Physiological Modeling

Parametric

- *Structural*
- *System Components*
- *Differential Equations*

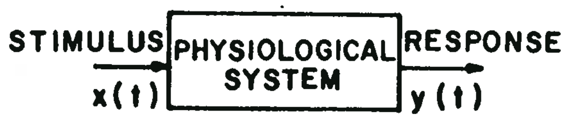
Rate Processes

Nonparametric

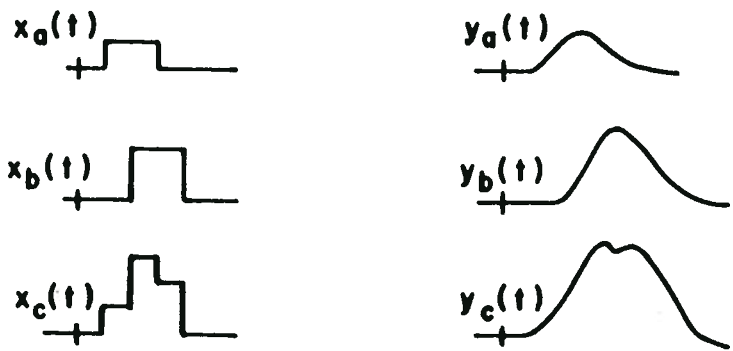
- *Functional*
- *System Kernels*
- *Integral Equations*

Input-Output Maps

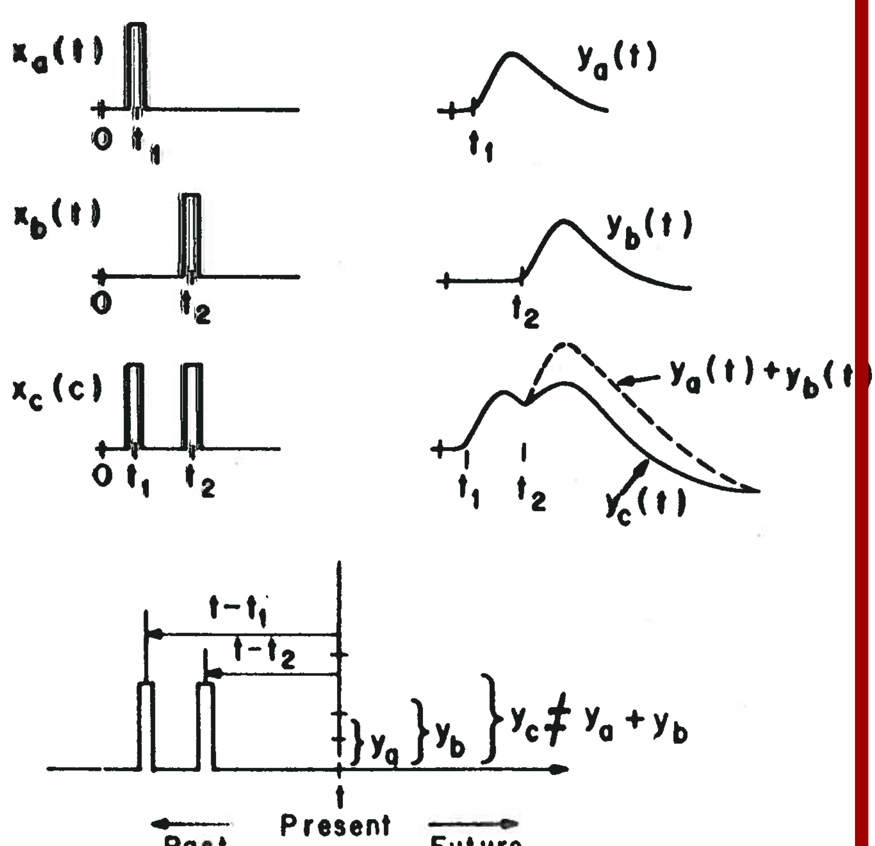
Linear and Nonlinear Systems



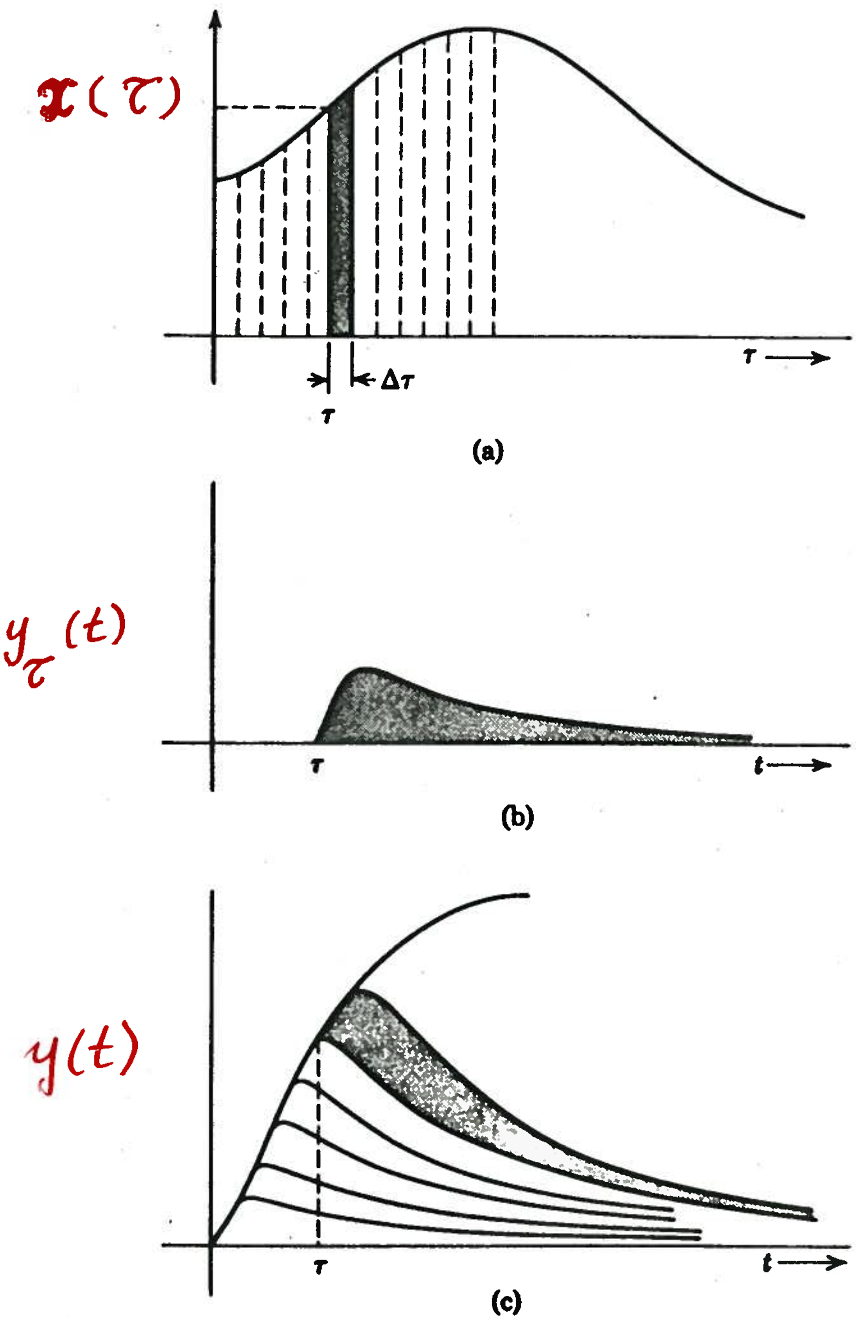
Linear



Non linear



Response of a Linear System



* $x(\tau)$ is represented as a sum of narrow pulses of width $\Delta\tau$ and height $x(\tau)$. As $\Delta\tau \to 0$ each narrow pulse becomes an impulse of strength $x(\tau)\Delta\tau$.

* The response of the system to a unit impulse at τ is $h_L(t-\tau)$. Hence, the response to an impulse of strength $x(\tau)\Delta\tau$ is

$$y_\tau(t) = \lim_{\Delta\tau \to 0} x(\tau)\Delta\tau\, h_L(t-\tau)$$

* The response $y(t)$ is the sum of such responses from $\tau=0$ to $\tau=t$

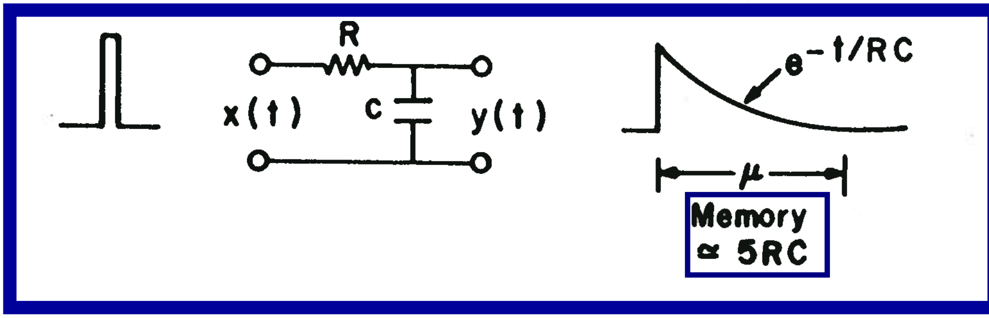
$$y(t) = \lim_{\Delta\tau \to 0} \sum_{\tau=0}^{t} x(\tau)\, h_L(t-\tau)\,\Delta\tau$$

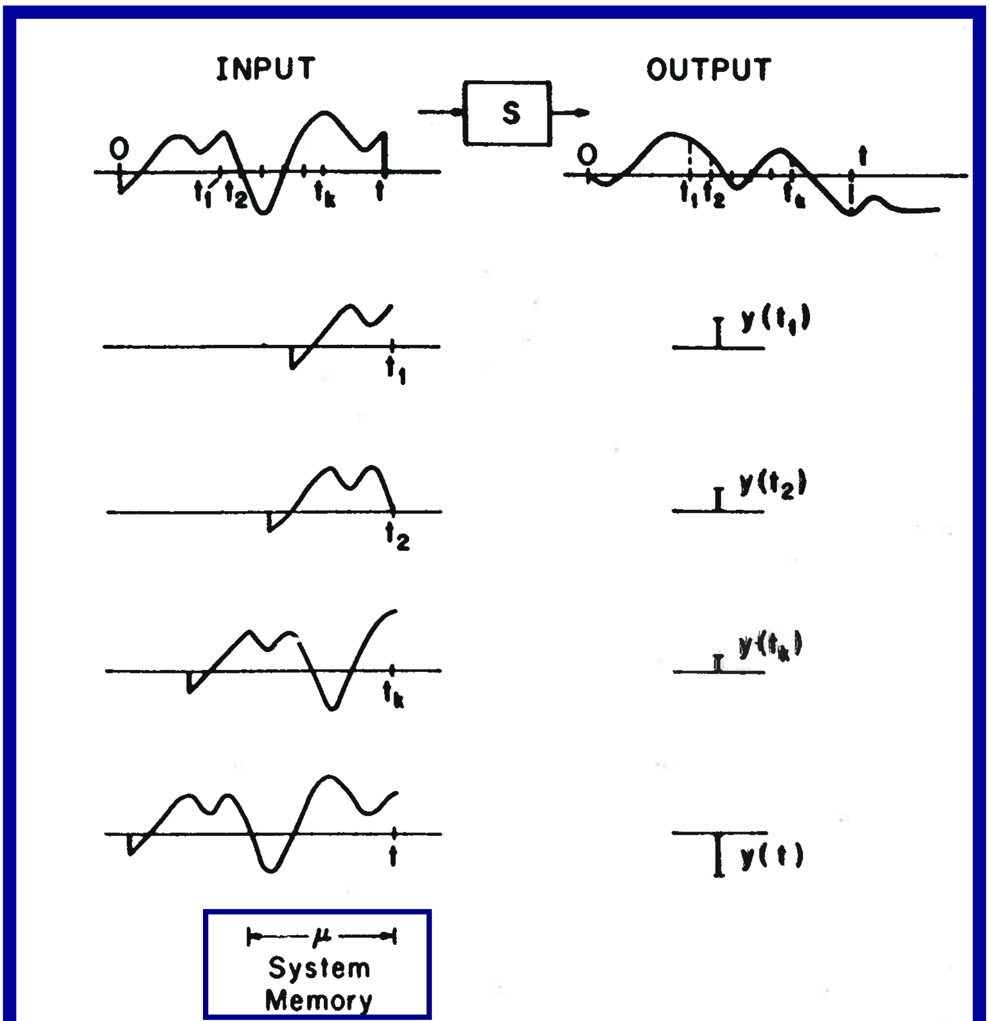
$$= \int_0^t x(\tau)\, h_L(t-\tau)\, d\tau$$

This is called the **Convolution Integral**

Output Dependence on the Iput Past

System Memory





Definition: A functional is a function of a function (i.e. a function whose argument is a function) and whose value is a number. We may write it as

$$\boxed{y = F\,[x(t)]}$$

where F is the functional, the function $x(t)$ is the argument, and y is the value of the functional.

