Chapter Four

Simulation and Discuss

4.1 Introduction

In this chapter the recorded results are expressed as visualized using MATLAB functions and properties, generally there are two comparisons which between the constellations of 16QAM and rotated QPSK, in each one two antenna configurations are used which are 4x1 and 4x4 antenna configuration that can applies ML pair wise decoding. Finally these two schemes are compared with the proposed method which states that using the beamforming technique with the QOSTBC and also with the rotated QPSK constellation.

4.2 Simulation description:

The general idea is to make a simple system that generates random data or symbols, then modulate these symbols then coding with QOSTBC and add the beamforming weights, then pass this constructed data over a fading channel then add the AWGN at the receiver side, then detect, decode and demodulate these constructed data to recover the firstly transmitted data, and finally calculate the BER if there is error existence. This procedure is repeated over two loops. First for a number of iteration which is firstly set as $\mathbf{10^3}$ and $\mathbf{10^4}$ with SNR over 20 dB, this iteration can be assumed as the sampling frequency. Secondly for the various SNR values which are set as 0,5,10,…,35 dB.

Table (4.1): Simulation parameters

Parameters	Value
Sampling frequency (No. of iteration)	$\mathbf{10^3}$ for 0-20 dB, and $\mathbf{10^4}$ for higher SNR.
No. of Tx and Rx	4,1
SNR values	0,5,10,15,…,35 dB
Modulation	16QAM, rotated QPSK.

4.3 Channel capacity:

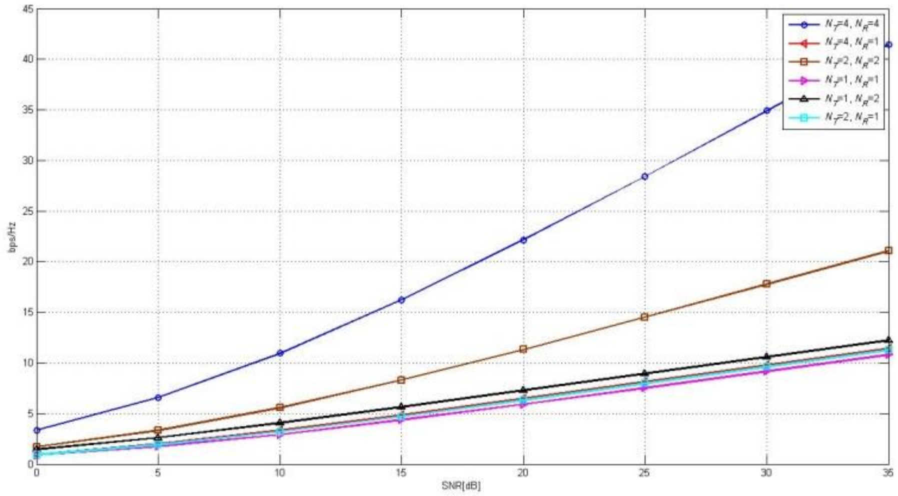

Figure (4.1): channel capacity for different antenna configuration

As mentioned before in chapter three the calculation of channel capacity in MIMO systems is a basic procedure for working with MIMO relatives, so the figure (4.1) below shows a comparison of ergodic channel capacity for different antenna configuration in terms of throughput versus SNR the results shows that the 4x4 and 2x2 has the best performance in term of throughput specially on higher SNR values, that means when the number of transmitters equals the number of receivers higher data rate is achievable, but may be with the cost of data reliability or BER.

4.4 QOSTBC with 16QAM constellation:

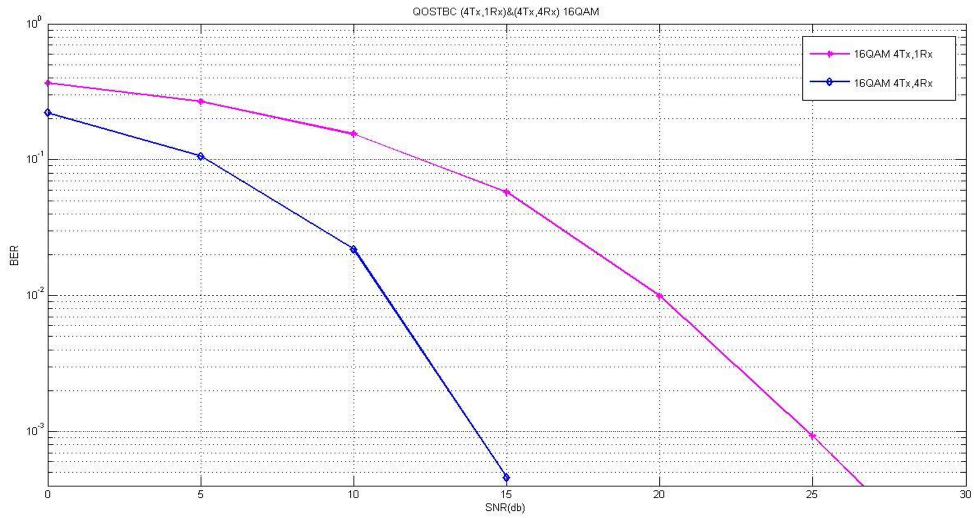

Figure (4.2): BER of QOSTBC with 16QAM

Figure (4.2) shows the BER of QOSTBC using 16QAM for the two different antenna configurations, the results shows that 4x4 configuration has lower and shorter BER curves which denotes a good performance compared with the 4x1 antenna configuration, but both has starting BER of higher that $\mathbf{10^{-1}}$ which is not acceptable in data transmission reliability. In 4x4 case the error eliminated with higher SNR exactly in 15 dB the error doesn't occurs below this SNR value, and in case of 4x1 the error still exist until the maximum value of SNR as set to 35 dB.

