Algoritmos Paralelos en Procesamiento de la Señal

Profesorado

Dr. Antonio M. Vidal

Dr. Víctor M. García Mollá

Febrero 2016

Objetivos

- Mostrar como surgen problemas matriciales en el campo del procesado de señal que pueden resolverse técnicas propias de la computación paralela.
- Mostrar ejemplos concretos donde hemos aplicado estas ideas
- Analizar la paralelización de dichos algoritmos y las herramientas secuenciales y paralelas existentes actualmente para su uso

Programa

- **1. Problemas computacionales en procesado digital de señales. Computadores específicos para procesado de señales**
- **2. Problemas de mínimos cuadrados: Resolución secuencial y paralela**
- **3. Computación de Trasformadas Discretas: La FFT y sus aplicaciones**
- **4. Computación de Trasformadas Discretas: La Transformada Wavelet y sus aplicaciones**
- **5. La Descomposición en Valores Singulares: Resolución secuencial y paralela**

Calendario

- Miércoles, 17 de Febrero de 2016. 16:00 a 20:30. Aula 0S03
- Martes, 23 de Febrero de 2016. 16:00 a 20:30. Aula 0S03
- Miércoles, 24 de Febrero de 2016. 16:00 a 20:30. Aula 0S03
- Martes, 1 de Marzo de 2016. 16:00 a 20:30. Aula 0S03
- Miércoles, 2 de Marzo de 2016. 16:00 a 20:30. Aula 0S03

Evaluación

- Seguimiento del curso, participación en actividades, desarrollo de algoritmos,…: 70%
- Trabajo sobre relacionado con la temática del seminario: 30%

- Fechas de evaluación:
  - Solicitud de un trabajo a los profesores de la asignatura antes del 2 de marzo de 2015
  - Presentación del trabajo antes del 15 de Abril

Bibliografía

- Alan V. Oppenheim, Ronald W. Schafer with John R. Buck. “Discrete-time signal processing”, - 2nd ed. - Upper Saddle River : Prentice-Hall International, cop. 1999. ISBN 0137549202
- Maurice Bellanger. “Digital processing of signals : Theory and practice”, John Wiley, cop. 1984. ISBN 0471903183
- Maurice Bellanger. “Tratamiento numerico de la señal : teoría y practica”. Barcelona : Masson, 1991.
- T.M. Apostol “Análisis Matemático”. Addison-Wesley. 1960
- Ake Björck. “Numerical methods for least squares problems”. Philadelphia. SIAM, 1996.
- V.Klema & A.J.Laub, “The Singular Value Decomposition: its Computation and some Aplications”. IEE Transactions on Automatic Control. Vol. AC-25, No. 2, April 1980
- G.H.Golub and C.F. Van Loan. "Matrix Computations“. The Johns Hopkins Univ. Press, 3rd Ed. Baltimore, 1996.

Bibliografía

- Edited by Ed. F. Deprettere. “SVD and signal processing : Algorithms, applications and architectures”. Amsterdam. North-Holland, 1988.
- Edited by Richard J. Vaccaro. “SVD and signal processing, II : algorithms, analysis, and applications”. Amsterdam. Elsevier, 1991

- Charles Van Loan.“Computational frameworks for the fast Fourier transform”. Philadephia : SIAM, 1992

- Edward Aboufadel and Steven Schlicker. “Discovering wavelets”. New York. John Wiley & Sons, 1999
- Gerald Kaiser. “A friendly guide to wavelets”. Boston. Birkhäuser (1994)
- Robi Polikar. “The Wavelet Tutorial”

http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

Indice

- Introducción
- Computadores para procesado digital de señales
- Representación de la señal
  - Problemas de mínimos cuadrados
  - Cambio de dominio: FFT, Wavelets,…
- Sistemas y Decodificación
  - Filtrado: Convolución
  - Problemas de rango: SVD
  - Sistemas MIMO: Problemas de mínimos cuadrados discretos

Introducción: ¿Qué es una Señal?

José F.M. Moura, president of the Signal Processing Society

IEEE Signal Processing magazine. Volume: 26. Issue: 6. Pp 6-6 Date: November 2009

- What do we mean by *signal*?" and "How do we define *processing*?" Ages ago, *signal* referred to **some physical manifestation of information that changed with time** and/or space. By signal we may still be referring to a physical manifestation but we might also be dealing with other symbolic or abstract information formats like a sequence of millions of the four symbols of the genetic code (the DNA bases A, C, G, T) arranged into genes and noncoding sections. Or, we may be referring to some other abstract attributes of sequenced information: cold, hot, high, low. Examples of signals include audio, video, speech, language, image, multimedia, sensor, communication, geophysical, sonar, radar, biological,…

Introducción: ¿Qué es una Señal?

José F.M. Moura, president of the Signal Processing Society

IEEE Signal Processing magazine. Volume: 26. Issue: 6. Pp 6-6 Date: November 2009

- As for *processing*, it comprises operations of representing, filtering, coding, transmitting, estimating, detecting, inferring, discovering, recognizing, synthesizing, recording, or reproducing signals by digital or analog devices, techniques, or algorithms, in the form of software, hardware, or firmware

Introducción: ¿Qué es una Señal?

José F.M. Moura, president of the Signal Processing Society

IEEE Signal Processing magazine. Volume: 26. Issue: 6. Pp 6-6 Date: November 2009

So, putting it together, can we say that *signal processing* is an enabling technology that encompasses the fundamental theory, applications, algorithms, and implementations of processing or transferring information contained in many different physical, symbolic, or abstract formats broadly designated as signals and uses mathematical, statistical, computational, heuristic, and/or linguistic representations, formalisms, and techniques for representation, modeling, analysis, synthesis, discovery, recovery, sensing, ac quisition, extraction, learning, security, or forensics?

Introducción: ¿Qué es una Señal?

- Matemáticamente:

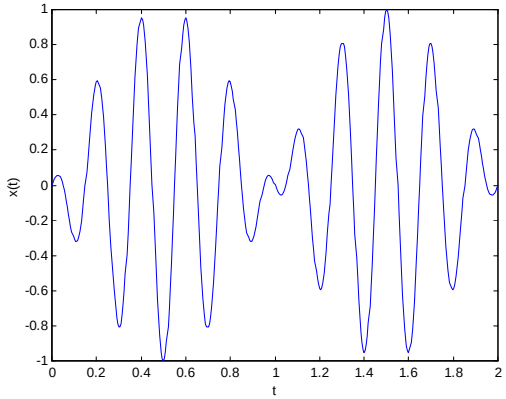
$$F : A \subset \Re \rightarrow \Re \; or \; C$$

- Físicamente:

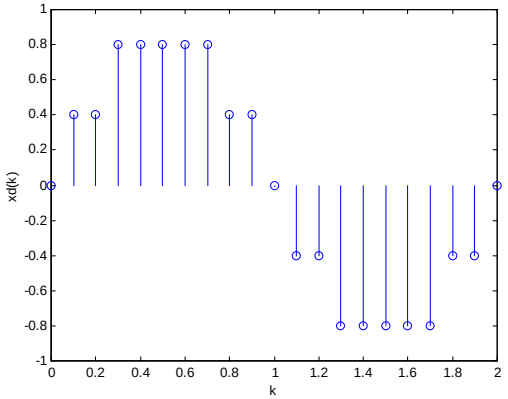
Magnitud que varía con el tiempo

- Señales analógicas: A es un subconjunto continuo
- Señales digitales: A es un subconjunto de Z

Señal análogica $x(t)$



Señal digital $x_d(k)$



Procesado de Señal: Área de la matemática aplicada que trata con las operaciones sobre señales o con el análisis de señales, en tiempo continuo o en tiempo discreto para realizar operaciones útiles con dichas señales.

Computadores específicos para Procesado de Señal

- Problemas de simulación:
  - Relacionados con estudio de campos electromagnéticos. Ecuaciones integrales, diferenciales:
    - **Computadores de propósito general con potencia dependiente del tamaño y complejidad de los problemas**

- Problemas de tiempo real:
  - El tiempo de respuesta es crítico.
    - **DSPs**
    - **FPGAs**
    - **Computadores de propósito general (Multicores)**
    - **Aceleradores / Tarjetas gráficas (GPUs)**

Gonzalez, J.A. Belloch, F.J. Martinez-Zaldivar, P. Alonso, V. Garcia, E.S. Quintana-Orti, A Remon, and A.M.Vidal. The impact of the multi-core revolution on signal processing. Waves ITEAM, (2):64–75, November 2010.

http://www.iteam.upv.es/revista/2010/8_ITEAM_2010.pdf

Procesado en tiempo real

Todos los algoritmos que se tienen que aplicar a S_{IN} se han de computar en un tiempo $\leq$ Ts=1/fs

Ej. MODEM 20 Mbps: todos los algoritmos que intervienen en la cadena TX o RX se han de computar en menos de 50 ns/bit

PROCESADORES DIGITALES DE SEÑAL (DSP)

Un Procesador Digital de Señales (DSP, sigla en inglés de *Digital Signal Processor*) es un tipo de microprocesador, diseñado para procesar señales en tiempo real.

•Señal analógicas: continua en el tiempo: cambia suavemente desde un estado a otro.
•Computadores digitales: manejan la información discontinuamente, como una serie de números binarios

•Primera etapa en la mayoría de los sistemas basados en DSP's:
transformar las señales analógicas en digitales, mediante Conversores Analógico – Digitales (ADC).
•Procesamiento:
Los datos digitales se entregan al DSP el cual está ahora en condiciones de procesarlos.
•Etapa final:
El DSP deberá devolver los datos ya procesados que se transforman al formato análogico (Conversores DAC)

La capacidad del procesador es una función de su ancho de datos (el número de bits manipulados) y el tipo de aritmética que posee (coma fija o flotante).

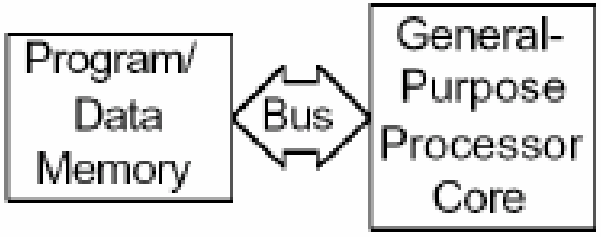
•DSP's de 16 bits son ideales para sistemas de voz tales como teléfonos ya que ellos trabajan con un estrecho rango de frecuencias de audio.

•Equipos estereos de alta fidelidad requieren ADCs de 16 bits y un procesador de 24 bits de punto fijo. Los 16 bits del conversor permiten capturar todo el rango de la señal de audio y los 24 bits del procesador permiten operar cómodamente los grandes valores resultantes de la operación con los datos.

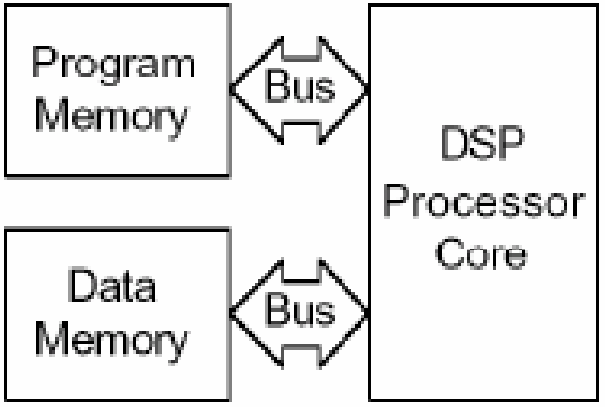
•Procesamiento de imágenes,gráficos 3–D y simulaciones científicas necesitan un rango dinámico mucho mayor y por lo tanto requieren procesadores de coma flotante de 32 bits y ADC's de 24 bits.

