

Esercizi obbligatori su serie di Fourier e Python

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

(Dated: version 5 - FF, 22 novembre 2016)

Questa nota propone alcuni esercizi sullo sviluppo in serie di Fourier che dovete svolgere, da soli o in piccolo 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 $\omega_k = k\omega$, con k intero positivo (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, al momento il modo migliore consiste nell'usare una sovrapposizione (serie) di funzioni seno e coseno, da cui il nome di *serie di Fourier di seni e coseni*:

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

dove $n \rightarrow \infty$ e il coefficiente a_0 tiene conto del valore medio della funzione. Quando possibile, nel seguito faremo riferimento a funzioni periodiche *alternate*, per le quali $a_0 = 0$. I *coefficienti* dell'espansione di Fourier in seni e coseni, 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 strettamente 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. Il metodo sviluppato nell'analisi di Fourier consente di trasformare le informazioni ottenute "in un dominio" (per esempio, nel dominio dei tempi, cioè analizzando il comportamento di un sistema in funzione del tempo) in informazioni rilevanti per il dominio "coniugato", o reciproco (per esempio, nel dominio delle frequenze, dove spesso le informazioni hanno la forma di "spettri"). Questa possibilità ha ricadute bellissime, come avrete modo di verificare nel futuro in altri corsi

e anche, in forma limitata agli aspetti più squisitamente legati all'analisi dei dati, in questo nostro corso.

Per i nostri scopi attuali, dove ci restringiamo a usare lo sviluppo in serie di seni e coseni per esprimere funzioni periodiche, è evidente che l'approccio della serie di Fourier ha conseguenze notevolissime. Infatti abbiamo già sviluppato una tecnica, il metodo simbolico (quello dei "fasori", con linguaggio da elettrotecnici), 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 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*, che per noi vuol dire, al momento, quadra o triangolare. Vedremo le potenzialità dell'approccio negli esercizi 2-4.

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. Normalmente il calcolo di integrali si esegue, con grande efficacia, con metodi numerici e, nell'ambito dell'analisi di Fourier, sono stati sviluppati degli algoritmi potentissimi, chiamati in genere *FFT* (Fast Fourier Transform), o *DFT* (Discrete Fourier Transform), che permettono di calcolare numericamente i coefficienti anche per funzioni non periodiche nel tempo. Su tutto ciò torneremo più avanti nel corso.

Per ora, il nostro obiettivo è quello di determinare *analiticamente* i coefficienti dell'espansione di Fourier di semplici funzioni periodiche, precisamente di funzioni tipo onda quadra e onda triangolare (simmetriche e alternate).

Prendiamo come $g(t)$ un'onda quadra di ampiezza $1/2$ (in unità arbitrarie), dunque di ampiezza picco-picco unitaria: questa specifica sull'ampiezza serve per semplificare la matematica, ed è ovvio che essa non è una restrizione (potremo sempre riscalare, attraverso semplice moltiplicazione, il valore dei coefficienti nel caso in cui l'ampiezza picco-picco non sia unitaria). Sempre per comodità, facciamo in modo che l'onda quadra sia alternata (come già affermato, $a_0 = 0$) 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. Anche in questo caso, non si perde in generalità, dato che, se l'onda quadra considerata non fosse dispari, potremmo sempre immaginare di aggiungere un termine di fase costante per renderla dispari. Rifrasando: i risultati che seguono dipendono dalle scelte fatte, ma l'estensione a casi generali è immediata.

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 $b_k = 0$ (per ogni k intero positivo).

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 basso 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

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

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*.

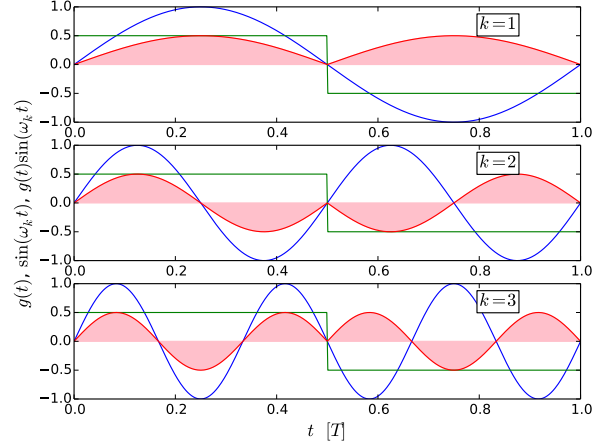


Figura 1. Figura esemplificativa del calcolo dei primi tre coefficienti di Fourier c_k ($k = 1-3$) per un'onda quadra dispari alternata di ampiezza picco-picco unitaria. La scala orizzontale si estende su un singolo periodo. Le linee verdi rappresentano l'onda quadra, le blu le funzioni $\sin(\omega_k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

Per l'onda triangolare, che per semplicità supponiamo *pari*, a media nulla, sempre di ampiezza $1/2$ e ampiezza picco-picco unitaria, i conti sono un po' più complicati. Per cominciare, avendo scelto un'onda pari, 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 già a partire dal primo termine non nullo, b_1 . Potete provare a operare "per parti", giungendo a $b_1 = 4/\pi^2$.

