Xử lý ảnh & Thị giác máy tính

Chương 2. Các phép biến đổi

- Tích chập (Convolution)
- Biến đổi Fourier
- Biến đổi DCT

2.1 Tích chập

- Tích chập thực hiện đối với 2 tín hiệu (hàm số) **f** và **g**, cho ra 1 tín hiệu (hàm số) thứ 3 **h**

$$(f * g)(i) = h(n)$$

- Tích chập một chiều
- Tích chập hai chiều
- Cross-correlation 2 chiều

2.1.1 Tích chập một chiều

- Công thức tích chập 1 chiều rời rạc

$$(f * g)[n] = \sum_{i=-M}^{M} f[n-i]g[i]$$

Với:

N là độ dài của f

K là độ dài của g

$0 \leq i \leq N - K$

Tích chập một chiều

$$f = \boxed{10 \mid 50 \mid 60 \mid 10 \mid 20 \mid 40 \mid 30}$$

f: tín hiệu đầu vào

$$g = \boxed{1/3 \mid 1/3 \mid 1/3}$$

g: bộ lọc

h: output (valid)

```
[40.          40.          30.          23.33333333 30.          ]
```

* Chú ý: phải lật kernel khi tính chập

Các mode trong tích chập một chiều

- valid : Kích thước output nhỏ hơn kích thước input.
- full : Kích thước output lớn hơn kích thước input.
- same : Kích thước output bằng kích thước input.

Kích thước tín hiệu đầu ra:

- Valid : N−K+1
- Full : N+K-1
- Same : N

Với N, K lần lượt là kích thước của tín hiệu đầu vào và bộ lọc (kernel)

Bài tập

- Tính phép tích chập sau dùng tích chập $(f * g)$ với

f = [1, 3, 5, 6, 8, 9, 10, 11, 12]

g = [1/2, 1/2, 1/2]

Kiểu ‘valid’, ‘same’, ‘full’

Tích chập một chiều trong Python

```
import numpy as np


h = np.convolve(f, g, 'valid') #kieu valid
h = np.convolve(f, g, 'same') #kieu same
h = np.convolve(f, g, 'full') #kieu full
```

2.1.2 Tích chập hai chiều

- Tích chập 2 chiều trong xử lý ảnh $(*)$

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v]F[i-u, j-u]$$

- Cross-correlation $(\otimes)$

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v]F[i+u, j+u]$$

Cross-correlation

$$G[i,j] = \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} F[i+u, j+u] \cdot h[u,v]$$

F: Đầu vào (ma trận 2 chiều)

h: Bộ lọc (kernel, thường nhỏ hơn F, **không lật Flip**)

G: Đầu ra

K, L: Kích thước của bộ lọc

Ví dụ

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

$\otimes$

1	1	0
1	0	1
0	1	0

$=$

25	30
45	50

Tích chập

$$G[i,j] = \sum_{u=0}^{K-1} \sum_{v=0}^{L-1} F[i-u, j-v] \cdot h[u,v]$$

F: Đầu vào (ma trận 2 chiều)

h: Bộ lọc (kernel, thường nhỏ hơn F, cần **lật Flip** trước khi tích chập)

G: Đầu ra

K, L: Kích thước của bộ lọc

Tích chập

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

$*$

1	1	0
1	0	1
0	1	0

$=$

35	40
55	60

0	1	0
1	0	1
0	1	1

Phải "lật" (flip) kernel trước khi chập

```
import scipy as sp
h = sp.signal.convolve2d(f,g,'valid')
```

Tính chất tích chập

- Tính giao hoán

$$f(x,y) * g(x,y) = g(x,y) * f(x,y)$$

- Tính kết hợp

$$[f(x,y) * g(x,y)] * h(x,y) = f(x,y) * [g(x,y) * h(x,y)]$$

- Tính phân phối

$$f(x,y) * [g(x,y) + h(x,y)] = [f(x,y) * g(x,y)] + [f(x,y) * h(x,y)]$$

Tích chập trong xử lý ảnh (2D conv)

Input * Bộ lọc = Output

kernel: ma trận nhỏ kích thước nxn, dùng để biến đổi giá trị pixel ban đầu của ma trận input

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

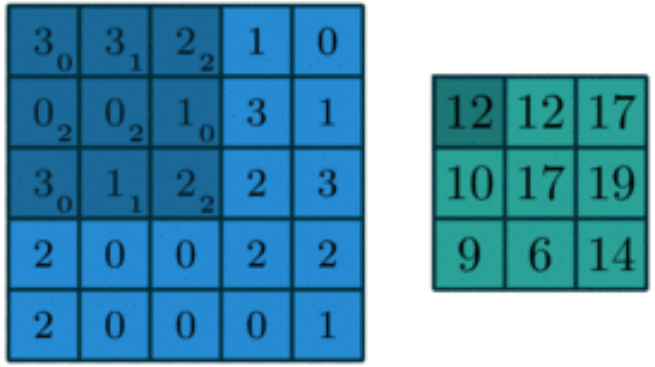
*

0	1	2
2	2	0
0	1	2

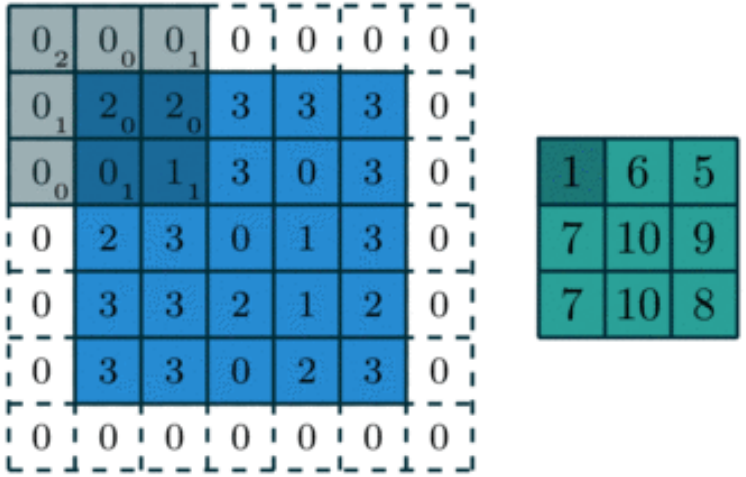
=

Padding

- Không có padding



- Có padding (zero)
  - Stride = 2



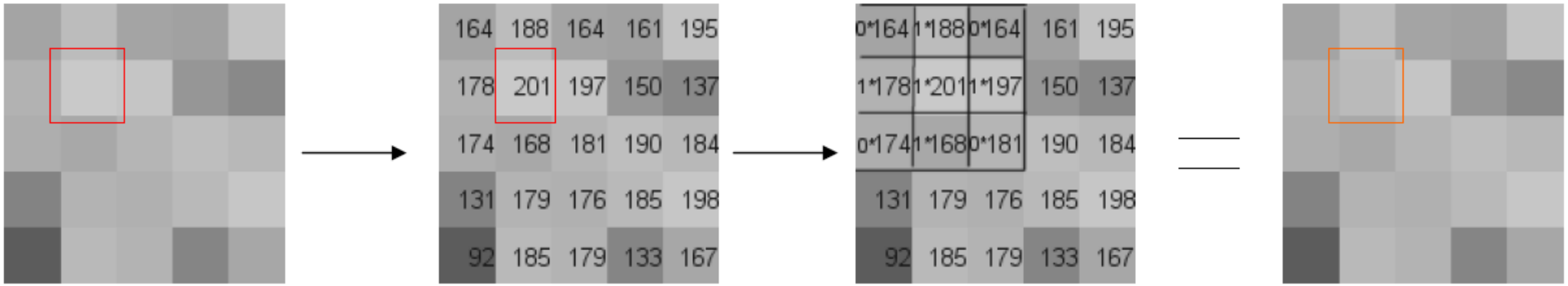
Các mode

- full (kích thước output lớn hơn kích thước input)
- same (kích thước output bằng kích thước input)
- valid (kích thước output nhỏ hơn kích thước input)

Python:

```
import scipy as sp
h = sp.signal.correlate2d(f,g,’valid’)
h = sp.signal.convolve2d(f,g,’same')
```

