Paper: VEXPA: Validated EXPonential Analysis through regular sub-sampling.pdf

Matteo Briani, Annie Cuyt and Wen-shin Lee

VEXPA: Validated EXPonential Analysis through regular sub-sampling

Outlier experiment 5.1.

Contents

Script environment

This script does depend on the random number generator state.

```
clear
close all
```

Outlier experiment 5.1.

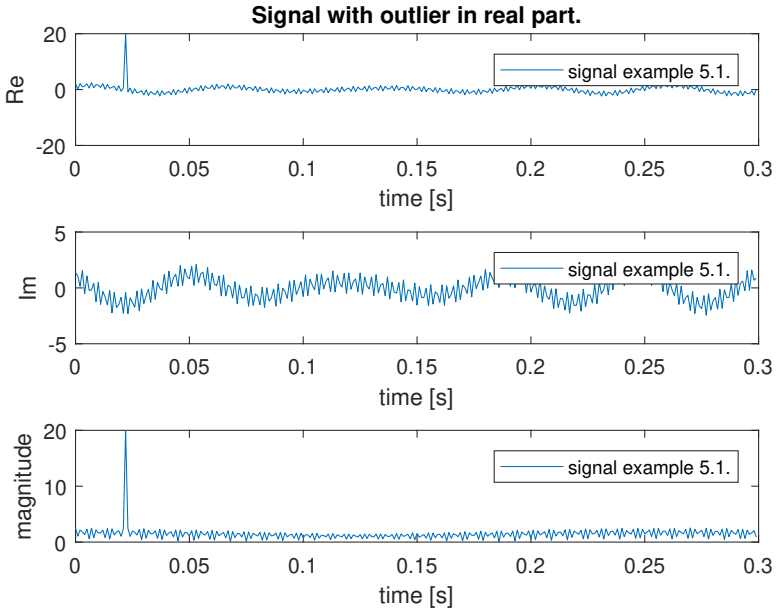
We consider $\phi(t)$ defined by the parameters $\beta_i, \gamma_i, \omega_i, \psi_i,\ i = 1, 2, 3$ listed in Table 1. In addition we add white circular Gaussian noise with SNR = 30 dB and we create an outlier by perturbing sample ϕ_{23} by adding 20 to it. The total number of samples is N = 300.

```
params = params_from_table1;

signal = params.construct(300,'signal example 5.1.');
signal.add_white_gaussian_noise(30,'db');

outlier = zeros(1,300);
outlier(23) = 20;
signal.add_noise(outlier);

plot_signal(signal,'--plot-what','s');
subplot(3,1,1); title('Signal with outlier in real part.');
subplot(3,1,2); title('');
subplot(3,1,3); title('');
```



The full signal is analyzed using the Prony-like algorithm ESPRIT [22], of which the result can be found in the Figures 7 (time domain) indicated with blue squares, and 8 (frequency domain) in blue at the left. In Figure 7 the signal, perturbed by noise and an outlier, is depicted using black triangles.

```
bcsolver = MultiExponentialSolver...
            (BSolverEsprit('--nsamples',300,'--ncols',100,...
                           '--nrows',201,'--nterms',100),...
            CSolverVandermondeLS('--nrows',300,'--delta',1),...
            '--nsamples',300);

params_esprit = bcsolver.solve(signal);
signal_esprit = params_esprit.construct(300);


fig7 = figure;
hold on
plot(real(signal.samples(1:50)),'k^','MarkerSize',7,...
     'LineWidth',1.5)
plot(real(signal_esprit.samples(1:50)),'-bs','LineWidth',1.5)
title({['Figure 7. Outlier experiment with original data '...
       '(black triangles) '],['ESPRIT reconstruction (blue '...
       'squares).']})

figure;
plot_unit_circle
```

```
hold on
plot(params.b,'k^','MarkerSize',7,'LineWidth',1.5)
plot(params_esprit.b,'bs','MarkerSize',7)
title(['Figure 8. (left) The (\omega_i,\beta_i) output '...
       'from ESPRIT.'])
```

Figure 7. Outlier experiment with original data (black triangles) ESPRIT reconstruction (blue squares).

