con los módulos que gestionan los spikes. Esta FPGA se puede configurar mediante los siguientes

componentes de los que consta la USB-AER: el microcontrolador SiLabs C8051F320 y un puerto USB 1.1 Full Speed. Además, por medio del puerto USB y el microcontrolador se puede intercambiar datos entre el PC y la FPGA mediante un bus asíncrono compuesto por 8 líneas de datos, además de algunas señales adicionales de control que interconectan la FPGA con el microcontrolador. Este mecanismo no lo podemos usar para excitar en tiempo real los monitores de spikes porque tiene un ancho de banda máximo de 6Mbits/s ( $\sim 182\text{KEvents/s}$ ) y necesitamos una tasa máxima de eventos de 5MEventos/s para comprobar las prestaciones de nuestros sistemas. Por lo que vamos a usar esta comunicación para enviar los spikes a la memoria RAM estática que tiene esta placa para una posterior excitación de los monitores. Esta memoria es una memoria RAM estática de 2MB estructurada en palabras de 32bits y con un tiempo de acceso de 12ns. Posteriormente se accederá a esta memoria para tomar los datos que van a excitar los monitores. Cada acceso representa 32 líneas de spikes que pueden proporcionar de 0 a 16 spikes simultáneos, emulando una colisión temporal. Esta placa tiene dos puertos paralelos AER que cumplen el estándar CAVIAR (Serrano-Gotarredona et al., 2009), de los cuales se usa el puerto de salida para realizar la comunicación AER con la placa USB-AERmini2 (Berner et al. 2007). En la Figura 4.5, izquierda, se muestra una fotografía de esta placa, destacando en rojo los componentes que acabamos de exponer.

La placa USB-AERmini2 tiene 3 puertos AER paralelos, de los cuales vamos a usar uno para recibir los eventos que provienen del monitor programado en la FPGA de la placa USB-AER tras su excitación. Esta tarjeta es un puente completo entre el bus AER y el bus USB de un PC (Berner et al., 2007). Este dispositivo permite secuenciar (reproducir) y monitorizar (capturar) tráfico AER con una resolución temporal tanto de 1 $\mu$ Sec como 0.2 $\mu$ Sec. Puede ser tanto insertada como monitor entre dos dispositivos AER (modo pass-through), o puede ser usada en modo terminal (como hacemos en este trabajo). Este dispositivo consta de un microcontrolador, fabricado por Cypress y de la familia FX2LP y una CPLD Xilinx CoolRunner 2 que tiene 256 macroceldas. En la CPLD hay implementadas cuatro FSM, para manejar los puertos AER (tanto de envío como de recepción), generar marcas de tiempo (timestamps) y enviar los datos por el puerto USB2.0

High-Speed (velocidad máxima de 480Mbps) del que consta. La CPLD tiene conectado un reloj de 30MHz, obteniendo una tasa de monitorización de 6 mega-eventos AER por segundo de pico, pudiendo mantenerse a una frecuencia constante de 4.5 Meventos/s, aunque estas tasas están limitadas por la capacidad del PC. Esta placa envía a través del USB tanto la dirección AER que codifica al spikes disparado como el instante de tiempo en el que ha sido disparado codificado como un número de 32 bits. En la Figura 4.5 se observa esta placa y la interconexión con la placa USB\_AER.

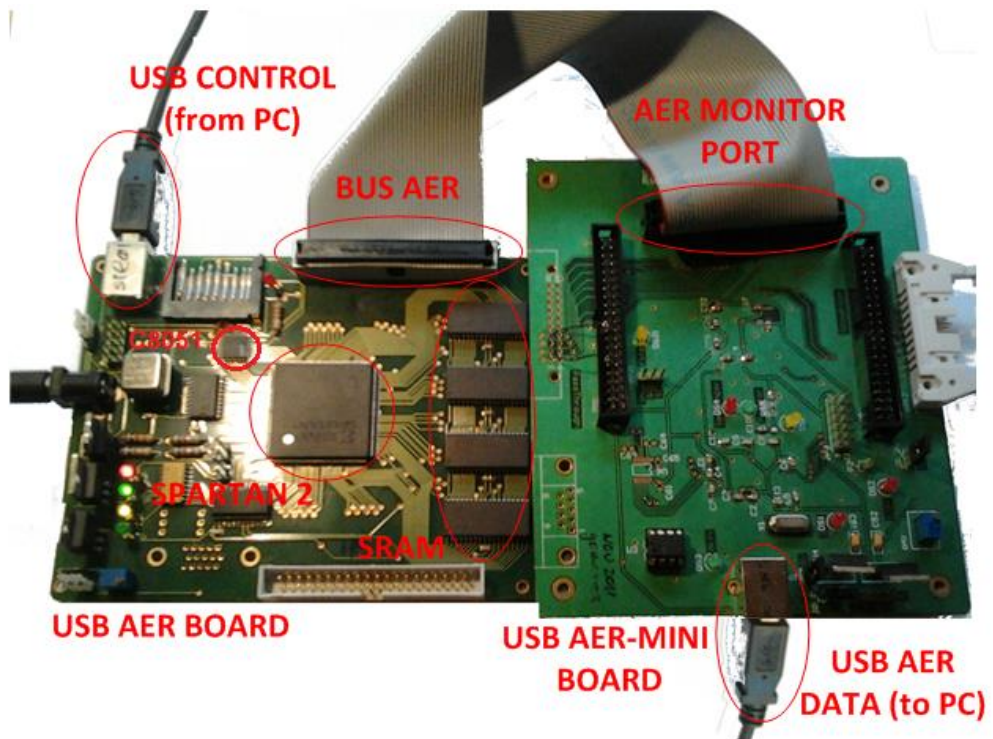


Figura 4.5. Fotografía del escenario experimental

#### 4.3.2. Excitación de los monitores y captura de los eventos generados

En este apartado se expone el componente VHDL que se ha diseñado y con el que se programa la FPGA Spartan II de la USB-AER. Está compuesto por tres módulos:



- Un módulo que recibe los bytes del microcontrolador con la información necesaria para excitar el monitor (Figura 4.6-1). Este módulo se encarga de almacenar dicha información organizada en palabras de 4Bytes en la memoria SRAM de la placa (Figura 4.6-2). Una vez que están todos los datos escritos en la SRAM ya están listos para que el siguiente módulo realice su función.
- El siguiente módulo se encarga de leer de la memoria palabras de 32 bits (Figura 4.6-3), para usar los bits de dichas palabras como los valores de las líneas de entrada del monitor (Figura 4.6-4) Los valores de los bits que estén a 1 se consideran spikes activos de entrada del monitor.
- Por último, el módulo que contiene el MSM o DSM, dependiendo de cuál de los dos se quiere evaluar con un experimento concreto.

En la Figura 4.6 se observan cada uno de los componentes de la placa USB\_AER y las conexiones que existen entre ellos en el escenario de este trabajo.

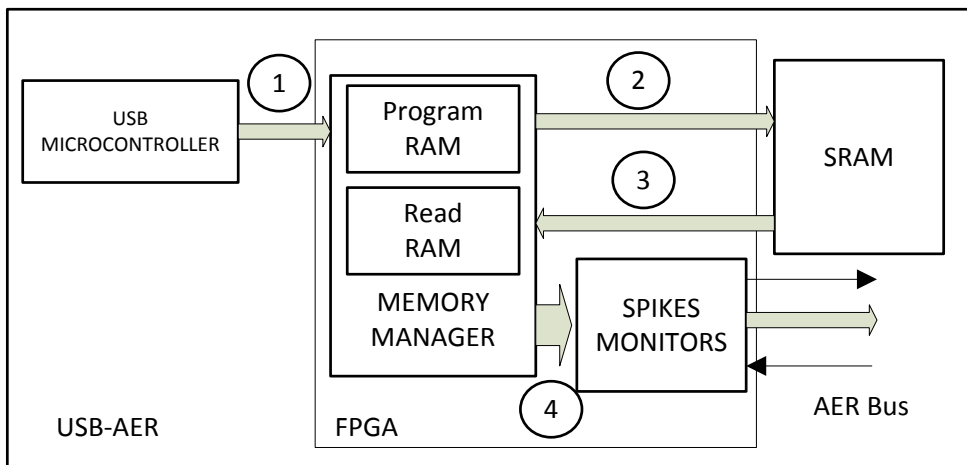


Figura 4.6. Componentes de la placa USB-AER para la evaluación de los monitores de spikes

### 4.3.3. Generación de los spikes y procesamiento de la salida

Hemos desarrollado dos algoritmos en Matlab: uno para generar la batería de pruebas y otro para procesar la información obtenida del monitor testado.

La generación de los estímulos la hemos parametrizado acorde con la probabilidad de activación de un spike en un instante de tiempo determinado y el número de spikes activos simultáneamente. El tamaño del monitor que queremos testear lo hemos marcado a 32 líneas de entrada. Por lo tanto, se generan palabras de 32bits que se van a almacenar en 4bytes consecutivos de la SRAM, donde si un bit está a 1 significa que se ha disparado ese spike. La posición de los spikes activos en la palabra de 32 bits se hace de forma aleatoria. Se generan tantas palabras de spikes como caben en la memoria SRAM de la placa USB-AER, hasta completar los 2MB de la SRAM.

Por otro lado, tenemos un script que se encarga de leer la salida del monitor mediante el USB conectado a la placa USB\_AERmini2 y una serie de comandos de control de alto nivel (Berner et al. 2007). Una vez almacenados los eventos AER de salida del monitor, se calculan la tasa de eventos de salida y el número de pérdidas debidas a colisiones con el objetivo de comparar el rendimiento entre ambos monitores.

#### 4.4. Resultados experimentales

Para caracterizar el comportamiento de los monitores hemos realizado una batería de experimentos, realizando un barrido de distintos valores relativos a la probabilidad de activación de los spikes y el número de spikes que se disparan simultáneamente. Los valores de la probabilidad de activación de spikes los hemos fijado en el rango [0.1, 1] y el número de spikes activos simultáneamente máximo posible es 16, porque los spikes positivos y negativos no se pueden disparar a la vez. La primera medida que hemos calculado es la tasa media de eventos AER de salida del monitor respecto a la tasa de spikes de entrada. Para calcular la tasa de spikes de entrada según los dos parámetros variables, hemos usado la Ecuación 4.5. La tasa de eventos generado varía entre 1MSpikes/s a 15MSpikes/s

$$Avg. Spikes Rate = \frac{Number\ of\ Active\ Spikes * Probability}{Total\ Stimulus\ Time} (Events/Sec)$$

Ecuación 4.5.

En la Figura 4.7 se observa la tasa de eventos de salida (eje  $z$ ), respecto al barrido de la tasa de spikes de entrada (eje  $x$ ) y al número de spikes disparados simultáneamente (eje  $y$ ). Debido a la arquitectura del monitor masivo, muchos spikes se pierden, por ejemplo, cuando se excita al monitor masivo con 5MSpikes/s, la tasa de eventos de salida no alcanza a 3MEvents/s como se observa en la Figura 4.7. La máxima capacidad que tiene la placa USB-AERmini2 de monitorizar eventos es de 5MEvents/s sin embargo este monitor no alcanza esa tasa máxima, hasta 8.5MSpikes/s de entrada, a partir de la cual ya satura la placa USB-AERmini2. En cambio, el monitor distribuido tiene mejor comportamiento, generando una tasa de eventos de salida similar a la tasa de spikes de entrada, saturando el bus AER a casi los 5MEvents/s de máximo. En ambos casos se observa que el número de spikes simultáneos disparados no afecta significativamente la resolución de tasa de eventos de ambos monitores. Esto se debe a que el tamaño de la FIFO que almacena las direcciones AER es suficientemente grande como para mitigar este efecto, esta FIFO puede almacenar 1024 eventos AER. Con los mismos resultados del experimento anterior, es decir, la tasa de eventos obtenida para un barrido de tasa de spikes de entrada y el número de spikes disparados simultáneamente, hemos calculado el porcentaje de spikes perdidos. La Figura 4.8 muestra el porcentaje de spikes perdidos respecto a la tasa de eventos de entrada (eje  $x$ ) y el número de spikes disparados simultáneamente de ambos monitores (eje  $y$ ), arriba el monitor masivo y abajo el distribuido. El monitor masivo tiene unas pérdidas despreciables por debajo de 3MEvents/s. Sin embargo, por encima de esta medida y conforme la tasa de eventos aumenta, la FIFO de spikes se satura, perdiéndose muchos eventos. En cambio, el monitor distribuido tiene una tasa pequeña de pérdida de spikes gracias a las memorias FIFOS distribuidas, y a la máquina de estados *MERGE AER* ( Figura 4.3) que realiza el procesamiento de los spikes de entrada de forma distribuida. Este monitor empieza a perder una cantidad considerable de spikes cuando se excita por encima de 5MEvents/s, la cual coincide con la mayor tasa de spikes que podemos medir, ya que es la máxima tasa de comunicación del bus AER de la placa USB-AERmini2.

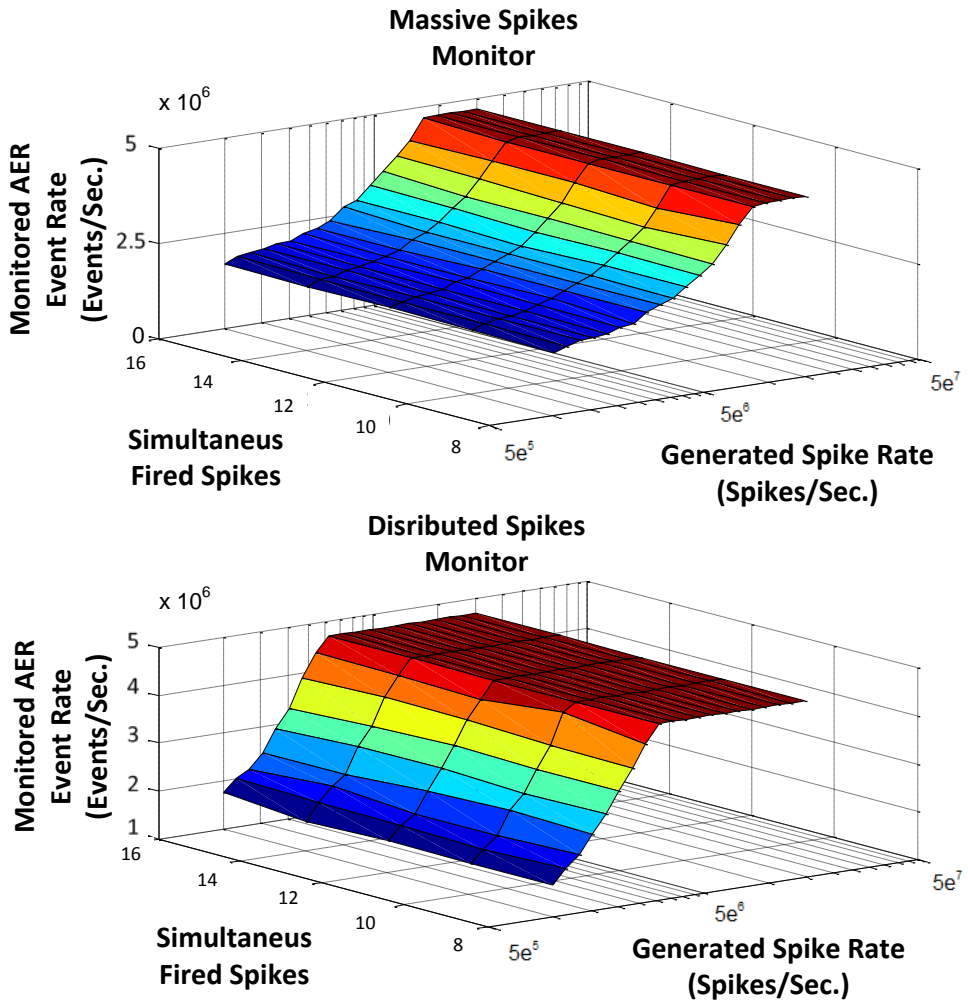


Figura 4.7. Tasa de eventos de salida del monitor masivo (arriba) y del monitor distribuido (abajo)

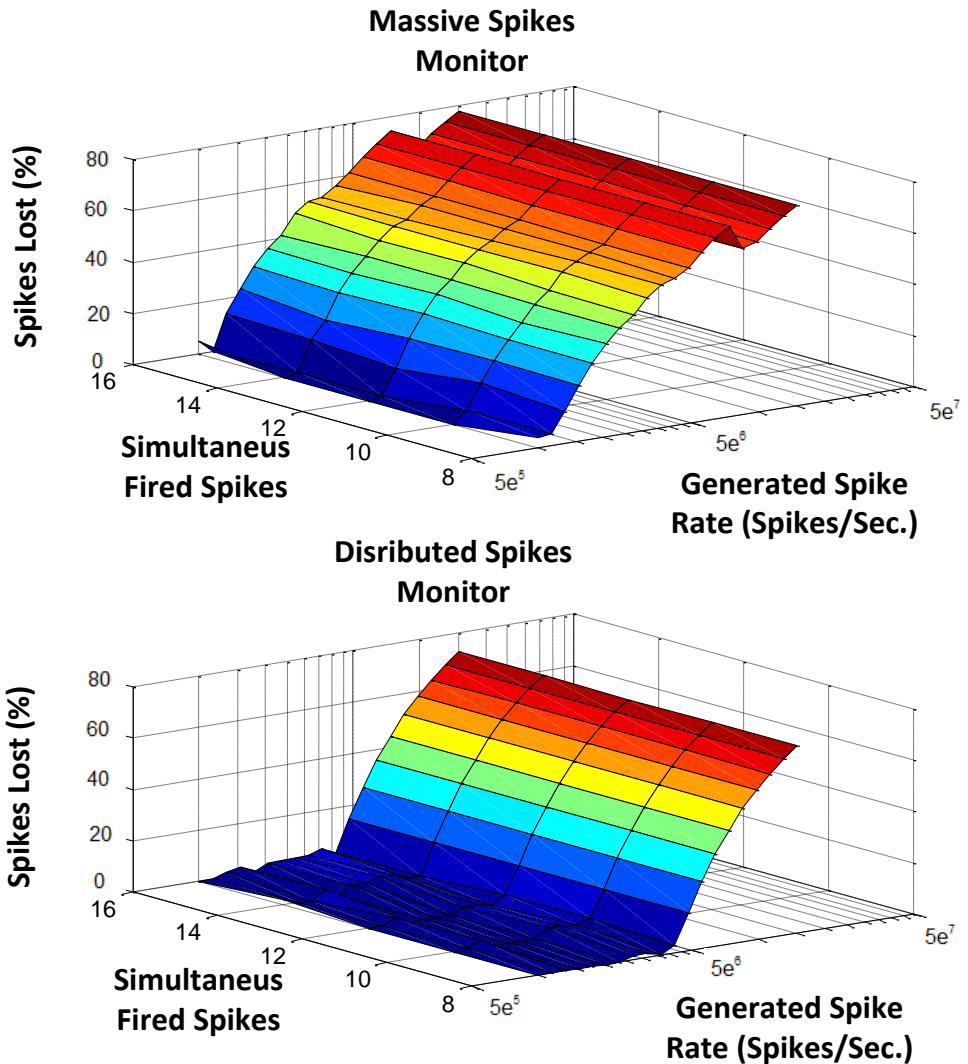


Figura 4.8. Porcentaje de spikes perdidos para el monitor masivo (arriba) y distribuido (abajo)

Después de este experimento, destacamos que el monitor distribuido tiene un mejor comportamiento que el monitor masivo, siendo adecuado para sistemas que necesiten un gran ancho de banda, es decir, mayor de 2.5MEvents/s (que es el máximo que permite el monitor masivo). De esta forma, este circuito deja de ser el

cuello de botella de la comunicación en nuestros sistemas y pasa a ser el cuello de botella el ancho de banda del bus que usemos para la comunicación. Por lo tanto, bajo estas condiciones de operación el DSM tiene una capacidad de comunicación de eventos AER superior al ancho de banda, en términos de eventos por segundo, de las herramientas de monitorización de eventos AER disponibles. Gracias a la reciente implantación de los estándares USB3.0 Super Speed, podremos realizar nuevas pruebas de rendimiento para poder comprobar qué tasas máximas reales alcanza el DSM.

#### 4.5. Consumo hardware

Como se ha expuesto previamente, los monitores se han diseñado de forma genérica, dejando indeterminado el número de líneas de spikes de entrada y las profundidades de las distintas FIFOs. A continuación vamos a hacer una comparativa del consumo de recursos hardware de ambos monitores en relación con el número de líneas de spikes de entrada.

Ambos monitores tienen las mismas señales de entrada/salida: la señal de reloj, la señal de reset, el bus con los spikes de entrada, el bus AER de salida que consta de 16 líneas para la dirección del evento AER y otras dos señales de control (REQ y ACK). Por lo tanto, el número total de señales va a depender del tamaño del bus acorde con el número de señales de spikes de entrada, representado por la variable  $N$  en la Ecuación 4.6.

$$I/O\ signals = CLK + RST + N + 18(bus\ AER)$$

Ecuación 4.6.

En la Tabla 4.2 se muestran el número de slices que hacen falta para cada monitor dependiendo del número de líneas de spikes. Los resultados son muy interesantes, para un bajo número de líneas de entrada, el consumo hardware de monitor masivo es significativamente menor que el número de slices que necesita el monitor distribuido, para tamaños mayores se observa como el monitor distribuido necesita menos slices que el masivo. Esto es debido a la memoria inicial que almacena la “fotografía” de todos los spikes en un instante de tiempo (se

observa en la Figura 4.2 como *Spikes FIFO*) aumenta el ancho de todas las posiciones de la memoria. En cambio, en el monitor distribuido se almacenan en un registro en cada módulo que realiza el procesamiento distribuido (se observa en la Figura 4.3 como *MODULE*).

Tabla 4.2. Consumo hardware en slices del monitor masivo y distribuido

Nº de líneas de spikes de entrada	Slices usados por el monitor masivo	Slices usados por el monitor distribuido
32	110	623
64	174	721
128	296	849
256	536	1003
512	1022	1541
1024	3346	2424
2048	6583	4179

En la Figura 4.9 se muestra la gráfica de los slices usados de ambos monitores junto a las rectas de regresión que se aproximan a ambas secuencia de puntos. Se observa que la pendiente de la recta del monitor masivo es casi el doble de la pendiente de la recta de regresión de los slices para el monitor distribuido. Aunque hay que tener precaución si se va a usar la recta de regresión del monitor masivo para estimar el número de slices necesarios para tamaños mayores de spikes de entrada porque el error es elevado (Norm of residuals = 572.65). En cambio la recta de regresión del monitor distribuido se adapta mejor a la secuencia de datos (Norm of residuals = 79.962).

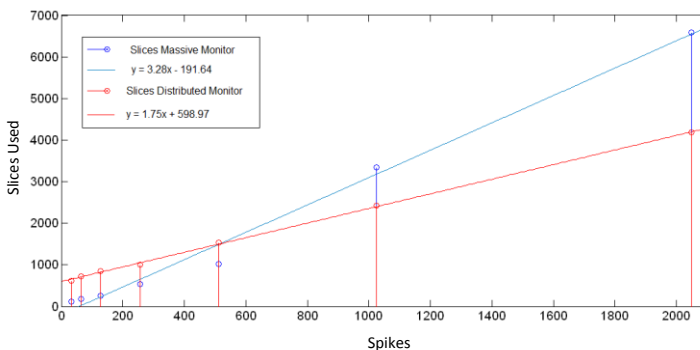


Figura 4.9. Slices usados por cada monitor y recta de regresión para cada monitor







## 5. Sistema Neuromórfico de Audición

*“Nada es tan útil como escuchar mucho”, Juan Luis Vives*

La intensa actividad referida al diseño de sensores y sistemas procesamiento de sonido de manera neuro-inspirada que se está llevando a cabo en centros de investigaciones nos empujó a implementar nuestro sistema de procesamiento de audio neuro-inspirado, de forma que podamos evitar los inconvenientes que presentan las cócleas sintéticas aVLSI y aportar una innovación a este tipo de sensores al usar componentes pulsantes para la implementación del sistema. Como se ha expuesto en el capítulo 3, las cócleas sintéticas previamente desarrolladas realizan un procesamiento del audio tanto analógico como digital, para después convertir la información de cada canal de la cóclea a spikes o eventos AER, en cambio, nuestro sistema filtra directamente el sonido codificado en spikes. De esta manera conseguiremos combinar las ventajas de las cócleas analógicas, ofreciendo una alta escalabilidad y alta velocidad de procesamiento, y las cócleas digitales, ofreciendo una alta inmunidad al ruido y evitando el fenómeno miss-match, además de usar lógica programable comercial (FPGA).

En la cóclea biológica, la onda acústica es filtrada mecánicamente y produce pulsos que representan en el nervio auditivo la descomposición espectral de la onda de entrada. En la Figura 5.1 se muestra de qué forma procesan y transforman la información las diferentes tecnologías usadas para implementar cócleas sintéticas, todas ellas de forma análoga al procesamiento de la información de la cóclea biológica. Las cócleas analógicas transforman la onda sonora en una señal analógica para procesarla por un banco de filtros analógicos. La mayoría de ellas, convierten la salida de los filtros en representación pulsante para representar la

información tal como se hace en el nervio auditivo. En las cócleas digitales, la onda acústica se transforma en una señal digital mediante un codec de audio<sup>18</sup> y dicha señal digital es procesada por filtros digitales. La mayoría de las cócleas digitales tienen mecanismos para convertir la salida de los filtros en información pulsante. El sistema que exponemos en este trabajo procesa la onda sonora de forma novedosa: transforma la señal sonora en representación pulsante y dicha información pulsante es procesada por un banco de filtros pulsantes. La salida de los filtros es pulsante tal y como ocurre en la cóclea biológica.

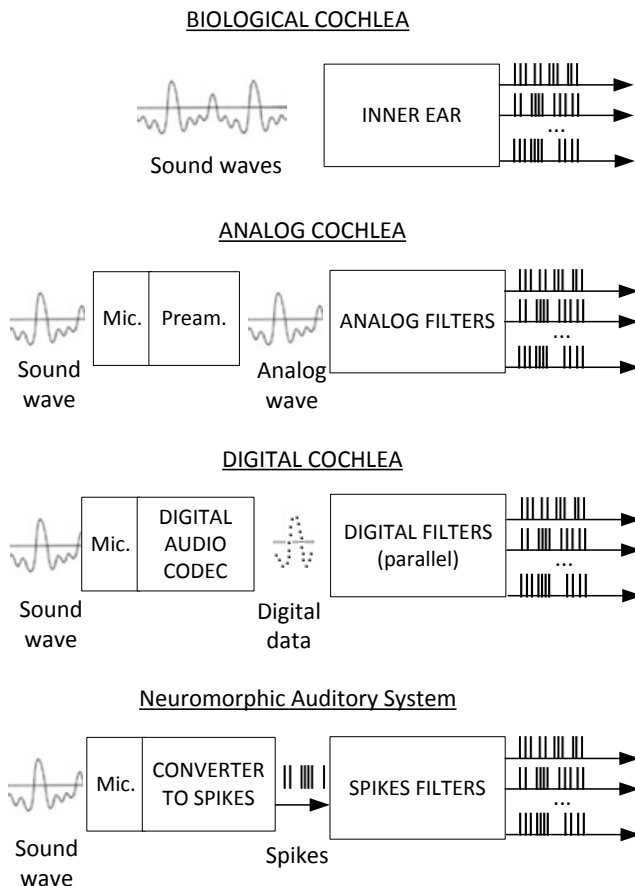


Figura 5.1. Forma de procesar la información de los distintos tipos de cócleas

<sup>18</sup> Un codec de audio es un dispositivo capaz de codificar o decodificar una señal digital en una señal de audio

A continuación vamos a exponer el sistema neuro-mórfico de audición, en adelante **NAS** (Neuromorphic Auditory System), que hemos desarrollado basándonos en la arquitectura planteada en el trabajo (Jiménez-Fernández 2010) y usando los componentes detallados en los trabajos (Jimenez-Fernandez et al. 2011) y (Domínguez-Morales et al. 2011) para obtener un dispositivo que emula el comportamiento de la cóclea biológica, consistente en la descomposición en componentes frecuenciales del sonido que la excita. Resumiendo, en esos trabajos se presentan bloques pulsantes que replican sistemas analógicos basándose en elementos simples de tratamiento de pulsos (sumador, integrador, derivador...). Este dispositivo, como característica peculiar, procesa la información directamente codificada mediante spikes usando la modulación por frecuencia de pulso, o Pulse Frequency Modulation (PFM). En este capítulo, vamos a comenzar explicando la arquitectura del NAS, a continuación se expondrá el procedimiento que se ha realizado para la sintonización de las bandas y del resto de componentes y terminaremos analizando su comportamiento de manera experimental.

## **5.1. Arquitectura general del sistema neuromórfico de audición**

Para desarrollar la arquitectura de nuestro NAS nos hemos basado en el funcionamiento del aparato auditivo, en concreto en el comportamiento de la cóclea biológica ante estímulos auditivos, acorde con el modelo de Lyon (Lyon 1982) y con una arquitectura en cascada planteada por Lyon y Mead (Lyon & Mead 1988), de forma que el sonido se propague en nuestro sistema y se realice la descomposición en frecuencias imitando en cierta medida a la membrana basilar de la cóclea humana. El objetivo es obtener una implementación en VHDL para FPGAs cuya entrada es el sonido analógico y cuya salida en la descomposición en componentes frecuencias de la entrada codificada según la representación AER. Como se ha comentado previamente, partimos del trabajo (Jiménez-Fernández 2010) para desarrollar nuestro NAS, trabajo que no propone una cóclea sintética concreta, sino un procedimiento para implementarla, por lo tanto, lo primero que nos planteamos es qué características necesitamos que tenga nuestro NAS.

Para decidir las características relacionadas con la frecuencia de nuestro sistema, es decir, el rango de frecuencia y el número de bandas, es importante plantearse el procesamiento posterior que se quiere realizar con dicha información. El objetivo de la segunda parte de nuestro trabajo es reconocer sonidos basándonos en las frecuencias características que los forman, por lo tanto decidimos usar un rango de frecuencias similar al rango del oído humano, aunque gracias a la flexibilidad en el diseño de un NAS, estas bandas pueden adaptarse a la aplicación concreta a la que va destinado. El número de bandas por el que optamos es de 64 porque es el mayor número de bandas que se pueden sintetizar en la FPGA Virtex5 FPGA XC5VFX70T de la que consta la plataforma de evaluación que tenemos disponible para este trabajo, Xilinx Virtex-5 FXT FPGA ML507 (Xilinx-ML507 2015).

En la Figura 5.2 (a) se muestra el diagrama de bloques de la arquitectura general de un NAS estéreo, o binaural, basado en spikes con un número indeterminado de canales. El sistema recibe el sonido analógico que se digitaliza gracias al códec de audio AD1981, el cual cumple el estándar AC'97 (Intel 2002) incluido en la ML507. Este dispositivo soporta audio estéreo y usa 48KHz de tasa de muestreo y 16 bits para la cuantificación digital. Una vez que está el audio digitalizado, se transfiere a la FPGA usando el bus estándar AC-link (Intel 2002). En la FPGA, el sonido digital pasa a un módulo, nombrado en la Figura 5.2 (a) como *AC'97 FSM*, encargado de recibir el sonido del códec de audio de forma serie y transferirlo a dos generadores de spikes como dos palabras de bits en paralelo, una para el canal izquierdo y otra para el canal derecho. Como se observa en la Figura 5.2 (a), este módulo gestiona la información digital codificada en 20 bits porque aunque el códec de audio específico de este trabajo digitaliza a 16 bits, el sistema integrado en la FPGA está preparado para el máximo de bits que soporta el protocolo AC-link (Intel 2002). A continuación el sonido digital de 20 bits pasa a los generadores de spikes, que se encargan de generar una secuencia de spikes que codifica el sonido digital de entrada, modificando de la frecuencia de los spikes de salida. Una vez que tenemos la información codificada en spikes, está preparada para excitar el banco de filtros de spikes paso de banda (Domínguez-Morales et al. 2011), con distintas bandas de paso o canales, de los que se dan más detalles a lo largo de este capítulo. Finalmente, la salida de los bancos de filtros es dirigida

hacia un monitor AER distribuido, el cual codificará cada spike como un evento AER y lo transmitirá a las capas de procesamiento de audio basadas en spikes. El monitor de spikes que usamos es el monitor de spikes distribuido expuesto en el capítulo anterior, usamos este monitor en lugar del masivo porque el NAS que implementamos tiene un alto número de canales y por lo tanto de posibles colisiones y una alta tasa de eventos salida (mayor de 3MEvents/s por encima del límite que soporta el monitor masivo sin generar pérdidas).

Para implementar el banco de filtro paso de banda hemos usado la arquitectura del modelo en cascada propuesto por Lyon, de manera que vamos a conectar 65 filtros paso de baja de spikes de segundo orden, llamados *Spikes Low Pass Filter* (SLPF), en cascada para obtener los 64 bandas deseadas, siendo la salida de cada filtro paso de baja la entrada del siguiente. El primer filtro es el que tiene la frecuencia de corte más alta y los demás tienen frecuencias de corte decrecientes exponencialmente con el objetivo de imitar el funcionamiento de la membrana basilar. La salida del filtro paso de baja será propagada hacia el siguiente filtro y así de forma sucesiva hasta llegar al último filtro. Para obtener los spikes de la banda correspondiente se resta la salida de los filtros consecutivos usando un bloque llamado Spikes Hold&Fire (SH&F) (Jimenez-Fernandez et al. 2010). Ésta diferencia que el sistema hace para obtener la salida es la razón de que se necesite un filtro paso de baja más respecto al número de canales paso de banda queramos que tenga nuestro sistema. En Figura 5.2 (b) se muestra un extracto de las tres primeras bandas de este mecanismo de conexión de filtros en cascada para obtener los distintos canales de frecuencia de nuestro NAS. Es muy importante destacar el divisor de spikes que se implementa a la salida de cada banda, que se usa para que la ganancia de las bandas pueda ser adaptable. Ese componente se llama Spikes Frequency Divider (SFD) (Jimenez-Fernandez et al. 2010) y es capaz de dividir la tasa de spikes por un número constante.

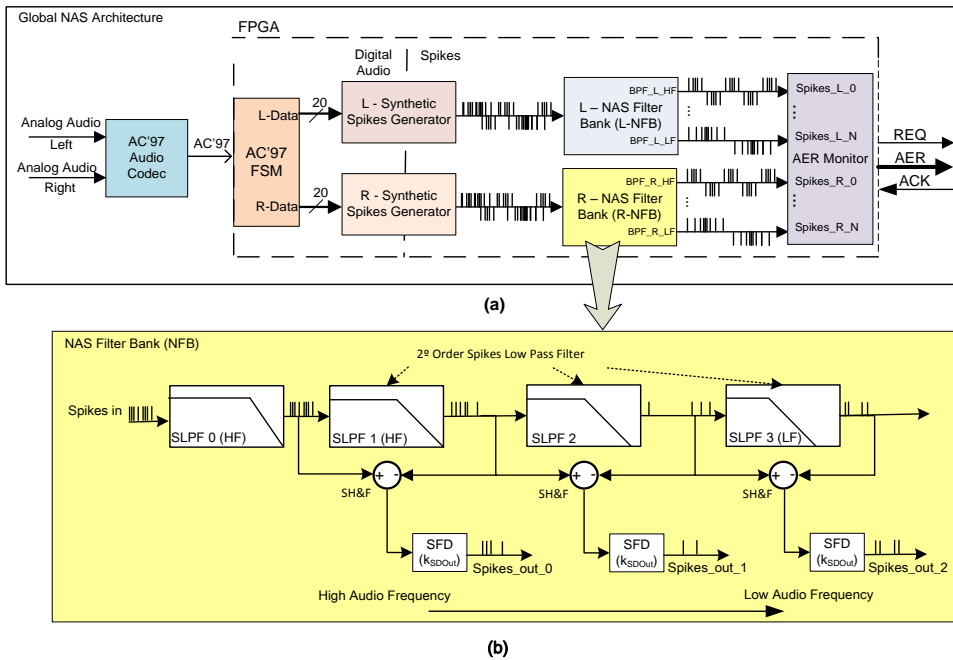


Figura 5.2. Arquitectura del NAS

La mayoría de los bloques que forman parte de la arquitectura ya se han utilizado en trabajos previos: el generador de spikes sintético que hemos usado se llama *Reverse bit-wise Synthetic Spikes Generator* y se ha usado en los trabajos (Gomez-Rodriguez et al. 2005) y (Paz-Vicente et al. 2009), los filtros de paso de baja que filtran señales codificadas en spikes (SLPF), el bloque que resta dos señales codificadas en spikes (SH&F) y el divisor de la frecuencia de los spikes por una constante (SFD) se exponen en los trabajos (Jimenez-Fernandez et al. 2010) y (Jimenez-Fernandez et al. 2011) y se han usado en los trabajos para el control PID (Jimenez-Fernandez et al. 2012) y seguimiento de objetos (Perez-Peña et al. 2013). El monitor de spikes distribuido es un componente innovador desarrollado en el contexto de este trabajo, presentado en el trabajo (Cerezuela-Escudero et al. 2013). Pero respecto a la funcionalidad de este conjunto de bloques descritos, es la primera vez que componen una cóclea artificial.

## 5.2. Implementación del NAS estéreo con 64 canales

Queremos obtener un sistema con la arquitectura previamente explicada para dos señales de entrada con 64 bandas para cada una. A continuación se van a concretar los parámetros para implementar nuestro NAS con los tamaños deseados.

El componente más importante del NAS y el que le va a dar su funcionalidad es el banco de filtros, nombrados en la Figura 5.2 (a) por *L-NAS Filter Bank* y *R-NAS Filter Bank*, según sea la parte del sistema que procesa el audio izquierdo o derecho. Por lo tanto, vamos a comenzar explicando el proceso para la sintonización de las bandas del banco de filtros.