2.1.3 Tích chập trong xử lý ảnh



Original image

Image with color values placed over it

Image with 3x3 kernel placed over it

Output image

164	188	164
178	201	197
174	168	181

×

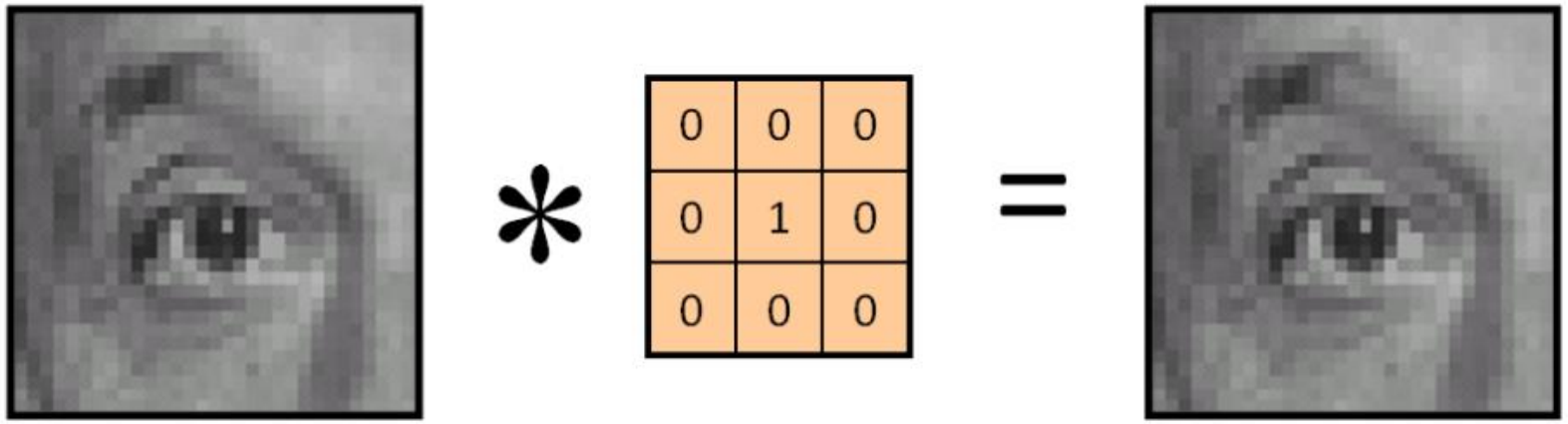
0	1	0
1	1	1
0	1	0

Divided by the sum of the kernel

932\5 = new pixel color

Tích chập trong xử lý ảnh

Ma trận ảnh ∗ Kernel (filter)

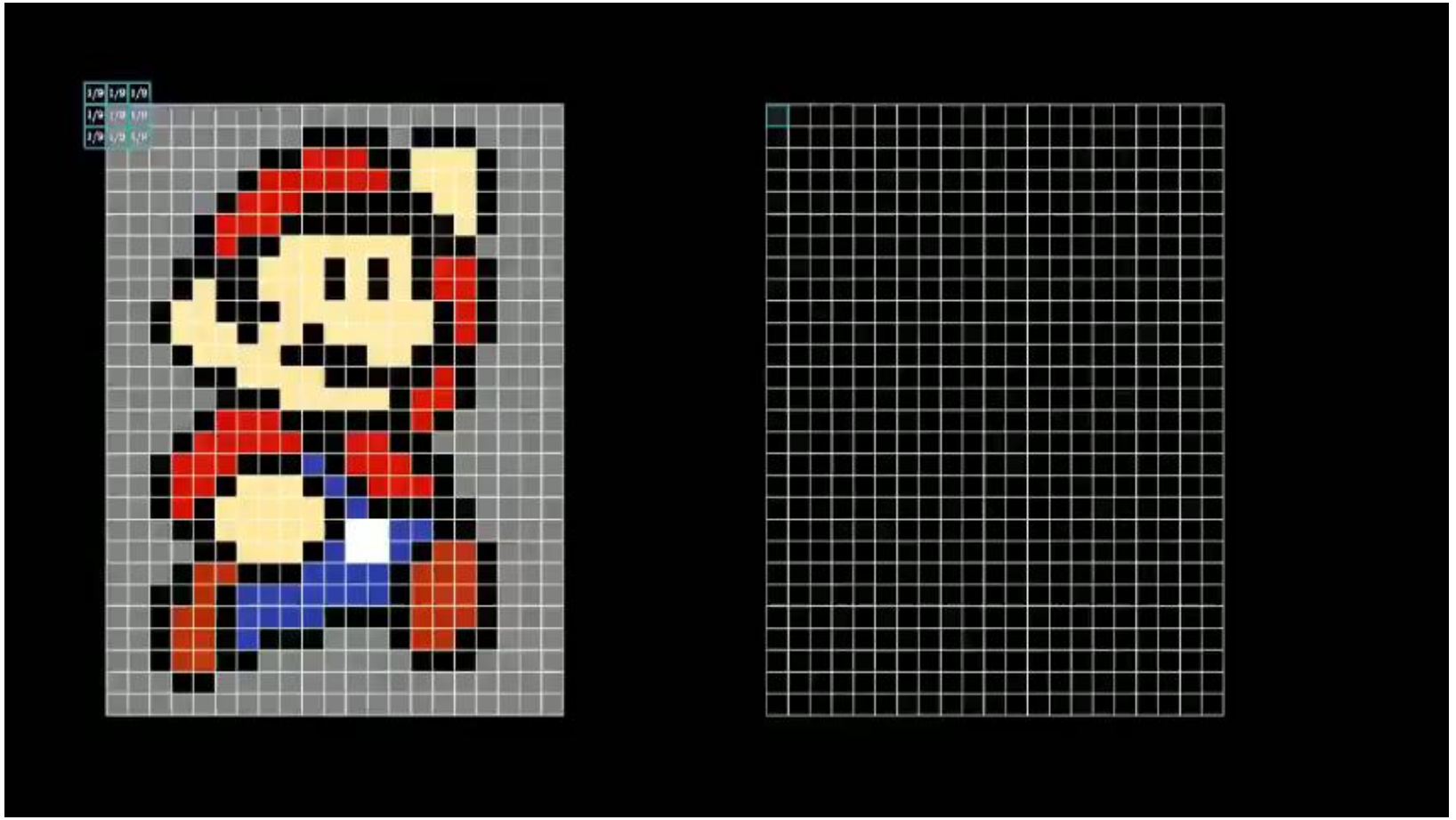


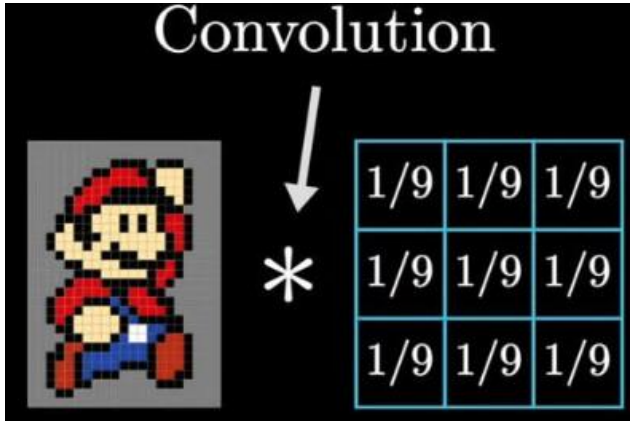
Original

Identical image

*Chú ý dữ liệu là dạng double
hay unit8

Làm mờ (blur)





2.1.4 Một số ứng dụng tích chập

Một số ứng dụng

- ❑ Làm mờ (bluring)
- ❑ Làm sắc nét (sharpen)
- ❑ Nhận biết vùng biên (edge detection)

* Chú ý kiểu dữ liệu ma trận là unit8 hay **double** và chuẩn hóa kernel trước khi dùng hàm chập

Ôn tập thực hành XLA cơ bản trên Python

Các hàm xử lý ảnh

- I = imread("logo_vaa.jpg") # đọc file ảnh tên logo_vaa.jpg, lưu vào biến I
- imshow("VAA", I) # show hình ảnh của I
- imwrite("logo_save .jpg", I) # lưu ảnh I thành ảnh "logo_save .jpg"
- cvtColor(I, cv2.COLOR_BGR2GRAY) # chuyển ảnh màu I thành ảnh xám

