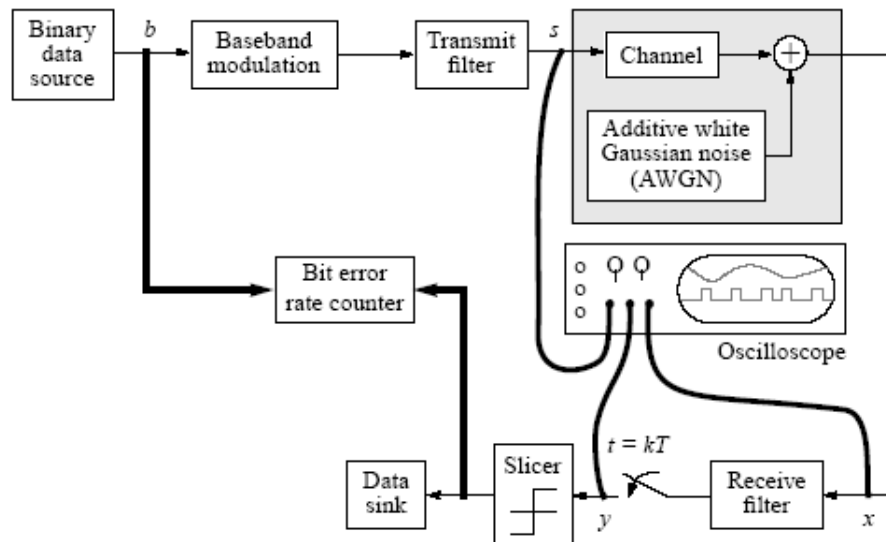


Percobaan 2

DETEKSI



Gambar 1. Diagram konseptual

TUJUAN

Percobaan ini mempelajari proses deteksi sinyal dengan mengamati tiga bagian penting dari sebuah penerima: filter penerima, *sampler* dan *slicer*. Pada khususnya, anda akan:

- mempelajari karakteristik dari filter penerima optimal, yaitu *matched filter*
- menggunakan pola mata (*eye pattern*) untuk menentukan waktu penyampelan optimal dan batas ambang (*decision threshold*) yang optimal;
- mempelajari unjuk kerja dari penerima dengan menggunakan filter penerima yang bukan optimal;
- mempelajari kekokohan (*robustness*) dari sebuah sistem komunikasi terhadap ISI dengan menggunakan filter pemancar dan penerima.

REFERENSI

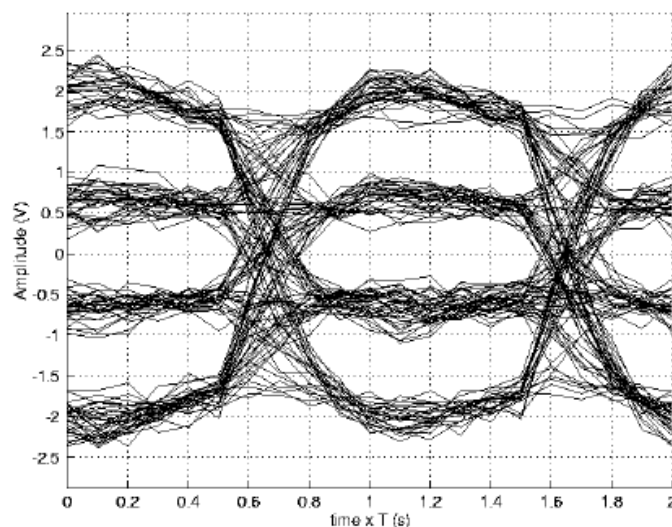
- [1] S. Haykin, *Communication Systems*, 4th ed., John Wiley, 2001, pp. 248-253.
- [2] B. Sklar, *Digital Communications*, 2nd ed., Prentice Hall, 1988.

