

AN ANALYSIS OF LMS AND MVDR ON BEAMFORMING APPLICATIONS

EE635 : Digital Signal Processing II, Spring 2000 University of New Haven

Instructor: Dr. Alain Bathelmy

Students : Raheela AMIR ,Wiwat THARATEERAPARB

Contents

Chapter 1 : Introduction.....	1
Chapter 2 : LMS-MVDR Application Responses.....	3
Chapter 3 : MVDR Spatial Power Spectrum.....	7
Appendix.....	11

Chapter 1

Introduction

Adaptive Beamforming

Beamforming is another name for spatial filtering where an array of sensors together with

appropriate signal processing can either direct or block the radiation or the reception of signals in specified directions. Adaptive beamformers are the one in which arrays adapt to noise and interfering signals in such a way that it places nulls in the direction of the source of interference automatically and in real time. By doing so, the output signal to noise ratio of the system increases and the directional response of the system is improved. In order to adapt the arrays for the optimum response, data received from the output of the sensors is used .

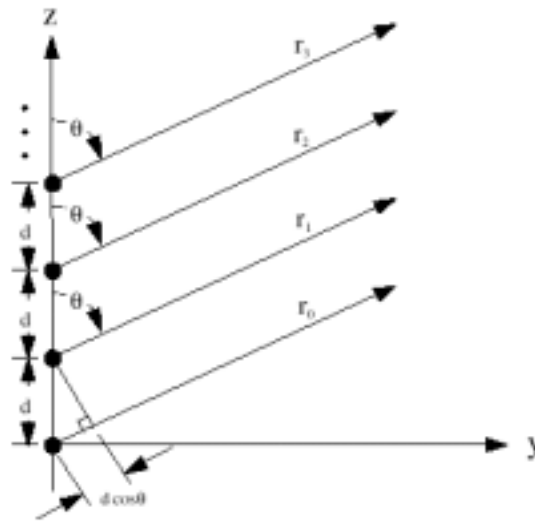


Figure 1: An illustration of incident waves on sensor elements

For our simulation, we consider an adaptive minimum variance distortionless response (MVDR) beamformer consisting of a linear array of five uniformly spaced sensors. The spacing d between the adjacent elements of the arrays equals one half of the received wavelength so as to avoid the appearance of grating lobes. The beamformer operates in an environment that consists of two components: a target signal impinging on the array along a direction of interest, and a single

source of interference originating from an unknown direction. It is further assumed that these two components originate from independent sources and that the received signal includes additive white Gaussian noise at the output of each sensor.

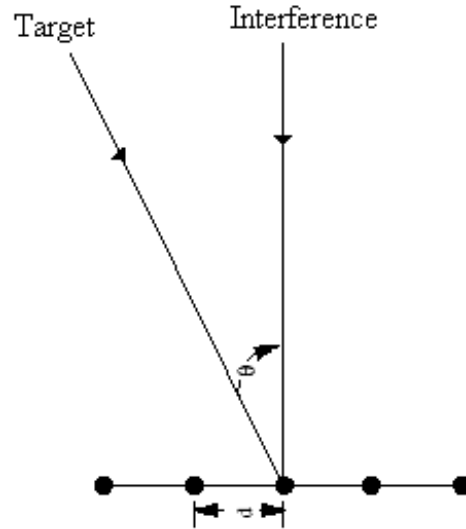


Figure 2 : Linear array antenna for simulation

The angle of incidence of the target and interfacing signal, measured in radians with respect to the normal to the line of the array are as follows:

(i) Target signal: $\theta_{\text{target}} = \sin^{-1}(-0.2)$

(ii) Interference: $\theta_{\text{interf}} = \sin^{-1}(0)$

A beamformer with optimum weight is constrained to pass the target signal with unit response while at the same time minimize the total output variance. The variance minimization process attenuates interference and noise not originating from the target. The optimum weight for this constrained beamformer is defined as:

$$w_0 = \frac{R^{-1}s(\phi_0)}{s^H(\phi_0)R^{-1}s(\phi_0)}$$

Where M is the number of sensors and R is M by M correlation matrix and w_0 is M by 1 optimum weight vector. M by 1 steering vector $s(\phi)$ is defined by:

$$s(\phi) = [1, e^{-j\phi}, \dots, e^{-j\phi(M-1)}]^T$$

Chapter 2

LMS-MVDR Amplitude Responses

LMS algorithm is used to adapt the weights of the beamformer close to the optimum solution. An analysis has been performed by plotting the adapted spatial response of MVDR beamformer by varying interference to noise ratio, signal to noise ratio, number of iterations and step size value of LMS.

