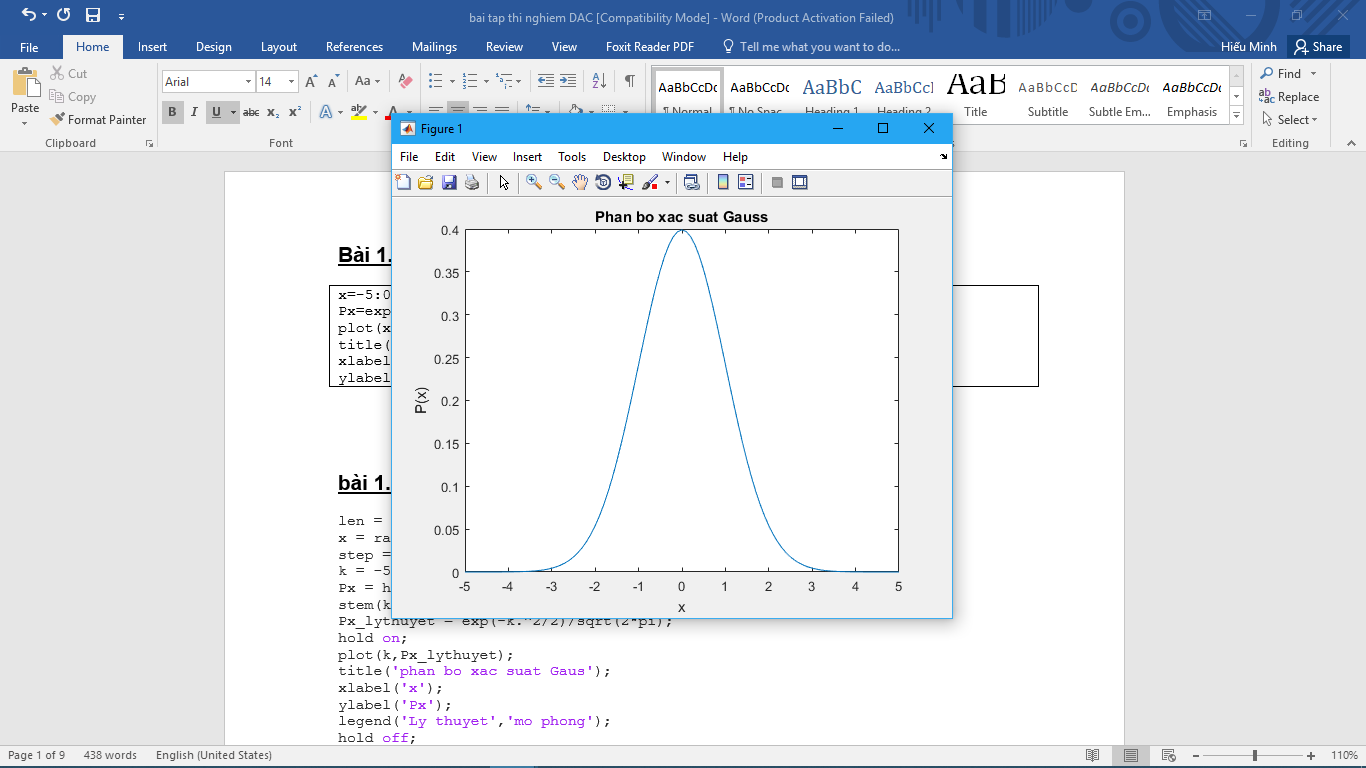
**Họ tên: Nguyễn Minh Hiếu**

**MSSV: 20151336**

**Lớp: Điện tử 3 K60**

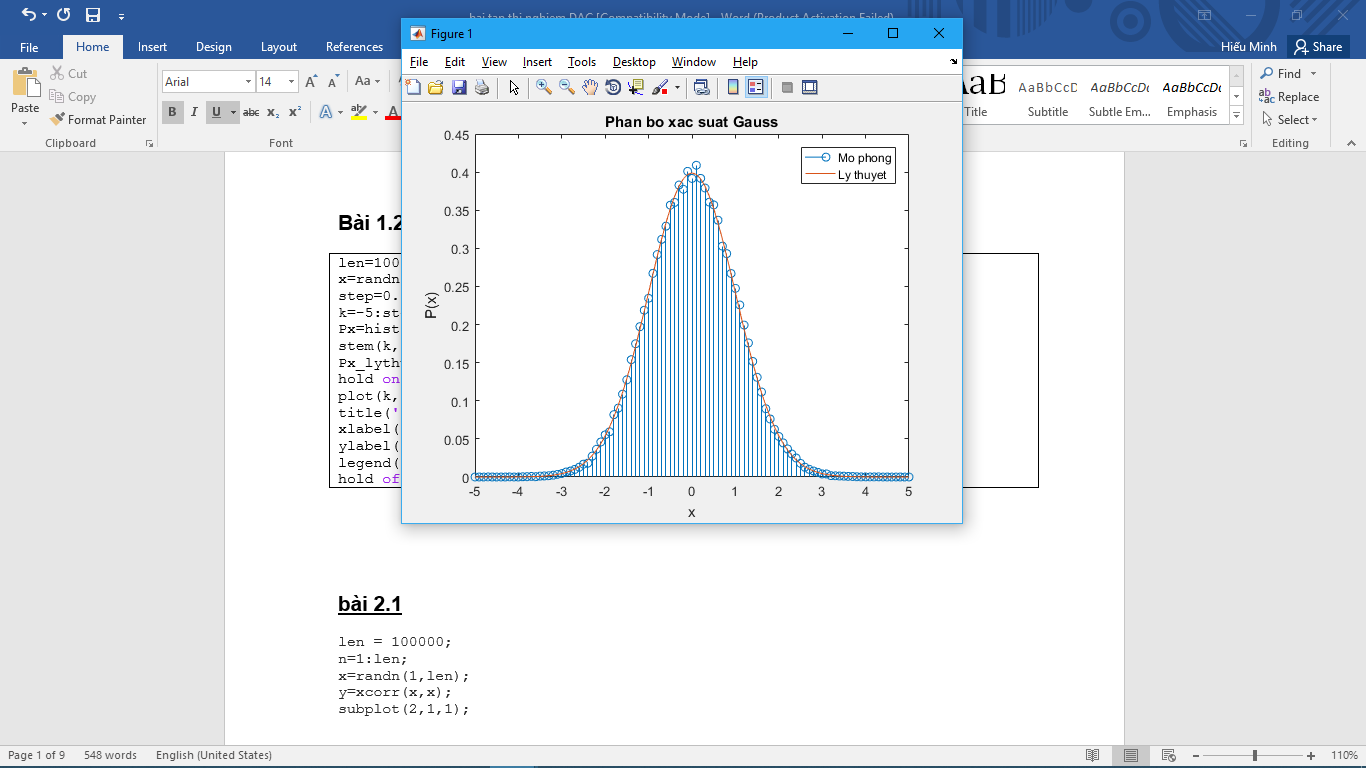
**Bài 1.1**

|  |
| --- |
| x=-5:0.1:5; % Tao vector hang x co gia tri tu -5 den 5 cach deu nhau 0.1  Px=exp(-x.^2/2)/sqrt(2\*pi); % Tinh ham Px  plot(x,Px); % Ve do thi  title('Phan bo xac suat Gauss'); % Dat ten do thi  xlabel('x'); % Dat ten truc hoanh  ylabel('P(x)'); % Dat ten truc tung |

