

Versuch 211 Gekoppeltes Pendel

Leonardo Karl Reiter

March 21, 2024

Contents

1 Zielsetzung	2
2 Grundlagen	2
2.1 Einzelnes ungekoppeltes Pendel	2
2.2 Gekoppeltes Pendel	2
2.2.1 Symmetrische Schwingung	4
2.2.2 Antisymmetrische Schwingung	4
2.2.3 Schwebungsschwingung	5
2.3 Kopplungsgrad	6
3 Durchführung	6
3.1 Skizze	6
3.2 Nullabgleich der Ruhelage	6
3.3 Schwingungsfrequenzen der einzelnen Pendel	6
3.4 Kopplungsgrade bei symmetrisch und antisymmetrischer Schwingung	6
3.5 Schwebungsschwingung	6
3.6 Elektrischer gekoppelter Schwingkreis	6
4 Messprotokoll	7
5 Auswertung	8
5.1 Numerische Frequenzanalyse	8
5.1.1 Python Module	8
5.1.2 Funktionen	8
5.2 Teil 1: Eigenfrequenzen ω_1, ω_2 ohne Kopplung	9
5.3 Teil 2 Anti- und Symmetrische Schwingung mit Kopplung	12
5.3.1 Symmetrische Schwingungen	12
5.3.2 Berechnung der Eigenfrequenz ω_1^{sys}	14
5.3.3 Antisymmetrische Schwingung	15
5.3.4 Berechnung der Eigenfrequenz ω_2^{anti}	18
5.4 Teil 3: Schwebungsschwingung	19
5.4.1 Berechnung der Eigenfrequenz ω_I Schwebungsfrequenz ω_{II}	19
5.5 Vergleich von ω_I und ω_{II} bei verschiedenen Kopplungen	24
5.5.1 Zusammenfassung der Ergebnisse	26
5.6 Teil 4: Kopplungsgrad κ	26
5.6.1 Verhältnisse der Kopplungsgrade	27
5.7 Teil 5: Gekoppelter elektrischer Schwingkreis	28
5.8 Qualitative Beobachtungen	29
6 Diskussion	30
6.1 Hohe σ -Abweichungen bei ω_{II}	30
6.2 Fazit	30

1 Zielsetzung

Ziel des Versuchs ist es, anhand zwei mechanisch gekoppelter Pendel die elementaren Eigenschaften gekoppelter Oszillatoren kennenzulernen. Dabei soll der Einfluss der gewählten Anfangsbedingungen auf Eigenfrequenz und Kopplungsstärke sowie der Energieübertragung innerhalb des Systems untersucht werden. Außerdem soll die Analogie zu anderen schwingungsfähigen Systemen, mithilfe eines elektrischen Schwingkreises qualitativ beschrieben werden.

2 Grundlagen

Der Schwingungszustand eines gekoppelten Systems wird mithilfe von Differentialgleichungen beschrieben. Dabei bietet es sich an ein geeignetes Koordinatensystem zu verwenden. Die Position des einfachen Pendels mit festem Abstand zur Pendelachse werden alle über den Winkel φ charakterisiert. Wir arbeiten also mit Polarkoordinaten.

2.1 Einzelnes ungekoppeltes Pendel

Betrachtet man zunächst ein einfaches Pendel, der Länge L mit Masse m , Trägheitsmoment J , auf der Erde mit Erdbeschleunigung g , so für den Winkel φ :

$$J\ddot{\varphi} = -mgL \cdot \varphi \quad (1)$$

$$= -D\varphi \quad (2)$$

mit dem sogenannten Direktionsmoment $D = mgL$.

Die Lösung beschreibt eine harmonische Schwingung mit der Kreisfrequenz:

$$\omega = \sqrt{\frac{D}{J}} = \sqrt{\frac{g}{L}} \quad (3)$$

2.2 Gekoppeltes Pendel

Werden nun zwei solche Pendel durch eine Feder mit Direktionsmoment $D' = D_F \cdot l^2$ (wobei D_F der Federkonstante und l der Abstand der Aufhängung zur Pendelachse entspricht), miteinander verbunden, so wirken zusätzliche Drehmomente in Abhängigkeit der Auslenkwinkel φ_1 , und φ_2 :

$$\textbf{Pendel 1 : } M_1 = D'(\varphi_2 - \varphi_1) \quad (4)$$

$$\textbf{Pendel 2 : } M_2 = D'(\varphi_1 - \varphi_2) \quad (5)$$

Diese zusätzlichen Drehmomente M_1, M_2 müssen nun in den Bewegungsgleichungen der beiden Pendel hinzu addiert werden. Daraus ergeben sich die Differentialgleichungen:

$$J\ddot{\varphi}_1 = -D\varphi_1 + D'(\varphi_2 - \varphi_1) \quad (6)$$

$$J\ddot{\varphi}_2 = -D\varphi_2 + D'(\varphi_1 - \varphi_2) \quad (7)$$

Die Bewegung von Pendel 1 hängt also auch von dem Auslenkwinkel φ_2 ab und vice versa. Somit sind die Bewegungsgleichungen der Beiden Pendel miteinander **gekoppelt**.

Um eine analytische Lösung zu finden müssen sich die Gleichungen allerdings wieder entkoppeln lassen. Dazu wählen wir Folgenden Ansatz:

$$u = \varphi_1 + \varphi_2 \quad (8)$$

$$v = \varphi_1 - \varphi_2 \quad (9)$$

Dieser ist vergleichbar mit dem einer Schwerpunktbewegung u und einer Relativbewegung v und liefert die Bewegungsgleichungen:

$$J\ddot{u} + Du = 0 \quad (10)$$

$$J\ddot{v} + (D + 2D')v = 0 \quad (11)$$

Die Lösungen sind auch hier wieder harmonische Schwingungen mit den Kreisfrequenzen ω_1, ω_2 .

$$u = u(t) = A_1 \cos(\omega_1 t) + B_1 \sin(\omega_1 t) \quad \text{mit } \omega_1 = \sqrt{\frac{D}{J}} \quad (12)$$

$$v = v(t) = A_2 \cos(\omega_2 t) + B_2 \sin(\omega_2 t) \quad \text{mit } \omega_2 = \sqrt{\frac{D + 2D'}{J}} \quad (13)$$

Substituieren wir zurück zum Laborsystem mit:

$$\varphi_1 = \frac{u + v}{2} \quad (14)$$

$$\varphi_2 = \frac{u - v}{2} \quad (15)$$

$$(16)$$

So folgen die Allgemeinen Lösungen für die Auslenkwinkel φ_1, φ_2 :

$$\varphi_1(t) = \frac{1}{2} (A_1 \cos(\omega_1 t) + B_1 \sin(\omega_1 t) + A_2 \cos(\omega_2 t) + B_2 \sin(\omega_2 t)) \quad (17)$$

$$\varphi_2(t) = \frac{1}{2} (A_1 \cos(\omega_1 t) + B_1 \sin(\omega_1 t) - A_2 \cos(\omega_2 t) - B_2 \sin(\omega_2 t)) \quad (18)$$

Diese Gleichung beschreiben vollständig die Bewegung der beiden Pendel, welche sich sehr kompliziert und auch teils chaotisch verhalten können und sich lediglich durch die Anfangsbedingungen $\varphi_i(t = 0), \dot{\varphi}_i(t = 0)$ ($i = 1, 2$) unterscheiden. Wir wollen uns allerdings folgende 3 Spezialfälle ansehen:

2.2.1 Symmetrische Schwingung

Die beiden Pendel werden in dem gleichen Winkel ausgelenkt $\varphi_1 = \varphi_2$ und bei $t = 0$ losgelassen:

$$\text{Anfangsbedingung: } \varphi_1(t=0) = \varphi_2(t=0) = \varphi_0 \quad (19)$$

$$\dot{\varphi}_1(t=0) = \dot{\varphi}_2(t=0) = 0 \quad (20)$$

\Rightarrow Gleichung 17, 18:

$$A_1 = 2\varphi_0 \quad B_1 = 0 \quad (21)$$

$$A_2 = 0 \quad B_2 = 0 \quad (22)$$

\Rightarrow Gleichung 17, 18:

$$\varphi_1(t) = 2\varphi_0 \cdot \cos(\omega_1 t) \quad (23)$$

$$\varphi_2(t) = \varphi_1(t) \quad (24)$$

Die beiden Pendel schwingen harmonisch und phasengleich mit der Frequenz ω_1 . Die Kopplungsfeder erfährt keine Auslenkung da der Pendelabstand zu jedem Zeitpunkt gleich bleibt. Somit sind die Bewegungen vollständig äquivalent zueinander.

2.2.2 Antisymmetrische Schwingung

Die beiden Pendel werden im gleichen Winkel gegeneinander ausgelenkt $\varphi_1 = -\varphi_2$ und bei $t = 0$ losgelassen.

$$\text{Anfangsbedingung: } \varphi_1(t=0) = -\varphi_2(t=0) = \varphi_0 \quad (25)$$

$$\dot{\varphi}_1(t=0) = \dot{\varphi}_2(t=0) = 0 \quad (26)$$

\Rightarrow Gleichung 17, 18:

$$A_1 = 0 \quad B_1 = 0 \quad (27)$$

$$A_2 = 2\varphi_0 \quad B_2 = 0 \quad (28)$$

\Rightarrow Gleichung 17, 18:

$$\varphi_1(t) = 2\varphi_0 \cdot \cos(\omega_2 t) \quad (29)$$

$$\varphi_2(t) = -\varphi_1(t) \quad (30)$$

Die beiden Pendel schwingen harmonisch gegenphasig zueinander mit der Frequenz ω_2 , welche sowohl vom Direktionsmoment D , wie vom Kopplungsmoment D' abhängt.

2.2.3 Schwebungsschwingung

Sind die Auslenkwinkel φ_1 , φ_2 unterschiedlich zueinander, so kommt es zur Schwebung. Die Energie im System "schwingt" zwischen den beiden Pendeln hin und her:

$$\text{Anfangsbedingung: } \varphi_1(t=0) = 0, \varphi_2(t=0) = \varphi_0 \quad (31)$$

$$\dot{\varphi}_1(t=0) = \dot{\varphi}_2(t=0) = 0 \quad (32)$$

⇒ Gleichung 17, 18:

$$A_1 = \varphi_0 \quad B_1 = 0 \quad (33)$$

$$A_2 = -\varphi_0 \quad B_2 = 0 \quad (34)$$

⇒ Gleichung 17, 18:

$$\varphi_1(t) = \varphi_0 \sin\left(\frac{\omega_2 - \omega_1}{2}t\right) \sin\left(\frac{\omega_2 + \omega_1}{2}t\right) \quad (35)$$

$$\varphi_2(t) = \varphi_0 \cos\left(\frac{\omega_2 - \omega_1}{2}t\right) \cos\left(\frac{\omega_2 + \omega_1}{2}t\right) \quad (36)$$

Das System wird durch zwei Frequenzen beschrieben. Zum einen die Frequenz mit der jedes einzelne Pendel schwingt $\omega_I = \frac{1}{2}(\omega_1 + \omega_2)$.

