

BME I5100: Biomedical Signal Processing

Stochastic Processes



Lucas C. Parra Biomedical Engineering Department City College of New York



Lucas Parra, CCNY



Schedule

Week 1: Introduction Linear, stationary, normal - the stuff biology is **not** made of.

Week 1-5: Linear systems Impulse response Moving Average and Auto Regressive filters Convolution Discrete Fourier transform and z-transform Sampling

Week 6-7: Analog signal processing Operational amplifier Analog filtering

Week 8-11: Random variables and stochastic processes Random variables Moments and Cumulants Multivariate distributions, Principal Components Stochastic processes, linear prediction, AR modeling

Week 12-14: Examples of biomedical signal processingHarmonic analysis- estimation circadian rhythm and speechLinear discrimination- detection of evoked responses in EEG/MEGHidden Markov Models and Kalman Filter- identification and filtering



Stochastic Processes

A stochastic process is a random quantity that changes over time. More specifically it is a **set of random variables sorted in time** X[t]. Sampling a random process repeatedly gives and ensemble:



Examples: Spike train, position of a particle moving randomly, number of bacteria in a culture, thermal noise in an amplifier, etc.



Stochastic Process - Poisson Process

Poisson process arises when we count the number of occurrences N(t) of **independent** random events where the likelihood of occurrence of an event is **proportional to the time** interval Δt .

$$Pr(N(t + \Delta t) - N(t) = 1) = \nu \Delta t$$

One can proof by induction that the count follows a Poisson distribution with $\lambda = vt$

$$Pr(N(t)=k) = \frac{(\nu t)^k e^{-\nu t}}{k'}$$

Interesting is the first induction step. For k=0 using definition and limes $\Delta t \rightarrow 0$ it is easy to see that

$$Pr(N(t)=0)=e^{-\nu t}$$

Which means that the ISI is exponentially distributed.



Stochastic Process - Shot Noise

Shot noise we call Poisson process with events at times t_i convolved with a IR.

$$X(t) = \sum_{i=1}^{N} h(t) \delta(t - t_i) = \sum_{i=1}^{N} h(t - t_i)$$

The expected value of *X*(*t*) is

$$E[X(t)] = v(t) \int_{-\infty}^{\infty} h(u) du$$

Event or spike rate v can be estimated by convolving a spike train with a sufficiently long IR. This in effect is what neurons do, where the IR is given by the post-synaptic potential (PSP).

Sometimes $(t_{i}t_{i})^{1}$ is considered "instantaneous" spike rate v(t).



Stochastic Process - Gaussian Process

A stochastic process x(t) is said to be a **Gaussian process** if $x = [x(t_1), x(t_2), ..., x(t_n)]^T$ are jointly Gaussian distributed

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]$$

In general the covariance matrix Σ and mean μ depend on the time instances t_1, t_2, \dots, t_n .

Classic example is Brownian motion with $r_{ij} = Bmin(t_i, t_j)$



Stochastic Process - Wide Sense Stationary

If $p(x(t_1), x(t_2), ..., x(t_n))$ does not depend on times $t_1, t_2, ..., t_n$ the process is called **stationary**.

Any stochastic process for which the *mean and covariance* are *independent of time* is called **Wide Sense Stationary (WSS)**.

A WSS process is called **ergodic** if the sample average converges to the ensemble average. With samples $x[k] = x(t_k)$ this reads:

$$E[f(x(t))] = \lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} f(x[k])$$



Stochastic Process -Auto-correlation

If we have equal distant samples, x[k] = x(kT), the second moments correspond to the **auto-correlation**

$$r_x[k,l] = E[x[k]x^*[l]]$$

For WSS stochastic processes:

The mean is independent of time $m_x = E[x[k]]$ The Auto-correlation

$$r_{x}[k] = E[x[n+k]x^{*}[n]]$$
depends only on time differences
$$r_{x}[k,l] = r_{x}[k-l]$$
it is symmetric,
$$r_{x}[k] = r_{x}^{*}[-k]$$
the zero lag is the power
$$r_{x}[0] = E[|x[k]|^{2}]$$
upper bounded by the power
$$r_{x}[0] \ge |r_{x}[k]| = 8$$



Stochastic Process - Auto-correlation

Assuming ergodic WSS a direct estimate of the autocorrelation is the (unbiased) **sample auto-correlation**