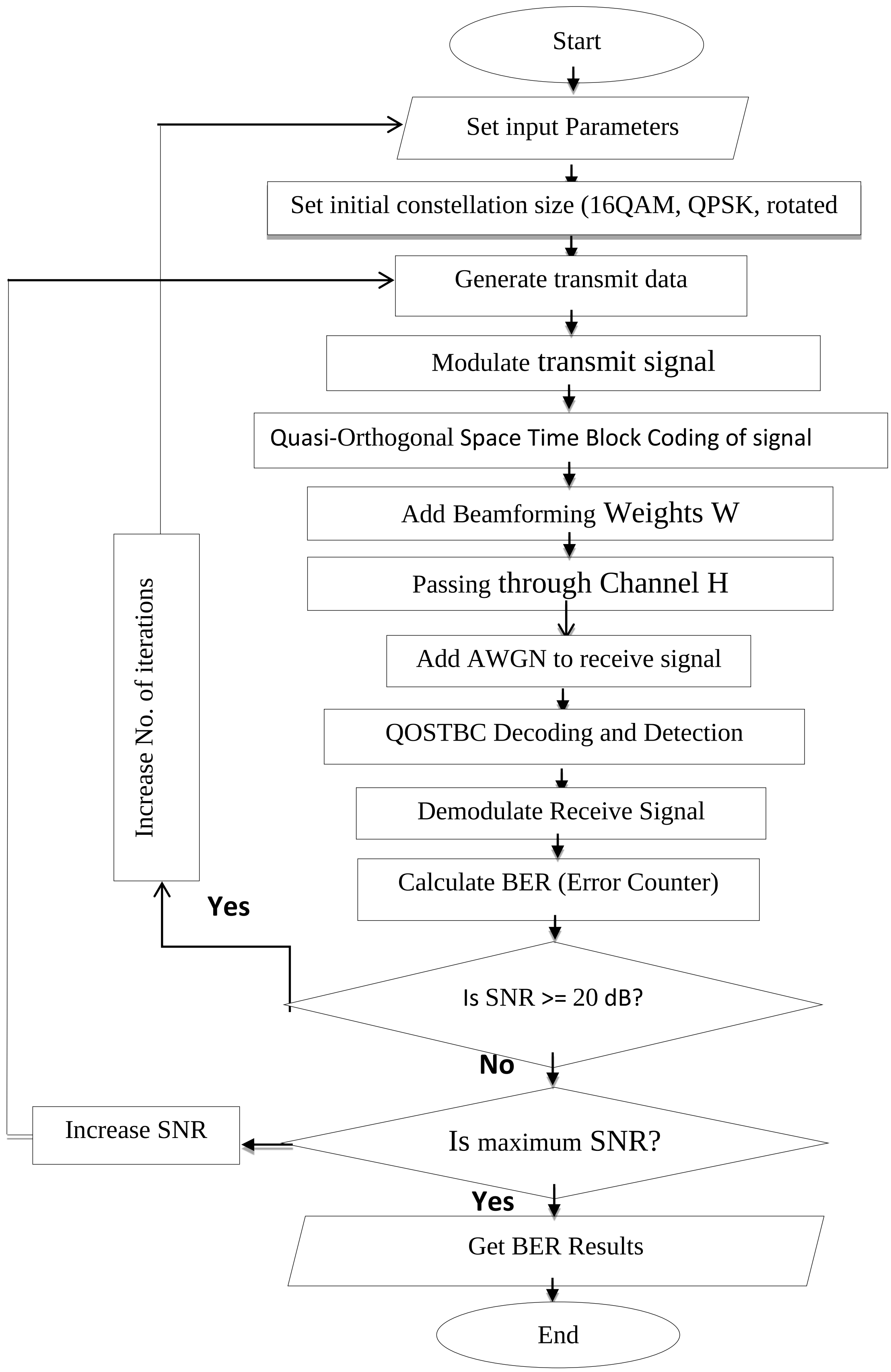

Figure (4.1): flowchart of QOSTBC

4.5 QOSTBC with rotated QPSK:

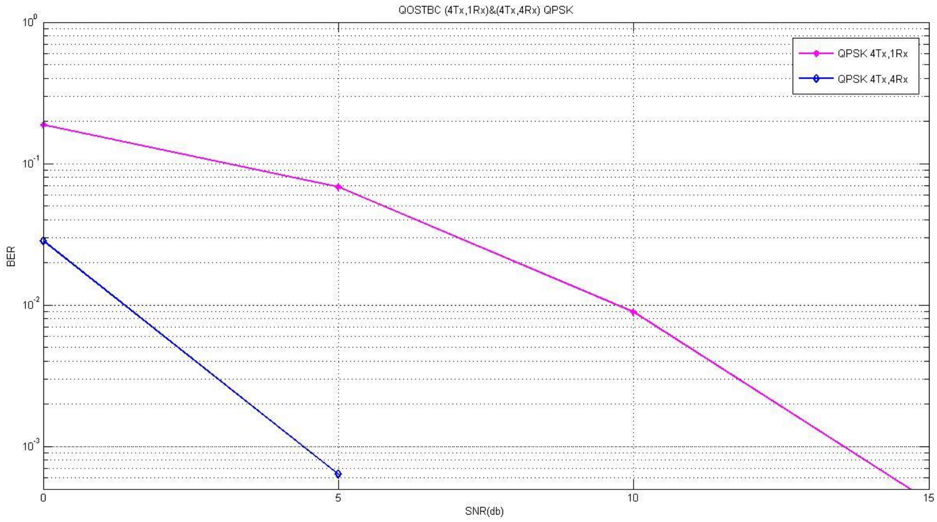

Figure (4.3): BER of QOSTBC with rotated QPSK

As mentioned before the rotation technique here is used to achieve full diversity so it should also introduce better performance in case of BER, in figure (4.3) below the BER of rotated QPSK of QOSTBC is illustrated and generally as before in case of 16QAM it’s obvious that the 4x4 configuration has the advantages over one receiver antenna configuration, but here in case of 4x1 the error is eliminated with 20 dB SNR value, and for 4x4 when the SNR is approximately 5 dB there is no more errors occurs, and this improve the performance of QOSTBC BER for rotated QPSK over the 16QAM as it can be clear in figure (4.4).

