UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

Scuola Politecnica e delle Scienze di Base

Area Didattica di Scienze Matematiche Fisiche e Naturali

Dipartimento di Fisica "Ettore Pancini"

Laurea Triennale in Fisica

Studio e realizzazione di filtri digitali tramite software

Relatori:
Prof. Mariagrazia Alviggi
Prof. Riccardo de Asmundis

Candidato:
Flavia Flaminio
Matr. N85001080

Anno Accademico 2018/2019

Indice

Introduzione

I concetti di segnale e di sistema sono presenti in un'ampia gamma di discipline e campi applicativi. Le idee e i metodi sviluppati attorno a questi concetti giocano un ruolo importante in diverse aree scientifiche e tecnologiche come le telecomunicazioni, la progettazione di circuiti, il controllo di processi industriali, l'acustica, l'elaborazione del parlato. Sebbene la natura fisica dei segnali e dei sistemi possa essere differente, essi hanno in comune caratteristiche essenziali. I segnali, che sono grandezze dipendenti da una o più variabili temporali o spaziali, contengono informazioni riguardo lo stato di qualche fenomeno, mentre i sistemi rispondono a sollecitazioni dovute a qualche segnale producendo a loro volta segnali.
Le principali problematiche che si affrontano nello studio dei sistemi sono quelle di analisi e di disegno o progettazione. L'analisi riguarda lo studio di caratteristiche del comportamento di un dato sistema, il disegno si propone di identificare, se esiste, un sistema che esibisca un dato comportamento.
L'obiettivo di questo lavoro di tesi è stato quello di progettare un sistema in grado di ritardare e attenuare un segnale emulando il cosiddetto "effetto eco". In particolare è stato realizzato un filtro COMB a retroazione tramite l'utilizzo di LabVIEW NXG.
La tesi di articola in 6 Capitoli e un'Appendice. Nel Capitolo 1 vengono proposte alcune nozioni di teoria dei segnali. Nel Capitolo 2 viene descritto il passaggio da analogico a digitale (tramite campionamento e quantizzazione). Nel Capitolo 3 viene mostrata la struttura di un DSP. Nei Capitoli 4 e 5 vengono analizzati i filtri digitali (con particolare approfondimento sui filtri COMB) e alcune tecniche di progettazione degli stessi. Nel capitolo 6 viene descritto il lavoro svolto su LabVIEW NXG. Infine, nell'Appendice vi sono alcune nozioni matematiche utili per la comprensione dell'analisi dei segnali e della struttura dei sistemi.

Capitolo 1

Segnali e informazione

Si può pensare ad un segnale come una grandezza fisica che varia con il tempo, lo spazio o una qualsiasi variabile indipendente. Più in generale esso è un'entità contenente qualche tipo di informazione che può essere estratta, trasmessa o elaborata.
Sul piano matematico, un segnale è rappresentato da una funzione $g = f(x)$, dove:

- g denota una variabile (dipendente) sui valori di una grandezza fisica;
- x denota una variabile (indipendente) sui valori di un vettore di coordinate, generalmente spaziali o temporali;
- f denota la relazione funzionale che associa ad ogni valore di x il corrispondente valore della grandezza fisica g.

1.1 Segnali continui e discreti

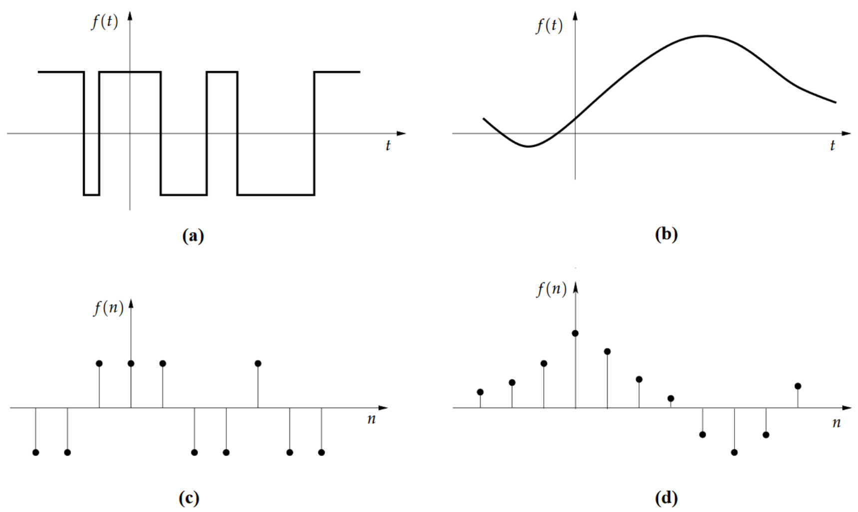
Generalmente un segnale può essere descritto da una funzione $f : A \to B$, dove gli insiemi A e B sono sottoinsiemi di spazi vettoriali di dimensione finita $\mathbb{R}^m$; i valori di A si riferiscono usualmente a coordinate spaziali e/o temporali, mentre quelli in B denotano valori assunti da grandezze fisiche come pressione, tensioni, correnti. Consideriamo qui per semplicità segnali temporali del tipo $y = f(t)$. Abbiamo i seguenti casi:

1. La variabile t assume valori reali oppure valori in un sottoinsieme discreto di numeri reali. Nel primo caso il segnale sarà detto a tempo continuo, nel secondo a tempo discreto.

2. La variabile y assume valori reali oppure valori in un sottoinsieme finito di numeri reali (ad esempio i 2^m numeri codificabili in notazione binaria con m bit). Nel primo caso il segnale sarà detto a valori continui, nel secondo a valori finiti.

Potremmo di conseguenza classificare i segnali come:

- segnali a valori continui e tempo continuo, detti anche segnali analogici (a);
- segnali a valori continui e tempo discreto (b);
- segnali a valori finiti e tempo continuo (c);
- segnali a valori finiti e tempo discreto, detti anche segnali digitali (d).

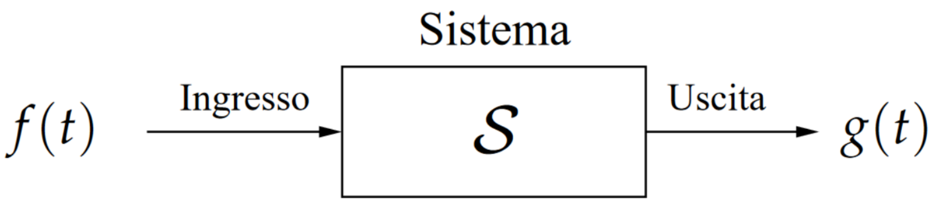


Riguardo ai segnali digitali, è opportuno precisare che essi sono ottenuti mediante campionamento regolare di segnali analogici dove si suppone fissato il passo di campionamento e i campioni vengono ricavati ad istanti che sono multipli interi di esso.

1.2 Sistemi per l'elaborazione di segnali deterministici

Supponiamo di avere una sorgente, cioè una "scatola nera" che, opportunamente sollecitata, produca un segnale $f(t)$. Questo ci permette di riprodurre

il segnale $f(t)$ in modo deterministico: una sollecitazione alla sorgente produrrà sempre lo stesso segnale. Questo termine è anche utilizzato per dare risalto al fatto che tutti i valori passati, presenti e futuri del segnale sono noti senza incertezza alcuna.
Un sistema fisico è un apparato che, ricevendo in ingresso un segnale, dà in uscita un nuovo segnale.



dove $g(t)$ è la risposta del sistema all'ingresso $f(t)$ e S rappresenta una trasformazione.

Un'importante classe di sistemi è quella dei sistemi tempo-invarianti, per i quali una traslazione temporale nel segnale di ingresso produce una identica traslazione temporale nel segnale di uscita. Sistemi fisici di questo tipo hanno caratteristiche che non variano nel tempo e che quindi risponderanno allo stesso modo in ogni istante passato, presente e futuro. Dunque, se S è un sistema lineare tempo-invariante (LTI), il suo comportamento è completamente individuato dalla sua risposta alla funzione impulsiva $\delta(t)$.
Se S è un sistema lineare tempo-invariante, allora

$$S[f(t)] = \int_{-\infty}^{\infty} f(x)h(t-x)dx$$

dove $h(t)$ è la risposta $S[\delta(t)]$ del sistema all'impulso $\delta(t)$.
La legge che associa a due segnali a tempo continuo f e h il segnale $S[f(t)]$ è detta integrale di convoluzione di f e h, ed è denotata da $(f * h)$.

Analogamente per segnali a tempo discreto $x(n)$ e $h(n)$ viene definita la somma di convoluzione.

1.3 Sistemi causali e stabili

Un sistema si dice causale se ad ogni istante di tempo l'uscita dipende unicamente dai valori dell'ingresso al tempo presente e ai tempi passati.

Con particolare riferimento ai sistemi lineari tempo-invariante, un sistema S è causale se la sua risposta $h(t)$ all'impulso è nulla per valori di t negativi, cioè quando $h(t) = 0$ per $t < 0$.
In questo caso il prodotto di convoluzione diventa:

$$S[f(t)] = \int_0^\infty f(x)h(t-x)dx$$

Un'altra nozione di interesse pratico è quella di stabilità. Ricordiamo che un segnale a tempo continuo $f(t)$ è detto limitato se esiste $M > 0$ tale che $|f(t)| < M$ per ogni t. Allora diremo che un sistema S è stabile (o BIBO, cioè Bounded Input Bounded Output) se trasforma segnali limitati in segnali limitati.

1.4 Rappresentazione di segnali digitali

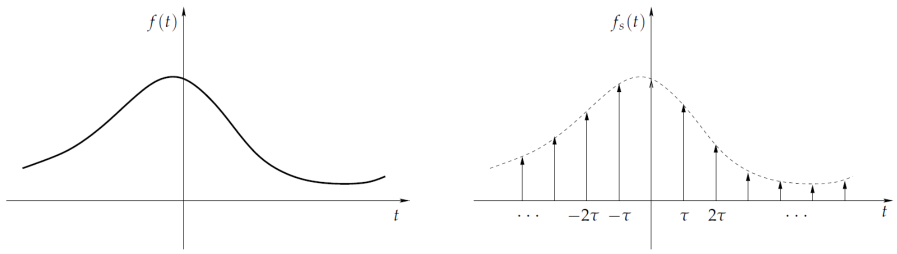
I segnali a tempo discreto possono essere approssimati con segnali digitali. L'analisi in frequenza di tali segnali può essere ottenuta mediante la Trasformata di Fourier a tempo discreto (DTFT).
La trasformata e l'antitrasformata di Fourier operano su segnali continui, sia nel dominio dei tempi che delle frequenze, e sono definite da:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt, \quad f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega$$

Fissato un intervallo di ampiezza τ, per ogni segnale $f(t)$ consideriamo il segnale $f_s(t)$ ottenuto campionando $f(t)$ ai tempi $n\tau$, e concentrando l'energia ai tempi di campionamento; mediante la funzione impulsiva $\delta(t)$ tale segnale può essere riscritto:

$$f_s(t) = \sum_{n=-\infty}^{+\infty} f(n\tau)\delta(t-n\tau)$$

La Figura mostra un segnale $f(t)$ ed il corrispondente segnale campionato $f_s(t)$.



Ricordando che $\mathcal{F}\{\delta(t-t_0)\} = e^{-i\omega t_0}$, per la proprietà di linearità la trasformata di Fourier di $f_s(t)$ risulta essere:

$$F_s(\omega) = \sum_{n=-\infty}^{+\infty} f(n\tau)e^{-i\omega n\tau}$$

Operiamo ora il cambio di variabile $\omega\tau = \Omega$. Denotando con $\nu_s = \frac{1}{\tau}$ la frequenza di campionamento in Hz, risulta che $\Omega = \frac{\omega}{\nu_s}$. La variabile Ω assume dunque significato di frequenza normalizzata alla frequenza di campionamento; poiché ω è data in rad/sec, mentre ν_s è data in cicli/sec, Ω è data in radianti.

Poniamo ora:

$$x(n) = f(n\tau) \qquad X(\Omega) = F_s\left(\frac{\Omega}{\tau}\right)$$

Possiamo definire la trasformata di Fourier di un segnale a tempo discreto come:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-i\Omega n}$$

Osserviamo due differenze sostanziali tra la trasformata di un segnale a tempo continuo avente energia finita e un segnale a tempo discreto anch'esso ad energia finita. La prima è che la trasformata del segnale continuo, e quindi il suo spettro, ha frequenze nell'insieme $(-\infty, +\infty)$. Diversamente, per segnali a tempo discreto, il range di frequenze è unico nell'intervallo $[0, 2\pi]$ o equivalentemente $[-\pi, \pi]$.

La restrizione all'intervallo $[-\pi, \pi]$ per la trasformata a tempo discreto risulta evidente dalla seguente proprietà di periodicità:

$$X(\Omega + 2\pi) = \sum_{n=-\infty}^{+\infty} x(n)e^{-i(\Omega+2\pi)n} = \sum_{n=-\infty}^{+\infty} x(n)e^{-i\Omega n} = X(\Omega)$$

ossia, la funzione $X(\Omega)$ risulta periodica di periodo 2π. Dunque:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-i\Omega n} \qquad x(n) = \frac{1}{2\pi}\int_0^{2\pi} X(\Omega)e^{i\Omega n}d\Omega$$

La trasformata di Fourier a tempo discreto è un caso particolare di trasformata z che si ottiene ponendo $z = e^{i\omega}$. Dal momento che $|e^{i\omega}| = 1$, la trasformata di Fourier a tempo discreto è la valutazione della trasformata zeta sul cerchio unitario nel piano complesso.

1.5 Calcolo veloce della trasformata discreta: FFT

Per calcolare la DFT utilizzando la formula base sono necessarie N moltiplicazioni e somme per ogni campione. Per calcolare la DFT di una sequenza di N campioni servono N^2 moltiplicazioni (generalmente complesse).

La trasformata discreta può essere scritta come:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \qquad k = 0, 1..., N-1;$$

con:

$$W_N^{kn} = e^{-\frac{i2\pi kn}{N}}$$

Sono stati studiati algoritmi (FFT) che riducono la complessità di calcolo della DFT sfruttando alcune regolarità (periodicità, simmetria) di W_N. Essi adottano la tecnica divide-et-impera che consiste nella decomposizione ricorsiva della DTFT in trasformate di dimensioni ridotte ogni volta della metà. Presentiamo qui l'idea base quando la dimensione della DTFT N è una potenza di 2.

Separando nella sommatoria i termini di indice pari da quelli di indice dispari, si ottiene:

$$X(k) = \sum_{npari} x(n)W_N^{kn} + \sum_{ndispari} x(n)W_N^{kn} \tag{1.1}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x(2n)W_N^{kn} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_N^{kn} \tag{1.2}$$

In questo modo il calcolo della DFT di lunghezza N si divide in due DFT di lunghezza $\frac{N}{2}$. Il processo viene iterato finchè si giunge alla trasformata di 2 campioni. Poichè vale che $W_N^2 = W_{\frac{N}{2}}$, allora:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)W_{\frac{N}{2}}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_{\frac{N}{2}}^{kn} \tag{1.3}$$

$$= X_1(k) + W_N^k X_2(k) \tag{1.4}$$

Dato che le sequenze $X_1(k)$ e $X_2(k)$ sono periodiche di periodo $\frac{N}{2}$, cioè $X_1(k + \frac{N}{2}) = X_1(k)$ e $X_2(k + \frac{N}{2}) = X_2$ e, siccome vale $W_N^{k+\frac{N}{2}} = -W_N^k$, possiamo riscrivere la $X(k)$ come:

$$X(k) = X_1(k) + W_N^k X_2(k), \qquad k = 0, 1, ... \frac{N}{2} - 1$$

$$X(k + \frac{N}{2}) = X_1(k) - W_N^k X_2(k), \qquad k = 0, 1, ... \frac{N}{2} - 1$$

Osserviamo che il calcolo di $X_1(k)$, così come di $X_2(k)$, richiede $(\frac{N}{2})^2$ moltiplicazioni di due numeri complessi; il numero di moltiplicazioni richieste per $W_N^k X_2(k)$ è invece $\frac{N}{2}$. Complessivamente, per calcolare $X(k)$ occorrono

$2(\frac{N}{2})^2 + \frac{N}{2} = \frac{N^2}{2} + \frac{N}{2}$ moltiplicazioni, che rappresenta una riduzione di un fattore 2 circa (per N grande) rispetto alle N^2 operazioni richieste. Se operiamo ricorsivamente secondo la strategia appena illustrata, dimezziamo la complessità ad ogni passo di ricorsione dimezzando al contempo la lunghezza della sequenza numerica su cui si opera. Così facendo, se $N = 2^l$, quando la sequenza di ingresso si riduce ad un solo campione la procedura ricorsiva è stata invocata $l = log_2 N$ volte.
Può essere dimostrato che complessivamente, la FFT ha una complessità di calcolo proporzionale a $N log_2 N$.

