

Digital Signal Processing

Module 7: Stochastic Signal Processing and Quantization

Module Overview:



- Module 7.2: A/D and D/A conversion 2013

 Page 1.3

 Module 7.2: Quantization

 Page 2013



Digital Signal Processing

Digital Signal Processing

Module 7.1: Stochastic signal processing



- Filtering a stochastic signal processing Vetterli

 Noise Paolo Prandoni and Martin Vetterli

 © 2013



- ► Filtering a stochastic sign Prandoni and Martin Vetterli

 Noise Paolo Prandoni © 2013



- Filtering a stochastic signal processing Vetterli

 Noise

 Paolo Prandoni and Martin Vetterli

 © 2013



- Filtering a stochastic signal processing Vetterli

 Noise

 Paolo Prandoni and Martin Vetterli

 2013



- deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- interesting signals are *not* known in advance: s[n] = M what l'_1m_1 being to say next
- we usually know something, though ghalis a speech signal
- stochastic signals can be described probabilistically
- can we do signal properties with random signals? Yes!
- will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



- deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- ▶ interesting signals are *not* known in advance: s[n] = what I'm going to say next
- we usually know something, though goth signal
- stochastic signals can be described probabilistically
- can we do signal processing with random signals? Yes!
- will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



- deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- ▶ interesting signals are *not* known in advance: s[n] = what I'm going to say next
- ightharpoonup we usually know something, though: s[n] is a speech signal
- ► stochastic signals can be described probabilistically
- can we do signal processing with random signals? Yes!
- will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



- deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- ▶ interesting signals are *not* known in advance: s[n] = what I'm going to say next
- ightharpoonup we usually know something, though: s[n] is a speech signal
- stochastic signals can be described probabilistically
- can we do signal processing with random signals? Yes!
- will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



- ▶ deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- ▶ interesting signals are *not* known in advance: s[n] = what I'm going to say next
- ightharpoonup we usually know something, though: s[n] is a speech signal
- stochastic signals can be described probabilistically
- ▶ can we do signal processing with random signals? Yes!
- will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



- deterministic signals are known in advance: $x[n] = \sin(0.2 n)$
- ▶ interesting signals are *not* known in advance: s[n] = what I'm going to say next
- ightharpoonup we usually know something, though: s[n] is a speech signal
- stochastic signals can be described probabilistically
- ▶ can we do signal processing with random signals? Yes!
- ▶ will not develop stochastic signal processing rigorously but give enough intuition to deal with things such as "noise"



For each new sample, toss a fair coin:

$$x[n] = \begin{cases} +1 & \text{if the outcome of the n-th toss is head} \\ -1 & \text{if the outcome of the n-th toss is tail} \end{cases}$$
 each sample is independent from all others



For each new sample, toss a fair coin:

$$x[n] = \begin{cases} +1 & \text{if the outcome of the } n\text{-th toss is head} \\ -1 & \text{if the outcome of the } n\text{-th toss is tail} \end{cases}$$

- each sample is independent from all others
- ▶ each sample value has a 50% probability



- every time we turn on the generator we obtain a different realization of the signal
- we know the "mechanism" behind each instance
- but how can we analyze a random signal?

 Digital Signal Processing

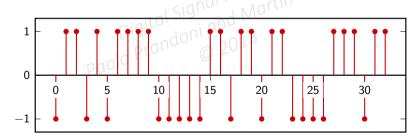
 Digital Signal Martin Vetterli

 Digital Signal Processing

 Di

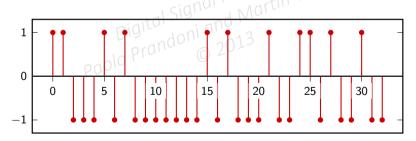


- every time we turn on the generator we obtain a different realization of the signal
- we know the "mechanism" behind each instance
- but how can we analyze a random signal? processing vetterly



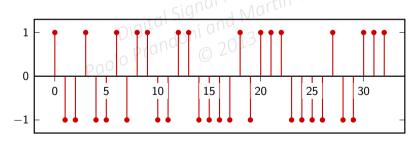


- every time we turn on the generator we obtain a different realization of the signal
- we know the "mechanism" behind each instance
- but how can we analyze a random signal? processing Vetterli



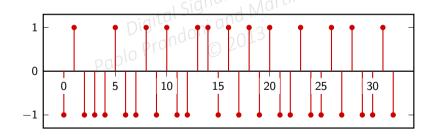


- every time we turn on the generator we obtain a different realization of the signal
- we know the "mechanism" behind each instance
- but how can we analyze a random signal? processing vetter





- every time we turn on the generator we obtain a different realization of the signal
- ▶ we know the "mechanism" behind each instance
- but how can we analyze a random signal?





- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?
- no clear pattern... we need a new strategy

 Digital Signal Processing

 Digital Signal Processing

 Processing

 Vetterli

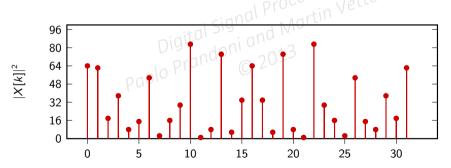
 Digital Signal Processing

 Digital Signal Processing



- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?

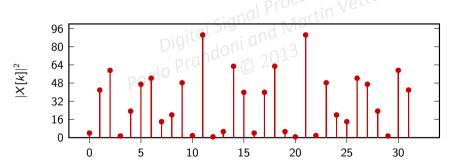
▶ no clear pattern... we need a new strategy





- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?

