

Digital Signal Processin Va Digital Signal Processin Va Paolo Prandoni and Martin Va © 2013 Digital Signal Processing

Module 2: Discrete-time signals

# Video Introduction Digital Sign and Marchine Paolo Prandoni and Marchine 2013

#### Module Overview:



- ► Module 2.1: discrete-time signals and operators

  Module 2.2: the discrete-time complex exponential
- ► Module 2.3: the Karplus-Strong algorithm 2013



Digital Signal Processing

Digital Signal Processing

Module 2.1. D:

### Overview:



- ▶ discrete-time signals
- signal classes
- Digital Signal Processing

  Digital Signal Processing

  Digital Signal Processing

  Office P elementary operators
- shifts
- energy and power

# Discrete-time signals have a long tradition...



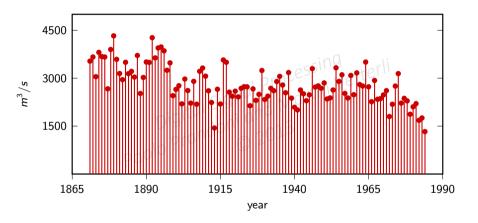
Meteorology (limnology): the floods of the Nile



Representations of flood data: circa 2500 BC

# Discrete-time signals have a long tradition...

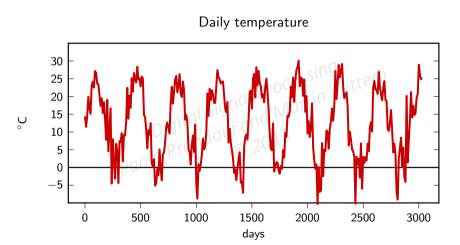




Representations of flood data: circa AD 2000

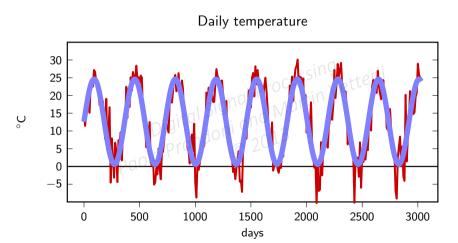
## Probably your first scientific experiment...





## Probably your first scientific experiment...

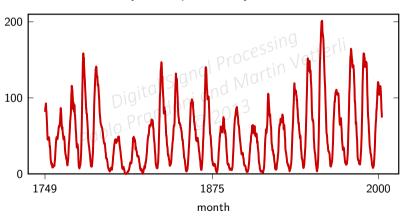




## Astronomy

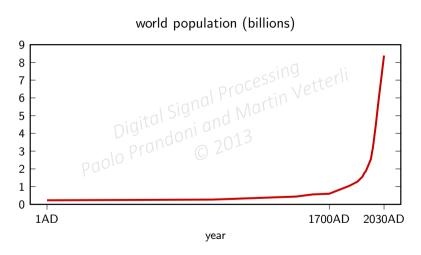


monthly solar spot activity, 1749 to 2003



## History and sociology

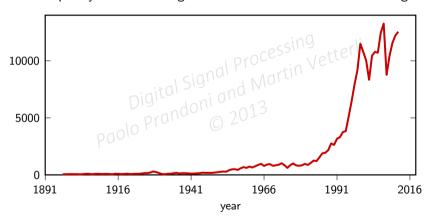




#### **Economics**



a purely man-made signal: the Dow Jones industrial average





- two-sided sequences: x: Zpigital Signal Processing

  two-sided sequences: x: Zpigital Signal Martin Vetterli

  n is dimension-less "Pigelo Prandoni and C 2013

  analysis: periodic prandoni c 2013



- two-sided sequences: x: Zpigital Signal Processing

  two-sided sequences: x: Zpigital Signal Martin Vetterli

  n is dimension-less "Fime" © 2013

  analysis: periodic processing processing vetterli

  analysis: periodic processing processing vetterli

  analysis: periodic processing processing vetterli

  analysis: periodic processing processing processing vetterli

  analysis: periodic processing processing processing vetterli

  by two-sided sequences: x: Zpigital Signal Processing processin

  - synthesis: stream of generated samples



- two-sided sequences:  $x : \mathbb{Z} \xrightarrow{p} C_{n} doni$  and X = 0 analysis: X = 0 anal

  - synthesis: stream of generated samples



- two-sided sequences:  $x : \mathbb{Z} \to \mathbb{C}$  and  $\mathbb{Z}$  is dimension-less "time"

  - synthesis: stream of generated samples



- x[n]two-sided sequences:  $x : \mathbb{Z} \to \mathbb{C}$  and  $\mathbb{Z}$  is dimension-less "time" 

  Palysis:  $\mathbb{R}^{n}$ 

  - analysis: periodic measurement



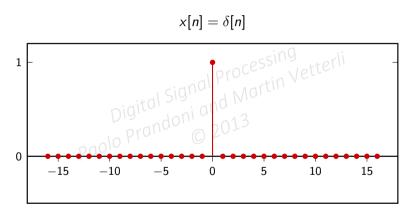
- x[n]two-sided sequences:  $x : \mathbb{Z} \to \mathbb{C}$  and  $\mathbb{Z}$  is dimension-less "time" 

  half in a signal processing which is a signal process which is a signal process. The signal process which is a signal process. The signal process which is a signal process

  - analysis: periodic measurement
  - synthesis: stream of generated samples

# The delta signal





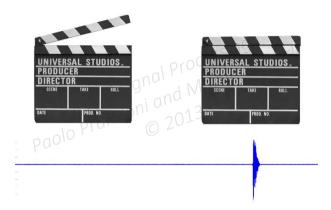
# How do you synchronize audio and video...



UNIVERSAL STUDIOS.		
PRODUCER		
DIRECTOR		
SCENE	TAKE	ROLL
DATE	PROD. NO.	

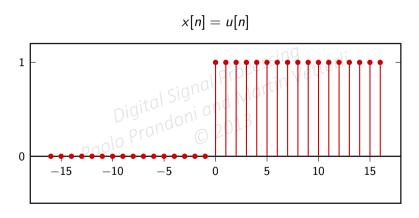
# How do you synchronize audio and video...





# The unit step





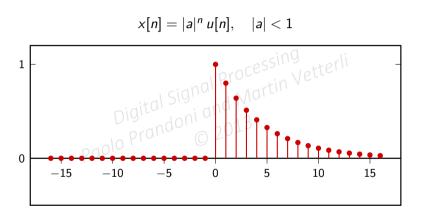
## The Frankenstein switch...





# The exponential decay





# How fast does your coffee get cold...





..1

# How fast does your coffee get cold...



Newton's law of cooling:

$$\frac{dT}{dt} = -c(T - T_{env})$$

$$T(t) = T_{env} + (T_0 + T_{env})e^{-ct}$$

In practice:

- ► must have convection only
- must have large conductivity

## How fast does your coffee get cold...



Newton's law of cooling:

$$\frac{dT}{dt} = -c(T - T_{\text{env}})$$

$$\frac{dT}{dt} = -c(T - T_{env})$$

$$T(t) = T_{env} + (T_0 + T_{env})e^{-ct}$$

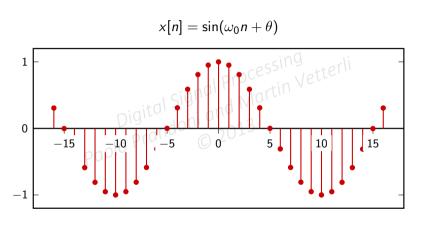
In practice:

must have convection only

must have large conductivity

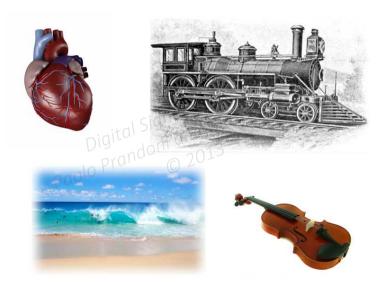
## The sinusoid





# Oscillations are everywhere!





..1



- ▶ finite-length
- ▶ infinite-length
- periodic
- finite-support

Digital Signal Processing

Digital Signal Processing

Martin Vetterli

2013

© 2013



- ▶ finite-length
- ▶ infinite-length
- periodic
- finite-support

Digital Signal Processing

Digital Signal Processing

Martin Vetterli

Paolo Prandoni and Martin

© 2013



- ▶ finite-length
- ▶ infinite-length
- periodic
- finite-support

Digital Signal Processing

Digital Signal Processing

Martin Vetterli

2013

2013



- ▶ finite-length
- ▶ infinite-length
- periodic
- finite-support

Digital Signal Processiny

Digital Signal Processiny

Digital Signal Processiny

Digital Signal Processiny

2013

## Finite-length signals



- ▶ sequence notation: x[n], n = 0, 1, ..., N processing

  ▶ vector notation:  $x = [x_0 \times_1 ..., x_m]$  and Martin

  ▶ practical entities, good for numerical packages (Matlab and the like)

# Finite-length signals



- ▶ sequence notation: x[n], n = 0, 1, ..., N processing

  ▶ vector notation:  $\mathbf{x} = [x_0 \ x_1 \ ... \ x_N + 1]^{\frac{1}{2}}$  and Martin

  ▶ practical entities, good for numerical packages (Matlab and the like)

## Finite-length signals



- sequence notation: x[n], n = 0, 1, ..., N = 10 cessing vetter!
   vector notation: x = [x<sub>0</sub> x<sub>1</sub> ... x<sub>N-1</sub>]<sup>T</sup>
   practical entities, good for numerical packages (Matlab and the like)

# Infinite-length signals



- ▶ sequence notation: x[n],  $n \in \mathbb{Z}$ abstraction, good for theoretical Signal Processing

  Paolo Prandoni and Martin Vetterli

  Paolo Prandoni and Prandoni a

# Infinite-length signals



- ▶ sequence notation: x[n],  $n \in \mathbb{Z}$ abstraction, good for theorems and Martin x[n]paolo Prandoni and Martin x[n]paolo Prandoni x[n]

# Periodic signals



- ► N-periodic sequence:  $\tilde{x}[n] = \tilde{x}[n+kN]$ ,  $n,k,N \in \mathbb{Z}$  vetterli

