

Esercitazione sullo sviluppo in serie di Fourier

Francesco Zazzu

Lo scopo dell'esercizio è quello di ricostruire tramite *serie di Fourier* varie forme d'onda. Per i nostri scopi sarà della forma

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

nell'esercizio faremo riferimento sempre a forme d'onda alternate (a parte dove indicato) dunque il coefficiente a_0 risulterà nullo. Per quanto riguarda gli altri due coefficienti si ha:

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

1 Sviluppo onda quadra e triangolare

Consideriamo la forma d'onda quadra (la nostra $g(t)$), avendo optato per un'onda dispari tutti i coefficienti b_k risulteranno nulli. Il calcolo dei vari c_k è stato effettuato con la seconda delle (2), da cui risulta $c_k = \frac{2}{k\pi}$ per k dispari, essendo l'area sottesa dalla curva $g(t)\sin(\omega_k t)$ nel secondo pannello (in verde nella figura (1)) nulla. La figura (1) riporta dunque le armoniche fondamentali (in rosso), la forma d'onda quadra (generata con il pacchetto *signal* di *scipy*) e appunto la curva $g(t)\sin(\omega_k t)$.

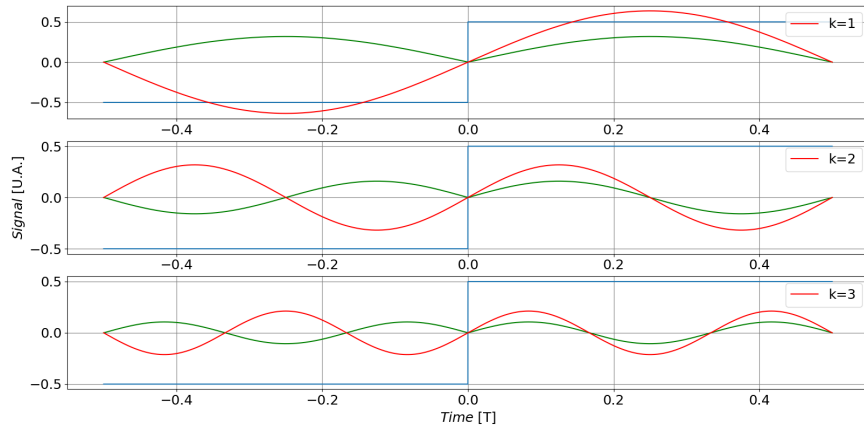


Figura 1: Calcolo dei primi 3 coefficienti della *serie di Fourier* per un'onda quadra di ampiezza unitaria alternata. Il segnale si estende per un periodo.

Per quanto riguarda la forma d'onda triangolare risultano nulli i coefficienti c_k , dunque usando la prima equazione delle (2) si ha $b_k = (\frac{2}{k\pi})^2$. Anche sta volta si hanno valori di k dispari.

Riproduciamo ora le 2 forme d'onda. Entrambe sono state graficate in funzione delle unità di periodo e sono costituite da 10000 punti. Quella quadrata è data dalla funzione

$$g(t) = \sum_{k=1}^n \frac{2}{k\pi} \sin(\omega_k t), \quad k \text{ dispari}$$

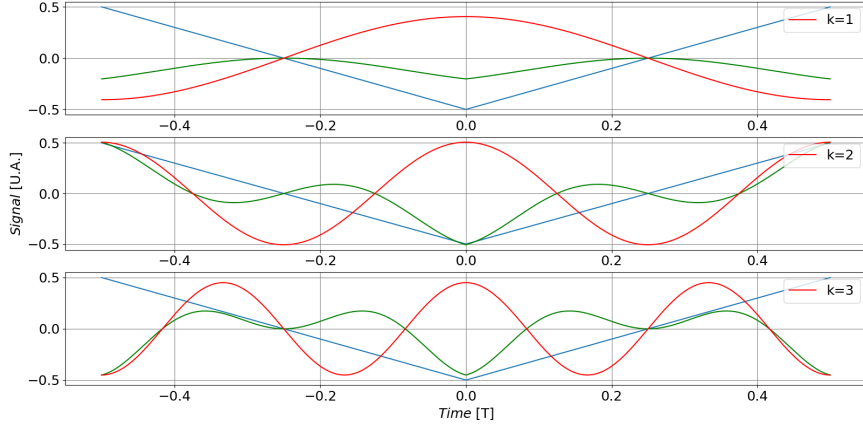


Figura 2: Calcolo dei primi 3 coefficienti della *serie di Fourier* per un'onda triangolare di ampiezza unitaria alternata. Il segnale si estende per un periodo.

ed è mostrata in figura (3).

