Lesson 3: Fading Memory Nonlinearities

Nonlinear Signal Processing – SS 2017

Christian Knoll
Signal Processing and Speech Communication Laboratory
Graz University of Technology

May 19, 2017
Session contents

▶ Today:
  ▶ Volterra Series as representation of fading-memory NL
  ▶ System identification using different inputs
  ▶ Time series modelling (Homework)

▶ Next time:
  ▶ Higher-order statistics and spectral analysis
Volterra series – Definition

- A finite Volterra series of order $p$ and memory length $M$:

$$y[n] = h_0 + \sum_{m_1=0}^{M-1} h_1[m_1] x[n - m_1]$$

$$+ \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} h_2[m_1, m_2] x[n - m_1] x[n - m_2] +$$

$$+ \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} \sum_{m_3=0}^{M-1} h_3[m_1, m_2, m_3] x[n - m_1] x[n - m_2] x[n - m_3] + \cdots +$$

$$+ \sum_{m_1=0}^{M-1} \cdots \sum_{m_p=0}^{M-1} h_p[m_1, \ldots, m_p] \prod_{i=1}^{p} x[n - m_i]$$

- Universal approximator for time-invariant causal operators with fading memory (for bounded input signals)
Volterra series – System identification (1)

- Only term with $h_1[\cdot]$ is linear!
- Complexity grows exponentially with $M^P$
- Choice of input signal for system identification?
- Remember RBF-fits: Model also linear in coefficients $\rightarrow$ Least-squares fit easy

$$y[n] = \sum_k \alpha_k \phi_k(x[n]) \quad \text{N equations}$$

- Arrange in equation system

$$\begin{bmatrix} y[0] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} \phi_1(x[0]) & \cdots & \phi_K(x[0]) \\ \vdots & \ddots & \vdots \\ \phi_1(x[N-1]) & \cdots & \phi_K(x[N-1]) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_K \end{bmatrix}$$

$$\Phi, \ (N \times K), \ N \gg K$$
Here we use the same method, just different basis functions

Second order Volterra system as example ($1 + M + M^2$ coeff.)

You need “quite some” data!

$$y[n] = h_0 + \sum_{m_1=0}^{M-1} h_1[m_1] x[n-m_1] + \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} h_2[m_1, m_2] x[n-m_1] x[n-m_2]$$

Basis vectors/functions are the data products
Volterra series – Matlab files

- Function `vkernels.m` does the LS-fit (use `p3_1.m` as tutorial)
- Nonlinearities defined in `nlsystem1.m`

\[
y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n]^2 + a_3 x[n] x[n-1]
\]

- Output of `vkernels.m` for this nonlinearity:

\[
H\{1\} = 0
\]
\[
H\{2\} = \begin{bmatrix} a_0 & a_1 \end{bmatrix}^T
\]
\[
H\{3\} = \begin{bmatrix}
x[n]^2 & x[n] x[n-1] \\
a_2 & a_3 \\
a_3 & 0 \\
x[n-1] x[n] & x[n-1]^2
\end{bmatrix}
\]

- Or optionally a structure `Vmodel` that can be directly passed to `vkernels_o.m` for Problem 3.2
Volterra series – Time series modelling/forecasting

- Given a correlated (not necessarily just second order!) time series \( s = [s_1, \ldots, s_N]^T \)
- Current sample \( s_n \) depends on past samples
- Volterra series one way to model the dependence
- Input \( x \) and output \( y \) both generated from \( s \)
- Partitioning of \( s \) into training and validation sequences

→ Homework
Higher order statistics and spectral analysis

- We are used to first- and second-order statistics
  - Mean: $\mu = E\{x[n]\}$
  - ACF: $m_2[k] = E\{x[n]x[n+k]\}$
- This is suitable as long as we deal with linear systems, e.g.
  $$y[n] = \sum_{k=0}^{K} x[k]h[n-k]$$
- Standard example for a random process: *Linear process*, i.e. $x[k]$ is a white-noise process driving a linear system
- But what if our system is nonlinear, e.g.
  $$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n]^2 + a_3x[n]x[n-1]$$
Higher order statistics and spectral analysis - Example

- Linear system w. $K = 10$, driven by white noise
Higher order statistics and spectral analysis - Example

- Linear system w. $K = 10$, driven by white noise

- Support of non-zero ACF indication for memory
Higher order statistics and spectral analysis - Example