•Un DSP estándar tiene muchos rasgos de una arquitectura tipo RISC, pero son procesadores de propósitos específicos cuya arquitectura está especialmente diseñada para operar en ambientes de alta necesidad de cálculo.
•Un DSP estándar ejecuta varias operaciones en paralelo mientras que un RISC usa unidades funcionales altamente eficientes que pueden iniciar y completar una instrucción simple en uno o dos ciclo de reloj.
•Varios buses y memorias incluidas en el chip se utilizan de forma que lecturas y escrituras a diferentes unidades de memoria se pueden hacer a la vez.
•Suelen usar dos memorias caché: Una para datos, y otra para instrucciones (**arquitectura Harvard**). Alto grado de concurrencia (lecturas y escrituras simultáneas).
•Los DSP's actuales usan varios buses y unidades de ejecución para alcanzar niveles incluso más altos de concurrencia.

Microprocesador



DSP



Características de DSP's típicos:

1. **Una unidad funcional rápida que puede multiplicar y acumular en un ciclo de instrucción.** Un ciclo de instrucción puede durar generalmente 1 ó 2 ciclos de reloj. Disponibles en DSP's de punto fijo y flotante.

2. **Varias unidades funcionales que realizan operaciones en paralelo, incluyendo accesos a memoria y cálculo de direcciones**. La unidades poseen típicamente una unidad principal (ALU) junto con dos o más unidades de generación de direcciones. Estas unidades funcionales poseen su propio conjunto de registros y muchas instrucciones se realizan en un solo ciclo de instrucción.

3. **Varias unidades de memoria on–chip (generalmente 2 ó 3) usadas para almacenar instrucciones, datos o tablas.** Cada unidad de memoria puede ser accedida una vez en cada ciclo de instrucción.

4. **Varios buses para incrementar las tasas de transferencia** hacia y desde memoria y evitar conflictos de direcciones.

5. **Soporte para tipos especiales de direccionamiento**, especialmente modulo y bit– reverse, requerido en el cálculo de la FFT, y para la implementación de buffers circulares.

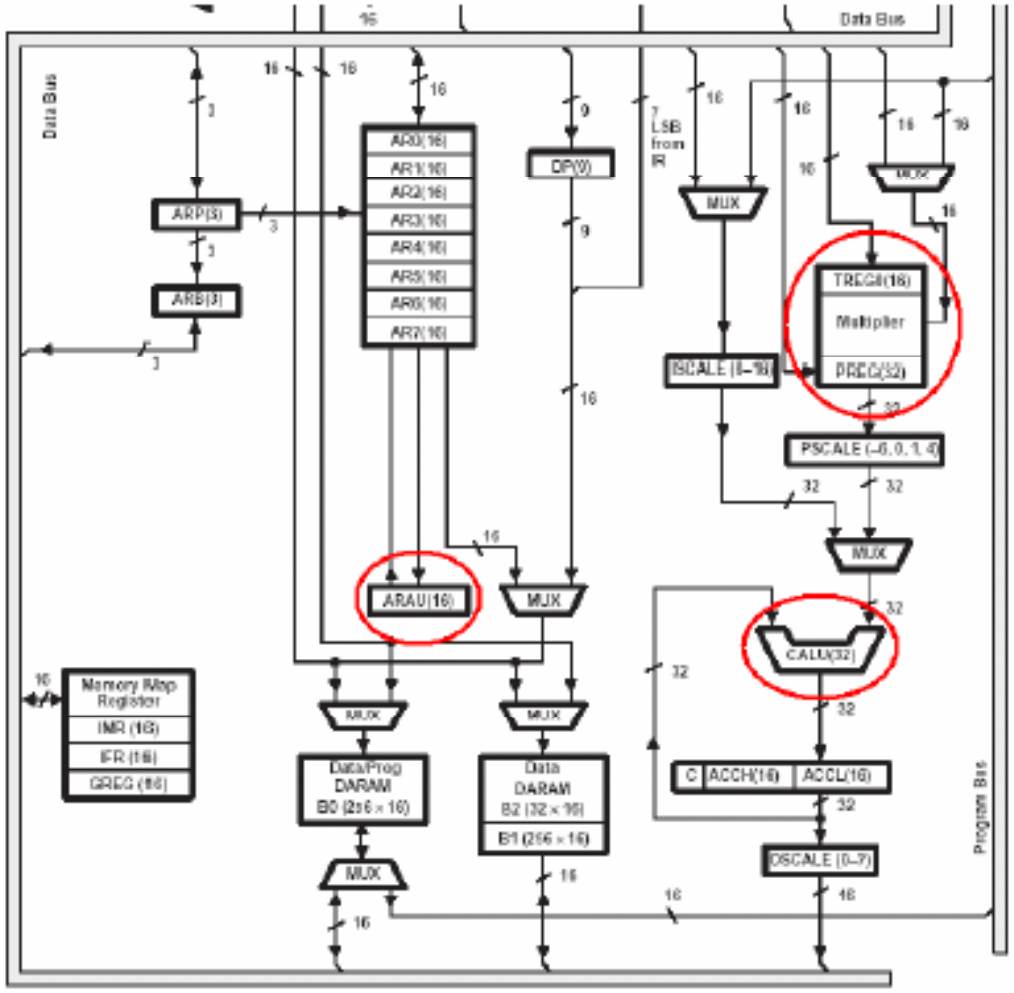


Diagrama de bloques del DSP TMS320F241.

3 unidades de cálculo:
CALU: operaciones aritmético – lógicas
ARAU: cálculos sobre registros auxiliares para direccionamientos indirectos
Multiplicador: ejecución rápida de operaciones tipo a=a+b*c

Fabricantes de DSPs más conocidos

1. Texas Instruments: http://dspvillage.ti.com/docs/dspproducthome.jhtml
Familias TMS320C6000 TMS320C5000 TMS320C2000

2. Motorola: http://www.motorola.com/semiconductors
Familias 56300 56800 56800E MSC8100 (StarCore)

3. Analog Devices: http://www.analog.com/technology/dsp/index.html
Familias Blackfin Familia Sharc TigerSharc ADSP-21xx

Evolución DSP → Multicore
Ejemplo: Familia C6000 de Texas Instruments
- Alto rendimiento,
- Bajo consumo
- Uso de procesadores ARM Cortex

FPGAs
(Field Programmable Gate Array)

Son dispositivos lógicos de propósito general programable por los usuarios, compuesto de bloques lógicos comunicados por conexiones programables.

El tamaño, estructura, número de bloques y la cantidad y conectividad de las conexiones varía en las distintas arquitecturas.

Modelos de programación similares a los Procesadores Sistólicos

Está formado por un circuito integrado que contienen celdas lógicas idénticas (64 hasta 8.000.000) que se puede ver como componentes estándar
Las celdas lógicas se interconectan por medio de una matriz de cables y switches programables

•Estructura : array bidimensional de bloques lógicos rodeados por conexiones configurables.
Una familia contiene idénticos bloques lógicos y conexiones , pero difieren en el tamaño del array

*Tecnología de programación : se programa por la carga de celdas de memoria de configuración, que controlan la lógica e interconexiones.

*Características: volatilidad, no volatilidad, memoria externa, reprogramabilidad, proceso de fabricación estándar y bajo consumo.

Tipos de FPGAs

Fabricantes

Actel, Altera, Atmel, Chip Express, Clear Logic, Cypress, DynaChip, Fast Analog Solutions, Gatefield, HammerCores, Lattice, Lucent Technologies, Motorola, Orbit, QuickLogic, QuickTurn, Vantis, Xilinx, ...

Por la tecnología de la memoria de programación

Volátiles → Basadas en RAM

Su programación se pierde al quitar la alimentación. Requieren una memoria externa no volátil para configurarlas al arrancar (antes o durante el reset)

No volátiles → Basadas en ROM

Reprogramables → Basadas en EPROM (Erasable-Programmable ROM) o flash

Se borran y se pueden voilver a programar (unos 10.000 ciclos)

No reprogramables → Basadas en fusibles

Sólo se pueden programar una vez. No aptas para laboratorios, pero sí para el espacio

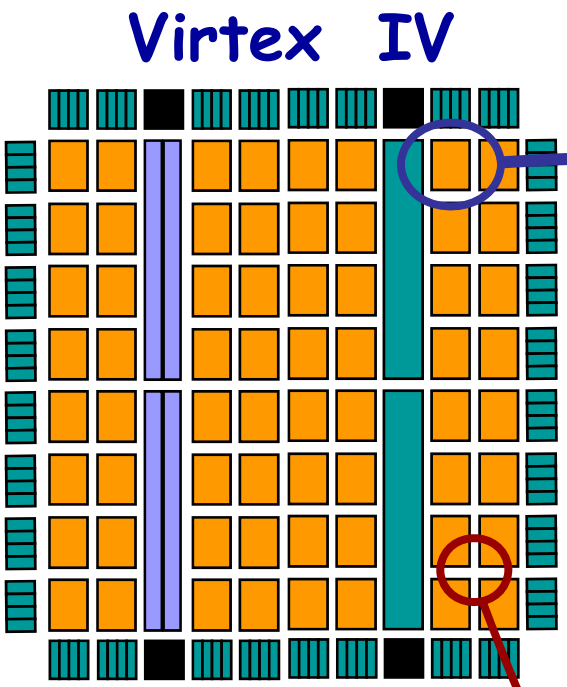
Por su tamaño (y estructura)

PLDs (Programmable Logic Devices)	→ Hasta 2,5 kpuertas
CPLDs (Complex PLDs)	→ Hasta 15 kpuertas
FPGAs	→ Hasta 4 Mpuertas

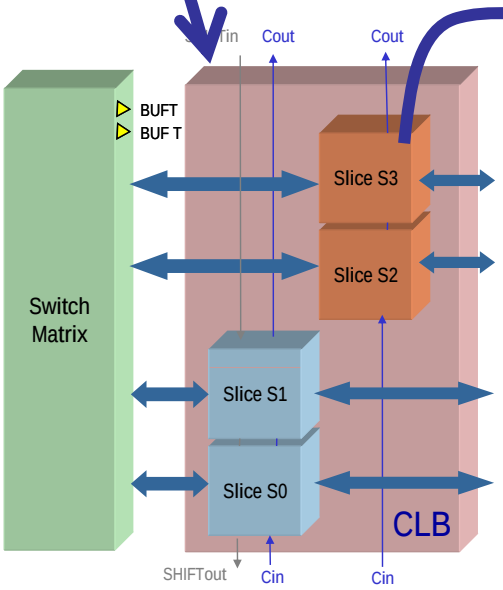
DSP vs. FPGA

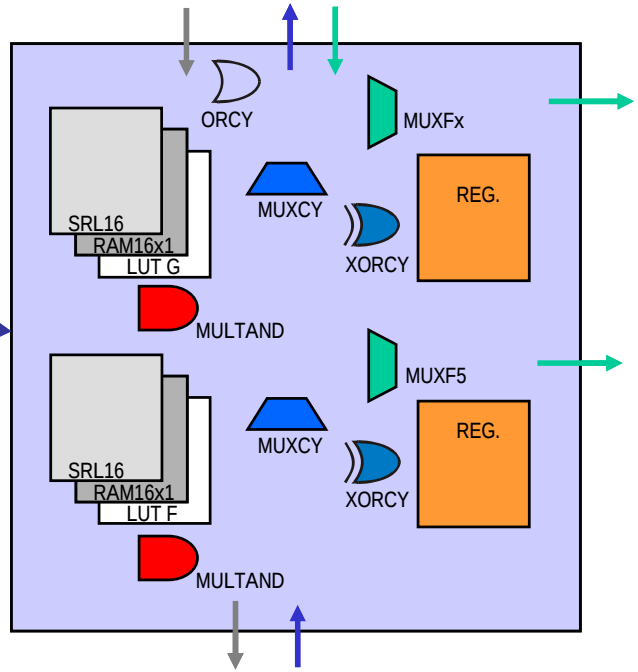
	DSP	FPGA
MMACS	1000x4 TMS320C64	400x512 XC4VSX55
Paralelización	Instrucción	Sí
Desarrollo en C	Sí	Incipiente
Tiempo de desarrollo	Bajo	Alto
Tamaño de palabra	±Fijo	Variado
Tipo de programación	SW	HW(+SW)

