

VI. PRAKTIKUM 5— Estimasi SNR dan *Correlation Receiver*.

VI.1 PENDAHULUAN

Pada Praktikum yang lalu (PRAKTIKUM 4) telah ditunjukkan bagaimana sistem menawarkan beberapa kandidat frekuensi pusat (CFC) yang akan digunakan sebagai referensi dalam proses demodulasi FSK. Setiap CFC memiliki estimasi sinyal terhadap noise (SNR) yang dihitung berdasarkan bandwidth WSPR sebagai berikut. Pertama, spektrum daya rata-rata diurutkan dari kecil ke besar, dan dapat diamati melalui TP-4. Estimasi level noise didapat berdasarkan pada asumsi bahwa spektrum daya noise adalah 30% dari level spektrum daya maksimum (yaitu, indeks frekuensi 123 dari keseluruhan indeks 411). SNR dari semua CFC kemudian dihitung dengan membagi masing-masing level daya spektrum CFC dengan estimasi level noise. Selanjutnya mahasiswa bertugas untuk melacak pendekatan ini melalui sintaks termasuk cara merepresentasikan SNR berdasarkan bandwidth saluran HF. Penjelasan tentang pendekatan ini telah dijelaskan dalam manual praktikum.

Korelator digunakan untuk mengubah setiap blok paket sinyal WSPR menjadi satu simbol. Agar supaya didapat hasil konversi terbaik, maka korelator memerlukan pewaktu/timing dan frekuensi yang akurat. Penyisipan bit-bit sinkronisasi pseudo-acak di sisi pemancar membuat penerima mendemodulasi sinyal WSPR lebih akurat. Deteksi dan pengenalan pola bit-bit sinkronisasi akan menentukan offset waktu yang diperlukan terutama untuk mengompensasi ketidak sinkronan waktu antara pemancar dan penerima. Diketahui bahwa panjang satu paket sinyal WSPR adalah 110,6 detik, sinkronisasi waktu dilakukan dengan menggeser maju dan mundur paket sinyal WSPR tersebut sambil mencari harga maksimum dari total koefisien korelasi yang merupakan akumulasi koefisien korelasi dari setiap blok yang memiliki sinyal paling berkorelasi (paling mirip).

Setiap blok sinyal didemodulasi menggunakan korelasi-silang terhadap empat frekuensi preset dari sinyal yang dihasilkan sumber lokal, untuk mengenali sinyal sebagai satu simbol dari 4 kemungkinan pilihan simbol. Lebih dari pada itu, memasang frekuensi preset dengan frekuensi offset diperlukan untuk melacak sekaligus mengompensasi drift karena adanya pergeseran Doppler serta drift/ingsutan osilator lokal disisi penerima. Waktu insut dan deviasi-frekuensi yang paling sesuai dengan total koefisien korelasi maksimum kemudian dicatat sebagai parameter kasar yang akan disempurnakan pada pemrosesan selanjutnya.

VI.2 PERANAN SYN-PATTERN (bit sinkronisasi)

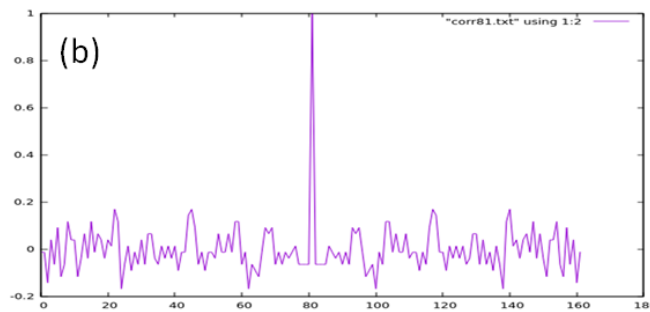
Bit sinkronisasi atau Syn-pattern mempunyai peran untuk membantu mengidentifikasi pattern simbol dari blok sinyal dimana masing-masing blok mengandung bit sinkronisasi sambil mempertahankan kondisi acak dari bit-bit data sinyal yang informasi.

Deretan bit penyinkron mempunyai panjang 162 bit dan bersifat acak yang dibuktikan dengan proses autokorelasi dari deretan bit tersebut, yang menunjukkan tidak terlihat periodisasi kecuali pada bit ke 81 yang menunjukkan korelasi maksimum. Hal ini disebabkan bit penyinkron ini terdiri

dari 2 blok bit (masing-masing 81 bit) yang dibuat sama namun setiap blok mempunyai 81 bit acak. Bit penyinkron dapat dilihat pada Gambar V-1 dibawah ini.

(a)

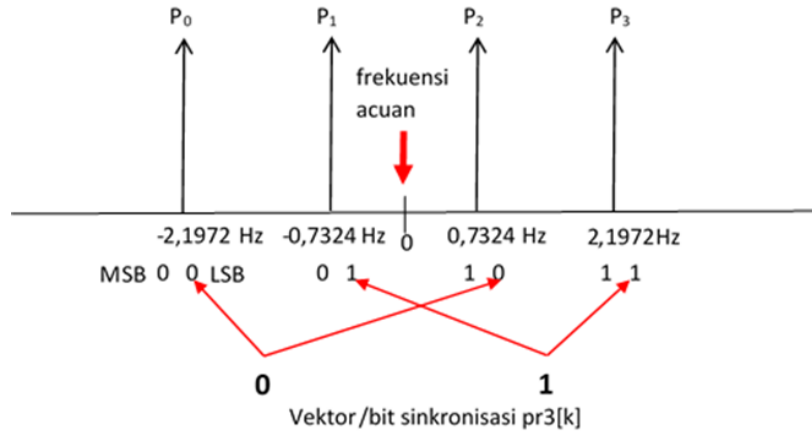
```
unsigned char pr3[162]=
{1,1,0,0,0,0,0,0,1,0,0,0,1,1,1,0,0,0,1,0,
 0,1,0,1,1,1,1,0,0,0,0,0,0,0,1,0,0,1,0,1,
 0,0,0,0,0,0,1,0,1,1,0,0,1,1,0,1,0,0,0,1,
 1,0,1,0,0,0,0,1,1,0,1,0,1,0,1,0,1,0,0,1,
 0,0,1,0,1,1,0,0,0,1,1,0,1,0,1,0,0,0,1,0,
 0,0,0,0,1,0,0,1,0,0,1,1,1,0,1,1,0,0,1,1,
 0,1,0,0,0,1,1,1,0,0,0,0,0,1,0,1,0,0,1,1,
 0,0,0,0,0,0,1,1,0,1,0,1,1,0,0,0,1,1,0,
 0,0};
```



