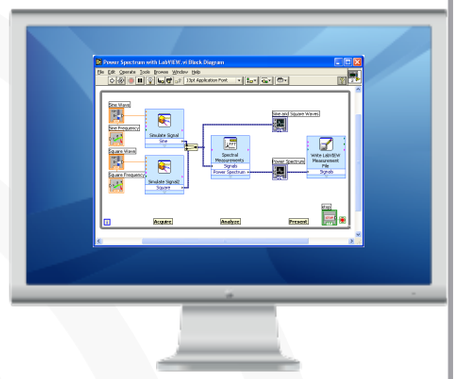
Adquisición de Señales

Sistema de Adquisición y Procesamiento de Datos.....

LABVIEW
MATLAB
PROCESSING
C
JAVA
PURE DATA,
PYTHON,
ETC.....



Aplicaciones

- Instrumentación Biomédica
- Estaciones Metereológicas
- Juegos con "interfaces" gestuales
- Instrumentos musicales electrónicos
- Instrumentación en General
- .....

Medición de fenómenos físicos

- Señales Eléctricas: Voltajes, Corrientes, impedancias, etc
- Señales Mécanicas: desplazamiento vibracion, aceleración, Sonido, etc
- Mediciones de humedad, temperatura, presión, densidad, opacidad, flujo, etc
- Otros

Sensores/ Transductores

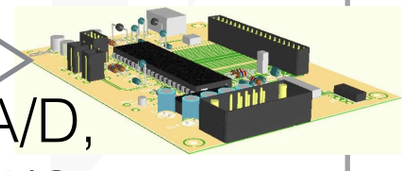
- Micrófonos
- Sensores Piezoeléctricos
- Medidores de Distancia
- Interruptores
- Sensores de contacto
- Sensores ópticos
- Antenas
- Bobinas
- Potenciómetros lineales
- Medidores de capacitancias
- Sensores de presión
- Acelerómetros
- Medidores de corriente
- Termómetros
- .....

I/O Sensores

- Voltaje,
- Corriente,
- Resistencias,
- Niveles lógicos
- Frecuencia,
- I/O serial,
- I^2C,
- SPI,
- etc..
- .....

Acondicionamiento de los sensores

Serial



A/D,
I/O

MICROCONTROLADOR "C"

Amplificación, Nivel DC, Sistemas de protección, Alimentación, Filtros, Adecuación al A/D, Cambios de nivel de voltaje 5v↔3v, etc..

Características eléctricas del cpu

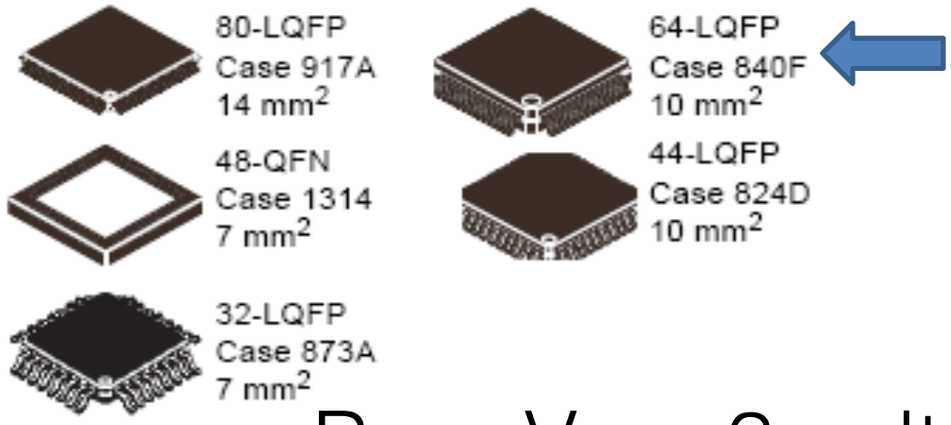
Table 4. Absolute Maximum Ratings

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +3.8	V
Maximum current into V_{DD}	I_{DD}	120	mA
Digital input voltage	V_{In}	–0.3 to V_{DD} + 0.3	V
Instantaneous maximum current Single pin limit (applies to all port pins)[1, 2, 3]	I_D	± 25	mA
Storage temperature range	T_{stg}	–55 to 150	°C

[1] Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive (V_{DD}) and negative (V_{SS}) clamp voltages, then use the larger of the two resistance values.

[2] All functional non-supply pins are internally clamped to V_{SS} and V_{DD}.

MC9S08QE128



Para V_{DD}=3 volt.

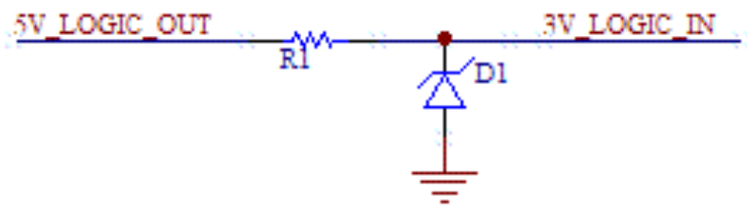
6	P	Input high voltage — all digital inputs	V_{IH}	$V_{DD} > 2.7$ V	0.70 x V_{DD}	—	—	V
	C			$V_{DD} > 1.8$ V	0.85 x V_{DD}	—	—	
7	P	Input low voltage — all digital inputs	V_{IL}	$V_{DD} > 2.7$ V	—	—	0.35 x V_{DD}	
	C			$V_{DD} > 1.8$ V	—	—	0.30 x V_{DD}	
11	P	Pull-up resistors — all digital inputs, when enabled	R_{PU}		17.5	—	52.5	kΩ

2,1 volt.

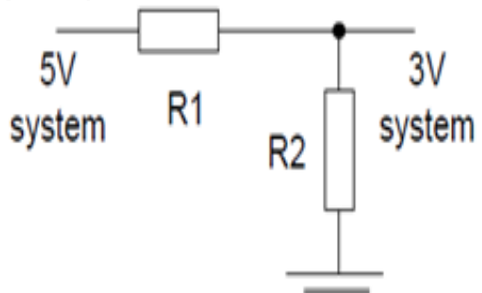
1.0 volt.

VoH= Vdd-0.5 voltios

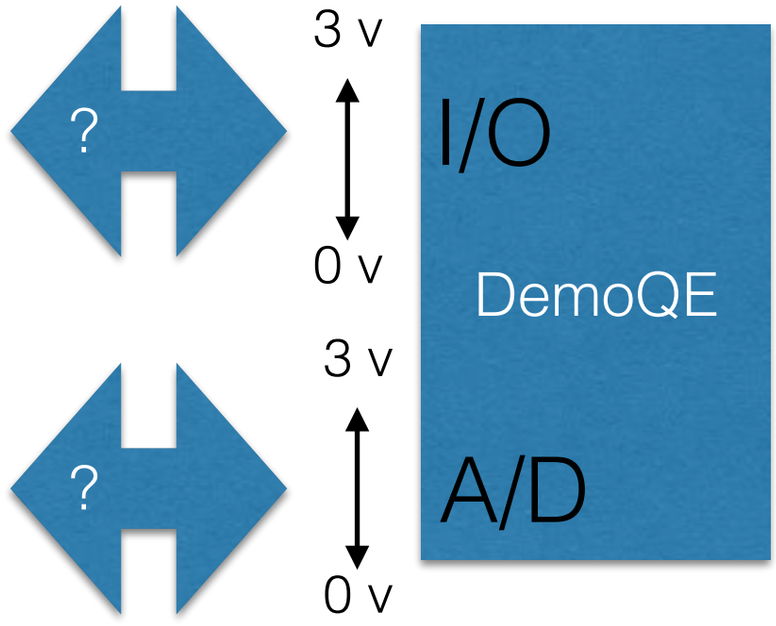
VoL= up to 0.5 voltios



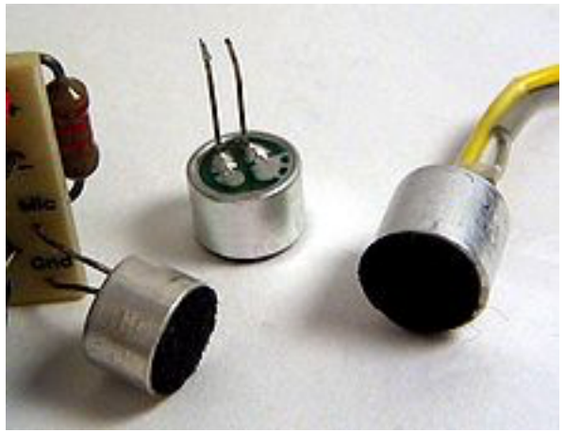
- Por ejemplo **R1=22kΩ, R2=33kΩ**

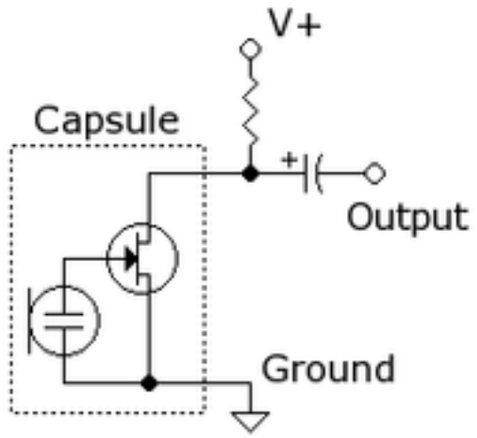


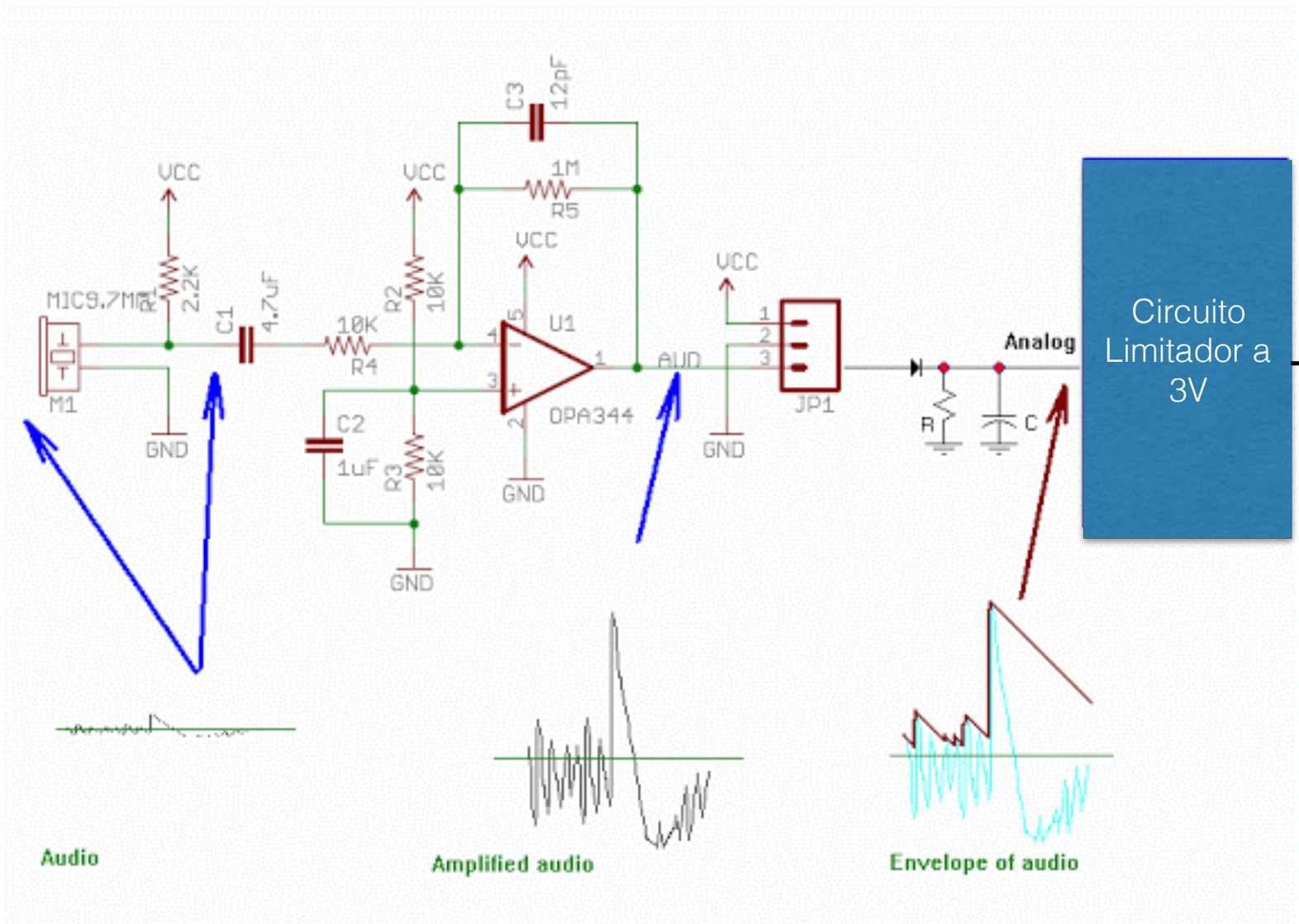
Sensores/ Transductores	Amplitud	Ancho de banda o tiempo de respuesta
Micrófonos	milivoltios	>10kHz
Sensores Piezoeléctricos	entre milivoltios y decenas de voltios	5 kHz
Medidores de Distancia	Pulsos de 5 voltios con duración variable.	100µs a 18ms
Fotoresistencias LDR(light-dependent resistor)	variaciones de resistencia	25-30 ms
Acelerómetros	1 a 3 v, cero=2v	100 Hz
Termómetros	5 v max	bajo



Micrófonos Electret



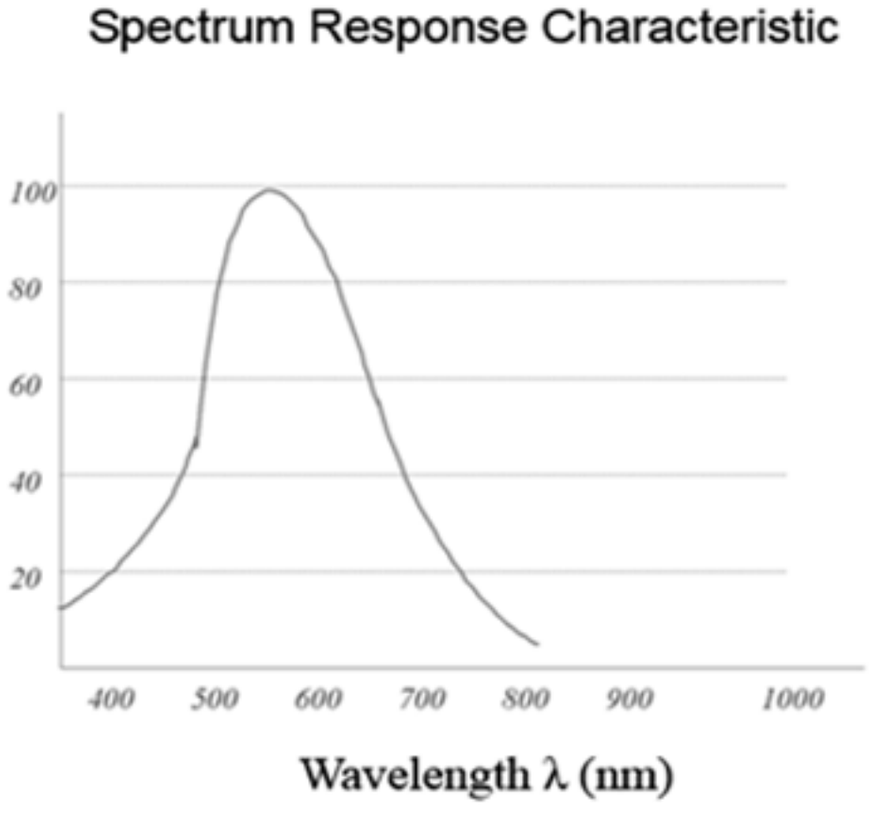




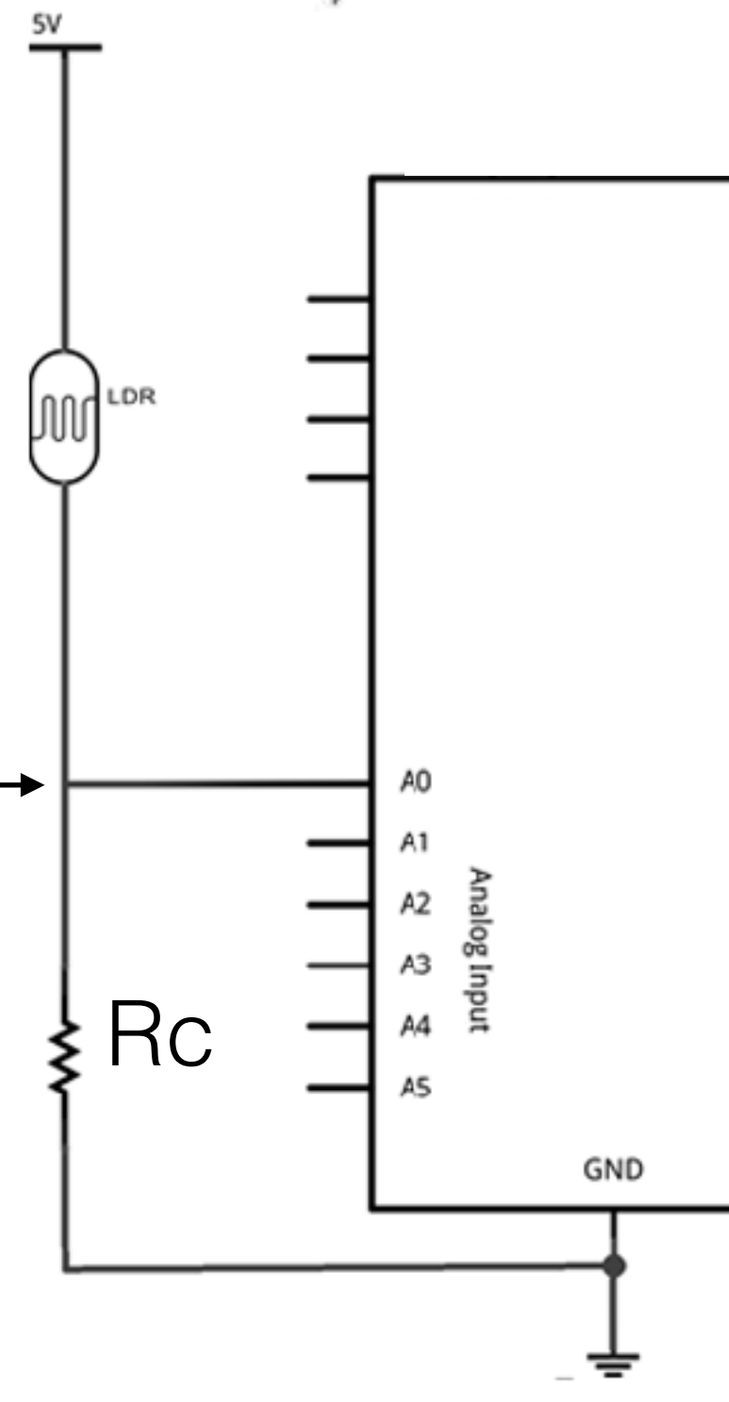
Fotoresistencias (LDR)

$$\frac{I}{I_0} = \left(\frac{R}{R_0}\right)^{-gamma}$$

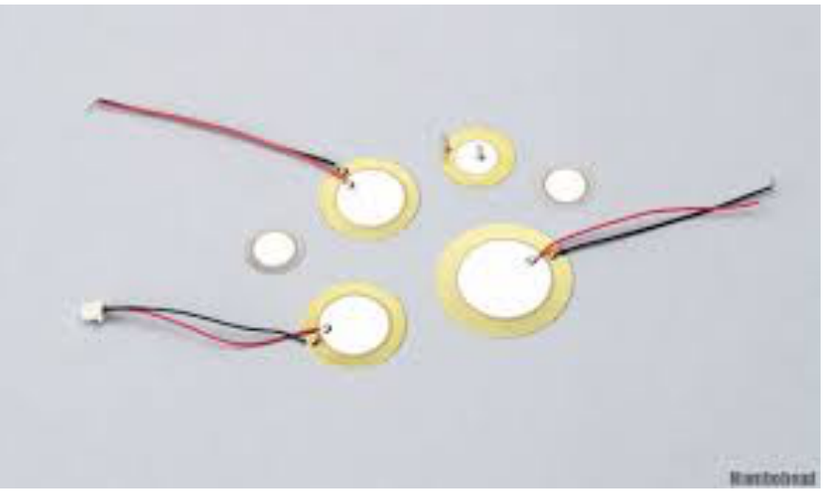
Modelo	Voltaje (V)	Temperatura (ºC)	Pico espectral (nm)	Resistencia oscuridad (KΩ)	Resistencia luz brillante (KΩ)	gamma	Tiempo respuesta (ms)
GL5516	150	-30º+70º	540	5-10	500	0.5	30
GL5528	150	-30º+70º	540	10-20	1000	0.6	25
GL5537-1	150	-30º+70º	540	20-30	2000	0.6	25
GL5537-2	150	-30º+70º	540	30-50	3000	0.7	25
GL5539	150	-30º+70º	540	50-100	5000	0.8	25
GL5549	150	-30º+70º	540	100-200	10000	0.9	25