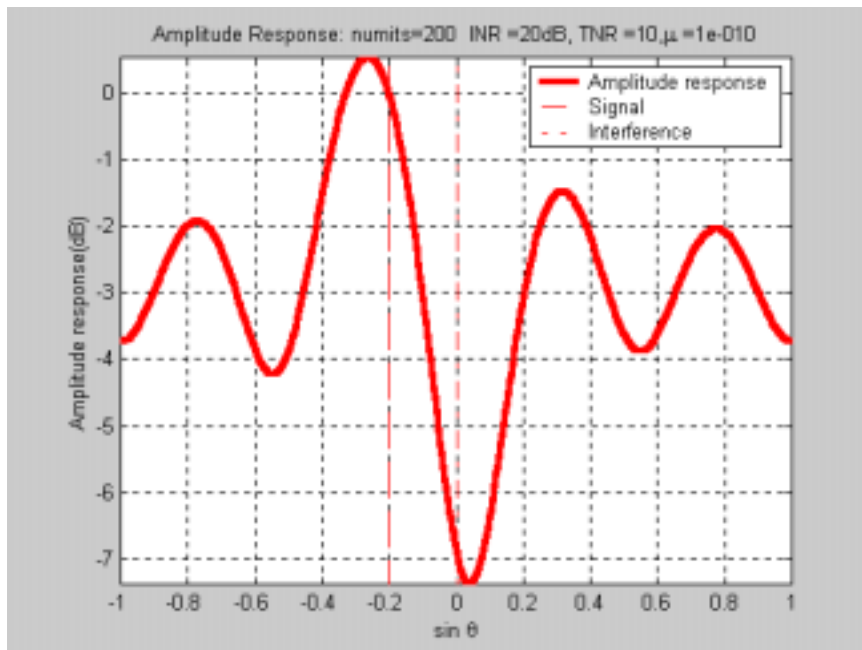


Figure 3: LMS-MVDR beamformer, INR=20dB, TNR=10dB

Figure 3 shows an amplitude spatial response of LMS-MVDR beamformer with interference to noise ratio of 20dB while target to noise ratio is 10dB. The response at target angle (-0.2) is flat (0 dB) while at interference angle (0) is more than -7 dB.

Figure 4 displays amplitude spatial response with varying interference to noise ratio. As it can be observed that stronger the interference signal is compared to noise (higher dB value) better the interference signal is attenuated. In other words, as the ratio of interference-to-target increases, the beamformer improves.

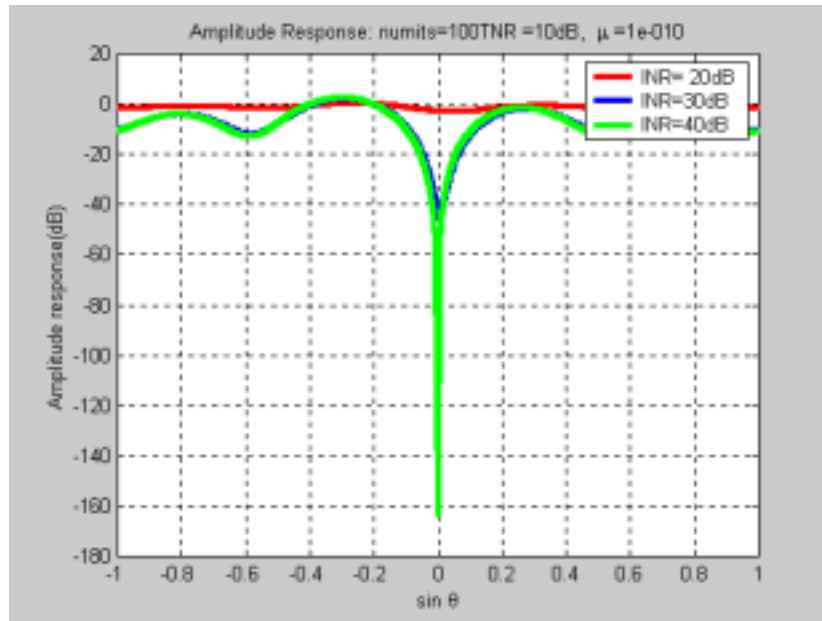


Figure 4: LMS-MVDR beamformer, varying of INR with numits=100

Figure 5 and Figure 6 perform the same analysis as in Figure 4. The only difference is the number of iterations. As it can be observed that for lower interference to noise ratio (INR = 20 dB), higher number of iterations helps in attenuating the interference signal more (see the red line of INR=20dB).

