Nonlinear Systems Theory III

[29]

Now let the x_k's represent inputs at k prior times, spaced by Δt.

$$R(t) = a_0 + \sum_k a_k s(t - k\Delta t) + \sum_k b_k\left(s(t-k\Delta t)^2 - P\right)$$
$$+ \sum_{k<l} c_{kl}\, s(t-k\Delta t)\, s(t-l\Delta t)$$
$$+ \cdots$$

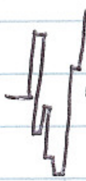
where $P = \langle s(t)^2 \rangle$.

The P-term in $b_k \cdots$ comes from the fact that $\langle s(t-k\Delta t)^2 \rangle \neq 0$,

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

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

$s(t)$ gets sampled more & more densely. But expect that the influence of $s(t-\tau)$ on $R(t)$ depends on τ, not the lag number. I.e.,

expect $a_k \to \Delta t\, A(k\Delta t)$. Think of $s(t) =$  , resample each Δt.

Similarly, there are $\left(\frac{1}{\Delta t}\right)^2$ - as many samples for c_{kl}, so expect that

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

$$b_k \to (\Delta t)^2 b(k\Delta t)$$

for dimensional correctness.

This motivates: $K_0 = a_0$

$$K_1(\tau) = A(\tau)$$

$$K_2(\tau_1, \tau_2) = \begin{cases} \frac{1}{2} c(\tau_1, \tau_2) & \tau_1 < \tau_2 \\ \frac{1}{2} c(\tau_2, \tau_1) & \tau_1 > \tau_2 \\ b(\tau_1) & \tau_1 = \tau_2 \end{cases}$$

$$R(t) = K_0 + \sum_k K_1(k\Delta t) s(t - k\Delta t)\, \Delta t$$
$$+ \sum_k K_2(k\Delta t, k\Delta t)\left(s(t - k\Delta t)^2 - P\right)(\Delta t)^2$$
$$+ \sum_{k \neq l} K_2(k\Delta t, l\Delta t)\left(s(t - k\Delta t) s(t - l\Delta t)\right)(\Delta t)^2 + \cdots$$

$$\rightarrow K_0 + \int K_1(\tau) s(t - \tau)\, d\tau$$
$$+ \int K_2(\tau_1, \tau_2) s(t - \tau_1) s(t - \tau_2)\, d\tau_1\, d\tau_2$$
$$- \Delta t P \int K_2(\tau, \tau)\, d\tau$$

Not what we expected. So we have to think of $s(t)$ as being a process definitely a constant $P\Delta t$, i.e., std dev. of $s(t)$ is $\sqrt{P\Delta t}$. Then we'll get a well-defined limit, if we keep $\tilde{P} = P\Delta t$ constant. i.e., $P = \frac{1}{\Delta t}\tilde{P}$ in any realization.

$\boxed{3}$ Calculating the a_k's (& the K's).

In general, $a_k = \dfrac{\langle f(x)\, \varphi_k^{\vec{\circ}}(x)\rangle}{\langle \varphi_k^{\vec{\circ}}(x)\, \varphi_k^{\vec{\circ}}(x)\rangle}$ [Multivariate discrete case]

where $\varphi_k^{\vec{\circ}}(x)$ is an orthogonal polynomial.

For a_k, $\varphi_k^{\vec{\circ}}(\vec{x}) = x_k$; $a_k = \dfrac{\langle f(x) \cdot x_k\rangle}{\langle x_k^2\rangle} = \dfrac{1}{P}\langle f(x) \cdot x_k\rangle$

In the continuum case, $\langle \varphi_k^2\rangle = \Delta t \cdot P = P$

so $K(\tau) = \dfrac{\langle R(t)\, s(t-\tau)\rangle}{P}$.

For c_{kl}, $\varphi_k^{\vec{\circ}}(\vec{x}) = x_k x_l$, $a_{kl} = \dfrac{\langle f(x)\, x_k x_l\rangle}{\langle x_k^2 x_l^2\rangle} = \dfrac{1}{P^2}\langle f(x)\, x_k x_l\rangle$

$\langle \varphi_k^{\vec{\circ}\,2}\rangle = (\Delta t\, P)^2 = P^2$

$K_2(\tau_1, \tau_2) = \dfrac{1}{2P^2}\langle R(t)\, s(t-\tau_1)\, s(t-\tau_2)\rangle$

from $k \neq l$ vs $k<l$.

b_k: $\varphi_k(\vec{x}) = x_k^2 - P$. $\langle \varphi_k^2\rangle = \langle x_k^4 - 2P x_k^2 + P^2\rangle = 2P^2$.