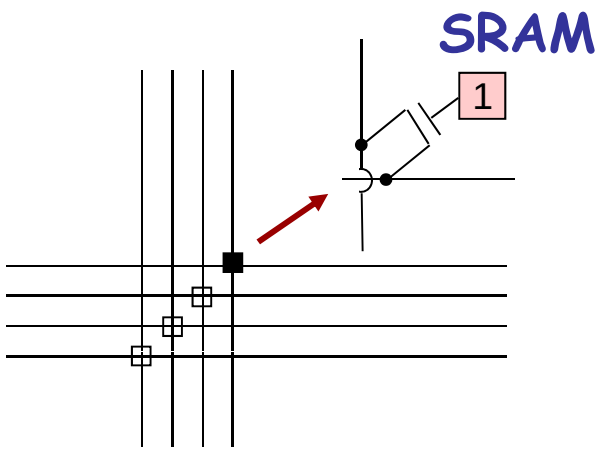
FPGAs



- CLBs con 4 slices
- BSRAM 18 Kbits
- DSP48 (Mult18x18+ acumulador de 48 bits)

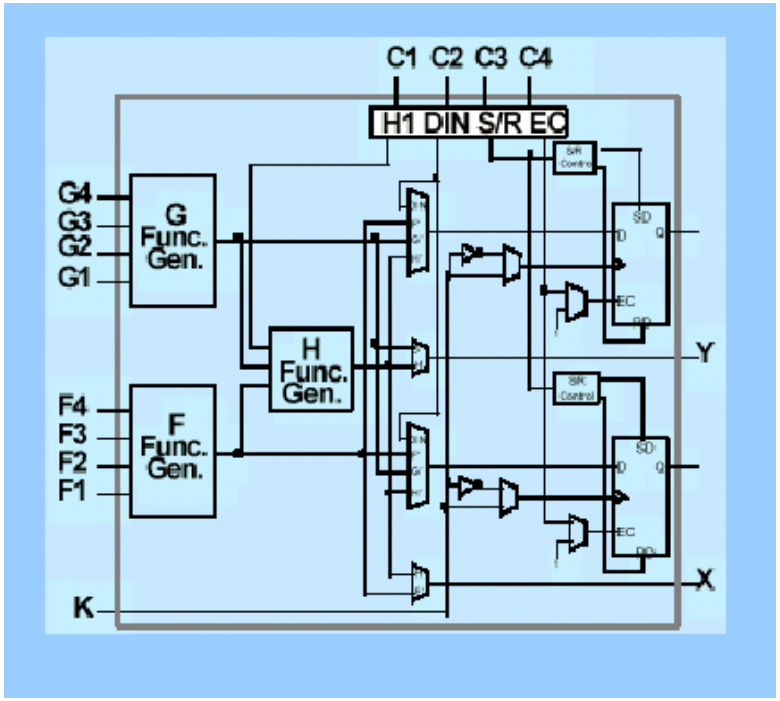






¿Qué hace una celda lógica?

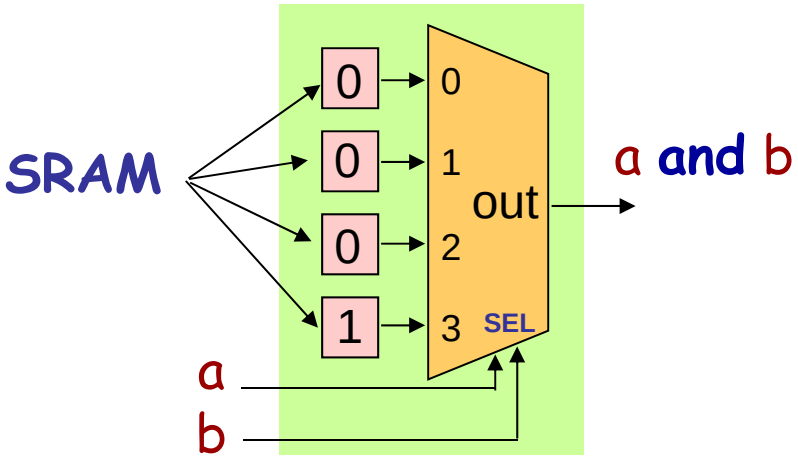
•Su arquitectura varia entre familias de dispositivos
•Cada una combina entradas binarias a una o dos salidas
•En la mayoría de las familias, el usuario puede colocar un registro a la salida de una celda, para implementar lógica secuencial
•La lógica combinacional (LUT) de las celdas se puede implementar físicamente como una pequeña memoria o como un conjunto de multiplexores y puertas lógicas

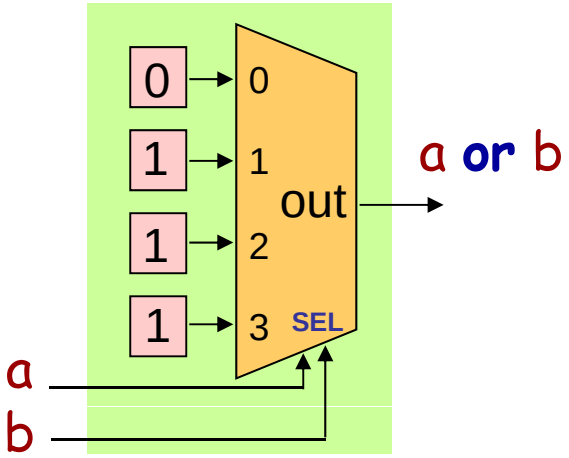


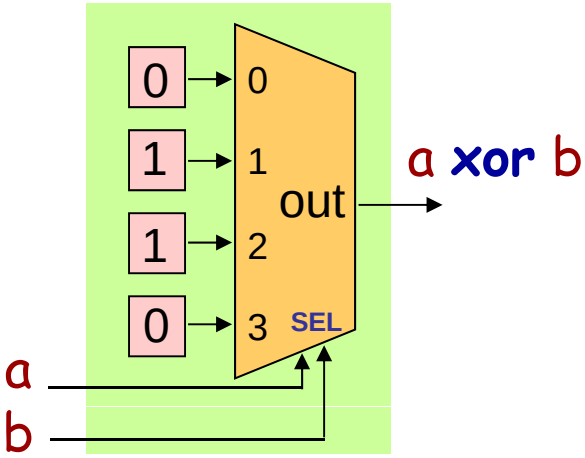
FPGAs

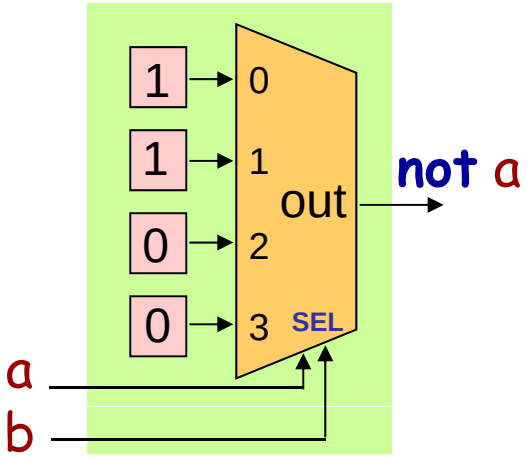
Ej: LUT 2

a b	a and b	a or b	a xor b	not a
00	0	0	0	1
01	0	1	1	1
10	0	1	1	0
11	1	1	0	0







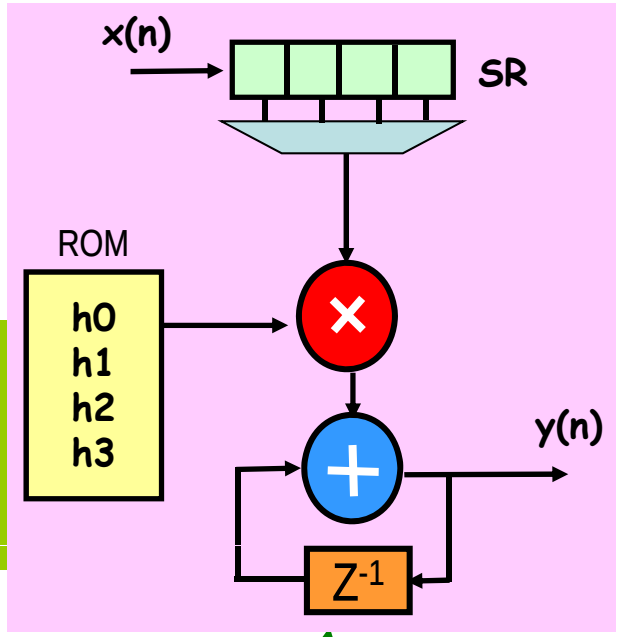


Paralelismo en operaciones: Filtro FIR

$$y(n) = \sum_{k=0}^{M-1} h_k x(n-k)$$

```
z(0) = 0
for k = 1 to M do
  z(k) = z(k-1)+ h(k-1)*x(n-k+1)
end
y(n) =z(M)
```

= FPGA



Ej: M=100

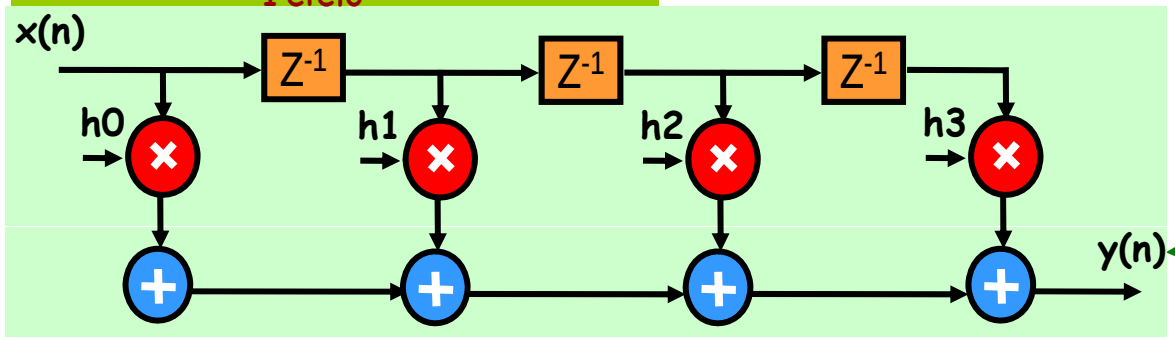
DSP(2 cores)

$$\frac{1\,GHz}{50\ ciclos} = 20\,MSPS$$

FPGA

$$\frac{500\ MHz}{100\ ciclos} = 5\ MSPS$$

FPGA $\frac{500\ MHz}{1\ ciclo} = 500\ MSPS$



Compromiso Área-*throughput*

Flujo de diseño con FPGAs

C<= A+B when S='1'
else A-B;

