

Appendice B

FAST FOURIER TRASFORM-FFT

La trasformata discreta di Fourier svolge un ruolo molto importante nello studio, nell'analisi e nell'implementazione di algoritmi dei segnali in tempo discreto. Come si è visto nel capitolo precedente, è molto conveniente studiare i sistemi nel dominio di Fourier. La trasformata discreta di Fourier campiona ad intervalli regolari la trasformata di Fourier del segnale ogni $\omega_k = 2\pi k/N$, questo implica calcolare N punti della trasformata discreta di Fourier. A livello computazionale queste operazioni potrebbero risultare molto pesanti, infatti se ad esempio abbiamo N campioni, la complessità di calcolo della trasformata discreta di Fourier è di tipo quadratico nell'ordine N. Per questo è nata la necessità di avvalersi di algoritmi con complessità minori, quali la FFT. In realtà, l'acronimo FFT indica una classe di algoritmi efficienti per il calcolo della trasformata discreta di una sequenza periodica. Alcuni di questi algoritmi sono Goertzel, quello di "*Decimation in time*".

B.1 Trasformata di Fourier Discreta

Nel capitolo precedente è stata trattata la rappresentazione di sequenze, si può vedere come nel caso in cui la sequenza da rappresentare è di durata finita, cioè ha soltanto un numero finito di valori non nulli, è possibile sviluppare una rappresentazione di Fourier alternativa, chiamata *trasformata di Fourier discreta* (DFT). La DFT è una rappresentazione di Fourier di una sequenza di lunghezza finita cioè che è essa stessa una sequenza anziché una funzione continua, e corrisponde a campioni egualmente spaziatati in frequenza della trasformata di Fourier del segnale.

B.1.1 Rappresentazione di sequenze periodiche – la serie di Fourier discreta

Si consideri una sequenza $\tilde{x}(n)$ periodica⁸ con periodo N , cioè tale che sia $\tilde{x}(n) = \tilde{x}(n + kN)$ per ogni valore intero di k . E' possibile rappresentare $\tilde{x}(n)$ per mezzo di una serie di Fourier, cioè come somma di sequenze sinusoidali o cosinusoidali o, in modo equivalente, di sequenze esponenziali complesse con frequenze multipli interi della frequenza fondamentale $\frac{2\pi}{N}$ associata alla sequenza periodica. A differenza della serie di Fourier valida per funzioni periodiche continue, esistono soltanto N esponenziali complessi distinti il cui periodo è un sottomultiplo intero del periodo fondamentale N . Ciò deriva dal fatto che l'esponenziale complesso

$$e_k(n) = e^{j\left(\frac{2\pi}{N}\right)nk} \quad (\text{B.1})$$

è periodico in k con periodo N . Perciò $e_0(n) = e_N(n), e_1(n) = e_{N+1}(n)$ ecc., e quindi l'insieme di N esponenziali complessi rappresentati nella 2.1 con $k=0,1,2,\dots,N-1$ definisce tutti gli esponenziali complessi distinti con frequenze che sono multipli interi di $\frac{2\pi}{N}$. Pertanto, per la rappresentazione in serie di Fourier di una sequenza periodica, $\tilde{x}(n)$, bastano soltanto N di questi esponenziali complessi e quindi essa può scriversi nella forma

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{j\frac{2\pi}{N}nk} \quad (\text{B.2})$$

La costante moltiplicativa $\frac{1}{N}$ è stata inserita per convenienza e non ha nessun effetto importante sulla natura della rappresentazione. I coefficienti $\tilde{X}(k)$ dalla sequenza periodica $\tilde{x}(n)$ si ottengono:

⁸ Il simbolo \sim indica sequenze periodiche, è importante per distinguere in seguito le sequenze periodiche da quelle aperiodiche

$$\frac{1}{N} \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}nr} = \begin{cases} 1, & \text{per } r = mN, m \text{ intero} \\ 0, & \text{altrove} \end{cases} \quad (\text{B.3})$$

Perciò moltiplicando entrambi i membri della relazione (B.2) per $e^{j\frac{2\pi}{N}nr}$ e sommando da $n=0$ a $n=N-1$, otteniamo

$$\frac{1}{N} \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}nr} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \tilde{X}(k) e^{j(\frac{2\pi}{N}n)(k-r)}$$

$$\sum_{n=0}^{N-1} \tilde{x}(n) e^{j\frac{2\pi}{N}nr} = \sum_{k=0}^{N-1} \tilde{X}(k) \left[\frac{1}{N} \sum_{n=0}^{N-1} e^{j(\frac{2\pi}{N})(k-r)n} \right]$$

per cui usando la (2.3) risulta

$$\sum_{n=0}^{N-1} \tilde{x}(n) e^{j\frac{2\pi}{N}nr} = \tilde{X}(r)$$

Perciò i coefficienti $\tilde{X}(k)$ nella (B.2) sono dati da

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\frac{2\pi}{N}nk} \quad (\text{B.4})$$

La sequenza $\tilde{X}(k)$ rappresentata dalla relazione (B.4) è periodica con periodo N , cioè $\tilde{X}(0) = \tilde{X}(N)$, $\tilde{X}(1) = \tilde{X}(N+1)$ ecc. Naturalmente, ciò è in accordo col fatto che gli esponenziali complessi rappresentati nell'espressione (B.1) sono distinti soltanto per

$k=0,1,2,\dots,N-1$, e perciò nella rappresentazione di una sequenza periodica in serie di Fourier possono esservi solo N coefficienti distinti.

I coefficienti della serie di Fourier possono essere considerati come una sequenza di lunghezza finita, data dall'espressione (B.4) per $k=0,1,2,\dots,N-1$ e zero per valori diversi di k , o come una sequenza periodica definita per ogni k dalla relazione (B.4). Le due interpretazioni sono equivalenti. In generale è più conveniente interpretare i coefficienti della serie di Fourier $\tilde{X}(k)$ come una sequenza periodica. Le relazioni (B.2) e (B.4) possono essere considerate una coppia di trasformate e costituiscono la rappresentazione di una sequenza periodica in serie di Fourier discreta (DFS). Per comodità di rappresentazione queste espressioni saranno generalmente scritte in termini di W_N definito come

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$

Quindi le formule (di analisi e sintesi) della DFS si scrivono come

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) W_N^{nk} \quad (\text{B.4})$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-nk} \quad (\text{B.5})$$

dove sia $\tilde{X}(k)$ che $\tilde{x}(n)$ sono sequenze periodiche.

B.1.2 Rappresentazione di Fourier per sequenze di durata finita – la trasformata di Fourier discreta

Nel paragrafo precedente si è considerata la rappresentazione di sequenze periodiche in termini della serie di Fourier discreta. La stessa rappresentazione può essere

applicata a sequenze di durata finita, purché la si interpreti correttamente. La rappresentazione di Fourier che ne risulta verrà indicata come la trasformata di Fourier discreta (DFT).