### 5.2.1. Sintonización del banco de filtros de spikes

Con el objetivo de obtener la implementación VHDL del NAS tal y como se plantea en el apartado anterior de este capítulo, es necesario calcular las frecuencias de corte de los 65 filtros paso de baja en función de las frecuencias centrales deseadas para cada banda. La ganancia de cada filtro paso de baja está fijada a uno, ya que al estar conectados en cascada, si los filtros paso de baja tuvieran una ganancia distinta de 1 irían amplificando y/o atenuando la señal de entrada que se propagaría en los filtros paso de baja siguientes (de menor frecuencia de corte). Teniendo en cuenta esta restricción, la función de transferencia equivalente de cada filtro paso de banda, como la resta entre dos filtros paso de baja adyacentes, y considerando que la función de transferencia de cada filtro paso de baja está compuesta por multiplicación de las funciones de transferencia de los filtros paso de baja con frecuencias de corte más altas (los filtros paso de baja anteriores), y por último considerando que cada filtro paso de baja es de segundo orden, la función de transferencia equivalente de cada filtro paso de banda de la cóclea podría ser calculado tal y como se indica en la Ecuación 5.1. Dicha ecuación muestra la función de transferencia del filtro paso de banda  $i$ -ésimo, donde  $w_{L_{PF}_k}$  es la frecuencia de corte del filtro paso de baja  $k$ -ésimo (anteriores a la banda  $i+1$ ) y  $w_{L_{PF}_{i+1}}$  es la frecuencia de corte del filtro paso de baja  $i+1$ -ésimo. Como se mostraba en la Figura 5.2 (b), la salida de la banda  $i$ -ésima



está formada por la diferencia de la salida del filtro paso de baja  $i$ -ésimo menos la salida del filtro paso de baja  $i+1$ -ésimo, tal y como se expresa en la Ecuación 5.1.

$$\begin{aligned}
 BPF_i(S) &= LPF_i(S) - LPF_{i+1}(S) = \prod_{k=1}^i LPF_k(S) - \prod_{k=1}^{i+1} LPF_k(S) \\
 &= \prod_{k=1}^i \frac{\omega_{LPF\_k}^2}{(s + \omega_{LPF\_k})^2} * \left( 1 - \frac{\omega_{LPF\_i+1}^2}{(s + \omega_{LPF\_i+1})^2} \right)
 \end{aligned}$$

Ecuación 5.1.

Para resolver este problema, es decir, calcular los valores de las frecuencias de corte de los 65 filtros paso de baja, usamos el algoritmo genético planteado en el trabajo (Jiménez-Fernández 2010). Para ello, tenemos que determinar las variables de entrada, compuestas por el número de bandas y las frecuencias centrales de las bandas deseadas. Como explicamos al principio de este capítulo, deseamos que las bandas tengan valores en el mismo rango de frecuencia al que es sensible el oído humano. Por lo tanto, como entrada de este algoritmo hemos seleccionado un banco de 64 filtros y con las frecuencias centrales distribuidas logarítmicamente entre 10Hz y 15kHz, ya que este rango de frecuencia es comúnmente usado en aplicaciones de procesamiento de audio y está dentro del rango de la membrana basilar de la cóclea humana. Aplicando el algoritmo genético, se obtienen las frecuencias centrales expresadas en la Tabla 5.1 y en la Figura 5.3. El error relativo de cada banda se calcula mediante la Ecuación 5.2 y el error relativo en global del resultado obtenido mediante la Ecuación 5.3. El error global de todas las bandas queda por debajo del 20%, que para el sistema de descomposición de audio que necesitamos es una medida aceptable.

$$Error_{Relative\_i} = \frac{|\omega_{BPFNAS\_i} - \omega_{BPFideal\_i}|}{\omega_{BPFideal\_i}}$$

Ecuación 5.2.

$$Error_{Relative} = \frac{\sum_{i=1}^N Error_{Relative.i}}{N}$$

Ecuación 5.3.

Tabla 5.1. Frecuencias centrales de cada banda del NAS

Canal del NAS	Frecuencia central paso de banda	Frecuencia central ideal paso de banda	Error relativo	Canal del NAS	Frecuencia central paso de banda	Frecuencia central ideal paso de banda	Error relativo
1	15.541kHz	15kHz	3.60	33	258,04	365.46	29.39
2	14060,47	13356.28	5.27	34	224,74	325.40	30.93
3	12065,30	11892.45	1.45	35	199,10	289.74	31.28
4	10331,27	10589.05	2.43	36	181,32	257.98	29.71
5	8496,31	9428.50	9.88	37	169,04	229.71	26.41
6	7384,27	8395.15	12.04	38	145,36	204.53	28.92
7	6840,327	7475.05	8.49	39	126,87	182.11	30.33
8	6098,60	6655.79	8.37	40	111,92	162.15	30.97
9	5402,75	5926.32	8.83	41	98,10	144.38	32.05
10	4626,26	5276.80	12.32	42	88,96	128.56	30.79
11	4231,18	4698.47	9.94	43	81,19	114.47	29.06
12	3693,04	4183.52	11.72	44	72,70	101.92	28.67
13	3285,60	3725.01	11.79	45	64,81	90.75	28.57
14	2867,73	3316.76	13.53	46	57,91	80.80	28.33
15	2487,09	2953.24	15.78	47	52,29	71.95	27.31
16	2255,43	2629.57	14.22	48	47,93	64.06	25.18
17	1960,23	2341.37	16.27	49	42,64	57.04	25.24
18	1682,07	2084.76	19.31	50	39,33	50.79	22.55
19	1410,03	1856.27	24.03	51	34,47	45.22	23.76
20	1275,98	1652.83	22.79	52	32,21	40.26	20.01
21	1215,10	1471.68	17.43	53	29,64	35.85	17.31
22	1096,94	1310.38	25.97	54	24,85	31.92	22.15
23	969,99	1166.77	16.86	55	21,83	28.42	23.20
24	794,32	1038.89	23.54	56	21,97	25.31	13.19
25	671,54	925.03	27.40	57	19,25	22.53	14.54
26	582,41	823.65	29.28	58	17,53	20.06	12.59
27	533,80	733.37	27.21	59	16,70	17.86	6.52
28	459,03	653.00	29.70	60	14,00	15.90	11.99
29	393,06	581.43	32.39	61	12,53	14.16	12.99
30	393,06	517.70	24.07	62	12,32	12.61	0.61
31	327,39	460.96	28.97	63	11,78	11.23	4.94
32	280,94	410.44	31.55	64	9,65	10	3.48

En la Figura 5.3 se puede observar gráficamente la diferencia entre las frecuencias centrales de las bandas deseadas y las obtenidas.

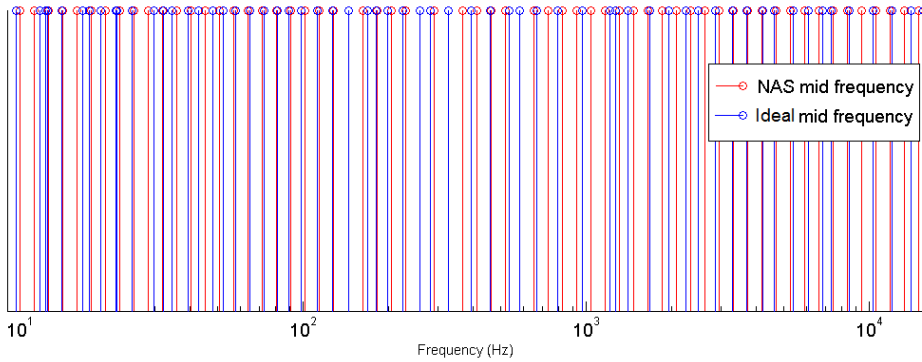


Figura 5.3. Gráfica que muestra la frecuencia central de cada banda deseada y obtenida

Una vez conseguidos los valores para sintonizar los filtros, nos faltan por obtener la implementación del resto de componentes que han sido desarrollados de forma que sean adaptables a diferentes tamaños de buses de señales y valores parametrizables.

### 5.2.1. Sintonización del generador de spikes reverse bit-wise

Como se observa en la Figura 5.2, el componente que recibe el audio digitalizado es el generador de spikes exhaustivo bit-wise que es capaz de proporcionar secuencias de spikes con una frecuencia proporcional en base a números discretos aplicados a su entrada. En este generador, la frecuencia de los spikes de salida es proporcional al valor discreto proporcionado a su entrada; tal y como se muestra en la Ecuación 5.4, donde  $x$  representa el valor de entrada, y  $k_{\text{Freq\_Modulation}}$  la constante de modulación en frecuencia.

$$f(x)_{\text{spikes}} = k_{\text{Freq\_Modulation}} * x$$

Ecuación 5.4.

Existen diversas propuestas de generadores de spikes sintéticos basados en FPGAs, y orientados generalmente a la generación de imágenes (Linares-Barranco et al. 2002), (Linares-Barranco 2003), (Gomez-Rodriguez et al. 2005): 'Scan', 'Uniform', 'Random', 'Random-Square' y 'Exhaustive'. Dado que para aplicaciones de control necesitamos métodos que distribuyan de manera homogénea los spikes en el tiempo, nos vamos a centrar en los métodos exhaustivos (Gomez-Rodriguez et al. 2005), (Linares-Barranco et al. 2002). Estos métodos discretizan el tiempo en frames, estando cada frame dividido a su vez por slices, los cuales toman un ciclo de reloj. Se disparan spikes en los slices adecuados para obtener una frecuencia de salida adecuada y con una distribución de los spikes lo más homogénea posible.

Uno de estos métodos exhaustivos es el método bit-wise expuestos en los trabajos (Paz-Vicente 2008) y (Paz-Vicente et al. 2009), este método consiste en comparar el valor de entrada con un contador; de tal manera que cuando el valor de entrada sea mayor que el contador se dispara un spike, pero para lograr una distribución homogénea en el tiempo de los spikes, en vez de comparar el valor de entrada con el contador, vamos a compararlo con el contador con sus bits invertidos. Gracias a la inversión de bits (bit-wise) el valor invertido del contador va alternando entre diferentes valores, evitando una comparación secuencial entre el valor de entrada y contador. En la Tabla 5.2 se observa la secuencia de emisión de spikes para una implementación del método exhaustivo bit-wise de 3 bits, donde en la primera fila se haya el contador, en el resto de las celdas el valor del contador con los bits invertidos (con el que se comparará el valor de entrada), y finalmente en gris se representan los disparados y en blanco los no disparados.

Tabla 5.2. Secuencia de emisión de spikes de un generador exhaustivo bit-wise de 3 bits

Entrada	Contador							
	0	1	2	3	4	5	6	7
0	0	4	2	6	1	5	3	7
1	0	4	2	6	1	5	3	7
2	0	4	2	6	1	5	3	7
3	0	4	2	6	1	5	3	7
4	0	4	2	6	1	5	3	7
5	0	4	2	6	1	5	3	7
6	0	4	2	6	1	5	3	7
7	0	4	2	6	1	5	3	7

En nuestro caso particular nos interesa poder disparar spikes con signo, es decir, spike positivos así como negativos. Para ello usamos una adaptación del método bit-wise pudiendo así manejar números con signo, codificados en complemento a 2. En primer lugar, la salida del generador serán dos señales de un bit, disparando en una de ellas los spikes positivos y en la otra los spikes negativos. A continuación, en vez de comparar directamente el dato de entrada con el contador se usa el valor absoluto del dato de entrada para generar los spikes; redirigiéndolos por la señal adecuada en base a su signo.

La Figura 5.4 muestra el diagrama de bloques representativo del circuito que implementa este método en hardware, descrito en VHDL. El contador digital está en la parte superior de la ilustración, previo al inversor de bits. El tamaño de ambos es  $n$  bits. Abajo a la izquierda vemos la señal de entrada, la cual atraviesa un bloque que calcula su valor absoluto, siendo ésta última la entrada del comparador digital, el cual disparará un spike si el valor de entrada es mayor que el valor del registro bit-wise. Finalmente nos encontramos un demultiplexor, con el que dirigiremos el spike generador por la señal adecuada, siendo la señal de selección del demultiplexor el bit más significativo del valor de entrada (1 en caso de ser un número negativo y 0 en otro caso). Además, tiene un divisor de frecuencia que activará una señal de *clock enable* (*CE*), tal como se observa en la esquina superior izquierda de la Figura 5.4. Este circuito toma un valor de entrada (*genFD*), y sólo activa la señal *CE* cuando han pasado un número de ciclos de reloj equivalente al valor de la señal *genFD*, actuando como un divisor de reloj entre el valor *genFD*.

Esta señal tiene dos funciones, la primera es que el contador digital sólo se incremente cuando está activa, así como sólo en este caso se pueden disparar los spikes, ya que si no los spikes ocuparían varios ciclos de reloj. En consecuencia, analíticamente, la frecuencia de los spikes generados puede ser calculada acorde a la Ecuación 5.5, donde  $F_{CLK}$  es la frecuencia de reloj del generador,  $n$  el número de bits del contador y  $genFD$  el divisor de reloj. Variando estos valores podremos ajustar la recta de modulación del generador a nuestras necesidades en cada caso. La ganancia del generador por lo tanto depende de la frecuencia del reloj, del número de bits que usemos para el contador y del valor de  $genFD$ . Para nuestro trabajo es importante ajustar con precisión la ganancia del generador porque está relacionado con la ganancia de nuestro sistema. Hemos usado un contador de 16 bits. La señal de reloj va a 27MHz y el valor de  $genF$  lo hemos fijado a 000Fh. Tenemos  $2^{16}$  valores distintos disponibles (porque son valores en binario natural), pero decidimos marcar un valor bajo para obtener una alta tasa de spikes.

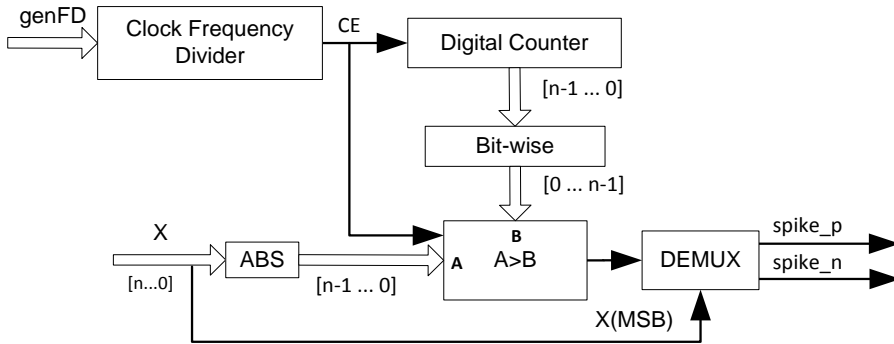


Figura 5.4. Diagrama de bloques del generador Reverse bit-wise

$$f(entrada)_{spikes} = \frac{F_{CLK}}{2^{n-1}(genFD + 1)} * entrada$$

Ecuación 5.5.

Esta arquitectura ha sido seleccionada principalmente por su simplicidad: sólo necesita un contador digital y comparadores. Además del reducido consumo de recursos hardware, presenta una buena distribución temporal, es decir, asegura en

gran medida una distribución homogénea de spikes a lo largo del tiempo (Paz-Vicente et al. 2009).

### 5.2.2. Sintonización del divisor de spikes basado en el método reverse bit-wise

Tal y como se ha explicado en el apartado anterior, el generador bit-wise generaba spikes distribuidos lo más homogéneamente posible a lo largo del tiempo (Paz-Vicente et al. 2009). El divisor de spikes basado en el método bit-wise radica en usar un contador, cuyas entradas se incrementan con cada spike que es recibido en su puerto de entrada, invirtiendo bit a bit la salida del contador y comparándola con el valor del divisor de spikes, habilitando finalmente, o no, un buffer situado a su salida. De esta manera, mantenemos abierto el “tránsito de spikes” un número de veces inversamente proporcional al valor del divisor de frecuencia de los spikes.

En la Figura 5.5 se muestran los componentes internos del divisor de spikes basado en el método bit-wise. En la parte superior de la figura encontramos el contador digital, el cual se incrementará cada vez que reciba un spike, ya sea negativo o positivo. La salida de dicho contador es invertida bit a bit y comparada con el valor del divisor de spikes, *spikesDiv*, en la parte central de la figura. Finalmente, la salida del comparador habilitará un buffer que permitirá el tránsito de los spikes presentes en su entrada. De tal manera que la proporción de tiempo que el buffer estará dispuesto a dejar pasar un spike, será el cociente entre el valor *spikesDiv* y la potencia del número de bits del contador del divisor de spikes. En consecuencia, la frecuencia de los spikes de salida es proporcional a dicho cociente, comprendido entre 0 y 1.

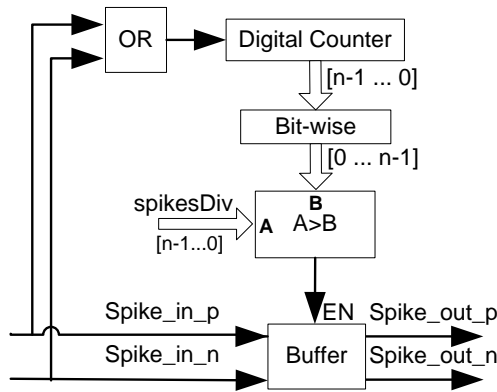


Figura 5.5. Arquitectura del divisor de spikes bit-wise

La frecuencia de salida cumple la Ecuación 5.6, donde el valor de *spikesDiv* está codificado en Complemento a 2 y en nuestra implementación usamos 16 bits como valor de *n* (tamaño el contador, del registro bit-wise y de *spikesDiv*) y sólo vamos a usar valores positivos para el coeficiente (nos interesa multiplicar por valores positivos). Por lo tanto, tenemos  $2^{15}$  valores posibles para *spikesDIV* y en una primera aproximación de implementación del NAS hemos decidido usar el valor “1FFFh” para los divisores de todas las bandas.

$$f_{spikesDivOut} = \frac{spikesDiv}{2^{n-1}} * f_{spikesDivIn}$$

Ecuación 5.6.

### 5.2.3. Sintonización del monitor distribuido de spikes

En el capítulo anterior se han explicado y analizado dos monitores de spikes, concluyendo que el monitor distribuido tiene un comportamiento óptimo ante alta tasa de spikes (menor de 5MSpikes/s). Para nuestro sistema hemos decidido usar el monitor distribuido porque en este sistema vamos a tener una alta actividad de spikes.



El monitor distribuido lo implementamos de forma que sólo hay que indicar los valores de dos parámetros: el número de líneas de entrada que se corresponden con distintas neuronas y el número de bits de salida necesarios para codificar las distintas neuronas. Según la arquitectura del NAS, los bancos de filtros son los que generan los spikes de salida de cada banda, por lo tanto en este sistema las neuronas son las diferentes bandas. El número de señales de salida de los banco de filtros se pueden calcular mediante la Ecuación 5.7. En nuestra implementación, el número de bancos de filtros es 2 porque implementamos un NAS estéreo; el número de canales de cada banco de filtros en nuestro caso lo hemos marcado a 64 debido a las restricciones respecto el máximo número de slices disponibles en la FPGA en la que vamos a sintetizar nuestro NAS; por último estos datos se multiplican por 2 porque cada banda puede disparar spikes positivos y negativos. Por lo tanto, tenemos un total de 256 señales para comunicar los spikes, es decir, el monitor tiene que gestionar eventos AER con direcciones en el rango [0, 255], concluyendo que se necesitan 8 bits del bus AER para enviar las direcciones AER generadas por el NAS. La salida de este componente son las señales necesarias para el bus AER, que son 16 líneas (de las que sólo vamos a usar los 8 bits menos significativos) para los datos de los eventos AER y la señal de ACK y REQ.

$$N^{\circ} \text{ de señales de spikes} = N^{\circ} \text{ bancos de filtros} * N^{\circ} \text{ de bandas} * 2$$

Ecuación 5.7.

#### 5.2.4. Síntesis de NAS

En el momento que ya tenemos los tamaños de los componentes y los parámetros calculados, procedemos a desarrollar el código VHDL que describa la arquitectura del banco de filtros planteada. Para ello, tenemos disponibles unos scripts de Matlab que generan el código VHDL de los bancos de filtros usando los valores de los parámetros obtenidos en el proceso de sintonización. Falta la integración de los dos bancos de filtros con los generadores de spikes con los módulos que se encargan de la comunicación con el exterior, que son el módulo AC'97 FSM, encargado de recibir el sonido digital del códec de audio AC'97 y

transferirlo a los generadores de spikes, y con el módulo que implementa el monitor de spikes distribuido. A continuación, vamos a presentar las características físicas que influyen en la implementación de estos dos componentes periféricos.

La FPGA en la que hemos sintetizado nuestro sistema es una Virtex5 (XC5VFX70T) incluida en la placa de desarrollo de Xilinx ML507 (Xilinx-ML507 2015). En dicha placa está integrado el códec de audio AD1981 que cumple el estándar AC'97 que soporta audio estéreo de 16 bits con una frecuencia de muestreo de 48Khz. El protocolo de comunicación AC-link es un protocolo serie de 6 señales con las que se van a transferir los datos a la FPGA al módulo AC'97 FSM (ver Figura 5.2). Dicho componente genera datos digitales de 20 bits porque está diseñado para que funcione con cualquier códec de audio que use el protocolo AC-link.

El componente que se encarga de la comunicación de salida del dispositivo es el monitor, dispositivo que transforma los spikes en eventos AER para su transferencia por el bus AER. El número de señales de salida que necesita dicha comunicación está relacionado con la estructura de las direcciones AER para el NAS. En la Figura 5.6 se observa el formato de la dirección AER que genera el NAS, por lo tanto, como en nuestro sistema tenemos 64 canales, necesitamos 6 bits para determinar el canal, un bit para determinar si es positivo o negativo el spike y otro bit para determinar si el spikes es del NAS izquierdo o derecho. Resumiendo, el tamaño de la dirección AER es 8 bits.

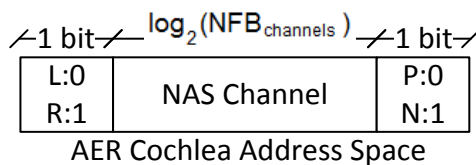


Figura 5.6. Estructura de las direcciones AER de salida del NAS

Por lo tanto, en nuestra implementación del NAS el número de señales de entrada/salida del sistema son: la señal de reloj del NAS, la señal de reset, 6 señales para el manejo de la comunicación AC'97 con el codificador analógico digital de

audio y las señales de comunicación del bus AER, que en nuestro caso son 8 para codificar las direcciones y 2 señales usadas para la comunicación asíncrona del bus AER (ACK y REQ). En total, 18 señales. Las señales del protocolo AER de salida del monitor se realiza mediante los pines GPIO disponibles en la placa de entrenamiento.

Los requisitos hardware necesarios para esta implementación del NAS son 11.141 slices, que es el 99% de los slices disponibles que tenemos en la FPGA que estamos usando. Se han analizado los requisitos hardware en función del número de canales del NAS, como son el número de slices requeridos por la FPGA, la máxima frecuencia de reloj, el consumo de potencia y el número de señales de entrada/salida. En la Tabla 5.3 se muestran los requisitos hardware para diferentes números de canales del NAS.

Tabla 5.3. Requisitos hardware para diferentes números de canales del NAS

Nº de canales	Nº de Slices % Utilización	Max. Frecuencia de CLK (MHz)	Power (mW)	Señales de E/S
12	4.286 / 38,26%	179,95	6,6	15
16	4.415 / 38,41%	171,73	7,2	16
24	6.301 / 56,25%	113,74	8,6	17
32	7.606 / 67,91%	99,84	14,3	17
48	10.241/ 91,43%	91,86	18,1	18
64	11.141/ 99,47%	87,31	29,7	18

En la Figura 5.7 se muestra la relación entre el número slices requeridos en función del número de canales, y la recta de regresión que cumple dicha relación. La Ecuación 5.8 se corresponde con dicha recta de regresión, que se desvía un 6% del valor real.

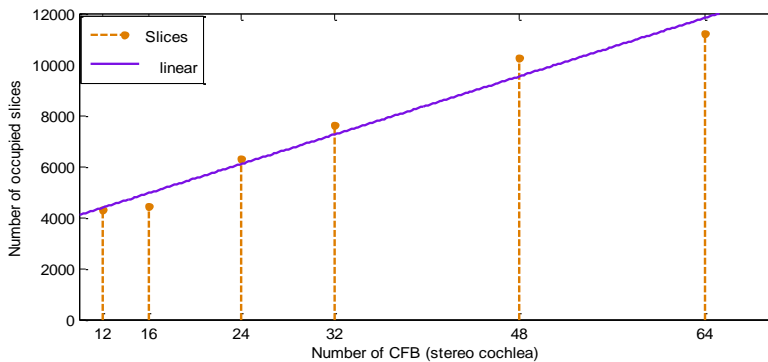


Figura 5.7. Relación entre los requisitos lógicos y el tamaño del banco de filtros

$$Total\ slices \approx 143.2 * channelNumber + 2663$$

Ecuación 5.8.

Hay que destacar que esta arquitectura no demanda recursos FPGA especializados, como son multiplicadores, procesadores embebidos, DSP; sólo necesita lógica digital común, como contadores, comparadores, sumadores y registros con un bajo número de bits y bajo nivel de conectividad, necesitando sólo dos cables de comunicación interna entre las etapas del banco de filtros y los pines de salida para transmitir los spikes de salida como eventos AER.

### 5.3. Escenario experimental del NAS

El escenario experimental del NAS ha sido diseñado y construido para tener disponible el NAS en una plataforma de forma que podamos analizar su comportamiento. Para ello hemos usado la placa de desarrollo de Xilinx ML507 que entre otros componentes, incluye los que necesitamos para este trabajo: FPGA Virtex5 (XC5VFX70T), un códec de audio con el estándar AC'97 y pines GPIO.

En la Figura 5.8 se muestra un diagrama de bloques del escenario experimental que hemos planteado. Para enviar el sonido analógico al NAS hemos usado la mezcladora de sonidos XENYX QX1002USB (Behringer 2015) que tiene una interfaz audio USB por la cual podemos enviarle datos de audio desde el PC. La

mezcladora de sonidos se conecta a la placa de evaluación por un jack de audio que es la línea de entrada del códec AC'97. En el códec se digitaliza el sonido y se envía a la FPGA con el NAS configurado. Los datos digitales son transformados en su representación pulsante por el generador de spikes. La representación pulsante del sonido atravesará los bancos de filtros generando los spikes de salida. Para poder monitorizar dicha actividad de spikes en el PC vamos a usar la placa USB-AERmini2, placa que actúa como puente entre el bus AER y el bus USB con una resolución temporal máxima de  $0,2\mu$ Segundos configurable. Esta placa está detallada en el Anexo de este trabajo y en el trabajo (Berner et al. 2007). También se ha usado en el contexto del trabajo expuesto en el capítulo anterior.

Para poder realizar la comunicación entre el puerto GPIO al bus AER siguiendo el estándar CAVIAR (Rafael Serrano-Gotarredona et al. 2009) desarrollamos un adaptador. Dicho adaptador se puede observar en la Figura 5.9, que es una fotografía del escenario real en el que está desarrollado este trabajo, en la que se puede observar la placa de evaluación Xilinx ML505 en el centro, el programador JTAG en la izquierda y la placa USB-AERmini2 en la derecha. Dentro de la placa de evaluación se puede destacar la FPGA Virtex5 y el códec de audio.

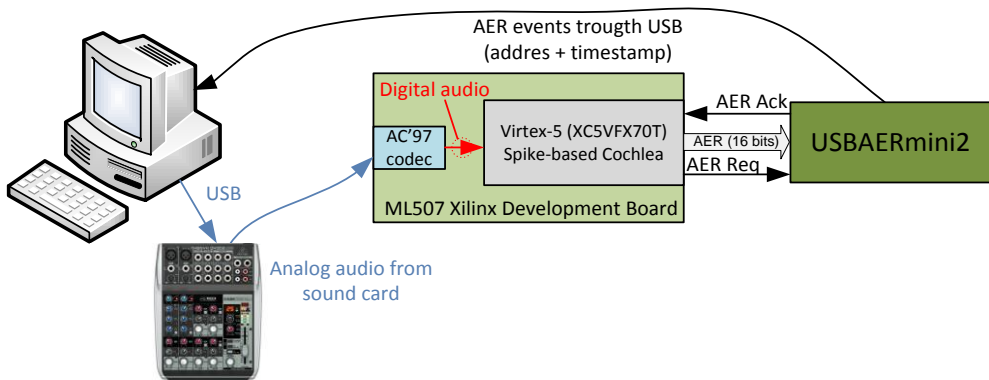


Figura 5.8. Esquema de conexión de los dispositivos del escenario

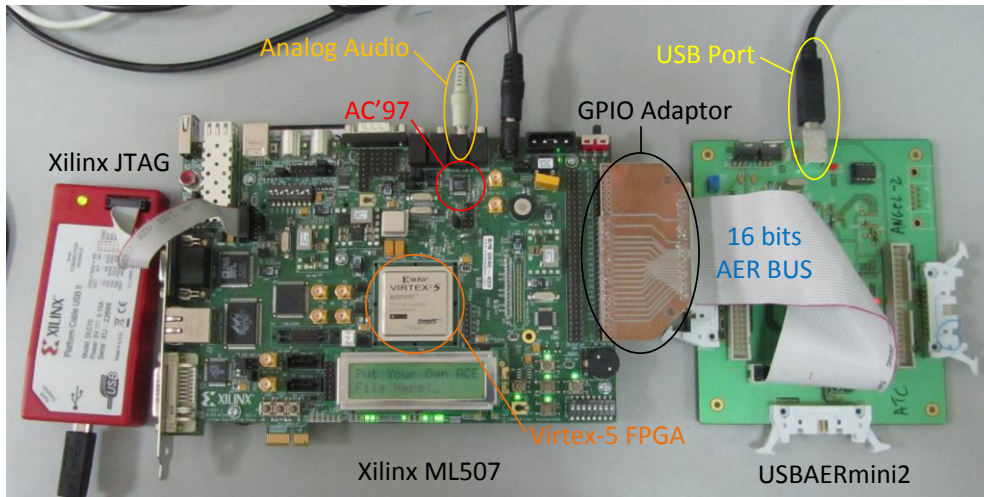


Figura 5.9. Fotografía del escenario experimental

Una vez que tenemos el escenario montado, comprobamos el comportamiento de nuestro sistema mediante una batería de experimentos.

## 5.4. Resultados experimentales del NAS implementado

En el entorno descrito en el apartado anterior hemos desarrollado un conjunto de experimentos con el objetivo de analizar el comportamiento del NAS y extraer sus características. Vamos a analizar el NAS que hemos sintetizado y configurado en el escenario anterior, NAS estero de 64 canales, con una señal de reloj de 27MHz y con un bus a su salida de 8 bits para poder codificar las 256 direcciones de eventos AER distintas que genera el NAS.

### 5.4.1. Respuesta temporal

El primer experimento consiste en obtener la respuesta temporal del NAS estéreo de 64 bandas. La Figura 5.10 muestra el cocleograma en presencia de una mujer diciendo “en un lugar de la mancha”, principio de “El Ingenioso Hidalgo

Don Quijote de la Mancha”; en esta figura el eje  $x$  representa el tiempo y el eje  $y$  representa las direcciones AER que se producen en cada instante de tiempo, que se representan en la gráfica con un punto violeta. Las direcciones desde 0 a 127 se corresponden con el NAS izquierdo y las direcciones desde la 128 a la 255 al derecho, siendo las direcciones pares las correspondientes a los spikes negativos de un canal y las impares a los spikes positivos. Se puede observar una respuesta retrasada en los canales debida a la arquitectura con el banco de filtros en cascada, donde se introduce un retraso en la fase en cada uno de dichos filtros tal y como se estima que ocurre en la cóclea biológica.

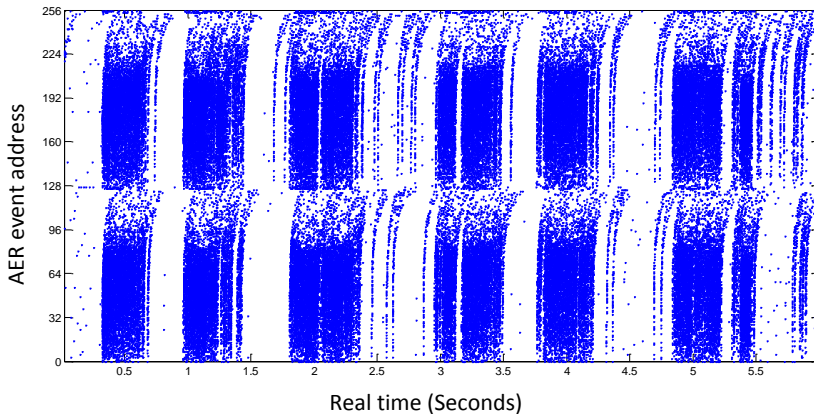


Figura 5.10. Cocleograma del NAS de 64 canales en presencia de voz humana

En este experimento, vamos a medir la tasa de eventos durante un periodo de tiempo de cada banda, obteniendo de cada banda la actividad que hay de spikes tanto positivos y negativos, o en otras palabras, la potencia absoluta de cada canal en términos de spikes. Las bandas desde la 0 a la 63 pertenecen al NAS izquierdo y desde la 64 a la 127 al NAS derecho. En la Figura 5.11 se muestra la tasa de spikes a lo largo del tiempo para los diferentes canales, obteniendo una representación equivalente al sonograma. Esta figura muestra la tasa de eventos del NAS izquierdo (bandas del 0 al 63) con un mapa de color, donde el eje  $x$  es el tiempo, el eje  $y$  es la banda del canal y el color representa la tasa de eventos en ese periodo de

tiempo. Observando esta figura se pueden diferenciar formas que se corresponden con palabras diferentes.

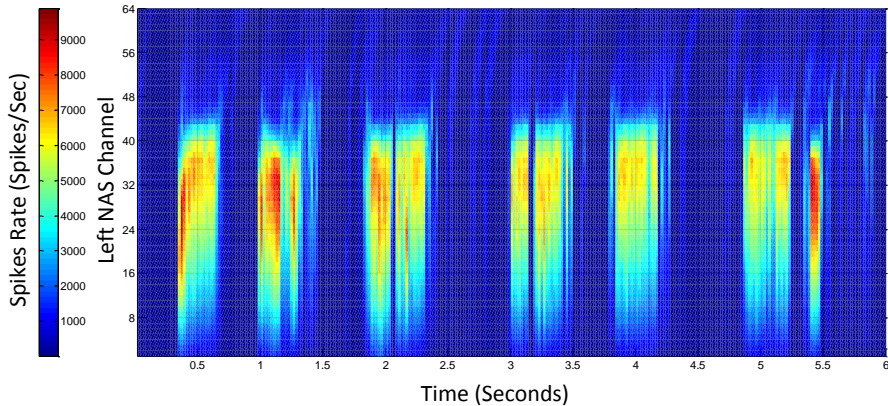


Figura 5.11. Sonograma del NAS izquierdo de 64 canales ante voz habla humana

#### 5.4.2. Características en frecuencia

A continuación, mostramos los resultados obtenidos de los experimentos que hemos realizado para obtener la respuesta de frecuencia de nuestro NAS estero de 64 bandas. Para obtener la respuesta frecuencia del NAS implementado, hemos generado tonos puros senoidales de 2.5mW de potencia con frecuencias que barren desde 10Hz a 20kHz y con un segundo de duración, y hemos obtenido la actividad AER de salida ante cada tono puro. A partir de los eventos AER generados para cada tono puro, hemos calculado la tasa de eventos de cada canal en el periodo de tiempo de captura. Estos datos se representan gráficamente en la Figura 5.12, donde arriba se representa la tasa de eventos del NAS izquierdo y abajo la tasa de eventos del NAS derecho. El eje x representa la frecuencia, el eje y la tasa de eventos y las curvas de distintos colores representan la tasa de eventos para los 64 canales. La línea que aparece en la figura arriba es la tasa de eventos total que se ha generado para el tono puro con la frecuencia que se indica en el eje x y que nos muestra como la tasa de eventos total es menor para frecuencias bajas ante el



mismo volumen. En esta figura se comprueba cómo cada canal del NAS se comporta como filtros paso de banda. También se observa como la tasa de eventos de los filtros paso de banda aumentan conforme aumenta la frecuencia media de las bandas y que las bandas de altas frecuencias presentan un desplazamiento en las zonas de bajas frecuencias.

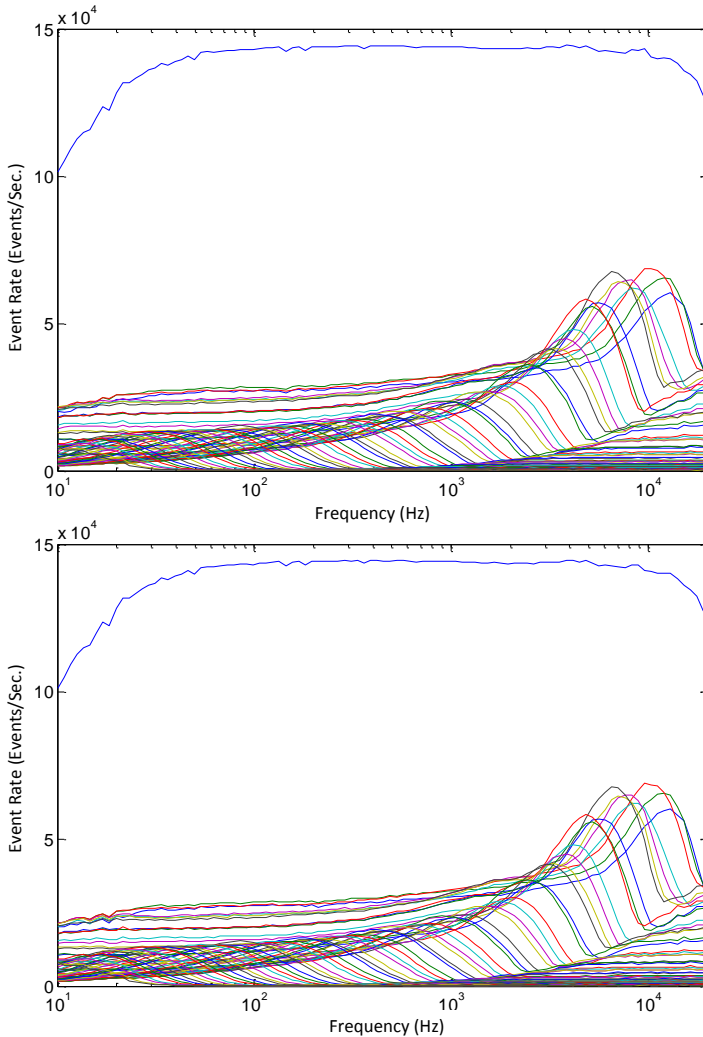


Figura 5.12. Diagrama de Bode del banco de filtros

En la Figura 5.13 se muestra la misma información que en la Figura 5.12 pero mediante un mapa de color, donde el eje  $x$  es la frecuencia, el eje  $y$  son los canales del NAS izquierdo y mediante diferentes colores se expresa la tasa de eventos de cada canal. Esta figura muestra como cada canal tiene su máxima actividad ante una frecuencia determinada (la diagonal que en su mayoría está entre rojo y celeste) y presentan una baja actividad fuera de su banda (azul oscuro). También se observa como conforme aumenta la frecuencia aumenta la actividad de las bandas mediante la diagonal que va cambiando de celeste a rojo. En la Figura 5.13 sólo hemos mostrado la gráfica del NAS izquierdo porque, tal y como se observa en la Figura 5.12, no hay una diferencia significativa en la respuesta obtenida por los canales izquierdo y derecho. La respuesta de ambas NAS, izquierdo y derecho son similares gracias a que este NAS esta implementado mediante circuitos digitales en FPGA a diferencia de las cócleas analógicas, explicadas en el capítulo 3, implementadas mediante filtros analógicos en circuitos aVLSI. Esta tecnología analógica se ve afectada durante la fabricación por un parámetro que se llama mismatch el cual hace que los circuitos idénticos en diseño no tengan exactamente la misma respuesta.