Zum anderen die Schwebungsfrequenz, mit der die Energie oszilliert $\omega_{II} = \frac{1}{2}(\omega_2 - \omega_1)$

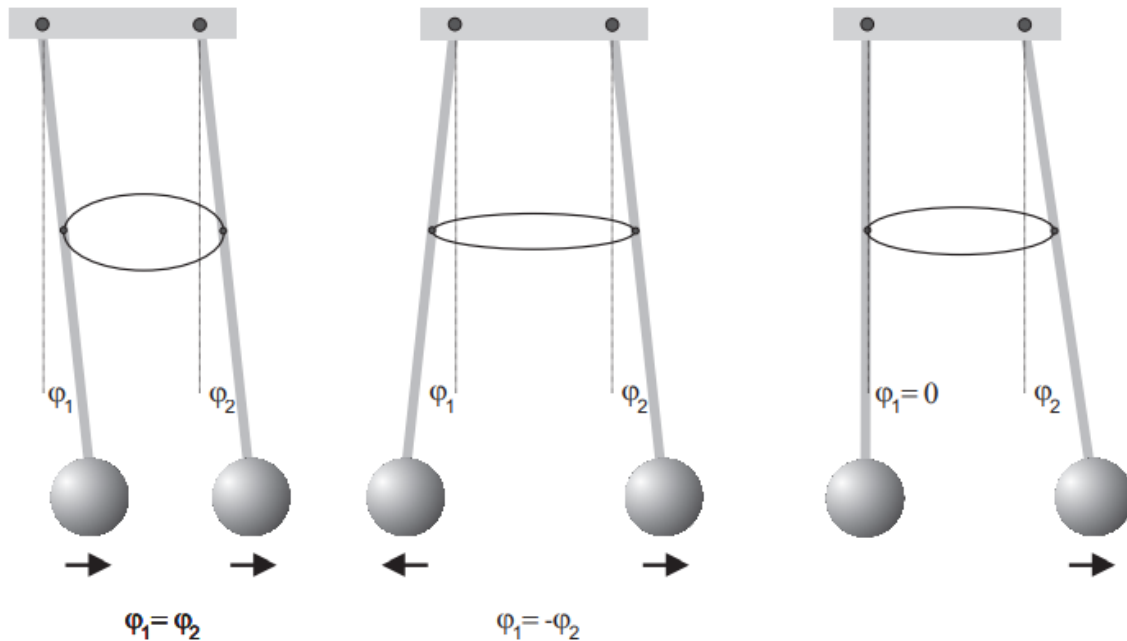


Figure 1: Schwingungsformen des gekoppelten Pendel für unterschiedliche Randbedingungen. Von links nach rechts: symmetrische Schwingung, antisymmetrische Schwingung, Schwebung.

2.3 Kopplungsgrad

Der Kopplungsgrad quantifiziert die Stärke der Kopplung und ergibt sich aus den beiden Direktionsmomenten D , D' :

$$\kappa = \frac{D'}{D + D'} \quad (37)$$

⇒ Gleichung 12, 13

$$\kappa = \frac{\omega_2^2 - \omega_1^2}{\omega_1^2 + \omega_2^2} = \frac{T_2^2 - T_1^2}{T_1^2 + T_2^2} \quad (38)$$

Über die Schwingzeiten der Normalschwingungen kann also auch der Kopplungsgrad indirekt gemessen werden.

3 Durchführung

3.1 Skizze

Es soll zunächst eine Skizze angefertigt werden, welche die Dimensionen der Anordnung veranschaulicht.

3.2 Nullabgleich der Ruhelage

In der Auswertung werden die relevanten Frequenzen numerisch berechnet, dies erfordert eine präzise Aufnahme der Ruhelage. Diese wird deshalb vor dem Beginn der eigentlichen Messungen erneut kalibriert.

3.3 Schwingungsfrequenzen der einzelnen Pendel

Es sollen nun die Eigenfrequenzen der beiden ungekoppelten Pendel gemessen werden sie werden aus den Oszilloskopbildern über die Fouriertransformation numerisch berechnet.

3.4 Kopplungsgrade bei symmetrisch und antisymmetrischer Schwingung

Die beiden Pendel sollen möglichst genau in dem gleichen Winkel, zum gleichen Zeitpunkt, gleich- und gegenphasig ausgelenkt werden. So werden die in den Grundlagen 2.2.1 und 2.2.2, besprochenen Fälle der symmetrischen und antisymmetrischen Schwingungen hervorgerufen, dessen Oszilloskopbilder wieder im Frequenzspektrum untersucht werden sollen.

3.5 Schwebungsschwingung

Nun soll eines der Pendel festgehalten werden, während das andere wieder um einen kleinen Winkel φ_0 ausgelenkt wird. Beide sollen gleichzeitig losgelassen werden, um so eine saubere Schwebungsschwingung hervorzurufen.

3.6 Elektrischer gekoppelter Schwingkreis

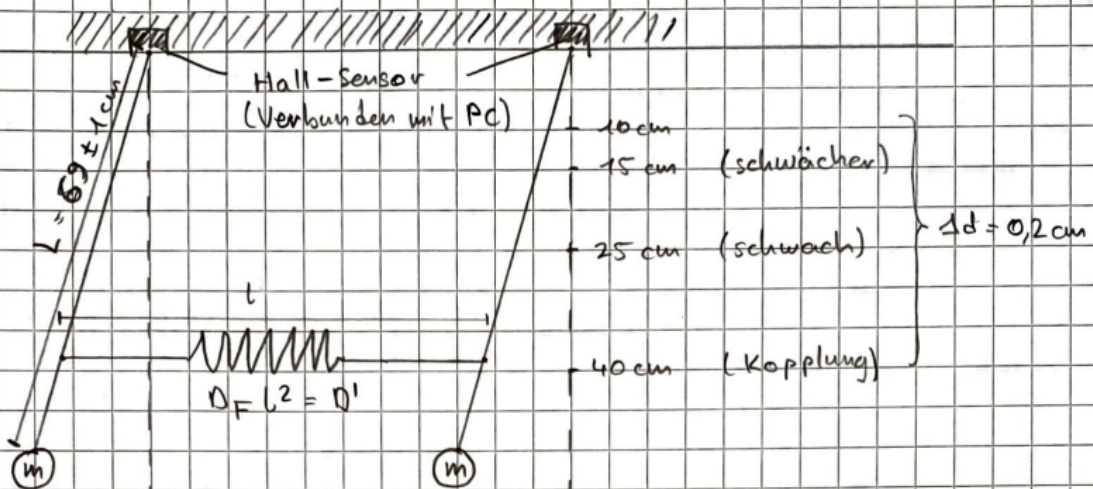
Der zweite Versuchsaufbau soll nun qualitativ beschrieben werden, indem die parallelen zum mechanischen Analogon gezogen werden und die Äquivalenz der, in den Grundlagen besprochenen Prinzipien, für beide Schwingungssysteme nachzuvollzogen wird.

4 Messprotokoll

Messprotokoll zu Versuch 211

Avon Schaeber
Leonardo Reller

1) Versuchsaufbau:



2) Das Protokoll für die Frequenzanalyse ist den Plots zu entnehmen, die dem Protokoll beigelegt sein werden.

3) Die Effekte sind analog und die Kopplungskonstante konnte mit variierendem Abstand der Spulen beeinflusst werden. Die Dämpfung war etwas stärker.

Master
27.04.22

5 Auswertung

5.1 Numerische Frequenzanalyse

Die Frequenzspektren werden mittels der Fastfouriertransformation (fft) analysiert. Die Funktion ist im *numpy* Modul für Python bereits integriert und erfordert lediglich den datensatz und ein paar Parameter. Die Höhe und Breite der Peaks werden mittels einer einfachen Gaussfunktion gefittet.

5.1.1 Python Module

```
[1]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
from scipy.stats import norm
from decimal import Decimal
import io
from scipy.optimize import curve_fit
from scipy.stats import chi2
plt.rcParams.update({'font.size': 20})
plt.rcParams["figure.figsize"] = 16, 9
```

5.1.2 Funktionen

```
[19]: def sig(val1,err1,val2,err2):
    sig = np.abs(val2-val1)**2/np.abs(np.abs(err2)+np.abs(err1))**2
    return round(sig,3)

def format_e(n):
    a = '%e' % Decimal(n)
    return a.split('e')[0].rstrip('0').rstrip('.')+'e'+a.split('e')[1]

def form(val,err):
    a = "{value:4.3f} ± {error:4.3f}".format(value=val,error=err)
    return a

def format_plt(n):
    a = '%e' % Decimal(n)
    b = r'${'+a.split('e')[0].rstrip('0').rstrip('.')+'}$'+r'${*10~{' + a.
    ↪split('e')[1]+'}$'
    return b

def comma_to_float(valstr):
    return float(valstr.decode('utf-8').replace(',','.'))

def open_data(n):
    filename = 'data/Messung'+str(n)+'.txt'
    X,Y1,Y2 = np.loadtxt(filename,skiprows=2,usecols=(0,1,2),
                        converters={0:comma_to_float,1:comma_to_float,
                                   2:comma_to_float},
                        unpack=True)
```

```

    return X,Y1,Y2

def gaussian(x, y, mu, sig):
    return norm.pdf(x, mu, sig)*y
def gaussian2(x, y1, mu1, sig1, y2, mu2, sig2):
    return norm.pdf(x, mu1,sig1)*y1+norm.pdf(x, mu2,sig2)*y2

def fastfourier(n):
    t, phi1, phi2 = open_data(n)

    timestep = 0.025

    #Fouriertransformation mit zeropadding
    spektrum_1 = np.fft.fft(np.concatenate((phi1, np.zeros(2*len(phi1)))))
    spektrum_2 = np.fft.fft(np.concatenate((phi2, np.zeros(2*len(phi2)))))

    #spektrum = np.fft.fft #Fouriertransformation
    freq_1 = np.fft.fftfreq(spektrum_1.size, timestep)
    freq_2 = np.fft.fftfreq(spektrum_2.size, timestep)

    n_1=spektrum_1.size    #Nur positive Werte
    n_2=spektrum_2.size

    n_halb_1 = np.ceil(n_1/2.0)
    n_halb_2 = np.ceil(n_2/2.0)
    spektrum_halb_1 = (2.0 / int(n_1)) * spektrum_1[0:int(n_halb_1)]
    spektrum_halb_2 = (2.0 / int(n_2)) * spektrum_2[0:int(n_halb_2)]
    freq_halb_1 = freq_1[0:int(n_halb_1)]
    freq_halb_2 = freq_2[0:int(n_halb_2)]
    amplitude_1 =np.abs(spektrum_halb_1)
    amplitude_2 =np.abs(spektrum_halb_2)
    return freq_halb_1, amplitude_1, freq_halb_2, amplitude_2

def fitgüte(x,y,y_err,popt,fit_funktion):
    chisquare=np.sum((fit_funktion(x,*popt)-y)**2/y_err**2)

    #Freiheitsgrade
    dof=dof = x.size-popt.size
    chisquare_red = chisquare/dof
    prob = round(1-chi2.cdf(chisquare,dof),2)*100 #Fitwahrscheinlichkeit

    print('chi_squared= ' + format_e(chisquare))
    print('chi_squared_red= ' + format_e(chisquare_red))
    print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: ' +
          str(prob) + ' %')
```

