

Esercizi obbligatori su serie di Fourier e Python

francesco.fuso@unipi.it; <http://www.df.unipi.it/~fuso/dida>

(Dated: version 4 - FF, 28 novembre 2015)

Questa nota propone alcuni esercizi sullo sviluppo in serie di Fourier che dovete svolgere, da soli o in gruppo, con Python. Prima del testo degli esercizi, si dà una rapida occhiata al background matematico e alle definizioni necessarie. Inoltre questa nota riporta anche alcuni hints per l'esecuzione degli esercizi e il risultato trovato da me, che può essere utile come referenza.

I. SVILUPPO IN SERIE DI FOURIER

Come si dimostra (in matematica), una *qualsiasi* funzione periodica $g(t)$, con periodo $T = 1/f = 2\pi/\omega$, può essere espressa come sovrapposizione lineare di una costante e di funzioni armoniche di pulsazione ω e $\omega_k = k\omega$ (dette rispettivamente armonica fondamentale e armoniche superiori di ordine k). Esistono vari modi per esprimere l'operazione, che dipendono fondamentalmente dall'impiego di grandezze reali o complesse. Per i nostri scopi il modo migliore consiste nell'usare una sovrapposizione (serie) di funzioni seno e coseno:

$$g(t) = \frac{a_0}{2} + \sum_{k=1}^n b_k \cos(\omega_k t) + \sum_{k=1}^n c_k \sin(\omega_k t), \quad (1)$$

dove $n \rightarrow \infty$ e il coefficiente a_0 tiene conto del valore medio della funzione (è nullo per funzioni alternate, che sono quelle qui considerate). I *coefficienti di Fourier* b_k e c_k si ricavano dalle espressioni

$$b_k = \frac{2}{T} \int_0^T g(t) \cos(\omega_k t) dt \quad (2)$$

$$c_k = \frac{2}{T} \int_0^T g(t) \sin(\omega_k t) dt. \quad (3)$$

Non è di certo questa la sede per entrare nel merito della tanta matematica che è coinvolta nell'argomento (definizioni, dimostrazioni, applicazioni, alcune strepitosamente importanti); mi piace però ricordare che, in qualche modo, l'approccio utilizzato, che serve a sviluppare su una base (volendo, ortonormale) una funzione *periodica*, può essere considerato un caso limite di un approccio più generale (analisi di Fourier) che permette di scrivere in forma integrale funzioni dipendenti in qualsiasi modo dal tempo. Tutto questo avrete modo di studiarlo in altri corsi.

Per i nostri scopi, è evidente che l'approccio della serie di Fourier ha conseguenze notevolissime. Infatti abbiamo già sviluppato una tecnica, il metodo simbolico (i "fasori"), che consente di determinare in modo semplice e efficace la risposta di un circuito, ad esempio un filtro, in regime sinusoidale. Poter scomporre un qualsiasi segnale periodico in armoniche sinusoidali consente di applicare a *ognuna* di queste la *funzione di trasferimento* del circuito. Ri-sommando le armoniche è infine possibile determinare la risposta del circuito quando ad esso è applicata una forma d'onda periodica *qualsiasi*, per esempio quadra o

triangolare. Vedremo le potenzialità dell'approccio negli esercizi 2 e 3.

A. Sviluppo di onda quadra e onda triangolare

I coefficienti di Fourier dipendono ovviamente dalla forma della funzione $g(t)$ che si sta considerando. Essi possono essere calcolati attraverso le relazioni Eq. 2, che richiedono di svolgere integrali sul tempo. Questi integrali possono essere facili o difficili da calcolare analiticamente, a seconda della funzione considerata. In linea di principio, essi possono comunque essere calcolati almeno numericamente, cosa che rende l'approccio di Fourier estremamente potente.

Ci sono due esempi molto significativi dal punto di vista pratico, almeno per i nostri scopi: essi riguardano lo sviluppo in serie di Fourier dell'onda quadra e dell'onda triangolare.

Prendiamo come $g(t)$ un'onda quadra di ampiezza $1/2$ (in unità arbitrarie), dunque di ampiezza picco-picco unitaria, e, per comodità, facciamo in modo che essa sia alternata e dispari; questo vuol dire che essa vale $-1/2$ per $(-T/2, 0)$ e $+1/2$ per $(0, T/2)$. In altre parole scegliamo per comodità l'origine dei tempi in modo tale che l'onda esca fuori proprio in questo modo. I risultati che seguono dipendono dalle scelte fatte, ma permettono anche di cogliere facilmente gli aspetti generali della procedura.

Cominciamo subito con il notare che, avendo supposto la $g(t)$ alternata, sarà sicuramente $a_0 = 0$. Infatti questo termine rappresenta il valore medio nel tempo della funzione, che è nullo essendo essa periodica e alternata. Inoltre, avendo scelto la nostra onda quadra dispari, è ovvio che ci aspettiamo che il suo sviluppo in serie di Fourier non contenga le armoniche espresse da coseni, cioè possiamo subito porre $b_k = 0$. Avessimo scelto un'onda quadra non dispari, avremmo avuto $b_k \neq 0$, al costo, però, di complicarci inutilmente i calcoli.