Iterando la procedura, cioè andando avanti con le armoniche, si trova

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

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 tipo seno che compaiono nello sviluppo dell'onda quadra significa trasformarle in armoniche di tipo coseno, per cui, se per l'onda quadra sono nulli i coefficienti b_k , per la triangolare lo sono i c_k . Inoltre quando si integrano termini del tipo $\sin(\omega_k t)$ 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: in sostanza essi sono il quadrato dei coefficienti c_k di Eq. 12.

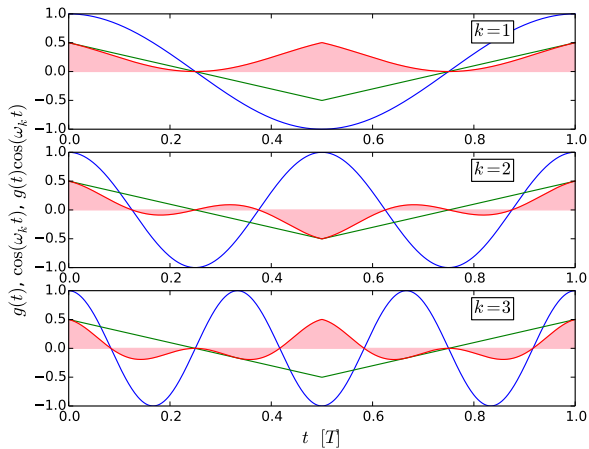


Figura 2. Analogo di Fig. 1 per il caso di onda triangolare alternata pari e ampiezza picco-picco unitaria.

Dunque abbiamo trovato il modo per esprimere forme d'onda quadre e triangolari come serie di coseni e seni. 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 se l'ampiezza picco-picco non è unitaria tutti i coefficienti vengono moltiplicati per un opportuno parametro di scala.

II. ESERCIZIO E HINTS

L'esercizio obbligatorio, da svolgere individualmente o in piccolo gruppo e da consegnare *tassativamente* entro la data stabilita, è in realtà costituito da diversi sotto-esercizi. Per la consegna, è sufficiente che mi mandate l'esito (i grafici) e gli script da voi prodotti via e-mail, assieme, se volete, a una semplice e concisa descrizione.

Lo scopo generale dell'esercizio è quello di “divertirsi” (per quanto possibile) costruendo espansioni di Fourier a partire da un foglio bianco, cioè senza usare pacchetti o comandi prefabbricati, e nel modo più semplice possibile. La finalità di questo divertimento è quella di replicare, se preferite “simulare” (ma il termine qui è un po' troppo pomposo), forme d'onda che avete visto in laboratorio, cioè onde quadre e triangolari, e la strana forma “a pinna di squalo” che si osserva, per certe scelte dei parametri di funzionamento, con un integratore RC alimentato da un'onda quadra. In questo caso, grazie all'acquisizione con Arduino dei dati sperimentali, avete addirittura la possibilità di sovrapporre in un unico grafico esperimento e simulazione, in una sorta di metafora semplice semplice di quanto si fa normalmente nel mestiere della fisica.

Prima di procedere oltre, può essere utile soffermarsi su alcuni consigli generali per l'implementazione pratica degli script di Python.

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(-2,2,5000)`, che genera un vettore di 5000 punti (dovrebbero essere sufficienti per evitare *aliasing* e problemi di sotto-campionamento, ma fate sempre attenzione) distribuiti in modo equispaziato tra -2 e 2. Un comando per creare un array nullo, sempre di 5000 punti, è `w=numpy.zeros(5000)` (`w` è il nome dell'array), se vogliamo invece un array che contenga il coseno della variabile t occorre scrivere `w=numpy.cos(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. Si possono adottare diverse tecniche per istruire il software a eseguire le somme necessarie. 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 e regolare opportunamente lo start, per esempio far partire la somma da 1 o da 2).

