Contents

1 FIGURE 1 – BACCALA ET AL. (2016) DTF: UNIFIED ASYMPTOTIC THEORY

DESCRIPTION:

Routine **figure1_example1_pdc_dtf_ns50.m** publish

Linear bivariate VAR(2) model

LA Baccala, DY Takahashi, K Sameshima (2016) Directed Transfer Function: Unified Asymptotic Theory and Some of its Implications. *IEEE Transactions on Biomedical Engineering* **PP**.

```
http://dx.doi.org/10.1109/TBME.2016.2550199
```

Example 1: Simple example VAR(2) without feedback

```
                    x1==>x2
```

Contents

- Generating data set for analysis
- Equation
- DTF estimation
- PDC estimation
- $|PDC(\lambda)|^2$ Matrix Layout Plotting
- Result from the original article, Baccala et al (2016)
- Some remarks:

Generating data set for analysis

```
clear; clc; format compact
flgPlotStyle = 'Print'; % or 'Screen'
flgRandomize = 0; % Generate the specific data set used in Fig. 1.
ns = 50;          % number of sample points
nDiscard = 2000;  % number of points discarded at beginning of simulation
p = 2;            % model order

if (exist('figure1_example1_pdc_dtf_ns50.mat') == 2) & is_octave & ~
   flgRandomize,
   load figure1_example1_pdc_dtf_ns50
else
   [u] = fbaccala2016_example1( ns, nDiscard, flgRandomize );
   if ~is_octave & ~flgRandomize,
      save figure1_example1_pdc_dtf_ns50 u
```

```
   end;
end;

chLabels = [];    % Using default labeling schema for channel identification
```

```
        ==========================================================================
      "DTF Unified Asymptotic Theory" Simple Example 1, ns =50
                 x1 ==> x2
==========================================================================
```

Equation

$$
\begin{aligned}
x_1(n) &= 0.95\sqrt{2}x_1(n-1) - 0.9025x_1(n-2) + w_1(n) \\
x_2(n) &= -0.5x_1(n-1) + 0.5x_2(n-1) + w_2(n)
\end{aligned}
$$

Equation (8) from Baccala et al. *IEEE Trans Biomed Engin.*, 2016.

Data pre-processing: detrending and normalization options

```
flgDetrend = 1;     % Detrending the data set
flgStandardize = 0; % No standardization
[nChannels,nSegLength] = size(u);
if nChannels > nSegLength, u = u.';
   [nChannels,nSegLength] = size(u);
end;
if flgDetrend,
   for i=1:nChannels, u(i,:) = detrend(u(i,:)); end;
   disp('Time series were detrended.');
end;
if flgStandardize,
   for i=1:nChannels, u(i,:) = u(i,:)/std(u(i,:)); end;
   disp('Time series were scale-standardized.');
end;
```

```
        Time series were detrended.
```

MVAR model estimation

```
maxIP = 30;         % maximum model order to consider.
alg = 1;            % 1: Nutall-Strand MVAR estimation algorithm
criterion = 1;      % 1: AIC, Akaike Information Criteria
disp('Running MVAR estimation and GCT analysis routines.')
[IP,pf,A,pb,B,ef,eb,vaic,Vaicv] = mvar(u,maxIP,alg,criterion);
disp(['Number of channels = ' int2str(nChannels) ' with ' ...
   int2str(nSegLength) ' data points; MAR model order = ' int2str(IP) '.']);
```

```
        Running MVAR estimation and GCT analysis routines.
maxOrder limited to 30
IP=1  vaic=467.864503
IP=2  vaic=403.041777
IP=3  vaic=408.042486

Number of channels = 2 with 50 data points; MAR model order = 2.
```

Testing for adequacy of MAR model fitting through Portmanteau test

```
h = 20; % testing lag
MVARadequacy_signif = 0.05; % VAR model estimation adequacy significance
                            % level
aValueMVAR = 1 - MVARadequacy_signif; % Confidence value for the testing
flgPrintResults = 1;
```

