TR-WEL-0004

003

リアクタンスドメイン信号処理による相関波到来方向推定とエスパアンテナを用いたMIMOの基礎的検討

DOA Estimation of Coherent Waves by Reactance Domain Signal Processing and a Fundamental Study of MIMO for ESPAR Antennas

小川　佳彦，平田　明史，青野　智之，大平　孝

Yoshihiko Ogawa, Akifumi Hirata, Tomoyuki Aono, Takashi Ohira

2004.8.16

(株)国際電気通信基礎技術研究所
波動工学研究所
〒619-0288 京都府相楽郡精華町光台二丁目2番地2
Tel: 0774-95-1501　Fax: 0774-95-1508

Advanced Telecommunications Research Institute International
Wave Engineering Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
Telephone: +81-774-95-1501　Fax: +81-774-95-1508

©2004 (株)国際電気通信基礎技術研究所
©2004 Advanced Telecommunications Research Institute International

目次

1. RD-CUBA-MUSIC

1.1. RD-CUBA-MUSIC の定式化

素子数(*N*+1)の円形配列エスパアンテナにおける RD-CUBA-MUSIC では，図 1 に示すように主ビームの方向を $\theta_0 (=2\pi/N)$ ずつ変えて信号を受信する．ここで，ビームパターンを $b(\theta)$ とすると，複素振幅 s_i，方向 θ_i から到来する信号を各ビームで受信した信号は，

$$\mathbf{y} = \sum_{i=1}^{D} [y_i(0), y_i(\theta_0), ..., y_i((N-1)\theta_0)] + \hat{\mathbf{n}} \tag{1}$$

$$y_i(n\theta_0) = s_i b(n\theta_0 - \theta_i) \quad (n=0,1,.....,N-1) \tag{2}$$

となる．D は到来波数，$\hat{\mathbf{n}}$ は熱雑音ベクトルである．また，$b(n\theta_0)$ $(n=0,1,...,N-1)$ は 2 π を周期とする周期関数であるため離散フーリエ変換を行い，対角行列で表すと，

$$\mathbf{B} = diag(\mathbf{b}\mathbf{F}_{NL}) \qquad with \qquad \mathbf{B} = diag[B(0), B(s_0), ..., B((L-1)s_0)], \qquad s_0 = \frac{1}{2\pi} \tag{3}$$

$$\mathbf{b} = [b(0),\ b(\theta_0),\ .....,b((N-1)\theta_0)] \tag{4}$$

となる．ここで $\mathbf{F}_{NL}$ は以下で与えられる N×L DFT MATRIX である．

$$\mathbf{F}_{NL} = \begin{bmatrix} 1 & ..... & 1 & ..... & 1 \\ 1 & ..... & e^{-j2\pi/N} & ..... & e^{-j2\pi(L-1)/N} \\ : & ..... & : & ..... & : \\ 1 & ..... & e^{-j2\pi(N-1)/N} & .... & e^{-j2\pi(N-1)(L-1)/N} \end{bmatrix} \tag{5}$$

次に式(5)の DFT MATRIX を用いて式 (1)を離散フーリエ変換すると

$$\begin{aligned} \hat{\mathbf{Y}} &= \mathbf{y}\mathbf{F}_{NL} + \hat{\mathbf{N}} \\ &= \sum_{i=1}^{D} [s_i B(0), s_i B(s_0)\exp(-j\theta_i), ..., s_i B((L-1)s_0)\exp(-j(L-1)\theta_i)] + \hat{\mathbf{N}} \\ &= \sum_{i=1}^{D} [s_i, s_i \exp(-j\theta_i), ..., s_i \exp(-j(L-1)\theta_i)]\mathbf{B} + \hat{\mathbf{N}}. \end{aligned} \tag{6}$$

となる．$\hat{\mathbf{N}}$ は熱雑音を離散フーリエ変換した項である．ここで式 (6)を $\mathbf{B}$ で規格化し，*l = 0～L-1* の L 個の要素をベクトル化したものは，

$$\mathbf{Y} = \hat{\mathbf{Y}}\,\mathbf{B}^{-1} = \sum_{i=1}^{D} s_i\ \mathbf{v}(\theta_i) + \mathbf{n} \tag{7}$$

$$\mathbf{v}(\theta) = [1, \exp(-j\theta), ... \exp(-jl\theta), ... \exp(-j(L-1)\theta)] \tag{8}$$

となる．但し，$\mathbf{n}$ は雑音ベクトルに関する項を表す．式(8)に MUSIC を適用するものが RD-CUBA-MUSIC である．モードベクトル $\mathbf{v}(\theta)$ は各要素間では θ だけ位相が変化しており， SSP を用いて信号間の相関を抑圧することでコヒーレント信号の分離が可能となる．

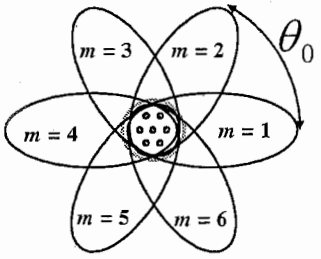


図１．ビームの回転(N=6)

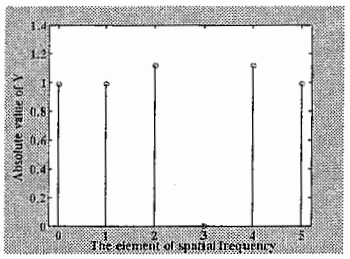


図２．**Y**の絶対値表示
(正のみの空間周波数領域)

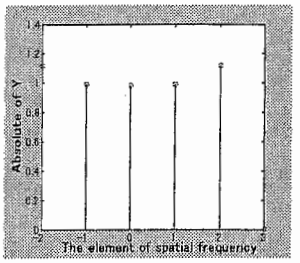


図３．**Y**の絶対値表示
(正負の空間周波数領域)

1.2. 問題点と解決法

２つの問題点と解決法について説明する．

一方は低いレベルのデータが存在するため推定精度が劣化することである．まず，図２に式(7)における**Y**の絶対値を示す．この際，$l=3$に低いレベルのデータ成分が存在することがわかる．このデータ成分を用いて到来方向推定を行なうと推定精度が劣化する．そこでフーリエ変換の際，負領域を含めたフーリエ変換を行ない低いレベルのデータ成分を端にシフトし，その後に取り除くことで低いレベルのデータ成分を利用せずに到来方向推定を行なうこととする．以下に式を用いて示す．方法としては，式(5)を式(9)で表される負領域を含むN×L DFT MATRIXに変換する．このとき，式(9)により式(3)は式(10)になる．

$$\hat{\mathbf{F}}_{NL}=\begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ e^{j2\pi\frac{(L-2)}{2}/N} & & & & e^{-j2\pi\frac{L}{2}/N} \\ & \cdots & 1 & \cdots & \\ \vdots & & \vdots & & \vdots \\ e^{j2\pi\frac{(L-2)}{2}(N-1)/N} & & & & e^{-j2\pi\frac{L}{2}(N-1)/N} \\ & \cdots & 1 & \cdots & \end{bmatrix} \qquad L \quad : \text{even number} \tag{9}$$

$$\mathbf{B} = diag(\mathbf{b}\hat{\mathbf{F}}_{NL}) \quad with \quad \mathbf{B}=diag[B(-\frac{L-2}{2}s_0),....,B(0),...,B(\frac{L}{2}s_0)], \quad s_0=1/2\pi \tag{10}$$

次に式(9)のDFT MATRIXを用いて式 (1)を離散フーリエ変換すると以下となる．

$$\hat{\mathbf{Y}}=\mathbf{y}\mathbf{F}_{NL}+\hat{\mathbf{N}}$$

$$=\sum_{i=1}^{D}[s_i\exp(j\frac{L-2}{2}\theta_i),.....,s_i\exp(-j\frac{L}{2}\theta_i)]\mathbf{B}+\hat{\mathbf{N}} \tag{11}$$

ここで式 (11)を**B**で規格化し，$l=-(L-2)/2$ to $L/2$のL個の要素をベクトル化すると，

$$\mathbf{Y}=\hat{\mathbf{Y}}\mathbf{B}^{-1}=\sum_{i=1}^{D}s_i\,\mathbf{v}(\theta_i)+\mathbf{n} \tag{12}$$

$$\mathbf{v}(\theta)=[\exp(\ j\frac{L-2}{2}\theta),.....,1,.....,\exp(-j\frac{L}{2}\theta)] \tag{13}$$

となり，式(12)における**Y**の絶対値は図３となる．この際，右端に低いレベルのデータ成分がシフトしていることがわかる．そこで，このデータ成分を除去してMUSICを適用する．

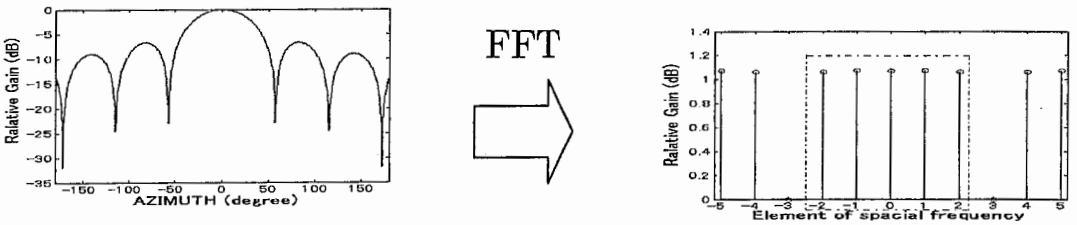


図 4. Sinc 関数で表されるビームパターンと FFT 後のスペクトル

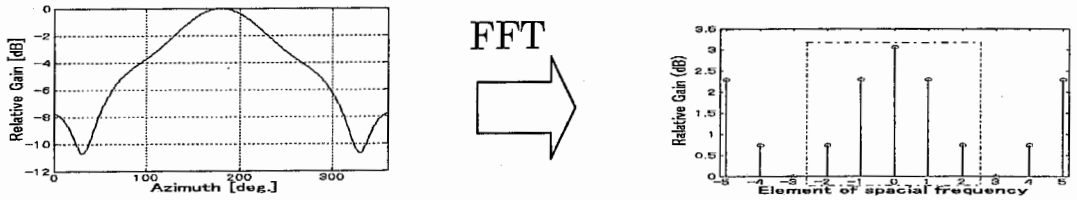


図 5. エスパアンテナで用いるビームパターンと FFT 後のスペクトル

もう一方は規格化を行なう際に雑音成分の無相関性が崩れることである．図 4 に Sinc 関数，図 5 にエスパアンテナ，それぞれで用いるビームパターンとフーリエ変換(FFT)後のスペクトルを示す．Sinc 関数ではフーリエ変換後のスペクトルにおいて点線内の Gain が一定であるが，エスパアンテナでは点線内の Gain が一定にならず，この値を用いて式(7)(12)のように規格化を行なうと雑音成分の無相関性が崩れる．そのため，式(1)(6)(11)における雑音項は熱雑音であるため無相関であるが，規格化された式(7)(12)における雑音項**n**は無相関が崩れてしまう．そこで規格化を行なう前に雑音成分を軽減する．この方法を以下で説明する．まず式(1)を用いて相関行列を求める．

$$\mathbf{R}_{yy} = E[\mathbf{y}^H \mathbf{y}] \tag{14}$$

ここで $E[\cdot]$はエルゴード性を仮定した時間平均，添え字 H はエルミート転置を表す．これを固有値分解すると

$$\begin{aligned} \mathbf{R}_{yy} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^H \qquad & with \qquad \mathbf{\Gamma} = [\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, ..., \hat{\mathbf{e}}_N] \\ & \mathbf{\Lambda} = \quad diag[\lambda_1, \lambda_2, ..., \lambda_N] \\ & \lambda_{D+1} = \cdots = \lambda_N = \sigma^2 \end{aligned} \tag{15}$$

が得られる． ここで $\hat{\mathbf{e}}_i$ ，λ_i は部分空間を張る第 i 固有ベクトル，第 i 固有値を，D は到来波数を表す．また，σ^2 は熱雑音電力であり熱雑音電力の推定値は

$$\hat{\sigma}^2 = \frac{1}{N-D} \sum_{i=D+1}^{N} \lambda_i \tag{16}$$

で表される．式(15)で示す固有値には熱雑音電力が含まれるため，固有値から式(16)で表される雑音電力成分を取り除くことで雑音軽減が可能である．この信号系列は以下の式で与えられる．

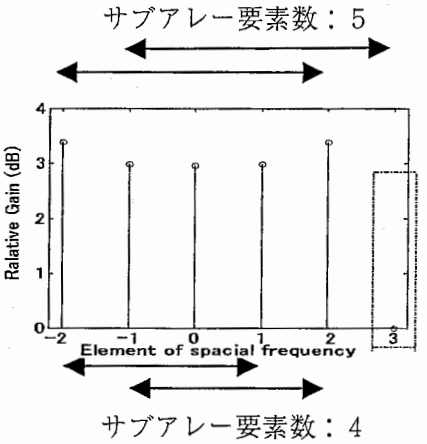
$$\sum_{i=1}^{D} (\lambda_i - \hat{\sigma}^2)\hat{\mathbf{e}}_i \quad \propto \quad \mathbf{y} - \hat{\mathbf{n}} \tag{17}$$

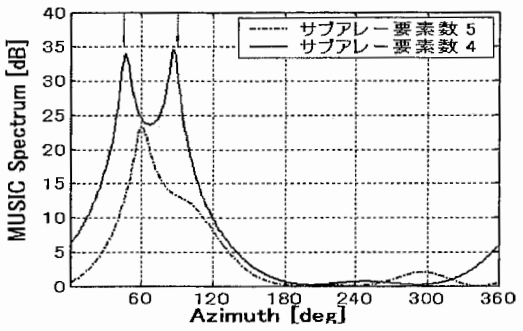
1.3. 電波暗室実験による評価と計算機シミュレーション評価

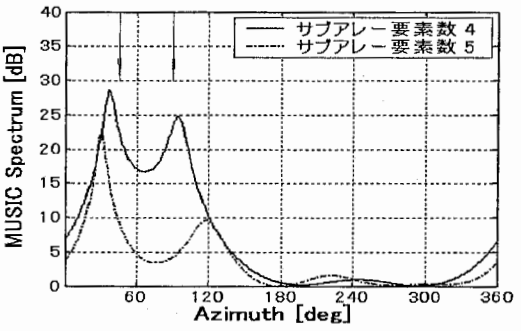
表1にシミュレーション諸元を示す．図 6 は低いレベル成分対策の評価，図7は雑音成分の相関性対策の評価，それぞれにおける計算機シミュレーション結果と電波暗室内実験での結果である．ここで電波暗室内実験の結果は 1.4 における文献を参照した．また，図8に電波暗室内実験における3波コヒーレント波の推定結果を示す．

