

Simulazione numerica di un circuito Integratore/Derivatore RC

L. Pettinari(*)

B. Tomelleri(*)

N. Valori(*)

2019/12/21

Riassunto: — Tramite la piattaforma di sviluppo Arduino[1] confrontiamo campionamenti digitali con simulazioni numeriche, basate sull'analisi di Fourier, di segnali in uscita da circuiti integratori/derivatori RC. Discutiamo la struttura di base e i comportamenti attesi del sistema e ne illustriamo il funzionamento con qualche esempio pratico.

PACS 01.40.-d – Education.

PACS 01.50.Pa – Laboratory experiments and apparatus.

PACS 07.05.Hd – Data acquisition: hardware and software.

1 CAMPIONAMENTO DIGITALE DI SEGNALI

In questa sezione descriviamo il metodo e gli strumenti impiegati nella misura dei segnali elettrici. Si sono costruiti due circuiti RC, un integratore o filtro passa-basso (LPF) e un derivatore o filtro passa-alto (HPF) che indichiamo rispettivamente come circuiti A e B. Prendiamo il valore nominale come riferimento per la capacità e misuriamo con un multimetro digitale le resistenze di entrambi i circuiti, da cui si trova:

$$R_A = 6.72 \pm 0.05 \text{ k}\Omega$$

$$C_A = 2.2 \pm 10\% \text{ }\mu\text{F}$$

$$f_{TA} = \frac{1}{2\pi R_A C_A} = 11 \pm 1 \text{ Hz}$$

$$R_B = 67.5 \pm 0.6 \text{ }\Omega$$

$$C_B = 0.2 \pm 10\% \text{ }\mu\text{F}$$

$$f_{TB} = \frac{1}{2\pi R_B C_B} = 11 \pm 1 \text{ kHz}$$

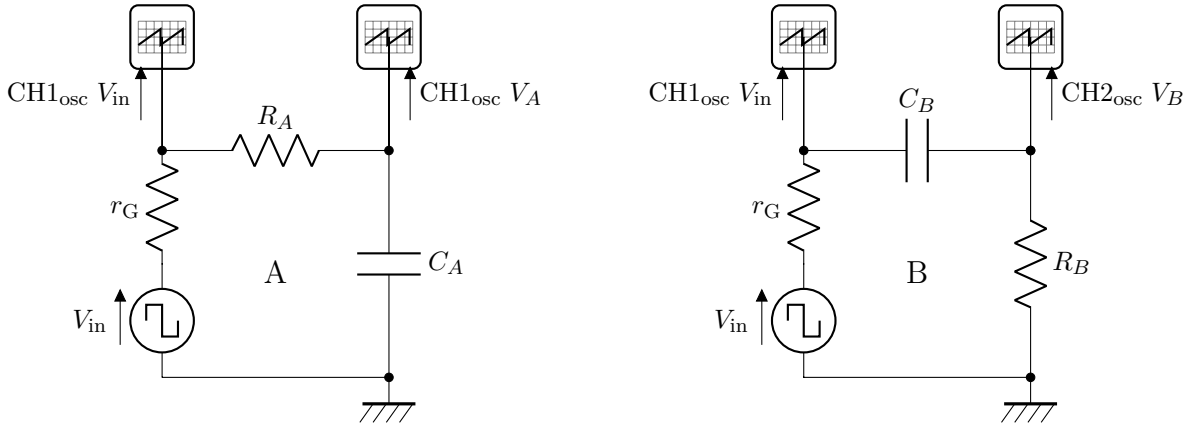
I due circuiti non sono indipendenti quando questi vengono collegati in cascata, poiché $R_B \ll R_A$ la caduta di tensione ai capi di R_B risulta trascurabile, perciò i condensatori si trovano approssimativamente in parallelo: per tenerne conto possiamo definire una capacità efficace per il circuito A+B

$$C_{\text{eff}} = C_A + C_B = 2.4 \pm 10\% \text{ }\mu\text{F}$$

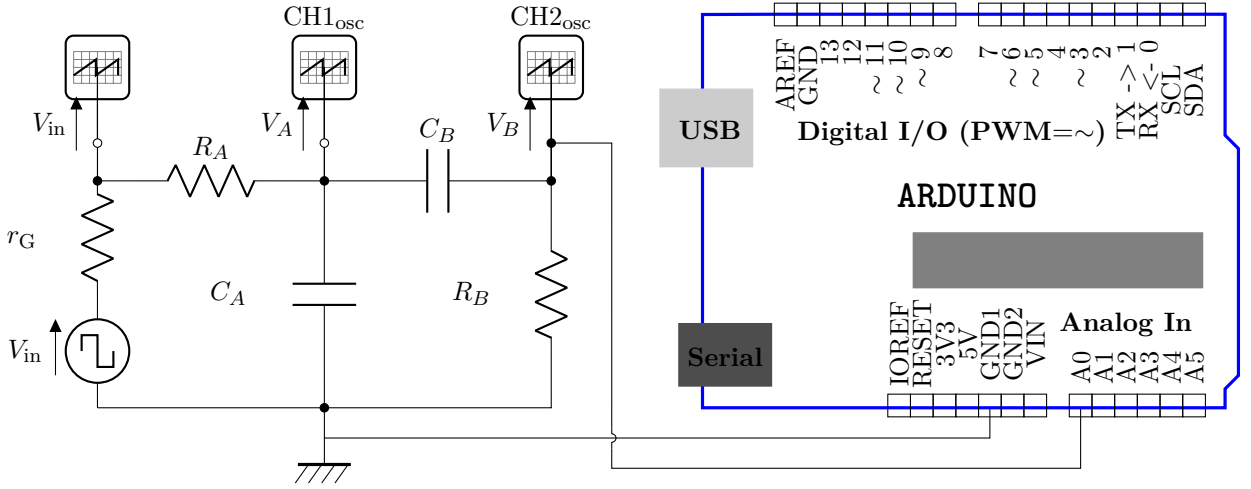
$$f_{TA} = \frac{1}{2\pi R_A C_{\text{eff}}} = 10 \pm 1 \text{ Hz}$$

Ma essendo $C_B \ll C_A$ le frequenze di taglio risultano compatibili e gli effetti dovuti al collegamento si possono effettivamente considerare trascurabili, per cui supponiamo che A e B agiscano indipendentemente sui segnali. Riportiamo lo schema dei circuiti RC dell'integratore A e del derivatore B:

*Dipartimento di Fisica E. Fermi, Università di Pisa - Pisa, Italy



E lo schema circuitale dell'intero apparato strumentale: integratore e derivatore in cascata da cui i segnali in uscita sono campionati dalla porta analogica A0 di Arduino e monitorate dai canali di un oscilloscopio.