Consideriamo una sequenza di durata finita $x(n)$ di lunghezza N in modo che $x(n)=0$ eccetto nell'intervallo $0 \leq n \leq (N-1)$. Anche se la sequenza è di lunghezza M minore di N , può essere considerata di lunghezza N con gli ultimi $(N-M)$ punti dell'intervallo aventi valore zero. la corrispondente sequenza periodica di periodo N , di cui $x(n)$ è un periodo, sarà indicata $\tilde{x}(n)$ ed è data da

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n+rN) \quad (\text{B.6a})$$

Poiché $x(n)$ è di lunghezza finita N , non vi è sovrapposizione tra i termini $x(n+rN)$ per valori di r differenti. Perciò la relazione (B.6 a) può essere scritta nella forma alternativa

$$\tilde{x}(n) = x(n \text{ modulo } N) \quad (\text{B.6 b})$$

La sequenza di durata finita $x(n)$ si ricava da $\tilde{x}(n)$ estraendone un periodo, cioè

$$x(n) = \begin{cases} \tilde{x}(n), & 0 \leq n \leq (N-1) \\ 0, & \text{altrove} \end{cases}$$

L'impulso rettangolare discreto $R_N(n)$ è definito come:

$$R_N(n) = \begin{cases} \tilde{x}(n), & 0 \leq n \leq (N-1) \\ 0, & \text{altrove} \end{cases}$$

La relazione precedente può essere espressa come:

$$x(n) = \tilde{x}(n)R_N(n) \quad (\text{B.7})$$

Da quanto visto nel paragrafo precedente, la relazione tra $\tilde{X}(k)$ e $\tilde{x}(n)$ è;

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

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

Poiché le somme nelle due relazioni precedenti (B.4 e B.5) riguardano solo l'intervallo tra 0 e (N-1), segue:

$$X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n) W_N^{nk} & 0 \leq k \leq N-1 \\ 0, & \text{altrove} \end{cases} \quad (\text{B.8})$$

$$x(n) = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} & 0 \leq n \leq N-1 \\ 0, & \text{altrove} \end{cases} \quad (\text{B.9})$$

La coppia di trasformate (B.8) e (B.9) è chiamata trasformata discreta di Fourier (DFT), la (B.8) è l'equazione di analisi e la B.9 è l'equazione di sintesi della sequenza $x(n)$.

B.2 Algoritmo FFT

Nel paragrafo precedente si è vista la trasformata discreta di Fourier, questa ricopre un ruolo importante nell'analisi, nel progetto e nella realizzazione di algoritmi e sistemi di elaborazione numerica dei segnali. Nelle relazioni (B.8) e (B.9) sia $x(n)$ che $X(k)$ possono essere complesse. Poiché $x(n)$ è in generale complessa si può scrivere

$$X(k) = \sum_{n=0}^{N-1} \{ (\operatorname{Re}[x(n)]\operatorname{Re}[W_N^{kn}] - \operatorname{Im}[x(n)]\operatorname{Im}[W_N^{kn}]) + j(\operatorname{Re}[x(n)]\operatorname{Im}[W_N^{kn}] + \operatorname{Im}[x(n)]\operatorname{Re}[W_N^{kn}]) \},$$

$$k = 0, 1, \dots, N-1 \quad (2.10)$$

Dall'espressione precedente risulta che il calcolo diretto di $X(k)$ richiede $4N$ moltiplicazioni reali e $(4N-2)$ addizioni reali per ogni valore di k . Poiché occorre valutare $X(k)$ per N diversi valori di k , il calcolo diretto della trasformata di Fourier discreta di una sequenza $x(n)$ richiede $4N^2$ moltiplicazioni reali e $N(4N-2)$ addizioni reali ovvero, in altri termini, N^2 moltiplicazioni complesse e $N(N-1)$ addizioni complesse. Oltre alle moltiplicazioni e alle addizioni contenute nella (B.10) l'esecuzione del calcolo della DFT su un calcolatore numerico d'impiego generale o con dispositivo ad essa dedicato implica la necessità di memorizzare e di leggere i valori della sequenza d'ingresso $x(n)$ e dei coefficienti W_N^{kn} . Poiché negli algoritmi di calcolo numerico la quantità di operazioni di lettura e scrittura è proporzionale al numero di operazioni aritmetiche, il numero di moltiplicazioni e addizioni sono una misura significativa della complessità dell'algoritmo (tempo richiesto per eseguire un algoritmo di calcolo). Quindi, per il calcolo diretto della trasformata di Fourier discreta, l'efficienza del metodo si può valutare sulla base del fatto che sono necessarie $4N^2$ moltiplicazioni reali e $N(4N-2)$ addizioni reali, Poiché la quantità (e quindi il tempo) dei calcoli è approssimativamente proporzionale a N^2 , per valori grandi di N il numero di operazioni per calcolare la DFT diventa enorme.

La maggior parte delle tecniche usate per migliorare l'efficienza del calcolo della DFT sfruttano una delle seguenti proprietà delle quantità W_N^{kn} :

$$1. W_N^{k(N-n)} = (W_N^{kn})^*$$

$$2. W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$$

Per esempio sfruttando la prima proprietà, cioè la simmetria delle funzioni seno e coseno, si possono nella (B.10) raggruppare dei termini come

$$\begin{aligned} & \operatorname{Re}[x(n)]\operatorname{Re}[W_N^{kn}] + \operatorname{Re}[x(N-n)]\operatorname{Re}[W_N^{k(N-n)}] = \\ & = (\operatorname{Re}[x(n)] + \operatorname{Re}[x(N-n)])\operatorname{Re}[W_N^{kn}] \end{aligned}$$

e

$$- \operatorname{Im}[x(n)]\operatorname{Im}[W_N^{kn}] - \operatorname{Im}[x(N-n)]\operatorname{Im}[W_N^{k(N-n)}] = -(\operatorname{Im}[x(n)] - \operatorname{Im}[x(N-n)])\operatorname{Im}[W_N^{kn}]$$

Analoghi raggruppamenti si possono fare per gli altri termini della (B.10). In questo modo si riesce a ridurre il numero di moltiplicazioni circa di un fattore 2. Si può anche sfruttare il fatto che per certi valori del prodotto kn le funzioni seno e coseno valgono 1 e 0, e non è quindi necessario eseguire moltiplicazioni corrispondenti. Tuttavia, dopo riduzioni di questo genere, resta sempre un numero esiguo di calcoli da effettuare dell'ordine di N^2 . Usando la seconda proprietà, cioè la periodicità della sequenza complessa W_N^{kn} , si ottiene una riduzione dei calcoli notevolmente maggiore.

Algoritmi di calcolo che sfruttano sia la simmetria che la periodicità della sequenza W_N^{kn} erano noti già da molto tempo prima dell'avvento dei calcolatori numerici veloci. Runge e più tardi Danielson e Lanczos hanno descritto algoritmi la cui complessità era all'incirca proporzionale a $N \log N$ invece che a N^2 . Questa differenza non era però di grande importanza, la possibilità di ridurre notevolmente il tempo di calcolo si ebbe soltanto verso il 1965, quando Cooley e Tukey pubblicarono un algoritmo per il calcolo della trasformata di Fourier discreta che vale quando N è un numero composto, cioè il prodotto di due o più interi. La pubblicazione di questo lavoro provocò un fiorire di applicazioni della trasformata di Fourier discreta all'elaborazione dei segnali ed diede luogo alla scoperta di numerosi algoritmi di calcolo che divennero noti come *algoritmi per la trasformata di Fourier veloce* o semplicemente FFT.

