



Taufik Hidayat Soi, ST,MT

SISTEM PEMROSESAN SINYAL 1



DSP (Digital Signal Processing) adalah teknik pengolahan sinyal yang mengubah sinyal-sinyal analog menjadi digital sehingga dapat diproses menggunakan komputer. DSP memiliki aplikasi yang sangat luas di berbagai bidang seperti komunikasi, pemrosesan citra, kendali sistem, pengolahan suara, dan lain sebagainya.

Beberapa dasar teori DSP yang penting antara lain:

1. Konversi sinyal analog ke sinyal digital: Proses konversi sinyal analog ke sinyal digital melibatkan tahapan sampling, quantization, dan encoding. Sampling adalah proses mengambil sampel sinyal analog pada waktu diskrit tertentu, quantization adalah proses memetakan nilai sinyal analog menjadi nilai digital terdekat dalam suatu rentang nilai, dan encoding adalah proses merepresentasikan nilai-nilai digital sebagai urutan bit-bit biner.



Taufik Hidayat Soi, ST,MT

KONVERSI ANALOG KE DIGITAL (ADC)



Konversi analog ke digital (ADC) adalah proses mengubah sinyal analog menjadi sinyal digital. Hal ini dilakukan dengan mem-sampling sinyal analog pada waktu tertentu dan mengukur nilai amplitudo sinyal pada waktu tersebut. Nilai amplitudo kemudian dikodekan menjadi bilangan digital dan disimpan dalam memori komputer atau perangkat digital lainnya.

Berikut adalah langkah-langkah dasar dalam konversi analog ke digital:

1. **Sampling:** Sinyal analog diambil pada interval waktu tertentu dan diubah menjadi sinyal diskrit. Interval waktu antara dua sampel disebut periode sampling dan inversi dari periode sampling adalah frekuensi sampling.



2. Kuantisasi: Setiap nilai amplitudo sinyal yang telah di-sampling dikuantisasi atau diubah menjadi bilangan digital. Proses ini melibatkan pembulatan nilai amplitudo ke nilai terdekat yang sesuai dengan resolusi yang telah ditentukan.
3. Encoding: Bilangan digital yang diperoleh dari proses kuantisasi dikodekan menjadi bentuk yang dapat disimpan dalam memori komputer atau perangkat digital lainnya. Encoding dapat dilakukan dalam berbagai format, seperti format biner, oktal, atau heksadesimal.
4. Filtering: Beberapa jenis filter dapat digunakan untuk menghilangkan noise atau informasi yang tidak diinginkan dari sinyal yang telah dikonversi ke dalam bentuk digital.



Hasil dari proses konversi analog ke digital adalah sinyal digital yang terdiri dari nilai diskrit yang merepresentasikan sinyal analog pada interval waktu tertentu. Konversi analog ke digital sangat penting dalam pemrosesan sinyal digital, seperti pada rekaman audio dan video, pengolahan sinyal medis, pengolahan sinyal radar, dan banyak lagi.



Taufik Hidayat Soi, ST,MT

SAMPLING



Sampling adalah proses pengambilan sampel atau potongan dari sinyal kontinu (analog) pada interval waktu tertentu untuk menghasilkan sinyal diskrit (digital).

Dalam teori sinyal dan sistem, teorema sampling menyatakan bahwa untuk merekonstruksi sinyal kontinu dari sampel diskrit, kita perlu menyimpan informasi pada dua kali frekuensi sinyal asli. Lebih spesifik, jika frekuensi tertinggi pada sinyal asli adalah f_{max} , maka sampling harus dilakukan dengan frekuensi minimal $2f_{max}$ agar dapat merekonstruksi sinyal asli secara akurat.



Rumus untuk menentukan frekuensi sampling minimum adalah:

$$F_s(\text{min}) = 2f_{\text{max}}$$

di mana $F_s(\text{min})$ adalah frekuensi sampling minimum yang dibutuhkan dan f_{max} adalah frekuensi maksimum pada sinyal asli.



- contoh soal analisis sampling:
- Sebuah sinyal analog memiliki frekuensi maksimum sebesar 8000 Hz. Tentukan frekuensi sampling minimum yang dibutuhkan untuk merekonstruksi sinyal tersebut secara akurat.
- Jawab:
- $F_s(\min) = 2f_{\max}$ $F_s(\min) = 2 \times 8000 \text{ Hz}$ $F_s(\min) = 16000 \text{ Hz}$
- Jadi, frekuensi sampling minimum yang dibutuhkan adalah 16000 Hz. Ini berarti bahwa kita perlu mengambil sampel sinyal analog setiap $1/16000$ detik atau setiap 62.5 mikro detik untuk merekonstruksi sinyal asli secara akurat.



- contoh script lengkap untuk proses sampling sinyal analog menjadi sinyal digital di MATLAB:
- % Set parameter sinyal
- $f = 100$; % Frekuensi sinyal analog
- $f_s = 1000$; % Frekuensi sampling
- % Generate sinyal analog
- $t = 0:1/(10*f):1$; % Waktu
- $x_a = \cos(2*\pi*f*t)$; % Sinyal analog



- % Sampling sinyal analog
- `ts = 0:1/fs:1;` % Waktu sampling
- `xs = cos(2*pi*f*ts);` % Sinyal sampel
- % Plot sinyal analog dan sampel
- `subplot(2,1,1)`
- `plot(t,xa)`
- `xlabel('Waktu (s)')`
- `ylabel('Amplitudo')`
- `title('Sinyal Analog')`
- `subplot(2,1,2)`
- `stem(ts,xs)`



- `xlabel('Waktu (s)')`
- `ylabel('Amplitudo')`
- `title('Sinyal Sampel')`
- `% Simpan sinyal sampel dalam file`
- `filename = 'sinyal_sampel.mat';`
- `save(filename, 'xs', 'fs');`



Taufik Hidayat Soi, ST,MT

QUANTIZATION



Quantization adalah proses diskritisasi amplitudo sinyal analog pada saat melakukan konversi dari sinyal analog ke sinyal digital. Quantization biasanya dilakukan setelah proses sampling dalam sistem pemrosesan sinyal digital.