  ► same information as finite-length of length M Martin

   "natural" bridge between finite and infinite lengths paolo

#### Periodic signals



- ▶ N-periodic sequence:  $\tilde{x}[n] = \tilde{x}[n+kN], \quad n, k, N \in \mathbb{Z}$ ▶ same information as finite-length of length N
- "natural" bridge between finite and infinite lengths

#### Periodic signals



- ▶ N-periodic sequence:  $\tilde{x}[n] = \tilde{x}[n+kN], \quad n,k,N \in \mathbb{Z}$ ▶ same information as finite-length of length N
- "natural" bridge between finite and infinite lengths

#### Finite-support signals



► Finite-support sequence:

$$\bar{x}[n] = \begin{cases} x[n] & \text{if } 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$

$$n \in \mathbb{Z}$$

- same information as finite-length of length Λ
- ▶ another bridge between finite and infinite lengths

#### Finite-support signals



► Finite-support sequence:

$$\bar{x}[n] = \begin{cases} x[n] & \text{if } 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$$

$$n \in \mathbb{Z}$$

- ▶ same information as finite-length of length *N*
- ▶ another bridge between finite and infinite lengths

#### Finite-support signals



Finite-support sequence:

$$\bar{x}[n] = \begin{cases} x[n] & \text{if } 0 \leq n < N \text{ Vetterli} \\ 0 & \text{if } 0 \leq n \leq N \end{cases}$$

$$n \in \mathbb{Z}$$
otherwise

- ▶ same information as finite-length of length N
- another bridge between finite and infinite lengths



scaling:

$$y[n] = \alpha x[n]$$

Digital Shinal Processing Vetterli

Digital Shinal Processing Vetterli

Paolo Prandoni and 2013

product:

$$y[n] = x[n] \cdot z[n]$$

 $\triangleright$  shift by k (delay):

$$y[n] = x[n-k]$$



scaling:

$$y[n] = \alpha x[n]$$

sum:

$$y[n] = ax[n]$$

$$processing Vetterli Ve$$

product:

$$y[n] = x[n] \cdot z[n]$$

 $\triangleright$  shift by k (delay):

$$\sqrt{[n]} = x[n-k]$$



scaling:

$$y[n] = \alpha x[n]$$

sum:

$$y[n] = \alpha x[n]$$

$$y[n] = x[n] + z[n]$$

$$y[n] = x[n] \cdot z[n]$$

$$y[n] = x[n] \cdot z[n]$$

product:

$$y[n] = x[n] \cdot z[n]$$

 $\triangleright$  shift by k (delay):

$$y[n] = x[n-k]$$



scaling:

$$y[n] = \alpha x[n]$$

sum:

$$y[n] = \alpha x[n]$$

$$y[n] = x[n] + z[n]$$

$$y[n] = x[n] \cdot z[n]$$

$$y[n] = x[n] \cdot z[n]$$

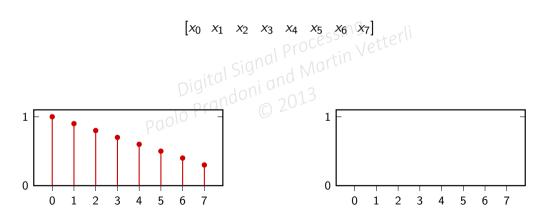
product:

$$y[n] = x[n] \cdot z[n]$$

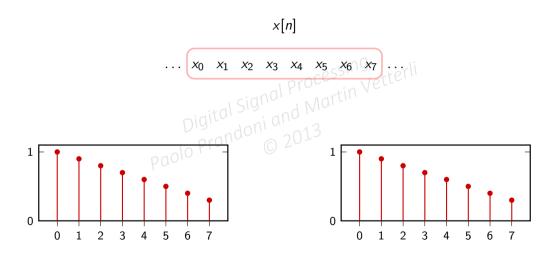
▶ shift by *k* (delay):

$$y[n] = x[n-k]$$

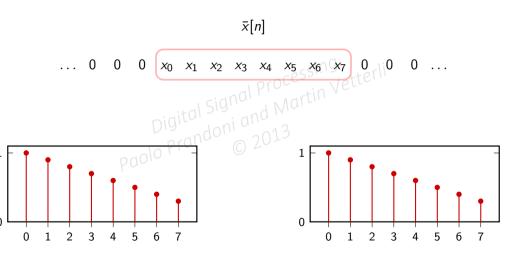




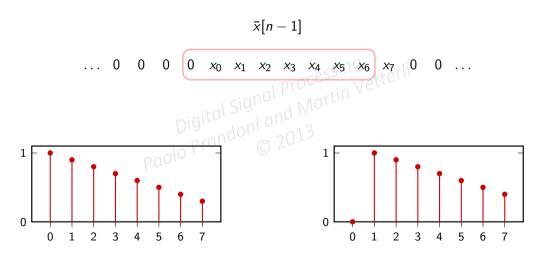




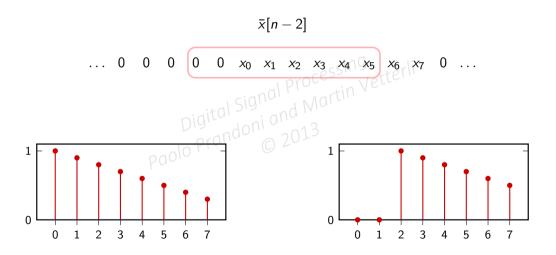




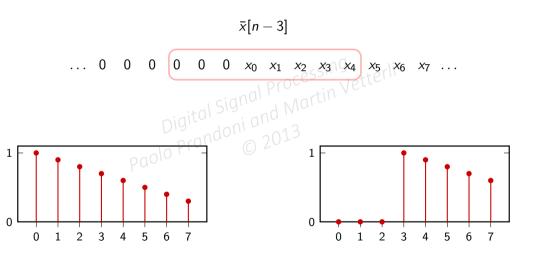




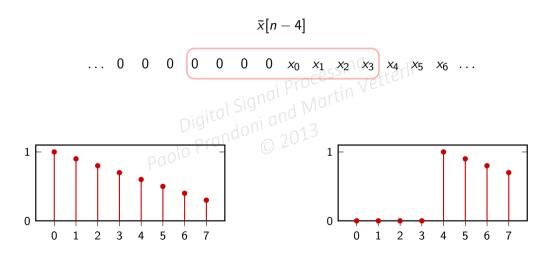




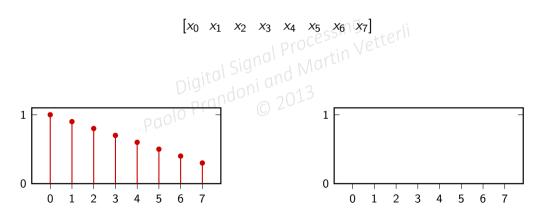




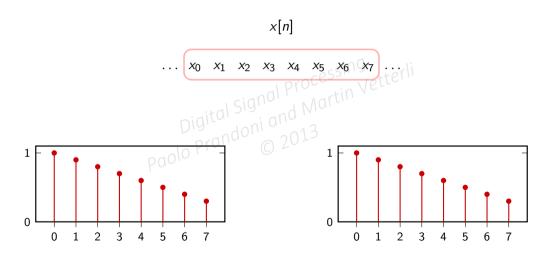




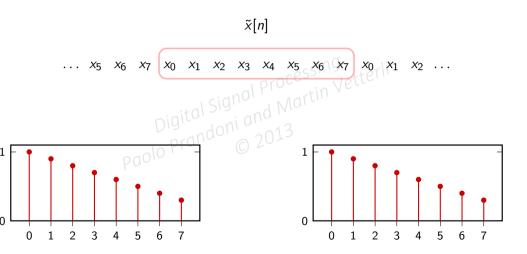




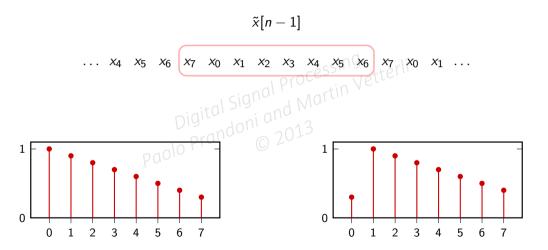




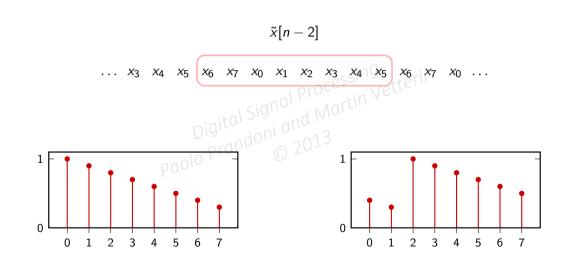




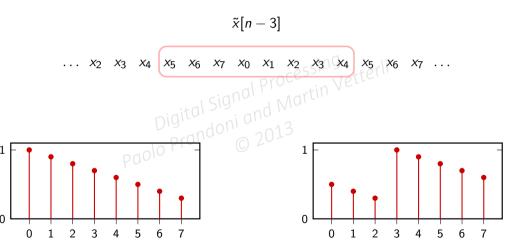




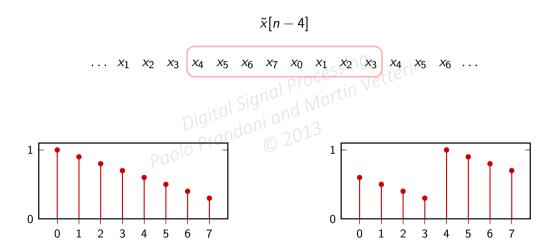












#### Energy and power



$$E_{x} = \sum_{\substack{n \in \mathbb{Z} \\ \text{proposed prandoni}}}^{\infty} |x[n]|^{2} \sin^{9}$$

$$\text{Digital Signard Martin Vetterli}$$

$$\text{Paolo Prandoni} \text{ and Martin Vetterli}$$

$$\text{Paolo P}_{x} = \lim_{N \to \infty} \frac{\mathbb{C}_{1} 2013}{2N+1} \sum_{n=-N}^{N} |x[n]|^{2}$$

