



UNIVERSITA' DEGLI STUDI DI CAGLIARI
FACOLTA' DI INGEGNERIA E ARCHITETTURA

L'analisi di Fourier e la DFT

Corso di Laurea Triennale in Ingegneria Elettrica, Elettronica e Informatica

Candidato:

Nicola Lombardi

Relatore:

Prof. Giuseppe Rodriguez

Anno accademico 2019-2020

*Uno speciale ringraziamento al Professor Giuseppe Rodriguez,
relatore di questa Tesi e docente dei miei corsi preferiti.
Un grazie anche alle persone che mi sono state vicine.*

“Follia è fare sempre la stessa taratura a ciclo aperto aspettandosi risultati uguali su sistemi diversi”

(Me stesso)

Abstract

L'analisi armonica di Fourier è uno strumento potente di analisi/sintesi e di affiancamento allo studio di segnali. Si basa su come le funzioni, definite su un continuum (in tutti i punti di un intervallo), possano essere rappresentate e analizzate in termini di funzioni periodiche, in seni e coseni. Questo immenso ed elegante argomento coinvolge molti problemi pratici. Sulla base delle innumerevoli applicazioni, questa tesi si costituisce di uno studio completo della Discrete Fourier Transform: i suoi usi, le sue proprietà, le sue applicazioni e la sua relazione con la Serie di Fourier e le Trasformate di Fourier, sia di segnali analogici che digitali. Tratterà, dunque, le fasi più cruciali dell'elaborazione e del passaggio "analogico-digitale": il campionamento, il troncamento e l'interpolazione.

Da un punto di vista concettuale la trattazione che seguirà riguarda una specifica applicazione dell'ampio concetto di analisi in frequenza e dell'uso di tale in forma discreta e unidimensionale: l'Analisi delle sequenze numeriche. Per comprendere l'importanza dell'analisi in frequenza di un segnale, o sequenza numerica, è sufficiente ricordare che l'intera teoria delle telecomunicazioni poggia sulla trattazione dei segnali nel dominio della frequenza. La posizione e l'ampiezza di una portante, la larghezza di banda di un segnale modulato, la quantità di rumore presente su un canale, le distorsioni armoniche sono soltanto alcune delle informazioni emergenti dall'osservazione dello spettro di un segnale.

In diversi settori della scienza e della tecnologia, poi, occorre assolutamente visualizzare ed elaborare il segnale per consentire l'estrazione dell'informazione desiderata oppure usare gli strumenti di Fourier per la valutazione di diversi parametri associati ad un generico segnale: energia, potenza, durata efficace, coefficienti di Fourier, densità spettrale di potenza e funzione di autocorrelazione.

Nel corso dell'evoluzione delle telecomunicazioni, in generale, il tradizionale approccio analogico è stato sempre più affiancato da quello digitale, caratterizzato da una sempre più crescente validità di prestazioni ottenibili in presenza di determinate condizioni e dalla facilità di elaborazione e possibilità di memorizzazione di moli di dati sempre maggiori, in parallelismo con l'evoluzione delle memorie e delle loro prestazioni. Nell'affrontare e sviluppare l'argomento della tesi riguardante l'uso e le potenzialità dello "strumento DFT", si è messo costantemente in evidenza questo ormai inevitabile passaggio, ponendo l'accento in modo particolare sulle motivazioni e sui vantaggi conseguenti.

La tesi rimane pur sempre legata agli aspetti di applicazione pratica, in ambiti di Ingegneria, legata anche a tutti i concetti teorici ad essi adibiti, sulla Trasformata Discreta. Infatti, vengono sottolineati i settori di applicazione dell'analisi armonica, della FFT e in particolare quello riguardante il filtraggio di segnali e Analisi di Fourier su campioni di funzioni.

Altrettanto importante è ciò che in questa tesi non c'è! Indubbiamente, una delle ragioni per l'aumento spettacolare della popolarità dell'analisi pratica di Fourier è l'invenzione dell'algoritmo noto come trasformata veloce di Fourier: la tesi tratta solo la versione monodimensionale della trasformata, ossia non tratta l'immagine processing. Per tutta la sua magia e potenza, la FFT è solo un modo estremamente efficiente per calcolare la DFT: si parla sempre dello stesso strumento di analisi. Va detto chiaramente dall'inizio che questa tesi non riguarda la FFT, che ha una vasta gamma di implementazioni. Si dedicherà solo una parte degli esperimenti a quest'ultima, mentre non verrà trattato né toccato l'immagine processing e tutto ciò che riguarda Digital Media.

Indice

0 Introduzione

Le radici dell'analisi di Fourier

1 Teoria di Fourier: generalità ed essenzialità

1.1 La serie di Fourier

1.1.1 Analisi di Fourier: generalità

1.1.2 La forma della serie di Fourier

1.1.3 Sintesi con un numero finito di armoniche

1.1.4 Corollario

1.1.5 Proprietà dei coefficienti della serie di Fourier

1.2 La trasformata di Fourier

1.2.1 Definizione generale

2 Il passaggio dal continuo al discreto

2.1 Il campionamento dei segnali: leggi di Poisson

2.2 Dal tempo continuo al tempo discreto

2.3 Teoremi di prodotto e di convoluzione di sequenze numeriche aperiodiche

2.4 Dimostrazione sulla Trasformata di una sequenza

2.5 La condizione di Nyquist

2.6 Interpolazione

2.7 Troncamento del segnale

2.8 La Trasformata discreta di Fourier : come nasce

3 La DFT

3.1 La matrice W

3.2 Proprietà della trasformata

3.3 Derivazione della DFT da minimi quadrati

3.4 Nozioni sulla parte sperimentale

3.5 Complessità computazionale

3.6 Conclusione della trattazione

4 Esperimenti

4.1 DFT: approssimazioni

4.1.1 Approssimazione di soluzioni di ode

4.1.2 Approssimazione di coefficienti di Fourier

4.1.3 Interpolazione di data-set

4.1.4 Approssimazione trigonometrica

4.2 DFT: applicazione al campionamento

4.2.1 Aliasing

4.2.2 Decimazione dei campioni

4.3 Digital filtering

4.4 DFT: analisi spettrale

4.4.1 Analisi e zero padding

4.4.2 DFT: analisi spettrale di potenza

4.5 Applicazione DFT: calcolo di convoluzioni

4.5.1 DFT: overlap and save

4.5.2 Modulazione analogica: amplitude modulation (AM)

4.6 DFT: background della FFT

4.6.1 Applicazione della FFT

4.6.2 Moltiplicazione di polinomi

4.7 Considerazioni sulle prove sperimentali

Conclusione

APPENDICE

Bibliografia

0

Introduzione

Prima di trattare le generalità della teoria di Fourier, è interessante indagare quali fossero stati, all'epoca, i problemi che portarono a definire un insieme di teoremi ed enunciati tanto importanti.

Le radici dell'analisi di Fourier

Ci sono stati tre problemi della storia che hanno fondato le radici della cosiddetta "analisi di Fourier".

Il primo problema era descrivere la **vibrazione di una corda tesa** ancorata ad entrambe le estremità (o equivalentemente la propagazione del suono in un mezzo elastico). Sorprendentemente, l'onda descritta dall'equazione come la conosciamo oggi: era già stata formulata. I matematici Jean d'Alembert, Leonhard Euler, Daniel Bernoulli e Joseph-Louis Lagrange avevano metodi di soluzione proposti intorno al 1750. La soluzione di Bernoulli prese la forma di una serie trigonometrica.

Le onde sono oscillazioni temporali o spaziali (ambidue nel caso di una funzione a più variabili) di tutte quelle quantità che possono essere così descritte, tramite le loro variazioni tempo-spaziali. Ad esempio, il livello dell'acqua attorno al livello medio del mare, in presenza di onde in acqua profonda o la posizione di una corda di una chitarra pizzicata, o la densità dell'aria fatta vibrare dalle corde vocali o da un pistone.

Si consideri una porzione di corda spostata meccanicamente dall'equilibrio.

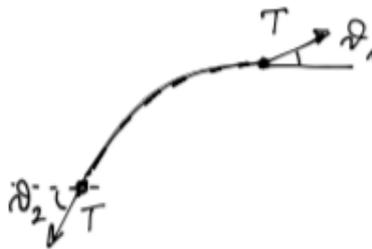


Figura 1 : Tensioni su un tratto di corda oscillante.

La posizione lungo la corda è x , e lo spostamento laterale della corda dalla sua posizione di equilibrio è y , che è funzione di x e del tempo t . Ai capi di un suo tratto generico infinitesimo dx , la corda è soggetta a tensioni, cioè forze di richiamo, che non si cancellano in generale (lo fanno se il tratto di corda è localmente e momentaneamente dritto). La forza risultante dalla dinamica delle leggi di Newton:

$$F_{TOT} = T(\sin \theta_1 - \sin \theta_2).$$

Sviluppando in serie per piccolo angolo, $\sin \theta_1 \sim \theta_1$. Ma poiché anche la tangente ha lo stesso sviluppo, al primo ordine (linearizzando): $\sin \theta_1 \sim \tan \theta_1$; allora dalla geometria elementare, sappiamo che $\tan \theta_1$ è la pendenza locale di $y(x)$ rispetto all'asse x , cioè la sua derivata; quindi è facile costruire:

$$F_{TOT} = T(\tan \theta_1 - \tan \theta_2) \cong \frac{\partial y}{\partial x}(x_0 + h, t) - \frac{\partial y}{\partial x}(x_0, t).$$

Come diretta conseguenza del significato geometrico di tangente e derivata prima. Allora, dall'Analisi Matematica, recuperiamo la notazione del differenziale primo e linearizziamo : approssimazione tipica dei problemi di Ingegneria.

In generale si ha:

$$\partial z = f(x, y) - f(x_0, y_0) = \frac{\partial f}{\partial x} \Big|_{x_0, y_0} (x - x_0) + \frac{\partial f}{\partial y} \Big|_{x_0, y_0} (y - y_0) + o\left(\sqrt{(x - x_0)^2 + (y - y_0)^2}\right).$$

All'aumentare dei fattori di ingrandimento, i grafici della funzione e del suo piano tangente in (x_0, y_0) diventano indistinguibili (l'errore o "distanza" della funzione dal piano tangente, tende a zero più rapidamente dell'incremento, ossia dell'errore rappresentato con la notazione o-piccolo). Il differenziale primo rappresentato fa riferimento ad entrambe le variabili. Nel nostro caso, la corda ha posizione di equilibrio y e questa è una funzione di due variabili, ma possiamo linearizzare considerando delle variazioni di y , relative alla variabile indipendente x , che non sono corrispondenti direttamente a variazioni causate da t , ossia poter considerare il caso di linearizzazione rispetto ad una singola variabile.

Otteniamo dunque la seguente espressione:

$$\frac{\partial^2 y}{\partial x^2} = \frac{\mu}{T} \frac{\partial^2 y}{\partial t^2}$$

E' stata applicata la seconda legge di Newton in forma differenziale espressa rispetto alla posizione della corda e la massa della corda è stata espressa in termini di densità μ .

Notiamo che la soluzione dell'equazione d'onda Eq.6.3 deve soddisfare una sola proprietà, e precisamente deve dipendere da x e t come:

$$y(x, t) = y(x - vt) = e^{-\frac{x^2 - (vt)^2}{x_0^2}}$$

Dove l'ultima uguaglianza esprime una possibile soluzione in una forma generica.

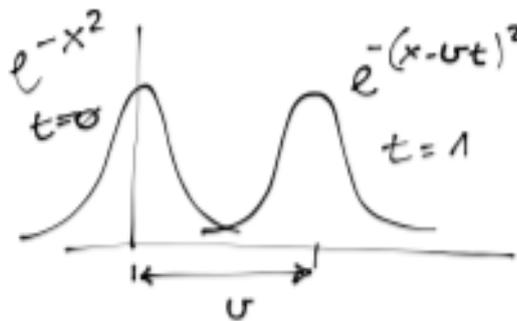


Figura 2 :Gaussiana, soluzione tipo dell'equazione delle onde.

Una soluzione fisica generale sarà una combinazione lineare (l'equazione, ricordiamo, è lineare) del tipo $Ay(x - vt) + By(x + vt)$. Più precisamente, per avvicinarci al fulcro dell'analisi di Fourier, la soluzione proposta da Bernoulli è del tipo:

$$y(x, t) = A \sin(x) \cos(at) + B \sin(2x) \cos(2at) + \dots + Z_n \sin(nx) \cos(nat) + \dots$$

Già questa soluzione anticipò la forma continua di una serie, quella che sarebbe stata la Serie di Fourier. Sembra che sia Eulero che Lagrange abbiano effettivamente discretizzato il problema della corda vibrante immaginando la corda come un sistema fisico su cui coesistono un numero finito di particelle connesse. La

soluzione di questo problema discreto ha richiesto la ricerca di campioni della funzione che descrive lo spostamento della stringa.

Vediamo un altro classico modello matematico del **fenomeno della conduzione del calore**, ad un duplice scopo: introdurre una delle principali equazioni differenziali alle derivate parziali della fisica matematica e spiegare l'origine delle serie di Fourier. Studiamo il fenomeno della conduzione del calore attraverso un filo sottile, rettilineo (oppure una sbarra), isotropo, omogeneo, e di sezione costante S . Ci interessiamo soltanto della conduzione del calore nella direzione dell'asse del filo (asse x) trascurando la conduzione nelle direzioni trasversali, come pure le eventuali dispersioni di calore. Ammettiamo che la temperatura dei punti del filo sia una funzione $u(x, t)$ che dipende solo dall'ascissa x e dal tempo t . La quantità di calore dQ che, nell'intervallo di tempo dt , attraversa la sezione del filo di ascissa x da sinistra verso destra data è da una forma molto simile alla quantità di calore dQ che, nell'intervallo di tempo dt , attraversa la sezione del filo di ascissa $x + dx$, da destra verso sinistra:

$$\begin{aligned}\frac{d}{dt} Q_1 &= -kS \frac{\partial u}{\partial t}(x, t) \\ \frac{d}{dt} Q_2 &= kS \frac{\partial u}{\partial t}(x+dx, t)\end{aligned}$$

Dalla definizione di capacità termica, calore specifico e massa possiamo ottenere un'espressione finale. Partendo dalla capacità termica abbiamo:

$$C = \frac{dQ}{du(x, t)} \Rightarrow \frac{dQ}{dt} = \frac{\partial u(x, t)}{\partial t} (c\mu S) dx.$$

Considerando come Q il calore totale, inteso come somma delle prime due equazioni, allora possiamo eguagliarlo alla terza e spostare il differenziale dove più conviene.

A partire da $c\mu S \frac{\partial u}{\partial t} = \frac{kS}{dx} \left[\frac{\partial u}{\partial t}(x + dx, t) - \frac{\partial u}{\partial t}(x, t) \right]$ si ottiene: $\mu c \frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}$, detta equazione del calore in una dimensione spaziale.

Analogamente, la conduzione del calore in un corpo tridimensionale $\Omega \subset \mathbb{R}^3$ si può modellizzare tramite l'equazione $\mu c \frac{\partial u}{\partial t} = k \Delta u$, essendo Δ l'operatore di Laplace. Poiché i teoremi di esistenza e le proprietà qualitative delle soluzioni non dipendono dal valore numerico delle costanti $c, \mu, k > 0$, si vuole concentrare l'attenzione sull'equazione a derivate parziali con coefficienti unitari.

Consideriamo un filo disposto come la circonferenza di equazioni parametriche:

$$\begin{cases} x = \cos(\alpha) \\ y = \sin(\alpha) \\ \alpha \in [0, 2\pi] \end{cases}$$

e spostando il contesto spaziale in radiale (coordinate polari), allora avremo il risultato $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial \alpha^2}$ tale per cui si vede, per sostituzione (o per separazione delle variabili), che le funzioni $e^{-k^2 t} \cos(k\alpha)$ per $k = 0, 1, 2, \dots$ e le funzioni $e^{-k^2 t} \sin(k\alpha)$, per $k = 1, 2, 3, \dots$ sono soluzioni. Fourier ritiene che l'integrale generale di tale equazione sia:

$$u(\alpha, t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} e^{-k^2 t} (a_k \cos k\alpha + b_k \sin k\alpha)$$

dove i coefficienti a_k e b_k si devono ricavare dalla temperatura iniziale $u(\alpha, 0)$ tramite l'uguaglianza

$$u(\alpha, 0) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos k\alpha + b_k \sin k\alpha)$$

e questa è la serie di Fourier.

J.-B. J. Fourier, nel trattato “Theorie analytique de la chaleur” (1822) studia la conduzione del calore in corpi di forme diverse, incluso l’anello, ossia la trattazione appena illustrata. Successivamente determina l’espressione dei coefficienti a_k e b_k tramite particolari serie numeriche, dedotte sotto l’ipotesi che la funzione generatrice $u(\alpha, 0)$ sia analitica. In un momento ancora successivo, risolvendo un’opportuna equazione differenziale ordinaria, egli perviene alla rappresentazione di a_k e b_k tramite integrali:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} u(t) \cos(kt) dt,$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} u(t) \sin(kt) dt .$$

Il terzo problema che ha nutrito le terze radici dell’analisi di Fourier, in particolare nella sua forma discreta, era quella di determinare le **orbite dei corpi celesti**. Eulero, Lagrange e Alexis Claude Clairaut hanno apportato contributi fondamentali proponendo che i dati presi dalle osservazioni siano approssimati da combinazioni lineari di funzioni di carattere periodico. Il calcolo dei coefficienti, in queste espansioni trigonometriche, ha portato ad un calcolo conosciuto come trasformata di Fourier discreta: la DFT, Discrete Fourier Transform.

1 Teoria di Fourier: generalità ed essenzialità

1.1 La serie di Fourier

1.1.1 Analisi di Fourier: generalità

Gli strumenti di analisi dei segnali che compongono la teoria di Fourier si caratterizzano per il fine per cui gli stessi vengono usati. Esistono diverse classi di segnali e per ognuna, si può affermare, esista una rappresentazione e che questa sia utile ai problemi che si riscontrano quando si analizza un sistema ingegneristico e/o fisico dove si valutano i segnali attribuiti alle variabili che li compongono. Si pensi ai segnali sinusoidali: essi costituiscono una classe speciale di segnali periodici. Per tale classe di segnali esiste quella che è definita come trasformata di Steinmetz:

$$S\{f(t)\}(\omega) = \frac{\omega}{\pi} \int_{-\frac{\pi}{\omega}}^{\frac{\pi}{\omega}} f(t)e^{-i\omega t} dt$$

Questa trasformata integrale è usata anche per altre motivazioni: trasforma le equazioni integrali e le equazioni differenziali in equazioni polinomiali, che sono più immediate da risolvere ed è strettamente legata alla trasformata di Fourier.

Insomma, per ogni classe di segnali nel dominio del tempo, esiste una rappresentazione che sposta l'attenzione in un nuovo dominio in cui la variabile indipendente è la frequenza.

1.1.2 La forma della serie di Fourier

Un polinomio trigonometrico di ordine n è rappresentato da una funzione periodica del tipo:

$$T_n(t) = a_0 + \sum_{k=1}^n [a_k \cos k\omega t + b_k \sin k\omega t],$$

dove a_0, a_k, b_k, ω sono numeri reali. Tale funzione ha periodo $T = \frac{2\pi}{\omega}$ in quanto combinazione lineare delle $2n+1$ funzioni elementari $1, \cos\omega t, \sin\omega t, \dots, \cos n\omega t, \sin n\omega t$, le quali hanno medesimo periodo.

Considerando che il sistema $\{1, \cos(t), \sin(t), \dots, \cos(kt), \sin(kt)\}_{k \in \mathbb{Z}}$ ortogonale in $\mathcal{L}_{[0, 2\pi]}^2$, il procedimento proposto da Fourier consiste nell'imporre che le proiezioni (calcolate mediante il prodotto scalare) della funzione di periodo $f(t)$ e del polinomio $T_n(t)$, su ciascuna funzione di base $1, \cos\omega t, \sin\omega t, \dots, \cos n\omega t, \sin n\omega t$, siano identiche. Tale imposizione equivale a richiedere che l'integrazione in $[0, T]$ della funzione per le funzioni di base risulti uguale all'integrazione della funzione approssimante per le stesse funzioni di base. Questo fatto, in conseguenza della ortogonalità delle funzioni di base, implica che i coefficienti di Fourier abbiano la seguente espressione:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(0) dt, \quad \text{detto anche valor medio del segnale } f(t);$$

Esprimendo il periodo tramite la sua definizione, ponendo $\alpha = \omega t$ e adattando gli estremi di integrazione, è possibile anche riformulare tali definizioni come:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt,$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt) dt .$$

Inoltre è possibile riassumere il Lemma di ortogonalità sfruttando l'identità di Eulero sul sistema ortogonale in versione complessa $\{e^{jkt}\}_{k \in \mathbb{Z}}$ e usando l'espressione del prodotto scalare in $\mathcal{L}_{[0,2\pi]}$ ²:

$$\langle e^{jkt}, e^{jmt} \rangle = \frac{e^{j(k-m)t}}{2\pi j(k-m)} \Big|_0^{2\pi} \text{ solamente se } k \neq m \text{ altrimenti } \langle e^{jkt}, e^{jmt} \rangle = 1.$$

Dunque, secondo Fourier, se f è una funzione integrabile in $[-T, T]$, allora ad essa è formalmente associabile, in modo univoco, la serie di Fourier che altro non è che l'estensione infinita della somma che compone il famoso polinomio trigonometrico.

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{L} + b_n \sin \frac{n\pi t}{L} \right) \text{ dove } T=2L.$$

In taluni ambiti è opportuno esprimere la serie di Fourier in alcune forme alternative. Per vari motivi, in diversi contesti applicativi, è utile sostituire la forma precedente del polinomio trigonometrico con la seguente

$$T_n(t) = a_0 + \sum_{k=1}^n c_k \cos(k\omega t + k\phi)$$

Per vedere se sussiste l'uguaglianza tra le due forme è sufficiente sviluppare il coseno tramite la formula di addizione degli archi ed imporre l'uguaglianza con l'altro polinomio, dove invece è presente anche il seno, per ottenere le relazioni di conversione.

Segue immediatamente che: $\begin{cases} c_k \cos k\phi = a_k \\ c_k \sin k\phi = -b_k \end{cases}$

$$\begin{cases} c_k = \sqrt{a_k^2 + b_k^2} \\ k\phi = \arctan\left(-\frac{b_k}{a_k}\right) \end{cases}$$

Esiste ancora un altro modo per rappresentare lo stesso polinomio che, per quanto sia del tutto equivalente a quelli già illustrati, risulta più adatto nell'analisi e controllo dei segnali digitali: la forma complessa del polinomio trigonometrico o della serie di Fourier (che nel caso limite coincidono), dove i coefficienti sono proprio numeri complessi.

$$T(t) = \sum_{k=1}^{\infty} (X_k e^{j2\pi kft});$$

$$X_k = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-j2\pi kft} dt$$

$$\omega = 2\pi f = \frac{2\pi}{T}.$$

La seconda delle tre è una equazione di analisi che permette di stabilire qual è il contenuto in termini di oscillazioni armoniche del segnale (in una parola, di analizzare il segnale). La prima delle tre, viceversa, è una equazione di sintesi che, note le ampiezze e fasi delle varie armoniche (cioè noti i coefficienti di Fourier) permette di ricostruire, cioè sintetizzare, il segnale dato a partire dalle proprie componenti frequenziali (armoniche).

Tale forma consegue direttamente dalle formule di Eulero:

$$e^{\pm jx} = \cos(x) \pm j\sin(x),$$

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2},$$

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2j}.$$

Quanto finora enunciato è valido per un segnale periodico e dunque anche per un segnale sinusoidale, ovviamente, dato che è stata citata la Trasformata di Steinmetz che opera su segnali anche non periodici, ossia aperiodici. La convergenza della serie di Fourier risulta un argomento molto importante dal punto di vista matematico poiché possiede tantissimi teoremi che la citano. Tuttavia non è scopo della tesi enunciarne e dimostrarne la tesi. La serie è uno strumento efficace dal punto di vista dell'analisi delle componenti del segnale, ossia le componenti armoniche o sinusoidi che lo costituiscono e che, dunque, ne rappresentano il contenuto energetico.

Il Teorema di Parseval è uno dei risultati più significativi dell'importanza di tale sviluppo per l'analisi dei segnali, basti pensare che tramite la serie di Fourier è possibile esprimere il valore efficace di un segnale (o radice quadrata della potenza media del segnale).

La potenza media di un segnale troncato è definita come:

$$P = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |x(t)|^2 dt$$

tramite il teorema di Parseval, possiamo legare il segnale periodico ai suoi coefficienti di Fourier e la forma migliore, in questo contesto, è proprio la complessa che ci permette di arrivare al risultato compatto:

$$P = \sum_{k=-\infty}^{+\infty} |X_k|^2$$

da cui, considerando il segno positivo della radice quadrata, si ha il valore efficace.

1.1.3 Sintesi con un numero finito di armoniche

Evidentemente, l'equazione di sintesi prevede l'uso di infinite armoniche per ricostruire il segnale, d'altro canto, anche la potenza richiede la conoscenza di tali: ciò non è ovviamente possibile dal punto di vista computazionale, sia umano che elettronico. Si può fare un'osservazione che porta già ad una discretizzazione. D'altronde, condizione necessaria alla convergenza della serie è che l'ampiezza $|c_k|$ delle armoniche tenda a zero quando k inizia ad essere molto grande (al limite). Questo comporta che le armoniche più "importanti" ai fini della sintesi del segnale sono un numero limitato, e che quindi la serie può essere sostituita ai fini pratici con una sommatoria di un numero finito di termini.

Si possono illustrare due esempi per arrivare al teorema sulla velocità di convergenza.

Si consideri l'onda quadra.

Osservazione

la sequenza dei coefficienti di Fourier dell'onda quadra (qui in versione a quarto d'onda di simmetria) è immaginaria pura e dispari, e i coefficienti di ordine pari sono tutti nulli.

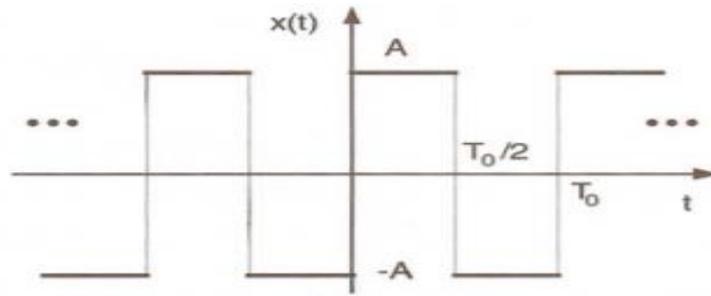


Figura 3: onda quadra a quarto d'onda dispari

$$\begin{aligned}
 X_k &= -\frac{2j}{T_0} \int_0^{T_0/2} x(t) \sin(2\pi k f_0 t) dt = -j \frac{2A}{T_0} \int_0^{T_0/2} \sin(2\pi k f_0 t) dt \\
 &= j \frac{2A}{2\pi k f_0 T_0} \cos(2\pi k f_0 t) \Big|_{t=0}^{t=T_0/2} = j \frac{A}{\pi k} [\cos(\pi k) - 1] = j \frac{A}{\pi k} [(-1)^k - 1]
 \end{aligned}$$

L'espressione del coefficiente di Fourier commuta, valida per indice dispari, nell'espressione:

$$X_k = \frac{2A}{j\pi k} \quad \text{oppure, usando lo sviluppo in forma trigonometrica estesa, i termini } b_k = \frac{4A}{\pi k} .$$

L'aspetto dell'espressione delle successioni di entrambi i due sviluppi suggerisce che tali coefficienti, e quindi la potenza, non continuo più nulla per indici molto elevati.

Avendo un indice con esponente unitario di elevamento a potenza, si può attribuire un ordine unitario della "velocità" di convergenza a zero dei coefficienti.

Lo spettro, infatti, è rappresentato da un grafico discreto antisimmetrico, essendo il segnale periodico dispari:

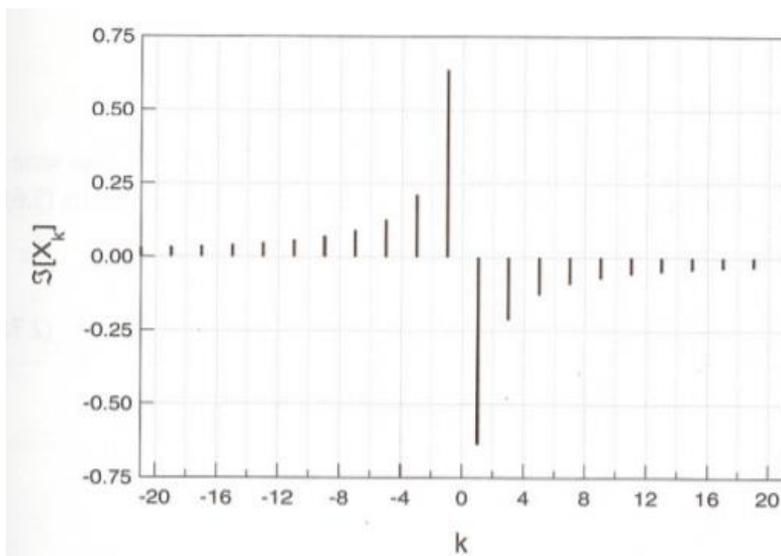
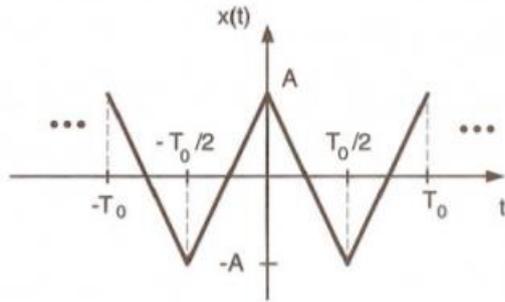


Figura 4: spettro del segnale onda quadra

Da qui scaturisce la motivazione per cui nelle definizioni come THD (total harmonic distortion); si prende, infatti, come riferimento la componente fondamentale con cui misurare l'impatto della distorsione della forma d'onda causata dalla presenza di armoniche. Si può dimostrare, infatti, che ogni armonica di un segnale ha un valore di picco esprimibile in funzione del valore di picco dalla fondamentale.

Si consideri l'onda triangolare simmetrica.



$$X_k = \frac{2A}{(\pi k)^2} [1 - (-1)^k] = \begin{cases} 0 & k \text{ pari} \\ \frac{4A}{(\pi k)^2} & k \text{ dispari} \end{cases}$$

Figura 5: onda triangolare simmetrica con relativo coefficiente

Si nota innanzitutto che, anche in questo caso, tutti i coefficienti con indice pari sono nulli perché $x(t)$ è dispari. La caratteristica più evidente, in un confronto con l'onda quadra, è che l'ampiezza delle armoniche tende a zero molto più velocemente quando k è molto grande. L'espressione del coefficiente suggerisce che tale velocità è proporzionale a $1/k$, ma stavolta quadro; quindi le armoniche superiori hanno meno "importanza" nella sintesi dell'onda triangolare rispetto al caso dell'onda quadra. Per un segnale di questo tipo è dunque possibile approssimare il polinomio infinito di Fourier con un numero finito di termini più piccolo: è lecito anticipare il troncamento dello sviluppo ad un indice prima di quello dello sviluppo dell'onda quadra.

Questo risultato è molto importante perché la Trasformata discreta di Fourier si ottiene esattamente da una discretizzazione che si esegue sia sulla trasformata continua che sull'espressione del coefficiente di Fourier, per cui, è di notevole importanza capire già dalla Serie fin dove serve dilungarsi con l'analisi armonica, a livello di indice.

Si è ribadito che la convergenza della serie di Fourier non sarà oggetto di studio ed illustrazione di tale lavoro. Per cui è lecito solo enunciare il corollario del Teorema di Riemann-Lebesgue.

1.1.4 Corollario

Supposto che il segnale da sviluppare in serie di Fourier sia derivabile con continuità per 'r-2' volte nell'intervallo in cui lo stesso è periodico e che la sua r-1-esima derivata sia regolare a tratti, allora la successione dei coefficienti di Fourier tende a zero come l'inverso di k^r .

Ciò che è, però, interessante è che questo comportamento ha un suo motivo dettato dalla forma d'onda nel dominio del tempo. L'onda triangolare ha variazioni d'ampiezza meno brusche dell'onda quadra e il polinomio oscillante di Fourier, troncato, riesce a commutare meglio alla forma del segnale originale con un numero di indici inferiore proprio perché le derivate del segnale stesso indicano una possibilità di "rilassamento", della combinazione lineare di seni/coseni, migliore.

A seconda del segnale, dunque, un indice di "numero minimo" di armoniche sarà proprio l'ordine di regolarità.

1.1.5 Proprietà dei coefficienti della serie di Fourier

Le proprietà di quella che verrà definita come Trasformata Discreta di Fourier vengono conservate nell'operazione di discretizzazione del coefficiente di Fourier. E' dunque sensato illustrare almeno le principali peculiarità e caratteristiche di quest'ultimo in una tabella. Fondamentalmente, da delle considerazioni

geometriche, il coefficiente k-esimo di Fourier prende una forma differente a seconda del caso. Quanto riportato sotto si riferisce allo sviluppo in serie trigonometrico classico del tipo:

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{L} + b_n \sin \frac{n\pi t}{L} \right)$$

con i coefficienti:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt,$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt) dt.$$

Simmetria	Condizioni richieste	a_n e b_n
Pari	$f(-t) = f(t)$	$b_n = 0$ $a_n = \frac{2}{\pi} \int_0^{\pi} f(t) \cos(h\omega t) d(\omega t)$
Dispari	$f(-t) = -f(t)$	$a_n = 0$ $b_n = \frac{2}{\pi} \int_0^{\pi} f(t) \sin(h\omega t) d(\omega t)$
Semionda	$f(-t) = -f(t + (1/2)T)$	$a_n = b_n = 0$ per h pari $a_n = \frac{2}{\pi} \int_0^{\pi} f(t) \cos(h\omega t) d(\omega t)$ per h dispari $b_n = \frac{2}{\pi} \int_0^{\pi} f(t) \sin(h\omega t) d(\omega t)$ per h dispari
Quarto d'onda pari	pari e semionda	$b_n = 0$ per qualunque h $a_n = \begin{cases} \frac{4}{\pi} \int_0^{\pi/2} f(t) \cos(h\omega t) d(\omega t) & \text{per } h \text{ dispari} \\ 0 & \text{per } h \text{ pari} \end{cases}$
Quarto d'onda dispari	dispari e semionda	$a_n = 0$ per qualunque h $b_n = \begin{cases} \frac{4}{\pi} \int_0^{\pi/2} f(t) \sin(h\omega t) d(\omega t) & \text{per } h \text{ dispari} \\ 0 & \text{per } h \text{ pari} \end{cases}$

Figura 6: schema riassuntivo

Dal punto di vista del segnale, reale o complesso, pari o dispari, si possono ottenere ulteriori informazioni sullo spettro del segnale, sia sullo spettro delle ampiezze che sullo spettro delle fasi.

Considerando un segnale reale generico, il suo sviluppo in serie di Fourier in forma complessa (ma anche le altre versioni) gode della simmetria Hemitiana:

$c_{-k} = (c_k)^*$, dove l'asterisco indica l'operazione di coniugazione nel campo complesso e che coincide con il generico c_k nel caso questi sia un numero reale puro.

La simmetria coniugata, dunque, rende di fatto due equazioni che dettano la forma dello spettro del segnale. Si prende come esempio principe la funzione seno e ridenominando con "X" i coefficienti complessi c (si terr  tale notazione per tutto il resto dell'elaborato).

$$x(t) = a \sin(2\pi ft)$$

Figura 7:

Spettro delle ampiezze dei coefficienti

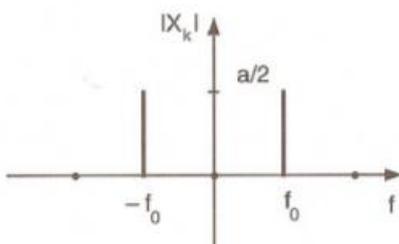
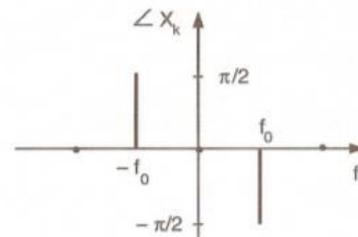


Figura 8:

Spettro delle fasi dei coefficienti



Essendo un segnale reale, infatti il seno rispetta la condizione:

$$X_{-k} = X_k^* \Leftrightarrow \begin{cases} |X_k| = |X_{-k}| \\ \angle X_k = -\angle X_{-k} \end{cases}$$

Infine, generalizzando in base alle simmetrie, si ha:

- $x(t)$ è una funzione pari $\rightarrow X_k$ è reale puro ed è una funzione pari di $k \rightarrow$ lo sviluppo trigonometrico ha soli coseni
- $x(t)$ è una funzione dispari $\rightarrow X_k$ è immaginario puro ed è una funzione dispari di $k \rightarrow$ lo sviluppo trigonometrico ha soli seni
- qualsiasi segnale né pari né dispari può essere parametrizzato come

$$x_p(t) + x_d(t) = x(t) \text{ dove } 2x_p(t) = x(t) + x(-t) \text{ e } 2x_d(t) = x(t) - x(-t).$$

Sulla base di tale scomposizione, per linearità dell'operazione di scomposizione in serie, si può scrivere la versione generalizzata di qualsiasi segnale periodico asimmetrico:

$$X_{pk} + X_{dk} = X_k$$

Il resto delle proprietà, come la linearità garantita dalla stessa operazione di integrazione, oppure il comportamento dei coefficienti di Fourier nei confronti di una traslazione del segnale, o di una operazione di derivata temporale, o di integrale, ecc.) si riassumono meglio usando il metodo simbolico dell'analisi fasoriale. Sapendo che un'infinità di armoniche corrisponde ad un'infinità di diagrammi fasoriali, ciascuno valido solo per fasori isofrequenziali, è possibile considerare il seno come coseno traslato e, con la legge di Eulero, esprimere i segnali in forma esponenziale e osservare diversi fatti importanti.

L'operazione di derivazione nel tempo si può osservare come una traslazione di 90 gradi:

Dominio del tempo	Dominio della frequenza
$a(t) = A_M \cos(\omega t + \alpha)$	$\bar{A} = A_M e^{j\alpha}$
$\frac{d}{dt} \cdot a(t) = -\omega A_M \sin(\omega t + \alpha) =$	$j\omega \bar{A} = \bar{C} = C_M e^{j\gamma}$
$= \omega A_M \cos(\omega t + \alpha + \frac{\pi}{2}) =$	$\begin{cases} C_M = \omega A_M \\ \gamma = \alpha + \frac{\pi}{2} \end{cases}$
$= C_M \cos(\omega t + \gamma)$	
$\begin{cases} C_M = \omega A_M \\ \gamma = \alpha + \frac{\pi}{2} \end{cases}$	

dal punto di vista del diagramma fasoriale si ha un'omotetia immaginaria:

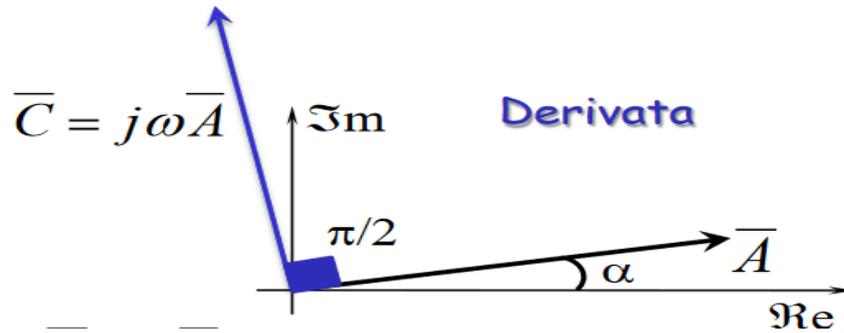


Figura 9: fasori

1.2 La trasformata di Fourier

Lo sviluppo di una funzione mediante le serie di Fourier è possibile, ed estremamente efficace, per le funzioni periodiche. In questo caso, le sole frequenze necessarie sono gli interi, oppure i multipli di ω per le funzioni aventi periodo T . Per le funzioni non periodiche, tutte le frequenze sono ammissibili e la serie deve essere sostituita da un integrale. I coefficienti X_k della forma complessa, che determinano le ampiezze di ogni armonica, diventano i valori della trasformata di Fourier calcolata in k , la quale risulta definita sull'intera retta reale, ossia è una funzione continua della variabile indipendente. Essa dà una misura della densità di ciascuna frequenza $k \in \mathbb{R}$ nella funzione.

1.2.1 Definizione generale

La procedura precedentemente descritta per la scomposizione di un segnale in serie di Fourier può essere applicata solo a segnali periodici. Si consideri tuttavia un segnale $x(t) : \mathbb{R} \rightarrow \mathbb{C}$ non periodico, ma che soddisfa la condizione di assoluta sommabilità. Sia infatti $x(t)$, $t \in \mathbb{R}$, è assolutamente integrabile su tutto \mathbb{R}

$$\int_{-\infty}^{+\infty} |x(t)| dt < +\infty$$

e sviluppabile in serie di Fourier su ogni intervallo $[0, T]$.

Si consideri il segnale periodico x_T di periodo T e sviluppabile in serie di Fourier.

$$x_T(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j2\pi n f_0 t}) = \sum_{n=-\infty}^{\infty} \left(\frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-j2\pi n f_0 t} dt e^{j2\pi n f_0 t} \right)$$

Al crescere di T , la funzione x_T si raccorda con la funzione x su intervalli sempre maggiori e la sommatoria si trasforma in un integrale

$$\lim_{T \rightarrow \infty} x_T(t) = x(t)$$

ed esprimendo il segnale in versione periodica tramite la serie complessa di Fourier, è possibile, con l'operazione di limite arrivare ad una definizione dove la somma di infiniti coseni e seni commuta in un integrale.

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$$

La cui definizione inversa che definisce la antitrasformata, detta anche Integrale di Fourier:

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft} df$$

La seconda delle due rappresenta evidentemente un'equazione di sintesi che permette di rappresentare il segnale come sovrapposizione di segnali elementari, ed è chiaramente analoga alla serie per i segnali periodici; la prima è un'equazione di analisi che permette di determinare il peso che le varie componenti frequenziali (a tutte le possibili frequenze variabili con continuità da $-\infty$ a $+\infty$) hanno nella composizione di $x(t)$. Tali relazioni mettono in corrispondenza un segnale del tempo con la propria trasformata di Fourier, funzione a valori complessi della frequenza.

In analogia a quanto visto per i coefficienti di Fourier si è soliti estrarre dalla funzione complessa $X(f)$ le funzioni reali modulo e fase, essendo la stessa funzione un numero complesso.

$$X(f) = R(f) + jI(f)$$

Se $x(t)$ è un segnale reale allora:

$$R(f) = \int_{-\infty}^{+\infty} x(t) \cos(2\pi ft) dt = R(-f)$$

$$I(f) = - \int_{-\infty}^{+\infty} x(t) \sin(2\pi ft) dt = -I(-f)$$

ovvero la parte reale della trasformata di un segnale reale è una funzione pari della frequenza, mentre la parte immaginaria ne è una funzione dispari. Queste simmetrie si possono riassumere con la simmetria Hermitiana: $X(f) = [X(-f)]^*$, dove l'asterisco indica l'operazione di coniugazione.

Le proprietà della funzione $x(t)$ si riflettono nella sua trasformata:

- se $x(t)$ è reale e pari :

$$R(f) = 2 \int_0^{+\infty} x(t) \cos(2\pi ft) dt$$

$$I(f) = 0$$

- se $x(t)$ è reale e dispari :

$$I(f) = -2 \int_0^{+\infty} x(t) \sin(2\pi ft) dt$$

$$R(f) = 0$$

Dunque si ha una riflessione dal dominio del tempo al dominio della frequenza: la trasformata di un segnale reale e pari è una funzione reale e pari della frequenza; la trasformata di un segnale reale e dispari è una funzione immaginaria pura e dispari.

Molto rapidamente, le proprietà della serie di Fourier possono essere riassunte secondo una tabella:

<i>dualità:</i>	$F\{X(t)\} = x(-f)$
<i>scalatura:</i>	$F\{x(at)\} = \frac{1}{ a } X\left(\frac{f}{a}\right)$,
<i>inversione del tempo:</i>	$F\{x(-t)\} = X(-f)$
<i>traslazione nel tempo:</i>	$F\{x(t + t_0)\} = X(f) e^{j2\pi f t_0}$,
<i>traslazione in frequenza:</i>	$F\{x(t) e^{j2\pi f_0 t}\} = X(f - f_0)$,
<i>coniugazione nel tempo:</i>	$F\{x^*(t)\} = X^*(-f)$,
<i>coniugazione in frequenza:</i>	$F\{x^*(-t)\} = X^*(f)$,
<i>derivazione nel tempo:</i>	$F\left\{\frac{d^n}{dt^n} x(t)\right\} = (j2\pi f)^n X(f)$,
<i>derivazione in frequenza:</i>	$F\{-j2\pi t\}^n x(t) = \frac{d^n}{df^n} X(f)$,
<i>integrazione nel tempo:</i>	$F\left\{\int_{-\infty}^t x(\tau) d\tau\right\} = \frac{X(f)}{j2\pi f} + \frac{1}{2} X(0) \delta(f)$, ($X(0) \hat{=} \text{area}[x(t)]$),
<i>convoluzione nel tempo:</i>	$F\{x(t) * y(t)\} = X(f) Y(f)$,
<i>convoluzione in frequenza:</i>	$F\{x(t) y(t)\} = X(f) * Y(f)$.

