

Pendolo quadrifilare

Gruppo di lavoro A-1-3-2

Marco RICCI (m. 641084) m.ricci66@studenti.unipi.it
Martino BARBIERI (m. 640522) m.barbieri20@studenti.unipi.it

UniPi - Laboratorio di Fisica 1 - a.a. 2021-2022
Docente: Prof. Luca Baldini

1 Introduzione

Si consideri un pendolo fisico non smorzato. L'equazione del moto è

$$\ddot{\theta} = -\omega^2 \sin \theta$$

che in prima approssimazione si può approssimare a

$$\ddot{\theta} = -\omega^2 \theta$$

che dà come soluzione oscillazioni isocrone attorno alla posizione di equilibrio

$$\theta(t) = \theta_0 \sin(\omega t + \phi_0)$$

Ciò vale però solo per piccole oscillazioni; per grandi moti, si dimostra che il periodo dipende dall'ampiezza del moto. I primi termini dell'espansione di Taylor del periodo in funzione dell'ampiezza sono¹

$$T(\theta_0) = \frac{2\pi}{\omega} \left(1 + \frac{1}{16}\theta_0^2 + \frac{11}{3072}\theta_0^4 + \frac{173}{737280}\theta_0^6 + \frac{22931}{1321205760}\theta_0^8 + \dots \right)$$

In quanto detto fin'ora abbiamo trascurato l'effetto dell'attrito. Se esso è piccolo, possiamo supporre che esso non modifichi la singola oscillazione ma agisca solo su molti periodi riducendo progressivamente l'ampiezza del moto. Possiamo supporre che l'energia dispersa durante un ciclo abbia:

- una componente del primo grado nella velocità, come quella causata dall'attrito dinamico dei fili che strisciano meccanicamente contro i supporti;
- una componente del secondo grado nella velocità, come quella causata dall'urto anelastico del filo contro il bordo del foro nel quale esso è fissato oppure quella causata dall'attrito viscoso dell'aria per flussi laminari²;

¹L'unica fonte che abbiamo trovato per i termini successivi al quarto è stata Wikipedia in Hindi: <https://hi.wikipedia.org/wiki/%E0%A4%B2%E0%A5%8B%E0%A4%B2%E0%A4%95>

²Esso è proporzionale allo spazio percorso per la velocità di percorrenza, quindi in prima approssimazione al quadrato della velocità.

- una componente di terzo grado nella velocità, come quella causata dall'attrito viscoso in regime vorticoso dell'aria o alcuni sviluppi di ordine successivo al primo non banale delle componenti già descritte.

Durante un ciclo, dunque

$$\Delta E \approx mv\Delta v \approx -\bar{\alpha}v - \bar{\beta}v^2 - \bar{\gamma}v^3 \quad \Rightarrow \quad \Delta v \approx -(\alpha + \beta v + \gamma v^2)$$

1.1 Obiettivo

Il fine di questa esperienza è di verificare sperimentalmente se vi è accordo tra i dati sperimentali e il modello ricavato analiticamente riguardo alla relazione tra ampiezza e periodo di un pendolo fisico. Vogliamo valutare anche come l'attrito agisca sull'ampiezza di oscillazione nel tempo. Quindi vogliamo comprendere in quali condizioni la coincidenza tra modello e realtà è maggiore ed infine ipotizzare le ragioni per cui vi sono delle discrepanze.

1.2 Materiale utilizzato

1. pendolo quadrifilare con bandierina;
2. metro a nastro;
3. calibro ventesimale;
4. Arduino con led, fotocellula e suite Plasduino.

2 Raccolta dati

In questa sezione descriveremo brevemente gli accorgimenti che abbiamo utilizzato per ridurre potenziali errori sistematici. Riporteremo poi gli esiti di qualche misura che abbiamo preso.

Sul sito <https://unilab.mbarbieri.it> rendiamo disponibile tutto il materiale raccolto ed elaborato.

2.1 Accorgimenti

2.1.1 Disposizione della fotocellula

Cerchiamo di fare in modo che la congiungente led-fotocellula sia perpendicolare alla bandierina e che la fotocellula sia centrata con la bandierina quando il pendolo è all'equilibrio.