# Energy and power



$$E_{x} = \sum_{\substack{n = +\infty \\ \text{pigital Signard Martin Vetterli}}}^{\infty} |x[n]|^{2} \sin \theta$$

$$\text{pigital Signard Martin Vetterli}$$

$$\text{prandoni and } \sum_{n=-N}^{N} |x[n]|^{2}$$

# Energy and power: periodic signals



$$E_{\tilde{x}} = \infty$$

$$Processing Processing Proce$$

# Energy and power: periodic signals



$$E_{\tilde{\mathbf{x}}} = \infty_{\text{cessing}}$$
Digital Signal Processing Vetterli Processing Vetterli Processing P

END OF MODULE 2.1

Digital Signa and Martill Et 2.1

Paolo Prandoni and Martill Et 2.1



Digital Signal Processing

Digital Signal Processing

Module 2.2: the analogous processing

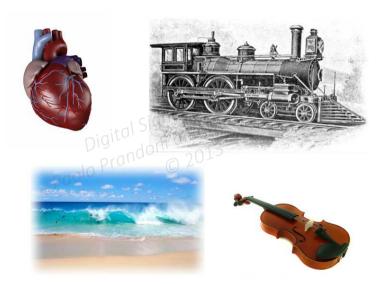
#### Overview:



- the complex exponential
- periodicity
- wagonwheel effect and maximum "speed" 2013
  'igital and real-world frequency
- digital and real-world frequency

# Oscillations are everywhere





# The oscillatory heartbeat



#### Ingredients:

- an amplitude A (units depending on underlying measurement)

  a trigonometric function paolo Prando © 2013

e.g. 
$$x[n] = A\cos(\omega n + \phi)$$

# The oscillatory heartbeat



#### Ingredients:

- an amplitude A (units depending on underlying measurement)

  a trigonometric function of Prando © 2013

e.g. 
$$x[n] = A\cos(\omega n + \phi)$$

## The oscillatory heartbeat



#### Ingredients:

- $\triangleright$  a frequency  $\omega$  (units: radians)
- an initial phase  $\phi$  (units: radians) an amplitude A (units depending on underlying measurement)
- a trigonometric function lo Prando © 2013

e.g. 
$$x[n] = A\cos(\omega n + \phi)$$

## The oscillatory heartbeat



#### Ingredients:

- $\triangleright$  a frequency  $\omega$  (units: radians)
- $\blacktriangleright$  an initial phase  $\phi$  (units: radians)
- an initial phase  $\phi$  (units: radians) an amplitude A (units depending on underlying measurement)
- a trigonometric function prandor

e.g. 
$$x[n] = A\cos(\omega n + \phi)$$

## The oscillatory heartbeat



#### Ingredients:

- $\triangleright$  a frequency  $\omega$  (units: radians)
- $\blacktriangleright$  an initial phase  $\phi$  (units: radians)
- an initial phase  $\phi$  (units: radians) an amplitude A (units depending on underlying measurement)
- a trigonometric function of Prandol

e.g. 
$$x[n] = A\cos(\omega n + \phi)$$



the trigonometric function of choice in DSP is the complex exponential:

$$x[n] = Ae^{j(\omega n + \phi)} \text{ and } Ae^{j(\omega n + \phi)}$$

$$x[n] = Ae^{j(\omega n + \phi)} \text{ and } Ae^{j(\omega n + \phi)}$$

$$x[n] = A[\cos(\omega n + \phi) + j\sin(\omega n + \phi)]$$

# Why complex exponentials?



- makes sense: sines and cosines always go togethers vetterli
   simpler math: trigonometry becomes algebral Martin
   we can use complex numbers in digital systems

# Why complex exponentials?



- makes sense: sines and cosines always go together
   simpler math: trigonometry becomes algebra
- we can use complex numbers indigital systems

## Why complex exponentials?



- makes sense: sines and cosines always go together
   simpler math: trigonometry becomes algebra
- we can use complex numbers in digital systems



$$\cos(\omega n + \phi) = a\cos(\omega n) + b\sin(\omega n), e^{-5\sin\theta}$$

$$a = \cos\phi, b = \sin\phi$$

$$b = a\cos(\omega n) + b\sin(\omega n), e^{-5\sin\theta}$$

$$a = \cos\phi, b = \sin\phi$$

$$a = \cos\phi, b = \cos\phi, b = \sin\phi$$

$$a = \cos\phi, b = \cos\phi, b = \sin\phi$$

$$a = \cos\phi, b = \cos\phi, b = \cos\phi, b = \cos\phi$$

$$a = \cos\phi, b = \cos\phi, b = \cos\phi, b = \cos\phi, b = \cos\phi$$

$$a = \cos\phi, b =$$



$$\cos(\omega n + \phi) = a\cos(\omega n) + b\sin(\omega n), \cos(\omega n) + b\cos(\omega n) +$$

- each sinusoid is always a sum of sine and cosine
- we have to remember complex trigonometric formulas
- we have to carry more terms in our equations



$$\cos(\omega n + \phi) = a\cos(\omega n) + b\sin(\omega n), \qquad a = \cos\phi, \quad b = \sin\phi$$

- each sinusoid is always a sum of sine and cosine
- we have to remember complex trigonometric formulas
- we have to carry more terms in our equations



$$\cos(\omega n + \phi) = a\cos(\omega n) + b\sin(\omega n), \qquad a = \cos\phi, \quad b = \sin\phi$$

- each sinusoid is always a sum of sine and cosine
- ▶ we have to remember complex trigonometric formulas
- ▶ we have to carry more terms in our equations





Example: change the phase of a pure cosine 
$$\operatorname{Re}\{e^{j(\omega n+\phi)}\} = \operatorname{Re}\{e^{j\omega n}e^{j\phi}\} \text{ etterli}$$
 
$$\bullet \text{ sine and cosine "live" together and one and the phase shift is simple multiplication of 2013}$$
 
$$\bullet \text{ notation is simpler}$$



$$\operatorname{Re}\{e^{j(\omega n+\phi)}\} = \operatorname{Re}\{e^{j\omega n}e^{j\phi}\}$$
 sine and cosine "live" together below the phase shift is simple multiplication and the phase shift is simple multiplication.

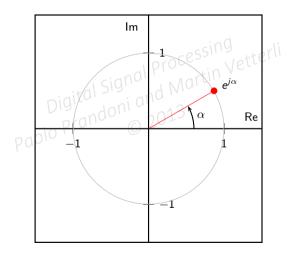


$$\operatorname{Re}\{e^{j(\omega n+\phi)}\} = \operatorname{Re}\{e^{j\omega n}e^{j\phi}\}$$
 sine and cosine "live" together below the phase shift is simple multiplication and the phase shift is simple multiplication.

- notation is simpler

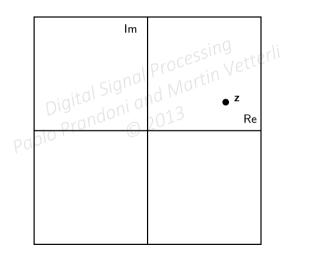


$$e^{j\alpha} = \cos\alpha + j\sin\alpha$$



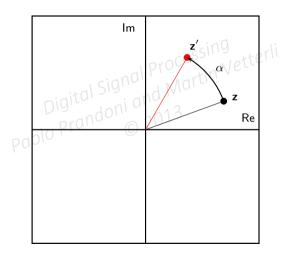


### z: point on the complex plane



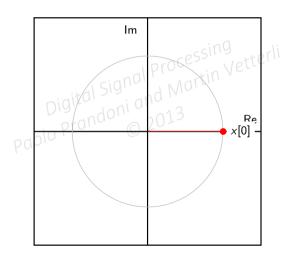


rotation:  $\mathbf{z}' = \mathbf{z} \, e^{j\alpha}$ 



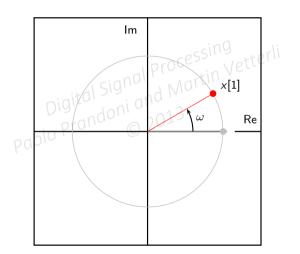


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



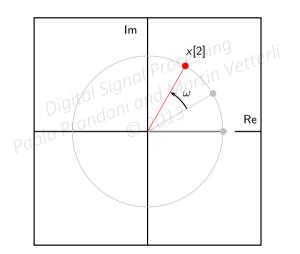


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



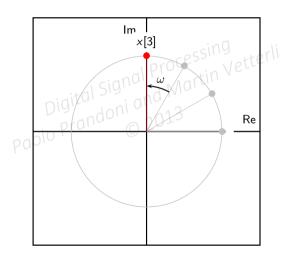


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



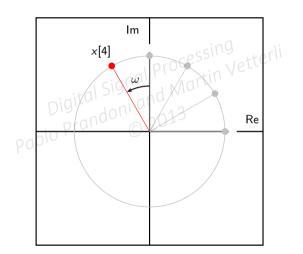


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



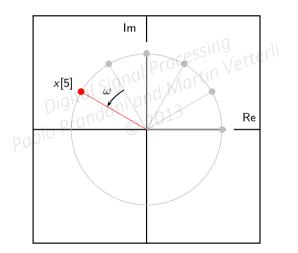


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



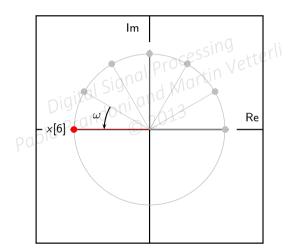


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



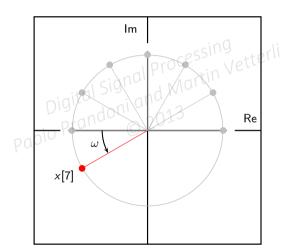


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



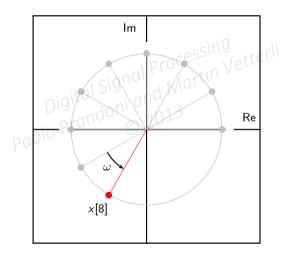


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



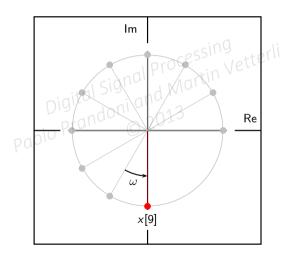


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



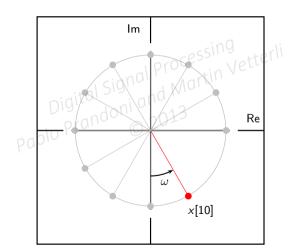


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



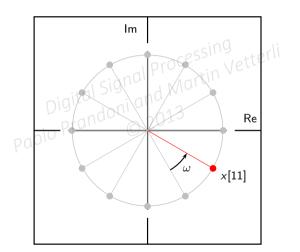


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



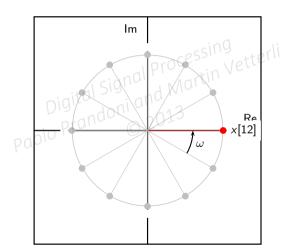


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



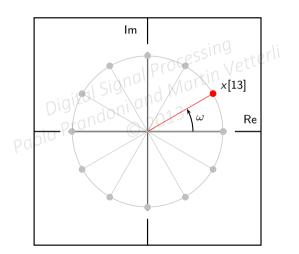


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



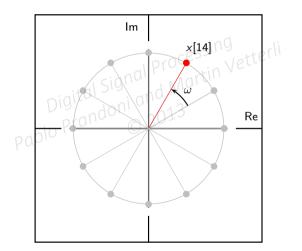


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$





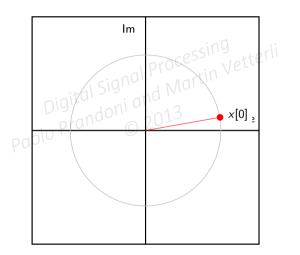
$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



