

Computación aplicada a la música II

PRINCIPIOS DE AUDIO DIGITAL

Ken C. Pohlmann

Traducción y revisión técnica

ANTONIO MÍNGUEZ OLIVARES
Catedrático de E.U.

FRANCISCO JAVIER TABERNERO GIL
Profesor Titular de E.U.

JORGE GRUDMAN ISLA
Profesor Titular de E.U.

Departamento de Ingeniería Audiovisual y Comunicaciones
Escuela Universitaria de Ingeniería Técnica de Telecomunicaciones
Universidad Politécnica de Madrid



**McGraw-Hill
Profesional**

MADRID • BUENOS AIRES • CARACAS • GUATEMALA • LISBOA • MÉXICO
NUEVA YORK • PANAMÁ • SAN JUAN • SANTAFÉ DE BOGOTÁ • SANTIAGO • SÃO PAULO
AUCKLAND • HAMBURGO • LONDRES • MILÁN • MONTREAL • NUEVA DELHI • PARÍS
SAN FRANCISCO • SIDNEY • SINGAPUR • ST. LOUIS • TOKIO • TORONTO

*Para
Leslie Hope
y su isla huracanada
salpicada con montañas de blancas cumbres*

PRINCIPIOS DE AUDIO DIGITAL

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.

DERECHOS RESERVADOS © 2002, respecto a la cuarta edición en español,
por McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
Edificio Valrealty, 1.ª planta
Basauri, 17
28023 Aravaca (Madrid)

Traducido de la cuarta edición en inglés de
Principles of Digital Audio
Copyright © MM, por McGraw-Hill, Inc.
ISBN: 0-07-134819-0

ISBN: 84-481-3625-X
Depósito legal: 34.803-2002

Editor: Antonio García Brage
Diseño de cubierta: Magma Estudio, E. Quirós, S. L.
Compuesto en: FER Fotocomposición, S. A. Bocángel, 45. 28028 Madrid
Impreso en: EDIGRAFOS, Volta, 2. Pol. Ind. San Marcos. 28096 Getafe (Madrid)

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

ACERCA DEL AUTOR

Ken C. Pohlmann es profesor de la Universidad de Miami en Coral Gables, Florida, y director del programa de Ingeniería Musical en la Escuela de Música de la misma Universidad. Ha impartido diversos cursos de pregrado y de postgrado sobre acústica, psicoacústica, audio digital, audio digital avanzado, audio en Internet y postproducción de audio, y en 1986 implantó en los Estados Unidos el primer programa Master de Ingeniería Musical.

El señor Pohlmann también es presidente de los Laboratorios Hammer, compañía dedicada a la investigación, el desarrollo y a la evaluación de nuevos equipos tecnológicos en el campo del audio. También realiza servicios de consultoría en el diseño de equipos de audio digital, como experto en el campo del audio y en el litigio de patentes. Entre sus clientes se encuentran, entre otras, las siguientes empresas: Alpine Electronics, Bertlesmann Music Group, Blockbuster Entertainment, Baker & McKeenzie, BMW, Canadian Broadcasting Corporation, Daimler-Chrysler, Fish & Neave, Ford Motor Company, Fujitsu Ten, Harman International, Hughes Electronics, Hyundai Motor Company, IBM, Lexus, Microsoft, Motorola, Nippon Columbia, Onkyo America, Philips, RealNetworks, Recording Industry Association of America, Sensormatic, Sonopress, Sony, TDK, Time Warner, Toyota, United Technologies y el Departamento de Justicia de los Estados Unidos. El señor Pohlmann es ingeniero en ingeniería eléctrica (grados «Bachelor Science» y «Master Science») por la Universidad de Illinois en Urbana-Champaign.

El señor Pohlmann es cofundador de Microcomputer Arts, Inc. (1980), de International Bussiness Information Systems Inc. (1982) y de U.S. Digital Disc Corporation (1985). Ha presidido la conferencia internacional sobre Audio Digital patrocinada por la Sociedad de Ingeniería del Audio (AES), celebrada en Toronto (1989), y ha copresidido la conferencia internacional sobre Audio en Internet celebrada en Seattle (1997), también patrocinada por la Sociedad de Ingeniería del Audio (AES). Ha sido propuesto dos veces para los premios que concede el Comité Ejecutivo del AES (1989 y 1998) y una vez al premio honorífico que se otorga a los miembros significativos de la Sociedad (1990), por su trabajo docente y como autor. En 1991, el señor Pohlmann fue elegido miembro del Comité Ejecutivo del AES y en 1993 también fue elegido Vicepresidente del AES de la Región Este de los Estados Unidos y de la Región de Canadá. En 1992 fue propuesto para el Premio Philip Frost que otorga la Universidad de Miami por su dedicación docente e investigadora.

El señor Pohlmann es autor del libro *Principles of Digital Audio* (McGraw-Hill) y del libro *The Compact Disc Handbook* (ediciones A-R), coautor del libro *Writing for New Media* (John Wiley and Sons) y editor y coautor del libro *Advanced Digital Audio* (Howard W. Sams). Desde 1982, ha escrito más de 1.400 artículos en revistas de audio, entre las que se encuentran *Audio magazine*, *dB magazine*, *Car Stereo Review*, *CD Review*, *Electronics Australia*, *Guitar Player magazine*, *IEEE Spectrum*, *Journal of the Audio Engineering Society*, *Laserdisk Profesional*, *Mix magazine*, *PC magazine*, *Sound & Vision*, *Scientific American*, *Stereo Review* y *Video magazine*. Es columnista de la revista *Car Stereo Review* y editor y columnista de *Sound & Vision*.

CONTENIDO

| | |
|--|-----------|
| Prólogo a la 4. ^a edición | xvii |
| Capítulo 1. Sonido y números | 1 |
| Física del sonido | 1 |
| Nivel de presión sonora | 3 |
| Armónicos | 4 |
| Fundamentos digitales | 5 |
| Sistemas numéricos | 5 |
| El sistema binario | 7 |
| Códigos binarios | 8 |
| Códigos binarios ponderados | 8 |
| Códigos binarios sin ponderación | 9 |
| Complemento a dos | 10 |
| Álgebra booleana | 12 |
| Analógico versus digital | 16 |
| Capítulo 2. Fundamentos del audio digital | 19 |
| Muestreo discreto | 19 |
| Teorema del muestreo | 20 |
| La frecuencia Nyquist | 21 |
| <i>Aliasing</i> | 25 |
| Prevención | 27 |
| Cuantificación | 28 |
| Relación señal a ruido | 29 |
| Distorsión de cuantificación | 32 |
| Otras arquitecturas | 34 |
| <i>Dither</i> | 35 |
| Tipos de <i>dither</i> | 38 |
| Conclusión | 43 |
| Capítulo 3. Grabación digital de audio | 45 |
| Modulación por codificación de pulsos | 45 |
| Generador de <i>dither</i> | 48 |
| Filtro paso-bajo | 48 |
| Círculo de muestreo y retención | 51 |

| | |
|--|------------|
| Convertidor analógico digital | 54 |
| Convertidor A/D de aproximaciones sucesivas | 57 |
| Conversor A/D con sobremuestreo | 59 |
| Procesado en grabación | 61 |
| Códigos de canal | 63 |
| Códigos simples | 66 |
| Códigos de grupo | 68 |
| Aplicaciones de los códigos | 72 |
| Capítulo 4. Reproducción del audio digital | 75 |
| Proceso de reproducción | 75 |
| Conversión digital/analógica | 77 |
| Convertidor D/A de resistencias ponderadas | 80 |
| Convertidor D/A con red en escalera R-2R | 81 |
| Distorsión de cruce | 83 |
| Conversión de alto número de bits | 84 |
| Circuito de muestreo y retención de la salida | 85 |
| Filtro recuperador | 89 |
| Respuesta al impulso | 90 |
| Filtro recuperador digital | 92 |
| Conformador de ruido | 98 |
| Tratamiento de salida | 100 |
| Métodos de codificación alternativos | 100 |
| Codificación en «coma flotante» | 101 |
| Sistemas de conversión de coma flotante en bloques | 103 |
| Sistemas de codificación no uniforme | 104 |
| Companders de ley μ y ley A | 105 |
| PCM diferencial | 106 |
| Modulación delta | 107 |
| Modulación delta adaptativa (ADM) | 109 |
| Compander con modulación delta predictiva | 111 |
| Modulación adaptativa diferencial (ADPCM) | 111 |
| Corrección de la base de tiempos | 113 |
| Inestabilidad temporal («jitter») | 114 |
| Diagrama de ojos | 115 |
| «Jitter» en el muestreo y en la conexión entre equipos | 116 |
| «Jitter» en los soportes de almacenamiento | 117 |
| «Jitter» en la transmisión de datos | 119 |
| «Jitter» en convertidores | 122 |
| Capítulo 5. Corrección de errores | 127 |
| Fuentes de error | 128 |
| Objetivos de la corrección de errores | 131 |
| Detección de errores | 131 |
| Paridad simple | 132 |
| ISBN | 134 |
| Código de chequeo de redundancia cíclica | 135 |
| Códigos de corrección | 140 |
| Códigos de bloque | 141 |
| Códigos convolucionales | 147 |

| | |
|---|------------|
| Entrelazado | 149 |
| Intercalación cruzada | 151 |
| Códigos Reed-Solomon | 151 |
| Código Reed-Solomon de intercalación cruzada | 157 |
| Ocultación de errores | 161 |
| Interpolación | 162 |
| Silenciamiento de la salida | 163 |
| Duplicación | 164 |
| Capítulo 6. Almacenamiento en cinta magnética | 165 |
| Ancho de banda de grabación | 165 |
| La cinta magnética digital | 166 |
| Interferencia entre símbolos | 168 |
| Grabación magnética longitudinal | 169 |
| Grabación magnética vertical | 170 |
| Magnetófonos de cabezas estacionarias | 171 |
| El formato DASH | 173 |
| Magnetófonos de cabezas giratorias | 176 |
| Funcionamiento de un grabador de vídeo | 177 |
| El procesador digital de audio | 180 |
| Formatos de vídeo profesional | 182 |
| Multipista digital modular ADAT | 185 |
| Magnetófonos multipista modulares Tascam | 188 |
| Ventajas e inconvenientes de los sistemas de cabezas giratorias | 191 |
| Capítulo 7. Digital Audio Tape (DAT) | 193 |
| La casete DAT | 193 |
| Modos del DAT | 195 |
| Diseño electrónico del DAT | 196 |
| Cabezas rotativas | 197 |
| Formato de la cinta | 200 |
| Grabación con cambio de acimut | 205 |
| Seguimiento automático de pista (ATF) | 205 |
| Modulación ocho a diez (ETM) | 207 |
| Corrección de errores en el DAT | 208 |
| Serial Copy Management System (SCMS) | 212 |
| DAT pregrabados | 213 |
| DAT profesional | 213 |
| Interfases del DAT | 213 |
| Código de tiempo en el DAT | 214 |
| Características de los DAT profesionales | 216 |
| Capítulo 8. Almacenamiento en disco óptico | 217 |
| Principios ópticos | 217 |
| Difracción | 220 |
| Resolución de los sistemas ópticos | 222 |
| Polarización | 223 |
| Diseño de los medios de almacenamiento ópticos | 225 |
| Prestaciones de los soportes ópticos | 227 |
| Almacenamiento óptico no borrrable | 228 |
| Almacenamiento óptico de sólo lectura | 228 |
| Almacenamiento óptico de escritura única | 230 |

| | |
|--|------------|
| Almacenamiento óptico borrrable | 232 |
| Registro magneto-óptico..... | 232 |
| Almacenamiento por cambio de fase | 235 |
| Almacenamiento óptico borrrable en polímeros coloreados..... | 236 |
| El audio digital en el cine..... | 237 |
| Capítulo 9. El sistema de disco compacto..... | 241 |
| Invención del disco compacto | 241 |
| Una aproximación al disco compacto | 242 |
| Codificación del disco | 244 |
| Características del disco | 244 |
| Codificación de los datos | 247 |
| Diseño óptico del reproductor | 252 |
| Captador óptico | 252 |
| Sistema de autoenfoco..... | 254 |
| Sistema de seguimiento de pista..... | 256 |
| Detector de un solo haz | 257 |
| Control del detector | 259 |
| Diseño eléctrico del reproductor | 259 |
| Demodulación EFM | 261 |
| Detección y corrección de errores | 263 |
| Tratamiento de salida | 264 |
| Subcódigo | 264 |
| Fabricación del disco | 268 |
| Masterización en cinta..... | 269 |
| Masterización del disco | 269 |
| Electromoldeo..... | 271 |
| Fabricación de las copias del disco | 271 |
| Formatos de CD alternativos | 273 |
| CD-ROM | 274 |
| CD-R..... | 279 |
| CD-RW | 286 |
| CD-MO | 287 |
| CD-i | 288 |
| Photo CD..... | 290 |
| CD+G y CD+MIDI | 292 |
| CD-3 | 293 |
| Vídeo CD..... | 293 |
| Super Audio CD | 293 |
| Modulación DSD..... | 296 |
| Diseño del reproductor | 298 |
| Capítulo 10. Codificación perceptual..... | 301 |
| Psicoacústica..... | 301 |
| Fisiología del oído humano y bandas críticas | 305 |
| Umbral de audición y enmascaramiento | 307 |
| Fundamentos de la codificación perceptual | 313 |
| Codificación de la reducción de datos..... | 315 |
| Codificación sub-banda | 317 |
| Codificación por transformada | 322 |
| Bancos de filtros | 323 |

| | |
|--|------------|
| Estándar de audio MPEG-1 | 325 |
| Modelos psicoacústicos | 330 |
| Layer I | 331 |
| Ejemplo de algoritmo MPEG-1 layer I | 333 |
| Layer II | 336 |
| Layer III..... | 338 |
| Estándar de audio MPEG-2 | 342 |
| MPEG-2 AAC | 344 |
| Codificador AC-3 (Dolby Digital)..... | 347 |
| Codificador apt-X100 | 356 |
| Evaluación de la calidad en la codificación perceptual..... | 356 |
| Conclusiones..... | 361 |
| Capítulo 11. DVD | 363 |
| Introducción..... | 363 |
| Especificaciones físicas del DVD..... | 366 |
| Fabricación y reproducción del disco..... | 369 |
| Codificación de los datos | 371 |
| El UDF («Universal Disc Format») | 376 |
| DVD-Vídeo..... | 378 |
| Codificación del vídeo..... | 378 |
| Codificación del audio..... | 380 |
| Características de la reproducción..... | 381 |
| Protección anticopia en el DVD-Vídeo..... | 383 |
| Resumen de términos para desarrolladores de DVD-Vídeo | 385 |
| DVD-Audio | 389 |
| Opciones de codificación y número de canales | 390 |
| Contenido del disco | 395 |
| Protección anticopia | 398 |
| Información para desarrolladores de DVD-Audio | 398 |
| Compresión de datos | 401 |
| Codificación con entropía..... | 401 |
| Sistemas de compresión de los datos de audio | 405 |
| Sistema de codificación MLP (Meridian Lossless Packing)..... | 410 |
| Otros formatos de DVD..... | 413 |
| Capítulo 12. El MiniDisc | 417 |
| Características generales del sistema | 417 |
| Formato de la señal..... | 419 |
| Diseño del disco | 420 |
| Detector óptico | 423 |
| Almacenamiento magneto-óptico y modulación del campo..... | 426 |
| Sistema ATRAC de reducción de datos..... | 429 |
| Masterización del disco y proceso de fabricación..... | 436 |
| Capítulo 13. Interconexión | 439 |
| Interfaces de audio..... | 439 |
| Interconexión SDIF-2..... | 440 |
| Interface digital óptico multicanal de Alesis | 442 |
| Interface Profesional AES3 (AES/EBU)..... | 442 |
| Bloque de <i>status</i> de canal..... | 444 |
| Implementación | 448 |

| | |
|--|------------|
| Interface multicanal (MADI) AES10 | 449 |
| Interconexión de consumo S/PDIF..... | 450 |
| Sistema de protección contra copias | 453 |
| Señal de referencia de audio digital AES11 | 455 |
| Canales de datos de usuario AES18 | 457 |
| Control de equipos de audio (AES24) | 458 |
| Convertidores de frecuencia de muestreo..... | 459 |
| Interconexión por fibra óptica | 461 |
| Sistemas de fibra óptica..... | 462 |
| Implementación de la fibra óptica..... | 463 |
| Interconexión e instalación..... | 468 |
| Costes de los sistemas de fibra óptica..... | 470 |
| Capítulo 14. Audio en el PC | 471 |
| Interfaces y Bus del PC | 471 |
| IEEE 1394 «FireWire» | 473 |
| Universal Serial Bus (USB)..... | 476 |
| Tarjetas de sonido | 479 |
| Interfaz MIDI | 480 |
| Síntesis musical | 480 |
| Procesado de sonido envolvente..... | 481 |
| Audio Codec '97 (AC '97)..... | 482 |
| API multimedia de Windows y DirectX..... | 484 |
| MMX | 485 |
| Formatos de archivo | 486 |
| Extracción de audio digital..... | 489 |
| Grabación en disco duro..... | 492 |
| Características del sistema..... | 492 |
| Unidades de disco duro | 494 |
| Aplicaciones de software de audio en el PC | 498 |
| Edición en disco duro..... | 500 |
| Audio para estaciones de trabajo de vídeo..... | 502 |
| Capítulo 15. Audio en Internet | 505 |
| Redes de ordenadores y transferencia de archivos..... | 505 |
| Servicios telefónicos..... | 506 |
| RDSI..... | 508 |
| Líneas de Suscripción Asimétrica Digital (ADSL)..... | 509 |
| Redes de ordenadores | 511 |
| ATM..... | 513 |
| MediaNet | 514 |
| Internet y el audio en Internet | 515 |
| MP3 | 518 |
| SDMI | 520 |
| Transmisión continua de audio («streaming audio»)..... | 521 |
| Codificador de audio RealAudio G2..... | 523 |
| Difusión de audio por Internet | 526 |
| MPEG-4..... | 527 |
| Interactividad | 527 |
| Codificación del audio..... | 528 |
| Criptografía y marcas de agua (watermarking)..... | 531 |

| | |
|--|------------|
| Capítulo 16. Difusión digital de radio y televisión | 535 |
| Funcionamiento de los satélites..... | 535 |
| Transmisión digital | 539 |
| Satélites de emisión directa | 541 |
| Radio digital | 542 |
| Métodos de transmisión..... | 542 |
| El espectro | 543 |
| Reducción de datos..... | 545 |
| Consideraciones técnicas..... | 545 |
| Eureka 147/DAB | 549 |
| Radio digital en-banda..... | 553 |
| La radio digital en EE.UU. | 555 |
| Radio digital Lucent (LDR) | 559 |
| Algoritmo de codificación perceptual de audio (PAC) | 561 |
| Radio vía satélite | 563 |
| Televisión digital (DTV)..... | 563 |
| Vídeo en MPEG | 564 |
| Televisión digital ATSC..... | 573 |
| Estándar ATSC..... | 574 |
| Implantación del sistema DTV | 579 |
| Capítulo 17. Procesado digital de señales | 583 |
| Fundamentos del procesado digital de señales..... | 583 |
| Aplicaciones DSP | 584 |
| Sistemas discretos | 585 |
| Linealidad e invarianza en el dominio del tiempo | 585 |
| Respuesta al impulso y convolución | 586 |
| Números complejos | 589 |
| Transformadas | 590 |
| Círculo unidad y región de convergencia..... | 594 |
| Polos y ceros..... | 595 |
| Elementos de un DSP | 597 |
| Filtros digitales | 598 |
| Filtros FIR | 599 |
| Filtros IIR | 603 |
| Aplicaciones de los filtros | 606 |
| Fuentes de error | 607 |
| Circuitos integrados DSP..... | 610 |
| Un procesador DSP multimedia..... | 613 |
| Programación de los DSP..... | 614 |
| Aplicaciones de los DSP | 619 |
| Retardo digital | 620 |
| Reverberación digital..... | 621 |
| Consolas de mezcla digital..... | 623 |
| Corrección de la respuesta de los altavoces | 626 |
| Eliminación del ruido | 627 |
| Capítulo 18. Conversión sigma-delta y conformación de ruido | 631 |
| Conversión sigma-delta | 631 |
| Modulación sigma-delta | 633 |
| Patrones de secuencias en la señal de ruido («ruido en vacío»)..... | 640 |

| | |
|---|---------|
| Conversión D/A-1bit con conformación de segundo orden..... | 640 |
| Conversión D/A multibit con conformación de tercer orden..... | 643 |
| Conversión D/A multibit con conformación de cuarto orden..... | 646 |
| Conversión A/D sigma-delta..... | 649 |
| Modulador A/D sigma-delta..... | 651 |
| Filtro de diezrado..... | 653 |
| Chip conversor A/D sigma-delta..... | 657 |
| Chip conversor A/D-D/A sigma-delta..... | 658 |
| Conformación de ruido sin sobremuestreo..... | 659 |
| Conformación espectral de ruido psicoacústica..... | 661 |
| Enmascaramiento binario de datos..... | 666 |
| Conclusiones..... | 667 |
| Apéndice. El teorema del muestreo..... | 669 |
| Bibliografía..... | 673 |
| Índice..... | 699 |

PRÓLOGO A LA 4.^a EDICIÓN

En una convención del AES (Audio Engineering Society), celebrada en Nueva York en 1984, impartí una conferencia sobre los fundamentos del audio digital con material del curso que acababa de inaugurar en la Universidad de Miami. Después de terminar la conferencia un gran número de asistentes me preguntaron dónde podían conseguir más información de este nuevo y maravilloso tema. Un astuto editor notó la respuesta del público y, en un estilo realmente mefistofélico, me persuadió para que escribiera un libro sobre el audio digital. Después de todo, me explicó, *es simplemente poner todo esto en un papel*. Inocentemente acepté esta *ganga* y el libro *Principles of Digital Audio* salió a la luz en 1985. La edición del libro coincidió con un tremendo auge en el desarrollo de la tecnología del audio digital. Virtualmente, cada segmento de la industria del entretenimiento, las telecomunicaciones, el procesado digital de señales y el sector de los ordenadores sufrieron una gran revolución, apareciendo muchas industrias completamente nuevas. El audio digital realmente ha cambiado nuestras vidas.

La popularidad del audio digital encajó perfectamente con una demanda deseosa de explicaciones sobre el tema. Los ingenieros de audio, y los estudiantes que desean serlo en un futuro, beben vorazmente cada trago de información que son capaces de encontrar, impulsando sin duda el desarrollo de esta ciencia. La primera edición dio paso a la segunda, y ésta a su vez a la tercera. Mi pacto con el editor había echado raíces y mi alma estaba realmente atrapada por sus tentáculos. Servilmente escribí una edición tras otra, pues los cheques de los royalties me consolaban.

Cuando publiqué la tercera edición me prometí a mí mismo que nunca escribiría una cuarta edición. Pero imprudentemente rompí mi promesa. En particular, me dije que si la tecnología continuaba sorprendiéndonos con su inventiva, y había gente deseosa de aprenderla, era necesario escribir una nueva edición. Claramente, con el agujero que había dejado en mi palabra y con la continua expansión que sufre la tecnología de audio, la siguiente edición era inevitable. Con la llegada de esta cuarta edición, he aprendido una lección. No voy a hacer más promesas, ni en un sentido ni en otro, sobre la quinta edición.

Los lectores de las ediciones anteriores no estarán demasiado sorprendidos de las actualizaciones que he incorporado en esta edición. La esencia fundamental en temas como el muestreo discreto no ha cambiado, pero nuestra comprensión de dichos temas sí que lo ha hecho. Análogamente, la importancia relativa de los temas evoluciona continuamente; los avances en la tecnología merman la magnitud de algunos de ellos, mientras que simultáneamente se magnifican otros. De la misma forma que la historia se reescribe continuamente a sí misma, nuestra comprensión de los principios científicos también evoluciona. Además, e incluso más significativamente, en los últimos cinco años desde la última edición se han desarrollado toda una serie de nuevas aplicaciones de audio. Más significativamente, la tecnología de los ordenadores personales ha utilizado con entusiasmo el audio digital, Internet ha llegado a ser todo un coloso de audio y, después de años de desarrollo, el DVD y la DTV han visto por fin la luz.

Al examinar esta nueva edición encontrará que una tercera parte del texto ha sido revisado y cómo se han añadido nuevos párrafos e ilustraciones. Se han eliminado una serie de temas con objeto de conservar el número de páginas del libro (y el coste), otros temas han sido ampliados significativamente y se han incorporado una serie de temas nuevos, tanto con un enfoque teórico como práctico. Los lectores encontrarán una gran cantidad de material nuevo y actual (y muchos nuevos acrónimos). Dentro de los temas relacionados con el audio PC se pueden encontrar: IEEE 1394, USB, AC '97 y DirectX. Dentro de los temas relacionados con Internet: MP3, SDMI y RealNetworks G2 *streaming audio*. En los temas relacionados con los sistemas de baja tasa binaria: MPEG-2, AAC, MPEG-4, Dolby Digital y PAC. En los temas de DVD se incluye: DVD-Vídeo, DVD-Audio, DVD grabable, UDF y MLP. En la radiodifusión de audio y vídeo se incluye: ATSC DTV, AM-IBOC y FM-IBOC (incluyendo los prototipos para la radio digital de EE.UU. y LDR). Aparecen nuevos temas dentro del disco compacto: CD-R, CD-RW y el Super Audio CD. Algunos temas son difíciles de encasillar, como los estándares AES, el jitter, las tarjetas de sonido, la compresión de datos, la extracción de datos de audio y el *watermarking* (marca de agua). Asimismo, toda la bibliografía ha sido actualizada y ampliada.

En la medida de lo posible se ha conservado una redacción de estilo «amigable al lector», pero la gran profundidad que se ha dado a alguno de los temas da lugar a una lectura algo más sofisticada. Asimismo, inevitablemente, las tecnologías cada vez más avanzadas como el DVD requieren descripciones más complejas. Obviamente, el lector puede seleccionar el material de acuerdo a su nivel y necesidad. También, el amplio espectro que abarcan todos estos temas debería satisfacer a un gran abanico de profesionales y estudiantes dentro del mundo del audio. En este sentido, se ha dado un especial énfasis a las aplicaciones prácticas del audio digital.

Una cosa no ha cambiado. Este libro no pretende ser un compendio de hechos ni un tratado científico avanzado. Es un texto introductorio que intenta proporcionar explicaciones de la forma más lúcida posible y diferenciar entre lo que es simple información de lo que es realmente comprensión. En otras palabras, este libro es realmente una herramienta de aprendizaje que ha sido escrita por una persona que disfruta a ambos lados del atril.

El material de este libro es la consecuencia del trabajo de muchos pioneros y lectores dentro del campo de la tecnología del audio. Tenemos una gran deuda con aquellos ingenieros altamente creativos, y sobre todo a su esfuerzo, que dedican su vida al desarrollo de la ciencia del audio digital. Claramente, su visión de la potencialidad de esta joven ciencia ha cambiado profundamente nuestra industria y nuestra sociedad.

Una nota final: de la misma forma que Fausto disfrutó de la recompensa de su pacto con el diablo, espero que encuentre en esta última entrega del libro una gran satisfacción en su lectura. Si usted es de los que se dedica a diseñar, enseñar o aprender los sistemas más avanzados de esta tecnología, sabrá exactamente de lo que estoy hablando.

KEN C. POHLMANN
Coral Gables, Florida

Capítulo 1

SONIDO Y NÚMEROS

El audio digital es una tecnología altamente sofisticada. Su desarrollo está ligado al avance de nuevas tecnologías en distintos campos de ingeniería. Los fundamentos se conocen desde 1920, pero su uso comercial ha tenido que esperar al desarrollo tecnológico de la década de los setenta. Dada la complejidad implícita que existe en el audio digital, es conveniente comenzar con los conceptos básicos en los que se fundamenta. Concretamente, en este capítulo se ven los procedimientos numéricos para codificar la información representada por una señal de audio.

FÍSICA DEL SONIDO

Sería un error abordar el estudio del audio digital ignorando su aspecto acústico, el principal objetivo de diseño en esta tecnología. La música es un fenómeno acústico. Se propaga en el aire, tanto si se genera de forma electrónica o directamente mediante instrumentos musicales. Al final, lo que realmente importa es el sonido que produce y la percepción que se tiene de ella. Es, por tanto, necesario repasar brevemente la naturaleza del sonido.

La acústica estudia las ondas sonoras en su generación, propagación y recepción, así como las circunstancias que producen estos tres procesos cuando se crea una perturbación acústica. Por ejemplo, cuando se golpea la membrana de un tambor se produce una perturbación acústica en el entorno circundante. Esta perturbación es el sonido que produce el tambor. El mecanismo es así de simple: se golpea la membrana y ésta vibra en uno y otro sentido, produciendo compresiones y rarefacciones de las moléculas de aire que la rodean. La perturbación acústica, en definitiva, genera una serie de variaciones de presión acústica, respecto a la presión atmosférica de equilibrio. A las áreas con un mínimo desplazamiento de las partículas del aire se las denomina nodos, y a las áreas con un desplazamiento máximo de las partículas se las denomina anti-nodos. Este desplazamiento es muy pequeño; en una conversación normal, el desplazamiento de las partículas es del orden de micras (una millonésima de metro), mientras que el desplazamiento producido por una gran cantidad de gente puede llegar a ser del orden de milímetros.

El sonido se propaga a través de los sucesivos desplazamientos de las moléculas del aire, producidos por la perturbación acústica inicial. En otras palabras, las moléculas de aire chocan unas con otras propagando, lejos de la fuente, la energía acústica inicial. La propagación acústica se lleva a cabo, por tanto, a través de perturbaciones acústicas locales que van originando las moléculas del aire en sus desplazamientos. Estos desplazamientos se producen en la misma

dirección en la que se propaga la energía acústica; por consiguiente, la propagación del sonido se produce de forma longitudinal. Un receptor (por ejemplo, el diafragma de un micrófono) colocado en el campo acústico, se moverá de acuerdo con la presión que se ejerza sobre él. Cabe señalar, además, que la propagación es más fácil cuando el medio es más denso. Por ejemplo, las ondas sonoras viajan más fácilmente en el agua que en el aire.

Mediante transductores es posible acceder al medio acústico; estos dispositivos son capaces de transformar distintos tipos de energía. Pueden actuar tanto como generadores o como receptores acústicos. Por ejemplo, un tambor transforma la energía mecánica, producida al golpear con las baquetas, en energía acústica. Un micrófono responde a la energía acústica produciendo energía eléctrica. Un altavoz invierte el proceso, transforma la energía eléctrica en energía acústica.

Los cambios de presión de las vibraciones sonoras pueden ser producidos de forma periódica o no periódica. Un violín mueve el aire en uno y otro sentido de una forma periódica. (En la práctica, las vibraciones son cuasi-periódicas como las realizadas en un vibrador.) Sin embargo, cuando se golpea un címbalo no se realiza de forma periódica. El sonido es, por tanto, no periódico. Un ciclo es una secuencia completa de una vibración periódica, desde una rarefacción hasta una compresión. La frecuencia de la onda sonora, medida en hercios (Hz), es el número de ciclos que pasan por un mismo punto durante un segundo. Por ejemplo, un violín en un concierto puede generar ondas sonoras que se repiten 440 veces por segundo; su frecuencia será, por tanto, de 440 Hz. Por otra parte, el recíproco de la frecuencia es el período: el intervalo de tiempo que transcurre durante un ciclo. En la naturaleza, la frecuencia puede tener un gran margen de valores, desde los más bajos (10^{-5} Hz), debidos a los cambios en la presión atmosférica, hasta valores extremadamente altos (10^{22} Hz), producidos por los rayos cósmicos. El sonido tiene un margen de frecuencias bastante inferior, aproximadamente desde los 20 Hz hasta los 20 kHz; margen que corresponde a la percepción auditiva. Los dispositivos de audio se diseñan, por tanto, para que respondan en este margen de frecuencias.

La longitud de onda es la distancia que recorre el sonido en un ciclo completo de cambio de presión, y es la medida física de la longitud de un ciclo. La longitud de onda se puede calcular fácilmente si tenemos en cuenta que la velocidad del sonido es relativamente estable (aproximadamente 340 m/s), simplemente dividiendo ésta por la frecuencia. Un cálculo rápido nos puede dar idea del margen de valores de la longitud de onda. Por ejemplo, a una frecuencia de 20 kHz le corresponde una longitud de onda de 1,7 cm, mientras que a una frecuencia de 20 Hz le corresponde una longitud de onda de 17 metros. No existe ningún transductor (incluido el oído) que sea capaz de generar o recibir linealmente todo este margen de longitudes de onda. La respuesta en frecuencia de un sistema acústico normalmente no es plana, y está limitada a un determinado margen de frecuencias. Este margen, en el que responde un sistema, se define como el ancho de banda del sistema. Cuando dos ondas sonoras coinciden en tiempo en sus variaciones positivas y negativas de presión se dice que están en fase. Cuando esta coincidencia es opuesta se dice que las ondas están en oposición de fase. Un desplazamiento de fase ocurre cuando existe un retardo/adelanto temporal entre ambas ondas. Cuando se combinan dos ondas con un determinado desplazamiento de fase se genera una nueva onda de presión, resultante de una interferencia constructiva o destructiva. Idealmente, la onda resultante será nula cuando se combinen en oposición de fase y se reforzará cuando se combinen en fase.

El sonido se curva en presencia de aberturas u obstáculos. A este efecto se le conoce por el nombre de difracción. La difracción depende de la longitud de onda; las grandes longitudes de onda se difractan más que las pequeñas longitudes de onda. Ésta es la razón de que el sonido de alta frecuencia sea mucho más directivo. Pruebe este experimento: coloque una revista enfrente de un altavoz y compruebe lo siguiente: las altas frecuencias no traspasan la barrera, mientras que las bajas (grandes longitudes de ondas) no sufren ninguna alteración y rodean el obstáculo.

El sonido también puede refractarse, esto es, curvarse debido a cambios de velocidad. Por ejemplo, el sonido se refracta cuando existen cambios de temperatura en un medio de propaga-

ción, alejándose de las zonas calientes y aproximándose a las zonas más frías. Concretamente, la velocidad del sonido en el aire aumenta 33,5 m/s por cada grado centígrado de temperatura. Este efecto lo conocen bien los instrumentistas de viento. Las diferencias de temperatura producen variaciones de frecuencia en el sonido, de hasta medio semitono. Ésta es la razón de que los instrumentos de viento deban calentarse previamente a la temperatura normal de trabajo.

La velocidad del sonido en el aire es relativamente pequeña, unos 340 m/s. El tiempo que tarda en propagarse el sonido se puede calcular dividiendo la distancia emisor-receptor entre la velocidad de propagación. Por ejemplo, en una distancia de 100 metros el tiempo de propagación es de 0,29 sg. Otro fenómeno que se produce en la propagación del sonido es la absorción. Las frecuencias altas se atenúan más rápidamente en el aire; un trueno se escucha como un gran impulso cuando se está cerca de la tormenta, mientras que cuando se está alejado se percibe como rugido, debido a la atenuación que han sufrido precisamente las frecuencias altas. La humedad produce una atenuación en la propagación de las ondas sonoras; concretamente, el aire húmedo absorbe mejor el sonido que el aire seco. Otro hecho curioso, el aire húmedo es menos denso que el aire seco (las moléculas de agua pesan menos que el nitrógeno y el oxígeno) y esta circunstancia hace que se incremente la velocidad del sonido.

Nivel de presión sonora

El término amplitud se emplea para describir el desplazamiento de la presión sonora, arriba y abajo, respecto a la presión atmosférica de equilibrio. En términos absolutos, la presión sonora es muy pequeña; si la presión atmosférica es de 0,044 kg/m², un sonido fuerte puede causar una desviación entre 4,40 y 4,41 kg/m². Sin embargo, el margen de valores desde el sonido más débil hasta el más fuerte es amplísimo (margen dinámico). De hecho, el oído humano (y por tanto los sistemas de audio) tiene un margen dinámico de miles de millones de veces con respecto al sonido más débil. Debido a esto, para expresar los niveles de presión sonora es mucho más conveniente utilizar una escala logarítmica. El decibelio (dB) emplea unidades logarítmicas, en base 10, para realizar este tipo de conversión. Un logaritmo en base 10 de un número es la potencia a la que debe elevarse 10 para obtener dicho número. Por ejemplo, un número difícil de manejar como 100.000.000 tiene un logaritmo (en base 10) de 8, ya que $10^8 = 100.000.000$. Específicamente, el decibelio se define como diez veces el logaritmo de una relación de potencias:

$$\text{Nivel de intensidad} = 10 \log \left(\frac{P_1}{P_2} \right) \text{ dB}$$

donde P_1 y P_2 son dos valores de potencia, acústica o eléctrica.

El nivel de intensidad se puede estandarizar colocando en el denominador del logaritmo un valor de referencia. En medidas acústicas, para expresar en dB el nivel de intensidad (L_I), es necesario utilizar la referencia de intensidad de 10^{-12} W/m² (vatios por metro cuadrado), correspondiente al umbral de audición. Así por ejemplo, el nivel de intensidad de un grupo de rock que genera 10 W/m² es:

$$\text{Nivel de intensidad} = 10 \log \left(\frac{P_1}{P_2} \right) = \left(\frac{10^1}{10^{-12}} \right) = 130 \text{ dB SPL}$$

Cuando se emplean relaciones de corrientes, tensiones o presiones sonoras (cantidades cuyas potencias al cuadrado son proporcionales a la potencia) la expresión anterior debe multiplicarse por 2.

El valor de referencia para el nivel de presión sonora (SPL) corresponde a una presión de 0,00002 dinas/cm² (pascuales). Este valor es el umbral de audición, es decir, el nivel más débil que puede percibirse, y es igual a 0 dB SPL. El umbral de dolor, el nivel más alto antes de producir molestia, es igual a 120 dB SPL. Los valores SPL se pueden presentar en una escala com-

parativa. El nivel de presión sonora dentro de una casa silenciosa puede estar en torno a los 35 dB SPL, en una calle con gran densidad de tráfico en torno a los 70 dB SPL y en las cercanías de un motor a reacción puede llegar a superar los 150 dB SPL. Los sonidos más débiles de una orquesta pueden oscilar alrededor de los 30 dB SPL, llegando a los 110 dB SPL en los sonidos más fuertes. Es decir, el rango dinámico puede alcanzar los 80 dB SPL.

La naturaleza logarítmica de los dB merece cierta reflexión. No son valores fácilmente reconocibles ya que no son medidas lineales. Así por ejemplo, dos motocicletas que produzcan cada una de ellas un nivel de intensidad (L_I) de 80 dB, la suma no produce un valor de 160 dB. La suma logarítmica de niveles se realiza de forma diferente. En este caso concreto se produce un incremento de 3 dB, con lo que el nivel de intensidad total sería de 83 dB. Si se hubiera utilizado una escala lineal para expresar la intensidad sonora, cada motocicleta estaría produciendo un valor de 0.0001 W/m^2 y la suma, ahora sí, sería el doble, 0.0002 W/m^2 .

Armónicos

El movimiento periódico más simple corresponde a una senoide; se manifiesta en osciladores simples como en péndulos o diapasones. Un movimiento periódico sinusoidal tiene únicamente una frecuencia fundamental. El resto de movimientos periódicos son complejos, presentan una frecuencia fundamental y una serie de frecuencias múltiplos de la anterior. Los movimientos no periódicos, como el sonido producido por el motor de una motocicleta, no tienen esta cualidad. Muchos instrumentos musicales son ejemplos típicos en los que puede comprobarse la relación que existe entre los armónicos y la frecuencia fundamental. Por ejemplo, un sonido con una frecuencia fundamental de 150 Hz producirá armónicos de 300 Hz, 450 Hz, 600 Hz, 750 Hz, etc.

Los armónicos que superan el umbral de audición determinan el timbre del sonido. El timbre depende de las amplitudes y fases de los armónicos. Por ejemplo, un violonchelo y una trompeta pueden tocar la misma nota con el mismo tono; sin embargo, el timbre de cada instrumento es diferente, ya que los armónicos de cada instrumento también son diferentes en amplitud y fase. Cuando un violonchelista quiere tocar la nota D4 lo que hace es pulsar la cuerda D. Esta cuerda realmente produce la nota D3 (una octava inferior), pero el instrumentista pulsa la cuerda justamente en su punto medio para anular el tono fundamental y los armónicos impares, y poder obtener así un tono con una octava superior (primer armónico par de D3) con unos armónicos diferentes (resto de armónicos pares de D3). En definitiva, el tono y el timbre cambian. El oído no siempre tiene la capacidad de diferenciar el timbre, sobre todo cuando los armónicos son de alta frecuencia. El primer armónico de un tono de 10 kHz es de 20 kHz. Muy pocas personas pueden percibir este armónico. Aun así, en una grabación el margen de frecuencias debe llegar hasta los 20 kHz (límite de audición) y preservar hasta este límite todos los armónicos que constituyen el timbre del sonido.

La naturaleza armónica de las ondas periódicas se demuestra por el teorema de Fourier. Este teorema establece que una onda periódica está compuesta de una serie armónica de senoideas; de forma recíproca, estas ondas periódicas pueden ser reconstruidas a partir de su serie completa de senoideas (fundamental y armónicos). Además, la onda periódica puede descomponerse en sus componentes sinusoidales para analizar su naturaleza espectral (en el dominio de la frecuencia). Esta descomposición se realiza a través de una transformación matemática conocida por el nombre de Transformada de Fourier. Por ejemplo, una onda cuadrada puede descomponerse en una componente fundamental y toda una serie infinita de armónicos impares. La Transforma inversa de Fourier (herramienta matemática recíproca de la anterior) permite realizar el proceso inverso.

Dada la evidente complejidad de las señales acústicas, sería inocente creer que la tecnología de audio (analógica o digital) está lo suficientemente desarrollada para registrar completamente todo el fenómeno acústico implícito. Para complicar aún más las cosas, debe decirse que no se

conocen exactamente los límites precisos de la percepción auditiva. Una cosa sí que es cierta: en el mejor de los casos, con los equipos de audio más sofisticados, sólo tendremos una aproximación del sonido original.

FUNDAMENTOS DIGITALES

La acústica y la tecnología analógica del audio están relacionadas principalmente a través de funciones matemáticas; sin embargo, el audio digital está formado por valores discretos. Específicamente, la amplitud de una forma de onda se representa con una serie de números. Este primer principio es muy importante, ya que los números van a permitir manejar la información de audio de una manera mucho más eficiente. Con técnicas digitales es posible aumentar de forma significativa todo el proceso que se realiza a una señal de audio. La tecnología digital ha permitido desarrollar nuevos equipos de grabación, de procesamiento y de reproducción de audio; el desarrollo de software aplicado al mundo del audio puede considerarse revolucionario. Así, antes de cualquier otra cosa, el audio digital es una tecnología numérica. Para comprender este hecho adecuadamente, vamos a empezar dando un repaso a los sistemas numéricos.

Cuando abordamos cualquier sistema de audio, el problema básico es la representación de la información en un formato digital. Aunque puede haber muchas alternativas en esta representación, la elección lógica es utilizar un sistema de números binarios. Esta representación en base 2 es ideal para almacenar y procesar la información numérica. Las operaciones aritméticas básicas se implementan mucho más fácilmente considerándolas como operaciones lógicas.

Sistemas numéricos

Todo empieza a partir de números. En audio digital, tratamos con información y con números, en contraposición a una representación analógica del audio. Los números ofrecen la posibilidad de codificar, procesar y decodificar la información de un modo impresionante. Normalmente los números se tratan como símbolos. La simbología tiene muchas ventajas, ya que los símbolos numéricos son altamente versátiles; el significado puede variar de acuerdo a su uso.

Por ejemplo, considere mi motocicleta, una BMW R50/2 1962, de 500 centímetros cúbicos, con matrícula 129907 (Figura 1.1). Existen varios números que la describen; no es obvio la importancia y el contexto de cada uno de ellos. R50/2 representa el modelo de la moto, 1962 es el año de fabricación y 500 representa los centímetros cúbicos del motor. El número de licencia 129907 (matrícula) representa específicamente una información codificada que hace que las multas por exceso de velocidad que cometa se carguen directamente en mi cuenta corriente. Estos números son sólo útiles en virtud del contexto al que arbitrariamente han sido asignados. Si el contexto es

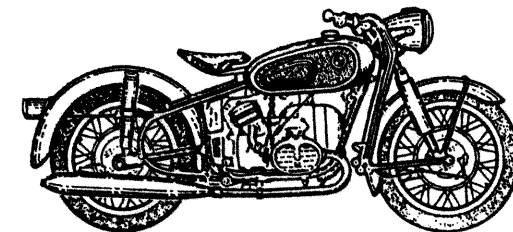


Figura 1.1. La clásica motocicleta del autor: BMW R50/2 de 1962.

confuso, la información codificada se tergiversa. Puede terminar con una moto que tuviera una matrícula 1962, fabricada en el año 500 y con un motor de 129907 centímetros cúbicos.

De forma análoga, las operaciones numéricas llevadas a cabo con los números pueden ser objeto de interpretación. La cuenta de mis multas puede llevar aparejada la suspensión de mi carné de conducir, sin embargo, la suma de mis recibos de circulación no es tan problemática. Los números, cuando están perfectamente definidos, proporcionan un método excelente para guardar y procesar los datos. Por otra parte, éstos y lo que representan deben emplearse con sumo cuidado para no cometer errores desproporcionados.

Para la mayoría de la gente, los números más habituales, y con los que está familiarizada, son los números en base 10, aparentemente creados en el siglo noveno por astrónomos hindúes: asignaron el valor nulo (la nada) al número 0 y añadieron otros nueve dígitos, dígitos que ya eran usados desde tiempos pretéritos. Las primeras sociedades utilizaron sistemas numéricos unitarios, que empleaban un único símbolo en una serie de marcas, para responder a una cuestión fundamental: ¿cuántos? Este tipo de sistema es inviable cuando se trabaja con números grandes; así, se empezaron a crear sistemas de números con bases mayores. Los matemáticos babilonios inventaron un sistema numérico de 60 símbolos. Era un poco incómodo, pero incluso hoy, 3.700 años después, la esencia de ese sistema perdura cuando se divide una hora en 60 minutos, un minuto en 60 segundos y un círculo en 360 grados.

Elegir un buen sistema numérico es una cuestión de importancia, ya que cualquier número entero puede expresarse con una base cualquiera. Cuando elegimos un determinado sistema numérico estamos diciendo el número de símbolos diferentes. La base 10 utiliza 10 números; la raíz del sistema es 10. Además, el sistema emplea una notación ordenada; la posición de los números determina la cantidad de unidades, decenas, centenas, etc. En otras palabras, el número de cada posición sucesiva se multiplica por la base, elevada a la potencia siguiente a esa posición. El sistema en base 10 se ajusta a nuestra forma de contar al disponer de manos con 10 dedos, pero para otras aplicaciones puede ser más conveniente utilizar otros sistemas con una base distinta. En cualquier sistema, debemos conocer su raíz; por ejemplo, el número 10 en base 10 representa la totalidad de los dedos de las manos, pero el número 10 en base 8 puede representar todos los dedos de las manos menos los dos pulgares. De forma similar, nos podemos hacer la siguiente pregunta: ¿Qué nos interesa más, tener 10.000 euros en base 6 o 100 en base 60? En la Tabla 1.1 se muestran cuatro sistemas numéricos con bases distintas.

Tabla 1.1. Cuatro sistemas numéricos corrientes

| Hexadecimal (Base 16) | Decimal (Base 10) | Octal (Base 8) | Binario (Base 2) |
|-----------------------|-------------------|----------------|------------------|
| 0 | 0 | 0 | 0000 |
| 1 | 1 | 1 | 0001 |
| 2 | 2 | 2 | 0010 |
| 3 | 3 | 3 | 0011 |
| 4 | 4 | 4 | 0100 |
| 5 | 5 | 5 | 0101 |
| 6 | 6 | 6 | 0110 |
| 7 | 7 | 7 | 0111 |
| 8 | 8 | 10 | 1000 |
| 9 | 9 | 11 | 1001 |
| A | 10 | 12 | 1010 |
| B | 11 | 13 | 1011 |
| C | 12 | 14 | 1100 |
| D | 13 | 15 | 1101 |
| E | 14 | 16 | 1110 |
| F | 15 | 17 | 1111 |

El sistema binario

Gottfried Wilhelm von Leibnitz, filósofo y matemático, creó el sistema binario el 15 de marzo de 1679. El origen de los sistemas digitales de hoy en día se remonta a aquella fecha. Aunque la base 10 es muy práctica para los humanos, la base 2, o el sistema binario, es más eficiente para los ordenadores y los equipos digitales. Sólo se necesitan dos números para representar dos estados eléctricos de la máquina: «on/off». Además, estas dos condiciones pueden representarse fácilmente con dos números: 0 y 1; a estos dígitos binarios se les denomina bits. Desde un punto de vista exclusivo de la máquina, el sistema binario es muy eficiente y rápido. Imagine con que rapidez se puede conmutar un interruptor; esa velocidad representa la tasa a la que se puede procesar la información. Imagine ahora una onda cuadrada; esta onda puede representar perfectamente la conmutación que realiza la máquina. Considere también las ventajas en el almacenamiento (registro). En vez de almacenar los infinitos valores posibles de una señal analógica, debemos recordar ahora únicamente dos valores. Los circuitos digitales exprimen al máximo las ventajas de la representación binaria para poder procesar de forma óptima toda la información contenida en una señal de audio.

Un procesador digital codifica toda la información en bits —una señal de audio convertida en un formato binario—, empleando un sistema en base 2, independientemente del poco parecido que pueda haber entre dicha información y los dígitos binarios. Vamos a ver brevemente la aritmética binaria para comprender mejor cómo se procesa el audio en un sistema digital. De hecho, veremos constantemente que la codificación binaria de los datos es una de las cuestiones más importantes en el diseño y operación de los sistemas digitales de audio.

Esencialmente, todos los sistemas numéricos realizan la misma función; por tanto, podemos familiarizarnos con el sistema binario comparándolo con el sistema decimal. Un mismo valor puede representarse en sistemas numéricos de base distinta. Existen varios métodos para convertir los dígitos de una base a otra. Un número decimal se puede convertir en un número binario dividiéndolo sucesivamente por 2 y colocando el resto de cada división parcial en el orden en el que aparece. De forma similar, la conversión de un sistema binario a uno decimal se realiza multiplicando cada dígito binario por la potencia de 2, de acuerdo con la posición que ocupan; sumando todos los productos se obtiene el resultado final.

Esta conversión pone de relieve que el sistema numérico de base 2 también emplea una notación ordenada. En base 2, cada posición sucesiva representa un valor doble. La columna más a la derecha representa 1, la siguiente 2, después 4, 8, 16, etc. Es importante designar la base que está siendo usada; por ejemplo, en base 2 el símbolo 10 puede representar las dos manos de una persona.

De la misma forma que el punto decimal separa la parte entera de la parte fraccionaria, el punto binario realiza la misma función en un sistema de base 2. La parte fraccionaria de un número decimal se puede convertir en un número binario simplemente multiplicando el número decimal por 2. La conversión puede acarrear infinitos dígitos binarios; por lo tanto, el número de dígitos debe limitarse a un determinado valor.

Al igual que en el sistema decimal, las operaciones aritméticas de suma, resta, producto y división, también son aplicables en el sistema binario. Como en cualquier base, la suma en base 2 se realiza aplicando unas reglas determinadas (tabla de sumar). El procedimiento es el mismo que en el sistema decimal; sin embargo, la tabla resultante es mucho más simple. Sólo existen cuatro operaciones posibles, comparadas con las más de 100 que resultan en una suma decimal. El acarreo, igual que en el sistema decimal, es necesario cuando el resultado es más grande que el dígito mayor del sistema. Los algoritmos de resta, multiplicación y división en el sistema binario son idénticos a los del sistema decimal.

Así, un número es lo que hagamos de él; los sistemas numéricos —diferenciados por la base— operan esencialmente de la misma forma. La utilización del sistema binario por parte

de los ordenadores es una cuestión de conveniencia; no presenta barrera alguna para entender las técnicas digitales. Simplemente es la aproximación más lógica. Pregúntese usted mismo: ¿cuántos dígitos manejaría mejor: 10, 60, infinitos o simplemente 2 niveles de tensión eléctrica?

CÓDIGOS BINARIOS

Aunque los sistemas digitales de audio estén basados en una abstracción matemática binaria, la implementación de estos conceptos requiere un nivel mayor de procesamiento. Específicamente, el siguiente paso es el estudio de la codificación de la información binaria. Por ejemplo, los dígitos binarios, o los bits, se pueden agrupar en palabras para formar una estructura mayor, con un cierto significado. De esta forma, los sistemas digitales pueden procesar más fácilmente la información simbólica y la información numérica.

Igual que los dígitos de la matrícula de una motocicleta tienen un determinado significado, los grupos de dígitos binarios pueden codificarse con una cierta información. Por ejemplo, un número decimal se puede convertir en su equivalente binario; este número binario está codificado y representa el número decimal original. Obviamente, hay una limitación en el número posible de valores a codificar. Específicamente, un número binario de n bits puede codificar a 2^n números decimales. Tres bits, por ejemplo, pueden codificar ocho estados: 000, 001, 010, 011, 100, 101, 110, 111. Éstos se pueden corresponder perfectamente con los números decimales 0, 1, 2, 3, 4, 5, 6 y 7.

Los números negativos representan un problema, ya que en éstos también debe codificarse el signo. Por ejemplo, un 1 a la izquierda puede designar un número negativo, un 0 puede designar un número positivo y el resto de bits corresponde al valor absoluto. Esta clase de codificación se denomina codificación de amplitud y signo. Las palabras de tres bits 000, 001, 010, 011, 100, 101, 110, 111 pueden corresponder ahora a +0, +1, +2, +3, -0, -1, -2, -3. Existe una irregularidad con la presencia de +0 y -0. Ya se verá más adelante cómo se utilizan otros métodos para representar de una forma óptima los números negativos.

Puesto que vivimos en un mundo decimal, a menudo es útil crear palabras binarias codificando sus correspondientes valores decimales, preservando de esta forma las características del formato decimal. Desgraciadamente, no existe ningún grupo binario que directamente represente los 10 dígitos decimales. Para ganar en eficiencia, se hace necesario un método de codificación más sofisticado. Este método se consigue fácilmente con grupos de cuatro bits, cada uno de ellos representa un dígito decimal:

| | | |
|----------------------|-----------------------|-----------------------|
| Primer grupo decimal | Segundo grupo decimal | Enésimo grupo decimal |
| $a_3a_2a_1a_0$ | $b_3b_2b_1b_0$ | $n_3n_2n_1n_0$ |

Con este tipo de enfoque, existen muchas formas de codificar los 10 dígitos binarios con grupos de cuatro bits. Es conveniente utilizar la que mejores beneficios reporte. Por ejemplo, un buen código debe facilitar las operaciones aritméticas, corregir errores y minimizar el espacio de almacenamiento y el diseño de los circuitos lógicos. Análogamente, los sistemas digitales de audio siempre se diseñan con sistemas de codificación que tengan en cuenta estas premisas.

Códigos binarios ponderados

En algunas aplicaciones, los códigos ponderados son mucho más eficientes. En un código ponderado, a cada bit se le asigna un valor decimal, un peso. Cada número, representado por el

código binario, se calcula a partir de la suma de todos los dígitos. Por ejemplo, los pesos w_3, w_2, w_1, w_0 y los bits a_3, a_2, a_1, a_0 representan a un número decimal $N = w_3 \times a_3 + w_2 \times a_2 + w_1 \times a_1 + w_0 \times a_0$.

El código BCD (código decimal binario) se emplea normalmente en aplicaciones digitales. El código BCD es un código ponderado según la posición de los dígitos; cada número está representado por una palabra de 4 bits. También se le conoce por el nombre de código 8-4-2-1, según el valor de sus pesos. La representación de este código se muestra en la Tabla 1.2, junto a otros códigos. Únicamente los códigos 8-4-2-1 y el 6-3-1-1 son códigos ponderados, el resto de los mostrados en la tabla no lo son. Cualquier número decimal puede representarse con el código BCD. Por ejemplo, el número 5995 sería 0101 1001 1001 0101. Hay que resaltar que el número binario resultante es completamente diferente del obtenido tras aplicar una conversión del sistema decimal al sistema binario. El código BCD permite representar grandes números; los sistemas que necesitan mostrar dígitos decimales suelen utilizar este tipo de código. Tales sistemas tienen que realizar conversiones binaria-BCD y BCD-binaria. El código 6-3-1-1 es otro tipo de código ponderado. Por ejemplo, empleando los pesos asignados a este código (6-3-1-1), los dígitos 1001 representarían el número decimal $N = 6 \times 1 + 3 \times 0 + 1 \times 0 + 1 \times 1 = 7$.

Códigos binarios sin ponderación

En otras aplicaciones son más útiles este tipo de códigos. Algunos de éstos son el código en exceso 3, el código 2-5 y el código Gray. El primero de ellos se deriva del código 8-4-2-1, simplemente sumando 3 (0011) a cada una de las palabras del código. En otras palabras, el valor decimal resultante es el valor del código BCD + 3. De esta forma cada una de las palabras tiene al menos un «1» en su representación. En el código 2-5 las palabras del código tienen 5 bits y dos de ellos siempre son 1. Esta representación permite chequear errores de una manera muy simple; un código sería erróneo cuando violara esta condición.

Se pueden definir otros códigos sin ponderación, por ejemplo, códigos en los que las palabras tengan un número de «1» entre uno y dos, ni más ni menos. Esto minimiza las transiciones lógicas al cambiar de un número a otro (números consecutivos), potencial causa de errores de distorsión en las señales de salida de los circuitos. En el código Gray, llamado a veces código reflejado, sólo puede cambiar uno de los dígitos. Un inconveniente que presenta este tipo de códigos es que el valor decimal que representan no se calcula tan fácilmente como en el de los códigos ponderados.

Tabla 1.2. Ejemplos de códigos binarios. Los códigos 8-4-2-1 y 6-3-1-1 son códigos ponderados; el resto no son ponderados

| Dígito decimal | Código 8-4-2-1 | Código 6-3-1-1 | Código exceso 3 | Código 2 de 5 | Código Gray |
|----------------|----------------|----------------|-----------------|---------------|-------------|
| 0 | 0000 | 000 | 0011 | 00011 | 0000 |
| 1 | 0001 | 0001 | 0100 | 00101 | 0001 |
| 2 | 0010 | 0011 | 0101 | 00110 | 0011 |
| 3 | 0011 | 0100 | 0110 | 01001 | 0010 |
| 4 | 0100 | 0101 | 0111 | 01010 | 0110 |
| 5 | 0101 | 0111 | 1000 | 01100 | 1110 |
| 6 | 0110 | 1000 | 1001 | 10001 | 1010 |
| 7 | 0111 | 1001 | 1010 | 10010 | 1011 |
| 8 | 1000 | 1011 | 1011 | 10100 | 1001 |
| 9 | 1001 | 1100 | 1100 | 11000 | 1000 |

COMPLEMENTO A DOS

Aunque pueda ser más cómodo definir las operaciones binarias de una manera familiar a nuestro modo de operar, esto no sería lo más acertado. Tiene más sentido especificar los números y las operaciones sobre ellos de aquella manera que le sea más fácil de operar a la propia máquina. Por ejemplo, cuando los números binarios se almacenan de una forma complementada, la operación suma puede utilizarse tanto para realizar sumas como restas. Además, cuando estos números binarios se almacenan y se procesan con registros de memoria, es aconsejable emplear un sistema numérico modular.

Asimismo, cuando se almacena el resultado de una operación aritmética pueden surgir ciertos problemas. Por ejemplo, suponga que se suman los números 100 y 110 en un sistema binario de tres bits. El resultado de dicha suma sería 1010. El bit de la izquierda es el acarreo producido en la suma. Sin embargo, en un sistema binario de tres bits se desecharía el bit de acarreo y el resultado final sería 010. Por tanto, el resultado sería incorrecto a menos que el sistema sea capaz de identificar los resultados que estén fuera de rango (*overflow*). Con una aritmética modular es posible identificar este tipo de problemas.

Este desbordamiento del resultado es inherente a cualquier sistema que trabaje con un número finito de dígitos. Requiere cierta reflexión para lograr entenderlo. En sistemas que trabajan con un número infinito de dígitos, la operación suma, por ejemplo, de 6 más 4 puede representarse como la unión de dos segmentos de longitudes 6 y 4 que producen como resultado un segmento de longitud 10. Sin embargo, con un número finito de dígitos es mucho mejor realizar las operaciones de forma circular, sobre una circunferencia de perímetro la longitud total de los dígitos. Por ejemplo, cuando sumamos 6_{10} (110_2) y 4_{10} (100_2), el resultado sería 2_{10} (010_2). Este resultado es el exceso que tiene el resultado (10_{10}) sobre 8_{10} , número de dígitos en un sistema binario de tres bits ($8 = 2^3$). Dos números A y B son equivalentes en módulo N si el resto de la división de A/N es idéntico al resto de la división de B/N . Por ejemplo, si $A = 12$ y $B = 20$, entonces $A = B$ en módulo 8, ya que el resto de $12/8$ es igual al resto de $20/8$. La aritmética de módulo 2 se utiliza en muchas aplicaciones binarias; en general, cuando se realizan operaciones con palabras de n bits la aritmética que se emplea es de módulo 2^n .

Como se ha señalado anteriormente, los números negativos en formato binario se expresan con un bit de signo. Aunque es adecuado para nuestra comprensión, esta representación es ilógica para las máquinas. Se necesitarían circuitos independientes para realizar sumas y restas. En lugar de eso, utilizando una propiedad de la representación en módulo, la operación suma sirve también para realizar las restas. El acarreo en el proceso de una suma sigue siendo necesario, pero no así el acarreo de una resta. Específicamente, un número negativo se codifica con una representación en complemento, en vez de utilizar un bit de signo seguido de otros bits que denotan la amplitud. Existen dos clases de complemento. En el complemento «raíz-menos-uno» cada dígito se forma restando a la raíz de la base menos uno (9 en el sistema decimal, 1 en el sistema binario) el dígito inicial. A este complemento se le conoce como el «complemento a nueve» para un sistema decimal y como el «complemento a uno» para un sistema binario. Un verdadero complemento se forma a partir del «complemento raíz-menos-uno» y sumándole uno al dígito menos significativo. Este complemento se llama «complemento a diez» para un sistema decimal y «complemento a dos» para un sistema binario. Con este tipo de complemento (base complementada), cuando se suman dos números complementados entre sí el resultado que se obtiene es la base elevada al número de dígitos de los números. Por ejemplo, el «complemento a diez» de 35 es 65 y su suma es $10^2 = 100$, y el «complemento a dos» de 1011 es 0101 y su suma es $1000 = 2^4$. Como puede apreciarse en este ejemplo, el complemento se lleva a cabo complementando cada símbolo y sumándole 1, y el complemento de cada símbolo restándole al símbolo mayor de cada sistema (9 en decimal, 1 en binario) el dígito correspondiente.

En un sistema binario el «complemento a uno» y el «complemento a dos» se obtienen fácilmente, acelerándose el proceso de una forma significativa. Puesto que 2 es la raíz del sistema, se

debe restar 1 a cada uno de los bits. Así, el «complemento a uno» se forma sustituyendo los ceros por unos y viceversa, es decir, complementando los bits. El «complemento a dos» se forma sumando un 1 al número resultante (complemento a uno), incluyendo el posible acarreo que se produzca. Por ejemplo, dado un número 0100, su «complemento a uno» es 1011 y su «complemento a dos» es 1100. El «complemento a dos» es el más utilizado. Por ejemplo, en la Tabla 1.3 se muestra la representación en «complemento a dos» de los números comprendidos entre +7 y -7. La suma de un número positivo con su correspondiente negativo siempre da cero.

Volviendo a la discusión inicial de los sistemas que trabajan en módulo, podemos observar ahora que una resta puede ser sustituida por una suma. Si $A + B = 0$ en un sistema de módulo N , entonces B es valor negativo de A . Existen en realidad muchísimos valores que cumplen esta condición, todos aquellos que $B = kN - A$ donde $k = 0, 1, 2, 3, \dots$ Para k igual a 1 y A menor que N , $B = N - A$ y es, por tanto, un número positivo, menor que N . Podemos emplear entonces la expresión $B = N - A$ en lugar de $-A$ siempre que se realice una operación en módulo N . Por ejemplo, $C = D - A$ es equivalente a $C = D + (N - A)$. En otras palabras, si podemos obtener $N - A$ sin realizar una resta, entonces la resta se puede realizar haciendo una suma. Después de reflexionar brevemente en la materia, se observa que $N - A$ es realmente el «complemento a dos» de A .

La resta se puede realizar mediante una suma. Veamos primero la resta normal en el sistema binario:

$$\begin{array}{r} 10001 \\ - 01011 \\ \hline 00110 \end{array}$$

Sin embargo, esta misma operación puede realizarse sumando el «complemento a dos» del sustraendo:

$$\begin{array}{r} 10001 \\ + 10101 \\ \hline 100110 \end{array} \text{ (el bit de acarreo se descarta)}$$

Tabla 1.3. Tabla de valores en la que se muestra la representación en complemento a dos de los números positivos y negativos

| Número decimal | Número binario |
|----------------|----------------|
| 7 | 0111 |
| 6 | 0110 |
| 5 | 0101 |
| 4 | 0100 |
| 3 | 0011 |
| 2 | 0010 |
| 1 | 0001 |
| 0 | 0000 |
| -1 | 1111 |
| -2 | 1110 |
| -3 | 1101 |
| -4 | 1100 |
| -5 | 1011 |
| -6 | 1010 |
| -7 | 1001 |

Cuando el minuendo es menor que el sustraendo no se produce acarreo. Por ejemplo, $2 - 8$ resulta $2 + (10 - 8)$ o $2 + 2 = 4$. Observe que 4 es el «complemento a diez» del resultado negativo: $-(10 - 4) = -6$. En un sistema binario ocurre lo mismo:

$$\begin{array}{r} 101 \\ - 11011 \\ - 10110 \end{array}$$

En «complemento a dos»:

$$\begin{array}{r} 101 \\ + 00101 \\ 01010 \end{array}$$

El resultado es negativo, pero en «complemento a dos». Tomando el «complemento a dos» y asignando el operador de la resta (-) el resultado es negativo (-10110). Cuando se realiza una resta en «complemento a dos», el acarreo final nos dice el signo del resultado. Un 1 en el acarreo final indica que es positivo y un 0 que es negativo, en «complemento a dos», representación positiva.

La representación en complemento puede parecer algo tediosa, pero está compensada con las múltiples ventajas que aporta a la hora de presentar los valores positivos y negativos (bipolares), por ejemplo, de una señal de audio. El bit más significativo (MSB) es el bit de signo. Cuando éste es cero indica un número positivo y cuando es 1 indica un número negativo. En una representación binaria pura, el número 5 se representa por 00000101 y -5 por 10000101. Cuando este mismo número se representa en «complemento a dos», -5 se transforma en 11111011, y el signo se maneja de una forma automática. El resultado de una suma o de una resta, cuando éste sea positivo, tendrá una representación binaria pura, y si es negativo, tendrá una representación en «complemento a dos», con una asignación automática del bit de signo (MSB). Se puede pensar que lo más importante en el «complemento a dos» es el bit de signo. Sin embargo, lo verdaderamente importante es poder realizar restas mediante simples sumas.

ÁLGEBRA BOOLEANA

El sistema binario ofrece tremendas oportunidades en el diseño electrónico de hardware y de software, incluyendo, por supuesto, las aplicaciones en audio digital. El álgebra de Boole es una herramienta que permite manejar y manipular las señales binarias. El nombre se debe a su inventor, George Boole, que publicó en 1854 un curioso trabajo titulado *An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities*. Por cierto, los historiadores dicen de él que no superó el tercer grado en su formación académica.

En aplicaciones digitales, el álgebra de Boole proporciona la base matemática para realizar toma de decisiones, chequeo de condiciones y operaciones lógicas. Utilizando el álgebra de Boole, todas las decisiones lógicas se realizan a partir de dígitos binarios (0 y 1), un conjunto de operadores y una serie de leyes y teoremas. Los estados on/off se adaptan perfectamente al modo de funcionamiento de los sistemas digitales. El conjunto de operadores lógicos constituye la herramienta necesaria para poder manipular los bits y poder, por consiguiente, diseñar la lógica que se utiliza en dichos sistemas. Con este tipo de matemática se pueden diseñar desde las máquinas expendedoras de tabaco hasta los más modernos ordenadores.

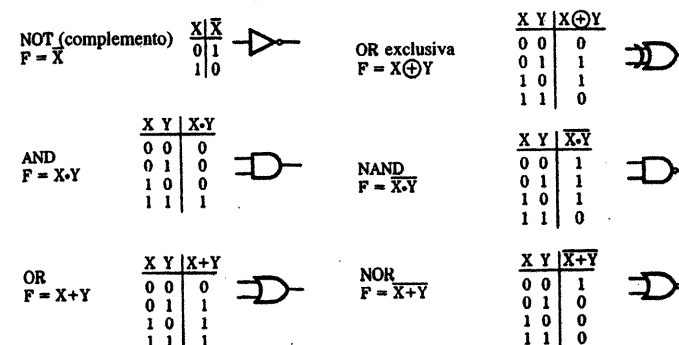


Figura 1.2. Los seis operadores booleanos utilizados para manipular condiciones lógicas.

Los operadores matemáticos se muestran en la Figura 1.2. Los operadores OR, AND y OR EXCLUSIVA producen un dígito binario a partir de dos iniciales. El operador booleano NOT invierte el dígito. Los operadores NAND y NOR se derivan de los operadores AND y OR. Los operadores se pueden utilizar de forma aislada o en combinación con otros para realizar cualquier función lógica.

La operación NOT invierte los dígitos del dato. Un 0 pasa a ser un 1 y un 1 pasa a ser un 0. Esta operación se representa con una barra sobre el dato.

El resultado de una operación AND es 1 siempre y cuando los dos datos de entrada (X, Y) sean 1; en cualquier otro caso es un 0. La operación se representa con un punto ($X \cdot Y$), o directamente como XY.

El resultado de una operación OR es 1 siempre y cuando alguno de los dos datos de entrada (X, Y) sean 1; en cualquier otro caso es un 0. La operación se representa por el signo +.

La operación OR-EXCLUSIVA (XOR) nos sirve para saber si los dos dígitos tienen igual valor. El resultado es 1 cuando X e Y son distintos, y es 0 cuando ambos son iguales. La operación XOR realiza, por tanto, una suma binaria en módulo 2. Se representa mediante el símbolo + encerrado en un círculo.

Al combinar una operación AND con una NOT se produce la operación NAND, y al combinar una operación OR con una NOT se produce la operación NOR; el resultado que se obtiene es la inversión del resultado de la operación AND o de la operación OR, respectivamente.

Los operadores booleanos se pueden combinar entre sí para buscar la expresión simplificada que mejor se acerque a la condición que estamos buscando. Por ejemplo, un sistema digital sólo necesita los operadores OR y NOT, ya que cualquier otro operador se puede derivar de estos dos. Esta relación queda definida por el teorema de Morgan, que dice:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

De acuerdo con el teorema de Morgan, podemos observar que la expresión

$$A \cdot B = \overline{\overline{A} + \overline{B}}$$

genera una AND a partir de una OR y una NOT, y la expresión

$$A \oplus B = \overline{(\overline{A} + B)} + (\overline{A + B})$$

genera una XOR a partir de una OR y una NOT.

En este ejemplo se muestra cómo pueden combinarse los operadores booleanos. En este caso, el teorema de Morgan se utiliza para formar el complemento de las expresiones. Esta habilidad para formar expresiones lógicas nos permite utilizar los operadores booleanos, junto con una o más variables o constantes, para resolver algunos problemas. Los paréntesis indican el orden en que deben realizarse las operaciones; en primer lugar aquellas que están dentro de los paréntesis. Cuando no hay paréntesis, la primera operación es la negación (NOT), después la operación AND y por último la operación OR.

Las expresiones lógicas corresponden a circuitos de puertas lógicas, implementados en un hardware o en un software. Por ejemplo, en la Figura 1.3A se muestra una expresión lógica y su correspondiente circuito de puertas lógicas. Una expresión lógica se puede evaluar sustituyendo cada una de las variables por un 1 o por un 0, y realizando las operaciones indicadas en la misma. Cada vez que aparezca la variable en la expresión (o su complemento) la llamaremos literal. Para ilustrar todas las posibles combinaciones que pueden darse en una expresión se define lo que se llama una tabla de verdad. En otras palabras, todas las posibles salidas de una expresión se calculan a partir de todas las posibles entradas. Por ejemplo, en la tabla de verdad de la Figura 1.3B se muestran los resultados obtenidos por el circuito lógico de la Figura 1.3A.

