analisi

December 18, 2024

1 Analisi dati per la Velocità della Luce

Questo file contiene l'analisi dati dell'esperimento svolto in laboratorio seguendo l'esperimento di Friazou/Focault

1.1 Import

1.1.1 Import delle librerie

[1]:
```
from typing import Tuple

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scienceplots  # noqa
from colorama import Fore, Style
from scipy import stats

plt.style.use(["science", "ieee"])
```

1.1.2 Import dei dati

[2]:
```
# dati preliminari # * copiati da excel
D = 13.38833333  # m
a = 0.4826667  # m
f2 = 0.252  # m

D_err = 0.02  # m
a_err = 0.003  # m
f2_err = 0  # m

# dati delle misure
f_name = "Dati_Grezzi.xlsx"
set_names = ("CW", "CCW", "CWCCW", "CCWCW")

data_dict = dict()
for _set in set_names:
```

```
    data_dict[_set] = pd.read_excel(f_name, sheet_name=f"Exp_{_set}",␣
 ↪header=None)
    data_dict[_set] = data_dict[_set][(data_dict[_set].T != 0).any()].T.
 ↪to_numpy()
```

[3]:
```
# remove unused variable
del f_name, set_names
```

1.2 Formule statistiche

[4]:
```
def weighted_avg_and_std(values: np.ndarray, errors: np.ndarray) ->␣
 ↪Tuple[float]:
    # Return the weighted average and standard deviation.
    weights = 1 / np.square(errors)
    average = sum(values * weights) / sum(weights)
    sigma = 1 / np.sqrt(sum(weights))
    return average, sigma
```

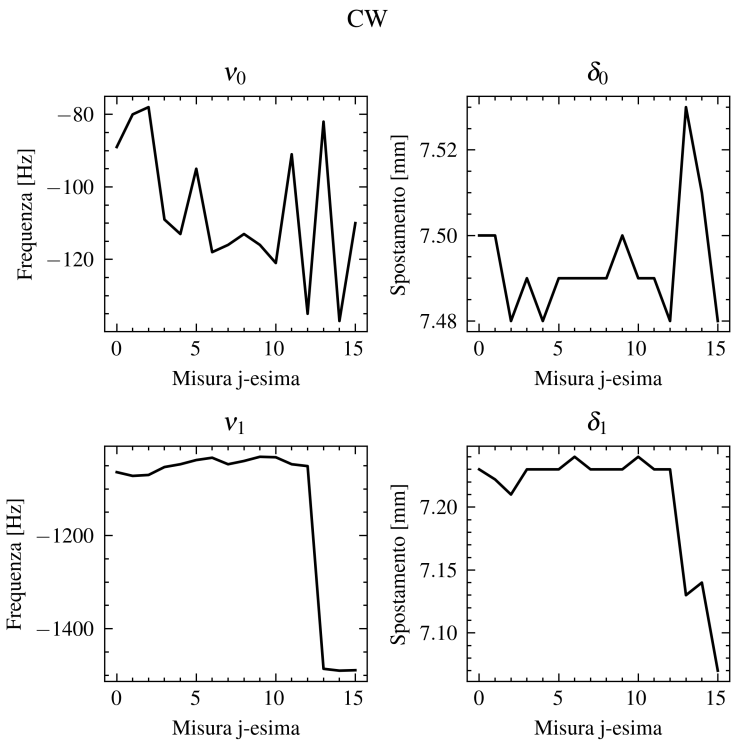
1.3 Inserire un grafico delle misure per controllare che non ci siano valori anomali

