

Vierpole_und_Leitungen

December 9, 2023

1 Fakultät für Physik

1.1 Physikalisches Praktikum P1 für Studierende der Physik

Versuch P1-53, 54, 55 (Stand: September 2023)

[Raum F1-13](#)

2 Vierpole und Leitungen

Name: *surname1* Vorname: *name1* E-Mail: *mail1*

Name: *surname2* Vorname: *name2* E-Mail: *mail2*

Gruppennummer: *groupnumber*

Betreuer:

Versuch durchgeführt am: *date*

Beanstandungen:

Testiert am: _____ Testat: _____

```
[1]: # Hilfsfunktionen. Nicht weiter beachten
from IPython.core.display import Markdown
from IPython.display import display
def M_helper_func(txt):
    #print(txt)
    #print('fr'+txt.replace("$", "$").replace("{", "{{").replace("}", "}}").
    ↪replace("[", "{").replace("!", "!")+''')
    excreturnvalue = ""
    loc = {}
    exec('excreturnvalue = fr"<div style=\'margin-left: 8px;\'>'+txt.
    ↪replace("{", "{{").replace("}", "}}").replace("[", "{").replace("!", "!")+</
    ↪div>'", globals(), loc)
    display(Markdown(loc["excreturnvalue"]))
    return loc["excreturnvalue"]

from IPython.core.magic import (register_line_magic,
                                register_cell_magic)

from IPython.core import magic

def register_line_magic(f):
    setattr(f, magic.MAGIC_NO_VAR_EXPAND_ATTR, True)
    return magic.register_line_magic(f)

@register_line_magic
def m(line):
    M_helper_func(line)
```

```

import os
import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import scipy as sci
import kafe2
import pandas as pd
import uncertainties
from uncertainties import ufloat, unumpy
from uncertainties.unumpy import uarray as uarr
from uncertainties.unumpy import nominal_values, nominal_values as nomv, u
↳std_devs, std_devs as stdv
from uncertainties.umath import sqrt as usqrt, sin as usin,exp as uexp
import scipy
import matplotlib.image as mpimg
from io import StringIO

mpl.rcParams['figure.dpi'] = 100
class ng:
    def __init__(self,x=None,stat=None,syst=None):
        if x == None:
            self.syst = None
            self.stat = None
        else:
            self.syst = ufloat(x,0)
            self.stat = ufloat(x,0)
            if syst != None:
                self.syst = ufloat(x,syst)
            if stat != None:
                self.stat = ufloat(x,stat)
    def SetSyst(self,syst):
        self.syst = ufloat(nomv(self.syst),syst)
    def SetStat(self,stat):
        self.stat = ufloat(nomv(self.stat),stat)
    def __str__(self):
        return f'{nomv(self.stat):.2}± {stdv(self.stat):.2}± {stdv(self.syst):
↳.2}'
    def __add__(self,o):
        res = ng()
        if type(o) == ng:
            res.syst = self.syst + o.syst
            res.stat = self.stat + o.stat
        else:
            res.syst = self.syst + o
            res.stat = self.stat + o
        return res

```

```

def __iadd__(self,o):
    return self.__add__(o)
def __sub__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst - o.syst
        res.stat = self.stat - o.stat
    else:
        res.syst = self.syst - o
        res.stat = self.stat - o
    return res
def __isub__(self,o):
    return self.__sub__(o)
def __mul__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst * o.syst
        res.stat = self.stat * o.stat
    else:
        res.syst = self.syst * o
        res.stat = self.stat * o
    return res
def __imul__(self,o):
    return self.__mul__(o)
def __truediv__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst / o.syst
        res.stat = self.stat / o.stat
    else:
        res.syst = self.syst / o
        res.stat = self.stat / o
    return res
def __itruediv__(self,o):
    return self.__truediv__(o)
def __pow__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst**o.syst
        res.stat = self.stat**o.stat
    else:
        res.syst = self.syst**o
        res.stat = self.stat**o
    return res
def __ipow__(self,o):
    return self.__pow__(o)
def sin(o):

```

```

    if type(o) == ng:
        res = ng()
        res.syst = usin(o.syst)
        res.stat = usin(o.stat)
        return res
    else:
        return np.sin(o)
def exp(o):
    if type(o) == ng:
        res = ng()
        res.syst = uexp(o.syst)
        res.stat = uexp(o.stat)
        return res
    else:
        return np.exp(o)
def sqrt(o):
    if type(o) == ng:
        res = ng()
        res.syst = usqrt(o.syst)
        res.stat = usqrt(o.stat)
        return res
    else:
        return np.sqrt(o)

def uplot(x,y,*args, **kwargs):
    utypes = (uncertainties.core.
Variable,uncertainties.core.AffineScalarFunc)
    xplt = x
    yplt = y
    if type(x[0]) in utypes:
        xplt = nomv(x)
        kwargs["xerr"] = stdv(x)
    if type(y[0]) in utypes:
        yplt = nomv(y)
        kwargs["yerr"] = stdv(y)
    plt.errorbar(xplt,yplt,*args,**kwargs)

def plting(path):
    try:
        img = mpimg.imread(path)
        imgplot = plt.imshow(img)
        plt.axis('off')
        plt.show()
    except Exception as e:
        print("Image Not Found")
        print(e)