表1．シミュレーション諸元

素子数	7素子エスパアンテナ(N= 6)
空間周波数要素	L=6
サブアレー数	2 (Forward-Backward SSP)
スナップショット	1000
SNR	20(dB)
送信信号	BPSK
到来波	コヒーレント波



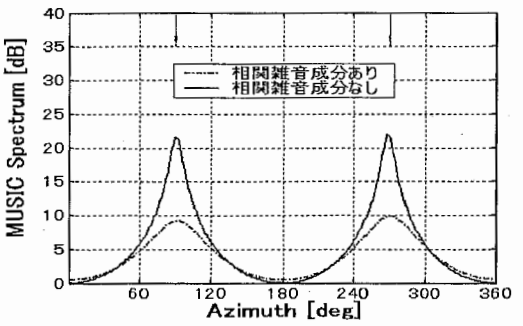


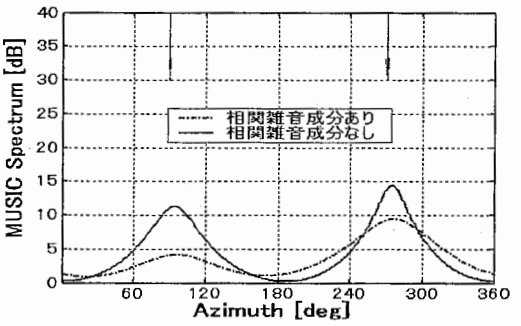


(I) 計算機シミュレーションによる結果 (II) 電波暗室内実験よる結果

(到来方向 45,90 度，推定方向 47,88 度) (到来方向 45,90 度，推定方向 37,94 度)

図6　低いレベル成分対策の評価

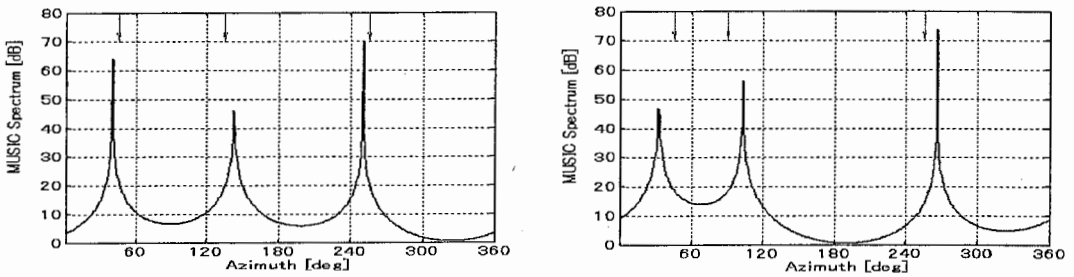




(I) 計算機シミュレーションによる結果 (II) 電波暗室内実験よる結果

(到来方向 90,270 度，推定方向 91,270 度) (到来方向 90,270 度，推定方向 94,274 度)

図7　雑音成分の相関性対策の評価



(I) 到来方向 45,135,255 度
推定方向 40,142,250 度

(II) 到来方向 45,90,255 度
推定方向 32,103,266 度

図 8. 電波暗室におけるコヒーレント 3 波の到来方向推定実験

図 6 において，サブアレー要素数 4 が低いレベル成分のデータを取り除いたもの，サブアレー要素数 5 が低いレベル成分のデータも用いているものである．これらの結果より，計算機シミュレーション，電波暗室内実験どちらにおいてもサブアレー要素数 4 における推定では到来方向を示す矢印付近にピークが向いているが，サブアレー要素数 5 の推定ではピークが 1 箇所しか現れていない，もしくは矢印から大きく離れていることがわかる．このことより，低いレベルのデータ成分を除去することで高精度の推定が可能になることがわかる．

図 7 では，雑音成分の相関を軽減することで，矢印方向のピークが鋭くなることがわかる．このことより，規格化前に雑音成分を軽減する方法が有効であることがわかる．

図 8 では，到来方向の間隔が狭くなるにつれ誤差は大きくなる傾向にあるが，コヒーレント 3 波の到来方向推定が可能であることがわかる．

参考文献

[1] A.Richter and R.S.Thomä, "CUBA-ESPRIT for Angle Estimation with Circular Uniform Beam Arrays", Millennium Conference on Antennas and Propagation (AP2000), CD-ROM, 1156, April 2000.

[2] 高梨 昌樹, 田辺 康彦, 西村 寿彦, 小川 恭孝, 大鐘 武雄, "CUBA-MUSIC を用いたコヒーレント波の到来方向推定", 信学総大, B-1-17, pp.33 . March 2002.

[3] 高梨 昌樹, 西村 寿彦, 小川 恭孝, 大鐘 武雄, "円柱アレーを用いた二次元 DOA 推定に関する検討", 信学技報, AP2002-53,July 2002.

[4] 小川 佳彦,平田 明史, 山田 寛喜, 大平 孝, "エスパアンテナを用いた CUBA-MUSIC 法によるコヒーレント波の到来方向推定",信学総大,B-1-271,March 2004.

[5] 小川 佳彦,平田 明史, 山田 寛喜, 大平 孝, "7 素子エスパアンテナによる RD-CUBA-MUSIC 到来方向推定実験",信学技報,AP.2004-5,(2004-4).

[6] Yoshihiko Ogawa, Akifumi Hirata, Hiroyoshi Yamada, Takashi Ohira, "Experiment of DOA Estimation with RD-CUBA-MUSIC Using 7-element ESPAR", 2004 International Symposium on Antennas and Propagation , Aug. 21, 2004

```
%%% Spatial Smoothing with CUBA

clc;
clear all;

n = 1000;                                   %number of data
deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;d=lambda/4;
M = 6 ;                                     %numbers of sensors
L = 6;
noise_rem = 2;                              % 1:ノイズ軽減なし, 2:ノイズ軽減あり

Pn = -15; sigma2 = 10^(Pn/10);             % [dB]:noise level
%psk = [pi*1.25 pi*0.75 pi*1.75 pi*0.25];   %%% in case of QPSK
psk = [0 1]*pi;                             %%% in case of BPSK

theta = [1:1.0:360];            %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M;

%%% steering vector %%%
V = exp(j*2*pi*d/lambda*(cos(theta(ones(M,1),:)*deg-phi(ones(1,length(theta)),:).')));    %%%
atheta(1,1:360) = 1 ;
for in = 1:M
    atheta(in+1,:) = V(in,:);
end


z0 = 50;  vvs = 100;

% リアクタンス値
y00 =  0.00023344-j*0.0066918;    %補正
y10 = -0.0001816 +j*0.00242469;
y11 =  0.00300676-j*0.0044176;
y21 =  0.00097427+j*0.00299986;
y31 = -0.0003066 -j*0.0003067;
y41 = -0.0000973 -j*0.0001353;

% リアクタンス行列
Y = [y00 y10 y10 y10 y10 y10 y10;
     y10 y11 y21 y31 y41 y31 y21;
     y10 y21 y11 y21 y31 y41 y31;
     y10 y31 y21 y11 y21 y31 y41;
     y10 y41 y31 y21 y11 y21 y31;
     y10 y31 y41 y31 y21 y11 y21;
     y10 y21 y31 y41 y31 y21 y11];

Y0 = Y(:,1);
I = eye(M+1,M+1);
Z = inv(Y);

min = -j*90 ; max = 0 ;
xno = [ 0   0   0   0   0   0   0   ;
        max min max max max max max  ;
        max max min max max max max  ;
        max max max min max max max  ;
        max max max max min max max  ;
        max max max max max min max  ;
        max max max max max max min ];
clear max


%%%%% weight %%%%%
iin = [];
for jloop = 1:M+1
    X = diag(xno(jloop,:));
    X(1,1) = z0*2;
    MX = I+Y*X;
    MXinv = inv(MX);
    ii = vvs*MXinv*Y0;
    iin = [iin; ii.'];
end

%%% calibration %%%
```

```
%%%%%%%%%%%%% 13素子エスパアンテナ形状のステアリングベクトル0度から1度刻みで360度まで %%%%%%%%%%%%%%%%%% ↙
%%%

theta = [0:1:359];        %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M; %


%%% weight without omni pattern %%%
iin=iin(2:end,:);
wei = iin ;
pat = wei*atheta;
Wei=wei ;



%%% DOA Setting
ddeg = [40 110 320];      % Driection Of Arrival
ths = [ddeg];
Psig = [0 0 0];           % Power of signal
kw = [2 1];               % 1 : 無相関波,  2 : 相関波
D = length(ths);          % number of signal
As2 = 10.^(Psig/10);

%%% beam space
[yt,signal] = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M, Wei);


switch noise_rem
case 2                %%% reduce the noise componet
    Ryy = yt*yt'/n ;
    [Uy, Dy, vy] = svd(Ryy);

    diagDy=diag(Dy);
    sigma=sqrt(sum(diagDy(D+1:end))/(M-D));

    yt1=zeros(length(diagDy),1);

    for kk=1:D
         yt1=yt1+(Dy(kk,kk)-sigma^2)*Uy(:,kk) ;
    end
    yt=yt1;
end

%%%%% CUBA %%%%%

b = [pat(1,360)];
for kk=1:M-1
   b = [b pat(1,360*kk/M)];
end

ml = [0:M-1]'*[-2:3];             % lの設定
ex = exp(-j*2*pi*ml/M);

B = (b*ex).';
Cp = (yt.'*ex).';

switch noise_rem

case 1
   for k = 1:n
       xt(:,k) = Cp(:,k)./B;
   end

case 2
   xt = Cp./B;
end

Rxx = xt*xt'/n;
[Ux, Dx, vx] = svd(Rxx);


%%% spatial smoothing by 3 directions
K = L-2;                % number of subarray
```

```
order = [1:5];

% ＃1
ordElem1 = order(1,:);
vtheta = exp(-j*[0:L-1].'*theta*deg);    % mode vector %

[dino1, Pvmu] = spectrumSpatialSmoothing(ordElem1, Rxx, vtheta, K, L, 0, D);
[pMax,index] = max(Pvmu);
plotSpectrum(ths, Psig, theta, D, Pvmu, index, pMax, 60);
```

```
%%% Spatial Smoothing with CUBA
%%% Esper Antenna with 13 element

clc;
clear all;

n = 1000;                                          %number of data
deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;d=lambda/4;
M = 12 ;                                           %numbers of sensors
L = 12;

Pn = -3000; sigma2 = 10^(Pn/10);                   % [dB]:noise level
%psk = [pi*1.25 pi*0.75 pi*1.75 pi*0.25];          %%% in case of QPSK
psk = [0 1]*pi;                                    %%% in case of BPSK

% create response vector
theta = [1:1.0:360];        %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M;

%% steering vector
V = exp(j*2*pi*d/lambda*(cos(theta(ones(M,1),:)*deg-phi(ones(1,length(theta)),:).')));     %%%
atheta(1,1:360) = 1 ;
for in = 1:M
    atheta(in+1,:) = V(in,:);
end

% reactance
 y00= 0.00067051910000-0.01008076000000*j;          %%% espar20031203.dat
 y10=-0.00029015030000+0.00152941430000*j;
 y11= 0.00117776600000-0.01710948100000*j;
 y21= 0.00080483640000+0.01077154000000*j;
 y31= 0.00008289476000-0.00123396900000*j;
 y41=-0.00017468471000-0.00019989180000*j;
 y51=-0.00009107783000-0.00006201371000*j;
 y61= 0.00000130228800-0.00006929386200*j;
 y71= 0.00002199928000-0.00009529652000*j;

z0 = 50; vvs = 100;
% reactance matrix
Y = [y00 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10;
     y10 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21;
     y10 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31;
     y10 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41;
     y10 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51;
     y10 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61;
     y10 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71;
     y10 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61;
     y10 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51;
     y10 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41;
     y10 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31;
     y10 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21;
     y10 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11];

Y0 = Y(:,1);
I = eye(M+1,M+1);
Z = inv(Y);
Zs = z0*2*eye(M+1,M+1);
ZC = Zs*inv(Z+Zs);
x=j./ZC.*(Z.'*ZC);

min=-j*58;                  % -52.632+j*47.283
max=-j*0;                   % -j*100
xno = [ 0   0   0   0   0   0   0   0   0   0   0   0   0;
        max min max max max max max max max max max max max;
        max max min max max max max max max max max max max;
        max max max min max max max max max max max max max;
        max max max max min max max max max max max max max;
        max max max max max min max max max max max max max;
        max max max max max max min max max max max max max;
        max max max max max max max min max max max max max;
        max max max max max max max max min max max max max;
        max max max max max max max max max min max max max;
        max max max max max max max max max max min max max;
        max max max max max max max max max max max min max;
```

```
       max max max max max max max max max max max min max;
       max max max max max max max max max max max max min];
clear max

%% equivalent weight vector
iin = [];
for jloop = 1:M+1
    X = diag(xno(jloop,:));
    X(1,1) = z0*2;
    MX = I+Y*X;
    MXinv = inv(MX);
    ii = vvs*MXinv*Y0;
    iin = [iin; ii.'];
end

theta = [0:1:359];          %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M; %

% weight without omni pattern
iin=iin(2:end,:);
wei = iin ;

% beam pattern
pat = wei*atheta;
Wei=wei ;

%%% DOA Setting
ddeg = [40];                % Direction of signal
ths = [ddeg];
Psig = [0 0 0 0];           % power of signal
kw = [1];                   % type of signal
D = length(ths);            % number of signal
As2 = 10.^(Psig/10);

%%% beam space
yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M, Wei);

%%% CUBA
b = [pat(1,360)];
for kk=1:M-1
   b = [b pat(1,360*kk/M)];
end

ml = [0:M-1]'*[-5:6];
ex = exp(-j*2*pi*ml/M);
B = (b*ex).';

Cp = (yt.'*ex).';
for k = 1:n
    xt(:,k) = Cp(:,k)./B;
end

Rxx = xt*xt'/n;
[Ux, Dx, vx] = svd(Rxx);

vtheta = exp(-j*[0:L-1].'*theta*deg);    %%%

%%% spatial smoothing by 3 directions
K = L-1;                            %number of subarray
order = [1:11];

% ＃1
ordElem1 = order(1,:);
vtheta = exp(-j*[0:L-1].'*theta*deg);    %%%

[dino1, Pvmu] = spectrumSpatialSmoothing(ordElem1, Rxx, vtheta, K, k, 0, D);

[pMax,index] = max(Pvmu);
plotSpectrum(ths, Psig, theta, D, Pvmu, index, pMax, 75);
```

```
function [yt,signal] = beamSrfsignal(doa, atheta, sigma2, kw, n, qpskiq, m, iin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  doa : direciton of signal
%  atheta : steering vector
%  sigma2 : SNR
%  kw : type of signal
%  n  :  number of data
%  iin : weight vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% DOA
ths = doa;
L = length(ths);
Psig = zeros(1,L);
As2 = 10.^(Psig/10);

Vs = [];
for iv = 1:L
    iths = ths(iv);
    Vs = [Vs atheta(:,iths)];
end

% reactance domain
%
%     seed = 54321; rand('seed',seed); randn('seed',seed);

an = sqrt(sigma2); as = sqrt(As2);
nwav = length(kw);                      %%%
isousa = zeros(1,L);

signal = [];

stphi = qpskiq(randint(nwav,n,length(qpskiq))+1);
kk = 0;
for in = 1:nwav
    psi = 0;
    ks = kk+1; kk = kk+kw(in);
    for ik = ks:kk
        As_psi = as(ik)*exp(j*(psi+stphi(in,:)+isousa(ik)'));          %setting initial phase
        signal = [signal; As_psi];
    end
end

yt = iin*Vs*signal+(an/sqrt(2))*(randn(m,n)+j*randn(m,n));
```

```
function [dino, Pmusic3] = spectrumSpatialSmoothing(ord, Rxx, vtheta, K, L,eleva,D)

    Rx1 = [Rxx(ord(ord),:)];
    Rx2 = [Rx1(:,ord(ord))];

    % foward / backward SSP
    Rss = fbeespCoherentssp(Rx2,K);

    [Us,Ds,vs] = svd(Rss);

    %% MUSIC
    vtheta = vtheta(1:K,:);
    numv = real(diag(vtheta'*vtheta)).';
    VUx = vtheta'*Us(:,D+1:end);
    Pmusic3 = 10*log10(numv./sum(VUx.*conj(VUx),2).');
    [pMax,index] = max(Pmusic3);
    dino=0;                 %dummy
```

```
%%% foward / backward ssp.m

function Ryy = fbeespCoherentssp(cr,K);
% spatial smoothing

[M,MM]=size(cr);
N = M-K+1 ;
J = fliplr(eye(M)) ;
crfb = (cr + J*cr.'*J)/2 ;
Ryy = zeros(K,K);

for in = 1:N
     Ryy = Ryy + crfb(in:in+K-1,in:in+K-1);
end

Ryy = Ryy / N;

% End coherentssp.m
```

```
function plotSpectrum(ths, Psig, theta, L, Pemu, index, pMax, range)

    figure;
    plot(theta,Pemu,'k-');

    hold on; quiver(ths,range(ones(1,L)),zeros(1,L),-10*ones(1,L),0); hold off;
    grid on; axis([1 360 0 range]);
    h = get(gcf,'CurrentAxes');
    set(h, 'FontWeight','Bold','FontSize',16,'FontName','Times');

    set(gca,'XTick',0:60:360)
    set(gca,'XTickLabel',{'360','60','120','180','240','300'})

%     legend('with spatial smoothing');
    xlabel('Azimuth [deg]'); ylabel('MUSIC Spectrum [dB]');
```