Gambar VI-1: Bit Sinkronisasi atau Syn-Pattern yang mempunyai sifat random dan hanya berkorelasi pada bit ke 81; (a) adalah deretan bit penyinkron; (b) adalah autokorelasi dari bit penyinkron

Gambar VI-1(a) menunjukkan deretan bit penyinkron atau syn-pattern yang terdiri dari deretan bit random sepanjang 162 bit, sedangkan Gambar V-1(b) adalah grafik autokorelasi yang menunjukkan bahwa 162 bit penyinkron tersebut adalah random dan hanya berkorelasi pada geseran ke 81 karena memang 162 bit penyinkron tersebut terdiri dari 2 blok bit yang sama namun setiap blok terdiri dari bit-bit random.

Setelah bit penyinkron dipasang dengan bit data, maka simbol yang dibangkitkan seperti pada Gambar V-2.



Gambar VI-2: Tempat kedudukan spektrum frekuensi simbol setelah bit data disandingkan dengan bit sinkronisasi.

Tabel simbol setelah bit sinkronisasi digabung dengan bit message seperti ditunjukkan pada Tabel V-1 dibawah ini.

Tabel VI-1: Hubungan antara bit sinkronisasi dengan Frekuensi relatif yang membentuk FSK

Bit Message	Bit Sinkronisasi	Bit Simbol	Spektrum	Frekuensi relatif terhadap acuan
0	0	0 0	P0	-2,1972 Hz
0	1	0 1	P1	-0,7324 Hz
1	0	1 0	P2	+0,7324 Hz
1	1	1 1	P3	+2,1972 Hz

Proses sinkronisasi dilaksanakan dengan membandingkan deretan bit sinkronisasi dengan deretan level spektrum power sepanjang 162 blok simbol, dengan parameter insut waktu, kandidat frekuensi pusat (CFC), drift frekuensi, insut dopler dan nilai korelasi, seperti pada persamaan berikut ini,

$$ifreq0, shift0, drift0, sync0 = \max_{-10 \leq m \leq 22} \max_i \sum_{n=0}^{161} x(n) P_i(n - m)$$

dimana,

$$P_i(n - m) = (P1 + P3) - (P0 + P2)$$

dimana,

$$P0 = P_{i-3}(n - m)$$

$$P1 = P_{i-1}(n - m)$$

$$P2 = P_{i+1}(n - m)$$

$$P3 = P_{i+3}(n - m)$$

Dimana,

$x(n)$: vektor untuk sinkronisasi sepanjang 162 data.

$P_i(n - m)$: spektrum power setiap blok.

i : indeks frekuensi dengan resolusi 0,7324 Hz/indeks

n : indeks waktu dengan unit blok data.

m : indeks insut waktu (jitter) dengan unit blok data.

Dan selanjutnya,

Derajat kemiripan atau korelasi antara deretan bit sinkronisasi dengan deretan spektrum power sepanjang 162 simbol, selanjutnya disebut sebagai **korelasi-paket**, dihitung dengan cara seperti dibawah ini,

$$ss = ss + \left\{ \left(\frac{(0,1) \rightarrow (-1,+1)}{2 \times pr3[k] - 1} \right) \times ((P1 + P3) - (P0 + P2)) \right\}$$

$$pow = pow + P0 + P1 + P2 + P3$$

$$sync1 = \frac{ss}{pow}$$

Nilai ss (angka koefisien **korelasi-paket**) akan menjadi maksimum jika deretan bit sinkronisasi persis sama dengan nilai deretan

$(P1+P3)-(P0+P2)$. Sedangkan pow dipakai sebagai bilangan untuk menormalisasi nilai ss menjadi nilai yang akan dicatat sebagai koefisien **korelasi-paket** yang telah dinormalisasi $sync1$. Tabel kebenaran yang menunjukkan kondisi idial saat deteksi $P0, P1, P2$ dan $P3$.

Tabel VI-2: Tabel kebenaran yang menggambarkan hasil deteksi idial

Jika Terdeteksi				Jika Diketahui	Maka
$P1$	$P3$	$P0$	$P2$	Bit Sinkronisasi	$(P1 + P3) - (P0 + P2)$
besar	kecil	kecil	kecil	+1	Harus bernilai +
kecil	besar	kecil	kecil	+1	Harus bernilai +
kecil	kecil	besar	kecil	-1	Harus bernilai -
kecil	kecil	kecil	besar	-1	Harus bernilai -

Pada saat proses perhitungan untuk menentukan nilai *ss* dan *pow*, masih harus mengakomodasi drift dari osilator lokal, dengan cara memodelkan frekuensi drift sebagai fungsi waktu. Dalam algoritma WSPR, frekuensi drift dimodelkan berubah secara linear dengan waktu, seperti pada persamaan berikut ini,

$$ifd = ifr + \left(\frac{k-81}{81}\right) \left(\frac{idrift}{2}\right) \left(\frac{1}{df}\right), \quad k = 0,1,2,3 \dots,162$$

$$if0 - 2 \leq ifr \leq if0 + 2$$

$$if0 = \frac{freq0}{df} + 256$$

Dimana,

ifd: indeks frekuensi tengah setelah mengakomodasi frekuensi drift maupun sebaran doppler (sebaran Doppler menurut referensi berkisar beberapa Hz).

ifr: indeks frekuensi sebaran doppler yang diestimasi tidak lebih dari 4 Hz.