4.6 QOSTC With 16QAM vs Rotated QPSK:

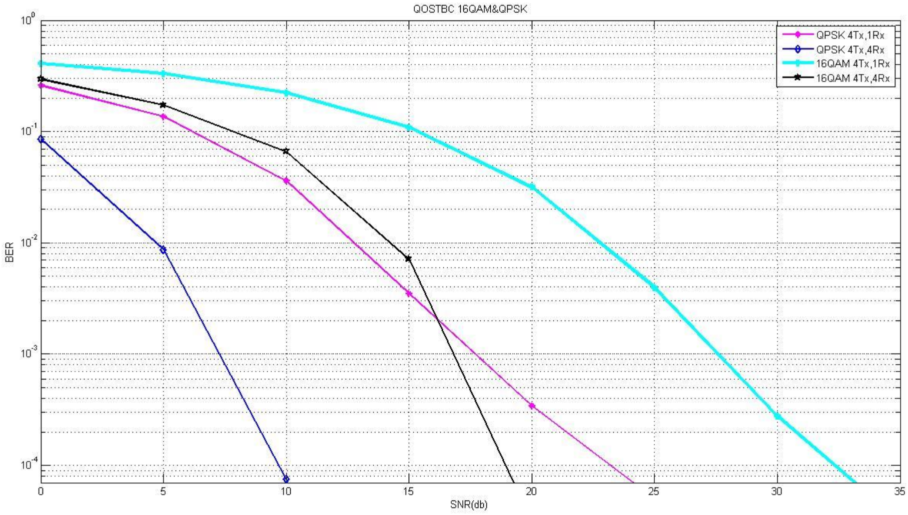

Figure (4.4): BER of QOSTBC for 16QAM vs rotated QPSK

Figure (4.4) shows the cross comparison between the QOSTBC with 16QAM and rotated QPSK constellation, and here we increase the number of iterations which meant by the sampling frequency from $\mathbf{10^3}$ $\boldsymbol{to}$ $\mathbf{10^4}$ and to $\mathbf{2} \times \mathbf{10^4}$ in higher SNR value to achieve a fair comparison, and it's clear that QOSTBC with rotated QPSK has the advantage over using 16QAM, and also it's introduce a reasonable BER which is less than $\mathbf{10^{-1}}$ in case of 4x4 antenna configuration.

4.7 QOSTBC with proposed beamforming scheme:

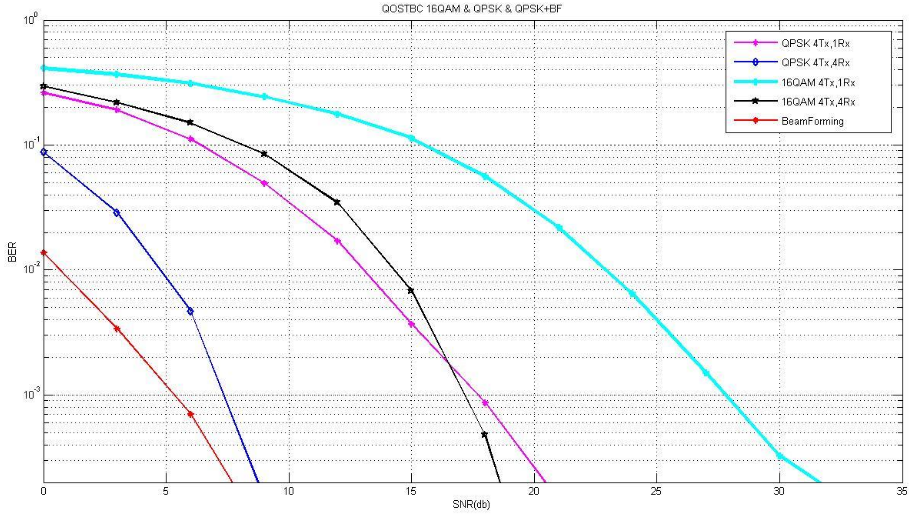

Figure (4.5): BER comparison for QOSTBC with proposed beamforming schemes

Finally the proposed scheme results are compared with the previously mentioned systems the QOSTBC with 16QAM and rotated QPSK, and as expected the beamforming scheme introduced better performance over other ones, instead of that the beamforming scheme proposed is only for 4x1 and it's harder for implementation with 4x4, but it achieves a minimum starting BER value of $\mathbf{10^{-2}}$, and this can be shown below in figure (4.5)

Chapter Five

Conclusion and Recommendations

5.1 Conclusion:

Demand for capacity in wireless communication systems has been rapidly growing worldwide. This has been driven by the increasing data rate requirements of wireless communication systems, and increasing demand for wireless Internet and multimedia services. As the available radio spectrum is limited, higher data rates can only be achieved by designing more efficient signaling techniques.

To improve spectral efficiency for future high data rate transmissions, it is desirable to construct Q OSTBCs with high order signal constellations. However, the design of QOSTBC increasing exponentially with constellation size and the number of transmit antennas.

This Work improve STBC reliability and data rates, we have derived Symbol Error Rate of Q OSTBC with 16 QAM and rotated QPSK versus SNR over Rayleigh fading channel. And we have obtained the numerical results by numerical integral that agree with the simulation results

In transmit diversity and beamforming; we transmit redundant information on different antennas. Full diversity is achieved with code rate equals to 1, the code rate of four transmit antennas used in this study is already proven as full diversity QOSTBC, so when combined with the beamforming technique it can maintain better reliability and network efficiency which achieved by lower bit error rate, and as checked in this study the proposed scheme of using beamforming technique with QOSTBC results better performance than with pure QOSTBC schemes.

A rather intuitive way to increase the overall power at the receiver is to use multiple antennas at the receiver side. This is known as receive diversity. Multiple antennas can also be used at the transmitter side, in what is known as transmit diversity. Transmit diversity approach based on beamforming uses multiple transmit antennas to focus the transmitted power in the direction of the receiver. This can potentially increase the received signal power and allow for higher data rates.

5.2 Recommendations:

This study can be developed by using a Channel State Information CSI as a feedback in case of mobility to perform a closed loop adaptive beamforming QOSTBC scheme.

Aclass of QOSTBC called 4Gp-QOSTBC that can always achieve full rate and full diversity for any number of transmit antennas, hence they can be applied in any system without the need for rate matching may be tested .

References

[1] A. Goldsmith. wireless Communications. 1st edition, New York, Cambridge

University Press,2005.