potrebbe avere senso fare uno scatter plot con i deltaomega e i delta delta

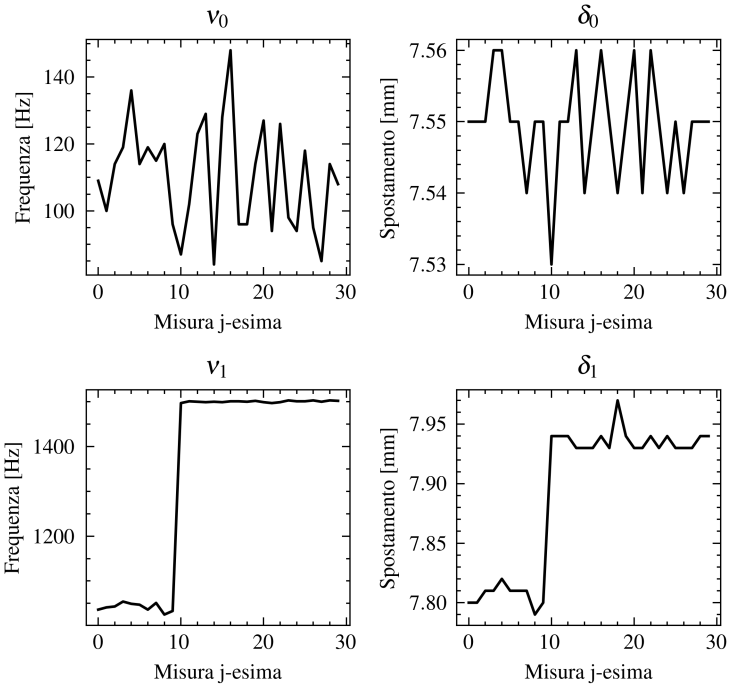
[5]:
```
plot_titles = [r"$\nu_0$", r"$\delta_0$", r"$\nu_1$", r"$\delta_1$"]
y_labels = ["Frequenza [Hz]", "Spostamento [mm]", "Frequenza [Hz]",␣
 ↪"Spostamento [mm]"]

for set_name, set_data in data_dict.items():
    fig, axs = plt.subplots(2, 2, figsize=(4, 4))
    fig.suptitle(set_name)
    for i in range(4):
        row, col = divmod(i, 2)
        axs[row, col].plot(set_data[i, :])
        axs[row, col].set_title(plot_titles[i])
        axs[row, col].set_xlabel("Misura j-esima")
        axs[row, col].set_ylabel(y_labels[i])

    fig.tight_layout()  # rect=[0, 0, 1, 0.95]
    fig.show()

fig = plt.subplots()
for set_name, set_data in data_dict.items():
    plt.scatter(
        set_data[3, :] - set_data[1, :], set_data[2, :] - set_data[0, :],␣
 ↪label=set_name
    )
plt.title("All data collected")
```

```
plt.xlabel(r"$\Delta\delta [mm]$")
plt.ylabel(r"$\Delta\nu [Hz]$")
plt.legend(loc="best")
plt.savefig("Images/AllData.svg")
plt.show()
```

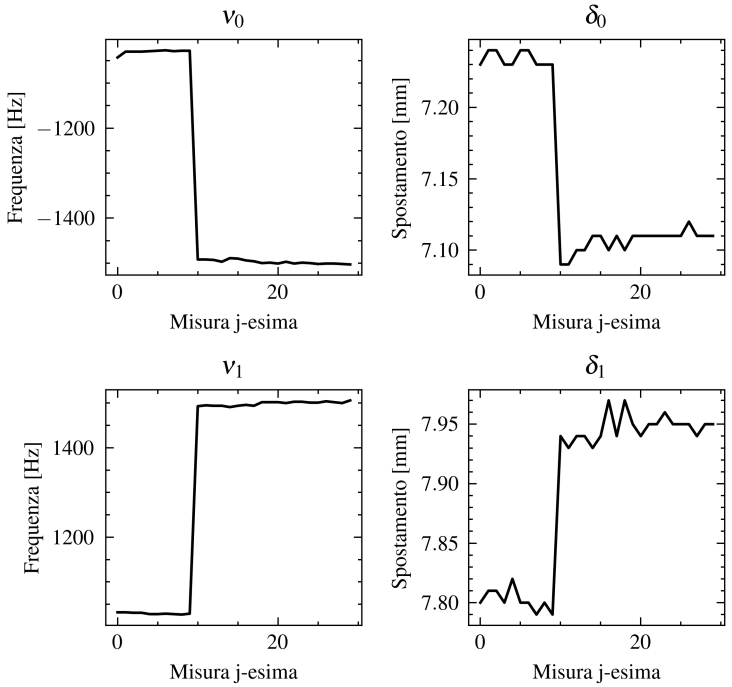
```
/var/folders/h4/lp363j8s0xd_71n6qf896t0r0000gn/T/ipykernel_82322/3825615484.py:1
5: UserWarning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  fig.show()
```