2 RICOSTRUZIONE DI FORME D'ONDA

2.1 Onda quadra

Si è implementata in Python l'espressione dell'onda prevista dall'espansione in serie:

$$f(t) = \frac{f_0}{2} + \sum_{k=1}^{\lambda} c_k \sin(\omega_k t) = \frac{f_0}{2} + \sum_{k=0}^{\lambda-1} \frac{2}{(2k+1)\pi} \sin((2k+1)\omega t) \quad (1)$$

```
def sqw(t, f=1, A_pp=1, B=0, phi=0):
    fk = 0
    t+= phi/(2*f)
    for k in range(1, sums, 2):
        c[k] = 2./(k*np.pi)
        w[k] = k*2*np.pi*f
        fk += c[k]*np.sin(w[k]*t)
    t-= phi/(2*f)
    return fk*A_pp + B
```

Si può dimostrare che tutte le funzioni periodiche, che presentano discontinuità di primo tipo o "a scalino", una volta sviluppate tramite la serie di Fourier troncata ad un numero finito λ di termini, presentano delle "sovraelongazioni" intorno ai punti di discontinuità, dove la somma parziale assume valori superiori alla funzione ricostruita. Questo è noto in letteratura come Fenomeno di Gibbs[2], da cui sappiamo che, anche nel limite in cui $\lambda \rightarrow \infty$ la sovraelongazione della serie tende ad una quantità finita, ovvero è riducibile ma ultimamente ineliminabile. Nella nostra ricostruzione dunque valutiamo il numero di termini necessari per evitare errori sistematici introdotti da queste discontinuità studiando la differenza fra i valori dell'onda modello e di quella sviluppata con Fourier, come sempre facendo riferimento alla somma dei residui quadrati come indicatore della qualità di riproduzione.

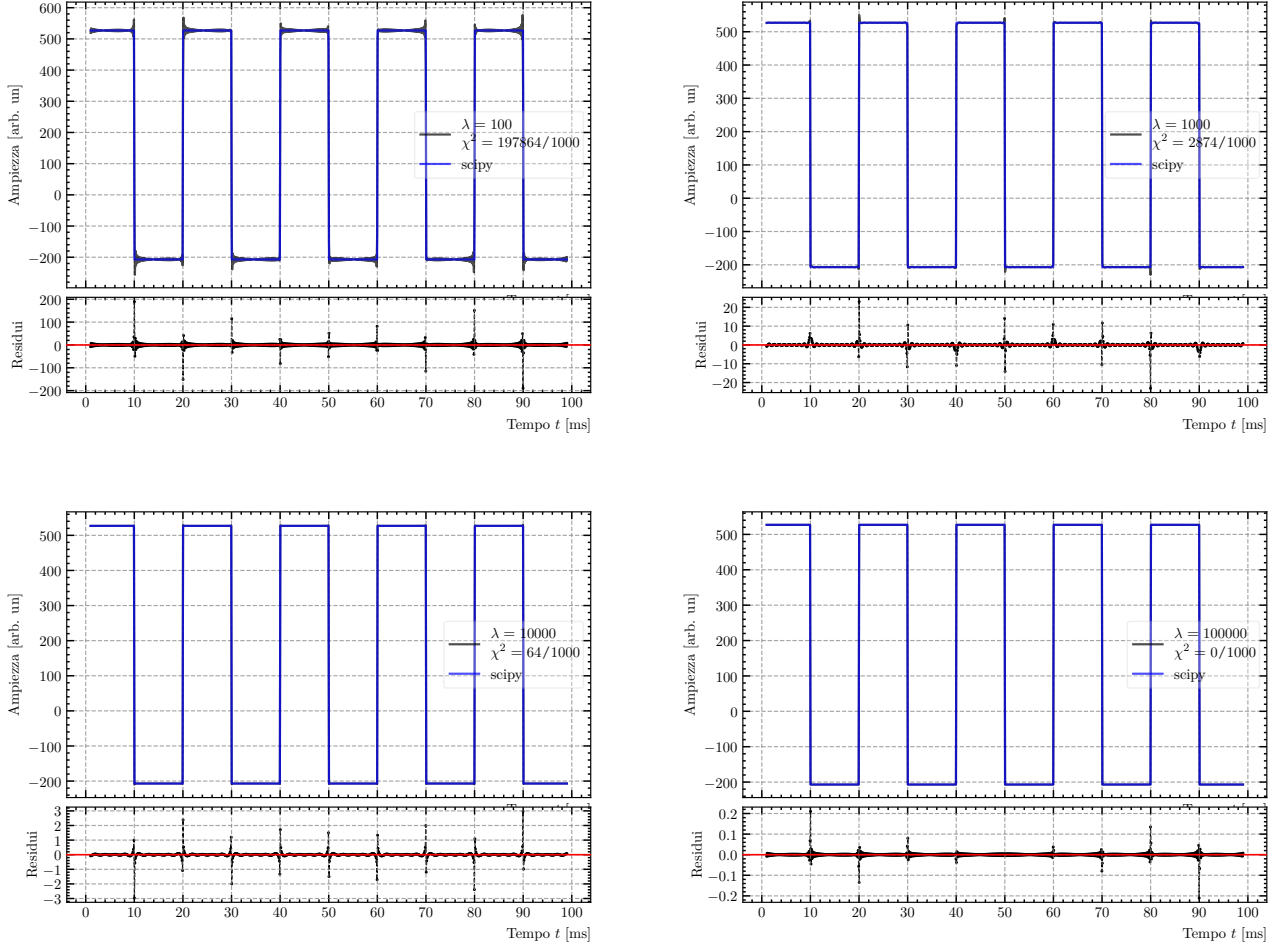


Figura 1: Sviluppo in serie di Fourier per un'onda quadra per λ iterazioni confrontato con un onda quadra definita analiticamente da un if-else. L'asse dei tempi è sempre un array da 1000 punti equispaziati linearmente.

