

## XI. PRAKTIKUM 10— Pembangkitan Lokal Osilator Secara Perangkat Lunak

### XI.1 PENDAHULUAN

FSK detektor memerlukan osilator yang mampu secara otomatis melacak secara diskrit semua insutansi frekuensi yang acak maupun yang terprediksi. Untuk itulah diperlukan pembangkit sinusoidal yang diharapkan mempunyai waktu eksekusi yang cukup cepat untuk menjejak perubahan frekuensi baik akibat proses modulasi maupun akibat sebaran/spread doppler saat propagasi maupun disebabkan oleh drif akibat kualitas lokal osilator yang ada pada radio penerima. Persoalan utama pada proses pembangkitan sinyal sinusoidal secara perangkat lunak adalah waktu prosesing untuk pembangkitan sinyal. Perlu upaya mencari algoritma yang lebih efisien guna dapat memangkas waktu prosesing pembangkitan sinyal sinusoidal. Pada PRAKTIKUM 10 ini akan dibahas bagaimana memilih teknik pembangkitan sinusoidal yang lebih efektif sehingga dapat memangkas waktu dalam proses pembangkitan sinusoidal tersebut.

### XI.2 Fungsi Transendental

Fungsi sinusoidal adalah fungsi transendental yang memerlukan cara pembangkitan berdasarkan pada deret Maclaurin yang melibatkan banyak proses penambahan, pengurangan maupun perkalian dan pembagian, seperti tampak pada persamaan  $\sin(x)$  dan  $\cos(x)$  dengan deret Maclaurin dibawah ini,

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$
$$\cos(x) = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

Tampak bagaimana komputer mengeksekusi satu fungsi  $\sin(x)$  dan  $\cos(x)$ , sehingga dapat dibayangkan betapa lambatnya komputer saat mengeksekusi fungsi transendental  $\sin(x)$  dan  $\cos(x)$  tersebut.

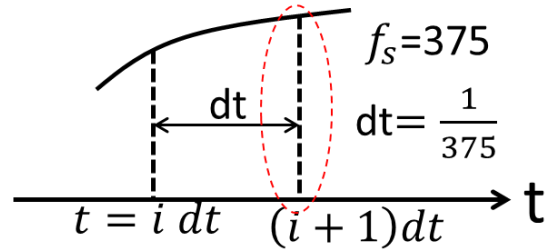
Selain dari pada itu, fungsi-fungsi transendental pasti melibatkan operasi dengan variable bilangan floating yang akan memakan waktu eksekusi aritmatika sekitar 10 kali lebih lambat dibanding operasi aritmatika untuk bilangan-bilangan integer.

Dalam rangka pembangkitan sinyal sinusoidal, WSPR memiliki cara yang lebih cepat dibandingkan cara konvensional hanya sekedar memanggil fungsi,

$$s_0(n) = \sin\left(2\pi \frac{f}{f_s} n\right) \quad n = 1, 2, 3, \dots, 255,$$

dimana,  $f$  adalah frekuensi sinyal dan  $f_s$  adalah frekuensi sampling.

Cara yang lebih cepat yang dipakai oleh WSPR dengan cara mengurangi kekerapan pemanggilan fungsi  $\sin(x)$  dan  $\cos(x)$  adalah sbb,



**Gambar X-1:** Gambaran 2 sampel digital yang berdampingan dari suatu sinyal sinusoidal  
 Sampel ke  $(i+1)$  dari sinyal sinusoidal digital pada Gambar X-1 dapat direpresentasikan sebagai,

$$x(i+1) = \sin 2\pi f \left[ \frac{i}{f_s} + \frac{1}{f_s} \right] = \sin 2\pi f \left[ \frac{i}{f_s} \right] \cos 2\pi f \left[ \frac{1}{f_s} \right] + \cos 2\pi f \left[ \frac{i}{f_s} \right] \sin 2\pi f \left[ \frac{1}{f_s} \right],$$

$$\sin 2\pi f \left[ \frac{i+1}{f_s} \right] = \sin 2\pi f \left[ \frac{i}{f_s} \right] \cos 2\pi f \left[ \frac{1}{f_s} \right] + \cos 2\pi f \left[ \frac{i}{f_s} \right] \sin 2\pi f \left[ \frac{1}{f_s} \right],$$

dimana  $f$  adalah frekuensi sinyal, dan  $f_s$  adalah frekuensi sampling.