Concentriamoci quindi sui coefficienti c_k , iniziando per esempio con il calcolo esplicito di c_1 ($k = 1$). Allo scopo, dobbiamo calcolare l'integrale del prodotto tra la nostra $g(t)$ e la funzione $\sin(\omega t)$ nel "primo" periodo, cioè tra gli estremi di integrazione $(0, T)$. Il pannello in alto di Fig. 1 mostra queste due funzioni e il loro prodotto: l'integrale da calcolare è rappresentato dall'area (segnata) colorata in rosa, che è evidentemente non nulla. Si vede

che l'integrale da calcolare è infatti

$$c_1 = 2 \frac{2}{T} \int_0^{T/2} \frac{\sin(\omega t)}{2} dt = \quad (4)$$

$$= \frac{2}{\omega T} \int_0^{\omega T/2} \sin \xi d\xi = \quad (5)$$

$$= -\frac{2}{2\pi} \cos \xi \Big|_0^\pi = \quad (6)$$

$$= \frac{2}{\pi}, \quad (7)$$

dove la vergognosa prolissità dei passaggi dovrebbe fugare ogni dubbio.

Il pannello centrale di Fig. 1 mostra le stesse funzioni per $k = 2$: si vede subito che l'area (segnata) è stavolta nulla, per cui $c_2 = 0$. Il pannello in bassa della stessa figura si riferisce invece a $k = 3$, dove l'area non è nulla, per cui $c_3 \neq 0$. Stavolta l'integrale da calcolare è

$$c_1 = 2 \frac{2}{T} \int_0^{T/6} \frac{\sin(3\omega t)}{2} dt = \quad (8)$$

$$= \frac{2}{3\omega T} \int_0^{3\omega T/6} \sin \xi d\xi = \quad (9)$$

$$= -\frac{2}{2\pi} \frac{1}{3} \cos \xi \Big|_0^\pi = \quad (10)$$

$$= \frac{2}{3\pi}. \quad (11)$$

Andando avanti con le armoniche, ci si rende facilmente conto che $c_k = 0$ per *tutti i k pari*, mentre per i termini dispari si ha

$$c_k = \frac{2}{k\pi}, k \text{ dispari}. \quad (12)$$

Si può dimostrare con un po' di lavoro che, anche rilassando la condizione sulla disparità della funzione, lo sviluppo di un'onda quadra contiene *solo armoniche dispari*.

Per l'onda triangolare, che per semplicità supponiamo *pari*, a media nulla, e sempre di ampiezza $1/2$, i conti sono un po' più complicati. Per cominciare, avendo preso un'onda triangolare dispari, è ovvio che ci aspettiamo che il suo sviluppo in serie di Fourier non contenga le armoniche espresse da seni, cioè possiamo subito porre $c_k = 0$. Come ulteriore passo, osserviamo la Fig. 2, concettualmente analoga alla Fig. 1: si vede subito che anche in questo caso l'area sottesa al prodotto tra le funzioni è nulla per $k = 2$, e, per estensione, per tutti i k pari. Dunque anche stavolta lo sviluppo contiene solo i termini con k dispari. Il calcolo, però, non è banale. Potete provare a operare "per parti", giungendo a

$$b_k = \frac{4}{(k\pi)^2}, k \text{ dispari}. \quad (13)$$

Un modo più semplice per giungere allo stesso risultato consiste nel notare che un'onda triangolare (pari) può essere considerata come l'integrale nel tempo di un'onda quadra (dispari). Integrare nel tempo le armoniche di

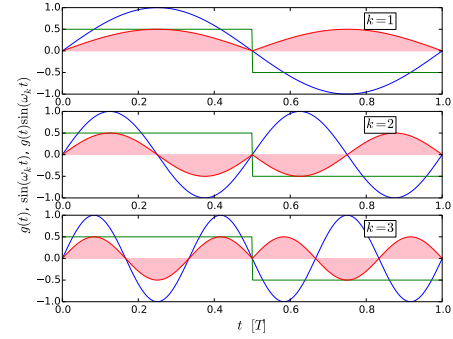


Figura 1. Figura esemplificativa del calcolo dei primi tre coefficienti di Fourier c_k ($k = 1-3$) per un'onda quadra dispari alternata di ampiezza picco-picco unitaria. La scala orizzontale si estende su un sigolo periodo. Le linee verdi rappresentano l'onda quadra, le blu le funzioni $\sin(\omega_k t)$, le rosse il prodotto tra le due. In rosa è ombreggiata l'area sottesa al prodotto tra le funzioni. Grafici ovviamente fatti con Python.

tipo seno che compaiono nello sviluppo dell'onda quadra significa trasformarle in armoniche di tipo coseno. Inoltre l'operazione di integrazione fa comparire, messo in evidenza, un termine $1/\omega_k = 1/(k\omega)$, che è responsabile per lo specifico valore dei coefficienti b_k di Eq. 13 (che sono, in sostanza, il quadrato dei coefficienti c_k di Eq. 12).