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. In genere, per capire se il numero di iterazioni nella somma è sufficientemente grande basta guardare il grafico che si ottiene e verificare “a occhio” se la forma è quella attesa, in particolare che non compaiano “spigolosità” o altri artefatti. In alternativa si possono generare, con appositi pacchetti disponibili in Python, delle onde quadre o triangolari “modello”, e verificare quantitativamente la differenza fra valori dell'onda modello e dell'onda creata per espansione di Fourier, per esempio usando la somma dei residui quadrati come indicatore della qualità di riproduzione.

III. SOTTO-ESERCIZIO 1 (OBBLIGATORIO): QUADRA E TRIANGOLARE

Il primo esercizio proposto è molto semplice: si tratta di ricostruire delle forme d'onda quadra e triangolare

usando i coefficienti di Eqs. 12,13. Può essere interessante osservare le forme d'onda corrispondenti a un numero via via crescente di iterazioni, cioè di elementi considerati nella somma di seni o coseni. 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 (non si notano evidenti problemi di aliasing nonostante il numero relativamente ridotto di punti).

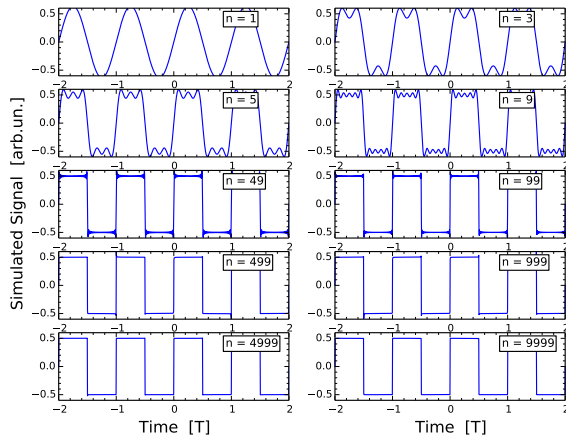


Figura 3. Sviluppo in serie di Fourier per un'onda quadra: i vari grafici si riferiscono a un numero n via via crescente di iterazioni, come indicato in legenda. Gli array graficati sono composti da 1000 punti.

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.

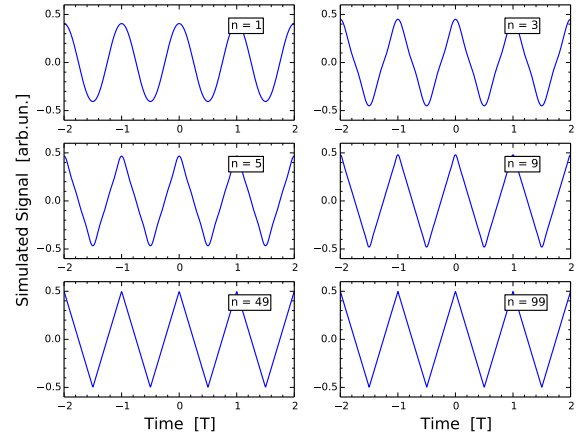


Figura 4. Sviluppo in serie di Fourier per un'onda triangolare: i vari grafici si riferiscono a un numero n via via crescente di iterazioni, come indicato in legenda. Gli array graficati sono composti da 1000 punti.

IV. SOTTO-ESERCIZIO 2 (OBBLIGATORIO): PINNA DI SQUALO

L'esercizio 2 qui proposto fa riferimento all'esperienza pratica in cui si osservava l'uscita di un integratore al cui ingresso era inviata un'onda quadra: all'aumentare della frequenza, l'uscita tendeva ad assomigliare sempre più a un'onda triangolare, con un vasto regime intermedio in cui il segnale assumeva le sembianze di una sorta di pinna di squalo. Questa forma d'onda è stata acquisita con Arduino compatibilmente con il rate di campionamento disponibile, per varie frequenze f scelte arbitrariamente in un intervallo di oltre una decade.

Nell'esperimento era $C = 1 \mu\text{F}$ (con tolleranza $\pm 10\%$) e $R = (3.28 \pm 0.03) \text{ kohm}$ (misurata con multimetro digitale), per cui la frequenza di taglio attesa era $f_T = (48.6 \pm 4.9) \text{ Hz}$. Arduino era regolato in modo da campionare a intervalli di durata nominale $\Delta t = 100 \mu\text{s}$ e i record sono stati acquisiti usando la combinazione di sketch e script `syncLong2016`, in modo da ottenere 2048 coppie di punti (tempo e $V_{out}(t)$, quest'ultima lasciata in unità digitalizzate). Per poter impiegare Arduino il generatore di funzioni, impostato su onda quadra (di ampiezza $V_{in} \simeq 4.6 \text{ Vpp}$), è stato regolato in modo da introdurre un offset che rendesse sempre positivo il segnale (in ingresso e in uscita). Infatti in un circuito integratore la componente continua del segnale passa da ingresso a uscita.