Dado un conjunto de operadores y todas sus posibles combinaciones, el siguiente paso es desarrollar un sistema de relaciones booleanas algebraicas. Este conjunto forma la base del procesamiento digital, de la misma forma que el álgebra gobierna las operaciones que se realizan en el sistema decimal. De hecho, el sistema binario y el sistema decimal son sistemas muy similares. Las propiedades conmutativa, asociativa, distributiva y el complemento forman toda la base matemática para desarrollar sistemas lógicos. En estas leyes se asienta todo el álgebra de Boole.

El doble complemento de una variable da como resultado la misma variable. En otras palabras, si se complementa el valor 1 resulta 0, y si se complementa de nuevo 0 tendremos el valor inicial 1. La propiedad conmutativa nos dice que el resultado no depende del orden de los operadores (suma y multiplicación). La propiedad asociativa establece que el resultado de realizar sumas o multiplicaciones no depende del orden en que se realicen las mismas. La propiedad distributiva establece que el producto de una suma es igual a la suma de los productos.

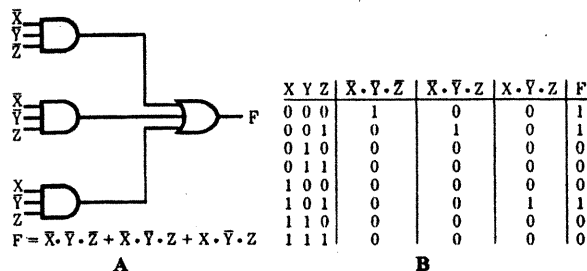


Figura 1.3. Las expresiones booleanas se pueden implementar mediante hardware o software, con circuitos o con instrucciones lógicas. A. La implementación lógica de una expresión booleana. B. La tabla de verdad para verificar la solución de la expresión lógica.

Mediante los teoremas booleanos y de otro tipo, como el teorema de Morgan, se puede simplificar drásticamente las expresiones lógicas obteniendo, de esta forma, resultados más simples. Por ejemplo, considere las siguientes expresiones:

$$\begin{aligned}
 F &= X \cdot Y + X \cdot \bar{Y} + \bar{Y} (X + \bar{X}) \\
 F &= X \cdot Y + X \cdot \bar{Y} + \bar{X} \cdot \bar{Y} \\
 F &= \bar{X} \cdot Y + X \cdot \bar{Y} + \bar{Y} \\
 F &= \bar{Y} + Y \cdot X \\
 F &= X + \bar{X} \cdot \bar{Y} \\
 F &= X + \bar{Y}
 \end{aligned}$$

En todas estas funciones se muestran diferentes operaciones, pero el resultado siempre es el mismo. Cuando el resultado final está ligado a un diseño hardware de puertas lógicas o al de un software, se hace necesario una simplificación con objeto de optimizar los recursos. Por ejemplo, el álgebra de Boole se puede aplicar a la expresión de la Figura 1.3 para conseguir una expresión mucho más simple, como se muestra en la Figura 1.4. En este caso, el circuito original necesitaba tres puertas AND, una puerta OR y tres puertas inversoras NOT. Sin embargo, una vez simplificada la expresión, el circuito de la Figura 1.4 sólo requiere una puerta AND, una OR y dos puertas inversoras NOT. Ambos circuitos realizan, por tanto, la misma función, idénticas salidas para idénticas entradas.

De esta manera, el desarrollo hardware y software, aplicado al procesamiento digital de señales, se puede llevar a cabo de una forma óptima. En un ámbito más general, el álgebra de Boole se puede aplicar a múltiples problemas, y expresarlos con expresiones lógicas. Por ejemplo, considere el siguiente silogismo:

1. Los niños son ilógicos.
2. Nadie que adiestre cocodrilos es despreciable.
3. Las personas ilógicas son despreciables.

Aplicando los principios del simbolismo lógico, la conclusión lógica absoluta es:

4. Una persona que adiestre cocodrilos no es un niño.

La lógica binaria se puede aplicar de forma elegante, reduciendo el problema a estados binarios, independientemente de que éste trate niños, cocodrilos, motocicletas o violines. Desde esta simplicidad, se pueden abordar cuestiones tremendamente complejas. Los sistemas digitales son equivalentes. Difieren, sin embargo, en el tipo de información que procesan y en

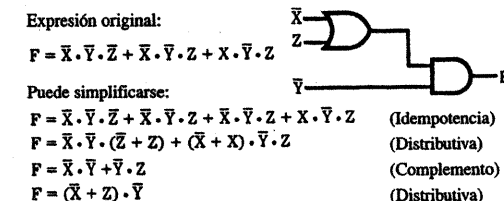


Figura 1.4. Las expresiones lógicas complejas pueden ser simplificadas por las leyes de Boole para permitir simplificar la circuitería (hardware) o su implementación software.

las manipulaciones de datos que realizan. Un sistema de audio digital es, por tanto, un sistema diseñado específicamente para procesar datos de audio. Realmente, un sistema digital de audio está más relacionado con el silogismo anterior que con los propios sistemas analógicos de audio.

ANALÓGICO VERSUS DIGITAL

El audio digital comprende nuevos conceptos y técnicas, distintas de las utilizadas en la tecnología analógica. Desde el principio, debemos considerar la información, su registro y su tratamiento con una nueva perspectiva. A este respecto, podemos realizar algunas comparaciones entre la tecnología analógica y la tecnología digital para vislumbrar las grandes diferencias que existen entre ambas, así como las ventajas y desventajas de cada una de ellas.

Comparemos una señal analógica con una señal digital; considere dos cubos iguales, uno lleno de agua y el otro de canicas. El volumen del cubo se puede determinar a partir de su contenido, pero el procedimiento para realizar este cálculo va a ser diferente según estemos midiendo el agua o contabilizando las canicas. Un procedimiento podría ser: primero pesar el cubo de agua, después hervir el agua para que se evapore, pesar de nuevo el cubo, restar este peso del peso inicial y, por último, calcular el volumen del cubo a partir del peso del agua evaporada. También se puede utilizar un recipiente calibrado y utilizarlo para ir sacando el agua del cubo. En cualquiera de los dos casos, se corre el riesgo de verter algo de agua o dejar algún residuo en el propio cubo. La medida sería muy imprecisa.

Con un cubo de canicas, simplemente hay que contarlas y después calcular el volumen a partir del número total de canicas. La medida es relativamente precisa, tal vez algo tediosa (se puede utilizar un ordenador para realizar el proceso). Las canicas representan, en este caso, los valores discretos en un sistema digital, y dan una muestra de la capacidad que pueden tener las técnicas digitales a la hora de cuantificar los valores y obtener la máxima precisión en un proceso de medida. En general, la precisión es fundamental en cualquier sistema digital. Por ejemplo, tiene menos utilidad expresar la capacidad del cubo como 1,5 litros que decir que en el cubo caben 8.263 canicas. Además, esta forma de realizar el proceso proporciona una repetitividad en la medida. Si se llena otro cubo de agua, existe la posibilidad de cometer un error, si este nuevo cubo no tiene una capacidad exactamente igual que la anterior. Se podría pensar que el cubo tiene una capacidad de 1,5 litros cuando en realidad sólo tiene 1,4 litros. Este error no puede darse con las canicas ya que el cubo debería admitir las mismas de antes.

La utilidad de un sistema digital frente a un sistema analógico es paradójica. Conceptualmente, un sistema digital es mucho más simple; contar números siempre es más fácil que medir un flujo continuo. Es decir, es más fácil trabajar con la representación de una señal que con la señal misma. Sin embargo, en la práctica, el equipo que se necesita para realizar este procesado es bastante más sofisticado que el correspondiente analógico.

Obviamente, los equipos que graban y reproducen el audio son algo más complejos que los cubos. Pero el ejemplo anterior nos sirve de forma resumida para saber cuáles son las diferencias entre la tecnología analógica y la digital. La cadena de la señal audio analógica, desde su grabación en el estudio hasta que la escuchamos en casa, debe preservar las continuas variaciones de amplitud que tiene la señal en función del tiempo. Esto es bastante problemático ya que los equipos de audio y de grabación son dispositivos analógicos, y sus imperfecciones de ruido, distorsión, etc., se incorporan a la propia señal analógica, degradándola. En resumen, cuando una señal analógica de audio se procesa con un equipo analógico, ésta sufre un deterioro, inversamente proporcional a la calidad del equipo.

En audio digital, la señal analógica se convierte en una secuencia de datos binarios que se procesan, se graban y se distribuyen con una representación numérica. El proceso contrario, reconvertir los datos binarios en una señal analógica, sólo ocurre en el proceso de reproducción,

eliminando por tanto toda posibilidad de tener algún tipo de degradación en cualquier otro punto de la cadena. La mayor preocupación radica en las conversiones analógica-digital y digital-analógica. No obstante, se elimina la mayoría de la degradación que sufría la señal analógica ya que los datos binarios son «prácticamente inmunes» al pasar ahora por una cadena digital (equipos digitales). Además, los equipos digitales son mucho más baratos de diseñar y de fabricar, son inmunes al ruido, no envejecen, no se degradan con la temperatura y son bastante robustos en su funcionamiento (fiabilidad). Incluso más importante que lo anterior, la potencia de los algoritmos digitales permite tener una flexibilidad de diseño y una precisión inimaginables desde un punto de vista analógico.

En teoría, los equipos digitales de audio son más «elegantes» en su funcionamiento y superan fácilmente las prestaciones de cualquier equipo analógico. En la práctica, aunque los sistemas digitales de audio están libres de las típicas deficiencias analógicas, tienen también sus propias anomalías. En los siguientes capítulos sólo vamos a ver por encima lo necesario para comprender toda la complejidad que encierran los sistemas digitales de audio. Además, como cualquier ciencia en continua evolución, por cada nuevo conocimiento surgen multitud de nuevas preguntas.

Capítulo 2

FUNDAMENTOS DEL AUDIO DIGITAL

Las técnicas digitales utilizadas para grabar, reproducir, procesar y transmitir las señales digitales de audio se fundamentan en conceptos ya utilizados en los sistemas analógicos de audio. De hecho, el funcionamiento de estos sistemas digitales está claramente ligado al de los sistemas analógicos. El audio es analógico en su origen y, por lo tanto, los sistemas digitales deben transformar este carácter analógico, mediante los procesos de muestreo y cuantificación —los dos pilares en los que se basa la digitalización—. Las muestras de una señal de audio deben cumplir el denominado teorema de muestreo. Las características de la señal de audio vienen determinadas por la frecuencia de muestreo. En concreto, el teorema establece una condición: la señal de audio debe ser de banda limitada. Además, el muestreo debe realizarse con cierta precaución, ya que se puede producir un error típico, conocido por el nombre de «aliasing». El error de cuantificación aparece cuando se representa la amplitud de las muestras en un formato binario; este error se puede minimizar aplicando técnicas de *dithering* (previas a la cuantificación).

MUESTREO DISCRETO

Una grabación analógica se puede realizar, por ejemplo, modificando la posición de las partículas magnéticas de una cinta, o variando la profundidad de los surcos en un disco de vinilo, de una forma continua y dinámica. Pero en una grabación digital la información está representada por números. Estos números se crean mediante los procesos de muestreo y cuantificación, codificando la señal analógica en una secuencia de valores de amplitud. Estos dos procesos se estudian en este capítulo. Vamos a considerar primero el muestreo periódico, el concepto más importante en un sistema de audio digital.

El tiempo parece fluir continuamente. Las manecillas de un reloj analógico recorren toda la superficie del mismo a medida que pasa el tiempo. Un reloj digital también dice la hora, pero lo hace presentando números en una pequeña pantalla. En otras palabras, visualiza muestras del tiempo. De una manera similar, la música varía continuamente en el tiempo y se puede grabar y reproducir, tanto de una forma analógica como discreta. El muestreo discreto en tiempo caracteriza a un sistema digital de audio de un sistema analógico, y es el proceso que permite realizar una conversión analógica-digital.

Sin embargo, surge inmediatamente una duda. Si un sistema digital sólo opera con muestras de la señal analógica, ¿qué ocurre con el resto de la señal? ¿Esta información se ha perdido

definitivamente? La respuesta, sorprendentemente, es negativa. Si se dan las condiciones adecuadas en el muestreo, en un sistema de digitalización no existe pérdida de información entre la entrada y la salida. Las muestras contienen la misma información que la señal analógica. Para ilustrar esto, veamos el siguiente experimento conceptual.

Suponga que ponemos una cámara de cine sujeta en el manillar de una motocicleta BMW para tener, de esta forma, la sensación subjetiva del movimiento de la motocicleta. Cuando vemos esta toma en la sala de cine, descubrimos que los fotogramas que componen la película reproducen fielmente el movimiento de la motocicleta; parece genial. Pero cuando la motocicleta atraviesa una zona de baches la imagen se torna borrosa. Es decir, los cambios bruscos de movimiento son demasiado rápidos para ser capturados en cada fotograma. Se puede sacar la siguiente conclusión: si aumentamos la frecuencia de los fotogramas, empleando más fotogramas por segundo, se podrían capturar movimientos más rápidos. Si nos quejamos al ayuntamiento y éste arregla el tramo que está salpicado de baches, la imagen dejará de estar borrosa, utilizando incluso una frecuencia de fotogramas inferior. La película reproduciría fielmente el movimiento de la motocicleta. Se puede llegar entonces a un cierto compromiso entre unas calzadas razonablemente firmes y una adecuada frecuencia de fotogramas, para tener así una imagen completamente nítida.

La analogía es un tanto infantil. (Para los principiantes, la proyección de una película se realiza de forma discontinua, nuestros ojos son los que crean la sensación de continuidad. La reproducción del audio en un equipo digital es realmente continua, existe una interpolación entre las muestras.) No obstante, la analogía nos muestra que a partir de fotogramas discretos es posible ver una película de forma continua y, de forma similar, a partir de las muestras de una señal de audio también es posible obtener una señal continua. Como se ha visto, el muestreo es un proceso sin pérdidas siempre y cuando se acondicione la señal de entrada. Así, en un sistema digital de audio, debemos «suavizar los baches» de la señal. Específicamente, la señal debe ser filtrada con un filtro paso-bajo; es decir, para realizar un muestreo óptimo, es necesario eliminar las frecuencias altas. Por tanto, una señal de banda limitada puede muestrearse sin ninguna pérdida de información; las muestras contienen toda la información de la señal analógica original. La señal puede recuperarse completamente a partir de sus muestras. Generalmente, existe un método para reconstruir la señal a partir de sus valores de amplitud, obtenidos de forma periódica en el tiempo.

TEOREMA DEL MUESTREO

El muestreo se realiza en múltiples disciplinas, y el origen de los teoremas de muestreo también es diverso. La mayoría de los ingenieros de audio norteamericanos otorgan a Harry Nyquist el honor de ser el autor del teorema del muestreo en el que se basa el audio digital moderno. Este reconocimiento se fundamenta en que Nyquist fue el primero en expresar este teorema en términos coherentes de telecomunicación. Sin embargo, la historia del muestreo es anterior a Nyquist.

El matemático francés Augustin-Louis Cauchy ya se ocupó del muestreo, «en los ratos libres que tenía cuando diseñaba fortificaciones militares para Napoleón». En 1841, Cauchy demostró que las funciones podían ser muestreadas de una manera no uniforme y promediadas en un determinado intervalo. A finales del siglo XIX se pensaba (incorrectamente) que una función podía ser muestreada a un ritmo idéntico al de su máxima frecuencia. En 1915, el matemático escocés E. T. Whittaker, trabajando con series de interpolación, formuló una expresión matemática (tal vez la primera) sobre el teorema general de muestreo, demostrando que la señal podía reconstruirse completamente a partir de sus muestras. En 1920, el matemático japonés K. Ogura demostró, de forma análoga, que si una función era muestreada al menos al doble de su frecuencia máxima, las muestras contendrían toda la información de la función, y podían ser utilizadas

para reconstruir nuevamente la función. También en 1920, el ingeniero norteamericano John Carson obtuvo, aunque no lo publicó, el mismo resultado en aplicaciones de telecomunicación.

Nyquist fue el primero en clarificar y publicar las aplicaciones del teorema de muestreo en el ámbito de las comunicaciones. En 1925, en un artículo titulado «Certain Factors Affecting Telegraph Speed», demostró que el número de impulsos que podían transmitirse sobre una línea telegráfica era proporcional al ancho de banda de la línea. En 1928, en un artículo titulado «Certain Topics in Telegraph Transmission Theory», demostró que para tener una reconstrucción completa de la señal, el ancho de banda necesario tenía que ser proporcional a la velocidad de la señal, y que el ancho de banda mínimo tenía que ser igual a la mitad del número de códigos por segundo. Posteriormente, el ingeniero ruso V. A. Kotelnikov publicó un artículo sobre el teorema del muestreo en 1933. En 1949 el matemático norteamericano Claude Shannon unificó todos los diversos aspectos del muestreo y sentó las bases de la teoría de la información. Hoy en día, los ingenieros generalmente atribuyen el teorema de muestreo a Shannon o a Nyquist. La mitad de la frecuencia de muestreo se conoce habitualmente por frecuencia Nyquist. Nyquist, que nació en Suecia en 1889 y murió en Texas (EEUU) en 1976, trabajó para los laboratorios Bell y fue autor de 138 patentes.

Quien quiera que tenga el mérito, el teorema de muestreo establece que una señal continua limitada en banda puede ser reemplazada por una secuencia discreta de muestras sin pérdida de información, y describe cómo se puede reconstruir la señal original a partir de sus muestras; además, el teorema especifica que la frecuencia de muestreo debe ser al menos el doble de la máxima frecuencia de la señal. Más específicamente, las señales de audio con frecuencias entre 0 y $S/2$ Hz pueden representarse exactamente con S muestras por segundo. De una forma más general, el principio de muestreo se aplica en múltiples disciplinas de ingeniería y de matemáticas como en óptica, holografía, meteorología y oceanografía, por nombrar algunas.

La frecuencia Nyquist

Cuando el teorema de muestreo se aplica a las señales de audio, éstas se filtran paso-bajo, de tal forma que su respuesta en frecuencia quede limitada en banda y no exceda de la frecuencia Nyquist ($S/2$). Idealmente, el filtro paso-bajo se diseña de manera que sólo se eliminen las frecuencias que no percibamos. Posteriormente, la señal se muestrea para obtener valores instantáneos de amplitud. La señal muestreada contiene la misma información que la señal analógica filtrada paso-bajo. A la salida del sistema, la señal puede ser reconstruida sin ningún tipo de pérdida de información (debido al muestreo) entre la señal de salida y la señal de entrada (filtrada paso-bajo). Desde el punto de vista del muestreo, la señal de salida no es una aproximación: es sencillamente idéntica a la señal de entrada. La señal limitada en banda se muestra en la Figura 2.1.

Considere una función analógica, continuamente cambiante, que ha sido muestreada para obtener una serie de impulsos de la misma. La amplitud de cada impulso, determinada por la cuantificación, es un número que representa la amplitud de la señal en ese instante. Para poder cuantificar esto, la frecuencia de muestreo se define como el número de muestras por segundo de la señal. El período de muestreo es su recíproco, es decir, el intervalo de tiempo existente entre muestras. Por ejemplo, a una frecuencia de muestreo de 40.000 muestras por segundo le corresponde un período de muestreo de $1/40.000$ segundos. Una señal con cambios rápidos —es decir, con altas frecuencias— requiere una frecuencia de muestreo más alta. Por tanto, la frecuencia de muestreo determina la frecuencia máxima que puede tener un sistema de digitalización. La selección de la frecuencia de muestreo es uno de los parámetros más importantes en la digitalización, ya que determina el ancho de banda del sistema.

El teorema de muestreo establece el ritmo al que se debe muestrear una señal para asegurar un determinado ancho de banda. Específicamente, como ya se ha señalado anteriormente, una

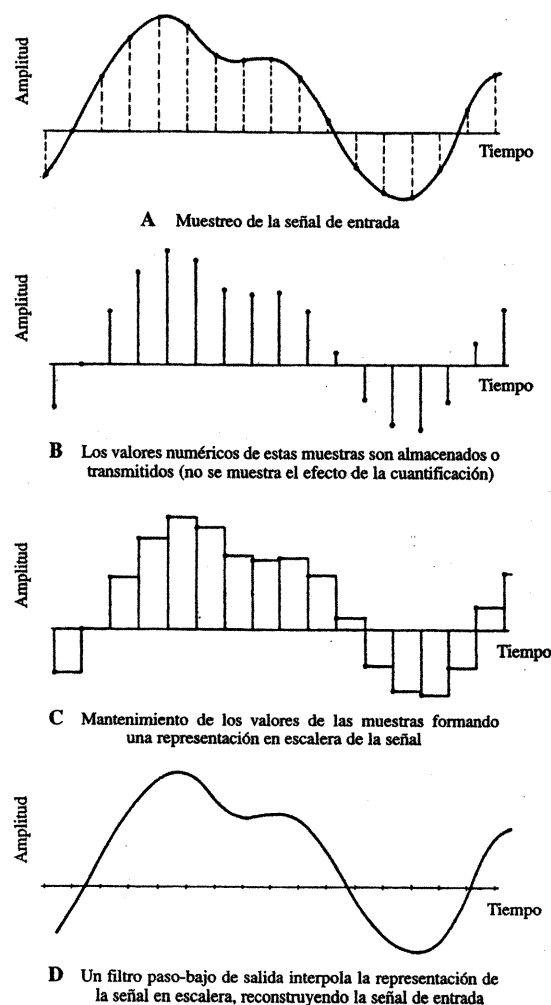


Figura 2.1. Realizando un muestreo discreto, una señal limitada en banda puede ser muestreada y reconstruida sin ningún tipo de pérdidas.

frecuencia de muestreo de S muestras/segundo determina un ancho de banda de $S/2$ Hz. En otras palabras, la frecuencia de muestreo debe ser al menos el doble de la máxima frecuencia de la señal de audio para tener un muestreo sin pérdida de información. Por ejemplo, una señal de audio con una respuesta en frecuencia de 0 a 20 kHz teóricamente necesitaría una frecuencia de muestreo de 40 kHz para que el muestreo fuera correcto. Es importante observar la condición, que impone el teorema de muestreo, de limitar el ancho de banda de señal a la mitad de la frecuencia de muestreo (frecuencia Nyquist). Las frecuencias de la señal superiores a este valor

producen una distorsión conocida por aliasing, como se describirá más adelante en este capítulo. Con objeto de eliminar dichas frecuencias se coloca previamente un filtro paso-bajo. También es necesario utilizar un filtro paso-bajo en la salida del sistema para reconstruir, a partir de sus muestras, la señal analógica o, lo que es lo mismo, para eliminar todas las nuevas frecuencias creadas por el sistema en el proceso de digitalización. Este filtro se estudiará en detalle en el Capítulo 4.

Veamos otro hecho relacionado con el teorema de muestreo. Cuando muestreamos una señal de audio de baja frecuencia podemos observar, debido a su gran longitud de onda, que en cada período existe un gran número de muestras. Pero a medida que la frecuencia de la señal aumenta, el período de la señal se hace más pequeño y el número de muestras en el mismo disminuye. En el límite, en el caso de un muestreo crítico, una señal cuya frecuencia sólo fuera la mitad de la frecuencia de muestreo tendría dos muestras por período. Sin embargo, incluso en este caso, el sistema podría recuperar la señal a partir de las muestras. Por ejemplo, considere el caso de una frecuencia de muestreo de 40 kHz y una senoide de 20 kHz. El muestreo produce dos muestras por período, transformando la señal sinusoidal en una onda cuadrada de 20 kHz. Sin embargo, el filtro paso-bajo que se coloca a la salida del sistema elimina todas las componentes de frecuencia superiores a la mitad de la frecuencia de muestreo. (La onda cuadrada de 20 kHz tiene armónicos impares: senoideas a 20, 60, 100, 140, etc.). Al eliminar todo este contenido espectral, la salida del sistema presenta de nuevo una senoide de 20 kHz, la misma que fue muestreada a la entrada. Además, podemos asegurar que la entrada era una senoide y no una onda cuadrada, ya que el filtro paso-bajo de entrada eliminó también todas aquellas frecuencias superiores a la frecuencia Nyquist (20 kHz). Análogamente, un sistema de digitalización puede reproducir toda la información contenida en el margen 0 y $S/2$ Hz, incluyendo la componente de $S/2$ Hz. De forma inversa, todo el contenido espectral superior a $S/2$ Hz es eliminado. Podemos decir entonces que frecuencias altas de muestreo permiten grabar y reproducir frecuencias altas de la señal de audio. Pero el aumentar la frecuencia de muestreo no repercutirá en la calidad de la señal en frecuencias bajas, sólo en la posibilidad de adquirir nuevas frecuencias altas.

En un muestreo crítico no existe garantía de que las muestras coincidan con los máximos y los mínimos de la señal. Podrían coincidir con pequeños valores de la onda e incluso con los valores nulos (cruce por cero). En la práctica, esto no supone ningún problema. Nunca habrá muestreo crítico; siempre habrá una zona de guarda entre el límite teórico y el límite real. Como hemos visto anteriormente, para cumplir el teorema de muestreo, en la entrada del sistema se coloca un filtro paso-bajo, previo al proceso de muestreo. Este filtro paso-bajo no puede realizar una atenuación instantánea a partir de la frecuencia Nyquist. En la práctica existe una banda de transición. La frecuencia de corte del filtro es algo inferior y produce así una atenuación gradual hasta la mitad de la frecuencia de muestreo. Esto asegura que en el proceso de muestreo no entren componentes de frecuencias superiores a la frecuencia Nyquist. A la señal nunca se le aplica un muestreo crítico; siempre hay más de dos muestras por período. Además, no existe ninguna relación de fase entre las muestras y la señal de entrada; las señales acústicas no se pueden sincronizar con el proceso de muestreo. En cualquier caso, la señal de salida no se reconstruye muestra a muestra; en realidad, se forma a partir de un conjunto de muestras, de igual forma que lo hace nuestro oído; el sonido lo percibimos promediando su nivel durante un intervalo de tiempo.

La necesidad de limitar el ancho de banda de la señal no supone ningún perjuicio para la misma, como pudiera parecer a primera vista. La frecuencia límite se puede extender al valor que deseemos, siempre que aumentemos de igual forma la frecuencia de muestreo. Dependiendo de la aplicación, podemos utilizar frecuencias de muestreo desde 8 kHz a 192 kHz. Nada es gratis, y la frecuencia de muestreo también determina las características de los dispositivos electrónicos de muestreo y las demandas de datos en transmisión y almacenamiento. A medida que aumentemos la frecuencia de muestreo necesitaremos dispositivos de muestreo más rápidos y mayores demandas de almacenamiento y transmisión. Existen factores económicos a considerar en el diseño. Los fabricantes seleccionaron la frecuencia de muestreo de

44,100 kHz a raíz del disco compacto, por ejemplo, por el tamaño del soporte, por el tiempo de reproducción y por su coste. Por otro lado, en el disco de audio DVD se utilizan frecuencias de muestreo de hasta 192 kHz.

Todo el proceso de muestreo (incluida la reconstrucción de la señal a partir de sus muestras) se presenta en la Figura 2.2. En ella aparecen las señales que existen en distintos puntos del proceso. En la parte izquierda de la figura se muestran las señales en el dominio del tiempo y en la parte derecha en el dominio de la frecuencia. En otras palabras, podemos observar las variaciones de amplitud en el dominio del tiempo y su respuesta en frecuencia. Por ejemplo, en las Figuras 2.2A y 2.2B se puede apreciar cómo la señal de audio está limitada en banda, con un filtro paso-bajo, hasta la mitad de la frecuencia de muestreo $S/2$. Este filtro elimina todo el contenido espectral a partir de la frecuencia Nyquist, $S/2$ Hz. La señal de muestreo (Figuras 2.2C y 2.2D) se repite al ritmo marcado por la frecuencia de muestreo S , y su espectro está formado por impulsos, situados en las frecuencias 0, S , $2S$, $3S$, etc. Cuando se muestrea una señal de audio, como se muestra en las Figuras 2.2E y 2.2F, lo que se obtiene es la amplitud de la misma

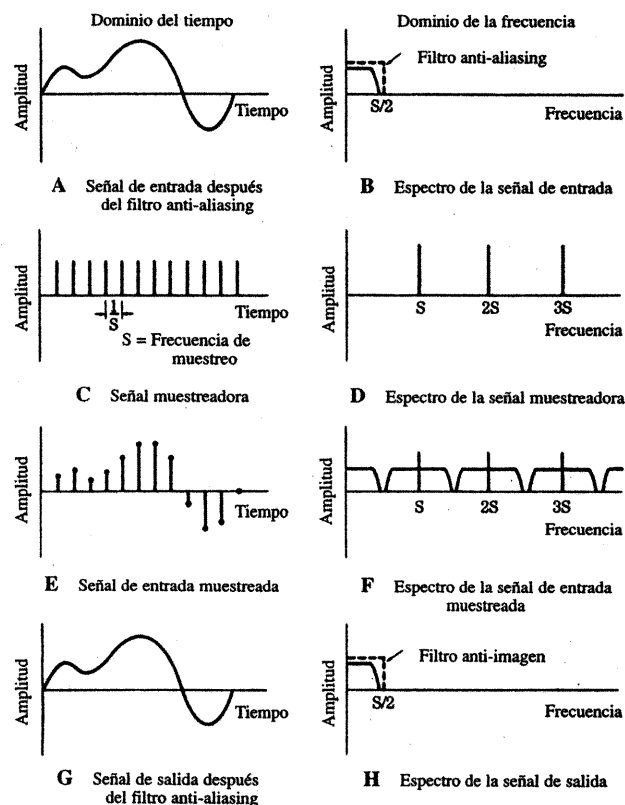


Figura 2.2. Señales en el dominio del tiempo (columna de la izquierda) y en el dominio de la frecuencia (columna de la derecha) ilustrando el proceso de muestreo de señales de banda limitada y el proceso de reconstrucción.

en los instantes de muestreo; sin embargo, la respuesta en frecuencia de esta señal muestreada da lugar a réplicas del espectro de la señal original centradas en múltiplos de la frecuencia de muestreo. Para reconstruir la señal, como se muestra en las Figuras 2.2G y 2.2H, las muestras han de pasar por un filtro paso-bajo anti-imagen que elimine todas estas réplicas a partir de la frecuencia $S/2$. Este filtro se encarga de realizar una interpolación entre las muestras, reconstruyendo de esta manera la señal analógica de entrada (limitada en banda). Como se describe en el Capítulo 4, la respuesta al impulso del filtro de salida transforma los impulsos de entrada en una señal continua.

El teorema de muestreo es inequívoco: una señal limitada en banda puede ser muestreada; almacenada, transmitida o procesada en valores discretos; posteriormente, si se quiere reconstruir la señal analógica es necesario realizar un proceso de interpolación entre las muestras. No hay pérdida de información en el proceso de muestreo. Los teoremas de muestreo, como el teorema de Nyquist, lo prueban de forma concluyente. Por supuesto, después de muestrear la señal, el sistema digital debe asignar valores numéricos a las amplitudes de los impulsos (amplitud de la señal en los instantes de muestreo). Este proceso, la cuantificación, se estudiará más adelante en este capítulo. Para un estudio más detallado del muestreo discreto, con una demostración matemática concisa, consulte el apéndice.

ALIASING

El aliasing es un fenómeno anómalo que aparece en el proceso de muestreo. Al igual que un criminal puede tomar varios nombres para confundir su identidad, el aliasing puede crear falsas componentes de una señal. Estas componentes aparecen dentro del ancho de banda de la señal y son indistinguibles de las verdaderas. Obviamente, es responsabilidad del diseñador prevenir tal fenómeno para que éste nunca ocurra. En la práctica, el aliasing no supone una limitación seria. Simplemente enfatiza la importancia del teorema del muestreo.

Anteriormente hemos visto que, bajo ciertas condiciones, el muestreo es un proceso sin pérdidas. La condición más importante es que la señal sea de banda limitada; es decir, debe colocarse un filtro paso-bajo antes del proceso de muestreo. Si no se hace este filtrado, la señal podría llegar a estar inframuestreada. Considere el siguiente experimento conceptual: tome una cámara de cine y filme mientras me desplazo con mi moto BMW. En la película, cuando acelero, los radios de las ruedas giran hacia adelante, después giran lentamente hasta que se paran, luego giran en sentido inverso, aumentando la velocidad de giro hasta que nuevamente éste disminuye y se para, para empezar otra vez a girar hacia adelante. Éste es un ejemplo típico de aliasing. La cámara, con una frecuencia de 24 fotogramas por segundo, es incapaz de capturar los cambios de velocidad que se producen en las ruedas.

El aliasing aparece cuando se viola el teorema de muestreo. La frecuencia más alta de una señal de audio debe ser igual o inferior a la frecuencia Nyquist. Si es más grande que la frecuencia Nyquist aparecerá el fenómeno del aliasing. Cuando la frecuencia de la señal de audio se incrementa, disminuye el número de muestras por ciclo. Al alcanzar la frecuencia Nyquist, existen dos muestras por ciclo, el mínimo número de muestras para registrar el carácter bipolar de la señal. Con frecuencias más altas, el proceso de muestreo continuará creando muestras a su ritmo (frecuencia de muestreo), pero estas muestras representan una información errónea de la señal, manifestándose como frecuencias de aliasing. Específicamente, si S es la frecuencia de muestreo, F es una frecuencia mayor que la mitad de la frecuencia de muestreo, y N es un número entero, entonces las frecuencias de aliasing que se crean, F_f , tienen un valor de $F_f = \pm NS \pm F$. En otras palabras, las frecuencias de aliasing aparecen en el ancho de banda de la señal (y en las réplicas de este ancho de banda), plegadas desde la frecuencia de muestreo. Aunque el fenómeno pueda parecer preocupante, no es sorprendente. El muestreo es realmente una modulación; de hecho, el proceso de muestreo es equivalente a una demodulación heterodina de AM (ampli-

tude modulation). La señal de un oscilador se multiplica por la señal de radio para desplazar el espectro a una frecuencia estándar IF (*intermediate frequency*). Aunque éste es el efecto que se desea en un receptor de radio, en el proceso de muestreo no debe aparecer nunca.

Considere un sistema de digitalización que tenga una frecuencia de muestreo de 44 kHz. Suponga que una señal de 36 kHz entra en el circuito encargado de realizar el muestreo, tal como se muestra en la Figura 2.3. La primera componente producida por el aliasing corresponde a $S - F = F_f$ o, lo que es lo mismo, a $44 - 36 = 8$ kHz. El circuito muestreador produce muestras incorrectas aunque registre de forma fidedigna los valores de amplitud de la señal en los instantes de muestreo. Dadas estas muestras, no existe ningún dispositivo capaz de determinar cuál es la frecuencia correcta: 36 kHz u 8 kHz. Recuerde además que existe un filtro paso bajo al final del proceso de digitalización, con objeto de suavizar la forma escalonada que presenta la señal, reconstruyendo perfectamente la misma (original). La frecuencia de corte de este filtro es la frecuencia de Nyquist, eliminado todo contenido espectral superior a este valor. En el caso anterior, de acuerdo con las características del filtro paso-bajo de salida, la componente de 36 kHz desaparece y sólo permanece la componente alias de 8 kHz, como si realmente se tratara de una señal de 8 kHz. Esta componente produce una distorsión en la señal de audio.

Existen otras manifestaciones del aliasing. Aunque en el margen de frecuencias de audio sólo aparece una componente interferente de valor $S - F$, siempre aparecerá una componente alias, independientemente de lo grande que sea F . Considere una frecuencia de muestreo de 44 kHz; una sinusoide de frecuencia variable creciente sonará correctamente en el intervalo 0-22 kHz, pero en el intervalo 22-44 kHz el aliasing producirá una sinusoide cuya frecuencia desciende desde 22 a 0 kHz. Si la sinusoide varía entre 44 y 66 kHz, el sonido resultante equivale a una sinusoide entre 0 y 22 kHz, y así sucesivamente.

Las componentes de aliasing no sólo ocurren alrededor de la frecuencia de muestreo, sino también en múltiplos de la misma (ver Figura 2.2F). Cuando se cumple el teorema de muestreo, la banda de audio y todas sus infinitas imágenes están completamente separadas entre sí, tal

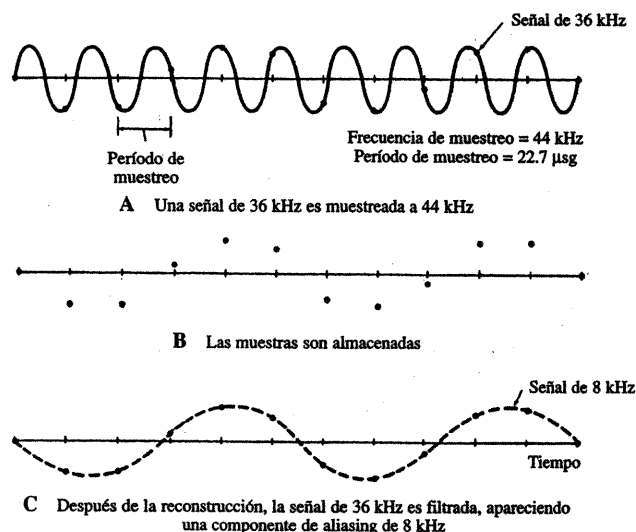


Figura 2.3. Una señal de entrada con una frecuencia superior a la mitad de la frecuencia de muestreo generará un componente de aliasing, de menor frecuencia.

como se muestra en la Figura 2.4A y 2.4B. Sin embargo, cuando la banda de audio supera la frecuencia Nyquist, las bandas imagen se solapan, apareciendo el fenómeno del aliasing, tal como se muestra en la Figura 2.4C y D. Todas estas componentes producirían valores: $\pm S \pm F$, $\pm 2S \pm F$, $\pm 3S \pm F$, etc. Por ejemplo, dada una frecuencia de muestreo de 44 kHz y una señal de entrada de 27 kHz, algunas de las componentes de aliasing serían: 17, 61, 71, 105, 115, 149, 159 kHz, tal como se muestra en la Figura 2.4D.

Prevención

En la práctica, el problema del aliasing es fácilmente superable. De hecho, en un sistema de digitalización el aliasing nunca ocurre. La solución es inmediata: la señal de entrada ha de limitarse en banda con un filtro paso-bajo (filtro antialiasing), que proporcione una atenuación significativa a partir de la frecuencia Nyquist, para asegurar que el contenido espectral de la señal que se muestrea nunca supere este valor. Un filtro ideal antialiasing produciría una atenuación infinita en la banda atenuada y de forma instantánea. En la práctica, los filtros antialiasing tienen una banda de transición por encima de la frecuencia Nyquist y una atenuación en la banda

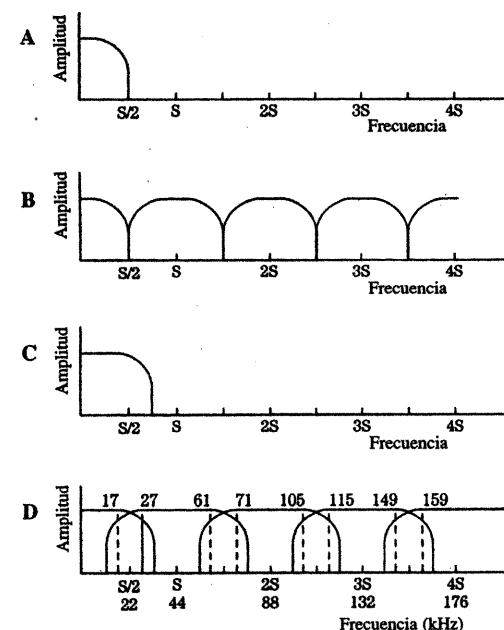


Figura 2.4. Representación espectral del muestreo y del aliasing. **A.** La señal de entrada se limita en banda hasta la frecuencia de Nyquist. **B.** Después de la reconstrucción, aparecen imágenes del espectro en múltiplos de la frecuencia Nyquist. **C.** La señal de entrada no está limitada en banda. **D.** Después de la reconstrucción, las imágenes del espectro están fuera de los múltiplos de la frecuencia Nyquist: este solapamiento espectral es de aliasing; por ejemplo, una señal de 27 kHz producirá componentes de aliasing en un circuito muestreador que trabaje con una frecuencia de muestreo de 44 kHz.

atenuada suficiente para que las componentes de aliasing sean inferiores a la resolución del convertidor A/D. Realmente, como se describe en el Capítulo 3, la mayoría de los sistemas utilizan un convertidor A/D con sobremuestreo, con un filtro paso-bajo de características poco exigentes, una frecuencia de muestreo muy alta y un posterior proceso de diezmado para eliminar el aliasing. Con este proceso se asegura que el sistema cumpla el teorema de muestreo. De esta forma, el problema del aliasing nunca llega a aparecer.

El teorema de muestreo y el filtrado paso-bajo de la señal de entrada debe cumplirse siempre en cualquier caso. Si se permite que aparezca el fenómeno del aliasing, no habrá forma alguna de eliminar las componentes interferentes que se hayan introducido en la banda de audio.

CUANTIFICACIÓN

La medida de la variación de un suceso tiene sentido si se registra tanto el valor de su amplitud como el instante de tiempo en el que se realiza la medida. El muestreo representa los instantes de medida, y la cuantificación representa los valores de la medida, o en el caso del audio, la amplitud de la señal en los instantes de muestreo. El muestreo y la cuantificación son, por tanto, los dos pilares básicos de la digitalización, caracterizando por completo al fenómeno acústico. El muestreo y la cuantificación son dos variables que determinan, respectivamente, el ancho de banda y la resolución de amplitud. Una señal analógica puede representarse mediante una serie de pulsos; la amplitud de cada pulso se expresa con un número que indica el valor de la señal analógica en ese instante. Con la cuantificación, al igual que con cualquier otra medida, la precisión final está determinada por la resolución del sistema. La resolución en un sistema digital de audio está determinada por la longitud binaria de los datos; por tanto, siempre se introduce un cierto error. Este error es parecido al ruido de fondo de los sistemas analógicos; sin embargo, en la percepción es un tipo de ruido más perjudicial, ya que varía en función de la amplitud de la señal de entrada.

En una cuantificación uniforme, la amplitud de la señal se transforma en un determinado número de niveles de cuantificación, todos ellos de igual tamaño. Los infinitos valores que existen en la amplitud de una muestra deben cuantificarse en un número finito de niveles de cuantificación; esto acarrea un cierto error. Para representar una señal con una calidad óptima es necesario tener una gran cantidad de niveles de cuantificación; por ejemplo, una señal de audio de alta calidad puede precisar 65.536 niveles de cuantificación, e incluso más. Sin embargo, con sólo unos cuantos niveles de cuantificación también es posible codificar parte de la información de la señal; por ejemplo, con sólo dos niveles de cuantificación se puede codificar la inteligibilidad de una señal de voz.

Considere dos voltímetros, uno analógico y otro digital, realizando una medida de tensión de una señal. Con un buen indicador de nivel y con una buena vista, se podría tener, por ejemplo, una lectura de 1,27 V (voltios). Un medidor digital con tan sólo dos dígitos podría dar un valor de 1,3 V. Con tres dígitos la lectura sería de 1,27, y con cuatro sería de 1,274. Tanto en la medida analógica como en la digital existe un error. El error en el sistema analógico depende del propio mecanismo de lectura (aguja) y de la posible dificultad de la misma. Incluso con condiciones ideales, la resolución de una medida analógica depende del ruido que pueda introducir el equipo.

En un medidor digital, la naturaleza del error es diferente. La precisión está limitada por la resolución del medidor, es decir, por el número de dígitos. A mayor número de dígitos mayor precisión; en el último dígito se hace un redondeo del valor exacto; por ejemplo, 1,27 se redondearía a 1,3. En el mejor de los casos, la medida podría corresponder verdaderamente con el valor real; por ejemplo, un voltaje de 1,3000 podría tener una lectura de 1,3. En el peor de los casos, el redondeo realizado podría ser justamente la mitad del intervalo; por ejemplo, una tensión de 1,250 voltios se redondearía a 1,2 o a 1,3. Análogamente, cuando se utiliza el sistema

binario para hacer una medida, el error de resolución del sistema corresponde a la mitad del bit menos significativo, LSB (*least significant bit*). En sistemas analógicos y digitales, cuando se realiza la medida de un fenómeno analógico como, por ejemplo, la amplitud, siempre aparece un error. Hablando específicamente de voltímetros, una medida digital siempre es una medida mucho más robusta. Siempre se tiene más información del fenómeno analógico si éste está caracterizado en términos de datos digitales. Hoy en día, la utilidad de un voltímetro analógico es comparable a la de una regla de cálculo.

La cuantificación es, por tanto, una técnica que mide señales de audio con datos numéricos. Un sistema digital utiliza un sistema de números binarios. El número de valores binarios depende de la longitud binaria de los datos, esto es, el número de bits que forman un dato de audio. La resolución en un grabador digital de audio depende de la longitud binaria de los datos (número de bits), al igual que en un voltímetro digital dependía del número de dígitos de la medida. El número de bits es un indicador de precisión, aunque puedan existir otro tipo de limitaciones. En la práctica, la resolución depende fundamentalmente del conversor A/D.

Teóricamente el muestreo es un proceso que no produce pérdidas, pero no así la cuantificación, es decir, la asignación del valor de amplitud en cada una de las muestras. No importa cuál sea la escala o el código que se utilice, con una digitalización no se puede codificar de forma perfecta una señal analógica. Una señal tiene infinitos valores de amplitud, mientras que en un cuantificador el número de escalones de cuantificación es finito. Todos los valores de amplitud dentro de un intervalo de cuantificación son codificados con un mismo número. Por tanto, un valor cuantificado no deja de ser una simple aproximación del valor real.

Relación señal a ruido

En un sistema binario, la longitud binaria de la palabra determina el número de intervalos de cuantificación; este valor es 2 elevado a longitud de la palabra. Dicho de otra forma, una palabra de n bits produce 2^n valores de cuantificación. En la Tabla 2.1 se muestran los valores de cuantificación para longitudes binarias de palabra entre 1 y 24 bits. Por ejemplo, a una palabra de 8 bits le corresponde $2^8 = 256$ intervalos, mientras que a una de 16 bits le corresponde $2^{16} = 65.536$ intervalos. Como se aprecia en la tabla, cada vez que se aumenta en un bit la longitud de la palabra se duplica el número de niveles de cuantificación. A medida que se aumenta el número de bits se obtiene una mejor aproximación; pero como se ha señalado, siempre existe un error aso-

Tabla 2.1. El número (N) de intervalos de cuantificación en una palabra binaria es $N = 2^n$ donde n es el número de bits de la palabra

| | |
|-----------------|---------------------|
| $2^1 = 2$ | $2^{13} = 8192$ |
| $2^2 = 4$ | $2^{14} = 16384$ |
| $2^3 = 8$ | $2^{15} = 32768$ |
| $2^4 = 16$ | $2^{16} = 65536$ |
| $2^5 = 32$ | $2^{17} = 131072$ |
| $2^6 = 64$ | $2^{18} = 262144$ |
| $2^7 = 128$ | $2^{19} = 524288$ |
| $2^8 = 256$ | $2^{20} = 1048576$ |
| $2^9 = 512$ | $2^{21} = 2097152$ |
| $2^{10} = 1024$ | $2^{22} = 4194304$ |
| $2^{11} = 2048$ | $2^{23} = 8388608$ |
| $2^{12} = 4096$ | $2^{24} = 16777216$ |

ciado al proceso de cuantificación, ya que el número de niveles codificados por una palabra binaria nunca podrá llegar a abarcar los infinitos valores existentes de una amplitud analógica.

Es difícil de apreciar la precisión que se puede obtener con 16 bits. Una analogía nos puede ayudar: en una columna de 6.5 metros, construida con paquetes de folios, cada uno de los folios representaría un intervalo de cuantificación en un sistema binario de palabras de 16 bits. Con longitudes mayores, la analogía es incluso más impresionante. En sistemas de 20 bits, la columna tendría 105 metros de altura. En sistemas de 24 bits, la columna llegaría a casi 2 km de altura. El cuantificador podría medir los 2 km con una precisión equivalente al espesor de una página de un libro. Si elimináramos una sola página, sería equivalente a cambiar el bit menos significativo de 1 a 0. Mirémoslo de otra manera: si la distancia entre Nueva York y Los Ángeles se midiera con un cuantificador de 24 bits de resolución, la precisión de la medida rondaría los 20 cm. Un sistema digital de audio de alta calidad requiere una precisión semejante, y no es una cuestión baladí.

A partir de un punto, el error de cuantificación empieza a ser inaudible. La mayoría de los fabricantes han llegado al acuerdo de que con 16 o 20 bits es suficiente para tener una correcta representación del audio; sin embargo, no deben descartarse palabras de longitud superior, o técnicas de procesamiento digital con objeto de optimizar la cuantificación y reducir así el error de cuantificación.

La longitud binaria de la palabra determina la resolución de la digitalización y, por lo tanto, proporciona una cierta medida de la calidad del sistema. Algunas veces el intervalo de cuantificación podrá ser exactamente el valor analógico; normalmente esto no sucederá. En el peor de los casos, el valor analógico se desviará la mitad del intervalo de cuantificación, es decir, el error será la mitad del bit menos significativo de la palabra cuantificada. Por ejemplo, considere la Figura 2.5. Suponga que la palabra binaria 101000 corresponde al valor analógico 1,4 V, 101001 corresponde a 1,5 V y el valor analógico en el instante de muestreo es, desafortunadamente, 1,45 V. El cuantificador debe redondear a 101001 o a 101000; el código 101000½ no existe. En cualquiera de los dos códigos habrá un error igual a la mitad del intervalo de cuantificación.

El error de cuantificación es la diferencia entre el valor real analógico en el instante de muestreo y el valor del intervalo de cuantificación asignado. En el instante de muestreo, la amplitud se redondea al intervalo de cuantificación más próximo, tal como se muestra en la Figura 2.6. En el mejor de los casos (instantes de muestreo 11 y 12 de la figura), la onda coincidirá con los intervalos de cuantificación. En el caso más desfavorable (instante de muestreo 1 de la figura), la

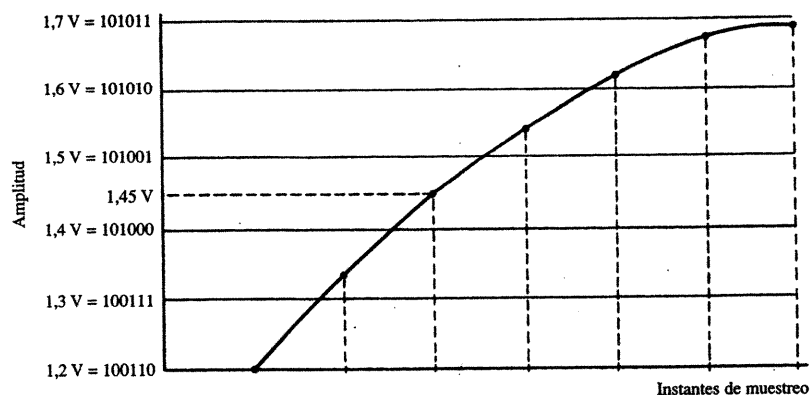


Figura 2.5. El error de cuantificación está limitado a la mitad del bit menos significativo.

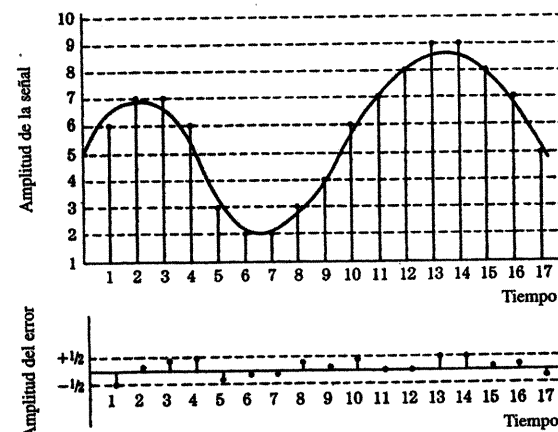


Figura 2.6. Error de cuantificación en los instantes de muestreo.

onda estará exactamente entre los dos intervalos de cuantificación. Por tanto, el error de cuantificación está limitado a un margen de valores comprendido entre $+Q/2$ y $-Q/2$, donde Q es el intervalo de cuantificación. Este proceso de selección, entre uno y otro intervalo, es el proceso básico de cuantificación, y ocurre en cada una de las muestras que se digitalizan. Además, el error de cuantificación siempre es inferior o igual a $1/2$ LSB. Este error representa una distorsión que siempre está presente en cualquier amplitud de la señal de audio. Cuando la señal tiene un nivel alto, la distorsión es relativamente pequeña y queda enmascarada. Sin embargo, cuando la señal es pequeña, la distorsión es relativamente grande y puede llegar a ser audible.

Para caracterizar la calidad de un equipo digital, se determina la relación entre la máxima amplitud de la señal y el máximo error de cuantificación; a esta relación se la denomina relación S/E (señal/error) del sistema. La relación señal/error de un sistema digital es similar, pero no idéntica, a la relación señal/ruido (S/N) de un sistema analógico. La relación S/E se puede calcular a partir de los niveles de tensión de la señal y del error.

Considere un cuantificador en donde n sea el número de bits y N sea el número de pasos de cuantificación. Como ya se señaló:

$$N = 2^n$$

La mitad de estos 2^n valores se utilizan para codificar sólo una parte de una onda bipolar. Si Q es el intervalo de cuantificación, los valores máximos que se pueden representar en una señal son $\pm Q2^{n-1}$. Suponiendo una señal sinusoidal de entrada, el valor máximo RMS (valor cuadrático medio) de la señal S_{RMS} es:

$$S_{RMS} = \frac{Q2^{n-1}}{(2)^{1/2}}$$

También es posible determinar la energía del error de cuantificación. Cuando la señal de entrada tiene una amplitud de valor alto y su espectro es de banda ancha, el error de cuantificación es estadísticamente independiente, y uniformemente distribuido en el intervalo $+Q/2$ y $-Q/2$, incluyendo el valor nulo, donde Q es el intervalo de cuantificación. Esto supone una función densidad de probabilidad de valor $1/Q$; el error tiene un carácter aleatorio de muestra a

muestra; el espectro del error es plano. Ignorando el error fuera de la banda de frecuencias de la señal, el valor RMS del error, E_{RMS} , se obtiene sumando (integrando) el error cuadrático por su probabilidad:

$$E_{rms} = \left[\int_{-\infty}^{+\infty} e^2 p(e) de \right]^{1/2} = \left[\frac{1}{Q} \int_{-Q/2}^{+Q/2} e^2 de \right]^{1/2} = \left[\frac{Q^2}{12} \right]^{1/2} = \frac{Q}{(12)^{1/2}}$$

La relación de potencias entre la señal y el error de cuantificación es:

$$\frac{S}{E} = \left[\frac{S_{rms}}{E_{rms}} \right]^{1/2} = \left[\frac{\left[\frac{Q2^{n-1}}{(2)^{1/2}} \right]^2}{\left[\frac{Q}{(12)^{1/2}} \right]^2} \right]^{1/2} = \frac{3}{2} (2^{2n})$$

Expresando esta relación en decibelios:

$$\frac{S}{E} (dB) = 10 \log \left[\frac{S_{rms}}{E_{rms}} \right]^{1/2} = 20 \log \left[\left(\frac{3}{2} \right)^{1/2} (2^n) \right] = 6,02n + 1,76$$

Podemos observar, sustituyendo valores en la expresión anterior, que un cuantificador ideal de 16 bits produce una relación señal/error de unos 98 dB, y uno de 15 bits produce unos 92 dB. En otras palabras, cada bit adicional reduce el ruido de cuantificación en 6 dB, o en un factor de dos. La tasa binaria de la señal de audio (bits/segundo) se incrementa proporcionalmente con número de bits de los datos. Sin embargo, el aumento de la relación señal con respecto a la tasa binaria es exponencial. Ésta es una relación eficiente que se aproxima a un máximo teórico, característico en los sistemas de codificación como, por ejemplo, en el sistema de codificación PCM (modulación de pulsos codificados), descrito en el Capítulo 3. El valor 1,76 de la expresión anterior se obtiene considerando el comportamiento estadístico de los valores de la señal —factor de cresta: (valor de pico)/(valor RMS)—, en este caso, el de una senoide; este valor diferirá con otro tipo de señal.

También es importante destacar que el modelo supone una distribución uniforme del ruido de cuantificación, y que la cuantificación es lo suficientemente precisa para eliminar toda posibilidad de correlación con la propia señal de audio. Esto es normalmente cierto para señales complejas de gran nivel, donde las componentes de distorsión que aparecen están totalmente incorreladas; estas componentes se extienden en todo el margen audible de audio, y son percibidas como una señal de ruido blanco. Sin embargo, no es el caso para señales de bajo nivel, donde la distorsión generada puede ser debida a productos de intermodulación.

Distorsión de cuantificación

Un análisis del error de cuantificación en señales de bajo nivel revela que el espectro del mismo depende de la señal de entrada. El error no se comporta como un ruido aleatorio (como en las señales de nivel alto); está correlado con la propia señal. En la señal de salida del sistema, cuando se ha recuperado nuevamente la señal analógica a partir de las muestras cuantificadas, pueden aparecer componentes periódicas (de banda estrecha). Estas componentes no pueden considerarse como ruido, ya que el error de cuantificación es una función de la señal de entrada; más bien debemos hablar de distorsión.

Como ya se señaló anteriormente, cuando el error de cuantificación es aleatorio de muestra a muestra, el valor RMS toma la expresión $E_{RMS} = Q/(12)^{1/2}$. Esta expresión demuestra que el error de cuantificación es independiente de la señal de entrada y sólo depende del intervalo de cuantificación; a mayor número de intervalos menor distorsión. Sin embargo, este número de intervalos no es únicamente los intervalos que tenga el cuantificador sino también los intervalos utilizados para cuantificar una determinada señal. Con una señal cuya amplitud oscile entre los valores máximos del cuantificador (como se ha considerado en el análisis anterior) tenemos el mejor de los escenarios posibles, ya que se utilizan todos los intervalos de cuantificación. Sin embargo, cuando el nivel de la señal decrece, se van utilizando menor número de intervalos, tal como se muestra en la Figura 2.7. Por ejemplo, dado un cuantificador de 16 bits, una señal con una amplitud la mitad del máximo posible del cuantificador utilizaría también la mitad de los intervalos disponibles. En vez de 65.536 niveles, utilizaría 32.767 intervalos. En otras palabras, estaría siendo cuantificada con una resolución de 15 bits. El problema se agudiza a medida que decae el nivel de la señal. En niveles muy bajos, por ejemplo, podríamos estar cuantificando con sólo un bit o incluso podríamos estar perdiendo la señal (no cuantificada). Dicho de otra forma, cuando el nivel de la señal decrece, el porcentaje de distorsión aumenta. Aunque este valor pueda ser extremadamente pequeño con una señal de entrada a fondo de escala (0 dB), el aumento es significativo en señales de muy bajo nivel (por ejemplo, -90 dB). Como se describe en la siguiente sección, debe aplicarse *dither* para solventar este problema.

El nivel mínimo del error en un sistema de audio digital es distinto del nivel mínimo de ruido (ruido de fondo) de un sistema analógico, ya que en un sistema digital el error depende de la señal de entrada. La naturaleza del error de cuantificación varía de acuerdo con la amplitud y con las características de la señal de audio. Para señales de banda ancha, señales de entrada con niveles altos (señales que normalmente encontramos en la música), el error de cuantificación se percibe como ruido blanco. Una señal compleja de nivel alto puede tener algún tipo de compor-

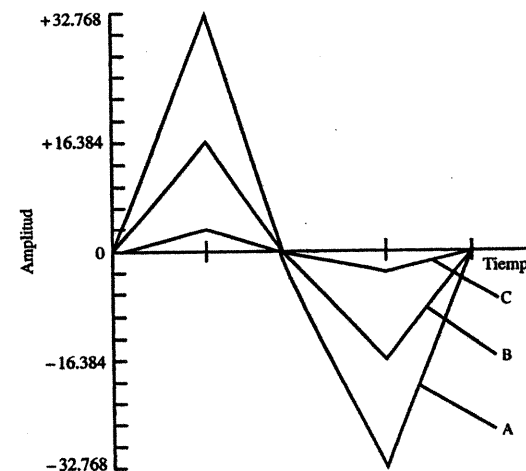


Figura 2.7. El porcentaje del error de cuantificación se incrementa cuando disminuye el nivel de la señal. La señal A a fondo de escala tiene relativamente un error mínimo (16 bits de resolución). La señal B a mitad de fondo de escala tiene un mayor error (resolución efectiva de 15 bits). La señal C de bajo nivel tiene un gran error.

tamiento periódico muestra a muestra; sin embargo, este comportamiento periódico no se manifiesta en el error de cuantificación. Cuando la señal es compleja y de nivel alto, el error de cuantificación es independiente de la señal de entrada (asumiendo que ésta tenga una naturaleza aleatoria). La única diferencia con respecto al ruido de los sistemas analógicos es que el margen de valores está más restringido para un mismo valor RMS. En otras palabras, todos los valores tienen la misma probabilidad de ser positivos o negativos en el margen considerado. Por el contrario, el ruido de los sistemas analógicos tiene una distribución estadística gaussiana y los valores de pico son mucho mayores que su valor RMS.

Sin embargo, las cualidades de percepción del error de cuantificación son menos benignas con señales de bajo nivel o con señales de nivel alto pero de banda estrecha (marcadamente periódicas). Esta afirmación está basada en el hecho de que el ruido blanco es perceptualmente benigno, al ser aleatorios todos los valores, mientras que las señales con un comportamiento previsible se perciben más fácilmente. En señales de banda ancha y nivel alto, la correlación entre las muestras es muy pobre; sin embargo, la correlación aumenta en señales de bajo nivel y en señales de banda estrecha. A medida que se incrementa la correlación entre las muestras, el error de cuantificación, percibido inicialmente como un ruido blanco benigno, se va transformando en una compleja forma de distorsión.

El error de cuantificación se puede presentar de varias formas. Por ejemplo, la señal cuantificada puede contener componentes por encima de la frecuencia Nyquist; por lo tanto, aparecerán componentes de aliasing. Estas componentes, aunque aparecen después del circuito de muestreo, han sido ya muestreadas. Como apunta Barry Blesser, no existe ninguna diferencia entre muestrear una señal antes o después de pasar por un limitador de nivel. Si la señal es de nivel alto y compleja, las componentes de aliasing se mezclarán con el resto de errores de la señal, errores aleatorios. Si la señal es de nivel bajo y simple, las componentes de aliasing pueden ser perfectamente audibles. Considere un sistema con una frecuencia de muestreo de 50 kHz, limitado en banda a 25 kHz. Cuando se cuantifica una sinusoide de 7 kHz de amplitud similar a un intervalo de cuantificación, ésta se convierte en una señal cuadrada. Aparecen armónicos en 21, 35 y 49 kHz. Estos dos últimos armónicos aparecen como componentes alias en 15 y en 1 kHz, respectivamente. De forma similar, el resto de armónicos de la onda cuadrada aparecen también como componentes de aliasing dentro de la banda de audio.

Las componentes de aliasing producidas por la cuantificación se conocen como ruido granular, así llamado por su parecido al ruido que produce la arena al caer al suelo. En señales de nivel alto, el ruido es enmascarado por la propia señal. Sin embargo, en niveles bajos, el ruido es audible. Esta mezcla de ruido (ruido de modulación y de distorsión) no aparece en los sistemas analógicos y su presencia es bastante desagradable (audiblemente hablando). Además, si las componentes de aliasing que aparecen se encuentran próximas a la frecuencia de muestreo, pueden crearse nuevos tonos por intermodulación, produciendo un sonido característico conocido por el nombre «pajaritos» (*bird singing*). Si observamos el error de cuantificación en un tono de amplitud variable, en niveles altos tiene un carácter de ruido blanco y conforme se disminuye el nivel del tono el ruido se va transformando en componentes de distorsión. El problema se agrava porque incluso en señales periódicas musicales, a medida que se disminuye el nivel, tienden a tener un carácter sinusoidal. Además, el tono cuyo nivel se extingue tiende a modular en amplitud las componentes de distorsión. Con el *dither*, como se verá más adelante, es posible eliminar este tipo de error.

Otras arquitecturas

La cuantificación no sólo depende de la longitud binaria de los datos; hay que considerar también el dispositivo que realiza dicha función. Existen distintas estrategias para realizar la asignación de los niveles de cuantificación en una señal de audio. Por ejemplo, un cuantificador

puede distribuir todos los intervalos de cuantificación de una forma lineal o no lineal, según el caso. Otra alternativa es la modulación delta; el cuantificador utiliza un único bit (bit de signo) para codificar toda la amplitud de la señal. También se pueden emplear técnicas de sobremuestreo y de conformación de ruido (*noise-shaping*) para extraer el ruido de la banda de audio y llevarlo a frecuencias superiores. La bondad de los algoritmos siempre determina la eficiencia de la codificación binaria, así como la percepción auditiva del error. Por ejemplo, como ya se comentó anteriormente, un cuantificador lineal presenta un gran error de cuantificación en los niveles bajos de señal, cuando la amplitud está confinada en sólo unos cuantos intervalos de cuantificación. Un sistema no lineal que utilizara un cuantificador en coma flotante podría aumentar la amplitud de estas señales de bajo nivel, con objeto de aumentar el número de intervalos de cuantificación. Esto supondría una mejora en la relación señal/error pero produciría, por otra parte, ruidos de intermodulación, totalmente indeseables. Históricamente, después de examinar los inconvenientes que presentan cada uno de los sistemas, los fabricantes se han decantado por cuantificadores de tipo lineal y con intervalo de cuantificación fijo; son los que mejor se ajustan a las características de las señales de música. No obstante, con los nuevos sistemas de codificación perceptual se pone en entredicho esta afirmación. En el Capítulo 4 se examinarán estos sistemas alternativos de cuantificación. La codificación perceptual se verá en el Capítulo 10.

DITHER

En señales de audio complejas y de nivel alto apenas existe correlación entre el error de cuantificación y la señal de audio; el error puede considerarse aleatorio y perceptualmente similar al ruido blanco. En señales de nivel bajo, el carácter aleatorio del error cambia y puede llegar a estar correlado con la señal de audio; el error en este caso se manifiesta en forma de distorsión, potencialmente audible. Un sistema de digitalización debe suprimir estos fenómenos anómalos que se producen en el error de cuantificación. Obviamente, la solución más inmediata es aumentar el número de bits de los datos, produciendo una disminución del error de 6 dB por cada nuevo bit. Esta solución es antieconómica y podrían ser necesarios bastantes bits para llevar al error a niveles inaudibles. Además, el error siempre es significativo en los niveles bajos de la señal.

El *dither* es una técnica mucho más eficiente. El *dither* es un ruido de bajo nivel incorrelado con la señal de audio. El *dither* se añade a la señal de audio antes de ser muestreada. Esto hace que el proceso de cuantificación sea algo más lineal en los niveles bajos. Cuando se añade *dither*, las amplitudes de la señal de audio se balancean en torno a los niveles de cuantificación. En lugar de aparecer réplicas periódicas de la señal cuantificada, se fuerza a que cada período de la señal sea diferente. El error de cuantificación se decorrela así de la señal y se aleatoriza su efecto. No obstante, aunque la distorsión se reduce drásticamente, el *dither* siempre añade algo de ruido en la señal. En cierto sentido, el *dither* es similar a la corriente bias de alta frecuencia que se utiliza en los sistemas de grabación magnética analógica.

El *dither* no enmascara el error de cuantificación; más bien, permite al sistema de digitalización codificar amplitudes inferiores al bit menos significativo, de forma parecida al modo en el que un sistema analógico es capaz de registrar niveles inferiores a su nivel de ruido. Un sistema digital con un nivel de *dither* óptimo puede superar fácilmente la relación señal/ruido que ofrece cualquier sistema analógico. Por otra parte, un sistema digital sin *dither* puede tener peores prestaciones que un sistema analógico, especialmente en niveles bajos. Para tener una digitalización de calidad es necesario sumar a la señal de audio, antes de que pase al convertidor A/D, una pequeña cantidad de *dither*. Además, en todas las operaciones de procesamiento digital, también es conveniente la utilización del *dither*, eliminando todos los posibles errores de cuantificación.

Considere una señal de audio con una amplitud en torno a un intervalo de cuantificación. La señal de entrada podría estar comprendida completamente dentro de un intervalo de cuantificación, lo que resultaría una señal cuantificada de valor constante —continua (dc)—, o podría por el contrario estar bailando alrededor del umbral de cuantificación produciendo una señal cuadrada, tal como se muestra en la Figura 2.8A y B. Esto demuestra que el cuantificador, en niveles bajos, puede actuar como un limitador de nivel; en otras palabras, aparece una gran distorsión. El efecto es completamente diferente cuando se suma *dither* a la señal de audio. El resultado se muestra en la Figura 2.8C y D; es una señal pulsante que preserva la información de la señal de audio. La señal cuantificada conmuta arriba y abajo, de acuerdo con la variación que proporciona la señal de *dither*, siguiendo el valor medio de la señal de entrada. La información está codificada en la variación de la anchura de los pulsos de la señal cuantificada. Este tipo de codificación se denomina modulación por anchura de pulsos, y preserva de forma precisa la forma de onda de la señal de entrada. El valor medio de la señal cuantificada oscila continuamente entre dos niveles, mitigando los efectos del error de cuantificación. De forma similar, el ruido analógico podría codificarse con una señal binaria; los valores 0 y 1 aparecerían en el bit menos significativo (LSB) en cada instante de muestreo, manteniendo en el espectro el carácter de ruido blanco. El resultado perceptual: señal original sumada con ruido, un resultado mucho más deseable que la anterior onda cuadrada.

Matemáticamente, cuando se emplea *dither*, el error de cuantificación deja de ser una función determinista de la señal de entrada para convertirse en una variable aleatoria de media nula. Dicho de otra forma, en lugar de cuantificar sólo la señal de entrada, lo que entra al cuantificador es la suma de la propia señal con el *dither*, dando éste un carácter aleatorio al error de cuantificación y dando también una cierta linealidad al proceso de cuantificación. Esta técnica se conoce por el nombre de «*dither* no sustractivo», ya que éste queda definitivamente inmerso en la señal de audio; el error total no es independiente de la señal, ni tampoco son independien-

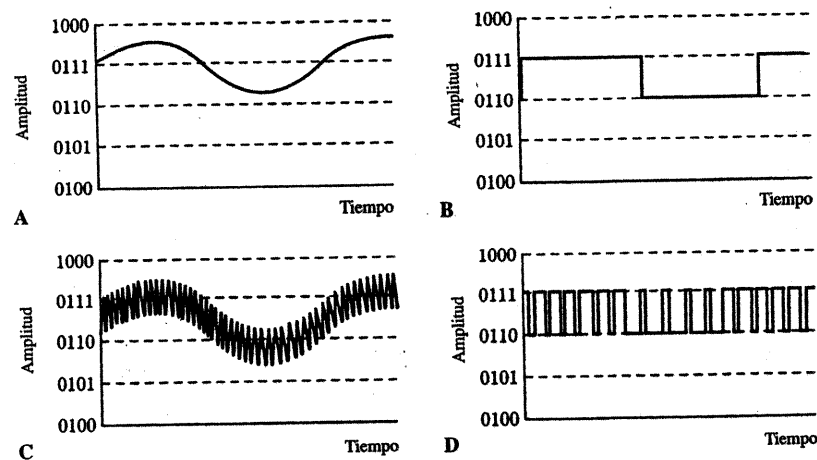


Figura 2.8. El *dither* se utiliza para mejorar los efectos del error de cuantificación. A. Una señal de entrada sin *dither*, con una amplitud equivalente a un LSB. B. Resultados de la cuantificación en una codificación de sólo 2 niveles. C. Una señal de entrada con *dither*. D. La cuantificación produce una forma de onda PWM, codificando información por debajo de un LSB.

tes entre sí los errores entre muestras sucesivas. Sin embargo, un *dither* no sustractivo no altera las propiedades estadísticas del cuantificador, dejando los momentos estadísticos del *error total* independientes de la señal de entrada, descorrelando de forma eficiente el error de cuantificación de las muestras de la señal, y descorrelando también el error entre muestras sucesivas. El espectro de potencia de la señal del *error total* se transforma en un espectro de ruido blanco. El «*dither* sustractivo», en el que el *dither* se elimina después de realizar la cuantificación, teóricamente proporciona una independencia estadística en la señal del error, pero es más difícil de implementar.