TUGAS PRA-PERCOBAAN:

1. Misalkan sebuah pulsa persegi dengan amplitudo satu dan lebar $T = 10$ mdtk:

$$p(t) = \text{rect}((t - T/2)/T)$$

2. Misal $x(t) = s(t) + n(t)$ menyatakan gelombang dari output sebuah kanal berderau dan tanpa distorsi. $s(t)$ adalah sebuah gelombang NRZ bipolar dengan amplitudo pulsa satu dan laju data biner F_d sebesar 1 kb/dtk. Derau mempunyai PSD $S_n(f) = N_o/2 = 1.0 \times 10^{-4}$ W/Hz. Jika $x(t)$ dimasukkan sebagai input dari penerima *matched filter*, maka:
 - a. Tentukan daya derau rata-rata σ_n^2 pada output dari *matched filter*;
 - b. Tentukan amplitudo puncak dari komponen yang mengandung dara dari sinyal output dari *matched filter*;
 - c. Tentukan energi rata-rata E_b dari $s(t)$ dalam satu periode bit;
 - d. Hitung probabilitas kesalahan bit $P_e = 0.5 \cdot \text{erfc}((E_b/N_o)^{1/2})$.
3. Sebuah sistem NRZ 4-level dengan laju data 500 simbol per detik menggunakan 4 amplitudo untuk mengirim simbol-simbol 2 bit: $\{-3A, -A, A, 3A\}$ volt. Pada output dari penerima *matched filter* sebuah pola mata 4-level diamati seperti pada Gambar 2.
 - a. Jika semua 4 simbol mempunyai probabilitas sama, tentukan energi rata-rata tiap simbol dan energi rata-rata tiap bit sebagai fungsi A .
 - b. Misalkan sebuah sistem NRZ bipolar (biner) dengan amplitudo satu dan laju 1000 bit per detik. Tentukan E_b dan bandingkan dengan energi rata-rata tiap bit pada bagian a).
 - c. Untuk sistem 4-level, berapakah nilai yang dibutuhkan untuk mendapatkan margin terhadap derau yang sama dengan sistem biner pada bagian b), pada semua tiga bukaan mata pada output *matched filter*?
 - d. Tradeoff apakah yang dijumpai jika kita berpindah dari sistem biner ke sistem 4-level?



Gambar 2. Pola mata

LANGKAH-LANGKAH PERCOBAAN:

Catatan: percobaan ini menggunakan perintah khusus `bbchannel` yang banyak digunakan untuk urutan data yang panjang.

1. Matched filter sebagai penerima optimal

Pada percobaan ini kita memperkenalkan perintah Matlab khusus `matched`

```
impulse_resp = matched(<pulse_shape>, <sampling_rate>, <Fs>);
```

yang membangkitkan matched filter yang berkaitan dengan `pulse_shape`, dan di mana `Fs` adalah frekuensi sampling simulasi yang lebih besar daripada `sampling_rate`.

1.1 Mulai dengan membangkitkan pulsa tunggal dengan amplitudo satu dan durasi 1 mdetik:

```
>>global Fd Fs;  
>>Fd = 1000; Fs = 100000; T = 1/Fd;  
>>s = modul(1, 'unipolar_nrz');
```

Sekarang, hasilkan tanggapan impuls dari filter yang sesuai (*matched*) dengan pulsa ini:

```
>>mf = matched('unipolar_nrz');
```

Plot pulsa asli dan filter yang sesuai dengannya:

```
>>subplot(311), scope(s, 'one-shot', 2*T);  
>>subplot(312), scope(mf, 'one-shot', 2*T);
```

Sekarang plot output dari matched filter jika `s` digunakan sebagai input:

```
>>subplot(313), scope(afilter(s,mf), 'one-shot', 2*T);
```

Pada waktu berapa output dari filter mencapai nilai puncak?

```
t = ..... [mdtk]
```

1.2 Sekarang anda akan mengulangi langkah-langkah di atas untuk pulsa Manchester dengan lebar 1 mdtk dan amplitudo satu:

```
>>s = modul(1, 'manchester');  
>>mf = matched('manchester');
```

```
>>subplot(311), scope(s, 'one-shot', 2*T);
>>subplot(312), scope(mf, 'one-shot', 2*T);
>>subplot(313), scope(afilter(s,mf), 'one-shot', 2*T);
```

Dari pengamatan plot-plot di atas, pada waktu berapakah seharusnya gelombang disampel?