2.1.2 Avvio del moto

Teniamo il pendolo in posizione; successivamente lo lasciamo libero di muoversi cercando di farlo partire con velocità iniziale nulla. Cerchiamo di fare in modo che il moto del pendolo sia solo lungo la direzione voluta con grandi oscillazioni rispetto alla posizione di equilibrio (angolo iniziale di circa 45°).

2.1.3 Metodo di raccolta dati

L'apparato di misura della posizione del pendolo è formato nel seguente modo:

- al pendolo è collegata una bandierina metallica;
- il led è oscurato dalla bandierina quando il pendolo è prossimo alla posizione di equilibrio;
- la scheda Arduino misura per quanto tempo la fotocellula non riceve la luce del led e con quale frequenza succede questo; con semplici operazioni aritmetiche calcola il periodo di oscillazione del pendolo e il tempo di oscuramento del led;
- la scheda Arduino invia tutti i dati raccolti al PC, il quale tramite formule note calcola (sapendo la larghezza della bandierina e altri parametri fisici del pendolo) la velocità del pendolo alla posizione di equilibrio e l'ampiezza di oscillazione.

2.2 Misure

- Distanza tra punto di ancoraggio del pendolo e baricentro del pendolo: $l_g = (111.0 \pm 0.1)$ cm
- Distanza (verticale) tra punto di ancoraggio del pendolo e led: $l_f = (114.0 \pm 0.1)$ cm
- Larghezza della bandierina: $s = (1.985 \pm 0.005)$ cm

3 Analisi dati

3.1 Metodi di analisi

Plasduino raccoglie i dati in file `.txt`; essi vengono quindi elaborati con uno script in Python che rendiamo disponibile su GitLab. Siccome l'incertezza sulle ascisse non è in generale trascurabile rispetto a quella sulle ordinate (ipotesi richiesta dal metodo dei minimi quadrati), utilizziamo una regressione ODR.

Per studiare l'evoluzione temporale della velocità in funzione della velocità usiamo come modello quello che risolve l'equazione differenziale

$$\dot{v} = -(\alpha + \beta v + \gamma v^2)$$

Per studiare la relazione tra ampiezza e periodo usiamo un modello polinomiale:

$$T = T_0 (1 + a\theta_0^2 + b\theta_0^4 + \dots)$$

3.2 Incertezze

Come incertezza sulle distanze misurate, adottiamo la sensibilità strumentale. Stimiamo l'incertezza sui tempi misurati con 1×10^{-4} s, siccome Arduino impiega questo intervallo di tempo per effettuare una lettura analogica da un pin; in realtà Arduino non misura direttamente le misure passate al programma, ma siccome l'incertezza relativa sui tempi è molto più piccola di tutte le altre incertezze relative, consideriamo le misure date come indipendenti e ignoriamo il problema.

3.3 Codice Python utilizzato

Descriviamo brevemente il programma in Python che esegue l'analisi dei dati. Prima cosa importiamo le librerie necessarie (**numpy** per gestire array di dati, **pyplot** per creare i grafici, **scipy.odr** per eseguire la regressione).

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.odr import *
```

Inizializziamo alcune parametri per disegnare i grafici e importiamo le misure.

```
mar_size = 2
bar_l     = 1

t, T, tt = np.loadtxt('data.txt', comments='#', delimiter='\t', unpack=True)
s = 1.985
lg = 111.
lf = 114.
v = s / tt * ( lg / lf )
th0 = np.arccos(1-v**2/(2*981*lg))
```

Calcoliamo le incertezze delle nostre misure.

```
sigma_t = 1e-4*np.ones_like(t)
sigma_th0 = (
    ( 4*lg/(4*lf**2*981*tt**2-lg*s**2)*0.005**2/12 )+
    # dipendenza da s
    ( s**2/(lg*(4*lf**2*981*tt**2-lg*s**2))*0.1**2/12 )+
    # dipendenza da lg
    ( 4*lg*s**2/(lf**2*(4*lf**2*981*tt**2-lg*s**2))*0.1**2/12 )+
    # dipendenza da lf
    ( 4*lg*s**2/(tt**2*(4*lf**2*981*tt**2-lg*s**2))*1e-4**2/12 )
    # dipendenza da tt
) **0.5
sigma_v = v*(
    (0.005/s/12)**2+ # dipendenza da s
    (0.1/lg/12)**2+ # dipendenza da lg
    (0.1/lf/12)**2+ # dipendenza da lf
    (1e-4/tt/12)**2 # dipendenza da tt
) **0.5
```

Eseguiamo la prima regressione ($v(t)$): definiamo la funzione del nostro modello (calcolata per integrazione numerica con Metodo di Eulero con passo minore però del passo delle misure) e utilizziamo la libreria *scipy.odr*.