Jonh Vanderkooy y Stanley Lipshitz han demostrado las ventajas que ofrece el *dither* en una señal sinusoidal de 1 kHz con una amplitud de un LSB, tal como se muestra en la Figura 2.9. Cuando no se aplica *dither*, la salida del convertidor analógico-digital es una onda cuadrada. Si se aplica a la sinusoide de entrada un *dither* con una amplitud RMS de 1/3 LSB, aparece como resultado una onda modulada por anchura de pulsos. La sinusoide vuelve a aparecer cuando se promedia la onda resultante durante unos cuantos períodos. Es decir, la sinusoide está dentro de la señal de PWM (*pulse width modulation*). Este promedio es realizado realmente por nuestro oído; así percibimos las señales acústicas, el oído se comporta como un filtro paso-bajo. En este caso concreto, se escucharía una sinusoide sumada con ruido blanco en vez de una señal cuadrada.

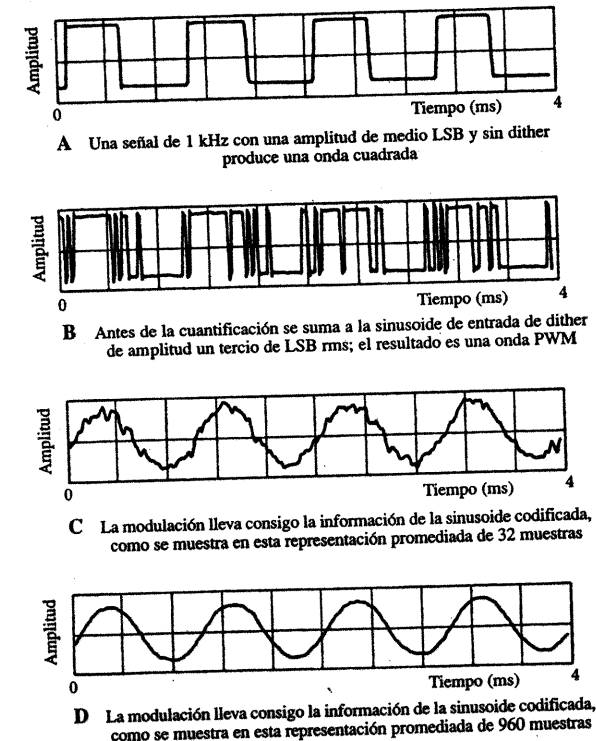


Figura 2.9. Con el *dither* es posible codificar señales cuyo nivel esté por debajo del bit menos significativo (Vanderkooy y Lipshitz).

El oído es capaz de diferenciar señales de banda estrecha con niveles inferiores al propio nivel de ruido; la membrana basilar del oído interno se encarga de realizar este tipo de filtrado. El oído se comporta como un banco de filtros de tercio de octava; el error de cuantificación, que tiene un carácter aleatorio por efecto del *dither*, es promediado por el oído y, por tanto, la sinusoide original de banda estrecha es percibida completamente sin distorsión. En otras palabras, el *dither* cambia la naturaleza digital del error de cuantificación por un ruido blanco, y así el oído es capaz de diferenciar señales con un nivel inferior a un LSB.

Esta conclusión es importante. Con *dither* la resolución del cuantificador es inferior al bit menos significativo; teóricamente no hay un límite en la resolución. Cuando se codifica una señal de audio con *dither*, modulando la señal cuantificada, es posible recuperar información de audio que esté incluso por debajo del intervalo de cuantificación más pequeño. Además, el *dither* puede eliminar la distorsión causada en la cuantificación, transformándola en ruido blanco. En la Figura 2.10 se prueba esta afirmación con una simulación por ordenador llevada a cabo por John Vanderkooy, Robert Wannamaker y Stanley Lipshitz. La figura muestra una sinusoide de 1 kHz de amplitud 4 LSB pico a pico. La primera columna muestra la señal sin *dither*. La segunda columna muestra la misma señal con un *dither* con una función densidad de probabilidad triangular y amplitud 2 LSB pico a pico. Para ambos casos, la primera fila muestra la señal de entrada. La segunda fila muestra la señal de salida. La tercera fila muestra el error total de cuantificación. La cuarta fila muestra el espectro de potencia de la señal de salida (estimado en ventanas tipo Hanning de 512 puntos, solapadas al 50 por 100 y con una frecuencia de muestreo de 44,1 kHz). La señal de salida sin *dither* (D) presenta distorsión armónica, visible en los múltiplos de la frecuencia de entrada, y distorsión no armónica producida por efecto del aliasing. En la señal de error (G) de la señal con *dither* se pueden observar las anomalías de la señal de entrada; no es una señal estadísticamente independiente. Sin embargo, sorprendentemente, esta señal de error suena como un ruido blanco (aunque claramente no se parezca en nada) y la señal de salida suena como una sinusoide con ruido. Esto se demuestra con el espectro de potencia (H), en el que se aprecia el aspecto plano del ruido blanco. No obstante, podemos ver también que el *dither* incrementa el nivel de ruido en la señal de salida.

Tipos de *dither*

Hay varios tipos de *dither*, diferenciados normalmente por su función pdf (densidad de probabilidad). Dada una señal aleatoria continua, la integral de la función densidad de probabilidad representa la probabilidad en un determinado intervalo. La probabilidad de que la señal esté comprendida en un intervalo de valores es justamente el área de la función pdf en ese intervalo. La probabilidad puede ser constante o variable. En aplicaciones de audio, el interés está centrado en tres tipos de *dither*: pdf gaussiano, pdf uniforme (rectangular) y pdf triangular, tal como se muestra en la Figura 2.11. Por ejemplo, se podría hablar de un *dither* aleatorio blanco, estadísticamente independiente, con una función pdf triangular de amplitud 2 LSB. Normalmente, las señales *dither* tienen un espectro blanco (plano); no obstante, éste puede moldearse aplicando algún tipo de correlación en las muestras, sin alterar la función pdf; por ejemplo, podríamos tener un *dither* pdf triangular paso alto. Los tres tipos de *dither* dan cierta linealidad a la curva de transferencia de cuantificación, pero los resultados son algo diferentes entre sí. Las señales de *dither* triangular y uniforme incrementan en menor medida el ruido en la señal de audio que el *dither* gaussiano, pero éste, por el contrario, es más fácil de implementar en el dominio analógico.

Las señales de *dither* triangular y rectangular son difíciles de implementar en el dominio analógico, con dispositivos electrónicos. Por ejemplo, habría que utilizar un generador digital de números pseudoaleatorios y aplicar esta secuencia de datos a un convertidor digital-analógico. Por tanto, con un diseño electrónico, es más fácil generar un ruido gaussiano que se pueda

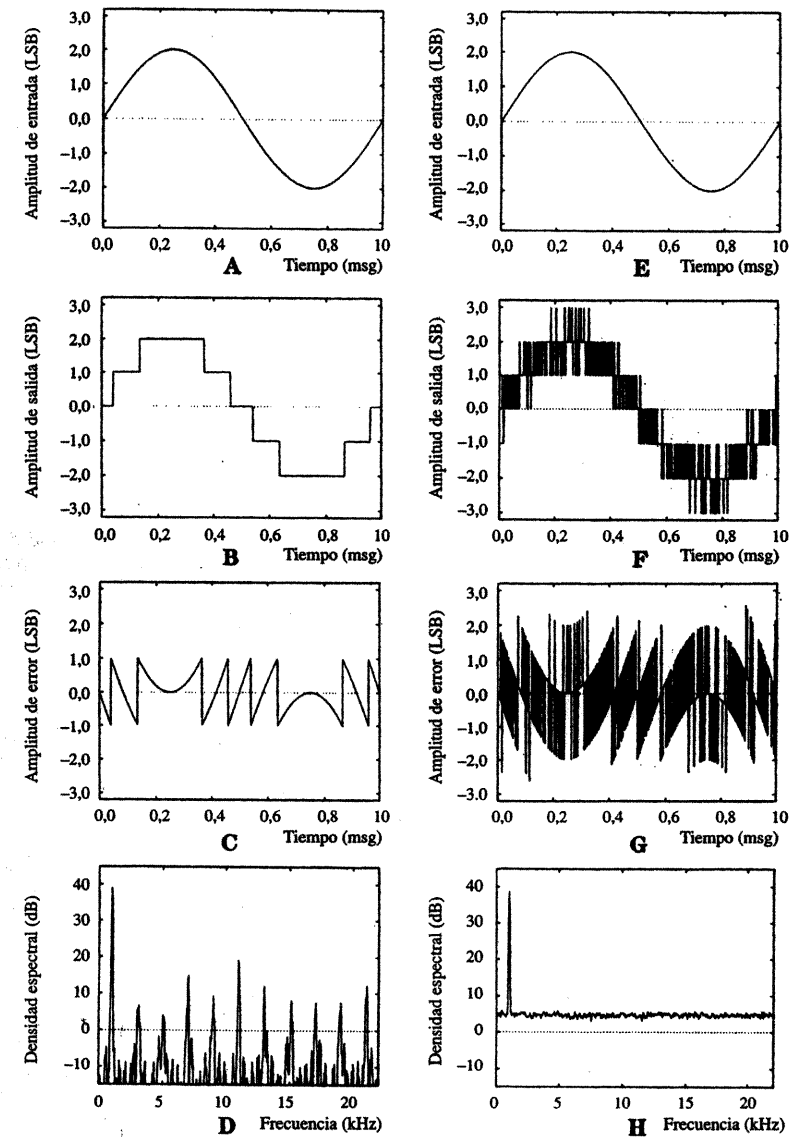


Figura 2.10. Simulación por ordenador de la cuantificación de una sinusoide de bajo nivel de frecuencia 1 kHz, sin *dither* y con *dither*. A. Señal de entrada. B. Señal de salida (sin *dither*). C. Señal de error total (sin *dither*). D. Densidad espectral de potencia de la señal de salida (sin *dither*). E. Señal de entrada. F. Señal de salida (*dither* con pdf triangular). G. Señal de error total (*dither* con pdf triangular). H. Densidad espectral de potencia de la señal de salida (*dither* con pdf triangular) (Lipshitz, Wannamaker y Vanderkooy).

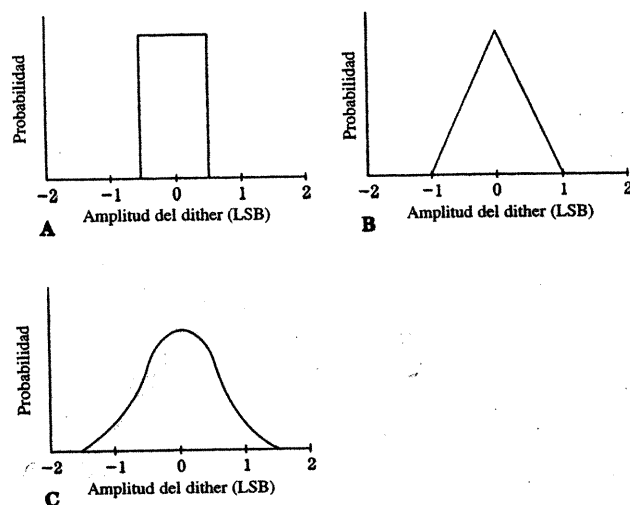


Figura 2.11. Para describir las señales de *dither* se utilizan las funciones densidad de probabilidad. Las más habituales en los sistemas de digitalización de audio son: pdf rectangular, pdf triangular y pdf gaussiana. A. *Dither* con pdf rectangular. B. *Dither* con pdf triangular. C. *Dither* con pdf gaussiana.

aplicar como *dither* a la señal de audio, antes de la conversión A/D. El *dither* gaussiano se genera fácilmente en circuitos analógicos, por ejemplo utilizando un diodo como fuente de ruido. El ruido de *dither* debe alternar entre valores positivos y negativos en los instantes de muestreo; el ancho de banda debe ser al menos la mitad de la frecuencia de muestreo. Un *dither* gaussiano de sólo $1/2$ LSB RMS puede dar linealidad a la curva de transferencia del cuantificador; sin embargo, también añade un pequeño ruido modulado.

La potencia del ruido de cuantificación sin *dither* es $Q^2/12$ (o $Q/(12)^{1/2}$ RMS). La contribución del *dither* gaussiano es $Q^2/4$, por lo que la suma total del ruido de cuantificación con *dither* es de $Q^2/3$ (o $Q/(3)^{1/2}$ RMS), donde recuerde que Q era el intervalo de cuantificación o el LSB. Este incremento en la potencia total del ruido es significativo.

En el *dither* rectangular, la tensión del ruido se distribuye uniformemente en un intervalo. El ruido toma valores en el intervalo $\pm 1/2$ LSB (es decir, el ruido tiene una función densidad de probabilidad constante en un intervalo de un LSB) y produce una linealidad completa en la curva de transferencia del cuantificador, eliminando además la distorsión causada en la cuantificación. Con *dither* uniforme, el ruido de cuantificación tiende a depender (estadísticamente) de la señal de audio. El ruido de modulación que aparece puede ser insoportable en bajas frecuencias y en ciertas señales. Un *dither* rectangular de $\pm Q/2$ añade en el ruido de cuantificación una potencia de ruido de $Q^2/12$; por tanto, la suma total de la potencia del ruido de cuantificación es $Q^2/6$ (o $Q/(6)^{1/2}$ RMS).

El *dither* óptimo es el *dither* con una función densidad de probabilidad triangular de amplitud 2 LSB, formado al sumar (convolución de las funciones de densidad) dos *dither* con funciones pdf rectangulares de amplitud 1 LSB. El *dither* triangular elimina tanto la distorsión de la cuantificación como la modulación del ruido de fondo; por el contrario, el nivel de ruido que se añade a la señal de audio es algo superior al *dither* rectangular. El *dither* triangular añade al

ruido de cuantificación una potencia de ruido de $Q^2/6$, lo que hace que la potencia total de ruido sea de $Q^2/4$ (o $Q/2$ RMS).

De esta forma, utilizando señales de *dither* con las amplitudes óptimas reseñadas anteriormente, el *dither* rectangular produce un incremento de 3 dB, el *dither* triangular de 4.77 dB y el *dither* gaussiano de 6 dB. Normalmente, el ruido rectangular se emplea más en señales de prueba y el *dither* triangular, aun a pesar de tener un mayor nivel de ruido, en aplicaciones reales con señales de música. El empleo del ruido gaussiano está penalizado con un gran nivel de ruido. En recuantificaciones digitales (pasar de «m» a «n» bits, $m > n$), previas a la conversión D/A, se utilizan habitualmente señales de *dither* triangular y rectangular por su simplicidad de implementación. A la hora de evaluar la distorsión de un equipo digital de audio, hay que utilizar siempre señales de prueba con *dither*; de lo contrario, los valores medidos podrían ser de la propia señal de prueba y no del equipo evaluado.

La amplitud del *dither* también es una cuestión a considerar. En la Figura 2.12 se muestra la linealidad de la curva de transferencia de cuantificación en el primer intervalo, en función de la amplitud del *dither* (anchura de la función pdf) y en dos casos distintos: *dither* gaussiano y *dither* rectangular. En ambos casos, a medida que se aumenta la amplitud del *dither* se reducen las anomalías del proceso de cuantificación. Como puede apreciarse en la figura, un *dither* gaussiano de amplitud $1/2$ LSB RMS produce la linealidad buscada. Con *dither* rectangular es necesario un nivel de 1 LSB. En cualquier caso, si se utiliza un *dither* demasiado alto se empeora la relación señal a ruido del sistema.

El aumento de ruido por efecto del *dither* es normalmente inapreciable cuando existe una gran relación señal a ruido, y puede minimizarse aún más si se utiliza un *dither* filtrado (filtrado paso alto). Esto puede conseguirse fácilmente con algoritmos digitales. Por ejemplo, un *dither* triangular puede generarse de manera que predominen las altas frecuencias en su espectro. De esta forma, ya que el oído es menos sensible en este margen de frecuencias, este *dither* es menos audible que otro que tuviera un espectro uniforme (plano), eliminando además la distorsión y el ruido de modulación, como con cualquier *dither* triangular. Estas técnicas se utilizan para reducir la percepción del ruido de cuantificación, por ejemplo, cuando se convierte una señal digital de 20 bits en otra de 16 bits. De una forma más general, el ruido de cuantificación se puede moldear espectralmente con técnicas de procesamiento digital con objeto de minimizar la

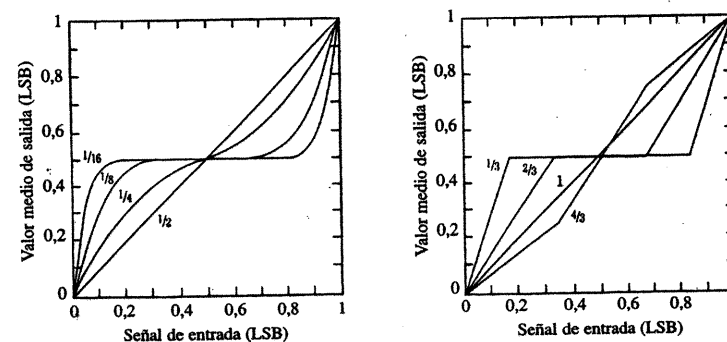


Figura 2.12. Características de la función de transferencia entrada/salida mostrando los efectos del *dither* al variar la amplitud del mismo. A. Un *dither* con una pdf gaussiana y un nivel de $1/2$ LSB rms linealiza la señal de entrada. B. Con un *dither* de pdf rectangular y un nivel de 1 LSB rms es posible linealizar la señal de entrada (Vanderkooy y Lipshitz).

percepción psicoacústica del mismo. En el Capítulo 18 se ven con detalle estas aplicaciones de moldeado espectral (*noise shaping*).

La amplitud del *dither* puede disminuirse aún más si a la señal de audio se le suma una componente sinusoidal de nivel ínfimo (1 o $\frac{1}{2}$ LSB). Esta componente debe ser de una frecuencia superior a 20 kHz e inferior a la frecuencia Nyquist, consiguiendo de esta forma que sea totalmente inaudible. El espectro del error de cuantificación se modifica con objeto de minimizar su percepción auditiva y sobre todo no añade tanto ruido a la señal de audio como lo haría un *dither* clásico. Por ejemplo, un *dither* con una pdf triangular produciría un incremento de sólo 2 dB, en contraposición a los 4.77 dB que produce un *dither* triangular clásico. Sin embargo, esta estrategia puede producir cierta distorsión de intermodulación en las señales de audio. Las señales de *dither* de banda ancha no tienen estos problemas.

El problema de las señales de *dither* aditivas, como las vistas hasta ahora, es que empeoran la relación señal a ruido (S/N) del sistema. Barry Blesser propuso un *dither* sustractivo con objeto de mantener la relación S/N inicial de la señal digital de audio. Esta técnica se muestra en la Figura 2.13. El *dither* uniforme o rectangular es una señal aleatoria que puede generarse a partir de una secuencia pseudoaleatoria de datos digitales. Este proceso puede implementarse con una serie de registros de desplazamiento, en una estructura realimentada de puertas lógicas OR-EXCLUSIVA. La secuencia de datos se convierte en un ruido analógico mediante un convertidor D/A y se mezcla con la señal de audio para obtener una digitalización con *dither*. Una vez digitalizada esta señal, con el convertidor A/D se puede restar de la misma el *dither* (ruido generado), preservando de esta forma el rango dinámico inicial de la señal de audio. La ventaja de este proceso es que las posibles imperfecciones del convertidor A/D adquieren un carácter aleatorio. Existe una forma alternativa de generar un *dither* sustractivo con dos generadores aleatorios sincronizados; uno de ellos se utiliza en la conversión A/D y el otro en la conversión D/A, para sustraer esa misma cantidad de *dither* (*dither* uniforme). Como alternativa, en un sistema con *auto-dither*, la señal de audio se puede aleatorizar con el fin de crear una conversión A/D con *dither* y después, en la conversión D/A, generar la misma aleatoriedad y restársela a la misma señal de audio para recuperar su rango dinámico original.

También es necesario aplicar *dither*, en este caso digital, cuando se manipulan los datos de audio con objeto de minimizar los errores de redondeo. Por ejemplo, con los errores de truncamiento que aparecen en las operaciones de multiplicación, los cuales pueden llegar a ser inadmisibles, como se verá en el Capítulo 17.

Para terminar, a modo de anécdota, decir que fue en la segunda guerra mundial donde se empezó a utilizar el *dither*, como señala Jim MacArthur. Los bombarderos utilizaban sistemas mecánicos en la navegación y en el cálculo de trayectorias de las bombas. Curiosamente, estos sistemas (cajas con centenares de engranajes y ruedas dentadas) funcionaban mejor cuando estaban en el aire, volando, que cuando se hacían simulaciones en tierra. Los ingenieros se die-

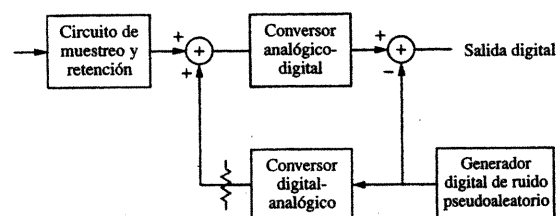


Figura 2.13. Ejemplo de *dither* sustractivo digital utilizando un generador numérico pseudoaleatorio.

ron cuenta que las vibraciones del avión reducían los errores de las partes que estaban sujetas entre sí. En vez de moverse con pequeñas sacudidas, se movían de una manera más continua. Las pequeñas vibraciones que producían los motores se implementaron posteriormente en los computadores, y estas vibraciones fueron conocidas con el nombre de *dither*, palabra que se deriva del inglés antiguo «*didderen*», que significa temblar. Hoy en día, cuando damos un golpecito a un medidor mecánico para aumentar la precisión de la lectura, lo que realmente estamos haciendo es aplicar *dither*. Los diccionarios modernos definen la palabra *dither* como «estado altamente nervioso, confuso o agitado». En pequeñas cantidades, el *dither* proporciona al sistema digital cierto carácter analógico en el mejor sentido de la palabra.

CONCLUSIÓN

El muestreo y la cuantificación son los dos criterios fundamentales en la digitalización. La frecuencia de muestreo determina el ancho de banda de la señal y por tanto la respuesta en frecuencia. Aunque complejo, el muestreo está basado en principios perfectamente definidos; la piedra angular del muestreo produce resultados totalmente previsibles. Si no se cumple el teorema de muestreo puede aparecer un fenómeno anómalo conocido por el nombre de aliasing. La cuantificación determina el rango dinámico del sistema, medido como relación señal/error. El muestreo es un proceso sin pérdidas, mientras que la cuantificación sólo es una aproximación. La cuantificación puede producir ciertas anomalías en el sistema. No obstante, el *dither* puede eliminar la distorsión producida por la cuantificación y mantener la fidelidad de la señal digitalizada de audio. Normalmente, una frecuencia de muestreo de 44,1 o 48 kHz, y datos de audio a 16 o 20 bits (con *dither*), producen una fidelidad comparable, e incluso mejor, a cualquier sistema analógico, con la ventaja añadida de la longevidad y la fidelidad de las copias digitales de datos. Frecuencias de muestreo más altas y datos con mayor número de bits pueden dar una calidad extrema a la señal de audio.

Un último apunte sobre las frecuencias de muestreo. Antes de terminar nuestra discusión sobre el muestreo, considere la siguiente hipótesis sobre la naturaleza del tiempo. El tiempo parece ser continuo. Sin embargo, algunos físicos han sugerido que, al igual que la energía y la materia, el tiempo puede transmitirse en paquetes discretos. De igual forma que este libro está formado por un conjunto finito de átomos y pueden convertirse en una cantidad finita de energía, el tiempo que tarda usted en leer este libro puede estar formado por partículas. Específicamente, el período indivisible del tiempo podría ser 10^{-42} segundos (un punto decimal seguido de 41 ceros y un uno). Según esta teoría, no podría haber un intervalo inferior, ya que la energía necesaria para realizar una nueva división sería tan grande que se formaría un agujero negro y el suceso sería engullido por el propio agujero. Si alguno de ustedes está experimentando en el sótano de su casa con tales frecuencias de muestreo, por favor, tenga cuidado*.

* Nota del traductor. Este párrafo, aunque parezca mentira, lo ha escrito el autor. El traductor no suscribe lo dicho en el mismo. Piensa, por el contrario, que el tiempo sencillamente no existe.

Capítulo 3

GRABACIÓN DIGITAL DE AUDIO

En el diseño electrónico de grabadores digitales de audio están presentes los dos principios fundamentales de la digitalización: el muestreo y la cuantificación. La señal analógica se muestrea, se transforma en un formato numérico y se procesa antes de ser grabada, transmitida o procesada digitalmente. La cadena de codificación está formada por los siguientes circuitos electrónicos: sistema dither, filtro paso bajo antialiasing, circuito de muestreo y retención, convertidor analógico-digital y modulador de canal. Aunque existen otros métodos de codificación, la codificación PCM (*linear pulse-code modulation*) es la más ilustrativa y la más utilizada en los sistemas digitales de audio. Este capítulo y el siguiente se centran en la arquitectura de los sistemas electrónicos PCM. Estos sistemas están presentes en los grabadores digitales y en los procesadores que trabajan en tiempo real (unidades multiefectos), realizando el tratamiento necesario (pre y post) para generar, a partir de la señal analógica de entrada, los datos digitales de audio (pre) y para reconstruir (a partir de los datos digitales) la señal analógica de salida (post).

MODULACIÓN POR CODIFICACIÓN DE PULSOS

En teoría, se podrían utilizar un sinnúmero de técnicas numéricas para codificar digitalmente las señales de audio. Todas ellas son fundamentalmente idénticas a la hora de representar las señales de audio como datos digitales, pero en la práctica difieren en la eficiencia y en los resultados obtenidos. La modulación no es nada más que la forma en la que se codifica la información que va a ser grabada o transmitida. Técnicas tales como la modulación de amplitud (AM) y la modulación de frecuencia (FM) se han utilizado ampliamente en radiodifusión para modular frecuencias portadoras con una información analógica de audio. Son modulaciones en las que se varía de forma continua un parámetro de la onda portadora.

Para transmitir información muestreada se pueden utilizar varios tipos de codificación de pulsos. Por ejemplo, la anchura del pulso, o la posición del pulso, puede servir para indicar la amplitud de la señal en el instante de muestreo; en el primer caso tendremos una modulación PWM (*pulse width modulation*) y en el segundo una modulación PPM (*pulse position modulation*). En ambos casos, la amplitud de la señal se codifica y se transmite en pulsos de amplitud constante. La amplitud de la señal también se puede codificar directamente en la amplitud de los pulsos; la modulación PAM (*pulse amplitude modulation*) es un ejemplo de este tipo de codifi-

cación. La amplitud de cada uno de los pulsos refleja la amplitud de la señal en los instantes de muestreo. En la Figura 3.1 (A-D) se muestran las modulaciones PWM, PPM y PAM. En otras modulaciones, las amplitudes de las muestras de la señal se codifican mediante diversos métodos numéricos. Por ejemplo, en la modulación por número de pulsos (PNM, *pulse-number modulation*) el modulador genera una serie de pulsos; el número de pulsos representa la amplitud de la señal en el instante de muestreo; esta modulación se muestra en la Figura 3.1 (E-F). Sin embargo, en esta modulación se necesita un gran número de pulsos para tener una gran resolución en amplitud. Aunque estas modulaciones (PWM, PPM, PAM y PNM) se pueden utilizar en un contexto de conversión, no son óptimas para realizar una grabación o una transmisión digital debido a sus pobres limitaciones en ancho de banda y al error de codificación.

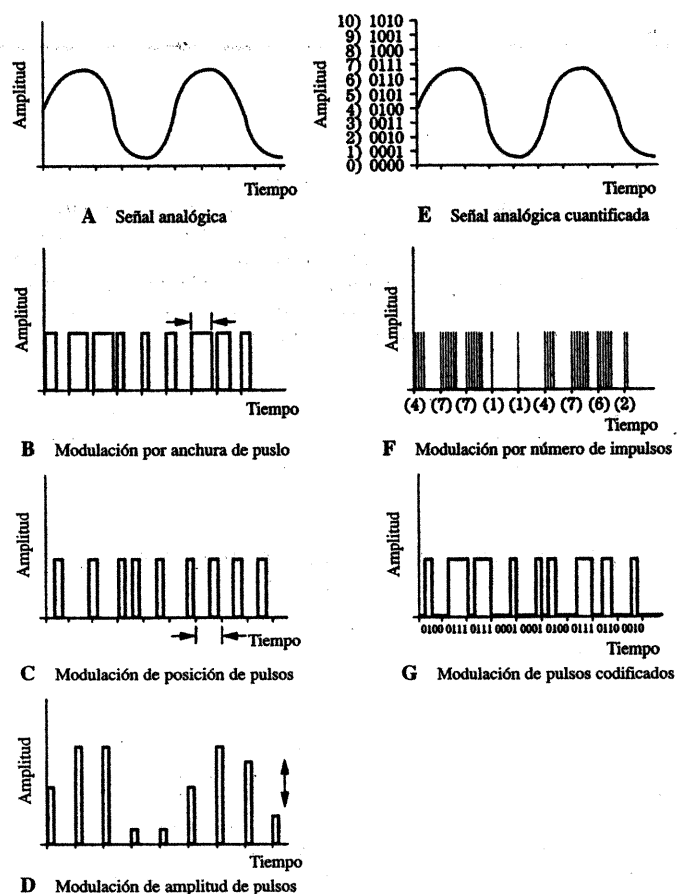


Figura 3.1. Las modulaciones PWM, PPM y PAM son ejemplos de modulaciones en las que se modifica alguno de los parámetros de los pulsos. Las modulaciones PNM y PCM son ejemplos de modulaciones numéricas de los parámetros de los pulsos.

La modulación más utilizada es la modulación por codificación de pulsos (PCM). La modulación PCM fue creada en 1937 por Alec Reeves, ingeniero de los laboratorios de la compañía francesa *International Telephone and Telegraph* (ITT) (Reeves también desarrolló la modulación PWM). En la modulación PCM, la señal se muestrea, se cuantifica y se codifica. En esta modulación, la amplitud de la señal en los instantes de muestreo se codifican con una representación binaria, que a su vez determina la codificación final de los pulsos. En el decodificador, la codificación de cada uno de los pulsos se utiliza para reconstruir la señal analógica. Las palabras binarias que representan las amplitudes de las muestras se codifican directamente en señales PCM, tal como se muestra en la Figura 3.1G.

En las modulaciones PWM, PPM y PAM sólo es necesario un pulso para representar la amplitud de la muestra, a diferencia de la modulación PCM, donde se necesitan varios pulsos por muestra. Como resultado, la modulación PCM requiere un mayor ancho de banda. Sin embargo, las señales PCM son señales muy robustas, ya que para reconstruir una señal sólo hay que decidir entre dos estados: presencia o ausencia de pulso. En resumen, una señal PCM puede ser regenerada sin ningún tipo de pérdidas. Por tanto, la calidad de una transmisión PCM depende de la calidad del muestreo y de la cuantificación, no de la calidad del canal. En suma, dependiendo de la frecuencia de muestreo y de la capacidad del canal, se pueden combinar varias señales PCM y ser transmitidas de forma simultánea mediante una multiplexación en el dominio del tiempo. Esta característica ha sido la que realmente ha impulsado la utilización de la modulación PCM, por ejemplo, para transmitir señales estéreo. Aunque existen otros métodos de codificación, y otros nuevos aparecerán en el futuro, la calidad de los mismos siempre se comparará con el referente de calidad que proporciona la modulación PCM. En la mayoría de las ocasiones, se utilizan complejos códigos de canal (a continuación de la modulación PCM) antes de grabar la señal en un soporte físico. Estos códigos de canal también se describen en este capítulo.

El diseño de un sistema lineal PCM (algunas veces denominado LPCM) está ligado al diseño del sistema de digitalización. La señal analógica se filtra y se muestrea, y su amplitud se cuantifica con un convertidor analógico-digital A/D. Los códigos binarios son representados mediante una serie de pulsos modulados que representan a su vez la amplitud de la señal PCM en cada uno de los instantes de muestreo. Cuando se muestrean dos canales, los datos de cada uno de ellos se pueden multiplexar para formar un único flujo de datos. Los datos pueden ser manipulados para proporcionarles sincronización y protección contra errores, así como nuevos datos auxiliares pueden ser añadidos. En la reproducción, los datos son demodulados, decodificados y corregidos (en caso de error), para recuperar la amplitud original PCM en cada uno de los instantes de muestreo, reconstruyéndose finalmente la señal analógica con un convertidor digital-analógico D/A, seguido de un filtro paso bajo.

La sección de codificación en un grabador digital PCM estéreo estándar está formada por amplificadores, generadores de dither, filtros paso bajo, circuitos de muestreo y retención, convertidores A/D, un multiplexador, un procesador digital con circuitos de modulación y un soporte físico, como por ejemplo una cinta digital o un disco óptico. El diagrama de bloques de una sección de codificación digital se muestra en la Figura 3.2. Este diseño electrónico es una implementación del teorema de muestreo. En la práctica, pueden añadirse otras técnicas en el diagrama de bloques, como el sobremuestreo.

Un sistema de digitalización de audio no es más que un transductor que procesa una señal de audio para ser almacenada o transmitida, y que la vuelve a procesar de nuevo para reproducirla. Aunque pueda parecer simple, el diseño electrónico es muy complejo y preciso; la calidad de la señal de audio depende completamente del diseño del sistema. Cada subsistema debe considerarse de forma precisa.

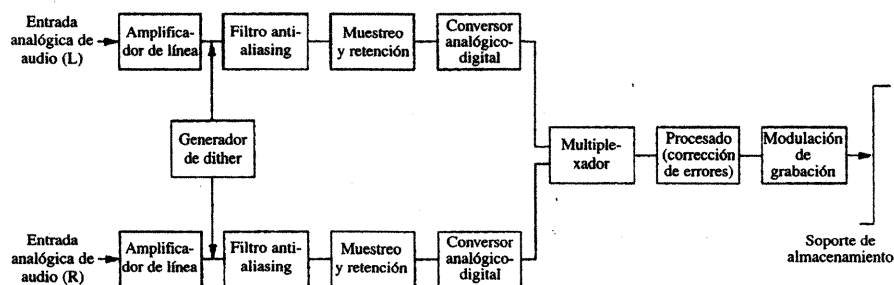


Figura 3.2. Sección de grabación PCM mostrando los principales elementos.

GENERADOR DE DITHER

La señal de dither es una señal de ruido que se añade a la señal de audio para eliminar las anomalías que presenta la cuantificación, como se apuntó en el Capítulo 2. El dither provoca que la señal de audio bascule entre niveles de cuantificación adyacentes. Decorrela el error de cuantificación de la señal de audio, elimina la distorsión y es capaz de codificar amplitudes inferiores a un intervalo de cuantificación. Sin embargo, aunque reduce la distorsión, el dither añade cierta cantidad de ruido a la señal de audio. Perceptualmente, el dither es beneficioso, ya que el oído tolera más fácilmente el ruido que la distorsión.

El dither analógico, aplicado antes del convertidor A/D, provoca en el propio convertidor nuevas transiciones de nivel que preservan las señales de bajo nivel en un ciclo de trabajo, o en una modulación por anchura de pulsos. Esto da cierta linealidad al proceso de cuantificación. La distorsión, por ejemplo, se transforma en un ruido de banda ancha. Se pueden aplicar varios tipos de dither, con distintas funciones de densidad de probabilidad (pdf): gaussiana, rectangular y triangular, a discreción del diseñador; en algunos sistemas el usuario tiene la posibilidad de elegir el tipo de dither. La amplitud del dither también es aspecto crítico. En algunos casos, la señal de entrada pudiera tener un alto nivel de ruido residual. Por ejemplo, una cinta analógica puede tener un nivel residual de ruido suficiente para ser considerado como dither en el cuantificador. No obstante, un sistema digital debe proporcionar un rango dinámico suficiente para albergar toda la información analógica, incluyendo la señal residual de ruido, sin que la cuantificación produzca distorsión. La longitud binaria de los datos del cuantificador debe ser suficiente para el programa de audio, y su bit menos significativo (LSB) debe estar «modulado» por el dither. Siempre que se reduzca la longitud binaria de los datos, por ejemplo al pasar una mezcla máster de 20 bits a una mezcla para CD de 16 bits, debe aplicarse un nuevo dither, incluyendo además un moldeado espectral del mismo (*noise shaping*). El dither se discutió con detalle en el Capítulo 2; el moldeado espectral (psicoacústico) del dither está descrito en el Capítulo 18.

FILTRO PASO BAJO

En un grabador digital clásico (sin sobremuestreo) la señal de entrada se filtra con un filtro paso bajo con una gran pendiente de caída en la banda de transición; esto limita el ancho de banda de la señal a un valor igual o inferior a la frecuencia Nyquist. Por ejemplo, en un grabador digital

con una frecuencia de muestreo de 48 kHz, la frecuencia de corte debe estar en torno a los 22 kHz para permitir una atenuación máxima en la mitad de la frecuencia de muestreo. Al filtro de entrada se le conoce habitualmente por el nombre de filtro antialiasing.

El filtro paso bajo de entrada debe atenuar todas las señales de frecuencias superiores a la mitad de la frecuencia de muestreo, sin afectar a las señales en la banda de audio. Un filtro paso bajo ideal debe tener una banda de paso plana, una banda de transición instantánea y una banda de atenuación infinita, tal como se muestra en la Figura 3.3A. Además de estas características de respuesta en frecuencia, un filtro ideal no debe alterar la fase de la señal. Aunque en la práctica el filtro se puede aproximar a uno ideal, su implementación lleva aparejado una serie de problemas. Una banda de transición extremadamente estrecha impide cumplir otras especificaciones, como por ejemplo una respuesta plana en la banda de paso y una baja distorsión de fase. Por tanto, en el diseño del filtro se permite un determinado margen de frecuencias en la banda de transición y así reducir los problemas de distorsión de fase. Sin embargo, un filtro paso bajo de un orden reducido, con una determinada banda de transición, debe tener una frecuencia de corte mayor para impedir que se introduzcan frecuencias indeseadas por aliasing. Para evitar el aliasing, la frecuencia de muestreo debe ser lo suficientemente alta para tener una gran atenuación en la mitad de la frecuencia de muestreo (frecuencia Nyquist). Una frecuencia de muestreo mayor, tal vez tres veces el valor de la frecuencia de corte del filtro ideal, es necesaria para cumplir las especificaciones de ancho de banda de la señal de audio. La frecuencia de muestreo mínima, dejando pasar todas las componentes de la señal por debajo de la frecuencia Nyquist, sólo puede utilizarse con un filtro paso bajo ideal. Es irónico que este filtro presente problemas de implementación, ya que su diseño es analógico. Después del convertidor A/D, este mismo filtro puede implementarse digitalmente de una manera extremadamente sencilla. Más adelante en este capítulo, y en detalle en el Capítulo 18, se tratará la técnica del sobremuestreo en la conversión que permitirá solventar todos estos problemas, utilizando además un filtro paso bajo analógico muy simple.

En los grabadores digitales sin sobremuestreo con una frecuencia de muestreo de 48 kHz, los filtros de entrada se diseñan con una respuesta plana desde la componente continua dc (*direct current*) hasta los 22 kHz. Este margen proporciona una banda de guarda de 2 kHz que permite una suficiente atenuación. La banda de paso debe tener una respuesta plana; en la práctica existen algunas irregularidades (rizado en torno a los 0.1 dB de variación máxima). La banda atenuada se diseña con una atenuación que sea igual o mayor que el margen dinámico del sistema, determinado por el número de bits de los datos. Por ejemplo, en un sistema de 16 bits se requiere una atenuación mínima de 95 dB. Una atenuación de 80 dB produce, en el peor

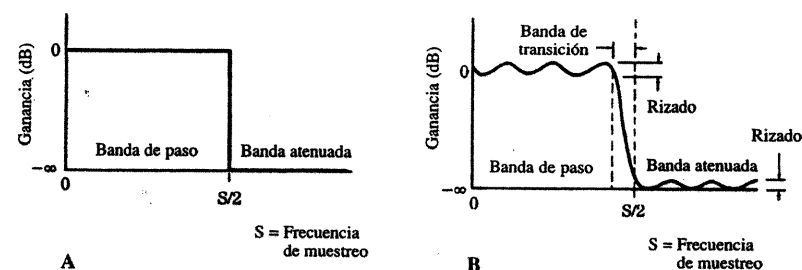


Figura 3.3. Características de un filtro paso bajo. A. Un filtro ideal paso bajo tiene una respuesta plana en la banda de paso y una frecuencia de corte instantánea. B. En la práctica, los filtros presentan un rizado en la banda de paso y en la banda atenuada, así como una banda de transición entre ambas.

de los casos, una distorsión por aliasing del 0,01 por 100. En la Figura 3.3B se muestran las características de un filtro paso bajo.

Otras características importantes del filtro son: la respuesta temporal (*overshoot*), el rizado y la linealidad de fase. Pendientes elevadas en la banda de transición producen resonancias alrededor de la frecuencia de corte y el rizado puede causar alguna «coloración» en la respuesta en frecuencia. Cuanto menor sea la banda de transición mayor será el rizado. No obstante, algunos tipos de filtro tienen un rizado mínimo. La respuesta en fase también es un parámetro a considerar. Los sistemas de grabación magnética analógica, los micrófonos y los altavoces siempre introducen distorsión de fase; los filtros paso bajo analógicos presentan un retardo en función de la frecuencia, denominado retardo de grupo, en las cercanías de la frecuencia de corte, produciendo una distorsión de fase. Este problema puede corregirse con un circuito analógico, antes o después del filtro, que compense el retardo y consiga una linealidad de fase en el sistema, es decir, un retardo constante con la frecuencia (inaudible). No se sabe a ciencia cierta cuál es el umbral de audición a partir del cual se detectan los efectos del rizado y del retardo de grupo de los filtros.

Los filtros analógicos se clasifican en función del polinomio matemático que determina sus características. Existen varios tipos; los más utilizados son los filtros de *Bessel*, *Butterworth* y *Chebyshev*. En cada caso, para incrementar el orden del filtro y obtener una banda de transición más estrecha, hay que implementar estructuras básicas en cascada, propias de cada tipo de filtro. A medida que se aumente el orden del filtro, éste se aproximará más a la respuesta de un filtro ideal, con una banda de transición mínima. En la Figura 3.4 se presenta un filtro *Chebyshev*; la pendiente de la banda de transición se acentúa a medida que se incrementa el orden del filtro, es decir, añadiendo estructuras básicas en cascada, tal como aparece en la figura. Sin embargo, la distorsión de fase también aumenta proporcionalmente con el orden. El filtro paso bajo más sencillo está constituido por una de las secciones RC (resistencia-condensador) dispuestas en cascada según la figura; cada sección produce un incremento en la pendiente de caída (banda de transición) de 6 dB/octava. Aunque el filtro no presente ni *overshoot* ni rizado, siempre existen anomalías en la respuesta en frecuencia de la banda de paso.

En la banda de paso, en las proximidades de la frecuencia de corte, se pueden colocar pequeñas resonancias con objeto de suavizar la pendiente en la banda de transición. Estas resonancias no influyen en el valor de la pendiente, ya que ésta sólo depende del orden del filtro; éste es el caso de los filtros de *Butterworth*. Estos filtros requieren un elevado orden para conseguir una banda de transición mínima y una gran atenuación en la banda atenuada. Por ejemplo, para obtener una banda de transición de 0,4 octavas y una atenuación de 80 dB es necesario un filtro de orden 33.

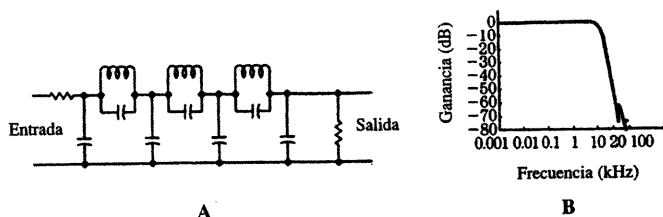


Figura 3.4. Ejemplo de un filtro paso bajo tipo Chebyshev y su respuesta en frecuencia. A. Esquema de un filtro paso bajo pasivo. B. Respuesta en frecuencia de un filtro paso bajo en la que se muestra la pendiente de caída en la banda de transición.

Se puede diseñar un filtro con una menor banda de transición a expensas de modificar la respuesta en frecuencia de la banda de paso, introduciendo resonancias de magnitud mayor a las introducidas en un diseño *Butterworth*. El tipo de filtro que realiza esta aproximación es el filtro *Chebyshev*. Con un filtro de noveno orden se puede conseguir un rizado máximo de 0,1 dB hasta los 20 kHz, y una atenuación de 70 dB a partir de los 25 kHz.

La mayoría de los filtros producen una atenuación mayor que la requerida, a partir de la mitad de la frecuencia de muestreo. Esto sucede con curvas de atenuación monótonas. La banda de transición puede reducirse, colocando anti-resonancias en la banda atenuada. Además, los elementos reactivos del filtro (bobinas y condensadores) pueden producir a un mismo tiempo resonancias y anti-resonancias. Esto reduce la complejidad electrónica del filtro al utilizarse un menor número de componentes. El resultado es un filtro elíptico, también denominado filtro *Cauer*. Un filtro elíptico es el que menor banda de transición produce dado un determinado orden. Por ejemplo, un filtro elíptico de séptimo orden produce un rizado de 0,25 dB en la banda de paso, una banda de transición de 0,4 octavas y una atenuación de 80 dB. En la práctica, estos requisitos sólo se cumplen con un diseño de 13 polos.

En general, dado un determinado orden, los filtros paso bajo *Chebyshev* y *Cauer* se aproximan mejor a la respuesta ideal que los filtros *Bessel* o *Butterworth*; sin embargo, los filtros *Chebyshev* pueden producir un rizado acusado en la banda de paso y los filtros elípticos pueden provocar grandes distorsiones de fase. Los filtros según la aproximación de *Bessel* son los que mejor respuesta en fase presentan; prácticamente el retardo en función de la frecuencia es constante. Por el contrario, es necesario un orden elevado para producir una gran atenuación. Los filtros de *Butterworth* son monótonos en la banda de paso, aunque pueden introducir pequeñas respuestas transitorias. Generalmente no existe un filtro ideal, siempre hay un compromiso entre la atenuación y la respuesta en tiempo. En la práctica, debido a los problemas inherentes que presentan los filtros analógicos de elevado orden, los diseñadores han optado por técnicas de conversión sigma-delta, como se verá en el Capítulo 18. En cualquier caso, siempre es necesario colocar un filtro paso bajo que impida el aliasing de frecuencias superiores a la frecuencia Nyquist (mitad de la frecuencia de muestreo).

CIRCUITO DE MUESTREO Y RETENCIÓN

Como su propio nombre indica, el circuito de muestreo y retención (S/H —*sample hold*—) realiza dos simples, pero críticas, operaciones. Muestra la señal analógica de una forma periódica, poniendo en práctica el teorema de muestreo. También se encarga de mantener constante la amplitud de la muestra a la entrada del convertidor A/D mientras ésta entrega a su salida el dato binario. Esto es importante, ya que si no se mantiene constante el valor de la muestra durante el período de muestreo, el convertidor A/D podría entregar un dato erróneo. En la Figura 3.5 se muestra la entrada y la salida de un circuito S/H. La señal de salida es una señal intermedia, una señal modulada en amplitud (PAM) que representa la señal analógica de entrada; pero no son los datos binarios. El circuito es relativamente fácil de diseñar; no obstante, su implementación debe ser tal que realice sus dos tareas de forma precisa. Las muestras deben capturarse exactamente en los instantes de muestreo y la amplitud capturada debe mantenerse constante dentro de una determinada tolerancia.

Como ya hemos visto, la información del tiempo y de la amplitud caracteriza completamente a una onda acústica. El circuito de muestreo y retención es el encargado de capturar esta información de la señal paso bajo filtrada. Las muestras se toman con una determinada frecuencia de muestreo y se reproducen con la misma frecuencia de muestreo. El circuito S/H es gobernado mediante pulsos periódicos (frecuencia de muestreo) producidos por un reloj, un circuito oscilador.

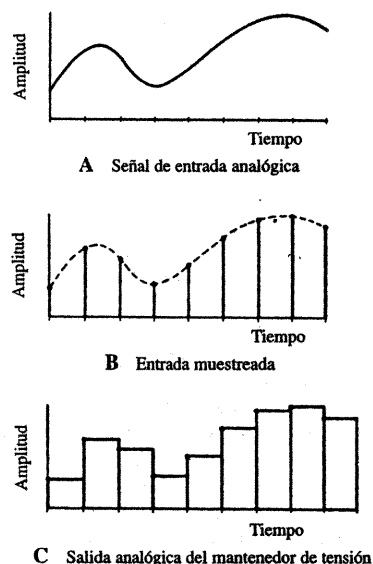


Figura 3.5. El circuito de muestreo y retención captura el valor analógico de tensión y la mantiene constante mientras se produce la conversión.

Conceptualmente, el circuito S/H está formado por un condensador y un conmutador. El circuito va siguiendo la señal analógica hasta que se produce el orden de muestreo, momento en el cual se abre el condensador; el condensador mantiene el valor de tensión durante el proceso de conversión A/D. En la Figura 3.6 se muestra un circuito de S/H ideal; el conmutador se cierra cada vez que se recibe el orden de muestreo. El circuito S/H debe tener un tiempo de adquisición casi nulo; de lo contrario, el convertidor A/D dará un valor que será un promedio de la amplitud de la señal en el instante de muestreo, en vez del valor en el instante exacto de muestreo. Además, las variaciones en la frecuencia de los pulsos que gobiernan el circuito S/H pueden producir un error en la conversión; el oscilador debe ser extremadamente preciso y el circuito S/H debe estar perfectamente diseñado.

El jitter es el parámetro que mide las fluctuaciones temporales de una señal, en este caso, de la señal de muestreo, tal como se refleja en la Figura 3.7. El jitter añade ruido y distorsión a la señal muestreada, por lo que la conmutación de la señal de reloj en el circuito S/H debe tener una tolerancia máxima. El jitter es particularmente significativo en el caso de señales de nivel

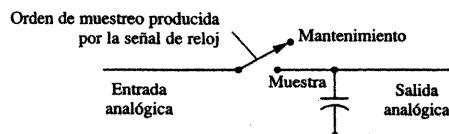


Figura 3.6. En un circuito ideal de muestreo y retención existe un conmutador y un elemento de almacenamiento.

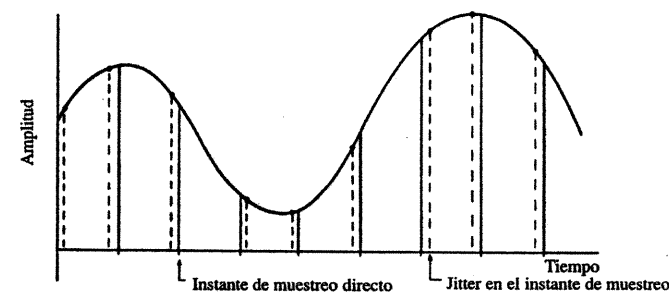


Figura 3.7. El jitter es una variación temporal del instante de muestreo que produce ruido y distorsión en la señal analógica de salida.

alto y en altas frecuencias. En una conversión A/D se requiere una alta precisión en la señal de muestreo. Por ejemplo, en una conversión A/D de 16 bits de una señal sinusoidal de 20 kHz a fondo de escala, el jitter de la señal de reloj del circuito S/H debe ser inferior a los 200 ps (ps = pico-segundo, 10^{-12} sg). En el caso de 18 bits debe ser como máximo de 100 ps. Sólo con estos valores de jitter se puede asegurar que la distorsión introducida estará por debajo del nivel del error de cuantificación. Evidentemente, la señal de reloj que controla el circuito S/H debe ser generada por un oscilador de cristal de cuarzo de alta precisión. El jitter se discutirá en detalle en el Capítulo 4.

El tiempo de adquisición es el intervalo que transcurre desde que se recibe la orden de muestreo hasta que se entrega el valor digital de la muestra. Este retardo produce una muestra de un valor diferente al que debiera tener en el propio instante de muestreo. Un retardo constante no afecta a la señal muestreada, puede ser corregido; sin embargo, este retardo no es constante sino que varía en función de la amplitud de la señal. La mejor solución ante este problema es prevenirlo; por tanto, el intervalo de adquisición debe ser mínimo. La otra tarea del circuito S/H es mantener constante la tensión de la muestra mientras se produce la conversión analógica-digital. Este voltaje debe permanecer constante porque cualquier variación superior al intervalo de cuantificación puede producir un error en el dato de salida del convertidor A/D. En la práctica, el valor de la tensión tiende a caer debido a la corriente de fugas del condensador. El diseño del circuito y la selección de los componentes se realizará para impedir que la caída de tensión sea inferior a la mitad del intervalo de cuantificación durante 20 μ s (μ s = microsegundo, 10^{-6} sg), es decir, durante un período de muestreo. Por ejemplo, un convertidor de 16 bits con márgenes de tensión ± 10 voltios debe impedir que la tensión caiga 1 mV (milivoltio) durante el proceso de conversión. En la Figura 3.8 se muestra tanto el tiempo de adquisición como la caída de tensión.

Un tiempo de adquisición corto y una baja caída de tensión son dos parámetros de diseño incompatibles en un circuito S/H. Si se necesita un tiempo de adquisición corto se debe utilizar un condensador pequeño para que el tiempo de carga sea mínimo. Sin embargo, para mantener constante la tensión en un intervalo mayor, es decir, con una mínima pérdida de tensión, es necesario utilizar un condensador grande. En la práctica, con condensadores en torno a 1 nF se cumplen ambos requisitos. Además, estos condensadores de alta calidad suelen estar fabricados con dieléctricos de polipropileno o teflón. Son unos materiales que responden rápidamente, mantienen la carga y presentan efectos casi nulos de absorción e histéresis, efectos que provocan variaciones de la misma.

En realidad, un circuito S/H tiene algo más que un simple conmutador y un condensador. Existen también circuitos activos como amplificadores operacionales que separan el circuito

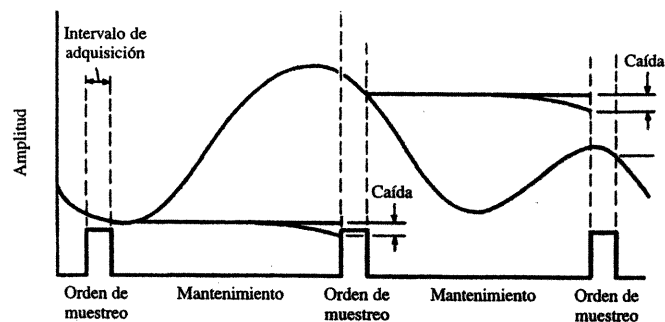


Figura 3.8. Dos condiciones de error en el circuito de muestreo y retención: intervalo de adquisición y caída de nivel de tensión.

S/H de la señal de entrada, dan rapidez a la conmutación y ayudan a impedir que exista una corriente de fugas en el condensador. Sólo unos pocos amplificadores operacionales cumplen estas especificaciones: gran ancho de banda y mínimo tiempo de respuesta. Los llamados operacionales JFET (*junction field-effect transistor*) son los que normalmente funcionan mejor. Generalmente, un circuito S/H está diseñado con un operacional JFET que impide que la fuente de señal cargue al circuito y tenga una rápida respuesta en la conmutación, aislando el condensador y suministrándole una corriente de carga. El conmutador del S/H puede ser otro operacional JFET, que trabaje «limpiamente» con un mínimo jitter. El condensador, por otra parte, debe presentar una baja histéresis. Por último, un operacional JFET de salida será el encargado de garantizar que la carga del condensador sea constante.

CONVERTIDOR ANALÓGICO DIGITAL

El convertidor analógico digital (A/D) es realmente el corazón, en la parte de la codificación, del sistema de digitalización de audio, y es el dispositivo más crítico de todo el sistema electrónico. Cualquier error introducido por el convertidor A/D acompañará a los datos durante el resto de la cadena digital de audio y se reflejará al final en la señal analógica, reconstruida mediante un proceso de conversión digital-analógica. Fundamentalmente, este dispositivo debe examinar el nivel de la señal muestreada, determinar el intervalo de cuantificación más cercano y asignar un código binario (dato de salida); todas estas tareas deben de realizarse en cada período de muestreo (20 μ s para una frecuencia de muestreo de 48 kHz). La precisión que se requiere es considerable: 15 partes por millón en una resolución de 16 bits y una parte por millón en resoluciones de 20 bits. En un diseño tradicional, el código binario se obtiene comparando la tensión de la muestra con un voltaje de referencia variable, insertado en un lazo de realimentación; a este método se le conoce por el nombre de aproximaciones sucesivas. Los convertidores A/D con sobremuestreo se verán más adelante en este capítulo y con detalle en el Capítulo 18.

El convertidor A/D debe realizar una conversión en cada período de muestreo. Además, el código binario debe ser una representación precisa de la tensión de entrada. En sistemas PCM de 16 bits, cada uno de los 65.536 intervalos deben estar uniformemente distribuidos a lo largo de todo el margen de tensión admisible, de manera que incluso el bit menos significativo (LSB) del código binario sea un valor representativo de tensión (intervalo de cuantificación). Por tanto, la velocidad y la precisión son los dos parámetros clave en un convertidor A/D. Un convertidor siempre tendrá un error de $\pm 1/2$ LSB, error inherente en el proceso de cuantificación; por supuesto, en la conversión debe aplicarse una señal de dither.

El tiempo de conversión es el tiempo que necesita el convertidor A/D para producir en su salida un código binario. Éste debe ser inferior al período de muestreo. Algunas veces es difícil lograr la precisión necesaria para realizar una correcta conversión debido al tiempo de inicialización del propio convertidor o a otro tipo de error temporal que se haya podido arrastrar. Por tanto, una conversión podría condicionar a la siguiente. Si la tensión de entrada al convertidor se mueve de un voltaje A a otro B y después de C a B, el código binario de esta última tensión B podría ser erróneo, ya que el convertidor puede que sea incapaz de volver a la situación previa por falta de tiempo (tiempo de establecimiento). Obviamente, estos errores dinámicos se hacen más acusados a medida que se necesitan conversiones más rápidas. En la práctica, los convertidores son lo suficientemente rápidos para las frecuencias de muestreo que se utilizan en audio y presentan una distorsión inapreciable. Realmente, los convertidores A/D procesan simultáneamente dos señales de audio, alternan entre el canal izquierdo y el derecho de una pareja estéreo.

La precisión es otro parámetro importante a considerar. Algunas de las especificaciones del convertidor A/D nos sirven para evaluar su calidad. La linealidad integral mide la «rectitud» de la salida del convertidor. Describe los puntos de transición de tensión, es decir, los puntos de tensión de la señal analógica de entrada en los cuales se produce la asignación de los códigos binarios, y la separación respecto a los puntos ideales, dispuestos en una línea recta. En otras palabras, la linealidad integral determina la desviación entre el intervalo real y el intervalo ideal que produce la transición de un bit en código binario, en todo el margen de tensiones del convertidor. En la Figura 3.9A se muestra la linealidad integral de un convertidor A/D. Siempre se ha de registrar en todo el margen del convertidor. Es el parámetro más importante de un convertidor A/D y no admite corrección alguna. Un convertidor de n bits no puede considerarse como tal a menos que se garantice una linealidad integral inferior a $1/2$ LSB. El convertidor de la Figura 3.9A tiene una linealidad integral de $1/4$ LSB.

El error de linealidad diferencial es la diferencia entre la altura real del escalón de cuantificación y el valor ideal de un LSB. Su valor se halla calculando la distancia entre los puntos de transición de tensión, esto es, la anchura de los escalones (tensión de entrada). La linealidad diferencial se muestra en la Figura 3.9B. Idealmente, todos los escalones de cuantificación deberían tener la misma anchura: 1 LSB. Una linealidad diferencial máxima de $\pm 1/2$ LSB significa que la tensión de entrada puede variar desde $1/2$ LSB hasta $1 1/2$ LSB sin que afecte al código binario de salida. Si se excede esta especificación, por ejemplo a un valor de ± 1 LSB, algunos de los escalones de cuantificación podrían tener una anchura máxima de 2 LSB y otros de 0 LSB (nulos); en otras palabras, algunos códigos podrían desaparecer. La bondad de los convertidores A/D debe asegurarse para que no desaparezcan códigos en un determinado margen de temperaturas. El convertidor de la Figura 3.9B tiene un error de $\pm 3/4$ LSB; algunos niveles sólo tienen una anchura de $1/4$ LSB y otros de $7/4$ LSB. La rapidez del convertidor puede afectar tanto a la linealidad integral como a la linealidad diferencial. Los convertidores A/D siempre se diseñan para que sean monótonos; es decir, ante un aumento del nivel en la señal de entrada, el valor del código binario siempre se incrementará o, en el peor de los casos, no variará. Sin embargo, si la linealidad diferencial es mayor que 1 LSB, el convertidor dejará de ser monótono.

El error de precisión absoluta, según se muestra en la Figura 3.9C, es la diferencia entre el nivel ideal en el que debe ocurrir una transición de código y el nivel con el que realmente sucede. Un buen convertidor debería tener un error inferior a $\pm 1/2$ LSB. Este error se debe: a derivas de tensión, a errores de ganancia en la curva de transferencia y a ruidos indeseados. En el convertidor de la Figura 3.9C, cada intervalo tiene un error de $1/8$ LSB. En la práctica, incluso los convertidores con una gran calidad sufren derivas con las variaciones de temperatura, introduciendo de esta forma una serie de imprecisiones en la conversión.

La anchura del código, denominado algunas veces «quantum», es el margen de tensión de entrada que produce una transición en el valor del código. La anchura ideal es 1 LSB. Los convertidores A/D presentan errores de ganancia y de offset. El rango dinámico de un convertidor que trabaje sólo con valores de tensión positivos está comprendido entre 0 voltios y el fondo de

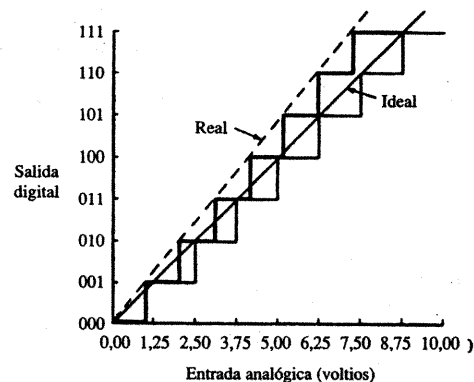
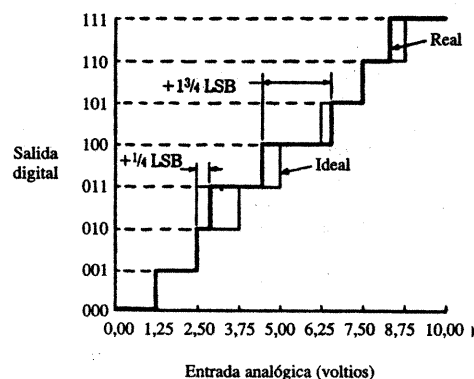
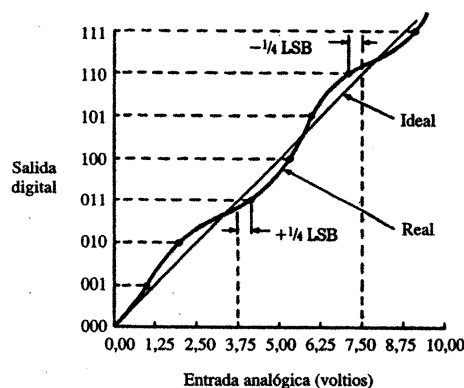


Figura 3.9. La calidad de un convertor A/D se puede expresar de diferentes maneras. **A.** Especificación de la linealidad integral de un convertor A/D. **B.** Especificación de la linealidad diferencial de un convertor A/D. **C.** Especificación de la precisión absoluta de un convertor A/D.

escala positivo. La primera transición de código debe producirse para una tensión de entrada de $+1/2$ LSB. Un error de offset es la desviación entre el primer valor ideal de transición, $+1/2$ LSB, y el valor real que lo produce. Cuando se trabaja con señales bipolares, tensiones positivas y negativas, el error de offset está definido para el fondo de escala negativo. Es decir, el error de offset es la desviación entre el primer valor ideal de transición, $1/2$ LSB por encima del fondo de escala negativo, y el valor que realmente produce la transición. Por otra parte, el error de ganancia es la desviación entre el último valor ideal de transición, $3/2$ LSB por debajo del fondo de escala positivo, y el valor que realmente produce la transición. Los errores de ganancia y de offset son ajustados en el proceso de fabricación, y pueden ser reajustados posteriormente mediante potenciómetros. Los mejores potenciómetros son los denominados multivuelta, ya que presentan una baja deriva tanto con la temperatura como con el tiempo.

La señal analógica de entrada debe escalarse en amplitud con objeto de ajustarla al margen de conversión del convertidor y utilizar así su máxima resolución. Normalmente, un convertidor A/D se puede utilizar directamente con una fuente de baja impedancia, como por ejemplo con la salida de un amplificador operacional de banda ancha. Las variaciones en la corriente de entrada del convertidor A/D se deben normalmente a los cambios de corriente que se producen en su propio convertidor A/D interno, utilizado en funciones de verificación del código producido. La tensión de salida de la fuente, que entrega la señal analógica al convertidor D/A, debe permanecer constante cuando ocurren estas variaciones de corriente.

Las variaciones que sufra la fuente de alimentación también pueden afectar a la precisión del convertidor. El fondo de escala positivo del convertidor puede llegar a ser distinto, produciendo de esta forma un error que será proporcional en todos los códigos binarios; es decir, se produce un error de ganancia. Normalmente, se utilizan fuentes de alimentación reguladas con un rizado inferior al 1 por 100. Además, se utilizan condensadores de tántalo ($\approx 10 \mu\text{F}$) entre la entrada y la salida de la fuente (en bypass), para eliminar componentes eléctricas de ruido. Si se utiliza una fuente de alimentación conmutada, los transitorios y el ruido eléctrico producido por la misma deben ser filtrados para que no afecten a la conversión A/D. También deben utilizarse osciladores de cristal de cuarzo (relojes), sumamente precisos, en todos los circuitos S/H y A/D, con objeto de minimizar los efectos de jitter.

Anteriormente se comentó que la resolución de 16 bits se ha convertido prácticamente en un estándar dentro de los dispositivos digitales de audio; normalmente, con una resolución de 16 bits es más que suficiente para representar fielmente la señal de audio. Cada vez más, los equipos de audio trabajan con un mayor número de bits; por ejemplo, en algunos procesadores digitales de señal (DSP) se trabaja internamente con palabras de 56 bits; esta resolución es necesaria para no acumular los errores de cómputo en los bits menos significativos de la señal de audio. Los discos DVD almacenan datos de 20 o 24 bits. Muchos convertidores A/D y D/A ya trabajan con resoluciones de hasta 24 bits. Sin embargo, con la actual tecnología, es difícil, por no decir imposible, realizar una verdadera conversión a 24 bits. Una resolución a 24 bits representa un nivel de ruido de cuantificación en torno a los -145 dBFS (dB Full Scale —fondo de escala—). Si el fondo de escala fuera de 2 voltios rms, un valor de -145 dBFS correspondería a un valor aproximado de tensión de $0,1 \mu\text{V rms}$. Este nivel es aproximadamente el nivel de ruido térmico en una resistencia de 6 ohmios a temperatura ambiente. El ruido existente en cualquier cadena de audio impediría tener tal resolución. En el procesado de datos de un DSP sí que es conveniente tener una gran resolución, pero es improbable que los convertidores A/D y D/A lleguen a aprovecharla.

Convertidor A/D de aproximaciones sucesivas

Existen muchos tipos de convertidores A/D que se utilizan en distintas aplicaciones. En la digitalización de audio, el margen de selección se reduce debido a exigencias de precisión y rapi-

dez. La conversión A/D por aproximaciones sucesivas (SAR, *successive approximation register*) es un método clásico que permite obtener una calidad óptima en la señal de audio; en la Figura 3.10 se muestra la estructura de un convertidor SAR. El convertidor está formado por un convertidor A/D, dispuesto en un lazo de realimentación, un comparador y un circuito de control. Su funcionamiento en esencia es como sigue: el convertidor compara la tensión analógica de entrada con un voltaje provisional, representativo de un código binario; el código binario va cambiando hasta tener la misma tensión de entrada, dentro de la resolución de trabajo. El convertidor es gobernado por un algoritmo que, bit a bit, establece qué código binario de salida corresponde a la tensión analógica de entrada.

Por ejemplo, considere una tensión analógica de entrada de 6,92 V y un convertidor A/D de 8 bits. En la Figura 3.11 se muestran cada uno de los pasos de la conversión. El bit más significativo del registro SAR se pone a 1, estando el resto a 0; por tanto, al convertidor D/A se le aplica el código binario 10000000. Esta palabra produce una tensión de salida en el convertidor D/A de 5 voltios, la mitad de su valor máximo. El comparador permanece en estado alto al ser la tensión de entrada superior a la tensión producida por el convertidor D/A. El primer bit se guarda como valor definitivo (1). El siguiente bit más significativo del registro SAR se pone a 1 y el código binario resultante (11000000) se aplica al convertidor D/A, produciendo esta vez una tensión de 7,5 V. Esta tensión es más grande que el voltaje de entrada y esto provoca que el

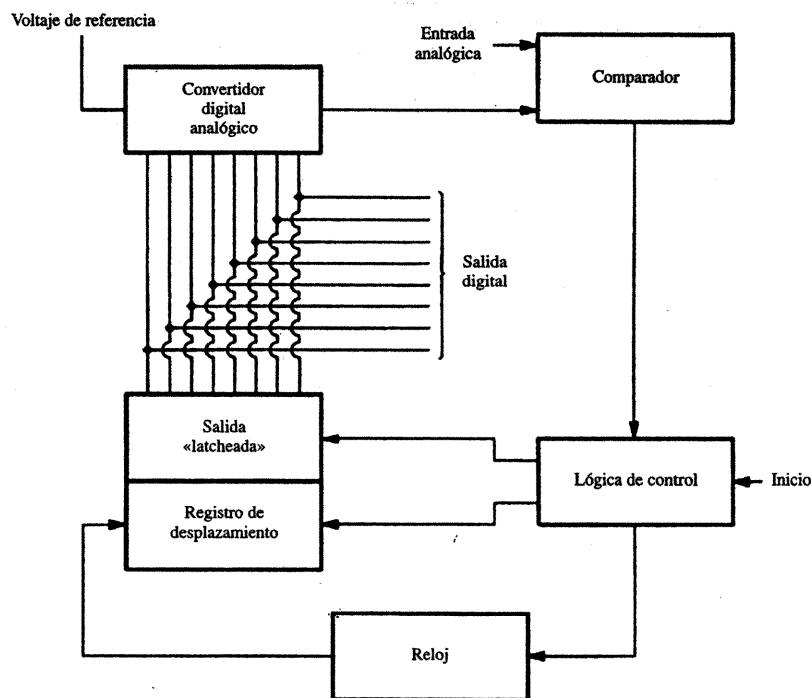


Figura 3.10. Convertidor A/D de aproximaciones sucesivas en el que se muestra un convertidor D/A y un comparador.

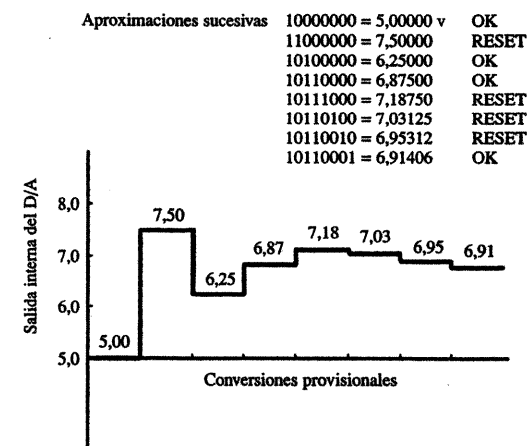


Figura 3.11. Pasos intermedios en una conversión SAR mostrando los resultados de las conversiones provisionales del D/A.

valor que se guarde del segundo bit sea ahora de un 0. El tercer bit más significativo se pone a 1 y el código binario 10100000 se aplica al convertidor D/A, produciendo una tensión de 6,25 voltios. Esta tensión es menor que la tensión de entrada y por lo tanto el valor del tercer bit debe ser un 1. Este proceso continúa hasta llegar al bit menos significativo, obteniendo un código binario final 10110001, el cual representa una tensión de 6,91 V.

Este método de aproximaciones sucesivas debe realizar n conversiones digitales-analógicas en una de las conversiones A/D, donde n es el número de bits del código binario. A pesar de todo este proceso repetitivo, los convertidores por aproximaciones sucesivas son relativamente rápidos y económicos. No obstante, el convertidor debe diseñarse con suma precisión. Por ejemplo, la precisión de un convertidor de 16 bits con márgenes de tensión de ± 10 V, y con un error permisible de $\frac{1}{2}$ LSB, es de 3 mV. Una variación de 10 V en la tensión de entrada del convertidor D/A debe realizarse en tan sólo 100 ns (10^{-9} segundos).

