Digital Signal Processing

Module 4: Fourier Analysis

Module Overview:

- Modules 4.1: Introduction to Fourier Analysis
- Modules 4.2: The Discrete Fourier Transform (DFT)
- Modules 4.3: DFT in practice
- Modules 4.4: The Discrete-Time Fourier Transform (DTFT)
- Modules 4.5: DTFT properties
- Modules 4.6: Relationships between transforms
- Modules 4.7: Sinusoidal modulation and applications
- Modules 4.8: The Short-Time Fourier Transform
- Modules 4.9: The FFT: History and Examples advanced topics: DTFT as a formal basis expansion Relationships between transforms

Digital Signal Processing

Module 4.1: Exploration via a change of basis

Overview:

- The frequency domain
- Frequency analysis as a change of basis
- The Fourier basis

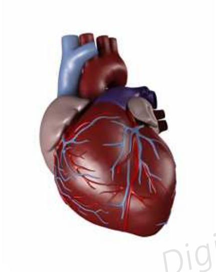
Overview:

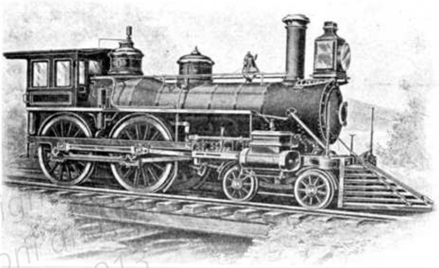
- The frequency domain
- Frequency analysis as a change of basis
- The Fourier basis

Overview:

- The frequency domain
- Frequency analysis as a change of basis
- The Fourier basis

Oscillations are everywhere!





Oscillations are everywhere

- sustainable dynamic systems exhibit oscillatory behavior
- intuitively: things that don't move in circles can't last:
  - bombs
  - rockets
  - human beings...

Oscillations are everywhere

- sustainable dynamic systems exhibit oscillatory behavior
- intuitively: things that don't move in circles can't last:
  - bombs
  - rockets
  - human beings...

Oscillations are everywhere

- sustainable dynamic systems exhibit oscillatory behavior
- intuitively: things that don't move in circles can't last:
  - bombs
  - rockets
  - human beings...

Oscillations are everywhere

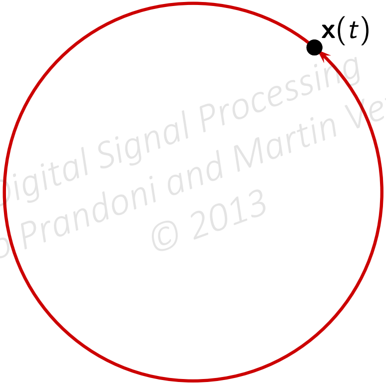
- sustainable dynamic systems exhibit oscillatory behavior
- intuitively: things that don't move in circles can't last:
  - bombs
  - rockets
  - human beings...

Oscillations are everywhere

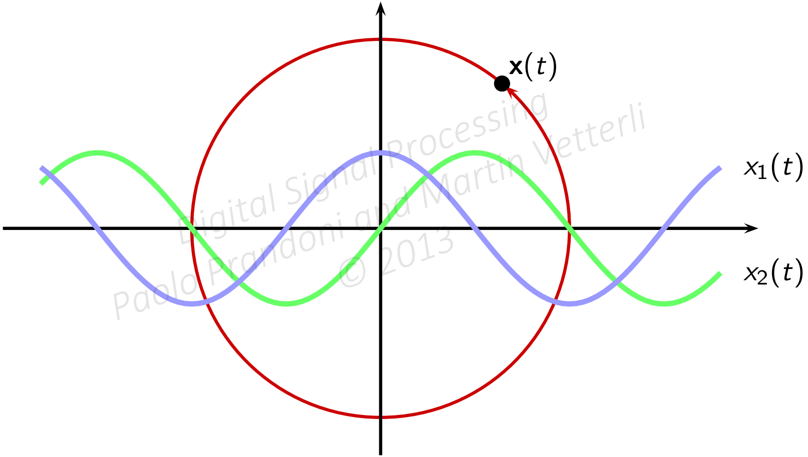
- sustainable dynamic systems exhibit oscillatory behavior
- intuitively: things that don't move in circles can't last:
  - bombs
  - rockets
  - human beings...

Oscillations are everywhere

Oscillations are everywhere



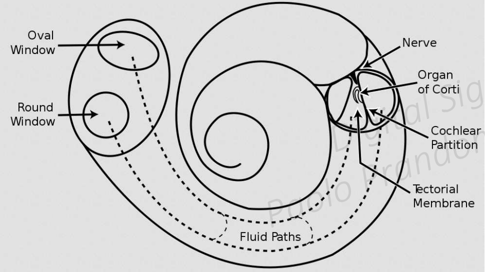
Oscillations are everywhere



You too can detect sinusoids!

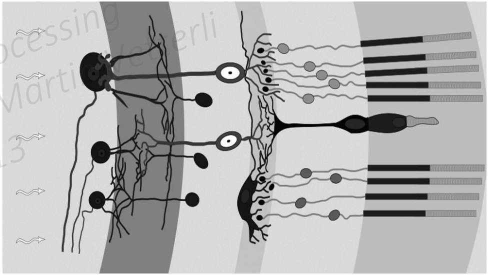
the human body has two receptors for sinusoidal signals:

cochlea (inner ear)



- air pressure sinusoids
- frequencies from 20Hz to 20KHz

rods and cones (retina)



- electromagnetic sinusoids
- frequencies from 430THz to 790THz

The intuition

- humans analyze complex signals (audio, images) in terms of their sinusoidal components
- we can build instruments that “resonate” at one or multiple frequencies (tuning fork vs piano)
- the “frequency domain” seems to be as important as the time domain

The intuition

- humans analyze complex signals (audio, images) in terms of their sinusoidal components
- we can build instruments that "resonate" at one or multiple frequencies (tuning fork vs piano)
- the "frequency domain" seems to be as important as the time domain

The intuition

- humans analyze complex signals (audio, images) in terms of their sinusoidal components
- we can build instruments that "resonate" at one or multiple frequencies (tuning fork vs piano)
- the "frequency domain" seems to be as important as the time domain

Fundamental question

can we decompose any signal into sinusoidal elements?

yes, and Fourier showed us how to do it *exactly*!

analysis

- from time domain to frequency domain
- find the contribution of different frequencies
- discover "hidden" signal properties

synthesis

- from frequency domain to time domain
- create signals with known frequency content
- fit signals to specific frequency regions

Fundamental question

can we decompose any signal into sinusoidal elements?

yes, and Fourier showed us how to do it *exactly*!

analysis	synthesis
▸ from time domain to frequency domain	▸ from frequency domain to time domain
▸ find the contribution of different frequencies	▸ create signals with known frequency content
▸ discover “hidden” signal properties	▸ fit signals to specific frequency regions

Fundamental question

can we decompose any signal into sinusoidal elements?

yes, and Fourier showed us how to do it *exactly*!

analysis

- from time domain to frequency domain
- find the contribution of different frequencies
- discover "hidden" signal properties

synthesis

- from frequency domain to time domain
- create signals with known frequency content
- fit signals to specific frequency regions

Fundamental question

can we decompose any signal into sinusoidal elements?

yes, and Fourier showed us how to do it *exactly*!

analysis

- from time domain to frequency domain
- find the contribution of different frequencies
- discover "hidden" signal properties

synthesis

- from frequency domain to time domain
- create signals with known frequency content
- fit signals to specific frequency regions

The mathematical setup

- let's start with finite-length signals (i.e. vectors in $\mathbb{C}^N$)
- Fourier analysis is a simple change of basis
- a change of basis is a change of perspective
- a change of perspective can reveal things (if the basis is good)

The mathematical setup

- let's start with finite-length signals (i.e. vectors in $\mathbb{C}^N$)
- Fourier analysis is a simple change of basis
- a change of basis is a change of perspective
- a change of perspective can reveal things (if the basis is good)

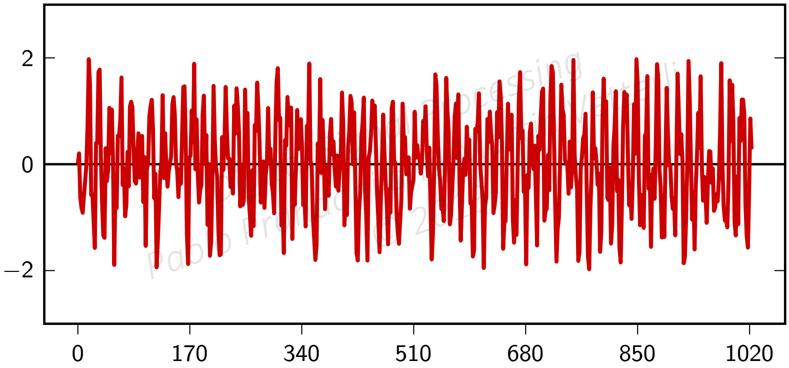
The mathematical setup

- let's start with finite-length signals (i.e. vectors in $\mathbb{C}^N$)
- Fourier analysis is a simple change of basis
- a change of basis is a change of perspective
- a change of perspective can reveal things (if the basis is good)

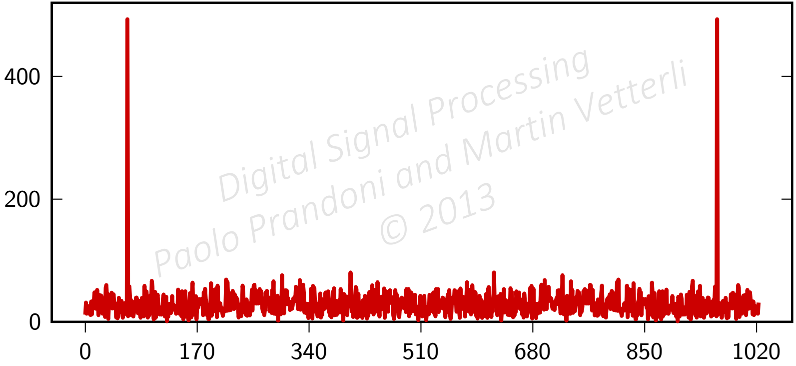
The mathematical setup

- let's start with finite-length signals (i.e. vectors in $\mathbb{C}^N$)
- Fourier analysis is a simple change of basis
- a change of basis is a change of perspective
- a change of perspective can reveal things (if the basis is good)

Mystery signal



Mystery signal in the Fourier basis



The Fourier Basis for $\mathbb{C}^N$

Claim: the set of N signals in $\mathbb{C}^N$

$$w_k[n] = e^{j\frac{2\pi}{N}nk}, \qquad n, k = 0, 1, \ldots, N-1$$

is an orthogonal basis in $\mathbb{C}^N$.

In vector notation:

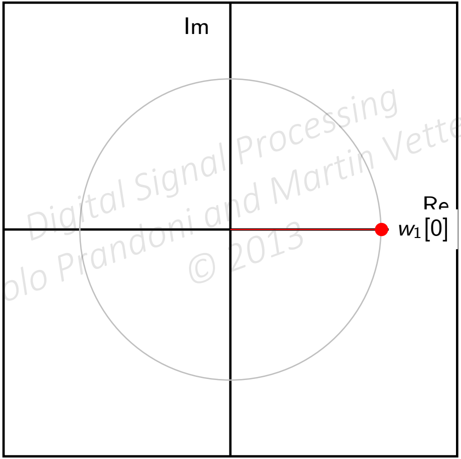
$$\{\mathbf{w}^{(k)}\}_{k=0,1,\ldots,N-1}$$

with

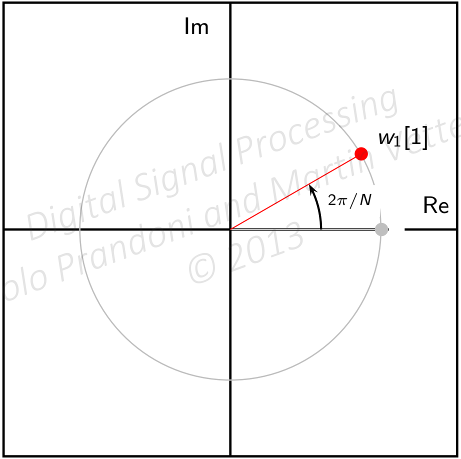
$$w_n^{(k)} = e^{j\frac{2\pi}{N}nk}$$

is an orthogonal basis in $\mathbb{C}^N$

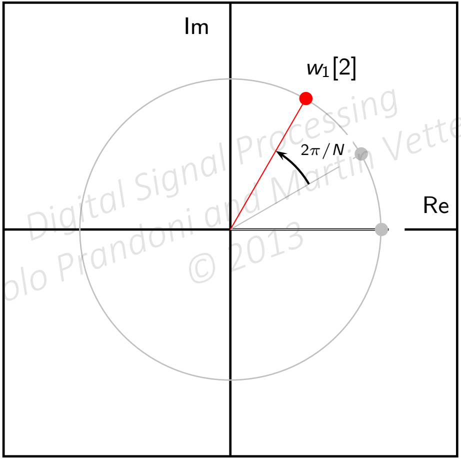
Recall the complex exponential generating machine...



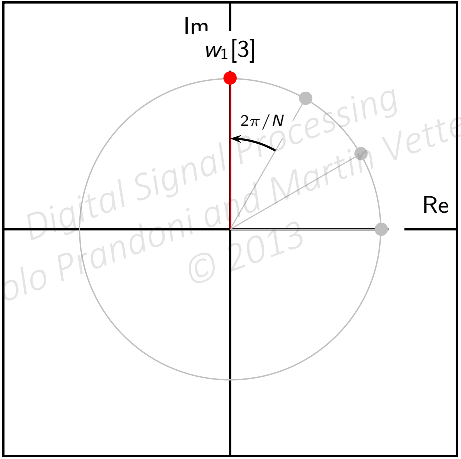
Recall the complex exponential generating machine...



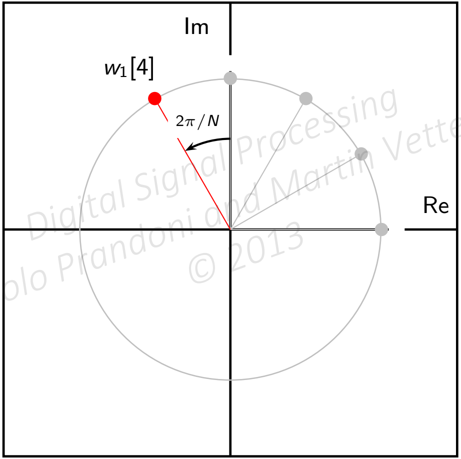
Recall the complex exponential generating machine...



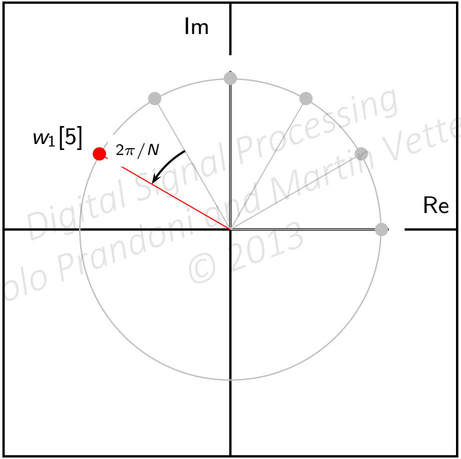
Recall the complex exponential generating machine...



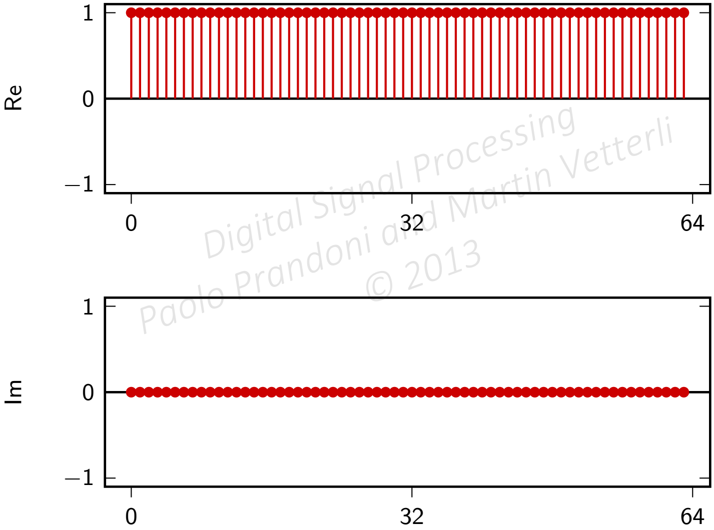
Recall the complex exponential generating machine...



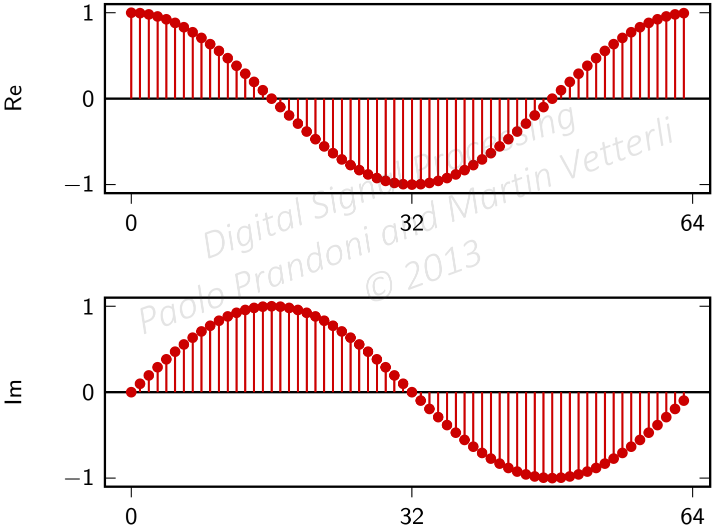
Recall the complex exponential generating machine...



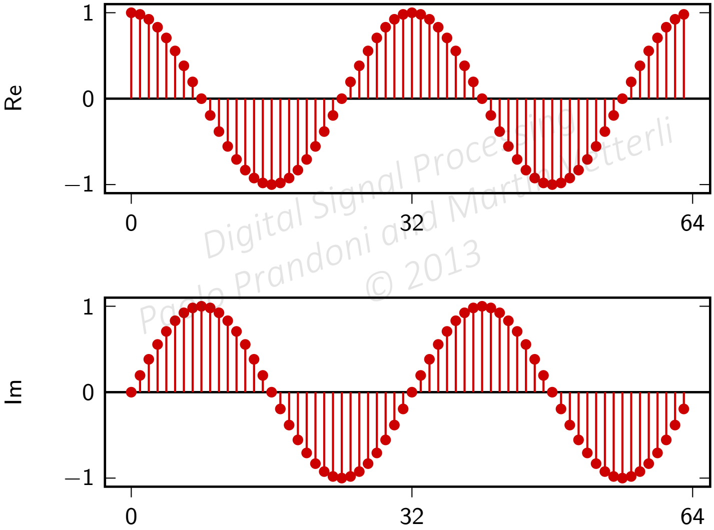
Basis vector $\mathbf{w}^{(0)} \in \mathbb{C}^{64}$



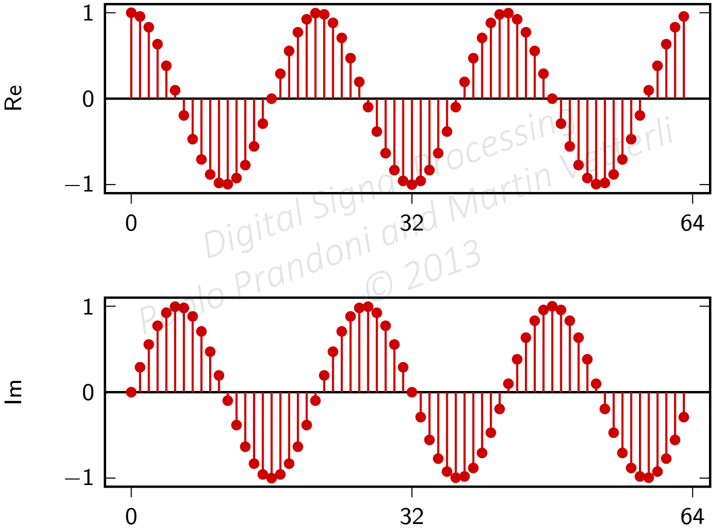
Basis vector $\mathbf{w}^{(1)} \in \mathbb{C}^{64}$



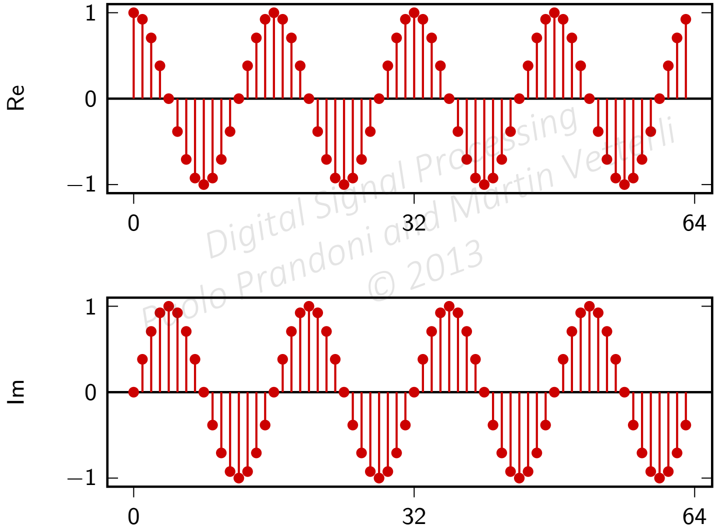
Basis vector $\mathbf{w}^{(2)} \in \mathbb{C}^{64}$



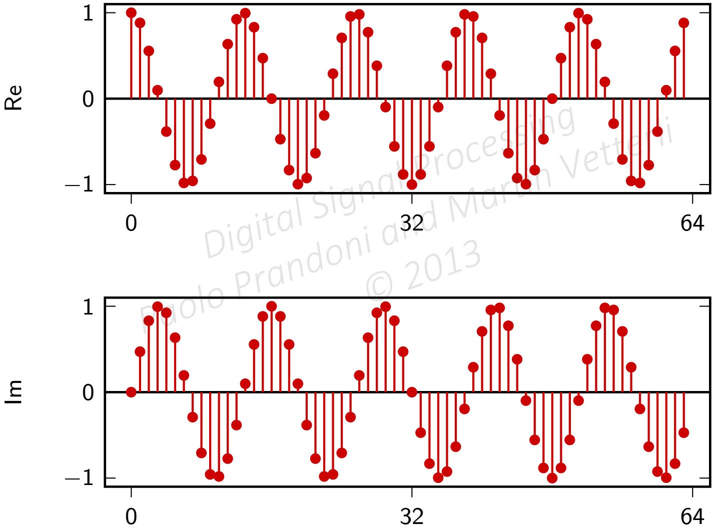
Basis vector $\mathbf{w}^{(3)} \in \mathbb{C}^{64}$



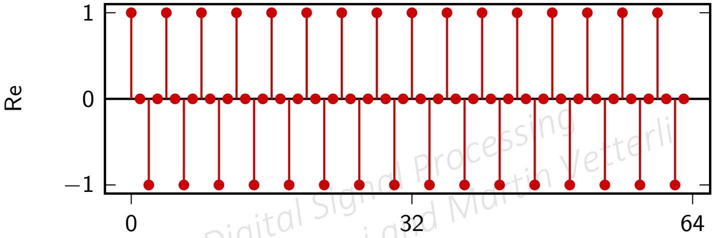
Basis vector $\mathbf{w}^{(4)} \in \mathbb{C}^{64}$

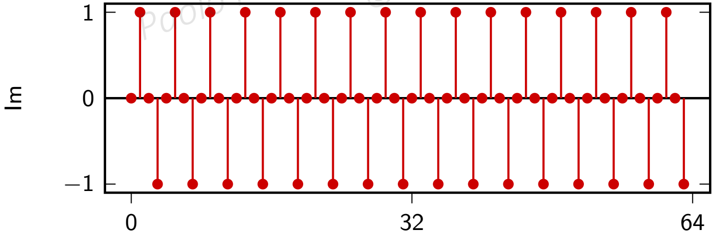


Basis vector $\mathbf{w}^{(5)} \in \mathbb{C}^{64}$

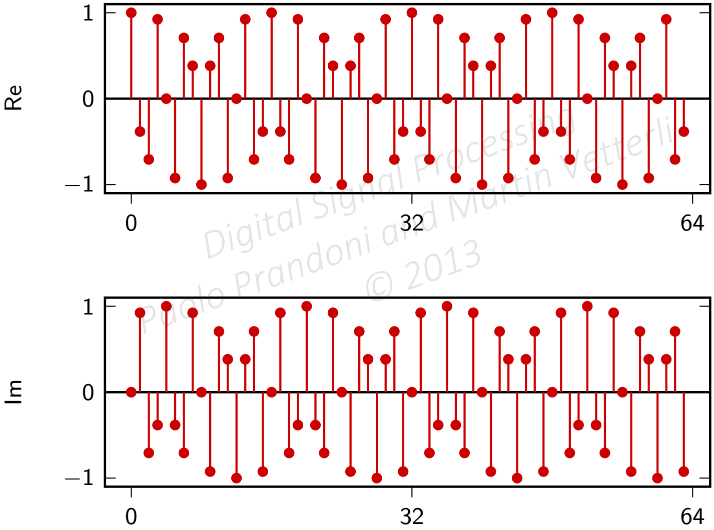


Basis vector $\mathbf{w}^{(16)} \in \mathbb{C}^{64}$

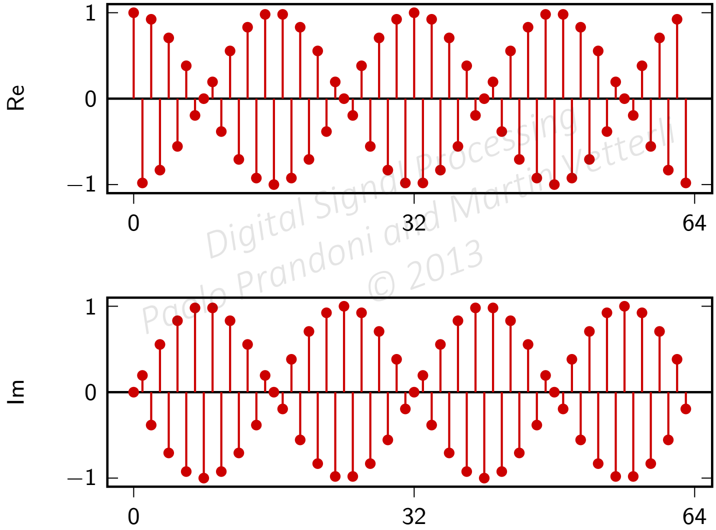




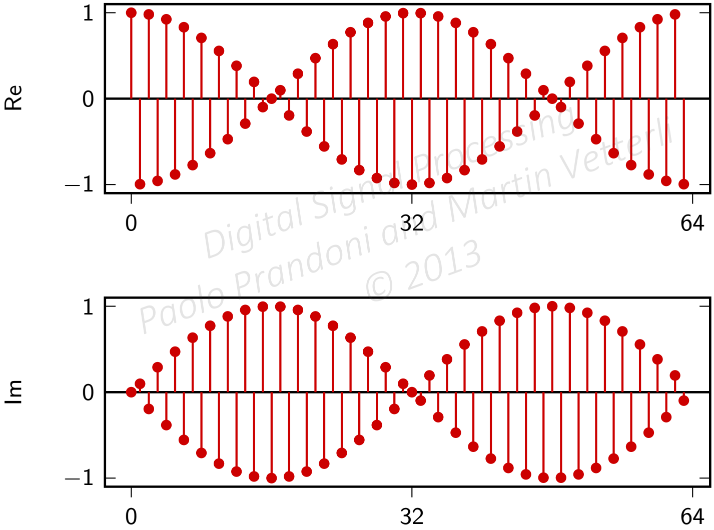
Basis vector $\mathbf{w}^{(20)} \in \mathbb{C}^{64}$



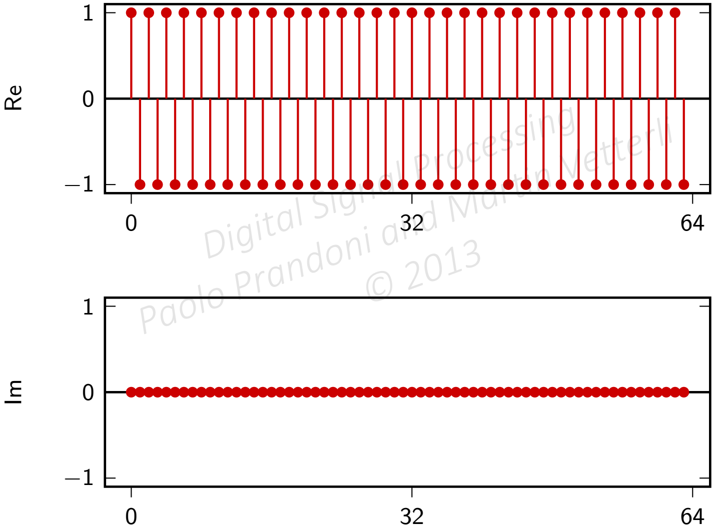
Basis vector $\mathbf{w}^{(30)} \in \mathbb{C}^{64}$



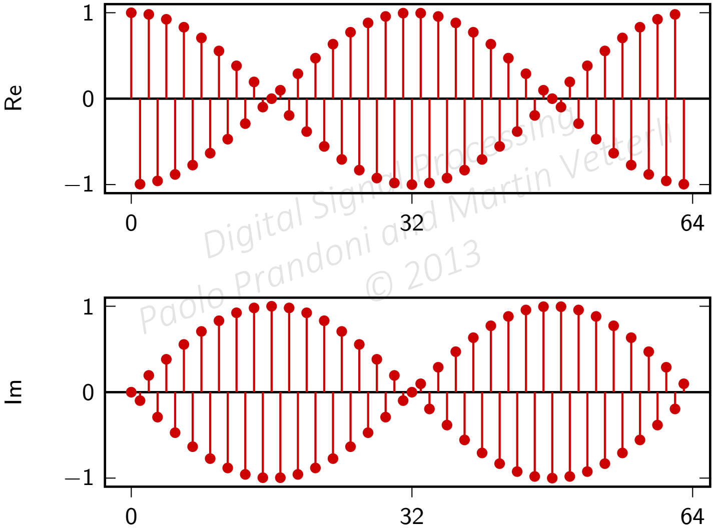
Basis vector $\mathbf{w}^{(31)} \in \mathbb{C}^{64}$



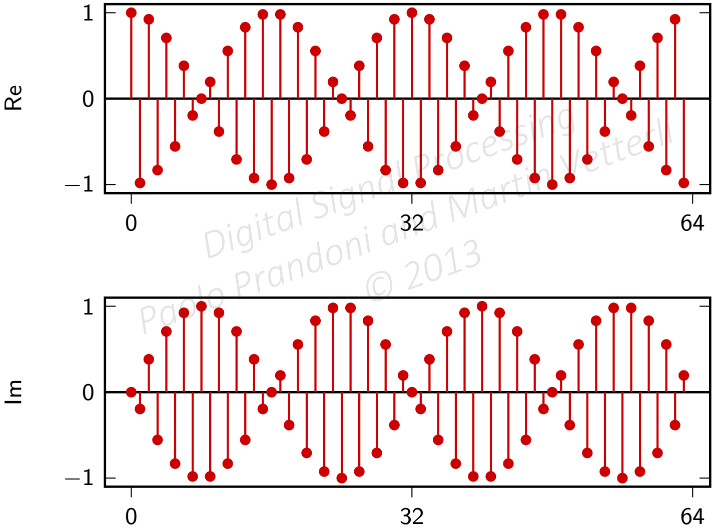
Basis vector $\mathbf{w}^{(32)} \in \mathbb{C}^{64}$



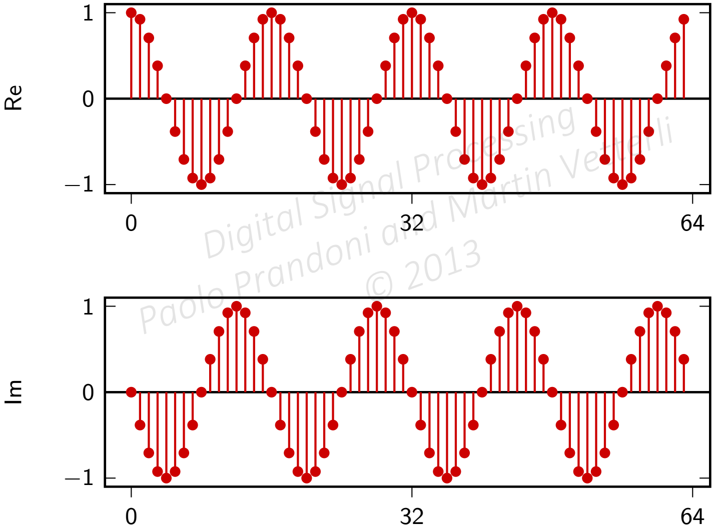
Basis vector $\mathbf{w}^{(33)} \in \mathbb{C}^{64}$



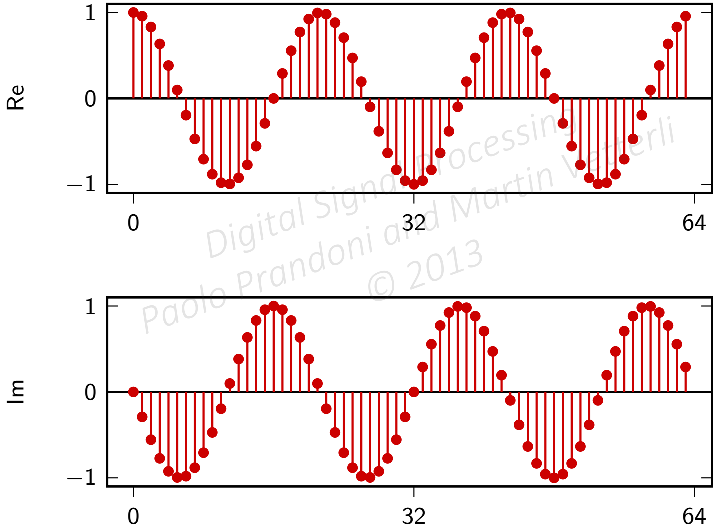
Basis vector $\mathbf{w}^{(34)} \in \mathbb{C}^{64}$



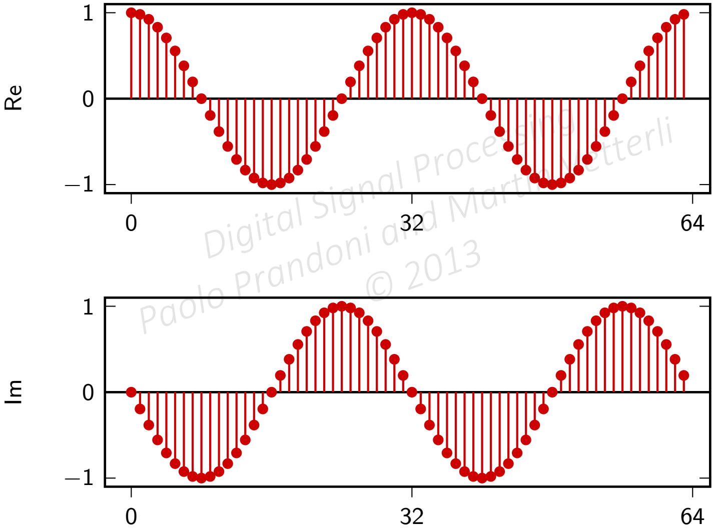
Basis vector $\mathbf{w}^{(60)} \in \mathbb{C}^{64}$



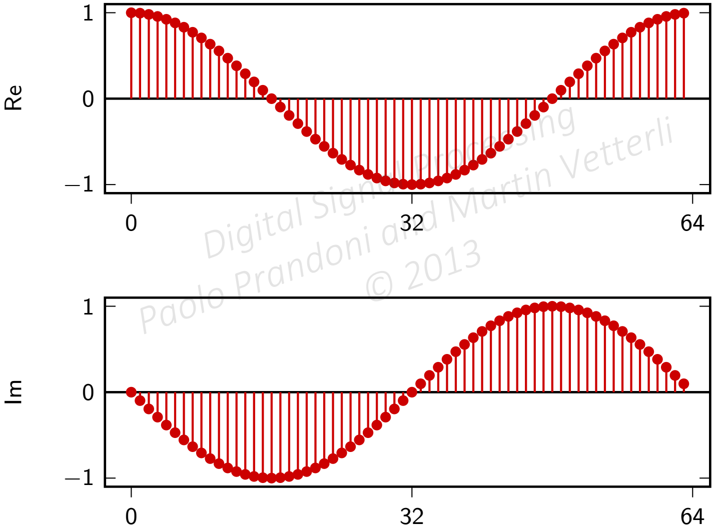
Basis vector $\mathbf{w}^{(61)} \in \mathbb{C}^{64}$



Basis vector $\mathbf{w}^{(62)} \in \mathbb{C}^{64}$



Basis vector $\mathbf{w}^{(63)} \in \mathbb{C}^{64}$



Proof of orthogonality

$$
\begin{aligned}
\langle \mathbf{w}^{(k)}, \mathbf{w}^{(h)} \rangle &= \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* \, e^{j\frac{2\pi}{N}nh} \\
&= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(h-k)n} \\
&= \begin{cases} N & \text{for } h = k \\ \dfrac{1 - e^{j2\pi(h-k)}}{1 - e^{j\frac{2\pi}{N}(h-k)}} = 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Proof of orthogonality

$$\langle \mathbf{w}^{(k)}, \mathbf{w}^{(h)} \rangle = \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* \, e^{j\frac{2\pi}{N}nh}$$

$$= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(h-k)n}$$

$$= \begin{cases} N & \text{for } h = k \\ \dfrac{1 - e^{j2\pi(h-k)}}{1 - e^{j\frac{2\pi}{N}(h-k)}} = 0 & \text{otherwise} \end{cases}$$