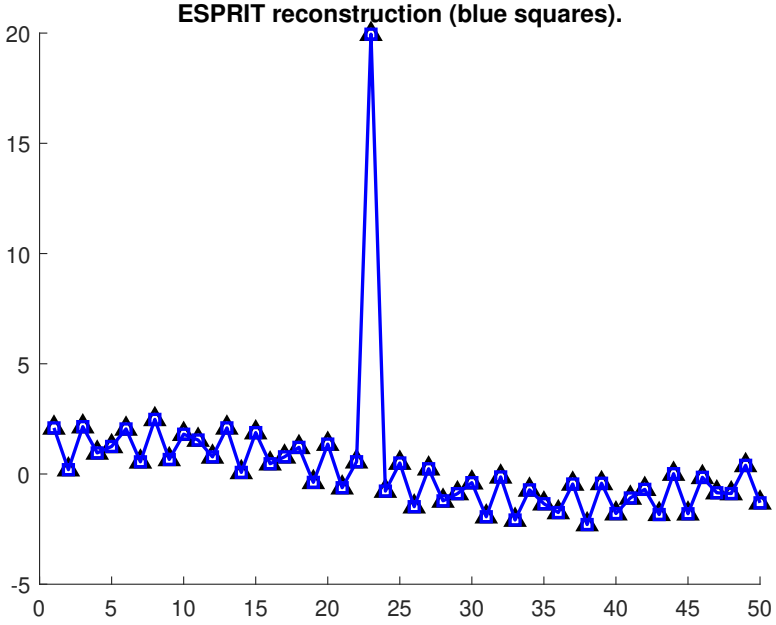
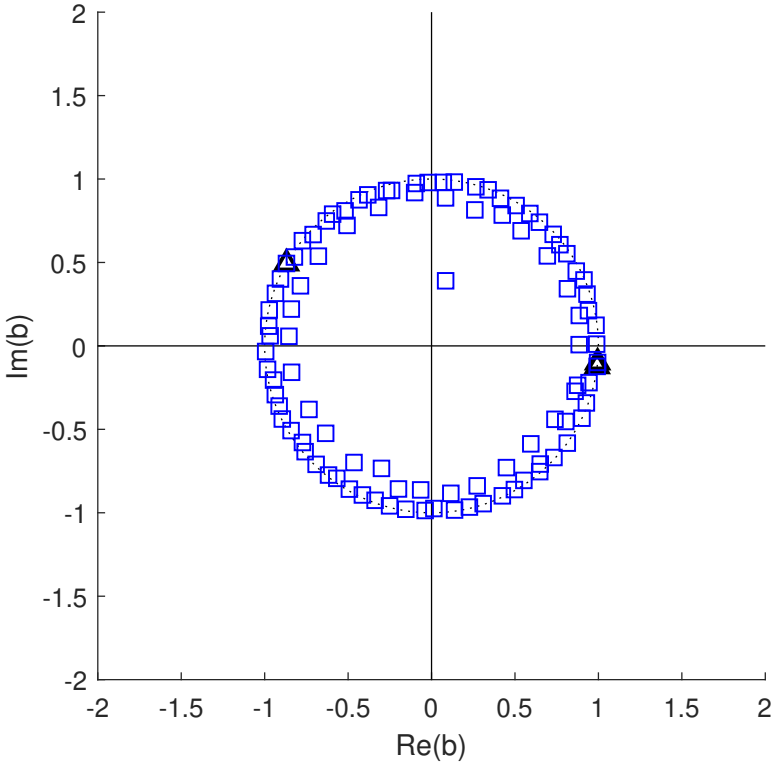


Figure 8. (left) The (ω_i,β_i) output from ESPRIT.



