Paper: Regular Sparse Array Direction of Arrival in One Dimension.

Ferre Knaepkens, Annie Cuyt, Wen-shin Lee, Dirk de Villiers

Regular Sparse Array Direction of Arrival in One Dimension

V.A. Example 2

Contents

Script environment

This script depends on the random number generator state.

```
clear
close all
warning off;
```

V.A. Example 2

In a second experiment we compare our method to the co-prime array method discussed in [41]. Although the array geometry for both methods is not the same, they are comparable in the sense that they consider two interleaved sparse arrays. In our case we have one array subsampled by a factor σ and then shifted over ρ, while their method just combines the results of two sparse arrays, which are respectively subsampled by factors σ_1 and σ_2. Then, a matching step is performed to be able to link the results of both sparse arrays. This matching is based on a projection of a 2-D point on a 1-D line segment that corresponds to the entire angular domain due to the co-primeness of σ_1 and σ_2.

Similar to the first experiment, we use both methods to retrieve the $n = 6$ angles given in Table II, for increasing noise levels. For this experiment we only consider 6 instead of 10 signals, since the matching in [41] becomes considerably worse for larger n-values.

```
signal.ampl = [0.3, 0.2, 0.4, 0.5, 0.3,...
               0.4];
signal.phase = [0.9, 1.2, 0.8, 0.7, 1.1,...
                0.7]*pi;
signal.angles = deg2rad([35, 62.5, 90, 96.5, 123.5, 151]);
```

For both methods 256 snapshots are collected by two sparse ULAs of 20 elements each, where the distance between the elements of the virtual dense array is 0.48λ, $\sigma = \sigma_1 = 10$ and $\rho = \sigma_2 = 3$. We also pass the number of signals $n = 6$ on to the co-prime array method, while our method is able to detect this number of impending signals automatically. For the cluster analysis we use DBSCAN on ULA1 with $\mu = 218$ for 85 percent validation and $\delta = 0.01, 0.02, 0.04, 0.08, 0.16, 0.32$, while for ULA2 we require 70 percent validation with $\mu = 179$ and $\delta = 0.6$. The results are shown in Fig. 5, where for every noise level we plot the joint output of 100 different noise realizations to observe the effect of noise on both methods.

We see that for low noise levels, both methods perform comparably, however, for higher noise levels, the difference becomes clear. The co-prime arrays have trouble matching the results of both ULAs, hence, especially for high noise levels, we observe that mismatches are not uncommon and thus result in erroneously retrieved angles. On the other hand, our method may not validate all 6 angles in case of high noise levels, however, it also does not yield any erronuous results. Note that it is possible to obtain al 6 angles for a low SNR with the proposed method when we are less strict on the validation part. However, this can lead to angles which are slightly less accurate than the ones returned in this experiment.

```
signal.freq = 1.5*1e6;
c = physconst('LightSpeed');
signal.dist = 0.48*(2*pi*c/signal.freq);
signal.nrelems = [20,20];
signal.rate = 10;
signal.shift = 3;
t = linspace(0,1,256);

cluster_spec.MinPts1 = 218;
cluster_spec.MinPts2 = 179;
cluster_spec.epsvec = [0.01,0.02,0.04,0.08,0.16,0.32];
cluster_spec.eps2 = 0.6;

SNRvec = 5:40;
angles_exp = cell(numel(SNRvec),1);
SNR_exp = cell(numel(SNRvec),1);

nr_exp = 100;
total_exp = nr_exp*numel(SNRvec);
exp_count = 1;

wtbr = waitbar(0,'Please wait...');
for j = 1:numel(SNRvec)
```

```
    SNR = SNRvec(j);

    angles_exp{j} = cell(nr_exp,1);
    SNR_exp{j} = cell(nr_exp,1);
    for k = 1:nr_exp
        waitbar(exp_count/total_exp,wtbr,sprintf(...
             'SNR = %i\nExperiment %i of %i',SNR,k,nr_exp));

        [signal.samples1,signal.samples2] = ...
                                create_DOA_signal(signal,t,SNR);
        angles_exp{j}{k} = DOAsolver(signal,cluster_spec,8);
        angles_exp{j}{k} = rad2deg(angles_exp{j}{k}(:));
        SNR_exp{j}{k} = repmat(SNR,numel(angles_exp{j}{k}),1);
        exp_count = exp_count+1;
    end
    angles_exp{j} = cell2mat(angles_exp{j});
    SNR_exp{j} = cell2mat(SNR_exp{j});
end
close(wtbr)

fig_new = figure;
plot(cell2mat(SNR_exp),cell2mat(angles_exp),'r.','MarkerSize',15)
hold on
plot(repmat([SNRvec(1);SNRvec(end)],1,6),repmat(rad2deg(signal.angles),2,1),'k');
xlabel('SNR (dB)')
ylabel('Angles (degrees)')
title({['Fig. 5. (bottom) The solution of the DOA problem given by ',...
        'Table II for increasing'],['noise levels of the new ',...
        'proposed method.']})
```

Fig. 5. (bottom) The solution of the DOA problem given by Table II for increasing noise levels of the new proposed method.