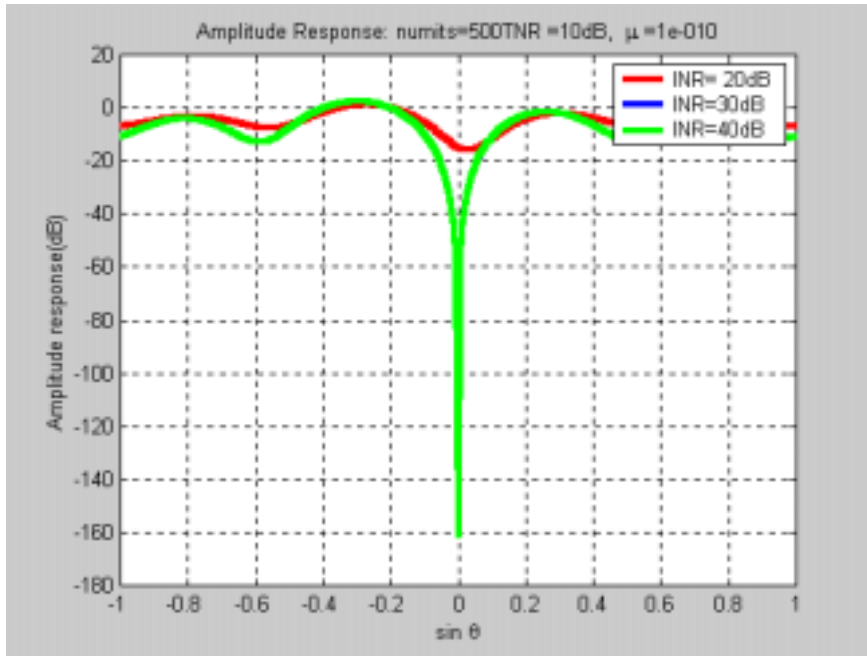


Figure 5: LMS-MVDR beamformer, varying of INR with numits=500

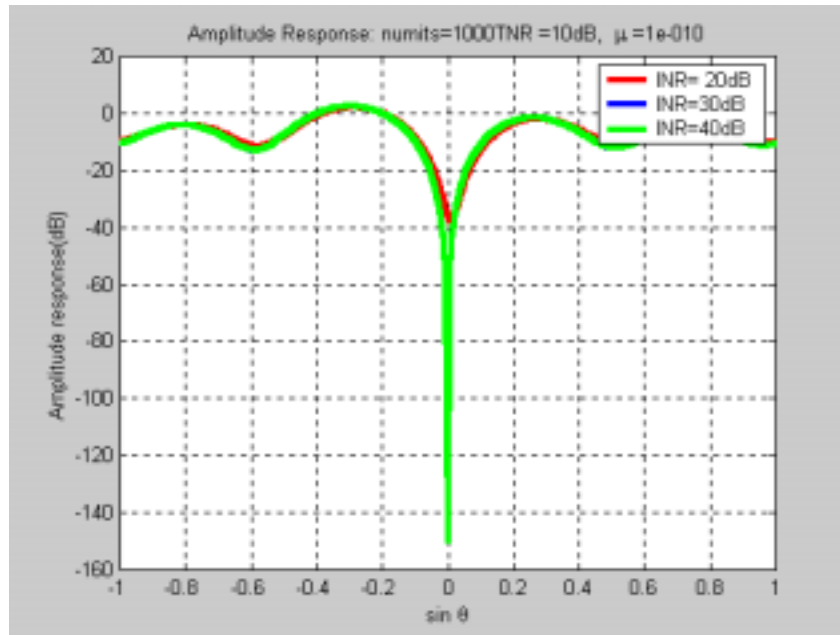


Figure 6: LMS-MVDR beamformer, varying of INR with numits=1000

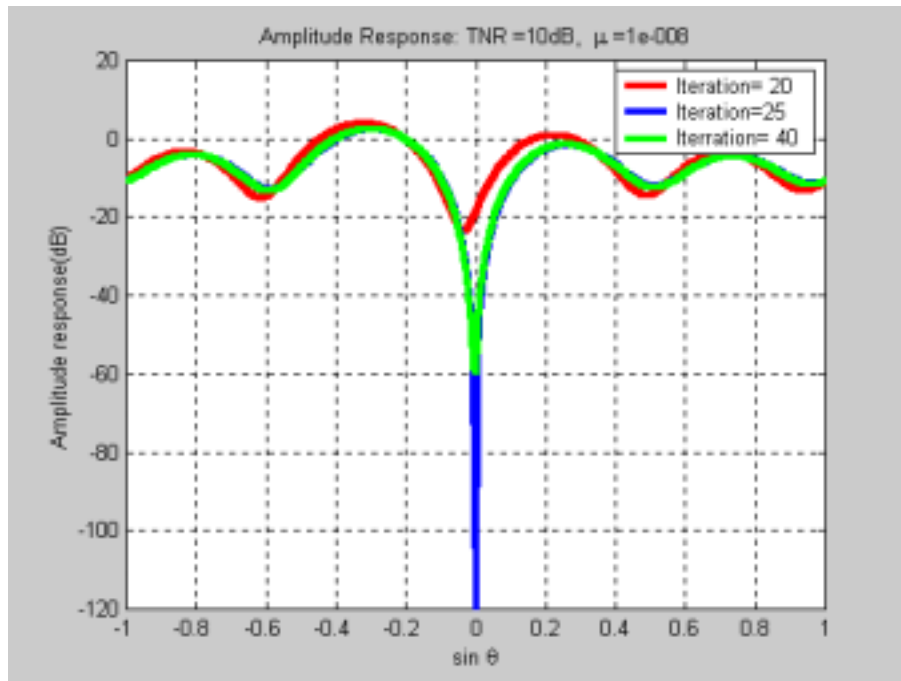


Figure 7: LMS-MVDR beamformer, INR=20dB, TNR=10dB and varying of iterations

Figure 7 displays varying number of iterations with all the other parameters remain the same (INR and TNR are help fixed at 20 and 10 dB respectively). The interference-nulling capability is better as the higher number of iterations.

Conclusion of this chapter:

Response of the MVDR beamformer always show value of unity (0 dB) at the angle of incidence of target (- 0.2) no matter what set of parameters are chosen. The ability to cancel the interference signal depends on

- (i) The number of *iterations* (snapshots of data).
- (ii) By increasing *interference-to-target* signal ratio.

Chapter 3

MVDR Spatial Power Spectrum

An M by M autocorrelation matrix R is generated based on the input received data from the sensors. The spatial MVDR power spectrum can be calculated based on the following formula

$$S_{MVDR}(\phi) = \frac{1}{s^H(\phi)R^{-1}s(\phi)}$$

M by 1 steering vector $s(\phi)$ is defined by :

$$s(\phi) = [1, e^{-j\phi}, \dots, e^{-j\phi(M-1)}]^T$$

Where M is the number of sensors.