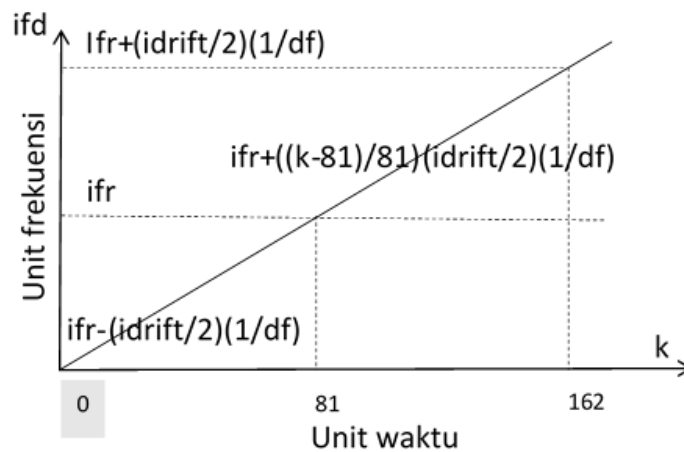
if0: indeks frekuensi referensi (CFC)

idrift: indeks frekuensi drift maksimum dari osilator lokal. Dalam algoritma WSPR frekuensi drift masih diperlebar dengan faktor 1/df yakni $1/0,7324 = 1,3653$.

df: resolusi sampel frekuensi yakni sebesar 0,7324 Hz per sampel. [$df = 375/512$]

freq0: kandidat frekuensi yang direpresentasikan menjadi frekuensi relatif terhadap 150 Hz, angka 256 memiliki arti bahwa power spektrum dikembalikan pada posisi semula yakni lebar observasi 375 Hz berpusat di 187,5 Hz, dari proses translasi dengan pusat 150 Hz dan lebar observasi 300 Hz yang dilaksanakan pada salah satu item PRAKTIKUM 4.

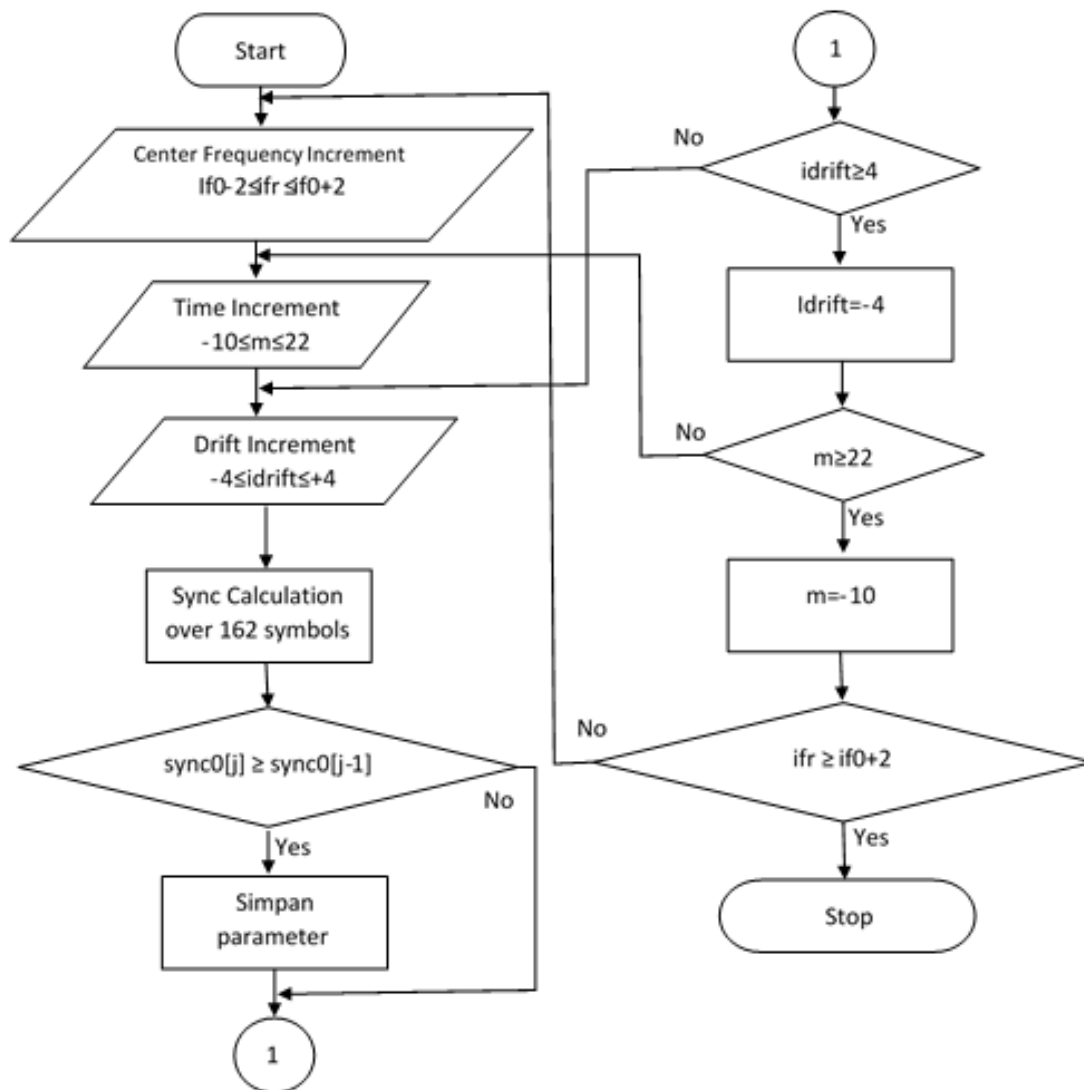
Rumus diatas secara grafis dapat ditunjukkan seperti pada Gambar VI-3 dibawah ini.



