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:

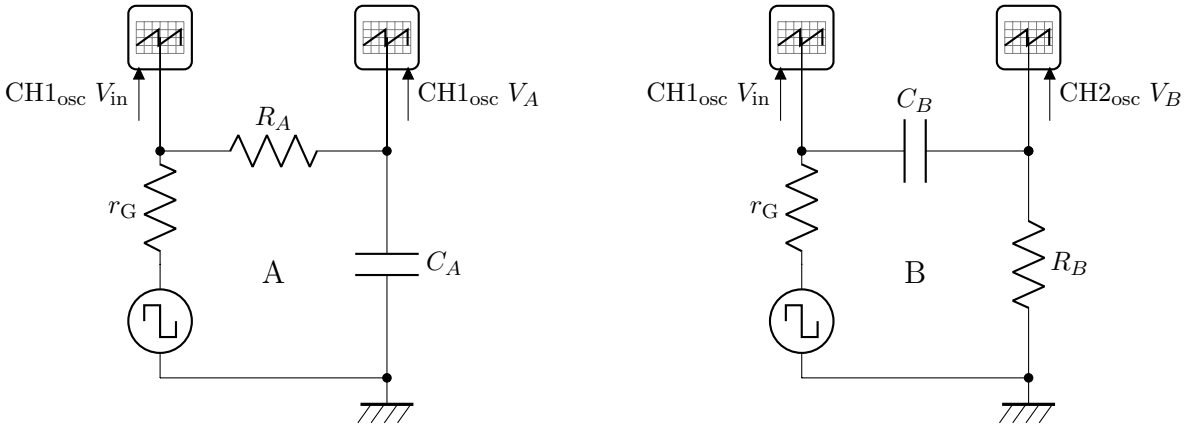
$$R_A = 6.72 \pm 0.05\ \text{k}\Omega$$
$$C_A = 2.2 \pm 10\%\ \mu\text{F}$$
$$f_{\text{TA}} = \frac{1}{2\pi R_A C_A} = 11 \pm 1\ \text{Hz}$$

$$R_B = 67.5 \pm 0.6\ \Omega$$
$$C_B = 0.2 \pm 10\%\ \mu\text{F}$$
$$f_{\text{TB}} = \frac{1}{2\pi R_B C_B} = 12 \pm 1\ \text{kHz}$$

I due circuiti non sono indipendenti, quando questi vengono collegati in cascata i condensatori si trovano in parallelo, dunque possiamo definire una capacità efficace per il circuito A+B

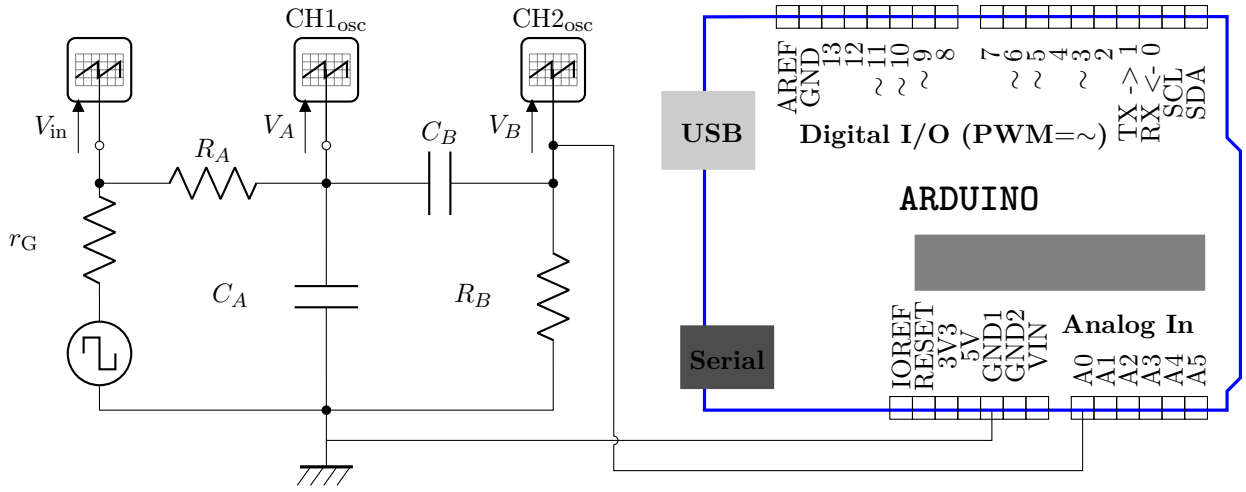
$$C_{\text{eff}} = C_A + C_B = 2.4 \pm 10\%\ \mu\text{F}$$
$$f_{\text{TA}} = \frac{1}{2\pi R_A C_{\text{eff}}} = 10 \pm 1\ \text{Hz}$$