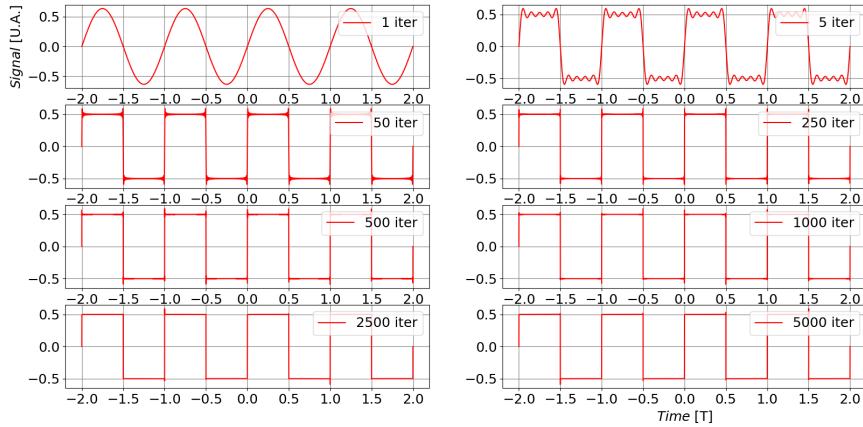


Figura 3: Sviluppo in *serie di Fourier* per un'onda quadra per varie iterazioni.

Il segnale triangolare è stato generato usando la funzione

$$g(t) = \sum_{k=1}^n \left(\frac{2}{k\pi}\right)^2 \cos(\omega_k t), \quad k \text{ dispari}$$

ed è mostrato in figura (4).

Come è possibile osservare, a dispetto di quella quadra, l'onda triangolare è già ben ricostruita dopo 10 iterazioni.

2 Sviluppo onda a pinna di squalo

In una precedente esperienza pratica si è osservata l'uscita di un integratore al cui ingresso si mandava un'onda quadra. Il filtro era costituito da un condensatore di capacità $C = (2.2 \pm 10\%) \mu F$ nominali ed una resistenza $R = (6.76 \pm 0.06) k\Omega$ misurata col multimetro digitale. Abbiamo dunque una frequenza di taglio $f_T = (11 \pm 2) Hz$. Un integratore alimentato

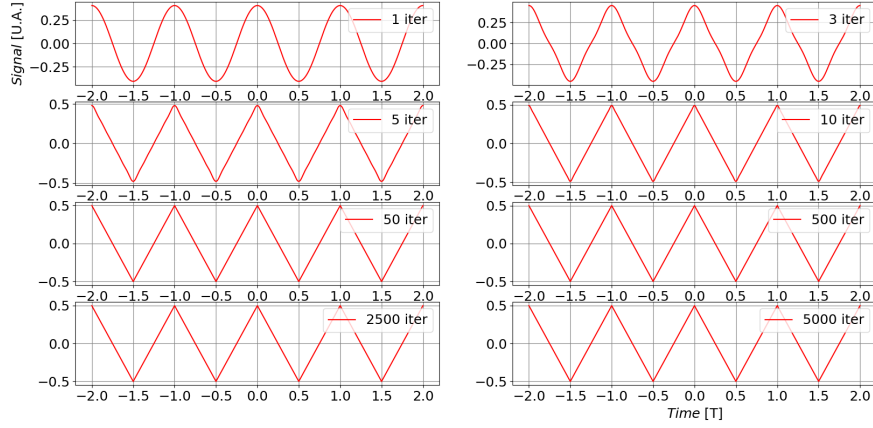


Figura 4: Sviluppo in *serie di Fourier* per un'onda triangolare per varie interazioni.

con un'onda sinusoidale da luogo a un'attenuazione (o guadagno) del segnale e ad un certo sfasamento dati da:

$$A(f_k) = \frac{1}{\sqrt{1 + (f_k/f_T)^2}} \quad \Delta\varphi(f_k) = \arctan(-f_k/f_T). \quad (3)$$

Da queste due equazioni, e dallo sviluppo in *serie di Fourier* dell'onda quadra otteniamo lo sviluppo in serie per simulare il segnale a pinna di squalo:

$$\omega(t) = \sum_{k=1}^n A(f_k) \frac{2}{k\pi} \sin(\omega_k t + \Delta\varphi(f_k)), \quad k \text{ dispari con } \omega_k = 2\pi f_k$$

La figura (5) riporta i dati acquisiti da Arduino per varie frequenze.