Il principio fondamentale su cui si basano tutti questi algoritmi è la scomposizione del calcolo della trasformata di Fourier discreta di una sequenza lunga N in trasformate di Fourier discrete di dimensioni via via più piccole. Il modo in cui questo principio è

applicato dà luogo a una varietà di algoritmi diversi, tutti caratterizzati da miglioramenti circa della stessa entità nella velocità di calcolo. Si esamineranno due classi di algoritmi, la prima, detta a *decimazione nel tempo*, la seconda, detta a *decimazione in frequenza*. La prima prende il nome dal fatto che nello scomporre il calcolo in trasformate di dimensioni più piccole, la sequenza $x(n)$ viene suddivisa in sequenze sempre più corte. Nella seconda classe, la sequenza che viene scomposta in sottosequenze sempre più corte è quella dei coefficienti della trasformata di Fourier discreta. Saranno esposti diversi tipi di algoritmi con diverse complessità, il primo tra questi sarà l'algoritmo di Goertzel con complessità proporzionale a N^2 .

B.2.1 Algoritmo Goertzel

L'algoritmo di Goertzel è un procedimento più efficiente del metodo diretto ed è un esempio di come si possa sfruttare la periodicità della sequenza W_N^{kn} per ridurre i calcoli. Più precisamente, vedremo che la trasformata di Fourier discreta può essere considerata come la risposta di un filtro numerico la cui struttura può essere progettata in modo da ridurre il numero delle operazioni aritmetiche. Per ricavare l'algoritmo di Goertzel:

$$W_N^{-kn} = e^{j\left(\frac{2\pi}{N}\right)Nk} = e^{j2\pi k} = 1 \quad (\text{B.11})$$

Questa è una conseguenza immediata della periodicità di W_N^{kn} . In base alla (B.11) possiamo moltiplicare il secondo membro della relazione (B.8) per W_N^{-kN} senza alterare l'uguaglianza. Quindi

$$\begin{aligned} X(k) &= W_N^{-kN} \sum_{r=0}^{N-1} x(r) W_N^{rk} = \\ &= \sum_{r=0}^{N-1} x(r) W_N^{-k(N-r)} \end{aligned} \quad (\text{B.12})$$

Introducendo, per comodità, la sequenza

$$y_k(n) = \sum_{r=0}^{N-1} x(r)W_N^{-k(n-r)} \quad (\text{B.13})$$

Dalle (B.12) e (B.13) segue che

$$X(k) = y_k(n) |_{n=N}$$

La relazione (B.13) è la convoluzione discreta della sequenza di durata finita $x(n)$, $0 \leq n \leq (N-1)$, con la sequenza W_N^{-kn} . Di conseguenza, $y_k(n)$ può essere vista come la risposta di un sistema con risposta all'impulso W_N^{-kn} a un ingresso $x(n)$. In particolare, $X(k)$ è il valore per $n=N$. Un sistema con risposta all'impulso W_N^{-kn} è rappresentato in fig. 2.1

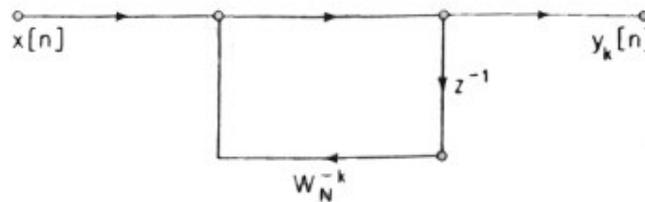


Figura 2.1 Grafo di flusso per un sistema del primo ordine per il calcolo ricorsivo di $X(k)$

Poiché sia l'ingresso $x(n)$ che il coefficiente W_N^{-k} sono complessi, il calcolo di ogni nuovo valore di $y_k(n)$ richiede 4 moltiplicazioni reali e quattro addizioni reali. Siccome poi occorre calcolare tutti i valori intermedi $y_k(1), y_k(2), \dots, y_k(N-1)$ per

ottenere , l'uso dello schema in fig. 2-1 richiede $4N$ moltiplicazioni reali e $4N$ addizioni reali per ricavare $X(k)$ per un particolare valore di k . Perciò questo schema è lievemente meno efficiente del metodo diretto L'aspetto positivo dello schema è che non richiede né la memorizzazione né il calcolo dei coefficienti W_N^{-kn} , in quanto questi vengono calcolati attraverso il procedimento ricorsivo.

E' possibile mantenere questa semplificazione e ridurre la complessità. di un fattore 2. La funzione di trasferimento del sistema di fig.2.1 è

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \quad (\text{B.14})$$

Moltiplicando numeratore e denominatore per il fattore $(1 - W_N^k z^{-1})$, si ottiene

$$\begin{aligned} H_k(z) &= \frac{1 - W_N^k z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^k z^{-1})} \quad (\text{B.15}) \\ &= \frac{1 - W_N^k z^{-1}}{1 - 2 \cos\left(\frac{2\pi}{N} k\right) z^{-1} + z^{-2}} \end{aligned}$$

Per realizzare i poli del sistema corrispondente alla (B.15) sono necessarie solo due moltiplicazioni, in quanto i coefficienti sono reali e il coefficiente (-1) non deve essere contato come moltiplicazione; il numero di addizioni è invece sempre quattro. Poiché basta portare il sistema in uno stato in cui sia possibile calcolare $y_k(n)$, la moltiplicazione complessa per $-W_N^k$, corrispondente allo zero, non deve essere eseguita ad ogni iterazione dell'equazione alle differenze, ma solo dopo l'N-esima. Quindi l'ammontare totale di operazioni è di $2N$ moltiplicazioni reali e $4N$ addizioni reali per i poli, più quattro moltiplicazioni reali e quattro addizioni reali per lo zero. Il peso complessivo dei calcoli è di $2(N + 2)$ moltiplicazioni reali e $4(N + 1)$ addizioni reali, cioè circa la metà del numero di moltiplicazioni reali richieste per il metodo

diretto. Purtroppo usando l'algoritmo di Goertzel, per il calcolo di tutti gli N punti la complessità è ancora proporzionale a N^2 come per il calcolo diretto.

Nel metodo diretto o in quello di Goertzel non occorre ricavare tutti gli N diversi valori di $X(k)$. Anzi, in generale si può valutare $X(k)$ su M qualsiasi valori di K . In questo caso il peso totale dei calcoli è proporzionale a MN . Queste tecniche sono convenienti quando M è piccolo; sono però disponibili algoritmi più sofisticati per i quali il numero proporzionale di operazioni è proporzionale a $N \log_2 N$ quando N è una potenza di 2. Quindi, quando M è minore di $\log_2 N$, il metodo di Goertzel o quello diretto sono più efficienti.

B.2.2 Algoritmi di FFT basati sulla decimazione nel tempo

Per ottenere un aumento di efficienza è necessario scomporre il calcolo della DFT in calcoli di DFT di dimensioni sempre più piccole. Si sfrutta sia la simmetria che la periodicità dell'esponenziale complesso $W_N^{kn} = e^{-j(\frac{2\pi}{N})nk}$. Gli algoritmi nei quali il procedimento di scomposizione si attua suddividendo la sequenza $x(n)$ in sottosequenze via via più piccole, si chiamano *algoritmi a decimazione nel tempo*.