Convertidor A/D con sobremuestreo

Como ya se señaló, un filtro paso bajo analógico presenta bastantes limitaciones de ruido, distorsión, retardo de grupo, rizado en la banda de paso, y es difícil para un convertidor A/D obtener resoluciones superiores a los 18 bits por muestra. En muchas aplicaciones, el filtro paso bajo antialiasing y el convertidor A/D de aproximaciones sucesivas se reemplaza por un convertidor A/D con sobremuestreo con un filtrado digital. La implementación de un filtro digital antialiasing no tiene sentido, ya que la señal analógica es muestreada y digitalizada antes de que se pueda aplicar un filtrado digital. Este enigma ha sido resuelto de forma inteligente; se utiliza un filtro digital de diezmo conjuntamente con la propia conversión A/D. Los filtros digitales se verán con detalle en el siguiente capítulo; aquí solo se verán los fundamentos de la conversión A/D con sobremuestreo.

En una conversión A/D con sobremuestreo, la señal de entrada es filtrada con un filtro paso bajo muy simple, que produce la suficiente atenuación sólo en las frecuencias altas. La señal se

muestrea y se cuantifica con una frecuencia de muestreo muy alta, por lo que obviamente la frecuencia Nyquist también se incrementa notablemente. Después de la cuantificación, se utiliza un filtro digital paso bajo de diezmo para reducir la frecuencia de muestreo y para impedir el aliasing que se produce en esta reducción. Los datos filtrados salen a una frecuencia de muestreo inferior (por ejemplo, 48 o 96 kHz). El filtro paso bajo de diezmo elimina las componentes de frecuencia por encima de la nueva frecuencia Nyquist, impidiendo la aparición de componentes de aliasing cuando la señal de salida del filtro se re-muestrea (infra-muestreada) a la frecuencia de muestreo nominal del sistema.

Considere un convertidor A/D y otro D/A, trabajando ambos con sobremuestreo (el doble de la frecuencia de muestreo original), tal como se muestra en la Figura 3.12. El filtro analógico paso bajo antialiasing limita el ancho de banda de la señal a $1,5 f_s$, donde f_s es la frecuencia de muestreo. El ancho de banda de transición es relativamente grande, desde $0,5 f_s$ a $1,5 f_s$, y produce una buena respuesta en fase. Por ejemplo, podría utilizarse un filtro de *Butterworth* de sólo séptimo orden. La señal se muestrea a continuación a una frecuencia de muestreo f_s . Pasa por el convertidor A/D para ser cuantificada y codificada en un formato binario de n bits. Posteriormente, se filtra digitalmente con un filtro FIR de fase lineal, es decir, con un retardo de grupo constante. Una de cada dos muestras en la salida del filtro se desecha con objeto de reducir a la mitad la frecuencia de muestreo. En la reproducción, un filtro de sobremuestreo dobla la frecuencia de muestreo de los datos, éstos se convierten en una señal analógica y las réplicas de alta frecuencia son eliminadas por el filtro paso bajo de mínimo orden dispuesto en la salida.

Los llamados convertidores sigma-delta utilizan una frecuencia de muestreo muy alta. Al ser tan alta la frecuencia de muestreo, las muestras pueden representarse con muy pocos bits o

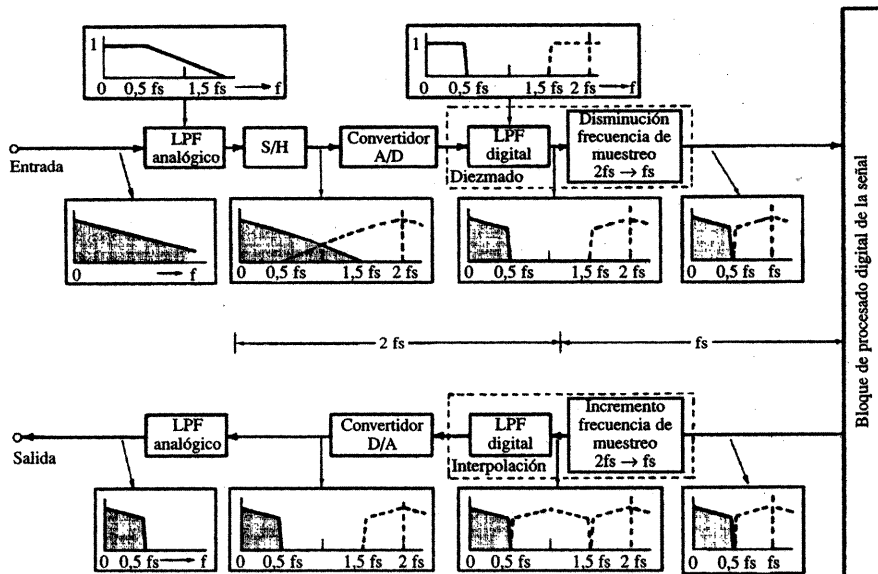


Figura 3.12. Sistema de conversión A/D y D/A con sobremuestreo doble. Los filtros de diezmo y de interpolación disminuyen y aumentan la frecuencia de muestreo al tiempo que eliminan los componentes de aliasing y las réplicas espectrales de la señal (Pioneer Electronics Corporation).

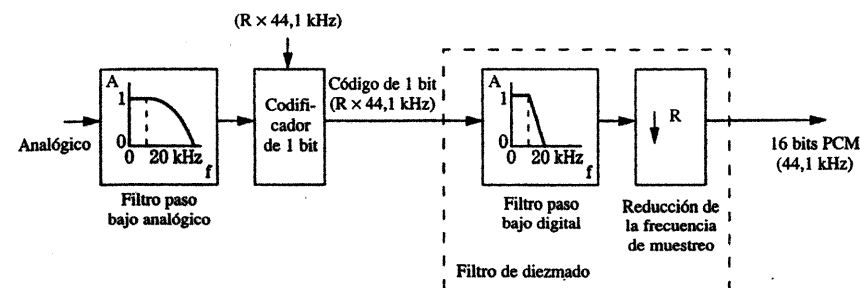


Figura 3.13. Convertidor A/D con sobremuestreo con una codificación de 1 bit a una alta frecuencia de muestreo, y con un filtro de diezmo.

incluso con un solo bit. Además, un modulador sigma-delta produce un moldeado espectral del ruido de cuantificación (*noise-shaping*) para minimizar su efecto en la banda de audio. Mediante un filtro de diezmo la codificación sigma-delta se transforma en una codificación PCM de 16 bits (o más), de menor frecuencia de muestreo. Considere, por ejemplo, una codificación de un solo bit por muestra con un sobremuestreo de 72 veces; esto es, $72 \times 44,1 \text{ kHz} = 3,1752 \text{ MHz}$, tal como se muestra en la Figura 3.13. La banda atenuada del filtro de diezmo se extiende desde 20 kHz hasta la mitad de la frecuencia de muestreo (1,5876 MHz). La conversión A/D de un solo bit simplifica enormemente el diseño del filtro digital. No es necesario entregar un dato filtrado por cada bit de entrada. Al existir un sobremuestreo de 72 veces, sólo se entrega un dato por cada 72 bits de entrada. La estructura del filtro es de tipo transversal, con coeficientes adaptados al factor de diezmo (72). Después del diezmo, con la nueva frecuencia de muestreo de 44,1 kHz, los datos filtrados se redondean a 16 bits.

Además de eliminar los filtros analógicos de acentuadas pendientes de caída en la banda de transición, los convertidores A/D con sobremuestreo ofrecen otras muchas ventajas. La resolución en amplitud es mayor que la que ofrecen los convertidores de aproximaciones sucesivas. El espectro del ruido de cuantificación se extiende por encima de la banda de audio. De esta forma, el ruido contenido en la misma es bastante pequeño. El filtro digital que impide el aliasing cuando se diezman los datos se encarga de eliminar todo el ruido de cuantificación que está fuera de la banda de audio. Los convertidores A/D con sobremuestreo se utilizan cada vez más en la digitalización de la señal de audio. Este tipo de conversión sigma-delta se explica en detalle en el Capítulo 18. En cualquier método de conversión, el objetivo final es la digitalización de la señal de audio, con la señal representada en complemento a dos o en cualquier otro formato binario.

En los equipos de audio digital que trabajan en tiempo real procesando la señal de audio, produciendo por ejemplo un retardo o una reverberación mediante un algoritmo digital —hardware o software—, los datos digitales, obtenidos en un proceso de digitalización, pueden ser utilizados de forma inmediata. En el caso de sistemas de grabación digital es necesario realizar un procesamiento posterior.

PROCESADO EN GRABACIÓN

Después de que la señal analógica sea convertida en una secuencia de números binarios, se deben realizar ciertas operaciones sobre ellos antes de que puedan ser grabados o transmitidos.

Aunque éstas puedan ser específicas de acuerdo con el destino final de los datos, los datos PCM: a) son multiplexados, b) se les añade otros datos (datos redundantes) para protegerlos contra errores, c) se realiza una dispersión o entrelazado para desordenarlos, y d) se les aplica un código de canal. Aunque todo este proceso sea tedioso y sin apenas interés, es crucial preparar correctamente los datos antes de aplicarles el código de canal, para asegurar en último caso que la reproducción de audio se realice satisfactoriamente.

En algunos programas de audio, destinados a grabación o a transmisión, se realiza un proceso de énfasis: un sencillo método para reducir el ruido de la señal. El pre-énfasis es una ecualización en frecuencias altas (ganancia), previa a la transmisión o a la grabación. Después, es necesario realizar una ecualización de de-énfasis para atenuar esas mismas frecuencias. El resultado es una disminución del nivel de ruido. Un énfasis típico tiene como constantes de tiempo 50 y 15 μ s, que corresponden a frecuencias de 3.183 y de 10.610 Hz, con una ecualización de 6 dB/octava entre ambas, tal como se aprecia en la Figura 3.14. El pre-énfasis debe ser señalado en los datos grabados o transmitidos, de forma que en la salida se realice en correspondiente de-énfasis.

En una grabación analógica cualquier error que tenga el sistema queda reflejado en la propia señal, degradándola irremediamente. En una grabación digital, los sistemas de detección y corrección de errores minimizan estos posibles problemas. Si no existiera la corrección de errores, la calidad de la señal de audio quedaría sensiblemente mermada. Son varios los pasos que deben hacerse para combatir los errores. Durante la codificación se añaden datos de paridad; estos nuevos datos se crean a partir de los propios datos de audio para detectar y corregir errores. Para impedir que los errores afecten simultáneamente a los datos de audio y a los datos de paridad (encargados de corregirlos) se realiza una dispersión o entrelazado de todos los datos; esto produce una diseminación en todo el flujo de bits, de manera que el error también se disperse cuando se reordenen los datos en la reproducción. Los datos de paridad, los códigos de chequeo, la redundancia, el entrelazado y la corrección de errores se verá con detalle en el Capítulo 5.

El multiplexado de datos se emplea para formar un flujo de datos en serie. La mayoría de los sistemas digitales de grabación y transmisión trabajan con datos en serie; esto es, los datos están contenidos en un único flujo de información digital. Sin embargo, los datos de un convertidor A/D pueden salir en paralelo; por ejemplo, los datos de 16 bits en complemento a dos pueden salir de forma simultánea. El multiplexor es el elemento encargado de transformar los datos paralelo en datos en serie; en la entrada puede haber múltiples datos simultáneos (varios canales de audio) y en la salida habrá un único flujo de bits, en serie, bit a bit.

Con objeto de simplificar el proceso de grabación o de transmisión, según sea el caso, los datos se agrupan en estructuras definidas, de forma que sea posible la recuperación de los mismos. Existen varias alternativas para realizar este proceso. Los datos multiplexados normalmente se agrupan en tramas; para identificarlas dentro del flujo de datos, se añade un código de sincronización al inicio de las mismas. Un código de sincronización es un patrón de bits, distin-

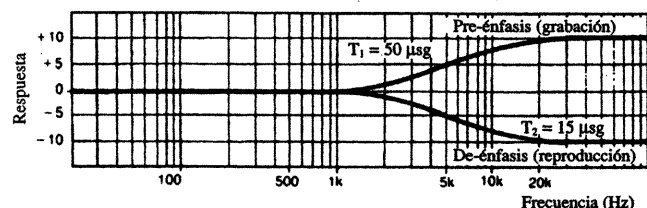


Figura 3.14. El pre-énfasis realza las altas frecuencias en grabación, y el de-énfasis las reduce en la reproducción con objeto de disminuir el ruido de cuantificación.

to de cualquier otro que pueda existir en el flujo de datos, de igual forma que una coma en una frase se distingue del resto de caracteres. En la mayoría de los casos, al inicio de un fichero de datos se incluye también una cabecera que informa acerca del contenido del mismo.

Existen otros datos de numeración y de conteo que permiten identificar cada uno de los datos grabados o transmitidos. Los datos se distribuyen ordenados, de forma secuencial, en el flujo de datos, permitiendo distinguir las diferentes secciones o estructuras realizadas en la grabación o transmisión. Como se apuntó anteriormente, se introducen también datos de corrección de errores. Los códigos de identificación suelen llevar información acerca de algunos parámetros específicos, necesarios para realizar una correcta reproducción de los datos de audio. Por ejemplo, parámetros como la frecuencia de muestreo, la utilización del pre-énfasis, la tabla de contenidos del programa de audio, señal de reloj, seguimiento de pista e incluso información de los derechos de copia (*copyright*).

CÓDIGOS DE CANAL

La codificación de canal es tal vez el proceso que menos se conoce y sin embargo de crucial importancia en un sistema digital de audio. Los códigos de canal han sido acertadamente descritos por Thomas Stockham como «la letra» del audio digital.

La modulación (codificación) de canal se realiza antes de la grabación o de la transmisión. El audio digital está compuesto por unos (1) y ceros (0), pero este tipo de información binaria no se suele transmitir directamente. Un código de canal transmite datos de audio y otro tipo de información suplementaria. Es por tanto una señal modulada que será interpretada en el decodificador para recuperar los datos binarios iniciales y, por tanto, la señal de audio para que ésta pueda ser reproducida correctamente. La modulación va a facilitar la lectura de los datos cuando el soporte de grabación o el canal de transmisión deteriore las transiciones de nivel, representativas de los cambios de estado (nivel alto-nivel bajo, 1-0). Además, con la modulación de canal se consigue una mayor eficiencia de codificación; aunque puede que se incluyan bits ajenos a los de audio, en la mayoría de los casos se aumenta la tasa binaria total de audio (bits/sg).

Un código de canal describe la forma en que la información: a) se modula en una señal de canal, b) se almacena o se transmite, y c) se demodula. Específicamente, los bits de datos se transforman en bits de canal. Las técnicas de modulación permiten solventar toda la problemática que presentan las funciones de transferencia de los soportes físicos de grabación o de transmisión. Un código de canal debe: a) ser autosincronizado para permitir la sincronización en el receptor, b) tener un nivel mínimo en bajas frecuencias para no interferir con los servomecanismos del receptor, c) tener un espectro limitado en frecuencia, d) ser inmune al ruido de canal y e) detectar señales falsas. Desafortunadamente, estos requisitos suelen ser contradictorios y sólo unos pocos códigos de canal son idóneos en las aplicaciones de audio.

La señal de reloj en el decodificador debe ser de la misma frecuencia y sincronizada en fase con la señal transmitida, normalmente implícita en el propio flujo de bits. En la mayoría de los casos, las tramas marcan la sincronización de los datos. Si no hubiera sincronización sería imposible distinguir cada uno de los bits de canal. Incluso con ella, series seguidas de ceros (0) o unos (1) forman una señal continua que impiden que pueda haber una señal de sincronismo implícita. Si no existe otra información de sincronismo es imposible recuperar los datos. Por tanto, los datos normalmente se graban de tal forma que queden claramente reflejados los cambios de transición de los pulsos. Los códigos con una elevada tasa de transiciones permiten regenerar la señal de sincronismo en el receptor; son los llamados códigos autosincronizados.

Así, uno de los objetivos de la modulación de canal es combinar los datos en serie con una señal de reloj para producir una señal autosincronizada. Generalmente, la eficiencia de un código disminuye al introducir una señal de sincronización, ya que la señal de reloj incrementa el número de transiciones, y esto hace aumentar la tasa binaria total (de canal). El contenido de

alta frecuencia de la señal modulada, debido a la señal de reloj implícita, se verá recortado en soportes de grabación de capacidad media, y puede incluso degradarse en líneas/cables de gran longitud. La distancia mínima entre transiciones (T_{\min}) determina la frecuencia máxima de la señal modulada, y normalmente es la máxima frecuencia que soporta el medio de grabación o transmisión. Uno de los parámetros que caracterizan a un código de canal es la relación de densidad, relación entre T_{\min} y el período de bit. Desde el punto de vista del ancho de banda, interesa un código con T_{\min} grande. T_{\max} es la distancia máxima entre transiciones para que sea posible regenerar el sincronismo en el receptor; por tanto, interesa un código de canal con un T_{\max} pequeño.

Las variaciones en el tiempo conocidas como jitter reflejan las variaciones de fase de una señal, apareciendo por tanto una componente de modulación en frecuencia. La correcta recuperación de los datos en una codificación de canal depende del número de períodos de bit que pueden insertarse entre transiciones de nivel (característica propia de cada código), es decir, del número de transiciones que pueden ser detectadas entre T_{\min} y T_{\max} . Uno de los parámetros a considerar en un código de canal es su tolerancia a localizar exactamente la transición. Este parámetro se llama de múltiples maneras: margen temporal, margen de fase o margen de jitter, y se denota por T_w . Especifica la diferencia mínima entre longitudes de onda del código; cuanto mayor sea el margen temporal, mayor inmunidad al ruido de jitter. La eficiencia de un código se evalúa por su relación de densidad (DR, *density ratio*), que es el número de bits transmitidos con relación al número de transiciones de nivel que presenta el código. Una evaluación más completa de la eficiencia del código se obtiene con el parámetro denominado figura de mérito (FoM). Su valor es el producto $DR \times T_w$; cuanto mayor sea este parámetro mayor será la eficiencia del código.

Los datos no se pueden grabar directamente sin modular; la eficiencia conseguida deja mucho que desear. Se obtiene una mayor densidad de grabación (eficiencia) cuando se utiliza un código de canal, aunque éste sea mediocre. La eficiencia, como se ha apuntado, se valora por el número de bits transmitidos con relación al número de transiciones necesarias para ser codificados. Ésta varía entre aproximadamente el 50 y el 150 por 100. Por ejemplo, el código PCM es el menos indicado en grabación digital; deben utilizarse otras técnicas de modulación. Aunque en una grabación digital lo que se trata es de grabar ceros (0) y unos (1), lo que realmente se registra es algo diferente. Normalmente, lo que interesa es grabar transiciones de nivel, más que el propio nivel representativo de los datos modulados. A este respecto, lo más importante en una señal codificada son los instantes en los que ésta cambia de estado.

Una buena codificación siempre tendrá un bajo nivel de continua (dc), ya que la sincronización de la señal puede verse afectada. Generalmente, los sistemas digitales no responden a la corriente continua, por lo que ésta se perderá. Además, la componente dc empeora la relación señal a ruido (S/N). El contenido dc se estima como la relación entre el tiempo en que la señal está a nivel alto (unos o ceros del flujo de datos) y el tiempo en el que la señal está a nivel bajo. El resultado es un valor medio no nulo. Por ejemplo, en una señal NRZ (en la cual los unos se representan como un nivel alto y los ceros como un nivel bajo, o viceversa), una secuencia seguida de ceros o unos daría como resultado un valor dc del 100 por 100. Este valor puede visualizarse calculando la suma digital (DSV, *digital sum value*); el parámetro DSV de un determinado código puede considerarse como la variación de la carga que tendría un condensador de desacoplo de corriente alterna (ac) si se le aplicara dicho código. Referido de otra forma, refleja la componente continua que acumula el código. En la Figura 3.15 se muestran dos códigos distintos y sus valores DSV; en el intervalo de medida, el primero de ellos no presenta valor alguno y el segundo sí que tiene un cierto contenido dc. Este valor dc podría causar problemas en los entrehierros de las cabezas en un sistema de grabación magnética digital, cuya respuesta en baja frecuencia es casi nula. También puede ocasionar problemas de sincronización y problemas en los servomecanismos que controlan el seguimiento de pista y el enfoque de los sistemas ópticos. Estos servos operan siempre en bajas frecuencias y la componente continua puede

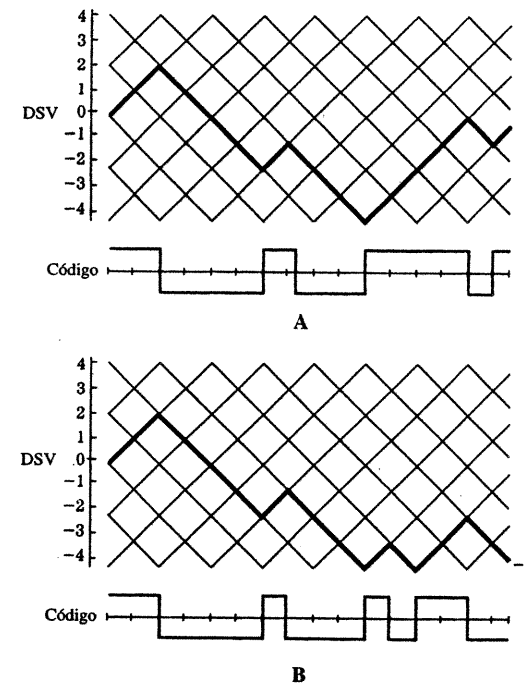


Figura 3.15. La suma digital (DSV) monitoriza el nivel de continua (dc) en el flujo binario de datos. A. Onda codificada libre de componente continua (dc) dentro del intervalo medido. B. Onda codificada con nivel de continua (dc).

inestabilizarlos. Un código libre de corriente continua produce una mejora tanto en el ancho de banda como en la relación S/N de los servomecanismos. Por ejemplo, en el sistema de disco compacto, el margen de frecuencias del código de canal está comprendido entre 20 kHz y 1,5 MHz; los servomecanismos operan sólo en el margen 0-20 kHz.

Un pulso de muestreo periódico es fácil de analizar debido a su naturaleza periódica; mediante un análisis de Fourier se puede apreciar un espectro claramente definido. Sin embargo, un flujo de datos está formado por pulsos que ocurren de manera aperiódica y que de hecho se pueden considerar como aleatorios. Con la densidad espectral de potencia, o con el espectro de potencia, se puede observar su respuesta en el dominio de la frecuencia. Por ejemplo, en la Figura 3.16 se muestran los espectros de tres códigos de canal modulados con secuencias aleatorias: código no retorno a cero (NRZ), modulación en frecuencia modificada (MFM) y código bifase. Idealmente, una señal modulada debe tener: a) un bajo nivel en frecuencias bajas para evitar problemas con los servomecanismos del sistema y para evitar problemas de sincronización, y b) un bajo nivel en frecuencias altas para reducir el ancho de banda de la señal. Obsérvese que el código NRZ presenta un alto nivel de continua. Los códigos bifase (existen varios tipos, uno de ellos es el FM) tienen una distribución espectral de banda ancha. El código MFM presenta, por el contrario, una distribución espectral bastante estrecha. Los códigos MFM y bifase son similares, ya que apenas tienen componentes de baja frecuencia (inferiores a

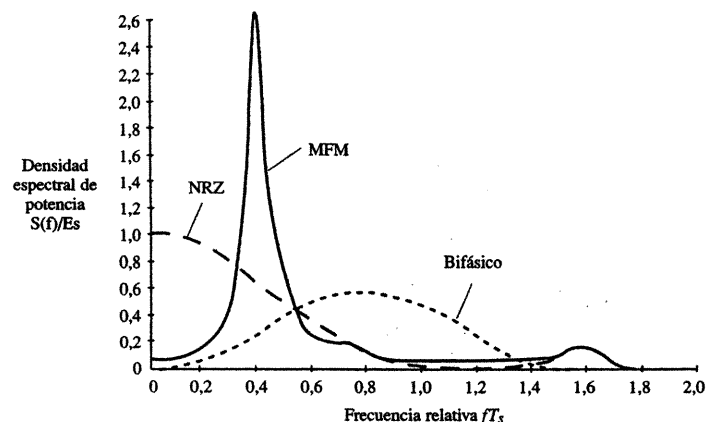


Figura 3.16. La densidad espectral de potencia muestra la respuesta de un flujo binario de datos aleatorios. El código NRZ tiene un alto contenido dc; el código MFM presenta un espectro muy estrecho; el código bifásico presenta una amplia distribución espectral de energía.

$0,2f$, con $f = 1/T$). Para una frecuencia $f = 500$ kHz, por ejemplo, estos dos códigos serían idóneos, ya que las señales de los servomecanismos no llegan a los 15 kHz. El código NRZ tiene un fuerte contenido en continua y probablemente causaría algún tipo de conflicto con los servos.

Los códigos de canal también se diseñan para que los datos que llevan sean totalmente diferenciables (patrones distintos) y que, de esta manera, la codificación sea más robusta frente a errores. Por ejemplo, en el código 8-14 (EFM, *eight to fourteen modulation*) utilizado en el disco compacto, símbolos de 8 bits se transforman en otros de 14 bits, seleccionados de tal forma que exista la máxima diferencia entre ellos. Así, los datos no válidos se detectan más fácilmente. De forma similar, los símbolos se pueden generar en base a símbolos precedentes de manera que el decodificador sea capaz no sólo de reconocer un símbolo, sino también de reconocer los símbolos que le preceden. Esta técnica de modulación utiliza un diagrama de generación en el que todas las transiciones están perfectamente definidas en función de los símbolos adyacentes.

Como ya se ha señalado, en muchos códigos la información está realmente en las transiciones, no en la dirección (de nivel bajo a nivel alto o viceversa) de las mismas. Esto es de enorme importancia, ya que hace al código independiente de la polaridad; la información no varía si la señal se invierte. El código EFM utilizado en el disco compacto presenta, por ejemplo, esta propiedad.

Códigos simples

Las entradas a un código de canal son los estados lógicos «0» y «1». Podemos adoptar una relación entre los estados lógicos y la amplitud del código; por ejemplo, un 1 significaría un nivel alto y un 0 un nivel bajo. Sin embargo, existen otras muchas posibilidades de relación; por ejemplo, en una variante del código de modulación en frecuencia (FSK), un 1 produce una salva de 100 kHz mientras que un 0 produce otra de 150 kHz. Esta técnica de codificar dos estados lógicos se adapta «como anillo al dedo» a los sistemas digitales de grabación; las varia-

ciones que pueda sufrir el soporte físico de grabación prácticamente no afectan a la recuperación de los datos. Puesto que estamos ante un método robusto de grabación, es posible alcanzar grandes densidades de grabación. Existen distintos códigos de modulación que adaptan la señal modulada al soporte específico de grabación. De todos los posibles, sólo unos pocos se aplican para la grabación digital de audio, ya sea magnética u óptica; en la Figura 3.17 se muestran algunos de ellos.

El código más sencillo envía un pulso cada vez que aparece un «1»; cuando aparece un «0» no hace nada. Ésta es la razón de su nombre: *retorno a cero* (RZ); el nivel de la señal siempre retorna a cero al final del período de bit (T).

El código *no retorno a cero* (NRZ) también puede considerarse como un código básico; los estados lógicos «1» y «0» se representan directamente como niveles alto y bajo. La dirección de la transición al principio de cada período de bit indica si es un «1» o un «0». El intervalo mínimo es T , mientras que el máximo intervalo puede llegar a ser infinito (cuando los datos no cambian); por lo tanto, este código sufre algunos de los problemas anteriormente apuntados que limita su aplicación en los sistemas de audio: las ristas de ceros o unos no producen transiciones en la señal, no pudiéndose recuperar la señal de reloj de la misma. Además, tiene un gran nivel de continua. La densidad de los datos (número de bits por transición de nivel) para el código NRZ es 1.

El código *no retorno a cero invertido* (NRZI) es similar al código NRZ, excepto que los unos están representados directamente por las transiciones de nivel (de alto a bajo y viceversa) en la mitad del período de bit; los ceros no producen transiciones de nivel. Por ejemplo, cualquier variación del flujo magnético en una grabación magnética indicará un «1». Con esta modulación, la señal es inmune al cambio de polaridad. El intervalo mínimo es T y el máximo

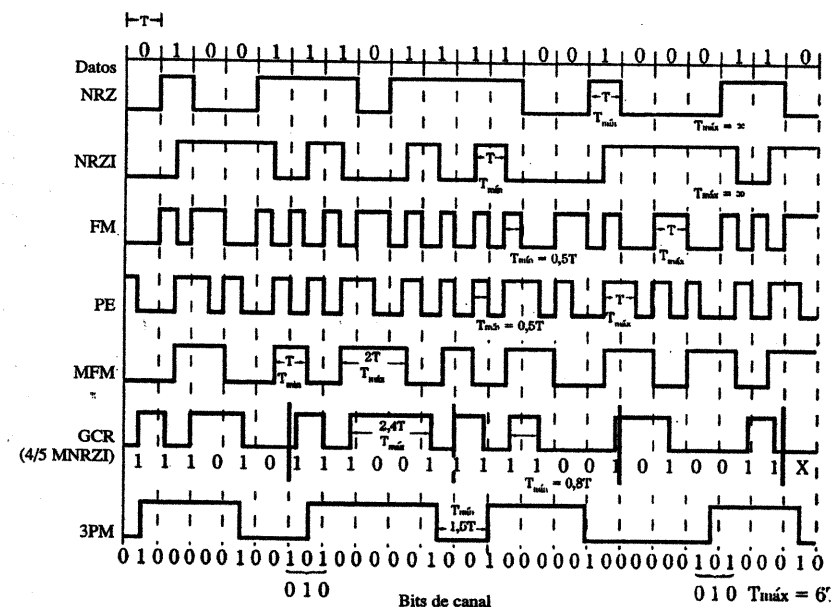


Figura 3.17. Comparación de códigos simples y códigos de grupo para una misma secuencia binaria de datos.

es infinito. La señal de reloj no puede extraerse directamente de la señal. Una serie continua de «1» genera transiciones en cada período de bit, por lo que la frecuencia de la señal modulada, para este caso concreto, es la mitad de la frecuencia de la señal del reloj. Una secuencia seguida de ceros no produce transiciones. La densidad de los datos es 1.

En el *código de modulación en frecuencia binaria* (FM), también conocido por el nombre de *código de marcas bifásicas* (BMC), los «1» producen dos transiciones de nivel y los «0» sólo una; este código es una particularización (extrema) del código FSK. Invierte los «papeles» de los estados lógicos «0» y «1». El intervalo mínimo es $0,5 T$ y el máximo es T . No existe componente continua. En el peor de los casos, hay dos transiciones en cada período de bit, produciendo una relación de densidad de $0,5$ o, lo que es lo mismo, una eficiencia del 50 por 100. Este código se emplea en el protocolo estándar de interconexión digital AES3, descrito en el Capítulo 13.

En la *codificación de fase* (PE), también conocida por los nombres de *modulación de fase* (PM), *modulación de nivel bifásica* o *código Manchester*, los «1» se codifican como transiciones negativas y los «0» como transiciones positivas. A las series seguidas de ceros o unos se les aplica la misma regla, por lo que hay que añadir una transición más. Estos códigos siguen las mismas técnicas de la modulación por fase. El intervalo mínimo es $0,5 T$ y el máximo es T . El código no tiene componente continua y es autosincronizado. La relación de densidad es $0,5$.

En la *modulación modificada de frecuencia* (MFM), también conocida por los nombres de *modulación por retardo* o *código Miller*, los unos se codifican tanto con transiciones negativas como positivas a mitad del período de bit. Los ceros aislados entre unos no producen transiciones; una serie continua de ceros produce transiciones al final de cada período de bit. Cada bit de información se codifica como dos bits de canal. El número de ceros entre unos debe estar comprendido entre uno y tres (ceros y unos de canal —mitad de bit—). El intervalo mínimo de este código es T y el máximo es $2 T$. Es un código autosincronizado y puede tener algo de componente continua. La relación de densidad es 1.

Códigos de grupo

Los códigos simples como el NRZ y el NRZI codifican cada uno de los bits de datos en bits de canal. Los códigos de grupo siguen una estrategia de codificación más compleja con objeto de ganar en eficiencia. Utilizan tablas de código para transformar datos de m bits en otros datos de n bits; los datos de salida se seleccionan en función de las características del propio código y en función de la singularidad de los mismos para permitir una fácil detección de errores. La tasa de un código de grupo es m/n . Esta tasa debe ser igual al margen de jitter tolerado. En algunos códigos de grupo no existe una correspondencia biunívoca entre los datos de entrada y los datos de salida; puede variar en función de la propia secuencia de datos. La eficiencia del código aumenta siguiendo este tipo de estrategia.

Los códigos de grupo también pueden considerarse como códigos de longitud finita (RLL, *run-length limited*). Específicamente, estos códigos detectan las transiciones sólo en los múltiplos del período de bit, tal como se muestra en la Figura 3.18. Aquí no sirve la distinción entre señal de datos y transiciones de reloj, hemos de hablar ahora de número mínimo de ceros d y de número máximo de ceros k entre unos («1»). Los valores T_{\min} y T_{\max} definen la longitud del código y, por tanto, los límites de su densidad espectral de potencia; obviamente, la relación de densidad, el nivel dc y la señal de reloj implícita dependen de estos dos parámetros: d y k . La relación de densidad es igual a T_{\min} , es decir, $DR = T_{\min} = (d + 1)(m)/n$. De forma similar, el margen de jitter es $T_w = m/n$ y la figura de mérito es $FoM = (d + 1)(m^2)/n^2$.

Como siempre, la relación de densidad debe estar en función de otros factores, como por ejemplo de la sincronización implícita de la señal. Normalmente, d se escoge bastante grande para aumentar la densidad, y k también tiende a ser un valor alto, como máximo, que permita una autosincronización estable del código. Obsérvese, por el contrario, que una alta relación k/d

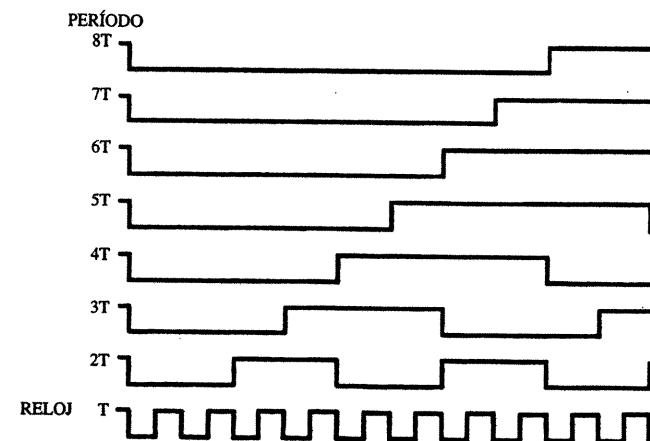


Figura 3.18. Los códigos de longitud límite (RLL) regulan el número de transiciones que representan el código del canal. De esta manera los espacios de transición pueden ser cualquier múltiplo del período del canal, aumentando la densidad de los datos.

puede producir secuencias con un alto contenido en continua; éste es el principal inconveniente de los códigos RLL. Las longitudes máxima y mínima del código determinan, a su vez, el margen de variación de la tasa binaria del código; el contenido espectral del mismo puede moldearse jugando con los valores de estos dos parámetros.

Los códigos RLL utilizan una serie de reglas de conversión para transformar los bits de los datos de entrada en bits de canal, de acuerdo con una determinada relación entre ellos. Un bit de canal no corresponde a un bit de datos; los bits de canal pueden considerarse como pequeñas ventanas temporales, fracciones del período de bit. La densidad de los datos puede aumentarse incrementando el número de transiciones por período de bit. A los bits de canal normalmente se les aplica un nuevo código de canal, el NRZI; los unos del canal están representados por las transiciones de nivel a mitad de bit (los ceros no producen transiciones). La tasa binaria del canal (bits/sg) es generalmente mayor que la tasa binaria de datos (antes de modular); la densidad aumenta cuando están perfectamente diferenciadas las longitudes del código. Generalmente, los códigos RLL son sólo viables en canales con poco ruido; por ejemplo, la mínima relación S/N requerida aumenta a medida que lo hace el parámetro d . Afortunadamente, las características de los discos ópticos permiten el uso de los códigos RLL. Técnicamente, los códigos NRZ y NRZI son códigos RLL de valores $d = 0$ y $k = \infty$; el código MFM puede considerarse como un código RLL (1,3), es decir, $d = 1$ y $k = 3$.

En el *código de grabación de grupos codificados* (GRC), los datos se dividen en grupos de 4 bits, los cuales se transforman en palabras de 5 bits, según se muestra en la Tabla 3.1, y se les aplica el código NRZI. Esta codificación se la conoce habitualmente por el nombre de $4/5$ MNRZI (NRZI modificada). Existe una transición por cada tres bits, el intervalo mínimo es $0,8 T$ (debido a la relación $4/5$) y el intervalo máximo es $2,4 T$. Se permiten unos seguidos, ya que el parámetro $d = 0$ y el máximo número de ceros entre unos es de 2 ($k = 1$). El código es autosincronizado con gran inmunidad al jitter, pero presenta un cierto nivel de continua. La relación de densidad es de $0,8$.

El *código de modulación de tres posiciones* (3PM) es un código RLL (2,7) adaptativo. Tres bits de entrada se transforman en grupos de 6 bits con un $T_{\min} = 1,5 T$ y un $T_{\max} = 6 T$, tal como

Tabla 3.1. Conversión del código GCR (o $\frac{1}{5}$ MNRZI). Los grupos de 4 bits de información se codifican en grupos de 5 bits, y se escriben de acuerdo con el código NRZI

| Bits de datos | Bits de canal |
|---------------|---------------|
| 0000 | 11001 |
| 0001 | 11011 |
| 0010 | 10010 |
| 0011 | 10011 |
| 0100 | 11101 |
| 0101 | 10101 |
| 0110 | 10110 |
| 0111 | 10111 |
| 1000 | 11010 |
| 1001 | 01001 |
| 1010 | 01010 |
| 1011 | 01011 |
| 1100 | 11110 |
| 1101 | 01101 |
| 1110 | 01110 |
| 1111 | 01111 |

se muestra en la Tabla 3.2. Debe haber al menos dos ceros entre unos. Al unirse las palabras del código, puede aparecer la secuencia 101. Esta secuencia viola la regla de $T_{\min} = 1,5 T$. Para prevenir esto, la secuencia 101 se sustituye por 010; el último bit del código se reserva para realizar una correcta unión entre las palabras y cumplir las reglas del mismo. Su nombre se debe a las tres posiciones existentes en la distancia mínima ($d + 1$). Es un código autosincronizado. Comparándolo con el código MFM, la distancia mínima entre transiciones es de sólo 0,5 T mientras que en el MFM es de T. Por tanto, la densidad del código 3PM es un 50 por 100 superior. Sin embargo, la distancia máxima entre transiciones es un 100 por 100 mayor, y su margen de jitter es un 50 por 100 peor. El código 3PM también presenta un cierto contenido en dc y la densidad de datos es de 1,5.

Tabla 3.2. Conversión del código 3PM (2,7). Grupos de tres bits se codifican en palabras de seis bits en el que el intervalo mínimo es 1,5 T

| Bits de datos | | | Bits de canal | | | | | |
|---------------|---|---|---------------|----|----|----|----|----|
| | | | P1 | P2 | P3 | P4 | P5 | P6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| ←T→ | | | ←T→ | | | | | |

Como ya se ha apuntado, los códigos RLL son códigos eficientes debido principalmente a que las transiciones de nivel ocurren en múltiplos del período de bit. El soporte físico de grabación es el que determina la longitud mínima del código. Sólo se seleccionan los códigos que aseguran una longitud mayor que la mínima que permite el soporte. Por tanto, todos los datos que entran a la tabla de conversión producen aquellos códigos que cumplen con este requisito. El código EFM es buen ejemplo de esta técnica de codificación. El incremento mínimo entre códigos distintos corresponde a un tercio de la mínima longitud de onda resoluble (impuesta por el soporte físico de grabación). La densidad del código, por el contrario, no se triplica, ya que grupos de entrada de 8 bits se transforman en códigos de 14 bits. Para la recuperación de los datos, es necesario un reloj con un período de 1 T.

El código EFM, modulación ocho a catorce, se utiliza para la grabación de datos en el disco compacto (CD); es un código RLL (2,10) eficiente y altamente estructurado. Bloques de datos de 8 bits se transforman en otros de 14 bits mediante una tabla de conversión que realiza una asignación arbitraria y unívoca. Los unos están separados al menos por dos ceros ($d = 2$) y no por más de 10 ($k = 10$). Es decir, el T_{\min} está formado por tres bits de canal y T_{\max} por 11. Un «1» produce una transición a mitad del período de bit de canal, representado en la superficie del disco compacto por una variación en la profundidad de la misma. Con este código se puede conseguir una alta densidad. Además, se utilizan tres bits denominados «bits de merging» que sirven para concatenar las palabras del código, impidiendo que se violen las reglas del mismo (T_{\min} y T_{\max}); por tanto, grupos de 8 bits se transforman finalmente en otros grupos de 17 bits. Estos tres bits disminuyen la densidad del código, pero añaden otras ventajas al mismo. T_{\min} es 1,41 T y T_{\max} es 5,18 T. La eficiencia teórica de grabación se calcula multiplicando el mínimo número de bits de canal ($d + 1$) por el factor corrector del código 8/17, dando un valor de 24/17 o, lo que es lo mismo, una relación de densidad de 1,41. Esto es, se pueden registrar 1,41 bits en la mínima longitud de onda (longitud del pit del CD). Consideraciones prácticas de implementación, tal como la relación S/N y el jitter en la recuperación del sincronismo, obligan a tener realmente una densidad cercana a 1,25. En cualquier caso, siempre existen más bits grabados que transiciones. Los denominados bits de merging eliminan la componente continua del código pero reducen su eficiencia en cerca de un 6 por 100. La tabla de selección del código se obtuvo mediante un algoritmo, desarrollado para optimizar esta conversión. El código EFM es bastante tolerante a errores, proporciona una alta densidad de grabación y garantiza una recuperación estable del sincronismo en la señal de reloj. Se utiliza en el CD, el cual se estudiará en detalle en el Capítulo 8. El código EFMplus, utilizado en el formato de grabación DVD, se tratará en el Capítulo 11.

La modulación cero (ZM) es un código RLL con $d = 1$ y $k = 3$; más que una codificación de grupo, utiliza una estrategia de modulación basada en una convolución. Cada bit se transforma en otros dos, dependiendo tanto de los bits precedentes como de los posteriores; y posteriormente, al igual que en todos los códigos RLL, se aplica el código simple NRZI. El código está optimizado para sistemas de grabación magnética. El flujo de bits de datos se interpreta como uno de los casos siguientes: serie continua de ceros, dos ceros separados entre un número impar de unos (o número de unos) o dos ceros separados por un número par de unos. Los dos primeros casos se codifican de forma idéntica al código Miller, y en el último caso también, pero con la salvedad de que los unos se codifican como si fueran ceros y sin alternar en las transiciones. La relación de densidad es aproximadamente 1. No existe contenido en continua.

El código HDM-1 (*high density modulation*) se utiliza en los sistemas de grabación magnética multipista según el formato DASH (se verá en el Capítulo 6). El código 8/10 se emplea en el sistema de grabación magnética estéreo según el formato DAT (se estudiará en el Capítulo 7). En la Tabla 3.3 se muestran las especificaciones de algunos códigos (simples y de grupo).

Tabla 3.3. Especificaciones de códigos simples y códigos de grupo

| Parámetro | NRZ | PE | | ZM | GCR | 3PM | EFM | HDM-1 | 8/10 |
|--|----------|---------|---------|--------------|----------|---------|-----------|---------|----------|
| | NRZI | FM | MFM | | | | | | |
| Margen de ventana (T_w) | T | $0,5 T$ | $0,5 T$ | $\sim 0,5 T$ | $0,8 T$ | $0,5 T$ | $0,471 T$ | $0,5 T$ | $0,8 T$ |
| Transición mínima (T_{\min}) | T | $0,5 T$ | T | $\sim T$ | $0,8 T$ | $1,5 T$ | $1,41 T$ | $1,5 T$ | $0,8 T$ |
| Transición máxima (T_{\max}) | ∞ | T | $2T$ | $\sim 2T$ | $2,4 T$ | $6T$ | $5,18 T$ | $4,5 T$ | $3,2 T$ |
| Contenido DC | sí | no | sí | no | sí | sí | no | sí | no |
| Frecuencia de reloj CLK | no | $2/T$ | $2/T$ | $2/T$ | $1,25 T$ | $2/T$ | $17/8T$ | $2/T$ | $10/8 T$ |
| Tasa de densidad DR | 1 | 0,5 | 1 | ~ 1 | 0,8 | 1,5 | 1,41 | 1,5 | 0,8 |
| Figura de mérito $T_w T_{\min} (T^{-2})$ | 1 | 0,25 | 0,5 | 0,5 | 0,64 | 0,75 | 0,664 | 0,75 | 0,64 |
| Tasa máxima/mínima $T_{\max} T_{\min}$ | ∞ | 2 | 2 | 2 | 3 | 4 | 3 | 3 | 4 |

Aplicaciones de los códigos

A pesar de las diferencias, existe cierta semejanza en los códigos utilizados tanto en el registro óptico como en el registro magnético. Las diferencias estriban en el propio diseño del código, adaptándolo específicamente al soporte físico de grabación. Algunos códigos, como el 3PM, se diseñaron específicamente para la grabación magnética, aunque últimamente también se está empleando en sistemas ópticos. Aun así, la mayoría de los códigos se emplean en aquellas aplicaciones para las cuales fueron diseñados, magnéticas u ópticas.

El registro óptico de los datos requiere un código de una alta densidad. En este soporte el jitter no es un factor a tener en consideración, el disco está prácticamente libre de jitter. De forma similar, las longitudes del código pueden hacerse mayores en el disco, al no haber prácticamente problemas en la recuperación del reloj. Éste proporciona la sincronización de los datos y el control del motor de arrastre. La señal debe tener el suficiente número de transiciones para poder extraer la señal del reloj (detectando, por ejemplo, los bordes de los pits), pero la distancia máxima entre transiciones debe ser la mínima posible. Los arañazos y la suciedad en la superficie del disco pueden cambiar la forma de la señal, añadiendo un pequeño ruido de baja frecuencia. Esto trae como consecuencia que la señal tenga un menor valor medio. Si el nivel de la señal cae por debajo del umbral de detección se produce un error en la lectura de datos. El ruido de baja frecuencia puede ser atenuado mediante un filtro paso-alto, pero sólo si la señal no tiene componentes de baja frecuencia. Por tanto, interesa un código que no tenga contenido en continua y que permita introducir un filtro paso-alto para eliminar el ruido producido por la suciedad del disco. Los códigos RLL cumplen este requisito y se adaptan perfectamente a las características ópticas de grabación.

En grabación magnética se limita la longitud máxima de los códigos, ya que los sistemas magnéticos no responden en continua (dc). Tienen una longitud bastante pequeña para permitir una recuperación de la señal de reloj, en un medio propenso al ruido de jitter como lo es el soporte magnético. Ésta es la razón de que los códigos utilizados en grabación magnética sean pobres en eficiencia. El jitter en este tipo de soportes flexibles es más acentuado que en un sistema óptico (disco). En presencia de jitter, el número de períodos de bit que pueden transcurrir antes de que llegue una transición está determinado por la necesidad de impedir cualquier ambigüedad en el número de períodos entre transiciones. Por tanto, la codificación en cinta magnética limita la longitud máxima admisible en los códigos. El jitter se discutirá con mayor profundidad en el Capítulo 4. En cualquier caso, siempre es necesario realizar un análisis detallado del funcionamiento del código en la correspondiente aplicación.

En muchos de los circuitos del receptor se utiliza un circuito enganchado en fase (PLL, *phase-locked loop*) para resincronizar el código de canal, procedente del medio físico de graba-

ción. El código de canal actúa como referencia de entrada. El lazo del PLL compara las diferencias de fase entre su propia señal y la que entra como referencia; la señal de error sirve de control a un oscilador controlado por tensión para que genere exactamente la señal de reloj y elimine el jitter de la señal de entrada. La comparación se produce en cada transición de la señal, contándose los períodos que transcurren hasta la siguiente transición y recuperando de esta forma la señal modulada. El PLL se engancha en fase mediante la detección de un código de sincronización, transmitido en la propia señal modulada. En los códigos RLL se puede utilizar un determinado código que viole las reglas del mismo y que sirva de sincronismo al PLL; por ejemplo, en el CD, dos códigos de duración $11T$ preceden a una trama de datos modulada con el código EFM; el reproductor se puede enganchar en fase con los bits de canal y no los interpretará como datos (códigos distintos a los creados en la codificación de canal EFM).

Una vez realizada la codificación de canal, los datos están preparados para ser grabados o transmitidos, según sea el caso. Por ejemplo, si se trata de una grabación magnética digital, los datos modulados deben pasar por un circuito que genere la corriente necesaria para producir una grabación magnética por saturación. Las variaciones de flujo magnético grabadas en la cinta representan las transiciones de la señal modulada. La señal grabada pudiera parecer que está altamente distorsionada; nada más lejos, no afecta en absoluto a la integridad de los datos y permite alcanzar una alta densidad de grabación. En sistemas ópticos como el disco compacto, los datos modulados están representados por pits (hendiduras o huecos en la superficie del disco). A su vez, cada extremo del pit representa un «1» (bit de canal) y el espacio entre ellos representa una serie de «0». La grabación de los datos en un soporte físico, la transmisión de los mismos o el procesamiento digital en tiempo real son los elementos de audio que marcan el final de la cadena de digitalización.

Capítulo 4

REPRODUCCIÓN DIGITAL DE AUDIO

Los sistemas de grabación y reproducción de la señal de audio actúan como transductores de entrada y salida al dominio digital. Convierten la forma de onda del audio analógico en una señal apropiada para su tratamiento, almacenamiento o transmisión digitales, para posteriormente reconvertir la señal a su forma analógica. En un sistema PCM lineal, las funciones de los circuitos de reproducción se invierten con respecto a las del almacenamiento, incluyendo, además, la corrección de la base de tiempos, demodulación, demultiplexación, corrección de errores, conversión digital/analógica, circuito de muestreo y retención de la salida, y filtro recuperador, siendo hoy en día también habituales los filtros digitales de sobremuestreo.

PROCESO DE REPRODUCCIÓN

La cadena de reproducción de la señal (Fig. 4.1) recibe una señal codificada degradada, debiendo finalmente convertirla en una forma de onda lo más parecida posible a la señal original. El objetivo de los circuitos de reproducción es minimizar los efectos negativos que sobre la señal tienen el almacenamiento y la transmisión de los datos; por ejemplo, las limitaciones derivadas de las variaciones mecánicas y posibles pérdidas de datos que poseen los medios de almacenamiento. En los sistemas de almacenamiento analógicos, los problemas deben ser corregidos dentro del propio medio; por ejemplo, para minimizar el Wow & Flutter en un giradiscos, es la velocidad de giro del disco la que debe tener una gran precisión. En los sistemas digitales, y dada la gran densidad de datos, la posibilidad de aparición de errores es mucho mayor; y sin embargo, el procesado digital de la señal ofrece la posibilidad de corregir la mayoría de los mismos. Por ejemplo, la memoria intermedia («buffer») de los circuitos de reproducción minimiza los efectos de las variaciones de temporización de los datos, actuando como elementos de corrección de estos errores.

Para conseguir una elevada densidad de datos en el almacenamiento o transmisión de la señal, permitimos que se pierda gran parte de la fidelidad de la forma de onda de la señal codificada, por lo que ya no tiene unas características tan definidas como las que poseían los datos originales. Por ejemplo, una señal almacenada digitalmente, tal y como se lee de la cabeza de reproducción no aparece claramente delineada, sino acompañada de ruido y redondeada debido a la limitación del ancho de banda. Un circuito conformador de onda identifi-

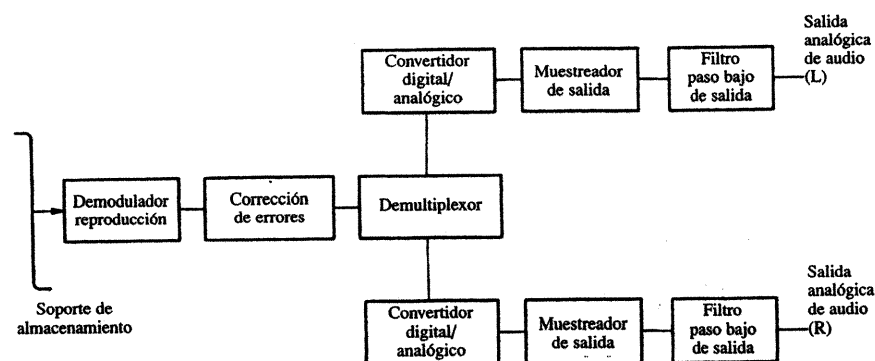


Figura 4.1. Sección de reproducción de un sistema PCM lineal. En la práctica se utilizan filtros digitales de sobremuestreo en la salida.

ca las transiciones de la señal, reconstruyéndola como una señal digital válida. Así, los datos pueden ser recuperados sin penalización, a pesar del deterioro sufrido por la forma de onda de la señal.

Los impulsos de sincronización y el resto de señales de reloj dentro de una cadena de datos son identificados y usados por los circuitos de corrección de base de tiempos para sincronizar la reproducción de la señal, delimitar individualmente cada trama de datos y determinar el valor binario de cada pulso. En la mayoría de los casos, la señal recibida contiene errores de temporización (como el «jitter»), pero los circuitos de lazo enganchado de fase (PLL) y memorias de almacenamiento intermedio («buffers») son utilizados para minimizar este problema. Un «buffer» puede ser comparado con un depósito de agua: el agua puede ser vertida en él descuidadamente, pero una espita en su parte inferior permite mantener un flujo constante de salida. Específicamente, un «buffer» es una memoria en la cual los datos se introducen irregularmente, tal y como son recibidos; sin embargo, la extracción de los datos se realiza con una velocidad controlada, asegurando así una temporización muy precisa en la salida de los datos. De esta forma, un conjunto de muestras pueden ser extraídas de un equipo a la misma velocidad con la que fueron tomadas. A este proceso se le denomina corrección de base de tiempos, y será posteriormente discutido en este mismo capítulo.

Los datos de audio modulados, ya sea en HDM-1, EFM, MFM o cualquier otro código de canal, son normalmente demodulados a NRZ, un código simple en el que el nivel de la señal representa directamente la información binaria (así los datos de audio retoman su sencilla forma binaria, estando preparados para el siguiente paso del proceso de reproducción). Además, se efectúa el demultiplexado de los datos, restableciendo la estructura de bits en paralelo que tenían originalmente. El circuito demultiplexor tiene como entrada una cadena serie de bits, va contando éstos mediante su reloj, y cuando completa una palabra de audio la presenta en su salida con todos los bits a la vez, formando lo que se denomina un dato en paralelo.

Los datos de audio y los códigos de corrección de errores son identificados y separados del resto de datos de la trama. A partir de los códigos de corrección de errores, se comprueba que los datos de audio no han sufrido errores desde su codificación. Debido a la elevada densidad de datos usada tanto en transmisión como en almacenamiento, la aparición de errores es inevitable, variando sólo la frecuencia de aparición y la gravedad de los mismos. Previamente, en la etapa de almacenamiento, los datos fueron dispersados (entrelazado) cuando constituían una cadena serie de bits, asegurando que un defecto en el soporte o interrupción momentánea de la

transmisión no afectara a datos consecutivos. Los circuitos de corrección de errores deben ahora desentrelazar los datos, tomando de nuevo su orden secuencial. Así, los errores causados por los defectos del soporte aparecen dispersos dentro de la cadena de datos, con lo que pueden ser más fácilmente corregidos. Posteriormente se comprueba la integridad de los datos empleando técnicas de redundancia. Cuando la paridad calculada no coincide con la paridad recuperada de la cadena de datos, es probable que haya ocurrido un error, aplicando los algoritmos de corrección de errores para calcular los valores correctos. Si los errores se mantienen dentro de una determinada tolerancia, pueden ser detectados y corregidos con absoluta fidelidad a los datos originales, haciendo de la reproducción y transmisión digital una técnica altamente fiable. Si los errores son demasiado extensos para su corrección, se utilizan técnicas para enmascararlos. La más simple consiste en repetir el último dato correcto hasta que el error desaparece. Otra alternativa es el empleo de técnicas de interpolación, para calcular nuevos datos que construyan un puente sobre los datos erróneos. Una discusión completa de las técnicas de corrección de errores podrá encontrarse en el Capítulo 5. La cadena serie de bits consiste en los datos originales de audio, o al menos los datos tan parecidos a los originales como nos permita el sistema de corrección de errores. Abandonando la circuitería del proceso de reproducción, recapitulemos, los datos han ganado una estabilidad en el tiempo que no tenían, han sido demultiplexados, desentrelazados y corregidos los errores que se produjeron en el almacenamiento/transmisión. Ahora ya están preparados para el proceso de conversión digital/análogo, quedando únicamente el filtrado digital previo, tal y como se verá en la próxima sección.

CONVERSIÓN DIGITAL/ANALÓGICA

El convertidor digital/análogo (D/A) es uno de los elementos críticos en la cadena de reproducción. Así como el convertidor analógico/digital (A/D) determina en buena medida la calidad total de la señal codificada, el convertidor D/A determina con qué precisión la señal digitalizada será devuelta al dominio analógico. La tarea que tiene que desarrollar exige una precisión muy elevada, y como ejemplo podemos considerar que si con un fondo de escala de ± 10 V un convertidor de 16 bits debe entregar pasos de 0,000305 voltios, en un convertidor de 24 bits los escalones de tensión son de tan sólo 0,00000119 voltios. Los convertidores D/A tradicionales suelen tener una alinealidad similar a la de los convertidores A/D, mientras que los convertidores sigma-delta, funcionando básicamente en el tiempo más que en amplitud, pueden mejorar el comportamiento en amplitud, aunque para ello deban utilizar un conformador de ruido para reducir el umbral de ruido dentro de la banda de audio. Afortunadamente, hoy en día disponemos de convertidores D/A de elevada calidad a unos costes relativamente bajos.

Los convertidores D/A tradicionales trabajan con palabras en paralelo, estando sometidos a los mismos errores que los convertidores A/D vistos en el Capítulo 3. En la práctica, la resolución está principalmente determinada por los errores de linealidad absoluta y diferencial. El error de linealidad es la desviación entre el nivel de salida real e ideal del D/A, debiendo mantenerse inferior a $\pm 1/2$ LSB. El error de linealidad diferencial es la desviación relativa entre cada nivel de salida y el valor ideal del paso de cuantificación asociado. Si bien para elevados niveles de señal es independiente de la amplitud de la señal, no sucede así en bajos niveles, de forma que en éstos se hace más evidente su dependencia con el valor de la señal, apareciendo entonces como una distorsión de la misma. El error de linealidad diferencial recae en regiones tanto amplias como reducidas de la función de transferencia del convertidor, provocando que se pierdan zonas enteras de ésta. El error suele ser despreciable en los niveles altos, pero en las señales de bajo nivel puede ser enormemente pernicioso. Veámoslo con un ejemplo: una señal de -80 dBFS en un convertidor de 16 bits utiliza tan sólo seis o siete códigos diferentes, o dicho de otra forma, emplea como mucho los tres bits menos significativos para su representación. Suponiendo que de ellos fueran erróneos menos de la mitad de los códigos, significaría

que un cambio en los dos bits de menor peso no provoca valores diferentes de tensión en la salida, con lo que sería análogo a trabajar con un convertidor de 14 bits. Para una señal sinusoidal de -90 dBFS, dependiendo del nivel medio sobre el que aparezca superpuesta la señal, el error de linealidad diferencial en un convertidor D/A de 16 bits puede provocar que se generen cambios de nivel de $-85,9$ dB a $-98,2$ dB. Debido a que los bits y sus errores asociados se generan en un punto determinado de la función de transferencia, el error depende entonces del nivel de la señal. Así, la distorsión armónica y de intermodulación generadas varían con la señal; y que este tipo de error está correlado con la señal, es aún más fácil de percibir. El error de monotonicidad es un caso extremo de falta de linealidad. Consiste en que un incremento en el código de la señal no produce un incremento en la tensión de salida, por lo que la mayoría de los convertidores garantizan el estar libres de él.

Una prueba de linealidad mide la capacidad de un convertidor de grabar y reproducir varias señales con la amplitud apropiada. Específicamente, la linealidad mide la capacidad del convertidor para entregar una señal analógica cuyo nivel coincida con el de la palabra digital asociada. Por ejemplo: cuando un bit cambia de 1 a 0 en un convertidor D/A, la señal de salida debe decrecer exactamente una cantidad proporcional al peso de ese bit. La falta de linealidad provoca una deformación en la forma de onda de la señal de salida, siendo la magnitud del cambio función de la posición que ocupa el bit dentro de la palabra. Cada palabra digital en PCM está formada por una serie de dígitos binarios, dispuestos en potencias de dos. El bit más significativo («most significant bit», MSB) tiene el peso asociado a un cambio de amplitud de la mitad del valor máximo de salida, mientras que al bit menos significativo («least significant bit», LSB) le corresponde el menor cambio de amplitud posible, lo que representa en una palabra de 18 bits (con $2^{18} = 262.144$ niveles posibles) menos de una cuarta parte por millón. Tal precisión sólo existe en teoría, ya que los tradicionales convertidores en escalera poseen una alinealidad superior debido al ruido térmico, envejecimiento y variaciones internas de temperatura.

Muchos convertidores D/A tienen la posibilidad de ser calibrados. Considerar un convertidor D/A de 16 bits con la posibilidad de calibrar su MSB. Como el peso del MSB es muy superior al del resto de bits, un error del MSB de tan sólo el 0,01 por 100 corresponde a un error de una parte por 10.000; con lo que podría perfectamente enmascarar la contribución de los dos bits menos significativos (que correspondería a un error de una parte cada 21.845 de la amplitud total). De la misma forma, un error del 0,1 por 100 del MSB podría enmascarar la acción de los cinco bits menos significativos, por lo que para que el sistema funcionase de forma ideal, la precisión respecto al MSB debería ser mejor que una parte cada 65.536. Algunos convertidores poseen un mecanismo de calibración de los cuatro bits más significativos, siendo interesante destacar que el 93 por 100 de las señales analógicas de salida son representadas mediante estos cuatro bits. Estos cuatro bits son, naturalmente, los que mayor peso aportan en la señal de salida; y cuando están calibrados correctamente en un D/A, la globalidad de los niveles de salida se benefician de esta mayor precisión. Esta precisión es más significativa en los niveles bajos de señal, habitualmente para valores inferiores a -60 dB; y como se ha comentado, cualquier error aparece como una desviación de la amplitud de la muestra sobre su valor original. Y lo peor es que esta falta de linealidad es percibida en la escucha; por ejemplo, es fácil apreciar la pobre linealidad de un D/A reproduciendo un disco de prueba con un tono (con «dither») cuyo nivel se atenúa lentamente.

Las prestaciones de un convertidor D/A a bajo nivel deben ser evaluadas mediante pruebas específicas de linealidad a bajo nivel. Por ejemplo, la Figura 4.2 muestra la medida de la linealidad a bajo nivel de un convertidor D/A. Se aprecia cómo las señales inferiores a -100 dB muestran una clara falta de linealidad. Por ejemplo, una señal de -100 dB es reproducida con una amplitud de -99 dB, y una señal de -110 dB, con un nivel aproximado de -105 dB. Dependiendo de la propia señal, este tipo de error produce modificaciones en la dinámica de las señales de pequeño nivel que pueden llegar a ser audibles.

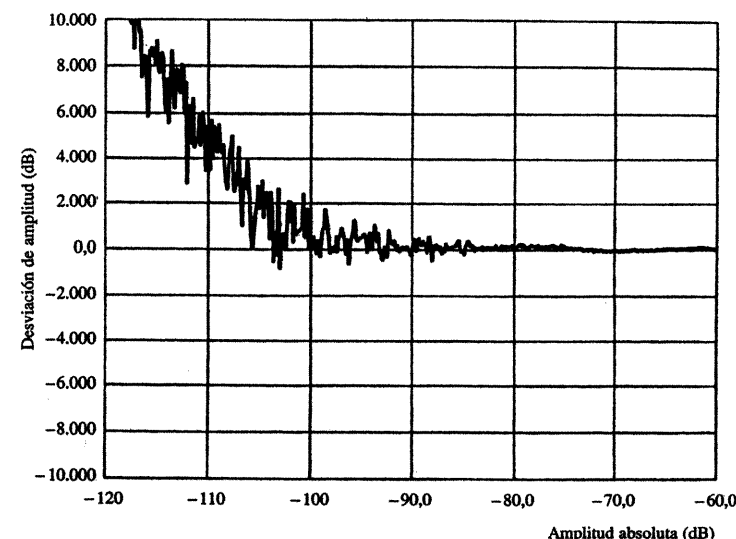


Figura 4.2. Un ejemplo de la medida de linealidad a bajo nivel en un convertidor D/A, mostrando el incremento de alinealidad a medida que disminuye la amplitud.

En la práctica, una conversión lineal de 16 bits es insuficiente para una palabra dato de 16 bits, de forma que los convertidores deben tener un mayor rango dinámico que la señal de audio en sí misma. Por muchos bits que tenga que convertir, la precisión en la conversión y por supuesto la fidelidad de la señal de audio dependen de la linealidad del convertidor. Los errores de linealidad en los convertidores tienen generalmente una desviación estocástica, variando de forma aleatoria de una muestra a otra. Sin embargo, el error relativo siempre se incrementa al disminuir el nivel de la señal. En resumen: la precisión de la conversión D/A está dada por el error de linealidad, y no por el número de bits del convertidor.

El tiempo de asentamiento es otro de los parámetros importantes de un convertidor. Es el intervalo temporal que transcurre desde que se produce la entrada de un nuevo código al convertidor y que la salida alcanza un valor constante con una tolerancia prefijada (que suele ser de $\pm 1/2$ LSB). El tiempo de asentamiento puede variar con la magnitud del cambio en la palabra de entrada.

La mayoría de los convertidores trabajan con una entrada en complemento a dos. Por ejemplo: en un convertidor D/A de 8 bits, el mayor valor positivo se representa por 01111111, mientras que la mayor amplitud negativa se codifica con 10000000, actuando en este formato el MSB como bit de signo. Como el complemento a dos es el formato habitual para los datos de audio, la mayoría de los convertidores D/A tienen una entrada de datos con este formato, pues de lo contrario se debe realizar la decodificación previa a binario natural.

En la medida de convertidores D/A se suelen utilizar tonos puros generados digitalmente, aunque es importante elegir frecuencias no correladas con la frecuencia de muestreo, ya que, de ser así, se reproducirán una serie de códigos que no corresponderán a los niveles fondo de escala elegidos. Dependiendo de la linealidad del convertidor en estos códigos particulares, el resultado de la medida de la distorsión de salida es diferente del verdadero valor de la misma, pudiendo obtenerse tanto valores mejores como peores a los reales. Por ejemplo, cuando se

reproduce un fragmento de un segundo, de un tono de 1 kHz a 0 dB y muestreado a 44.100 Hz, sólo se utilizan 441 códigos diferentes para representar las 44.100 muestras de señal. Un tono de 997 Hz y del mismo nivel puede usar hasta 20.542 códigos distintos, dando una mejor visión de las características del convertidor. Los tonos estándar de medida han sido elegidos específicamente para evitar dicha anomalía, siendo algunas de las frecuencias: 17, 31, 61, 127, 251, 499, 997, 1.999, 4.001, 7.993, 10.007, 12.503, 16.001, 17.989 y 19.997 Hz.

Convertidor D/A de resistencias ponderadas

Para la digitalización de audio se utilizan convertidores de varios tipos, aunque los principios de funcionamiento de convertidores tradicionales y los de sigma-delta son completamente diferentes. Los primeros son de fácil comprensión, por lo que comenzaremos con los convertidores D/A del tipo de resistencias en escalera. Recordemos: un convertidor digital-analógico debe tomar la palabra de n bits aplicada en su entrada y entregar una tensión (o corriente) de salida proporcional al código de la palabra aplicada. El tipo más simple de convertidor D/A es el denominado de resistencias ponderadas. El circuito está formado por una serie de resistencias e interruptores, dispuestos como se muestran en la Figura 4.3, para el caso de un convertidor de 8 bits. Consta básicamente de tres elementos:

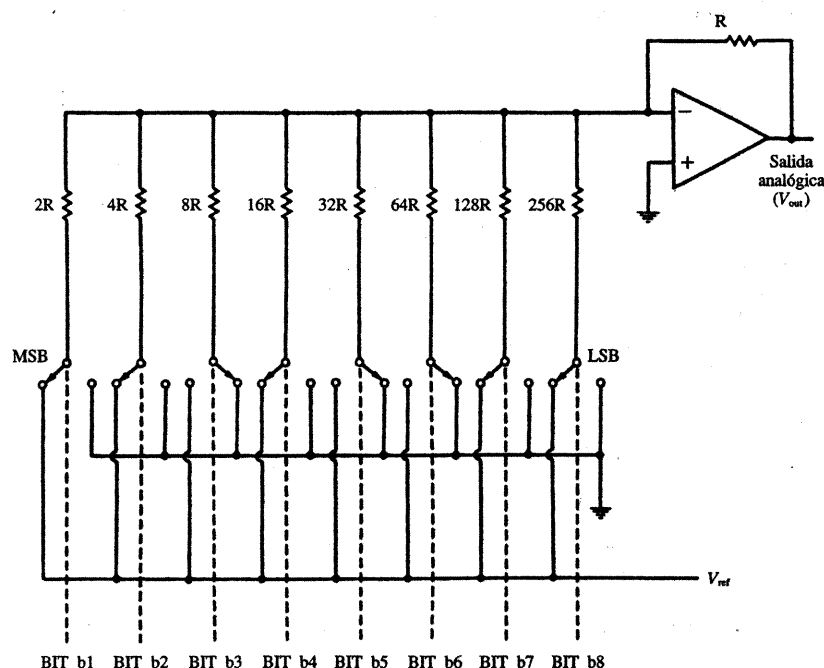


Figura 4.3. Un convertidor D/A de resistencias ponderadas utiliza resistores cuyo valor crece en potencias de dos, lo cual limita la resolución.

- Por cada bit, un interruptor y una resistencia cuyo valor representa el peso del bit¹ en cuestión.
- Una tensión de referencia (V_{ref}), encargada de generar la corriente que atravesará las resistencias cuyo interruptor esté cerrado. Su magnitud permite establecer la tensión de salida máxima, asociada al fondo de escala.
- Un amplificador operacional, que suma todas las corrientes y las convierte en la tensión de salida.

Un bit a 1 cierra el interruptor correspondiente, contribuyendo a la circulación de una corriente proporcional al valor de la resistencia introducida, mientras que un bit a 0 mantiene abierto el interruptor, evitando la circulación de corriente por esa rama. Una palabra binaria de bajo valor tiene muchos ceros; con lo que cerrando pocos interruptores el voltaje resultante de salida será también bajo. Una palabra digital de alto valor posee muchos unos, con lo que al cerrar muchos interruptores proporciona una elevada tensión de salida. Por ejemplo, en el convertidor de la figura, la expresión de la tensión de salida en función de la palabra digital aplicada es:

$$V_{out} = -V_{ref} \left(\frac{b1}{2} + \frac{b2}{4} + \frac{b3}{8} + \frac{b4}{16} + \frac{b5}{32} + \frac{b6}{64} + \frac{b7}{128} + \frac{b8}{256} \right)$$

donde los $b1$ representan los bits de la palabra digital en la entrada ($b1$ el MSB y $b8$ el LSB, respectivamente). Por ejemplo, con una referencia de 10 V y la palabra 11010011, la tensión de salida es:

$$\begin{aligned} V_{out} &= -10 \left(\frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{1}{128} + \frac{1}{256} \right) \\ &= -10 \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{128} + \frac{1}{256} \right) \\ &= -8,24 \text{ V} \end{aligned}$$

Aunque este circuito trabaja muy bien sobre el papel, en la práctica casi no se usa. Esto se debe a la dificultad de integrar resistencias de elevada precisión y con valores tan dispares (obsérvese que sus valores crecen en potencias de 2). Por poner un ejemplo, en un convertidor de 16 bits, la relación entre la resistencia mayor y menor sería de 65.536 a 1, con lo que si la más pequeña fuera de 1 kΩ, la mayor superaría los 65 MΩ. La misma disparidad aparece con las corrientes que se establecen en el circuito, y si la mayor fuera de 2 mA la menor escasamente superaría los 30 nA. En definitiva, el circuito exige unas condiciones de fabricación que difícilmente pueden ser satisfechas.