5.2 Teil 1: Eigenfrequenzen ω_1, ω_2 ohne Kopplung

```
[20]: def plot_schwingung(n,title):
        t, phi1, phi2 = open_data(n)
        F1, A1, F2, A2 = fastfourier(n)
```

```

popt1, pcov1 = curve_fit(gaussian,F1,A1)
perr1 = np.sqrt(np.diag(pcov1))

popt2, pcov2 = curve_fit(gaussian,F2,A2)
perr2 = np.sqrt(np.diag(pcov2))

fig, axs = plt.subplots(2, 2, constrained_layout=True)

axs[0,0].plot(t, phi1)
#axs[0].set_xlabel('time [sec]')
axs[0,0].set_ylabel('$Auslenkwinkel \\varphi_1$ [deg]')
axs[0,0].grid(ls=':')

axs[1,0].plot(t, phi2,c='red')
axs[1,0].set_xlabel('Zeit $t$ [sec]')
axs[1,0].set_ylabel('Auslenkwinkel $\\varphi_2$ [deg]')
axs[1,0].grid(ls=':')

lim = 1

axs[0,1].plot(F1, A1)
axs[0,1].plot(F1, gaussian(F1,popt1[0],popt1[1],popt1[2]),c='g',ls='--',
              label=r'$\mu = $'+format_plt(popt1[1])+' Hz\n'
              +r'$\sigma = $'+format_plt(popt1[2])+' Hz\n'
              +r'$FWHM = $'+format_plt(popt1[2]*2.355)+' Hz')
axs[0,1].set_xlim(0,lim)
axs[0,1].set_ylabel('Amplitude $A_1$ [dBm]')
axs[0,1].grid(ls=':')
axs[0,1].legend(loc='best', fontsize=14)

axs[1,1].plot(F2, A2,c='red')
axs[1,1].plot(F2, gaussian(F2,popt2[0],popt2[1],popt2[2]),c='g',ls='--',
              label=r'$\mu = $'+format_plt(popt2[1])+' Hz\n'
              +r'$\sigma = $'+format_plt(popt2[2])+' Hz\n'
              +r'$FWHM = $'+format_plt(popt2[2]*2.355)+' Hz')
axs[1,1].set_xlim(0,lim)
axs[1,1].set_xlabel('Frequenz $f$ [Hz]')
axs[1,1].set_ylabel('Amplitude $A_2$ [dBm]')
axs[1,1].grid(ls=':')
axs[1,1].legend(loc='best', fontsize=14)

fig.suptitle(title)
plt.show()

w1, dw1, w2, dw2 = popt1[1], popt1[2], popt2[1], popt2[2]

print('\nGaussfit 1:')
fitgüte(F1,A1,0.025,popt1,gaussian)
print('\nGaussfit 2:')
fitgüte(F2,A2,0.025,popt2,gaussian)

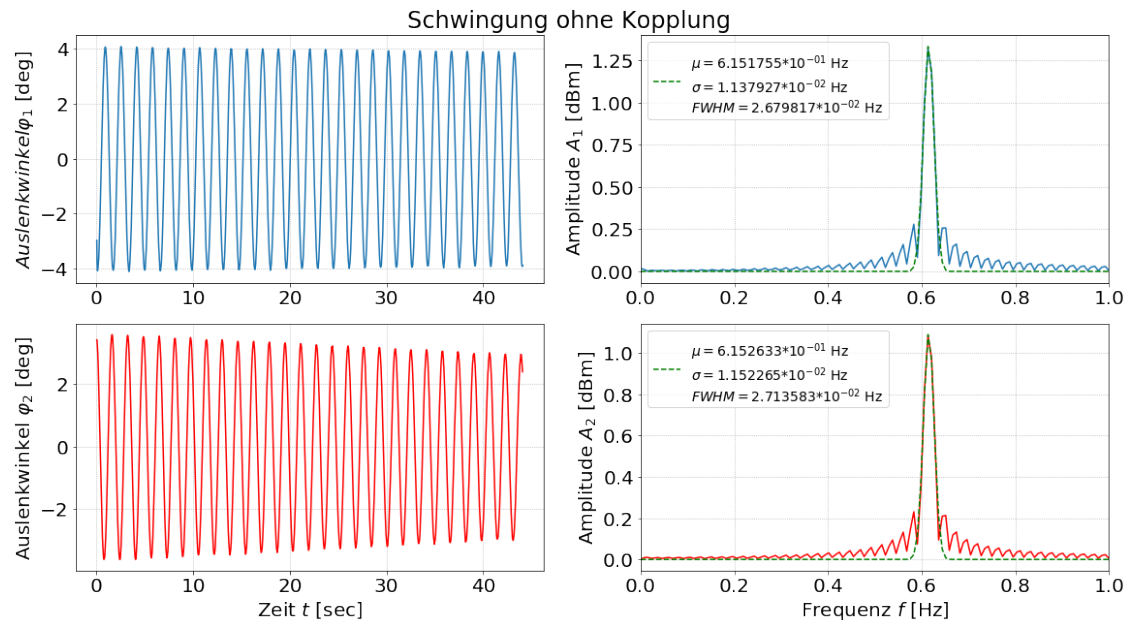
print('\nEigenfrequenzen:\n')
print('Pendel 1: w_1 =', form(w1,dw1), ' [Hz]')

```

```
print('Pendel 2: w_2 =', form(w2,dw2), ' [Hz]')
print('=> sig = ',sig(w1,dw1,w2,dw2), '\n')
```

```
return popt1[1], popt1[2], popt2[1], popt2[2]
```

```
E1,dE1,E2,dE2 = plot_schwingung(9,'Schwingung ohne Kopplung')
```



Gaussfit 1:

chi_squared= 7.851123e+02

chi_squared_red= 2.977294e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 5.283554e+02

chi_squared_red= 2.003623e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.615 \pm 0.011$ [Hz]

Pendel 2: $w_2 = 0.615 \pm 0.012$ [Hz]

=> sig = 0.0

$$\omega_1 = 0.615 \pm 0.027 \text{ [Hz]} \quad (39)$$

$$\omega_2 = 0.615 \pm 0.027 \text{ [Hz]} \quad (40)$$

$$\Rightarrow 0.000\sigma \quad (41)$$

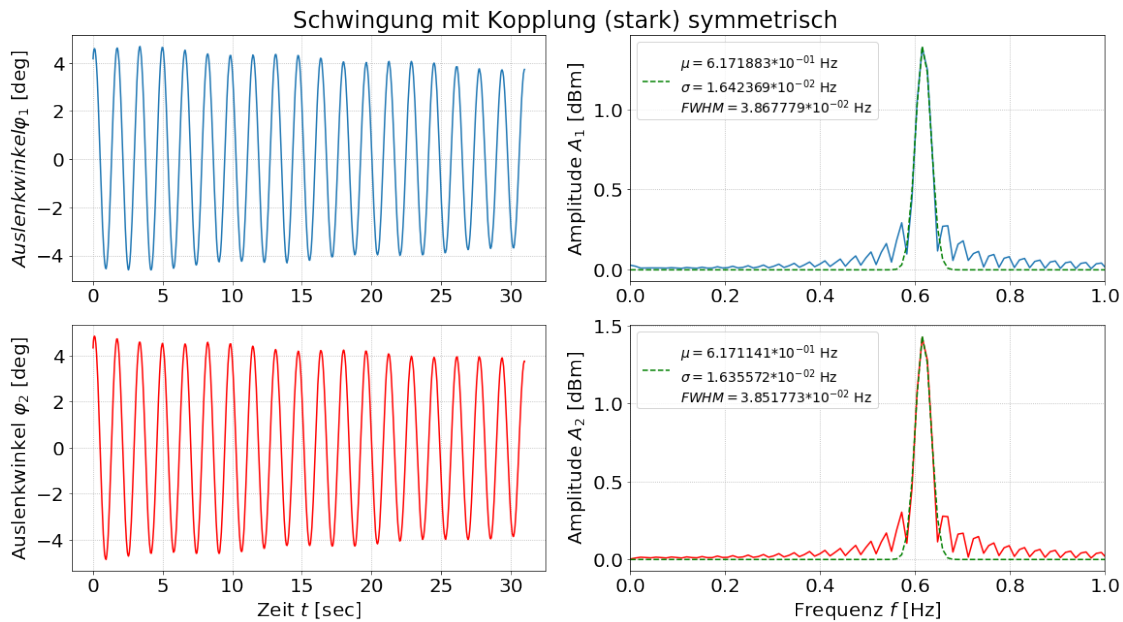
Die Eigenfrequenzen ω_1 , ω_2 sind somit eindeutig nicht voneinander verschieden. Die beiden Pendel sind

somit in sehr guter Näherung identisch.