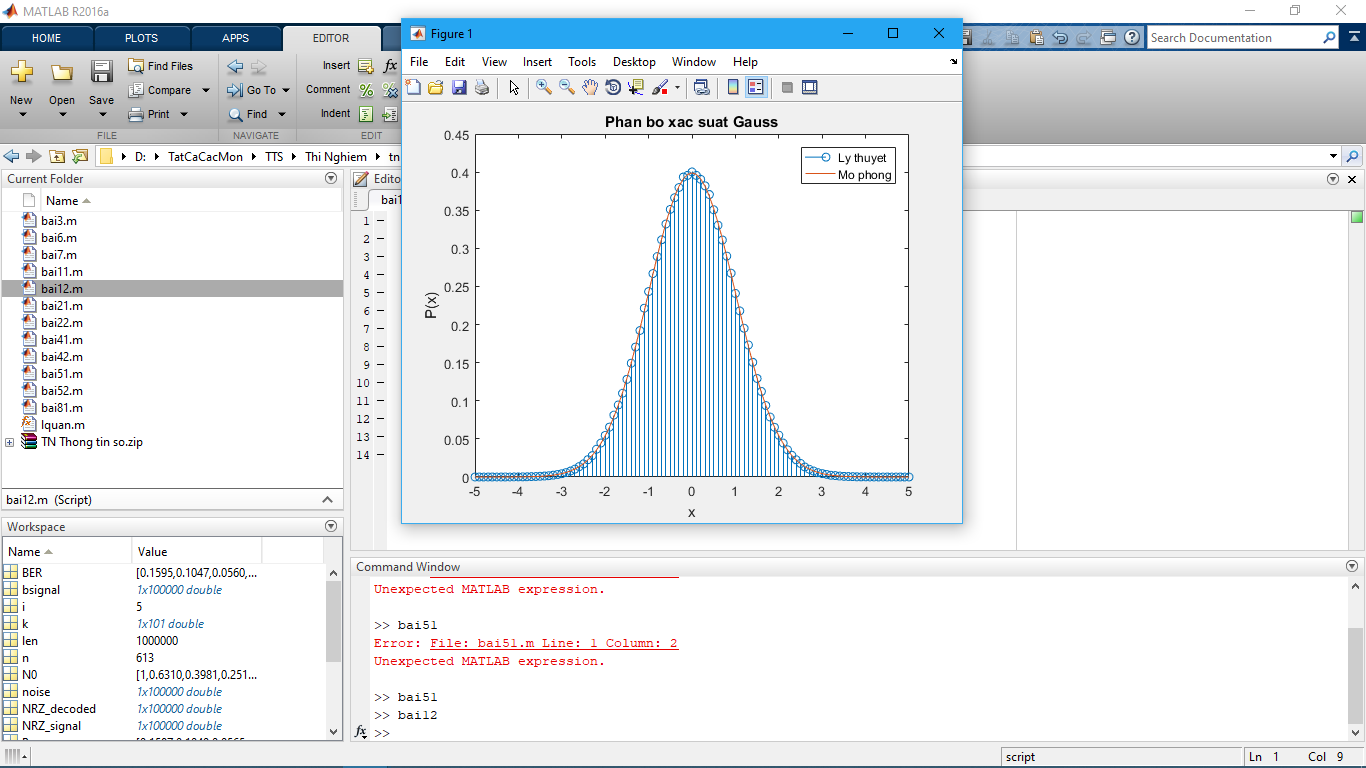


**Bài 1.2**

|  |
| --- |
| len=100000; % Do dai cua qua trinh ngau nhien  x=randn(1,len); % Tao qua trinh ngau nhien theo phan phoi chuan  step=0.1; % step = 0.1  k=-5:step:5; % khoang xet tu -5 den 5 cach deu 0.1  Px=hist(x,k)/len/step; % Xac dinh so vector  stem(k,Px); % Ve do thi roi rac cua Px mo phong  Px\_lythuyet=exp(-k.^2/2)/sqrt(2\*pi); % Tinh Px ly thuyet  hold on; % Giu do thi da ve  plot(k,Px\_lythuyet); % Ve do thi Px ly thuyet  title('Phan bo xac suat Gauss'); % Dat ten do thi  xlabel('x'); % Dat ten truc hoanh  ylabel('P(x)'); % Dat ten truc tung  legend('Mo phong','Ly thuyet'); % Ki hieu  hold off; |



**Q6: Thay đổi độ dài quá trình ngẫu nhiên lên 1.000.000 phần từ**

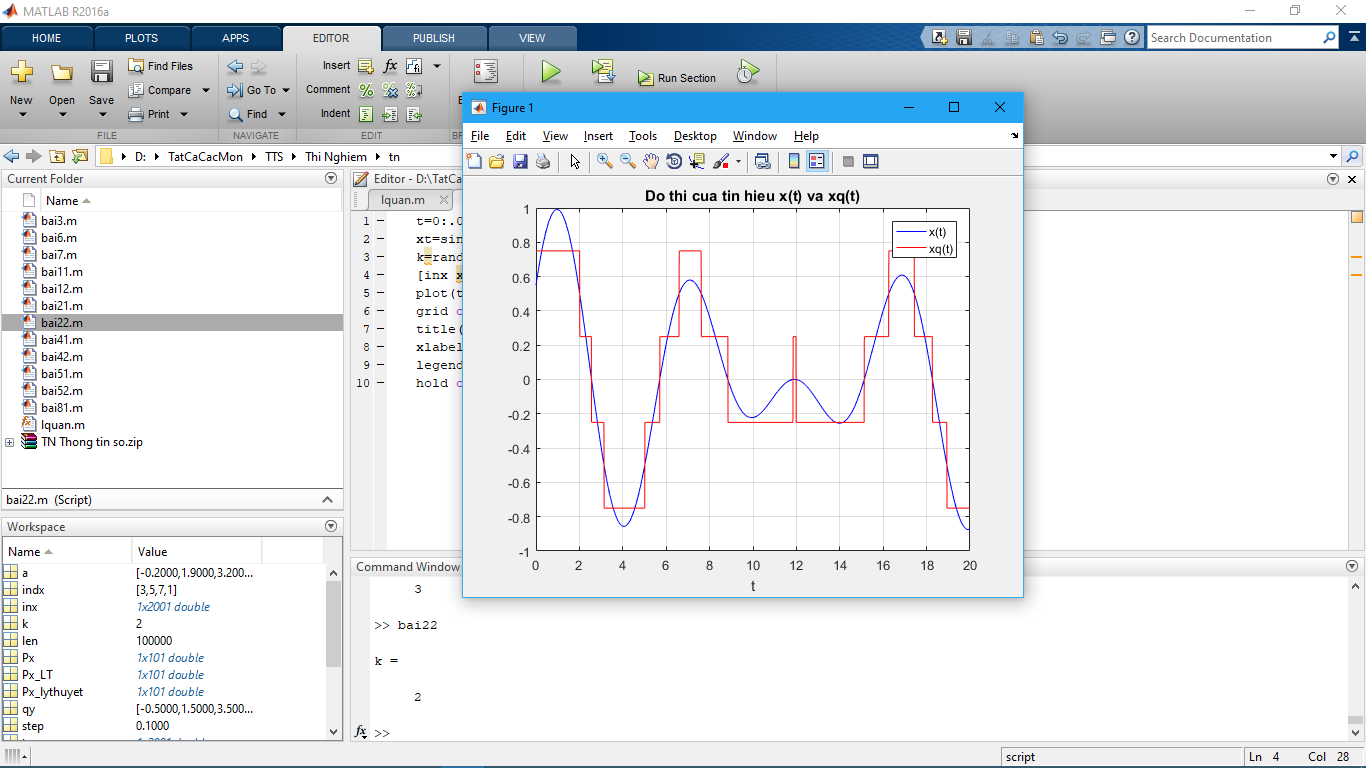


**Bài 2.1**

|  |
| --- |
| function [id qy]= lquan(x,xmin,xmax,nbit)  nlevel = 2^nbit; % So muc luong tu hoa  q = (xmax-xmin)/nlevel; % Buoc luong tu  [id qy] = quantiz(x,xmin+q:q:xmax-q,xmin+q/2:q:xmax-q/2); % Su dung ham quantiz co san  xs = rand(1,5)\*2-1  [xi xq] = lquan(xs,-1,1,3) |