Granger causality test (GCT) and instantaneous GCT

```
gct_signif  = 0.01;  % Granger causality test significance level
igct_signif = 0.01;  % Instantaneous GCT significance level
metric = 'euc';  % euc  = original PDC or DTF;
                 % diag = generalized PDC (gPDC) or directed coherence (DC);
                 % info = information PDC (iPDC) or iDTF.
flgPrintResults = 1;
[Tr_gct, pValue_gct, Tr_igct, pValue_igct] = gct_alg(u,A,pf,gct_signif, ...
                                              igct_signif,flgPrintResults);
```

```
        ----------------------------------------------------------------------
                         GRANGER CAUSALITY TEST
======================================================================
Connectivity matrix:
   NaN     0
     1   NaN
Granger causality test p-values:
       NaN    0.2550
         0       NaN
----------------------------------------------------------------------
                  INSTANTANEOUS GRANGER CAUSALITY TEST
======================================================================
Instantaneous connectivity matrix:
   NaN     0
     0   NaN
Instantaneous Granger causality test p-values:
       NaN    0.9098
    0.9098       NaN
>>>> Instantaneous Granger causality NOT detected.
```

DTF estimation

DTF analysis results are saved in **c** structure. See asymp_dtf.m or issue >> help asymp_dtf command for more detail.

```
metric = 'euc'; % euc  = Euclidian   = original DTF;
%               % diag = diagonal    =  DC;
%               % info = information = iDTF.
nFreqs = 128;
alpha = 0.01;

c = asymp_dtf(u,A,pf,nFreqs,metric,alpha);
```

```
        * Original DTF and asymptotic statistics
```

$|DTF(\lambda)|^2$ Matrix Layout Plotting

```
switch lower(flgPlotStyle)
   case 'print'
      flgColor =[0];       % white background
      flgMax = 'TCI';
      flgSignifColor = 1; % black + gray
      flgScale = 3;       % [0 max(flgMax)]
   otherwise % 'screen'
      flgColor =[1];       % Colored background
      flgMax = 'TCI';
      flgSignifColor = 3; % red + green
      flgScale = 2;       % [0 1]/[0 .1]/[0 .01]
end;

% ---------------Plotting options flag setting-----------------------------
%            [1 2 3 4 5 6 7]
flgPrinting=[1 1 1 2 2 0 1];
%             | | | | | | 7 Spectra(0: w/o SS; 1: Linear; 2: log-scale)
%             | | | | | 6 Coherence
%             | | | | 5 Plot lower confidence limit (legacy)
%             | | | 4 Plot upper confidence limit
%             | | 3 Significant DTF(w) in red line  (legacy)
%             | 2 Patnaik threshold level in black dashed-line
%             1 plot DTF
%---------------------------------------------------------------------------

fs = 1;       % sampling frequency
% axis_scale = [0 0.50 -0.02 1.05];
w = fs*(0:(nFreqs-1))/2/nFreqs;
w_max = fs/2;

h = figure;
set(h,'NumberTitle','off','MenuBar','none', ...
   'Name','[Asymptotic DTF] Fig 1. Example 1 DTF, ns = 50')
[hxlabel hylabel] = xplot(c,...
   flgPrinting,fs,w_max,chLabels,flgColor,flgScale,flgMax,flgSignifColor);

[ax,hT] = suplabel(['Example 1 - simple example; sigma2 = 100,' ...
   ' sigma1 = sigma3 = 1; ns = ' ...
   int2str(ns) '; p = ' int2str(p) ...
   ' \alpha = ' int2str(100*alpha) '%.'],'t');
set(hT,'FontSize',10); % Subtitle font size
drawnow
```