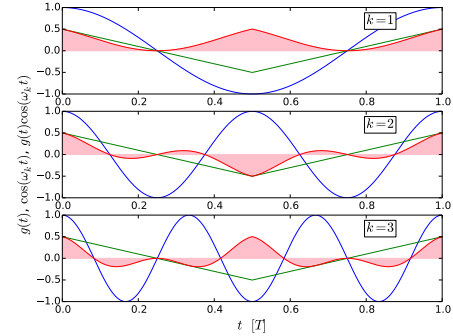


Figura 2. Analogo di Fig. 1 per il caso di onda triangolare alternata pari e ampiezza picco-picco unitaria. Grafici ovviamente fatti con Python.

Dunque abbiamo trovato il modo per esprimere forme d'onda quadre e triangolari come serie di funzioni armoniche sinusoidali. L'espressione specifica dei coefficienti determinati dipende dalle scelte eseguite (parità, ampiezza, carattere alternato). Come già affermato: (i) per un'onda quadra non dispari compaiono nello sviluppo anche i termini b_k , e per un'onda triangolare non pari compaiono i termini c_k (ma in ogni caso solo con k dispari); (ii) se l'onda non è alternata, il termine a_0 è diverso da zero. Inoltre, come è facile intuire, se l'ampiezza picco-picco non è unitaria *tutti* i coefficienti vengono moltiplicati per un opportuno parametro di scala.

II. HINTS

Gli esercizi proposti in questa nota si basano proprio sull'impiego di Python per ricostruire forme d'onda periodiche (quadre e triangolari) a partire dalle armoniche sinusoidali. Può essere utile soffermarsi su alcuni consigli generali per l'implementazione pratica degli script.

In primo luogo è ovvio che si ha a che fare con arrays unidimensionali, cioè vettori: la variabile indipendente (il tempo) è un array, ogni armonica è un array, la funzione ottenuta sommando le armoniche è un array. Lavorare con gli arrays significa fare uso del pacchetto `numpy`. Per esempio, per creare un array (qui chiamato `t`) per la variabile indipendente t si può usare il comando `t = numpy.linspace(-2,2,1000)`, che genera un vettore di 1000 punti (dovrebbero essere sufficienti) distribuiti in modo equispaziato tra -2 e 2. Un comando per creare un array nullo, sempre di 1000 punti, è `w=numpy.zeros(1000)` (`w` è il nome dell'array), se vogliamo invece un array che contenga il coseno della variabile `t` occorre scrivere `w=numpy.sin(t)`, e così via.

Quindi ognuna delle armoniche può essere definita calcolando l'appropriata funzione, per esempio `wk = numpy.cos(omegak*t)` crea una funzione coseno dell'argomento specificato. Infine le varie armoniche, debitamente moltiplicate per i propri pesi (i coefficienti di Fourier), possono essere facilmente sommate tra loro come arrays.

Naturalmente la somma di Eq. 1 dovrebbe, in matematica, comprendere infiniti elementi. Dal punto di vista numerico sarà però sufficiente estendere la somma a un numero finito, *sufficientemente grande*, di elementi. Si possono adottare diverse tecniche per istruire il software a eseguire tali somme. Probabilmente il più semplice consiste nell'introdurre dei *cicli* nello script. Questo può per esempio essere fatto con l'istruzione `for counter in range(start, stop, step)`: (ricordate che le righe di script che appartengono al ciclo *devono essere indentate*, cioè scritte premettendo una tabulazione, e fate attenzione ai due punti a fine istruzione), dove la possibilità di introdurre lo step size può essere utile per considerare solo armoniche pari o dispari (è sufficiente porre lo step pari a 2).

III. ESERCIZIO 1 (OBBLIGATORIO)

Il primo esercizio proposto è molto semplice: si tratta di ricostruire le forme d'onda quadra e triangolare usando le Eqs. 12,13. Esso è *obbligatorio*, nel senso che tutti dovete svolgerlo, singolarmente o riunendovi a gruppi, e mostrarmi l'uscita e lo script (potete inviarmeli via e-mail o portarmi la stampa), entro la scadenza che abbiamo concordato (inizio vacanze di Natale).

Può essere interessante osservare le forme d'onda corrispondenti a un numero via via crescente di iterazioni, cioè di elementi considerati nella somma di Fourier. Come atteso, occorre che la somma sia estesa a un cer-

to, non piccolo, numero di elementi affinché le forme d'onda ottenute approssimino quelle richieste (quadra e triangolare).

La Fig. 3 mostra il risultato da me ottenuto per l'onda quadra con un numero crescente di iterazioni (il parametro n che compare nei grafici). Notate che le forme d'onda sono graficate in funzione del tempo espresso in unità di periodo T : infatti in questo esercizio non è rilevante il valore fisico della frequenza. Inoltre, ovviamente, l'ampiezza ha unità arbitrarie (il valore picco-picco è 1 [arb. un.]). Si vede come nel caso dell'onda quadra siano necessarie almeno alcune centinaia di iterazioni per ottenere la forma desiderata. Questo è dovuto al fatto che la forma d'onda quadra, con i suoi fronti di salita e discesa molto ripidi, non può essere riprodotta sommando su un numero troppo piccolo di componenti. Notate che il numero di componenti necessarie dipende, in parte, dal numero di punti che costituiscono la forma d'onda, ovvero l'array, che nel mio esempio è limitato a 1000.

