Capitolo 2

Algoritmo Cooley Tukey

Per il dominio del tempo discreto la trasformata di Fourier applicata a una sequenza di dati viene effettuata mediante la formula 2.1

$$X_N(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \tag{2.1}$$

Tale formula può essere riscritta come:

$$X_N(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

dove il fattore W_N prende il nome di "twiddle factor" e può essere inoltre espresso come in 2.2.

$$W_N^m = cos(\frac{2\pi}{N}m) - j \cdot sin(\frac{2\pi}{N}m) \tag{2.2}$$

L'algoritmo 2.1 presentato, risulta essere decisamente lento, in quanto richiede un numero di operazioni dell'ordine di N^2. L'algoritmo di Cooley Tukey permette di minimizzare il numero di operazioni ripetute, scomponendo ricorsivamente la trasformata come somme di trasformate su insiemi più piccoli di valori di ingresso. Fissando $n = rn' + i$, con

$$\begin{cases} 0 \leq n' \leq \frac{N}{r} - 1 \\ 0 \leq i \leq r - 1 \end{cases} \tag{2.3}$$

Si può manipolare la 2.1 ottenendo la formula di Cooley-Tukey in 2.4, dove r è definito "radix", ovvero indica il numero di campioni su cui le sommatorie più interne sono estese.

$$\begin{aligned} X_N(k) = \; & \textstyle\sum_{n'=0}^{\frac{N}{r}-1} \sum_{0}^{r-1} x(rn'+i) W_N^{rn'k} \cdot W_N^{ik} \\ & = \textstyle\sum_{n'=0}^{\frac{N}{r}-1} [\sum_{0}^{r-1} x(rn'+i) W_N^{ik}] W_N^{rn'k} \end{aligned} \tag{2.4}$$

La sostituzione presentata in 2.4 può essere iterata producendo l'effetto definibile come decimazione nel tempo e in frequenza, poichè ogni passo riduce di r volte il numero di campioni N su cui si applica la FFT (le r trasformate ottenute dovranno poi essere sommate).
Nel nostro caso è stato scelto $r = 2$ ovvero la FFT ha radix 2. Seguendo tale decisione la formula 2.4 può essere riscritta come in 2.5.

$$X_N(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)e^{-j\frac{2\pi}{N/2}nk} + e^{-j\frac{2\pi}{N}k} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)e^{-j\frac{2\pi}{N/2}nk} \tag{2.5}$$

La formula 2.5 rende quindi possibile esprimere la trasformata scomponendola recursivamente su una somma di due trasformate fatte su metà dei campioni.
Ipotizzando un caso d'esempio di $N = 4$ campioni si otterrà esplicitando la 2.5:

$$\begin{array}{ll} X_4(0) & = [x(0)W_4^0 + x(2)W_4^0] + W_4^0[x(1)W_4^0 + x(3)W_4^0] \\ X_4(1) & = [x(0)W_4^0 + x(2)W_4^2] + W_4^1[x(1)W_4^0 + x(3)W_4^2] \\ X_4(2) & = [x(0)W_4^0 + x(2)W_4^4] + W_4^2[x(1)W_4^0 + x(3)W_4^4] \\ X_4(3) & = [x(0)W_4^0 + x(2)W_4^6] + W_4^3[x(1)W_4^0 + x(3)W_4^6] \end{array} \tag{2.6}$$

Come si può vedere dalla 2.6 i campioni pari (nel nostro caso $x(0)$ e $x(2)$) e quelli dispari ($x(1)$ e $x(3)$) presentano gli stessi twiddle factor (W). In particolare considerando che i twiddle factor sono periodici, con periodo N, si nota che le operazioni interne alle parentesi quadre, sono ripetute identicamente per la seconda metà dei campioni. Ad una ulteriore analisi i twiddle factor necessari all'elaborazione sono riducibili a $\frac{N}{2}$, in quanto siccome giacciono tutti sul cerchio a modulo unitario vale preso $W^k = W_r + jW_i$ con $0 \leq k \leq \frac{N}{2} - 1$, allora $W^{k+\frac{N}{2}} = -W_r - jW_i$, dove W_r e W_i indicano rispettivamente la parte reale e immaginaria del valore considerato. Tali fatti ci permettono di rappresentare la singola operazione svolta all'interno delle parentesi (o considerando le parentesi come un unico valore) come in figura 2.1.