Proof of orthogonality

$$\langle \mathbf{w}^{(k)}, \mathbf{w}^{(h)} \rangle = \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* \, e^{j\frac{2\pi}{N}nh}$$

$$= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(h-k)n}$$

$$= \begin{cases} N & \text{for } h = k \\ \dfrac{1 - e^{j2\pi(h-k)}}{1 - e^{j\frac{2\pi}{N}(h-k)}} = 0 & \text{otherwise} \end{cases}$$

Proof of orthogonality

$$
\begin{aligned}
\langle \mathbf{w}^{(k)}, \mathbf{w}^{(h)} \rangle &= \sum_{n=0}^{N-1} (e^{j\frac{2\pi}{N}nk})^* \, e^{j\frac{2\pi}{N}nh} \\
&= \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(h-k)n} \\
&= \begin{cases} N & \text{for } h = k \\ \dfrac{1 - e^{j2\pi(h-k)}}{1 - e^{j\frac{2\pi}{N}(h-k)}} = 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Remarks

- N orthogonal vectors $\longrightarrow$ basis for $\mathbb{C}^N$
- vectors are not ortho*normal*. Normalization factor would be $1/\sqrt{N}$

Remarks

- N orthogonal vectors $\longrightarrow$ basis for $\mathbb{C}^N$
- vectors are not ortho*normal*. Normalization factor would be $1/\sqrt{N}$

END OF MODULE 4.1

Digital Signal Processing

Module 4.2: The Discrete Fourier Transform

Overview:

- The Discrete Fourier Transform
- DFT examples
- interpreting a DFT plot

Overview:

- The Discrete Fourier Transform
- DFT examples
- interpreting a DFT plot

Overview:

- The Discrete Fourier Transform
- DFT examples
- interpreting a DFT plot

The Fourier Basis for $\mathbb{C}^N$

- in "signal" notation: $w_k[n] = e^{j\frac{2\pi}{N}nk}, \qquad n, k = 0, 1, \ldots, N-1$
- in vector notation: $\{\mathbf{w}^{(k)}\}_{k=0,1,\ldots,N-1}$ with $w_n^{(k)} = e^{j\frac{2\pi}{N}nk}$

The Fourier Basis for $\mathbb{C}^N$

- in "signal" notation: $w_k[n] = e^{j\frac{2\pi}{N}nk}, \qquad n, k = 0, 1, \ldots, N-1$
- in vector notation: $\{\mathbf{w}^{(k)}\}_{k=0,1,\ldots,N-1}$ with $w_n^{(k)} = e^{j\frac{2\pi}{N}nk}$

The Fourier Basis for $\mathbb{C}^N$

- N orthogonal vectors $\longrightarrow$ basis for $\mathbb{C}^N$
- vectors are not ortho*normal*. Normalization factor would be $1/\sqrt{N}$
- will keep normalization factor explicit in DFT formulas

Analysis formula:

$$X_k = \langle \mathbf{w}^{(k)}, \mathbf{x} \rangle$$

Synthesis formula:

$$\mathbf{x} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \mathbf{w}^{(k)}$$

Define $W_N = e^{-j\frac{2\pi}{N}}$
(or simply W when N is evident from the context)

Change of basis matrix $\mathbf{W}$ with $\mathbf{W}[n,m] = W_N^{nm}$:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

Change of basis in matrix form

Define $W_N = e^{-j\frac{2\pi}{N}}$
(or simply W when N is evident from the context)

Change of basis matrix $\mathbf{W}$ with $\mathbf{W}[n,m] = W_N^{nm}$:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

Change of basis in matrix form

Analysis formula:

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

Synthesis formula:

$$\mathbf{x} = \frac{1}{N}\mathbf{W}^H\mathbf{X}$$

Basis expansion (signal notation)

Analysis formula:

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

N-point signal in the *frequency domain*

Synthesis formula:

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

N-point signal in the *"time" domain*

Basis expansion (signal notation)

Analysis formula:

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

N-point signal in the *frequency domain*

Synthesis formula:

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

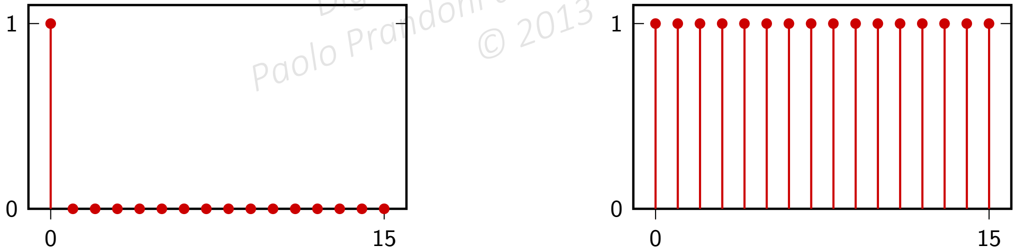
N-point signal in the *"time" domain*

DFT is obviously linear

$$\text{DFT}\{\alpha\, x[n] + \beta\, y[n]\} = \alpha\, \text{DFT}\{x[n]\} + \beta\, \text{DFT}\{y[n]\}$$

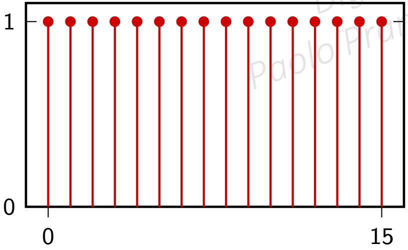
$$X[k] = \sum_{n=0}^{N-1} \delta[n]\, e^{-j\frac{2\pi}{N}nk}$$

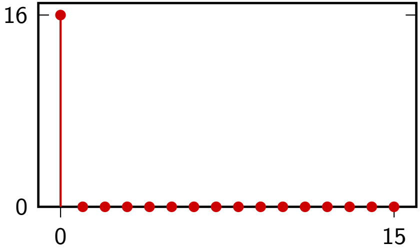
$$= 1$$



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

$$= N\delta[k]$$





$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right] \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{j\frac{2\pi}{64}60n}\right] \\
&= \frac{3}{2}(w_4[n] + w_{60}[n])
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n)$, $\quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right] \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{j\frac{2\pi}{64}60n}\right] \\
&= \frac{3}{2}\left(w_4[n] + w_{60}[n]\right)
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n)$, $\quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right] \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{j\frac{2\pi}{64}60n}\right] \\
&= \frac{3}{2}(w_4[n] + w_{60}[n])
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n), \quad x[n] \in \mathbb{C}^{64}$

$$x[n] = 3\cos\left(\frac{2\pi}{16}\, n\right)$$

$$= 3\cos\left(\frac{2\pi}{64}\, 4\, n\right)$$

$$= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right]$$

$$= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{j\frac{2\pi}{64}60n}\right]$$

$$= \frac{3}{2}(w_4[n] + w_{60}[n])$$

DFT of $x[n] = 3\cos(2\pi/16\, n), \quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{-j\frac{2\pi}{64}4n}\right] \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} + e^{j\frac{2\pi}{64}60n}\right] \\
&= \frac{3}{2}(w_4[n] + w_{60}[n])
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n), \quad x[n] \in \mathbb{C}^{64}$

$$X[k] = \langle w_k[n], x[n] \rangle$$

$$= \langle w_k[n], \frac{3}{2}(w_4[n] + w_{60}[n]) \rangle$$

$$= \frac{3}{2}\langle w_k[n], w_4[n] \rangle + \frac{3}{2}\langle w_k[n], w_{60}[n] \rangle$$

$$= \begin{cases} 96 & \text{for } k = 4,\ 60 \\ 0 & \text{otherwise} \end{cases}$$

DFT of $x[n] = 3\cos(2\pi/16\, n), \quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
X[k] &= \langle w_k[n], x[n] \rangle \\
&= \langle w_k[n], \frac{3}{2}(w_4[n] + w_{60}[n]) \rangle \\
&= \frac{3}{2}\langle w_k[n], w_4[n] \rangle + \frac{3}{2}\langle w_k[n], w_{60}[n] \rangle \\
&= \begin{cases} 96 & \text{for } k = 4,\ 60 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

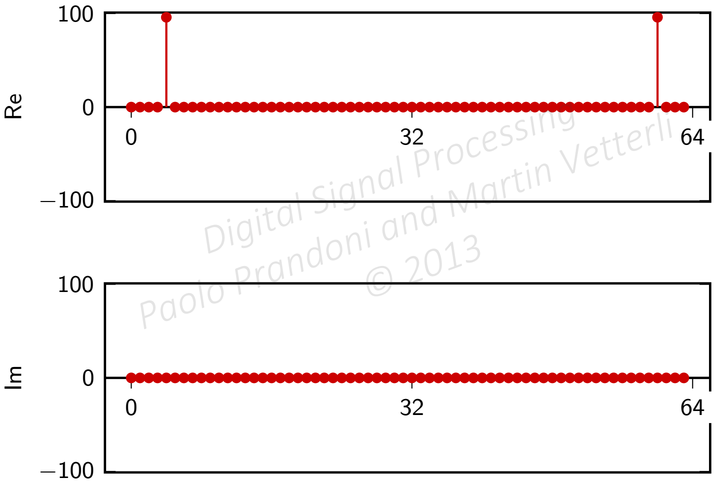
DFT of $x[n] = 3\cos(2\pi/16\, n), \quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
X[k] &= \langle w_k[n], x[n] \rangle \\
&= \langle w_k[n], \frac{3}{2}(w_4[n] + w_{60}[n]) \rangle \\
&= \frac{3}{2}\langle w_k[n], w_4[n] \rangle + \frac{3}{2}\langle w_k[n], w_{60}[n] \rangle \\
&= \begin{cases} 96 & \text{for } k = 4,\ 60 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n)$, $\quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
X[k] &= \langle w_k[n], x[n] \rangle \\
&= \langle w_k[n], \frac{3}{2}(w_4[n] + w_{60}[n]) \rangle \\
&= \frac{3}{2}\langle w_k[n], w_4[n] \rangle + \frac{3}{2}\langle w_k[n], w_{60}[n] \rangle \\
&= \begin{cases} 96 & \text{for } k = 4,\ 60 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n)$, $\quad x[n] \in \mathbb{C}^{64}$



DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3)$, $\quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n + \frac{\pi}{3}\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n + \frac{\pi}{3}\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} e^{j\frac{\pi}{3}} + e^{-j\frac{2\pi}{64}4n} e^{-j\frac{\pi}{3}}\right] \\
&= \frac{3}{2}\left(e^{j\frac{\pi}{3}}\, w_4[n] + e^{-j\frac{\pi}{3}}\, w_{60}[n]\right)
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3), \quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n + \frac{\pi}{3}\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n + \frac{\pi}{3}\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} e^{j\frac{\pi}{3}} + e^{-j\frac{2\pi}{64}4n} e^{-j\frac{\pi}{3}}\right] \\
&= \frac{3}{2}\left(e^{j\frac{\pi}{3}}\, w_4[n] + e^{-j\frac{\pi}{3}}\, w_{60}[n]\right)
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\,n + \pi/3), \quad x[n] \in \mathbb{C}^{64}$

$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\,n + \frac{\pi}{3}\right) \\
&= 3\cos\left(\frac{2\pi}{64}\,4\,n + \frac{\pi}{3}\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n}e^{j\frac{\pi}{3}} + e^{-j\frac{2\pi}{64}4n}e^{-j\frac{\pi}{3}}\right] \\
&= \frac{3}{2}\left(e^{j\frac{\pi}{3}}\,w_4[n] + e^{-j\frac{\pi}{3}}\,w_{60}[n]\right)
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3)$, $\quad x[n] \in \mathbb{C}^{64}$

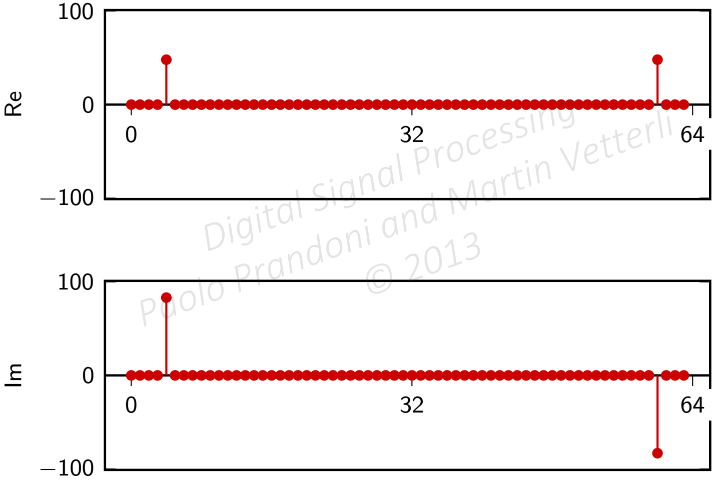
$$
\begin{aligned}
x[n] &= 3\cos\left(\frac{2\pi}{16}\, n + \frac{\pi}{3}\right) \\
&= 3\cos\left(\frac{2\pi}{64}\, 4\, n + \frac{\pi}{3}\right) \\
&= \frac{3}{2}\left[e^{j\frac{2\pi}{64}4n} e^{j\frac{\pi}{3}} + e^{-j\frac{2\pi}{64}4n} e^{-j\frac{\pi}{3}}\right] \\
&= \frac{3}{2}(e^{j\frac{\pi}{3}}\, w_4[n] + e^{-j\frac{\pi}{3}}\, w_{60}[n])
\end{aligned}
$$

DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3), \quad x[n] \in \mathbb{C}^{64}$

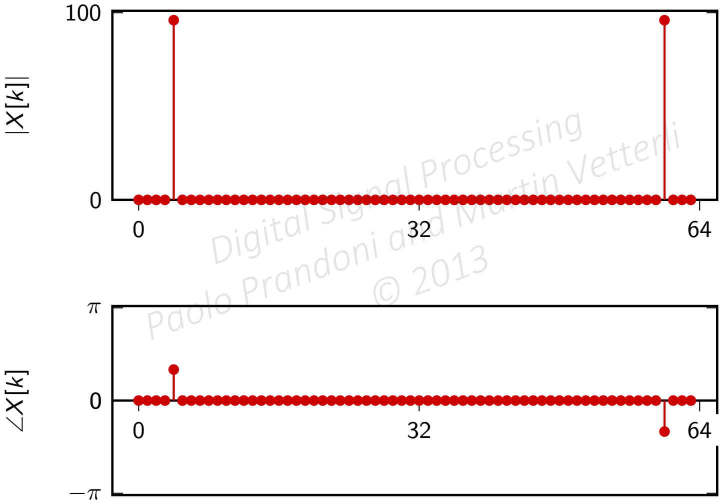
$$X[k] = \langle w_k[n], x[n] \rangle$$

$$= \begin{cases} 96 e^{j\frac{\pi}{3}} & \text{for } k = 4 \\ 96 e^{-j\frac{\pi}{3}} & \text{for } k = 60 \\ 0 & \text{otherwise} \end{cases}$$

DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3), \quad x[n] \in \mathbb{C}^{64}$



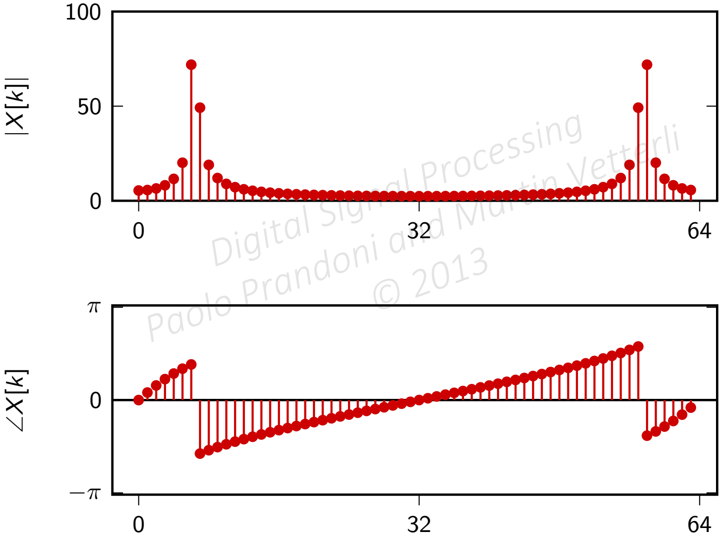
DFT of $x[n] = 3\cos(2\pi/16\, n + \pi/3), \quad x[n] \in \mathbb{C}^{64}$



DFT of $x[n] = 3\cos(2\pi/10\, n), \quad x[n] \in \mathbb{C}^{64}$

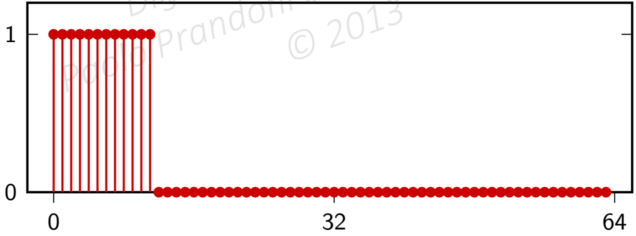
$$\frac{2\pi}{64}6 < \frac{2\pi}{10} < \frac{2\pi}{64}7$$

DFT of $x[n] = 3\cos(2\pi/10\, n)$, $\quad x[n] \in \mathbb{C}^{64}$



DFT of length-M step in $\mathbb{C}^N$

$$x[n] = \sum_{h=0}^{M-1} \delta[n-h], \qquad n = 0, 1, \ldots, N-1$$



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

$$= \frac{1 - e^{-j\frac{2\pi}{N}kM}}{1 - e^{-j\frac{2\pi}{N}k}}$$

$$= \frac{e^{-j\frac{\pi}{N}kM}\left[e^{j\frac{\pi}{N}kM} - e^{-j\frac{\pi}{N}kM}\right]}{e^{-j\frac{\pi}{N}k}\left[e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right]}$$

$$= \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

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

$$= \frac{1 - e^{-j\frac{2\pi}{N}kM}}{1 - e^{-j\frac{2\pi}{N}k}}$$

$$= \frac{e^{-j\frac{\pi}{N}kM}\left[e^{j\frac{\pi}{N}kM} - e^{-j\frac{\pi}{N}kM}\right]}{e^{-j\frac{\pi}{N}k}\left[e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right]}$$

$$= \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

DFT of length-M step in $\mathbb{C}^N$

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

$$= \frac{1 - e^{-j\frac{2\pi}{N}kM}}{1 - e^{-j\frac{2\pi}{N}k}}$$

$$= \frac{e^{-j\frac{\pi}{N}kM}\left[e^{j\frac{\pi}{N}kM} - e^{-j\frac{\pi}{N}kM}\right]}{e^{-j\frac{\pi}{N}k}\left[e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right]}$$

$$= \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

DFT of length-M step in $\mathbb{C}^N$

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{M-1} e^{-j\frac{2\pi}{N}nk} \\
&= \frac{1 - e^{-j\frac{2\pi}{N}kM}}{1 - e^{-j\frac{2\pi}{N}k}} \\
&= \frac{e^{-j\frac{\pi}{N}kM}\left[e^{j\frac{\pi}{N}kM} - e^{-j\frac{\pi}{N}kM}\right]}{e^{-j\frac{\pi}{N}k}\left[e^{j\frac{\pi}{N}k} - e^{-j\frac{\pi}{N}k}\right]} \\
&= \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}
\end{aligned}
$$

DFT of length-M step in $\mathbb{C}^N$

$$X[k] = \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

- $X[0] = M$, from the definition of the sum
- $X[k] = 0$ if Mk/N integer ($0 < k < N$)
- $\angle X[k]$ linear in k (except at sign changes for the real part)

DFT of length-M step in $\mathbb{C}^N$

$$X[k] = \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

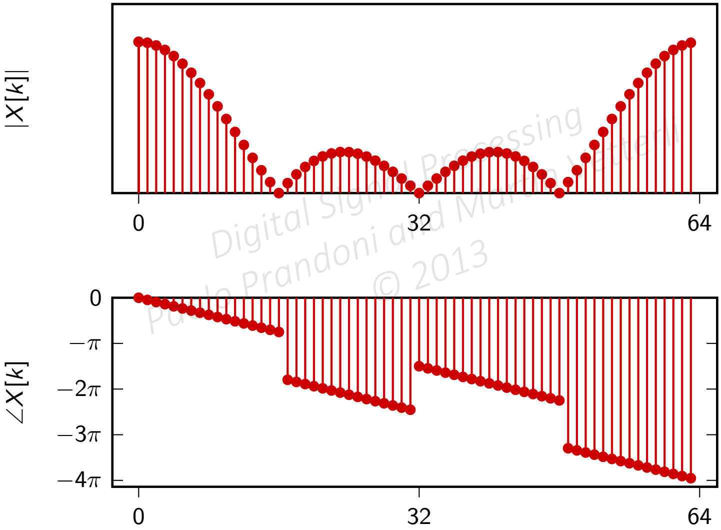
- $X[0] = M$, from the definition of the sum
- $X[k] = 0$ if Mk/N integer $(0 \leq k < N)$
- $\angle X[k]$ linear in k (except at sign changes for the real part)

DFT of length-M step in $\mathbb{C}^N$

$$X[k] = \frac{\sin\left(\frac{\pi}{N}Mk\right)}{\sin\left(\frac{\pi}{N}k\right)} e^{-j\frac{\pi}{N}(M-1)k}$$

- $X[0] = M$, from the definition of the sum
- $X[k] = 0$ if Mk/N integer $(0 \leq k < N)$
- $\angle X[k]$ linear in k (except at sign changes for the real part)

DFT of length-4 step in $\mathbb{C}^{64}$

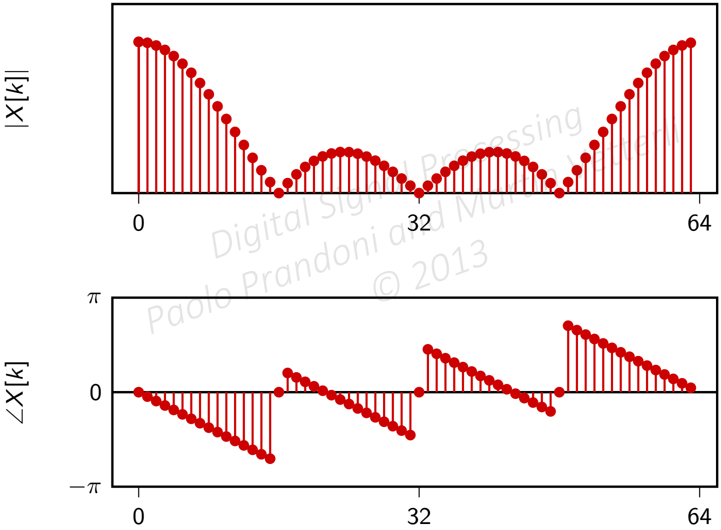


Wrapping the phase

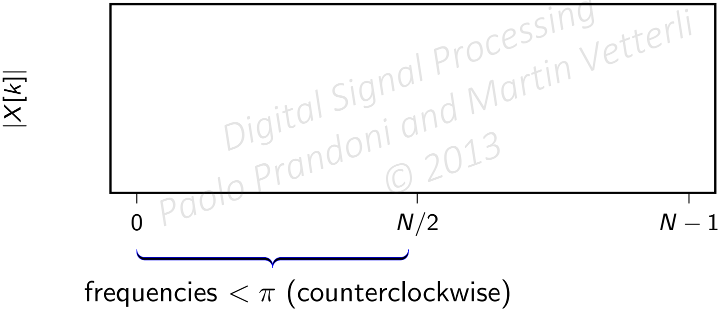
Often the phase is displayed "wrapped" over the $[-\pi, \pi]$ interval.

- most numerical packages return wrapped phase
- phase can be unwrapped by adding multiples of 2π

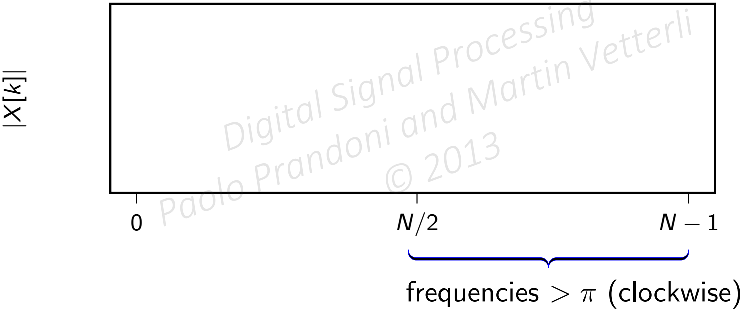
DFT of length-4 step in $\mathbb{C}^{64}$ (phase wrapped)



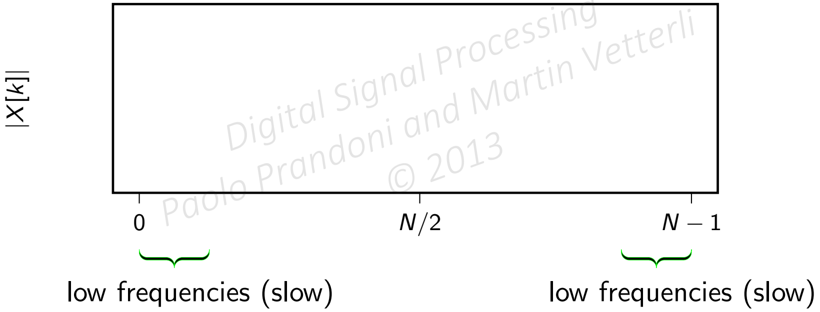
Interpreting a DFT plot



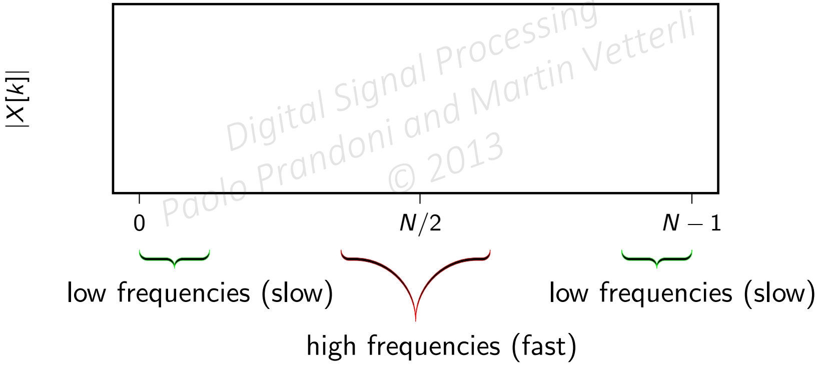
Interpreting a DFT plot



Interpreting a DFT plot

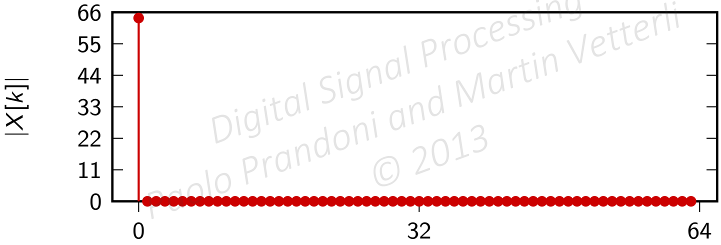


Interpreting a DFT plot



Interpreting a DFT plot

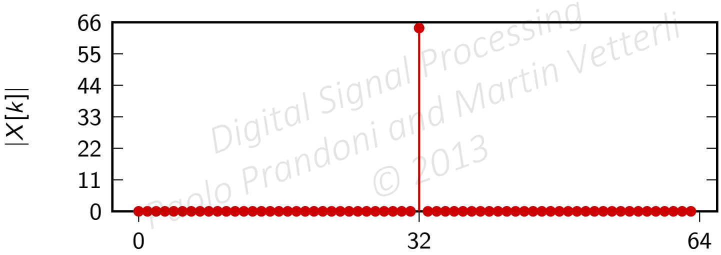
$x[n] = 1$ (slowest signal)



only lowest frequency

Interpreting a DFT plot

$$x[n] = \cos \pi n = (-1)^n \text{ (fastest signal)}$$



only highest frequency

Energy distribution

Recall Parseval (Module 3.3): $\|\mathbf{x}\|^2 = \sum |\alpha_k|^2$

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

square magnitude of k-th DFT coefficient proportional to signal's energy at frequency $\omega = (2\pi/N)k$

Energy distribution

Recall Parseval (Module 3.3): $\|\mathbf{x}\|^2 = \sum |\alpha_k|^2$

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

square magnitude of k-th DFT coefficient
proportional to signal's energy at frequency $\omega = (2\pi/N)k$

Energy distribution

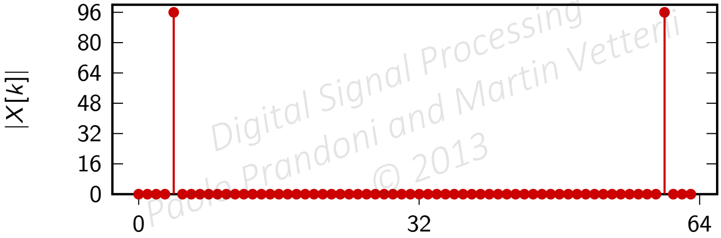
Recall Parseval (Module 3.3): $\|\mathbf{x}\|^2 = \sum |\alpha_k|^2$

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

square magnitude of k-th DFT coefficient
proportional to signal's energy at frequency $\omega = (2\pi/N)k$

Interpreting a DFT plot

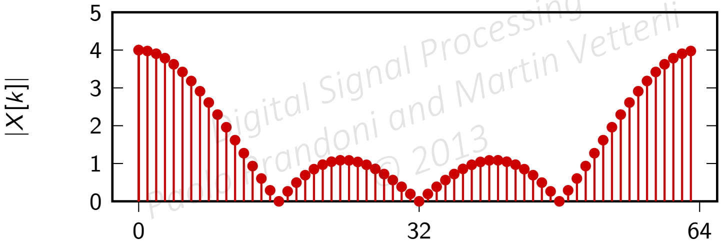
$$x[n] = 3\cos(2\pi/16\,n) \text{ (sinusoid)}$$



energy concentrated on single frequency
(counterclockwise and clockwise combine to give real signal)

Interpreting a DFT plot

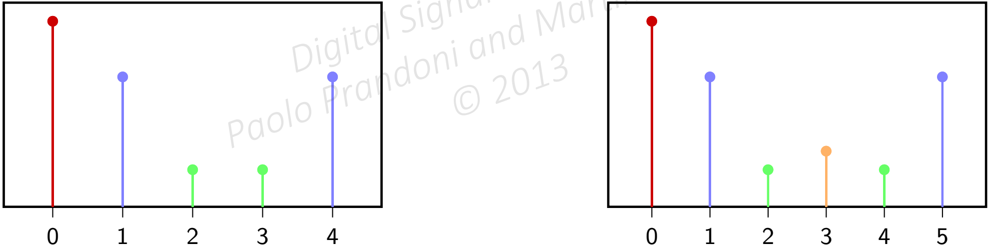
$$x[n] = u[n] - u[n-4] \text{ (step)}$$



energy mostly in low frequencies

For real signals the DFT is "symmetric" in magnitude:

$$|X[k]| = |X[N-k]| \text{ for } k = 1, 2, \ldots, \lfloor N/2 \rfloor$$

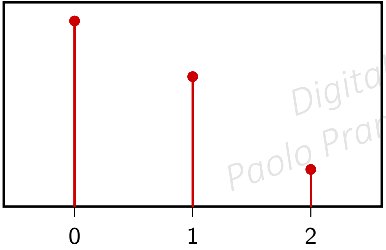


$N = 5$, odd length

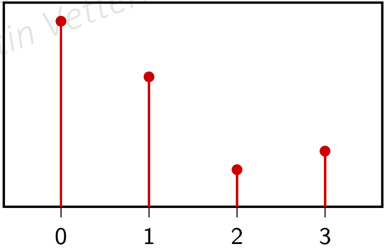
$N = 6$, even length

DFT of real signals

For real signals, magnitude plots need only $\lfloor N/2 \rfloor + 1$ points



$N = 5$, odd length



$N = 6$, even length

END OF MODULE 4.2

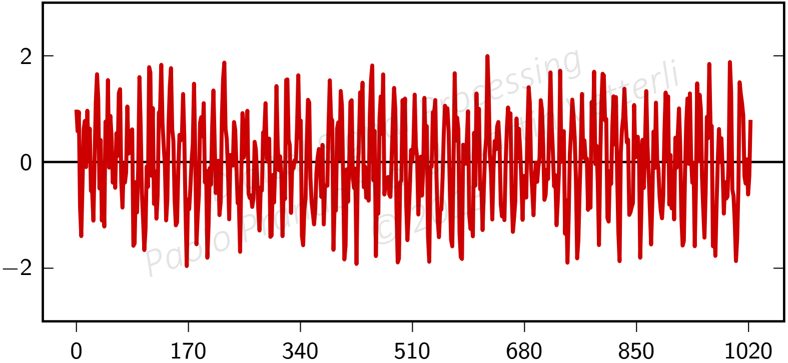
Digital Signal Processing

Module 4.3: DFT in practice

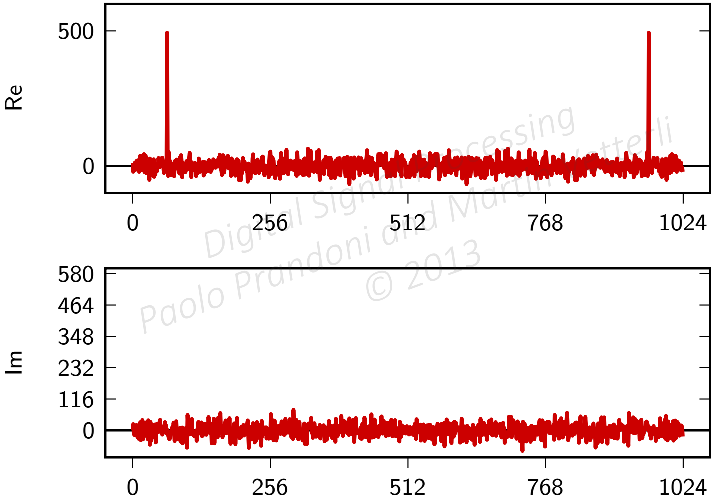
Overview:

- DFT analysis examples
- Labeling the DFT axes
- DFT synthesis
- Inherent periodicities and DFS

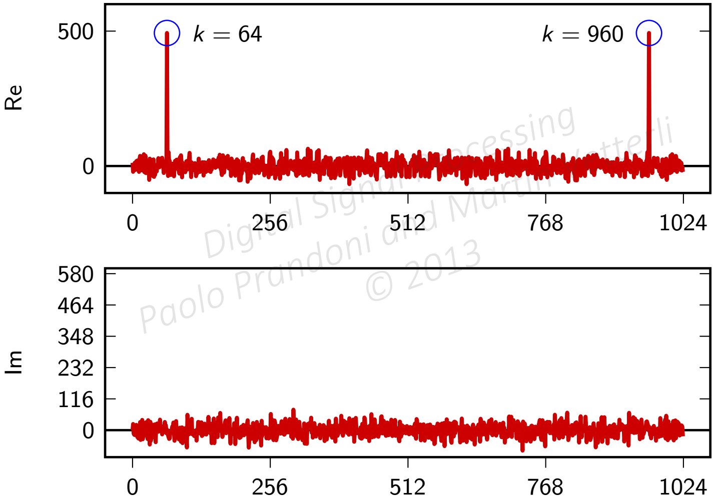
Mystery signal revisited



Mystery signal revisited



Mystery signal revisited



Mystery signal revisited

$$x[n] = \cos(\omega n + \phi) + \eta[n]$$

with

$$\begin{aligned} \phi &= 0 \\ \omega &= \frac{2\pi}{1024}64 \end{aligned}$$

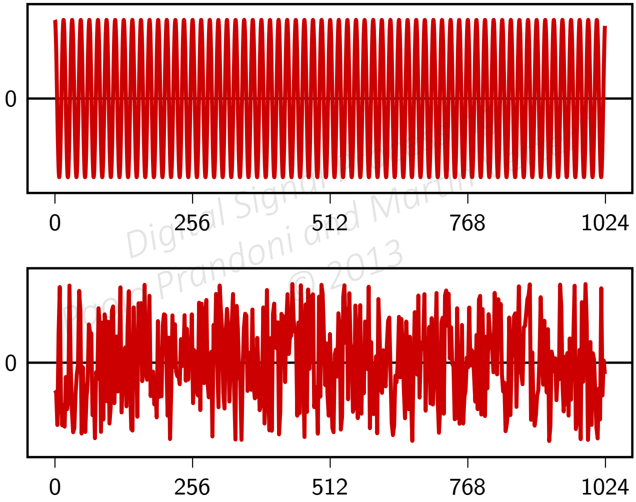
Mystery signal revisited

$$x[n] = \cos(\omega n + \phi) + \eta[n]$$

with

$$\begin{aligned} \phi &= 0 \\ \omega &= \frac{2\pi}{1024} 64 \end{aligned}$$

Mystery signal unveiled



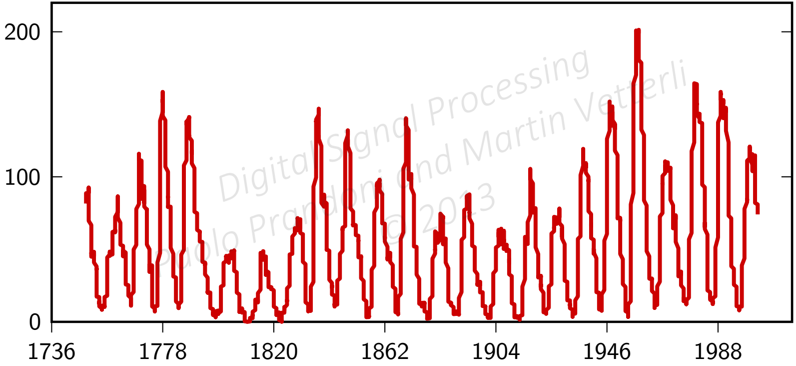
Solar spots

- sunspot number: $s = 10 \times \#$ of clusters $+ \#$ of spots
- data set from 1749 to 2003, 2904 months