1.6 Sistemi LTI causali a tempo discreto

Un ulteriore raffinamento nella classificazione dei sistemi a tempo discreto deriva dalla durata della risposta all'impulso $h(n)$ di sistemi causali: cioè durata finita oppure infinita. Parliamo nel primo caso di sistemi FIR(Finite Impulse Response) e nel secondo caso di sistemi IIR(Infinite Impulse Response). Quindi un sistema FIR causale ha una risposta all'impulso nulla tranne che all'interno di una finestra temporale finita, cioè $h(n) = 0$ per $n \geq M$ per un opportuno intero M, riducendo la somma di convoluzione generale alla seguente espressione:

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

Al contrario, i sistemi IIR hanno una risposta all'impulso definita dalla somma di convoluzione:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

I sistemi FIR fanno parte della più generale famiglia di sistemi detti non ricorsivi, perché l'uscita al tempo n dipende solamente dall'ingresso al tempo n e ai tempi precedenti.
I sistemi IIR invece, sono un sottoinsieme della famiglia dei sistemi ricorsivi o retroazionati in cui l'uscita al tempo n è determinata da un numero fissato di uscite ritardate, dall'ingresso al tempo n e ai tempi precedenti.

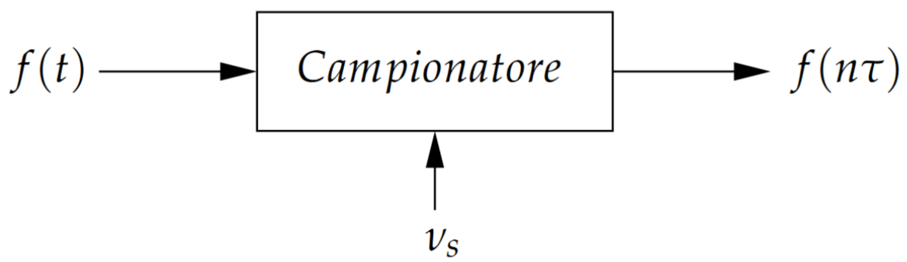
Capitolo 2

Conversione analogico-digitale

I sistemi che trasformano un segnale analogico nel corrispondente digitale sono detti convertitori analogico-digitali (ADC), mentre quelli che realizzano l'operazione inversa, ossia trasformare un segnale digitale in un segnale analogico, sono detti convertitori digitali-analogici (DAC).
I principi di base che sovrintendono detto processo di conversione sono quelli espressi dalle operazioni di campionamento e di quantizzazione.

2.1 Campionamento

Campionare un segnale a tempo continuo significa rilevare le ampiezze del segnale su un insieme discreto di tempi. Ad esempio, fissato un intervallo di tempo di ampiezza τ, un campionamento uniforme con periodo τ di un segnale $f(t)$ corrisponde all'osservazione del segnale ai tempi $n\tau$; il segnale campionato può essere interpretato come il segnale a tempo discreto $f(n\tau)$.
Il sistema campionatore uniforme con frequenza di campionamento $\nu_s = \frac{1}{\tau}$, trasforma quindi un segnale a tempo continuo $f(t)$ nel segnale a tempo discreto $f(n\tau)$.

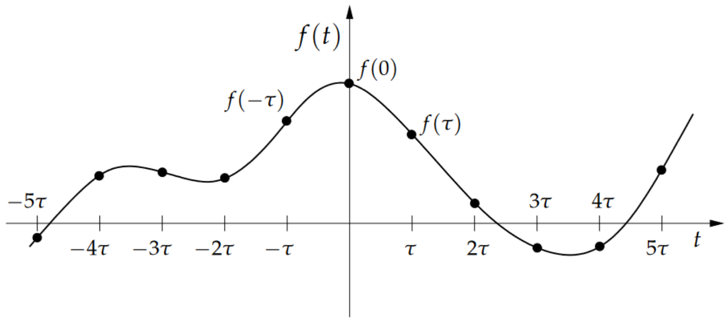


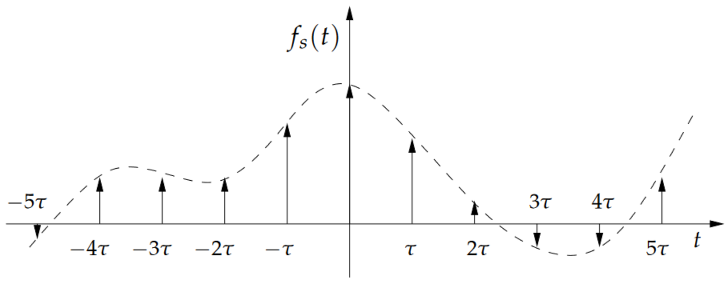
Il campionamento ideale è assimilabile ad un processo di modulazione in cui il segnale portante è definito dal treno di impulsi di Dirac

$$\delta_\tau(t) = \sum_{-\infty}^{\infty} \delta(t - n\tau)$$

Il segnale modulato, denotato con $f_s(t)$, risulta essere pertanto

$$f_s(t) = \sum_{-\infty}^{\infty} f(n\tau)\delta(t - n\tau)$$





È utile osservare che il segnale $f_s(t)$ non risulta essere un modello per segnali reali così come il campionamento ideale non è un sistema fisicamente realizzabile; corrisponde invece ad un'operazione realizzata da sistemi campionatori reali la generazione della sequenza $f(n\tau)$ che qui rappresenta il peso delle funzioni impulsive implicate nel campionamento idealizzato.
Il passaggio successivo è l'analisi nel dominio delle frequenze del sistema. Ricordando che la trasformata di Fourier della funzione impulsiva è la costante unitaria e che la trasformata di una funzione periodica è un treno di impulsi con ampiezza data dai coefficienti della serie di Fourier associata, moltiplicati per 2π, si ha che:

$$\delta(t - n\tau) \leftrightarrow \omega_s \sum_{-\infty}^{\infty} \delta(\omega - n\omega_s)$$

dove $\omega_s = 2\pi\nu_s = \frac{2\pi}{\tau}$ è la frequenza di campionamento in rad/sec.

2.2 Teorema del campionamento

Dato un segnale $f(t)$, è necessario stabilire con quale frequenza deve essere campionato affinché il segnale campionato $f(n\tau)$ contenga la stessa informazione di $f(t)$.
Passando al dominio delle frequenze, se denotiamo con $F_s(\omega)$ la trasformata della funzione $f_s(t)$ allora, per le proprietà della convoluzione, otteniamo:

$$f_s(t) = f(t) \sum_{-\infty}^{\infty} \delta(t - n\tau) \leftrightarrow \frac{1}{2\pi} F(\omega) * \omega_s \sum_{-\infty}^{\infty} \delta(\omega - n\omega_s) = \tag{2.1}$$

$$= \frac{1}{\tau} \sum_{-\infty}^{\infty} F(\omega) * \delta(\omega - n\omega_s) = \tag{2.2}$$

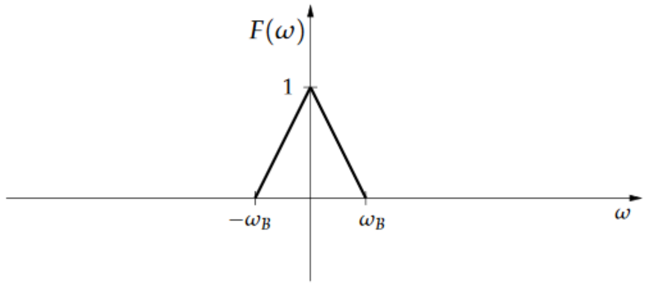
$$= \frac{1}{\tau} \sum_{-\infty}^{\infty} F(\omega - n\omega_s) = F_s(\omega) \tag{2.3}$$

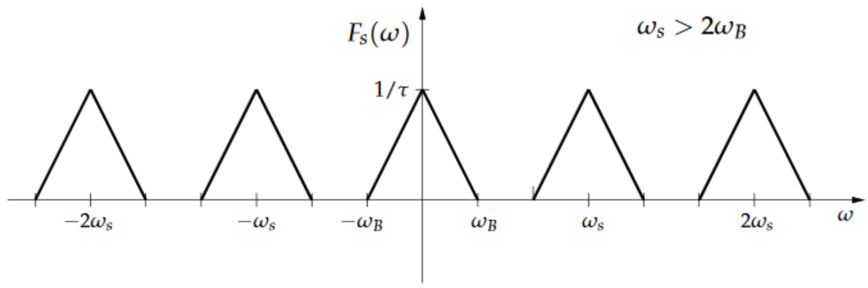
Dunque lo spettro del segnale campionato idealmente è costituito da repliche dello spettro di $f(t)$ traslate in frequenza di $\frac{2\pi}{\tau}$ rad/sec e scalate in ampiezza secondo il fattore $\frac{1}{\tau}$.

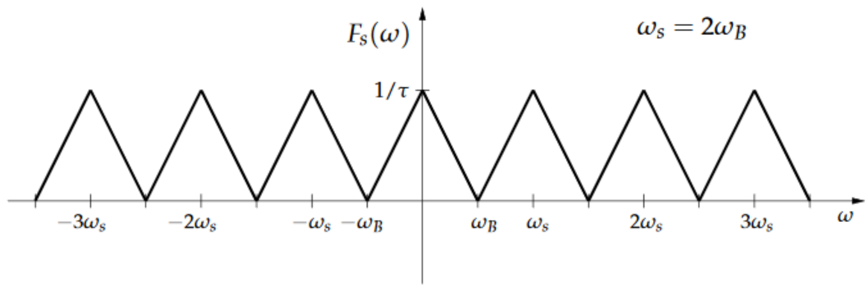
Una relazione importante sussiste tra la frequenza di campionamento ω_s e la massima frequenza $\omega_B = 2\pi\nu_B$ presente nello spettro del segnale campionato $f(t)$.

Teorema 1. *Un segnale $f(t)$ a banda limitata da ω_B rad/sec, la cui trasformata di Fourier $F(\omega)$ è quindi nulla per $|\omega| > 2\pi\nu_B$ rad/sec, può essere univocamente ricostruito dai suoi campioni $f(n\tau)$ presi a frequenza $\nu_s = \frac{1}{\tau}$, se $\omega_s \geq 2\omega_B$. La frequenza $2\nu_B$ Hz è detta tasso o frequenza di Nyquist.*

La seguente figura mostra un segnale campionato con frequenza $\omega_s > 2\omega_B$ e un segnale campionato con frequenza $\omega_s = 2\omega_B$ (frequenza limite).



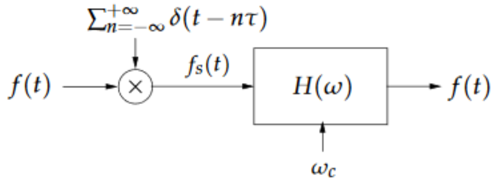




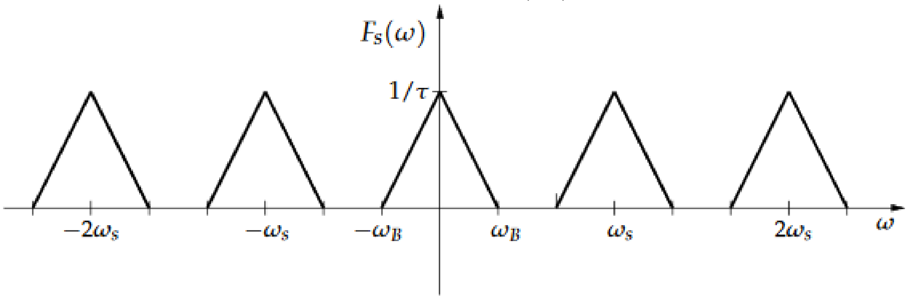
Sotto l'ipotesi del teorema di Shannon, il segnale $f(t)$ può essere ricostruito dalla sua versione campionata $f_s(t)$ mediante l'applicazione di un filtro passa-basso ideale avente frequenza di taglio ω_c tale che:

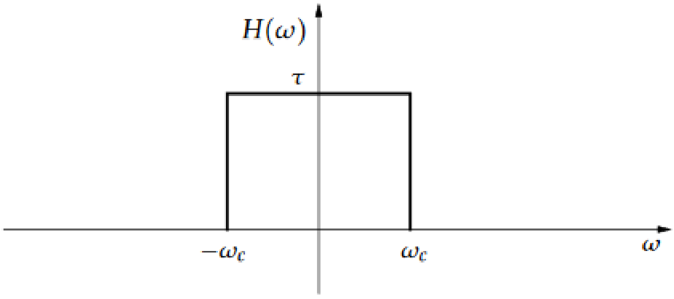
$$\omega_B \leq \omega_c \leq \omega_S - \omega_B$$

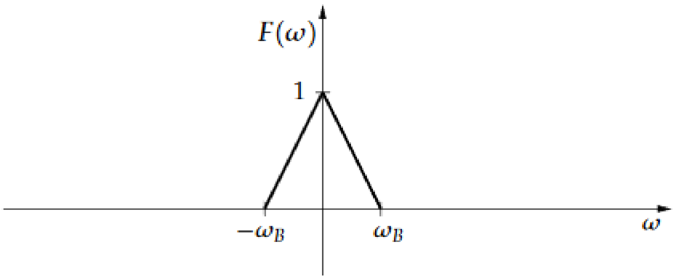
In figura è mostrato uno schema che rappresenta l'applicazione del filtro:



Sappiamo che $h(t) = sinc\left(\frac{\omega_s t}{2}\right)$ del filtro implicato nel processo di ricostruzione ha come spettro $H(\omega) = \tau rect\left(\frac{\omega}{\omega_s}\right)$ come mostrato in figura:







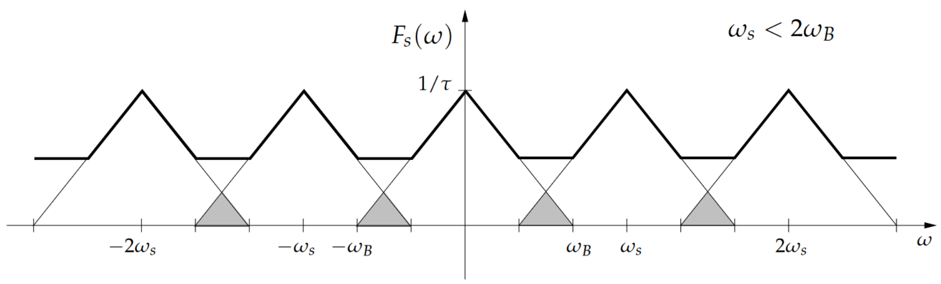
Per le proprietà della convoluzione si ricava:

$$
\begin{aligned}
f(t) = \mathcal{F}^{-1}[H(\omega)F_s(\omega)] &= h(t) * f_s(t) && (2.4)\\
&= sinc\left(\frac{\omega_s t}{2}\right) * \sum_{-\infty}^{\infty} f(n\tau)\delta(t - n\tau) && (2.5)\\
&= \sum_{-\infty}^{\infty} sinc\left(\frac{\omega_s t}{2}\right) * f(n\tau)\delta(t - n\tau) && (2.6)\\
&= \sum_{-\infty}^{\infty} f(n\tau) sinc\left(\frac{\omega_s (t - n\tau)}{2}\right) && (2.7)
\end{aligned}
$$

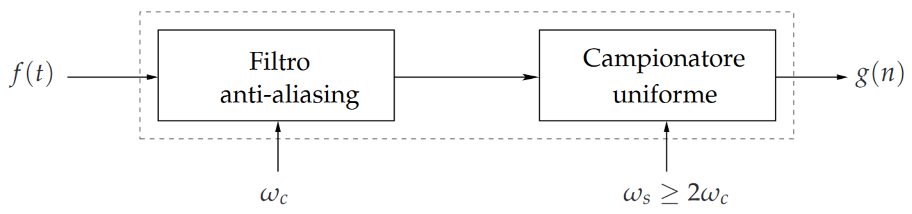
L'aggettivo ideale tiene conto del fatto che il filtro passabasso è ideale e di conseguenza la sua risposta all'impulso è non causale e di durata infinita; fatto questo d'altro canto ben evidente dalle funzioni interpolatrici *sinc*, la cui sovrapposizione ricostruisce il segnale desiderato, che sono infinitamente estese nel tempo in ambedue le direzioni: in altre parole il filtraggio di ricostruzione ideale non è fisicamente realizzabile.

2.3 Aliasing

Quando non sono soddisfatte le condizione del teorema di Shannon, cioè quando la banda del segnale è illimita oppure il segnale è a banda limitata ma la frequenza di campionamento è inferiore al tasso di Nyquist, il metodo di ricostruzione sopra descritto perde la sua efficacia a causa delle sovrapposizioni che si creano nella ripetizione periodica del segnale trasformato. La sovrapposizione di alcune componenti in frequenza crea distorsioni irreversibili perché introduce nuove componenti non presenti nello spettro originale, rendendo impossibile il compito di ricostruzione del segnale dato ad opera del filtro passabasso ideale. Questo fenomeno è detto *aliasing*.