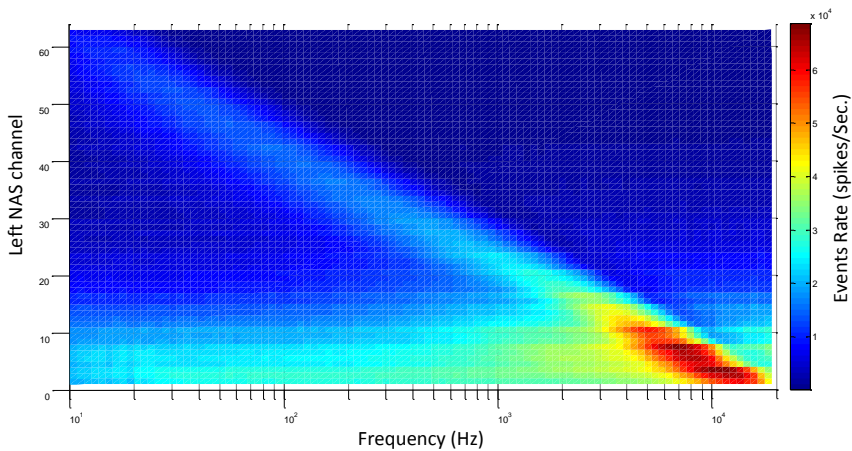


Figura 5.13. Representación en superficie de la respuesta en frecuencia del NAS izquierdo

Usando esta información, podemos medir otros parámetros relacionados con los filtros paso de banda, como son la frecuencia media, el factor de calidad (Q factor) y el ancho de banda, caracterizando el banco de filtros del NAS implementado.

La Figura 5.14 muestra las frecuencias medias y las tasas de eventos máxima de cada canal del banco de filtros obtenida usando los datos del experimento anterior. En esta figura el eje  $x$  representa la frecuencia y el eje  $y$  representa la tasa de eventos. La gráfica representa la frecuencia central de cada canal mediante una línea vertical, además la altura de la línea indica el número de spikes disparados a dicha frecuencia para cada canal. La distribución de las frecuencias medias de las bandas es relativamente homogénea, sin embargo no es perfecto, porque como ya se comentó en el apartado Sintonización del banco de filtros de spikes de este capítulo, es muy difícil calcular analíticamente las frecuencias de corte de los filtros paso de baja que forman un filtro paso de banda y por lo tanto optamos por un algoritmo genético que resuelve el problema pero sólo aproximándose a la solución perfecta. El canal 0 tiene la mayor frecuencia media de 15.541kHz y el canal 63 tiene la menor frecuencia media de 9.6Hz, obteniendo un rango similar al de la cóclea humana. La tasa de eventos del canal de mayor frecuencia es de unos 60kEventos/s mientras que para el canal de menor frecuencia es de 8kEventos/s. Este efecto, que disminuya la tasa de eventos al disminuir la frecuencia media de la banda, se introduce debido a la arquitectura en cascada del banco de filtros.

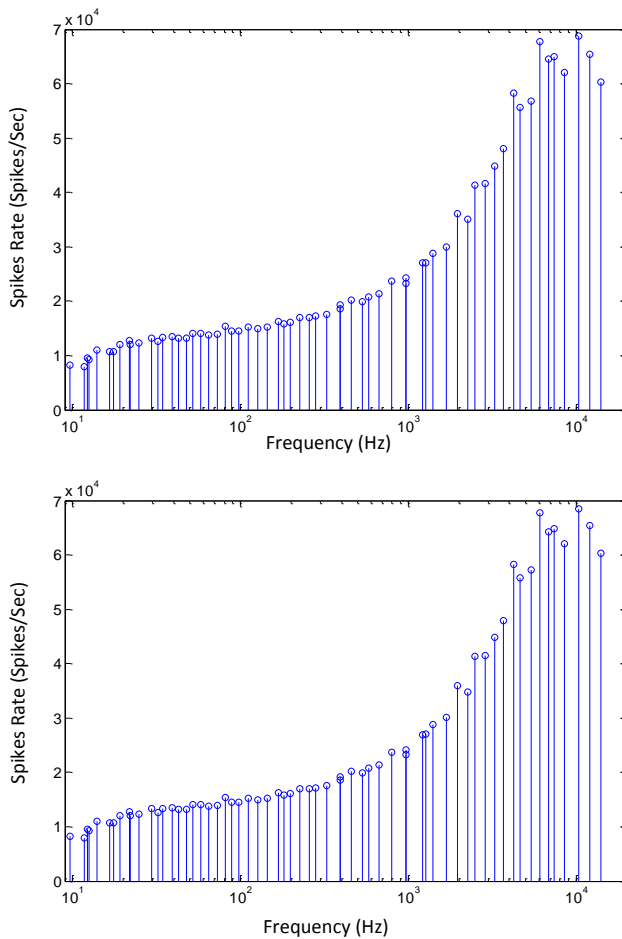


Figura 5.14. Frecuencia media y tasa máxima de los eventos de los 64 canales del NAS izquierdo (arriba) y derecho (abajo)

Otro parámetro que hemos estudiado en este experimento es el factor de calidad de cada filtro paso de banda del banco de filtros. La Figura 5.15 muestra el factor de calidad de la respuesta del banco de filtros del NAS izquierdo, donde el eje  $x$  representa el canal del NAS izquierdo y el eje  $y$  el factor de calidad de esa banda. El factor de calidad varía entre 1.2 a 0.75 con un valor medio de 0.85. El factor de calidad cambia entre las distintas bandas debido, otra vez, a los valores imperfectos obtenidos por el procedimiento de sintonización del banco de filtros, porque este factor depende directamente del valor de la distancia entre dos filtros paso de baja

que forman el banco de filtros. El factor de calidad es similar para ambos NAS, izquierdo y derecho, porque tienen la misma distancia entre la frecuencia de corte de los filtros paso de baja que forman ambos bancos de filtros.

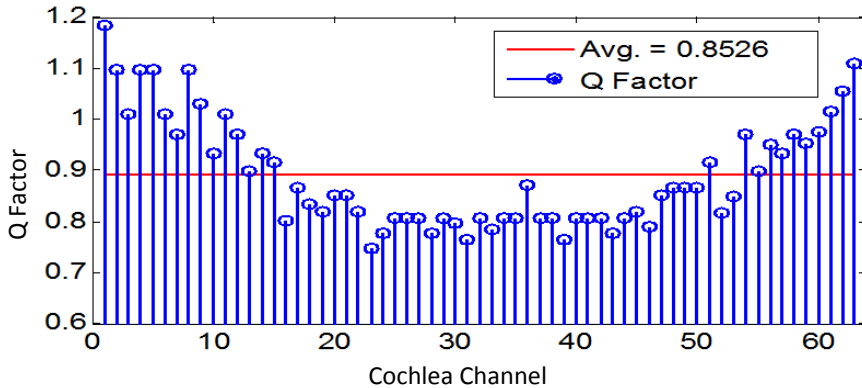


Figura 5.15. Factor de calidad de cada banda del NAS izquierdo

Para terminar con este experimento, la Figura 5.16 muestra el ancho de banda de cada banda del banco de filtros, donde el eje  $x$  representa el número de la banda y el eje  $y$  representa el ancho de banda de cada banda del NAS izquierdo. En esta figura se observa como el ancho de banda está distribuido logarítmicamente desde 15kHz a 10Hz.

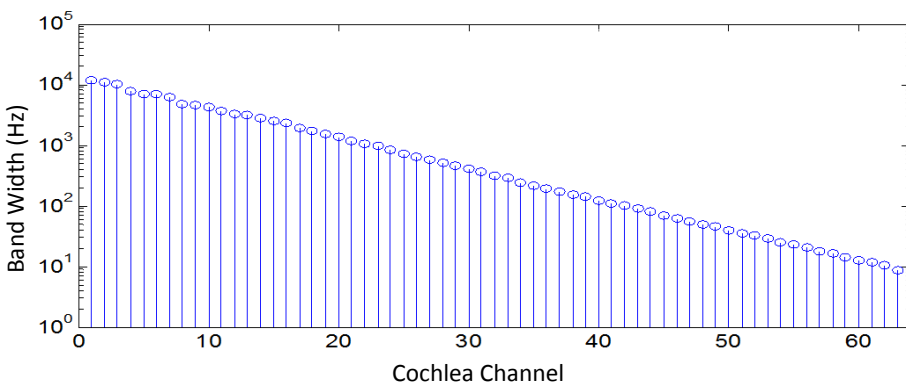


Figura 5.16. Ancho de banda de cada banda del NAS izquierdo

### 5.4.3. Rango dinámico

En esta sección hemos medido el rango dinámico del NAS estéreo de 64 canales. El rango dinámico del NAS representa la potencia del rango de audio frente a la sensibilidad del NAS implementado. El experimento consiste en excitar al NAS con ruido blanco a diferentes niveles de volumen y analizar la tasa de eventos de salida. La Figura 5.17 muestra los resultados experimentales. En el eje  $x$  se representa la potencia del ruido blanco frente al eje  $y$  que representa la tasa de eventos total (para todos los canales) generados por el NAS. La salida del NAS implementado en circunstancia de ausencia de sonido o una baja potencia de audio de entrada tiene una actividad de salida de eventos AER menor a 4 kEvents/s. A un nivel de -60 dBW la actividad AER empieza a incrementar hasta +15dBW, donde la actividad se satura en con una tasa de eventos de 3.3MEvents/s. Por lo tanto, nuestro sistema tiene un rango dinámico de 75dBW en términos de volumen de audio de excitación.

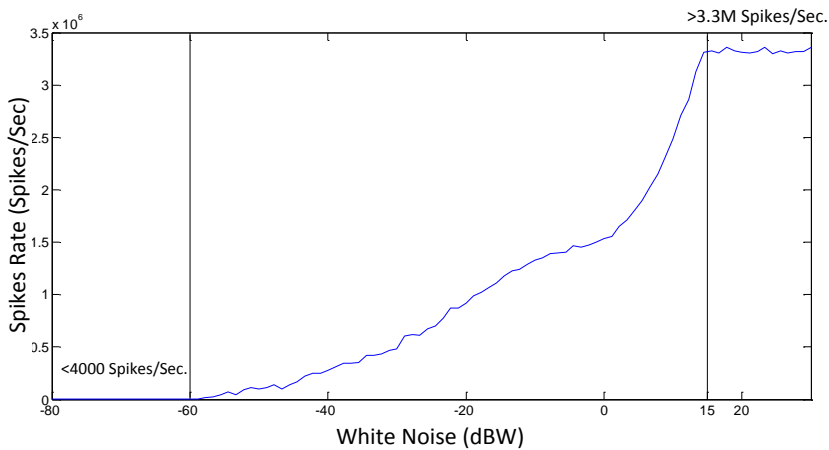


Figura 5.17. Tasa de eventos resultante ante la excitación del NAS para una secuencia de ruido blanco con diferente volumen

## 5.5. Normalización de la primera versión del NAS

Como se observa en la Figura 5.12, que se corresponde con la respuesta frecuencial de la implementación del NAS estéreo de 64 canales, las bandas de mayor frecuencia tienen una mayor tasa de spikes que los canales de baja frecuencia. Nos planteamos si para algunos tipos de procesamiento de audio, como pueden ser la estimación del tono y en los sistemas de reconocimiento donde no interviene un entrenamiento previo, sería más adecuado usar una cóclea cuya tasa de eventos de salida de los canales esté normalizada. Por lo tanto, en este apartado vamos a exponer el proceso por el cual hemos normalizado la tasa de spikes de forma que todas las bandas del NAS tengan la misma ganancia. Para realizar esta tarea es necesario estudiar la arquitectura de los bancos de filtros explicada en el primer apartado de capítulo actual. En dicha arquitectura, la salida de los filtros paso de banda que hemos implementado en el NAS proviene de la diferencia de los spikes de dos filtros paso de baja y a continuación tiene un divisor de la frecuencia de spikes, tal y como se muestra en la Figura 5.18.

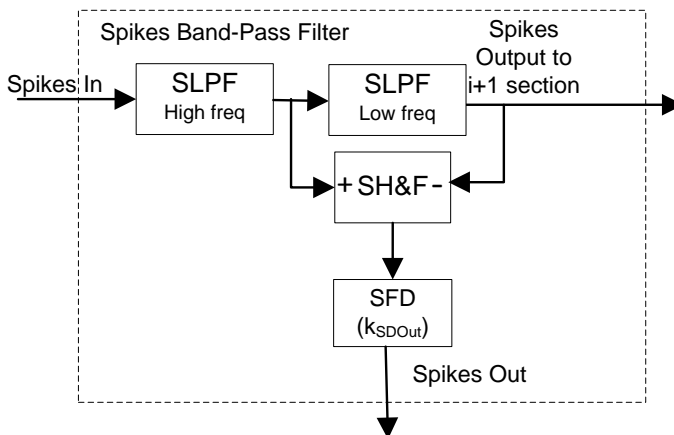


Figura 5.18. Diagrama de bloques de un filtro paso de banda del NAS

La función de transferencia del filtro anterior, suponiendo que es el primer filtro del banco de filtros en cascada cumple la Ecuación 5.9. Tenemos que calcular los

valores de  $k_{SFDout}$  para todos los filtros paso de banda que conforman el banco de filtros de forma que se normalicen las ganancias.

$$F_{SBPF1}(s) = \frac{F_{outSpikes}(s)}{F_{inputSpikes}(s)}$$

$$= k_{SFDout} \left( \frac{W_{HF}^2}{(s + W_{HF})^2} - \frac{(W_{LF} * W_{HF})^2}{(s + W_{HF})^2 * (s + W_{LF})^2} \right)$$

Ecuación 5.9.

Los valores que tenemos que calcular dependen de la función de transferencia del atenuador de spikes, que tal y como se expuso en el apartado Sintonización del divisor de spikes basado en el método reverse bit-wise, cumple la Ecuación 5.10 donde N representa el número de bits que tiene el divisor de spikes y *spikesDiv* es el parámetro con el cual marcar el valor del divisor. La función de transferencia del SFD es equivalente a un bloque de ganancia con un valor en el rango [-1,1], con  $2^N$  valores posible. En nuestro caso tenemos  $2^{15}$  valores distintos porque sólo necesitamos valores positivos.

$$F_{SFD} = \frac{F_{outSpikes}}{F_{inputSpikes}} = \frac{spikesDiv}{2^{N-1}}$$

Ecuación 5.10.

En el NAS que hemos expuesto previamente usa el mismo valor para todos los divisores de salida de los filtros, pero para la implementación con las ganancias iguales para todos los filtros hemos desarrollado un sistema automático que calcula los valores de *spikesDiv* para los 128 canales con el objetivo de conseguir que la ganancia de todos los filtros sea la misma. La arquitectura de este NAS es igual al previamente implementado, la diferencia es que en el apartado de sintonización de los divisores de spikes hay que incluir el proceso automático que calcula los valores de estos componentes.



### 5.5.1. Sintonización de los divisores de spikes

En este apartado vamos a exponer cómo se han calculado los valores de los divisores de spikes de cada banda de forma que se normalicen las ganancias de todas las bandas. Basándonos en la funcionalidad del divisor de spikes explicada previamente, hemos calculado los valores de las constantes necesarias para cada divisor de spikes (spikesDiv) de cada banda mediante un proceso compuesto por los dos pasos siguientes:

- Se ha dividido la tasa de eventos máxima de cada canal por la tasa de eventos total del sonido que ha generado ese valor máximo en el canal. De esta forma se obtiene una medida que relaciona el número de eventos máximo de un canal entre el número de eventos total. Como se observaba en la Figura 5.12, la tasa de eventos total era menor para frecuencias bajas, al generar menos spikes. Por lo tanto este paso lo hemos hecho para obtener una ganancia en las bandas relacionada con la tasa de eventos totales que se generan para el audio de entrada. En la Figura 5.19 se observan estos valores, que son los que hemos usado para obtener las constantes de los atenuadores de spikes.
- Una vez que tenemos los valores obtenidos en el paso anterior, a la banda con menor ganancia, que tal y como se observa en la Figura 5.19 es la segunda banda, se le asigna el valor de la constante que hace mínima la atenuación (valor máximo para 15 bits que es 0x7FFF) y el resto de valores se calculan a partir de ese valor y de los valores obtenidos en el paso 1, tal y como se muestran en la Figura 5.19.

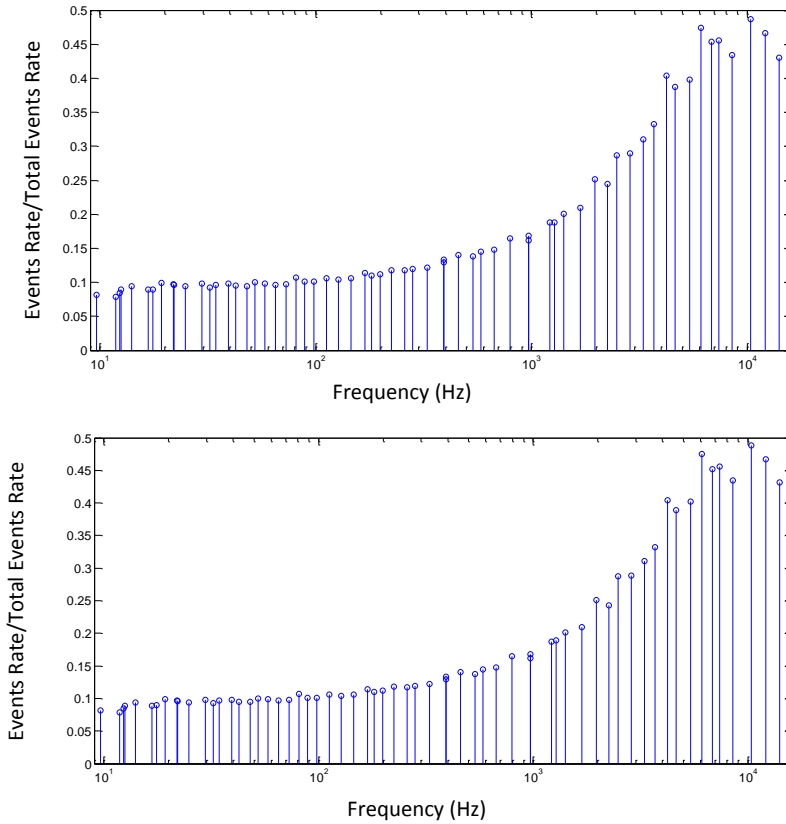


Figura 5.19. Valores a usar para normalizar las ganancias de las bandas

Con los valores anteriores generamos el código VHDL del NAS modificado. Se realiza la síntesis en el mismo escenario propuesto en el apartado Escenario experimental del NAS y procedemos a realizarle los mismos experimentos que en la versión anterior del NAS.

### 5.5.2. Resultados experimentales del NAS normalizado

Para evaluar las características y comportamiento de esta nueva implementación del NAS vamos a realizar los mismos experimentos que realizamos para la implementación anterior.

Con el objetivo de comparar la respuesta temporal de ambas implementaciones, vamos a excitar ambos sistemas con una señal senoidal de 440Hz, una amplitud de un voltio pico a pico y duración 0.2 segundos. En la Figura 5.20 se observa arriba el cocleograma que ha generado el NAS sin normalizar las ganancias (en adelante **NASv1**) y abajo se observa el cocleograma del NAS normalizado (en adelante **NASv2**). El eje  $x$  representa el tiempo y el eje  $y$  representa las direcciones AER que se producen en cada instante de tiempo, que se representan en la gráfica con un punto azul. En ambos casos se puede observar una respuesta retrasada en los canales debida a la arquitectura con el banco de filtros en cascada, donde se introduce un retraso en la fase en cada uno de dichos filtros. En el NASv2 se observan más eventos que en el NASv1 porque al normalizar hemos aumentado la tasa de spikes de la mayoría de los canales (se ha aumentado el factor atenuador de spikes de 53 canales respecto a NASv1). La tasa de eventos ante este experimento para NASv1 es  $6,8e^5$  Spikes/s y de NASv2 es de  $13.14e^5$ .

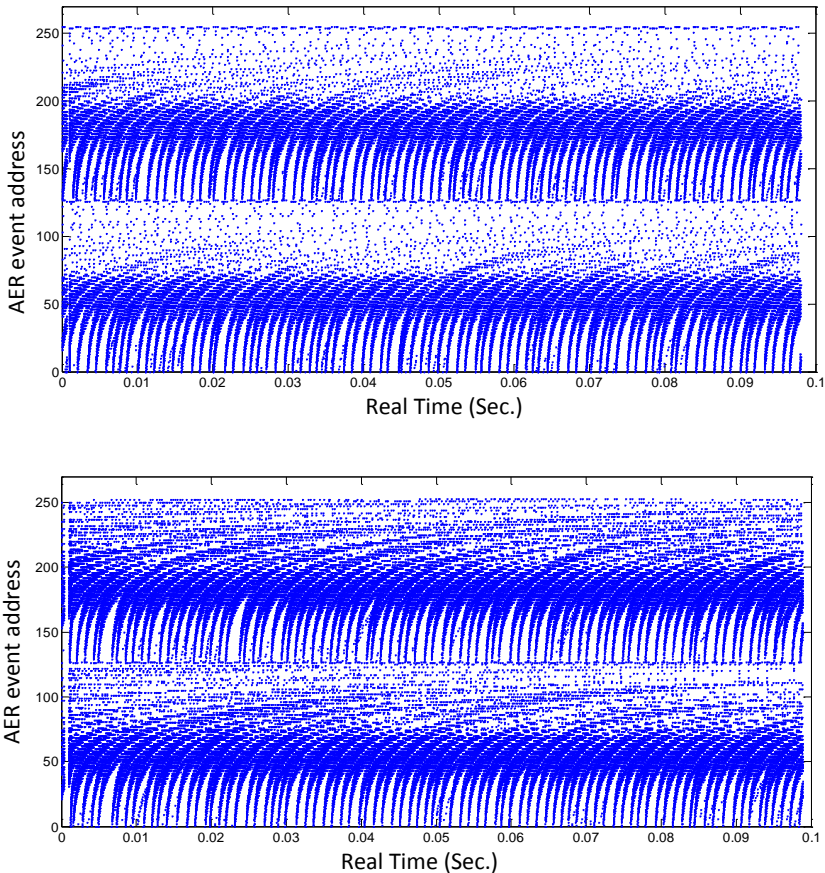


Figura 5.20. Cocleograma de un tono puro a 440Hz en NASv1 (arriba) y NASv2 (abajo)

Esta modificación del NASv1 la hemos realizado para obtener una ganancia similar en todos los canales respecto a la tasa de spikes que disparan, por lo tanto a continuación vamos a mostrar la gráfica donde se muestra cómo se comporta el banco de filtros de NASv2 ante el mismo experimento con el que calculamos el diagrama del bode del banco de filtros. El experimento consiste en calcular la tasa de eventos que se han generado para cada canal ante tonos puros senoidales de 2.5mW de potencia con frecuencias que barren desde 10Hz a 20kHz y duración un segundo. Estos datos se representan gráficamente en la Figura 5.21, donde arriba se representa la tasa de eventos del NAS izquierdo y abajo la tasa de eventos del NAS derecho. El eje  $x$  representa la frecuencia y el eje  $y$ , la tasa de eventos y las curvas

de colores distintos representan la tasa de eventos para cada canal. La línea que aparece en la figura arriba es la tasa de eventos total que se ha generado para el tono puro con la frecuencia que se indica en el eje  $x$ . En esta figura se observa como cada canal del NAS se comporta como un filtro de paso de banda y como efectivamente con las modificaciones realizadas en este apartado todas las bandas tienen la misma ganancia respecto la tasa total de spikes que se generan para esa frecuencia. Se sigue observando en las bandas de altas frecuencias que presentan un desplazamiento en las zonas de bajas frecuencias. Se aprecia que la tasa de eventos máxima para todos los canales ronda los 50kEvents/s. Esta es la tasa máxima de spikes que se puede obtener con esta configuración de los valores de los divisores de los filtros.

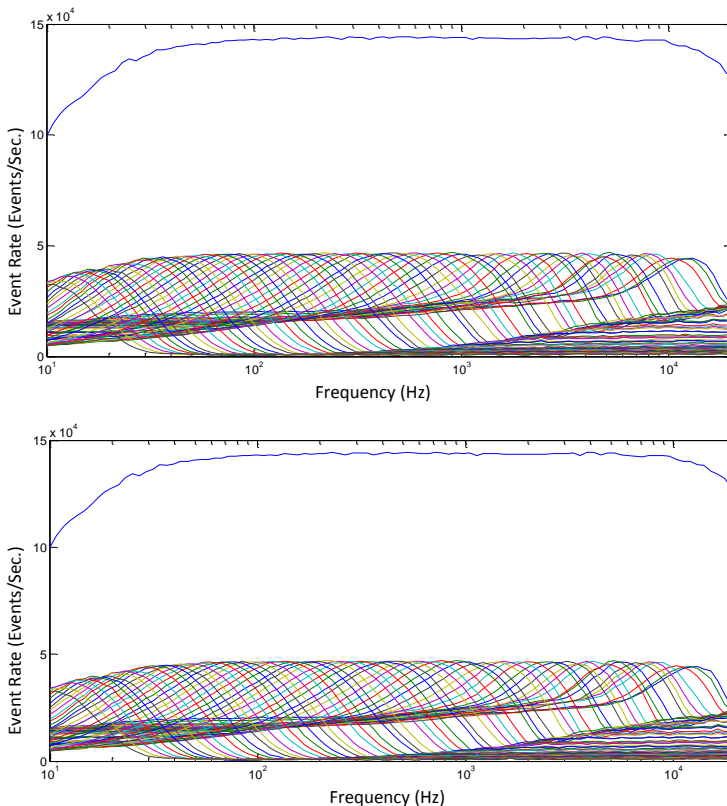


Figura 5.21. Diagrama de bode de NASv2 izquierdo (arriba) y derecho (abajo)

Esta misma gráfica se puede observar en la Figura 5.22 en su representación en superficie en la que el eje  $x$  es la frecuencia, el eje  $y$  las bandas y mediante diferentes colores se expresa la tasa de eventos de cada canal. Como en la Figura 5.21, se observa que el NAS izquierdo y derecho tienen la misma respuesta, sólo se va a representar en la Figura 5.22 la respuesta del NAS izquierdo. Esta figura muestra como cada canal tiene su máxima actividad ante una frecuencia determinada (la diagonal formada por las máximas tasas de eventos) y presentan una baja actividad fuera de su banda (azul). De esta figura se puede destacar como la diagonal tiene una tasa de eventos similar para todos los canales, en cambio para el NASv1 la tasa de eventos tiene un mayor rango de diferencia entre canales.

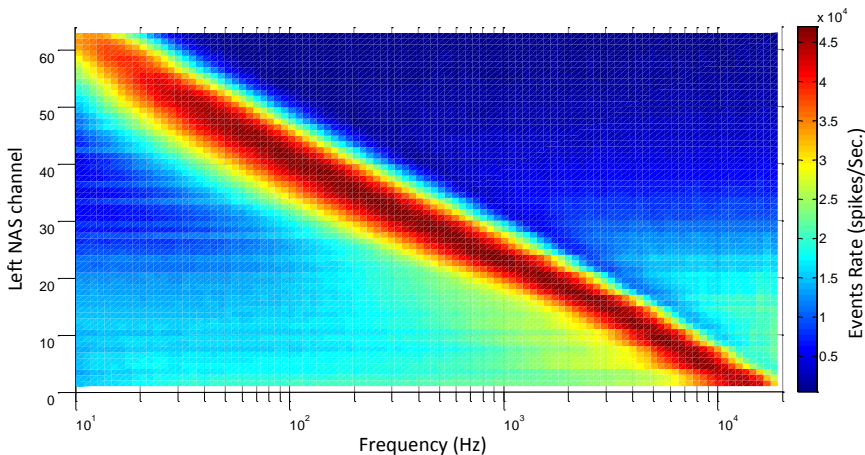


Figura 5.22. Tasa de eventos en representación en superficie

A partir de estas gráficas podemos obtener los valores de las ganancias máximas de cada banda en relación con la tasa de spikes generados inicialmente para el sonido a esa frecuencia. Dichos valores se observan en la Figura 5.23, donde el eje  $x$  es la frecuencia media de las bandas del banco de filtros y el eje  $y$  es la relación entre la ganancia máxima de dicho canal entre la tasa de eventos total que se generan para el tono puro a esa frecuencia. Esta misma medida es la que se calculó para NASv1 con el objetivo de normalizar la ganancia de los canales y se muestra

en la Figura 5.19. Sólo se muestra la gráfica del NAS izquierdo porque no hay una diferencia sustancial respecto al NAS derecho.

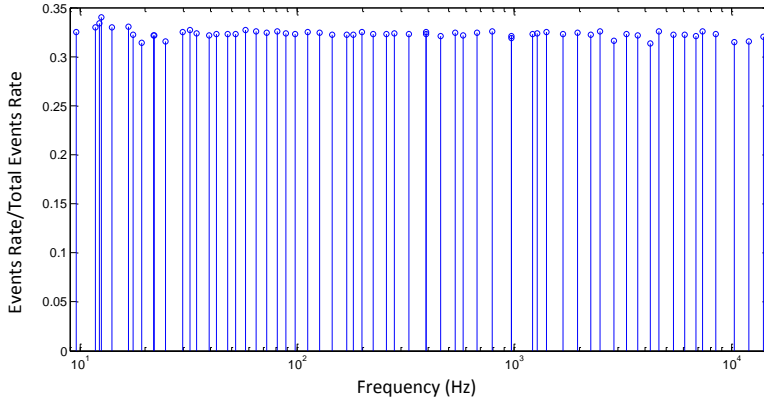


Figura 5.23. Ganancias máximas relativas de cada banda del NASv2 izquierdo

El factor de calidad del banco de filtros es el mismo que para NASv1, porque este factor depende directamente del valor de la distancia entre dos filtros paso de baja que forman el banco de filtros. El ancho de banda de cada canal del banco de filtros tampoco ha cambiado respecto a la implementación del NASv1, porque no hemos cambiado la frecuencia media de los canales, sólo la ganancia de cada canal.

El rango dinámico del NASv2 presenta cambios respecto al del NASv1 porque como se observaba en la comparación de la respuesta temporal entre NASv1 y NASv2 (ver Figura 5.20) la tasa de eventos ante el mismo estímulo se ha duplicado para el NASv2. El experimento que hemos realizado para obtener el rango dinámico es el mismo que se realizó para el NASv1, consiste en excitar al NAS con ruido blanco a diferentes niveles de volumen, analizando posteriormente la tasa de eventos de salida. La Figura 5.24 muestra los resultados experimentales. En el eje  $x$  se representa la potencia del ruido blanco, y el eje  $y$  representa la tasa de eventos total, es decir, para todos los canales, generados por el NAS. La salida del NAS implementado en circunstancia de ausencia de sonido, o una baja potencia de audio de entrada, tiene una actividad de salida de eventos AER menor a 8kE vents/Sec, el doble que para la implementación del NASv1. A un nivel de -60 dBW la actividad

AER se empieza a incrementar hasta 14dBW, donde la actividad se satura en con una tasa de eventos de 3.3MEvents/s. Por lo tanto, nuestro sistema tiene un rango dinámico de 74dBW en términos de volumen de audio de excitación. El rango dinámico tiene un decibelio menos porque el NASv2 tiene una mayor tasa de eventos, por lo tanto satura a menor volumen que el NASv1.

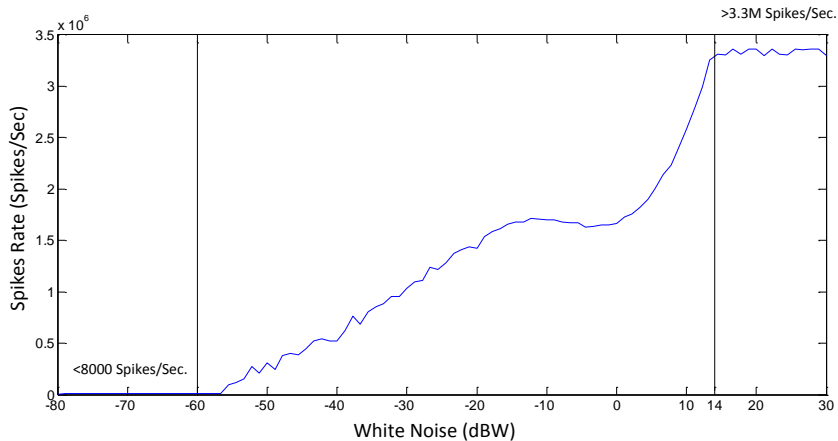


Figura 5.24. Rango dinámico del NASv2

En el siguiente apartado se muestra un estudio realizado para poder controlar la tasa de eventos de salida deseada mediante el parámetro de ganancia del generador de spikes Reverse bit-wise. También se puede controlar cambiando el valor del parámetro de los divisores de spikes.

## 5.6. Análisis de la tasa de eventos del NAS para diferentes configuraciones en el generador de spikes

El generador de spikes que hemos usado tiene un parámetro del que depende la tasa de spikes de la salida, o en otras palabras, la profundidad de modulación PFM. Por lo tanto, vamos a analizar el comportamiento del NAS para diferentes configuraciones del generador de spikes con el objetivo de poder automatizar la variación de la tasa de eventos de salida del NAS. Tal y como se explicó en este



capítulo en el apartado Sintonización del generador de spikes reverse bit-wise, la frecuencia de los spikes de salida cumplen la Ecuación 5.11. Por lo tanto, la tasa de spikes de salida depende inversamente del valor de  $genFD$ .

$$f(entrada)_{spikes} = \frac{F_{CLK}}{2^{n-1}(genFD + 1)} * entrada$$

Ecuación 5.11

Para las implementaciones previas del NAS hemos fijado este parámetro a 000Fh. Tenemos  $2^{16}$  valores distintos disponibles (porque son valores en binario natural) para el parámetro  $genFD$  y decidimos asignar el valor 001Eh, disminuyendo así al 50% la frecuencia de salida de los spikes.

Generamos la implementación del NAS con el generador de spikes modificado al valor 001Eh y le aplicamos la misma batería de tonos puros de entrada para obtener la tasa de eventos de cada canal para un barrido de frecuencias. En la Figura 5.25 se puede observar, mediante la curva de arriba de la gráfica, como la tasa de spikes de entrada se ha reducido a la mitad, tal y como se preveía según la Ecuación 5.11, en cambio la ganancia de todas las bandas no se ha reducido a la mitad. En las bandas cuya frecuencia media es menor a  $10^3$ Hz, la ganancia efectivamente se ha reducido a la mitad, en cambio, para frecuencias mayores sólo se ha reducido la tasa de spikes un 3%.

En la Figura 5.26 se puede observar esta modificación de las ganancias de las bandas de la siguiente implementación del NAS, el NASv3, en dicha figura el eje  $x$  representa la frecuencia media de cada banda, y el eje  $y$  la ganancia máxima de cada banda de ambas implementaciones del NAS. Con estos datos llegamos a la conclusión que para frecuencias menores a 1kHz podemos configurar nuestro NAS para que tenga una tasa de eventos variable automáticamente.

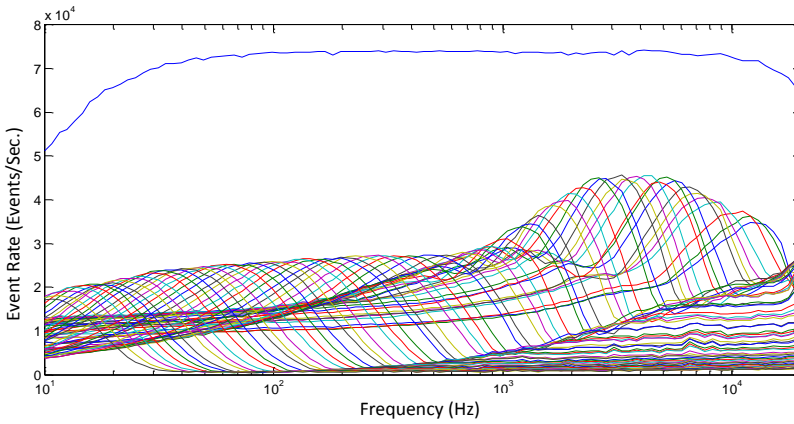


Figura 5.25. Diagrama de Bode de NAS con generador al 001Eh

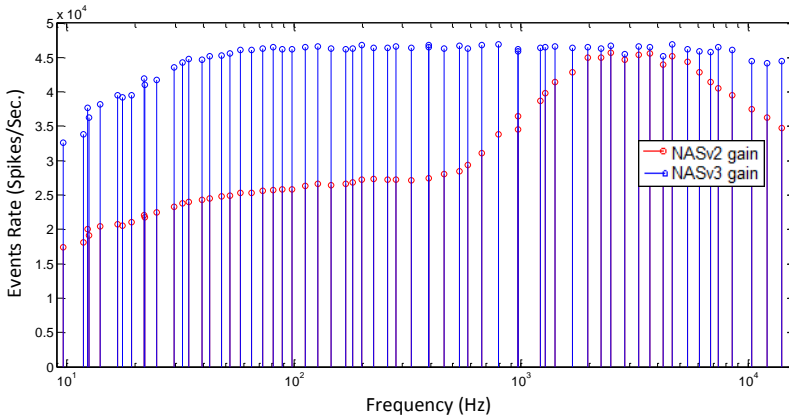


Figura 5.26. Ganancias máximas de cada banda para NASv2 y NASv3

A lo largo de este capítulo se han desarrollado tres sistemas que descomponen la señal de audio de entrada según sus componentes frecuenciales, los tres basados en la misma arquitectura. En la segunda parte de este trabajo se usarán estos sistemas para extraer la información relevante, mediante la descomposición del sonido en spikes que contiene la información del sonido según la Transformada de Fourier, para realizar el reconocimiento de sonidos mediante redes neuronales.