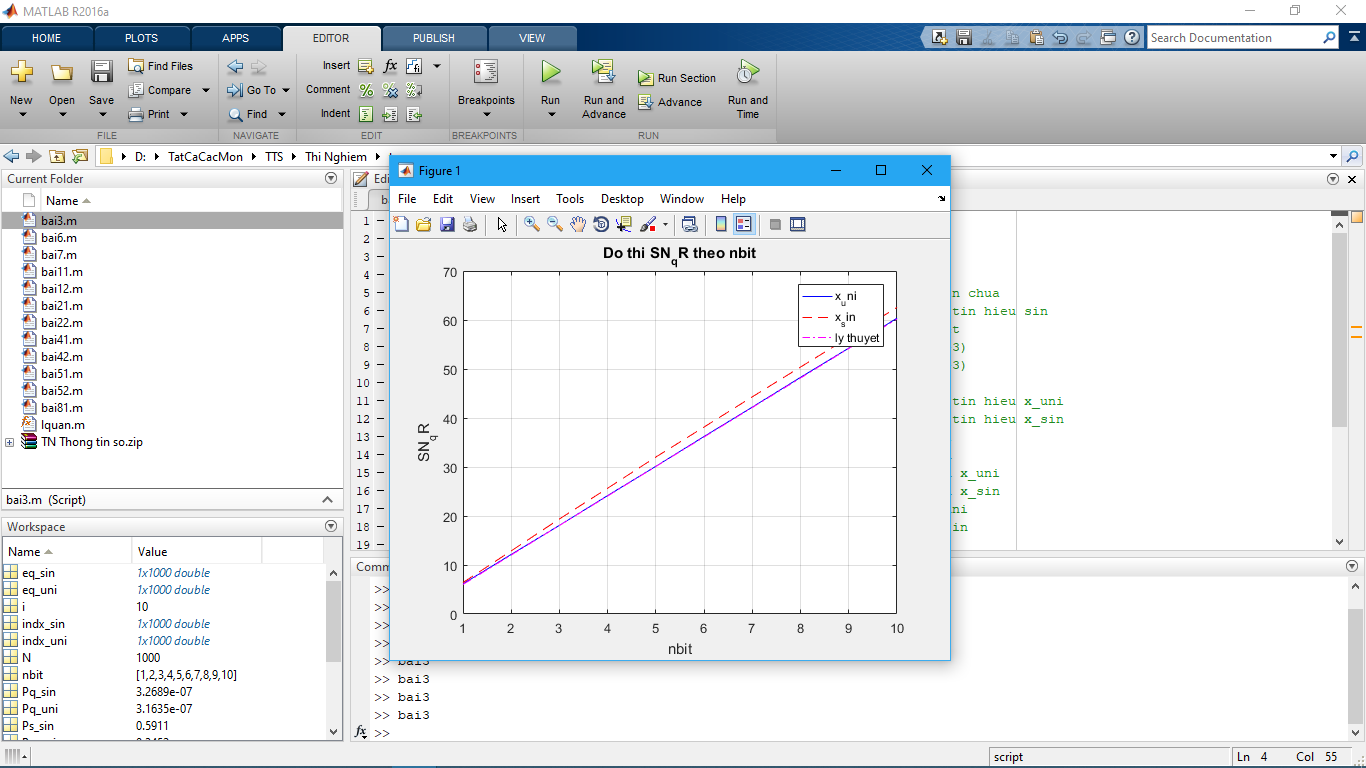
**Bài 2.2**

|  |
| --- |
| t=0:.01:20; % Khoang thoi gian xet tu 0 den 20 cach nhau 0.01  xt=sin(randn()+t).\*cos(rand()\*t); % Tin hieu ngau nhien co bien do -1 den 1  [inx xqt] = lquan(xt,-1,1,randi(3)+1); % Tinh xqt  plot(t,xt,'b',t,xqt,'r'); % Ve do thi xt va xqt  grid on; % Bat luoi  title('Do thi cua tin hieu x(t) va xq(t)'); % Dat ten do thi  xlabel('t'); % Dat ten truc hoanh  legend('x(t)','xq(t)'); % Ki hieu  hold off; |