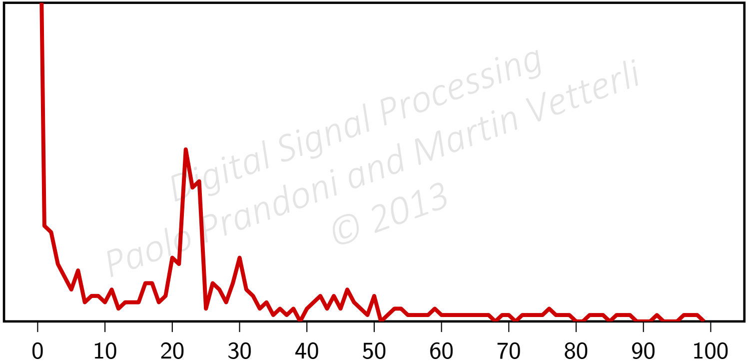
Solar spots

- sunspot number: $s = 10 \times \#$ of clusters $+ \#$ of spots
- data set from 1749 to 2003, 2904 months

Solar spots

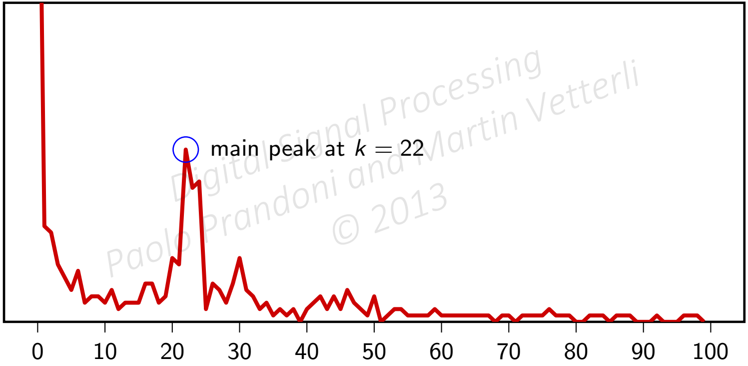


200
100
0
1736
1778
1820
1862
1904
1946
1988
Digital Signal Processing
Paolo Prandoni and Martin Vetterli
© 2013



0
10
20
30
40
50
60
70
80
90
100

Solar spots



Solar spots

- DFT main peak for $k = 22$
- 22 cycles over 2904 months
- period: $\dfrac{2904}{22} \approx 11$ years

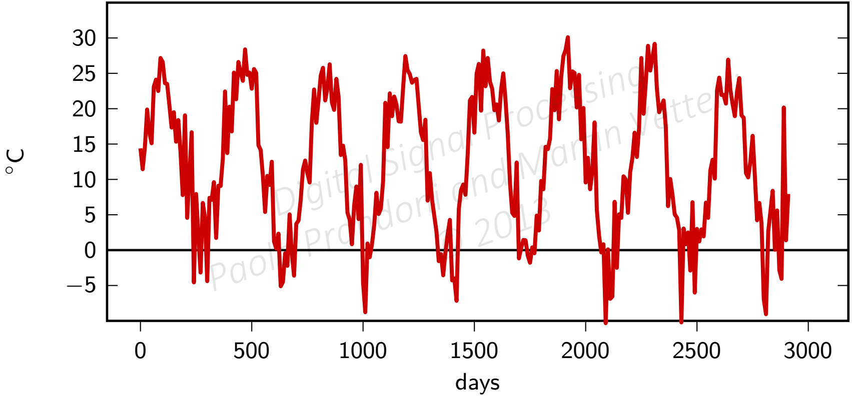
Solar spots

- DFT main peak for $k = 22$
- 22 cycles over 2904 months
- period: $\frac{2904}{22} \approx 11$ years

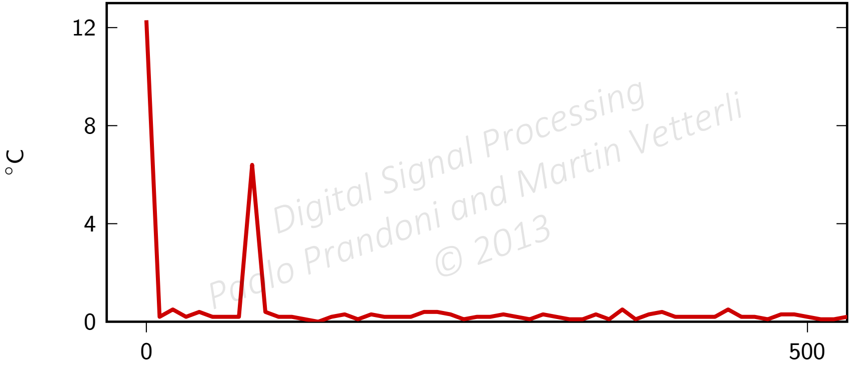
Solar spots

- DFT main peak for $k = 22$
- 22 cycles over 2904 months
- period: $\dfrac{2904}{22} \approx 11$ years

Daily temperature (2920 days)

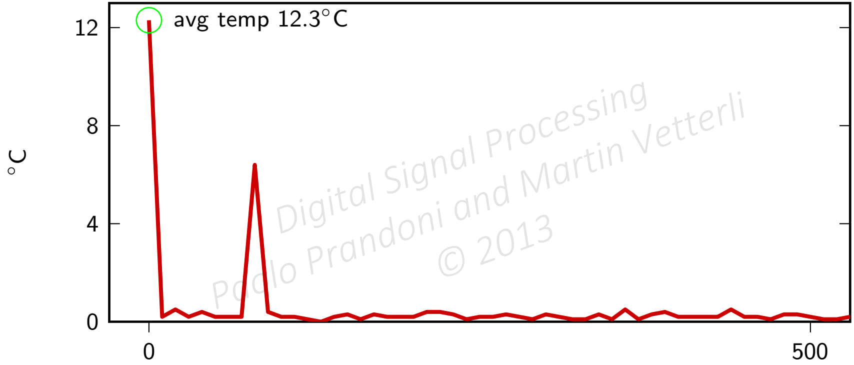


Daily temperature: DFT



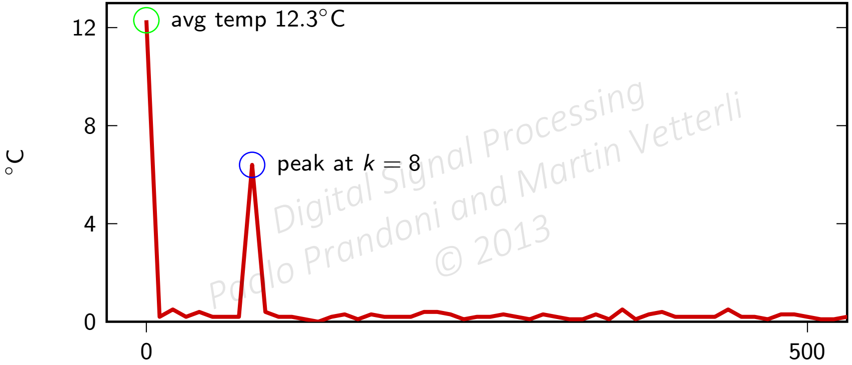
first few hundred DFT coefficients
(in magnitude and normalized by the length of the temperature vector)

Daily temperature: DFT



first few hundred DFT coefficients
(in magnitude and normalized by the length of the temperature vector)

Daily temperature: DFT



first few hundred DFT coefficients
(in magnitude and normalized by the length of the temperature vector)

Daily temperature

- average value (0-th DFT coefficient): $12.3^{\circ}\mathrm{C}$
- DFT main peak for $k = 8$, value $6.4^{\circ}\mathrm{C}$
- 8 cycles over 2920 days
- period: $\dfrac{2920}{8} = 365$ days
- temperature excursion: $12.3^{\circ}\mathrm{C} \pm 12.8^{\circ}\mathrm{C}$

Labeling the frequency axis

If we know the "clock" of the system T_s (see Module 2.2)

- fastest (positive) frequency is $\omega = \pi$
- sinusoid at $\omega = \pi$ needs two samples to do a full revolution
- time between samples: $T_s = 1/F_s$ seconds
- real-world period for fastest sinusoid: $2T_s$ seconds
- real-world frequency for fastest sinusoid: $F_s/2$ Hz

Labeling the frequency axis

If we know the "clock" of the system T_s (see Module 2.2)

- fastest (positive) frequency is $\omega = \pi$
- sinusoid at $\omega = \pi$ needs two samples to do a full revolution
- time between samples: $T_s = 1/F_s$ seconds
- real-world period for fastest sinusoid: $2T_s$ seconds
- real-world frequency for fastest sinusoid: $F_s/2$ Hz

Labeling the frequency axis

If we know the "clock" of the system T_s (see Module 2.2)

- fastest (positive) frequency is $\omega = \pi$
- sinusoid at $\omega = \pi$ needs two samples to do a full revolution
- time between samples: $T_s = 1/F_s$ seconds
- real-world period for fastest sinusoid: $2T_s$ seconds
- real-world frequency for fastest sinusoid: $F_s/2$ Hz

Labeling the frequency axis

If we know the “clock” of the system T_s (see Module 2.2)

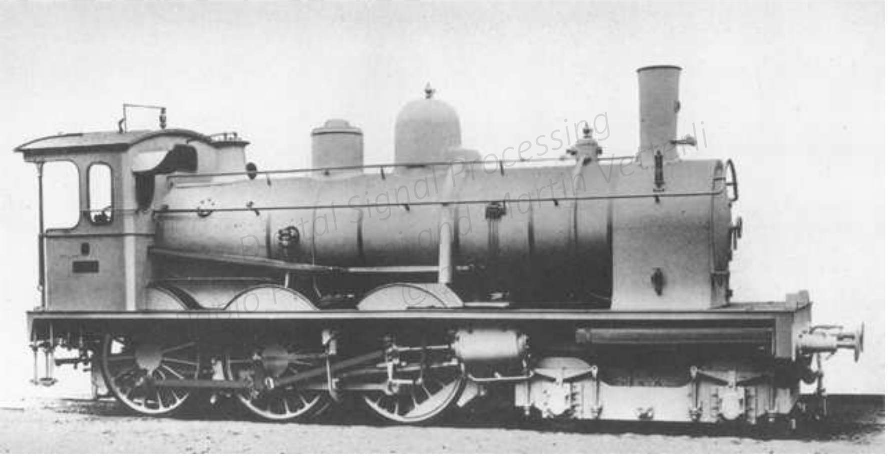
- fastest (positive) frequency is $\omega = \pi$
- sinusoid at $\omega = \pi$ needs two samples to do a full revolution
- time between samples: $T_s = 1/F_s$ seconds
- real-world period for fastest sinusoid: $2T_s$ seconds
- real-world frequency for fastest sinusoid: $F_s/2$ Hz

Labeling the frequency axis

If we know the "clock" of the system T_s (see Module 2.2)

- fastest (positive) frequency is $\omega = \pi$
- sinusoid at $\omega = \pi$ needs two samples to do a full revolution
- time between samples: $T_s = 1/F_s$ seconds
- real-world period for fastest sinusoid: $2T_s$ seconds
- real-world frequency for fastest sinusoid: $F_s/2$ Hz

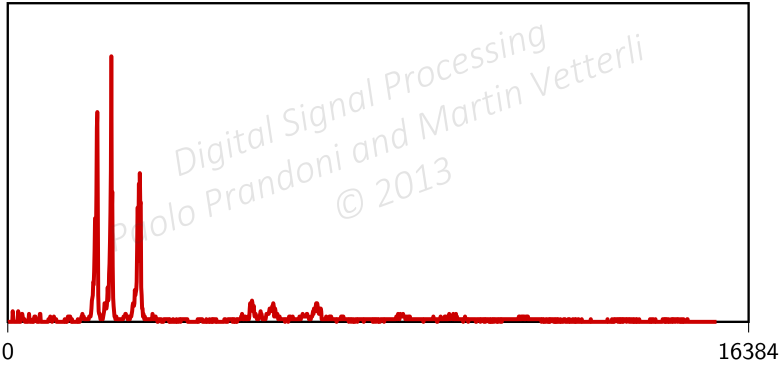
Example: train whistle



Play

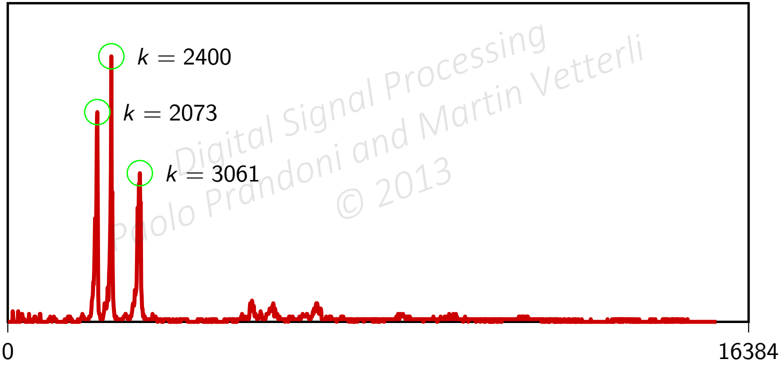
Example: train whistle

32768 samples (the "clock" of the system $F_s = 8000\text{Hz}$)



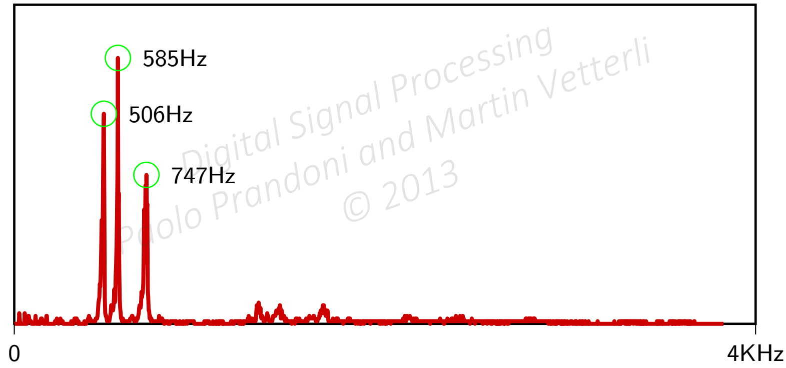
Example: train whistle

32768 samples (the “clock” of the system $F_s = 8000\text{Hz}$)



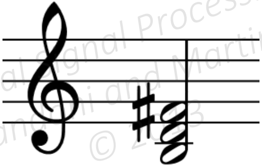
Example: train whistle

the "clock" of the system $F_s = 8000\text{Hz}$



Example: train whistle

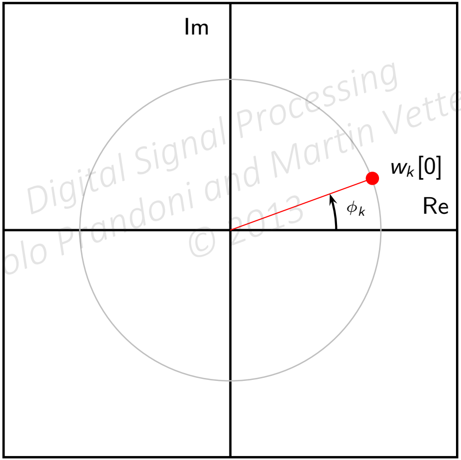
if we look up the frequencies:



B minor chord

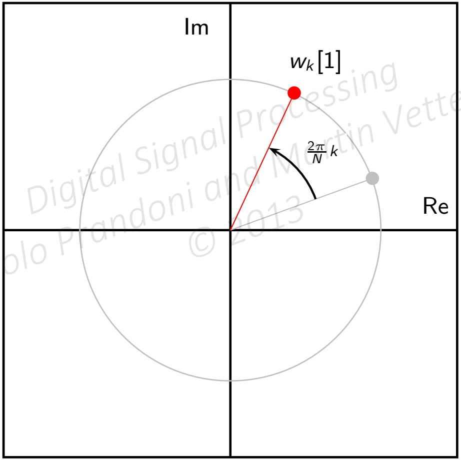
Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn+\phi_k)}$$



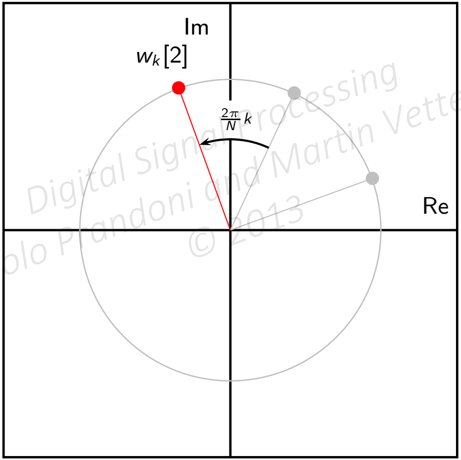
Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn + \phi_k)}$$



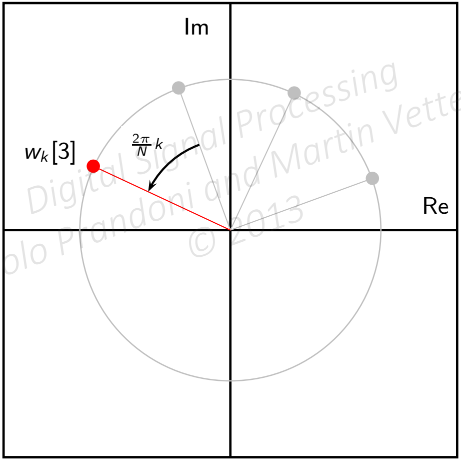
Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn+\phi_k)}$$



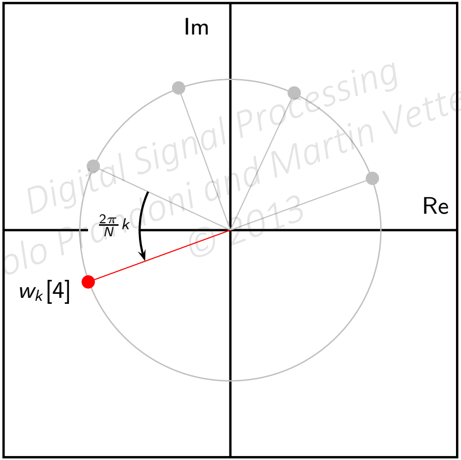
Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn + \phi_k)}$$



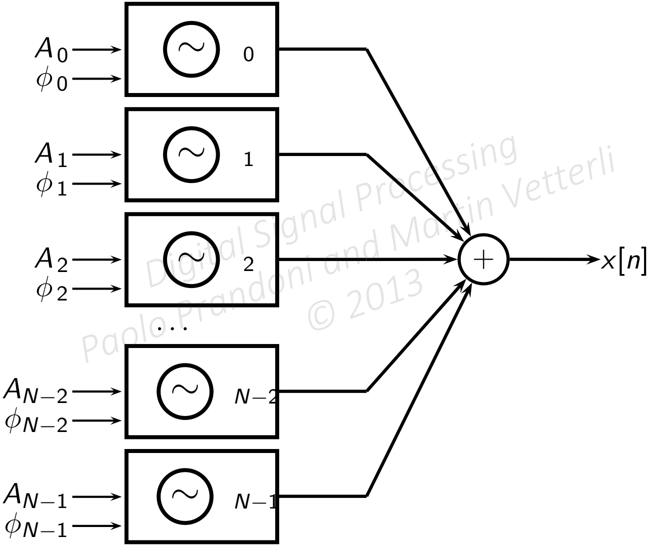
Synthesis: the sinusoidal generator

$$w_k[n] = e^{j(\frac{2\pi}{N}kn+\phi_k)}$$



Synthesis: the sinusoidal generator

DFT synthesis formula



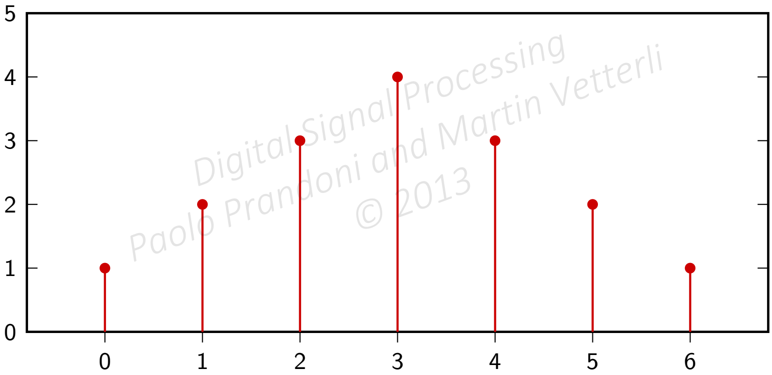
Initializing the machine

$$A_k = |X[k]|/N$$

$$\phi_k = \angle X[k]$$

Example

$$\mathbf{x} = [1\ 2\ 3\ 4\ 3\ 2\ 1]^T$$



Example

k	A_k	ϕ_k
0	2.2857	0.0000
1	0.7213	-2.6928
2	0.0440	0.8976
3	0.0919	-1.7952
4	0.0919	1.7952
5	0.0440	-0.8976
6	0.7213	2.6928

Example

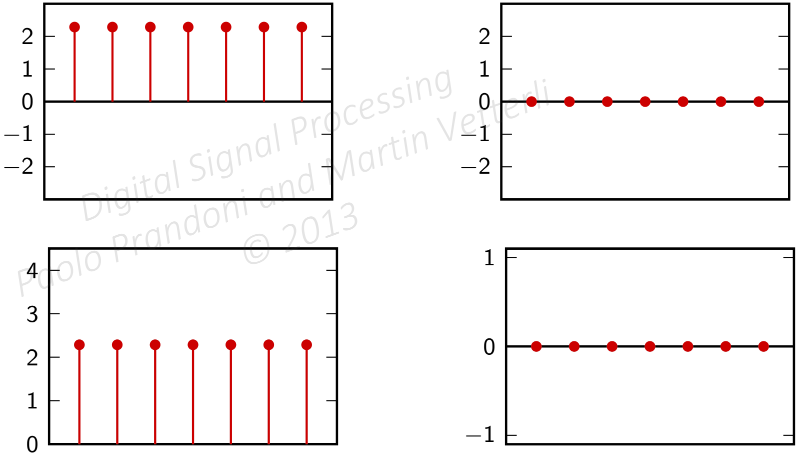
$k = 0$

Re

Im

$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$



Example

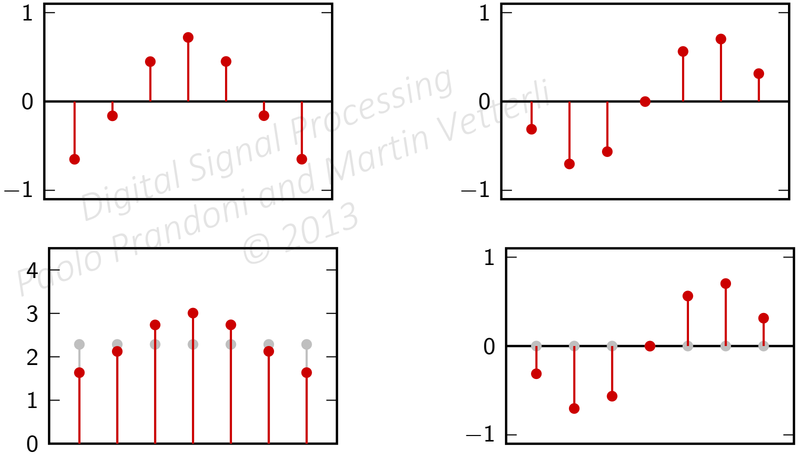
$k = 1$

Re

Im

$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$



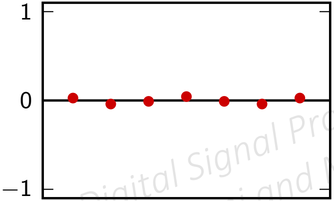
Example

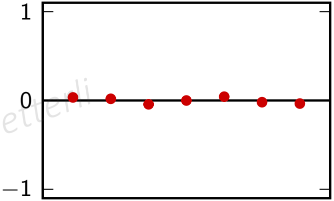
$k = 2$

Re

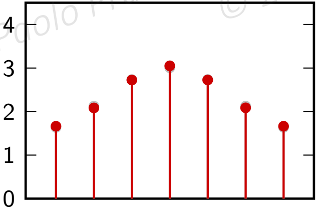
Im

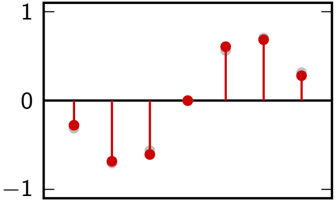
$A_k e^{j(\frac{2\pi}{N}kn + \phi_k)}$





$\sum A_k e^{j(\frac{2\pi}{N}kn + \phi_k)}$





Example

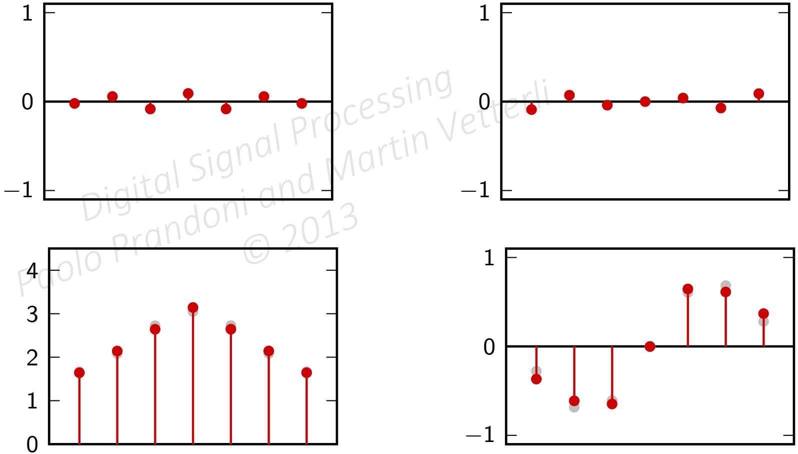
$k = 3$

Re

Im

$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

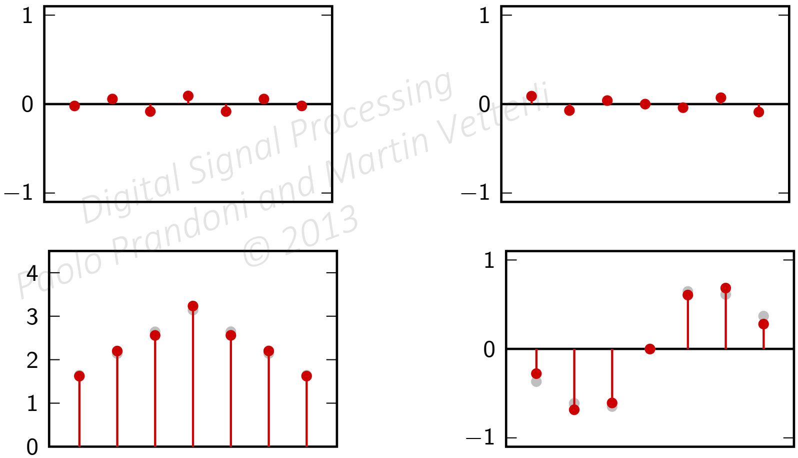


Example

$k = 4$ Re Im

$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

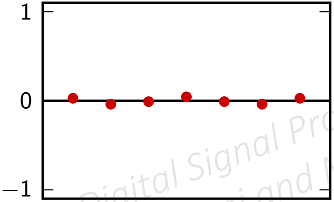
$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

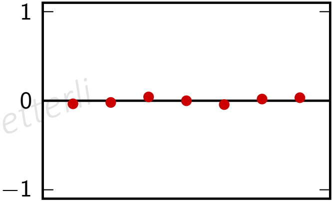


Example

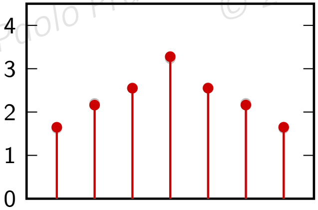
$k = 5$ Re Im

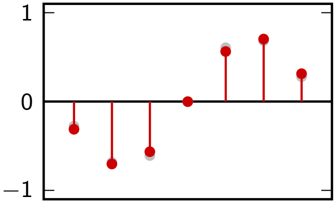
$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$





$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$



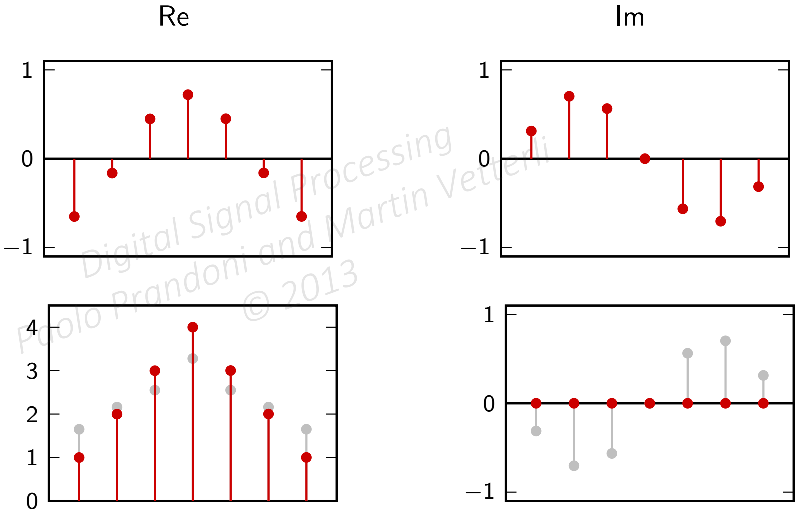


Example

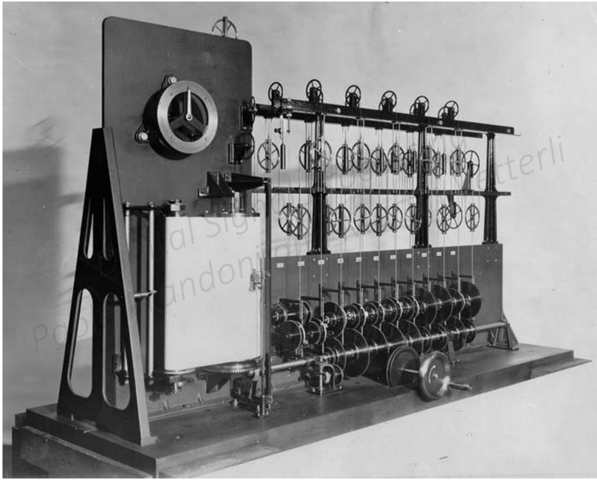
$k = 6$

$A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$

$\sum A_k e^{j(\frac{2\pi}{N}kn+\phi_k)}$



The machine before DSP



tide-predicting machine (originally invented by Lord Kelvin)

Running the machine too long...

$$x[n + N] = x[n]$$

output signal is N-periodic!

the synthesis formula:

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

produces an N-point signal in the time domain

the analysis formula:

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

produces N-point signal in the frequency domain

Inherent periodicities in the DFT

the synthesis formula:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n \in \mathbb{Z}$$

produces an N-periodic signal in the time domain

the analysis formula:

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

produces N-point signal in the frequency domain the synthesis formula:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}, \qquad n \in \mathbb{Z}$$

produces an *N*-periodic signal in the time domain

the analysis formula:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk}, \qquad k \in \mathbb{Z}$$

produces *N*-periodic signal in the frequency domain

Discrete Fourier Series (DFS)

DFS $=$ DFT with periodicity explicit

- the DFS maps an N-periodic signal onto an N-periodic sequence of Fourier coefficients
- the inverse DFS maps an N-periodic sequence of Fourier coefficients a set onto an N-periodic signal
- the DFS of an N-periodic signal is mathematically equivalent to the DFT of one period

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\tilde{X}[k]\right\} = \tilde{x}[n-M]$

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\tilde{X}[k]\right\} = \tilde{x}[n-M]$

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\tilde{X}[k]\right\} = \tilde{x}[n-M]$

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{e^{-j\frac{2\pi}{N}Mk}\,\tilde{X}[k]\right\} = \tilde{x}[n-M]$

Finite-length time shifts revisited

The DFS helps us understand how to define time shifts for finite-length signals.

For an N- periodic sequence $\tilde{x}[n]$:

- $\tilde{x}[n-M]$ is well-defined for all $M \in \mathbb{N}$
- $\text{DFS}\{\tilde{x}[n-M]\} = e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]$ (easy derivation)
- $\text{IDFS}\left\{\boxed{e^{-j\frac{2\pi}{N}Mk}}\,\tilde{X}[k]\right\} = \tilde{x}[n-M]$

delay factor

Finite-length time shifts revisited

For an N-point signal $x[n]$:

- $x[n-M]$ is *not* well-defined
- build $\tilde{x}[n] = x[n \mod N] \Rightarrow \tilde{X}[k] = X[k]$
- $\mathrm{IDFT}\left\{e^{-j\frac{2\pi}{N}Mk}X[k]\right\} = \mathrm{IDFS}\left\{e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]\right\} = \tilde{x}[n-M] = x[(n-M) \mod N]$
- shifts for finite-length signals are "naturally" circular (see Module 2.1)

Finite-length time shifts revisited

For an N-point signal $x[n]$:

- $x[n-M]$ is *not* well-defined
- build $\tilde{x}[n] = x[n \mod N] \Rightarrow \tilde{X}[k] = X[k]$
- $\text{IDFT}\left\{e^{-j\frac{2\pi}{N}Mk} X[k]\right\} = \text{IDFS}\left\{e^{-j\frac{2\pi}{N}Mk} \tilde{X}[k]\right\} = \tilde{x}[n-M] = x[(n-M) \mod N]$
- shifts for finite-length signals are "naturally" circular (see Module 2.1)

Finite-length time shifts revisited

For an N-point signal $x[n]$:

- $x[n-M]$ is *not* well-defined
- build $\tilde{x}[n] = x[n \mod N] \;\Rightarrow\; \tilde{X}[k] = X[k]$
- $\text{IDFT}\left\{e^{-j\frac{2\pi}{N}Mk}\, X[k]\right\} = \text{IDFS}\left\{e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]\right\} = \tilde{x}[n-M] = x[(n-M) \mod N]$
- shifts for finite-length signals are "naturally" circular (see Module 2.1)

Finite-length time shifts revisited

For an N-point signal $x[n]$:

- $x[n-M]$ is *not* well-defined
- build $\tilde{x}[n] = x[n \mod N] \;\Rightarrow\; \tilde{X}[k] = X[k]$
- $\text{IDFT}\left\{e^{-j\frac{2\pi}{N}Mk}\, X[k]\right\} = \text{IDFS}\left\{e^{-j\frac{2\pi}{N}Mk}\tilde{X}[k]\right\} = \tilde{x}[n-M] = x[(n-M) \mod N]$
- shifts for finite-length signals are “naturally” circular (see Module 2.1)

END OF MODULE 4.3

Digital Signal Processing

Module 4.4: To each its own: DFT, DFS, DTFT

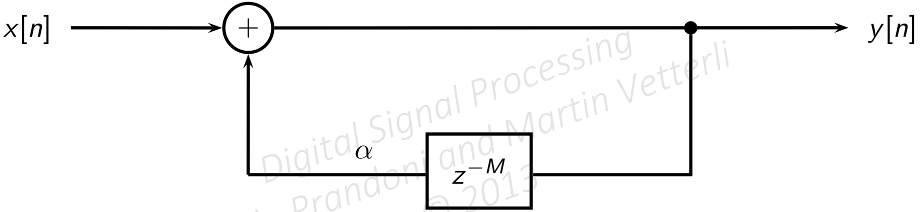
Overview:

- Karplus-Strong revisited Part I: the DFS
- Karplus-Strong revisited Part II: the DTFT

Periodic sequences: a bridge to infinite-length signals

- N-periodic sequence: N degrees of freedom
- DFS: only N Fourier coefficients capture all the information

Karplus-Strong revisited



$$y[n] = \alpha\, y[n-M] + x[n]$$

Karplus-Strong revisited

- choose a signal $\bar{x}[n]$ that is nonzero only for $0 \leq n < M$
- $\alpha = 1$ (for now)

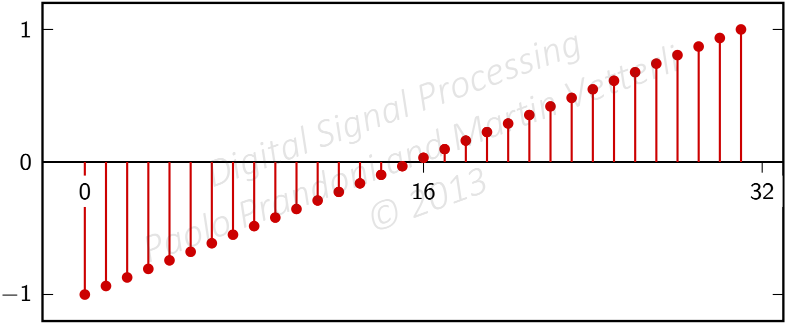
$$y[n] = \bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1], \bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1], \bar{x}[0], \bar{x}[1], \ldots$$

Karplus-Strong revisited

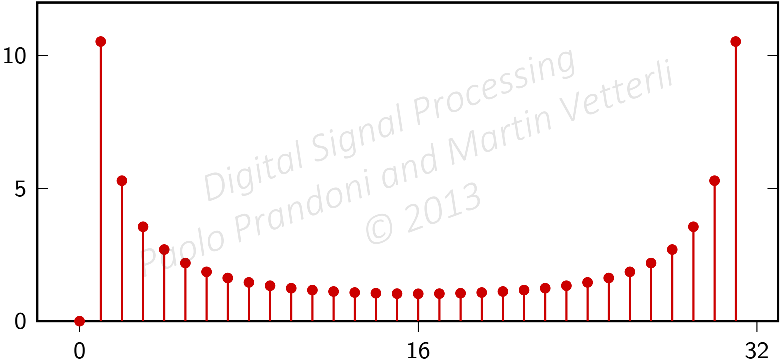
- choose a signal $\bar{x}[n]$ that is nonzero only for $0 \le n < M$
- $\alpha = 1$ (for now)

$$y[n] = \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{1^{\text{st}} \text{ period}}, \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{2^{\text{nd}} \text{ period}}, \underbrace{\bar{x}[0], \bar{x}[1], \ldots}_{\ldots}$$

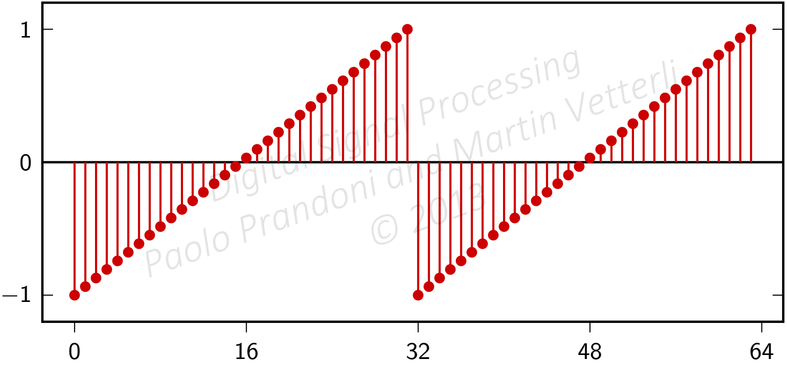
Example: 32-tap sawtooth wave



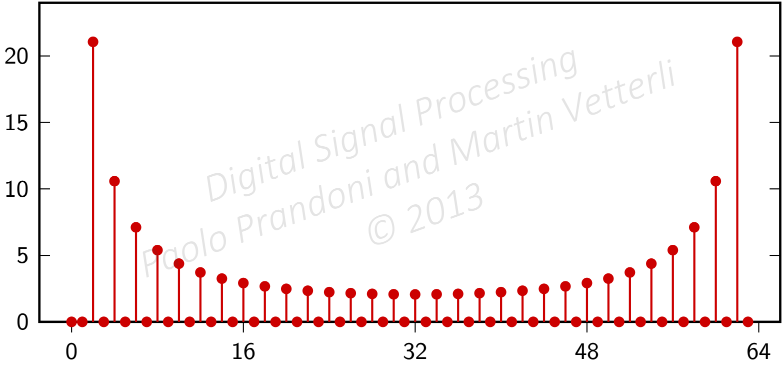
Example: DFT of 32-tap sawtooth wave



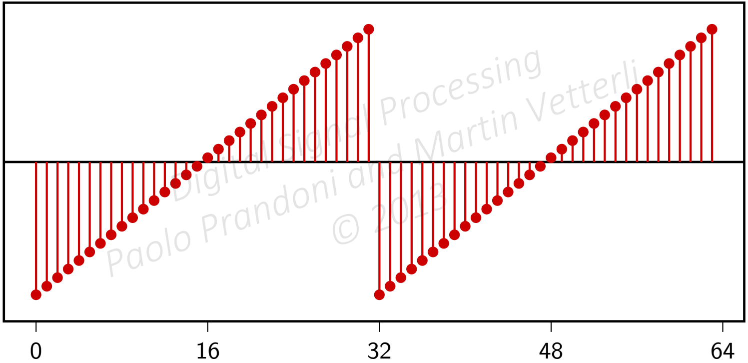
What if we take the DFT of two periods?