Convertidor D/A con red en escalera R-2R

Un circuito de conversión D/A más adecuado es el que utiliza una red en escalera R-2R, tal y como se muestra en la Figura 4.4. Este circuito también está formado por resistencias y conmutadores, aunque ahora con una disposición muy diferente y con dos resistencias por bit. Cada interruptor contribuye con un peso determinado a la salida: cuando el bit es 1 conmuta la resis-

¹ Estrictamente hablando, el valor de cada resistencia se elige para que la corriente que aporte corresponda con el peso binario del bit. En el ejemplo, el LSB (bit $b8$) aportaría 1/256 avo de la corriente total, el $b7$ 1/128 avo, el $b6$ 1/64 avo...

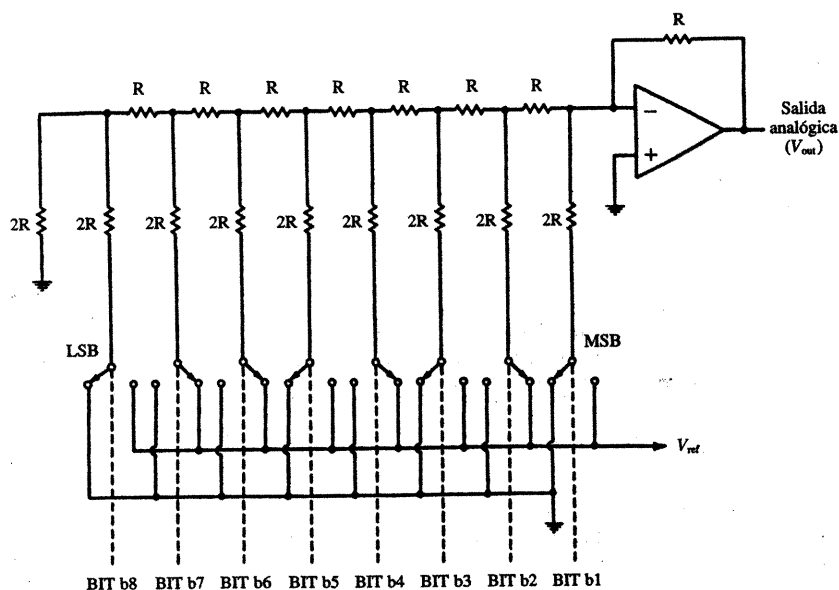


Figura 4.4. Un convertidor D/A R-2R utiliza sólo resistores de dos valores, mejorando la resolución.

tencia a V_{ref} y cuando es cero la dispone a masa. La corriente se divide en cada nudo de la red, resultando una corriente que fluye a través de las resistencias y que es ponderada en potencias de dos. Si de la referencia de tensión sale una corriente I , la mitad ($I/2$) fluye a través del primer interruptor, la mitad de la mitad ($I/4$) a través de la segunda rama, la mitad de la mitad de la mitad ($I/8$) por la tercera, y así sucesivamente, dividiendo por dos la corriente en cada nodo. La tensión de salida es de la forma:

$$V_{out} = -V_{ref} \left(\frac{b1}{2} + \frac{b2}{4} + \frac{b3}{8} + \frac{b4}{16} + \frac{b5}{32} + \frac{b6}{64} + \frac{b7}{128} + \frac{b8}{256} \right)$$

con lo que para $V_{ref} = 10$ V y una señal de entrada 01010110 (la mostrada con la disposición de interruptores de la figura) la tensión de salida sería:

$$V_{out} = -10 \left(\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{128} \right) = -3,36$$

La red R-2R es uno de los circuitos preferidos, ya que puede ser implementada de forma sencilla y precisa, necesitando únicamente resistencias de dos valores diferentes. En algunos diseños, la obtención de la adecuada estabilidad con respecto a la temperatura se consigue mediante un lazo de realimentación. Como se comentó anteriormente, en muchos convertidores uno o varios bits de los más significativos pueden ser calibrados para mejorar la linealidad. El proceso consiste en generar una señal (del nivel del MSB) de elevada precisión y compararla

con la salida del convertidor. La diferencia entre las dos señales (error cometido en la conversión de ese código) se almacena en una memoria, de forma que aplicándose como palabra de corrección a la entrada del D/A hace que el error de salida para ese código sea mínimo. Así, los errores provocados por las derivas de los componentes son autocorregidos, minimizando la distorsión generada.

Distorsión de cruce

Un convertidor digital-analógico es un componente de precisión. Un D/A de 16 bits con fondo de escala de ± 10 V tiene unos pasos de cuantificación (diferencia entre dos códigos consecutivos) de $20/65.536 = 0,000305$ V. Esto significa que la tensión de salida de la palabra 1000 0000 0000 0000 es sólo 0,3 mV mayor que la del código anterior 0111 1111 1111 1111, lo que visto de otro modo significa que el conjunto de los 15 bits de menor peso deben ser tan exactos como los 0,3 mV que le separan del MSB. Eso significa que tanto estos 15 bits como aisladamente el MSB deben tener un error menor que la mitad de un paso de cuantificación. Sin embargo, la no linealidad diferencial es mayor para el MSB, que además produce el mayor nivel de salida. A esto se añade que el MSB cambia cada vez que la señal pasa por cero, con lo que es realmente difícil mantener la precisión en el centro del rango de salida del D/A. Estos errores en los pasos por cero provocan lo que se denomina distorsión de cruce.

La distorsión de cruce aparece en los pasos por cero de la señal, o dicho de otra forma, siempre que cambia su polaridad. Cuando en un convertidor D/A de resistencias en escalera se conmuta el MSB pasando de 1000 0000 0000 0000 a 0111 1111 1111 1111 con vistas a reflejar un cambio de polaridad, todos los conmutadores internos deben cambiar de estado. Las fluctuaciones instantáneas de la corriente, y los diferentes retardos de conmutación de los bits, provocan dos efectos negativos (véase Fig. 4.5): primero elevar la alinealidad diferencial y segundo una fluctuación rápida e imprevisible de la tensión de salida («glitch») hasta que ésta adquiere un valor estable. A estos defectos se les denomina conjuntamente distorsión de cruce. Como

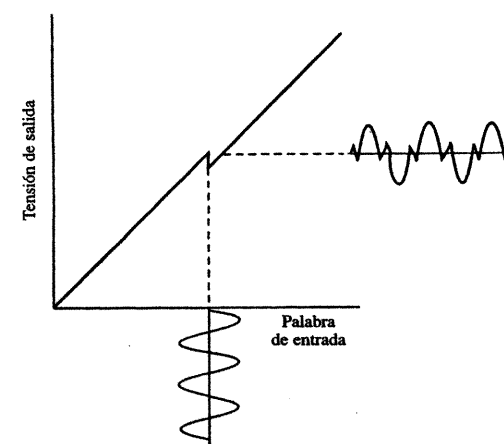


Figura 4.5. La distorsión de cruce aparece en los pasos por cero. Cuando la palabra de entrada del convertidor D/A cambia de polaridad, aparece un «glitch» (transitorio aleatorio de bajo nivel) en la forma de onda de salida.

la señal musical está incesantemente cambiando de polaridad, los cruces por cero son continuos, así como la conmutación del MSB. El error es particularmente problemático en la reproducción de señales de bajo nivel, ya que entonces la amplitud de los «glitches» es comparativamente grande con respecto a la señal. Además, la audibilidad de la distorsión es elevada por el «dither», ya que éste incrementa los pasos por cero que aparecen por período de señal.

Idealmente, cuando no hay diferencia de valor entre las resistencias de la red en escalera, el error es nulo. Sin embargo, en el caso real, el MSB introduce un error generalmente más elevado que el LSB, pudiendo este error superar con creces el valor del LSB. El motivo es la dificultad de generar un MSB mediante una corriente, con una precisión superior a 1/65.536 de la empleada para producir un LSB. Esto demuestra que el ajuste del MSB en los convertidores es de suma importancia para minimizar el error de linealidad. Algo similar ocurre cuando se conmutan los bits segundo, tercero, cuarto... El error decrece proporcionalmente a medida que el nivel de la señal aumenta, y viceversa; en niveles bajos de señal, el error se vuelve comparativamente más elevado. Así, a medida que la longitud de la palabra aumenta, también lo hace la cantidad de resistencias y la precisión requerida, con lo que las dificultades para su implementación real crecen exponencialmente.

La distorsión de cruce puede ser rebajada mediante una cuidadosa calibración de los bits de mayor peso del convertidor, aunque en ocasiones se buscan modelos de conversión alternativos. Un ejemplo: se pueden usar configuraciones de conversión separando magnitud (amplitud) y signo (polaridad), utilizando una etapa de salida que entregue la señal con polaridad positiva o negativa, en función del bit de signo de la palabra código de partida. Ahora, el paso por cero se realiza alrededor del LSB (mínimo valor implicado en torno al cambio de signo), con lo que la distorsión (y la amplitud relativa de los «glitches») disminuye notablemente. Otra alternativa consiste en usar dos convertidores D/A complementarios en paralelo, uno para polaridad positiva y otro para la negativa. De esta forma, nunca aparecen conmutaciones de la red de resistencias en los pasos por cero, con lo que se mejora notablemente el comportamiento del conjunto. Por otro lado, en el Capítulo 18 estudiaremos los convertidores sigma-delta, configuración que no presenta este tipo de problemas.

Conversión de alto número de bits

Algunos fabricantes han introducido convertidores de 18, 20 o incluso más bits en sus sistemas de reproducción PCM de 16 bits, con el fin de incrementar el grado de fidelidad de las grabaciones de 16 bits. La razón de esto se debe a los defectos inherentes a los convertidores D/A. Excepto en teoría, los convertidores de 16 bits no pueden decodificar la información de 16 bits sin introducir un cierto grado de error. Por ejemplo, si de una palabra de 16 bits se obtienen 18 bits, y se convierten mediante un D/A con resolución de 18, los errores se reducen y las especificaciones de la cadena de reproducción mejoran. Para poder llegar a obtener el máximo de fidelidad en la reproducción de la señal de audio, la digitalización y todos los pasos de tratamiento de la señal deben efectuarse con un margen dinámico mayor que el que corresponde a la grabación final.

La elección de una longitud de palabra de 16 bits para el CD, DAT y otros formatos de almacenamiento fue efectuada por la buena disponibilidad de convertidores D/A de 16 bits, a lo que se sumó el hecho de que la utilización de mayores longitudes acortaba el tiempo de grabación de los soportes. Sin embargo, los convertidores de 18 bits, por ejemplo, pueden aportar una conversión mejor de la señal almacenada en 16. Cuando el proceso se efectúa correctamente, una conversión con 18 bits mejora la resolución en amplitud, asegurando una conversión completamente lineal de la señal de 16 bits. Un convertidor D/A de 18 bits tiene 262.144 niveles de salida, exactamente cuatro veces más que uno de 16 bits. En consecuencia, cualquier falta de linealidad es cuatro veces más pequeña, por lo que el incremento en la longitud de la pala-

bra en la etapa de conversión resulta en una mejora de la relación S/R, disminuyendo también la amplitud de los chasquidos producidos en la cuantificación. En resumen, un convertidor de 18 bits realiza una mejor conversión de las palabras de 16 bits. De hecho, no necesitaríamos ni conectar los dos bits extra en un convertidor de 18 bits para ya mejorar las prestaciones en la conversión de una señal de 16 bits.

El uso de convertidores D/A con elevado número de bits puede ser comparado al sobremuestreo: a medida que la frecuencia de muestreo aumenta, no es que creemos nueva información, es que hacemos un mejor uso de la existente. El sobremuestreo brinda la oportunidad de trabajar con convertidores de elevado número de bits en la conversión de datos que originalmente eran de 16 bits. Cuando sobremuestreamos una señal de 44,1 kHz y 16 bits, ambas magnitudes se incrementan. La frecuencia, por el propio hecho del sobremuestreo, y la longitud de la palabra, debido a las multiplicaciones que sufre por los coeficientes del filtro digital. Éste debe ser adecuadamente diseñado para que la palabra de salida contenga la información útil en los 16 bits de menor nivel.

Lo que no es significativo es, como indicativo de las prestaciones de los convertidores D/A y A/D, tomar la longitud de palabra. La relación señal ruido nos da la relación entre la máxima señal y el ruido sin señal. Debido a que los equipos enmudecen la salida con señal nula, todos los errores de bajo nivel (como la distorsión de cruce) se eliminan. Otra forma es medir el rango dinámico, evaluando la relación entre la máxima señal y ruido de banda ancha (0-20 kHz) con un nivel de -60 dB. Así, esta medida nos aporta información de la distorsión que introduce el equipo a bajos niveles. A partir de esta medida del rango dinámico se puede calcular el número efectivo de bits (NEB) con el que trabaja el equipo:

$$NEB = \frac{\text{Rango dinámico} - 1,72}{6,02}$$

Por ejemplo, un convertidor de 16 bits con un rango dinámico de 90 dB tiene 14,7 bits de resolución, con lo que 1,3 bits se han perdido entre distorsión y ruido.

En muchos diseños se utilizan convertidores D/A sigma-delta, con la ventaja de minimizar los problemas inherentes a los diseños tradicionales. Los sistemas basados en sigma-delta necesitan frecuencias de sobremuestreo muy elevadas, conformación del ruido y conversor multibit o incluso de un solo bit. Para representar la forma de onda de la señal de audio, un verdadero sistema de un solo bit necesita tener como salida una forma de onda digital a alta velocidad. En otros sistemas multibit, la salida es un pulso multinivel, o representa la señal de audio mediante pulsos de anchura variable (modulación por anchura de pulsos, «PWM»). Pero en todos, debido al elevado nivel de ruido de los sistemas sigma-delta, es imprescindible la utilización de algoritmos para la conformación del ruido. Este tipo de conversión se verá en detalle en el Capítulo 18.

Aunque los convertidores sigma-delta son utilizados ampliamente, los tradicionales ofrecen algunas ventajas. Puesto que no emplean conformador de ruido, no producen ruido fuera de la banda. Además, con señales de muy bajo nivel la mayoría de los interruptores (de la red de resistencias) no se conmutan, con lo que el ruido y los «glitches» introducidos son bastante bajos. Esto hace que los convertidores tradicionales sigan siendo atractivos para bastantes aplicaciones, sobre todo cuando dos o más son usados en paralelo.

CIRCUITO DE MUESTREO Y RETENCIÓN DE LA SALIDA

Algunos equipos de audio incluyen dos etapas de muestreo y retención: una en la entrada para tomar en un instante la amplitud de la señal de entrada y mantenerla mientras el A/D realiza la conversión, y otra en la salida del D/A tomando y manteniendo la salida del D/A para eliminar

los transitorios de la conmutación («glitches»). Debido a este muestreo, se produce una pérdida en la respuesta en frecuencia denominada error de apertura, por lo que al muestreador que la produce a veces también se le llama circuito de apertura.

La mayoría de los convertidores D/A, en la transiciones entre palabras generan señales espúreas o «glitches», las cuales se superponen a la señal de audio. Cuando una palabra digital se aplica a la entrada de un D/A, no todos los conmutadores responden instantáneamente al unísono, sino que necesitan un tiempo pequeño, pero finito, para disponerse en su nueva posición. A este tiempo que transcurre desde que se aplica un cambio de palabra en la entrada hasta que la señal de salida se ha estabilizado en su valor correcto, se le denomina tiempo de asentamiento. Durante ese intervalo hay continuas redistribuciones de corriente, provocando estados intermedios que no se corresponden con la palabra de entrada, pero que sí producen una salida de señal errónea. Por ejemplo, en la transición de la palabra 01111111 a 10000000, suponemos que el conmutador del MSB cambia a 1 antes que el resto. Esto se traduce en que internamente y de forma instantánea se provoca una salida de señal correspondiendo al código 11111111, generando un transitorio de valor muy diferente al de las palabras de entrada («glitch»). Aunque el tiempo de asentamiento es muy corto, los «glitches» siempre existen; y si permitimos que salgan de la cadena de conversión, se manifestarán como distorsión de la señal original.

Para eliminar estos «glitches» utilizamos una etapa de muestreo y retención («S/H») a la salida del D/A, al cual se le suele denominar «deglitcher». Su cometido es tomar una muestra de la salida del convertidor sólo cuando éste ha alcanzado un valor de salida estable (muestreo), manteniendo dicha señal en su salida (retención) en las transiciones entre palabras (cuando aparecen los «glitches»). Este proceso asegura una salida libre de «glitches» (véase Fig. 4.6), pero convierte la salida del D/A (serie de niveles analógicos) en una serie de pulsos con amplitud igual a la de la muestra original.

Desde el punto de vista circuital, el diseño de este muestreador en nada difiere del usado en la etapa de entrada. Incluso, en algunas especificaciones como la caída de tensión («droop») el circuito de salida requiere de menor precisión. La razón es que cualquier caída se traduce en un

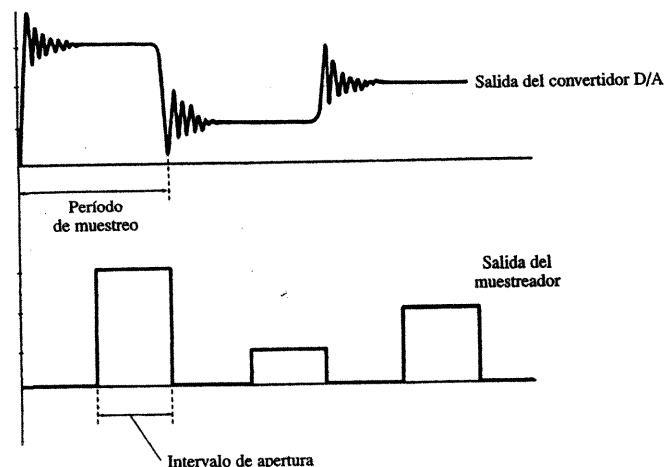


Figura 4.6. Un circuito de muestreo en la salida es el encargado de eliminar los «glitches» producidos por el convertidor D/A.

desplazamiento constante de los niveles de salida, pudiendo ser eliminado fácilmente en etapas posteriores. En otros aspectos, este muestreador debe ser cuidadosamente diseñado e implementado, ya que en él requieren especial atención parámetros como el tiempo de retención y la velocidad de paso de muestreo a retención.

El circuito de muestreo y retención de salida es el responsable del error de apertura, atenuación que sufren las altas frecuencias de señal de audio, debido a que la duración del pulso de salida (tiempo de apertura) es más corta que el período de muestreo. En nuestro caso, cuanto más estrecho es el tiempo de apertura menor es el error de apertura. Si en la etapa de conversión A/D utilizáramos un muestreo ideal (instantáneo) de la señal, la salida del convertidor D/A debería ser un tren de impulsos reproduciendo los instantes originales de muestreo, por lo que no existiría atenuación en alta frecuencia. Sin embargo, no es posible utilizar un convertidor D/A ideal. La forma de onda a escalones típica de la salida de un convertidor D/A ideal está compuesta por pulsos de duración igual al período de muestreo (matemáticamente hablando, la salida corresponde a la convolución de las muestras originales, con un pulso de duración un período de muestreo).

El espectro de una serie de pulsos de anchura finita introduce una atenuación en altas frecuencias. Esto le diferencia de la respuesta plana, obtenida del muestreo ideal con impulsos (pulsos de duración infinitesimalmente pequeña); y por tanto, a la atenuación introducida en las altas frecuencias de audio, ya que es un error de amplitud, se le denomina error de apertura. Matemáticamente, a la atenuación introducida se la conoce como función de muestreo, y es de la forma $|\text{sinc}(x)|$. Cuando la anchura del pulso es igual al período de muestreo, la respuesta en frecuencia se hace cero en los múltiplos de la frecuencia de muestreo, tal y como se muestra en la Figura 4.7A. La atenuación introducida no sólo afecta a las altas frecuencias de la banda base, sino que afecta también a los espectros imagen creados en el proceso de muestreo.

A la mitad de la frecuencia de muestreo, el valor de la función es 0,64 y por tanto se introduce una atenuación de unos 4 dB. Esto puede ser corregido mediante el muestreador de la salida, convirtiendo la salida ideal del D/A en un tren de pulsos, haciendo cada vez más pequeño el tiempo de apertura en el muestreador. La teoría del muestreo demuestra que el ancho de banda de la respuesta está determinado por la anchura del pulso. Acortando la duración del pulso aumentamos proporcionalmente el ancho de banda. De forma específica, si reducimos la anchura del pulso a la mitad (disminuyendo el tiempo de apertura) con respecto al caso 4.7A, la atenuación introducida a la mitad de la frecuencia de muestreo también disminuirá (Fig. 4.7B). En el caso de un tiempo de apertura que sea una cuarta parte del período de muestreo, la amplitud a la frecuencia de Nyquist pasa a ser 0,97 del total; o lo que es lo mismo, sólo se atenúa en 0,2 dB. Éste es el caso que se toma como óptimo ($\tau = T_m/4$), ya que tiempos de apertura menores degradan considerablemente la relación señal-ruido de la salida.

Otra solución para el error de apertura consiste en compensar la atenuación que introduce en una etapa previa a la conversión D/A, siendo el filtro recuperador digital una etapa habitual para realizar esta corrección. Consiste en realzar las altas frecuencias la misma magnitud que las atenúa el error de apertura, con lo que el resultado completo es una respuesta en frecuencia plana, sin producir además degradación de la relación señal-ruido. El realce también puede ser introducido en cualquier etapa restante de la cadena, el filtro recuperador de salida o bien como pre-énfasis en la etapa de entrada. El único requisito que siempre se debe cumplir es que el realce sea complementario a la atenuación introducida por el error, forma de asegurar una respuesta global plana.

Hemos visto que el uso principal del muestreador es eliminar los errores de conmutación producidos por el convertidor D/A en la transición entre palabras; sin embargo, el muestreador debe evitar introducir errores como resultado de su propia conmutación. Un circuito ideal de muestreo y retención toma en un instante la amplitud de la señal (muestreo o adquisición) y mantiene su valor en la salida mientras está en el estado de retención («hold»). En los circuitos reales, cuando cambia de muestreo a retención, la salida no adquiere instantáneamente el valor

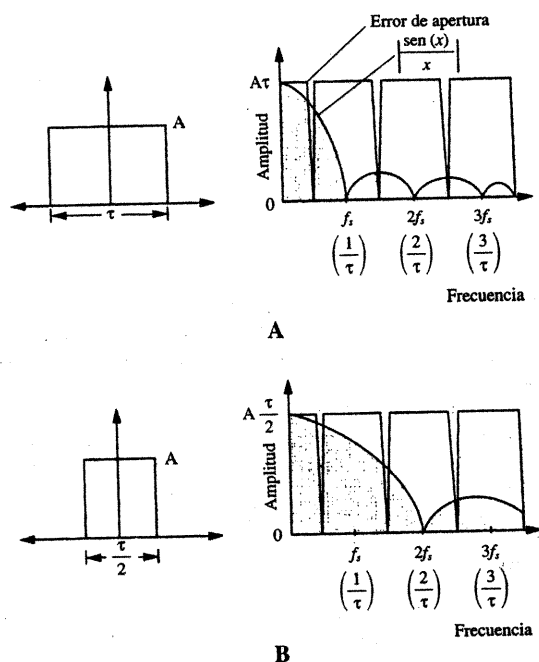


Figura 4.7. El error de apertura puede ser minimizado disminuyendo la anchura del pulso de salida. A. Un pulso de anchura igual a un período de muestreo produce una atenuación en alta frecuencia dentro de la banda de audio. B. La respuesta en alta frecuencia de la banda base mejora cuando la anchura del pulso es la mitad del período de la frecuencia de muestreo.

esperado, sino que introduce unos valores intermedios que nada tienen que ver con la amplitud del pulso al que debe representar. Este problema no tiene ninguna repercusión en el muestreo de la etapa de entrada, ya que el convertidor A/D realiza sus operaciones dentro del modo de mantenimiento, ignorando por completo las transiciones. No ocurre lo mismo en el muestreador de salida, ya que lo que éste entrega es una señal continua (en el tiempo, naturalmente) al filtro recuperador, y todos los errores que aparezcan, aunque sea en las transiciones, formarán parte de la señal de salida final. O dicho de otro modo, no sólo son importantes los niveles de señal, sino el comportamiento del muestreador en las transiciones entre palabras.

La distorsión es mayor a altas frecuencias debido a la gran diferencia que existe entre las amplitudes de muestras consecutivas. Una señal de 20 Hz no cambia apreciablemente en el intervalo entre dos muestras; sin embargo, en una señal de 20 kHz de alto nivel el cambio puede cubrir todo el rango de amplitudes del convertidor. Aunque los productos de distorsión son eliminados por el filtro recuperador, dentro de la cadena digital pueden producir batidos con la frecuencia de muestreo, generando productos de distorsión que caigan dentro de la banda base. Para minimizar este problema, el muestreador de la salida debe conmutar de muestreo a retención lo más rápidamente posible, ya que idealmente deberían aparecer pulsos rectangulares.

En teoría, esto eliminaría la posibilidad de que aparezca distorsión como consecuencia de las transiciones. En la práctica, Barry Blesser ha establecido que el slew-rate mínimo debería

situarse en torno a los 5 V/ns (voltios por nanosegundo), velocidad que no es posible alcanzar dentro de esta circuitería, con lo que debemos introducir una modificación adicional a esta etapa de salida. Un cambio exponencial de la amplitud de la señal, en el intervalo de un paso de cuantificación al siguiente, no afecta a la linealidad de la propia señal; y si a continuación viene el filtrado de salida, estas transiciones exponenciales acabarán como una respuesta lineal. Puede demostrarse que una transición exponencial de una muestra a la siguiente causa sólo una pequeña atenuación en altas frecuencias, pero no distorsión ni pérdida de linealidad. Un circuito de muestreo y retención que integre la diferencia entre el valor actual y el de la muestra siguiente valdría para producir una transición exponencial. La atenuación en altas frecuencias producida por un circuito integración y mantenimiento es menor que la producida por el proceso de muestreo y retención en sí mismo, con la ventaja añadida de poder ser ecualizado.

Así, el circuito de muestreo y retención de salida elimina los «glitches» de la señal de salida entregada por el convertidor D/A. Para que el error de apertura sea mínimo, el tiempo de mantenimiento debe ser inferior al período de muestreo. Muchos convertidores D/A son diseñados específicamente para evitar estos «glitches» en su salida, y por tanto son capaces de trabajar sin el muestreador de la salida, con lo que el error de apertura es compensado en la etapa de filtrado digital. En otros casos, la etapa del muestreador aparece incluida en el propio circuito integrado del convertidor. Sea cual fuere el método usado, la señal escalonada de la salida del D/A está preparada para el filtrado de salida (filtro recuperador) y la recuperación de la forma de onda original.

FILTRO RECUPERADOR

Los circuitos primero y último en una cadena de digitalización de audio son el filtro antialiasing, o filtro paso bajo de entrada, y el filtro recuperador, o filtro paso bajo de salida. Aunque su diseño desde el punto de vista analógico es casi idéntico, sus funciones son completamente distintas. Hoy en día, en lugar de usar los tradicionales filtros recuperadores analógicos de elevado orden, empleamos filtros digitales con técnicas de sobremuestreo.

Dado el criterio de Nyquist para la elección de la frecuencia de muestreo, la función del filtro antialiasing es muy clara: debe eliminar todas las señales de frecuencia superior a la mitad de la frecuencia de muestreo, para evitar precisamente este efecto de aliasing. De forma similar, el filtro paso bajo de la salida debe eliminar todo el contenido de señal de frecuencia superior a la mitad de la frecuencia de muestreo. De forma intuitiva, este filtro debe convertir la salida «escalonada» del D/A en la suave envolvente propia de la señal analógica, recuperando la forma de onda original. Matemáticamente, la salida del D/A contiene el espectro original (banda base) y las bandas superiores generadas en el proceso de muestreo, siendo estas últimas las que corresponden a las transiciones bruscas de nivel, propias de la señal escalonada del D/A. El filtro recuperador debe eliminar estas bandas superiores, redondeando o suavizando la señal escalonada hasta convertirla en la forma de onda original (véase Fig. 2.2).

Los criterios de diseño para un filtro recuperador analógico son similares a los del filtro antialiasing. La banda de paso debe mantenerse plana, y a partir de la frecuencia de corte introducir una atenuación elevada. Esto implica que la pendiente debe ser muy abrupta, y como a medida que ésta crece, las variaciones de fase alrededor de la frecuencia de corte se acentúan, puede aparecer lo que se denomina distorsión de fase. Consiste en la falta de linealidad (proporcionalidad) entre la fase y la frecuencia dentro de ese margen de frecuencia, lo que provoca que el retardo introducido por el filtro a todas las frecuencias no sea el mismo. Este hecho puede producir efectos audibles, por lo que se debe subsanar mediante circuitos específicos de corrección de fase. Pueden disponerse en cualquier etapa de la cadena de conversión, tanto en la entrada como en la salida. Su misión es corregir la falta de linealidad de la fase, consiguiendo un retardo constante para todas las frecuencias y, por tanto, unos defectos audibles mínimos. Otra

consideración en lo que se refiere al filtro recuperador es su respuesta transitoria, ya que a diferencia del filtro de entrada, debe procesar una señal con unos cambios de nivel muy acentuados (los escalones de salida del D/A). Otro hecho, poco considerado en la mayoría de los casos, es la aparición en la entrada del filtro de componentes de frecuencia del orden de los megahercios. Los filtros analógicos no están diseñados para trabajar correctamente con señales de tan alta frecuencia, con lo que el resultado puede ser la aparición de dichas señales en la salida, ante la imposibilidad de atenuarlas.

Viendo todo el proceso desde un punto de vista matemático, podemos observar cómo el muestreo crea la necesidad del filtro recuperador. El proceso del muestreo visto en el dominio del tiempo consiste en la multiplicación de la señal de audio (señal muestreada) por un tren de pulsos (señal muestreadora). En el dominio de la frecuencia, el espectro resultante se obtiene mediante la convolución de los espectros de las dos señales (el audio y el tren de pulsos). El resultado es el mantenimiento del espectro de audio original y la aparición de unos espectros adicionales (espectros imagen), que son la repetición del espectro de audio alrededor de los múltiplos de la frecuencia de muestreo. Por ejemplo, una señal de 1 kHz muestreada a 44 kHz provoca la aparición de componentes adicionales a 43, 45, 87, 89 kHz, etc. Aunque la utilización de un circuito de muestreo y retención reduce fuertemente la amplitud de los espectros adicionales, en la salida del muestreador el nivel de estas señales es elevado, creciendo a medida que nos aproximamos a la banda original (véase Fig. 4.7). Para convertir la señal muestreada de nuevo en la información original, los espectros imagen deben ser eliminados, debiendo dejar únicamente la banda base original. Esto se tiene que realizar mediante un filtro paso bajo.

Alguien podría cuestionar la necesidad de filtrar (eliminar) unas señales por encima de la frecuencia de Nyquist, argumentando que superan el límite de audición del oído humano, pero este razonamiento es fácilmente refutable. Si la señal no se filtrara, podríamos escucharla perfectamente, pero los espectros imagen que atravesaran otros equipos podrían provocar efectos imprevisibles. En equipos analógicos, como magnetófonos o transmisores, los osciladores intermodularían con los espectros imagen, pudiendo caer dichas componentes perfectamente dentro de la banda de audio. Los equipos digitales filtran la señal (en su entrada analógica, claro está), pero los filtros antialiasing no están diseñados para eliminar componentes de frecuencias ni tan elevadas ni de nivel tan alto. Por otro lado, un sistema trabajando con frecuencias de muestreo bajas (por ejemplo, 8 kHz) genera espectros imagen que pueden llegar a ser audibles, razón de más para eliminarlos con el filtro recuperador.

RESPUESTA AL IMPULSO

Lógicamente, cuando un impulso es aplicado a un dispositivo como un filtro paso bajo, a la salida del filtro se le denomina respuesta al impulso. Como comentaremos en el Capítulo 17, la respuesta al impulso puede caracterizar completamente un sistema. Un filtro puede ser descrito perfectamente por su respuesta al impulso (en el dominio del tiempo) o por su respuesta en frecuencia, las cuales están relacionadas por la transformada de Fourier. Por otro lado, obsérvese que multiplicar el espectro de la señal de entrada por la función de transferencia del filtro (en el dominio de la frecuencia) es equivalente (en el dominio del tiempo) a la convolución de la señal de entrada con la respuesta al impulso del filtro. La acción del filtrado paso-bajo y la de la respuesta al impulso son mostradas en la Figura 4.8. En el ejemplo, la señal de entrada está formada por dos sinusoides de frecuencias diferentes (A), estando su espectro (B) formado por dos rayas espectrales (f_1 y f_2). Supongamos que queremos eliminar la componente de alta frecuencia mediante un filtro paso-bajo. Lo podemos realizar multiplicando el espectro de entrada por un filtro paso-bajo ideal con la respuesta mostrada en D, con lo que obtenemos en la salida una sola componente espectral (F). La respuesta al impulso del filtro ideal citado tiene la forma $\text{sen}(x)/x$, tal y como se muestra en C; si la señal de entrada en el dominio del tiempo es

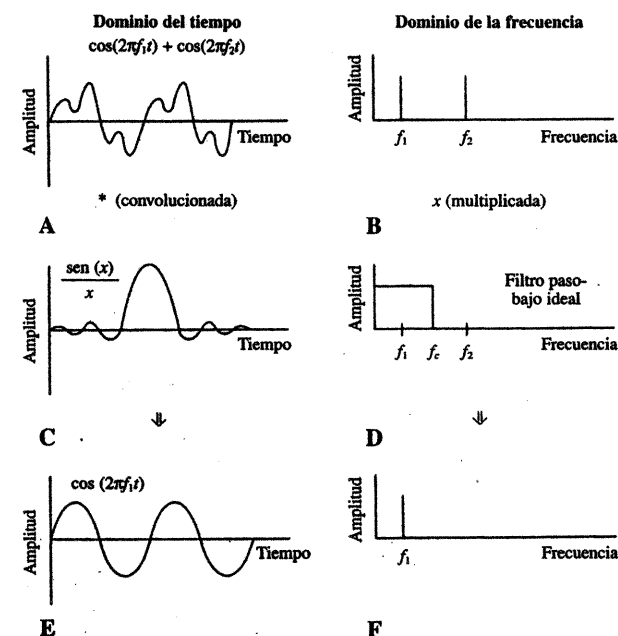


Figura 4.8. Ejemplo de filtrado paso bajo tanto en el dominio del tiempo (columna izquierda) como en el dominio de la frecuencia (columna derecha). A. La señal de entrada comprende dos ondas sinusoidales. B. Espectro de la señal de entrada. C. La respuesta al impulso del filtro paso bajo es una función $\text{sen}(x)/x$. D. Función de transferencia del filtro paso bajo deseado. E. La salida filtrada de la señal es la convolución de la señal de entrada con la respuesta al impulso. F. El espectro de la señal de salida filtrada es la multiplicación de la señal de entrada por la función de transferencia del filtro.

convolucionada con la función $\text{sen}(x)/x$, el resultado en el dominio del tiempo es la salida filtrada mostrada en E. En otras palabras, en el dominio del tiempo, la señal muestreada puede ser filtrada aplicándole la respuesta al impulso que describe las características del filtro. En sistemas digitales, tanto la señal como la respuesta al impulso son representadas por valores discretos, siendo la respuesta al impulso de un filtro paso-bajo ideal cero en todas las muestras excepto en una, en el punto central.

Veamos la función del filtro recuperador. Aunque en esencia la idea de «redondear» la salida del muestreador para eliminar los espectros imagen es correcta, un estudio analítico muestra exactamente cómo la reconstrucción de las muestras es efectuada mediante el filtro paso-bajo recuperador. El teorema del muestreo impone que un filtro paso-bajo ideal es el necesario para reconstruir exactamente la forma de onda de la señal a partir de sus muestras. Nótese que un filtro ideal tiene una respuesta al impulso de la forma $\text{sen}(x)/x$. El teorema de muestreo garantiza que, cuando las muestras de una señal limitada en banda son convolucionadas con la función $\text{sen}(x)/x$, la señal de entrada es reproducida de forma exacta. Específicamente, y como se muestra en la Figura 4.9A, cuando una simple muestra de audio pasa a través de un filtro paso-bajo ideal, transcurrido el retardo propio del filtro sale con una respuesta de la forma $\text{sen}(x)/x$.

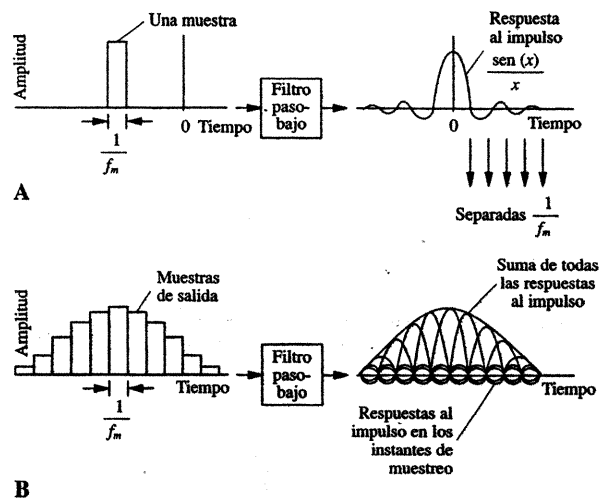


Figura 4.9. La respuesta al impulso de un filtro paso bajo ideal reconstruye la forma de onda analógica de la señal en la salida. A. La respuesta al impulso de un filtro paso bajo ideal es una función $\text{sen}(x)/x$, presentándose truncada en este dibujo. B. Cuando una serie de impulsos (muestras) pasan a través de un filtro paso bajo, se suman las respuestas al impulso individualmente formando la onda de salida.

Si el filtro tiene como frecuencia de corte la mitad de la frecuencia de muestreo ($f_m/2$), entonces la curva $\text{sen}(x)/x$ pasa por cero en los múltiplos de $1/f_m$. Cuando una serie de muestras consecutivas pasan a través del filtro, la forma de onda resultante es la suma retardada de las componentes $\text{sen}(x)/x$ asociadas a cada una de las muestras, como se puede apreciar en la Figura 4.9B. Cada respuesta al impulso de una muestra es cero justo en $1/f_m$, posición de máxima respuesta de otra de las muestras. Así, una vez sumadas las contribuciones de las diferentes respuestas al de todas las muestras, la forma de onda de la salida resultante adquiere el valor de cada muestra justo en el instante asociado a ella misma. Esta suma, o superposición, de las respuestas al impulso individuales de cada muestra reconstruye todos los puntos intermedios de la forma de onda de la señal.

De esta forma, es el filtrado de salida con un filtro paso-bajo ideal el proceso que origina una forma de onda de salida que permite reproducir la señal original. O dicho en otras palabras, es la respuesta al impulso del filtro aplicada a las muestras de audio lo que reconstruye la señal original.

FILTRO RECUPERADOR DIGITAL

Debido a la distorsión de fase que introducen los filtros analógicos de elevado orden, han sido abandonados por los fabricantes de equipos de audio en favor de los filtros digitales. Un filtro digital es un circuito (un algoritmo) que procesa muestras de audio, y modificando sus valores produce una alteración del espectro (filtrado) de la señal. En el caso que nos ocupa, el filtro digital debe simular el proceso del filtro paso bajo ideal y producir la recuperación de la forma de onda original. En vez de suprimir los espectros imagen después de que la señal haya pasado por la etapa de conversión, los filtros digitales realizan la misma función en el dominio digital,

y por tanto, antes de aplicar las muestras al convertidor D/A. Después de éste, un filtro paso bajo de pequeño orden es suficiente para eliminar los espectros imagen que quedan, ya que éstos aparecen ahora en frecuencias muy alejadas de la banda base. En la mayoría de los casos, se utilizan filtros de respuesta finita al impulso (FIR, «finite impulse response») conjuntamente con técnicas de sobremuestreo, permitiendo un filtro cuyo diseño es de baja complejidad. Con el sobremuestreo se obtienen muestras adicionales (al fin y al cabo, valores) mediante interpolación entre las muestras existentes. Puesto que por cada muestra real se generan muestras adicionales (quizás dos, cuatro u ocho por cada una), la frecuencia de muestreo de la señal de salida resulta (en la proporción anterior) más elevada que la original de la entrada. La estructura de filtros empleada habitualmente es la transversal, estando formada por una serie de retardos, multiplicadores y sumadores.

La tarea del filtro de sobremuestreo es doble: primero aplicar el sobremuestreo, y luego, mediante interpolación, filtrar la señal. La señal de entrada al filtro fue muestreada a f_m , apareciendo los espectros imagen centrados alrededor de los múltiplos de f_m , tal y como aparece representado en la Figura 4.10A y B. El sobremuestreo comienza con el incremento de la frecuencia de muestreo multiplicándola por un factor denominado factor de sobremuestreo. Se realiza intercalando un determinado número de muestras nulas por cada muestra real; por ejemplo, un sobremuestreo cuádruple se consigue insertando tres muestras nulas por cada muestra original. La frecuencia de sobremuestreo es igual al factor de sobremuestreo por la frecuencia de muestreo original, pero el espectro de la señal sobremuestreada es el mismo que el espectro original, tal y como se muestra en la Figura 4.10C y D. En nuestro ejemplo, la frecuencia de muestreo de 44,1 kHz se convierte en una frecuencia de sobremuestreo de 176,4 kHz. Los datos entran en un filtro paso-bajo digital con una frecuencia de corte de $f_m/2$, pero que funciona a una frecuencia de 176,4 kHz. Aunque los datos originales fueron muestreados a 44,1 kHz, con el sobremuestreo son indistinguibles de que hubieran sido muestreados a 176,4 kHz. La salida del filtro (véase Figs. 4.10E y F) es una señal ya interpolada, con los espectros imagen centrados alrededor de los múltiplos de la frecuencia de sobremuestreo. Esta interpolación se efectúa mediante un filtro paso bajo, cuya respuesta al impulso (véase Fig. 4.10G) comprende los coeficientes del filtro transversal, y cuyo espectro aparece en la Figura 4.10H.

Recapitulando, la interpolación es utilizada para crear muestras intermedias entre las reales. En un filtro de sobremuestreo cuádruple, el filtro entrega cuatro muestras por cada una de entrada. Sin embargo, para que puedan ser utilizables, sus valores deben ser calculados a partir de un cierto algoritmo. Específicamente, cada muestra intermedia debe ser multiplicada por el coeficiente apropiado de $\text{sen}(x)/x$, que corresponde a su contribución a la respuesta al impulso total del filtro paso bajo en el dominio del tiempo (ver Fig. 4.9B). La función $\text{sen}(x)/x$ en el dominio del tiempo tiene sus ceros alineados exactamente con todas las muestras de la señal, excepto la que está interpolando en ese instante. Así, cada muestra interpolada es una combinación lineal del resto de muestras, ponderadas todas con la función $\text{sen}(x)/x$. Los resultados de todas las multiplicaciones son sumados, obteniendo la salida de la muestra filtrada. Conceptualmente, la operación del filtro digital corresponde exactamente a la suma de las respuestas al impulso de un filtro paso bajo ideal. Los espectros imagen aparecen a los múltiplos de la frecuencia de muestreo, y puesto que la separación entre la banda base y los espectros imagen es muy grande, un filtro analógico de bajo orden es suficiente para eliminar las componentes imagen sin provocar alteraciones de fase ni otras alteraciones de la señal.

El factor de sobremuestreo se define como:

$$R = \frac{f_s}{f_m}$$

donde:

f_s = es la frecuencia de sobremuestreo.

f_m = es la frecuencia de muestreo original.

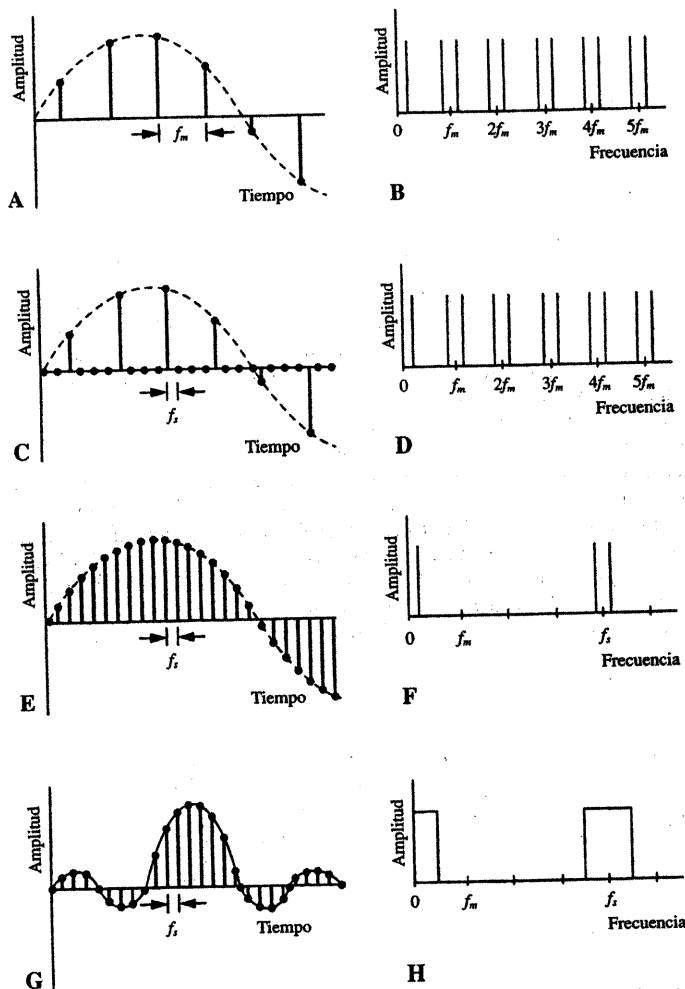


Figura 4.10. Un filtro de sobremuestreo remuestrea e interpola la señal usando la respuesta al impulso. Esto se muestra en el dominio del tiempo (columna izquierda) y en el dominio de la frecuencia (columna derecha). A. La señal de entrada es muestreada a f_m . B. Los espectros imagen aparecen centrados alrededor de los múltiplos de f_m . C. Con el remuestreo, muestras de valor nulo se disponen entre las muestras originales a alguna relación de interpolación. D. El espectro de la señal sobremuestreada es el mismo que el original. E. El filtro digital realiza la interpolación para generar valores nuevos a las muestras. F. La señal de salida filtrada tiene imágenes centradas alrededor de los múltiplos de la frecuencia de sobremuestreo f_s . G. Los valores de una respuesta al impulso muestreada corresponden a los coeficientes del filtro digital utilizado para generar los valores interpolados de las muestras. H. La función de transferencia del filtro muestra una banda de paso en la banda de audio y la banda centrada en la frecuencia de sobremuestreo.

El sobremuestreo requiere la inserción de $(R - 1)$ muestras nulas (ceros) por cada muestra de entrada; debiendo, además, quedar dispuestas de forma simétrica con respecto a las muestras reales. Un filtro paso-bajo es utilizado para limitar en banda (a $f_m/2$) los datos de entrada, apareciendo entonces los espectros imagen sólo en los múltiplos enteros de $(R \times f_m)$. Además, el filtro paso-bajo establece nuevos valores en las muestras nulas, actuando como interpolador. Así, mejor que efectuar multiplicaciones con las muestras nulas, la potencia de cálculo economizada puede emplearse para diseñar un filtro más eficiente.

La Figura 4.11 muestra un filtro de sobremuestreo cuádruple, generando tres muestras intermedias por cada muestra de entrada. El filtro consiste en un registro de desplazamiento de 24 etapas, retardando cada muestra de entrada de 16 bits un período (T_m) de la frecuencia de muestreo. Cada muestra permanece en la etapa durante un tiempo T_m , y una vez transcurrido se desplaza a la siguiente etapa. Durante este intervalo de tiempo, cada muestra de 16 bits es multiplicada por cuatro coeficientes distintos (C1 a C4), y diferentes en cada etapa, que están almacenados en una ROM interna. En total, los cuatro conjuntos de coeficientes son aplicados a las muestras de cada etapa, generando cuatro valores de salida en cada etapa. Los 24 productos se suman 4 veces en cada período, dando como resultado los cuatro valores de salida del filtro en ese período. Las características del filtro ($\text{sen}(x)/x$) determinan los valores de las muestras interpoladas. Completadas las operaciones, cada palabra de 16 bits pasa a la etapa siguiente, repitiéndose el proceso hasta que atraviesa toda la línea de retardo. El resultado de cada multiplicación es una palabra de $16 + 12 = 28$ bits, y cuando estos productos se suman, lo que se obtiene es una media ponderada de un gran número de muestras. Después del sobremuestreo, aparecen cuatro veces más muestras que en la entrada, con los valores de las muestras interpoladas calculadas por el filtro, con lo que en realidad la frecuencia de muestreo se ha incrementado en cuatro veces, pasando a ser de 176,4 kHz. Como resultado de todo el proceso obtenemos el incremento de la frecuencia de muestreo y el filtrado de la señal. Mediante la elección de los coeficientes del filtro y su número de etapas se obtiene la respuesta del fil-

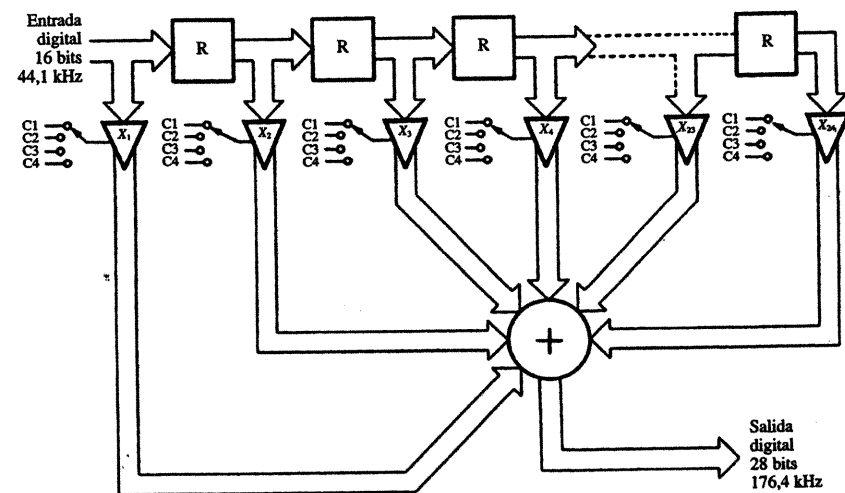


Figura 4.11. Filtro digital de veinticuatro etapas mostrando las líneas de retardo, coeficientes multiplicadores y el sumador.

tro deseada, incluyendo la corrección necesaria en el filtro recuperador. Debido al desplazamiento que sufren los datos en este tipo de estructura, este diseño se suele denominar filtro transversal.

La Figura 4.12 presenta gráficamente cómo un filtro digital de sobremuestreo cuádruple simula el efecto de un filtro recuperador analógico, mostrando el proceso de cálculo de las muestras interpoladas. Las muestras de entrada I_3, I_2, I_1 e I_0 son tratadas por respuestas al impulso $\text{sen}(x)/x$ centradas sucesivamente en los instantes de las muestras. El máximo de amplitud de la respuesta al impulso $\text{sen}(x)/x$ es igual al valor de la amplitud de la muestra, y la anchura de la respuesta al impulso está determinada por la respuesta del filtro, que en este caso es un filtro cuya frecuencia de corte es la frecuencia de Nyquist que tienen las muestras de entrada. La suma de estas contribuciones van creando los valores de las muestras interpoladas, y aunque en la figura sólo se representan cuatro muestras, son muchas más las que aportan su contribución. Cada una de las tres muestras interpoladas se forma sumando los cuatro productos, tal y como aparece representado en la figura. Las muestras originales pasan a través del filtro inalteradas, ya que utilizan un conjunto de coeficientes formado por tres ceros y un uno de forma que la muestra de salida se la hace coincidir con la original empleando dicho uno como coeficiente multiplicador.

El efecto total del filtro de sobremuestreo cuádruple se muestra en la Figura 4.13. El filtro de pendiente muy abrupta limita la banda del espectro de salida. Con el filtro de sobremuestreo se eliminan los espectros imagen entre 20 kHz y 156,4 kHz (que aparecían centrados en 44,1, 88,2 y 132,3 kHz), dejando únicamente como espectros imagen los situados en la frecuencia de

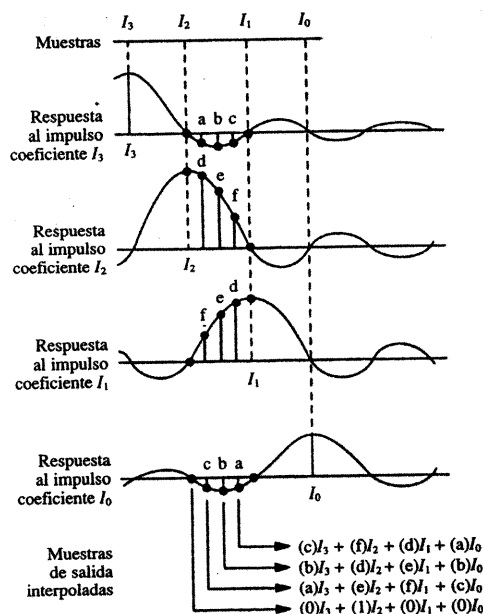


Figura 4.12. Un filtro de sobremuestreo cuádruple trata las muestras de entrada como puntos de una curva $\text{sen}(x)/x$. Esto reconstruye la forma de onda de salida como en un filtro paso bajo ideal. En la práctica se necesitan muchas muestras.

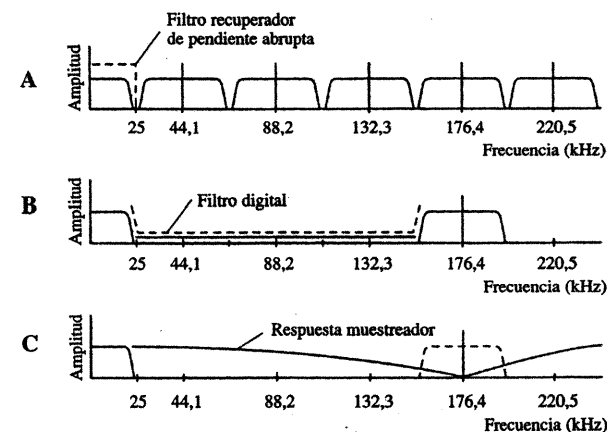


Figura 4.13. Los espectros imagen en una etapa de reconstrucción sin sobremuestreo y con sobremuestreo. A. Se necesita un filtro con una pendiente muy abrupta para limitar el espectro de salida. B. Con un sobremuestreo cuádruple las imágenes aparecen sólo en la frecuencia de sobremuestreo. C. El muestreador de salida puede utilizarse para atenuar el espectro centrado en la frecuencia de sobremuestreo.

sobremuestreo (y sus múltiplos). El circuito muestreador de la salida (el «deglitcher») puede ser también aprovechado para atenuar parte de esos espectros imagen restantes, eligiendo el tiempo de apertura para que el primer cero que introduzca coincida con la frecuencia de sobremuestreo.

En la práctica, un gran número de consideraciones determinan el diseño del filtro de sobremuestreo. La forma de onda de la respuesta $\text{sen}(x)/x$ se extiende hacia el infinito tanto en la dirección positiva como en la negativa, por lo que teóricamente, para reconstruir perfectamente la señal de entrada se necesitarían todos los valores de las infinitas formas de onda de todas las muestras. Aunque un filtro analógico puede teóricamente tener un número infinito de muestras para cada valor reconstruido, un filtro digital de respuesta finita al impulso, como su propio nombre indica, no. Así, el filtro de sobremuestreo es diseñado para trabajar con un número de muestras tal, que el error introducido será inferior que la resolución total del sistema. En el ejemplo expuesto sólo se emplearon cuatro coeficientes, y aunque se necesitarían del orden de 300 coeficientes de 28 bits, en la práctica quizás podría ser suficiente con el empleo de unos 100 coeficientes. Sin embargo, debido a que la respuesta $\text{sen}(x)/x$ es simétrica, sólo la mitad de los coeficientes necesitan ser almacenados en memoria, pudiendo leer ésta bidireccionalmente. Como se comentó, la operación producto en un filtro digital incrementa la longitud de la palabra; ya que truncar este resultado incrementaría notablemente la distorsión, la solución consiste en aplicarle «dither».

A lo largo de estos ejemplos hemos utilizado un filtro de sobremuestreo cuádruple. Sin embargo, puede usarse sobremuestreo doble, cuádruple u óctuple, en los que la frecuencia de muestreo de 44,1 kHz se convertiría en 88,1 kHz, 176,4 kHz o 352,8 kHz, respectivamente. Por ejemplo, en un filtro óctuple serían calculadas siete nuevas muestras de audio por cada muestra original, elevando la frecuencia de sobremuestreo hasta los 352,8 kHz. La ventaja es que el primer espectro imagen aparece centrado en 352,8 kHz, con lo que puede eliminarse fácilmente con un sencillo filtro paso bajo analógico. Exceptuando la velocidad a la que funcionan, todos

los filtros digitales operan de forma similar, aunque las características de los filtros analógicos sí cambian de uno a otro equipo. Disminuir la frecuencia de muestreo conlleva aproximar el primer espectro imagen a la banda de audio, con lo que obliga a la utilización de filtros de pendientes más abruptas. Cuando se emplean los convertidores D/A tradicionales, el límite práctico ronda el sobremuestreo óctuple, ya que la mayoría de los convertidores no pueden trabajar a velocidades más elevadas. Por otro lado, los convertidores D/A pueden más fácilmente convertir una señal con sobremuestreo alto que la original correspondiente, debiéndose esto a que con el sobremuestreo los cambios de amplitud de una muestra a la siguiente son más reducidos. Hablando de forma más técnica, diríamos que el «slew-rate» o velocidad de variación en la forma de onda de una muestra a la siguiente disminuye. Este hecho, junto con un menor sobreimpulso y oscilación transitoria de una muestra a la siguiente, provoca una distorsión de intermodulación más baja que en montajes con la configuración tradicional. Además, hay fabricantes que opinan que con factores de sobremuestreo óctuplos o superiores, los espectros imagen aparecen en frecuencias tan elevadas que un simple filtro de segundo orden basta para eliminarlos.

En términos de prestaciones, los filtros digitales representan una gran mejora sobre los analógicos, debido a que pueden ser diseñados con una gran precisión y ser muy estables. Puesto que el filtrado digital es únicamente un proceso de cálculo numérico, las características del filtro no cambian con la temperatura, envejecimiento, etc., cosa que no sucede con los filtros analógicos. Un filtro digital puede tener un rizado en la banda de paso tan pequeño como $\pm 0,00001$ dB, una atenuación en la banda atenuada de 120 dB y una respuesta en fase lineal. La atenuación introducida por el error de apertura puede ser compensada en el filtro transversal eligiendo los coeficientes de forma que se genere un pequeño crecimiento en la respuesta en frecuencia antes de la frecuencia de corte. La teoría de los filtros digitales se estudiará en el Capítulo 17.

Conformador de ruido

Otra ventaja importante del sobremuestreo es la disminución del ruido de cuantificación dentro de la banda de audio. Por ejemplo, en un filtro de sobremuestreo cuádruple, los datos abandonan el filtro a una velocidad cuatro veces más elevada que la de la señal de entrada. La potencia del ruido de cuantificación se extiende a lo largo de una banda cuatro veces más ancha, reduciéndose la densidad de potencia de ruido a una cuarta parte, como se muestra en la Figura 4.14. En este ejemplo se introduce una reducción del ruido de 6 dB (3 dB cada vez que se duplica el ancho de banda). Factores de sobremuestreo mayores permiten disminuir de forma proporcional el ruido; así que, por ejemplo, con un sobremuestreo óctuple la mejora sería de otros 6 dB sobre el caso anterior.

El conformador espectral («Noise Shaping») puede reducir aún más el ruido que acompaña a la señal de audio. Este proceso se efectúa recuantificando la señal de forma previa a la conversión D/A. Cuando estos errores de cuantificación son independientes de la señal, el espectro del error es blanco, mientras que si el error depende de la señal puede modelarse su espectro. Esto se realiza mediante un algoritmo que produce «errores» que son estadísticamente dependientes de la señal. Por ejemplo, el esquema de un conformador de ruido sencillo aparece representado en la Figura 4.15A. Las palabras de salida del filtro tienen 28 bits de longitud, se redondean y se les aplica «dither» para quedarnos con los 16 bits de mayor peso de la palabra de 28. Los 12 bits menos significativos (diferencia entre el valor original y el representado por los 16 bits) son retardados un período y restados de la siguiente palabra de datos. A frecuencias próximas a 88,2 kHz, esta realimentación de la señal de error aparece en fase con la señal de entrada, con lo que el error (ruido) se incre-

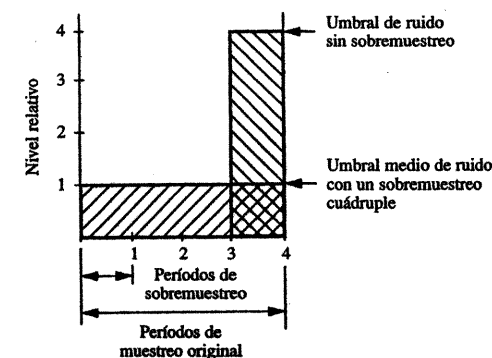


Figura 4.14. El sobremuestreo extiende el error de cuantificación sobre una banda de frecuencia mayor, reduciendo el ruido de cuantificación en la banda en la misma proporción.

menta, tal y como se muestra en la Figura 4.15B. Como la potencia de ruido se mantiene constante, este incremento fuera de la banda de audio permite que en ésta el ruido disminuya en una magnitud aproximada de unos 7 dB. El conformador de ruido se estudiará en detalle en el Capítulo 18.

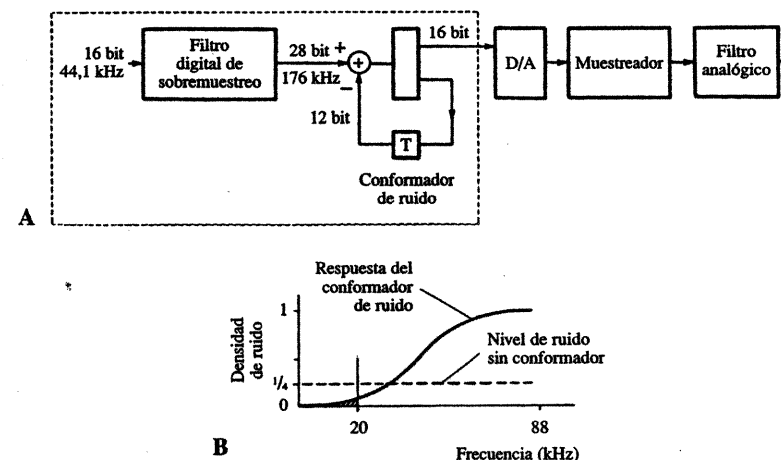


Figura 4.15. El uso del conformador de ruido tras el sobremuestreo disminuye el error de cuantificación en la banda de audio. A. Un conformador simple. B. El conformador suprime el ruido en la banda de audio, realizando el ruido fuera de la banda donde será filtrado.

Tratamiento de salida

A continuación del filtro digital, los datos son convertidos de nuevo en una señal analógica mediante el convertidor D/A. En el caso de un sobremuestreo cuádruple, el error de apertura (sen $(x)/x$) del circuito de muestreo y retención de salida crea un cero en 176,4 kHz ayudando a la eliminación del espectro imagen centrado en esa frecuencia. Por otra parte, el filtro digital debe compensar las ligeras pérdidas que en el extremo superior de la banda de audio aparecen como consecuencia del error de apertura. El resto de los espectros imagen, comenzando por el de 176,4 kHz, pueden ser eliminados por un filtro analógico de bajo orden, finalizando con esta etapa la cadena de conversión digital-analógica. El filtro puede ser uno de segundo orden (atenuación de 12 dB/octava) y con frecuencia de corte (caída de la respuesta de 3 dB respecto a la banda de paso) situada en torno a los 30 o 40 kHz. Su diseño no es crítico, pudiendo usar una función de aproximación (como la de Bessel) que asegure una buena linealidad de fase con errores inferiores a $\pm 0,1^\circ$ dentro de toda la banda de audio.

Ya que el sobremuestreo ha disminuido el ruido de cuantificación en 6 dB y el conformador de ruido unos 7 dB, cualquier convertidor D/A de 16 bits puede producir buenas prestaciones.

MÉTODOS DE CODIFICACIÓN ALTERNATIVOS

La codificación en PCM lineal suele considerarse como un esquema de codificación clásico para la señal de audio, permitiendo una buenas prestaciones. Hay otros métodos de digitalización, aunque todos presentan claras ventajas e inconvenientes comparados con el PCM. Aunque un sistema PCM lineal utiliza unos niveles de cuantificación de la misma amplitud independientemente del nivel de la muestra, hay sistemas de codificación especiales que modifican por completo esta característica. Una ventaja de estos sistemas es la reducción de datos que se puede conseguir, de forma que necesitan un menor número de bits para codificar la señal, aunque el inconveniente principal es que reducen la calidad de la señal de audio.

En un sistema PCM lineal, los niveles de cuantificación son constantes e independientes del nivel de la señal a cuantificar. La longitud de palabra del cuantificador determina el número de niveles disponibles para la codificación de una muestra de audio, siendo los intervalos de cuantificación de igual amplitud y asignados a palabras digitales en un orden monótono creciente.

La utilización de palabras con un número de bits elevado permite reducir el error de cuantificación, aunque el inconveniente es la necesidad de transmitir/almacenar más bits por muestra, y por tanto requerir un mayor ancho de banda. Aunque el esquema de cuantificación uniforme es óptimo para señales con una distribución de amplitud uniforme, esta característica no se cumple en la mayoría de las señales de audio. Con el fin de minimizar el error de cuantificación, en los sistemas alternativos al PCM, lo que se hace es utilizar pasos de cuantificación de amplitud no constante, entrando en los denominados sistemas de cuantificación no uniforme. Todos ellos se basan en adaptar los pasos de cuantificación a las propiedades estadísticas de la señal. Por ejemplo, para las señales vocales es adecuado un sistema de cuantificación exponencial; éste asume que las señales son predominantemente de pequeña amplitud, y tomando en raras ocasiones un nivel elevado. Así, destinando muchos pasos en la cuantificación de los niveles bajos y pocos pasos de cuantificación a los niveles elevados, se puede reducir el error de cuantificación cometido. Estos sistemas se suelen denominar companders², comprimiendo la dinámica de la señal antes de la conversión A/D, para aplicar tras la conversión D/A un expansor que devuelva las señales a sus niveles originales (teóricamente). Los companders y los sistemas de punto flotante utilizan un elemento que cambie el nivel de la señal a convertir, per-

mitiendo una expansión del rango dinámico del sistema. Una modificación sustancial del sistema PCM lo constituye la modulación delta. Utiliza sólo un bit para efectuar la cuantificación de la señal, aunque requiere una frecuencia de muestreo muy elevada para funcionar correctamente. Otra variante de la modulación delta incluye los sistemas de modulación delta adaptativa y modulación delta predictiva adaptativa, ofreciendo todos ellos ventajas e inconvenientes propios. Otra forma de reducir los datos transmitidos se basa en la codificación perceptual, aunque de ella hablaremos en el Capítulo 10.

Codificación en coma flotante

Los sistemas de codificación en coma flotante corresponden a una modificación de los sistemas PCM empleando un factor de escala. En los verdaderos sistemas de coma flotante se utiliza un factor de escala en la codificación de cada muestra, aunque hay otros sistemas que trabajan en bloques de datos, aplicando el factor de escala a un conjunto grande de datos.

Un sistema de coma flotante es un sistema de cuantificación no uniforme, utilizando un cuantificador para crear una palabra dividida en dos partes: la mantisa o valor de la muestra y el exponente o factor de escala. La mantisa representa el valor de la forma de onda (muestra), siendo la magnitud del fondo de escala determinada por el exponente. En otras palabras, el exponente actúa como un factor de escala que varía la ganancia de la señal antes de la conversión PCM. Ajustando la ganancia con la que trabaja la etapa previa al convertidor, se optimiza la utilización del A/D. La etapa de control de ganancia (generación del exponente) se encarga de aportar una amplificación tal que se amplifiquen las señales de bajo nivel y atenúen las de nivel alto, de forma que antes de entrar en el convertidor, la señal haya adquirido un nivel lo más elevado posible sin exceder el rango del convertidor. Este proceso de amplificar/atenuar la señal antes de la conversión produce que, de forma efectiva, el paso de cuantificación dependa de la amplitud de la señal. Por último, la corrección de ganancia introducida se debe compensar tras la conversión D/A, de forma que se restablezcan los niveles originales de la señal.

Tomemos como ejemplo un sistema de coma flotante con una mantisa de 10 bits (convertidor de 10 bits) y un exponente de 3 bits (selector de ganancia), tal y como se muestra en la Figura 4.16. El exponente de tres bits permite una selección de la ganancia con 8 (2^3) rangos para cada conversión de 10 bits, lo que es equivalente a tener un margen de ganancia de 1 a 128^3 . Teniendo en cuenta la acción conjunta del convertidor de 10 bits ($2^{10} = 1.024$ niveles) y el control de ganancia (127 márgenes), los niveles de cuantificación que se consiguen son, aproximadamente, $1.024 \times 127 = 130.048$. A la vista de esta cifra se puede observar cómo la representación de la muestra con 13 bits es equivalente al empleo de un convertidor de 17 bits, con la ventaja de necesitar sólo un verdadero convertidor de 10 bits. Sin embargo, y como en toda conversión no uniforme, lo que no se puede mantener a lo largo de todo el margen es una resolución constante. Como muestra un ejemplo relacionado con el anterior, uno de los valores que se pueden representar es el 65.408 (ganancia 128 y escalón 511); el valor inmediato inferior no es el 65.407 como ocurre en un sistema PCM, si no el 65.280 (ganancia 128, escalón 510). En general, a medida que el nivel de la señal se incrementa, el número de intervalos de cuantificación no crece en proporción, con lo que el error de cuantificación crece y la relación Señal/Ruido disminuye. En todos estos sistemas, la relación S/R es función del nivel de la señal, y su magnitud es siempre menor que el rango dinámico.

³ Nota del traductor. Para hacer que los márgenes de conversión se solapen y produzcan un margen de conversión continuo, la ganancia que aporta una etapa debe ser justo el doble que la anterior. Así, si a la primera etapa le asociamos una ganancia de 1, la siguiente sería de valor doble ($2^1 = 2$), la siguiente el doble de la anterior, $2^2 = 4$, ..., y la última $2^7 = 128$. De esta forma con m bits de exponente se consiguen márgenes de ganancia de 1 a 2^{m-1} .

² Nota del traductor. Contracción derivada de la denominación anglosajona «Compressor/expander», que hace alusión al funcionamiento del conjunto.

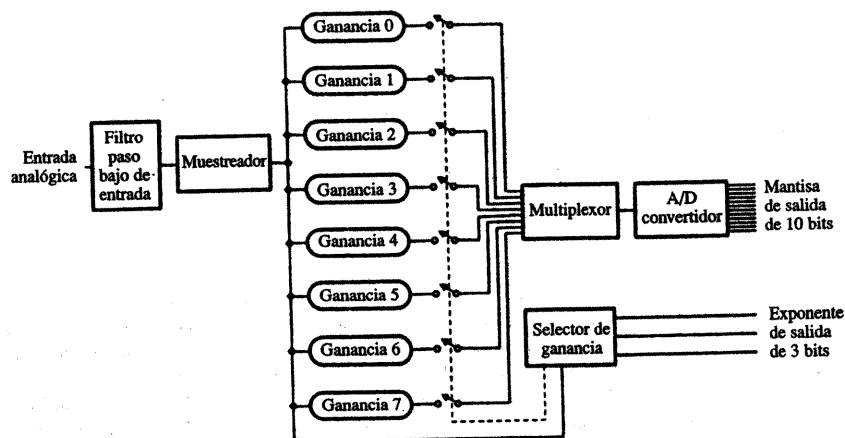


Figura 4.16. Un convertidor de coma flotante utiliza etapas de ganancia múltiple unas de otras para manipular la amplitud de la señal y optimizar la conversión A/D.

Los sistemas de punto flotante utilizan convertidores cuya longitud de palabra es función de la relación S/R a obtener, palabra larga para señales de gran dinámica y corta para las de dinámica más reducida. Por ejemplo, un sistema de punto flotante usando un convertidor A/D de 16 bits y exponente de 3 bits ajustado sobre una dinámica de 42 dB (en pasos de 6 dB) podría permitirnos un margen dinámico de 138 dB ($96 + 42$ dB). Este tipo de sistema es elegido en la codificación de señales en condiciones extremas. Las ventajas de este sistema es que sólo requiere trabajar con una palabra de 19 bits, mientras que su equivalente PCM necesitaría una palabra de 23 bits. Además, cuando la ganancia de la etapa previa a la conversión se establece en pasos de 6 dB, las palabras codificadas pueden ser fácilmente convertidas a código PCM bien sea para su procesamiento o almacenamiento sin requerir cálculo adicional. Únicamente, provocar en los bits de la mantisa una operación de desplazamiento lógico en función de la magnitud del exponente.