La Fig. 5 mostra alcuni esempi dei record acquisiti a diverse frequenze, secondo quanto indicato in legenda. Per esigenze di chiarezza, i grafici sono stati disegnati usando diverse scale per gli assi: in particolare, l'asse orizzontale si estende per 4 periodi, cioè la scala va dallo zero dei tempi (ricordiamo che l'acquisizione è sincrona, per cui lo zero corrisponde sempre, entro l'incertezza, a un fronte di

discesa dell'onda quadra in ingresso) al valore $4/f$, con f frequenza impostata e letta sul frequenzimetro del generatore di funzioni. Poiché l'intervallo di campionamento nominale è sempre lo stesso, la “densità” dei punti rappresentati, cioè il numero di dati effettivamente riportati nei grafici, dipende dalla frequenza, come è facile osservare. Inoltre, poiché anche l'offset del generatore è stato correttamente lasciato inalterato, i grafici a frequenze più alte, dove la componente alternata è attenuata, non partono dallo zero dell'asse verticale.

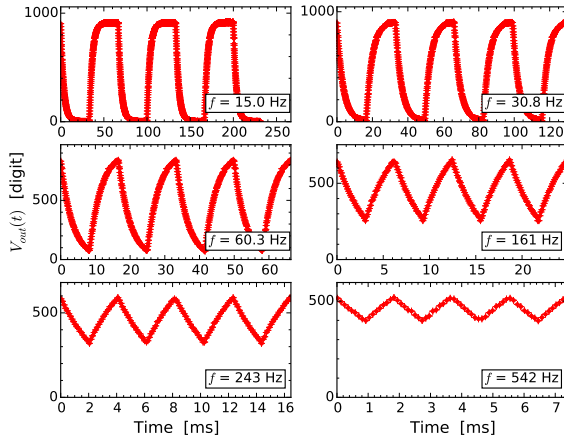


Figura 5. Dati acquisiti sperimentalmente come descritto nel testo: nei vari pannelli sono rappresentate (con punti e barre di errore “convenzionali”) i record $V_{out}(t)$ vs t registrati per diversi valori della frequenza, come indicato in legenda (per semplicità grafica, si omette l'indicazione precisa della frequenza e dell'incertezza sulla sua misura, che è più piccola dell'ultima cifra riportata). Si notino le diverse scale orizzontali e verticali.

Gli andamenti temporali osservati possono facilmente essere interpretati dal punto di vista qualitativo, ragionando in termini di confronto tra i tempi caratteristici di carica e scarica del condensatore e il periodo dell'onda quadra. L'obiettivo di questo esercizio è però quello di simulare questa osservazione in maniera quantitativa usando il “metodo simbolico” e l'espansione di Fourier in seni e coseni.

Grazie all'applicazione del metodo simbolico, sappiamo che un integratore *alimentato con un'onda sinusoidale* di frequenza f dà luogo a:

1. un' *guadagno*, o *attenuazione*, $A(f) = \frac{1}{\sqrt{1 + (f/f_T)^2}}$;
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. Ci occuperemo in seguito della presenza di un offset non nullo, cioè del termine $a_0 \neq 0$ in Eq. 12, e dei problemi di fase iniziale, che nell'acquisizione sperimentale o è random (se si usa la combinazione sketch/script ardu2016) oppure è determinata dalla strategia di sincronizzazione del generatore di funzioni con Arduino (se si usa, come in questo esempio, la combinazione sync/long2016).

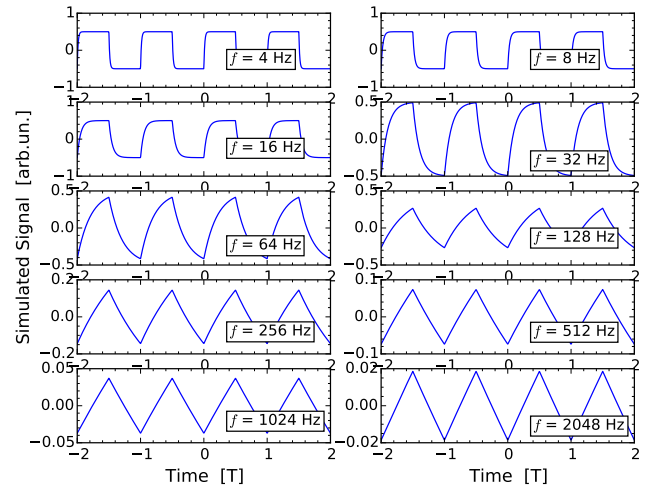


Figura 6. “Simulazione” dell'integratore descritto nel testo ($f_T = 48.6$ Hz) con all'ingresso un'onda quadra. I vari pannelli si riferiscono a diverse frequenze, come in legenda. Fate attenzione alla scala orizzontale, in unità di periodi $T = 1/f$, e alla scala verticale, che cambia di grafico in grafico. Gli array graficati sono composti da 1000 punti e realizzati sommando su 1000 iterazioni.