Lo schema dei circuiti RC dell'integratore A e del derivatore B:



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

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:

```
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 $\lambda \longrightarrow \infty$ 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 discontuinità 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.

2.2 *Onda triangolare*

Si è implementata in Python l'espressione dell'onda:

```
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

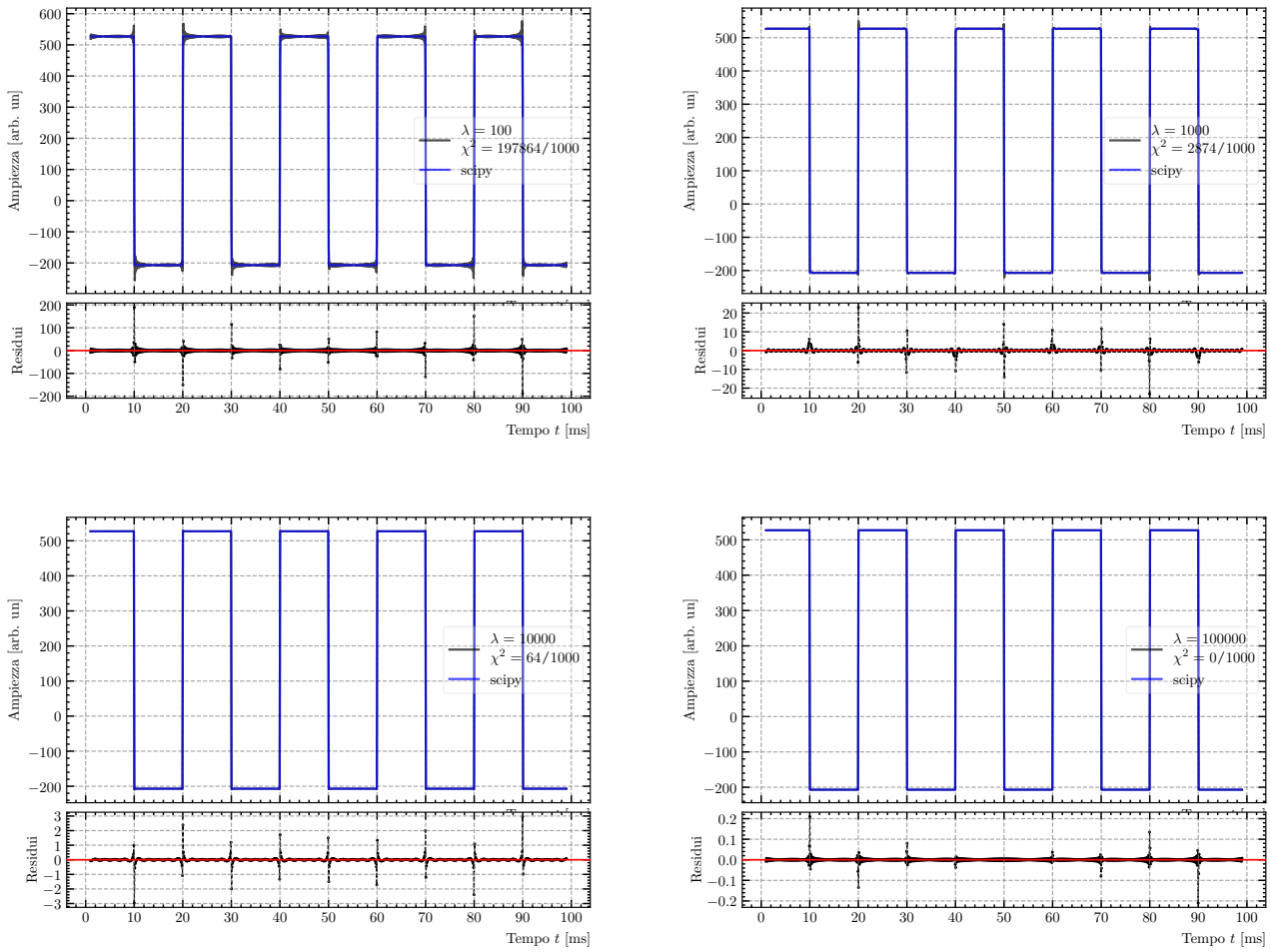


Figura 1: Sviluppo in serie di Fourier per un'onda quadra per λ iterazioni confrontato con un onda quadra definita analiticamente da un if-else. L'asse dei tempi è sempre un array da 1000 punti equispaziati linearmente.