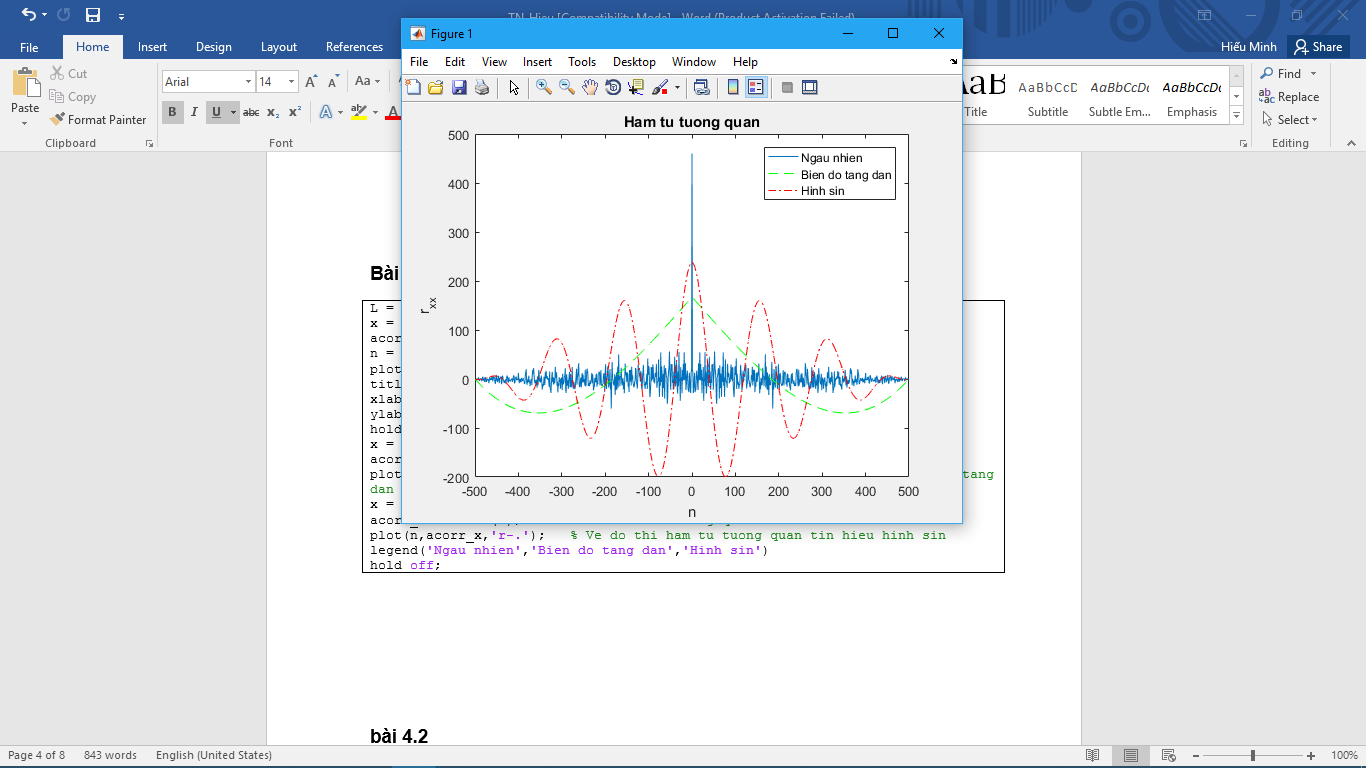
**Bài 3**

|  |
| --- |
| N = 1000;  x\_uni = 2\*rand(1,N)-1; % x\_uni phan bo deu tu -1 den 1  x\_sin = sin(linspace(1,5,N)); % tin hieu sin  nbit = 1:10; % so bit luong tu hoa tu 1 den 10  SNqR\_uni = zeros(size(nbit)); % Khoi tao mang SNqR\_uni va SNqR\_sin chua  SNqR\_sin = zeros(size(nbit)); % SNqR cua tin hieu phan bo deu va tin hieu sin  SNqR\_lt = 6.02\*nbit; % Mang chua SNqR tinh theo ly thuyet  Ps\_uni = sum(x\_uni.^2)/N; % Cong suat tin hieu x\_uni theo (3-3)  Ps\_sin = sum(x\_sin.^2)/N; % Cong suat tin hieu x\_sin theo (3-3)  for i = 1:size(nbit,2) % size(n,2) tra ve so cot cua n  [indx\_uni xq\_uni] = lquan(x\_uni,-1,1,nbit(i)); % Luong tu hoa tin hieu x\_uni  [indx\_sin xq\_sin] = lquan(x\_sin,-1,1,nbit(i)); % Luong tu hoa tin hieu x\_sin  eq\_uni = x\_uni - xq\_uni; % tinh sai so luong tu hoa x\_uni  eq\_sin = x\_sin - xq\_sin; % tinh sai so luong tu hoa x\_sin  Pq\_uni = sum(eq\_uni.^2)/N; % tinh cong suat tap am luong tu x\_uni  Pq\_sin = sum(eq\_sin.^2)/N; % tinh cong suat tap am luong tu x\_sin  SNqR\_uni(i) = 10\*log10(Ps\_uni/Pq\_uni); % tinh SNR\_db cua x\_uni  SNqR\_sin(i) = 10\*log10(Ps\_sin/Pq\_sin); % tinh SNR\_db cua x\_sin  end  plot(nbit,SNqR\_uni,'b-'); % ve do thi SNqR tin hieu phan bo deu mo phong  hold on;  plot(nbit,SNqR\_sin,'r--'); % ve do thi SNqR tin hieu sin mo phong  plot(nbit,SNqR\_lt,'m-.'); % Ve do thi SNqR tin hieu phan bo deu ly thuyet  title('Do thi SN\_qR theo nbit');  xlabel('nbit');  ylabel('SN\_qR');  legend('x\_uni','x\_sin','ly thuyet');  grid on;  hold off; |



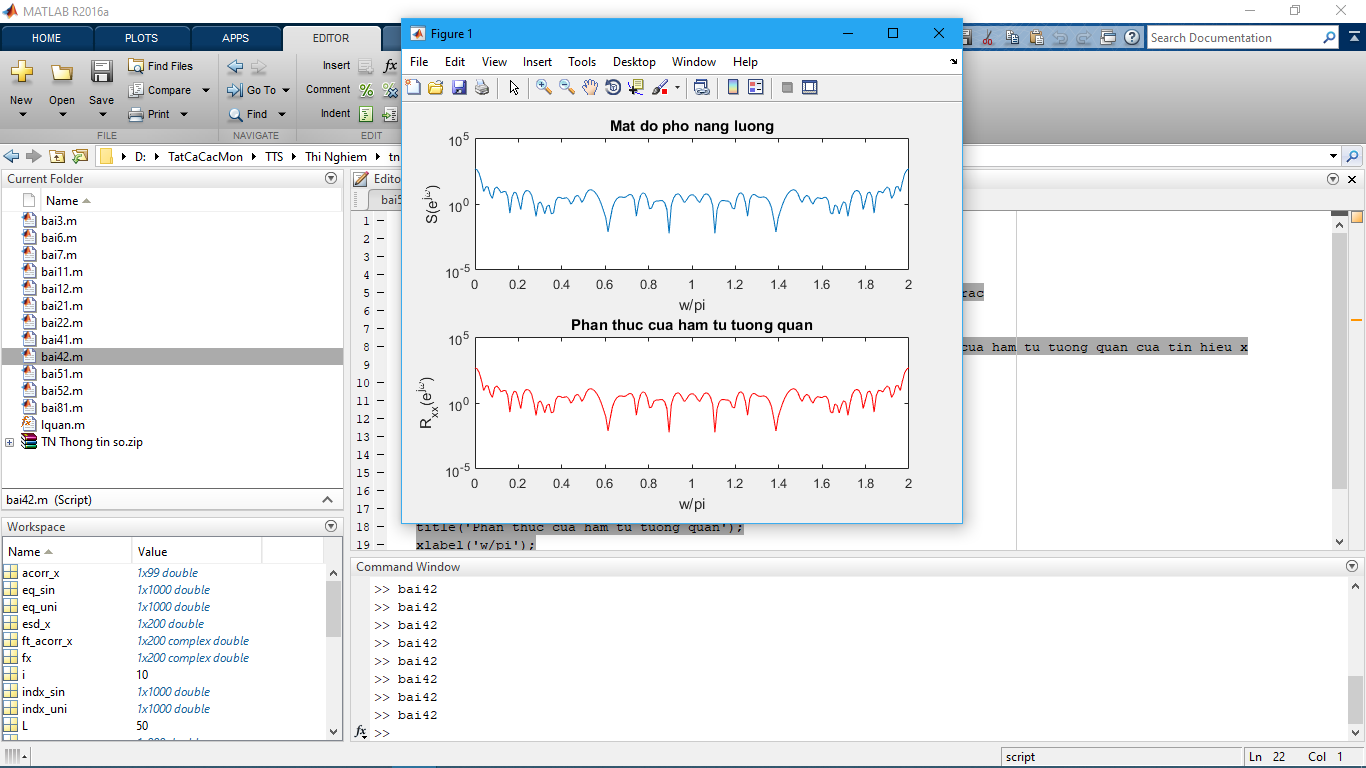
**Bài 4.1**

|  |
| --- |
| L = 500; % Chieu dai tin hieu  x = randn(1,L); % Tao tin hieu ngau nhien  acorr\_x = xcorr(x); % Tinh ham tu tuong quan  n = (-L+1):(L-1); % Cac mau gia tri  plot(n,acorr\_x); % Ve do thi ham tu tuong quan tin hieu ngau nhien  title('Ham tu tuong quan');  xlabel('n');  ylabel('r\_x\_x');  hold on;  x = linspace(-1,1,L); % Tao tin hieu co bien do tang dan  acorr\_x = xcorr(x); % Tinh ham tu tuong quan  plot(n,acorr\_x,'g--'); % Ve do thi ham tu tuong quan tin hieu co bien do tang dan  x = sin(linspace(-10,10,L)); % Tao tin hieu hinh sin  acorr\_x = xcorr(x); % Tinh ham tu tuong quan  plot(n,acorr\_x,'r-.'); % Ve do thi ham tu tuong quan tin hieu hinh sin  legend('Ngau nhien','Bien do tang dan','Hinh sin')  hold off; |