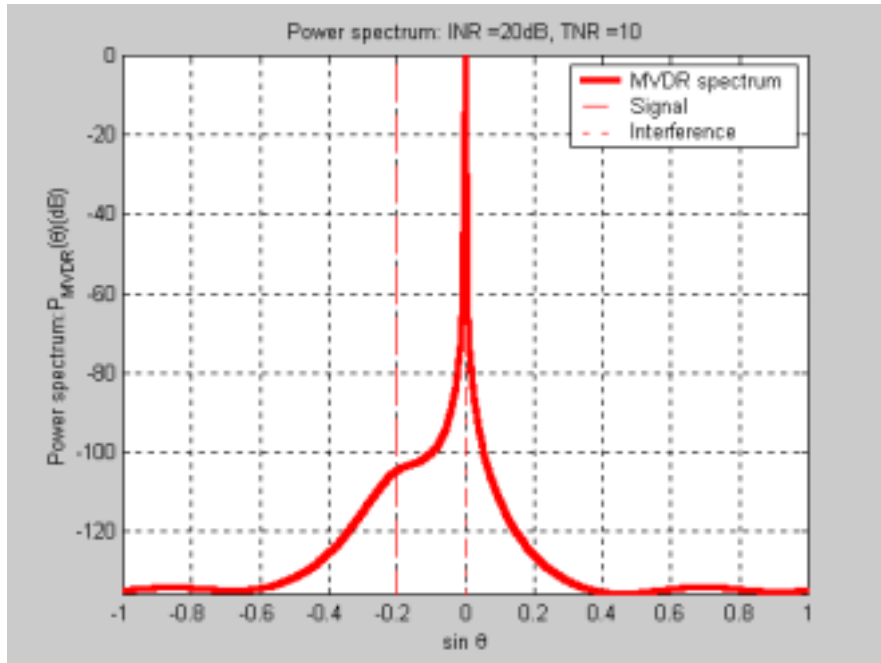


Figure 8: MVDR power spectrum, INR=20dB, TNR=10 dB

Figure 8 displays for the case of interference to noise ratio equals to 20 dB while target to noise ratio is 10 dB. It shows strong interference power at angle 0 compared to target signal power at angle -0.2 . When increasing the INR to 40dB, the MVDR power spectrum is shown in figure 9.

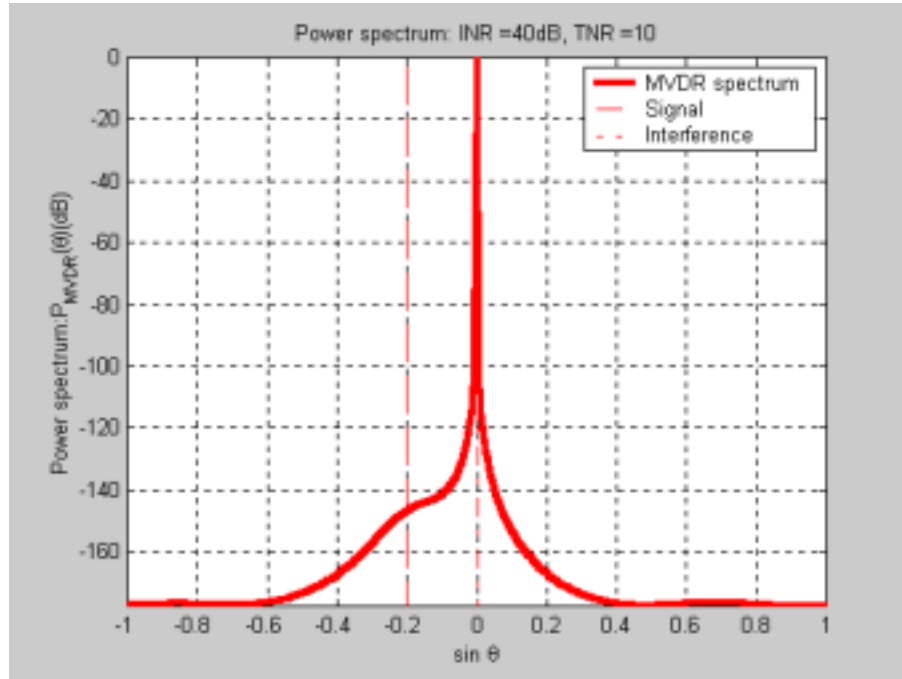


Figure 9: MVDR power spectrum, INR=40dB, TNR=10 dB

At this point, we can see that if the interference-to-target ratio increases, the MVDR power spectrum yields the poor response compared to LMS-MVDR amplitude response. This is because the MVDR power spectrum does not have *interferece-nulling* capability like LMS-MVDR. Actually, the MVDR power spectrum strongly depends on the target signal-to-noise ratio.

By increasing the TNR to 40 and 50 dB in figure 10 and figure 11, the MVDR power spectrum response can distinguish the location of the target and the interference properly.