HDLs

Síntesis

netlist

Lista de componentes específicos de la tecnología y sus conexiones

Mapeado

Agrupación de componentes en slices y CLBs

Emplazamiento y rutado

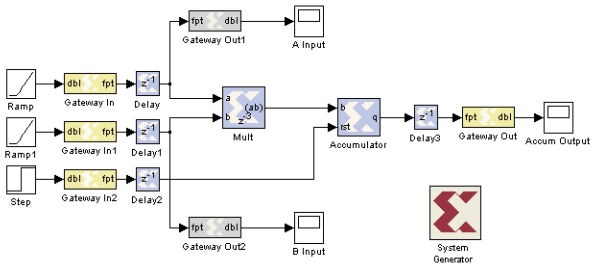
Ubicación en CLBs concretos del dispositivo y realización de conexiones

*.bit

Configuración del dispositivo y test

Entornos gráficos:

- System generator
- CoCentric



Simulación

Diseño con FPGAs

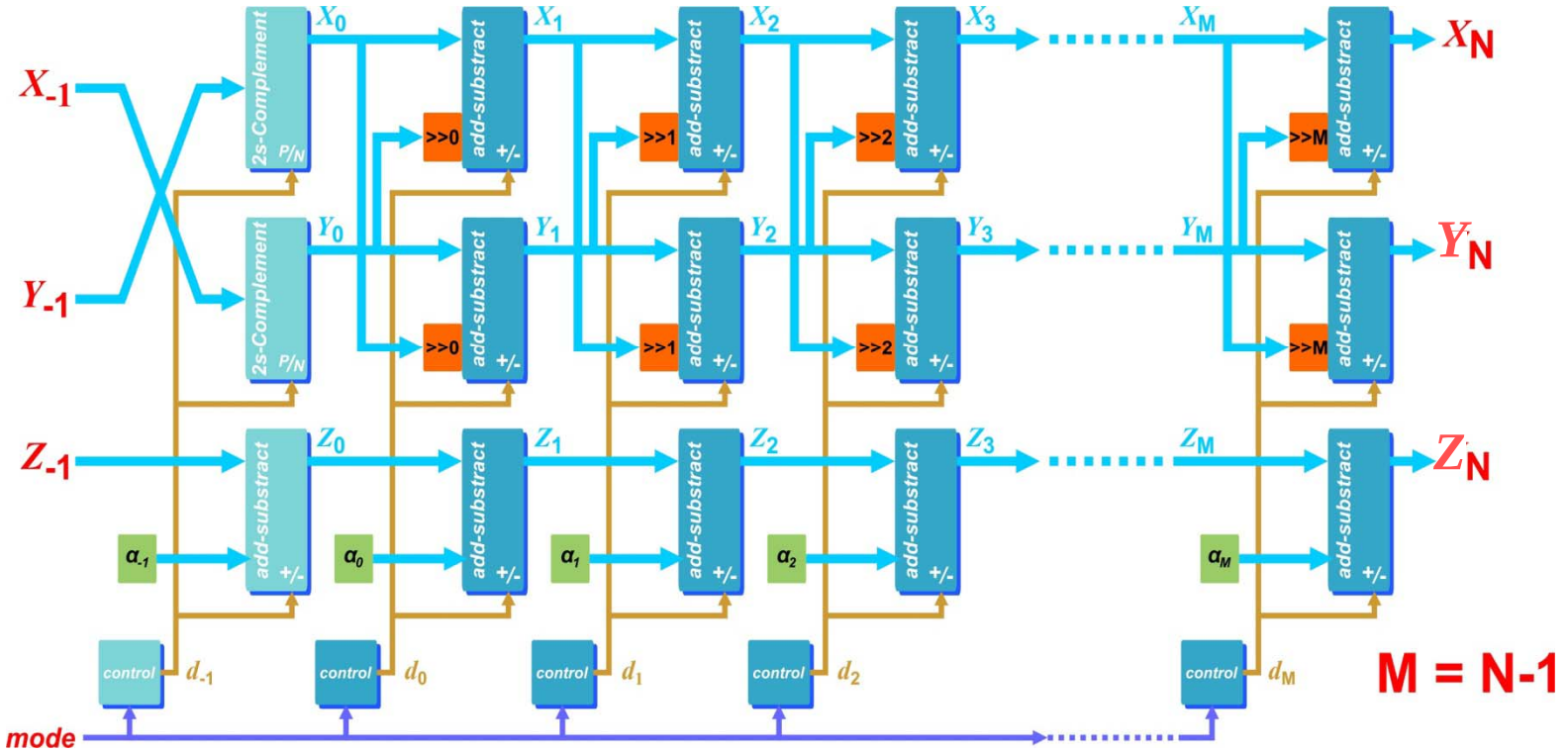
- **Para** facilitar la implementación HW **se trabaja en** punto fijo
  - Suma en coma flotante es costosa
- **El** tamaño de palabra **y la** posición del punto binario **se adaptan al** rango y precisión **de la señal en cada operador**
- **Es necesario un** análisis de precisión finita
  - Simulación en coma flotante -> elección de formatos en cada operador -> ¿modelos pf = cf?

Conclusiones

- Flexibilidad a cambio de coste de diseño
  - Transformaciones arquitecturales (paralelismo, segmentación, retiming, arrays sistólicos) o algorítmicas (técnicas look-ahead, técnicas de reducción de complejidad)
- Limitación de recursos
  - Precisión finita, distribución de recursos
  - Conocimientos de implicaciones HW

Estructura HW

CORDIC: arquitectura paralela



Alternativa: MULTICORES y TARJETAS GRÁFICAS

IEEE Signal Processing magazine. Volume: 26. Issue: 6. November 2009

Gonzalez, J.A. Belloch, F.J. Martinez-Zaldivar, P. Alonso, V. Garcia, E.S. Quintana-Orti, A Remon, and A.M.Vidal. The impact of the multi-core revolution on signal processing. Waves ITEAM, (2):64–75, November 2010.

A.Gonzalez, J.A.Belloch, G.Piñero, J.Llorente, M.Ferrer, S.Roger, C.Roig, F.J.Martínez, M. De Diego, P.Alonso, V,M.García, E.S.Quintana-Ortí, A.Remón, A.M.Vidal. “Applications of Multi-core and GPU Architectures on Signal Processing: Case Studies”. iTeAM Research Journal. Vol. Pp. 86-96

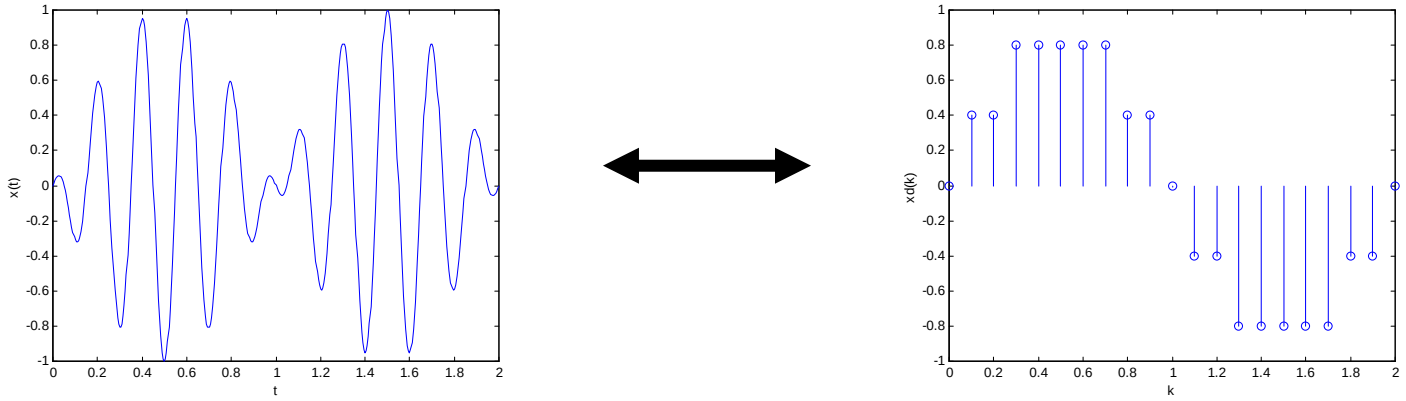
¿Qué es una Señal?

- Procesado de señales digitales

Secuencias y funciones

Señal análogica $x(t)$

Señal digital $x_d(k)$



Conversión de función a secuencia: Muestreo y cuantización
Conversión de secuencia a función: Ajuste por mínimos cuadrados

Representación de la señal

Problemas de mínimos cuadrados planteados:

- Problema lineal de mínimos cuadrados

$$\min_{x\in\Re^n} \left\| Ax-b \right\|_2, \; con \; A\in\Re^{mxn}, b\in\Re^m$$

- Problemas de mínimos cuadrados con restricciones, ponderados, generalizados,…

$$\min_{sujeto\ a\ Bx=d} \left\| Ax-b \right\|_2, \; con \; A\in\Re^{mxn}, b\in\Re^m, B\in\Re^{pxn}, d\in\Re^p, p\le n$$

- Problema incremental de mínimos cuadrados: “Updating”

¿Cómo actualizar el ajuste de una curva cuando se conoce un nuevo punto?

- Problema decremental de mínimos cuadrados: “Downdating”

¿Cómo actualizar el ajuste de una curva cuando se elimina un punto?

Representación de la señal

Problemas de mínimos cuadrados planteados:

Herramientas básica: Transformaciones ortogonales: $Q^T Q = QQ^T = I$

Conservan la dos-norma vectorial

$$\|Qx\|_2 = \sqrt{(Qx)^T (Qx)} = \sqrt{xQ^T Qx} = \sqrt{x^T x} = \|x\|_2$$

Descomposiciones QR y RQ

$$Dada\ A \in \Re^{mxn}\ encontrar\ Q \in \Re^{mxm}, R \in \Re^{mxn}, R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, R_1 \in \Re^{nxn}$$

$$tales\ que\ A = QR$$

$$Dada\ A \in \Re^{mxn}\ encontrar\ Q \in \Re^{nxn}, R \in \Re^{mxn}, tales\ que\ A = RQ$$

$$con\ R = \begin{bmatrix} R_0 \\ R_1 \end{bmatrix}, R_1 \in \Re^{nxn}, \text{triangular superior}, si\ m \geq n,$$

$$\text{o}\ \ R = \begin{bmatrix} 0 & R_1 \end{bmatrix}, R_1 \in \Re^{mxm}, \text{triangular superior}, si\ n \geq m,$$

Representación de la señal

Problemas de mínimos cuadrados planteados:

Problema lineal de mínimos cuadrados

$$\min_{x \in \Re^n} \|Ax - b\|_2^2$$

$$Dada\ A \in \Re^{mxn}\ \text{y}\ Q \in \Re^{mxm}, R \in \Re^{mxn}, tales\ que\ A = QR$$

$$\|Ax - b\|_2^2 = \|Q^T(Ax - b)\|_2^2 = \|Q^T Ax - Q^T b\|_2^2 = \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - c \right\|_2^2 = \|R_1 x - c_1\|_2^2 + \|c_2\|_2^2$$

$\min_{x \in \Re^n} \|Ax - b\|_2^2$ se convierte en resolver el sistema triangular $R_1 x = c_1$

G.H.Golub and C.F. Van Loan. "Matrix Computations"
The Johns Hopkins Univ. Press, 3rd Ed. Baltimore, 1996.