Note that correlation is convolution with opposite sign. It can be computed fast with the FFT. Use xcorr()



Stochastic Process - Auto-correlation

Examples



10



Stochastic Process -Auto-correlation

For WSS the **auto-correlation matrix** becomes Hermitian Toeplitz

$$\boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}} = \begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2) & \cdots & r_x^*(p) \\ r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_x(p) & r_x(p+1) & r_x(p+2) & \cdots & r_x(0) \end{bmatrix}$$

Important WSS process is (Gaussian) uncorrelated noise

$$R_{xx} = \sigma^2 I \qquad m_x = 0$$

>> randn()



Stochastic Process - Power spectrum

Oscillations of the correlation are best analyzed in the frequency domain, which leads to the **Power Spectrum**

$$P_{x}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_{x}[k]e^{-jk\omega}$$

One can show that $P_r(e^{j\omega})$ is *real*, *even* and *positive*.

The auto-correlation can be recovered with the inverse Fourier transform

$$r_x[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega P_x(e^{j\omega}) e^{jk\omega}$$



Stochastic Process - Power spectrum

In particular, the total power is given by

$$r_{x}[0] = E[\|x[k]\|^{2}] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega P_{x}(e^{j\omega})$$

the power spectrum is sometimes called **spectral density** because it is *positive* and the signal power can always be *normalized* to $r(0) = (2\pi)^{-1}$.

Example: Uncorrelated noise has a constant power spectrum

$$r[k] = E[x[n+k]x^*[n]] = \sigma^2 \delta(k)$$
$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \sigma^2 \delta(k) e^{-jk\omega} = \sigma^2$$

Hence it is also called white noise.



14

Stochastic Process - Power spectrum

The power spectrum captures the spectral content of the random sequence. It can be estimate directly from the Fourier transform of the data:

$$\hat{P}_{x}(e^{j\omega}) = \frac{1}{N} |X(e^{j\omega})|^{2}$$
$$X(e^{j\omega}) = \sum_{k=0}^{N-1} x[k] e^{-j\omega k}$$

This estimate improves in the mean with increasing N.

$$P_x(e^{j\omega}) = \lim_{N \to \infty} E[\hat{P}_x(e^{j\omega})]$$

Remember that when estimating auto-correlation and power spectrum we implicitly assume WSS processes!



Stochastic Process - Power spectrum

Unfortunately, the direct estimate is inconsistent, i.e. its variance does not converge to 0 for increasing N.



A classic heuristic, called the **periodogram**, is to smooth neighboring frequencies: Compute Fourier transform for window of size N/K and average over K windows. Additionally do some windowing: pwelch()15

Lucas Parra, CCNY



Stochastic Process - Filtering

The effect of filtering on the statistics of a stochastic process:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

$$x[n] \qquad h[k] \qquad y[n]$$

Filtering corresponds to convolution of the autocorrelation

$$r_{y}[k] = \sum_{l=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h[l]r_{x}[m-l+k]h^{*}(m)$$

= $r_{x}[k] * h[k] * h^{*}[-k]$
$$r_{x}[n] \qquad h[k] \qquad r_{yx}[k] \qquad h^{*}[-k] \qquad r_{y}[n]$$

Lucas Parra, CCNY



Stochastic Process - Filtering

With the convolution theorem of the Fourier transform we find for the power spectrum

$$P_{y}(e^{j\omega}) = P_{x}(e^{j\omega})H(e^{j\omega})H^{*}(e^{-j\omega}) = P_{x}(e^{j\omega})|H(e^{j\omega})|^{2}$$



Or in the z-domain

$$P_{y}(z) = P_{x}(z)H(z)H^{*}(1/z^{*})$$



Stochastic Process - Filtering



pwelch(x*sqrt(pi),128)

Lucas Parra, CCNY



Stochastic Process - Spectral Content

"The power spectrum gives the spectral content of the data." To see that consider the power of a signal after filtering with a narrow bandpass filter around ω_0 .