## Initial phase



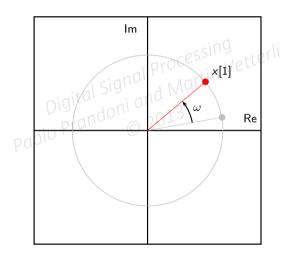
$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



# Initial phase



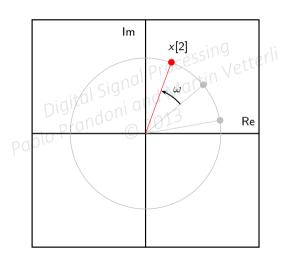
$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



## Initial phase

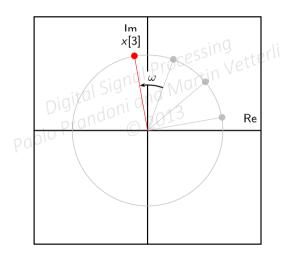


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



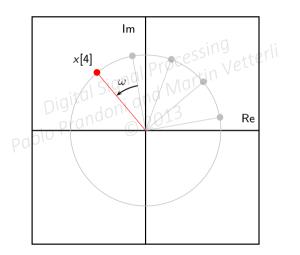


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



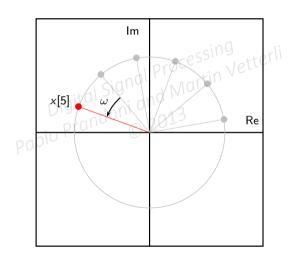


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



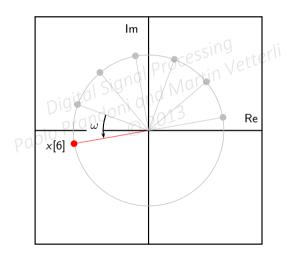


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



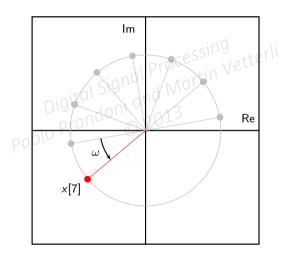


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



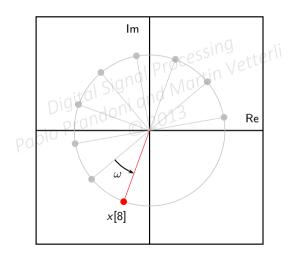


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



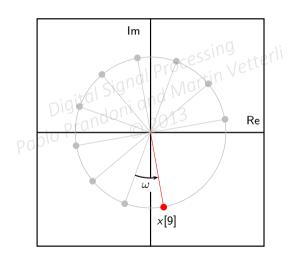


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



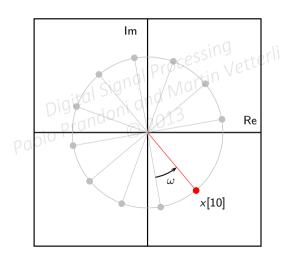


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



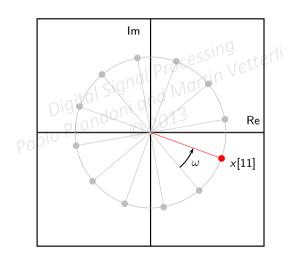


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



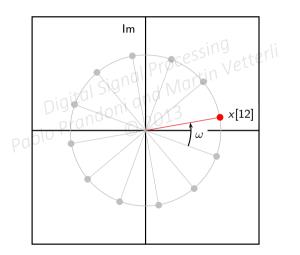


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



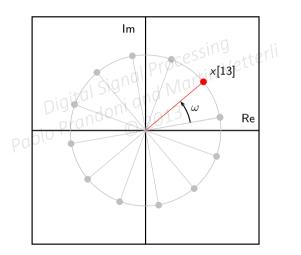


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



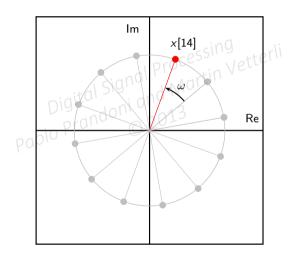


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



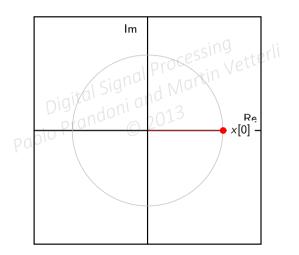


$$x[n] = e^{j(\omega n + \phi)};$$
  $x[n+1] = e^{j\omega}x[n],$   $x[0] = e^{j\phi}$ 



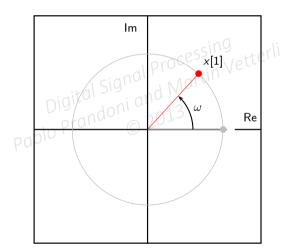


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



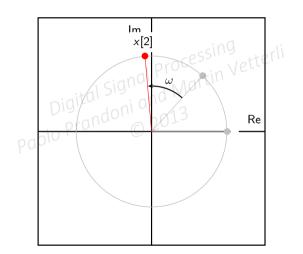


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



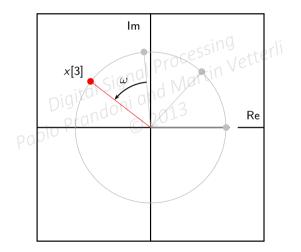


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



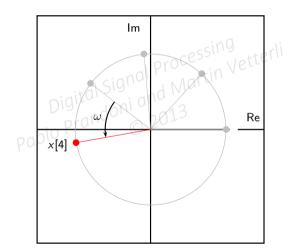


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



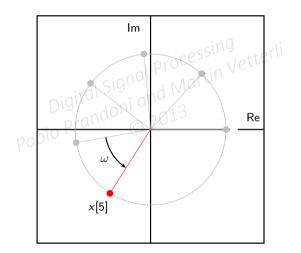


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



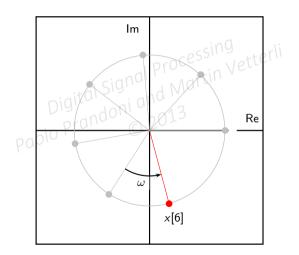


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



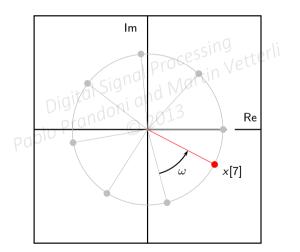


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



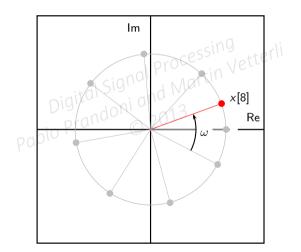


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



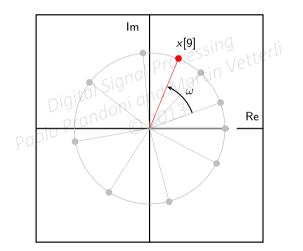


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



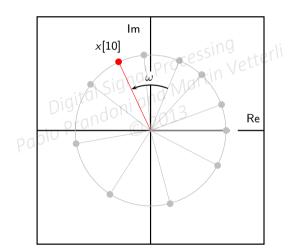


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



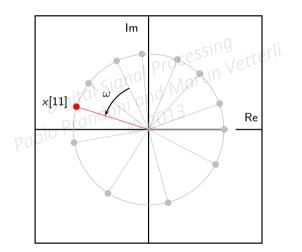


$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



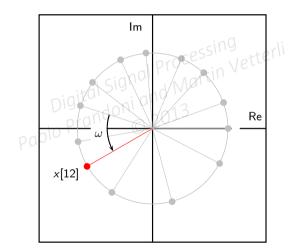


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



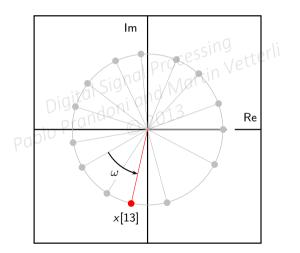


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$



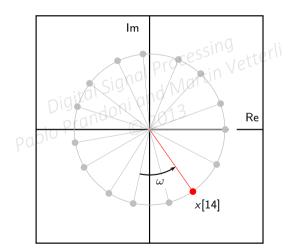


$$x[n] = e^{j\omega n}; \qquad x[n+1] = e^{j\omega}x[n]$$





$$x[n] = e^{j\omega n};$$
  $x[n+1] = e^{j\omega}x[n]$ 



## Periodicity



$$e^{j\omega n}$$
 periodic  $\iff \omega = \frac{M}{N} 2\pi, M, N \in \mathbb{N}$ 