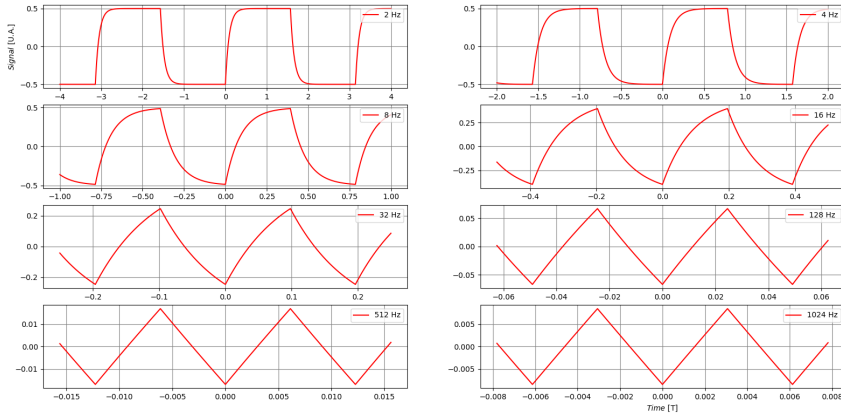


Figura 5: Sviluppo in *serie di Fourier* di una forma d'onda a pinna di squalo per varie frequenze. La scala orizzontale è sempre in unità di periodo. Si osserva come all'aumentare della frequenza di lavoro la forma d'onda tenda a diventare triangolare per $f \gg f_T$ in quanto l'integratore lavora bene al di sotto della frequenza di taglio.

Nella figura (6) si sono sovrapposti i dati acquisiti (blu e barre d'errore grigie) con il segnale simulato (rosso). Ovviamente l'onda in uscita dall'integratore risulta sfasata di un certo fattore