Figura 10: proprietà della FT

Infine, il Teorema di Parseval può essere riformulato così:

$$\int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |X(f)|^2 df$$

cioè, l'energia del segnale può essere espresso anche tramite l'integrale del quadrato della sua densità spettrale di energia. Ciò può interpretarsi fisicamente dicendo che l'energia totale di un segnale può essere calcolata sommando sia l'energia di una serie di suoi campioni nel tempo, che la densità spettrale di una serie di campioni nella frequenza. E' inevitabile dover chiamare il modulo quadro della trasformata come densità spettrale di energia: non verrà dimostrato il perché della scelta di tale formalismo. Comunque, la tesi di tale teorema rappresenta l'energia del segnale.

2 Il passaggio dal continuo al discreto

Come già accennato, il lavoro di tesi si occupa anche dello studio dei sistemi caratterizzati da segnali discreti, nel contesto della DFT. La stessa Trasformata Discreta di Fourier può essere ottenuta in due modi: discretizzando l'integrale del coefficiente di Fourier complesso e discretizzando la trasformata di Fourier. I due risultati sono analoghi.

E' fondamentale vedere perché è necessario dover discretizzare.

I principali metodi di analisi dei segnali possono essere riassunti nei concetti di "analisi nel dominio del tempo" e "analisi nel dominio della frequenza". E' importante osservare che questi due modi di affrontare un problema sono tra loro intercambiabili, nel senso che, sotto opportune condizioni, nessuna informazione viene persa nel passare da un dominio all'altro. Il vantaggio che deriva dall'introduzione dei due domini è la possibilità di cambiare la prospettiva con la quale si osserva un dato fenomeno. In questo modo un problema che appare di difficile soluzione in un dominio può risultare molto più semplice nell'altro. Lo strumento matematico che consente di trasferire lo studio dei segnali aperiodici e dei sistemi dal dominio del tempo al dominio della frequenza è la trasformata di Fourier. La trasformata di Fourier $X(f)$ di una funzione continua nel tempo $x(t)$ è data dalla relazione:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$$

Poiché in generale la $X(f)$ è complessa, essa può essere descritta mediante gli spettri di ampiezza e fase. La relazione che definisce la trasformata di Fourier non è direttamente implementabile mediante un elaboratore digitale di segnale, sia perché essa richiede l'analisi di segnali continui, $x(t)$ e $X(f)$, sia perché l'integrale si estende all'infinito e richiederebbe dunque un numero infinito di dati da elaborare. Per poter effettuare tale trasformazione con un sistema digitale sono dunque necessarie tre operazioni fondamentali: il campionamento, il troncamento del segnale e la discretizzazione dell'asse delle frequenze. Ciascuna di queste operazioni può influenzare significativamente l'attendibilità del risultato dell'analisi. Lo scopo delle considerazioni che seguono in questo capitolo è pertanto quello di studiare gli effetti di tali operazioni e arrivare alla definizione finale di Trasformata Discreta di Fourier.

2.1 Il campionamento dei segnali: leggi di Poisson

Per introdurre il concetto di trasformata di un segnale campionato e analizzare gli effetti del campionamento e le metodologie con cui eseguirlo, è necessario introdurre alcuni concetti chiave che saranno la base di alcuni esperimenti della parte pratica del lavoro di tesi.

Un risultato importante per poter introdurre quella che è la prima formula di Poisson è l'invarianza del treno di impulsi, o pettine delta, per trasformazione di Fourier.

Si consideri il segnale periodico di periodo T e che è sviluppabile in serie di Fourier:

$$x(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$

essendo, il coefficiente complesso di Fourier, costante per ogni valore dell'indice k , allora lo sviluppo sarà:

$$X_k = \frac{1}{T} \sum_{k=-\infty}^{+\infty} x(t) e^{-\frac{j2\pi kt}{T}} = \frac{1}{T}$$

La trasformata di Fourier si ottiene trasformando lo sviluppo in serie: trasformata della serie.

Pertanto lo spettro del segnale pettine di δ è anch'esso un pettine di δ :

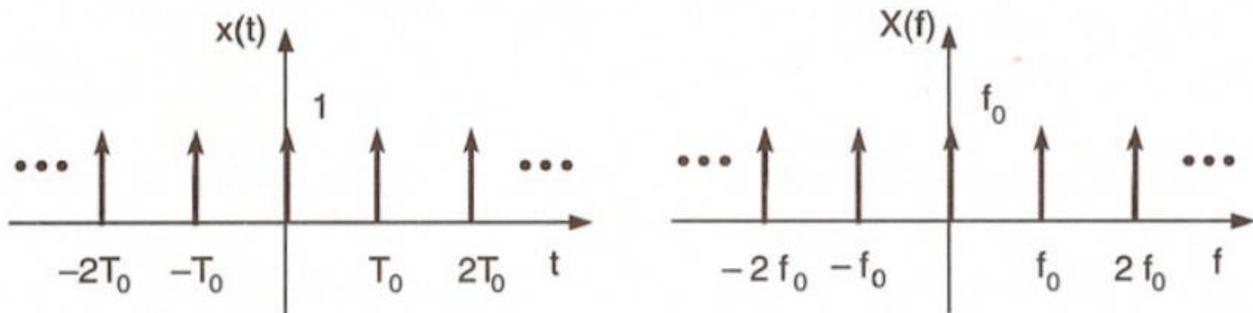


Figura 10: invarianza per trasformazione

Si consideri un segnale aperiodico $x(t)$, costruiamo il segnale $y(t)$ periodico di periodo T secondo la relazione di periodizzazione usata per il pettine delta:

$$y(t) = \sum_{n=-\infty}^{+\infty} x(t - nT)$$

Tale segnale ammette sviluppo in serie ed è interessante vedere cosa succede se si calcola il coefficiente complesso k -esimo della sua serie:

$$\begin{aligned} Y_k &= \frac{1}{T_0} \sum_{n=-\infty}^{+\infty} \int_{-T_0/2}^{T_0/2} x(t - nT_0) e^{-j2\pi k f_0 t} dt = \frac{1}{T_0} \sum_{n=-\infty}^{+\infty} \int_{-T_0/2 - nT_0}^{T_0/2 - nT_0} x(\alpha) e^{-j2\pi k f_0 (\alpha + nT_0)} d\alpha \\ &= \frac{1}{T_0} \sum_{n=-\infty}^{+\infty} \int_{-T_0/2 - nT_0}^{T_0/2 - nT_0} x(\alpha) e^{-j2\pi k f_0 \alpha} d\alpha \end{aligned}$$

La funzione integranda dell'ultimo passaggio non dipende dall'indice della serie n ; tale indice agisce infatti solo sugli estremi di integrazione. Ci si rende conto facilmente, che, al variare di n tra $-\infty$ e $+\infty$, gli intervalli di integrazione $[-T_0/2 - nT_0, T_0/2 - nT_0]$ della stessa funzione integranda ricoprono tutto l'asse reale senza sovrapposizioni. Pertanto l'espressione finale che consegue dalla periodizzazione di un segnale aperiodico è:

$$Y_k = \frac{1}{T} X\left(\frac{k}{T}\right)$$

che è un risultato fondamentale. La relazione trovata, detta di campionamento in frequenza, afferma che ogni riga dello spettro discreto del segnale periodico è componente dello spettro continuo del segnale aperiodico. In altre parole, i coefficienti di Fourier del segnale in versione periodica sono dunque, a meno di una riduzione di ampiezza, non sono i valori della trasformata continua, ma i valori campionati in corrispondenza delle frequenze armoniche multiple di quella del segnale periodico. Si nota che l'operazione di periodizzazione,

secondo un periodo T, nel dominio del tempo ha conseguito un campionamento in frequenza, ma con la frequenza di campionamento che è la stessa del segnale periodico.

Sostituendo l'espressione del coefficiente di Fourier nell'equazione dello sviluppo in serie, si ottiene la prima formula di Poisson:

$$\sum_{n=-\infty}^{+\infty} x(t - nT) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} \frac{1}{T} X\left(\frac{k}{T}\right) e^{\frac{j2\pi kt}{T}}$$

Se si tiene conto di tale relazione e la si applica ad un segnale del tipo $h(t) = |\cos(2\pi f_0 t)|$ e lo si periodicizza con un treno di impulsi, si avrà il risultato della seconda formula di Poisson:

$$y(t) = \sum_{k=-\infty}^{+\infty} x(t - kT_0)$$

$$X(f) = T_0 \frac{\text{sinc}((fT_0 + 1)/2) + \text{sinc}((fT_0 - 1)/2)}{4}$$

dove il segnale base è $x(t) = h(t)\text{rect}\left(\frac{t}{T_0/2}\right) \rightarrow$

ed il coefficiente di Fourier ottenuto dal campionamento:

$$Y_k = \frac{2}{T_0} X\left(\frac{2k}{T_0}\right) = \frac{\text{sinc}(k + 1/2) + \text{sinc}(k - 1/2)}{2}$$

Considerando y(t) sviluppato in serie, sostituendo l'espressione del coefficiente di Fourier e considerando la trasformata della sommatoria in cui c'è x(t), essendo la trasformata una funzione biunivoca lineare:

$$\sum_{n=-\infty}^{+\infty} X(t - nT_0) = \sum_{k=-\infty}^{+\infty} \frac{1}{T_0} x\left(-\frac{k}{T_0}\right) e^{j\frac{2\pi kt}{T_0}} \Rightarrow \sum_{n=-\infty}^{+\infty} X(t - nT_0) = \sum_{k=-\infty}^{+\infty} \frac{1}{T_0} x\left(\frac{k}{T_0}\right) e^{-j\frac{2\pi kt}{T_0}}$$

e applicando il Teorema di dualità:

$$\sum_{n=-\infty}^{+\infty} x(nT) e^{-j2\pi n f T} = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X\left(f - \frac{k}{T}\right)$$

Si è ottenuta la seconda formula di Poisson e questo, si vedrà più avanti, è un risultato fondamentale ed un passo enorme verso la definizione di Trasformata Discreta di Fourier.

2.2 Dal tempo continuo al tempo discreto

Campionare un segnale x(t) significa "estrarre" dal segnale stesso i valori che esso assume a istanti temporali equispaziati, cioè multipli di un intervallo T detto periodo di campionamento. Aprendo e chiudendo la porta dell'aula con una certa frequenza, l'insegnante può vedere solo una parte di ciò che accade e ciò che fanno, nel mentre, gli alunni: si perde dell'informazione. E' interessante capire quali siano le condizioni affinché non si perda informazione.

L'operazione di campionamento viene simbolicamente effettuata da un dispositivo, il campionatore, indicato con una sorta di "interruttore" che si chiude per un intervallo di durata infinitesima. La cadenza con cui l'interruttore si chiude, cioè con la quale il segnale viene campionato, è proprio la frequenza di campionamento. L'elaborazione del segnale viene eseguita sui valori digitali estratti dal segnale stesso tramite conversione A/D, e si risolve nell'esecuzione di un opportuno programma da parte del microprocessore



Figura 11: sistema di conversione

La nuova definizione di trasformata di Fourier di una sequenza è :

$$X(f) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T}$$

ed essa è una funzione continua della frequenza, periodica e di periodo quello di campionamento. Questa periodicità è caratteristica delle trasformate delle sequenze, e diventa chiara se introduciamo la relazione, inversa o di antitrasformazione:

$$x[n] = x(nT) = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X(f) e^{j2\pi n f T} df$$

Il significato della relazione appena enunciata, che è un'espansione della sequenza $x[n]$, è quello consueto per l'analisi di Fourier: il segnale dato viene espresso come sovrapposizione di un continuum di componenti frequenziali di ampiezza e fase regolate dall'andamento di $X(f)$. Mentre un segnale analogico necessita di componenti a tutte le frequenze sull'asse reale da $-\infty$ a $+\infty$, cioè in un ambito illimitato, per esprimere una sequenza in ambito frequenziale sono sufficienti le sole componenti a tutte le frequenze comprese nell'intervallo limitato $[-1/2T, 1/2T]$. In un certo modo, questo risultato è pienamente giustificato dalla periodicità della trasformata di una sequenza, nel senso che le sole componenti veramente significative sono quelle nel periodo base. Anche per la trasformata di una sequenza è comunque d'uso introdurre lo spettro di ampiezza $A(f) = |X(f)|$ e lo spettro di fase $\phi(f) = \angle X(f)$.

La periodicità della trasformata di una sequenza, e/o la limitatezza dell'intervallo frequenziale significativo nello spettro del segnale, sono conseguenze di un unico fenomeno: nell'ambito dei segnali tempo discreto, due oscillazioni sinusoidali (complesse) rispettivamente alla frequenza f_0 e $f_0 + m/T$ (m intero) sono indistinguibili

$$\exp[j2\pi(f_0 + m/T)nT] = \exp(j2\pi f_0 nT) \exp(j2\pi mn) = \exp(j2\pi f_0 nT)$$

Si parla di induzione della periodicizzazione.

Le proprietà della trasformata di Fourier di una sequenza (simmetria Hermitiana ecc.) relativamente a quelle del segnale temporale (segnale reale, segnale pari o dispari ecc.) sono sostanzialmente identiche a quelle a proposito della trasformata dei segnali analogici e dunque verranno trattate solo le proprietà della vera e propria Trasformata Discreta, che finora non è ancora stata enunciata.

Ai fini della comprensione, da parte del lettore, della parte sperimentale sulle sequenze e la convoluzione, sono enunciati i risultati dei teoremi di prodotto e convoluzione sulle sequenze numeriche aperiodiche, che poi, durante l'esercitazione, saranno confrontate con quelle in versione periodica.

2.3 Teoremi di prodotto e di convoluzione di sequenze numeriche aperiodiche

Si definisce una sequenza numerica $z[n]$ come somma di convoluzione tra due sequenze aperiodiche

$$z[n] = x[n] * y[n] = \sum_{k=-\infty}^{+\infty} x[n-k]y[k];$$

eseguendo la trasformata, si arriva a: $Z(f) = Y(f)X(f)$.

Se si considera una sequenza $p[n]$ come prodotto termine a termine delle stesse due sequenze e ne si calcola la trasformata, si giunge a:

$$P(f) = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} Y(f-u)X(u) du$$

L'integrale a secondo membro dell'ultima relazione rappresenta la cosiddetta convoluzione ciclica o periodica fra le due trasformate. La convoluzione ciclica è un'operazione che si definisce tra funzioni periodiche come le trasformate delle sequenze. Si nota che la funzione integranda è analoga a quella che si ha nella convoluzione cosiddetta lineare o aperiodica eseguita tra funzioni aperiodiche nel classico dominio tempo-continuo, ma l'integrale viene calcolato su di un solo periodo, e il risultato viene moltiplicato per l'ampiezza del periodo stesso.

2.4 Dimostrazione sulla Trasformata di una sequenza

Tramite l'operazione di campionamento, si è detto, è stato fatto un cambio di dominio: quello del tempo discreto. E' abbastanza facile ricavare la forma della trasformata di un segnale campionato e dunque l'espressione della trasformata di Fourier di una sequenza numerica che esce da un convertitore analogico digitale che ha campiona il segnale tempo-continuo $s(t)$.

Si consideri il segnale campionato:

$$s_c = \sum_{n=-\infty}^{+\infty} s(nT_c)\delta(t - nT_c)$$

Procedendo con la trasformazione della sequenza s_c che però è una funzione continua:

$$S_c(f) = \int_{-\infty}^{+\infty} s_c(t)e^{-j2\pi ft} dt$$

allora:

$$\int_{-\infty}^{+\infty} e^{-j2\pi ft} \sum_{n=-\infty}^{+\infty} s(nT_c)\delta(t - nT_c) dt$$

applicando il Teorema di linearità è possibile scambiare i due simboli di somma e integrale:

$$\sum_{n=-\infty}^{+\infty} s(nT_c) \int_{-\infty}^{+\infty} \delta(t - nT_c)e^{-j2\pi ft} dt$$

Con il cambio di variabile $nT_c + p = t$, si ottiene:

$$S_c(f) = \sum_{n=-\infty}^{+\infty} s(nT_c) e^{-j2\pi ft}$$

Alternativamente, di cui è omessa la dimostrazione perché costituita di singoli passi, già tutti dimostrati, è:

$$S_c(f) = \frac{1}{T_c} \sum_{n=-\infty}^{+\infty} S\left(f - \frac{n}{T_c}\right)$$

Ancora una volta, si ottiene lo spettro in banda base che viene traslato su frequenze centrate in multipli interi della frequenza di campionamento. Tutte le strade portano a Roma.

Spettro in banda base:

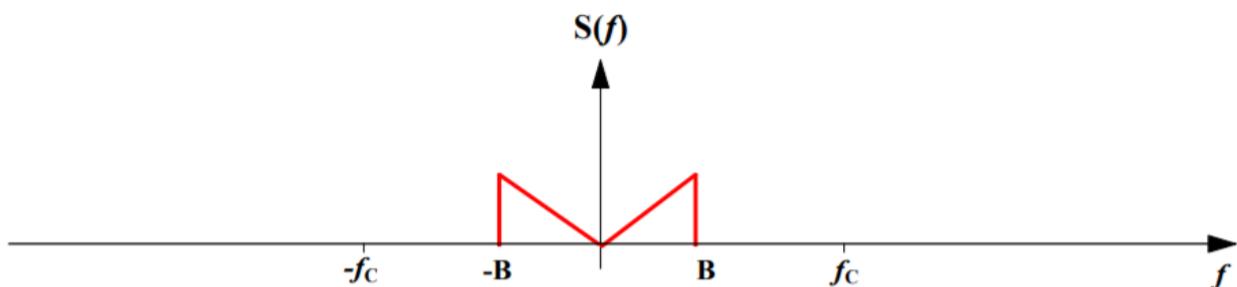


Figura 12: spettro in banda base del segnale

Replicato:

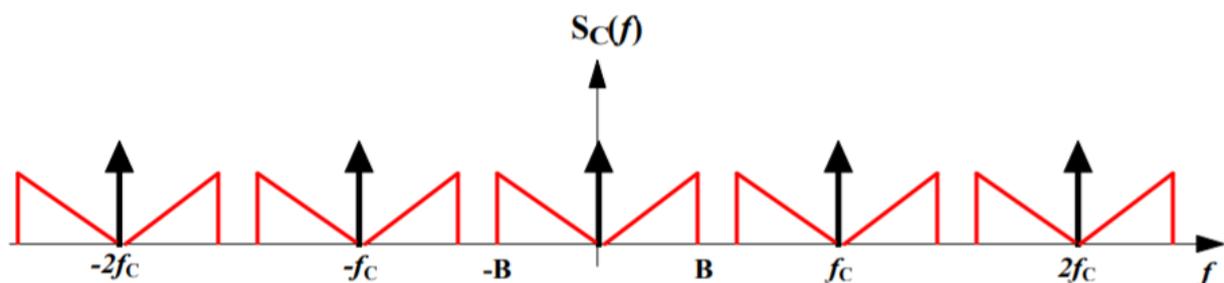


Figura 13: spettro di un segnale campionato

2.5 La condizione di Nyquist

Riprendere in considerazione il campionamento di un segnale a tempo continuo $x(nT) = x[n]$ e determinare le conseguenze in ambito frequenziale di questa relazione valida in ambito temporale è l'idea giusta per arrivare all'espressione che mette in risalto che effetto si abbia nel dominio della frequenza.

Come quanto ricavato dal precedente paragrafo, si scrive la trasformata del segnale campionato e la si denomina come $\overline{X}(f)$:

$$\sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T} = \sum_{n=-\infty}^{+\infty} x(nT) e^{-j2\pi n f T}$$

Esprimiamo adesso i campioni del segnale a tempo continuo $x(t)$ attraverso l'integrale di Fourier per spostarci nel dominio della frequenza ed indagare

$$\bar{X}(f) = \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} X(v) e^{j2\pi v n T} dv \right) e^{-j2\pi f n T} = \int_{-\infty}^{\infty} X(v) \sum_{n=-\infty}^{\infty} e^{-j2\pi n(f-v)T} dv$$

Sospensione del processo: si pensi al pettine di delta.

$$y(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_0)$$

Sviluppando in serie il pettine, ottenendo l'espressione del suo k-esimo coefficiente di Fourier complesso e applicando il Teorema di dualità, si ha l'espressione generale:

$$\sum_{n=-\infty}^{\infty} e^{-j2\pi n f T} = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta\left(f - \frac{k}{T}\right)$$

Grazie a questo risultato, è possibile riprendere l'equazione precedente e riscrivere l'espressione dell'integrale di Fourier come:

$$\int_{-\infty}^{+\infty} X(v) \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta\left(f - v - \frac{k}{T}\right) dv$$

Scambiando somma con integrale:

$$\frac{1}{T} \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} X(v) \delta\left(v - \left(f - \frac{k}{T}\right)\right) dv$$

Utilizzando la proprietà campionatrice dell'impulso si commuta all'espressione decisiva:

$$\bar{X}(f) = \frac{1}{T_0} \sum_{k=-\infty}^{+\infty} X(f - kf_0) = X_s(f)$$

dove il pedice s indica "sample" perché il segnale è campionato.

Questa relazione mostra che la trasformata di Fourier di una sequenza ottenuta per campionamento si ottiene come periodicizzazione della trasformata del segnale analogico di partenza, con un periodo di ripetizione in frequenza pari alla frequenza di campionamento $f_0 = \frac{1}{T_0}$.

Poiché la frequenza è misurata in Hertz, ma anche in sample/sec, allora è più appropriato usare un pedice che si riferisca ai sample.

$$f_s = \frac{1}{T_s}$$

In particolare, il segnale di partenza $x(t)$ ha banda f_m rigorosamente limitata. Si mostrano due situazioni che differiscono per il valore della frequenza di campionamento.

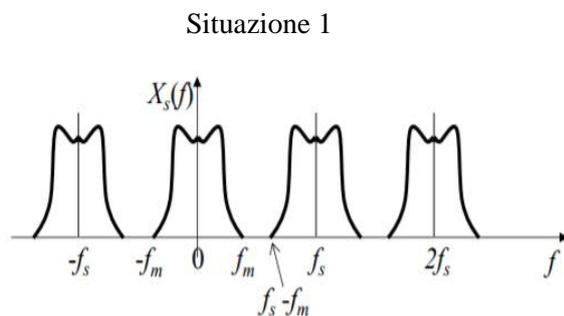


Figura 14

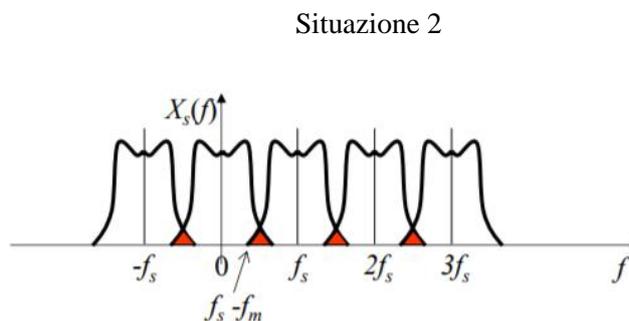


Figura 15

Affinché non esistano sovrapposizioni fra le repliche risulta evidente che il periodo di ripetizione in frequenza deve essere maggiore o al più uguale al doppio della banda del segnale, come nella prima situazione.

Se la frequenza di campionamento è maggiore almeno del doppio della massima frequenza contenuta nel segnale è possibile isolare lo spettro del segnale $X(f)$ in banda base e quindi ricostruire il segnale originario $x(t)$. Questa è la condizione di ricostruzione corretta del segnale, meglio conosciuta come condizione di Nyquist.

Se viceversa $f_s < 2f_m$, ossia se la velocità con cui si effettua il campionamento è insufficiente e quindi i campioni acquisiti sono troppo radi, non è possibile riottenere il segnale originario in alcun modo, a causa della sovrapposizione delle repliche che crea un disturbo da spettro adiacente. Tale fenomeno è detto aliasing e sarà oggetto di prove sperimentali svolte con Matlab.

Da quanto appena detto si evince che condizione necessaria affinché si possa eseguire correttamente la trasformata di Fourier sul segnale campionato $x_s(t) = x(nT_s)$ è che il segnale originario $x(t)$ sia a spettro limitato.

2.6 Interpolazione

La ricostruzione di un segnale a tempo-continuo a partire da una sequenza viene realizzata mediante un interpolatore. I vari tipi di interpolazione possono, in un certo senso, considerarsi come una generalizzazione dell'operazione compiuta in pratica da un convertitore D/A per fornire in uscita un segnale a tempo continuo $x(t)$ a partire dai valori (rappresentati su di un certo numero di cifre binarie) di una sequenza $x[n]$. Lo strumento che esegue il campionamento del segnale e la successiva interpolazione, senz'alcuna elaborazione intermedia, è rappresentato come cascata di due blocchi A/D e D/A, ovvero come successione di un campionatore ideale e di un interpolatore a mantenimento.

L'operazione svolta da quest'ultimo componente è: costruire il segnale analogico di uscita. Il valore n -esimo della sequenza d'ingresso $x[n]$ viene mantenuto a partire dall'istante nT e fino a che non sia disponibile (all'istante $(n+1)T$) il successivo valore $x[n+1]$.

Si può scrivere l'espressione del segnale interpolato $x(t)$ in funzione dei valori della sequenza $x[n]$ e tale espressione è indipendente dal tipo di interpolatore usato

$$\hat{x}(t) = \sum_{n=-\infty}^{+\infty} x[n] p(t - nT)$$

Per l'appunto, il segnale interpolato è una convoluzione discreta ed è immediato sapere che forma abbia la sua trasformata.

Si faccia attenzione ai vari segnali in gioco:

- le varie versioni del segnale originale

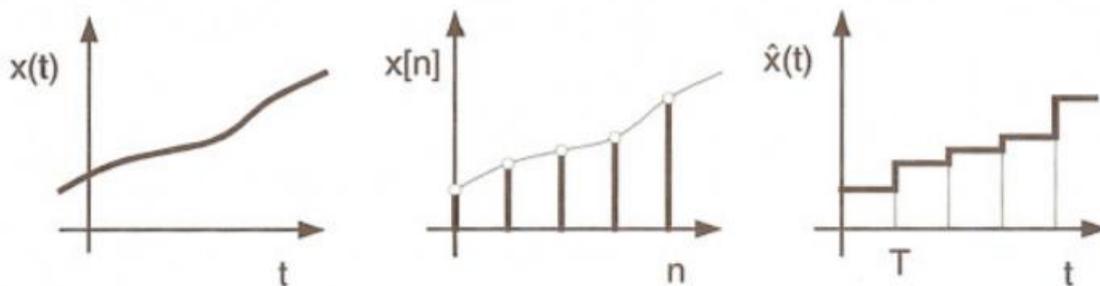


Figura 16: schema di campionamento e quantizzazione

- l'operazione di interpolazione a mantenimento:

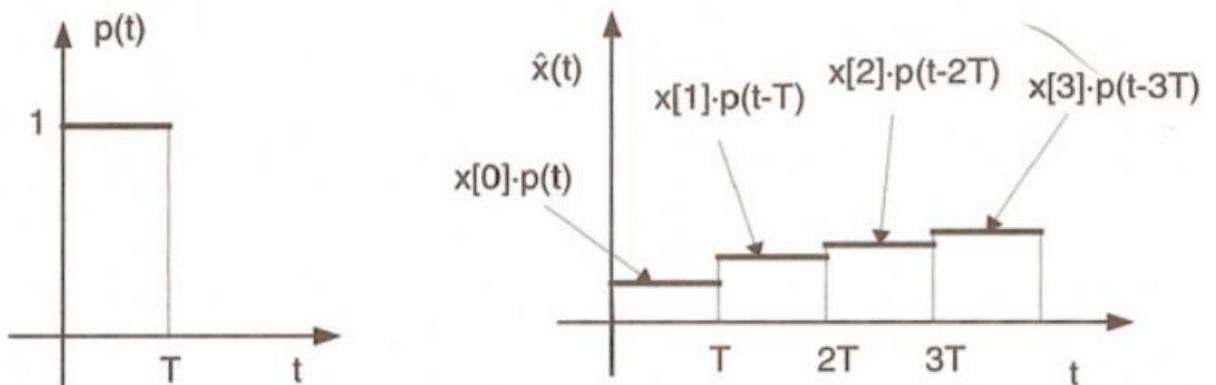


Figura 17: interpolatore ed interpolazione

ove $p(t)$ è per l'appunto l'impulso rettangolare.

Si hanno altre versioni di interpolazione, ossia si può progettare un interpolatore scegliendo un segnale differente da far convolvere con il segnale campionato (dato che la stessa definizione di segnale interpolato è una convoluzione discreta). Vengono esposti altri due esempi che verranno implementati tramite script Matlab.

Interpolazione versione cardinale:

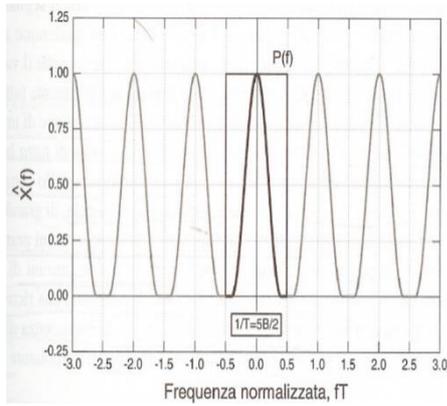


Figura 18

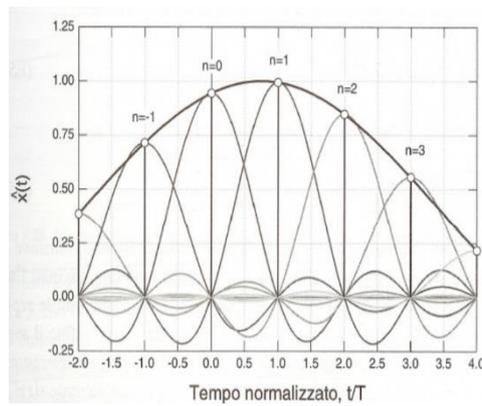


Figura 19

Interpolazione versione lineare:

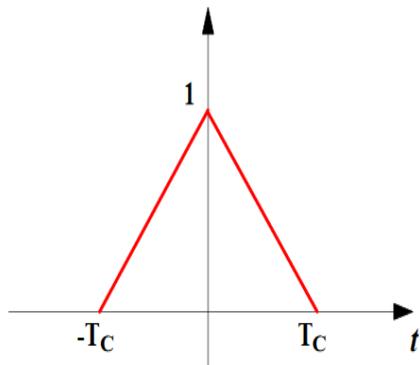


Figura 20

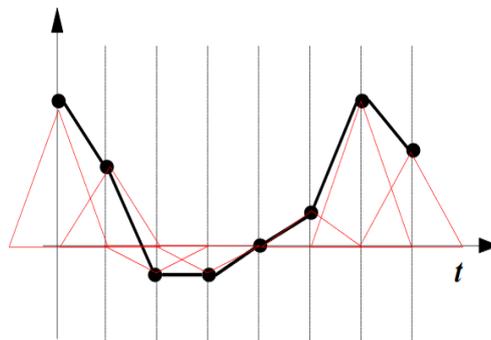
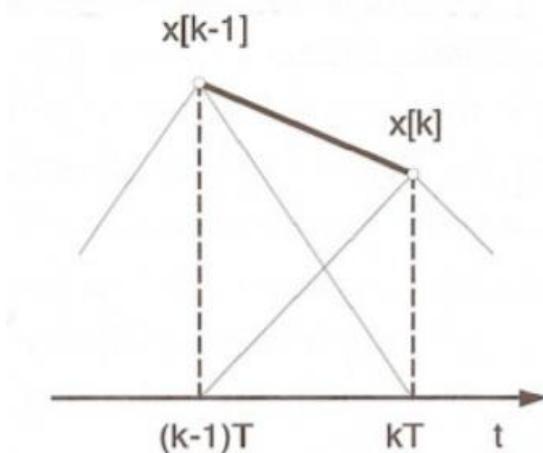


Figura 21

L'azione di ricostruzione esercitata da un interpolatore lineare, ossia un interpolatore con funzione di risposta all'impulso di tipo triangolare: ogni triangolo è collocato in corrispondenza degli istanti di campionamento ed è scalato di una quantità pari al campione cui si riferisce. Si può scrivere anche l'equazione che rappresenta tale operazione, poiché proveniente dalla geometria del piano:



$$\hat{x}(t) = x[k-1] p(t - (k-1)T) + x[k] p(t - kT)$$

Figura 22: interpolazione lineare

Per giustificare analiticamente figura di fianco, osserviamo che nel generico intervallo $[(k-1)T, kT)$, compreso fra i due campioni consecutivi $x[k-1]$ e $x[k]$, solo due addendi della sommatoria della definizione di $\hat{x}(t)$ danno un contributo non nullo, cioè quelli con $n = k-1$ e $n = k$.

Si consideri l'interpolatore cardinale. L'espressione del segnale interpolato è facilmente scrivibile nel dominio del tempo:

$$\hat{x}(kT) = \sum_{n=-\infty}^{+\infty} x[n] \operatorname{sinc}(k - n) = \sum_{n=-\infty}^{+\infty} x[n] \delta[k - n] = x[k] = x(kT)$$

Questo risultato conferma che il segnale interpolato coincide con il segnale di partenza negli istanti di campionamento. Se si considera un qualunque altro istante non coincidente con uno di quelli di campionamento, si nota che il valore del segnale interpolato è ottenuto dalla combinazione lineare di tutti gli infiniti campioni $x[n]$ del segnale $x(t)$. In altre parole, la ricostruzione di un segnale a banda limitata a un certo istante richiede la conoscenza di tutta la sequenza di campioni del segnale stesso, in istanti sia antecedenti quello considerato, sia successivi. Pertanto la formula di interpolazione cardinale, di grande rilevanza teorica, è inutilizzabile nella sua forma esatta nelle applicazioni pratiche per due motivi: in primo luogo, sono in teoria richiesti infiniti termini di una sommatoria per ricostruire il segnale originario ed è già stato visto, dal punto di vista elettronico, che non è possibile; secondariamente, una ricostruzione in tempo reale è impossibile perché si richiederebbe la conoscenza di valori di segnale in istanti successivi a quello di interpolazione (interpolatore non causale).

2.7 Troncamento del segnale

Si è già visto che l'analisi di Fourier si applica formalmente a segnali di durata infinitamente estesa e pertanto anche la sequenza dei campioni che rappresenta il segnale in forma discreta dovrà essere teoricamente di lunghezza infinita. Tale ipotesi non è ovviamente realizzabile nella pratica. In generale, con riferimento a un processo di campionamento reale, la sequenza dei campioni avrà necessariamente un inizio e una fine e pertanto il numero dei campioni a disposizione sarà un numero finito.

Il troncamento avviene semplicemente come: $x_w(t) = x(t) \cdot w(t)$, dove:

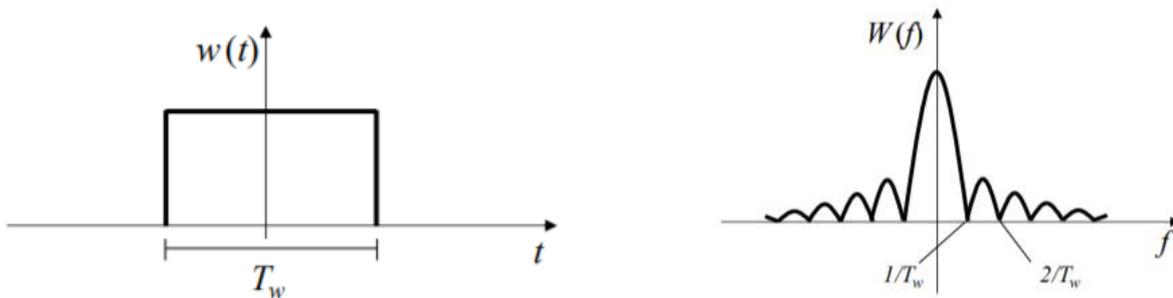


Figura 23: finestra rettangolare e relativo spettro delle ampiezze

Il cui corrispettivo effetto, nel dominio della frequenza, può facilmente visualizzarsi con un coseno che ha uno spettro bilatero e dunque non cambia molto la situazione:

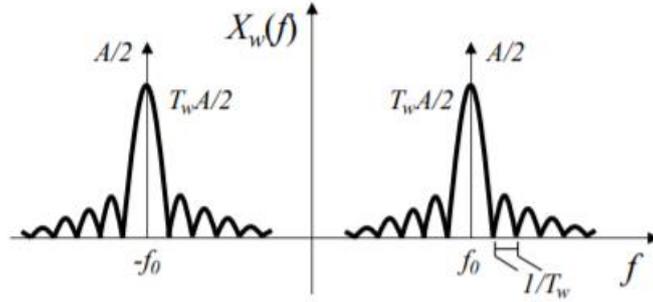


Figura 24: replica spettrale nel caso di un coseno

L'entità della dispersione in frequenza dipende dalla durata T_w della finestra di osservazione e dal suo andamento temporale. In particolare l'andamento nel tempo della finestra di troncamento determina l'ampiezza dei lobi laterali della dispersione e risulta quindi direttamente responsabile della accuratezza con cui viene stimato lo spettro del segnale troncato. Sotto questo aspetto, concreti vantaggi possono essere ottenuti ampliando, entro limiti accettabili dal punto di vista pratico, la durata T_w o utilizzando finestre temporali non rettangolari, ma con transizione più graduale delle estremità (smoothing windows), delle quali si parlerà più avanti.

2.8 La Trasformata discreta di Fourier : come nasce

Come si è detto, i passi per poter arrivare alla trasformata in versione discreta sarebbero stati tre. Il terzo ed ultimo passo verrà eseguito durante il procedimento sottoriportato e verrà giustificato.

Si consideri la versione classica di trasformata di Fourier: quella di un segnale tempo continuo e a banda limitata.

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt$$

Essendo il segnale generico:

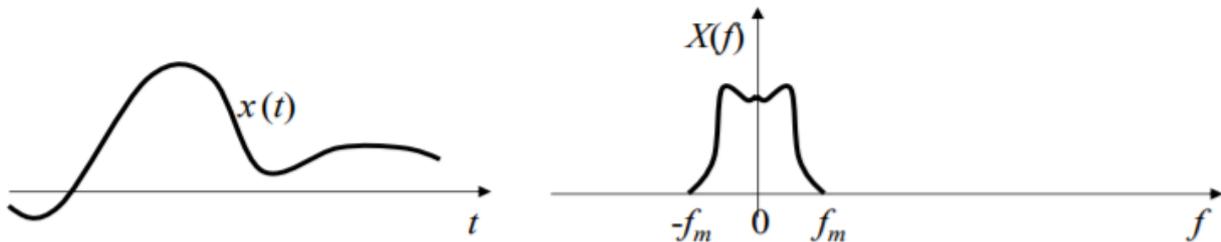


Figura 25: "ad un segnale a durata infinita può corrispondere una banda limitata"

Il primo passo è quello di considerare il campionamento del segnale che, come visto più volte, produce il seguente risultato:

$$x_s(t) = x(t) \cdot s(t) = x(t) \cdot \sum_{i=-\infty}^{+\infty} \delta(t - iT_s) = \sum_{i=-\infty}^{+\infty} x(iT_s) \cdot \delta(t - iT_s)$$

La definizione diventa quindi:

$$X(f) = \int_{-\infty}^{+\infty} x(t_n) e^{-j2\pi f t_n} dt$$

Per quanto detto, il segnale deve essere troncato.

L'integrale è in realtà una somma discreta di Riemann e dunque il differenziale diventa un delta e l'integrale diventa una somma:

$$X(f) = \sum_{n=0}^{+T_w-1} x(t_n) e^{-j2\pi f t_n} \Delta x$$

Si osservi che le reciproche relazioni sono:

$$NT_s = T_w$$

$$Nf_w = f_s$$

$$t_n = nT_s$$

$$f_k = kf_w$$

Dunque la discretizzazione avviene, in realtà, in entrambi i domini:

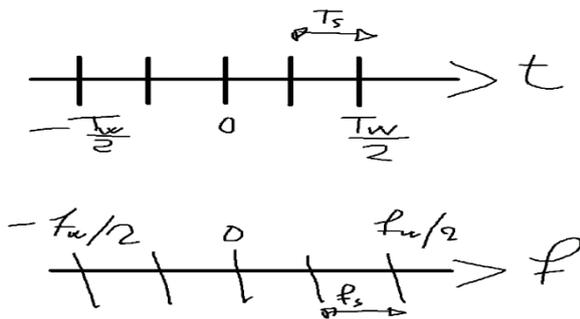


Figura 26: discretizzazione equivalente nei due domini

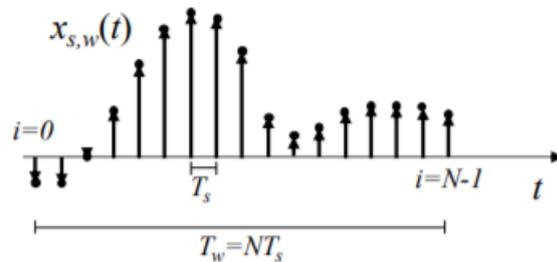


Figura 27: data-set

$$X(f) = \Delta x \sum_{n=0}^{N-1} x(nT_s) e^{-j2\pi f n T_s}$$

La ripetizione dello spettro in frequenza dipende dal campionamento nel tempo, così come il campionamento in frequenza è dovuto alla periodicità del segnale nel tempo. Il legame di trasformazione fra i campioni nel tempo $x_n = x(nT_s)$ e i campioni in frequenza $X_k = X(kf_w)$ è dato dalla trasformata discreta diretta e inversa di Fourier, che si sta cercando di ottenere. E' ora di utilizzare le reciproche relazioni al completo.

$$X(kf_w) = \Delta x \sum_{n=0}^{N-1} x(nT_s) e^{-j2\pi k f_w n T_s}$$

Ossia, la versione definitiva sapendo che $x(nT_s) = x[n]$

$$X(kf_w) = \frac{T_w}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

dove

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Si potrebbe re-iterare la dimostrazione con il coefficiente di Fourier complesso, ma è inutile. Basti pensare ad una cosa. Il segnale è troncato: dunque considerando la periodizzazione del data-set, a questo punto lo sviluppo in serie di Fourier contiene armoniche con frequenze multiple della f_w : ossia ad un certo punto il coefficiente di Fourier e la trasformata di Fourier già discretizzata sono direttamente proporzionali come:

$$X_k = f_w X(kf_w)$$

Dunque diventa banale poter affermare che, a differenza della discretizzazione tramite la trasformata, con la discretizzazione del coefficiente della serie complessa allora si arriva direttamente alla DFT. Ergo, la DFT è periodica, per natura del suo kernel.

Dunque, ricapitolando il passaggio dal tempo continuo al tempo discreto:

-definizione di sequenza aperiodica tramite antitrasformata di Fourier:

$$x[n] = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X(f) e^{j2\pi n f T} df$$

-definizione di trasformata di Fourier di una sequenza aperiodica:

$$\overline{X(f)} = \sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T}, (\overline{X(f)}) \text{ indica che è periodica}$$

-relazione del campionamento nel tempo tra segnale discreto e segnale analogico:

$$x[n] = x(nT)$$

-sviluppo in serie di Fourier in forma complessa di un segnale periodico tempo-continuo:

$$x(t) = \sum_{n=-\infty}^{\infty} (X_n e^{j2\pi n f_0 t})$$

-definizione di coefficiente complesso della serie di Fourier di un segnale tempo-continuo:

$$X_n = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-j2\pi n f_0 t} dt$$

-definizione di sequenza periodica tramite antitrasformata di Fourier discreta (IDFT)

$$x[n] = \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}$$

-definizione di trasformata di Fourier discreta (DFT) di una sequenza periodica:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Si osservi che, a differenza di quanto avviene per la trasformata continua, in questo caso il segnale tempo-discreto $x[n]$ e ciascun coefficiente X_k vengono espressi mediante le stesse unità di misura. Ciò è dovuto al fatto che, mentre la $X(f)$ rappresenta una densità spettrale, i coefficienti X_k rappresentano direttamente i valori delle singole componenti spettrali. In pratica, di tutte le possibili armoniche di ordine k , per un segnale $x(t):\mathbb{R}\rightarrow\mathbb{R}$ solo le prime $N/2$ sono significative e portano informazione, in quanto le successive $N/2$ risultano speculari rispetto alla frequenza di folding $f_s/2$ e coniugate.