$t = \dots\dots [\text{mdtk}]$

Berapakah energi bit E_b dari pulsa Manchester dengan amplitudo satu?

$E_b = \dots\dots [\text{J}]$

1.3 Buat gelombang RZ bipolar yang menyatakan urutan biner {1,0,0,1,0,1,1} dengan menggunakan amplitudo satu dan laju data biner 1 kb/s:

```
>>s = modul([1 0 0 1 0 1 1], 'bipolar_rz');
>>clf, subplot(211), scope(s, 'one-shot', 7*T);
```

Lewatkan s melalui matched filter dan plot sinyal output yang diperoleh:

```
>>y = afilter(s, matched('bipolar_rz'));
>>subplot(212), scope(y, 'one_shot', 7*T);
```

Catat amplitudo maksimum dari dua sinyal tersebut. Gunakan fungsi Matlab `plotyy` to menggambar dua sinyal pada gambar yang sama tetapi dengan skala vertikal yang berbeda. Pertama, buat vektor terskala yang sesuai (waktu dalam detik) untuk absis

```
>>clf
>>t = 0:1/Fs:(7*T-1/Fs);
>>plotyy(t, s, t, y(1:length(t)));
>>title('Sinyal RZ bipolar dan output matched filter');
```

Tambahkan label yang sesuai (lihat perintah-perintah `xlabel` dan `ylabel`)

Cetak gambar tersebut.

2. Deteksi sinyal

2.1 Bangkitkan uruan biner acak 10-bit dan modulasikan ke gelombang NRZ.

```
>>b = round(rand(1, 10));
>>s = modul(b, 'bipolar_nrz');
```

Sekarang kirimkan s melalui kanal baseband dengan bandwidth 4.9 kHz dan sebuah derau aditif dengan PSD $N_0/2 = 2 \times 10^{-4} \text{ W/Hz}$. Tampilkan output x :

```
>>x = bbchannel(s, 1, 2e-4, 4900);
>>clf
>>subplot(312), scope(x, 'one-shot', 10*T);
```

Cobalah mendekodekan secara visual urutan b yang dikirimkan dengan hanya mengamati output x :

b =

Sekarang olah output x dengan sebuah matched filter. Tampilkan sinyal y yang diperoleh:

```
>>y = afilter(x, matched('bipolar_nrz'));
>>subplot(313), scope(y, 'one-shot', 10*T);
```

Dengan mengingat bahwa $T = 1$ mdtk adalah durasi bit, secara visual sampellah output y pada waktu-waktu kT di mana $k = 1, \dots, 10$ dan terapkan aturan keputusan (*decision rule*):

$$\hat{b}_k = \begin{cases} 0, & y(kT) \leq 0 \\ 1, & y(kT) > 0 \end{cases}$$

Anda akan memperoleh perkiraan \hat{b} dari urutan biner yang asli b :

$\hat{b} =$

Sekarang tampilkan gelombang asli s dan bandingkan urutan biner yang asli b dengan perkiraan anda:

```
>>subplot(311), scope(s, 'one-shot', 10*T);
```

b =

2.2 Ubahlah frekuensi sampling simulasi ke $F_s = 10000$, bangkitkan urutan biner 10000-bit dan modulasikan dengan menggunakan NRZ bipolar:

```
>>Fs = 10000;
>>b = round(rand(1,10000));
>>s = modul(b, 'bipolar_nrz');
```

Kirimkan s melalui kanal dengan bandwidth 4.9 kHz dan derau Gaussian putih aditif (AWGN) dengan PSD $N_0/2 = 0.0 \times 10^{-4}$ W/Hz.

```
>>x = bbchannel(s, 1, 0.0e-4, 4900);
```

Olah x dengan menggunakan matched filter. Plot pola mata dari output matched filter y untuk 300 bit pertama:

```
>>y = afilter(x, matched('bipolar_nrz'));  
>>clf, scope(y(1:300*Fs/Fd), 'continuous', 2*T);
```

Dengan mengamati pola mata, tentukan waktu sampling optimal t_{opt} dan ambang keputusan optimal λ . Ingat bahwa nilai pada sumbu waktu dari pola mata yang dihasilkan oleh perintah `scope` harus dikalikan dengan periode simbol T . Sebagai contoh dalam hal ini, jika waktu sampling terbaca 0.5, pada kenyataannya adalah 0.5×10^{-3} .