Selanjutnya substitusi,

$$j = i + 1 \rightarrow i = j - 1 \rightarrow j = 1, 2, 3, \dots, 256,$$

didapat,

$$\sin 2\pi \left[ \frac{f}{f_s} j \right] = \sin 2\pi \left[ \frac{f}{f_s} (j-1) \right] \cos 2\pi \left[ \frac{f}{f_s} \right] + \cos 2\pi \left[ \frac{f}{f_s} (j-1) \right] \sin 2\pi \left[ \frac{f}{f_s} \right]$$

dimana,

$$\sin 2\pi \left[ \frac{f}{f_s} j \right] = 0, \text{ untuk } j = 0$$

Sedangkan untuk sinyal  $\cos(x)$  dengan cara yang sama diturunkan sebagai berikut,

$$\cos 2\pi f \left[ \frac{i}{f_s} + \frac{1}{f_s} \right] = \cos 2\pi f \left[ \frac{i}{f_s} \right] \cos 2\pi f \left[ \frac{1}{f_s} \right] - \sin 2\pi f \left[ \frac{i}{f_s} \right] \sin 2\pi f \left[ \frac{1}{f_s} \right],$$

$$\cos 2\pi f \left[ \frac{i+1}{f_s} \right] = \cos 2\pi f \left[ \frac{i}{f_s} \right] \cos 2\pi f \left[ \frac{1}{f_s} \right] - \sin 2\pi f \left[ \frac{i}{f_s} \right] \sin 2\pi f \left[ \frac{1}{f_s} \right]$$

Selanjutnya substitusi,

$$j = i + 1 \rightarrow i = j - 1 \rightarrow j = 1, 2, 3, \dots, 256,$$

didapat,

$$\cos 2\pi \left[ \frac{f}{f_s} j \right] = \cos 2\pi \left[ \frac{f}{f_s} (j-1) \right] \cos 2\pi \left[ \frac{f}{f_s} \right] - \sin 2\pi \left[ \frac{f}{f_s} (j-1) \right] \sin 2\pi \left[ \frac{f}{f_s} \right]$$

dimana,

$$\cos 2\pi \left[ \frac{f}{f_s} j \right] = 1, \text{ untuk } j = 0$$

Nampak pada persamaan generator sinusoidal cepat, bahwa pemanggilan fungsi  $\sin(x)$  maupun  $\cos(x)$  hanya dilakukan sekali saja yakni pada saat menghitung  $\sin(2\pi f/f_s)$  dan  $\cos(2\pi f/f_s)$ . Selanjutnya adalah perhitungan iterasi 255 kali dengan memakai fungsi aritmatika biasa. Bandingkan dengan jika hanya memanggil fungsi  $\sin(x)$  dan  $\cos(x)$  sepanjang 255 kali perhitungan.

Dibawah ini adalah sintaksis dari pembangkitan sinyal referensi secara standar,

```

/*
for (j=1; j<256; j++){
    c0[j]=cos(dphi0*(double)j);
    s0[j]=sin(dphi0*(double)j);
    c1[j]=cos(dphi1*(double)j);
    s1[j]=sin(dphi1*(double)j);
    c2[j]=cos(dphi2*(double)j);
    s2[j]=sin(dphi2*(double)j);
    c3[j]=cos(dphi3*(double)j);
    s3[j]=sin(dphi3*(double)j);
}
*/

```

Sedangkan sintaksis dari pembangkitan sinyal referensi secara cepat dapat dilihat berikut ini,

```

c0[0]=1; s0[0]=0;
c1[0]=1; s1[0]=0;
c2[0]=1; s2[0]=0;
c3[0]=1; s3[0]=0;
/*
for (j=1; j<256; j++) {
    c0[j]=c0[j-1]*cdphi0 - s0[j-1]*sdphi0;
    s0[j]=c0[j-1]*sdphi0 + s0[j-1]*cdphi0;
    c1[j]=c1[j-1]*cdphi1 - s1[j-1]*sdphi1;
    s1[j]=c1[j-1]*sdphi1 + s1[j-1]*cdphi1;
    c2[j]=c2[j-1]*cdphi2 - s2[j-1]*sdphi2;
    s2[j]=c2[j-1]*sdphi2 + s2[j-1]*cdphi2;
    c3[j]=c3[j-1]*cdphi3 - s3[j-1]*sdphi3;
    s3[j]=c3[j-1]*sdphi3 + s3[j-1]*cdphi3;
}
*/

```

### XI.3 LANGKAH PERCOBAAN

1. Masuk ke direktori PRAKTIKUM-OSC  
%> cd PRAKTIKUM-OSC \r
2. Pada program "wsprd.c" ubah rutin generator sinusoidal menjadi rutin standar pada fungsi void `sync_and_demodulate()` dan fungsi void `noncoherent_sequence_detection()` dengan cara merevisi remarks.
3. Hapus/remarks potongan sintaksis pembangkitan sinusoidal cepat atau standar sesuai dgn langkah percobaan.
4. kompilasi ulang.  
%> make -k \r

5. Jalankan program "wsprd" dengan file wav yang mempunyai hasil SNR kecil  
%> ./wsprd ./170244\_0730-25.WAV \r
6. Lihat dan catat isi file wspr\_timer.out
7. Kembali langkah 1 dengan mengubah kembali ke rutin pembangkitan sinyal cepat.
8. Lanjutkan ke langkah 2, 3 dan 4.
9. Bandingkan waktu eksekusi antara rutin standar dan rutin cepat dengan melihat tabulasi waktu yang ada di file "wspr\_timer.out" (Perhatian: file wspr\_timer.out adalah hanya untuk percobaan saat itu, artinya setiap langkah percobaan maka file lama akan terhapus dengan file hasil pengukuran waktu yang terbaru).
10. Lakukan percobaan diatas paling tidak untuk 10 file wav yang berbeda.