La Fig. 6 mostra l'esito della “simulazione” supponendo $f_T = 48.6$ Hz, cioè il valore nominale (senza incertezza) dell'integratore realizzato sperimentalmente. I vari pannelli mostrano il segnale simulato per diverse frequenze, come indicato in legenda: le frequenze scalano di un fattore due da un grafico al successivo e non corrispondono ai valori sperimentali, poiché questa figura ha lo scopo principale di illustrare l'evoluzione della forma d'onda da quadra a triangolare all'aumentare della frequenza. Fate attenzione al fatto che le scale orizzontali sono in unità

di periodo $T = 1/f$ e quelle verticali sono diverse per i vari grafici.

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 un'attenuazione dell'ampiezza e da uno sfasamento, che determinano, per $f \gg f_T$, un'uscita di forma pressoché triangolare, evidentemente sfasata di circa $|\pi/2|$ rispetto all'ingresso. Dal punto di vista qualitativo, già per $f \sim 10f_T$ la forma d'onda in uscita è ben approssimata da un andamento triangolare.

La Fig. 7 mostra, sovrapposti, i risultati sperimentali per tre diverse frequenze e quelli ottenuti da simulazioni. Come già sottolineato, ci sono alcune differenze tra l'uscita della simulazione, realizzata come descritto sopra, e i dati acquisiti da Arduino:

1. c'è un fattore di fase costante, che vale circa $\Delta\Phi \simeq \pi$, causato dalla strategia di sincronizzazione: i dati sono infatti acquisiti a partire da un istante nel quale l'onda quadra è nel suo fronte di discesa;
2. c'è un offset, cioè $a_0 \neq 0$ in Eq. 12, dovuto alla circostanza che Arduino accetta in ingresso solo d.d.p. positive (o nulle);
3. c'è un fattore di scala, cioè l'onda quadra in ingresso, e di conseguenza quella in uscita, non hanno ampiezza picco-picco unitaria.

Le tre differenze possono essere facilmente sanate implementando *a mano* (e “a occhio”) alcuni accorgimenti, ad esempio:

1. la forma d'onda simulata viene graficata dopo averla traslata di un semiperiodo (che corrisponde a $\Delta\Phi = \pi$);
2. alla forma d'onda simulata viene aggiunto un termine costante desunto dal valore medio (su un periodo) delle acquisizioni;
3. la forma d'onda simulata viene moltiplicata per un fattore che tiene conto dell'ampiezza del segnale acquisito, desunto dalle registrazioni a bassa frequenza (dove l'attenuazione è trascurabile).

Infine, è ovvio che, allo scopo di migliorare qualitativamente l'accordo tra simulazione e dati sperimentali, il valore della frequenza di taglio f_T impiegata nella simulazione può essere aggiustato all'interno dell'incertezza con cui esso è noto, determinata sostanzialmente dalla tolleranza sulla capacità C del condensatore: le simulazioni di figura sono state calcolate ponendo $f_T = 46$ Hz, compatibile con il valore nominale dato dalla misura di R e dalla conoscenza di C . Naturalmente, la procedura a mano, o “a occhio”, potrebbe essere sostituita da un (complicato) best-fit finalizzato a individuare i valori dei parametri attraverso minimizzazione del χ^2 : tuttavia

esso non è certamente necessario. Infatti, come si può facilmente osservare, la simulazione ottenuta a mano e “a occhio” riproduce in modo qualitativamente soddisfacente i dati sperimentali e ciò è sufficiente per i nostri scopi.

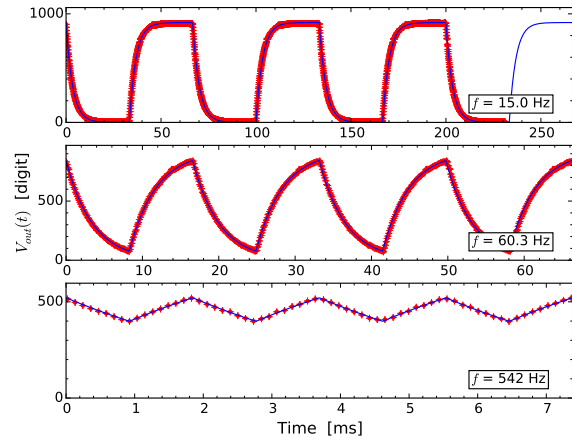


Figura 7. Dati acquisiti sperimentalmente (punti rossi con barre di errore) e simulazioni (linee continue blu) per tre diverse frequenze. Dettagli sull'acquisizione e sulla realizzazione delle simulazioni sono dati nel testo. Si noti, nel pannello superiore, la diversa durata del record sperimentale e di quello simulato.