```

2.0.1 Aufgabe 1.1

```
[2]: C = 1E-9
f = 1.7E3
    = 2*np.pi*f
U_e = 8.0

%m Bei der Aufgabe geht es darum, die Übertragungsfunktion eines
↳ Hochpassfilters zu untersuchen. Wir haben eine Eingangsspannung von [!U_e!]V
↳ an den Hochpassfilter angelegt. Dabei haben wir bei verschiedenen Frequenzen
↳ folgende Daten gemessen:

a11 = """
R,DeltaT,U_a
1E3,144E-6,200E-3
1E4,136E-6,840E-3
1E5,92E-6,5.4
1E6,16E-6,7.12
"""

mpg = pd.read_csv(StringIO(a11))
mpg.rename(columns={"R": 'Widerstand in Ohm', "DeltaT": "Δt in s", "U_a":
↳ "Ausgangsspannung in V"}, inplace=True)
display(mpg)

%m Der verwendete Kondensator hat eine Kapazität von 1nF und es wurde bei einer
↳ Frequenz von 1.7KHz gemessen.
%m Es gilt  $\mu_0 = \frac{1}{C R}$  womit sich folgende zwei Plots ergeben:

R_theo = np.logspace(3,6,500)
_theo = 1/np.sqrt(1 + 1/(**2*C**2*R_theo**2))
Delta_theo = np.arctan(1/( *C*R_theo))
_0_theo = 1 / (R_theo * C)
log_theo = np.log10( / _0_theo)

R, Deltat, U_a = np.genfromtxt(StringIO(a11), delimiter=',')[1:].transpose()

    = U_a / U_e
Delta = 2*pi*Deltat*f
_0 = 1 / (R * C)
log = np.log10( / _0)

plt.title("Spannungsübertrag")
plt.plot(log_theo, _theo, "--", label="$\mu_{Theo}$")
plt.plot(log , , "X", label="$\mu$")
plt.legend()
plt.ylabel(r"Spannungsverhältnis  $\left|\frac{U_1}{U_0}\right|$ ")
```

```

plt.xlabel(r"$\log(\omega/\omega_{0})$")
plt.show()

plt.title("Phasendifferenz")
plt.plot(log_theo, Delta_theo, "--", label="$\Delta_{\text{Theo}}$")
plt.plot(log, Delta, "X", label="$\Delta$")
plt.ylabel(r"Phasendifferenz arg$\left(\frac{U_1}{U_0}\right)$")
plt.xlabel(r"$\log(\omega/\omega_{0})$")
plt.legend()
plt.show()

```

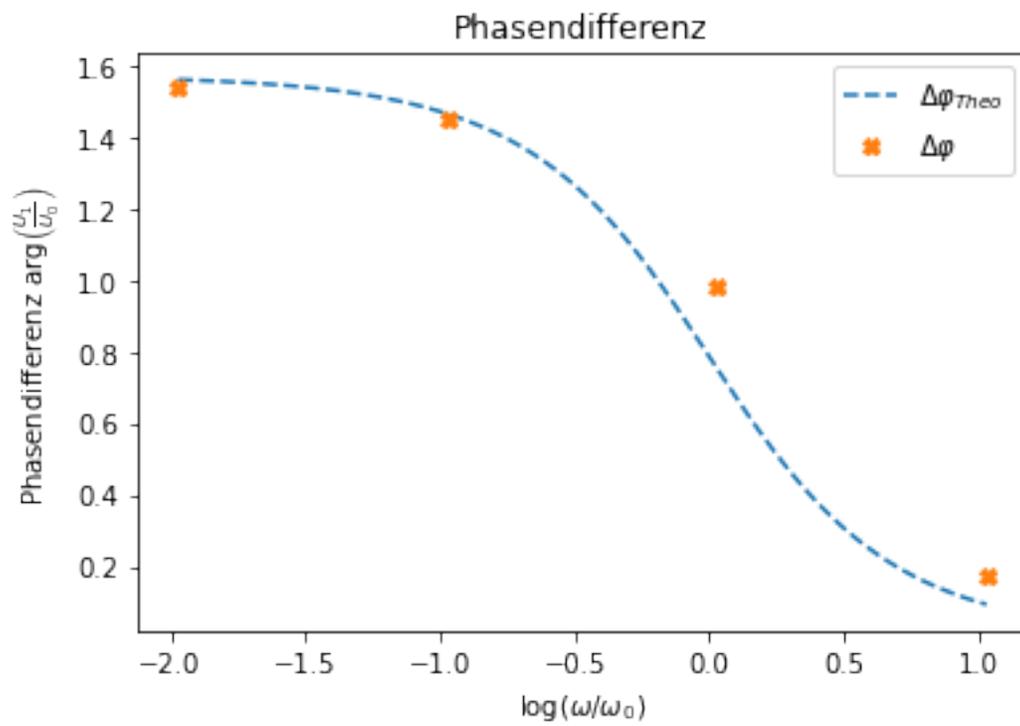
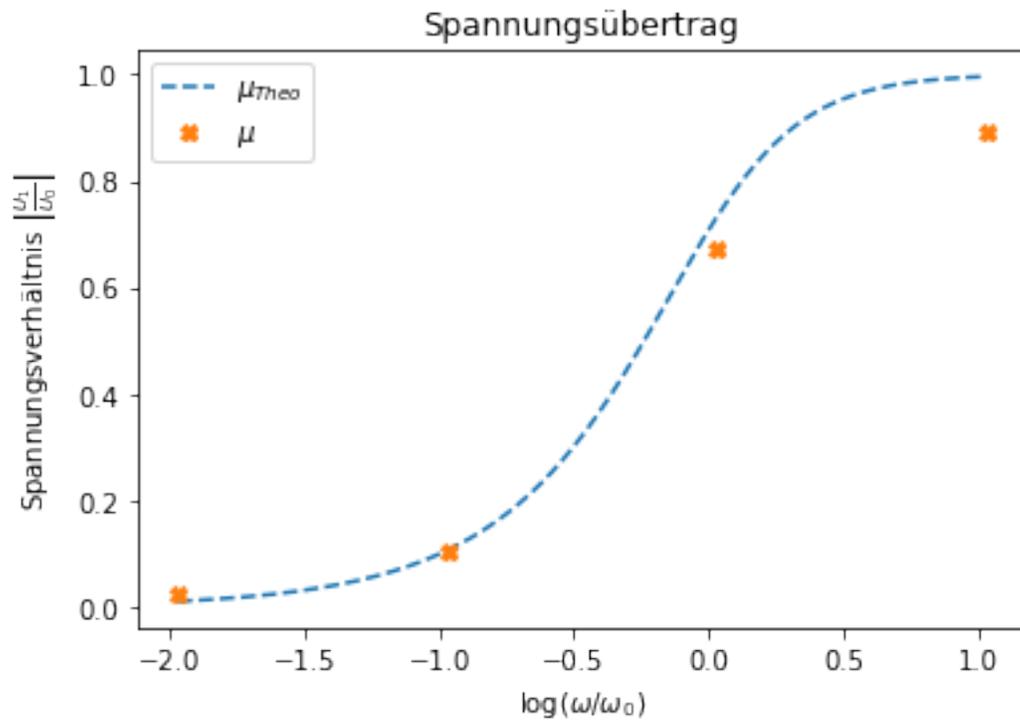
%m Wobei nach der Versuchsdokumentation die theoretischen Erwartungswerte $\mu = \frac{1}{\sqrt{\omega^2 C^2 + R^2}}$ sowie $\Delta_{\text{varphi}} = \arctan\left(\frac{1}{\omega C R}\right)$ als gestrichelte Linien aufgetragen sind. Es ist zu sehen, dass die Messwerte recht nah an den theoretischen Werten liegen. Die Abweichungen für höhere Frequenzen kommen dadurch zustande, dass das theoretische Modell die reellen Widerstände vernachlässigt.