Ake Björck. "Numerical methods for least squares problems"
Philadelphia. SIAM, 1996.

Representación de la señal

Es conveniente:
- Utilizar representaciones cómodas: desarrollos en serie de funciones conocidas
-Utilizar representaciones orientadas a objetivos: propiedades útiles como la periodicidad,…

Análisis de Fourier:
Métodos que permiten descomponer una señal en suma de componentes individuales que pueden ser fácilmente producidas y observadas.

•Se puede utilizar el sistema trigonométrico como conjunto ortonormal de funciones base: **Series de Fourier**

•Se pueden utilizar otros conjuntos de funciones base que permitan analizar otras/más propiedades:

- •Transformada **Wavelet**: se usa un conjunto de funciones que son versiones desplazadas y escaladas de una función finita que se denomina Wavelet madre.

Representación de la señal: Series de Fourier

15—4 DEFINICIÓN. *Sea* $S = \{\phi_0, \phi_1, \phi_2, \ldots\}$ *ortonormal en* $[a, b]$ *y supongamos que* $f \in R$ *en* $[a, b]$. *La notación*

14) $$f(x) \sim \sum_{n=0}^{\infty} c_n \phi_n(x)$$

significará que los números $c_0, c_1, c_2, \ldots$ *están dados por las fórmulas siguientes:*

15) $$c_n = (f, \phi_n) = \int_a^b f(x)\,\overline{\phi_n(x)}\,dx \qquad (n = 0, 1, 2, \ldots).$$

La serie 14) *se llama la serie de Fourier de* f *relativa al sistema* S *y los números* $c_0, c_1, c_2, \ldots$ *se llaman los coeficientes de Fourier de* f *relativos a* S.

NOTA. Cuando S es el conjunto ortonormal particular de funciones trigonométricas descrito en 6), la serie se llama simplemente la *serie de Fourier generada por* f. En este caso, escribimos 14) en la forma

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \operatorname{sen} nx),$$

¿Cómo está de cerca una Serie de Fourier de su función?

15—5 TEOREMA. *Sea* $S = \{\phi_0, \phi_1, \phi_2, \dots\}$ *ortonormal en* $[a, b]$, *supongamos que*

$$f(x) \sim \sum_{n=0}^{\infty} c_n \phi_n(x).$$

Designemos por $s_n(x)$ *la suma parcial*

17) $$s_n(x) = \sum_{k=0}^{n} c_k \phi_k(x) \quad (n = 0, 1, 2, \dots).$$

Si $b_0, b_1, b_2, \dots$, *son números complejos arbitrarios, consideremos*

18) $$t_n(x) = \sum_{k=0}^{n} b_k \phi_k(x) \quad (n = 0, 1, 2, \dots).$$

Entonces tenemos la identidad siguiente:

19) $$\int_a^b |f(x) - t_n(x)|^2 \, dx = \int_a^b |f(x)|^2 \, dx - \sum_{k=0}^{n} |c_k|^2 + \sum_{k=0}^{n} |c_k - b_k|^2,$$

de la cual se obtiene la desigualdad

20) $$\int_a^b |f(x) - s_n(x)|^2 \, dx \leq \int_a^b |f(x) - t_n(x)|^2 \, dx.$$

¿Qué pasa cuando una Serie de Fourier está de cerca de su función?

15—6 TEOREMA. *Sea* $\{\phi_0, \phi_1, \phi_2, \ldots\}$ *ortonormal en* $[a, b]$, *supongamos que* $f \in R$ *en* $[a, b]$, *y que*

$$f(x) \sim \sum_{n=0}^{\infty} c_n \phi_n(x).$$

Entonces

a) *La serie* $\sum |c_n|^2$ *converge y satisface la desigualdad*

21) $$\sum_{n=0}^{\infty} |c_n|^2 \leq \int_a^b |f(x)|^2 \, dx$$ (*desigualdad de Bessel*).

b) *La igualdad*

$$\sum_{n=0}^{\infty} |c_n|^2 = \int_a^b |f(x)|^2 \, dx$$ (*fórmula de Parseval*)

es válida si, y únicamente si, también se verifica que

$$\lim_{n \to \infty} ||f - s_n|| = 0,$$

donde $\{s_n\}$ *es la sucesión de sumas parciales definidas en* 17).

¿Cuándo converge una Serie de Fourier a la función que la ha generado?

15–18 TEOREMA. (*Jordan*). *Supongamos que* $f \in R^*(0, 2\pi)$ *y que* f *tiene período* 2π. *Admitamos que existen un punto* x *y un intervalo* $[x - \delta, x + \delta]$ *en el cual* f *es de variación acotada. En estas condiciones la serie de Fourier generada por* f *converge para aquel valor de* x *hacia la suma* $[f(x+) + f(x-)]/2$. *Si, además* f *es continua en* x, *la serie converge hacia* $f(x)$.

T.M. Apostol. "Análisis Matemático"
Addison-Wesley. 1960

Representación de la señal

Transformada de Fourier Discreta

Nos ocuparemos particularmente del sistema especial trigonométrico $S = \{\phi_0, \phi_1, \phi_2, \ldots\}$ para el cual

$$6) \qquad \phi_0(x) = \frac{1}{\sqrt{2\pi}}, \quad \phi_{2n-1}(x) = \frac{\cos nx}{\sqrt{\pi}}, \quad \phi_{2n}(x) = \frac{\operatorname{sen} nx}{\sqrt{\pi}} \quad (n = 1, 2, \ldots).$$

15-14 Otras formas de series de Fourier. Utilizando las fórmulas $2\cos nx = e^{inx} + e^{-inx}$ y $2i \operatorname{sen} nx = e^{inx} - e^{-inx}$, la serie de Fourier generada por f puede expresarse en función de exponenciales complejas como sigue:

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \operatorname{sen} nx) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (\alpha_n e^{inx} + \beta_n e^{-inx}),$$

donde $\alpha_n = (a_n - ib_n)/2$ y $\beta_n = (a_n + ib_n)/2$. Si ponemos $\alpha_0 = a_0$ y $\alpha_{-n} = \beta_n$ podemos escribir la forma exponencial más brevemente así:

$$f(x) \sim \sum_{n=-\infty}^{\infty} \alpha_n e^{inx}.$$

Las fórmulas 16) para los coeficientes son ahora

$$\alpha_n = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-int}\, dt \qquad (n = 0, \pm 1, \pm 2, \ldots).$$

Si f tiene período 2π, el intervalo de integración puede reemplazarse por cualquier otro intervalo de longitud 2π.

Con mayor generalidad, si $f \in R$ en $[0, p]$ y f tiene período p, escribimos

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2\pi n x}{p} + b_n \operatorname{sen} \frac{2\pi n x}{p} \right)$$

para indicar que los coeficientes vienen dados por las fórmulas

$$a_n + \frac{2}{p} \int_0^p f(t) \cos \frac{2\pi n t}{p} \, dt,$$

$$b_n = \frac{2}{p} \int_0^p f(t) \operatorname{sen} \frac{2\pi n t}{p} \, dt \qquad (n = 0, 1, 2, \ldots).$$

En forma exponencial podemos escribir

$$f(x) \sim \sum_{n=-\infty}^{\infty} a_n e^{2\pi i n x/p},$$

donde

$$a_n = \frac{1}{p} \int_0^p f(t) e^{-2\pi i n t/p} \, dt, \qquad \text{si } n = 0, \pm 1, \pm 2, \ldots .$$

Todos los teoremas de convergencia para series de Fourier de período 2π pueden también aplicarse al caso de un período cualquiera p efectuando un cambio de escala conveniente.

Representación de la señal

Señales analógicas periódicas: Series de Fourier

Dada una señal periódica, de período T, que cumple ciertas condiciones se puede expresar como una suma de funciones exponenciales complejas:

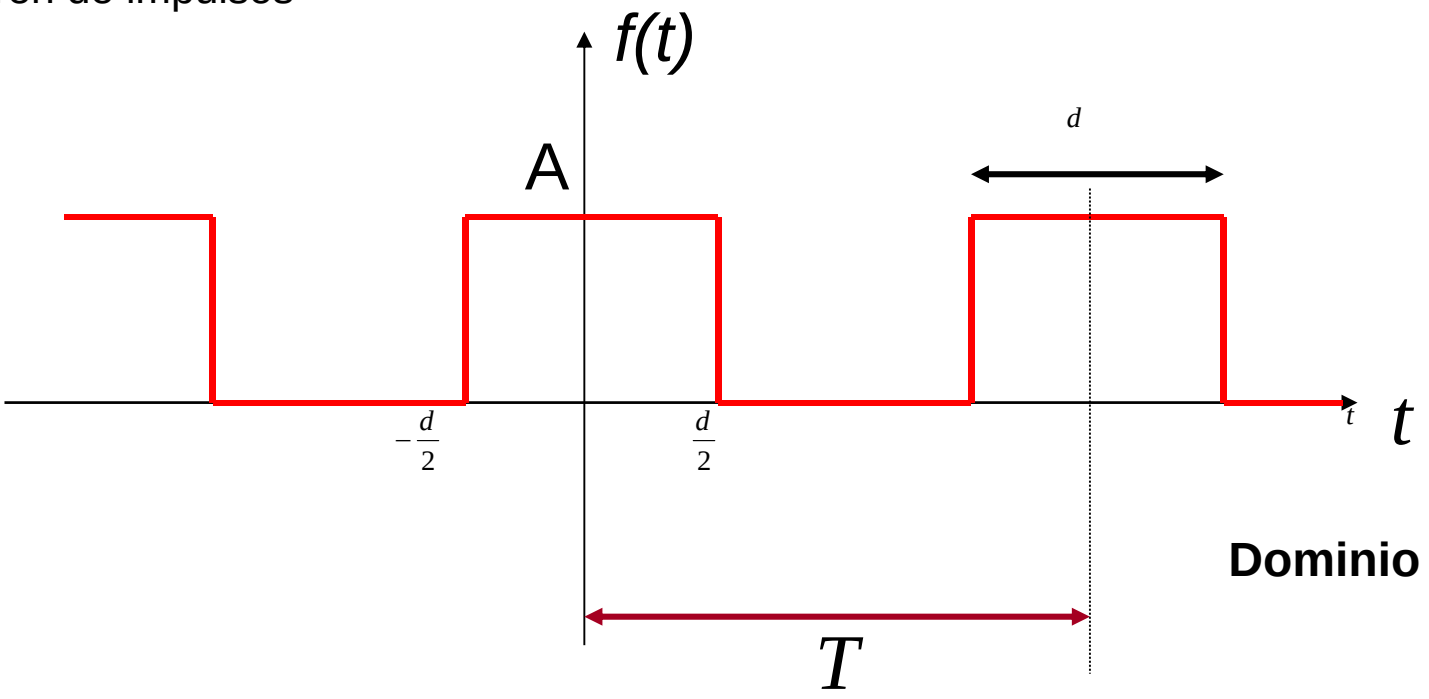
$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{jnw_0 t}$$

siendo los coeficientes C_n de la forma

$$C_n = \frac{1}{T}\int_0^T f(t) e^{-jnw_0 t} dt \quad \text{con} \quad w_0 = \frac{2\pi}{T}$$