2.2 Onda triangolare

Si è implementata in Python l'espressione dell'onda prevista dallo sviluppo in serie di Fourier:

$$g(t) = \frac{g_0}{2} + \sum_{k=1}^{\lambda} b_k \cos(\omega_k t) = \frac{g_0}{2} + \sum_{k=0}^{\lambda-1} \left(\frac{2}{(2k+1)\pi} \right)^2 \cos((2k+1)\omega t) \quad (2)$$

```
def trg(t, f=1, A_pp=1, B=0, phi=0):
    t += phi/(2*f)
    fk = 0
    for k in range(1, sums, 2):
        c[k] = (2./(k*np.pi))**2
        w[k] = k*2*np.pi*f
        fk += c[k]*np.cos(w[k]*t)
    t -= phi/(2*f)
    return fk*A_pp + B
```

Da uno studio sulla qualità della forma d'onda analogo al caso precedente, dalla figura 2 si nota subito come il numero di termini necessari perché i residui quadrati dell'onda siano inferiori all'unità si ha già per $\lambda \leq 1000$, due ordini di grandezza in meno rispetto a prima. Questo è un ovvia conseguenza del fatto che, l'onda triangolare non è affetta dal fenomeno di Gibbs, in quanto priva di punti di discontinuità a salto.

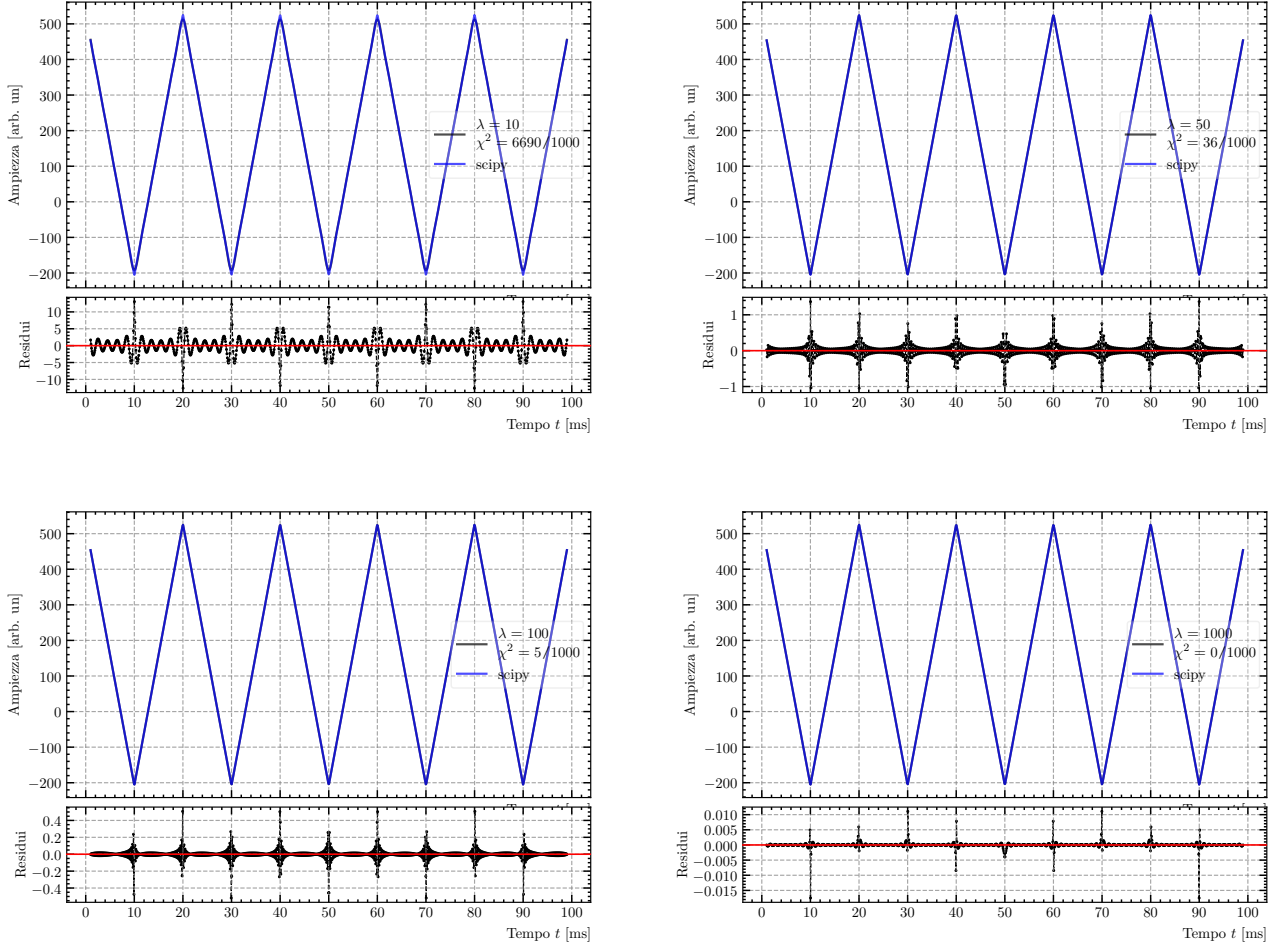


Figura 2: Sviluppo in serie di Fourier per un'onda triangolare per λ iterazioni confrontato con una stessa definita analiticamente da un if-else. L'asse dei tempi è sempre un array da 1000 punti equispaziati linearmente.