$$E[|y[n]|^{2}] = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\omega |H(e^{j\omega})|^{2} P_{x}(e^{j\omega})$$

$$= \frac{1}{2\pi} \int_{\omega_{0}-\Delta\omega/2}^{\omega_{0}+\Delta\omega/2} d\omega P_{x}(e^{j\omega})$$

$$\approx \frac{\Delta\omega}{2\pi} P_{x}(e^{j\omega_{0}})$$



Stochastic Process - Regular process

Colored noise process *x*[*n*] is filtered white noise *v*[*n*]



If h[n] is *causal* and *invertible*, x[n] is called a **regular process** and v[n] the corresponding **innovation process** with power σ_v^2 . Its power spectrum is continuous and given by

$$P_x(e^{j\omega}) = \sigma_v^2 |H(e^{j\omega})|^2$$

The inverse filter 1/H(z) is called the whitening filter of x[n].



Stochastic Process - Predictive process

A process x[n] is called **predictive** or **singular** if it can be predicted exactly from a linear combination of its past:

$$x[n] = \sum_{k=1}^{\infty} a[k]x[n-k]$$

Examples are sinusoids or sums thereof:

$$x[n] = \sum_{k=1}^{\infty} A_k e^{in\omega_k + \phi_k}$$

The spectrum of a predictive process has only discrete lines:

$$P_{x}(e^{j\omega}) = \sum_{k=1}^{\infty} \alpha_{k} \delta(\omega - \omega_{k})$$



Stochastic Process - Wold decomposition

The **Wold decomposition theorem** tells us that *any WSS process* x[n] can be decomposed into a sum of a regular process $x_r[n]$ and predictive process $x_p[n]$:

$$x[n] = x_p[n] + x_r[n]$$

where the processes are **orthogonal**:

$$E[x_p[n]x_r^*[m]]=0$$

Hence the spectrum of any WSS is composed of a continuous component with invertible H(z) plus discrete lines:

$$P_{x}(e^{j\omega}) = |H(e^{j\omega})|^{2} + \sum_{k=1}^{\infty} \alpha_{k} \delta(\omega - \omega_{k})$$
²²



Stochastic Process - Wold decomposition

Note that for orthogonal processes the power spectrum is additive.

If
$$z[n]=x[n]+y[n]$$
 with

$$E[x[n]y^*[m]]=0$$

Then

If

$$P_{z}(e^{j\omega}) = P_{x}(e^{j\omega}) + P_{y}(e^{j\omega})$$



Stochastic Process -Spectrogram

Spectrogram: Power spectrum with strong spectral component can be estimate on short sequences, and hence, followed as it develops over time.



Assumption: WSS within each small window



Frequency (units of π)

Stochastic Process - ARMA process

An important regular process is the **ARMA process**: white noise v[n] filtered with a rational transfer function





Stochastic Process - ARMA modeling

ARMA modeling means to find a[k], and b[k] that approximate an observed $P_x(e^{j\omega})$ for a given x[n]



Note that now the input is not given. Instead only statistical assumptions on the input are know (white WSS process).



Stochastic Process - ARMA modeling

We derive now statistical conditions for a[k] and b[k]. Multiply the difference equation by $x^*[n - k]$ and take the expectation:

$$\sum_{l=0}^{P} a[l]r_{x}[k-l] = \sum_{l=0}^{Q} b[l]r_{vx}[k-l]$$

with cross-correlation: $r_{vx}[k-l] = E[v[n-l]x^*[n-k]]$ Using x[n] = h[n] * v[n]

$$E[v[n]v^*[m]] = \delta(n-m)$$

we obtain (non-linear) conditions for *a*[*k*] and *b*[*k*]

$$\sum_{l=0}^{P} a[l]r_{x}[k-l] = \sum_{l=0}^{Q} b[l]h^{*}[l-k]$$



Stochastic Process - ARMA modeling

These conditions are called the Yule-Walker equations

$$\sum_{l=0}^{P} a[l]r_{x}[k-l] = \sum_{l=0}^{Q-k} b[l+k]h^{*}[l] \equiv c[k]$$

In matrix form and using Hermitian symmetry of the autocorrelation we can write for lags 0...L:

$$\begin{bmatrix} r_x[0] & r_x^*[1] & r_x^*[2] & \cdots & r_x^*[P] \\ r_x[1] & r_x[0] & r_x^*[1] & \cdots & r_x^*[P-1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_x[Q] & r_x[Q-1] & r_x[Q-2] & \cdots & r_x[Q-P] \\ r_x[Q+1] & r_x[Q] & r_x[Q-1] & \cdots & r_x[Q-P+1] \\ \vdots & \vdots & \vdots & \vdots \\ r_x[L] & r_x[L-1] & r_x[L-2] & \cdots & r_x[L-P] \end{bmatrix} \begin{vmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[P] \end{vmatrix} = \begin{bmatrix} c[0] \\ c[1] \\ \vdots \\ c[Q] \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Since $c[k]$ depend on $h[k]$ these equations are non-linear in $a[k]$

and b[k]. They simplify for Q=0, i.e. AR model only.