Il fattore di scala che si era trovato con la formula di Poisson era $1/T$, ossia la formula era:

$$Y_k = \frac{1}{T} X\left(\frac{k}{T}\right)$$

Dove era avvenuta una periodizzazione del segnale aperiodico. Qua invece è avvenuto un troncamento del data-set e si è periodizzato e, per coerenza, si è ottenuto:

$$X_k = \frac{1}{T_w} X(kf_w)$$

La ricostruzione è dunque ben posta ed è giustificata in ogni passo.

3 La DFT

In tale capitolo verranno affrontate le proprietà più importanti dello “strumento DFT” e le proprietà della matrice di Vandermonde dietro il passaggio da campione nel tempo a campione nella frequenza (applicazione lineare vettoriale).

Dare un’occhiata alla DFT con il linguaggio dell’algebra lineare è l’ideale!

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] w_N^{-nk}$$

Questa sarà la nuova notazione usata per tutto il capitolo.

3.1 La matrice W

Non è strano affermare che la DFT, ottenuta dalla discretizzazione di una funzione biunivoca, sia ancora un operatore come tale. Essa prende in input un data-set periodico e restituisce un data-output periodico complesso, se non lo è già direttamente l’input. Insomma, la matrice W è sicuramente invertibile e dunque il suo determinante è non-nullo, ossia essa è non-singolare ed è una matrice complessa NxN, ossia $W \in \mathbb{C}^{N \times N}$.

Più formalmente:

si pensi di avere il seguente input di dati:

$$x = [x(0) \ x(1) \ \dots \ x(N-1)], \ x \in \mathbb{C}^{N \times 1}$$

e sia il seguente vettore degli output:

$$X = [X(0) \ X(1) \ \dots \ X(N-1)], \ X \in \mathbb{C}^{N \times 1}$$

E’ possibile rappresentare in forma compatta il sistema lineare derivante: $X = W x$, dove la matrice dei coefficienti delle N radici dell’unità immaginaria è data dal prodotto:

$$W = \frac{1}{N} \begin{pmatrix} \omega_N^0 & \omega_N^0 & \omega_N^0 & \dots & \omega_N^0 \\ \omega_N^0 & \omega_N^{-1} & \omega_N^{-2} & \dots & \omega_N^{-(N-1)} \\ \omega_N^0 & \omega_N^{-2} & \omega_N^{-4} & \dots & \omega_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_N^0 & \omega_N^{-(N-1)} & \omega_N^{-2(N-1)} & \dots & \omega_N^{-(N-1)(N-1)} \end{pmatrix}$$

Figura 28: matrice delle radici ennesime

Alcune osservazioni:

- simmetria $W^T = W$.
- quasi Hermitiana $W^{-1} = N W^*$

- Per $N=4$, gli autovalori sono: $\lambda_1 = \sqrt{N}$, $\lambda_2 = -\sqrt{N}$, $\lambda_3 = 0 - i\sqrt{N}$, $\lambda_4 = 0 + i\sqrt{N}$.

3.2 Proprietà della trasformata

Non essendo scontato, vale la pena ribadire che esistono diversi formalismi di DFT, a livello di indicizzazione. Lo stesso Matlab mette a disposizione funzioni di FFT che mostrano lo spettro centrato rispetto all'indice 0 oppure ad un altro valore (come è, ovviamente, possibile anche con la trasformata continua).

Quanto alle proprietà, non c'è nulla da dimostrare. La trattazione fino alla nascita della DFT e la trattazione sulle trasformate di sequenze e segnali continui consentono al lettore di intuire il motivo di ciascuna proprietà. Per quelle meno scontate sarà aggiunto un commento con eventuale dimostrazione, se la proprietà è di rilievo in campo sperimentale come in analisi spettrale, digital filtering e signal processing.

- 1) Periodicità: $w_N^{-n(k+N)} = w_N^{-nN} w_N^{-nk}$ oppure $w_N^{(n+N)k} = w_N^{nN} w_N^{nk}$
- 2) Linearità.

E' giusto usare una notazione compatta ed indicare con $D\{\}_k$ l'operatore di DFT.

$$D\{ax[n]+by[n]\}_k = aD\{x[n]\}_k + bD\{y[n]\}_k$$

- 3) Shift e Modulazione.

Esprimendo la sequenza, shiftata di un intero j , tramite IDFT:

$$x[n-j] = D^{-1}\{X_k w_N^{-jk}\}_n \text{ applicando l'operatore DFT diretto, segue subito la proprietà di shift.}$$

La modulazione è semplicemente la trasformata di una sequenza moltiplicata per un'esponenziale complessa: $D\{w_N^{nj} x[n]\}_k = X_{k-j}$.

- 4) Simmetria Hermitiana: $x[n] = D\{NX_k^*\}^*$ oppure $X_k = D^{-1}\{x[n]^* 1/N\}^*$ dove l'asterisco si riferisce all'operazione di coniugazione del numero complesso.
- 5) Trasformata di un data-set di reali puri: la condizione $x[n]=x[n]^*$ si ripercuote sulle armoniche come $X_k^* = X_{-k}$.
- 6) DFT di sequenze pari/dispari.

Riepilogando le condizioni di simmetria: la sequenza è pari se verifica $x[-n]=x[n]$ dunque le sue armoniche verificano la medesima condizione; dispari se verifica $x[-n]=-x[n]$ dunque le sue armoniche verificano, ancora, la medesima condizione.

- 7) Ogni sequenza asimmetrica può essere scomposta in componenti di sequenze pari e dispari, come nel caso dei segnali periodici né pari né dispari. Banale ribadire che questo, di fatto, sia possibile perché la sequenza proviene da un campionario che non mutua queste proprietà.
- 8) Convulsione ciclica

Tale proprietà sarà affrontata in fase sperimentale, dunque ci si limita a illustrare l'essenziale.

Prese due sequenze periodiche $x[n]$ e $y[n]$, se convolute, tale operazione di alta complessità computazionale si può svolgere anche tramite la DFT, essendo la definizione:

$$z[n] = x * y = \frac{1}{N} \sum_{k=-\infty}^{+\infty} x[n-k]y[k]$$

Considerando inizialmente la trasformata della sequenza z :

$$\bar{Z}_k = \frac{1}{N} \sum_{n=0}^{N-1} z[n] e^{-j2\pi kn/N}$$

Dunque i tre passi da eseguire sono:

$$\bar{Z}_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] * y[n] e^{-j2\pi kn/N}$$

$$D^{-1}\{D\{z[n]\}\} = D^{-1}\left\{\frac{1}{N} \sum_{n=0}^{N-1} x[n] * y[n] e^{-j2\pi kn/N}\right\}$$

$$D^{-1}\{D\{z[n]\}\} = D^{-1}\left\{\frac{1}{N} \bar{X}_k \bar{Y}_k\right\}$$

Ossia il prodotto di convoluzione ciclica commuta in tre steps: trasformata delle due sequenze, prodotto termine a termine, antitrasformata.

Non vi è altro da commentare.

Infine, tutti i teoremi su ritardi, energia e quant'altro, sono sostanzialmente gli stessi. Prendendo un riassunto delle proprietà dal libro "An Owner's Manual for the Discrete Fourier Transform", che usa una notazione sostanzialmente uguale a quella della trattazione, si riportano tutte le proprietà:

Property	Input property	DFT property	Fourier series property
Periodicity	Arbitrary	$f_{n+N} = f_n$ $F_{k+N} = F_k$	$f(x+A) = f(x)$
Linearity	Arbitrary	$\mathcal{D}\{\alpha f_n + \beta g_n\}_k$ $= \alpha F_k + \beta G_k$	FC $\{\alpha f(x) + \beta g(x)\}$ $= \alpha c_k + \beta d_k$
Shift	Arbitrary	$\mathcal{D}\{f_{n-j}\}_k = \omega_N^{-jk} F_k$	FC $\{f(x-y)\}$ $= e^{-i2\pi ky/A} c_k$
Modulation	Arbitrary	$\mathcal{D}\{f_n \omega_N^{nj}\}_k$ $= F_{k-j}, j \in \mathbf{Z}$	FC $\{f(x)e^{i2\pi jx/A}\}$ $= c_{k-j}, j \in \mathbf{Z}$
Hermitian symmetry	Arbitrary	$f_n = \mathcal{D}\{NF_k^*\}^*$ $F_k = \mathcal{D}^{-1}\{f_n/N\}^*$	-
Reversal	Arbitrary	$\mathcal{D}\{f_{-n}\}_k = F_{-k}$	FC $\{f(-x)\} = c_{-k}$
Real	$f_n \in \mathbf{R}$ $f(x) \in \mathbf{R}$	$F_{-k} = F_k^*$	$c_{-k} = c_k^*$
Conjugate symmetric	$f_{-n} = f_n^*$ $f(-x) = f^*(x)$	$F_k^* = F_k$	$c_k^* = c_k$
Even	$f_{-n} = f_n$ $f(-x) = f(x)$	$F_{-k} = F_k$	$c_{-k} = c_k$
Odd	$f_{-n} = -f_n$ $f(-x) = -f(x)$	$F_{-k} = -F_k$	$c_{-k} = -c_k$
Real even	$f_n, f(x) \in \mathbf{R}$ $f_{-n} = f_n$ $f(-x) = f(x)$	$F_{-k} = F_k$ $F_k \in \mathbf{R}$	$c_{-k} = c_k$ $c_k \in \mathbf{R}$
Real odd	$f_n, f(x) \in \mathbf{R}$ $f_{-n} = -f_n$ $f(-x) = -f(x)$	$F_{-k} = -F_k$ $iF_k \in \mathbf{R}$	$c_{-k} = -c_k$ $ic_k \in \mathbf{R}$
Convolution	Arbitrary	$\mathcal{D}\{f_n * g_n\}_k = NF_k G_k$	-
Correlation	Arbitrary	$\mathcal{D}\{f_n \otimes g_n\}_k = NF_k^* G_k$	-

Figura 29: tabella del libro

3.3 Derivazione della DFT da minimi quadrati

Le derivazioni mostrate finora si sono evolute dal problema dell'approssimazione dei coefficienti della serie di Fourier e della trasformata di Fourier di una particolare funzione (segnale troncato). Un altro modo segue per scoprire la DFT considerando il problema dell'approssimazione (o fitting) di un insieme di dati con una funzione nota come polinomio trigonometrico. L'obiettivo è trovare una combinazione lineare di seno e coseno che "meglio" si avvicina a un dato set di dati: è un'occasione anche per mettere in luce la periodicità della trasformata discreta e la semplificazione del calcolo complessivo per sequenze numeriche (data-set) reali.

Si suppone che ci si riceva un data-set del tipo:

$$P_n = (x_n, f_n) \text{ con } n = \left[-\frac{N-1}{2}, \frac{N-1}{2}\right], \text{ ossia una successione di punti nel piano cartesiano.}$$

E' utile richiamare quelle che sono la relazione sul campionamento nel tempo.

$$NT_s = T_w \text{ e } x_n = nT_s$$

Gli x_n variano dunque nell'intervallo T_w . Per non rendere la trattazione troppo astratta, si può pensare che un A/D converter abbia prodotto il seguente set di dati (campionamento di un segnale nel dominio del tempo) e si vuole effettuare la ricostruzione. Dunque non è molto diverso dall'interpolazione a mantenimento che effettua il dispositivo interpolatore, visto nella sezione 2 e successivamente implementato su Matlab in fase sperimentale. Qua, però, si vuole ottenere la DFT con l'intenzione di interpolare dei punti e dunque cercare di derivare la sua espressione analitica proprio da questo processo.

Si parla di cercare la migliore approssimazione possibile, in termini di polinomio trigonometrico, che, come visto nella sezione 1, è stato analizzato in diverse forme: rettangolare (trigonometrica pura con seno e coseno), armonica (solo coseno) e complessa (applicata legge di Eulero). Per far ciò, si sceglie la forma complessa perché nel contesto della minimizzazione è anche la più comoda da dover derivare (nel contesto della derivata).

Il polinomio trigonometrico usato è:

$$\psi_N(t) = \sum_{k=-(N-1)/2}^{(N-1)/2} a_k e^{j\frac{2\pi kt}{T_w}}$$

E' abbastanza immediato capire il senso di tale espressione. La serie di Fourier complessa vede il coefficiente k -esimo moltiplicato per il kernel della serie (esponente positivo) e la sommatoria, di cui sopra, è di fatto una serie di Fourier troncata, con il periodo di osservazione che coincide con quello di troncamento del segnale, usato nella sezione due. Non è essenziale considerare l'operazione di troncamento, perchè si sta discutendo un altro contesto, ma per coerenza di trattazione si considera come periodo di ripetizione del data-set proprio quello di osservazione del segnale campionato.

Si prosegue con tale filosofia:

‘Si devono determinare i coefficienti del polinomio affinché si abbia l'errore minimo’.

Si tratta di ricercare il minimo di una funzione quadratica, detta “discrete least squares error”:

che è una funzione a valori reali, non negativa e di N variabili, ossia i coefficienti α_k .

Operativamente si può rappresentare così:

$$\min_{\alpha \in \mathbb{R}^n} \|f_n - \psi_N(t)\|^2 \rightarrow \nabla E(\alpha_{-(N-1)/2}, \dots, \alpha_{(N-1)/2}) = 0,$$

dove si ricerca il vettore α di \mathbb{R}^n , avente come componenti i coefficienti che minimizzano l'errore.

Pertanto, una condizione necessaria, per il Teorema di Fermat, ossia per la minimizzazione di $E(\cdot)$, è che le k-derivate parziali si annullino: ergo il gradiente della funzione scalare si annulli.

$$\frac{\partial E}{\partial \alpha_k} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left[e^{-i2\pi nk/N} \left(f_n - \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p e^{i2\pi np/N} \right) \right] = 0$$

Riarrangiando i termini:

$$\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n e^{-i2\pi nk/N} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p e^{i2\pi np/N} e^{-i2\pi nk/N}$$

notando che è possibile applicare il famoso Lemma di ortogonalità, in forma complessa, allora si scrive:

$$\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk} = \sum_{p=-\frac{N-1}{2}}^{\frac{N-1}{2}} \alpha_p \underbrace{\sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_N^{(p-k)n}}_{N \delta_N(p-k)}$$

dove la delta di Kronecker ha il seguente formalismo di definizione:

$$\delta_{ij} := \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

dove, in linea ancora più generale e più correlata con il caso in esame, gli indici i e j possono essere anche multipli interi tra loro. Di fatto, dunque, la sommatoria con il kernel complesso (gli esponenziali complessi) conduce al risultato N, perchè si somma 1 per N volte.

$$\alpha_k = \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_n \omega_N^{-nk}$$

Confrontando questo coefficiente con l'espressione del coefficiente complesso della DFT, si nota che questa espressione è proprio la DFT stessa.

Un breve esempio di polinomio interpolante per un data-set di dodici punti dentro un periodo unitario potrebbe essere:

$$\psi_{12}(x) = \sum_{k=-5}^6 \alpha_k e^{i2\pi kx}$$

Supponendo, ulteriormente, che il data-set sia a valori reali si ritrova la proprietà della DFT di sequenze numeriche reali:

$$\text{Re} \{ \alpha_k \} = \text{Re} \{ \alpha_{-k} \}$$

$$\text{Im} \{ \alpha_k \} = -\text{Im} \{ \alpha_{-k} \}$$

Applicando la legge di Eulero si può riscrivere tutto come:

$$\psi_{12}(x) = \alpha_0 + 2 \sum_{k=1}^5 (\text{Re} \{ \alpha_k \} \cos(2\pi kx) - \text{Im} \{ \alpha_k \} \sin(2\pi kx)) + \alpha_6 \cos(12\pi x)$$

Tale costruzione verrà ripresa nella sezione 4.1.3

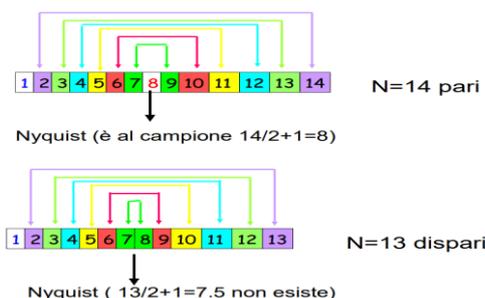
3.4 Nozioni sulla parte sperimentale

Poiché la parte sperimentale del lavoro di tesi è stata concentrata su ambiti ben specifici, durante lo studio, sono stati effettuati dei passaggi preliminari che vengono inseriti per dare prova di alcune caratteristiche della DFT.

Come già visto, il pedice “k” è privilegiato per le componenti spettrali, dunque le varie frequenze avranno un significato diverso per valori “importanti” (tipo la DC-component ha il pedice 0). Se la sequenza è reale, la DFT ha modulo con simmetria pari intorno alla frequenza di Nyquist:

$$\begin{aligned} \text{real}(F_k) &= \text{real}(F_{N-k}) & k=1, \dots, N/2 \\ \text{imag}(F_k) &= -\text{imag}(F_{N-k}) & k=1, \dots, N/2 \end{aligned}$$

Ossia F_{N-k} sono i complessi coniugati di F_k , non danno informazioni aggiuntive e spesso non è necessario visualizzarli. Osserviamo che non c'è simmetria per F_0 . Se individuiamo con $k=0$ la frequenza di Nyquist si ha $F_k = F_{-k}^*$ (complesso coniugato). Negli schemi seguenti si mettono in evidenza i campioni che godono della simmetria nel caso di N pari e dispari (gli indici sono shiftati ad 1 poiché in Matlab i vettori cominciano da 1, la DC component ha indice 1, quella di Nyquist $N/2+1$).

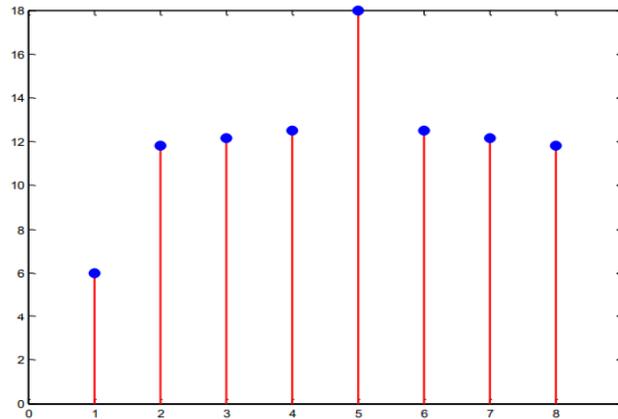


Si verifica la proprietà di simmetria nei due casi con alcuni esempi in Matlab:

```
>> f=[4 3 7 -9 1 0 0 0]';      %N=8
>> y=fft(f)                    % function Matlab che cal-
cola la DFT y =
 6.0000                        DC component
11.4853 - 2.7574i
-2.0000 -12.0000i
-5.4853 +11.2426i
18.0000                        Nyquist
-5.4853 -11.2426i
-2.0000 +12.0000i
11.4853 + 2.7574i
```

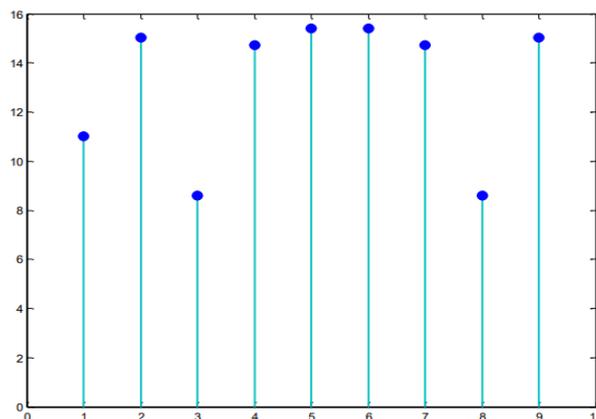
La prima componente è la DC component e la quinta è quella corrispondente alla frequenza di Nyquist. Si verifica la IDFT e se si ottiene il vettore di partenza:

Il grafico del modulo è:



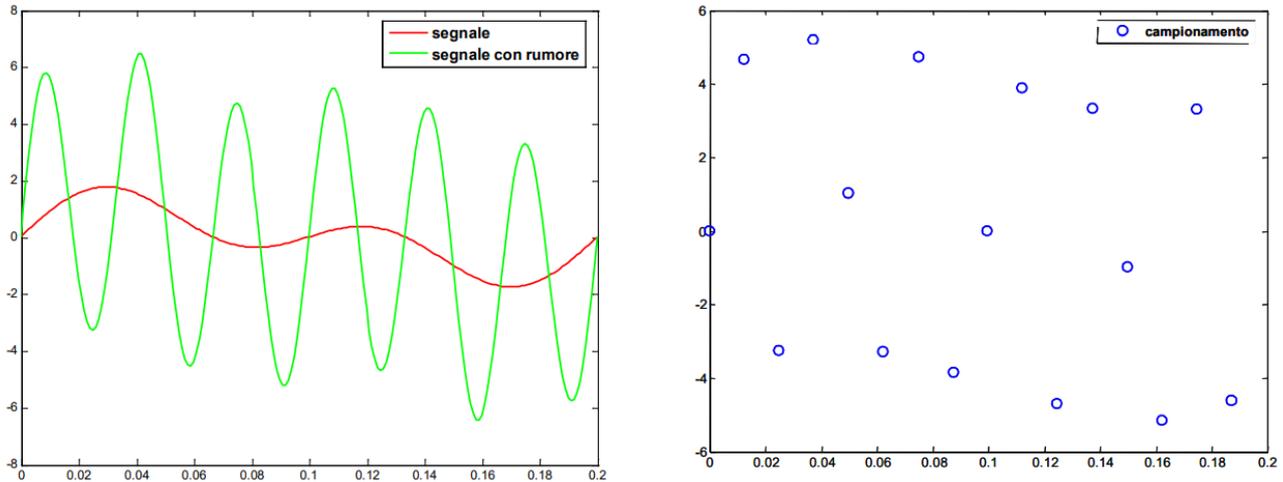
Si vede nel grafico come vi sia simmetria attorno alla frequenza di Nyquist tranne che per la componente DC.

Se si considera $f=[4 3 7 -9 1 0 0 0 5]'$, si ottiene:

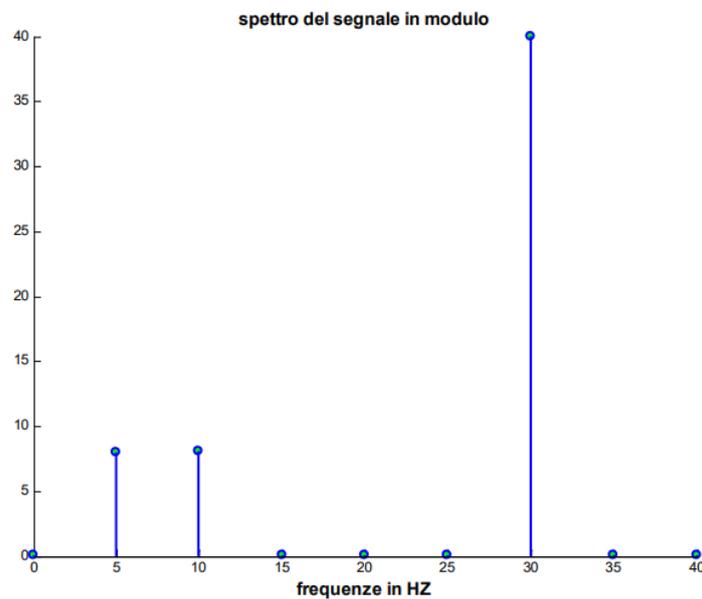


Un altro aspetto importante è il filtraggio che la DFT ci può regalare. Supponiamo che il segnale ricevuto è il segnale originale $f(t)=\sin(2\pi 5t)+\sin(2\pi 10t)$ a cui si è sovrapposto il "rumore" $5\sin(2\pi 30t)$. Siano poi, i parametri di costruzione della DFT (mantenendo la notazione usata nel paragrafo 2.8):

$T_s=0.0125$, $T_w=0.2$, $N=16$, si ha $1/T_w=5$, $1/T_s=80$, $1/(2T_s)=40$



Applicando la DFT si ottengono i valori delle ampiezze delle armoniche relative alle frequenze 0,5,10,15,...40. Ricordando che, se il segnale è reale, la DFT ha modulo con simmetria pari intorno alla frequenza di Nyquist: $F_k=F_{-k}$, $k=0=Nyquist$. Di seguito vi è il l'output del programma matlab che produce il grafico del modulo (la prima metà per la simmetria).



A questo punto, come avviene nella realtà, l'individuazione del rumore dipende dalle informazioni sul segnale e dagli scopi del filtraggio: si opera una decimazione dei campioni. Supponiamo di sapere che il segnale originale non può avere frequenze superiori a 10 Hz. Eliminando quindi le armoniche con frequenza > 10 (ricordando la proprietà di simmetria), si ottiene il segnale ripulito.

Inizialmente si aveva il vettore data-set con rumore e che trasformato dava i seguenti valori:

fy =

```

3.472833387713614e-014
2.975657543793466e-014 +7.999999999999966e+000i
-9.967995541639870e-015
+8.000000000000004e+000i
2.479866865842304e-014 -2.031708135064037e-
014i
4.530267607331881e-015 -1.776356839400251e-014i
1.858141972052217e-014 -6.472600233564663e-014i
-8.533243355846108e-014
+4.000000000000000e+001i
1.487958690795757e-014
+8.437694987151190e-015i
-2.922051234127287e-014
1.487958690795757e-014 -8.437694987151190e-015i
-8.533243355846108e-014 -
4.000000000000000e+001i
1.858141972052217e-014
+6.472600233564663e-014i
4.530267607331881e-015 +1.776356839400251e-014i
2.479866865842304e-014 +2.031708135064037e-014i
-9.967995541639870e-015 -
8.000000000000004e+000i
2.975657543793466e-014 -
7.999999999999966e+000i

```

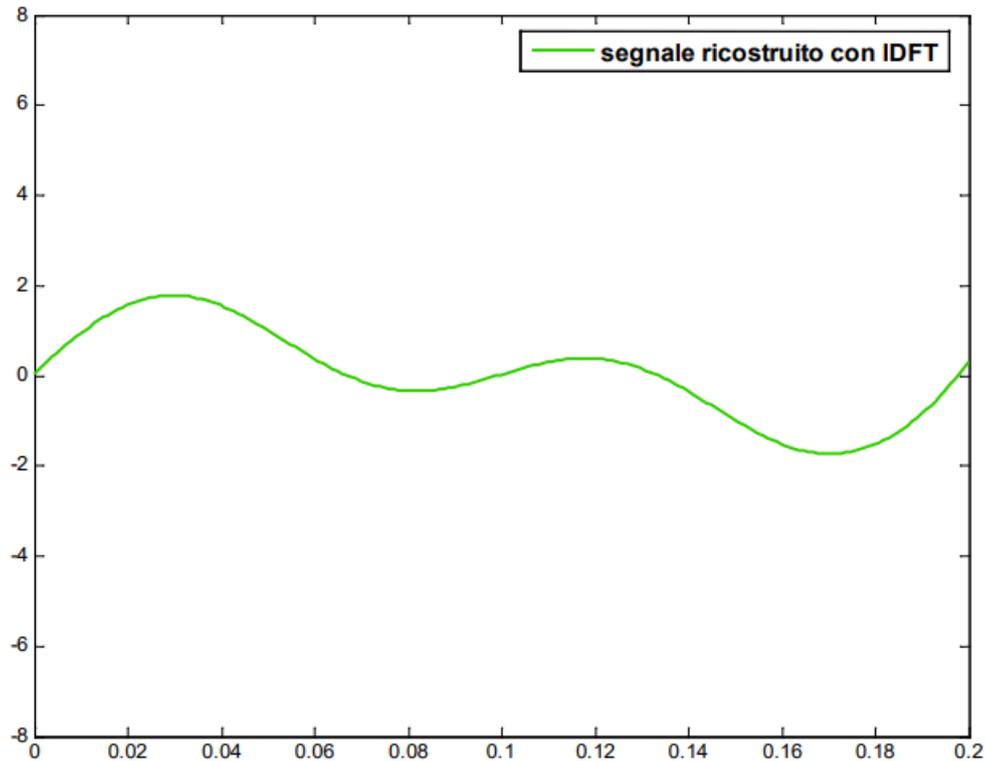
Effettuando la decimazione sopra i 10kHz:

```

3.472833387713614e-014
2.975657543793466e-014+7.999999999999966e+000i
-9.967995541639870e-015+8.000000000000004e+000i
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
9.967995541639870e-015-8.000000000000004e+000i
2.975657543793466e-014 -7.999999999999966e+000i

```

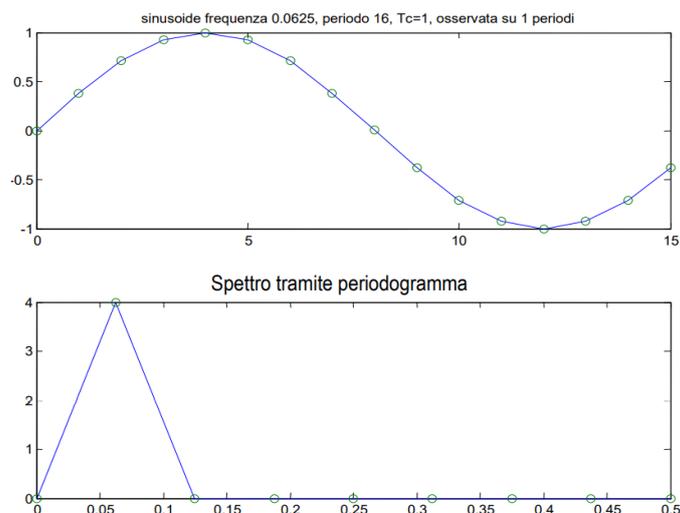
Applichiamo al vettore così ottenuto la IDFT ed otteniamo il segnale originale (si può effettuare un'interpolazione , ad esempio con spline cubiche).



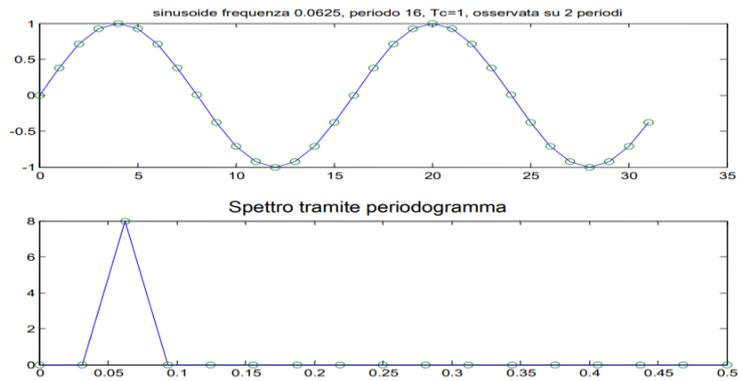
Potrebbe non essere scontato per il lettore e, dunque, si riprende cosa sia successo. Affermare che si filtra sopra i 10 kHz equivale, a livello di indice, eliminare tutti i samples che non rispettano $k=0,5,10$, essendo la frequenza di campionamento proprio 5. Allora, nell'output di matlab, si vedono le prime righe non nulle. Di fatto la sequenza reale porta a simmetria ed $N=16$ (che è pari!) e dunque induce asimmetria nell'altro "quadrante" e i samples nel dominio della frequenza risultano solo due.

Un altro importante concetto, nonché rimedio su diversi problemi, è lo zero-padding. Tale rimedio è solo cosmetico: mirato più all'apparenza che alla sostanza, soprattutto nel campionamento. Vediamo perché.

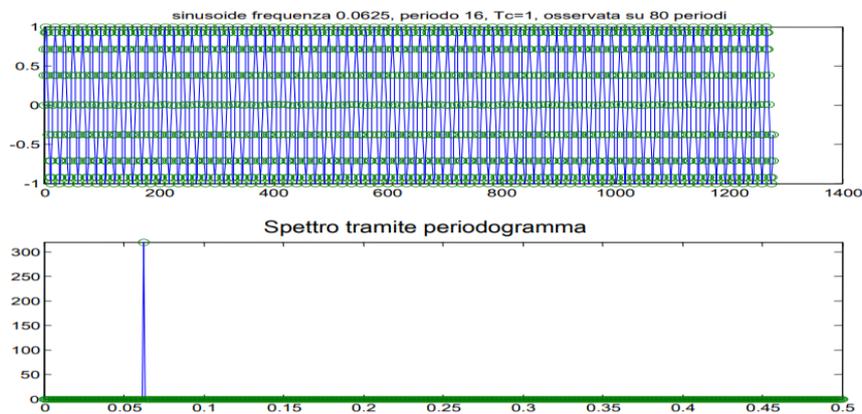
Si consideri un segnale sinusoidale di periodo $T_o=T_w=16$ s, un periodo di sampling $T_s=1$ s e, dunque, 16 campioni.



Si nota subito dallo spettro del segnale che la DFT ha modificato lo spettro originale del seno. In questo caso particolare il periodo di osservazione e quello del segnale coincidono. Si sperimenta cosa succede se la finestra di osservazione (periodo T_w) contiene due periodi del segnale.

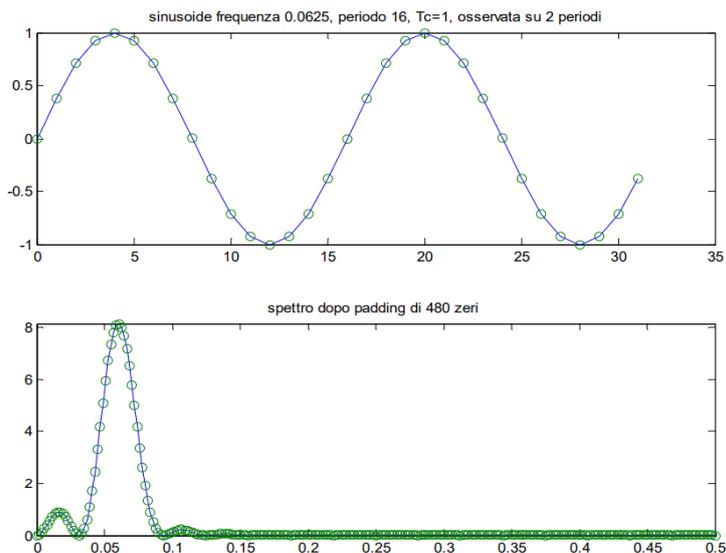


Com'è facilmente osservabile, l'impulso si restringe all'allargarsi della finestra. Infatti risulta che con 80 periodi venga fuori, praticamente, un impulso vero.

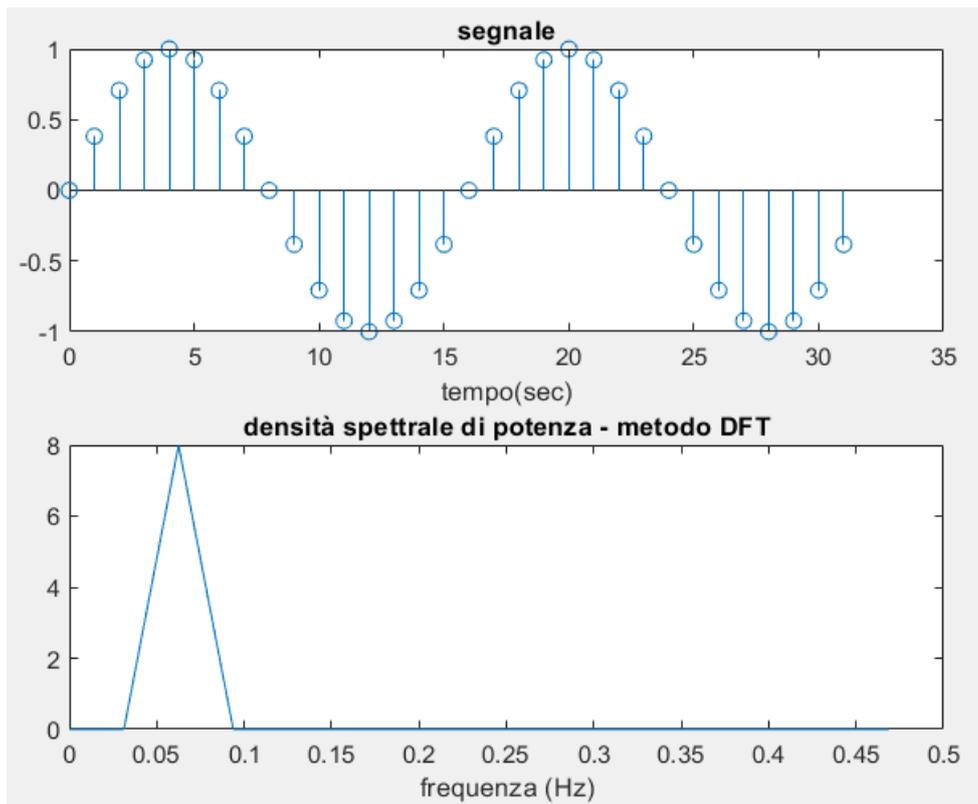


Dunque la qualità dello spettro è aumentata all'aumentare della finestra di osservazione, infatti la f_w è la risoluzione in frequenza.

Se invece si considerano 512 campioni, di cui 16 campionati nel periodo fondamentale del segnale e altri 16 in una ripetizione del segnale (dunque considerando il periodo della finestra come due volte quello del segnale originale da campionare), con 480 zeri (zero padding), si ottiene uno spettro che non è né un impulso e né un triangolo:



Per stimare lo spettro di potenza del segnale sinusoidale si può usufruire della funzione Matlab che effettua lo zero_padding:



E' possibile dimostrare analiticamente perché si ha il picco di 8. Il segnale sinusoidale ha una funzione di autocorrelazione (funzione di tau) cosinusoidale di ampiezza medesima, ma quadra e divisa per due. Eseguendo la trasformata di Fourier di tale funzione si arriva alla densità spettrale di potenza che integrata restituisce il risultato classico per una sinusoide. Passando al tempo-discreto, tali risultati confermano quanto plottato.

3.5 Complessità computazionale

E' utile introdurre che cosa sia la FFT e in cosa risulterà utile la sua applicazione e perché è così importante.

Considerando un calcolatore elettronico, si suppone di avere disponibile nella memoria gli N valori base di una sequenza periodica $x[n]$ e volerne calcolare numericamente la trasformata discreta di Fourier per eseguire un'analisi spettrale. Si vuole stimare il numero di operazioni che lo stesso calcolatore deve eseguire per ricavare gli N valori della trasformata.

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}}$$

Notando che in termini di numero di operazioni della trasformata, si ha lo stesso anche per una antitrasformata, allora le considerazioni che seguono valgono per entrambi i casi:

$$x[n] = \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi kn}{N}}$$

Purchè i valori $x[0], \dots, x[N-1]$ siano numeri complessi e si ignori il fattore di scala della trasformata diretta. Dedotto ciò, per entrambe le definizioni si devono effettuare le stesse operazioni sul kernel, salvo un inessenziale cambiamento di segno all'esponente. In questo modo, si può unificare la valutazione della complessità dell'algoritmo di DFT e di quello di IDFT. Supponendo che i fattori esponenziali complessi siano precalcolati, cioè già disponibili in memoria, la determinazione del coefficiente X_k (ovvero del campione $x[n]$)

richiede N moltiplicazioni complesse ed $N-1$ addizioni complesse. Tenendo conto che un'addizione complessa richiede in realtà due addizioni reali, e una moltiplicazione complessa richiede quattro moltiplicazioni reali e due addizioni, sono necessarie complessivamente $8N-2$ operazioni (reali) per ogni valore di k . Poiché k assume tutti i valori compresi tra 0 e $N-1$, il numero complessivo di operazioni da compiere per calcolare la trasformata discreta di Fourier di una sequenza periodica di periodo N è pari a

$$N_{\text{DFT}}(n) = n(8n-2) = 8n^2 - 2 \rightarrow N \cong 8n^2$$

ove l'ultima approssimazione ha senso per un numero di operazioni enorme e quindi una mole di dati consistente. Si può notare che la complessità di computazionale della trasformata discreta è di tipo quadratico nell'ordine n di trasformazione. Se si immagina di disporre di un elaboratore che svolge operazioni con una cadenza f_{clock} pari a 100 MHz (cioè 100 milioni di operazioni al secondo), e si immagina di prendere $n=1024$, il tempo necessario per effettuare il suddetto calcolo è pari a

$$T_n = n \cdot T_{ck} \cong 80 \text{ms}$$

Si suppone ora che la sequenza $x[n]$ venga generata campionando un segnale analogico $x(t)$ a frequenza f_c , e che si vogliano calcolare le trasformate discrete ripetute su spezzoni temporali adiacenti (le cosiddette finestre) di N campioni consecutivi della sequenza. Se in un tempo pari a 80 ms riusciamo a elaborare 1024 campioni, allora per poter operare in tempo reale si deve utilizzare una frequenza di campionamento tale che sia soddisfacente la condizione:

$$f_c \leq \frac{n}{T_n} = 12.8 \text{ kHz}$$

altrimenti nuovi campioni di una nuova finestra di segnale verrebbero presi prima che il precedente calcolo della trasformata sia terminato. Questo risultato rappresenta un vincolo sulla banda B del segnale $x(t)$ da elaborare poiché la condizione di Nyquist richiede che valga $B \leq f_c / 2$.

Un deciso miglioramento della velocità di elaborazione può essere conseguito utilizzando un algoritmo veloce di calcolo della trasformata discreta che, sfruttando particolari simmetrie insite nei fattori w_N^{nk} della trasformata stessa, riduce la complessità computazionale del medesimo. Tale algoritmo, noto come Fast Fourier Transform (FFT), usato dallo stesso Matlab con la funzione `fft` e altre versioni, a parità di frequenza di clock dell'elaboratore, permette l'utilizzo di frequenze di campionamento notevolmente maggiori rispetto a quella indicata dalla relazione precedente. Per arrivare a tale risultato è necessario svolgere alcuni calcoli.

Di seguito viene descritto un algoritmo molto popolare nell'ordinamento dei vettori e che è noto come algoritmo a decimazione nel tempo. Si considera $N=2^M$.

Si consideri ancora, a meno del fattore di scala, la definizione:

$$X_k = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{N}}$$

Da cui, suddividendo gli addendi in due gruppi:

$$\sum_{m=0}^{N_0/2-1} x[2m] e^{-j \frac{2\pi(2m)k}{N_0}} + \sum_{m=0}^{N_0/2-1} x[2m+1] e^{-j \frac{2\pi(2m+1)k}{N_0}}$$

La prima sommatoria rappresenta la trasformata discreta di una sequenza costituita dagli $N/2$ campioni di indice pari di $x[n]$, mentre la seconda sommatoria è la trasformata discreta degli $N/2$ campioni di indice dispari. Dunque non è stato cambiato niente finora. Si può dire che questa scomposizione è "ricorsiva nell'ordine", nel senso che la trasformata di ordine N è espressa come combinazione lineare di due trasformate di ordine $N/2$. Riflettendo questa scomposizione nella complessità computazionale dei numeri complessi con cui si ha a che

fare, indicata con la funzione $N_{fft}(n)$, la complessità computazionale in funzione del numero n di operazioni, si ha:

$$N_{fft}(n) = N_{fft}(n/2) + N_{fft}(n/2) + 6n + 2n = 2 [N_{fft}(n/2) + 4n] \quad \text{passo 1}$$

$$N_{fft}(n/2) = 2[N_{fft}(n/4) + 4n/2]$$

$$N_{fft}(n) = 2[2[N_{fft}(n/4) + 4n/2] + 4n] = 4 N_{fft}(n/4) + 2(8n) \quad \text{passo 2}$$

$$N_{fft}(n/4) = 2[N_{fft}(n/8) + 4n/4]$$

$$N_{fft}(n) = 4[2[N_{fft}(n/8) + n]] + 2(8n) = 8 N_{fft}(n/8) + 3(8n) \quad \text{passo 3}$$

E' possibile dimostrare anche in tre passi (passo iniziale, passo buono e passo induttivo), a partire dalla forma finale della tesi, che l'espressione della complessità computazionale è:

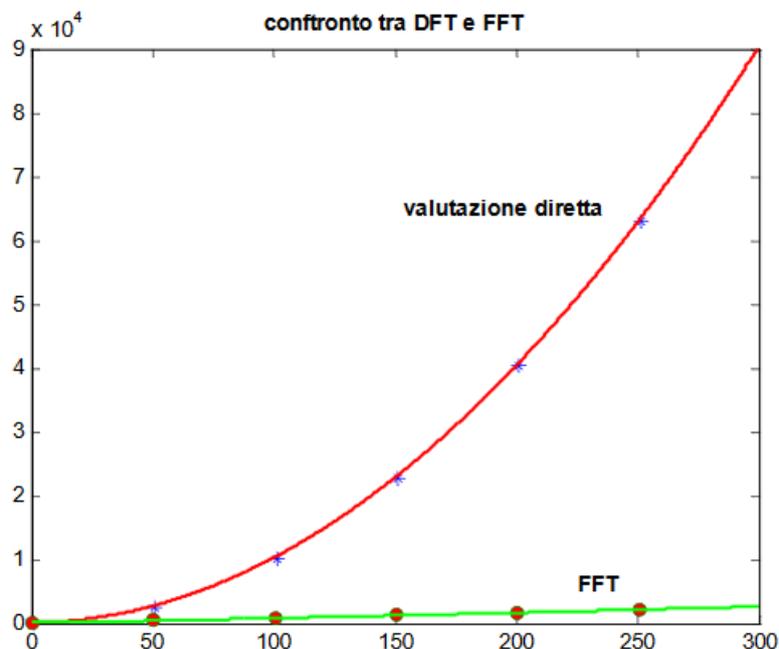
$$N_{FFT}(n) = nN_{FFT}\left(\frac{n}{2}\right) + \log_2(n)8n = \log_2(n)8n = O(n\log_2 n).$$

Valutando gli aspetti di prima, è interessante vedere come cambia la frequenza di campionamento. Il rapporto tra il numero di operazioni necessarie nei due casi è

$$\frac{N_{DFT}}{N_{FFT}} = \frac{N}{\log_2 N}$$

Tra l'altro, questo è il fattore di forma che ci permette di ottenere la nuova frequenza, secondo una proporzione ed essendo maggiore di 1, allora necessariamente sarà aumentata. Usando le quantità di prima, si ottiene un fattore di forma di 100 e dunque una frequenza di 1.28MHz.