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.

2.3 *Onda a pinna di squalo*

Si è implementata in Python l'espressione dell'onda:

```
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
```

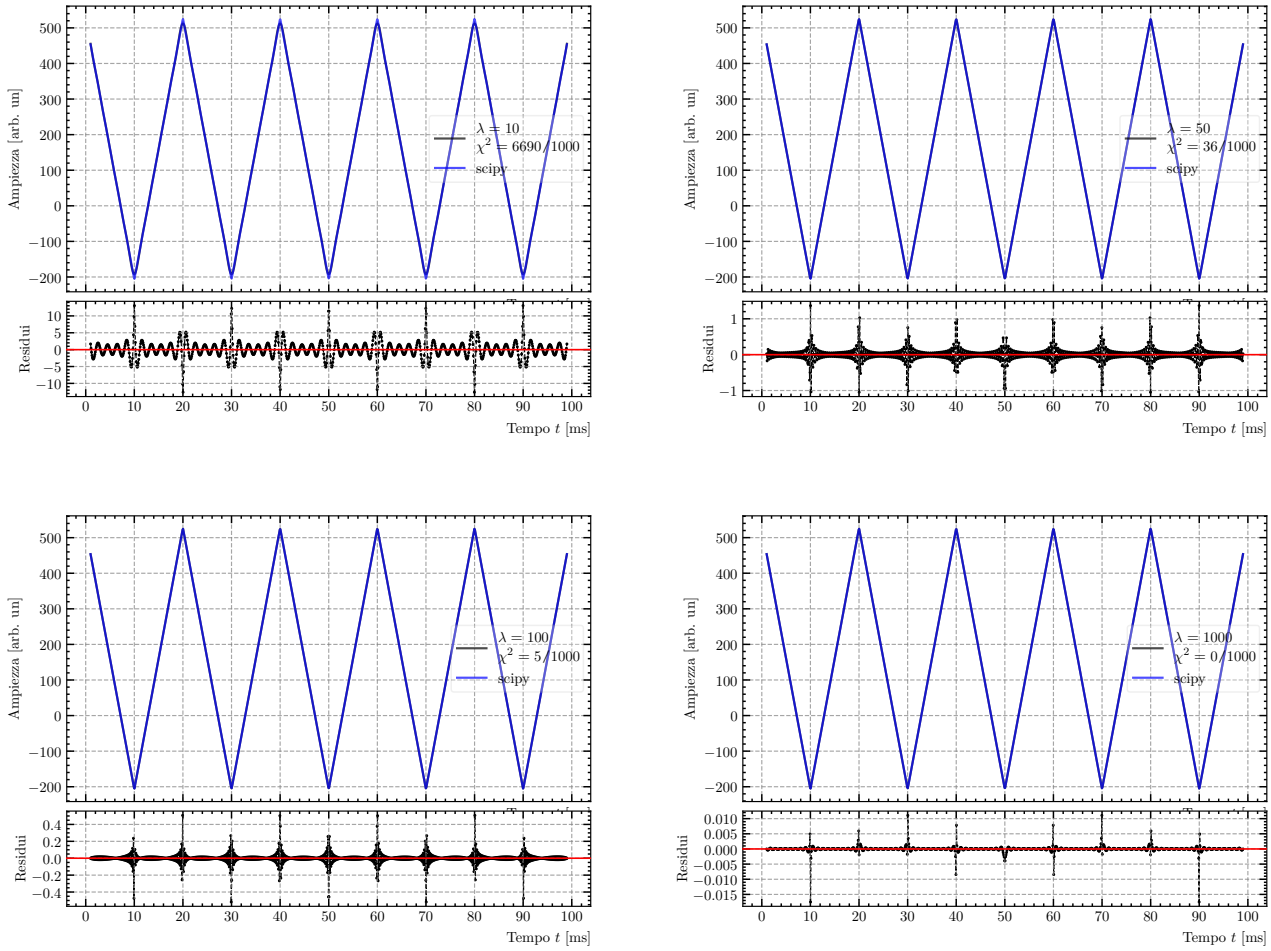


Figura 2: Sviluppo in serie di Fourier per un'onda triangolare per λ iterazioni confrontato con una stessa definita analiticamente da un if-else. L'asse dei tempi è sempre un array da 1000 punti equispaziati linearmente.