Stochastic Process - AR modeling

For *Q*=0 the Yule-Walker equations simplify to

$$\sum_{l=0}^{P} a[l]r_{x}[k-l] = |b[0]|^{2}\delta(k)$$

In matrix form and using Hermitian symmetry of the autocorrelation we obtain the AR **normal equations**:

$$\begin{bmatrix} r_x[0] & r_x^*[1] & r_x^*[2] & \cdots & r_x^*[P] \\ r_x[1] & r_x[0] & r_x^*[1] & \cdots & r_x^*[P-1] \\ r_x[2] & r_x[1] & r_x[0] & \cdots & r_x^*[P-2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_x[P] & r_x[P-1] & r_x[P-2] & \cdots & r_x[0] \end{bmatrix} \begin{vmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[P] \end{vmatrix} = \begin{vmatrix} b[0] \\ 2 \\ 0 \\ \vdots \\ a[P] \end{vmatrix}$$

Which are a set of linear equations for the unknown a[k].



Stochastic Process - AR modeling

The AR normal equations can be rewritten as:

$$\begin{bmatrix} r_{x}[0] & r_{x}^{*}[1] & \cdots & r_{x}^{*}[P-1] \\ r_{x}[1] & r_{x}[0] & \cdots & r_{x}^{*}[P-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{x}[P-1] & r_{x}[P-2] & \cdots & r_{x}[0] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[P] \end{bmatrix} = \begin{bmatrix} r_{x}[1] \\ r_{x}[2] \\ \vdots \\ r_{x}[P] \end{bmatrix}$$
$$R_{xx} a = -r_{x}$$

Linear equations with Toeplitz Hermitian form can be solved efficiently with the Levinson-Durbin recursion.

Computes the auto-correlations and solves this linear equation:



Stochastic Process - Linear prediction

The goal of linear prediction is to predict sample x[n] from its past *P* samples:

$$\hat{x}[n] = -\sum_{k=1}^{P} a[k]x[n-k]$$

And a[k] are called the **linear prediction coefficients** (LPC). The prediction error is given by

$$e[n] = x[n] - \hat{x}[n]$$

The LPC that minimize the expected error

$$a = argmin E[|e[n]|^2]$$

satisfy the same linear equations as the AR model parameters:

$$\sum_{l=0}^{P} a[l]r_{x}[k-l] = \epsilon \delta(k) \qquad \epsilon = E[|e[n]|^{2}]$$

which we obtain by taking derivatives with respect to $a^*[k]$. 31



Stochastic Process - LPC and Wold Decomp.

In matlab all this is done with:

```
>> [a,e] = lpc(x,P);
>> b = sqrt(e);
```

The error process e[n] represents the "new" or "innovative" part over the linear prediction, hence the name innovation process.

One can show that the innovation in linear prediction is white and uncorrelated to the prediction,

$$E[e[n]e^*[m]] = \delta(n-m)$$
$$E[e[n]\hat{x}^*[m]] = 0$$



Stochastic Process - AR and LPC

Assignment 9:

- Pick either speech or heart rate >> load hrs.mat; x=hrs;

 - >> [x,fs]=audioread('speech.wav');x=x(5501:6012);
- Generate an AR model by solving normal equations.
- Plot an estimated power spectrum together with the AR model spectrum.
- Show the innovation process and its power spectrum.
- Compute the LPC coefficient using lpc(x, P)
- Compare your AR results with LPC.
- Show the linear prediction estimation error.
- Compute the signal to noise ratio (SNR) of your LPC model:

$$SNR = \frac{\sum_{n} |\hat{x}[n]|^2}{\sum_{n} |e[n]|^2}$$

• Show SNR in dB as a function of model order *P*.