rispetto a quella simulata, in più essa non è alternata e non ha un'ampiezza unitaria . Son stati aggiunti dunque un offset, una fase e alcuni coefficienti al segnale simulato determinati con stime "a mano". Lo script con cui son stati generati i pannelli della figura (8) è il seguente:

```
import pylab
import numpy as np
from scipy.optimize import curve_fit
from scipy import signal
import matplotlib.pyplot as plot

Directory='./'

omt=11.

U1= np.zeros(10000)
U2= np.zeros(10000)
U3= np.zeros(10000)
U4= np.zeros(10000)

om=7.31
T=2*np.pi/om
t=np.linspace(0.3,4*T,10000)

for n in range (1,1000, 2):
    U1=U1+(2/(n*np.pi))*1/(np.sqrt(1+(n*om/omt)**2))*np.sin(n*t*om+np.arctan(-n*om/omt))

pylab.subplot(2,2,1)
FileName=(Directory+'intqdr1')
x,y = pylab.loadtxt(FileName,unpack='True')
x=x*(0.9*10**(-5))+0.3
Dx = np.full(x.size, 4e-6)
Dy = np.full(y.size, 25.)
pylab.errorbar(x,y,Dy,Dx,linestyle='-',color='blue',ecolor='gray',label='Record')
pylab.grid(color = 'gray')
pylab.ylabel(r'$V_{out}$ [digit]')
pylab.plot(t,567.*U1+491,color='red', label = "f=7.31 Hz")
pylab.legend(loc='upper right')

om=17.36
T=2*np.pi/om
t=np.linspace(0.5,5*T,10000)
for n in range (1,1000, 2):
    U2=U2+(2/(n*np.pi))*1/(np.sqrt(1+(n*om/omt)**2))*np.sin(n*t*om+np.arctan(-n*om/omt))

pylab.subplot(2,2,2)
FileName=(Directory+'intqdr2')
x,y = pylab.loadtxt(FileName,unpack='True')
x=x*(0.9*10**(-5))+0.5
Dx = np.full(x.size, 4e-6)
Dy = np.full(y.size, 10.)
pylab.errorbar(x,y,Dy,Dx,linestyle='-',color='blue',ecolor='gray',label='Record')
pylab.grid(color = 'gray')
pylab.plot(t,472.+U2*460.,color='red', label = "f=17.36 Hz")
pylab.legend(loc='upper right')

om=34.71
```

```

T=2*np.pi/om
t=np.linspace(0.4,7*T,10000)
for n in range (1,1000, 2):
    U3=U3+(2/(n*np.pi))*1/(np.sqrt(1+(n*om/omt)**2))*np.sin(n*t*om+np.arctan(-n*om/omt))

pylab.subplot(2,2,3)
FileName=(Directory+'intqdr3')
x,y = pylab.loadtxt(FileName,unpack='True')
x=x*(0.9*10**(-5))+0.5
Dx = np.full(x.size, 4e-6)
Dy = np.full(y.size, 5.)
pylab.errorbar(x,y,Dy,Dx,linestyle='-',color='blue',ecolor='gray',label='Record')
pylab.grid(color = 'gray')
pylab.plot(t,481+U3*450.,color='red', label = "f=34.71 Hz")
pylab.legend(loc='upper right')

om=52.32
T=2*np.pi/om
t=np.linspace(0.4,8*T,10000)
for n in range (1,1000, 2):
    U4=U4+(2/(n*np.pi))*1/(np.sqrt(1+(n*om/omt)**2))*np.sin(n*t*om+np.arctan(-n*om/omt))

pylab.subplot(2,2,4)
FileName=(Directory+'intqdr4')
x,y = pylab.loadtxt(FileName,unpack='True')
x=x*(0.9*10**(-5))+0.4
Dx = np.full(x.size, 4e-6)
Dy = np.full(y.size, 5.)
pylab.errorbar(x,y,Dy,Dx,linestyle='-',color='blue',ecolor='gray',label='Record')
pylab.grid(color = 'gray')
pylab.plot(t,443*U4+479,color='red', label = "f=52.32 Hz")
pylab.legend(loc='upper right')

pylab.rc('font',size=18)
pylab.xlabel(r'$Time$ [T]')
pylab.grid(color = 'gray')

# show the plot
pylab.show()

```

Consideriamo ora l'attenuazione a cui è sottoposto il segnale, il modello dell'integratore permette di predire l'andamento nel caso di onde sinusoidali. lo sviluppo in serie di Fourier può modificare quest'andamento da causa della diversa attenuazione a cui sono sottoposte le varie armoniche. La figura (7) mostra l'andamento dell'attenuazione, in funzione della frequenza, predetta dal modello (linea tratteggiata) e quella invece ricavata tramite fit dei minimi quadrati. I dati per il fit sono stati ricavati dal tra le ampiezze picco-picco del segnale in uscita (V_{out}) e di quello in ingresso (V_{in}).

3 Cascata integratore-derivatore

Simuliamo ora il segnale all'uscita di un circuito con integratore e derivatore in cascata. In ingresso all'integratore (V_{in}) avremo una forma d'onda quadra alternata che si tramuterà in un'onda triangolare nell'uscita V_A e, data in pasto al derivatore, si ottiene nell'uscita V_B il segnale simulato di figura (8). Per quanto riguarda l'attenuazione e lo sfasamento del derivatore si hanno:

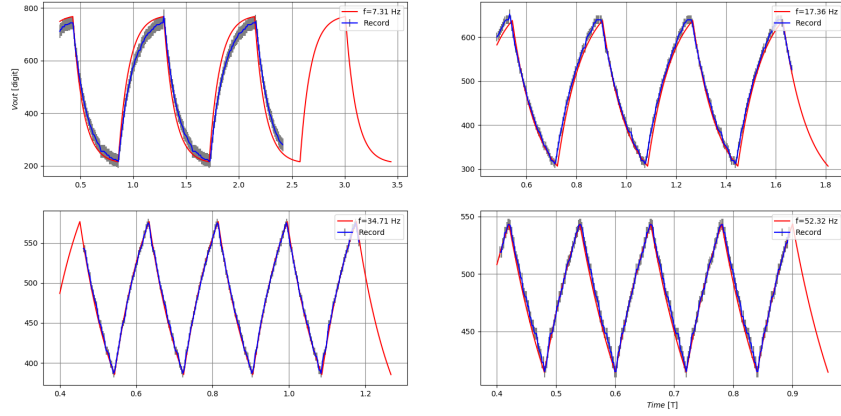


Figura 6: Sovrapposizione dell'onda a pinna di squalo simulata e del segnale acquisito con Arduino. La scala dei tempi è stata riadattata in unità di periodo anzi che in digit.

