Linear Systems: Discrete case & 2D

January 15, 2004

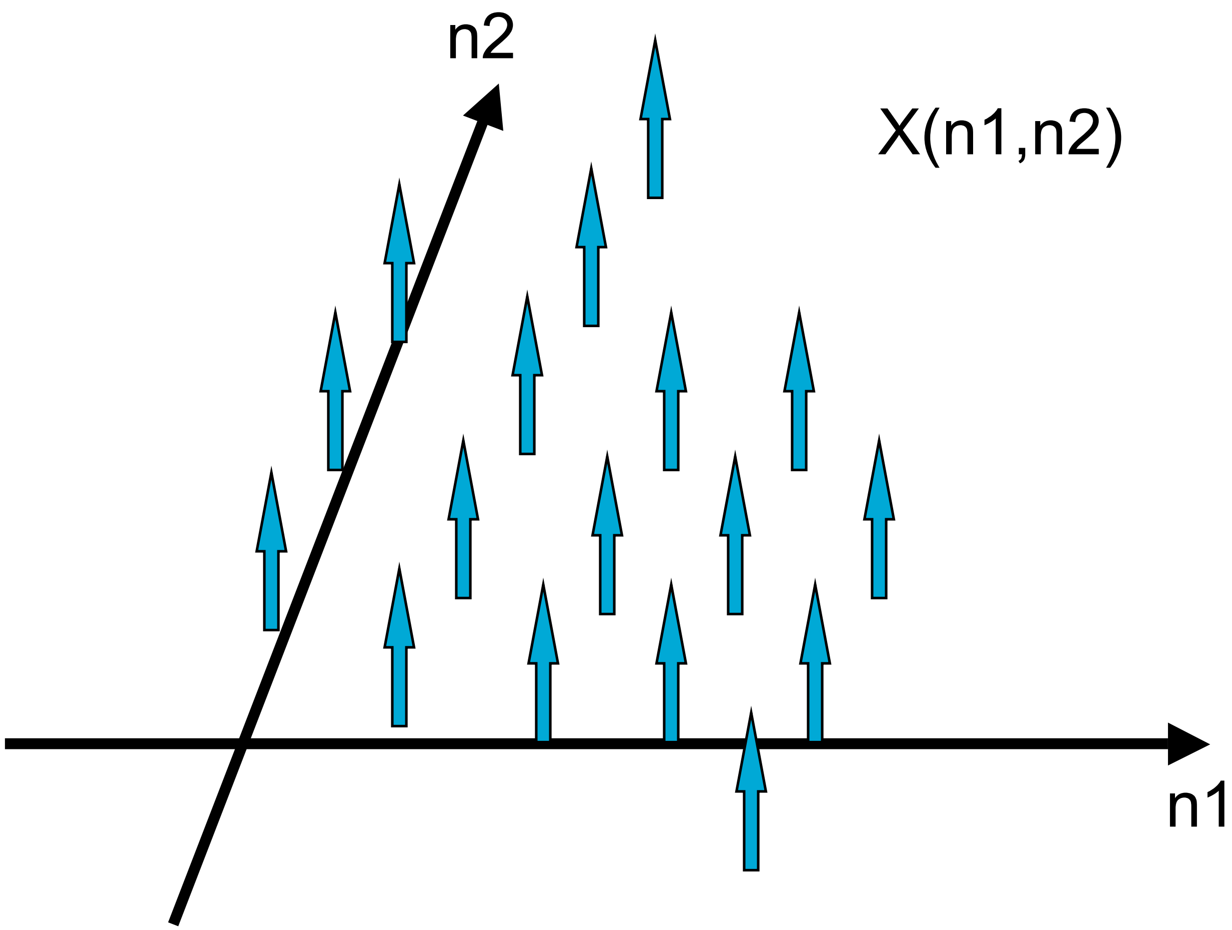
Linear systems-review

Part 1: Review from G&W (continuous case)