Gambar VI-3: Drift frekuensi lokal osilator dimodelkan perubahannya secara linear terhadap waktu.

Pemodelan drift linear ini berdasarkan asumsi bahwa drift maksimum terjadi pada awal dan akhir dan tidak ada tak ada drift pada saat blok ke 81 dari transmisi paket data, yang harganya digradasi sebesar $1/162$. Ini artinya bahwa 1 gradasi memiliki nilai indeks frekuensi $idrift/162$.

Diagram alir proses sinkronisasi waktu dapat dilihat pada Gambar VI-4.



Gambar VI-4: Diagram alir proses untuk menentukan coarse time shift (insut waktu kasar) yang nanti akan diperhalus dengan langkah-langkah selanjutnya.

Diagram alir pada Gambar VI-4 mempunyai bentuk sintak seperti pada Gambar VI-5 dibawah ini.

```

for(j=0; j<npk; j++) {      /* untuk semua kandidat frekuensi */
    smax=-1e30;
    if0=freq0[j]/df+256; /* dikembalikan ke frekuensi semula -->
                        dari relatif terhadap 150 Hz (indeks 205)
                        menjadi relatif terhadap 187,5 Hz (indeks 256) */
    for (ifr=if0-2; ifr<=if0+2; ifr++) {          //Freq search
        for( k0=-10; k0<22; k0++) {              //Time search
            for (idrft=-maxdrift; idrft<=maxdrift; idrft++) { /* Drift search,
maxdrift=4 */
                ss=0.0;
                pow=0.0;
                for (k=0; k<162; k++) { //Sum over symbols
                    ifd=ifr+((float)k-81.0)/81.0*( (float)idrft )/(2.0*df);
                    kindex=k0+2*k;
                    if( kindex < nffts ) {
                        p0=ps[ifd-3][kindex];
                        p1=ps[ifd-1][kindex];
                        p2=ps[ifd+1][kindex];
                        p3=ps[ifd+3][kindex];

                        p0=sqrt(p0);
                        p1=sqrt(p1);
                        p2=sqrt(p2);
                        p3=sqrt(p3);

                        ss=ss+(2*pr3[k]-1)*((p1+p3)-(p0+p2));
                        pow=pow+p0+p1+p2+p3;
                    }

                }

                sync1=ss/pow;

                if( sync1 > smax ) {      /** simpan parameter kasar **/
                    smax=sync1;
                    shift0[j]=128*(k0+1);
                    drift0[j]=idrft;
                    freq0[j]=(ifr-256)*df;
                    sync0[j]=sync1;
                }
            }
        }
    }
}

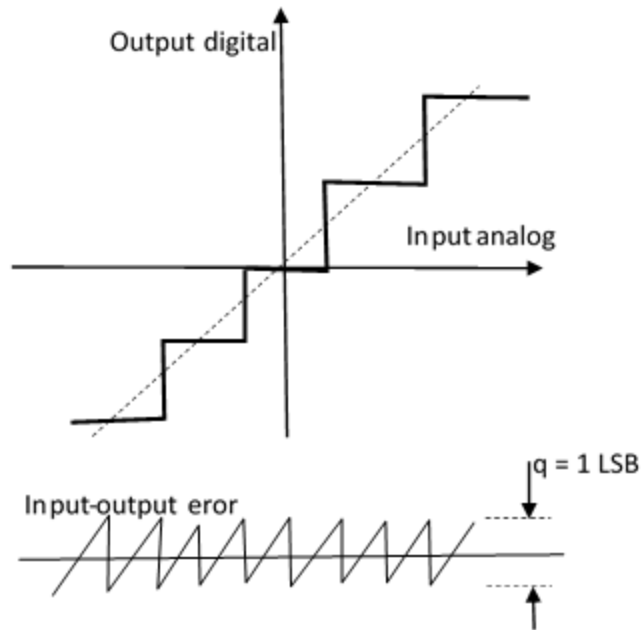
```

Gambar VI-5: Sintaksis dari proses mencari timing yang paling tepat dengan menyertakan pemodelan offset frekuensi berdasarkan efek doppler dan drift lokal osilator.

VI.3 Model Quantisasi Nois

Dibawah ini akan dijelaskan secara teoritis suatu proses quantisasi nois untuk suatu N-bit konverter analog-to-digital (ADC). Pertama tama dicari quatisasi tegangan nois, kemudian dari sini dapat dihitung SNR nya.

Suatu ADC idial akan mempunyai nilai eror yang berkisar pada besaran $\pm\frac{1}{2}$ LSB (least significant bit) seperti yang terlihat pada grafik transfer function suatu ADC N-bit idial seperti pada Gambar VI-6 dibawah ini,

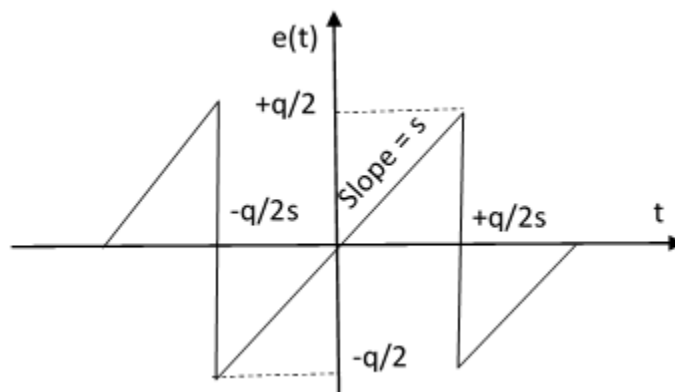


Gambar VI-6: Nois kuantisasi dari ADC idial N-bit

Dengan penyederhanaan dapat diasumsikan bahwa error kuantisasi merupakan suatu fungsi sinyal gigi gergaji. Untuk suatu analisis dapat dikatakan bahwa pendekatan ini mempunyai akurasi yang cukup baik. Persamaan sinyal gigi gergaji dapat ditulis sbb,

$$e(t) = st, \quad \frac{-q}{2s} < t < \frac{+q}{2s}$$

Representasi grafis dari persamaan ini seperti pada Gambar IV-6 dibawah ini,



Gambar VI-7: Nois kuantisasi sebagai fungsi dari waktu

Selanjutnya, kuadrat rerata dari $e(t)$ dapat ditulis sbb,

$$\overline{e^2(t)} = \frac{1}{T} \int_{-\frac{q}{2s}}^{\frac{q}{2s}} (st)^2 dt$$