Aunque en los sistemas de punto flotante el rango dinámico sea grande, tiene una naturaleza diferente a la de los sistemas de coma fija, teniendo una relación S/N siempre inferior. Esto se debe a que el margen dinámico expresa la diferencia entre la máxima señal admisible y el ruido en ausencia de señal, mientras que en la relación señal ruido se mide éste cuando la señal está presente. En un sistema de coma fija, el rango dinámico es aproximadamente igual a la relación S/R. Sin embargo, en un sistema de coma flotante la relación S/N queda prefijada por la resolución del convertidor A/D; y siendo aproximadamente 6n dB es independiente de la etapa de amplificación previa, y por tanto del margen dinámico máximo. Los cambios en la amplitud de la señal son los que provocan cambios de ganancia en la etapa previa, hecho que motiva que la amplitud de los pasos de cuantificación, y por tanto la magnitud del error de cuantificación, sea dependiente de dicha ganancia. Así, la relación S/N está variando continuamente con los cambios del exponente (ganancia) de cada palabra. Por ejemplo, considerar un sistema con una mantisa de 10 bits y un exponente de 3 bits con un intervalo de ganancia de 6 dB. La máxima relación S/N es de 60 dB. Si el nivel de la señal disminuye, también lo hace la S/N, cayendo a un valor mínimo de 54 dB. En ese momento, la disminución de la señal provoca el cambio de ganancia (exponente), con lo que de nuevo el convertidor pasa a trabajar a fondo de

escala, y la S/N vuelve a ser la máxima. En convertidores de longitud de palabra larga, una señal compleja produce el enmascaramiento del error de cuantificación; sin embargo, con tonos puros el error puede ser perceptible. Más defectos audibles: en señales de baja frecuencia, la modulación del ruido de fondo, y en señales de muy bajo nivel (casi inaudibles), la percepción del ruido de cuantificación. Otro problema añadido en estas estructuras aparece por la falta de precisión en la calibración de la etapa de ganancia previa, donde un error en el solapamiento entre márgenes puede provocar discontinuidades en los códigos generados.

En los convertidores de coma flotante, la forma de llevar a la práctica la etapa de cambio de ganancia afecta considerablemente la audibilidad de los errores. La conmutación instantánea de la ganancia de una muestra a la siguiente provoca una acentuación de los problemas. Lo ideal es que el cambio de ganancia se realice con un algoritmo que siga la tendencia de la amplitud de la señal, basándose en el tipo de programa a codificar. Por ejemplo, en la señal de voz se pueden utilizar algoritmos silábicos, adaptados a la velocidad con la que se suceden las sílabas en una determinada forma de habla. Las disminuciones de ganancia se efectúan casi instantáneamente (para evitar exceder el margen dinámico del convertidor), mientras que los aumentos de ganancia son retrasados, hasta confirmar el decrecimiento o anulación de la señal. Esto aproxima dichos sistemas a las estructuras de coma flotante que trabajan no muestra a muestra sino por bloques, y que serán descritas a continuación.

Sistemas de conversión de coma flotante en bloques

Estas arquitecturas se basan en los convertidores de coma flotante. Su principal ventaja sobre los sistemas vistos es la reducción de datos que se produce, lo que los convierte en sistemas ideales para la transmisión de datos vía satélite o mediante cualquier otro medio, donde el coste de transmisión dependa del tiempo o del ancho de banda consumido. Además, estos sistemas son fáciles de adaptar a la codificación mediante algoritmos silábicos o companders.

En estos sistemas, un convertidor A/D PCM tradicional precede la etapa de escalado (control del fondo de escala). El proceso a seguir es muy sencillo: se toma la señal analógica en un intervalo corto de señal, por ejemplo, 1 ms, convirtiendo en palabras digitales las muestras del intervalo, que ahora constituirán un bloque de datos. A continuación, se calcula el factor de escala, que representa el mayor valor en el bloque; los datos se escalan hacia arriba para que el mayor valor del bloque coincida con el fondo de escala. Esto reduce el número de bits necesarios para representar la señal. El bloque de datos se transmite junto con el factor de escala aplicado, para que en la etapa de decodificación las palabras sean correctamente des-escaladas. En el ejemplo de la Figura 4.17 se utiliza un convertidor de 16 bits, con el fin de producir bloques

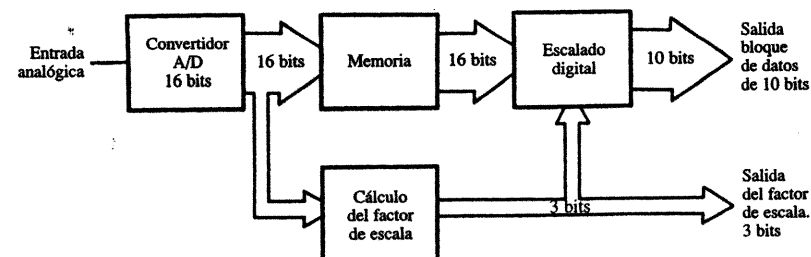


Figura 4.17. Un sistema de conversión de coma flotante por bloques utiliza un único factor de escala para todos los datos del bloque.

de datos de 10 bits de longitud de palabra (mantisa) y un factor de escala (exponente) de 3 bits. Debido a que sólo se necesita un exponente por bloque de datos, se incrementa la eficiencia en la transmisión con respecto a los sistemas tradicionales de coma flotante, aunque el inconveniente es la necesidad de un A/D con una longitud de palabra más elevada.

Los sistemas de coma flotante por bloques evitan los problemas que aparecen en los esquemas de punto flotante asociados a los cambios de escala producidos al trabajar independientemente con cada muestra. Ahora, la variación del nivel de ruido de fondo (al permutar de margen) se produce al cambiar de bloque (p. ej., 1 ms); y debido a que este intervalo es muy corto comparado con el de la percepción humana, sus efectos son apenas detectados.

Un esquema de coma flotante trabaja mejor cuando la señal de audio tiene un factor de cresta (relación entre el valor de pico de la señal y su valor eficaz medio) bajo a lo largo de cortos intervalos temporales. Puesto que la mayor parte de los estilos musicales poseen estas características, las prestaciones de estos sistemas suelen ser buenas. Pero cuando estos sistemas destacan es cuando el cambio de nivel entre muestras es grande (como en los picos de señal), aunque cuando esto no se cumple sus prestaciones son inferiores a las de los esquemas lineales de digitalización. Lo que hay que tener presente es que la principal desventaja de los sistemas alternativos de digitalización es que sus prestaciones y su mejor o peor comportamiento dependen del tipo de programa musical.

Sistemas de codificación no uniforme

Hemos visto diferentes sistemas de conversión A/D. Primero los sistemas PCM lineales, caracterizados por el empleo de pasos de cuantificación de amplitud constante a lo largo de todo el rango dinámico. Posteriormente vimos los esquemas de coma flotante, sistemas donde el paso de cuantificación era función de la amplitud de la señal, variación obtenida mediante la conmutación de rango (escala) efectuada en la etapa previa a la conversión. Ahora vamos a ver otro esquema de codificación no uniforme denominado compander, derivado de la compresión de la señal previa a la conversión. No es un método óptimo para lograr una cuantificación no uniforme, pero permite una fácil implementación desde el punto de vista electrónico.

En un sistema de codificación no uniforme por compresión, los pasos de cuantificación están muy separados en los niveles altos de señal, mientras que aparecen mucho más próximos en los niveles bajos de señal. Esto se consigue mediante compresión y expansión de la señal, de ahí el término compander para designar este tipo de sistemas. Con la compresión de la señal previa a la conversión se logra amplificar los niveles bajos, mientras que se atenúan los niveles altos de señal. Este tratamiento se ajusta perfectamente a señales como la voz, en las cuales las pequeñas amplitudes de la señal suceden mucho más a menudo que los picos de gran nivel.

Para obtener el compander se utiliza un compresor con una función de transferencia logarítmica, seguido de un convertidor PCM con cuantificación lineal. La compresión previa asociada al convertidor lineal produce el efecto de una cuantificación no uniforme. En la etapa de salida se utiliza un expansor para devolver los niveles de señal a sus valores originales. Así, los pasos de cuantificación son aprovechados de una forma más efectiva dentro del rango dinámico de la señal de entrada. Un compander del tipo comentado aparece representado en la Figura 4.18, observando las funciones de transferencia conjuntas de la etapa de codificación (compresor-A/D) y decodificación (D/A-expansor). Esta técnica es similar a la que utilizan los sistemas reductores de ruido en los magnetófonos analógicos. La diferencia estriba en nuestro caso en que tanto compresor como expansor están enlazados directamente con los convertidores, con lo que los errores de seguimiento que aparecen en los sistemas analógicos aquí son inexistentes. Las señales de elevado nivel son fácilmente codificadas, mientras que las de bajo nivel (que en el tiempo ocupan la mayor parte de la duración del programa) quedan afectadas de un error de cuantificación muy pequeño, derivado del proceso de la compresión. El resulta-

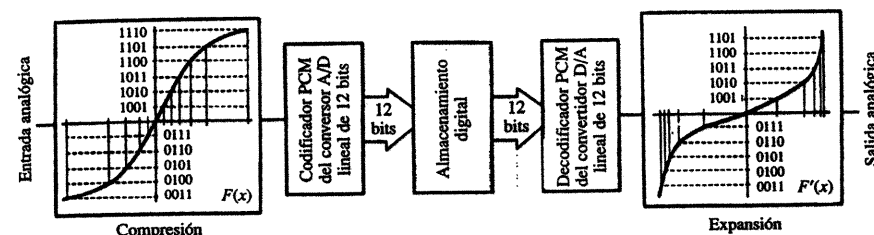


Figura 4.18. Un sistema de conversión no lineal utiliza un compresor antes de convertir la señal en digital y un expansor tras la conversión de la señal en analógica.

do para las señales de bajo nivel es el de gozar de una gran relación S/R, y por tanto de un incremento del rango dinámico en comparación con los sistemas PCM tradicionales. A costa de estas mejoras, lo que sí se obtiene es un incremento del ruido de cuantificación en los niveles altos, siendo además dependiente de la señal; sin embargo, la elevada amplitud de la señal tiende a enmascarar dicho ruido. Por otro lado, la mayor audibilidad del efecto de modulación del ruido aparece en señales de baja frecuencia con rápidos cambios de amplitud, aunque estas componentes no sean habituales en señales de tipo vocal.

Companders de ley μ y ley A

El sistema de codificación con ley μ es comúnmente empleado en aplicaciones de señal vocal, particularmente en Norteamérica y Japón, apareciendo definido por el CCITT («Committee Consultative International Telegraph and Telecommunication») bajo la recomendación G.711. Es un sistema de cuantificación no uniforme, con pasos de cuantificación que se incrementan logarítmicamente, a medida que crece el nivel de la señal. Su ley de compresión queda definida por:

$$y = \frac{\log(1 + \mu x)}{\log(1 + \mu)} \quad \text{para } x \geq 0$$

siendo:

y = magnitud de la señal de salida.

x = magnitud de la señal de entrada.

μ = un parámetro positivo cuyo valor depende de la característica de compresión deseada.

La señal es comprimida (con la función dada) antes de la cuantificación, siendo la función inversa utilizada para la expansión de la señal. Un valor de $\mu = 0$ corresponde a una amplificación lineal, es decir, sin compresión, y por tanto a una cuantificación uniforme. Valores elevados de μ provocan una mayor compresión, y el sistema estandarizado para la codificación vocal corresponde a un valor de $\mu = 255$. Con esta compresión previa y la utilización de un convertidor lineal de 8 bits se alcanza una relación S/N (en bajos niveles) y un rango dinámico total equivalente al que se obtendría con una cuantificación uniforme de 12 bits, con la ventaja de la disminución del régimen binario para la transmisión de la señal.

La ley A es también una ley de compresión logarítmica muy similar a la ley μ . Aparece también estandarizada por el CCITT, aunque se utiliza en Europa y en otros lugares del mundo (a excepción de los de la ley μ). Su ley de compresión es de la forma:

61

$$y = \begin{cases} \frac{Ax}{1 + \log A} & \text{para } 0 \leq |x| \leq \frac{1}{A} \\ \frac{1 + \log(Ax)}{1 + \log A} & \text{para } \frac{1}{A} \leq |x| \leq 1 \end{cases}$$

siendo A un parámetro positivo.

La Figura 4.19 muestra las leyes A y μ para varios valores de dichos parámetros. Generalmente, los esquemas de codificación con companders requieren en torno a cuatro bits menos por muestra para aportar una calidad equivalente a la de PCM lineal, con lo que, por ejemplo, ocho bits podrían ser suficientes en lugar de doce. Debido a que las señales de voz se muestrean generalmente a 8 kHz, el régimen binario estándar para sistemas con estas leyes de compresión es de 64.000 bits/segundo (64 kbps). Los dispositivos empleados para convertir/expandir las señales con estos sistemas son denominados codecs (contracción de «coder/decoder»).

PCM diferencial

Los sistemas PCM diferenciales (DPCM) son otros esquemas de conversión no lineales, perteneciendo a los métodos de cuantificación no uniforme. Intuitivamente, un sistema que mejora al PCM es aquel en el que en vez de cuantificar el valor de cada muestra, se cuantifique la variación (diferencia) entre dos muestras consecutivas. Como la diferencia entre muestras siempre es más pequeña que la amplitud de cada una de ellas, se necesitan muy pocos bits para cuantificar estas diferencias. Además, si la frecuencia de muestreo es lo suficientemente elevada, el valor medio de la onda cambia muy ligeramente de una muestra a la siguiente, por lo que la mayoría de las señales muestran un grado de correlación elevada entre muestras sucesivas. Los sistemas diferenciales aprovechan y explotan la redundancia que existe de una muestra a la siguiente, por lo que mejor que codificar la amplitud de la señal, con pocos bits se codifica la pequeña diferencia entre muestras consecutivas.

Además, los sistemas diferenciales emplean un sistema de codificación predictiva. Ahora, la señal que se cuantifica es la diferencia entre el valor real de la muestra y la predicción de

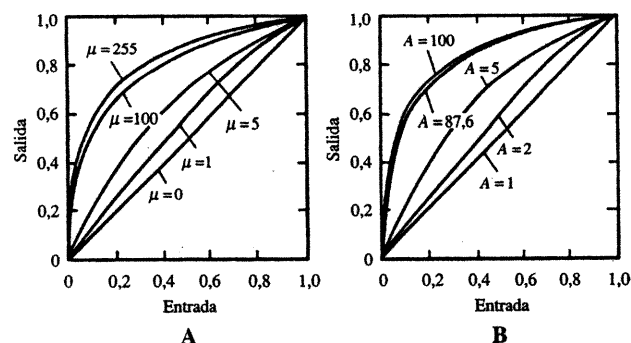


Figura 4.19. Las características de compresión-expansión determinan cómo varía la magnitud del paso de cuantificación en función del nivel de la señal. A. Características de la ley μ . B. Características de la ley A.

amplitud que estimamos que va a tener esa señal, por lo que en realidad, esa señal diferencia corresponde al error que cometemos en la predicción de la señal. El decodificador efectúa una predicción a partir de los datos previos ya recibidos, y con la predicción y la señal diferencia recibida va reconstruyendo la señal muestra a muestra. Este método requiere muy pocos bits para codificar una señal de audio, pero la calidad obtenida depende del tipo de función empleada para efectuar la predicción y su capacidad de anticiparse a los cambios de señal. La velocidad con la que la amplitud de la señal puede variar es un factor inherentemente limitado en todos los sistemas PCM diferenciales. La amplitud de la señal codificada decrece a 6 dB/oct a medida que aumenta la frecuencia, por lo que la relación S/R también decrece a 6 dB/oct. La respuesta en frecuencia de la señal codificada puede ser ecualizada para mejorar la relación S/R. Por ejemplo: una señal con un contenido bajo en alta frecuencia puede ser ecualizada para incrementar el nivel de esta banda, e invirtiendo el proceso en la decodificación, el ruido de fondo puede ser enmascarado por el elevado contenido en bajas frecuencias de la señal. Por último, desde el punto de vista económico son sistemas muy rentables, por el bajo coste de la circuitería y por el escaso régimen binario de transmisión que requieren.

Modulación delta

Los sistemas diferenciales codifican la diferencia entre la señal de entrada y la predicción que se realiza de la misma. A medida que la frecuencia de muestreo se eleva, la posible magnitud del cambio entre muestras disminuye y la resolución en la codificación aumenta. La modulación delta (MD) es un tipo de PCM diferencial que lleva este método al límite. Emplea una frecuencia de muestreo lo suficientemente elevada como para que en la cuantificación de la señal diferencia baste un solo bit para codificar la forma de onda de la señal. El modo de operación de un sistema basado en la modulación delta aparece representado en la Figura 4.20. La forma de codificar la señal es asociar a un nivel positivo o negativo si la señal crece o decrece con respecto a la predicción. Puesto que la forma de onda de la señal sólo puede variar de una muestra a la siguiente en un paso de cuantificación, para seguir correctamente los transitorios de la

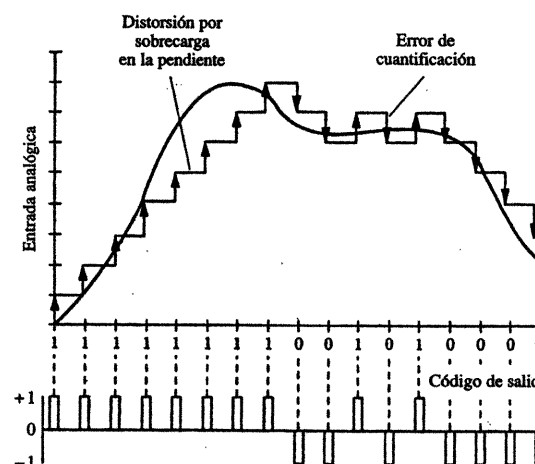


Figura 4.20. En un codificador de modulación delta, se utiliza un bit diferencial para codificar la señal de audio.

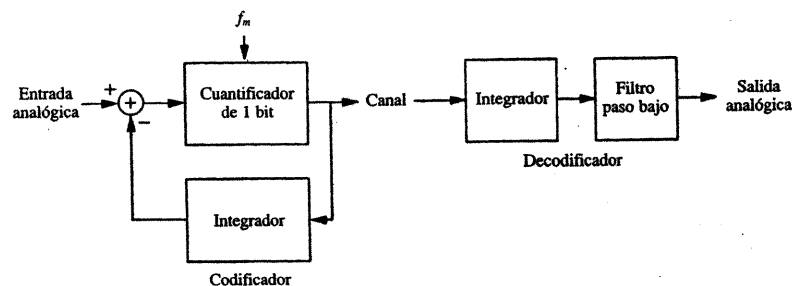


Figura 4.21. Codificador y decodificador de modulación delta.

señal se requiere una frecuencia de muestreo muy elevada. En el ejemplo de la Figura 4.20, la frecuencia de muestreo es insuficiente para seguir el crecimiento tan rápido de la señal. En estos instantes de sobrecarga del sistema, en vez de codificarse la forma de onda de la señal lo que se transmite es la máxima variación que permite el codificador, deformando la forma de onda de la señal original.

La modulación delta es un sistema sencillo y eficiente desde el punto de vista circuital, tal y como muestra su diagrama de bloques representado en la Figura 4.21. Los integradores son utilizados como predictores de primer orden. La diferencia entre la señal de entrada y el valor predicho es el error; se cuantifica como una palabra de un bit, y se genera en el mismo instante del muestreo. El sistema determina si el signo del error es positivo o negativo, y aplica un pulso del mismo signo a la entrada del integrador. Éste genera una nueva predicción, palabra digital con un valor de un paso de cuantificación mayor o menor que la anterior, de acuerdo al signo del error, y siempre en el sentido de aproximarse al valor de la muestra. La precisión de la codificación se basa en el tamaño del incremento, o paso de cuantificación. Además, la señal debe cambiar en cada muestra, no pudiendo mantenerse el nivel constante. En la salida, la señal es decodificada mediante un integrador, el mismo empleado para estimar la señal de entrada en el codificador. Como en cualquier sistema DPCM, la amplitud de la señal codificada disminuye con la frecuencia a un ritmo de 6 dB/oct, por lo que la relación S/R decrece también con la misma velocidad. Únicamente puede efectuarse una corrección en cada intervalo de muestreo, pero es claro que, teóricamente al menos, una frecuencia de muestreo lo suficientemente elevada permitiría seguir correctamente cualquier transitorio por muy rápido que éste fuese. La modulación delta ofrece unas buenas prestaciones desde el punto de vista de los errores. En un sistema PCM lineal, un error en el MSB supone una gran discontinuidad de la señal. En la modulación delta no hay MSB, por lo que cada bit se limita a seguir la diferencia entre muestras, limitando la magnitud del error cometido precisamente a esa diferencia. Lo que sí hay es posibilidad de degradación de la señal, para lo que se utilizan bits de paridad y el entrelazado.

Desde el punto de vista práctico, las prestaciones de la modulación delta caen en las aplicaciones de alta fidelidad, debido al equilibrio o compromiso que existe entre la frecuencia de muestreo y el paso de cuantificación. La resolución del codificador depende directamente del paso de cuantificación. Reduciendo la amplitud del paso, se obtiene una mejor aproximación a la señal original, y por tanto se reduce el error de cuantificación. Sin embargo, cuando se usa un paso muy pequeño, la codificación no puede seguir la compleja forma de onda de la señal de audio, convirtiendo estas limitaciones del «slew-rate» en distorsión transitoria. La frecuencia de muestreo puede ser elevada para compensar este defecto, pero para codificar señales de gran ancho de banda, las velocidades que se necesitan son tan elevadas que hacen el sistema casi impracticable.

Perceptualmente, cuando hay una distorsión por sobrecarga causada por una señal de alta frecuencia, las propias componentes de la señal tienden a enmascararla. El error de cuantificación está presente siempre, y es más audible en las señales de bajo nivel. El paso de cuantificación óptimo es aquel que minimiza la suma de los valores cuadráticos medios de los valores de la distorsión debida a la cuantificación y la distorsión de sobrecarga. Timothy Darling y Malcolm Hawksford han demostrado que la relación señal-erro (S/E) de un esquema de codificación basado en la modulación delta es de la forma:

$$S/E \text{ (dB)} = \left[-16 + 10 \log \left(\frac{f_m^3}{f_b \cdot f_i^2} \right) \right]$$

donde:

f_m = frecuencia de muestreo.

f_b = ancho de banda donde se mide el ruido.

f_i = frecuencia a la que se mide la señal.

Para obtener una relación S/R de 96 dB en un ancho de banda de 20 kHz, un sistema de MD requiere un reloj de 200 MHz. Aunque la duplicación de la frecuencia de muestreo en MD produce un incremento de 9 dB en la relación S/R, duplicar la longitud de palabra en un sistema PCM produce un crecimiento exponencial de la S/R. Desde el punto de vista de transmisión de la información, podemos apreciar cómo la propia naturaleza de la MD es un obstáculo en su capacidad de codificar la información de audio. Una frecuencia de muestreo de 500 kHz, por ejemplo, podría teóricamente permitir la codificación de frecuencias de hasta 250 kHz, pero la mayor parte del ancho de banda es empleado en la codificación de las señales de audio de frecuencia más baja. En pocas palabras, desde el punto de vista de teoría de la comunicación, el sistema de modulación delta es poco eficiente en la codificación de señales de audio.

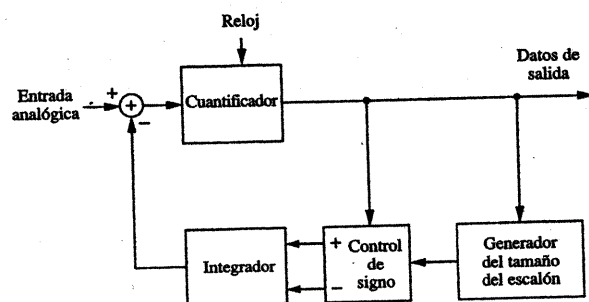
Por otro lado, la elevada frecuencia de muestreo requerida por la modulación delta, por lo menos, ofrece una ventaja: permite usar «noise-shaping». Cuando veíamos el sobremuestreo, una de las ventajas es que cada vez que duplicábamos la frecuencia de muestreo el error de cuantificación (ruido) se reducía en 3 dB. El ruido total permanece constante, pero como se reparte en un ancho de banda mayor, el ruido dentro de la banda de audio disminuye. Este hecho ayuda a que en un esquema trabajando con MD el umbral de ruido esté situado en niveles muy bajos. Además, con la elevada frecuencia de muestreo usada, el filtro recuperador no es necesario que tenga una pendiente muy abrupta, con lo que un filtro de pendiente suave (bajo orden) y baja distorsión de fase basta para atenuar los espectros superiores a la banda de audio. Por supuesto que los conversores no son los convencionales A/D y D/A que hemos utilizado hasta ahora, sino que son diseñados especialmente para este esquema. Por último, y debido a sus evidentes limitaciones, la modulación delta no es muy utilizada en aplicaciones de alta fidelidad. Sin embargo, una variación de ella conocida como modulación sigma-delta es ampliamente utilizada, ofreciendo excelentes resultados, mediante sobremuestreo, tanto en la conversión A/D como en D/A (veremos este sistema en el Capítulo 18). El formato de almacenamiento Super Audio CD (SACD) utiliza un esquema de codificación sigma-delta denominado «Direct Stream Digital» (DSD), y que describiremos en el Capítulo 9. Está basado en la transmisión de una señal de un solo bit, pero a frecuencias mucho más elevadas que la de muestreo tradicional.

Modulación delta adaptativa (ADM)

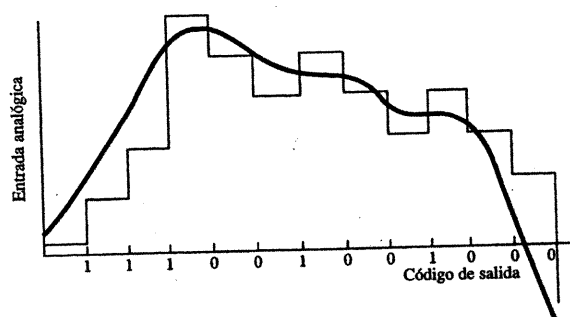
Este tipo de codificación permite variar la amplitud del paso de cuantificación, con el fin de vencer las limitaciones de la modulación delta y mantener el error de cuantificación en unos

valores razonables. El diagrama de bloques de un codificador basado en modulación delta adaptativa es mostrado en la Figura 4.22A. El codificador analiza el dato de entrada y determina cuál sería la amplitud óptima del paso de cuantificación que mejor se ajustaría al caso. Por ejemplo, con un algoritmo adaptativo sencillo; si la señal de error es una serie de datos todos positivos o todos negativos, esto indica un cambio muy rápido en la señal, por lo que debe efectuarse un cambio rápido en la aproximación. El tamaño del paso debería, por tanto, incrementarse (para intentar seguir la señal), de tal manera que cuanto más grande sea la condición de sobrecarga, mayor debería ser el tamaño del paso seleccionado. La alternancia de valores positivos y negativos es indicación de un buen seguimiento de la señal, reduciendo en este caso el sistema el paso de cuantificación para aumentar la precisión, tal y como se muestra en la Figura 4.22B. Esto produce una mejora en la relación S/R, sin necesidad de aumentar la frecuencia de muestreo o el régimen binario de transmisión. Cuantos más bits en la cadena de datos sean dedicados para diagnosticar el comportamiento de la señal, la selección del paso de cuantificación puede ser mejorada.

El diseño de un codificador ADM es complicado, sobre todo debido a que el decodificador debe estar sincronizado con el codificador para reconocer un cambio en el paso de cuantificación. Además, es difícil cambiar el paso de forma rápida y enérgica para acomodarse a los transitorios de audio. Como las señales de alta frecuencia y elevada amplitud necesitan un rápido incremen-



A Diagrama de bloques de un codificador de modulación delta adaptativa con tamaño del escalón variable



B Los cambios en el tamaño del escalón son disparados por 1 o 0 continuos

Figura 4.22. Operaciones en un codificador de modulación delta adaptativa.

to del paso, el error de cuantificación se incrementa notablemente, provocando modulación del error, con un nivel de ruido de fondo también cambiante. Además, es difícil introducir el «dither» en un sistema ADM; y puesto que el paso de cuantificación cambia, la suma de una amplitud de ruido fija (como en el método tradicional) no es efectiva. La realimentación del error puede reducir el ruido dentro de la banda. Un filtro previo de preénfasis puede reducir subjetivamente el ruido en señales de bajo nivel, enmascara la variación del ruido en el cambio del paso, reduce el ruido de baja frecuencia en amplitudes elevadas y en las señales de alta frecuencia. A medida que la pendiente de la señal de audio se incrementa, la señal de control del modulador delta (la misma señal utilizada para controlar el tamaño del escalón) eleva la frecuencia del filtro y atenúa las bajas frecuencias. Otra variante de la ADM es la modulación delta de pendiente continuamente variable («Continuously Variable Slope Delta Modulation», CVSDM) en la que el paso de cuantificación es continuamente variable y no incremental.

Compander con modulación delta predictiva

Corresponde a un esquema en el cual se opta, en vez de utilizar técnicas adaptativas, por emplear un compander. En lugar de ir variando el paso de cuantificación en función del nivel de la señal, es la amplitud de ésta la que se modifica, antes de aplicarla a un modulador delta de paso de cuantificación constante. Para reducir el umbral de cuantificación, se usa un filtro predictivo, en el cual un algoritmo utiliza el valor de las muestras anteriores para efectuar una predicción de la muestra siguiente.

La sección dedicada a la compresión/expansión consiste en un amplificador controlado digitalmente, ajustando su amplificación en función del nivel de la propia señal digital. Así, la transmisión de ésta al decodificador permite controlar la ganancia del expansor, minimizando el error de seguimiento. Los amplificadores controlados digitalmente permiten controlar el nivel de la señal sobre un gran margen dinámico, adaptando el nivel de la señal al paso constante del modulador delta. Estos sistemas incluyen un circuito detector de transitorios, que situado en el camino de la señal ayuda a «acelerar» la reducción de ganancia durante los transitorios de la señal de audio. Las cadenas de ceros o unos indican el comienzo de una sobrecarga, disparando la compresión para asegurar que los transitorios no provoquen el recorte (saturación) del modulador. La velocidad del cambio de ganancia puede ser rápida o lenta, dependiendo de la dinámica de la música. También se puede utilizar compresión espectral, acotando así las variaciones de la señal en diversas zonas del espectro; el circuito atenúa las altas frecuencias cuando el espectro de entrada contiene predominantemente estas componentes, realzándolas cuando la mayor energía se sitúa en la zona de frecuencias bajas. Con esto se consigue que el espectro en la entrada del convertidor A/D se mantenga prácticamente constante.

Modulación adaptativa diferencial (ADPCM)

La modulación adaptativa diferencial usa una codificación predictiva para conseguir la reducción de datos. El esquema ADPCM combina la modulación adaptativa diferencial ADM con la codificación binaria del PCM. Aunque el método es diferente, en la mayoría de los casos la señal diferencia (entre el valor de entrada y la predicción) a codificar es primero escalada mediante un factor de escala adaptativo y cuantificada de acuerdo a una cuantificación con una determinada curva de transferencia. El factor de escala es seleccionado de acuerdo a las propiedades de la señal; por ejemplo, el paso del cuantificador es variado proporcionalmente a la amplitud del valor medio de la señal. Las señales con variaciones grandes de nivel causan rápi-

dos cambios en los factores de escala, mientras que señales más estables provocan una adaptación más lenta. El paso de cuantificación puede ser efectivamente variado modificando directamente su magnitud, o bien escalando la señal con un factor de ganancia. Un predictor lineal, optimizado de acuerdo al tipo de señal a codificar, es utilizado para entregar en su salida una estimación (predicción) del valor de cada muestra. A la muestra de la señal de entrada se le resta el valor de la predicción, y esta señal diferencia es la que se codifica en PCM. Esta cuantificación emplea una longitud de palabra muy corta (quizás 4 u 8 bits), constituyendo la señal a transmitir. De esta forma, la señal es adaptativamente ecualizada, y el ruido de cuantificación moldeado adaptativamente para ayudar al enmascaramiento del umbral de ruido para cada aplicación en particular. Por ejemplo, el ruido puede ser moldeado (ecualizado) para que su espectro tras la decodificación sea blanco (el proceso de conformación de ruido se verá con detalle en el Capítulo 18). El decodificador realiza las mismas operaciones que el codificador; leyendo la cadena de datos selecciona el paso de cuantificación, y la señal diferencia se utiliza para la regeneración de las muestras originales. La ventaja de la ADPCM es la gran reducción en el régimen binario de la transmisión, basada en la tendencia de la señal a tener una distribución de amplitudes y un espectro muy concentrado en una región específica. Los factores de escala y otros elementos en el diseño de un algoritmo ADPCM aprovechan la ventaja que brindan estas propiedades estadísticas de la señal de audio. En aplicaciones de transmisión de la señal vocal, un régimen binario de 32 kbps (kilobits por segundo) es alcanzado con suma facilidad, utilizándose ampliamente en la industria de telecomunicación y en videojuegos.

Las prestaciones del ADPCM son muy competitivas y realmente superiores a las de un sistema PCM lineal. Cuando la señal de audio se aproxima a su máxima frecuencia, las prestaciones del ADPCM son similares a las de un esquema PCM, aunque este caso se da en raras ocasiones en la señal de audio. Por otro lado, cuando la frecuencia instantánea de la señal es relativamente baja, entonces la señal cambia muy lentamente, y los cambios de amplitud de una muestra a la siguiente son mínimos. El resultado es que la cuantificación con ADPCM entonces posee un error considerablemente menor que el PCM. En teoría, para el mismo número de bits del cuantificador, el sistema ADPCM puede aportar una mejor resolución; o dicho en otras palabras, se necesitan menos bits para obtener unas buenas prestaciones. En la práctica, un esquema de codificación ADPCM de 4 bits aporta una calidad subjetiva similar a un sistema PCM de 8 bits.

El formato CD-ROM/XA utiliza varios niveles de codificación ADPCM con el fin de permitir diferentes grados de calidad en función de la que se necesita. Un sistema ADPCM de 8 bits puede entregar una relación S/R de 90 dB en un ancho de banda de hasta 17 kHz. Hay dos niveles de 4 bits que permiten una S/R de 60 dB en anchos de banda de 17 kHz y 8,5 kHz. En la codificación la frecuencia de muestreo original de 44,1 kHz se reduce, mediante un convertidor de frecuencia, hasta 37,8 kHz o 18,9 kHz; y dependiendo de la calidad de señal seleccionada, la longitud de palabra original (16 bits PCM) se reduce a 4 u 8 bits (ADPCM). Hay cuatro filtros distintos con los que se puede realizar la predicción; su elección se efectúa en función del contenido espectral de la señal, actuando sobre bloques de 28 muestras consecutivas. Con el fin de optimizar la relación S/R instantánea, se utilizan diferentes sistemas: PCM para los pasajes de gran contenido en frecuencias altas; combinaciones de PCM diferencial de primer orden, con otros dos tipos de PCM diferencial de segundo orden en los pasajes con predominio de frecuencias medias y bajas, empleando además companders y un conformador de ruido para incrementar el rango dinámico de los diferentes esquemas. Para la correcta decodificación, los datos que describen el tipo de filtro son transmitidos en un bloque de datos especial, conjuntamente con los factores de escala o información del rango de datos. En la etapa de reproducción, los datos de los diferentes bloques ADPCM son decodificados y expandidos a los 16 bits de longitud de palabra original. Por último, la conversión D/A y el filtro recuperador completan el proceso de reproducción. Dependiendo del nivel de calidad de la señal de audio, el codificador ADPCM puede tener un régimen binario de salida entre los 80 y 309 kbps (para

cada canal de audio), con lo que se consigue una considerable reducción en el régimen binario de la transmisión.

El ADPCM con una relación de compresión 4:1 es un esquema muy utilizado en aplicaciones de ordenador; por ejemplo, el software QuickTime y los sistemas operativos Windows lo utilizan en los archivos de sonido de tipo *aiff* y *wav*, respectivamente. Además, también es un formato de datos habitual entre las diversas plataformas de videojuegos. A diferencia de los algoritmos de los sistemas de codificación perceptual, los procesos de codificación/decodificación pueden ser ejecutados con una potencia computacional muy reducida. Por ejemplo, en la versión ADPCM de la Interactive Multimedia Association (IMA) se utilizan cuatro bits para la codificación de cada muestra. La codificación es muy simple: el codificador halla la diferencia entre dos muestras consecutivas, la divide por el paso de codificación actual y el resultado es la palabra de salida. Para recrear la muestra, el codificador invierte el proceso: multiplica el valor recogido por el paso de cuantificación, y al resultado le suma la muestra anterior. El paso de cuantificación no se almacena directamente, sino a través de una tabla con 88 posibles valores, siguiendo una progresión cuasi exponencial. El decodificador toma como referencia la tabla almacenada, usando los valores previos para efectuar una actualización del escalón. Cuando la señal diferencia es pequeña, se elige un paso de cuantificación pequeño, y viceversa. Así, la codificación ADPCM puede ser efectuada por software de una forma rápida y eficiente.

Al igual que otros métodos de codificación, el ADPCM constituye una alternativa al PCM muy a tener en cuenta. Utiliza los mismos principios de muestreo, cuantificación y almacenamiento digital; sin embargo, las formas de implementar el proceso son muy distintas. En los sistemas de codificación perceptual, donde mantener el régimen binario lo más bajo posible es el objetivo principal, estas técnicas de codificación han tomado un gran auge, por lo que profundizaremos en ellas en el Capítulo 10.

CORRECCIÓN DE LA BASE DE TIEMPOS

Como ya comentamos en el Capítulo 3, los códigos de canal se emplean en el almacenamiento y transmisión de la señal. Su finalidad es la de codificar la información obteniendo una mayor eficacia en la transmisión, y la de permitir la autosincronización de los datos en recepción. Sin embargo, la recuperación de los datos queda limitada por la precisión en la base de tiempos, obtenida a partir del reloj de la señal recibida. Por ejemplo, en un magnetófono, las variaciones de velocidad en el arrastre de la cinta, la inestabilidad de la temporización en la cadena de datos de la transmisión y la falta de precisión en el oscilador empleado para temporizar los convertidores A/D y D/A, pueden degradar tanto los datos que se puede llegar a producir ruido y modificaciones en la forma de onda de la señal analógica recuperada. Con el fin de recuperar correctamente los datos, los circuitos del receptor deben minimizar los errores de base de tiempos que se producen en el soporte de la información, en el medio de transmisión o en el proceso de regeneración de la señal digital o la conversión de la misma. Por ejemplo, los lazos enganchados en fase («phase-locked loop», o abreviadamente PLL) son usados para resincronizar el receptor, mediante la señal de reloj recuperada de la transmisión. La inestabilidad en la temporización de las señales digitales, a las que se ha superpuesto el ruido e interferencias captadas, se erige como el obstáculo más importante para la recuperación de la señal en los entornos digitales. Por ejemplo, una conversión de 20 bits requiere una precisión en la base de tiempos mucho más elevada que una conversión de 16 bits. Además, la precisión en la temporización es problemática en sí misma, ya que requiere unas tolerancias difíciles de mantener. Por ejemplo, un reloj podría requerir una precisión mejor que 20 ps; y 1 ps (10^{-12} s) es el período de una señal de 1 THz ($1 \text{ THz} = 1.000 \text{ Ghz} = 1.000.000 \text{ MHz}$), magnitud poco habitual en la circuitería genérica de audio.

Inestabilidad temporal («jitter»)

A la variación en el eje del tiempo de una señal digital se le denomina «jitter». En particular, es la variación en el tiempo de una señal de reloj, medida con respecto a una señal de referencia ideal.

Por ejemplo, el jitter se manifiesta como las desviaciones en los instantes del cruce por cero de la forma de onda de la señal. Las variaciones temporales en una señal analógica son perfectamente audibles, ya que se manifiestan como inestabilidad en el tono de la señal. Sin embargo, el «jitter» de una señal digital puede provocar errores de bit en la cadena de datos; o hacerse audible de forma indirecta como incremento del ruido y distorsión de la forma de onda de salida; o si la inestabilidad es corregida («de-jittered»), puede que no provoque errores de bit o defectos audibles. El «jitter» está siempre presente, y sus efectos y la tolerancia requerida dependen del eslabón en el que aparezca dentro de la cadena de tratamiento de la señal. Niveles relativamente elevados de «jitter» pueden aparecer en la transmisión de señal entre dispositivos, sin que tengan repercusión alguna sobre la señal. Sin embargo, en los procesos de conversión, incluso niveles de «jitter» extremadamente bajos son capaces de producir errores en la forma de onda resultante.

El «jitter» se manifiesta como variaciones en los instantes de paso por cero de la señal, como se muestra claramente en la Figura 4.23. La variación instantánea del período de la señal alrededor de cada transición ideal aparece como una indeterminación en el instante de llegada de la señal, y a esta variación se le denomina valor pico a pico de la incertidumbre («peak-to-peak jitter»). El «jitter» también se produce en los medios de almacenamiento, canales de transmisión o circuitos de procesado y regeneración de señal como convertidores A/D y D/A. El «jitter» puede ser de naturaleza aleatoria (como las variaciones en los relojes de los circuitos digitales), puede ser función de la anchura de los pulsos de reloj («FM jitter»), puede estar relacionado con otros eventos (correlado con otras señales presentes en la circuitería), o incluso a veces puede tener una naturaleza periódica.

El «jitter» puede ser descrito por sus características espectrales, mostrando la amplitud y variación de frecuencia de la señal. El «jitter» aleatorio muestra un espectro de banda ancha; y cuando se reconstruyen los datos para obtener la señal analógica, se manifiesta como un incremento en el ruido de fondo. El espectro del «jitter» periódico es una simple línea espectral, apareciendo las bandas laterales o modulación del ruido espaciadas a cada lado de la frecuencia de la señal. El «jitter» a frecuencias inferiores a la de muestreo provoca una acumulación del error en la temporización de la señal, dependiendo la magnitud del error de la amplitud y frecuencia de la forma de onda de la señal moduladora. Generalmente, una medida correcta del «jitter» es la del «valor pico a pico» de la desviación, aunque para un «jitter» aleatorio es mejor medir su valor eficaz (rms). También debe tenerse precaución especificando la medida

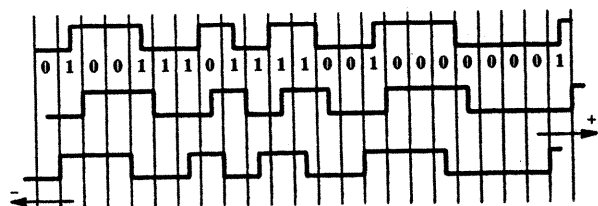


Figura 4.23. Una variación de la temporización de la señal en el medio, la regeneración, o los circuitos de procesado, pueden provocar un «jitter» en la base de tiempos de la señal.

del «jitter». Por ejemplo, si se mide la desviación de una señal de reloj realizando un promedio en el tiempo, el valor medio va siempre a converger a cero, con lo que el resultado es una buena forma de no medir el «jitter».

Diagrama de ojos

Un osciloscopio, sincronizado con el período de bit de una señal digital, puede visualizar tanto las variaciones de amplitud de la forma de onda de la señal como el «jitter», desplazamiento temporal de las transiciones de la señal digital. En la pantalla del osciloscopio, además pueden observarse distintas anomalías asociadas a la evolución de la señal en el tiempo como nivel de continua, deformaciones de la señal digital, etc. La visualización de forma superpuesta de las sucesivas transiciones de la señal se denomina diagrama de ojos, y puede utilizarse para la determinación de la calidad de la señal recibida. Además, también puede utilizarse para establecer el éxito en la regeneración de la señal digital. El ruido que introduce el canal tiende a cerrar el diagrama de ojos verticalmente (en amplitud), aproximando los niveles altos y bajos, mientras que el «jitter» lo cierra horizontalmente (en el tiempo) tal y como se muestra en la Figura 4.24. La suma de ruido y «jitter» actuando sobre la señal digital pueden llegar a degradar la forma de onda hasta el punto de no poder recuperar la señal con precisión. Un indicador del deterioro sufrido por la señal se obtiene midiendo la extensión de las variaciones de amplitud, denominando relación de apertura a la razón de amplitudes exteriores e interiores del ojo:

$$E = \frac{a_2}{a_1} = \frac{(a_1 - 2 \cdot \Delta a)}{a_1}$$

siendo:

a_1 = amplitud exterior del ojo.

a_2 = amplitud interior del ojo.

Δa = variación de la amplitud.

Ambas amplitudes se pueden medir entre picos, o desde el centro de apertura del ojo (en cuyo caso desaparece el 2 multiplicando a Δa en la expresión anterior). La amplitud del ojo se mide en voltios, mientras que su anchura se expresa como un porcentaje del período de la celda,

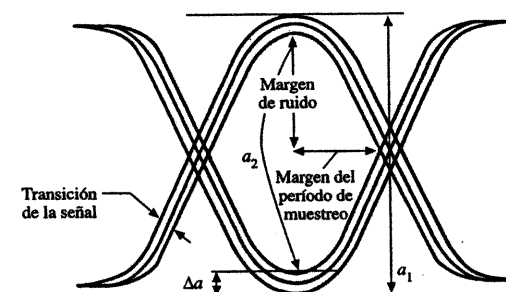


Figura 4.24. El diagrama de ojos puede usarse para interpretar la calidad de una señal de datos RF. El mínimo de la abertura del ojo determina el límite utilizable de la señal.

o como fracción del período de bit. La anchura del ojo nos da la fracción (porcentaje) del período de bit disponible para establecer el valor lógico de la señal, mientras que la altura del ojo muestra la máxima diferencia entre niveles obtenidos en el período de un bit. Generalmente, el receptor es capaz de interpretar correctamente el nivel de la señal mientras: se mantenga un mínimo de apertura del ojo (amplitud), durante al menos una fracción determinada del período de bit (tiempo); indicando el centro de apertura del ojo, el instante⁴ y umbral de toma de decisión de la señal digital. Sin embargo, el «jitter» que posee el propio reloj de los circuitos de recuperación de señal del receptor puede provocar que los instantes de decisión no coincidan con el centro del período de bit.

Cuando se observa el «jitter» en un osciloscopio se aprecian las variaciones dinámicas de la señal, pero otro método más preciso permite obtener información detallada, aplicando la señal digital a un demodulador de FM y su salida (variaciones del período de bit) a un analizador de espectro. El «jitter» aleatorio se muestra entonces como un espectro de banda ancha, elevando el nivel de ruido en la señal analógica reconstruida. El «jitter» periódico se observa como una simple raya espectral, situada en muy baja frecuencia cuando la variación del reloj es lenta, y en la parte alta del espectro cuando las variaciones del reloj son rápidas. Lo que siempre ocurre es que la señal analógica recuperada aparece con el ruido de fondo modulado en amplitud o bien la aparición de bandas laterales de la señal, o ambas alteraciones a la vez.

«Jitter» en el muestreo y en la conexión entre equipos

La trascendencia del «jitter» depende básicamente del punto de la cadena en el que se introduzca. Por ejemplo, es muy importante distinguir entre el «jitter» que aparece en la interconexión entre equipos (que afecta a una transferencia digital de datos) y el «jitter» del muestreo (cuando se convierte la señal analógica a digital), ya que los errores que provocan son de naturaleza completamente diferente. El «jitter» en la interconexión influye en la temporización (periodicidad bit a bit) de los datos enviados. Es preocupante cuando su magnitud es tal que provoca errores en la señal recuperada, por lo que la calidad de los datos recibidos debe ser monitorada por los circuitos de detección de errores⁵ del receptor. En la mayoría de las cadenas de datos el código de canal utilizado permite regenerar el reloj; el receptor recupera el reloj de la cadena de datos que recibe, y se sincroniza con el reloj transmitido. Un circuito receptor con un reloj fijo no sería capaz de engancharse a una señal con ligeras variaciones en su temporización ni tan siquiera cuando la velocidad de transmisión de los datos fuera exactamente igual en emisor y receptor (debido a posibles diferencias de fase entre los mismos). Por este motivo, los circuitos de recepción utilizan lazos enganchados en fase (PLL) para poder «alinearse» los relojes internos del equipo con la secuencia de datos recibida. En la Figura 4.25 se muestra el diagrama de bloques de un PLL en recepción. Toma la señal de entrada como referencia de tiempo, mide la diferencia de fase (señal de error) entre la señal de entrada y su propia salida y usa esta señal de error para controlar un oscilador controlado por tensión (VCO, «voltage-controlled oscillator») que aparece dentro del lazo. La respuesta del VCO ante la señal de error es modificar ligeramente su frecuencia de oscilación en el sentido de igualar ambas señales, tendiendo a que la señal de error sea mínima. Una vez que el VCO se «engancha» a la fase de la señal de

⁴ Nota del traductor. El instante de decisión es aquel en el que los circuitos de entrada del receptor validan la señal, decidiendo en función del nivel si el estado asociado a ese instante corresponde a un estado alto o bajo. Este instante de decisión se suele tomar en el centro del bit, ya que es cuando la señal tiene mayor amplitud y el ruido y «jitter» le afectan en menor medida. Lógicamente, la periodicidad de los instantes de decisión coincide con el período de bit de la señal.

⁵ Nota del traductor. Se refiere al mecanismo de detección de errores desde el punto de vista de transmisión/recepción de datos e interpretación correcta de niveles en la señal digital recibida, que nada tiene que ver con el sistema de corrección de errores de la señal de audio.

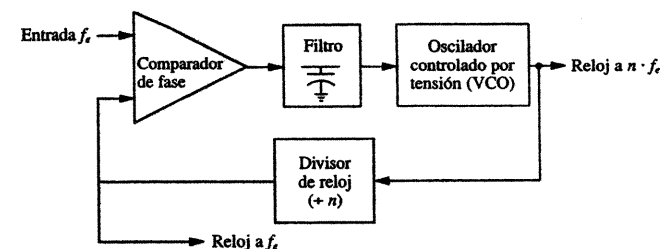


Figura 4.25. Un ejemplo de lazo enganchado de fase (PLL) de un interfaz. El oscilador evita que el «jitter» pase a través del PLL a la salida de los datos o a la salida del reloj.

entrada, el oscilador corre a la frecuencia de referencia o a un múltiplo de ella. El control del oscilador se desacopla del comparador de fase a través de un filtro paso bajo, que eliminando las rápidas variaciones de fase reduce el «jitter» de alta frecuencia en la salida de reloj del PLL. Mediante esta estructura, un PLL puede reducir el «jitter» resincronizando las señales de reloj a una base de tiempos común, estable y precisa. Por el contrario, cualquier ganancia próxima a la frecuencia de corte del filtro paso-bajo del PLL puede incrementar el «jitter».

Si los datos recuperados están libres de errores es que el «jitter» no les ha afectado. Sin embargo, si posteriores resincronizaciones no eliminan el «jitter» de la cadena de datos, es posible que se provoquen defectos audibles en el convertidor D/A.

El «jitter» en el muestreo puede afectar notablemente la calidad de la señal de audio tanto en el muestreo como en el cambio de frecuencia de muestreo. Este «jitter» puede ser provocado por el reloj de muestreo, afectando la calidad de la señal de audio añadiendo ruido y distorsión. Las tolerancias en esta temporización son muy pequeñas, por lo que estos errores deben ser minimizados. Los circuitos de generación de reloj son utilizados en los dispositivos que van a realizar el muestreo de la señal. En los dispositivos que realizan el muestreo de la señal, los circuitos que generan el reloj de muestreo deberían poder obtener su reloj de referencia de la propia señal de entrada del interfaz de transmisión de datos. Así, un PLL se necesitaría para eliminar el «jitter» del interfaz y obtener un reloj de muestreo cuya pureza evite productos de modulación audibles. El «jitter» puede también aparecer a lo largo de la cadena de señal, por lo que deben tomarse precauciones en cada etapa para desacoplarlas entre sí, y que los datos pasen de una a otra sin error. Este hecho es particularmente importante en una cadena de equipos interconectados, puesto que el «jitter» podría irse acumulando a lo largo de las diferentes etapas. Cada dispositivo puede contribuir con una pequeña cantidad de «jitter», pudiendo abocar al conjunto a la aparición de errores en la detección de datos o problemas en la conversión. A pesar de todo, una señal con graves problemas de «jitter» siempre puede ser regenerada temporizando los datos con su frecuencia y fase correcta, por lo que esta acumulación de «jitter» puede ser convenientemente aminorada, manteniendo el «jitter» dentro de los márgenes de tolerancia deseados.

El «jitter» en los soportes de almacenamiento

El control del «jitter» dentro de la cadena digital de audio debe empezar en el medio de almacenamiento de la señal. Los soportes de información como cintas magnéticas y discos ópticos introducen errores de base de tiempos en sus datos de salida, debidos a las variaciones de velocidad que se producen en los sistemas de arrastre. Con el fin de limitar las variaciones de ve-

locidad, se deben diseñar relojes de precisión y sistemas servocontrolados que controlen los sistemas de arrastre, amén del empleo de memorias tampón («buffers») en la entrada y salida de datos, de forma que sean capaces de absorber las irregularidades del flujo de datos. Debido a que las cintas magnéticas son un medio flexible, son más propensas a provocar «jitter» que los discos ópticos, aunque ambos necesitan sistemas específicos de control. Las variaciones de velocidad en el transporte, originadas por la excentricidad del cabezante («capstan») o de los ejes de los motores, pueden ocasionar que el flujo de datos no se mantenga constante, pudiendo la velocidad fluctuar lenta o rápidamente sobre la velocidad nominal. Si la magnitud de la variación se encuentra dentro de la tolerancia del equipo, los datos almacenados pueden ser recuperados de forma efectiva, por lo que no se introducen errores en la señal. Los circuitos servocontrolados se utilizan con el fin de leer la temporización (velocidad de los datos) con la que se recupera la señal, y con esta temporización generar una señal de control que corrija la velocidad del transporte. En la mayoría de los sistemas, para el servocontrol se utiliza un circuito basado en un PLL, como muestra la Figura 4.26. El control de la velocidad se obtiene mediante el PLL, comparando las palabras de sincronización en la salida de datos del sistema de lectura (que sabemos con el código de canal que aparecen a una velocidad prefijada) con una señal de referencia (interna del equipo). La diferencia entre las dos señales se emplea para servocontrolar la velocidad de giro del motor, minimizando de forma dinámica la diferencia entre las dos velocidades del equipo: la teórica (dada por la referencia) y la instantánea real (obtenida de la cadena de datos). Además, puede introducirse un sistema de control fino de la velocidad usando un segundo PLL; por ejemplo, con el fin de mantener la velocidad lineal constante en un reproductor de disco compacto.

Aunque los servosistemas basados en PLL mantengan la velocidad del soporte en unos valores constantes y con gran precisión, los errores de temporización (base de tiempos) de los datos aún persisten. Así, el «jitter» también estará presente en los datos cuando se transmitan, bien sea a través de un protocolo de interconexión eléctrico u óptico. Para minimizar el efecto de la

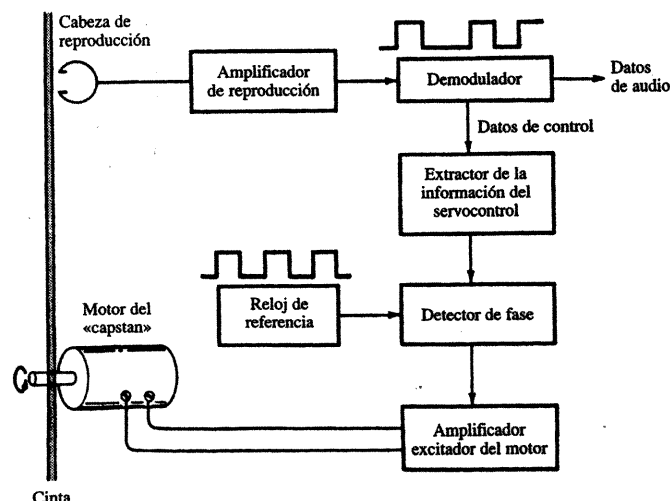


Figura 4.26. Un ejemplo del lazo extractor/controlador del motor de arrastre de la cinta.

variación en la temporización de los datos, se regenera su temporización mediante una memoria tampón («buffer»). Esta memoria aumenta el margen de captura⁶ de una cadena de datos; por ejemplo, el «buffer» puede absorber variaciones de la frecuencia de muestreo en una cadena de conversión, minimizando la pérdida de palabras. Sin embargo, aumentar la longitud del «buffer» provoca un aumento del retardo que sufre una señal cuando pasa a través de un equipo, hecho poco deseable en algunas aplicaciones. Hay que tener en cuenta que retardos del orden de unos pocos milisegundos pueden ser problemáticos en aplicaciones en tiempo real, como en radiodifusión, donde se necesita realizar un monitoreo tanto de la señal en antena como la de los equipos del estudio. Un «buffer» puede ser diseñado usando memoria RAM, de forma que el desbordamiento periódico de su puntero («overflows») convierta la memoria RAM en un «buffer» circular. Naturalmente, el reloj que controla la lectura de datos de la memoria debe ser independiente del reloj de entrada de datos; y la cantidad de datos que hay en el «buffer» en un momento dado⁷ puede ser usada para controlar la velocidad del transporte. Por ejemplo, la diferencia entre los punteros de escritura y lectura, en función de la capacidad de la memoria, puede ser convertida en una señal analógica para el servocontrol de velocidad. Si el nivel de llenado del «buffer» se aleja de la condición ideal (llena la mitad de la capacidad), se ordena al servo bien elevar o bajar la velocidad del transporte, y de esta forma los datos de audio nunca ni desbordarán ni vaciarán el «buffer».

Otra alternativa de implementación del «buffer» es mediante una memoria FIFO («First In First Out», primero en entrar primero en salir). Los datos de entrada son introducidos en lo alto de la memoria temporizados con la velocidad con la que son recuperados del medio y los datos de salida se recuperan de la parte inferior de la memoria con la temporización correcta. Además de su aplicación como mecanismo para la reducción del «jitter», estos métodos de corrección de la temporización (base de tiempos) se emplean en aquellos sistemas de almacenamiento con un proceso de lectura/escritura de datos no continuo, como sucede en los sistemas de almacenamiento en disco duro. El «buffer» debe comenzar su llenado cuando se lee un sector del disco y mantener siempre de forma continua la salida de datos entre ciclo y ciclo de lectura. Además, mientras se escribe en la memoria, debe mantenerse un flujo continuo de datos hacia el disco, para completar la grabación de cada sector. Con todo lo dicho, hay que precisar que cuando el «jitter» es considerablemente inferior al período de bit, no es necesaria la utilización de un «buffer».

El «jitter» en la transmisión de datos

El «jitter» debido (o que aparece) en un interfaz de conexión entre equipos debe ser minimizado en la transmisión de datos. Independientemente de los errores de «jitter» introducidos por el circuito transmisor y el cable, el receptor tiene dos tareas: recuperar los datos y recuperar el reloj. El «jitter» en la señal puede afectar a la recuperación de ambos, pero el efecto del «jitter» depende de la aplicación. Cuando los datos son transferidos pero no pueden ser regenerados (convertidos a analógicos) en el receptor, únicamente es necesaria la recuperación de los datos. El «jitter» del interfaz es únicamente uno de los factores, si éste provoca errores de datos en la recepción. La tolerancia que puede tener el «jitter» es relativamente baja; por ejemplo, es posible la recuperación de la señal sin errores con valores de «jitter» del orden de 5 a 10 ns. Sin embargo, cuando los datos van a ser regenerados o reuanticificados, es necesaria la recuperación

⁶ Nota del traductor. El margen de captura es el rango máximo de variación en la temporización de los datos que se puede dar sin pérdida de señal. Se suele dar como una fracción del período de bit de la señal.

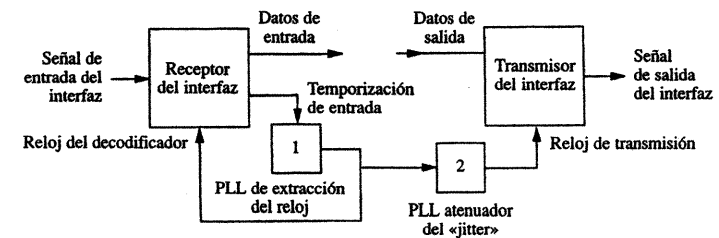
⁷ Nota del traductor. La cantidad de datos que en un instante hay en el «buffer» es función de la diferencia entre la velocidad teórica nominal del sistema y la real en ese instante, con lo que es un buen indicativo de la corrección de velocidad que hay que introducir.

de los datos y particularmente de su reloj (temporización). Elevados niveles de «jitter» pueden comprometer la capacidad del receptor para obtener (de la propia señal) un reloj lo suficientemente estable como el que se necesita para la conversión, ya que aunque depende del diseño electrónico del convertidor D/A, se requieren niveles de «jitter» tan bajos como 20 ps.

Por ejemplo, cuando se transfieren los datos de un reproductor de CD a un grabador DAT, o a una estación de trabajo, sólo el «jitter» introducido por el interfaz es importante para la recuperación de los datos⁸, haciendo necesaria su atenuación sólo en el caso de que pudieran aparecer errores en los datos. Sin embargo, cuando los datos se utilizan para la conversión de la señal en analógica, la atenuación del «jitter» se vuelve imprescindible. El «jitter» que en recepción aparece en la temporización de los datos perjudica notablemente el proceso de recuperación de la señal analógica. El problema aparece no por la recuperación de los datos en sí, sino porque puede comprometer la capacidad del receptor para, a partir de la propia cadena de datos, derivar un reloj lo suficientemente estable como para servir luego en el proceso de conversión D/A.

Como ya se ha comentado, el PLL del receptor separa el reloj recibido de los datos en sí, empleando el propio reloj para recuperar los datos, regenerar el reloj (atenuando el «jitter») y emplearlo como base de tiempos para «resincronizar» los datos que se reciben (véase Fig. 4.25). En algunos diseños, el receptor debe extraer la señal de sincronización de los propios datos que se están recibiendo del interfaz, ponerlos en una memoria intermedia («buffer») y, una vez regenerado el reloj, dar salida a los datos con la frecuencia y fase requerida por el equipo destino de los mismos. El «buffer» debe ser lo suficientemente largo como para prevenir su desbordamiento, tanto en su posible vaciamiento como llenado. En el primer caso, y al ser un problema de falta de muestras, se van repitiendo los valores que estaban en el «buffer» (o el último de ellos o preferiblemente muestras nulas en los silencios de programa) hasta recuperar el flujo de datos. En el segundo caso, el desbordamiento se produce por un exceso en la velocidad de llenado del «buffer», y antes de que se desborde se tienen que ir eliminando muestras en los vacíos de señal. El método de sincronización entre equipos a partir del reloj embebido en la propia señal transmitida se denomina «genlock», siendo un sistema que trabaja muy bien en la transmisión entre equipos punto a punto. Cuando la frecuencia de muestreo en el equipo que transmite los datos es distinta de la del equipo local, se necesita un convertidor de frecuencia de muestreo, cuya finalidad es generar una nueva temporización sobre los datos entrantes para adecuarlos a la frecuencia de muestreo del equipo local (este proceso se describirá en el Capítulo 13). Los equipos convertidores de frecuencia de muestreo también se pueden utilizar para la disminución de «jitter», sobre todo en los casos en los que éste es de magnitud elevada.

Muchos receptores utilizan para la regeneración de los datos esquemas con un doble reloj, tal y como se representa en la Figura 4.27. El primer paso consiste en la extracción del reloj. Se recupera el reloj embebido en los datos recibidos y con éste se sincroniza perfectamente la recepción de datos, de forma que el «jitter» de la señal no influya en la recuperación de los datos. Para ello, un PLL inicial utiliza las transiciones de los datos como referencia, diseñándose el PLL para seguir perfectamente la señal, incluido el «jitter» con el que llega, por lo que no introduce ninguna atenuación de éste ni es necesario un reloj con gran precisión. Los datos así recuperados se almacenan en un «buffer» o una memoria FIFO, aunque no sería necesario si el «jitter» es inferior al período de bit de los datos. El segundo paso consiste en la atenuación del «jitter». Ahora se necesita un PLL con un «jitter» muy bajo, que temporeice la salida de los datos del «buffer» regenerando una cadena de datos con la precisión (de reloj) necesaria. Como se puede apreciar, el segundo PLL no se ha diseñado para «seguir» el «jitter» de los datos entrantes, sino para que se sincronice al reloj de los datos recibidos (tomando la señal de la extracción del primer PLL) y su temporización de los datos se realice con precisión y un



Nota: Los relojes de la decodificación y la salida de datos son diferentes.

Figura 4.27. Un ejemplo de arquitectura de un transceptor (receptor/emisor) de doble reloj, con extractor de reloj y atenuación del «jitter».

«jitter» mínimo (por tanto, sirviendo para su atenuación). Con este sistema, el receptor desacopla los relojes de recepción de datos con los internos de retransmisión o conversión, ya que de otra manera el «jitter» de recepción degradaría notablemente la señal.

La transmisión y generación de «jitter» dependen a menudo del cable utilizado en la interconexión. Hay un tipo de «jitter» dependiente de la señal cuya magnitud es proporcional a los estados (alto y bajo) que toma la señal que se transmite. Por ejemplo, el código de canal de un protocolo de comunicación específica que: tiene que haber un cambio de nivel entre un bit y el siguiente, que transmitir un cero significa un no cambio de nivel dentro del período de bit, y que transmitir un uno implica, además, un cambio de nivel en el centro del período de bit. Con estas premisas, transmitir una cadena de unos impone el doble de transiciones que transmitir una cadena de ceros; si las pérdidas de señal (y el retardo) crecen con la frecuencia, el «jitter» no será el mismo cuando se transmitan largas cadenas de ceros que cuando son cadenas de unos. Otro ejemplo: el «jitter» puede ser inferior a 1 ns si el cable tiene un ancho de banda a 3 dB de más de 4 MHz, pero se incrementa hasta 9 ns cuando el ancho de banda del cable se reduce a la mitad. Por esta razón, muchos protocolos de comunicación utilizan para su sincronización un patrón de bits fijo (o preámbulo), y ya que este patrón es siempre el mismo, se evita una causa de «jitter» que es que la sincronización dependa de los estados que puede tomar la señal. Sin embargo, el «jitter» dependiente del patrón de la señal («jitter» correlado con la señal) puede ser generado por otros factores, como, por ejemplo, que dependa de la polaridad y frecuencia de la señal de audio. Cuando la señal codificada es un tono puro de bajo nivel y frecuencia, los bits más significativos cambian todos al unísono reflejando un cambio de polaridad, por lo que aparece «jitter» dependiente del valor de la señal. Cualquier formato de comunicación serie que no altere la secuencialidad de los datos es propenso a sufrir este fenómeno, mientras que introduciendo alguna alteración en la secuencialidad, el «jitter» podría ser descorrelado con la señal, con lo que sería mucho más benigno.

En instalaciones con un gran número de equipos digitales, la protección contra el «jitter» y sus acumulaciones es mucho más compleja de llevar a cabo. Los datos deben transmitirse con precisión entre elementos de la instalación que tienen la misma frecuencia de muestreo, debiendo tener cada equipo «enganchado» su reloj con el reloj principal («master clock») mediante una configuración en estrella. Todos los equipos deberían tener entradas externas de reloj. La utilización de un reloj externo por parte de los equipos se denomina sincronización por reloj maestro, y es el método recomendado cuando el número de equipos a interconectar es elevado. Los elementos y problemas derivados de la sincronización entre equipos se tratarán con detalle en el Capítulo 13.

⁸ Nota del traductor. En este caso no se necesita recuperar la temporización a partir de los datos porque los destinatarios de la señal trabajan con su propia frecuencia de muestreo.

«Jitter» en convertidores

Aunque el «jitter» debe ser controlado a lo largo de todas las etapas de la cadena, donde es más crítico es en las etapas de conversión. Los relojes utilizados tanto en la conversión A/D como en la D/A deben poseer una gran estabilidad, ya que el «jitter» siempre va a provocar la degradación de la forma de onda de la señal analógica de salida. Es sencillo de imaginar: piense que el «jitter» en el reloj de un convertidor A/D provoca la salida de una palabra errónea en el instante incorrecto. Y lo que es peor: si esas palabras se aplican a un convertidor D/A sin «jitter», el resultado es que muestras de señal portando amplitudes erróneas saldrán en los instantes correctos. El «jitter» es más crítico en el reloj de un convertidor A/D. Los osciladores de cristal poseen un «jitter» inferior a 10 ps rms, debiéndose usar como referencia dentro de los sistemas digitales complejos. Por otro lado, es esencial en este tipo de circuitería el buen diseño de las pistas del circuito impreso así como la distribución cuidadosa de los componentes en la placa, para evitar capacidades parásitas que siempre son adversas.

Los efectos del «jitter» en el reloj de la etapa de muestreo de un convertidor A/D son parecidos a los de una modulación FM: la señal de entrada actúa como frecuencia portadora y el «jitter» del reloj como señal moduladora. El «jitter» periódico de baja frecuencia reduce la amplitud de la señal de entrada y le añade unas bandas laterales a cada lado de la señal de entrada equiespaciadas en frecuencia y con una separación múltiplo de la frecuencia del «jitter». A medida que el «jitter» crece, la amplitud de las bandas laterales se eleva, y lógicamente sus efectos se vuelven más perniciosos. Los efectos del «jitter» se acentúan a medida que la frecuencia de la señal de entrada crece, aumentando la magnitud de error cuando crece la pendiente o la velocidad de subida («slew-rate») de la forma de onda de la señal analógica. En el proceso de conversión A/D, el «jitter» no debe interferir con el correcto muestreo de la señal, por lo que el error temporal cometido debe provocar en la señal un error de amplitud inferior a un LSB (paso de cuantificación). Un «jitter» aleatorio de 2 ns aplicado a un convertidor de aproximaciones sucesivas de 16 bits puede degradar su rango dinámico teórico pasando de 98 a 91 dB, como se muestra en la Figura 4.28.

La precisión requerida en la temporización de la conversión A/D es muy elevada. La máxima velocidad de cambio de una señal sinusoidal aparece en los pasos por cero de la señal, siendo de la forma $2 \cdot \pi \cdot A \cdot f$, donde A es la amplitud de pico de la señal y f la frecuencia en Hz. Haciendo una sencilla estimación, un «jitter» de 250 ps permitiría la conversión A/D con una precisión de 16 bits de una señal a fondo de escala y 20 kHz de frecuencia sólo cuando las componentes del «jitter» estuvieran por debajo del ruido de cuantificación. Un «jitter» de pico inferior a 400 ps podría provocar chasquidos que disminuyeran el margen dinámico en 0,5 dB. Steve Harris ha demostrado que los convertidores sigma-delta por sobremuestreo son, al igual que los tradicionales convertidores de aproximaciones sucesivas, susceptibles al «jitter» sinusoidal del reloj. Los convertidores sigma-delta son menos susceptibles al «jitter» aleatorio, ya que las componentes que aparecen de este «jitter» se extienden en todo el margen de frecuencias del sobremuestreo, para posteriormente reducir el ancho de banda al de audio mediante un filtro paso-bajo.