Tale operazione atomica, chiamata farfalla (o butterfly), necessita quindi di soli tre dati. Ripetendo questa unità di base, opportunamente interconnessa come in 2.2, ci permette di rappresentare graficamente tutte le operazioni necessarie per passare dal set di dati in ingresso alla corretta trasformata discreta. L'elaborazione che quindi si ottiene, sfruttando l'algoritmo proposto, avviene in $log_2(N)$ passi; i risultati intermedi sono gli unici che servono per il passo successivo e sono in numero pari ai dati di ingresso. Questo tipo di algoritmo sarà quindi working in place, e tale fatto ci garantisce l'utilizzo di una quantità di memoria necessaria fissata.
Definendo tutte le butterfly che lavorano sugli stessi twiddle factor appartenenti allo stesso gruppo, si nota come a ogni passo dell'elaborazione si ricominci l'ordine dei twiddle factor secondo il "bit reverse order" e che si cresca per ogni passo di un numero pari ai gruppi.

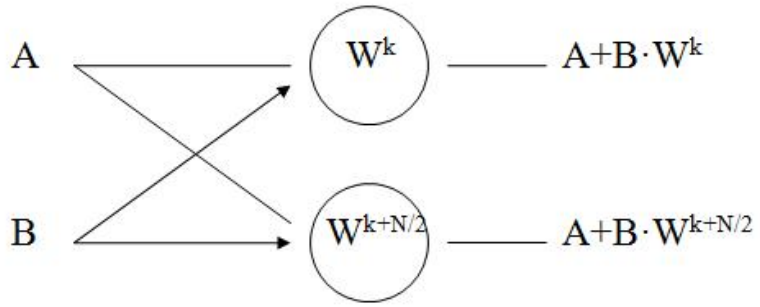

Figura 2.1. *Signal Graph della farfalla (butterfly) radix 2.*

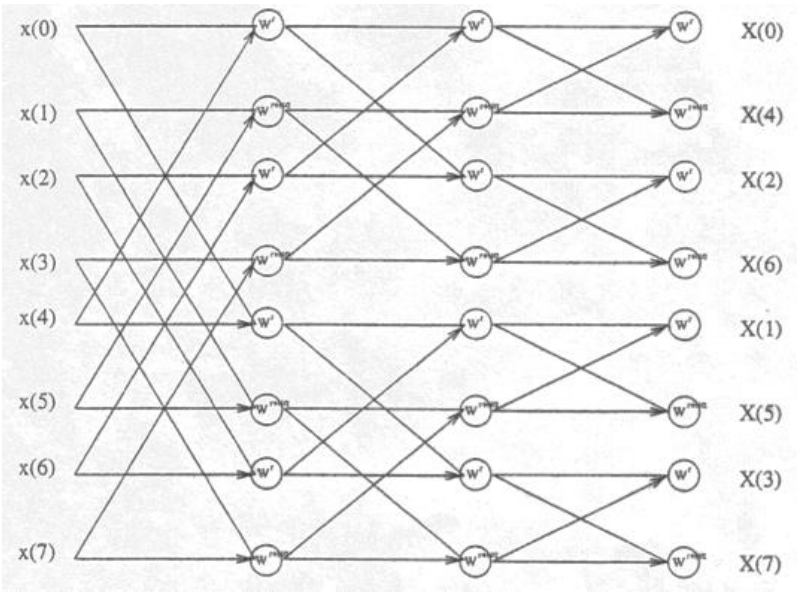

Figura 2.2. *Signal Graph con N=8.*

Il bit reverse order è l'ordine di numeri che si ottiene sommando un MSB alla codifica binaria di un dato numero e propagando i riporti dal bit più significativo verso quello meno significativo. Questa operazione equivale a una somma dei bit invertiti del numero di partenza con 1, e quindi il risultato binario viene nuovamente invertito.
Le uscite al termine dei passi saranno nuovamente da ordinare con il bit reverse.

2.0.1 pseudocodice risultante

Algoritmo FFT secondo Cooley Tukey

```
#passi = log(N)/log(2)
#gruppi = 1
#farfalle= N/2;
  W_pointer=0000
DO #passi
    DO #gruppi
        DO #farfalle
            BUTTERFLY_Alg(W_pointer)
        END farfalle
    W_pointer<-[W_pointer+N/4]%reverse carry
    END gruppi
#gruppi=#gruppi*2
#farfalle=#farfalle/2
W_pointer=0000
#passi=#passi-1
END passi
```