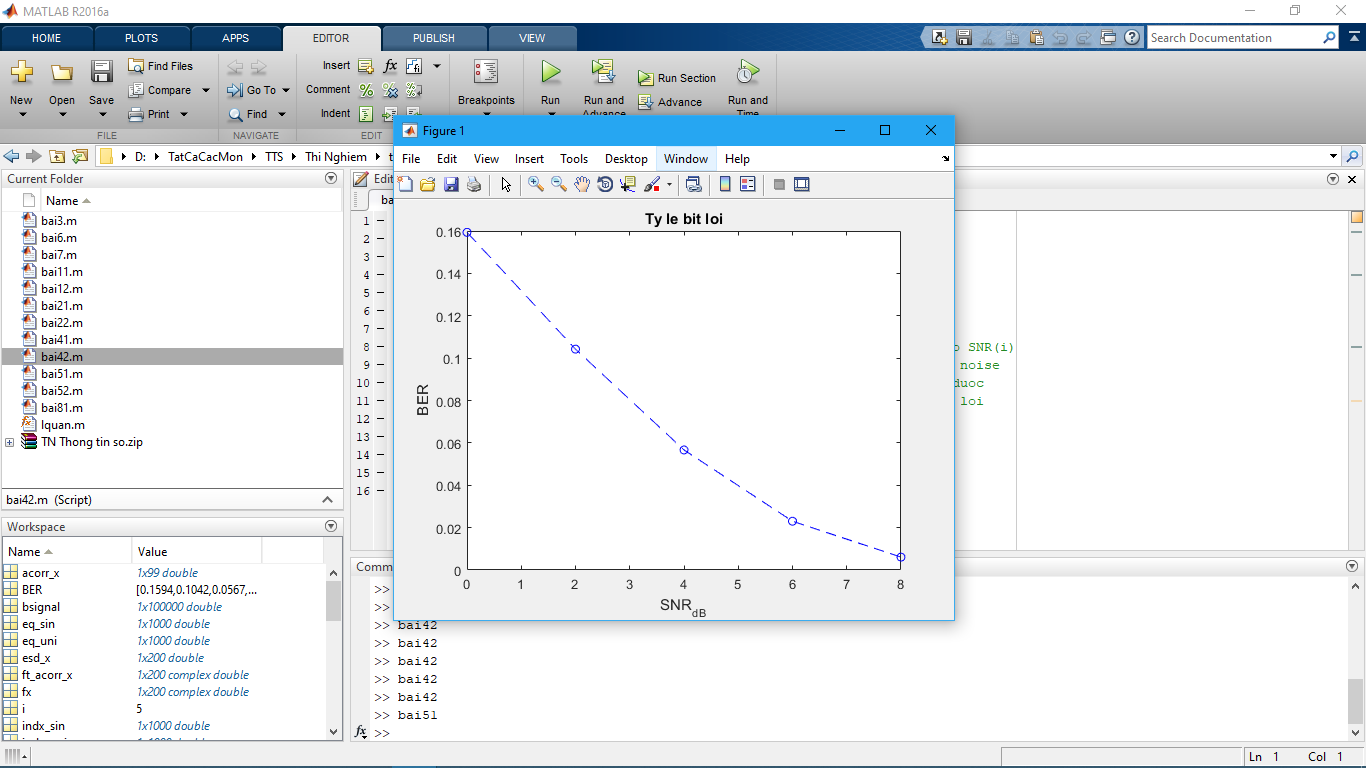
**Bài 4.2**

|  |
| --- |
| L = 50; % Do dai tin hieu  N = 200; % So luong cac tan so roi rac trong khoang 0 den 2\*pi  x = rand(1,L); % Tao tin hieu ngau nhien  w = linspace(0,2\*pi,N); % Tao N tan so tang dan tu 0 den 2\*pi  fx = freqz(x,1,w); % Bien doi Fourier cua x tai cac tan so roi rac  esd\_x = fx.\*conj(fx); % Tinh ham mat do pho nang luong  acorr\_x = xcorr(x); % Tinh ham tu tuong quan cua tin hieu x  ft\_acorr\_x = freqz(acorr\_x,1,w).\*exp(j\*w\*(L-1)); % Bien doi Fourier cua ham tu tuong quan cua tin hieu x  % Ve do thi  subplot(2,1,1);  semilogy(w/pi,esd\_x);  title('Mat do pho nang luong');  xlabel('w/pi');  ylabel('S(e^j^\omega)')  hold on;  subplot(2,1,2);  semilogy(w/pi,real(ft\_acorr\_x),'r');  title('Phan thuc cua ham tu tuong quan');  xlabel('w/pi');  ylabel('R\_x\_x(e^j^\omega)');  hold off; |