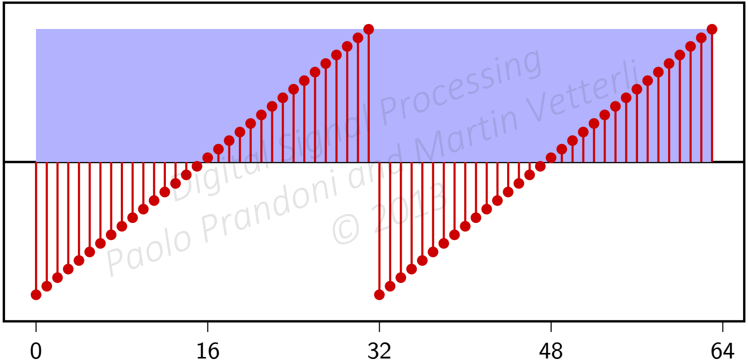
Example: 64-point DFT of two periods



DFT of two periods: intuition

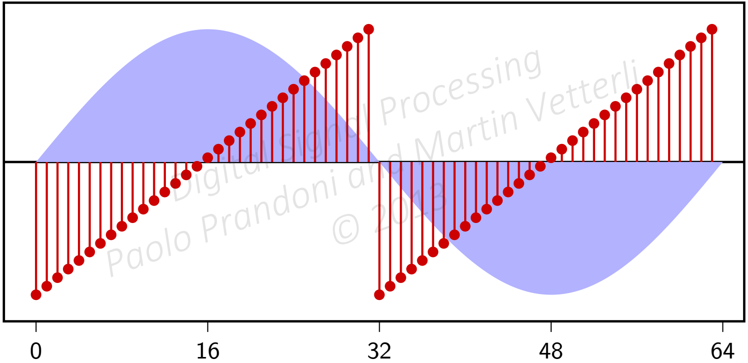


DFT of two periods: intuition



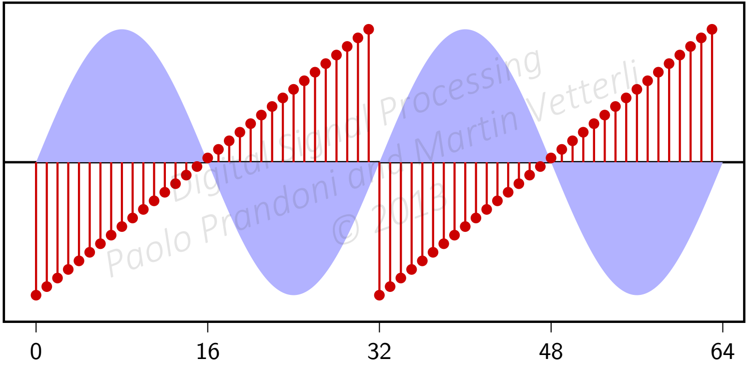
$k = 0$

DFT of two periods: intuition



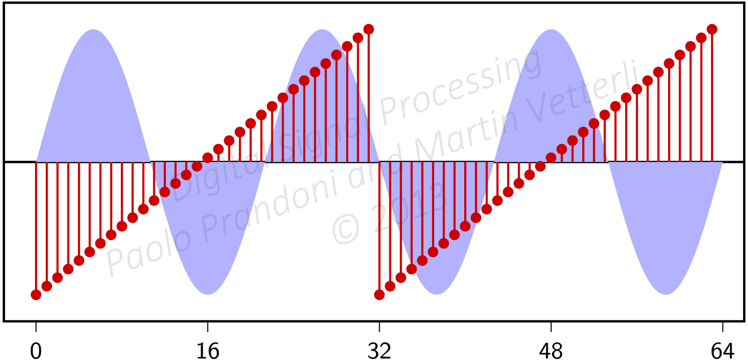
$k = 1$

DFT of two periods: intuition



$k = 2$

DFT of two periods: intuition



$k = 3$

DFT of L periods

$$X_L[k] = \sum_{n=0}^{LM-1} y[n]e^{-j\frac{2\pi}{LM}nk} \qquad k = 0, 1, 2 \ldots, LM-1$$

$$= \sum_{p=0}^{L-1}\sum_{n=0}^{M-1} y[n+pM]e^{-j\frac{2\pi}{LM}(n+pM)k}$$

$$= \sum_{p=0}^{L-1}\sum_{n=0}^{M-1} y[n]e^{-j\frac{2\pi}{LM}nk}e^{-j\frac{2\pi}{L}pk}$$

$$= \left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk}\right)\sum_{n=0}^{M-1} \bar{x}[n]\, e^{-j\frac{2\pi}{LM}nk}$$

DFT of L periods

$$X_L[k] = \sum_{n=0}^{LM-1} y[n] e^{-j\frac{2\pi}{LM}nk} \qquad k = 0, 1, 2 \ldots, LM-1$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n+pM] e^{-j\frac{2\pi}{LM}(n+pM)k}$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n] e^{-j\frac{2\pi}{LM}nk} e^{-j\frac{2\pi}{L}pk}$$

$$= \left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} \right) \sum_{n=0}^{M-1} \bar{x}[n]\, e^{-j\frac{2\pi}{LM}nk}$$

DFT of L periods

$$X_L[k] = \sum_{n=0}^{LM-1} y[n] e^{-j\frac{2\pi}{LM}nk} \qquad k = 0, 1, 2 \ldots, LM-1$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n+pM] e^{-j\frac{2\pi}{LM}(n+pM)k}$$

$$= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n] e^{-j\frac{2\pi}{LM}nk} e^{-j\frac{2\pi}{L}pk}$$

$$= \left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} \right) \sum_{n=0}^{M-1} \bar{x}[n] \, e^{-j\frac{2\pi}{LM}nk}$$

DFT of L periods

$$
\begin{aligned}
X_L[k] &= \sum_{n=0}^{LM-1} y[n] e^{-j\frac{2\pi}{LM}nk} \qquad\qquad k = 0, 1, 2 \ldots, LM-1 \\
&= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n+pM] e^{-j\frac{2\pi}{LM}(n+pM)k} \\
&= \sum_{p=0}^{L-1} \sum_{n=0}^{M-1} y[n] e^{-j\frac{2\pi}{LM}nk} e^{-j\frac{2\pi}{L}pk} \\
&= \left(\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} \right) \sum_{n=0}^{M-1} \bar{x}[n]\, e^{-j\frac{2\pi}{LM}nk}
\end{aligned}
$$

We've seen this before

$$\sum_{p=0}^{L-1} e^{-j\frac{2\pi}{L}pk} = \begin{cases} L & \text{if } k \text{ multiple of } L \\ 0 & \text{otherwise} \end{cases}$$

(remember the orthogonality proof for the DFT basis)

DFT of L periods

$$X_L[k] = \begin{cases} L\,\bar{X}[k/L] & \text{if } k = 0, L, 2L, 3L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

DFT and DFS

- again, all the spectral information for a periodic signal is contained in the DFT coefficients of a single period
- to stress the periodicity of the underlying signal, we use the term DFS

DFT and DFS

- again, all the spectral information for a periodic signal is contained in the DFT coefficients of a single period
- to stress the periodicity of the underlying signal, we use the term DFS

The situation so far

Fourier representation for signal classes:

- N-point finite-length: DFT
- N-point periodic: DFS
- infinite length: ?

The situation so far

Fourier representation for signal classes:

- N-point finite-length: DFT
- N-point periodic: DFS
- infinite length: ?

The situation so far

Fourier representation for signal classes:

- N-point finite-length: DFT
- N-point periodic: DFS
- infinite length: ?

Karplus-Strong revisited, part 2

- consider now $\alpha < 1$
- generated signal is infinite-length but not periodic:

$$y[n] = \bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1], \alpha\bar{x}[0], \alpha\bar{x}[1], \ldots, \alpha\bar{x}[M-1], \alpha^2\bar{x}[0], \alpha^2\bar{x}[1], \ldots$$

- what is a good spectral representation?

Karplus-Strong revisited, part 2

- consider now $\alpha < 1$
- generated signal is infinite-length but not periodic:

$$y[n] = \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{1^{\text{st}}\text{ period}}, \underbrace{\alpha\bar{x}[0], \alpha\bar{x}[1], \ldots, \alpha\bar{x}[M-1]}_{2^{\text{nd}}\text{ period}}, \underbrace{\alpha^2\bar{x}[0], \alpha^2\bar{x}[1], \ldots}_{\ldots}$$

- what is a good spectral representation?

- consider now $\alpha < 1$
- generated signal is infinite-length but not periodic:

$$y[n] = \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{1^{\text{st}}\text{ period}}, \underbrace{\alpha\bar{x}[0], \alpha\bar{x}[1], \ldots, \alpha\bar{x}[M-1]}_{2^{\text{nd}}\text{ period}}, \underbrace{\alpha^2\bar{x}[0], \alpha^2\bar{x}[1], \ldots}_{\ldots}$$

- what is a good spectral representation?

DFT of increasingly long signals

- Start with the DFT. What happens when $N \to \infty$?
- $(2\pi/N)k$ becomes denser in $[0, 2\pi]$...
- In the limit $(2\pi/N)k \to \omega$:

$$\sum_n x[n]\, e^{-j\omega n} \qquad \omega \in \mathbb{R}$$

DFT of increasingly long signals

- Start with the DFT. What happens when $N \to \infty$?
- $(2\pi/N)k$ becomes denser in $[0, 2\pi]$...
- In the limit $(2\pi/N)k \to \omega$:

$$\sum_{n} x[n]\, e^{-j\omega n} \qquad \omega \in \mathbb{R}$$

- Start with the DFT. What happens when $N \to \infty$?
- $(2\pi/N)k$ becomes denser in $[0, 2\pi]$...
- In the limit $(2\pi/N)k \to \omega$:

$$\sum_n x[n]\, e^{-j\omega n} \qquad \omega \in \mathbb{R}$$

Discrete-Time Fourier Transform (DTFT)

Formal definition:

- $x[n] \in \ell_2(\mathbb{Z})$
- define the *function* of $\omega \in \mathbb{R}$

$$F(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- inversion (when $F(\omega)$ exists):

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega) e^{j\omega n} d\omega, \qquad n \in \mathbb{Z}$$

Formal definition:

- $x[n] \in \ell_2(\mathbb{Z})$
- define the *function* of $\omega \in \mathbb{R}$

$$F(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- inversion (when $F(\omega)$ exists):

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega) e^{j\omega n} d\omega, \qquad n \in \mathbb{Z}$$

Formal definition:

- $x[n] \in \ell_2(\mathbb{Z})$
- define the *function* of $\omega \in \mathbb{R}$

$$F(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- inversion (when $F(\omega)$ exists):

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega) e^{j\omega n} d\omega, \qquad n \in \mathbb{Z}$$

DTFT periodicity and notation

- $F(\omega)$ is 2π-periodic
- to stress periodicity (and for other reasons) we will write

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- by convention, $X(e^{j\omega})$ is represented over $[-\pi, \pi]$

DTFT periodicity and notation

- $F(\omega)$ is 2π-periodic
- to stress periodicity (and for other reasons) we will write

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- by convention, $X(e^{j\omega})$ is represented over $[-\pi, \pi]$

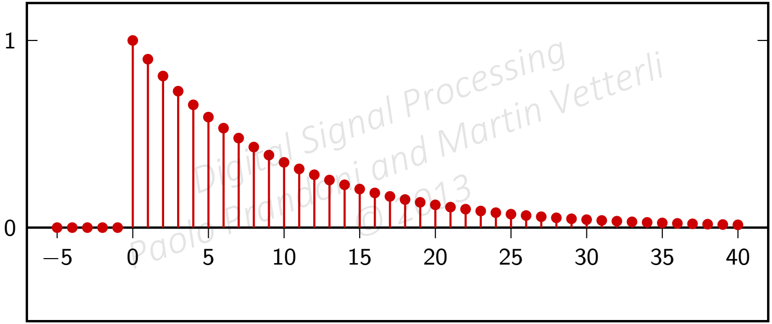
DTFT periodicity and notation

- $F(\omega)$ is 2π-periodic
- to stress periodicity (and for other reasons) we will write

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- by convention, $X(e^{j\omega})$ is represented over $[-\pi, \pi]$

$x[n] = \alpha^n \, u[n], \quad |\alpha| < 1$



DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

$$= \sum_{n=0}^{\infty} \alpha^n e^{-j\omega n}$$

$$= \sum_{n=0}^{\infty} (\alpha\, e^{-j\omega})^n$$

$$= \frac{1}{1 - \alpha e^{-j\omega}}$$

DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

$$
\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} \alpha^n\, e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} (\alpha\, e^{-j\omega})^n \\
&= \frac{1}{1 - \alpha e^{-j\omega}}
\end{aligned}
$$

DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

$$= \sum_{n=0}^{\infty} \alpha^n e^{-j\omega n}$$

$$= \sum_{n=0}^{\infty} (\alpha\, e^{-j\omega})^n$$

$$= \frac{1}{1 - \alpha e^{-j\omega}}$$

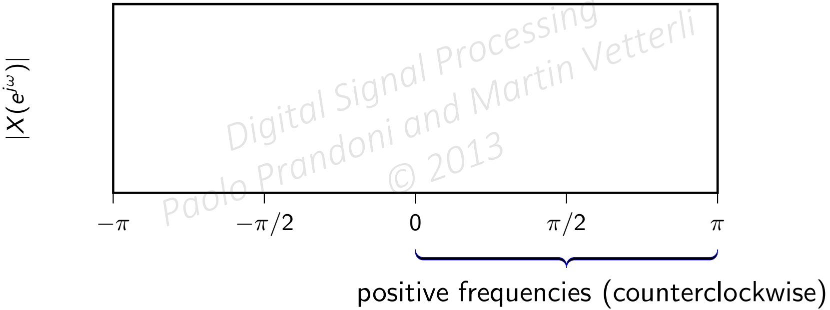
DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

$$
\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} \alpha^n\, e^{-j\omega n} \\
&= \sum_{n=0}^{\infty} (\alpha\, e^{-j\omega})^n \\
&= \frac{1}{1 - \alpha e^{-j\omega}}
\end{aligned}
$$

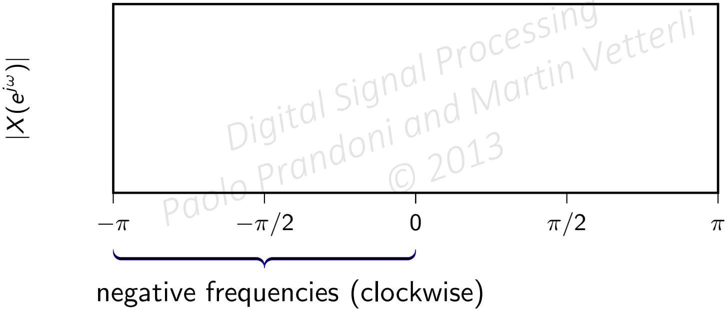
DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

$$|X(e^{j\omega})|^2 = \frac{1}{1 + \alpha^2 - 2\alpha \cos \omega}$$

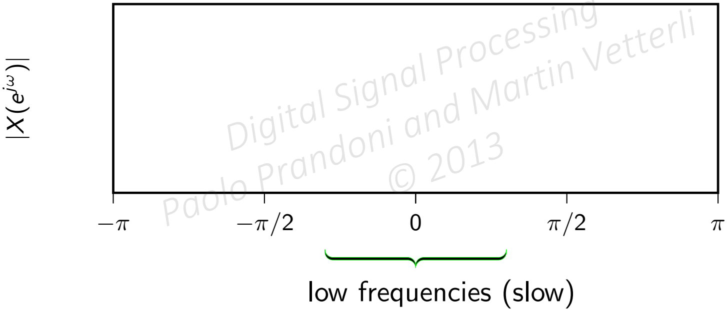
Plotting the DTFT



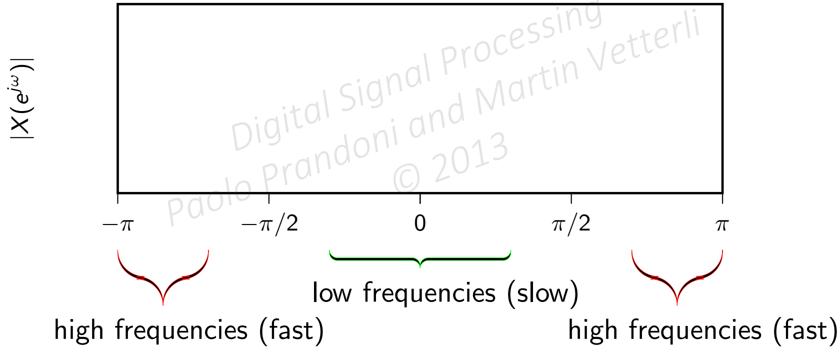
Plotting the DTFT



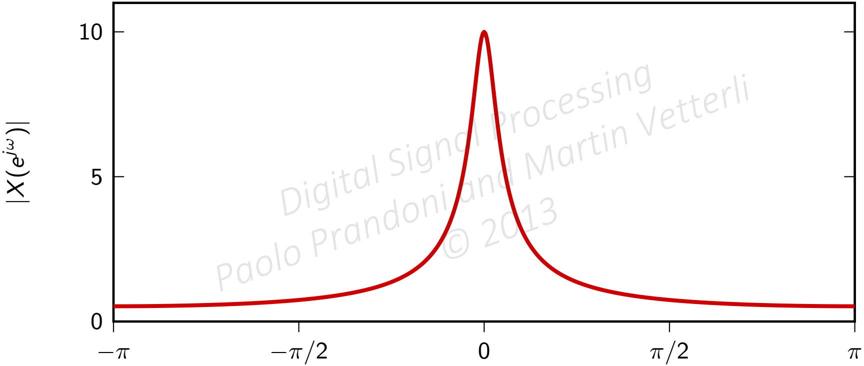
Plotting the DTFT



Plotting the DTFT

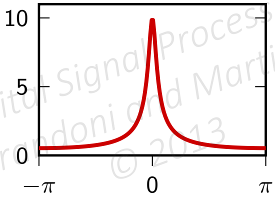


DTFT of $x[n] = \alpha^n u[n], \quad |\alpha| < 1$

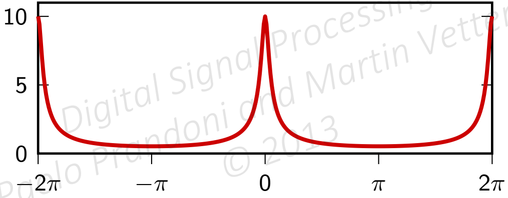


$\alpha = 0.9$

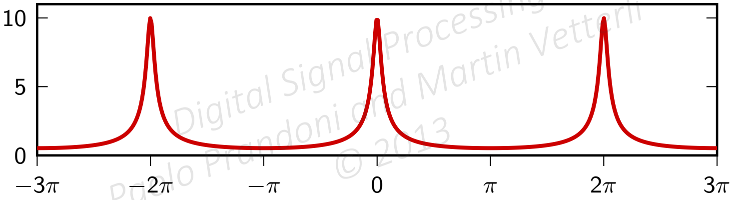
Remember the periodicity!



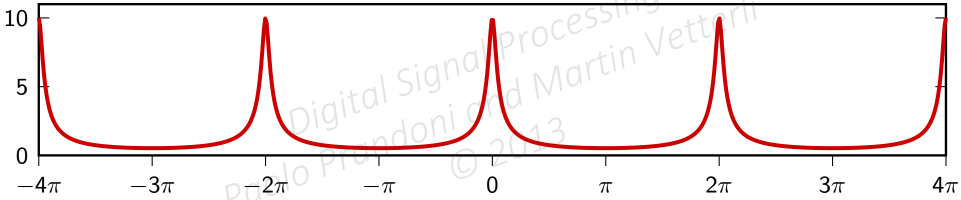
Remember the periodicity!



Remember the periodicity!

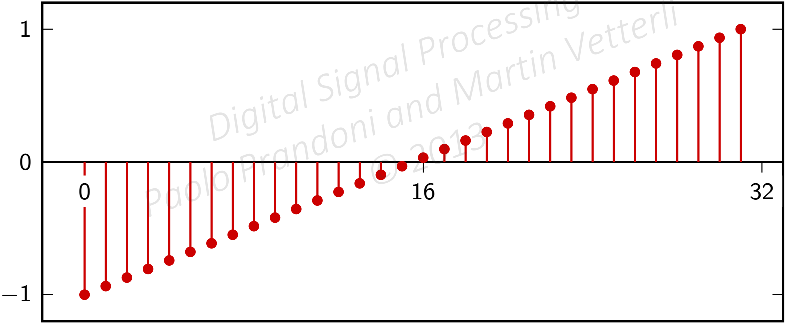


Remember the periodicity!



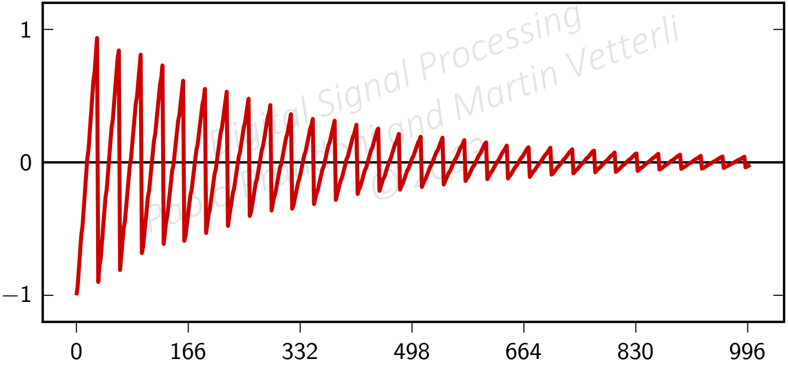
KS revisited, part 2: 32-tap sawtooth wave

$$x[n] = 2n/(M-1) - 1, \quad n = 0, 1, \ldots, M-1$$



KS revisited, part 2: decay $\alpha = 0.9$

$$y[n] = \alpha^{\lfloor n/M \rfloor}\,\bar{x}[n \mod M]\,u[n]$$



DTFT of KS signal

$$
\begin{aligned}
Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]\, e^{-j\omega n} \\
&= \sum_{p=0}^{\infty} \sum_{n=0}^{M-1} \alpha^p \bar{x}[n]\, e^{-j\omega(pM+n)} \\
&= \sum_{p=0}^{\infty} \alpha^p e^{-j\omega M p} \sum_{n=0}^{M-1} \bar{x}[n] e^{-j\omega n} \\
&= A(e^{j\omega M}) \bar{X}(e^{j\omega})
\end{aligned}
$$

DTFT of KS signal

$$
\begin{aligned}
Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]\, e^{-j\omega n} \\
&= \sum_{p=0}^{\infty} \sum_{n=0}^{M-1} \alpha^p \bar{x}[n]\, e^{-j\omega(pM+n)} \\
&= \sum_{p=0}^{\infty} \alpha^p e^{-j\omega M p} \sum_{n=0}^{M-1} \bar{x}[n] e^{-j\omega n} \\
&= A(e^{j\omega M}) \bar{X}(e^{j\omega})
\end{aligned}
$$

DTFT of KS signal

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty} y[n]\, e^{-j\omega n}$$

$$= \sum_{p=0}^{\infty} \sum_{n=0}^{M-1} \alpha^p \bar{x}[n]\, e^{-j\omega(pM+n)}$$

$$= \sum_{p=0}^{\infty} \alpha^p e^{-j\omega Mp} \sum_{n=0}^{M-1} \bar{x}[n] e^{-j\omega n}$$

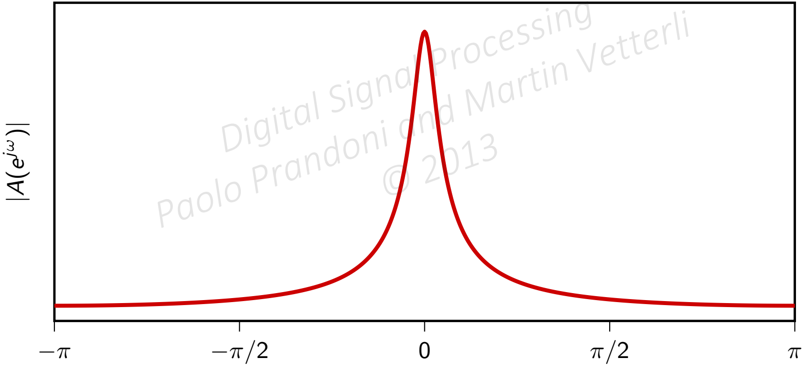
$$= A(e^{j\omega M})\bar{X}(e^{j\omega})$$

DTFT of KS signal

$$
\begin{aligned}
Y(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} y[n]\, e^{-j\omega n} \\
&= \sum_{p=0}^{\infty} \sum_{n=0}^{M-1} \alpha^p \bar{x}[n]\, e^{-j\omega(pM+n)} \\
&= \sum_{p=0}^{\infty} \alpha^p e^{-j\omega Mp} \sum_{n=0}^{M-1} \bar{x}[n] e^{-j\omega n} \\
&= A(e^{j\omega M}) \bar{X}(e^{j\omega})
\end{aligned}
$$

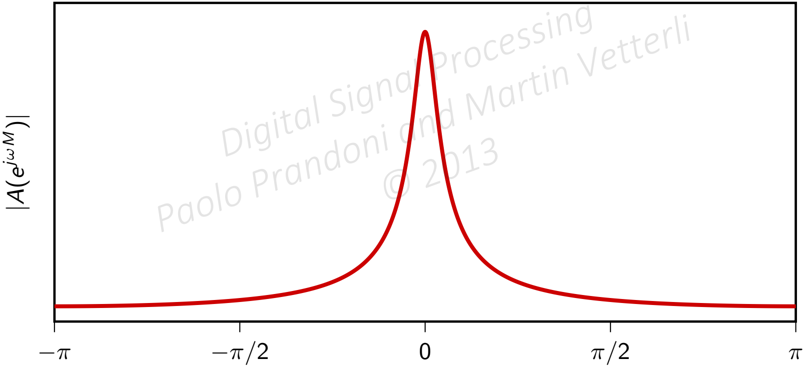
We know the first term

$$A(e^{j\omega}) = \text{DTFT}\left\{\alpha^n\, u[n]\right\} = \frac{1}{1-\alpha e^{-j\omega}}$$



We know the first term

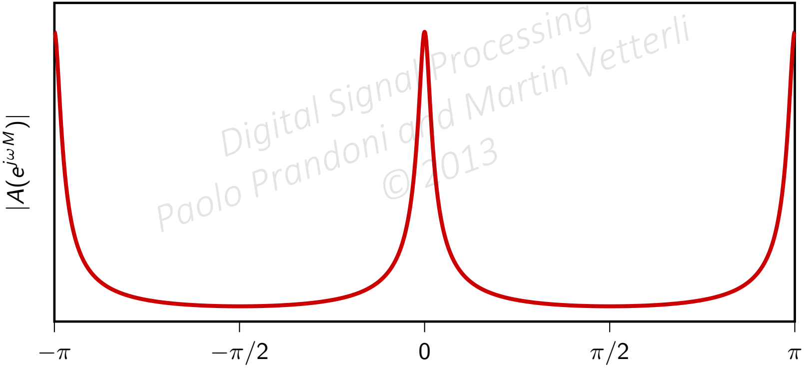
$A(e^{j\omega M})$ rescales the frequency axis: periodicity!



$M = 1$

We know the first term

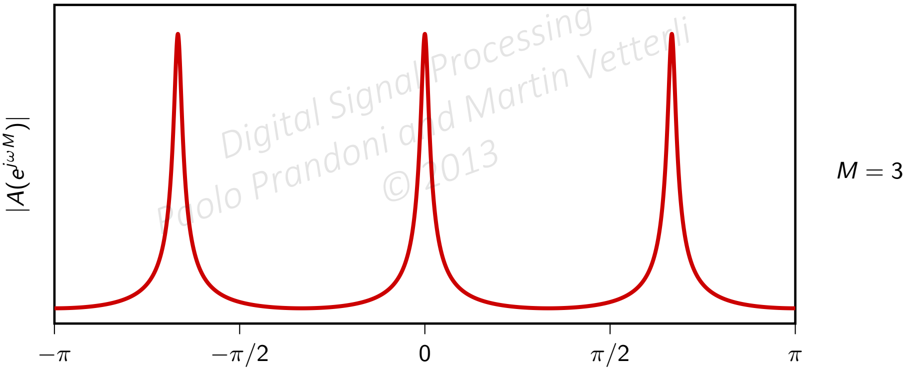
$A(e^{j\omega M})$ rescales the frequency axis: periodicity!



$M = 2$

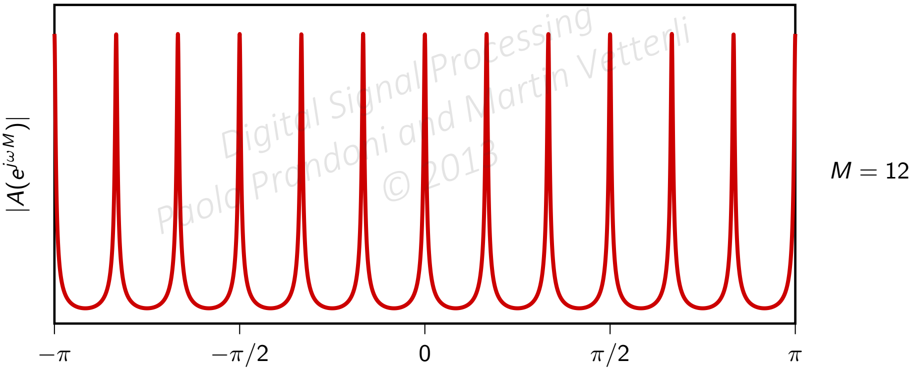
We know the first term

$A(e^{j\omega M})$ rescales the frequency axis: periodicity!

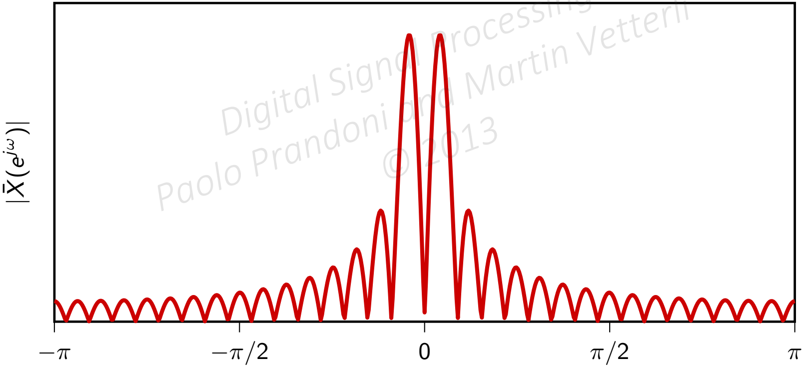


We know the first term

$A(e^{j\omega M})$ rescales the frequency axis: periodicity!

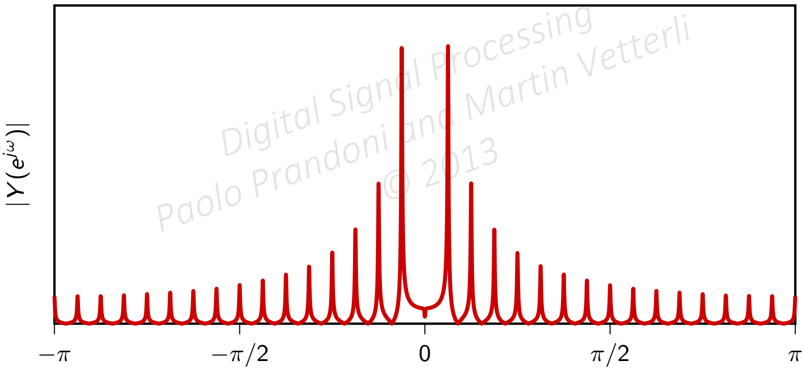


$$\bar{X}(e^{j\omega}) = e^{-j\omega}\left(\frac{M+1}{M-1}\right)\frac{1-e^{-j(M-1)\omega}}{(1-e^{-j\omega})^2} - \frac{1-e^{-j(M+1)\omega}}{(1-e^{-j\omega})^2}$$



DTFT of KS with decay

$$Y(e^{j\omega}) = A(e^{j\omega M})\bar{X}(e^{j\omega})$$



Recap of spectral representations:

- finite-length signals: DFT
- periodic sequences: DFS
- infinite sequences: DTFT

Recap of spectral representations:

- finite-length signals: DFT
- periodic sequences: DFS
- infinite sequences: DTFT

Recap of spectral representations:

- finite-length signals: DFT
- periodic sequences: DFS
- infinite sequences: DTFT

END OF MODULE 4.4

Digital Signal Processing

Module 4.5: DTFT: intuition and properties

Overview:

- Existence
- Properties
- DTFT as basis expansion

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- when does it exist?
- is it a change of basis?

Discrete-Time Fourier Transform

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}$$

- when does it exist?
- is it a change of basis?