For VEXPA we take $u = 7$ and $s = 11$. So each Φ_k contains 42 or 41 samples. The decimation Φ_2 is the one containing the outlier. So we can expect to find clusters of 6 elements in ${}_uL$ instead of 7. The values ${}^{ms}\alpha_{i,k}$ perturbed by the outlier are the ones where $k = 1$ and $k = 5$. So we expect the clusters in ${}^sL$ to contain 4 elements (only 6 elements in ${}^sL$ are connected to the possible 6 cluster elements in ${}_uL$ to start with). In Figure 9 we show the results of the DBSCAN cluster analysis on ${}_uL$ and on ${}^sL$.

```
u = 7;
s = 11;

bsolver  = BSolverVexpa( '--bsolver',   BSolverEsprit( ...
                                         '--nterms',15) ...
                       , '--csolver',   CSolverVandermondeLS ...
                       , '--rate'   ,   u ...
                       , '--shift'  ,   s ...
                       , '--M'      ,   8 ...
                       , '--u-epsilon', 0.05 ...
                       , '--u-minpts' , 6 ...
                       , '--s-epsilon', 0.1 ...
```

```
                        , '--s-minpts' , 4 ...
                        , '--plot'     , true ...
                        , '--time'     , false ...
                        );

b_vexpa = bsolver.solve(signal);

csolver = CSolverVandermondeLS('--nrows',300,'--delta',1);
c_vexpa = csolver.solve(signal,b_vexpa);

params_vexpa = MultiExponentialParameters(1000, ...
                                          {b_vexpa,c_vexpa}, ...
                                          'normalized');
signal_vexpa = params_vexpa.construct(300);

nfig = get(gcf,'Number');
figure(nfig-4)
title(['Figure 9. (left) Cluster detection in _uL for the '...
       'outlier experiment.'])
figure(nfig-3)
title(['Figure 9. (right) Cluster detection in ^sL for the '...
       'outlier experiment.'])
figure(nfig);close;figure(nfig-1);close;figure(nfig-2);close;
```

Figure 9. (left) Cluster detection in $_{u}$L for the outlier experiment.

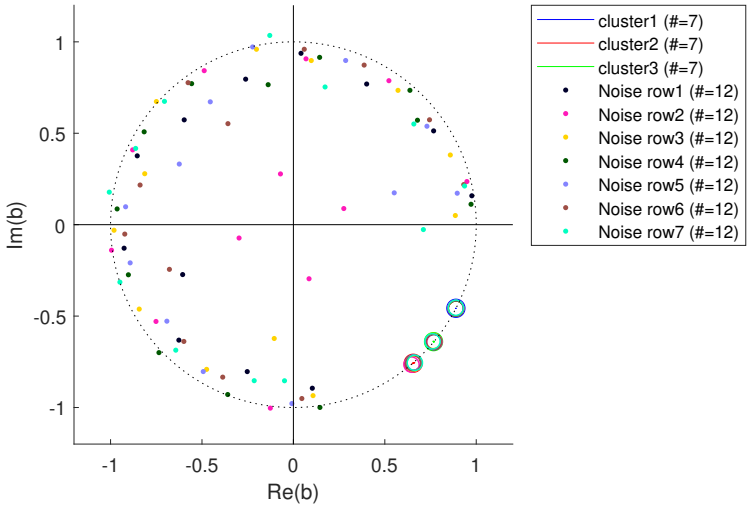
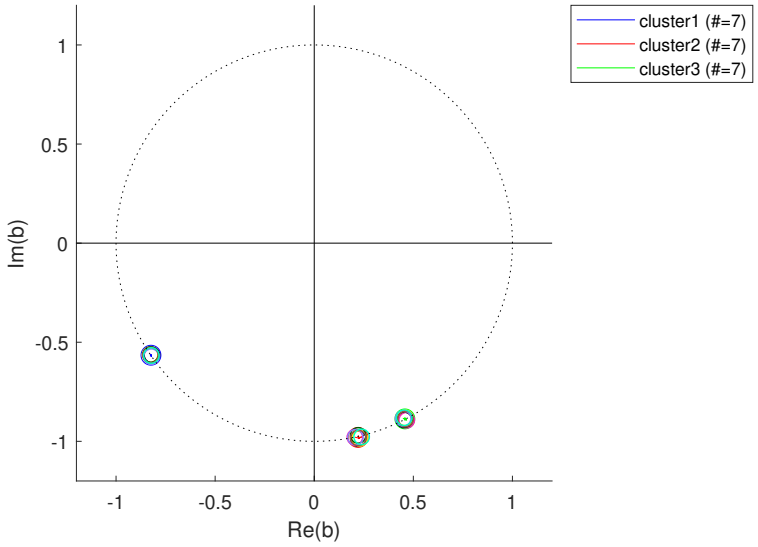


