Joint Signal Analysis Overview

- Cross-correlation
- Cross Power Spectrum
- Examples

Introduction

- Mostly we have focused on estimating statistical properties of a single univariate signal
  - Autocorrelation function (ACF)
  - Partial autocorrelation function (PACF)
  - Power spectral density
- In many applications we have two or more signals, \( x(n) \) and \( y(n) \)
- Would like to say something about how they are related

System Identification

- Joint signal analysis is related to system identification
- The goal of system identification is to build a model
- That is, estimate \( H(z) \) and \( G(z) \), given \( x(n) \) and \( y(n) \)
  - Parametric, though order may be estimated
  - Mostly LTI systems
  - Some methods for MIMO systems

Joint Signal Analysis

- Joint signal analysis characterizes the relationship between a pair of signals
  - We will focus on nonparametric methods
  - LTI systems
  - Only 2 signals
- We have already discussed many of the possible properties
  - Normalized cross-correlation, aka cross-correlation function (CCF))
  - Cross-power spectral density (CPSD)
  - Coherence
Normalized Cross-Correlation Defined

The Normalized Cross-Correlation, also known as the Cross-correlation Function (CCF), is defined for a WSS signal as

$$\rho_{yx}(\ell) = \frac{\gamma_{yx}(\ell)}{\sqrt{\gamma_y(0)\gamma_x(0)}}$$

where $$\gamma_{yx}(\ell)$$ is the cross-covariance of $$y(n)$$ and $$x(n)$$,

$$\gamma_{yx}(\ell) = \mathbb{E}[(y(n+\ell) - \mu_y)(x(n) - \mu_x)]$$

If we assume that $$x(n)$$ and $$y(n)$$ are related by an LTI system $$H(z)$$, then we can estimate the properties of $$H(z)$$ given realizations of $$x(n)$$ and $$y(n)$$

- Magnitude
- Phase
- Group delay
- Impulse response

Causality

- Strictly speaking, given only realizations of $$x(n)$$ and $$y(n)$$ we cannot determine whether one signal caused the other or not
- This is a fundamental idea rooted in statistics
- However, under certain assumptions causality can be determined
- For example, if we assume that any system that relates $$x(n)$$ to $$y(n)$$ or vice versa is causal, then we may be able to get a sense of which caused which from the estimated cross-correlation

Estimated Cross-Covariance

$$\hat{\gamma}_u(\ell) \triangleq \frac{1}{N-|\ell|} \sum_{n=0}^{N-1-|\ell|} \left[ y(n+|\ell|) - \hat{\mu}_y \right] \left[ x(n) - \hat{\mu}_x \right]$$

$$\hat{\gamma}_b(\ell) \triangleq \frac{N - |\ell|}{N} \hat{\gamma}_u(\ell)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1-|\ell|} \left[ y(n+|\ell|) - \hat{\mu}_y \right] \left[ x(n) - \hat{\mu}_x \right]$$

for $$|\ell| < N$$. $$\hat{\gamma}_u(\ell) = \hat{\gamma}_b(\ell) = 0$$ for $$|\ell| \geq N$$.

- There are similar trade-offs between biased and unbiased estimates of the CCF as there were for the ACF
- As with ACF, we generally assume the means are zero to simplify calculations and eliminate DC artifacts
- The bias caused by estimating the means is usually negligible
### Estimated Cross-Covariance: Biased versus Unbiased

If $\mu_y = \mu_x = 0$, 

$$
\hat{\gamma}_u(\ell) = \frac{1}{N - |\ell|} \sum_{n=0}^{N-1-|\ell|} y(n + |\ell|) x(n)
$$

$$
\hat{\gamma}_b(\ell) = \frac{1}{N} \sum_{n=0}^{N-1-|\ell|} y(n + |\ell|) x(n)
$$

- The true CCF is not positive definite and the cross-spectral density is not non-negative, in general, so this is no longer a concern
- Nonetheless, the unbiased estimator contains excessive variance at large lags and thus the biased estimate is generally preferred

### Estimated Cross-Correlation Functions

The natural estimators of the CCF are

$$
\hat{\rho}_b(\ell) \triangleq \frac{\hat{\gamma}_b(\ell)}{\hat{\sigma}_x \hat{\sigma}_y}
$$

$$
\hat{\rho}_u(\ell) \triangleq \frac{\hat{\gamma}_u(\ell)}{\hat{\sigma}_x \hat{\sigma}_y}
$$

- In general, it is not possible to obtain confidence intervals for the estimated CCF because the variance of the estimator depends on the true CCF and signal ACF’s
- Instead, it is common practice to plot the confidence intervals of a purely random process
- If $N$ is large enough, the central limit theorem applies and $\text{var}\{\hat{\rho}_b(\ell)\}$ is approximately normal for all $\ell$
- In this case, we can use the Normal cdf to plot confidence intervals of an IID sequence
- These are proportional to $\pm \sqrt{\text{var}\{\hat{\rho}_{yx}(\ell)\}}$
- This is the same as it was for the ACF

### Example 1: CCF Estimation

Estimate the CCF of the ARMA process investigated earlier for PSD estimation.