Los convertidores D/A son también susceptibles al «jitter». Aunque las muestras se hayan tomado en un convertidor A/D libre de «jitter», si el reloj que temporiza la salida de las muestras en el D/A posee «jitter», la señal de salida estará formada por una serie de muestras perfectas en instantes erróneos. A pesar de que los valores numéricos de las muestras sean correctos, las desviaciones temporales ocasionadas por el «jitter» provocarán en la señal analógica un incremento en el ruido y la distorsión. Afortunadamente, la distorsión en la salida de señal sólo es problemática en la reproducción; los datos en sí mismos son correctos, y sólo necesitan una temporización de salida correcta para recuperar la señal original. No todos los receptores de datos digitales (tal y como algunos receptores S/PDIF que se verán en el Capítulo 13) tienen el «jitter» lo suficientemente bajo. Como ya se ha comentado, mediante

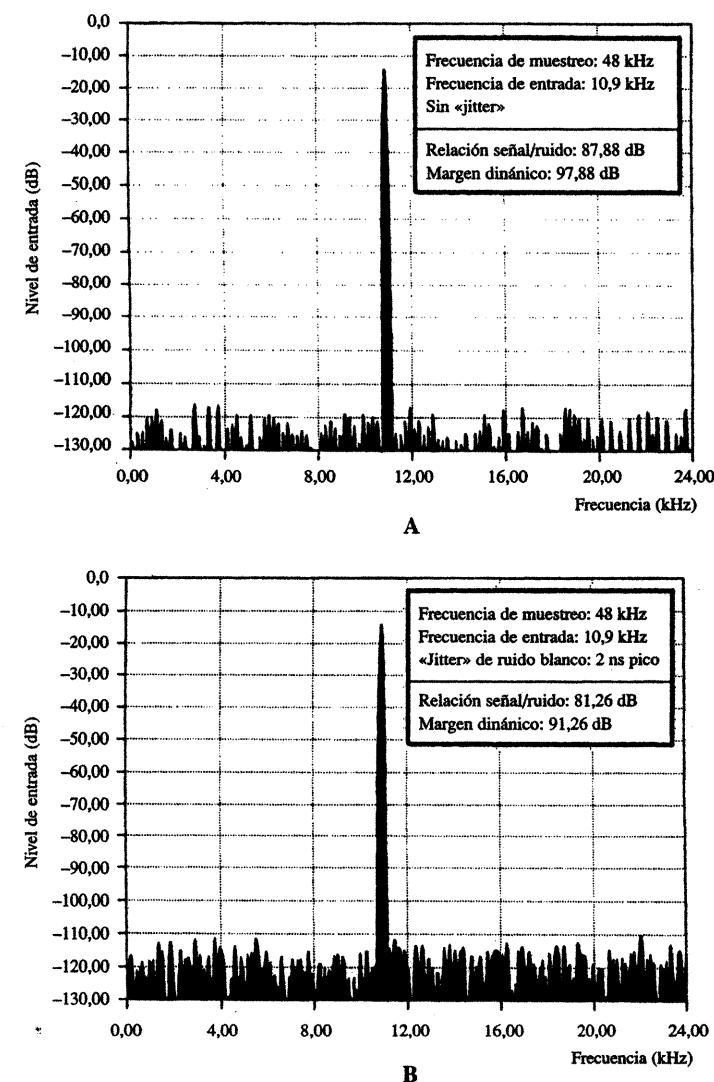


Figura 4.28. Simulación mostrando el espectro de salida de una conversión A/D sin sobremuestreo. A. Sin «jitter» en el reloj. B. Con un «jitter» de espectro aleatorio y continuo de 2 ns de valor de pico (Harris).

circuitos basados en PLL o más sofisticados, los datos que provienen de un protocolo de interconexión o de un equipo electrónico deben ser resincronizados a un nuevo reloj. La finalidad es eliminar el «jitter» con el que se reciben y preparar los datos con vistas a la conversión D/A de la señal.

El efecto del «jitter» en la salida de un convertidor D/A con red de resistencias en escalera puede ser observado restando a la señal escalonada de la salida del convertidor (afectada con «jitter») la misma señal ideal (sin «jitter»). La señal diferencia contiene los picos correspondientes a las diferencias de temporización entre las muestras con la temporización real y la ideal (ya que en esto consiste el «jitter»). De estos picos, su anchura es proporcional a las diferencias en el instante de llegada entre la señal real y la simulada (ideal), mientras que su amplitud corresponde a la diferencia entre la tensión de una muestra y la de la siguiente, con lo que una mayor amplitud denota un error mayor. La señal de audio actúa modulando la señal de error, produciendo los valores mayores en los picos de la señal. El espectro de la señal de error es uniforme (como un ruido blanco), puesto que no hay relación estadística entre los diferentes valores de la señal. La amplitud es función del tamaño del escalón, o más estrictamente, del valor absoluto de la pendiente media de la señal (variación de la señal). Así, el caso peor para un «jitter» aleatorio en un convertidor de resistencias en escalera aparece para una señal con un nivel a fondo de escala y de frecuencia la mitad de la de muestreo. Dependiendo del diseño del convertidor (y otros factores adicionales), un valor del «jitter» inferior a 1 ns es el mínimo necesario para obtener las prestaciones correspondientes a un convertidor de 16 bits, aunque lo que sería deseable es manejar tolerancias de al menos la mitad, 500 ps. Desafortunadamente, la mayoría de los equipos de consumo incluyen relojes de baja estabilidad, con lo que el «jitter» puede provocar chasquidos con un valor de pico de -70 o -80 dB con respecto al nivel máximo. Cuando se utiliza un filtro de sobremuestreo como etapa previa al convertidor en escalera, la sensibilidad al «jitter» aleatorio se reduce en proporción al factor de sobremuestreo empleado. Por ejemplo, un convertidor D/A con un sobremuestreo octuple es cuatro veces menos sensible al «jitter» que el mismo convertidor con un sobremuestreo doble. Por otro lado, el «jitter» de baja frecuencia que aparece correlado con la señal no es afectado por el sobremuestreo, ya que su proceso es diferente.

Los convertidores D/A sigma-delta, tratados con detalle en el Capítulo 18, pueden ser más o menos sensibles al «jitter» dependiendo de su arquitectura. Cuando la salida del convertidor es de un solo bit, los pulsos de error provocados por el «jitter» son todos de la misma amplitud. En un convertidor de un solo bit en el que la salida es aplicada a un filtro analógico, el «jitter» aleatorio es independiente de la señal, y de hecho los pulsos del «jitter» siguen saliendo del convertidor aunque no haya señal presente. En convertidores de un solo bit y con filtro recuperador analógico, para no degradar las prestaciones de un sistema de 16 bits se requiere un «jitter» inferior a los 20 ps de nivel de pico. Como apunta Robert Adams, esto se debe a que la modulación de fase provoca que el ruido, conformado para caer fuera de la banda de audio, ahora lo hace en ella, incrementando el nivel de ruido dentro de la banda de nuestro interés. Algunos convertidores de un solo bit utilizan un filtro recuperador de condensadores conmutados, y puesto que el condensador de la salida establece el nivel independientemente de cuándo aparece el flanco de la señal de reloj, este tipo de etapa de salida posee una menor sensibilidad a los errores del «jitter». Sus prestaciones resultan similares a las de un convertidor de resistencias en escalera trabajando con sobremuestreo; sin embargo, al final se debe eliminar todo el ruido fuera de la banda de audio⁹. Los convertidores multibit son, generalmente, menos susceptibles a los errores de «jitter» que los convertidores de un solo bit. El motivo se deriva del empleo de múltiples niveles de cuantificación, ya que los errores de fase en los relojes empleados para generar la temporización introducen un menor error que el que aparece en los moduladores sigma-delta.

A menudo, numerosos audiófilos han informado de las diferencias audibles que existen entre diferentes cables digitales. En un principio podrían ser atribuidas a los convertidores D/A, ya que podrían introducir un diseño deficiente en lo que se refiere a la recuperación del reloj (lo suficientemente estable) a partir de la cadena serie de bits de su entrada. Pero un convertidor D/A bien diseñado y con un reloj estable será inmune a las variaciones que sufra la cadena digital de datos en su camino, mientras no se alteren los valores de los datos en sí mismos.

El «jitter» debe ser controlado en cada etapa de la cadena digital de audio. Aunque los diseñadores deben controlar y medir el «jitter» especialmente en los relojes en los circuitos, las medidas analógicas tradicionales, como la THD + N y el análisis espectral, utilizados para evaluar la calidad de la señal de salida, pueden incluir efectos provocados por el «jitter». Por ejemplo, si la THD + N medida a 20 kHz y 0 dB es menor que la THD + N medida a 1 kHz, ello es debido a que el «jitter» del convertidor no es significativo. De hecho, medidas como la THD + N pueden evaluar el «jitter» de forma más adecuada que las propias medidas de «jitter». Un ejemplo ya comentado: si un receptor puede recuperar los datos sin error, es que el «jitter» del interfaz de interconexión no es relevante. Éste es uno de los motivos por los que los datos digitales pueden ser fácilmente copiados sin error. Sin embargo, hay que evitar el «jitter» en dos lugares decisivos: en la etapa de muestreo, donde el convertidor A/D debe tener una temporización muy precisa, y en la recuperación y regeneración del reloj en la etapa previa a la conversión D/A. En los equipos de audio de calidad media/alta y de cuidado diseño, el «jitter» no ha sido un factor decisivo en la calidad de los equipos. Pero como los equipos que salen al mercado cada vez utilizan longitudes de palabra más largas y frecuencias de muestreo más elevadas, el envejecimiento de los componentes provoca que las tolerancias del «jitter» cada vez se vuelven más críticas. Últimamente, el «jitter» es una de las principales causas de distorsión en la señal digitalizada de audio, por lo que es imprescindible el concurso de buenas prácticas de diseño para minimizar su repercusión.

⁹ Nota del traductor. Los filtros de condensadores conmutados se basan en redes de condensadores e interruptores controlados electrónicamente mediante una señal de reloj. Son dispositivos analógicos, pero la señal de salida no es continua sino que aparece en instantes discretos (como tras un muestreo). La frecuencia de reloj fija la frecuencia de corte del filtro, siendo la relación entre ambas del orden de 100:1 o 50:1. Esto provoca que aparezcan bandas superiores a la de la señal.

Capítulo 5

CORRECCIÓN DE ERRORES

La llegada del audio digital supuso un avance notable dentro de la ingeniería de audio, cambiándola e introduciendo nuevas técnicas. De ellas, la corrección de errores ha sido, probablemente, la más revolucionaria. Con el audio analógico no existía la posibilidad de corregir errores; si la señal transmitida se distorsionaba o corrompía, el daño sufrido era irrecuperable. Con el audio digital, la propia naturaleza de los datos binarios permite la recuperación de los mismos en caso de deterioro. Cuando los datos de audio se almacenan o transmiten, pueden ser codificados de una forma especial y acompañarlos de la redundancia¹, permitiendo la comprobación de que los datos reproducidos están libres de errores. En el caso de detectar un error, debe ser realizado un nuevo tratamiento con la finalidad de corregirlo, y si no puede serlo, intentar disimularlo sintetizando un nuevo dato. En cualquier caso, los sistemas de corrección de errores consiguen que la transmisión y almacenamiento de los datos digitales sea mucho más robusta. Una mejora inmediata que se obtiene con las técnicas de corrección de errores es que permiten aumentar notablemente las tolerancias en el proceso de fabricación de los sistemas de almacenamiento destinados al gran público, tales como el CD y el DVD, reduciendo los costes y simplificando los procesos de fabricación.

Sin embargo, la corrección de errores es algo más que un hecho casual que se pueda utilizar de forma anecdótica, ya que en la práctica es de uso obligado en nuestros equipos digitales. Debido a la elevada densidad con la que se almacenan los datos, un defecto insignificante en el soporte o una simple partícula de polvo puede arruinar cientos de miles de bits. En comparación con el almacenamiento de datos informáticos, donde un bit incorrecto establece la diferencia entre sumar o restar anotaciones en una cuenta bancaria, los datos de audio digital son enormemente indulgentes con los errores, ya que el disfrute en la escucha de un programa musical no es completamente arruinado por un pequeño error en la forma de onda de salida. Sin embargo, la corrección de errores es imprescindible, debido a los duros entornos en los que la mayoría de los soportes de audio se desenvuelven.

La corrección de errores en audio digital supone una oportunidad de preservar la integridad de los datos que no estaba disponible con los sistemas analógicos, siendo imprescindible para asegurar el éxito en el almacenamiento digital, ya que los errores son habituales e inevita-

¹ El término redundancia, como veremos a lo largo del capítulo, desde el punto de vista técnico no tiene el significado habitual, considerándolo nosotros como información que sin ser de audio la codifica y acompaña para disminuir la probabilidad de error en la misma.

bles. Con un diseño adecuado, sistemas de audio como el CD o el DVD pueden aproximarse a los estándares de la industria de ordenadores, la cual especifica tasas de error de 10^{-12} , es decir, inferiores a un bit erróneo cada 10^{12} (¡un billón!) bits. Aunque para la mayoría de las aplicaciones de audio son adecuadas prestaciones menos estrictas, sin la protección de la corrección de errores el almacenamiento digital de audio no sería viable. Por otro lado, el avance de la tecnología digital puede ser medido, y debido en parte, por el desarrollo en la corrección de errores.

FUENTES DE ERROR

La degradación de la señal puede aparecer en cualquier etapa de la cadena de almacenamiento del audio digital. El error en la cuantificación, la falta de linealidad del convertidor, el «jitter» y el desplazamiento del nivel de continua, son factores que limitan las prestaciones del sistema. La utilización de componentes de calidad, el cuidadoso diseño de la circuitería y los estrictos procesos de fabricación constituyen a su vez elementos que deberían siempre minimizar estos errores. Por ejemplo: un convertidor A/D puede tener una linealidad a bajo nivel satisfactoria y ver sus prestaciones completamente arruinadas si su señal de reloj es generada por un PLL con una tasa de «jitter» excesiva.

Los sistemas de corrección de errores se ocupan principalmente del almacenamiento y transmisión de datos, puesto que los errores que se presentan en los soportes son más severos y más difíciles de controlar. Por ejemplo, una cinta magnética puede ser afectada por el polvo, arañazos, huellas dactilares, estiramiento de la cinta, impurezas en el óxido de su capa magnética, arrugas e incluso cortes irregulares de la cinta. Por otro lado, los soportes ópticos pueden verse afectados de asimetría en los «pits», burbujas y defectos del sustrato o en las diferentes capas del disco. La radiodifusión digital está sujeta a interferencias por reflexiones múltiples y sometida a las condiciones atmosféricas y otras señales interferentes.

Si descartamos los defectos de diseño o el mal funcionamiento de los equipos, la mayoría de los errores críticos aparecen en el medio de almacenamiento o en la etapa de transmisión. Estos errores provocan la corrupción de los datos, y por tanto una señal de audio defectuosa. Las causas más importantes de errores en los medios de almacenamiento son los «dropouts», en esencia defectos en el soporte que provocan una momentánea pérdida de la cadena de datos. Los «dropouts» pueden aparecer en cualquier soporte magnético, disco óptico o transmisión inalámbrica, pudiéndose deber a dos causas: un defecto de fabricación del soporte o una deficiencia en el uso del mismo. Por ejemplo, las cintas magnéticas y discos ópticos son fabricados en condiciones ambientales extremadamente limpias; sin embargo, partículas microscópicas de polvo y otros materiales extraños pueden entrar en los materiales de base para la fabricación, y llegar a producir «dropouts» que generen errores en los datos. Los defectos en el medio pueden provocar transiciones en la señal digital, que siendo mal interpretadas, dan lugar a datos erróneos. Una pérdida de datos o la recuperación de datos no válidos puede provocar chasquidos, ya que debido a la conversión de una palabra incorrecta, la salida del convertidor D/A puede «saltar» a un valor que nada tenía que ver con la información original. La importancia del error, generalmente, depende de la naturaleza del mismo. Un error en el bit menos significativo de una palabra PCM puede pasar completamente inadvertido, mientras que si el error recae en el bit más significativo provoca un cambio tan brusco de nivel que rara vez será no percibido.

Los errores que pueden aparecer son de varios tipos, y mediante su clasificación podemos identificarlos mejor y calibrar su alcance. Hay errores en los cuales no existe relación entre un error y el siguiente, denominándolos errores aleatorios o errores de bit («random-bit error»). Aparecen de forma individual y aislada, siendo generalmente de fácil corrección. Un error de ráfaga («burst error») es un error continuo que puede recaer sobre cientos de bits, siendo su

naturaleza muy diferente de los errores de bits. Suele ser provocado por un defecto en el proceso de fabricación del soporte, una partícula extraña en un disco o cinta, un pico de ruido (como el producido por una descarga eléctrica), diafonía o problemas en un conector en un canal de transmisión. Una característica importante de los sistemas de corrección de errores es la longitud de la ráfaga que puede ser corregida, esto es, el máximo número de bits erróneos contiguos que pueden ser corregidos. Tanto los errores de bit como los de ráfaga se pueden dar en un mismo medio, y por tanto un sistema de corrección de errores debe ser diseñado para poder corregir ambos errores al unísono. Además, como la naturaleza de los errores depende básicamente del medio, las técnicas de corrección de errores deben ser optimizadas para cada medio en particular. Además de los errores en los soportes ópticos y magnéticos, deben considerarse los errores producidos en la transmisión de los datos, por lo que también debe diseñarse el sistema de corrección de errores en función del tipo de código de canal utilizado en el transporte de los datos.

En los soportes magnéticos digitales hay varias condiciones que provocan graves errores, siendo los «dropouts» los más importantes. Con el proceso de desgaste habitual que sufre la superficie de la cinta, las partículas de óxido comienzan a desprenderse del sustrato, depositándose en otros puntos de la cinta. Otras partículas extrañas como polvo, suciedad del ambiente, grasa de las huellas dactilares y los microscópicos arañazos provocados por las guías metálicas de la cinta son elementos que contribuyen a generar e incrementar los «dropouts». En la Figura 5.1 aparecen una serie de elementos extraños, habituales contaminantes de los soportes. Aunque sus dimensiones son muy pequeñas para la escala a la que nos movemos los humanos, sus tamaños son desproporcionadamente grandes cuando se los compara con el que ocupa un bit en un soporte, y que puede ser tan pequeño como $1\text{ }\mu\text{m}$ (1 milésima parte de milímetro). En una cinta magnética, un «dropout» puede aparecer en un punto fijo de la cinta, o ir «viajando» a puntos diferentes, desplazándose como una partícula errante y creando un «dropout» fantasma (por sus apariciones esporádicas). Los «dropouts» que se generan en el proceso de fabricación de la cinta aparecen con mayor frecuencia en el principio y final de ésta, estando generalmente asociados al proceso de corte de los extremos de la cinta. En los procesos de grabación y reproducción, una partícula puede provocar la separación instantánea entre cabeza y cinta, provocando una caída momentánea del campo magnético, denominándose este error pérdidas por alejamiento cabeza/cinta.

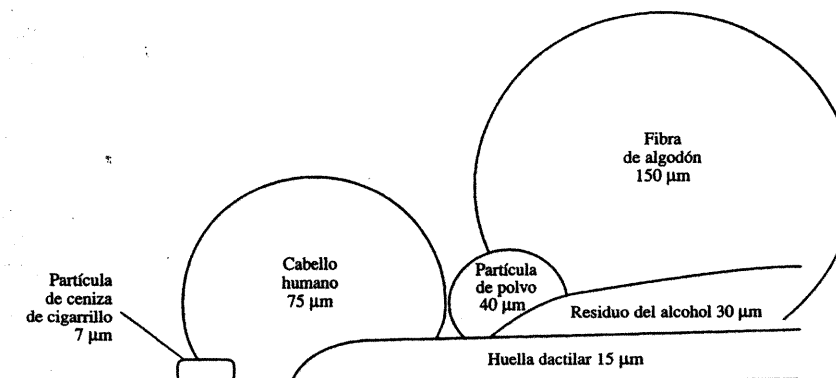


Figura 5.1. El tamaño de objetos habituales es muy grande en comparación con el tamaño ($1\text{ }\mu\text{m}$) de un bit en la grabación.

Los discos ópticos también sufren de «dropouts», aunque naturalmente su origen recae en los procesos de fabricación y manipulación del soporte. Cuando se fabrica un disco maestro, una incorrecta intensidad del haz láser, o un tiempo de revelado erróneo, o un defecto en la capa fotoresistiva, cada uno de estos factores puede provocar la creación de pits de una geometría incorrecta. El polvo o los arañazos producidos en los procesos de fabricación de los troqueles, o una presión incorrecta en el moldeado por presión, o microscópicas perforaciones de la capa metalizada del disco pueden crear «dropouts». Como en el caso de una cinta magnética, las estrictas condiciones de fabricación y los exhaustivos controles de calidad pueden prevenir muchos de estos errores en el proceso de fabricación. Los discos ópticos se ensucian y deterioran con el uso. El polvo, la suciedad y la grasa se pueden eliminar de la superficie del soporte, mientras que los arañazos pueden entorpecer el proceso de lectura realizado por el fonocaptor óptico. Para la limpieza de un disco óptico (como un disco compacto) debe utilizarse un paño suave, frotando el disco radialmente, esto es, desde el centro del disco hacia la circunferencia, y evitando en todo caso un movimiento circular. El motivo es que los arañazos perpendiculares a las pistas son muy fáciles de corregir, mientras que un simple arañazo que se alinee sobre un segmento de la pista es imposible corregirlo al provocar una pérdida muy elevada de datos consecutivos.

Un gran número de factores del canal conspiran para degradar la calidad de los datos transmitidos, incluyendo la limitación en el ancho de banda, la interferencia entre símbolos, atenuación, ruido, oscilaciones transitorias y reflexiones, entre otras. Por ejemplo, como los datos viajan a través de un cable, éste actúa como un filtro paso bajo reduciendo el ancho de banda e introduciendo retardo de grupo. Una señal cuadrada tiene unos flancos de subida y bajada con transiciones más o menos graduales; si el régimen binario es demasiado alto, la señal no podrá alcanzar completamente los niveles lógicos asociados al uno y al cero. En otras palabras, el nivel que alcanza un bit en particular depende, en parte, de los niveles de los bits previos, y ésta es la denominada interferencia entre símbolos. De forma similar, también se puede producir interferencia debido a la reflexión de la señal en el extremo del cable, pudiendo los ecos generados por los circuitos de los extremos (terminales) reforzar o cancelar la señal en determinados puntos. Estas reflexiones se generan por la desadaptación de impedancias entre el cable y los circuitos terminales, siendo imprescindible efectuar una adaptación correcta, tanto más cuanto más larga sea la línea y mayor régimen binario se pretenda alcanzar. Las interferencias externas pueden considerarse como otras fuentes de error, incluyendo el ruido de baja frecuencia, que sumándose a la señal del cable puede provocar desplazamientos en las transiciones de los datos y generar «jitter». Las interferencias de RF, o las de alta frecuencia, deben evitarse mediante las técnicas de apantallamiento y puesta a tierra. La transmisión a través de fibra óptica posee una gran ventaja frente al cable, la inmunidad frente a las interferencias de radiofrecuencia y otras señales eléctricas. La calidad de una señal digital puede evaluarse a partir de su diagrama de ojos, como veremos más adelante en el Capítulo 4.

Hemos definido un gran número de parámetros para cuantificar la integridad de los datos. El BER («bit-error rate») es la relación entre el número de bits erróneos y número de bits totales recibidos. Por ejemplo, un BER de 10^{-9} especifica un error en un bit cada 10^9 bits recibidos, con lo que en definitiva el BER es una forma de contar los bits erróneos, o la velocidad a la que se producen. Un sistema de disco óptico, por ejemplo, puede incluir un algoritmo de corrección de errores capaz de manejar BER entre 10^{-5} a 10^{-4} . El BER especifica el número de errores, pero no la distribución de los mismos. Por ejemplo, el valor del BER puede ser el mismo para un gran error de ráfaga que para el mismo número de pequeños errores distribuidos en el soporte, no siendo un indicador fiable en aquellos canales propensos a sufrir errores de ráfaga, como por ejemplo, en un disco óptico, donde un 80 por 100 de los errores que aparecen son de ráfaga. Otra forma distinta de medir los errores es el BLER («Block Error Rate»), medida del número de bloques o tramas de datos por segundo en los que aparece un error no corregible. Otro parámetro es el BERL («Block Error Length»), que cuenta el número de bloques erróneos consecutivos en los que aparece un error. El BLER y el BERL se expresan, a menudo, como veloci-

dades de error por segundo; por ejemplo, un CD puede tener 20 BLER por segundo. Estos valores no reflejan exactamente el número de errores que se producen, pero son un buen indicador de la importancia de los mismos. El BLER se discutirá con mayor detalle cuando hablemos del algoritmo de corrección de errores de intercalación cruzada Reed-Solomon. Del mismo modo, puesto que los errores de ráfaga aparecen intermitentemente, hay dificultades en cuantificar su magnitud, o la velocidad a la que se dan, midiendo en algunas ocasiones el número de bits erróneos por segundo. Al igual que el BLER o el BELR, esta medida no muestra cuántos bits hay erróneos, pero sí es un indicador de los mismos. Independientemente de cómo sean tabulados o medidos los errores, el objetivo de cualquier sistema de corrección de errores está claro: detectar los errores, corregir el máximo número de ellos y mantener en el mínimo posible los que no se pueden corregir.

Objetivos de la corrección de errores

Cualquier sistema real de almacenamiento o canal de transmisión introduce errores en los datos que almacena o transmite. Con el fin de recuperar, de la forma más precisa posible, los datos perdidos, hay que acompañarlos de la redundancia que permita superar el error. Sin embargo, la redundancia en sí misma no asegura la recuperación de la información, debiendo usar un código de detección y corrección de errores apropiado.

Aunque el diseño de un sistema ideal de detección y corrección de errores (en el cual cada error sea detectado y corregido) es teóricamente posible, un sistema de estas características necesitaría transmitir tal cantidad de redundancia que sería impracticable². En audio, las aspiraciones de un sistema eficiente de corrección de errores son: mantener una tasa de errores audibles lo suficientemente baja y minimizar tanto la cantidad de redundancia a transmitir como el tratamiento (procesado) requerido para la operación global. En general, un sistema de corrección de errores comprende tres etapas:

- 1.ª Detección de errores. Empleando la redundancia para comprobar la validez de los datos recibidos.
- 2.ª Corrección de errores. A partir de la redundancia, calcular los nuevos datos que sirvan para reemplazar los datos erróneos.
- 3.ª En el caso de errores grandes o datos insuficientes para realizar la corrección, utilizar técnicas de ocultación de errores. Éstas consisten en ir sustituyendo los datos erróneos por los nuevos datos calculados que sin coincidir con los originales sean lo más próximos posibles a ellos.

En el peor de los casos, cuando el flujo de errores desborda el sistema y no hay forma de corregirlos ni disimularlos, los sistemas digitales deben enmudecer la salida, impidiendo que tal conjunto de datos inconexos lleguen a convertirse en la escucha de ruidos insoportables.

DETECCIÓN DE ERRORES

Todas las técnicas de detección y corrección de errores están basadas en la redundancia de los datos. Se denominan datos redundantes aquellos que derivándose completamente de los datos existentes no transportan información adicional. En general, a mayor probabilidad en la aparición de errores, mayor es la cantidad de redundancia que se necesita. Los sistemas de transmi-

² En estas técnicas se suele denominar «overhead» al número de bits que asociados al sistema de corrección de errores sirven a este propósito. Nosotros le denominaremos *exceso de bits* o *sobrecarga*.

sión de la información dependen fundamentalmente de la redundancia para realizar unas comunicaciones fiables. Por ejemplo, el lenguaje oral y el escrito contienen una gran redundancia, y si recibe el mensaje «PXINC%PIOS DE ?UDIO D#GIT6L», a pesar de la cantidad de errores introducidos la frase es sin duda reconocida³.

De forma similar, en la comunicación de datos se necesita la redundancia para asegurar la correcta interpretación del mensaje. Si aisladamente se genera un dato, se transmite y se recibe de forma aislada, en recepción no hay ninguna forma de comprobar su validez. En el mejor de los casos, si una palabra difiere radicalmente de sus vecinas puede plantear sospechas sobre su validez. En audio digital, en donde hay una pequeña correlación entre un dato y sus vecinos, ya que les separa sólo una milésima de segundo, un algoritmo que tuviera esto en cuenta sería razonablemente útil; y sin embargo, no tenemos forma de detectar el error, ni mucho menos de corregirlo. Necesitamos una información adicional para detectar los errores que se produjeron sobre los datos recibidos, y puesto que la procedencia de esta información adicional es la misma que los datos originales, es posible que las mismas condiciones que crearon el error sobre los datos actúen sobre la redundancia. La tarea del sistema de detección de errores es transmitir o almacenar los códigos apropiados, para que si en recepción los datos se pierden o se suponen incorrectos, la presencia de un error sea cuando menos detectada.

En primera instancia, el método más simple para detectar errores es enviar repetido el mensaje original. Por ejemplo, transmitir en un mensaje cada palabra dos veces. El problema es que una diferencia entre las dos palabras recibidas revelaría que una de ellas es incorrecta, pero sería imposible identificar cuál de ellas. Si cada palabra fuera repetida tres veces, la probabilidad nos sugiere que las dos que concuerden sean las correctas, mientras que la diferente sería sobre la que hubiera recaído el error; aunque también es posible que las tres coincidieran y fueran erróneas, siendo entonces incapaces de descubrirlo. Repitiendo cada palabra un número suficiente de veces, la probabilidad de detectar un error podría ser muy alta, aunque el incremento de datos sería enorme. Además, con esta redundancia el incremento de los nuevos datos podría introducir errores adicionales, por lo que conviene desestimar este proceso y diseñar sistemas más eficientes.

Paridad simple

En la práctica, los sistemas de detección de errores están basados en técnicas en las que los datos de redundancia se codifican con la finalidad de servir para la comprobación de errores. Uno de estos métodos es la paridad. Uno de los primeros métodos de detección de errores fue inventado en el siglo noveno por matemáticos árabes, siendo conocido como la «prueba del 9». En esta técnica, los términos implicados en una operación (suma o producto) son divididos por 9, quedando un resto o residuo. La comprobación de la bondad del resultado puede realizarse efectuando la operación entre los residuos y comparándola con el residuo del resultado. Por ejemplo: el residuo de la suma (o producto) de dos números es igual a la suma (o producto) de los residuos de los operandos. La importancia de la prueba es que la operación se vuelve a realizar entre los residuos, y si no coinciden los resultados es que hay un error en el cálculo (véase Fig. 5.2). Pero otro truco hace el método aún más eficiente, y es que la suma de los dígitos de un número tiene el mismo residuo del número en sí. La técnica de la prueba del nueve puede ser adaptada a cualquier sistema de numeración, y su forma básica en el sistema binario se denomina paridad.

³ Nota del traductor. Como la frase original no tenía sentido traducirla, me he permitido usar una muy sencilla y realizar el siguiente comentario. Hay que hacer notar que la relación entre un carácter y el siguiente es casi inexistente, pero la concatenación de caracteres en la formación de una palabra introduce una estructura reglada, que es la que en este caso aporta la «redundancia». Son nuestros conocimientos de la lengua, la lógica y, por último, la correlación entre la palabra recibida y los patrones que conocemos los elementos que nos permiten detectar dónde hay un error y mentalmente «corregirlo».

Prueba del 9:

$$240 + 578 \stackrel{?}{=} 818 \rightarrow (2 + 4 + 0 = 6) \\ (5 + 7 + 8 = 20, 2 + 0 = 2) \\ (8 + 1 + 8 = 17, 1 + 7 = 8)$$

$$6 + 2 \stackrel{?}{=} 8 \quad \text{La prueba del 9 sobre la suma se cumple, por tanto no hay error}$$

$$227 \times 67 \stackrel{?}{=} 15.209 \\ 2 \times 4 \stackrel{?}{=} 8 \quad \text{La prueba del 9 sobre el producto se cumple, por tanto no hay error}$$

$$154 \times 95 \stackrel{?}{=} 14.613 \\ 1 \times 5 \neq 6 \quad \text{La prueba del 9 sobre el producto no se cumple, por tanto el cálculo es erróneo}$$

Prueba del 2:

La suma de 11001011 es 5, el cual es impar. La prueba del dos da 1, el cual le añadimos a la palabra y tenemos 110010111. Así, el número de unos (1) es siempre par.

Figura 5.2. La prueba del 9 y la del 2 aportan un sencillo método de detección de errores.

Dado un número binario, el bit de residuo se forma con la prueba del 2. Este bit extra se crea cuando la palabra se transmite o almacena, añadiéndose a la palabra dato para su transmisión conjunta. El bit añadido, denominado paridad, permite la detección de los errores, pero no su corrección, por lo que se debe añadir un algoritmo que mejore la eficiencia del sistema. Un bit de paridad par se genera con una regla muy simple: si el número de unos en la palabra dato es par (o cero) al bit de paridad se le asigna un cero, mientras que si el número de bits a uno es impar, el bit de paridad se dispone a uno. En pocas palabras, con la paridad par hay siempre un número par de bits a uno en la transmisión. La obtención de esta paridad se efectúa sumando los bits en módulo 2, tal y como aparece representado en el diagrama de la Figura 5.3. Así, una palabra de datos de 8 bits, con la agregación de la paridad se convierte en un dato de 9 bits. Este método de paridad se denomina paridad par, mientras que si forzamos a que el número de bits a uno sea impar, se denomina lógicamente paridad impar, aunque ambos métodos son funcionalmente idénticos.

En la etapa de reproducción, la validez de los datos recibidos se comprueba mediante el bit de paridad, calculando la paridad sobre los datos recibidos, lo cual se puede realizar de forma sencilla sumando todos los bits. Si la paridad calculada no corresponde con la utilizada esto muestra la aparición de un error. Este método permite la detección de un número impar de bits erróneos, pero no tiene posibilidad de detectar sobre cuál o cuáles se ha producido el error, y en consecuencia no posee capacidad de corrección (véase Fig. 5.4). La posibilidad de que aparezca un error es tal, que si surge, probablemente recaiga sobre los bits de datos en vez de sobre la paridad; aunque nada impide lo contrario, y por tanto que recayendo el error sobre la paridad, sean tomados unos datos correctos como erróneos. Por otro lado, ya dijimos la tendencia a que aparezcan errores de ráfaga en la reproducción de los datos, de tal manera que si afectan a un número elevado de palabras consecutivas, la paridad no es capaz de actuar como un método de detección fiable. La conclusión es que en el almacenamiento y transmisión de audio digital, la paridad en sí misma no constituye un sistema fiable de detección de errores.

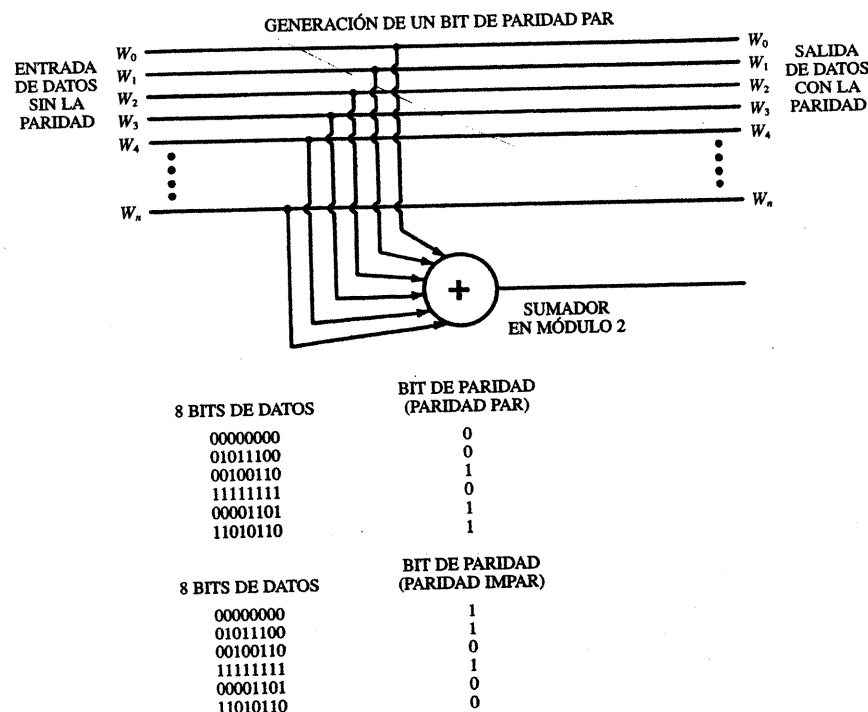


Figura 5.3. La paridad puede generarse a partir de la suma en módulo 2 de los bits de datos.

ISBN

En la práctica, un simple bit de paridad no es de utilidad en muchas aplicaciones, por lo que se requieren sistemas más sofisticados de detección de errores que hagan un uso más racional de la redundancia. Un ejemplo de estos sistemas es el ISBN (International Standard Book Number), codificación utilizada en la designación de todos los libros publicados en el mundo. No hay dos libros, ni tan siquiera dos ediciones del mismo libro, con el mismo ISBN, llegando incluso a que ediciones con pastas blandas y duras del mismo libro y edición poseen diferente ISBN.

| Palabra transmitida | | Palabra recibida | | Paridad calculada a partir de los datos recibidos | |
|---------------------|---------|------------------|---------|---|----------------------------|
| Datos | Paridad | Datos | Paridad | | |
| 00011001 | 1 | 00001001 | 1 | 0 | Error detectado |
| 10101011 | 1 | 11001011 | 1 | 1 | Errores no detectados |
| 01110100 | 0 | 01110100 | 1 | 0 | Detectado error de paridad |
| 01101011 | 1 | 00000011 | 0 | 0 | Errores no detectados |

Figura 5.4. Ejemplos de detección de error de paridad en un bit.

El código ISBN es algo más que una serie de dígitos. Por ejemplo, considerar el ISBN 0-14-044118-2. El primer dígito (0) es el código del país; por ejemplo, el 0 es el código para Estados Unidos y otros países de habla inglesa. Los siguientes dos dígitos (14) corresponden al código del editor. Los seis dígitos siguientes son el código del libro (044118); y el último dígito es para nosotros el más interesante, ya que corresponde a un dígito de chequeo utilizado para verificar el resto. Este dígito corresponde a la suma en módulo 11 de los dígitos previos; y para mantener la longitud uniforme de diez dígitos, se emplea el número romano X en la representación del dígito 10. Consideremos un ejemplo de la comprobación de un código ISBN. Para formar el código de chequeo de un número de diez dígitos *abcdefghij* hay que calcular la suma ponderada de los números multiplicando cada dígito por su posición y comenzando por el dígito más a la izquierda:

$$10a + 9b + 8c + 7d + 6e + 5f + 4g + 3h + 2i + 1j$$

para el ISBN 0-14-044118-2, la suma ponderada es:

$$10 \times 0 + 9 \times 1 + 8 \times 4 + 7 \times 0 + 6 \times 4 + 5 \times 4 + 4 \times 1 + 3 \times 1 + 2 \times 8 + 1 \times 2 = 110$$

y la suma en módulo 11 corresponde al resto de dividir 110 entre 11:

$$\frac{110}{11} = 10, \text{ con } 0 \text{ como resto}$$

indicando el resto 0 que el ISBN es correcto. De esta forma se comprueba que el ISBN es el correcto. El empleo de una suma ponderada, comparado con el cálculo del resto en una aritmética con un determinado módulo, aporta un método muy potente de detección de errores, siendo éste, de hecho, el principio básico en el que se basan los sistemas de detección de errores.

CÓDIGO DE CHEQUEO DE REDUNDANCIA CÍCLICA

El código de chequeo de redundancia cíclica (CCRC) es el método de corrección de errores preferido dentro de las aplicaciones de audio, ya que tiene la habilidad de detectar los errores de ráfaga que se producen en la transmisión o en los medios de almacenamiento. El CCRC es un código aplicado a bloques de datos de una manera cíclica, generando una o más palabras de chequeo o paridad. Pongamos un ejemplo: los bits de una palabra dato se suman todos para formar una palabra de chequeo de paridad; así en la palabra 1011011010 los seis unos binarios se suman para dar 0110 (seis en binario), y esta palabra se añade a la palabra dato original para conjuntamente formar un código que sea el que se transmita o almacene. Al igual que con la paridad simple, cualquier discrepancia entre el dato recibido y la suma de control podría indicar, con gran probabilidad de acierto, la ocurrencia de un error.

El CCRC trabaja de forma parecida, aunque con un cálculo más sofisticado. En primer lugar separa los datos en bloques, considerando cada bloque como un número; a continuación divide el bloque por un número arbitrario y constante, y el resto obtenido lo añade al bloque de datos original para su almacenamiento o transmisión. En la reproducción se realiza la división sobre los datos recibidos (véase Fig. 5.5), y si el resto obtenido y recibido coinciden se puede suponer una recepción libre de errores. Un examen más detallado de los pasos seguidos en el proceso de codificación y decodificación de un algoritmo CCRC se muestra en la Figura 5.6. Sobre el mensaje *m* formado por un bloque de datos de *k* bits, se opera hasta obtener un conjunto de *n - k* datos de CCRC, donde *n* es la longitud (o número de bits) del bloque completo. El bloque de datos original (de *k* bits) es multiplicado por X^{n-k} para desplazar los datos y prepararlos para añadir los códigos de chequeo. Ahora se dividen los datos por el polinomio genera-

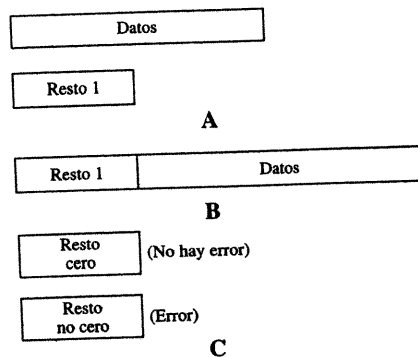


Figura 5.5. Un código cruzado de redundancia cíclica (CCRC) en su forma más simplificada muestra cómo se genera el resto. A. Un bloque de datos es dividido para producir un resto. Se descarta el cociente y nos quedamos sólo con el resto. B. Se añade el resto a la palabra dato y se transmiten o almacenan ambas palabras. C. Los datos recibidos se dividen de nuevo para producir un resto, utilizado para la comprobación de errores.

donde g , obteniendo un cociente q y un resto r . El polinomio a transmitir, u , se forma con el mensaje original m y el resto r , siendo por tanto un múltiplo del polinomio generador g . Cuando se recupera el polinomio u , se realizan sobre él los cálculos destinados a la obtención de un síndrome, en el sentido de que éste sirve para la detección de un error o mal funcionamiento. Específicamente, el síndrome c se genera con la suma en módulo 2 de la paridad recibida y la calculada a partir del mensaje recibido, con lo que si el síndrome obtenido es cero se asume la recepción sin error y viceversa. Así se obtiene la detección de condición de error en la recepción, pero también podemos extender el proceso para obtener un patrón de error que nos muestre la diferencia entre los datos originales y los recuperados, con lo que el sistema serviría también para indicar los bits en los que se ha producido el error y permitir su corrección. Matemáticamente esto es posible debido a que el polinomio erróneo e dividido por el polinomio generador produce un cociente y un resto, el síndrome, de forma que podemos utilizar éste para formar el patrón de error que nos permita recuperar los datos. Lo importante es seleccionar el polinomio generador g para que los patrones de error e no sean divisibles por g , ya que si no el CCRC fallaría en la detección del error al ser el síndrome cero.

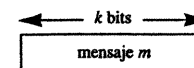
En la práctica, el CCRC y otros métodos de detección y corrección de errores se describen de forma matemática, donde los bits de datos son tratados como coeficientes de un polinomio binario. Como se comentó en el Capítulo 1, el sistema de numeración binaria es un código posicional, donde cada posición representa una potencia de 2. Así, podemos escribir los números binarios en una notación donde, por ejemplo, el número 1001011 (siendo el primero el bit más significativo) es expresado como:

$$1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

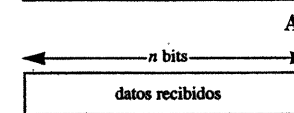
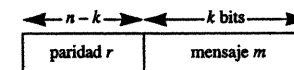
$$\text{o:} \quad 2^6 + 2^3 + 2^1 + 2^0$$

que también puede ser expresado como:

$$x^6 + x^3 + x + 1$$



$m = m_0, m_1, m_2, \dots, m_{k-1}$
en forma polinómica, queda
 $m(X) = m_0 + m_1 X + m_2 X^2 + \dots + m_{k-1} X^{k-1}$
multiplicando $m(X)$ por X^{n-k}
 $X^{n-k} m(X) = m_0 X^{n-k} + m_1 X^{n-k+1} + \dots + m_{k-1} X^{n-1}$
dividiendo $X^{n-k} m(X)$ por $g(X)$, el polinomio generador
 $X^{n-k} m(X) = q(X) g(X) + r(X)$
donde $q(X)$ y $r(X)$ son el polinomio cociente y resto respectivamente
donde $r(X) = r_0 + r_1 X + r_2 X^2 + \dots + r_{n-k-1} X^{n-k-1}$
ordenando la ecuación previa y sumándole $r(X)$
 $r(X) + X^{n-k} m(X) = q(X) g(X) + r(X) + r(X)$
sin embargo, $r(X) + r(X) = 0$; y entonces
 $r(X) + X^{n-k} m(X) = q(X) g(X)$
con lo que $r(X) + X^{n-k} m(X)$ es un múltiplo de $g(X)$
el polinomio a transmitir es entonces $v(X) = r(X) + X^{n-k} m(X)$
 $v(X) = r(X) + X^{n-k} m(X) = r_0 + r_1 X + r_2 X^2 + \dots + r_{n-k-1} X^{n-k-1}$
 $+ m_0 X^{n-k} + m_1 X^{n-k+1} + \dots + m_{k-1} X^{n-1}$
lo que corresponde a la palabra código a transmitir:
 $(r_0, r_1, r_2, \dots, r_{n-k-1}, m_0, m_1, m_2, \dots, m_{k-1})$



$u = u_0, u_1, u_2, u_3, \dots, u_{n-1}$
en forma de polinomio
 $u(X) = u_0 + u_1 X + u_2 X^2 + \dots + u_{n-1} X^{n-1}$
donde $u_0, u_1, u_2, \dots, u_{n-k-1}$ son los bits de paridad y $u_{n-k}, u_{n-k+1}, \dots, u_{n-1}$ son los bits de información.
El síndrome s se calcula haciendo la suma en módulo 2 entre los bits de paridad recibidos y los bits de paridad calculados sobre la información recibida. Así, el síndrome $s(X)$ es igual al resto de $u(X)$ dividido por $g(X)$:
 $u(X) = p(X) g(X) + s(X)$
Un valor distinto de cero para s implica un error. La diferencia entre la información recibida u y la transmitida v es un patrón de error $e(X)$. A partir de e podemos recuperar v mediante el síndrome
 $u(X) = v(X) + e(X)$
como
 $v(X) = m(X) g(X)$
 $u(X) = m(X) g(X) + e(X) = p(X) g(X) + s(X)$
con lo que $e(X) = [p(X) + m(X)] g(X) + s(X)$

Cuando el patrón de error se divide por el polinomio generador, el resto que queda es el síndrome, el cual puede utilizarse para corregir los errores. Nótese que el polinomio generador se elige para que el polinomio de error resultante consista en un patrón de errores no divisible por g . Las conclusiones anteriores se derivan de utilizar las propiedades de la aritmética en módulo 2.

B

Figura 5.6. Algoritmos de codificación y decodificación de un CCRC. A. Método de codificación. B. Decodificación y cálculo del síndrome para la detección y corrección de errores.

Ésta es la denominada notación polinómica, terminología estándar en el campo de la corrección de errores. De la misma forma que la utilización de una aritmética en módulo 2 fuerza a que los valores posibles sean en módulo 2 (pertenecan al rango de valores 0 y 1), los polinomios aseguran que los valores posibles a manejar caigan dentro de un determinado

Dado el mensaje $m = (1001)$ a codificarse, el polinomio que lo representa es $m(x) = x^3 + 1$. Multiplicándolo por x^{n-k} , $x^3 \cdot m(x) = x^6 + x^3$.

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline \end{array} \quad m(x)$$

$$x^3 + 1$$

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline \end{array} x^3 m(x)$$

$x^6 \quad + \quad x^3$

Dado el polinomio generador $g(x) = x^3 + x^2 + 1$, se realiza la división por $g(x)$:

$$\begin{array}{r} x^3 + x^2 + 1 \overline{) \begin{array}{l} x^3 + x^2 + x + 1 \\ x^6 + x^5 + x^4 + x^3 \end{array} } \\ \underline{x^3 + x^2} \\ x^5 + x^4 + x^3 + 1 \\ \underline{x^5 + x^4} \\ x^3 + 1 \\ \underline{x^3} \\ 1 \end{array}$$

Resto = $r(x)$

La palabra código en forma de polinomio queda

$$\begin{aligned}(x) &= x^3 m(x) + r(x) \\ &= x^6 + x^3 + x + 1\end{aligned}$$

$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

$$\underbrace{x^6 + x^3 + x + 1}_{\text{Mensaje}} \quad \underbrace{}_{\text{Paridad}}$$

Figura 5.7. Un ejemplo de codificación mediante un código cíclico. El mensaje 1001 es escrito como polinomio $x^3 + 1$. La salida del codificador está compuesta por el mensaje original y la palabra de paridad.

rango⁴. Ahora todos los polinomios serán divididos por un polinomio de grado n , y el resto (que tendrá por tanto n coeficientes) es al que se reconocerá como un símbolo que pertenecerá al conjunto de códigos válidos. Por otro lado, queremos que el resto resultante sea un número primo, para lo que utilizamos un polinomio que será también primo (que no sea divisible por otro, ni se pueda descomponer como el producto de otros dos polinomios).

Veamos cómo trabaja el CCRC utilizando esta notación. Un ejemplo de codificación se muestra en la Figura 5.7. El mensaje a transmitir es 1001, y se puede escribir como $x^3 + 1$. Si queremos utilizar tres bits de paridad y disponerlos al final del mensaje, hay que desplazarlo tres bits hacia la izquierda; o lo que es lo mismo, multiplicarlo por x^3 , con lo que queda $x^3 \cdot (x^3 + 1) = x^6 + x^3$. El polinomio generador empleado es $x^3 + x^2 + 1$; con lo que el paso siguiente es dividir el dato ($x^6 + x^3$) por el generador, resultando como resto $x + 1$. Si al mensaje original se añade este resto nos queda como mensaje a transmitir $x^6 + x^3 + x + 1$ (1001011). La implementación del codificador se efectúa mediante el registro de desplazamiento que aparece represen-

⁴ *Nota del traductor.* Inferior siempre al rango del polinomio generador. El proceso es análogo a nuestra división con números enteros, donde el resto siempre es menor que el divisor.

tado en la Figura 5.8. El registro se carga inicialmente con ceros, y los cuatro bits del mensaje van entrando secuencialmente en el registro y apareciendo en la salida de éste. Para efectuar todas las operaciones se utilizan sumadores en módulo 2, con lo que otra ventaja añadida de este sistema de detección de error es que su implementación electrónica es realmente simple.

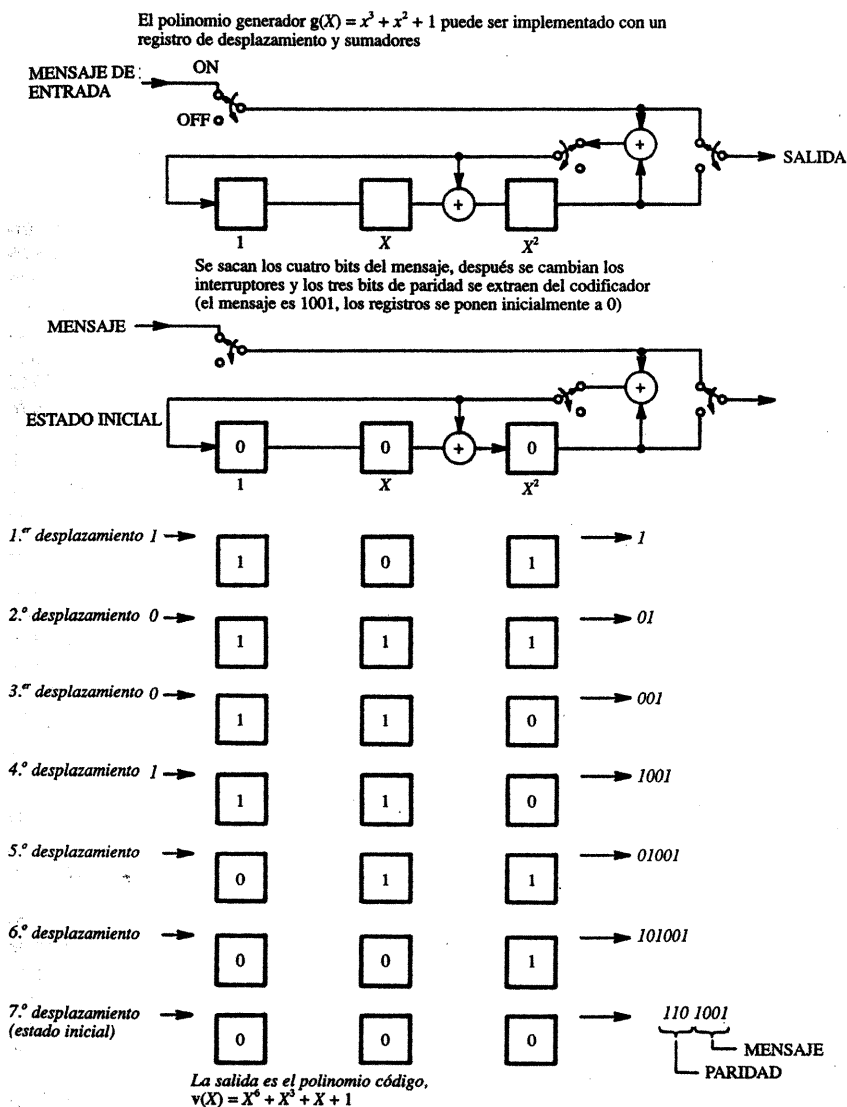


Figura 5.8. Implementación de un codificador cíclico utilizando registros de desplazamiento. Se utilizan sumadores en módulo 2 (OR Exclusiva).

A medida que los bloques de datos aumentan de tamaño necesitan una menor redundancia, con lo que un detallado análisis matemático mostraría que la capacidad de detección de errores se mantiene. Sin embargo, si aparecen con elevada frecuencia errores de ráfaga, la capacidad de detección de errores decrece notablemente, hecho que fuerza a trabajar en la práctica con bloques de datos de tamaño más reducido. La capacidad de detectar errores en un CCRC es la siguiente. Dada una palabra de datos de k bits, con m bits de CCRC, y por tanto formando un código a transmitir de n bits ($n = k + m$), se cumple:

1. Los errores de ráfaga de longitud menor o igual a m bits son siempre detectables.
2. La probabilidad de detectar errores de ráfaga de $m + 1$ bits es de $1 - 2^{-m+1}$.
3. La probabilidad de detectar errores de ráfaga más largos de $m + 1$ bits es de $1 - 2^{-m}$.
4. Los errores aleatorios en hasta tres bits consecutivos son siempre detectados.

(Obsérvese que los apartados 1, 2 y 3 no están afectados por la longitud n del mensaje.)

El sistema CCRC es un método de detección muy fiable. Por ejemplo, si se utiliza una paridad de 16 bits, la probabilidad de detectar un error es de $1 - 2^{-16}$, o lo que es lo mismo de un 99,99 por 100. En última instancia, el medio de almacenamiento es el que determina el diseño del CCRC y del resto de elementos del sistema de corrección de errores. Por ejemplo, los sistemas de grabación magnética pueden utilizar bloques de datos relativamente largos, mientras que los discos ópticos requieren el empleo de bloques de datos de una longitud considerablemente más corta. La capacidad del sistema de corrección de errores, etapa que sigue al CCRC, también influye en cómo debe diseñarse éste. Habitualmente, el CCRC se utiliza como puntero indicador del número y longitud de los errores presentes en la recepción del mensaje, antes de que se realice el tratamiento correspondiente a la corrección de errores propiamente dicha.

CÓDIGOS DE CORRECCIÓN DE ERRORES

Con el uso de la redundancia, es posible la corrección de los errores que aparecen en el almacenamiento o la transmisión de los datos digitales de audio. El caso más simple es el de la duplicación de los datos. Por ejemplo, en vez de escribir los datos en una pista de datos en un sistema de grabación magnética, podrían escribirse dos pistas con los datos idénticos. La primera pista podría emplearse para la reproducción de los datos, y si se detectan errores en ésta a través de la paridad u otros medios, podría emplearse la segunda pista de datos para recuperar los datos de la señal. Para minimizar el problema de encontramos con datos erróneos simultáneamente en ambas pistas, los datos de audio podrían grabarse desplazados en una pista con respecto a la otra, mejorando la probabilidad de recuperación de la señal.

Además, la codificación de canal puede utilizarse en nuestro beneficio en la corrección de errores. Por ejemplo, una secuencia de tres bits podría codificarse como una palabra de 7 bits, seleccionando de las 2^7 combinaciones posibles aquellas 2^3 que sean más dispares. El receptor examinaría la palabra de 7 bits recibida y la compararía con las 8 secuencias posibles; si apareciera algún error, cambiaría la secuencia al código de 7 bits más parecido, tras lo cual lo decodificaría obteniendo la palabra de tres bits que le correspondiera. Se necesitarían cuatro bits de corrección para cada tres bits de datos, siendo este sistema capaz de corregir un bit erróneo en cada bloque de 7 bits. El hecho de ante un error «elegir el código más parecido» o «asociarle el código que menos difiera de entre los posibles» es lo que en teoría de errores se denomina distancia mínima, siendo un concepto de los más importantes en los sofisticados sistemas de corrección de errores.

Aunque estos métodos tan simples son en la práctica viables, no son eficientes debido al «exceso de redundancia» («overhead») que producen, debiendo introducir otros sistemas de corrección que persigan resultados cada vez más fiables con la menor redundancia posible.

De la misma forma que la redundancia creada a partir de los bits de paridad es utilizada para la detección de errores, también la redundancia se utiliza para formar códigos de corrección de errores. Los datos de audio digital son codificados mediante algoritmos, que introducen sus correspondientes códigos de corrección de errores, de forma que en la reproducción, los errores sean identificados y corregidos por el decodificador. La codificación de los datos redundantes constituye la esencia de la corrección de errores, aunque hay muchos tipos de códigos diferentes, tanto en su diseño como en sus funciones.

El campo de los códigos de corrección de errores es eminentemente matemático, habiéndose desarrollado multitud de códigos destinados a las aplicaciones más dispares. En general, se utilizan dos enfoques distintos: los códigos de grupo que utilizan métodos algebraicos, y los códigos convolucionales que se basan en métodos probabilísticos. Los códigos de grupo forman un mensaje codificado (generan un código) basándose en el análisis del mensaje contenido dentro de un bloque de datos. En los códigos convolucionales, el mensaje a transmitir se forma a partir de los datos actuales presentes en el codificador y los datos previos. En algunos casos se mezclan ambos sistemas, aplicándose algoritmos convolucionales sobre un bloque de datos; denominándose a estos sistemas códigos de intercalación cruzada. Por ejemplo, un código de este tipo es el que se utiliza en el sistema de disco compacto.

Códigos de bloque («Block codes»)

Los codificadores por bloques actúan tomando un determinado número de datos para formar un bloque y, operando sobre los datos del mismo, generar una o más palabras de paridad que se añaden al propio bloque de datos. En la decodificación, un algoritmo genera un síndrome que detecta los errores; y si la redundancia es lo suficientemente elevada, los corrige. Tales algoritmos son de gran efectividad independientemente de la aplicación de audio en la que se usen, mejorando la corrección de errores mediante el entrelazado de palabras consecutivas. Los códigos de bloque realizan los cálculos de la paridad sobre el conjunto de datos del bloque, pudiendo generarse dicha paridad a partir de cada una de las palabras de datos del bloque, utilizando un simple bit de paridad o un código cíclico. De esta forma, a medida que se utiliza mayor redundancia, se mejora la capacidad de corrección total. Por ejemplo, un CCRC puede utilizarse para la detección de los errores y un código de grupo para la corrección de los mismos.

Un código de grupo puede ser concebido como un mensaje binario consolidado dentro de un grupo⁵, con su propia paridad asociada a cada fila y cada columna. Una palabra errónea provoca un error en su fila y columna correspondiente, con lo que el dato erróneo puede ser corregido. Por ejemplo, un mensaje puede ser agrupado en cuatro palabras de 8 bits (denominadas símbolos), sobre las que se genera un bit de paridad para cada fila y cada columna, tal y como se muestra en la Figura 5.9. En el decodificador se comprueba la paridad de los datos, corrigiendo cualquier error que aparezca sobre un símbolo. En el ejemplo anotado, el bit de paridad (fila) muestra que la tercera palabra es errónea; y conocido esto, con la palabra de paridad y el dato incorrecto se obtiene la palabra corregida. Con este sistema puede ser detectado el error en dos símbolos, pero no puede efectuarse la corrección correspondiente. Si aparece un número de errores más elevado el problema puede ser irresoluble, ya que produce un fallo en la detección, en la corrección o en ambos procesos.

Los códigos de corrección de grupo utilizan diferentes métodos para la generación y transmisión de los datos y la paridad; sin embargo, la característica fundamental de todos ellos es que sólo se emplea la información del propio bloque para la generación de la paridad. El alcance de las posibilidades de corrección en estos sistemas se puede ilustrar con el siguiente ejemplo.

⁵ Un grupo no es ni más ni menos que un conjunto de datos, habitualmente audio y paridad. Desde el punto de vista operativo se suelen utilizar estructuras de datos de tipo matricial, de forma que el cálculo de paridad recae sobre las filas y columnas.

| Bloque de datos transmitido | Bit de paridad transmitido |
|---|--|
| 00010111 | 0 |
| 01101010 | 0 |
| 10010111 | 1 |
| 11010110 | 1 |
| 00111100 | Palabra de paridad transmitida |
| Bloque de datos recibido | Bit de paridad recibido |
| 00010111 | 0 |
| 01101010 | 0 |
| 11100100 | 1 |
| 11010110 | 1 |
| 00111100 | Palabra de paridad recibida |
| 01110011 | Palabra de paridad calculada a partir de los datos recibidos y la palabra de paridad |
| Paridad calculada sobre el bloque de datos recibido | |
| 0 | |
| 0 | |
| 0 | Indica error en la palabra 3 |
| 1 | |
| 01110011 | Palabra de paridad calculada |
| + 11100100 | Palabra 3 incorrecta |
| 10010111 | Palabra 3 corregida |

Figura 5.9. Un ejemplo de paridad en un bloque de datos con bits de paridad para las filas y palabra de paridad en las columnas.

Dado un bloque de seis palabras de datos, se calcula la paridad sobre los bits individuales de cada palabra, generándose una séptima palabra (paridad) que se añade a las seis de datos. Para la comprobación de errores se genera un síndrome, comparando (por sustracción en este ejemplo) la paridad (suma) de los datos recibidos con la palabra de paridad. Si el resultado de esta comparación es cero, lo más probable es que no haya acaecido ningún error, tal y como se muestra en la Figura 5.10A. Si mediante un bit de paridad asociado a cada palabra dato o mediante CCRC se apunta que existe un error en una sola palabra dato⁶, condición denominada error simple («single erasure»), se puede poner a cero la palabra errónea; se calcula el síndrome y en este caso el valor no cero del mismo es el que corresponde a la palabra errónea (caso 5.10B). Aun en el caso de que el error pueda aparecer en la propia paridad, y cree un falso aviso de error (condición de puntero erróneo), el síndrome calculado sigue entregando el valor correcto, tal y como se muestra en 5.10C. Este código de corrección de grupo es capaz de detectar las palabras erróneas y corregir una palabra incorrecta, bien sea con un puntero correcto o falso. La capacidad de corrección depende de la habilidad en la generación de punteros a datos (palabras y bits) erróneos. En el caso del ejemplo, salvo que el error sea identificado con el puntero, detectado con un CCRC, o corresponda a una palabra entera perdida, el error no puede ser corregido.

Con el fin de mejorar las prestaciones del sistema anterior, se pueden generar dos palabras de paridad para proteger los datos del bloque. Por ejemplo, una palabra de paridad puede calcu-

⁶ Nota del traductor. En el caso que nos ocupa, se suele denominar puntero a cualquier indicador o método que señala un dato, palabra o bit para advertir de su posible falsedad.

Palabras de datos originales y paridad

| | |
|-------|---|
| W_1 | 10 |
| W_2 | 30 |
| W_3 | 20 |
| W_4 | 25 |
| W_5 | 30 |
| W_6 | 15 |
| P | $130 = W_1 + W_2 + W_3 + W_4 + W_5 + W_6$ |

Palabras de datos y paridad recibidas

| | |
|-------|-----|
| W_1 | 10 |
| W_2 | 30 |
| W_3 | 20 |
| W_4 | 25 |
| W_5 | 30 |
| W_6 | 15 |
| P | 130 |

Síndrome $S = W_1 + W_2 + W_3 + W_4 + W_5 + W_6 - P$
 $= 10 + 30 + 20 + 25 + 30 + 15 - 130 = 0$

Esto indica que no hay error

A Código de corrección en un bloque mostrando la condición de no error

Palabras de datos y paridad recibidas

| | |
|-------|-----|
| W_1 | 10 |
| W_2 | 30 |
| W_3 | 20 |
| W_4 | 15 |
| W_5 | 30 |
| W_6 | 15 |
| P | 130 |

Puntero de error CCRC

Síndrome $S = 10 + 30 + 20 + 15 + 30 + 15 - 130 = -10$

Corrección del error: $W_4 = W'_4 - S$
 $= 15 - (-10)$
 $= 25$

B Código de corrección en un bloque mostrando la condición de error

Palabras de datos y paridad recibidas

| | |
|-------|-----|
| W_1 | 10 |
| W_2 | 30 |
| W_3 | 20 |
| W_4 | 25 |
| W_5 | 30 |
| W_6 | 15 |
| P | 130 |

Puntero de error falso

Síndrome $S = 10 + 30 + 20 + 25 + 30 + 15 - 130 = 0$

Corrección del error: $W_3 = W'_3 - S$
 $= 30 - 0$
 $= 30$

C Corrección en un bloque con un falso puntero

Figura 5.10. Ejemplos de codificación de bloque con paridad simple.

larse mediante la suma de los datos, y la segunda paridad mediante una suma ponderada, tal y como aparece en la Figura 5.11A. Si dos palabras cualesquiera son erróneas y aparecen marcadas como tal, el código permite la corrección. Si dos palabras cualesquiera aparecen marcadas como «errores simples», el código puede usar los dos síndromes para corregir los datos. A diferencia de la paridad simple, esta paridad doble puede corregir cualquier palabra errónea, incluso aunque no sea identificada mediante un puntero, tal y como se muestra en la Figura 5.11B, siendo un sistema utilizado frecuentemente en aplicaciones de audio.

Los códigos cíclicos, tales como el CCRC, son un subconjunto de los códigos de grupo lineales, que pueden ser utilizados para la corrección de errores. Hay códigos de grupo especiales, como los códigos Hamming, que crean síndromes que apuntan directamente a la posición del error. Para ello se utilizan paridades múltiples generadas para cada palabra de datos y una

Datos recibidos y dos palabras de paridad

| | |
|-------|--|
| W_1 | 10 |
| W_2 | 30 |
| W_3 | 20 |
| W_4 | 25 |
| W_5 | 30 |
| W_6 | 15 |
| P | $130 = W_1 + W_2 + W_3 + W_4 + W_5 + W_6$ |
| Q | $440 = 6W_1 + 5W_2 + 4W_3 + 3W_4 + 2W_5 + W_6$ |

$$\text{Síndrome } S_1 = W_1 + W_2 + W_3 + W_4 + W_5 + W_6 - P = 10 + 30 + 20 + 25 + 30 + 15 - 130 = 0$$

$$S_2 = 6W_1 + 5W_2 + 4W_3 + 3W_4 + 2W_5 + W_6 - Q = 60 + 150 + 80 + 75 + 60 + 15 - 440 = 0$$

A Código de corrección de bloque con doble paridad mostrando la condición libre de errores

Datos recibidos y las dos palabras de paridad

| | | |
|-------|-----|-------------|
| W_1 | 10 | |
| W_2 | 30 | |
| W_3 | 20 | $S_1 = -20$ |
| W_4 | 25 | $S_2 = -40$ |
| W_5 | 10 | |
| W_6 | 15 | |
| P | 130 | |
| Q | 440 | |

Algebraicamente se puede ver que

- Si $6S_1 = S_2$ entonces W_1 es erróneo
- Si $5S_1 = S_2$ entonces W_2 es erróneo
- Si $4S_1 = S_2$ entonces W_3 es erróneo
- Si $3S_1 = S_2$ entonces W_4 es erróneo
- Si $2S_1 = S_2$ entonces W_5 es erróneo
- Si $S_1 = S_2$ entonces W_6 es erróneo
- Si $S_1 \neq 0$ y $S_2 = 0$ entonces P es erróneo
- Si $S_1 = 0$ y $S_2 \neq 0$ entonces Q es erróneo

En este caso $2S_1 = S_2$, W_5 es erróneo, entonces (como un error simple)

$$S_1 = 10 + 30 + 20 + 25 + 0 + 15 - 130 = -30$$

$$W_5 = W_5' - S_1$$

$$= 0 - (-30)$$

$$= 30 \quad \text{Corregido}$$

B Código de corrección de bloque con doble paridad mostrando un error simple sin puntero

Figura 5.11. Ejemplos de codificación de bloques con doble paridad.

codificación única. Por ejemplo, a una palabra de datos de 4 bits (0, 1, 2 y 3) se le añaden tres bits de paridad (4, 5 y 6), con lo que se transmiten siete bits. Supóngase, por ejemplo, que los tres bits de paridad se definen de la siguiente manera:

- el bit de paridad cuarto está formado por la suma en módulo dos de los bits de datos: 1, 2 y 3,
- el bit de paridad quinto formado a partir de los bits de datos 0, 2 y 3
- y el bit de paridad sexto por los bits de datos 0, 1 y 3.

Así, la palabra de datos 1100 a la que se añade la paridad 110 se transmite como la palabra código de siete bits 1100110, obteniendo los códigos a partir de la tabla de datos que aparece representada en la Figura 5.12A, y que corresponden al algoritmo de cálculo representado en 5.12B. En la recepción, la localización de un error en la palabra dato se efectúa examinando

| X_0 | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 |
|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

A

X_0, X_1, X_2, X_3 Bits de datos
 $X_4 = X_1 + X_2 + X_3$ Bits de paridad (módulo 2)
 $X_5 = X_0 + X_2 + X_3$ (módulo 2)
 $X_6 = X_0 + X_1 + X_3$ (módulo 2)
 $X_0, X_1, X_2, X_3, X_4, X_5, X_6$ Palabra código transmitida

B

$$\begin{array}{r} X_1 + X_2 + X_3 + X_4 = 0 \\ X_0 + X_1 + X_2 + X_3 + X_5 = 0 \\ X_0 + X_1 + X_3 + X_6 = 0 \end{array} \quad \text{Algoritmo de decodificación}$$

$$\begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = H \end{array} \quad \text{Matriz de comprobación de paridad}$$

C

Ejemplo: 1100110 Palabra transmitida
 1000110 Palabra recibida

$$P_4 = X_1 + X_2 + X_3 = 0 + 0 + 0 = 0 \quad \text{Paridad de los datos recibidos}$$

$$P_5 = X_0 + X_2 + X_3 = 1 + 0 + 0 = 1$$

$$P_6 = X_0 + X_1 + X_3 = 1 + 0 + 0 = 1$$

Los síndromes se calculan mediante la suma en módulo 2 de la paridad de los datos recibidos y los bits de paridad recibidos:

$$P_4 = 0, \quad X_4 = 1, \quad 0 + 1 = 1 \quad (\text{Error})$$

$$P_5 = 1, \quad X_5 = 1, \quad 1 + 1 = 0 \quad (\text{Correcto})$$

$$P_6 = 1, \quad X_6 = 0, \quad 1 + 0 = 1 \quad (\text{Error})$$

El síndrome resultante del patrón de error $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ corresponde a la segunda columna de H

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{Entonces el bit } X_1 \text{ es erróneo}$$

D

Figura 5.12. En los códigos Hamming, el síndrome apunta a la localización del error. A. En este ejemplo con 4 bits de datos y 3 bits de paridad el código tiene una distancia Hamming 3. B. Los bits de paridad se forman en el codificador. C. Matriz de comprobación de paridad en el decodificador. D. Corrección de un error simple usando un síndrome que apunte a la posición del error.

qué paridades son las afectadas. Para esto se necesitan las ecuaciones de decodificación de la paridad, mostradas en la Figura 5.12C; y que podemos representar de forma matricial como H . Cada una de las filas de H representa una de las ecuaciones de codificación de la paridad, así que comprobando con H los datos recibidos podemos detectar en dónde ha aparecido un error. Con la paridad recibida y la calculada sobre los datos recibidos se genera (mediante suma en módulo 2) un síndrome, de forma que si su valor es cero la información se da como correcta y si es 1 es que ha ocurrido un error. El patrón de error resultante se compara con la matriz H para localizar el bit erróneo. En el ejemplo de la Figura 5.12D, si la palabra transmitida fue 1100110 y la recibida es 1000110, el síndrome detecta un error y genera un patrón de error 101. Si comparamos este patrón con la matriz H vemos que dicho patrón corresponde a la segunda columna, indicando esto que el error se ha producido en el bit 1. Este algoritmo constituye un código muy simple de corrección de errores, ya que sólo permite identificar y corregir un bit erróneo, pero es una muestra del potencial que pueden tener estos sistemas a poco que se complique la redundancia.