5.3 Teil 2 Anti- und Symmetrische Schwingung mit Kopplung

5.3.1 Symmetrische Schwingungen

```
[21]: S1s,dS1s,S2s,dS2s=plot_schwingung(1,'Schwingung mit Kopplung (stark) symmetrisch')
      S1m,dS1m,S2m,dS2m=plot_schwingung(26,'Schwingung mit Kopplung (mittel) symmetrisch')
      S1w,dS1w,S2w,dS2w=plot_schwingung(37,'Schwingung mit Kopplung (schwach) symmetrisch')
```



Gaussfit 1:

chi_squared= 8.608355e+02

chi_squared_red= 4.650651e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 9.298682e+02

chi_squared_red= 5.023599e-01

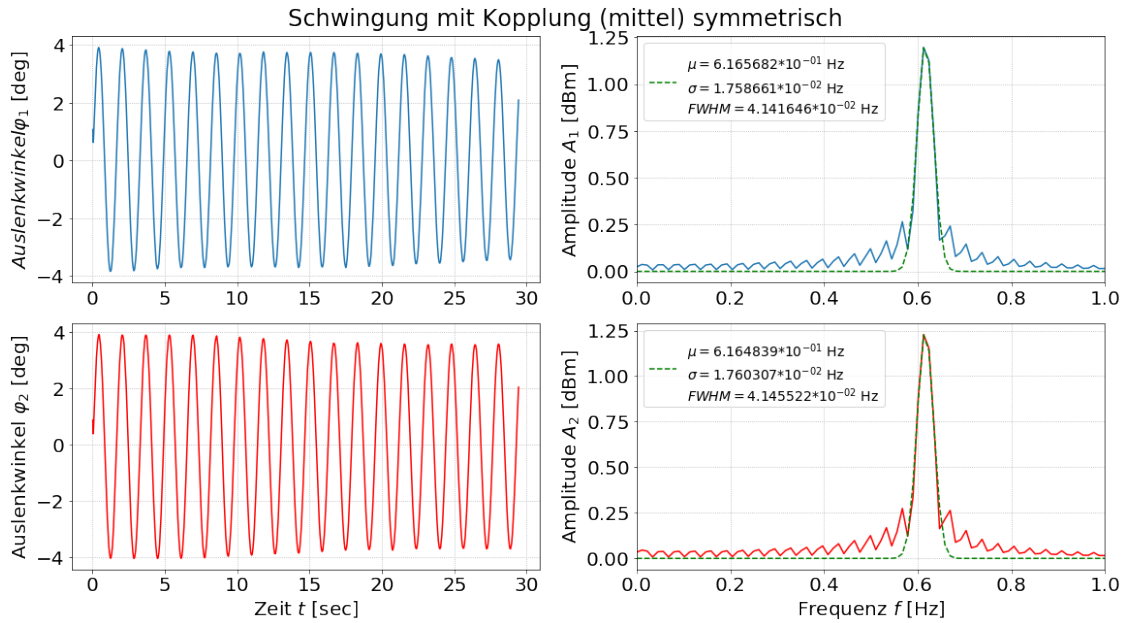
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.617 \pm 0.016$ [Hz]

Pendel 2: $w_2 = 0.617 \pm 0.016$ [Hz]

=> sig = 0.0



Gaussfit 1:

chi_squared= 6.247922e+02

chi_squared_red= 3.549956e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 6.933806e+02

chi_squared_red= 3.939663e-01

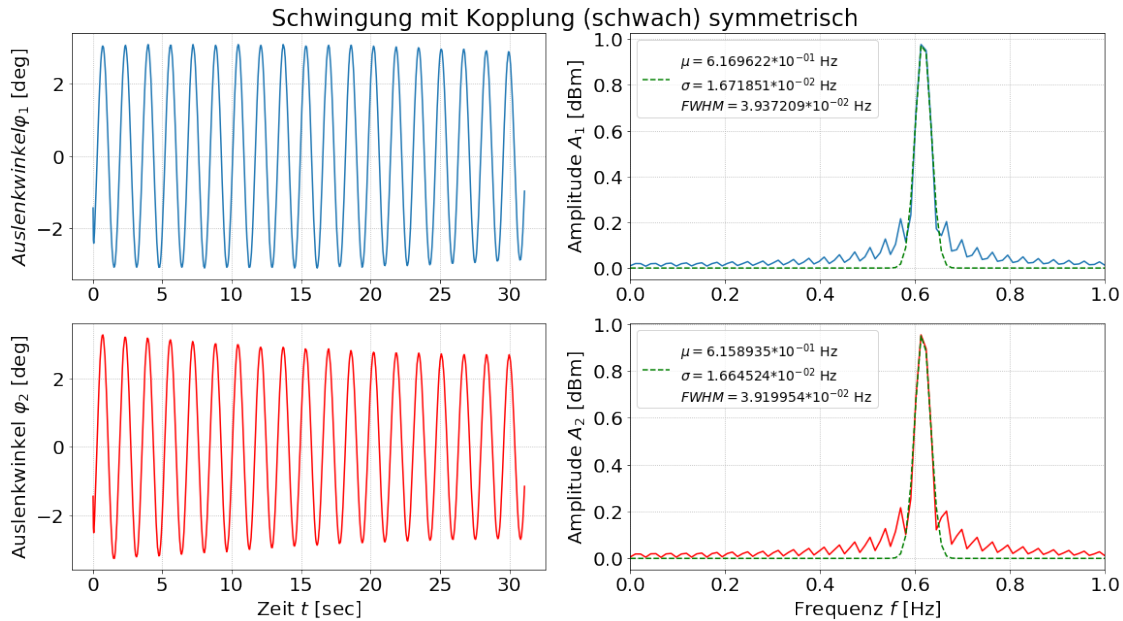
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.617 \pm 0.018$ [Hz]

Pendel 2: $w_2 = 0.616 \pm 0.018$ [Hz]

=> sig = 0.0



Gaussfit 1:

chi_squared= 4.0424e+02

chi_squared_red= 2.174503e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 4.297384e+02

chi_squared_red= 2.311664e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.617 \pm 0.017$ [Hz]

Pendel 2: $w_2 = 0.616 \pm 0.017$ [Hz]

=> sig = 0.001

Die Eigenfrequenzen der beiden Pendel sind bei weitem nicht signifikant voneinander verschieden. Das gekoppelte System schwingt mit **einer** Frequenz. Wir wollen nun sehen ob sich diese bei verschiedenen starken Kopplungen ändert.

5.3.2 Berechnung der Eigenfrequenz ω_1^{sys}

```
[5]: w_stark_sym, dw_stark_sym = np.mean([S1s,S2s]), dS1s+dS2s
     w_mittel_sym, dw_mittel_sym = np.mean([S1m,S2m]), dS1m+dS2m
     w_schwach_sym, dw_schwach_sym = np.mean([S1w,S2w]), dS1w+dS2w

     print('\nEigenfrequenzen symmetrische Schwingung:\n')
     print('Stark:   w_s =', form(w_stark_sym,dw_stark_sym), ' [Hz]')
     print('Mittel:  w_m =', form(w_mittel_sym,dw_mittel_sym), ' [Hz]')
```

```

print('Schwach: w_w = ', form(w_schwach_sym,dw_schwach_sym), ' [Hz]\n')
print('Abweichungen:\n')
print('=> sig_sm = ',sig(w_stark_sym,dw_stark_sym,w_mittel_sym,dw_mittel_sym))
print('=> sig_mw = ',sig(w_mittel_sym,dw_mittel_sym,w_schwach_sym,dw_schwach_sym))
print('=> sig_sw = ',sig(w_stark_sym,dw_stark_sym,w_schwach_sym,dw_schwach_sym))

```

Eigenfrequenzen symmetrische Schwingung:

Stark: w_s = 0.617 ± 0.033 [Hz]
Mittel: w_m = 0.617 ± 0.035 [Hz]
Schwach: w_w = 0.616 ± 0.033 [Hz]

Abweichungen:

=> sig_sm = 0.0
=> sig_mw = 0.0
=> sig_sw = 0.0

Die σ -Abweichungen sind wieder nicht signifikant. Das system Schwingt also **unabhängig** von der Kopplung mit **einer** Frequenz. Die Eigenfrequenz des symmetrischen Systems ist ω_1^{sym}

$$\omega_1^{\text{sym}} = \sqrt{\frac{D}{J}} \quad (42)$$

Sie ist von der Kopplung **unabhängig** und ergibt sich aus dem Mittelwert der gemessenen eigenfrequenzen der symmetrischen Schwingung.

```

[6]: w_sys = np.mean([w_stark_sym,w_mittel_sym,w_schwach_sym])
dw_sys = (max([dw_stark_sym,dw_mittel_sym,dw_schwach_sym])
          + np.std([w_stark_sym,w_mittel_sym,w_schwach_sym]))
print('\nEigenfrequenz der sym. Schwingung\n\nw_1_sys = ', form(w_sys,dw_sys))

```

Eigenfrequenz der sym. Schwingung

w_1_sys = 0.617 ± 0.036

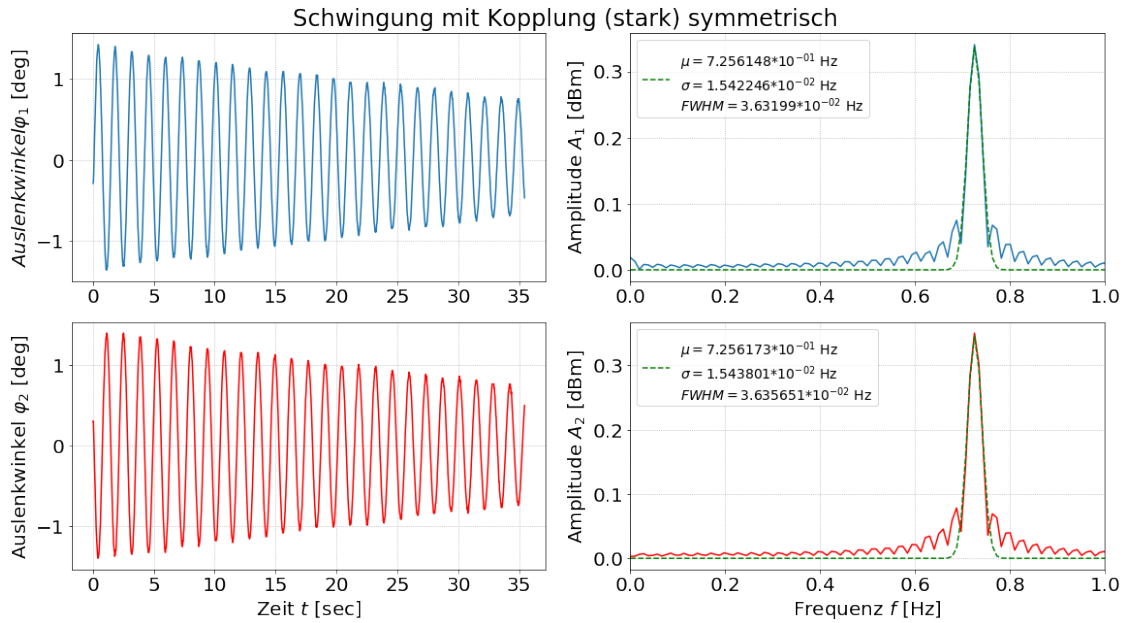
$$\omega_1^{\text{sym}} = 0.617 \pm 0.036 \text{ [Hz]} \quad (43)$$

5.3.3 Antisymmetrische Schwingung

```

[22]: A1s,dA1s,A2s,dA2s=plot_schwingung(2,'Schwingung mit Kopplung (stark) symmetrisch')
A1m,dA1m,A2m,dA2m=plot_schwingung(29,'Schwingung mit Kopplung (mittel) symmetrisch')
A1w,dA1w,A2w,dA2w=plot_schwingung(38,'Schwingung mit Kopplung (schwach) symmetrisch')