Dunque 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:

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

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

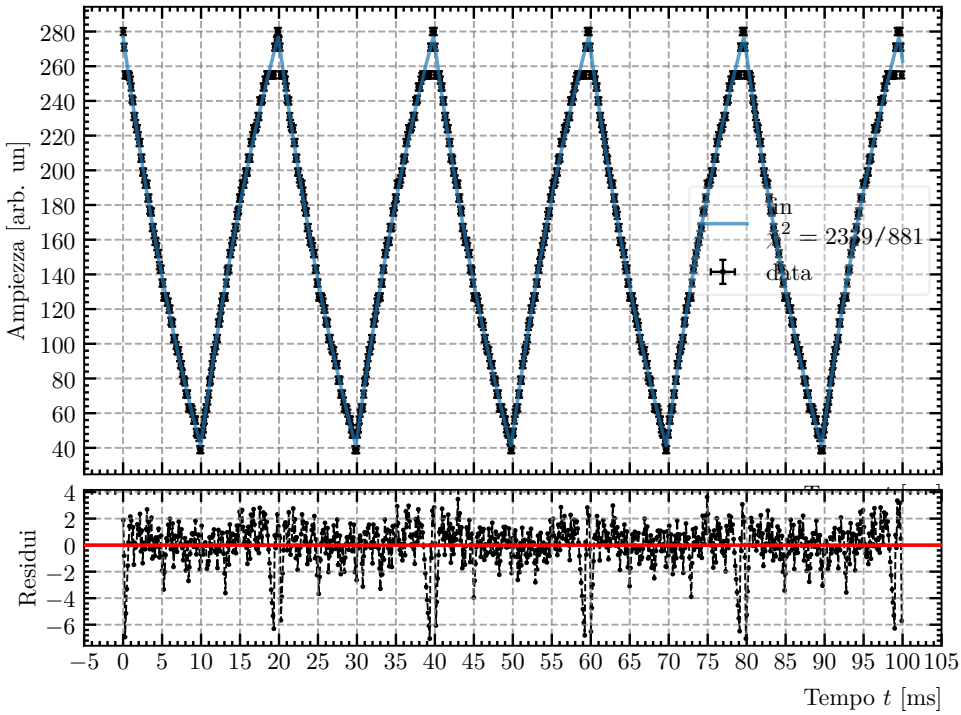


Figura 3: Sovrapposizione dei campionamenti di Arduino con la ricostruzione della pinna di squalo

di squalo ricostruita con il metodo simbolico continui a descrivere -bene- i dati sperimentali:

$$f = 69.626 \pm 0.008 \text{ Hz}$$
$$A = 591.7 \pm 1.0 \text{ digit}$$
$$B = 212.17 \pm 0.07 \text{ digit}$$
$$\varphi = 1039.6 \pm 0.9 \text{ m } \pi\text{rad}$$
$$\chi^2 = 4253/882$$
$$\text{abs\_sigma} = \text{True}$$

$$f = 169.958 \pm 0.009 \text{ Hz}$$
$$A = 289.9 \pm 0.3 \text{ digit}$$
$$B = 179.29 \pm 0.13 \text{ digit}$$
$$\varphi = 968.2 \pm 0.6 \text{ m } \pi\text{rad}$$
$$\chi^2 = 893/252$$
$$\text{abs\_sigma} = \text{True}$$