Figure 9. (right) Cluster detection in ^{S}L for the outlier experiment.



The signal reconstructed from the VEXPA output is depicted in Figure 7 using red circles. The spectral output is shown in Figure 8 in red at the right. Both the Figures 7 and 8 illustrate that the stand-alone ESPRIT method suffers from the outlier. The new VEXPA add-on is able to filter out the outlier and reconstruct the original signal because it retrieves the parameters correctly. The ESPRIT implementation introduces, besides the correct frequencies and amplitudes, a lot of additional terms that are hard to discard.

```
if ~isgraphics(fig7)
    fig7 = figure;
end
figure(get(fig7,'Number'))
plot(real(signal_vexpa.samples(1:50)),'-ro','LineWidth',1.5)
title({['Figure 7. Outlier experiment with original data '...
       '(black triangles) '],['ESPRIT reconstruction (blue '...
       'squares) and VEXPA reconstruction '],...
       '(red circles).'})

figure
plot_unit_circle
hold on
plot(params.b,'k^','MarkerSize',7,'LineWidth',1.5)
plot(params_vexpa.b,'ro','MarkerSize',7)
title(['Figure 8. (right) The (\omega_i,\beta_i) output '...
       'from VEXPA.'])
```

Figure 7. Outlier experiment with original data (black triangles) ESPRIT reconstruction (blue squares) and VEXPA reconstruction (red circles).

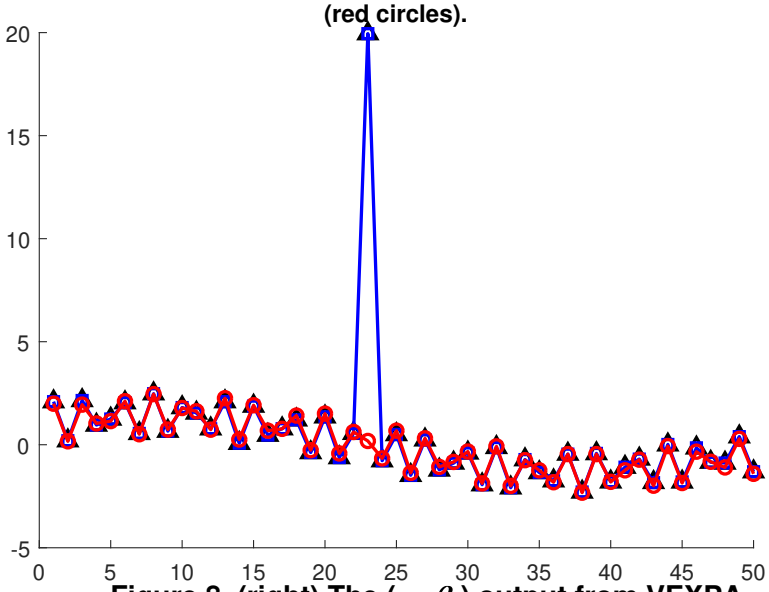


Figure 8. (right) The (ω_i,β_i) output from VEXPA.