Lập trình trên Pycharm

- Tạo project, tạo file .py trong project
- Import thư viện (ví dụ: import numpy as np)
- Comment/uncomment trên PyCharm bằng ctrl + / và ctrl + /, hoặc sau dấu #

$$* \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad =$$

$$* \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad =$$

cameramen.tif

Original

Sharp

-1	-1	-1
-1	9	-1
-1	-1	-1

Intensified sharper image

Nhiều các bộ lọc thiết kế đối xứng

Blur

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Edge detection

0	-1	0
-1	5	-1
0	-1	0

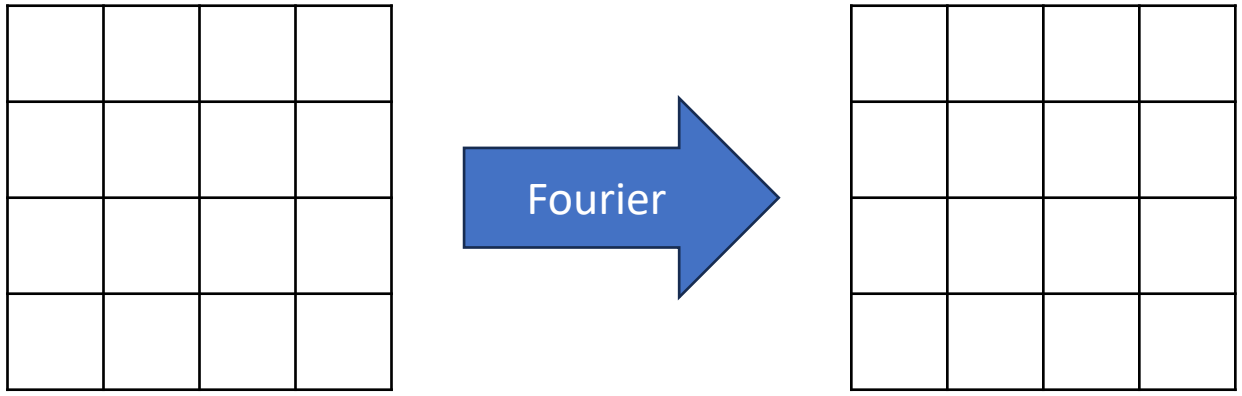
- Tích chập được áp dụng chủ yếu để cải thiện ảnh trên miền không gian, tức tính toán trực tiếp trên ma trận
  - Từ ảnh ban đầu, dùng một số bộ lọc được thiết kế để tích chập với ảnh gốc theo yêu cầu sử dụng

- Trên miền tần số, cần phải có một công cụ để phân tích → Biến đổi Fourier

2.2 Biến đổi Fourier

Cải thiện ảnh trong miền tần số

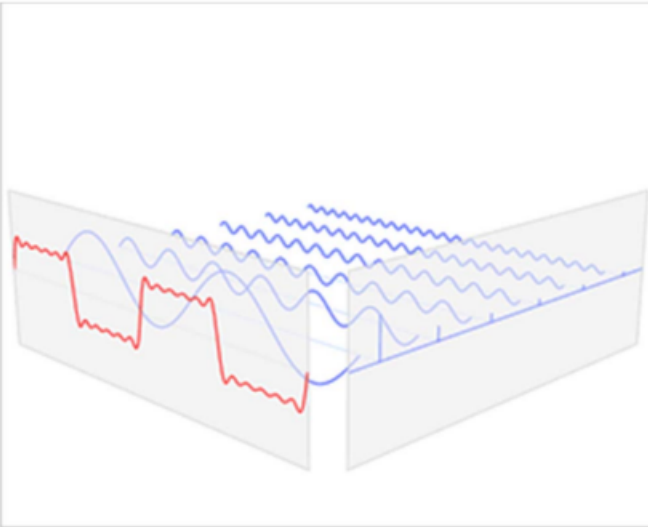
- Khái niệm biến đổi Fourier – biến đổi Fourier
- Lọc trong miền tần số
  - Lọc thông thấp
  - Lọc thông cao
  - Lý tưởng (ideal)
  - Gaussian
  - Butterworth



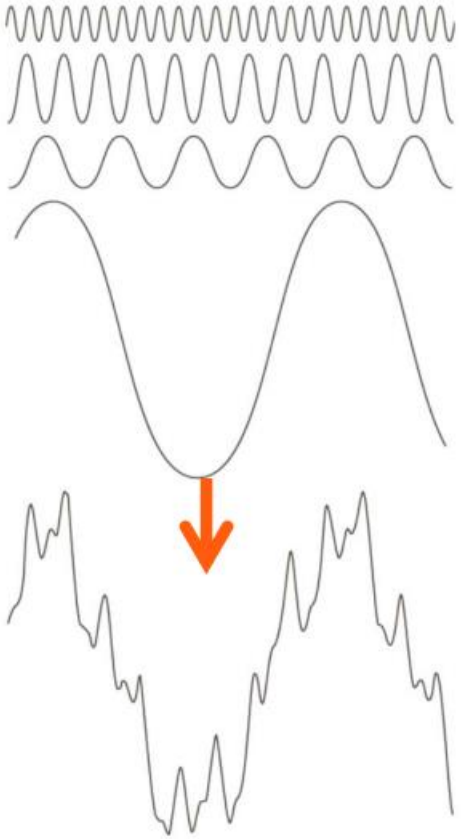
2.2 Biến đổi Fourier

Trong xử lý tín hiệu: biến đổi tín hiệu từ miền thời gian sang miền tần số

Fourier Series: f(t) →F(s)



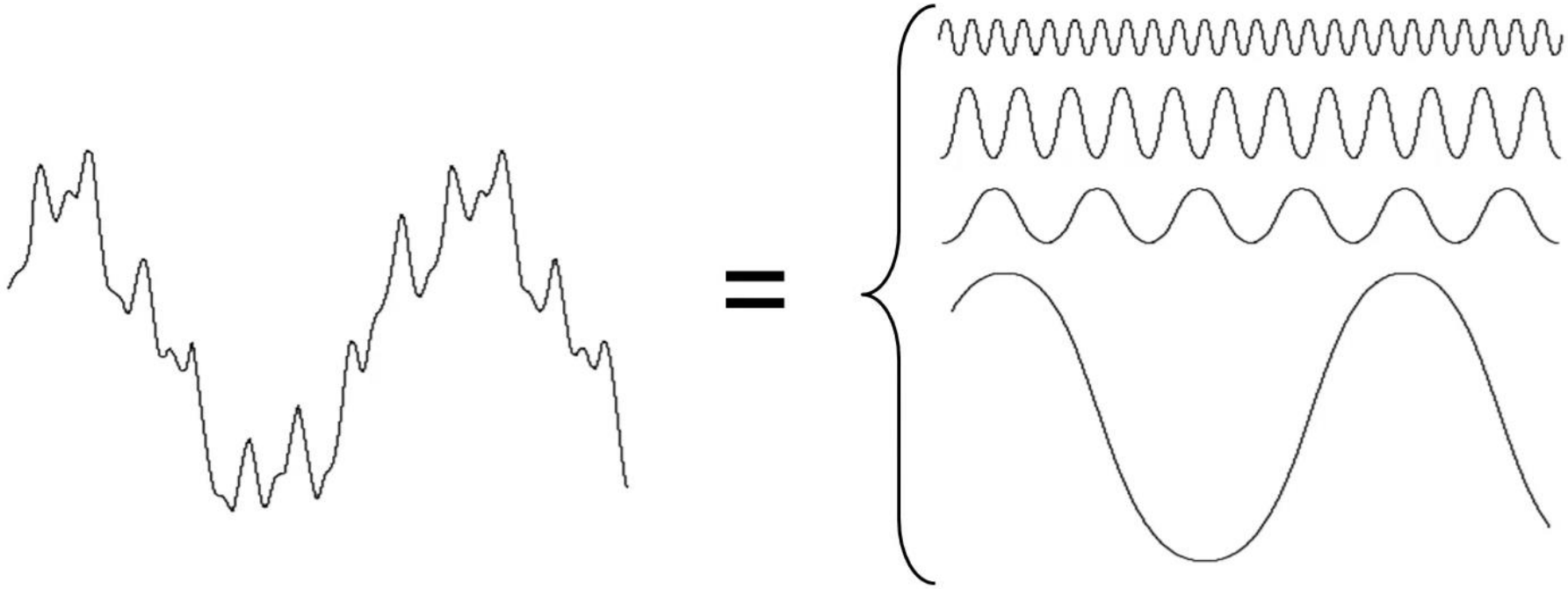
Function $s(x)$ (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. Their summation is called a Fourier series. The Fourier transform, $S(f)$ (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.