2. 信号の個数判定

信号の個数のパラメータは，アレーアンテナにおいて高精度なDOA推定手法などでは重要なパラメータとして扱われる．DOA推定法の多くは観測信号の相関行列の固有値分解を行い，信号の個数に対応する固有値，固有ベクトル（もしくは雑音空間に対応する固有値，固有ベクトル）を利用する．このとき信号の個数情報が必要となるが，一般的に未知である場合が多い．そこで，観測信号の相関行列から信号の個数を判定する方法が多くの研究されている．これは信号相関行列または信号部分空間のランクを決定する問題に帰着でき，AICやMDLなどが報告されている．

2.1. 個数判定法

最も簡単な方法の一つとして，固有値を調べる方法がある．受信信号の相関行列の固有値は$\lambda_1 \geq \cdots \geq \lambda_K > \lambda_{K+1} = \cdots = \lambda_M = \sigma^2$となる．ここで，Kを信号の個数，アレー素子数をM，データ数をNとする．これより固有値の大小を比較することにより，信号の個数Pを推定することは可能となる．しかし，SNRが低いときや推定に用いる受信信号のデータ数が少ないときには，きれいな分布にはならず判定が難しくなる．

そこで信号の個数を推定する手法として，赤池氏により提案されたAIC規範とSchwartzらによるMDL規範をアレー信号の個数判定に応用した手法などが提案されている．λ_iは行列の固有値とすると，AICとMDLによる判定は以下の式で表され，これを最小にするKを信号の個数とする．

$$AIC(K) = -2N(M-K)\ln\delta(K) + 2K(2M-K)$$

$$\delta(K) = \left(\prod_{i=K+1}^{M}\hat{\lambda}_i\right)^{\frac{1}{M-K}} \Bigg/ \left(\frac{1}{M-K}\sum_{i=K+1}^{M}\hat{\lambda}_i\right) \qquad for \qquad AIC$$

$$MDL(K) = -2N(M-K)\ln\delta(K) + K(2M-K)\ln N$$

$$\delta(K) = \left(\prod_{i=K+1}^{M}\hat{\lambda}_i\right)^{\frac{1}{M-K}} \Bigg/ \left(\frac{1}{M-K}\sum_{i=K+1}^{M}\hat{\lambda}_i\right) \qquad for \qquad MDL$$

2.2. 一般アレーアンテナにおける個数判定

受信アンテナは8素子リニアアレーを用いる．信号は30,80,300度方向から3波が到来するものとして，3波とも無相関波，2波が相関波で1波が無相関波の二通りについて検討した．またSNRは10(dB)と-10(dB)の二通りについて検討した．ここで，スナップショット数は1024とした．以下に表を用いて個数判定結果を示す．1段目横軸における数字は推定される到来波数である．また，2段目，3段目横軸における数字はそれぞれAIC，MDL判定法を用い，50回の試行における信号個数判定の結果である．

表2はSNR10(dB)で3波無相関の個数判定の結果である．この表は50回試行のうち，AICでは到来波数3個とした推定が43回で，到来波数4個とした推定が7回であることを

意味している．つまり到来波数4個とした7回の判定は不正確な推定であることになる．

この表より，AIC規範では実際の個数よりも多めに個数判定することがあるが，MDL規範では正確な個数判定が可能であることがわかる．

表3は同条件でF/B SSPを適用した場合の結果である．3波とも無相関である場合にF/B SSPを適用しても個数判定に影響を与えないことがわかり，AIC規範においては精度が良好になる事がわかる．これはF/B-SSPによる合成により信号電力が大きくなる一方で，雑音成分は平均化処理され軽減されるため，信号部分固有値と雑音成分固有値の見分けがつきやすくなったためと考えられる．

表2. SNR 10(dB)，3波無相関，w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	43	7	0	0	0
MDL	0	0	50	0	0	0	0

表3. SNR 10 (dB)，3波無相関，with F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	50	0	0	0	0
MDL	0	0	50	0	0	0	0

表4はSNR-10(dB)で3波無相関の個数判定の結果である．この表より，AIC規範では先程同様に実際の個数よりも多めに個数判定をすることがわかる．反対に，MDL規範では実際の個数よりも少なめに判定することがわかる．

表5は同条件でF/B SSPを適用した場合の結果である．F/B SSP適用後においても，AIC規範はほぼ正確な個数を示しているが，MDL規範では間違った個数判定をしていることがわかる．このことから，MDL規範は低いSNRでF/B SSPを適用すると個数判定が困難になる事がわかる．

表4. SNR -10(dB)，3波無相関，w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	45	4	1	0	0
MDL	0	7	43	0	0	0	0

表5. SNR -10 (dB)，3波無相関，with F/B SSP

	1	2	3	4	5	6	7
AIC	0	1	49	0	0	0	0
MDL	0	50	0	0	0	0	0

表6はSNR10(dB)で相関波2波と無相関1波の個数判定の結果である．表より，AIC規範でもMDL規範でも相関波が存在すると間違った個数判定を行なう確率が高いことがわかる．ここでは2波の相関波を信号個数1としているため，この相関波1と無相関波1で信号個数が2となる確率が高くなっている．

表7は同条件でF/B SSPを適用した場合の結果である．ここではAIC規範でもMDL規

範でも正確な個数判定ができており，F/B SSP 後の個数判定が有効である事がわかる．

表 8，表 9 はそれぞれ SNR10(dB)における相関係数 0.5 および 0.98 である場合の相関波 2 波と無相関 1 波の個数判定の結果である．表 6 のように相関係数が 1 である場合は個数判定が困難であるが，表 9 に示すように相関係数が 0.98 と高くても正確な個数判定ができることがわかる．

表 6. SNR 10(dB)，3 波(2 波相関)，w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	43	7	7	0	0	0
MDL	0	50	0	0	0	0	0

表 7. SNR 10 (dB)，3 波(2 波相関)，with F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	50	0	0	0	0
MDL	0	0	50	0	0	0	0

表 8. SNR 10 (dB)，3 波(2 波相関，相関係数 0.5)，w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	48	2	0	0	0
MDL	0	0	50	0	0	0	0

表 9. SNR 10 (dB)，3 波(2 波相関，相関係数 0.98)，w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	0	46	3	1	0	0
MDL	0	0	50	0	0	0	0

表 10 は SNR-10(dB)で相関波 2 波と無相関 1 波の個数判定の結果である．この表より，先程同様に AIC 規範でも MDL 規範でも相関波が存在すると個数判定が困難になる事がわかる．ここでも 2 波相関波を 1 波と判定しているため，信号個数 2 の確率が高くなっている．

表 11 は同条件で F/B SSP を適用した場合の結果である．ここで AIC 規範では，ある程度正確な個数判定をしており F/B SSP 後の個数判定が有効である事がわかる．しかし，MDL 規範では表 5 と同様に低い SNR-10(dB)であるため間違った個数判定になっている事がわかる．

これらより，完全相関波（相関係数 1）に対して SSP を行なわない表 6 と表 10 は推定が正確でないので別として，MDL 規範は低い SNR よりも高い SNR で有効であることがわかる．また，反対に AIC 規範は高い SNR よりも低い SNR で有効であることがわかる．

表 10. SNR -10(dB), 3 波(2 波相関), w/o F/B SSP

	1	2	3	4	5	6	7
AIC	0	45	4	1	0	0	0
MDL	0	50	0	0	0	0	0

表 11. SNR -10 (dB), 3 波(2 波相関), with F/B SSP

	1	2	3	4	5	6	7
AIC	0	10	40	0	0	0	0
MDL	0	50	0	0	0	0	0

2.3 エスパアンテナにおける個数判定

受信アンテナは7素子円形配列エスパアンテナを用い，ビームパターンを変化させ時系列の信号系列を取得するリアクタンスドメイン信号処理を用い，これらの時系列信号により個数判定を行なう．ここでは 50 回の試行における信号個数判定の結果を示す．信号は30,80,度方向から 2 波が到来するものとして，無相関波，相関波の二通りについて検討した．またSNR は 10(dB)について検討した．ここで，スナップショット数は 1024 とした．

表 12. は SNR 10(dB)で無相関波 2 波の個数判定の結果である．この表より，一般的なアレーアンテナの場合と同様に AIC 規範では個数を多めに判定する傾向にあり，MDL 規範では個数を少なめに判定する傾向にあることがわかる．また，エスパアンテナのリアクタンスドメイン信号処理で取得した信号を用いても AIC 規範や MDL 規範を用いた個数判定が可能であることがわかる．

表 13. は同条件で菱形サブアレーによる F/B SSP を適用後における個数判定の結果である．ここで，AIC 規範でも MDL 規範でも個数判定が正確でないことがわかる．

表 12. SNR 10 (dB), 2 波無相関 , w/o diamond F/B SSP

	1	2	3	4	5
AIC	0	48	2	0	0
MDL	5	45	0	0	0

表 13. SNR 10 (dB), 2 波無相関 , with diamond F/B SSP

	1	2	3
AIC	0	3	47
MDL	0	5	45

表 14. は SNR 10(dB)で相関波 2 波の個数判定の結果である．この表より，一般的なアレーアンテナの場合と同様に相関波が存在する場合に SSP 適用なしに行なう個数判定は困難であることがわかる．

表 15. は同条件で菱形サブアレーによる F/B SSP を適用後における個数判定の結果である．この場合においても不正確な個数判定になっていることがわかる．参考までに表 16. に一般的な 6 素子円アレーの中心に 1 素子のアンテナを付加した 7 素子アレーの個数判定

結果を示す．ここでは，菱形サブアレーを用いて個数判定が可能になっている．したがって，菱形アレーによるSSPを用いることに問題があるのではないことがわかる．また表12よりエスパアンテナでリアクタンスドメイン信号処理を用いることに問題があるのではないこともわかる．よって，エスパアンテナで菱形サブアレーによるSSPを適用する際に用いる素子間結合を除去するエレメントスペース変換に問題があり，この変換によって固有値の分布が崩れることが原因の1つであると考えられる．