$t_{\text{opt}} = \text{waktu sampling} = \dots \text{ [dtk]}$

$\lambda = \text{ambang} = \dots \text{ [V]}$

Perhatikan bahwa t_{opt} yang ditentukan dari pola mata tidak berkaitan dengan nilai yang diperoleh dari Bagian 1 dari percobaan karena hal-hal yang berada di dalam modul khusus Matlab.

Anda akan menggunakan nilai-nilai ini sebagai parameter pada detektor yang mengolah output matched filter. Jika waktu sampling dan nilai ambang λ telah dipilih dengan benar, maka ia akan menghasilkan unjuk kerja detektor yang optimal dan karena itu juga probabilitas kesalahan minimum. Kita menggunakan perintah Matlab khusus `detect` agar diperoleh probabilitas kesalahan bit (BER):

`ber = detect(<mf_output>, <threshold>, <sampling_time>, <tx_data>);`

di mana `<mf_output>` adalah sinyal yang diterima setelah matched filter dan `<tx_data>` adalah data urutan biner yang dikirim.

```
>>detect(y, <threshold>, <sampling_time>, b)
```

Ulangi langkah percobaan untuk mengirimkan s melalui kanal dan melewatkan outputnya ke sebuah matched filter yang diikuti sebuah detektor dengan mengubah-ubah PSD derau dari 0.5, 1, 1.5 dan 2×10^{-4} W/Hz. Karena pada Percobaan 1 anda sudah mendapatkan bahwa nilai ambang dan waktu sampling optimal tidak tergantung pada daya derau, anda dapat menggunakan parameter-parameter tersebut untuk semua harga PSD derau. Catatlah probabilitas kesalahan bit P_e yang diperoleh pada Tabel 1, dan amati hubungan antara diagram mata dan probabilitas kesalahan bit.

Tabel 1: BER vs daya derau

$N_0/2$	P_e terukur	P_e teoritis
0.5		
1.0		
1.5		
2.0		

Untuk mempercepat proses yang berulang ini, anda dapat menggunakan perintah Matlab khusus `transmit` untuk memodulasi, mengirim, dan mengolah dengan matched filter sinyal tersebut. Perintah ini juga menampilkan diagram mata.

```
y = transmit(<tx_data>, <line_code>, <noise_PSD>, <bandwidth>);
```

```
>>y = transmit(b, 'bipolar_nrz', <noise_PSD>, 4900);
>>ber = detect(y, <threshold>, <sampling_time>, b);
```

- 2.3 Untuk mengamati pengaruh dari waktu sampling yang tidak optimal, anda akan membuat plot P_e versus waktu sampling. Gunakan PSD derau kanal sebesar 0.5×10^{-4} W/Hz untuk mendapatkan x.

Catatan: Matlab tidak akan menampilkan prompt `>>` jika anda berada dalam bagian dalam `for` loop. Selain itu, pada akhir dari baris yang berisi pernyataan `for` tidak perlu ditutup dengan tanda titik-koma (`;`).

```
>>y = transmit(b, 'bipolar_nrz', 0.5e-4, 4900);
>>st = 0:1/Fs:1/Fd;
>>pe = zeros(1,length(st));
>>for i = 1:length(st)
    pe(i) = detect(y, <threshold>, st(i), b);
end;
>>plot(st, pe);
>>xlabel('waktu sampling');
>>ylabel('Bit error rate (BER)');
>>title('BER vs waktu sampling');
```

Cetaklah grafik ini.

- 2.4 Sekarang kita akan mengamati pengaruh dari bandwidth kanal terhadap BER. Ikuti langkah percobaan yang sama pada Bagian 2.2, dengan mengirim pada kanal dengan PSD derau sebesar 0.5×10^{-4} W/Hz dan bandwidth 1500, 1000, 500 dan 350 Hz. Catatlah laju kesalahan bit yang diperoleh pada Tabel 2.