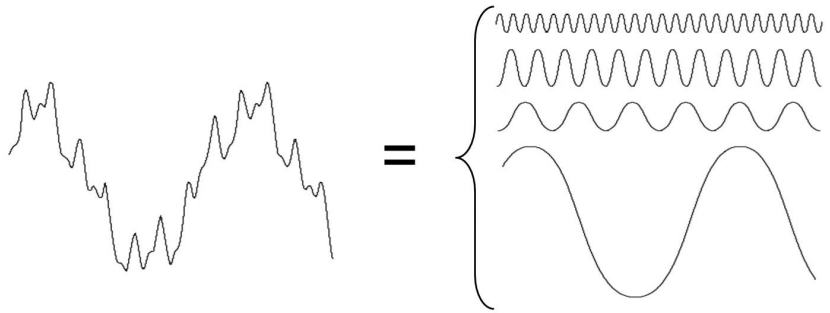
source: http://en.wikipedia.org/wiki/Fourier_series

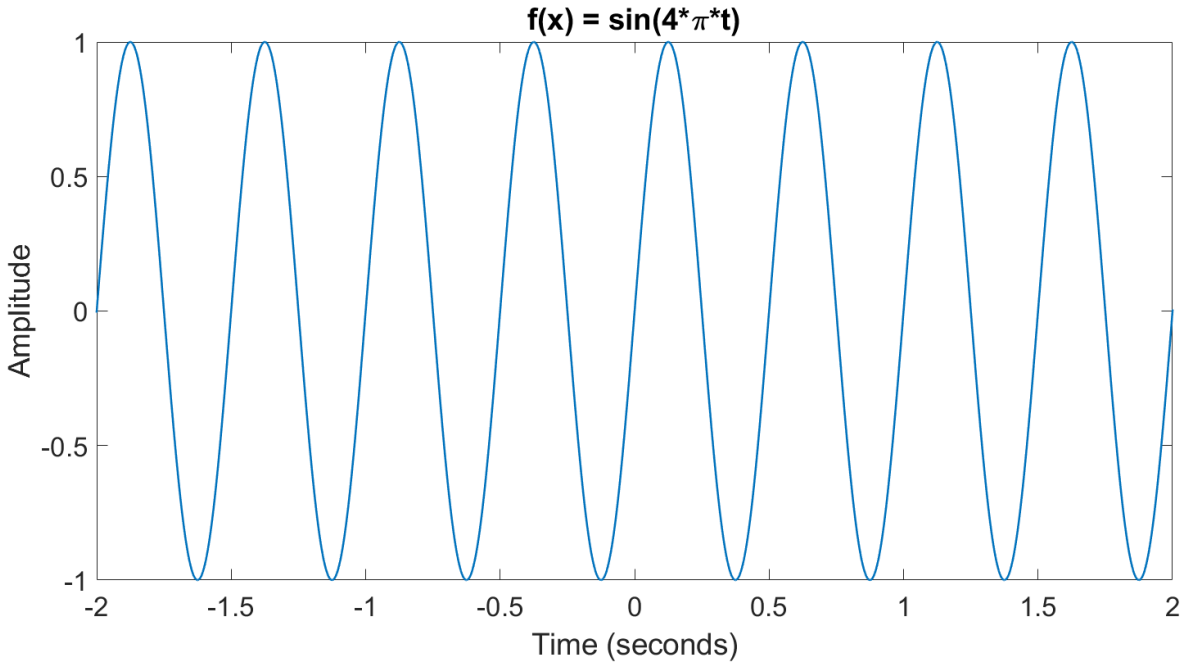
Biến đổi Fourier

- Một hàm bất kỳ lặp lại có tính chu kỳ có thể được biểu diễn bằng tổng những hàm sin và cos ở các tần số khác nhau