表14. SNR 10 (dB), 2波相関 , w/o diamond F/B SSP

	1	2	3	4	5
AIC	48	2	0	0	0
MDL	50	0	0	0	0

表15. SNR 10 (dB), 2波相関 , with diamond F/B SSP

	1	2	3
AIC	0	1	49
MDL	0	5	45

表16. SNR 10 (dB), 2波相関 , with diamond F/B SSP

AIC	0	50	0
MDL	0	50	0

```
% 複数回の試行による個数判定規範の評価

clear
clc

Pd = 2000;          % Pd : 送信されたシンボル数
Fd = 1;             % Fd : シンボルレート; 1Hz に正規化
Fs = 4*Fd ;         % Fs : シンボル辺りの標本数
R = 0.5 ;           % R : roll-off factor
Delay = 5 ;         % Delay : フィルタの遅延時間 [symbol]
No = 1 ;            % No : 雑音の電力密度スペクトル
M = 4 ;             % M : 多値数

% 信号系列発生
x1 = randint(Pd,1,M) ;
x2 = randint(Pd,1,M) ;
x3 = randint(Pd,1,M) ;

% 信号配置変調のためのマッピング
y1 = modmap(x1,Fd,Fs,'qask',M) ;
y2 = modmap(x2,Fd,Fs,'qask',M) ;
y3 = modmap(x3,Fd,Fs,'qask',M) ;

% フィルタリング(raised cosine filter)
[rev_a1,ti] = rcosflt(y1,Fd,Fs,'fir/sqrt/Fs',R,Delay) ;
[rev_a2,ti] = rcosflt(y2,Fd,Fs,'fir/sqrt/Fs',R,Delay) ;
[rev_a3,ti] = rcosflt(y3,Fd,Fs,'fir/sqrt/Fs',R,Delay) ;

% 変調波信号の発生複素振幅変調
s1 = amodce(rev_a1,10,'qam') ;
s2 = amodce(rev_a2,10,'qam') ;
s3 = amodce(rev_a3,10,'qam') ;
s4 = s2*sqrt(0.98^2) + s3*sqrt(1-0.98^2) ;

% アレーアンテナ受信信号の計算
i=sqrt(-1) ; j=i ;  % i,j : 複素数
m = 8 ;  % m : アレー素子数
p = 3 ;  % p : 到来波数
pset = 4 ; % 1 : 無相関３波, 2 : 相関２波と無相関１波, 3 : 相関３波
           % 4 : 相関２波(相関係数 0.5) と 無相関１波,
angle = [ 30; -60; 80] ; % 到来角の設定
th = angle(1:p) ;   % 到来角ベクトル
nn = 1024 ;         % nn : データ数
SN = [10; 10; 10] ; % SNR の定義
sn = SN(1:p) ;      % SN ベクトル
deg2rad = pi/180 ;  % ラジアンへの変換係数

% 信号源ベクトルの作成
tt = 1:nn ;
switch pset
case 1
   SS = [s1(tt).';s2(tt).';s3(tt).'] ;
   S = SS(1:p,:) ;
case 2
   SS = [s1(tt).';s1(tt).';s2(tt).'] ;
   S = SS(1:p,:) ;
case 3
   SS = [s1(tt).';s1(tt).';s1(tt).'] ;
   S = SS(1:p,:) ;
case 4
   SS = [s1(tt).';s2(tt).';s4(tt).'] ;
   S = SS(1:p,:) ;
end

% SNの計算
Ps = S*S'/nn ;
ps = diag(Ps) ;
refp = 2*10.^(sn/10) ;
tmp = sqrt(refp./ps) ;
S2 = diag(tmp)*S ;

% ステアリング行列の作成
tmp = -i*pi*sin(th'*deg2rad) ;
```

```
tmp2 = [0:m-1]' ;
a2 = tmp2*tmp ;
A = exp(a2) ;

Nk = 50 ;
ans_aic_1 = zeros(1,m-1) ; ans_mdl_1 = zeros(1,m-1) ;
ans_aic_2 = zeros(1,m-2) ; ans_mdl_2 = zeros(1,m-2) ;
ans_aic_3 = zeros(1,m-2) ; ans_mdl_3 = zeros(1,m-2) ;

for kkk = 1:Nk
    % 観測雑音の生成
    nr = randn(m,nn) ;
    ni = randn(m,nn) ;
    U = nr + j*ni ;

    % アレーアンテナ受信信号の計算
    X = A*S2 +U ;

    % 相関行列と逆行列の計算
    Rxx1 = X*X'/nn ;                    % w/o SSP
    Rxx2 = ( Rxx1(1:m-1,1:m-1) + Rxx1(2:m,2:m) )/2 ;  % SSP
    J = fliplr(eye(m-1)) ;
    Rxx3 = (Rxx2 + J*Rxx2.'*J)/2 ;      %F/B SSP

    % 固有値の計算
    [U_1,S_1,V_1] = svd(Rxx1) ;  lambda_1 = diag(S_1) ;
    [U_2,S_2,V_2] = svd(Rxx2) ;  lambda_2 = diag(S_2) ;
    [U_3,S_3,V_3] = svd(Rxx3) ;  lambda_3 = diag(S_3) ;

    % 個数判定規範
    for k = 1:m-1
        number_a1(k) = aic(k,m,nn,lambda_1) ; % AIC criterion
        number_m1(k) = mdl(k,m,nn,lambda_1) ; % MDL criterion
    end
    for k = 1:m-2
        number_a2(k) = aic(k,m-1,nn,lambda_2) ; % AIC criterion
        number_m2(k) = mdl(k,m-1,nn,lambda_2) ; % MDL criterion

        number_a3(k) = aic(k,m-1,nn,lambda_3) ; % AIC criterion
        number_m3(k) = mdl(k,m-1,nn,lambda_3) ; % MDL criterion
    end

    % 複数回の試行による評価
    [min_a1 ,num_a1] = min(number_a1) ;
    [min_m1 ,num_m1] = min(number_m1) ;
    ans_aic_1(num_a1) = ans_aic_1(num_a1) + 1 ;
    ans_mdl_1(num_m1) = ans_mdl_1(num_m1) + 1 ;

    [min_a2 ,num_a2] = min(number_a2) ;
    [min_m2 ,num_m2] = min(number_m2) ;
    ans_aic_2(num_a2) = ans_aic_2(num_a2) + 1 ;
    ans_mdl_2(num_m2) = ans_mdl_2(num_m2) + 1 ;

    [min_a3 ,num_a3] = min(number_a3) ;
    [min_m3 ,num_m3] = min(number_m3) ;
    ans_aic_3(num_a3) = ans_aic_3(num_a3) + 1 ;
    ans_mdl_3(num_m3) = ans_mdl_3(num_m3) + 1 ;
end

% 出力結果を表示
disp('Criterion for estimating the number of signals, w/o SSP')
No_of_signals = [1:m-1]
ans_aic_1
ans_mdl_1

disp('Criterion for estimating the number of signals, with SSP')
No_of_signals = [1:m-2]
ans_aic_2
ans_mdl_2

disp('Criterion for estimating the number of signals, with F/B SSP')
No_of_signals = [1:m-2]
ans_aic_3
```