(Obsérvese que los coeficientes son complejos y las exponenciales también pero su producto es real

Ejemplo: Tren de impulsos



Dominio del tiempo

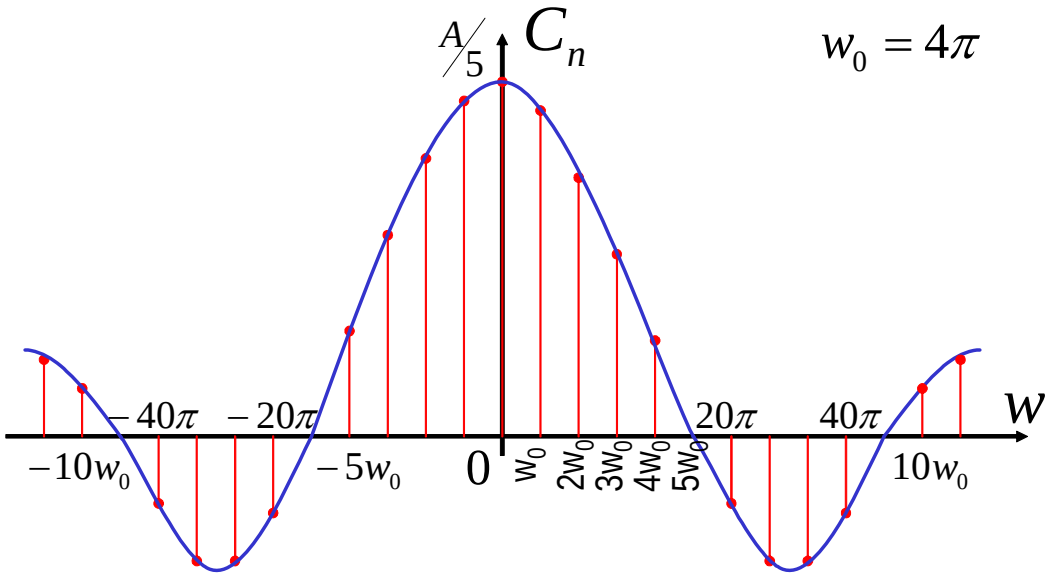
Representación de la señal

Dominio de la frecuencia

$$i_p(t) = \frac{Ad}{T} \sum_{n=-\infty}^{\infty} \frac{\sin(n\pi d/T)}{n\pi d/T} e^{j2\pi nt/T} \qquad C_n = \frac{Ad}{T} Sinc\left(\frac{n\pi d}{T}\right)$$

Si representamos C_n en función de $\frac{2\pi}{T} n = nw_0$

Sea: $d = \frac{1}{10}$ $T = \frac{1}{2}$ $\therefore$ $C_n = \frac{A}{5} Sinc\left(\frac{n\pi}{5}\right)$



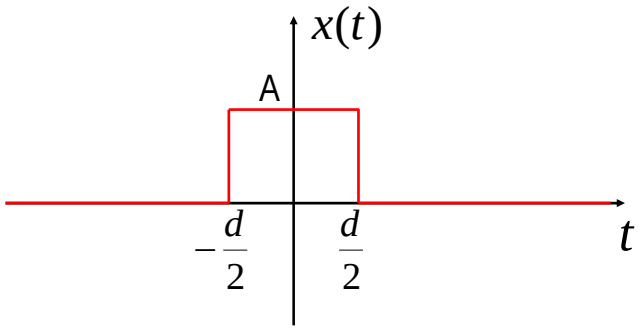
Representación de la señal

Señales analógicas NO periódicas

Si la señal no es periódica se puede tratar de extender la expresión anterior, por ejemplo calculando el límite cuando T tiende a infinito

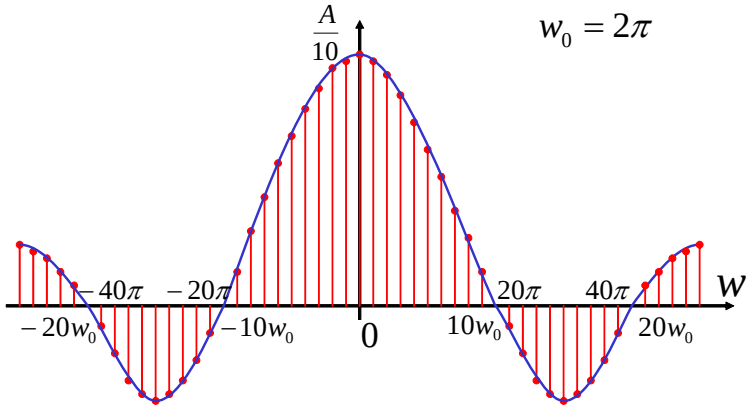
Ejemplo: Pulso aislado

Sea una señal no periódica:



¿Será posible obtener un espectro para este tipo de señal?

Ahora $d = \frac{1}{10} \quad T = 1 \quad \therefore \quad C_n = \frac{A}{10} Sinc\left(\frac{n\pi}{10}\right)$



Cuando aumenta el período T:

- El espectro de amplitud conserva su forma aunque la amplitud disminuye
- El espaciamiento entre las componentes decrece, por lo que el espectro se hace más denso
- Cuando $T \to \infty$ $\Delta w = w_0 = \frac{2\pi}{T} \to 0$
- $w = nw_0$ se convierte en una variable continua

Representación de la señal

Generalizamos esta idea para funcione no periódicas con el paso al límite:

Transformada de Fourier de una señal s(t)

$$S(w) = \int_{-\infty}^{+\infty} s(t) e^{-i\,2\pi wt}\, dt$$

Permiten el cambio de señales del dominio del tiempo al de la frecuencia y viceversa

Transformada inversa

$$s(t) = \int_{-\infty}^{+\infty} S(w) e^{i\,2\pi wt}\, dw$$

Para calcular la Transformada de Fourier en un computador es necesario sustituir la señal continua $s(t)$ por una señal discreta $s(nT)$ y limitar el muestreo a un cierto número de veces N.

Así:

$$S^*(w) = \sum_{n=0}^{N-1} s(nT) e^{-j\,2\pi wnT}$$

También es posible calcular S únicamente para un número limitado de valores de la frecuencia, $k\Delta w$

$$S^*(k\Delta w) = \sum_{n=0}^{N-1} s(nT) e^{-j\,2\pi k\Delta wnT}$$

Señales digitales: Transformada de Fourier Discreta (DFT)

Dadas dos secuencias periódicas con período N se define las Transformadas Discretas de Fourier directa e inversa de la forma

Transformada directa

$$X[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[k]\, e^{-j\frac{2\pi}{N}kn}$$

Transformada inversa

$$x[n] = \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn}$$

Cálculo de la DFT: FFT

Obsérvese que

$$X[n] = \sum_{k=0}^{N-1} x[k] W_N^{kn} \text{ con } W_N = e^{-j\frac{2\pi}{N}}$$

Por tanto si llamamos con $n=0,1,\ldots,N-1$

$$y_n = X[n], n = 0,1,\ldots,N-1 \qquad x_k = X[k], k = 0,1,\ldots,N-1$$

$$F_N = \left[(F_N)_{k,n}\right] = \left[W_N^{kn}\right]$$

se tiene $y = F_N x$

FFT = Producto Matriz-Vector

Charles Van Loan. “Computational frameworks for the fast Fourier transform” . Philadephia : SIAM, 1992

Representación de la señal

Transformada **Wavelet**: se usa un conjunto de funciones que son versiones desplazadas y escaladas de una función finita que se denomina Wavelet madre. Genera una matriz ***W***, con buenas propiedades (ortogonalidad, dispersión, …)

Se puede usar para resolver problemas numéricos, por ejemplo transformando un sistema de ecuaciones denso en uno disperso:

$$Ax = b \Leftrightarrow WAW^T Wx = Wb \Leftrightarrow Cy = c$$

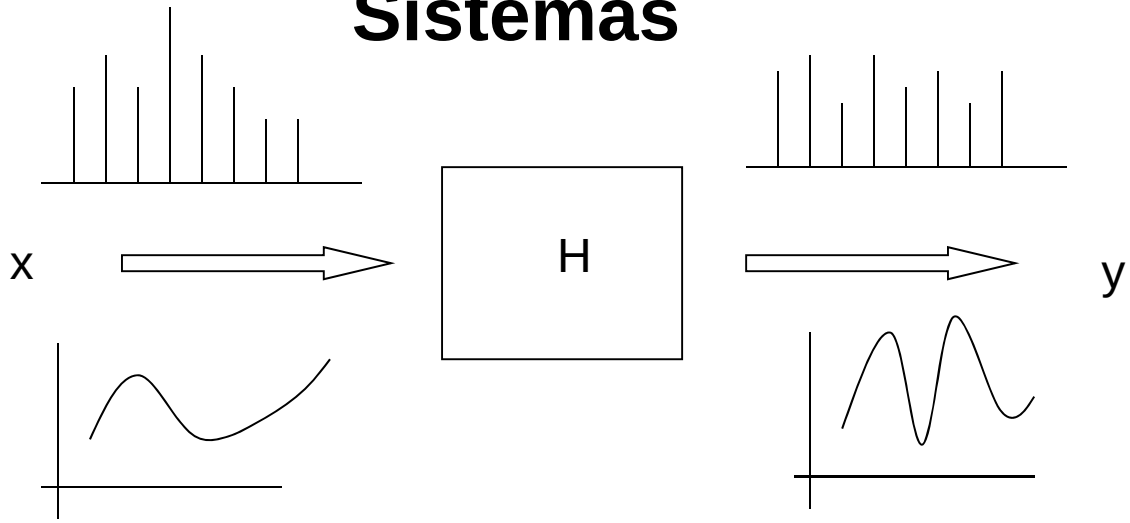
I.Daubechies. "Ten Lectures on Wavelets". SIAM: Society for Industrial and Applied Mathematics, 1992.