Example:

The convolution integral

$$\boxed{y(t) = \int_0^t x(\tau)\, h_L(t-\tau)\, d\tau}$$

is a functional representation of a linear system.

Vito Volterra [1860 (Ancona) - 1940 (Roma)]

Vito Volterra was an **Italian mathematician and physicist**. He is known for his contributions to **mathematical biology and integral equations**. He joined the opposition to the Fascist regime (1922). Because of his political philosophy, he also refused to take a mandatory oath of loyalty (1931). He lived largely abroad, returning to Rome just before his death.

Royal Society (1910) - Royal Society of Edinburgh (1913)
A moon crater is named after him

FIRST-ORDER VOLTERRA SYSTEMS

$$y(t) = V_1[x(t)]$$

$$= \int_0^{\infty} k_1(\tau)\, x(t-\tau)\, d\tau$$

where

$V_1[x(t)] \triangleq$ First-order Volterra functional.

$k_1(t) \triangleq$ First-order Volterra kernel.

Example: Linear time-invariant systems

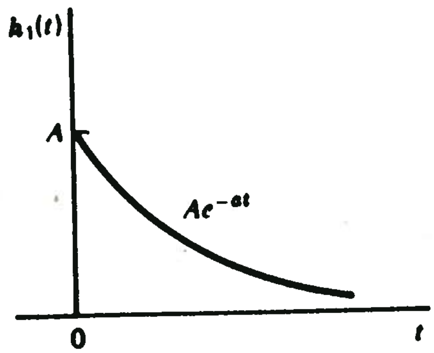
In this case $k_1(t) = h_L(t)$

where $h_L(t)$ is the impulse response of the system.

For example, $h_L(t) = A e^{-at}$ for $t \geq 0$ and $a > 0$.

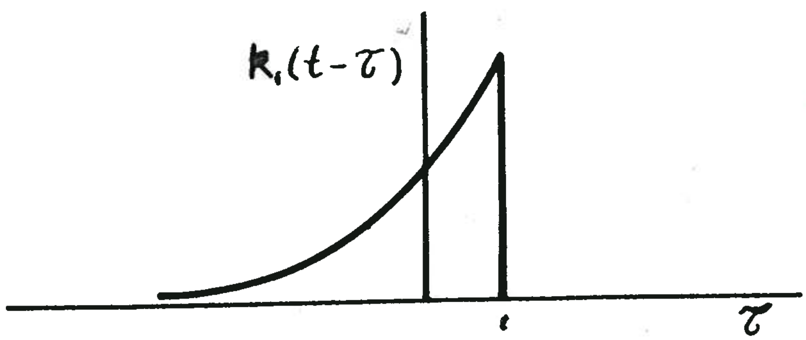
Linear Time-Invariant System

$$\boxed{k_1(t) = h_L(t)}$$



$k_1(t)$

Memory of System



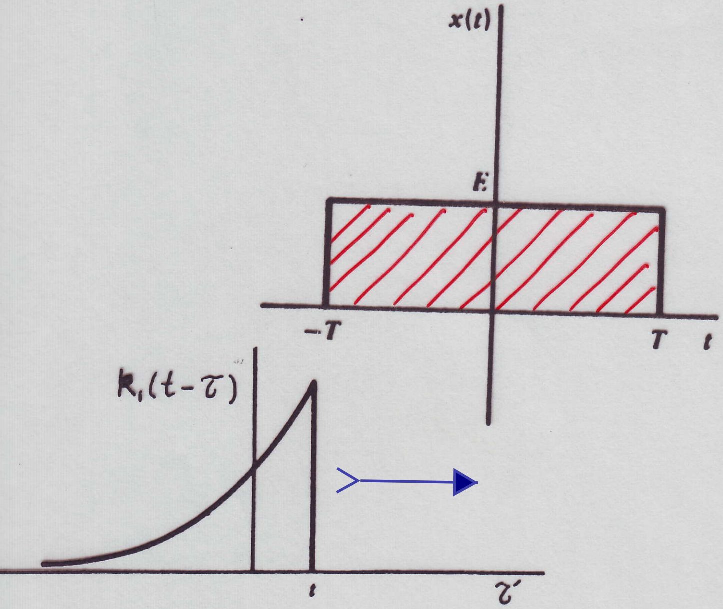
$k_1(t-\tau)$

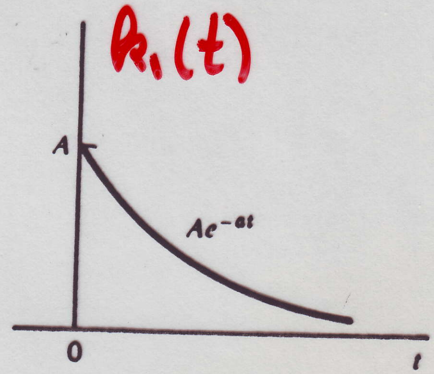
$$y(t) = V_1[x(t)] = \int_0^{\infty} x(\tau)\, k_1(t-\tau)\, d\tau$$

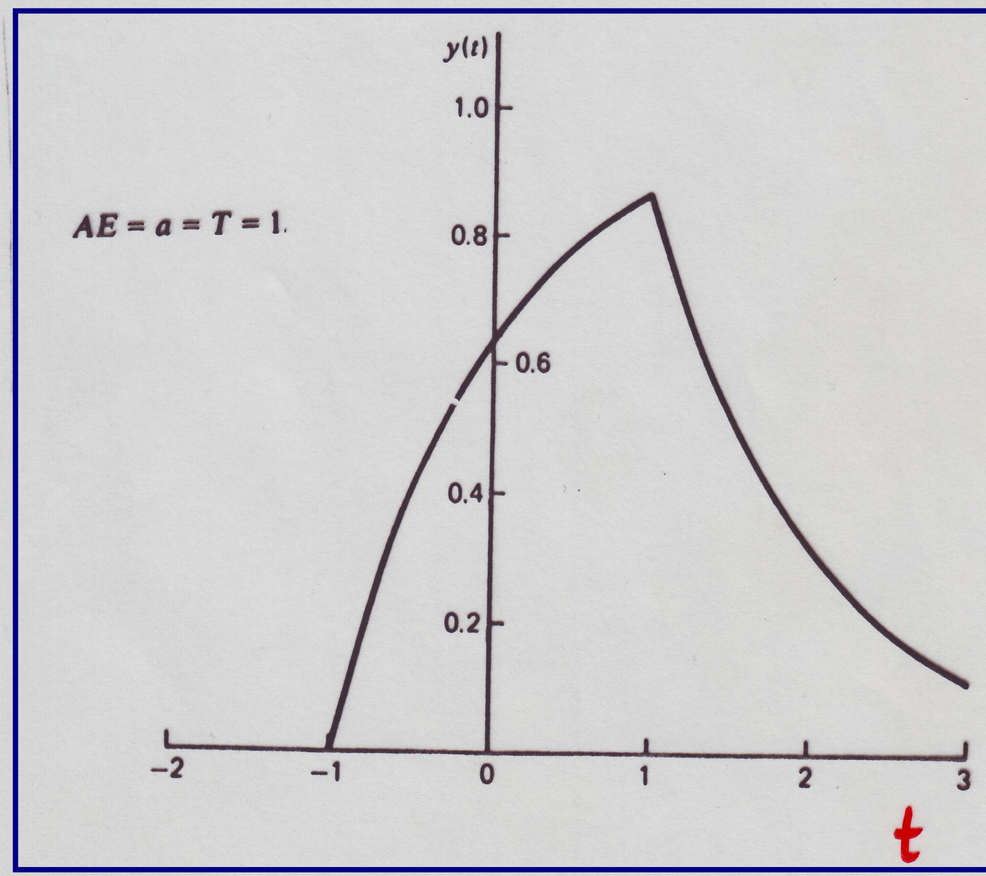
$$= \int_0^{\infty} k_1(\tau)\, x(t-\tau)\, d\tau$$

(The Convolution Integral)

For a rectangular input pulse x(t)







SECOND-ORDER VOLTERRA SYSTEMS

$$y(t) = V_2\left[x(t)\right]$$

$$= \int_0^\infty \int_0^\infty k_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1\, d\tau_2$$

where

$V_2[x(t)] \triangleq$ Second-order Volterra functional.

$k_2(\tau_1, \tau_2) \triangleq$ Second-order Volterra Kernel.

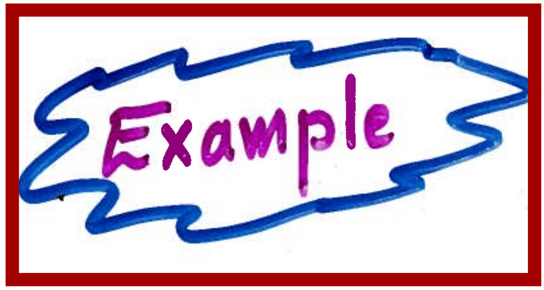
General Second-Order Volterra Systems

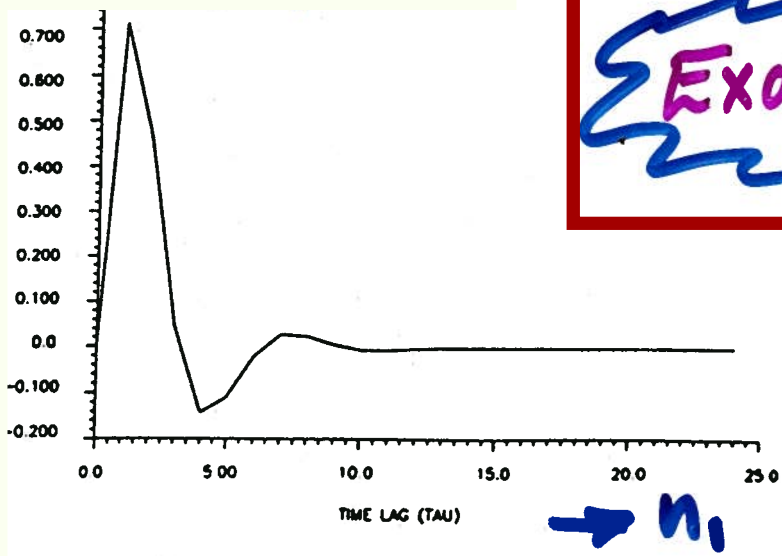
$$y(t) = V_0[x(t)] + V_1[x(t)] + V_2[x(t)]$$

$$= k_0 + \int_0^\infty k_1(\tau)\, x(t-\tau)\, d\tau$$

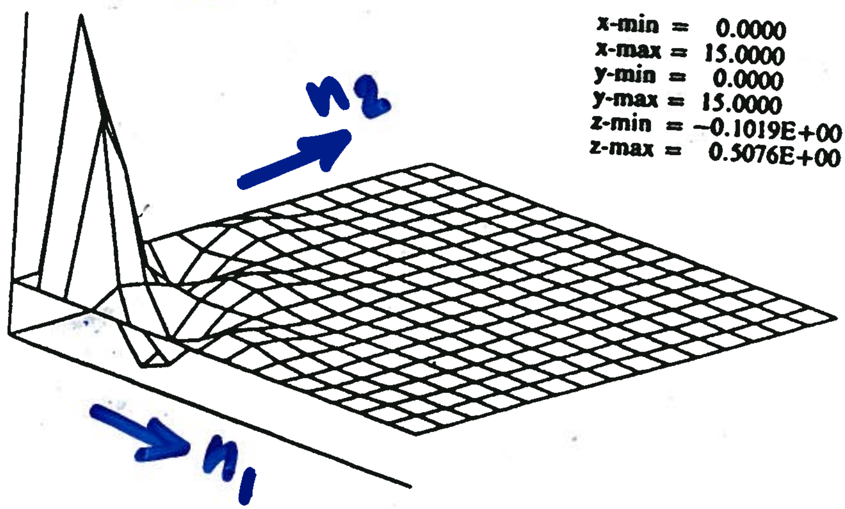
$$+ \int_0^\infty \int_0^\infty k_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1\, d\tau_2$$

First Order Kernel





Second Order Kernel



Volterra Series

$$x(t) \longrightarrow \boxed{\quad\quad} \longrightarrow y(t)$$

$$y(t) = k_0 + \int_0^{\infty} k_1(\tau)\, x(t-\tau)\, d\tau$$
$$+ \int_0^{\infty}\!\!\int_0^{\infty} k_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1\, d\tau_2$$
$$+ \int_0^{\infty}\!\!\int_0^{\infty}\!\!\int_0^{\infty} k_3(\tau_1, \tau_2, \tau_3)\, x(t-\tau_1)\, x(t-\tau_2)\, x(t-\tau_3)\, d\tau_1\, d\tau_2\, d\tau_3$$
$$+ \cdots$$

where $k_i(\cdot)$ is the $i\underline{th}$ Volterra Kernel.