ans_mdl_3

```
% 複数回の試行による個数判定規範の評価 (ESPAR Antenna)

clc;
clear all;

deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;d=lambda/4;
n = 1024;                       % Pd : 送信されたシンボル数
M = 6 ;                         % M : numbers of sensors
L = 6;

Pn = -20 ;
sigma2 = 10^(Pn/10);            % Pn : SNR[dB]
psk = [1/4 3/4 5/4 7/4]*pi;  % in case of BPSK

% create response vector
theta = [1:1.0:360];        %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M;

V = exp(j*2*pi*d/lambda*(cos(theta(ones(M,1),:)*deg-phi(ones(1,length(theta)),:).')));     %%%
atheta(1,1:360) = 1 ;
for in = 1:M
    atheta(in+1,:) = V(in,:);
end


z0 = 50;  vvs = 100;
y00 =  0.00023344-j*0.0066918;     %補正
y10 = -0.0001816 +j*0.00242469;
y11 =  0.00300676-j*0.0044176;
y21 =  0.00097427+j*0.00299986;
y31 = -0.0003066 -j*0.0003067;
y41 = -0.0000973 -j*0.0001353;

% アドミタンス行列
Y = [y00 y10 y10 y10 y10 y10 y10;
     y10 y11 y21 y31 y41 y31 y21;
     y10 y21 y11 y21 y31 y41 y31;
     y10 y31 y21 y11 y21 y31 y41;
     y10 y41 y31 y21 y11 y21 y31;
     y10 y31 y41 y31 y21 y11 y21;
     y10 y21 y31 y41 y31 y21 y11];

Y0 = Y(:,1);
I = eye(M+1,M+1);


% リアクタンス行列
min=-j*90;max=0-j*4.77;
xno =[  0   0   0   0   0   0   0   ;
        max min max max max max max  ;
        max max min max max max max  ;
        max max max min max max max  ;
        max max max max min max max  ;
        max max max max max min max  ;
        max max max max max max min ];
clear max ; clear min ;


% 等価ウエイトベクトル
iin = [];
for jloop = 1:M+1
    X = diag(xno(jloop,:));
    X(1,1) = z0*2;
    MX = I+Y*X;
    MXinv = inv(MX);
    ii = vvs*MXinv*Y0;
    iin = [iin; ii.'];
end

%
theta = [0:1:359];        %[deg]:the expected range of input signal
phi = [0:M-1]*2*pi/M; %
```

```
% ビームパターン
wei = iin(2:end,:);
Wei = wei ;
pat = wei*atheta;


%%% DOA Setting
ddeg = [30 300];
ths = [ddeg];
Psig = [0];
kw = [2];
D = length(ths);
As2 = 10.^(Psig/10);


%%%  Criterion for estimating the number of signals, w/o SSP , before CUBA
Nk = 50 ;
ans_aic_1 = zeros(1,M) ;
ans_mdl_1 = zeros(1,M) ;
for kkk = 1:Nk

    %%% 受信信号作成
    yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M+1, iin);

    %%% 相関行列作成
    Ryy = yt*yt'/n;
    [Uy, Dy, vy] = svd(Ryy);

    [U,S,V] = svd(Ryy) ;
    lambda_1 = diag(S) ;
    for k =1:M
        number_a1(k) = aic(k,M+1,n,lambda_1) ;
        number_m1(k) = mdl(k,M+1,n,lambda_1) ;
    end
    [min_a1 ,num_a1] = min(number_a1) ;
    [min_m1 ,num_m1] = min(number_m1) ;
    ans_aic_1(num_a1) = ans_aic_1(num_a1) + 1 ;
    ans_mdl_1(num_m1) = ans_mdl_1(num_m1) + 1 ;
end
disp('Criterion for estimating the number of signals, w/o SSP ,before CUBA')
No_of_signals = [1:M]
ans_aic_1
ans_mdl_1


%%%  Criterion for estimating the number of signals, with SSP , before CUBA
Nk = 50 ;
K = 4;
ans_aic_5 = zeros(1,M-1) ; ans_mdl_5 = zeros(1,M-1) ;
ans_aic_11 = zeros(1,K-1) ; ans_mdl_11 = zeros(1,K-1) ;
ans_aic_12 = zeros(1,K-1) ; ans_mdl_12 = zeros(1,K-1) ;
ans_aic_13 = zeros(1,K-1) ; ans_mdl_13 = zeros(1,K-1) ;

for kkk = 1:Nk
    %%% 受信信号作成
    yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M+1, iin);

    %%% 相関行列作成
    Ryy = yt*yt'/n ;%+ 0.0001*eye(7,7);
    [Uy, Dy, vy] = svd(Ryy);

    noiseEigenValue = diag(Dy);
    %eleSsigma2 = 0.0001;
    eleSsigma2 = sum(noiseEigenValue(D+1:M+1))/(M+1-D);
    xyt = inv(iin)*yt;

    Rxx = xyt*xyt'/n-eleSsigma2*inv(iin)*inv(iin)';
    %Rxx = xyt*xyt'/n + eleSsigma2*eye(7,7);%-eleSsigma2*inv(iin)*inv(iin)';%

    %[Ux, Dx, vx] = svd(Rxx);

%%%%%%%%%%%
    [U_5,S_5,V_5] = svd(Rxx) ; lambda_5 = diag(S_5) ;
```

```
    for k =1:K-1
        number_a5(k) = aic(k,K,n,lambda_5) ;
        number_m5(k) = mdl(k,K,n,lambda_5) ;
    end

    [min_a5 ,num_a5] = min(number_a5) ;
    [min_m5 ,num_m5] = min(number_m5) ;
    ans_aic_5(num_a5) = ans_aic_5(num_a5) + 1 ;
    ans_mdl_5(num_m5) = ans_mdl_5(num_m5) + 1 ;
end

disp('Criterion for estimating the number of signals, w/o diamond SSP ,before CUBA')
No_of_signals = [1:M-1]
ans_aic_5
ans_mdl_5

%%%%%%%%%%%%

ans_aic_11 = zeros(1,K-1) ; ans_mdl_11 = zeros(1,K-1) ;
ans_aic_12 = zeros(1,K-1) ; ans_mdl_12 = zeros(1,K-1) ;
ans_aic_13 = zeros(1,K-1) ; ans_mdl_13 = zeros(1,K-1) ;
for kkk = 1:Nk

%     for k =1:K-1
%         number_a_s(k) = aic(k,K,n,lambda_1) ;
%         number_m_s(k) = mdl(k,K,n,lambda_1) ;
%     end

    order1 = [2 3 7 1 4 6 5] ;
    order2 = [3 4 2 1 5 7 6] ;
    order3 = [4 5 3 1 6 2 7] ;

    Rx1 = [Rxx(order1,:)];
    Rx2 = [Rx1(:,order1)];
    Rss1 = espCoherentssp(Rx2,K);

    Rx1 = [Rxx(order2,:)];
    Rx2 = [Rx1(:,order2)];
    Rss2 = espCoherentssp(Rx2,K);

    Rx1 = [Rxx(order3,:)];
    Rx2 = [Rx1(:,order3)];
    Rss3 = espCoherentssp(Rx2,K);

    [U_1,S_1,V_1] = svd(Rss1) ; lambda_11 = diag(S_1) ;
    [U_2,S_2,V_2] = svd(Rss2) ; lambda_12 = diag(S_2) ;
    [U_3,S_3,V_3] = svd(Rss3) ; lambda_13 = diag(S_3) ;

    for k =1:K-1
        number_a11(k) = aic(k,K,n,lambda_11) ;
        number_m11(k) = mdl(k,K,n,lambda_11) ;

        number_a12(k) = aic(k,K,n,lambda_12) ;
        number_m12(k) = mdl(k,K,n,lambda_12) ;

        number_a13(k) = aic(k,K,n,lambda_13) ;
        number_m13(k) = mdl(k,K,n,lambda_13) ;

    end

    [min_a11 ,num_a11] = min(number_a11) ;
    [min_m11 ,num_m11] = min(number_m11) ;
    ans_aic_11(num_a11) = ans_aic_11(num_a11) + 1 ;
    ans_mdl_11(num_m11) = ans_mdl_11(num_m11) + 1 ;

    [min_a12 ,num_a12] = min(number_a12) ;
    [min_m12 ,num_m12] = min(number_m12) ;
    ans_aic_12(num_a12) = ans_aic_12(num_a12) + 1 ;
    ans_mdl_12(num_m12) = ans_mdl_12(num_m12) + 1 ;

    [min_a13 ,num_a13] = min(number_a13) ;
    [min_m13 ,num_m13] = min(number_m13) ;
```

```
    ans_aic_13(num_a13) = ans_aic_13(num_a13) + 1 ;
    ans_mdl_13(num_m13) = ans_mdl_13(num_m13) + 1 ;

end


disp('Criterion for estimating the number of signals, with diamond SSP ,before CUBA')
No_of_signals = [1:K-1]
ans_aic_11
ans_mdl_11
ans_aic_12
ans_mdl_12
ans_aic_13
ans_mdl_13


% %%% Criterion for estimating the number of signals, w/o SSP , after CUBA
% Nk = 50 ;
% ans_aic_2 = zeros(1,M-1) ;
% ans_mdl_2 = zeros(1,M-1) ;
% for kkk = 1:Nk
%     %%% 受信信号作成
%     yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M+1, iin);
%     yt2 = yt(2:end,:);
%
%     %%% CUBA
%     b=[pat(1,360)];
%     for kk=1:M-1
%         b = [b pat(1,360*kk/M)];
%     end
%
%     ml = [0:M-1]'*[-2:3];
%     ex = exp(-j*2*pi*ml/M);
%
%     B = (b*ex).';
%     Cp = (yt2.'*ex).';
%     for k = 1:n
%         xt(:,k) = Cp(:,k)./B;
%     end
%
%     Rxx = xt*xt'/n;
%     [U,S,V] = svd(Rxx) ;
%     lambda_2 = diag(S) ;
%
%     for k =1:M-1
%         number_a2(k) = aic(k,M,n,lambda_2) ;
%         number_m2(k) = mdl(k,M,n,lambda_2) ;
%     end
%     [min_a2 ,num_a2] = min(number_a2) ;
%     [min_m2 ,num_m2] = min(number_m2) ;
%     ans_aic_2(num_a2) = ans_aic_2(num_a2) + 1 ;
%     ans_mdl_2(num_m2) = ans_mdl_2(num_m2) + 1 ;
% end
%
% disp('Criterion for estimating the number of signals, w/o SSP ,after CUBA')
% No_of_signals = [1:M-1]
% ans_aic_2
% ans_mdl_2
%
%
% %%% Criterion for estimating the number of signals, w/o SSP , after CUBA ,w/o low level data
% Nk = 50 ;
% ans_aic_22 = zeros(1,M-2) ;
% ans_mdl_22 = zeros(1,M-2) ;
% for kkk = 1:Nk
%     %%% 受信信号作成
%     yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M+1, iin);
%     yt2 = yt(2:end,:);
%
%     %%% CUBA
%     b=[pat(1,360)];
%     for kk=1:M-1
%         b = [b pat(1,360*kk/M)];
```

```
%     end
%
%     ml = [0:M-1]'*[-2:3];
%     ex = exp(-j*2*pi*ml/M);
%
%     B = (b*ex).';
%     Cp = (yt2.'*ex).';
%     for k = 1:n
%         xt(:,k) = Cp(:,k)./B;
%     end
%
%     Rxx = xt*xt'/n;
%     [U,S,V] = svd(Rxx(1:5,1:5)) ;
%     lambda_22 = diag(S) ;
%
%     for k =1:M-2
%         number_a22(k) = aic(k,M-1,n,lambda_22) ;
%         number_m22(k) = mdl(k,M-1,n,lambda_22) ;
%     end
%     [min_a22 ,num_a22] = min(number_a22) ;
%     [min_m22 ,num_m22] = min(number_m22) ;
%     ans_aic_22(num_a22) = ans_aic_22(num_a22) + 1 ;
%     ans_mdl_22(num_m22) = ans_mdl_22(num_m22) + 1 ;
% end
%
% disp('Criterion for estimating the number of signals, w/o SSP ,after CUBA, w/o low level data')
% No_of_signals = [1:M-2]
% ans_aic_22
% ans_mdl_22
%
%
% %%% Criterion for estimating the number of signals, with SSP , after CUBA ,w/o low level data
% Nk = 50 ;
% ans_aic_3 = zeros(1,K) ;
% ans_mdl_3 = zeros(1,K) ;
% for kkk = 1:Nk
%     %%% 受信信号作成
%     yt = beamSrfsignal( ths, atheta, sigma2, kw, n, psk, M+1, iin);
%     yt2 = yt(2:end,:);
%
%     %%% CUBA
%     b=[pat(1,360)];
%     for kk=1:M-1
%         b = [b pat(1,360*kk/M)];
%     end
%
%     ml = [0:M-1]'*[-2:3];
%     ex = exp(-j*2*pi*ml/M);
%
%     B = (b*ex).';
%     Cp = (yt2.'*ex).';
%     for k = 1:n
%         xt(:,k) = Cp(:,k)./B;
%     end
%
%     Rxx = xt*xt'/n;
%     [Ux, Dx, vx] = svd(Rxx);
%
%     K = L-2;
%     order = [1:K+1];
%
%     Rx1 = [Rxx(order,:)];
%     Rx2 = [Rx1(:,order)];
%     Rss = fbeespCoherentssp(Rx2,K);
%
%     [U,S,V] = svd(Rss) ;
%     lambda_3 = diag(S) ;
%
%     for k =1:K-1
%         number_a3(k) = aic(k,K,n,lambda_3) ;
%         number_m3(k) = mdl(k,K,n,lambda_3) ;
%     end
%     [min_a3 ,num_a3] = min(number_a3) ;
```

```
%      [min_m3 ,num_m3] = min(number_m3) ;
%      ans_aic_3(num_a3) = ans_aic_3(num_a3) + 1 ;
%      ans_mdl_3(num_m3) = ans_mdl_3(num_m3) + 1 ;
% end
%
% No_of_signals = [1:M-1]
% ans_aic_3
% ans_mdl_3
%
```

```
function out = mdl(k,m,n,lambda)

% MDL criterion for estimating the number of signals
% k : number of signals for evaluation
% m : number of sensors
% n : number of samples
% lambda : eigen value vector

num1 = (sum(lambda(k+1:m))/(m-k))^(m-k) ;
den1 = prod(lambda(k+1:m)) ;
out = n*log(num1/den1) + k*(2*m-k)*log(n)/2 ;
```

```
function out = aic(k,m,n,lambda)

% AIC criterion for estimating the number of signals
% k : number of signals for evaluation
% m : number of sensors
% n : number of samples
% lambda : eigen value vector

num1 = (sum(lambda(k+1:m))/(m-k))^(m-k) ;
den1 = prod(lambda(k+1:m)) ;
out = n*log(num1/den1) + k*(2*m-k) ;
```

```
function [yt] = beamSrfsignal(doa, atheta, sigma2, kw, n, qpskiq, m, iin)

% DOA
ths = doa;
L = length(ths);
Psig = zeros(1,L);
As2 = 10.^(Psig/10);

Vs = [];
for iv = 1:L
    iths = ths(iv);
    Vs = [Vs atheta(:,iths)];
end

% reactance domain
an = sqrt(sigma2); as = sqrt(As2);
nwav = length(kw);                    %%%
isousa = zeros(1,L);

signal = [];
stphi = randint(nwav,n)*2-1;
kk = 0;
for in = 1:nwav
%     psi = rand(1,1);
    psi = 0;
    ks = kk+1; kk = kk+kw(in);
    for ik = ks:kk
        As_psi = as(ik)*exp(j*(psi+stphi(in,:)+isousa(ik)'));          %setting initial phase
        signal = [signal; As_psi];
    end
end
%signal = ones(1,n);

yt = iin*Vs*signal+(an/sqrt(2))*(randn(m,n)+j*randn(m,n));
%yt = iin*Vs*signal;
```

```
%%% coherentssp.m

function Ryy = fbeespCoherentssp(Rxx,K);

% spatial smoothing
[M,MM]=size(Rxx);
N = M-K+1 ;
J = fliplr(eye(M)) ;
fwRxx = (Rxx + J*Rxx.'*J)/2 ;
Ryy = zeros(K,K);

for in = 1:N
     Ryy = Ryy + fwRxx(in:in+K-1,in:in+K-1);  %部分相関
end

Ryy = Ryy / N;

% End coherentssp.m
```

```
%%% coherentssp.m

function Rxx = espCoherentssp(Ryy,K);
% spatial smoothing
[M,MM]=size(Ryy);
% N=M-K+1;
N = 2;

J = fliplr(eye(M)) ;
Ryy = (Ryy + J*Ryy.'*J)/2 ;

Rxx = zeros(K,K);
% for in = 1:N
%     Rxx = Rxx + Ryy(in:in+K-1,in:in+K-1);%部分相関
% end

Rxx = Rxx + Ryy(1:4,1:4);
Rxx = Rxx + Ryy(4:7,4:7);

Rxx = Rxx / N;

% End coherentssp.m
```

3. リアクタンスドメイン信号処理による時系列データを用いたMIMO方式

MIMO(Multi Input Multi Output)方式では，複数の送信アンテナから送信信号系列が同一時間に同一周波数で送信され，受信側ではそれらの信号が複数の受信アンテナを用いて受信され，信号分離アルゴリズムを用いることで送信信号系列が復元される．この方法による通信容量の増大は，大きく分けて2種類の実現法がある．一つはSDM (Space Division Multiplexing)を利用し同時に送信される信号を分離することで容量増大を実現する方式，もう一方は，STC (Space Time Coding)のようなダイバーシチ技術による信号電力の増大を利用し多値変調を用いて容量増大を実現する方式である．信号の分離法としては，MLDやV-BLAST などの方法があり，MLD は高精度な信号分離が可能であるが処理速度遅く，V-BLASTは信号分離の精度はMLDよりも劣るが処理速度が速いという特徴がある．本稿では，これらの方法のうちSDMでV-BLASTアルゴリズムを用いることとする．この際，容量を増大させるためには送受信アンテナ数を増やすことが必要であり，小型化を要求される端末においては複数のアンテナを実装することが困難であるという問題がある．そこで，受信アンテナを増やすことなく容量を増大させるためエスパアンテナを用いてリアクタンスドメイン信号処理を行い，一般的なアレーアンテナにおいて素子ごとに取得していた信号系列を時系列信号として取得することでSDMを擬似的に行なう．ここでは，伝送速度の遅いシステムを対象として，同一信号系列をリアクタンスドメイン信号処理により異なるビームパターンでオーバーサンプリングして受信した後にV-BLAST で信号分離を行なうものとする．

3.1. V - BLASTによる信号分離

V - BLASTは，誤り率特性が良好な送信信号系列を先に識別し，その結果を受信信号ベクトルから引き算して新たなウエイトを求める方法である．ウエイトベクトルはMMSEアルゴリズムで求めることも可能であるが，Zero-Forcing 法を用いるものとする．K を送信アンテナ数，N を受信アンテナ数とすると，Zero-Forcing アルゴリズムによるウエイトベクトルは以下の式で表される．

$$\mathbf{w}_k = (\mathbf{H}^T)^+ \mathbf{g}_k$$

但し， $\mathbf{g}_k$ はk番目の要素のみ1で他が0のK次元列ベクトルである．また，$\mathbf{H}$ は伝播路行列であり，$(\mathbf{H}^T)^+$ は $\mathbf{H}^T$ のMoore-Penrose一般逆行列である．これを全てのk(k＝1～K)について求め，

$$\mathbf{y}_k = \mathbf{w}_k^T \mathbf{x}(t)$$

より，送信信号系列 s_k の検出が可能となる．

これにより全ての送信信号 $s_1(t), s_2(t), \cdots, s_K(t)$ の検出が可能であるが，V-BLASTでは，より良好な特性を得るため次のような処理を行い，誤り率特性が良好な送信信号系列を先に識別している．送信信号のうち熱雑音の影響が最小のものは，ノルム最小であるウエイトベクトルであり，$\mathbf{w}_k$ は $(\mathbf{H}^T)^+$ のk番目の列ベクトルであるから，$(\mathbf{H}^T)^+$ の最もノルムの小さい列ベクトルをウエイトベクトルにすることになる．このような熱雑音電力の影響を

最も受けない最適ウエイトベクトルによって送信信号$s_a(t)$を推定する．

次に信号ベクトル$\mathbf{x}(t)$からs_aを含む項を除き，$\{\mathbf{x}(t)-s_a(t)\mathbf{h}_a\}$を求める．そして，先ほどと同様の処理により信号を分離する．この場合，K-1 個の送信信号しか含まれていないので自由度に 1 個多くの余裕ができ，ダイバーシチに使われることになる．これを繰り返す方法が V-BLAST である．

これらを整理して式で表すと，まず初めに(1 ステップ目)，

$$\mathbf{x}^{(1)}(t)=\mathbf{x}(t), \qquad \mathbf{H}^{(1)}=\mathbf{H}$$

とおく．$(\mathbf{H}^{(1)T})^{+}$の各列ベクトルのノルムを計算し，その最小のものを第 1 ステップ目のウエイトベクトルとする．$\mathbf{y}^{(1)}(t)=\mathbf{w}^{(1)T}\mathbf{x}^{(1)}(t)$により，送信信号の検出を行なう．これが第 1 ステップ目での識別結果で$s^{(1)}(t)$と表す．$s^{(1)}(t)$に対応するアレー応答ベクトルを$\mathbf{h}^{(1)}$とする．$s^{(1)}(t)$を送信しているアンテナを j 番目のアンテナであったとすると，

$$\mathbf{h}^{(1)}=[h_{1j},h_{2j},\cdots h_{Nj}]$$

となる．ここから，2 ステップとなる．

$$\mathbf{x}^{(2)}(t)=\mathbf{x}^{(1)}(t)-s^{(1)}(t)\mathbf{h}^{(1)},$$

$\mathbf{H}^{(1)}$から$\mathbf{h}^{(1)}$の列要素を除いた N 行(K-1)列のチャネル行列を$\mathbf{H}^{(2)}$とする．ここで，1 ステップと同様の処理を繰り返すことで全ての送信信号が取得できる．

3.2. 計算機シミュレーションによる結果

エスパアンテナによるリアクタンスドメイン信号処理を用いてビームパターンを回転させることで時系列の信号系列を取得し，V-BLAST による信号分離アルゴリズムを適用したものを ESP-V-BLAST とする．