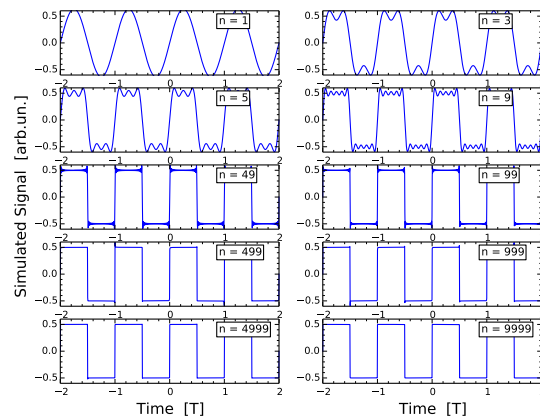


Figura 3. Sviluppo in serie di Fourier per un'onda quadra: i vari subplots si riferiscono a un numero n via via crescente di iterazioni, come indicato nei grafici. Gli array graficati sono composti da 1000 punti.

La Fig. 4 si riferisce invece all'onda triangolare: qui, grazie all'assenza di ripidi fronti d'onda, è sufficiente un numero di iterazioni di poche decine per ottenere un risultato soddisfacente. In ogni caso l'efficienza del software permette di tenere alto il numero di iterazioni senza pregiudicare (troppo) la rapidità con cui i cicli vengono eseguiti.

IV. ESERCIZIO 2: PINNA DI SQUALO (OBBLIGATORIO)

L'esercizio 2 qui proposto, anch'esso obbligatorio, fa riferimento all'esperienza pratica in cui si osservava l'uscita di un filtro passa-basso, o integratore, al cui ingresso era posta un'onda quadra: al variare della frequenza dell'onda quadra in ingresso, l'uscita tendeva sempre più ad

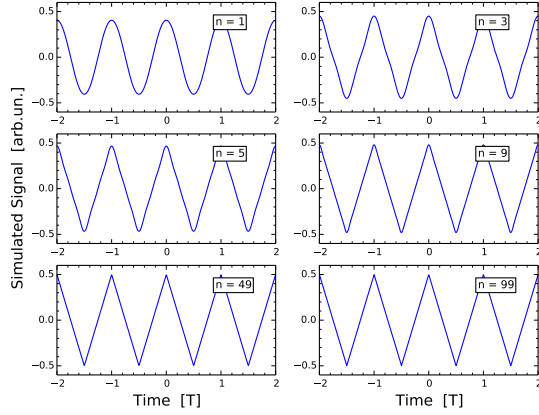


Figura 4. Sviluppo in serie di Fourier per un'onda triangolare: i vari subplots si riferiscono a un numero n via via crescente di iterazioni, come indicato nei grafici. Gli array graficati sono composti da 1000 punti.

assomigliare a un'onda triangolare, con un vasto regime intermedio in cui la forma d'onda in uscita assumeva una forma simile a una sorta di pinna di squalo.

Dal punto di vista qualitativo si può trovare facilmente un'interpretazione legata ai tempi caratteristici di carica e scarica del condensatore. L'obiettivo di questo esercizio è quello di simulare questa osservazione in maniera quantitativa.

L'integratore era costituito da $R = 330$ ohm e $C = 0.1$ μ F (naturalmente potete anche usare altri valori, nell'esercizio), per cui si aveva una frequenza di taglio $f_T = 1/(2\pi RC) \approx 4825$ Hz. Per semplicità si usano i valori nominali e si suppone che gli effetti delle resistenze interne di generatore e oscilloscopio siano trascurabili. Grazie all'applicazione del metodo simbolico, sappiamo che un integratore alimentato con un'onda sinusoidale di frequenza f dà luogo a:

1. un'attenuazione $A(f) = 1/\sqrt{1 + (f/f_T)^2}$;
2. uno sfasamento $\Delta\phi = \arctan(-f/f_T)$.

Attenuazione e sfasamento agiranno indipendentemente su tutte le componenti dell'onda quadra supposta in ingresso al circuito, essendo per ogni componente considerata $f = f_k = \omega_k/(2\pi)$. In altre parole, si può agevolmente "simulare" la forma d'onda $w(t)$ in uscita dal circuito sommando le componenti armoniche dell'onda quadra, cioè costruita con i coefficienti di Eq. 12, moltiplicando l'ampiezza di ogni componente per la dovuta attenuazione e mettendo nell'argomento delle armoniche di Fourier il dovuto sfasamento.

In altre parole avremo:

$$A_k = \frac{1}{\sqrt{1 + (\omega_k/\omega_T)^2}} \quad (14)$$

$$\Delta\phi_k = \arctan(-\omega_k/\omega_T) \quad (15)$$

$$w(t) = \sum_{k=1}^n c_k A_k \sin(\omega_k t + \Delta\phi_k), \quad (16)$$

con k dispari e c_k dato da Eq. 12 (abbiamo tenuto conto del fatto che $a_0 = 0$, essendo l'onda alternata e che $b_k = 0$ trattandosi di sviluppo di onda quadra).