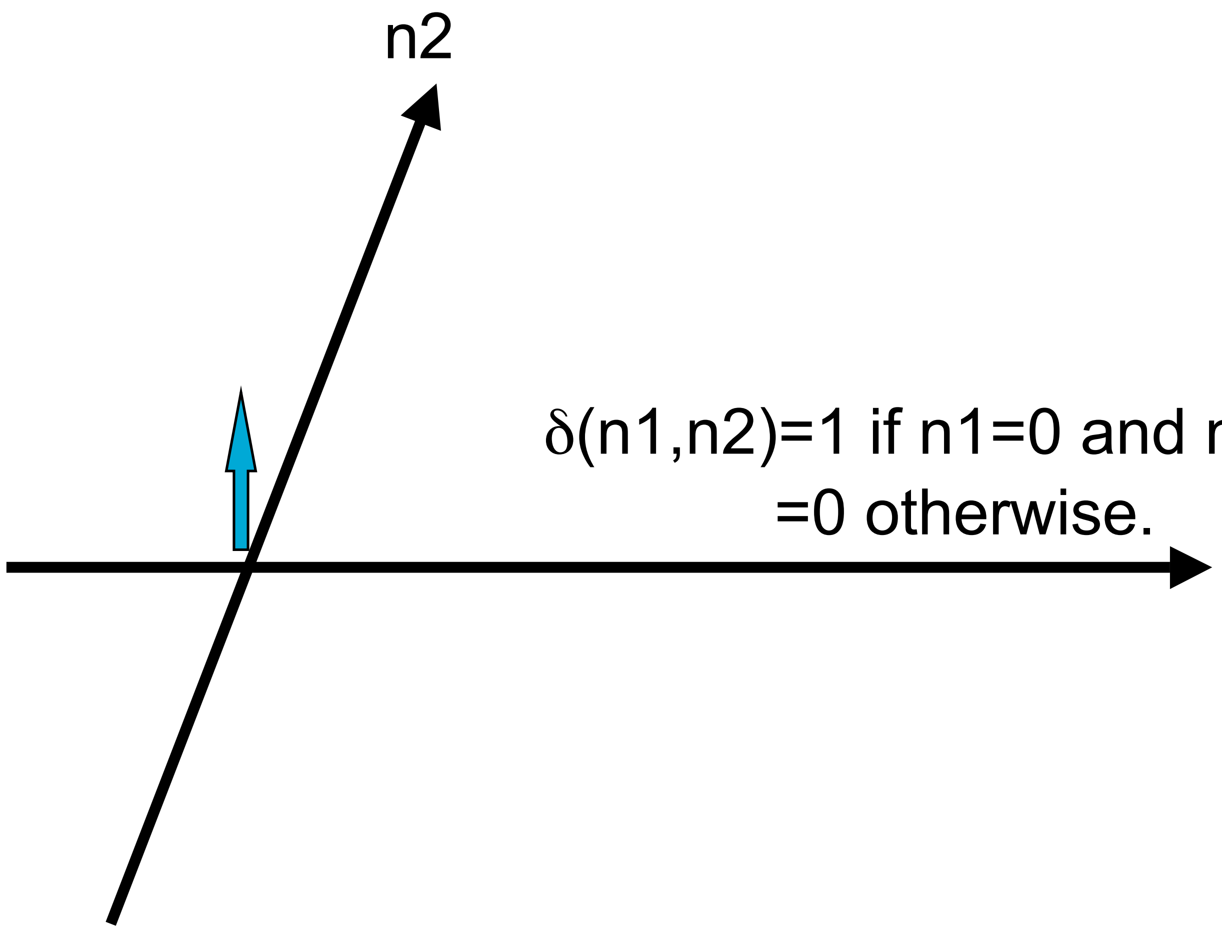
Part 2: Discrete case & 2D

- 2D impulse function
- Line function
- Step function
- Linear systems and Shift invariance
- Impulse Response of LSI Systems
- 2-D Convolution