2.3 Onda a pinna di squalo

Si è implementata in Python l'espressione dell'onda, applicando la $T(\omega)$ ad ogni armonica nella serie dell'onda quadra (1) secondo quanto previsto dal metodo dei fasori per un'onda sinusoidale in uscita da un integratore RC:

$$\begin{cases} \text{Attenuazione } A_A(\omega_k) &= \frac{1}{\sqrt{1+(\omega_k/\omega_{T_A})^2}} \\ \text{Sfasamento } \Delta\varphi_A(\omega_k) &= \arctan(-\omega_k/\omega_{T_A}) \end{cases} \quad (3)$$

```
def intg(f, fT=fT_a):
    return 1./np.sqrt(1.+(f/fT)**2)
def fin(t, f=1, A_pp=1, B=0, phi=0):
    t+= phi/(2*f)
    fk = 0
    for k in range(1, sums, 2):
        c[k] = 2./(k*np.pi)
        w[k] = k*2*np.pi*f
        A[k] = intg(w[k], wT_a)
        Df[k] = np.arctan(-w[k]/wT_a)
        fk += A[k]*c[k]*np.sin(w[k]*t + Df[k])
    t-= phi/(2*f)
    return fk*A_pp + B
```

Dunque si sono acquisite le forme d'onda in uscita dal nostro circuito A, con sampling rate stabilito dallo sketch di Arduino, regolato in modo da campionare a intervalli di durata nominale $\Delta t = 100 \mu s \Rightarrow$ corrispondente ad un rate di campionamento $f_s = 10 \text{ kSa/s}$. Per confrontare i risultati della simulazione con quanto osservato sperimentalmente otteniamo i parametri ottimali dell'onda da un fit per la "pinna di squalo" sugli stessi dati a cui verrà sovrapposta:

$$\begin{aligned}
 f &= 50.219 \pm 0.005 \text{ Hz} \\
 A &= 727.8 \pm 0.7 \text{ digit} \\
 B &= 159.11 \pm 0.07 \text{ digit} \\
 \varphi &= 1000.6 \pm 0.5 \text{ m } \pi \text{rad} \\
 \sigma_{f,A} &= 0.01 \\
 \sigma_{f,B} &= -0.16 \\
 \sigma_{f,\varphi} &= -0.86 \\
 \sigma_{A,B} &= -0.01 \\
 \sigma_{A,\varphi} &= -0.007 \\
 \sigma_{B,\varphi} &= 0.14 \\
 \chi^2 &= 2339/881 \\
 \text{abs_sigma} &= \text{True}
 \end{aligned}$$

Riportiamo quindi il risultato della "simulazione" in figura 3. Si noti inoltre come questi valori fungano, nel resto del testo, da parametri di riferimento per ogni altra ricostruzione d'onda, come già visto in 1 e in seguito per la figura 5.

Cambiando i parametri a piacere, ma fissando lo stesso intervallo nei tempi, dalla figura 4 si vede come la pinna

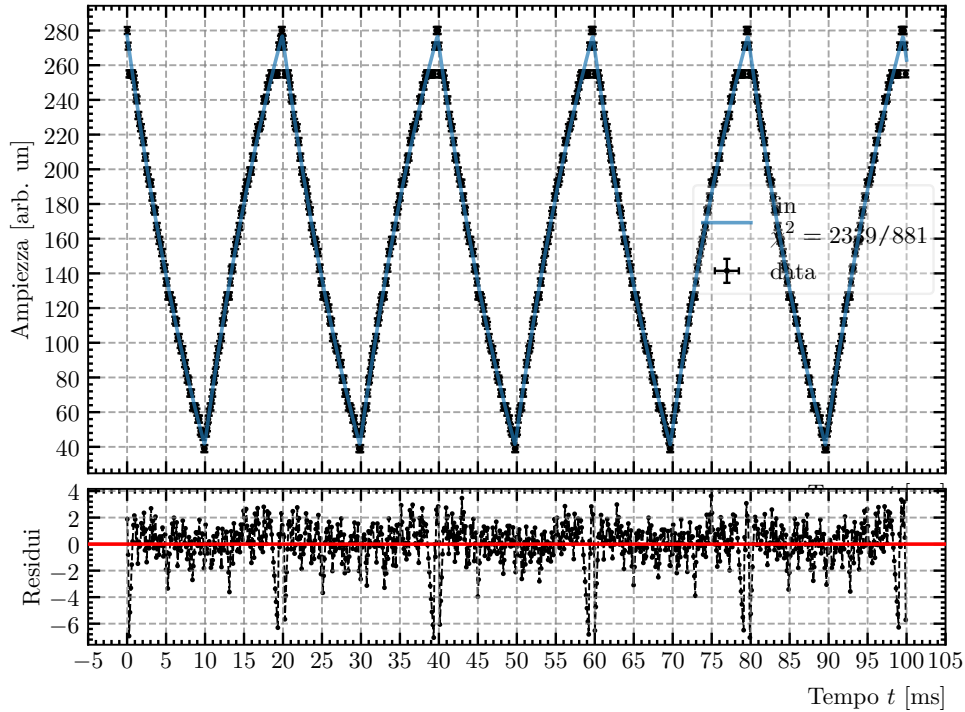


Figura 3: Sovrapposizione dei campionamenti di Arduino con la ricostruzione della pinna di squalo

di squalo ricostruita con il metodo simbolico continui a descrivere -bene- i dati sperimentali. Per averne una descrizione più quantitativa, operando di nuovo un *fit* per i parametri ottimali dei segnali si trova:

$f = 69.626 \pm 0.008 \text{ Hz}$	$f = 169.958 \pm 0.009 \text{ Hz}$
$A = 591.7 \pm 1.0 \text{ digit}$	$A = 289.9 \pm 0.3 \text{ digit}$
$B = 212.17 \pm 0.07 \text{ digit}$	$B = 179.29 \pm 0.13 \text{ digit}$
$\varphi = 1039.6 \pm 0.9 \text{ m } \pi\text{rad}$	$\varphi = 968.2 \pm 0.6 \text{ m } \pi\text{rad}$
$\chi^2 = 4253/882$	$\chi^2 = 893/252$
abs_sigma = True	abs_sigma = True

Si noti come i punti considerati "outliers", a più di 2.5 barre d'errore di distanza dal modello, si dispongano

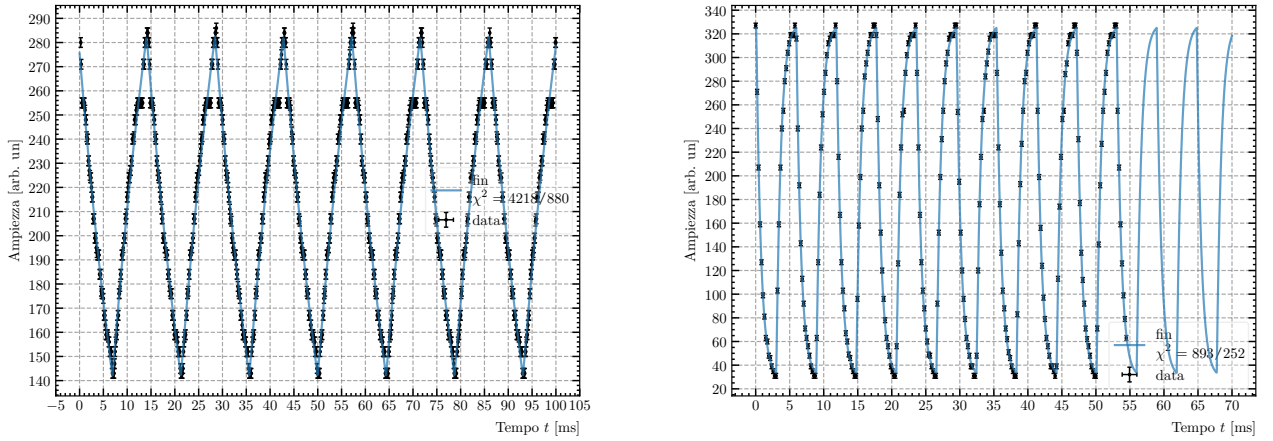


Figura 4: Sovrapposizione dei campionamenti di Arduino con la ricostruzione della pinna di squalo per due altre frequenze, $f_1 = 70 \text{ Hz}$ e $f_2 = 170 \text{ Hz}$.

sempre alla stessa altezza nell'onda, come lungo una retta costante: in effetti su ogni periodo i punti attorno al valore $255 = 2^8 - 1$ assumono andamento "a scalino" a seguito della digitalizzazione di Arduino. Questo si deve al difetto caratteristico degli ADC, per cui il sistema mostra una qualche "riluttanza" a far scattare il bit successivo (i.e. il nono) nella lettura della tensione. Dunque il test del χ^2 potrà essere indice della bontà del fit solo una volta operata la rimozione di questi punti e altri (ove necessario) che si devono ad artefatti introdotti dal sistema di campionamento.

2.4 Serie di derivatore e integratore

Supponiamo trascurabili sia gli effetti delle resistenze interne dei componenti sia quelli di mismatch tra i sottocircuiti A e B collegati in cascata, per cui secondo quanto previsto dal metodo simbolico si aggiungono, all'espressione dell'onda in uscita da A (3), le trasformazioni delle armoniche dovute al derivatore B:

$$\begin{cases} \text{Attenuazione } A_B(\omega_k) &= \frac{1}{\sqrt{1 + (\omega_{TB}/\omega_k)^2}} \\ \text{Sfasamento } \Delta\varphi_B(\omega_k) &= \arctan(\omega_{TB}/\omega_k) \end{cases} \quad (4)$$

```
def derv(f, fT_int=fT_a, fT_der=fT_b):
    return intg(f, fT_int)*1./np.sqrt(1.+(fT_der/f)**2)
def cas(t, f=1, A_pp=1, B=0, phi=0):
    t+= phi/(2*f)
    fk = 0
    for k in range(1, sums, 2):
        c[k] = 2./(k*np.pi)
        w[k] = k*2*np.pi*f
        A[k] = derv(w[k], wT_a, wT_b)
```

```

Df[k] = np.arctan(-w[k]/wT_a) + np.arctan(wT_b/w[k])
fk += A[k]*c[k]*np.sin(w[k]*t + Df[k])
t-= phi/(2*f)
return fk*A_pp + B