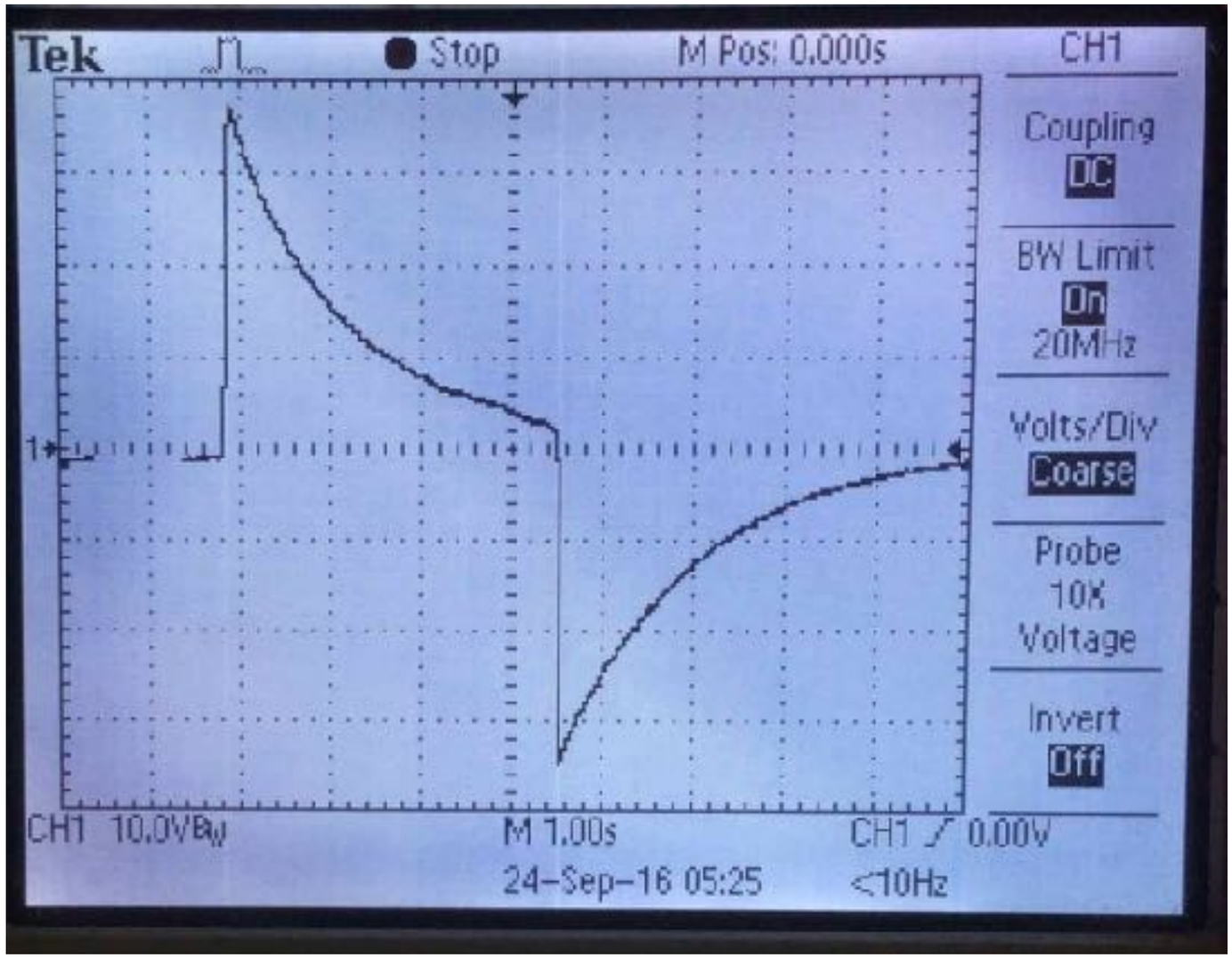


Garantizar que **NO** pase de 3 V

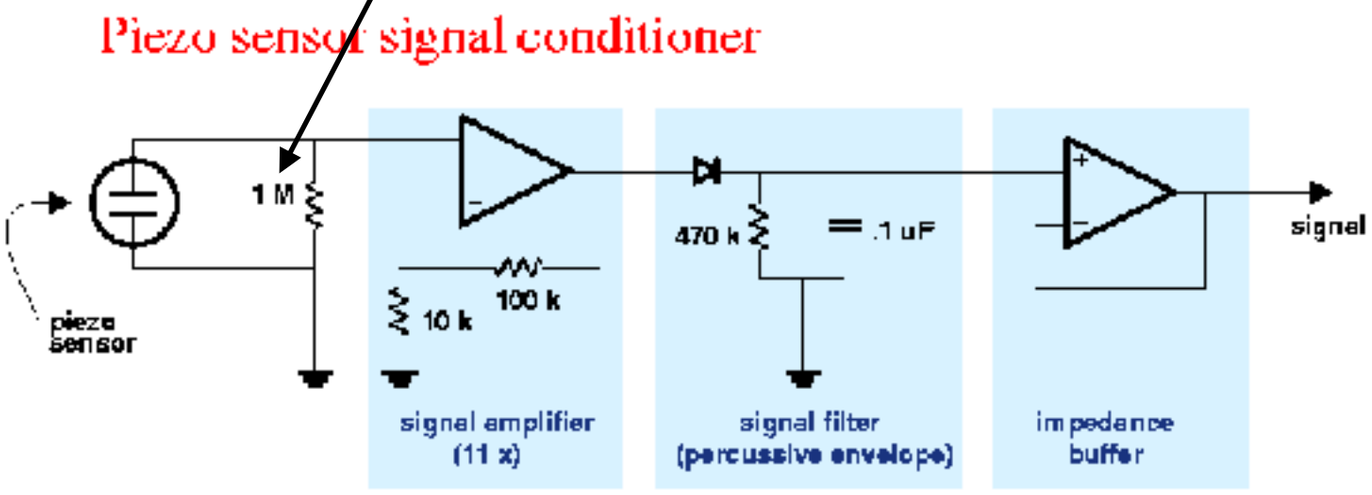


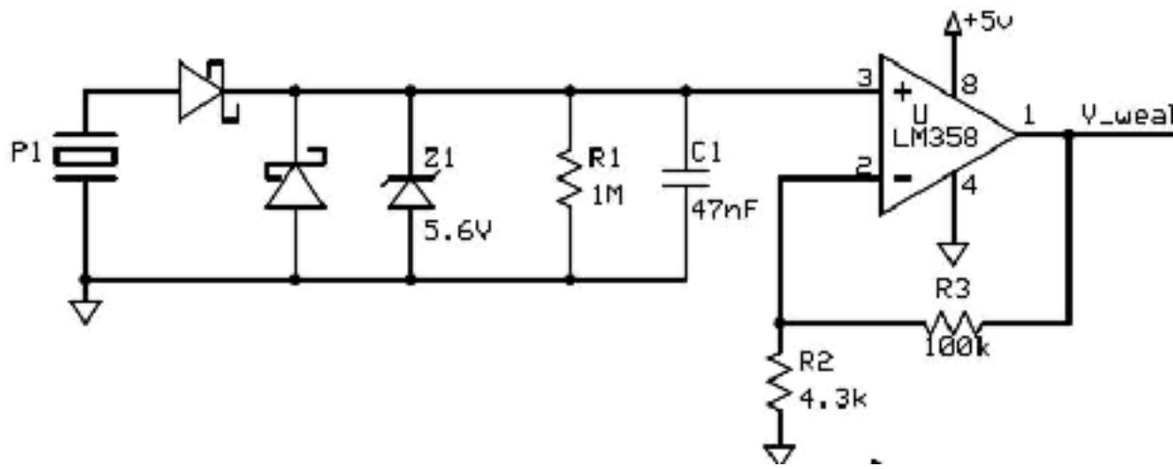
Sensor Piezo-eléctrico



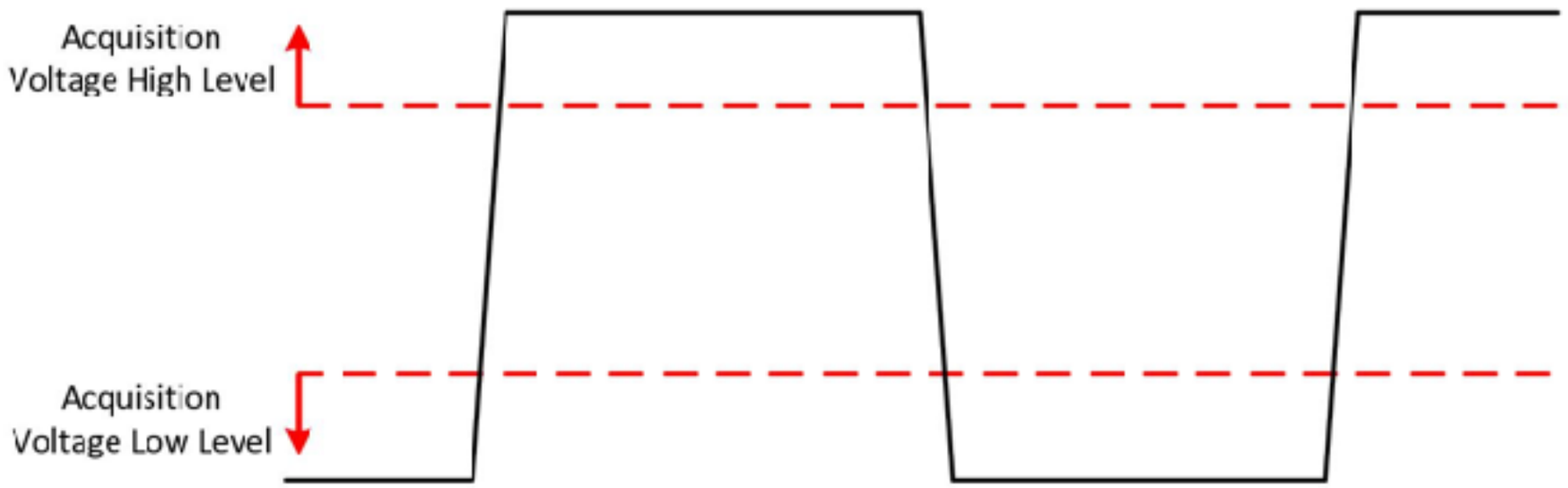


Para 100 ms de caída





Añadir un limitador de voltaje de 3 voltios
Para no dañar el micro


Acquisition
Voltage High Level
Acquisition
Voltage Low Level

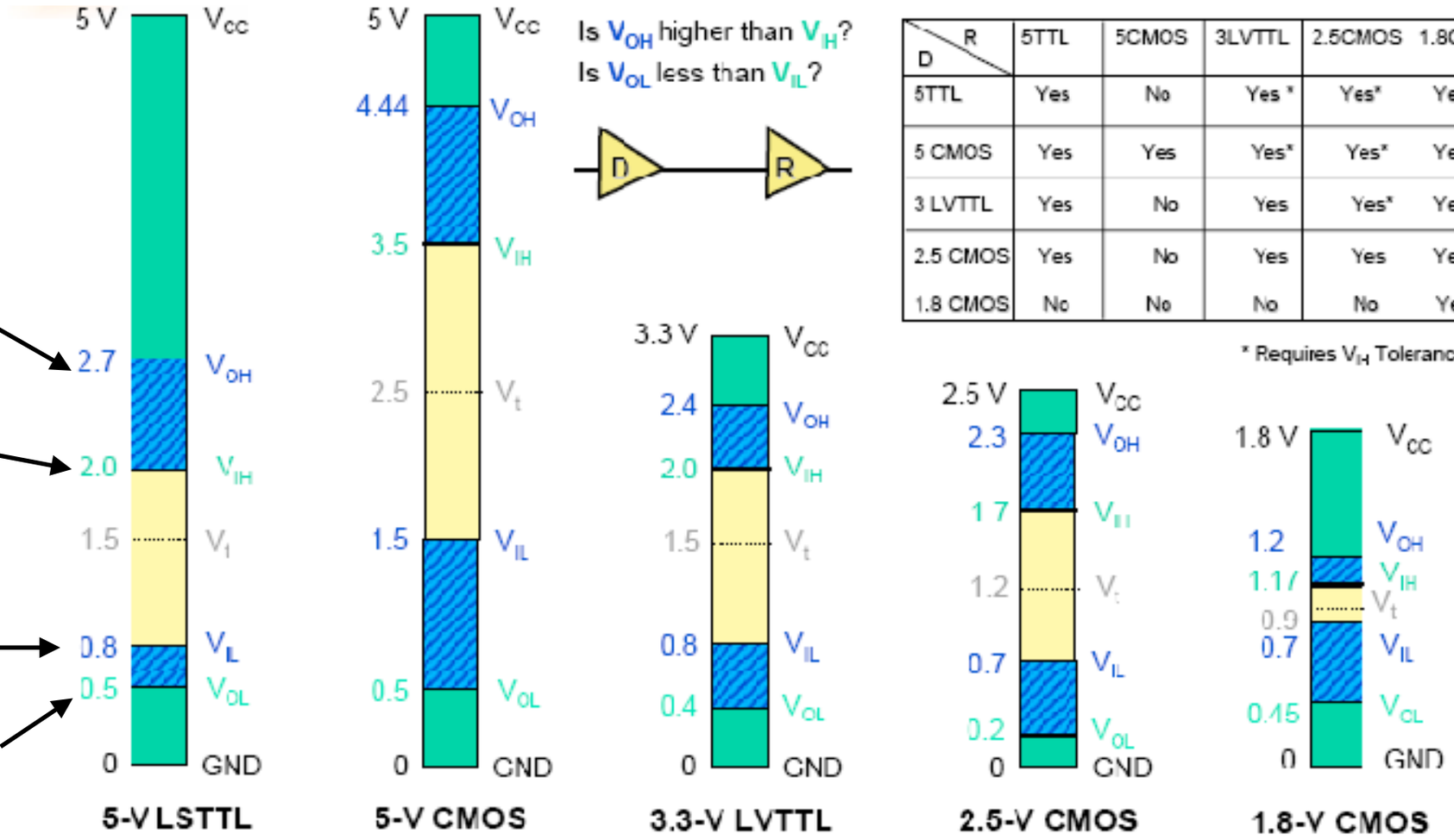
Niveles lógicos

Voltage mínimo de salida como “1” lógico

Voltage mínimo aceptado como “1” lógico

Voltage máximo aceptado como “0" lógico

Voltage máximo de salida como “0” lógico



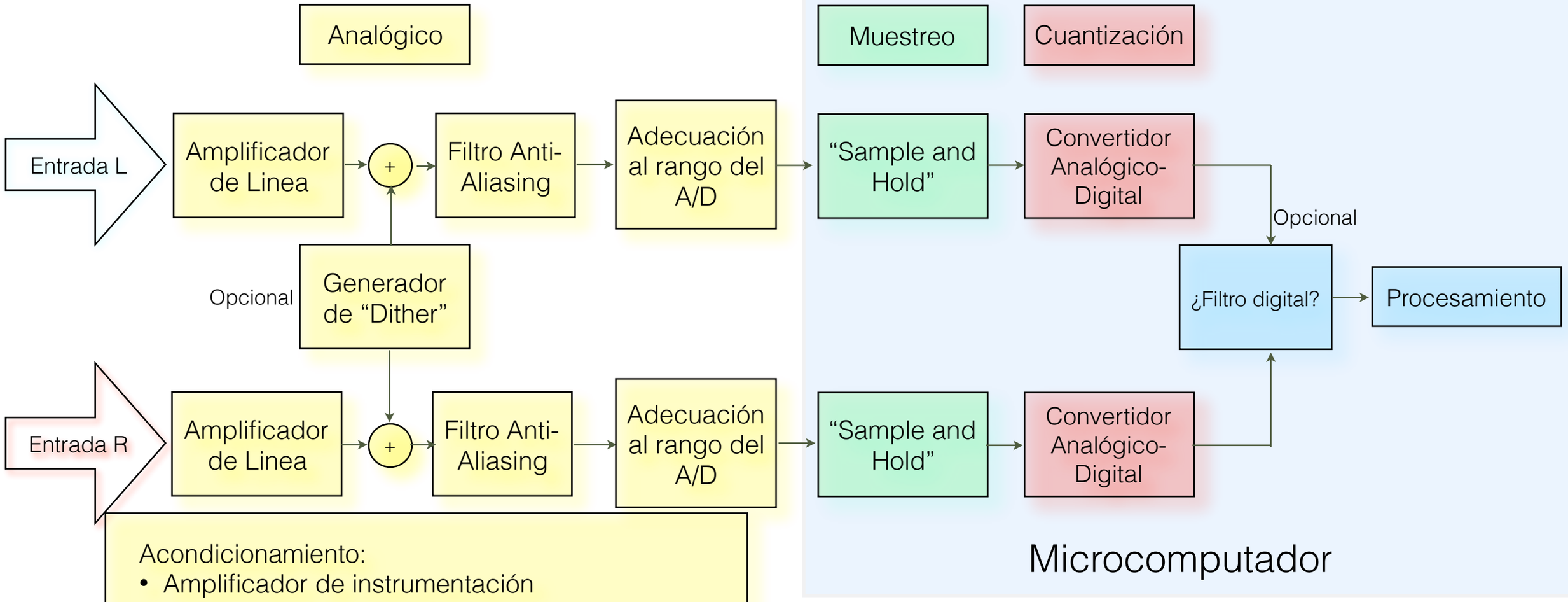
D \ R	5TTL	5CMOS	3LVTTL	2.5CMOS	1.8CMOS
5TTL	Yes	No	Yes *	Yes*	Yes*
5 CMOS	Yes	Yes	Yes*	Yes*	Yes*
3 LVTTL	Yes	No	Yes	Yes*	Yes*
2.5 CMOS	Yes	No	Yes	Yes	Yes*
1.8 CMOS	No	No	No	No	Yes

* Requires V_{IH} Tolerance

Peligro de Quemar

Sin problemas lógicos excepto 1.8-v CMOS

Sistema de Adquisición de señales



Acondicionamiento:

- Amplificador de instrumentación
- "Buffer"
- Ganancia
- Limitador y protector
- Bipolar a unipolar
- Filtros: pasivos, activos, switched capacitor, pasa-bajo, pasa-alto, pasa-banda
- Filtro Antialiasing
- Dual, single, rail to rail supply.

Procesamiento:

- Frecuencia de muestreo
- Rango dinámico: número de bits de cuantificación
- Filtros digitales: moving average, FIR, IIR.
- Mediciones de tiempos entre eventos
- etc

Conversión Analógica-Digital

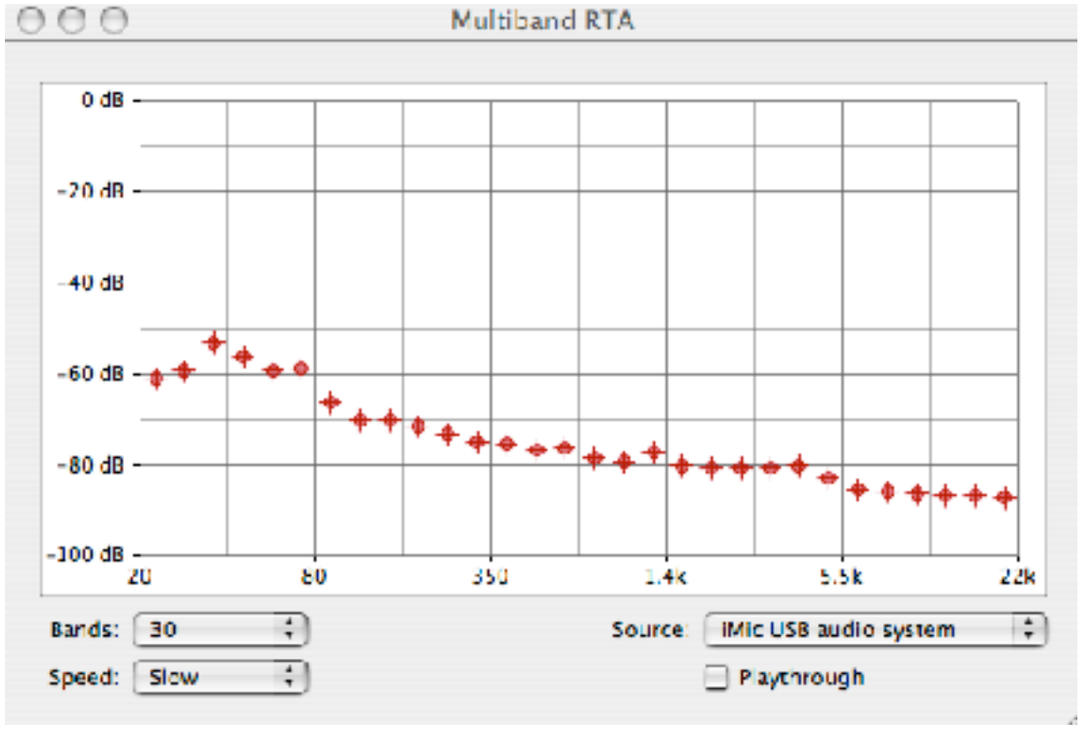
(Muestreo Discreto)

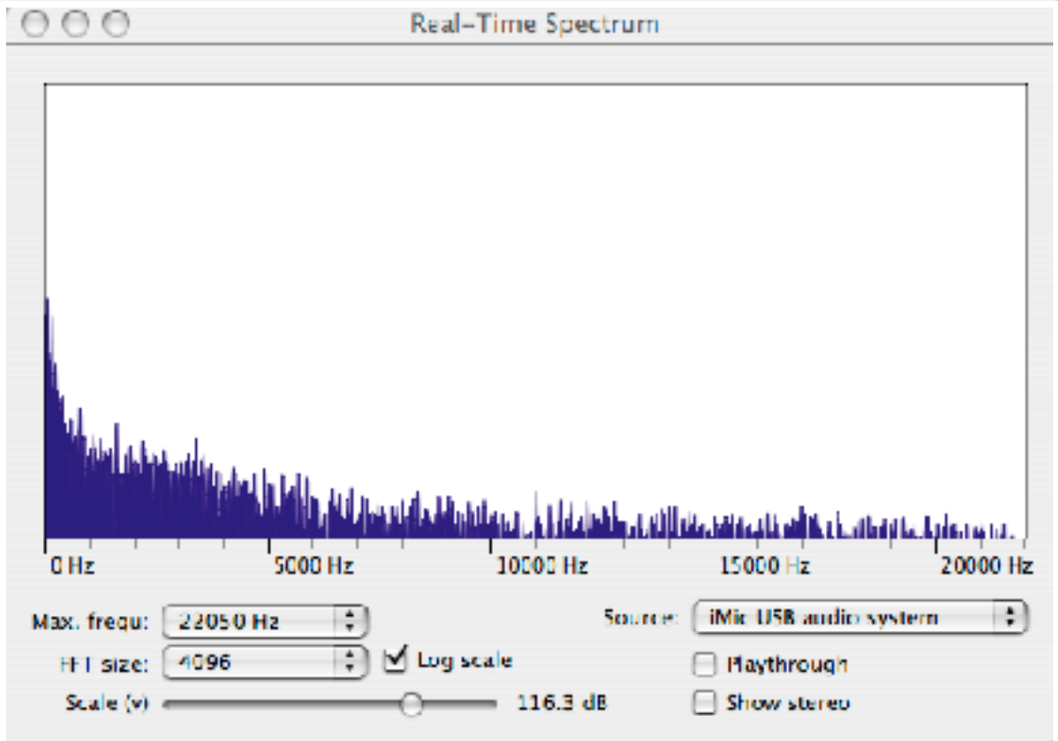
- En amplitud (Número de bits)
  - Rango dinámico
  - Relación señal-ruido
  - Error de cuantificación
    - ¿Dither?
- En tiempo
  - Muestreo Discreto
    - Teorema del muestreo
      - Nyquist
    - Aliasing

