

ماتریس مولد کدخطی سیستماتیک به صورت زیر است:

$G =$

0	0	1	1	1	0	1
0	1	0	0	1	1	1
1	0	0	1	1	1	0

از آنجا که فرم ماتریس به صورت وجود یک ماتریس همانی نمی باشد با انجام عملیات سطری (جا به جا کردن سطر ۱ و ۳) آن را به ماتریس زیر تبدیل می کنیم.

$G =$

1	0	0	1	1	1	0
0	1	0	0	1	1	1
0	0	1	1	1	0	1

ماتریس H که برابر است که یک ماتریس همانی به اندازه $n-k=4$ و ترانواده بقیه ماتریس. همچنین دستور `gen2par` نیز همین را خروجی می دهد.

$H =$

1	0	1	1	0	0	0
1	1	1	0	1	0	0
1	1	0	0	0	1	0
0	1	1	0	0	0	1

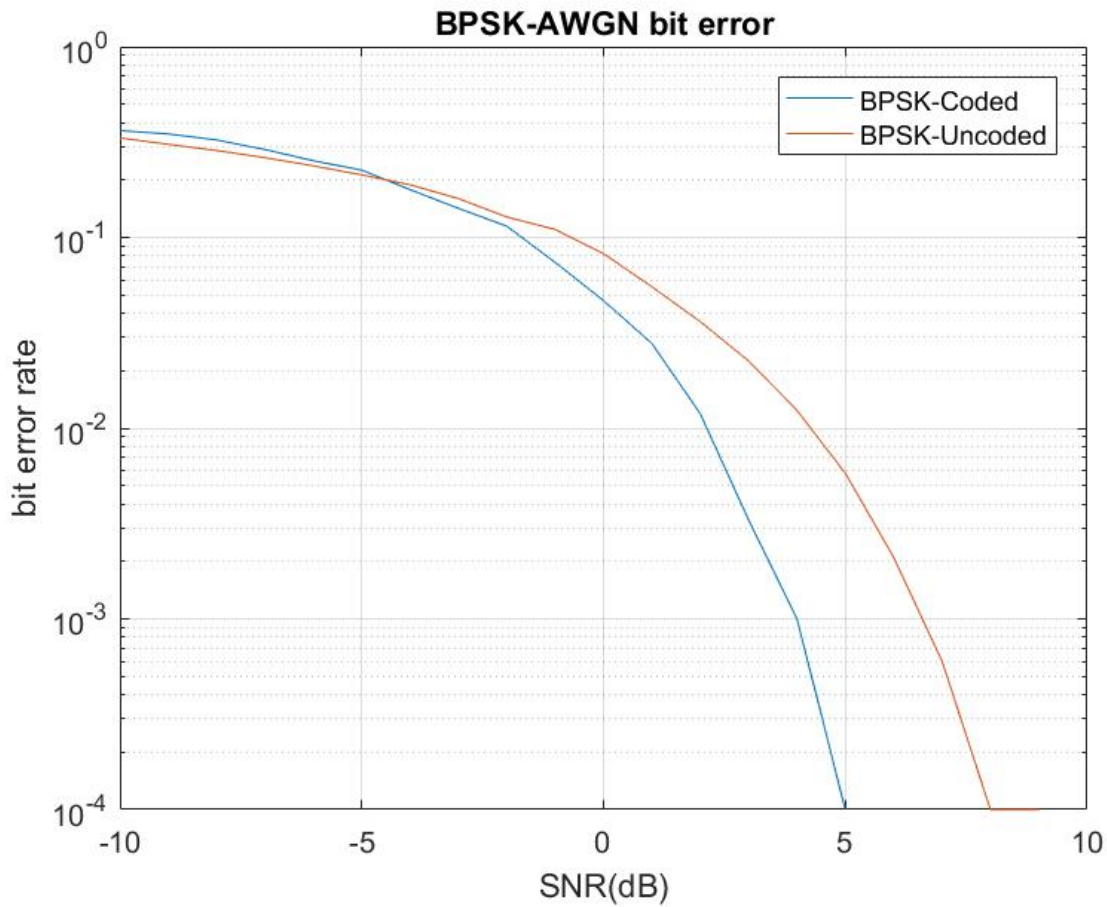
نیاز به جمع ۴ ستون از ماتریس H حداقل می باشد تا بردار تمام ۰ ساخته شود لذا می نیمم فاصله برابر ۴ است.

syndt, $V.H^T$

0	0	0	0	0	0	0	0000.
1	0	0	0	0	0	0	1110.
0	1	0	0	0	0	0	0111.
0	0	1	0	0	0	0	1101.
0	0	0	1	0	0	0	1000.
0	0	0	0	1	0	0	0100.
0	0	0	0	0	1	0	0010.
0	0	0	0	0	0	1	0001.
0	0	0	0	0	1	1	0011.
0	0	0	0	1	1	0	0110.
0	0	0	1	1	0	0	1100.
0	0	1	1	0	0	0	0101.
0	1	1	0	0	0	0	1010.
1	1	0	0	0	0	0	1001.
0	1	0	1	0	0	0	1111.
0	1	1	0	0	0	1	1011.