表 17 にシミュレーション諸元を示す．送信アンテナでは QPSK 変調を用い，それぞれのアンテナ毎に異なる信号を送信するものとした．つまり，信号多重数と送信アンテナ数は等しいものとした．伝播路は，位相の異なる同一電力の到来波が 36 度間隔で 10 方向から到来する静的レイリーフェージング伝播路とし，それぞれの受信アンテナで無相関であるとした．また，伝播路は既知とし，信号分離アルゴリズムとしては Zero-Forcing を用いた．SNR は−10 (dB) ～ 30 (dB)を変化させるものとし，1 アンテナから送信した信号系列をそれぞれの受信アンテナにおいてオムニパターンで受信した電力平均に対する雑音電力の比として表す．ここで，エスパアンテナにおける SNR はオムニパターンを基準としており，指向性ビーム方向によって雑音の影響が大きいパターンと小さいパターンが生じる．

まず受信アンテナをエスパアンテナ，2 素子アレーアンテナとした場合のそれぞれの SN-BER 特性の比較を行なう．図 9 に信号多重数 2，図 10 に信号多重数 4，図 11 に信号多重数 6，における SN-BER 特性を示す．エスパアンテナにおいて，指向性をもつビームパターンでの SNR とオムニパターンでの SNR は異なり，さらに指向性の方向によっても SNR が異なるので，アレーアンテナの場合と完全には同条件ではないが，図 9，図 10，図

11 のグラフのように，ほぼ似た特性になることがわかる．また，信号多重数 6 においてもビームパターンを 6 方向に変化させることで信号の分離が可能であることがわかる．このことより，エスパアンテナのリアクタンスドメイン信号処理を用いた V-BLAST の適用が可能であることがわかる．

次に，信号多重数を 4，エスパアンテナにおけるビームパターン数を 4 として，ビームパターンの組み合わせを変化させた場合の SNR-BER 特性を示す．ここでは，ビームパターンを『オムニ，0，120，240 度』，『オムニ，0，60，120 度』，『0，60，120，180 度』，『0，60，180，300 度』の 4 種類とした場合について比較した．この結果より，オムニパターンを含まない特性がオムニパターンを含む特性よりも良好であることがわかる．この原因の 1 つとしてオムニパターンでの受信電力が指向性を持ったパターンでの受信電力と比較して低いことがある．

最後に，図 13 に 7 方向ビームパターン(オムニ,0,60,120,180,240,300 度)を用いた場合に信号多重数を 2,4,6,7 とした SNR-BER 特性，図 14 に 6 方向ビームパターン(オムニ,0,60,120,180,240 度)を用いた場合に信号多重数を 2,4,5,6 とした SN-BER 特性，図 15 に 6 方向ビームパターン(60,120,180,240,300 度)を用いた場合に信号多重数を 2,4,5,6 とした SNR-BER 特性を示す．図 13 より，信号多重数 7 においては特性が良くないことがわかる．また，信号多重数 2,4,6 においては図 15 の BER 特性とほぼ同様であることがわかる．ここでは信号多重数に対してビームパターンが多いにも関わらずダイバーシチ利得が得られていない．この原因の 1 つとしてオムニビームパターンの特性が良好でないことが考えられる．また，図 14，図 15 の結果より，ここでもオムニパターンを含まない場合の BER 特性が良好であることがわかる．

表 17. シミュレーション諸元

変調方式	QPSK
受信アンテナ	7 素子エスパアンテナ (ビーム：オムニ二,0,60,120,180,240,300 度) アレーアンテナ (素子間隔: $\lambda/2$)
伝播環境	無相関静的レイリーフェージング
信号分離アルゴリズム	Zero-Forcing
伝播路特性	既知
SNR	−10 (dB) ～ 30 (dB)

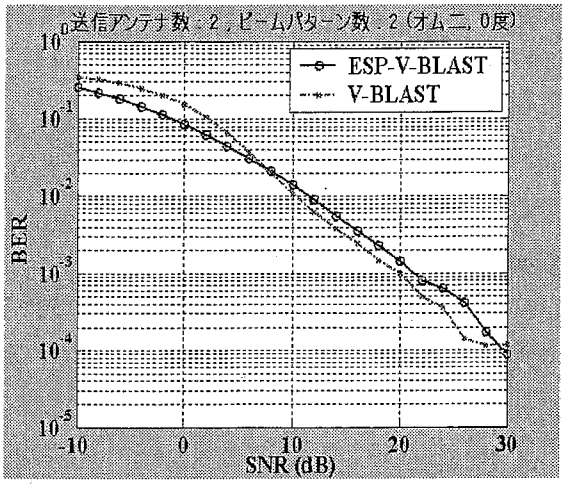


図 9. 送信信号 2，ビーム数 2 における特性

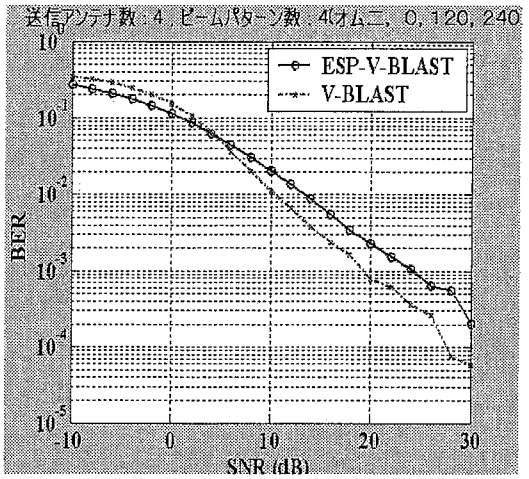


図 10 送信信号 4，ビーム数 4 における特性

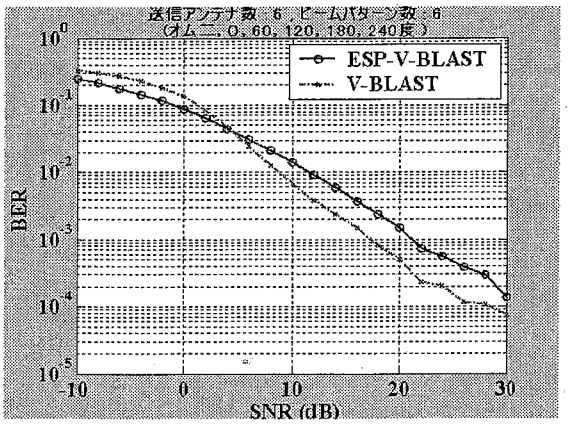


図 11.送信信号 6，ビーム数 6 における特性

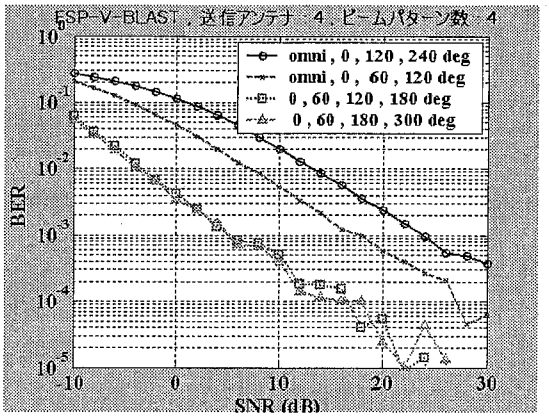


図 12.ビームパターンによる特性変化

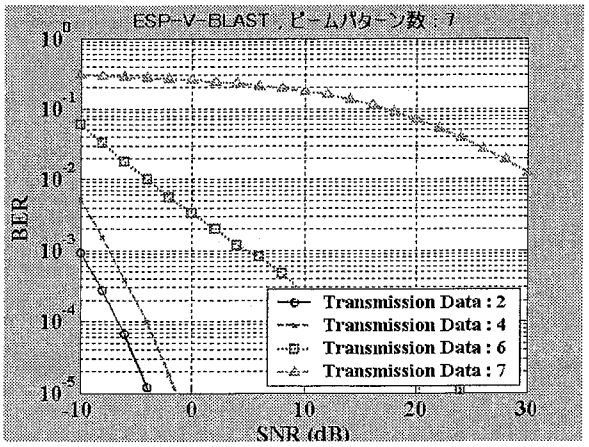


図 13. 送信信号数による特性変化
（7 ビームパターン）

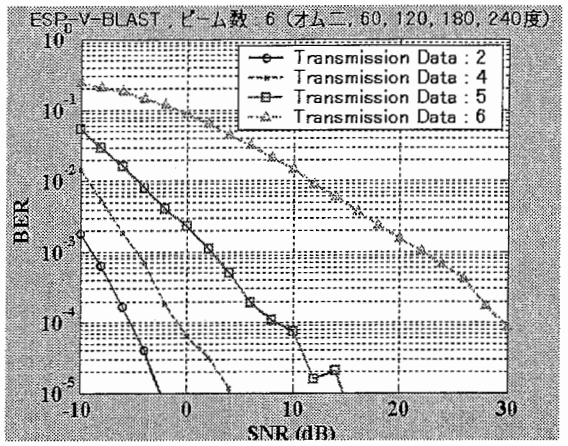


図 14. 送信信号数による特性変化
(6 ビームパターン,オムニを含むパターン)

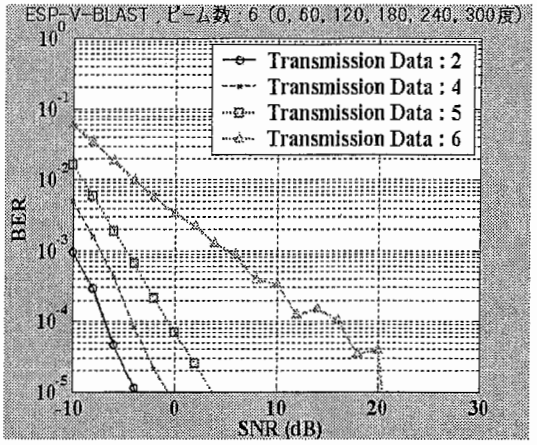


図 15. 送信信号数による特性変化
(6 ビームパターン,オムニを含まないパターン)

また, 参考までに図 16 に ESP-V-BLAST を用いた電波暗室内での実験結果を示す. ここで, 信号多重数は 2, SNR を 20(dB), データ数 800, 変調方式 BPSK を用いたビームパターン数対 BER 特性である. 伝播路推定は 100 シンボルを用いて行なった. ビームパターンに関しては全てオムニパターンを含み, それ以外のパターンは 0,60,120,180,240,300 度に指向性をもつパターンから選んだ. 結果としてビーム数が 3 では送信信号分離が正確ではないが, ビーム数が 4 以上である場合は正確に分離できている事がわかる.

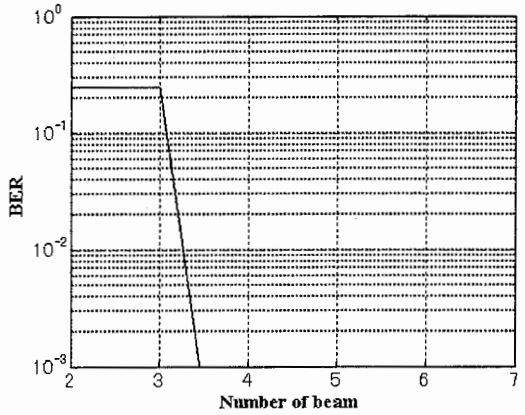


図 16. ESP-V-BLAST 実験結果

[1] 宮下 和巳, 西村 寿彦, 大鐘 武雄, 小川 恭孝, 鷹取 泰司, 長 敬三, "MIMO チャネルにおける固有ビーム空間分割多重(E-SDM)方式", 信学技報, RCS2002-53 (2002-05) , pp13-pp18