Dimana $T = \frac{+q}{2s} - \left(-\frac{q}{2s}\right) = \frac{2q}{2s} = \frac{q}{s}$

$$\overline{e^2(t)} = \frac{s}{q} \int_{-\frac{q}{2s}}^{\frac{q}{2s}} (st)^2 dt$$

$$\overline{e^2(t)} = \frac{s^3}{3q} \left[t^3 \right]_{-\frac{q}{2s}}^{\frac{q}{2s}}$$

$$\overline{e^2(t)} = \frac{s^3}{3q} \left[\frac{q^3}{8s^3} + \frac{q^3}{8s^3} \right]$$

$$\overline{e^2(t)} = \frac{s^3}{3q} \left[\frac{2q^3}{8s^3} \right]$$

$$\overline{e^2(t)} = \frac{q^2}{12}$$

Nilai RMS atau harga efektif dari nois kuantisasi adalah $\sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}}$

Sinyal eror $e(t)$ menghasilkan harmonisa dari komponen harmonisa DC hingga harmonisa-harmonisa dengan frekuensi diatas lebar pita Nyquist ($F_s/2$). Namun demikian semua komponen harmonisa diatas $F_s/2$ akan tertekuk kedalam pita Nyquist dan saling menambahkan (fenomena ini juga disebut aliasing) sehingga menghasilkan nilai RMS nois sebesar $\frac{q}{\sqrt{12}}$

Beberapa penelitian menyimpulkan bahwa nois kuantisasi mempunyai distribusi mendekati Gaussian yang menyebar hampir secara uniform sepanjang $F_s/2$, dengan sama sekali tidak berkorelasi terhadap sinyal input.

Secara teoritis SNR dapat dihitung terhadap input sinusoidal dengan keadaan sekala penuh.

Sinyal sinusoidal,

$$v(t) = \frac{q2^N}{2} \sin(2\pi ft),$$

dimana q adalah step kuantisasi, sedangkan 2^N adalah jumlah anak-tangga kuantisasi yang merepresentasi peak to peak dari sinyal sinusoidal.

Nilai RMS dari sinyal sinusoidal tersebut adalah $\frac{q2^N}{2\sqrt{2}}$, sehingga nilai SNR maksimum dari suatu ADC ideal N-bit adalah,

$$SNR = 20 \log_{10} \frac{RMS v(t)}{RMS e(t)}$$

$$SNR = 20 \log_{10} \left[\frac{q2^N / 2\sqrt{2}}{q\sqrt{12}} \right] = 20 \log_{10} 2^N + 20 \log_{10} \sqrt{\frac{3}{2}}$$

$$SNR = 6,02N + 1,76dB, \quad \text{pada BW sepanjang lebar pita antara dc hingga } \frac{F_s}{2}$$

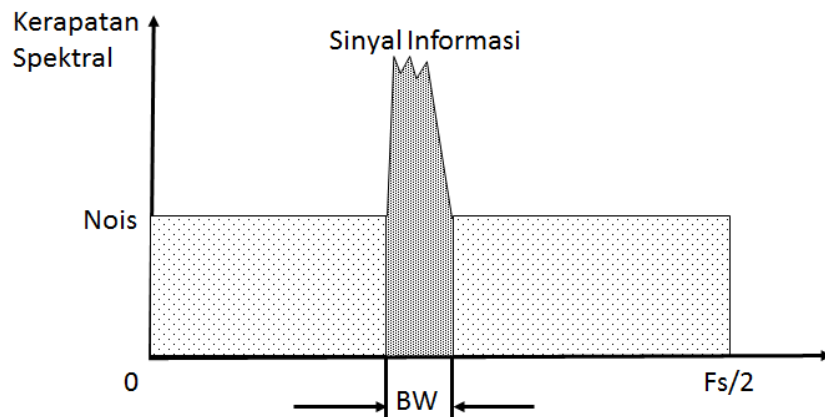
Rumus ini sangat lazim dipakai untuk pendekatan perhitungan SNR maksimum untuk ADC ideal.

VI.4 Spektrum Frekuensi dari Nois Kuantisasi

Dalam banyak desain sistem modulasi modern, sering sinyal informasi hanya menempati porsi kecil dari lebar pita (BW) Nyquist. Jika filter digital dipasang untuk membuang semua komponen nois yang berada diluar BW informasi, maka pada saat menghitung SNR, faktor koreksi harus dikenakan, yang tentu saja menyebabkan nilai SNR sinyal informasi menjadi lebih besar, seperti pada persamaan dibawah ini. Faktor koreksi ini juga lazim disebut gain-proses.

$$SNR = 6,02N + 1,76dB + 10 \log_{10} \frac{F_s}{2 \times BW} \quad \text{sepanjang bandwidth BW}$$