در مورد کد اول:

یک بردار ۹۹۹ تایی رندم تولید کردیم و بعد آن را کد و انکد کردیم با مقادیر مختلف SNR و از خود توابع متلب استفاده کردم و در عوض در تمرین دوم خودم ان‌هایی که نیاز بود را پیاده کردم.



در مورد کد دوم:

سندرم بدست آمده در قسمت بالا را در متغیری syndt سیو کردم که قسمت اول آن خود سندرم آن‌ها و قسمت دوم کاست لیدرها می باشد. همچنین کد وردهای معتبر که ۸ عدد می باشد را نیز در متغیر v ریختم. از آنجا که تعداد ۱۰۰۰ تا بیت داشتیم و قالب‌های ورودی ۳ تایی بودند از اضافه و کم کردن بین ۰ استفاده کردیم. همچنین آن کد وردهای پس از تصحیح که داخل مجموعه کد وردها قرار نگرفتند را هم به جاشون در کد عدد ۲ را گذاشتیم.

از یک جا به بعد کدر باعث بدتر شدن تصحیح خطا می شود .

کد اول :

```
G=[1,0,0,1,1,1,0;0,1,0,0,1,1,1;0,0,1,1,1,0,1]
H=gen2par(G)

syndt=syndtable(H)

data_uncoded= randi([0 1],1,9999);
data=encode(data_uncoded,7,3,'linear',G);

bpskmod_data= pskmod(data',2);
bpskmod_adata=pskmod(data_uncoded',2);

brate=[];
burate=[];

for snr=-10:1:20
    bch_data=awgn(bpskmod_data,snr);
    bch_adata=awgn(bpskmod_adata,snr);

    bpskdm_data=pskdemod(bch_data,2);
    bdata=decode(bpskdm_data',7,3,'linear',G);
    [n,r]=biterr(data_uncoded,bdata);
    brate=[brate r];

    bpskdm_adata=pskdemod(bch_adata,2);
    [n,r]=biterr(data_uncoded,bpskdm_adata');
    burate=[burate r];
end

semilogy([-10:1:20],brate,[-10:1:20],burate)
legend('BPSK-Coded','BPSK-Uncoded')
grid on
title('BPSK-AWGN bit error')
ylabel('bit error rate')
xlabel('SNR(dB)')
```

کد دوم:

```
clear

G=[1,0,0,1,1,1,0;0,1,0,0,1,1,1;0,0,1,1,1,0,1];

H=gen2par(G);
syndt=[
    1 1 1 0 1 0 0 0 0 0 0;
    0 1 1 1 0 1 0 0 0 0 0;
    1 1 0 1 0 0 1 0 0 0 0;
    1 0 0 0 0 0 0 1 0 0 0;
    0 1 0 0 0 0 0 0 1 0 0;
    0 0 1 0 0 0 0 0 0 1 0;
    0 0 0 1 0 0 0 0 0 0 1;
    0 0 1 1 0 0 0 0 0 1 1;
    0 1 1 0 0 0 0 0 1 1 0;
    1 1 0 0 0 0 0 1 1 0 0;
    0 1 0 1 0 0 1 1 0 0 0;
    1 0 1 0 0 1 1 0 0 0 0;
    1 0 0 1 1 1 0 0 1 0 0;
    1 1 1 1 0 1 0 1 0 0 0;
    1 0 1 1 0 1 1 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0
];

v=[
    0 0 0 0 0 0 0 0 0 0;
    0 0 1 0 0 1 1 1 0 1;
    0 1 0 0 1 0 0 1 1 1;
    0 1 1 0 1 1 1 0 1 0;
    1 0 0 1 0 0 1 1 1 0;
    1 0 1 1 0 1 0 0 1 1;
    1 1 0 1 1 0 1 0 0 1;
    1 1 1 1 1 1 0 1 0 0;
];
```

```

data_uncoded= randi([0 1],1,10000);

data_uncoded_reshape=reshape([data_uncoded 0
0],[],3);

codewords=mod(data_uncoded_reshape*G,2);
codewords_reshape_padded=[reshape(codewords,1,[])
zeros(1,2)];

data_coded_2=bin2dec(int2str(reshape(codewords_resh
ape_padded,[],1)));
data_coded_8=bin2dec(int2str(reshape(codewords_resh
ape_padded,[],3)));
data_coded_16=bin2dec(int2str(reshape(codewords_res
hape_padded,[],4)));

data_coded_bpsk=pskmod(data_coded_2,2);
data_coded_8psk=pskmod(data_coded_8,8);
data_coded_16psk=pskmod(data_coded_16,16);

data_uncoded_8=bin2dec(int2str(reshape([data_uncode
d zeros(1,8)],[],3)));
data_uncoded_16=bin2dec(int2str(reshape([data_uncod
ed zeros(1,8)],[],4)));

data_uncoded_bpsk=pskmod(data_uncoded,2);
data_uncoded_8psk=pskmod(data_uncoded_8,8);
data_uncoded_16psk=pskmod(data_uncoded_16,16);

rub=[];
ru8=[];
ru16=[];
rcb=[];
rc8=[];
rc16=[];
parfor snr_db=-10:1:20

data_coded_bpsk_AWGN_dmod=reshape(decimalToBinaryVe