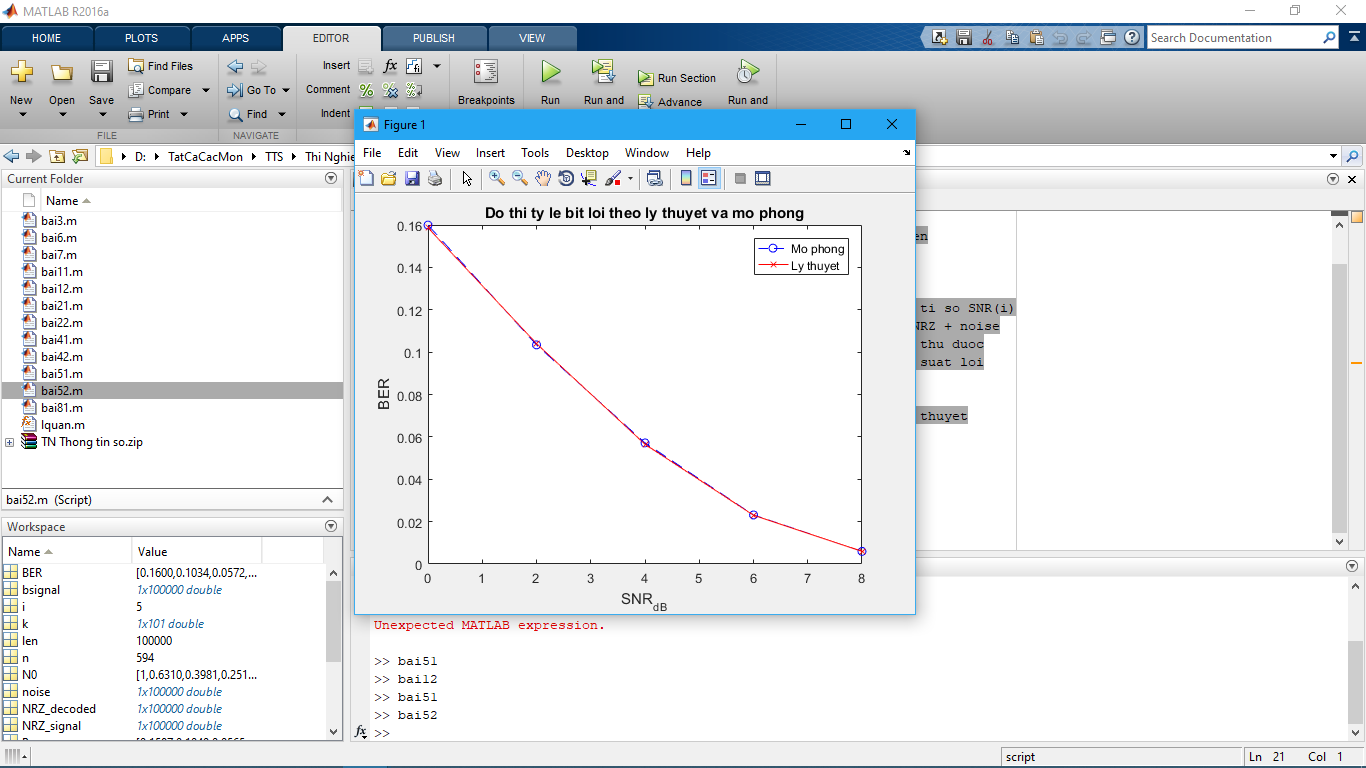
**Bài 5.1**

|  |
| --- |
| len = 100000; % Do dai dong bit mo phong  SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8  SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan  bsignal = randi([0 1],1,len); % Dong bit ngau nhiên do dai len  NRZ\_signal = bsignal\*2-1; % Bien doi dòng bit 0 1 sang -1 1  N0 = 1./SNR; % Cong suat tap am  for i=1:length(SNR\_db)  noise = sqrt(N0(i))\*randn(1,len); % Tao tap am noise voi ti so SNR(i)  r\_signal = NRZ\_signal + noise; % Tin hieu thu duoc = NRZ + noise  NRZ\_decoded = sign(r\_signal); % Giai ma tin hieu NRZ thu duoc  [n,BER(i)] = symerr(NRZ\_decoded,NRZ\_signal); % Tinh xac suat loi  end  plot(SNR\_db,BER,'bo--'); % Ve do thi  title('Ty le bit loi');  xlabel('SNR\_d\_B');  ylabel('BER'); |



**Bài 5.2**

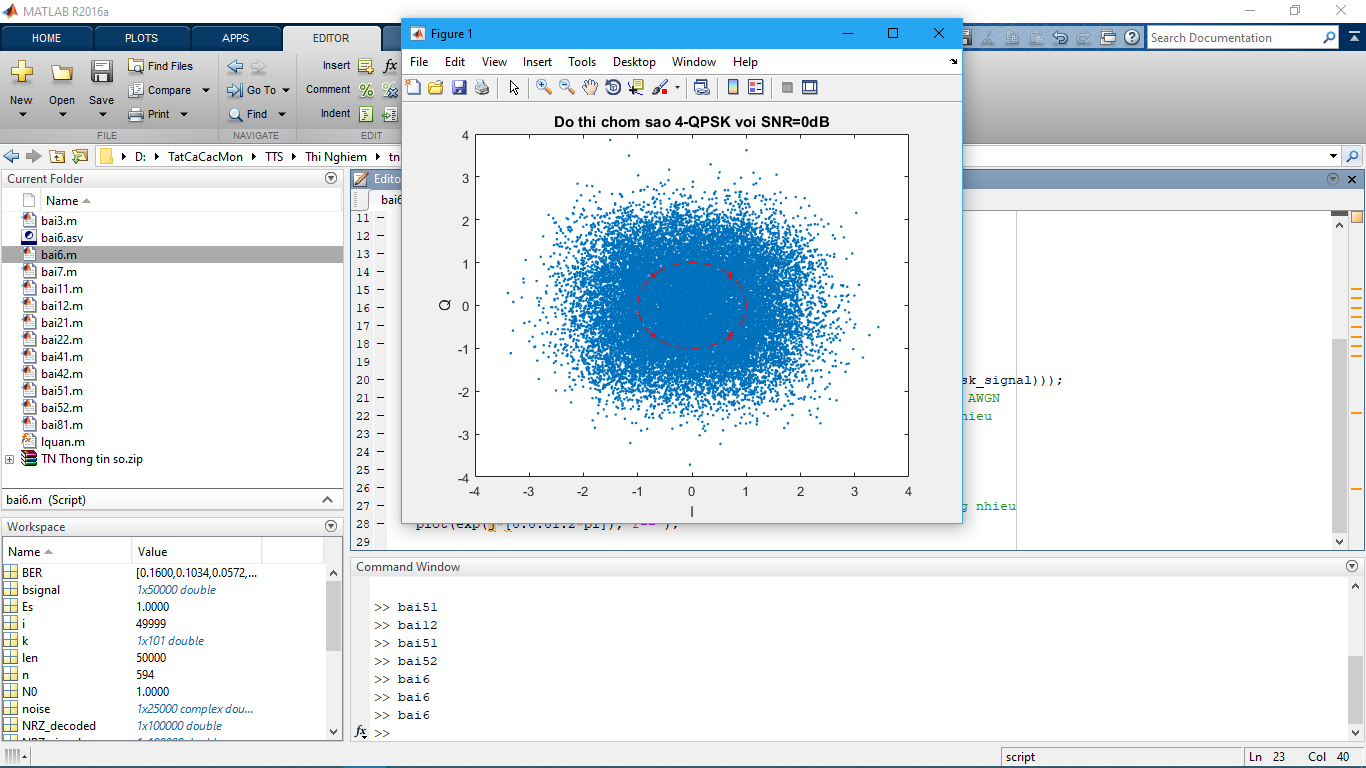
|  |
| --- |
| len = 100000; % Do dai dong bit mo phong  SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8  SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan  bsignal = randi([0 1],1,len); % Dong bit ngau nhiên do dai len  NRZ\_signal = bsignal\*2-1; % Bien doi dòng bit 0 1 sang -1 1  N0 = 1./SNR; % Cong suat tap am  for i=1:length(SNR\_db)  noise = sqrt(N0(i))\*randn(1,len); % Tao tap am noise voi ti so SNR(i)  r\_signal = NRZ\_signal + noise; % Tin hieu thu duoc = NRZ + noise  NRZ\_decoded = sign(r\_signal); % Giai ma tin hieu NRZ thu duoc  [n,BER(i)] = symerr(NRZ\_decoded,NRZ\_signal); % Tinh xac suat loi  end  plot(SNR\_db,BER,'bo--'); % Ve do thi BER  Pe = 1/2\*(1-erf(sqrt(SNR)/sqrt(2))); % Xac suat loi theo ly thuyet  hold on;  plot(SNR\_db,Pe,'rx-'); % Ve do thi Pe  title('Do thi ty le bit loi theo ly thuyet va mo phong');  xlabel('SNR\_d\_B');  ylabel('BER');  legend('Mo phong','Ly thuyet'); |