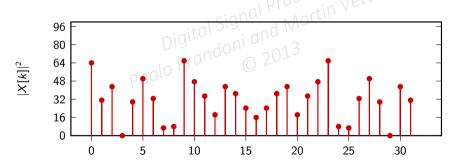
▶ no clear pattern... we need a new strategy





- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?

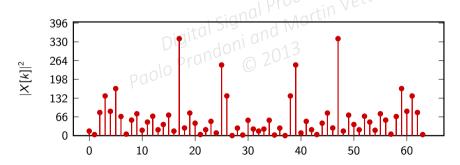
▶ no clear pattern... we need a new strategy





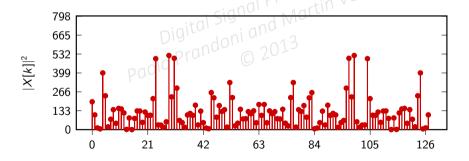
- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?

▶ no clear pattern... we need a new strategy





- ▶ let's try with the DFT of a finite set of random samples
- every time it's different; maybe with more data?
- ▶ no clear pattern... we need a new strategy





- ▶ when faced with random data an intuitive response is to take "averages"
- ▶ for the coin-toss signal:



- when faced with random data an intuitive response is to take "averages"
- ▶ in probability theory the average is across realizations and it's called *expectation*
- ► for the coin-toss signal:

toss signal:
$$Digital Signal Martin$$

$$Digital Signal Martin$$

$$E[x[n]] = Digital Signal Martin$$

$$Digital Signal Martin$$

▶ so the average value for each sample is zero...



- ▶ when faced with random data an intuitive response is to take "averages"
- ▶ in probability theory the average is across realizations and it's called *expectation*
- ▶ for the coin-toss signal:

ty theory the average is across realizations and it's called *expectations* signal:
$$E[x[n]] = -1 \cdot P[n-\text{th toss is tail}] + 1 \cdot P[n-\text{th toss is head}] = 0$$

▶ so the average value for each sample is zero...



- ▶ when faced with random data an intuitive response is to take "averages"
- ▶ in probability theory the average is across realizations and it's called *expectation*
- ► for the coin-toss signal:

▶ so the average value for each sample is zero...

Averaging the DFT



- ... as a consequence, averaging the DFT will not work Vetterli

 E[X[k]] = 0• however the signal "moves", profits energy 20 power must be nonzero paolo

Averaging the DFT



- ... as a consequence, averaging the DFT will not work vetter!!
 E[X[k]] = 0
 however the signal "moves", profits energy 20 power must be nonzero paolo

Averaging the DFT



- ... as a consequence, averaging the DFT will not work vetter E[X[k]] = 0
- ▶ E[X[k]] = 0
- ▶ however the signal "moves", so its energy or power must be nonzero

Energy and power



▶ the coin-toss signal has infinite energy (see Module 2.1):

$$E_{x} = \lim_{N \to \infty} \sum_{n=-N}^{N} |x[n]|^{2} = \lim_{N \to \infty} (2N + 1) = \infty$$

▶ however it has finite power over any interval: 13

$$P_{x} = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^{2} = 1$$



let's try to average the DFT's square magnitude, normalized:

- pick a number of iterations M | Signal Processing Vetterli
 run the signal generator Mitmes and obtain M3N-point realizations
 compute the DFT of peach realization



let's try to average the DFT's square magnitude, normalized:

- ▶ pick a number of iterations M
 ► run the signal generator Mitmes and obtain M3N-point realizations
 ► compute the DFT of peach realization

Averaging



let's try to average the DFT's square magnitude, normalized:

- pick an interval length N
- \triangleright pick a number of iterations M
- "tal Signal Processing

 * tal Signal Martin Vetterli

 * and okand Martin run the signal generator M times and obtain M N-point realizations
- compute the DFT of each realization

Averaging



let's try to average the DFT's square magnitude, normalized:

- pick an interval length N
- ightharpoonup pick a number of iterations M
- ightharpoonup run the signal generator M times and obtain M N-point realizations
- ▶ compute the DFT of each realization
- average their square magnitude divided by N

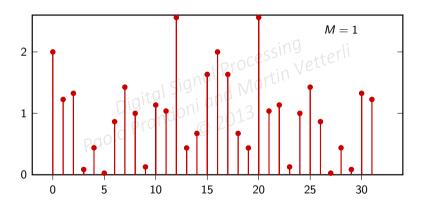
Averaging



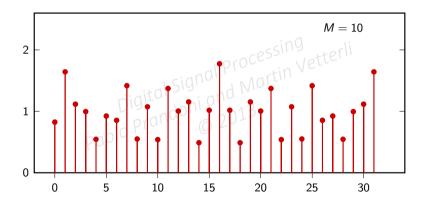
let's try to average the DFT's square magnitude, normalized:

- pick an interval length N
- ightharpoonup pick a number of iterations M
- ightharpoonup run the signal generator M times and obtain M N-point realizations
- compute the DFT of each realization
- lacktriangle average their square magnitude divided by N

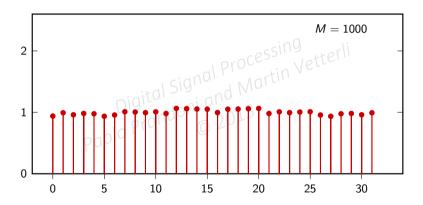




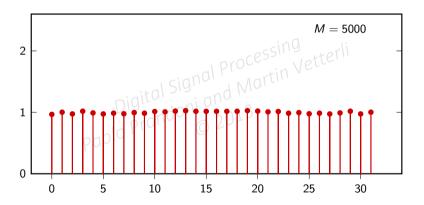














$$P[k] = E[|X_N[k]|^2/N]$$
• it looks very much as if $P[k] \neq 1$ | Signal Processing
• if $|X_N[k]|^2$ tends to the energy distribution (aka density) in frequency...