Un confronto grafico:



3.6 Conclusione della trattazione

Il lavoro di Tesi è stato posizionato al centro dell'analisi del passaggio dal tempo-continuo al tempo-discreto, dettato dalla DFT. Tale strumento di analisi è stato utilizzato per diversi ambiti abbastanza scontati perché non c'è dubbio che sia uno strumento di analisi dei segnali. La questione sorprendente è stato l'aspetto pratico in altri ambiti non scontati: la programmazione e il digital filtering. Grazie agli approcci classici, ormai pattern di riduzione della complessità computazionale, quali, per esempio, il Divide Et Impera.

4 Esperimenti

La parte più importante del lavoro di Tesi è proprio questa sezione. Vengono affrontati tutti i problemi generali che richiedono l'uso della DFT o dove la stessa è la soluzione migliore. Verranno comunque, di volta in volta, ripresi molto brevemente i concetti di base per giustificare determinate scelte e osservazioni.

4.1 DFT: approssimazioni

4.1.1 Approssimazione di soluzioni di ode

Diversamente dalla trasformata continua di Fourier, pertanto, la DFT richiede in ingresso una funzione discreta i cui valori sono in generale complessi e non nulli, ed hanno una durata limitata. Questo rende la DFT ideale per l'elaborazione di informazioni su un elaboratore elettronico. In particolare la trasformata discreta di Fourier è ampiamente utilizzata nel campo dell'elaborazione numerica dei segnali e nei campi correlati per analizzare le frequenze contenute in un segnale, per risolvere equazioni differenziali alle derivate parziali e per compiere altre operazioni, come la convoluzione o la moltiplicazione di numeri interi molto grandi. Alla base di questi utilizzi c'è la possibilità di calcolare in modo efficiente la DFT usando gli algoritmi per trasformata di Fourier veloce.

Sulla base di quanto dimostrato nella sezione 2.8, tramite la DFT è possibile approssimare la soluzione delle equazioni differenziali come conseguenza del fatto che la stessa può approssimare la trasformata di Fourier e dunque la soluzione che si calcolerebbe con quest'ultima. E' bene ricordare che la risoluzione di un'equazione differenziale necessita, in linea generale, di condizioni iniziali per poter determinare una soluzione unica, nel caso la si voglia o serva. La soluzione trovata con la Trasformata è univocamente determinata anche se non è stata prefissata alcuna condizione iniziale. Tale risultato è una diretta conseguenza del fatto che la trasformata di Fourier seleziona la sola soluzione continua e limitata esistente sull'intera retta reale: dal punto di vista dello spazio di Lebesgue e della sua teoria, si parla di funzioni misurabili ed integrabili, ossia nelle condizioni in cui ci troviamo Misurabilità e Integrabilità, nel senso di Lebesgue, coincidono.

Di seguito un esempio di risoluzione di equazione differenziale del secondo ordine.

$$y''(t) + 6y'(t) + 5y(t) = f(t)$$

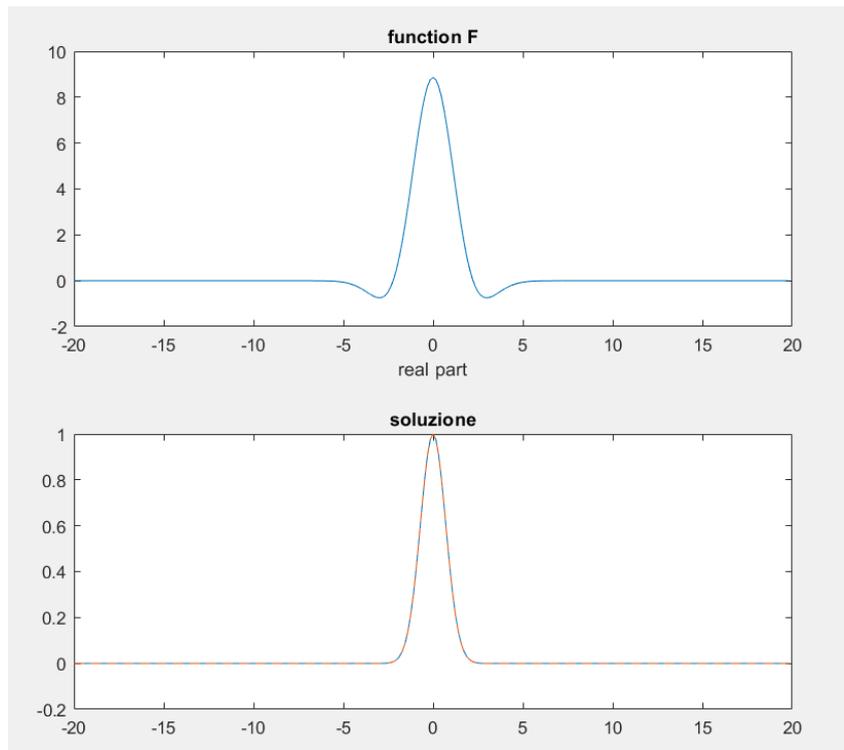
$$f(t) = e^{-t^2}(4t^2 - 12t + 3)$$

Procedendo con dei semplici calcoli analitici è possibile dare a Matlab la forma di integrale di Fourier da approssimare tramite ifft per poter risalire al segnale $y(t)$, la soluzione che, come già detto, sarà un unico segnale.

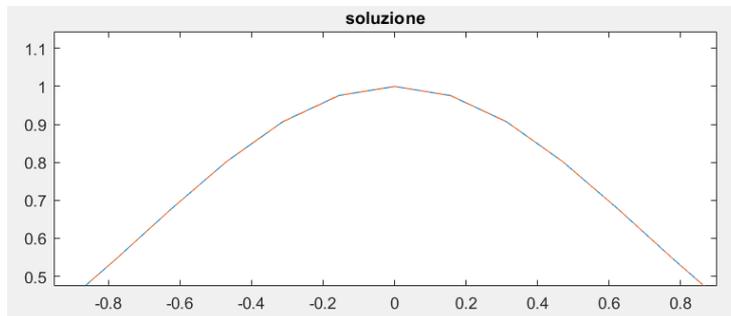
Infatti, anti-trasformando, il segnale che dovrà rendere Matlab sarà la "conseguenza" dell'approssimazione del seguente integrale:

$$y(t) = \int_{-\infty}^{+\infty} e^{j2\pi ut} \frac{F(u)}{-4u^2\pi^2 + j12\pi u + 5} du$$

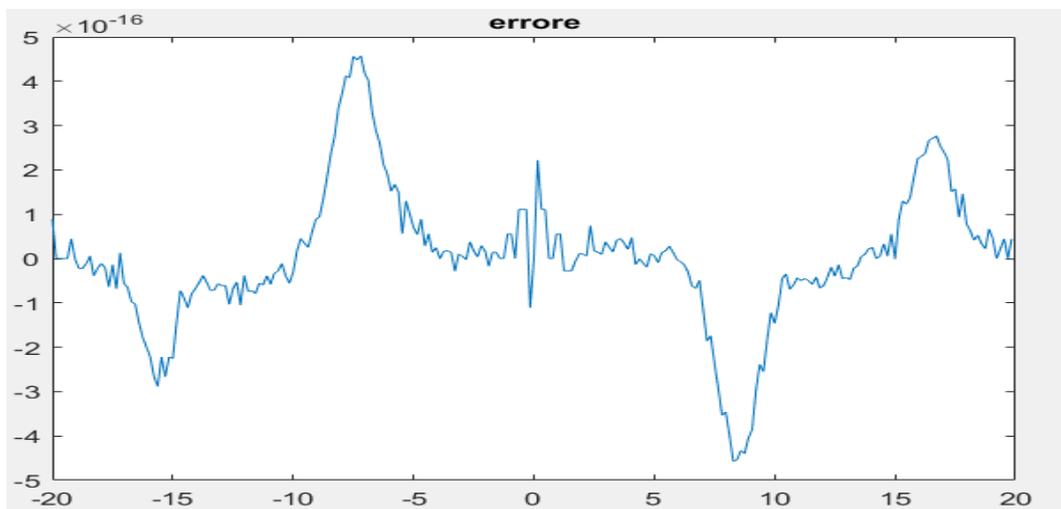
Ottenendo:



Nel secondo grafico vi è una sovrapposizione di grafici: soluzione analitica esatta e soluzione approssimata tramite ifft. Zoomando la campana, per esempio, si vede ancora il contrasto rosso-blu perché il calcolo è molto preciso:



Tra l'altro, lo stesso errore di approssimazione, si mantiene molto piccolo:



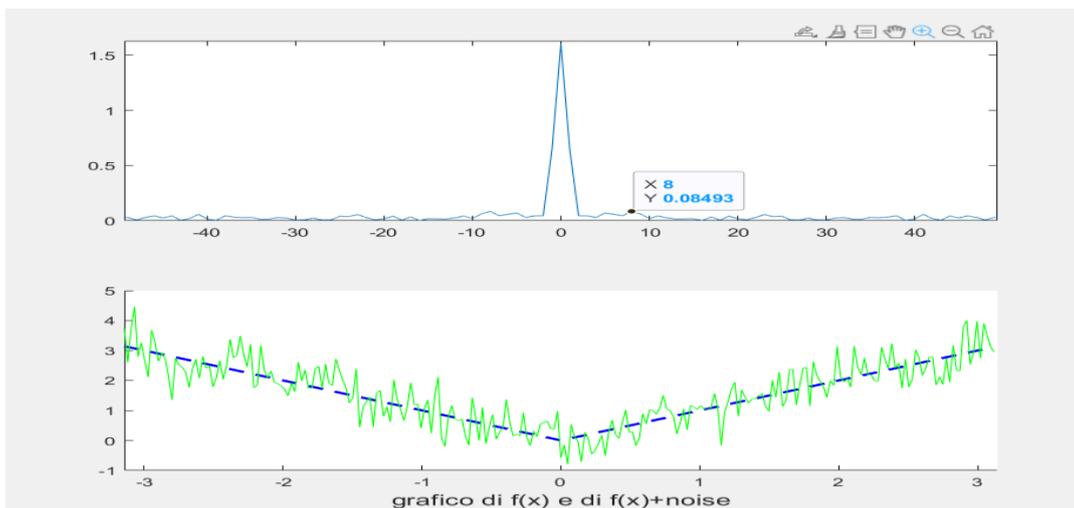
4.1.2 DFT: approssimazione di coefficienti di Fourier

Dal punto di vista del ruolo che la DFT gioca nell'ambito delle approssimazioni, è ancora interessante osservare come la trasformata discreta possa approssimare la serie di Fourier, o meglio, i coefficienti di Fourier. La questione è semplicemente più interessante anche da un punto di vista prettamente matematico: esistono diversi teoremi dell'analisi matematica che portano a risultati eleganti sulla convergenza della serie alla sua somma o funzione generatrice. E' dunque interessante andare a vedere come si comporta la DFT nel caso di approssimazioni interessanti e che mettono a dura prova le somme ridotte della serie di Fourier. Sicuramente gli esempi più interessanti sono sempre gli stessi, ma vengono proposti per osservare come la "doppia" approssimazione si comporta a seconda della situazione: discontinuità e irregolarità sono i fattori più insidiosi per la convergenza puntuale della serie, e dunque della trasformata discreta, o meglio, serie.

Un primo esempio sull'approssimazione dei coefficienti di Fourier, è proprio il segnale periodico a "dente di sega isoscele", il famoso "modulo di x".

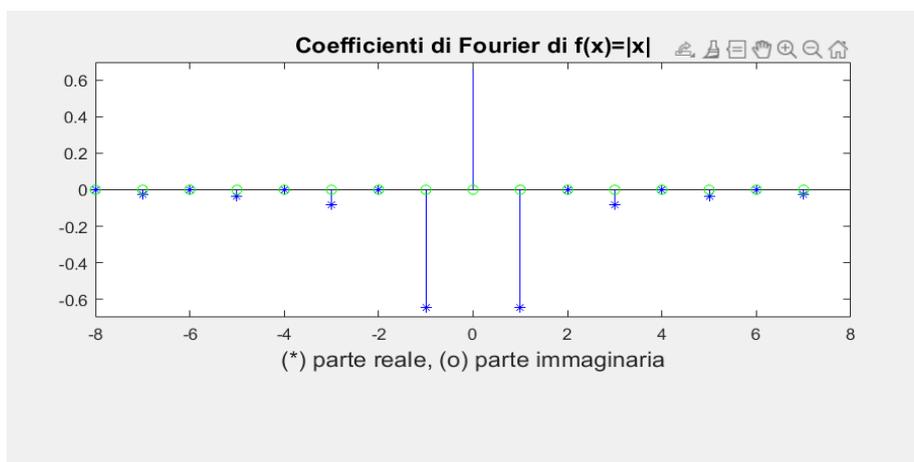
Viene sovrapposto del rumore ulteriore con la funzione Matlab randn.

Si mostrano lo spettro del modulo dei coefficienti di Fourier e il segnale nel dominio del tempo, entro l'intervallo $[-\pi, \pi]$, del segnale $x(t)=|t|+\text{rumore}$.



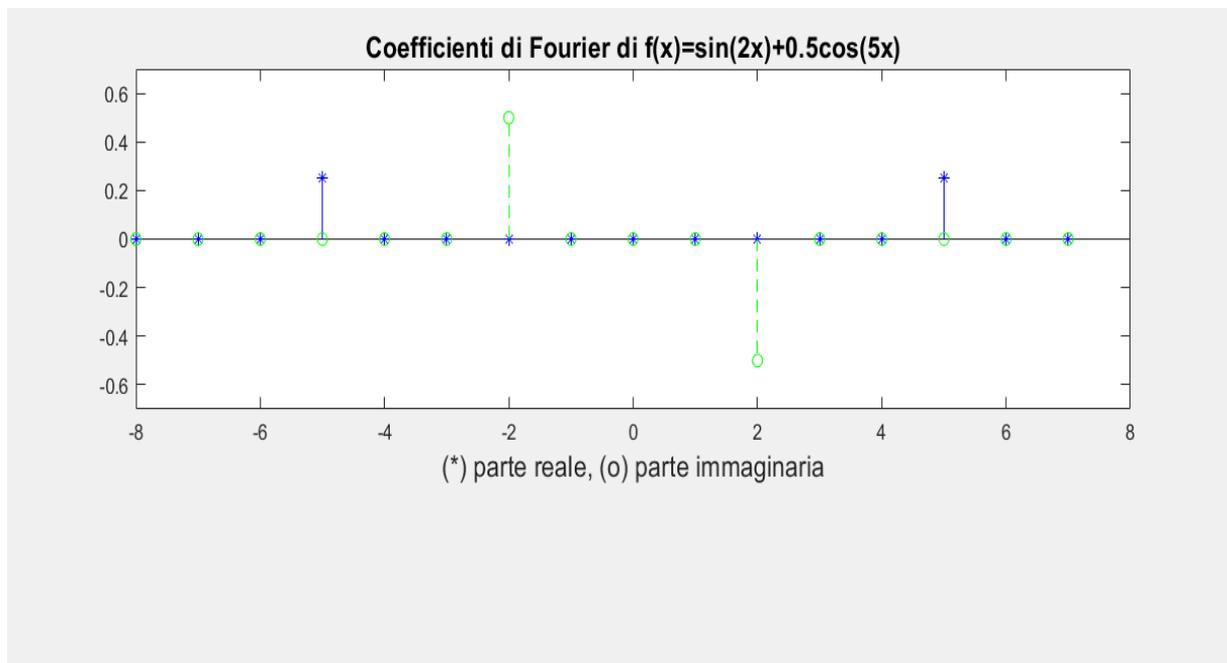
La presenza del rumore è evidente dallo spettro del modulo: la "forma d'onda" è stata alterata.

E' interessante visualizzare cosa succede ai coefficiente, ricordando che si parla di una funzione pari e che lo sviluppo in serie di Fourier originale, riceve alcune semplificazioni:



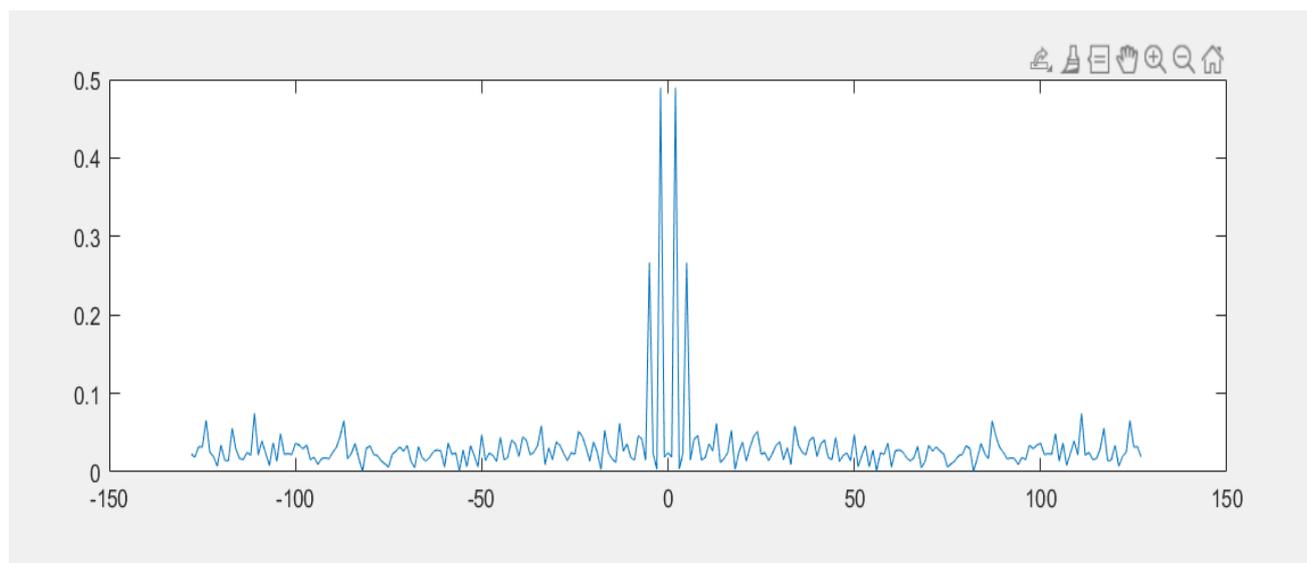
E' interessante andare ad osservare anche i casi "dispari" e "né pari né dispari".

Una funzione, per il secondo caso, viene scelta oscillante: prima si considera il segnale singolo e, successivamente, si altera il segnale con la sovrapposizione di rumore.



La presenza della somma delle due funzioni "duali", dal punto di vista della simmetria nel dominio del tempo, stronca lo spettro in coefficienti di postazioni pari e dispari.

Sovrapponendo il rumore, si ottiene:

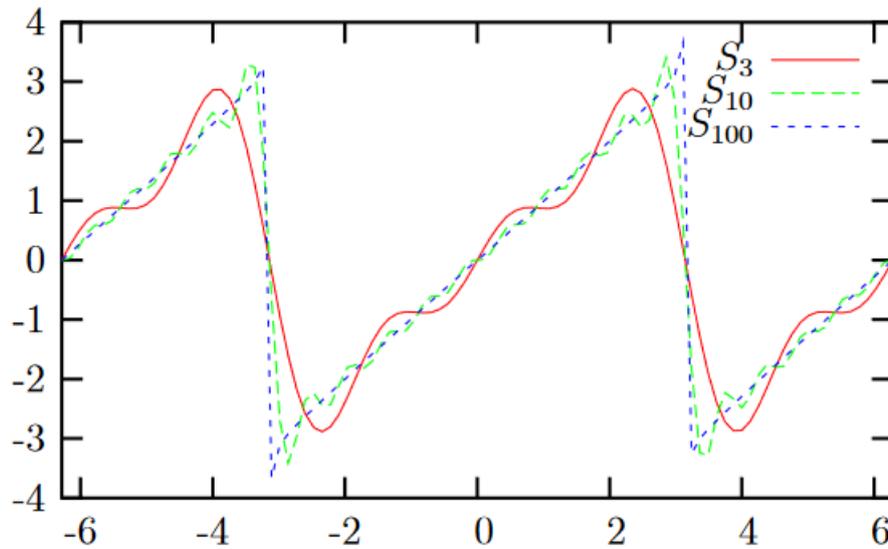


Come ultimo esempio è interessante notare come si comporta la convergenza della DFT, dal punto di vista dell'approssimazione, nel caso in cui le somme ridotte si allontanino agli estremi dell'intervallo. Infatti, se si considera il segnale dispari $y(t)=x$, si nota che non sussiste l'uguaglianza:

$$y(-\pi) = y(\pi) = P(\pm\pi)$$

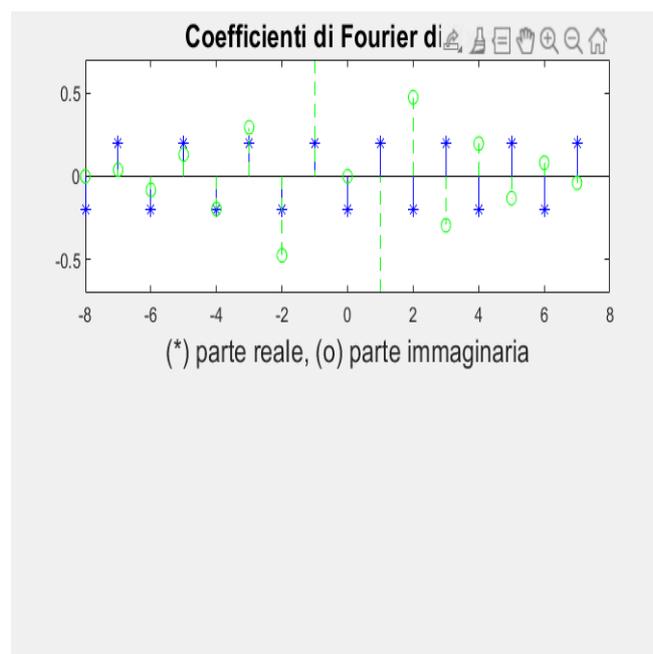
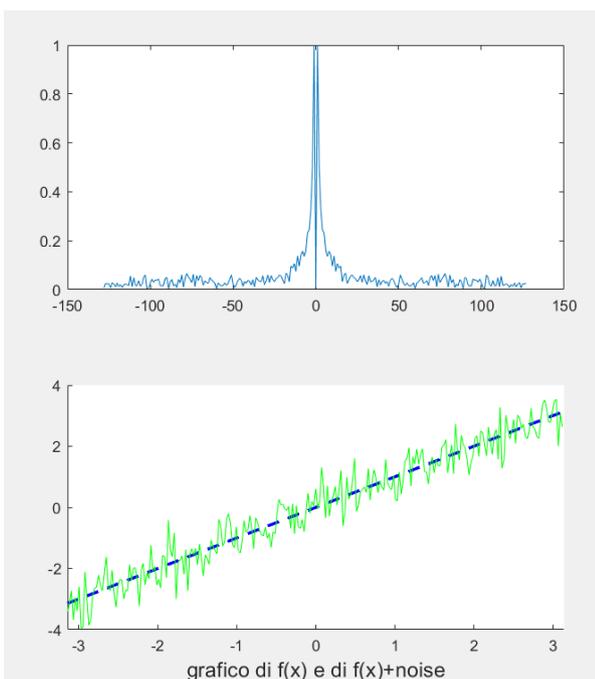
con $P(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{L} + b_n \sin \frac{n\pi t}{L} \right)$; con $L = \pi$ e $T = 2\pi$.

Si può notare considerando la sovrapposizione della funzione "bisettrice" con il polinomio trigonometrico:



La ridotta ennesima, in generale, non convergerà mai all'ordinata $y = \pm\pi$: ma anzi, all'aumentare dell'indice di somma, il valore della somma ridotta si allontana sempre più.

Si considera l'approssimazione mediante DFT e la sovrapposizione di rumore.



4.1.3 Interpolazione di data-set

Un'altra applicazione della DFT è l'interpolazione trigonometrica di data-set periodici. Sarà estremamente utile generalizzare il problema che verrà proposto in seguito per un importante risultato: ottenere un'anteprima del DFT. Ci sono alcuni vantaggi nel farlo e il motivo è proprio ottenere le condizioni interpolanti in termini di esponenziale complesso. Si considera un caso con $N = 4$ come lunghezza del data-set in input e la forma reale delle condizioni interpolanti.

Si consideri il periodo fondamentale di ripetizione della sequenza numerica o vettore di punti:

$$P = [(0,2) \quad (0.5\pi, 1) \quad (\pi, 3) \quad (1.5\pi,2)].$$

Si consideri il polinomio trigonometrico:

$$y(t) = a_0 + a_1 \cos(t) + b_1 \sin(t) + a_2 \cos(2t) = y(t + nT)$$

dove l'ultima precisazione serve a puntualizzare che il polinomio o funzione $y(t)$ è 2-pigreca-periodica.

La condizione di interpolazione è: $y(t_n) = y_n$, dove il primo membro indica il valore assunto dal polinomio nel punto t_n e il secondo membro indica l'ordinata di ciascuno dei punti del vettore P o data-set in input. La condizione è un vincolo, in pratica e questo perché il polinomio oscillante deve passare, in corrispondenza dell'ascissa t_n sull'ordinata assunta dalla funzione che è, appunto, rappresentata da un vettore di punti.

Un altro polinomio trigonometrico interpolante è il seguente:

$$y_n = \sum_{k=0}^3 C_k e^{\frac{j2\pi nk}{N}} \text{ ove } N=4 \text{ e } n=0,1,2,3. \text{ Per chiarire meglio, si sviluppano le equazioni del sistema lineare:}$$

$$\text{equazione 1: } n=0 \quad y_0 = \sum_{k=0}^3 C_k e^{\frac{j2\pi 0k}{4}}$$

$$\text{equazione 2: } n=1 \quad y_1 = \sum_{k=0}^3 C_k e^{\frac{j2\pi 1k}{4}}$$

$$\text{equazione 3: } n=2 \quad y_2 = \sum_{k=0}^3 C_k e^{\frac{j2\pi 2k}{4}}$$

$$\text{equazione 4: } n=3 \quad y_3 = \sum_{k=0}^3 C_k e^{\frac{j2\pi 3k}{4}}$$

Il motivo di questa scelta sarà chiaro più avanti. Al variare di k , si ottengono 4 termini per equazione e sostituendo ai valori di y_n . Il sistema lineare $b=Ac$:

$$\begin{bmatrix} 2 \\ 1 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & 1 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

- $A \in \mathbb{C}^{4 \times 4}$
- $c \in \mathbb{C}^{4 \times 1}$
- $b \in \mathbb{R}^{4 \times 1}$

La cui soluzione è il vettore di complessi:

$$c = [C_0 \ C_1 \ C_2 \ C_3] = [(43+i)/20 \quad (-3+4i)/10 \quad (7-i)/20 \quad (-1-2i)/5].$$

La motivazione di tale conto è la seguente: il polinomio trigonometrico iniziale è stato, in realtà utilizzato. Più precisamente, la soluzione trovata può essere adattata a quella cercata con una discretizzazione dell'asse dei tempi. L'ascissa temporale ennesima è pensabile come $t_n = \frac{2\pi n}{N}$, ossia il periodo del polinomio è direttamente proporzionale al periodo di ripetizione del data-set.

Come già detto, l'analisi spettrale di sequenze numeriche periodiche è influenzata da un fatto molto importante: il campionamento di un segnale periodico a tempo continuo non genera necessariamente una sequenza periodica. Affinché si abbia una sequenza periodica, è necessario che un numero intero N di intervalli di campionamento sia esattamente pari a un qualche numero intero n di periodi di ripetizione del segnale originario: $NT=n2\pi$. Ciò significa che il rapporto tra i due periodi deve essere razionale.

Sotto tali ipotesi, allora il campionamento è coerente e le reciproche relazioni, su cui si basa la DFT, consentono di instaurare un legame tra le due versioni del polinomio trigonometrico. In pratica i due polinomi, sotto tali ipotesi di campionamento, sono la stessa cosa.

Tra l'altro, avendo a che fare con un data set di numeri reali, si ottengono le seguenti relazioni tra i coefficienti del polinomio trigonometrico reale e i coefficienti del polinomio trigonometrico complesso:

$$a_k = 2\text{Re}\{c_k\} ; \quad b_k = -2\text{Im}\{c_k\} ; \quad a_0 = c_0 ; \quad .$$

Tali relazioni sono proprio quelle tra i coefficienti di Fourier delle varie versioni dello stesso sviluppo, a seconda del segnale del tipo "odd" oppure "even".

Infatti:

$$x_{odd}(t) = 2j \sum_{n=1}^{\infty} (b_n \sin(n\omega t)) ; \quad x_{even}(t) = 2a_0 + 2 \sum_{n=1}^{\infty} (a_n \cos(n\omega t)) ;$$

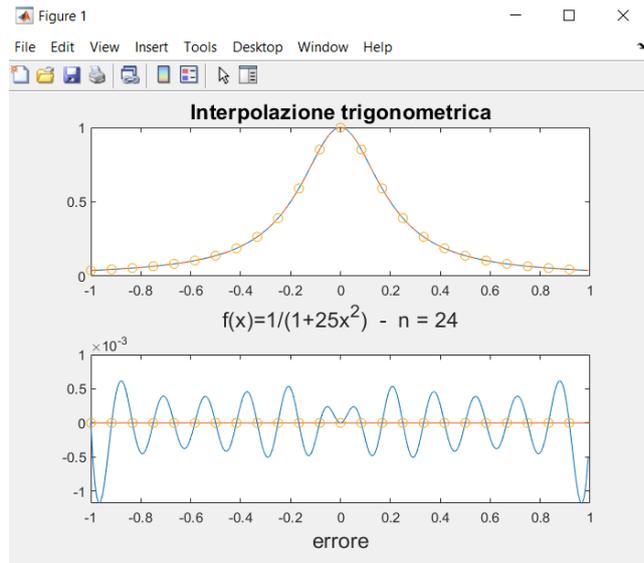
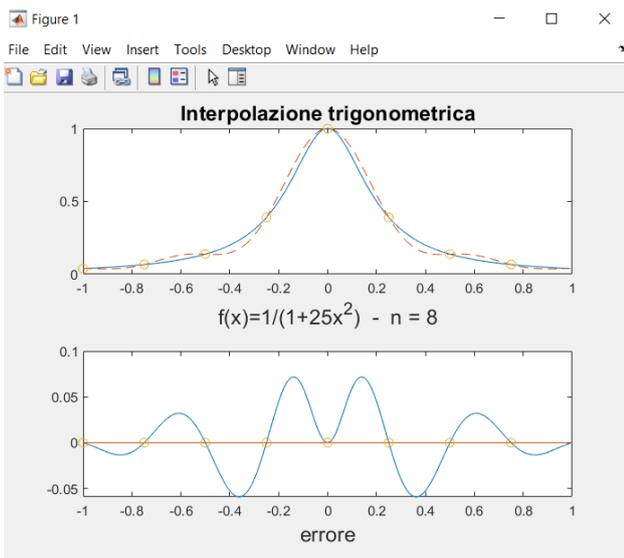
$$\text{Mentre lo sviluppo generale è proprio: } x_0(t) = \sum_{k=-\infty}^{\infty} C_k e^{j2\pi f_k t}.$$

4.1.4 Approssimazione Trigonometrica

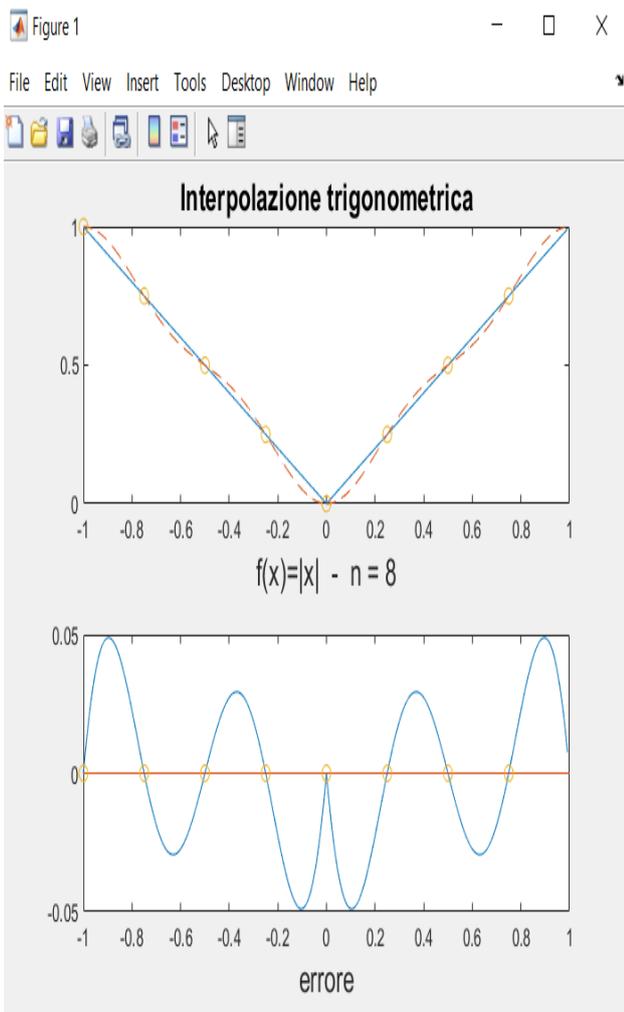
Dato un problema di interpolazione di data-set periodico, è stato naturale poter definire la DFT, in pochi passaggi: questo è il terzo modo in cui viene ottenuto lo strumento DFT! L'esigenza del dover interpolare un set di dati o, meglio, l'aver impostato un'interpolazione ha conseguito la definizione naturale di trasformata discreta. Essa è usata per esigenze che richiedono passaggi risolvibili con le caratteristiche naturali della stessa: un polinomio trigonometrico è sempre associabile ad un problema di interpolazione e la DFT è definita come polinomio trigonometrico.

Di seguito sono mostrati alcuni esempi di approssimazione trigonometrica di funzioni, scelte appositamente soddisfacenti le condizioni di convergenza della serie di Fourier.

Si noti il contenimento dell'errore oscillante in range sempre più piccoli al crescere del numero di punti di valutazione della funzione approssimante.



Nuovamente si consideri la funzione modulo:



Curiosità.

Si noti come l'errore sia minore con soli 8 punti nel caso di una funzione regolare come $|x|$ piuttosto che con la funzione frazionaria dell'esempio sopra. La DFT nasce proprio con l'idea di calcolare ciò che la Serie e la Trasformata andrebbero a ricercare in un numero infinito di termini della somma (nel caso della serie) ed estremi di integrazione illimitati e valori illimitati di una funzione continua della frequenza (nel caso della trasformata). Il passaggio al mondo digitale consente infatti di effettuare in modo automatico diverse operazioni di misura e in molti casi di svolgere elaborazioni più complesse, come l'analisi in frequenza con la Trasformata Rapida di Fourier (FFT).

4.2 DFT: applicazione nel campionamento

4.2.1 Aliasing

Si consideri il seguente sistema rappresentato tramite notazione della Teoria dei Sistemi:

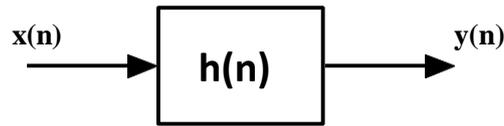


Figura 30: funzione di trasferimento

$$x[n] = [1 \ 2 \ 2 \ 1]$$

$$h[n] = [1 \ 2 \ 3]$$

Si vuole calcolare la convoluzione lineare di $x(n)$ e $h(n)$ usando il metodo, descritto nella sezione 3.2, basato sull'uso della DFT ed effettuando un'analisi dal punto di vista del campionamento, o se si vuole, sul numero di campioni da considerare e cosa succede al variare di tale numero. In primo luogo, essendo $x(n)$ di lunghezza $L=4$ e $h(n)$ di lunghezza $M=3$, otterremo una sequenza $y(n)$ di lunghezza $N=L+M-1=6$; per cui dovremo calcolare la DFT di $x(n)$ e la DFT di $h(n)$ basate entrambe su almeno 6 campioni. Scegliamo allora di calcolare le due DFT con 8 campioni: la scelta di 8 campioni ha un motivo ben preciso e cioè quello per cui gli algoritmi di calcolo della DFT sono ottimizzati quando il numero di campioni da calcolare è una potenza di 2. Il più semplice algoritmo di FFT si applica quando l'ordine N della trasformata è proprio una potenza di 2, cioè $N = 2^m$.

Dovendo calcolare due DFT basate su 8 campioni, si deve allungare le due sequenze $x(n)$ e $h(n)$, tramite campioni nulli posti in coda, al fine di ottenere una lunghezza 8 elementi per entrambe le sequenze con la tecnica di zero-padding:

$$x[n] = [1 \ 2 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0]$$

$$h[n] = [1 \ 2 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$X(k) = 1 + 2e^{-j\pi k/4} + 2e^{-j\pi k/2} + e^{-j3\pi k/4} ; k=0,1,\dots,7 ;$$

Il vettore $X(k)$ è:

6	$\frac{2+\sqrt{2}}{2} - j\frac{4+3\sqrt{2}}{2}$	$-1-j$	$\frac{2-\sqrt{2}}{2} + j\frac{4+3\sqrt{2}}{2}$	0	$\frac{2-\sqrt{2}}{2} - j\frac{4-3\sqrt{2}}{2}$	$-1+j$	$\frac{2+\sqrt{2}}{2} + j\frac{4+3\sqrt{2}}{2}$
---	---	--------	---	---	---	--------	---

In modo del tutto analogo dobbiamo procedere per $H(k)$: facendo i conti, si trova che tale vettore è dato da

$$\boxed{6 \quad 1 + \sqrt{2} - j(3 + \sqrt{2}) \quad -2 - j2 \quad 1 - \sqrt{2} + j(3 - \sqrt{2}) \quad 2 \quad 1 - \sqrt{2} - j(3 - \sqrt{2}) \quad -2 + j2 \quad 1 + \sqrt{2} + j(3 + \sqrt{2})}$$

Il vettore $Y(k)$:

$$\longrightarrow \boxed{36 \quad -14.07 - j17.48 \quad j4 \quad 0.07 + j0.515 \quad 0 \quad 0.07 - j0.515 \quad -j4 \quad -14.07 + j17.48}$$

Antitrasformando tramite IDFT, si ottiene: $y[n] = [1 \ 4 \ 9 \ 11 \ 8 \ 3 \ 0 \ 0]$.

Vi sono alcune importanti osservazioni da fare. Gli ultimi due campioni, $y(6)$ e $y(7)$, sono nulli perché la sequenza risultante era previsto avesse lunghezza 6, essendo 4 e 3 le lunghezze dei due vettori in convoluzione ed essendo la DFT una trasformazione con “memoria” sulla lunghezza. Ci si aspetta, allora, che se la DFT venisse calcolata su 6 campioni, piuttosto che 8, allora si otterrebbe lo stesso vettore $y[n]$, ma ripulito dei due zeri in coda.

Allora, come naturalmente deve essere, ci si chiede cosa possa succedere nel caso in cui non si rispetti la disuguaglianza sulla dimensione del vettore $y[n]$ che di fatto è una condizione di costruzione della DFT corretta: ossia ci fornisce il range entro cui devono variare n e k del kernel della trasformata. Nella sezione 3 è stato visto che questo equivale a rispettare il teorema del campionamento.

Si sceglie di campionare solo 4 elementi in frequenza e dunque si può lasciare le due sequenze al loro stato naturale e procedere con la DFT delle singole, il loro prodotto, far variare l'indice k per la $Y(k)$ e applicare la IDFT ed osservare che cosa si è ottenuto come output.

$$X(k) = [6 \quad -1-j \quad 0 \quad -1+j] \quad H(k) = [6 \quad -2-j2 \quad 2 \quad -2+j2]$$

Sono dunque i campioni nelle posizioni pari dei vettori prima calcolati: questa ragione è dovuta al fatto che l'equazione che governa le reciproche relazioni è $T_w = NT_s$, ossia il periodo di campionamento ed osservazione (windowing) sono rimasti inalterati e ciò che è stato dimezzato è il numero di campioni prelevati nel tempo, ergo in frequenza.

Come diretta conseguenza, la IDFT conserva tutte queste variazioni anche per la sequenza $y[n]$, essendo:

$$Y(k) = [36 \quad j4 \quad 0 \quad -j4] \quad \square\square y[n] = [9 \quad 7 \quad 9 \quad 11]$$

Vi è un inevitabile colpo di scena: $Y(k)$ assume le caratteristiche motivate da quanto detto e finora fatto, ma è completamente diversa.

La motivazione di tale diversità è l'**aliasing**, ossia il non aver rispettato la condizione di Nyquist sulla frequenza di campionamento, in poche parole. Nel caso della trasformata discreta si parla di costruzione dell'intervallo di campionamento basata sugli indici n e k .

Aver scelto di calcolare la DFT con 4 campioni ha spinto la IDFT a restituire la sequenza y con periodo di ripetizione $N=4$. Essendo la sequenza di natura “lunghezza 6”, allora vi è un'inevitabile sovrapposizione delle repliche dello spettro del segnale discreto. Quando la ripetizione arriva al penultimo ed ultimo termine, ecco che le prime due componenti dello spettro in banda base (ossia lo spettro che ogni volta viene replicato dal campionamento come dimostrato nella sezione 2.7) vanno a sommarsi con queste due e producono componenti incoerenti con l'operazione di partenza.

E' facile valutare questa sovrapposizione considerando l'equazione base di periodizzazione, da cui, come già detto, si ottiene il Teorema del campionamento tramite le due formule di Poisson, e vedere che prendendo i valori agli estremi di ripetizione, si ha sempre la sovrapposizione di due componenti armoniche con altre due.

$$y_P = \sum_{m=-\infty}^{+\infty} y[n - mN]$$

k=-2 : y(n+8) è un esempio non significativo per quanto detto

k=-1 : y(n+4) è un esempio significativo

k=1 : y(n-4) è un esempio significativo

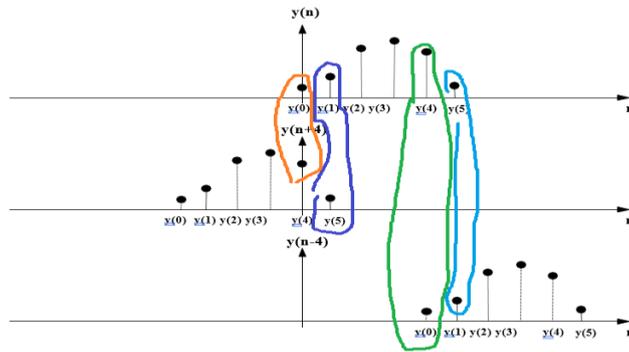


Figura 31: aliasing

Si vede come y(n-4) e y(n+4) si sovrappongono, rispettivamente, ai primi e ultimi due samples.

Si conclude che, dal punto di vista del campionamento, si ha: se x(n) ha lunghezza N e h(n) ha lunghezza M, la loro convoluzione lineare produce una sequenza lunga Y=N+M-1, la quale è univocamente rappresentabile, nel dominio discreto delle frequenze, da una DFT calcolata almeno su Y campioni; se invece calcoliamo la DFT (e poi la IDFT) su Y-M campioni, l'uscita sarà una sequenza di lunghezza Y-M con aliasing sui primi M campioni.

4.2.2 Uso della DFT: decimazione dei campioni

Tramite la dimostrazione sull'equazione rappresentante l'interpolatore lineare, avvenuta nella sezione 2 sull'interpolatore, è allora possibile poter progettare con Matlab un interpolatore a forma di triangolo e mettere in evidenza l'effetto del ricampionamento di un fattore non intero e dell'interpolazione su dati non sufficientemente sovracampionati.