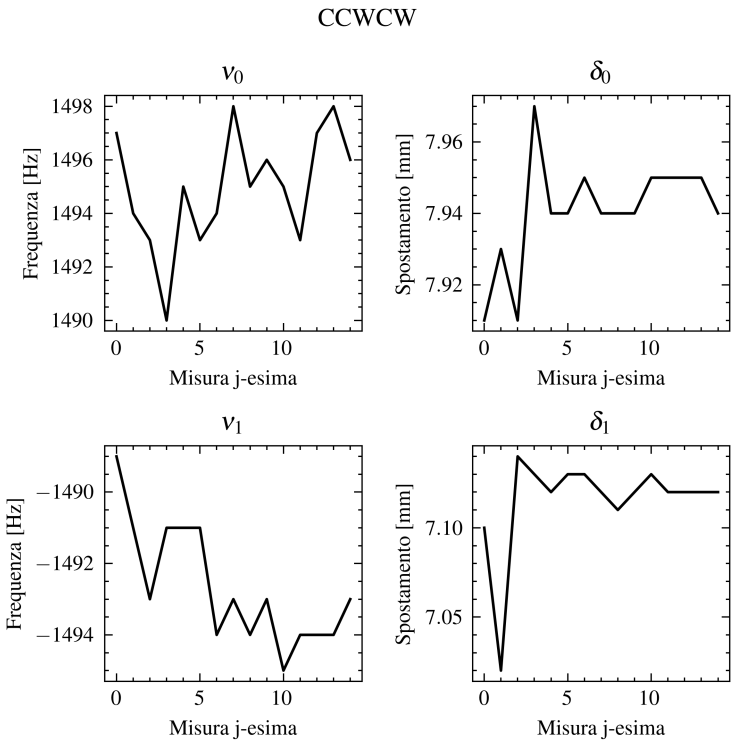


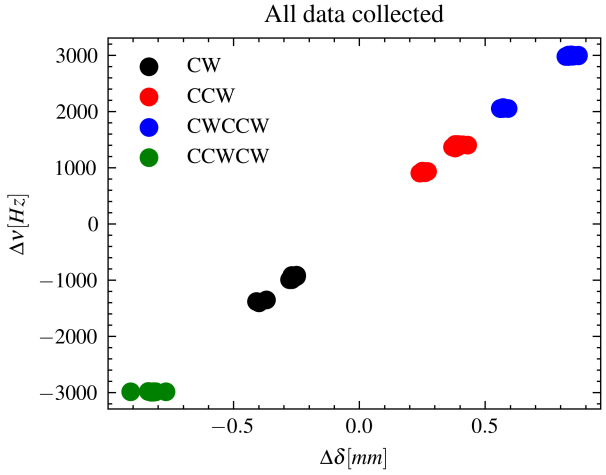
CCW


ν_0
δ_0
Frequenza [Hz]
Spostamento [mm]
Misura j-esima
ν_1
δ_1

CWCCW


ν_0
δ_0
Frequenza [Hz]
Spostamento [mm]
Misura j-esima
ν_1
δ_1





1.4 Analisi Statistica

`d<x>` è la differenza di `x`, che può essere: `v` per la frequenza, `w` per la velocità angolare, `d` per il δ (lo spostamento misurato col microscopio).