## 5.7. Comparativa del NAS con los sistemas previos

En la Tabla 5.4 se muestra un resumen de las características del NAS expuesto en este capítulo para compararlas con las implementaciones previas expuestas en el apartado Sistemas electrónicos auditivos bioinspirados. El rango dinámico del NAS es al menos 20dB mayor que de las implementaciones previas. El número de canales del NAS es igual o menor que otras implementaciones. El rango de frecuencias y la tasa de eventos son similares a la media de estas características de las implementaciones previas. El consumo de potencia es menor que la mayoría de trabajos previos.

Respecto las implementaciones de cócleas digitales, NAS necesita un número menor de slices y puede trabajar a frecuencias similares a las de las implementaciones previas.

Tabla 5.4. Resumen de las características del NAS

<b>Numero de Bandas</b>	64x2 (ajustable)
<b>Rango Frecuencial</b>	9,6Hz-15.54kHz (ajustable)
<b>Rango Dinámico</b>	75dB (NASv2: 74dB)
<b>Tasa de eventos</b>	3.3Mevent/s
<b>Consumo de potencia</b>	29,7mW
<b>Slices Requeridos</b>	11.141
<b>Frecuencia del reloj del sistema</b>	27MHz

## 6. Sistemas de reconocimiento automático de sonido

*“Lo que los ojos ven y los oídos oyen, la mente piensa”,  
Harry Houdini*

Los sistemas de reconocimiento de sonidos procesan el audio comúnmente en dos fases: extracción de parámetros característicos del sonido y clasificación usando las características obtenidas en la fase anterior. Para la fase de extracción de parámetros, hemos desarrollado un sistema de descomposición de audio en sus componentes frecuenciales inspirado en la cóclea biológica, expuesto en el capítulo 5. Los sistemas que hemos desarrollado para la fase de clasificación se exponen en el capítulo 7, pero previamente, en este capítulo, se exponen las diferentes metodologías para realizar la clasificación de sonidos y una revisión de los trabajos relacionados con reconocimiento de sonidos.

El reconocimiento automático de sonidos es una tarea inherentemente difícil debido a la variabilidad de las señales acústicas, así como su naturaleza dinámica. Si la señal se registra en condiciones favorables, se consiguen muy buenas prestaciones en el proceso de reconocimiento. Sin embargo, cuando el sistema funciona en situaciones reales se encuentra con condiciones adversas motivadas fundamentalmente por cambios en el entorno acústico (ruidos, reverberación y ecos) o eléctrico (ruido o distorsiones de la señal provocados por el micrófono o el canal de transmisión).

La bibliografía de clasificación de patrones recoge las tres siguientes aproximaciones: comparación de plantillas o patrones, modelo oculto de Markov y redes neuronales artificiales. Los primeros métodos de clasificación de sonidos se basan en la comparación de patrones, es decir, en comparar los patrones obtenidas

en la primera fase de extracción de información característica respecto a información de referencia. La programación dinámica es ineficaz cuando el número de clases aumenta, la semejanza entre clases distintas es elevada y el ambiente es ruidoso. Ante este problema, aparecieron las otras dos soluciones clásicas: *Modelos Ocultos de Markov (HMMs)* y las *Redes Neuronales Artificiales (RNAs)*.

Tanto las RNAs, (Guerrero-turrubiates et al. 2014), (Qian & Nian 2007), (Newton & Smith 2011), (Azarloo & Farokhi 2012) como los HMMs, (Rabiner 1989), (Barbancho et al. 2012), (Jackel et al. 2010) se han usado exitosamente para tareas de diferentes complejidades en reconocimiento de sonidos.

En este capítulo se exponen las características de los HMMs y de las RNAs. Respecto a las RNAs se expone el funcionamiento de dos modelos de redes neuronales que se diferencian en el tipo de información que procesan: pulsante o discretizada (modelos pulsante y modelos de tasas es como se suelen denominar). Se exponen las características e investigaciones de un tipo particular de red basada en la operación de convolución, denominada *red neuronal de convolución*. Para finalizar el capítulo, se hace un repaso de los sistemas de reconocimiento de sonidos publicados, donde la información cuantitativa servirá para comparar con los sistemas de reconocimiento expuestos en el capítulo 7.

## 6.1. Modelo estadístico: Modelo Oculto de Markov

Un *Modelo Oculto de Markov (HMM)* es una técnica de reconocimiento de plantillas o patrones, basado en un modelado estocástico o aleatorio de dichos patrones, permitiendo una mayor flexibilidad para representar secuencias de duración variable (Rabiner & Juang 1986), (Rabiner 1989).

Un HMM se define como una máquina de estados estocásticos finitos, donde la probabilidad de pasar al estado siguiente depende únicamente del estado actual (proceso de *Markov*), y asociado a cada transición entre estados se produce un vector de observaciones. Por tanto, cambia de estado, siguiendo una distribución

probabilística, una vez en cada instante de tiempo. Por cada instante de tiempo en que se produce un cambio de estado se genera una nueva salida, teniendo en cuenta también ciertas densidades de probabilidad. La particularidad de los HMMs, es que el paso de unos estados a otros de la cadena no es directamente observable, está oculto. Por tanto, se puede decir, que un *HMM* está compuesto de 2 procesos estocásticos, el oculto, correspondiente a las transiciones entre estados, y el observable o no oculto, correspondiente a la generación del vector de observaciones que se produce en cada estado, y que representa la plantilla a reconocer. Además, cada estado tiene asociada una distribución de probabilidad sobre los posibles símbolos de salida, con lo que la secuencia de símbolos generada por un *HMM* proporciona cierta información sobre la sucesión de estados. En la Figura 6.1 se muestra un sistema que puede ser descrito por un conjunto de 5 estados  $S=\{S_1, S_2, S_3, S_4, S_5\}$  y las correspondientes transiciones entre los estados con sus probabilidades  $a_{ij}$ , siendo  $i$  el estado origen de la transición y  $j$  el estado destino de la transición.

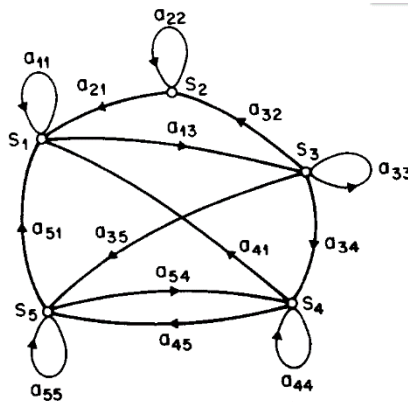


Figura 6.1. Cadena de un ejemplo de HMM de 5 estados (etiquetados  $S_1$  a  $S_5$ ) con las correspondientes transiciones entre los estados con sus probabilidades ( $a_{ij}$ , siendo  $i$  el estado de origen de la transición y  $j$  el estado destino de la transición). Imagen tomada de (Rabiner 1989).

Para el reconocimiento de sonidos, cada nodo representaría una unidad acústica, que pueden ser fonemas o sílabas en el reconocimiento del habla o un cierto periodo de tiempo para el reconocimiento de otros tipos de sonidos. En los sistemas

de reconocimiento del habla, la sucesión de unidades acústicas con su probabilidad de transición asociada, representará una palabra determinada. La evaluación de la secuencia de unidades acústicas respecto el objetivo se hace mediante la búsqueda de la secuencia de estados más probable que genere una secuencia de salida dada. Es frecuente usar el algoritmo de Viterbi para hacer un alineamiento no lineal entre los datos de entrada y las palabras de un diccionario cuyos términos son patrones estocásticos. Los alineamientos se establecen como probabilidades de que la secuencia analizada sea generada por los distintos Modelos de *Markov*.

Las excelentes prestaciones ofrecidas por los *HMMs* se deben, especialmente, a la forma en que modelan las deformaciones temporales de los patrones acústicos y a la existencia de algoritmos eficientes, como son el algoritmo de Baum-Welch (Baum & Eagon 1967) y el algoritmo de Viterbi (Jelinek 1976), que permiten estimar los parámetros de los modelos a partir de un conjunto de muestras de entrenamiento supervisado. Sin embargo, estos algoritmos presentan un importante inconveniente: la escasa capacidad discriminativa del conjunto de modelos resultante. Otra limitación de esta aproximación, reside en que se precisa realizar una serie de asunciones (casi siempre poco realistas) acerca de la naturaleza de las funciones de densidad de probabilidad que se quieren modelar.

## 6.2. Redes neuronales artificiales

Existen múltiples definiciones de Red Neuronal Artificial (RNA), entre la que cabe destacar: “Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico” (Kohonen 1988). Es decir, son sistemas de procesamiento de información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. Como en la naturaleza, las conexiones y valores asociados de dichas conexiones determinan la función de la red.

Las RNA se suelen usar en aplicaciones donde el análisis formal es difícil o imposible, como reconocimiento de patrones en entornos ruidosos, correlación de

todas las permutaciones posibles de una trama compleja y sistemas no lineales de identificación y control. Es decir, estos sistemas sirven para resolver problemas que no se pueden resolver matemáticamente y la búsqueda algorítmica, incluso dentro de un espacio de entrada relativamente pequeño, puede llegar a consumir mucho tiempo.

Los modelos neuronales se basan en el comportamiento de las neuronas biológicas, que consiste en generar y transmitir un pulso nervioso, denominado spike, si el potencial de la membrana de la neurona alcanza un umbral, debido a una serie de pulsos externos que han llegado previamente a la neurona. Después de disparar el spike, no es capaz de transmitir un nuevo spike hasta transcurrido un período de tiempo conocido como período refractario.

Hay distintos modelos neuronales inspirados en el comportamiento de la neurona biológica, dependiendo si la actividad neuronal se describe mediante pulsos eléctricos (*modelos neuronales pulsantes*) o la actividad neuronal se describe en términos de tasas de impulsos eléctricos (*modelos neuronales de tasas*). Por lo tanto, los modelos de tasas son los modelos neuronales tradicionales, en los que las entradas y salidas de las neuronas son números reales. Por lo tanto, respecto a el comportamiento de las neuronas biológica, estas redes tienen un nivel de abstracción mayor que los modelos neuronales pulsantes y se las denomina de tasas porque se considera que la información que procesan son las tasas de los pulsos de las neuronas biológicas.

En el capítulo 7, se exponen los sistemas de clasificación que hemos desarrollado para la segunda fase del sistema de reconocimiento de sonidos. El primero de ellos consiste en una red neuronal tradicional o de tasas, y otro consiste en una red neuronal de convolución. A continuación se exponen los conceptos generales de ambos tipos de redes neuronales.



### 6.2.1. Redes neuronales de información muestreada

Las redes neuronales están formadas por una serie de procesadores elementales, denominados neuronas artificiales, que constituyen dispositivos simples de cálculo que, bien a partir de un vector de entrada procedente del mundo exterior, bien a partir de estímulos recibidos de otras neuronas, proporcionan una respuesta única (salida). Resulta útil la caracterización de tres tipos de neuronas artificiales:

- Unidades de entrada, que son las neuronas que reciben las señales desde el entorno, provenientes de sensores o de otros sectores del sistema.
- Las neuronas de salida envían su señal directamente fuera del sistema una vez finalizado el tratamiento de la información (salidas de la red).
- Las neuronas ocultas reciben estímulos y emiten salidas dentro del sistema, sin mantener contacto con el exterior. En ellas se lleva a cabo el procesamiento básico de la información, estableciendo la representación interna de ésta.

En la Figura 6.2 se muestra un ejemplo de red neural que está compuesta por los tres tipos de neuronas,  $N_i$  representa la capa que contiene las neuronas de entrada,  $N_h$  representa la capa constituida por las neuronas ocultas y  $N_o$  representa a las neuronas de la capa de salida.

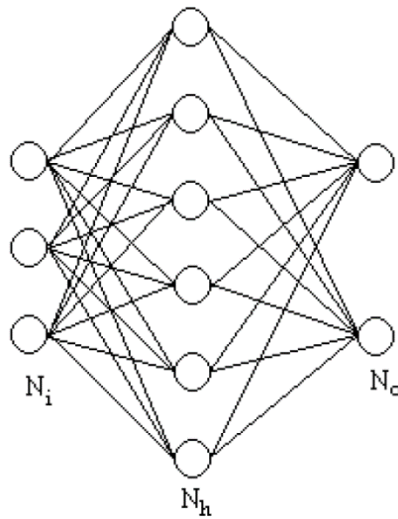


Figura 6.2. Ejemplo de perceptrón Multicapa con una única capa oculta ( $N_h$ )

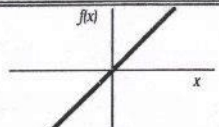
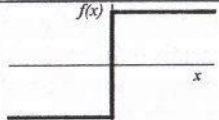
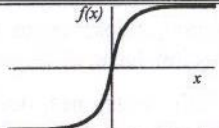
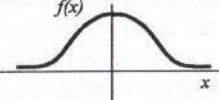
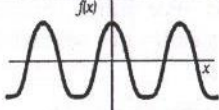
Tratando de mimetizar las características más relevantes de las neuronas biológicas, cada neurona artificial se caracteriza por los siguientes elementos básicos:

- Un conjunto de conexiones con valores asociados denominados *pesos* o sinapsis que determinan el comportamiento de la neurona, las cuales pueden ser excitadores, presentan un signo positivo (conexiones positivas), e inhibitoras presentan un signo negativo (conexiones negativas).
- Una *función de propagación*, que es el procedimiento a seguir para combinar los valores de entrada a una unidad y los pesos de las conexiones que llegan a esa unidad. De esta forma, la función de propagación permite obtener, a partir de las entradas y los pesos recibidos, el valor del potencial post-sináptico de la neurona en un momento determinado. La función más habitual es de tipo lineal y se basa en la suma ponderada de las entradas con los pesos sinápticos a ellas asociadas.
- Una *función de activación* que puede ser lineal o no lineal empleada para limitar la amplitud de la salida de la neurona. Generalmente, la

función de transferencia tiene carácter determinista, y en la mayor parte de los modelos es monótona creciente y continua respecto al nivel de excitación de la neurona, tal como se observa en los sistemas biológicos. A menudo es sigmoidea, y suele ser la misma para todas las unidades de la misma capa. En la Tabla 6.1 se muestran los principales tipos de función de activación.

- Una *ganancia* exterior que determina el umbral de activación de la neurona.
- La *señal de salida*, que en los casos de problemas de clasificación suele considerarse un conjunto finito de salidas (en muchos casos binarias), mientras que las tareas de ajuste de funciones suelen precisar salidas continuas de un determinado intervalo. El tipo de salida deseada determinará la función de activación que debe implementarse en las neuronas de la última capa de la red.

Tabla 6.1. Principales tipos de función de activación de las neuronas artificiales

	Función	Rango	Gráfica
<b>Identidad</b>	$y = x$	$[-\infty, +\infty]$	
<b>Escalón</b>	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
<b>Sigmoidea</b>	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
<b>Gaussiana</b>	$y = Ae^{-Bx^2}$	$[0, +1]$	
<b>Sinusoidal</b>	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Uno de los principales modelos de RNA es el *perceptrón*, que se organiza en dos capas de neuronas, una de entrada y otra de salida. Una de las neuronas de este modelo se muestra en la Figura 6.3, en la que se observan cada uno de los elementos de dicha neurona artificial. Las entradas  $x_i$  son los estímulos de las neuronas, cuyas conexiones con la neurona tienen asignados un peso de valor  $w_i$ . La función de propagación consiste en sumar todas las entradas multiplicadas por sus pesos correspondientes. La función de activación es una función escalón, cuya salida es -1 si el valor de la neurona artificial no ha alcanzado 0, y 1 en caso contrario. Otro aspecto común en las RNAs que se observa en la figura, es la existencia de una entrada especial que siempre tiene un valor fijo, +1, el cual puede ser usado como un valor fijo de referencia (ROSENBLATT 1958).

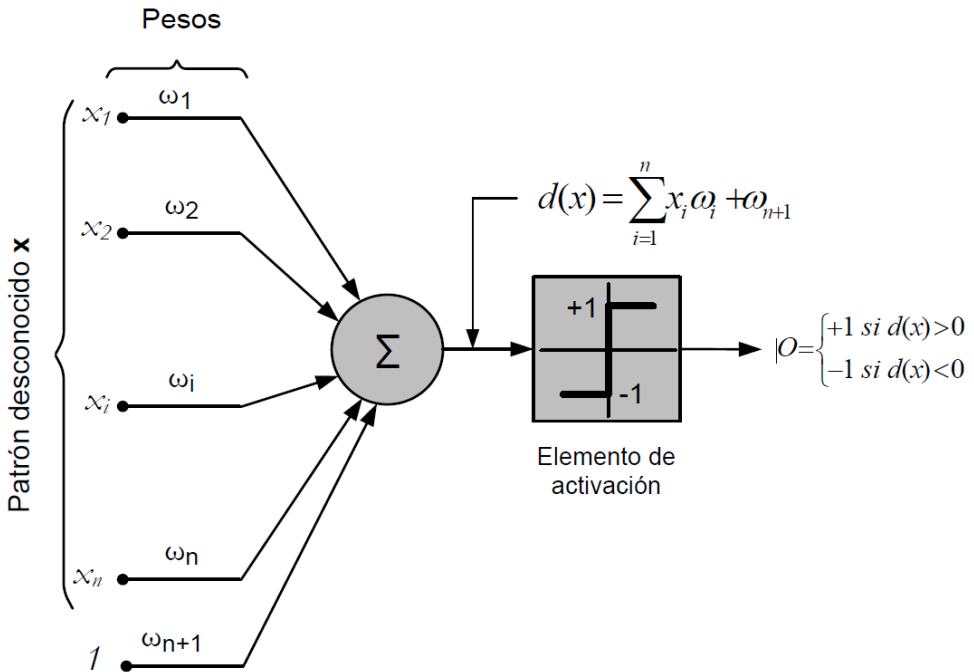


Figura 6.3. Esquema de neurona artificial del perceptrón

### Arquitectura de las redes neuronales artificiales

La topología o arquitectura de las RNAs hace referencia a la organización y disposición de las neuronas en la red formando capas de procesadores

interconectados entre sí a través de sinapsis unidireccionales. La arquitectura de la RNA depende de cuatro parámetros principales: el número de capas del sistema; el número de neuronas por capa; el grado de conectividad entre las neuronas; y el tipo de conexiones neuronales.

Las arquitecturas neuronales pueden clasificarse atendiendo a distintos criterios: Según su estructura en capas se pueden distinguir *redes monocapas*, compuestas por una única capa de neuronas y *redes multicapa*, cuyas neuronas se organizan en varias capas (de entrada, ocultas y de salida). Según el flujo de datos de la red se diferencian las redes *unidireccionales* o de *propagación hacia adelante* (*feed-forward*), en las que ninguna salida neuronal es entrada de unidades de la misma capa o de capas precedentes, y las redes de propagación hacia atrás (*feed-back*), en las que sí existen ese tipo de conexiones.

### Entrenamiento o aprendizaje de una red neuronal artificial

El aprendizaje o entrenamiento de una RNA es el proceso por el que una red neuronal crea, modifica o destruye sus conexiones (pesos) en respuesta a una información de entrada. En la mayoría de los modelos neuronales (redes off-line) existen dos modos de funcionamiento: el *modo de entrenamiento*, en el que se establecen los pesos de la red, y el *modo de ejecución*, en el que se usa la red para resolver el problema. No obstante, existen modelos neuronales en los que las fases de entrenamiento y ejecución coinciden (redes on-line), de forma que la red puede aprender y modificar sus conexiones durante el modo de ejecución, por lo que los pesos varían de forma dinámica cada vez que se presenta al sistema nueva información. Dependiendo si en el entrenamiento además de requerir los datos ejemplos, también requiere los resultados asociados, se diferencia entre aprendizaje *supervisado* o *no supervisado*. Se pueden distinguir tres tipos de aprendizaje supervisado: por corrección de errores, por refuerzo o estocástico.

El funcionamiento del *entrenamiento por corrección de errores* se basa en el ajuste de los pesos en función de la diferencia entre los valores deseados y los obtenidos por el sistema. El algoritmo de *retropropagación del error* (*backpropagation*) es un ejemplo de este tipo de entrenamiento.

En el *aprendizaje por refuerzo* se usa una señal de refuerzo que indica si la salida obtenida por la red se ajusta o no a la deseada y, en función de ello, se procede al ajuste de los pesos utilizando un mecanismo basado en probabilidades.

El *aprendizaje estocástico* se basa en la introducción de cambios aleatorios en los valores de los pesos de la red, evaluando su efecto a partir de la salida deseada y de una distribución de probabilidad.

### El perceptrón multicapa

Existen muchos modelos muy conocidos y de gran aplicación práctica como son: el asociador lineal, el perceptrón simple, las redes Adaline y Madaline, la red de Hopfield, las redes estocásticas, la red ART, el Perceptrón Multicapa (*Multi-Layer Perceptron, MLP*) y los mapas Auto-organizados de Características de Kohonen.

El modelo de RNA más usado en reconocimiento de sonidos es el MLP. Además, es el modelo más utilizado tanto para la resolución de problemas de clasificación como de regresión, al haber demostrado su condición de aproximador universal de funciones (Ripley 1994). Este modelo de red surgió como una solución para superar el problema detectado en el Perceptrón Simple, esto es, la imposibilidad de aprender clases de funciones no linealmente separables debido a no disponer de un mecanismo que permitiera obtener y actualizar los pesos intermedios del sistema. De esta forma, la arquitectura del MLP viene a coincidir con la del Perceptrón simple, con la diferencia de la inclusión de una o varias capas ocultas. Durante las décadas de los años 80 y 90, diversos grupos de investigadores propusieron teoremas similares que demostraban de forma matemática que un MLP de una única capa oculta constituía un aproximador universal de funciones (Funahashi 1989), (Hornik et al. 1989), (Hornik 1991), (Barron 1993). En la Figura 6.2, se observa la arquitectura de un MLP con una única capa oculta.

El algoritmo de *retropropagación de errores* o *backpropagation* (BP) constituye el modelo de aprendizaje de la red MLP más utilizado en la práctica, debido a su sencillez y eficacia para la resolución de problemas arbitrariamente complejos. Está basado en el método del gradiente descendente, que constituye a su

vez uno de los métodos de optimización de funciones multivariantes más antiguos conocidos. El nombre de este método de aprendizaje viene dado por las dos fases diferencias de su ejecución. En la fase de aprendizaje *hacia adelante* los patrones de entrada son presentados a la primera capa de la red, que propaga dicho estímulo a través de todas las capas posteriores hasta generar una salida del sistema. La fase de aprendizaje *hacia atrás* compara la salida generada por la red y la salida deseada, calcula el valor de error para cada neurona de la última capa del sistema y estos errores se transmiten a la capa intermedia ponderados según la participación en la salida original. El proceso se repite capa por capa hasta que todas las neuronas de la red hayan recibido un error que propague su aportación relativa a la salida final. En la Figura 6.4 se muestra de forma esquemática el proceso del modelo de aprendizaje de retropropagación de errores.

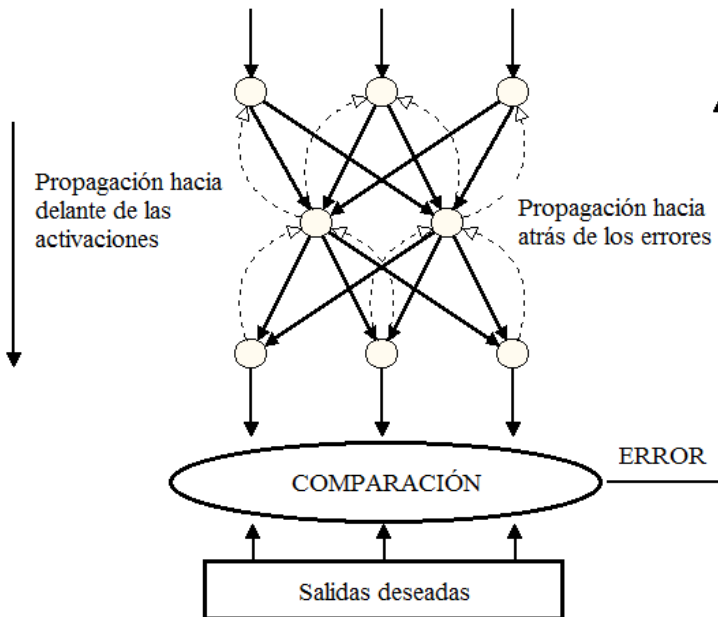


Figura 6.4. Esquema del modelo de retropropagación de errores

En general, se pueden utilizar todos los datos que estén disponibles para entrenar la red, aunque quizás no sea necesario utilizarlos todos. Con cierta

frecuencia, lo único que se necesita para entrenar con éxito una red es un pequeño subconjunto de los datos de entrenamiento de los que se dispone. Los datos restantes pueden emplearse para probar la red, con objeto de verificar que la red puede llevar a cabo la asociación deseada al utilizar vectores de entrada que nunca haya encontrado durante el entrenamiento. Hay que asegurarse que los datos de entrenamiento cubran todo el espacio de entradas esperado y no se puede entrenar por completo la red con vectores de una clase, pasando después a otra clase, porque la red *olvidará* el entrenamiento original (Freeman & Skapura 1991).

Como se ha comentado previamente el modelo MLP con tres capas constituía un aproximador universal de funciones. Por lo tanto, en general, con tres capas es suficiente para resolver un problema. Hay veces, sin embargo, en que parece que un problema es más rápido de resolver con más de una capa oculta porque aprende más deprisa.

El tamaño de la capa de entrada suele venir dictado por la naturaleza de la aplicación. A menudo, es posible determinar el número de nodos de salida decidiendo si se desean valores analógicos o valores binarios en las unidades de salida. Determinar el número de neuronas de la capa oculta no suele ser tan evidente. La idea principal consiste en utilizar el menor número posible de unidades de capa oculta, porque cada unidad supone una carga para la CPU durante las simulaciones y en sistema construido en hardware aumenta las conexiones entre neuronas.

### 6.2.2. Redes neuronales pulsantes

La diferencia entre las redes neuronales pulsantes y las de tasas es que las de tasa omiten la estructura pulsante de la salida de la señal. Ambos modelos tienen los mismos elementos: entradas con pesos asociados, función de propagación, función de activación y señal de salida, con la diferencia que en los modelos neuronales pulsantes las señales de entrada y las de salida están codificadas en pulsos. En el apartado 2.1 de este trabajo, se explican las distintas formas de codificar la información pulsante: codificación de la información en la frecuencia



de los spikes, siguiendo una modulación en frecuencia de pulsos (Westerman et al. 1997; Maass & Bishop 1999), codificación mediante el intervalo de tiempo entre los spikes (Giacomo Indiveri et al. 2006), o mediante el tiempo desde el primer spike (Thorpe et al. 2010).

Los distintos modelos neuronales pulsantes dependen de con qué detalle y precisión se modelan las características de las neuronas biológicas. Existen modelos neuronales, denominados *modelos basados en la conductancia*, que con gran exactitud reproducen las mediciones electrofisiológicas de las neuronas biológicas, pero debido a la complejidad intrínseca, estos modelos requieren de un alto costo de tiempo de ejecución. Por esta razón, modelos neuronales pulsantes más simples y basados en la fenomenología, se han hecho muy populares para estudios de codificación neuronal, memoria y dinámica de la red. A estos modelos se les *denomina modelos Umbral-Disparo (Threshold-Fire)* porque su objetivo es modelar la electrofisiología de las membranas neuronales, es decir, la generación de spikes cuando el potencial de la membrana sobrepasa un cierto umbral. El momento en que se sobrepasa el umbral se denomina *tiempo de disparo* (Maass & Bishop 1999).

El modelo neuronal Integra-y-Dispara (*Integrate-and-Fire, IF*) es probablemente el mejor ejemplo conocido de un modelo neuronal de Umbral-Disparo. Si el modelo reproduce la pérdida de potencial que sufren las membranas neuronales mientras no recibe ningún impulso, se denomina Integra-y-Dispara *con fugas (leaky integrate-and-fire)* y a la cantidad de potencial que se pierde *valor de fuga*. El modelo Integra-y-Dispara con fugas con un valor de fuga constante se describe mediante 5 operaciones básicas: 1) Integración sináptica, 2) integración con fugas, 3) umbral, 4) disparo de spike y 5) reset o valor de descanso. Estas operaciones se resumen mediante las ecuaciones que se muestran en la Figura 6.5. Para la neurona  $j$  en el instante de tiempo  $t$ , el potencial de la membrana  $V_j(t)$  es la suma del potencial de la membrana en el instante previo de tiempo  $V_j(t-1)$  y de las entradas activas  $x_i(t)$  multiplicadas por el valor (con signo) del peso  $s_i$  (Ecuación 1) de la Figura 6.5). La pérdida de potencial de la membrana basilar cuando no recibe impulsos se imita mediante la operación de la ecuación 2) de la Figura 6.5,

restando el valor  $\lambda$  al potencial de la membrana  $V_j(t)$ . En la membrana de las neuronas sólo se pierde potencial mientras no se reciben impulsos, en cambio, en este modelo, se resta el valor de fuga independientemente de la actividad sináptica. Luego se compara el potencial de la membrana  $V_j(t)$  con el umbral de la neurona  $\alpha_j$ ; si el potencial de la membrana ha alcanzado el valor umbral, emite un spike y se resetea el potencial de la membrana. En el caso típico, el valor de usado para resetear es 0.

SYNAPTIC INTEGRATION	
$V_j(t)$	$= V_j(t - 1) + \sum_{i=0}^{N-1} x_i(t) s_i$ (1)
LEAK INTEGRATION	
$V_j(t)$	$= V_j(t) - \lambda_j$ (2)
THRESHOLD, FIRE, RESET	
if	$V_j(t) \geq \alpha_j$ (3)
	Spike (4)
	$V_j(t) = R_j$ (5)
endif	(6)

Figura 6.5. Ecuaciones que describen la operación de las neuronas del modelo Integra-y-dispara con fugas. Imagen tomada de (Cassidy et al. 2013).

### 6.2.3. Redes Neuronales de Convolución

Un caso particular de red neuronal es la red neuronal de convolución (ConvNet). En este tipo de redes, la operación de las neuronas se basa en la operación de convolución.

En el espacio unidimensional, la *convolución* de dos funciones  $f(x)$  y  $g(x)$  se define en la Ecuación 6.1, donde  $\alpha$  es una variable de integración.

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha$$

Ecuación 6.1

Aplicando el *Teorema de la convolución*, la convolución discreta de las funciones  $f(x)$  y  $g(x)$  se define mediante la expresión de la Ecuación 6.2, siendo  $M$  la longitud de las series muestreadas. La función de convolución es una distribución discreta y periódica de longitud  $M$ , de forma que los valores  $x=0, 1, 2, \dots, M-1$  describen un periodo completo de  $f_e(x)*g_e(x)$ .

$$f_e(x) * g_e(x) = \sum_{m=0}^{M-1} f_e(m) * g_e(x - m)$$

Ecuación 6.2

Tal como se observa en las ecuaciones anteriores, la función de la convolución discreta es esencialmente la misma que la de la convolución continua. Las únicas diferencias son que los desplazamientos tienen lugar en forma de incrementos discretos correspondientes con la separación entre muestras, y que se realiza sumatorio en lugar de integración.

La operación de convolución se ha usado extensamente en el procesamiento de imágenes como filtro para diferentes procesamientos dependiendo de los valores del núcleo de convolución; es muy útil para detectar características en una imagen, porque puede detectar las características incluso con variaciones espaciales. Con el objetivo de aprovechar dicha ventaja de la operación de convolución, surgieron las ConvNet, en la que cada neurona aplica la operación de convolución. Estas redes combinan tres ideas que aseguran la precisión del sistema ante distorsiones, ruido y variaciones espaciales: campos receptivos locales, pesos compartidos y muestreo espacial. Otra ventaja de este tipo de red es lo apropiada que es para implementar en hardware.

El primer desarrollo software de ConvNet para el reconocimiento de patrones fue propuesta por (Fukushima 1980), y se ha seguido usando con éxito para sistemas de reconocimiento de caracteres (LeCun et al. 1990), (Fukushima & Wake

1991), (LeCun & Bengio 1995), (LeCun et al. 1998), (Neubauer 1998), detección de objetos (LeCun et al. 1998), (Ríos et al. 2012), y reconocimiento de caras (Neubauer 1998), (Lawrence et al. 1997), (Fasel 2002).

Existen muchos trabajos de implementación de procesadores de convolución y redes neuronales de convolución en hardware en el campo de la visión artificial, en los que vamos a diferenciar los que trabajan con números digitales y frames, y los que tratan la información de forma pulsante. Los sistemas tradicionales digitales de convolución basados en frames (Cope 2006), (Cope et al. 2005) multiplican el valor de cada pixel involucrado en la convolución por el valor asignado del núcleo de convolución y realizan el sumatorio de dichos resultados de multiplicación. Esta operación se representa analíticamente en la Ecuación 6.3, siendo  $O$  la salida al aplicar el núcleo de convolución al pixel  $ij$ -ésimo de la matriz  $I$ ,  $M$  las filas y  $N$  las columnas del núcleo de convolución.

$$O = \sum_{a=0}^M \sum_{b=0}^N (C(a, b) * I(a + i, b + j))$$

Ecuación 6.3

En la Figura 6.6 se muestra esquemáticamente un ejemplo de aplicación de la operación de convolución al primer cuadrante de la imagen  $I$ , para el núcleo de convolución  $C$  de tamaño  $2 \times 2$ , siendo  $I_{ij}$  los valores de los píxeles de la posición  $ij$  de la imagen y la matriz  $C$  los valores del núcleo de convolución.

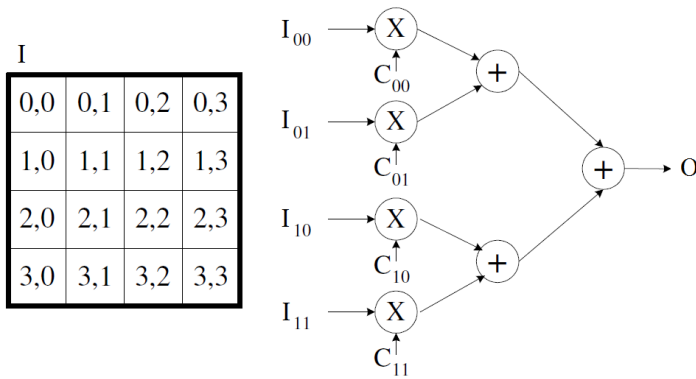


Figura 6.6. Esquema de funcionamiento de la operación de convolución para frames digitales

Otros desarrollos hardware que implementan redes de convolución 2-D para detección de caras son (Farrugia et al. 2008) y (Farabet et al. 2009)

Para implementar la operación de convolución con información pulsante, se suele actuar de forma que cada vez que llega un spike, se aplica la operación de convolución centrada en la posición de dicho spike y los spikes vecinos espacialmente. Por lo tanto, para implementar ConvNets en hardware se suele usar las neuronas Integra-y-Dispara, de forma que cada vez que llega un spike a una neurona de la red, se aplica la operación de convolución y si el resultado alcanza un valor umbral, se produce un spike y se resetea la neurona.

Existen varios trabajos que proponen hardware que implementa convoluciones 2-D con información pulsante como: (Serrano-Gotarredona et al. 2008), (Linares-Barranco et al. 2010), (Camuñas-Mesa et al. 2011), (Camuñas-Mesa et al. 2010), (Camuñas-Mesa et al. 2012).

Los resultados de estas investigaciones muestran que las redes neuronales de convolución son un buen ejemplo de tecnología bio-inspirada que obtiene muy buenos resultados en sistemas de reconocimiento de imágenes, incluso ante variaciones de la posición y en entornos ruidosos, como ocurre en los sensores de visión pulsante. Estos buenos resultados, nos motiva a usar este tipo de tecnología para realizar reconocimiento de sonidos, porque comparten igual que las imágenes,

se tienen grandes colecciones de ejemplos en los que la variación de los patrones a buscar es temporal y también de duración.

### 6.3. Sistemas de reconocimiento de sonidos

En el contexto del procesamiento de señales de audio y de voz, el *tono* se define como la frecuencia fundamental de la señal percibida por el sistema auditivo. Tal y como está detallado en el apartado 3.2.3., la estimación de esta característica es uno de los desarrollos necesarios para una gran variedad de aplicaciones como la transcripción de música, el reconocimiento de instrumentos, el reconocimiento de voz y cualquier otro procesamiento que se necesite conocer la frecuencia fundamental del sonido. Por lo tanto, en los últimos años se han realizado muchas investigaciones sobre la detección del tono en audio de música y voz, motivados por la búsqueda de técnicas eficientes y robustas con una complejidad computacional baja.

A continuación se exponen los trabajos relacionados con sistemas de detección del tono en sonidos musicales con el objetivo de realizar una comparativa de nuestros sistemas con los publicados previamente.

No existen muchos sistemas de detección del tono en sonidos musicales que usen cócleas artificiales para la fase de parametrización de la señal de voz. En cambio, sí hay muchos sistemas que usan cócleas artificiales para realizar tareas relacionadas con el reconocimiento de voz como (Kumar et al. 1998), (Miró-Amarante 2013), (Kim et al. 2009) y para realizar localización de la fuente emisora de sonidos (Chan et al. 2007), (Van Schaik et al. 2009).

El trabajo expuesto en (Gómez-Rodríguez et al. 2007) usa la primera versión de la cóclea artificial denominada *AER EAR* (Chan et al. 2006) para detectar el ritmo de los sonidos. Posteriormente, los trabajos expuestos en (Yu et al. 2009) y (Jackel et al. 2010) usan la cóclea analógica *AER EAR* expuesta en (Chan et al. 2007) para realizar reconocimientos entre dos clases de sonidos. El trabajo presentado en (Yu et al. 2009) discrimina entre dos sonidos de voz (arrullo y siseo) mediante el

histograma de los intervalos entre spikes, es decir, basándose en el contenido periódico. El trabajo expuesto en (Jackel et al. 2010) detecta entre una palmada o el sonido de un bombo, incluso con diferente volumen que el de las muestras de entrenamiento, usando en la fase de descodificación el HMM.