Pada saat quantization, setiap nilai amplitudo sinyal analog diaproksimasi ke level diskrit yang paling dekat dalam suatu rentang yang ditentukan. Semakin banyak level yang digunakan dalam quantization, semakin akurat aproksimasi dari sinyal analog ke sinyal digital yang dihasilkan.

Setelah quantization, nilai amplitudo sinyal digital ditampung dalam bentuk bilangan bulat yang direpresentasikan dalam bentuk bit. Semakin banyak bit yang digunakan dalam representasi, semakin banyak level yang tersedia untuk quantization, sehingga semakin akurat representasi sinyal digital.



- terdapat beberapa rumus yang digunakan dalam proses quantization:

1. Rumus untuk menghitung jarak antara dua level quantization (delta):

- $\text{delta} = (V_{\text{max}} - V_{\text{min}}) / (2^B)$

- dimana:

- V_{max} adalah maksimum tegangan input

- V_{min} adalah minimum tegangan input

- B adalah jumlah bit dalam representasi bilangan bulat



2. Rumus untuk menghitung nilai kuantisasi (Q) pada sinyal analog (V_{in}):

- $Q = \text{delta} * \text{round}(V_{in} / \text{delta})$
- dimana:
- round adalah fungsi pembulatan ke nilai terdekat
- V_{in} adalah nilai tegangan sinyal analog
- Dalam proses quantization, nilai Q akan merepresentasikan nilai amplitudo sinyal digital yang akan disimpan dalam bentuk bilangan bulat pada komputer. Semakin tinggi resolusi quantization (semakin banyak bit yang digunakan), semakin akurat representasi sinyal digital.



- Berikut ini adalah contoh soal mengenai quantization:

Misalkan sebuah sinyal analog memiliki rentang amplitudo 0-10V dan akan direpresentasikan dalam bentuk sinyal digital 8-bit dengan nilai maksimum 255. Tentukan nilai delta dan hitung nilai kuantisasi untuk sinyal analog dengan nilai 6.8V.

1. Menghitung delta: $\text{delta} = (V_{\text{max}} - V_{\text{min}}) / (2^B)$
 $\text{delta} = (10 - 0) / (2^8)$
 $\text{delta} = 0.039$

2. Menghitung nilai kuantisasi: $Q = \text{delta} * \text{round}(V_{\text{in}} / \text{delta})$
 $Q = 0.039 * \text{round}(6.8 / 0.039)$
 $Q = 0.039 * \text{round}(174.35)$
 $Q = 0.039 * 174$
 $Q = 6.786$

- Sehingga nilai kuantisasi untuk sinyal analog dengan nilai 6.8V adalah 6.786. Nilai ini akan direpresentasikan dalam bentuk bilangan bulat pada sinyal digital dengan nilai maksimum 255.



Taufik Hidayat Soi, ST,MT

ENCODING



Encoding adalah proses pengubahan data atau informasi menjadi format yang dapat dibaca oleh komputer atau sistem elektronik lainnya. Contoh encoding yang umum digunakan adalah ASCII (American Standard Code for Information Interchange) dan Unicode.

ASCII adalah encoding 7-bit yang memetakan 128 karakter termasuk huruf, angka, tanda baca, dan karakter khusus ke nilai numerik antara 0 dan 127. Setiap karakter memiliki representasi numeriknya masing-masing dalam ASCII. Misalnya, huruf A memiliki nilai numerik 65, huruf B memiliki nilai numerik 66, dan seterusnya.

Sementara itu, Unicode adalah standar encoding yang lebih luas dan mampu memetakan semua karakter dalam semua bahasa di dunia, termasuk karakter-karakter khusus seperti emotikon. Unicode menggunakan 16-bit atau 32-bit untuk merepresentasikan setiap karakter, sehingga dapat memetakan hingga 1.114.112 karakter unik.



- contoh encoding dalam format teks:

1. Mengubah karakter menjadi bilangan dalam bentuk ASCII:

- H: 72
- a: 97
- l: 108
- o: 111
- ,: 44 (koma)
- space: 32
- d: 100
- u: 117



- n: 110
- i: 105
- !: 33 (seru)

2. Mewakili bilangan dalam bentuk biner:

- 72: 01001000
- 97: 01100001
- 108: 01101100
- 111: 01101111
- 44: 00101100



- 32: 00100000
- 100: 01100100
- 117: 01110101
- 110: 01101110
- 105: 01101001
- 33: 00100001

3. Menggabungkan bilangan biner menjadi satu rangkaian bit:

01001000 01100001 01101100 01101111 00101100 00100000
01100100 01110101 01101110 01101001 01100001 00100001



4. Mewakili rangkaian bit dalam bentuk basis lain seperti basis heksadesimal atau basis 64.

- Heksadesimal: 48616C6F2C2064756E696121
- Basis 64: SGFsbCxkdW5pYSE=

Contoh di atas merupakan contoh encoding teks ke dalam bentuk bit yang sering digunakan dalam komunikasi data dan penyimpanan data di komputer.