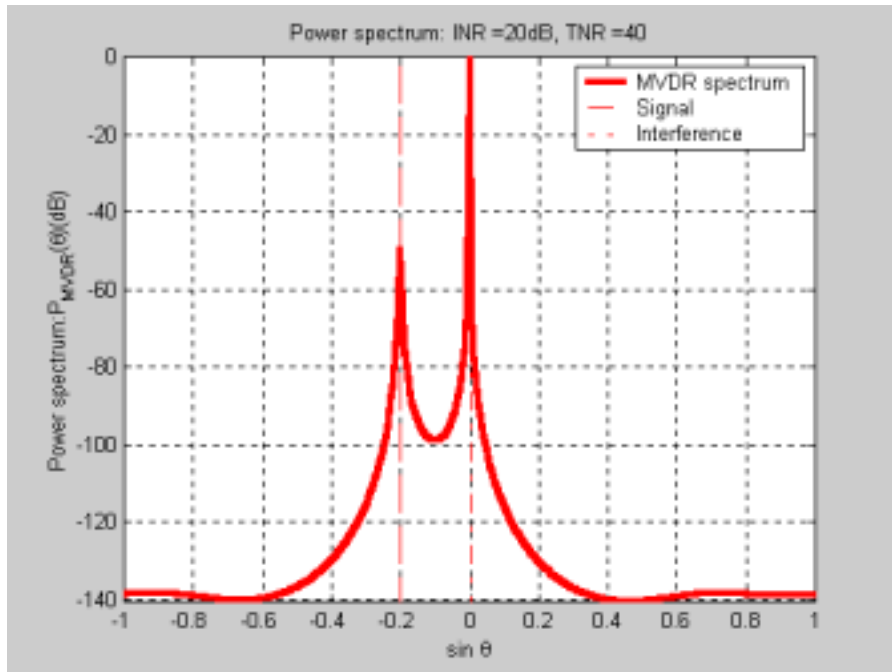


Figure 10: MVDR power spectrum, INR=20dB, TNR=40 dB

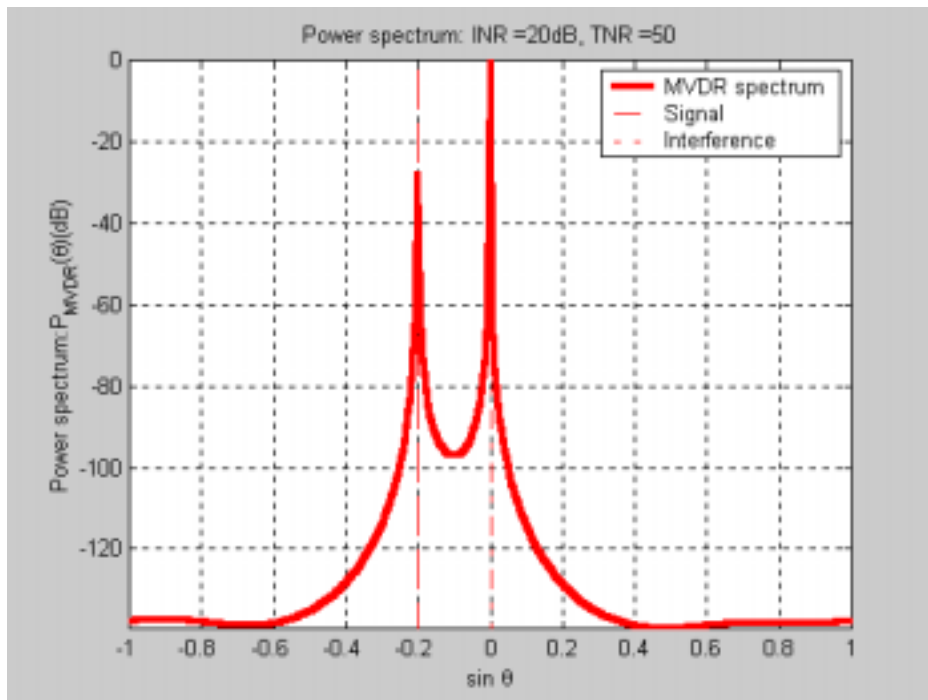


Figure 11: MVDR power spectrum, INR=20dB, TNR=50 dB

Conclusion

From the works we have done so far, we conclude that:

1. The LMS algorithm can be applied to MVDR application. This so-called LMS-MVDR has the capability of interference-nulling. The amplitude response of LMS-MVDR will *suppress* the interfered signal and have the target amplitude held at 0dB.

Moreover, the amplitude response can be improved by:

- (i) Increasing the number of iterations (snapshots).
 - (ii) Increasing the interference-to-target signal ratio.
2. The spatial power spectrum MVDR shows the power of incoming signals but it tries to maximize the interfered signal instead of suppressing. The drawback is the target signal can be overpowered by the interference in the MVDR power spectrum.