Robi Polikar "The Wavelet Tutorial".
http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

V.M. Garcia, L. Acevedo, A.M. Vidal. "Variants of algebraic wavelet-based multigrid methods: Application to shifted linear systems". Applied Mathematics and Computation. Volumen: 202 (2008) PP. 287-299. 2008

L. Acevedo, V. M. Garcia, and A.M.Vidal. "Compatibility of Scalapack with the Discrete Wavelet Transform". Lecture Notes in Computer Science. Proceedings of ICCS 2007. Beijing (China) May 2007. Volumen: 4487 Pp. 152-159. 2007

Sistemas



$$y = Hx, H \in \Re^{Nxp}, x \in \Re^{n}, N \geq p$$ Filtrado, Identificación,…

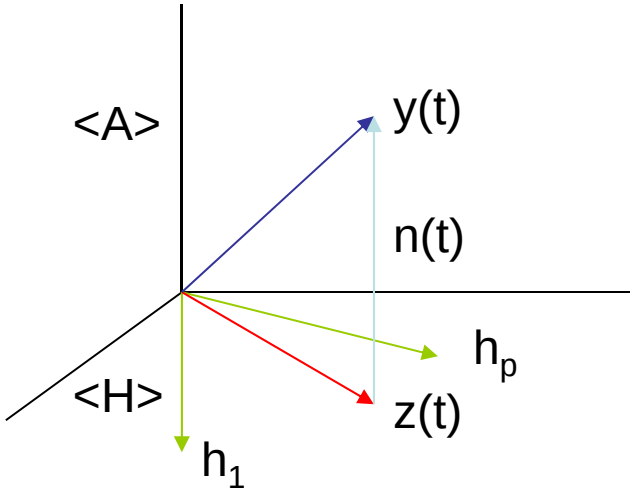
$$y = Hx + n, H \in \Re^{Nxp}, x \in \Re^{n}, N \geq p$$ n es el ruido

Las columnas de H generan el subespacio de la señal: <H>=span[h_1, h_2,…, h_p]. Contiene todas las medidas (señales) que se pueden construir como C.L. de estas columnas (normalmente son L.I.). p=dim(H)

Se pueden construir (N-p) modos ortogonales a h_i, i=1,2,...,p y organizarlos en la forma A=[a_1, a_2,…, a_{N-p}] con <A>= span[a_1, a_2,…, a_{N-p}] . dim(A)= N-p

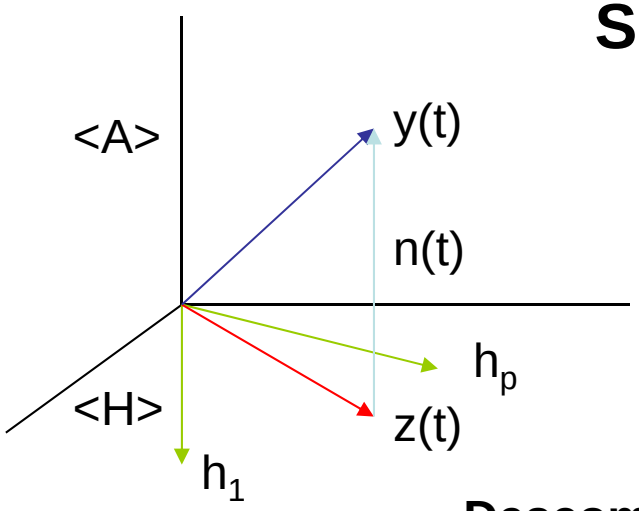
Problemas de rango: la Descomposición en Valores Singulares (SVD)

- De tal forma que <S> = <H> + <A> es una suma directa de supespacios que genera el espacio de señales recibidas
- Normalmente A se denomina el subespacio del ruido. Las señales medidas toman la forma:
- y(t)=z(t)+n(t) donde z(t) es la señal y n(t) es el ruido



Es importante disponer de herramientas que permitan separar la señal del ruido. Para ello es necesario poder determinar de forma precisa el rango de H y encontrar bases ortonormales para los espacios <H> y <A>

Problemas de rango: la Descomposición en Valores Singulares (SVD)



Es importante disponer de herramientas que permitan separar la señal del ruido. Para ello es necesario poder determinar de forma precisa el rango de H y encontrar bases ortonormales para los espacios <H> y <A>

Descomposición en Valores Singulares de H

$$H = U\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}V^T, H \in \Re^{Nxp}, N \geq p, rank(H) = r \leq p, U \in \Re^{NxN}, V \in \Re^{pxp}, \text{ ortogonales,}$$

$$S \in \Re^{rxr}, diagonal, con\ \ S = diag(\sigma_1, \sigma_2, ..., \sigma_r), \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > \sigma_{r+1} = ... = \sigma_p = 0$$

Las r primeras columnas de V permiten expandir el subespacio de la señal

Las N-r últimas columnas de V permiten expandir el subespacio del ruido

V.Klema & A.J.Laub. “The Singular Value Decomposition: its Computation and some Aplications”. IEE Transactions on Automatic Control. Vol. AC-25, No. 2, April 1980

G.H.Golub and C.F. Van Loan. "Matrix Computations"
The Johns Hopkins Univ. Press, 3rd Ed. Baltimore, 1996.

On-line algorithm for signal separation based on SVD
D. Callaerts, J. Vandewalle, W. Sansen, M. Moonen

Edited by Ed. F. Deprettere. “SVD and signal processing : Algorithms, applications and architectures”. Amsterdam. North-Holland, 1988.

Edited by Richard J. Vaccaro. “SVD and signal processing, II : algorithms, analysis, and applications”. Amsterdam. Elsevier, 1991

Sistemas: Convolución

La Convolución es una operación que permite, dado un sistema caracterizado por una respuesta al impulso *h(t)*, predecir cual va a ser la señal que encontramos a la salida *y(t)*. Si consideramos sólo sistemas lineales e invariantes en el tiempo, para una determinada entrada x(t), la salida *y(t)* se obtiene de forma teórica como:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)\mathrm{d}\tau$$

Discretizando esta integral

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Limitándola a señales periódicas:

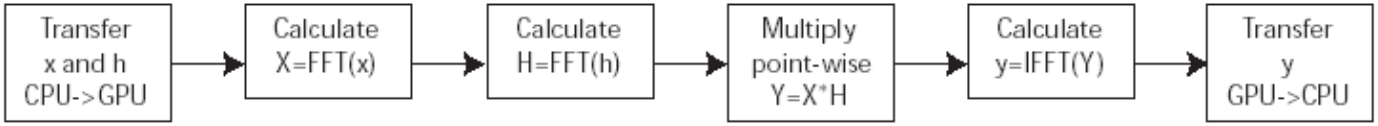
$$y[n] = \sum_{k=0}^{N-1} x_1[k]x_2[n-k] \qquad \longrightarrow \qquad y[n] = x_1[n] \otimes x_2[n];$$

Cálculo de la convolución

$$X[k] = DFT(x[n])$$
$$H[k] = DFT(h(n])$$
$$Y[k] = X[k] * Y[k] \quad \text{(producto punto a punto)}$$
$$y[n] = x[n] \otimes h[n] = IDFT(Y[k])$$

- Existen numerosos ejemplos donde se aplica la operación de convolución.
- Puede desarrollarse sobre tarjetas GPU

Esquema de convolución que sigue el ejemplo

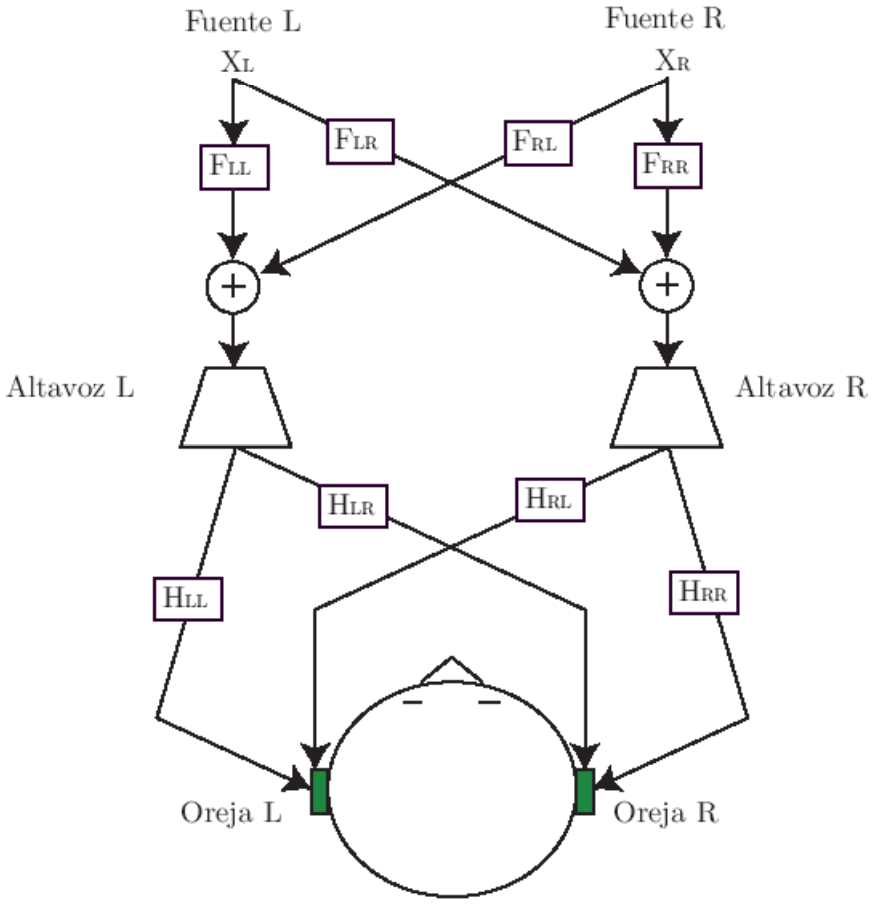


NVIDIA utiliza para hacer la DFT su propia libreria CUFFT [16], que está optimizada para ejecutar tanto transformaciones de Fourier directas como inversas, así como múltiples transformaciones en paralelo.

Sin embargo, este esquema no permite realizar convoluciones en tiempo real, ni permite explotar el paralelismo de multiples canales.

Ejemplo de sistemas donde se aplica la convolución

Cancelador de Crosstalk



Sistemas

Cancelador de Crosstalk

Analizando para la figura 6.2 para la Oreja Izquierda, observamos que la señal que llega a ésta es:

$$L = (X_L * F_{LL} + X_R * F_{RL}) * H_{LL} + (X_L * F_{LR} + X_R * F_{RR}) * H_{RL} \tag{6.1}$$

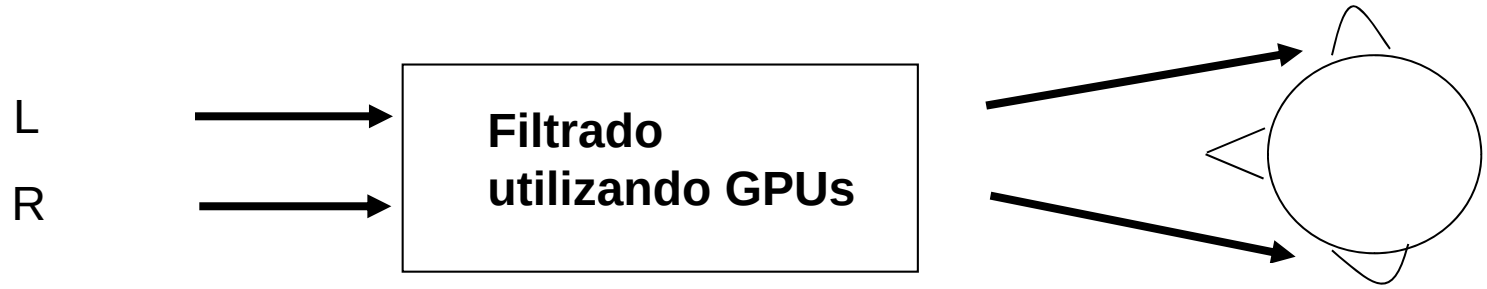
Poniendo la ecuación anterior en función de las dos fuentes:

$$L = X_L * (F_{LL} * H_{LL} + F_{RL} * H_{RL}) + X_R * (F_{RL} * H_{LL} + F_{RR} * H_{RL}) \tag{6.2}$$

Una aplicación crosstalk, consistirá en diseñar un banco de filtros F_{RL}, F_{RL}, F_{RL} y F_{RL} de forma que:

$$F_{LL} * H_{LL} + F_{LR} * H_{RL} = 1 \tag{6.3}$$

$$F_{RL} * H_{LL} + F_{RR} * H_{RL} = 0 \tag{6.4}$$



http://www.upv.es/noticias-upv/noticia-8010-control-activo-es.html

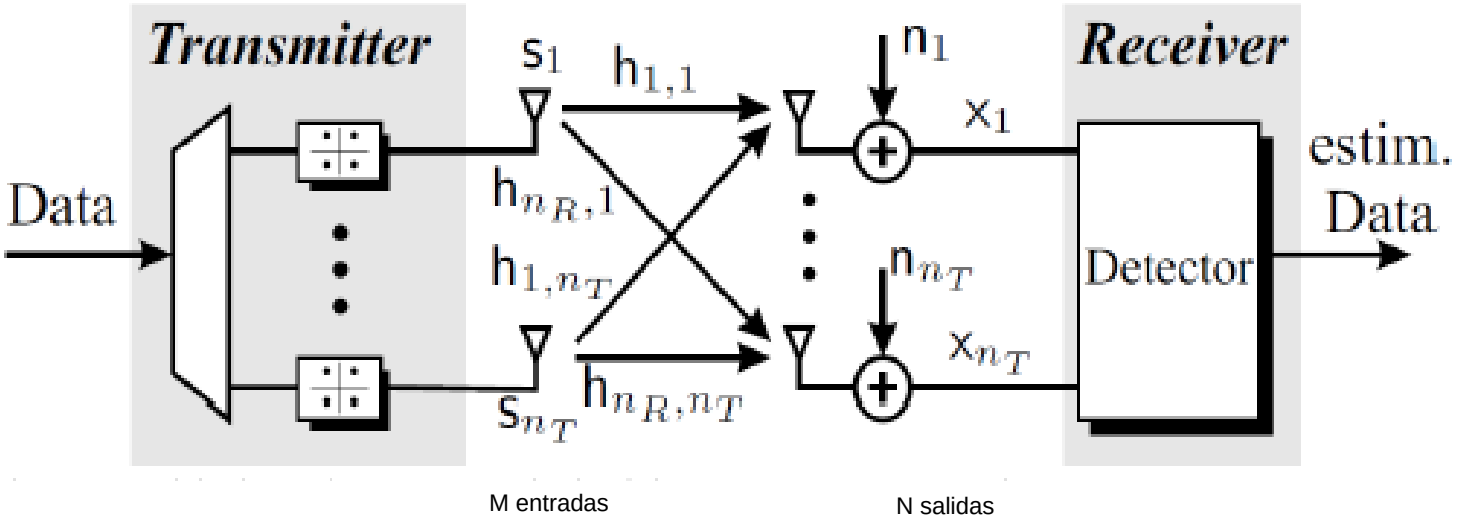
Samir S. Soliman and Mandyam D. Srinath. “Continuous and Discrete Signals and Systems”.1997.

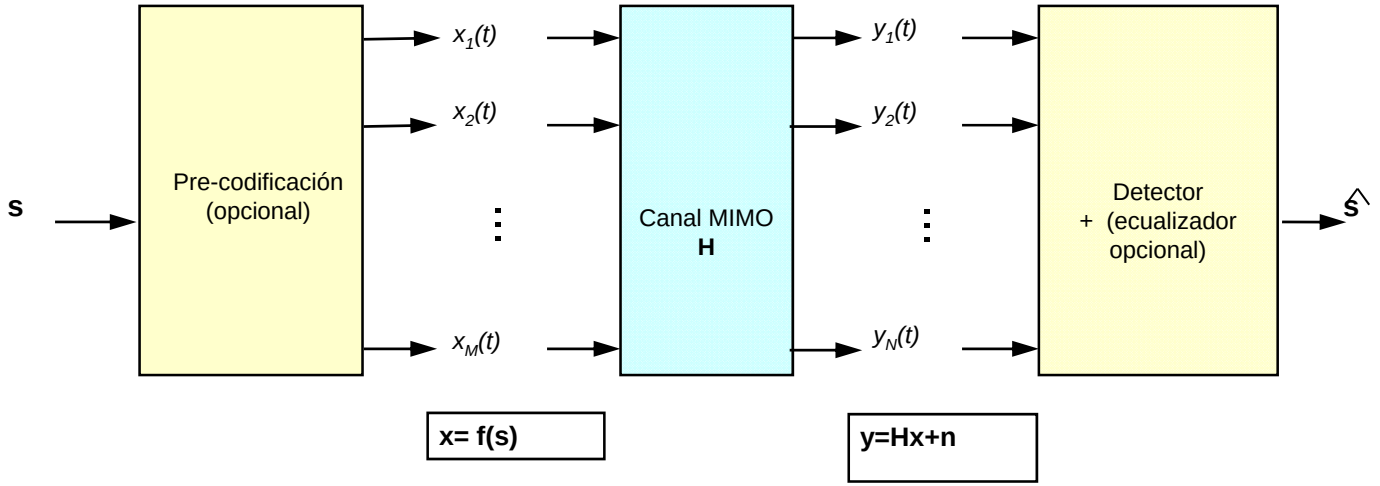
Charles Van Loan. “Computational frameworks for the fast Fourier transform” . Philadephia : SIAM, 1992

F. J. Matinez-Zaldivar A. Gonzalez J.A. Belloch, A. M. Vidal. “Multichannel acoustic signal processing on gpu”. Procedings of the 10th International Conference on Computational and Mathematical Methods in Science and Engineering, 1:181–187, June 2010.

F. J. Matinez-Zaldivar A. Gonzalez J.A. Belloch, A. M. Vidal. “Real-time massive convolution for audio applications on gpu”. Journal of Supercomputing, Published online: 19 April 2011

Sistemas MIMO





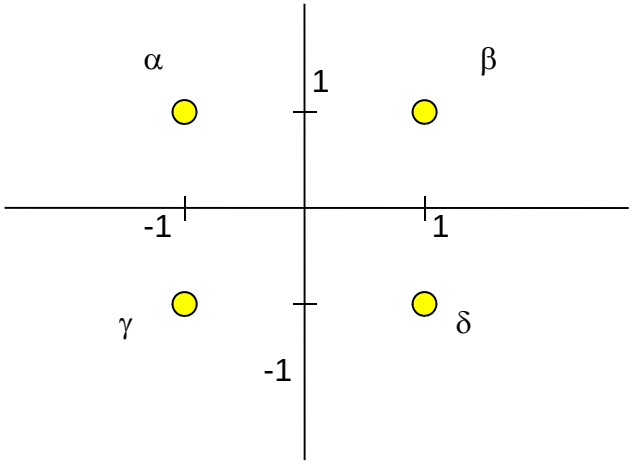
Objetivo: Estimar x

Sistemas MIMO

Ejemplo: Detección

Universo con cuatro caracteres: S={α, β, γ, δ}

$$\alpha = \begin{bmatrix} -1 \\ 1 \end{bmatrix}; \beta = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \gamma = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; \delta = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



$$y = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

$$\min_{x \in K^2} \left\| \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\|$$

$$K \subset Z$$

Objetivo: Estimar x

Sistemas Mimo

Caso general:

$$\min_{x \in Z^n} \left\| \begin{bmatrix} h_{11} & h_{12} & . & . & h_{1n} \\ h_{21} & h_{22} & . & . & h_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ h_{m1} & h_{m2} & . & . & h_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ . \\ x_n \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_m \end{bmatrix} \right\|$$

Problema planteado: **Problema de mínimos cuadrados con solución entera.**

•Formas de resolverlo:

- •Resolver el problema de mínimos cuadrados real y aproximar al entero más próximo. Método subóptimo
- •Búsqueda exhaustiva. Método de máxima verosimilitud.
- •Búsqueda exhaustiva mediante “ramificación y poda” tras aplicar una descomposición QR: Algoritmo “Sphere decoding”. Método de máxima verosimilitud.
- •Otros métodos heurísticos. Métodos subóptimos

Sistemas MIMO

T. Kailath, H. Vikalo, and B. Hassibi, “Space-Time Wireless Systems: From Array Processing to MIMO Communications”, Cambridge University Press, 2006, ch. MIMO receive algorithms, pp. 302-322.”

D. Wubben, R. Bohnke, V. K¨uhn, K.-D. Kammeyer, Near-Maximum-Likelihood Detection of MIMO Systems using MMSE-Based Lattice Reduction, in: IEEE International Conference on Communications, Paris, France, 2004.

G. J. Foschini, G. D. Golden, R. A. Valenzuela, P. W. Wolniansky, Simplified processing for high spectral eciency wireless communication employing multi-element arrays, IEEE Journal on Selected Areas in Communications 17 (11) (1999) 1841–1852.

B. Hassibi, H. Vikalo, On Sphere Decoding algorithm. Part I, the expected complexity, IEEE Transactions on Signal Processing 54 (5)

A. J. Paulraj, D. A. Gore, R. U. Nabar, H. Bolcksei, An overview of MIMO communications - a key to gigabit wireless, Proceedings of the IEEE 92 (2) (2004) 198–218.

Sistemas MIMO

Sandra Roger, Alberto Gonzalez, Vicenç Almenar, Antonio M. Vidal. "Extended LLL Algorithm for Efficient Signal Precoding in Multiuser Communication Systems". IEEE Communications Letters. Volumen: E14, No. 3. Pp. 220-222. Marzo 2010

V,M.García, S.Roger, R.A.Trujillo, A.M.Vidal, A.Gonzalez. "A Deterministic Lower Bound for Radius in Sphere Decoding Search". The 2010 International Conference on Avanced Technologies for Communications. Saigon. October 2010

S. Roger, A. Gonzalez, V. Almenar, A. M. Vidal. "Lattice-Reduction-Aided K-Best MIMO Detector based on the Channel Matrix Condition Number". International Symposium on Communications, Control and Signal Processing (ISCCSP). Limassol, Chipre. Marzo 2010

S. Roger, A. Gonzalez, V. Almenar, A. Vidal. "Variable-Breadth K-Best Detector for MIMO Systems". International Wireless Communications & Mobile Computing Conference (IWCMC).Caen, Francia Junio 2010

OTROS TÓPICOS QUE IMPLICAN COMPUTACIÓN EN PROCESAMIENTO DE LA SEÑAL

- Espacios vectoriales, Álgebra Lineal, Análisis Matemático
- Análisis Funcional

- Señales y Sistemas lineales. Transformadas

- Probabilidad y procesos estocásticos.
- Teoría estadística de la decisión

- Programación

- Métodos Numéricos

- Optimización

- Métodos iterativos

Conclusiones

- El campo del Procesado Digital de Señales es fértil en aplicaciones susceptibles de ser tratadas mediante Computación de Altas Prestaciones
- Aunque solo hemos citado unas pocas existen muchas más, por ejemplo, Procesamiento de Matrice Estructuradas (Toeplitz, Hankel,…)
- Las herramientas computacionales disponibles actualmente son especialmente adecuadas para este campo:
  - Multicores →Problemas de gran dimensión
  - GPUs→ Problemas en tiempo real

Trabajo propuesto

- Del libro “Matrix Computations”, Golub&Van Loan, leer el capítulo 5, puntos 5.1, 5.2 y 5.3