```

Gaussfit 1:

chi_squared= 5.591189e+01

chi_squared_red= 2.63611e-02

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 5.844937e+01

chi_squared_red= 2.755746e-02

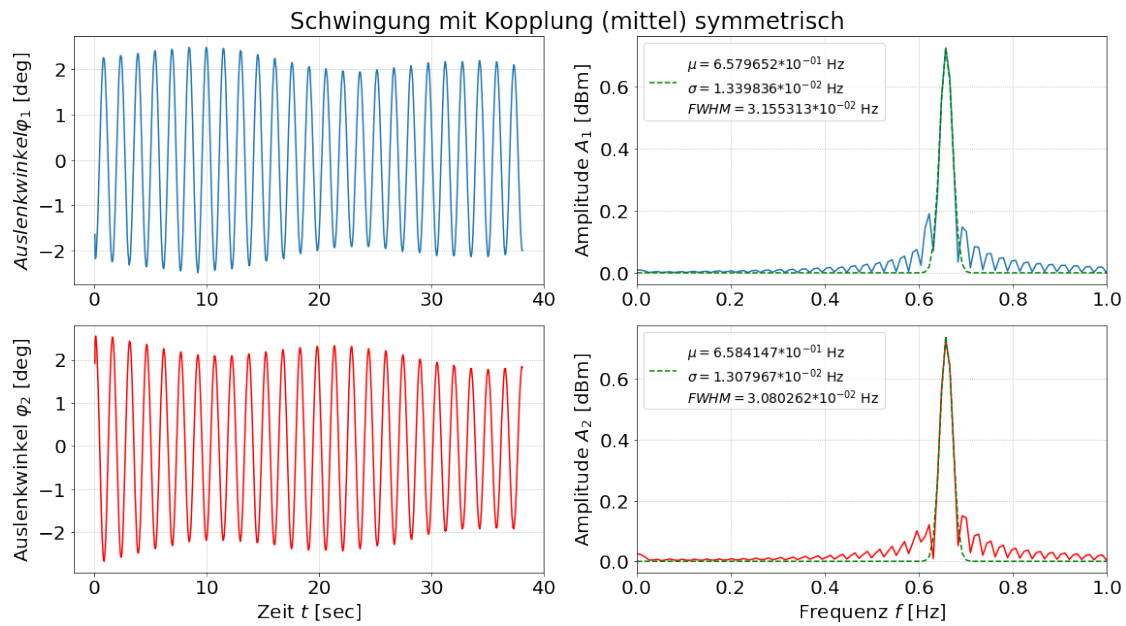
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.726 \pm 0.015$ [Hz]

Pendel 2: $w_2 = 0.726 \pm 0.015$ [Hz]

=> sig = 0.0



Gaussfit 1:

chi_squared= 2.635228e+02

chi_squared_red= 1.155802e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 2.514409e+02

chi_squared_red= 1.102811e-01

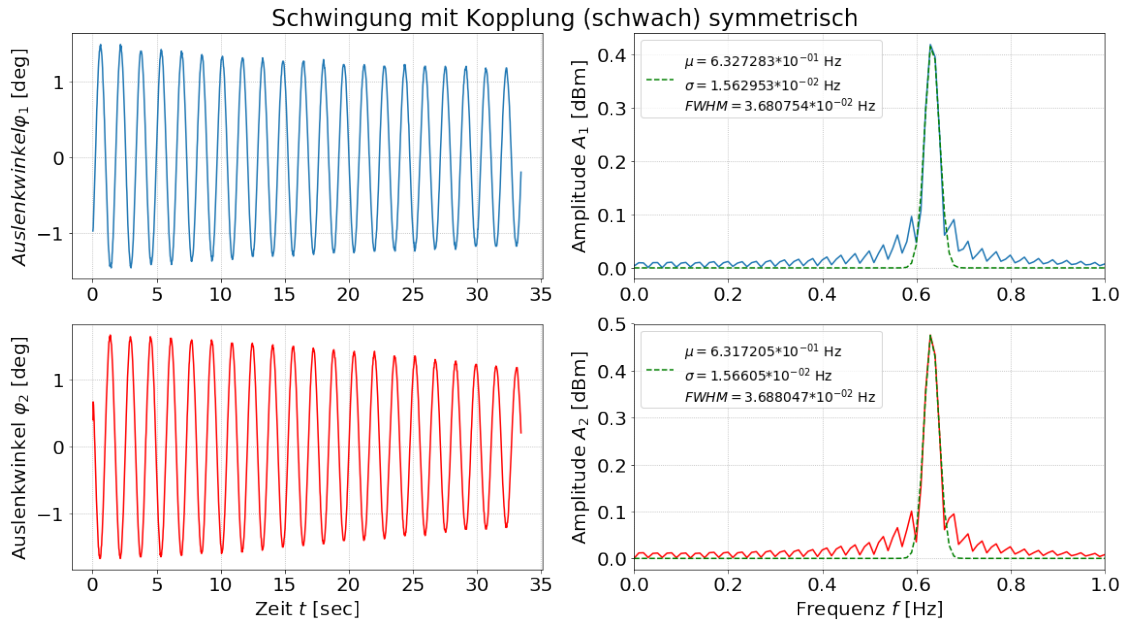
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.658 \pm 0.013$ [Hz]

Pendel 2: $w_2 = 0.658 \pm 0.013$ [Hz]

=> sig = 0.0



Gaussfit 1:

chi_squared= 8.452824e+01

chi_squared_red= 4.226412e-02

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 9.715834e+01

chi_squared_red= 4.857917e-02

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_1 = 0.633 \pm 0.016$ [Hz]

Pendel 2: $w_2 = 0.632 \pm 0.016$ [Hz]

=> sig = 0.001

Die Eigenfrequenzen der beiden Pendel unterscheiden sich in jedem Fall nicht voneinander. Das System schwingt mit **einer** Frequenz. Es ist aber ein deutlicher Abfall der Frequenz mit abnehmender Kopplung zu erkennen.

5.3.4 Berechnung der Eigenfrequenz ω_2^{anti}

```
[8]: w_stark_ant, dw_stark_ant = np.mean([A1s,A2s]), dA1s+dA2s
     w_mittel_ant, dw_mittel_ant = np.mean([A1m,A2m]), dA1m+dA2m
     w_schwach_ant, dw_schwach_ant = np.mean([A1w,A2w]), dA1w+dA2w

     print('\nEigenfrequenzen antisymmetrische Schwingung:\n')
     print('Stark:   w_s =', form(w_stark_ant,dw_stark_ant), ' [Hz]')
     print('Mittel:  w_m =', form(w_mittel_ant,dw_mittel_ant), ' [Hz]')
```

```

print('Schwach: w_w =', form(w_schwach_ant,dw_schwach_ant), ' [Hz]\n')
print('Abweichungen:\n')
print('=> sig_sm = ',sig(w_stark_ant,dw_stark_ant,w_mittel_ant,dw_mittel_ant))
print('=> sig_mw = ',sig(w_mittel_ant,dw_mittel_ant,w_schwach_ant,dw_schwach_ant))
print('=> sig_sw = ',sig(w_stark_ant,dw_stark_ant,w_schwach_ant,dw_schwach_ant))

```

Eigenfrequenzen antisymmetrische Schwingung:

Stark: w_s = 0.726 ± 0.031 [Hz]
Mittel: w_m = 0.658 ± 0.026 [Hz]
Schwach: w_w = 0.632 ± 0.031 [Hz]

Abweichungen:

=> sig_sm = 1.383
=> sig_mw = 0.202
=> sig_sw = 2.258

Die σ -Abweichung ist nun deutlich höher als im symmetrischen Fall, teilweise um eine Größenordnung mehr. Die Eigenfrequenz ω_2^{anti} des antisymmetrischen Systems ist somit **abhängig** von der Kopplung.

$$\omega_2^{\text{anti}} = \sqrt{\frac{D + 2D'}{J}} \quad (44)$$

Sie kann also nicht aus dem Mittelwert errechnet werden und muss für jede Kopplung einzeln aufgenommen werden. Es folgt somit:

Kopplung	Eigenfrequenz ω_2^{anti} [Hz]
(stark)	0.726 ± 0.031
(mittel)	0.658 ± 0.026
(schwach)	0.623 ± 0.031

5.4 Teil 3: Schwebungsschwingung

5.4.1 Berechnung der Eigenfrequenz ω_I Schwebungsfrequenz ω_{II}

```

[29]: def plot_schwingung_doppelgauss(n,title,h,a,b):
    t, phi1, phi2 = open_data(n)
    F1, A1, F2, A2 = fastfourier(n)

    from scipy.signal import find_peaks
    peaks1, _ = find_peaks(A1, height=0.4)
    init_vals1 = [0.002, F1[peaks1][0], 0.005, 0.002, F1[peaks1][1], 0.005]
    peaks2, _ = find_peaks(A2, height=0.4)
    init_vals2 = [0.002, F2[peaks2][0], 0.005, 0.002, F2[peaks2][1], 0.005]

    popt1, pcov1 = curve_fit(gaussian2,F1, A1, p0=init_vals1)
    perr1 = np.sqrt(np.diag(pcov1))
    popt2, pcov2 = curve_fit(gaussian2,F2, A2, p0=init_vals2)
    perr2 = np.sqrt(np.diag(pcov2))

```

```

fig, axs = plt.subplots(2, 2, constrained_layout=True)

axs[0,0].plot(t, phi1)
#axs[0].set_xlabel('time [sec]')
axs[0,0].set_ylabel('Auslenkwinkel  $\varphi_1$  [deg]')
axs[0,0].grid(ls=':')

axs[1,0].plot(t, phi2,c='red')
axs[1,0].set_xlabel('Zeit  $t$  [sec]')
axs[1,0].set_ylabel('Auslenkwinkel  $\varphi_2$  [deg]')
axs[1,0].grid(ls=':')

axs[0,1].plot(F1, A1)
axs[0,1].plot(F2, gaussian2(F2,popt2[0],popt2[1],popt2[2],
                           popt2[3],popt2[4],popt2[5]),c='g',ls='--',
              label=r'$\mu_1 = $'+format_plt(popt1[1])+' Hz\n'
              +r'$\sigma_1 = $'+format_plt(popt1[2])+' Hz\n'
              +r'$FWHM_1 = $'+format_plt(popt1[2]*2.355)+' Hz\n'
              +r'$\mu_2 = $'+format_plt(popt1[4])+' Hz\n'
              +r'$\sigma_2 = $'+format_plt(popt1[5])+' Hz\n'
              +r'$FWHM_2 = $'+format_plt(popt1[5]*2.355)+' Hz')
axs[0,1].set_xlim(a,b)
axs[0,1].set_ylabel('Amplitude  $A_1$  [dBm]')
axs[0,1].grid(ls=':')
axs[0,1].legend(loc='best', fontsize=14)

axs[1,1].plot(F2, A2,c='red')
axs[1,1].plot(F2, gaussian2(F2,popt2[0],popt2[1],popt2[2],
                           popt2[3],popt2[4],popt2[5]),c='g',ls='--',
              label=r'$\mu_1 = $'+format_plt(popt2[1])+' Hz\n'
              +r'$\sigma_1 = $'+format_plt(popt2[2])+' Hz\n'
              +r'$FWHM_1 = $'+format_plt(popt2[2]*2.355)+' Hz\n'
              +r'$\mu_2 = $'+format_plt(popt2[4])+' Hz\n'
              +r'$\sigma_2 = $'+format_plt(popt2[5])+' Hz\n'
              +r'$FWHM_2 = $'+format_plt(popt2[5]*2.355)+' Hz')
axs[1,1].set_xlim(a,b)
axs[1,1].set_xlabel('Frequenz  $f$  [Hz]')
axs[1,1].set_ylabel('Amplitude  $A_2$  [dBm]')
axs[1,1].grid(ls=':')
axs[1,1].legend(loc='best', fontsize=14)

fig.suptitle(title)
plt.show()

wII1,dwII1,wII2,dwII2 = popt1[1], popt1[2]*2.355, popt2[1], popt2[2]*2.355
wI1,dwI1,wI2,dwI2 = popt1[4], popt1[5]*2.355, popt2[4], popt2[5]*2.355

diff_peaks1 = np.abs(np.diff(A1[peaks1]))
diff_peaks2 = np.abs(np.diff(A1[peaks1]))

print('\nGaussfit 1:')
fitgüte(F1,A1,diff_peaks1,popt1,gaussian2)
print('\nGaussfit 2:')