Existence easy for absolutely summable sequences

$$|X(e^{j\omega})| = |\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}|$$

$$\leq \sum_{n=-\infty}^{\infty} |x[n]\, e^{-j\omega n}|$$

$$= \sum_{n=-\infty}^{\infty} |x[n]|$$

$$< \infty$$

Existence easy for absolutely summable sequences

$$
\begin{aligned}
|X(e^{j\omega})| &= \left| \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \right| \\
&\leq \sum_{n=-\infty}^{\infty} \left| x[n]\, e^{-j\omega n} \right| \\
&= \sum_{n=-\infty}^{\infty} |x[n]| \\
&< \infty
\end{aligned}
$$

Existence easy for absolutely summable sequences

$$
\begin{aligned}
|X(e^{j\omega})| &= \Big| \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \Big| \\
&\leq \sum_{n=-\infty}^{\infty} |x[n]\, e^{-j\omega n}| \\
&= \sum_{n=-\infty}^{\infty} |x[n]| \\
&< \infty
\end{aligned}
$$

Existence easy for absolutely summable sequences

$$\begin{aligned}
|X(e^{j\omega})| &= \left|\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}\right| \\
&\leq \sum_{n=-\infty}^{\infty} \left|x[n]\, e^{-j\omega n}\right| \\
&= \sum_{n=-\infty}^{\infty} |x[n]| \\
&< \infty
\end{aligned}$$

Inversion easy for absolutely summable sequences

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{j\omega n} d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{k=-\infty}^{\infty} x[k]\, e^{-j\omega k}\right) e^{j\omega n} d\omega$$

$$= \sum_{k=-\infty}^{\infty} x[k] \int_{-\pi}^{\pi} \frac{e^{j\omega(n-k)}}{2\pi} d\omega$$

$$= x[n]$$

Inversion easy for absolutely summable sequences

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{j\omega n} d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{k=-\infty}^{\infty} x[k]\, e^{-j\omega k}\right) e^{j\omega n} d\omega$$

$$= \sum_{k=-\infty}^{\infty} x[k] \int_{-\pi}^{\pi} \frac{e^{j\omega(n-k)}}{2\pi} d\omega$$

$$= x[n]$$

Inversion easy for absolutely summable sequences

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{j\omega n} d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(\sum_{k=-\infty}^{\infty} x[k]\, e^{-j\omega k} \right) e^{j\omega n} d\omega$$

$$= \sum_{k=-\infty}^{\infty} x[k] \int_{-\pi}^{\pi} \frac{e^{j\omega(n-k)}}{2\pi} d\omega$$

$$= x[n]$$

A formal change of basis

- formally DTFT is an inner product in $\mathbb{C}^\infty$:

$$\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} = \langle e^{j\omega n}, x[n] \rangle$$

- "basis" is an infinite, uncountable basis: $\{e^{j\omega n}\}_{\omega \in \mathbb{R}}$
- something "breaks down": we start with sequences but the transform is a function
- we used absolutely summable sequences but DTFT exists for all square-summable sequences (proof is rather technical)

A formal change of basis

- formally DTFT is an inner product in $\mathbb{C}^\infty$:

$$\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} = \langle e^{j\omega n}, x[n] \rangle$$

- "basis" is an infinite, uncountable basis: $\{e^{j\omega n}\}_{\omega \in \mathbb{R}}$
- something "breaks down": we start with sequences but the transform is a function
- we used absolutely summable sequences but DTFT exists for all square-summable sequences (proof is rather technical)

A formal change of basis

- formally DTFT is an inner product in $\mathbb{C}^\infty$:

$$\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} = \langle e^{j\omega n}, x[n] \rangle$$

- "basis" is an infinite, uncountable basis: $\{e^{j\omega n}\}_{\omega \in \mathbb{R}}$
- something "breaks down": we start with sequences but the transform is a function
- we used absolutely summable sequences but DTFT exists for all square-summable sequences (proof is rather technical)

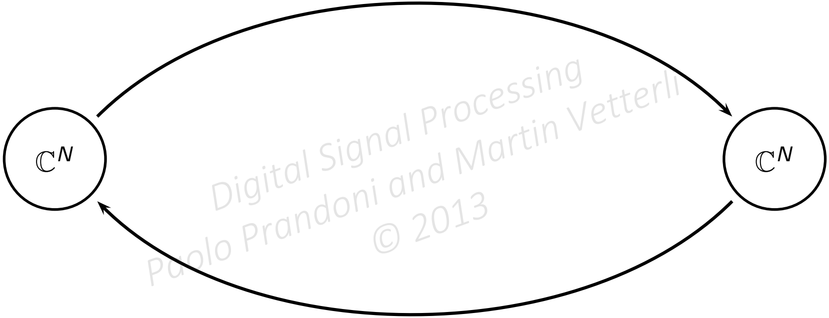
A formal change of basis

- formally DTFT is an inner product in $\mathbb{C}^\infty$:

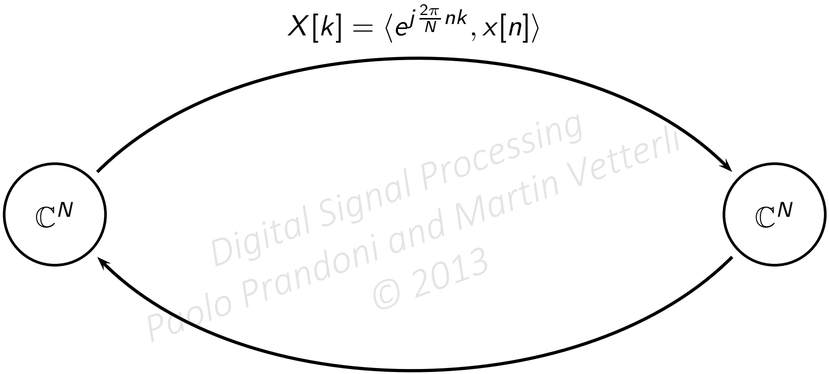
$$\sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} = \langle e^{j\omega n}, x[n] \rangle$$

- "basis" is an infinite, uncountable basis: $\{e^{j\omega n}\}_{\omega \in \mathbb{R}}$
- something "breaks down": we start with sequences but the transform is a function
- we used absolutely summable sequences but DTFT exists for all square-summable sequences (proof is rather technical)

Review: DFT



Review: DFT



Review: DFT

$$\begin{tikzcd}[column sep=huge]
\mathbb{C}^N \arrow[r, bend left, "X[k] = \langle e^{j\frac{2\pi}{N}nk}, x[n]\rangle"] & \mathbb{C}^N \arrow[l, bend left, "x[n] = (1/N)\sum X[k]\, e^{j\frac{2\pi}{N}nk}"]
\end{tikzcd}$$

Review: DFT

$$X[k] = \langle e^{j\frac{2\pi}{N}nk}, x[n] \rangle$$

$$\begin{tikzcd} \mathbb{C}^N \arrow[r, bend left, "X{[k]} = \langle e^{j\frac{2\pi}{N}nk}{,}\, x{[n]} \rangle"] & \mathbb{C}^N \arrow[l, bend left, "x{[n]} = (1/N)\sum X{[k]}\, e^{j\frac{2\pi}{N}nk}"] \end{tikzcd}$$

basis: $\{e^{j\frac{2\pi}{N}nk}\}_k$

$$x[n] = (1/N) \sum X[k]\, e^{j\frac{2\pi}{N}nk}$$

Review: DFS

$$\tilde{X}[k] = \langle e^{j\frac{2\pi}{N}nk}, \tilde{x}[n] \rangle$$

$$\begin{tikzcd}
\tilde{\mathbb{C}}^N \arrow[rr, bend left] & \text{basis: } \{e^{j\frac{2\pi}{N}nk}\}_k & \mathbb{C}^N \arrow[ll, bend left]
\end{tikzcd}$$

$$\tilde{x}[n] = (1/N) \sum \tilde{X}[k]\, e^{j\frac{2\pi}{N}nk}$$

DTFT

$$\begin{tikzcd}[column sep=huge] \ell_2(\mathbb{Z}) \arrow[r, bend left, "X(e^{j\omega}) = \langle e^{j\omega n}, x[n]\rangle"] \arrow[r, phantom, "\text{``basis'': } \{e^{j\omega n}\}_{\omega}"] & L_2([-\pi, \pi]) \arrow[l, bend left, "x[n] = (1/2\pi)\int X(e^{j\omega})e^{j\omega n}d\omega"] \end{tikzcd}$$

DTFT properties

- linearity

$$\text{DTFT}\{\alpha x[n] + \beta y[n]\} = \alpha X(e^{j\omega}) + \beta Y(e^{j\omega})$$

- time shift

$$\text{DTFT}\{x[n-M]\} = e^{-j\omega M} X(e^{j\omega})$$

- modulation (dual)

$$\text{DTFT}\{e^{j\omega_0 n} x[n]\} = X(e^{j(\omega-\omega_0)})$$

DTFT properties

- linearity

$$\text{DTFT}\{\alpha x[n] + \beta y[n]\} = \alpha X(e^{j\omega}) + \beta Y(e^{j\omega})$$

- time shift

$$\text{DTFT}\{x[n-M]\} = e^{-j\omega M} X(e^{j\omega})$$

- modulation (dual)

$$\text{DTFT}\{e^{j\omega_0 n} x[n]\} = X(e^{j(\omega-\omega_0)})$$

- linearity

$$\text{DTFT}\{\alpha x[n] + \beta y[n]\} = \alpha X(e^{j\omega}) + \beta Y(e^{j\omega})$$

- time shift

$$\text{DTFT}\{x[n-M]\} = e^{-j\omega M} X(e^{j\omega})$$

- modulation (dual)

$$\text{DTFT}\{e^{j\omega_0 n} x[n]\} = X(e^{j(\omega-\omega_0)})$$

DTFT properties

- time reversal

$$\text{DTFT}\{x[-n]\} = X(e^{-j\omega})$$

- conjugation

$$\text{DTFT}\{x^*[n]\} = X^*(e^{-j\omega})$$

DTFT properties

- time reversal

$$\text{DTFT}\{x[-n]\} = X(e^{-j\omega})$$

- conjugation

$$\text{DTFT}\{x^*[n]\} = X^*(e^{-j\omega})$$

Some particular cases:

- if $x[n]$ is symmetric, the DTFT is symmetric:

$$x[n] = x[-n] \Longleftrightarrow X(e^{j\omega}) = X(e^{-j\omega})$$

- if $x[n]$ is real, the DTFT is Hermitian-symmetric:

$$x[n] = x^*[n] \Longleftrightarrow X(e^{j\omega}) = X^*(e^{-j\omega})$$

- special case: if $x[n]$ is real, the magnitude of the DTFT is symmetric:

$$x[n] \in \mathbb{R} \Longrightarrow |X(e^{j\omega})| = |X(e^{-j\omega})|$$

- more special case: if $x[n]$ is real and symmetric, $X(e^{j\omega})$ is also real and symmetric

Some particular cases:

- if $x[n]$ is symmetric, the DTFT is symmetric:

$$x[n] = x[-n] \iff X(e^{j\omega}) = X(e^{-j\omega})$$

- if $x[n]$ is real, the DTFT is Hermitian-symmetric:

$$x[n] = x^*[n] \iff X(e^{j\omega}) = X^*(e^{-j\omega})$$

- special case: if $x[n]$ is real, the magnitude of the DTFT is symmetric:

$$x[n] \in \mathbb{R} \Longrightarrow |X(e^{j\omega})| = |X(e^{-j\omega})|$$

- more special case: if $x[n]$ is real and symmetric, $X(e^{j\omega})$ is also real and symmetric

Some particular cases:

- if $x[n]$ is symmetric, the DTFT is symmetric:

$$x[n] = x[-n] \iff X(e^{j\omega}) = X(e^{-j\omega})$$

- if $x[n]$ is real, the DTFT is Hermitian-symmetric:

$$x[n] = x^*[n] \iff X(e^{j\omega}) = X^*(e^{-j\omega})$$

- special case: if $x[n]$ is real, the magnitude of the DTFT is symmetric:

$$x[n] \in \mathbb{R} \implies |X(e^{j\omega})| = |X(e^{-j\omega})|$$

- more special case: if $x[n]$ is real and symmetric, $X(e^{j\omega})$ is also real and symmetric

Some particular cases:

- if $x[n]$ is symmetric, the DTFT is symmetric:

$$x[n] = x[-n] \iff X(e^{j\omega}) = X(e^{-j\omega})$$

- if $x[n]$ is real, the DTFT is Hermitian-symmetric:

$$x[n] = x^*[n] \iff X(e^{j\omega}) = X^*(e^{-j\omega})$$

- special case: if $x[n]$ is real, the magnitude of the DTFT is symmetric:

$$x[n] \in \mathbb{R} \Longrightarrow |X(e^{j\omega})| = |X(e^{-j\omega})|$$

- more special case: if $x[n]$ is real and symmetric, $X(e^{j\omega})$ is also real and symmetric

DTFT as basis expansion

Some things are OK:

- $\text{DFT}\{\delta[n]\} = 1$
- $\text{DTFT}\{\delta[n]\} = \langle e^{j\omega n}, \delta[n] \rangle = 1$

DTFT as basis expansion

Some things are OK:

- $\text{DFT}\{\delta[n]\} = 1$
- $\text{DTFT}\{\delta[n]\} = \langle e^{j\omega n}, \delta[n]\rangle = 1$

DTFT as basis expansion

Some things aren't:

- DFT $\{1\} = N\delta[k]$
- DTFT $\{1\} = \sum_{n=-\infty}^{\infty} e^{-j\omega n} = ?$
- problem: too many interesting sequences are *not* square summable!

DTFT as basis expansion

Some things aren't:

- $\text{DFT}\{1\} = N\delta[k]$
- $\text{DTFT}\{1\} = \sum_{n=-\infty}^{\infty} e^{-j\omega n} = ?$

- problem: too many interesting sequences are *not* square summable!

DTFT as basis expansion

Some things aren't:

- $\text{DFT}\{1\} = N\delta[k]$
- $\text{DTFT}\{1\} = \sum_{n=-\infty}^{\infty} e^{-j\omega n} = ?$

- problem: too many interesting sequences are *not* square summable!

The Dirac delta functional

Defined by the "sifting" property:

$$\int_{-\infty}^{\infty} \delta(t-s)f(t)dt = f(s)$$

for all functions of $t \in \mathbb{R}$.

Intuition

- family of *localizing* functions $r_k(t)$ with $k \in \mathbb{N}$ and $t \in \mathbb{R}$
- support inversely proportional to k
- constant area

Intuition

- family of *localizing* functions $r_k(t)$ with $k \in \mathbb{N}$ and $t \in \mathbb{R}$
- support inversely proportional to k
- constant area

Intuition

- family of *localizing* functions $r_k(t)$ with $k \in \mathbb{N}$ and $t \in \mathbb{R}$
- support inversely proportional to k
- constant area

Intuition

$$\text{rect}(t) = \begin{cases} 1 & \text{for } |t| < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Consider the localizing family $r_k(t) = k\,\text{rect}(kt)$:

- nonzero over $[-1/2k, 1/2k]$, i.e. support is $1/k$
- area is 1

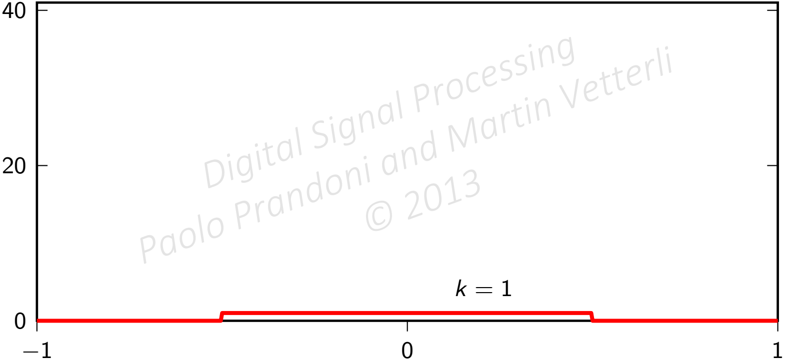
Intuition

$$\mathrm{rect}(t) = \begin{cases} 1 & \text{for } |t| < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

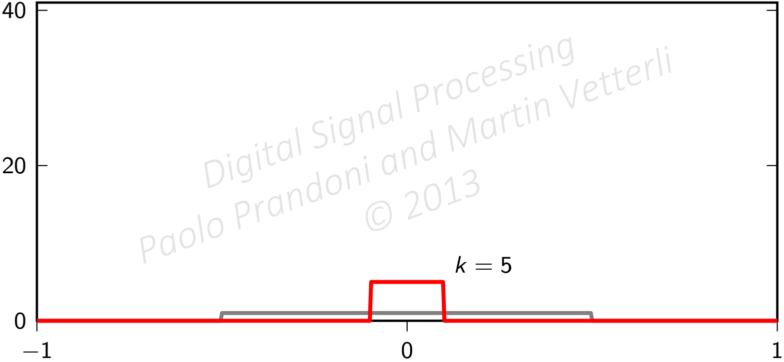
Consider the localizing family $r_k(t) = k\,\mathrm{rect}(kt)$:

- nonzero over $[-1/2k, 1/2k]$, i.e. support is $1/k$
- area is 1

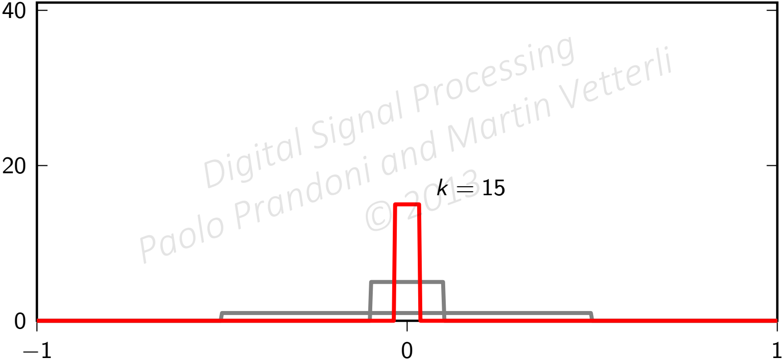
The family $r_k(t) = k\,\mathrm{rect}(kt)$



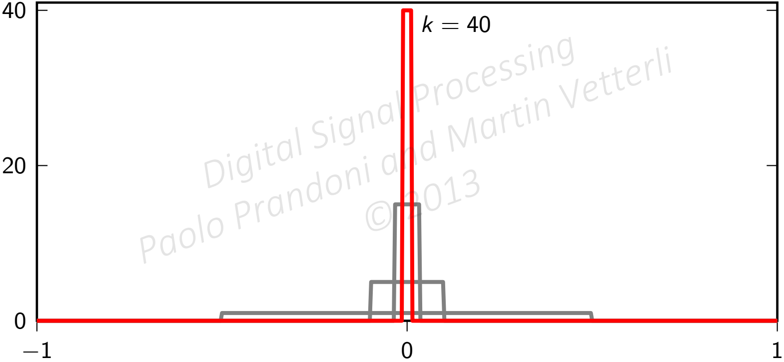
The family $r_k(t) = k\,\mathrm{rect}(kt)$



The family $r_k(t) = k\,\mathrm{rect}(kt)$



The family $r_k(t) = k\,\mathrm{rect}(kt)$



Extracting a point value

By the Mean Value theorem:

$$\int_{-\infty}^{\infty} r_k(t)f(t)dt = k\int_{-1/2k}^{1/2k} f(t)dt$$
$$= f(\gamma)\big|_{\gamma\in[-1/2k,1/2k]}$$

and so:

$$\lim_{k\to\infty}\int_{-\infty}^{\infty} r_k(t)f(t)dt = f(0)$$

The delta functional is a shorthand. Instead of writing

$$\lim_{k\to\infty} \int_{-\infty}^{\infty} r_k(t-s)f(t)dt$$

we write

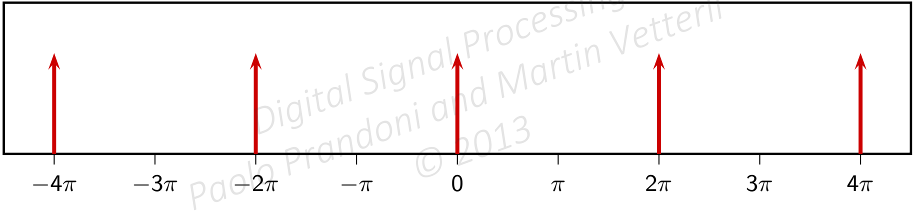
$$\int_{-\infty}^{\infty} \delta(t-s)f(t)dt.$$

as if $\lim_{k\to\infty} r_k(t) = \delta(t)$,

The “pulse train”

$$\tilde{\delta}(\omega) = 2\pi \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$$

just a technicality to use the Dirac delta in the space of 2π-periodic functions


-4π
-3π
-2π
$-\pi$
0
π
2π
3π
4π
Digital Signal Processing
Paolo Prandoni and Martin Vetterli
© 2013

Now let the show begin!

$$\text{IDTFT}\left\{\tilde{\delta}(\omega)\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\tilde{\delta}(\omega)e^{j\omega n}d\omega$$

$$= \int_{-\pi}^{\pi}\delta(\omega)e^{j\omega n}d\omega$$

$$= e^{j\omega n}\big|_{\omega=0}$$

$$= 1$$

Now let the show begin!

$$
\begin{aligned}
\text{IDTFT}\left\{\tilde{\delta}(\omega)\right\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\tilde{\delta}(\omega)e^{j\omega n}d\omega \\
&= \int_{-\pi}^{\pi}\delta(\omega)e^{j\omega n}d\omega \\
&= e^{j\omega n}\big|_{\omega=0} \\
&= 1
\end{aligned}
$$

Now let the show begin!

$$\text{IDTFT}\left\{\tilde{\delta}(\omega)\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\tilde{\delta}(\omega)e^{j\omega n}d\omega$$

$$= \int_{-\pi}^{\pi}\delta(\omega)e^{j\omega n}d\omega$$

$$= e^{j\omega n}\big|_{\omega=0}$$

$$= 1$$

Now let the show begin!

$$\begin{aligned}
\text{IDTFT}\left\{\tilde{\delta}(\omega)\right\} &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\tilde{\delta}(\omega)e^{j\omega n}d\omega \\
&= \int_{-\pi}^{\pi}\delta(\omega)e^{j\omega n}d\omega \\
&= e^{j\omega n}\big|_{\omega=0} \\
&= 1
\end{aligned}$$

In other words

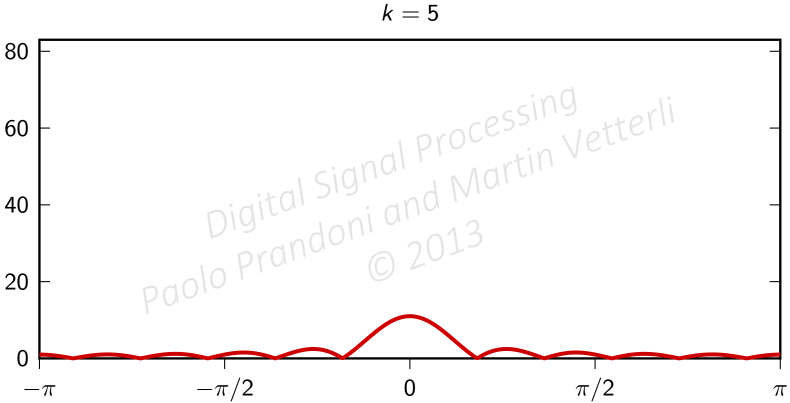
$$\mathrm{DTFT}\left\{1\right\} = \tilde{\delta}(\omega)$$

Does it make sense?

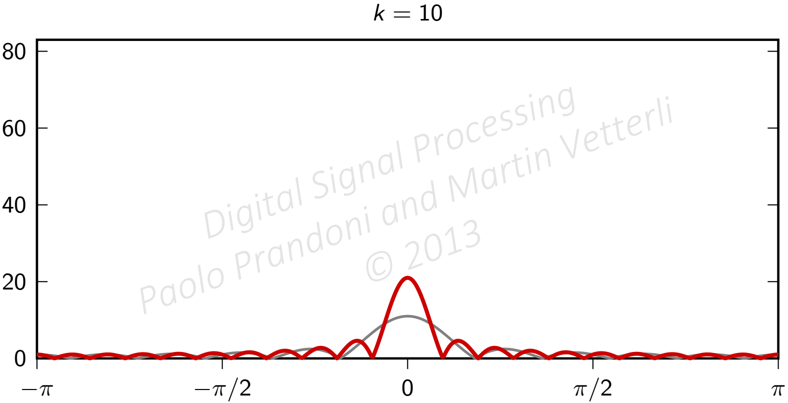
Partial DTFT sum:

$$S_k(\omega) = \sum_{n=-k}^{k} e^{-j\omega n}$$

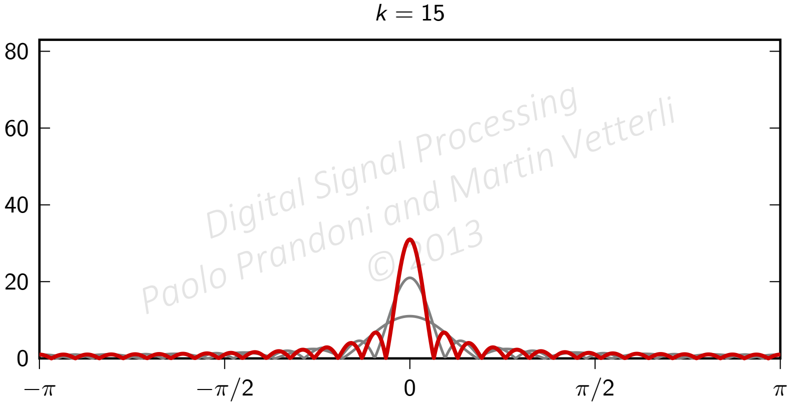
Plotting $|S_k(\omega)|$



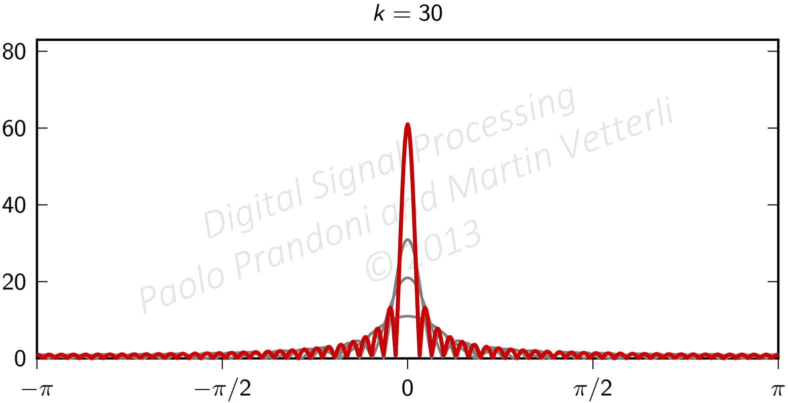
Plotting $|S_k(\omega)|$



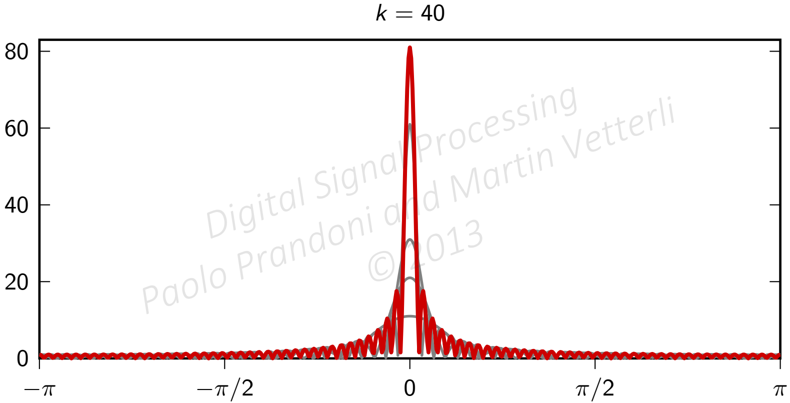
Plotting $|S_k(\omega)|$



Plotting $|S_k(\omega)|$



Plotting $|S_k(\omega)|$



Does it make sense?

Partial DTFT sums look like a family of localizing functions:

$$S_k(\omega) \rightarrow \tilde{\delta}(\omega)$$

$$\text{IDTFT}\left\{\tilde{\delta}(\omega-\omega_0)\right\} = e^{j\omega_0 n}$$

So:

- $\text{DTFT}\{1\} = \tilde{\delta}(\omega)$
- $\text{DTFT}\{e^{j\omega_0 n}\} = \tilde{\delta}(\omega-\omega_0)$
- $\text{DTFT}\{\cos\omega_0 n\} = [\tilde{\delta}(\omega-\omega_0)+\tilde{\delta}(\omega+\omega_0)]/2$
- $\text{DTFT}\{\sin\omega_0 n\} = -j[\tilde{\delta}(\omega-\omega_0)-\tilde{\delta}(\omega+\omega_0)]/2$

Using the same technique

$$\text{IDTFT}\left\{\tilde{\delta}(\omega - \omega_0)\right\} = e^{j\omega_0 n}$$

So:

- $\text{DTFT}\{1\} = \tilde{\delta}(\omega)$
- $\text{DTFT}\left\{e^{j\omega_0 n}\right\} = \tilde{\delta}(\omega - \omega_0)$
- $\text{DTFT}\{\cos \omega_0 n\} = [\tilde{\delta}(\omega - \omega_0) + \tilde{\delta}(\omega + \omega_0)]/2$
- $\text{DTFT}\{\sin \omega_0 n\} = -j[\tilde{\delta}(\omega - \omega_0) - \tilde{\delta}(\omega + \omega_0)]/2$

Using the same technique

$$\text{IDTFT}\left\{\tilde{\delta}(\omega - \omega_0)\right\} = e^{j\omega_0 n}$$

So:

- $\text{DTFT}\{1\} = \tilde{\delta}(\omega)$
- $\text{DTFT}\left\{e^{j\omega_0 n}\right\} = \tilde{\delta}(\omega - \omega_0)$
- $\text{DTFT}\{\cos\omega_0 n\} = [\tilde{\delta}(\omega - \omega_0) + \tilde{\delta}(\omega + \omega_0)]/2$
- $\text{DTFT}\{\sin\omega_0 n\} = -j[\tilde{\delta}(\omega - \omega_0) - \tilde{\delta}(\omega + \omega_0)]/2$

Using the same technique

$$\text{IDTFT}\left\{\tilde{\delta}(\omega - \omega_0)\right\} = e^{j\omega_0 n}$$

So:

- $\text{DTFT}\{1\} = \tilde{\delta}(\omega)$
- $\text{DTFT}\left\{e^{j\omega_0 n}\right\} = \tilde{\delta}(\omega - \omega_0)$
- $\text{DTFT}\{\cos\omega_0 n\} = [\tilde{\delta}(\omega - \omega_0) + \tilde{\delta}(\omega + \omega_0)]/2$
- $\text{DTFT}\{\sin\omega_0 n\} = -j[\tilde{\delta}(\omega - \omega_0) - \tilde{\delta}(\omega + \omega_0)]/2$

END OF MODULE 4.5

Digital Signal Processing

Module 4.6: Relationships between transforms

Overview:

- DFT, DFS, DTFT
- DTFT of periodic sequences
- DTFT of finite-support sequences
- Zero padding

- DFT, DFS: change of basis in $\mathbb{C}^N$
- DTFT: "formal" change of basis in $\ell_2(\mathbb{Z})$
- basis vectors are "building blocks" for any signal

- DFT: numerical algorithm (computable)
- DTFT: mathematical tool (proofs)

Transforms

- DFT, DFS: change of basis in $\mathbb{C}^N$
- DTFT: "formal" change of basis in $\ell_2(\mathbb{Z})$
- basis vectors are "building blocks" for any signal
- DFT: numerical algorithm (computable)
- DTFT: mathematical tool (proofs)

Transforms

- DFT, DFS: change of basis in $\mathbb{C}^N$
- DTFT: "formal" change of basis in $\ell_2(\mathbb{Z})$
- basis vectors are "building blocks" for any signal
- DFT: numerical algorithm (computable)
- DTFT: mathematical tool (proofs)

Transforms

- DFT, DFS: change of basis in $\mathbb{C}^N$
- DTFT: "formal" change of basis in $\ell_2(\mathbb{Z})$
- basis vectors are "building blocks" for any signal

- DFT: numerical algorithm (computable)
- DTFT: mathematical tool (proofs)

- DFT, DFS: change of basis in $\mathbb{C}^N$
- DTFT: "formal" change of basis in $\ell_2(\mathbb{Z})$
- basis vectors are "building blocks" for any signal

- DFT: numerical algorithm (computable)
- DTFT: mathematical tool (proofs)

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \le n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

Embedding finite-length signals

- N-tap signal $x[n]$
- natural spectral representation: DFT $X[k]$
- two ways to embed $x[n]$ into an infinite sequence:
  - periodic extension: $\tilde{x}[n] = x[n \mod N]$
  - finite-support extension: $\bar{x}[n] = \begin{cases} x[n] & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$
- how does $X[k]$ relate to the DTFT of the embedded signals?

DTFT of periodic signals

$$\tilde{x}[n] = x[n \mod N]$$

$$\tilde{X}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \tilde{x}[n] e^{-j\omega n}$$

$$= \sum_{n=-\infty}^{\infty} \left(\frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] \left(\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi}{N}nk} e^{-j\omega n} \right)$$

DTFT of periodic signals

$$\tilde{x}[n] = x[n \mod N]$$

$$\tilde{X}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \tilde{x}[n] e^{-j\omega n}$$

$$= \sum_{n=-\infty}^{\infty} \left(\frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] \left(\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi}{N}nk} e^{-j\omega n} \right)$$

DTFT of periodic signals

$$\tilde{x}[n] = x[n \mod N]$$

$$\begin{aligned}
\tilde{X}(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \tilde{x}[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \left(\frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] \left(\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi}{N}nk} e^{-j\omega n} \right)
\end{aligned}$$

We've seen this before

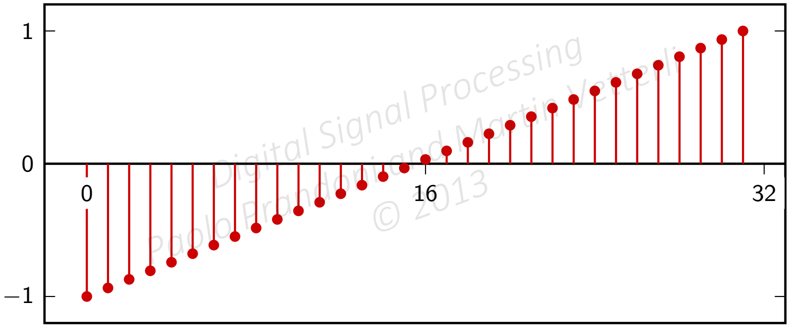
$$\sum_{n=-\infty}^{\infty} e^{j\frac{2\pi}{N}nk} e^{-j\omega n} = \text{DTFT}\left\{e^{j\frac{2\pi}{N}nk}\right\}$$

$$= \tilde{\delta}(\omega - \frac{2\pi}{N}k)$$

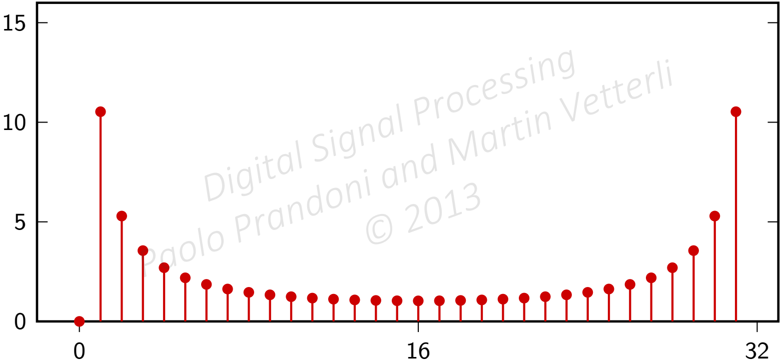
DTFT of periodic signals

$$\tilde{X}(e^{j\omega}) = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\tilde{\delta}(\omega - \frac{2\pi}{N}k)$$

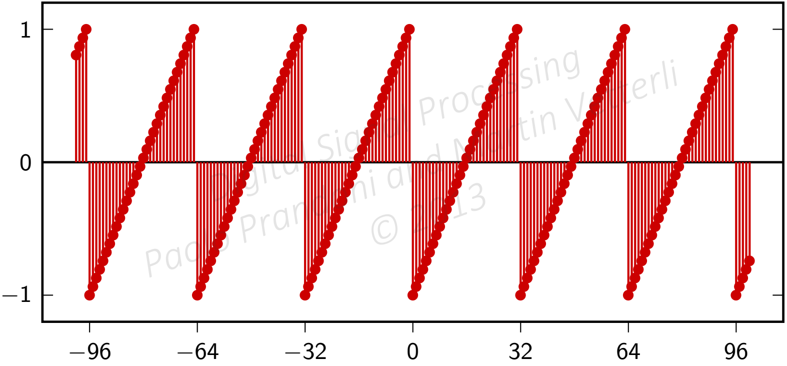
32-tap sawtooth



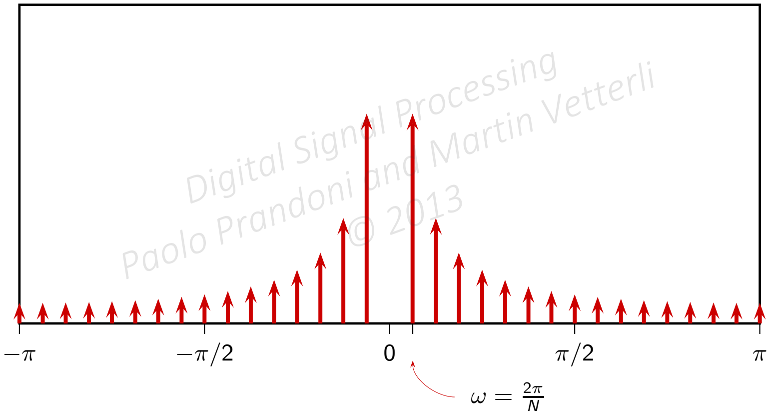
DFT of 32-tap sawtooth



32-periodic sawtooth



DTFT of periodic extension



DTFT of finite-support signals

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

$$\bar{X}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \bar{x}[n] e^{-j\omega n} = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

$$= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left(\sum_{n=0}^{N-1} e^{-j(\omega - \frac{2\pi}{N}k)n} \right)$$

DTFT of finite-support signals

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

$$\bar{X}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \bar{x}[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$$

$$= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left(\sum_{n=0}^{N-1} e^{-j(\omega - \frac{2\pi}{N}k)n} \right)$$

DTFT of finite-support signals

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

$$\begin{aligned} \bar{X}(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \bar{x}[n] e^{-j\omega n} = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \\ &= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\omega n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left(\sum_{n=0}^{N-1} e^{-j(\omega - \frac{2\pi}{N}k)n} \right) \end{aligned}$$

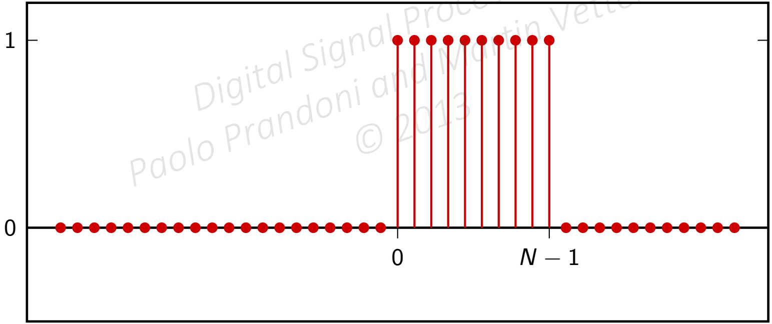
DTFT of finite-support signals

$$\sum_{n=0}^{N-1} e^{-j(\omega - \frac{2\pi}{N}k)n} = \bar{R}(e^{j(\omega - \frac{2\pi}{N}k)})$$

where $\bar{R}(e^{j\omega})$ is the DTFT of $\bar{r}[n]$, the interval indicator signal:

$$\bar{r}[n] = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{r}[n] = \begin{cases} 1 & 0 \leq n < N \\ 0 & \text{otherwise} \end{cases}$$



DTFT of interval signal

$$\bar{R}(e^{j\omega}) = \sum_{n=0}^{N-1} e^{-j\omega n}$$

$$= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$$

$$= \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]}$$

$$= \frac{\sin\left(\frac{\omega}{2}N\right)}{\sin\left(\frac{\omega}{2}\right)} e^{-j\frac{\omega}{2}(N-1)}$$

DTFT of interval signal

$$\bar{R}(e^{j\omega}) = \sum_{n=0}^{N-1} e^{-j\omega n}$$

$$= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$$

$$= \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]}$$

$$= \frac{\sin\left(\frac{\omega}{2}N\right)}{\sin\left(\frac{\omega}{2}\right)} e^{-j\frac{\omega}{2}(N-1)}$$

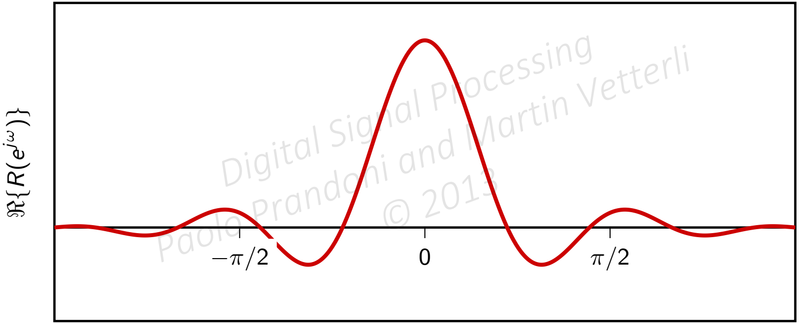
DTFT of interval signal

$$
\begin{aligned}
\bar{R}(e^{j\omega}) &= \sum_{n=0}^{N-1} e^{-j\omega n} \\
&= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \\
&= \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]} \\
&= \frac{\sin\left(\frac{\omega}{2}N\right)}{\sin\left(\frac{\omega}{2}\right)} e^{-j\frac{\omega}{2}(N-1)}
\end{aligned}
$$

DTFT of interval signal

$$
\begin{aligned}
\bar{R}(e^{j\omega}) &= \sum_{n=0}^{N-1} e^{-j\omega n} \\
&= \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \\
&= \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]} \\
&= \frac{\sin\left(\frac{\omega}{2}N\right)}{\sin\left(\frac{\omega}{2}\right)} e^{-j\frac{\omega}{2}(N-1)}
\end{aligned}
$$

DTFT of interval signal ($N = 9$)

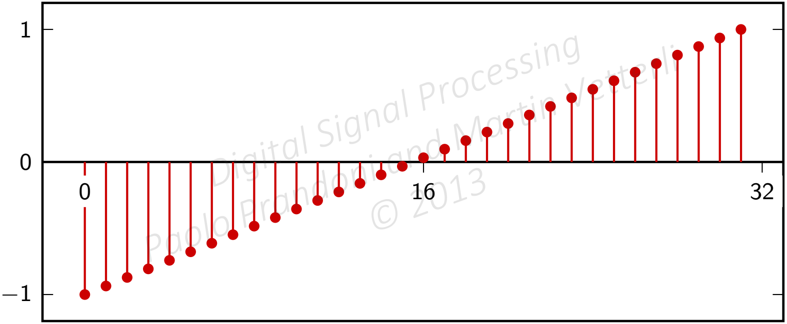


DTFT of finite-support signals

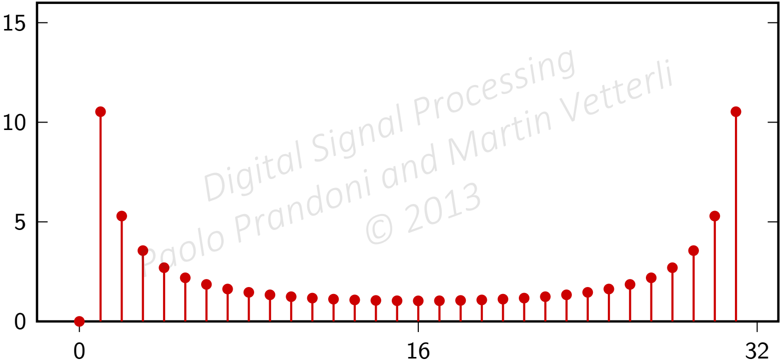
$$\bar{X}(e^{j\omega}) = \sum_{k=0}^{N-1} X[k]\Lambda(\omega - \frac{2\pi}{N}k)$$

with $\Lambda(\omega) = (1/N)\bar{R}(e^{j\omega})$: smooth interpolation of DFT values.