$b_k = \dfrac{1}{2P^2}\langle f(x_k)\,(x_k^2 - P)\rangle$

37

$$K_2(\tau,\tau) = \frac{1}{2P^2}\langle R(t)\,[s(t-\tau)^2 - P]\rangle$$

Unify $K_2(\tau_1,\tau_2)$ for $\tau_1 \neq \tau_2$ & $\tau_1 = \tau_2$.

The difference is whether we subtract $P = \frac{1}{\Delta t}P$.

The subtracted term integrates to P (it is 0 except on a window Δt, where it is $\frac{1}{\Delta t}P$)

So we can write

$$K_2(\tau_1,\tau_2) = \frac{1}{2P^2}\langle R(t)\,(s(t-\tau_1)s(t-\tau_2) - \delta(\tau_1-\tau_2)P)\rangle$$

Put another way, to calculate $K_2(\tau_1,\tau_2)$, we cross correlate with $s(t_0-\tau_1)s(t-\tau_2) - \delta(\tau_1-\tau_2)$

$\uparrow$
for orthogonality.

To reconstruct the input,

$$R(t) = K_0 + \int K_1(\tau)\,s(t-\tau)\,d\tau +$$

$$+\int K_2(\tau_1,\tau_2)\,(s(t-\tau_1)s(t-\tau_2) - P\delta(\tau_1-\tau_2)\,d\tau_1 d\tau_2$$

$+ \cdots$

531

The issues that arise in a practical implementation:

Choose ΔT. Choose $P = \frac{1}{\Delta T} \mathcal{P}$.

Choose maximum time lag for analysis.

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

$$\langle R(s) \left(s(t-\tau_1) s(t-\tau_2) \right) \rangle$$

Two Main issues: samples of s may not be typical of the Gaussian noise, and therefore,

the orthogonal functions $s(t-\tau)$; $s(t_1-\tau_1)s(t-\tau_2)$; $s(t-\tau)^2 - P$,

$s(t-\tau_1)^3 - 3Ps(t-\tau_1)$, etc may not

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

So if, really, $R = \varphi_k'(s)$ but

if the estimate of $\langle \varphi_k(s) \varphi_\ell(s) \rangle \neq 0$
then
R will appear to have a component of $\varphi_\ell(s)$.

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

Broaden the issue: ① choose some other ensemble $\mathcal{R}$ of signals (not necessarily gaussian white noise).
② construct an orthogonal functional series
③ design inputs to sample $\mathcal{R}$

We could do this with a sequence of R's that approach Gaussian white noise -
either analytically, or, from the systems point of view

or

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

Relation to regression (functional imaging analysis)

We have $R(t)$ & several variables $v_1(t), \ldots, v_n(t)$

and want to model $R(t)$ as a sum of "effects":

$$R(t) = \sum_{k=1}^{n} \sum_{\tau} \alpha_k(\tau) V_k(t-\tau).$$

or even $$\sum_{\substack{k=1 \\ l=1}}^{n\,n} \sum_{\tau_1,\tau_2} \beta_k(\tau_1,\tau_2)\left[V_k(t-\tau_1)V_k(t-\tau_2)\right]$$

$$= \sum_{\vec{h}} c_{\vec{h}} V_{\vec{h}}(t).$$

Want to orthogonalize the $V_{\vec{h}}$'s into $\varphi_{\vec{h}}$'s.

Best if they were already orthogonal, bad if they are linearly dependent

Two strategies for "a sequence of R's that approach GWN"

m-sequences · "pseudorandom binary sequences"
sum of sinusoids $\Sigma a_i \cos(\omega_i t + \varphi_i)$

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

any real system has a front end that sums over time,

turning $s(t)$ into $s'(t) = \int f(\tau)\, s(t-\tau)\, d\tau$

So even if $s(t)$ is only 0's & 1's, $s'(t)$ is ~ Gaussian.