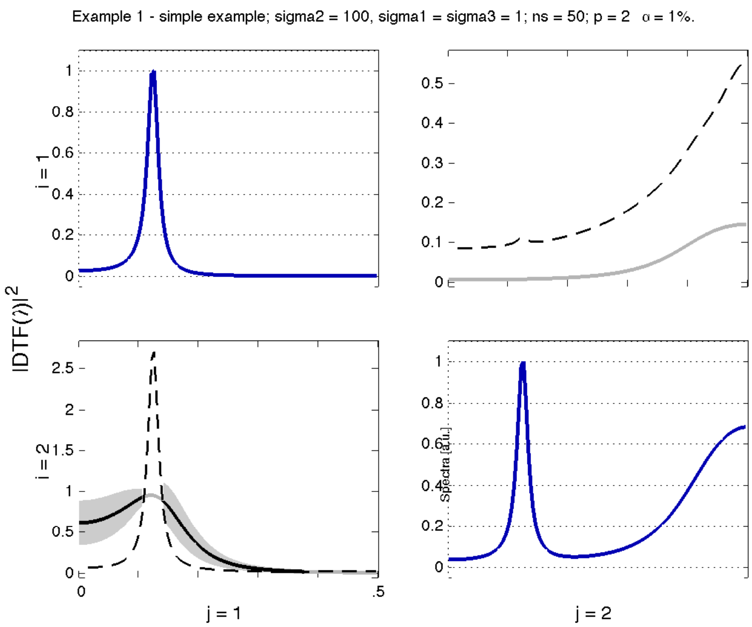
Uncomment the command line bellow to generate an eps output file

% print -depsc2 -painters Fig1_example1_dtf_ns50.eps

PDC estimation

PDC analysis results are saved in **d** structure. See asymp_dtf.m or issue >> help asymp_pdc command for more detail.

```
nFreqs = 128;
metric = 'euc'; % euc  = Euclidian = original PDC;
%               % diag = diagonal   = generalized PDC or gPDC;
%               % info = information = iPDC
```

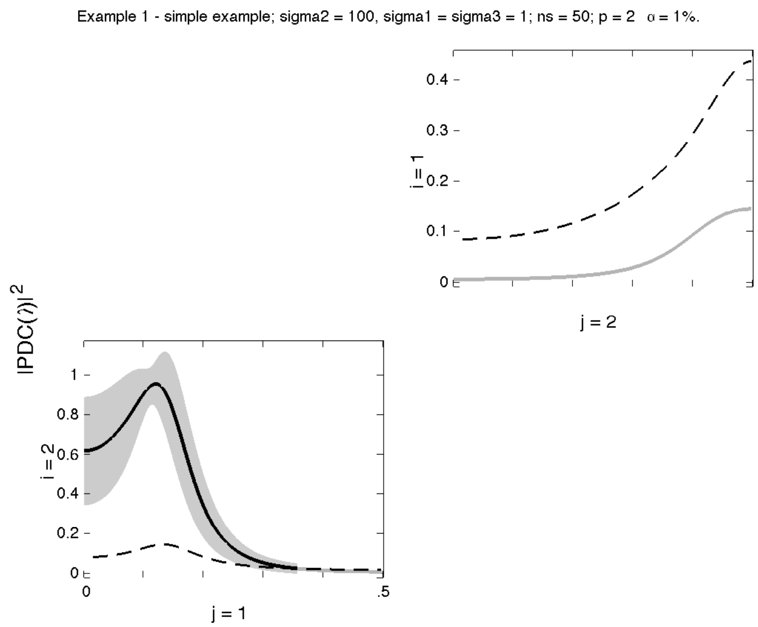



```
alpha = 0.01;

d = asymp_pdc(u,A,pf,nFreqs,metric,alpha);
```

```
        * Original PDC and asymptotic statistics
```

$|PDC(\lambda)|^2$ Matrix Layout Plotting

```
% ---------------Plotting options flag setting-----------------------------
%            [1 2 3 4 5 6 7]
flgPrinting=[1 1 1 2 2 0 0];
%             | | | | | | 7 Spectra(0: w/o SS; 1: Linear; 2: log-scale)
%             | | | | | 6 Coherence
%             | | | | 5 Plot lower confidence limit (legacy)
%             | | | 4 Plot upper confidence limit
%             | | 3 Significant DTF(w) in red line  (legacy)
%             | 2 Patnaik threshold level in black dashed-line
%             1 plot DTF
%-------------------------------------------------------------------------
fs = 1;       % sampling frequency
w_max = fs/2; % x-axis maximum value
h = figure;
set(h,'NumberTitle','off','MenuBar','none', ...
      'Name','[Asymptotic DTF] Fig 1. Example 1 PDC, ns = 50')
[hxlabel hylabel] = xplot(d,...
   flgPrinting,fs,w_max,chLabels,flgColor,flgScale,flgMax,flgSignifColor);

[ax,hT] = suplabel(['Example 1 - simple example; sigma2 = 100,' ...
   ' sigma1 = sigma3 = 1; ns = ' ...
   int2str(ns) '; p = ' int2str(p) ...
   ' \alpha = ' int2str(100*alpha) '%.'],'t');
set(hT,'FontSize',10); % Subtitle font size
drawnow
```