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

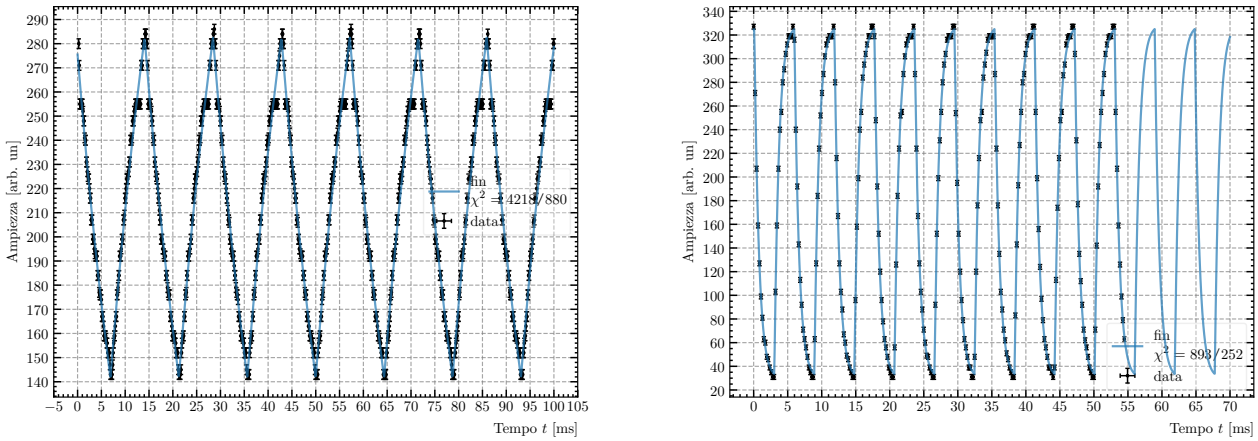


Figura 4: Sovrapposizione dei campionamenti di Arduino con la ricostruzione della pinna di squalo per due altre frequenze, $f_1 = 70$ Hz e $f_2 = 170$ Hz.

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. Dal metodo simbolico sappiamo che gli effetti sulle armoniche in uscita da A andranno ad aggiungersi a quelli dovuti al derivatore B, per cui in uscita ci aspettiamo un'espressione del tipo:

```
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
```

Dunque, come prima per confrontare i risultati della simulazione con quanto osservato sperimentalmente otteniamo i parametri ottimali dell'onda quadra da un fit sugli stessi dati a cui verrà sovrapposta:

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 $R_{\mathrm{osc}} = 1\mathrm{M}\Omega$ nominale[1]) 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 $\sim f_T = (2\pi R_{\mathrm{osc}} C_{\mathrm{AC}})^{-1}$, 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 $C_{\mathrm{AC}} \approx 4$ nF. Questa distorsione può essere ricostruita con lo stesso metodo simbolico applicando attenuazione e sfasamento a un'onda quadra scritta in armoniche di Fourier. Nella figura 5 si simula la visualizzazione all'oscilloscopio 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 rimossi

[1] In realtà lo stadio di ingresso dell'oscilloscopio prevede il parallelo tra R_{osc} e un piccolo condensatore $C_{\mathrm{osc}} = 20pF$ 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 $Z_{\mathrm{osc}} = R_{\mathrm{osc}} + \frac{1}{j\omega C_{\mathrm{osc}}}$. I moduli delle impedenze dei due componenti in parallelo diventano comparabili per $\omega = 5 \times 10^4 \mathrm{rad\,s^{-1}}$. Perciò trascurare la presenza del condensatore C_{osc} è, a rigore, giustificato fintanto che la frequenza di lavoro è $f << 300$ kHz