Consideriamo il caso particolare di N potenza intera di 2, cioè

$$N = 2^v$$

Poiché N è un intero pari, possiamo pensare di calcolare $X(k)$ dividendo $x(n)$ in due sequenze di $\frac{N}{2}$ punti (campioni) ciascuna, costituite l'una dai punti che hanno indice pari in $x(n)$, e l'altra da quelli con indice dispari. Se nell'espressione di $X(k)$

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0,1,\dots,N-1 \quad (\text{B.16})$$

$$X(k) = \sum_{n \text{ pari}} x(n)W_N^{nk} + \sum_{n \text{ dispari}} x(n)W_N^{nk}$$

ponendo $n=2r$ per n pari e $n=2r+1$ per n dispari,

$$\begin{aligned} X(k) &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)(W_N^2)^{rk} \end{aligned} \quad (\text{B.17})$$

Ma è $W_N^2 = W_{\frac{N}{2}}$ in quanto

$$W_N^2 = e^{-2j\left(\frac{2\pi}{N}\right)} = e^{-j\frac{2\pi}{\frac{N}{2}}} = W_{\frac{N}{2}}$$

Di conseguenza la (2.17) può essere riscritta come

$$\begin{aligned} X(k) &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{\frac{N}{2}}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_{\frac{N}{2}}^{rk} \\ &= G(k) + W_N^k H(k) \end{aligned} \quad (\text{B.18})$$

Questa espressione è composta da due sommatorie entrambe da $\frac{N}{2}$ punti, la prima sommatoria è una DFT lunga $\frac{N}{2}$ dei punti con indice pari della sequenza originaria. la seconda $\frac{N}{2}$ dei punti con indice dispari. Anche se l'indice k può assumere N valori,

$k=0,1,\dots,N-1$, occorre calcolare ogni somma solo per k tra 0 e $\frac{N}{2}-1$, in quanto sia $G(k)$ che $H(k)$ sono periodiche in k con periodo $\frac{N}{2}$. Dopo che sono state calcolate le due DFT corrispondenti alle due sommatorie della (2.18), esse devono essere combinate per ottenere la DFT su N punti, $X(k)$. La fig. 2.2 illustra il tipo di calcoli richiesto per ottenere $X(k)$ in base alle (2.18) per una sequenza di 8 punti, cioè per $N=8$. In questa figura, i rami che entrano in un nodo si sommano per produrre la variabile nodo. Quando manca l'indicazione del coefficiente di trasmissione del ramo, il coefficiente è unitario, negli altri rami il coefficiente di trasmissione è una potenza intera di W_N .

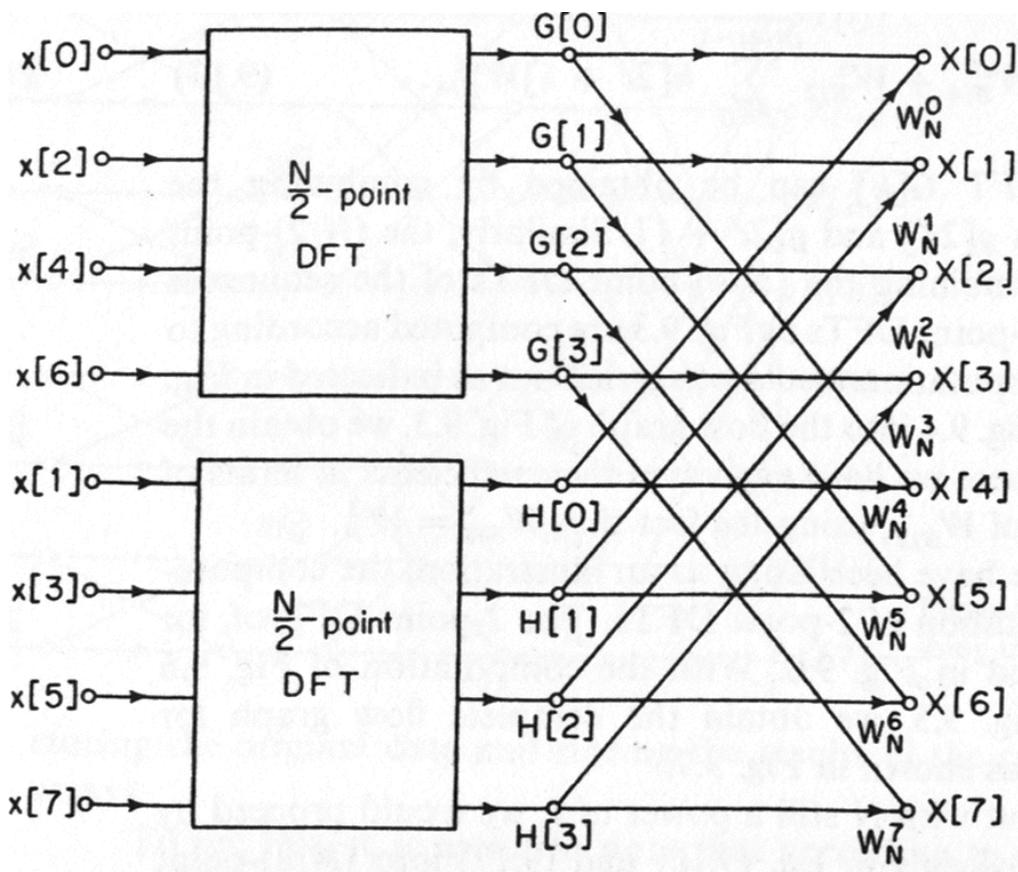


Figura 2.2 Grafo di flusso per la scomposizione del calcolo di una DFT di $N/2$ punti, con il metodo della decimazione nel tempo

Dalla fig. 2.2. si vede che vengono calcolate due DFT di 4 punti: $G(k)$ indica la DFT lunga quattro dei punti con indice pari e $H(k)$ indica la DFT lunga quattro dei punti dispari. $X(0)$ si ottiene poi moltiplicando $H(0)$ per W_N^0 e sommando il risultato a $G(0)$. $X(1)$ si ottiene moltiplicando $H(1)$ per W_N^1 e sommando il risultato a $G(1)$. $X(4)$ si ottiene moltiplicando $H(4)$ per W_N^4 e sommando il risultato a $G(4)$. Siccome però $G(k)$ e $H(k)$ sono entrambe periodiche in k con periodo 4, risulta $H(4)=H(0)$ e $G(4)=G(0)$. Perciò $X(4)$ si ottiene moltiplicando $H(0)$ per W_N^4 e sommando il risultato a $G(0)$.

La B.18 richiede il calcolo di due DFT su $\frac{N}{2}$ che a sua volta richiede $2\left(\frac{N}{2}\right)^2$ moltiplicazioni complesse e circa $2\left(\frac{N}{2}\right)^2$ addizioni complesse. Le due DFT su $\frac{N}{2}$ punti devono poi essere combinate come indicato 2.18, il che richiede altre N moltiplicazioni complesse, corrispondenti al prodotto della seconda sommatoria per W_N^k , ed altre N addizioni complesse, corrispondenti alla somma di quel prodotto con la prima sommatoria. Di conseguenza il calcolo della 2.18 per tutti i valori di k richiede $N + 2\left(\frac{N}{2}\right)^2$, cioè $N + \frac{N^2}{2}$, moltiplicazioni complesse e addizioni complesse.