### Example 2: MATLAB Code

```matlab
L = 200; % Length of autocorrelation calculated
N = 250;
c1 = 99; % Confidence level
NP = norminv((1-c1/100)/2); % Find corresponding lower percentile of Normal
b = poly([-0.8,0.97*exp(j*pi/4),0.97*exp(-j*pi/4),0.97*exp(j*pi/6),0.97*exp(-j*pi/6)]); % Numerator
a = poly([0.8,0.95*exp(j*3*pi/4),0.95*exp(-j*3*pi/4),0.95*exp(j*2.5*pi/4),0.95*exp(-j*2.5*pi/4)]); % Denominator
b = b*sum(a)/sum(b); % Set DC gain to 1
x = randn(N,1);
y = filter(1,a,x);```

---
Example 2: Signals

N=250

Example 2: MATLAB Code

```matlab
figure
n = 0:min(200,N);
subplot(2,1,1);
h = stem(n,x(n+1));set(h(1),'MarkerFaceColor','b');
set(h(1),'MarkerSize',2);
xlim([0 max(n)]);ylim([min(x) max(x)]);
ylabel('x(n)');title(sprintf('N=%d',N));box off;
subplot(2,1,2);
h = stem(n,y(n+1),'g');set(h(1),'MarkerFaceColor','g');
set(h(1),'MarkerSize',2);
xlim([0 max(n)]);ylim([min(y) max(y)]);
xlabel('Sample (n)');ylabel('y(n)');box off;
```

Example 2: Pole Zero Map of ARMA Process

Example 2: MATLAB Code

```matlab
figure
h = circle;
z = roots(b);
p = roots(a);
hold on;
h2 = plot(real(z),imag(z),'bo',real(p),imag(p),'rx');
hold off;
axis square;
xlim([-1.1 1.1]);
ylim([-1.1 1.1]);
box off;
```
Example 2: Biased CCF

\[ \rho(l) = \frac{cc}{N \cdot \sqrt{|l|}} \]

Example 2: MATLAB Code

```matlab
np = 2^nextpow2(2*N-1); % Figure out how much to pad the signal
Y = fft(y,np);
X = fft(x,np);
cp = ifft(Y.*conj(X));
cc = real([cp(np-(L-1:-1:0));cp(1:L+1)]);
cc = cc/(std(y)*std(x));
l = (-L:L).

figure
b = stem(l,cc);
set(b,'Marker','None');xlim([-L L]);
ylim([-1 1]);
hold on;
plot(xlim,NP*[1 1]/sqrt(N),'k:',xlim,-NP*[1 1]/sqrt(N),'k:');
plot([0 L],[1 (N-L)/N],'g',[0 L],[-1 (N-L)/N],'g');
plot([-L 0],[0 L]/N,1,'g',[-L 0],[-(N-L)/N 1],'g');
hold off;
ylabel('\rho(l)');xlabel('Lag (l)');title(sprintf('N=%d',N));
box off;
```

Example 2: Unbiased CCF

\[ \rho(l) = \frac{cc}{N \cdot \sqrt{|l|}} \]

Example 2: MATLAB Code

```matlab
np = 2^nextpow2(2*N-1); % Figure out how much to pad the signal
Y = fft(y,np);
X = fft(x,np);
cp = ifft(Y.*conj(X));
cc = real([cp(np-(L-1:-1:0));cp(1:L+1)]);
cc = cc/(std(y)*std(x));cc = cc/N;
l = (-L:L).