Per evitare questo fenomeno, un sistema campionatore a frequenza ω_s viene normalmente fatto precedere da un filtro passabasso con frequenza di taglio ω_c al più $\frac{\omega_s}{2}$, detto filtro anti-aliasing.



2.4 Quantizzazione

La quantizzazione è il processo che permette di trasformare un segnale a valori continui in un segnale che assume un numero finito di valori. Un modo semplice di quantizzare consiste nel prefissare un un insieme finito di l valori numerici $(x_1, ..., x_l)$ e di associare ad ogni numero x il valore numerico x_k che è più vicino a x. Il passo ulteriore è quello della codifica dei valori in parole binarie opportunamente codificate.
Se i segnali che prendiamo in considerazione hanno ampiezze comprese tra $\frac{V}{2}$ e $-\frac{V}{2}$, questo può essere ottenuto dividendo l'insieme $\left[-\frac{V}{2}, \frac{V}{2}\right]$ in l intervalli, detti livelli, ed attribuendo ad un punto $x \in \left[-\frac{V}{2}, \frac{V}{2}\right]$ il centro del livello in cui x cade. Detti $(x_1, ..., x_l)$ i centri dei vari livelli, l'operazione di quantizzazione può essere allora descritta dalla funzione Q che ad ogni x associa il centro più vicino:

$$Q(x) = \underset{x \in (x_1, ..., x_l)}{\arg\min} \; |x - x_i|$$

Dato che la quantizzazione Q è una funzione molti-uno, essa introduce un errore irreversibile nel segnale quantizzato: dato il segnale quantizzato, non è possibile ricostruire in modo esatto il segnale d'origine.

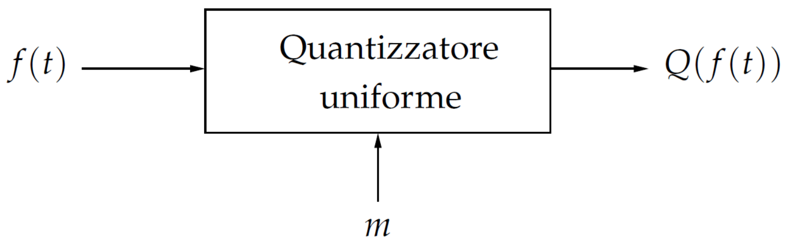
Se i livelli in cui è diviso l'intervallo sono di uguale ampiezza, il quantizzatore si dice uniforme e

$$\delta = \frac{V}{l} = x_i - x_{i+1} \quad \text{con} \quad 1 \leq i \leq l - 1$$

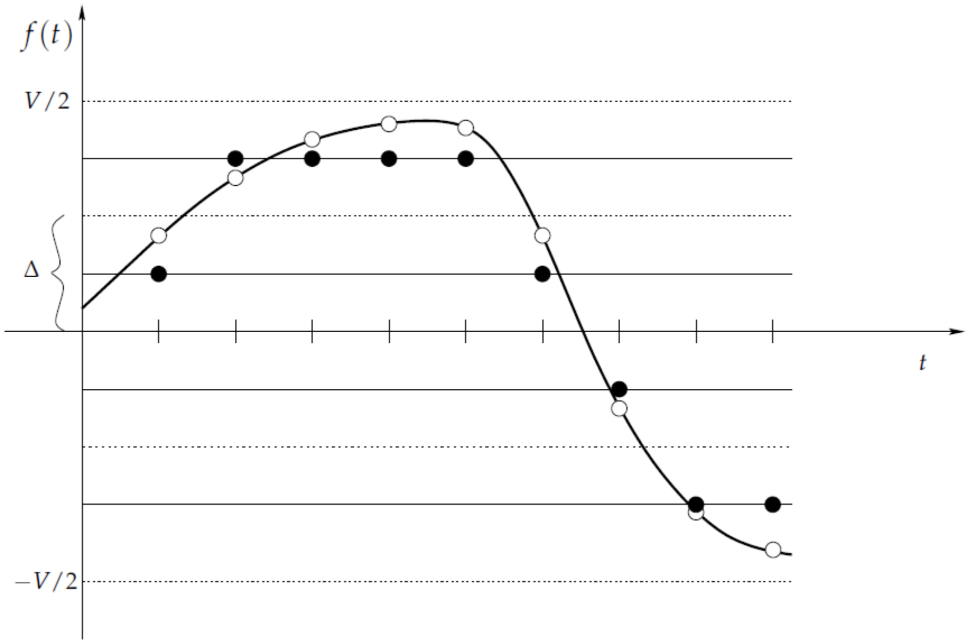
è chiamato passo di quantizzazione.
Se $l = 2^m$, gli elementi $(x_1, ..., x_l)$ possono essere codificati con parole di m bit:

$$x_i = b_{i1}...b_{im} \quad \text{con} \quad b_{ik} \in \{0, 1\} \quad (1 \leq i \leq l)$$

Il sistema in questo caso è detto quantizzatore uniforme:



La figura mostra il risultato del campionamento (pallino bianco) e campionamento più quantizzazione uniforme a quattro livelli (pallino nero) di un segnale $f(t)$.



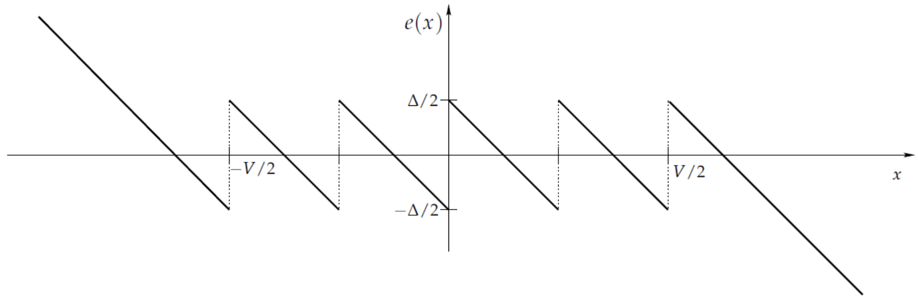
La quantizzazione Q è una funzione molti-uno che introduce un errore irreversibile nel segnale quantizzato. Una naturale misura dell'errore sul numero x è la seguente:

$$e(x) = Q(x) - x$$

L'errore di quantizzazione ha un comportamento ben differenziato in due zone:

1. Se $x < -\frac{V}{2}$ oppure $x > \frac{V}{2}$, l'errore può essere arbitrariamente grande: in questo caso l'errore è detto errore da sovraccarico (overload) e lo si controlla cercando di garantire che i valori del segnale $f(t)$ in ingresso al quantizzatore rientrino nel range del quantizzatore, cioè che $-\frac{V}{2} \leq f(t) \leq \frac{V}{2}$.

2. Se x è invece interno all'intervallo $-\frac{V}{2} \leq x \leq \frac{V}{2}$, l'errore $e(x)$ si mantiene in valore assoluto minore o uguale a $\frac{\Delta}{2}$; tale errore è detto rumore granulare.



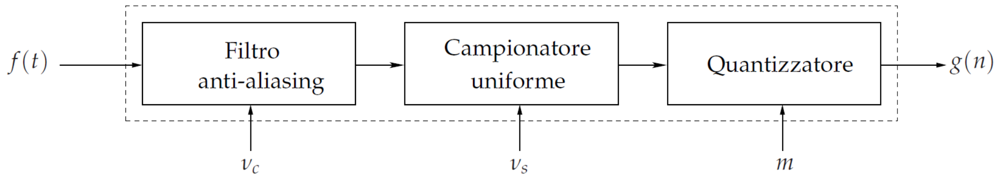
Una misura di prestazione del quantizzatore è data dal rapporto segnale-rumore di quantizzazione SQNR (Signal-to-Quantization-Noise Ratio), espresso in scala logaritmica e misurato in decibel (dB):

$$SQNR = 10 \log_{10} \frac{\sigma^2}{\sigma_e^2}$$

dove σ^2 è la varianza (o potenza) del segnale e σ_e^2 e l'errore di quantizzazione quadratico medio (varianza dell'errore).
Spesso per segnali reali la probabilità che un segnale abbia valore tra y e $y + dy$ viene a dipendere da y. In questo caso si utilizza un quantizzatore non uniforme.

2.5 Convertitore analogico-digitale

Applicando un campionatore a frequenza ν_s e consecutivamente un quantizzatore a m bit, un segnale $f(t)$ osservato per un tempo T può essere trasformato in un vettore di $T\nu_s$ componenti a m bit: esso può quindi essere memorizzato in forma digitale usando $T\nu_s m$ bit ed eventualmente modificato.
Il sistema che realizza questa trasformazione è detto convertitore analogico-digitale (ADC):



Esistono essenzialmente due differenti tipologie di convertitori analogico-digitale:

- Nel primo tipo, rappresentato nella figura precedente, il campionatore opera vicino alla frequenza di Nyquist del segnale e il quantizzatore è un quantizzatore ad m bit ($m \gg 1$). Ne è un esempio il Flash ADC.
- Nel secondo tipo si usa un campionatore a frequenza molto superiore alla tasso di Nyquist (sovracampionamento), un quantizzatore a 1 bit e varie operazioni digitali.

Il flash ADC è sicuramente il più veloce convertitore disponibile, ma è costituito da un numero elevato (2^m) di componenti che devono essere molto accurati: questo rende difficile e costosa la realizzazione per alti valori di m. Questo fatto è una caratteristica dei convertitori analogico-digitali di questo tipo, che richiedono un'estrema accuratezza nella costruzione del sistema quantizzatore. Ciò limita la costruzione di flash ADC ad un numero massimo di 6-8 bit. Risulta allora conveniente un diverso tipo di convertitore, basato sull'idea di operare con frequenze di campionamento molto superiori al tasso di Nyquist utilizzando però un quantizzatore a pochi bit. Il vantaggio ottenuto è duplice:

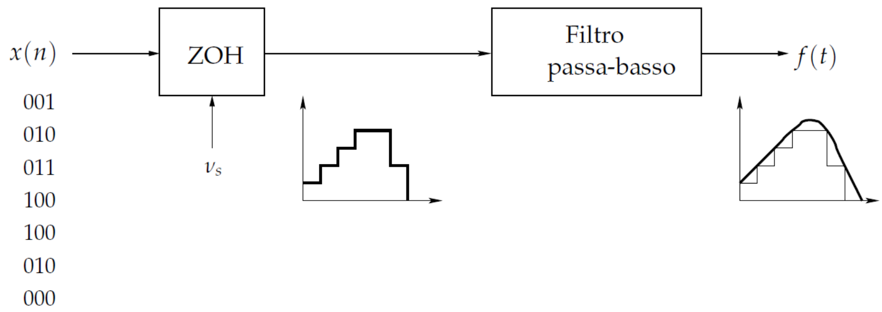
1. operando a frequenze molto superiori al tasso di Nyquist (sovracampionamento), il filtro anti-aliasing diventa meno critico di quanto non lo sia nei convertitori del primo tipo, e può essere progettato con maggior facilità;
2. l'aumento della frequenza di campionamento si può tradurre in un miglioramento del SQNR del convertitore.

2.6 Convertitore digitale-analogico

Il convertitore digitale-analogico (DAC) trasforma un segnale digitale, a tempo e valori discreti, in un segnale analogico. Un modo semplice per convertire un segnale digitale $x(n)$ a frequenza ν_s (i cui valori sono specificati da parole di m bit) è quello di trasformarlo nel segnale analogico $g(t)$, dove:

$$g(t) = x(n) \quad \text{per} \quad n\tau \leq t < (n+1)\tau \quad \text{con} \quad \tau = \frac{1}{\nu_s}$$

Questo tipo di convertitore è detto di tipo ZOH (Zero-Order-Hold): la parola binaria al tempo $n\tau$ è convertita nel corrispettivo valore analogico, e tale valore viene mantenuto per tutto l'intervallo seguente di ampiezza τ. Il segnale ottenuto è descritto da una funzione a scala, che può essere "lisciata" applicando un opportuno filtro passabasso.



2.7 Sovracampionamento nella Conversione Analogico-Digitale

Sovracampionare significa operare un campionamento con frequenza ν_s molto superiore al tasso di Nyquist; il tasso di sovracampionamento è dato dal rapporto tra frequenza di campionamento e tasso di Nyquist:

$$sovracampionamento = \frac{\nu_s}{2\nu_B}$$

Una caratteristica degli attuali ADC è quella di utilizzare al massimo le potenzialità del sovracampionamento, ottenendone due vantaggi rilevanti:

- possibilità di utilizzo di filtri anti-aliasing con prestazioni non elevate, poiché un filtro passabasso, lavorando a frequenze

$$\frac{\nu_s}{2}$$

molto maggiori della sua frequenza di taglio ν_B, è in grado di garantire una miglior attenuazione;

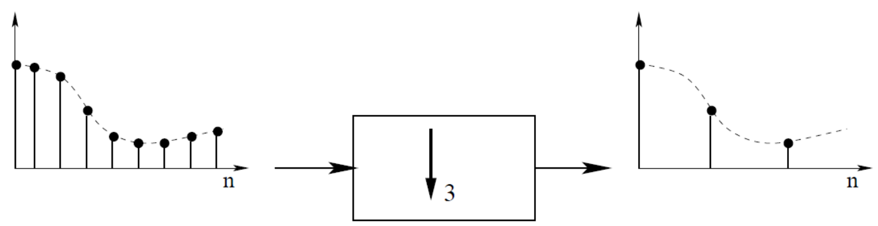
- miglioramento del rapporto segnale-rumore SQNR dovuto al sovracampionamento.

2.8 Sistemi in Multifrequenza: Decimazione e Interpolazione

I moderni sistemi digitali possono processare i dati a più di una frequenza di campionamento. Le due operazioni base che permettono di modificare digitalmente le frequenze sono la decimazione e l'interpolazione: l'interpolazione aumenta la frequenza di campionamento, la decimazione la riduce comprimendo di fatto i dati. Naturalmente questi obbiettivi devono essere ottenuti senza introdurre effetti indesiderati, come errori di quantizzazione o aliasing.

La decimazione $\downarrow_M$ di un fattore M è ottenuta da un sistema con la seguente relazione ingresso-uscita:

$$M(x(n)) = x(Mn)$$

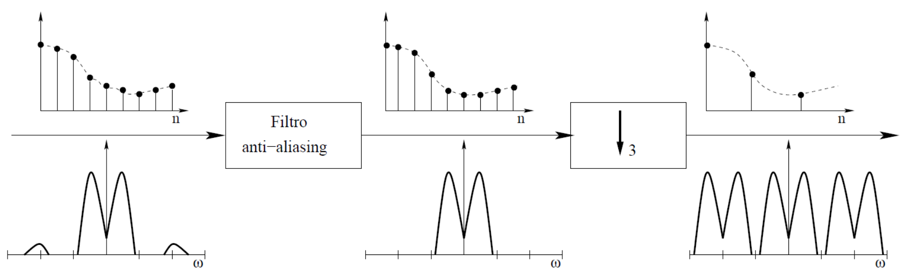


Tale sistema è chiaramente lineare, ma non a tempo invariante: vogliamo qui studiarne la risposta in frequenza. A tal riguardo, siano $X(z)$ e $Y(z)$ le trasformate zeta rispettivamente dell'ingresso $x(n)$ e dell'uscita $y(n) = x(Mn)$; le funzioni $F(\omega) = X(e^{i\omega})$ e $G(\omega) = Y(e^{i\omega})$, periodiche di periodo 2π, descrivono lo spettro di frequeza di $x(n)$ e $y(n)$. La relazione tra $F(\omega)$ e $G(\omega)$ è:

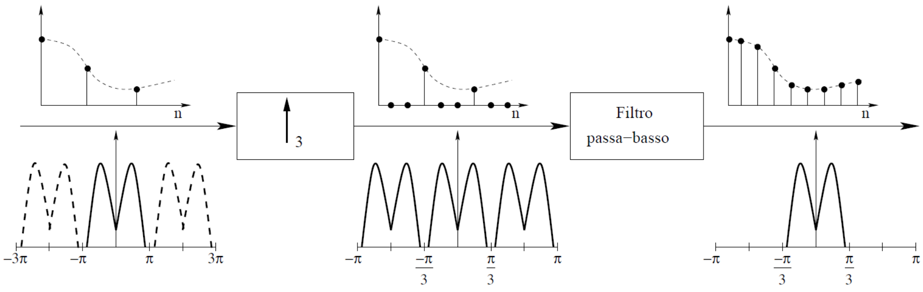
$$G(\omega) = \frac{1}{M}(F(\omega) + F(\omega - \frac{2\pi}{M}) + ... + F(\omega - \frac{2(M-1)\pi}{M}))$$

Osserviamo che la funzione $G(\omega)$ è periodica di periodo $2\pi M$: se F_s è la frequenza di campionamento del segnale di ingresso, la frequenza del segnale di uscita risulta $\frac{F_s}{M}$.