The impulse response of a nonlinear system

$$x(t) = \delta(t)$$

$$y(t) = k_0 + k_1(t) + k_2(t,t) + k_3(t,t,t) + \dots$$

GENERAL VOLTERRA SYSTEMS

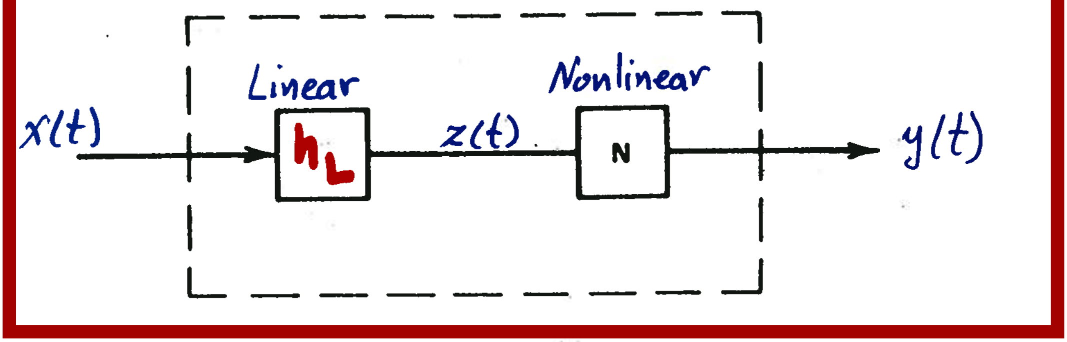
$$y(t) = \sum_{n=0}^{\infty} V_n[x(t)]$$

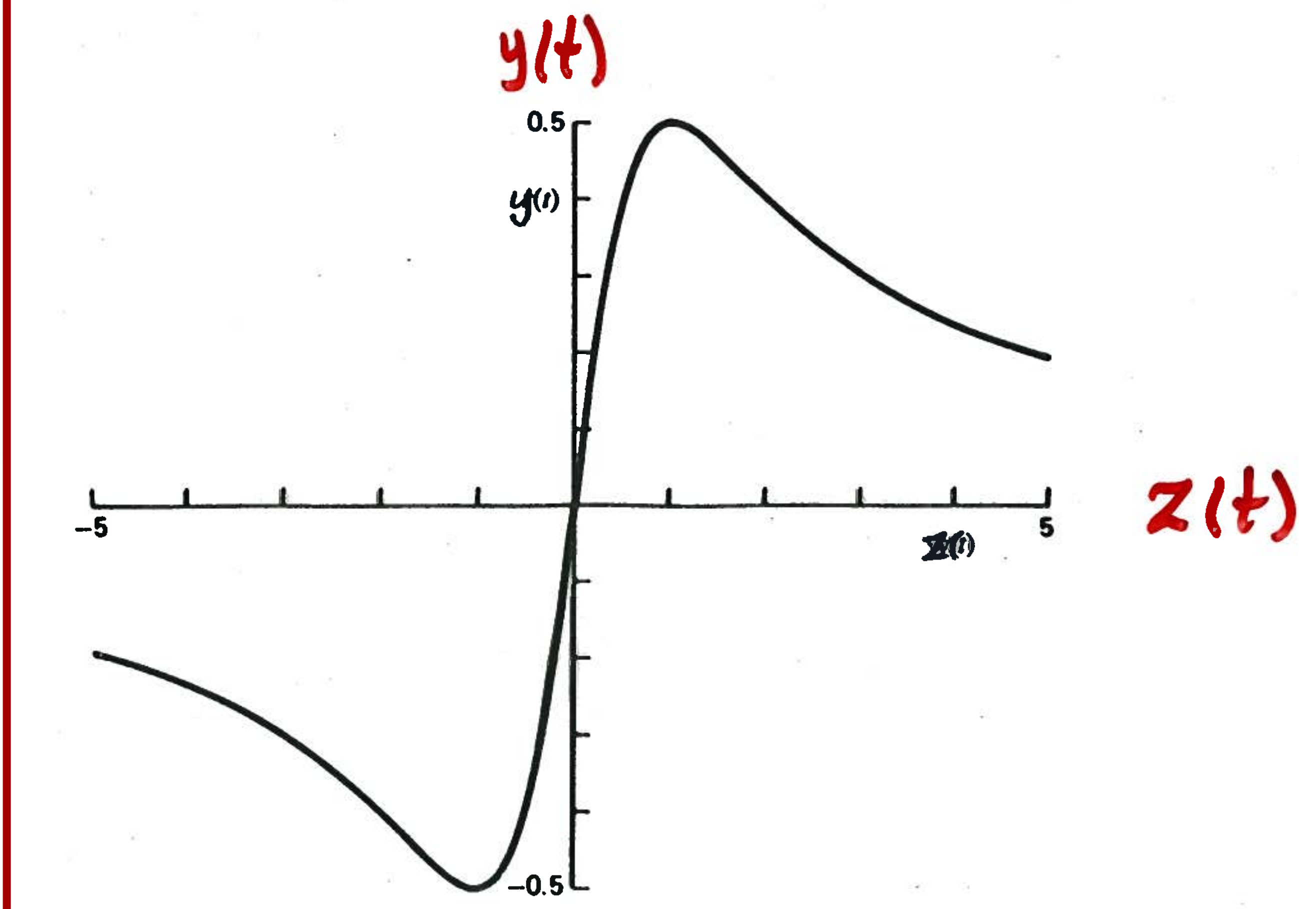
$$= \sum_{n=0}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} k_n(\tau_1, \ldots, \tau_n)\, x(t-\tau_1) \cdots x(t-\tau_n)\, d\tau_1 \cdots d\tau_n$$

♫ This is a generalization of the convolution integral representation of a linear system.

Example

LN Cascade





$$y(t) = \frac{z(t)}{1 + z^2(t)} = \mathcal{N}[z(t)]$$

Output of linear component is $z(t)$

$$z(t) = \int_0^{\infty} h_L(\tau)\, x(t-\tau)\, d\tau$$

Output of Nonlinear Component is $y(t)$

$$y(t) = \frac{z(t)}{1+z^2(t)} = \mathcal{N}[z(t)]$$

A Taylor series expansion of $y(t)$ is

$$y(t) = \sum_{n=0}^{\infty} (-1)^n \, [z(t)]^{2n+1}$$

which converges only for $z^2(t) < 1$.

Volterra series

$$y(t) = \sum_{n=0}^{\infty} V_{2n+1}[x(t)]$$

$$h_{2n+1}(\tau_1, \dots, \tau_{2n+1}) = (-1)^n h_L(\tau_1)\, h_L(\tau_2) \dots h_L(\tau_{2n+1})$$

Volterra series will diverge for $|z(t)| \geq 1$.

Limitations of the Volterra Series :

1. Convergence : Same limitations as those encountered in Taylor series.
2. Measurement of Volterra kernels.

Norbert Wiener : A famous child prodigy, he later became an early researcher in stochastic and noise processes, contributing work relevant to electronic engineering, electronic communication, and control systems.

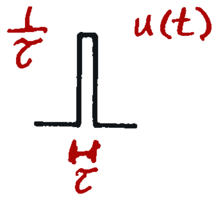
Wiener is considered the originator of ***cybernetics,*** a formalization of the notion of feedback, with implications for engineering, systems control, computer science, biology, neuroscience, philosophy, and the organization of society.

Cybernetics: Or Control and Communication in the Animal and the Machine was written by Norbert Wiener and published in 1948. It is the first public usage of the term "cybernetics" to refer to ***self-regulating mechanisms***. The book laid the theoretical foundation for servomechanisms (whether electrical, mechanical or hydraulic), automatic navigation, analog computing, and reliable communications.

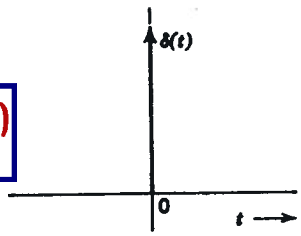
Wiener Filter : For signal processing. It's purpose is to reduce the amount of noise present in a signal by comparison with an estimation of the desired noiseless signal.

INPUT STIMULI

Pulse



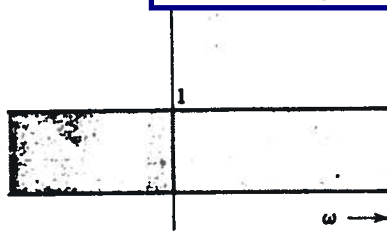
$x(t)$



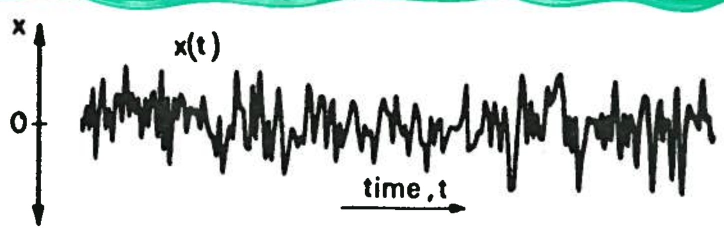
$\boxed{\lim_{\tau \to 0} u(t)}$

unit impulse

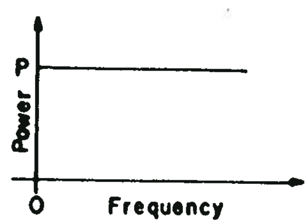
Fourier Transform



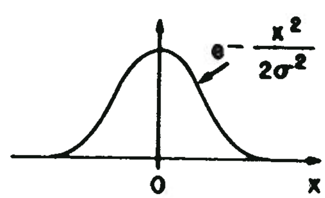
Gaussian White Noise



'zero mean'



Power Spectrum



Amplitude Distribution

- Any two samples are statistically independent.

- Stationary random process.

THE WIENER SERIES

$$y(t) = \sum_{n=0}^{\infty} G_n [h_n ; x(t)]$$

The Wiener G-Functionals are orthogonal to each other with respect to a Gaussian White Noise (GWN) input $x(t)$.

Definition: The nth order Wiener G-functional is the sum of n+1 homogeneous Wiener functionals of decreasing order.

$$G_n [h_n ; x(t)] = W_n [x(t)] + \sum_{i=0}^{n-1} W_{i(n)} [x(t)]$$

for $n \geq 1$

A homogeneous functional is such that

$$W_n [c\, x(t)] = c^n\, W_n [x(t)]$$

♫ A change in input amplitude causes a change in out amplitude but not output waveform.

$$
\begin{aligned}
y(t) = {} & h_0 \\
& + \int_0^\infty h_1(\tau_1)\, x(t-\tau_1)\, d\tau_1 + h_{0(1)} \\
& + \int_0^\infty \int_0^\infty h_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1\, d\tau_2 \\
& + \int_0^\infty h_{1(2)}(\tau_1)\, x(t-\tau_1)\, d\tau_1 + h_{0(2)} \\
& + \int_0^\infty \int_0^\infty \int_0^\infty h_3(\tau_1, \tau_2, \tau_3)\, x(t-\tau_1)\, x(t-\tau_2)\, x(t-\tau_3) \\
& \qquad d\tau_1\, d\tau_2\, d\tau_3 \\
& + \int_0^\infty \int_0^\infty h_{2(3)}(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2)\, d\tau_1\, d\tau_2 \\
& + \int_0^\infty h_{1(3)}(\tau_1)\, x(t-\tau_1)\, d\tau_1 \\
& + h_{0(3)} \\
& + \cdots\cdots
\end{aligned}
$$

$h_i(\cdot) \triangleq$ ith order Wiener Kernel

$h_{j(i)}(\cdot) \triangleq$ derived jth order Wiener Kernel associated with ith order Wiener kernel

Orthogonality Condition on G-functionals

$\Rightarrow \mathcal{E}\{G_i G_j\} = 0$ for GWN

$$h_{0(1)} = 0$$

$$h_{1(2)}(\tau_1) = 0$$

$$h_{0(2)} = -P\int_0^\infty h_2(\tau_1, \tau_1)\, d\tau_1$$

where $P \triangleq$ input power level

$$h_{2(3)}(\tau_1, \tau_2) = 0$$

$$h_{1(3)}(\tau_1) = -3P\int_0^\infty h_3(\tau_1, \tau_2, \tau_2)\, d\tau_2$$

$$h_{0(3)} = 0$$

G-functional	0^{th}-order kernels	1^{st}-order kernels	2^{nd}-order kernels	3^{rd}-order kernels	4^{th}-order kernels	5^{th}-order kernels
G_0	h_0					
G_1		h_1				
G_2	$h_{0(2)}$		h_2			
G_3		$h_{1(3)}$		h_3		
G_4	$h_{0(4)}$		$h_{2(4)}$		h_4	
G_5		$h_{1(5)}$		$h_{3(5)}$		h_5

The Relation Between Wiener & Volterra Kernels

$$y(t) = \sum_{n=0}^{\infty} V_n[x(t)] = \sum_{n=0}^{\infty} G_n[h_n; x(t)]$$

There is a unique analytical relation between the Wiener and Volterra kernels of a system, which represents the transformation of the functional expansion basis.