- $\longrightarrow ... |X_N[k]|^2/N$ tends to the power distribution (aka density) in frequency



- $\longrightarrow ... |X_N[k]|^2/N$ tends to the power distribution (aka density) in frequency



$$P[k] = E\left[|X_N[k]|^2/N\right]$$

- $P[k] = \operatorname{E} \left[|X_N[k]|^2 / N \right]$ it looks very much as if P[k] = 1 $|X_N[k]|^2 \text{ tends to the } energy \text{ distribution}$ $|X_N[k]|^2 / N \text{ tends to } 1$ $\rightarrow ... |X_N[k]|^2/N$ tends to the power distribution (aka density) in frequency



$$P[k] = \mathsf{E}\left[|X_N[k]|^2/N\right]$$

- $P[k] = \mathbb{E}\left[|X_N[k]|^2/N\right]$ it looks very much as if P[k] = 1• $|X_N[k]|^2$ tends to the energy distribution $|X_N[k]|^2/N$ tends to $|X_N[k]|^2/N$ \triangleright ... $|X_N[k]|^2/N$ tends to the power distribution (aka density) in frequency
 - ▶ the frequency-domain representation for stochastic processes is the power spectral density

Power spectral density: intuition



- ightharpoonup P[k] = 1 means that the power is equally distributed over all frequencies
- i.e., we cannot predict if the signal singles "slowled" or "super-fast"

 this is because each sample is independent of each other: we could have a realization of all

Power spectral density: intuition



- ightharpoonup P[k] = 1 means that the power is equally distributed over all frequencies
- ▶ i.e., we cannot predict if the signal moves "slowly" or "super-fast"
- b this is because each sample is independent of each other: we could have a realization of all ones or a realization in which the sign changes every other sample or anything in between

Power spectral density: intuition



- ightharpoonup P[k] = 1 means that the power is equally distributed over all frequencies
- ▶ i.e., we cannot predict if the signal moves "slowly" or "super-fast"
- ▶ this is because each sample is independent of each other: we could have a realization of all ones or a realization in which the sign changes every other sample or anything in between



- ▶ let's filter the random process with a 2-point Moving Average filter
- y[n] = (x[n] + x[n-1])/2Digital Signal Martin

 what is the power spectral density doni and Martin

 Paolo

 Paolo

 Paolo



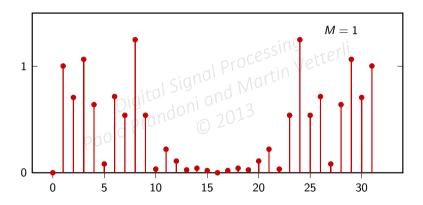
- ▶ let's filter the random process with a 2-point Moving Average filter
- ► y[n] = (x[n] + x[n-1])/2Note that is the power spectral density doni and Martin paolo 2013



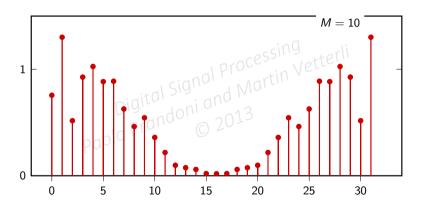
- ▶ let's filter the random process with a 2-point Moving Average filter
- ▶ y[n] = (x[n] + x[n-1])/2▶ what is the power spectral density?

 2013

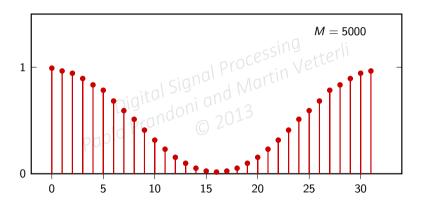




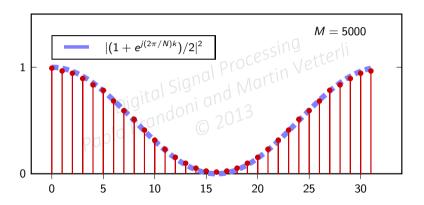














- it looks like $P_y[k] = P_x[k] |H[k]|^2$, where $H[k] = DFT\{h[n]\}$
 - can we generalize these repulge beyond in the set of samples?

 Paolo Prand © 2013



- ▶ it looks like $P_y[k] = P_x[k] |H[k]|^2$, where $H[k] = DFT\{h[n]\}$ ▶ can we generalize the second of the seco
- ► can we generalize these results beyond a finite set of samples?



- a stochastic process is characterized by its power spectral density (PSD)
- $P_{x}(e^{i\omega}) = \text{DTFT}_{x}[n] \text{ where } r_{x}[n] = \text{E}[x[k] \times [n-D] \text{ is the mittocorrelation of the process.}$ $P_{x}(e^{i\omega}) = \text{DTFT}_{x}[n] \text{ and } \text{Markin} \text{ is the mittocorrelation of the process.}$ $P_{x}(n) = P_{x}[n] = P_{x}[n], \text{ it is:}$

$$P_{y}(e^{j\omega}) = |H(e^{j\omega})|^{2} P_{x}(e^{j\omega})$$



- a stochastic process is characterized by its power spectral density (PSD)

$$P_{\mathsf{x}}(e^{j\omega}) = \mathsf{DTFT}\left\{r_{\mathsf{x}}[n]\right\}$$

 $P_{x}(e^{i\omega}) = \mathsf{DTFT}\{r_{x}[n]\}$ where $r_{x}[n] = \mathsf{E}[x[k] \times [n+k]]$ is the autocorrelation of the process. for a filtered stochastic phocess $y[n] = \mathcal{H}\{x[n]\}$, it is:

$$P_{y}(e^{j\omega}) = |H(e^{j\omega})|^{2} P_{x}(e^{j\omega})$$



- ▶ a stochastic process is characterized by its power spectral density (PSD)
- ▶ it can be shown (see the textbook) that the PSD is

$$P_{\mathsf{x}}(e^{j\omega}) = \mathsf{DTFT}\left\{r_{\mathsf{x}}[n]\right\}$$

 $P_{x}(e^{j\omega}) = \mathsf{DTFT}\left\{r_{x}[n]\right\}$ where $r_{x}[n] = \mathsf{E}\left[x[k]\,x[n+k]\right]$ is the autocorrelation of the process.