Osserviamo inoltre che il segnale di ingresso può essere perfettamente ricostruito dal segnale di uscita se si evita il fenomeno dell'aliasing, e cioè se il limite di banda del segnale di ingresso è $\frac{\pi}{M}$. Se quindi desideriamo che l'operazione di decimazione non crei perdita di informazione rispetto al segnale di ingresso, tale operazione dovrà essere preceduta da un filtro passa-basso, con frequenza di taglio $\frac{\pi}{M}$. In figura viene mostrato il sistema Decimatore, composto da un filtro anti-aliasing e dall'operazione di decimazione:



Un interpolatore con fattore L trasforma un segnale campionato con frequenza F_s in uno campionato con frequenza LF_s. In Figura è rappresentato un sistema Interpolatore:



Esso è costituito da un'operazione di interpolazione $\downarrow_L$ che inserisce, tra $x(n)$ e $x(n+1)$, $L-1$ campioni a valore 0. Il nuovo segnale viene poi filtrato con un filtro digitale passabasso a frequenza di taglio $\frac{F_s}{2L}$, dove F_s è la frequenza di campionamento di $x(n)$. Poiché l'inserzione di $L-1$ zeri "distribuisce" l'energia di un campione su L campioni, l'uscita $y(n)$ risulta attenuata di un fattore $\frac{1}{L}$: questo fatto può essere compensato moltiplicando per L il valore dell'uscita.

Come è ben evidenziato dalla rappresentazione in frequenza, il Decimatore e l'Interpolatore rappresentano sistemi l'uno inverso dell'altro.

Capitolo 3

Architetture DSP

In generale, si possono distinguere fondamentalmente due approcci all'elaborazione digitale dei segnali:

- **Approccio mediante programmazione:** in questo caso le caratteristiche di sistemi che trasformano segnali vengono simulate da programmi che vengono eseguiti da un elaboratore digitale. La velocità di esecuzione del programma dipende dal tipo di elaboratore ed in particolare dal livello di parallelismo e dalla velocità con cui vengono eseguite le istruzioni. Questo tipo di approccio risulta di grande flessibilità: sistemi totalmente diversi vengono simulati sulla stessa macchina, semplicemente eseguendo algoritmi diversi.

- **Approccio mediante realizzazione h/w:** in questo caso gli algoritmi che simulano i sistemi vengono realizzati in h/w con circuiti dedicati a applicazioni specifiche (ASIC) o mediante circuiti programmabili (PLD). La velocità di elaborazione dipende dalla frequenza di clock e dalla propagazione dei ritardi nei circuiti logici. Questo approccio permette realizzazioni più efficienti delle soluzioni s/w, con velocità di elaborazione maggiore di un fattore 10^2, risultando tuttavia molto meno flessibile, in quanto la simulazione di un nuovo sistema costringe a ridisegnare e rimpiazzare gran parte del circuito.

3.1 Architettura Von Neumann e Harvard

I microprocessori programmabili per l'elaborazione digitali dei segnali possono essere raggruppati nelle seguenti categorie: microprocessori general-purpose, microcontrollori, processori specializzati all'elaborazione dei segnali

(DSP).

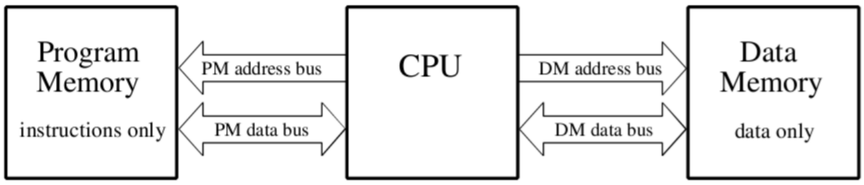
- **Microprocessori general-purpose:** questi microprocessori devono supportare le più disparate applicazioni, quindi la loro architettura viene progettata per l'ottimizzazione della gestione della memoria; le loro prestazioni nell'elaborazione digitale dei segnali risultano tuttavia mediocri.

- **Microcontrollori:** questi strumenti implementano singole parti di un elaboratore, ad esempio apparecchi per l'input-output, memorie RAM e ROM; l'architettura è generalmente funzionale all'ottimizzazione delle caratteristiche input-output.

- **Processori specializzati all'elaborazione dei segnali (DSP):** questi microprocessori sono appositamente studiati per ottimizzare le prestazioni nell'elaborazione dei segnali. Poiché gli algoritmi per la simulazione di sistemi per segnali consistono spesso nella iterazione di sequenze di semplici operazioni aritmetiche, grande attenzione è posta nell'ottimizzazione dell'unità aritmetico-logica (ALU): i primi DSP sono addirittura stati motivati dalla necessità di accelerare l'esecuzione dell'operazione di moltiplicazione, rispetto agli usuali microprocessori. La necessità di grande velocità di elaborazione imposte dalle applicazioni in tempo-reale richiede inoltre l'introduzione di architetture che sfruttino l'inerente parallelismo di alcune funzionalità, pur sacrificando la semplicità realizzativi e la flessibilità rispetto alle applicazioni.

Per quanto detto, i DSP richiedono architetture di calcolo piuttosto differenti rispetto agli usuali microprocessori general-purpose. Un elaboratore programmabile riceve in ingresso dati di due diversi tipi: istruzioni per il programma e dati veri e propri. La maggior parte dei microprocessori general-purpose è basata sull'architettura proposta da Von Neumann, in cui dati e programmi vengono memorizzati nella stessa area di memoria. Esiste un'unica area di memoria per dati e programmi, ed il processore usa gli stessi bus dati e indirizzi per accedervi. Le unità basilari sono l'unità aritmetico-logica (ALU) e l'unità di input-output (IO). La ALU permette l'esecuzione di operazioni aritmetiche, mentre l'unità di input-output gestisce il flusso di dati esterni alla macchina. Per questo tipo di macchina, i programmi sono sequenze di istruzioni e la singola istruzione generalmente contiene un comando di operazione e l'indirizzo del dato su cui il comando deve essere eseguito.
Poiché dati e programmi sono memorizzati nello stesso spazio di memoria,

non è possibile prelevare (fetch) l'istruzione seguente mentre l'istruzione corrente è in esecuzione dato che entrambe le operazioni prevedono un accesso a memoria: questo comporta un collo di bottiglia che rallenta la velocità di esecuzione.
Poiché la velocità di esecuzione è un elemento critico nell'elaborazione dei segnali, l'architettura Von Neumann non è adatta per questo tipi di applicazioni. Va per contro segnalato che la presenza di un'unica area per dati o programmi permette una buona flessibilità nell'uso della memoria: se l'applicazione cambia, il sistema può con facilità riallocare le risorse di memoria per permettere la nuova applicazione.
Un'architettura alternativa che permette un aumento di velocità, come richiesto dalle applicazioni concernenti l'elaborazione dei segnali, è l'architettura di Harvard. Questa architettura prevede due spazi di memoria, uno per i dati, l'altro per i programmi, e corrispondentemente diversi bus dati e bus indirizzi per accedervi, e permette di accedere contemporaneamente sia ai dati che alle istruzioni ottenendo migliori prestazioni in velocità.



L'architettura Harvard aumenta la velocità di esecuzione, ma introduce alcuni svantaggi:

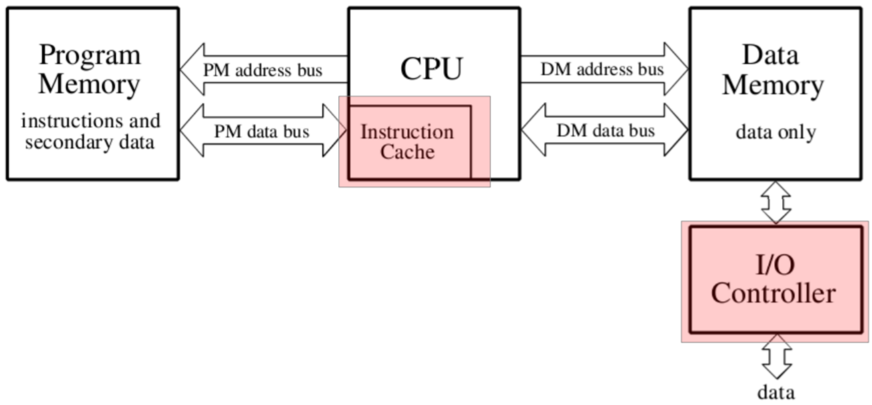
- Le componenti h/w necessarie alla realizzazione di una architettura Harvard sono più complesse che per l'architettura Von Neumann, e questo comporta aumento di prezzo.

- Le due distinte aree di memoria per dati e programmi tipiche dell'architettura Harvard non possono essere riallocate con la flessibilità usuale nell'architettura Von Neumann.

- Scrivere programmi per una architettura Harvard è un po' più complicato che per l'architettura Von Neumann.

Un'ulteriore modifica porta all'architettura di tipo Super Harvard, termine coniato dalla Analog Devices per denominare i dispositivi della famiglia ADSP-2106x (SHARC, Super Harvard ARChitecure). In un dispositivo SHARC sono inseriti due nuovi elementi:

- una cache per le istruzioni;
- un controllore per I/O.



3.2 Integrazioni delle componenti di un DSP

La richiesta di eseguire ogni istruzione in un singolo ciclo di clock comporta la necessità di realizzare su un singolo chip di silicio il processore, l'area di memoria per i dati, l'area di memoria per i programmi e i rispettivi bus.
Tra le apparecchiature fondamentali che rendono possibile l'elaborazione digitale vi sono i convertitori analogico-digitale (ADC) e digitale-analogico (DAC): interfacciare in modo efficiente il processore digitale con i convertitori ADC e DAC è di grande importanza per una efficiente elaborazione.
Sono proposte due soluzioni h/w:

1. integrazione dei convertitori ADC e DAC nello stesso chip del DSP;

2. presenza di h/w speciale per interfacciare efficientemente convertitori e DSP.

3.3 SIMD e Superscalari

Avendo come obbiettivo miglioramenti significativi nelle prestazioni, l'attuale tendenza nelle architetture per DSP è di aumentare sia il numero di operazioni compiute per istruzione che il numero di istruzioni eseguite per ogni ciclo. Di conseguenza, le architetture dei nuovi DSP devono saper sfruttare in modo intelligente le tecniche di elaborazione parallela. I due principali approcci sono: architetture SIMD e architetture superscalari.
Sistemi con architettura SIMD (Single Instruction Multiple Data) devono permettere l'esecuzione parallela della stessa operazione su differenti dati. Per ottenere questo obbiettivo, il DSP deve contenere diversi bus dati e diverse unità di elaborazione: in questo modo in un ciclo la stessa istruzione può essere eseguita dalle varie unità di elaborazione su differenti dati.

L'elaborazione superscalare è una tecnica per aumentare la velocità di calcolo sfruttando invece il potenziale parallelismo a livello di istruzioni. E' noto infatti che, in certe circostanze, alcune istruzioni possono essere eseguite in modo indipendente: avendo a disposizione un adeguato numero di unità di elaborazione, la loro esecuzione in parallelo sulle varie unità accelera il calcolo.

Capitolo 4

Filtri FIR e IIR

I filtri digitali FIR e IIR sono particolari sistemi LTI causali a tempo discreto. Essi possono essere implementati e simulati su macchine digitali (microprocessori o processori specializzati come i DSP); per molti anni addirittura essi sono risultati la più comune applicazione dei DSP.

Il vantaggio sui filtri analogici è duplice:

- essi possono essere riprogrammati via software sullo stesso hardware;
- è possibile modificare in tempo reale i coefficienti dei filtri, ottenendo in tal modo filtri "adattativi".

4.1 FIR

Nella teoria dei segnali, un sistema dinamico finite impulse response (spesso abbreviato in FIR), è un sistema LTI causale con risposta finita all'impulso, cioè che si annulla ad un tempo finito.
L'uscita $y(t)$ di un sistema dinamico lineare tempo invariante (LTI) a tempo continuo soggetto ad un segnale in ingresso $x(t)$ è descritta dalla convoluzione $y(t) = x(t) * h(t)$, dove $h(t)$ è la risposta del sistema quando l'ingresso $x(t)$ è una funzione a delta di Dirac. L'uscita y è quindi proporzionale alla media dell'ingresso x pesata dalla funzione $h(-\tau)$, traslata di un tempo t.
Un sistema dinamico lineare stazionario discreto trasforma la successione in ingresso x in un'altra successione y, data dalla convoluzione discreta con la risposta h alla delta di Kronecker:

$$y(n) = \sum_{-\infty}^{\infty} x(k)h(n-k) = \sum_{-\infty}^{\infty} x(n-k)h(k)$$

Per un filtro a tempo discreto l'uscita è una somma pesata dei valori assunti dall'ingresso al tempo corrente ed a tempi precedenti. Tale operazione è descritta dalla seguente equazione:

$$y(n) = h_0 x(n) + h_1 x(n-1) + ... + h_N x(n-N) = \sum_{i=0}^{N} h_i x(n-i)$$

dove h_i sono detti coefficienti del filtro ed N l'ordine del filtro.
Passando alle trasformate z e applicando la proprietà della traslazione temporale, si ottiene:

$$Y(z) = H(z)X(z)$$

dove $H(z) = \sum_{k=0}^{M-1} h(k)z^{-k}$ e $X(z)$, $Y(z)$ sono le trasformate z rispettivamente di $x(n)$ e $y(n)$. $H(z)$ è chiamata funzione di trasferimento del sistema.

4.2 Caratteristiche

Le caratteristiche più interessanti dei filtri FIR sono le seguenti:

1. Un filtro FIR è sempre causale e stabile (l'uscita ad un ingresso limitato è anch'essa limitata). Ciò può essere rilevato dal fatto che $H(z)$ è un polinomio in z^{-1}, e quindi ha un solo polo in $z = 0$, di fatto interno alla cerchio di raggio 1.

2. Un filtro FIR può avere fase lineare: se la funzione $h(n)$ è simmetrica o antisimmetrica rispetto a $\left(\frac{M-1}{2}\right)$, cioè $h(k) = h(M-1-k)$ oppure $h(k) = -h(M-1-k)$, allora la fase di $H(e^{i\omega})$ è lineare.
   Infatti, se $h(k) = h(M-1-k)$:

   $$H(e^{i\omega}) = \sum_{k=0}^{M-1} h(k)(e^{-ik\omega}) =$$

$$= \sum_{k=0}^{M-1} h(k) \left(\frac{e^{-ik\omega} + e^{-i(M-1-k)\omega}}{2} \right) =$$

$$= e^{-i\frac{M-1}{2}} \sum_{k=0}^{M-1} h(k) \left(\frac{e^{i\left(\frac{M-1}{2}-k\right)\omega} + e^{-i\left(\frac{M-1}{2}-k\right)\omega}}{2} \right) =$$

$$= e^{-i\frac{M-1}{2}} \sum_{k=0}^{M-1} h(k) \cos\left(\frac{M-1}{2} - k \right) \omega$$

Poichè

$$\sum_{k=0}^{M-1} h(k) \cos\left(\frac{M-1}{2} - k \right) \omega$$

è un numero reale, la fase risulta essere lineare, cioè il ritardo introdotto dal filtro è costante per tutte le componenti in frequenza.
Analogamente, se $h(k) = -h(M-1-k)$, la fase risulta essere:

$$\sum_{k=0}^{M-1} h(k) \sin\left(\frac{M-1}{2} - k \right) \omega$$

4.3 IIR

Con filtri IIR intendiamo quella sottoclasse dei sistemi LTI causali con risposta anche infinita all'impulso. I filtri IIR sono spesso descritti attraverso un'equazione alle differenze, che definisce il comportamento dell'uscita $y(n)$ in funzione del segnale x in ingresso e in funzione del segnale di uscita stesso $y(n)$:

$$y(n) = \frac{1}{a_0}[b_0 x(n) + b_1 x(n-1) + ... + b_P x(n-P))-$$

$$-a_1 y(n-1) - a_2 y(n-2) - ... - a_Q y(n-Q)]$$

ove P è l'ordine del filtro, b_i sono i coefficienti del filtro, Q è l'ordine della retroazione del filtro, a i a_i i coefficienti della retroazione. In una forma più concisa si ha:

$$y(n) = \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i x(n-i) - \sum_{j=1}^{Q} a_j y(n-j)\right)$$

che può essere scritta come:

$$\sum_{j=0}^{Q} a_j y(n-j) = \sum_{i=0}^{P} b_i x(n-i)$$

Osserviamo che se $Q = 0$ l'equazione precedente definisce un filtro FIR.