Ilustrasi gain-proses dapat dilihat pada Gambar VI-7 dibawah ini,



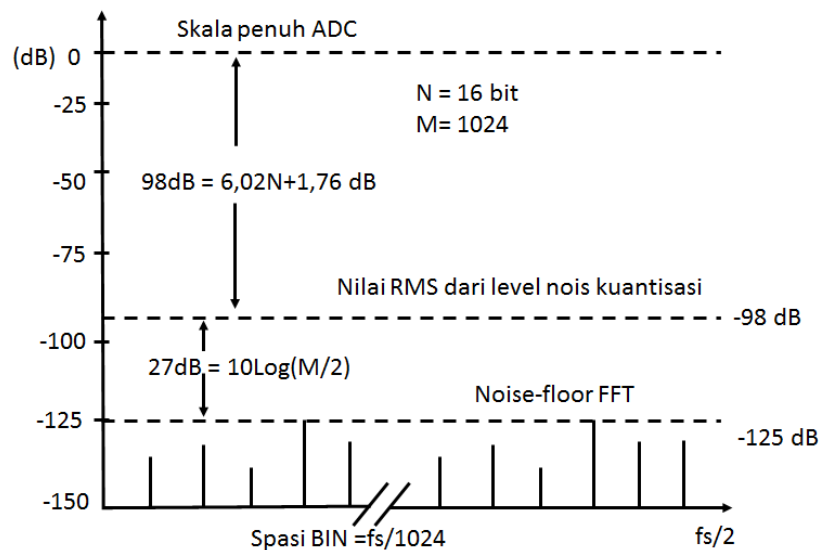
Gambar VI-8: Spektrum Nois Kuantisasi menunjukkan Gain-Proses

Menghitung SNR adalah proses pembagian antara luasan power spektral sinyal dengan luasan power spektral nois. Maka jika yang dipakai sebagai faktor pembagi adalah nois selebar BW saja hasilnya akan lebih besar jika dibanding dengan faktor pembagiya adalah nois selebar pita Nyquist ($F_s/2$).

VI.5 SNR, Gain-Proses dan Noise-Floor FFT

Gambar VI-9 menunjukkan output FFT dengan jumlah titik 1024 ($M=1024$), untuk suatu 16 bit ADC idial. Sebagai catatan bahwa nilai relatif rerata noise floor adalah sekitar -98 dB (jika dianggap skala penuh ADC adalah 0 dB). Noise floor FFT bukanlah nilai SNR dari ADC, karena FFT berperan sebagai analiser spektrum analog dengan BW satu Bin (ember) F_s/M , dimana M adalah jumlah titik FFT. Jika 1 bin mewakili F_s/M Hz, maka jumlah titik FFT M idial adalah sama dengan F_s . Sehingga secara teoritis noise floor FFT identik dengan nilai $10 \log_{10} \left(\frac{M}{2} \right) dB$ dibawah nilai noise floor kuantisasi karena adanya gain pemrosesan dari FFT.

Untuk suatu ADC idial 16-bit dengan SNR idial 98 dB, maka FFT 1024 titik akan menyebabkan gain pemrosesan sebesar $10 \log_{10} \left(\frac{1024}{2} \right) = 27dB$, sehingga secara keseluruhan noise-floor FFT adalah senilai $98 \text{ dB} + 27 \text{ dB} = 125 \text{ dBc}$. Dapat dilihat disini bahwa 27 dB level nois sekitar 30% dari skala penuh ADC. Pada saat mmentukan SNR, didalam program WSPR angka prosentase empirik 30% dipakai untuk menentukan level nois yang akan dibandingkan dengan level maksimum pada saat spektrum diurutkan levelnya dari yang kecil ke yang paling besar. Selanjutnya dapat juga dilihat bahwa noise-floor FFT dapat terus ditekan lebih kecil dengan cara menambah jumlah titik FFT. Ilustrasi dari proses ini dapat dilihat pada Gambar VI-8 dibawah ini.



Gambar VI-9: Noise floor untuk suatu ADC 16-bit idial dengan 1024 titik FFT

VI.6 Nilai SNR negatif

Anda dan pasangan mendatangi jamuan resepsi pernikahan lebih awal. Beberapa keluarga dekat sudah hadir namun percakapan penuh rasa persaudaraan dengan mudah di antara para tamu yang hadir awal tersebut. Pada saat acara resepsi dimulai, area resepsi hampir penuh, musik pengiringpun sudah mulai memainkan perannya dengan hinggar bingar. Percakapan diantara pengunjung menjadi sulit. Perlu berteriak agar lawan bicara dapat mengerti isi pembicaraan kita. Apa yang telah terjadi adalah bahwa tingkat kebisingan latar belakang (ambien) telah meningkat, dan untuk mempertahankan komunikasi Anda harus mengimbangi dengan berbicara lebih keras (lebih banyak kekuatan sinyal) untuk mempertahankan tingkat suara percakapan terhadap tingkat kebisingan yang meningkat (lebih banyak kekuatan tingkat kebisingan). Ini identik dengan apa yang terjadi di dunia Radio Frequency (RF) di mana sinyalnya adalah sinyal radio Anda dan suara bising adalah atmosfer, listrik, atau sinyal radio lain yang mengganggu.

Signal to Noise Ratio (SNR) adalah gambaran keunggulan yang membandingkan tingkat sinyal yang diinginkan dengan tingkat kebisingan latar belakang. Ini dihitung sebagai rasio kekuatan sinyal terhadap daya noise. SNR dinyatakan dalam dB (desibel).

$$[SNR]_{dB} = 10 \times \log_{10} \left[\frac{\text{power sinyal dalam watt}}{\text{power noise dalam watt}} \right]$$