Lo schema logico di tale esperimento sarà:

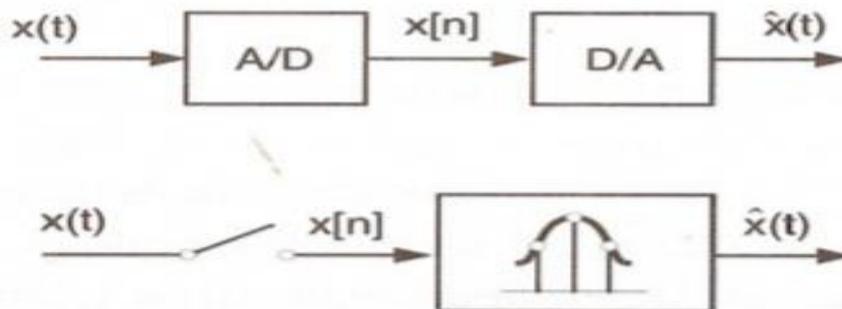
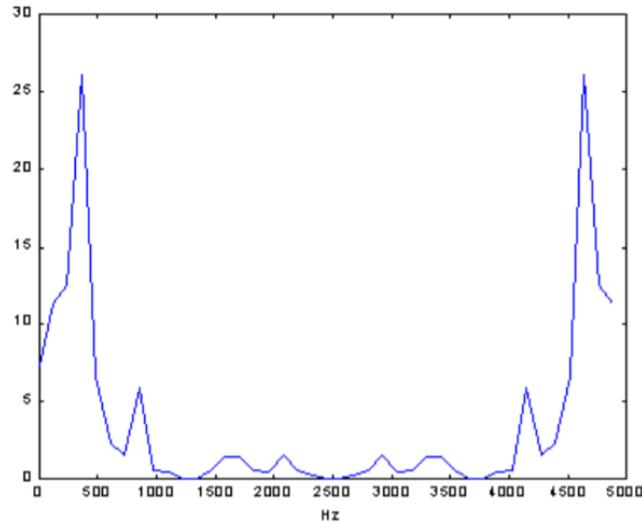


Figura 32: schema digitale

Si consideri il segnale:

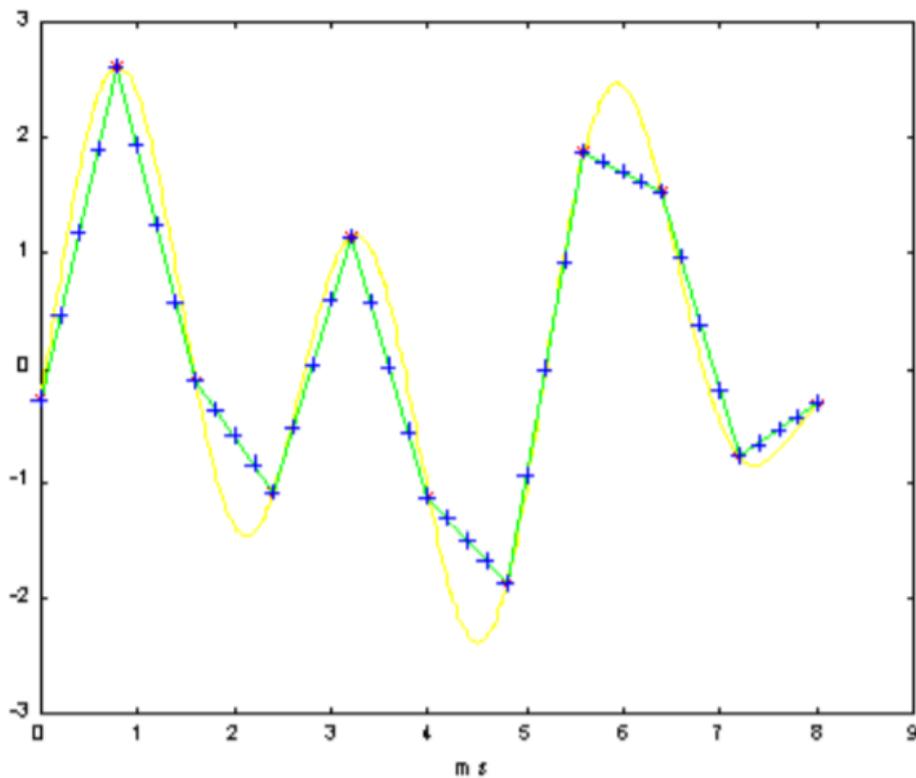
$$x(t) = \sin(2\pi 200t) + \sin\left(2\pi 350t + \frac{\pi}{7}\right) + \sin\left(2\pi 420t - \frac{\pi}{4}\right)$$

La funzione rappresentata è la somma di tre sinusoidi, per cui la sua trasformata di Fourier dovrebbe avere 3 impulsi localizzati alle frequenze di 200 Hz, 350 Hz e 420 Hz.



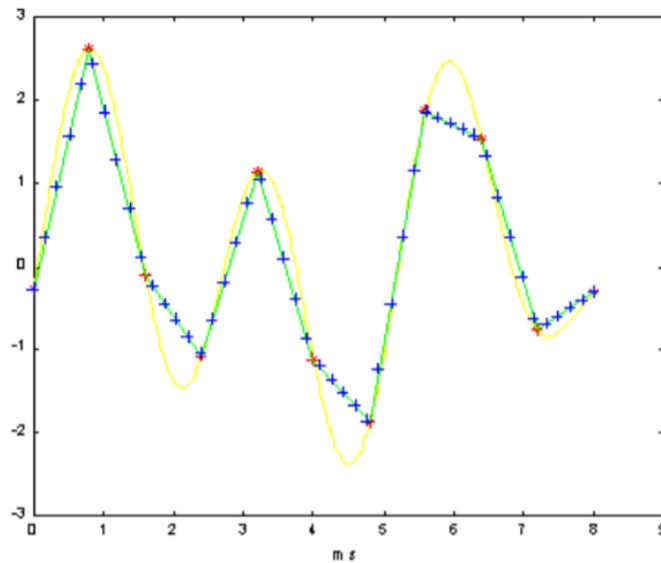
Come si può notare da quest'ultimo grafico, invece è possibile individuare soltanto l'impulso a 420 Hz perchè il passo di campionamento e di sovracampionamento non sono molto alti per poter trascurare gli effetti di bordo che si hanno durante il calcolo della DFT (infatti ci sono anche impulsi di ampiezza inferiore localizzati a frequenze intermedie che sono indesiderati e ciò è dovuto alla distorsione introdotta dal filtro interpolante).

Fattore ricampionamento 4.



Per quanto riguarda il grafico successivo con questo tipo di ricampionamento si riesce a rispettare i dati : questo accade se introduciamo un fattore di ricampionamento intero. In caso contrario si avrà che il segnale ricampionato (punti in blu) si discosta dai valori effettivi nei punti in cui andiamo a campionare all'inizio (in rosso), come succede ora.

Fattore ricampionamento 4.7 .

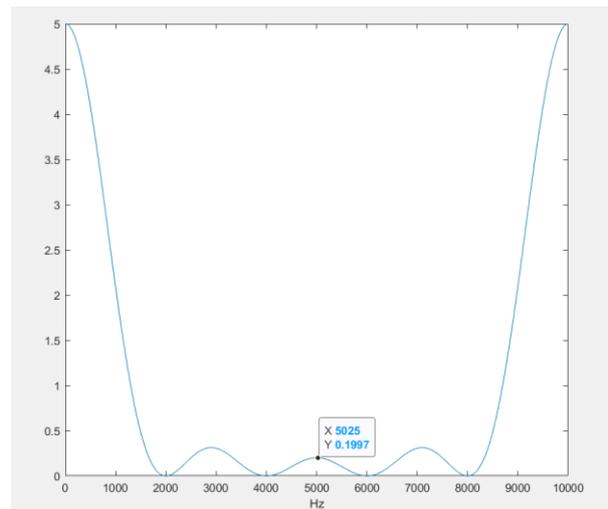
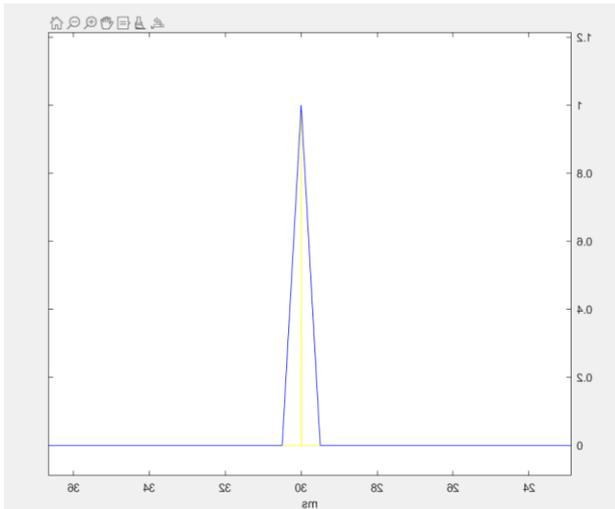


Si osservi che non è comunque cambiato tanto e ciò è conseguito dal fatto che il numero di campioni è sostanzialmente identico ed è sufficiente.

La cosa più interessante è che tutti questi fenomeni sono osservabili da delle equazioni che regolano la scelta della durata della finestra tramite il parametro T_w , la frequenza di campionamento f_c (il cui periodo di campionamento è l'inverso) e il periodo del segnale T_o (o i periodi delle componenti che lo costituiscono e caratterizzano il suo contenuto armonico) il cui inverso è la frequenza fondamentale (unica nel caso di una sinusoide) della componente generica del segnale.

È interessante osservare che il campionamento di un segnale periodico a tempo continuo non genera necessariamente una sequenza numerica periodica. Affinché si abbia una sequenza periodica è necessario che un numero intero N di intervalli di campionamento sia esattamente pari a un qualche numero intero m di periodi di ripetizione del segnale originario: $N1/f_c = m T_o$. In pratica, gli impulsi di campionamento del convertitore A/D devono essere sincronizzati con il segnale periodico analogico: essi non possono avere una cadenza arbitraria senza un legame preciso con la cadenza di ripetizione fondamentale del segnale dato. Se il rapporto N/m non è un numero razionale, l'operazione di campionamento non dà origine a una sequenza $x[n]$ periodica e si parla di campionamento incoerente.

Meno interessante risulta essere l'interpolazione di un segnale impulsivo che vede l'entità originale del filtro interpolante, ossia due spezzate di pendenza uguale e di segno opposto che si incontrano nel vertice del segnale:



Si nota che il triangolo è isoscele e l'impulso è il suo "segmento medio".

4.3 DFT: digital filtering

Un esempio di filtraggio di segnali viene esposto con un codice scritto da un professore universitario [Dr. Deepak Kumar Rout](#), professore associato di [School of Electronics Engg. KIIT University](#).

Come già visto, l'operazione di convoluzione è lineare, ossia il filtraggio di segnali è lineare. Più avanti verrà affrontato il calcolo delle convoluzioni, in situazioni sia ideali che reali. Quanto alla linearità, è bene mostrare un esempio significativo.

In questo programma si dimostra l'operazione di filtraggio lineare utilizzando DFT. Prese due onde sinusoidali di frequenze diverse e le si tracciano.

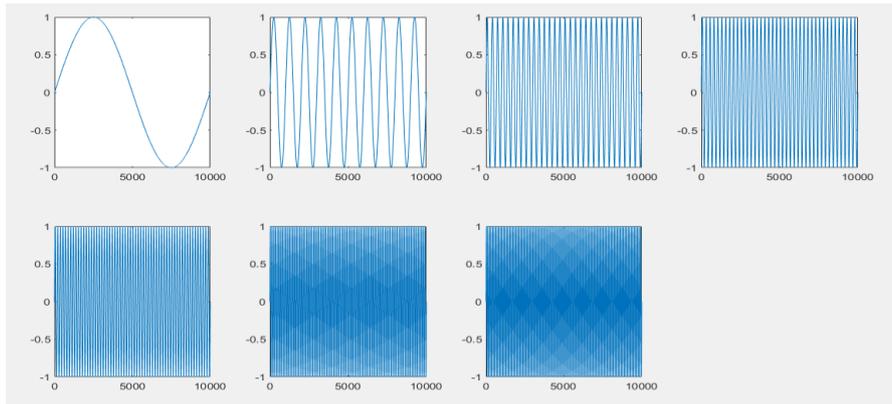
Quindi si mescolano tutte le onde sinusoidali per creare un segnale composito. Considerando (prendendo) la DFT del segnale composito, lo si traccia per mostrare che tutte le frequenze sono presenti. Quindi si moltiplica con un FIR la DFT del segnale: per filtrare il segnale stesso. Presa la IDFT e tracciato nuovamente il grafico, ora si plotta il segnale filtrato. La dimostrazione è grafica e afferma che effettivamente il segnale è stato filtrato. Nell'esempio è implementato un filtro passa basso.

Il mix di frequenze avviene tramite l'acquisizione di un input memorizzato in un vettore di frequenze e un ciclo for si occupa di plottare:

```
% the set of frequencies in the mixed signal are
f=[1 10 30 40 60 90 100]; % Hz

% generate a set of sine waves
for i=1:length(f)
x(i,:)=sin(2*pi*f(1,i)*t);
end
```

Il risultato dei plot di matlab, delle singole frequenze, ossia dei singoli seni di frequenza differente:

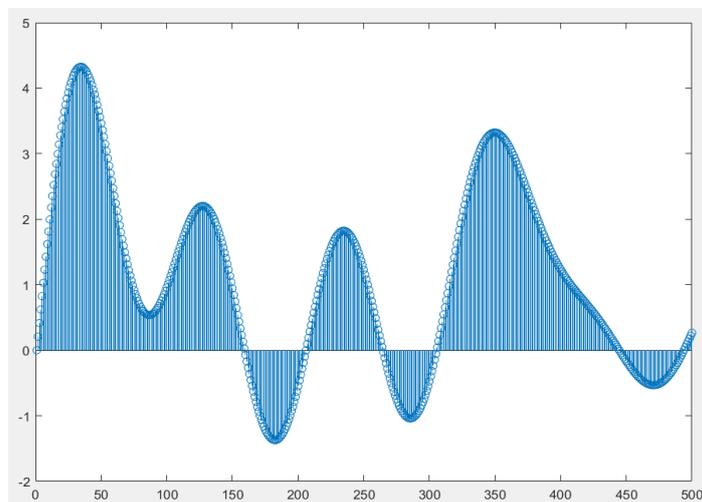


Anche da un punto di vista estetico, è interessante vedere la forma d'onda risultante una volta eseguito il mix tramite una semplice somma:

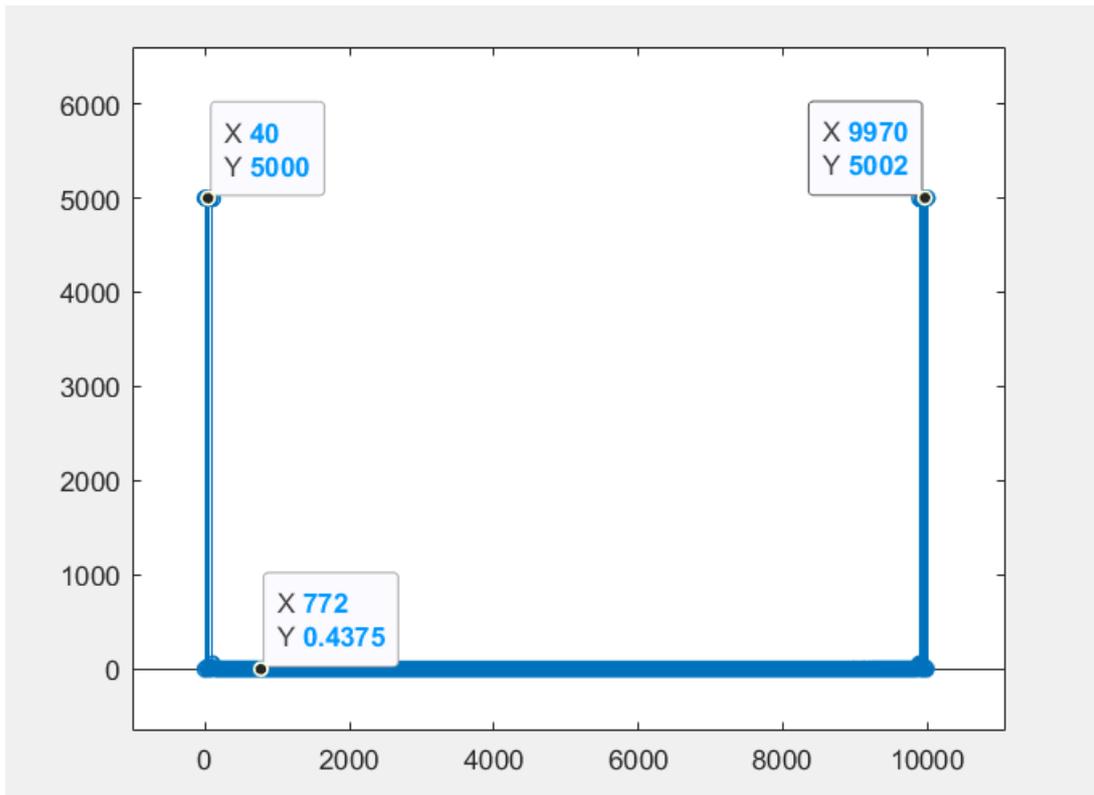
```

%% mix all the signals
x_s=sum(x);
figure
stem(x_s(1,1:500))
% %% let us see the frequency content of the signal
%
% [x_p,w]=periodogram(x_s);
% semilogy(x_p)

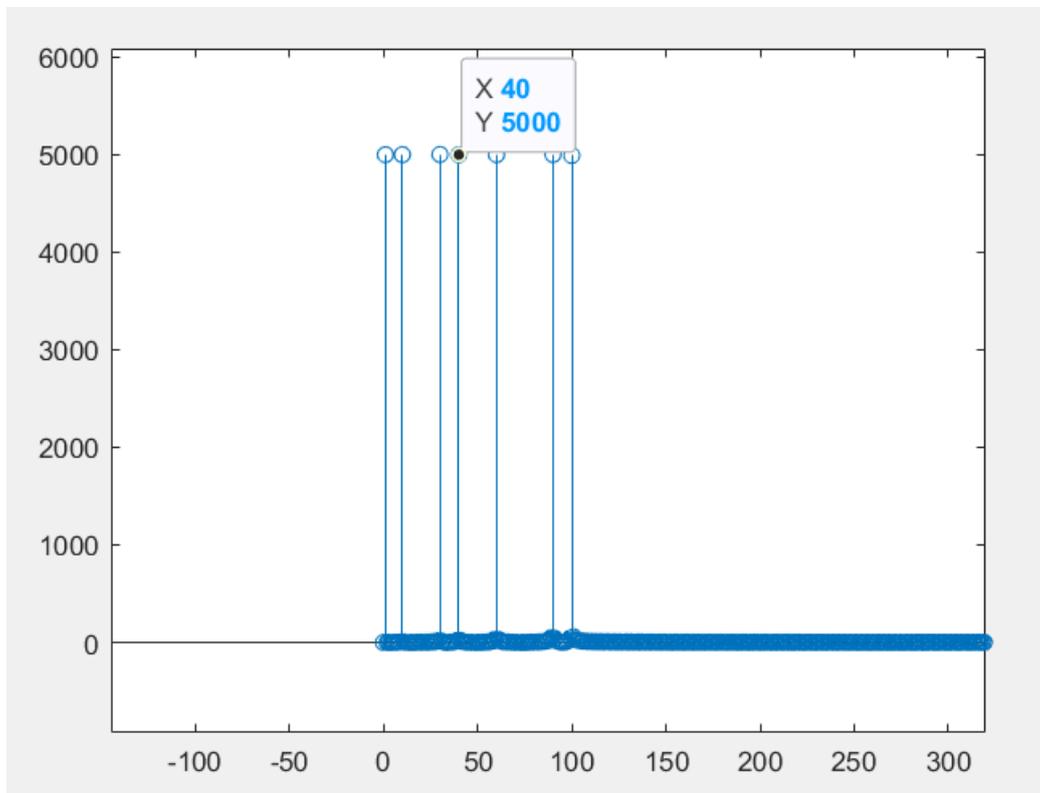
```



Dunque, dalla figura si nota che è già stato eseguito il campionamento: siamo già nel dominio del tempo discreto. Il campionamento di 1.000.000 sample nell'unità di tempo plottata (500s). Dunque calcolando il modulo dello spettro del segnale risultante tramite la solita funzione matlab, si ottiene:

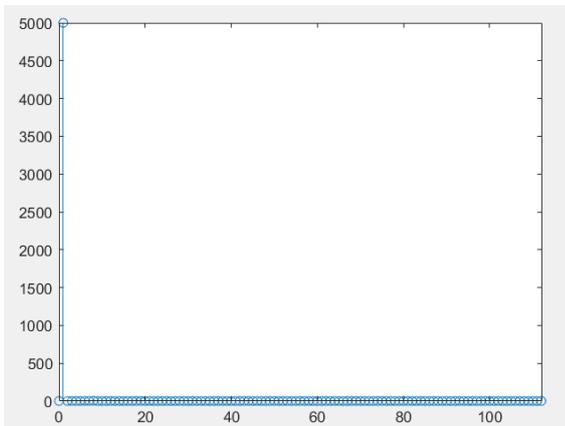


Zoomando, come ovviamente doveva essere:

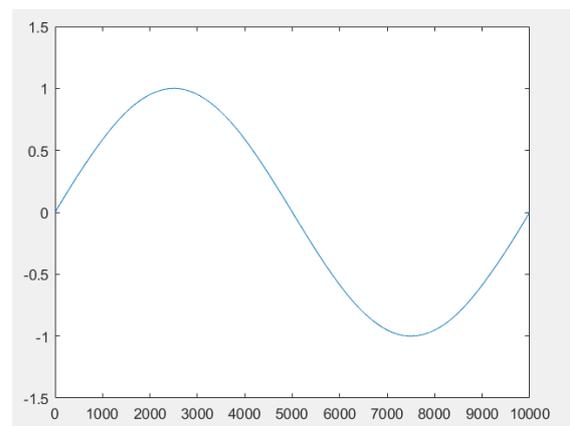


Procedendo con un FIR passabanda, filtriamo tutto, tranne che la sinusoide perfetta di 1Hz.

Dominio della frequenza dopo prodotto con FIR



Dominio del tempo dopo prodotto



→

Come si vedrà più avanti, la potenza della DFT è dettata da un'altra questione. Questa è l'applicazione per il filtraggio tramite tecniche di filtraggio note dalla Teoria dei Sistemi. Non sorprende affatto che si filtri in questa maniera, basti pensare alla parola "risposta impulsiva" e la trasformata di Laplace di quest'ultima ed è stato detto tutto. Tra l'altro, esiste un metodo di sintesi che è chiamato "sintesi in s ", dove s è la variabile di Laplace. In generale, tramite le specifiche ricevute come input di progetto, si individua la fascia di ammissibilità del piano complesso (o piano di Gauss), una volta che viene fatta, dall'ingegnere, la "traduzione" dei parametri di progetto in parametri formali. I poli del sistema possono essere fatti cambiare (e dunque variare lo spettro dei segnali in gioco) tramite il filtraggio, che nell'ambito delle trasformate, diventa il gioco di prodotto tra funzioni di trasferimento, nella variabile considerata. Quando si parla di segnali set-point e aperiodici, le trasformate di Fourier e di Laplace coincidono, a meno di restrizioni del dominio, dunque diventa facile capire che la DFT si possa usare anche per la sintesi in s .

Un esempio di rappresentazione di fascia di ammissibilità (dove si vuole che siano concentrate le frequenze libere, che nel caso del piano complesso, sono i poli):

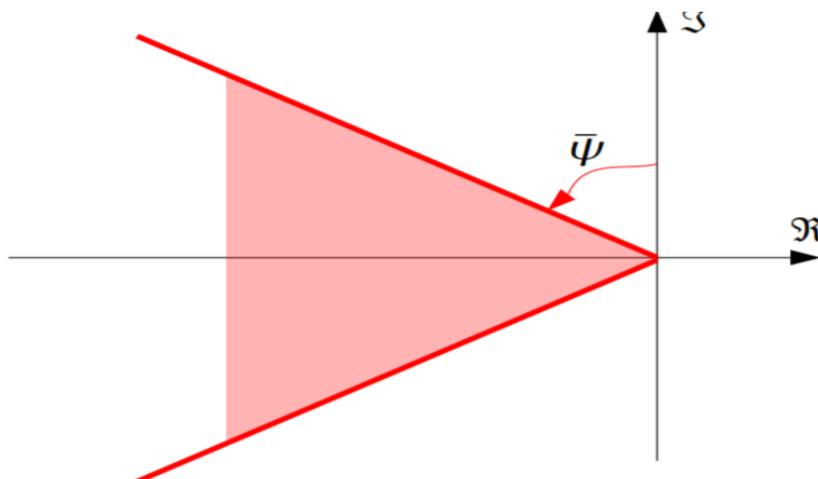


Figura 33: fascia di ammissibilità sul piano di Gauss per la taratura a ciclo chiuso

4.4 DFT: analisi spettrale

4.4.1 Analisi e zero padding

Una delle esigenze più frequenti nell'elaborazione dei segnali è quella di calcolare lo spettro di una sequenza data. Ovviamente, non si ha a disposizione un'espressione analitica dei valori della sequenza $x[n]$, ma solo i valori medesimi (ad esempio acquisiti mediante un convertitore A/D) in un intervallo finito, diciamo $1 \leq n \leq N$. La trasformata di una sequenza aperiodica è una funzione continua e dunque $X(f) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T}$ non potrà essere calcolata per gli infiniti valori della variabile f in un periodo, cioè, ad esempio, nell'intervallo $[0, 1/T]$. Ci si accontenterà di ottenere il valore della trasformata per un numero finito di punti normalmente equispaziati nell'intervallo suddetto. L'operazione può essere accorciata in maniera efficiente con l'equazione di Poisson per segnali a tempo-discreto (dimostrata nella sezione 2 per segnali a tempo-continuo e nella sezione 3 per segnali a tempo discreto) per tutti i valori dell'indice k che varia da 0 a $N-1$:

$$Y_k = \frac{1}{N} X\left(\frac{k}{NT}\right)$$

Ovviamente tale equazione nasce dall'idea di considerare una sequenza numerica y che è stata definita dalla periodicizzazione della sequenza originale aperiodica x . Dunque grazie al campionamento in frequenza ci si è svincolati dal problema "infiniti valori" e la sequenza è stata periodicizzata con periodo N . In definitiva è possibile calcolare i valori della funzione continua in N punti della frequenza equispaziati nell'intervallo $[0, 1/T]$ e questi sono:

$$f_k = \frac{k}{NT}$$

Vengono illustrati alcuni esempi.

Si consideri la sequenza aperiodica $x[n]$ di durata finita pari a $N=3$ e la sua versione periodica $y[n]$ di periodo la durata della sequenza originale.

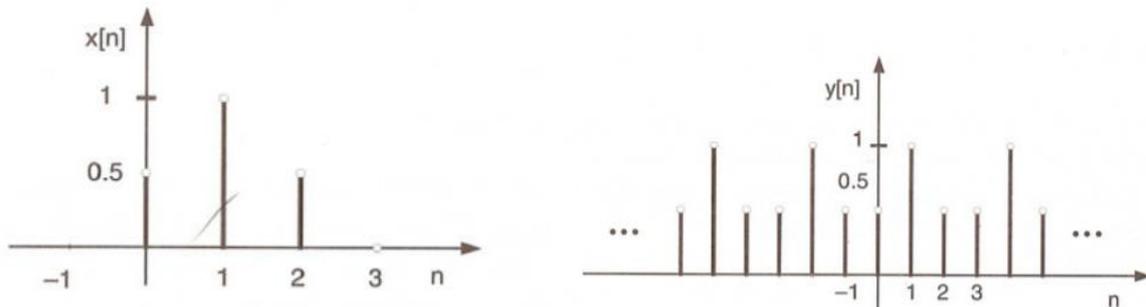


Figura 34: operazione di periodicizzazione

Per mettere in risalto il passaggio frequenza-continua in frequenza-discreta, dettato dall'equazione di Poisson, si effettua la trasformata analitica di $x[n]$.

$$X(f) = e^{-j2\pi f T} [1 + \cos(2\pi f T)]$$

e si ricavano i 3 "campioni" della funzione continua $X(f)$ per le frequenze $f_k = 0, 1/3T, 2/3T$ tramite l'equazione:

$$Y_k = \frac{1}{3} X\left(\frac{k}{3T}\right)$$

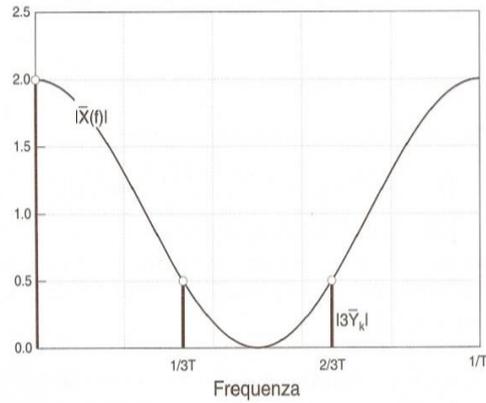


Figura 35: campionamento

Come suggerisce la sensazione “passaggio continuo-discreto”, il numero di campioni è troppo basso: per avere una stima fedele è evidente che più è alto il numero di campioni, e quindi la frequenza di campionamento, e più è fedele il grafico ottenuto per punti dello spettro continuo della sequenza aperiodica di partenza. Se si fa crescere il valore del parametro N_0 , ossia si aggiungono molti campioni nulli aumentando l'ordine della FFT, si può aumentare il numero di campioni della funzione $X(f)$, cioè si aumenta la risoluzione dell'analisi spettrale della sequenza $x[n]$ aperiodica e di durata finita originaria (che equivale ad aumentare la finestra di osservazione).

La domanda sorge spontanea: “se la sequenza di partenza ha solo 3 valori e la y è una periodicizzazione della medesima, allora come è possibile aumentare N ? Aumentando il periodo! Come però?”.

Una delle applicazioni della tecnica di zero-padding è proprio l'aumento della risoluzione in frequenza dell'analisi spettrale!

Re-iterando l'equazione che restituisce la tupla (x,y) dello spettro discreto per $N=8$, è già evidente l'aumento di campioni e della densità spettrale recuperata e della significatività acquisita:

la nuova sequenza periodica

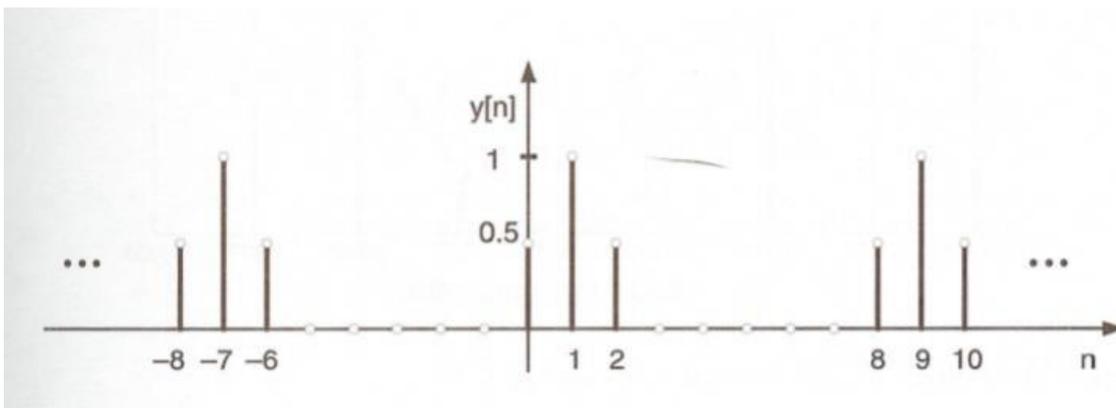


Figura 36: zero padding

la nuova risoluzione in frequenza

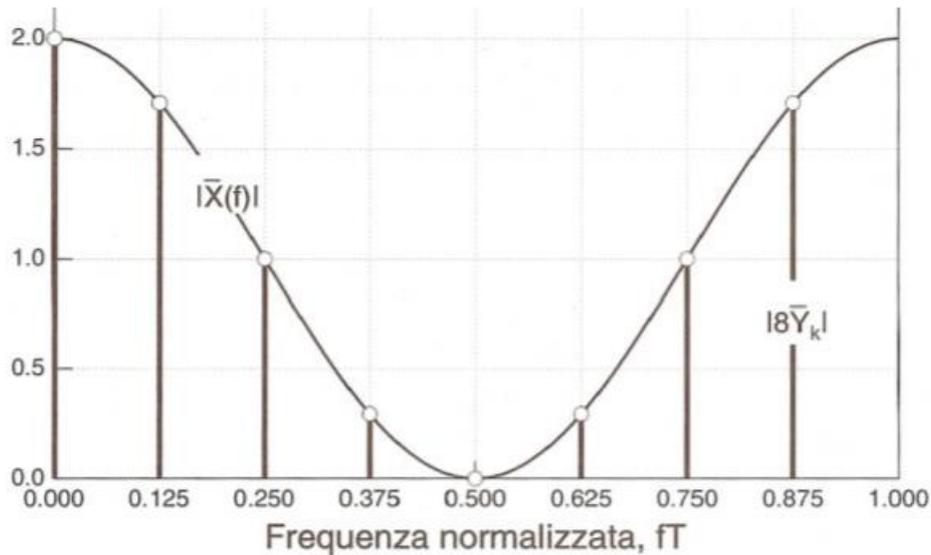


Figura 37: aumento della densità di campioni

L'andamento dello spettro di ampiezza $|X(f)|$ viene ora ricavato con una risoluzione maggiore che nel caso precedente. Vale la pena osservare che l'operazione di periodicizzazione concettualmente necessaria prima di calcolare la trasformata discreta non viene eseguita in pratica. Ciò che serve infatti per il calcolo della trasformata sono soltanto gli N campioni della sequenza $y[n]$ nel periodo base. Viceversa, l'operazione di zero-padding è fondamentale per costruire correttamente $y[n]$ a partire da $x[n]$, e deve essere effettivamente eseguita prima di calcolare la trasformata discreta: ossia allungare la lunghezza d'onda discreta.

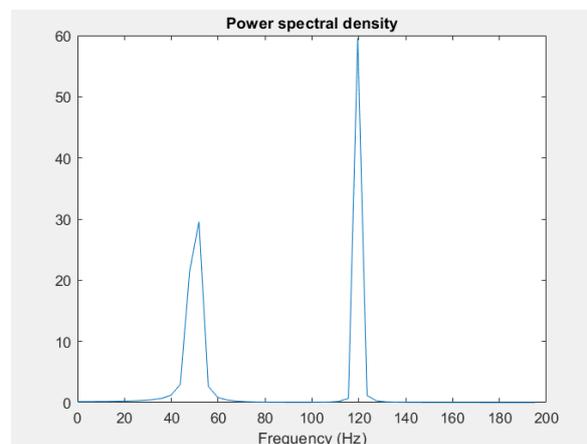
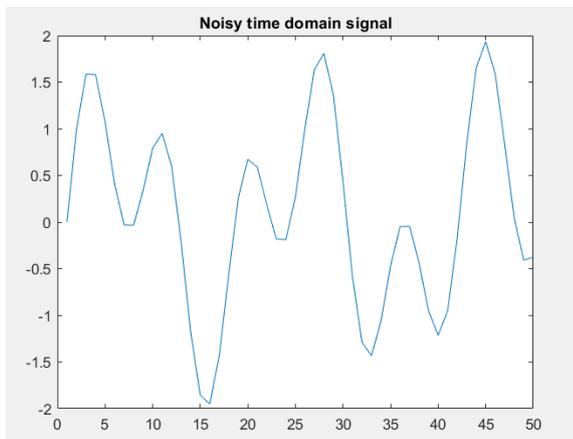
4.4.2 DFT: analisi spettrale di potenza

Tutti i circuiti elettrici hanno sempre del rumore sovrapposto, a partire dai casi di interesse più popolari come rumore bianco e rumore termico di un bipolo elettrico.

Come primo esempio è allora interessante analizzare un segnale con due componenti sinusoidali e vedere che cosa succede se si sovrappone ad esso del rumore, dal punto di vista dello spettro di potenza.

Si consideri il segnale $x(t) = \sin(2\pi 50t) + \sin(2\pi 120t)$

Dapprima effettua un'analisi nel dominio del tempo, dal punto di vista della forma d'onda e lo spettro di potenza del segnale nel dominio della frequenza.



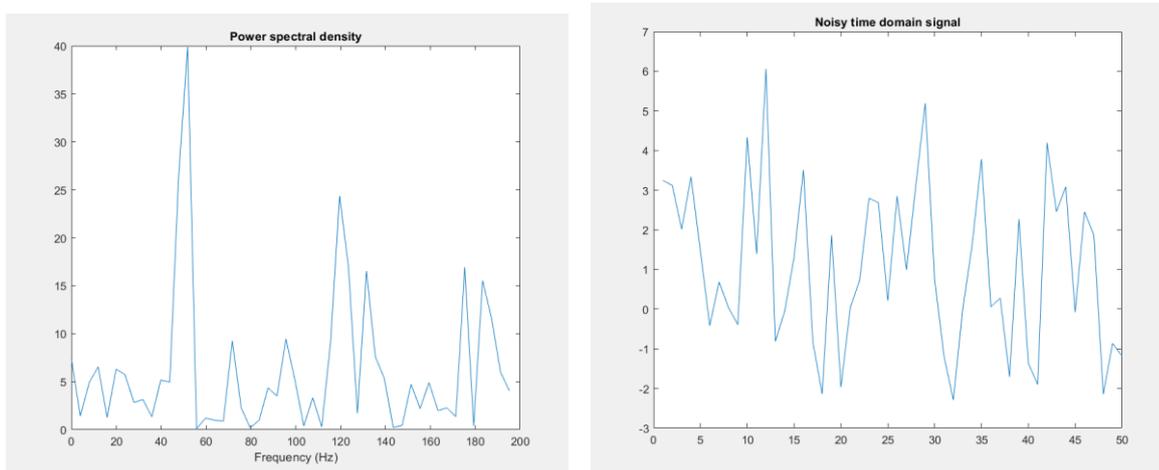
Non c'è da sorprendersi su ciò che si vede: la potenza è concentrata nelle due frequenze ed è mostrato il semiasse positivo.

Sfruttando la funzione `randn` si generano dei numeri distribuiti secondo una gaussiana di media zero e larghezza 1: si produce il cosiddetto Gaussian Noise. Per ottenere numeri distribuiti in modo normale con media e larghezza qualunque, basta moltiplicare per la larghezza desiderata ed aggiungere la media.

Con il seguente codice, eseguiamo l'esperimento:

```
t = 0:.001:.25;
x = sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t));
plot(y(1:50))
title('Noisy time domain signal')
Y = fft(y,251);
Pyy = Y.*conj(Y)/251;
f = 1000/251*(0:127);
plot(f(1:50),Pyy(1:50))
title('Power spectral density')
xlabel('Frequency (Hz)')
```

In output possiamo osservare quanto sia “fastidioso” ciò che avviene nel dominio del tempo e che si riversa sulla visualizzazione del contenuto di potenza spettrale del segnale: viene mutato.



Un'osservazione utile per l'esempio successivo è che i picchi di potenza più elevati sono i dati più fedeli con lo spettro originale.

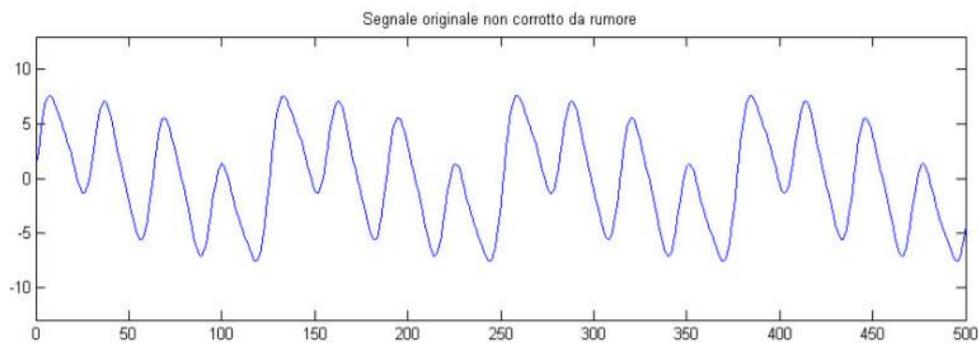
Sorge spontaneo chiedersi: è possibile sfruttare la DFT, in qualche modo, per poter evitare di cadere in tali visualizzazioni “prive di senso” e che distolgono l'attenzione dal segnale originale? La risposta si chiama digital filtering. E' possibile filtrare una certa banda di frequenze. Tramite il filtro passa-banda, per esempio, è possibile filtrare un segnale con del rumore sovrapposto. L'esempio precedente, però, non era rivolto al rumore, ma solo al filtraggio di impulsi. Verrà affrontato nei dettagli più avanti. Si pone un nuovo quesito: è possibile “riarrangiare” il risultato? Anche qui esiste una risposta ed è affermativa.

Si consideri allora l'esempio successivo.

In questo caso il rumore viene generato da Matlab con la funzione `awgn`; un esempio di chiamata per questa funzione è: `x = awgn(x, SNR, 'measured')` dove:

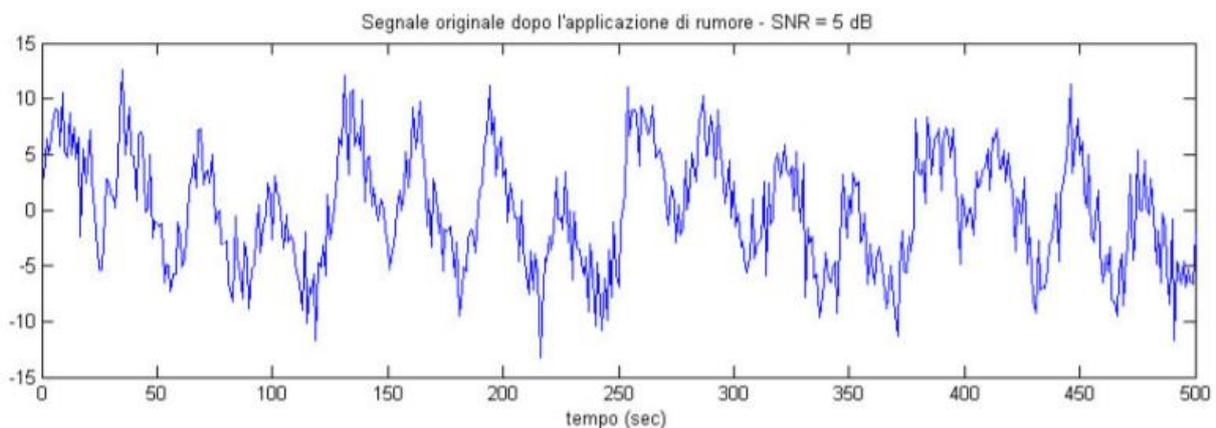
- `x` è il segnale da sommare con rumore gaussiano bianco a media nulla
- `SNR` è il rapporto segnale rumore da impostare espresso in dB
- `'measured'` è la stringa che passata come parametro che indica alla funzione di calcolare la potenza di `x` prima di sommare il rumore.

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



Il segnale è stato stimato nell'intervallo $[0,500]$ e dunque per 500 valori di ordinata.

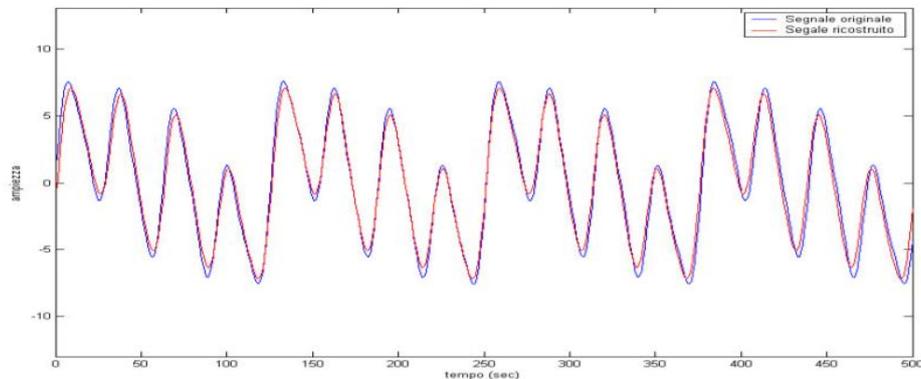
Il rapporto segnale-rumore è il rapporto, normalmente espresso in decibel, tra il valore efficace del segnale di ingresso e il valore efficace del rumore, cioè di tutte le componenti spettrali presenti, ad eccezione delle armoniche e della componente continua. Dunque, come già detto, usando la funzione `"awgn"` è possibile, per esempio, sovrapporre 5 decibel di rumore. Si ottiene:



In cui è immediato notare come l'escursione picco-picco sia quasi raddoppiata. Come appare evidente, il segnale corrotto da rumore sembra non aver più niente a che vedere con il segnale originario. E' curioso 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.