Uncomment the command line bellow to generate an eps output file

%print -depsc2 -painters Fig1_example1_pdc_ns50.eps

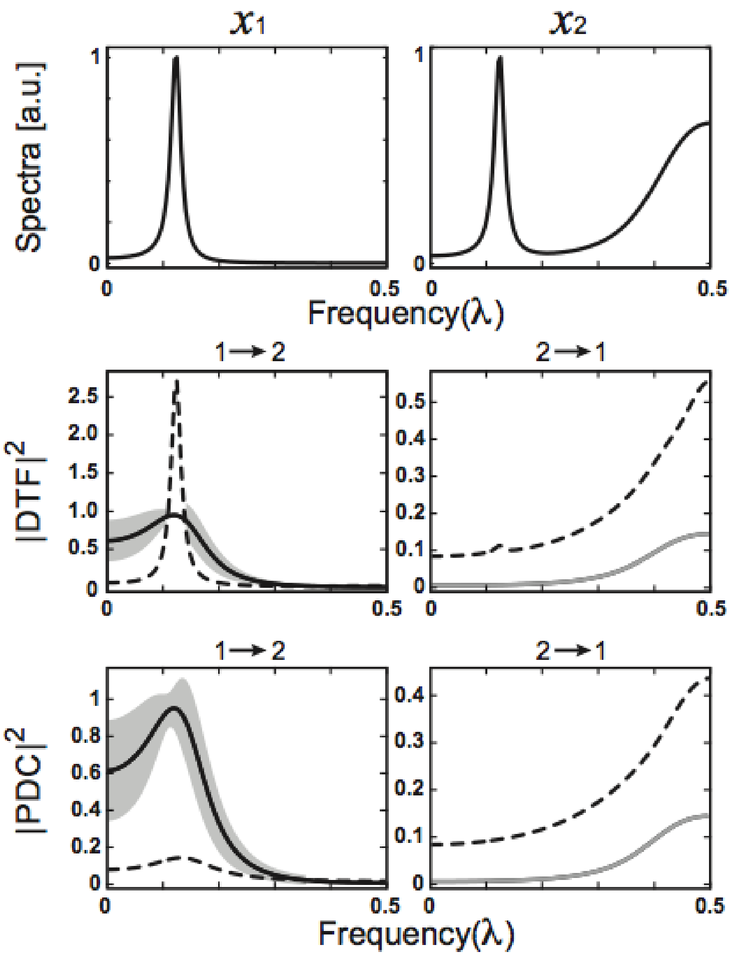
Result from the original article, Baccala et al (2016)

Figure 1, from article.

Some remarks:

1. For the final figure 1, we did some cosmetic edit with AI
2. Erratum: **If we were not able to correct this typo.** In the caption of Figure 1, it states that "Gray shades describe 95% confidence intervals when above threshold.", however as we have used $\alpha = 1\%$, conversely the confidence interval is actually 99%, instead of 95%.

This completes the Figure 1 generation (Baccala et al, 2016)'



x_1
x_2
Spectra [a.u.]
1
0
0
0.5
Frequency(λ)
1→2
2→1
|DTF|^2
2.5
2.0
1.5
1.0
0.5
0.5
0.4
0.3
0.2
0.1
|PDC|^2
1
0.8
0.6
0.4
0.2
0.4
0.3
0.2
0.1
Frequency(λ)