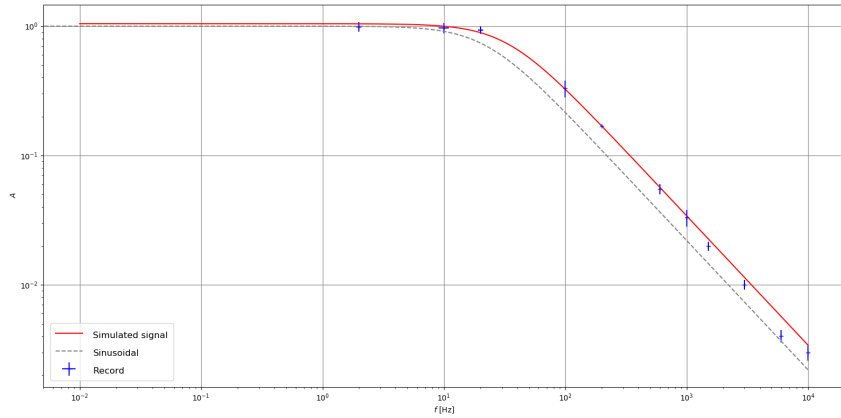


Figura 7: Confronto fra l'andamento dell'attenuazione secondo il modello dell'integratore, ricavato dalla prima delle equazioni (3) (linea tratteggiata) e quello ottenuto nella simulazione (linea rossa).

$$A_B(f_k) = \frac{1}{\sqrt[2]{1 + (f_T/f_k)^2}} \quad \Delta\varphi_B(f_k) = \arctan(f_T/f_k),$$

e dunque la funzione che genera il segnale è :

$$\omega(t) = \sum_{k=1}^n \frac{2}{k\pi} A_A(f_k) A_B(f_k) \sin(\omega_k t + \Delta\varphi_A(f_k) + \Delta\varphi_B(f_k)), \quad k \text{ dispari con } \omega_k = 2\pi f_k$$

Consideriamo le attenuazioni anche in questo caso , procedendo come fatto per l'integratore si è ottenuto l'andamento in figura (9).

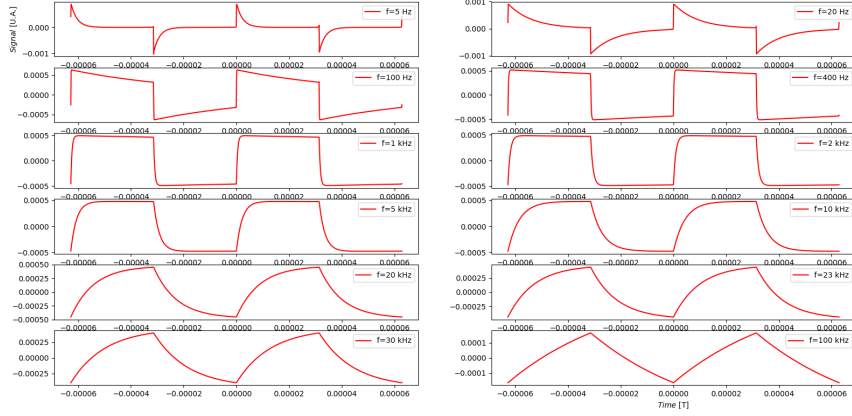


Figura 8: Segnale simulato in uscita dal derivatore per varie frequenze. Il circuito derivatore aveva una frequenza di taglio $f_{TB} = (23 \pm 5) kHz$ mentre il circuito integratore presentava una frequenza di taglio $f_{TA} = (22 \pm 5) Hz$. Osserviamo che per $f_T \gg f$ il circuito si comporta come un derivatore. Tra i 400 Hz e 1 kHz integratore e derivatore vanno a regime. Aumentando la frequenza, andando oltre a quella di taglio la forma d'onda in uscita dal derivatore rimane inalterata rispetto a quella in ingresso, dunque essa tende ad essere a pinna di squalo.