L'espressione (B.18) corrisponde a spezzare il calcolo originario su N punti in due calcoli su $\frac{N}{2}$ punti. Se $\frac{N}{2}$ è pari allora il calcolo di ciascuna DFT su $\frac{N}{2}$ punti nella B.18 si può effettuare mediante il calcolo e la successiva combinazione di due DFT su $\frac{N}{4}$ punti. Pertanto $G(k)$ e $H(k)$ della B.18 verrebbero calcolate così:

$$G(k) = \sum_{r=0}^{\frac{N}{2}-1} g(r)W_{\frac{N}{2}}^{rk} = \sum_{l=0}^{\frac{N}{4}-1} g(2l)W_{\frac{N}{2}}^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} g(2l+1)W_{\frac{N}{2}}^{(2l+1)k}$$

oppure

$$G(k) = \sum_{l=0}^{\frac{N}{4}-1} g(2l)W_{\frac{N}{4}}^{lk} + W_{\frac{N}{2}}^k \sum_{l=0}^{\frac{N}{4}-1} g(2l+1)W_{\frac{N}{4}}^{lk} \quad (B.19)$$

Analogamente

$$H(k) = \sum_{l=0}^{\frac{N}{4}-1} h(2l)W_{\frac{N}{4}}^{lk} + W_{\frac{N}{2}}^k \sum_{l=0}^{\frac{N}{4}-1} h(2l+1)W_{\frac{N}{4}}^{lk} \quad (B.20)$$

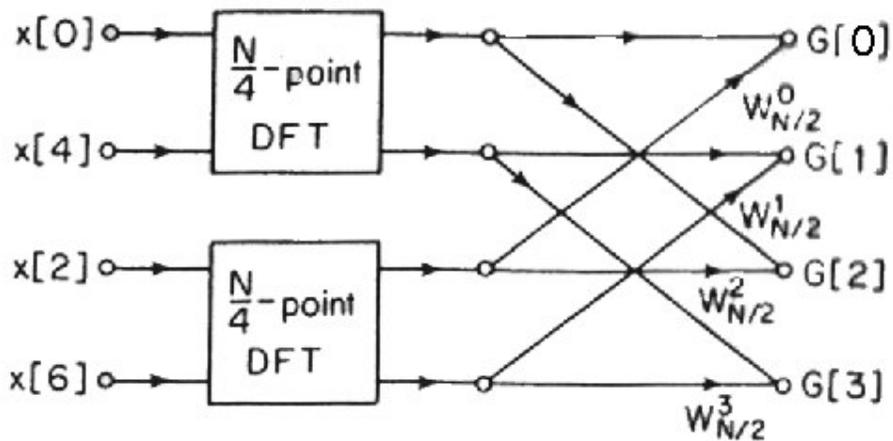


Figura 2.3 Grafo di flusso per la scomposizione del calcolo di una DFT di N/4 punti, con il metodo della decimazione nel tempo. (N=8)

Per il caso particolare della fig. 2.3, se ne deduce che, se le due DFT su 4 punti vengono calcolate seguendo le (B.19) e (B.20), i loro schemi di calcolo corrispondenti

vengono ad essere come quelli indicati nella figura 2.4. Quindi il calcolo di trasformata discreta di Fourier su 8 punti si è ridotto al calcolo di due DFT su 2 punti. nella figura 2.5 si può vedere il grafo di flusso completo .

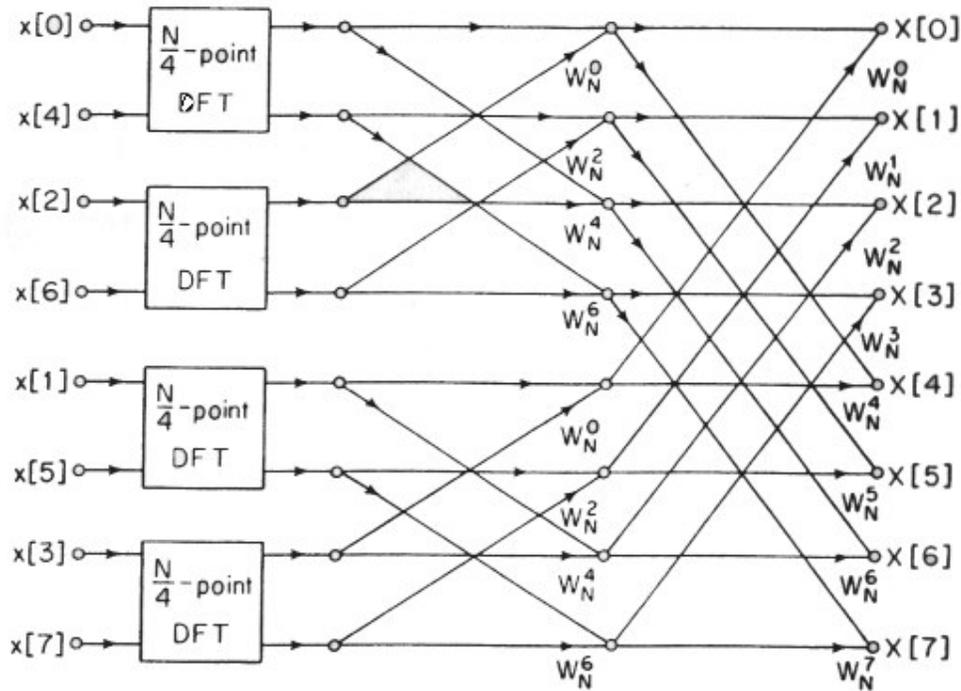


Figura 2.4 Grafo che si ottiene introducendo nella figura 2.2 gli schemi della figura 2.3

Per il caso più generale in cui N è una potenza di due con esponente maggiore di 3, si procede componendo le trasformate su $\frac{N}{4}$ punti delle (B.19) e (B.20) in trasformate su $\frac{N}{8}$, e così di seguito finché non si riduce trasformate su 2 punti. . Ciò richiede ν stadi di calcolo, dove $\nu = \log_2 N$. Primo si è visto come la scomposizione di una trasformata N punti in due trasformate su $\frac{N}{2}$ punti, richieda tra moltiplicazioni e addizioni complesse $N + 2\left(\frac{N}{2}\right)^2$. Quando le trasformate su $\frac{N}{2}$ punti vengono

scomposte in trasformate su $\frac{N}{4}$ punti, allora il fattore $\left(\frac{N}{2}\right)^2$ è sostituito con $\frac{N}{2} + 2\left(\frac{N}{4}\right)^2$, così che il calcolo complessivo richiede $N + N + 4\left(\frac{N}{4}\right)^2$ moltiplicazioni e addizioni complesse. Se $N = 2^v$, Ciò può essere al massimo $v = \log_2 N$ volte, quindi si conclude che dopo aver iterato al massimo la scomposizione, il numero di moltiplicazioni e addizioni complesse diventa $N \log_2 N$.