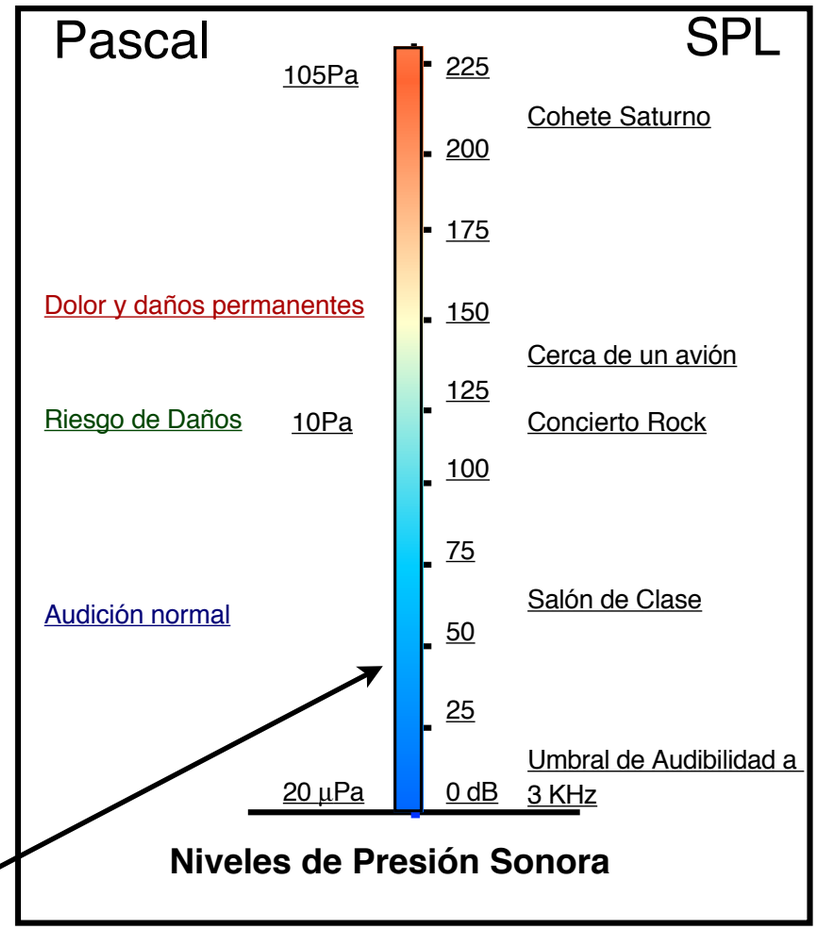
En amplitud

Rango dinámico
Relación Señal-ruido

Ejemplo: Sonido

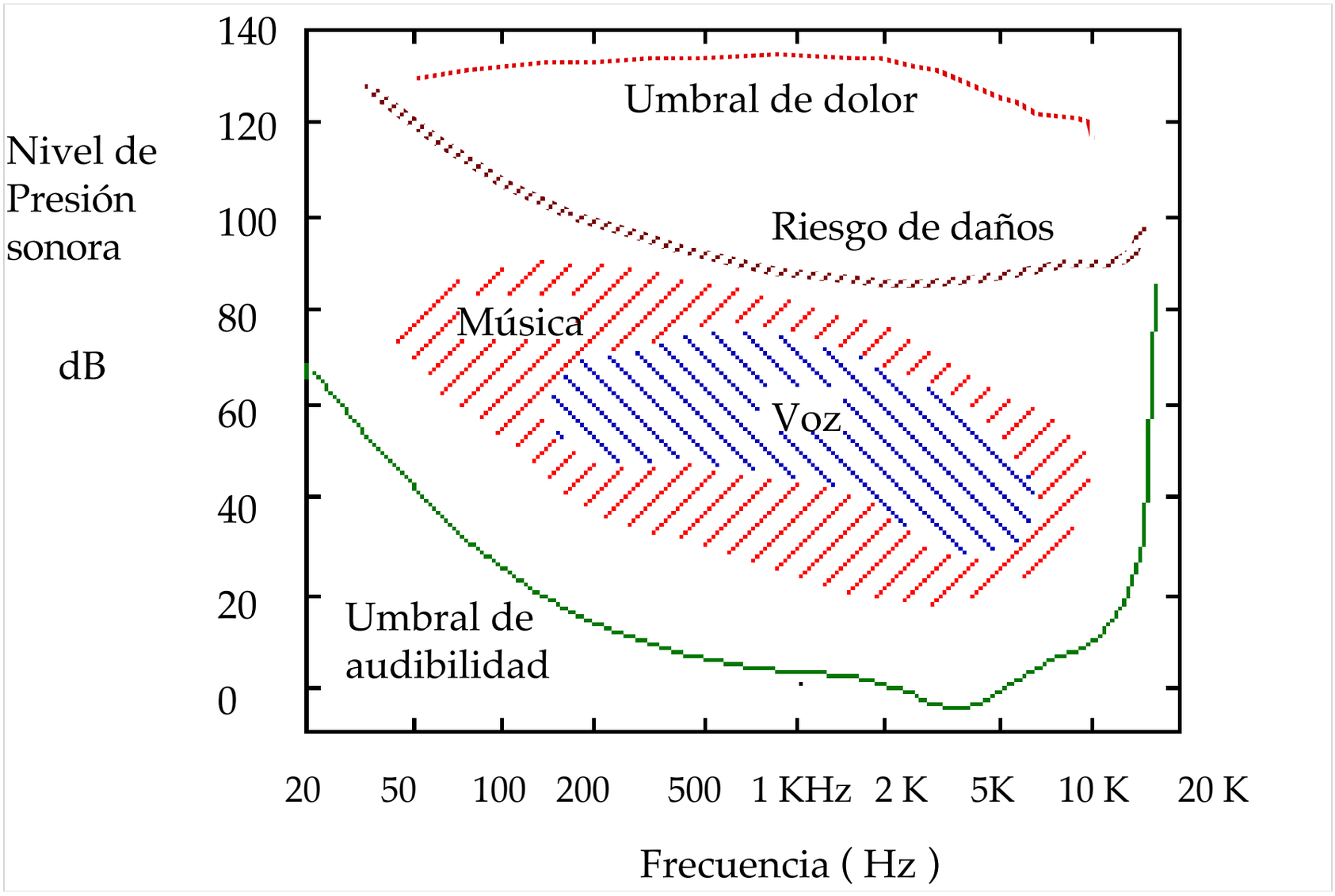






Nivel de ruido de fondo de un ambiente “silencioso” casero

Rango de operación del Sistema Auditivo



Rango dinámico

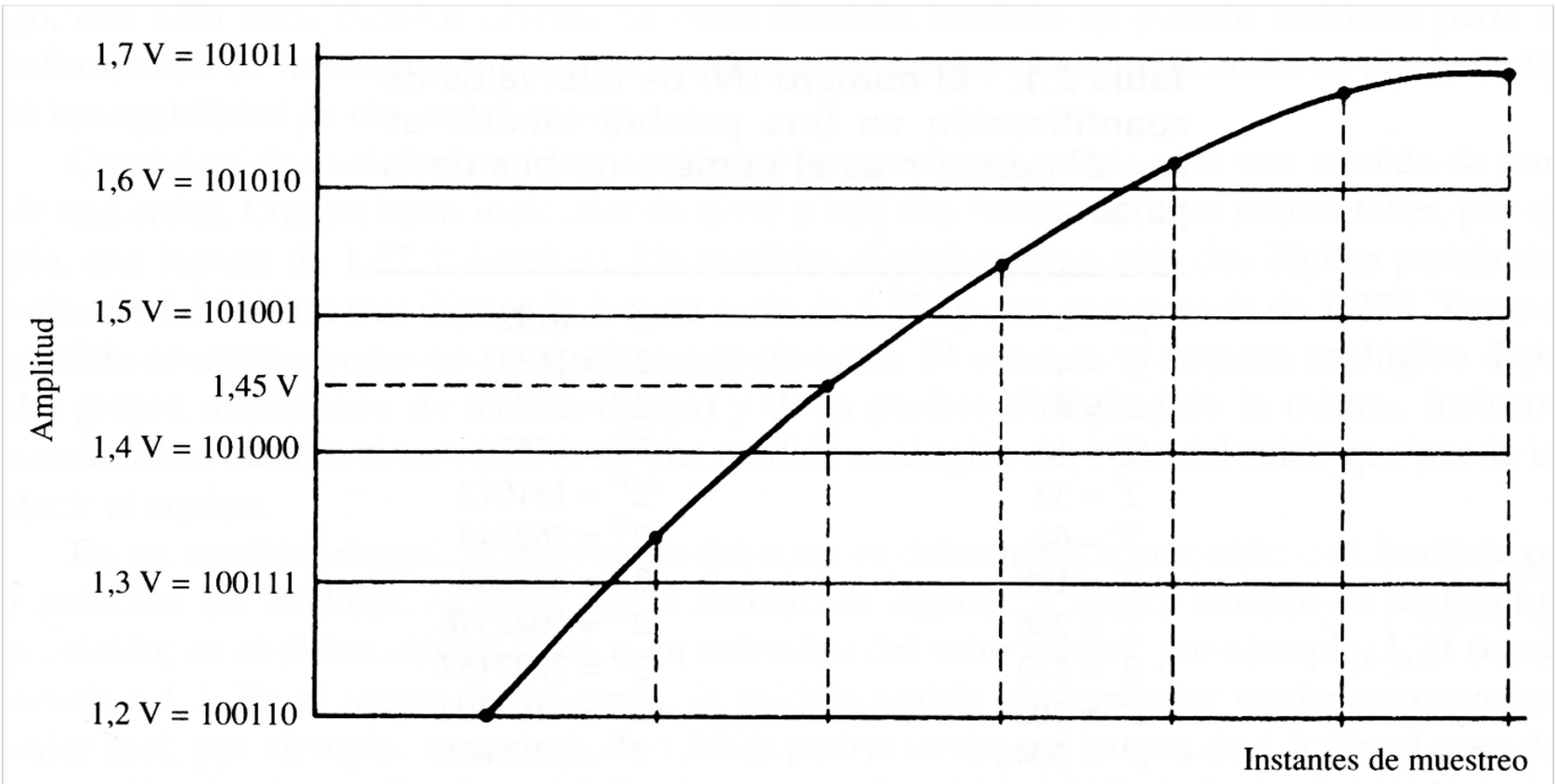
En el caso de Audio debemos cubrir un rango entre 25 dB y 120 dB = 95 dB en un ambiente casero.

y unos 15 dB más en los estudios

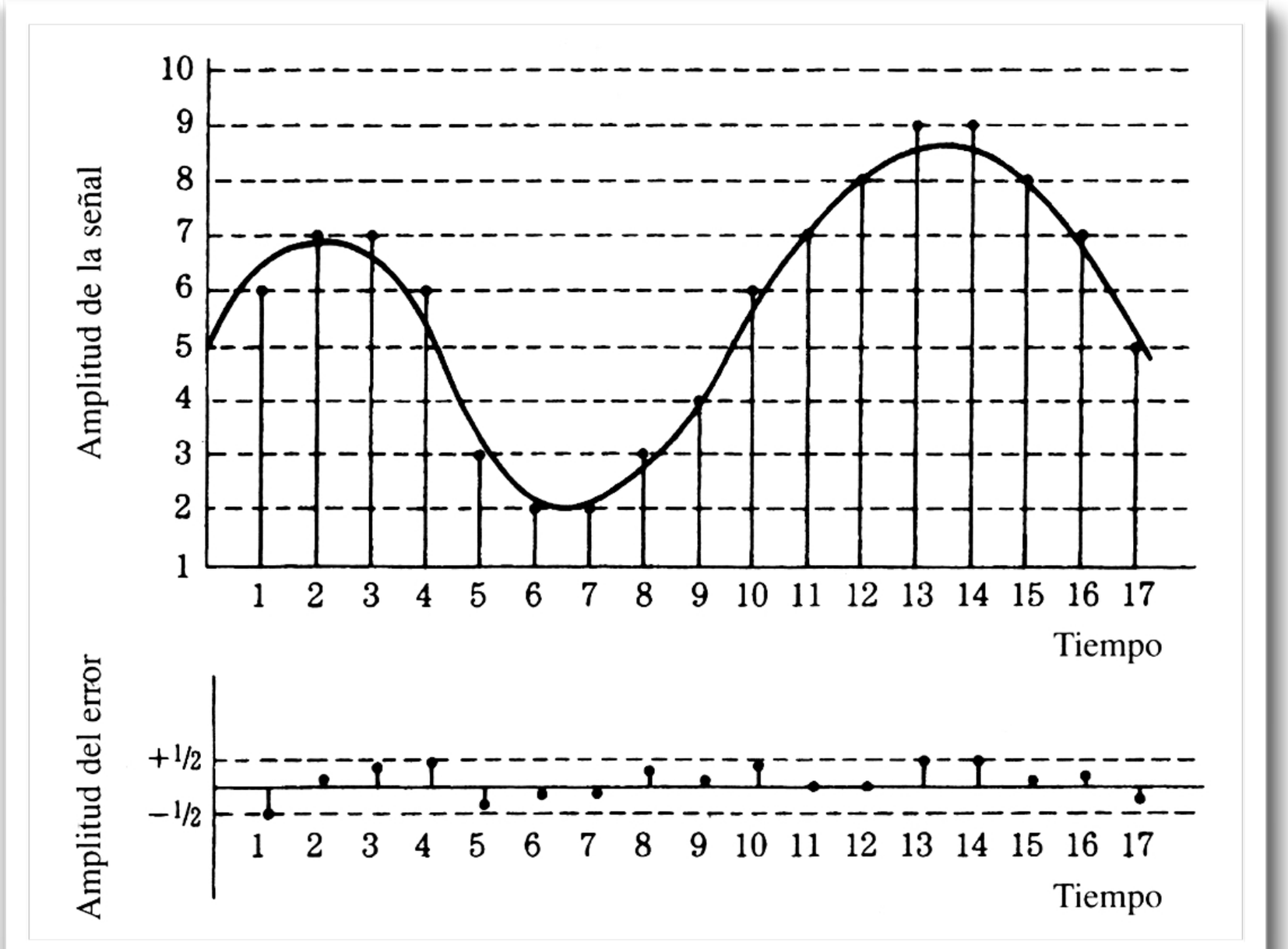
Cuantización

¿Cuantos bits hacen falta para cuantizar una señal?

El error de cuantización esta limitado a 1/2 del bit menos significativo

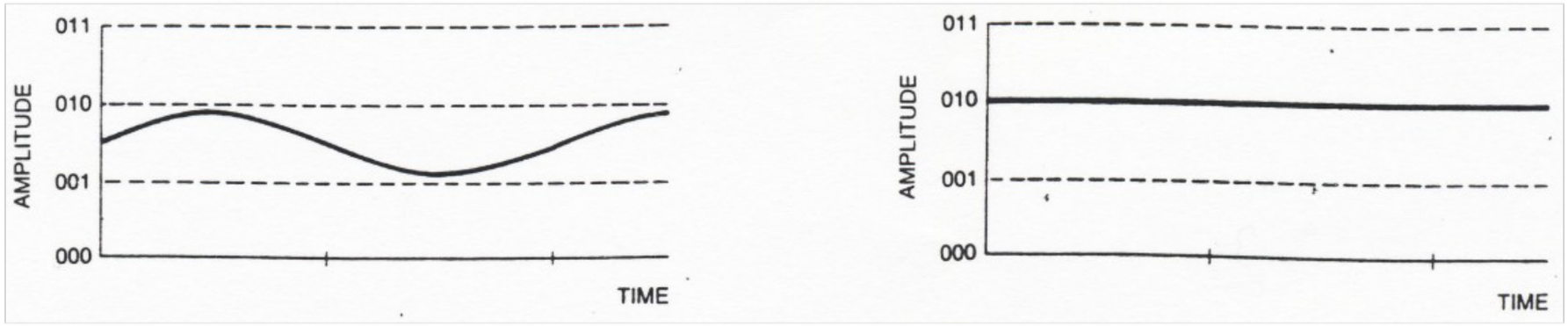


Señal de error

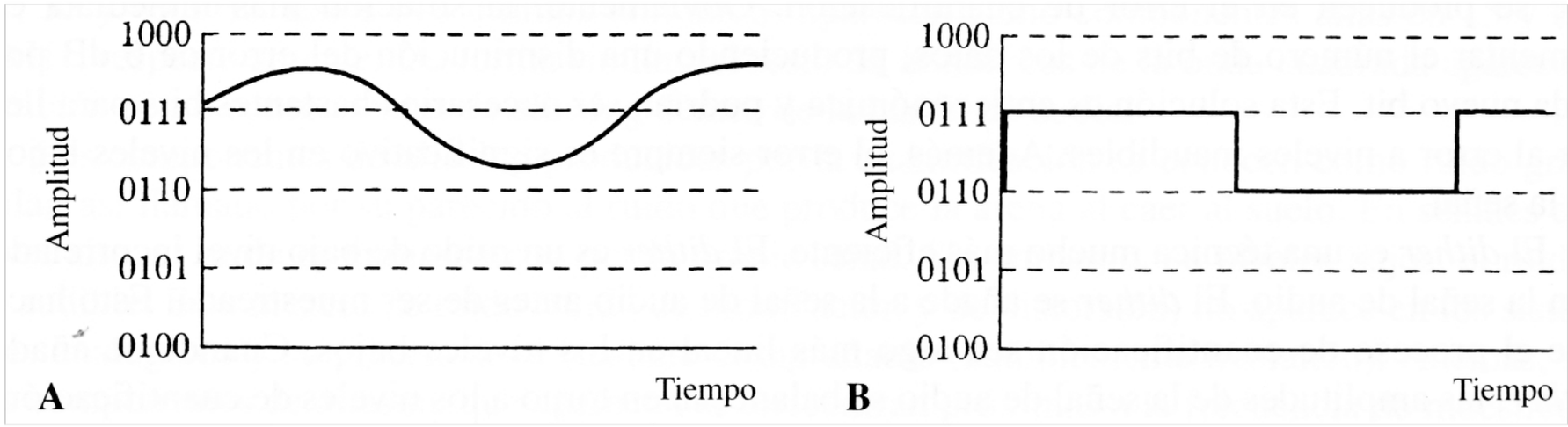


¿Que sucede si la señal es menor que el escalón de cuantización

No se escucha

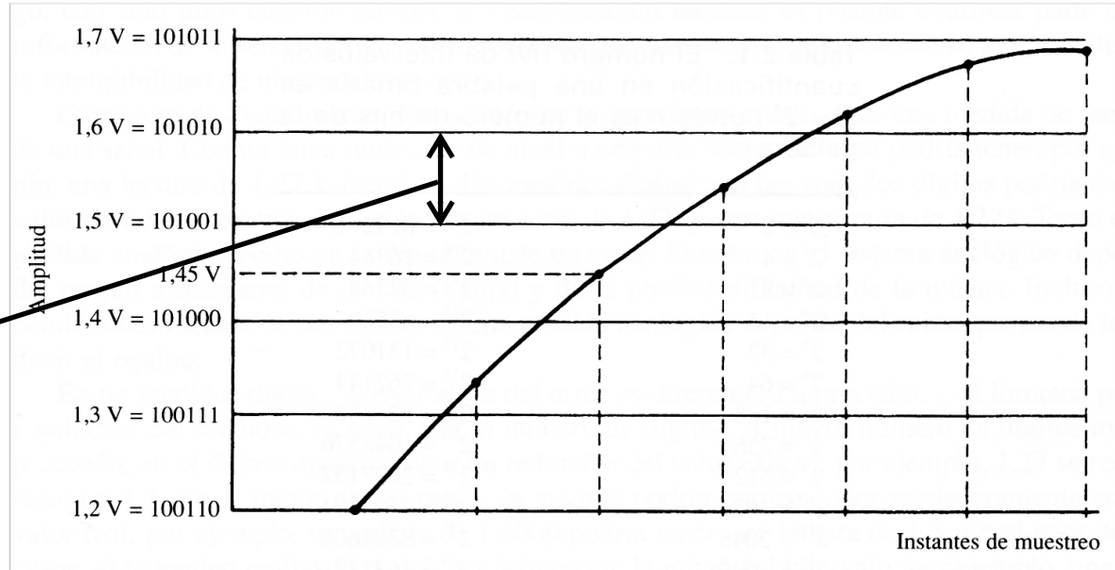


Se escucha con distorsión



Error de cuantización

Sea Q el tamaño del escalón de cuantización



Sea **n** el número de bits y **N** el número de escalones de cuantificación → $N=2^n$

Si la **codificación es bipolar** tenemos que el valor máximo será $\pm Q2^{n-1}$

Con un A/D con 3 v_{max} de entrada con:

8 bits:	$Q=V_{pico}/128$	=23 mv
16 bits	$Q=V_{pico}/32768$	= 91.5 μv
24 bits	$Q=V_{pico}/8288608$	= 0.357 μv

Valor rms de una sinusoide en función del número de escalones que ocupa

$$S_{rms} = \frac{Q2^n / 2}{\sqrt{2}} = \frac{Q2^{n-1}}{\sqrt{2}}$$

Supongamos que el error de cuantización se distribuye uniformemente entre +Q/2 y -Q/2 con una amplitud de 1/Q (espectro plano)

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

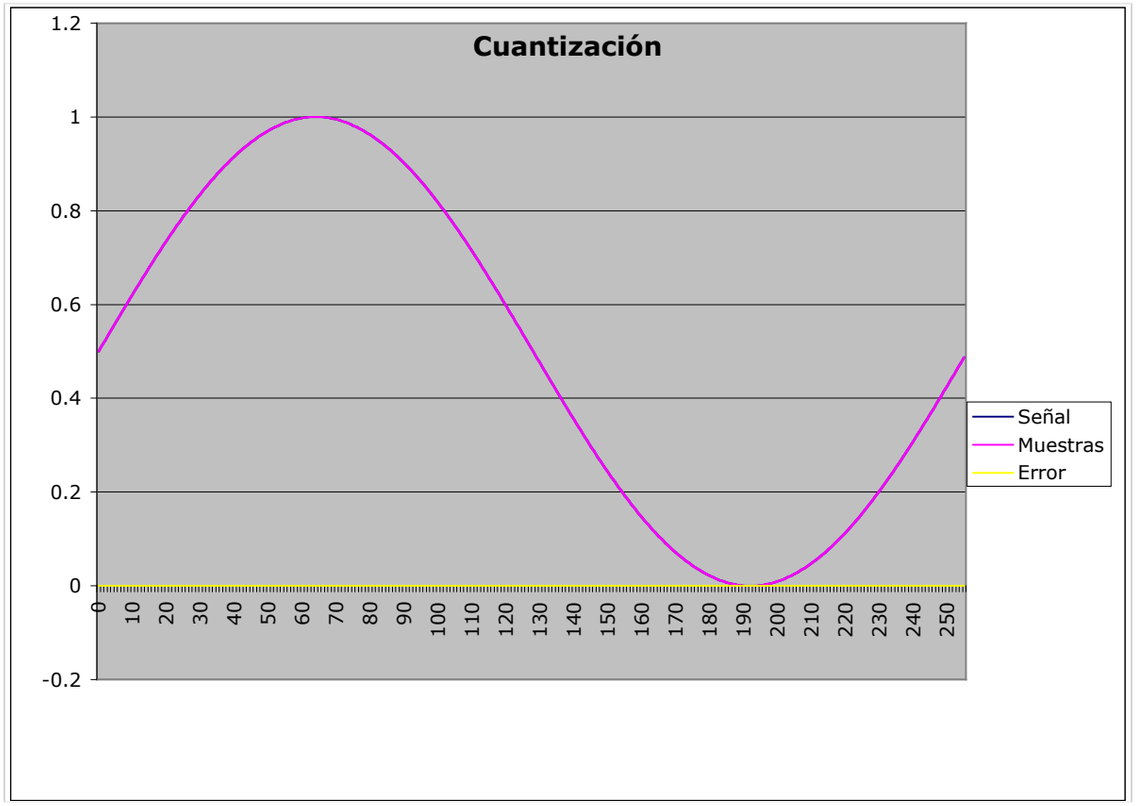
Relación Señal-Error

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

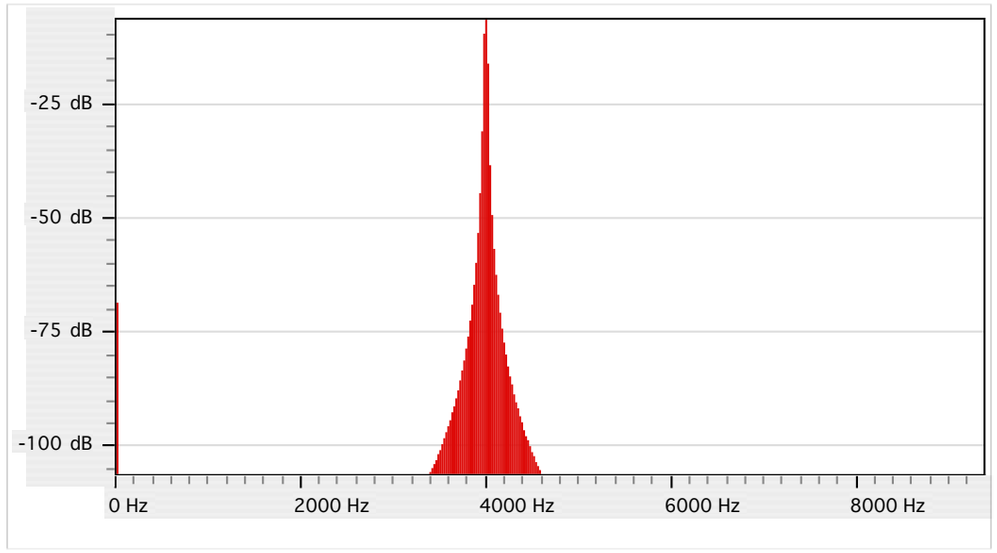
$$\frac{S}{E}(\text{dB}) = 10\log\left[\frac{3}{2}(2^{2n})\right] = 20\log\left[\left(\frac{3}{2}\right)^{1/2}\left(2^n\right)\right]$$