Bei der Aufgabe geht es darum, die Übertragungsfunktion eines Hochpassfilters zu untersuchen. Wir haben eine Eingangsspannung von 8.0V an den Hochpassfilter angelegt. Dabei haben wir bei verschiedenen Frequenzen folgende Daten gemessen:

	Widerstand in Ohm	Δt in s	Ausgangsspannung in V
0	1000.0	0.000144	0.20
1	10000.0	0.000136	0.84
2	100000.0	0.000092	5.40
3	1000000.0	0.000016	7.12

Der verwendete Kondensator hat eine Kapazität von 1nF und es wurde bei einer Frequenz von 1.7KHz gemessen.

Es gilt $\omega_0 = \frac{1}{CR}$ womit sich folgende zwei Plots ergeben:



Wobei nach der Versuchsdokumentation die theoretischen Erwartungswerte $\mu = \frac{1}{\sqrt{\omega^2 C^2 R^2 + 1}}$ sowie $\Delta\varphi = \arctan\left(\frac{1}{\omega CR}\right)$ als gestrichelte Linien aufgetragen sind. Es ist zu sehen, dass die Messwerte recht nah an den theoretischen Werten liegen. Die Abweichungen für höhere Frequenzen kommen dadurch zustande, dass das theoretische Modell die realen Widerstände vernachlässigt.