Basic idea (but not M-sequences)

Say we have 3 time lags: 8 possible stimulus histories

000
001
010
011
100
101
110
111

We could do 8 experiments, presenting each one several times (for sim)

Or, 000101110001 0111 ...
preset.

Note that past histories contain each of the 3-bit inputs; but we gain 3× in efficiency.

Basic idea is to choose a sequence of 0's & 1's for which the φ's are nearly orthogonal.

We will construct a sequence of length $2^n - 1$ which

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

(b) is shift-orthogonal, i.e., $\langle \sigma(k)\, \sigma(k-n) \rangle = \pm \frac{1}{2^n - 1}$

(a) will allow for measurement of high-order kernels at short lags
(b) " " " " " first-order kernels at all lags,
which often is enough to characterize the system.

σ: $+1$ or -1, but helpful to think of it sometimes as 0's & 1's

Recipe: We will create a finite field of size 2^n, & write a table of logarithms in it.

$n = 4$.

Consider the finite field $\mathbb{Z}_2 = \{0, 1\}$, and let's choose a polynomial of degree n that does not factor in $\mathbb{Z}_2$ (other conditions too).

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

$x^0 =$	1
$x^1 =$	x
$x^2 =$	x^2
$x^3 =$	x^3
$x^4 =$	$x + 1$
$x^5 =$	$x^2 + x$
$x^6 =$	$x^3 + x^2$
$x^7 =$	$x^3 + x + 1$
$x^8 =$	$x^2 + 1$
$x^9 =$	$x^3 + x$
$x^{10} =$	$x^2 + x + 1$
$x^{11} =$	$x^3 + x^2 + x$
$x^{12} =$	$x^3 + x^2 + x + 1$
$x^{13} =$	$x^3 + x^2 + 1$
$x^{14} =$	$x^3 + 1$
$x^{15} =$	1

E.g., $x^8 = x^7 \cdot x = (x^3 + x + 1) \cdot x$
$= x^4 + x^2 + x$
$= (x + 1) + x^2 + x$
$= x^2 + 1$

Then any column is the m-sequence.

$\boxed{371}$

A recurrence rule holds within a column:

$p(x) = x^n + q(x)$

$q(x) = \sum_{r=0}^{n-1} b_r x^r$

$x^k \cdot x^n = x^k \cdot q(x)$

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

Here, $b_0 = 1, b_1 = 1, b_2 = b_3 = 0$

So, for example $k = 5$

$b=0$	$x^5 =$		x^2	$+x$		
$b=1$	$x^6 =$	x^3	$+x^2$			
$b=2$	$x^7 =$	x^3		$+x$	$+1$	
$b=3$	$x^8 =$		x^2		$+1$	
$b=4$	$x^9 =$	x^3		$+x$		

$x^9 = x^6 + x^5$

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

If the sequence runs 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-orthogonality?

$a \neq 0$: $x^k + x^{k+a} = \begin{cases} 0 \text{ if match} \\ 1 \text{ if mismatch} \end{cases}$

const term is σ_k — const term is σ_{k+a}

so $x^k + x^{k+a}$ is cross correlation of σ_k & σ_{k+a}.

$x^k + x^{k+a} = x^k(1+x^a) = x^k \cdot x^{\lambda(a)}$

because $1+x^a$ must be some $\lambda(a)$, so $x^k x^{\lambda(a)} = x^{k+\lambda(a)}$.

The above shows that an m-sequence XOR'd a shift of itself is just another shift of the same m-sequence.

So $X^{k+\lambda(a)}$ has ~ equal # of 1's & 0's, so X^k & X^{k+a} are independent

The drawback is that high-order regressors (polynomials) overlap with low-order ones:

$$\langle r(t)\,\sigma(t-\tau_1)\,\sigma(t-\tau_2)\rangle$$

must $= \langle r(t)\,\sigma(t-\lambda(\tau_1,\tau_2))\rangle$

since $\sigma(t-\tau_1)\sigma(t-\tau_2) = \sigma(t-\lambda(\tau_1,\tau_2))$ by above argument.

What can you do? Choose $p(x)$ so $\lambda(\tau_1,\tau_2)$ is large.

Or, let R contain $\{\sigma_k\}$ and $\{-\sigma_k\}$

so that now, $\sigma(t-\tau)$ and $\sigma(t-\tau_1)\sigma(t-\tau_2)$

are orthogonal over R.

("Inverted repeat" method)