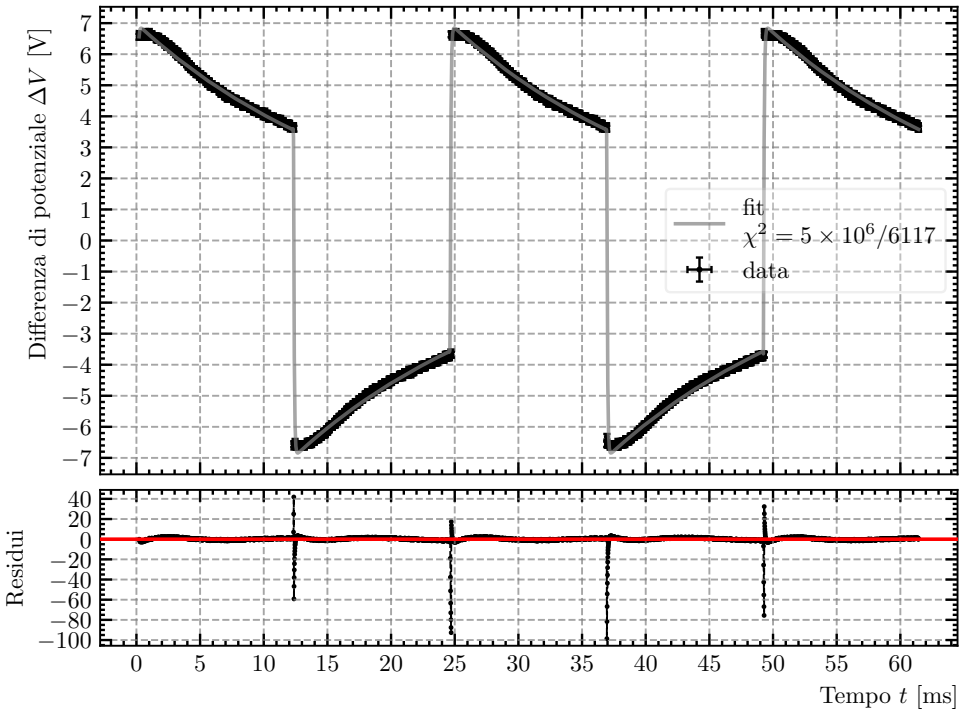


Figura 5: Fit dell'onda quadra distorta in accoppiamento AC da campionamento con oscilloscopio digitale

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 6

$$
\begin{aligned}
f &= 40.647 \pm 0.003 \text{ Hz} \\
A &= 3.62 \pm 0.05 \text{ V} \\
B &= 17.8 \pm 1.2 \text{ mV} \\
\varphi &= 1996.7 \pm 0.2 \text{ m } \pi\text{rad} \\
f_{\text{TA}} &= 8.69 \pm 0.01 \text{ Hz} \\
f_{\text{TB}} &= 2.99 \pm 0.04 \text{ kHz}
\end{aligned}
$$

$$
\begin{aligned}
\sigma_{f,A} &= 0.22 & \sigma_{A,B} &= 0.004 \\
\sigma_{f,B} &= -0.04 & \sigma_{A,\varphi} &= -0.37 \\
\sigma_{f,\varphi} &= -0.92 & \sigma_{A,f_{\text{TA}}} &= -0.26 \\
\sigma_{f,f_{\text{TA}}} &= -0.01 & \sigma_{A,f_{\text{TB}}} &= 0.99 \\
\sigma_{f,f_{\text{TB}}} &= 0.23 & \sigma_{B,\varphi} &= 10^{-}4 \\
\sigma_{B,f_{\text{TA}}} &= 0.03 & \sigma_{\varphi,f_{\text{TA}}} &= 0.01 \\
\sigma_{\varphi,f_{\text{TB}}} &= -0.37 & \sigma_{f_{\text{TA}},f_{\text{TB}}} &= -0.17
\end{aligned}
$$

$$
\begin{aligned}
\chi^2 &= 6483/6060 \\
\text{abs\_sigma} &= \text{False}
\end{aligned}
$$