Digital Signal Martin

Digital Signal Martin

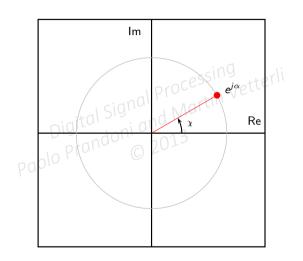
 $0 = e^{i(\Omega_2 k\pi)} \quad \forall k \in \mathbb{N}$ 

## Periodicity

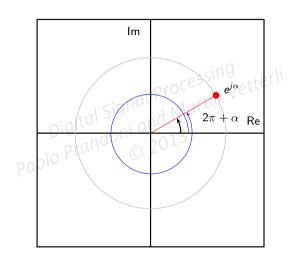


$$e^{j\omega n}$$
 periodic  $\iff \omega = \frac{M}{N} 2\pi, M, N \in \mathbb{N}$ 

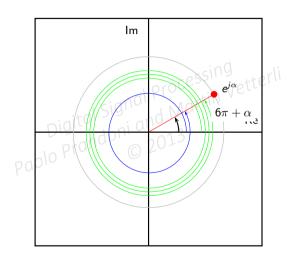




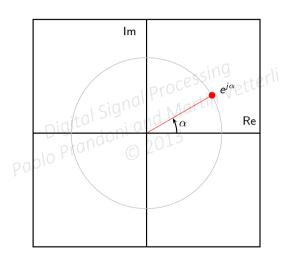




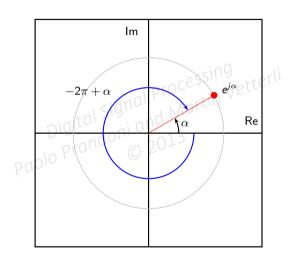












# How "fast" can we go?

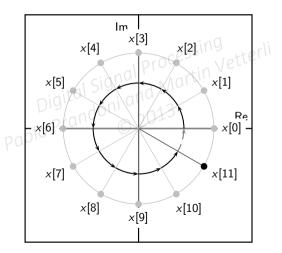


Digital Signal Processin Vetterii Paolo Prandoni and Martin Vetterii © 2013

### How "fast" can we go?



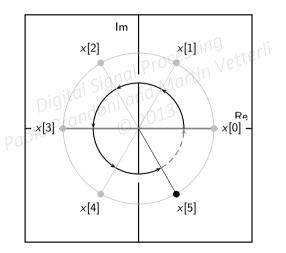
$$\omega=2\pi/12$$



2.2 46

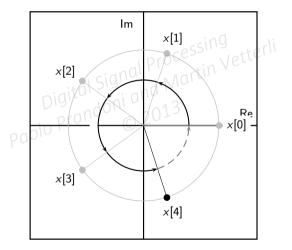


$$\omega = 2\pi/6$$



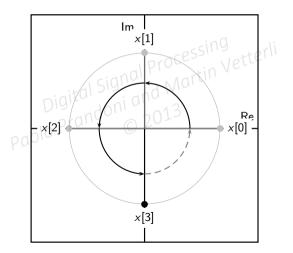


$$\omega=2\pi/5$$



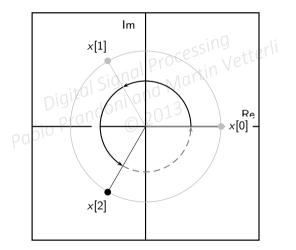


$$\omega = 2\pi/4$$





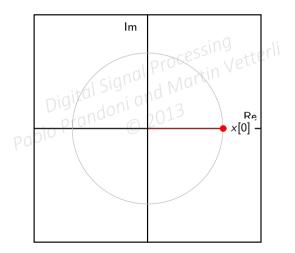
$$\omega = 2\pi/3$$



2.2 50

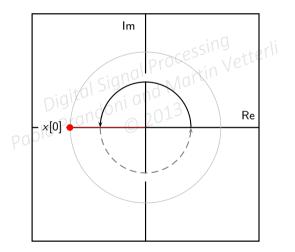


$$\omega = 2\pi/2 = \pi$$



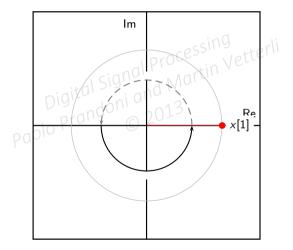


$$\omega = 2\pi/2 = \pi$$



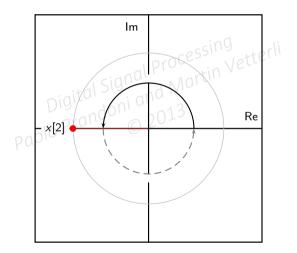


$$\omega = 2\pi/2 = \pi$$



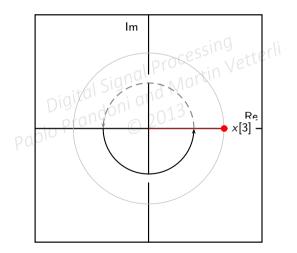


$$\omega = 2\pi/2 = \pi$$



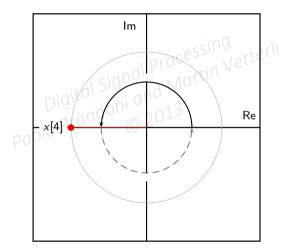


$$\omega = 2\pi/2 = \pi$$



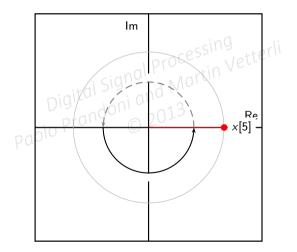


$$\omega = 2\pi/2 = \pi$$





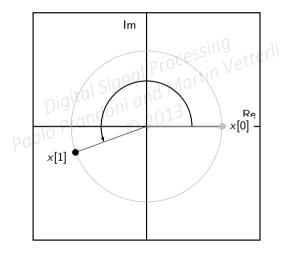
$$\omega = 2\pi/2 = \pi$$



## What if we go "faster"?



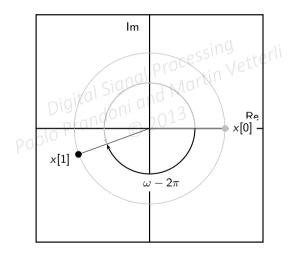
$$\pi < \omega < 2\pi$$



## What if we go "faster"?

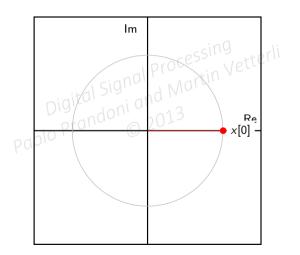






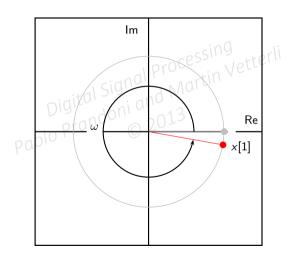


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



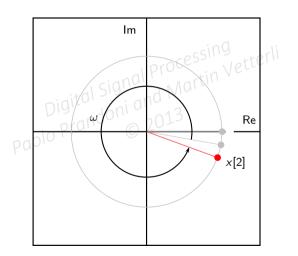


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



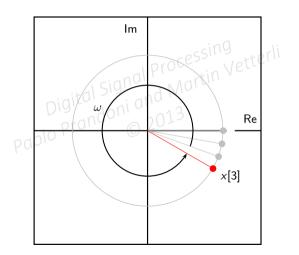


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



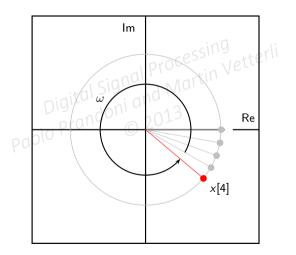


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



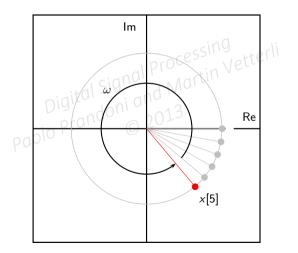


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



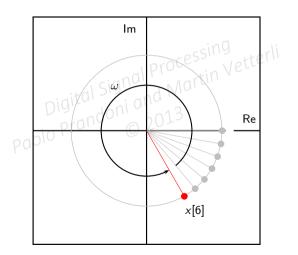


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



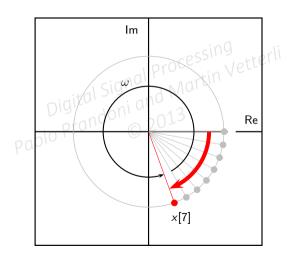


$$\omega = 2\pi - \alpha$$
,  $\alpha$  small





$$\omega = 2\pi - \alpha$$
,  $\alpha$  small



# The wagonwheel effect



Digital Signal Processing VetterII

Paolo Prandoni and Martin VetterII

Paolo Prandoni © 2013



#### Discrete time:

- periodicity: how many samples before pattern repeats Vetterli

  "Real world":

   periodicity: how paologoonds before pattern repeats

   periodicity: how paologoonds before pattern repeats



- Discrete time:

  - periodicity: how many samples before pattern repeats

    "Real world":

     periodicity: how paoloeconds before pattern repeats

    "Real world":

     periodicity: how paoloeconds before pattern repeats



- Discrete time:

  - periodicity: how many samples before pattern repeats



- Discrete time:

  - periodicity: how many samples before pattern repeats
- "Real world":

  - frequency measured in Hz  $(s^{-1})$



- Discrete time:

  - periodicity: how many samples before pattern repeats
- "Real world":
  - periodicity: how many seconds before pattern repeats
  - frequency measured in Hz  $(s^{-1})$

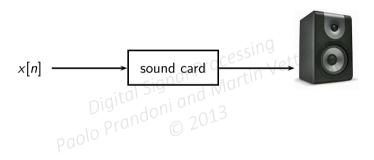


- Discrete time:

  - periodicity: how many samples before pattern repeats
- "Real world":
  - periodicity: how many seconds before pattern repeats
  - frequency measured in Hz  $(s^{-1})$

### How your PC plays sounds

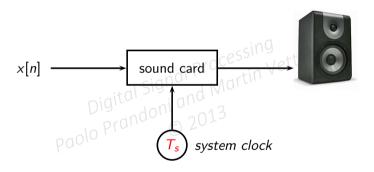




2.2 5

#### How your PC plays sounds







- $\triangleright$  set  $T_s$ , time in seconds between samples
- ► real world frequency:



- $\triangleright$  set  $T_s$ , time in seconds between samples
- ▶ periodicity of M samples  $\longrightarrow$  periodicity of  $MT_s$  seconds
- real world frequency: Digital Signal Martin Paolo Prandoni and  $\varphi = \frac{2013}{MT_c}$

22



- $\triangleright$  set  $T_s$ , time in seconds between samples
- ▶ periodicity of M samples  $\longrightarrow$  periodicity of  $MT_s$  seconds y: Digital Signation and Martin Paolo Prandoni and  $\frac{2013}{9}$
- ► real world frequency:

$$G = \frac{1}{MT_s}$$

END OF MODULE 2.2

Digital Signand Martin Et 2.2

Paolo Prandoni and Martin Et 2.2

Paolo Prandoni and Martin Et 2.2



Digital Signal Processing

Digital Signal Processing

Module 2.3: the Karplus-Strong algorithm

#### Overview:



- DSP building blocks
- moving averages and simple feedback loops d Martin Vetterli

  sound synthesizer

  Paolo Prandon 2013

#### Overview:



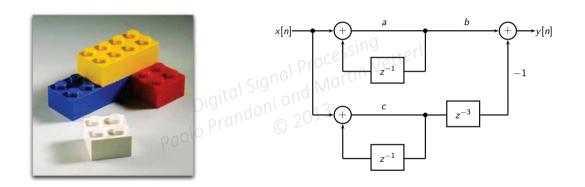
- Recursion: Revisiting your bank account and Martin Vetterli

  Building a simple roc ▶ DSP as Lego: The fundamental building blocks

  - Building a simple recursive synthesizer © 2013
  - Examples of sounds

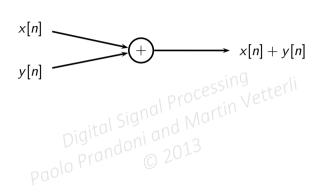
# DSP as Lego





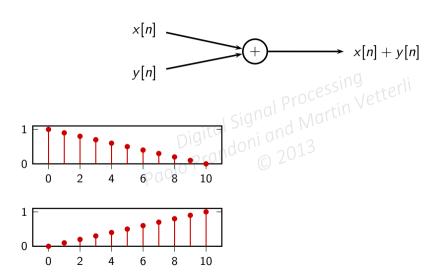
# Building Blocks: Adder





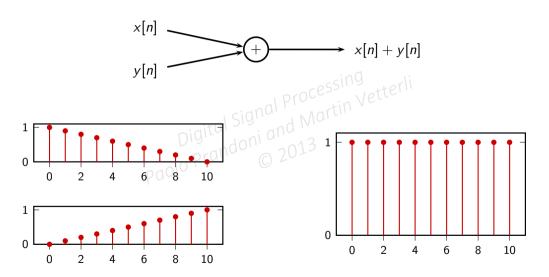
## Building Blocks: Adder





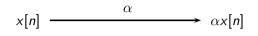
#### Building Blocks: Adder





# Building Blocks: Multiplier

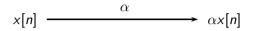


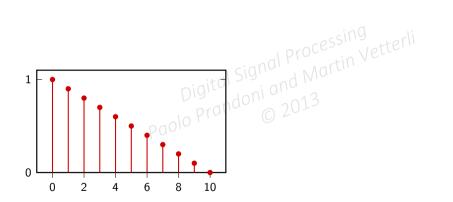


Digital Signal Processing Vetterli Paolo Prandoni and Martin Vetterli © 2013

# Building Blocks: Multiplier

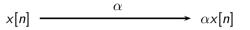


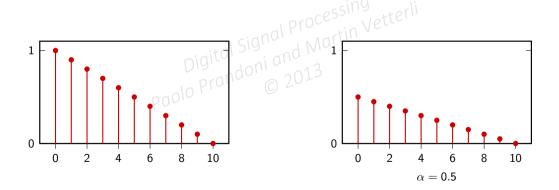




## Building Blocks: Multiplier

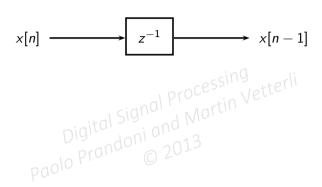






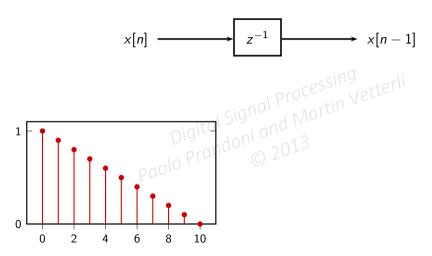
# Building Blocks: Unit Delay





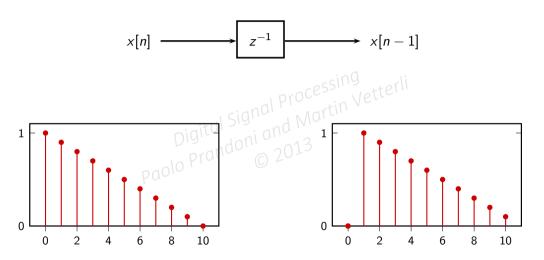
# Building Blocks: Unit Delay





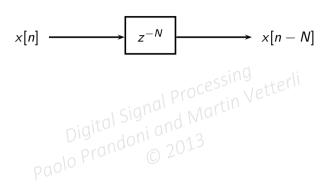
### Building Blocks: Unit Delay





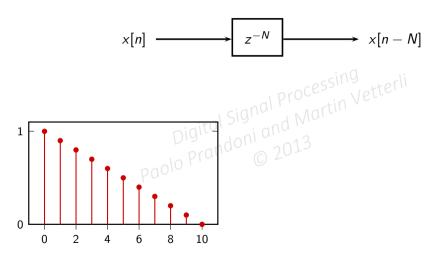
# Building Blocks: Arbitrary Delay





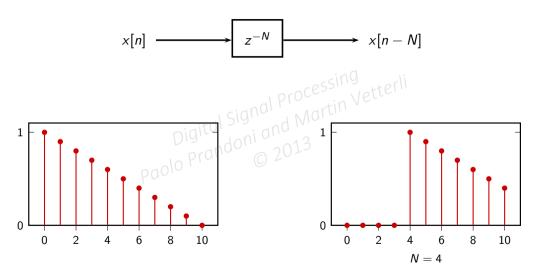
# Building Blocks: Arbitrary Delay





### Building Blocks: Arbitrary Delay





# The 2-point Moving Average



simple average:

$$m = \frac{a + b}{2} \sin y$$
 etterli

 $m = \frac{a+b}{P^{r}2} \text{ sing }$ moving average: take a "localital verage and Martin Vetterli Paolo Prandoni and Martin Vetterli Paolo Prandoni 2

paolo Pranaor 
$$y[n]$$
  $x[n-1]$ 

## The 2-point Moving Average



simple average:

$$m=\frac{a+b}{2}$$

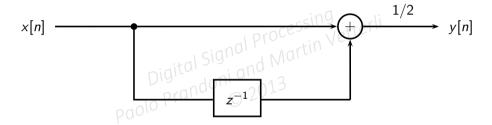
simple average: 
$$m = \frac{a+b}{2} \text{ yetterli}$$

$$m = \frac{a+b}{2} \text{ wetterli}$$

$$y[n] = \frac{x[n] + x[n-1]}{2}$$

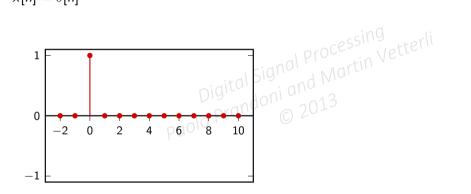
# The 2-point Moving Average Using Lego





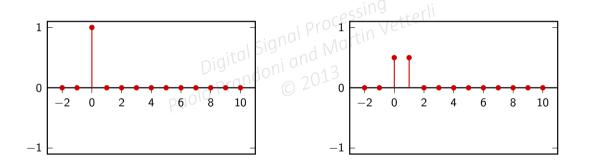


$$x[n] = \delta[n]$$



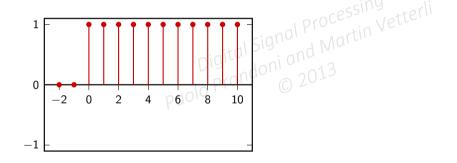


$$x[n] = \delta[n]$$



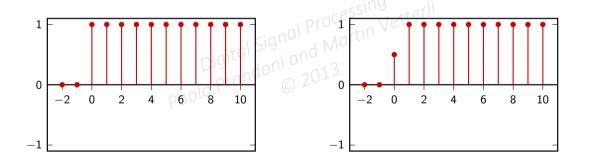


$$x[n] = u[n]$$



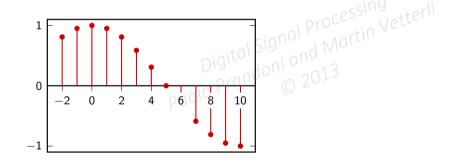


$$x[n] = u[n]$$



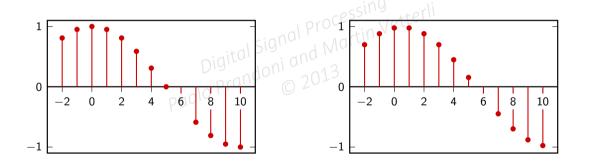


$$x[n] = \cos(\omega n), \quad \omega = \pi/10$$



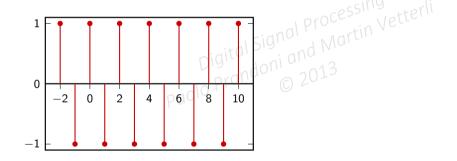


$$x[n] = \cos(\omega n), \quad \omega = \pi/10$$



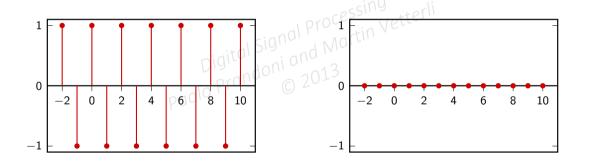


$$x[n] = \cos(\omega n), \quad \omega = \pi$$



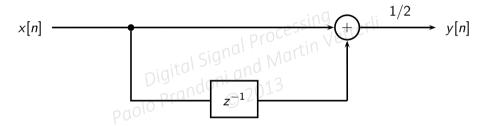


$$x[n] = \cos(\omega n), \quad \omega = \pi$$



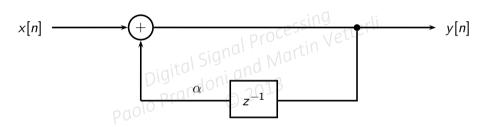
#### What if we reverse the loop?