```
def vfoo(Par, t):
    v0, k, l, m = Par
    dt = min( t[1:]-t[:-1] )/9
    temp = t[0]
    i = 1
    vout = np.zeros_like(t)
    v = v0
    vout[0] = v
    while temp <= t[-1]:
        pv = v
        temp += dt
        v -= (m+l*v+k*v**2)*dt
        if temp > t[i]:
            vout[i] = (pv*(temp-t[i])+v*(t[i]+dt-temp))/dt
```



```
popt = tout.beta
```

Stampiamo a video i valori di best-fit con lo scarto dal valore atteso. Creiamo i grafici.

```
print("\np1\t\ta...")
print(popt)
print(tout.sd_beta)
print()

print("T")
print(" esp val \t fit val \t std dev \t difference \t difference % ")
print(
    "{:10.8f}\t".format(2*np.pi*(lg/981.)**0.5)+
    "{:10.8f}\t".format(popt[0])+
    "{:10f}\t".format(tout.sd_beta[0])+
    "{:10.8f}\t".format(2*np.pi*(lg/981.)**0.5-popt[0])+
    "{:10.8f}\t".format(100*(2*np.pi*(lg/981.)**0.5-popt[0])\
    /(2*np.pi*(lg/981.)**0.5))
)
for i in range(1,len(popt)):
    print("Coefficiente di grado {:1.0f}".format(2.*i))
    print(" esp val \t fit val \t std dev \t difference \t difference % ")
    print(
        "{:10.8f}\t".format(especCoeff[i-1])+
        "{:10.8f}\t".format(popt[i])+
        "{:10.8f}\t".format(tout.sd_beta[i])+
        "{:10.8f}\t".format(especCoeff[i-1]-popt[i])+
        "{:10.8f}\t".format(100*(especCoeff[i-1]-popt[i])/especCoeff[i-1])
    )

fig = plt.figure()
fig.add_axes((0.17, 0.280, 0.82, 0.67))
plt.errorbar(th0,T,xerr=sigma_th0,yerr=sigma_t,fmt='o',
             markersize=mar_size,linewidth=bar_l)
plt.plot(th0,Tee(popt,th0))
plt.ylabel(r'$T(\theta_0)$ [ s ]')
plt.gca().get_xaxis().set_visible(False)
plt.grid(which='both', ls='dashed', color='gray')
fig.add_axes((0.17, 0.17, 0.82, 0.125))
plt.errorbar(th0,T-Tee(popt,th0),xerr=sigma_th0,yerr=sigma_t,
             fmt='o',markersize=mar_size,linewidth=bar_l)
plt.grid(which='both', ls='dashed', color='gray')
plt.ylabel(r'Residui $\Delta T$ [ s ]')
plt.xlabel(r'Angolo iniziale $\theta_0$ [ rad ]')
plt.savefig("Ttheta.png")