En el trabajo (O'Connor et al. 2013) se propone un sistema de fusión sensorial entre el sensor neuro-inspirado de visión, *Dynamic Vision Sensor* (P. Lichtsteiner et al. 2008) , la cóclea analógica presentada en (Liu et al. 2010) y, para la fase de clasificación, usa las redes *de profunda creencia* (*Deep belief network*). El experimento realizado en este trabajo consiste en reconocer los dígitos escritos a mano mediante la DVS, y, para mejorar la precisión del sistema, asocia un tono puro a cada dígito, de forma que cada dígito tiene su tono particular. Debido a que este trabajo no presenta los resultados de reconocimiento de sonido sin los datos del sensor de visión, no podemos compararlo con los sistemas de reconocimiento propuestos en este trabajo.

Otro trabajo que realiza reconocimiento de sonidos usando en la etapa de extracción de características un sistema bio-inspirado es (Newton & Smith 2011). Se inspira en la cóclea biológica y en la representación de la salida pulsante usando 100 filtros gammatone de paso de banda y neuronas Integra-y-Dispara. Para la etapa de clasificación usa una red neuronal recurrente (*Echo state network, ESN*), formada por 100 neuronas en la capa de entrada, 1500 neuronas en la capa oculta y 5 neuronas en la capa de salida. Realiza una clasificación entre 5 instrumentos obteniendo una precisión del 45%.

El trabajo expuesto en (Iwasa et al. 2007) propone un sistema de localización y reconocimiento mediante una red neuronal pulsante sintetizada en un FPGA. La extracción de características y generación de pulsos las realiza en software mediante filtros paso de banda.

Hay muchos desarrollos sobre reconocedores de sonidos basados en la detección del tono sin inspiración biológica. Se puede diferenciar entre los sistemas de reconocimiento de sonidos que estiman la frecuencia fundamental: (Penttinen et al. 2005), (Amado & Filho 2008), (Mahendra et al. 2009), (Zölzer et al. 2012); y

los que realizan el reconocimiento mediante clasificación: (Nielsen et al. 2006), (Zhu & Kankanhalli 2006), (Barbancho et al. 2012), (Pishdadian & Nelson 2013), (Guerrero-turrubiates et al. 2014).

En el trabajo expuesto en (Amado & Filho 2008) propone mejoras a dos algoritmos de estimación del tono, uno basado en la tasa de eventos que cruzan el cero (*Zero-cross rate ZCR*) y el otro basado en la función de autocorrelación (*Autocorrelation Function ACF*). El primer método tiene resultados imprecisos cuando la señal tiene componentes armónicas y oscilaciones alrededor de cero, como ocurre en los sonidos de las notas musicales y ambos métodos no tienen mucha precisión ante señales ruidosas.

En la referencia (Mahendra et al. 2009) se estima el tono de 13 notas musicales de la música clásica india, generadas por un instrumento musical típico de la música clásica india y por la voz de un hombre cantando, aplicando la función de autocorrelación. Este método también se usa en el trabajo (Penttinen et al. 2005) para estimar la frecuencia fundamental de los punteos en una guitarra.

Respecto a los sistemas que usan métodos de clasificación para reconocer aspectos de los sonidos musicales se puede destacar los resultados obtenidos por el sistema propuesto en (Barbancho et al. 2012), que obtiene un 95% de precisión para 48 clases de sonidos (4 tipos de acordes por 12 raíces de acordes) mediante múltiples estimadores de frecuencia y HMM. También destacamos el sistema de clasificación de notas musicales (Guerrero-turrubiates et al. 2014), que usando la transformada rápida de Fourier y el producto del espectro armónico (*Harmonic Product Spectrum*) en la etapa de extracción de características y el MLP para la etapa de clasificación, consigue una precisión del 97.5% para 12 notas usando 20 neuronas en la capa oculta y 10 neuronas en la siguiente capa. Otro sistema que usa el producto del espectro armónico para extraer el tono de sonidos con el objetivo de diferenciar entre sonidos musicales, de voz y ruido es el expuesto en (Nielsen et al. 2006).

En la referencia (Pishdadian & Nelson 2013) se expone una propuesta para transcribir melodías de un solo instrumento mediante el algoritmo K-Vecinos más



cercanos (*K-Nearest Neighbor*, *KNN*) para la etapa de clasificación. Para la fase de extracción de características calcula el espectrograma usando la ventana de Hamming. Consigue un porcentaje de aciertos del 91.45% para clasificar 26 notas de piano.

Otro trabajo relacionados con reconocimiento de sonidos mediante la detección del tono es (Zhu & Kankanhalli 2006), en el cual se propone un algoritmo para detectar si la pieza musical está en escala mayor o menor mediante la *Transformada Constante Q* (*Constant Q Transform*), método afín a la transformada de Fourier, para transformar los datos en el dominio de la frecuencia. El algoritmo alcanza un porcentaje de éxito del 92% para piezas musicales populares y el 81% para piezas de música clásica.

Una metodología que está en auge por su capacidad para resolver problemas de clasificación, las máquinas de vectores de soporte (Support Vector Machines), también se ha usado para tareas de transcripción de piezas musicales de un solo instrumento en los trabajos (Poliner & Ellis 2005) y (Poliner & Ellis 2006).

Para terminar este capítulo, en la tabla tal se muestra un resumen de las características de estos trabajos, con el objetivo de compararlos con el trabajo presentado en el siguiente capítulo.

Tabla 6.2. Resumen de las características de los sistemas de reconocimiento de sonidos expuestos

Referencia del sistema	Nº de clases	% de éxito	Coste
(Guerrero-turrubiates et al. 2014)	12	97.5%	30 neuronas (MLP)
(Newton & Smith 2011)	5	45%	1605 neuronas (ESN)
(Pishdadian & Nelson 2013)	26	91.45%	KNN
(Barbancho et al. 2012)	48	95%	330 estados ocultos (HMM)

## 7. Reconocimiento de sonidos

En este capítulo se describen los sistemas de reconocimiento de sonidos desarrollados a lo largo de este trabajo, los cuales identifican el sonido que se está produciendo a partir de la información filtrada por los NAS (capítulo 5). El NAS es un sistema de descomposición del sonido en componentes frecuenciales, de manera que decidimos reconocer sonidos cuyas frecuencias fundamentales sean característica del sonido: tonos puros, notas musicales y el sonido que genera un motor dependiendo de la potencia.

La estimación de la frecuencia fundamental de los sonidos es necesaria para una amplia variedad de aplicaciones; en el campo musical se usa para la transcripción y para reconocer instrumentos; en el campo del reconocimiento de voz, tanto para reconocer el hablante como para el reconocimiento del habla. El reconocimiento del sonido del motor, se puede aplicar en sistemas de control de calidad mediante el sonido que se genera ante diferentes funcionamientos del producto que se esté auditando.

Todo sistema de reconocimiento de sonidos tiene dos fases, la fase de filtrado, en la que se detectan las características relevantes respecto al reconocimiento que se quiere hacer y la segunda fase de identificación. En este capítulo presentamos los sistemas que hemos desarrollado para resolver la segunda fase del reconocimiento de sonidos. Se plantean 3 tipos de soluciones para resolver esta fase: método estadístico, redes neuronales de tasas, y por último, redes neuronales pulsantes. Las razones por la que nos decantamos por usar redes neuronales pulsantes es que, además de ser adecuadas para el procesamiento temporal de series de datos como son las señales de audio, se pueden implementar en hardware sin necesidad de componentes complejos de procesado. Además, este tipo de

técnicas son robustas ante el ruido y se pueden integrar en sistemas de reconocimiento empotrados en tiempo real.

Al principio de este capítulo se expone el sistema de identificación de la frecuencia de un motor en función del sonido que emite. En segundo lugar, se expone la red neuronal de tasas planteada como primera aproximación al reconocimiento de tonos puros y notas musicales. Por último, se expone la red pulsante de convolución desarrollada en VHDL para reconocer tonos puros y notas musicales.

### 7.1. Sistema de identificación de la frecuencia de un motor

A continuación se expone el primer sistema de identificación de sonidos en tiempo real que se ha desarrollado en el contexto del trabajo. Consiste en identificar la frecuencia de un motor según el sonido que genera. Este sistema de identificación se engloba en un experimento que consiste en determinar la frecuencia de rotación de un motor DC mediante el sonido que genera y mediante una retina DVS128 (*Dynamic Vision Sensor*) (Lichtsteiner et al. 2006) que captura el movimiento de un disco conectado al motor.

En este apartado, nos vamos a centrar en la parte de identificación de la frecuencia del motor en función del sonido que genera. El trabajo completo está disponible en (Rios-Navarro et al. 2014).

#### 7.1.1. Escenario del experimento para el reconocimiento del sonido generado por un motor

En la Figura 7.1 se muestra un diagrama de bloques representativo del escenario del experimento. A la izquierda está representado el motor DC con el disco, en él que está situado un encoder<sup>19</sup> óptico para realizar la medida con la que vamos a comparar nuestro sistema de identificación de la velocidad del motor, basado en las

---

<sup>19</sup> Un encoder es un sensor electro-opto-mecánico que unido a un eje, proporciona información de la posición angular.

capturas visuales mediante la retina artificial DVS128 y mediante las capturas realizadas por el NASv1. En el centro está representada la placa AER SWITCH MERGE que gestiona la información que se obtiene de ambos sensores. Esta placa es comercial, XEM6010 Opal Kelly (Opal-Kelly 2015), y consta de una FPGA Spartan 6, en la cual, mediante un módulo VHDL, se gestiona la mezcla de los eventos de los dos sensores mediante el bit más significativo y monitoriza los eventos para enviarlos con su marca de tiempo mediante la interfaz USB 2.0 (Rios-Navarro et al. 2015). Por último, se encuentra el PC, donde está desarrollado el sistema de identificación de la frecuencia del motor en función de los eventos recibidos. Este sistema de identificación se ha desarrollado en JAVA y está integrado en la herramienta software jAER (JAER 2015). Como se comentó previamente, en este trabajo se presenta la parte de identificación mediante el sonido.

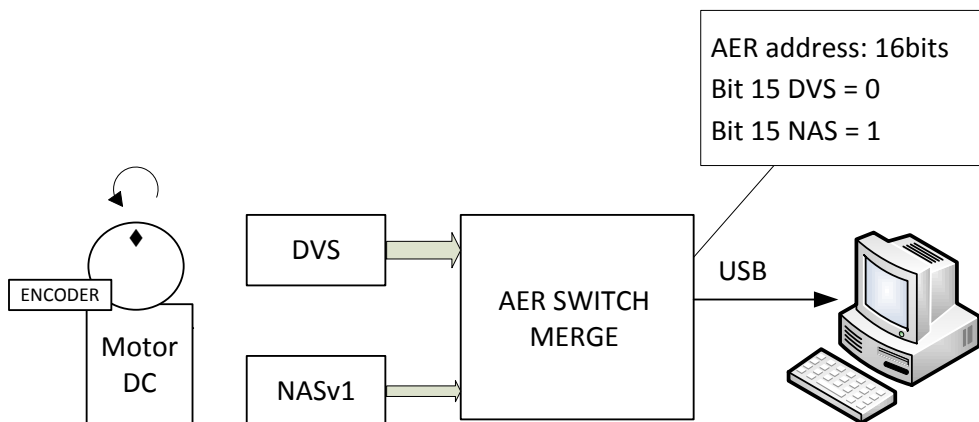


Figura 7.1. Diagrama de bloques representativo del sistema de estimación de la frecuencia de un motor DC mediante los sensores NASv1 y DVS128

En la Figura 7.2 se observa una fotografía del escenario real de este experimento, en el que además de los componentes nombrados previamente, se observa el micrófono, la mezcladora de sonidos XENYX QX1002USB (Behringer 2015) y la placa que contiene el NASv1 (ML507 Virtex 5 evaluation board).

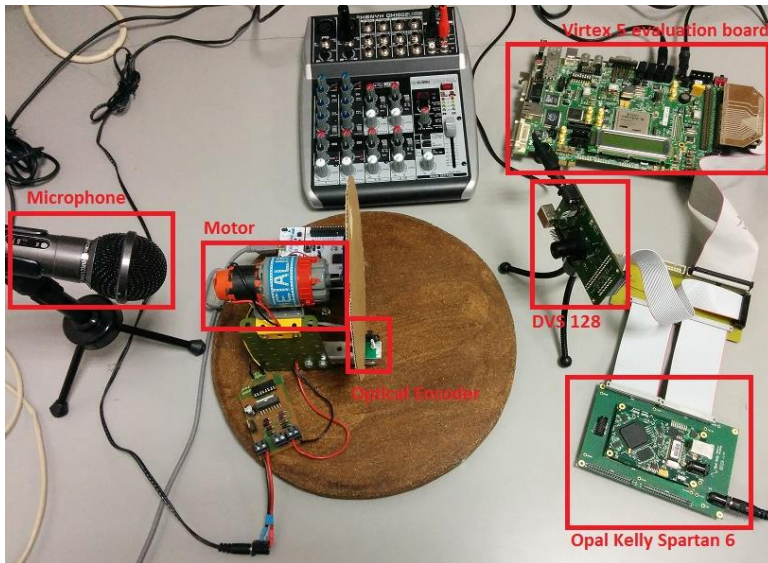


Figura 7.2. Escenario del experimento de identificación de la frecuencia de un motor DC mediante NAS y DVS128

### 7.1.2. Sistema de identificación de la frecuencia del motor

En el contexto que se ha explicado previamente, se ha desarrollado un filtro en JAVA integrado en la herramienta de gestión de eventos AER, jAER (JAER 2015), cuyo objetivo es determinar la frecuencia del motor en función de los datos recibidos. Los datos recibidos están formados por las direcciones de los eventos de salida del NASv1 y de la DVS junto con una marca de tiempo que indica en el instante en que se produjo el evento, impuesta por la funcionalidad del módulo VHDL sintetizado en la FPGA de la placa Opal Kelly (Rios-Navarro et al. 2015).

Mediante la experiencia, observamos que el sonido que genera el motor va cambiando conforme la frecuencia cambia. La modificación se produce en dos aspectos característicos del sonido: la frecuencia y la intensidad. Para buscar la relación entre la intensidad del sonido y la frecuencia del motor, se generaron nueve muestras representativas de esta relación con el motor a frecuencias del intervalo [180, 2000] rpm y se calculó la tasa de eventos que produce NASv1 para cada una de ellas. Se obtuvo la tasa de eventos generada por el NASv1 izquierdo y

derecho, y a partir de esos datos, se obtuvo las curvas de regresión mostradas en la Figura 7.3, que representan la relación entre la tasa de eventos de la salida del NAS y la frecuencia del motor.

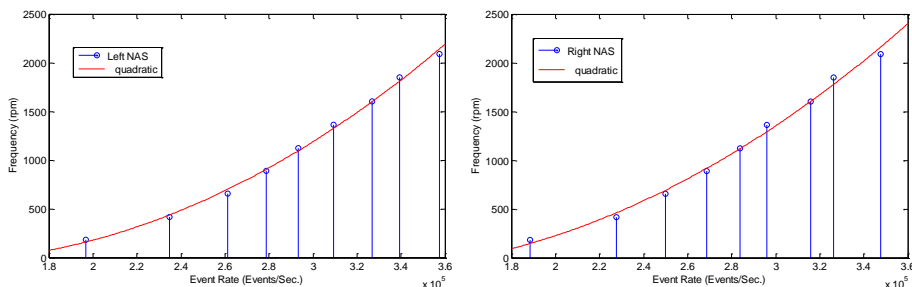


Figura 7.3. Relación entre la frecuencia del motor DC y la tasa de eventos generada por NAS izquierdo y derecho, respectivamente

La Ecuación 7.1 es la curva de regresión obtenida a partir de la tasa de eventos del NAS izquierdo y la Ecuación 7.2 respecto el NAS derecho.

$$y = 4.09e - 8x^2 - 1.03e - 2x + 611.66$$

$$\text{Norm of residuals} = 85.64$$

Ecuación 7.1.

$$y = 3.85e - 8x^2 - 7.99e - 3x + 285.46$$

$$\text{Norm of residuals} = 93.736$$

Ecuación 7.2.

Una vez que se tienen las curvas de regresión, se desarrolló en el entorno de la aplicación jAER un filtro que recibe paquetes de una capacidad máxima de 4096 pares eventos-marca de tiempo, con los que se calcula la tasa de eventos total de canal izquierdo y derecho como entrada de las curvas de regresión calculadas durante el entrenamiento y la media de ellas se corresponde a la frecuencia del motor estimada. Como las revoluciones del motor no se pueden cambiar de forma brusca, planteamos un método de programación de alto nivel para mejorar la estimación de las revoluciones por minuto. Este método consiste en almacenar los

últimos 10 valores calculados en un vector ordenado para tomar como resultado el valor medio. El tiempo de muestreo depende de la tasa de eventos porque los paquetes tienen un tamaño de 4096 eventos. Como se observa en las gráficas, la tasa de eventos cambia en función de las revoluciones, cuyo valor medio es de 0.27Meventos/s, por lo tanto, de media, cada 15ms se realiza una estimación.

Para calcular el error medio producido en la estimación, se almacena el valor estimado junto con el instante de tiempo en el que se ha generado. A la vez, gracias al encoder, se almacena en otro archivo la frecuencia de giro obtenida por el encoder junto con el instante de tiempo en el que se ha generado. Los instantes de tiempo almacenados tienen una resolución de un segundo.

### 7.1.3. Experimento y sus resultados

La ejecución del experimento consiste en hacer girar el disco conectado al motor en frecuencias creciente durante 1 minuto, empezando en 180rpm y terminando en 2000rpm, abarcando todas las frecuencias intermedias. Mientras que el disco está girando, se almacenan los dos ficheros de texto, uno de ellos con los valores determinados por el encoder en cada instante de tiempo y otro con los valores estimados por el sistema de identificación en cada instante de tiempo. Se calcula el error relativo del experimento mediante la Ecuación 7.3, siendo  $rpm_{jAER_i}$  los valores obtenidos mediante el filtro implementado en JAER,  $rpm_{encoder_i}$  los valores determinados por el encoder en el mismo instante de tiempo (representado por  $i$ ) y  $N$  el número de estimaciones que se han hecho. El error relativo resultante que obtenemos de dicha ecuación con los valores obtenidos en la ejecución del experimento explicado es **7.6%**.

$$Error_{Relative} = \frac{\sum_{i=1}^N \frac{|rpm_{jAER_i} - rpm_{encoder_i}|}{rpm_{encoder_i}}}{N}$$

Ecuación 7.3.

Este sistema depende del volumen, es decir, depende de la distancia que se encuentre el micrófono del motor, por lo tanto, este sistema necesita un micrófono

direccional y ser entrenado si se desea mover o modificar los elementos del escenario.

## 7.2. Sistema de clasificación de sonidos mediante redes neuronales de tasas

Los objetivos de este apartado son: comprobar cómo se comportan nuestro NAS como sistema de filtrado para sistemas de reconocimiento basados en redes neuronales tradicionales, comparar estos resultados con la ConvNet propuesta en el siguiente apartado y mejorar el manejo de la información codificada en spikes que se obtiene mediante el NAS.

Para una primera aproximación a sistemas de reconocimiento mediante patrones hemos decidido usar el *toolbox* de Matlab *Neural Network Toolbox*, el cual proporciona herramientas para el diseño, implementación, visualización y simulación de redes neuronales. Estas herramientas están disponibles en dos formatos: mediante funciones y mediante interfaces gráficas, y nos van a permitir resolver los cuatro tipos de problemas siguientes:

- *Function fitting*: en este tipo de problemas se usa una red neuronal para mapear un conjunto de datos numéricos de entrada con un conjunto de objetivos.
- *Pattern recognition*: en este tipo de problemas se usa una red neuronal para clasificar los datos de entradas en un conjunto de categorías metas.
- *Data clustering*: en este tipo de problemas se usa una red neuronal para agrupar datos por sus similitudes.
- *Time series analysis*: para problemas en los que necesitamos la predicción mediante valores tomados en uno o más instantes de tiempo previos se usan para predecir valores futuros.

Nuestro objetivo consiste en identificar dos tipos de sonidos: tonos puros y notas musicales de un piano electrónico. Los datos que tenemos sobre estos sonidos son los que se han obtenido de excitar el NAS (capítulo 5). Estos datos son



la secuencia de eventos AER generados por cada banda del NAS. *Pattern Recognition* de Matlab propone una red basada en el perceptrón multicapa con una capa oculta de propagación hacia atrás, que, como se explicó en el capítulo anterior, es una buena solución para el reconocimiento de patrones en entornos ruidosos.

La información que necesita este tipo de red neuronal para su entrenamiento es una pareja de matrices, una de las cuales tiene las muestras y la otra los resultados. La matriz de muestras tiene tantas filas como características identificativas de las muestras y tantas columnas como muestras se tengan. La matriz con los objetivos tiene tantas filas como clases entre las que se quiere identificar y tantas columnas como muestras haya. Cada columna de la matriz de objetivo se corresponde con una columna de la matriz de muestras.

Las redes neuronales de tasas necesitan información discretizada en el tiempo y organizada en características, por lo tanto nos planteamos cómo discretizar la secuencia de spikes y qué características podemos obtener de estas secuencias de spikes. Las características que tenemos de cada sonido están relacionadas con la tasa de spikes que genera cada banda, por lo tanto, planteamos que los sonidos tiene 64 características, y que cada una de ellas se corresponde con la tasa de eventos de cada canal en un periodo de tiempo. El periodo de tiempo lo determinamos basándonos en la duración menor de tiempo de un sonido audible por el oído humano, 10ms, pero como esta duración mínima del sonido audible no se cumple para todo el rango de frecuencias, decidimos marcar 20ms, que tal como se expuso en el apartado 3.2 Psicoacústica, es el periodo característica de integración en los procesamientos de audio. De esta forma, se consigue unos patrones que contienen información sobre el tono y la intensidad del sonido original.

Queremos desarrollar un sistema de reconocimiento que sea independiente del volumen y como la tasa de eventos está relacionada con el volumen, tal como está expuesto en el capítulo 5, en lugar de usar la tasa de eventos de cada banda como los valores de las muestras vamos a normalizar estos valores entre [0,1], dividiendo la tasa de eventos de cada banda entre la tasa de eventos máxima, es decir, entre la

tasa de eventos de la banda que genera el mayor número de spikes por segundo. Por lo tanto, este experimento tiene cuatro procesos diferenciados: 1) la excitación del NASv1 con los tonos puros para capturar las muestras 2) captura, mediante la USBAERmini2, de los eventos y *time stamps* generados por el NAS ante cada tono puros 3) cálculo de las matrices de entrada de la red neuronal como se ha explicado previamente 4) construcción, entrenamiento y obtención de resultados de la red neuronal. En la Figura 7.4 se muestran los componentes que intervienen en cada uno de estos procesos.

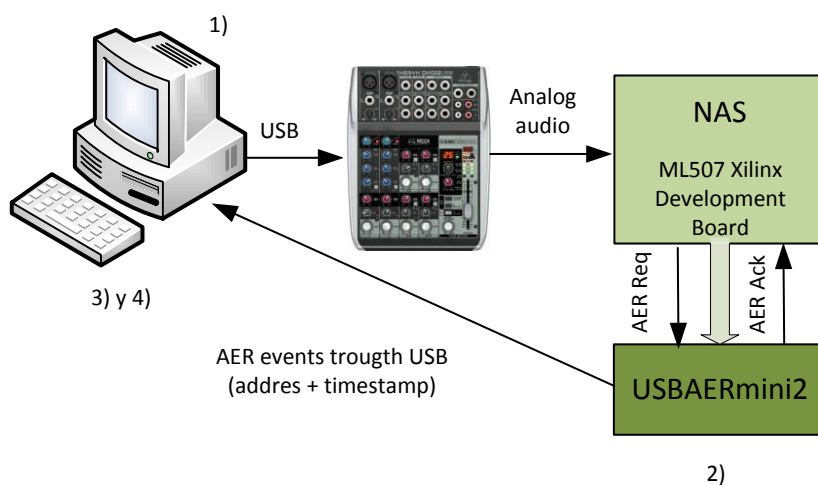


Figura 7.4. Componentes que intervienen en los experimentos de redes neuronales de tasas

El número de muestras para construir y probar la red de propagación hacia atrás tiene que ser *suficiente* y *adecuado*. Estos dos conceptos no se pueden cuantificar fácilmente, pero se deben seguir unas directrices. En general, se pueden usar todos los datos que estén disponibles, pero se debe dejar un subconjunto para verificar el comportamiento de la red con datos que no se haya encontrado la red durante el entrenamiento. Si se está entrenando la red para que funcione en un entorno ruidoso, hay que incluir datos con ruido en el conjunto de entrenamiento. Hay que asegurarse que los datos de entrenamiento cubran todo el espacio de entradas esperado. Durante el entrenamiento, hay que seleccionar aleatoriamente los pares

de vectores, porque si se entrena la red con todos los datos de una clase, pasando luego a otra, la red *olvidará* el entrenamiento original (Freeman & Skapura 1993).

Por otro lado, el número de nodos en la capa oculta es otra cuestión que no tiene una solución analítica. Como se expuso en el capítulo 6, en general, con 3 capas es suficiente para resolver cualquier problema, pero puede ocurrir que con más capas sea más rápida de entrenar la red. El número de nodos en la capa oculta se intenta minimizar, por lo tanto, si converge con un número de neuronas, se debe probar con un número inferior de nodos ocultos y determinar un tamaño final basándose en el rendimiento global del sistema (Freeman & Skapura 1993).

### 7.2.1. Clasificación de tonos puros mediante red neuronal MLP

El experimento que se expone en este apartado consiste en la clasificación de tonos puros por su frecuencia característica. Las frecuencias seleccionadas para este experimento se corresponden con las frecuencias fundamentales de notas musicales porque el siguiente experimento consiste en el reconocimiento de notas. Se han seleccionado las frecuencias que se observan en la Tabla 7.1, frecuencias relacionadas con las notas musicales correspondientes.

Tabla 7.1. Frecuencias características de los tonos puros a identificar

Nota	C3	F3	C4	F4	C5	F5	C6	F6
Freq.(Hz)	130.81	174.61	261.63	349.23	523,25	698,45	1046,5	1896,91

En la fase de obtención de los eventos que van a ser la entrada de la red de clasificación, se excita el NASv1 mediante tonos puros generados mediante una señal senoidal con cada una de las frecuencias indicadas en la tabla, potencia de 2.5mW y duración un segundo, para obtener la secuencia de eventos AER de salida. Una vez que tenemos la secuencia de spikes, obtenemos las muestras de esa clase mediante un script que se ha desarrollado en Matlab, que consiste en obtener la tasa de eventos de cada banda para un periodo de tiempo determinado que hemos marcado en 20ms. De esta formase obtienen 50 muestras de 128 características cada una, debido a los 64 canales del canal izquierdo y del derecho. Como

introducimos el mismo sonido por el canal derecho e izquierdo y mientras más características, más tiempo necesita la red neuronal para entrenarse e identificar, en este caso, no vamos a usar la salida de las bandas 64 a 127, porque como se comentó anteriormente la respuesta del NAS derecho e izquierdo son muy similares ante el mismo sonido.

Una vez que tenemos los datos de entrenamiento preparados, el siguiente paso es crear la red neuronal. Vamos a plantear una red neuronal realimentada de dos capas, una de ellas oculta y la otra la capa de salida. Las neuronas que forman las capas tienen función de activación sigmooidal. La capa de salida tiene 8 neuronas ya que es el número de clases que queremos identificar del sonido de entrada, y el número de neuronas de la capa oculta lo hemos determinado experimentalmente: para obtener el 96.2% de porcentaje de acierto hacen falta mínimo **10** neuronas en la capa oculta.

El entrenamiento de dicha red se hace mediante el método de retropropagación de errores en conjunción con el método del gradiente conjugado escalado, usando las muestras que hemos generado en la primera fase. Tenemos disponibles 50 entradas de audio de 20ms segundos de duración para cada clase, en total 400, las que se dividen de forma aleatoria en tres grupos, el 70% al grupo de entrenamiento, el 15% al grupo de validación y el 15% restante al grupo de pruebas. En la Figura 7.5 se observan la arquitectura de la red implementada y los datos relacionados con el entrenamiento realizado. Los resultados obtenidos con dicho entrenamiento se observan en la Tabla 7.2.

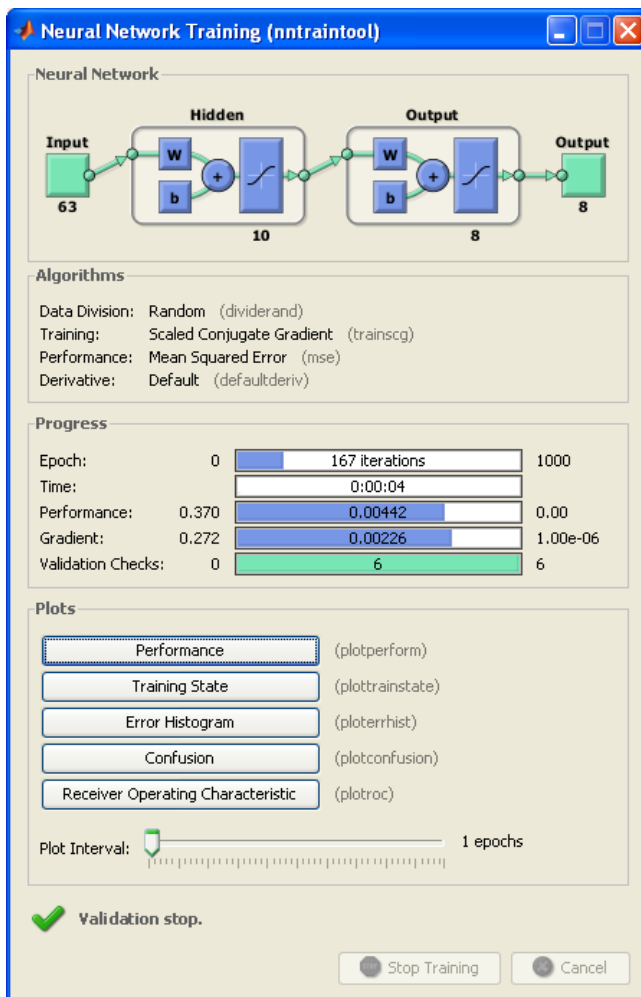


Figura 7.5. Datos del entrenamiento de la red neuronal tradicional

Tabla 7.2. Resultados de la red neuronal de 10 neuronas en la capa oculta para reconocer 8 tonos puros usando el NASv1

	Frecuencias de tono puro (Hz)								Total
	130,81	174,61	261,62	349,22	523,25	698,45	1046,5	1896,91	
%de éxito	100	91,73	84,61	100	95,7	100	100	100	96,5

Para comprobar si efectivamente la red neuronal es independiente del volumen del sonido de entrada, se repite el experimento, pero esta vez, las muestras están formadas por tonos puros a una potencia de 1.2mW.

Los resultados obtenidos para la misma arquitectura de red neuronal no consiguen tan buenos resultados, siendo el error general de 17.4% y es necesario duplicar el número de neuronas internas para obtener el mismo porcentaje de muestras correctamente clasificadas. Como se expone en el capítulo 5, la salida de los canales los filtros que forman el NASv1 tienen diferentes ganancias, siendo mayores en los canales de altas frecuencias, por lo tanto, al cambiar el volumen, cambian la tasa de spikes de los canales de altas frecuencias respecto a los de bajas frecuencias.

Para comprobar el funcionamiento de la salida de NASv2 ante redes neuronales tradicionales, se repite el experimento anterior con las mismas características excepto que usamos NASv2 para procesar el audio de entrada. En primer lugar, excitamos el NASv2 con los mismos 8 tonos puros con 2.5mW de potencia y generamos las muestras con el mismo procedimiento previamente explicado. Con una red neuronal de las mismas características que la usada previamente, es decir, con 10 neuronas en la capa oculta, se consigue clasificar estos sonidos con un 98,96% de éxito. Al añadir las muestras generadas a partir de la excitación del NASv2 con los mismos tonos puros pero con 1.2mW, se sigue realizando la clasificación con un error inferior al 2% con 10 neuronas en la capa interna, es decir, no es necesario ampliar el número de neuronas.

Tabla 7.3. Resultados de la red neuronal de 10 neuronas en la capa oculta para reconocer 8 tonos puros usando el NASv2

	Frecuencias de tono puro (Hz)								Total
	130.81	174.61	261.62	349.22	523.25	698.45	1046.5	1896.91	
%de éxito	100	100	91,86	99,8	100	100	100	100	98.96

Las conclusiones que obtenemos de estos experimentos son dos: por un lado comprobamos como la salida de ambos NAS es adecuada para plantear un sistema de reconocimiento mediante una red neuronal. Además, comparamos las

prestaciones de ambos NAS y deducimos que NASv2 es más útil en sistemas de reconocimiento de sonidos que se necesita que sean independientes del volumen, en cambio, en sistemas que sea necesario adaptarse a diferentes volúmenes de audio, es más útil NASv1.

Por último, se comprueba la precisión de la red en presencia de ruido blanco. Para probar la red en presencia de ruido, se han generado mediante Matlab 1500 tonos puros de cada clase a los que se les ha añadido ruido blanco, variando el SNR entre [32.19, -40] dBW en 30 intervalos. Estas muestras se han usado tanto para entrenar la red como para testarla. Los resultados para la red con 10 neuronas en la capa oculta se muestran en la Figura 7.6. En estas gráficas, el eje  $x$  representa la frecuencia de los tonos puros, el eje  $y$  representa el nivel de SNR que tienen los sonidos y el eje  $z$  la tasa de aciertos del sistema. La gráfica de arriba se corresponde con los resultados usando el NASv1 y la de abajo los resultados usando el NASv2. Se observa como NASv2 obtiene mejores resultados que NASv1. En las Tabla 7.4 y Tabla 7.5 se muestra un resumen de los datos de las gráficas. A esta tabla se le ha añadido una columna que muestra el porcentaje de acierto total. Con la misma potencia de señal que de ruido (0dBW SNR) se consigue un 87% de aciertos para NASv1 y 92% para NASv2, por lo tanto se puede concluir que NASv2 se comporta mejor ante presencia de ruido blanco. También están disponibles los datos completos en el anexo de este trabajo.

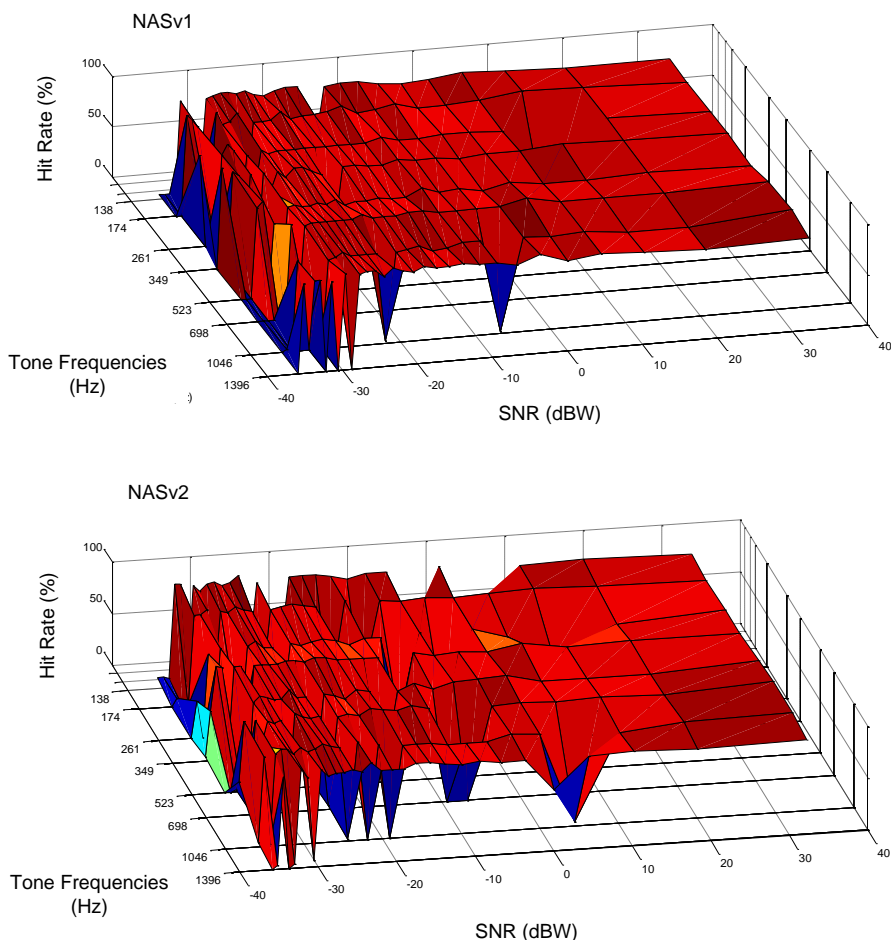


Figura 7.6. Tasa de acierto del sistema de reconocimiento de tonos puros en presencia de ruido blanco, para el NASv1 (arriba) y el NASv2 (abajo)

Tabla 7.4. Resumen de la tasa de aciertos del sistema de reconocimiento de tonos puros usando el NASv1

SNR (dBW)	Frecuencias de tono puro (Hz)								
	130,81	174,61	261,63	349,23	523,25	698,45	1046,5	1896,91	T.
32.18	91,66	87,5	90,47	89,58	93,75	95,74	95,83	91,67	92,02
0	95,83	87,5	88,09	87,5	95,83	91,48	91,67	59,52	87,18
-24.45	4,76	0	0	91,67	2,04	4,25	2,08	89,58	24,29
-35.84	6,25	0	0	4,17	0	32,48	4,76	0	5,95



Tabla 7.5. Resumen de la tasa de aciertos del sistema de reconocimiento de tonos puros usando el NASv2

SNR (dBW)	Frecuencias de tono puro (Hz)								
	130,81	174,61	261,63	349,23	523,25	698,45	1046,5	1896,91	T.
32.18	91,48	89,58	91,30	91,48	89,58	95,55	97,91	91,3	92,27
0	91,48	87,5	91,48	91,48	93,75	100	91,66	89,58	92,12
-24.45	95,74	87,5	89,36	93,62	87,50	93,33	89,58	89,36	90,75
-35.84	0	0	91,48	0	0	0	0	0	11,43

### 7.2.2. Identificación de notas musicales mediante red neuronal MLP

Es este apartado se expone el comportamiento de la red neuronal de tasas para realizar el reconocimiento de notas musicales de un piano electrónico basado en el estándar MIDI<sup>20</sup>. Las entradas de estas redes neuronales se obtienen de los datos de salida de los NAS planteados en el capítulo 5 ante el sonido generado por el piano electrónico.

