Linear Systems: Discrete case & 2D

Required Reading: 3.4

Linear systems-review

Discrete case & 2D

 2D impulse function
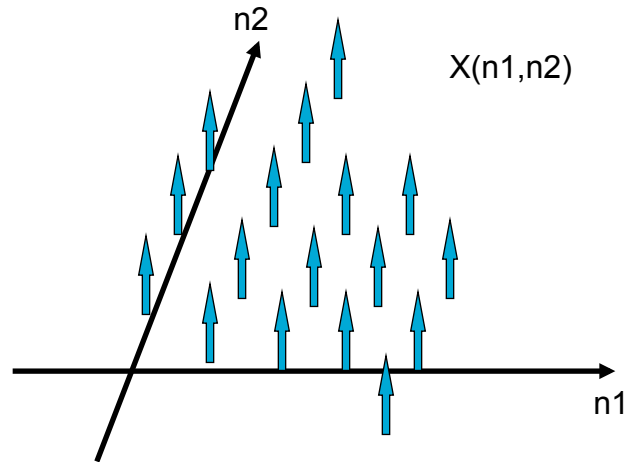
 Line function

 Step function

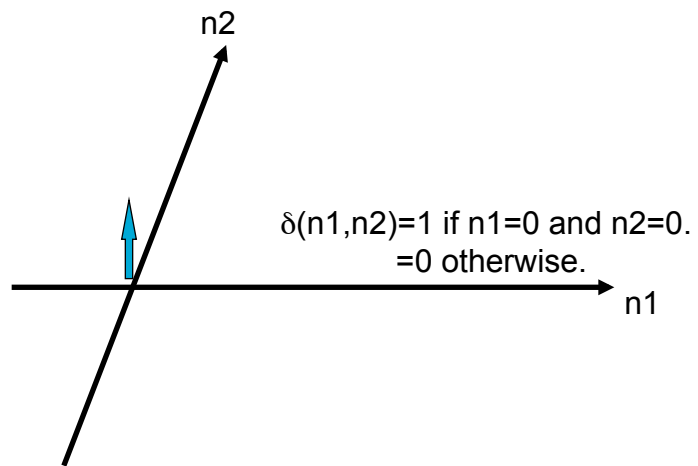 Linear systems and Shift invariance

 Impulse Response of LSI Systems

 2-D Convolution

2-D Systems

X(n1,n2)

n2

n1

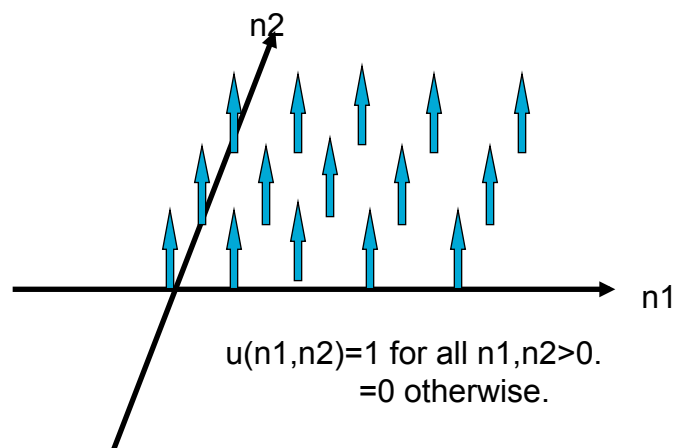# Impulse Function(Kronecker Delta)

n2

δ(n1,n2)=1 if n1=0 and n2=0.
=0 otherwise.

n1

Line Impulse

n2

$\delta_T(n1,n2)=1$ for all $n1=0$.
    $=0$ otherwise.

n1
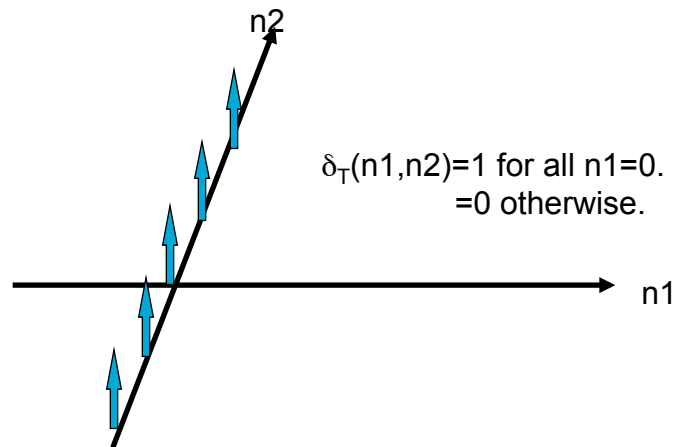
Unit Step Function

n2

n1

$u(n1,n2)=1$ for all $n1,n2>0$.
    $=0$ otherwise.