I filtri IIR possono essere instabili e la loro fase, in generale, non è lineare. Gli svantaggi sono compensati dalla maggior semplicità realizzativa e dalle migliori caratteristiche di attenuazione a parità di ordine delle equazioni dei filtri FIR.

4.4 Zeri di filtri a fase lineare

Un filtro FIR ha funzione di trasferimento del tipo

$$H(z) = \frac{P(z)}{z^M}$$

dove $P(z)$ è un polinomio di grado M a coefficienti reali.Le due seguenti proposizioni sono equivalenti:

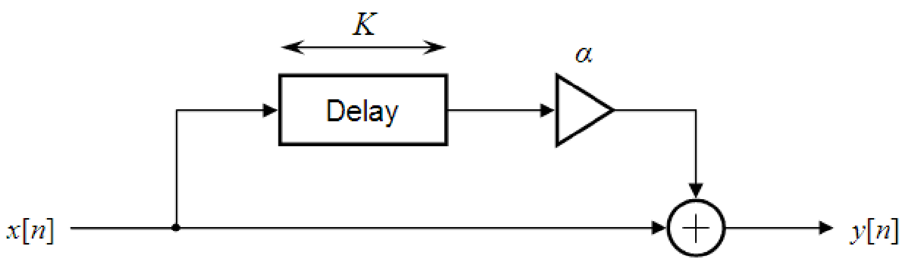
1. Il filtro ha fase lineare.
2. Se $re^{i\theta}$ è uno zero di $P(z)$, allora anche $\frac{1}{r}e^{i\theta}$ è uno zero di $P(z)$.

Ne segue in particolare che filtri i cui zeri sono tutti sulla circonferenza di raggio 1 hanno fase lineare.

4.5 Filtri COMB

Un'importante famiglia di filtri che hanno gli zeri sulla circonferenza unitaria è quella dei filtri comb (pettine). Essi sono particolari filtri che aggiungono al segnale al tempo presente una sua versione ritardata (delay) di un certo numero di passi. La risposta in frequenza di un filtro comb consiste in una serie di impulsi equispaziati che ricordano i singoli denti di un pettine. I filtri comb esistono in due differenti tipologie con feedforward o in retroazione (feedback).

Lo schema di un filtro COMB feed forward è il seguente



e può essere descritto dalla seguente equazione alle differenze:

$$y[n] = x[n] + \alpha x[n - K]$$

dove K è la lunghezza della linea di ritardi (delay line) misurata, essendo a tempo discreto, in numero di campioni α è il fattore di scala applicato al segnale nella linea di ritardo. Attraverso la trasformata z dell'equazione precedente si ottiene:

$$Y(z) = (1 + \alpha z^{-K})X(z)$$

Con la seguente funzione di trasferimento:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \alpha z^{-K}$$

Per ottenere il segnale in frequenza del dominio della trasformata z applichiamo la seguente sostituzione: $z = e^{i\omega}$. Quindi otteniamo:

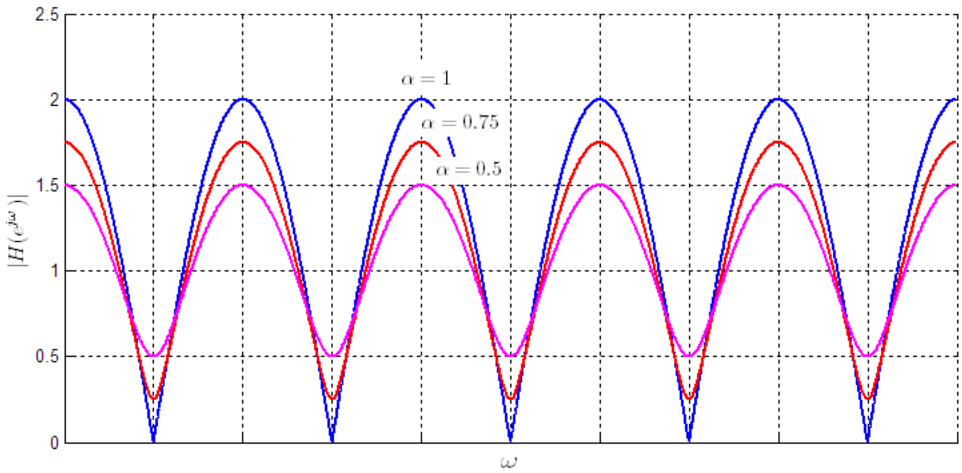
$$H(e^{i\omega}) = 1 + \alpha e^{-i\omega K} =$$

$$= (1 + \alpha \cos K\omega) - i\alpha \sin K\omega$$

Ignorando la fase e quindi considerando il modulo di $H(e^{i\omega})$ otteniamo:

$$|H(e^{i\omega})| = \sqrt{(1+\alpha^2) + 2\alpha cos K\omega}$$

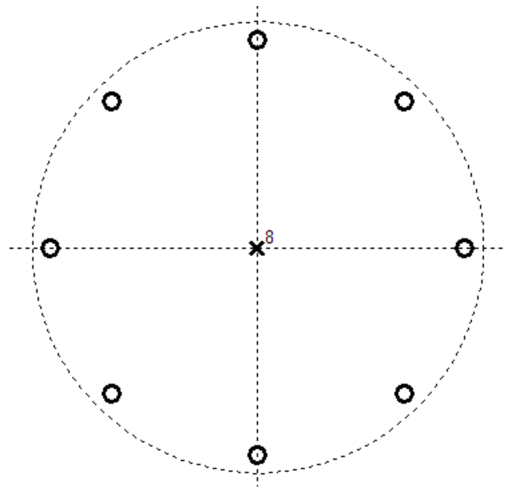
Il termine $(1+\alpha^2)$ è una costante, mentre il termine $2\alpha cos K\omega$ varia periodicamente.



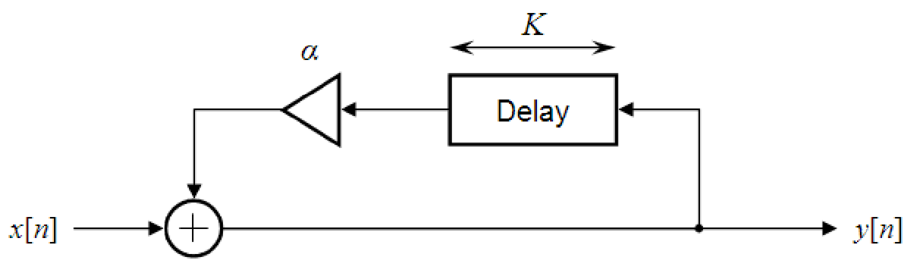
Volendo dare un'interpretazione polare al grafico, consideriamo l'equazione:

$$H(z) = \frac{z^K + \alpha}{z^K}$$

Essa ha zeri solo in $z^K = -\alpha$. Quest'ultima equazione ha K soluzioni equispaziate attorno ad un cerchio nel piano complesso. Il denominatore è zero quando $z^K = 0$, ottenendo K poli per $z = 0$. Questo porta a grafici di poli e zeri come il seguente (per $K = 8$ ad esempio):



La struttura di un filtro comb a retroazione è la seguente:



Il filtro si descrive attraverso la seguente equazione alle differenze:

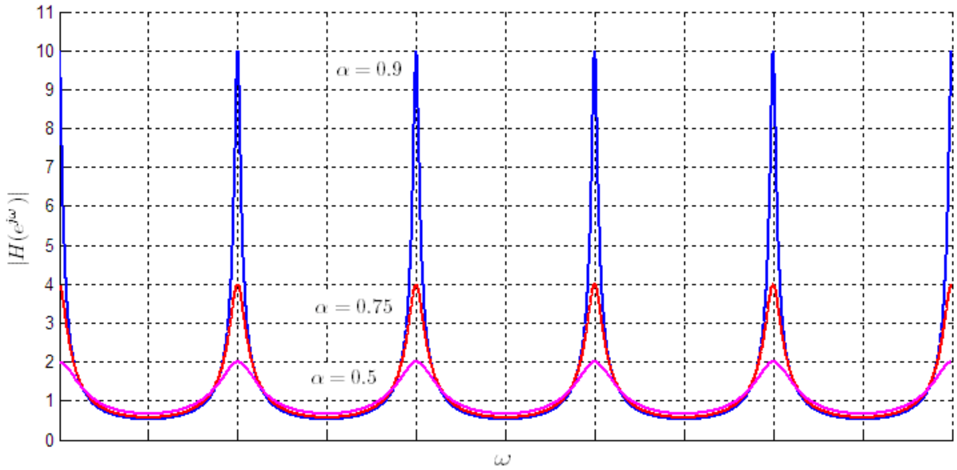
$$y[n] = x[n] - \alpha y[n-K]$$

Manipolando questa equazione in modo che tutti i termini in y vengono portati dalla parte sinistra e applicando la trasformata z si ottiene:

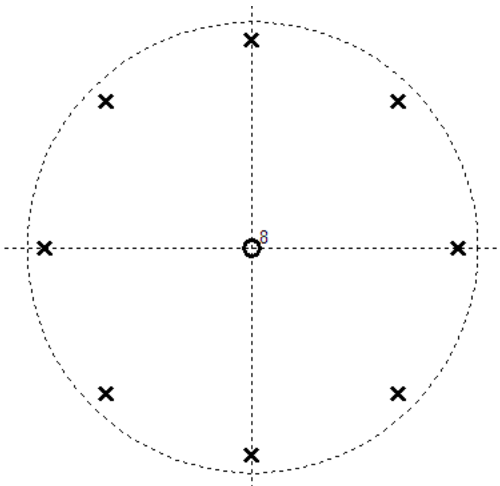
$$H(z) = \frac{z^K}{z^K - \alpha}$$

Applicando la sostituzione del caso precedente e considerando il modulo di $H(e^{i\omega})$ otteniamo:

$$|H(e^{i\omega})| = \frac{1}{\sqrt{(1+\alpha^2) - 2\alpha cos K\omega}}$$



Il numeratore, questa volta, vale zero per $z^K = 0$ K zeri per $z = 0$. Il denominatore vale zero ogni volta che $z^K = \alpha$. Questo ha K soluzioni, equamente spaziate attorno ad un cerchio nel piano complesso, i poli della funzione di trasferimento. Questo porta a dei grafici di poli e zeri come il seguente (per $K = 8$ ad esempio):



È possibile ottenere da un filtro comb un filtro passa-basso semplicemente eliminando lo zero $z_0 = 1$.

Capitolo 5

Progetti di filtri digitali

Un filtro digitale è un sistema LTI a tempo discreto, realizzato con aritmetica a precisione finita. La progettazione di tali filtri richiede dunque tre passi principali:

1. Specificazione delle proprietà desiderate del filtro, ad esempio la frequenza di taglio per un filtro passa-basso; poiché i filtri ideali non sono realizzabili, ci dovremo tuttavia accontentare di un'approssimazione e la specifica dovrà riguardare il livello di errore che riteniamo di poter tollerare.

2. Determinazione di un filtro che soddisfa le specifiche stesse; nel caso di un filtro FIR basterà ad esempio determinare i coefficienti che definiscono la sua risposta all'impulso, in un filtro IIR basterà determinare i coefficienti dell'equazione alle differenze finite che lo caratterizzano.

3. Realizzazione del sistema con una rete a precisione finita, con eventuale implementazione su un DSP o su un circuito programmabile.

5.1 Specifiche di filtri digitali

Le proprietà di un filtro digitale sono generalmente ben esprimibili nel dominio delle frequenze. In particolare, per filtri selettivi come filtri passa-basso o passa-banda, le specifiche possono essere date attraverso uno schema di tolleranza.
Gli elementi principali di uno schema di tolleranza esemplificati per il filtro passa-basso si riferiscono al guadagno e sono:

1. Le frequenze ω_{bp} e ω_{bs} che definiscono condizioni di accettabilità per la banda passante e la banda di transizione; queste condizioni possono essere espresse come segue:

$$\omega_{bp} \leq \omega_c < \omega_s \leq \omega_{bs}$$

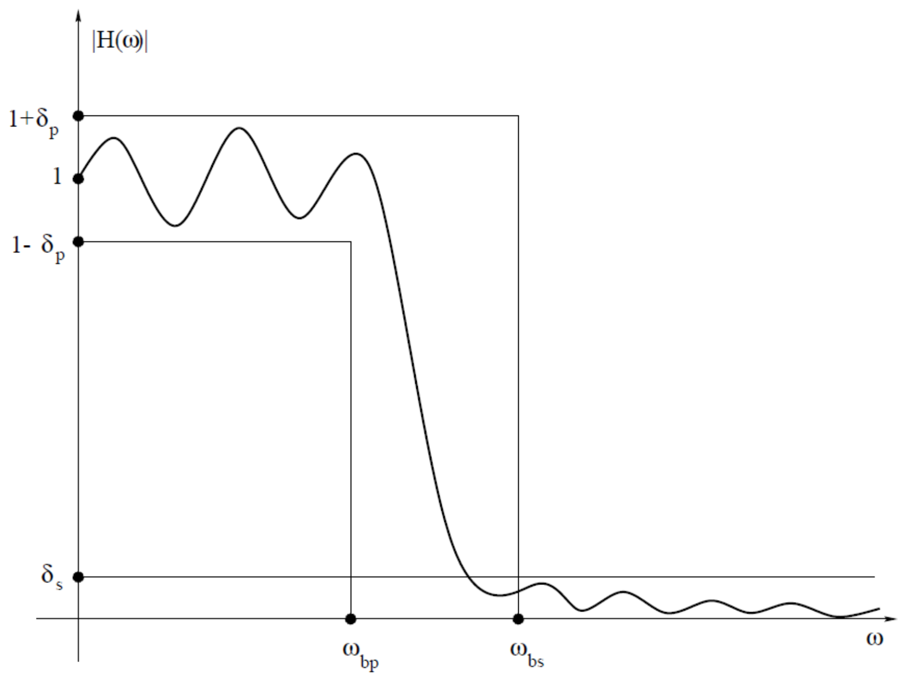
   dove ω_c è la frequenza di taglio e ω_s è la frequenza di stop del filtro.

2. Le dimensioni massime δ_p e δ_s permesse alle oscillazioni rispettivamente in banda passante e in banda proibita; esse sono usualmente chiamate deviazioni. Le deviazioni in banda passante e proibita possono essere espresse anche in decibel:

$$A_p = 20 \log \frac{1 + \delta_p}{1 - \delta_p}$$

$$A_s = -20 \log \delta_s$$

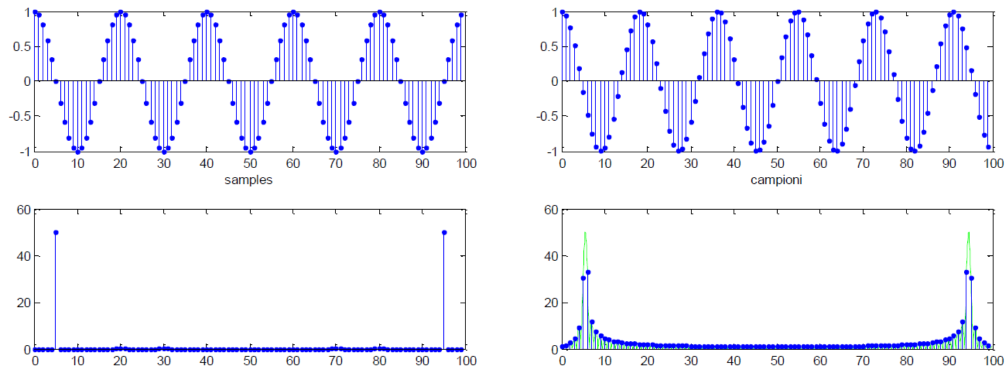
   dove A_p denota l'oscillazione in banda passante, mentre A_s denota l'attenuazione in banda proibita.



Tutte le frequenze specificate devono essere riferite alla frequenza di campionamento ω_0 ed in particolare devono essere inferiori a $\frac{\omega_0}{2}$ così da soddisfare il criterio di Nyquist.

5.2 Progetto di FIR tramite finestre

La tecnica di progettazione mediante finestre è basata sull'idea di approssimare un filtro desiderato, eventulmente non causale e con risposta all'impulso $h_d(n)$ di durata infinita, azzerando $h_d(n)$ al di fuori di una finestra temporale di ampiezza N, nella speranza che l'approssimazione sia tanto più buona quanto più la dimensione N della finestra è grande.
Una delle premesse dell'analisi di Fourier è che il segnale analizzato deve essere un periodo esatto di una forma d'onda periodica. Questo in generale non è vero. Il mancato raccordo fra ultimo campione del blocco analizzato e primo campione del blocco idealmente successivo (identico a quello analizzato, visto che il segnale è assunto periodico), causa un "click", che si traduce in uno spettro contaminato da rumore a larga banda ("leakage").