[2] 西尾 敬介, 小川 恭孝, 西村 寿彦, 大鐘 武雄, "MIMO-OFDM 空間分割多重に関する基礎的検討", 信学技報, DSP2002-204, SAT2002-154, RCS2002-271 (2003-01) , pp121-pp126

```
%%%% V-BLAST %%%%

clear
clc

%%------パラメーター設定---------%%
%データ関連%
n = 100 ;
MM = 2 ;
M = 2^MM;                              %多値数
pilot_num = 16 ;

%伝送路関連%
% u = 0 ;                              %移動速度(m/s)%
f_c = 2.484e9 ;                        %使用周波数%
f_d = 5e6 ;                            %伝送速度(bps)%
c = 3e8 ;                              %光速
ramda = c/f_c ;                        %波長
% f_m = u/ramda ;                      %最大ドップラーシフト(現実)%
% frate = f_m/f_d ;                    %フェージングの速度%
% fm = frate * 1 ;                     %最大ドップラー周波数(相対)%
N = 10 ;                               %レイリーの波数%

space = 4 ;                            %多重数
tr_ant = 4 ;                           %送受信アンテナ数
re_ant = 4 ;
dd = ramda/2;                          %アンテナ間隔
d = 0:dd:(re_ant-1)*dd ;               %素子位置

%その他%
noiselevel = 0:2:30 ;                      %SNR%
SNR = sqrt(10.^(-noiselevel/10));
loop = 20;                             %ループ回数%
deg = pi/180;
t = 1 : n ;

%%% ステアリングベクトル %%%
% theta = 1:360 ;
% atheta = [exp(-j*2*pi*d.'*sin(theta*pi/180))] ;


%%--------シュミレーション開始---------%%%%%%%%
t0 = clock;
fprintf('プログラム開始時刻     %d月%d日%d時%d分¥n',[t0(2:5)]);

ERR = zeros(1,length(SNR));
for k3 = 1 : length(SNR)
  num = zeros(1,loop) ;
  for k1 = 1 : loop
    %%%% 送信データ %%%%
    H1 = [1 1 ;1 -1];    H2 = [H1 H1 ; H1 -H1];
    H3 = [H2 H2 ; H2 -H2];  H4 = [H3 H3 ; H3 -H3];
    H5 = [H4 H4 ; H4 -H4];
    data = randint(n-pilot_num,space,M) ;          % data
    mod_data = dmodce(data,1,1,'qam',M) ;          % 変調
    pilot = (H5(1:space,1:pilot_num).'==1)*2-1;    % パイロットシンボル
    tr_data = [pilot;mod_data] ;

    %%%% S/N 定義用 %%%%
    S_power = mean(mean(mod_data.*conj(mod_data))) ;


    %%%% 受信データ %%%%
    resignal = zeros(n,re_ant) ;
    for k2 = 1 : tr_ant
        randph = randint(re_ant,N,360)*pi/180 ;
        z1(:,k2) = sum(exp(j*randph),2)/sqrt(N) ;          %準静的レイリー
        resignal = resignal + tr_data(:,k2) * z1(:,k2).' ;
    end

    %%% 雑音付加 %%%
    for kk = 1 : re_ant
        Recsignal(:,kk) = resignal(:,kk) + sqrt(S_power)*(SNR(k3)/sqrt(2))*(randn(n,1)+j*randn(n,1)) ;
```

```
    end

    %%%% 伝播路推定 %%%%
    Rss = (pilot.'*conj(pilot)) / pilot_num ;
    Rys = (Recsignal(1:pilot_num,:).'*conj(pilot)) / pilot_num ;
    %Hss = Rys*inv(Rss);                          % 伝播路推定
    Hss = z1;                                    % 伝播路既知

    %%%% V-BLAST %%%%
    index_num = [];
    ind = [1:space];
    for kk = 1 : space
       pinvHss = pinv(Hss.');
       [value index] = min(sum(abs(pinvHss),1));
       g = zeros(space-kk+1,1);
       g(index) = 1;
       index_num = [index_num ind(index)] ;
       weight = pinvHss * g ;
       [va,i] = find(ind(index)~= ind) ;
       ind = ind(i);

       ydata(kk,:) = weight.' * Recsignal.' ;
       Re_de_data(kk,:) = ddemodce(ydata(kk,:),1,1,'qam',M) ;
       s_data = dmodce(Re_de_data(kk,:),1,1,'qam',M) ;
       Recsignal = Recsignal - s_data.' * Hss(:,index).' ;
       [non_value,non_index] = find(index~=[1:space-kk+1]);
       Hss = Hss(:,non_index) ;
    end

    %%%%%%%%

    bi_Re_de_data = de2bi(Re_de_data(:,pilot_num+1:end) );
    bi_data = de2bi(data(:,index_num).');
    num(1,k1) = biterr( bi_data , bi_Re_de_data) ;

  end

  ERR(1,k3) = sum(num)/(n-pilot_num)/space/loop/MM + 10^(-7);
end
t1=clock ;

fprintf('¥nプログラム終了時間  %2d日%2d時%2d分 ¥n',[t1(3:5)])

    figure;
    semilogy(noiselevel,ERR)
    xlabel('SNR (dB)','FontWeight','Bold','FontSize',16,'FontName','Times');
    ylabel('BER ','FontWeight','Bold','FontSize',16,'FontName','Times');
    grid on; axis([ 0 30  10^(-5) 1]);
    h = get(gcf,'CurrentAxes');
    set(h, 'FontWeight','Bold','FontSize',16,'FontName','Times');
```

```
%%%% ESP-V-BLAST %%%%
clear
clc

%%------パラメーター設定---------%%
%データ関連%
n = 300 ;
MM = 2 ;
M = 2^MM;                              %多値数
pilot_num = 16 ;                       %パイロットシンボル数

%伝送路関連%
% u = 0 ;                               %移動速度(m/s)%
% f_c = 2.484e9 ;                       %使用周波数%
% f_d = 5e6 ;                           %伝送速度(bps)%
% c = 3e8 ;                             %光速
% ramda = c/f_c ;                       %波長
% f_m = u/ramda ;                       %最大ドップラーシフト(現実)%
% frate = f_m/f_d ;                     %フェージングの速度%
% fm = frate * 1 ;                      %最大ドップラー周波数(相対)%
N = 10 ;                               %レイリーの波数%

space = 4 ;
tr_ant = 4 ;                           %送受信アンテナ数
re_ant = 1 ;
%psk = [0 1]*pi;                       %%% in case of BPSK

%その他%
noiselevel = 0:2:30 ;                  %SNR%
SNR = 10.^(-noiselevel/10);
loop = 10;                             %ループ回数%
deg = pi/180;
t = 1 : n ;

deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;d=lambda/4;
m = 6 ;                                %numbers of sensors

% create response vector
theta = [1:1.0:360];        %[deg]:the expected range of input signal
phi = [0:m-1]*2*pi/m;

%% steering vector
V = exp(j*2*pi*d/lambda*(cos(theta(ones(m,1),:)*deg-phi(ones(1,length(theta)),:).')));    %%%
atheta(1,1:360) = 1 ;
for in = 1:m
    atheta(in+1,:) = V(in,:);
end

z0 = 50;  vvs = 100;


% reactance
y00 =  0.00023344-j*0.0066918;     %補正
y10 = -0.0001816 +j*0.00242469;
y11 =  0.00300676-j*0.0044176;
y21 =  0.00097427+j*0.00299986;
y31 = -0.0003066 -j*0.0003067;
y41 = -0.0000973 -j*0.0001353;

% reactance matrix
Y = [y00 y10 y10 y10 y10 y10 y10;
     y10 y11 y21 y31 y41 y31 y21;
     y10 y21 y11 y21 y31 y41 y31;
     y10 y31 y21 y11 y21 y31 y41;
     y10 y41 y31 y21 y11 y21 y31;
     y10 y31 y41 y31 y21 y11 y21;
     y10 y21 y31 y41 y31 y21 y11];

Y0 = Y(:,1);
I = eye(m+1,m+1);
Z = inv(Y);
Zs = z0*2*eye(m+1,m+1);
```

```
min=-j*90;max=0;
xno =[  0   0   0   0   0   0   0   ;
       max min max max max max max   ;
       max max min max max max max   ;
       max max max min max max max   ;
       max max max max min max max   ;
       max max max max max min max   ;
       max max max max max max min ];
clear max

%%% weight
iin = [];
for jloop = 1:m+1
    X = diag(xno(jloop,:));
    X(1,1) = z0*2;
    MX = I+Y*X;
    MXinv = inv(MX);
    ii = vvs*MXinv*Y0;
    iin = [iin; ii.'];
end

wei = iin([1 2:2:end],:);   % select of beam pattern
m = 3 ;                     % beam pattern - 1


%%-------シュミレーション開始---------%%%%%%%%
t0 = clock;
fprintf('プログラム開始時刻     %d月%d日%d時%d分¥n',[t0(2:5)]);

%ERR = zeros(1,length(SNR));
for k3 = 1 : length(SNR)
  num = zeros(1,loop) ;
  for k1 = 1 : loop
    %%%% 送信データ %%%%
    H1 = [1 1 ;1 -1];    H2 = [H1 H1 ; H1 -H1];
    H3 = [H2 H2 ; H2 -H2];  H4 = [H3 H3 ; H3 -H3];
    H5 = [H4 H4 ; H4 -H4];
    pilot = [H5(1:space,1:pilot_num).'] ;
    data = randint(space,n-pilot_num,M);
    mod_data = dmodce(data,1,1,'qam',M);
    tr_data = [pilot(:,1:space).' mod_data];

    %%%% S/N 定義用 %%%%
    S_power = mean(mean(mod_data.*conj(mod_data))) ;


    %%%% 受信データ %%%%
    for k2 = 1 : tr_ant
        angle =  randint(1,N,360) + 1 ;
        Vs = [ atheta(:,angle) ];

        beam_power = abs( iin * Vs ).^2 ;
        omni_power = sum( beam_power(1,:) ) ;

        z1(:,k2) = wei * Vs * exp(j*angle.' * pi/180)/sqrt(omni_power) ;
        yt_1(:,:,k2) = z1(:,k2) * tr_data(k2,:) ;
    end

    yt_2 = sum(yt_1,3);
    yt = yt_2 + (sqrt(S_power)*sqrt(SNR(k3))/sqrt(2))*(randn(m+1,n)+j*randn(m+1,n));

    Recsignal = yt.';


    %%%% 伝播路推定 %%%%
    Rss = (pilot.'*conj(pilot)) / pilot_num ;
    Rys = (Recsignal(1:pilot_num,:).'*conj(pilot)) / pilot_num ;
    %Hss = Rys*inv(Rss);                          % 伝播路推定
    Hss = z1;                                    % 伝播路既知

    %%%% V-BLAST %%%%
    index_num = [];
    ind = [1:space];
```

```
    for kk = 1 : space
       pinvHss = pinv(Hss.');
       [value index] = min(sum(abs(pinvHss),1));
       g = zeros(space-kk+1,1);
       g(index) = 1;
       index_num = [index_num ind(index)] ;
       weight = pinvHss * g ;
       [va,i] = find(ind(index)~= ind) ;
       ind = ind(i);

       ydata(kk,:) = weight.' * Recsignal.' ;
       Re_de_data(kk,:) = ddemodce(ydata(kk,:),1,1,'qam',M) ;
       s_data = dmodce(Re_de_data(kk,:),1,1,'qam',M) ;
       Recsignal = Recsignal - s_data.' * Hss(:,index).' ;
       [non_value,non_index] = find(index~=[1:space-kk+1]);
       Hss = Hss(:,non_index) ;
    end

    %%%%%%%%

    bi_Re_de_data = de2bi(Re_de_data(:,pilot_num+1:end) );
    bi_data = de2bi(data(index_num,:));
    num(1,k1) = biterr( bi_data , bi_Re_de_data) ;

  end

  ERR(1,k3) = sum(num)/(n-pilot_num)/space/loop/MM + 10^(-7);
end

t1=clock ;
%save test5 ERR noiselevel

fprintf('¥nプログラム終了時間  %2d日%2d時%2d分 ¥n',[t1(3:5)])


    figure;
    semilogy(noiselevel,ERR)
    xlabel('SNR (dB)','FontWeight','Bold','FontSize',16,'FontName','Times');
    ylabel('BER ','FontWeight','Bold','FontSize',16,'FontName','Times');
    grid on; axis([ 0 30  10^(-5) 1]);
    h = get(gcf,'CurrentAxes');
    set(h, 'FontWeight','Bold','FontSize',16,'FontName','Times');
```

4. 位相モード変換によるコヒーレント波到来方向推定

4.1. 位相モード変換の定式化

一般的にモードベクトルが下記のような Vandermonde 構造：

$$\mathbf{a}=\begin{bmatrix}1\\ z^{-1}\\ \vdots\\ z^{-(N-1)}\end{bmatrix}$$

になれば，リニアアレーで用いられる SSP などの適用が可能である．しかしながら，円アレーにおけるモードベクトルは以下で表され，Vandermonde 構造ではない．

$$\mathbf{a}(\theta)=\mathbf{a}(\theta,\phi)=\begin{bmatrix}\exp(j\zeta\cos(\phi-\gamma_0))\\ \exp(j\zeta\cos(\phi-\gamma_1))\\ \vdots\\ \exp(j\zeta\cos(\phi-\gamma_{N-1}))\end{bmatrix}$$

ここで，N 素子円アレー各々の素子における ϕ 方向(アジマス方向)の位置角度が $\gamma_i\,(i=0,1,\cdots N-1)$であり，n 番目素子の位置角度は x 軸から $\gamma_i=2\pi n/N$ 離れていることとなる．また，ζ は θ 方向(エレベーション方向)を用い，$\zeta=k_0 r\sin\theta$，$(k_0=2\pi/\lambda)$で表される．ここで，r はアレー中心とした原点としたアレー半径を表す．

そこで，位相モード変換を行ない Vandermonde 構造になるように変換する．ここで，位相モード変換に関する変換式は文献[1]に詳しく書かれているので，ここでは省略して簡単に示す．まず，(2h+1)×N の空間フーリエ変換行列を以下で表す．

$$\mathbf{F}=\begin{bmatrix}1 & w^{-h} & w^{-2h} & \cdots & w^{-(N-1)h}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & w^{-1} & w^{-2} & \cdots & w^{-(N-1)}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & 1 & 1 & \cdots & 1\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & w^{h} & w^{2h} & \cdots & w^{(N-1)h}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 1 & w^{1} & w^{2} & \cdots & w^{(N-1)}\end{bmatrix}$$

$$w=\exp(j2\pi/N)$$

次に，モードベクトルの振幅項を補償するための行列 J を以下の式で定義する．

$$\mathbf{J}=diag\left\{\frac{1}{\sqrt{N}\,j^m J_m(\zeta)}\right\}\qquad m=-h,\cdots,0,\cdots h$$

ここで，$J_m(\zeta)$ はm次のベッセル関数である．ここで定義された F，J を用いると

$$\tilde{\mathbf{a}}(\theta)=\mathbf{JFa}(\theta)=[\exp(-jh\phi),\cdots,\exp(-j\phi),1,\exp(j\phi),\cdots,\exp(jh\phi)]$$

となり，モードベクトルが位相モード空間において Vandemonde 構造となる．

すなわち，入射波を d 波として $N\times d$ モードベクトル行列を $\mathbf{A}(\theta)$，複素振幅信号ベクト

ルを s(t), 雑音ベクトルを n(t)とすると時間 t における各素子の出力をそれぞれのベクトル要素とした N次元ベクトルを $\mathbf{x}(t)$ は

$$\mathbf{x}(t)=\mathbf{A}(\theta)s(t)+\mathbf{n}(t)$$

で表され，下記のようにエレメント空間データを変換すればよいことがわかる．

$$\tilde{\mathbf{x}}(t)=\mathbf{JFx}(t)$$

4.2. エスパアンテナを用いた位相モード変換

位相モード変換では，r= λ, h＝6 において N＞12 が必要であるが残余項の誤差の影響を少なくするには N＞15 程度が良いという数値結果がある[2]. しかしながら，一般的なアレーアンテナでは素子数が増加するにつれて回路規模が増大する．そこで RF ポートが一系統のみであり素子数が増加しても回路規模が増大しないエスパアンテナを用いて位相モード変換を適用する．この際，エスパアンテナにおける素子間結合が問題となるので，素子間結合を取り除くエレメントスペース変換を行うことが必要となる．

以下に 13 素子エスパアンテナを用いた位相モード変換を式により示す．まず，エスパアンテナを用いて 0 度から 30 度おきに指向性を回転させ 330 度までの信号系列 $y(t)=[y_1(t), y_2(t),\cdots y_{12}(t)]$ を取得する．これらの信号系列から相関行列 $\mathbf{R}_{yy}=E[y(t)y^H(t)]$ が求められる．この際，相関行列には素子間結合が含まれている．そこで，素子間結合を打ち消すエレメントスペース変換を行う．

$$\mathbf{R}_{xx}=\mathbf{W}^{-1}(\mathbf{R}_{yy}-\hat{\sigma}^2)(\mathbf{W}^{-1})^H$$