• for a filtered stochastic process $y[n] = \mathcal{H}\{x[n]\}$, it is:

$$P_{y}(e^{j\omega}) = |H(e^{j\omega})|^{2} P_{x}(e^{j\omega})$$



key points:

- ey points:

 filters designed for deterministic signals still work (in magnitude) in the stochastic case
- we lose the concept of phase since we don't know the shape of a realization in advance



key points:

- ▶ filters designed for deterministic signals still work (in magnitude) in the stochastic case
- ▶ we lose the concept of phase since we don't know the shape of a realization in advance



- noise is everywhere:
- quantization and numerical processing Vetterli

 quantization and numerical production and Martin Vetterli

 ...

 Digital Signal Martin Vetterli

 ...

 ...

 ve can model noise as a stochastic signal

 the most important points



- noise is everywhere:
- quantization and numerical processing Vetterli

 quantization and numerical production and Martin Vetterli

 ...

 Digital Signal Martin Vetterli

 ...

 ...

 ve can model noise as a stochastic signal

 the most important points



- noise is everywhere:
- quantization and numerical errors Signal Processing

 quantization and numerical errors and Martin Vetterli

 ...

 model noise as a stochastic signal

 the most important points.



- noise is everywhere:
- quantization and numerical errors ignal processing vetterli
 quantization and numerical errors and Martin vetterli
 ...

 we can model noise as a stochastic signal

 the most important point.



- noise is everywhere:
- quantization and numerical errors ignal processing vetterli
 quantization and numerical errors and Martin vetterli
 ...
 we can model noise as a stochastic signal

 the most important point.



- noise is everywhere:
 - thermal noise
 - sum of extraneous interferences
 - quantization and numerical errors
 - ..
- we can model noise as a stochastic signal
- ▶ the most important noise is white noise



- noise is everywhere:
 - thermal noise
 - sum of extraneous interferences
 - quantization and numerical errors
 - ..
- we can model noise as a stochastic signal
- ▶ the most important noise is white noise

White noise



- "white" indicates uncorrelated samples
- $ightharpoonup r_w[n] = \sigma^2 \delta[n]$
- $P_w(e^{j\omega}) = \sigma^2$

Digital Signal Processing

Digital Signal Martin Vetterli

Paolo Prandoni and Martin

© 2013



- "white" indicates uncorrelated samples
- $ightharpoonup r_w[n] = \sigma^2 \delta[n]$
- $P_w(e^{j\omega}) = \sigma^2$

Digital Signal Processing

Digital Signal Martin Vetterli



- Digital Signal Processing

 Digital Signal Martin Vetterli

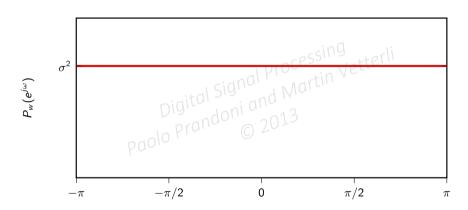
 Digital Signal Martin Vetterli

 Digital Signal Martin Vetterli

 Digital Signal Martin Vetterli

 Digital Signal Martin Vetterli "white" indicates uncorrelated samples
- $ightharpoonup r_w[n] = \sigma^2 \delta[n]$
- $P_w(e^{j\omega}) = \sigma^2$







- ▶ the PSD is independent of the probability distribution of the single samples (depends only on the variance)

 I distribution is important to estimate bounds for the signal

 Very often a Gaussian distribution models the experimental data the best



- the PSD is independent of the probability distribution of the single samples (depends only on the variance)
- distribution is important to estimate bounds for the signal
- very often a Gaussian distribution models the experimental data the best
- ► AWGN: additive white Gaussian noise



- the PSD is independent of the probability distribution of the single samples (depends only on the variance)
- distribution is important to estimate bounds for the signal
- very often a Gaussian distribution models the experimental data the best

► AWGN: additive white Gaussian noise



- the PSD is independent of the probability distribution of the single samples (depends only on the variance)
- distribution is important to estimate bounds for the signal
- very often a Gaussian distribution models the experimental data the best

► AWGN: additive white Gaussian noise

END OF MODULE 7.1

Digital Signa and Martin E 7.1

Paolo Prandoni and Martin E 7.1



Digital Signal Processing

Digital Signal Processing

Modula 7

Paolo Prandoni and Martin Modula 7

Overview:



- Uniform quantization and error and signal Processing Vetterli Vetterli Digli and Martin Digli and Digli

Overview:



- ▶ Uniform quantization and error analysis
 Dignal processing
 Vetterli
 Dignal processing
 And Martin Vetterli
 Dignal processing
 O 2013
 Paolo
 O 2013
 O 2013

Overview:



- Uniform quantization and error analysis
 Clipping, saturation, companding

Quantization



- ▶ digital devices can only deal with integers (b bits per sample)
- we need to map the range of atsignal onto alfinite set of values