plt.show()
```

3.4 Risultati dell'analisi

Le equazioni orarie seguono le equazioni previste, come mostrano i residui casualmente distribuiti intorno allo 0.

Riporto i parametri di best-fit e i grafici ottenuti.

v_0	α	β	γ
$(238.0 \pm 0.2) \text{ cm s}^{-1}$	$(30.6 \pm 2.6) 10^{-3} \text{ cm s}^{-2}$	$(92.7 \pm 3.4) 10^{-5} \text{ s}^{-1}$	$(99.8 \pm 1.1) 10^{-7} \text{ cm}^{-1}$

Parametri di best-fit per il primo fit. Il modello è un problema di Cauchy con $v(0) = v_0$ e $\dot{v} = -(\alpha + \beta v + \gamma v^2)$

	Best-fit	Valore aspettato	Differenza percentuale
T_0	2.116	2.114 ± 0.002	-0.1%
a	0.06124	0.06250	2%
b	0.003613	0.003581	-0.9%

Parametri di best-fit per il secondo fit confrontati con i valori aspettati. Il modello è polinomiale: $T = T_0(1 + a\theta_0^2 + b\theta_0^4)$.

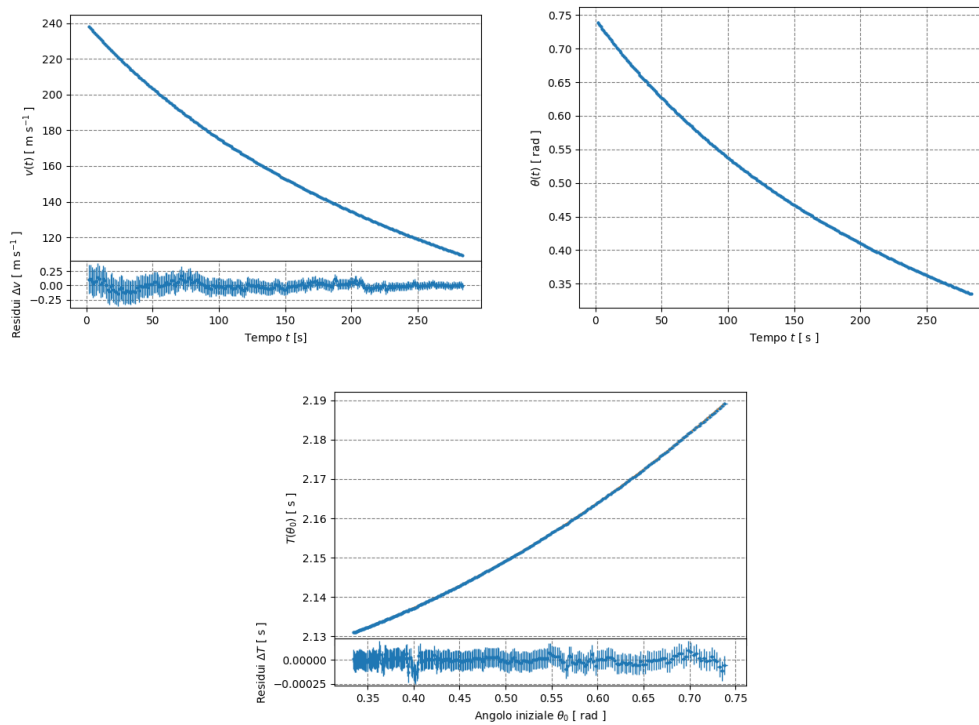


Figura 1: I primi due grafici sono riferiti all'andamento temporale rispettivamente della velocità al punto di minima altezza e dell'ampiezza delle oscillazioni del pendolo. L'ultimo grafico è riferito alla relazione tra ampiezza e periodo delle oscillazioni.

4 Conclusioni

Per i due fit, i residui dispersi casualmente attorno allo zero indicano che i modelli adottati sono adatti.

v_0 indica che il pendolo è stato fatto partire da un angolo di $(44.2 \pm 0.2)^\circ$, in accordo con quanto avvenuto sperimentalmente.

Per il fit delle velocità in funzione del tempo, considerare il differenziale della velocità un polinomio nella velocità di grado inferiore al secondo non porta a buon termine il fit; considerare un polinomio di grado superiore al secondo non migliora notevolmente il risultato; interpretiamo questo fatto nel seguente modo: nel modello utilizzato ogni termine ha il senso fisico dato sopra. Considerare meno parametri non ci permetterebbe di adottare un modello consono, considerarne di più ci farebbe cadere in overfitting.

Per la relazione tra periodo e ampiezza, come T_0 si ottiene un valore compatibile con quello calcolato tramite l'usuale formula $T = 2\pi\sqrt{\frac{l}{g}}$. Con il fit si riescono a trovare i primi due coefficienti correttivi per grandi oscillazioni con precisione entro il 2%. Aggiungendo al modello altri termini non si trovano i corretti coefficienti per i termini successivi e talvolta si peggiora la stima pure per i primi due coefficienti.