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 (tutto questo potrebbe costituire l'oggetto di una relazione semestrale).

V. SOTTO-ESERCIZIO 3: AMPIEZZA DELLA FORMA D'ONDA INTEGRATA

Lo svolgimento di questo sotto-esercizio è fortemente consigliato (ma non strettamente obbligatorio). Esso è finalizzato a determinare l'andamento dell'ampiezza in uscita dall'integratore in funzione della frequenza f , sempre supponendo una forma d'onda quadra in ingresso. Il modello dell'integratore, ovvero del filtro passa-basso, permette di predire facilmente tale andamento nel caso di onde sinusoidali attraverso la relazione, già ampiamente citata, $A(f) = 1/\sqrt{1 + (f/f_T)^2}$. Lo sviluppo in armoniche di Fourier può modificare questo andamento a causa della diversa attenuazione a cui sono sottoposte le varie armoniche.

Dal punto di vista sperimentale il guadagno, o attenuazione, $A(f)$ è stato misurato per l'integratore costruito come specificato in precedenza. A questo scopo è stato calcolato il rapporto tra le ampiezze picco-picco (più facili da individuare) del segnale in uscita e di quello in

ingresso, entrambi monitorati con i due canali dell'oscilloscopio. Grazie all'elevata resistenza in ingresso di questo strumento, è stato supposto, senza dimostrazione (questo argomento è trattato più in dettaglio altrove), che l'oscilloscopio perturbasse in maniera trascurabile il circuito. Analogo trattamento è stato riservato alla resistenza di uscita del generatore, che è stata supposta trascurabile.

Dal punto di vista della simulazione, essa è stata fatta girare per diversi valori della frequenza f su un intervallo di alcune decadi, grosso modo corrispondente a quello esplorato sperimentalmente. Nelle simulazioni si è supposta un'ampiezza picco-picco unitaria in ingresso, mentre quella del segnale in uscita è stata dedotta numericamente come differenza tra valore massimo e valore minimo del segnale simulato. Per ottenere previsioni affidabili, si è fatto in modo che le forme d'onda simulate non presentassero "spigolosità" dovute a sotto-campionamento. Inoltre si è scelta $f_T = 46$ Hz, sulla base dei risultati precedenti,

La Fig. 8 mostra i dati sperimentali (punti e barre di errore rossi) con sovrapposti i risultati della simulazione (linea continua blu) e l'andamento previsto per la forma d'onda sinusoidale (linea tratteggiata grigia). È evidente come il comportamento sperimentale sia diverso dalle previsioni per il caso sinusoidale e anche come esso sia qualitativamente ben riprodotto dalla simulazione.

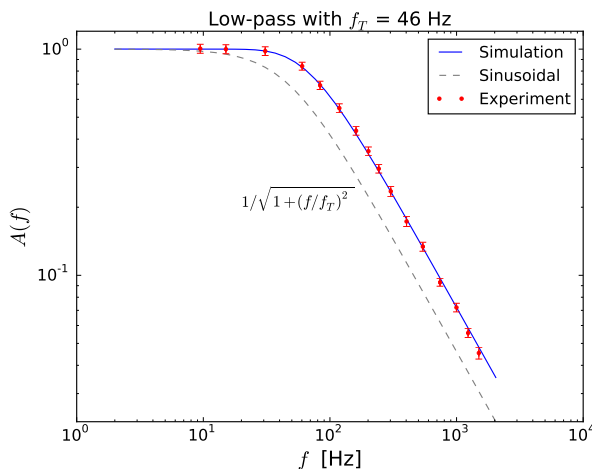


Figura 8. Guadagno, o attenuazione, A in funzione della frequenza f determinato sperimentalmente come descritto nel testo (punti e barre di errore rossi), ottenuto dalla "simulazione" descritta nel testo (la linea continua blu raccorda diversi punti simulati, equispaziati logicamente nell'intervallo di interesse) e dalla previsione per onde sinusoidali (linea tratteggiata grigia), calcolata secondo la funzione riportata nel grafico. Per le previsioni si è supposto $f_T = 46$ Hz, compatibile con il valore sperimentale.