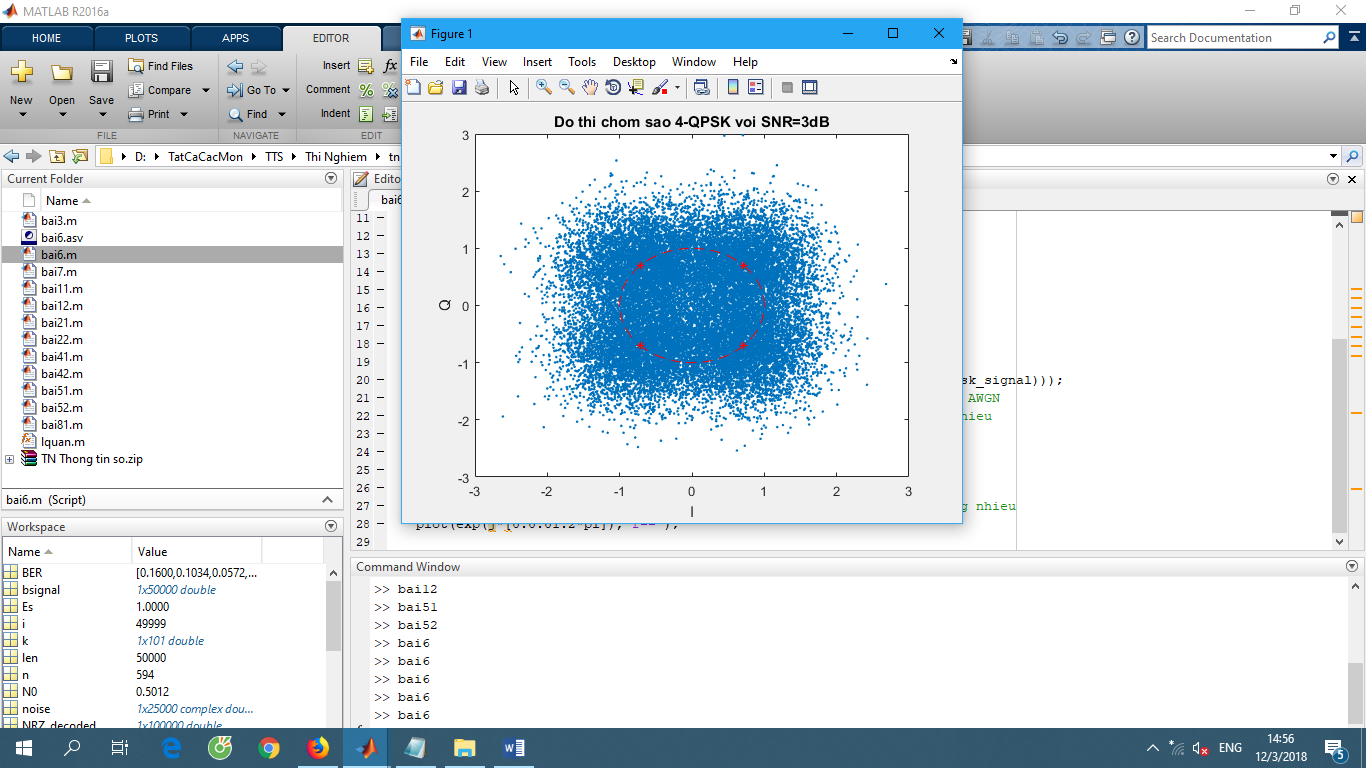
**Bài 6**

|  |
| --- |
| len = 50000; % Do dai dong bit mo phong  SNR\_db = 0; % SNR co don vi Decibel  SNR = 10^(SNR\_db/10); % Doi SNR tu Decibel sang lan  bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len  % Thuc hien dieu che QPSK  for i=1:2:len  if bsignal(i)==0 & bsignal(i+1)==0 % 00  qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);  elseif bsignal(i)==0 & bsignal(i+1)==1 % 01  qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);  elseif bsignal(i)==1 & bsignal(i+1)==1 % 11  qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);  elseif bsignal(i)==1 & bsignal(i+1)==0 % 10  qpsk\_signal((i+1)/2) = exp(j\*pi/4);  end  end  Es = std(qpsk\_signal)^2; % Nang luong ky hieu  N0 = Es/SNR; % Cong suat tap am  % Tao nhieu Gauss  noise = sqrt(N0/2)\*(randn(1,length(qpsk\_signal))+j\*randn(1,length(qpsk\_signal)));  qpsk\_awgn = qpsk\_signal + noise; % Cho tin hieu dieu che di qua kenh AWGN  plot(qpsk\_awgn,'.'); % Ve bieu do chom sao tin hieu co nhieu  title('Do thi chom sao 4-QPSK');  xlabel('I');  ylabel('QPSK');  hold on;  plot(qpsk\_signal,'r\*'); % Ve bieu do chom sao tin hieu khong nhieu  plot(exp(j\*[0:0.01:2\*pi]),'r--');  legend('QPSK co nhieu','QPSK khong nhieu'); |