2.0.2 Aufgabe 1.2: Differenzierglied

Wiederholen Sie die Messreihe aus Aufgabe 1.1, aber legen statt einer sinus- eine **dreieckförmige** Wechselspannung (gleicher Frequenz und Amplitude) am Eingang der Schaltung an. Für $\omega \ll \omega_0$ wird die Schaltung zum Differenzierglied. Beschreiben Sie was Sie sehen und vergleichen Sie Ihre Beobachtung mit Ihrer Erwartung.

```
[3]: U_e = 8

%m In diesem Versuch wird das differenzierte Verhalten des Hochpassfilters
↳ untersucht. Wir haben eine Dreiecksfrequenz von [!U_e!]V an den
↳ Hochpassfilter angelegt zu sehen auf Channel1 in gelb.

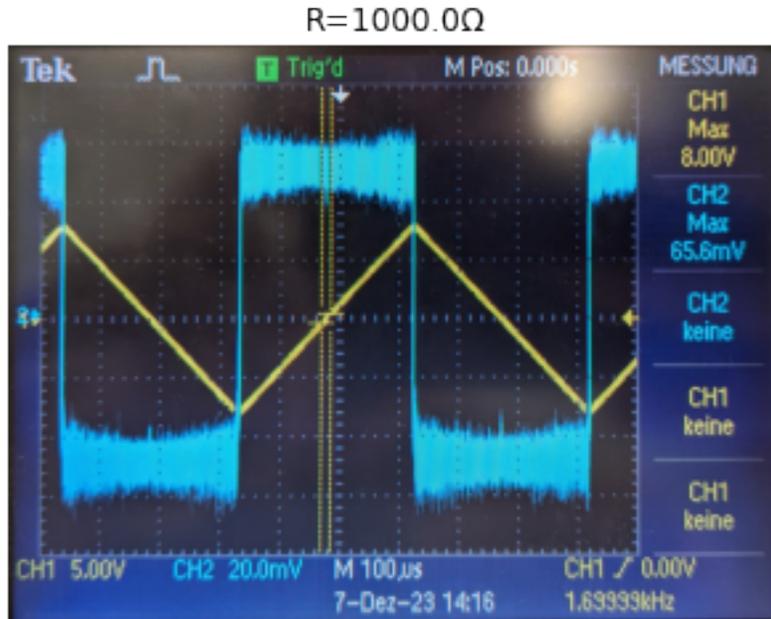
plt.title(f"R={R[0]}Ω")
pltimg("Fotos/A12_1.jpg")
%m Wie zu erwarten war, verhält sich der Hochpass wie ein Differenzierglied für
↳ den kleinsten Widerstand, da hier  $\omega \ll \omega_0$ . Das Oszilloskop
↳ zeigt also eine Rechteckwelle am Ausgang wobei ein hoher Anteil an Rauschen
↳ zu erkennen ist, da einerseits ein Differenzierglied immer das Rauschen auf
↳ dem Eingang verstärkt und andererseits die Signalspannung sehr stark
↳ geschwächt ist (hier nur noch 65.6mV am Ausgang) was das
↳ Signal-to-Noise-Ratio weiter verschlechtert.

plt.title(f"R={R[1]}Ω")
pltimg("Fotos/A12_2.jpg")
%m Auch für den zweitkleinsten Widerstand erzeugt der Hochpass noch eine
↳ Rechteckwelle am Ausgang wobei die Ecken abgerundet sind, weil der
↳ größere Widerstand dafür sorgt, dass bei gleicher Eingangsspannung weniger
↳ Strom fließt wodurch der Kondensator langsamer geladen wird.

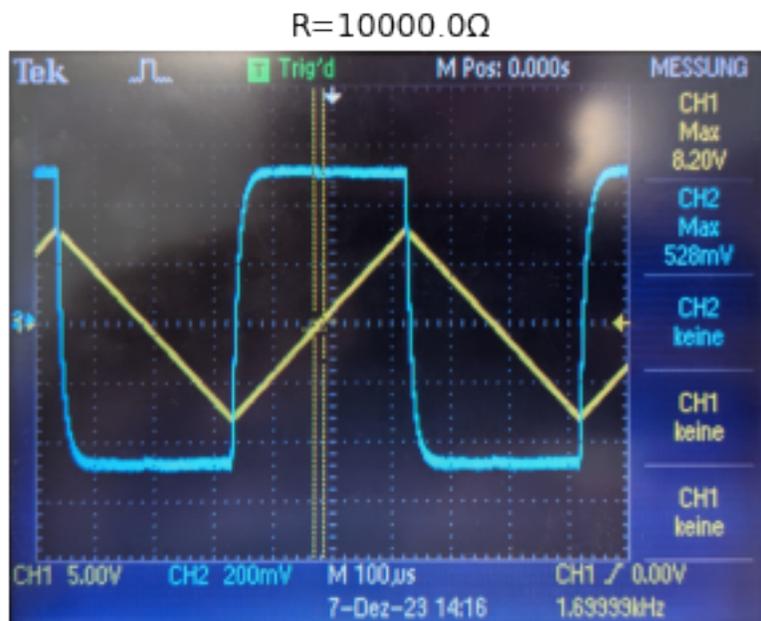
plt.title(f"R={R[2]}Ω")
pltimg("Fotos/A12_3.jpg")
%m Mit weiter steigendem Widerstand werden die Ecken weiter
↳ abgerundet.

plt.title(f"R={R[3]}Ω")
pltimg("Fotos/A12_4.jpg")
%m Im Grenzfall  $\omega \rightarrow \infty$  nähert sich das Ausgangssignal einer Dreieckswelle
↳ an ohne einen differenzierenden Effekt vorzuweisen. Hier ist jetzt  $\omega$ 
↳  $\gg \omega_0$ 
```