Respecto al diseño de este código tan particular, podemos observar otra característica interesante. Volviendo sobre la Figura 5.12A, se puede apreciar cómo en cada palabra de siete bits, cuatro son de datos y tres de paridad. Con siete bits tenemos $2^7 = 128$ combinaciones posibles, aunque de ellas solamente necesitamos para codificar la información $2^4 = 16$, así que 112 patrones son combinaciones ilegales, con lo que la recepción de cualquiera de ellos denotaría claramente la presencia de un error. De esta forma, ahora nos podríamos preguntar: ¿en cuántos bits tiene que cambiar una palabra de la tabla para convertirse en otra combinación válida? Por ejemplo, podemos observar cómo para que la palabra 0101010 se convierta en el código 0110011 deben cambiar de valor tres bits; y de forma similar, cómo cualquier código de la tabla requiere al menos un cambio de tres bits para convertirse en otra palabra válida. Este hecho es importante, puesto que esta falta de similitud en los datos no es sino otra capacidad adicional del código de corrección de errores. Por ejemplo, si recibimos la palabra 1110101, que no pertenece a las combinaciones válidas y que se diferencia de la palabra 1010101 (que sí que es un código válido) en un solo bit, se puede determinar de forma bastante fiable que ha ocurrido un error y que la palabra original era la que más se le parezca. En otras palabras, cualquier error que repercuta en un único bit permite al receptor corregirle, sin más que tomar como correcta la palabra que menos difiera de la recibida.

El número de bits que una palabra de las permitidas debe cambiar para convertirse en otra combinación permitida es lo que se conoce como distancia Hamming, o distancia mínima entre dos símbolos. En nuestro ejemplo, la distancia Hamming es tres, definiendo la capacidad de corrección del código. Una distancia de uno determina únicamente la singularidad de un código (código simple). Una distancia de dos permite la detección de un error simple (en un bit). Una distancia de tres permite la corrección de un error en un bit y la capacidad de detectar dos bits erróneos. Una distancia de cuatro puede permitir la corrección de un bit y la detección de dos incorrectos más, o la capacidad de detectar tres bits incorrectos. Una distancia de cinco permite la corrección de dos errores..., de forma que a medida que la distancia Hamming se incrementa, también lo hace proporcionalmente la capacidad de corrección en el código. Y viceversa, a mayor capacidad de corrección necesitada, mayor debe ser la distancia Hamming del código que elijamos. En general, para detectar un número de errores t_d la distancia del código debe ser mayor o igual a $t_d + 1$. En el caso de e bits perdidos o «borrados» («erasure») y que han sido detectados y puestos a cero, la distancia mínima debe ser mayor o igual a $e + 1$. Con el fin de poder corregir cualquier combinación de t_c errores (o menos), la distancia mínima requerida es de $2 \cdot t_c + 1$. Para la corrección de una combinación de errores, la distancia Hamming debe ser igual o mayor que $t_d + e + 2 \cdot t_c + 1$, siendo esto aplicable tanto para códigos orientados a bit como para códigos de palabras. Si en un código de bloques se introducen m bloques de paridad, la distancia mínima es mayor o igual a $m + 1$. En los códigos de distancia máxima separable (MDS), como los códigos Reed-Solomon, la distancia mínima es de $m + 1$.

Los códigos de bloques se caracterizan en términos de relación entre los datos de entrada y los de salida. Los datos son agrupados en símbolos, siendo un bit el tamaño más pequeño que puede tomar un símbolo. Un mensaje de k símbolos es utilizado para generar un símbolo de n bits, denominándose el código como (n, k) . Por ejemplo, si son necesarios 12 símbolos de entrada para construir un mensaje de 20 símbolos de salida, el código se denomina (12,20); o en otras palabras, se generan $n - k$ símbolos de paridad. La velocidad de salida de los datos V se define como k/n , y en este ejemplo V es 12/20.

Códigos convolucionales

Los códigos convolucionales o recurrentes difieren de los códigos de bloques en la forma de agrupar los datos para la codificación. En vez de dividir el mensaje en bloques de datos de k dígitos y generar n dígitos a partir de ellos, los códigos convolucionales no dividen los datos en bloques. Actúan tomando k dígitos del mensaje y generando un código de n dígitos, formado no sólo con los k dígitos del mensaje actual, sino con algunos dígitos anteriores almacenados en memoria y por tanto retardados; así el código de salida contiene una «historia de los datos previos». Un código convolucional utiliza $(N - 1)$ bloques de datos de k dígitos, limitando su actuación (en el tiempo) a un conjunto de N bloques (de nN dígitos) iguales a $n(m + 1)$, donde m es el número de retardos utilizado en el codificador. La velocidad de los datos V es k/n , donde tanto k como n suelen ser enteros de pequeño valor.

Como en los códigos de bloque lineales, efectuada la codificación se realiza la transmisión o almacenamiento de los datos. En su recuperación, el decodificador utiliza la generación de síndromes para la detección de palabras erróneas. Los registros de almacenamiento se utilizan en la implementación de los retardos requeridos tanto en el codificador como en el decodificador, mientras que la magnitud de dichos retardos determina la longitud límite temporal del código, término análogo a la longitud del bloque en los códigos de bloque. Un ejemplo de codificador convolucional es el representado en la Figura 5.13, donde seis células de retardo conforman la longitud límite. El resto de los parámetros son: $q = 2$, $V = 1/2$, $k = 1$, $n = 2$ y el polinomio característico es $x^6 + x^5 + x^2 + 1$. Como se muestra en el diagrama, el mensaje pasa a través del codificador de forma que los bits anteriores afectan al código de salida del sistema.

Otro ejemplo de código convolucional, introducido por Andrew Viterbi, aparece en la Figura 5.14A. Los códigos superiores se forman con el polinomio $x^2 + x + 1$, mientras que a los inferiores se les aplica $x^2 + 1$. La secuencia de datos entra en el circuito por la izquierda desplazándose hacia la derecha de bit en bit. Las dos secuencias son generadas a partir de la suma en módulo 2 de la secuencia original, siendo multiplexadas para conformar la cadena de bits de salida. El código resultante tiene una memoria de dos, puesto que sobre el actual bit de entrada

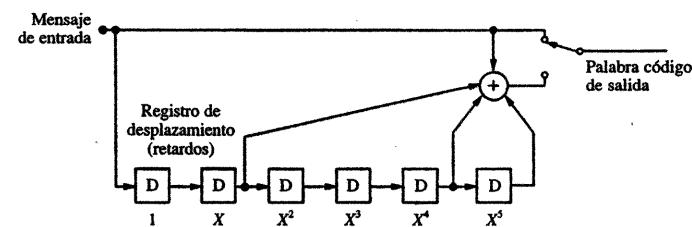


Figura 5.13. Un condicionador convolucional con seis unidades de retardo.

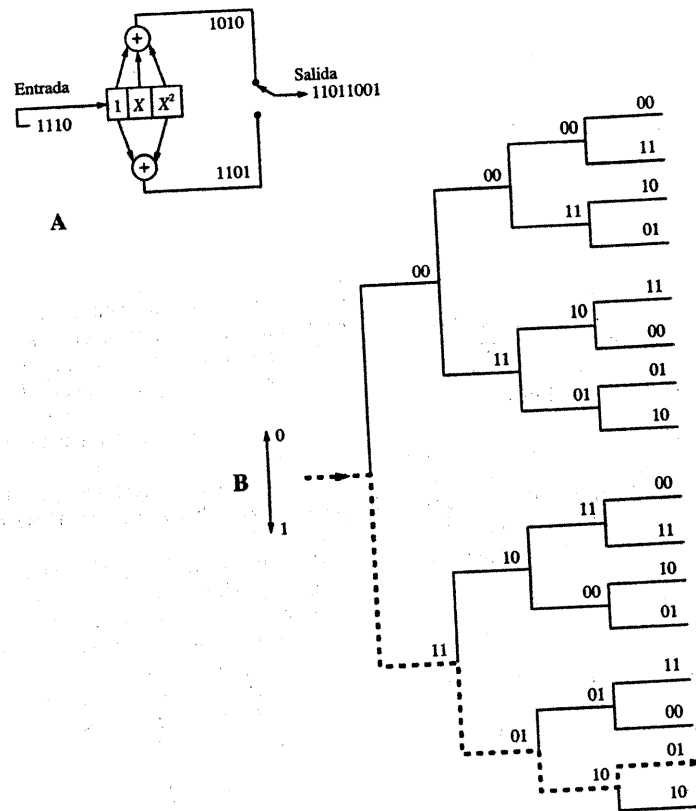


Figura 5.14. Un ejemplo de codificación convolucional. A. Codificador convolucional con $k = 3$ y $r = 1/2$. B. Diagrama de árbol del código convolucional.

actúan los dos bits que le precedieron. Para cada bit de entrada hay dos bits de salida, siendo la velocidad $V = 1/2$ y la longitud máxima de $k = 3$.

Un código convolucional puede ser analizado con un diagrama de árbol como se muestra en la Figura 5.14B. Representa las primeras cuatro secuencias de un árbol infinito, cuyos nodos están espaciados n dígitos y con $2k$ ramificaciones en cada uno. Cada rama es un código de bloque de n dígitos, y se corresponde con un bloque de mensaje de k dígitos. Cualquier secuencia de n dígitos, y se representa como un camino a través del árbol. Por ejemplo, la secuencia del código es representada como un camino a través del árbol. Si la entrada es 0 el código del símbolo se anterior ejemplo puede ser trazada a través del árbol. Si la entrada es uno, mientras que si el código de entrada obtiene subiendo un paso en la siguiente ramificación, mientras que si el código de entrada es uno, se toma la ramificación hacia abajo. Así, el mensaje de entrada dicta el camino que se sigue a través del árbol, asociando cada símbolo de entrada a una instrucción. La secuencia de selecciones en los nodos forma el código de salida. Para el ejemplo anterior, el código de entrada 1110 genera un código de salida 11011001. Tras la reproducción o recuperación de la señal,

los datos se decodifican secuencialmente y los errores son detectados y corregidos comparando la secuencia recibida con las posibles. La comparación de los datos recibidos se realiza en la decisión, y por tanto rama a rama. En la decodificación, el camino que sigue la señal está dado por el seguimiento del algoritmo, buscando la secuencia transmitida que más se asemeje con la secuencia recibida.

Otro código convolucional es el sugerido por Thosi Doi, mostrando el codificador en la Figura 5.15. Utiliza cuatro retardos de palabra, generando las palabras de paridad cada cuatro palabras de entrada, con lo que cada palabra de paridad contiene información de las ocho palabras anteriores. La longitud máxima del código es 14. Los códigos convolucionales son sencillos y baratos de implementar y trabajan muy bien en condiciones de elevada tasa de errores. Su desventaja fundamental es la propagación de errores, de forma que un símbolo que no ha podido ser corregido, introduce errores en la decodificación de los símbolos que vienen a continuación.

ENTRELAZADO

La corrección de errores depende de la habilidad del algoritmo en obtener la reconstrucción de los datos originales a partir de los datos no válidos recuperados. Cuando un error se manifiesta de forma continuada, como es el caso de los errores de ráfaga, se pierden tanto los datos como la redundancia, con lo que la recuperación es imposible. Para evitar este hecho, se efectúa un entrelazado o dispersión de los datos antes de realizar la transmisión o almacenamiento de la cadena de datos. Así, si se produce un error de ráfaga que afecte a un conjunto de datos contiguos, al desentrelazarse la cadena en la reproducción y volver los datos a su secuencia original los errores aparecerán de forma dispersa. Ahora, entre los datos correctos y la paridad que los circunda es fácil que el algoritmo de recuperación de errores sea capaz de reconstruir todos los datos originales a partir de esos pocos que eran válidos. La Figura 5.16 muestra un ejemplo de entrelazado y desentrelazado, donde aparece un error de ráfaga introducido en el almacenamiento o transmisión. A continuación del desentrelazado, los errores aparecen dispersos, con lo que la corrección de errores es más sencilla.

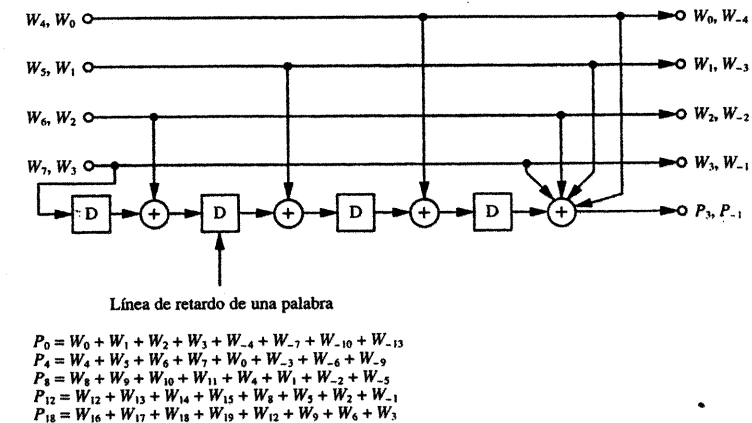


Figura 5.15. Un ejemplo de codificador convolucional, generando una palabra de chequeo por cada cuatro palabras de datos (Doi).

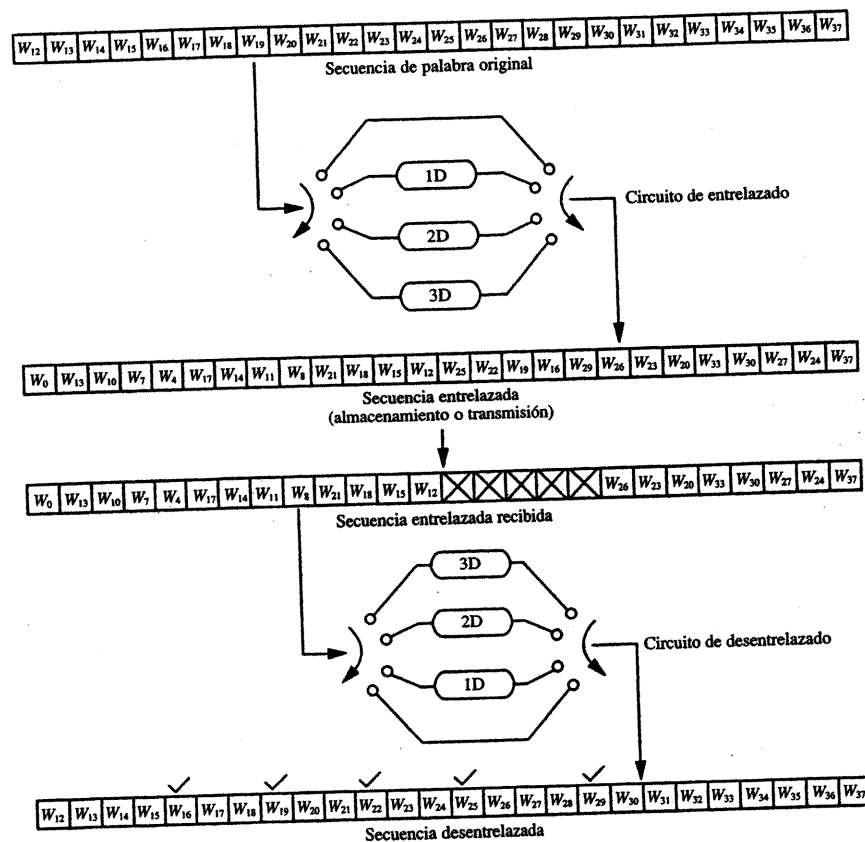


Figura 5.16. Cero, uno, dos y tres palabras de retardo permiten realizar el entrelazado y desentrelazado para la dispersión de los datos previa a la corrección (Doi).

El proceso del entrelazado introduce una gran ventaja en todo el proceso, ya que sin él, la magnitud de redundancia a introducir sería fijada por el tamaño del mayor error de ráfaga que quisiéramos corregir. Con el entrelazado, el mayor error que puede aparecer en un bloque está limitado al tamaño de las secciones entrelazadas. Así, la cantidad de redundancia a introducir no depende de la magnitud del posible error de ráfaga, sino del tamaño o separación entre las secciones entrelazadas.

El entrelazado y retardo de las muestras provoca una dispersión efectiva de los datos, y muchos sistemas de comprobación de paridad trabajan de forma adecuada con una palabra errónea por bloque. Un error de ráfaga viola esta regla, pero el entrelazado y desentrelazado de datos puede dispersar los errores de forma que recaigan en una sola palabra incorrecta por bloque. Así, el entrelazado incrementa notablemente la posibilidad de corrección en los códigos de bloque frente a los errores de ráfaga. Hasta ahora hemos comentado el entrelazado de palabras o de bloque, pero el entrelazado de bits podría tener las mismas prestaciones, permitiendo que

los errores de ráfaga fueran mejor tratados al convertirlos en errores de bit. Como último detalle, considerar que cualquier proceso de entrelazado requiere una memoria intermedia lo suficientemente grande como para almacenar los datos y distribuirlos en el entrelazado y desentrelazado.

Intercalación cruzada

El entrelazado puede resultar inadecuado cuando a los errores de ráfaga le acompañan errores de bit. Aunque los errores de ráfaga sean dispersados, los errores de bit incrementan los errores en cada palabra, con lo que se puede provocar el desbordamiento del algoritmo de corrección de errores. Una solución es generar dos códigos de corrección de errores, separados por un entrelazado por un retardo entre ellos. Cuando los códigos de bloque son estructurados en dos dimensiones (filas y columnas), el código es llamado código producto o código cruzado, siendo un ejemplo de ellos el utilizado en el DVD. La distancia mínima es el producto de las distancias de cada código. Cuando los dos códigos de bloques están separados ambos por retardos y los diferentes datos son entrelazados dentro del bloque, se denominan códigos de intercalación cruzada. En pocas palabras, un código de intercalación cruzada comprende dos (o más) códigos de bloque ensamblados con una estructura convolucional, tal y como muestra la Figura 5.17. El sistema es muy eficiente debido a que los síndromes de cada uno de los códigos pueden ser utilizados para apuntar a los errores, los cuales serán corregidos mediante otro código. Debido a que la situación del error es conocida, la mejora introducida en la capacidad de corrección es elevada. Por ejemplo, un error aleatorio de bit puede ser corregido por el código de entrelazado, mientras que los errores de ráfaga son corregidos tras el desentrelazado. Cuando ambos códigos permiten la corrección de errores de bit dentro de un símbolo, el código resultante es conocido como código de intercalación cruzada (CIC). En el sistema de disco compacto se utiliza el código Reed-Solomon y el algoritmo es conocido como código de intercalación cruzada Reed-Solomon (Cross Interleave Reed-Solomon Code, CIRC).

Un ejemplo de codificador CIC sugerido por Toshi-Doi aparece mostrado en la Figura 5.18. Las unidades de retardo producen el entrelazado, mientras que los sumadores en módulo 2 generan códigos de corrección de bit. Se generan dos palabras de paridad (denominadas P y Q), permitiendo la corrección de dos errores de bit de una forma muy eficiente. Los errores en tres palabras también pueden ser corregidos; sin embargo, los errores en cuatro palabras producen un error doble en dos palabras, resultando imposible la corrección. El CIC posee la elevada eficacia de los códigos convolucionales, pero sin la desventaja de la propagación de errores, debido a que un error no corregible en una secuencia siempre se convierte en un error de palabra en la secuencia siguiente, siendo entonces fácilmente corregido. Un sistema que utiliza un código CIC es el formato DASH de grabación magnética.

CÓDIGOS REED-SOLOMON

Los códigos Reed-Solomon fueron desarrollados en 1960 por Irving Reed y Gustave Solomon en los laboratorios Lincoln del MIT. Son un ejemplo de una importante subclase de los códigos BCH (Bose-Chaudhuri-Hocquenghem), que a su vez son un tipo de códigos Hamming. Los códigos Reed-Solomon son códigos cíclicos, que pertenecen a los códigos de corrección de errores múltiples. Definen símbolos de n -bit bytes, que transforman en $2^n - 1$ palabras código de n -bit bytes. Si el patrón del error afecta a s bytes en una palabra código, se necesitan $2s$ bytes para efectuar la corrección de errores. Así, nos quedan disponibles para datos $2^n - 1 - 2s$ bytes. Cuando se combina con la intercalación cruzada, el código Reed-Solomon es uno de los más efectivos para aplicaciones de audio.

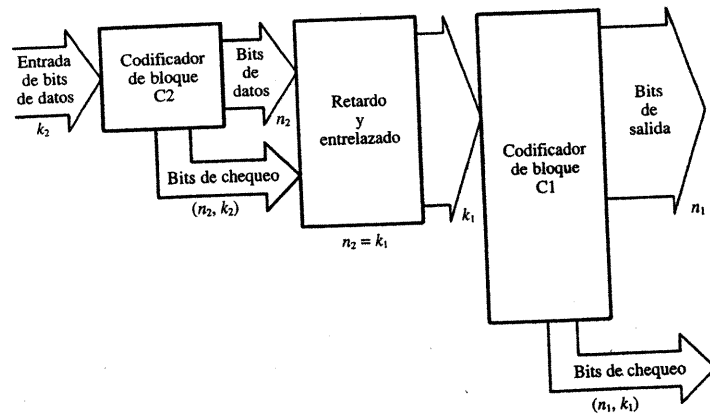


Figura 5.17. Un codificador de intercalación cruzada. Los síndromes del primer bloque que se emplean como punteros de error del segundo bloque. En el formato CD, $k_2 = 24$, $n_2 = 28$, $k_1 = 28$, $n_1 = 32$. Los codificadores C_1 y C_2 son del tipo Reed-Solomon.

Los códigos Reed-Solomon emplean para la codificación y decodificación de bloques de datos exclusivamente unos polinomios derivados de un campo de las matemáticas denominado campos de Galois. Los campos de Galois son denominados así en honor del extraordinario y atormentado genio de las matemáticas que fue Evariste Galois (muerto en un duelo a la edad de 20 años), y comprenden un número finito de elementos con un conjunto de propiedades especiales. Cada multiplicación o adición se pueden aplicar para la combinación de elementos, sien-

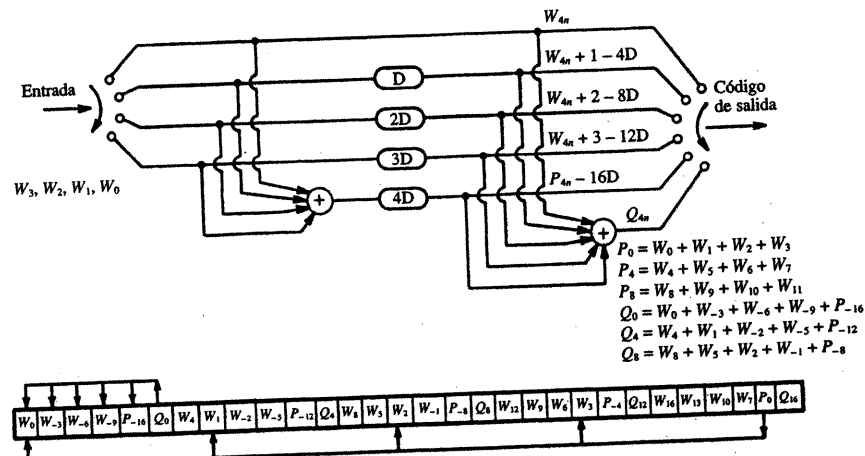


Figura 5.18. Un ejemplo de codificador de intercalación cruzada (CIC) y su secuencia de salida (Doi).

do el resultado de estas operaciones siempre un elemento perteneciente al campo. Por ejemplo, cuando un elemento es elevado a una potencia alta, el resultado siempre es un elemento que pertenece al campo. Estos campos generalmente existen cuando el número de elementos es un número primo o una potencia de un número primo. Además, se debe cumplir que exista al menos un elemento llamado «primitiva», tal que cualquier otro elemento del campo pueda expresarse como una potencia entera de ésta.

En la corrección de errores, los campos de Galois aportan un conjunto de códigos altamente estructurados, lo que simplifica enormemente la implementación del código. En los códigos Reed-Solomon (RS) los datos se construyen como símbolos miembros del campo de Galois empleado en la codificación, siendo por tanto códigos BCH no binarios. Esta característica permite aumentar la distancia mínima entre los códigos de la entrada y salida dentro de un bloque de datos de una longitud dada. La distancia mínima es el número de símbolos no binarios en los que la secuencia difiere, siendo ahora $d = n - k + 1$. El tamaño del campo de Galois, el cual determina a la postre el número de símbolos en el código, está basado en el número de bits que constituyen un símbolo, siendo lo usual emplear símbolos de ocho bits. En este caso, el código contiene $2^8 - 1$ (255) símbolos de 8 bits, y uno de los polinomios utilizado como primitiva es $x^8 + x^4 + x^3 + x^2 + 1$.

Como en otros códigos, el Reed-Solomon produce polinomios cuyas raíces permiten localizar los errores y generan síndromes para corregirlos. Por ejemplo, el código puede utilizar la palabra de entrada para generar dos tipos de paridad P y Q . La paridad P es una suma en módulo 2 de los símbolos, mientras que la paridad Q multiplica cada palabra de entrada por una potencia distinta de la primitiva utilizada en el campo. Si un símbolo es erróneo, la paridad P entrega un síndrome S_1 distinto de cero. La paridad Q produce un síndrome S_2 , cuyo valor es S_1 elevado a una potencia cuyo valor depende de la posición del error, por lo que comprobando la relación entre S_1 y S_2 el código RS puede localizar el error. Cuando se ha apuntado a un símbolo que es erróneo, S_2 es igual a S_1 multiplicado por el elemento elevado a la posición del error. La corrección se puede realizar añadiendo el valor de S_1 a la posición apuntada. Como alternativa, si se conoce la posición de dos errores mediante dos punteros de detección, entonces se pueden corregir los dos errores. Por ejemplo, si el segundo y tercer símbolos son marcados como erróneos, entonces S_2 es la suma de los errores multiplicada por el cuadrado y el cubo del generador.

Para ilustrar la operación del código Reed-Solomon, considerar un campo de Galois de 2^8 , comprendiendo símbolos de tres bits. En este código, α es el elemento considerado como primitiva, siendo solución de la ecuación:

$$F(x) = x^3 + x + 1 = 0$$

tal que el polinomio irreducible puede ponerse de la forma:

$$\alpha^3 + \alpha + 1 = 0$$

donde $+$ indica suma en módulo 2. Los elementos del código (símbolos) pueden ponerse de la forma:

$$\begin{aligned} 000 &= &= 0 \\ 001 &= &+ 1 = 1 \\ 010 &= &+ x = x \\ 011 &= &+ x + 1 = x + 1 \\ 100 &= x^2 &= x^2 \\ 101 &= x^2 &+ 1 = x^2 + 1 \\ 110 &= x^2 + x &= x^2 + x \\ 111 &= x^2 + x + 1 &= x^2 + x + 1 \end{aligned}$$

Como $\alpha = x$, aplicando las propiedades del módulo 2 (donde $1 + 1 = \alpha + \alpha = \alpha^2 + \alpha^2 = 0$) y sabiendo que es un campo de Galois, podemos crear una representación logarítmica de los elementos irreducibles del campo, donde las posiciones de los bits indican las posiciones de los polinomios:

$$\begin{aligned}
 0 &= 000 \\
 1 &= 001 \\
 \alpha &= 010 \\
 \alpha^2 &= 100 \\
 \alpha^3 &= \alpha + 1 = 011 \\
 \alpha^4 &= \alpha \cdot \alpha^3 = \alpha(\alpha + 1) = \alpha^2 + \alpha = 110 \\
 \alpha^5 &= \alpha^2 + \alpha + 1 = 111 \\
 \alpha^6 &= \alpha \cdot \alpha^5 = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha \\
 &= \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1 = 101 \\
 \alpha^7 &= \alpha(\alpha^2 + 1) = \alpha^2 + \alpha = \alpha + 1 + \alpha = 1 = 001 = 1
 \end{aligned}$$

De esta forma, todos los posibles símbolos de tres bits pueden ser expresados como elementos del campo ($0, 1 = \alpha^7, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$ y α^6), donde α es la primitiva (010). Los elementos pueden multiplicarse sin más que incrementar el exponente, siendo el resultado siempre otro elemento del campo. Por ejemplo:

$$\begin{aligned}
 \alpha \cdot \alpha &= \alpha^2 = (010)(010) = 100 \\
 1 \cdot \alpha^2 &= \alpha^2 = (001)(100) = 100 \\
 \alpha^2 \cdot \alpha^3 &= \alpha^5 = (100)(011) = 111
 \end{aligned}$$

La tabla de la multiplicación completa para este ejemplo de un código GF(2³) aparece desarrollada en la Figura 5.19, observando cómo quedan los elementos en módulo α^7 . Por ejemplo, $\alpha^4 \cdot \alpha^6$ es igual a α^{10} , que lo es a su vez a α^3 . Utilizando la tabla de multiplicación y los polinomios irreducibles, se puede construir la tabla de corrección. Supóngase que A, B, C y D son símbolos de datos, y P y Q símbolos de paridad. El código RS debe satisfacer las siguientes ecuaciones:

$$\begin{aligned}
 A + B + C + D + P + Q &= 0 \\
 \alpha^6 A + \alpha^5 B + \alpha^4 C + \alpha^3 D + \alpha^2 P + \alpha^1 Q &= 0
 \end{aligned}$$

Utilizando las leyes de los productos dadas, podemos resolver estas ecuaciones para obtener:

$$\begin{aligned}
 P &= \alpha^1 A + \alpha^2 B + \alpha^5 C + \alpha^3 D \\
 Q &= \alpha^3 A + \alpha^5 B + \alpha^4 C + \alpha^1 D
 \end{aligned}$$

Por ejemplo, dada la tabla de polinomios irreducibles, si:

$$\begin{aligned}
 A &= 001 = 1 \\
 B &= 101 = \alpha^6 \\
 C &= 011 = \alpha^3 \\
 D &= 100 = \alpha^2
 \end{aligned}$$

| Bits | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|------|----------------|-----|------------|------------|------------|------------|------------|------------|------------|
| | Elementos | 0 | 1 | α | α^3 | α^2 | α^6 | α^4 | α^5 |
| 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 001 | $\alpha^7 = 1$ | 0 | 1 | α | α^3 | α^2 | α^6 | α^4 | α^5 |
| 010 | α | 0 | α | α^2 | α^6 | α^3 | 1 | α^5 | α^4 |
| 011 | α^3 | 0 | α^3 | α^4 | α^6 | α^5 | α^2 | 1 | α |
| 100 | α^2 | 0 | α^2 | α^3 | α^5 | α^4 | α | α^6 | 1 |
| 101 | α^6 | 0 | α^6 | 1 | α^2 | α | α^5 | α^3 | α^4 |
| 110 | α^4 | 0 | α^4 | α^5 | 1 | α^6 | α^3 | α | α^2 |
| 111 | α^5 | 0 | α^5 | α^6 | α | 1 | α^4 | α^2 | α^3 |

Figura 5.19. La tabla de multiplicar de un código GF(2³) con $F(x) = x^3 + x + 1$ y como primitiva el elemento 010.

podemos resolverla para obtener P y Q utilizando la tabla de la multiplicación:

$$\begin{aligned}
 P &= \alpha^1 \cdot 1 + \alpha^2 \cdot \alpha^6 + \alpha^5 \cdot \alpha^3 + \alpha^3 \cdot \alpha^2 = \alpha + \alpha + \alpha + \alpha^5 = \\
 &= \alpha + \alpha + \alpha + (\alpha^2 + \alpha + 1) = \alpha^2 + 1 = 101 \\
 Q &= \alpha^3 \cdot 1 + \alpha^6 \cdot \alpha^6 + \alpha^4 \cdot \alpha^3 + \alpha^1 \cdot \alpha^2 = \alpha^3 + \alpha^5 + 1 + \alpha^3 = \\
 &= (\alpha + 1) + (\alpha^2 + \alpha + 1) + 1 + (\alpha + 1) = \alpha^2 + \alpha = 110
 \end{aligned}$$

Con lo que:

$$\begin{aligned}
 P &= 101 = \alpha^6 \\
 Q &= 110 = \alpha^4
 \end{aligned}$$

Los errores recibidos en los datos pueden ser corregidos usando síndromes, donde las primas (') indican los datos recibidos:

$$\begin{aligned}
 S_1 &= A' + B' + C' + D' + P' + Q' \\
 S_2 &= \alpha^6 A' + \alpha^5 B' + \alpha^4 C' + \alpha^3 D' + \alpha^2 P' + \alpha^1 Q'
 \end{aligned}$$

Si cada posible patrón de error es expresado por E_i , podemos poner las ecuaciones como:

$$\begin{aligned}
 S_1 &= E_A + E_B + E_C + E_D + E_P + E_Q \\
 S_2 &= \alpha^6 E_A + \alpha^5 E_B + \alpha^4 E_C + \alpha^3 E_D + \alpha^2 E_P + \alpha^1 E_Q
 \end{aligned}$$

Como $\alpha = x$, aplicando las propiedades del módulo 2 (donde $1 + 1 = \alpha + \alpha = \alpha^2 + \alpha^2 = 0$) y sabiendo que es un campo de Galois, podemos crear una representación logarítmica de los elementos irreducibles del campo, donde las posiciones de los bits indican las posiciones de los polinomios:

$$\begin{aligned}
 0 &= 000 \\
 1 &= 001 \\
 \alpha &= 010 \\
 \alpha^2 &= 100 \\
 \alpha^3 &= \alpha + 1 = 011 \\
 \alpha^4 &= \alpha \cdot \alpha^3 = \alpha(\alpha + 1) = \alpha^2 + \alpha = 110 \\
 \alpha^5 &= \alpha^2 + \alpha + 1 = 111 \\
 \alpha^6 &= \alpha \cdot \alpha^5 = \alpha(\alpha^2 + \alpha + 1) = \alpha^3 + \alpha^2 + \alpha \\
 &= \alpha + 1 + \alpha^2 + \alpha = \alpha^2 + 1 = 101 \\
 \alpha^7 &= \alpha(\alpha^2 + 1) = \alpha^2 + \alpha = \alpha + 1 + \alpha = 1 = 001 = 1
 \end{aligned}$$

De esta forma, todos los posibles símbolos de tres bits pueden ser expresados como elementos del campo ($0, 1 = \alpha^7, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$ y α^6), donde α es la primitiva (010). Los elementos pueden multiplicarse sin más que incrementar el exponente, siendo el resultado siempre otro elemento del campo. Por ejemplo:

$$\begin{aligned}
 \alpha \cdot \alpha &= \alpha^2 = (010)(010) = 100 \\
 1 \cdot \alpha^2 &= \alpha^2 = (001)(100) = 100 \\
 \alpha^2 \cdot \alpha^3 &= \alpha^5 = (100)(011) = 111
 \end{aligned}$$

La tabla de la multiplicación completa para este ejemplo de un código GF(2³) aparece desarrollada en la Figura 5.19, observando cómo quedan los elementos en módulo α^7 . Por ejemplo, $\alpha^4 \cdot \alpha^6$ es igual a α^{10} , que lo es a su vez a α^3 . Utilizando la tabla de multiplicación y los polinomios irreducibles, se puede construir la tabla de corrección. Supóngase que A, B, C y D son símbolos de datos, y P y Q símbolos de paridad. El código RS debe satisfacer las siguientes ecuaciones:

$$\begin{aligned}
 A + B + C + D + P + Q &= 0 \\
 \alpha^6 A + \alpha^5 B + \alpha^4 C + \alpha^3 D + \alpha^2 P + \alpha^1 Q &= 0
 \end{aligned}$$

Utilizando las leyes de los productos dadas, podemos resolver estas ecuaciones para obtener:

$$\begin{aligned}
 P &= \alpha^1 A + \alpha^2 B + \alpha^5 C + \alpha^3 D \\
 Q &= \alpha^3 A + \alpha^6 B + \alpha^4 C + \alpha^1 D
 \end{aligned}$$

Por ejemplo, dada la tabla de polinomios irreducibles, si:

$$\begin{aligned}
 A &= 001 = 1 \\
 B &= 101 = \alpha^6 \\
 C &= 011 = \alpha^3 \\
 D &= 100 = \alpha^2
 \end{aligned}$$

| Bits | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|------|----------------|-----|------------|------------|------------|------------|------------|------------|------------|
| | Elementos | 0 | 1 | α | α^3 | α^2 | α^6 | α^4 | α^5 |
| 000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 001 | $\alpha^7 = 1$ | 0 | 1 | α | α^2 | α^2 | α^6 | α^4 | α^5 |
| 010 | α | 0 | α | α^2 | α^6 | α^3 | 1 | α^3 | α^6 |
| 011 | α^3 | 0 | α^3 | α^4 | α^6 | α^5 | α^2 | 1 | α |
| 100 | α^2 | 0 | α^2 | α^3 | α^5 | α^4 | α | α^6 | 1 |
| 101 | α^6 | 0 | α^6 | 1 | α^2 | α | α^5 | α^3 | α^4 |
| 110 | α^4 | 0 | α^4 | α^5 | 1 | α^6 | α^3 | α | α^2 |
| 111 | α^5 | 0 | α^5 | α^6 | α | 1 | α^4 | α^2 | α^3 |

Figura 5.19. La tabla de multiplicar de un código GF(2³) con $F(x) = x^3 + x + 1$ y como primitiva el elemento 010.

podemos resolverla para obtener P y Q utilizando la tabla de la multiplicación:

$$\begin{aligned}
 P &= \alpha^1 \cdot 1 + \alpha^2 \cdot \alpha^6 + \alpha^5 \cdot \alpha^3 + \alpha^3 \cdot \alpha^2 = \alpha + \alpha + \alpha + \alpha^5 = \\
 &= \alpha + \alpha + \alpha + (\alpha^2 + \alpha + 1) = \alpha^2 + 1 = 101 \\
 Q &= \alpha^3 \cdot 1 + \alpha^6 \cdot \alpha^6 + \alpha^4 \cdot \alpha^3 + \alpha^1 \cdot \alpha^2 = \alpha^3 + \alpha^5 + 1 + \alpha^3 = \\
 &= (\alpha + 1) + (\alpha^2 + \alpha + 1) + 1 + (\alpha + 1) = \alpha^2 + \alpha = 110
 \end{aligned}$$

Con lo que:

$$\begin{aligned}
 P &= 101 = \alpha^6 \\
 Q &= 110 = \alpha^4
 \end{aligned}$$

Los errores recibidos en los datos pueden ser corregidos usando síndromes, donde las primas (') indican los datos recibidos:

$$\begin{aligned}
 S_1 &= A' + B' + C' + D' + P' + Q' \\
 S_2 &= \alpha^6 A' + \alpha^5 B' + \alpha^4 C' + \alpha^3 D' + \alpha^2 P' + \alpha^1 Q'
 \end{aligned}$$

Si cada posible patrón de error es expresado por E_i , podemos poner las ecuaciones como:

$$\begin{aligned}
 S_1 &= E_A + E_B + E_C + E_D + E_P + E_Q \\
 S_2 &= \alpha^6 E_A + \alpha^5 E_B + \alpha^4 E_C + \alpha^3 E_D + \alpha^2 E_P + \alpha^1 E_Q
 \end{aligned}$$

87

Con las siguientes condiciones:

Si no hay error, $S_1 = S_2 = 0$

Si el símbolo A' es erróneo, $S_1 = E_A$ y $S_2 = \alpha^6 S_1$.

Si el símbolo B' es erróneo, $S_1 = E_B$ y $S_2 = \alpha^5 S_1$.

Si el símbolo C' es erróneo, $S_1 = E_C$ y $S_2 = \alpha^4 S_1$.

Si el símbolo D' es erróneo, $S_1 = E_D$ y $S_2 = \alpha^3 S_1$.

Si el símbolo P' es erróneo, $S_1 = E_P$ y $S_2 = \alpha^2 S_1$.

Si el símbolo Q' es erróneo, $S_1 = E_Q$ y $S_2 = \alpha^1 S_1$.

En otras palabras, un error produce un síndrome distinto de cero, mientras que el valor de los símbolos erróneos puede ser determinado por la diferencia ponderada de S_1 y S_2 . Como la ponderación es diferente para cada palabra, este hecho permite la corrección de una palabra incorrecta. Los errores de bit que recaigan sobre dos palabras pueden corregirse, debido a que existen dos ecuaciones con dos incógnitas. Por ejemplo, si los datos recibidos fueran:

$$A' = 001 = 1$$

$$B' = 101 = \alpha^6$$

$$C' = 001 = 1 \text{ (error)}$$

$$D' = 100 = \alpha^2$$

$$P' = 101 = \alpha^6$$

$$Q' = 110 = \alpha^4$$

Podemos realizar el cálculo de los síndromes (teniendo en cuenta que $1 + 1 = \alpha + \alpha = \alpha^2 + \alpha^2 = 0$):

$$\begin{aligned} S_1 &= 1 + \alpha^6 + 1 + \alpha^2 + \alpha^6 + \alpha^4 = \\ &= 1 + (\alpha^2 + 1) + 1 + \alpha^2 + (\alpha^2 + 1) + (\alpha^2 + \alpha) = \\ &= \alpha = 010 \end{aligned}$$

$$\begin{aligned} S_2 &= \alpha^6 \cdot 1 + \alpha^5 \cdot \alpha^6 + \alpha^4 \cdot 1 + \alpha^3 \cdot \alpha^2 + \alpha^2 \cdot \alpha^6 + \alpha^1 \cdot \alpha^4 = \\ &= \alpha^6 + \alpha^4 + \alpha^4 + \alpha^5 + \alpha + \alpha^5 = \\ &= (\alpha^2 + 1) + (\alpha^2 + \alpha) + (\alpha^2 + \alpha) + (\alpha^2 + \alpha + 1) + \alpha + (\alpha^2 + \alpha + 1) = \\ &= \alpha^2 + \alpha + 1 = \alpha^5 = 111 \end{aligned}$$

Ya que $S_2 = \alpha^4 S_1$ (puesto que $\alpha^5 = \alpha^4 \cdot \alpha$), el símbolo C' debe ser erróneo, y ya que $S_1 = E_C = 010$, $C = C' + E_C = 001 + 010 = 011$, con lo que queda corregido el error.

En la práctica, los polinomios utilizados en el CCRC son:

$$P = \alpha^6 A + \alpha^1 B + \alpha^2 C + \alpha^5 D + \alpha^3 E$$

$$Q = \alpha^2 A + \alpha^3 B + \alpha^6 C + \alpha^4 D + \alpha^1 E$$

y los síndromes:

$$S_1 = A' + B' + C' + D' + E' + P' + Q'$$

$$S_2 = \alpha^7 A' + \alpha^6 B' + \alpha^5 C' + \alpha^4 D' + \alpha^3 E' + \alpha^2 P' + \alpha^1 Q'$$

Debido a que los códigos Reed-Solomon son particularmente efectivos en la corrección de errores de ráfaga, en audio se utilizan asociados a sistemas de detección de errores como el CCRC y al entrelazado. El código Reed-Solomon se utiliza en sistemas como el disco compacto, DAT, DVD, radiodifusión vía satélite, radiodifusión terrestre y televisión digital.

CÓDIGO REED-SOLOMON DE INTERCALACIÓN CRUZADA

En el sistema de disco compacto se utiliza un código Reed-Solomon (de distancia mínima de cinco) de corrección de un cuádruple error de bit (error doble), denominándosele habitualmente código Reed-Solomon de intercalación cruzada («Cross-Interleave Reed-Solomon Code», CIRC). Consiste en la aplicación secuencial de dos códigos Reed-Solomon en cascada, asociados a un proceso de entrelazado. La codificación consiste en tomar los datos, entrelazarlos, aplicarlos a un codificador denominado C2, entrelazar de nuevo su salida y volverlos a codificar con el codificador C1.

El codificador C2 es un (28,24), es decir, un código con 24 símbolos de entrada y 28 símbolos de salida, introduciendo por tanto cuatro símbolos de paridad. El codificador C1 es (32,28) aceptando como entrada 28 símbolos, a partir de los cuales calcula los 32 símbolos de salida, que vuelven a incluir a su vez otros cuatro de paridad. En ambos casos, puesto que se utilizan símbolos de 8 bits, estamos empleando un campo de Galois de dimensión 2^8 , utilizando como primitiva para los cálculos el polinomio $x^8 + x^4 + x^3 + x^2 + 1$. La distancia mínima es de 5, siendo posible corregir hasta cuatro símbolos si se conoce la posición del error y dos símbolos si su posición es desconocida.

Utilizando una combinación de paridad y entrelazado se puede conseguir hacer el sistema más robusto frente a los errores propios del soporte, codificando los datos antes de su almacenamiento y decodificándolos tras la reproducción. El algoritmo de codificación CIRC se muestra en la Figura 5.20, pudiendo observar su similitud con la estructura general de estos sistemas mostrada en la Figura 5.17. Con la utilización de este algoritmo de codificación y la intercalación cruzada se generan cuatro símbolos de paridad P y otros cuatro de paridad Q .

El codificador CIRC toma 24 símbolos de ocho bits y se introduce un retardo de dos símbolos entre las muestras pares y la impares. Puesto que las muestras pares son retardadas dos bloques de datos con respecto a las impares, incluso con la pérdida de dos bloques de datos consecutivos, es posible aplicar interpolación para restablecer la señal. Los símbolos asociados a las muestras pares e impares son ahora entrelazados, lo que ayudará en el proceso de ocultación de errores. La palabra de entrada de 24 bytes pasa al codificador C2, el cual genera 4 símbolos de paridad Q . Ésta se utilizará para la corrección de un símbolo erróneo, o de hasta cuatro bits perdidos en una palabra. El motivo de disponer los símbolos de paridad en el centro del bloque persigue simplemente el mayor alejamiento posible entre muestras pares e impares, ya que posibilita la interpolación incluso con largos errores de ráfaga.

En la intercalación cruzada, cada uno de los 28 símbolos se retarda un período diferente. Estos períodos corresponden a múltiplos enteros de cuatro bloques, de forma que este entrelazado convolucional provoca el almacenamiento de una palabra C2 en 28 bloques de datos diferentes, estando a su vez separadas hasta 109 bloques. De esta forma, el conjunto de datos es entrelazado en dos direcciones, y como los retardos son relativamente grandes y diferentes, vuelve a facilitar la corrección de errores de ráfaga. Los veintiocho símbolos (que ya provienen de 28 bloques distintos) alimentan ahora el codificador C1, obteniéndose cuatro símbolos de paridad P . Con ella se puede corregir un símbolo erróneo y detectar y marcar dos o tres símbolos para que la paridad Q pueda corregirlos.

Por último, otra nueva etapa de entrelazado introduce un retardo de un símbolo sobre símbolos alternos de la salida del codificador C1, dispersando los datos de salida en dos bloques adyacentes. Con esto se consigue que los errores de bit aleatorios no puedan afectar a más de

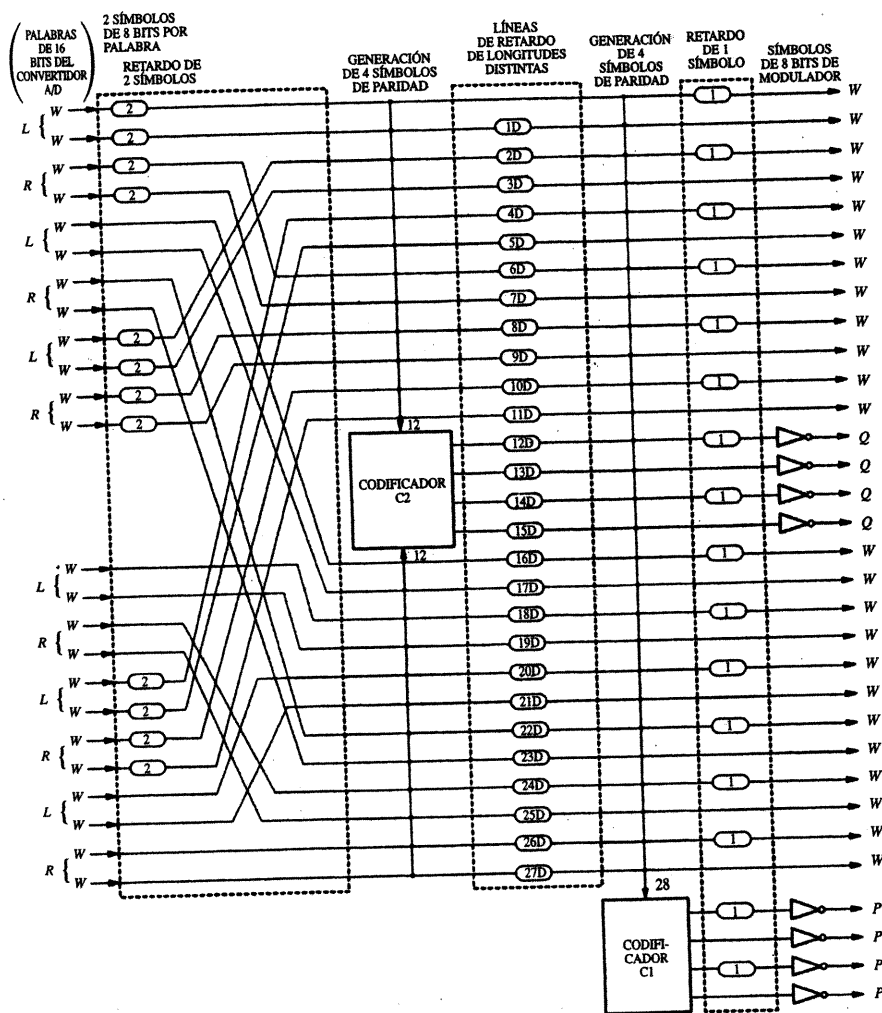


Figura 5.20. Algoritmo de codificación CIRC.

un símbolo en una misma palabra, aunque haya dos símbolos erróneos contiguos en un mismo bloque. Como último detalle, los símbolos de la paridad P y Q no se almacenan como tal sino invertidos, de forma que cuando los datos de la señal son cero la paridad nunca sea nula. Así se efectúa la codificación de la señal, abandonando el codificador CIRC 32 símbolos de ocho bits.

Cuando se reproduce el disco, el sistema de corrección de errores debe comenzar con el desentrelazado de los datos, antes de poder comenzar propiamente los procesos de detección y corrección. Ahora, cuando el decodificador Reed-Solomon recibe un bloque de datos (com-

puesto por los símbolos de los datos originales más la paridad) utiliza los símbolos recibidos para recalcular la paridad. Si los símbolos de paridad calculados coinciden con los recibidos, se asume como correctos los datos del bloque. Si las paridades no coinciden, los síndromes calculados se emplean para la localización del error. Las palabras incorrectas son marcadas como recuperables (las que pueden ser corregidas), irrecuperables (sin posibilidad de corrección) o posiblemente corregibles. Posteriormente se analizarán las marcas para determinar cuándo los errores pueden ser corregidos o cuándo las palabras pasan directamente al interpolador.

El proceso completo del decodificador CIRC aparece en la Figura 5.21. Una trama compuesta por treinta y dos símbolos de ocho bits constituye la entrada al decodificador CIRC, de los que veinticuatro corresponden a símbolos de audio y ocho son símbolos de paridad. Los símbolos pares pasan a través de un retardo de un símbolo, y de esta forma, los símbolos pares de una trama son desentrelazados con los impares de la trama siguiente. Así, cuando los símbolos de audio recuperen su orden original, los errores del disco quedarán dispersados, lo que favorece la corrección mediante C1, especialmente sobre pequeños errores en símbolos adyacentes.

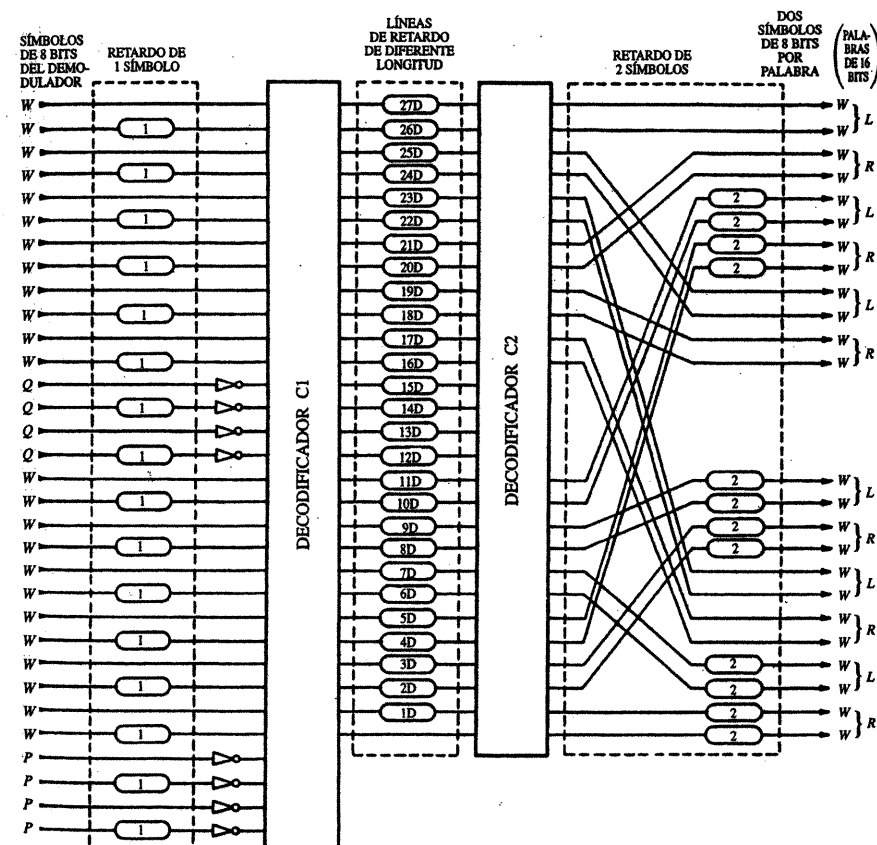


Figura 5.21. El algoritmo decodificador del CIRC.

Realizado este desentrelazado se invierten los bits de paridad, y mediante los cuatro símbolos de paridad P el decodificador $C1$ corrige los errores aleatorios y detecta los errores de ráfaga. $C1$ puede corregir un símbolo erróneo en cada trama; si hay más de uno, marca los 28 símbolos con un indicador y los pasa al decodificador $C2$, mientras que si son correctos no sufren tratamiento alguno. El desentrelazado convolucional entre $C1$ y $C2$ permite a este último corregir errores de ráfaga que recaigan sobre un conjunto grande de datos consecutivos. El conjunto de datos que se aplica a $C2$ contiene símbolos que provienen de $C1$ decodificados en instantes de tiempo muy diferentes (recordar los retardos de longitud variable introducidos en la codificación entre $C2$ y $C1$), con lo que aquellos símbolos marcados como erróneos por $C1$ aparecen dispersados sobre numerosas tramas. Esto facilita la corrección de errores de ráfaga mediante $C2$.

Con la pre-corrección realizada sobre los datos y la ayuda del desentrelazado, $C2$ puede corregir tanto errores de ráfaga como errores de bit que $C1$ era incapaz de corregir. Con los cuatro símbolos de paridad Q , $C2$ puede: detectar y corregir un solo símbolo erróneo, corregir hasta cuatro errores o corregir cualquier símbolo mal corregido por $C1$, llegando incluso a poder corregir errores del propio proceso de codificación. Cuando $C2$ no puede completar la corrección (por ejemplo, cuando fueron marcados como erróneos más de cuatro símbolos por trama), los 24 símbolos de datos son considerados (y marcados) como incorrectos, por lo que pasarán al proceso de interpolación. La etapa final consiste en el desentrelazado y retardo de los símbolos de las palabras pares, para compensar los introducidos en el proceso de codificación.

Hay que hacer notar que en última instancia el éxito en la corrección de errores depende de la implementación del algoritmo. Generalmente el CIRC puede permitir la corrección de hasta 3.874 bits consecutivos, lo que correspondería a un defecto en la pista de una longitud de 2,5 mm. Un buen sistema de detección y corrección puede extender ésta hasta 13.282 bits consecutivos, lo que corresponde a un defecto de 8,7 mm en el disco, aunque un tratamiento adicional puede extender la corrección hasta aproximadamente 15.500 bits.

La integridad de los datos almacenados en un CD puede ser evaluada contando el número de errores. Utilizando una nomenclatura de dos dígitos, el primero especifica el número de símbolos erróneos y el segundo en qué etapa del decodificador ($C1$, $C2$) aparecieron. Mediante un contador de tres dígitos ($E11$, $E21$ y $E31$) evaluamos la salida del decodificador $C1$, de forma que $E11$ especifica el número de errores de un solo símbolo (corregibles) que aparecen en un segundo y en la salida de $C1$. El contador $E21$ indica la frecuencia en la aparición de errores de dos símbolos (corregibles) en la salida de $C1$; y $E31$ la frecuencia de aparición de errores de tres símbolos (sin posibilidad de corrección en $C1$) en la salida de la etapa $C1$.

El BLER («Block Error Rate») mide el número de tramas de datos por segundo en las que al menos aparece un dato erróneo en la entrada de $C1$ ($E11 + E21 + E31$), siendo por tanto una medida de los errores, corregibles o no, que aparecen en la entrada del decodificador. El BLER expresa el número de errores por segundo. En un formato como el de disco compacto, las especificaciones establecen BLER máximo de 220 (bloques erróneos por segundo), promediados en 10 segundos, aunque en los discos correctamente fabricados es inferior a 50, llegando incluso a ser un valor habitual el de 10. Un BLER más elevado indica una pobre geometría de los pits, la cual afecta seriamente la habilidad de lectura del fonocaptor óptico y provoca una elevación considerable de los errores de bit. La velocidad de trama en un CD es de 7.350 bloques por segundo, y por tanto un BLER de 220 indica errores en el 3 por 100 de las tramas de datos. Esta magnitud constituye un límite aceptable en lo referente a la aparición de errores, ya que frecuencias (BLER) superiores corresponden a defectos audibles. El BLER no aporta información sobre los defectos individuales de longitudes entre 100 y 300 μm , debido a que el BLER sólo responde a los defectos del tamaño de pit. A menudo se expresa el BLER como número de errores en un segundo, o como el valor medio en un fragmento de 10 segundos, o incluso dando el valor máximo obtenido en los 10 segundos de duración de la medida.

De forma análoga, en el decodificador $C2$ aparecen tres contadores de error: $E12$, $E22$ y $E32$. El contador $E12$ muestra la ocurrencia de errores de un solo símbolo (corregibles por

tanto) en la salida de $C2$. Un valor de $E12$ elevado no suele ser problemático, puesto que un error $E31$ puede provocar hasta 30 errores $E12$ debido al entrelazado. El contador $E22$ indica la frecuencia en la aparición de dos símbolos erróneos (corregibles) en el decodificador $C2$. Los errores $E22$ son los de corrección más difícil, puesto que el contador $E22$ indica que el sistema está muy cerca de sufrir un error no corregible; y como ejemplo, un CD-ROM con 15 errores $E22$ podría ser inaceptable aunque los errores pudieran ser corregidos. Un valor mayor de $E22$ indica la localización de una zona deteriorada en el disco, bien sea debido a un defecto de fabricación o a un daño producido por la manipulación del mismo. El contador $E32$ indica errores en $C2$ y en tres símbolos (no corregibles); o en general, una zona ilegible del disco, por lo que idealmente no debería aparecer ninguno en el disco. Los errores $E32$ son, a veces, denominados errores de interpolación, ya que cuando aparecen implican la utilización de esta técnica para la reconstrucción de la señal. Si un disco no tiene errores $E32$, la salida digital de los datos es perfecta.

Las señales $E21$ y $E22$ pueden combinarse para darnos otra medida de errores, que es el contador de errores de ráfaga CER. Este contador nos da la medida del número de grupos, formados por un mínimo de X bloques consecutivos, en los que aparece un error en la salida de $C2$. En general, este error indica un defecto físico en la superficie del disco (u otro soporte), tal como un arañazo, y que afecta a series de bloques de datos. Por ejemplo, si el umbral está situado en 7 bloques consecutivos erróneos, y en la reproducción del disco aparecen 14 bloques erróneos seguidos, el contador de CER indicaría el valor de dos. Esta forma de medir los errores se suele aplicar a todo el disco. Desde el punto de vista del control de calidad de un disco, una buena especificación no debería incluir un CER diferente de cero, ya que significaría que no han aparecido errores de ráfaga tan importantes como para superar los siete bloques erróneos consecutivos. En la práctica, un disco recién fabricado debería tener un $\text{BLER} = 5$, $E11 = 5$, $E22 = 0$, $E31 = 0$. Por último, $E32$ debería ser cero, ya que errores sin posibilidad de corrección no deberían aparecer en un disco nuevo.

Para aplicaciones tan variadas como las nuestras existen muchos tipos de sistemas de corrección de errores. Los diseñadores deben tener en cuenta la posibilidad de corregir los errores aleatorios (de bit) y los de ráfaga, el exceso de bits provocado por la redundancia, la probabilidad de aparición de una detección incorrecta, la longitud máxima de errores a corregir u ocultar frente a un error de ráfaga y, por último, el coste económico del codificador y decodificador. En audio digital, los errores que no se pueden corregir es fácil ocultarlos (disimularlos); sin embargo, un fallo en la detección no puede ser disimulado, pudiendo dar como resultado un chasquido en la salida del sistema. Los diseñadores de sistemas de corrección de errores deben asegurarse de que esto no va a suceder ni en el caso más desfavorable, dentro de unas condiciones normales. Además, se deberán tener en cuenta otros criterios específicos de la aplicación particular. Por ejemplo, las cintas de magnetófonos digitales de bobina abierta son muy sensibles a la aparición de errores de ráfaga provocados por las huellas dactilares, mientras que los discos compactos son relativamente tolerantes con estos defectos, ya que la capa de sustrato transparente hace que la interferencia provocada por ellas quede desenfocada, no produciendo por tanto excesivos errores.

Debido a la variada naturaleza de los errores, algunos predecibles y otros no tanto, los sistemas de corrección de errores deben utilizar el concurso de varias técnicas para proteger la señal de ellos. La introducción de retardos para alejar entre sí datos consecutivos, el entrelazado, la adición de paridad y la intercalación cruzada constituyen el conjunto de estrategias empleadas para la corrección de los errores habituales en los sistemas de almacenamiento de la señal de audio.

OCULTACIÓN DE ERRORES

Como ya se ha comentado, un sistema de detección y corrección de errores perfecto es aquel en el que los errores son completamente reemplazados mediante redundancia o recalculando los datos, de forma que restablezca los mismos con total precisión. Sin embargo, un esquema de

estas características es impracticable, debido al exceso de redundancia a codificar y al coste económico de los propios codificadores y decodificadores. En la práctica, un buen sistema de corrección de errores tiene que guardar un equilibrio entre las limitaciones apuntadas y la probabilidad de aparición de errores no corregibles, no perdiendo de vista que éstos alcanzarán la salida de la etapa de corrección de errores sin haber podido ser corregidos. Sin embargo, el tratamiento posterior al que vamos a someter los datos está encaminado a compensar e intentar ocultar todos estos errores con el objetivo de convertirlos en inaudibles. Para llevar a cabo esta tarea se utilizan dos técnicas: la interpolación y el silenciamiento⁷ o anulación de la salida («muting»).

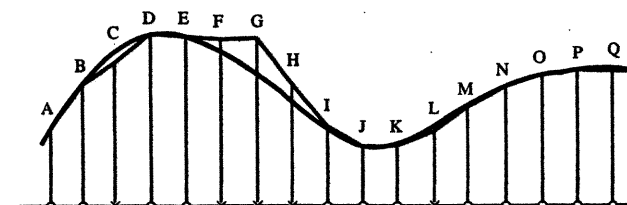
En la salida de nuestros algoritmos de corrección hay dos tipos de errores de difícil (o imposible) corrección: los que el sistema de corrección detectó pero fue incapaz de corregir, y aquellos que simplemente no fueron detectados. Al primer tipo de errores son a los que podemos aplicar las técnicas de ocultación de errores, ya que sobre los no detectados es verdaderamente difícil actuar, y suelen producir un chasquido audible en la salida de la cadena. Los errores detectados pero no corregidos se suelen dar como consecuencia de la aparición simultánea de errores de bit y de ráfaga con una magnitud tal que desbordan los sistemas de corrección. En este caso, el objetivo del sistema de corrección es que se reduzca al máximo los no detectados y dejar que los métodos de ocultación de errores resuelvan el caso de los detectados y no corregidos.

Interpolación

Tras el desentrelazado de los datos, la mayor parte de los errores, incluso los de ráfaga, han sido dispersados de tal forma que aparecen intercalados entre los datos correctos. En estos casos, parece razonable utilizar técnicas en las cuales se utilicen los datos válidos que rodean a una palabra errónea o perdida, para reemplazarla o recalcularla. Esta técnica trabaja muy bien cuando los errores aparecen lo suficientemente dispersos y hay un grado de correlación alto entre los propios datos de la señal. Afortunadamente, los datos digitales asociados a la señal musical cumplen este requisito, y puede aplicarse bien la interpolación sin que afecte de forma negativa la audibilidad de la señal. Aunque no es intuitivo, estudios realizados demuestran que, dentro de ciertos límites, la duración temporal de un error no afecta excesivamente la percepción del mismo.

El caso más simple de interpolación consiste en la repetición del valor de la última muestra correcta durante el intervalo de duración del error. Esto se denomina interpolación de orden cero o interpolación con el valor anterior. La interpolación de primer orden o interpolación lineal consiste en asociar el valor perdido al resultado de la media aritmética realizada sobre los valores de las muestras anterior y posterior a la errónea. En muchos sistemas digitales de audio se utiliza una combinación de interpoladores de orden cero y primer orden. Si a pesar del desentrelazado aparecen errores sobre muestras consecutivas, se repite el valor de la última muestra correcta hasta la penúltima errónea, calculando la última como la media entre el valor anterior repetido y el de la muestra siguiente, que ya es correcto. Si los errores son aleatorios, y las muestras erróneas aparecen rodeadas de valores reales, se suele utilizar la interpolación lineal. En la Figura 5.22 aparece representado un ejemplo de esta estrategia de interpolación. En ocasiones se utilizan otros sistemas de interpolación de orden más elevado, de forma que una interpolación de orden n utiliza un polinomio de ese grado en el cálculo de los datos a reemplazar. En la práctica, los sistemas de ocultación de errores suelen utilizar interpoladores de tercer o quinto orden, aunque, lógicamente, los cálculos se deben realizar con la suficiente rapidez

⁷ Es tan habitual el término «mute» o «muting» dentro de nuestro argot, que le mantendremos en el texto sin utilizar su traducción.



(Líneas gruesas para los datos analógicos de entrada. x en los datos erróneos)

Interpolación manteniendo el valor previo

$$G = F (= E)$$

Interpolación manteniendo el valor medio

$$C = \frac{1}{2} (B + D)$$

$$H = \frac{1}{2} (G + I) = \frac{1}{2} (E + I)$$

$$L = \frac{1}{2} (K + M)$$

Figura 5.22. La interpolación se utiliza para la ocultación de errores. Por ejemplo, se puede mantener el valor de una muestra previa, terminando con el cálculo del valor medio.

como para mantener constante la velocidad de salida de los datos. En la Figura 5.23 se representan las características del ruido introducido sobre un tono, dada una determinada longitud de error y en función del orden del interpolador empleado (según Toshi Doi).

Silenciamiento de la salida («muting»)

Es el proceso de reemplazar por un dato de valor cero aquellas muestras irrecuperables o que no fuimos capaces de corregir, siendo este silenciamiento preferible a la señal impredecible que

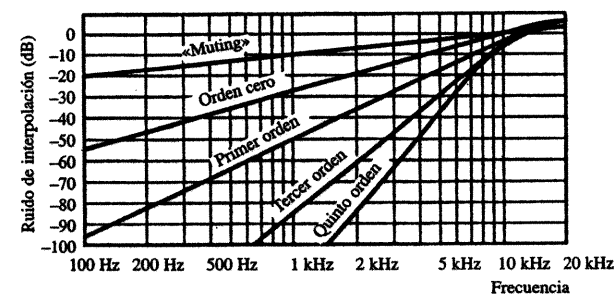


Figura 5.23. Características del ruido para diferentes métodos de interpolación (para ráfagas de 100 bits, tonos puros y errores no corregibles).

puede resultar de la decodificación de un dato incorrecto. El «muting» debe ser utilizado en aquellos casos donde el silenciamiento de la señal es preferible a reproducir un chasquido en la salida de la cadena. El incremento instantáneo de la distorsión que se genera con un «mute» momentáneo es casi imperceptible, mientras que los chasquidos («click») son siempre mucho más perceptibles. Además, en el caso de una pérdida considerable de datos o un mal funcionamiento del reproductor, es preferible tener ausencia de señal («mute») que tener una cadena de errores impredecibles. Con el fin de minimizar la percepción de un «mute», los algoritmos que lo generan atenúan gradualmente la señal de salida antes de producir el total silenciamiento, restableciendo también el nivel de forma gradual una vez que la condición que provocó el «mute» ha desaparecido. La forma de llevar esto a cabo es multiplicando las sucesivas muestras por coeficientes cada vez más pequeños, realizándose este proceso durante los milisegundos previos a la aparición del error. Para tener un orden de magnitud de la audibilidad que puede tener el «muting», hay que tener en cuenta que no superando el margen de 1 a 4 ms no suele ser percibido por el oído humano. Asociado al «mute» se suelen emplear otro tipo de estrategias para hacerlo menos notorio, como enmudecer sólo un canal en vez de los dos de un estéreo, etc.

DUPLICACIÓN

Una de las ventajas que posee el audio digital es la posibilidad de efectuar la copia de una grabación sin la inevitable degradación que aparece con la duplicación realizada en analógico. Aunque la duplicación en digital se realiza sin pérdidas, el éxito de la misma es consecuencia del éxito en la corrección de errores. Aunque los sistemas de corrección de errores aportan unos métodos destinados a corregir los datos defectuosos, los procesos de ocultación de errores no trabajan en este sentido; de forma que, aunque en ocasiones extraordinarias, estas técnicas pueden provocar errores considerablemente audibles cuando se efectúa una duplicación de la señal. Si se obtienen diferentes generaciones digitales a partir de esta copia, cada una generaría nuevos errores no contenidos en el original, con lo que es posible que tras varias copias la señal no sólo fuera totalmente irrecuperable sino inservible para su audición. La única solución consiste en la corrección de los datos en su origen, así que determinadas precauciones como utilizar soportes limpios, cuidar su manipulación, limpiar la mecánica de los magnetófonos y reproductores y utilizar interconexiones con bajo «jitter» constituyen principios básicos a mantener sobre todo cuando se prevé la generación de copias múltiples. En la práctica, y afortunadamente, los datos de audio digital pueden ser copiados con unas elevadas garantías de éxito.

Capítulo 6

ALMACENAMIENTO EN CINTA MAGNÉTICA

La grabación magnética ha sido el principal sistema de almacenamiento de la señal analógica de audio durante los últimos 50 años. Si a la facilidad y capacidad de leer, escribir y borrar una cinta magnética le añadimos que durante los últimos 50 años la densidad de almacenamiento se ha incrementado más de 750.000 veces, es lógico que se haya convertido en un medio de almacenamiento de características únicas. Con la proliferación de los ordenadores y el desarrollo paralelo de los soportes de almacenamiento de datos, era lógico que la señal digital de audio fuera almacenada magnéticamente, utilizando técnicas convergentes tanto en la industria de audio como en los sistemas de almacenamiento de datos. Hoy en día, la cinta magnética es un medio de almacenamiento indispensable en aplicaciones de audio. A diferencia de la cinta en los magnetófonos analógicos, con el almacenamiento digital (junto con el magneto-óptico) el soporte de almacenamiento no es un factor que condicione las características de la señal. Sin embargo, la enorme cantidad de datos que contiene un fragmento de audio pone a prueba la capacidad de los sofisticados sistemas de almacenamiento, ya sea de tipo estacionario o de cabezas rotativas. De estos sistemas, el DAT será visto en detalle en el Capítulo 7.

ANCHO DE BANDA DE GRABACIÓN

El ancho de banda de un dispositivo mide el rango de frecuencias que es capaz de acomodar con unas pérdidas de amplitud aceptables. En el caso de un magnetófono analógico, el ancho de banda de 20 kHz (0 a 20 kHz) es adecuado debido a que las frecuencias de audio se graban directamente en la cinta, proceso que se realiza ayudados de una señal de alta frecuencia («bias»). Sin embargo, el registro digital requiere un ancho de banda mucho más elevado, puesto que la frecuencia más alta a grabar es muy superior a los 20 kHz de la señal. Por ejemplo, si para una señal de 20 kHz se necesita una frecuencia de muestreo de 44,1 kHz, y cuantificamos las muestras con 16 bits, la señal a grabar necesita $44,1 \text{ kHz} \times 16 \text{ bits}$, cerca de 705 kbps (kilobits por segundo). Si a esto añadimos los datos de corrección de errores, sincronización y modulación de canal (que aunque incrementa el volumen total de datos, mejora la densidad de almacenamiento), la señal a grabar puede elevarse a 1 Mbps (megabit por segundo) sólo para un canal de audio. Con esto, el ancho de banda que se requeriría es superior en más de 50 veces al de un magnetófono analógico. Así, aunque tuviéramos una densidad lineal muy elevada, necesitaríamos una cinta largísima, incluso para un tiempo de grabación relativamente modes-