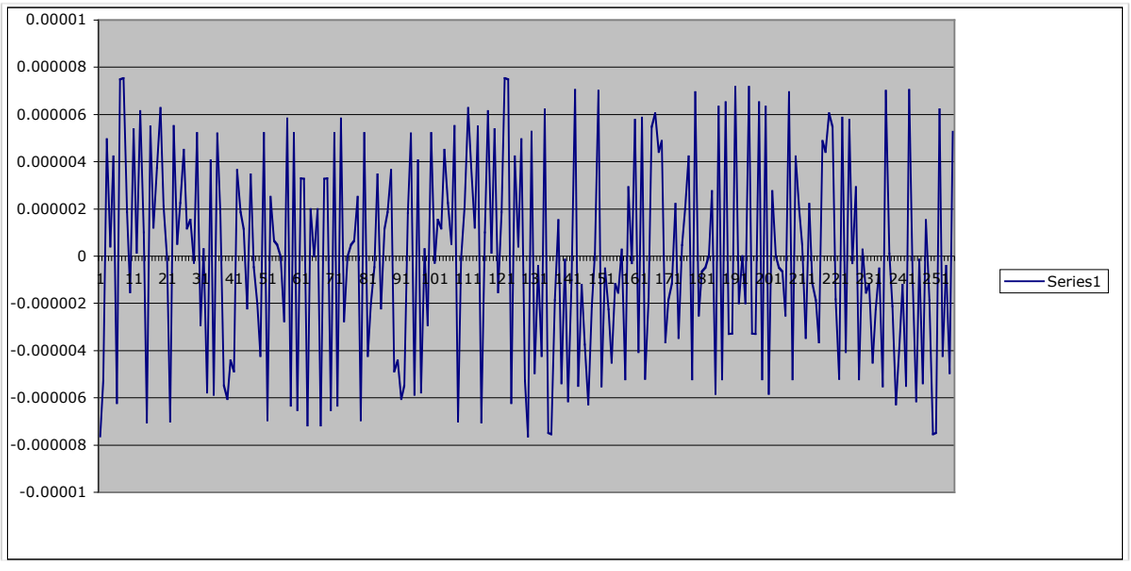
$$= 6.02n + 1.76$$

Para 8 bits corresponde un S/N de 49.92 dB
Para 16 bits corresponde un S/N de 98.08 dB
Para 24 bits corresponde un S/N de 146.24 dB

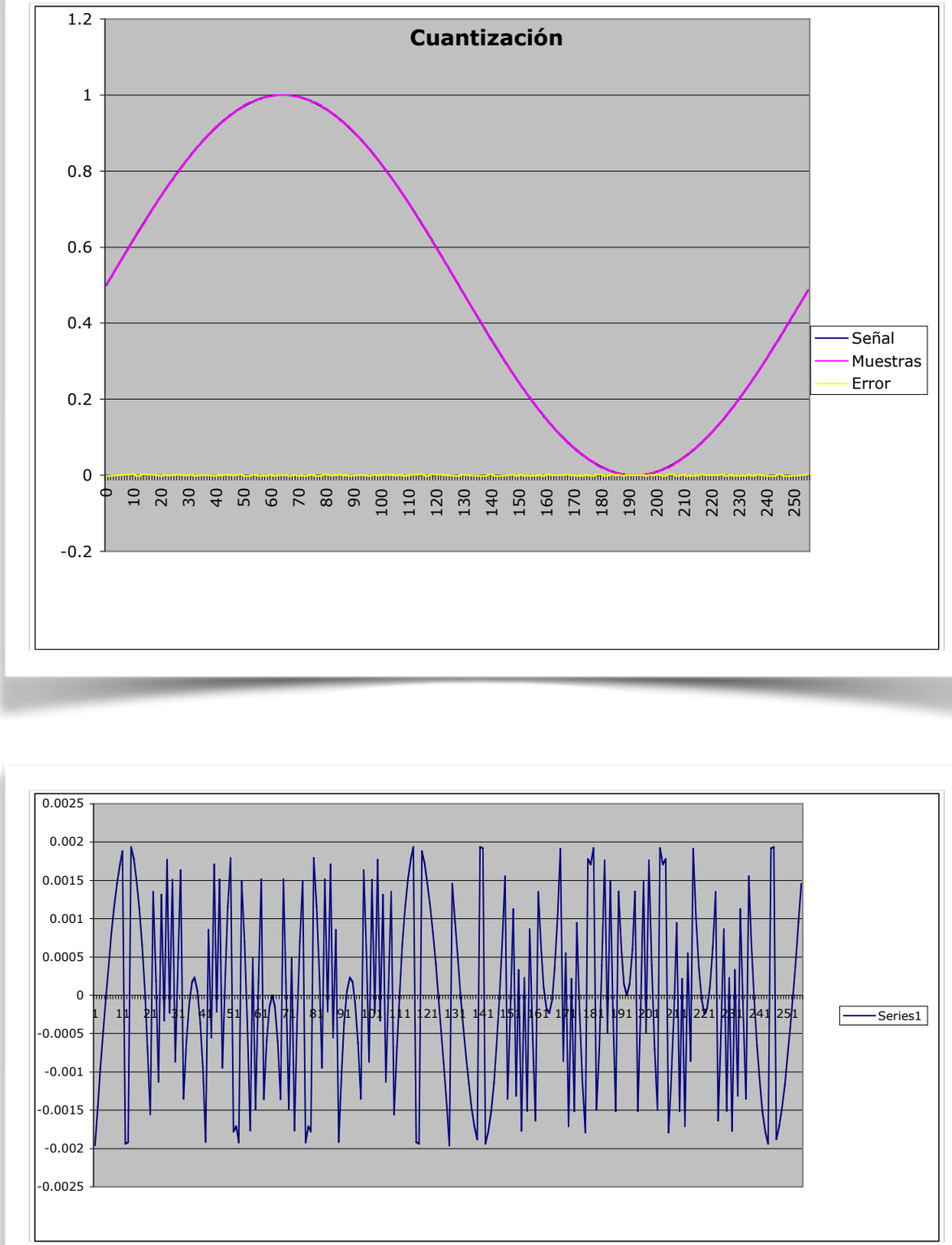


16 bits

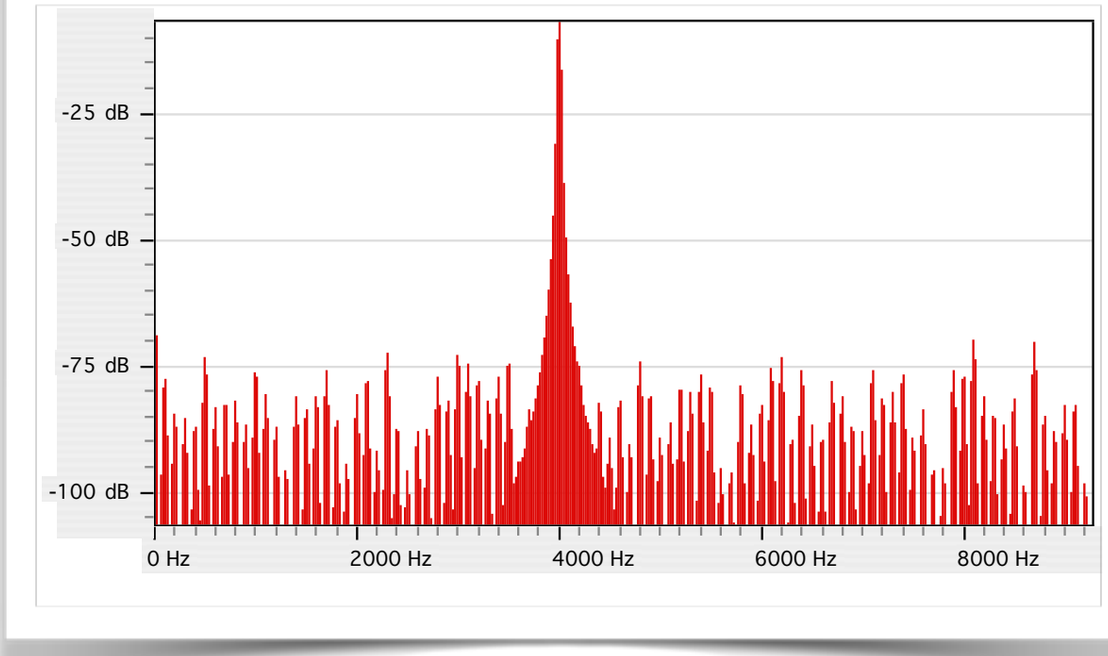




Error

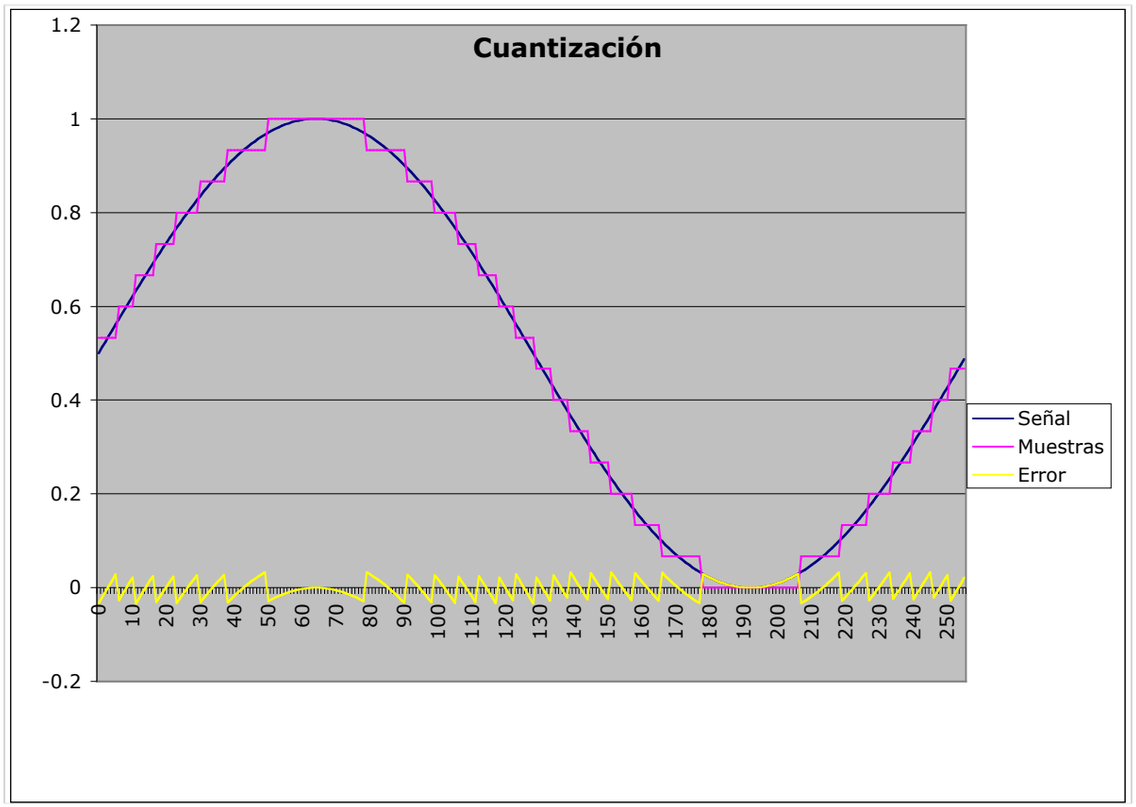

Cuantización
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250
Señal
Muestras
Error

8 bits

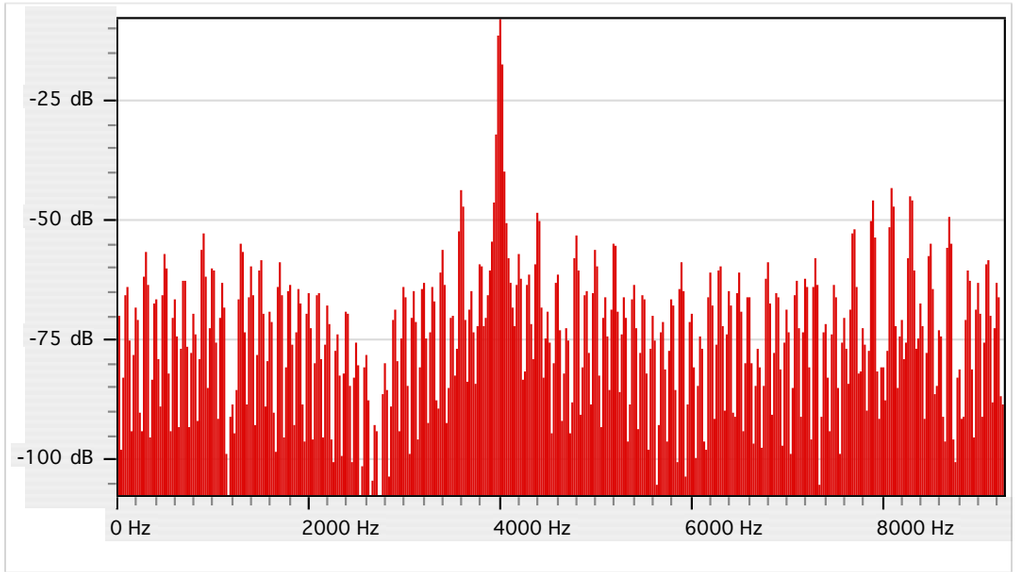

-25 dB
-50 dB
-75 dB
-100 dB
0 Hz
2000 Hz
4000 Hz
6000 Hz
8000 Hz

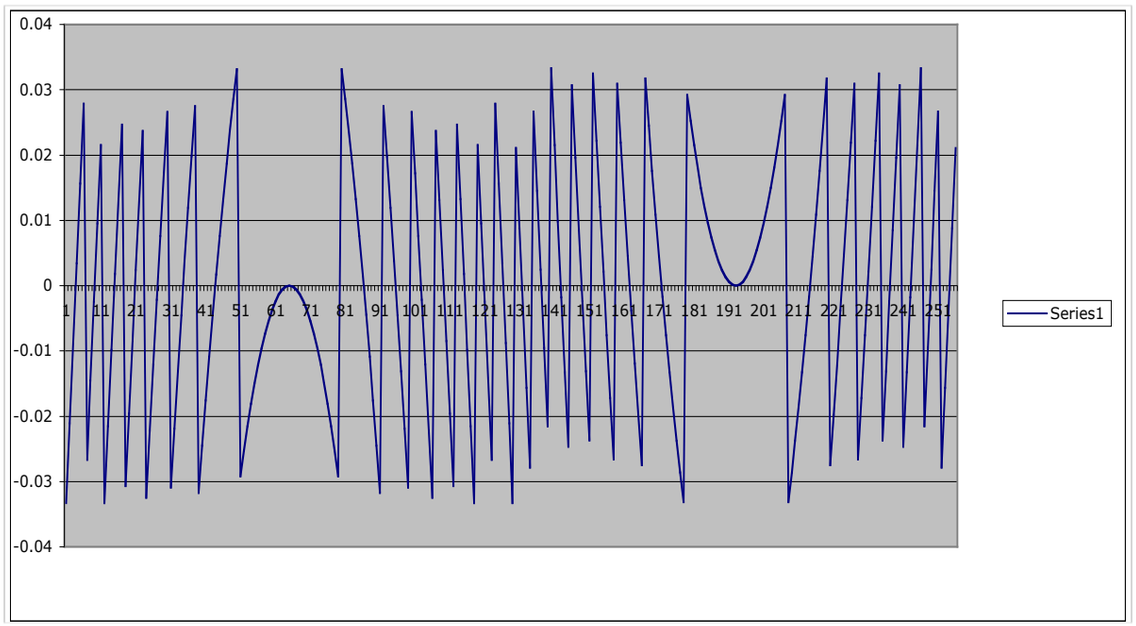
0.0025
0.002
0.0015
0.001
0.0005
0
-0.0005
-0.001
-0.0015
-0.002
-0.0025
1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251
Series1

Error


Cuantización
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
Señal
Muestras
Error