2-D Systems

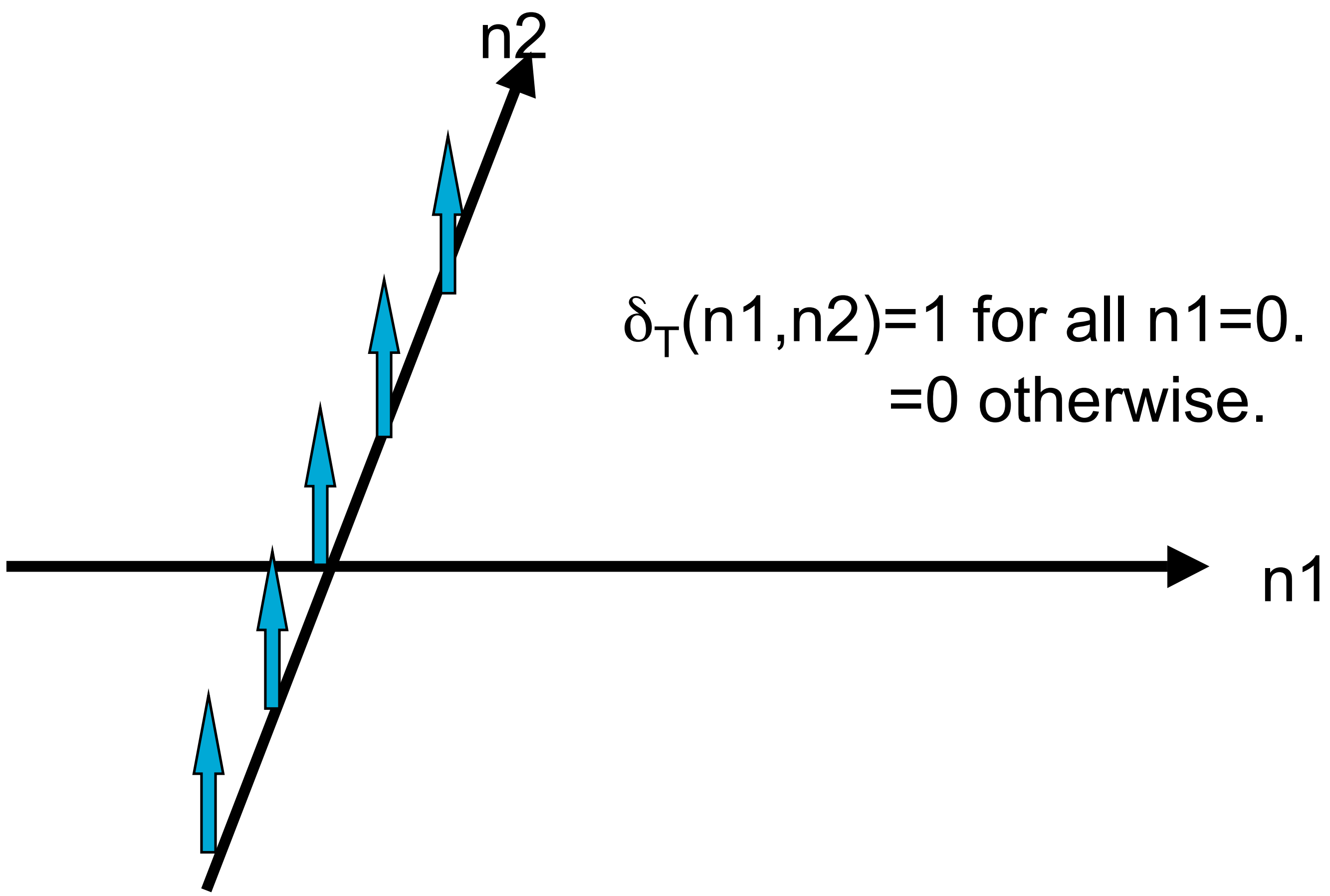


Impulse Function(Kronecker Delta)