[2] D. Parsons, The Mobile Radio Propagation Channel. 2nd edition, New York,

Halsted Press, 1992.

[3] A. Goldsmith, Wireless Communica- tions, Cambridge Univ. Press, 2005.

[4] Jiayi Zhang, Michail Matthaiou, George K. Karagiannidis, Haibo Wang, and Zhenhui Tan,"Gallager's Exponent Analysis of STBC MIMO Systems over η-μ and κ-μ Fading channels" IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 61, NO. 3, MARCH 2013.

[5] Antonis Phasouliotis and Daniel K.C. So, "Performance Analysis and Comparison of Downlink MIMO MC-CDMA and MIMO OFDMA Systems" IEEE 2009.

[6] M. Jankiraman. Space-time Codes and MIMO Systems. 1st edition, London,

ARTECH HOUSE, INC., 2004.

[7] E.Biglieri, Y. Hong, and E. Viterbo, "On fast-decodable space-time block codes," IEEE Trans. Inf. Theory, vol. 55, no. 2, pp. 524–530, Feb. 2009.

[8] S. Haykin, "Continuous Wave Modulations," in Communications Systems, New York, 2001.

[9] Abu Hadrous Ammar. (2011, Feb 1). Wireless Com. Course slides. Available: http://site.iugaza.edu.ps/ahdrouss/files/2010/03/01-wireless-com1-introduction.

[10] K.J. Liu, Ahmed Sadek, Weifeng Su, and Andres Kwasinski, “Introduction,” in Cooperative Communications and Networking, New York, Cambridge University Press, 2009.

[11] E. Biglieri et al., “Diversity, interference cancellation and spatial multiplexing in MIMO mobile WiMAX systems”, to appear in IEEE WiMAX ‘07, 2007.

[12] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, \Cooperative diversity in wireless networks: E±cient protocols and outage behavior," IEEE Trans. In- form. Theory, vol. 50, pp. 3062{3080, Dec. 2004.

[13] . N. Laneman and G. W. Wornell, Distributed space-time-coded protocols

for exploiting cooperative diversity in wireless networks," IEEE Trans. Inform.

Theory, vol. 49, pp. 24152425, Oct. 2003.

[14] A. Sendonaris, E. Erkip, and B. Aazhang, \User cooperation diversity - Part I:

System description," IEEE Trans. Commun., vol. 51, pp. 19271938, Nov. 2003

[15] Y. Jing and B. Hassibi, \Distributed space-time coding in wireless relay net-

works," IEEE Trans. Wireless Commun., vol. 5, pp. 3524{3536, Dec. 2006.

[16] K. G. Seddik, A. K. Sadek, A. S. Ibrahim, and K. J. R. Liu, \Design criteria

and performance analysis for distributed sapce-time coding," IEEE Trans. Veh.

Technol., vol. 57, pp. 22802292, Jul. 2008.

[17] Y. Jing and B. Hassibi, \Distributed space-time coding in wireless relay net-

works," IEEE Trans. Wireless Commun., vol. 5, pp. 35243536, Dec. 2006.

[18] Kapil Gupta, and Pradip Kumar Ghosh, “Comparison of bit Error Performance of Rotated PSK scheme in Rayleigh and Ricean Fading Channel,” International journal of computer and electrical engineering (IJCEE), Singapore, Vol.04, No.01, pp.37-41, Feb 2012.

[19] A. Hottinen, O. Tirkkonen, R. Wichman, “Multi-Antenna Transceiver Techniques for 3G and Beyond”, John Wiley and Son Ltd., 2003.

[20] Louay M.A. Jalloul and Sam. P. Alex, "Evaluation Methodology and Performance of an IEEE 802.16e System", Presented to the IEEE Communications and Signal Processing Society, Orange County Joint Chapter (ComSig), December 7, 2006. Available at.

[21] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” IEEE J. Sel. Areas Commun., vol. 16, no. 8, pp1451–1458, Oct. 1998.

[22] I. Bahceci and T. M. Duman. Trellis-coded unitary space-time modulation.

IEEE Trans. Wireless Commun., 3(6):2005–2012, November 2004.

[23] Shreedhar A Joshi, T S Rukmini and Mahesh H M,"Space Time Block Coding for MIMO Systems using Alamouti Method With Digital Modulation Techniques",World Journal of Science and Technology 2011.

[24] SACHIN CHOURASIA, DR. PRABHAT PATEL," A Comparative analysis of orthogonal space time block code with trellis coded modulation over Rayleigh and Rician fading channel ", International Journal of Science, Engineering and Technology Research , February 2013.

[25] Jafarkhani, H. 'A quasi-orthogonal space-time block code', IEEE Transactions on Communications, 2001

[26] Sezginer, S. and Sari, H. (2007) 'Full-rate full-diversity 2 × 2 space-time codes of reduced decoder complexity', IEEE Communications Letters, Vol. 11, No. 12, pp.973–975.Vol. 49, No. 1, pp.1–4, 2007.

[27] Kim, J. and Cheun,K. 'An efficient decoding algorithmforQO-STBC based on iterative interference, cancellation', IEEE Communications Letters, Vol. 12, No. 4, pp.292–294, 2008.

[28] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," IEEE Trans. Inform. Theory, vol. 45, pp. 1456–1467, July 1999

[29] A.V. Geramita and J. Seberry, Orthogonal Designs, Quadratic Forms and Hadamard Matrices, Lecture Notes in Pure and Applied Mathematics, vol. 43. New York and Basel: Marcel Dekker, 1979.

[30] S. Alamouti, "A simple transmit diversity technique for wireless communications," IEEE J. Select. Areas Commun., vol. 16, pp. 1451–1458, Aug. 1998.

[31] X.-B. Liang, “A high-rate orthogonal space-time block code,” IEEE Commun. Lett., vol. 7, May 2003.

[32] H. Jafarkhani, “A quasi-orthogonal space-time block code,” IEEE Trans. Commun., vol. 49, pp. 1–4, Jan. 2001.

[33] C. B. Papadias and G. J. Foschini, “Capacity-approaching space-time codes for systems employing four transmitter antennas,” IEEE Trans. Inform. Theory, vol. 49, pp. 726–732, Mar. 2003.