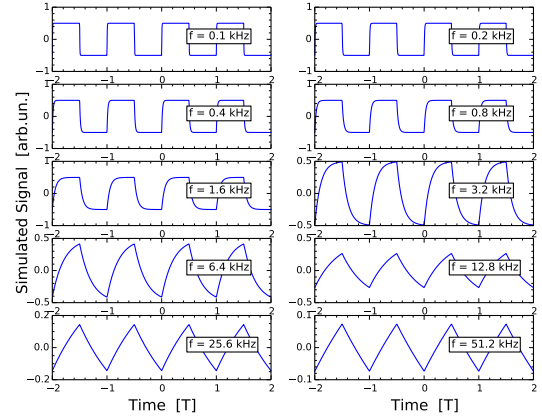


Figura 5. "Simulazione" dell'integratore descritto nel testo con all'ingresso un'onda quadra. I subplots si riferiscono a diverse frequenze, come riportato nei grafici. Fate attenzione alla scala orizzontale, che è sempre in unità di periodo T , e alla scala verticale, che invece cambia di subplot in subplot. Gli array graficati sono composti da 1000 punti.

La Fig. 5 mostra l'esito della "simulazione" per diversi valori della frequenza f della forma d'onda in ingresso, come indicato sui grafici: fate attenzione al fatto che l'asse orizzontale di tutti i subplots è in unità di periodo T , che ovviamente cambia in funzione della frequenza. Notate inoltre che la scala verticale, espressa in [arb. un.], cambia di subplot in subplot al diminuire del fattore $A(f)$.

Il risultato è in accordo con le osservazioni: a basse frequenze, inferiori a f_T , la forma d'onda non viene praticamente modificata e all'uscita si ritrova il segnale inviato all'ingresso. Aumentando la frequenza si nota una deformazione della forma d'onda, accompagnata da una attenuazione dell'ampiezza, che determina, per $f \gg f_T$, un'uscita di forma pressoché triangolare.

Naturalmente potete estendere questo esercizio ad altre situazioni che possono essere riprodotte nelle esperienze pratiche. Per esempio potete considerare in ingresso una forma d'onda triangolare, oppure esaminare un derivatore invece di un integratore.

V. ESERCIZIO 3: INTEGRATORE + DERIVATORE

Questo esercizio, che è un po' più laborioso del precedente e di conseguenza lasciato come "facoltativo" (ma *fortemente consigliato*, richiede di considerare il circuito composto da integratore e derivatore in cascata realizzato sperimentalmente nell'esperienza pratica. Lo scopo è anche in questo caso quello di "simulare" le forme d'onda

dei segnali in uscita da integratore e derivatore, rispettivamente V_A e V_B , e di verificare il rapporto tra le loro ampiezze e l'ampiezza del segnale in ingresso, V_{in} . Come nell'esperienza pratica, si suppone che all'ingresso ci sia un'onda quadra: quando integratore e derivatore funzionano come devono, l'uscita A avrà forma triangolare e l'uscita B quadra.

Per semplicità, in questo esercizio *non si considerano gli effetti dovuti al collegamento in cascata dei due sotto-circuiti*, né quelli legati alla presenza di resistenze interne. In buona sostanza, si suppone qui di avere un integratore che ha frequenza di taglio f_{TA} con in cascata un derivatore che ha frequenza di taglio f_{TB} .

Nel mio esempio ho supposto $f_{TA} = 50$ Hz e $f_{TB} = 25$ kHz; naturalmente voi siete invitati a usare i valori effettivi dei circuiti che avete montato e testato. Per determinare l'uscita in A (uscita dell'integratore) $w_A(t)$ ho impiegato in pratica le stesse relazioni di Eq. 14

$$A_{A,k} = \frac{1}{\sqrt{1 + (\omega_k/\omega_{TA})^2}} \quad (17)$$

$$\Delta\phi_{A,k} = \arctan(-\omega_k/\omega_{TA}) \quad (18)$$

$$w_A(t) = \sum_{k=1}^n c_k A_{A,k} \sin(\omega_k t + \Delta\phi_{A,k}), \quad (19)$$

dove k è dispari e c_k è dato dall'Eq. 12.

Come ben sapete, l'ulteriore stadio di derivazione, con frequenza di taglio f_{TB} , introduce:

1. un'ulteriore *attenuazione* $A(f) = 1/\sqrt{1 + (f_{TB}/f)^2}$;
2. un ulteriore *sfasamento* $\Delta\phi = \arctan(f_{TB}/f)$.

Queste ulteriori modifiche delle armoniche agiscono "in cascata": l'attenuazione andrà a moltiplicare l'ampiezza delle componenti armoniche già attenuate dall'integratore, e lo sfasamento andrà a sommarsi allo sfasamento prodotto dall'integratore. Detta $w_B(t)$ la forma d'onda in uscita da B, si può quindi scrivere

$$A_{B,k} = \frac{1}{\sqrt{1 + (\omega_{TB}/\omega_k)^2}} \quad (20)$$

$$\Delta\phi_{B,k} = \arctan(\omega_{TB}/\omega_k) \quad (21)$$

$$w_B(t) = \sum_{k=1}^n c_k A_{A,k} A_{B,k} \sin(\omega_k t + \Delta\phi_{A,k} + \Delta\phi_{B,k}) \quad (22)$$

dove $A_{A,k}$ e $\Delta\phi_{A,k}$ sono quelli determinati in Eq. 17. Notate che, come è ovvio, per le frequenze a cui integratore e derivatore si comportano come si deve i due sfasamenti si annullano a vicenda, tendendo rispettivamente a $-\pi/2$ (integratore) e $\pi/2$ (derivatore), per cui in uscita da B si ritrova la stessa forma che si ha all'ingresso, ovviamente attenuata.