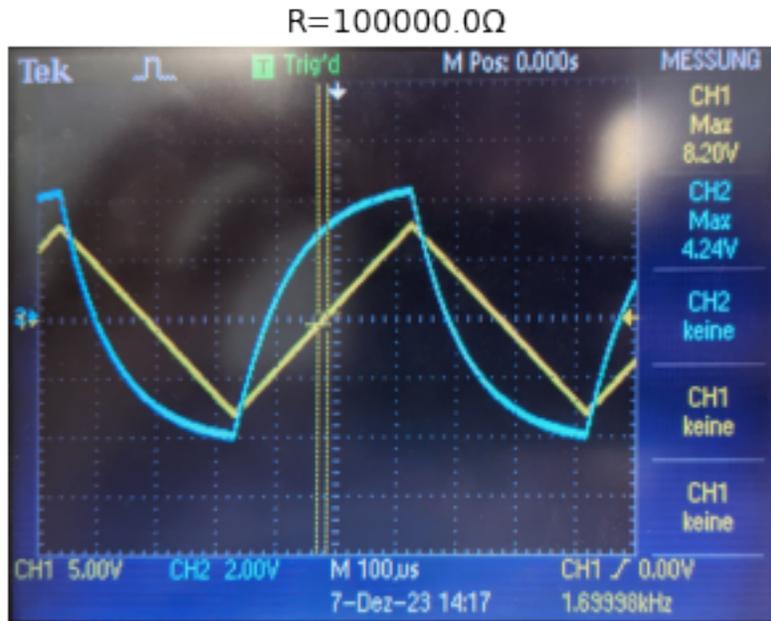
In diesem Versuch wird das differenzierte Verhalten des Hochpassfilters untersucht. Wir haben eine Dreiecksfrequenz von 8V an den Hochpassfilter angelegt zu sehen auf Channel1 in gelb.



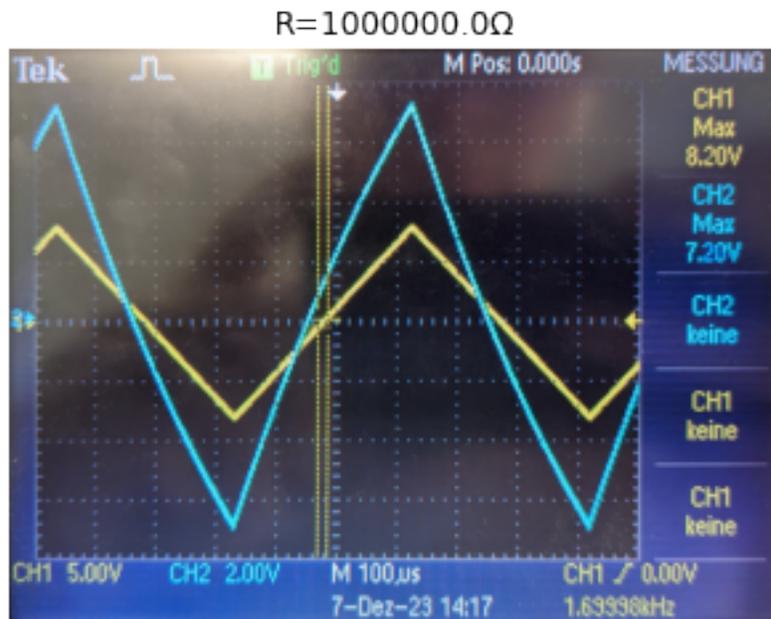
Wie zu erwarten war, verhält sich der Hochpass wie ein Differenzglied für den kleinsten Widerstand, da hier $\omega \ll \omega_0$. Das Oszilloskop zeigt also eine Rechteckwelle am Ausgang wobei ein hoher Anteil an rauschen zu erkennen ist, da einerseits ein Differenzglied immer das Rauschen auf dem Eingang verstärkt und andererseits die Signalspannung sehr stark geschwächt ist (hier nur noch 65.6mV am Ausgang) was das Signal-to-Noise-Ratio weiter verschlechtert.



Auch für den zweitkleinsten Widerstand erzeugt der Hochpass noch eine Rechteckwelle am Ausgang wobei führenden Ecken abgerundet sind, weil der größere Widerstand dafür sorgt, dass bei gleicher Eingangsspannung weniger Strom fließt wodurch der Kondensator langsamer geladen wird.



Mit weiter steigendem Widerstand werden die führenden Ecken werden weiter abgerundet.



Im Grenzfall $R \rightarrow \infty$ nähert sich das Ausgangssignal einer Dreieckswelle an ohne einen differenzierenden Effekt vorzuweisen. Hier ist jetzt $\omega \gg \omega_0$