```

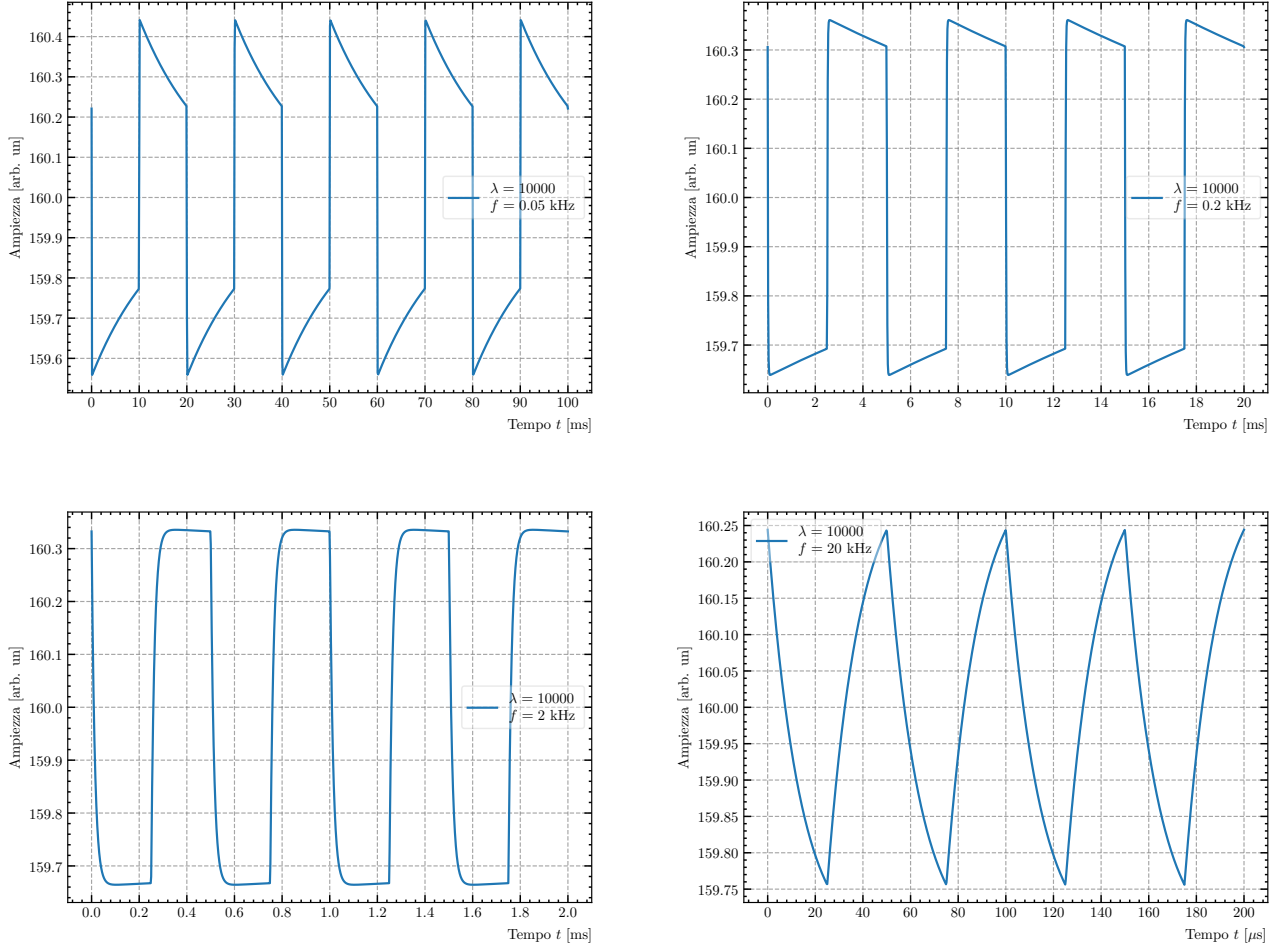


Figura 5: Sviluppo in serie di Fourier per l'onda in uscita dalla cascata di integratore e derivatore A+B per λ iterazioni al variare della frequenza f . L'asse dei tempi è sempre un array da 1000 punti equispaziati linearmente.

Confrontare come prima i risultati della simulazione con quanto osservato sperimentalmente non è più facilmente possibile per via del limitato rate di campionamento del sistema di acquisizione.

2.5 Accoppiamento AC dell'oscilloscopio

L'accoppiamento AC in ingresso all'oscilloscopio taglia la componente continua del segnale visualizzato mediante un condensatore di capacità C_{AC} in serie al segnale che, una volta carico, impedisce il passaggio di corrente. Assieme alla resistenza di ingresso dell'oscilloscopio (nel nostro caso $R_{osc} = 1 \text{ M}\Omega$ nominale¹) tra segnale e linea di terra, il condensatore realizza un circuito derivatore RC/HPF. Di conseguenza, in accoppiamento AC, forme d'onda quadre a bassa frequenza $\sim f_T = (2\pi R_{osc} C_{AC})^{-1}$ assumono la forma caratteristica dell'esponenziale di carica/scarica del condensatore. Dalle specifiche tecniche sappiamo che in AC l'oscilloscopio attenua le componenti del segnale sotto i 10 Hz, per cui ci si aspetta $C_{AC} \approx 4 \text{ nF}$. Si è ricostruita questa distorsione con lo stesso metodo simbolico applicando attenuazione e sfasamento a un'onda quadra scritta in armoniche di Fourier, dunque si confronta quanto previsto con il campionamento dell'onda eseguito dall'oscilloscopio digitale stesso

¹ In realtà lo stadio di ingresso dell'oscilloscopio prevede il parallelo tra R_{osc} e un piccolo condensatore $C_{osc} = 20 \text{ pF}$ nominali per modellare capacità parassite dovute ai connettori e ai circuiti interni allo strumento. Dunque lo stadio di ingresso ha una impedenza di ingresso data da $Z_{osc} = R_{osc} + \frac{1}{j\omega C_{osc}}$. I moduli delle impedenze dei due componenti in parallelo diventano paragonabili per $\omega = 5 \times 10^4 \text{ rad s}^{-1}$. Perciò trascurare la presenza del condensatore C_{osc} è, a rigore, giustificato fintanto che la frequenza di lavoro è $f \ll 300 \text{ kHz}$

che opera la distorsione. Come prima si ottengono i parametri di best *fit* per la forma d'onda mediante la stessa routine di minimizzazione del χ^2 , stavolta però imponendo `abs_sigma = False`, in quanto non è altrettanto nota la natura della fonte principale di errore sulle misure. Nella figura 6 si vede la sovrapposizione e simulazione di campionamento all'oscilloscopio, con rate di campionamento $f_s = 0.1$ MSa/s, di un'onda quadra di frequenza $f = 40$ Hz sottoposta all'azione di un filtro passa-alto con $f_T = 10$ Hz. Dunque ripetendo lo stesso fit una volta