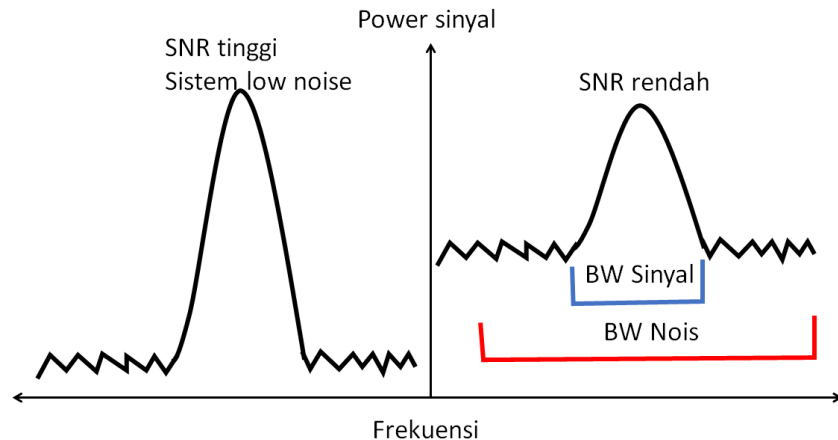
Untuk sinyal kuat, maka secara kasar dapat kita dapat menghitung SNR dari RSSI atau S-meter yang terpasang pada radio. Jika sinyal CW (morse) yang diterima adalah S9 dan saat tidak ada sinyal meter *field-strength* (S-meter) menunjuk S7, ini artinya SNR-nya adalah 12dB (lazimnya setiap unit S mewakili perbedaan 6dB). Diasumsikan bahwa sinyal ini diterima melalui bandwidth filter penerima dengan lebar 2400 Hz. Selanjutnya sinyal tersebut dilewatkan pada filter sempit yang khusus untuk keperluan CW selebar hanya 500 Hz. Maka pembacaan noise seharusnya turun hingga di bawah S6 untuk peningkatan 6.8dB. Hal ini adalah merupakan rasio dari dua bandwidth seperti pada persamaan dibawah ini,

$$10 \log_{10} \left[\frac{2400 \text{ Hz}}{500 \text{ Hz}} \right] = 6.81 \text{ dB}$$

Bandwidth yang sempit akan membuang/memfilter lebih banyak noise sehingga SNR yang didapat akan meningkat. SNR sekarang menjadi 12 dB + 6,81 dB = 18,81 dB. Namun perlu diperhatikan bahwa sinyal yang diinginkan masih harus melewati filter. Perlu diingat bahwa melewatkan sinyal SSB yang secara nominal memiliki BW 2400Hz kedalam filter dengan BW 500 Hz akan membuat kualitas sinyal menjadi tidak terbaca. Sehingga SNR bukan tujuan, namun harus sesuai dengan BW informasi asli yang sedang dibawa.

Nois ada di seluruh spektrum frekuensi radio. Dapat didominasi oleh gangguan eksternal yang diterima oleh antena (kasus biasa untuk sinyal RF <30MHz, HF) atau oleh gangguan internal yang ditambahkan oleh sistem penerima (kasus biasa untuk sinyal > 30MHz, VHF, UHF).

Pengukuran SNR harus merujuk pada lebar BW yang sudah ditentukan, atau lazim disebut sebagai Bandwidth Noise (NBW) seperti pada gambar dibawah ini,



Gambar VI-10: Perbedaan BW sinyal (biru) dan BW nois (merah) saat pengukuran SNR akan membuat salah pengertian (misleading) dalam mengartikan angka SNR.

Sinyal SSB biasanya diterima melalui filter baseband 2600 Hz atau 2400 Hz. Di seluruh komunitas radio amatir, angka 2500 (Hz) telah diadopsi sebagai gambaran untuk membandingkan kemampuan penerimaan suatu mode. Ragam/moda yang lazim dipakai oleh para amatir radio adalah SSB, CW, RTTY, WSPR, dll. Dalam hal ini SSB akan dipakai sebagai acuan bandwidth lebar 2500 Hz. Kenapa SSB dipilih sebagai acuan, karena SSB adalah ragam telefoni dimana jika dilewatkan kedalam filter yang lebih kecil, kita akan mulai kehilangan energi sinyal selain menurunkan energi noise. Hilangnya sinyal ini akan dengan cepat membuat sinyal SSB tidak dapat dipahami (intelligibility rendah). Bandwidth menjadi faktor pembatas untuk mode SSB.

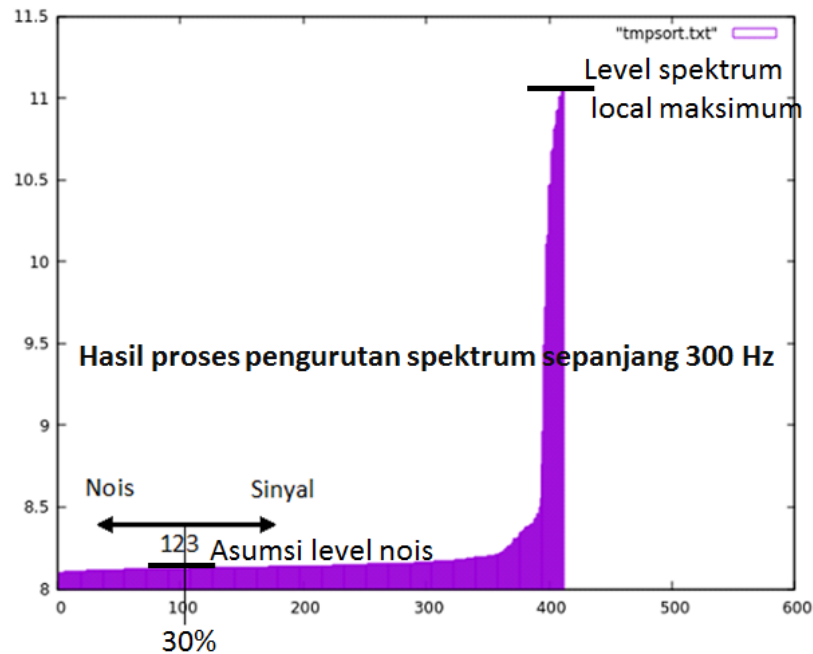
CW adalah sinyal yang jauh lebih sempit. Filter umum yang digunakan meliputi 2400Hz, 500Hz, 250Hz, dan 100Hz. Dengan DSP (Digital Signal Processing) bahkan kita bisa dapatkan filter yang lebih sempit lagi. Moda yang paling membutuhkan BW paling sempit adalah WSPR yang hanya menempati BW selebar 6 Hz.