Frequenze_reali (rad/sec)	Frequenze_stimate (rad/sec) con 128 punti	Frequenze_stimate (rad/sec) con 256 punti	Frequenze_stimate (rad/sec) con 512 punti
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

Per l'esistenza delle reciproche relazioni, i campionamenti nel tempo e nella frequenza sono strettamente correlati: a livello di punti e di passo di discretizzazione. E' naturale che utilizzando un numero di punti per la FFT pari alla lunghezza del segnale nel dominio del tempo (in modo da avere la IDFT pari alla lunghezza del segnale originario), in questo caso 500, la ricostruzione è quasi perfetta rispetto al grafico di partenza:



Come osservato nell'esempio precedente, i picchi di potenza sono i punti più fedeli del grafico dello spettro di potenza e dunque utili per risalire alle componenti utili del segnale (quelle che erano presenti fin dapprima che venisse sovrapposto rumore) ed effettuare una coerente ricostruzione (sintesi) del segnale. Nella pratica: calcolare la FFT ad N punti del segnale di partenza, ottenendo l'uscita y e calcolare lo spettro di potenza, valutando per ogni campione la potenza reale, ma considerare un numero di componenti frequenziali p di cui è composto sicuramente il segnale di partenza (p componenti che non includono neanche una componente del segnale sovrapposto o rumore). Dunque identificare le frequenze corrispondenti ai p-picchi in potenza più elevati. Di fatto si può, nel vettore y, annullare tutte le componenti tranne quelle relative alle p frequenze : calcolare la IDFT di y come $\text{ifft}(y)$ significa ottenere il segnale ricostruito e dunque una buona approssimazione: perché si è effettuata una pulizia del segnale.

Un altro esperimento interessante potrebbe essere quello di cercare di modificare l'output dei dati sia con iterate la DFT che eseguire la tecnica di zero-padding, sia nel dominio del tempo che nel dominio della frequenza.

Si campiona un segnale a banda limitata considerando un numero abbastanza elevato di campioni e si calcola la DFT. Dopo aver effettuato la decimazione dei campioni eliminando quelli di posto pari nel dominio del tempo, si itera la DFT: lo spettro periodico dovrà avere le repliche ad una distanza inferiore.

Si consideri un seno cardinale e troncato al 20esimo lobo.

$$y(t_k) = \frac{\sin\left(\frac{2\pi 1000 t_k}{2}\right)}{\frac{2\pi 1000 t_k}{2}} \text{ dove } t_k = kT \text{ con } T=0.2\text{ms periodo di campionamento.}$$

In generale, tale segnale tanto popolare, ha questa definizione:

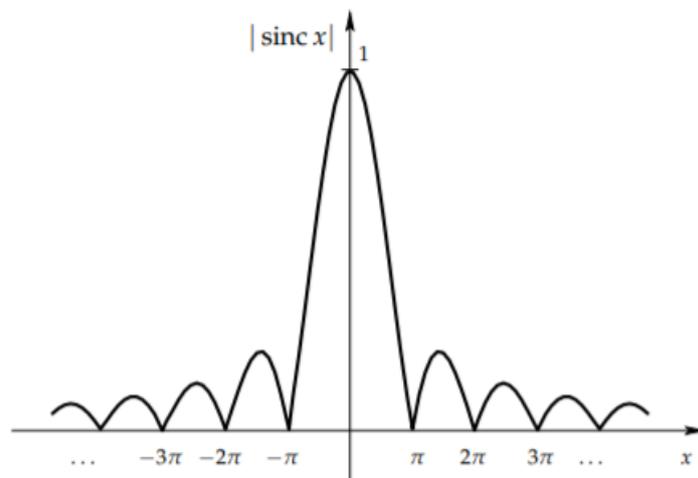
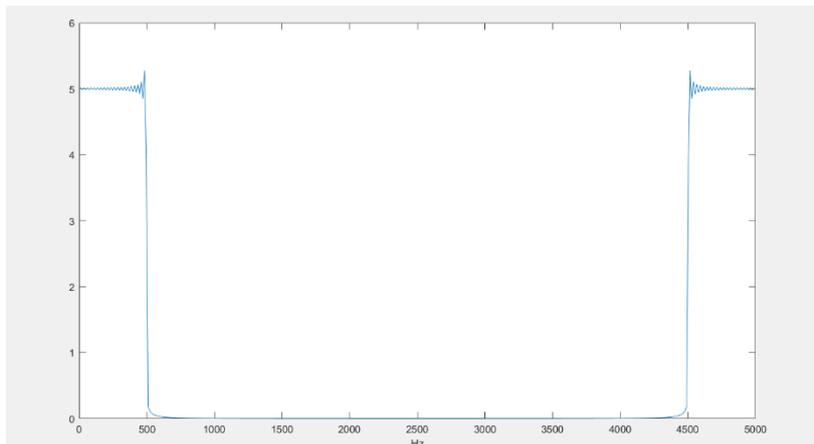
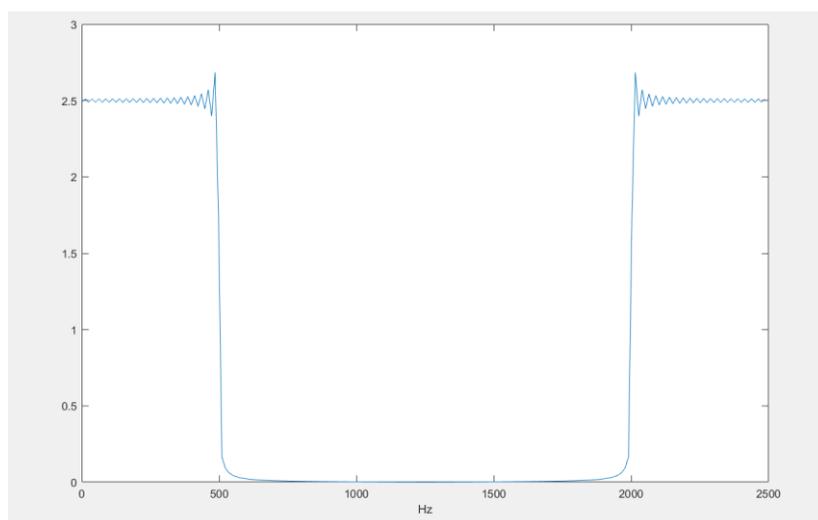


Figura 38: segnale sinc

Si esegue una prima dft e ciò che importa è tenere a mente quanto è largo l'asse delle ascisse:

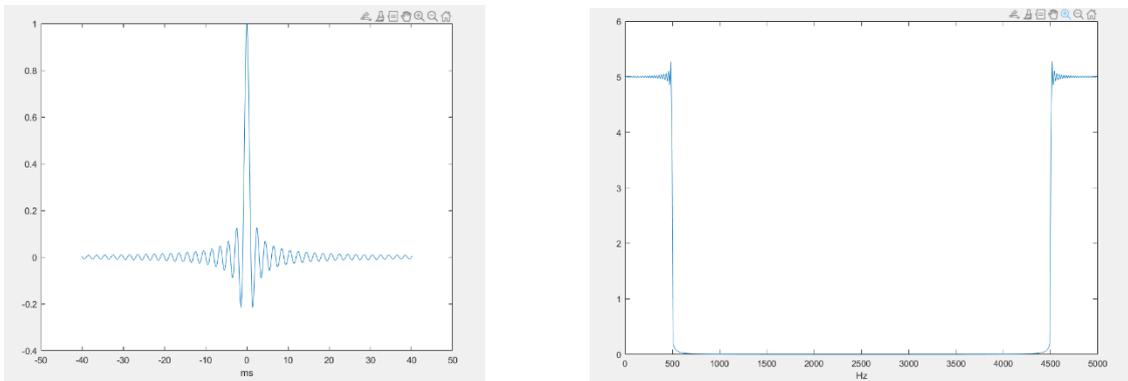


Ripetendo tale operazione si riesce a restringere i lobi.

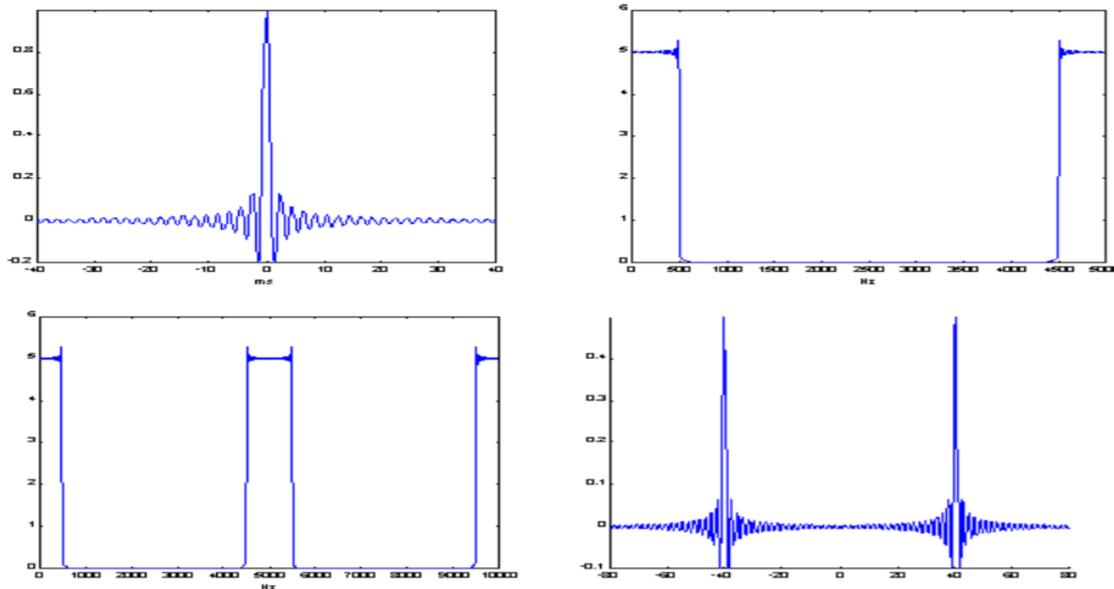


Come si era già affrontato, la densità del grafico, a livello di campioni, può essere regolata con la tecnica di zero-padding, ma il punto cruciale è che grazie, ancora una volta, all'esistenza delle reciproche relazioni, è possibile di fatto arrivare allo stesso risultato: entrambe le vie portano allo stesso grafico.

Partendo dal grafico del sinc, si compie la decimazione dei campioni



Invertendo tale processo e confrontando con i primi due grafici, si ha:



La sola osservazione di tale esperimento è proprio la regolazione del campionamento tramite l'aggiunta di zeri.

Si può seguire ciò che è avvenuto con il seguente codice.

```
k=[-201:1:201];
f0=1000;
fc=5000;
x=k/fc;
y=sin(pi*f0*x)./(pi*f0*x);
y(202)=1;
```

```

plot(x*1000,y); %in ms---->sec
xlabel('ms');
ff=fft(y);
figure
freq=[0:5000/403:5000*(1-1/403)];
plot(freq,abs(ff))
xlabel('Hz');
zoom
j=1;
%loop nel tempo
for i=1:403
    if i==2*j
        yi(j)=y(i);
        j=j+1;
    end
end
ff_i=fft(yi);
m=length(yi);
freq=[0:10000/m:10000*(1-1/m)];
figure
plot(freq,abs(ff_i))
xlabel('Hz');
j=1;
%loop in frequenza
for i=1:403
    ff_i(j)=ff(i);
    ff_i(j+1)=0;%zero padding
    j=j+2;
end
yii=ifft(ff_i);
n=length(yii);
xi=[2*x(1):(x(403)-x(1))/402:2*x(403)+(x(403)-x(1))/402];
figure
plot(xi*1000,real(yii))

```

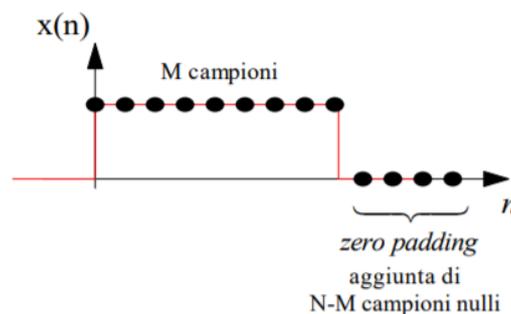
4.5 Applicazione DFT: calcolo di convoluzioni

Nel caso dei sistemi lineari tempo-invarianti tempo-continui, l'uscita del sistema nel dominio del tempo è la convoluzione tra l'ingresso e la risposta all'impulso; dualmente, nel dominio della frequenza, questo corrisponde al prodotto delle rispettive trasformate di Fourier. Ancora, il prodotto di due segnali nel dominio del tempo corrisponde alla convoluzione lineare o aperiodica delle rispettive trasformate, detta anche convoluzione tout-court.

Analoghe proprietà valgono anche per i sistemi lineari tempo-invarianti tempo-discreti: la convoluzione, secondo la sua definizione tempo-discreta, tra due sequenze numeriche, rappresentanti segnali campionati, equivale al prodotto delle trasformate delle stesse.

Ci si vuole interrogare su un fatto curioso: “E' possibile trovare la sequenza di uscita del sistema moltiplicando (termine a termine) le DFT dell'ingresso e della risposta all'impulso e poi applicando la IDFT su quanto trovato?”.

Un requisito fondamentale, affinché l'operazione di moltiplicazione abbia un senso, è che i rispettivi campioni delle trasformate dei due segnali a tempo-continuo siano prelevati nelle stesse frequenze. Infatti, la moltiplicazione termine a termine tra due vettori è, di fatto, possibile solo se questi sono di ugual dimensione e, nel contesto della DFT, questo accade se si considera il medesimo passo di campionamento per entrambe le due trasformate. D'altra parte si sa: tramite le reciproche relazioni, esiste uno stretto legame tra campionamento in frequenza e campionamento nel tempo. Dunque se le due trasformate devono avere lo stesso passo di discretizzazione, allora anche nel dominio del tempo i due segnali continui dovranno essere campionati con lo stesso periodo e questo implica che i due vettori abbiano stessa lunghezza. Il ragionamento sembra non fare una piega: ma quando si parla di sequenze provenienti, ad esempio, da segnali che decrescono a zero in tempi diversi, è allora ovvio che si avranno meno campioni per la sequenza proveniente dal campionamento del segnale più rapido nel convergere a zero ed estinguersi. Questo, però, non provoca un problema. Data la sequenza più corta, possiamo aggiungere, in coda, tanti zeri quanti ne mancano per eguagliare la lunghezza dell'altra sequenza. Applichiamo un'operazione di zero padding.



Descrizione del metodo dello “zero padding” nel caso in cui $x(n)$ sia frutto del campionamento di un rettangolo: agli M campioni ottenuti dal campionamento vengono aggiunti, in coda, $N-M$ campioni nulli, in modo da ottenere un numero N di campioni

Figura 39: zero padding

Solitamente è la sequenza risposta all'impulso ad essere riempita di zeri fino ad arrivare alla lunghezza della sequenza sollecitante il sistema.

Si fa attenzione ad un fatto molto sottile. Per il teorema della convoluzione la sequenza corrispondente all'uscita del sistema è ottenuta come somma di convoluzione ciclica tra le due sequenze periodiche corrispondenti all'ingresso e alla risposta impulsiva, entrambe del medesimo periodo. Finora non si è parlato di sequenze periodiche e dunque non vi è la possibilità di poter dare nulla per scontato: la moltiplicazione tra le due DFT non necessariamente equivale alla convoluzione desiderata: quella lineare.

A proposito di ciò, è necessario fare un focus sulle definizioni e le differenze a seconda del caso, poiché il discorso diventa sottile.

Per quanto riguarda una sequenza aperiodica vi sono le seguenti definizioni di Trasformata e Antitrasformata:

-definizione di sequenza aperiodica tramite antitrasformata di Fourier:

$$x[n] = T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} X(f) e^{j2\pi n f T} df$$

-definizione di trasformata di Fourier di una sequenza aperiodica:

$$X(f) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T}$$

-relazione del campionamento nel tempo tra segnale discreto e segnale analogico:

$$x[n] = x(nT)$$

-sviluppo in serie di Fourier in forma complessa di un segnale periodico:

$$x(t) = \sum_{n=-\infty}^{\infty} (c_n e^{j2\pi n f_0 t})$$

-definizione di coefficiente complesso della serie di Fourier

$$c_n = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t) e^{-j2\pi n f_0 t} dt$$

Ogniquale volta il segnale tempo-continuo è periodico, esso possiede uno spettro discreto: lo dice lo sviluppo in serie di Fourier. Viceversa, se il segnale è discreto nel tempo, esso possiede uno spettro periodico. Questo è l'ennesimo riflesso della dualità dei domini di tempo e frequenza. Da quest'ultimo punto di vista, è interessante notare che il segnale discreto aperiodico $x[n]$ è in pratica la successione dei coefficienti di Fourier dell'espansione in serie della funzione $X(f)$, periodica nella variabile continua frequenza; ciò in piena dualità rispetto al caso del segnale periodico nel tempo continuo $x(t)$ con la sua propria successione discreta dei coefficienti di Fourier.

Per quanto riguarda una sequenza periodica vi sono le seguenti definizioni di Trasformata discreta e Antitrasformata discreta per i segnali periodici a tempo discreto. Si notano subito le analogie con le corrispondenti relazioni di sintesi e analisi per i segnali periodici a tempo continuo

$$x[n] = \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi kn}{N}}$$

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

Sono state richiamate le definizioni generali di tutti i casi possibili. Per prima cosa è stata definita la trasformata di Fourier di una sequenza aperiodica $X(f)$, che risulta una funzione continua e periodica nella frequenza f di periodo pari alla frequenza di campionamento $1/T$, ma che peraltro gode di proprietà molto simili a quelle della trasformata continua di Fourier $X(f)$ per i segnali analogici. Si può sottolineare che:

-un segnale a tempo continuo e periodico commuta, tramite serie, in uno spettro discreto aperiodico;

-un segnale a tempo continuo aperiodico commuta, tramite trasformata, in uno spettro continuo;

- un segnale a tempo discreto periodico commuta, tramite trasformata discreta, in uno spettro discreto periodico;
- un segnale a tempo discreto aperiodico commuta, tramite trasformata continua, in uno spettro discreto.

Queste differenze, tra sequenze periodiche e aperiodiche, nel contesto in cui sono state richiamate, convergono nella somma di convoluzione:

-nel caso di due sequenze aperiodiche x ed y si ha:

$$x * y = \sum_{k=-\infty}^{+\infty} x[n-k]y[k];$$

-nel caso di due sequenze periodiche x ed y su ha:

$$x * y = \frac{1}{N} \sum_{k=-\infty}^{+\infty} x[n-k]y[k].$$

Per quanto detto finora allora risulta che l'operazione di prodotto delle trasformate delle due sequenze, ingresso e risposta impulsiva, equivale alla particolare convoluzione delle due sequenze in versione "periodicizzate". Questo si può vedere dalle formule di Poisson e, in particolare, proprio su ciò si basa il Teorema del campionamento (Teorema di Shannon) e la Condizione di Nyquist.

Viene proposto e commentato un esperimento: arrivare in qualche modo allo stesso risultato sia con la convoluzione ciclica che aciclica servendosi della tecnica dello zero-padding. Si considerino le due sequenze:

$$h[n] = [h(-1) \ h(0) \ h(1) \ h(2)] = [1 \ 2 \ 1 \ -1] \quad x[n] = [x(0) \ x(1) \ x(2) \ x(3)] = [1 \ 2 \ 3 \ 1]$$

e si applichi lo zero-padding ad entrambe, ottenendo:

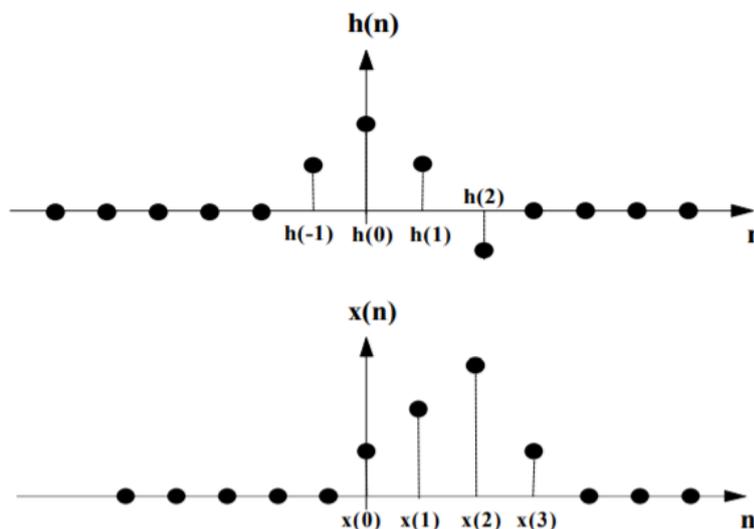


Figura 40: zero padding

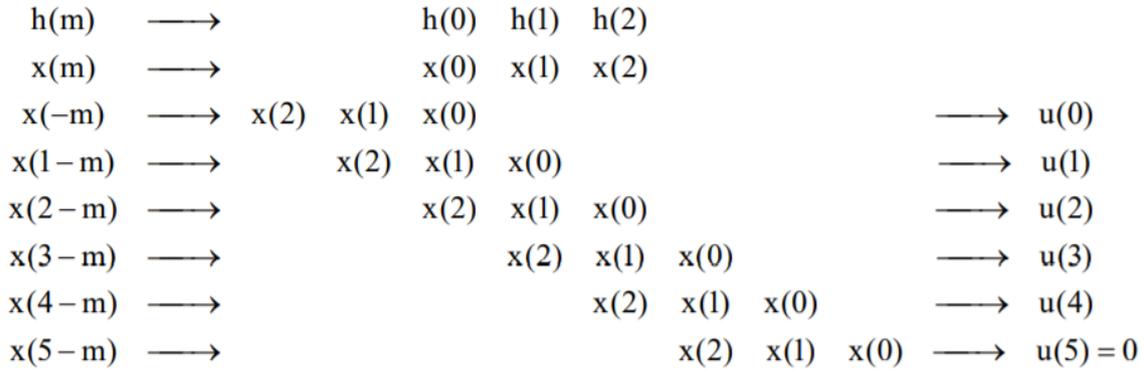
Operativamente, si deve iniziare ribaltando la sequenza h rispetto all'origine per ottenere h[-n] e procedere con la moltiplicazione termine a termine con x e sommare (si può considerare come un prodotto scalare tenendo "fisso" l'altro fattore).

Il risultato, l'uscita del sistema, è:

$$y[n] = [0 \ 0 \ 1 \ 4 \ 8 \ 8 \ 3 \ -2 \ -1 \ 0] \text{ con } y(0) = 4.$$

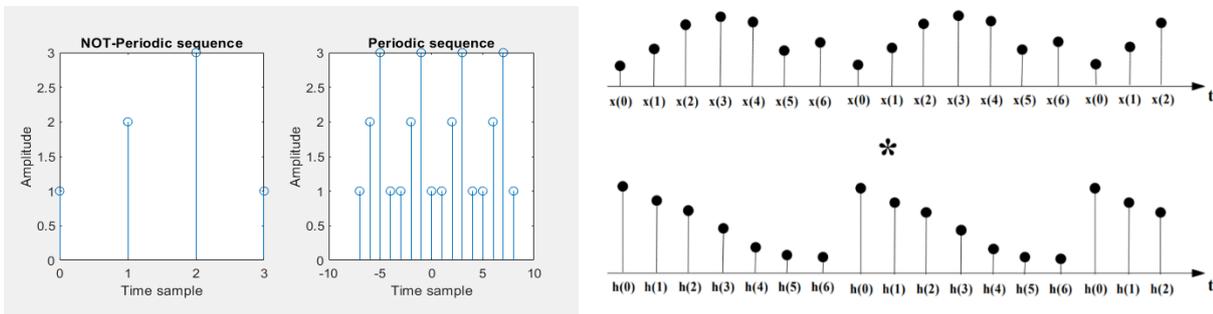
In generale, prendendo come esempio $z(t)=x(t)h(t)=\text{sinc}(2Bt)\text{sinc}(2Bt)$, si nota che le funzioni che convolvono, $X(f)$ e $H(f)$, hanno una "estensione frequenziale" (cioè sono diverse da zero su di un intervallo di ampiezza) pari a $2B$, mentre il loro integrale di convoluzione ha estensione pari a $4B$. In generale l'estensione dell'integrale di convoluzione tra due funzioni aventi estensione limitata è data dalla somma delle due estensioni stesse. Questo fatto si conserva anche nel dominio discreto e nella convoluzione di due sequenze numeriche di lunghezza N , si ha come output ancora una sequenza, ma di lunghezza $2N-1$, ossia lunghezza 7.

Si vede subito anche tramite un semplice schema:



Ripetendo il calcolo tramite la DFT, allora dobbiamo periodicizzare le sequenze.

Graficamente avremo:

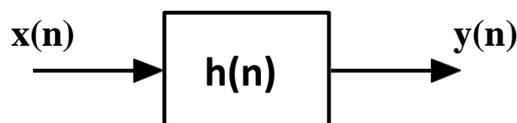


Nella seconda immagine si è periodicizzato fino alla lunghezza sette. Nel primo grafico si mette in risalto l'idea di periodicizzare. E' inutile non anticipare che si otterranno due risultati diversi.

Il calcolo si estende su un altro esempio per mettere in risalto come calcolare la convoluzione circolare per via grafica e con la DFT. Il calcolo viene ripetuto con:

- convoluzione circolare grafica;
- convoluzione circolare tramite DFT;
- convoluzione tout court o aciclica (lineare) con zero-padding.

Si consideri nuovamente un sistema del tipo:



$$x[n] = [2 \ 1 \ 2 \ 1] \quad h[n] = [1 \ 2 \ 3 \ 4]$$

- Calcolando la convoluzione circolare, si ottiene:

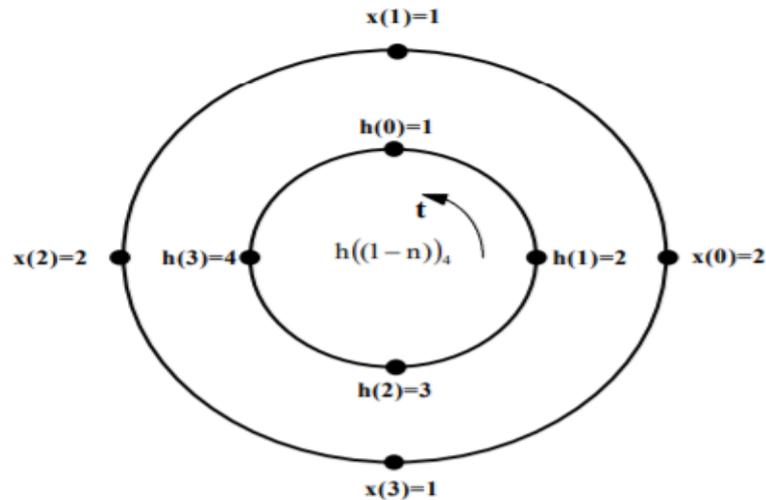


Figura 41: rappresentazione grafica della convoluzione circolare

$$y[n] = \sum_{n=0}^{4-1} y[m-n] x[n]$$

$$\begin{aligned} N=0 &\rightarrow y(0) = 14 \\ N=1 &\rightarrow y(1) = 16 \\ N=2 &\rightarrow y(2) = 14 \\ N=3 &\rightarrow y(3) = 16 \\ y[n] &= [14 \ 16 \ 14 \ 16] \end{aligned}$$

- Calcolando la convoluzione circolare tramite la DFT, si ottiene:

$$\begin{aligned} X_k &= [6 \quad 0 \quad 2 \quad 0] \\ H_k &= [10 \quad -2+j2 \quad -2 \quad -2-j2] \\ Y_k &= [60 \quad 0 \quad -4 \quad 0] * 1/N \text{ dove } N = 4 \\ y[n] &= (60 - 4 \cdot e^{j\pi n}) * 1/4 \\ \text{Facendo variare l'indice } n, \text{ troviamo:} \\ y[n] &= [14 \ 16 \ 14 \ 16] \end{aligned}$$

- Calcolando la convoluzione lineare, si ottiene :

si effettua l'operazione di zero-padding

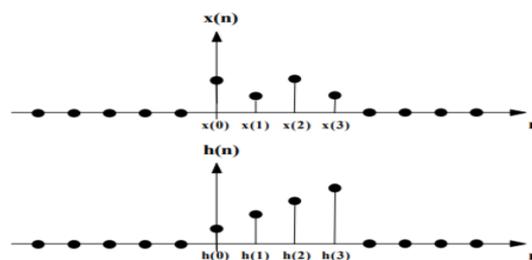


Figura 42: zero padding

Banalmente, riflettendo h rispetto all'asse verticale ed effettuando la sovrapposizione delle due figure, si ottiene una somma uguale a 2 e corrispondente a y(0) e dunque il risultato è già profondamente diverso.

Il problema di fondo è il kernel della trasformata discreta di Fourier o della Z-trasformata ristretta al cerchio unitario: usare tale trasformata equivale ad un campionamento in frequenza e quindi una periodizzazione nei tempi e il prodotto di due DFT equivale alla convoluzione circolare tra le rispettive sequenze temporali. Basti pensare che la stessa trasformata di una sequenza aperiodica è già una funzione periodica proprio a causa del kernel.

Una prima osservazione è la seguente. Se si considerano due sequenze generiche aperiodiche e di durata finita, di lunghezza L ed M, allora dalla loro convoluzione lineare sarà una stringa numerica di L+M-1 elementi e la nostra DFT dovrà per forza avere tale lunghezza. Nel dominio della frequenza sarà necessario prendere un numero di campioni N che sia maggiore o, al più, uguale alla dimensione della stringa risultante. Ma siccome abbiamo che la trasformata di una sequenza aperiodica è in generale:

$$Y(f) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j2\pi n f T} \text{ ed in particolare questa è definita come il prodotto } X(f)H(f), \text{ adattando}$$

l'espressione sulla base del fatto che i due vettori della frequenza hanno lunghezza finita e devono averla di almeno N elementi: la frequenza viene discretizzata (come già era ovvio che succedesse) e la lunghezza N sarà la medesima per rendere lecito il prodotto. Si ottiene la DFT di una sequenza aperiodica:

$$Y(k) = \sum_{n=-\infty}^{+\infty} y[n] e^{-\frac{j2\pi kn}{N}}$$

Implicitamente si è definito il campionamento nel dominio del tempo e con la lunghezza di N elementi (gli estremi della somma si restringeranno a 0 ed N-1) e dunque, partendo dal fatto che abbiamo considerato nell'esempio le seguenti sequenze:

$$x[n] = [2 \ 1 \ 2 \ 1] \quad h[n] = [1 \ 2 \ 3 \ 4] \text{ si effettua lo zero-padding:}$$

$$x[n] = [2 \ 1 \ 2 \ 1 \ 0 \ 0 \ 0] \quad h[n] = [1 \ 2 \ 3 \ 4 \ 0 \ 0 \ 0]$$

Dal punto di vista spettrale, si affronterà quale è il duplice beneficio di questa operazione nella sezione dedicata all'analisi spettrale.

A questo punto, abbiamo risolto il problema: infatti, avendo imposto le condizioni per cui Y(k) sia univocamente rappresentativa da y(n), abbiamo automaticamente garantito che il prodotto di X(k) ed H(k) dia, come risultato, la DFT della convoluzione lineare di x(n) e h(n). La questione più interessante è stato farlo ragionando per biunivocità (biattività della funzione trasformata di Fourier), dualità (campionamento in frequenza, nel tempo) e prodotto di convoluzione. Accertare che tale prodotto (delle due DFT) equivalga alla convoluzione circolare delle rispettive sequenze implica, infine, la deduzione che, nelle condizioni specificate, la convoluzione circolare e la convoluzione lineare danno lo stesso risultato.

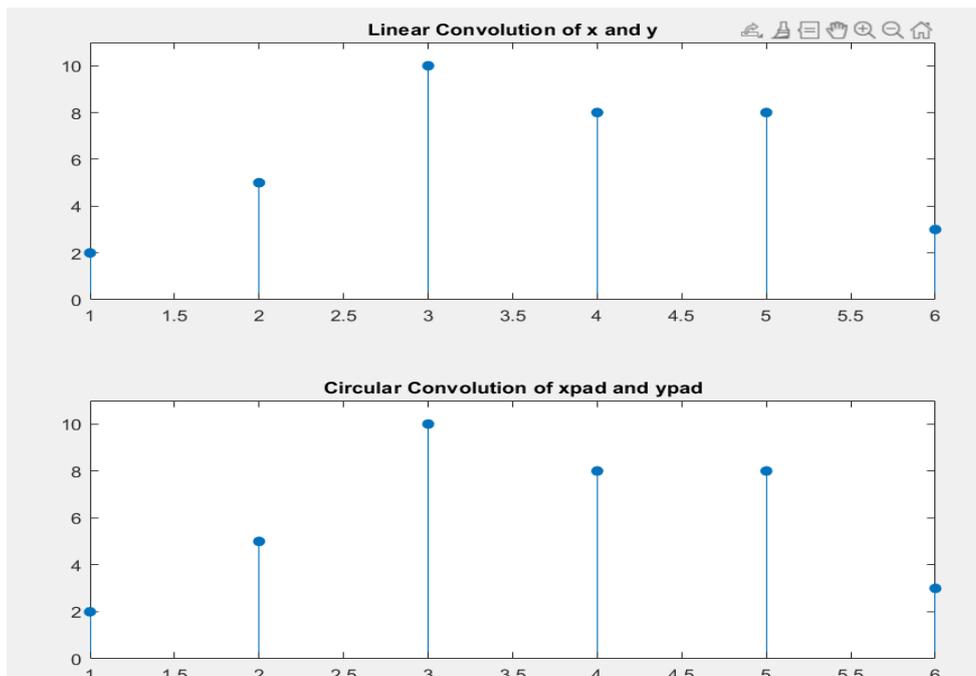
Il motivo di questa coincidenza di risultati è proprio nell'aggiunta degli zeri: *l'allungamento di x(n) e h(n) con un numero opportuno di zeri (tali da portare ad N la lunghezza di entrambe le sequenze) ha forzato la convoluzione circolare a dare gli stessi risultati ottenibili con la convoluzione lineare*, in quanto l'aggiunta degli zeri ha creato lo spazio necessario a far sì che la convoluzione non si riavvolgesse sulla parte iniziale dei dati.

In parole povere: la convoluzione lineare e circolare sono operazioni fondamentalmente diverse. Tuttavia, ci sono condizioni in cui la convoluzione lineare e circolare sono equivalenti. Stabilire questa equivalenza ha importanti implicazioni. Per due vettori, x e y, la convoluzione circolare è uguale alla trasformata di Fourier discreta inversa (IDFT) del prodotto delle DFT dei vettori. Conoscere le condizioni in cui la convoluzione lineare e quella circolare sono equivalenti consente di utilizzare la DFT per calcolare in modo efficiente le convoluzioni lineari, il cui costo computazionale è veramente elevato. E' interessante vedere, a

livello asintotico, la differenza tra i due costi computazionali, ossia quanto veramente sia efficiente la DFT. Verrà analizzato più avanti.

Di seguito un esempio in Matlab di quanto detto finora.

```
x = [2 1 2 1];  
y = [1 2 3];  
clin = conv(x,y);  
xpad = [x zeros(1,6-length(x))];  
ypad = [y zeros(1,6-length(y))];  
ccirc = ifft(fft(xpad).*fft(ypad));  
subplot(2,1,1);  
stem(clin,'filled');  
ylim([0 11]);  
title('Linear Convolution of x and y')  
subplot(2,1,2);  
stem(ccirc,'filled');  
ylim([0 11]);  
title('Circular Convolution of xpad and ypad');
```



Il grafico conferma palesemente il risultato e quanto detto.

4.5.1 DFT: overlap and save

Nelle applicazioni pratiche che coinvolgono il filtraggio lineare dei segnali, la sequenza di ingresso $x[n]$ è spesso una sequenza molto lunga; questo accade, tipicamente, in alcune applicazioni di elaborazione del segnale in tempo reale. Non potendo memorizzare segnali, sia pure campionati, di lunghezza infinita, abbiamo la necessità di segmentare sequenze molto lunghe in blocchi di dimensione più piccola e costante; questo comporta il fatto di dover combinare, in modo opportuno, i singoli blocchi con la funzione di risposta all'impulso $h[n]$ e di combinare i risultati ottenuti al fine di ricavare l'uscita complessiva. La possibilità di

risalire comunque ad una espressione corretta dell'uscita, nonostante l'*elaborazione a pacchetti*, è garantita dalla linearità del filtraggio, la cui relazione fondamentale è appunto $Y(f)=H(f)X(f)$.

Finora sono stati trattati diversi punti cruciali, tra cui:

- in primo luogo, il prodotto tra le DFT di due sequenze di durata finita da, in generale, la DFT corrispondente alla convoluzione circolare delle medesime;
- se, invece, allunghiamo le sequenze con un numero opportuno di zeri in coda, allora la convoluzione circolare viene forzata a diventare uguale alla convoluzione lineare.

Si affronta come varia la situazione con delle sequenze illimitate, ossia quando si è costretti a spezzare $x[n]$ in blocchi di lunghezza finita.

Problema:

- occorre acquisire l'intero segnale prima di poterlo filtrare e se il numero di impulsi è enorme, allora occorre una FFT che occupa troppa memoria.

Soluzione:

- Algoritmo "overlap and save".

Nel metodo dell'**overlap and save**, in particolare, si procede nel modo seguente:

- in primo luogo, si spezza la sequenza $x(n)$ in blocchi di lunghezza L ;
- ciascuno di tali blocchi viene allungato, in testa, con $M-1$ campioni facenti parte del blocco precedente (fa eccezione a questo solo il primo blocco, il quale, non avendo campioni precedenti, viene allungato con $M-1$ zeri); in tal modo, si ottengono blocchi $x_1(n)$, $x_2(n)$,... lunghi ciascuno $N=L+M-1$, come indicato nella figura seguente:

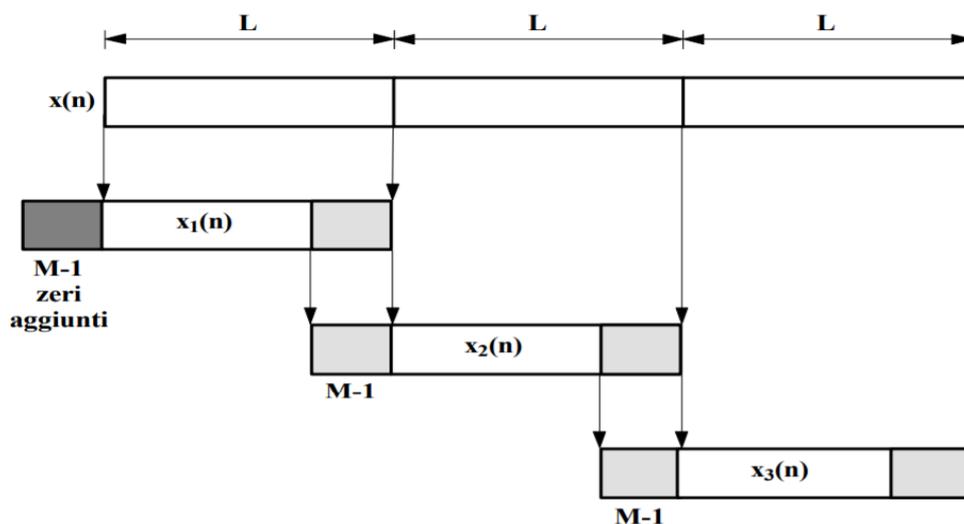


Figura 43: schema di decomposizione

- Viene eseguita la convoluzione lineare della sequenza h con ciascun pacchetto. L'esito della convoluzione è un blocco $u_i(n)$, di lunghezza $L+2(M-1)$, che presenta campioni non significativi solo sulle prime $M-1$ posizioni e sulle ultime $M-1$ posizioni (o meglio, privi di senso). Solo scartando

questi campioni non significativi e congiungendo i blocchi rimanenti, la sequenza di uscita risulta essere corretta.

Per rendere più significativo quanto descritto, si mostra un esempio tramite convoluzione lineare grafica. Per semplicità, supponiamo di spezzare la sequenza $x(n)$ in blocchi ciascuno di lunghezza $L=3$.

Per quanto detto, la prima operazione da fare è quella di selezionare i primi 3 campioni di $x(n)$ e di farli precedere da $M-1=2$ zeri, in modo da formare il blocco $x_1(n)$ lungo 5.

$h(m)$	→		$h(0)$	$h(1)$	$h(2)$				
$x_1(m)$	→		0	0	$x(0)$	$x(1)$	$x(2)$		
$x_1(-m)$	→	$x(2)$	$x(1)$	$x(0)$	0	0			→ NO
$x_1(1-m)$	→		$x(2)$	$x(1)$	$x(0)$	0	0		→ NO
$x_1(2-m)$	→			$x(2)$	$x(1)$	$x(0)$	0	0	→ $u_1(2)$
$x_1(3-m)$	→				$x(2)$	$x(1)$	$x(0)$	0	0
$x_1(4-m)$	→					$x(2)$	$x(1)$	$x(0)$	0
$x_1(5-m)$	→						$x(2)$	$x(1)$	$x(0)$
$x_1(6-m)$	→							$x(2)$	$x(1)$
$x_1(7-m)$	→								$x(2)$

In definitiva, se $x(n)$ viene partizionata in blocchi da 5, si ottengono 3 campioni significativi dell'uscita per ogni blocco (ossia i veri valori non nulli di x sono dello stesso numero della risposta all'impulso). Generalizzando, quando effettuiamo la convoluzione tra un blocco $x(n)$ di $N=L+M-1$ dati ed una risposta all'impulso $h(n)$ di M campioni, in uscita otteniamo L campioni significativi, che sono quelli intermedi.

Il prossimo pacchetto:

$h(m)$	→		$h(0)$	$h(1)$	$h(2)$				
$x_2(m)$	→		$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$		
$x_2(-m)$	→	$x(5)$	$x(4)$	$x(3)$	$x(2)$	$x(1)$			→ NO
$x_2(1-m)$	→		$x(5)$	$x(4)$	$x(3)$	$x(2)$	$x(1)$		→ NO
$x_2(2-m)$	→			$x(5)$	$x(4)$	$x(3)$	$x(2)$	$x(1)$	→ $u_2(2)$
$x_2(3-m)$	→				$x(5)$	$x(4)$	$x(3)$	$x(2)$	$x(1)$
$x_2(4-m)$	→					$x(5)$	$x(4)$	$x(3)$	$x(2)$
$x_2(5-m)$	→						$x(5)$	$x(4)$	$x(3)$
$x_2(6-m)$	→							$x(5)$	$x(4)$
$x(7-m)$	→								$x(5)$

Come già generalizzato per ciascun pacchetto.

Per quanto detto nel paragrafo 4.2.1 risulta ora scontato usare la DFT per agevolare la convoluzione lineare e renderla circolare usando la tecnica di zero-padding. Operativamente: si allunga la sequenza $h(n)$ con tanti 0 (in coda) quanti ne servono per arrivare ad una lunghezza $L+M-1$ pari a quella dei blocchi $x_i(n)$; si calcola la DFT delle due sequenze da convolvere. Infine la moltiplicazione delle due DFT e si effettua la IDFT.

La lunghezza minima richiesta sarebbe $N+M-1$, pari alla lunghezza della sequenza risultante; al contrario, si sceglie di calcolare le dft su N campioni. Questo comporta che il risultato della convoluzione sia una sequenza di lunghezza $L+M-1$, che però presenta aliasing sui primi $M-1$ campioni, che quindi andranno scartati.

Uno sketch per mostrare come si procede è:

- si spezza la sequenza $x(n)$ in blocchi di lunghezza L ;
- ciascuno di tali blocchi viene allungato, in testa, con $M-1$ campioni facenti parte del blocco precedente (fa eccezione a questo solo il primo blocco, che viene allungato con $M-1$ zeri); in tal modo, si ottengono blocchi $x_1(n), x_2(n), \dots$ lunghi ciascuno $N=L+M-1$, esattamente come visto prima;
- la sequenza $h(n)$ viene allungata con $L-1$ zeri in coda, in modo da avere lunghezza $L+M-1$ pari a quella dei blocchi con cui deve essere convoluta e forzare la convoluzione lineare a diventare circolare;
- si calcolano le DFT, su N campioni, del generico blocco $x_i(n)$ e di $h(n)$ (ovviamente, la DFT di $h(n)$ va calcolata una volta per tutte);
- si moltiplicano le due trasformate;
- si antitrasforma tramite una IDFT ancora basata (ovviamente) su N campioni;
- si ottiene in uscita un blocco $u_i(n)$ di lunghezza $L+M-1$ dove i primi $M-1$ campioni sono affetti da errore e si scartano;
- si effettua la ricostruzione:

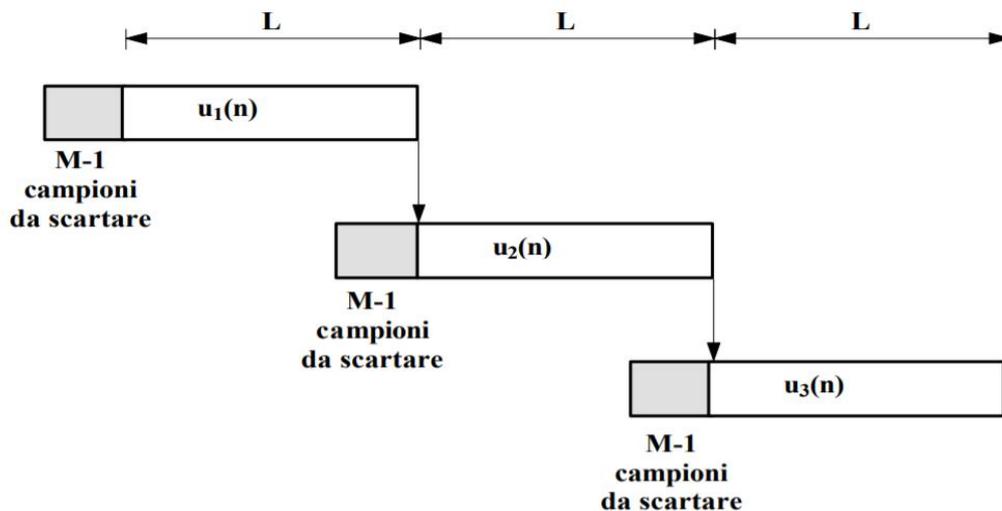


Figura 44: schema di decomposizione

Se $h(n)$ presenta molti campioni il vantaggio di usare una DFT, infatti, è proprio nell'efficienza computazionale con cui la DFT stessa può essere calcolata ossia tramite la FFT.