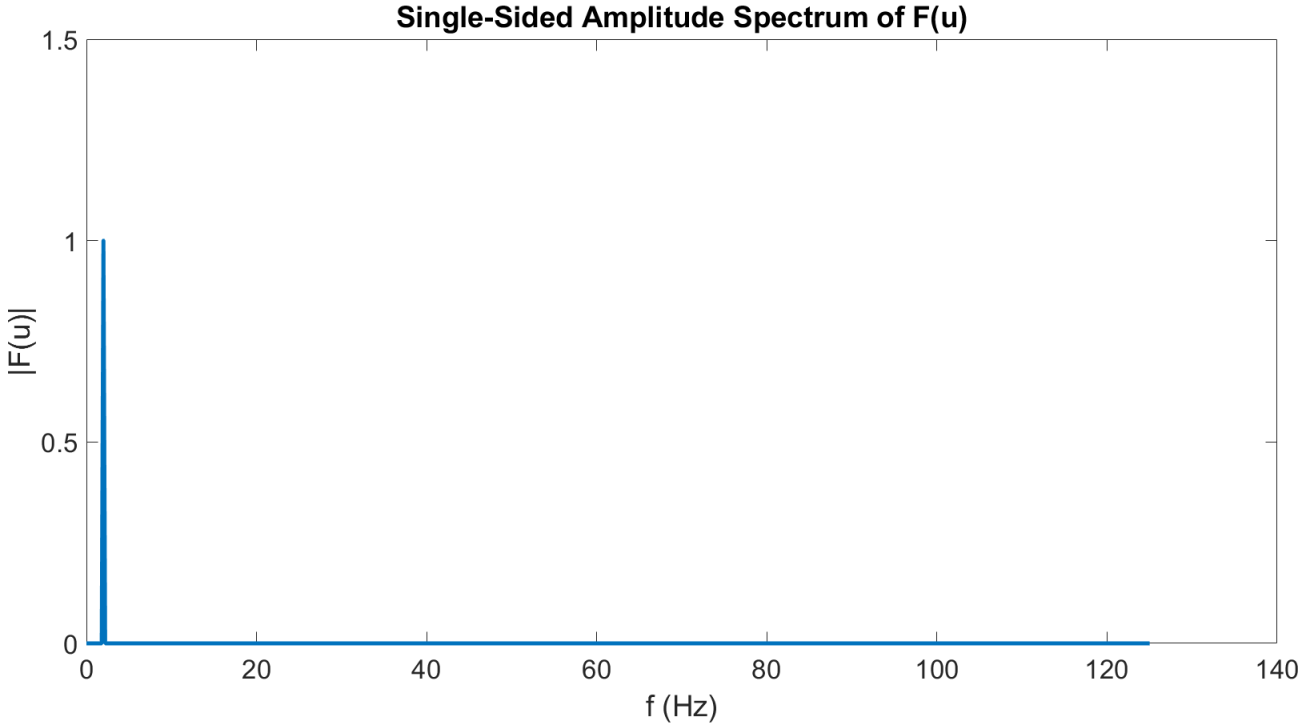


Ví dụ





$$f(x) = \sin(4\pi t)$$

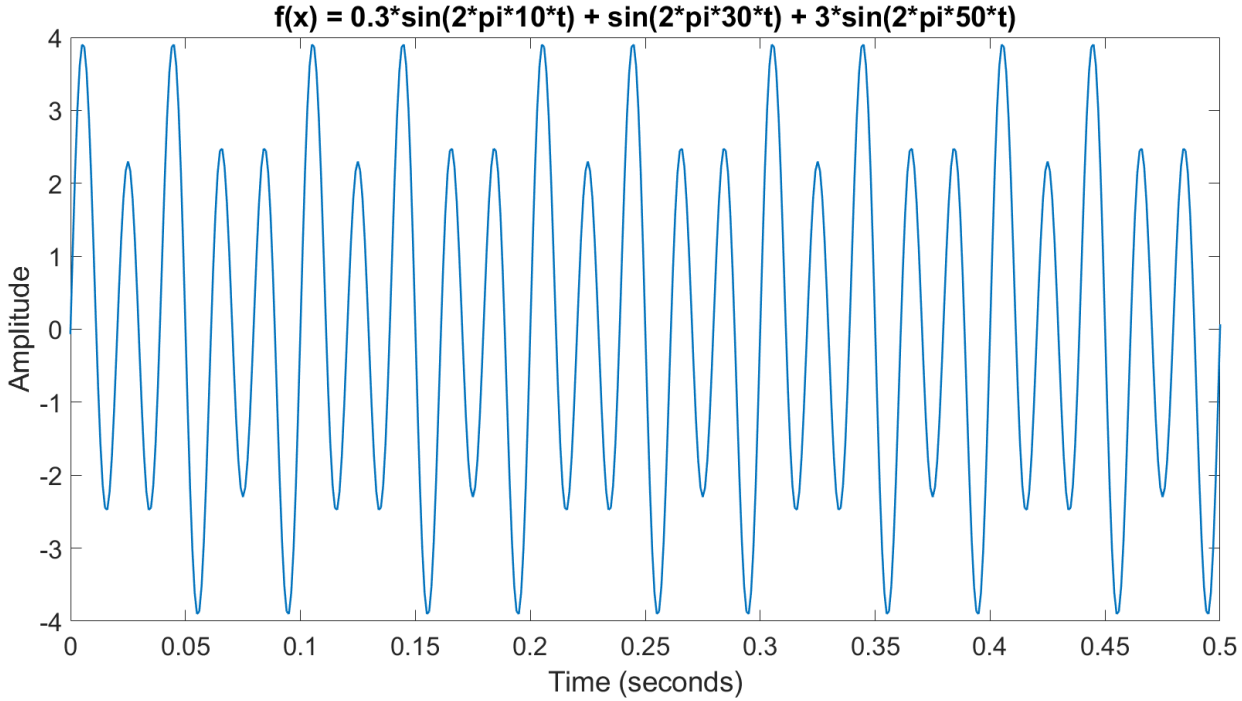


$$F(u)$$

Tín hiệu gốc f(x) tập trung tại tần số f = 2Hz

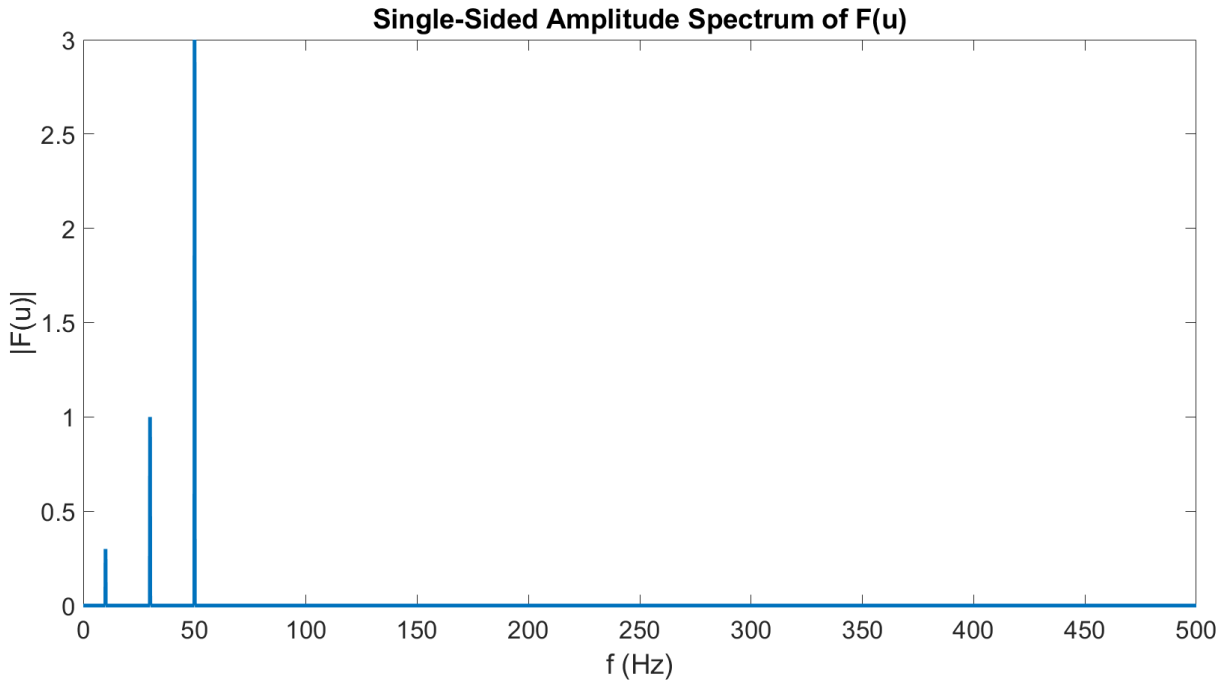
Ví dụ

f(x)



$f(x) = 0.3\sin(20\pi t) + \sin(60\pi t) + 3\sin(100\pi t)$

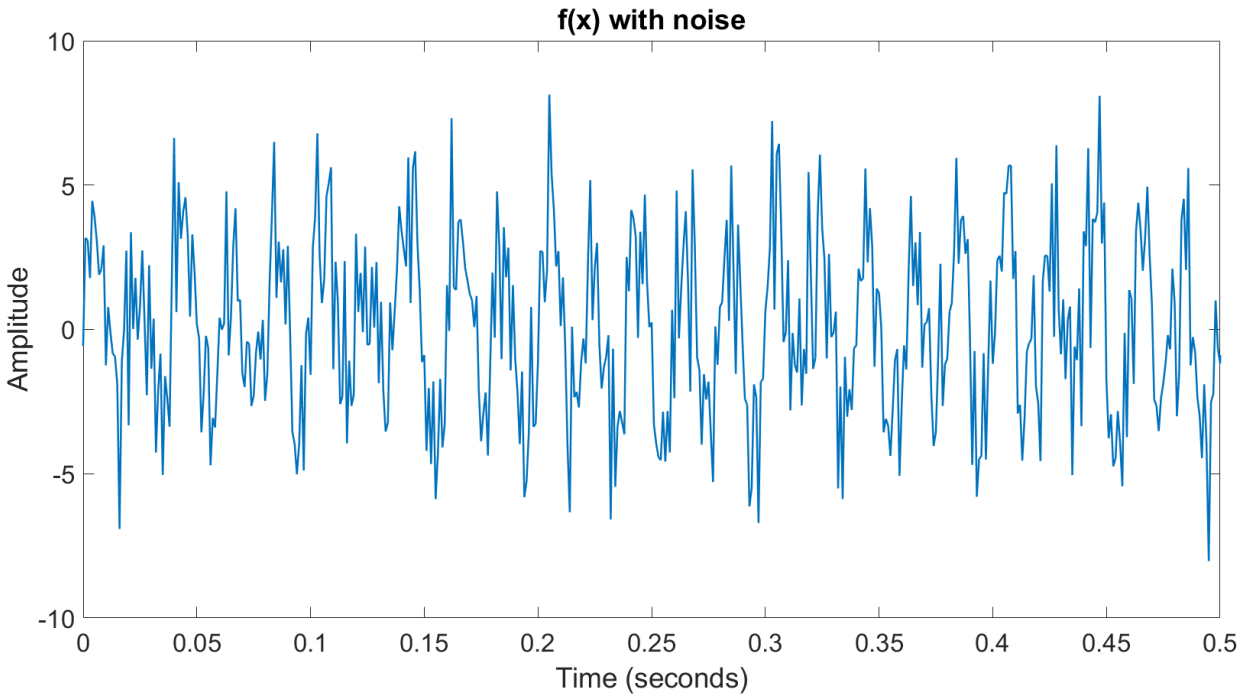
F(u)



Tín hiệu gốc f(x) tập trung tại các tần số
f = 10Hz, 30Hz, 50Hz

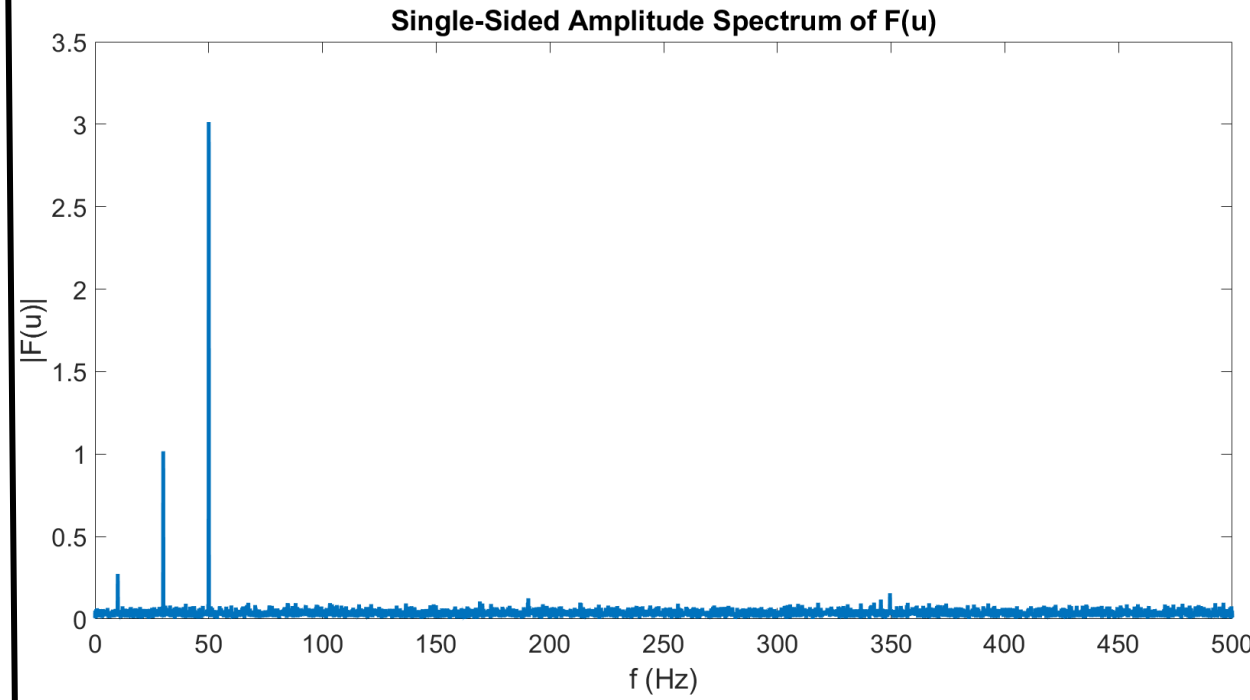
Ví dụ

f(x)