Le Figs. 6,7 mostrano i risultati, cioè le forme d'onda "simulate" in uscita rispettivamente da A e da B. Si ricorda che la forma d'onda in ingresso è quadra, di ampiezza picco-picco unitaria. Il range di frequenze f considerato parte da f_{TA} e arriva a f_{TB} . Notate che anche in queste figure l'asse orizzontale è espresso in unità di periodo T , mentre l'asse verticale cambia di subplot in subplot.

Cominciamo con l'esaminare l'uscita A (Fig. 6): si vede chiaramente come all'aumentare della frequenza f il sotto-circuito A si comporti in modo sempre più netto come integratore, fornendo in uscita un'onda di forma che approssima sempre meglio la triangolare. Un aspetto interessante riguarda il fattore di attenuazione A_A tra ampiezza picco-picco in uscita dall'integratore e ampiezza in ingresso (supposta unitaria). Questa attenuazione è stata misurata a qualche frequenza in laboratorio e confrontata con il valore atteso $A_{A,att} = f_{TA}/f$ determinato dall'analisi nel dominio dei tempi. Ovviamente questo valore atteso si applica solo quando l'integratore si comporta bene, cioè per $f \gg f_{TA}$. Verifichiamo l'accordo tra aspettativa e valore "simulato" a due frequenze, $f_1 = 0.8$ kHz e $f_2 = 6.4$ kHz, che soddisfano la condizione appena scritta. I corrispondenti valori attesi sono $A_{A,att1} = 62.5 \times 10^{-3}$ e $A_{A,att2} = 7.8 \times 10^{-3}$.

Osservando i grafici è immediato dedurre i valori "simulati" delle attenuazioni a queste due frequenze (ricordate sempre che l'onda quadra in ingresso ha ampiezza picco-picco unitaria), che risultano rispettivamente $A_{A,sim1} \approx 100 \times 10^{-3}$ e $A_{A,sim2} \approx 12 \times 10^{-3}$. Come si vede, l'accordo con le aspettative è piuttosto scarso e, come ragionevole, migliora quando la frequenza aumenta, cioè quando il regime di funzionamento dell'integratore soddisfa meglio l'approssimazione $A(f) \sim f_T/f$ (potete facilmente osservare che questa approssimazione è soddisfatta entro il 5% anche alla frequenza f_1 , per cui il problema non è qui).

L'interpretazione della discrepanza è semplice: il valore atteso è riferito ad onde sinusoidali, e quindi si basa sul fatto che tutte le componenti armoniche siano attenuate allo stesso modo, mentre l'attenuazione dipende dalla frequenza dell'armonica stessa. Di conseguenza non stupisce che l'ampiezza della forma d'onda in uscita dipenda dalla frequenza di lavoro. D'altra parte questo è in accordo con quanto osservato sperimentalmente. Quindi l'analisi in serie di Fourier rappresenta un valido metodo per predire l'attenuazione effettiva quando a circuiti integratori (o derivatori) vengono applicate forme d'onda non sinusoidali.

Vediamo ora cosa succede per la forma d'onda simulata in uscita dal derivatore (uscita B, Fig 7). Questo sotto-circuito funziona bene per $f \ll f_{TB}$. In effetti a basse frequenze il derivatore fa il suo mestiere, ma agisce su una forma d'onda che è praticamente quadra, dato che a queste frequenze l'integratore non integra "abbastanza". C'è poi un intervallo di frequenze (per esempio per i grafici corrispondenti a $f = 1.6$ kHz e $f = 3.2$ kHz) in cui l'uscita B riporta una forma d'onda quadra: qui sia l'integratore che il derivatore fanno il loro dovere. Se la frequenza viene ulteriormente aumentata si esce dalla condizione $f \ll f_{TB}$ e la forma d'onda in uscita dal derivatore tende a essere inalterata rispetto a quello al suo ingresso: dunque essa tende ad essere triangolare, o, se preferite, a pinna di squalo.

Sofferamoci anche qui a esaminare le attenuazioni tra ampiezza picco-picco in uscita B e ampiezza picco-

picco in ingresso *all'intero circuito*. Ricordiamo che, in questo caso, il valore atteso è $A_{B,att} = f_{TA}/f_{TB} = 2 \times 10^{-3}$ *indipendente dalla frequenza*. Osserviamo subito che esiste un intervallo di frequenze nel quale questa aspettativa è piuttosto ben verificata. Per esempio alla frequenza f_1 considerata prima si ha $A_{B,sim1} \approx 2 \times 10^{-3}$, e questo si verifica abbastanza bene anche nell'intero intervallo in cui le condizioni di lavoro sono tali da garantire il corretto funzionamento di integratore e derivatore.

Anche qui l'interpretazione è piuttosto semplice: presi separatamente, le attenuazioni di integratore e derivatore dipendono dalla frequenza di lavoro in un modo che non segue la semplice aspettativa. Tuttavia i due circuiti sotto-attenuano e sovra-attenuano le varie componenti armoniche in un modo che finisce per compensarsi. Alla fine, l'attenuazione complessiva risulta indipendente dalla frequenza di lavoro e in ragionevole accordo con le attese.