[34] O. Tirkkonen, A. Boariu, and A. Hottinen, “Minimal nonorthogonality rate 1 space-time block code for 3+ Tx antennas,” in Proc. IEEE 6th Int. Symp. Spread-Spectrum Techniques and Applications (ISSSTA 2000), Sept. 2000, pp. 429–432.

[35] S Ahmed Ghauri, Hannan Adeel, Bilal Irshad, Agha Saif, "Performance of quasi orthogonal space time codes on Rayleigh and Rician fading channel", International Journal of Computer and Electronics Research,Vol. 2, Issue 4, August 2013.

[36] Aresh Daruwalla, Allen Paul, “Space-Time Block Codes (STBC) for 4 transmits antennas", Press of School of Electrical Engineering and Telecommunications, University of New South Wales, Australia, 2009.

[37] Priyanka Mishra, Rahul Vij, Gurpreet Singh, Gaurav Chandil, "Study and Simulation of Quasi and Rotated Quasi Space Time Block Codes in MIMO systems using Dent Channel model", International Journal of Wireless & Mobile Networks (IJWMN) Vol. 4, Issue 5, October 2012.

[38] T. Cover and J. Thomas, Elements of Information Theory, Wiley, 2001.

[39] Khoa Nguyen Le, Anthony C. K. Soong, Erchin Serpedin, (2010), "Orthogonal Frequency Division Multiplexing with Diversity for Future Wireless Systems", School of Engineering, University of Western Sydney, Australia.

[40] F. Behnamfar, “Single and Multiple Antenna Communication “Systems: Performance Analysis and Joint Source-Channel Coding”, Ph.D. Dissertation, Dept. of Electrical & Computer Engineering, Queen’s University, Kingston, Canada, Sept. 2004.

[41] Lijun Liu, Myoung-Seob Lim, “An Efficient Selective Receiver for Multiple-Input Multiple-Output Scheme", IEEE Trans Inform Theory, Vol. 56, No 5, June 2010.

[42] Y.A.S. Dama, R.A. Abd-Alhameed, T.S. Ghazaany, S. Zhu, "A New Approach for OSTBC and QOSTBC", International Journal of Computer Applications, Vol 67, No.6, April 2013.