Appendix

MATLAB Script

```
clear all;
close all;
clc;
p = 5; % number of sensors
Ninit = p; % number of samples needed for initialization
Nsnaps = 200; % number of snapshots or iteration
Ndata = Ninit + Nsnaps; % total number of data
mean_v = 0; % white noise mean
var_v = 1; % white noise variance
TNRdB = 10; % target signal to noise ratio
INRdB = 40; % interference signal to noise ratio
numst = 301; % resolution in spatial response
sin_theta = [-0.2 0]; % location of signal and interference
phi = pi.*sin_theta; % equivalent electrical angle
A = sqrt(var_v)*10.^[TNRdB INRdB]./20; % parameter for target/interference
signal amplitude, angle

% steering vector along electrical angle of look direction of interest
e = exp(-1j*[1:(p-1)]'*phi(1));

% setup input and output sequences
sig_x = A(1)*exp(1j*[1:p]*phi(1));
for i = 1:Ndata,
    % random disturbances
    v_tmp = sqrt(var_v/2)*randn(2,p)+mean_v;
    v = v_tmp(1,:) + 1j*v_tmp(2,:); % additive white noise of desired mean/variance
    Psi = 2*pi*rand; % uniform random phase on interference
    Xi(i,:) = sig_x + A(2)*exp(1j*[1:p]*phi(2) + Psi) + v;
end;

% setup effective desired output and input vectors from original data
g = 1; % assume unity gain
d = g*Xi(:,1);
u = diag(Xi(:,1))*(ones(Ndata,1)*e.' - Xi(:,2:p));
mu = 1e-10; % step-size parameter

% run beamformer for indicated number of snapshots
[W, xp] = lms(u,d,mu);
Wo = g - W*conj(e);
W = [Wo W];

% now, generate test vectors to compute spatially sampled response
W_H = conj(W(Ndata,:)); % Hermitian transpose of last one
st = linspace(-1,1,numst); % sine(theta) space
est = exp(-1j*pi*[0:(p-1)]'*st); % steering matrix

% amplitude response
P = 20*log10(abs(W_H*est).^2);

%----- MVDR (spatial) Power Spectrum -----%
R = (Xi'*Xi)/Ndata; % correlation matrix from generated data
R_INV = inv(R);
sin_theta = linspace(-1,1,numst);
for i = 1:numst,
    s = exp(-1j*pi*[0:(p-1)]'*sin_theta(i));
    P_MVDR(i) = abs(1/(s'*R_INV*s));
end;
P_MVDR = 20*log10(P_MVDR./max(P_MVDR)); % normalized MVDR

%-----%

% plotting part
```

```

figure;
p1 = plot(st,P,'r-',[sin_theta(1) sin_theta(1)],[min(P) max(P)],'r--',[sin_theta(2)
sin_theta(2)],[min(P) max(P)],'r-.');
set(p1,'LineWidth',3);
%axis([-1 1 min(P) max(P)]);
axis([-1 1 -180 20]);
xlabel('sin \theta');
ylabel('Amplitude response (dB)');
legend('Amplitude response','Signal','Interference');
title(['MVDR beamformer, numts = ',num2str(Nsnaps),' \mu = ',num2str(mu),' , TNR =
',num2str(TNRdB),' , INR = ',num2str(INRdB)]);

```

```

figure;
p2 = plot(sin_theta,P_MVDR,'r-',[sin_theta(1) sin_theta(1)],[min(P_MVDR)
max(P_MVDR)],'r--',[sin_theta(2) sin_theta(2)],[min(P_MVDR) max(P_MVDR)],'r-.');
set(p2,'LineWidth',3);
%axis([-1 1 min(P_MVDR) max(P_MVDR)]);
axis([-1 1 -200 0]);
xlabel('sin \theta');
ylabel('Power spectrum: |P_{MVDR}(\theta)| (dB)');
legend('MVDR spectrum','Signal','Interference');
title(['MVDR powerspectrum beamformer, numts = ',num2str(Nsnaps),' \mu =
',num2str(mu),' , TNR = ',num2str(TNRdB),' , INR = ',num2str(INRdB)]);

```

%%%

```
function [W, e] = lms(u, d, mu);
```

```
% Maximum number of time step that can be predicted
```

```
N = min(size(u, 1), size(d, 1));
```

```
Nin = size(u, 2);
```

```
Nout = size(d, 2);
```

```
% Initialize weight matrix and associated parameters for LMS predictor
```

```
w = zeros(Nout, Nin);
```

```
W = [];
```

```
for n=1:N,
```

```
    W = [W; w];
```

```
    % Predict next sample and error
```

```
    xp(n, :) = u(n, :)*w';
```

```
    e(n, :) = d(n, :)-xp(n, :);
```

```
    % Adapt weight matrix ans step size
```

```
    w = w + mu * e(n, :)' * u(n, :);
```

```
end; % for n t
```

```
return;
```