```

```
ctor(pskdemod(awgn(data_coded_bpsk,snr_db*7/4),2)),  
1,[]);
```

```
data_coded_8psk_AWGN_dmod=reshape(decimalToBinaryVe  
ctor(pskdemod(awgn(data_coded_8psk,snr_db*7/4),8),3  
,1,[]);
```

```
data_coded_16psk_AWGN_dmod=reshape(decimalToBinaryV  
ector(pskdemod(awgn(data_coded_16psk,snr_db*7/4),16  
,4),1,[]);
```

```
data_uncoded_bpsk_AWGN_dmod=reshape(decimalToBinary  
Vector(pskdemod(awgn(data_uncoded_bpsk',snr_db),2))  
,1,[]);
```

```
data_uncoded_8psk_AWGN_dmod=reshape(decimalToBinary  
Vector(pskdemod(awgn(data_uncoded_8psk,snr_db),8),3  
,1,[]);
```

```
data_uncoded_16psk_AWGN_dmod=reshape(decimalToBinar  
yVector(pskdemod(awgn(data_uncoded_16psk,snr_db),16  
,4),1,[]);
```

```
data_coded_bpsk_reshape=reshape(data_coded_bpsk_AWG  
N_dmod(1:length(data_coded_bpsk_AWGN_dmod)-  
2),7,[]);
```

```
data_coded_8psk_reshape=reshape(data_coded_8psk_AWG  
N_dmod(1:length(data_coded_8psk_AWGN_dmod)-  
2),7,[]);
```

```
data_coded_16psk_reshape=reshape(data_coded_16psk_A  
WGN_dmod(1:length(data_coded_16psk_AWGN_dmod)-  
2),7,[]);
```

```
data_coded_bpsk_synd=mod(H*data_coded_bpsk_reshape,  
2);
```

```
data_coded_8psk_synd=mod(H*data_coded_8psk_reshape,  
2);
```

```
data_coded_16psk_synd=mod(H*data_coded_16psk_reshap  
e,2);
```

```
epaternb=[];  
epatern8=[];  
epatern16=[];
```

```
for i=1:length(data_coded_bpsk_synd(1,:))  
    temp1=data_coded_bpsk_synd(:,i);  
    temp2=data_coded_8psk_synd(:,i);  
    temp3=data_coded_16psk_synd(:,i);  
    for j=1:length(syndt(:,1))  
        if (temp1==syndt(j,1:4)')  
            epaternb=[epaternb  
syndt(j,5:end)'];  
        end  
        if (temp2==syndt(j,1:4)')  
            epatern8=[epatern8  
syndt(j,5:end)'];  
        end  
        if (temp3==syndt(j,1:4)')  
            epatern16=[epatern16  
syndt(j,5:end)'];  
        end  
    end  
end
```

```
data_coded_bpsk_noerr=mod(data_coded_bpsk_reshape+e  
paternb,2);
```



```
data_coded_8psk_noerr=mod(data_coded_8psk_reshape+e  
patern8,2);
```

```
data_coded_16psk_noerr=mod(data_coded_16psk_reshape  
+epatern16,2);
```

```
data_coded_bpsk_dcdr=[];
```

```
data_coded_8psk_dcdr=[];
```

```
data_coded_16psk_dcdr=[];
```

```
for i=1:length(data_coded_bpsk_noerr(1,:))
```

```
temp1=data_coded_bpsk_noerr(:,i);
```

```
temp2=data_coded_8psk_noerr(:,i);
```

```
temp3=data_coded_16psk_noerr(:,i);
```

```
c1=0;
```

```
c2=0;
```

```
c3=0;
```

```
for j=1:length(v(:,1))
```

```
if (temp1==v(j,4:end)')
```

```
data_coded_bpsk_dcdr=[data_coded_bpsk_dcdr  
v(j,1:3)'];
```