Riscrivendo l'algoritmo come una successione di passi, si ha:

1. Effettuo una FFT del filtro $h(n)$, zero-padded a N campioni e memorizzo:

$$h(n) = \begin{cases} h(n) & n = 0, 1, \dots, Q - 1 \\ 0 & n = Q, Q + 1, \dots, N - 1 \end{cases} \Rightarrow \text{FFT} \Rightarrow \text{H(f)}$$

2. Seleziono il blocco m-esimo di N punti dalla sequenza di ingresso x(n) :

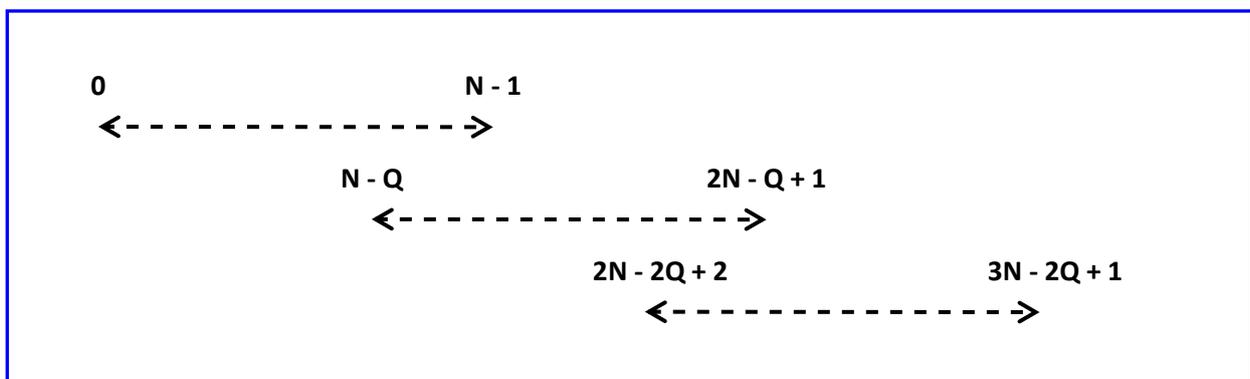
$$x_m(n) = x[n + (m - 1)(N - Q + 1) - (Q - 1)]$$

in cui: $n = 0, 1, 2, \dots, N-1$

$m = 1, 2, 3, \dots$

$N = \text{FFT length}$

$Q = \text{FIR length}$



3. Moltiplico lo spettro del filtro $H(f)$ per lo spettro FFT del blocco m .
4. Effettuo la trasformata inversa IFFT dello spettro risultante dal prodotto.
5. Butto via i primi $(Q-1)$ punti del blocco ottenuto allo step 4

$y_1(n)$	$n = Q - 1, \dots, N - 1$
$y_2[n - (N - Q + 1)]$	$n = N, \dots, 2N - Q$
⋮	⋮
$y_m[n - (m - 1)(N - Q + 1)]$	$n = (m - 1)(N - Q + 1) + (Q - 1),$ $\dots, (m - 1)(N - Q + 1) + (N - 1)$
⋮	⋮
⋮	⋮

e appendo gli $(N-Q+1)$ punti rimanenti alla sequenza di uscita $y(n)$: $y(n) = y_1(n), y_2(n), \dots, y_m(n), \dots$

Lo schema del metodo:

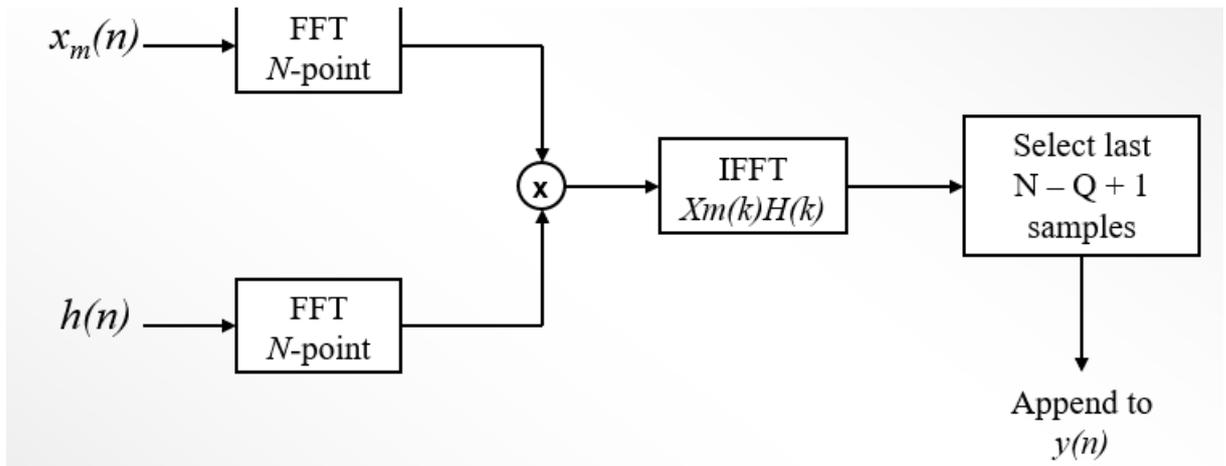


Figura 45: schema logico

In definitiva si può esporre un esempio in Matlab.

```

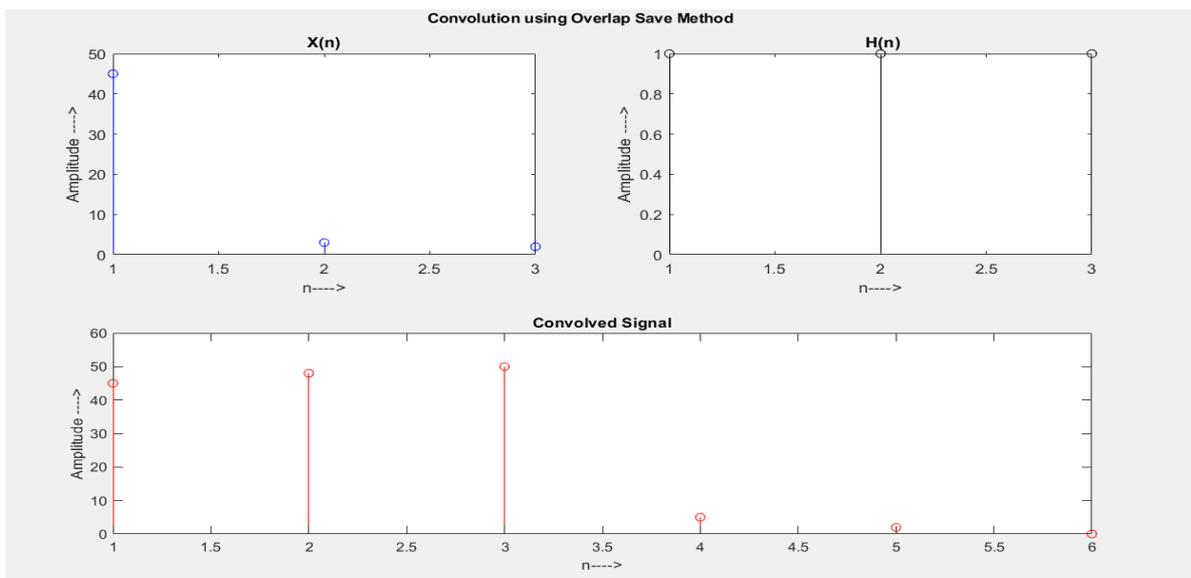
Enter 1st sequence X(n)= [45 3 2]
Enter 2nd sequence H(n)= [1 1 1]
Enter length of each block L = 3

Ma =
    45     3     2

Ma =
    45     3     2
     0     0     0

y =
    45.0000    48.0000    50.0000     5.0000     2.0000     0.0000
  
```

Ottenendo l'output:



4.5.2 Modulazione analogica: amplitude modulation (AM)

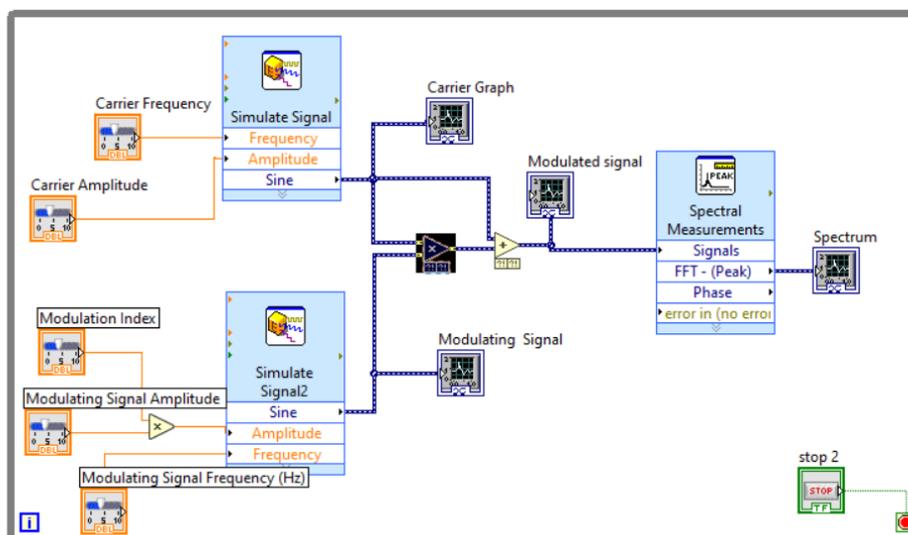
Oltre a essere utile per l'analisi dei circuiti, la trasformata di Fourier trova applicazione in una grande varietà di campi quali l'ottica, la spettroscopia, l'acustica, l'informatica e l'ingegneria elettrica in generale. Per l'ingegneria elettrica, essa trova in particolare vasta applicazione nei sistemi per le telecomunicazioni e nella elaborazione dei segnali, discipline per le quali la risposta in frequenza e il comportamento spettrale sono di vitale importanza. La modulazione di ampiezza di una radiazione elettromagnetica usata per trasmettere l'informazione attraverso lo spazio è diventata un elemento indispensabile per la vita della moderna società della tecnologia. La trasmissione di onde elettromagnetiche attraverso lo spazio risulta però efficiente ed economica soltanto alle radiofrequenze (sopra i 20 kHz). La trasmissione di segnali "intelligenti", quali per esempio quelli relativi al parlato o alla musica, limitati alla banda di bassa frequenza compresa tra 50 Hz e 20 kHz, si rivela perciò estremamente costosa, avendo bisogno di antenne molto grandi e di potenze molto elevate. Un metodo comune per trasmettere informazione audio a bassa frequenza è allora quello di trasmettere un segnale ad alta frequenza, detto portante, che viene in qualche modo controllato in modo da risultare in corrispondenza con l'informazione audio (sintonizzazione). Sono tre le caratteristiche di una portante (ampiezza, frequenza e fase) che possono venire controllate facendo in modo che trasportino il segnale intelligente, detto anche segnale modulante. Verrà qui trattato soltanto il caso di controllo della ampiezza della portante. L'espressione analitica del segnale modulato è descritta da tale equazione

$$x_c(t) = A_c \cos \omega_c t + A_c \mu x(t) \cos \omega_c t$$

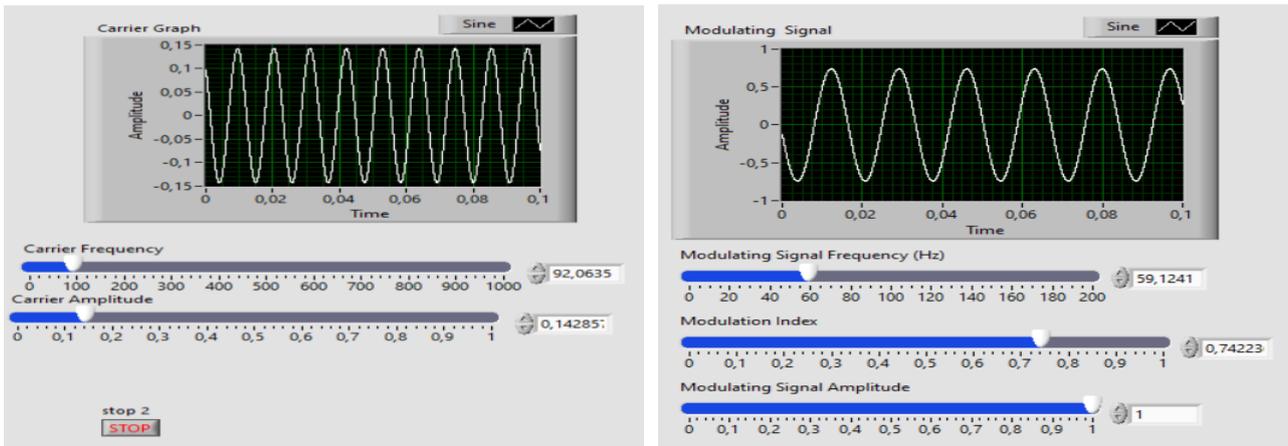
I cui parametri devono essere descritti:

- μ è *fattore di modulazione*
- x_c è *il segnale modulato in ampiezza*
- ω_c è *la frequenza portante del segnale portante che sintonizza il modulante, ossia x*

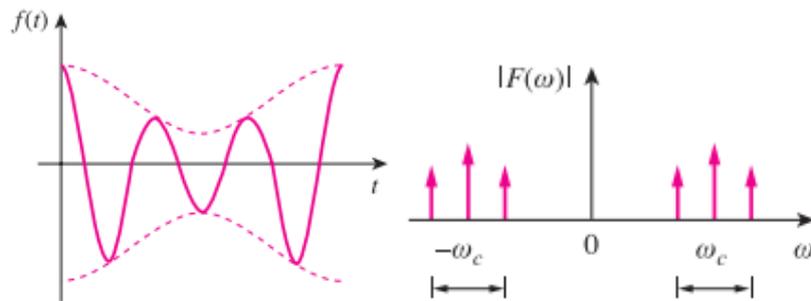
La stima dello spettro può essere effettuata tramite Labview e l'implementazione della DFT tramite FFT, tramite il seguente progetto:



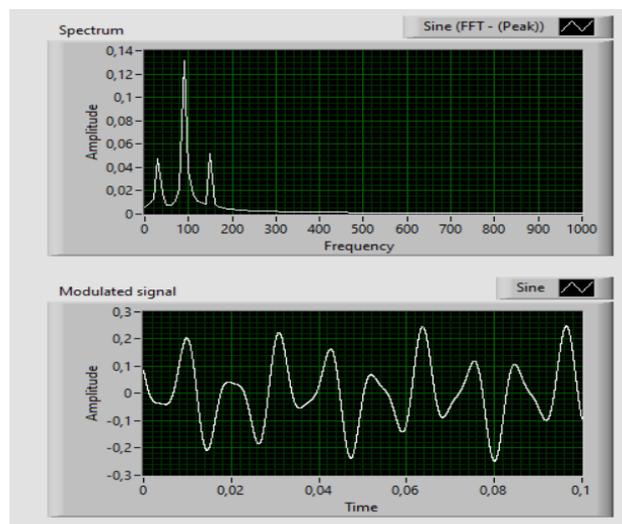
Si considerano i seguenti parametri.



Mandando in esecuzione il sistema, si vorrebbe arrivare alla rappresentazione perfetta del libro “Sadiku, Circuiti Elettrici”, riportato di seguito:



Come già dimostrato più volte, la convoluzione nel dominio del tempo viene stimata tramite prodotto delle DFT, adesso l'operazione di convoluzione si manifesta direttamente nella frequenza e l'approssimazione sarà effettuata su trasformate di Fourier a spettro discreto. La Trasformata sarà invece una funzione continua della frequenza e viene stimata tramite approssimazione delle trasformate e dunque la realtà sarà molto più “brutta”:



Si nota che lo spettro del segnale modulato, nell'asse positivo, non è quello del libro: non lo è per niente.

Oltre al fatto inevitabile, per natura dell'approssimazione, dello spettro continuo, l'altra irregolarità visiva è la forma d'onda nel dominio del tempo.

Di fatto, l'unico miglioramento possibile è la regolazione dell'indice di modulazione, che in questo caso è circa 0.7 .

4.6 DFT: background della FFT

4.6.1 Applicazione della FFT

L'algoritmo di FFT fu pubblicato nel 1965 da Cooley e Tukey; a questa data si fa risalire la nascita della moderna elaborazione numerica dei segnali. Attraverso la FFT, infatti, possono essere effettuate in maniera efficiente alcune operazioni fondamentali di analisi ed elaborazione dei segnali (analisi spettrale e filtraggio, come verrà mostrato nel seguito). Tali operazioni non furono realizzabili in pratica fino al momento dell'introduzione dell'algoritmo veloce, vista la ridotta velocità dei componenti elettronici e quindi dei calcolatori dell'epoca. Questo giustifica l'importanza centrale attribuita alla FFT nello sviluppo delle tecniche di elaborazione numerica.

Esistono varie versioni e livelli di ottimizzazione sul calcolo della DFT, ma l'idea più chiara e semplice è sicuramente quella basata sul Lemma di Danielson-Lanczos: "Una trasformata di Fourier discreta di lunghezza N può essere riscritta come la somma di due trasformate di Fourier discrete di lunghezza N/2, una formata dai termini di indice pari e una formata dai termini di indice dispari" :

$$\begin{aligned}
 H_n &= \sum_{j=0}^{N-1} e^{\frac{2\pi i j n}{N}} h_j = \sum_{j=0}^{N/2-1} e^{\frac{2\pi i (2j)n}{N}} h_{2j} + \sum_{j=0}^{N/2-1} e^{\frac{2\pi i (2j+1)n}{N}} h_{2j+1} = \\
 &= \sum_{j=0}^{N/2-1} e^{\frac{2\pi i 2j n}{N}} h_{2j} + W_N^n \sum_{j=0}^{N/2-1} e^{\frac{2\pi i j n}{N/2}} h_{2j+1} = F_n^p + W_N^n F_n^d
 \end{aligned}$$

Dove con p si indicano i termini "pari" e con d i termini "dispari".

Il risultato appena ottenuto non lascia emergere la sua reale potenza che invece si fa notare subito se consideriamo che possiamo iterare il procedimento appena effettuato per i primi due termini. Notiamo che i due termini ottenuti sono DFT di lunghezza N/2; se iteriamo il processo possiamo arrivare ad ottenere un numero N di termini che rappresentano delle DFT di lunghezza unitaria: questo se ipotizzato "N" come potenza di due. A questo punto è facile capire che il calcolo si velocizza tantissimo poiché la somma diventa il calcolo delle semplici radici ennesime.

A questo punto bisogna capire cosa sia una DFT di lunghezza unitaria: semplicemente un "copy and paste". Si ottiene una combinazione lineare degli input h_j pesati dai numeri complessi W_N . Vediamo di fare un esempio esplicativo con l'intento di far trasparire il procedimento generale per N molto grandi.

Per esprimere in maniera più compatta il kernel, si scrive: $w_N = e^{j2\frac{\pi}{N}}$.

Riferendosi all'equazione precedente, si rappresenta per N=8.

$$\begin{aligned}
H_n &= \sum_{j=0}^{8-1} e^{\frac{2\pi i j n}{8}} h_j = \sum_{j=0}^{4-1} e^{\frac{2\pi i (2j)n}{8}} h_{2j} + \sum_{j=0}^{4-1} e^{\frac{2\pi i (2j+1)n}{8}} h_{2j+1} = \\
&= \sum_{j=0}^{4-1} e^{\frac{2\pi i j n}{4}} h_{2j} + W_8^n \sum_{j=0}^{4-1} e^{\frac{2\pi i j n}{4}} h_{2j+1}
\end{aligned}$$

Si dividono i due termini ulteriormente, in termini pari-pari pp, pari-dispari pd, e combinazioni: dp e dd.

$$\begin{aligned}
H_n &= \sum_{j=0}^{2-1} e^{\frac{2\pi i (2j)n}{4}} h_{4j} + \sum_{j=0}^{2-1} e^{\frac{2\pi i (2j+1)n}{4}} h_{4j+2} + W_8^n \sum_{j=0}^{2-1} e^{\frac{2\pi i (2j)n}{4}} h_{4j+1} + \\
&+ W_8^n \sum_{j=0}^{2-1} e^{\frac{2\pi i (2j+1)n}{4}} h_{4j+3} = \sum_{j=0}^{2-1} e^{\frac{2\pi i j n}{2}} h_{4j} + W_4^n \sum_{j=0}^{2-1} e^{\frac{2\pi i j n}{2}} h_{4j+2} + \\
&+ W_8^n \sum_{j=0}^{2-1} e^{\frac{2\pi i j n}{2}} h_{4j+1} + W_8^n W_4^n \sum_{j=0}^{2-1} e^{\frac{2\pi i j n}{2}} h_{4j+3}
\end{aligned}$$

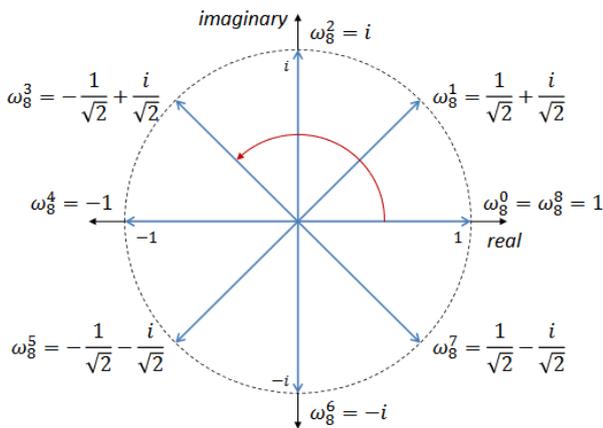
Ancora:

$$\begin{aligned}
H_n &= \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j)n}{2}} h_{8j} + \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j+1)n}{2}} h_{8j+4} + W_4^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j)n}{2}} h_{8j+2} + \\
&+ W_4^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j+1)n}{2}} h_{8j+6} + W_8^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j)n}{2}} h_{8j+1} + W_8^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j+1)n}{2}} h_{8j+5} + \\
&+ W_8^n W_4^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j)n}{2}} h_{8j+3} + W_8^n W_4^n \sum_{j=0}^{1-1} e^{\frac{2\pi i (2j+1)n}{2}} h_{8j+7} = \\
&= h_0 + W_2^n h_4 + W_4^n h_2 + W_4^n W_2^n h_6 + W_8^n h_1 + W_8^n W_2^n h_5 + W_8^n W_4^n h_3 + W_8^n W_4^n W_2^n h_7
\end{aligned}$$

Il risultato appena ottenuto è assolutamente significativo. Evidenziando i termini più importanti, si riporta in una tabella la logica di ordinamento e codifica dei termini.

Termine numero	che in binario è	combinazione di pari e dispari
0	000	000 (ppp)
1	001	100 (dpp)
2	010	010 (pdp)
3	011	110 (ddp)
4	100	001 (ppd)
5	101	101 (dpd)
6	110	011 (pdd)
7	111	111 (ddd)

Tale rappresentazione scaturisce dall'aver notato che la natura dei calcoli ha implicitamente inserito un algoritmo di reverse bit. Infatti associando lo stato logico "0" ad un termine "pari" e lo stato logico "1" ad un termine "dispari" abbiamo che, in ordine per come sono stati trovati, i termini h sono quanto riportato in tabella. Il reverse bit si manifesta nel fatto che le combinazioni della seconda e terza colonna sono le medesime combinazioni di bit ma invertite specularmente. Analizziamo adesso i coefficienti W_N^n ; questi si capisce bene come siano periodici in n. Inoltre è possibile portarli tutti in forma unica scrivendoli come W_8^{jn} sfruttando la proprietà che, sostanzialmente, si basa sul riportare, pensando ad $N=16,24,\dots$, ad una rappresentazione del genere:



Curiosità:

Mentre un segnale analogico necessita di componenti a tutte le frequenze sull'asse reale da $-\infty$ a $+\infty$, cioè in un ambito illimitato, per esprimere una sequenza in ambito frequenziale sono sufficienti le sole componenti frequenze comprese nell'intervallo limitato $[-1/2T, 1/2T]$. In un certo modo, questo risultato è pienamente giustificato dalla periodicità della trasformata di una sequenza, nel senso che le sole componenti veramente significative sono quelle nel periodo base. A sua volta, questo discorso si può proiettare sulla definizione pura della trasformata della sequenza numerica: è il kernel stesso, ossia le radici n-esime dell'unità immaginaria, ad indurre una

Figura 46: soluzioni dell'equazione $z^8=1$

-----> Periodicizzazione!!!!!!!

e calcolando gli otto valori della somma ottenuta sopra, calcolati per $n=1$, a titolo d'esempio:

1	1
$W_2^n = W_{8/4}^n$	$W_8^{4n} = -1$
$W_4^n = W_{8/2}^n$	$W_8^{2n} = i$
$W_4^n W_2^n = W_{8/2}^n W_{8/4}^n$	$W_8^{6n} = -i$
$W_8^n = W_8^n$	$W_8^n = e^{i\pi/4}$
$W_8^n W_2^n = W_8^n W_{8/4}^n$	$W_8^{5n} = -e^{i\pi/4}$
$W_8^n W_4^n = W_8^n W_{8/2}^n$	$W_8^{3n} = e^{i3\pi/4}$
$W_8^n W_4^n W_2^n = W_8^n W_{8/2}^n W_{8/4}^n$	$W_8^{7n} = -e^{i3\pi/4}$

Il calcolo di questi coefficienti richiede in genere un numero di operazioni trascurabili rispetto al calcolo principale dei singoli termini della FFT, che abbiamo detto essere $N \log_2 N$, anche perché è facile disporre di una matrice, considerando i successivi valori con $n > 1$, che memorizza tutto.

Si introduce una rappresentazione grafica, detta butterfly diagrams:



Figura 47: butterfly

La freccia verso l'alto indica una somma e la freccia verso il basso indica una sottrazione; il fattore W_n^j indica che prima di fare l'operazione di somma o sottrazione la variabile b viene moltiplicata per tale valore. Quindi gli output saranno la somma (in alto) e la differenza (in basso) tra a e $b \cdot W_n^j$.

Vale la pena verificare che l'algoritmo di FFT, secondo Cooley-Tukey, può essere rappresentato complessivamente dal seguente butterfly diagram per $N = 8$:

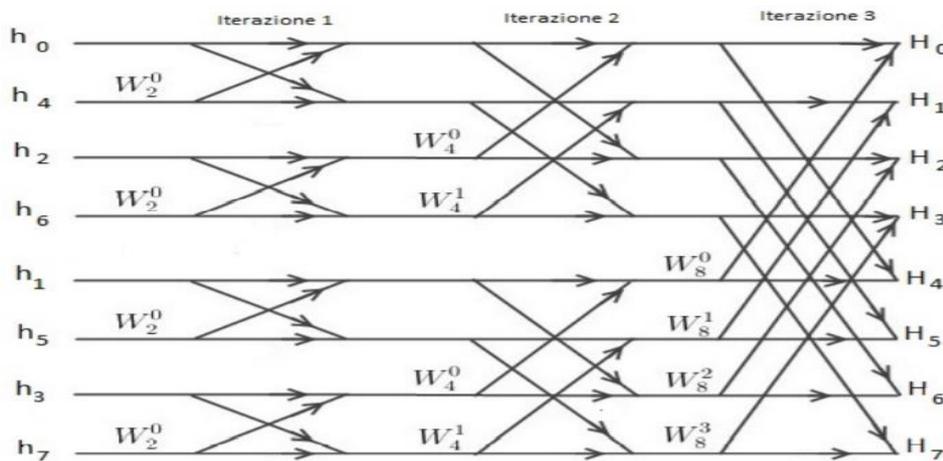


Figura 48: butterfly per N=8

Questa immagine aiuta a capire infine il numero effettivo di calcoli da effettuare. Notiamo che abbiamo 3 iterazioni ognuna composta da 8 somme. Non stupisce constatare che il numero di calcoli da effettuare effettivamente è $N \log_2(N) = 8 \log_2(8) = 24$, più un numero di operazioni (in genere irrilevante per N grandi) per calcolare i valori del kernel al variare dell'indice di somma.

Effettivamente con un numero che non è una potenza di 2 non è possibile trovare un numero di iterazioni pari a $\log_2(N)$ e quindi non varrebbe più il metodo sviluppato... Una soluzione pratica, nel caso di un numero di samples che non è potenza di 2, è quella di riempire il nostro pattern di dati con tanti termini nulli fino alla potenza di 2 successiva: ancora una volta si riscontra quanto sia utile la tecnica di zero-padding. Questa soluzione non introduce nessuna perturbazione al segnale perché ovviamente i termini nulli inseriti non contribuiscono in nessun modo al risultato finale della FFT: era già stato visto nella sezione sull'analisi spettrale. E' questo, d'altronde, l'unico limite di applicazione dell'algoritmo.

Ripercorrendo, per il lettore, i passi svolti su $N=2,4$, si mettono in evidenza le fasi essenziali del calcolo.

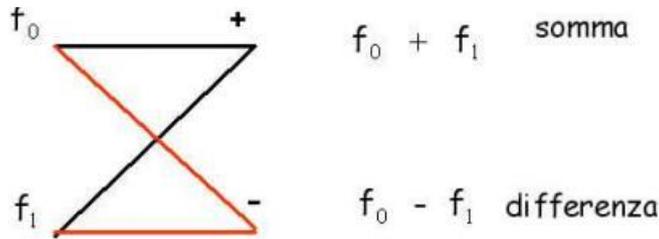
Caso per $N=2$. Si vuole calcolare la DFT del vettore $f=(f_0, f_1)$.

$$F_k = f_0 e^{j2\frac{0\pi k}{N}} + f_1 e^{j2\frac{1\pi k}{N}}, k=0,1$$

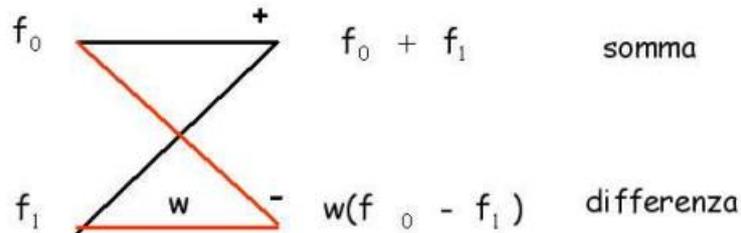
$$k=0 \rightarrow F_0=f_0+f_1$$

$$k=1 \rightarrow F_0=f_0-f_1$$

Usando la notazione butterfly, si ottiene:



A causa dell'indice troppo basso, non è possibile generalizzare con l'esponenziale complesso, poiché sempre unitario, dunque si scrive direttamente (il che si vede meglio aumentando N) :



Considerando il caso N=4, si ottiene:

$$F_0 = f_0 + f_1 + f_2 + f_3$$

$$F_1 = f_0 - if_1 - f_2 + if_3$$

$$F_2 = f_0 - f_1 + f_2 - f_3$$

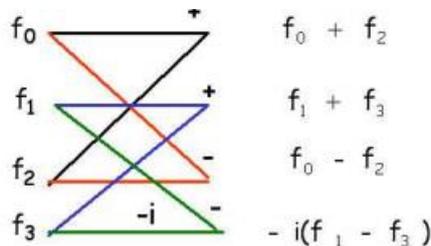
$$F_3 = f_0 + if_1 - f_2 - if_3$$

E' possibile riarrangiare i termini e notare un fatto molto importante.

$$F_0 = (f_0 + f_2) + (f_1 + f_3) \quad F_1 = (f_0 - f_2) - i(f_1 - f_3)$$

$$F_2 = (f_0 + f_2) - (f_1 + f_3) \quad F_3 = (f_0 - f_2) + i(f_1 - f_3)$$

Allora è proprio vero: il calcolo può essere ridotto perché si possono utilizzare dei risultati parziali che sono presenti, in questo caso, in due DFT di indice entrambe pari o dispari.



La tesi è chiara....

Esistono algoritmi diversi, che esulano dallo scopo di questa Tesi, per ogni numero di samples possibile, anche per numeri primi. Ovviamente per N molto grandi è necessario trovare una soluzione computazionale in un linguaggio di programmazione che permetta di fare, molto velocemente, il calcolo di una FT tramite l'algoritmo della FFT, dato il suo enorme risultato nel diminuire il numero di operazioni da svolgere effettivamente.

Una possibile implementazione in c++, a titolo puramente illustrativo, è la seguente.

```

* File: main.cpp
* Author: Nicola Lombardi
*
* Created on 3 ottobre 2020, 14.17
*/

#include <iostream>
#include <complex>
#define MAX 200
#define M_PI 3.14159265358979323846
using namespace std;

int log2(int N) //funzione per calcolare il logaritmo in base 2 di un intero
{
    int k = N, i = 0;
    while(k) {
        k >>= 1;
        i++;
    }
    return i - 1;
}

int check(int n) //usato per controllare se il numero di componenti del vettore di input è una potenza di 2
{
    return n > 0 && (n & (n - 1)) == 0;
}

int reverse(int N, int n) //calcola il reverse number di ogni intero n rispetto al numero massimo N
{
    int j, p = 0;
    for(j = 1; j <= log2(N); j++) {
        if(n & (1 << (log2(N) - j)))
            p |= 1 << (j - 1);
    }
    return p;
}

void ordina(complex<double>* f1, int N) //dispone gli elementi del vettore ordinandoli per reverse order
{
    complex<double> f2[MAX];
    for(int i = 0; i < N; i++)
        f2[i] = f1[reverse(N, i)];
    for(int j = 0; j < N; j++)
        f1[j] = f2[j];
}

void transform(complex<double>* f, int N) //calcola il vettore trasformato
{
    ordina(f, N); //dapprima lo ordina col reverse order
    complex<double> W[N / 2]; //vettore degli zeri dell'unità.
    //Prima N/2-1 ma genera errore con ciclo for successivo
    //in quanto prova a copiare in una zona non allocata "W[N/2-1]"
    W[1] = polar(1., -2. * M_PI / N);
    W[0] = 1;
    for(int i = 2; i < N / 2; i++)
        W[i] = pow(W[1], i);
    int n = 1;
    int a = N / 2;
    for(int j = 0; j < log2(N); j++) {
        for(int i = 0; i < N; i++) {
            if(!(i & n)) {
                /*ad ogni step di raddoppiamento di n, vengono utilizzati gli indici */
                /*i' presi alternativamente a gruppetti di n, una volta si e una no.*/
                complex<double> temp = f[i];
                complex<double> Temp = W[(i * a) % (n * a)] * f[i + n];
                f[i] = temp + Temp;
                f[i + n] = temp - Temp;
            }
        }
        n *= 2;
        a = a / 2;
    }
}

```

```

void FFT(complex<double>* f, int N, double d)
{
    transform(f, N);
    for(int i = 0; i < N; i++)
        f[i] *= d; //moltiplica il vettore per il passo in modo da avere il vettore trasformato effettivo
}

int main()
{
    int n;

    cout << "specifica la dimensione del vettore,che sia potenza di 2" << endl;
    cin >> n;

    double d;
    cout << "inserisci la dimensione del passo di campionamento" << endl;
    cin >> d;
    cout << "passo di campionamento = " << d << endl;
    complex<double> vec[MAX];
    cout << "inserisci il vettore di campionamento" << endl;
    for(int i = 0; i < n; i++) {
        cout << "inserisci la componente di indice " << i << endl;
        cin >> vec[i];
        cout << "indice " << i << " = " << vec[i] << endl;
    }
    FFT(vec, n, d);
    cout << "vettore trasformato" << endl;
    for(int j = 0; j < n; j++)
        cout << vec[j] << endl;
    return 0;
}

```

Il cui output:

```

specifica la dimensione del vettore,che sia potenza di 2
4
inserisci la dimensione del passo di campionamento
1
passo di campionamento = 1
inserisci il vettore di campionamento
inserisci la componente di indice 0
3
indice 0= (3,0)
inserisci la componente di indice 1
4
indice 1= (4,0)
inserisci la componente di indice 2
5
indice 2= (5,0)
inserisci la componente di indice 3
6
indice 3= (6,0)
vettore trasformato
(18,0)
(-2,2)
(-2,0)
(-2,-2)

```

Viene testato l'algoritmo di Cooley Tukey tramite uno script c++. La verifica può essere effettuata con un codice Matlab, puramente personalizzabile. Si è scelto di implementare la definizione stessa di DFT. Usando la funzione FFT e, successivamente, la IFFT di Matlab: si perviene alle stesse conclusioni.

Si considera la stessa sequenza in input allo script c++, ossia $x[n]=[3 \ 4 \ 5 \ 6]$, poiché di facile verifica, se si vuole, anche analitica.

```
x =
    3    4    5    6

l =
    4

array initialized with zero and length l

xk =
    0    0    0    0

displaying final transform
 18.0000 + 0.0000i -2.0000 + 2.0000i -2.0000 - 0.0000i -2.0000 - 2.0000i
```

```
clc
clear all
close all
x=[3 4 5 6]
l=length(x)
disp('array initialized with zero and length l')
xk=zeros(1,l)
for k=0:l-1
    for n=0:l-1
        xk(k+1)=xk(k+1)+x(n+1)*exp((-i)*2*pi*k*n/l);
    end
end
```

4.6.2 FFT: moltiplicazione di polinomi

Dati due polinomio $A(x)$ e $B(x)$, trova il prodotto $C(x) = A(x) * B(x)$. Esiste l'approccio "ingenuo quadratico" per risolvere questo problema (che tuttavia ha senso negli esercizi classici e didattici e non di certo per digital filtering).

Si considerino:

$$A(x) = 6x^3 + 7x^2 - 10x + 9$$

$$B(x) = -2x^3 + 4x - 5$$

rappresentazione del coefficiente di $A(x) = (9, -10, 7, 6)$

Rappresentazione del coefficiente di $B(x) = (-5, 4, 0, -2)$

Input :

$$A [] = \{9, -10, 7, 6\}$$

$$B [] = \{-5, 4, 0, -2\}$$

Output : $C [] = \{-12, -14, 44, -20, -75, 86, -45\}$

Per quanto esposto sulla FFT: si può fare di meglio se rappresentiamo il polinomio in un'altra forma.

Dapprima si introduce un'altra convenzione di rappresentazione.

L'idea è rappresentare il polinomio in forma di valore puntuale e quindi calcolare il prodotto. Una rappresentazione in punti di un polinomio $A(x)$ di grado n è un insieme di n coppie di valori in punti :

$$\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$$

Esempio

$x_i = 0, 1, 2, 3$

$A(x_i) = 1, 0, 5, 22$

Dunque tali che tutti gli x_i siano distinti e $y_i = A(x_i)$ per $i = 0, 1, \dots, n-1$.

La rappresentazione del valore del polinomio in determinati punti è propria del metodo di Horner

La valutazione in n punti richiede un costo asintotico rappresentabile come $O(n^2)$: quindi la complessità computazionale è quadratica e forse la DFT può smussare questo asintoto.

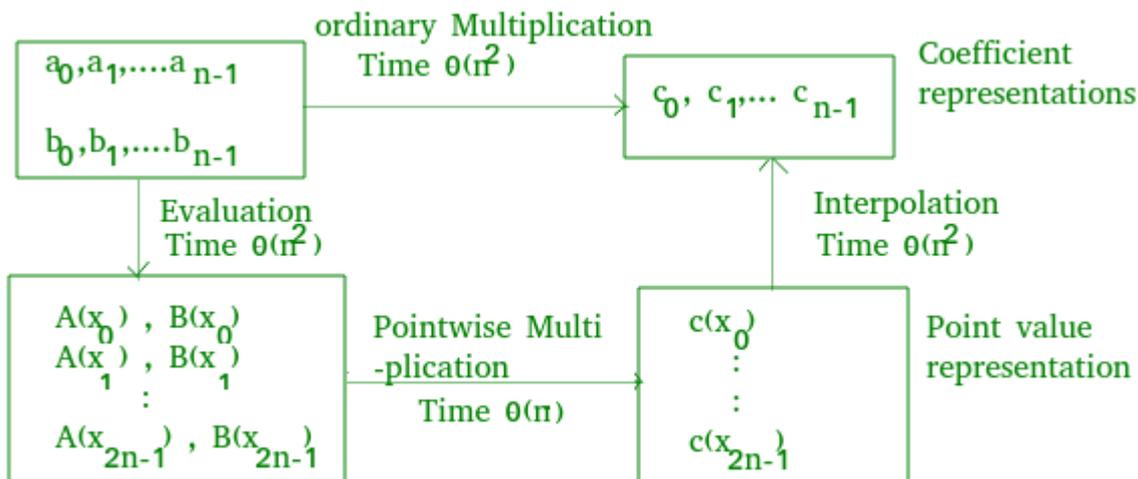
Riprendendo l'esempio, ora che il polinomio è convertito coppie di punti, può essere facilmente calcolato $C(x) = A(x) * B(x)$ usando di nuovo il metodo di Horner. Si svolge il prodotto "pointwise" tra i due vettori e se ne ottiene un altro.

Un'osservazione cruciale è che evita di far cadere nell'ingenua e frettolosa conclusione di aver finito, è proprio un fatto che sembra banale. Un fatto importante qui è che $C(x)$ ha un limite di grado dettato dalla natura dell'operazione di moltiplicazione: $2n$. Quindi n punti daranno solo n punti di $C(x)$, allora, in questo caso, si necessita di $2n$ diversi valori di x per calcolare $2n$ diversi valori di y .

Per tornare alla forma del vettore dei coefficienti è naturale dover ragionare con l'inverso di questa valutazione fatta finora: chiamata interpolazione. Usando la formula di Lagrange:

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)}$$

Lo schema logico dell'algoritmo.



Figura_48:schema

Questa idea risolve ancora il problema in una complessità computazionale quadratica. Solo scegliendo attentamente i punti di valutazione, si può scendere ad un costo minimo di $O(n \log n)$. E' da fare una scelta accurata e dunque l'algoritmo dovrebbe essere troppo intelligente. Esiste una soluzione più meccanica e che da la garanzia dello stesso costo allo stesso problema.

Se si scelgono le "radici complesse dell'unità immaginaria" come punti di valutazione sul piano complesso, si può produrre una rappresentazione in punti prendendo la trasformata discreta di Fourier (DFT) di un vettore di coefficienti. Insomma, dove il problema è il calcolo di un data-set che dipende da un input di dati, la DFT è pronta ad aiutare.

Come già mostrato, di fatto, è possibile eseguire l'operazione inversa, l'interpolazione, prendendo la "DFT inversa" di coppie di valori punti, ottenendo un vettore di coefficiente. La trasformata di Fourier veloce (FFT) può eseguire DFT e DFT inversa nel tempo $O(n \log n)$.

Algoritmo

1. Aggiungi n coefficienti zero di ordine superiore ad $A(x)$ e $B(x)$
2. Valuta $A(x)$ e $B(x)$ usando FFT per $2n$ punti
3. Moltiplicazione puntuale delle forme di valore in punti
4. Interpolare $C(x)$ utilizzando FFT per calcolare DFT inverso

Si illustra uno pseudo codice di FFT ricorsiva

```

Recursive_FFT(a) {
    n = length(a) // a is the input coefficient vector
    if n = 1
        then return a

    // wn is principle complex nth root of unity.
    wn = e^(2*pi*i/n)
    w = 1

    // even indexed coefficients
    A0 = (a0, a2, ..., an-2 )

    // odd indexed coefficients
    A1 = (a1, a3, ..., an-1 )

    y0 = Recursive_FFT(A0) // local array
    y1 = Recursive_FFT(A1) // local array

    for k = 0 to n/2 - 1

        // y array stores values of the DFT
        // of given polynomial.
        do y[k] = y0[k] + w*y1[k]
           y[k+(n/2)] = y0[k] - w*y1[k]
           w = w*wn
    return y
}
    
```

L'albero binario e completo che rappresenta questa lista di operazioni, si può rappresentare secondo quanto segue.