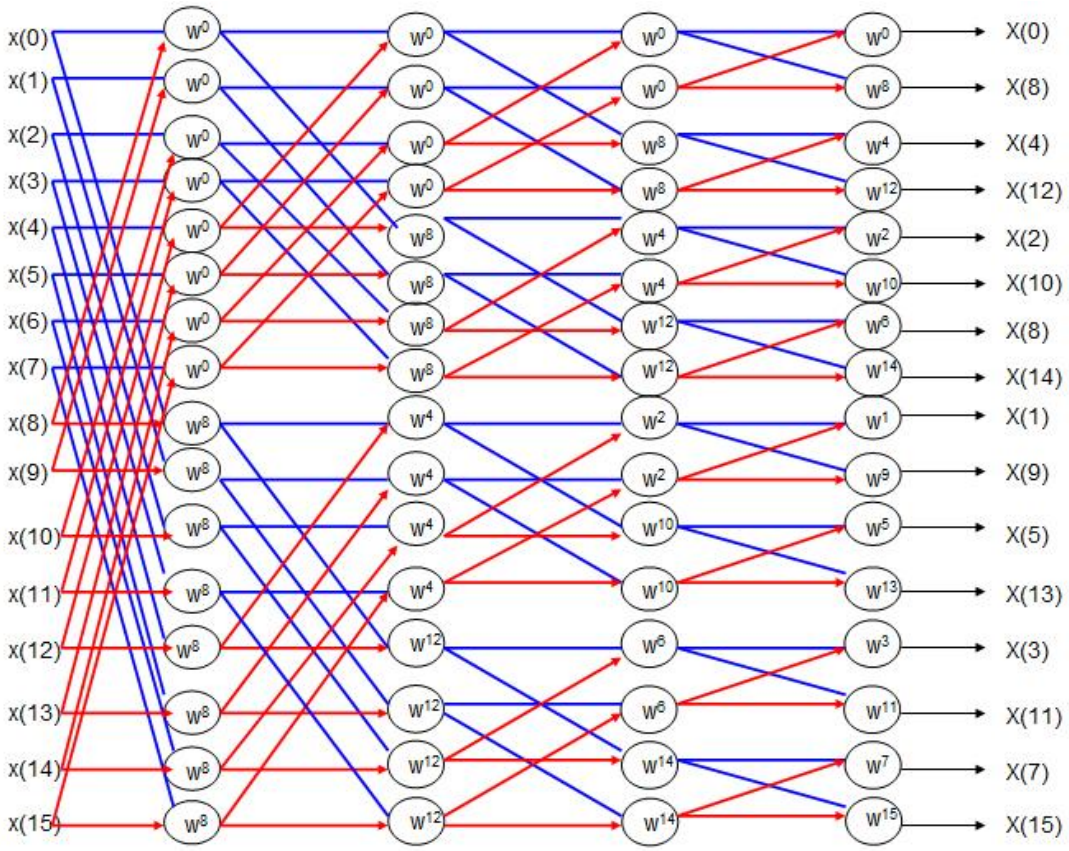

Figura 2.2. *Schema dell'algoritmo di Cooley Tukey con N=16*

Wn	$\cos(\frac{2\pi}{N}m) - j\sin(\frac{2\pi}{N}m)$	$\Re$	$\Im$
W0	$\cos(\frac{2\pi}{8}0) - j\sin(\frac{2\pi}{8}0)$	+1 011111111111	0 000000000000
W1	$\cos(\frac{2\pi}{8}1) - j\sin(\frac{2\pi}{8}1)$	0.92387953251129 011101100100	-0.38268343236509 110011110001
W2	$\cos(\frac{2\pi}{8}2) - j\sin(\frac{2\pi}{8}2)$	0.70710678118655 010110101000	-0.70710678118655 101001011000
W3	$\cos(\frac{2\pi}{8}3) - j\sin(\frac{2\pi}{8}3)$	0.38268343236509 001100001111	-0.92387953251129 100010011100
W4	$\cos(\frac{2\pi}{8}4) - j\sin(\frac{2\pi}{8}4)$	0 000000000000	-1 100000000000
W5	$\cos(\frac{2\pi}{8}5) - j\sin(\frac{2\pi}{8}5)$	-0.38268343236509 110011110001	-0.92387953251129 100010011100
W6	$\cos(\frac{2\pi}{8}6) - j\sin(\frac{2\pi}{8}6)$	-0.70710678118655 101001011000	-0.70710678118655 101001011000
W7	$\cos(\frac{2\pi}{8}7) - j\sin(\frac{2\pi}{8}7)$	-0.92387953251129 100010011100	-0.38268343236509 110011110001

Figura 2.3. *Schema dell'algoritmo di Cooley Tukey con N=16*