- Linear system w. $K = 10$, driven by white noise

- Support of non-zero ACF indication for memory
- How do you evaluate nonlinear combinations of input samples?
Higher order statistics and spectral analysis - Example

- Linear system $w. K = 10$, driven by white noise

- Support of non-zero ACF indication for memory
- How do you evaluate nonlinear combinations of input samples?
  - Evaluate HOS, e.g. third-order $E\{x[n]x[n + k]x[n + l]\}$
HOSA – Definitions

- Generalization of ACF: Non-central moments of order $r$ for stationary process $x[n]$
  $$m_r[k_1, \ldots, k_{r-1}] = E\{x[n]x[n + k_1] \cdots x[n + k_{r-1}]\}$$

- Cumulant of order $r$ again defined over characteristic function

- Can be expressed and estimated via moments

- Fourier transform of ACF is power spectral density

- Fourier transform (2D) of third order cumulant is the *Bispectrum*
  $$c_3^x[k_1, k_2] \xrightarrow{\text{DFT}} C_3^x[\omega_1, \omega_2]$$
HOSA – Important Properties

- $x[n]$ Gaussian:
  \[ c_r^x(k_1, \ldots, k_{r-1}) = C_r^x(\omega_1, \ldots, \omega_{r-1}) = 0 \text{ for } r > 2 \]
- $x[n]$ i.i.d.:
  \begin{align*}
  c_r^x(k_1, \ldots, k_{r-1}) &= a \cdot \delta(k_1, \ldots, k_{r-1}) \\
  C_r^x(\omega_1, \ldots, \omega_{r-1}) &= a
  \end{align*}
- $x[n]$ symmetrically distributed around zero:
  \[ c_r^x(k_1, \ldots, k_{r-1}) = C_r^x(\omega_1, \ldots, \omega_{r-1}) = 0 \text{ for } r = 0, 3, 5, 7, \ldots \]
- $z[n] = x[n] + y[n]$, where $x[n]$, $y[n]$ jointly stationary and statistically independent:
  \begin{align*}
  c_r^z(\cdot) &= c_r^x(\cdot) + c_r^y(\cdot) \\
  C_r^z(\cdot) &= C_r^x(\cdot) + C_r^y(\cdot)
  \end{align*}
HOSA – Pros and Cons

Pros

- Analysis of nonlinearities
- Cumulants are additive for independent processes
- Gaussian noise: HOS zero (blind to Gaussian noise)

Cons

- Difficult to estimate from finite length data
- Influence of window
- Once you have them, how do you interpret them?
HOSA – Example

Example for a third-order cumulant
HOSA – Example

- Example for a Bispectrum
HOSA – Example

- Example for a PSD
HOSA – Matlab (1)

- HOSA toolbox, free, included in download-file
- HOSA toolbox manual is a great resource (Matlab-central)
- You will need:
  - `cumest.m` used to estimate cumulants
  - `rpiid.m` used to help generating input processes
  - `gabrrao.m` used to calculate window for 2D-FFT
  - `viscumul3.m` and
  - `visbispec3.m` for visualization
HOSA – Matlab (2)

- `cumest.m` for third order cumulant calculates just one slice of the 2D correlation function
  
  ```matlab
  for k = -MaxLag : MaxLag
      c3(:, k+MaxLag+1) = cumest(#, #, #, #, #, #, #, k);
  end
  ```

- Bispectrum calculation: use `fftshift(fft2( c3 .* w ))` to have a familiar picture

- Window `w[n]` obtained from `gabrrao.m`, optimal smoothing window, minimum bias in estimation
Higher order statistics and spectral analysis - Problems (1)

- Limited amount of data leads to higher variance of estimators
- Problem even for ACF $\rightarrow$ Grows exponentially with order
- This makes visual interpretation much harder:
  - When is a Bispectrum zero?
  - When is a Bicoherence function flat?
Higher order statistics and spectral analysis - Problems (2)

- Linear system w. $K = 10$, driven by white noise (500 samples)
Higher order statistics and spectral analysis - Problems (2)

- Linear system w. $K = 10$, driven by white noise (500 samples)
Higher order statistics and spectral analysis - Problems (2)

- Linear system w. $K = 10$, driven by white noise (500 samples)
Higher order statistics and spectral analysis - Problems (2)

- Linear system w. $K = 10$, driven by white noise (500 samples)