Nonlinear System Theory

Now let the $x_k(t)$'s represent inputs at $k$ prior times, spaced by $\omega t$:

$$R(t) = a_0 + \sum_{k} a_k s(t - k\omega t) + \sum_{k} b_k (s(t - k\omega t)^2 - \rho)$$

$$+ \sum_{k < l} c_{kl} s(t - k\omega t) s(t - l\omega t)$$

$$+ \cdots$$

where $\rho = \langle s(t)^2 \rangle$.

The $\rho$-term in $b_{\cdots}$ comes from the fact that $\langle s(t - k\omega t)^2 \rangle \neq 0$,

so the $b$-term would otherwise contribute a constant, confounding the $a_0$-term.

Does this make sense as $\omega t \to 0$?

$s(t)$ gets sampled more and more densely. But expect that the influence of $s(t - k\omega t)$ on $R(t)$ depends on $\omega$, not the lag number. i.e.,

$$\lim_{\omega \to 0} a_k = \omega t A(k \omega t).$$

Think $s(t) = \frac{1}{\omega t}$, resample each $s(t)$.

Similarly, there are $(\omega t)^2$ - a more complex for $c_{kl}$, so expect that

$$c_{kl} \approx (\omega t)^2 C(k \omega t, l \omega t).$$

$$b_k \approx (\omega t)^2 b(k \omega t)$$

for dimensional correctness.
This notation: \( K_0 = a_0 \)

\( K_1(\gamma) = A(\gamma) \)

\[
K_2(\gamma_1, \gamma_2) = \begin{cases} 
\frac{1}{2} c(\gamma_1, \gamma_2) & \gamma_1 < \gamma_2 \\
\frac{1}{2} c(\gamma_2, \gamma_1) & \gamma_1 > \gamma_2 \\
b(\gamma_1) & \gamma_1 = \gamma_2 
\end{cases}
\]

\[
R(t) = K_0 + \sum_{k} \left[ K_1(k\alpha t) s(t - k\alpha t) \right] \alpha t
\]

\[
+ \sum_{k} K_2(k\alpha t, k\alpha t) \left( s(t - k\alpha t) s(t) - P \right) \alpha t^2 - \sum_{k \neq 1} K_2(k\alpha t, t - k\alpha t) s(t - k\alpha t) s(t - t - k\alpha t) \alpha t^2 + \ldots
\]

\[
= K_0 + \int K_1(\gamma) s(t - \gamma) d\gamma
\]

\[
+ \int K_2(\gamma_1, \gamma_2) s(t - \gamma_1) s(t - \gamma_2) d\gamma, d\gamma_2
\]

\[
- \Delta t \int K_2(\gamma, \gamma) d\gamma
\]

Note what we expected. So we have to think of \( s(t) \) as being a process defined by a constant \( \mu \), i.e., std dev of \( s(t) \) is

\[
\sqrt{\mu t}.
\]

Then we'll get a well-defined limit.

If we keep \( \mu = \mu t \) constant, i.e., \( \mu = \frac{\mu}{t} \) on any realization.
Calculate the $a_k$'s (or the $k$'s).

In general, $a_k = \frac{\langle f(x) \phi_k(x) \rangle}{\langle \phi_k(x)^2 \rangle}$.  
[Math units don't consist]

where $\phi_k(x)$ is an orthogonal polynomial.

For $p_k$, $\phi_k(x)^2 = x_k$, then $a_k = \frac{\langle f(x) x_k \phi_k(x) \rangle}{\langle \phi_k(x)^2 \rangle} = \frac{1}{p} \langle f(x) x_k \phi_k(x) \rangle$

In the continuum case, $\langle \phi_k^2 \rangle = A \cdot p^2 = p$

So, $k<\ell$, $\phi_k(x) = x_k x_\ell$, $a_k = \frac{\langle f(x) x_k x_\ell \phi_k(x) \rangle}{\langle \phi_k(x)^2 \rangle} = \frac{1}{p^2} \langle f(x) x_k x_\ell \phi_k(x) \rangle$

$\langle \phi_k^2 \rangle = (x + p)^2 = p^2$

$k<\ell$, $p^2 = \frac{1}{2} p^2 \langle R(t) \phi_k(x)^2 \phi(k+1) \phi(t-\frac{1}{2}) \rangle$

from $k<\ell$, $k<\ell$.