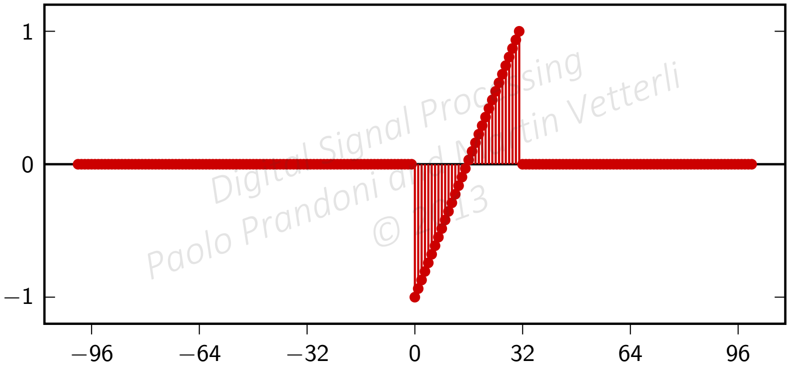
32-tap sawtooth



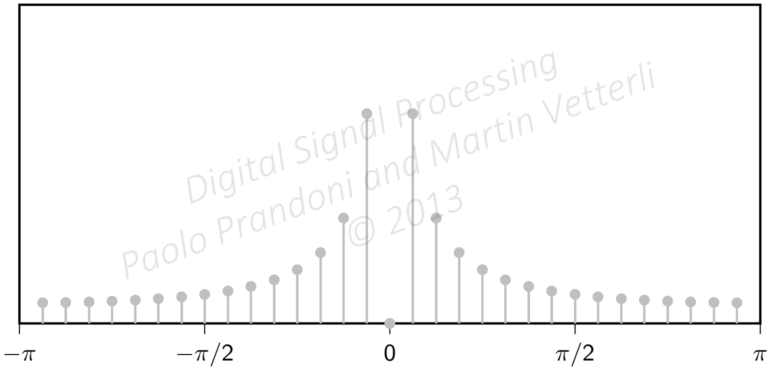
DFT of 32-tap sawtooth



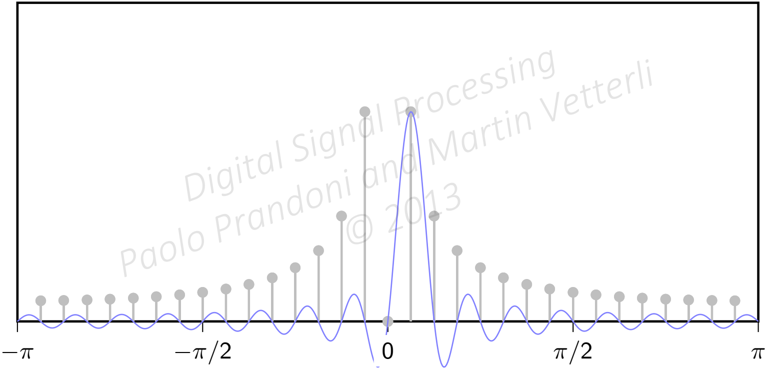
Sawtooth: finite support extension



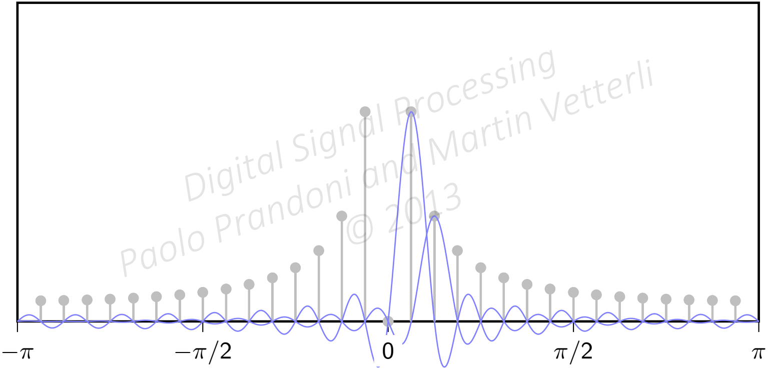
DTFT of finite support extension (sketch)



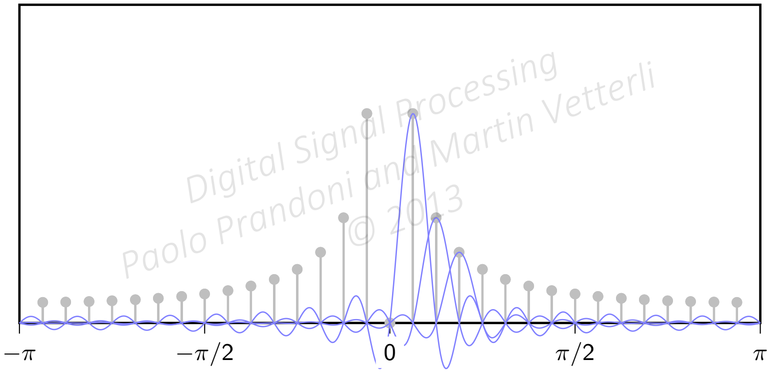
DTFT of finite support extension (sketch)



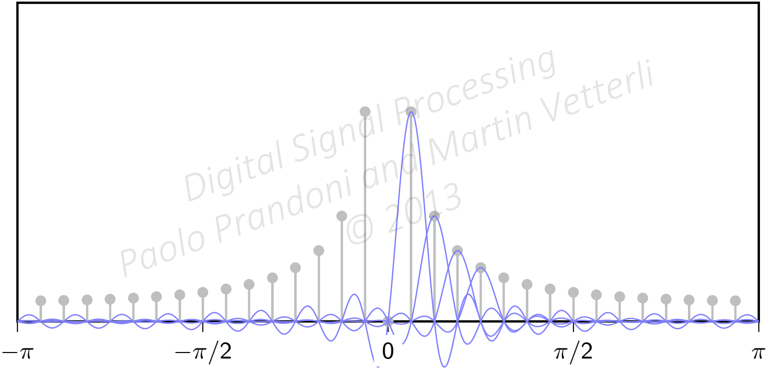
DTFT of finite support extension (sketch)



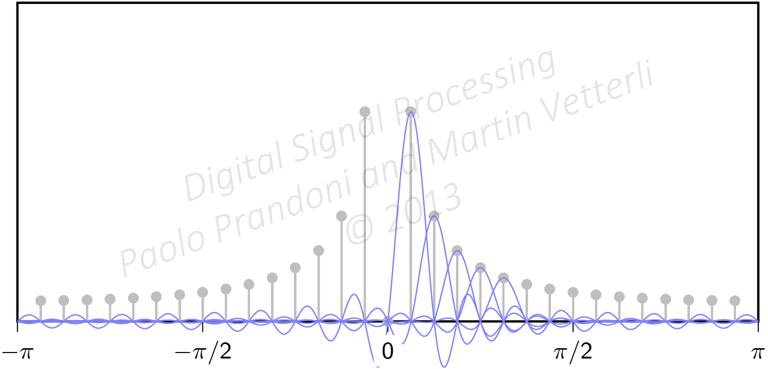
DTFT of finite support extension (sketch)



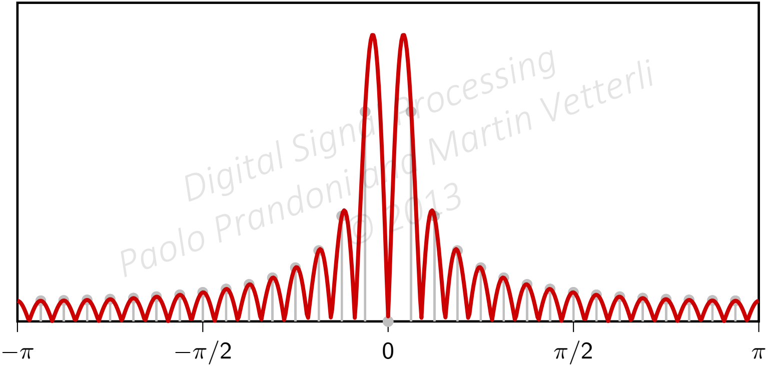
DTFT of finite support extension (sketch)



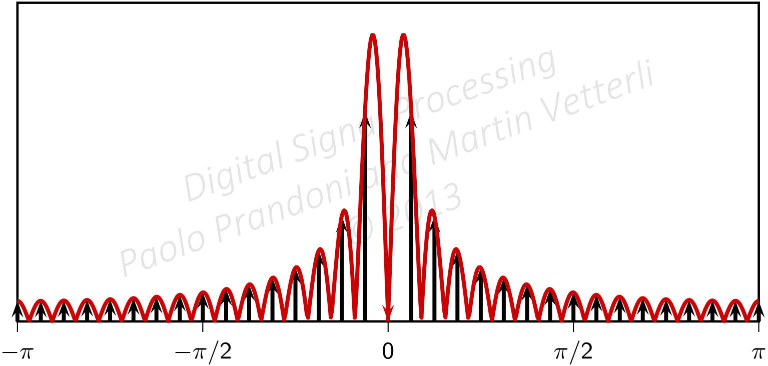
DTFT of finite support extension (sketch)



DTFT of finite support extension



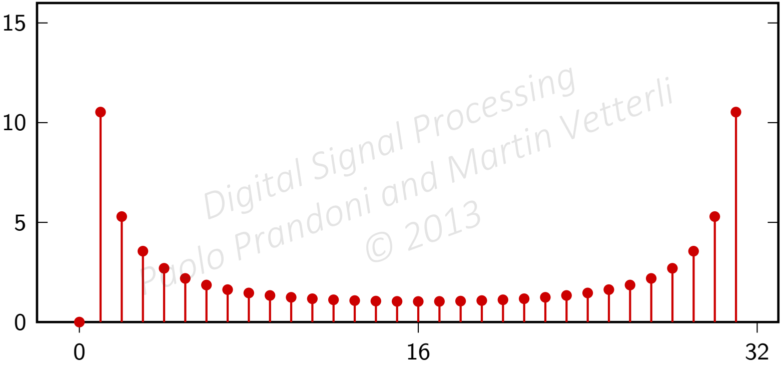
As a comparison...



About zero-padding

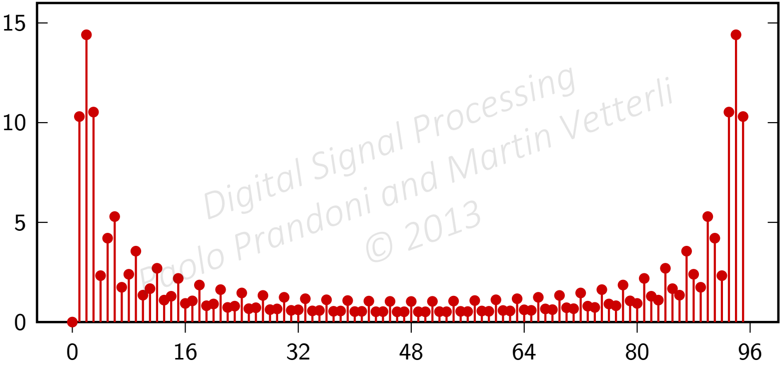
When computing the DFT numerically
one may “pad” the data vector with zeros to obtain “nicer” plots

DFT of 32-tap sawtooth



$$\mathbf{x} = [x_0\ x_1\ \ldots\ x_{31}]$$

DFT of 32-tap sawtooth, zero-padded



$$\mathbf{x} = [x_0\ x_1\ \ldots\ x_{31}\ 0\ \ldots\ 0]$$

$$X_M[h] = \sum_{n=0}^{M-1} x'[n]e^{-j\frac{2\pi}{M}nh} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{M}nh}$$

$$= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X_N[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\frac{2\pi}{M}nh}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X_N[k] \left(\sum_{n=0}^{N-1} e^{-j(\frac{2\pi}{M}h - \frac{2\pi}{N}k)n} \right)$$

$$= \bar{X}(e^{j\omega})\big|_{\omega=\frac{2\pi}{M}h}$$

About zero-padding

$$X_M[h] = \sum_{n=0}^{M-1} x'[n] e^{-j\frac{2\pi}{M}nh} = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{M}nh}$$

$$= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X_N[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\frac{2\pi}{M}nh}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X_N[k] \left(\sum_{n=0}^{N-1} e^{-j(\frac{2\pi}{M}h - \frac{2\pi}{N}k)n} \right)$$

$$= \bar{X}(e^{j\omega})\big|_{\omega=\frac{2\pi}{M}h}$$

About zero-padding

$$
\begin{aligned}
X_M[h] &= \sum_{n=0}^{M-1} x'[n] e^{-j\frac{2\pi}{M}nh} = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{M}nh} \\
&= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X_N[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\frac{2\pi}{M}nh} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} X_N[k] \left(\sum_{n=0}^{N-1} e^{-j(\frac{2\pi}{M}h - \frac{2\pi}{N}k)n} \right) \\
&= \bar{X}(e^{j\omega})\big|_{\omega=\frac{2\pi}{M}h}
\end{aligned}
$$

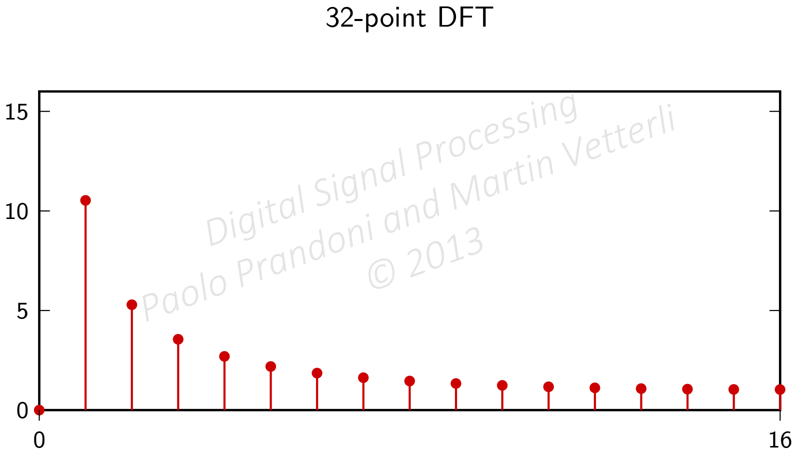
About zero-padding

$$
\begin{aligned}
X_M[h] &= \sum_{n=0}^{M-1} x'[n]e^{-j\frac{2\pi}{M}nh} = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{M}nh} \\
&= \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X_N[k] e^{j\frac{2\pi}{N}nk} \right) e^{-j\frac{2\pi}{M}nh} \\
&= \frac{1}{N} \sum_{k=0}^{N-1} X_N[k] \left(\sum_{n=0}^{N-1} e^{-j(\frac{2\pi}{M}h - \frac{2\pi}{N}k)n} \right) \\
&= \bar{X}(e^{j\omega})\big|_{\omega=\frac{2\pi}{M}h}
\end{aligned}
$$

About zero-padding

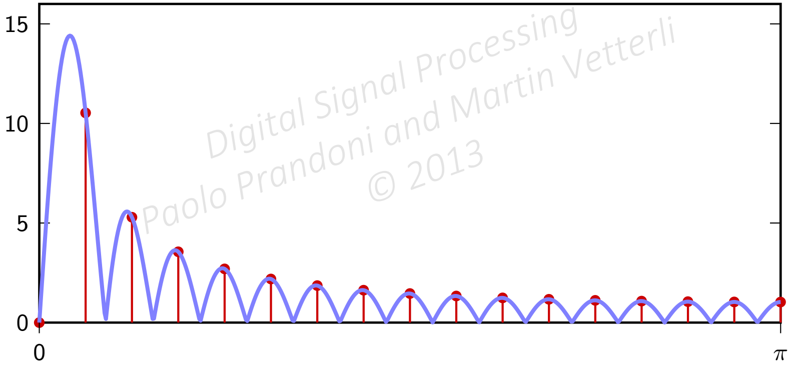
- zero padding does not add information
- a zero-padded DFT is simply a sampled DTFT of the finite-support extension

DFT of 32-tap sawtooth, zero-padded



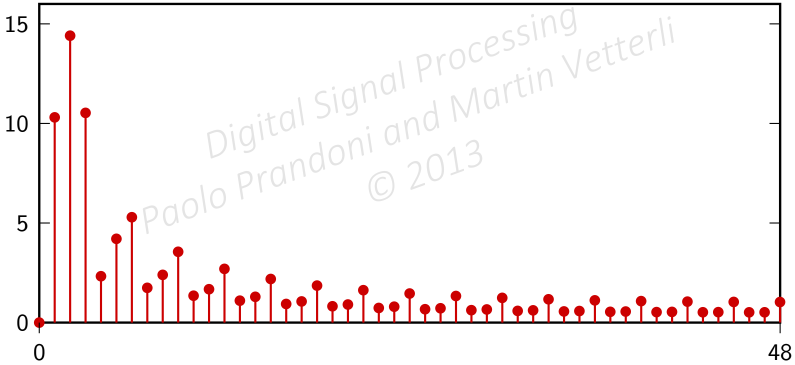
DFT of 32-tap sawtooth, zero-padded

32-point DFT

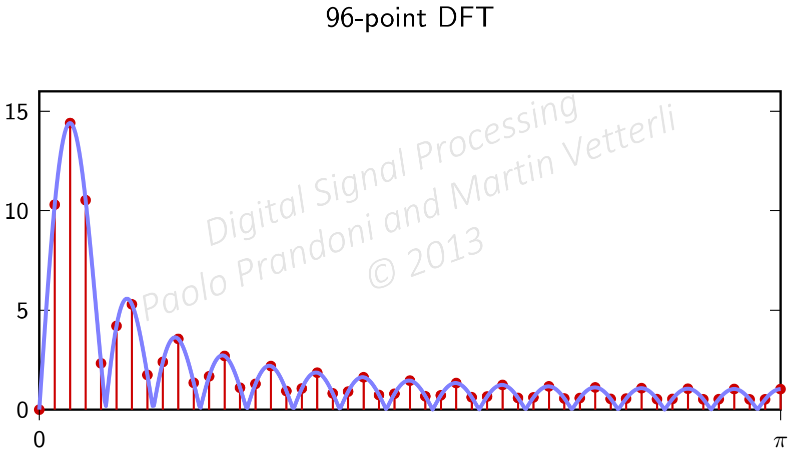


DFT of 32-tap sawtooth, zero-padded

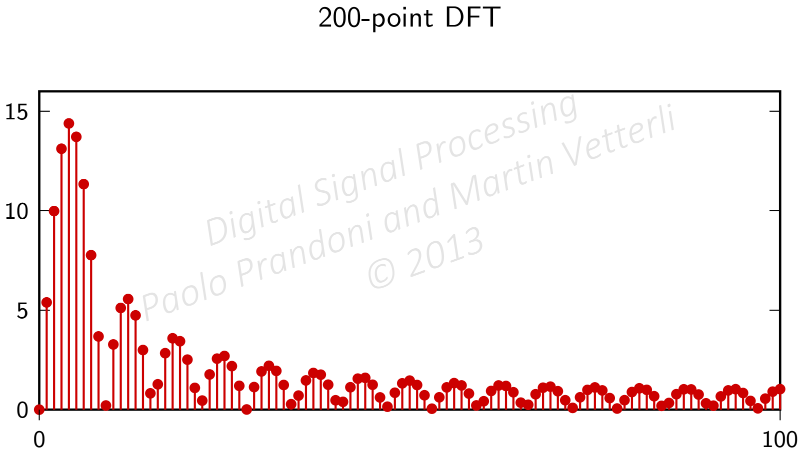
96-point DFT



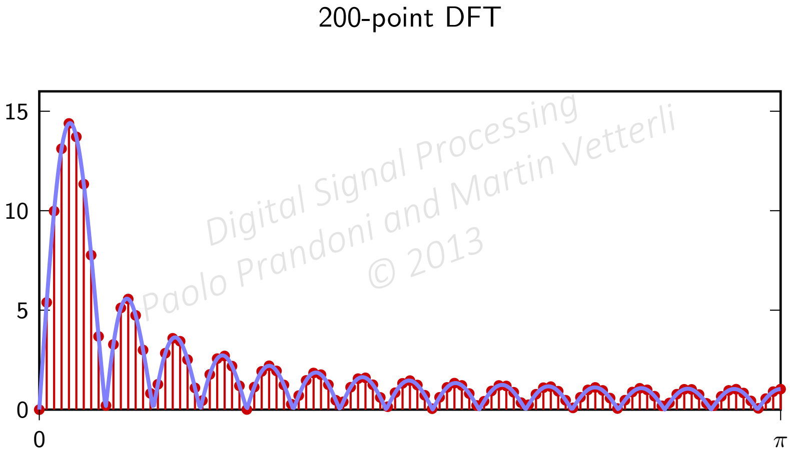
DFT of 32-tap sawtooth, zero-padded



DFT of 32-tap sawtooth, zero-padded



DFT of 32-tap sawtooth, zero-padded



END OF MODULE 4.6

Digital Signal Processing

Module 4.7: Sinusoidal Modulation

Overview:

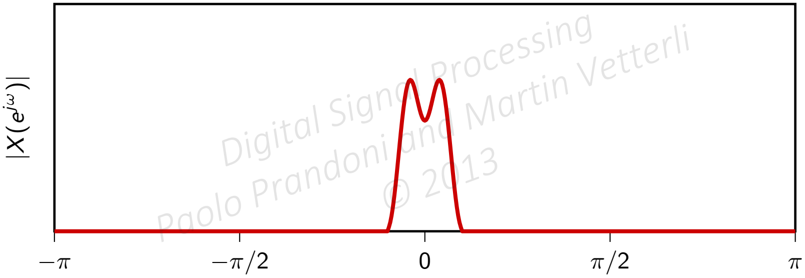
- Lowpass, highpass and bandpass signals
- Sinusoidal modulation
- Tuning a guitar

Classifying signals in frequency

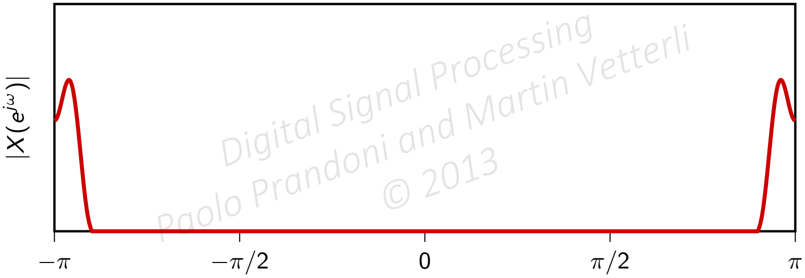
Three broad categories according to where most of the spectral energy resides:

- lowpass signals (also known as "baseband" signals)
- highpass signals
- bandpass signals

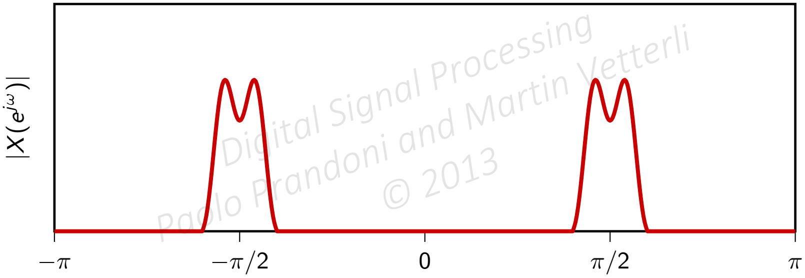
Lowpass example



Highpass example



Bandpass example



$$\text{DTFT}\{x[n]\cos(\omega_c n)\} = \text{DTFT}\left\{\frac{1}{2}e^{j\omega_c n}x[n] + \frac{1}{2}e^{-j\omega_c n}x[n]\right\}$$

$$= \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

- usually $x[n]$ baseband
- ω_c is the *carrier* frequency

Sinusoidal modulation

$$\text{DTFT}\left\{x[n]\cos(\omega_c n)\right\} = \text{DTFT}\left\{\frac{1}{2}e^{j\omega_c n}x[n] + \frac{1}{2}e^{-j\omega_c n}x[n]\right\}$$

$$= \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

- usually $x[n]$ baseband
- ω_c is the *carrier* frequency

Sinusoidal modulation

$$\text{DTFT}\{x[n]\cos(\omega_c n)\} = \text{DTFT}\left\{\frac{1}{2}e^{j\omega_c n}x[n] + \frac{1}{2}e^{-j\omega_c n}x[n]\right\}$$

$$= \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

- usually $x[n]$ baseband
- ω_c is the *carrier* frequency

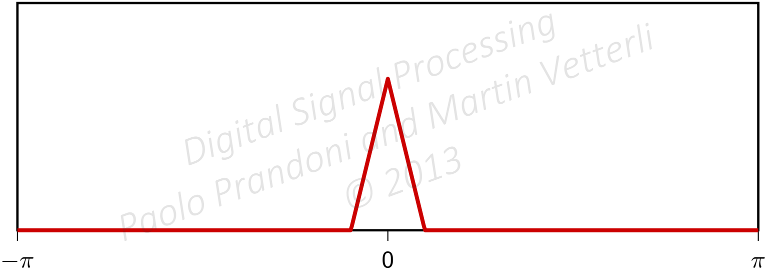
Sinusoidal modulation

$$\text{DTFT}\left\{x[n]\cos(\omega_c n)\right\} = \text{DTFT}\left\{\frac{1}{2}e^{j\omega_c n}x[n] + \frac{1}{2}e^{-j\omega_c n}x[n]\right\}$$

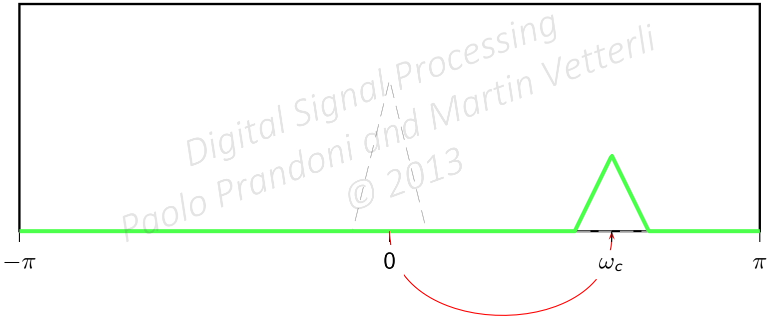
$$= \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

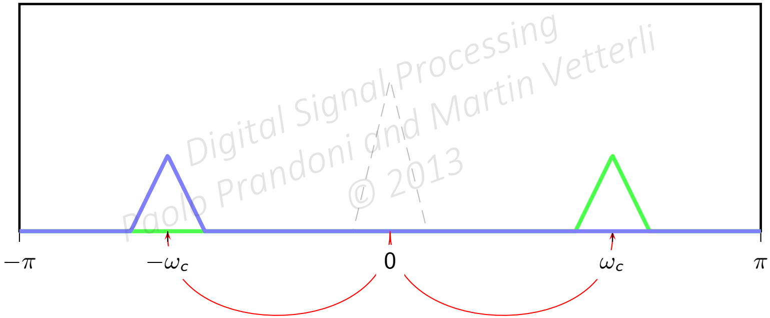
- usually $x[n]$ baseband
- ω_c is the *carrier* frequency

Example



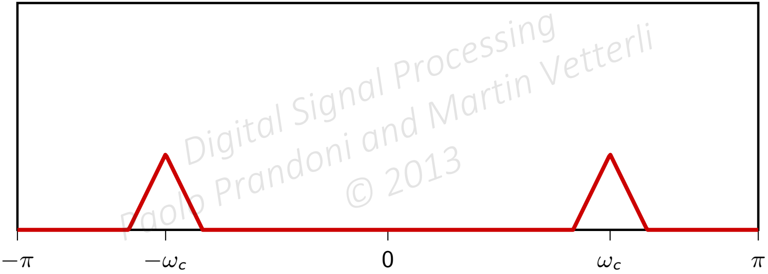
Example



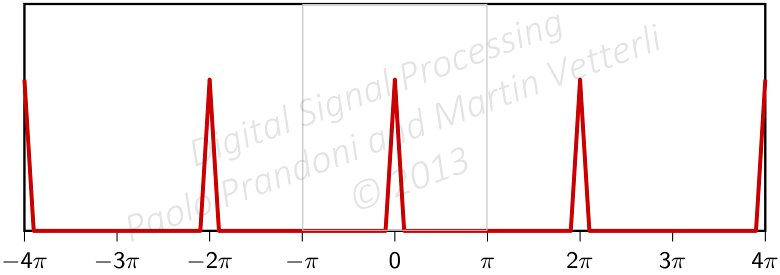


$-\pi$
$-\omega_c$
0
ω_c
π

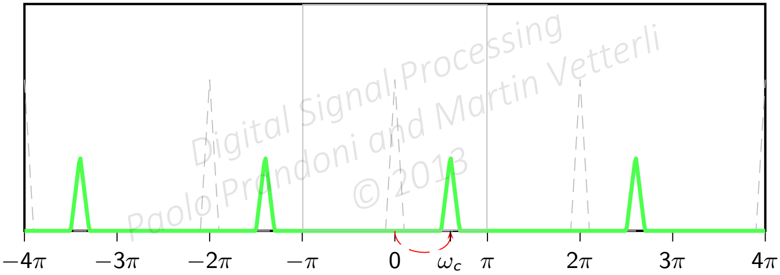
Example



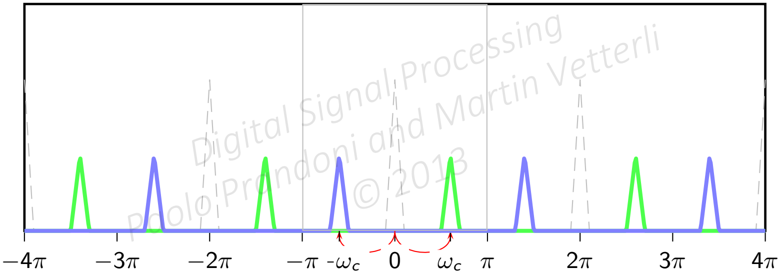
Again, explicitly showing the periodicity of the spectrum



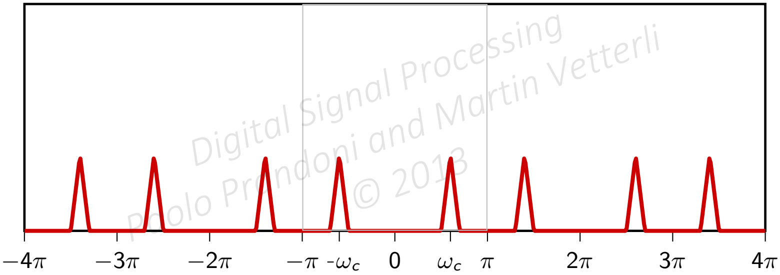
Again, explicitly showing the periodicity of the spectrum



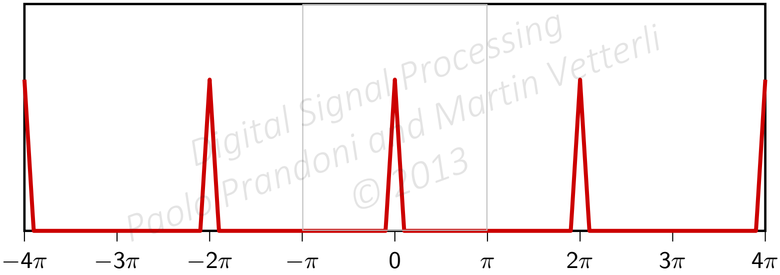
Again, explicitly showing the periodicity of the spectrum



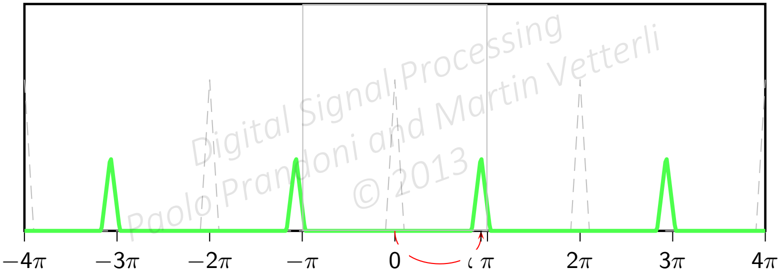
Again, explicitly showing the periodicity of the spectrum



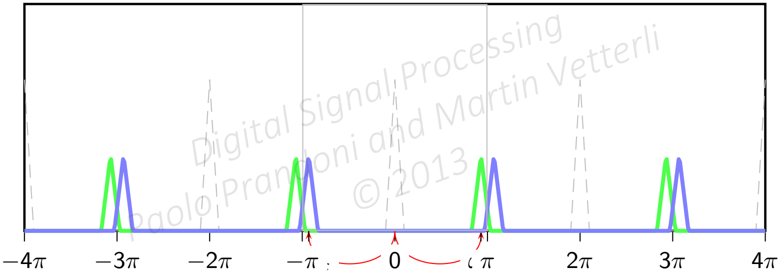
Careful when the modulation frequency is too large!