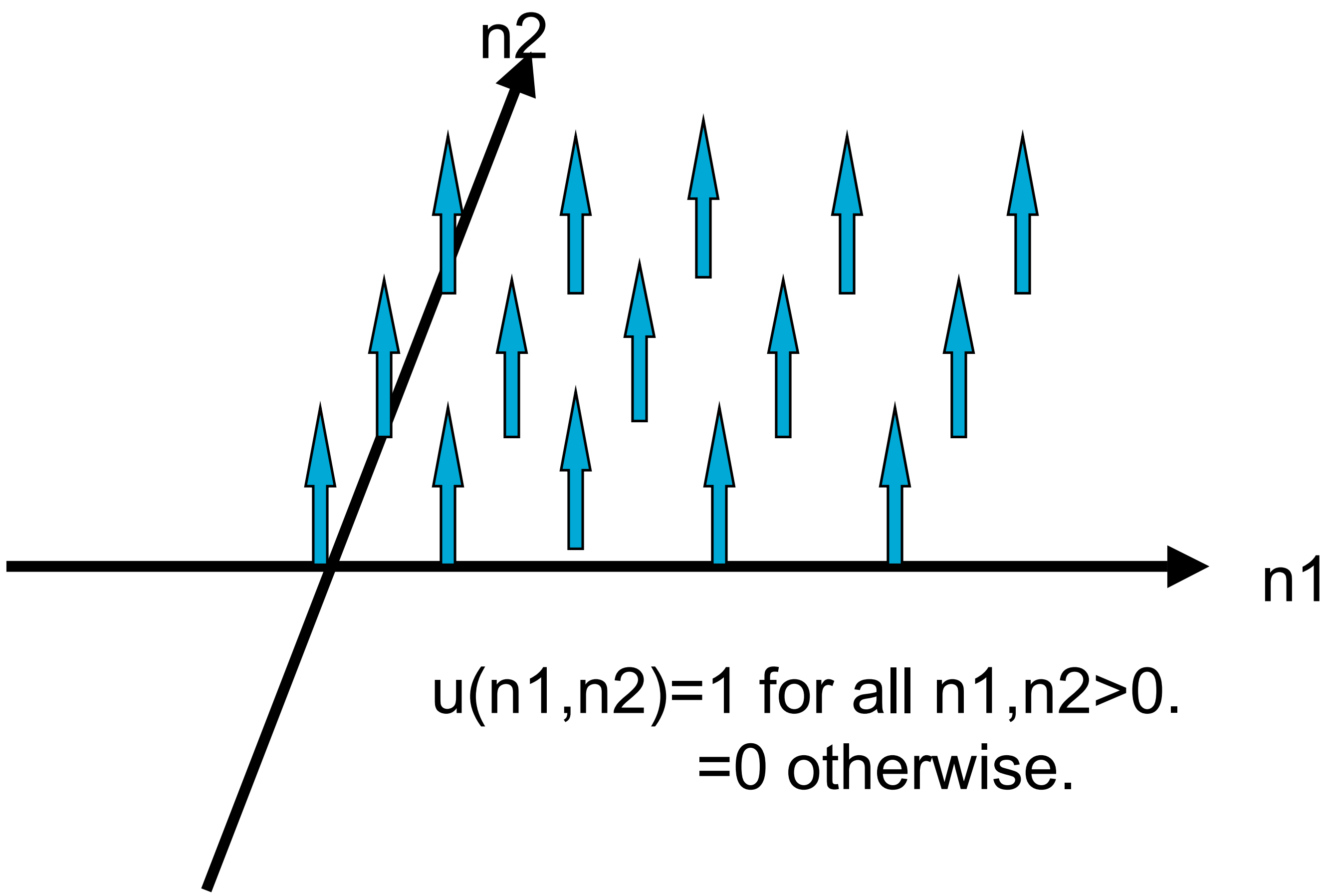
$\delta(n1,n2)=1$ if $n1=0$ and $n2=0$.
$=0$ otherwise.

Line Impulse



δ_T(n1,n2)=1 for all n1=0.
=0 otherwise.

Unit Step Function

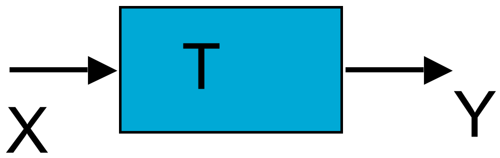


u(n1,n2)=1 for all n1,n2>0.
=0 otherwise.

“ 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 $\delta(m,n)$.