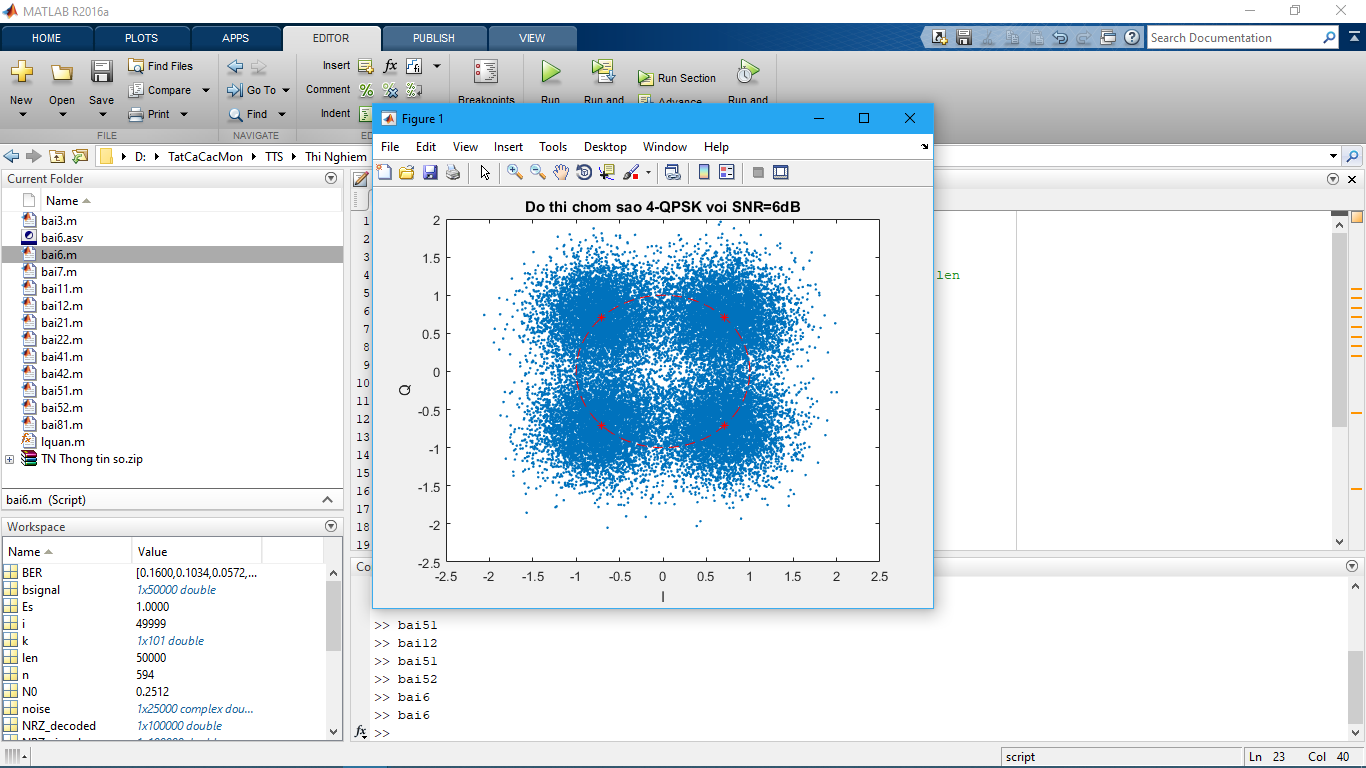
**Trường hợp SNR=0dB**



**Trường hợp SNR=3dB**

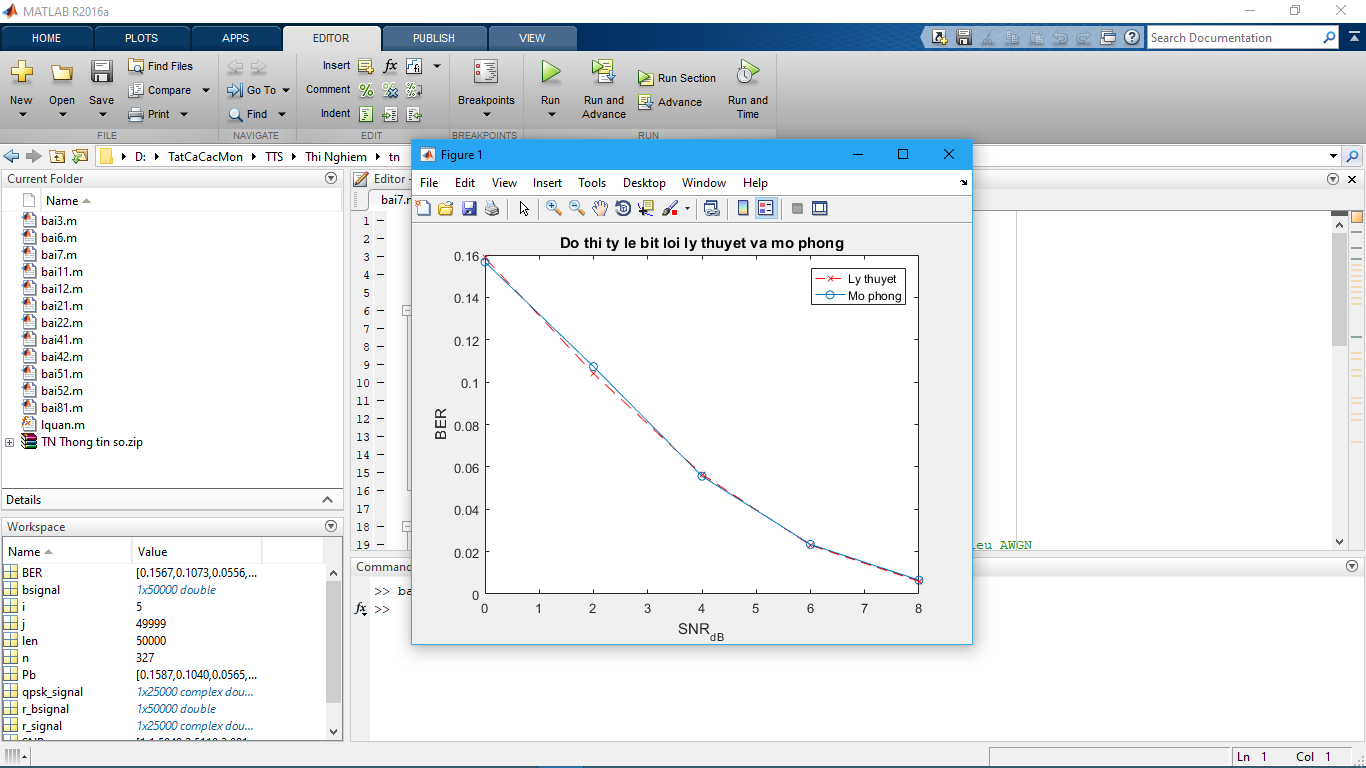


**Trường hợp SNR=6dB**



**Bài 7**

|  |
| --- |
| len = 50000; % Do dai dong bit mo phong  SNR\_db = 0:2:8; % Tao vector SNR\_db = 0 2 4 6 8  SNR = 10.^(SNR\_db/10); % Doi SNR tu Decibel sang lan  bsignal = randi([0 1],1,len); % Tao dong bit ngau nhien do dai len  % Thuc hien dieu che QPSK  for i=1:2:len  if bsignal(i)==0 & bsignal(i+1)==0 % 00  qpsk\_signal((i+1)/2) = exp(j\*3\*pi/4);  elseif bsignal(i)==0 & bsignal(i+1)==1 % 01  qpsk\_signal((i+1)/2) = exp(j\*5\*pi/4);  elseif bsignal(i)==1 & bsignal(i+1)==1 % 11  qpsk\_signal((i+1)/2) = exp(j\*7\*pi/4);  elseif bsignal(i)==1 & bsignal(i+1)==0 % 10  qpsk\_signal((i+1)/2) = exp(j\*pi/4);  end  end  % Tim BER mo phong  for i=1:length(SNR\_db)  r\_signal = awgn(qpsk\_signal,SNR\_db(i)); % Dieu che QPSK di qua nhieu AWGN  for j=1:2:len % Giai dieu che tin hieu QPSK co nhieu  if real(r\_signal((j+1)/2))>=0  if imag(r\_signal((j+1)/2))>=0 % Goc phan tu I  r\_bsignal(j) = 1;  r\_bsignal(j+1) = 0;  else % Goc phan tu IV  r\_bsignal(j) = 1;  r\_bsignal(j+1) = 1;  end  else  if imag(r\_signal((j+1)/2))>=0 % Goc phan tu II  r\_bsignal(j) = 0;  r\_bsignal(j+1) = 0;  else % Goc phan tu III  r\_bsignal(j) = 0;  r\_bsignal(j+1) = 1;  end  end  end  [n,BER(i)] = biterr(r\_bsignal,bsignal);  end  Pb = 1/2\*erfc(1/sqrt(2).\*sqrt(SNR)); % Xac suat loi bit  plot(SNR\_db,Pb,'rx--'); % Ve do thi Pb ly thuyet  title('Do thi ty le bit loi ly thuyet va mo phong');  xlabel('SNR\_d\_B');  ylabel('BER');  hold on;  plot(SNR\_db,BER,'o-');  legend('Ly thuyet','Mo phong');  hold off; |



**Bài 8**

|  |
| --- |
| n\_sym = 50000; % So ky tu dieu che  M = [16 32 64]; % So symbol ky hieu  SNR\_db = 0:25; % Tao vector SNR = 0 - 25 Decibel  BER = zeros(length(M),length(SNR\_db)); % BER de luu ti le loi bit  for k = 1:size(M,2) % size(M,2) la so cot cua M  s\_stream = randi([0 M(k)-1],1,n\_sym); % Tao dong bieu tuong do dai n\_sym  s\_mod = qammod(s\_stream,M(k),0,'GRAY'); % Dieu che M-QAM  for r = 1:size(SNR\_db,2) % Vong lap tinh BER  s\_mod\_awgn = awgn(s\_mod,SNR\_db(r),'measured'); % Tin hieu qua nhieu  s\_demod = qamdemod(s\_mod\_awgn,M(k),0,'GRAY'); % Giai dieu che M-QAM  [num ratio] = biterr(s\_stream,s\_demod); % Tinh ti le loi bit  BER(k,r) = ratio; % Luu ti le loi bit vao BER  end  end  semilogy(SNR\_db,BER(1,:),'bx-'); % Ve do thi BER ung voi M = 16  hold on;  semilogy(SNR\_db,BER(2,:),'rs-'); % Ve do thi BER ung voi M = 32  semilogy(SNR\_db,BER(3,:),'mo-'); % Ve do thi BER ung voi M = 64  grid on;  title('Do thi the hien ty le loi bit M-QAM');  xlabel('SNR\_d\_B');  ylabel('BER');  legend('16-QAM','32-QAM','64-QAM');  hold off; |