Example : For a fifth-order system

$$\sum_{n=0}^{5} V_n[x(t)] = \sum_{n=0}^{5} G_n[h_n ; x(t)]$$

Collect and Equate Similar terms

1. Constants

$$k_0 = h_0 + 0 + h_{0(2)} + 0 + h_{0(4)}$$

2. First-order Kernels

$$k_1 = 0 + h_1 + 0 + h_{1(3)} + 0 + h_{1(5)}$$

3. Second-order Kernels

$$k_2 = 0 + 0 + h_2 + 0 + h_{2(4)} + 0$$

4. Third-order Kernels

$$k_3 = 0 + 0 + 0 + h_3 + 0 + h_{3(5)}$$

5. Fourth-order Kernels

$$\boxed{k_4 = 0 + 0 + 0 + 0 + h_4 + 0}$$

6. Fifth-order Kernels

$$\boxed{k_5 = 0 + 0 + 0 + 0 + 0 + h_5}$$

The Lee-Schetzen Approach for Estimation of Wiener Kernels Using a GWN Input

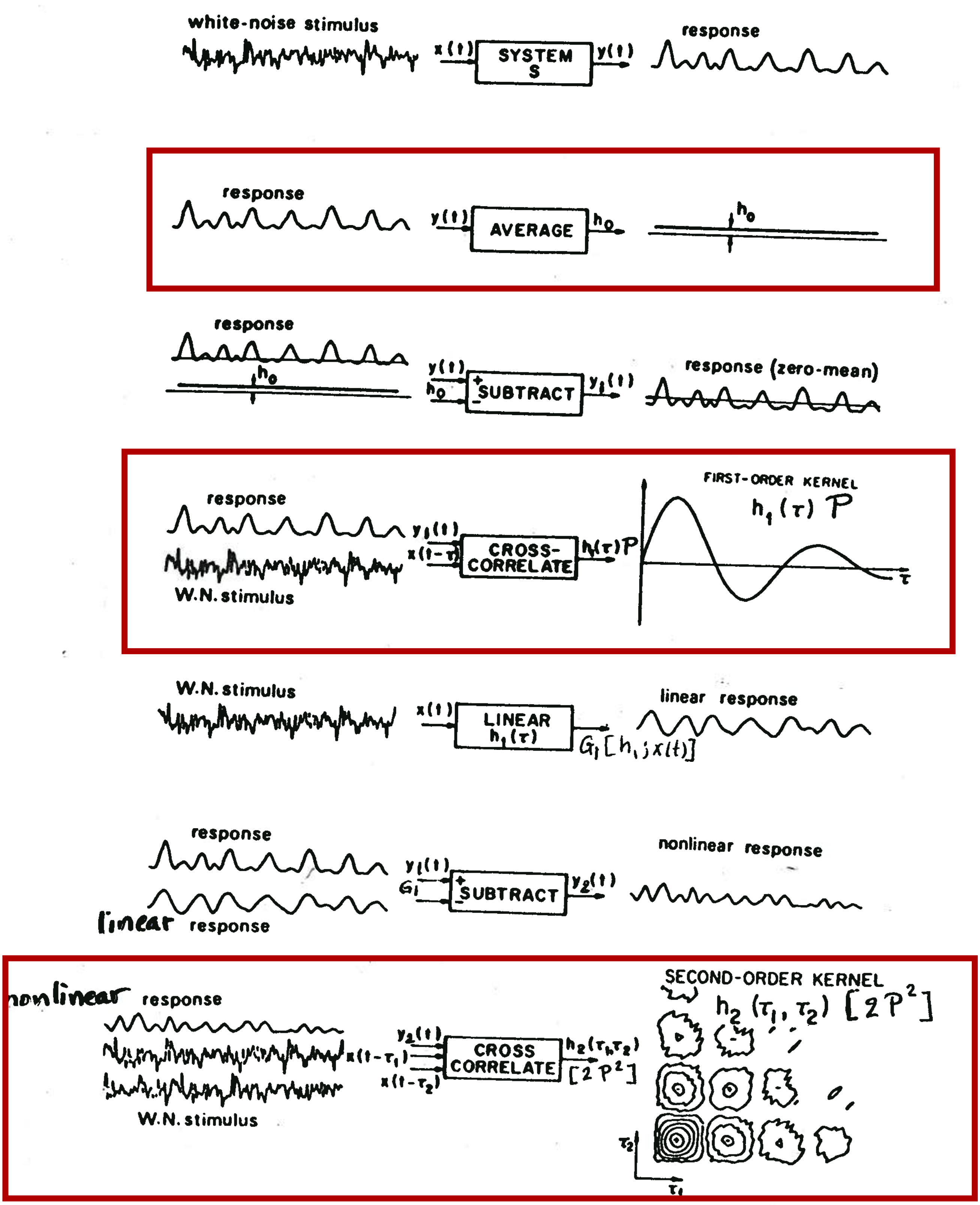
Estimation of $\boxed{h_n(\tau_1, \ldots, \tau_n)}$ is through input-output crosscorrelation for $n \geq 1$

$$\boxed{h_n(\tau_1, \ldots, \tau_n) = \frac{1}{n!\, P^n} \mathcal{E}\left\{ y_n(t)\, x(t-\tau_1) \ldots x(t-\tau_n) \right]}$$

where $y_n(t)$ represents the nth order residual

$$\boxed{y_n(t) = y(t) - \sum_{m=0}^{n-1} G_m[h_m; x(t)]}$$

The Lee-Schetzen Approach



Auto correlation Function

$$\phi_{xx}(\tau) = \lim_{R\to\infty} \frac{1}{2R} \int_{-R}^{R} x(t)\, x(t-\tau)\, dt$$

$$= \mathcal{E}\{x(t)\, x(t-\tau)\}$$

where $\mathcal{E}\{.\}$ denotes Expected value.

Practically ϕ_{xx} is estimated as

$$\hat{\phi}_{xx}(\tau) = \frac{1}{R-\tau} \int_{0}^{R} x(t)\, x(t-\tau)\, dt$$

* For white noise

$$\phi_{xx}(\tau) = \mathcal{E}\{x(t)\, x(t-\tau)\}$$

$$= P\, \delta(\tau)$$

Power Level

Cross correlation Function

$$\Phi_{yx}(\tau) = \lim_{R \to \infty} \frac{1}{2R} \int_{-R}^{R} y(t)\, x(t-\tau)\, dt$$

$$\hat{\Phi}_{yx}(\tau) = \frac{1}{R-\tau} \int_{0}^{R} y(t)\, x(t-\tau)\, dt$$

$$\Phi_{yx}(\tau) = \mathcal{E}\{ y(t)\, x(t-\tau) \}$$

$$= \mathcal{E}\{ y(t+\tau)\, x(t) \}$$

* For $x(t)$ is white noise

and a linear system characterized by its impulse response

$$\Phi_{yx}(\tau) = P\, h_L(\tau)$$

* $\Phi_{yx}(\tau) = \Phi_{xy}(-\tau)$

Both correlation "Functions" are "Functionals".

For a Linear System: $x(t) \longrightarrow \boxed{h_L(\cdot)} \longrightarrow y(t)$

$$\Phi_{yx}(\tau) = \mathcal{E}\{x(t)\, y(t+\tau)\}$$

$$= \mathcal{E}\left\{x(t) \int_0^\infty h_L(\sigma)\, x(t+\tau-\sigma)\, d\sigma\right\}$$

$$= \int_0^\infty h_L(\sigma)\, \underbrace{\mathcal{E}\{x(t)\, x(t+\tau-\sigma)\}}\, d\sigma$$

$$\boxed{\begin{array}{l} = \Phi_{xx}(\sigma-\tau) \\ = P\, \delta(\sigma-\tau) \\ \text{for Gaussian White Noise} \end{array}}$$

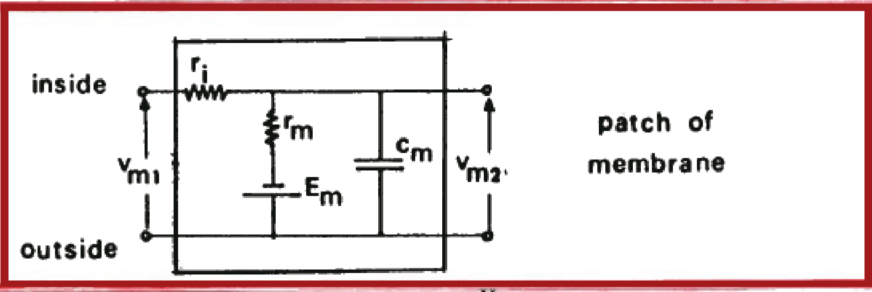
$$\boxed{\Phi_{yx}(\tau)} = P \int_0^\infty h_L(\sigma)\, \delta(\sigma-\tau)\, d\sigma$$

$$\boxed{= P\, h_L(\tau)}$$

♫ $\Phi_{xx}(\tau) = \Phi_{xx}(-\tau)$

$= P\, \delta(\tau)$ for GWN

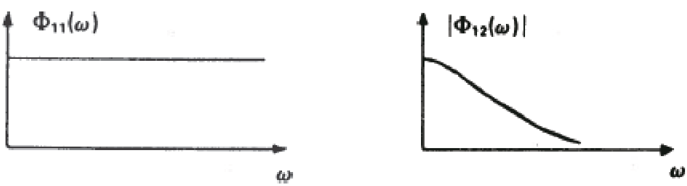
power level



Response to GWN

Correlation Functions

Fourier Transforms



Response Residuals and Wiener Kernels

$$y(t) = \sum_{n=0}^{\infty} G_n[h_n ; x(t)]$$

GWN

Zeroth Order Response Residual

$$y_0(t) = y(t) - [0]$$

$$\mathcal{E}\{y_0(t)\} = h_0$$

First Order Response Residual

$$y_1(t) = y(t) - \big[G_0[h_0 ; x(t)]\big]$$

$$\mathcal{E}\{y_1(t)\, x(t-\tau_1)\} = P\, h_1(\tau_1)$$

cross correlation

input power level

Second Order Response Residual

$$y_2(t) = y(t) - \big[G_0[h_0 ; x(t)] + G_1[h_1 ; x(t)]\big]$$

$$\mathcal{E}\{y_2(t)\, x(t-\tau_1)\, x(t-\tau_2)\} = 2P^2\, h_2(\tau_1, \tau_2)$$

cross correlation

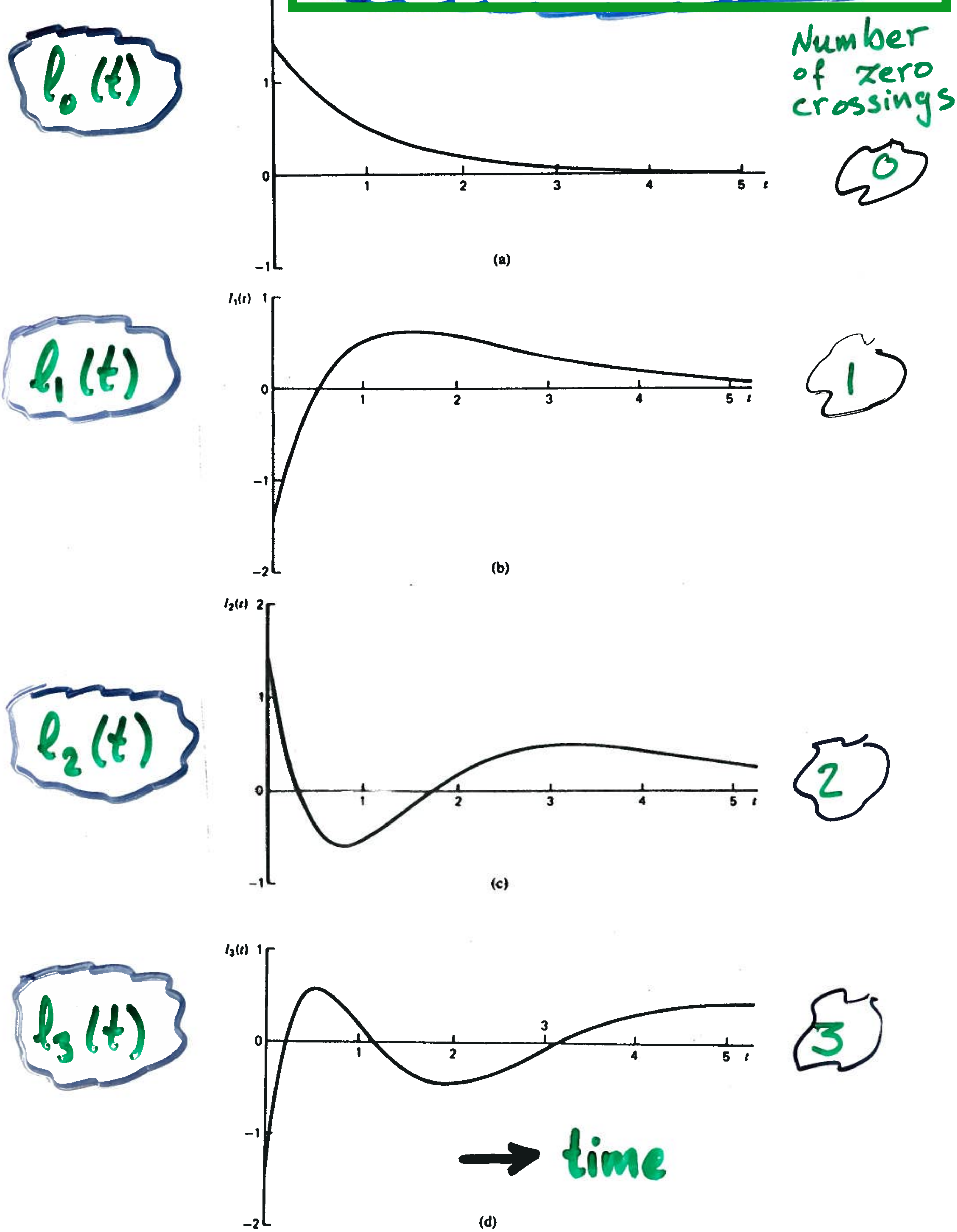
$$\boxed{y_2(t) = y_1(t) - G_1[h_1 ; x(t)]}$$

Limitations: (a) The requirement of a band-limited GWN stimulus that covers the entire system bandwidth.