4 bits


-25 dB
-50 dB
-75 dB
-100 dB
0 Hz
2000 Hz
4000 Hz
6000 Hz
8000 Hz


0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231
241
251
Series1

Error

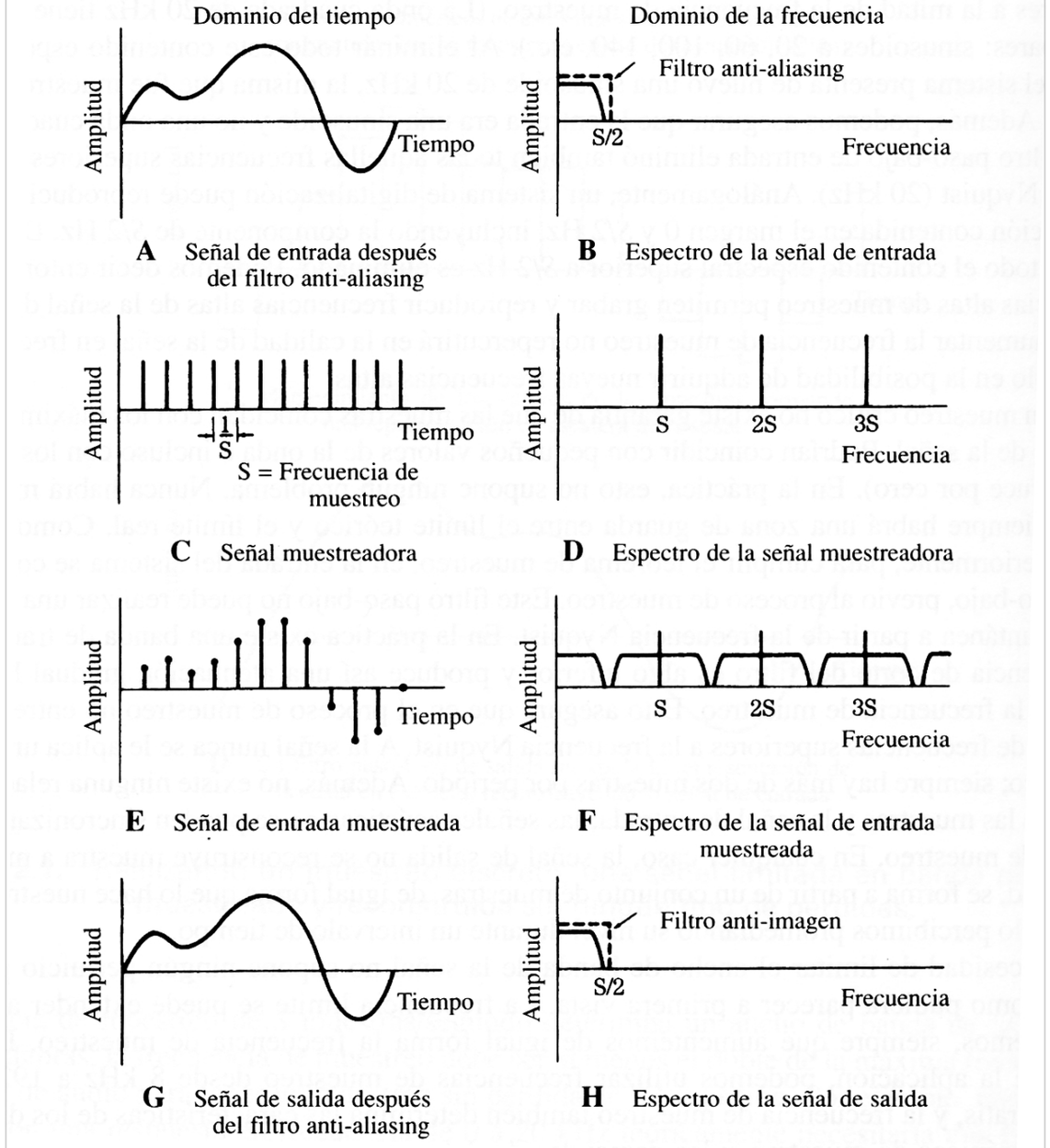
Muestreo

Fourier

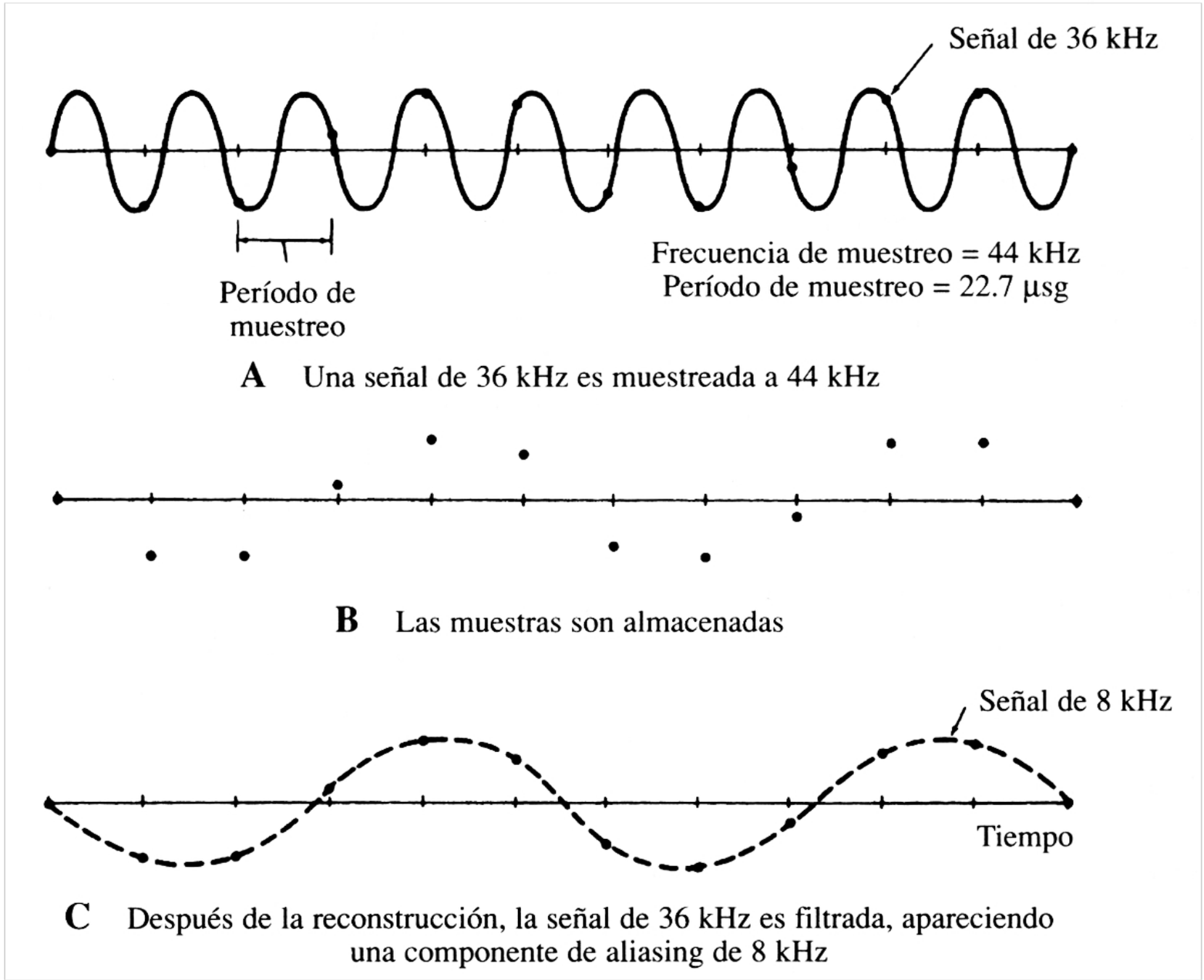
Harry Nyquist

Claude Shannon

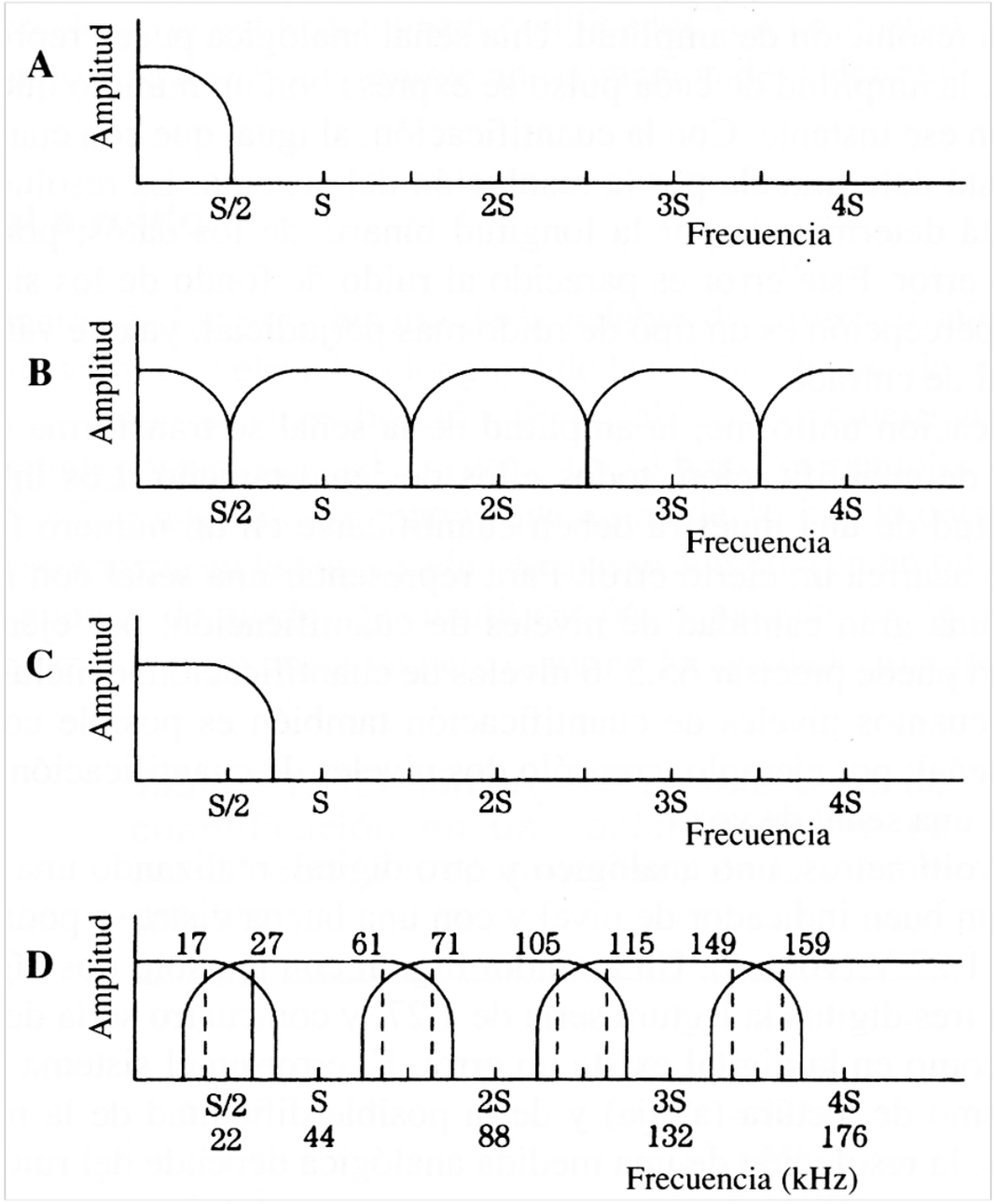
Nyquist-Shannon



“Aliasing”

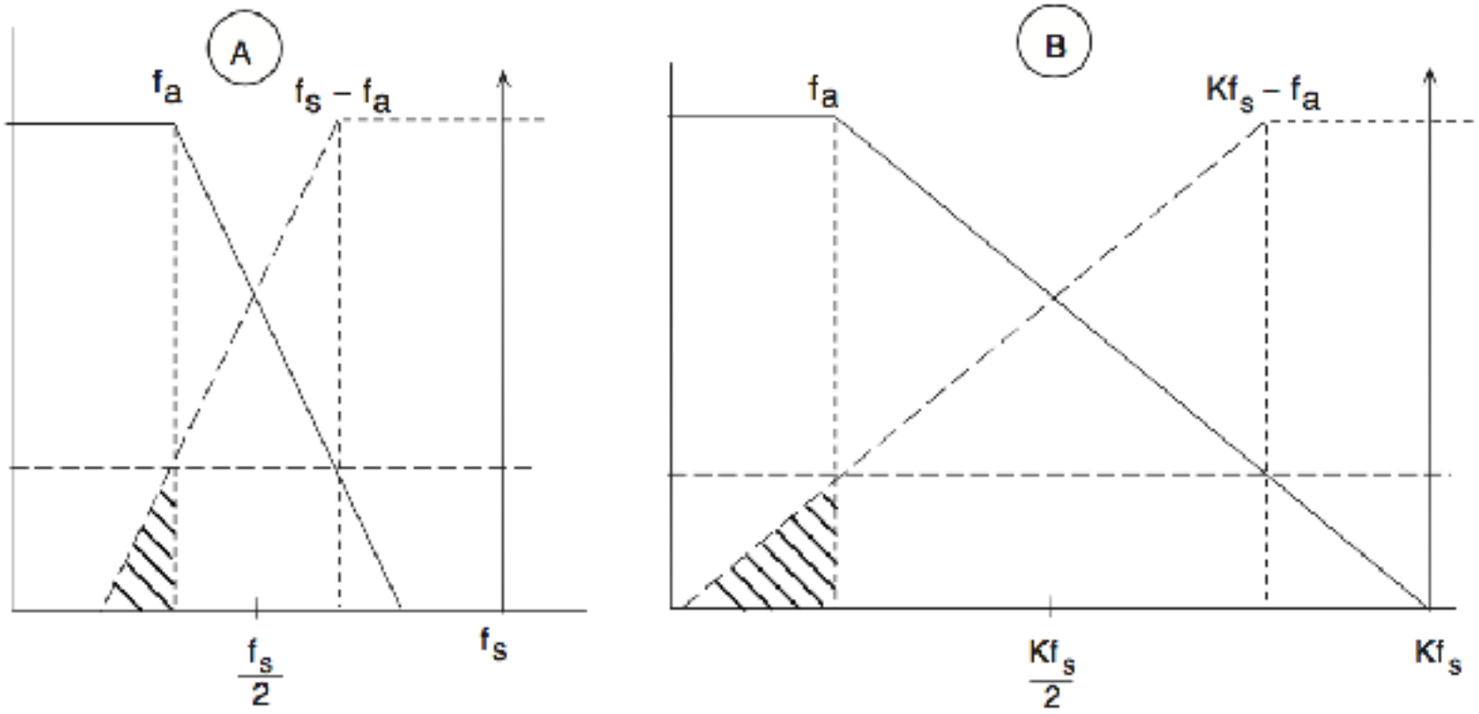

Señal de 36 kHz
Período de muestreo
Frecuencia de muestreo = 44 kHz
Período de muestreo = 22.7 µsg
A Una señal de 36 kHz es muestreada a 44 kHz
B Las muestras son almacenadas
Señal de 8 kHz
Tiempo
C Después de la reconstrucción, la señal de 36 kHz es filtrada, apareciendo una componente de aliasing de 8 kHz

“Aliasing”

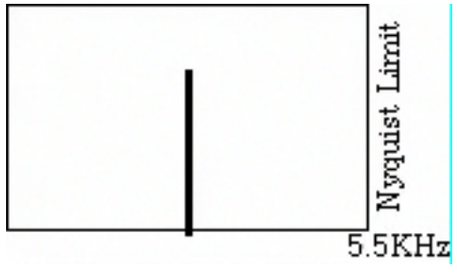

A
Amplitud
S/2
S
2S
3S
4S
Frecuencia
B
Amplitud
S/2
S
2S
3S
4S
Frecuencia
C
Amplitud
S/2
S
2S
3S
4S
Frecuencia
D
Amplitud
17
27
61
71
105
115
149
159
S/2
22
S
44
2S
88
3S
132
4S
176
Frecuencia (kHz)

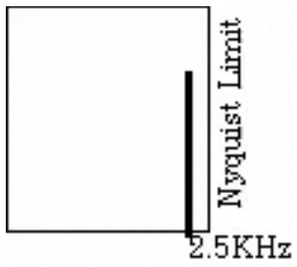
Filtro anti-aliasing

Para evitar el aliasing **NO** debe haber señales por encima de $f_m/2$

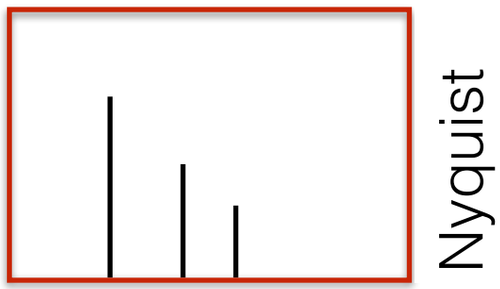


“Aliasing”

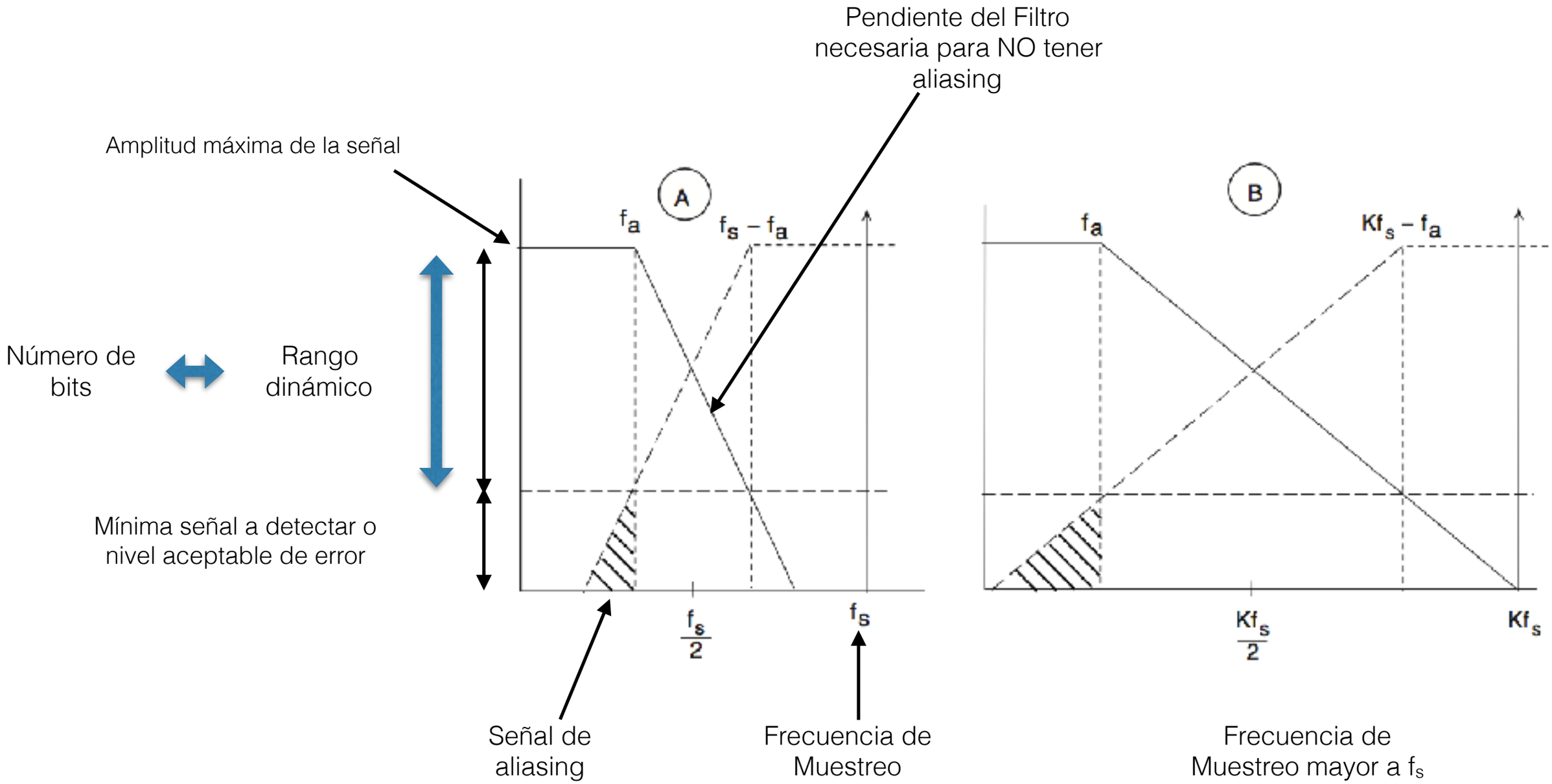




“Aliasing”



Relación entre Frecuencia, Muestreo, Filtro antialiasing y Rango Dinámico



La frecuencia de muestreo y la pendiente del filtro antialiasing debe elegirse de manera tal que, cualquier señal de aliasing quede por debajo del umbral de error (que viene dado por el número de bits del A/D).

Ejemplos: CD y las grabaciones digitales en estudio

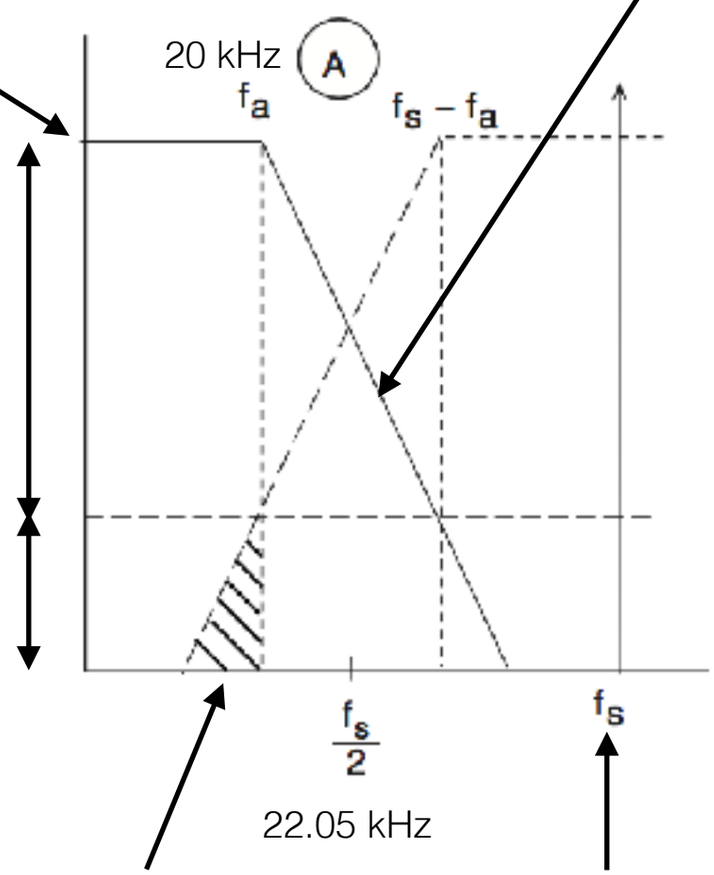
Pendiente del Filtro necesaria para NO tener aliasing