#### What if we reverse the loop?







- ► constant interest/borrowing rate of 5% per year

- ► constant interest/borrowing rate of 5% per year interest accrues on Dec 31

   deposits/withdrawals during year  $n_i$  of  $n_i$



- ► constant interest/borrowing rate of 5% per year

   interest accrues on Dec 31

   deposits/withdrawals during year  $n_i$  of following production (2013)• balance at year  $n_i$ :

   y[n] = 1.05 y[n-1] + x[n]

$$y[n] = 1.05 y[n-1] + x[n]$$



- deposits/withdrawals during year n: x[n]and Martin

  balance at year n:

balance at year 
$$n$$
:
$$y[n] = 1.05 y[n-1] + x[n]$$

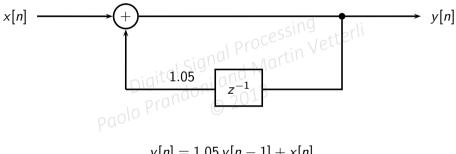


- deposits/withdrawals during year n: x[n]and Martin Vetterli
  balance at year n:

$$y[n] = 1.05 y[n-1] + x[n]$$

#### First-order recursion





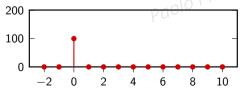
$$y[n] = 1.05 y[n-1] + x[n]$$



$$x[n] = 100 \delta[n]$$

- v[0] = 100

- ► y[2] = 110.25, y[3] = 115.7625 etcsignal Processing ► In general:  $y[n] = (1.05)^n Di@u[n] doni and Martin Control Co$



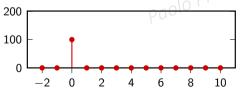


$$x[n] = 100 \delta[n]$$

- y[0] = 100
- ► y[2] = 110.25, y[3] = 115.7625 ets ignal Processing Vetterli

  ► In general:  $y[n] = (1.05)^n \text{Dig}_{u[n]}^{ital} \text{doni}$  and Martin Vetterli

    $y[n] = (1.05)^n \text{Dig}_{u[n]}^{ital} \text{doni}$  and  $y[n] = (1.05)^n \text{Dig}_{u[n]}^{ital} \text{doni}$  © 2013

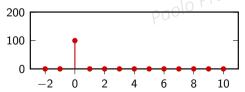




$$x[n] = 100 \delta[n]$$

- y[0] = 100
- ► y[2] = 110.25, y[3] = 115.7625 etc. ignal Processing Vetterli

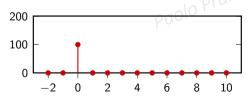
  ► In general:  $y[n] = (1.05)^n \text{Dig}[\frac{1}{n}]_{A} \text{ oni}$  and Martin





$$x[n] = 100 \delta[n]$$

- y[0] = 100
- y[1] = 105
- $y[2] = 110.25, y[3] = 115.7625 \text{ etc.}_{ign}$
- In general:  $y[n] = (1.05)^n \mathbb{D}i \Theta u[n] doni dn$

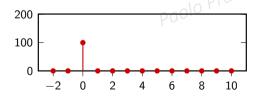


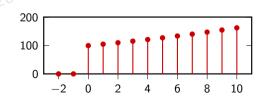
# Example: the one-time investment



$$x[n] = 100 \delta[n]$$

- y[0] = 100
- y[1] = 105
- y[2] = 110.25, y[3] = 115.7625 etc.
- ▶ In general:  $y[n] = (1.05)^n 100 u[n]$



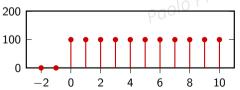




$$x[n] = 100 u[n]$$

- y[0] = 100
- ► y[2] = 315.25, y[3] = 431.0125 etc. ignal Processing

  ► In general:  $y[n] = 2000 ((Dig)^{n+1} to Dia[n] to 1)$   $y[n] = 2000 ((Dig)^{n+1} to Dia[n] to 1)$   $y[n] = 2000 ((Dig)^{n+1} to Dia[n] to 1)$

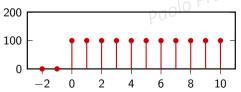




$$x[n] = 100 u[n]$$

- y[0] = 100
- ► y[2] = 315.25, y[3] = 431.0125 etsignal Processing

  ► In general:  $y[n] = 2000 ((Dog)^{n+1} dop) and Martin$



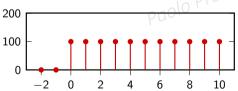


$$x[n] = 100 u[n]$$

- y[0] = 100
- ► y[2] = 315.25, y[3] = 431.0125 etc. ignal Processing Vetterli

  ► In general: y[n] = 2000 ((Distribution)) and Martin Vetterli

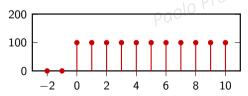
   <math>y[n] = 2000 ((Distribution)) and Martin Vetterli





$$x[n] = 100 u[n]$$

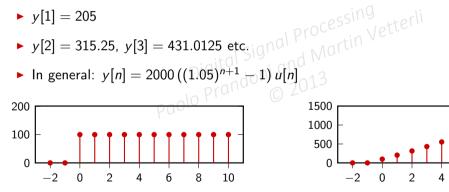
- y[0] = 100
- y[1] = 205
- $y[2] = 315.25, y[3] = 431.0125 \text{ etc.}_{ign}$
- In general:  $y[n] = 2000 \left( \left( \frac{\text{Dog}}{\text{Dog}} \right) \right) \frac{1}{\text{Action 10}} \frac{1}{\text{Dog}} \frac{1}{\text{D$

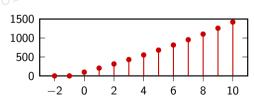




$$x[n] = 100 u[n]$$

- y[0] = 100





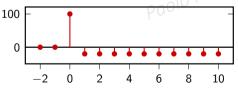


$$x[n] = 100 \,\delta[n] - 5 \,u[n-1]$$

- v[0] = 100

- In general: y[n] = 100 etc.

  Signal Processing Vetterli Signal Processing Vetterli Nartin Vetterli Processing Vetterli Nartin Nartin Vetterli Nartin Nartin Vetterli Nartin N





$$x[n] = 100 \,\delta[n] - 5 \,u[n-1]$$

- v[0] = 100

- In general: y[n] = 100 etc.

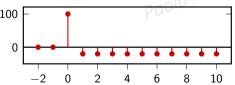
  Signal Processing

  Vetterli

  In general: y[n] = 100 u[n] pigital signal martin vetterli

  Paolo Prandoni and Martin vetterli

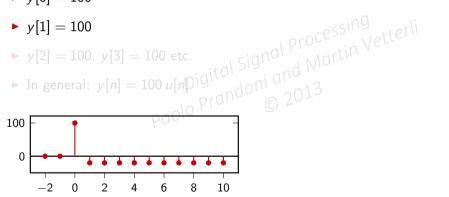
  2013





$$x[n] = 100 \,\delta[n] - 5 \,u[n-1]$$

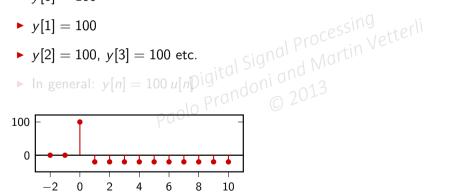
- v[0] = 100





$$x[n] = 100 \delta[n] - 5 u[n-1]$$

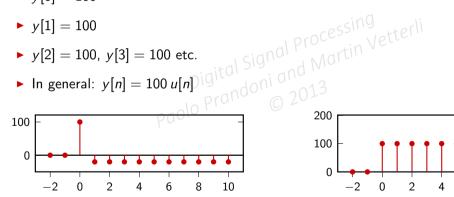
- v[0] = 100

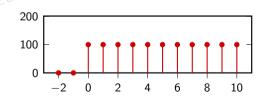




$$x[n] = 100 \delta[n] - 5 u[n-1]$$

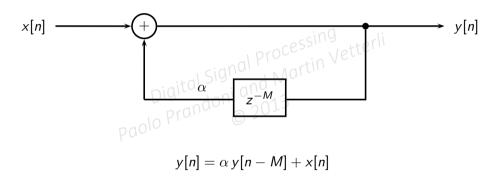
- v[0] = 100





# A simple generalization

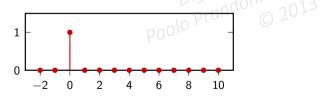






$$M = 3$$
,  $\alpha = 0.7$ ,  $x[n] = \delta[n]$ 

- | y[0] = 1, y[1] = 0, y[2] = 0
- ►  $y[6] = 0.7^2$ , y[7] = 0, y[8] = 0 tetsignal Processing Vetterli Digitatsignal Martin Vetterli



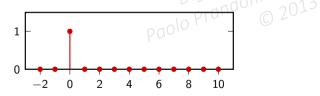


$$M = 3$$
,  $\alpha = 0.7$ ,  $x[n] = \delta[n]$ 

$$y[0] = 1, y[1] = 0, y[2] = 0$$

$$y[3] = 0.7, y[4] = 0, y[5] = 0$$

►  $y[6] = 0.7^2$ , y[7] = 0, y[8] = 0 tetsignal Processing Vetterli Digitatsignal Martin Vetterli



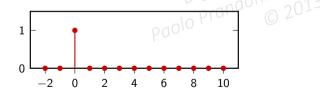


$$M = 3$$
,  $\alpha = 0.7$ ,  $x[n] = \delta[n]$ 

$$y[0] = 1, y[1] = 0, y[2] = 0$$

$$y[3] = 0.7, y[4] = 0, y[5] = 0$$

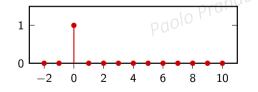
 $y[6] = 0.7^{2}, y[7] = 0, y[8] = 0$   $y[6] = 0.7^{2}, y[7] = 0, y[8] = 0$   $y[8] = 0.7^{2} \text{ and Martin Vetterli}$   $y[8] = 0.7^{2} \text{ and Martin Vetterli}$ 