Con el objetivo de elegir instrumentos para su reconocimiento, se observa que las notas de los instrumentos de cuerda, presentan una envolvente que está dividida en cuatro secciones diferentes: ataque, decaimiento, sostenimiento y relajación. A diferencia de las secciones de ataque y decaimiento donde están presentes una banda amplia de componentes de frecuencia, la frecuencia fundamental y sus armónicos son las componentes dominantes durante la etapa de sostenimiento. Por lo tanto, la sección de sostenimiento es la más recomendada para la extracción de la frecuencia característica. En cambio, en los instrumentos de viento no se produce este efecto porque no hay un golpe inicial en la cuerda.

En la primera fase de este experimento, vamos a capturar la salida del NASv1 y NASv2 ante las notas musicales F3, F4, F5 y F6 generadas por el piano virtual para el instrumento piano de cola acústico y en la segunda fase, vamos a capturar las mismas notas pero para un instrumento de viento como es la flauta, con el objetivo

<sup>20</sup> MIDI son las siglas de Musical Instrument Digital Interface es un protocolo para comunicación de música

de estudiar el comportamiento de ambos tipos de instrumentos. La potencia de la señal es de 800mV.

La construcción y testado de la red neuronal para el reconocimiento de las notas se realiza con los mismos pasos que el experimento anterior: 1) excitación del NAS, 2) captura salida del NAS, 3) cálculo de la tasa de eventos de cada canal para obtener las muestras de entrada de la red neuronal, 4) construcción, entrenamiento y testeo de la red. La tasa de eventos de cada canal se calcula en periodos de 20ms, tal y como lo hicimos en el experimento con los tonos puros. En la Figura 7.7 se observa la tasa de eventos, representada mediante colores, de cada canal (eje y) durante cada 20ms (el tiempo se representa en el eje x) para la captura realizada por el NASv2 de la nota F3 del piano electrónico. Sólo se representan los datos obtenidos del NASv2 izquierdo, porque la respuesta del derecho es similar. En esta figura se observa como las notas generadas por el piano virtual tienen actividad en un mayor rango de frecuencias y una mayor tasa de eventos en los primeros 0.5 segundos, para luego ir decayendo la intensidad de la nota. Este comportamiento es el que se espera de una nota generada por un instrumento como el piano.

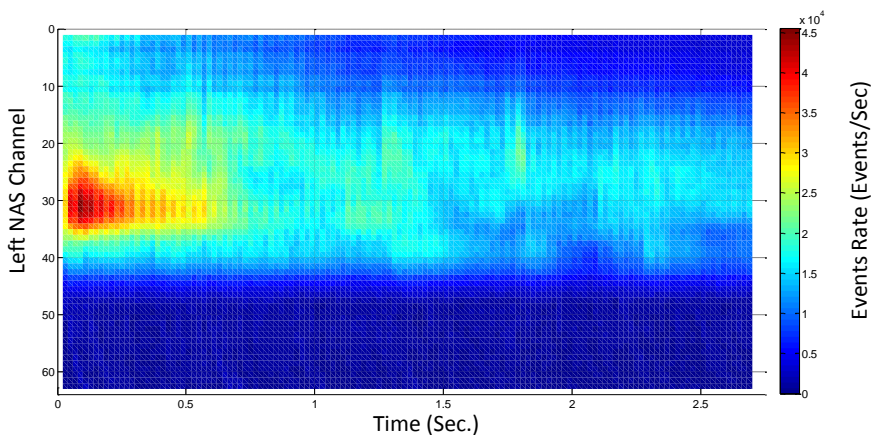


Figura 7.7. Tasa de eventos para cada canal del NASv2 cada 20ms

Tal y como se hizo en el experimento de reconocimiento de tonos puros, se normalizan los datos mostrados en la Figura 7.7 en el rango [0,1] dividiendo cada

valor por la tasa de eventos máxima en cada intervalo de tiempo. Con estas muestras se entrena la red neuronal que tiene las mismas características que en el experimento anterior, con la diferencia que ni con 40 neuronas en la capa oculta, se consigue una precisión mayor del 95%. En la Tabla 7.6 se muestran los resultados del sistema de reconocimiento con las redes neuronales tradicionales entrenadas por las muestras generadas previamente. Incluso con un alto número de neuronas en la capa interna, no se consiguen tan buenos resultados como se obtenían en el reconocimiento de tonos puros. Nos planteamos si las muestras no sean adecuadas debido al cambio de intensidad a lo largo del tiempo característico de las notas del piano. A continuación, se plantean una serie de modificaciones para obtener muestras que sean más adecuadas.

Tabla 7.6. % de aciertos en la clasificación para la red de reconocimiento de 4 notas del piano electrónico, calculando las muestras cada 20ms

Neuronas capa oculta	%Error total	
	NASv1	NASv2
10	87,82	90,3
20	94,1	94,2
30	94,91	95,9
40	94,94	95,24

En el sistema de reconocimiento que estamos desarrollando necesitamos determinar patrones significativos de lo que se desea reconocer, para ello, en este tipo concreto de redes, las muestras tienen que ser significativas de lo que queremos clasificar, por lo tanto, vamos a estudiar cómo se comporta la red neuronal ante distintos periodos de tiempo usados para el cálculo de las tasas de eventos de cada canal. En la Figura 7.7 se observa como la tasa de eventos de cada canal cambia a lo largo del tiempo para las notas el piano, por lo tanto, las muestras van a ser muy diferentes entre sí. Para obtener muestras más significativas, marcamos periodos de 0.1 segundos y 0.4 segundos para calcular las muestras con las que se entrenan la red neuronal. Los resultados obtenidos de entrenar a la red neuronal con estas muestras se observan en la Tabla 7.7 (NA significa No Aplica porque cuando se alcanza el 0% de errores ya el aumento de neuronas sólo

incrementa la complejidad de la red sin aportar mejoras). Para 0.1 segundos, las muestras de una misma clase siguen teniendo mucha variedad de valores y se obtienen resultados levemente mejores, en cambio, se obtienen mucho mejores resultados con el entrenamiento realizado con las muestras calculadas en periodos de tiempo de 0.4. Los valores de estas muestras se observan en la Figura 7.8, valores que se corresponden con la tasa de eventos de cada canal cada 0.4 segundos para tres pulsaciones de la tecla F3 del piano electrónico. Se observa que las muestras son más parecidas entre sí que en la Figura 7.7.

Tabla 7.7. % de fallos de clasificadas en la red de reconocimiento de patrones para las 4 notas del piano virtual, realizando el cálculo de las muestras en distintos periodos de tiempo

Neuronas capa oculta	%Error total para periodos de 0.1		%Error total para periodos de 0.4	
	NASv1	NASv2	NASv1	NASv2
20	7.5	6.9	4.6	3.2
30	1.5	0.9	0	0
40	0	0.9	NA	NA

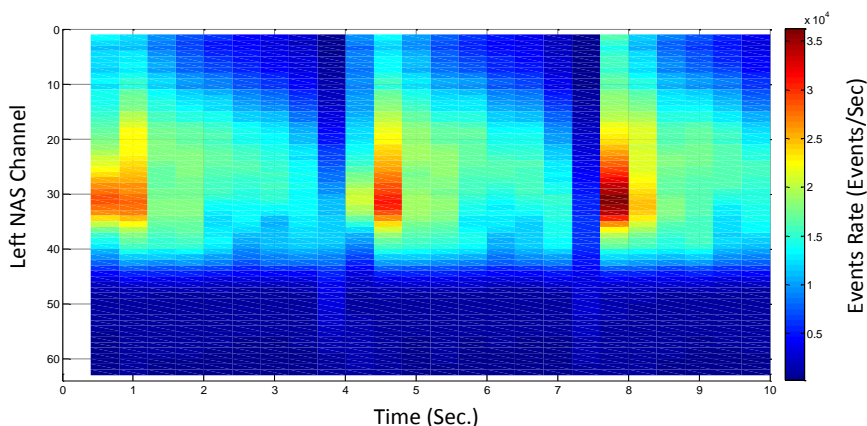


Figura 7.8. Tasa de eventos para tres notas F3 del piano cada 0.4 segundos

Después de observar el comportamiento de las notas del piano electrónico, vamos a comprobar cómo es la respuesta del NAS, y por lo tanto del sistema de reconocimiento, ante un instrumento de viento como es la flauta. A partir de los

datos capturados de la salida de ambos NAS para las notas F3, F4, F5 y F6 generadas por el piano virtual VMPK, calculamos la tasa de eventos de cada canal cada 20ms. La Figura 7.9 muestra la tasa de eventos (expresada mediante colores) para cada canal (eje  $y$ ) cada 20ms (eje  $x$ ) que se producen ante la excitación del NASv2 mediante la nota F3 de una flauta. En la figura se observa como las notas producidas por un instrumento de viento tienen siempre la misma respuesta frecuencial y puede mantener la misma intensidad del sonido. No obstante, hay técnicas como es el vibrato mediante la cual se puede cambiar la intensidad de una nota en la flauta. A partir de estos datos, vamos a calcular las muestras con las que vamos a entrenar la red neuronal de reconocimiento de patrones.

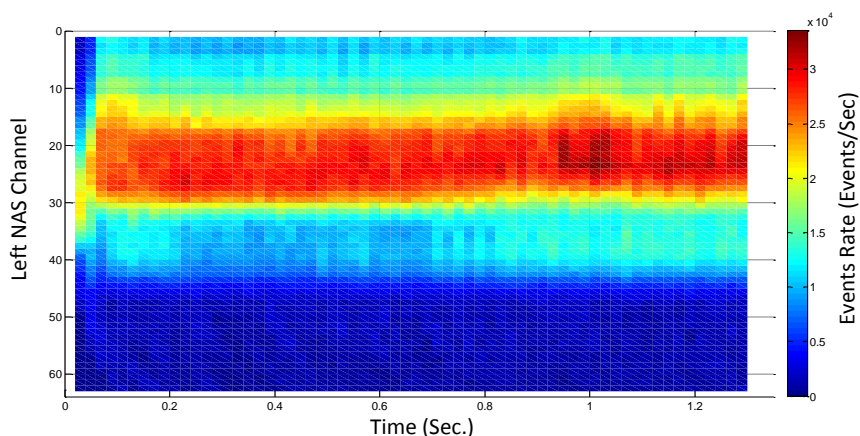


Figura 7.9. Tasa de eventos para la nota F3 de la flauta virtual cada 20ms

Al entrenar la red neuronal con las muestras calculadas a partir de la tasa de eventos cada 20ms, obtenemos los resultados mostrados en la Tabla 7.8. Se obtienen mejores resultados que en los experimentos con notas de piano.

Tabla 7.8. % de fallos de clasificadas en la red de reconocimiento de patrones para las 4 notas de la flauta, usando distintos tamaños en la capa oculta

Neuronas capa oculta	%Error total	
	NASv1	NASv2
10	89,72	91,4

<b>20</b>	94,3	95.51
<b>30</b>	96,54	96.3
<b>40</b>	97.1	100

### 7.3. Sistema de reconocimiento de sonidos mediante redes neuronales de convolución

Como se expone en el capítulo 6, las redes neuronales de convolución (ConvNet) consiguen una alta precisión para el reconocimiento de características en sistemas de visión, adaptándose a los cambios espaciales, incluso en ambientes ruidosos y con un bajo coste computacional. Esto lo consigue gracias a tres ideas: campos receptivos locales, pesos compartidos y muestreo espacial.

En este apartado se expone un sistema para el reconocimiento de sonidos que realiza la operación de convolución en una dimensión<sup>21</sup>, aplicada cada vez que llega un spike. La arquitectura del sistema consisten en una capa de neuronas de convolución, tantas como clases se quieran reconocer, de las cuales se almacena el potencial de la membrana y un valor umbral. Cada neurona tiene tantas entradas como canales de salida tenga el sistema de descomposición del audio en frecuencias. En este trabajo, usamos los sistemas descomposición de audio en frecuencias expuestos en el capítulo 5 (NAS), pero el sistema de clasificación se ha diseñado de forma que funcione con cualquier otro sensor de audio cuya salida se corresponda con la descomposición frecuencial del sonido a clasificar codificada en representación AER.

El sistema de convolución diseñado es un sistema pulsante, es decir, preparado para recibir y generar spikes. Se basa en las neuronas del tipo Integra-y-Dispara expuestas en el capítulo 6. En general, el funcionamiento del clasificador consiste en que cada vez que llegue un evento, se aplica la operación de convolución en cada neurona. Si el valor del resultado alcanza el valor umbral de una neurona, se resetea el potencial de la neurona y se genera un evento. En caso

---

<sup>21</sup> Los sistemas de reconocimiento basados en convoluciones aplicados en el campo de la visión son sistemas en 2 dimensiones.

contrario, el potencial de acción toma el valor del resultado. La operación de convolución de una neurona en un sistema digital queda definida matemáticamente en la Ecuación 7.4, donde  $x$  es un ciclo de reloj,  $W$  es el núcleo de la convolución,  $S$  es la salida de cada canal de la cóclea artificial (que es la entrada del sistema de clasificación) e  $Y$  es la salida de la convolución, cuyo valor depende del valor que tenía en el instante de tiempo representado por  $x$ . La Ecuación 7.5 representa la salida de una neurona, que sólo se produce si el valor de  $Y$  es mayor o igual al umbral de dicha neurona, representado por  $\theta$ , y como el valor del potencial de la neurona se resetea en esa condición. En dichas ecuaciones,  $M$  representa el número de canales que tenga la cóclea artificial de los que provienen los spikes de entrada.

$$Y(x + 1) = Y(x) + \sum_{m=0}^M (W(m) * S(x))$$

Ecuación 7.4

$$Out = Y(x) \geq \theta \rightarrow Y(x) = 0$$

Ecuación 7.5

La arquitectura hardware de cada neurona que implementa las ecuaciones anteriores se corresponde con la Figura 7.10. Este diseño se realiza para el NAS presentado en el capítulo 5, que tiene 128 canales de salida binaural, es decir, 64 canales para dos entradas, los 64 primeros canales se corresponde con la señal de audio izquierdo y los 64 siguientes para el derecho, por lo tanto, la neurona se corresponde al NAS izquierdo porque recibe los spikes generados por los 64 primeros canales. Tanto los spikes positivos y negativos de un canal activan la operación de convolución. En la Figura 7.10, la salida del canal  $i$ -ésimo del NAS se representa con  $Ch_i$ , el núcleo de convolución se representa con  $W$  y consiste en un vector de 64 elementos para cada sonido a reconocer. El valor del potencial de la membrana se almacenará en el registro *Accum*, valor que se resetea si se alcanza el valor umbral ( $\theta$ ). Se ha añadido la funcionalidad de resetear también el contador si después de un cierto tiempo no se ha llegado al umbral.

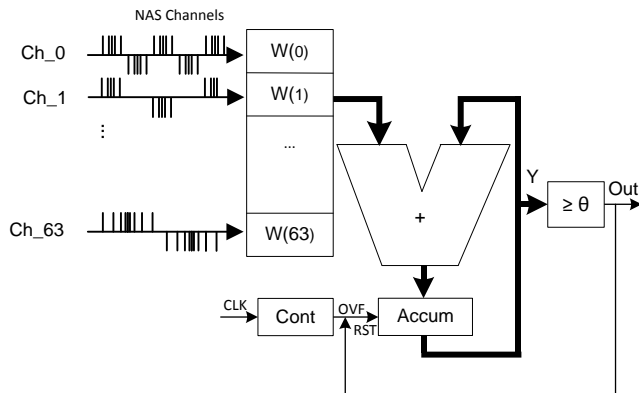


Figura 7.10. Arquitectura de una neurona de la red de convolución

Este sistema utiliza la información pulsante sobre la intensidad de cada una de las componentes frecuenciales del sonido de entrada, por lo tanto, es un sistema basado en las frecuencias características y la amplitud del sonido.

Los valores de los núcleos de convolución de cada neurona se obtienen a partir de la tasa de eventos de cada canal ante sonidos ejemplares de cada clase a clasificar. En el siguiente apartado se muestran ejemplos de los núcleos de sistemas de convolución de reconocimiento de sonidos.

Basado en el diseño explicado previamente, primero realizamos una simulación software de la arquitectura de la ConvNet de reconocimiento de sonidos, para posteriormente, una vez testada la arquitectura, implementar el sistema en hardware.

A continuación presentamos la arquitectura y comportamiento del sistema de reconocimiento basado en convoluciones, el escenario completo que se va a usar para probar la precisión del sistema de reconocimiento mediante dos experimentos diferente: reconocimiento de tonos puros ante la presencia de ruido blanco y reconocimiento de notas musicales.



### 7.3.1. Arquitectura del sistema de convolución

El sistema de convolución hardware expuesto en este trabajo está integrado en la FPGA Spartan6-150t que tiene la placa AER-NODE (Iakymchuk et al. 2014). El uso de esta placa aporta escalabilidad en la conexión con otros sistemas, por lo tanto, el sistema de convolución que exponemos en este apartado puede recibir spikes codificados en AER desde cualquier cóclea artificial, también permite conectar la salida del sistema de convolución a otro sistema de procesamiento cuya interfaz sea AER.

Esta PCB consta de dos puertos AER, uno de entrada y otro de salida. En la arquitectura de la ConvNet expuesta en la Figura 7.11 se observa como ambos puertos se usan para la comunicación del sistema con el exterior. El funcionamiento del sistema consiste en que cada vez que llega un evento, se recibe, se envía a todas las neuronas y se activa la señal STROBE. El tamaño del evento es 6 bits porque representa el canal del NAS izquierdo que ha generado el spike. El sistema debe contener tantas neuronas como clases se quieran reconocer. Cada neurona procesa el evento tal y como se observa en la Figura 7.12, sumando al valor del potencial de la membrana (*Accum*), el valor del núcleo del elemento indicado por el evento ( $W(EVENT)$ ). Si el resultado de la operación sobrepasa el valor umbral, la neurona genera un spike (*OUT*). Los spikes de salida de las neuronas son gestionados por el monitor distribuido, expuesto en el capítulo 4. Este componente se encarga de codificar los spikes en direcciones AER y los envía por el bus AER a otra capa de procesamiento neuronal.

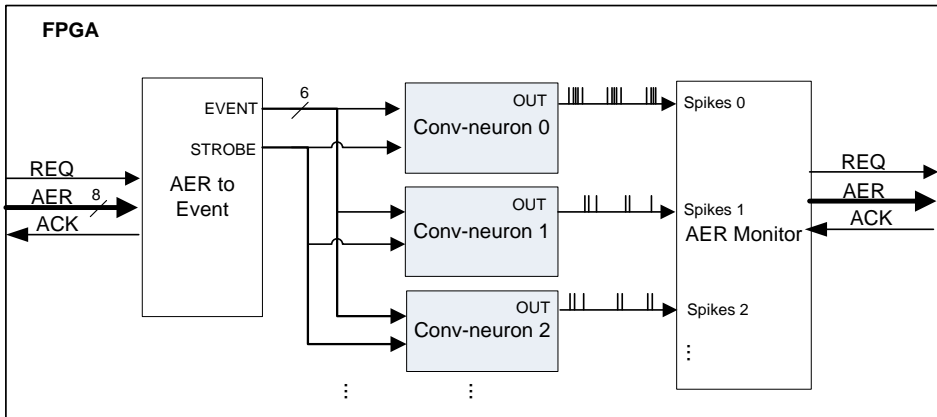


Figura 7.11. Arquitectura de la red neuronal de convolución

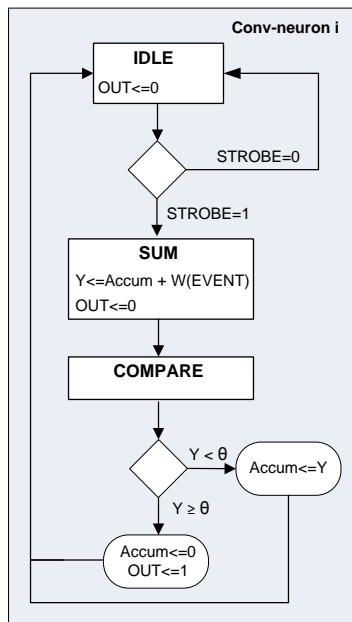


Figura 7.12. Diagrama de flujo de una neurona de convolución

En el sistema de reconocimiento las señales de entrada/salida del sistema son: la señal de reloj, la señal de reset, las señales de comunicación del bus AER de entrada, que en nuestro caso son 8 para codificar los spikes y 2 señales usadas para

la comunicación asíncrona del bus AER (ACK y REQ) y las señales del protocolo AER de salida, en el que el número de líneas es el número de clases que se quiera reconocer. En el apartado Síntesis del sistema de convolución se detalla el número de señales de entrada/salida para diferentes números de clases a reconocer.

### **7.3.2. Entrenamiento de la red de convolución**

El entrenamiento de la red de convolución se realiza offline y su objetivo es calcular el núcleo de convolución para cada clase que se quiera reconocer y el valor umbral de cada neurona que forma la red de convolución.

El núcleo de convolución se obtiene a partir de la tasa de eventos de cada canal del NAS ante sonidos ejemplares de cada clase que se quiere reconocer. Se han desarrollado scripts en Matlab que capturan los eventos de salida del NAS ante los sonidos de muestra y obtienen los valores del núcleo de convolución para cada clase. Estos scripts se han desarrollado de forma que sean configurables respecto el número de canales que posea la cóclea artificial que se vaya a usar. En este trabajo estamos usando el NAS expuesto en el capítulo 5, que tiene 128 canales, 64 para la señal de audio izquierda y 64 para la derecha. En este trabajo, de igual forma que se hizo en el sistema de reconocimiento MLP, sólo se trabaja con los eventos generados por el canal izquierdo. Con el objetivo que los núcleos de convolución sean valores independientes del volumen de la señal de entrada, se normalizan en un intervalo  $[0,1]$ . El número de bits necesario para representar los valores umbrales es parametrizable y para los experimentos realizados se ha marcado a 12.

Para calcular el valor umbral de cada neurona, una vez que se tienen los valores de los núcleos, hemos desarrollado scripts que calculan la salida de la operación de convolución según la Ecuación 7.4, con los valores de los núcleos calculados en la fase anterior para un determinado periodo de tiempo. Este periodo de tiempo se estable con la duración del menor sonido audible que es de 10ms. El número de bits necesarios para codificar estos valores también está parametrizado y para los experimentos realizados se ha marcado en 32.

Para integrar con rapidez los valores de los núcleos y de los umbrales en hardware, se ha desarrollado un script que genera el código VHDL necesario para implementa estos valores de las neuronas de convolución en hardware. De esta forma, el proceso de entrenamiento offline es automático y, resumiendo, consta de las siguientes fases: 1) capturar la salida del NAS ante sonidos de muestra de las clases que se quiera reconocer; 2) a partir de la tasa de eventos de cada canal del NAS, calcular el núcleo de convolución de cada neurona; 3) calcular el valor umbral de cada neurona a partir de la ecuación básica de convolución con los núcleos calculados en la fase anterior; 4) obtener el código VHDL que integra dichos valores en el sistema de convolución hardware.

### 7.3.3. Síntesis del sistema de convolución

Una vez desarrollada en VHDL la arquitectura y funcionalidad de la red de convolución y realizado el entrenamiento, se procede a sintetizar la red de convolución en la FPGA Spartan6-150t, integrada en la PCB AER-NODE (Iakymchuk et al. 2014). Los dos puertos AER de la PCB disponen de 28 bits para codificar las direcciones AER. Para el puerto de entrada, de estos 28 bits sólo son necesarios 8, porque tal y como se muestra en la Figura 7.13, el NAS que estamos usando necesita 8 bits para codificar los spikes generados. Este número varía en función del número de canales del que disponga la cóclea artificial. En la Figura 7.13 el bit menos significativo indica si el spike es negativo o positivo, los siguientes 6 bits codifican el canal del que proviene el spike, y el bit más significativo se usa para distinguir si es un spike generado por el NAS izquierdo o derecho.

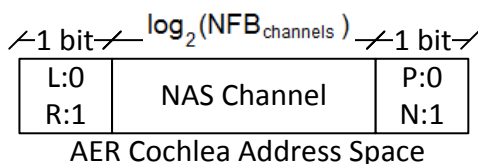


Figura 7.13. Bits necesarios para codificar los spikes del NAS

El número de bits necesarios para los eventos de salida depende del número de clases que se quiera reconocer. La relación entre estos valores se muestra en la Ecuación 7.6, siendo  $N_{clases}$  el número de clases a reconocer y  $N_{bits}$  el número de líneas necesarias para codificar el evento en el bus AER. Además de los dos puertos AER, el sistema también tiene la señal de reloj y la señal de reset.

$$N_{bits} \geq \log_2(N_{clases})$$

Ecuación 7.6

Los requisitos hardware necesarios para esta implementación depende también del número de clases a reconocer. Hay que destacar que esta arquitectura no demanda recursos FPGA especializados, como son multiplicadores, procesadores embebidos, DSP; sólo necesita lógica digital común como contadores, comparadores, sumadores y registros con un bajo número de bits y bajo nivel de conectividad. En la Tabla 7.9 se muestran los requisitos hardware de la arquitectura de red de convolución expuesta según el número de neuronas implementadas. En la Figura 7.14 se muestra la relación entre el número de neuronas de convolución y el número de slices necesarios para sintetizarlas, con los cuales se ha obtenido la recta de regresión de la relación mostrada en la Figura 7.14.

Tabla 7.9. Requisitos hardware de la red de convolución para diferentes números de neuronas

Nº de neuronas de convolución	Nº de Slices	Máxima frecuencia de reloj (MHz)	Nº de señales de entrada/salida
6	161	141,5	17
8	185	131,6	17
12	245	125,27	18
24	480	119,29	19

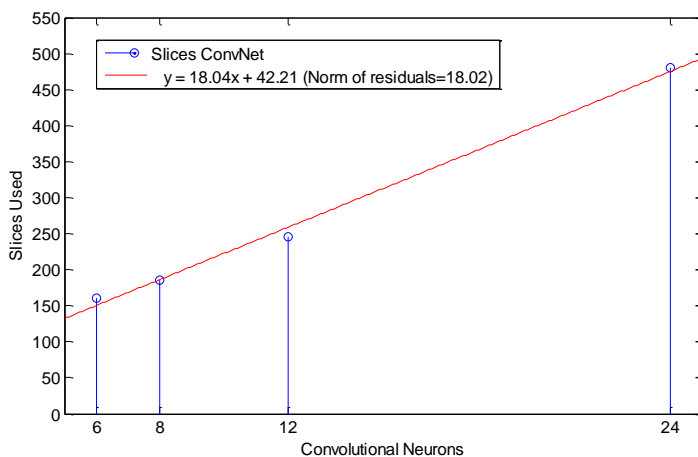


Figura 7.14. Relación entre el número de neuronas de convolución y el número de slices necesarios

### 7.3.4. Escenario experimental

Para evaluar la precisión del sistema de reconocimiento expuesto, se han desarrollado dos experimentos: reconocimiento de tonos puros en presencia del ruido blanco y reconocimiento de notas musicales ante un instrumento de cuerda y uno de viento.

Para poder ejecutar los experimentos, hemos construido un escenario experimental donde el sistema de reconocimiento de tiempo real es excitado con las salidas de los NAS<sup>22</sup> ante los sonidos a reconocer. La Figura 7.15 muestra una fotografía del escenario experimental, formado por la placa de desarrollo Xilinx ML507 (Xilinx-ML507 2015), dónde está sintetizado el NAS, la PCB AER-NODE (Iakymchuk et al. 2014), en la cual está sintetizado el sistema de reconocimiento ConvNet y la placa USB AERmini2, usada para monitorizar los eventos de salida. Debido a que la placa de entrenamiento ML507 no dispone de conector AER, está conectado un adaptador entre el puerto GPIO al bus AER. La placa USBAERmini2 envía mediante USB los eventos de salida del sistema de reconocimiento. En la

<sup>22</sup> Se realizan los experimentos tanto para el NASv1 y NASv2 expuestos en el capítulo 5

Figura 7.17 se muestra el diagrama de bloques las conexiones entre los componentes del experimento.

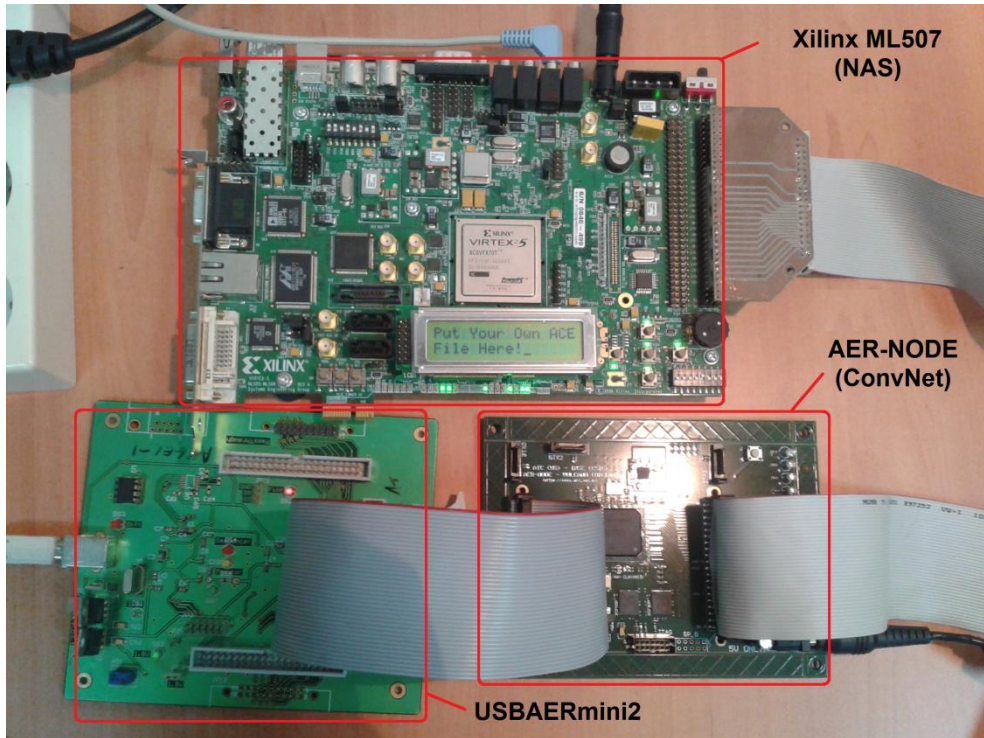


Figura 7.15: Fotografía del escenario experimental del sistema de reconocimientos de sonidos

Todo experimento se realiza en 3 fases, la fase de entrenamiento, la fase de ejecución y la fase de cálculo de la tasa de aciertos. Las fases de entrenamiento y cálculo de la tasa de aciertos son offline. En la Figura 7.16 se muestra el diagrama de bloques del escenario de trabajo para la fase de entrenamiento. En la Figura 7.17 se muestra el diagrama de bloques de las fases de ejecución y la fase de cálculo de la tasa de aciertos. El proceso de cada una de las fases se detalla a continuación:

1. La fase de entrenamiento se divide en las siguientes sub-fases:
  - a. **Generación de muestras de cada clase:** es la primera fase del entrenamiento. Se excita el NAS con los sonidos de muestra de cada clase a reconocer, y mediante la placa USB AERmini2 se

capturan los eventos de salida del NAS. Estos datos se usan como los datos de entrenamiento para obtener el núcleo de cada neurona de convolución.

- b. **Cálculo del núcleo de convolución de cada neurona:** es la segunda fase del entrenamiento y consiste en calcular el núcleo de cada neurona mediante la tasa de eventos de cada canal. Los valores del núcleo se codifican en hardware mediante 12 bits, por lo tanto pueden tomar valores en el intervalo [0,4095].
  - c. **Cálculo del valor umbral de cada neurona:** se calcula el valor del umbral mediante un script que implementa la Ecuación 7.4, usando los núcleos de cada neurona, para los eventos generados cada 20ms.
  - d. **Generación de VHDL:** se generan mediante scripts de Matlab el código VHDL de las neuronas a partir del número de clases, de los valores de los núcleos de convolución y de los umbrales. Ya podemos programar las FPGA Virtex5 (XC5VFX70T) con el NAS y la FPGA Spartan6 (XC6SLX150T) con la ConvNet.
2. **Ejecución:** se reproducen los sonidos de prueba con las que se excita el NAS y la salida del NAS se procesa por el sistema de reconocimiento. La ConvNet genera la salida en tiempo real. Mediante la placa USB-AER-mini2 (Berner et al. 2007), se envían los eventos AER al PC para su posterior análisis.
  3. **Calcula de la tasa de aciertos:** en el PC, mediante un script en Matlab, se calcula la tasa de aciertos del sistema de reconocimiento.

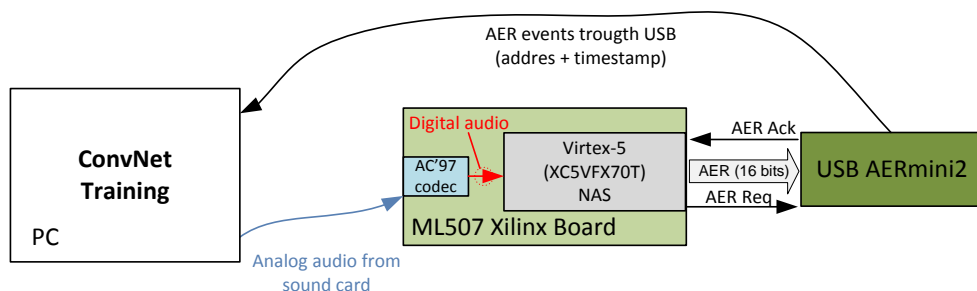


Figura 7.16. Diagrama de bloques de la fase de entrenamiento del sistema de reconocimiento basado en una red de convolución



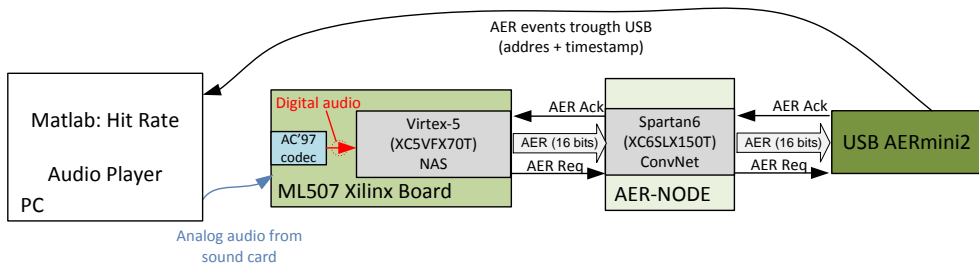


Figura 7.17. Diagrama de bloques de la fases de ejecución del sistema de reconocimiento basado en una red de convolución

Con el primer experimento queremos evaluar la inmunidad al ruido del sistema de reconocimiento. Para ello, hemos generado tonos puros mediante Matlab con las frecuencias que se muestran en la Tabla 7.10 y potencia 2.5 mW, para usarlos como sonidos para el entrenamiento. La selección de esas frecuencias se debe a que en el siguiente experimento se reconocen notas musicales, por lo tanto, hemos elegido las frecuencias fundamentales de las notas musicales que se observan en la Tabla 7.10. Para probar la red, se han generado mediante Matlab 1500 tonos puros de cada clase a los que se les ha añadido ruido blanco, variando el SNR entre [Infinito<sup>23</sup>, -35.84] dBW en 30 intervalos.

Tabla 7.10. Relación entre la frecuencia fundamental de los tonos a reconocer y las notas musicales

<b>Freq. (Hz)</b>	130,813	174,614	261,626	349,228	523,251	698,456	1046,5	1896,91
<b>Note</b>	C3	F3	C4	F4	C5	F5	C6	F6

Hemos usado tanto el NASv1 como el NASv2 como sistema de descomposición del audio en sus componentes frecuenciales. Los valores de los núcleos de convolución obtenidos normalizados entre [0, 1] se muestran en la Figura 7.18. El eje *x* de estas gráficas representa los canales del NAS izquierdo y el eje *y* representa la tasa de eventos normalizada para cada canal. Se observa como los tonos más graves tienen sus máximos en canales mayores que los agudos, porque

<sup>23</sup> Infinito se usa para indicar que no se ha añadido ruido

como se expone en el capítulo 5, los canales mayores tienen frecuencia fundamental menor que los canales menores. También se observa las diferencias entre las ganancias de los filtros del NASv1 y NASv2, expuestas en detalle en el capítulo 5.

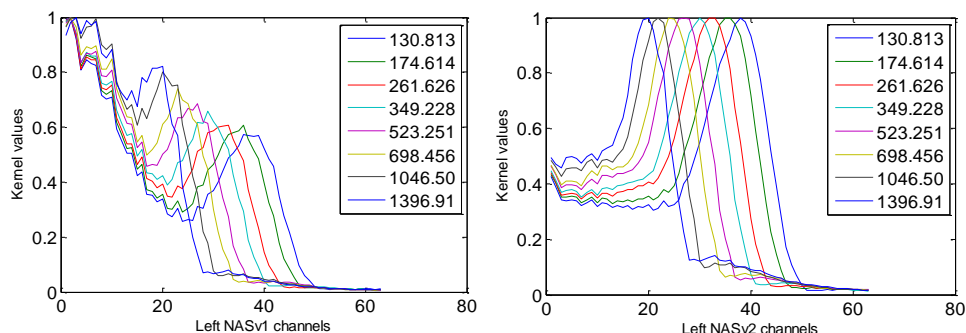


Figura 7.18. Valores de los núcleos de convolución para cada tono puro respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2 (derecha).

En el siguiente apartado se muestran los resultados del experimento tanto para el NASv1 como para el NASv2.

En el segundo experimento, se ha excitado tanto el NASv1 como el NASv2 con las notas musicales F3, F4, F5 y F6 cuya frecuencia fundamental se observa en la Tabla 7.11. Estas notas se han generado con el piano electrónico VMPK (VMPK n.d.) para los instrumentos piano y flauta.