2.0.3 Aufgabe 1.3: Tiefpass-Filter

Gehen Sie zur Charakterisierung des Tiefpass-Filters analog zu Aufgabe 1.1 vor.

```
[4]: C = 1E-9
f = 1.7E3
    = 2*np.pi*f
U_e = 8.0

%m Bei der Aufgabe geht es darum, die Übertragungsfunktion eines
↳Tiefpassfilters zu untersuchen. Wir haben eine Eingangsspannung von [!U_e!]V
↳an den Tiefpassfilter angelegt. Dabei haben wir bei verschiedenen Frequenzen
↳folgende Daten gemessen:

a13 = """
R,DeltaT,U_a
1E3,-780E-9,8.4
1E4,-8.8E-6,8.2
1E5,-71E-6,5.6
1E6,-126E-6,1.2
"""

mpg = pd.read_csv(StringIO(a13))
mpg.rename(columns={"R": 'Widerstand in Ohm', "DeltaT": "\u0394t in s", "U_a":
↳"Ausgangsspannung in V"}, inplace=True)
display(mpg)

%m Der verwendete Kondensator hat eine Kapazität von 1nF und es wurde bei einer
↳Frequenz von 1.7KHz gemessen.

R_theo = np.logspace(3,6,500)
_Theo = 1/np.sqrt(1 + **2*C**2*R_theo**2)
Delta_theo = -np.arctan(*C*R_theo)
_0_theo = 1 / (R_theo * C)
log_theo = np.log10( / _0_theo)

R, Deltat, U_a = np.genfromtxt(StringIO(a13), delimiter=',')[1:].transpose()

    = U_a / U_e
Delta = 2*pi*Deltat*f
_0 = 1 / (R * C)
```

```

log = np.log10( / _0)

%m Es gilt weiterhin  $\omega_0 = \frac{1}{CR}$  womit sich folgende zwei Plots
↪ergeben:

plt.title("Spannungsübertrag")
plt.plot(log_theo, _theo, "--", label="$\mu_{Theo}$")
plt.plot(log , , "X", label="$\mu$")
plt.legend()
plt.ylabel(r"Spannungsverhältnis  $\left|\frac{U_1}{U_0}\right|$ ")
plt.xlabel(r"$\log(\omega/\omega_{0})$")
plt.show()

plt.title("Phasendifferenz")
plt.plot(log_theo, Delta_theo, "--", label="$\Delta_{Theo}$")
plt.plot(log , Delta , "X", label="$\Delta$")
plt.ylabel(r"Phasendifferenz  $\arg\left(\frac{U_1}{U_0}\right)$ ")
plt.xlabel(r"$\log(\omega/\omega_{0})$")
plt.legend()
plt.show()

%m Wobei nach der Versuchsdokumentation die theoretischen Erwartungswerte  $\mu$ 
↪=  $\frac{1}{\sqrt{\omega^2 C^2 R^2 + 1}}$  sowie  $\Delta\varphi =$ 
↪ $-\arctan\left(\frac{\omega C R}{1}\right)$  als gestrichelte Linien aufgetragen sind.
↪Es ist zu sehen, dass die Messwerte recht nah an den theoretischen Werten
↪liegen.

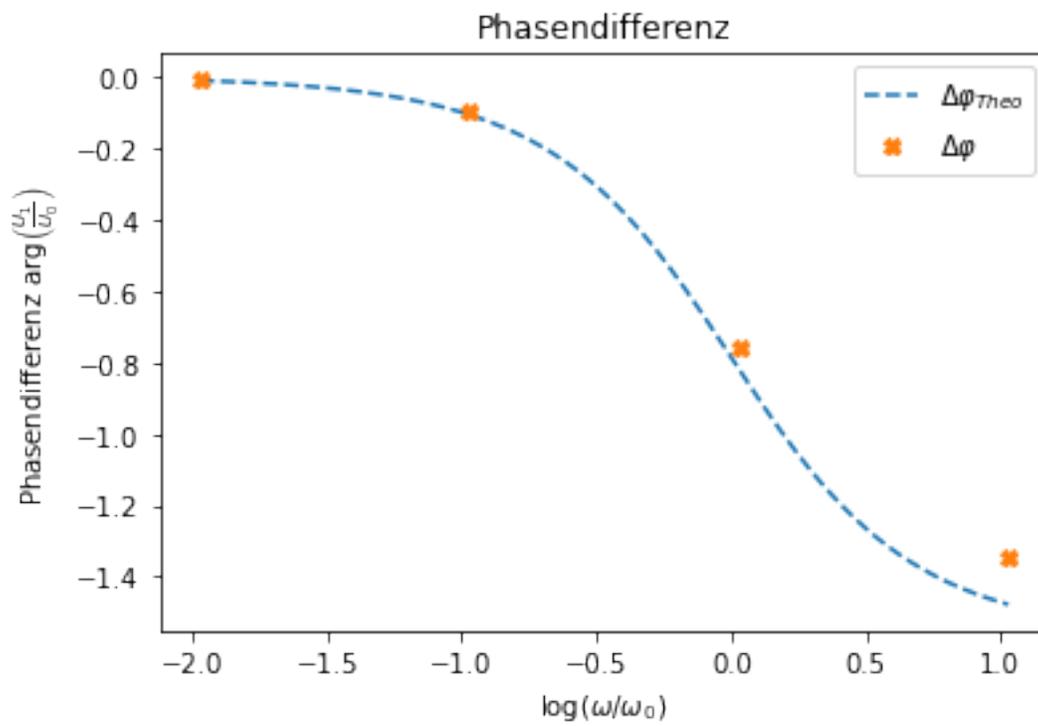
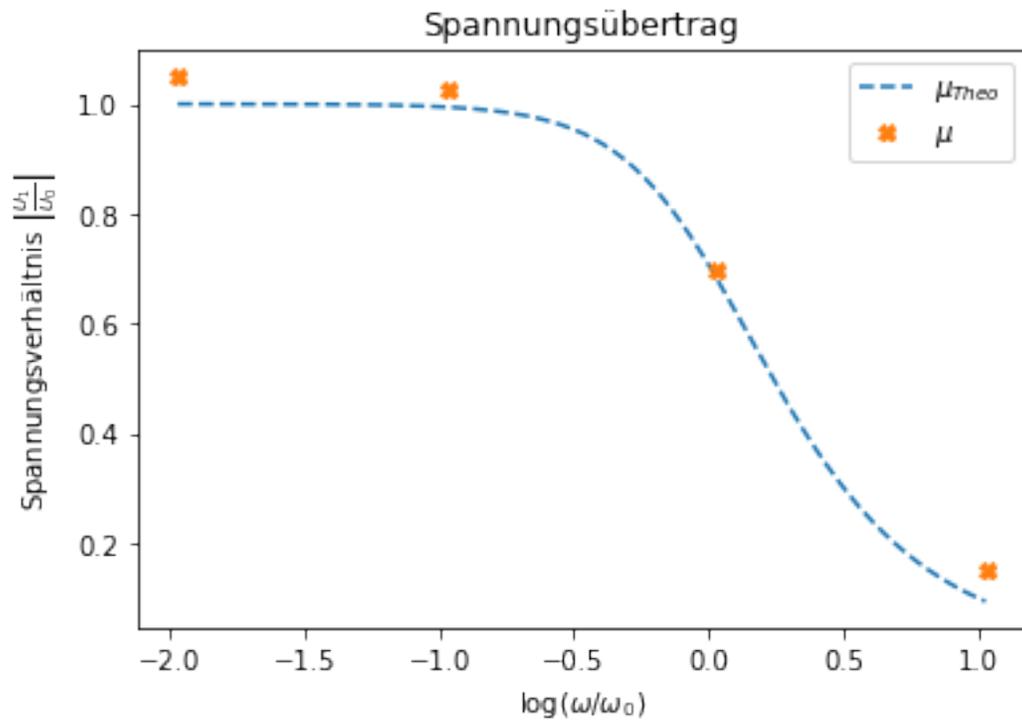
```