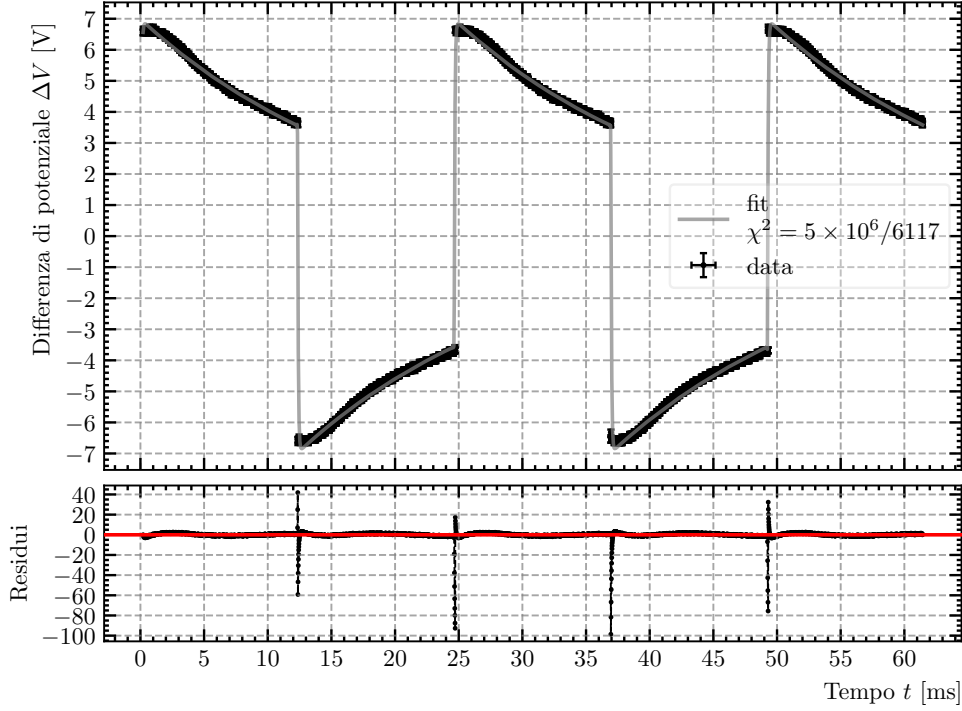


Figura 6: Fit dell'onda quadra distorta in accoppiamento AC da campionamento con oscilloscopio digitale

rimossi gli outlier a più di 4 barre d'errore di distanza nelle zone più spigolose del modello si ottiene un test del χ^2 ragionevole, come illustrato in figura 7

$$\begin{aligned}
 f &= 40.647 \pm 0.003 \text{ Hz} \\
 A &= 3.62 \pm 0.05 \text{ V} \\
 B &= 17.8 \pm 1.2 \text{ mV} \\
 \varphi &= 1996.7 \pm 0.2 \text{ m rad} \\
 f_{TA} &= 8.69 \pm 0.01 \text{ Hz} \\
 f_{TB} &= 2.99 \pm 0.04 \text{ kHz} \\
 \sigma_{f,A} &= 0.22 & \sigma_{A,B} &= 0.004 \\
 \sigma_{f,B} &= -0.04 & \sigma_{A,\varphi} &= -0.37 \\
 \sigma_{f,\varphi} &= -0.92 & \sigma_{A,f_{TA}} &= -0.26 \\
 \sigma_{f,f_{TA}} &= -0.01 & \sigma_{A,f_{TB}} &= 0.99 \\
 \sigma_{f,f_{TB}} &= 0.23 & \sigma_{B,\varphi} &= 10^{-4} \\
 \sigma_{B,f_{TA}} &= 0.03 & \sigma_{\varphi,f_{TA}} &= 0.01 \\
 \sigma_{\varphi,f_{TB}} &= -0.37 & \sigma_{f_{TA},f_{TB}} &= -0.17 \\
 \chi^2 &= 6483/6060 \\
 \text{abs_sigma} &= \text{False}
 \end{aligned}$$

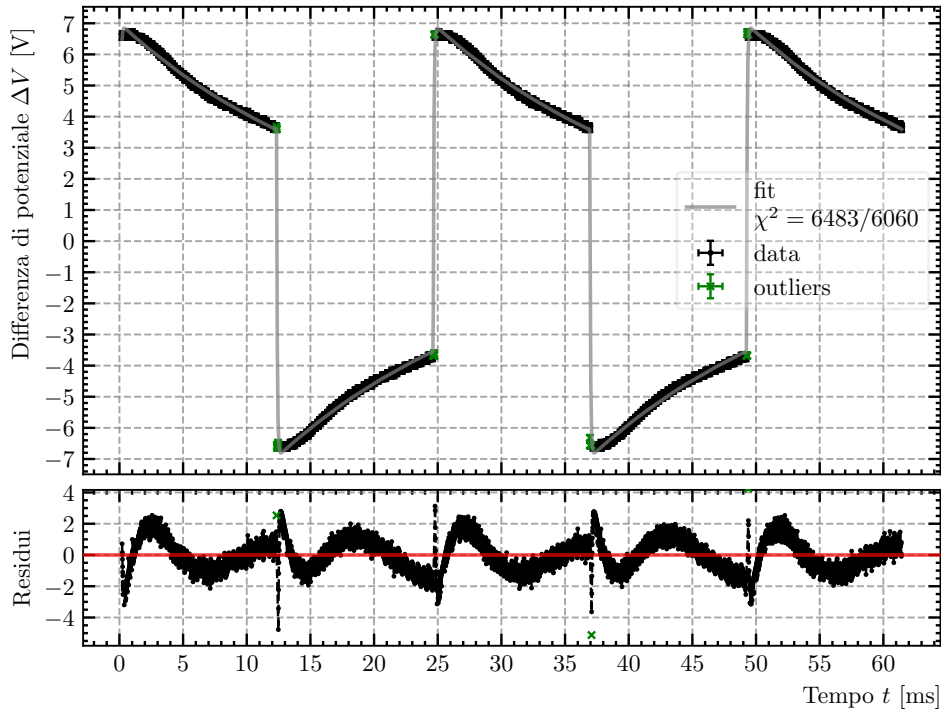


Figura 7: Fit dell'onda quadra distorta in accoppiamento AC con rimozione degli outlier