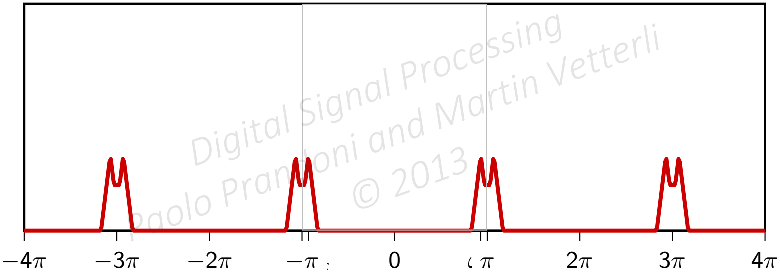
Careful when the modulation frequency is too large!



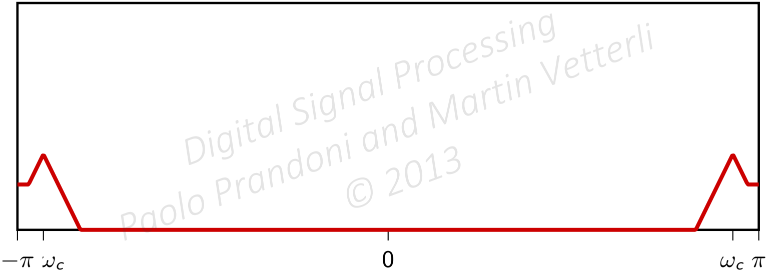
Careful when the modulation frequency is too large!



Careful when the modulation frequency is too large!



Careful when the modulation frequency is too large!



Sinusoidal modulation: applications

- voice and music are lowpass signals
- radio channels are bandpass, in much higher frequencies
- modulation brings the baseband signal in the transmission band
- demodulation at the receiver brings it back

just multiply the received signal by the carrier again

$$y[n] = x[n]\cos(\omega_c n) \qquad Y(e^{j\omega}) = \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

$$\begin{aligned}
\text{DTFT}\left\{y[n]\cdot 2\cos(\omega_c n)\right\} &= Y(e^{j(\omega-\omega_c)}) + Y(e^{j(\omega+\omega_c)}) \\
&= \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega)}) + X(e^{j(\omega)}) + X(e^{j(\omega+2\omega_c)})\right] \\
&= X(e^{j(\omega)}) + \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega+2\omega_c)})\right]
\end{aligned}$$

Sinusoidal demodulation

just multiply the received signal by the carrier again

$$y[n] = x[n]\cos(\omega_c n) \qquad Y(e^{j\omega}) = \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

$$\begin{aligned}
\mathrm{DTFT}\left\{y[n]\cdot 2\cos(\omega_c n)\right\} &= Y(e^{j(\omega-\omega_c)}) + Y(e^{j(\omega+\omega_c)}) \\
&= \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega)}) + X(e^{j(\omega)}) + X(e^{j(\omega+2\omega_c)})\right] \\
&= X(e^{j(\omega)}) + \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega+2\omega_c)})\right]
\end{aligned}$$

Sinusoidal demodulation

just multiply the received signal by the carrier again

$$y[n] = x[n]\cos(\omega_c n) \qquad Y(e^{j\omega}) = \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

$$\text{DTFT}\{y[n]\cdot 2\cos(\omega_c n)\} = Y(e^{j(\omega-\omega_c)}) + Y(e^{j(\omega+\omega_c)})$$

$$= \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega)}) + X(e^{j(\omega)}) + X(e^{j(\omega+2\omega_c)})\right]$$

$$= X(e^{j(\omega)}) + \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega+2\omega_c)})\right]$$

Sinusoidal demodulation

just multiply the received signal by the carrier again

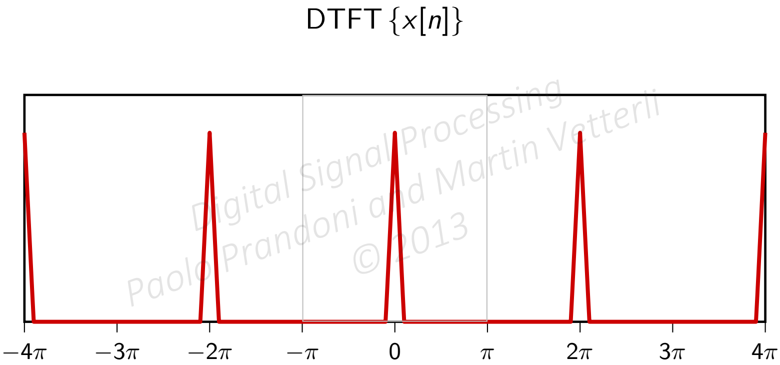
$$y[n] = x[n]\cos(\omega_c n) \qquad Y(e^{j\omega}) = \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

$$\begin{aligned}\mathrm{DTFT}\left\{y[n]\cdot 2\cos(\omega_c n)\right\} &= Y(e^{j(\omega-\omega_c)}) + Y(e^{j(\omega+\omega_c)}) \\ &= \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega)}) + X(e^{j(\omega)}) + X(e^{j(\omega+2\omega_c)})\right] \\ &= X(e^{j(\omega)}) + \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega+2\omega_c)})\right]\end{aligned}$$

just multiply the received signal by the carrier again

$$y[n] = x[n]\cos(\omega_c n) \qquad Y(e^{j\omega}) = \frac{1}{2}\left[X(e^{j(\omega-\omega_c)}) + X(e^{j(\omega+\omega_c)})\right]$$

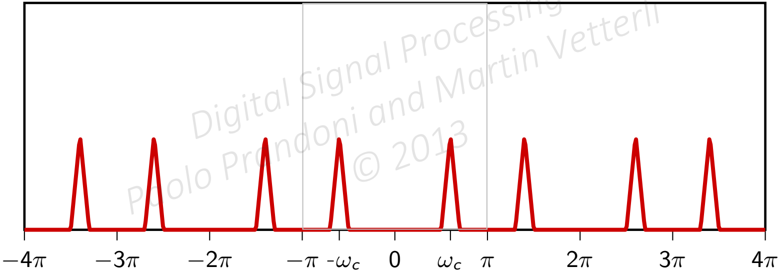
$$\begin{aligned}
\mathrm{DTFT}\left\{y[n]\cdot 2\cos(\omega_c n)\right\} &= Y(e^{j(\omega-\omega_c)}) + Y(e^{j(\omega+\omega_c)}) \\
&= \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega)}) + X(e^{j(\omega)}) + X(e^{j(\omega+2\omega_c)})\right] \\
&= X(e^{j(\omega)}) + \frac{1}{2}\left[X(e^{j(\omega-2\omega_c)}) + X(e^{j(\omega+2\omega_c)})\right]
\end{aligned}$$



DTFT $\{x[n]\}$
-4π
-3π
-2π
$-\pi$
0
π
2π
3π
4π

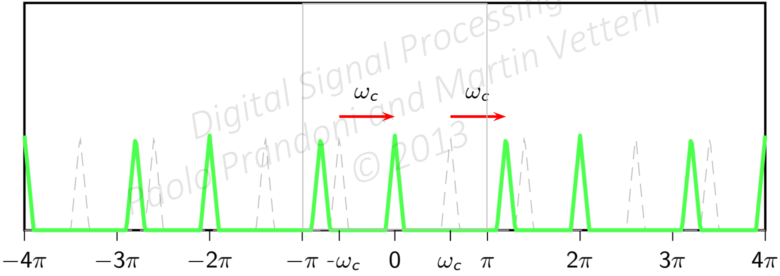
Demodulation in the frequency domain

$$\text{DTFT}\{y[n]\} = \text{DTFT}\{x[n]\cos\omega_c n\}$$



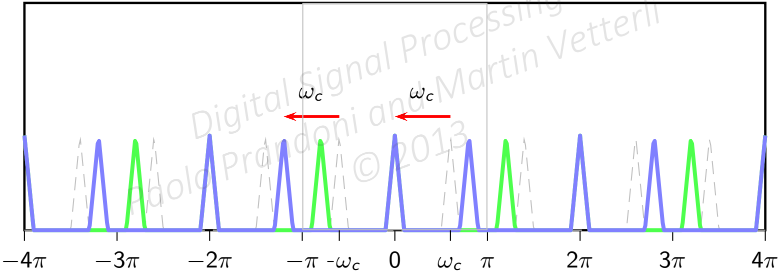
Demodulation in the frequency domain

$$\text{DTFT}\left\{y[n]\, 2\cos\omega_c n\right\}$$



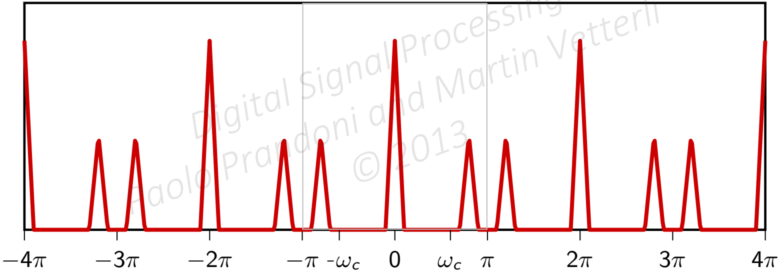
Demodulation in the frequency domain

$$\text{DTFT}\{y[n]\,2\cos\omega_c n\}$$



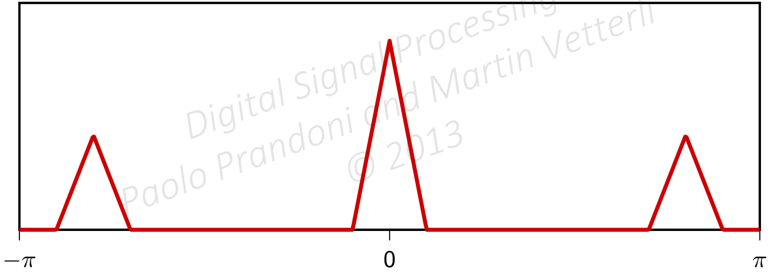
Demodulation in the frequency domain

$$\text{DTFT}\left\{y[n]\, 2\cos\omega_c n\right\}$$



Demodulation in the frequency domain

$$\text{DTFT}\left\{y[n]\cos\omega_c n\right\}$$



Demodulation in the frequency domain

- we recovered the baseband signal exactly...
- but we have some spurious high-frequency components
- in the next Module we will learn how to get rid of them!

Demodulation in the frequency domain

- we recovered the baseband signal exactly...
- but we have some spurious high-frequency components
- in the next Module we will learn how to get rid of them!

Demodulation in the frequency domain

- we recovered the baseband signal exactly...
- but we have some spurious high-frequency components
- in the next Module we will learn how to get rid of them!

Another application: tuning a guitar

Problem (abstraction):

- reference sinusoid at frequency ω_0
- tunable sinusoid of frequency ω
- make $\omega = \omega_0$ "by ear"

Another application: tuning a guitar

Problem (abstraction):

- reference sinusoid at frequency ω_0
- tunable sinusoid of frequency ω
- make $\omega = \omega_0$ "by ear"

Another application: tuning a guitar

Problem (abstraction):

- reference sinusoid at frequency ω_0
- tunable sinusoid of frequency ω
- make $\omega = \omega_0$ "by ear"

The procedure

1. bring ω close to ω_0 (easy)
2. when $\omega \approx \omega_0$ play both sinusoids together
3. trigonometry comes to the rescue:

$$\begin{aligned} x[n] &= \cos(\omega_0 n) + \cos(\omega n) \\ &= 2\cos\left(\frac{\omega_0 + \omega}{2} n\right) \cos\left(\frac{\omega_0 - \omega}{2} n\right) \\ &\approx 2\cos(\Delta_\omega n)\cos(\omega_0 n) \end{aligned}$$

The procedure

1. bring ω close to ω_0 (easy)
2. when $\omega \approx \omega_0$ play both sinusoids together
3. trigonometry comes to the rescue:

$$
\begin{aligned}
x[n] &= \cos(\omega_0 n) + \cos(\omega n) \\
&= 2\cos\left(\frac{\omega_0 + \omega}{2} n\right)\cos\left(\frac{\omega_0 - \omega}{2} n\right) \\
&\approx 2\cos(\Delta_\omega n)\cos(\omega_0 n)
\end{aligned}
$$

The procedure

1. bring ω close to ω_0 (easy)
2. when $\omega \approx \omega_0$ play both sinusoids together
3. trigonometry comes to the rescue:

$$
\begin{aligned}
x[n] &= \cos(\omega_0 n) + \cos(\omega n) \\
&= 2\cos\left(\frac{\omega_0 + \omega}{2} n\right) \cos\left(\frac{\omega_0 - \omega}{2} n\right) \\
&\approx 2\cos(\Delta_\omega n)\cos(\omega_0 n)
\end{aligned}
$$

Let's see what's happening

$$x[n] \approx 2\cos(\Delta_\omega n)\cos(\omega_0 n)$$

- "error" signal
- modulation at ω_0
- when $\omega \approx \omega_0$, the error signal is too low to be heard; modulation brings it up to hearing range and we perceive it as amplitude oscillations of the carrier frequency

Let's see what's happening

$$x[n] \approx \boxed{2\cos(\Delta_\omega n)} \cdot \boxed{\cos(\omega_0 n)}$$

- "error" signal
- modulation at ω_0
- when $\omega \approx \omega_0$, the error signal is too low to be heard; modulation brings it up to hearing range and we perceive it as amplitude oscillations of the carrier frequency

Let's see what's happening

$$x[n] \approx \boxed{2\cos(\Delta_\omega n)} \cdot \boxed{\cos(\omega_0 n)}$$

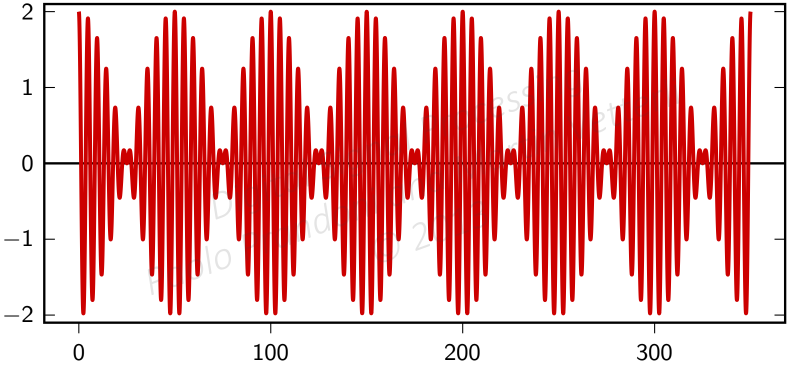
- "error" signal
- modulation at ω_0
- when $\omega \approx \omega_0$, the error signal is too low to be heard; modulation brings it up to hearing range and we perceive it as amplitude oscillations of the carrier frequency

Let's see what's happening

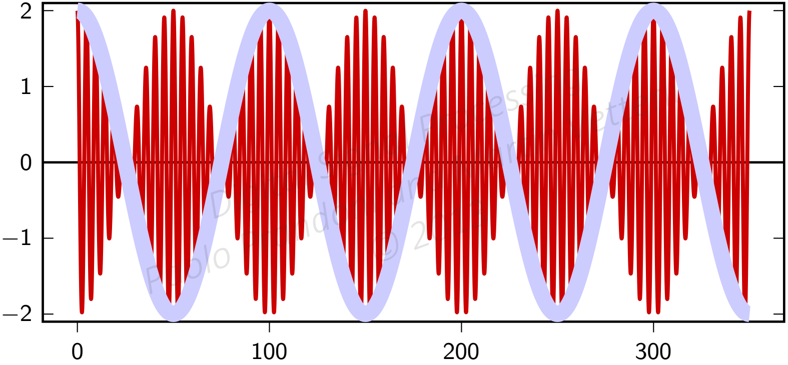
$$x[n] \approx \boxed{2\cos(\Delta_\omega n)} \cdot \boxed{\cos(\omega_0 n)}$$

- "error" signal
- modulation at ω_0
- when $\omega \approx \omega_0$, the error signal is too low to be heard; modulation brings it up to hearing range and we perceive it as amplitude oscillations of the carrier frequency

In the time domain...

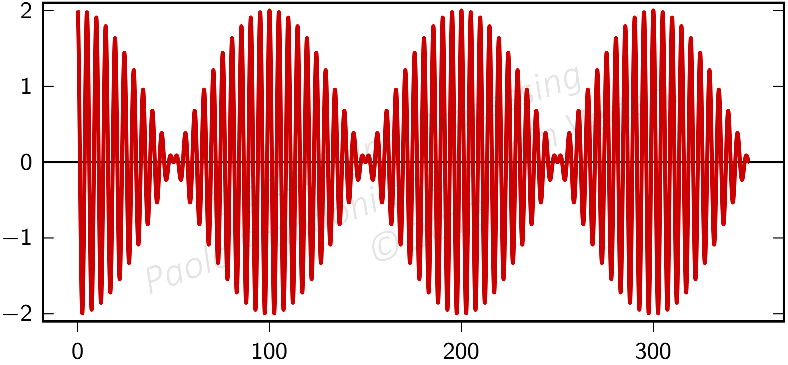


$$\omega_0 = 2\pi \cdot 0.2, \quad \omega = 2\pi \cdot 0.22, \quad \Delta_\omega = 2\pi \cdot 0.0100$$



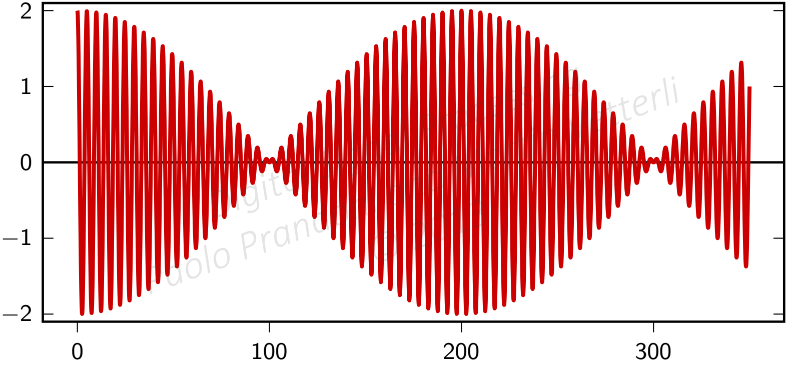
$$\omega_0 = 2\pi \cdot 0.2, \quad \omega = 2\pi \cdot 0.22, \quad \Delta_\omega = 2\pi \cdot 0.0100$$

In the time domain...



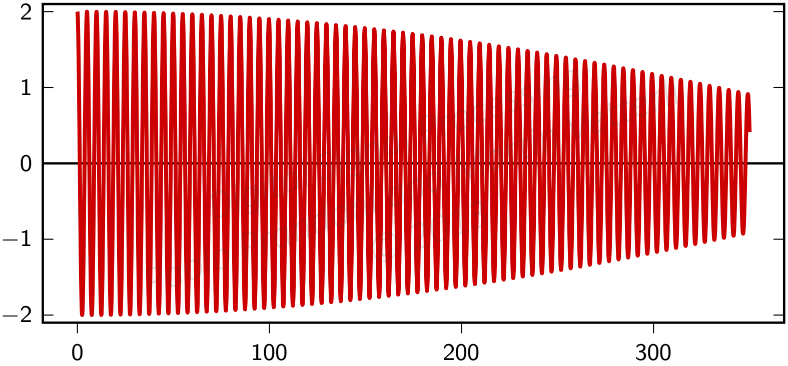
$$\omega_0 = 2\pi \cdot 0.2, \quad \omega = 2\pi \cdot 0.21, \quad \Delta_\omega = 2\pi \cdot 0.0050$$

In the time domain...



$$\omega_0 = 2\pi \cdot 0.2, \quad \omega = 2\pi \cdot 0.205, \quad \Delta_\omega = 2\pi \cdot 0.0025$$

In the time domain...



$$\omega_0 = 2\pi \cdot 0.2, \quad \omega = 2\pi \cdot 0.201, \quad \Delta_\omega = 2\pi \cdot 0.0005$$

Video demonstration

END OF MODULE 4.7

Digital Signal Processing

Module 4.8: The Short-Time Fourier Transform

Overview:

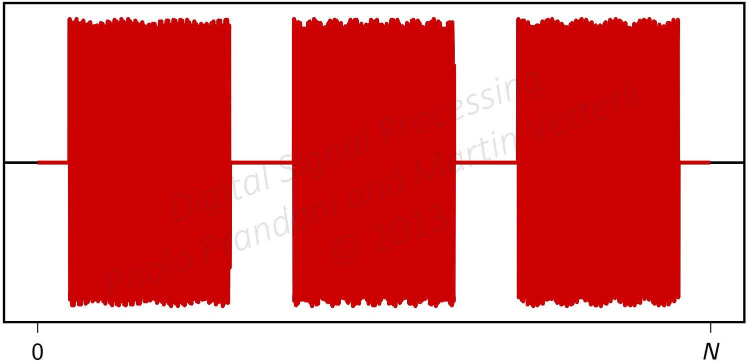
- Time vs frequency representations
- The STFT and the spectrogram
- Time-Frequency tilings

Dual-Tone Multi Frequency dialing

DTMF signaling

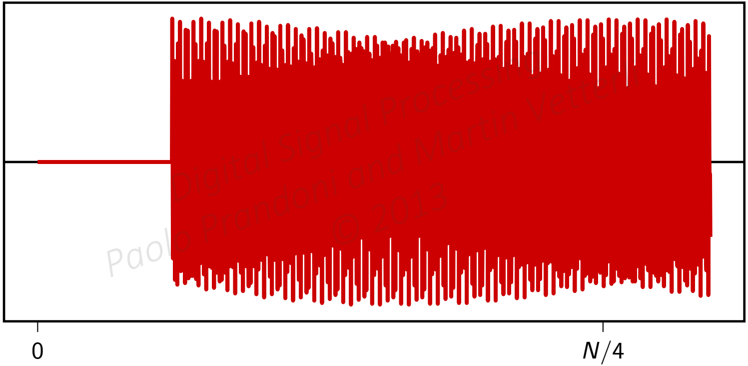
	1209Hz	1336Hz	1477Hz
697Hz	1	2	3
770Hz	4	5	6
852Hz	7	8	9
941Hz	*	0	#

1-5-9 in time

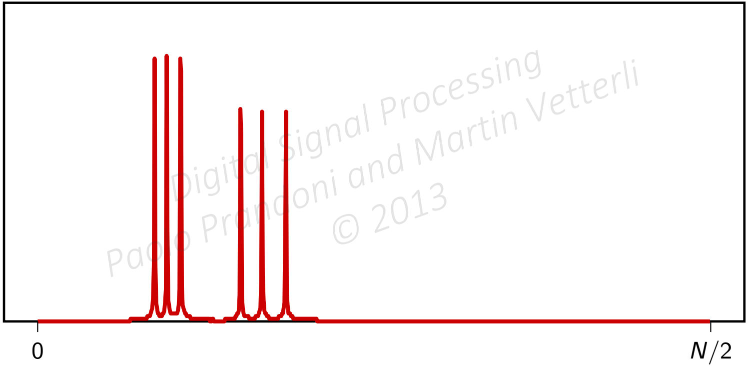


Play

1-5-9 in time (detail)



1-5-9 in frequency (magnitude)



The fundamental tradeoff

- time representation obfuscates frequency
- frequency representation obfuscates time

Idea:

- take small signal pieces of length L
- look at the DFT of each piece:

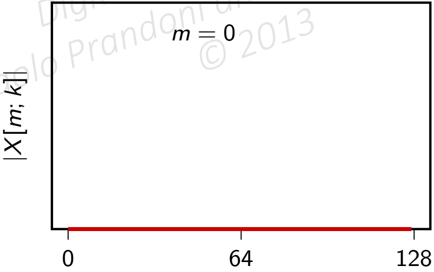
$$X[m;k] = \sum_{n=0}^{L-1} x[m+n]\, e^{-j\frac{2\pi}{L}nk}$$

Idea:

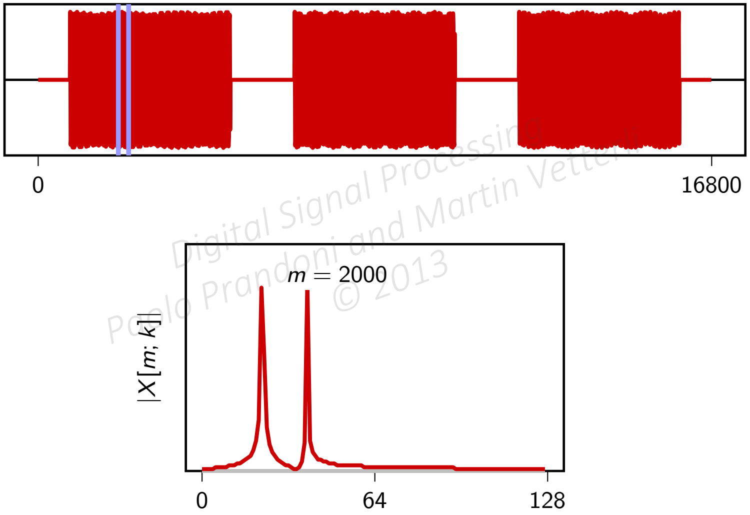
- take small signal pieces of length L
- look at the DFT of each piece:

$$X[m;k] = \sum_{n=0}^{L-1} x[m+n]\, e^{-j\frac{2\pi}{L}nk}$$

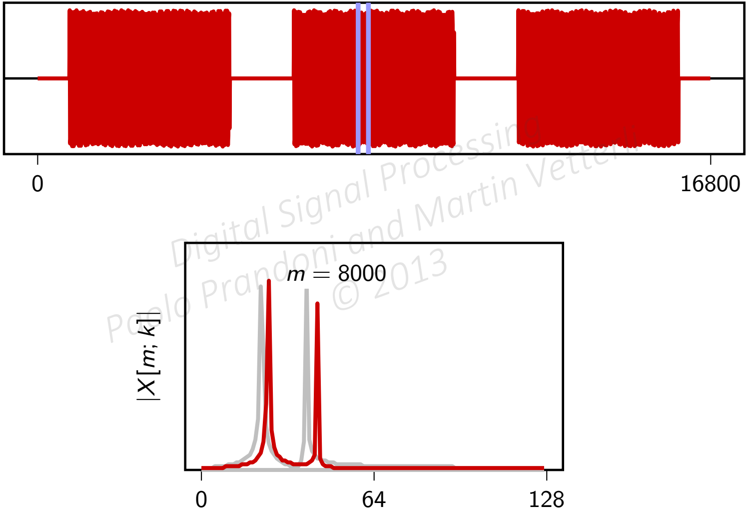
Short-Time Fourier Transform ($L = 256$)



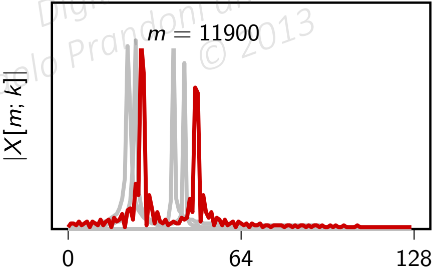
Short-Time Fourier Transform ($L = 256$)



Short-Time Fourier Transform ($L = 256$)



Short-Time Fourier Transform ($L = 256$)



Idea:

- color-code the magnitude: dark is small, white is large
- use $10\log_{10}(|X[m;k]|)$ to see better (power in dBs)
- plot spectral slices one after another

The Spectrogram

Idea:

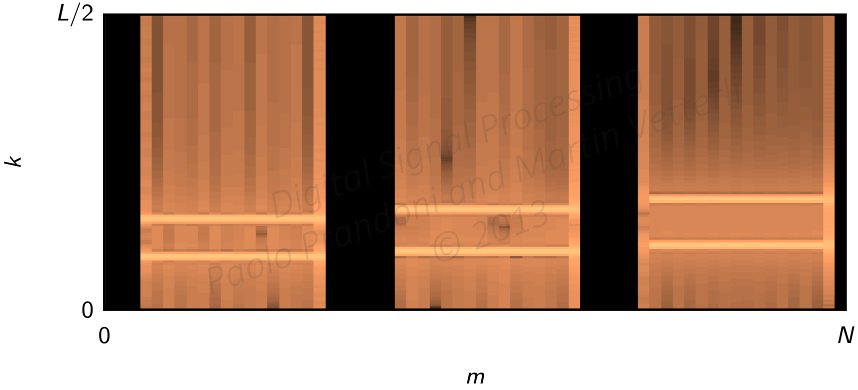
- color-code the magnitude: dark is small, white is large
- use $10\log_{10}(|X[m;k]|)$ to see better (power in dBs)
- plot spectral slices one after another

The Spectrogram

Idea:

- color-code the magnitude: dark is small, white is large
- use $10\log_{10}(|X[m;k]|)$ to see better (power in dBs)
- plot spectral slices one after another

DTMF spectrogram



Labeling the Spectrogram

If we know the "system clock" $F_s = 1/T_s$ we can label the axis

- highest positive frequency: $F_s/2$ Hz
- frequency resolution: F_s/L Hz
- width of time slices: LT_s seconds

Labeling the Spectrogram

If we know the "system clock" $F_s = 1/T_s$ we can label the axis

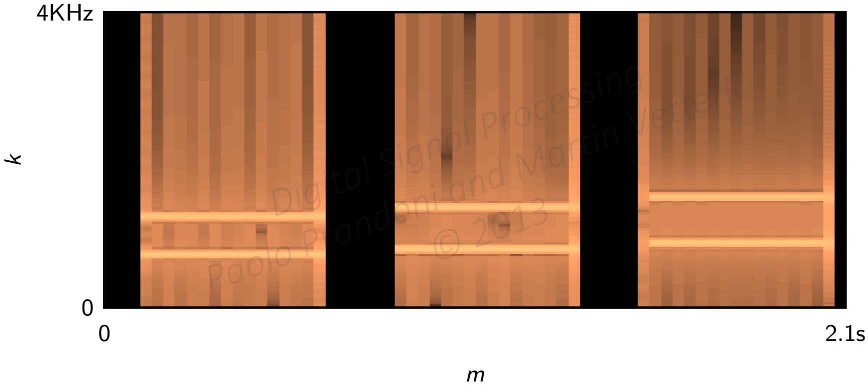
- highest positive frequency: $F_s/2$ Hz
- frequency resolution: F_s/L Hz
- width of time slices: LT_s seconds

Labeling the Spectrogram

If we know the "system clock" $F_s = 1/T_s$ we can label the axis

- highest positive frequency: $F_s/2$ Hz
- frequency resolution: F_s/L Hz
- width of time slices: LT_s seconds

DTMF spectrogram ($F_s = 8000$)



Questions:

- width of the analysis window?
- position of the windows (overlapping?)
- shape of the window (weighing the samples)

Questions:

- width of the analysis window?
- position of the windows (overlapping?)
- shape of the window (weighing the samples)

Questions:

- width of the analysis window?
- position of the windows (overlapping?)
- shape of the window (weighing the samples)

Wideband vs Narrowband

Long window: narrowband spectrogram

- long window $\Rightarrow$ more DFT points $\Rightarrow$ more frequency resolution
- long window $\Rightarrow$ more "things can happen" $\Rightarrow$ less precision in time

Short window: wideband spectrogram

- short window $\Rightarrow$ many time slices $\Rightarrow$ precise location of transitions
- short window $\Rightarrow$ fewer DFT points $\Rightarrow$ poor frequency resolution

Long window: narrowband spectrogram

- long window $\Rightarrow$ more DFT points $\Rightarrow$ more frequency resolution
- long window $\Rightarrow$ more "things can happen" $\Rightarrow$ less precision in time

Short window: wideband spectrogram

- short window $\Rightarrow$ many time slices $\Rightarrow$ precise location of transitions
- short window $\Rightarrow$ fewer DFT points $\Rightarrow$ poor frequency resolution

Wideband vs Narrowband

Long window: narrowband spectrogram

- long window ⇒ more DFT points ⇒ more frequency resolution
- long window ⇒ more "things can happen" ⇒ less precision in time

Short window: wideband spectrogram

- short window ⇒ many time slices ⇒ precise location of transitions
- short window ⇒ fewer DFT points ⇒ poor frequency resolution

Wideband vs Narrowband

Long window: narrowband spectrogram

- long window $\Rightarrow$ more DFT points $\Rightarrow$ more frequency resolution
- long window $\Rightarrow$ more "things can happen" $\Rightarrow$ less precision in time

Short window: wideband spectrogram

- short window $\Rightarrow$ many time slices $\Rightarrow$ precise location of transitions
- short window $\Rightarrow$ fewer DFT points $\Rightarrow$ poor frequency resolution

Wideband vs Narrowband

Long window: narrowband spectrogram

- long window $\Rightarrow$ more DFT points $\Rightarrow$ more frequency resolution
- long window $\Rightarrow$ more "things can happen" $\Rightarrow$ less precision in time

Short window: wideband spectrogram

- short window $\Rightarrow$ many time slices $\Rightarrow$ precise location of transitions
- short window $\Rightarrow$ fewer DFT points $\Rightarrow$ poor frequency resolution

Wideband vs Narrowband

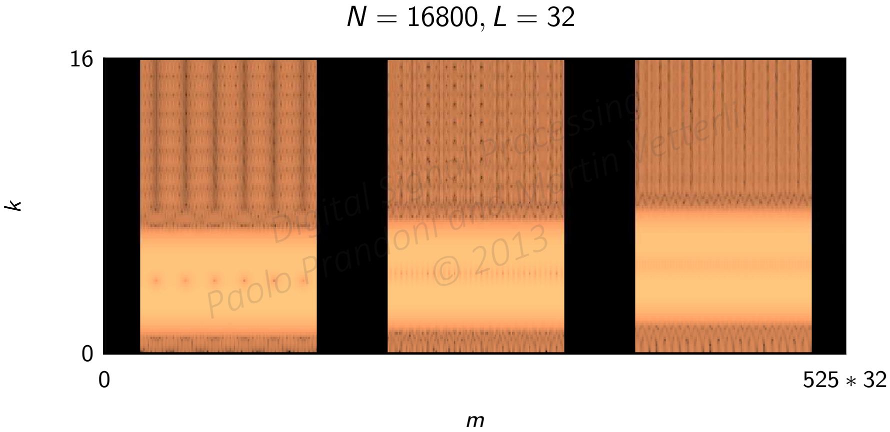
Long window: narrowband spectrogram

- long window $\Rightarrow$ more DFT points $\Rightarrow$ more frequency resolution
- long window $\Rightarrow$ more "things can happen" $\Rightarrow$ less precision in time

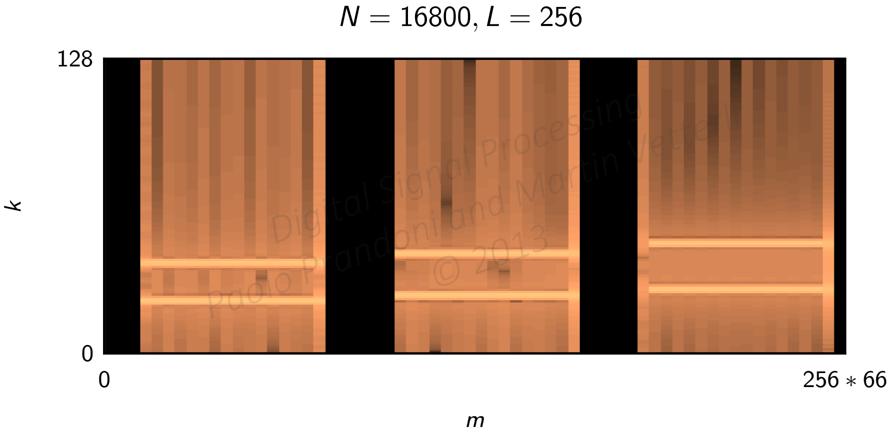
Short window: wideband spectrogram

- short window $\Rightarrow$ many time slices $\Rightarrow$ precise location of transitions
- short window $\Rightarrow$ fewer DFT points $\Rightarrow$ poor frequency resolution

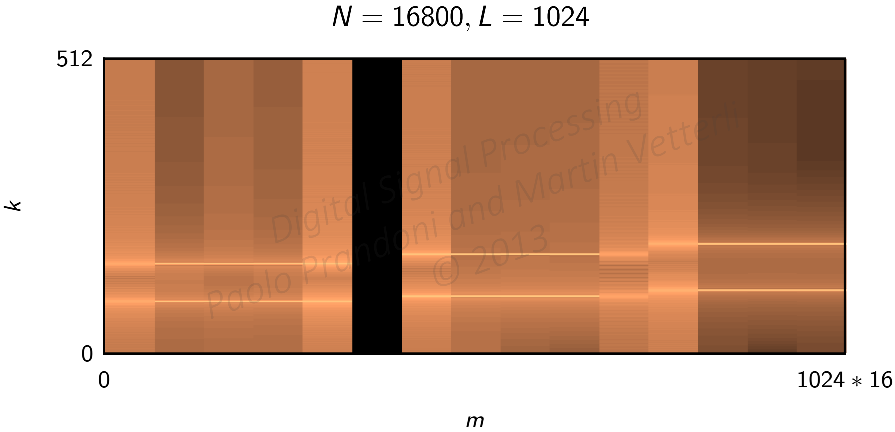
DTMF spectrogram (wideband)