(b) Long data-records are required to obtain estimates of satisfactory accuracy. Estimate converge to the true values at a rate proportional to $\sqrt{\text{record length}}$.

Wiener Kernel Expansion using Orthogonal Set of Basis Functions

Laguerre Functions



(a) (b) (c) (d)

A Block-Structured Wiener Model

The Wiener-Bose Model

Consider a first-order Wiener System

$$y(t) = G_1[h_1; x(t)]$$

Expand $h_1(\tau)$ in terms of the orthogonal Laguerre functions $\ell_0(\tau), \ell_1(\tau), \ldots, \ell_n(\tau), \ldots$

$$\boxed{h_1(\tau) = \sum_{n=0}^{\infty} c_n \ell_n(\tau)}$$

$$G_1[h_1; x(t)] = G_1\left[\sum_{n=0}^{\infty} c_n \ell_n ; x(t)\right]$$

$$= \sum_{n=0}^{\infty} c_n \, G_1[\ell_n ; x(t)]$$

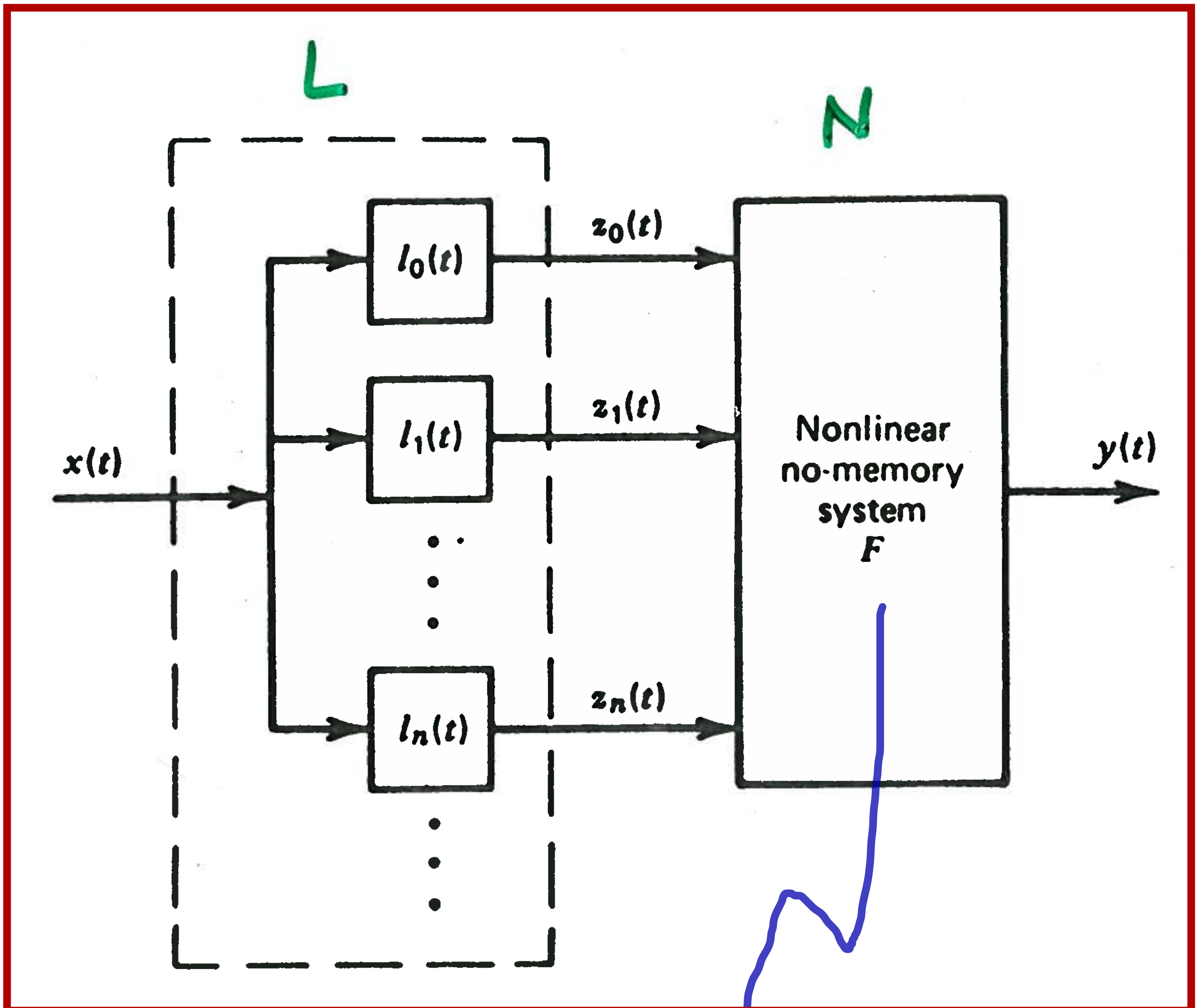
$$= \sum_{n=0}^{\infty} c_n \underbrace{\int_0^{\infty} \ell_n(\tau) x(t-\tau) \, d\tau}_{= z_n(t)}$$

$$y(t) = \sum_{n=0}^{\infty} c_n z_n(t)$$

♫ $\boxed{\text{In general} \quad y(t) = F(z_0(t), z_1(t), \ldots, z_n(t), \ldots)}$

and h_n is expanded in terms of $\{\ell_0, \ell_1, \ldots, \ell_n, \ldots\}$

The Wiener-Bose Model



♫ LN Cascade

♫ $y(t) = F(z_0(t), z_1(t), \ldots, z_n(t), \ldots)$

Let $F(x_0(t), x_1(t), \ldots)$ be analytic so that it can be expanded in a multidimensional power series

$$
\boxed{
\begin{aligned}
y(t) = h_0 &+ \sum_{j=0}^{\infty} c_1(j)\, z_j(t) \\
&+ \sum_{j_1} \sum_{j_2} c_2(j_1, j_2)\, z_{j_1}(t)\, z_{j_2}(t) \\
&+ \sum_{j_1} \sum_{j_2} \sum_{j_3} c_3(j_1, j_2, j_3)\, z_{j_1}(t)\, z_{j_2}(t)\, z_{j_3}(t) \\
&+ \cdots
\end{aligned}
}
$$

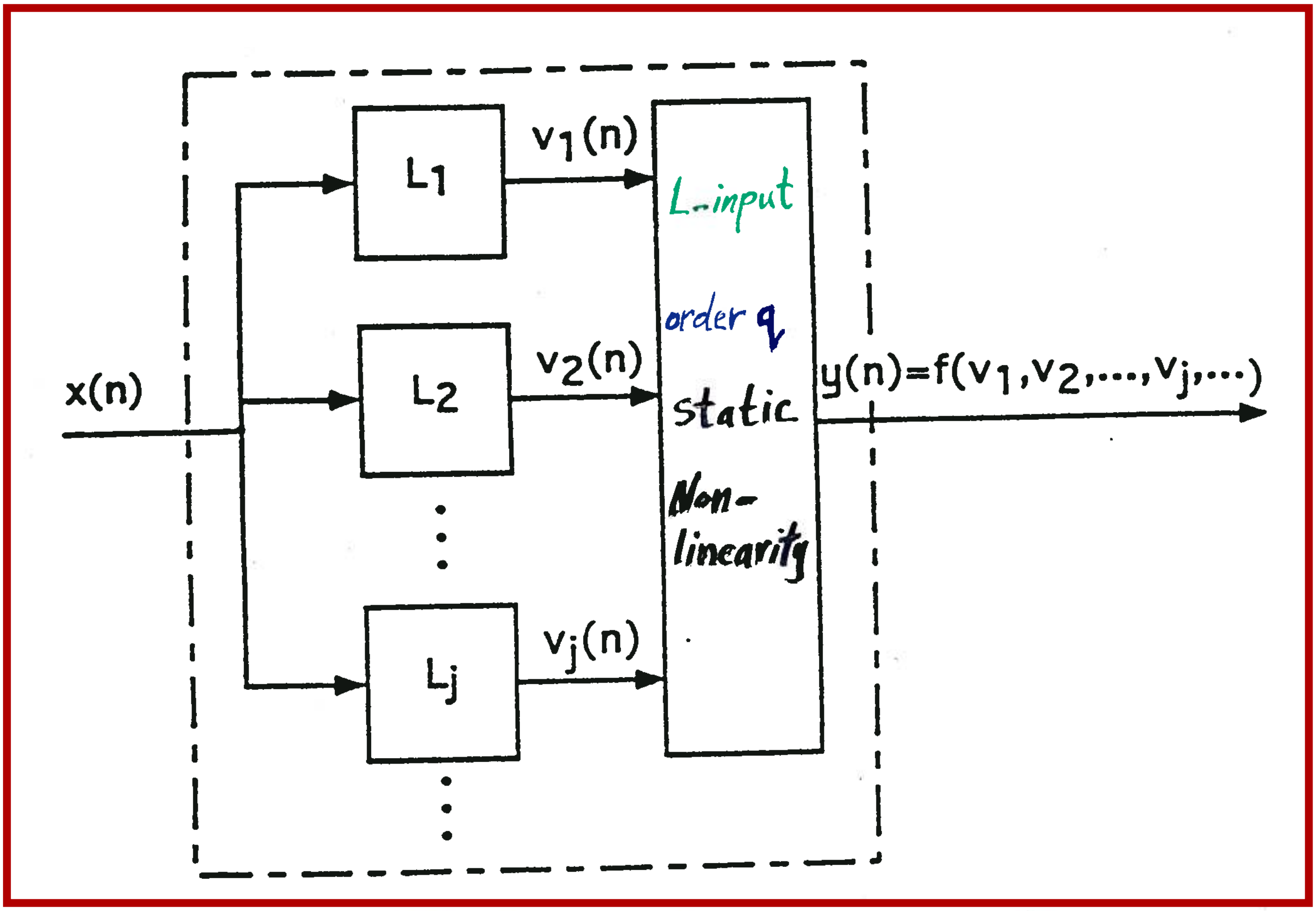
where

$$z_j = \int_0^{\mu} l_j(\tau)\, x(t-\tau)\, d\tau$$

$\mu \triangleq$ System memory

The Wiener and Volterra Kernels are described by summations involving coefficients $c_n(\cdot)$ and Laguerre fns $l_n(t)$.

The Discrete-Time Wiener-Bose Model



$L_j \triangleq$ Linear Filter whose impulse response $b_j(m)$ is the $(j-1)^{th}$ Laguerre function.

$h_m(n_1, \ldots, n_m) \triangleq m^{th}$-order Wiener Kernel

$$\boxed{h_m(n_1,\ldots,n_m) = \sum_{j_1} \cdots \sum_{j_m} c_m(j_1,\ldots,j_m)\, b_{j_1}(n_1) \cdots b_{j_m}(n_m)}$$

The model parameters:

$$q\ ,\ L\ ,\ \{c_r(1,\dots,r)\ ;\ 0 \le r \le q\}$$

order of system

number of Laguerre functions

Coefficients of power series of f

$$y(n) = c_0 + \sum_j c_1(j)\ v_j(n)$$

$$+ \sum_{j_1}\sum_{j_2} c_2(j_1, j_2)\ v_{j_1}(n)\ v_{j_2}(n)$$

$$+ \cdots$$

$$= f(v_1, v_2, \dots, v_j, \dots)$$

$$v_j(n) = \sum_m b_j(m)\ x(n-m)$$

For Laguerre basis functions, then

$$v_j(n) = \sqrt{\alpha}\left[v_j(n-1) + v_{j-1}(n)\right] - v_{j-1}(n-1)$$

$$v_1(n) = \sqrt{\alpha}\ v_1(n-1) + \sqrt{1-\alpha}\ x(n)$$

choice of α is based on the system memory and L.

