Transformada de Fourier

Roberto Cayro Cuadros
Luis Arroyo Pinto

Transformada de Fourier

- Transformar señales entre el dominio del tiempo y el dominio de la frecuencia.

Series de Fourier

1. Amplitude-phase form:

$$s(x) = \sum_{n=1}^{N} A_n cos(\frac{2\pi}{P} nx - \phi_n) + (\frac{A_0}{2})$$

2. Sine-Cosine-phase form:

$$x_k = (\sum_{n=0}^{\infty} a_n cos(\frac{2nx\pi}{P}) + b_n sen(\frac{2nx}{P})) + \frac{a_0}{2}$$

3. Exponential form:

$$x_k = \sum_{n=-N}^{N-1} x_n e^{i2\pi nx/P}$$

Algoritmo de Cooley-Tukey

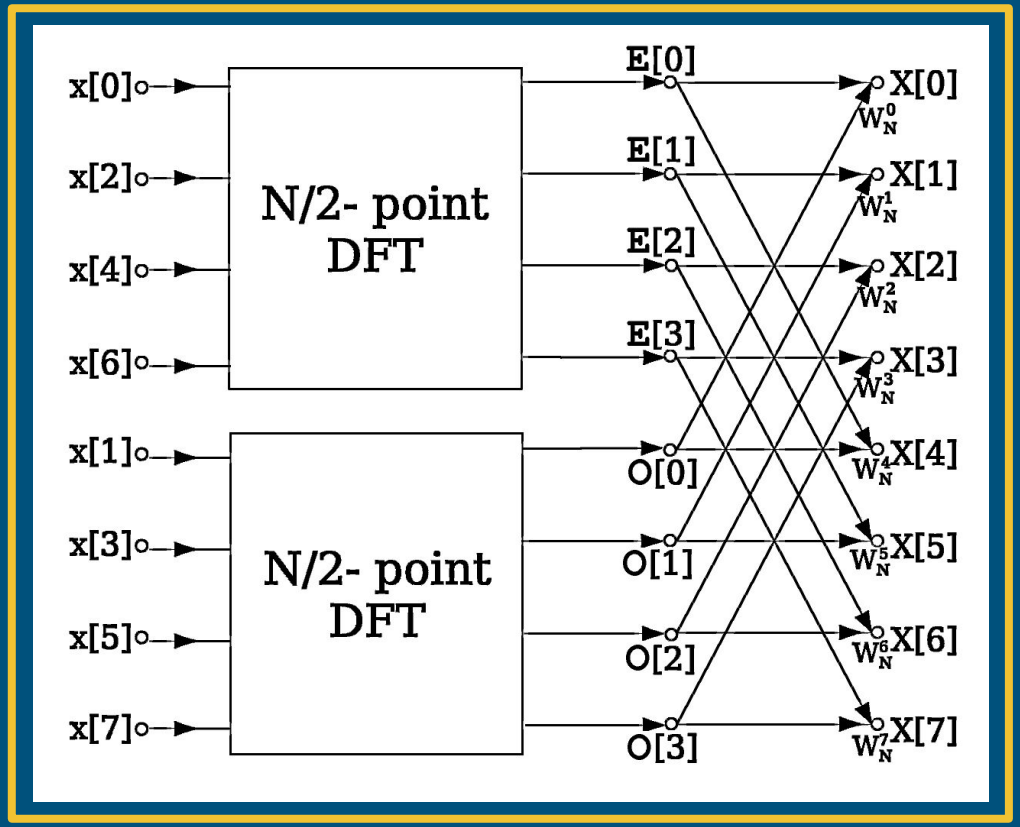
Radix-2 DIT

$$x_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$$

$$x_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-i2\pi(2m)k/N} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-i2\pi(2m+1)n/N}$$

$$X_k = E_k + e^{-i2\pi kn/N} O_k$$

Diagrama de Mariposas



Código

Algoritmo

```
void fft1(int *t, complex<double>* muestras_1, int tam) {


    for (int i = 0; i < tam; i++) {
        muestras_1[i] = complex<double>(t[i], 0);
        muestras_1[i] *= 1;
    }


    fft2(muestras_1,tam);
}
```

```cpp
void fft2(complex<double> *m1, int t) {

    if (t <= 1) {
        return;
    }

    //cout << "t=" << t << endl;
    complex<double> *par = new complex<double>[ t/ 2];
    complex<double> *imp = new complex<double>[ t/ 2];

    for (int i=0; i < t; i++) {
            par[i] = m1[i * 2];
            imp[i] = m1[i * 2 + 1];
    }


    fft2(par, t / 2);
    fft2(imp, t / 2);

    for (int i = 0; i < t / 2; i++) {
        complex<double> x = exp(complex<double> (0, -2 * 3.141592 * i / t)) * imp[i];
        m1[i] = par[i] + x;
        m1[t/2+i] = par[i] - x;
    }

}
```

$$x_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-i2\pi(2m)k/N} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-i2\pi(2m+1)n/N}$$

Gracias