VI. SOTTO-ESERCIZIO 4: INTEGRATORE + DERIVATORE

Anche questo sotto-esercizio, che è un po' più laborioso del precedente, è fortemente consigliato, pur se non strettamente obbligatorio. Esso richiede di considerare il circuito composto da integratore e derivatore in cascata realizzato 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. Come già affermato, gli effetti dovuti al matching tra diversi sotto-circuiti e alle resistenze interne potranno essere considerati in altra sede. 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 *guadagno*, o *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 Eqs. 17, 18. 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. 9, 10 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 grafico in grafico.

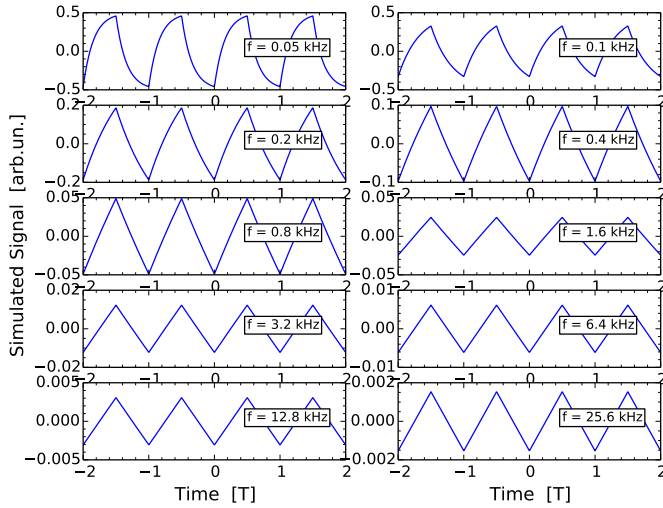


Figura 9. "Simulazione" del segnale in uscita dall'integratore descritto nel testo ($f_{TA} = 50$ Hz) con all'ingresso un'onda quadra. I vari pannelli si riferiscono a diverse frequenze, come in legenda. Fate attenzione alla scala orizzontale, in unità di periodi $T = 1/f$, e alla scala verticale, che cambia di grafico in grafico. Gli array graficati sono composti da 1000 punti e realizzati sommando su 1000 iterazioni.

Cominciamo con l'esaminare l'uscita A (Fig. 9): il risultato è qualitativamente analogo a quello già ottenuto, poiché sempre di integratore si tratta. Si vede chiaramente come all'aumentare della frequenza f l'uscita A tenda ad assumere una forma che approssima sempre meglio la triangolare.

Vediamo ora cosa succede per la forma d'onda simulata in uscita dal derivatore (uscita B, Fig 10). Questo sottocircuito funziona come derivatore solo 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

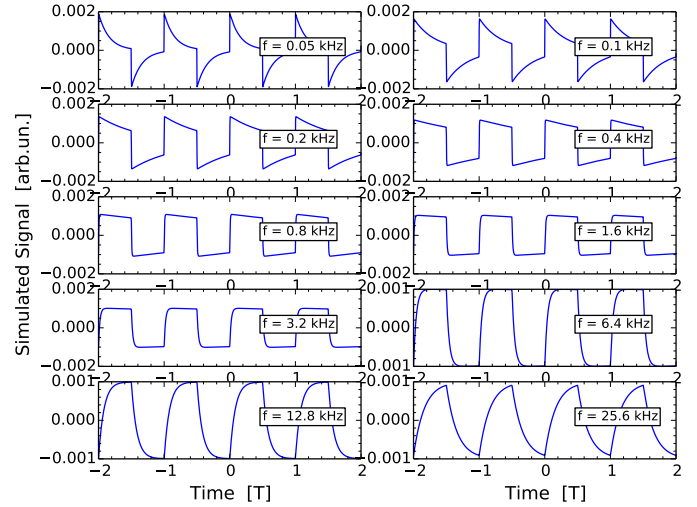


Figura 10. "Simulazione" del segnale in uscita dalla cascata integratore+derivatore descritta nel testo ($f_{TB} = 25$ kHz) con all'ingresso un'onda quadra. I vari pannelli si riferiscono a diverse frequenze, come in legenda. Fate attenzione alla scala orizzontale, in unità di periodi $T = 1/f$, e alla scala verticale, che cambia di grafico in grafico. Gli array graficati sono composti da 1000 punti e realizzati sommando su 1000 iterazioni.

si esce dalla condizione $f \ll f_{TB}$ e la forma d'onda in uscita dal derivatore tende a essere inalterata rispetto a quella che si trova al suo ingresso: dunque essa tende ad essere triangolare, o, se preferite, a pinna di squalo.