 Dignal onto alfinite set of values

 irreversible loss of information (**)

 Paolo propression (**)

Quantization



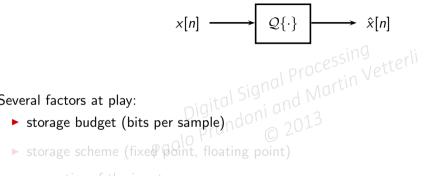
- ▶ digital devices can only deal with integers (b bits per sample)
- ▶ we need to map the range of a signal onto a finite set of values
- irreversible loss of information quantization noise

Quantization



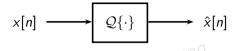
- ▶ digital devices can only deal with integers (b bits per sample)
- ▶ we need to map the range of a signal onto a finite set of values
- lacktriangle irreversible loss of information ightarrow quantization noise





Several factors at play:



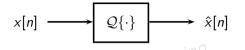


Several factors at play:

- ractors at play:

 Storage budget (bits per sample) doni and Martin vetterli atorage scheme (fixed point, floating parties of +1-





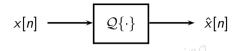
Several factors at play:

- storage scheme (fixed point, floating point)

 roperties of +□

 - - range
 - probability distribution



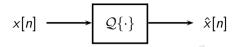


Several factors at play:

- storage scheme (fixed point, floating = 1 roperties of + 1 roperties of +

 - properties of the input
 - range



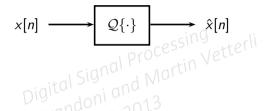


Several factors at play:

- storage scheme (fixed point, floating parties of +L

 - - range
 - probability distribution

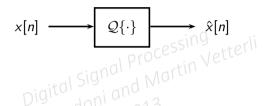




The simplest quantizer:

- each sample is encoded individually (hence scalar)
- each sample is quantized independently (memoryless quantization)
- each sample is encoded using *R* bits





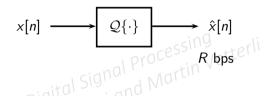
The simplest quantizer:

• each sample is encoded individually (hence scalar)

- caen sample is encoded individually (nence scalar
- each sample is quantized independently (memoryless quantization)

▶ each sample is encoded using *R* bits





The simplest quantizer:

- each sample is encoded individually (hence scalar)
- each sample is quantized independently (memoryless quantization)
- ▶ each sample is encoded using *R* bits



Assume input signal bounded: $A \le x[n] \le B$ for all n:

- \triangleright each sample quantized over 2^R possible values $\Rightarrow 2^R$ intervals.
- Digital Signal Processing

 Digital Signal Martin Vetterli

 Paolo Prandoni and Martin © 2013 each interval associated to a quantization value



Assume input signal bounded: $A \le x[n] \le B$ for all n:

- each sample quantized over 2^R possible values $\Rightarrow 2^R$ intervals.
- Digital Signal Processing

 Digital Signal Martin Vetterli

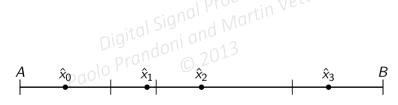
 Paolo Prandoni and Martin

 © 2013 each interval associated to a quantization value



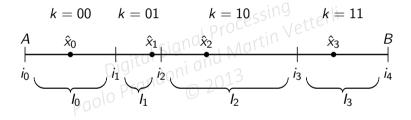
Assume input signal bounded: $A \le x[n] \le B$ for all n:

- each sample quantized over 2^R possible values $\Rightarrow 2^R$ intervals.
- each interval associated to a quantization value





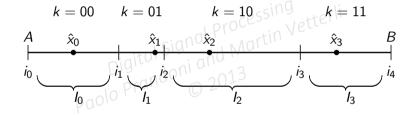
Example for R = 2:



- \triangleright what are the optimal interval boundaries i_k ?
- what are the optimal quantization values \hat{x}_k ?



Example for R = 2:



- \blacktriangleright what are the optimal interval boundaries i_k ?
- what are the optimal quantization values \hat{x}_k ?



$$e[n] = \mathcal{Q}\{x[n]\} - x[n] = \hat{x}[n] - x[n]$$

- ► model x[n] as a stochastic process Signal Processing Vetterli

 ► model error as a white noise segmence: © 2013

 error samples are paole lated



$$e[n] = \mathcal{Q}\{x[n]\} - x[n] = \hat{x}[n] - x[n]$$

- ▶ model x[n] as a stochastic process Signal Processing Vetterli
 ▶ model error as a white noise sequence: 2013
 ▶ error samples are uncorrelated
 - - all error samples have the same distribution



$$e[n] = \mathcal{Q}\{x[n]\} - x[n] = \hat{x}[n] - x[n]$$

- model x[n] as a stochastic process Signal Processing Vetterli
 model error as a white noise sequence:

 error samples are uncorrelated
 - - all error samples have the same distribution
 - we need statistics of the input to study the error



$$e[n] = \mathcal{Q}\{x[n]\} - x[n] = \hat{x}[n] - x[n]$$

- model x[n] as a stochastic process
 model error as a white noise sequence:

 error samples are uncorrelated

 - all error samples have the same distribution



$$e[n] = \mathcal{Q}\{x[n]\} - x[n] = \hat{x}[n] - x[n]$$

- model x[n] as a stochastic process
 model error as a white noise sequence:

 error samples are uncorrelated

 - all error samples have the same distribution
 - we need statistics of the input to study the error

Uniform quantization



- ► simple but very general case
- range is split into 2^R equal intervals of width $\Delta = (B A)2^{-R}$ $\begin{array}{c} \text{Digital Signal Processiny Vetterli} \\ \text{Digital Signal Martin Vetterli} \\ \text{Paolo Prandoni and } \end{array}$