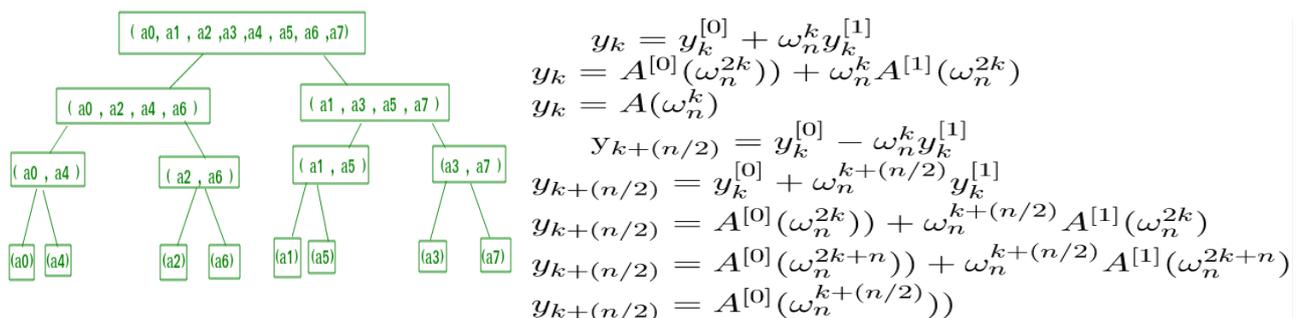


Figura 49: schema Divide et Impera e relative successioni

In parole povere, si ha il seguente algoritmo, dal punto di vista matematico:

$$p(x) = 1 + 5x + 17x^2, \quad q(x) = 11 + 6x - 4x^2$$

$$a = (1, 5, 17, 0, 0)$$

$$b = (11, 6, -4, 0, 0)$$

$$A = DFT[a] = (23, -11.2082 - 14.7476i, 2.082 + 13.229i, 2.2082 - 13.229i, -11.2082 + 14.7476i)$$

$$B = DFT[b] = (13, 16.0902 - 3.3552i, 4.9098 - 7.3309i, 4.9098 + 7.3309i, +16.0902 + 3.3552i)$$

$$C = DFT(c) = DFT[a * b] = A * B$$

$$\text{Ossia } C = [229, -229.82-199.69i, 108.82+48.76i, 107.82-48.76i, -229.82+199.69i]$$

$$c = IDFT(C) = (11, 6, -4, 0, 0)$$

A titolo solo illustrativo, è presente in rete il codice di geeksforgeeks.org che implementa tali operazioni.

```
#include <bits/stdc++.h>
using namespace std;

// For storing complex values of nth roots
// of unity we use complex<double>
typedef complex<double> cd;

// Recursive function of FFT
vector<cd> fft(vector<cd>& a)
{
    int n = a.size();

    // if input contains just one element
    if (n == 1)
        return vector<cd>(1, a[0]);

    // For storing n complex nth roots of unity
    vector<cd> w(n);
    for (int i = 0; i < n; i++) {
        double alpha = 2 * M_PI * i / n;
        w[i] = cd(cos(alpha), sin(alpha));
    }

    vector<cd> A0(n / 2), A1(n / 2);
    for (int i = 0; i < n / 2; i++) {

        // even indexed coefficients
        A0[i] = a[i * 2];

        // odd indexed coefficients
        A1[i] = a[i * 2 + 1];
    }

    // Recursive call for even indexed coefficients
    vector<cd> y0 = fft(A0);
```

```

// Recursive call for odd indexed coefficients
vector<cd> y1 = fft(A1);

// for storing values of y0, y1, y2, ..., yn-1.
vector<cd> y(n);

for (int k = 0; k < n / 2; k++) {
    y[k] = y0[k] + w[k] * y1[k];
    y[k + n / 2] = y0[k] - w[k] * y1[k];
}
return y;
}

// Driver code
int main()
{
    vector<cd> a{1, 2, 3, 4};
    vector<cd> b = fft(a);
    for (int i = 0; i < 4; i++)
        cout << b[i] << endl;

    return 0;
}

```

Un esempio di esecuzione del programma:

```

Input:  1 2 3 4
Output:
(10, 0)
(-2, -2)
(-2, 0)
(-2, 2)

```

4.7 Considerazioni sulle prove sperimentali

La vita non è bella come sembra: c'è il rumore.

In definitiva, dunque, *il rispetto del teorema del campionamento consente di evitare l'errore di aliasing*. In realtà, questo è vero solo dal punto di vista teorico, mentre invece sorgono diverse difficoltà quando si deve operare concretamente: il problema principale è che non si può campionare infinitamente un dato segnale, ma bisogna limitarsi a farlo in un intervallo di tempo comunque finito, che è stata definita come “finestra di osservazione” nel capitolo sul passaggio dal continuo al discreto. Questa finestra di osservazione comporta dunque una limitazione temporale del segnale analogico da analizzare, il che corrisponde ad un segnale teoricamente con spettro infinito e quindi inevitabilmente soggetto alla sovrapposizione delle repliche traslate. In altri termini, *l'errore di aliasing è teoricamente sempre presente*.

L'unico modo per evitare la sovrapposizione delle repliche traslate dello spettro del segnale è quello di limitarlo in banda prima di campionarlo: bisogna cioè utilizzare opportuni filtri (detti perciò filtri anti-aliasing), i quali eliminano, dal segnale analogico di partenza, un certo numero di frequenze, in modo da permettere la successiva corretta discretizzazione. Quali frequenze, allora, vanno eliminate? Per limitare al minimo le distorsioni del segnale filtrato rispetto al segnale di partenza, è necessario filtrare solo le componenti a contenuto energetico minore, ossia quelle che contribuiscono in maniera minima al contenuto informativo del segnale: c'è stato un esempio, nella sezione dell'analisi spettrale di potenza, che ha dettato la dimostrazione sperimentale.

Dal punto di vista dell'analisi: il progettista cosciente di avere a che fare con coefficienti di Fourier (e

quindi armoniche) che hanno una velocità di decrescita, può effettuare la sintesi con un numero finito di armoniche e dunque dimensionare la frequenza di carico del filtro anti-aliasing.

In generale, il criterio più sensato è quello di ritenere trascurabili quelle componenti di ampiezza confrontabile con le componenti rumore e che quindi da tale rumore sarebbero comunque indistinguibili.

Ancora, l'inevitabile presenza del rumore è un altro motivo fondamentale per cui non si può prescindere dal pre-filtraggio del segnale $s(t)$ da campionare: abbiamo detto che lo spettro del segnale campionato è una ripetizione periodica, a passo f_c , dello spettro del segnale $s(t)$; il campionatore, però, riceve in ingresso sia $s(t)$ sia il rumore sovrapposto, che, in generale, è un rumore bianco, cioè con spettro di potenza costante su tutte le frequenze; allora, se campionassimo l'ingresso senza pre-filtrare, noi ripeteremmo, a passo f_c , sia lo spettro di $s(t)$ sia lo spettro del rumore, ma quest'ultimo è esteso su tutte le frequenze, per cui, in banda base come nel resto delle frequenze, avremmo la somma (in potenza) di infiniti contributi di rumore, il che sarebbe inaccettabile. Pre-filtrando, invece, noi otterremmo in banda base un unico contributo di rumore, che è quello relativo alla banda di $s(t)$ e dal quale non possiamo ovviamente liberarci.

Un esempio tipico su un segnale vocale.

L'orecchio umano può udire segnali costituiti da componenti frequenziali comprese al più nella banda tra 20 Hz e 20 kHz. Suoni con frequenza inferiore ai 20 Hz vengono percepiti come "vibrazioni" con tutto il corpo, mentre suoni con frequenza maggiore di 20 kHz sono inudibili e vengono chiamati ultrasuoni. Concentrandosi su 20kHz, si considera un segnale audio, per esempio, concentrato nella banda [0,20kHz]. Adottando un Tape o un compact-disc come dispositivo di registrazione audio, non c'è problema, perché la frequenza di campionamento è maggiore del doppio della banda del segnale.

Quando il segnale audio campionato deve essere trasmesso, come in un sistema di radiodiffusione, è importante cercare di ridurre al minimo il numero di campioni, cioè la frequenza di campionamento. Infatti, più campioni devono essere trasmessi nello stesso intervallo di tempo, maggiore deve essere la capacità (e quindi il costo) del sistema di trasmissione, o del sistema di acquisizione dati.

Per questo motivo, in alcuni standard di radiodiffusione dell'audio digitale, si sceglie di campionare il segnale con $f_c=32\text{kHz}$. Questa frequenza non soddisfa la condizione di Nyquist rispetto alla banda $B=20\text{kHz}$. Allora, per evitare problemi di aliasing, si antepone al convertitore A/D un filtro anti-aliasing che limita convenientemente la banda del segnale analogico a un valore B' in modo da annullare l'aliasing per la frequenza di campionamento fissata. Nel caso dello standard con $f_c = 32\text{kHz}$, il filtro anti-aliasing ha una banda $B' = 15\text{kHz}$. È chiaro che in questo modo la qualità del segnale riprodotto sarà inferiore (dal punto di vista della DFT e dell'analisi spettrale è diminuita la risoluzione in frequenza) a quella dei sistemi CD e DAT per l'artificiale limitazione in banda, ma ancora sufficientemente elevata per una riproduzione godibile, e con una minore esigenza di capacità del sistema di trasmissione.

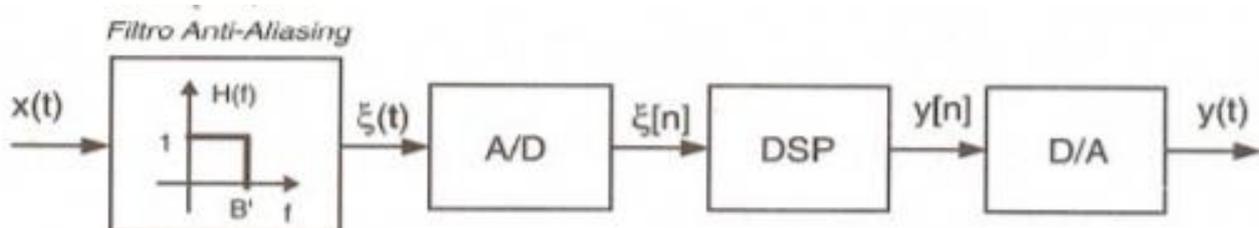


Figura 50: schema digitale

Come è prevedibile dalla trattazione esposta sull'interpolazione, il dispositivo di sample e hold introduce distorsione e dunque le frequenze immagini sono da eliminare a valle del D/A converter. Si pensi

all'operazione di hold: l'interpolazione a mantenimento introduce distorsione a causa delle discontinuità di prima specie nel dominio del tempo, all'interno del prodotto di convoluzione. L'effetto del filtro anti-immagine in ambito temporale è quello di smussare il segnale costante a tratti.

Il sistema completo:

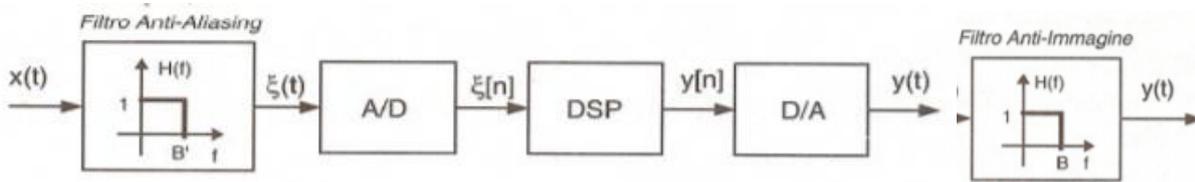


Figura 51: schema digitale

Conclusione

La conclusione più degna di nota è la grande potenza della DFT, la quale è riassunta in appena una decina di esperimenti diversi che sono stati presi in considerazione in questo lavoro di Tesi.

In questo modo il suo utilizzo si è espanso in ogni direzione, nei confronti dell'analisi dei segnali. Di certo non è l'unica branca in cui è usata. Oggi, infatti, l'algoritmo di fattorizzazione della FT è implementato in una miriade di applicazioni che vanno dallo scopo ludico (programmi di editing audio o foto, cd musicali ecc...), a quello industriale (come le telecomunicazioni) o medico (imaging digitale per risonanze magnetiche e tomografie assiali computerizzate). Esistono versioni modificate di tale algoritmo in grado di svolgere in maniera molto più efficiente casi più specifici come funzioni soltanto reali, o semplicemente funzioni note come sinusoidi o cosinusoidi. Ottimizzando in tal senso il codice (in base al tipo di funzione in ingresso appunto), si semplifica ancora di più il numero di calcoli da svolgere e soprattutto si rendono minimi i tempi impiegati, permettendo un rapidissimo responso in termini di frequenza sulla composizione di un segnale; questo che abbiamo descritto rappresenta, in ogni caso, il più generale degli algoritmi. Un ultimo commento va fatto riguardo la dimensionalità della FFT. È stato trattato il caso monodimensionale, ma il concetto è facilmente estendibile a più dimensioni; l'utilizzo multidimensionale più sfruttato è quello del caso bidimensionale poichè permette una FFT di intere "matrici" di bit o, forse con il loro termine più noto, pixels. Il candidato ha preferito non includere una parte così importante su cui è stato fatto uno studio molto più soft e meno approfondito rispetto a quello dell'analisi dei segnali.

APPENDICE 1

Richiami essenziali

L'analisi nel dominio della frequenza, per i sistemi lineari e stazionari, costituisce uno degli strumenti più importanti ed efficaci per lo studio di talune proprietà, quali ad esempio le **proprietà filtranti**. L'analisi in tale dominio si basa sulla particolare forma che l'uscita di un sistema lineare e stazionario stabile assume in risposta ad un segnale in ingresso di tipo **sinusoidale**. Tuttavia, in virtù del principio di sovrapposizione delle cause e degli effetti, tale tipo di analisi si può estendere a classi molto più ampie di segnali in ingresso, ossia a tutti quei segnali che possono essere rappresentati come una combinazione lineare, finita o infinita, di componenti sinusoidali.

A seconda del caso, avremo un formalismo relativo alla risposta all'ingresso sinusoidale che dipenderà proprio dalla classe del segnale.

Tornerà molto utile sfruttare quella che è nota come Formula di Eulero:

$$e^{\pm ix} = \cos(x) \pm i \sin(x),$$
$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2},$$
$$\sin(x) = \frac{e^{ix} - e^{-ix}}{2i}.$$

Infatti la **Trasformata di Steinmetz** è basata proprio su tale formalismo per convertire un segnale del tempo nel suo analogo nella frequenza e viceversa.

Sia $f(t): [-\pi/\omega, \pi/\omega]$ a valori reali, si definisce sua trasformata la funzione di variabile reale $F(\omega)$ data da:

$$S\{f(t)\}(\omega) = \frac{\omega}{\pi} \int_{-\frac{\pi}{\omega}}^{\frac{\pi}{\omega}} f(t) e^{-i\omega t} dt$$

Questa trasformata integrale trasforma le equazioni integrali e le equazioni differenziali in equazioni polinomiali, che sono più immediate da risolvere.

Anche la risposta (l'uscita) di un sistema dinamico lineare può essere calcolata come prodotto di convoluzione della sua risposta impulsiva unitaria con il segnale d'ingresso. Sviluppando questo calcolo la convoluzione diventa una moltiplicazione, che spesso rende il problema più semplice. La trasformata di Steinmetz può anche essere usata per risolvere le equazioni differenziali: se ne farà uso proprio in questo ambito.

Si prova che se una funzione $g(s)$ ha la trasformata inversa $f(t)$, ovvero è una funzione continua a tratti che soddisfa la condizione:

$$s\{f(t)\}(\omega) = g(\omega)$$

allora è univocamente determinata:

$$s^{-1}\{g\}(t) = \text{Re}(g(\omega)e^{i\omega t}).$$

La trasformata di Steinmetz è molto utile per esprimere in forma esponenziale i segnali sinusoidali, le risposte armoniche e le risposte a regime sinusoidale dei sistemi LTI, ovviamente stabili. E' strettamente legata alla trasformata di Fourier. A differenza di questa però il dominio di integrazione dipende dalla variabile di trasferimento

Ai fini della trattazione è necessario giustificare determinati termini rivolti alla Teoria dei Sistemi con una piccola introduzione ai sistemi **Single Input Single Output** e alla rappresentazione delle variabili di stato.

In generale, un SISO si ha quando l'ingresso, che causa la risposta forzata del sistema e la traiettoria forzata delle sue variabili di stato, è unico.

Indicati con:

$u(t) \in R^m$ il vettore delle variabili di ingresso;

$x(t) \in R^n$ il vettore delle variabili di stato;

$y(t) \in R^p$ il vettore delle variabili di uscita;

$f(t)$ e $g(t)$, due funzioni vettoriali, le cui proprietà classificano i sistemi;

un sistema dinamico a tempo continuo è definito dalle seguenti relazioni costitutive:

$$\begin{cases} \dot{x}(t) = f(x, u, t) \rightarrow \text{equazione di stato} \\ y(t) = g(x, u, t) \rightarrow \text{trasformazione di uscita} \\ x(t_0) = x_0 \rightarrow \text{stato iniziale del sistema} \end{cases}$$

Il numero n delle variabili di stato definisce l'ordine del sistema.

•L'equazione differenziale, nota come equazione di stato, pone l'ingresso $u(t)$ in relazione con le variabili che descrivono la situazione interna del sistema, dal punto di vista energetico, e definisce l'evoluzione dello stato $x(t)$ per $t > 0$, in corrispondenza dell'istante iniziale t_0 , della "condizione iniziale" $x(t_0) = x_0$ e della funzione di ingresso $u(t)$ definita per ogni $t \geq t_0$. La funzione $x(t)$ per $t \geq t_0$ si dice movimento dello stato del sistema e come per la risposta y , è possibile identificare la componente propria del sistema e quella indotta dal sistema sollecitante (l'ingresso o variabile forzante).

•L'equazione algebrica, detta trasformazione di uscita, consente di determinare l'uscita $y(t)$ a uno specifico istante di tempo sulla base della conoscenza della situazione interna del sistema e dell'ingresso allo stesso istante di tempo, ovvero permette di determinare l'evoluzione della uscita $y(t)$ per $t \geq t_0$, in corrispondenza della funzione $u(t)$ e dell'andamento dello stato $x(t)$ per $t \geq t_0$. La funzione $y(t)$ per $t \geq t_0$ è conosciuta col nome di movimento dell'uscita. L'equazione di stato e la trasformazione di uscita costituiscono una rappresentazione di stato, o rappresentazione [ingresso,stato,uscita], o rappresentazione interna di un sistema dinamico a tempo continuo. Si parla anche di rappresentazione nello spazio degli stati. E' importante sottolineare sempre la parola "continuo" perché le equazioni differenziali diventano equazioni alle differenze finite nel caso di un sistema ad eventi a tempo discreto (detto anche SED).

APPENDICE 2

RICHIAMI SULLA RISPOSTA IN FREQUENZA

2.1 Generalità

Come già definiti, i sistemi Single Input Single Output (i sistemi SISO), lineari, stazionari e a parametri concentrati sono descritti da un modello ingresso-uscita: tale modello consiste in una equazione differenziale ordinaria e lineare a coefficienti costanti. Dunque vi è un solo ingresso sollecitante, per volta, e dunque il gruppo di frequenze libere del sistema considerato che assume la dimensione in base alla complessità del sistema (banalmente torniamo all'ordine del sistema, ma vi sono anche dei modi che possono essere introdotti dall'ingresso sollecitante, che comunque non sono radici del Polinomio in s , variabile di Laplace). In realtà possiamo dedurre l'ordine in base al numero di componenti in grado di immagazzinare energia. Si pensi ad un circuito RLC nelle sue versioni, serie e parallelo. Peccato esistano le cosiddette situazioni patologiche. Esiste però un Teorema che può aiutare in questa valutazione e che viene enunciato perché sarà sulla base dell'analisi circuitale.

“Quando la rete elettrica non contiene i casi in cui si abbiano condizioni patologiche, ossia percorsi chiusi (maglie) di soli generatori di tensione e condensatori, o co-cicli di soli generatori di corrente e induttori, allora le variabili di stato sono meno discontinue dell'ingresso. Quando si applica un ingresso che ha nell'istante iniziale una discontinuità di prima specie, le variabili di stato si conservano, cioè non cambiano tra 0^- e 0^+ : sono funzioni continue”.

2.2 Risposta a regime ad un ingresso sinusoidale

Si consideri un sistema SISO lineare e stazionario avente funzione di trasferimento trasformata con Laplace $G(s)$ con poli tutti a parte reale negativa, ossia asintoticamente stabile. Si supponga che tale sistema venga eccitato a partire dall'istante di tempo $t = 0$ da un segnale di tipo sinusoidale avente pulsazione $\omega \in [0, +\infty)$ e ampiezza U : $u(t) = U \sin(\omega t)$. Siano $G(\omega)$ e $\varphi(\omega)$ rispettivamente, il modulo e la fase della funzione di trasferimento $G(s)$ valutati in $s = j\omega$ (modulo e fase di un numero complesso del Piano di Gauss). Ci stiamo spostando sull'asse immaginario per valutare le frequenze fisiche del sistema. In condizioni di regime la risposta di tale sistema $y(t)$ è anch'essa un segnale di tipo sinusoidale avente la stessa pulsazione del segnale in ingresso, modulo pari al prodotto “ $|G(j\omega)|U$ ” e il cui sfasamento rispetto al segnale in ingresso è pari a $\varphi(\omega)$, ossia :

$$y_r(t) = G(\omega)U \sin(\omega t + \varphi(\omega)).$$

Si omette la dimostrazione perché di scarso rilievo.

Il significato fisico attribuito alla $G(j\omega)$ nel caso dei sistemi con poli a parte reale negativa suggerisce anche un metodo per la sua determinazione sperimentale. In questo caso infatti è sufficiente applicare in ingresso al sistema un segnale sinusoidale, aspettare che l'uscita vada a regime, e quindi determinare il rapporto tra l'ampiezza del segnale in uscita e quella del segnale in ingresso nonché lo sfasamento tra i due. Ripetendo

questa operazione con diversi segnali sinusoidali in ingresso, caratterizzati da diversi valori della pulsazione, si risale all'andamento del modulo e della fase della risposta armonica nel campo delle pulsazioni di interesse.

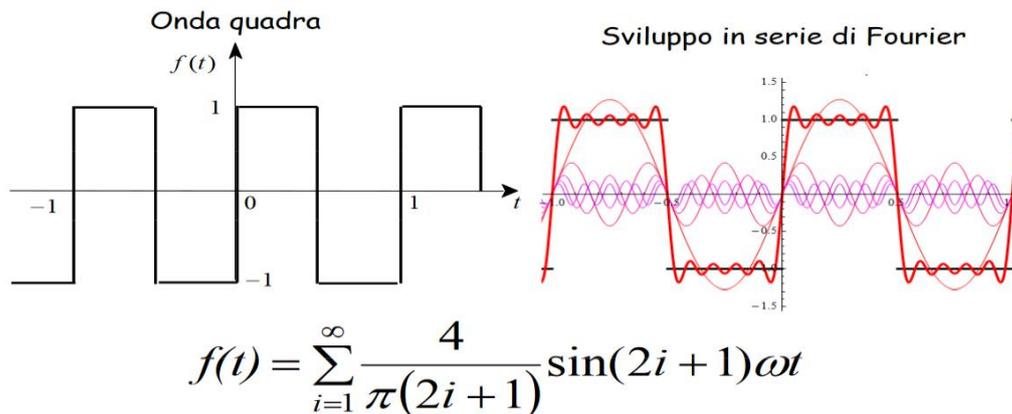
Avendo, dunque, sollecitato il sistema con diverse pulsazioni, tale metodo suggerisce una possibile trattazione del concetto di risposta Armonica anche con classi di segnali rappresentabili in Serie di Fourier e tramite la Trasformata di Fourier.

Ricordando che un qualunque segnale periodico è sempre rappresentabile come:

$$u(t) = S(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{L} + b_n \sin \frac{n\pi t}{L} \right)$$

con S(t) Serie di Fourier di $u(t)=u(t-T)$ e T periodo del segnale periodico u.

Un esempio principe per l'Elettronica Industriale di Potenza e Automazione Industriale è proprio l'onda quadra, ad esempio per un full-bridge per la conversione dell'energia elettrica nella direzione dettata dal flusso di potenza, a seconda della situazione, frenata o accelerazione.



Essendo una funzione dispari sono nulle la componente costante (il valor medio) e tutti i coefficienti dei termini in coseno.

La trattazione non si occuperà dei criteri di convergenza perché la serie è illustrata solo a titolo informativo.

APPENDICE 3

Cenni essenziali sui sistemi LTI e sistemi lineari

3.1 Introduzione al problema nel dominio della frequenza

Sapendo che un sistema, sempre e comunque **dinamico lineare stazionario**, anche detto **sistema lineare tempo-invariante** o **sistema LTI**, è un sistema dinamico lineare tempo-invariante, soggetto cioè al principio di sovrapposizione degli effetti: per esso vale il principio di traslazione causa-effetto nel tempo, cioè **il sistema risponde sempre con lo stesso effetto ad una data causa, a prescindere dall'istante di tempo in cui tale causa agisca sul sistema**. Allora possiamo pensare che sollecitandolo tramite un ingresso sinusoidale, ogni volta ad una frequenza differente, sarà possibile ottenere diversi punti del diagramma di Bode della sua Risposta Armonica.

Tramite alcune basilari nozioni di Teoria dei Sistemi LTI è possibile pensare di progettare un Trasduttore che restituisce l'ordine del sistema che risponde (input al trasduttore) : è stato uno dei lavori del candidato per il corso di Algebra Lineare numerica, ossia il corso di Algoritmi Numerici (in versione seminario da due cfu) del docente Giuseppe Rodriguez, nonché relatore di tesi.

La trasduzione di una grandezza fisica, in generale, consiste nell'acquisizione di un INPUT con una certa unità di misura e la restituzione in OUTPUT della stessa, ma convertita tramite l'analisi dimensionale regalata dal Sistema Internazionale, in un'altra unità di riferimento. Tramite la determinazione dell'ordine del sistema che viene preso in esame e risponde, verrà costruito un sistema lineare e verrà osservato il problema essere mal posto. Il caso *sovradeterminato* si verifica quando il sistema è stato sovravincolato - cioè quando le equazioni superano in numero le incognite. Al contrario, il caso *sottodeterminato* si verifica quando il sistema è stato sottovincolato - cioè, quando il numero di equazioni è inferiore al numero di incognite.

3.2 Sistemi lineari

Di seguito un breve background sui sistemi lineari e il rango della matrice dei coefficienti, analizzandolo anche dal punto di vista della Teoria dei Sistemi.

Sia $Ax=b$ un sistema lineare che in forma matriciale risulti:

$$\begin{pmatrix} b_1 \\ \dots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$$

In particolare:

- la matrice dei coefficienti del sistema o matrice incompleta: $A \in \mathbb{R}^{m \times n}$
- Il vettore delle soluzioni: $x \in \mathbb{R}^n$
- Il vettore dei termini noti: $b \in \mathbb{R}^m$
- $B=[A|b]$ la matrice A aumentata con il vettore dei termini noti: $B \in \mathbb{R}^{m \times (n+1)}$

Si affrontano i diversi casi.

- (i) Sia $Ax=b$ un sistema di equazioni lineari (n -equazioni in n -incognite) e dunque $A \in \mathbb{R}^{n \times n}$ una matrice quadrata e i vettori $x, b \in \mathbb{R}^n$;

è noto che se $\det(A) \neq 0$, allora il sistema $Ax=b$ ammette **una e una sola soluzione** data da: $x=A^{-1}b$: applicazione lecita del Teorema di Cramer per la risoluzione del sistema quadrato. Applicazione lecita, ma non la più efficiente! Infatti abbiamo una soluzione che ci evita il calcolo dei determinanti e che evita il noioso metodo dei Cofattori per l'inversione della matrice.

Sapendo che $AA^{-1}=I$ (matrice identità), allora se poniamo il sistema in maniera differente dal solito, otteniamo:

$$A [x_1 \ x_2 \ x_3 \ \dots \ x_n] = [e_1 \ e_2 \ e_3 \ \dots \ e_n],$$

dove $e_i = (0, \dots, 0, 1, 0, \dots, 0)^T$ è l' i -esima colonna della matrice identità (o l' i -esimo versore della base canonica). Allora possiamo vedere tutto come:

$$Ax_i = e_i, \quad i = 1, \dots, n,$$

le colonne di A^{-1} possono essere calcolate mediante la fattorizzazione LU e la risoluzione di n coppie di sistemi triangolari, con complessità $O(\frac{4}{3}n^3)$.

Si noti che la fattorizzazione LU, con pivoting parziale, può essere sfruttata direttamente per calcolare la soluzione: l'illustrazione intende soltanto sottolineare che i cofattori sono l'ultimo carro da prendere in considerazione.

Si illustra una situazione della Teoria dei Sistemi con tale caso.

Dato un sistema lineare autonomo $x'(t) = Ax(t)$, lo stato x_e è un punto di equilibrio se e solo se è soluzione del sistema lineare omogeneo $Ax_e = 0$.

“Un sistema lineare omogeneo ammette sempre la soluzione banale”

Da ciò derivano immediatamente i seguenti risultati.

- Se la matrice A è non singolare, l'unico stato di equilibrio del sistema è $x_e = 0$ ossia l'origine.
- Viceversa, se A è singolare allora il sistema ha un numero infinito di stati di equilibrio che descrivono uno spazio lineare: sono tutti i punti contenuti nello spazio nullo di A . Un sistema lineare autonomo non può pertanto avere stati di equilibrio isolati (siano essi in numero finito o infinito), come invece è possibile nel caso dei sistemi non lineari.

- (ii) Un sistema di equazioni lineari $Ax=b$ (m -equazioni in n -incognite) ammette soluzioni se e solamente se il $r(A)=r(A|b)$. Si parla di rango di una matrice e si tratta il caso matriciale esposto sopra. Si chiama rango di una matrice, i cui elementi non siano tutti nulli, l'ordine massimo dei suoi minori non nulli. Il rango per righe della matrice A è il massimo numero di righe/colonne linearmente indipendenti. Si dimostra in generale che il rango per righe coincide con il rango per colonne.

- (iii) Nel caso di un sistema con matrice a rango pieno in cui $m > n$ allora sono disponibili più equazioni che incognite per cui il problema potrebbe non avere soluzioni. E' lecito sfruttare delle caratteristiche note del fenomeno rappresentato dal sistema e procedere con l'identificazione mediante i minimi quadrati, servendosi del concetto di Varianza Sperimentale proveniente dalla Statistica:

$$s(a_n) = \sum_{n=1}^m (a_n - \mu)^2;$$

in cui il valor medio delle m misurazioni fatte sulla variabile aleatoria a sarà:

$$\mu = \bar{a} = \frac{1}{N} \sum_k^N a_k.$$

APPENDICE 4

Fattorizzazioni

4.1 Generalità sulle fattorizzazioni

Una qualsiasi matrice A può essere scomposta nel prodotto di due particolari matrici, cioè fattorizzata: ad esempio si può scomporre nel prodotto di una matrice triangolare inferiore e di una seconda matrice triangolare superiore (fattorizzazione LU), o anche nel prodotto di una matrice ortogonale e di una matrice triangolare superiore. Quest'ultima costituisce la fattorizzazione QR della matrice A ($A=QR$) e si può vedere che tale fattorizzazione si rivela particolarmente utile ed efficiente nella risoluzione di alcuni problemi nell'analisi numerica. In particolare, se la matrice A di partenza ha dimensioni $m \times n$, la fattorizzazione QR produrrà una matrice Q di dimensioni $m \times m$ e una matrice R di dimensioni $m \times n$. La matrice Q è ortogonale, con la seguente proprietà: $Q^t Q = I$ cioè la sua inversa è pari alla sua trasposta, dato che il prodotto delle due restituisce l'identità; questa proprietà è importante in quanto normalmente il calcolo dell'inversa di una matrice non è banale, e non è sempre possibile in quanto la matrice di partenza potrebbe essere singolare. Al contrario, trovare l'inversa di una matrice ortogonale è particolarmente semplice in quanto richiede solo di considerare la sua trasposta, senza effettuare calcoli. Ne consegue anche che l'inversa di una matrice ortogonale esiste sempre: ciò è confermato anche dal fatto che il suo determinante può assumere solo i valori 1 e -1, quindi si ha sempre $\det Q \neq 0$. Come già accennato, la matrice R è triangolare superiore. La fattorizzazione QR può essere ottenuta in diversi modi: uno dei metodi più famosi fa uso delle matrici di Givens, ma esiste anche un altro metodo che utilizza le matrici di Householder, ed è proprio con quest'ultimo metodo che si illustrerà come calcolare la fattorizzazione QR. Qui di seguito si analizzerà l'utilità di tale fattorizzazione per la risoluzione di sistemi lineari, ad esempio tutti quelli che derivano da LTI di ordine elevato.

4.2 Introduzione al problema di fattorizzazione

La fattorizzazione QR di una generica matrice A non singolare (di dimensioni $m \times n$, con $m > n$) permette di risolvere il sistema $Ax=b$ tramite la risoluzione di due sistemi più semplici:

$$\begin{cases} Qc = b \\ Rx = c \end{cases}$$

dove dal primo sistema, essendo Q una matrice ortogonale, si ricava la soluzione intermedia c : $c = Q^t \cdot b$ trovata c , essa si utilizza come vettore dei termini noti nel secondo sistema: $Rx=c$ con R matrice triangolare superiore, ricavando la soluzione finale x , che è la soluzione del sistema $Ax=b$ di partenza. Si preferisce questo approccio, nonostante i sistemi da risolvere siano due e non più uno, in quanto il calcolo di Q^t è immediato (non richiede al calcolatore l'esecuzione di somme e prodotti), richiedendo solo $O(n^2)$ operazioni per la risoluzione del primo sistema, mentre il secondo sistema si risolve per sostituzione all'indietro, che richiede $O\left(\frac{n^2}{2}\right)$ operazioni. La risoluzione del sistema con un minor numero di operazioni comporta due vantaggi:

- ogni operazione eseguita introduce errore, perciò un minor numero di operazioni introducono un errore più piccolo;
- ogni operazione richiede un certo tempo per poter essere eseguita dal calcolatore, perciò un minor numero di operazioni riduce il tempo di esecuzione totale, velocizzando il calcolo stesso.

Nonostante la fattorizzazione QR richieda un numero doppio di operazioni rispetto alla fattorizzazione LU (cioè rispetto all'algoritmo di Gauss), la fattorizzazione QR è preferita rispetto alla fattorizzazione LU per via della maggiore stabilità. Infatti nella fattorizzazione QR il numero di condizionamento rimane invariato:

$$k_2(A) = k_2(R)$$

ciò rende la fattorizzazione QR particolarmente utile nella risoluzione dei sistemi malcondizionati, a differenza della fattorizzazione LU dove il numero di condizionamento può crescere.

4.3 Matrici di Householder

La fattorizzazione QR di Householder è focalizzata sulla determinazione delle matrici elementari di Householder. Una matrice elementare di Householder è una matrice del tipo: $H = I - 2ww^T$, $w \in \mathbb{R}^n$, $\|w\| = 1$.

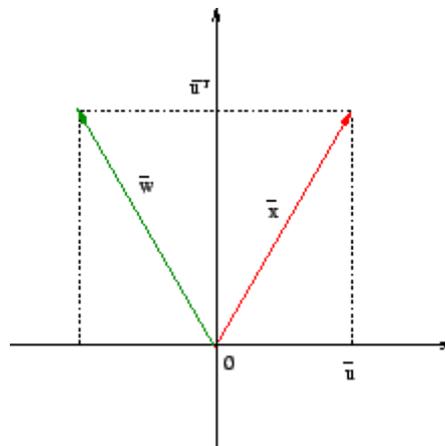
La matrice di Householder è per definizione:

- simmetrica, quindi sappiamo che $H=H^T$;

- è ortogonale, per cui $HH^T = I$

- è una matrice associata ad una riflessione, ogni vettore x moltiplicato per H viene riflesso rispetto al piano.

Di seguito una rappresentazione sul piano cartesiano con riflessione rispetto all'asse delle ordinate per chiarire il concetto:



Si può determinare un valore di w tale da fare in modo che la moltiplicazione a sinistra di questa matrice per un qualsiasi vettore di \mathbb{R}^n lo trasformi in un vettore parallelo al primo versore e_1 della base canonica, ossia: $Hx = ke_1$.

La determinazione analitica del vettore w e quindi della matrice di Householder H conduce alla definizione del seguente algoritmo:

1. $\sigma = \|x\|_2$
2. $k = -\text{sign}(x_1)\sigma$
3. $\lambda = \sqrt{2\sigma(\sigma + |x_1|)}$
4. $w = (x - ke_1)/\lambda$
5. $H = I - 2ww^T$

Si può esprimere la matrice H in una forma diversa che consente di ridurre gli errori e di scrivere un algoritmo modificato:

1. $\sigma = \|x\|_2$
2. $k = -\text{sign}(x_1)\sigma$
3. $\beta = \sigma(\sigma + |x_1|)$
4. $v = (x - ke_1)$
5. $H = I - 1/\beta * vv^T$

In MATLAB una traduzione “Algebra lineare->Italiano->Matlab” dei 5 passaggi potrebbe essere:

```
function [H] = MyHouseholder(x) %il vettore deve essere COLONNA
sigma=norm(x);
if x(1)<0
    k=sigma;
else
    k=-sigma;
```

```

end
beta=sigma*(sigma+abs(x(1)));
v=x;
v(1)=x(1)-k;
[b,a]=size(x);
H=eye(b)-(1/beta)*v*v';
end

```

4.4 Fattorizzazione A=QR di Householder

Di seguito un breve scheduling dell'algoritmo di fattorizzazione della matrice A, prima nel caso quadrato e poi rettangolare.

Passo 0

$$A = A^{(1)} = [a_1^{(1)} \dots a_i^{(1)} \dots a_n^{(1)}]$$

A è considerata in termini di vettori colonna e si costruisce la prima matrice H.

$$H_1 a_1^{(1)} = k_1 e_1;$$

$e_1 \in \mathbb{R}^n$;

Passo 1

$$A^{(2)} = H_1 A^{(1)} = [a_1^{(2)} \dots a_i^{(2)} \dots a_n^{(2)}]$$

$$a_1^{(2)} = k_1 e_1 \quad \square \quad a_j^{(2)} = H_1 a_j^{(1)} \quad ; j=2, \dots, n.$$

$$A^{(2)} = \begin{bmatrix} k_1 & * & * & \dots & \dots & * \\ 0 & * & * & \dots & \dots & \dots \\ 0 & * & * & * & * & \dots \\ \dots & \dots & * & * & * & \dots \\ 0 & * & * & * & * & * \end{bmatrix} = \begin{bmatrix} k_1 & (v_1)^t \\ 0 & A^{*(2)} \end{bmatrix}$$

$$\text{Dim}(A^{*(2)}) = (n-1)(n-1)$$

Passo 2

$$A^{(2)} = \begin{bmatrix} k_1 & a_{12}^{(2)} & a_{13}^{(2)} & \dots & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(2)} & \dots & \dots & \dots & a_{nn}^{(2)} \end{bmatrix} = \begin{bmatrix} k_1 & (v_1)^t \\ 0 & A^{*(2)} \end{bmatrix}$$

$$H_2 a_2^{*(2)} = k_2 \cdot e_1;$$

$$H_2^{*(2)} \in \mathbb{R}^{(n-1)(n-1)};$$

$$e_1 \in \mathbb{R}^{n-1};$$

$$A^{(3)} = \begin{bmatrix} k_1 & a_{12}^{(2)} & a_{13}^{(2)} & \dots & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(2)} & \dots & \dots & \dots & a_{nn}^{(2)} \end{bmatrix} = \begin{bmatrix} k_1 & (v_1)^t \\ 0 & H_2^{*(2)} A^{*(2)} \end{bmatrix}$$

$$H_2^{*(2)} A^{*(2)} \in \mathbb{R}^{(n-2)(n-2)}$$

Passo i

Predisposizione del passo i-1

$$A^{(i)} = \left[\begin{array}{cccc|ccc} k_1 & * & * & * & * & \dots & * \\ 0 & * & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_{i-1} & * & * & \dots \\ \dots & \dots & \dots & \dots & a_{11}^{*(i)} & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots & a_{nn}^{*(i)} \end{array} \right] = \begin{bmatrix} A_{11}^{(i)} & A_{12}^{(i)} \\ 0 & A^{*(i)} \end{bmatrix}$$

$$H_i^* a_i^{*(i)} = k_i \cdot e_1;$$

$$e_1 \in \mathbb{R}^{n-i};$$

$$H_i^* \in \mathbb{R}^{(n+1-i)(n+1-i)};$$

$$A^{(i+1)} = \begin{bmatrix} k_1 & A_{12}^{(i)} \\ 0 & H_i^{*(2)} A^{*(i)} \end{bmatrix}$$

Se A è quadrata, l'algoritmo termina al passo n-1.

Uno sketch dello scheduling analitico per la risoluzione del sistema:

$$\begin{cases} Ax = QRx = b \\ Rx = c \\ c = Q^t \cdot b \end{cases}$$

$$R = A^{(n)} = H_{n-1} A^{(n-1)} = \prod_{j=1}^{n-1} H_{n-j} \cdot A_1 = Q^t \cdot A$$

Nel caso rettangolare si dovrà riempire di zeri la matrice R fino a raggiungere le dimensioni di A e ottenere la blocchizzazione seguente:

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

4.5 Sviluppo dei minimi quadrati con fattorizzazione A=QR

Di seguito i vari passaggi.

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 = \nabla \phi(x_1, \dots, x_n) = \nabla \|QRx - Ib\|_2^2 \text{ è il punto di partenza.}$$

Possiamo scrivere in maniera differente l'argomento della norma .

$$\|QRx - QQ^t b\|_2^2 = \|Q[Rx - c]\|_2^2 = \|Rx - c\|_2^2$$

Passando per le blocchizzazioni di R e $c=(c_1,c_2)$ avremo:

$\nabla[\|R_1x - c_1\|_2^2 + \|c_2\|_2^2]$ che è la funzione da minimizzare. Anziché passare per una matrice Hessiana si avrà un'alternativa tale per cui

-si risolve $R_1x = c_1$

-il minimo residuo è proprio c_2 .

Se $C_2=0$ la soluzione coincide con la soluzione classica. Se $C_2 \neq 0$ la soluzione coincide con quella dei minimi quadrati e la sua norma fornisce la misura del residuo. Se il problema è sovradeterminato troveremo soltanto un minimo diverso da 0, invece nel caso di un problema sottodeterminato i minimi sono più di uno e in generale si cerca x tale che $\|x\|_2$ sia minimo.

Dal punto di vista del Sistema delle equazioni normali (ortogonali), che si ottengono allo stesso modo, senza impiegare la fattorizzazione QR, vanno fatte alcune precisazioni:

$A^tAx=A^tb$ è un sistema lineare e quadrato:

(i) $A \in \mathbb{R}^{m \times n}$

(ii) $A^t \in \mathbb{R}^{n \times m}$

(iii) $A^tA \in \mathbb{R}^{n \times n}$

(iv) $b \in \mathbb{R}^n$

(v) $A^tb \in \mathbb{R}^n$

(vi) $k=\text{rank}(A^tA)=\min(n,n)=n \Rightarrow$ è non singolare

La matrice A^tA è: quadrata e simmetrica \Rightarrow gli autovalori reali puri.

4.6 Fattorizzazione di Cholesky

Sia A una matrice quadrata, hermitiana e definita positiva su campo K ; tale matrice può essere decomposta come (applicazione alle equazioni normali):

$LL^tx = A^tb$ e il sistema prenderebbe la forma:

$$\begin{cases} L \cdot y = c \\ L^t \cdot x = y \end{cases}$$

Con complessità $O\left(\frac{2}{3}n^3\right)$.

Un'illustrazione dell'algorithm :

$$\forall i = 1, \dots, m$$

$$l_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2}$$

$$\forall j = (i + 1), \dots, m$$

$$l_{j,i} = \frac{1}{l_{i,i}} \left(a_{j,i} - \sum_{\ell=1}^{i-1} l_{j,\ell} l_{i,\ell} \right)$$

Bibliografia

- [1] *Giuseppe Rodriguez, Sebastiano Seatzu, Introduzione alla matematica applicata e computazionale*
- [2] *Marco Luise, Giorgio Vitetta, Teoria dei segnali*
- [3] *William L. Briggs, Van Emden Henson, An Owner's Manual for the Discrete Fourier Transform*
- [4] *Appunti docente Giuseppe Rodriguez*
- [5] *Materiale didattico, corso Misure elettriche ed elettroniche, Università di Cagliari, "Analisi di segnali campionati, docente Carlo Muscas*
- [6] *Materiale didattico, corso Analisi matematica 3, Università di Cagliari, "Introduzione alle successioni e alle serie di funzioni", docente Antonio Greco*
- [7] *Immagine, corso Elettrotecnica, Università di Cagliari, "Regime sinusoidale", docente Alessandra Fanni*
- [8] *Leon W. Couch II, Fonfamenti di telecomunicazioni*
- [9] *Sandro Petrizzelli, Politecnico di Bari, <http://users.iol.it/sandry>*
- [10] <https://www.mathworks.com>
- [11] *Giuseppe Rodriguez, Algoritmi Numerici. Pitagora Editrice, Bologna, 2008.*
- [12] *Wikipedia*