Sofferamoci anche qui a esaminare i guadagni, o attenuazioni, tra ampiezza picco-picco in uscita B, cioè del segnale $V_B(t)$, e ampiezza picco-picco in ingresso *all'intero circuito*, cioè $V_{in}(t)$. Ricordiamo che, in questo caso, il valore atteso per onde sinusoidali è $A_{B,att} = f_{TA}/f_{TB}$ indipendente dalla frequenza. Per la scelta delle frequenze di taglio operata nel nostro esempio si ha $A_{B,att} = 2 \times 10^{-3}$: dato che il segnale in ingresso è assunto avere un'ampiezza picco-picco unitaria (in unità arbitrarie), questo vuol dire che il segnale in uscita da B deve avere un'ampiezza di 2×10^{-3} [arb.un.]. Si vede subito da Fig. 10 che questo è approssimativamente quanto si ottiene con la simulazione, in particolare nel range in cui integratore e derivatore si comportano ambedue nel modo atteso. Anche qui l'interpretazione è piuttosto semplice: presi separatamente, i guadagni, o attenuazioni, di integratore e derivatore dipendono dalla frequenza di lavoro in un modo che non segue la semplice aspettativa riferita a onde sinusoidali. Tuttavia i due circuiti sottoattenuano 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. Questo accordo peggiora alle frequenze in cui integratore e derivatore non si comportano pienamente secondo le attese.

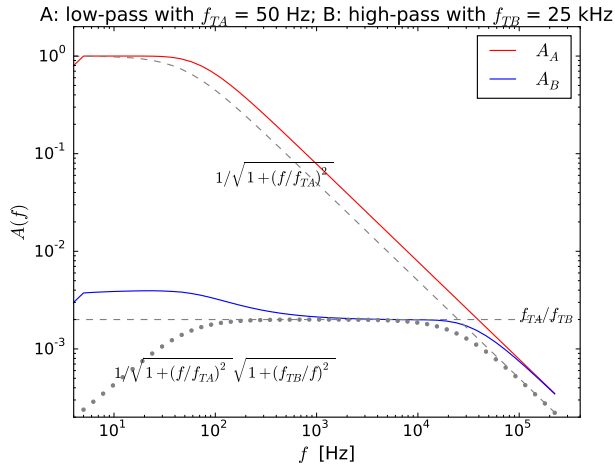


Figura 11. “Simulazione” dei guadagni, o attenuazioni, A_A e A_B in uscita dall’integratore e dalla cascata integratore+derivatore descritta nel testo in funzione della frequenza di lavoro e per i valori di frequenza di taglio riportati nel testo e nel titolo. Le curve tratteggiate e punteggiate rappresentano 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).

Il comportamento complessivo in termini di ampiezza del segnale in uscita è riassunto in Fig. 11: essa è stata costruita come la Fig. 8, cioè eseguendo un ciclo in un range di frequenze (equispaziate logicamente) e de-

ducendo l’ampiezza picco-picco del segnale in uscita (dal sotto-circuito A, linea continua rossa, o dalla cascata dei sotto-circuiti A e B, linea continua blu) come differenza tra valore massimo e minimo della forma d’onda simulata. Si noti che in questo caso la definizione impiegata potrebbe non essere del tutto affidabile, a causa delle spigolosità di alcune forme d’onda simulate (si vedano i primi pannelli di Fig. 10). Il risultato mostra che la cascata integratore+derivatore ha un intervallo di frequenza nel quale l’attenuazione complessiva per l’onda quadra in ingresso è simile a quanto previsto nel caso sinusoidale (il rapporto f_{TA}/f_{TB} indipendente dalla frequenza, indicato con una linea tratteggiata grigia nel grafico). Al di fuori di questo range l’attenuazione segue un andamento che tende a discostarsi da quello previsto per onde sinusoidali (il prodotto $A_A A_B$ rappresentato dalla linea punteggiata grigia nel grafico): in particolare, e anche a causa della definizione di ampiezza picco-picco impiegata, a frequenze basse l’attenuazione è minore (ovvero il guadagno è maggiore, attenzione alle parole) di quanto atteso per le onde sinusoidali. Per il resto, è ovvio che il comportamento dell’integratore, che già abbiamo analizzato in Fig. 8 (in questa figura esso è rappresentato da una linea tratteggiata grigia), gioca un ruolo nel determinare il guadagno complessivo della cascata.

Potrebbe naturalmente essere interessante confrontare le previsioni con dati sperimentali (che io non ho acquisito) e, 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.