Appendix

<u>Simulation MATLAB Code:</u>

```
% This program simulates the bit-error-rate (BER) performance of quasi-orthogonal
% space time block code (QOSTBC)
% with L=4 and M=1 antennas over the frequency flat Rayleigh block fading channel
% The costellation used: QPSK for s1 and s2. Rotated Pi/4 QPSK for s3 and s4.
% Such design yields to full rate code and 2 bis/sec/Hz
clc
close all;
%QPSK constellation
for modetype=2:2:4;
for Rnum=1:3:4
   switch modetype
      case 2
x_q=[1  1 -1 -1];
y_q=[1 -1  1 -1];
QPSK_s=x_q+y_q*i;
QPSK_s=QPSK_s/sqrt(2);
QPSK_r=QPSK_s;
```

```
        case 4
x_q=[-3 -3 -3 -3   -1 -1 -1 -1  3 3 3  3    1 1 1  1  ];
y_q=[3  1  -3 -1    3  1 -3 -1  3 1 -3 -1   3 1 -3 -1 ];
QPSK_s=x_q+y_q*i; % 16QAM
QPSK_s=QPSK_s/sqrt(10);
%rotated Pi/2 QPSK constellation
% x_r=[0         sqrt(2)  -sqrt(2)  0];
% y_r=[sqrt(2)   0        0       -sqrt(2)];
QPSK_r=QPSK_s;
% QPSK_r=QPSK_r/sqrt(2);
    end;
Es=4; %average constellation energy for d=2;
% index=1;
%  BER1=zeros(1,8);
%  BER2=zeros(1,8);
%
% for Rnum=1:3:4
  index=1;
for SNR=0:3:33
%number of bit errors set to zero
count=0;
%number of iterations
N=10000;
```

```
 if SNR>=20
     N=2*10^4;
 end; % increase for higher SNR
for it=1:N
%generate 8 bits uniformly distributed
switch modetype
    case 2
       A=round(rand(1,8));
% TX symbols
% QPSK
s1=QPSK_s(bi2de(A(1:2),'left-msb')+1);
s2=QPSK_s(bi2de(A(3:4),'left-msb')+1);
%rotated QPSK
s3=QPSK_r(bi2de(A(5:6),'left-msb')+1);
s4=QPSK_r(bi2de(A(7:8),'left-msb')+1);
C=[s1 s2 s3 s4; -conj(s2) conj(s1) -conj(s4) conj(s3); -conj(s3) -conj(s4)
conj(s1) conj(s2); s4 -s3 -s2 s1];
switch Rnum
    case 1
%Channel fading coefficients
K=1/sqrt(2)*(randn(4,1)+i*randn(4,1)); % normalize to the variance 1
CN(0,1)
%AWGN channel
N0=(2*Es/10^(SNR/10));
```

```
Z=sqrt(N0/2)*(randn(4,1)+i*randn(4,1)); % variance N0

%RX symbols

R=C*K+Z;

%pair-wise ML decoding QOSTBC

%find min s1 and s4 for different combinations

vector_c=combvec(QPSK_r,QPSK_s).';

qpsk_space=vector_c(:,2);

rotated_space=vector_c(:,1);

%cost function f1_4

f_1_4=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K).^2)) +
2*real ( (-K(1)*conj(R(1))- conj(K(2))*R(2)-conj(K(3))*R(3)-
K(4)*conj(R(4)))*qpsk_space+rotated_space*(-
K(4)*conj(R(1))+conj(K(3))*R(2)+conj(K(2))*R(3)-K(1)*conj(R(4)) )
)+4*real( K(1)*conj(K(4))-conj(K(2))*K(3) )*real(
qpsk_space.*conj(rotated_space));

%find for S1 corresponding bits

[~,I]=min(f_1_4);

QPSK_s=vector_c(I,2);

[~,I]=min(QPSK_s-vector_c(I,2));

dec_bits(1,1:2)=de2bi(I(1)-1,2,'left-msb');

%find for S4 corresponding bits

[~,I]=min(f_1_4);

[~,I]=min(QPSK_r-vector_c(I,1));

dec_bits(1,7:8)=de2bi(I(1)-1,2,'left-msb');
```

```
%cost function f2_3

f_2_3=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K).^2)) +
2*real ( (-K(2)*conj(R(1))+ conj(K(1))*R(2)-
conj(K(4))*R(3)+K(3)*conj(R(4)))*qpsk_space+rotated_space*(-
K(3)*conj(R(1))-conj(K(4))*R(2)+conj(K(1))*R(3)+K(2)*conj(R(4)) )
)+4*real( K(2)*conj(K(3))-conj(K(1))*K(4) )*real(
qpsk_space.*conj(rotated_space));

%find for S2 corresponding bits

[~,I]=min(f_2_3);

[~,I]=min(QPSK_s-vector_c(I,2));

dec_bits(1,3:4)=de2bi(I(1)-1,2,'left-msb');



%find for S3 corresponding bits

[~,I]=min(f_2_3);

[H1,I]=min(QPSK_r-vector_c(I,1));

dec_bits(1,5:6)=de2bi(I(1)-1,2,'left-msb');

    case 4

        Z=0;

K1=1/sqrt(2)*(randn(4,1)+i*randn(4,1)); % normalize to the variance 1
CN(0,1)

K2=1/sqrt(2)*(randn(4,1)+i*randn(4,1));

K3=1/sqrt(2)*(randn(4,1)+i*randn(4,1));

K4=1/sqrt(2)*(randn(4,1)+i*randn(4,1));
```

```
%AWGN channel
N0=(2*Es/10^(SNR/10));
Z1=sqrt(N0/2)*(randn(4,1)+i*randn(4,1)); % variance N0
Z2=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
Z3=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
Z4=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
%RX symbols
R1=C*K1+Z1;
R2=C*K2+Z2;
R3=C*K3+Z3;
R4=C*K4+Z4;
%pair-wise ML decoding QOSTBC
%find min s1 and s4 for different combinations
vector_c=combvec(QPSK_r,QPSK_s).';
qpsk_space=vector_c(:,2);
rotated_space=vector_c(:,1);
%cost function f1_4
f_1_4=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K1).^2))
+ 2*real ( (-K1(1)*conj(R1(1))- conj(K1(2))*R1(2)-conj(K1(3))*R1(3)-
K1(4)*conj(R1(4)))*qpsk_space+rotated_space*(-
K1(4)*conj(R1(1))+conj(K1(3))*R1(2)+conj(K1(2))*R1(3)-
K1(1)*conj(R1(4)) )  )+4*real( K1(1)*conj(K1(4))-conj(K1(2))*K1(3)
)*real( qpsk_space.*conj(rotated_space))+...
```

```
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K2).^2)) + 2*real
( (-K2(1)*conj(R2(1))- conj(K2(2))*R2(2)-conj(K2(3))*R2(3)-
K2(4)*conj(R2(4)))*qpsk_space+rotated_space*(-
K2(4)*conj(R2(1))+conj(K2(3))*R2(2)+conj(K2(2))*R2(3)-
K2(1)*conj(R2(4)) )  )+4*real( K2(1)*conj(K2(4))-conj(K2(2))*K2(3)
)*real( qpsk_space.*conj(rotated_space))+...