Bei der Aufgabe geht es darum, die Übertragungsfunktion eines Tiefpassfilters zu untersuchen. Wir haben eine Eingangsspannung von 8.0V an den Tiefpassfilter angelegt. Dabei haben wir bei verschiedenen Frequenzen folgende Daten gemessen:

	Widerstand in Ohm	Δt in s	Ausgangsspannung in V
0	1000.0	-7.800000e-07	8.4
1	10000.0	-8.800000e-06	8.2
2	100000.0	-7.100000e-05	5.6
3	1000000.0	-1.260000e-04	1.2

Der verwendete Kondensator hat eine Kapazität von 1nF und es wurde bei einer Frequenz von 1.7KHz gemessen.

Es gilt weiterhin $\omega_0 = \frac{1}{CR}$ womit sich folgende zwei Plots ergeben:



Wobei nach der Versuchsdokumentation die theoretischen Erwartungswerte $\mu = \frac{1}{\sqrt{\omega^2 C^2 R^2 + 1}}$ sowie $\Delta\varphi = -\arctan(\omega C R)$ als gestrichelte Linien aufgetragen sind. Es ist zu sehen, dass die Messwerte recht nah an den theoretischen Werten liegen.

2.0.4 Aufgabe 1.4: Integrierglied

Gehen Sie zur weiteren Untersuchung des Tiefpass-Filters analog zu Aufgabe 1.2 vor, aber verwenden Sie statt einer sinus- (oder dreieck-) eine **rechteckförmige** Wechselspannung. Für $\omega \gg \omega_0$ wird die Schaltung zum Integrierglied. Beschreiben Sie was Sie sehen und vergleichen Sie Ihre Beobachtung mit Ihrer Erwartung.

[5]: U_e = 8

```
%m In diesem Versuch wird das integrierende Verhalten des Tiefpassfilters
↳ untersucht. Wir haben eine Dreiecksfrequenz von 8V an den Tiefpassfilter
↳ angelegt zu sehen auf Channel1 in gelb. Da wir leider vergessen haben eine
↳ Rechteckwelle anstatt einer Dreieckswelle am Eingang anzulegen sind die
↳ Ergebnisse nicht ganz so schön wie erhofft aber nichtdestotrotz ist der
↳ Effekt sichtbar.

plt.title(f"R={R[3]}Ω")
plt.ing("Fotos/A14_4.jpg")
%m Wie zu erwarten war, verhält sich der Tiefpass wie ein Integrierglied für
↳ den größten Widerstand, da hier  $\omega \gg \omega_0$ . Das Oszilloskop zeigt
↳ hier jeweils einen Parabelförmigen verlauf am Ausgang, wie beim Integral
↳ einer geraden zu erwarten ist.

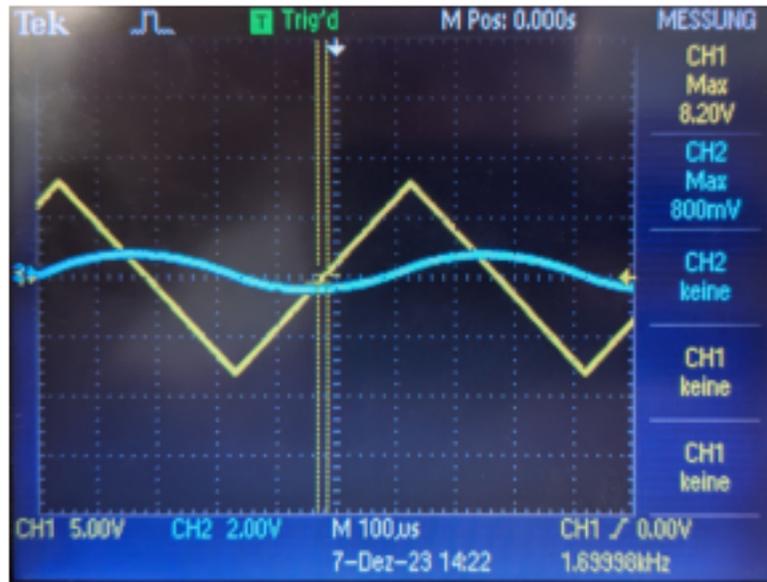
plt.title(f"R={R[2]}Ω")
plt.ing("Fotos/A14_3.jpg")
%m Auch für den zweitkleinsten Widerstand erzeugt der Tiefpass noch eine leicht
↳ parabelförmigen Verlauf wobei die Parabeln verzerrt sind.

plt.title(f"R={R[1]}Ω")
plt.ing("Fotos/A14_2.jpg")
%m Mit weiter reduziertem Widerstand werden die Kanten weiter verzerrt und die
↳ Kurve fängt an sich der Eingangsspannung anzunähern.

plt.title(f"R={R[0]}Ω")
plt.ing("Fotos/A14_1.jpg")
%m Im Grenzwert  $R \rightarrow 0$  nähert sich das Ausgangssignal der Eingangswelle an,
↳ ohne einen integrierenden Effekt. Hier ist jetzt  $\omega \ll \omega_0$ .
```