Tabla 7.11. Frecuencia fundamental de las notas musicales F3, F4, F5 y F6

Note	F3	F4	F5	F6
Freq. (Hz)	174,61	349,23	698,46	1396,91

En la Figura 7.19 se muestran los valores de los núcleos de convolución obtenidos normalizados entre [0, 1] para las notas de piano, a la izquierda los núcleos calculado mediante NASv1 y a la derecha los núcleos calculado con el NASv2. El eje x de esta figura representa los canales del NAS y el eje y representa

la tasa de eventos normalizada para cada canal. En la Figura 7.20 se muestra la misma gráfica, pero para las notas de la flauta.

Figura 7.19. Valores de los núcleos de convolución para cada nota de piano respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2.

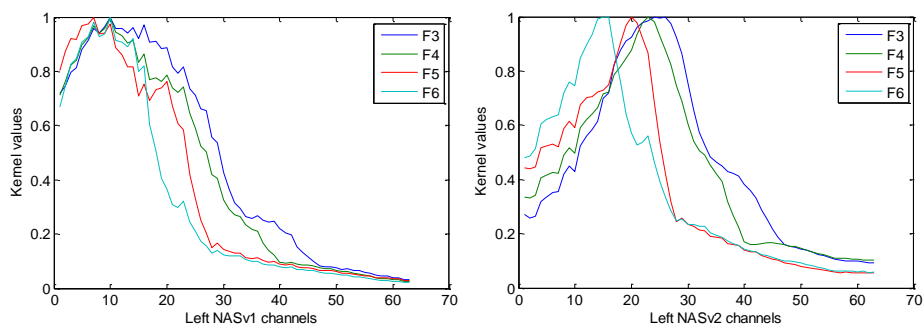


Figura 7.20. Valores de los núcleos de convolución para cada nota de piano respecto los canales de salida del NAS izquierdo. A la izquierda se muestra los núcleos obtenidos del NASv1 y a la derecha los obtenidos del NASv2.

Se observa como las notas del piano se ven afectadas por las dos primeras etapas de su envolvente temporal, ataque y decaimiento, en las que además de la frecuencia fundamental y los armónicos, se activan otras frecuencias.

### 7.3.5. Resultado de los experimentos

En el primer experimento, para evaluar la ConvNet implementada, se entrena la red neuronal de convolución con el objetivo de clasificar 8 tonos puros. Para la primera fase de extracción de datos del sistema de reconocimiento hemos usado tanto el NASv1 como el NASv2, expuestos en detalle en el capítulo 5. Por lo tanto, se exponen los resultados de ambos experimentos comparándolos entre ellos.

El sistema de reconocimiento ha sido estimulado con 1500 tonos puros con diferentes niveles de ruido blanco. Los resultados se muestran en la Figura 7.21. En estas gráficas, el eje  $x$  representa la frecuencia de los tonos puros, el eje  $y$  representa el nivel de SNR que tienen los sonidos y el eje  $z$  la tasa de aciertos del

sistema. En el eje  $y$ , se representa mediante ‘Inf.’ la ausencia de ruido. Se observa como sin la presencia de ruido blanco, el sistema con el NASv2 consigue una tasa de acierto del 100% y con el NASv1 se consigue 99%. Para ambos NAS, se observa como consiguen una tasa de acierto superior al 95% ante ruido blanco con la misma potencia que la señal a reconocer. Para potencias de ruido blanco superior a la potencia de la señal, la precisión del sistema para el NASv1 en baja; en cambio, para el NASv2 sigue consiguiendo una alta tasa de aciertos con tan elevada relación entre el ruido y la señal. En las Tabla 7.12 y Tabla 7.13 se muestran un resumen de los datos de las gráficas. También están disponibles los datos completos en el anexo de este trabajo.

El sistema de reconocimiento consigue una gran precisión aún con un elevado nivel de ruido, obtenido gracias al sistema de reconocimiento basado en convoluciones como por la gestión pulsante de la información en el NAS.

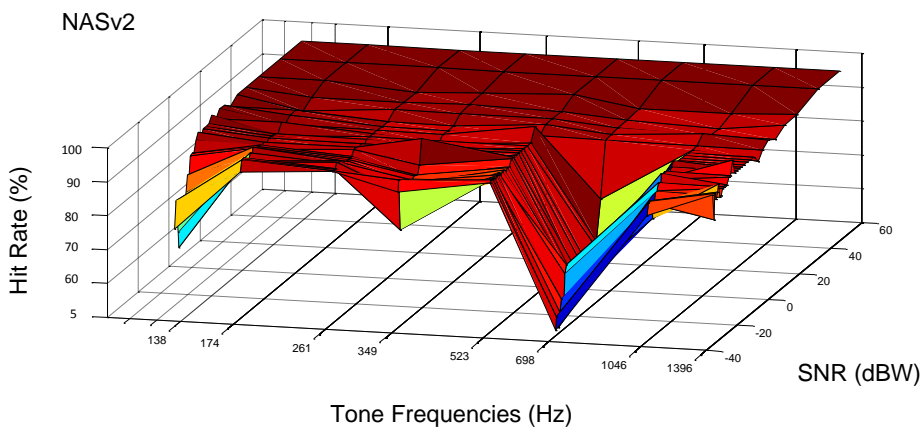
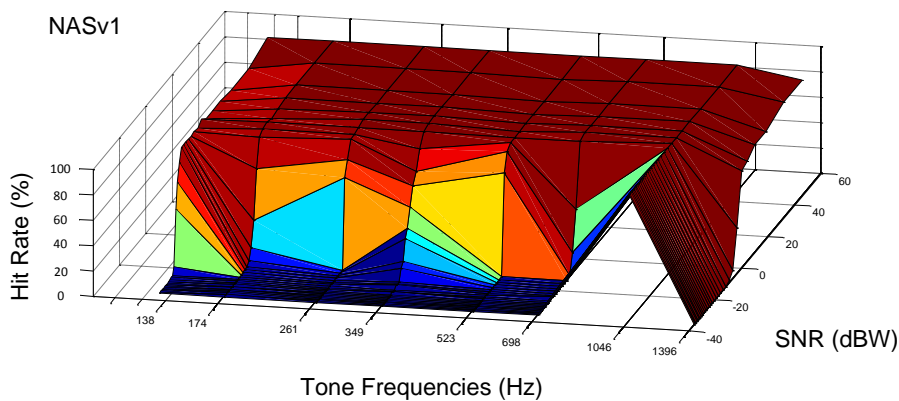


Figura 7.21. Tasa de acierto de la ConvNet para el reconocimiento de tonos puros en presencia de ruido blanco, para el NASv1 (arriba) y el NASv2 (abajo)

Tabla 7.12. Resumen de la tasa de aciertos de la ConvNet para el reconocimiento de tonos puros usando el NASv1

SNR (dBW)	Frecuencias de tono puro (Hz)								
	130,81	174,61	261,63	349,23	523,25	698,45	1046,5	1896,91	T.
Inf.	100	100	100	100	99,63	100	100	91,2	98,85
0	93,41	96,43	100	100	97,82	100	100	76,8	95,56
-24,45	65,32	1,32	0	2,9	0	0	100	0	21,19
-35,84	0	0	0	0	0	0	100	0	12,5

Tabla 7.13. Resumen de la tasa de aciertos de la ConvNet para el reconocimiento de tonos puros usando el NASv2

SNR (dBW)	Frecuencias de tono puro (Hz)								
	130,81	174,61	261,63	349,23	523,25	698,45	1046,5	1896,91	T.
<b>Inf.</b>	100	100	100	100	100	100	100	100	100
<b>0</b>	98,2	98,14	96,27	96,1	98	96,18	97,84	97,56	97,28
<b>-24.45</b>	93,63	96,67	94,7	90,91	94,46	72,31	96,51	95,8	91,87
<b>-35.84</b>	75,42	93,43	95,13	90	94,46	51,93	87,2	94,3	85,23

Los resultados del segundo experimento, que consiste en el reconocimiento de cuatro notas musicales de piano y flauta se muestran en la Tabla 7.14. Se observa como con sólo 4 neuronas, la ConvNet desarrollada consigue una alta precisión. Para aumentar la precisión del sistema de reconocimiento de notas del piano, se ha añadido otra capa de 4 neuronas del tipo integra-y-dispara, conectada a todas las neuronas de la capa de convolución. Con esta modificación, el sistema obtiene los resultados que se observan en la

Tabla 7.15.

Tabla 7.14. Porcentaje de aciertos del experimento de clasificación de notas musicales de piano y flauta

Clase	Piano		Flauta	
	NASv1	NASv2	NASv1	NASv2
<b>F3</b>	67,2	99,1	91,6	98,7
<b>F4</b>	14,4	29,6	71,5	83,3
<b>F5</b>	97,1	95,8	87,3	96,8
<b>F6</b>	73,3	88,4	91,5	94,9
<b>Total</b>	61,9	78,2	85,5	93,43

Tabla 7.15. Tasa de aciertos del experimento de clasificación de notas musicales de piano con dos capas de neuronas

Clase	Piano	
	NASv1	NASv2
<b>F3</b>	98,2	1
<b>F4</b>	96,4	97,1
<b>F5</b>	99,3	98,2
<b>F6</b>	97,1	96,8
<b>Total</b>	97,75	98,03

#### 7.4. Comparativa de los sistemas de reconocimiento expuestos

El sistema hardware (sintetizado en FPGA) de reconocimiento de sonidos se basa en un sistema de descomposición de audio en componentes espectrales inspirado en la cóclea biológica<sup>24</sup> y un sistema de clasificación basado en una ConvNet. Tal y como se expuso en el apartado 6.3, no hay ningún sistema que realice reconocimiento de tonos puros ni notas musicales en hardware: existen trabajos que clasifican entre dos tipos de sonidos usando cócleas analógicas y trabajos desarrollados en software que reconocen notas musicales.

A continuación, se compara el sistema de reconocimiento basado en convoluciones con el sistema de reconocimiento software basado en una red neuronal MLP de retropropagación expuesta en el apartado 7.2. También se compara la ConvNet con los desarrollos software expuestos en el apartado 6.3, aunque la mayoría de dichos procesamientos no se pueden desarrollar en hardware por el alto coste computacional.

<sup>24</sup> Este dispositivo está expuesto en el capítulo 5

Respecto a los experimentos con tonos puros, tal y como se observa en la Tabla 7.16, el sistema de clasificación basado en ConvNet, obtiene mejores resultados que la red MLP. Otra ventaja de la ConvNet es que sólo usa 8 neuronas respecto a las 18 que usa la MLP. Además, la fase de entrenamiento de la ConvNet necesita sólo una muestra representativa de cada clase, y no necesita incluir muestras con ruido para obtener dichos resultados, en cambio la MLP usa 400 muestras sin ruido y 400 muestras con ruido para entrenarse, lo que implica que necesita mayor tiempo de entrenamiento que la ConvNet.

Tabla 7.16. Resultados de la ConvNet y de la red MLP para el reconocimiento de tonos puros en presencia de ruido blanco

SNR (dBW)	ConvNet (8 neuronas)		Red MLP (18 neuronas)	
	NASv1	NASv2	NASv1	NASv2
<b>Sin ruido</b>	98,85	100	96,5	98,96
<b>0</b>	95,56	97,28	87,18	92,12
<b>-24.45</b>	21,19	91,87	24,29	90,75
<b>-35.84</b>	12,5	85,23	5,95	11,43

Respecto a los resultado de los experimentos con notas musicales de piano, que se resumen en la Tabla 7.17, la ConvNet obtiene un 61,9% de precisión para NASv1 y 78,2% para NASv2, que son resultados peores a los que presenta la red neuronal de tasas MLP, pero hay que tener en cuenta que la red MLP necesita 14 neuronas y la ConvNet sólo 4. Al ampliar la ConvNet con otra capa de 4 neuronas, en total 8 neuronas, se obtiene una tasa de aciertos de **97,75%** para NASv1 y **98,03%** para NASv2, resultados mejores que los conseguidos por la MLP de 14 neuronas.

Tabla 7.17. Resultados de la ConvNet y de la red MLP para el reconocimiento de notas musicales

Piano		Flauta	
ConvNet (4 neuronas)	Red MLP (14 neuronas)	ConvNet (4 neuronas)	Red MLP (14 neuronas)



NASv1	NASv2	NASv1	NASv2	NASv1	NASv2	NASv1	NASv2
61,9	78,2	87,82	90,3	85,5	93,43	89,7	91,4

Para las notas musicales de flauta, ambas redes obtienen mejores resultados que para las notas del piano. Esto es debido a que el sonido de los instrumentos de viento mantienen la frecuencia fundamental y los armónicos durante toda la reproducción, en cambio, las notas del piano tienen un mayor rango de frecuencias además de las fundamenta y armónicos durante las dos primeras etapas de la reproducción.

Resumiendo, el sistema de reconocimiento hardware expuesto en este trabajo, obtiene un 98,03% de acierto en el reconocimiento de 4 notas de piano usando 8 neuronas. Comparando estos datos con los sistemas de reconocimiento de notas musicales expuestos en el capítulo 6 Tabla 6.2 se observa que obtenemos mejores resultados que los trabajos previos respecto a la relación número de clases – coste computacional: el trabajo expuesto en (Guerrero-turrubiates et al. 2014) reconoce 12 notas musicales de la guitarra eléctrica mediante una red MLP de 30 neuronas con una precisión de 97,5%. El sistema expuesto en (Barbancho et al. 2012) reconoce entre 48 acordes de guitarra mediante HMM de 330 estados con una precisión del 95%. (Pishdadian & Nelson 2013) presenta un sistema basado en KNN para reconocer entre 26 notas de piano que obtiene un 91,45% de porcentaje de aciertos.

## 8. Conclusiones

Tras el desarrollo de este trabajo, expuesto a lo largo del presente documento, se remarcan las aportaciones realizadas y las conclusiones alcanzadas:

- Se ha realizado un estudio en profundidad del sistema auditivo humano y de los modelos físicos más relevantes sobre el comportamiento de la cóclea humana. Estos estudios han supuesto el punto de partida para el sistema de procesamiento de audio neuro-inspirado presentado.
- Se ha realizado un estudio del funcionamiento de las neuronas biológicas y la conexión entre ellas, además de un análisis de la codificación de la información pulsante.
- Se han estudiado los sistemas de monitorización de la información pulsante; a partir de los mismos se ha propuesto, diseñado e implementado un nuevo monitor pulsante. Se ha comprobado que el propuesto en este trabajo amplía las prestaciones con respecto a los existentes. Este módulo permite la comunicación entre capas de neuronas artificiales y por lo tanto, lo hemos usado para la conexión entre los diferentes sistemas que se han implementado en este trabajo.
- Se ha diseñado, implementado y caracterizado una cóclea neuromórfica en la que el procesamiento de la información es pulsante, siendo la primera vez que se construye una cóclea artificial que realiza el procesamiento de la información de forma exclusivamente pulsante. Se ha realizado en una plataforma hardware basada en una FPGA.
- Se han diseñado, implementado y evaluado sistemas de reconocimiento de sonidos basados en modelos neuronales artificiales, a partir de la información generada por la cóclea artificial pulsante. Se han realizado dos tipos de redes:

- Redes neuronales artificiales tradicionales.
- Redes neuronales artificiales pulsantes de convolución. Se ha implementado en una plataforma hardware basada en una FPGA. Según el estudio realizado, es el primer sistema basado en redes neuronales pulsante de convolución para el reconocimiento de sonidos implementado en hardware.
- Con objeto de evaluar el trabajo realizado, se han reconocido con éxito tres tipos de sonidos: tonos puros, notas musicales y el sonido que realiza un motor a diferentes revoluciones.
- Se han comparado los resultados obtenidos por el sistema de reconocimiento totalmente pulsante con otros trabajos, obteniendo el sistema pulsante mejores resultados y con menor coste computacional.

## 9. Trabajo futuro

Tras la exposición del trabajo realizado a lo largo de este documento, han surgido una serie de nuevas líneas de investigación y de desarrollo de sistemas neuromórficos de procesamiento de audio, tanto como evolución de los aquí planteados, como mejoras de los propuestos por otros autores.

En primer lugar se expondrán los trabajos futuros que surgen para cada uno de los dos ámbitos estudiados, la cóclea pulsante y los sistemas de reconocimientos de sonidos neuromórficos:

- Para el sistema neuromórfico de audición (cóclea pulsante), se proponen los siguientes trabajos futuros:
  - Con los filtros de spikes que tenemos disponibles actualmente sólo se pueden diseñar filtros paso de baja sobreamortiguados, obteniendo filtros paso de banda equivalentes con un factor de calidad ( $Q$ ) estrictamente superior a 1. Para aumentar el factor de calidad de los filtros paso de banda de la cóclea tenemos que disminuir el factor de amortiguamiento de los filtros paso de baja. Un trabajo futuro consiste en mejorar los filtros paso de baja pulsantes, pero teniendo en cuenta que esto aumenta en gran medida la carga computacional del algoritmo genético, incrementando la dificultad y el error cometido en la fase de sintonización. A pesar de esta dificultad que nos encontramos, nuestros filtros paso de banda de spikes tienen importantes mejoras respecto a los filtros analógicos, porque al estar diseñados en una FPGA no tiene problemas de miss-match, ni son necesarios los

- voltajes de bias, tienen un coste muy inferior, y una gran flexibilidad.
- Gracias a la escalabilidad de la cóclea diseñada, y al estar el número de canales sólo limitado por el número de elementos lógicos, el diseño presentado puede migrarse a FPGAs con mayor capacidad, incluyendo en una sola FPGA adicionalmente capas de procesado y clasificación.
  - Añadir la funcionalidad de los OHCs de cóclea biológica, que se encargan de realizar un control automático de la ganancia de los canales en función de la potencia de la señal de entrada. Desarrollando en este sentido modelos de cócleas activas, las cuales son capaces de adaptar dinámicamente su funcionamiento acorde con las características del sonido con el que está siendo excitada.
- En el apartado de reconocimiento de sonidos, continuando con el enfoque basado en convoluciones, planteamos futuras líneas de desarrollo tanto enfocadas a las aplicaciones prácticas de sistema, como a su mejora y ampliación de funcionalidades:
    - Realizar transcripción musical de un solo instrumento a partir de una melodía con varios instrumentos. Mostrando así capacidad de discriminación de distintos sonidos, así como su posterior clasificación
    - Realizar el reconocimiento de instrumentos mediante la red neuronal pulsante de convoluciones. Desarrollando un sistema de clasificación más generalista, capaz de aprender a reconocer las características únicas de cada instrumento.
    - Realizar reconocimientos relacionados con el habla. En esta línea se propone el desarrollo de sistemas capaces de reconocer fonemas, así como a un hablante en particular, y características propias de un idioma.
    - Implementación de modelos de localización. Basándonos en el “retraso” de la llegada del sonido a los oídos, se puede determinar

la coordenada angular del origen del sonido. En los sistemas digitales esta es una tarea muy pesada, ya que se busca la correlación entre dos vectores de muestras de sonido, sin embargo, mediante el uso de una cóclea pulsante esta búsqueda se realiza correlando prácticamente evento a evento, presentando una carga computacional muy baja y unos tiempos de respuesta ínfimos.

- Procesamiento basado en el efecto Doppler. Gracias a la desviación en frecuencia de un sonido emitido por un objeto en movimiento al acercarse o alejarse, se podría aumentar el sistema de reconocimiento implementado para detectar la variación en frecuencia de los patrones de un sonido determinado. Representado la derivada de esta variación la velocidad relativa del objeto. Combinando este sistema con la localización, se podría implementar un sistema completo de posicionamiento de fuentes de sonido.
- En este trabajo ha presentado uno de los primeros sistemas de fusión de sensorial neuromórfico desarrollado. El desarrollo de sistemas que combinen visión y audición nos abre una gran cantidad de posibilidades de aplicación en la industria, como por ejemplo su integración en robots autónomos, uso en los dispositivos móviles, aplicaciones relacionadas con el control de la calidad, etc...



## 10. Bibliografía

- Amado, R.G. & Filho, J.V., 2008. Pitch detection algorithms based on zero-cross rate and autocorrelation function for musical notes. *ICALIP 2008 - 2008 International Conference on Audio, Language and Image Processing, Proceedings*, pp.449–454.
- Azarloo, A. & Farokhi, F., 2012. Automatic musical instrument recognition using K-NN and MLP neural networks. *Proceedings - 2012 4th International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2012*, pp.289–294.
- Barbancho, A.M. et al., 2012. Automatic Transcription of Guitar Chords and Fingering From Audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(3), pp.915–921.
- Barbaro, M. et al., 2002. A 100x100 pixel silicon retina for gradient extraction with steering filter capabilities and temporal output coding. *IEEE Journal of Solid-State Circuits*, 37, pp.160–172.
- Barlow, H.B.H., 1961. Possible principles underlying the transformation of sensory messages. In *Sensory Communication*. pp. 217–234.
- Barron, A.R., 1993. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39, pp.930–945.
- Baum, L.E. & Eagon, J.A., 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73, pp.360–364.
- Behringer, 2015. XENYX QX1002USB. Available at: <http://www.behringer.com/EN/Products/QX1002USB.aspx>.
- Berge, H.K.O. & Hafliger, P., 2007. High-Speed Serial AER on FPGA. *2007 IEEE International Symposium on Circuits and Systems*.
- Berner, R. et al., 2007. A 5 Meps \$100 USB2.0 Address-Event Monitor-Sequencer Interface. *2007 IEEE International Symposium on Circuits and Systems*.
- Boahen, K., 1998. *Communicating neuronal ensembles between neuromorphic chips*, Available at: <http://www.springerlink.com/index/x85g465gn6k55033.pdf>.



- Boahen, K., 2000. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47, pp.416–434.
- Boahen, K., 1999. Retinomorphic chips that see quadruple images. *Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*.
- Boahen, K.A., 2000. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47, pp.416–434.
- Camuñas-Mesa, L. et al., 2011. A  $32 \times 32$  pixel convolution processor chip for address event vision sensors with 155 ns event latency and 20 Meps throughput. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(4), pp.777–790.
- Camuñas-Mesa, L. et al., 2012. An event-driven multi-kernel convolution processor module for event-driven vision sensors. *IEEE Journal of Solid-State Circuits*, 47(2), pp.504–517.
- Camuñas-Mesa, L. et al., 2008. Fully digital AER convolution chip for vision processing. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 652–655.
- Camuñas-Mesa, L. et al., 2010. On scalable Spiking ConvNet Hardware for Cortex-Like Visual Sensory processing systems. , 3, pp.249–252.
- Cassidy, A.S. et al., 2013. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. *Proceedings of the International Joint Conference on Neural Networks*.
- Cerezuela-Escudero, E. et al., 2013. Spikes monitors for FPGAs, an experimental comparative study. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 179–188.
- Chakrabartty, S. & Liu, S.C., 2010. Exploiting spike-based dynamics in a silicon cochlea for speaker identification. In *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*. pp. 513–516.
- Chan, V., Liu, S.C. & van Schaik, A., 2007. AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(1), pp.48–59.
- Chan, V., Schaik, A. van & Liu, S.-C.L.S.-C., 2006. Spike response properties of an AER EAR. *2006 IEEE International Symposium on Circuits and Systems*.
- Chan, V.Y.S., Jin, C.T. & van Schaik, A., 2012. Neuromorphic audio-visual sensor fusion on a sound-localizing robot. *Frontiers in Neuroscience*.

- Conde, C., Orbe, E., Diego, I.M. De, et al., 2011. Bio-inspired event based motion detection for traffic safety in a close-real automotive environment. In *Proceedings - 2011 IEEE Electronics, Robotics and Automotive Mechanics Conference, CERMA 2011*. pp. 120–125.
- Conde, C., Orbe, E., De Diego, I.M., et al., 2011. Event based visual codification in automotive environments. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. pp. 1261–1266.
- CoolRunner2, CoolRunner2 CPLD Family Datasheet. Available at:  
[http://www.xilinx.com/support/documentation/data\\_sheets/ds090.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds090.pdf).
- Cope, B. et al., 2005. Have GPUs made FPGAs redundant in the field of Video Processing? In *Proceedings - 2005 IEEE International Conference on Field Programmable Technology*. pp. 111–118.
- Cope, B., 2006. Implementation of 2D Convolution on FPGA , GPU and CPU. *Imperial College Report*, pp.2–5. Available at:  
[http://cas.ee.ic.ac.uk/people/btc00/index\\_files/Convolution\\_filter.pdf](http://cas.ee.ic.ac.uk/people/btc00/index_files/Convolution_filter.pdf).
- Costas-Santos, J. et al., 2007. An AER Contrast Retina with On-Chip Calibration. In *IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 3075–3078.
- Culurciello, E., Etienne-Cummings, R. & Boahen, K., 2003. An Address Event Digital Imager. *IEEE Journal of Solid-State Circuits Solid-State Circuits*, 38(2).
- CypressFX2, Cypress EZ-USB FX2 Data Sheet. Available at:  
[http://www.keil.com/dd/docs/datashts/cypress/cy7c68xxx\\_ds.pdf](http://www.keil.com/dd/docs/datashts/cypress/cy7c68xxx_ds.pdf).
- Domínguez-Morales, M. et al., 2011. On the designing of spikes band-pass filters for FPGA. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 389–396. Available at:  
[http://link.springer.com/chapter/10.1007/978-3-642-21738-8\\_50](http://link.springer.com/chapter/10.1007/978-3-642-21738-8_50).
- Dundur, R. et al., 2008. Digital Filter for Cochlear Implant Implemented on a Field-Programmable Gate Array. *Pwaset*, 33(7), pp.468–472.
- Farabet, C. et al., 2009. CNP: An FPGA-based processor for Convolutional Networks. In *FPL 09: 19th International Conference on Field Programmable Logic and Applications*. pp. 32–37.
- Farrugia, N. et al., 2008. Design of a Real-Time Face Detection Parallel Architecture Using High-Level Synthesis. *EURASIP Journal on Embedded Systems*, 2008, p.938256.
- Fasel, B., 2002. Robust face analysis using convolutional neural networks. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2, pp.40–43 vol.2.

- Fasnacht, D.B., Whatley, A.M. & Indiveri, G., 2008. A serial communication infrastructure for multi-chip address event systems. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 648–651.
- Fastl, H. & Zwicker, E., 2007. *Psychoacoustics: Facts and models*,
- Fletcher, H., 1940. Auditory patterns. *Reviews of Modern Physics*, 12, pp.47–65.
- Fletcher, H. & Munson, W.A., 1933. Loudness, its Definition, Measurement and Calculation. *Journal of the Acoustical Society of America*, 5, pp.82–108. Available at: <http://psycnet.apa.org/?fa=main.doiLanding&uid=1934-01463-001>.
- Fragniere, E. & Vittoz, E., 1998. *Analogue VLSI emulation of the cochlea*. Available at: <papers2://publication/uuid/2659F43E-8D68-4627-96B9-57EF244123EB>.
- Freeman, J.A. & Skapura, D.M., 1991. *Neural Networks Algorithms , Applications and Programming Techniques*, Available at: <http://portal.acm.org/citation.cfm?id=128933&dl=>.
- Freeman, J.A. & Skapura, D.M., 1993. *Redes Neuronales: Algoritmos, aplicaciones y técnicas de programación*,
- Fujii, H. et al., 1996. Dynamical cell assembly hypothesis - Theoretical possibility of spatio-temporal coding in the cortex. *Neural Networks*, 9, pp.1303–1350.
- Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, pp.193–202.
- Fukushima, K. & Wake, N., 1991. Handwritten alphanumeric character recognition by the neocognitron. *IEEE Transactions on Neural Networks*, 2, pp.355–365.
- Funahashi, K.-I., 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2, pp.183–192.
- Gambin, I. et al., 2010. Digital cochlea model implementation using xilinx XC3S500E Spartan-3E FPGA. In *2010 IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2010 - Proceedings*. pp. 946–949.
- Gerstner, W. et al., 2002. Spiking neuron models: Single neurons, populations, plasticity. *Books.Google.Com*. Available at: [http://books.google.com/books?hl=en&lr=&id=Rs4oc7HfxIUC&oi=fnd&pg=PR11&dq=Spiking+Neuron+Models:+Single+Neurons,+Populations,+and+Plasticity&ots=2Od4xZhNYa&sig=Dm6YPQkIYAjQFyohC3Pmxr9\\_SoM](http://books.google.com/books?hl=en&lr=&id=Rs4oc7HfxIUC&oi=fnd&pg=PR11&dq=Spiking+Neuron+Models:+Single+Neurons,+Populations,+and+Plasticity&ots=2Od4xZhNYa&sig=Dm6YPQkIYAjQFyohC3Pmxr9_SoM).
- Glasberg, B.R. & Moore, B.C.J., 1990. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47, pp.103–138.

- Goldberg, D.H., Cauwenberghs, G. & Andreou, A.G., 2001. Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons. *Neural Networks*, 14, pp.781–793.
- Gomez-Rodriguez, F., Linares-Barranco, A., Miro, L., et al., 2007. AER Auditory Filtering and CPG for Robot Control. *2007 IEEE International Symposium on Circuits and Systems*, pp.1201–1204.
- Gomez-Rodriguez, F., Linares-Barranco, A., Paz, R., et al., 2007. AER image filtering. *Proceedings of SPIE*, 6592, pp.659207–659207–10.
- Gomez-Rodriguez, F. et al., 2006. AER tools for communications and debugging. *2006 IEEE International Symposium on Circuits and Systems*.
- Gomez-Rodriguez, F. et al., 2005. Two Hardware Implementations of the Exhaustive Synthetic AER Generation Method. *Lecture Note in Computer Science*, 3512, pp.534–540.
- Gómez-Rodríguez, F. et al., 2007. AER Auditory Filtering and CPG for Robot Control. *2007 IEEE International Symposium on Circuits and Systems*.
- Gómez-Rodríguez, F. et al., 2006. AER tools for communications and debugging. *2006 IEEE International Symposium on Circuits and Systems*.
- Guerrero-turrubiates, J.D.J. et al., 2014. Pitch Estimation For Musical Note Recognition Using Artificial Neural Networks. , pp.53–58.
- Häfliger, P., 2007. CAVIAR Hardware Interface Standards , Version 2.01. *Interface*. Available at: <http://heim.ifi.uio.no/~hafliger/CAVIAR/Consortiumstandards.pdf>.
- Häfliger, P., 2004. *Especificaciones protocolo asíncrono AER para el proyecto CAVIAR*,
- Hamilton, T.J. et al., 2008. An active 2-D silicon cochlea. *IEEE Transactions on Biomedical Circuits and Systems*, 2, pp.30–43.
- Hamilton, T.J., 2008. *Analogue VLSI Implementations of Two Dimensional, Nonlinear, Active Cochlea Models*.
- HODGKIN, A.L. & HUXLEY, A.F., 1939. Action Potentials Recorded from Inside a Nerve Fibre. *Nature*, 144, pp.710–711.
- Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4, pp.251–257.
- Hornik, K., Stinchcombe, M. & White, H., 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, pp.359–366.

- Hynna, K. & Boahen, K., 2001. Space-rate coding in an adaptive silicon neuron. *Neural Networks*, 14, pp.645–656.
- Iakymchuk, T. et al., 2014. An AER handshake-less modular infrastructure PCB with x8 2.5Gbps LVDS serial links. *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pp.1556–1559.
- Indiveri, G., 2000. A 2D neuromorphic VLSI architecture for modeling selective attention. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 4.
- Indiveri, G., Chicca, E. & Douglas, R., 2006. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17, pp.211–221.
- Indiveri, G., Chicca, E. & Douglas, R., 2006. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17, pp.211–221.
- Intel, 2002. Audio Codec '97. Available at: [http://www-inst.eecs.berkeley.edu/~cs150/Documents/ac97\\_r23.pdf](http://www-inst.eecs.berkeley.edu/~cs150/Documents/ac97_r23.pdf).
- Iwasa, K. et al., 2007. A Sound Localization and Recognition System using Pulsed Neural Networks on FPGA. *2007 International Joint Conference on Neural Networks*, pp.902–907. Available at: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4371078>.
- Jackel, D., Moeckel, R. & Liu, S.-C.L.S.-C., 2010. Sound recognition with spiking silicon cochlea and Hidden Markov Models. *Ph.D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on*.
- JAER, 2015. jaER open-source software project. Available at: <http://jaer.wiki.sourceforge.net/>.
- Jelinek, F., 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64, pp.532–556.
- Jimenez-Fernandez, A. et al., 2012. A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs. *Sensors*, 12, pp.3831–3856.
- Jimenez-Fernandez, A. et al., 2010. Building Blocks for Spikes Signal Processing. *International Joint Conference on Neural Networks*.
- Jimenez-Fernandez, A. et al., 2011. Simulating building blocks for spikes signals processing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 548–556.

- Jimenez-Fernandez, A. et al., 2009. Spike-based control monitoring and analysis with Address Event Representation. In *2009 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2009*. pp. 900–906.
- Jiménez-Fernández, A., 2010. *Diseño y evaluación de sistemas de control y procesamiento de señales basados en modelos neuronales pulsantes*. University of Seville.
- Jiménez-Fernández, A. et al., 2010. Neuro-inspired system for real-time vision tilt correction. In *IEEE International Symposium on Circuits and Systems*.
- Johnston & Wu, 1995. *Foundations of cellular neurophysiology*, Available at: <http://books.google.com/books?id=f8JnQgAACAAJ&pgis=1>.
- Jones, S. et al., 2000. Toward a digital neuromorphic pitch extraction system. *IEEE Transactions on Neural Networks*, 11, pp.978–987.
- Katsiamis, A.G., Drakakis, E.M. & Lyon, R.F., 2007. Practical gammatone-like filters for auditory processing. *Eurasip Journal on Audio, Speech, and Music Processing*, 2007.
- Kim, K.H. et al., 2009. An improved speech processing strategy for cochlear implants based on an active nonlinear filterbank model of the biological cochlea. *IEEE Transactions on Biomedical Engineering*, 56, pp.828–836.
- Kohonen, T., 1988. An introduction to neural computing. *Neural Networks*, 1, pp.3–16.
- Kumar, N. et al., 1998. An analog vlsi chip with asynchronous interface for auditory feature extraction. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45, pp.600–606.
- Lamberti, P.W. & Rodríguez, V., 2007. Desarrollo del modelo matemático de Hodgkin y Huxley en neurociencias. *Electroneurobiología*, 14(4), pp.31–60. Available at: [http://electroneubio.secyt.gov.ar/Lamberti-Rodriguez\\_Hodgkin-Huxley.htm](http://electroneubio.secyt.gov.ar/Lamberti-Rodriguez_Hodgkin-Huxley.htm).
- Lawrence, S. et al., 1997. Face recognition: a convolutional neural-network approach. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 8, pp.98–113.
- Lazzaro, J., Mead, C. & Ismail, M., 1989. *Circuit Models of Sensory Transduction in the Cochlea*,
- Lazzaro, J. & Wawrzynek, J., 1995. A multi-sender asynchronous extension to the AER protocol. *Proceedings Sixteenth Conference on Advanced Research in VLSI*.
- LeCun, Y. et al., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, pp.2278–2323.
- LeCun, Y. et al., 1990. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems*. pp. 396–404. Available at:

- <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.5076>  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.5076&rep=rep1&type=pdf>.
- LeCun, Y. & Bengio, Y., 1995. Convolutional Networks for Images, Speech, and Time-Series. *The Handbook of Brain Theory and Neural Networks*.
- Leñero-Bardallo, J.A., Serrano-Gotarredona, T. & Linares-Barranco, B., 2009. A mismatch calibrated bipolar spatial contrast AER retina with adjustable contrast threshold. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 1493–1496.
- Leong, M.P., Jin, C.T. & Leong, P.H.W., 2003. An FPGA-based electronic cochlea. *Eurasip Journal on Applied Signal Processing*, 2003, pp.629–638.
- Lewis, M.A. et al., 2003. An in silico central pattern generator: Silicon oscillator, coupling, entrainment, and physical computation. *Biological Cybernetics*, 88, pp.137–151.
- Lewis, M.A. et al., 2001. Biomorphic Control of a Running Robot Leg using a Custom aVLSI CPG Chip. *Neurocomputing*, 38, pp.1409–1421.
- Lichtsteiner, P. & Delbruck, T., 2005. A 64x64 AER logarithmic temporal derivative silicon retina. In *2005 PhD Research in Microelectronics and Electronics - Proceedings of the Conference*. pp. 406–409.
- Lichtsteiner, P., Posch, C. & Delbruck, T., 2008. A 128 X 128 120 dB 15  $\mu$ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43, pp.566–576.
- Lichtsteiner, P., Posch, C. & Delbruck, T., 2006. A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*.
- Lichtsteiner, P., Posch, C. & Delbrück, T., 2008. A 128x128 120dB 15 $\mu$ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43, pp.566–576. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4444573>.
- Linares-Barranco, a. et al., 2010. On the AER convolution processors for FPGA. *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, pp.4237–4240.
- Linares-Barranco, A. et al., 2008. AER filtering using GLIDER: VHDL cellular automata description. In *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2008*. pp. 614–617.
- Linares-Barranco, A., 2003. *Estudio y Evaluación de Interfaces para conexión de Sistemas Neuromórficos mediante Address-Event-Representation*. Universidad de Sevilla.
- Linares-Barranco, A. et al., 2006. On algorithmic rate-coded AER generation. *IEEE Transactions on Neural Networks*, 17, pp.771–788.