$b_k : \phi_k(x^2) = x_k^2 - p$, $\langle \phi_k^2 \rangle = \langle x_k^4 - 2 p x_k^2 + p^2 \rangle = 2p^2$

$b_k = \frac{1}{2p^2} \langle f(x) \phi_k(x^2) \phi(k-\frac{1}{2}) \rangle$
\[ k_2(\tau_1, \tau_2) = \frac{1}{2} \rho^2 \left[ R(t) \left( s(t-\tau_1) s(t-\tau_2) - s(\tau_1-\tau_2) \rho \right) \right] \]

Unity \( k_2(\tau, \tau) \) for \( \tau = \tau_1, \tau_2 \) and \( \tau = \tau_2 \).

The derivative with respect to \( \rho \) is
\[
\frac{d}{d\rho} \left( k_2(\tau_1, \tau_2) \right) = \frac{1}{2} \rho \left( R(t) \right) \frac{d}{d\rho} \left( s(t-\tau_1) s(t-\tau_2) - s(\tau_1-\tau_2) \rho \right)
\]

The subprincipal term integrates to \( k_2(\tau_1, \tau_2) \) (it is 0 except near windows at \( \tau \), where it is \( \frac{d}{d\rho} k_2(\tau_1, \tau_2) \)).

So we conclude
\[
k_2(\tau_1, \tau_2) = \frac{1}{2} \rho^2 \left( R(t) \left( s(t-\tau_1) s(t-\tau_2) - s(\tau_1-\tau_2) \rho \right) \right)
\]

At another way, to calculate \( k_2(\tau_1, \tau_2) \), we cross correlate with \( s(t_0-\tau_1) s(t_0-\tau_2) - s(\tau_1-\tau_2) \rho \) for orthogonality.

To reconstitute the input
\[
R(t) = k_0 + \int k_1(\tau) s(t-\tau) d\tau + \int k_2(\tau_1, \tau_2) \left( s(t-\tau_1) s(t-\tau_2) - \rho s(\tau_1-\tau_2) \right) d\tau_1 d\tau_2 + \ldots
\]
The issue is one of a predicted interaction.

Choose DT: \( P = \frac{1}{\alpha} \).

Choose maximum time lag for analysis.

Estimate \( \langle R(s) \cdot s(t-\tau) \rangle \) from finite samples of \( s \).

\[
\langle R(s) \rangle (s(t-\tau) s(t-\tau))
\]

Two main issues: samples of \( s \) may not be typical of the Gaussian noise, and therefore:

the orthogonal factor \( s(t-\tau)^2 \) \( s(t-\tau) s(t-\tau) \) \( s(t-\tau)^2 \) \( - P \),

\( s(t-\tau)^2 - sP s(t-\tau) \), etc.

may not

be orthogonal w.r.t. the sample of noise.

So it is really \( R = \Pi_{\phi, s}(s) \) but

if the estimate \( \langle \Pi_{\phi, s}(s) \Pi_{\phi, s}(s) \rangle \neq 0 \) then

\( R \) will appear to have a component \( \Pi_{\phi, s}(s) \).

So, can we choose specific samples to make the orthogonality as close as possible?

Broaden the issue: choose some other ensemble \( R \) of signals
(not necessarily Gaussian white noise).

\( \phi \) construct one of several formal series
\( \phi \) design input to sample \( R \)
We could do this with a sequence of RE's that approach Gaussian white noise.

And, from a systemic viewpoint, on, from the systemic point of view,

we could even choose R based on a biological motivation - natural scenes, natural sounds.

Reorder to regressions (functional imaging analysis)

We have \( R(t) \) as several variables \( u_1(t), \ldots, u_n(t) \)

and we'd model \( R(t) \) as a sum of "effects",

\[
R(t) = \sum_{k=1}^{n} \sum_{i=1}^{n} \alpha_k(t) v_k(t + \tau_i).
\]

or even \( \sum_{k=1}^{n} \sum_{l=1}^{n} \beta_k(\tau_1, \tau_2) [v_k(t - \tau_1) v_l(t - \tau_2)] \)

\[
= \sum_{h} c_h v_h(t).
\]

We'd to orthogonalize the \( v_h's \) into \( \psi_h's \).

Best if they were orthogonal, but if they are linearly dependent
Two strategies for "a sequence of R's 4th approach GWN"

- M-sequences: "pseudorandom binary sequences"
- Sum of sinusoids: \( \text{Eq. cos}(\omega t + \phi) \)

M-sequences approach GWN from the "system's point of view".

Any real system has a first and fast sum over-time,

\[
\text{turing } s(t) \text{ into } s'(t) = \int f(\tau) s(t - \tau) \, d\tau
\]

So even if \( s(t) \) is only 0's or 1's, \( s'(t) \) is not Gaussian.