" System"

An input-output relationship is called a system if there is a unique output for any given input.

$Y(n1,n2) = T[X(n1,n2)]$

Linear Systems

The linearity of a system T is defined as

Linearity:

$T[a\, X_1(n1,n2) + b\, X_2(n1,n2)] = a\, Y_1 + b\, Y_2$

(i.e., principle of superposition holds).

Are these linear?

(a) $y(m,n) = x(m,n)\, g(m,n)$

(b) $y(m,n) = [x(m,n)]^2$

Linear Shift Invariant Systems

Shift Invariance:

    T[X(m-k, n-l)] = Y(m-k, n-l) where

    Y(m,n) = T[X(m,n)].

A LSI system is completely characterized by its response to the impulse function δ(m,n).

Convolution

Let $h(n_1,n_2) = T[\delta(n_1,n_2)]$; $y(n_1,n_2) = T[x(n_1,n_2)]$; then
$h(n_1- k_1 ,n_2 - k_2) = T[\ \delta(n_1- k_1 ,n_2 - k_2)\ ]$, and

$$
\begin{aligned}
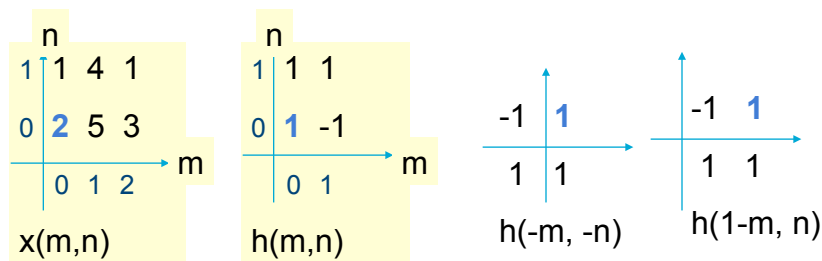y(n_1, n_2) &= T\left[\sum_{k_1=-\infty}^{\infty} \sum_{k_1=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2) \right] &\quad(1)\\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_1=-\infty}^{\infty} x(k_1, k_2)T[\delta(n_1 - k_1, n_2 - k_2)] &\quad(2)\\
&= \sum_{k_1=-\infty}^{\infty} \sum_{k_1=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2) &\quad(3)
\end{aligned}
$$

$$
y(n_1, n_2) = h(n_1, n_2) * x(n_1, n_2)
$$

Convolution: example

x(m,n) table:
```
n
1 │ 1  4  1
0 │ 2  5  3
  └──────────── m
    0  1  2
```

h(m,n) table:
```
n
1 │ 1   1
0 │ 1  -1
  └──────── m
    0   1
```

h(-m, -n):
```
-1 │ 1
───┼───
 1 │ 1
```

h(1-m, n):
```
-1   1
───────
 1   1
```

$y(1,0) = \Sigma_{k,l}\, x(k,l)h(1-k, -l) =$

```
0 │ 0   0   0
──┼──────────
0 │ -2  5   0   = 3
──┼──────────
0 │ 0   0   0
```

$y(m,n) =$
```
n
  │ 1   5   5   1
  │ 3  10   5   2
  │ 2   3  -2  -3
  └──────────────── m
```

verify!
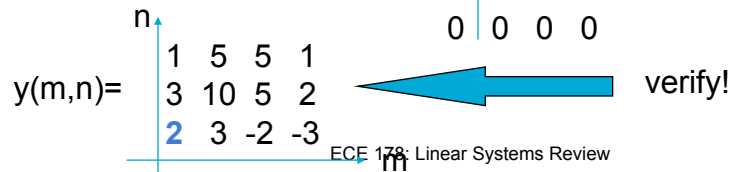
Discrete Convolution in 2D

x(m',n')

h(m',n')

h(m-m',n-n')
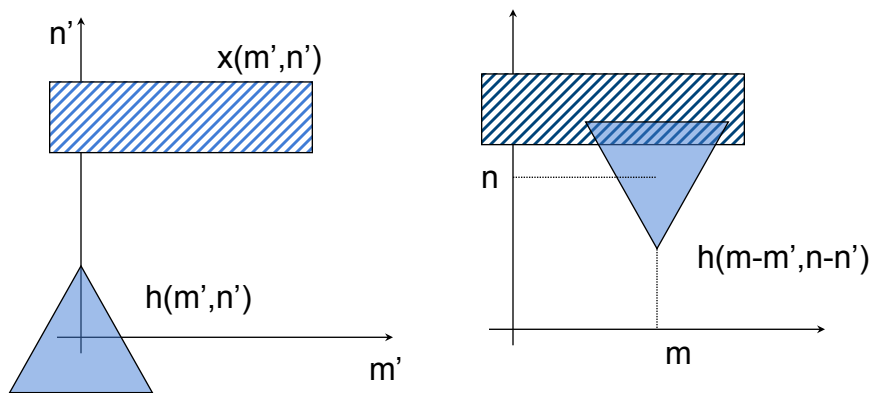
output=sum of the product
of the two in the overlapped region.