In particolare, fissato un intero N, si costruisce il filtro FIR con risposta all'impulso $h_N(n)$ tale che:

$$h_N(n) = \begin{cases} h_d(n), & \text{se } |n| \leq \frac{N-1}{2} \\ 0, & \text{altrimenti} \end{cases}$$

Esistono vari tipo di finestre. Iniziamo considerando la finestra rettagolare $rett_N(n)$:

$$rett_N(n) = \begin{cases} 1, & \text{se } |n| \leq \frac{N-1}{2} \\ 0, & \text{altrimenti} \end{cases}$$

possiamo scrivere $h_N(n) = rett_N h_d(n)$.
In altre parole, $h_N(n)$ è ottenuta moltiplicando la risposta all'impulso del filtro che si desidera approssimare per la finestra rettangolare $rett_N(n)$ di durata finita.
Più in generale possiamo scrivere:

$$h_N(n) = \omega_N(n) h_d(n)$$

e dal teorema di convoluzione si ottiene

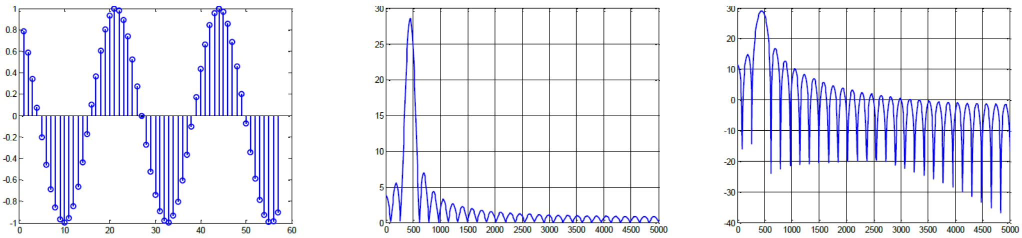
$$H_N(e^{i\omega}) = H_d(e^{i\omega}) * \Omega_N(e^{i\omega})$$

dove $H_d(e^{i\omega})$ è la risposta in frequenza di un filtro desiderato e $\Omega_N(e^{i\omega})$ è la risposta in frequenza della finestra $\omega_N(n)$.

In realtà il problema del troncamento, nel dominio del tempo, è il prodotto della sequenza da esaminare per una finestra rettangolare. Per effetto della proprietà del prodotto nel dominio del tempo abbiamo, nel dominio della frequenza, la convoluzione dello spettro del segnale con quello di una finestra rettangolare. Quest'ultimo è del tipo *sinc*, cioè $\frac{sin(f)}{f}$:

$$RETT_N(\omega) = \sum_{n=0}^{N} e^{i\omega n} = \frac{1 - e^{i\omega N}}{1 - e^{i\omega}} = e^{-i\omega \frac{N-1}{2}} \frac{\sin\left(\omega \frac{N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

Sono proprio i lobi laterali del *sinc* ad introdurre le frequenze addizionali.
In figura è rappresentato un segnale con relativo spettro in frequenza



Se però l'intervallo di troncamento viene scelto come un multiplo del periodo, il campionamento nel dominio delle frequenze coincide con gli zeri della funzione $fracsin(f)f$ e pertanto il risultato della trasformata discreta di F non appare alterato.

L'operazione che viene effettuata per ridurre il leakage consiste nel moltiplicare il segnale per una finestra che vada a zero ai bordi e che nel dominio delle frequenze abbia lobi laterali di ampiezza minore di quelli della finestra rettangolare. Minore è l'ampiezza dei lobi laterali, minore è il leakage sulla DFT.
Le finestre, in generale, sono caratterizzate da tre parametri:

- Il fattore di leakage – il rapporto tra la potenza nei lobi laterali e la potenza totale della finestra.
- L'attenuazione relativa del lobo laterale – differenza in altezza tra il picco del lobo principale e il picco laterale più alto.
- Larghezza del lobo principale (-3dB) – larghezza del lobo principale a 3 dB al di sotto del picco del lobo principale

Esempi di finestre comunemente usate oltre alla finestra rettangolare sono:

Finestra di Hanning:

$$\omega_N(n) = \frac{1}{2}\left[1 + \cos\frac{2\pi n}{N}\right]$$

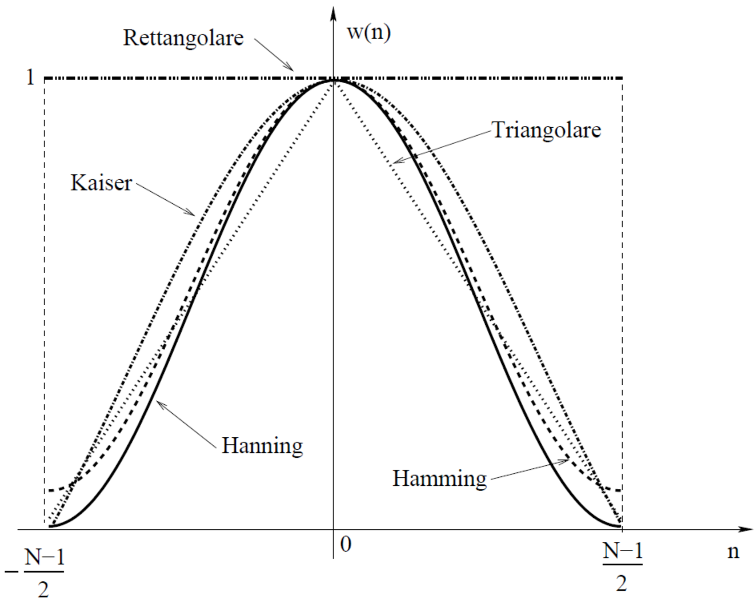
Finestra Triangolare (o Bartlett):

$$\omega_N(n) = \begin{cases} 1 + \frac{2n}{N}, & \text{se } -\frac{N-1}{2} \leq n \leq 0 \\ 1 - \frac{2n}{N}, & \text{se } 0 \leq n \leq \frac{N-1}{2} \end{cases}$$

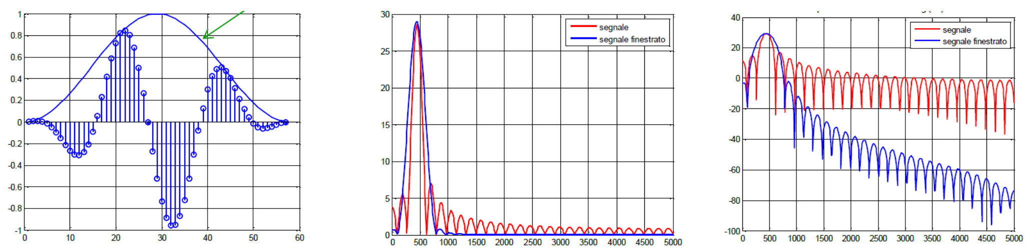
Finestra di Hamming:

$$\omega_N(n) = 0.54 + 0.46 \cos \frac{2\pi n}{N}$$

Kaiser ha inoltre proposto una famiglia di finestre $k_N(n, \omega_a)$, parametrizzata da ω_a. La Figura illustra il grafico delle finestre sopra elencate.



In figura è rappresentato un segnale finestrato mediante finestra di Hanning e relativo spettro. Il grafico in rosso rappresenta il segnale finestrato con finestra rettangolare:



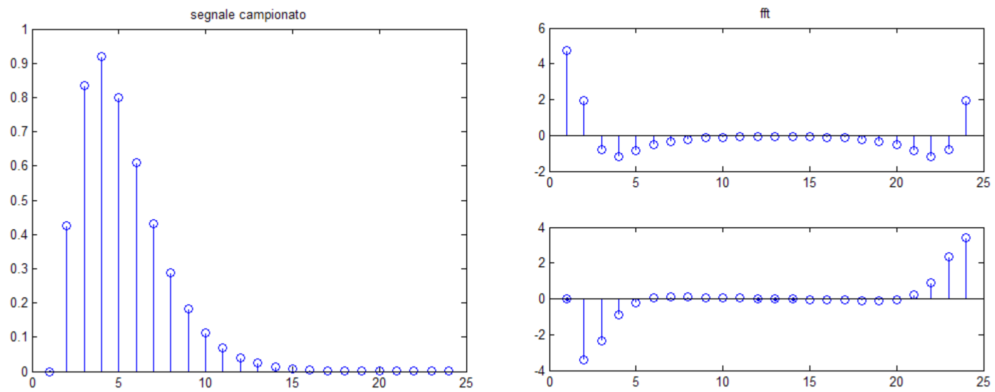
La progettazione mediante finestre richiede di operare due scelte:

1. il tipo di finestra;

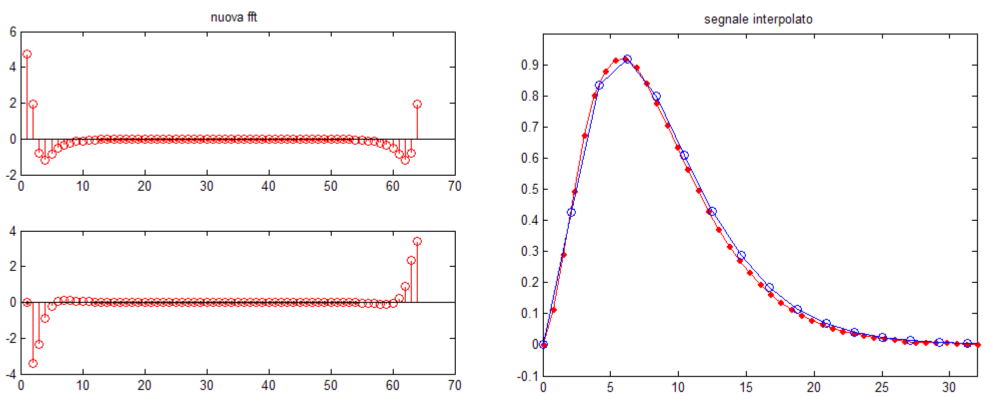
2. l dimensione N dell'intervallo.

5.3 Zero padding

Talvolta per migliorare il profilo della DFT vengono inseriti zeri nel segnale da esaminare. In generale lo zero padding nel dominio della frequenza aumenta i punti nel dominio del tempo (interpolazione):



Zero padding con 40 zeri al centro della DFT:



Il motivo più comune dell'utilizzo dello zero padding è fare in modo che una forma d'onda abbia un numero di campioni pari a una potenza di 2, ad esempio 4096, 8192, 16384, etc. Quando la lunghezza del dominio del tempo di una forma d'onda è una potenza di due, gli algoritmi FFT, estremamente efficienti, possono essere utilizzati per accelerare i tempi di elaborazione.

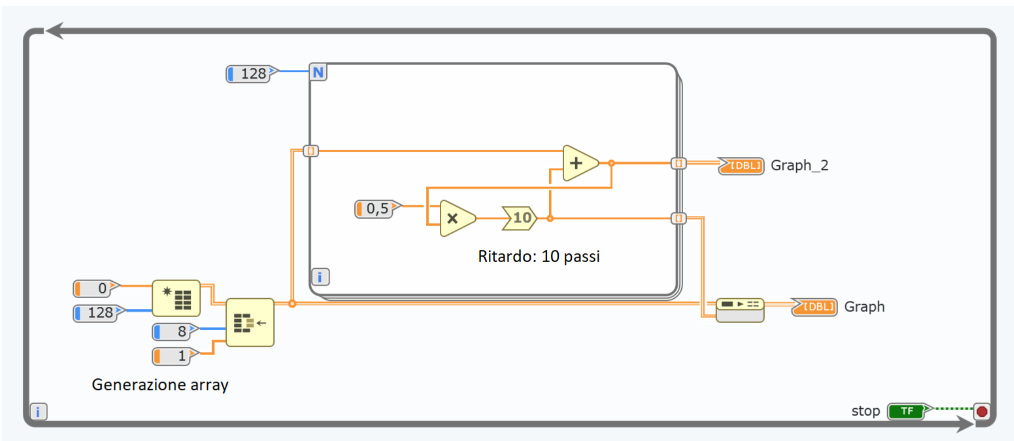
Capitolo 6

Realizzazione di un filtro COMB tramite LabviewNXG

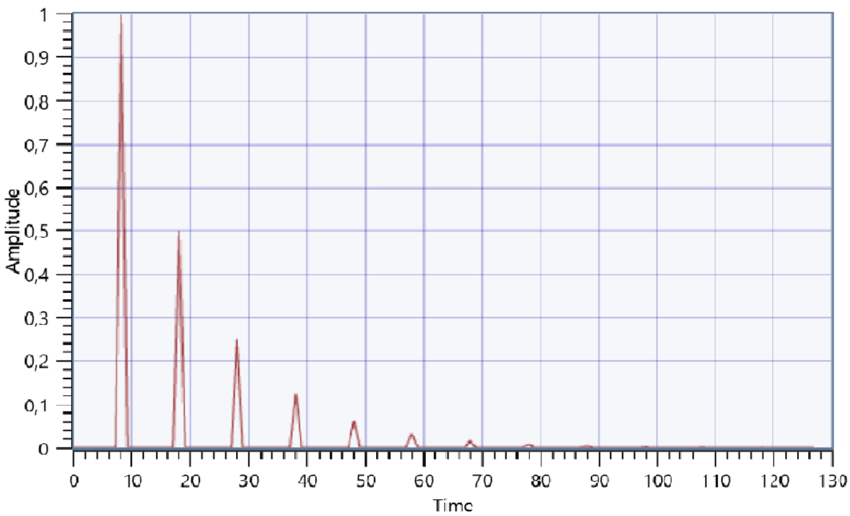
In questo lavoro di tesi si è progettato un filtro digitale COMB a retroazione tramite l'utilizzo del software LabviewNXG.

6.1 Filtro COMB per un array

Inizialmente è stato realizzato un diagramma in cui una funzione somma, una funzione moltiplicazione e una funzione delay operano su di un array di 128 elementi utilizzati come campione di un segnale nel quale è stato posto un unico elemento a valore 1 e tutti gli altri a 0. L'obiettivo è stato costruire pertanto uno schema che rappresenti un filtro COMB, come mostrato in figura:

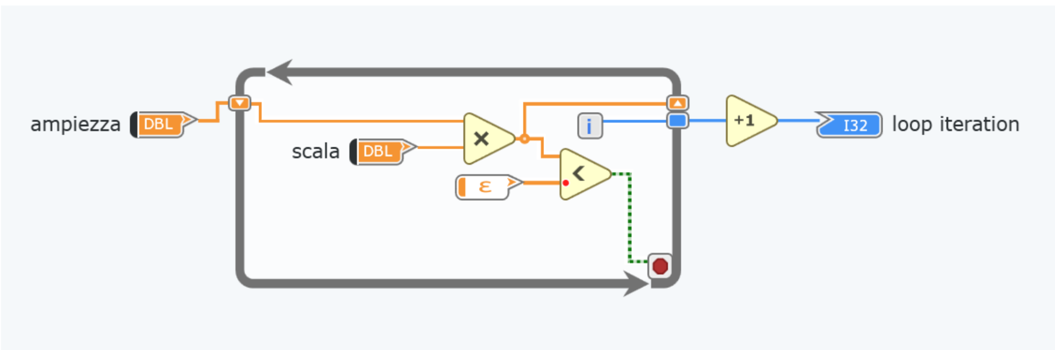


Si è realizzato un ciclo *while* con annessa funzione di stop al cui interno è stato inizialmente creato l'array di 128 elementi, i quali sono entrati uno alla volta in un ciclo for. In uscita al ciclo è stato posto il grafico che rappresenta il segnale in uscita (impulso più tutti i suoi ritardi).



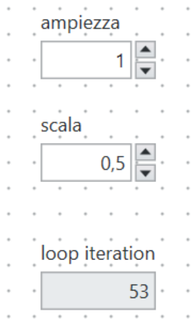
6.2 Contatore di loop

Il filtro realizzato è teoricamente un IIR perchè il segnale viene ritardato e ridotto all'infinito. Nella pratica però il filtro risulta essere un FIR dato che dopo alcune iterazioni i valori dei campioni che andremo ad inviare all'hardware sono compatibili con il rumore di fondo, e dunque non più significativi. Un'indicazione di questo limite nel numero di iterazioni significativo, ci può essere dato dal seguente calcolo:



in cui si va a verificare dopo quante iterazioni un numero ridotto di una "scala" < 1, divenga non più calcolabile dalla ALU nella rappresentazione "floating point" in uso. La costante ϵ rappresenta il valore minimo calcolato dalla macchina. Il ciclo *while* si interrompe appena il risultato della moltiplicazione risulta essere minore dell'epsilon macchina.