figure
b = stem(l,cc);
set(b,'Marker','None');xlim([-L L]);
ylim([-1 1]);
hold on;
plot(xlim,NP*[1 1]/sqrt(N),'k:',xlim,-NP*[1 1]/sqrt(N),'k:');
plot([-L 0],[0 L]/N,1,'g',[-L 0],[-1 (N-L)/N 1],'g');
hold off;
ylabel('\rho(l)');xlabel('Lag (l)');title(sprintf('N=%d',N));
box off;
```
Cross-Power Spectral Density

Recall that the cross-power spectral density (CPSD) is defined as

\[ R_{yx}(e^{j\omega}) \triangleq \sum_{\ell=-\infty}^{\infty} r_{yx}(\ell)e^{-j\omega \ell} \]

- Unlike the PSD, the CPSD is a complex-valued function
- Can work with CPSD in either rectangular or polar form
- Rectangular
  - Cospectrum: \( C_{yx}(\omega) \triangleq \text{Re}\{R_{yx}(e^{j\omega})\} \)
  - Quadrature spectrum: \( Q_{yx}(\omega) \triangleq \text{Im}\{R_{yx}(e^{j\omega})\} \)
- More commonly, it is expressed in terms of magnitude and phase
  - Cross-Amplitude Spectrum: \( A_{yx}(\omega) \triangleq |R_{yx}(e^{j\omega})| \)
  - Phase Spectrum: \( \Phi_{yx}(\omega) \triangleq \angle R_{yx}(e^{j\omega}) \)

Estimating Cross-Power Spectral Density

The Cross-periodogram is defined as

\[ \hat{R}_{yx}(e^{j\omega}) \triangleq \sum_{\ell=-\frac{N-1}{2}}^{\frac{N-1}{2}} \hat{r}_{yx}(\ell)e^{-j\omega \ell} \]

\[ = \frac{1}{N} \left[ \sum_{n=0}^{N-1} x(n)e^{-j\omega n} \right] \left[ \sum_{n=0}^{N-1} y(n)e^{-j\omega n} \right]^* \]

- We may use the same nonparametric techniques to estimate the CPSD that we used to estimate the PSD
- The same tradeoffs exist
  - Main lobe width versus side lobe height
  - Bias versus variance
- Can reduce variance at the expense of bias using either smoothing (Blackman-Tukey) or averaging (Welch-Bartlett)

Estimating Frequency Response

- The book argues that Welch’s approach to modified periodogram averaging is preferred
- Not clear to me why they prefer this to Blackman-Tukey, which I think is better
Use the Blackman-Tukey estimate to estimate the PSDs of $x(n)$ and $y(n)$ for various signal to noise ratios (SNRs). Also estimate the transfer function magnitude and phase for the PZ system described earlier.

- It is instructive to consider the case of additive noise
- Here we assume that $w(n)$ is statistically independent of $x(n)$ and all signals are stationary with zero mean
- Note: different notation than text
Example 3: Blackman-Tukey Estimate $R_y(e^{j\omega})$

Example 3: Blackman-Tukey Estimate $|H(e^{j\omega})|$

Example 3: Blackman-Tukey Estimate $\angle H(e^{j\omega})$

Example 3: Blackman-Tukey Estimate $R_x(e^{j\omega})$
Example 3: Blackman-Tukey Estimate $R_{xy}(ej\omega)$

- Real CPSD
- Imaginary CPSD

Example 3: Blackman-Tukey Estimate $|H(e^{j\omega})|$

- Estimated $R_y(e^{j\omega})$

Example 3: Blackman-Tukey Estimate $\angle H(e^{j\omega})$

- True PSD
- Estimated PSD
Example 3: Blackman-Tukey Estimate $R_y(e^{j\omega})$

![Graph of $R_y(e^{j\omega})$ with legends indicating True PSD, Confidence Bands, Average Estimate, and Single Estimate.]

Example 3: Blackman-Tukey Estimate $R_{yx}(e^{j\omega})$

![Graph of $R_{yx}(e^{j\omega})$ with legends indicating True Cross-PSD, Confidence Bands, Average Estimate, and Single Estimate.]

Example 3: Blackman-Tukey Estimate $R_x(e^{j\omega})$

![Graph of $R_x(e^{j\omega})$ with legends indicating True PSD, Confidence Bands, Average Estimate, and Single Estimate.]

Example 3: Blackman-Tukey Estimate $|H(e^{j\omega})|$

![Graph of $|H(e^{j\omega})|$ with legends indicating True $H$, Confidence Bands, Average Estimate, and Single Estimate.]

Frequency (cycles/sample)
Example 3: MATLAB Code Continued