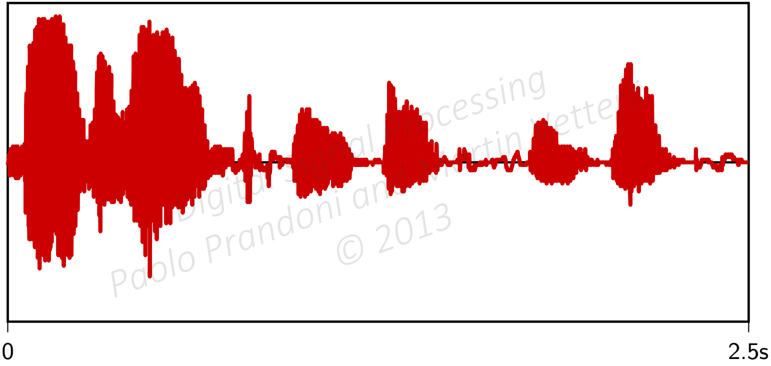


DTMF spectrogram



DTMF spectrogram (narrowband)

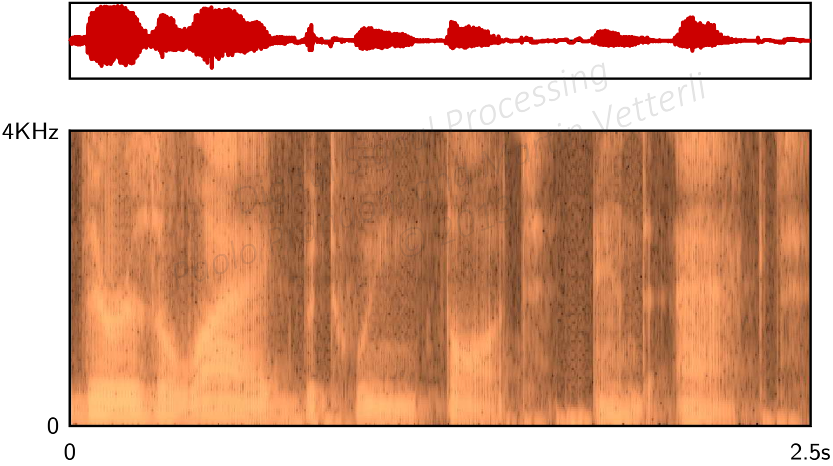



0
2.5s
Digital Signal Processing
Paolo Prandoni and Martin Vetterli
© 2013

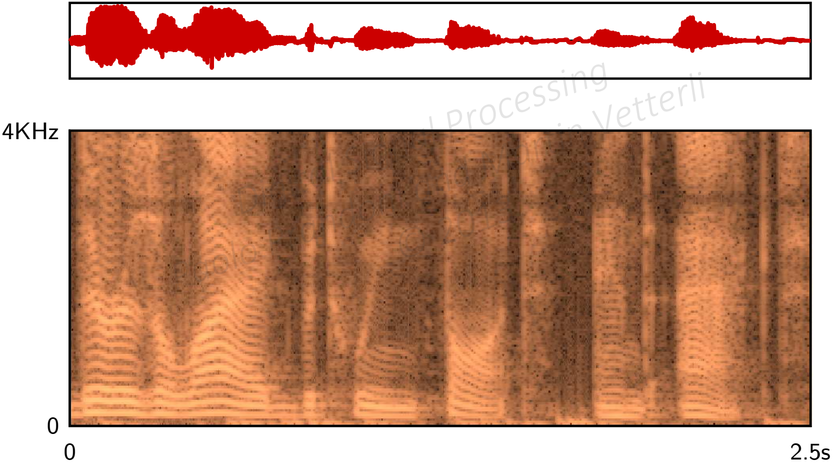
Play

Speech analysis

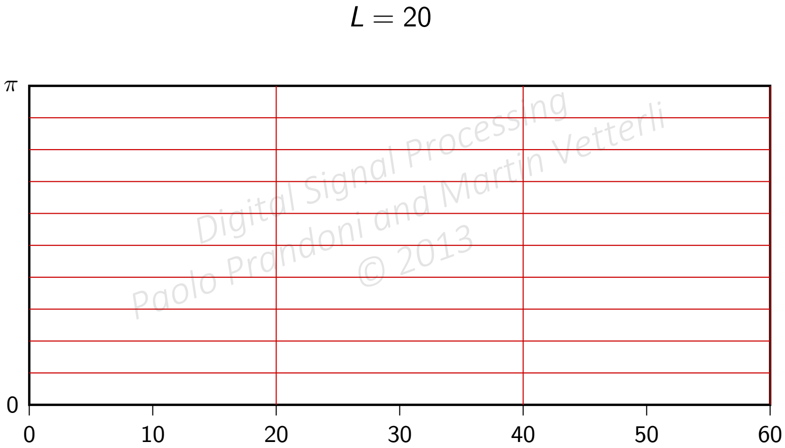
8ms analysis window (125Hz frequency bins) , 4ms shifts



32ms analysis window (31Hz frequency bins), 4ms shifts

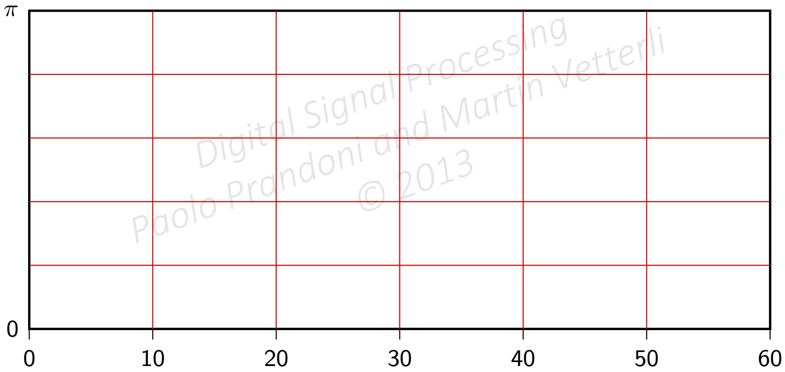


Time-Frequency tiling

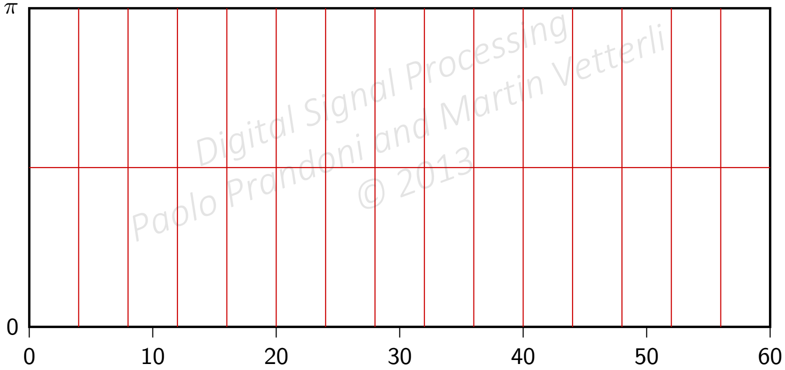


Time-Frequency tiling

$L = 10$



$L = 4$



Food for thought

- time “resolution” $\Delta t = L$
- frequency “resolution” $\Delta f = 2\pi/L$
- $\Delta t \Delta f = 2\pi$

uncertainty principle!

Food for thought

- time "resolution" $\Delta t = L$
- frequency "resolution" $\Delta f = 2\pi/L$
- $\Delta t \Delta f = 2\pi$

uncertainty principle!

Food for thought

- time "resolution" $\Delta t = L$
- frequency "resolution" $\Delta f = 2\pi/L$
- $\Delta t \Delta f = 2\pi$

uncertainty principle!

Food for thought

- time "resolution" $\Delta t = L$
- frequency "resolution" $\Delta f = 2\pi/L$
- $\Delta t \Delta f = 2\pi$

uncertainty principle!

Even more food for thought

more sophisticated tilings of the time-frequency planes
can be obtained with the *wavelet* transform

END OF MODULE 4.8

Digital Signal Processing

Module 4.9: The FFT, History, Factorizations and Algorithms

Overview

- A bit of history: From Gauss to the fastest FFT in the west
- Small DFT matrices
- The Cooley-Tukey FFT
- Decimation-in-Time FFT for length 2^N FFTs
- Conclusions: There are FFTs for any length!

Fourier had the Fourier transform

But Gauss had the FFT all along ;)

History

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

History

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

History

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

History

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

History

- Gauss computes trigonometric series efficiently in 1805
- Fourier invents Fourier series in 1807
- People start computing Fourier series, and develop tricks
- Good comes up with an algorithm in 1958
- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for N a power of a prime
- Winograd combines all methods to give the most efficient FFTs

The DFT matrix

- $W_N = e^{-j\frac{2\pi}{N}}$ (or simply W when N is clear from the context)
- powers of N can be taken modulo N, since $W^N = 1$.
- DFT Matrix of size N by N:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

The DFT matrix

- $W_N = e^{-j\frac{2\pi}{N}}$ (or simply W when N is clear from the context)
- powers of N can be taken modulo N, since $W^N = 1$.
- DFT Matrix of size N by N:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

The DFT matrix

- $W_N = e^{-j\frac{2\pi}{N}}$ (or simply W when N is clear from the context)
- powers of N can be taken modulo N, since $W^N = 1$.
- DFT Matrix of size N by N:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W^1 & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{2(N-1)} \\ & & & \dots & & \\ 1 & W^{N-1} & W^{2(N-1)} & W^{3(N-1)} & \dots & W^{(N-1)^2} \end{bmatrix}$$

Small DFT matrices: $N = 2$

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Small DFT matrices: $N = 3$

$$\mathbf{W}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W & W^2 \\ 1 & W^2 & W^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W & W^2 \\ 1 & W^2 & W \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \frac{-1-j\sqrt{3}}{2} & \frac{-1+j\sqrt{3}}{2} \\ 1 & \frac{-1+j\sqrt{3}}{2} & \frac{-1-j\sqrt{3}}{2} \end{bmatrix}$$

Small DFT matrices: $N = 4$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 \\ 1 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 \\ 1 & W^2 & 1 & W^2 \\ 1 & W^3 & W^2 & W \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Small DFT matrices: $N = 5$

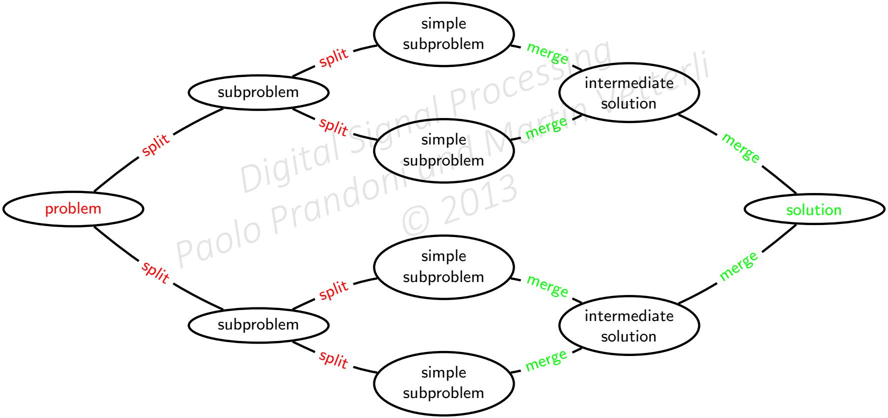
$$\mathbf{W}_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 \\ 1 & W^2 & W^4 & W^6 & W^8 \\ 1 & W^3 & W^6 & W^9 & W^{12} \\ 1 & W^4 & W^8 & W^{12} & W^{16} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 \\ 1 & W^2 & W^4 & W & W^3 \\ 1 & W^3 & W & W^4 & W^2 \\ 1 & W^4 & W^3 & W^2 & W \end{bmatrix}$$

Small DFT matrices: $N = 6$

$$\mathbf{W}_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 & W^5 \\ 1 & W^2 & W^4 & W^6 & W^8 & W^{10} \\ 1 & W^3 & W^6 & W^9 & W^{12} & W^{15} \\ 1 & W^4 & W^8 & W^{12} & W^{16} & W^{20} \\ 1 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 & W^5 \\ 1 & W^2 & W^4 & 1 & W^2 & W^4 \\ 1 & W^3 & 1 & W^3 & 1 & W^3 \\ 1 & W^4 & W^2 & 1 & W^4 & W^2 \\ 1 & W^5 & W^4 & W^3 & W^2 & W \end{bmatrix}$$

Divide et impera - Divide and Conquer (Julius Caesar)

Divide and conquer is a standard attack for developing fast algorithms.



Recall: computing $\mathbf{X} = \mathbf{W}_N\ \mathbf{x}$ has complexity $O(N^2)$.

Idea:

- Take a problem of size N where N is a power of 2.
- Cut into two problems of size $N/2$ that use complexity $\frac{N^2}{4}$ each,
- There might be some complexity to recover the full solution, say N.
- The divide-and-conquer solution has complexity $N^2/2 + N$ for one step
- For $N \geq 4$ this is better than N^2!

Divide and Conquer for DFT - One step

Recall: computing $\mathbf{X} = \mathbf{W}_N \mathbf{x}$ has complexity $O(N^2)$.

Idea:

- Take a problem of size N where N is a power of 2.
- Cut into two problems of size $N/2$ that use complexity $\frac{N^2}{4}$ each,
- There might be some complexity to recover the full solution, say N.
- The divide-and-conquer solution has complexity $N^2/2 + N$ for one step
- For $N \geq 4$ this is better than N^2!

Divide and Conquer for DFT - One step

Recall: computing $\mathbf{X} = \mathbf{W}_N \, \mathbf{x}$ has complexity $O(N^2)$.

Idea:

- Take a problem of size N where N is a power of 2.
- Cut into two problems of size $N/2$ that use complexity $\frac{N^2}{4}$ each,
- There might be some complexity to recover the full solution, say N.
- The divide-and-conquer solution has complexity $N^2/2 + N$ for one step
- For $N \geq 4$ this is better than N^2!

Divide and Conquer for DFT - One step

Recall: computing $\mathbf{X} = \mathbf{W}_N\ \mathbf{x}$ has complexity $O(N^2)$.

Idea:

- Take a problem of size N where N is a power of 2.
- Cut into two problems of size $N/2$ that use complexity $\frac{N^2}{4}$ each,
- There might be some complexity to recover the full solution, say N.
- The divide-and-conquer solution has complexity $N^2/2 + N$ for one step
- For $N \geq 4$ this is better than N^2!

Recall: computing $\mathbf{X} = \mathbf{W}_N\ \mathbf{x}$ has complexity $O(N^2)$.

Idea:

- Take a problem of size N where N is a power of 2.
- Cut into two problems of size $N/2$ that use complexity $\frac{N^2}{4}$ each,
- There might be some complexity to recover the full solution, say N.
- The divide-and-conquer solution has complexity $N^2/2 + N$ for one step
- For $N \geq 4$ this is better than N^2!

Divide and Conquer for DFT - One step

Graphically

- Split DFT input into 2 pieces of size $N/2$

Divide and Conquer for DFT - One step

Graphically

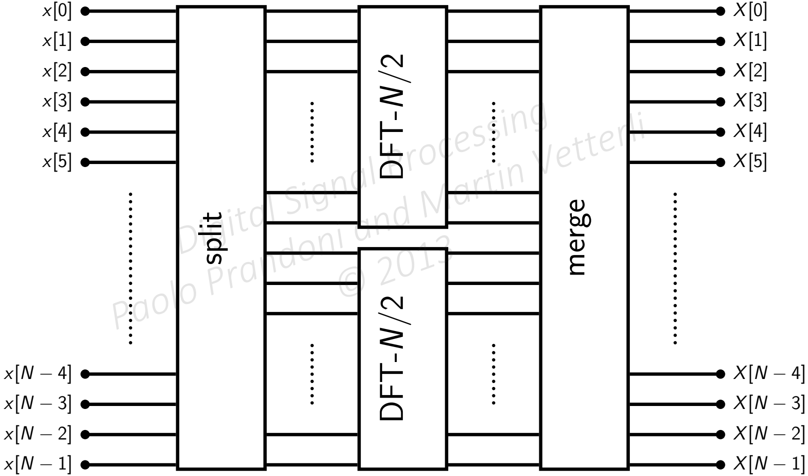
- Split DFT input into 2 pieces of size $N/2$
- Compute two DFT's of size $N/2$

Divide and Conquer for DFT - One step

Graphically

- Split DFT input into 2 pieces of size $N/2$
- Compute two DFT's of size $N/2$
- Merge the two results

Divide and Conquer for DFT - One step



Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Divide and conquer can be reapplied!

- If it worked once, it will work again (recall, $N = 2^K$)
- Cut the two problems of size $N/2$ into 4 problems of size $N/4$
- There might be some complexity to recover the full solution, say N at each step
- You can do this $\log_2 N - 1 = K - 1$ times, until problem of size 2 is obtained
- This requires order N complexity each time
- The divide-and-conquer solution has therefore complexity of order $N \log_2 N$
- For $N \geq 4$ this is much better than N^2!

Divide and Conquer for DFT - Multiple steps

Graphically

- Split DFT input into 2, 4 and 8 pieces of sizes $N/2$, $N/4$ and $N/8$, respectively
- Compute 8 DFT's of size $N/8$
- Merge the results successively into DFT's of size $N/4$, $N/2$ and finally N

Divide and Conquer for DFT - Multiple steps

Graphically

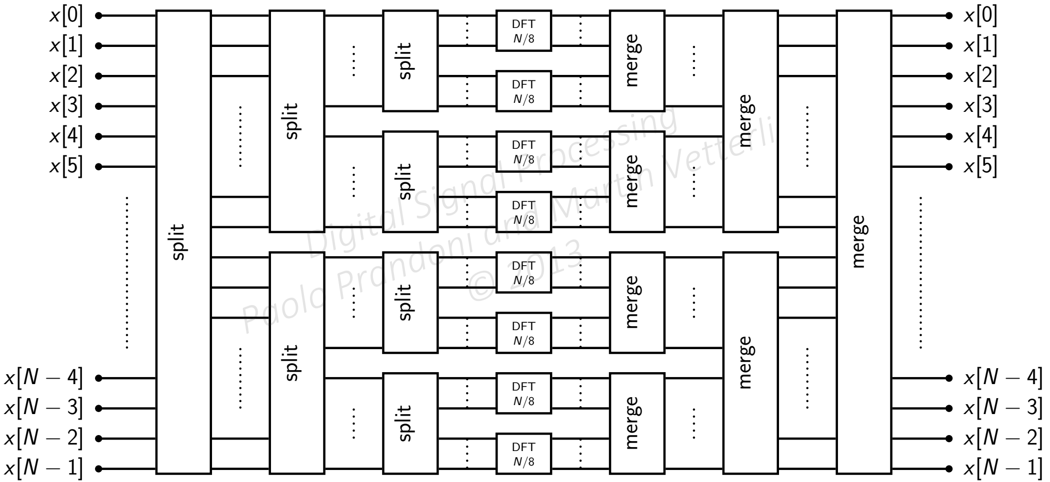
- Split DFT input into 2, 4 and 8 pieces of sizes $N/2$, $N/4$ and $N/8$, respectively
- Compute 8 DFT's of size $N/8$
- Merge the results successively into DFT's of size $N/4$, $N/2$ and finally N

Divide and Conquer for DFT - Multiple steps

Graphically

- Split DFT input into 2, 4 and 8 pieces of sizes $N/2$, $N/4$ and $N/8$, respectively
- Compute 8 DFT's of size $N/8$
- Merge the results successively into DFT's of size $N/4$, $N/2$ and finally N

Divide and Conquer for DFT - Multiple steps



$$X[k] = \sum_{n=0}^{N-1} x[n]\, W^{nk}, \qquad k = 0, 1, \ldots, N-1, \quad W = e^{-j\frac{2\pi}{N}}$$

Idea (a good guess is half of the answer!):

- break input into even and odd indexed terms (so-called "decimation in time"):

$$x[n], n = 0, 1, \ldots, N-1 \longrightarrow x[2n] \text{ and } x[2n+1], n = 0, 1, \ldots, \frac{N}{2} - 1$$

- break output into first and second half

$$X[k], k = 0, 1, \ldots, N-1 \longrightarrow X[k] \text{ and } X[k+N/2], k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W^{nk}, \qquad k = 0, 1, \ldots, N-1, \quad W = e^{-j\frac{2\pi}{N}}$$

Idea (a good guess is half of the answer!):

- break input into even and odd indexed terms (so-called "decimation in time"):

$$x[n], n = 0, 1, \ldots, N-1 \longrightarrow x[2n] \text{ and } x[2n+1], n = 0, 1, \ldots, \frac{N}{2} - 1$$

- break output into first and second half

$$X[k], k = 0, 1, \ldots, N-1 \longrightarrow X[k] \text{ and } X[k+N/2], k = 0, 1, \ldots, \frac{N}{2} - 1$$

Divide and Conquer for DFT- Analysis of DIT-1/8

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W^{nk}, \qquad k = 0, 1, \ldots, N-1, \quad W = e^{-j\frac{2\pi}{N}}$$

Idea (a good guess is half of the answer!):

- break input into even and odd indexed terms (so-called "decimation in time"):

$$x[n], n = 0, 1, \ldots, N-1 \longrightarrow x[2n] \text{ and } x[2n+1], n = 0, 1, \ldots, \frac{N}{2} - 1$$

- break output into first and second half

$$X[k], k = 0, 1, \ldots, N-1 \longrightarrow X[k] \text{ and } X[k+N/2], k = 0, 1, \ldots, \frac{N}{2} - 1$$

Divide and Conquer for DFT- Analysis of DIT-2/8

Consider even and odd inputs separately:

$$X[k] = \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{(2n+1)k}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{2nk+k}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} + W^k \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk}$$

$$= X'_k + W^k\, X"_k, \quad k = 0, 1, \ldots, N-1$$

Divide and Conquer for DFT- Analysis of DIT-2/8

Consider even and odd inputs separately:

$$X[k] = \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{(2n+1)k}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{2nk+k}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} + W^k \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk}$$

$$= X'_k + W^k\, X"_k, \quad k = 0, 1, \ldots, N-1$$

Consider even and odd inputs separately:

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{(2n+1)k} \\
&= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{2nk+k} \\
&= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} + W^{k} \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk} \\
&= X'_k + W^k X''_k, \quad k = 0, 1, \ldots, N-1
\end{aligned}
$$

Consider even and odd inputs separately:

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{(2n+1)k} \\
&= \sum_{n=0}^{N/2-1} x[2n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\, W^{2nk+k} \\
&= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} + W^{k} \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk} \\
&= X'_k + W^k\, X''_k, \quad k = 0, 1, \ldots, N-1
\end{aligned}
$$

In words: We can compute X with:

- 2 half-size DFT's, which we call X' and X''
- multiplying the second DFT by W^k
- adding the result

In words: We can compute X with:

- 2 half-size DFT's, which we call X' and X"
- multiplying the second DFT by W^k
- adding the result

In words: We can compute X with:

- 2 half-size DFT's, which we call X' and X"
- multiplying the second DFT by W^k
- adding the result

Divide and Conquer for DFT- Analysis of DIT-4/8

Consider now the first and second half of the outputs separately:

$$X[k] = X'_k + W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X[k + N/2] = \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{n(k+N/2)} + W^{k+N/2} \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{n(k+N/2)}$$

$$= \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{nk} - W^k \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{nk}$$

$$= X'_k - W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Divide and Conquer for DFT- Analysis of DIT-4/8

Consider now the first and second half of the outputs separately:

$$X[k] = X'_k + W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X[k + N/2] = \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{n(k+N/2)} + W^{k+N/2} \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{n(k+N/2)}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} - W^k \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk}$$

$$= X'_k - W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Divide and Conquer for DFT- Analysis of DIT-4/8

Consider now the first and second half of the outputs separately:

$$X[k] = X'_k + W^k\, X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X[k+N/2] = \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{n(k+N/2)} + W^{k+N/2} \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{n(k+N/2)}$$

$$= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} - W^k \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk}$$

$$= X'_k - W^k\, X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Consider now the first and second half of the outputs separately:

$$X[k] = X'_k + W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$\begin{aligned}
X[k + N/2] &= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{n(k+N/2)} + W^{k+N/2} \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{n(k+N/2)} \\
&= \sum_{n=0}^{N/2-1} x[2n]\, W_{N/2}^{nk} - W^k \sum_{n=0}^{N/2-1} x[2n+1]\, W_{N/2}^{nk} \\
&= X'_k - W^k X"_k, \quad k = 0, 1, \ldots, \frac{N}{2} - 1
\end{aligned}$$

In words: We can compute $X[k]$ and $X[k+N/2]$ with:

- Divide input into even and odd indexed samples
- Compute two DFTs of size $N/2$
- Multiplication of the output of the second DFT by W^k using $N/2$ multiplications
- Combine output with sum/difference

In words: We can compute $X[k]$ and $X[k + N/2]$ with:

- Divide input into even and odd indexed samples
- Compute two DFTs of size $N/2$
- Multiplication of the output of the second DFT by W^k using $N/2$ multiplications
- Combine output with sum/difference

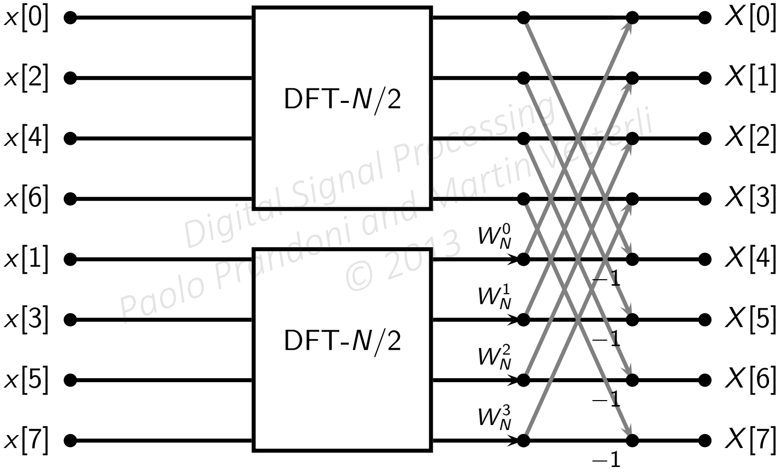
In words: We can compute $X[k]$ and $X[k+N/2]$ with:

- Divide input into even and odd indexed samples
- Compute two DFTs of size $N/2$
- Multiplication of the output of the second DFT by W^k using $N/2$ multiplications
- Combine output with sum/difference

In words: We can compute $X[k]$ and $X[k+N/2]$ with:

- Divide input into even and odd indexed samples
- Compute two DFTs of size $N/2$
- Multiplication of the output of the second DFT by W^k using $N/2$ multiplications
- Combine output with sum/difference

Divide and Conquer for DFT- Analysis of DIT-6/8



So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what is the complexity now?

- Split DFT input into 2 pieces of size N/2: free!
- Compute 2 DFT-N/2: twice $(N/2)^2$, or $N^2/2$
- Merge the two results: multiplication by $N/2$ complex numbers W^k
- Total: $N^2/2 + N/2$ which is indeed smaller than N^2 for any $N \geq 4$,
- In general, about half the complexity of the initial problem!

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

Divide and Conquer for DFT- Analysis of DIT-8/8

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

Divide and Conquer for DFT- Analysis of DIT-8/8

So, what if we repeat the process?

- Go until DFT-2, since this is trivial (sum and difference)
- Requires $\log_2 N - 1$ steps
- Each step requires a merger of order $N/2$ multiplications and N additions
- Total: $N/2(\log_2 N - 1)$ multiplications and $N \log_2 N$ additions
- Savings of order $\log_2 N/N$

Key Result: **A DFT of size N requires order $N \log_2 N$ operations!**

- Separate even and odd samples
- Compute two DFT's of size 2 having output $X'[k]$ and $X''[k]$
- Compute sum and difference of $X'[k]$ and $W^k X''[k]$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This uses 8 additions and no multiplications!

Matrix factorization view of DFT, N $= 4$

- Separate even and odd samples
- Compute two DFT's of size 2 having output $X'[k]$ and $X''[k]$
- Compute sum and difference of $X'[k]$ and $W^k X''[k]$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This uses 8 additions and no multiplications!

Matrix factorization view of DFT, N $= 4$

- Separate even and odd samples
- Compute two DFT's of size 2 having output $X'[k]$ and $X''[k]$
- Compute sum and difference of $X'[k]$ and $W^k X''[k]$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This uses 8 additions and no multiplications!

Matrix factorization view of DFT, N $= 4$

- Separate even and odd samples
- Compute two DFT's of size 2 having output $X'[k]$ and $X''[k]$
- Compute sum and difference of $X'[k]$ and $W^k X''[k]$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This uses 8 additions and no multiplications!

Now this is going to be big...

Too big for a single slide!

$$\mathbf{W}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 1 & W^1 & W^2 & W^3 & \ldots & W^7 \\ 1 & W^2 & W^4 & W^6 & \ldots & W^{14} \\ & & & \ldots & & \\ 1 & W^7 & W^{14} & W^{21} & \ldots & W^{49} \end{bmatrix} = \ldots$$

Matrix factorization view of DFT, $N = 8$, 2/8

Step 1: separate even from odd indexed samples
Call this $\mathbf{D}_8$ for decimation of size 8

$$\mathbf{D}_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This requires no arithmetic operations!

Step 2: Compute two DFTs of size N/2 on the even and on the odd indexed samples
Each submatrix is $\mathbf{W}_4$, and the matrix is block diagonal, where $\mathbf{0_4}$ stands for a matrix of 0's

$$\begin{bmatrix} \mathbf{W_4} & \mathbf{0_4} \\ \mathbf{0_4} & \mathbf{W_4} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & & & & \\ 1 & j & -1 & -j & & & & \\ 1 & -1 & 1 & -1 & & & & \\ 1 & -j & -1 & j & & & & \\ & & & & 1 & 1 & 1 & 1 \\ & & & & 1 & j & -1 & -j \\ & & & & 1 & -1 & 1 & -1 \\ & & & & 1 & -j & -1 & j \end{bmatrix}$$

This requires two DFT-4, or a total of 16 additions!

Matrix factorization view of DFT, N $= 8$, 4/8

Step 3: Multiply output of second DFT of size 4 by W^k
This is a diagonal matrix, with $\mathbf{I}_4$ for the identity of size 4,

$$\begin{bmatrix} \mathbf{I}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{\Lambda}_4 \end{bmatrix} = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ &&&&& W && \\ &&&&&& W^2 & \\ &&&&&&& W^3 \end{bmatrix} \text{ where } \quad \mathbf{\Lambda}_4 = \begin{bmatrix} 1 &&& \\ & W && \\ && W^2 & \\ &&& W^3 \end{bmatrix}$$

This requires 2 multiplications ($W^2 = -j$ is free)

Matrix factorization view of DFT, N = 8, 5/8

Step 4: Recombine final output $X[k]$ and $X[k + N/2]$ by sum and difference, $\mathbf{S}_8$

$$\mathbf{S}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

This requires 8 additions!

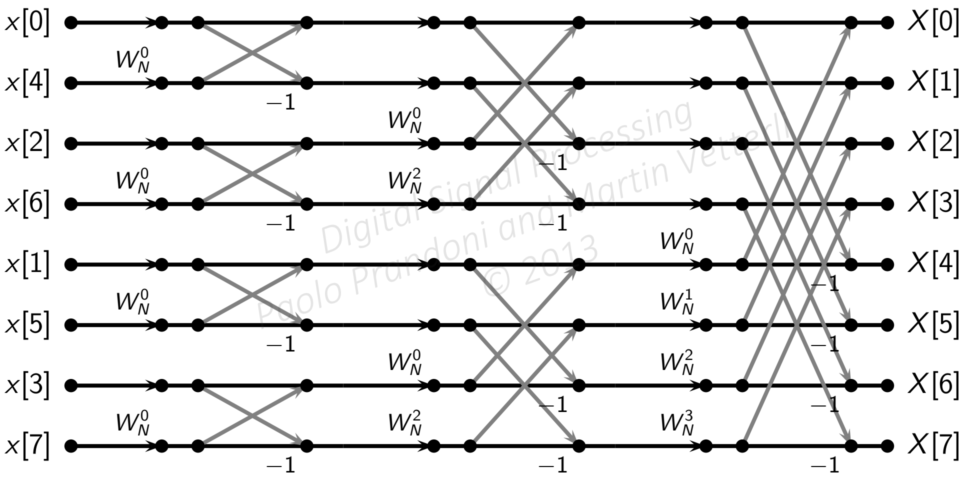
Matrix factorization view of DFT, N = 8, 6/8

In total:
Product of 4 matrices

$$\mathbf{W}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{\Lambda}_4 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{W}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{W}_4 \end{bmatrix} \cdot \mathbf{D}_8$$

This requires 24 additions and 2 multiplications!

Flowgraph view of DFT, N $= 8$, 7/8



Is this a big deal?

- In image processing (e.g. digital photography) one takes block of 8 by 8 pixels
- One computes a transform (called DCT) similar to a DFT
- It has a fast algorithm inspired by what we just saw

Matrix factorization view of DFT, $N = 8$, 8/8

Is this a big deal?

- In image processing (e.g. digital photography) one takes block of 8 by 8 pixels
- One computes a transform (called DCT) similar to a DFT
- It has a fast algorithm inspired by what we just saw

Is this a big deal?

- In image processing (e.g. digital photography) one takes block of 8 by 8 pixels
- One computes a transform (called DCT) similar to a DFT
- It has a fast algorithm inspired by what we just saw

Some numbers

- Direct: $64^2 = 4096$ multiplications required (dominant cost, fixed point multiplications)
- The transform can be computed in rows and columns separately, or 16 DFT's
- The algorithm we saw has 2 multiplications, or a total of 32 multiplications
- Saving of 2 orders of magnitude!

Some numbers

- Direct: $64^2 = 4096$ multiplications required (dominant cost, fixed point multiplications)
- The transform can be computed in rows and columns separately, or 16 DFT's
- The algorithm we saw has 2 multiplications, or a total of 32 multiplications
- Saving of 2 orders of magnitude!

Some numbers

- Direct: $64^2 = 4096$ multiplications required (dominant cost, fixed point multiplications)
- The transform can be computed in rows and columns separately, or 16 DFT's
- The algorithm we saw has 2 multiplications, or a total of 32 multiplications
- Saving of 2 orders of magnitude!

Some numbers

- Direct: $64^2 = 4096$ multiplications required (dominant cost, fixed point multiplications)
- The transform can be computed in rows and columns separately, or 16 DFT's
- The algorithm we saw has 2 multiplications, or a total of 32 multiplications
- Saving of 2 orders of magnitude!

Conclusions

Don't worry, be happy!

- The Cooley-Tukey is the most popular algorithm, mostly for $N = 2^N$
- Note that there is always a good FFT algorithm around the corner
- Do not zero-pad to lengthen a vector to have a size equal to a power of 2
- There are good packages out there (e.g. Fastest Fourier Transform in the West, SPIRAL)
- It does make a BIG difference!

And some people are obsessed with Fourier...

Exercise: Divide and Conquer for DFT- Analysis of DIF

Recall the computation of the DFT on an input $x[n]$ of length N

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W^{nk}, \qquad k = 0, 1, \ldots, N-1, \quad W = e^{-j\frac{2\pi}{N}}$$

with output $X[k]$ of length N

Idea:

Break input into first and second half

$$x[n], n = 0, 1, \ldots, N-1 \;\longrightarrow\; x[n] \;\text{ and }\; x[n+N/2], n = 0, 1, \ldots, \frac{N}{2} - 1$$

Break output into even and odd indexed terms, or decimation in frequency (DIF)

$$X[k], k = 0, 1, \ldots, N-1 \;\longrightarrow\; X[2k] \;\text{ and }\; X[2k+1], k = 0, 1, \ldots, \frac{N}{2} - 1$$

Initial computation

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W^{nk}, \qquad k = 0, 1, \ldots, N-1, \quad W = e^{-j\frac{2\pi}{N}}$$

Consider even outputs first, with inputs divided into first and second half

$$\begin{aligned} X[2k] &= \sum_{n=0}^{N/2-1} x[n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[n+N/2]\, W^{(n+N/2)2k} \\ &= \sum_{n=0}^{N/2-1} x[n]\, W^{2nk} + \sum_{n=0}^{N/2-1} x[n+N/2]\, W^{2nk+Nk} \\ &= \sum_{n=0}^{N/2-1} x[n]\, W_{N/2}^{nk} + \sum_{n=0}^{N/2-1} x[n+N/2]\, W_{N/2}^{nk} \\ &= \sum_{n=0}^{N/2-1} (x[n] + x[n+N/2])\, W_{N/2}^{nk} \end{aligned}$$

where we used again the fact that $W^{2nk} = W_{N/2}^{nk}$ and $W^{Nk} = 1$.
In words: We can compute the even terms of the output with the help of a half-size DFT, by summing $x[n]$ and $x[n+N/2]$.

Divide and Conquer for DFT- Analysis of DIF

Consider now odd outputs only, with inputs still divided into first and second half

$$\begin{aligned}
X[2k+1] &= \sum_{n=0}^{N/2-1} x[n]\, W^{n(2k+1)} + \sum_{n=0}^{N/2-1} x[n+N/2]\, W^{(n+N/2)(2k+1)} \\
&= \sum_{n=0}^{N/2-1} x[n]\, W^{n} W^{2nk} + \sum_{n=0}^{N/2-1} x[n+N/2]\, W^{n} W^{2nk} W^{N/2} W^{Nk} \\
&= \sum_{n=0}^{N/2-1} (x[n]\, W^{n}) W_{N/2}^{nk} - \sum_{n=0}^{N/2-1} (x[n+N/2]\, W^{n}) W_{N/2}^{nk} \\
&= \sum_{n=0}^{N/2-1} ((x[n] - x[n+N/2])\, W^{n}) W_{N/2}^{nk}
\end{aligned}$$

where we used the facts that $W^{kN} = 1$ and $W^{N/2} = -1$.
In words: We can compute the odd terms of the output with the help of a half-size DFT, namely by considering $x[n] - x[n+N/2]$ and multiplying this difference with W^n before taking the DFT of size $N/2$.

END OF MODULE 4.9

END OF MODULE 4