Uniform quantization



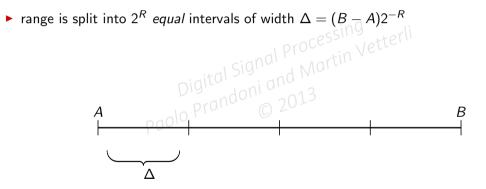
- simple but very general case

range is split into 2^R equal intervals of width $\Delta = (B - A)2^{-R}$ $\begin{array}{c} \text{Digital Signal Processiny Vetterli} \\ \text{Digital Signal Martin} \\ \text{Paolo Prandoni and Martin} \end{array}$

Uniform quantization



- simple but very general case





Mean Square Error is the variance of the error signal:

$$\sigma_{e}^{2} = E[|Q\{x[n]\} - x[n]|^{2}]_{hg}$$

$$= \int_{k=0}^{B} f_{k}(x) \langle y|^{2} \langle y|^{2} \rangle_{hg} d\tau$$

$$= \int_{k=0}^{B} f_{k}(x) \langle y|^{2} \langle y|^{2} \rangle_{hg} d\tau$$

$$= \int_{k=0}^{B} f_{k}(x) \langle y|^{2} \langle y|^{2} \rangle_{hg} d\tau$$

$$= \int_{k=0}^{B} f_{k}(x) \langle y|^{2} \rangle_{hg} d\tau$$

error depends on the probability distribution of the input



Mean Square Error is the variance of the error signal:

$$\sigma_{e}^{2} = E \left[|Q\{x[n]\} - x[n]|^{2} \right]_{Q}$$

$$= \int_{A}^{B} f_{x}(\tau) (Q\{\tau\} - \tau)^{2} d\tau$$

$$= \int_{k=0}^{B} f_{x}(\tau) (X[\tau] - \tau)^{2} d\tau$$

error depends on the probability distribution of the input



Mean Square Error is the variance of the error signal:

$$\sigma_e^2 = \mathbb{E}\left[|\mathcal{Q}\{x[n]\} - x[n]|^2\right]$$

$$= \int_A^B f_x(\tau)(\mathcal{Q}\{\tau\} - \tau)^2 d\tau$$

$$= \sum_{k=0}^{2^R+1} \int_{I_k} f_x(\tau)(\hat{x}_k - \tau)^2 d\tau$$



Mean Square Error is the variance of the error signal:

$$\sigma_e^2 = \mathbb{E}\left[|\mathcal{Q}\{x[n]\} - x[n]|^2\right]$$

$$= \int_A^B f_x(\tau)(\mathcal{Q}\{\tau\} - \tau)^2 d\tau$$

$$= \sum_{k=0}^{2^R+1} \int_{I_k} f_x(\tau)(\hat{x}_k - \tau)^2 d\tau$$

error depends on the probability distribution of the input



Uniform-input hypothesis:

$$f_{X}(\tau) = \frac{1}{B + A} ssing$$

$$pigital signal Martin Vetterli$$

$$paolo prandoni and Martin Vetterli$$

$$\sigma_{e}^{2} = \sum_{k=0}^{2^{R}-1} \int_{I_{k}} \frac{(\hat{x}_{k} - \tau)^{2}}{B - A} d\tau$$



Let's find the optimal quantization point by minimizing the error

$$\frac{\partial \sigma_{e}^{2}}{\partial \hat{x}_{m}} = \frac{\partial}{\partial \hat{x}_{m}} \sum_{k=0}^{2^{R}-1} \int_{I_{k}} \frac{(\hat{x}_{k} + \hat{x})^{2}}{B - A} d\tau^{k}$$

$$= \frac{(\hat{x}_{m} - \tau)^{2}}{B - A} \Big|_{A+m\Delta}^{A+m\Delta+\Delta}$$



Let's find the optimal quantization point by minimizing the error

$$\frac{\partial \sigma_e^2}{\partial \hat{x}_m} = \frac{\partial}{\partial \hat{x}_m} \sum_{k=0}^{2^{R}-1} \int_{I_k} \frac{(\hat{x}_k - \tau)^2}{B - A} d\tau$$

$$= \int_{I_m} \frac{2(\hat{x}_m - \tau)}{B - A} d\tau$$

$$= \frac{(\hat{x}_m - \tau)^2}{B - A} \Big|_{A+m\Delta}^{A+m\Delta+\Delta}$$



Let's find the optimal quantization point by minimizing the error

$$\frac{\partial \sigma_e^2}{\partial \hat{x}_m} = \frac{\partial}{\partial \hat{x}_m} \sum_{k=0}^{2^R - 1} \int_{I_k} \frac{(\hat{x}_k - \tau)^2}{B - A} d\tau$$

$$= \int_{I_m} \frac{2(\hat{x}_m - \tau)}{B - A} d\tau$$

$$= \frac{(\hat{x}_m - \tau)^2}{B - A} \Big|_{A + m\Delta}^{A + m\Delta + \Delta}$$



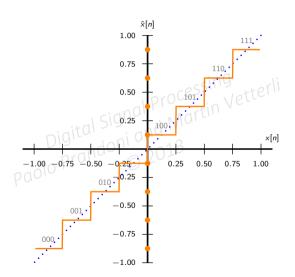
Minimizing the error:

$$\frac{\partial \sigma_e^2}{\partial \hat{x}_m} = 0 \quad \text{for } \hat{x}_m = A + m\Delta + \frac{\Delta}{2}$$

optimal quantization point is the interval's midpoint, for all intervals

Uniform 3-Bit quantization function







Quantizer's mean square error:

$$\sigma_{\rm e}^2 = \sum_{k=0}^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$