[6]:
```
c_means, c_errs = list(), list()
```

```
for set_name, set_data in data_dict.items():
    dv = set_data[2, :] - set_data[0, :]
    dd = (set_data[3, :] - set_data[1, :]) / 1000  # convert to meters
    dw = dv * 2 * np.pi

    # questo sarebbe da spostare su dove grafichiamo i dati se non printiamo␣
↪direttamente tutto prima
    print(Style.BRIGHT + Fore.GREEN + f" ~~~~~ {set_name}" + Style.RESET_ALL)
    print("differenze di " + Style.BRIGHT + "frequenza" + Style.NORMAL + ": ",␣
↪dv)
    print(
        "differenze di " + Style.BRIGHT + "velocità angolare" + Style.NORMAL +␣
↪": ", dw
    )
    print("differenze di " + Style.BRIGHT + "spostamento" + Style.NORMAL + ":␣
↪", dd)

    c = 4 * f2 * D**2 / (D + a - f2) * dw / dd
    print(
        Style.BRIGHT + "velocità della luce" + Style.NORMAL + " nell'aria␣
↪misurate: ", c
    )

    mean_c = c.mean()

    c_err = 0
    """
    # * errore sistematico
    alpha = 1 / abs((D + a - f2) ** 2 * dd)
    e_sist = alpha * np.sqrt(
        np.square(-4 * f2 * D**2 * dw * a_err)  # errore dovuto ad a
        + np.square(4 * D**2 * dw * (D + a) * f2_err)  # errore dovuto a f2
        + np.square(4 * f2 * D * dw * (D + 2 * a - 2 * f2) * D_err)  # errore␣
↪dovuto a D
    )
    mean_c, c_err = weighted_avg_and_std(c, e_sist)
    """
    # * errore casuale
    std_c = c.std(ddof=1) / np.sqrt(c.size)

    # * real error
    c_err = np.sqrt(np.square(c_err) + np.square(std_c))

    c_means.append(mean_c)
    c_errs.append(c_err)
    print(
```

```
        "Miglior valore di c: "
        + Fore.CYAN
        + Style.BRIGHT
        + f"{mean_c:.6g} ±{c_err:.5g}"
    )
```

```
 ~~~~~ CW
differenze di frequenza:  [ -975.  -992.  -992.  -944.  -934.  -943.
-915.  -931.  -927.  -915.
  -911.  -956.  -916. -1404. -1353. -1379.]
differenze di velocità angolare:  [-6126.1056745  -6232.91982472
-6232.91982472 -5931.32692998
 -5868.49507691 -5925.04374467 -5749.11455607 -5849.64552098
 -5824.51277976 -5749.11455607 -5723.98181484 -6006.72515366
 -5755.39774138 -8821.59217128 -8501.14972061 -8664.5125386 ]
differenze di spostamento:  [-0.00027  -0.000278 -0.00027  -0.00026
-0.00025  -0.00026  -0.00025
 -0.00026  -0.00026  -0.00027  -0.00025  -0.00026  -0.00025  -0.0004
 -0.00037  -0.00041 ]
velocità della luce nell'aria misurate:  [3.01015642e+08 2.97450764e+08
3.06264120e+08 3.02654307e+08
 3.11426152e+08 3.02333699e+08 3.05090931e+08 2.98486398e+08
 2.97203965e+08 2.82491603e+08 3.03757200e+08 3.06501608e+08
 3.05424364e+08 2.92587204e+08 3.04820580e+08 2.80368115e+08]
Miglior valore di c: 2.99867e+08 ±2.1228e+06

 ~~~~~ CCW
differenze di frequenza:  [ 927.  941.  929.  935.  913.  933.  917.
936.  905.  937. 1410. 1399.
 1377. 1370. 1416. 1371. 1353. 1405. 1404. 1388. 1372. 1403. 1373. 1405.
 1407. 1383. 1408. 1415. 1389. 1394.]
differenze di velocità angolare:  [5824.51277976 5912.47737406
5837.07915037 5874.77826221 5736.54818545
 5862.2118916  5761.68092668 5881.06144752 5686.282703   5887.34463283
 8859.29128312 8790.17624474 8651.94616799 8607.96387084 8896.99039497
 8614.24705614 8501.14972061 8827.87535659 8821.59217128 8721.06120637
 8620.53024145 8815.30898597 8626.81342676 8827.87535659 8840.4417272
 8689.64527983 8846.72491251 8890.70720966 8727.34439167 8758.76031821]
differenze di spostamento:  [0.00025 0.00025 0.00026 0.00025 0.00026
0.00026 0.00026 0.00027 0.00024
 0.00025 0.00041 0.00039 0.00039 0.00037 0.00039 0.00038 0.00038 0.00038
 0.00043 0.00039 0.00037 0.00039 0.00038 0.00038 0.0004  0.00038 0.00039
 0.00038 0.00039 0.00039]
velocità della luce nell'aria misurate:  [3.09092123e+08 3.13760181e+08
2.97845182e+08 3.11759585e+08
 2.92715448e+08 2.99127615e+08 2.93997881e+08 2.88975016e+08
 3.14329796e+08 3.12426450e+08 2.86670807e+08 2.99020746e+08
 2.94318489e+08 3.08650550e+08 3.02654307e+08 3.00747531e+08
```

```
 2.96798986e+08 3.08205894e+08 2.72174143e+08 2.96669617e+08
 3.09101135e+08 2.99875701e+08 3.01186258e+08 3.08205894e+08
 2.93212391e+08 3.03379895e+08 3.00944396e+08 3.10399531e+08
 2.96883356e+08 2.97952051e+08]
Miglior valore di c: 3.00703e+08 ±1.6814e+06