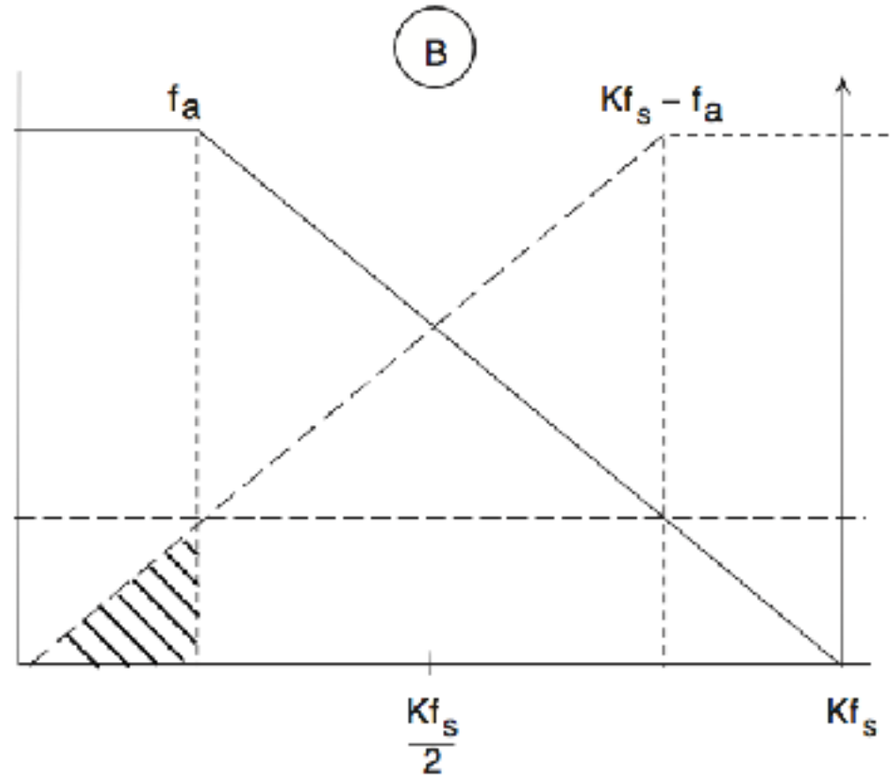
Amplitud máxima de la señal 120db

Número de bits: 16

Rango dinámico 98dB

Mínima señal a detectar o nivel aceptable de error 20 dB

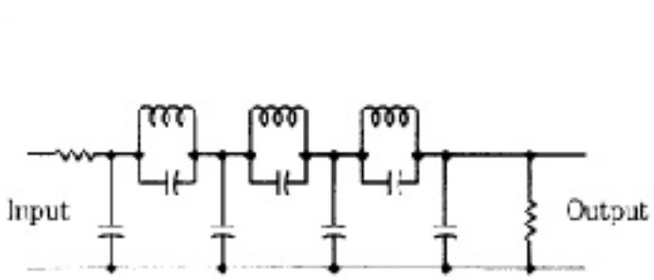


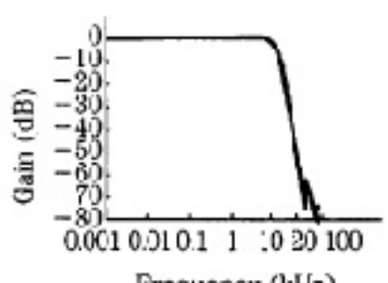


Señal de aliasing

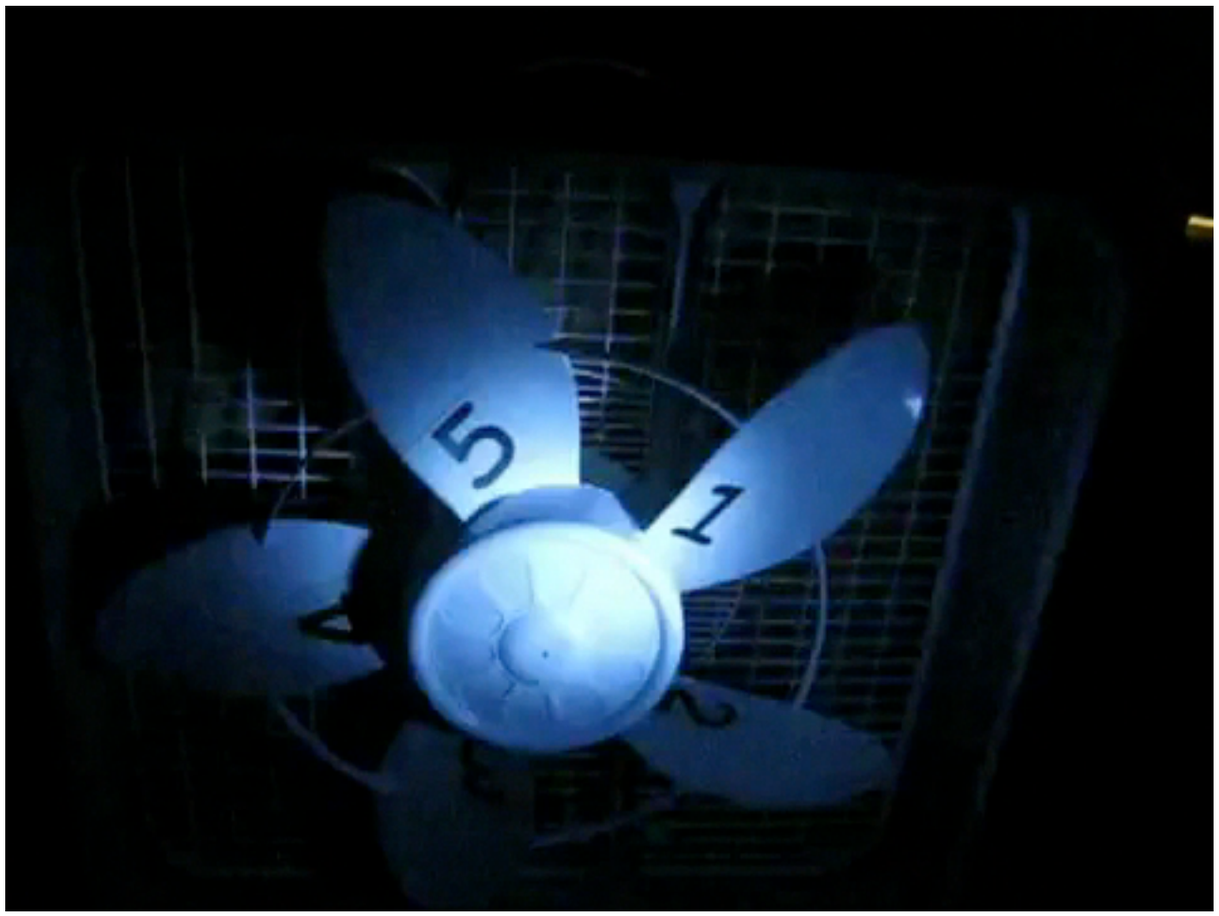
Frecuencia de Muestreo 44.1 kHz

Frecuencia de Muestreo mayor a f_s

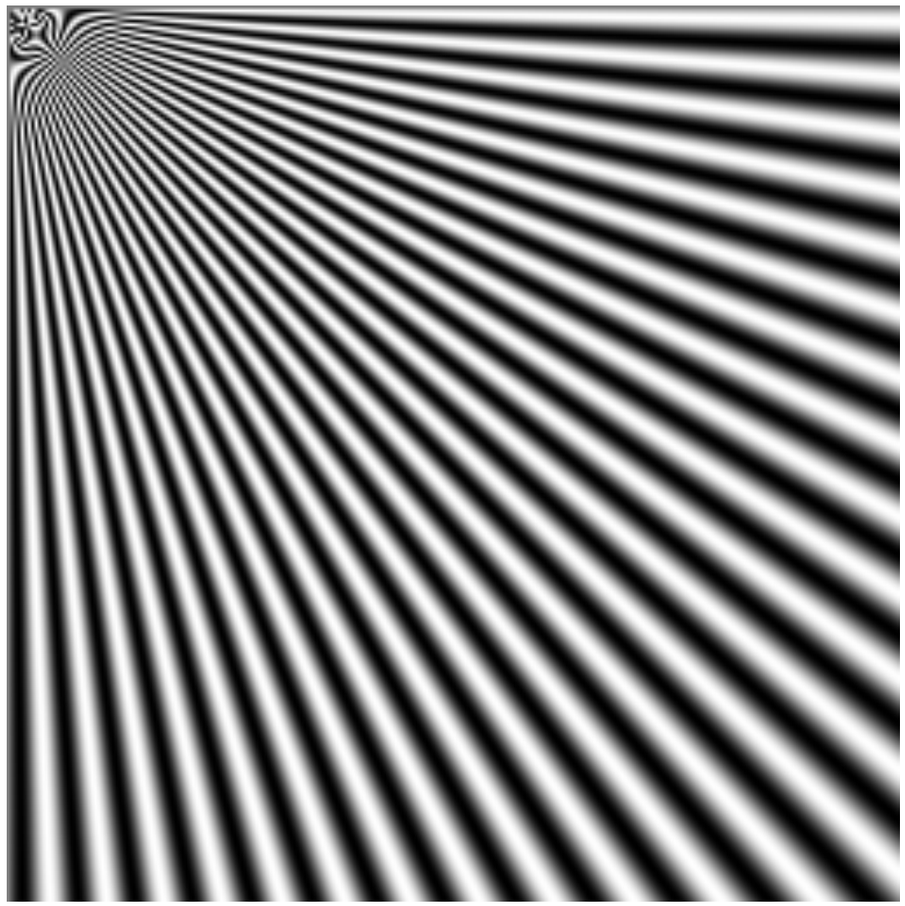




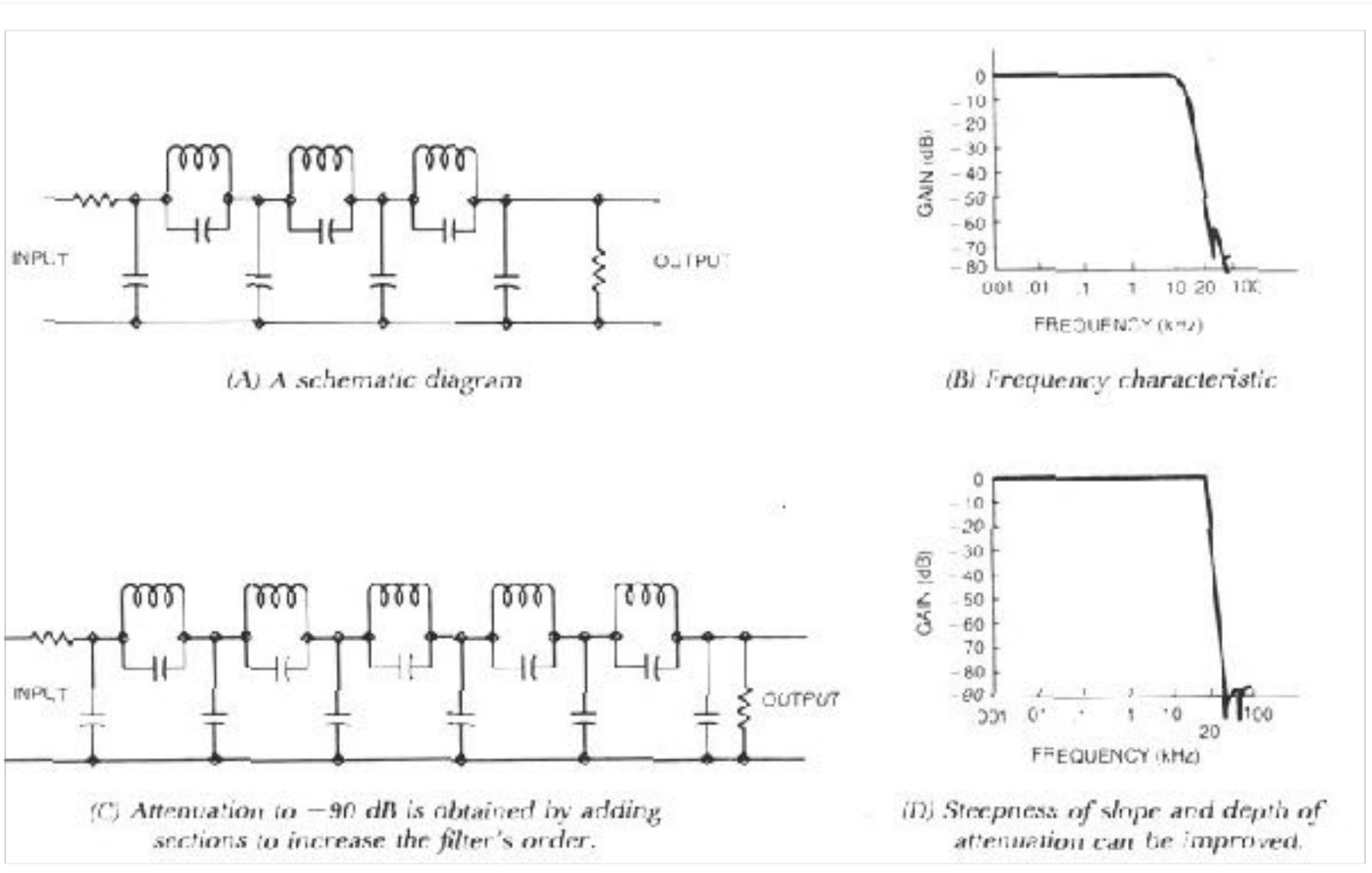
“Aliasing“ en Video



“Aliasing“ en Imágenes



Filtros de Alta Pendiente



(A) A schematic diagram

(B) Frequency characteristic

(C) Attenuation to −90 dB is obtained by adding sections to increase the filter's order.

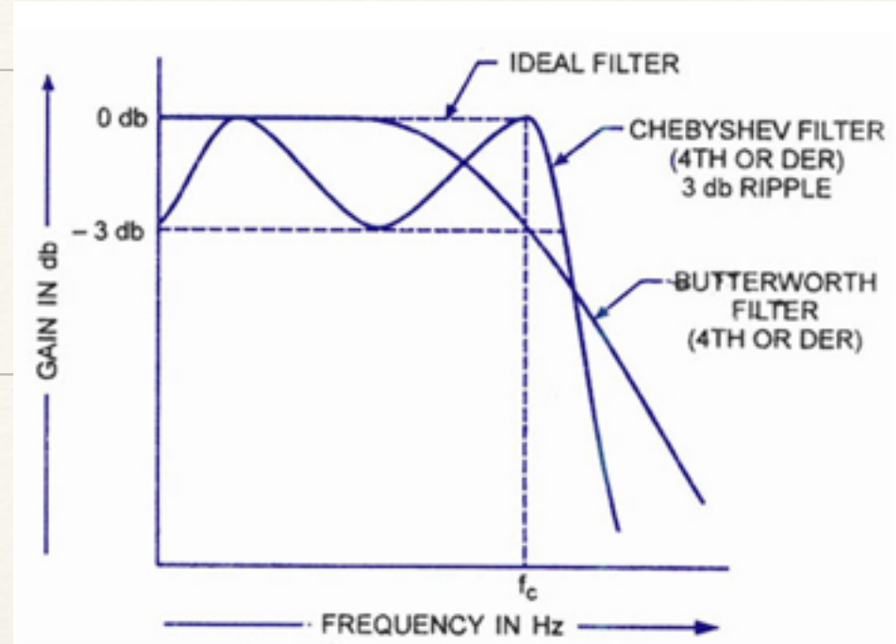
(D) Steepness of slope and depth of attenuation can be improved.

¡Filtros de orden 33!

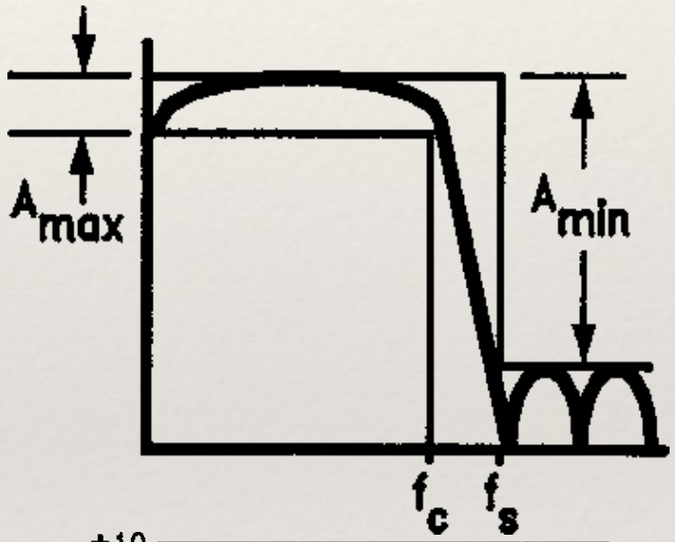
Filtros Analógicos

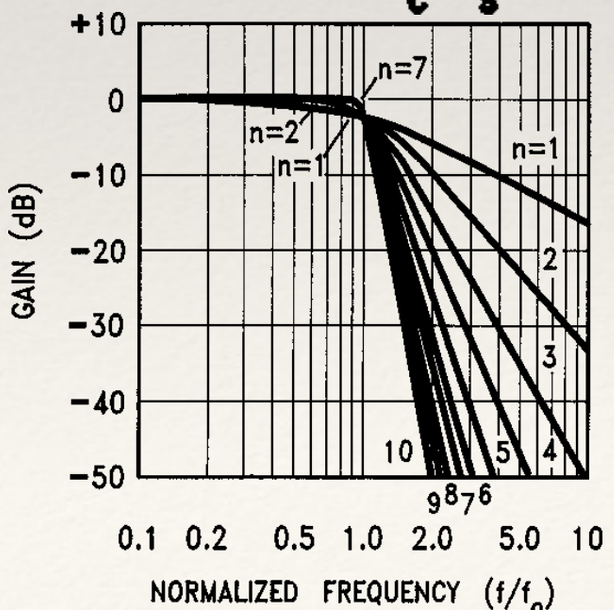
Requerimientos

- Respuesta en frecuencia: Butterworth, Chebyshev, etc.
- Tipo de Filtro: Pasa-bajo, pasa-alto, pasa-banda, elimina-banda, etc.
- Orden del filtro
- Frecuencia de corte o frecuencia central.
- Atenuación en la frecuencia de corte.
- Rizado en la banda pasante
- Rizado en la banda atenuante

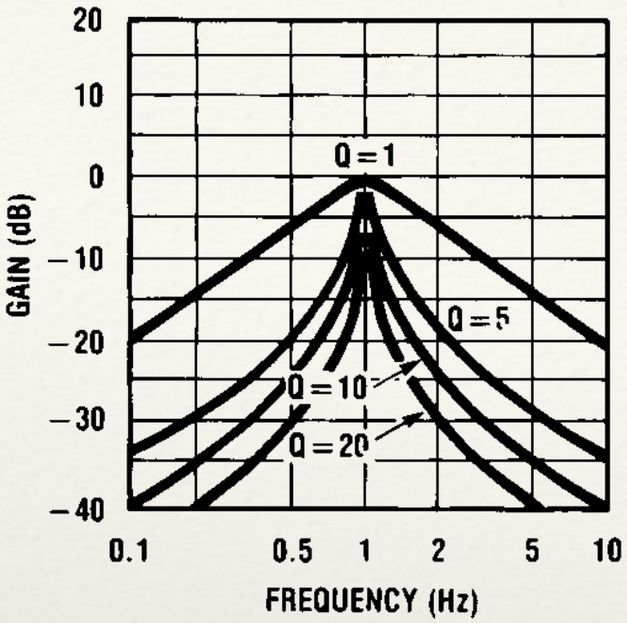


Chebyshev and Butterworth Filter

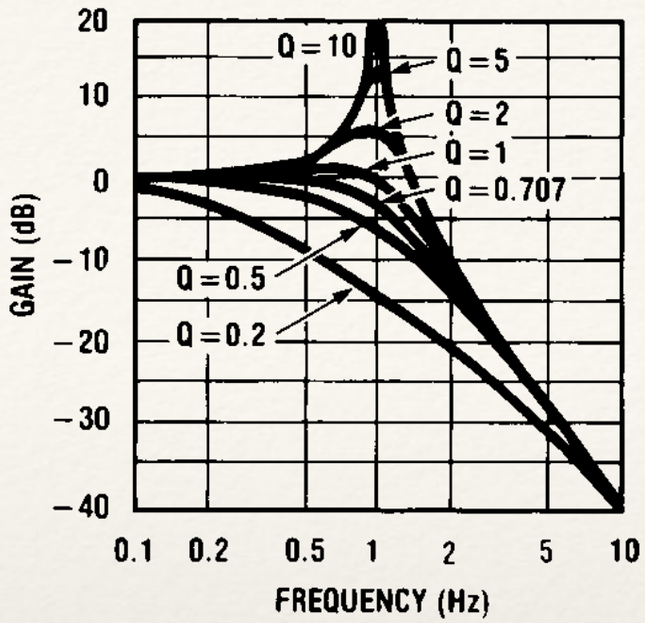




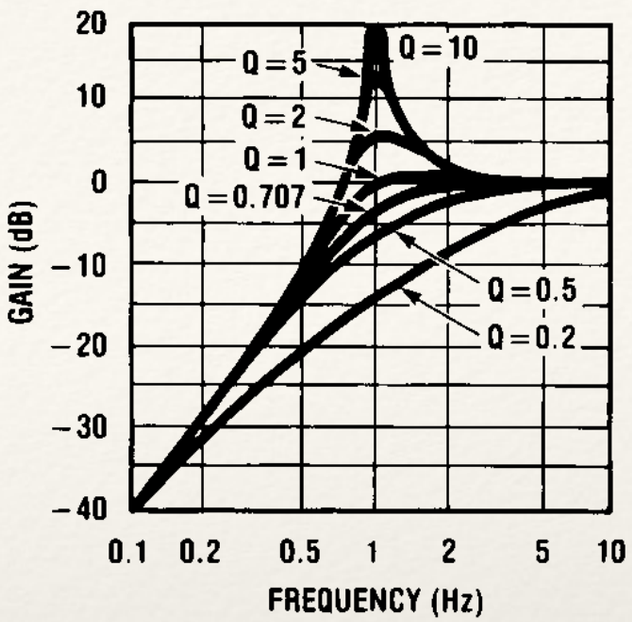
Q



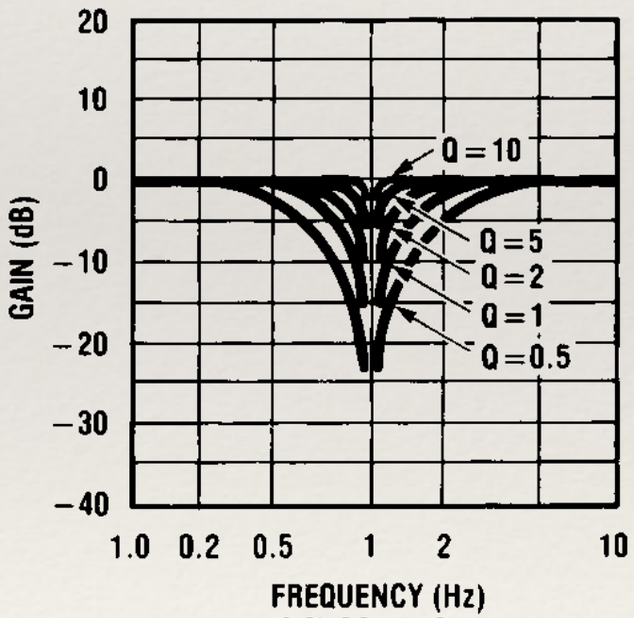
(a) Bandpass



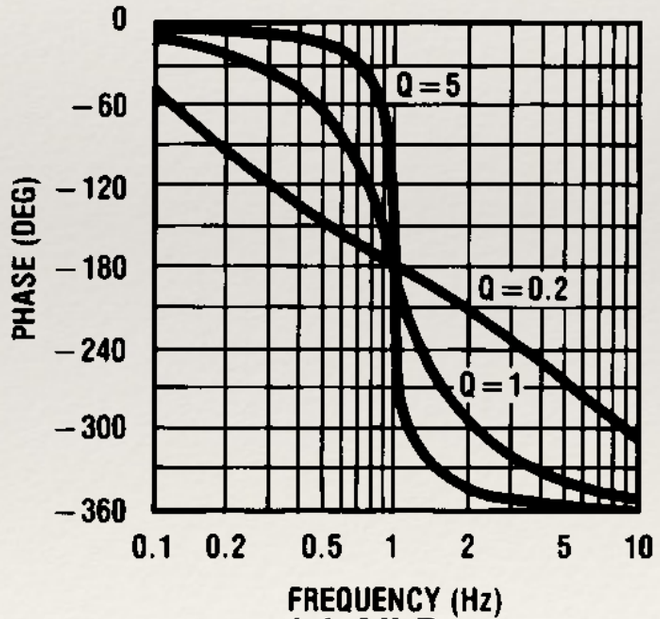
(b) Low-Pass



(c) High-Pass



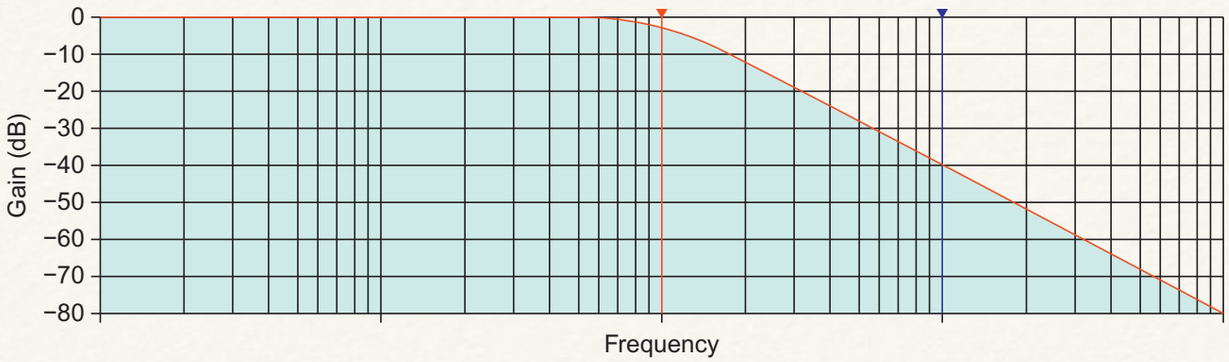
(d) Notch

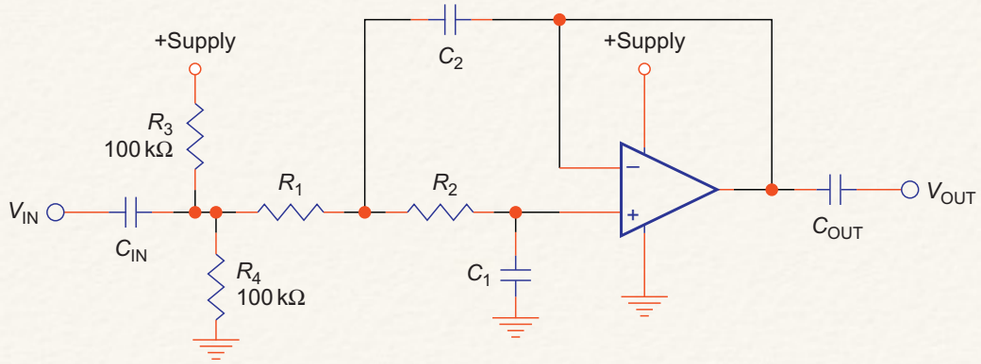


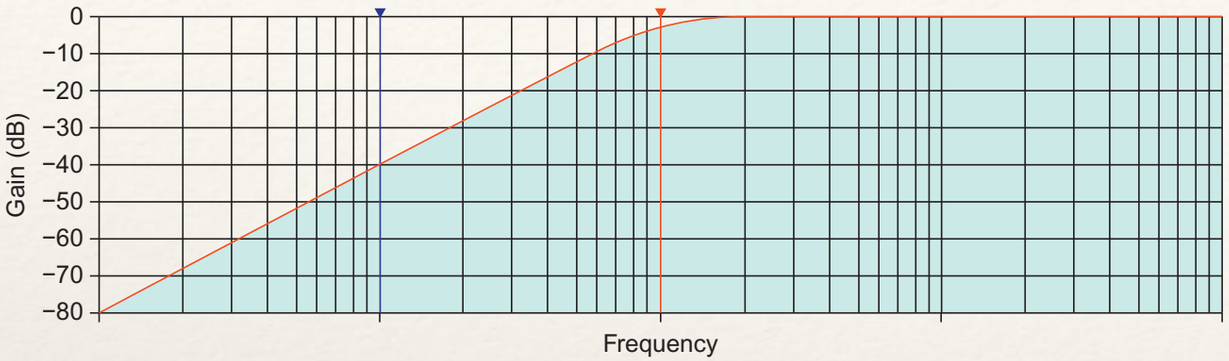
(e) All-Pass

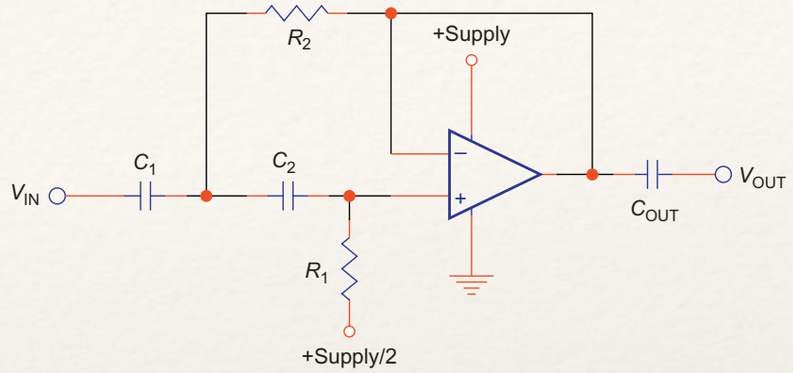
TL/H/11221–26

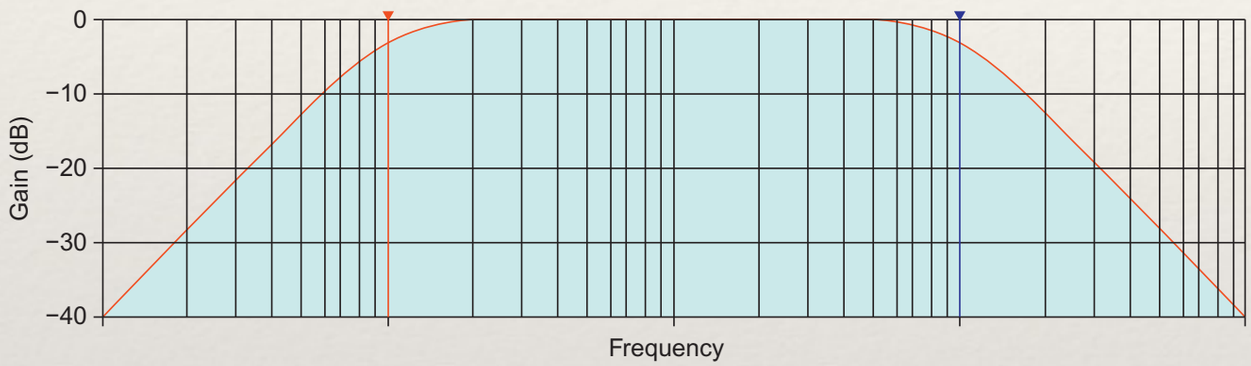
FIGURE 17. Responses of various 2nd-order filters as a function of Q. Gains and center frequencies are normalized to unity.