- Linares-Barranco, A. et al., 2002. Software generation of address-event-representation for interchip images communications. In *IECON Proceedings (Industrial Electronics Conference)*. pp. 1915–1919.
- Linares-Barranco, A., Oster, M. & Cascado-Caballero, D., 2005. Inter-spike-intervals analysis of poisson like hardware synthetic AER generation. *Computational Intelligence and Bioinspired Systems; Lecture Notes in Computer Science*, 3512, pp.479–485.
- Liu, S.C. et al., 2010. Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms. In *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*. pp. 2027–2030.
- Liu, S.C. & Douglas, R., 2004. Temporal coding in a network of silicon integrate-and-fire neurons. *IEEE Transactions on Neural Networks*, 15, pp.1305–1314.
- Liu, W., Andreou, A.G. & Goldstein, M.H., 1992. Voiced-speech representation by an analog silicon model of the auditory periphery. *IEEE Transactions on Neural Networks*, 3, pp.477–487.
- Liu, W., Andreou, A.G. & Goldstein, M.H. J., 1991. An analog integrated speech front-end based on the auditory periphery. *IJCNN-91-Seattle International Joint Conference on Neural Networks*, ii.
- López Bascuas, L.E., 1997. *La percepción del habla: problemas y restricciones computacionales.*,
- Luján-Martínez, C. et al., 2007. Spike processing on an embedded multi-task computer: Image reconstruction. In *Proceedings of the 5th International Workshop on Intelligent Solutions in Embedded Systems, WISES 07*. pp. 15–26.
- Lyon, R., 1982. A computational model of filtering, detection, and compression in the cochlea. *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 7, pp.1282–1285.
- Lyon, R.F. & Mead, C., 1988. ANALOG ELECTRONIC COCHLEA. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36, pp.1119–1134.
- Maass, W. & Bishop, C.M., 1999. *Pulsed Neural Networks*, Available at: <http://www.amazon.com/dp/0262632217>.
- Mahendra, S.R., Patil, H. a. & Shukla, N.K., 2009. Pitch estimation of notes in Indian classical music. *Proceedings of INDICON 2009 - An IEEE India Council Conference*, pp.0–3.
- Mahowald, M., 1992. *VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function*. Available at: <http://caltechctr.library.caltech.edu/591/>.
- McConnell, T.H. & Hull, K.L., 2010. *Human Form, Human Function: Essentials of Anatomy & Physiology*,



- Mead, C., 1990. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78, pp.1629–1636.
- Meddis, R., 1988. Simulation of auditory-neural transduction: further studies. *The Journal of the Acoustical Society of America*, 83, pp.1056–1063.
- Meddis, R., 1986. Simulation of mechanical to neural transduction in the auditory receptor. *The Journal of the Acoustical Society of America*, 79, pp.702–711.
- Meddis, R., Hewitt, M. & Shackleton, T., 1990. Implementation details of a computation model of the inner hair-cell auditory-nerve synapse. *Journal of the Acoustical Society of America*, 87, pp.1813–1816.
- Miró-Amarante, L. et al., 2006. A LVDS serial AER link. In *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*. pp. 938–941.
- Miró-Amarante, L. et al., 2007. LVDS Serial AER LINK Performance. In *IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 1537–1540.
- Miró-Amarante, M.L., 2013. *Una aportación al procesado neuromórfico de audio basado en modelos pulsantes*. Universidad de Sevilla.
- Molina-Vilaplana, J., Feliu-Battle, J. & López-Coronado, J., 2007. A modular neural network architecture for step-wise learning of grasping tasks. *Neural Networks*, 20, pp.631–645.
- Moore, G.E., 1998. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86, pp.82–85.
- Morgado-Estevez, A., 2004. *Análisis y modelado de sistemas pulsantes bioinspirados basados en buses de altas prestaciones: Bus AER*. Universidad de Sevilla.
- Mugliette, C. et al., 2011. FPGA active digital cochlea model. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2011*. pp. 699–702.
- Neubauer, C., 1998. Evaluation of convolutional neural networks for visual recognition. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 9(4), pp.685–696.
- Newton, M.J. & Smith, L.S., 2011. Biologically-inspired neural coding of sound onset for a musical sound classification task. In *Proceedings of the International Joint Conference on Neural Networks*. pp. 1386–1393.
- Nielsen, A.B., Hansen, L.K. & Kjems, U., 2006. Pitch Based Sound Classification. *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 3.
- O'Connor, P. et al., 2013. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*.

- O'Shaughnessy, D., 2000. *Speech communications: human and machine*. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11442089>.
- Opal-Kelly, 2015. XEM6010 Xilinx Spartan-6 FPGA, USB 2.0. Available at: <https://www.opalkelly.com/products/xem6010/>.
- Oster, M., Douglas, R. & Liu, S.C., 2007. Quantifying Input and Output Spike Statistics of a Winner-Take-All Network in a Vision System. *2007 IEEE International Symposium on Circuits and Systems*.
- Patterson, R.D. et al., 1992. Complex sounds and auditory images. *Simulation*, 83, pp.429–446. Available at: <http://www.pdn.cam.ac.uk/groups/cnbh/research/publications/pdfs/Petal92ish.pdf>.
- Paz-Vicente, R. et al., 2008. Image convolution using a probabilistic mapper on USB-AER board. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 1056–1059.
- Paz-Vicente, R. et al., 2006. PCI-AER interface for neuro-inspired spiking systems. *2006 IEEE International Symposium on Circuits and Systems*.
- Paz-Vicente, R. et al., 2009. Synthetic retina for AER systems development. In *2009 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2009*. pp. 907–912.
- Paz-Vicente, R. et al., 2005. Test infrastructure for address-event-representation communications. ... and *Bioinspired Systems*, 3512, pp.518–526.
- Paz-Vicente, R., 2008. *Una aportación al procesamiento de la información visual mediante técnicas bioinspiradas*. University of Seville.
- Penrose, R., 1989. *The Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*, Oxford University Press.
- Penttinen, H., Siikonen, J. & Valimäki, V., 2005. Acoustic guitar plucking point estimation in real time. *Laboratory of Acoustic and Audio Signal Processing*, 2(1), pp.209–212.
- Perez-Peña, F. et al., 2013. Neuro-inspired spike-based motion: From dynamic vision sensor to robot motor open-loop control through spike-VITE. *Sensors (Switzerland)*, 13, pp.15805–15832.
- Pishdadian, F. & Nelson, J.K., 2013. On the transcription of monophonic melodies in an instance-based pitch classification scenario. *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting, DSP/SPE 2013 - Proceedings*, pp.222–227.
- Poliner, G. & Ellis, D., 2005. A Classification Approach to Melody Transcription. , (1).
- Poliner, G.E. & Ellis, D.P.W., 2006. A discriminative model for polyphonic piano transcription. *Eurasip Journal on Advances in Signal Processing*, 2007, pp.1–16.

- Qian, D. & Nian, Z., 2007. Classification of recorded musical instruments sounds based on neural networks. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing, CIISP 2007*, (Ciisp), pp.157–162.
- Rabiner, L. & Juang, B.H., 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3, pp.4–16. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/18428778>.
- Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, pp.257–286. Available at: [http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed&cmd=Retrieve&dopt=AbstractPlus&list\\_uids=18626@ieeejrns](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed&cmd=Retrieve&dopt=AbstractPlus&list_uids=18626@ieeejrns).
- Rakic, P., 1988. Specification of Cerebral Cortical Areas. *Science (New York, N.Y.)*, 241, pp.170–176.
- Rios, A. et al., 2012. Detection system from the driver's hands based on Address Event Representation (AER). In *Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications III - Proceedings of the International Symposium, CompIMAGE 2012*. pp. 7–11.
- Rios-Navarro, A. et al., 2014. Live Demonstration: Real-Time Motor Rotation Frequency Detection by Spike-Based Visual and Auditory Sensory Fusion on AER and FPGA. In *Artificial Neural Networks and Machine Learning –ICANN 2014 and Machine Learning –ICANN 2014*.
- Rios-Navarro, A. et al., 2015. Real-time motor rotation frequency detection with event-based visual and spike-based auditory AER sensory integration for FPGA. *First IEEE International Conference on Event-Based Control, Communication and Signal Processing (EBCCSP 2015)*.
- Ripley, B.D., 1994. Neural networks and related methods for classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56, pp.409–456. Available at: <http://www.jstor.org/stable/10.2307/2346118>.
- ROSENBLATT, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65, pp.386–408.
- Van Schaik, A., Chan, V. & Jin, C., 2009. Sound localisation with a silicon cochlea pair. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. pp. 2197–2200.
- Van Schaik, A. & Fragnière, E., 2001. Pseudo-voltage domain implementation of a 2-dimensional silicon cochlea. In *IEEE International Symposium on Circuits and Systems*. pp. 185–188. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=921277>.
- Van Schaik, A., Fragnière, E. & Vittoz, E.A., 1996. Improved Silicon Cochlea using Compatible Lateral Bipolar Transistors. In *Advances in Neural Information Processing Systems*. pp. 671–677.
- Scott, J., 2004. *Audio in the 21st Century*,

- Serrano-Gotarredona, R. et al., 2005. AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems. *Control*, 15, pp.2–9.
- Serrano-Gotarredona, R. et al., 2009. CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20, pp.1417–1438.
- Serrano-Gotarredona, R. et al., 2009. CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20, pp.1417–1438.
- Serrano-Gotarredona, R. et al., 2008. On real-time AER 2-D convolutions hardware for neuromorphic spike-based cortical processing. *IEEE Transactions on Neural Networks*, 19, pp.1196–1219.
- Serrano-Gotarredona, R. et al., 2007. The Stochastic I-Pot: A circuit block for programming bias currents. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 54, pp.760–764.
- Serrano-Gotarredona, T., Andreau, A.G. & Linares-Barranco, B., 1999. AER image filtering architecture for vision-processing systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46, pp.1064–1071.
- Shadlen, M.N. & Newsome, W.T., 1994. Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4, pp.569–579.
- Shepherd, G.M., 2004. *The Synaptic Organization of the Brain*, Available at: <http://www.amazon.com/dp/019515956X>.
- Shiraishi, H., 2003. *Design of an Analog VLSI Cochlea*. University of Sydney. Available at: <http://hdl.handle.net/2123/556>.
- Silviotti, M., 1991. *Considerations in analog VLSI Systems with Application to Field-Programmable Networks*. California Institute of Technology.
- Slaney, M., 1993. An efficient implementation of the Patterson-Holdsworth auditory filter bank. *Apple Computer Perception Group Tech Report*, 1, p.40.
- Stevens, S.S., 1937. A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, 8, p.185.
- Summerfield, C.D. & Lyon, R.F., 1992. ASIC implementation of the Lyon cochlea model. [*Proceedings*] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, 5.
- Thakur, C.S. et al., 2014. FPGA Implementation of the CAR Model of the Cochlea. In *IEEE International Symposium on Circuits and Systems*. pp. 1853–1856.
- Thibodeau, G. & Patton, K., 2012. *Estructura y función del cuerpo humano*, Elsevier.

- Thorpe, S.J., Brilhault, A. & Perez-Carrasco, J.A., 2010. Suggestions for a biologically inspired spiking retina using order-based coding. *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, pp.265–268.
- VMPK, Electronic piano VMPK. Available at: <http://vmpk.sourceforge.net/index.es.shtml>.
- Vogelstein, R.J. et al., 2008. A silicon central pattern generator controls locomotion in Vivo. In *IEEE Transactions on Biomedical Circuits and Systems*. pp. 212–222.
- Vogelstein, R.J. et al., 2006. Phase-dependent effects of spinal cord stimulation on locomotor activity. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 14, pp.257–265.
- Watts, L., 1992. *Cochlear Mechanics: Analysis and Analog VLSI*. California Institute of Technology.
- Watts, L. et al., 1992. Improved implementation of the silicon cochlea. *IEEE Journal of Solid-State Circuits*, 27, pp.692–700.
- Wen, B. & Boahen, K., 2009. A silicon cochlea with active coupling. *IEEE Transactions on Biomedical Circuits and Systems*, 3(6), pp.444–455.
- Westerman, W.C., Northmore, D.P. & Elias, J.G., 1997. Neuromorphic Synapses for Artificial Dendrites. *Analog Integrated Circuits and Signal Processing*, 13(1), pp.167–184.
- Xilinx-ML507, 2015. Virtex-5 FXT FPGA ML507 evaluation platform. Available at: <http://www.xilinx.com/products/boards-and-kits/hw-v5-ml507-uni-g.html>.
- Yang, Z. et al., 2006. A neuromorphic depth-from-motion vision model with STDP adaptation. *IEEE Transactions on Neural Networks*, 17, pp.482–495.
- Yu, T. et al., 2009. Periodicity detection and localization using spike timing from the AER EAR. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 109–112.
- Zaghloul, K.A. & Boahen, K., 2004a. Optic Nerve Signals in a Neuromorphic Chip I: Outer and Inner Retina Models. *IEEE Transactions on Biomedical Engineering*, 51, pp.657–666.
- Zaghloul, K.A. & Boahen, K., 2004b. Optic Nerve Signals in a Neuromorphic Chip II: Testing and Results. *IEEE Transactions on Biomedical Engineering*, 51, pp.667–675.
- Zamarreño-Ramos, C. et al., 2008. LVDS interface for AER links with burst mode operation capability. In *Proceedings - IEEE International Symposium on Circuits and Systems*. pp. 644–647.
- Zhu, Y. & Kankanhalli, M.S., 2006. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Transactions on Multimedia*, 8, pp.575–584.

- Zölzer, U., Sankarababu, S.V. & Möller, S., 2012. PLL-based pitch detection and tracking for audio signals. *Proceedings of the 2012 8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IHH-MSP 2012*, (6), pp.428–431.
- Zwicker, E., 1961. Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen). *The Journal of the Acoustical Society of America*, 33, p.248.



## 11. Anexo: componentes hardwares y resultados

*“El logro más impresionante de la industria del software es su continua anulación de los constantes y asombrosos logros de la industria del hardware”, Henry Petroski*

En este anexo se realizará una exposición de los diversos componentes hardware utilizados durante este trabajo, y que se han ido nombrando a lo largo de este document.

### 11.1. USB-AER

La tarjeta USB-AER se basa fundamentalmente en la FPGA Xilinx Spartan II, conectada a una memoria RAM estática de 2 MB, con tiempo de acceso de 12ns, estructurada en palabras de 32 bits. Tiene una tarjeta SD/MMC para almacenar el firmware y el contenido de la memoria RAM, de forma que la FPGA pueda configurarse de forma automática sin necesidad de estar conectado a un PC, mediante un microcontrolador Cynal C8051F320. Además, por medio del puerto USB y el microcontrolador puede reconfigurarse la FPGA desde un PC e intercambiar datos desde el PC. El microcontrolador Cynal C8051F320 está basado en un núcleo 8051 con un interfaz USB 1.1 Full Speed. Adicionalmente, un bus asíncrono compuesto por 8 líneas de datos, además de algunas señales adicionales de control, interconectan la FPGA con el microcontrolador para el intercambio de datos.

El rendimiento máximo de la interfaz USB es de 6 Mbits/seg (~187KEvents/s), lo que limita el uso de esta interfaz para las comunicaciones basadas en eventos entre el PC y la tarjeta. Por lo tanto, esta interfaz recibe la información de control o



mapas de bits desde el PC y utiliza estos mapas en las transformaciones de eventos AER.

Los 2MB de SRAM integrados se pueden ampliar con la ranura SD/MMC que incorpora. Por último, y lo más importante, posee dos puertos IDE paralelo (Rafael Serrano-Gotarredona et al. 2009), (Häfliger 2007) para la conexión del bus AER paralelo. En la fotografía de la Figura 11.1 se observan los componentes principales de esta placa.



Figura 11.1. Fotografía de la placa USB-AER.

Debido a esta variedad de conexiones, hacen de esta placa un componente muy atractivo para desarrollar un interesante conjunto de AER-Tools. Tiene dos modos de operación, atendiendo a la funcionalidad requerida:

- Dependiente de PC: el firmware de la FPGA se descarga a través del puerto USB y los comandos y los datos son comunicados de vuelta a través de USB.
- Independiente del PC: la tarjeta de memoria SD/MMC contiene el firmware de la FPGA almacenado y los comandos y datos que necesita el microcontrolador.

Entre los firmwares (funcionalidades) más utilizados para procesar eventos AER, pueden destacarse los siguientes:

- **Generador AER:** Un mapa de bits se descarga desde un PC. Este mapa es utilizado como conjunto de eventos AER, de manera que la placa los genere de forma sintética, gracias a la dirección y timestamp que identifican cada evento (Linares-Barranco et al. 2006). Uno de los métodos de generación posibles sigue una distribución de Poisson (Linares-Barranco et al. 2005).
- **AER Mapper:** hay firmware disponibles para hacer un mapeo 1:1 y 1:N (con N de 0 a 8). También está disponible una versión probabilística que asigna una probabilidad a cada uno de los posibles eventos de salida asociados a un evento de entrada.
- **AER Frame-Grabber:** dos tipos de firmware están disponibles: frame-grabber por USB (con imágenes de tamaño 32x32 y 64x64) y por VGA (con imágenes de tamaño 64x64 y 256x256, que utiliza una placa hija adicional: AER-VGA).
- **Datalogger y reproductor:** usa los 2MB de SRAM para capturar hasta 512Keventos con 16 bits de resolución para cada marca de tiempo, que son relativas entre ellas. Y también es capaz de reproducir una secuencia de eventos almacenados en la memoria SRAM. Esta secuencia se recibe en el PC a través del puerto USB.

## 11.2. USB-AERmini2

Esta tarjeta es un puente completo entre el bus AER y el bus USB de un PC (Berner et al. 2007). Este dispositivo permite secuenciar (reproducir) y monitorizar (capturar) tráfico AER con una resolución temporal de 1uSec o de 0.2uSec. Para capturar el tráfico AER posee conexiones IDE paralelo y conector romano. Puede ser insertada tanto como monitor entre dos dispositivos AER (modo pass-through), y puede ser usada en modo terminal. En la Figura 11.2 se observa una fotografía de la placa, destacando los puertos que tiene.

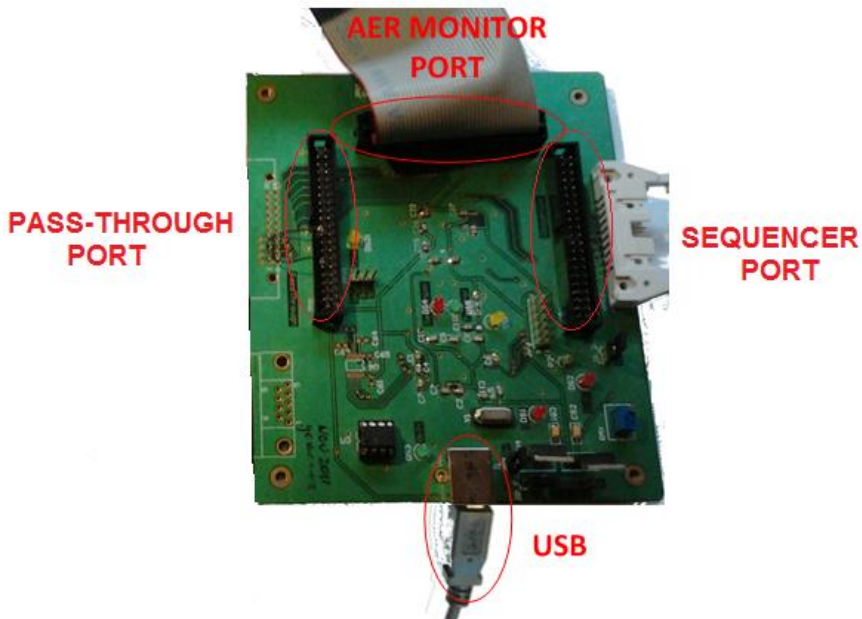


Figura 11.2. Fotografía de la tarjeta USBAERmini2, destacando los puertos que tiene

La arquitectura de la USBAERmini2 está basada en un microcontrolador y una CPLD. El microcontrolador, fabricado por Cypress y de la familia FX2LP (CypressFX2 n.d.), incluye un puerto USB2.0 high-speed (velocidad máxima de 480Mbits). La CPLD ha sido fabricada por Xilinx, en particular pertenece a la familia Coolrunner 2 (CoolRunner2 n.d.), y tiene 256 macroceldas. En la CPLD hay implementadas cuatro FSM, para manejar los puertos AER (tanto de envío como de recepción), generar marcas de tiempo (timestamps), y para leer/escribir las FIFOs USB del FX2LP.

En la Figura 11.3 se muestra el diagrama de bloques de la USBAERmini2. Para generar los timestamps se usa un contador de 14bits. Cada vez que se desborda, manda un mensaje al PC, el cual incrementa otro contador en el PC de 18bits, teniendo así un timestamp con una resolución de 32bits. La CPLD tiene conectado un reloj de 30MHz, obteniendo una tasa de monitorización de 6 mega-eventos AER por segundo de pico, pudiendo mantenerse a una frecuencia constante de 4.5 mega-eventos, aunque estas tasas están limitadas por la capacidad del PC.

La máxima frecuencia del secuenciador es de 3.75 mega-eventos. Este dispositivo fue diseñado para ser simple y barato de fabricar, así como simple de utilizar, es puramente plug-and-play. Podemos conectar a una cadena AER varias USBAERmini2, pudiendo sincronizarlas entre ellas para obtener timestamps sincronizados.

Está disponible el software jAER, que es un proyecto de código libre escrito en java y el cual, entre otras aplicaciones, sirve de interfaz en dispositivos AER, permitiendo la visualización y procesamiento de la información AER. Las clases del jAER son accesibles desde MATLAB, pudiendo así estimular dispositivos AER y capturar información AER fácilmente (JAER 2015).

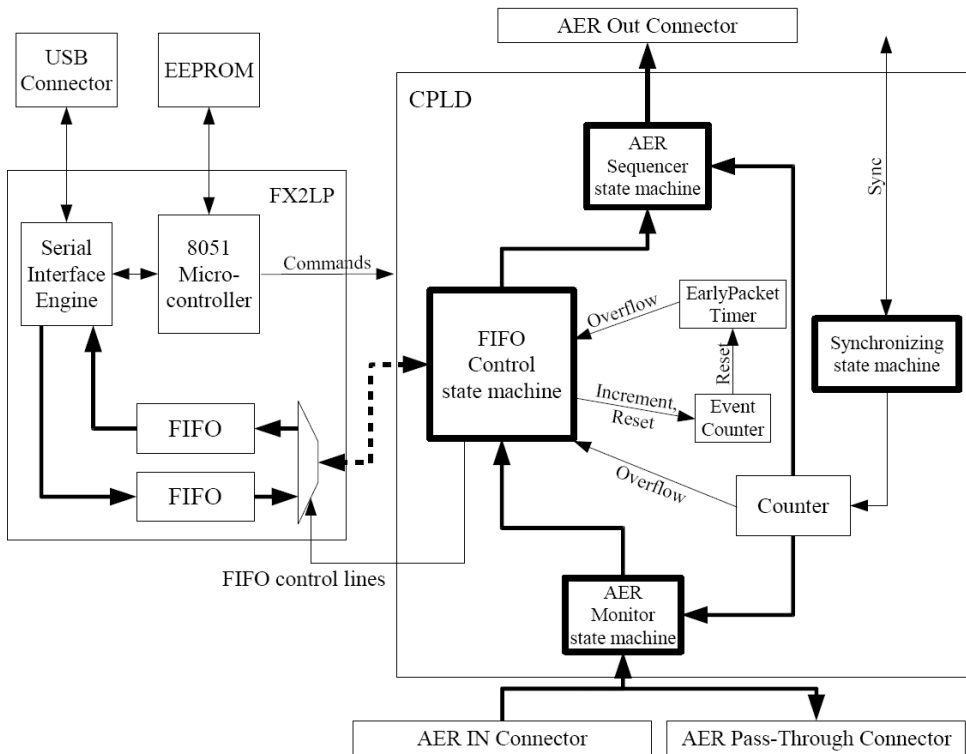


Figura 11.3. Diagrama de bloque de la USBAERMini2

En modo de secuenciación, las direcciones AER y los intervalos de tiempo entre eventos son leídos de la FIFO interna que posee y almacenados en los registros. Los intervalos de tiempo entre eventos se añaden a la marca de tiempo del último evento. Esto se hace para obtener una marca de tiempo absoluta. Debido a que la CPLD no tiene memoria RAM interna, los eventos se leen de la memoria FIFO, sólo uno por uno. Los eventos se envían desde el ordenador principal a la FIFO tan rápido como es posible y luego esperan en la FIFO hasta que se son recogidos por la CPLD. En el modo de monitorización, los eventos AER son almacenados en la FIFO interna y la CPLD los va recogiendo y mandando a través de la interfaz USB con dirección al ordenador.

### 11.3. Virtex-5 FXT FPGA ML507

En una plataforma de evaluación fabricado por Xilinx. Está construida alrededor de una FPGA de Xilinx Virtex-5 XC5VFX70T y permite diseños hardware y software.

Resumen de las características:

- Dos Flash PROMs (32Mb cada una) para almacenar la configuración del dispositivo.
- Memoria DDR2 SODIMM de 256MB con 64 bits de ancho.
- 8 switches, 8 LEDs, botones de pulsación y un encoder rotativo.
- Codec de audio estéreo AC'97, conector de micrófono y de auriculares
- Display LCD con 16 caracteres por dos líneas.
- Entrada de video VGA y salida de video con conector DVI
- Conector RJ-45 y transceptor Ethernet 10/100/100 velocidades.
- Puerto de configuración JTAG.
- 32 conectores de expansión I/O, de los que usamos 18 para el bus AER.

Desarrollamos una placa adaptador entre el conector que trae esta placa con el conector IDC40 que usamos para la comunicación siguiendo el protocolo AER.

Esta placa ha supuesto el núcleo central de la primera parte del trabajo porque sirve para contener el NAS requerido. Si bien no hemos usado todos los periféricos ni conexiones que ésta aporta, hemos aprovechado el 99% de la capacidad de su FPGA.

Por lo tanto, resumiendo, los componentes de esta placa utilizados han sido:

- La FPGA como núcleo del sistema.
- El bus de expansión para realizar la comunicación de los eventos AER de salida.
- El códec de audio AC'97 para digitalizar el audio y tanto el conector de micrófono y de auriculares.
- De forma marginal, los leds y botones como elementos adicionales de control y monitorización.

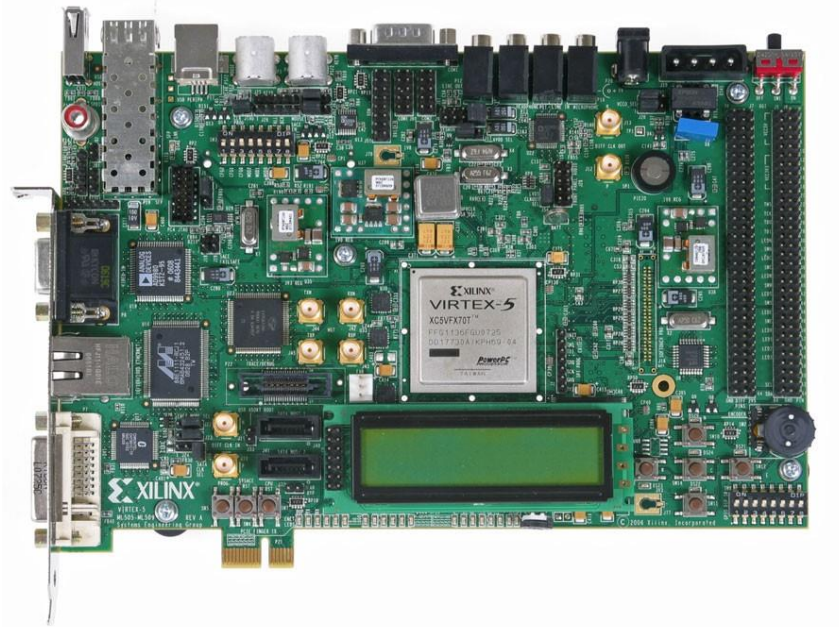


Figura 11.4. Kit de evaluación ML507 de Xilinx con la FPGA Virtex-5

### 11.4. Resultados de los experimentos de evaluación de los sistemas de reconocimiento para tonos puros

En la Tabla 11.1 se muestran los resultados de la red neuronal MLP ante el reconocimiento de ocho tonos puros en presencia de ruido blanco usando el NASv1 y en la Tabla 11.2 para el NASv2. Estas tablas contienen los datos mostrados en la Figura 7.6.

Las Tabla 11.3 y Tabla 11.4 contienen los resultados de la red ConvNet para el reconocimiento de ocho tonos puros en presencia de ruido blanco para el NASv1 y el NASv2 respectivamente. Estas tablas contienen los datos mostrados en la Figura 7.21

Tabla 11.1. Porcentaje de aciertos del sistema de reconocimiento de tonos puros mediante la red neuronal MLP usando el NASv1

SNR (dBW)	Frecuencias de tono puro (Hz)							
	130.813	174.614	261.626	349.228	523.251	698.456	1046.5	1896.91
32,189	91,66	87,5	90,47	89,58	93,75	95,74	95,83	91,67
18,326	93,7	89,5	83,3	89,5	87,5	89,3	93,7	89,5
10,217	91,6	91,6	55,7	87,5	87,5	91,4	87,5	91,6
4,463	6,25	89,5	77,8	93,7	87,5	89,3	4,16	89,5
0	91,48	87,5	91,48	91,48	93,75	100	91,66	89,58
-3,646	0	89,5	4,76	89,5	93,7	89,3	93,7	89,5
-6,729	93,7	87,7	88,0	87,5	93,0	0	87,5	89,5
-9,4	93,7	89,5	2,38	91,6	87,5	0	87,5	91,6
-11,756	89,5	7,14	85,7	0	93,7	91,4	87,5	89,5
-13,863	93,7	91,6	85,7	89,5	87,5	95,3	89,5	91,6
-15,769	95,8	0	80,5	89,5	0	91,4	89,5	93,7
-17,509	95,8	87,5	85,7	89,5	89,5	93,6	89,5	93,7
-19,11	95,8	89,5	88,0	89,5	91,6	95,7	0	89,5
-20,592	6,25	91,6	85,7	92,8	83,3	89,3	91,6	89,5
-21,972	2,08	89,5	83,3	93,7	87,5	91,4	0	89,5
-23,263	91,8	87,5	85,7	89,6	89,5	89,3	95,8	97,7
-24,475	4,76	0	0	91,67	2,04	4,25	2,08	89,58

-25,619	100	87,5	83,3	91,6	88,0	91,4	91,6	91,6
-26,7	93,7	89,5	83,3	89,5	87,5	89,3	93,7	89,5
-27,726	91,6	92,8	90,4	89,5	95,8	91,4	95,8	89,5
-28,702	95,8	87,5	85,7	89,6	87,5	95,7	91,6	87,5
-29,632	93,7	89,5	0	89,5	87,5	2,27	89,5	93,7
-30,521	87,7	4,16	11,1	0	89,5	89,3	93,7	6,25
-31,372	93,7	91,6	0	89,5	91,6	87,2	88,0	87,5
-32,189	0	87,5	85,7	89,7	85,4	89,3	93,7	95,8
-32,973	95,8	87,5	2,38	0	0	65,9	93,7	4,16
-33,728	95,8	0	80,5	89,5	0	91,4	89,5	0
-34,455	4,76	0	0	91,6	2,04	4,25	2,08	89,5
-35,157	6,66	7,66	35,7	48,500	0	91,489	87,500	4,1666
-35,835	6,25	0	0	4,17	0	32,48	4,76	0

Tabla 11.2. Porcentaje de aciertos del sistema de reconocimiento de tonos puros mediante la red neuronal MLP usando el NASv2

SNR (dBW)	Frecuencias de tono puro (Hz)							
	130.813	174.614	261.626	349.228	523.251	698.456	1046.5	1896.91
32,189	91,48	89,58	91,30	91,48	89,58	95,55	97,91	91,3
18,326	91,4	93,6	93,6	93,6	91,6	93,3	89,5	89,3
10,217	93,6	95,1	0	95,3	89,5	93,3	89,5	91,4
4,463	95,7	89,5	89,3	91,4	89,5	93,3	93,7	93,6
0	91,4	87,5	91,4	91,4	93,7	100	91,6	89,5
-3,646	89,3	87,5	91,4	89,31	87,5	93,3	93,7	97,8
-6,729	91,4	93,7	89,3	91,4	91,6	93,3	2,08	89,3
-9,4	95,7	89,5	89,3	91,4	89,5	93,3	93,7	93,6
-11,756	95,7	87,5	93,6	93,6	95,8	93,3	93,7	93,6
-13,863	93,6	95,8	89,3	89,3	87,5	95,5	89,5	95,7
-15,769	0	91,6	89,3	89,3	87,5	95,5	87,5	93,6
-17,509	95,7	89,5	89,3	91,4	89,5	93,3	93,7	93,6
-19,11	95,7	89,5	89,3	91,4	87,5	93,3	91,6	95,7
-20,592	93,6	87,5	89,3	0	87,5	93,3	89,5	91,4
-21,972	89,3	95,4	89,3	85,1	85,4	97,7	2,08	93,6



-23,263	91,4	87,5	89,3	89,3	91,6	93,3	89,5	97,5
-24,475	95,74	87,5	89,36	93,62	87,50	93,33	89,58	89,36
-25,619	93,6	87,5	89,3	93,6	89,5	93,3	89,5	95,7
-26,7	95,7	0	87,2	85,1	87,5	93,3	89,5	95,7
-27,726	95,7	97,9	91,4	91,4	87,5	93,3	93,7	93,6
-28,702	93,6	89,5	42,5	72,3	89,5	93,3	89,5	0
-29,632	91,4	87,5	87,5	91,4	87,5	91,1	87,5	95,7
-30,521	0	100	0	0	0	0	0	0
-31,372	0	0	93,6	91,4	72,3	93,3	0	89,3
-32,189	0	0	100	0	0	0	0	0
-32,973	91,4	87,5	89,3	91,4	95,8	0	0	91,4
-33,728	0	95,1	0	95,3	89,5	0	89,5	91,4
-34,455	0	100	0	0	0	0	0	22,8
-35,157	0	0	0	100	0	0	2,08	89,3
-35,835	0	0	91,48	0	0	0	0	0

Tabla 11.3. Tasa de aciertos (en tanto por 1) del sistema de reconocimiento de tonos puros ConvNet usando el NASv1

SNR (dBW)	Frecuencias de tono puro (Hz)							
	130.813	174.614	261.626	349.228	523.251	698.456	1046.5	1896.91
Inf.	1	1	1	1	0,996	1	1	0,912
32,189	0,926	1	1	1	0,986	1	1	0,909
18,326	0,926	1	1	1	0,988	1	1	0,847
10,217	0,94	1	1	1	0,986	1	1	0,846
4,463	0,939	0,981	1	1	0,985	1	1	0,79
0	0,934	0,964	1	1	0,978	1	1	0,768
-3,646	0,975	0,982	0,982	0,979	0,973	0,975	1	0,362
-6,729	0,938	0,966	0,967	0,883	0,971	0,478	1	0,065
-9,4	0,945	0,934	0,959	0,733	0,937	0,166	1	0,022
-11,756	0,941	0,714	0,824	0,651	0,796	0,019	1	0
-13,863	0,941	0,333	0,714	0,508	0	0	1	0
-15,769	0,944	0,1366	0	0,362	0	0	1	0
-17,509	0,943	0,0579	0	0,302	0	0	1	0
-19,11	0,943	0,028	0	0,145	0	0	1	0

-20,592	0,845	0,027	0	0,107	0	0	1	0
-21,972	0,702	0,022	0	0,015	0	0	1	0
-23,263	0,509	0,013	0	0,015	0	0	1	0
-24,475	0,065	0,013	0	0,028	0	0	1	0
-25,619	0,011	0,013	0	0	0	0,019	1	0
-26,7	0,011	0,013	0	0	0	0	1	0
-27,726	0	0,012	0	0,012	0	0	1	0
-28,702	0	0,012	0	0	0	0	1	0
-29,632	0	0	0	0	0	0	1	0
-30,521	0	0,012	0	0	0	0	1	0
-31,372	0	0	0	0	0	0	1	0
-32,189	0	0,011	0	0	0	0	1	0
-32,973	0	0,011	0	0	0	0	1	0
-33,728	0	0,011	0	0	0	0	1	0
-34,455	0	0	0	0	0	0	1	0
-35,157	0	0,011	0	0	0	0	1	0
-35,835	0	0	0	0	0	0	1	0

Tabla 11.4. Tasa de aciertos (en tanto por 1) del sistema de reconocimiento de tonos puros ConvNet usando el NASv2

SNR (dBW)	Frecuencias de tono puro (Hz)							
	130.813	174.614	261.626	349.228	523.251	698.456	1046.5	1896.91
Inf.	1	1	1	1	1	1	1	1
32,189	1	1	1	1	1	1	1	1
18,326	1	1	0,981	0,989	1	0,967	0,968	0,977
10,217	1	0,981	0,981	0,985	0,979	0,965	0,956	0,977
4,463	1	0,963	0,981	0,98	0,979	0,964	0,956	0,954
0	0,982	0,981	0,962	0,961	0,98	0,9618	0,978	0,975
-3,646	1	0,981	0,962	0,96	0,96	0,95744681	0,9565	0,954
-6,729	1	0,964	0,962	0,961	1	0,795	0,956	0,933
-9,4	0,982	0,982	0,962	0,943	0,96	0,682	0,957	0,956
-11,756	1	0,95	0,962	0,961	0,960	0,685	0,957	0,955
-13,863	0,982	0,965	0,963	0,943	0,961	0,622	0,957	0,955
-15,769	0,982	0,964	0,945	0,962	0,961	0,612	0,957	0,956

Anexos: componentes hardware y resultados

-17,509	0,982	0,965	0,963	0,962	0,961	0,568	0,958	0,956
-19,11	0,948	0,948	0,946	0,925	0,961	0,558	0,97	0,957
-20,592	0,982	0,965	0,964	1	0,961	0,622	0,958	0,957
-21,972	0,965	0,949	0,964	0,927	0,962	0,607	0,938	0,957
-23,263	0,932	0,965	0,947	0,892	0,944	0,603	0,938	1
-24,475	0,936	0,966	0,947	0,909	0,944	0,622	0,96	0,958
-25,619	0,932	0,966	0,948	0,91	0,923	0,622	0,94	0,958
-26,7	0,866	0,966	0,948	0,892	0,945	0,637	0,96	0,939
-27,726	0,883	0,97	0,949	0,842	0,945	0,655	0,882	0,94
-28,702	0,836	0,95	0,949	0,912	0,928	0,685	0,941	0,943
-29,632	0,786	0,967	0,949	0,877	0,946	0,66	0,923	0,941
-30,521	0,803	0,951	0,928	0,912	0,946	0,543	0,923	0,921
-31,372	0,737	0,951	0,966	0,844	0,947	0,561	0,891	0,92
-32,189	0,709	0,951	0,950	0,879	0,931	0,596	0,849	0,961
-32,973	0,688	0,951	0,966	0,883	0,947	0,56	0,905	0,941
-33,728	0,737	0,967	0,951	0,833	0,948	0,517	0,905	0,923
-34,455	0,796	0,967	0,951	0,783	0,949	0,537	0,909	0,942
-35,157	0,836	0,967	0,951	0,933	0,948	0,555	0,907	0,867
-35,835	0,754	0,934	0,95	0,9	0,944	0,519	0,872	0,943