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

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

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

Convolution: example

$x(m,n)$

n \ m	0	1	2
1	1	4	1
0	2	5	3

$h(m,n)$

n \ m	0	1
1	1	1
0	1	-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
0	0	0	0

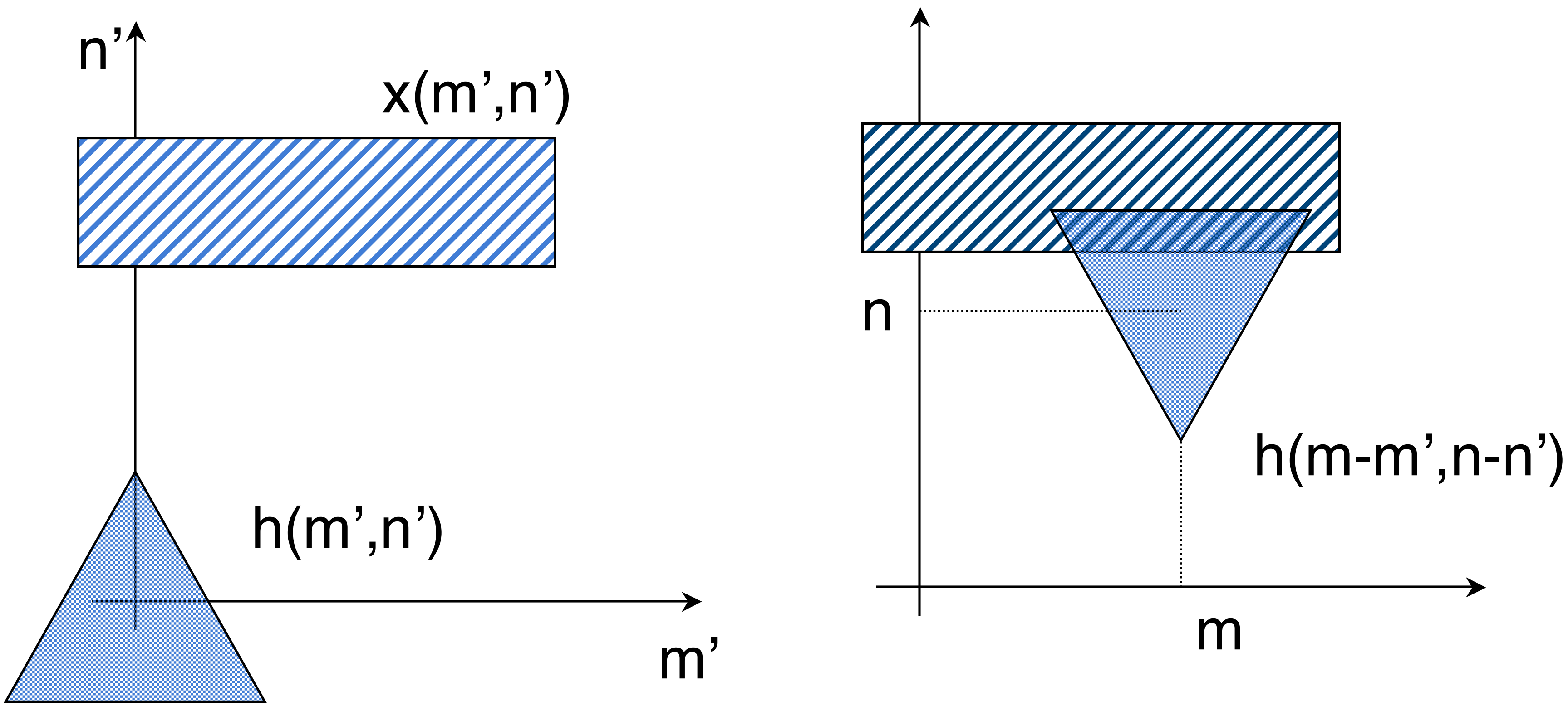
$= 3$

$y(m,n)=$

1	5	5	1
3	10	5	2
2	3	-2	-3

verify!

Discrete Convolution in 2D



output=sum of the product
of the two in the overlapped regio