 ~~~~~ CWCCW
differenze di frequenza:  [2075. 2062. 2061. 2061. 2057. 2056. 2056.
2057. 2055. 2057. 2985. 2987.
 2987. 2991. 2980. 2984. 2990. 2990. 3002. 3001. 3003. 2997. 3004. 3002.
 3001. 3003. 3005. 3003. 3002. 3009.]
differenze di velocità angolare:  [13037.6095124  12955.9281034
12949.6449181  12949.6449181
 12924.51217687 12918.22899156 12918.22899156 12924.51217687
 12911.94580625 12924.51217687 18755.30814193 18767.87451255
 18767.87451255 18793.00725377 18723.8922154  18749.02495662
 18786.72406847 18786.72406847 18862.12229215 18855.83910685
 18868.40547746 18830.70636562 18874.68866277 18862.12229215
 18855.83910685 18868.40547746 18880.97184807 18868.40547746
 18862.12229215 18906.1045893 ]
differenze di spostamento:  [0.00057 0.00057 0.00057 0.00057 0.00059
0.00056 0.00056 0.00056 0.00057
 0.00056 0.00085 0.00084 0.00084 0.00084 0.00082 0.00083 0.00087 0.00083
 0.00087 0.00084 0.00083 0.00084 0.00084 0.00085 0.00084 0.00084 0.00083
 0.00083 0.00084 0.00084]
velocità della luce nell'aria misurate:  [3.03453016e+08 3.01551864e+08
3.01405622e+08 3.01405622e+08
 2.90623342e+08 3.06043596e+08 3.06043596e+08 3.06192450e+08
 3.00528167e+08 3.06192450e+08 2.92734307e+08 2.96417711e+08
 2.96417711e+08 2.96814655e+08 3.02935817e+08 2.99687714e+08
 2.86483853e+08 3.00290303e+08 2.87633621e+08 2.97807014e+08
 3.01595913e+08 2.97410070e+08 2.98104722e+08 2.94401470e+08
 2.97807014e+08 2.98005486e+08 3.01796776e+08 3.01595913e+08
 2.97906250e+08 2.98600901e+08]
Miglior valore di c: 2.9893e+08 ±9.1273e+05

 ~~~~~ CCWCW
differenze di frequenza:  [-2986. -2985. -2986. -2981. -2986. -2984.
-2988. -2991. -2989. -2989.
 -2990. -2987. -2991. -2992. -2989.]
differenze di velocità angolare:  [-18761.59132724 -18755.30814193
-18761.59132724 -18730.1754007
 -18761.59132724 -18749.02495662 -18774.15769785 -18793.00725377
 -18780.44088316 -18780.44088316 -18786.72406847 -18767.87451255
 -18793.00725377 -18799.29043908 -18780.44088316]
differenze di spostamento:  [-0.00081 -0.00091 -0.00077 -0.00084
-0.00082 -0.00081 -0.00082 -0.00082
 -0.00083 -0.00082 -0.00082 -0.00083 -0.00083 -0.00083 -0.00082]
```

```
velocità della luce nell'aria misurate:  [3.07293233e+08 2.73433144e+08
3.23256518e+08 2.95822295e+08
 3.03545755e+08 3.07087411e+08 3.03749067e+08 3.04054036e+08
 3.00189872e+08 3.03850724e+08 3.03952380e+08 2.99989009e+08
 3.00390735e+08 3.00491166e+08 3.03850724e+08]
Miglior valore di c: 3.02064e+08 ±2.575e+06
```

[7]:
```
# ! si potrebbe fare un error bar plot per vedere se le varie medie siano␣
↪compatibili tra loro (dovrebbero)
```

[8]:
```
c_mean, c_std = weighted_avg_and_std(c_means, c_errs)

"""
e_sist = (
    c_mean
    * (D + a - f2)
    / D
    * np.sqrt(np.square(D * a_err) + np.square((D + 2 * a - 2 * f2) * D_err))
)
"""

alpha = mean_c / (D * (D + a - f2))  # (4 * f2 * D * dw) / abs((D + a - f2) **␣
↪2 * dd)
e_sist = alpha * np.sqrt(
    np.square(-D * a_err)  # errore dovuto ad a
    + np.square((D + 2 * a - 2 * f2) * D_err)  # errore dovuto a D
)

c_std = np.sqrt(np.square(c_std) + np.square(e_sist))

print(
    Fore.GREEN
    + Style.BRIGHT
    + "Valore finale di c: "
    + Fore.RED
    + f"{c_mean:.6g} ±{c_std:.3g}m/s"
)

na = 1.000283
print(
    Fore.GREEN
    + "Valore finale di c nel vuoto: "
    + Fore.RED
    + f"{c_mean * na:.6g} ±{c_std * na:.3g}m/s"
)
```

```
Valore finale di c: 2.99608e+08 ±8.57e+05m/s
```

```
Valore finale di c nel vuoto: 2.99693e+08 ±8.57e+05m/s
```

[9]:
```python
# norm-test
z_value = abs(c_mean - 299_792_458) / c_std
p_student = stats.norm.cdf(z_value)
p_value = 2 * (1 - p_student)

print(f"P-value di c nell'aria compatibile con c vero: {p_value:.2%}")
```

```
P-value di c nell'aria compatibile con c vero: 82.99%
```

[ ]:
```python

```