$c_n(j_1, \dots, j_n)$ can be estimated by a least squares fitting using the known signals $y(t)$ and $\{v_j(t)\}$.

♫ The discrete-time Laguerre function is

$$l_j(m) = \alpha^{\frac{m-j}{2}} (1-\alpha)^{\frac{1}{2}} \sum_{k=0}^{j} (-1)^k \binom{m}{k} \binom{j}{k} \cdot \left[\alpha^{(j-k)} (1-\alpha)^k \right]$$

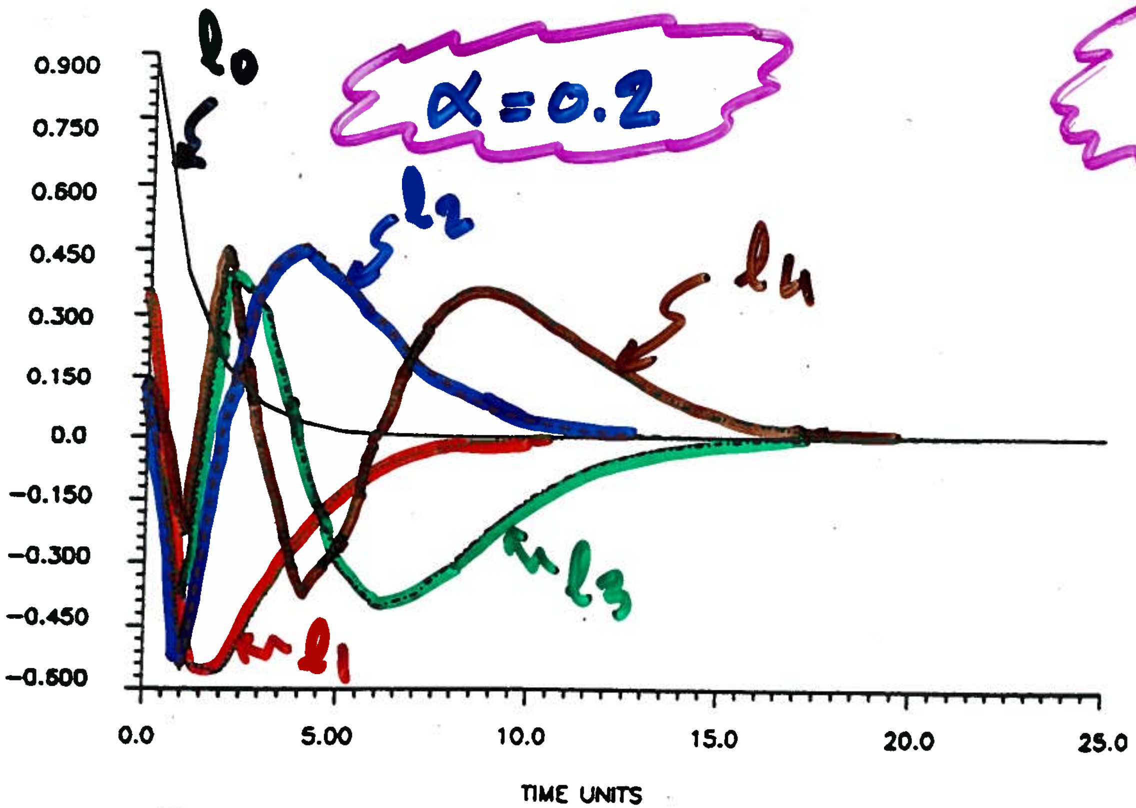
for $m \geq 0$, $j = 0, 1, \dots, L-1$,

$0 < \alpha < 1$

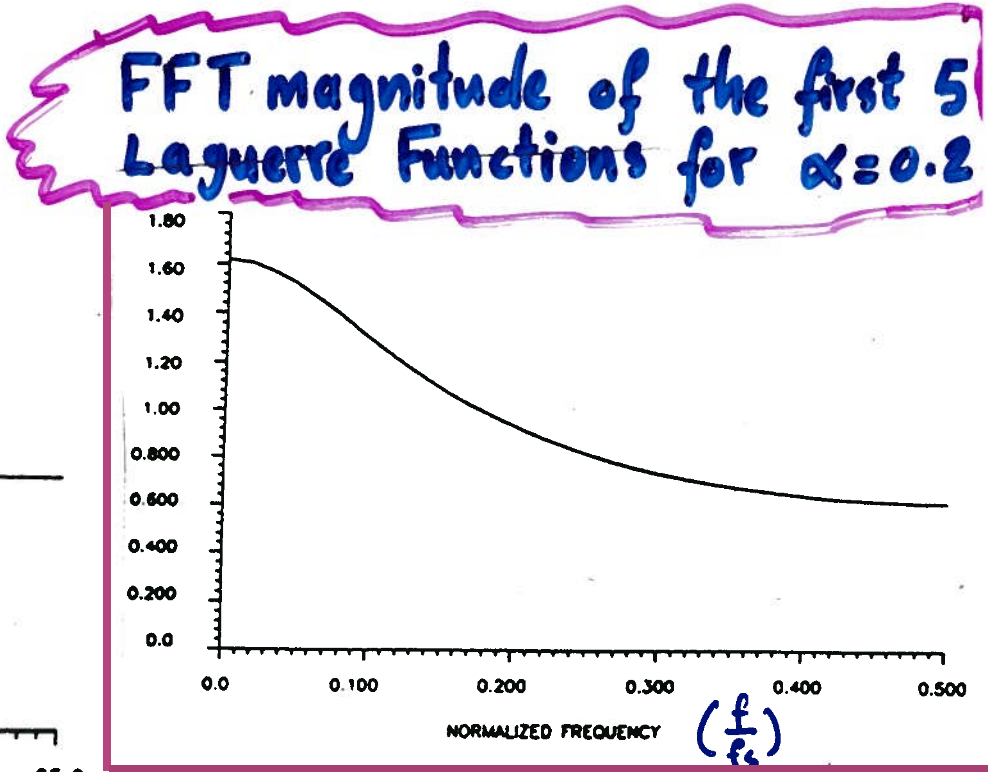
discrete-time Laguerre parameter

♫ $\binom{j}{k} = \frac{j!}{(j-k)!\,k!}$ for $k \leq j$

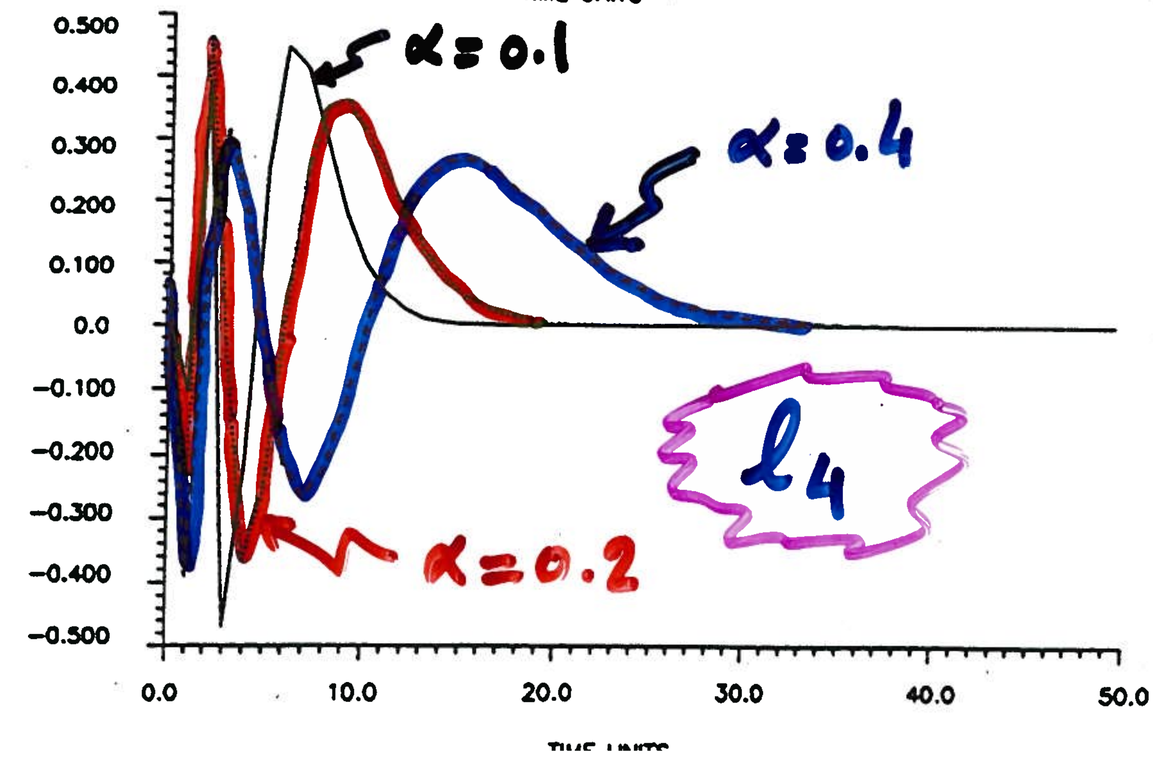
$$\boxed{b_j(m) = l_{j-1}(m) \; ; \; j = 1, \dots L}$$



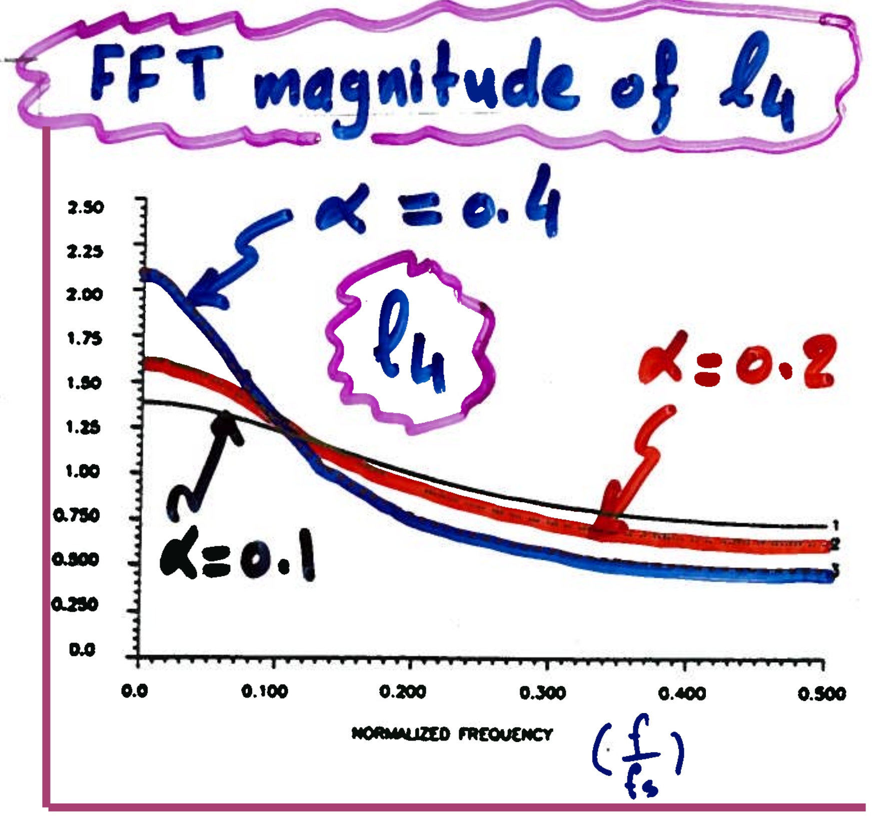
0.900
0.750
0.600
0.450
0.300
0.150
0.0
−0.150
−0.300
−0.450
−0.600
0.0
5.00
10.0
15.0
20.0
25.0
TIME UNITS
l_0
l_1
l_2
l_3
l_4
$\alpha = 0.2$



FFT magnitude of the first 5 Laguerre Functions for $\alpha = 0.2$
1.80
1.60
1.40
1.20
1.00
0.800
0.600
0.400
0.200
0.0
0.0
0.100
0.200
0.300
0.400
0.500
NORMALIZED FREQUENCY
$\left(\frac{f}{f_s}\right)$