```
else
```

```
c1=c1+1;
```

```
if (c1==(length(v(:,1))))
```

```
data_coded_bpsk_dcdr=[data_coded_bpsk_dcdr  
2*ones(3,1)];
```

```
end
```

```
end
```

```
if (temp2==v(j,4:end)')
```

```
data_coded_8psk_dcdr=[data_coded_8psk_dcdr  
v(j,1:3)'];
```

```
else
```

```
c2=c2+1;
```

```
if (c2==(length(v(:,1))))
```

```
data_coded_8psk_dcdr=[data_coded_8psk_dcdr  
2*ones(3,1)];
```

```
end
```

```
end
```

```
if (temp3==v(j,4:end)')
```

```
data_coded_16psk_dcdr=[data_coded_16psk_dcdr  
v(j,1:3)'];
```

```
else
```

```
c3=c3+1;
```

```
if (c3==(length(v(:,1))))
```

```
data_coded_16psk_dcdr=[data_coded_16psk_dcdr  
2*ones(3,1)];
```

```
end
```

```
end
```

```
end
```

```
end
```

```
data_coded_bpsk_dcdr_rshape=reshape(data_coded_bpsk  
_dcdr,1,[]);
```

```
data_coded_8psk_dcdr_rshape=reshape(data_coded_8psk  
_dcdr,1,[]);
```

```
data_coded_16psk_dcdr_rshape=reshape(data_coded_16p  
sk_dcdr,1,[]);
```

```
data_coded_bpsk_dcdr_rshape=data_coded_bpsk_dcdr_rs  
hape(1:(length(data_coded_bpsk_dcdr_rshape)-2));
```

```
data_coded_8psk_dcdr_rshape=data_coded_8psk_dcdr_rs  
hape(1:(length(data_coded_8psk_dcdr_rshape)-2));
```

```
data_coded_16psk_dcdr_rshape=data_coded_16psk_dcdr_  
rshape(1:(length(data_coded_16psk_dcdr_rshape)-2));
```

```

data_uncoded_bpsk_rec=data_uncoded_bpsk_AWGN_dmod;

data_uncoded_8psk_rec=data_uncoded_8psk_AWGN_dmod(1
:length(data_uncoded_8psk_AWGN_dmod)-8);

data_uncoded_16psk_rec=data_uncoded_16psk_AWGN_dmod
(1:length(data_uncoded_16psk_AWGN_dmod)-8);

[ntemp,rtemp]=biterr(data_coded_bpsk_dcdr_rshape,da
ta_uncoded);
    rcb=[rcb rtemp];

[ntemp,rtemp]=biterr(data_coded_8psk_dcdr_rshape,da
ta_uncoded);
    rc8=[rc8 rtemp];

[ntemp,rtemp]=biterr(data_coded_16psk_dcdr_rshape,d
ata_uncoded);
    rc16=[rc16 rtemp];

[ntemp,rtemp]=biterr(data_uncoded_bpsk_rec,data_unc
oded);
    rub=[rub rtemp];

[ntemp,rtemp]=biterr(data_uncoded_8psk_rec,data_unc
oded);
    ru8=[ru8 rtemp];

[ntemp,rtemp]=biterr(data_uncoded_16psk_rec,data_un
coded);
    ru16=[ru16 rtemp];

end
snr_db=-10:1:20;
figure

```

```
semilogy(snr_db, rub, snr_db, rcb)
grid on
title('BPSK')
legend('Uncoded', 'Coded')
ylabel('bit error rate')
xlabel('SNR(dB)')
figure
semilogy(snr_db, ru8, snr_db, rc8)
grid on
title('8PSK')
legend('Uncoded', 'Coded')
ylabel('bit error rate')
xlabel('SNR(dB)')
figure
semilogy(snr_db, ru16, snr_db, rc16)
grid on
title('16PSK')
legend('Uncoded', 'Coded')
ylabel('bit error rate')
xlabel('SNR(dB)')
```

کد golay و extend آن دیکدر و انکدر را نیز به پیوست ارسال می کنم.