Nel regime di frequenze alte in cui il derivatore non compie più il proprio dovere, per esempio già per f_2 considerata sopra, l'accordo non è più così buono, anche se le deviazioni dalle attese sono piuttosto limitate. Per esempio, alla frequenza $f_3 = 25.6$ kHz si ha $A_{B,sim3} \approx 1.8 \times 10^{-3}$, che rappresenta una discrepanza di pochi punti percentuali rispetto alle attese.

Infine, anche in questo caso potete estendere l'analisi simulata ad altre situazioni, per esempio verificare cosa succede se in ingresso supponete di avere un'onda triangolare, oppure se impiegate altri valori per resistenze e condensatori, o altri intervalli di frequenza.

VI. ESERCIZIO 4: AMPIEZZA IN USCITA DALL'INTEGRATORE

L'esercizio, da considerarsi anche questo facoltativo (più facoltativo del precedente), è in pratica un'estensione dell'esercizio 3. Abbiamo infatti già osservato che l'ampiezza del segnale in uscita dall'integratore (quella che nell'esperimento era denominata V_A) e dal derivatore in cascata ad esso (denominata V_B) sono funzioni della

frequenza che si comportano in modo diverso rispetto a quanto ci si attende per onde sinusoidali. Di conseguenza le attenuazioni A_A e A_B hanno un comportamento differente rispetto a quello delle onde sinusoidali. In questo esercizio si vuole determinare tale andamento su un certo intervallo di frequenze (come prima, si suppone $f_{TA} = 50$ Hz e $f_{TB} = 25$ kHz).

Poiché lo scopo è quello di costruire dei grafici, è evidente che il metodo che abbiamo, più o meno implicitamente, adottato sopra, consistente in sostanza nel costruire i grafici dei segnali simulati in uscita in funzione del tempo e di determinarne l'ampiezza picco-picco "a occhio", non può essere usato. Occorre infatti ripetere la simulazione per tante frequenze distinte (nel mio esempio ne ho usate 50, equispaziate logaritmicamente nell'intervallo 2 Hz – 0.2 MHz), cioè per ognuna di queste frequenze bisogna far costruire la forma d'onda simulata in uscita da integratore e derivatore, e poi occorre trovare un modo "automatico" che consenta di determinare l'ampiezza dell'onda risultante, senza dover visionare tutti i grafici prodotti.

Nel mio esempio ho trovato utile usare l'istruzione `numpy.amax(array)`, che fornisce il valore massimo dell'array. Ho deciso di utilizzare questo valore massimo come ampiezza dell'onda. Notate che questa scelta è non necessariamente corretta, visto che le forme d'onda possono essere, in qualche caso, irregolari, cioè presentare ad esempio dei "picchi" irregolari, come nei pannelli in alto di Fig. 7. Tuttavia essa rappresenta una possibile (e ovvia) definizione di ampiezza. Da queste definizioni di ampiezza ho dedotto le attenuazioni A_A e A_B , semplicemente normalizzando per l'ampiezza dell'onda quadra in ingresso [2].

La Fig. 8 mostra il risultato che ho ottenuto, sia per A_A che per A_B . Il grafico riporta anche gli andamenti attesi per i casi sinusoidali, permettendo di apprezzare la differenza quando si usa all'ingresso del circuito una forma d'onda quadra. Naturalmente questa differenza dipende dalla scelta, appena descritta, della misura di ampiezza.

-
- [1] L'analisi di Fourier di segnali con dipendenza temporale arbitraria (dunque descritti da integrali, e non da serie) può essere condotta usando degli algoritmi che consentono di determinare con grande efficacia i coefficienti rilevanti, noti come *algoritmi FFT* (Fast Fourier Transform). Probabilmente ne farete uso nelle vostre attività future.
- [2] Osservate anche che questo esercizio, più ancora dei precedenti, costituisce un buon banco di prova per testare la propria capacità di maneggiare un problema piuttosto serio nel caso di "simulazioni" di forme d'onda periodiche, che è quello del *campionamento*. Vi renderete facilmente

conto che per descrivere in maniera opportuna la forma d'onda è necessario che l'array corrispondente sia dotato di un numero "sufficiente" di punti. Per questo motivo, qui ancor più che negli esempi precedenti, è necessario "scalare" in modo furbo l'array dei tempi t , in modo tale che esso contenga il numero "sufficiente" di punti a prescindere dalla frequenza che si sta considerando. Non credo di essere stato chiarissimo, ma confido nel fatto che, se ci provate, vi renderete presto conto dell'importanza di questo aspetto.