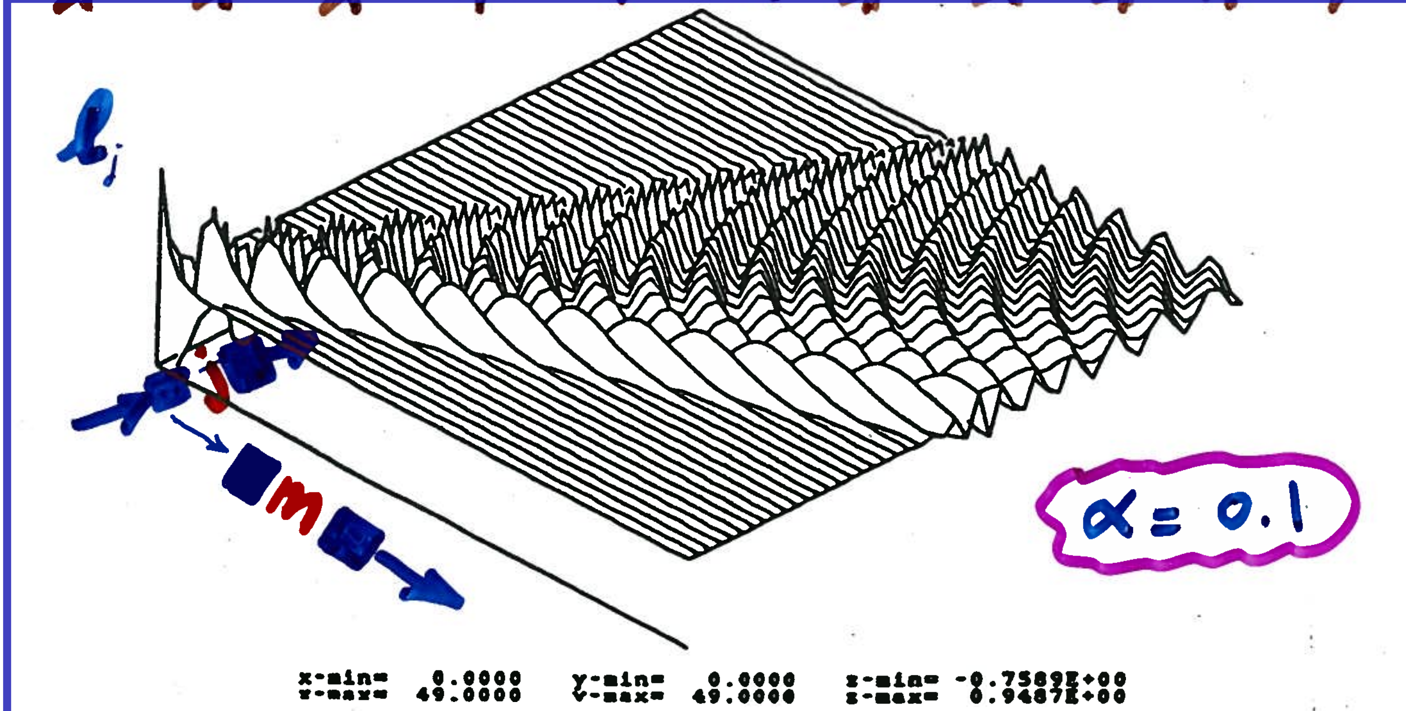


0.500
0.400
0.300
0.200
0.100
0.0
−0.100
−0.200
−0.300
−0.400
−0.500
0.0
10.0
20.0
30.0
40.0
50.0
$\alpha = 0.1$
$\alpha = 0.4$
$\alpha = 0.2$
l_4

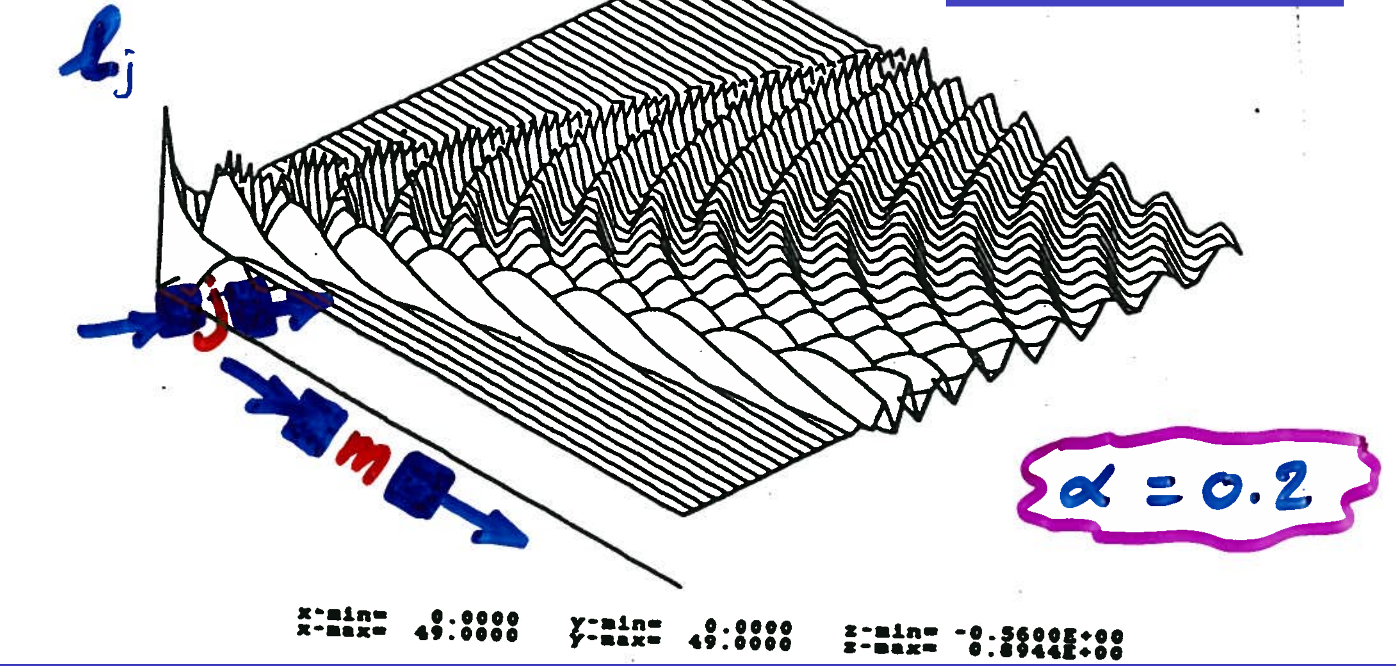


FFT magnitude of l_4
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.750
0.500
0.250
0.0
0.0
0.100
0.200
0.300
0.400
0.500
NORMALIZED FREQUENCY
$\left(\frac{f}{f_s}\right)$
$\alpha = 0.4$
$\alpha = 0.2$
$\alpha = 0.1$
l_4
1
2
3

The first 50 Laguerre Functions, plotted as a square matrix over 50 time units (0-49 lag)



j ≡ order
m ≡ time



Choice of L

- The number of Laguerre functions L must be much smaller than the memory-bandwidth product M of the system.

$$\boxed{L \ll M}$$

- e.g., $M = 100 \Rightarrow L = 10$

Practical Hints for LET

1. Inject $x(t) = \text{GWN}$ and estimate response $y(t)$.

2. $\text{FFT}\{y(t)\} \Rightarrow$ System Bandwidth

3. Estimate first order kernel $k_1(\tau)$ by:
   a) Cross correlation of $x(t)$ & $y(t)$.
   or b) $\mathcal{F}^{-1}\left\{\dfrac{\mathcal{F}[y(t)]}{\mathcal{F}[x(t)]}\right\}$

4. Duration of $k_1(\tau) \Rightarrow$ System Memory

5. Estimate number of Laguerre Functions
   $L \ll$ Memory-Bandwidth product

6. Estimate α to cover system memory.

ΛΥΣΙΣ

LYSIS

Nonlinear Biomedical Systems Analysis

LYSIS (the Greek word for "solution") is an interactive software of a set of modular programs (each performing a specific task) that provide an integrated computing environment for data analysis and system modeling. Unique capabilities of LYSIS include nonlinear system modeling and its relation to Artificial Neural Networks, as well as the novel methodology of "principal dynamic modes". This package has expanded through time to incorporate emerging methodologies developed by the Biomedical Simulations Resource (BMSR) at USC, under the supervision of Prof. V.Z. Marmarelis.

LYSIS 7.1 is currently available to run on Windows 95/98/NT/2000/XP. It is also available for UNIX, distributed as source code to be compiled for each UNIX implementation (e.g., Solaris, HPUX, Linux, etc). It is packed with all the necessary files to make all necessary modifications to the Windows 9x/NT/2000/XP environment. It requires minimal changes in environment (setting a few variables). This installation creates the LYSIS 7.1 Group (as well as ten subgroups) off the Programs group at the Start button.

Specific features cannot be found in other commercially available packages. For instance, efficient kernel estimation using Laguerre expansions or modeling of nonlinear dynamic systems using artificial neural networks with polynomial activation functions are unique to LYSIS. Along with the basic programs (including graphics), they constitute a powerful toolset for time-series data analysis and system modeling indispensable to cutting-edge research.

- **Accurate estimation of Volterra kernals from short date-records and associated model predictions**
- **Combined use of Volterra kernels and artificial neural networks for nonlinear system modeling**
- **Principal Dynamic Mode analysis of nonlinear systems**

http://bmsr.usc.edu/Software/Lysis/

Requirements

PC/486, Pentium

<u>Operating Systems</u>

Windows 95/98/NT/2000/XP

Sun/Unix

<u>Operating Systems</u>

Solaris 2.x

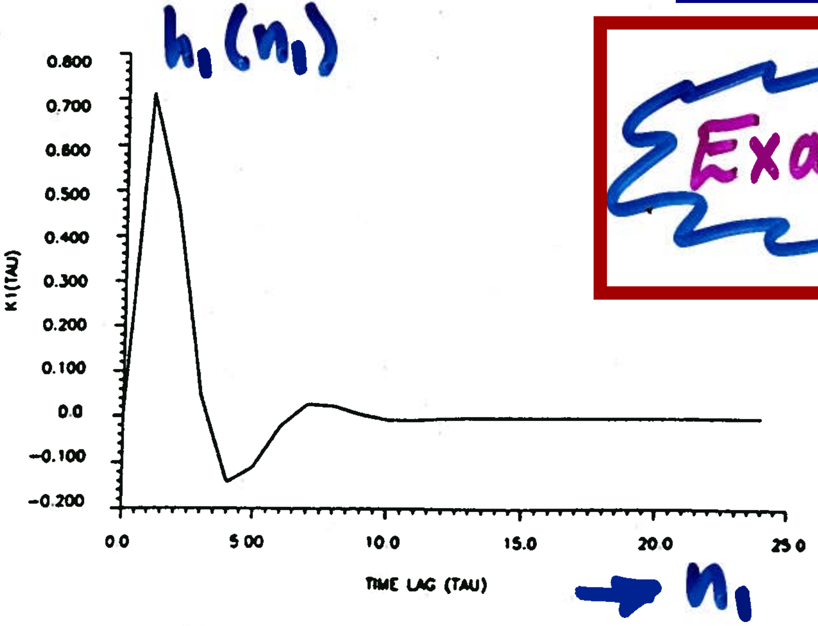
Minimum system requirements to run LYSIS 7.1:

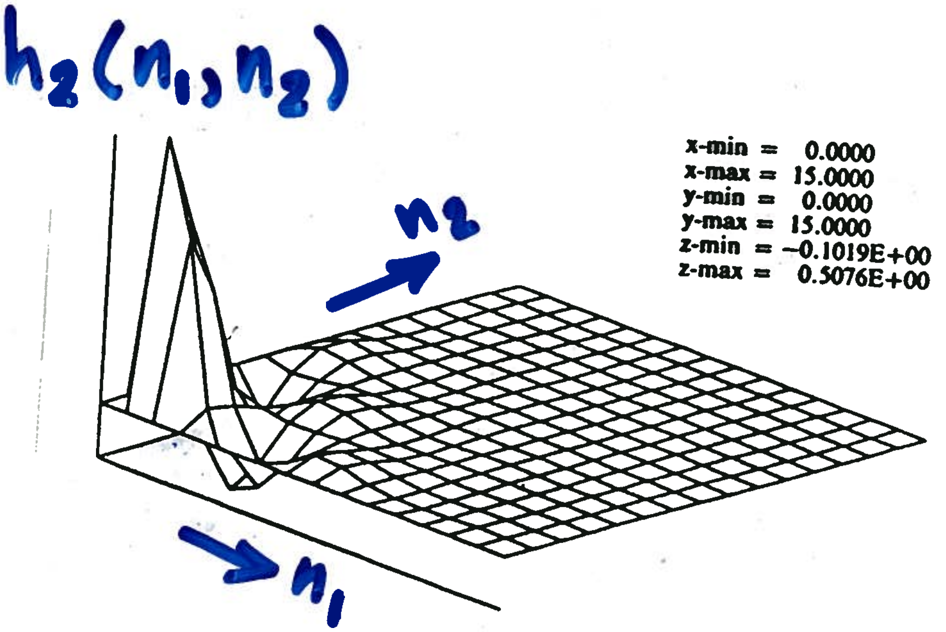
- Intel P5 (or better) based computer
- 256 Mbytes of RAM or more
- MS-Windows 95/98/NT/2000/XP
- Disk space of at least 35 Mbytes
- 1 GB pagefile
- video card capable of 256 colors or better