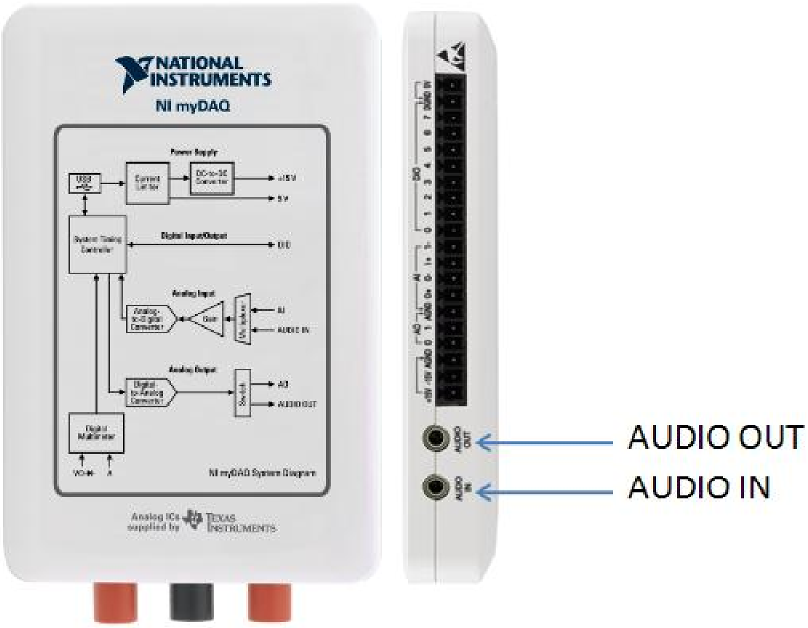
Infine, in uscita dal ciclo *while* è presente un contatore di loop, al quale è stata aggiunta una funzione $+1$ in quanto il contatore parte da 0.
L'interfaccia con l'utente è la seguente:



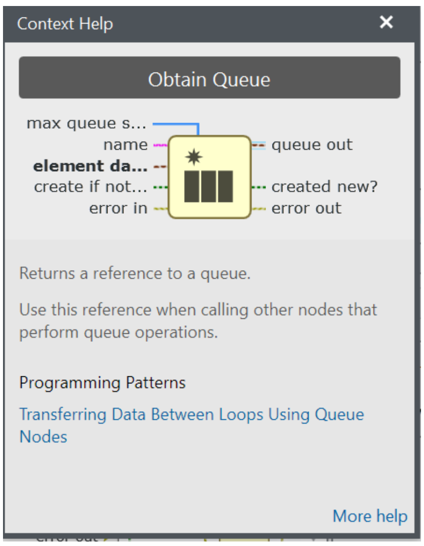
Dunque, ad esempio, per un'ampiezza campione pari a 1 e un fattore di scala pari a $0,5$, il filtro IIR risulta essere in realtà un filtro FIR il cui limite massimo M dei coefficienti è 53.

6.3 Filtro COMB per un segnale audio

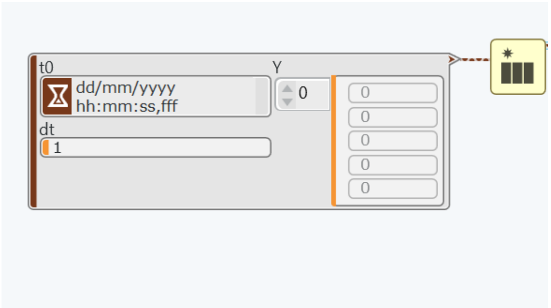
Successivamente, tramite l'utilizzo della NI myDaq della National Instruments è stato digitalizzato un segnale audio analogico, processato tramite un filtro COMB e scritto sul medesimo hardware; ciò è stato possibile tramite funzioni di scrittura e lettura del segnale (che analizzeremo in questo paragrafo) unite al filtro COMB creato in precedenza.
Inizialmente è stato collegato l'ingresso AUDIO IN ad un dispositivo che riproducesse un segnale audio e l'ingresso AUDIO OUT ad un altoparlante.



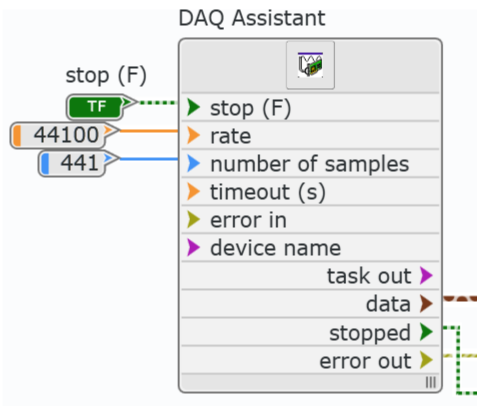
Prima di utilizzare le funzioni di lettura e scrittura è utile definire una coda. Creando una coda di dati un ciclo può elaborare un volume maggiore di dati a intervalli più regolari senza perdere informazioni. Ciò è particolarmente utile se i dati provenienti da più canali devono essere elaborati nell'ordine in cui arrivano. La funzione utilizzata per questo scopo è la seguente:



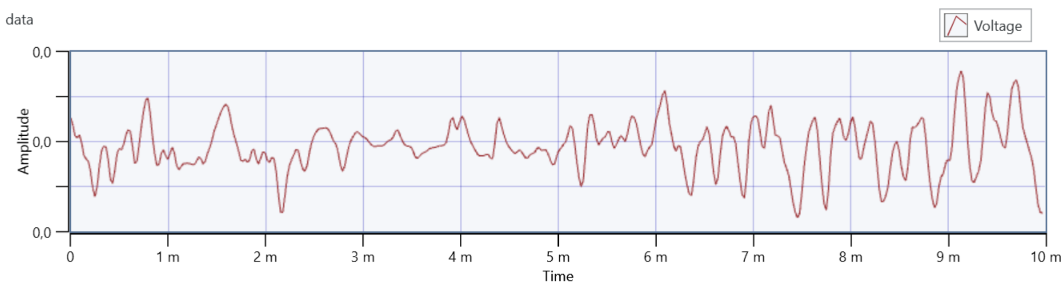
In particolare è stato comunicato alla precedente funzione il tipo di dati da mettere in coda, che nel nostro caso sono del tipo "forma d'onda", che consiste sostanzialmente in un array di valori (campionati) e le relative informazioni temporali (frequenza di campionamento):



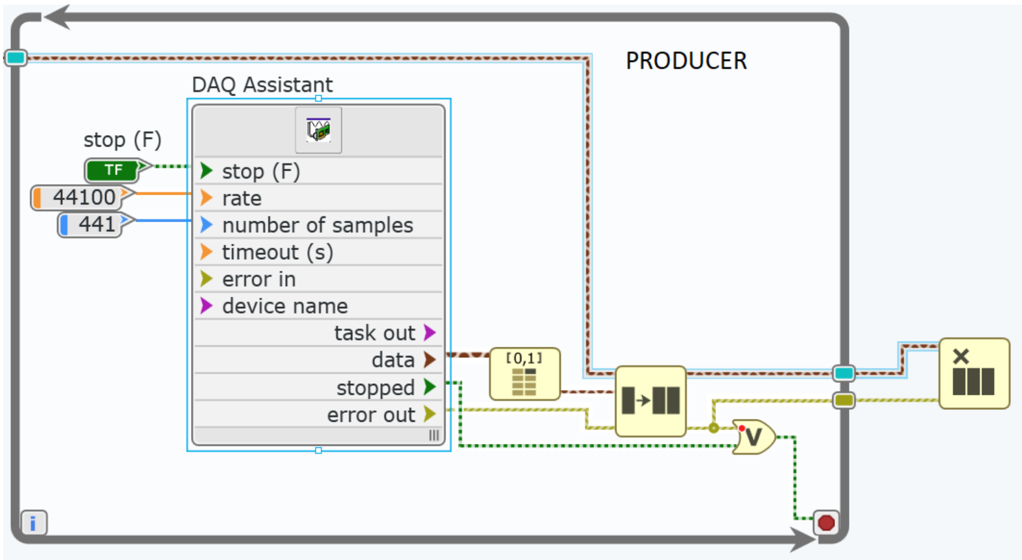
Le funzioni di scrittura e lettura permettono di stabilire il tasso di campionamento del segnale e il numero di campioni mostrati nel grafico (ampiezza in funzione del tempo). In questo caso è stato utilizzato un tasso di campionamento di 44100 Hz (frequenza di campionamento audio CD) e si è scelto di trattare 441 campioni alla volta, pari a 10ms di finestra temporale.



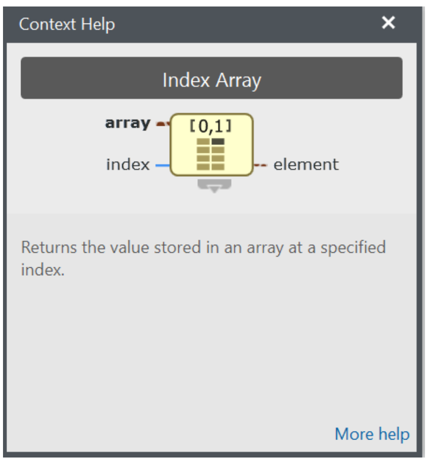
In questo modo otteniamo una finestra temporale di 0,01s.



Le funzioni di lettura e scrittura sono inserite in due cicli *while* che lavorano parallelamente. In realtà i dati in ingresso nel secondo ciclo *while* (contenente la funzione di scrittura) provengono dal primo ciclo *while* (contenente la funzione di lettura), quindi le azioni non sono effettivamente parallele. Tuttavia l'utilizzo di due cicli separati permette di ottimizzare le operazioni.
Questo tipo di architettura software è nota generalmente come "PRODUCER-CONSUMER".
La funzione di lettura è posta all'interno di un ciclo *while* e presenta uno *stop* in ingresso che permette all'utente di interrompere tutte le operazioni.

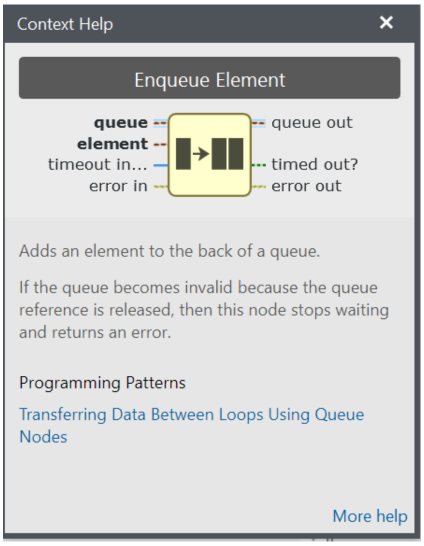


I dati uscenti dalla funzione di lettura raggiungono la seguente funzione:



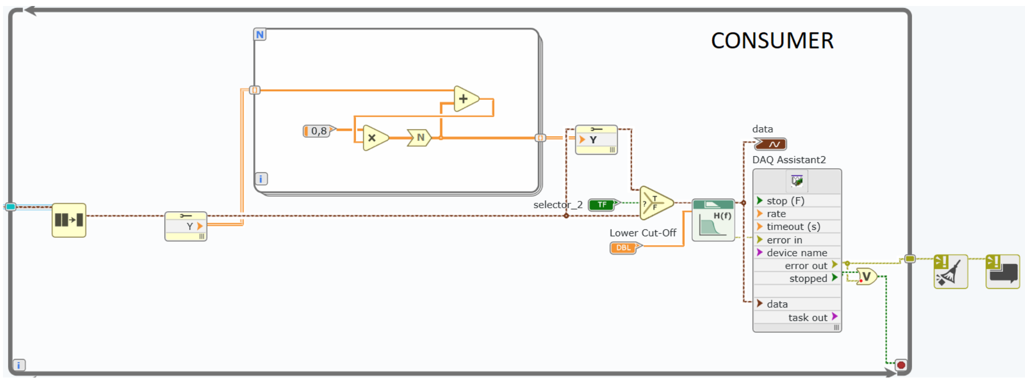
Essa permette di estrarre i dati di un canale. Nel nostro caso l'indice è sottinteso (0) in quanto abbiamo un solo segnale in uscita.

Successivamente i dati raggiungono la seguente funzione

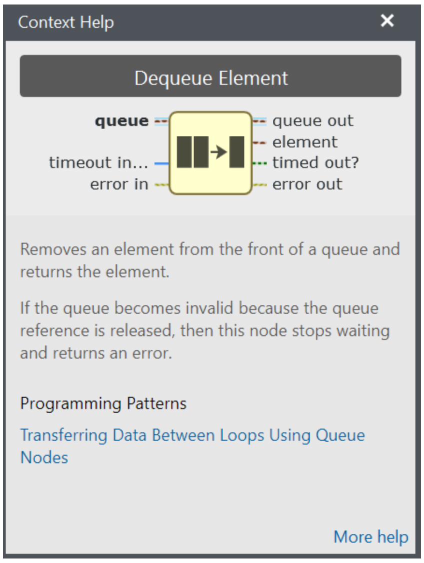


che li mette in coda.

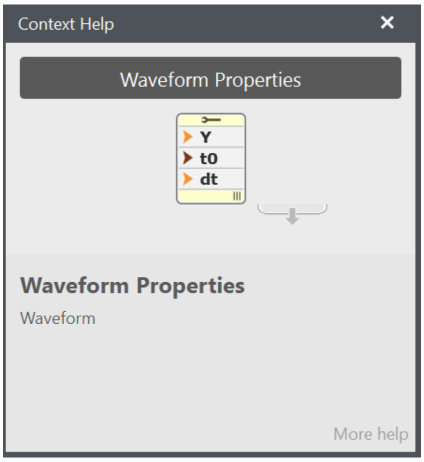
Usciti dal primo *while* i dati entrano in un secondo *while*:



I dati della forma d'onda vengono inizialmente rimossi dalla coda tramite la seguente funzione:



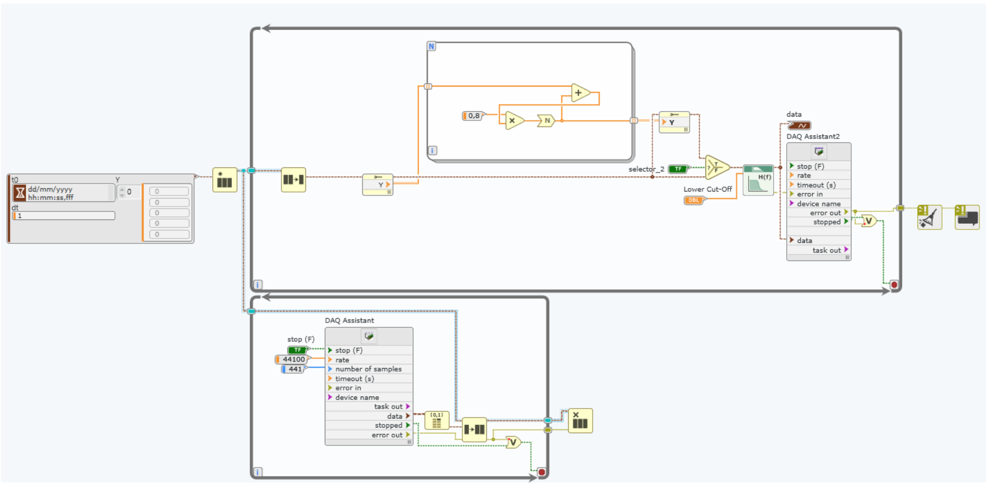
Il primo elemento di questo secondo *while* è un oggetto di questo tipo:



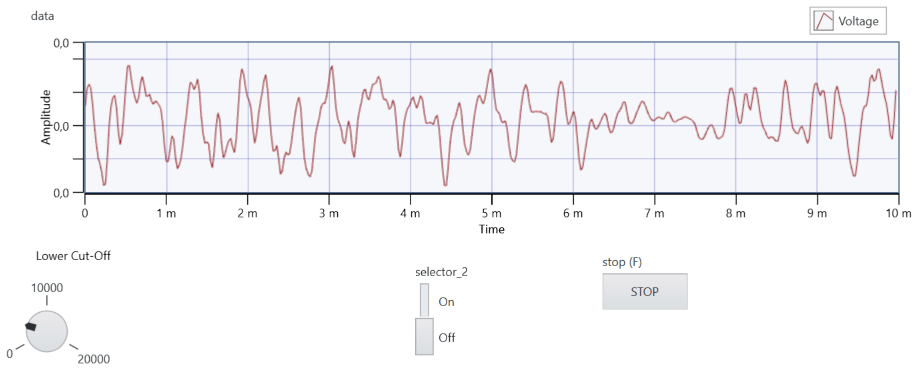
che permette di estrarre i singoli elementi della forma d'onda. Nel nostro caso è stato necessario estrarre solamente i valori di ampiezza del segnale, i quali raggiungono un ciclo for che ne causa l'effetto delay (analogo al ciclo descritto in precedenza).
Al termine di questo ciclo vi è una funzione analoga alla Waveform Properties che in questo caso scrive i nuovi valori di ampiezza del segnale (processati dal delay) e li unisce ai valori dei tempi per ricostruire una forma d'onda.
A questo punto un selettore permette all'utente di scegliere se applicare l'effetto delay al segnale o semplicemente riprodurlo invariato.