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

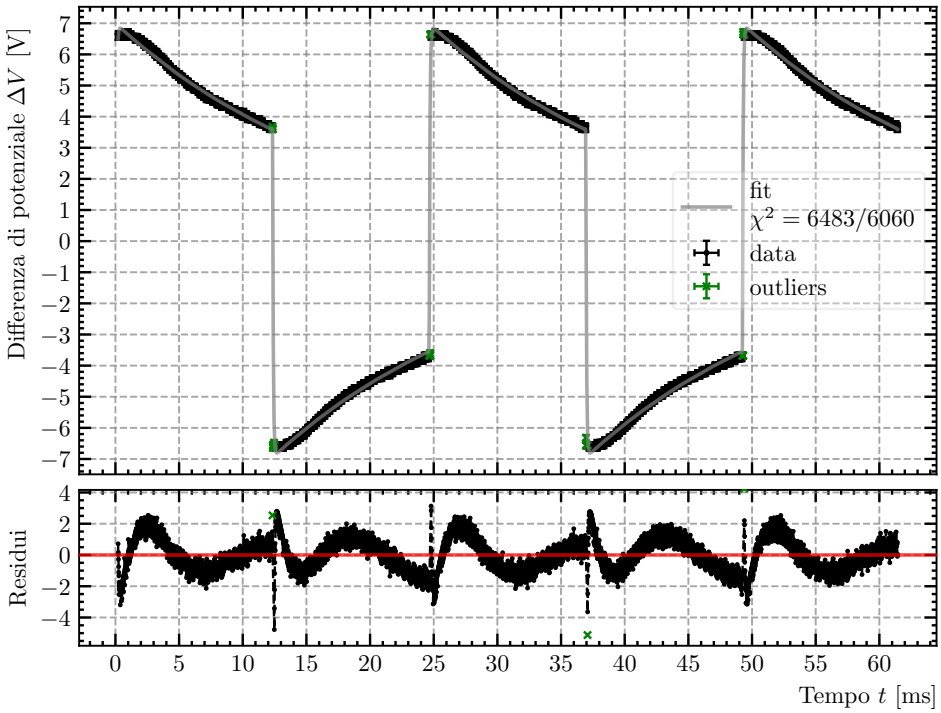


Figura 6: Fit dell'onda quadra distorta in accoppiamento AC con rimozione degli outlier

($r_G = 50\ \Omega$ 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 7 si vede come il comportamento sperimentale sia diverso dalle previsioni per il caso sinusoidale (linea tratteggiata grigia) e anche come esso sia, almeno per frequenze entro i 10^4 Hz, 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.

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 prende il valore convenzionale $\sigma = \pm 1$ [digit], per cui effettivamente si sta eseguendo un fit dei minimi quadrati.

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].

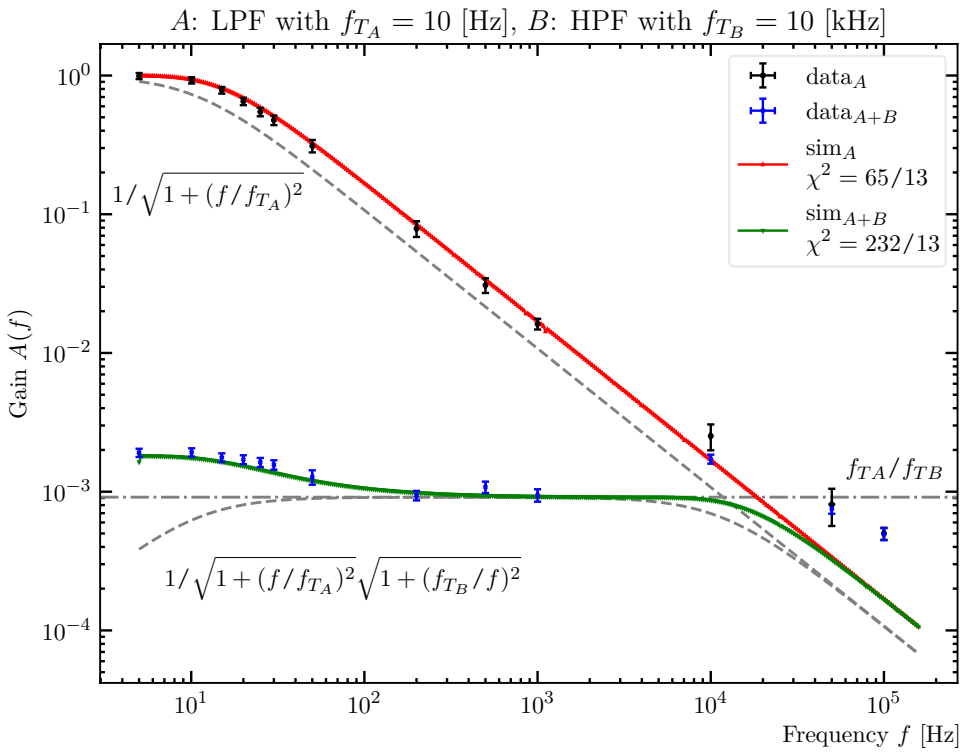


Figura 7: "Ricostruzione" delle attenuazioni in uscita dall'integratore e dalla cascata integratore+derivatore in funzione della frequenza di lavoro e di taglio (riportate nel titolo). Le curve tratteggiate e punteggiate rappresentano gli andamenti attesi per onde sinusoidali (le espressioni scritte sul grafico). Il grafico si riferisce a 400 valori di frequenza distinti ed equispaziati logaritmicamente