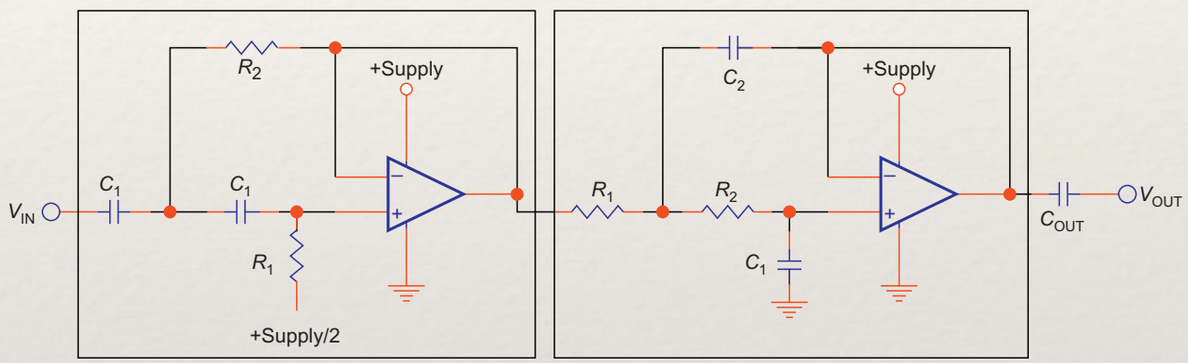

0
−10
−20
−30
−40
−50
−60
−70
−80
Gain (dB)
Frequency

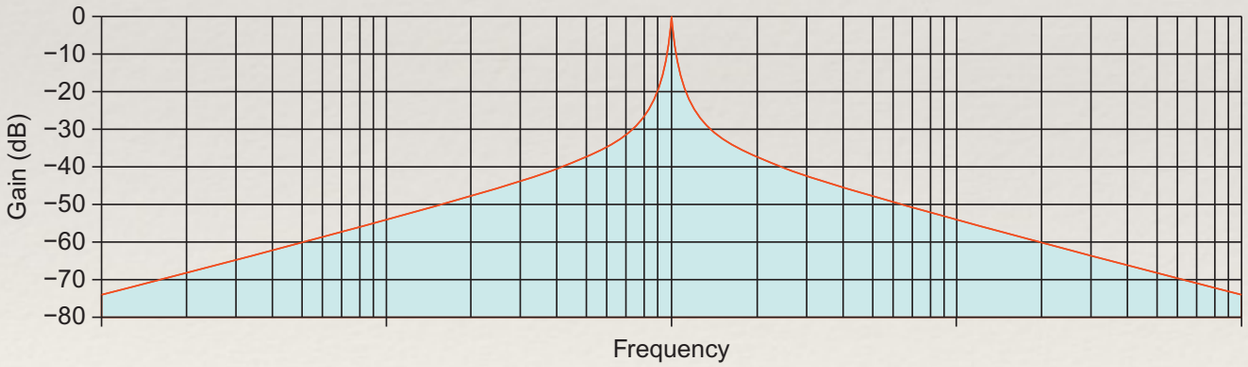

+Supply
C_2
+Supply
R_3
100 kΩ
R_1
R_2
V_{IN}
C_{IN}
R_4
100 kΩ
C_1
C_{OUT}
V_{OUT}

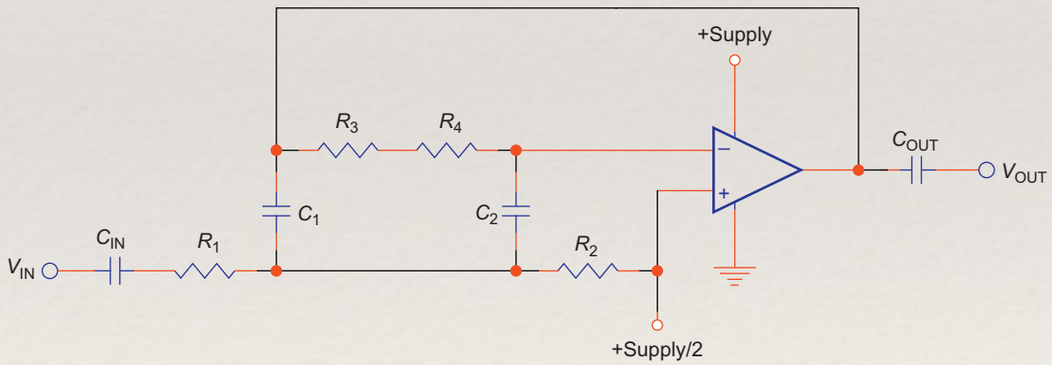

0
−10
−20
−30
−40
−50
−60
−70
−80
Gain (dB)
Frequency

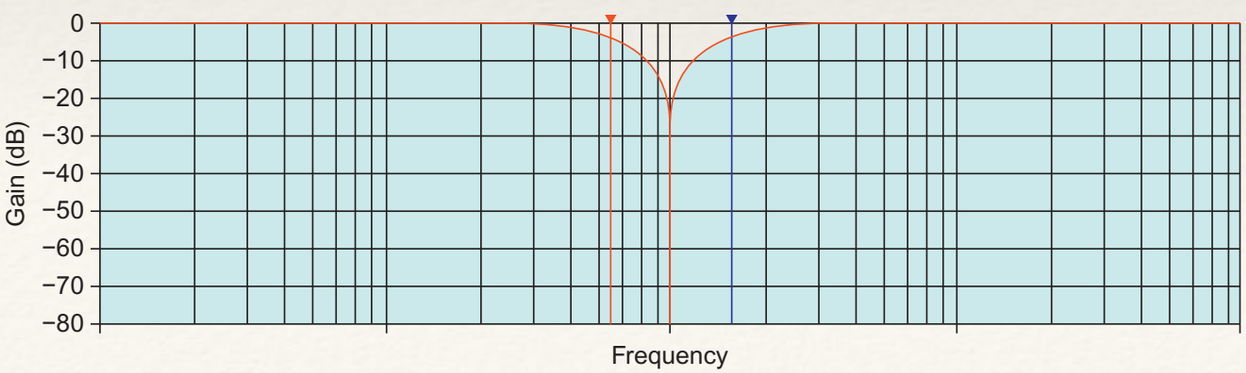

R_2
+Supply
C_1
C_2
V_{IN}
C_{OUT}
V_{OUT}
R_1
+Supply/2

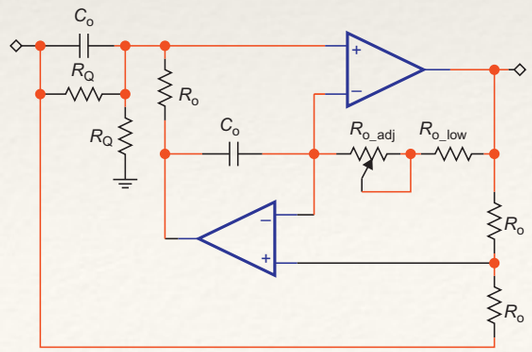

0
−10
−20
−30
−40
Gain (dB)
Frequency


R_2
+Supply
C_1
C_1
V_{IN}
R_1
+Supply/2
C_2
+Supply
R_1
R_2
C_1
C_{OUT}
V_{OUT}


0
−10
−20
−30
−40
−50
−60
−70
−80
Gain (dB)
Frequency

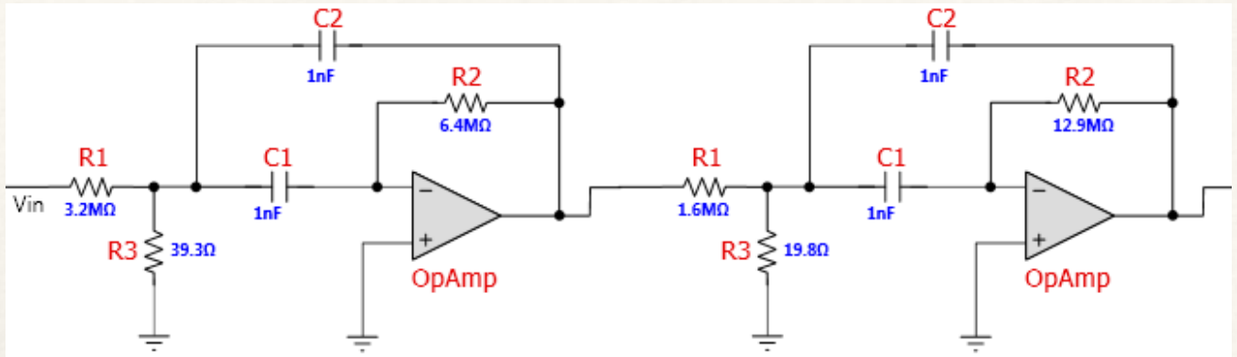

+Supply
R_3
R_4
C_{OUT}
V_{OUT}
C_1
C_2
C_{IN}
R_1
R_2
V_{IN}
+Supply/2


0
−10
−20
−30
−40
−50
−60
−70
−80
Gain (dB)
Frequency

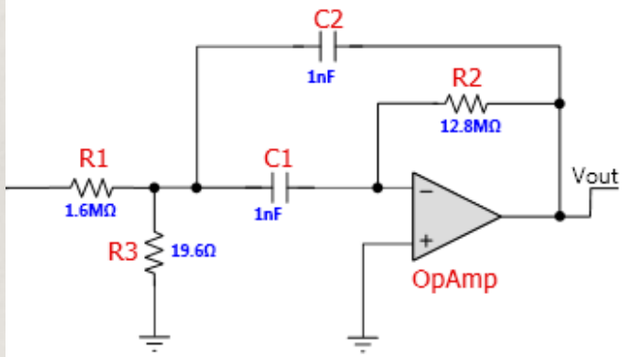

C_o
R_Q
R_Q
R_o
C_o
R_{o_adj}
R_{o_low}
R_o
R_o

FilterPro Design Report
Schematic

Design Name: Bandpass, Multiple Feedback, Chebyshev 1 dB **Part:** Ideal Opamp **Order:** 6 **Stages:** 3
Gain: 1 V/V (0 dB) **Allowable PassBand Ripple:** 1 dB **Center Frequency:** 10 kHz
Corner Frequency Attenuation: 0 dB **Stopband Attenuation:** -45 dB **Passband Bandwidth:** 100 Hz
Stopband Bandwidth: 1 kHz



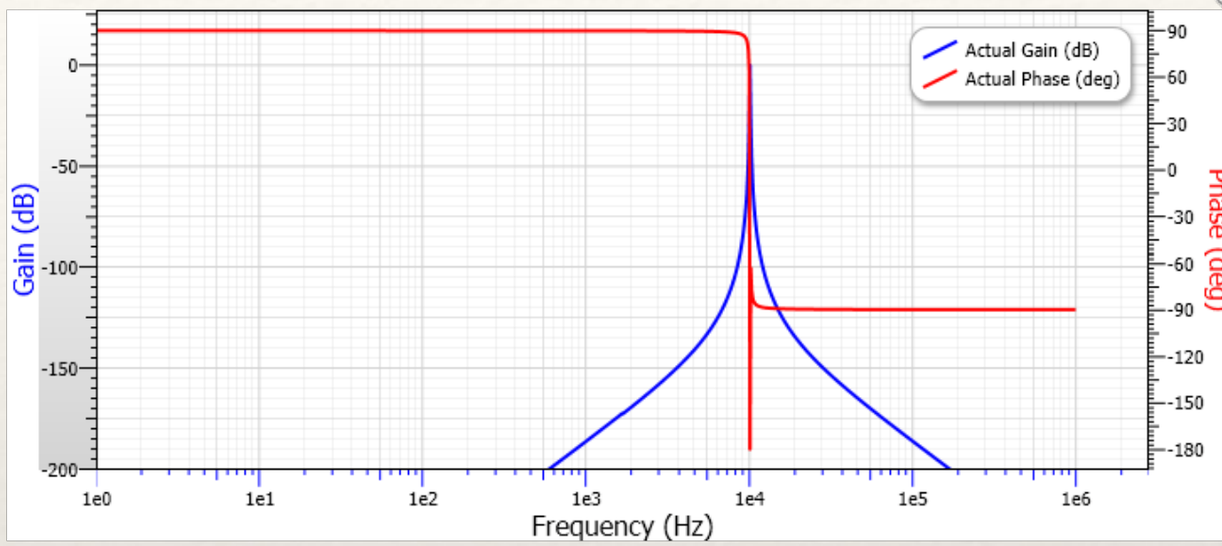
Filter Stage:	1	Filter Stage:	2
Passband Gain(Ao) :	1	Passband Gain(Ao) :	1
Center Frequency (fo):	10 kHz	Center Frequency (fo):	9.9518 kHz
QualityFactor (Q):	202.359	QualityFactor (Q):	404.723
Passband BW. (BW):	49.417 Hz	Passband BW. (BW):	24.589 Hz
Filter Response:	Chebyshev1dB	Filter Response:	Chebyshev1dB
Circuit Topology:	MultipleFeedback	Circuit Topology:	MultipleFeedback
Min GBW reqd.:	202.359 MHz	Min GBW reqd.:	402.7729 MHz

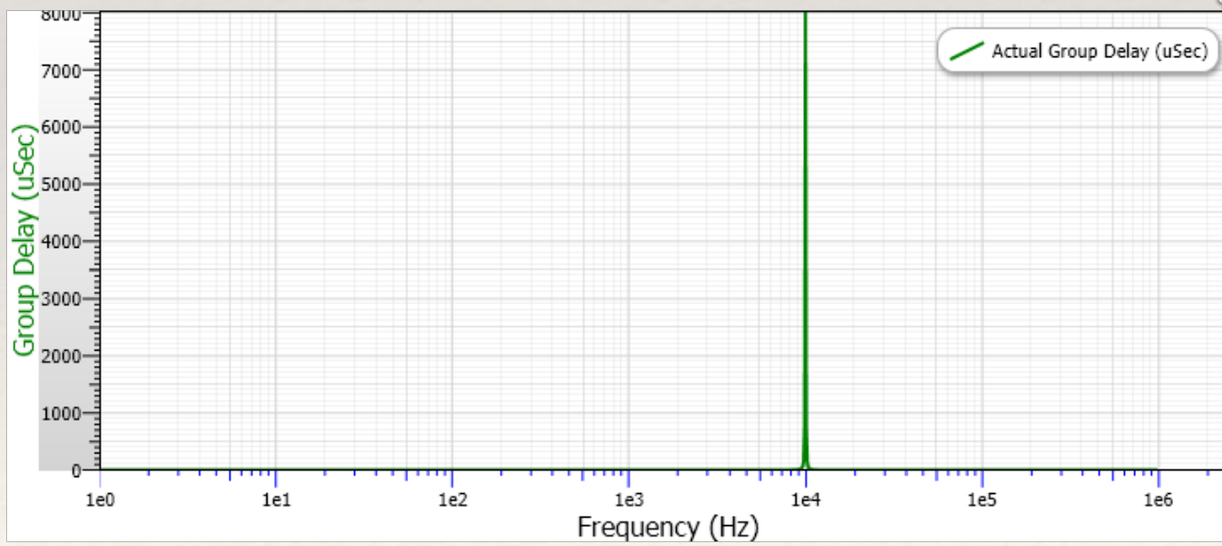


Filter Stage:	3
Passband Gain(Ao) :	1
Center Frequency (fo):	10.0484 kHz
QualityFactor (Q):	404.723
Passband BW. (BW):	24.828 Hz
Filter Response:	Chebyshev1dB
Circuit Topology:	MultipleFeedback
Min GBW reqd.:	406.6825 MHz

FilterPro Design Report
Frequency and Phase Responses

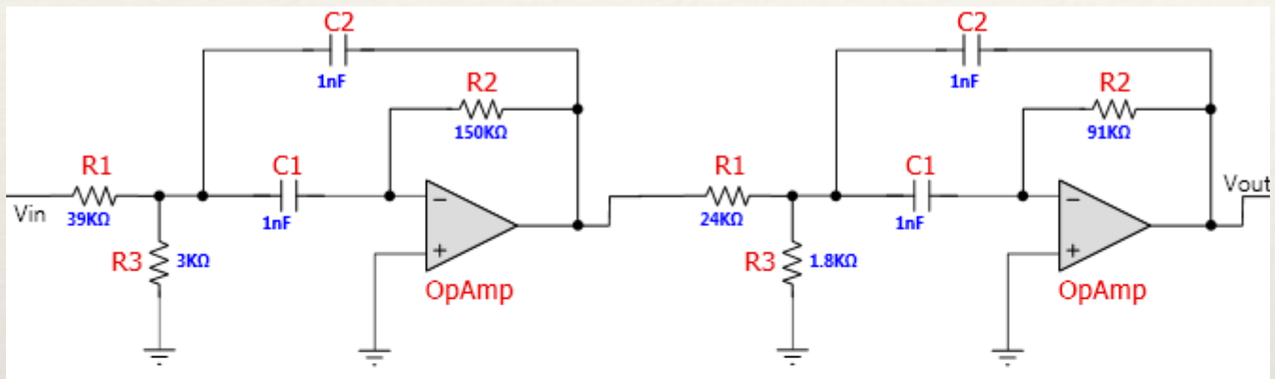
Design Name: Bandpass, Multiple Feedback, Chebyshev 1 dB **Part:** Ideal Opamp **Order:** 6 **Stages:** 3
Gain: 1 V/V (0 dB) **Allowable PassBand Ripple:** 1 dB **Center Frequency:** 10 kHz
Corner Frequency Attenuation: 0 dB **Stopband Attenuation:** -45 dB **Passband Bandwidth:** 100 Hz
Stopband Bandwidth: 1 kHz



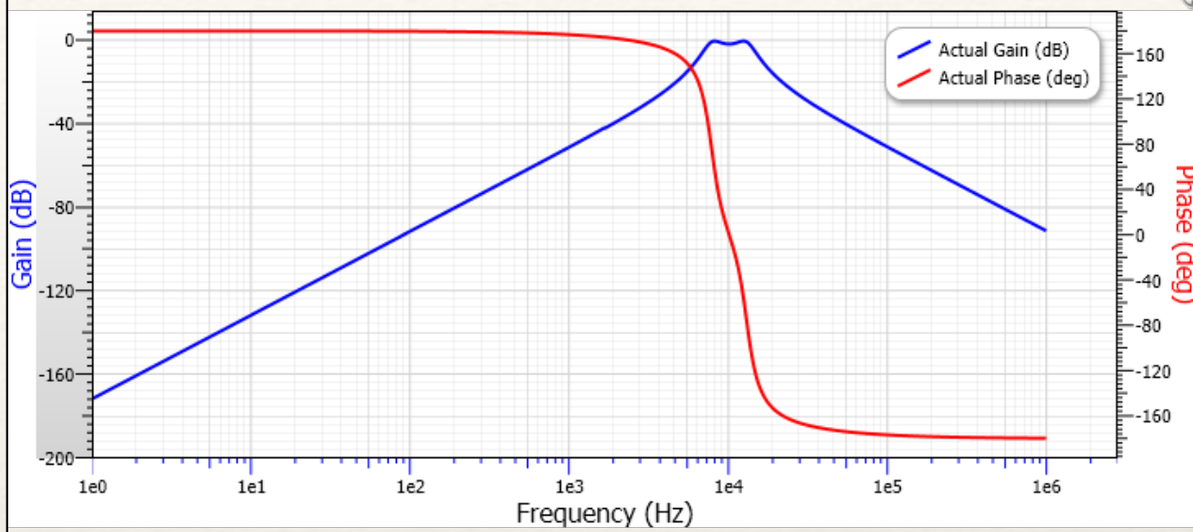


FilterPro Design Report
Schematic

Design Name: Bandpass, Multiple Feedback, Chebyshev 1 dB **Part:** Ideal Opamp **Order:** 4 **Stages:** 2
Gain: 1 V/V (0 dB) **Allowable PassBand Ripple:** 1 dB **Center Frequency:** 10 kHz
Corner Frequency Attenuation: 0 dB **Passband Bandwidth:** 5 kHz



Filter Stage:	1	Filter Stage:	2
Passband Gain(Ao) :	1	Passband Gain(Ao) :	1
Center Frequency (fo):	7.9936 kHz	Center Frequency (fo):	12.5099 kHz
QualityFactor (Q):	3.736	QualityFactor (Q):	3.736
Passband BW. (BW):	2.1398 kHz	Passband BW. (BW):	3.3488 kHz
Filter Response:	Chebyshev1dB	Filter Response:	Chebyshev1dB
Circuit Topology:	MultipleFeedback	Circuit Topology:	MultipleFeedback
Min GBW reqd.:	2.9864 MHz	Min GBW reqd.:	4.6737 MHz



FilterPro Design Report
Bill of Materials

Design Name: Bandpass, Multiple Feedback, Chebyshev 1 dB **Part:** Ideal Opamp **Order:** 4 **Stages:** 2
Gain: 1 V/V (0 dB) **Allowable PassBand Ripple:** 1 dB **Center Frequency:** 10 kHz
Corner Frequency Attenuation: 0 dB **Passband Bandwidth:** 5 kHz

Element ID	Quantity	Part Number	Value	Tolerance	Description	Manufacturer
R1 (Stage 1)	1	Standard	39KΩ	E24: 5%	Resistor	
R2 (Stage 1)	1	Standard	150KΩ	E24: 5%	Resistor	
R3 (Stage 1)	1	Standard	3KΩ	E24: 5%	Resistor	
C1 (Stage 1)	1	Standard	1nF	E24: 5%	Capacitor	
C2 (Stage 1)	1	Standard	1nF	E24: 5%	Capacitor	
OpAmp (Stage 1)	1	Standard			Ideal OpAmp	
R1 (Stage 2)	1	Standard	24KΩ	E24: 5%	Resistor	
R2 (Stage 2)	1	Standard	91KΩ	E24: 5%	Resistor	
R3 (Stage 2)	1	Standard	1.8KΩ	E24: 5%	Resistor	
C1 (Stage 2)	1	Standard	1nF	E24: 5%	Capacitor	
C2 (Stage 2)	1	Standard	1nF	E24: 5%	Capacitor	
OpAmp (Stage 2)	1	Standard			Ideal OpAmp	