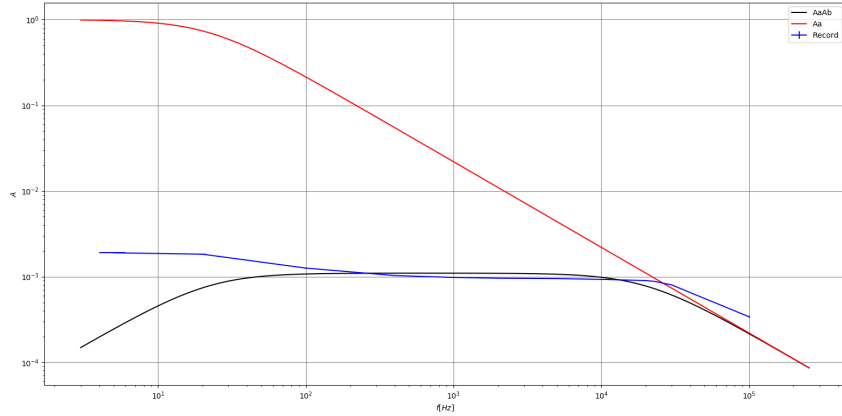


Figura 9: Il (tristissimo) grafico in figura presenta in rosso l'andamento del modello dell'integratore, in nero quello del prodotto fra le attenuazioni $A_A A_B$ e in blu i dati acquisiti in V_{in} e V_B . Non è stato possibile sta volta eseguire il fit sui dati raccolti.

4 Accoppiamento in AC

L'accoppiamento in AC nell'oscilloscopio risulta molto utile quando cerchiamo di visualizzare andamenti variabili nel tempo. Questa modalità dell'oscilloscopio può essere simulata supponendo la presenza di un condensatore in serie all'oscilloscopio. Insieme alla resistenza dell'oscilloscopio in entrata costituiscono un filtro passa alto, del quale possiamo simulare l'uscita, mostrata in figura (10).

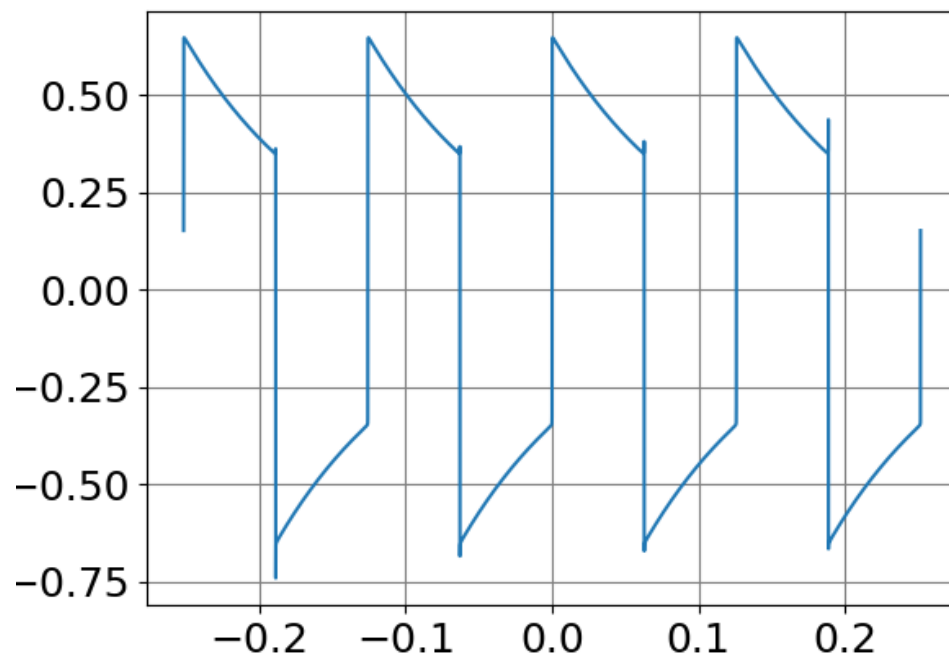


Figura 10: Segnale in uscita dal derivatore formato da oscilloscopio-consensatore con $f_T = 10Hz$ e $f = 50Hz$.