B.2.3 Algoritmi di FFT basati sulla decimazione in frequenza

Gli algoritmi di FFT basati sulla decimazione nel tempo sono stati sviluppati componendo il calcolo della DFT attraverso la formazione di sottosequenze della sequenza di ingresso $x(n)$ sempre più piccole. In alternativa si può pensare di dividere in modo analogo la sequenza di uscita, $X(k)$, in sottosequenza sempre più piccole. Gli algoritmi FFT originati da questo procedimento si dicono basati sulla decimazione in frequenza. Per derivarli, nel caso in cui N è una potenza di 2, si può innanzitutto dividere la sequenza di ingresso nella prima metà e nella seconda metà dei suoi punti, in modo da scrivere

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{nk}$$

oppure

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + W_N^{\frac{N}{2}k} \sum_{n=0}^{\frac{N}{2}-1} x\left(n + \frac{N}{2}\right)W_N^{nk} \quad (\text{B.21})$$

La B.21 contiene due sommatorie su $\frac{N}{2}$, ciascuna di esse non è una DFT su $\frac{N}{2}$ punti, in quanto nelle sommatorie appare W_N^{nk} e non $W_{\frac{N}{2}}^{nk}$. Mettendo insieme le due sommatorie nella 2.21 e utilizzando il fatto che $W_N^{\frac{N}{2}k} = (-1)^k$, si ottiene

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{nk} \quad (\text{B.22})$$

Dividendo in k pari e k dispari, si ottiene;

$$X(2r) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{2rm} \quad (\text{B.23})$$

$$X(2r+1) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n W_N^{2rm} \quad (\text{B.24})$$

con $r=0,1,\dots,\frac{N}{2}-1$

Queste due espressioni dono due DFT su $\frac{N}{2}$ punti, nel caso della (B.23) si tratta della DFT della somma della prima metà e della seconda metà della sequenza di ingresso, e nel caso della (2.24) si tratta della DFT del prodotto di W_N^n con la differenza fra la prima metà e la seconda metà della sequenza di ingresso. Differentemente dal caso della (B.22), le sommatorie nelle (B.23) e (B.24) corrispondono a delle DFT su $\frac{N}{2}$ punti in quanto

$$W_N^{2m} = W_{\frac{N}{2}}^m$$

Pertanto, sulla base delle (B.23) e (B.24), ponendo $g(n) = x(n) + x(n + \frac{N}{2})$ e $h(n) = x(n) - x(n + \frac{N}{2})$, la DFT può essere calcolata fornendo innanzitutto le sequenze $g(n)$ e $h(n)$, poi calcolando $h(n) W_N^n$ e infine calcolando la DFT su $\frac{N}{2}$ punti di queste due sequenze, ricavandone rispettivamente i valori di uscita di posto pari e quelli di posto dispari. In figura 2.5 è illustrato il procedimento nel caso di una DFT su 8 punti.

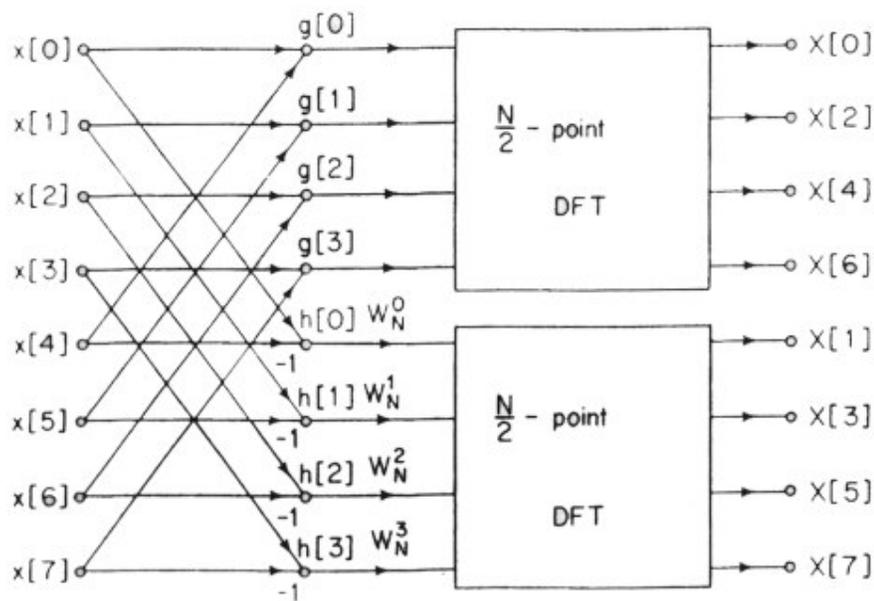


Figura 2.5 Grafo di flusso per la scomposizione del calcolo di una DFT di N punti in due DFT di N/2 punti, con il metodo della decimazione in frequenza (N=8)

Procedendo in modo simile a quello usato per derivare gli algoritmi basati sulla decimazione del tempo, dato che N è una potenza di 2, $\frac{N}{2}$ è pari quindi la DFT su $\frac{N}{2}$ punti possono essere effettuate calcolando separatamente per ciascuna di esse i valori di uscita di posto pari e quelli di posto dispari. Come nel caso della B.23 e B.24,

ciò si realizza combinando la prima metà dei punti e la seconda metà dei punti di ingresso per ognuna delle DFT su $\frac{N}{2}$ punti e calcolando quindi delle DFT su $\frac{N}{4}$ punti. Lo schema di flusso risultante nel caso dell'esempio su 8 punti è mostrato in fig. 2.6. In questo particolare caso il calcolo si è ridotto a quello di alcune DFT su 2 punti, le quali si calcolano sommando e sottraendo i punti di ingresso. Pertanto le DFT su 2 punti della figura 2.6 possono essere sostituite con lo schema di calcolo mostrato in fig. 2.7, così che il calcolo complesso della DFT su 8 punti diventa quello mostrato in fig. 2.8.