```

```

fitgüte(F2,A2,diff_peaks2,popt2,gaussian2)

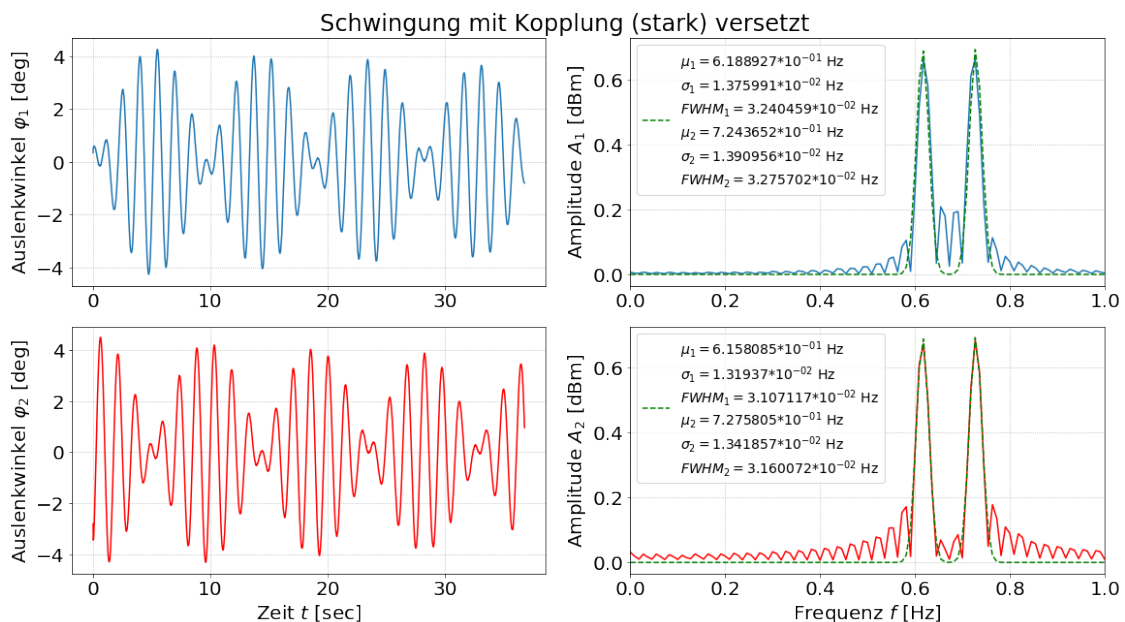
print('\nEigenfrequenzen:\n')
print('Pendel 1: w_I_1 =', form(wI1,dwI1), ' [Hz] ')
print('Pendel 2: w_I_2 =', form(wI2,dwI2), ' [Hz] ')
print('=> sig = ',sig(wI1,dwI1,wI2,dwI2), '\n')

print('Schwebungsfrequenzen:\n')
print('Pendel 1: w_II_1 =', form(wII1,dwII1), ' [Hz] ')
print('Pendel 2: w_II_2 =', form(wII2,dwII2), ' [Hz] ')
print('=> sig = ',sig(wII1,dwII1,wII2,dwII2), '\n')

w_I,dw_I = np.mean([wI1,wI2]), dwI1+dwI2
w_II,dw_II = np.mean([wII1,wII2]), dwII1+dwII2

return [w_I,dw_I],[w_II,dw_II]
a = 'Schwingung mit Kopplung (stark) versetzt'
Sch1s, Sch2s = plot_schwingung_doppelgauss(39,a,0.4,0,1)
b = 'Schwingung mit Kopplung (mittel) versetzt'
Sch1m, Sch2m = plot_schwingung_doppelgauss(32,b,0.4,0.3,0.8)
c = 'Schwingung mit Kopplung (schwach) versetzt'
Sch1w, Sch2w = plot_schwingung_doppelgauss(40,c,0.4,0.5,0.7)

```



Gaussfit 1:
chi_squared= 3.335984e+04
chi_squared_red= 1.519118e+01
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 0.0 %

Gaussfit 2:
chi_squared= 4.744221e+04

chi_squared_red= 2.160392e+01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 0.0 %

Eigenfrequenzen:

Pendel 1: $w_{I_1} = 0.724 \pm 0.033$ [Hz]

Pendel 2: $w_{I_2} = 0.728 \pm 0.032$ [Hz]

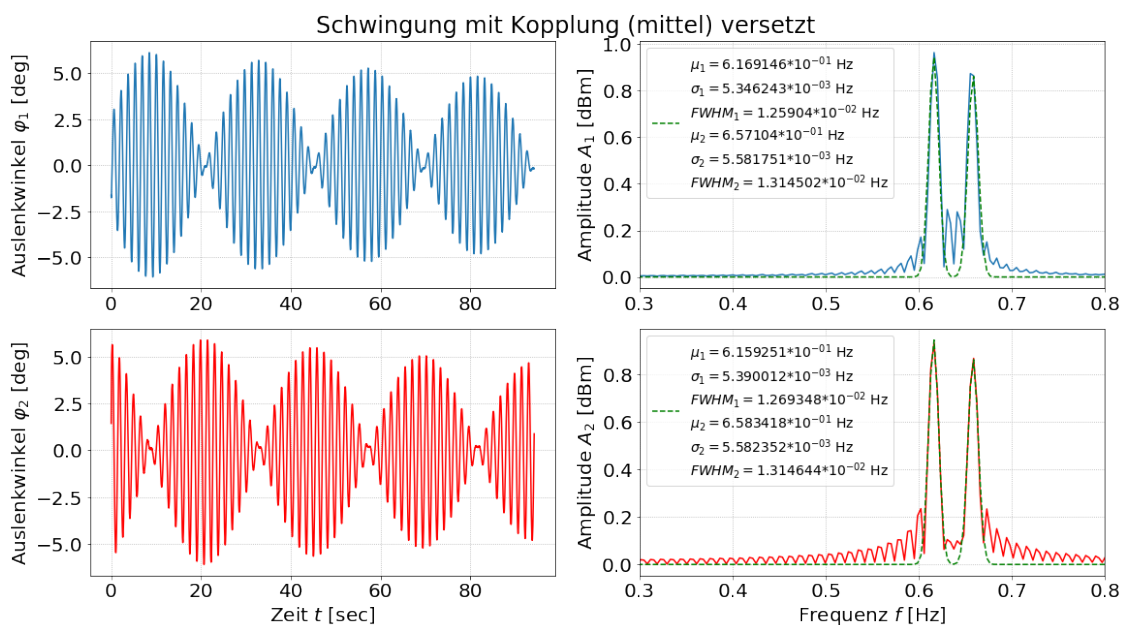
=> sig = 0.002

Schwebungsfrequenzen:

Pendel 1: $w_{II_1} = 0.619 \pm 0.032$ [Hz]

Pendel 2: $w_{II_2} = 0.616 \pm 0.031$ [Hz]

=> sig = 0.002



Gaussfit 1:

chi_squared= 4.649294e+01

chi_squared_red= 8.243429e-03

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 5.961257e+01

chi_squared_red= 1.056961e-02

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Eigenfrequenzen:

Pendel 1: $w_{I_1} = 0.657 \pm 0.013$ [Hz]

Pendel 2: $w_{I_2} = 0.658 \pm 0.013$ [Hz]

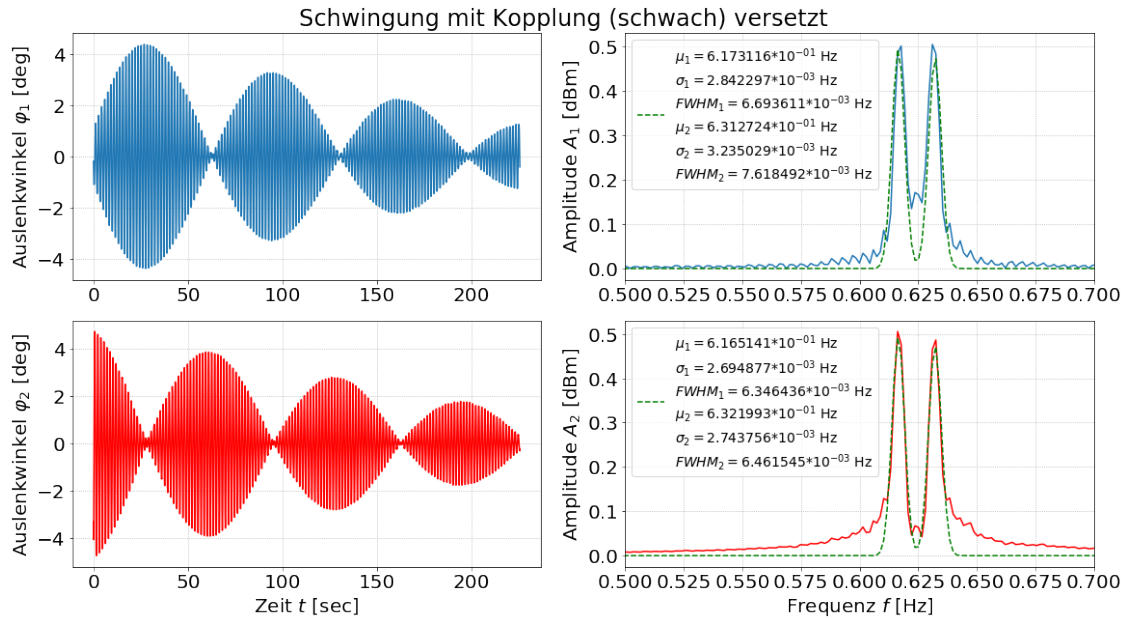
=> sig = 0.002

Schwebungsfrequenzen:

Pendel 1: $w_{II_1} = 0.617 \pm 0.013$ [Hz]

Pendel 2: $w_{II_2} = 0.616 \pm 0.013$ [Hz]

=> sig = 0.002



Gaussfit 1:

chi_squared= 5.617307e+03

chi_squared_red= 4.162202e-01

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 100.0 %

Gaussfit 2:

chi_squared= 1.39161e+04

chi_squared_red= 1.031128e+00

Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten: 1.0 %

Eigenfrequenzen:

Pendel 1: $w_{I_1} = 0.631 \pm 0.008$ [Hz]

Pendel 2: $w_{I_2} = 0.632 \pm 0.006$ [Hz]

=> sig = 0.004

Schwebungsfrequenzen:

Pendel 1: $w_{II_1} = 0.617 \pm 0.007$ [Hz]

Pendel 2: $w_{II_2} = 0.617 \pm 0.006$ [Hz]

=> sig = 0.004

5.5 Vergleich von ω_I und ω_{II} bei verschiedenen Kopplungen

```
[10]: w_I_s,dw_I_s,w_II_s,dw_II_s = Sch1s[0],Sch1s[1],Sch2s[0],Sch2s[1]
      w_I_m,dw_I_m,w_II_m,dw_II_m = Sch1m[0],Sch1m[1],Sch2m[0],Sch2m[1]
      w_I_w,dw_I_w,w_II_w,dw_II_w = Sch1w[0],Sch1w[1],Sch2w[0],Sch2w[1]

      print('\nEigenfrequenzen w_I bei verschiedenen Kopplungen:\n')
      print('Stark:   w_I_s =', form(w_I_s,dw_I_s), ' [Hz]')
      print('Mittel:  w_I_m =', form(w_I_m,dw_I_m), ' [Hz]')
      print('Schwach: w_I_w =', form(w_I_w,dw_I_w), ' [Hz]\n')
      print('Abweichungen:\n')
      print('=> sig_sm = ',sig(w_I_s,dw_I_s,w_I_m,dw_I_m))
      print('=> sig_mw = ',sig(w_I_m,dw_I_m,w_I_w,dw_I_w))
      print('=> sig_sw = ',sig(w_I_s,dw_I_s,w_I_w,dw_I_w),'\n')

      print('\nSchwebungsfrequenzen w_II bei verschiedenen Kopplungen:\n')
      print('Stark:   w_II_s =', form(w_II_s,dw_II_s), ' [Hz]')
      print('Mittel:  w_II_m =', form(w_II_m,dw_II_m), ' [Hz]')
      print('Schwach: w_II_w =', form(w_II_w,dw_II_w), ' [Hz]\n')
      print('Abweichungen:\n')
      print('=> sig_sm = ',sig(w_II_s,dw_II_s,w_II_m,dw_II_m))
      print('=> sig_mw = ',sig(w_II_m,dw_II_m,w_II_w,dw_II_w))
      print('=> sig_sw = ',sig(w_II_s,dw_II_s,w_II_w,dw_II_w))
```

Eigenfrequenzen w_I bei verschiedenen Kopplungen:

```
Stark:   w_I_s = 0.726 ± 0.064 [Hz]
Mittel:  w_I_m = 0.658 ± 0.026 [Hz]
Schwach: w_I_w = 0.632 ± 0.014 [Hz]
```

Abweichungen:

```
=> sig_sm = 0.567
=> sig_mw = 0.414
=> sig_sw = 1.443
```

Schwebungsfrequenzen w_{II} bei verschiedenen Kopplungen:

```
Stark:   w_II_s = 0.617 ± 0.063 [Hz]
Mittel:  w_II_m = 0.616 ± 0.025 [Hz]
Schwach: w_II_w = 0.617 ± 0.013 [Hz]
```

Abweichungen:

```
=> sig_sm = 0.0
=> sig_mw = 0.0
=> sig_sw = 0.0
```

Die Eigenfrequenz ω_I des Systems weist keine signifikanten Abweichungen mit steigender Kopplung auf (max. 0.34σ), während diese in der Schwebungsfrequenz ω_{II} deutlich zu erkennen sind (min. 8.31σ).

Außerdem soll überprüft werden wie ω_I , ω_{II} sich gemäß der Theorie verhalten:

$$\omega_I = \frac{1}{2}(\omega_1^{\text{sys}} + \omega_2^{\text{anti}}) \quad (45)$$

$$\omega_{II} = \frac{1}{2}(\omega_2^{\text{anti}} - \omega_1^{\text{sys}}) \quad (46)$$

```
[11]: def vergleich_theo(stärke):
    if stärke == 'Stark':
        w_ant = w_stark_ant
        dw_ant = dw_stark_ant
        w_I = w_I_s
        dw_I = dw_I_s
        w_II = w_II_s
        dw_II = dw_II_s
        a = 's'
    if stärke == 'Mittel':
        w_ant = w_mittel_ant
        dw_ant = dw_mittel_ant
        w_I = w_I_m
        dw_I = dw_I_m
        w_II = w_II_m
        dw_II = dw_II_m
        a = 'm'
    if stärke == 'Schwach':
        w_ant = w_schwach_ant
        dw_ant = dw_schwach_ant
        w_I = w_I_w
        dw_I = dw_I_w
        w_II = w_II_w
        dw_II = dw_II_w
        a = 'w'

    w_I_theo = 1/2*(w_sys + w_ant)
    dw_I_theo = np.sqrt((0.5*dw_sys)**2+(dw_ant)**2)
    w_II_theo = 1/2*(w_ant - w_sys)
    dw_II_theo = np.sqrt((0.5*dw_sys)**2+(dw_ant)**2)

    print('\n'+stärke+':')
    print('w_I_'+stärke+' = ', form(w_I,dw_I), ' [Hz]')
    print('w_I_'+stärke+'_theo = ', form(w_I_theo,dw_I_theo), ' [Hz]')
    print('=> sig = ', sig(w_I,dw_I,w_I_theo,dw_I_theo))
    print('w_II_'+stärke+' = ', form(w_II,dw_II), ' [Hz]')
    print('w_II_'+stärke+'_theo = ', form(w_II_theo,dw_II_theo), ' [Hz]')
    print('=> sig = ', sig(w_II,dw_II,w_II_theo,dw_II_theo))
    vergleich_theo('Stark')
    vergleich_theo('Mittel')
    vergleich_theo('Schwach')
```

Stark:

```
w_I_Stark      = 0.726 ± 0.064  [Hz]
w_I_Stark_theo = 0.671 ± 0.036  [Hz]
```

```

=> sig = 0.301
w_II_Stark      = 0.617 ± 0.063 [Hz]
w_II_Stark_theo = 0.054 ± 0.036 [Hz]
=> sig = 32.277

Mittel:
w_I_Mittel      = 0.658 ± 0.026 [Hz]
w_I_Mittel_theo = 0.637 ± 0.032 [Hz]
=> sig = 0.122
w_II_Mittel     = 0.616 ± 0.025 [Hz]
w_II_Mittel_theo = 0.021 ± 0.032 [Hz]
=> sig = 108.587

Schwach:
w_I_Schwach     = 0.632 ± 0.014 [Hz]
w_I_Schwach_theo = 0.624 ± 0.036 [Hz]
=> sig = 0.021
w_II_Schwach    = 0.617 ± 0.013 [Hz]
w_II_Schwach_theo = 0.008 ± 0.036 [Hz]
=> sig = 154.442

```

5.5.1 Zusammenfassung der Ergebnisse

Die numerisch errechneten Werte der Eigenfrequenzen ω_I stimmen gut mit den theoretischen Vorhersagen überein. Die σ -Abweichungen sind sehr gering. Die Ergebnisse für den Vergleich der Eigenfrequenzen ω_I der Schwebungsschwingung sind in folgender Tabelle zusammengetragen:

Kopplung	ω_1^{sym} [Hz]	ω_2^{anti} [Hz]	ω_I^{fft} [Hz]	ω_I^{theo} [Hz]	σ -Abw.
(stark)	0.617 ± 0.039	0.726 ± 0.031	0.726 ± 0.064	0.671 ± 0.036	0.30
(mittel)	0.617 ± 0.039	0.658 ± 0.026	0.658 ± 0.026	0.637 ± 0.037	0.12
(schwach)	0.617 ± 0.039	0.623 ± 0.031	0.623 ± 0.014	0.624 ± 0.036	0.02

Die Numerische berechnung der Werte für ω_{II} und die theoretische Vorhersage aus den zuvor bestimmten Messwerten stimmen nicht überein, die Abweichungen sind **sehr** hoch. Die Ergebnisse für den Vergleich der Schwebungsfrequenzen ω_{II} in der Schwebungsschwingung sind in folgender Tabelle zusammengetragen:

Kopplung	ω_1^{sym} [Hz]	ω_2^{anti} [Hz]	ω_{II}^{fft} [Hz]	$\omega_{II}^{\text{theo}}$ [Hz]	σ -Abw.
(stark)	0.617 ± 0.039	0.726 ± 0.031	0.617 ± 0.036	0.054 ± 0.036	32.28
(mittel)	0.617 ± 0.039	0.658 ± 0.026	0.616 ± 0.025	0.021 ± 0.032	108.59
(schwach)	0.617 ± 0.039	0.623 ± 0.031	0.617 ± 0.013	0.008 ± 0.036	154.44

Auch die mit der alternativen Definition für ω_{II} , ist die geringste σ -Abweichung die, der stärksten Kopplung, mit $\approx 26\sigma$ und immernoch **signifikant** verschieden.