3 SIMULAZIONE DELL'ATTENUAZIONE IN RISPOSTA DI UN CIRCUITO RC

Si è misurato il guadagno $A(f)$ per l'integratore A ed il derivatore B, calcolando il rapporto tra le ampiezze segnale in uscita e di quello in ingresso al circuito, entrambi monitorati con i due canali dell'oscilloscopio. Supponiamo che questo, per merito della sua alta resistenza in ingresso, perturbi in maniera trascurabile il sistema e che altrettanto trascurabili siano gli effetti dovuti alla resistenza in uscita dal generatore di funzioni ($r_G = 50 \Omega$ nominali). Dunque si è simulato lo stesso guadagno sulle onde quadre ricostruite numericamente per 400 valori della frequenza di lavoro f : Supponendo un'ampiezza picco-picco unitaria in ingresso, si è dedotta l'ampiezza in uscita come differenza tra valore massimo e minimo della serie. Dalla figura 8 si vede come il comportamento sperimentale sia diverso dalle previsioni per un segnale sinusoidale (linea tratteggiata grigia) e anche come esso sia, almeno per frequenze entro i 10 kHz, qualitativamente ben riprodotto dalla simulazione. Si vede anche come, per frequenze più alte, i punti si discostano dall'andamento atteso, a causa di capacità parassite che cessano di essere trascurabili, al contrario di quanto presuppone il nostro modello, dunque la simulazione. Per poterne modellare gli effetti si potrebbe, ad esempio, considerare un condensatore in parallelo alla resistenza² R_A che, formando una maglia impropria aggiungerebbero uno zero alla risposta in frequenza del circuito, che così assumerebbe un andamento più vicino a quello delle misure ad alte frequenze.

Nota sul metodo di fit

Per determinare i parametri ottimali e le rispettive varianze si è implementato un metodo di fit basato sui minimi quadrati mediante la funzione `curve_fit` di Python. Per tutti i fit con campionamenti digitali di Arduino si è imposto `abs_sigma = True` in quanto come incertezza si è assunto il valore convenzionale $\sigma = \pm 1$ [digit], per cui effettivamente si esegue un fit dei minimi quadrati.

² In effetti i capi della resistenza, come qualsiasi altri due conduttori percorsi da corrente e vicini tra loro formano dei condensatori: per le dimensioni dei nostri circuiti, di capacità nell'ordine della decina di pF, i cui effetti risultano apprezzabili solo per frequenze superiori alle migliaia di Hz. Per raffinare ulteriormente il modello si potrebbero anche considerare le resistenze in parallelo ai condensatori, per tenere conto delle "perdite" di carica che si determinano nel dielettrico reale al loro interno, di conseguenza modificando a loro volta l'attenuazione.

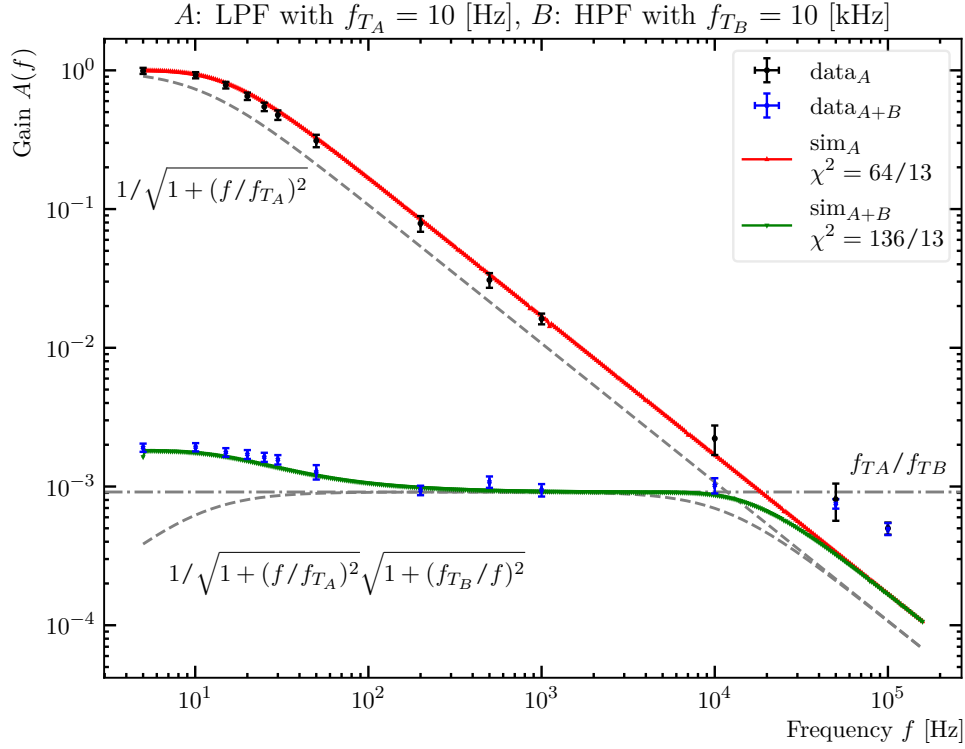


Figura 8: "Ricostruzione" delle attenuazioni in uscita dall'integratore e dalla cascata integratore+derivatore in funzione della frequenza di lavoro e di taglio (riportate nel titolo). Le curve tratteggiate e punteggiate rappresentano gli andamenti attesi per onde sinusoidali (le espressioni scritte sul grafico). Il grafico si riferisce a 400 valori di frequenza distinti ed equispaziati logaritmicamente

RIFERIMENTI BIBLIOGRAFICI

- [1] I. D. I. Ivrea *et al.* Arduino: Open-source electronic prototyping platform. Ivrea, Italy. [Online]. Available: <https://www.arduino.cc/>
- [2] Wikipedia contributors. (2019) Gibbs phenomenon — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Gibbs_phenomenon&oldid=931444514. [Online; accessed 26-December-2019].