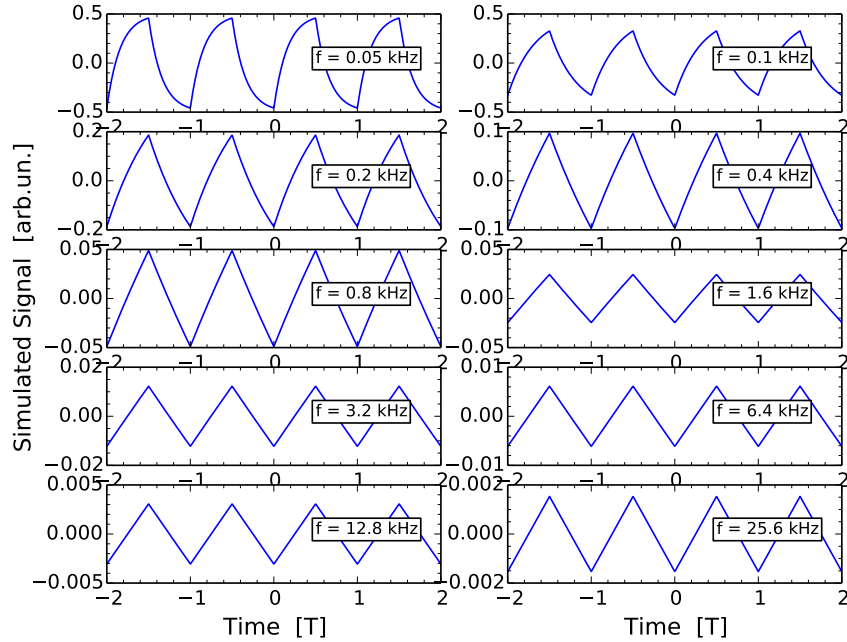


Figura 6. “Simulazione” del segnale in uscita dall’integratore descritto nel testo con all’ingresso un’onda quadra. I subplots si riferiscono a diverse frequenze, come riportato nei grafici. Fate attenzione alla scala orizzontale, che è sempre in unità di periodo T , e alla scala verticale, che invece cambia di subplot in subplot. Gli array graficati sono composti da 1000 punti.

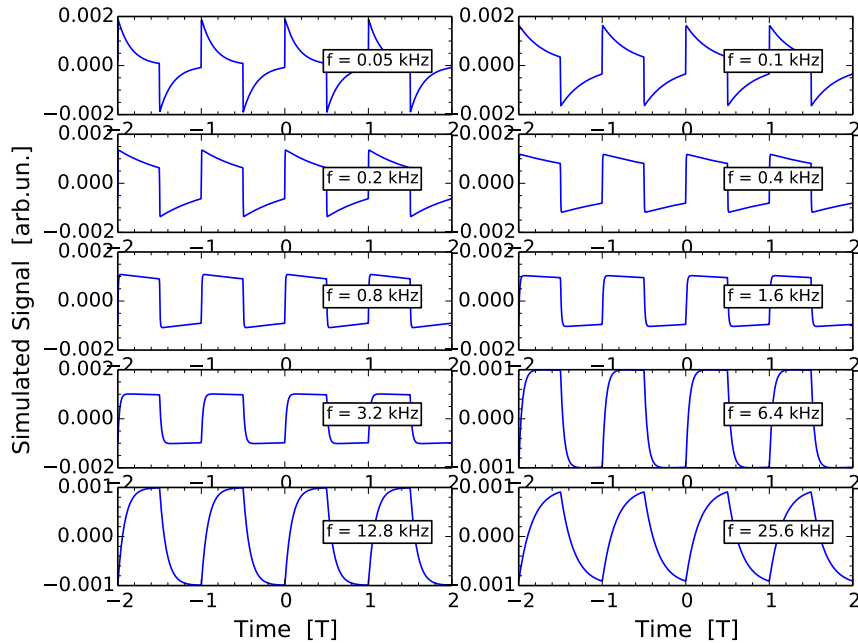


Figura 7. “Simulazione” del segnale in uscita dal circuito integratore + derivatore descritto nel testo con all’ingresso un’onda quadra. I subplots si riferiscono a diverse frequenze, come riportato nei grafici. Fate attenzione alla scala orizzontale, che è sempre in unità di periodo T , e alla scala verticale, che invece cambia di subplot in subplot. Gli array graficati sono composti da 1000 punti.

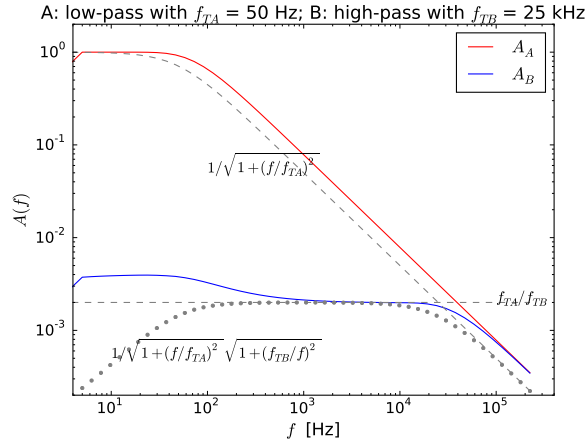


Figura 8. “Simulazione” delle attenuazioni A_A e A_B in uscita dall’integratore e dall’integratore + derivatore descritto nel testo in funzione della frequenza di lavoro e per i valori di frequenza di taglio riportati nel testo e nel titolo. Le curve tratteggiate e punteggiate rappresentano gli andamenti attesi per onde sinusoidali (le funzioni corrispondenti sono scritte sul grafico). Il grafico si riferisce a 50 valori di frequenza distinti e equispaziati logicamente nell’intervallo considerato (la linea continua raccorda i diversi punti).