Tabel 2: BER vs bandwidth kanal

bandwidth	P_e terukur
1500 Hz	
1000 Hz	
500 Hz	
350 Hz	

Menentukan waktu sampling yang tepat untuk bandwidth kecil (misal 500 Hz dan 350 Hz) adalah sulit.

- Jika pola mata yang diperoleh "tertutup", anda dapat menentukan t_{opt} dengan menggambarkan pola mata dengan bagian yang lebih pendek dari sampel sinyal y , misalnya:

```
>>clf, scope(y(1:1500), 'continuous', 2*T);
```

Meskipun anda masih memperoleh mata yang "tertutup", tetapi akan lebih mudah bagi anda untuk mengidentifikasi secara visual letak dari perlintasan-perlintasan nol (*zero crossing*) dan menentukan t_{opt} di antara keduanya. Jika hal tersebut masih sulit dilakukan, anda dapat menentukan t_{opt} secara coba-coba dengan mencari BER minimum yang diperoleh. Mulailah dengan nilai 0 mdtk kemudian perbesar pelan-pelan dengan 0.1 mdtk sampai BER yang diperoleh tidak turun lagi. Karena waktu sampling dan ambang tidak bergantung pada daya derau, anda dapat mencoba untuk sementara memberi harga PSD derau pada 0 W/Hz untuk memperoleh hasil yang lebih jelas dari pola mata.

- Jika BER yang diperoleh mendekati 0.5, sisipkan delay sebesar satu simbol sebagai parameter terakhir pada perintah `detect`:

```
>>detect(y,<threshold>,<sampling_time>,b,1)
```

Sebagai contoh, jika nilai `<sampling_time>` adalah 0.0001, maka sintaks di atas akan menghasilkan waktu sampling efektif 0.0011 (karena periode simbol adalah 0.001).

3. Unjuk kerja dari penerima optimal terhadap ISI

Pada Percobaan 1, telah ditunjukkan bahwa penggunaan filter pembentuk pulsa ideal akan memberikan perbaikan yang cukup besar jika dilakukan pengiriman melalui kanal yang bisa menyebabkan ISI. Kanal yang dipelajari saat itu mempunyai bandwidth keseluruhan yang cukup tetapi mempunyai lekukan (*notch*) yang besar di tengah yang menyebabkan ISI. Di sini kita akan melihat sebuah kanal yang terlalu sempit, yang juga merupakan sumber dari penyebaran waktu (*time spreading*) dan oleh karena itu juga ISI.

Kita akan melihat unjuk kerja berbagai pasangan filter kirim dan filter terima pada kondisi tersebut ditinjau dari laju kesalahan bit sistem.

- 3.1 Modulasikan urutan biner b dengan menggunakan NRZ bipolar dan kirimkan melalui kanal tanpa derau dengan bandwidth 550 Hz. Olah sinyal yang diterima dengan matched filter NRZ bipolar. Tampilkan pola mata dan tentukan sampling time dan ambang yang optimal. Jangan lupa untuk menyisipkan delay satu simbol jika BER mendekati 0.5, selain juga usulan pada Bagian 2.4 berkaitan dengan pola mata yang "tertutup"

```
>>y = transmit(b, 'bipolar_nrz', 0, 550);  
>>detect(y, <threshold>, <sampling_time>, b, <delay>)
```

Ulangi untuk bandwidth 350 Hz dan catatlah probabilitas kesalahan bit-nya.

Bipolar NRZ, $W = 350 \text{ Hz} \Rightarrow P_e = \dots$

- 3.2 Sekarang kita akan melakukan pengiriman melalui kanal ini dengan menggunakan pulsa-pulsa *raised cosine*.

Bentuk pulsa optimal untuk mengatasi ISI adalah pulsa Nyquist yang kita pelajari di Percobaan 1. Akan tetapi, pulsa ini mempunyai dua kesulitan untuk diimplementasikan secara praktis. Pulsa tersebut tidak kausal dan oleh karena itu secara praktis membutuhkan penundaan (delay) tertentu dan juga laju konvergensinya ke nol cukup lambat. Ekor dari pulsa mengecil dengan laju $1/t$ yang berarti bahwa kesalahan kecil saja pada waktu penyampelan terhadap output dari penerima matched filter akan berakibat deretan yang panjang dari komponen ISI.