Quantizer's mean square error:

$$\sigma_e^2 = \sum_{k=0}^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^R \int_0^\Delta \frac{(\Delta/2 - \tau)^2}{B - A} d\tau$$

$$= \frac{\Delta^2}{12}$$



Quantizer's mean square error:

$$\sigma_e^2 = \sum_{k=0}^{2^R - 1} \int_{A + k\Delta}^{A + k\Delta + \Delta} \frac{(A + k\Delta + \Delta/2 - \tau)^2}{B - A} d\tau$$

$$= 2^R \int_0^\Delta \frac{(\Delta/2 - \tau)^2}{B - A} d\tau$$

$$= \frac{\Delta^2}{12}$$



error energy

$$\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B - A)/2^R$$

signal energy

 $\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B-A)/2^R$ $\begin{array}{c} \Delta = (B-A)/2^R \\ \Delta = ($

▶ signal to noise ratio

$$SNR_{dB} = 10 \log_{10} 2^{2R} \approx 6R dB$$



error energy

$$\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B - A)/2^R$$

signal energy

▶ signal to noise ratio

 $\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B-A)/2^R$ $\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B-A)/2^R$ Page 12 processing vetterly vetterly page 22 processing vetterly page 22 processing vetterly page 32 processing vetterly page 32

$$SNR_{dB} = 10 \log_{10} 2^{2R} \approx 6R dB$$



error energy

$$\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B-A)/2^R$$

signal energy

▶ signal to noise ratio

$$\mathsf{SNR} = 2^{2R}$$

$$SNR_{dB} = 10 \log_{10} 2^{2R} \approx 6R dB$$



error energy

$$\sigma_e^2 = \Delta^2/12, \qquad \Delta = (B-A)/2^R$$

signal energy

$$\sigma_x^2 = (B - A)^2 / 12$$

▶ signal to noise ratio

$$SNR = 2^{2R}$$

$$\mathsf{SNR}_{\mathsf{dB}} = \mathsf{10} \, \mathsf{log}_{\mathsf{10}} \, \mathsf{2}^{\mathsf{2R}} \approx \mathsf{6R} \; \mathsf{dB}$$

The "6dB/bit" rule of thumb



```
max SNR = 96dBi Vetterli

Sign and M

Paolo Prandoni and M

paolo Prandoni max SNR = 144dB
```

The "6dB/bit" rule of thumb



max SNR = 96dB

max SNR = 96dB

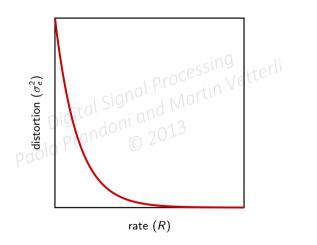
max SNR = 144dB

$$\mathsf{max}\ \mathsf{SNR} = \mathsf{96dB}$$

$$\max SNR = 144dE$$

Rate/Distortion Curve







If input is not bounded to [A, B]:

- ightharpoonup clip samples to [A, B]: linear distortion (can be put to good use in guitar effects!)
- smoothly saturate input: this simulates the saturation curves of analog electronics

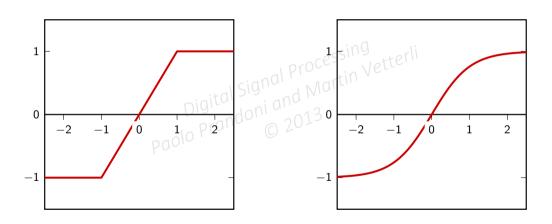


If input is not bounded to [A, B]:

- ightharpoonup clip samples to [A, B]: linear distortion (can be put to good use in guitar effects!)
- ▶ smoothly saturate input: this simulates the saturation curves of analog electronics

Clipping vs saturation







If input is not uniform:

Digital Signal Processing Vetterli Digital Signal Processing Vetterli Vett use uniform quantizer and accept increased error. For instance, if input is Gaussian:

Digital Signature
$$\sigma_e^2 = \frac{\sqrt{3}\pi}{21} \sigma^2 \Delta^2$$



If input is not uniform:

Digital Signal Processing $\frac{\sqrt{3}\pi}{\sqrt{21}}\sigma^2 \Delta^2$ or input distrib use uniform quantizer and accept increased error. For instance, if input is Gaussian:

$$\sigma_{\rm e}^2 = \frac{\sqrt{3}\pi}{2} \, \sigma^2 \, \Delta^2$$

- design optimal quantizer for input distribution, if known (Lloyd-Max algorithm)



If input is not uniform:

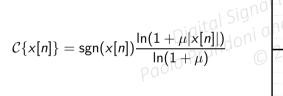
Digital Signal Processing $\frac{\sqrt{3}\pi}{\sqrt{21}} \sigma^2 \Delta^2$ or input distrib use uniform quantizer and accept increased error. For instance, if input is Gaussian:

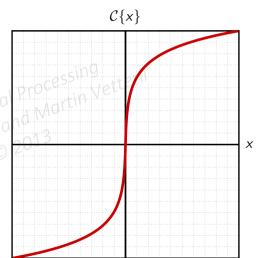
$$\sigma_e^2 = \frac{\sqrt{3}\pi}{2} \, \sigma^2 \, \Delta^2$$

- design optimal quantizer for input distribution, if known (Lloyd-Max algorithm)
- use "companders"