Dalam buku standar ITU-R F.339-8 perihal persyaratan SNR, disebutkan bahwa SNR 6dB adalah syarat untuk komunikasi moda telefoni SSB secara normal. Secara empirik di lapangan diketahui bahwa sinyal SSB hampir tidak dapat dibaca pada SNR 0 db. Kondisi kanal HF saat percobaan empirik ini adalah tanpa QRM (nois akibat ulah manusia), tidak ada nois karena perkabelan buruk dan tidak ada fading.

Pada buku standar yang sama kita melihat bahwa untuk operasi telegrafi/CW normal, terhadap SSB didapat 10 dB lebih rendah atau SNR -4 dB dengan asumsi BW sinyal telegrafi 250 Hz, untuk derajat keterbacaan yang sama seperti normal SSB. Namun apakah sebenarnya makna dari SNR -4 db? Bukankah itu berarti sinyal di bawah noise? Bagaimana kita dapat membacanya. Ingat ini relatif terhadap bandwidth 2500 Hz. Kita biasanya akan memasang filter 250 Hz sehingga sekarang SNR 6 dB untuk telegrafi yang akan mempunyai derajat keterbacaan sama dengan SNR 6 dB moda SSB.

VI.7 Mencari Lokal Maksimum dan Menentukan SNR

Sebelum menghitung/menentukan SNR, diperlukan proses untuk menentukan level noise yang akan dipakai sebagai acuan menghitung SNR. Setelah didapat harga `smspec[i]`, kemudian dilakukan proses pengurutan berdasarkan level spektrum frekuensi. Estimasi level noise dalam spektrum yang sedang diukur adalah urutan level spektrum kecil ke besar 30 % (123 dari 411). Angka 30% adalah angka empirik dengan asumsi level noise yang masih dapat diukur oleh instrumen pengukuran adalah maksimum 30% dari power maksimum sinyal yang ada. Untuk memperjelas langkah pengurutan ini maka perhatikan Gambar VI-11.



GAMBAR VI-11: PENENTUAN LEVEL NOIS DENGAN CARA PENGURUTAN KOMPONEN SPEKTRUM DAYA SINYAL

Komponen spektrum frekuensi daya yang ke 123 dapat dianggap sebagai level noise. SNR per unit spektrum dapat dirumuskan sbb,

$$SNR_{unit_spektrum} = \begin{cases} \frac{Level_{spektrum}}{Level_{nois}}, & \text{jika } > Level_{nois} \\ 0,1 \times SNR_{min}, & \text{jika } < Level_{nois} \end{cases}$$

Dimana untuk bandwidth WSPR 5,8592 Hz, $SNR_{min} = -8 \text{ dB}$, sehingga dalam perhitungan linear didapat

$$SNR_{min} = 10^{-0,8} = 0,1585$$

Sedangkan bandwidth kanal adalah 3000 Hz, maka faktor skala adalah,

$$Scaling_factor = 10 \times \log_{10} \frac{3000}{5,8592} \approx 27 \text{ dB}.$$

$$SNR_{kanal} = 10 \times \log_{10} SNR_{WSPR} - 27 \text{ dB}$$

VI.8 LANGKAH PERCOBAAN

1. Siapkan modul Raspberry Pi, nyalakan dan buka satu atau beberapa terminal window.
2. Masuk ke direktori "PRAKTIKUM-SYN"
3. Siapkan beberapa file WSPR hasil perekamanyang berasal dari beacon laboratorium dan yang berasal dari tempat yang jauh, dan letakkan pada direktori ini.
4. Kompilasi program "./wsprd" dengan menjalankan "Makefile"
%> make -k
5. Jalankan program "./wsprd"
%> ./wsprd xxxxxx.WAV \r atau
%> ./wsprd xxxxxx.raw \r
6. Sambil menghitung bulatan merah pada gambar plotting CFC, cocokkan dengan jumlah CFC (Centre Frequency Candidate) yang dipaparkan oleh printf.
7. Ulangi langkah 5 untuk beberapa file WSPR yang berbeda.
8. Amati fenomena jumlah CFC, dengan membandingkan file WSPR dari sumber lokal dan WSPR dari sumber pemancar yang jauh.
9. Laksanakan tes korelasi terhadap suatu CFC dengan cara memasukkan nomor kandidat dan tekan "enter"
10. Laksanakan langkah 9 beberapa kali dengan file WSPR yang sama namun memilih nomor kandidat yang berbeda.
11. Perhatikan fenomena yang terjadi antara nomor kandidat kecil dan nomor kandidat besar dengan melihat kesamaan antara bit sinkronisasi dan nilai power spektrum terdeteksi $((P1 + P3) - (P0 + P2))$, yang secara intuitif kasar dapat dilihat dari kesamaan tanda dan besar nilai power spektrum yang terdeteksi.
12. Secara intuitif kesimpulan apakah yang dapat Saudara katakan dari kasil langkah 11.

VI.9 LAPORAN PRAKTIKUM

1. Jelaskan arti npk pada sintak for(j=0; j<npk; j++) {
2. Jelaskan mengapa harga smax=-1e30;

3. Jelaskan angka 256 pada sintak `if0=freq0[j]/df+256;`
 4. Jelaskan arti `if0+2` dan `if0-2` pada sintak `for (ifr=if0-2; ifr<=if0+2; ifr++) {`
 5. Jelaskan arti angka -10 dan 22 pada sintak `for(k0=-10; k0<22; k0++) {`
 6. Jelaskan arti `-maxdrift` dan `+maxdrift` pada sintak `for(idrift=-maxdrift; idrift<=maxdrift; idrift++) {`
 7. Jelaskan maksud "`2*pr3[k]-1`" pada sintak `ss=ss+(2*pr3[k]-1)*((p1+p3)-(p0+p2));`
 8. Laporan dikumpulkan sebelum PRAKTIKUM-6 dimulai.
-
-