#### **XI.4 LAPORAN PRAKTIKUM**

1. Tunjukkan potongan sintaksis dalam wsprd.c yang membangkitkan sinyal sinusoidal secara cepat atau secara standar.
  2. Buatlah grafik perbandingan waktu dekode yang hanya menyangkut pembangkitan sinyal sinusoidal saja.
  3. Tunjukkan apakah ada relasi antara besar SNR dengan derajat perbandingan waktu antara pembangkitan standar dan pembangkitan cepat.
  4. Laporan dikumpulkan beserta jawaban quisioner.
- 
-

## REFERENSI

1. J. H. Taylor and B. Walker, "WSPRing around the world", QST, 94(11), 30–32, 2010.
2. J. H. Taylor, (2014. Feb 1), Weak Signal Propagation Reporter [Online]. Available: <http://physics.princeton.edu/pulsar/K1JT/wspr.html>.
3. R. Abdelmoumen, "A Review of Link Layer Protocols for Internet of Things", International Journal of Computer Applications (0975 – 8887), Vol. 182 – No.46, March 2019.
4. N. A. Frissell , J. S. Vega , E. Markowitz , A. J. Gerrard , W. D. Engelke , P. J. Erickson, E. S. Miller , R. C. Luetzelschwab , and Jacob Bortnik, "High-Frequency Communications Response to Solar Activity in September 2017 as Observed by Amateur Radio Networks", AGU Space Weather, DOI:10.1029/2018SW002008.
5. K. Schwab, "The Fourth Industrial Revolution", Currency Newyork, 2017.
6. Raspberry Pi. (2017). [Online]. Available: <http://www.raspberrypi.org>
7. Raspberry Pi Academy. (2017). [Online]. Available: <http://www.raspberrypi.org/picademy>
8. S. Güzelgöz and H. Arslan, "A Wireless Communications Systems Laboratory Course", IEEE Transaction on Education, Vol. 53, No. 4, Nov 2010.
9. Z. Dawy, A. Hussein, E. Yaacoub, and L. Al-Kanj, "A Wireless Communications Laboratory on Cellular Network Planning", IEEE Transaction on Education, Vol. 53, No. 4, NOV 2010.
10. G. Pasolini, A. Bazzi, and F. Zabini, "A Raspberry Pi-Based Platform for Signal Processing Education", IEEE Signal Processing Magazine, pp 151-159, Jul 2017.
11. J. Mills and D. F. Treagust, "Engineering Education – Is Problem Based Learning Or Project-based Learning The answer ?", Australian J. of Engng. Educ. [Online]. 2003-04. Available: [http://www.aeee.com.au/journal/2003/mills\\_treagust03.pdf](http://www.aeee.com.au/journal/2003/mills_treagust03.pdf)
12. GNU General Public License [online] Available: <http://www.gnu.org/education/education.html>
13. Jan Newmarch, "Linux Sound Programming", Oakleigh, Victoria, Australia, 2017
14. Matteo Frigo, Steven G. Johnson, "Manual documents version 3.3.6-pl1 of FFTW", 15 January 2017.
15. FFTW open source. [online] Available: <http://www.fftw.org>
16. GNUPLOT open source. [online] Available: <http://www.gnuplot.info>
17. WSPR open source. [online] Available: <https://sourceforge.net/p/wsjt/wsjt/6122/tree/branches/wsjitx/lib/wsprd/>
18. Reverse the bits in a byte with 4 operations (64-bit multiply, no division): <http://graphics.stanford.edu/~seander/bithacks.html#InterleaveBMN>
19. P. Ammann, and J. Offutt, "Introduction To Software Testing," 2nd Edition, Cambridge University Press, 2017.
20. P. Karn, "Toward New Link Layer Protocols", [online] Available: <https://www.semanticscholar.org/paper/Toward-New-Link-Layer-Protocols-Karn/02cf11ea008af92bec1d20318683251059a081ac>, 1996.
21. P. Karn, "Convolutional Decoders for Amateur Packet Radio", [online] Available: <https://www.semanticscholar.org/paper/Convolutional-Decoders-for-Amateur-Packet-Radio-Karn/eb1f03a9fb33692f5eeb87e60507e43f2346bc0b>, 2014.

22. A. Oppenheim, R.W. Schaffer, "Digital Signal Processing", Prentice-Hall, 1975
23. J.G. Proakis, M. Salehi, "Digital Communications ", 5<sup>th</sup> Edition, McGraw-Hill, 2005
24. ] A. Ghasemi, A. Abedi, F. Ghasemi, "Propagation Engineering in Wireless Communication", Springer, 2012
25. ] K. Oldham, J. Myland, J. Mespanier, "An Atlas of Functions With Equator, The Atlas Function Calculator", Second Edition, Springer, 2009.
26. Daniel Terlep, "How Quantization and Thermal Noise Determine an ADC's Effective Noise Figure", Application Note 1197: Application Note 1197:  
<http://www.maximintegrated.com/an1197>, Sep 2002
27. Brian D Hamerski, "Signal to Noise Ratio, definition and application to Radio Communications" , <https://kf6hi.net/radio/SNR.html>