μ -law compander







END OF MODULE 7.2

Digital Signa Martin Et 7.2

Paolo Prandoni and Martin Prandoni and Martin Et 7.2



Digital Signal Processing

Digital Signal Processing

Module 7.3: A/D and D/A Conversion

Overview:



- ► Analog-to-digital (A/D) conversion

 Digital-to-analog (D/A) cpire sion and Martin Vetterli

 Paolo Prandoni and Martin C 2013

Overview:



- Digital-to-analog (D/A) conversion and Martin Vetterli

 Paolo Prando (2013)



- Paolo Prandoni and Martin Vetterli

 Paolo Prandoni and Martin Vetterli

 Paolo Prandoni and Martin Vetterli

 □ Digitalde Signal Processing

 Output

 Digitalde S

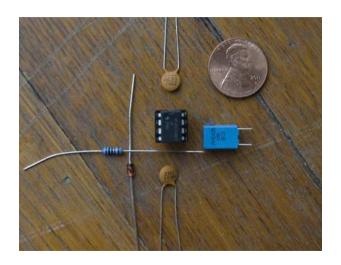


- quantization discretized amplitude Signal Processing Vetterli
 how is it done in practice? Prandoni and Processing Vetterli
 how is it done in practice? Prandoni © 2013



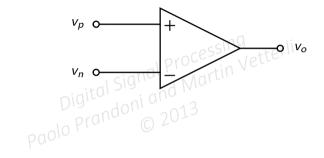
- Processing Vetterli
 Prandoni and Martin Vetterli





A tiny bit of electronics: the op-amp

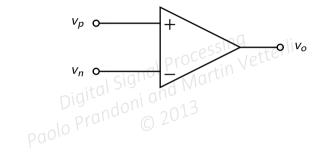




$$v_o = G(v_p - v_n)$$

A tiny bit of electronics: the op-amp





$$v_o = G(v_p - v_n)$$

The two key properties



- ▶ infinite input gain $(G \approx \infty)$
- Digital Signal Processing

 Digital Signal Processing

 Paolo Prandoni and Martin Vetterli

 Paolo Prandoni and Martin Vetterli

The two key properties



- ▶ infinite input gain ($G \approx \infty$)
- ... Digital Signal Processing

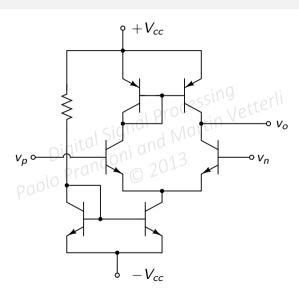
 Digital Signal Processing

 Paolo Prandoni and Martin Vetterli

 Paolo Prandoni and Martin Vetterli zero input current

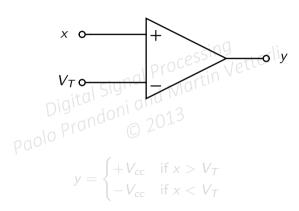
Inside the box





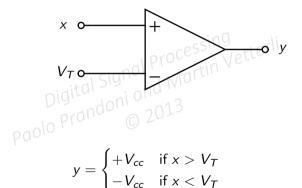
The op-amp in open loop: comparator





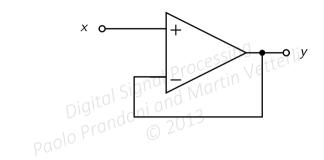
The op-amp in open loop: comparator





The op-amp in closed loop: buffer

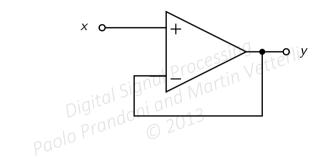




$$y = x$$

The op-amp in closed loop: buffer

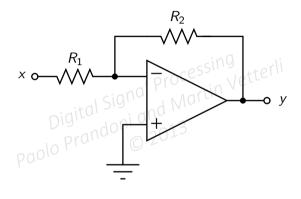




$$y = x$$

The op-amp in closed loop: inverting amplifier

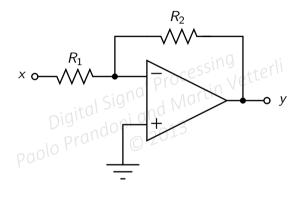




$$y = -(R_2/R_1)x$$

The op-amp in closed loop: inverting amplifier

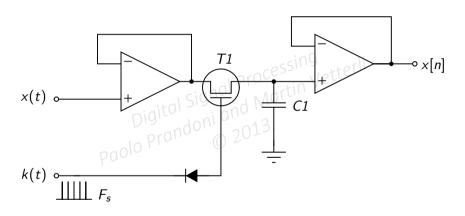




$$y = -(R_2/R_1)x$$

A/D Converter: Sample & Hold

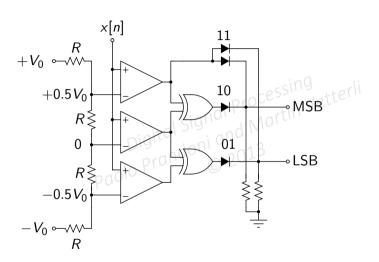




7.3 53

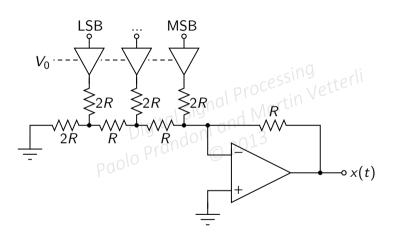
A/D Converter: 2-Bit Quantizer





D/A Converter





END OF MODULE 7.3

Digital Signal Martin Et 7.3

Paolo Prandoni and Martin Et 7.3

END OF MODULE 7

Digital Sign and Martin LE 7

Digital Sign and Martin LE 7

Digital Sign and Martin LE 7