5.6 Teil 4: Kopplungsgrad κ

```

[12]: def Kopplungsgrad(w1,dw1,w2,dw2):
      k = np.abs((w1**2-w2**2))/(w1**2+w2**2)
      dk = np.sqrt(((4*w1*w2**2)/(w1**2+w2**2)**2 *dw1)**2+
                    ((4*w1**2*w2)/(w1**2+w2**2)**2*dw2)**2)

```

```

    return k, dk

def Kappa(stärke):
    if stärke == 'Stark ':
        w_ant = w_stark_ant
        dw_ant = dw_stark_ant
        a = 's'
    if stärke == 'Mittel ':
        w_ant = w_mittel_ant
        dw_ant = dw_mittel_ant
        a = 'm'
    if stärke == 'Schwach':
        w_ant = w_schwach_ant
        dw_ant = dw_schwach_ant
        a = 'w'

    k, dk = Kopplungsgrad(w_sys, dw_sys, w_ant, dw_ant)
    print(stärke+' : k_'+a+' = ', format_e(k), '±', format_e(dk))
    #print(stärke+' : k_'+a+' = ', form(k, dk))
    return k, dk

print('\nKopplungsgrad:\n')
k_s, dk_s = Kappa('Stark ')
k_m, dk_m = Kappa('Mittel ')
k_w, dk_w = Kappa('Schwach')
print('')

```

Kopplungsgrad:

```

Stark   : k_s =  1.612167e-01 ± 6.972389e-02
Mittel  : k_m =  6.501623e-02 ± 6.994455e-02
Schwach : k_w =  2.48538e-02 ± 7.588058e-02

```

Kopplung	Kopplungsgrad κ
(stark)	0.161 ± 0.070
(mittel)	0.065 ± 0.070
(schwach)	0.025 ± 0.076

5.6.1 Verhältnisse der Kopplungsgrade

```

[13]: ls, lm, lw, dl = 0.40, 0.25, 0.15, 0.002 # Kopplungslänge [m]

diff_k_sm, ddiff_k_sm = k_s/k_m, max(dk_s, dk_m)
diff_k_mw, ddiff_k_mw = k_m/k_w, max(dk_m, dk_w)
diff_k_sw, ddiff_k_sw = k_s/k_w, max(dk_s, dk_w)

diff_l_sm, ddiff_l_sm = ls**2/lm**2, ls
diff_l_mw, ddiff_l_mw = lm**2/lw**2, lm
diff_l_sw, ddiff_l_sw = ls**2/lw**2, ls

print('\nVerhältnisse von k und l\n')
print('diff_k_sm = ', form(diff_k_sm, ddiff_k_sm))

```

```

print('diff_l_sm = ', form(diff_l_sm,ddiff_l_sm))
print('=> sig = ', sig(diff_k_sm,ddiff_k_sm,diff_l_sm, ddiff_l_sm),'\n')
print('diff_k_mw = ', form(diff_k_mw,ddiff_k_mw))
print('diff_l_mw = ', form(diff_l_mw,ddiff_l_mw))
print('=> sig = ', sig(diff_k_mw,ddiff_k_mw,diff_l_mw, ddiff_l_mw),'\n')
print('diff_k_sw = ', form(diff_k_sw,ddiff_k_sw))
print('diff_l_sw = ', form(diff_l_sw,ddiff_l_sw))
print('=> sig = ', sig(diff_k_sw,ddiff_k_sw,diff_l_sw, ddiff_l_sw),'\n')

```

Verhältnisse von κ und l

```

diff_k_sm = 2.480 ± 0.070
diff_l_sm = 2.560 ± 0.400
=> sig = 0.029

```

```

diff_k_mw = 2.616 ± 0.076
diff_l_mw = 2.560 ± 0.400
=> sig = 0.116

```

```

diff_k_sw = 6.487 ± 0.076
diff_l_sw = 7.111 ± 0.400
=> sig = 1.722

```

Der Vergleich zeigt, dass die Verhältnisse der Kopplungsgraden κ mit denen der Kopplungslängenquadraten l^2 gut übereinstimmen. Dabei ist die Übereinstimmung bei stärkerer Kopplung am höchsten. Die Ergebnisse sind in folgender Tabelle zusammengetragen:

Verhältniss	Kopplungsgrad κ	Kopplungslängen-quadrat l^2	σ -Abw.
(stark)/(mittel)	2.480 ± 0.070	2.560 ± 0.400	0.03
(mittel)/(schwach)	2.616 ± 0.076	2.560 ± 0.400	0.12
(stark)/(schwach)	6.487 ± 0.076	7.111 ± 0.400	1.72

5.7 Teil 5: Gekoppelter elektrischer Schwingkreis

Beim zweiten Aufbau handelt es sich um einen elektrischen Schwingkreis welcher über eine Induktivität mit einem anderen gekoppelt ist.

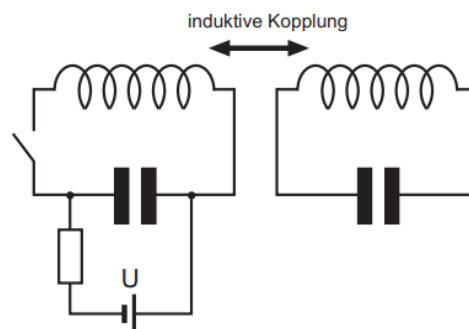


Figure 2: Gekoppelter elektrische Schwingkreis.

Die Kopplung erfolgt induktiv über die Spulen. Der Kondensator im linken Schaltkreis wird über den Widerstand geladen. Wird der Schalter geschlossen, so wird im linken Kreis eine Schwingung angeregt. Durch die Kopplung überträgt sich allmählich, ganz analog zu den gekoppelten Pendeln, die Energie in den rechten Schwingkreis und wieder zurück. Die Oszillation kann durch Abgriff über den Spulen mit einem Oszilloskop dargestellt werden.

5.8 Qualitative Beobachtungen

Ziel war es die Schwebungsschwingung, welche wir im vorherigen Versuchsteil (5.4) untersucht haben, am elektrischen Schwingkreis nachzustellen. Am Oszilloskop konnten wir folgende Schwingungsbilder ablesen:

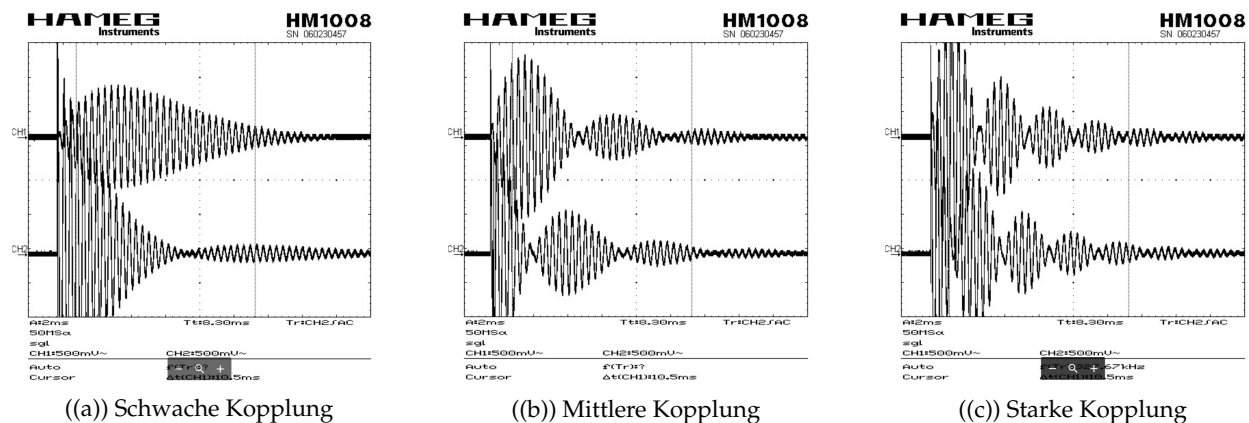


Figure 3: Die Kopplungsstärke ist antiproportional zum Abstand d der beiden Induktivitäten. Je kleiner der Abstand d desto größer die induktive Kopplung κ_L

Zu erkennen ist wie die Kopplungsstärke variiert und so die Schwebungsfrequenz immer dominanter am Verlauf der Schwingung ist. Außerdem ist eine deutliche Abnahme der Amplitude im Vergleich zum mechanischen Pendel zu erkennen, dies liegt an dem eingebauten Widerstand (siehe Abbildung 2), welcher die starke Dämpfung verursacht. Beim mechanischen Pendel ist kaum eine Dämpfung zu erkennen, bis auf den längeren Zeitskalen die wir gemessen haben um die Schwebung bei schwacher Kopplung besser sichtbar zu machen (siehe 5.4). Die des Widerstandes R scheint aber deutlich stärker zu sein.

6 Diskussion

Die Auswertung wurde erneut in Python durchgeführt um zu gewährleisten dass beide Pendel in der Frequenzanalyse betrachtet werden. Die so errechneten Frequenzen, im selben Spektrum, unterscheiden sich teilweise mit denen die das Pythonprogramm im Labor errechnet hat. Die wahrscheinlichste Ursache dafür ist eine andere peaksuchende Fitfunktion die im Versuchsrechner verwendet wurde, welche nicht in den Standardmodulen die auf den CIP-Pool servern installiert sind enthalten ist.

6.1 Hohe σ -Abweichungen bei ω_{II}

Bei der Berechnung der Eigen- und Schwebungsfrequenzen ω_I , ω_{II} (siehe 5.4.1) konnte keine hohe Übereinstimmung der theoretisch vorhersagten und der numerisch errechneten Schwebungsfrequenz ω_{II} gefunden werden. Mögliche Ursachen dafür die sensiblen Anfangsbedingungen welche wir per Hand vorbereitet haben. Diese haben vermutlich auch unseren Wert der Eigenfrequenzen ω_1 , ω_2 der Symmetrischen (5.3.1) und antisymmetrischen (5.3.3) Schwingung beeinflusst haben. Hier kann ein systematischer Fehler entstanden sein, welcher sich durch das gesamte Experiment gezogen hat, einen Wichtigen Hinweis liefern die Oszillationsbilder, besonders die der antisymmetrischen Schwingung zeigen eine leichte Schwebung (siehe 5.3.3), hieran ist zu erkennen wie sensibel die Anfangsbedingungen sind. Eine weitere mögliche Fehlerquelle ist vermutlich der statistische Fehler aus der Fitfunktion, die Fitgüte zeigt dass diese teilweise hohe Ungenauigkeiten aufweist, diese sind teilweise auch in den jeweiligen Abbildungen zu erkennen.

6.2 Fazit

Abgesehen von der Schwebungsfrequenz ω_{II} weisen die anderen Messwerte keine hohen σ -Abweichungen auf. In allen anderen Versuchsteilen sind theoretische Vorhersagen und gemessene Größen miteinander gut verträglich. Das mechanische Pendel scheint deutlich genauere Messungen zu ermöglichen, als zunächst erwartet. Der Einfluss von Luftreibung und Aufhängung sowie sonstige Störungen, scheinen für die kleinen Zeitskalen in denen wir gemessen haben keine ausschlaggebende Rolle zu spielen.

Der nicht zwingend notwendige Ansatz, die Peaks im Frequenzspektrum selbst berechnen zu wollen, um beide Pendel in unserer Analyse zu berücksichtigen. Bestätigte hervorragend die Vermutung, dass die beiden Pendel in sehr guter Näherung äquivalente Ergebnisse liefern. Allerdings waren wir nicht in der Lage die Funktion so akkurat zu fitten wie es das im Versuchsraum verwendete Pythonprogramm schafft. Und da wir unerklärlich hohe Abweichungen bei der Betrachtung von ω_{II} gefunden haben, müssen wir die von uns verwendete Fitfunktion als mögliche Fehlerquelle betrachten. Allerdings sind hier weitere Untersuchungen notwendig um genauere Aussagen machen zu können.