Programas

http://www.analog.com/designtools/en/filterwizard/#/type

http://www.microchip.com/pagehandler/en_us/devtools/filterlab-filter-design-software.html

http://www.ti.com/tool/filterpro

Filtros Digitales

	Convolución **Finite Impulse Response** **(FIR)**	**Recursivos** **Infinite Impulse Response** **(IIR)**
Dominio del Tiempo	Moving Average	Polo Simple
Dominio de la Fecuencia	Ventanas	Chebyshev, Butterworth
Deconvolución	FIR custom	Diseño Iterativo

http://www.dspguide.com/pdfbook.htm

Moving Average Filters

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

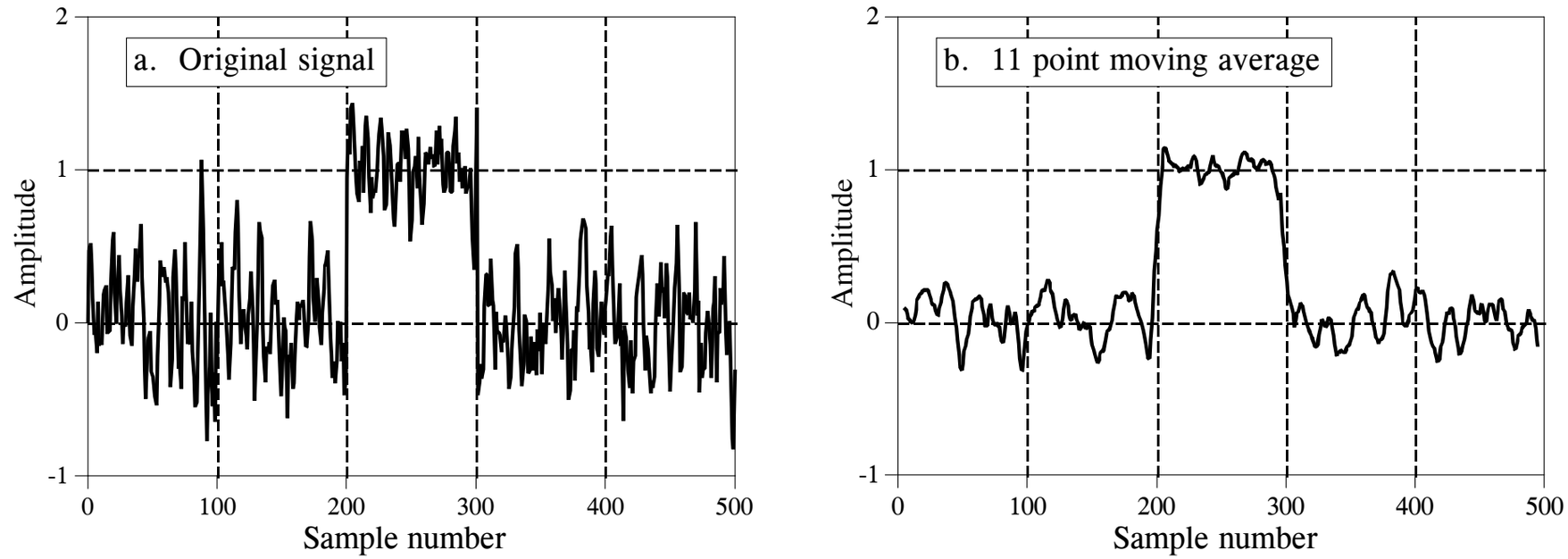
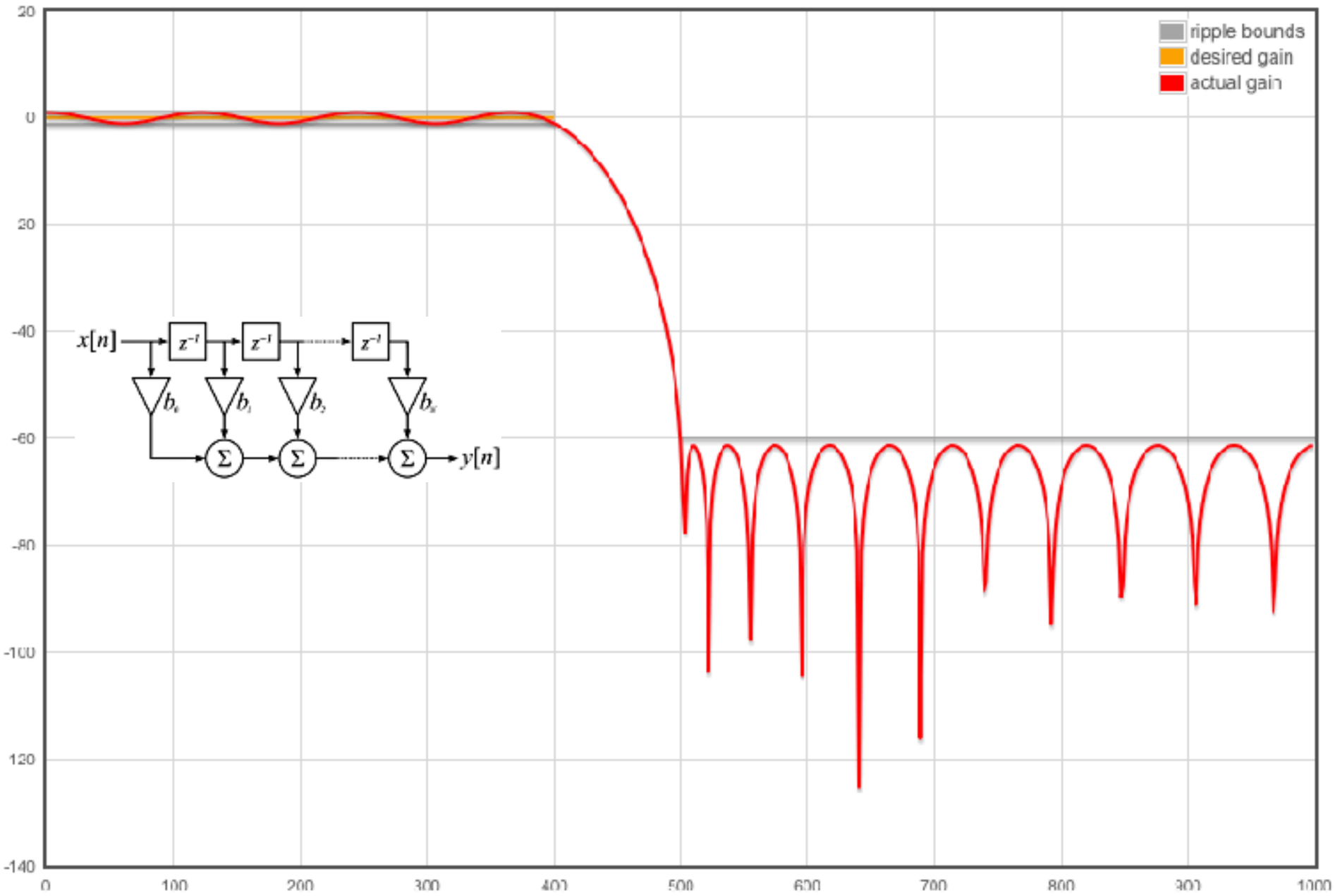


FIGURE 15-1
Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively. As the number of points in the filter increases, the noise becomes lower; however, the edges becoming less sharp. The moving average filter is the *optimal* solution for this problem, providing the lowest noise possible for a given edge sharpness.

Filtros FIR

http://t-filter.engineerjs.com

Gain vs. Frequency | Impulse Response | Source Code | Feature Request | Enterprise | IIR Design



need a FIR filter? TFilter BETA

plain text | double

0.0036424957397956912
0.015366656813275168
0.027003204939607993
0.024022584867401598
0.0007290843056071 9
-0.019511842448753997
-0.010514099441628916
0.017180018589579162
0.021242216298420546
-0.011480133602415641
-0.03400269851455443
-0.001766558484396119 2
0.047140188605091 73
0.027812614031606 59
-0.058616311088855 35
-0.082538292532277 24
0.056401994878518 53
0.310081701378692 2
0.430826306823455 33
0.310081701378692 2
0.056401994878518 53
-0.082538292532277 24
-0.058616311088855 35
0.027812614031606 59
0.047140188605091 73
-0.001766558484396119 2
-0.034002698514554 43
-0.011480133602415641
0.021242216298420546
0.017180018589579162
-0.010514099441628916
-0.019511842448753997

TFilter, th...
Like Page

Buy me a beer | Tweet

add passband | add stopband | !predefined

from	to	gain	ripple/att.	act. rpl
0 Hz	400 Hz	1	3 dB	2.02 dB
500 Hz	1000 Hz	0	-60 dB	-61.36 dB

sampling freq.	2000 Hz
desired #taps	minimum
actual #taps	37

DESIGN FILTER

I am working on **TFilter2**. Screenshot here. Features include CIC (Sinc) filters, effect of quantization, save/load/share, aliasing visualization, and signal chain.

If you want to **advertise here**, contact me at peterisza@gmail.com.

TFilter is being used by many tech companies and universities.

Filtros FIR

http://t-filter.engineerjs.com

Gain vs. Frequency | **Impulse Response** | Source Code | Feature Request | Enterprise | IIR Design

need a FIR filter? BETA TFilter

plain text | double

0.0036424957397856912
0.015366656813275186
0.027303284939667993
0.024022584867401598
0.0007290843060719
-0.019511842440753997
-0.010514099441628916
0.017180018589579182
0.021242216298420546
-0.011480133602415641
-0.034002698514554443
-0.0017655584343961192
0.04714018860509178
0.027812814031606590
-0.058616311088855335
-0.08253829253227724
0.06640199487351858
0.3100817013786922
0.43082639682345536
0.3100817013786922
0.06640199487351858
-0.08253829253227724
-0.058616311088855335
0.027812814031606590
0.04714018860509178
-0.0017655584343961192
-0.034002698514554443
-0.011480133602415641
0.021242216298420546
0.017180018589579182
-0.010514099441628916
-0.019511842440753997

TFilter, th...
Like Page

Buy me a beer | Tweet

add passband | add stopband | !predefined

from	to	gain	ripple/att.	act. rpl	
0 Hz	400 Hz	1	3 dB	2.02 dB	
500 Hz	1000 Hz	0	-60 dB	-61.36 dB	

sampling freq.	2000 Hz
desired #taps	minimum
actual #taps	37

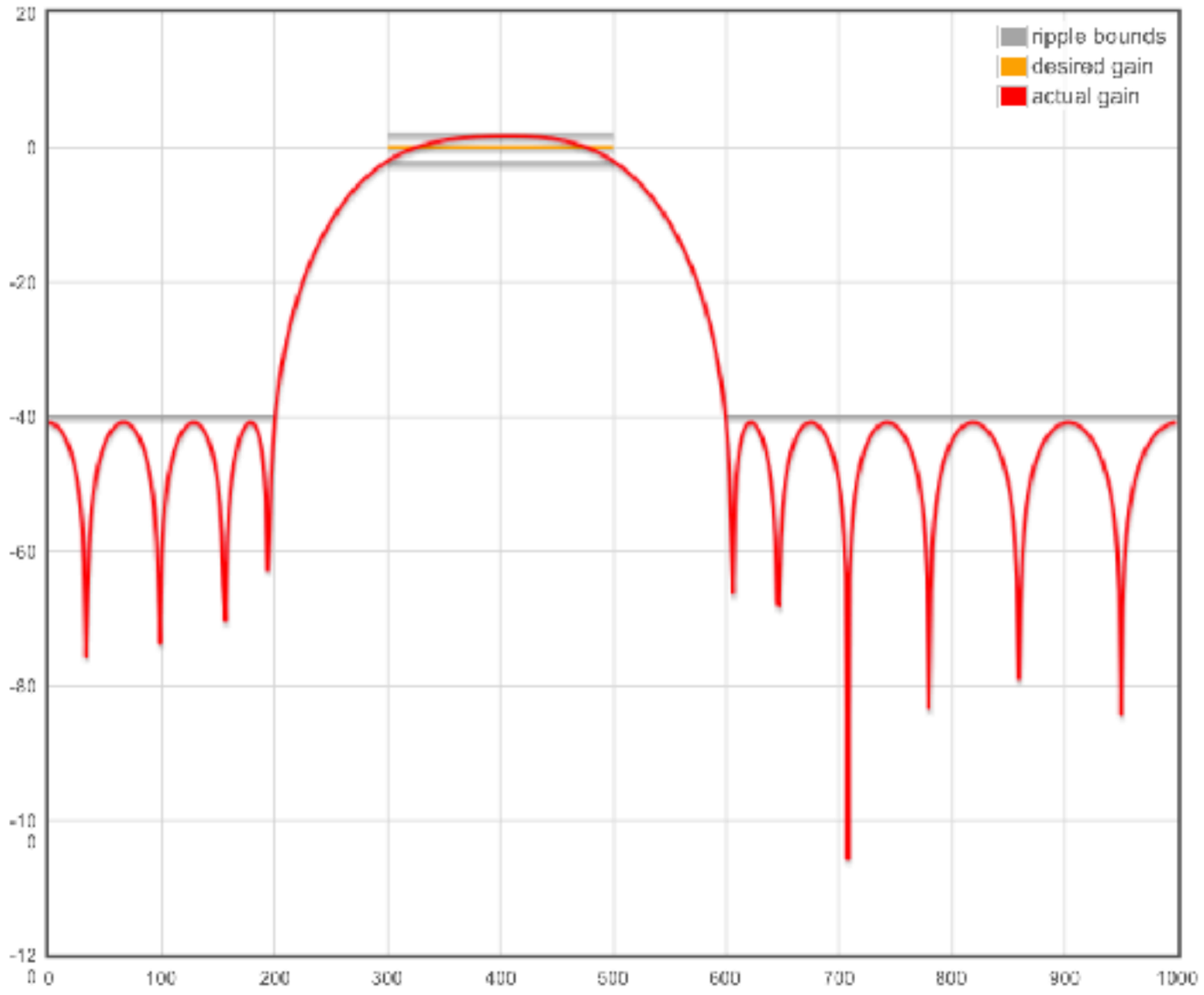
DESIGN FILTER

I am working on **TFilter2**. Screenshot here. Features include CIC (Sinc) filters, effect of quantization, save/load/share, aliasing visualization, and signal chain.

If you want to **advertise here**, contact me at peterisza@gmail.com.

TFilter is being used by many tech companies and universities.

Gain vs. Frequency | Impulse Response | Source Code | Feature Request | Enterprise | IIR Design



need a FIR filter? TFilter BETA

C/C++ array | double

```
/*

FIR filter designed with
http://t-filter.appspot.com

sampling frequency: 2000 Hz

* 0 Hz - 200 Hz
  gain = 0
  desired attenuation = -40 dB
  actual attenuation = -40.9645338753

* 300 Hz - 500 Hz
  gain = 1
  desired ripple = 5 dB
  actual ripple = 3.7226521905529983

* 600 Hz - 1000 Hz
  gain = 0
  desired attenuation = -40 dB
  actual attenuation = -40.9645338753

*/

#define FILTER_TAP_NUM 27

static double filter_taps[FILTER_TAP_
  0.008315515510920160,
```

TFilter, th... Like Page

Buy me a beer | Tweet

Copyright © 2011 Peter Isza

add passband | add stopband | !predefined

from	to	gain	ripple/att.	act. rpl	
0 Hz	200 Hz	0	-40 dB	-40.96 dB	
300 Hz	500 Hz	1	5 dB	3.72 dB	
600 Hz	1000 Hz	0	-40 dB	-40.96 dB	

sampling freq.	2000 Hz
desired #taps	minimum
actual #taps	27

DESIGN FILTER

I am working on **TFilter2**. Screenshot here. Feature
effect of quantization, save/load/share, aliasing visua

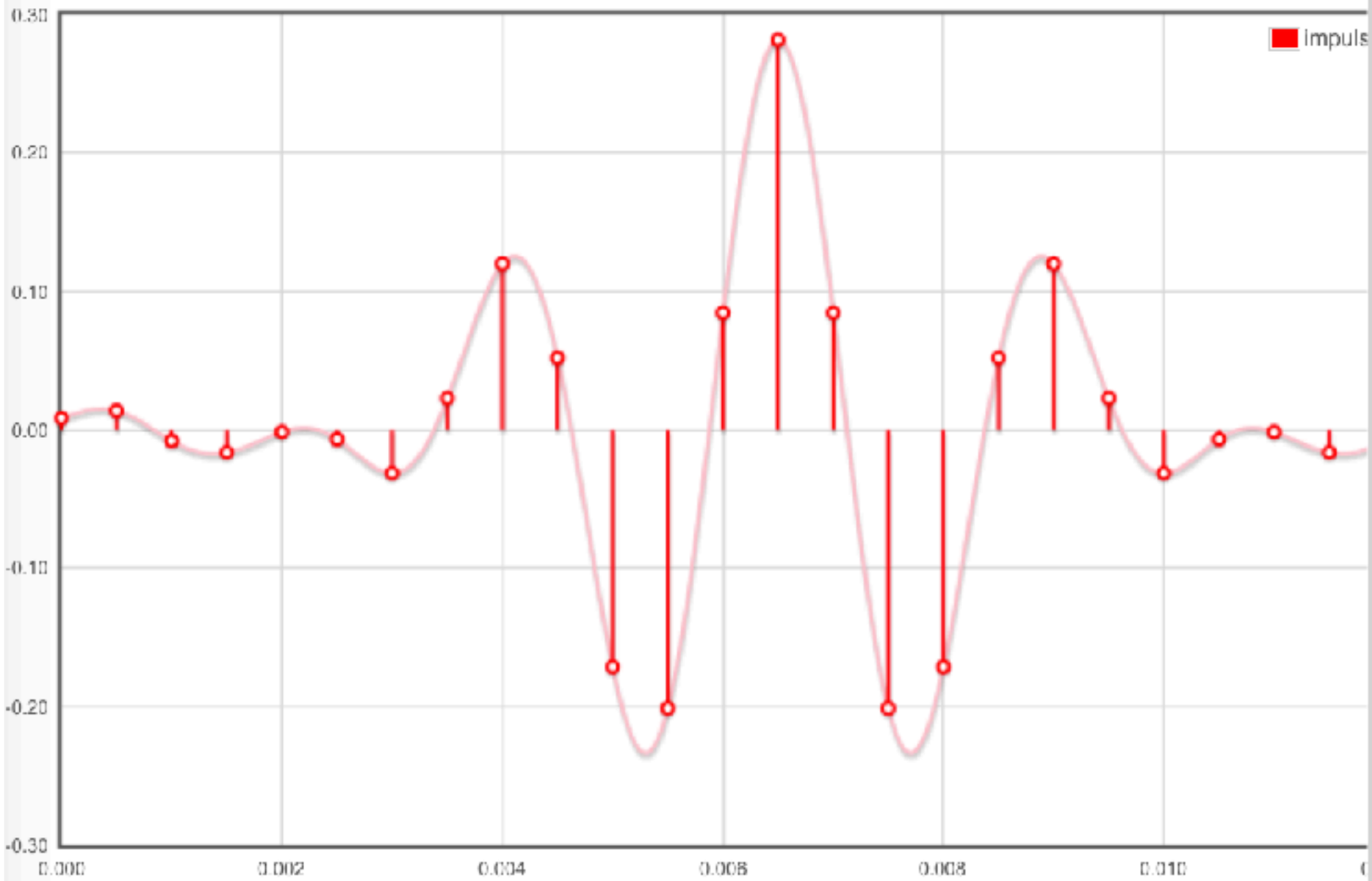
If you want to **advertise here**, contact me at peterisz

TFilter is being used by many tech companies and ur

```
/*

FIR filter designed with
http://t-filter.appspot.com

sampling frequency: 2000 Hz

* 0 Hz - 200 Hz
  gain = 0
  desired attenuation = -40 dB
  actual attenuation = -40.96453387533025 dB

* 300 Hz - 500 Hz
  gain = 1
  desired ripple = 5 dB
  actual ripple = 3.7226521905529983 dB

* 600 Hz - 1000 Hz
  gain = 0
  desired attenuation = -40 dB
  actual attenuation = -40.96453387533025 dB

*/

#define FILTER_TAP_NUM 27

static double filter_taps[FILTER_TAP_NUM] = {
  0.008315515510920168,
  0.01370300881920289,
  -0.00812525725784532,
  -0.01649214060817744,
  -0.001688459347147192,
  -0.006913035271285233 5,
  -0.03139161346522067,
  0.022740863526438886,
  0.1198490872411675,
  0.051863550355234773,
  -0.17137740316854042,
  -0.20124347467075904,
  0.08441813048666598,
  0.281631430933639,
  0.08441813048666598,
  -0.20124347467075904,
  -0.17137740316854042,
  0.051863550355234773,
  0.1198490872411675,
  0.022740863526438886,
  -0.03139161346522067,
  -0.006913035271285233 5,
  -0.001688459347147192,
  -0.01649214060817744,
  -0.00812525725784532,
  0.01370300881920289,
  0.008315515510920168
};
```

Gain vs. Frequency | **Impulse Response** | Source Code | Feature Request | Enterprise | IIR Design

need a FIR filter? TFilter BETA



plain text | double

```
0.008315515510920168
0.013703008819202889
-0.008125257257845832
-0.016492140608177444
-0.0016884593471471920
-0.006913035271205233 5
-0.031391613465220670
0.022740863526438886
0.11984908724116750
0.051863550355234773
-0.171377403168540420
-0.201243474670759040
0.084418130486665980
0.261631430933639
0.084418130486665980
-0.201243474670759040
-0.171377403168540420
0.051863550355234773
0.11984908724116750
0.022740863526438886
-0.031391613465220670
-0.006913035271205233 5
-0.0016884593471471920
-0.016492140608177444
-0.008125257257845832
0.013703008819202889
0.008315515510920168
```

TFilter, th…

Like Page

Buy me a beer | Tweet

add passband | add stopband | !predefined

from	to	gain	ripple/att.	act. rpl
0 Hz	200 Hz	0	-40 dB	-40.96 dB
300 Hz	500 Hz	1	5 dB	3.72 dB
600 Hz	1000 Hz	0	-40 dB	-40.96 dB

sampling freq.	2000 Hz
desired #taps	minimum
actual #taps	27

DESIGN FILTER

I am working on **TFilter2.** Screenshot here. Features i
effect of quantization, save/load/share, aliasing visualiza

If you want to **advertise here**, contact me at peterisza@

TFilter is being used by many tech companies and unive

fir_filter.cpp

```cpp
/*
    This is a very simple C++ implenentation of a floating point FIR filter.
 */

template <typename FloatType, int num_coeffs, const FloatType* coeffs>
class FirFilter {
public:
    FirFilter():
        current_index_(0) {
    for(int i = 0; i < num_coeffs; ++i)
        history_[i] = 0.0;
    }

    void put(FloatType value) {
        history_[current_index_++] = value;
    if(current_index_ == num_coeffs)
        current_index_ = 0;
    }

    FloatType get() {
        FloatType output = 0.0;
        int index = current_index_;
        for(int i = 0; i < num_coeffs; ++i) {
            if(index != 0) {
                --index;
            } else {
                index = num_coeffs - 1;
            }
            output += history_[index] * coeffs[i];
        }
        return output;
    }

private:
    FloatType history_[num_coeffs];
    int current_index_;
};
```

```cpp
/* This is a test program that shows how to use the FirFilter c

#include <iostream>
#include <assert.h>
#include <math.h>

#define FILTER1_LENGTH 4

float filter1_coeffs[FILTER1_LENGTH] = {
    0.1,
    0.2,
    0.3,
    0.4
};

bool approx_equal(float a, float b) {
    return fabs(a-b) < 1e-6;
}

int main() {
    FirFilter<float, FILTER1_LENGTH, filter1_coeffs> filter1;

    filter1.put(10.0);
    assert(approx_equal(filter1.get(), 1));
    filter1.put(10.0);
    assert(approx_equal(filter1.get(), 3));
    filter1.put(10.0);
    assert(approx_equal(filter1.get(), 6));
    filter1.put(10.0);
    assert(approx_equal(filter1.get(), 10));

    filter1.put(0.0);
    assert(approx_equal(filter1.get(), 9));
    filter1.put(0.0);
    assert(approx_equal(filter1.get(), 7));
    filter1.put(0.0);
    assert(approx_equal(filter1.get(), 4));
    filter1.put(0.0);
    assert(approx_equal(filter1.get(), 0));

    std::cout << "Tests OK." << std::endl;

    return 0;
}
```