Basic idea (break M-sequences)

Say we have 3 time lags: 8 possible stimuli histories

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We could do 8 experiments, present each one several times (fusing)

Or,

\[
\text{0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 ...}
\]

Note that past histories contain each of the 3-bit inputs, but we gained 3^x in efficiency.

Basic idea is to choose a sequence of 0's and 1's for which the 1's are nearly orthogonal.
We will construct a sequence of length $2^n - 1$ which

(a) contains all $n$-tuples except $(0 \ldots 0)$ exactly once

(b) is $n$-shift-orthogonal, i.e., $<d(k) \sigma(k-1)> = \frac{1}{2^n - 1}$

(c) will allow for measurement of high-order kernels at short lags

Recipe: We will create a finite field of size $2^n$, and write a table of algorithms with $n = 4$.

Consider the finite field $\mathbb{F}_2 = \{0, 1\}$, and let's choose a polynomial

of degree $n$ that does not factor in $\mathbb{F}_2$ (other conditions too).

$p(x) = x^4 + x + 1$. This construct a finite field as follows: Set $p(x) = 0$, multiply!

\[
\begin{align*}
  x^0 & = 1 \\
  x^1 & = x \\
  x^2 & = x^2 \\
  x^3 & = x^3 \\
  x^4 & = x^4 + x \\
  x^5 & = x^5 + x^2 + x \\
  x^6 & = x^6 + x^3 + x^2 \\
  x^7 & = x^7 + x^3 + x^4 + 1 \\
  x^8 & = x^8 + x^2 + 1 \\
  x^9 & = x^9 + x \\
  x^{10} & = x^{10} + x + 1 \\
  x^{11} & = x^{11} + x^3 + x^2 + x \\
  x^{12} & = x^{12} + x^3 + x^2 + 1 \\
  x^{13} & = x^{13} + x^3 + x^2 + 1 \\
  x^{14} & = x^{14} + x^3 + 1 \\
  x^{15} & = x^{15} + 1
\end{align*}
\]

Then any column is the $n$-sequence.
A recurrence rule holds within a column:

\[ p(x) = x^n + g(x) \]

\[ x^k, x^n = x^k \cdot g(x) \]

\[ x^{k+n} = \sum_{r=0}^{n-1} b_r x^{k+r} \]

Here, \( b_0 = 1, b_1, b_2 = b_3 = 0 \)

So, for example:

\[ \begin{array}{c|c|c}
   k & b \neq 0 & x^5 = x^2 + x \\
   \hline
   k=5 & b=0 & x^5 = x^3 x^2 \\
   k=6 & b=1 & x^6 = x^3 + x + 1 \\
   k=7 & b=2 & x^7 = x^3 + x + 1 \\
   k=8 & b=3 & x^8 = x^3 + x \\
   k=9 & b=4 & x^9 = x^6 + x^5 \\
\end{array} \]

"Shift register" generation rule is the original rule, equivalent to this recursion.

If the sequence has its maximum length, then no \( n \)-tuples can repeat. (Otherwise it would close early.
And no \( 0 \)-tuple.
So all \( n \)-tuples appear once.

What about shift- or Pseudorandomness?

\[ a \neq 0: \quad x^k + x^k a = \begin{cases} 0 & \text{if match} \\ \text{if mismatch} & \end{cases} \]

So \( x^k + x^k a \) is cross correlat of \( \delta_k \) and \( \delta_k a \).

\[ x^k + x^k a = x^k (1 + x^a) = x^k x^{2a} \]

because \( 1 + x^a \) must be some \( \delta_k \).

\[ x^k x^{2a} = x^{k+2a} \]
The above shows that an m-sequence XOR'd with a shift of itself is just another shift of the same m-sequence.

So \( X^k \) and \( X^k + a \) are equal to \( \frac{1}{2} \) ones and zeros, so \( X^k - X^k + a \) are independent.

The drawback is that high-order registers (polynomials) overlay with low-order ones:

\[
\langle r(t), \sigma(t - \gamma), \sigma(t - \gamma_2) \rangle
\]

must be equal to

\[
\langle r(t), \sigma(t - \gamma), \sigma(t - \gamma_2) \rangle
\]

since \( \sigma(t - \gamma), \sigma(t - \gamma_2) = \sigma(t - \gamma_2(t, \gamma_2)) \)

by above argument.

What can we do? Choose \( p(x) \) so \( f(\gamma, \gamma_2) = (\gamma, \gamma) \).

Or, let \( R \) contain \( \delta_{\text{inc}} \) or \( \delta_{\text{incr}} \).

So that now, \( \sigma(t - \gamma) \) o \( \sigma(t - \gamma), \sigma(t - \gamma_2) \)

are orthogonal over \( R \).

("must repeat" method)