Correlation vs Convolution

Image origin

y

Mask

Image $f(x, y)$

x

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

Mask coefficients, showing coordinate arrangement

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Pixels of image section under mask

FIGURE 3.12 The mechanics of linear spatial filtering. The magnified drawing shows a 3×3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.
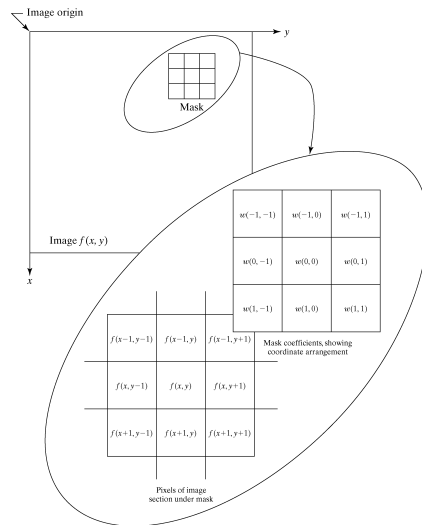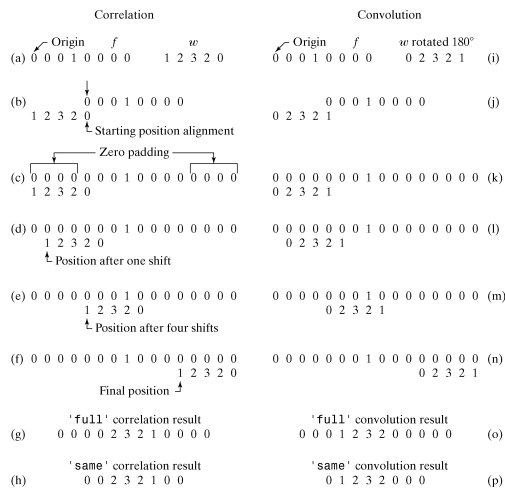
13

1-D example

	Correlation		Convolution	
	Origin f w		Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 3 2 0		0 0 0 1 0 0 0 0 0 2 3 2 1	(i)

(b)   0 0 0 1 0 0 0 0
    1 2 3 2 0
↳ Starting position alignment

          0 0 0 1 0 0 0 0   (j)
    0 2 3 2 1

⎯ Zero padding ⎯

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
   1 2 3 2 0   0 2 3 2 1

(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
    1 2 3 2 0   0 2 3 2 1
  ↳ Position after one shift

(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
        1 2 3 2 0   0 2 3 2 1
    ↳ Position after four shifts

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
           1 2 3 2 0   0 2 3 2 1
    Final position ↳

	'full' correlation result		'full' convolution result	
(g)	0 0 0 0 2 3 2 1 0 0 0 0		0 0 0 1 2 3 2 0 0 0 0 0	(o)

	'same' correlation result		'same' convolution result	
(h)	0 0 2 3 2 1 0 0		0 1 2 3 2 0 0 0	(p)

ECE 178: Linear Systems Review   14

2-D example

Padded f

```
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```
(b)

Origin of $f(x, y)$
```
0 0 0 0 0
0 0 0 0 0       w(x, y)
0 0 1 0 0        1 2 3
0 0 0 0 0        4 5 6
0 0 0 0 0        7 8 9
```
(a)

Initial position for w
```
1 2 3 0 0 0 0 0 0
4 5 6 0 0 0 0 0 0
7 8 9 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```
(c)

'full' correlation result
```
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 9 8 7 0 0 0
0 0 0 6 5 4 0 0 0
0 0 0 3 2 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```
(d)

'same' correlation result
```
0 0 0 0 0
0 9 8 7 0
0 6 5 4 0
0 3 2 1 0
0 0 0 0 0
```
(e)

Rotated w
```
9 8 7 0 0 0 0 0 0
6 5 4 0 0 0 0 0 0
3 2 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```
(f)

'full' convolution result
```
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 1 2 3 0 0 0
0 0 0 4 5 6 0 0 0
0 0 0 7 8 9 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```
(g)

'same' convolution result
```
0 0 0 0 0
0 1 2 3 0
0 4 5 6 0
0 7 8 9 0
0 0 0 0 0
```
(h)

15

Matlab command: *imfilter*

Options	Description
Filtering Mode	
'corr'	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
'conv'	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
P	The boundaries of the input image are extended by padding with a value, P (written without quotes). This is the default, with value 0.
'replicate'	The size of the image is extended by replicating the values in its outer border.
'symmetric'	The size of the image is extended by mirror-reflecting it across its border.
'circular'	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
'full'	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
'same'	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.