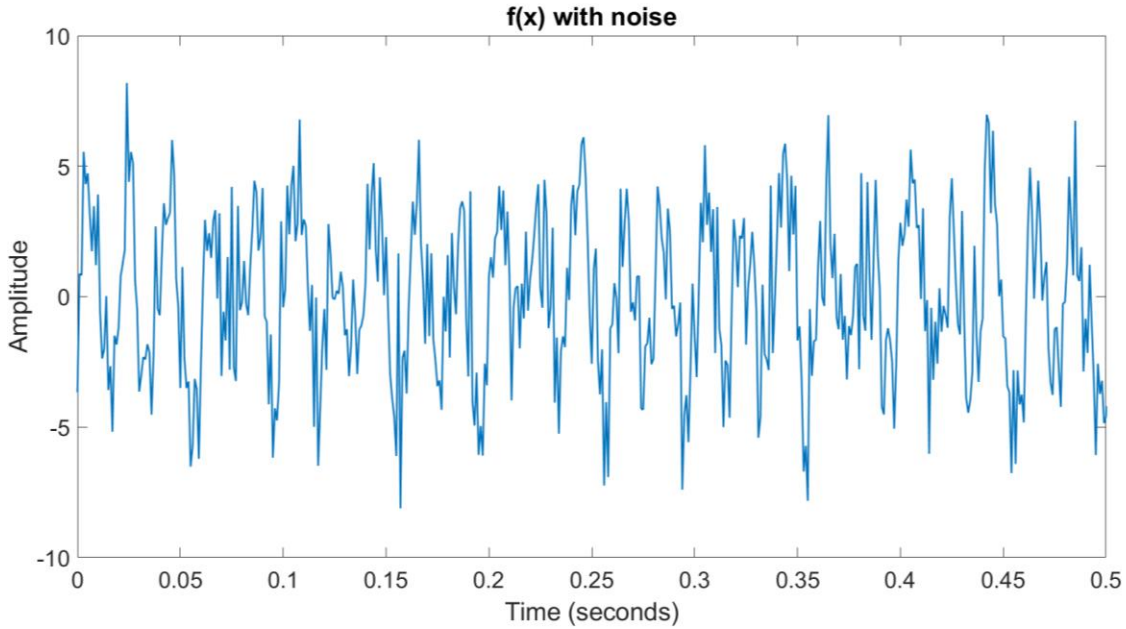
$f(x) = 0.3\sin(20\pi t) + \sin(60\pi t) + 3\sin(100\pi t)$ + **Noise**

F(u)