```matlab
for c1 = 1:length(SNR),
    nx = P+N;
    Rhx = zeros(NA,NZ/2+1); Rhy = zeros(NA,NZ/2+1);
    Rhyx = zeros(NA,NZ/2+1);
    Hhm = zeros(NA,NZ/2+1); Hha = zeros(NA,NZ/2+1);
    kh = 1:NZ/2+1;
    fh = (kh-1)/NZ;
    no = (L+1)/2; % Number of one-sided points to included in estimate
    wn = blackman(no*2-1); % Create window
    wn = wn/max(wn); % Make maximum = 1
    v = 2*N/(sum(wn.^2));
    for c2 = 1:NA,
        x = randn(P+N,1); % System with known PSD
        y = filter(b,a,x); % System with known PSD
        npw = var(y)/SNR(c1); % Noise power
        y = y(nx-N+1:nx); % Eliminate start-up transient (make stationary)
        x = x(nx-N+1:nx); % Eliminate first set of samples

        xf = fft(x,np); % Calculate FFT of x
        rx = ifft(xf.*conj(xf)); % Calculate biased autocorrelation
        rx = real(rx(1:nx))/N; % Eliminate superfluous zeros
        rx = [rx(no:-1:2);rx(1:no)]; % Make two-sided, using symmetry
        rhx = abs(fft(rx.*wn,NZ)); % Window autocorrelation

        yf = fft(y,np);
        ry = ifft(yf.*conj(yf)); % Calculate biased autocorrelation
        ry = real(ry(1:nx))/N; % Eliminate superfluous zeros
        ry = [ry(no:-1:2);ry(1:no)]; % Make two-sided, using symmetry
        rhy = abs(fft(ry.*wn,NZ)); % Window autocorrelation

        ryx = ifft(yf.*conj(xf)); % Calculate biased cross-correlation
        ryx = real(ryx(1:nx))/N; % Eliminate superfluous zeros
        ryx = [ryx(no:-1:2);ryx(1:no)]; % Make two-sided, using symmetry
        rhyx = exp(j*(0:NZ-1).*(2*pi/NZ)*(no-1)).*fft(ryx.*wn,NZ);

        Rhx(c2,:) = rhx(kh).';
        Rhy(c2,:) = rhy(kh).';
        Rhyx(c2,:) = rhyx(kh).';
        Hhm(c2,:) = abs(Rhyx(c2,:)./Rhx(c2,:));
        Hha(c2,:) = rem(angle(Rhyx(c2,:))*180/pi,180);
    end;
end;
```
Example 3: Observations

- Phase can be estimated accurately at frequencies where SNR is high

Example 3: MATLAB Code Continued

```matlab
for c2=1:2,
    switch c2,
    case 1,
        R = Rx;
        Rha = mean(Rhx); % Average
        Rhu = prctile(Rhx,100-(100-cb)/2); % Upper confidence band
        Rhl = prctile(Rhx, (100-cb)/2); % Lower confidence band
        Rh1 = Rhx(1,:);
        fn = sprintf('BTRx%03d',SNR(c1)*10);
        yl = 'Estimated R_x(e^{j\omega})';
    case 2,
        R = H2 + npw;
        Rha = mean(Rhy); % Average
        Rhu = prctile(Rhy,100-(100-cb)/2); % Upper confidence band
        Rhl = prctile(Rhy, (100-cb)/2); % Lower confidence band
        Rh1 = Rhy(1,:);
        fn = sprintf('BTRy%03d',SNR(c1)*10);
        yl = 'Estimated R_y(e^{j\omega})';
    end;
end;
```

Example 3: MATLAB Code Continued

```matlab
figure;
subplot(2,1,1);
    h = plot(f,R,'r',fh,Rhl,'b',fh,Rhu,'b',fh,Rha,'k',fh,Rh1,'g');
    set(h(1) ,'LineWidth',1.3);
    set(h(2:3),'LineWidth',0.5);
    set(h(4) , 'LineWidth',0.5);
    set(h(5) , 'LineWidth',0.5);
    ylabel(yl);
    title(sprintf('N:%d L:%d NA:%d SNR:%3.1f Confidence Bands:%d%%',N,L,NA,SNR(c1),cb));
    xlim([0 0.5]);
    ylim([0 max(R)*2.]);
    legend(h([1 3 4 5]),'True PSD','Confidence Bands','Average Estimate','Single Est
subplot(2,1,2);
    h = semilogy(f,R,'r',fh,Rhl,'b',fh,Rhu,'b',fh,Rha,'k',fh,Rh1,'g');
    set(h(1) ,'LineWidth',1.3);
    set(h(2:3),'LineWidth',0.5);
    set(h(4) , 'LineWidth',0.5);
    set(h(5) , 'LineWidth',0.5);
    ylabel(yl);
    xlabel('Frequency (cycles/sample)');
    xlim([0 0.5]);
    ylim([0.2*min(R) max(R)*5]);
end;
```