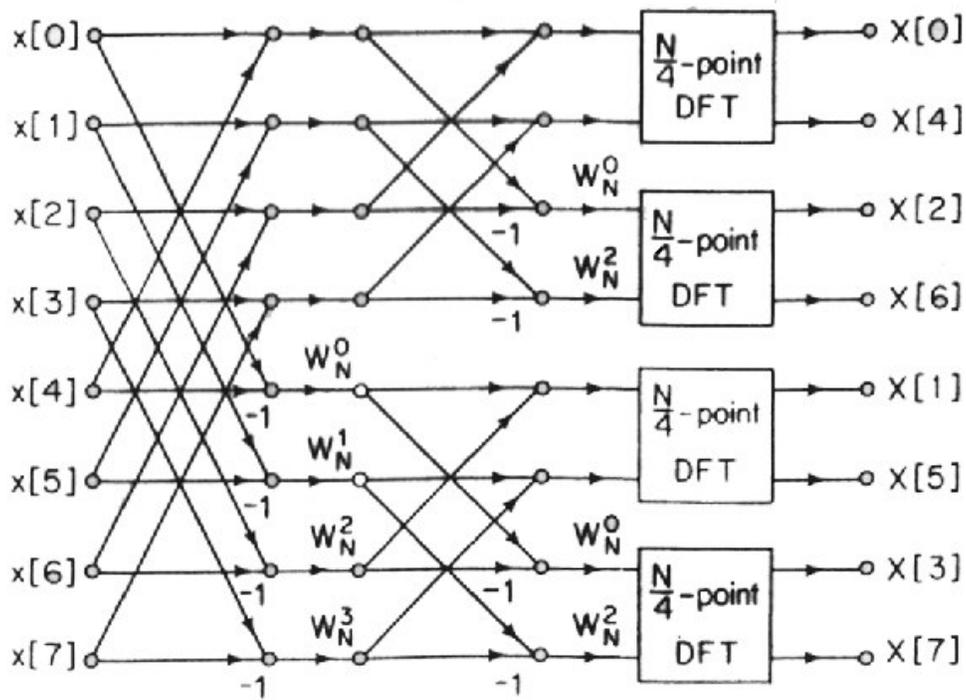


Figura 2.6 Grafo di flusso per la scomposizione del calcolo di una DFT di 8 punti nel calcolo di 4 DFT di 2 punti, con il metodo della decimazione in frequenza.

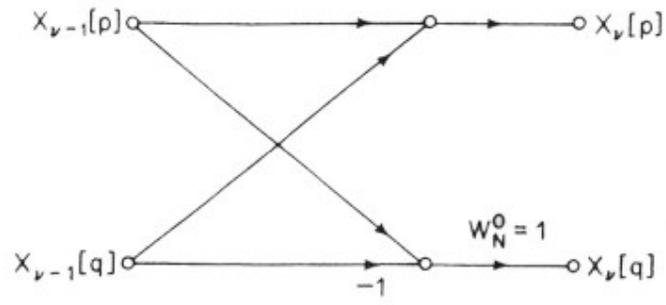


Figura 2.7 Grafo di flusso su 2 punti del tipo richiesto all'ultimo stadio di calcolo di una scomposizione basata sulla decimazione in frequenza.

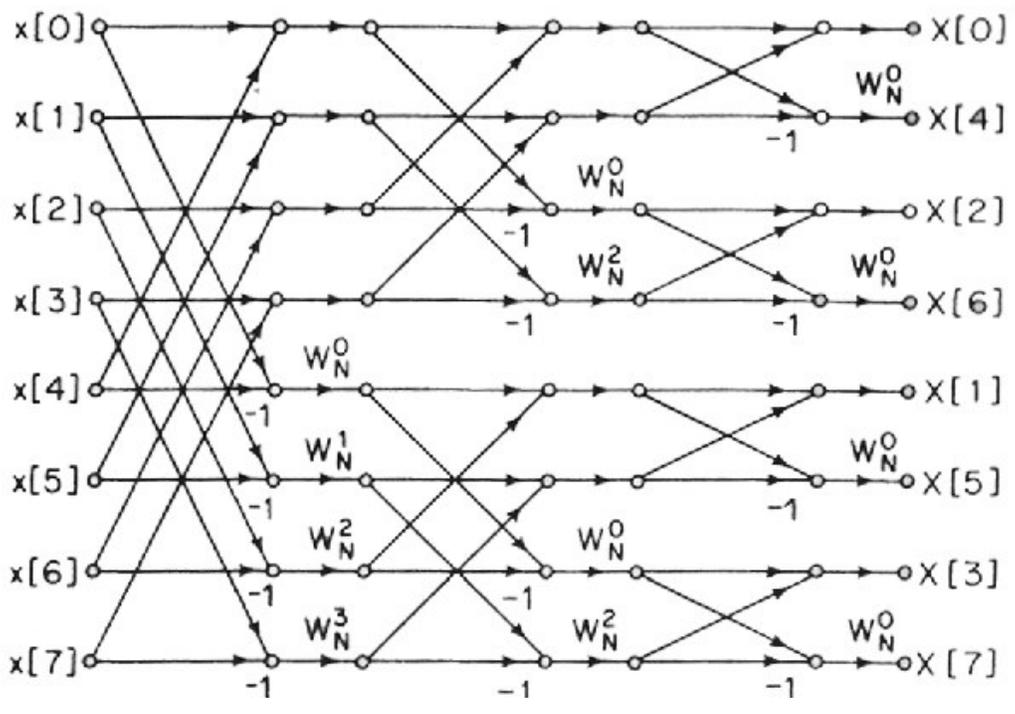


Figura 2.8 Grafo di flusso

Contando il numero di operazioni aritmetiche nella fig. 2.8, e generalizzando al caso $N = 2^v$, il calcolo della DFT richiede $\frac{N}{2} \log_2 N$ moltiplicazioni complesse e $N \log_2 N$ addizioni complesse. Pertanto il numero complessivo di operazioni è lo stesso per gli algoritmi basati sulla decimazione in frequenza e sulla decimazione nel tempo.

B.4 Esempi

In questo paragrafo si proporranno alcuni esempi sull'utilizzo della FFT, fornendo anche spiegazioni sull'utilizzo di tale algoritmo in Matlab.

All'interno di Matlab, l'algoritmo è già implementato nella funzione $y = \text{fft}(x,n)$ (la funzione accetta anche altri parametri, ma a noi non interessano), dove:

x è il segnale d'ingresso

n è il numero di punti utilizzati del calcolo della DFT; se n è maggiore del numero di campioni del segnale, si effettua lo zero padding su x , mentre se n è minore x viene tagliato

y rappresenta la DFT del segnale d'ingresso x

La funzione inversa della fft , ossia quella che calcola il segnale a partire dalla DFT, è la $y = \text{ifft}(x,n)$, dove:

x rappresenta la DFT del segnale calcolata attraverso la FFT

n è il numero di punti utilizzati del calcolo della IDFT

y rappresenta la IDFT del segnale di x

In questo caso, se il segnale iniziale è reale, l'operazione $iffi(fft(x))$ può dare alcune componenti complesse, quindi è conveniente considerarne la sola parte reale.

Il rumore viene generato sotto Matlab con la funzione *awgn*; un esempio di chiamata per questa funzione è:

$x = awgn(x, SNR, 'measured')$

dove:

- x è il segnale da sommare con rumore gaussiano bianco a media nulla
- SNR è il rapporto segnale rumore da impostare espresso in dB
- 'measured' è la stringa che passata come parametro indica alla funzione di calcolare la potenza di x prima di sommare il rumore

Si considera il segnale

$$x(t) = 2\sin(0.1t) + 3\sin(0.05t) + 5\sin(0.2t) + \sin(0.4t)$$

rappresentato in figura. Il segnale è scelto appositamente complesso per dimostrare le potenzialità dell'algoritmo FFT.