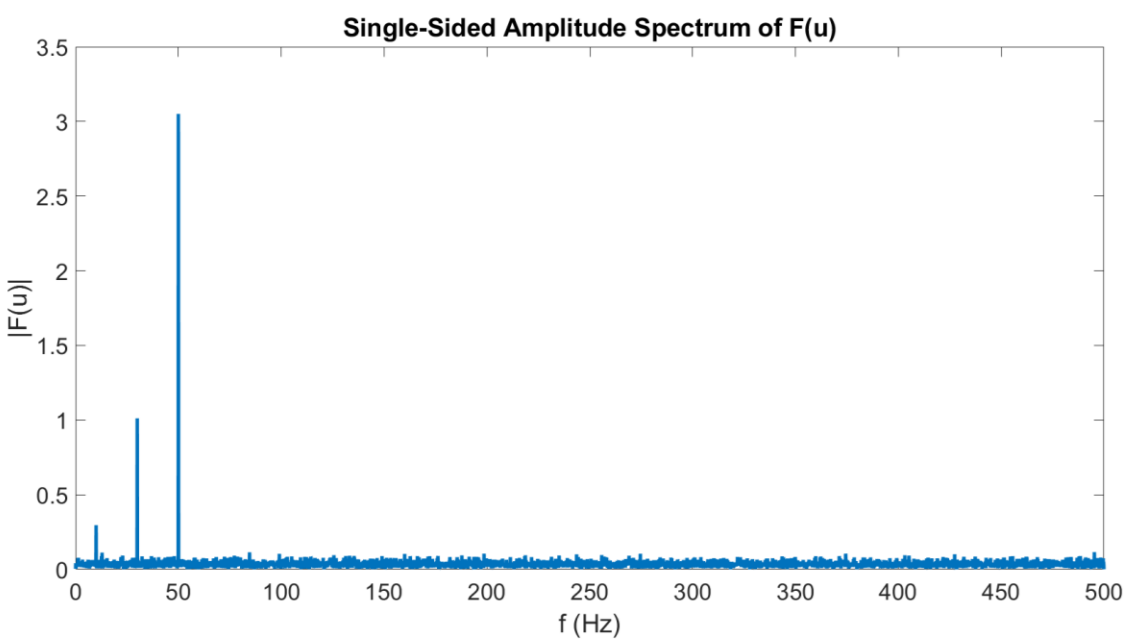


Tín hiệu gốc f(x) tập trung tại các tần số

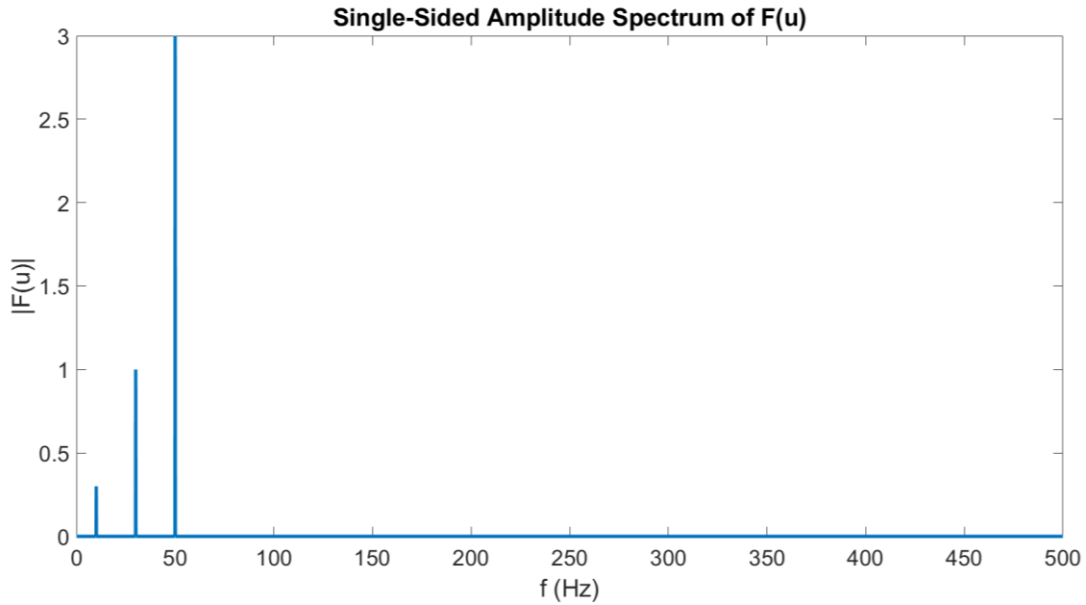
f = 10Hz, 30Hz, 50Hz



DFT

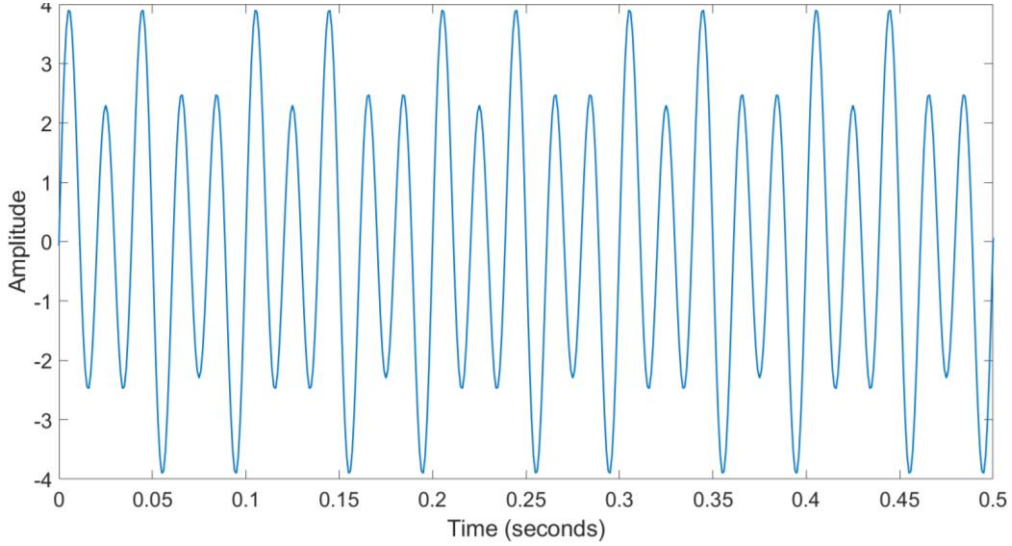


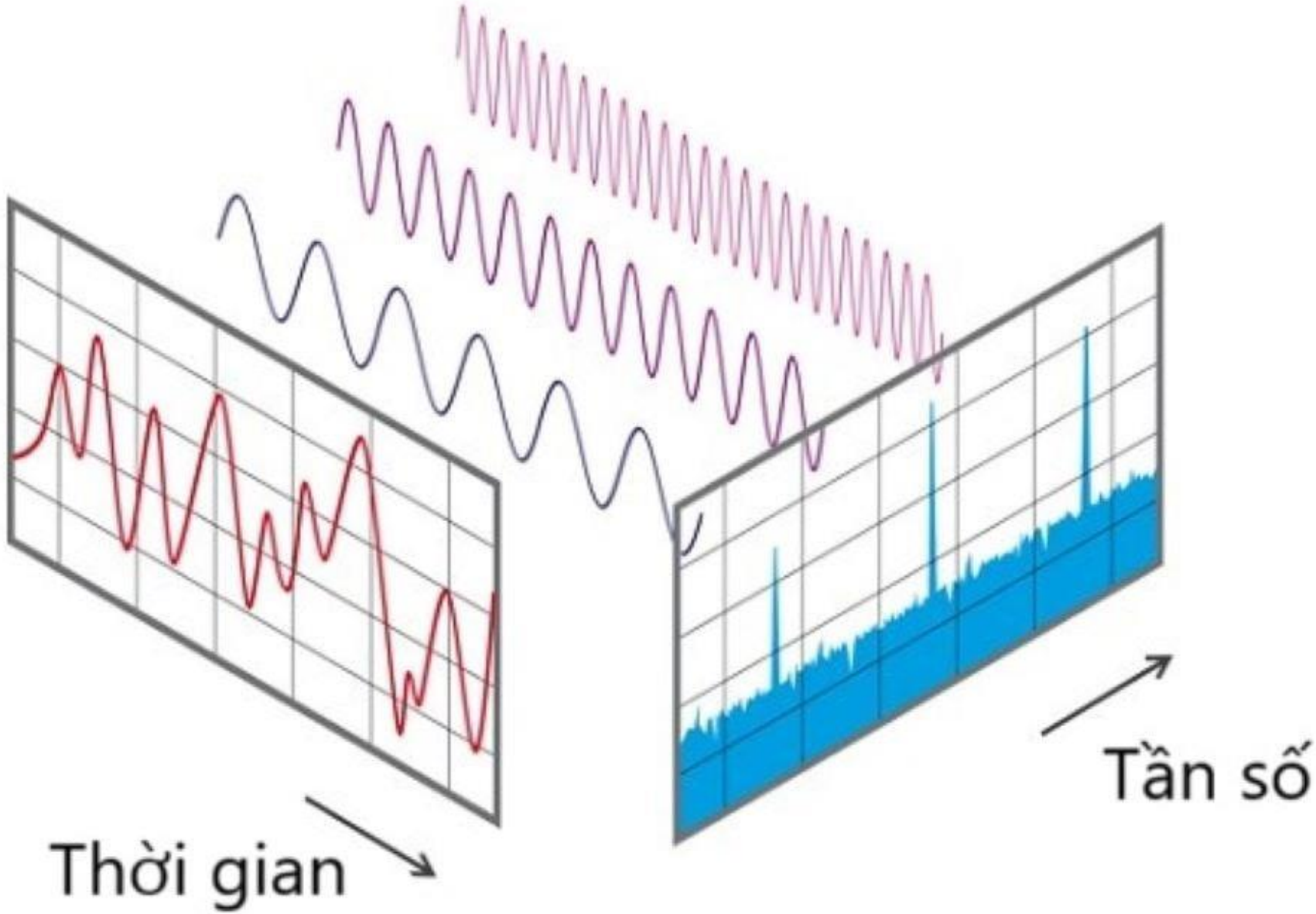
Filter



DFT ngược

Tín hiệu được cải thiện




Thời gian
Tần số

Số phức

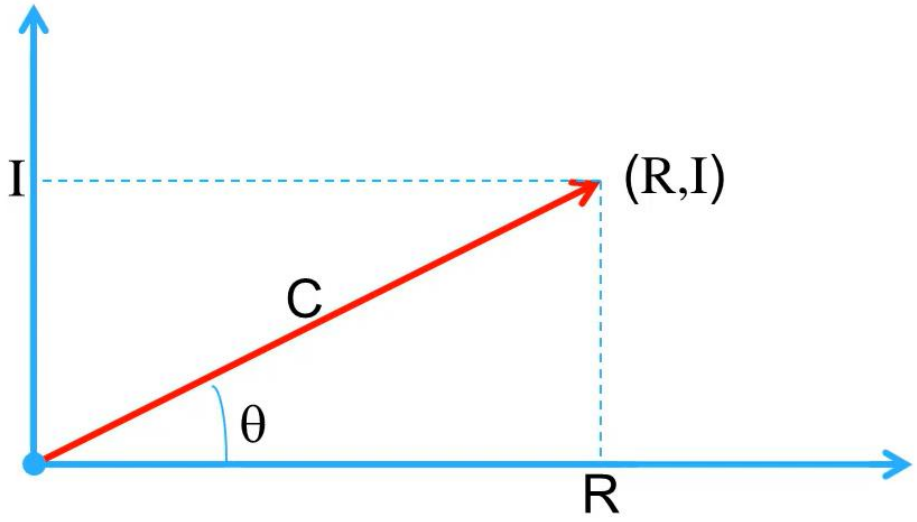
- Số phức được định nghĩa:

$$c = a + jb$$

- Trong đó:
  - a là phần thực
  - b là phần ảo
  - $j^2 = -1$

Tọa độ cực số phức

- Biểu diễn trong tọa độ cực : $C = |C| \times (\cos\theta + j\sin\theta)$
  - $\theta = \arctan(\frac{I}{R})$
  - $|C| = \sqrt{R^2 + I^2}$: bán kính



- $e^{j\theta} = cos\theta + jsin\theta$
- $a + jb$ có thể viết dạng $|R|e^{j\theta}$
- Trong biến đổi Fourier Số phức được biểu diễn biên độ (R) và phổ (θ)

2.2 Biến đổi Fourier rời rạc

- Biến đổi Fourier rời rạc (DFT) 1 chiều

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad (1)$$

- Công thức Euler

$$e^{j\theta} = cos\theta + jsin\theta$$

- Thế công thức Euler vào (1)

$$F(u) = \sum_{x=0}^{M-1} f(x)[\frac{cos2\pi ux}{M} - j\frac{sin2\pi ux}{M}]$$

Trong đó:

- **F(u)**: là giá trị biến đổi Fourier tại chỉ số tần số k
- **f(x)**: là giá trị của tín hiệu tại mẫu thứ x
- $\mathbf{e^{-j2\pi ux/M}}$: là hàm mũ phức
- **M**: số lượng mẫu
- **u =0, 1…M-1**: là chỉ số tần số rời rạc