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K3).^2)) + 2*real
( (-K3(1)*conj(R3(1))- conj(K3(2))*R3(2)-conj(K3(3))*R3(3)-
K3(4)*conj(R3(4)))*qpsk_space+rotated_space*(-
K3(4)*conj(R3(1))+conj(K3(3))*R3(2)+conj(K3(2))*R3(3)-
K3(1)*conj(R3(4)) )  )+4*real( K3(1)*conj(K3(4))-conj(K3(2))*K3(3)
)*real( qpsk_space.*conj(rotated_space))+...
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K4).^2)) + 2*real
( (-K4(1)*conj(R4(1))- conj(K4(2))*R4(2)-conj(K4(3))*R4(3)-
K4(4)*conj(R4(4)))*qpsk_space+rotated_space*(-
K4(4)*conj(R4(1))+conj(K4(3))*R4(2)+conj(K4(2))*R4(3)-
K4(1)*conj(R4(4)) )  )+4*real( K4(1)*conj(K4(4))-conj(K4(2))*K4(3)
)*real( qpsk_space.*conj(rotated_space)) ;
  %find for S1 corresponding bits
[~,I]=min(f_1_4);
[~,I]=min(QPSK_s-vector_c(I,2));
dec_bits(1,1:2)=de2bi(I(1)-1,2,'left-msb');
%find for S4 corresponding bits
[~,I]=min(f_1_4);
[~,I]=min(QPSK_r-vector_c(I,1));
dec_bits(1,7:8)=de2bi(I(1)-1,2,'left-msb');
```

```
%cost function f2_3

f_2_3=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K1).^2))
+ 2*real ( (-K1(2)*conj(R1(1))+ conj(K1(1))*R1(2)-
conj(K1(4))*R1(3)+K1(3)*conj(R1(4)))*qpsk_space+rotated_space*(-
K1(3)*conj(R1(1))-
conj(K1(4))*R1(2)+conj(K1(1))*R1(3)+K1(2)*conj(R1(4)) )  )+4*real(
K1(2)*conj(K1(3))-conj(K1(1))*K1(4) )*real(
qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K2).^2)) + 2*real
( (-K2(2)*conj(R2(1))+ conj(K2(1))*R2(2)-
conj(K2(4))*R2(3)+K2(3)*conj(R2(4)))*qpsk_space+rotated_space*(-
K2(3)*conj(R2(1))-
conj(K2(4))*R2(2)+conj(K2(1))*R2(3)+K2(2)*conj(R2(4)) )  )+4*real(
K2(2)*conj(K2(3))-conj(K2(1))*K2(4) )*real(
qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K3).^2)) + 2*real
( (-K3(2)*conj(R3(1))+ conj(K3(1))*R3(2)-
conj(K3(4))*R3(3)+K3(3)*conj(R3(4)))*qpsk_space+rotated_space*(-
K3(3)*conj(R3(1))-
conj(K3(4))*R3(2)+conj(K3(1))*R3(3)+K3(2)*conj(R3(4)) )  )+4*real(
K3(2)*conj(K3(3))-conj(K3(1))*K3(4) )*real(
qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K4).^2)) + 2*real
( (-K4(2)*conj(R4(1))+ conj(K4(1))*R4(2)-
conj(K4(4))*R4(3)+K4(3)*conj(R4(4)))*qpsk_space+rotated_space*(-
K4(3)*conj(R4(1))-
conj(K4(4))*R4(2)+conj(K4(1))*R4(3)+K4(2)*conj(R4(4)) )  )+4*real(
```

```
K4(2)*conj(K4(3))-conj(K4(1))*K4(4) )*real(
qpsk_space.*conj(rotated_space));


%find for S2 corresponding bits
[~,I]=min(f_2_3);
[~,I]=min(QPSK_s-vector_c(I,2));
dec_bits(1,3:4)=de2bi(I(1)-1,2,'left-msb');
%find for S3 corresponding bits
[~,I]=min(f_2_3);
[~,I]=min(QPSK_r-vector_c(I,1));
dec_bits(1,5:6)=de2bi(I(1)-1,2,'left-msb');
end;
%count errors
%count errors
% count=count+sum(abs(A-dec_bits));
   case 4
A=round(rand(1,16));
% TX symbols
% QPSK
s1=QPSK_s(bi2de(A(1:4),'left-msb')+1);
s2=QPSK_s(bi2de(A(5:8),'left-msb')+1);
%rotated QPSK
s3=QPSK_r(bi2de(A(9:12),'left-msb')+1);
```

```
s4=QPSK_r(bi2de(A(13:16),'left-msb')+1);
C=[s1 s2 s3 s4; -conj(s2) conj(s1) -conj(s4) conj(s3); -conj(s3) -conj(s4)
conj(s1) conj(s2); s4 -s3 -s2 s1];
switch Rnum
    case 1
%Channel fading coefficients
K=1/sqrt(2)*(randn(4,1)+i*randn(4,1)); % normalize to the variance 1
CN(0,1)
%AWGN channel
N0=(4*Es/10^(SNR/10));
Z=sqrt(N0/2)*(randn(4,1)+i*randn(4,1)); % variance N0
%RX symbols
R=C*K+Z;



%pair-wise ML decoding QOSTBC
%find min s1 and s4 for different combinations
vector_c=combvec(QPSK_r,QPSK_s).';
qpsk_space=vector_c(:,2);
rotated_space=vector_c(:,1);
%cost function f1_4
f_1_4=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K).^2)) +
2*real ( (-K(1)*conj(R(1))- conj(K(2))*R(2)-conj(K(3))*R(3)-
K(4)*conj(R(4)))*qpsk_space+rotated_space*(-
```

```
K(4)*conj(R(1))+conj(K(3))*R(2)+conj(K(2))*R(3)-K(1)*conj(R(4)) )
)+4*real( K(1)*conj(K(4))-conj(K(2))*K(3) )*real(
qpsk_space.*conj(rotated_space));

%find for S1 corresponding bits

[~,I]=min(f_1_4);

QPSK_s=vector_c(I,2);

[~,I]=min(QPSK_s-vector_c(I,2));

dec_bits(1,1:4)=de2bi(I(1)-1,4,'left-msb');

%find for S4 corresponding bits

[~,I]=min(f_1_4);

[~,I]=min(QPSK_r-vector_c(I,1));

dec_bits(1,13:16)=de2bi(I(1)-1,4,'left-msb');

%cost function f2_3

f_2_3=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K).^2)) +
2*real ( (-K(2)*conj(R(1))+ conj(K(1))*R(2)-
conj(K(4))*R(3)+K(3)*conj(R(4)))*qpsk_space+rotated_space*(-
K(3)*conj(R(1))-conj(K(4))*R(2)+conj(K(1))*R(3)+K(2)*conj(R(4)) )
)+4*real( K(2)*conj(K(3))-conj(K(1))*K(4) )*real(
qpsk_space.*conj(rotated_space));

%find for S2 corresponding bits

[~,I]=min(f_2_3);

[~,I]=min(QPSK_s-vector_c(I,2));

dec_bits(1,5:8)=de2bi(I(1)-1,4,'left-msb');

%find for S3 corresponding bits

[~,I]=min(f_2_3);
```

```
[H1,I]=min(QPSK_r-vector_c(I,1));
dec_bits(1,9:12)=de2bi(I(1)-1,4,'left-msb');
    case 4
        Z=0;
K1=1/sqrt(2)*(randn(4,1)+i*randn(4,1)); % normalize to the variance 1
CN(0,1)