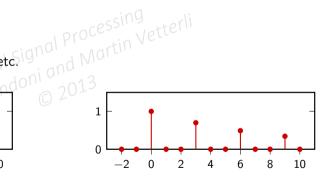




$$M = 3$$
,  $\alpha = 0.7$ ,  $x[n] = \delta[n]$ 

- y[0] = 1, y[1] = 0, y[2] = 0
- ►  $y[6] = 0.7^2$ , y[7] = 0, y[8] = 0, etc. ignal Process

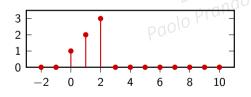


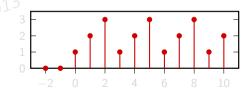




$$M = 3$$
,  $\alpha = 1$ ,  $x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$ 

- v[0] = 1, v[1] = 2, v[2] = 3
- andoni and Martin Vetterli y[6] = 1, y[7] = 2, y[8] = 3, etc. | Signal processing | Signal

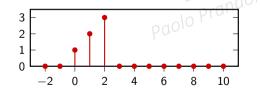


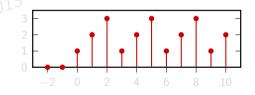




$$M = 3$$
,  $\alpha = 1$ ,  $x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$ 

- y[0] = 1, y[1] = 2, y[2] = 3
- ndoni and Martin Vetterli y[6] = 1, y[7] = 2, y[8] = 3, etc. | Signal processing | Signal

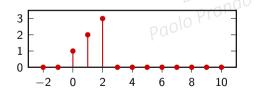


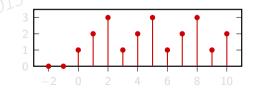




$$M = 3$$
,  $\alpha = 1$ ,  $x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$ 

- y[0] = 1, y[1] = 2, y[2] = 3
- ndoni and Martin Vetterli y[6] = 1, y[7] = 2, y[8] = 3, etc. | Signal Processing Ve

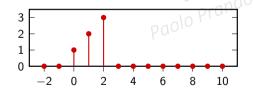


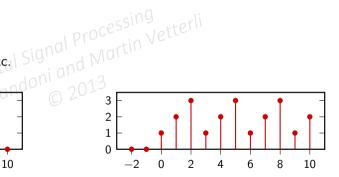




$$M = 3$$
,  $\alpha = 1$ ,  $x[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$ 

- y[0] = 1, y[1] = 2, y[2] = 3
- y[6] = 1, y[7] = 2, y[8] = 3, etc. | Signal procession | Signal







- ▶ build a recursion loop with a delay of M
- ► choose a signal  $\bar{x}[n]$  that is nonzero only for  $\bar{x}[n]$  and  $\bar{x}[n]$  that is nonzero only for  $\bar{x}[n]$  and  $\bar{x}[n]$  that is nonzero only for  $\bar{x}[n]$  and  $\bar{x}[n]$  input  $\bar{x}[n]$  to the system  $\bar{x}[n]$  to  $\bar{x}[n]$  to  $\bar{x}[n]$  to  $\bar{x}[n]$  that is nonzero only for  $\bar{x}[n]$  to  $\bar{x}[n]$  that is nonzero only for  $\bar{x}[n]$  that is nonzero only fo



- ▶ build a recursion loop with a delay of M
- ► choose a signal  $\bar{x}[n]$  that is nonzero only for  $0 \le n \le M$  etterli

  ► choose a decay factor

  input  $\bar{x}[n]$  to the system part of  $0 \le n \le M$  and  $0 \le n \le M$ .

  Figure 1. The plant is nonzero only for  $0 \le n \le M$ .

  Figure 2. The part of  $0 \le n \le M$ .

  Figure 2. The part of  $0 \le n \le M$ .

  Figure 2. The part of  $0 \le n \le M$ .

  - play the output



- ▶ build a recursion loop with a delay of M
- ► choose a signal  $\bar{x}[n]$  that is nonzero only for  $0 \le n \le M$  etterli

  ► choose a decay factor

  input  $\bar{x}[n]$  to the system of N and N and N below the output

  - play the output



- ▶ build a recursion loop with a delay of M
- ▶ choose a signal  $\bar{x}[n]$  that is nonzero only for  $0 \le n \le M$ ⇒ a uecay factor Digital Signard Martinesis  $\bar{x}[n]$  to the system Paolo Prandoni © 2013 play the output



- ▶ build a recursion loop with a delay of M
- ▶ choose a signal  $\bar{x}[n]$  that is nonzero only for  $0 \le n \le M$ input  $\bar{x}[n]$  to the system play the output



- ► *M*-tap delay → *M*-sample "periodicity"
- example:  $T = 22.7 \mu \text{p.OM} = 100$



- ► *M*-tap delay → *M*-sample "periodicity"
- example:  $T = 22.7 \mu \text{p.a.d.} = 100$



- ► *M*-tap delay → *M*-sample "periodicity"

► M-tap delay 
$$\longrightarrow$$
 M-sample "periodicity"

► associate time  $T$  to sample interval

► periodic signal of frequency

Digital Signal Processing

Vetterli

Franco  $f = \frac{1}{MT}Hz$ 

Example:  $T = 22.7\mu$  Mark  $f \approx 440$ Hz

$$f \approx 440 \text{Hz}$$



- ► *M*-tap delay → *M*-sample "periodicity"

► M-tap delay 
$$\longrightarrow$$
 M-sample "periodicity"

► associate time  $T$  to sample interval

► periodic signal of frequency

Digital Signal Processing

Frequency

 $f = \frac{1}{MT}$ Hz

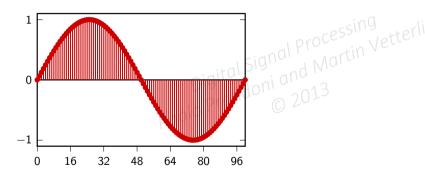
► example:  $T = 22.7\mu s$ ,  $M = 100$ 
 $f \approx 440$ Hz

$$f \approx 440 \text{Hz}$$

# Playing a sine wave



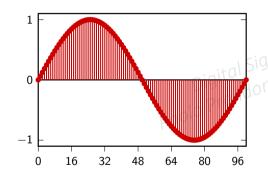
$$M=100, \ \alpha=1, \ \bar{x}[n]=\sin(2\pi\,n/100)$$
 for  $0\leq n<100$  and zero elsewhere

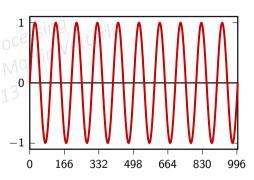


# Playing a sine wave



$$M=100,\ \alpha=1,\ \bar{x}[n]=\sin(2\pi\,n/100)$$
 for  $0\leq n<100$  and zero elsewhere





# Introducing some realism



- $\alpha$  controls envelope (decay) Digital Signal Martin Vetterli  $\bar{x}[n]$  controls color (timbre) Prandoni and Processing Vetterli  $\alpha$  controls envelope (decay) Digital Signal Martin Vetterli  $\alpha$  controls envelope (decay) Digital Signal Processing Vetterli  $\alpha$  controls envelope (decay) Digital Signal Processing Vetterli  $\alpha$  controls envelope (decay) Digital Signal Processing Vetterli

# Introducing some realism



# Introducing some realism



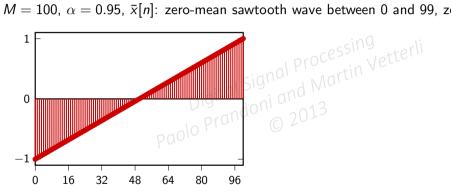
- ►  $\alpha$  controls envelope (decay) gital Signal Processing Vetterli

    $\bar{x}[n]$  controls color (timbre) Prandoni and Martin Paolo Prandoni 2013

# A proto-violin



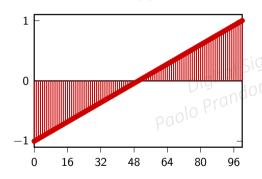
 $M=100, \ \alpha=0.95, \ \bar{x}[n]$ : zero-mean sawtooth wave between 0 and 99, zero elsewhere

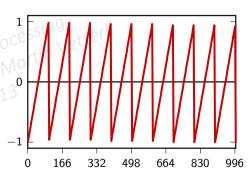


# A proto-violin



 $M=100,~\alpha=0.95,~\bar{x}[n]$ : zero-mean sawtooth wave between 0 and 99, zero elsewhere

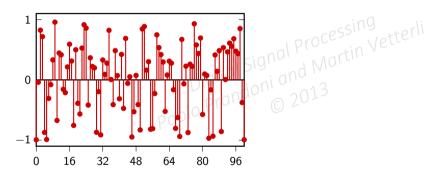




# The Karplus-Strong Algorithm



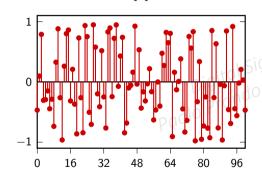
 $M=100, \ \alpha=0.9, \ \bar{x}[n]$ : 100 random values between 0 and 99, zero elsewhere

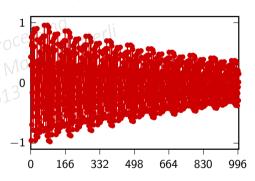


# The Karplus-Strong Algorithm



 $M=100,~\alpha=0.9,~\bar{x}[n]$ : 100 random values between 0 and 99, zero elsewhere





### Recap



- We have seen basic elements:
  - adders
- We have seen two systems igital Signal Processing Vetterli
   moving averages
   recursive systems Paolo Prandoni and Martin

  - ▶ We were able to build simple systems with interesting properties
  - to understand all of this in more details we need a mathematical framework!

END OF MODULE 2.3

Digital Signa Martine 2.3

Paolo Prandoni and Martine 2013

W

Digital Signal Processin Vetters

# END OF MODULE 2 Digital Signal Martin LE 2 Paolo Prandoni and Martin LE 2