Biên độ và phổ

- Do hàm trong miền tần số là số phức, ta có thể viết lại trong tọa độ cực

$$F(u) = |F(u)|e^{-j\phi(u)}$$

Với $|F(u)| = [R^2(u) + I^2(u)]^{1/2}$ được gọi là biên độ

$\phi(u) = tan^{-1}\left[\frac{I(u)}{R(u)}\right]$ được gọi là phổ

- Năng lượng phổ: $P(u) = |F(u)|^2$

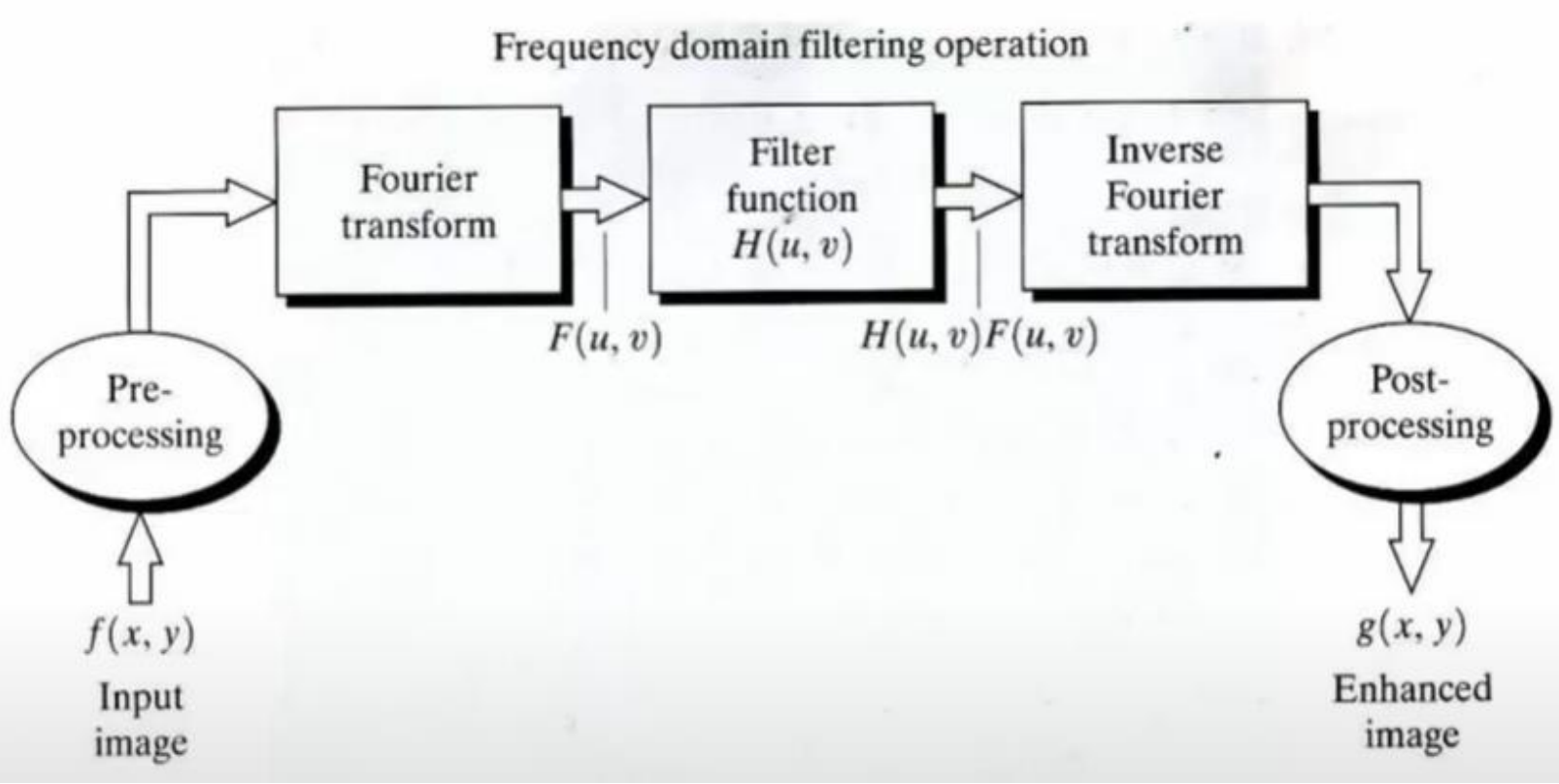
Ứng dụng Fourier

- Thực tế các tín hiệu thu được đều có nhiễu, do đó khi phân tích trong miền tần số chúng ta có thể lọc để loại bỏ các thành phần này để cải thiện chất lượng tín hiệu

- Các bộ lọc thông dụng như lọc thông thấp (low-pass filter), lọc thông cao (high-pass filter)

Biến đổi Fourier trong xử lý ảnh

- DFT trong xử lý ảnh là DFT 2 chiều



Ứng dụng:

- Rút trích đặt trưng trong miền tần số
- Lọc nhiễu
- Xác định tần số của một bức ảnh

Biến đổi Fourier trong xử lý ảnh

- Công thức biến đổi Fourier rời rạc 2 chiều

$$G(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x,y)\, e^{-j\frac{2\pi(ux)}{M}} e^{-j\frac{2\pi(vy)}{N}}$$

Trong đó:

- g(x,y) là ảnh gốc trong miền không gian, kích thước MxN
- G(u,v) được gọi là ảnh Fourier trong miền tần số
- u, v là các trục tần số, tương ứng với trục x,y
- $e^{-j\frac{2\pi(ux)}{M}} e^{-j\frac{2\pi(vy)}{N}}$: là hàm mũ phức biểu diễn các thành phần tần s

Biến đổi Fourier trong xử lý ảnh

- Công thức biến đổi Fourier ngược

$$g(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} G(u,v)\, e^{j\frac{2\pi(ux)}{M}} e^{j\frac{2\pi(vy)}{N}}$$

Trong đó u = 0,1,…,M-1 và v = 0,1,…,N-1 được gọi là các trục tần số, g(x,y) được gọi là ảnh sau khi biến đổi Fourier ngược

Công thức biến đổi Fourier cho ảnh

- Nếu G(u,v) là ảnh Fourier của một ảnh g(x,y) có M dòng, N cột:

$$G = U^T \times g \times V \qquad (1)$$

Với $U(x,u) = e^{-\frac{j2\pi xu}{M}}; x,u = 0:M-1$

$\mathrm{V}(y,v) = e^{-\frac{j2\pi yv}{N}}; y,v = 0:N-1$

Công thức biển đổi Fourier cho ảnh

- Nếu $U(u,x) = U(x,u) \rightarrow U = U^T$, (1) có thể viết lại

$$G = U \times g \times V$$

- Biển đổi Fourier ngược:

$$g = U^{-1} \times G \times V^{-1}$$

Ví dụ biến đổi Fourier

- Cho một ảnh 4x4 g(x,y). Tìm ảnh Fourier của ảnh trên

$$g(x,y)=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

$$U(x,u)=e^{-\frac{j2\pi xu}{M}};x,u=0:M-1$$

$$U=\begin{bmatrix} e^{\frac{-j\times2\times\pi\times0\times0}{4}} & e^{\frac{-j\times2\times\pi\times0\times1}{4}} & e^{\frac{-j\times2\times\pi\times0\times2}{4}} & e^{\frac{-j\times2\times\pi\times0\times3}{4}}\\ e^{\frac{-j\times2\times\pi\times1\times0}{4}} & e^{\frac{-j\times2\times\pi\times1\times1}{4}} & e^{\frac{-j\times2\times\pi\times1\times2}{4}} & e^{\frac{-j\times2\times\pi\times1\times3}{4}}\\ e^{\frac{-j\times2\times\pi\times2\times0}{4}} & e^{\frac{-j\times2\times\pi\times2\times1}{4}} & e^{\frac{-j\times2\times\pi\times2\times2}{4}} & e^{\frac{-j\times2\times\pi\times2\times3}{4}}\\ e^{\frac{-j\times2\times\pi\times3\times0}{4}} & e^{\frac{-j\times2\times\pi\times3\times1}{4}} & e^{\frac{-j\times2\times\pi\times3\times2}{4}} & e^{\frac{-j\times2\times\pi\times3\times3}{4}}\end{bmatrix}$$

$$U=\begin{bmatrix} e^{\frac{-j2\pi 0x0}{4}} & e^{\frac{-j2\pi 0x1}{4}} & e^{\frac{-j2\pi 0x2}{4}} & e^{\frac{-j2\pi 0x3}{4}} \\ e^{\frac{-j2\pi 1x0}{4}} & e^{\frac{-j2\pi 1x1}{4}} & e^{\frac{-j2\pi 1x2}{4}} & e^{\frac{-j2\pi 1x3}{4}} \\ e^{\frac{-j2\pi 2x0}{4}} & e^{\frac{-j2\pi 2x1}{4}} & e^{\frac{-j2\pi 2x2}{4}} & e^{\frac{-j2\pi 2x3}{4}} \\ e^{\frac{-j2\pi 3x0}{4}} & e^{\frac{-j2\pi 3x1}{4}} & e^{\frac{-j2\pi 3x2}{4}} & e^{\frac{-j2\pi 3x3}{4}} \end{bmatrix}$$

$$\Rightarrow U=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

- Tương tự cho ma trận V:

$$\mathrm{V}(y,v)=e^{-\frac{j2\pi yv}{N}};y,v=0:N-1$$

$$V=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$G = U * g * V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$G = \begin{bmatrix} 4 & -2-j2 & 0 & -2+j2 \\ -2-j2 & j2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2+2j & 2 & 0 & -j2 \end{bmatrix}$$

= Real + j*Imp

Tính Fourier của ảnh sau

$$g = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Tính Fourier của ảnh sau

$$g = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Biến đổi Fourier trong xử lý ảnh

- Sau khi biến đổi ảnh Fourier sẽ có 2 thành phần thực và ảo. Một pixel sẽ có 2 thành phần
  - Phần thực
  - Phần ảo

- Có thể biểu diễn DFT của 1 ảnh dưới hàm log:

$$d(u,v) = \log(1 + |G(u,v)|)$$

Biên độ và phổ

$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$ được gọi là biên độ

$\phi(u,v) = tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$ được gọi là pha

- Năng lượng phổ: $P(u,v) = |F(u,v)|^2$
- Thông thường biểu diễn biên độ sử dụng hàm log: $\log(1+ |F(u,v)|)$