ここで，$\mathbf{W}$ は等価ウエイト行列であり，$\hat{\sigma}$ は雑音成分の固有値を平均した雑音電力の推定値であり以下の式で表される．

$$\hat{\sigma}^2=\frac{1}{12-L}\sum_{i=L+1}^{12}\lambda_i$$

L は到来波数，λ_i は固有値を表す．ここで，エレメントスペース変換で取得した相関行列を用いて位相モード変換を行なう．

$$\tilde{\mathbf{R}}_{xx}=(\mathbf{JF})\mathbf{R}_{xx}(\mathbf{JF})^H$$

ここで取得した相関行列を用いて SSP を適用することで，コヒーレント波の到来方向推定を可能となる．

4.3. 計算機シミュレーションによる結果

円アレーは円周上に等間隔で並んだ 12 素子円アレーを，エスパアンテナは円周上に 12 素子と中心に 1 素子である 13 素子エスパアンテナを用いた．上記の式における h は 3 とし，変換後のデータ数が 7 になるようにした．また，SSP は F/B-SSP を用い，SNR は 20(dB) とした．

図 17 に一般的な円アレーで，図 18 にエスパアンテナで位相モード変換を行なった到来方向推定結果を示す．矢印は到来方向である 130 度，250 度を示し，到来波はコヒーレン

ト波とした．図 17，図 18 より矢印付近にピークが向いており，推定が可能であることがわかる．ここでエスパアンテナと円アレーにおける場合を比較するとエスパアンテナにおけるピークが小さいことがわかる．この原因の１つとしては，円アレーの場合よりも１つ多くの変換，つまりエレメントスペース変換を用いるため，その影響で特性が劣化していることが考えられる．

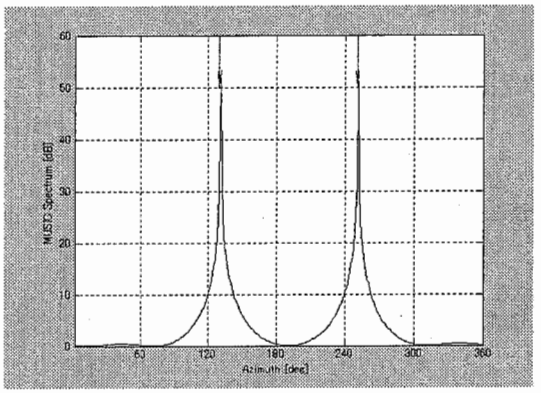


図 17.一般アレーにおける位相モード変換

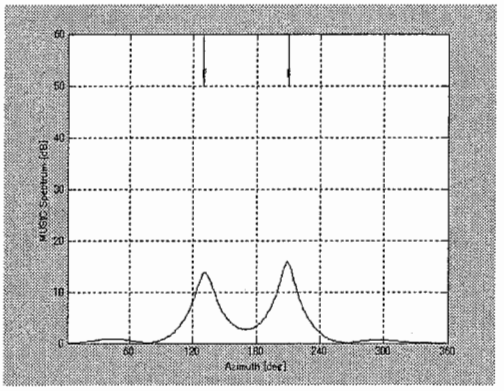


図 18.エスパアンテナにおける位相モード変換

参考文献

[1] “円アレーによる高分解能 DOA 推定に関するメモ”

[2] Cherian P. Mathews, Michael D. Zoltowski, “Eigenstructure Techniques for 2-D Angle Estimation with Uniform Circular Arrays”,IEEE Transactions on signal processing, Vol 42, No. 9, Sep.1994

```
% 円アレーで bessel 関数を用いた相関波到来方向推定
clc
clear all;
n = 1000;
deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;
m = 12;                                 %numbers of sensors
phi = [0:m-1]*2*pi/m;

ddeg = [130];                           % [Θ,Φ]・・・所望波到来方向[deg]
ideg = [250];                           % [Θ,Φ]・・・干渉波到来方向[deg]
DOA = [ddeg(1) ideg(1)];                % Θ信号到来方向[deg]
%ELE = [ddeg(2) ideg(2)];               % Φ信号到来方向[deg]
Psig = [0 0];                           % [dB]信号パワー
As2 = 10.^(Psig/10);                    % 信号パワーの真数
k = length(DOA);                        % 到来波数

Pn = -20; sigma2 = 10^(Pn/10);          % [dB]:noise level
kw = [2]; des = 1;
qpskiq = [pi*1.25 pi*0.75 pi*1.75 pi*0.25];

% create response vector
theta = [0:1:359];        %[deg]:the expected range of input signal
phi = [0:m-1]*2*pi/m;

theta = [1:360];          %[deg]:the expected range of input signal
V = exp(j*pi*(cos(theta(ones(m,1),:)*deg-phi(ones(1,length(theta)),:).')));    %%%
atheta = V;

we = exp(j*2*pi/m);
h = 3;
ho = [-h:h];
F = we.^(ho.'*[0:m-1])/sqrt(m);

zeta=pi*sin(pi/2);

jho = j.^ho;
jJzeta = jho.*besselj(ho,zeta);
J = diag(1./(sqrt(m)*jJzeta));

adash = J*F*atheta;

yt = beamSrfsignal( DOA, adash, sigma2, kw, n, qpskiq, 7, 7);
Ryy=yt*yt'/n;
[Uy, Dy,vy]=svd(Ryy);

%%% spatial smoothing by 3 directions
K = 6;
dino = []; Pssp = [];
order = [1 2 3 4 5 6 7];

ordElem1 = order(1,:);
[dino1, Pmusic] = spectrumSpatialSmoothing(ordElem1, Ryy, adash, K, k);

plotSpectrum(DOA, Psig, theta, k, Pmusic, 30, 300, 60)
```

```
% エスパアンテナで bessel 関数を用いた相関波到来方向推定
clc
clear all;
n = 1000;
deg = pi/180; c = 3e+8; f0 = 2484e6; lambda = c/f0;
m = 12;                                 %numbers of sensors
phi = [0:m-1]*2*pi/m;

ddeg = [130 ];                          % [Θ,Φ]・・・所望波到来方向[deg]
ideg = [210 ];                          % [Θ,Φ]・・・干渉波到来方向[deg]
DOA = [ddeg(1) ideg(1)];                % Θ信号到来方向[deg]
ELE = [];                               % Φ信号到来方向[deg]
Psig = [0 0];                           % [dB]信号パワー
As2 = 10.^(Psig/10);                    % 信号パワーの真数
k = length(DOA);                        % 到来波数

Pn = -20; sigma2 = 10^(Pn/10);                  % [dB]:noise level
kw = [2]; des = 1;
qpskiq = [pi*1.25 pi*0.75 pi*1.75 pi*0.25];

% create response vector
theta = [0:1:359];        %[deg]:the expected range of input signal
%eleva = [0:1:89];
phi = [0:m-1]*2*pi/m; %

atheta=[];

    for AZIMUTH = min(theta):1:max(theta)
        ATHETA(1,AZIMUTH+1)=1;
        ATHETA(2:m+1,AZIMUTH+1) = [ exp(j*pi*cos(AZIMUTH*deg-phi.'))];
    end
    atheta=[atheta;ATHETA];


 y00= 0.00067051910000-0.01008076000000*j;        %%% espar20031203.dat
 y10=-0.00029015030000+0.00152941430000*j;
 y11= 0.00117776600000-0.01710948100000*j;
 y21= 0.00080483640000+0.01077154000000*j;
 y31= 0.00008289476000-0.00123396900000*j;
 y41=-0.00017468471000-0.00019989180000*j;
 y51=-0.00009107783000-0.00006201371000*j;
 y61= 0.00000130228800-0.00006929386200*j;
 y71= 0.00002199928000-0.00009529652000*j;

z0 = 50; vvs = 100;
Y = [y00 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10 y10;
     y10 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21;
     y10 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31;
     y10 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51 y41;
     y10 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61 y51;
     y10 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71 y61;
     y10 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61 y71;
     y10 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51 y61;
     y10 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41 y51;
     y10 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31 y41;
     y10 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21 y31;
     y10 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11 y21;
     y10 y21 y31 y41 y51 y61 y71 y61 y51 y41 y31 y21 y11];


Y0 = Y(:,1);
I = eye(m+1,m+1);
% Z = inv(Y);
% Zs = z0*2*eye(m+1,m+1);
% ZC = Zs*inv(Z+Zs);
% x=j./ZC.*(Z.'*ZC);

% iZC = inv(ZC);
%min=j*52.632+47.283;
%min=47.283;
min=60;
xno = [ 0   0   0   0   0   0   0   0   0   0   0   0   0;
        0  min  0   0   0   0   0   0   0   0   0   0   0;
```

```
        0   0  min  0   0   0   0   0   0   0   0   0   0;
        0   0   0  min  0   0   0   0   0   0   0   0   0;
        0   0   0   0  min  0   0   0   0   0   0   0   0;
        0   0   0   0   0  min  0   0   0   0   0   0   0;
        0   0   0   0   0   0  min  0   0   0   0   0   0;
        0   0   0   0   0   0   0  min  0   0   0   0   0;
        0   0   0   0   0   0   0   0  min  0   0   0   0;
        0   0   0   0   0   0   0   0   0  min  0   0   0;
        0   0   0   0   0   0   0   0   0   0  min  0   0;
        0   0   0   0   0   0   0   0   0   0   0  min  0;
        0   0   0   0   0   0   0   0   0   0   0   0  min];

iin = [];
for jloop = 1:m+1
    X = diag(-j*xno(jloop,:));
    X(1,1) = z0*2;
    MX = I+Y*X;
    MXinv = inv(MX);
    ii = vvs*MXinv*Y0;
    iin = [iin; ii.'];
end
wei = iin(2:end,:) ;


we = exp(j*2*pi/m);
h = 3;
ho = [-h:h];
F = we.^(ho.'*[0:m-1])/sqrt(m);

jho = j.^ho;
jJzeta = jho.*besselj(ho,pi);
J = diag(1./(sqrt(m)*jJzeta));

ytt = beamSrfsignal( DOA, atheta, sigma2, kw, n, qpskiq, m+1, iin);
yt = ytt(2:end,:) ;
Rxx = yt * yt' / n ;
[Ux, Dx, vx] = svd(Rxx);


noiseEigenValue = diag(Dx);
eleSsigma2 = sum(noiseEigenValue(k+1:12))/(12-k);
xt = pinv(wei)*yt;
%xt = invW*yt;
xt = xt(2:end,:);

Rxx = (J*F)* (xt*xt'/n-eleSsigma2*pinv(wei)'*pinv(wei) ) *(J*F)' ;
%Rxx = xt*xt'/n-eleSsigma2*invW*invW';
[Ux, Dx, vx] = svd(Rxx);

%%% spatial smoothing by 3 directions
K = 6;
dino = []; Pssp = [];
order = [1 2 3 4 5 6 7];

adash = J*F*wei*atheta;
adash1 = J*F*atheta(2:end,:);

ordElem1 = order(1,:);
[dino1, Pmusic] = spectrumSpatialSmoothing(ordElem1, Rxx, adash1, K, k);

plotSpectrum(DOA, Psig, theta, k, Pmusic, 30, 300, 60)
```

```
function [yt] = beamSrfsignald2(DOA, atheta, sigma2, kw, n, qpskiq, m, iin)

% DOA
L = length(DOA);
Psig = zeros(1,L);
As2 = 10.^(Psig/10);

%%%%%%%%% ステアリングベクトルVs %%%%%%%%%%
Vs = [];
for iv = 1:L
    iths = DOA(iv);
    Vs = [Vs, atheta(:,iths+1)];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

an = sqrt(sigma2); as = sqrt(As2);
nwav = length(kw);                    %%%
% isousa = zeros(1,L);
isousa = rand(1,L);

signal = [];
 code = floor(rand(nwav,n)*4);                    %%% in case of QPSK
%code = floor(rand(nwav,n)*2);                    %%% in case of BPSK
stphi = qpskiq(code+1);
kk = 0;
for in = 1:nwav
    psi = rand(1,1);
    ks = kk+1; kk = kk+kw(in);
    for ik = ks:kk
        As_psi = as(ik)*exp(j*(psi+stphi(in,:)+isousa(ik)'));          %setting initial phase
        signal = [signal; As_psi];
    end
end

S_power = mean( mean( abs(iin*Vs*signal(:,:)).^2 ) );
 yt = iin*Vs*signal(:,:)/sqrt(S_power) +(an/sqrt(2))*(randn(m,n)+j*randn(m,n));
```

```
function [dino, Pmusic3] = spectrumSpatialSmoothing1(ord, Rxx, atheta, K, L)

  %  Rx1 = [Rxx(ord(1),:); Rxx(ord(2),:); Rxx(ord(3),:); Rxx(ord(4),:); Rxx(ord(5),:); Rxx(ord(6),:); R ↙
xx(ord(7),:)];
  %  Rx2 = [Rx1(:,ord(1))  Rx1(:,ord(2))  Rx1(:,ord(3))  Rx1(:,ord(4))  Rx1(:,ord(5))  Rx1(:,ord(6))  R ↙
x1(:,ord(7))];
    Rx2 = Rxx ;
    Rss = fbeespCoherentssp(Rx2,K);
    [Us,Ds,vs] = svd(Rss);

    vtheta = atheta(1:K,:);
    numv = real(diag(vtheta'*vtheta)).';
    VUx = vtheta'*Us(:,L+1:end);
    Pmusic3 = 10*log10(numv./sum(VUx.*conj(VUx),2).');
    [pMax,index] = max(Pmusic3);
    dino=0;                 %dummy
```

```
%%% coherentssp.m

function Ryy = fbeespCoherentssp(cr,K);
% spatial smoothing
[M,MM]=size(cr);
% N=M-K+1;
N = M-K+1 ;
J = fliplr(eye(M)) ;
crfb = (cr + J*cr.'*J)/2 ;
Ryy = zeros(K,K);

for in = 1:N
     Ryy = Ryy + crfb(in:in+K-1,in:in+K-1);%部分相関
end

Ryy = Ryy / N;

% End coherentssp.m
```

```
function plotSpectrum(ths, Psig, theta, L, Pemu, index, pMax, range)

    figure;
    plot(theta,Pemu,'k-');
    hold on; quiver(ths,range(ones(1,L)),zeros(1,L),-10*ones(1,L),0); hold off;
    grid on; axis([1 360 0 range]);
%    text(155,26,['DoA = ',num2str(index),' [deg.]    Peak = ',num2str(pMax), '[dB]']);
    set(gca,'XTick',0:60:360)
    set(gca,'XTickLabel',{'360','60','120','180','240','300'})
%     legend('with spatial smoothing');
    xlabel('Azimuth [deg]'); ylabel('MUSIC Spectrum [dB]');
```