Il segnale uscente dal selettore raggiunge un filtro passa basso (che serve ad attenuare i rumori di campionamento). Esso viene poi graficato e raggiunge una funzione di scrittura con rate di campionamento analogo a quello della funzione di lettura.
Infine, fuori dal secondo ciclo *while*, vi sono due funzioni dedicate alla gestione degli errori.

Lo schema completo è il seguente:

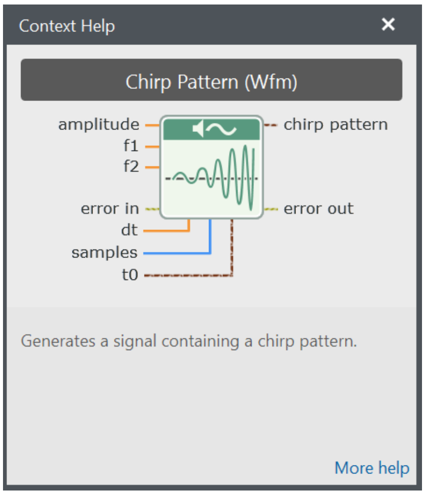


L'interfaccia con l'utente presenta il grafico del segnale in uscita, un selettore che permette di stabilire se si vuole ascoltare il segnale originale o un segnale con delay. Infine vi è il pulsante di *stop*.



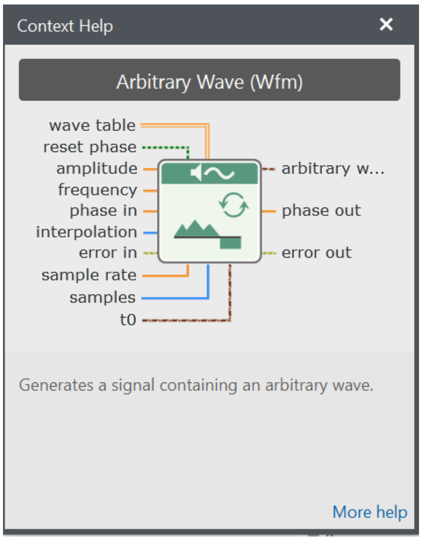
6.4 Filtro COMB per un segnale chirp

Allo schema precedente è stato aggiunto uno *switch* nel *while* contenente la funzione di lettura in modo da dare la possibilità all'utente di scegliere se processare il segnale audio proveniente dall'ingresso AUDIO IN oppure il segnale *chirp* generato da un'apposita funzione di LabVIEW NXG e quindi poter verificare il funzionamento del delay con un segnale audio molto semplice.
Per la realizzazione del segnale "*chirp*" è stata utilizzata una funzione CHIRP PATTERN che presenta le seguenti caratteristiche:

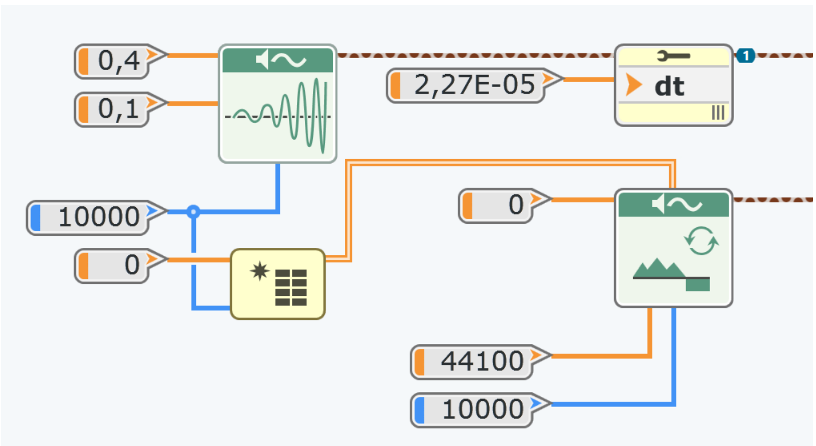


In particolare, è stata fornita al segnale un'ampiezza di $0,4V$, una frequenza iniziale (f1) di 0 (in unità normalizzate di cicli/campione) e una frequenza finale (f2) di $0,1$. Inoltre gli sono stati forniti 10000 campioni.

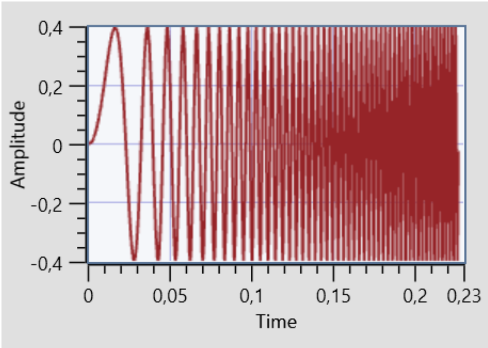
Insieme alla funzione CHIRP PATTERN è stata utilizzata anche una fuzione ARBITRARY WAVE con le seguenti caratteristiche



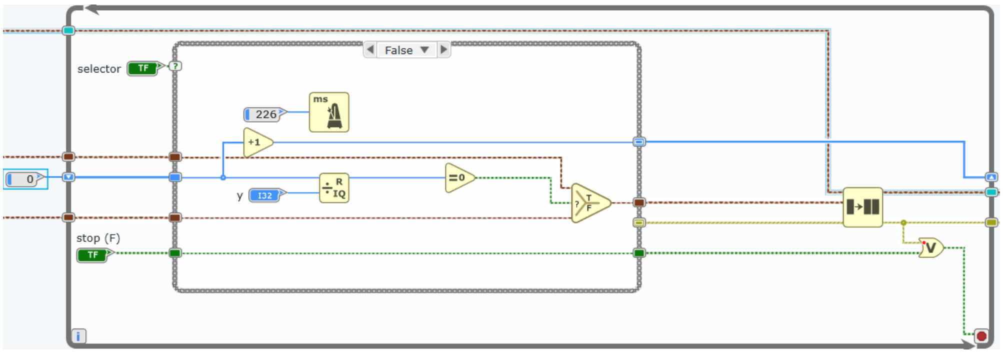
alla quale è stato inviato array di 10000 elementi uguali a 0. Il tasso di campionamento è stato impostato a 44100 Hz come nel segnale audio. L'utilità di questa funzione è quella di permettere ai chirp di intervallarsi con momenti di silenzio.
Successivamente è stata utilizzata una funzione WAVEFORM PROPERTIES per rendere il *dt* del chirp compatibile con quello del segnale audio.



La funzione generata dal CHIRP PATTERN è la seguente:



I due segnali generati entrano quindi nel primo *while* al cui interno è inserita una funzione di *switch* in cui alla voce "True" è associata la funzione di lettura descritta in precedenza, alla voce "False" è associato il seguente diagramma:

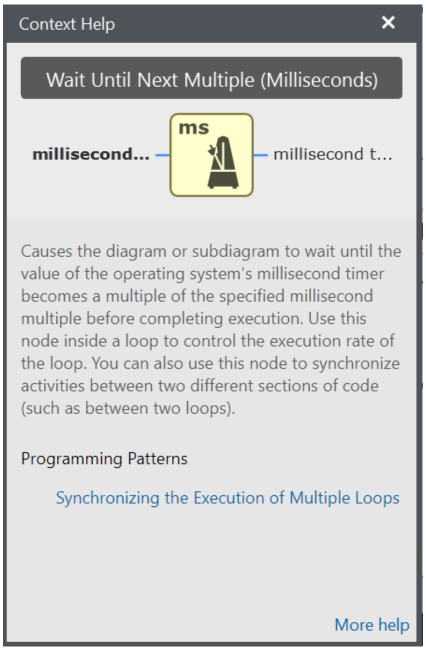


All'interno del ciclo *for* è presente un selettore che permette di passare dalla funzione chirp alla funzione costantemente uguale a 0. Il criterio tramite il quale agisce questo selettore è il seguente:

- all'esterno del ciclo *while* vi è una costante pari a 0, collegata al ciclo *while* tramite uno SHIFT REGISTER (dopo l'iterazione del ciclo iniziale, il registro a scorrimento sinistro nella coppia restituisce il valore che riceve dal registro a scorrimento destro dall'iterazione precedente).
- una volta entrata nello SHIFT REGISTER, la costante viene divisa per un numero (intero) stabilito dall'utente. Il resto di questa divisione entra in una funzione che lo "confronta con 0".
- se il resto è diverso da zero viene riprodotta la funzione costantemente uguale a 0, altrimenti viene riprodotta la funzione chirp.

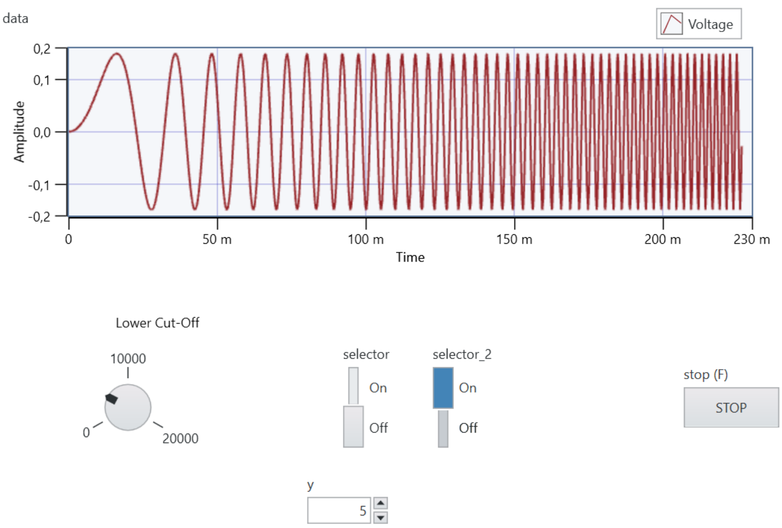
In questo modo è possibile stabilire la frequenza di ripetizione del chirp.

All'interno dello *switch* vi è infine una funzione di questo tipo



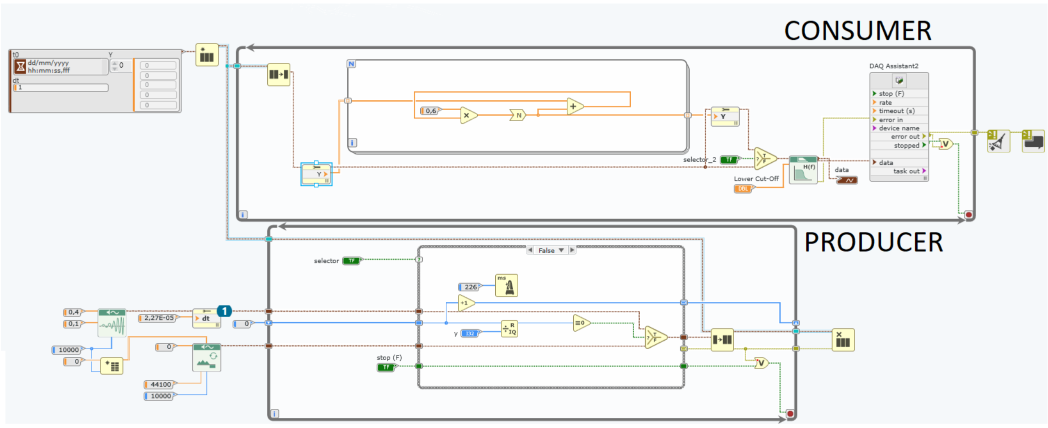
che permette di porre il tempo di durata del loop a $226ms$ (che è proprio il tempo di durata di un chirp poichè abbiamo 10000 campioni e una frequenza di campionamento di $44100Hz$).

L'iterfaccia con l'utente è la seguente:



Il primo selettore permette di scegliere se utilizzare il segnale chirp o il segnale audio provenite dall'ingresso AUDIO IN, il secondo selettore permette di scegliere se utilizzare o meno il delay.

Lo schema complessivo è il seguente:



Capitolo 7

Appendice

7.1 Traformata z

In analisi funzionale la trasformata zeta è una trasformata integrale che permette di trasformare una funzione discreta in una funzione più semplice, utilizzata principalmente nella teoria dei segnali.
Sia $x[n]$ una successione di numeri complessi, indicizzata con $n \in N$. La sua trasformata z è definita come la serie formale di potenze complesse

$$X(z) = Z\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n}$$

In teoria dei segnali questa definizione è utilizzata per valutare la trasformata della risposta all'impulso unitario di un sistema causale tempo-discreto. Solitamente, in tale ambito la successione $x[n]$ rappresenta il campionamento regolare di un segnale $f : R \to C$ causale, in corrispondenza dei tempi della forma $t = n\tau$. Il passo di campionamento $\tau > 0$ è fissato.
Tra le varia proprietà delle trasformate z vi è quella della traslazione temporale. Sia $x[n-k]$, allora

$$Z\{x[n-k]\} = \sum_{n=0}^{\infty} x[n-k] z^{-n}$$

Posto $j = n - k$ (con $x[j] = 0$ se $j < 0$) si ha:

$$\sum_{n=0}^{\infty} x[n-k] z^{-n} = \sum_{j=0}^{\infty} x[j] z^{-(j+k)} = \sum_{j=0}^{\infty} x[j] z^{-j} z^{-k} = z^{-k} \sum_{j=0}^{\infty} x[j] z^{-j} = z^{-k} X(z)$$

L'espressione della trasformata inversa, che può essere ottenuta utilizzando il teorema integrale di Cauchy, è la seguente:

$$x[n] = Z^{-1}X(z) = \left(\frac{1}{2\pi j}\right) \oint_C X(z)z^{n-1}dz$$

Un caso di particolare importanza si presenta quando C è la circonferenza unitaria. In tal caso la trasformata zeta inversa assume la forma della trasformata di Fourier discreta inversa:

$$x[n] = Z^{-1}X(z) = \left(\frac{1}{2\pi}\right) \int_{-\pi}^{+\pi} X(e^{j\omega})e^{j\omega n}d\omega$$

7.2 Teorema di Paley-Wiener

Il teorema di Paley-Wiener è una relazione matematica che consente di determinare se un sistema lineare tempo invariante è causale o meno e afferma che se una sequenza $h(n)$ è nulla per $n < 0$ e ha energia finita, cioè $\sum_{n=0}^{\infty} |h(n)|^2 < \infty$, allora la sua trasformata di Fourier verifica:

$$\int_{-\pi}^{+\pi} |ln|H(e^{i\omega})||d\omega < \infty$$

Viceversa, se $|H(e^{i\omega})|$ è quadrato integrabile, cioè

$$\int_{-\pi}^{+\pi} |H(e^{i\omega})|^2 d\omega < \infty$$

e se la condizione di Paley-Wiener è rispettata, allora è sempre possibile associare al modulo di $H(e^{i\omega})$ una risposta in fase $\phi(\omega)$ in modo tale che il filtro con risposta in frequenza

$$H(e^{i\omega}) = |H(e^{i\omega})|e^{i\phi(\omega)}$$

sia causale. Un'importante conseguenza di questo risultato è che un filtro causale ha un risposta in frequenza che non può essere nulla su una banda finita, il che conferma la non causalità del filtro ideale.

Conclusioni

In questo lavoro di tesi sono stati trattati alcuni concetti fondamentali nella teoria dei segnali e sono stati analizzati i processi di conversione analogico-digitale tramite campionamento e quantizzazione.
Successivamente sono stati descritti brevemente i DSP per quanto riguarda la loro struttura.
Infine si è parlato di filtri digitali (particolare di filtri COMB) e di alcune tecniche di progettazione degli stessi.
Ciò che si è realizzato in questo lavoro di tesi è un filtro COMB a retroazione tramite LabVIEW NXG. Il filtro realizzato è teoricamente un IIR ma risulta essere nella pratica un FIR data la limitatezza della capacità di calcolo della macchina. Il filtro ottenuto permette di riprodurre un segnale audio proveniente dall'ingresso AUDIO IN, dando all'utente la possibilità di scegliere se applicare l'effetto del delay o meno. Esso inoltre, con l'aggiunta di una funzione chirp, consente all'utente di ascoltare, in alternativa al segnale audio proveniente dall'ingresso AUDIO IN, un segnale squillante di breve durata, utile a verificare il funzionamento del delay in modo più semplice e diretto.

Bibliografia

[1] J. D. Gibson. *Principles of Digital and Analog Communications.* Macmillan, second edition,1993.

[2] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice Hall, 1975.

[3] A. V. Oppenheim and R. W. Schafer. *Signals and Sistems.* Prentice Hall, second edition, 1997.

[4] C. L. Phillips, J. M. Parr, and E. A. Riskin. *Signals, Systems, and Transforms.* Prentice Hall, third edition, 2003.

[5] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing. Principles, Algorithms, and Applications.* Pearson Prentice Hall, fourth edition, 2007.

[6] S. V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction.* Wiley and Teubner, 1996.