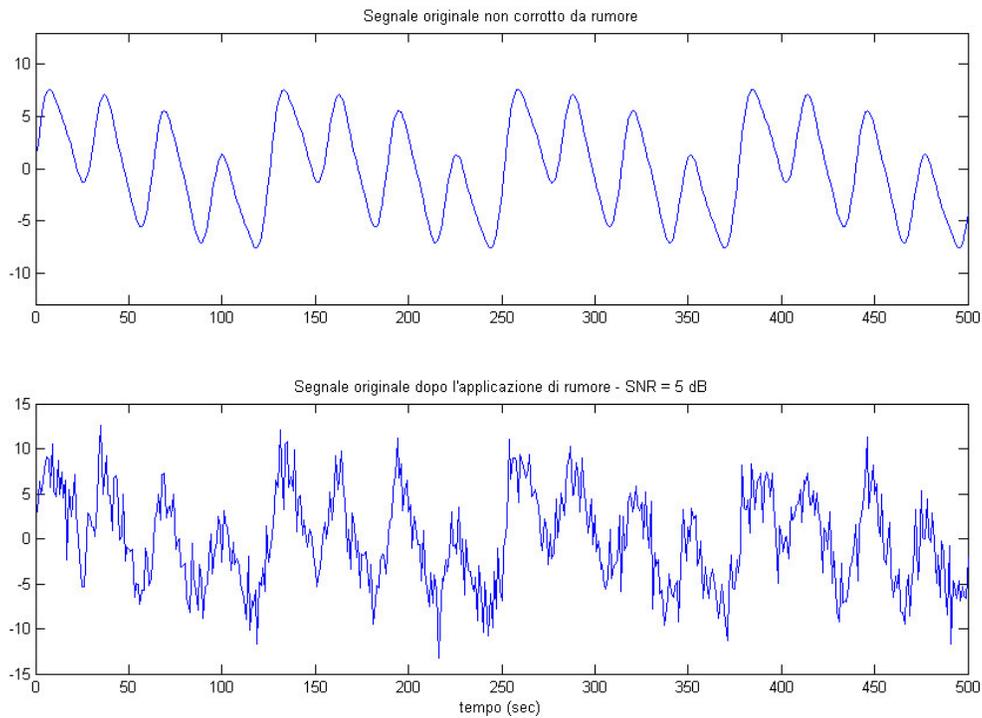


Figura 2.6. Segnale $x(t)$ prima e dopo l'applicazione del rumore

Come appare evidente, il segnale corrotto da rumore (SNR = 5dB) sembra non aver più niente a che vedere con il segnale originario. Vogliamo vedere se applicando l'algoritmo FFT è possibile estrapolare qualche informazione utile. I test vengono eseguiti al variare del numero di campioni della FFT; in particolare si utilizzeranno rispettivamente 128, 256, 512 campioni.

I risultati ottenuti sono evidenziati in tabella:

Frequenze reali (rad/sec)	Frequenze stimate (rad/sec) con punti FFT = 128	Frequenze stimate (rad/sec) con punti FFT = 256	Frequenze stimate (rad/sec) con punti FFT = 512
0.05	0.0503	0.0491	0.0491
0.1	0.1005	0.0982	0.0982
0.2	0.2011	0.1963	0.1963
0.4	0.4021	0.3927	0.4050

Sebbene ad occhio nudo non si riesca più nemmeno ad associare il segnale rumoroso a quello originario, mediante la FFT è possibile ricostruire il segnale ripulito dal rumore.

Utilizzando un numero di punti per la FFT pari alla lunghezza del segnale (in modo da avere la IDFT pari alla lunghezza del segnale originario), in questo caso 500, la ricostruzione è quasi perfetta:

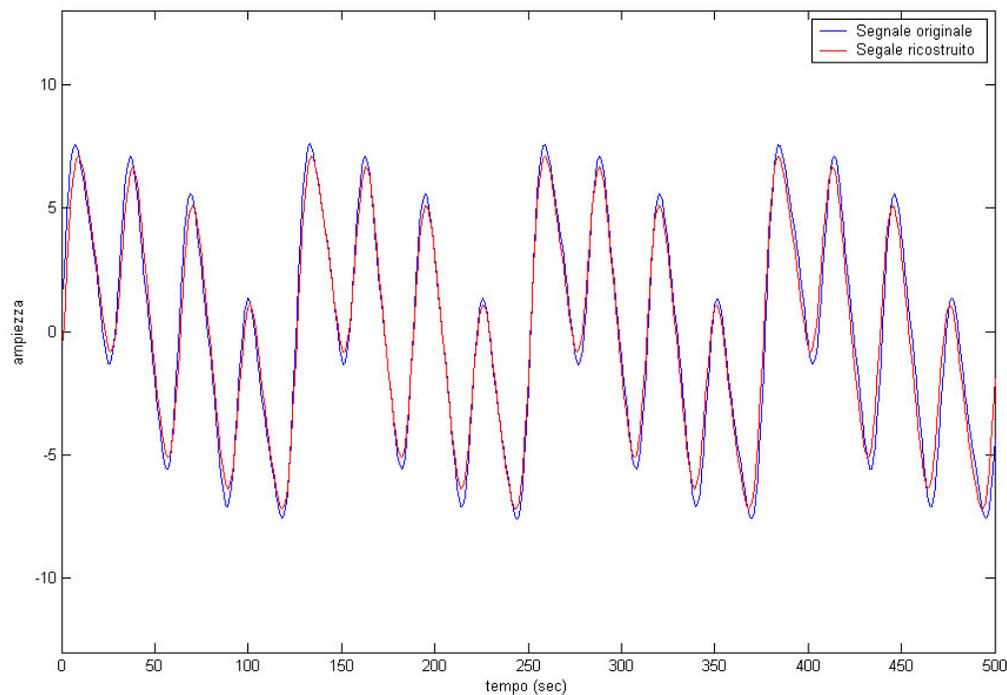


Figura 2.2. Sovrapposizione del segnale originale e quello ricostruito

Appare evidente come la FFT possa essere utilizzata con successo in molti campi dove si richiede la cancellazione del rumore su un segnale anche fortemente corrotto.

In pratica, per arrivare al risultato della figura sopra, sono possibili almeno due strade:

- 1) La prima possibilità consiste nel percorrere i seguenti passi:
 - a) calcolare la FFT ad N punti del segnale di partenza, ottenendo l'uscita y
 - b) calcolare lo spettro, valutando per ogni campione la potenza come $y \cdot \text{conj}(y) / N$
 - c) conoscendo il numero di componenti frequenziali p di cui è composto il segnale di partenza (p componenti), identificare le frequenze corrispondenti ai p picchi in potenza più elevati
 - d) nel vettore y , annullare tutte le componenti tranne quelle relative alle frequenze ottenute
 - e) calcolare la IDFT di y come $\text{ifft}(y)$, ottenendo il segnale ricostruito

- 2) La seconda strada percorribile consiste in:
 - a) come nel punto 1-a, calcolare la FFT ad N punti del segnale di partenza, ottenendo l'uscita y
 - b) come nel punto 1-b, valutare la potenza spettrale
 - c) impostare una soglia s , il cui valore dipende dall'influenza del rumore sul segnale
 - d) per il vettore y , azzerare tutte le componenti che presentano potenza inferiore a $s \cdot P_{max}$, dove P_{max} è la potenza massima
 - e) calcolare il segnale ricostruito mediante la funzione $\text{ifft}(y)$

La funzione FFT è stata utilizzata all'interno di questa tesi ponendo il numero di punti N pari al numero di campioni ottenuti dalla simulazione.