Exact Kernels

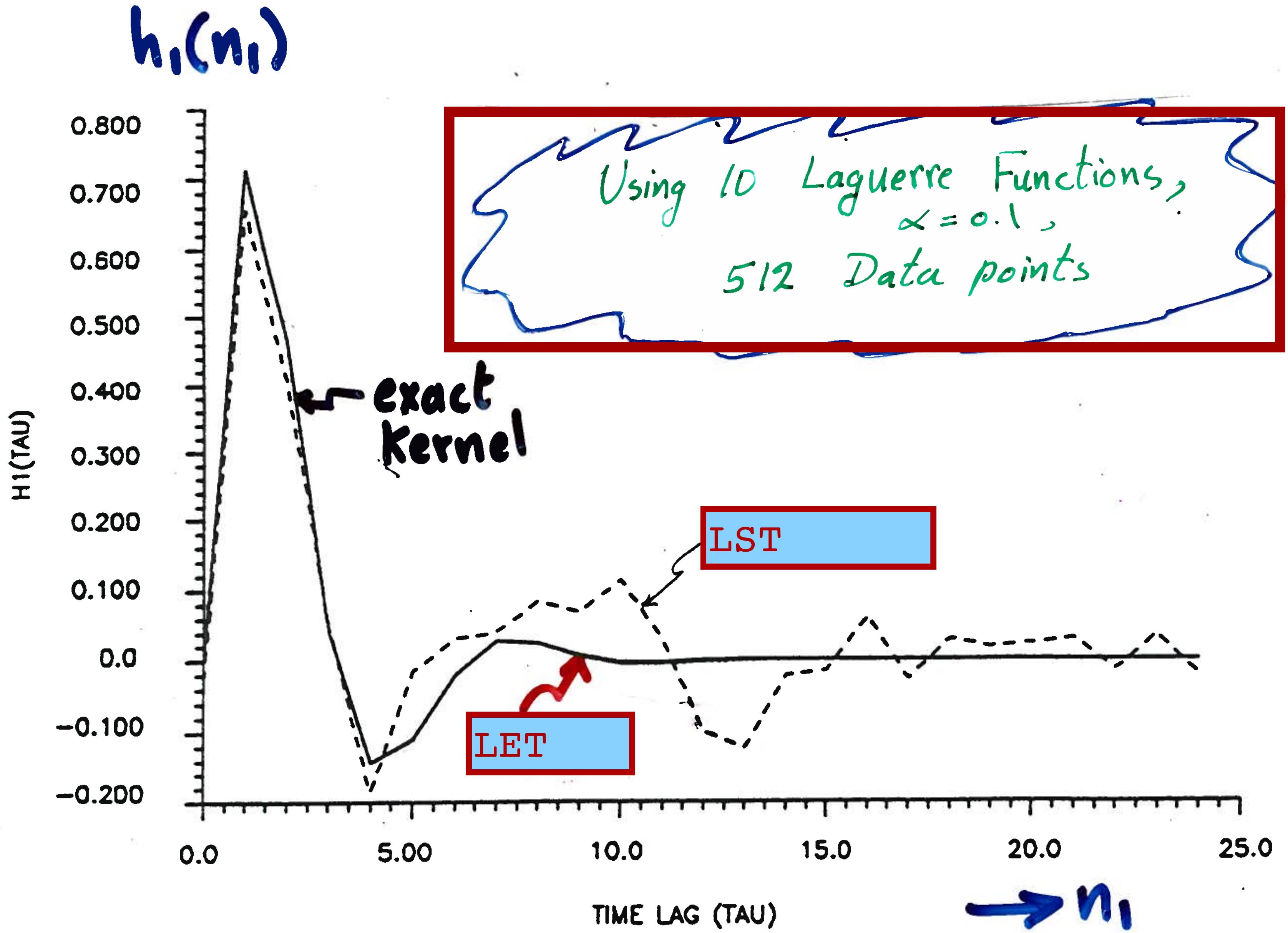
$q = 2$

Example





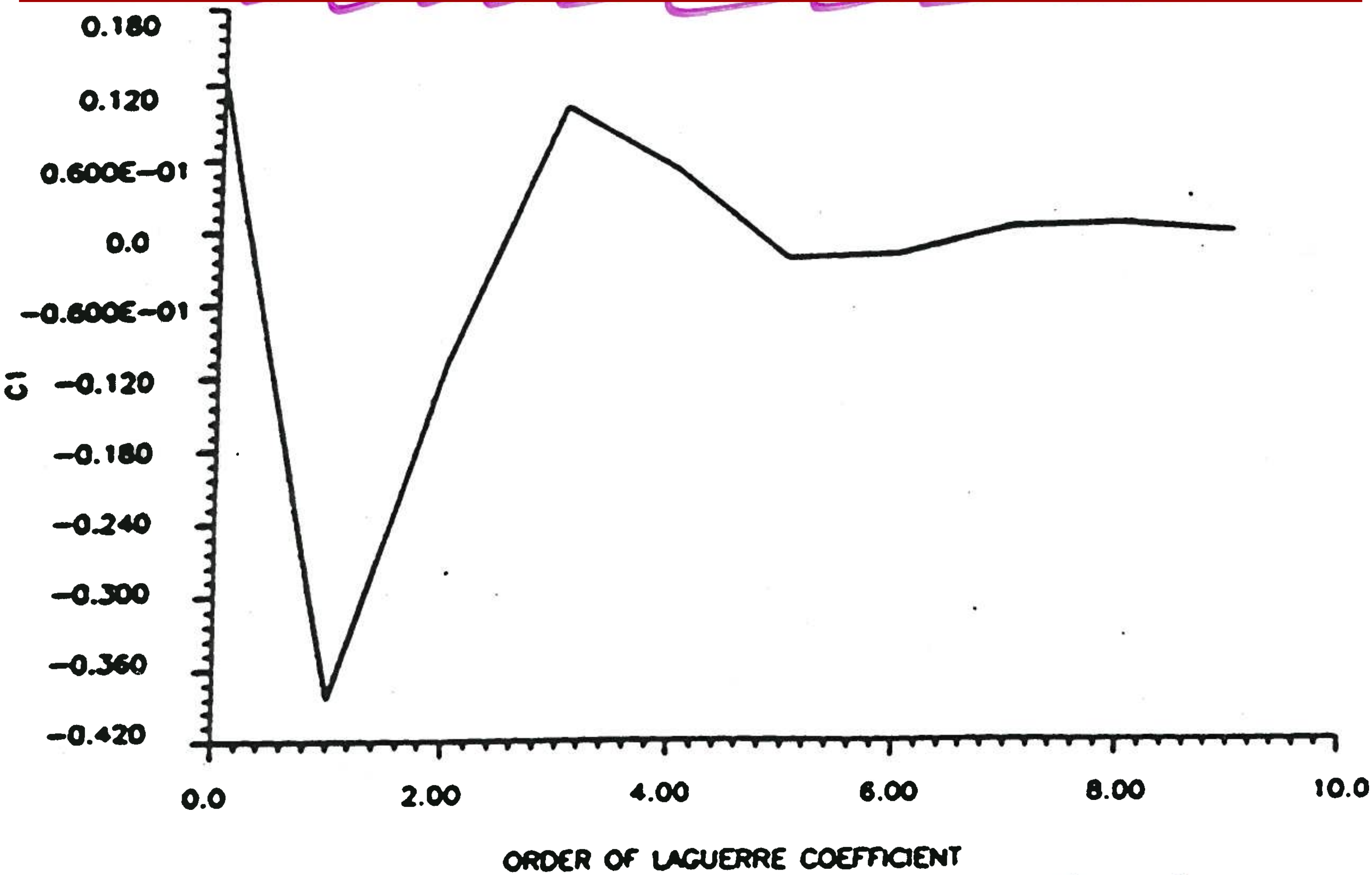
The model output is produced by convolution of GWN with the system kernels



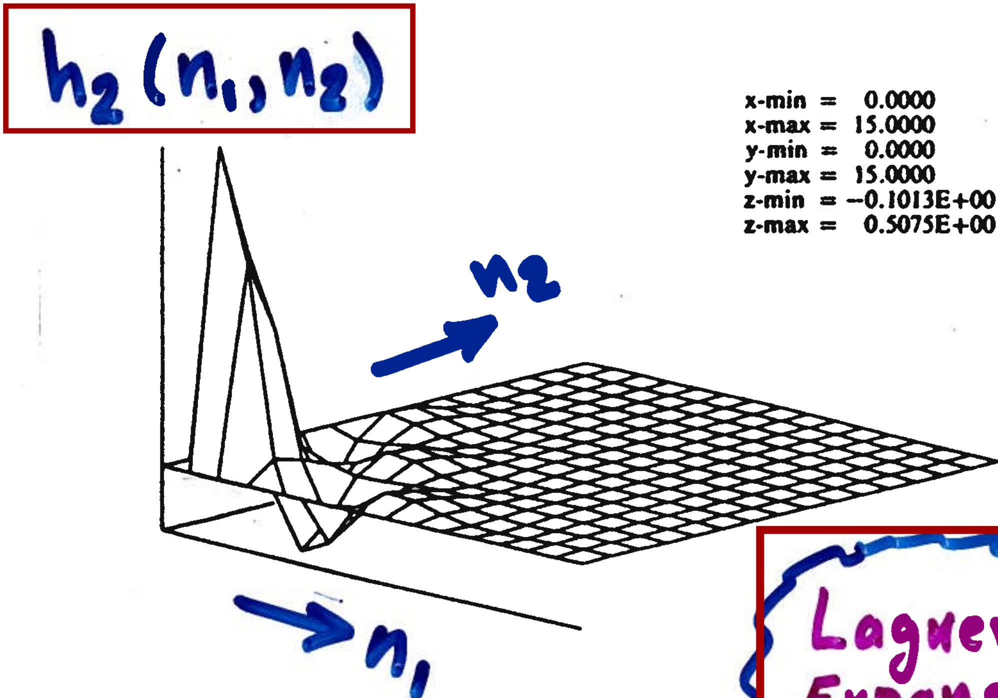
LST ≜ Lee-Schetzen Technique

LET ≜ Laguerre Expansion Technique

Estimates of the 10 Laguerre Expansion Coefficients for the first-order Kernel

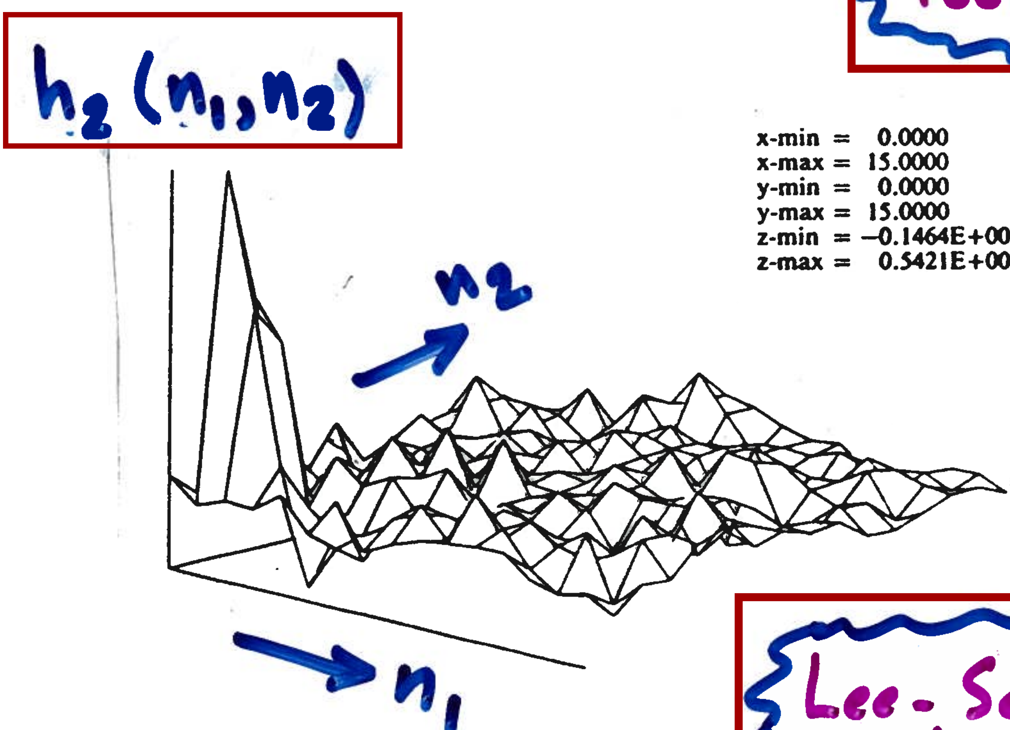


The required L can be determined by estimating the first-order kernel for a large L, then inspect coeff^t estimates to select minimum number for signifiant values.



Laguerre Expansion Technique

LET



LST

Lee-Schetzen Technique