Keluarga dari pulsa-pulsa raised-cosine (nama tersebut sesuai dengan bentuk spektrumnya) juga memenuhi kriteria Nyquist tetapi lebih memungkinkan untuk direalisasi dan laju pengecilan ekornya cukup cepat. Sebaliknya konsekuensi yang harus dibayar adalah kelebihan bandwidth yang bisa diatur oleh faktor *roll-off* β .

Modulasikan urutan yang sama dengan menggunakan pulsa yang dibentuk oleh akar kuadrat (*square root*) dari filter *raised-cosine* dengan faktor *roll-off* $\beta = 0.2$. Kirimkan sinyal yang diperoleh melalui kanal tanpa derau dengan bandwidth 550 Hz, olahlah dengan matched filter yang sesuai, dan amati diagram matanya:

```
>>y = transmit(b, 'raised cosine', 0, 550, <roll_off>);
```

Tentukan ambang keputusan dan waktu sampling yang benar, dan tentukan probabilitas kesalahan bit-nya. Karena sifat ketidakkausalnya dan panjang dari pulsa raised cosine, akan terdapat delay transmisi tambahan, yang ditentukan sebesar 20 simbol dalam simulasi Matlab ini. Oleh karenanya, delay sebesar 20 simbol harus juga diberikan pada detektor:

```
>>ber = detect(y,<threshold>,<sampling_time>,b,20)
```

Seperti sebelumnya, anda harus memasukkan delay tambahan 1 simbol (cukup mengganti 20 dengan 21).

Ulangi percobaan ini untuk kanal-kanal dengan bandwidth 450, 400, dan 350 Hz.

Raised cosine akar kuadrat, $\beta = 0.2$, $W = 550$ Hz $\Rightarrow P_e = \dots$
 $W = 450$ Hz $\Rightarrow P_e = \dots$
 $W = 400$ Hz $\Rightarrow P_e = \dots$
 $W = 350$ Hz $\Rightarrow P_e = \dots$

3.3 Untuk kanal dengan bandwidth 350 Hz, catat hasil dari Bagian 3.1 dan 3.2 pada kotak yang tersedia di Tabel 3. Ulangi langkah-langkah yang anda lakukan pada Bagian 3.2 untuk semua faktor roll-off pada tabel. Mengubah factor *roll-off* tidak akan mengubah waktu sampling, sehingga anda tidak perlu menampilkan diagram mata lagi.

Ulangi untuk bandwidth 400 Hz. Ingat bahwa mengubah bandiwdth kanal mungkin perlu menentukan t_{opt} lagi.

Tabel 3: BER versus faktor roll-off filter raised cosine

Bentuk pulsa		P_e pada 350 Hz	P_e pada 400 Hz
bipolar NRZ			
raised cosine	$\beta = 0$		
	$\beta = 0.2$		
	$\beta = 0.5$		
	$\beta = 1.0$		

4. Pengkodean correlative-level (duobinary)

Untuk memenuhi kriteria *zero*-ISI (Nyquist), laju pensinyalan harus lebih kecil daripada $2W$ di mana W adalah bandwidth nominal dari kanal. Meskipun secara teori dimungkinkan untuk mencapai laju $2W$ dengan menggunakan pulsa Nyquist ideal, tetapi secara praktis tidak bisa direalisasikan. Akan tetapi, secara praktis mungkin dengan filter yang dapat direalisasikan untuk mecapai $2W$ dengan menambahkan sejumlah ISI yang dapat diatur dengan suatu cara sedemikian hingga dapat dihilangkan di penerima. Bentuk pulsa duobinary adalah contoh yang paling baik yang berkaitan dengan konsep ini: pulsa ini mempunyai amplitudo satu di titik asal dan di T , dan amplitudo nol di semua kelipatan bulat T .