K2=1/sqrt(2)*(randn(4,1)+i*randn(4,1));
K3=1/sqrt(2)*(randn(4,1)+i*randn(4,1));
K4=1/sqrt(2)*(randn(4,1)+i*randn(4,1));
%AWGN channel
N0=(4*Es/10^(SNR/10));
Z1=sqrt(N0/2)*(randn(4,1)+i*randn(4,1)); % variance N0
Z2=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
Z3=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
Z4=sqrt(N0/2)*(randn(4,1)+i*randn(4,1));
%RX symbols
R1=C*K1+Z1;
R2=C*K2+Z2;
R3=C*K3+Z3;
R4=C*K4+Z4;
%pair-wise ML decoding QOSTBC
%find min s1 and s4 for different combinations
vector_c=combvec(QPSK_r,QPSK_s).';
```

```
qpsk_space=vector_c(:,2);

rotated_space=vector_c(:,1);

%cost function f1_4

f_1_4=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K1).^2))
+ 2*real ( (-K1(1)*conj(R1(1))- conj(K1(2))*R1(2)-conj(K1(3))*R1(3)-
K1(4)*conj(R1(4)))*qpsk_space+rotated_space*(-
K1(4)*conj(R1(1))+conj(K1(3))*R1(2)+conj(K1(2))*R1(3)-
K1(1)*conj(R1(4)) )  )+4*real( K1(1)*conj(K1(4))-conj(K1(2))*K1(3)
)*real( qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K2).^2)) + 2*real
( (-K2(1)*conj(R2(1))- conj(K2(2))*R2(2)-conj(K2(3))*R2(3)-
K2(4)*conj(R2(4)))*qpsk_space+rotated_space*(-
K2(4)*conj(R2(1))+conj(K2(3))*R2(2)+conj(K2(2))*R2(3)-
K2(1)*conj(R2(4)) )  )+4*real( K2(1)*conj(K2(4))-conj(K2(2))*K2(3)
)*real( qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K3).^2)) + 2*real
( (-K3(1)*conj(R3(1))- conj(K3(2))*R3(2)-conj(K3(3))*R3(3)-
K3(4)*conj(R3(4)))*qpsk_space+rotated_space*(-
K3(4)*conj(R3(1))+conj(K3(3))*R3(2)+conj(K3(2))*R3(3)-
K3(1)*conj(R3(4)) )  )+4*real( K3(1)*conj(K3(4))-conj(K3(2))*K3(3)
)*real( qpsk_space.*conj(rotated_space))+...

(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K4).^2)) + 2*real
( (-K4(1)*conj(R4(1))- conj(K4(2))*R4(2)-conj(K4(3))*R4(3)-
K4(4)*conj(R4(4)))*qpsk_space+rotated_space*(-
K4(4)*conj(R4(1))+conj(K4(3))*R4(2)+conj(K4(2))*R4(3)-
K4(1)*conj(R4(4)) )  )+4*real( K4(1)*conj(K4(4))-conj(K4(2))*K4(3)
)*real( qpsk_space.*conj(rotated_space)) ;
```

```
 %find for S1 corresponding bits
[~,I]=min(f_1_4);
[~,I]=min(QPSK_s-vector_c(I,2));
dec_bits(1,1:4)=de2bi(I(1)-1,4,'left-msb');
%find for S4 corresponding bits
[~,I]=min(f_1_4);
[~,I]=min(QPSK_r-vector_c(I,1));
dec_bits(1,13:16)=de2bi(I(1)-1,4,'left-msb');
%cost function f2_3
f_2_3=(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K1).^2))
+ 2*real ( (-K1(2)*conj(R1(1))+ conj(K1(1))*R1(2)-
conj(K1(4))*R1(3)+K1(3)*conj(R1(4)))*qpsk_space+rotated_space*(-
K1(3)*conj(R1(1))-
conj(K1(4))*R1(2)+conj(K1(1))*R1(3)+K1(2)*conj(R1(4)) )  )+4*real(
K1(2)*conj(K1(3))-conj(K1(1))*K1(4) )*real(
qpsk_space.*conj(rotated_space))+...
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K2).^2)) + 2*real
( (-K2(2)*conj(R2(1))+ conj(K2(1))*R2(2)-
conj(K2(4))*R2(3)+K2(3)*conj(R2(4)))*qpsk_space+rotated_space*(-
K2(3)*conj(R2(1))-
conj(K2(4))*R2(2)+conj(K2(1))*R2(3)+K2(2)*conj(R2(4)) )  )+4*real(
K2(2)*conj(K2(3))-conj(K2(1))*K2(4) )*real(
qpsk_space.*conj(rotated_space))+...
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K3).^2)) + 2*real
( (-K3(2)*conj(R3(1))+ conj(K3(1))*R3(2)-
conj(K3(4))*R3(3)+K3(3)*conj(R3(4)))*qpsk_space+rotated_space*(-
```

```
K3(3)*conj(R3(1))-
conj(K3(4))*R3(2)+conj(K3(1))*R3(3)+K3(2)*conj(R3(4)) )  )+4*real(
K3(2)*conj(K3(3))-conj(K3(1))*K3(4) )*real(
qpsk_space.*conj(rotated_space))+...
(abs(qpsk_space).^2+abs(rotated_space).^2)* (sum(abs(K4).^2)) + 2*real
( (-K4(2)*conj(R4(1))+ conj(K4(1))*R4(2)-
conj(K4(4))*R4(3)+K4(3)*conj(R4(4)))*qpsk_space+rotated_space*(-
K4(3)*conj(R4(1))-
conj(K4(4))*R4(2)+conj(K4(1))*R4(3)+K4(2)*conj(R4(4)) )  )+4*real(
K4(2)*conj(K4(3))-conj(K4(1))*K4(4) )*real(
qpsk_space.*conj(rotated_space));
%find for S2 corresponding bits
[~,I]=min(f_2_3);
[~,I]=min(QPSK_s-vector_c(I,2));
dec_bits(1,5:8)=de2bi(I(1)-1,4,'left-msb');
%find for S3 corresponding bits
[~,I]=min(f_2_3);
[H1,I]=min(QPSK_r-vector_c(I,1));
dec_bits(1,9:12)=de2bi(I(1)-1,4,'left-msb');


end;
end;
%count errors
count=count+sum(abs(A-dec_bits));
end;
```

```
switch modetype
   case 2
switch Rnum
   case 1
      BER1indexcounT(N*8)
      index=index+1;
   case 4
      BER2
indexcount(N*8)
      index=index+1;
end;
   case 4
      switch Rnum
         case 1
            BER3
          indexcount(N*16)
            index=index+1;
         case 4
            BER4
indexcount(N*16)
            index=index+1;
      end;
end;
```

```
end;%HERE
end;
end;

SNR_dB=0:3:33;
len=length(SNR_dB);
for i=1:1:len
    SNR_dBi;
    BF_BE
iQOSTBC_QPSK_BF(SNR_dB(i));
end
xaxis=0:3:33;
figure(1);
semilogy(xaxis, BER1,'*-m',xaxis,BER2,'d-b',xaxis,BER3,'+-
c',xaxis,BER4,'p-k',xaxis,BF_BE,'*-r');
legend('QPSK 4Tx,1Rx','QPSK 4Tx,4Rx','16QAM 4Tx,1Rx','16QAM
4Tx,4Rx','BeamForming');
grid on;
xlabel('SNR(db)');
ylabel('BER');
title('QOSTBC 16QAM & QPSK & QPSK+BF');
```