4.1 Kita akan memodulasi urutan b (yang dibangkitkan di Bagian 2.2) dengan menggunakan pulsa-pulsa duobinary, mengirimkannya melalui kanal tanpa derau dengan bandiwdth 500 Hz, mengolah sinyal yang diterima dengan sebuah matched filter, dan menampilkan diagram mata yang diperoleh:

```
>>y = transmit(b,'duobinary',0,500);
```

Karena ISI yang digunakan di sini dapat dikontrol, pola mata menunjukkan dua bukaan salah satu di atas yang lain yang berkaitan dengan tiga level terima $\{-A, 0, A\}$ yang harus didekodekan. Oleh karena itu, dua nilai ambang, satu positif dan satu negatif, harus dipilih sebagai titik tengah vertikal dari bukaan mata atas dan bawah.

Catatan: perhatikan skala dari pola mata jika menentukan ambang positif dan negatif.

Anda akan menggunakan fungsi deteksi khusus `duodetect` untuk menerapkan aturan keputusan (*decision rule*):

```
ber =  
duodetect(<mf_output>,<neg_threshold>,<pos_threshold>,<sampling_  
g_time>,<mode>,<tx_data>,<delay>);
```

di mana `<mode>` diisi dengan string 'normal' dan `delay` diisi 20 bit.

```
>>duodetect(y,<neg_threshold>,<pos_threshold>,<sampling_t  
ime>,'normal',b,20)
```

duobinary, noiseless, $W = 500$ Hz $\Rightarrow P_e = \dots$

Anda akan mendapatkan laju kesalahan bit yang sempurna yaitu 0. Meskipun anda mendapatkan hasil yang sama dengan pulsa raised cosine dengan bandwidth 500 Hz, tetapi ada kelebihan bandwidth (*excess*) yang ditentukan oleh faktor *roll-off*-nya dan oleh karena itu juga berarti mengirim energi di luar 500 Hz yang bisa berinterferensi dengan kanal-kanal lain pada sistem praktis. Pengkodean duobinary memungkinkan pengiriman tanpa kesalahan melalui sistem yang benar-benar terbatas bandwidth-nya sebesar setengah dari laju dari dengan filter dengan filter yang dapat direalisasikan.

4.2 Kirim ulang urutan data `b` melalui kanal yang sama tetapi dengan menambahkan dengan PSD 1×10^{-7} W/Hz. Catat P_e pada Tabel 4 pada kolom 'normal'. Ulangi langkah ini untuk semua PSD derau yang tercantum.

4.3 Sekarang anda akan melakukan precoding terhadap urutan biner sebelum memodulasi dan mengirimkannya:

```
>>bb = precoding(b);  
>>y = transmit(bb,'duobinary',0,500);
```

Ulangi percobaan untuk semua level daya derau. Nilai ambang dan waktu sampling yang sama dapat digunakan. Catat hasil yang anda peroleh di Tabel 4 pada kolom "precoded". Gunakan mode keputusan 'coded' pada detektor:

```
>>duodetect(y,<neg_threshold>,<pos_threshold>,<sampling_time>,'coded',b,20)
```

Tabel 4: BER pada sistem duobinary

Daya derau	P_e – normal	P_e - precoded
1×10^{-7}		
5×10^{-7}		
10×10^{-7}		
50×10^{-7}		

TUGAS PASCA PERCOBAAN:

1. Gambarkan gelombang dari output sebuah matched filter jika inputnya adalah gelombang NRZ unipolar yang menyatakan urutan $\{1, 0, 1, 1, 0\}$.
2. Lihat Bagian 2 dari percobaan. Apakah lebih mudah untuk mendekodekan output dari kanal sebelum matched filter atau sesudahnya? Mengapa? Jika waktu sampling selain dari yang optimal (yang dapat diamati dari diagram mata) dipilih, P_e akan lebih besar. Mengapa?
3. Hitung probabilitas kesalahan bit teoritis untuk semua kondisi pada Tabel 1. Catat semua jawaban dalam bentuk tabel.
4. Amati pada Tabel 3 hubungan antara faktor *roll-off* dan BER untuk bandwidth 350 Hz. Dapatkah anda menjelaskannya berkaitan dengan sifat-sifat dari pulsa *raised-cosine*?
5. Apakah anda jumpai pada Tabel 4 unjuk kerja yang lebih baik dengan precoding? Jelaskan hasil yang anda peroleh berdasarkan teori dari literatur (referensi atau buku teks yang anda miliki).