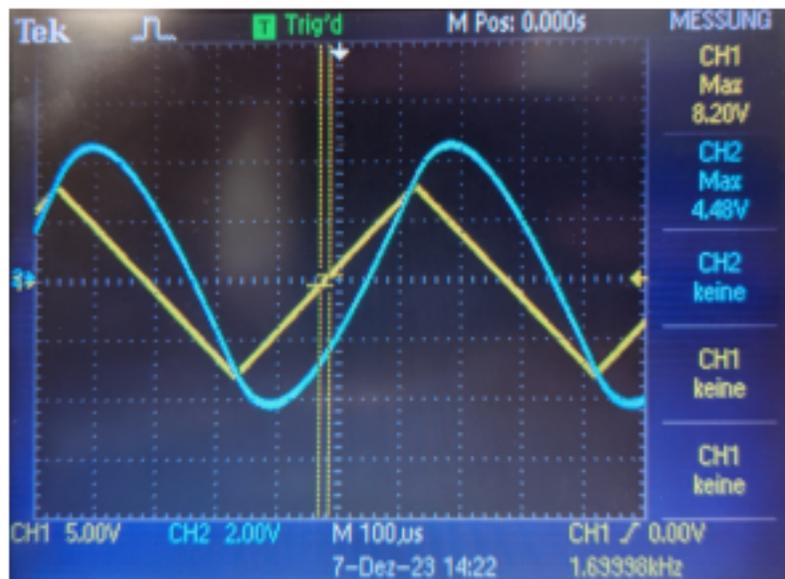
In diesem Versuch wird das integrierende Verhalten des Tiefpassfilters untersucht. Wir haben eine Dreiecksfrequenz von 8V an den Tiefpassfilter angelegt zu sehen auf Channel1 in gelb. Da wir leider vergessen haben eine Rechteckwelle anstatt einer Dreieckswelle am Eingang anzulegen sind die Ergebnisse nicht ganz so schön wie erhofft aber nichtdestotrotz ist der Effekt sichtbar.

R=1000000.0Ω



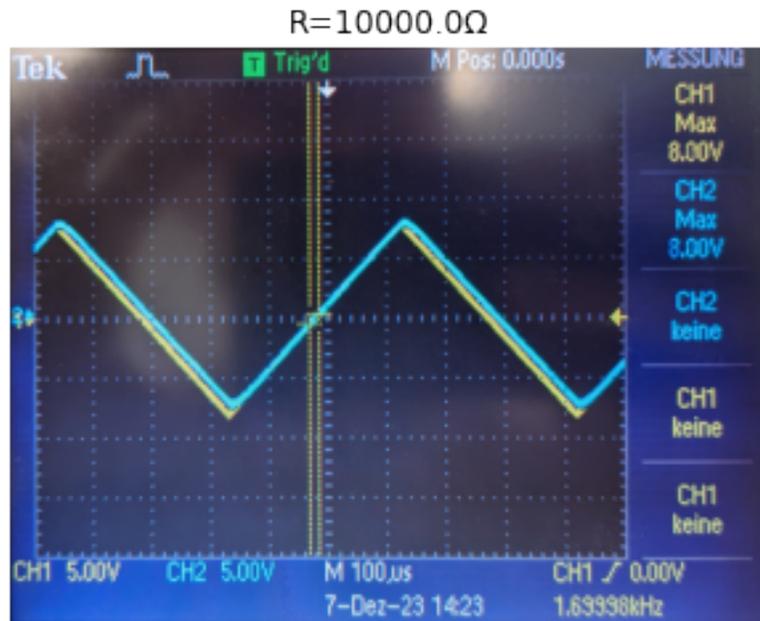
Wie zu erwarten war, verhält sich der Tiefpass wie ein Integrierglied für den größten Widerstand, da hier $\omega \gg \omega_0$. Das Oszilloskop zeigt hier jeweils einen Parabelförmigen Verlauf am Ausgang, wie beim Integral einer geraden zu erwarten ist.

R=100000.0Ω

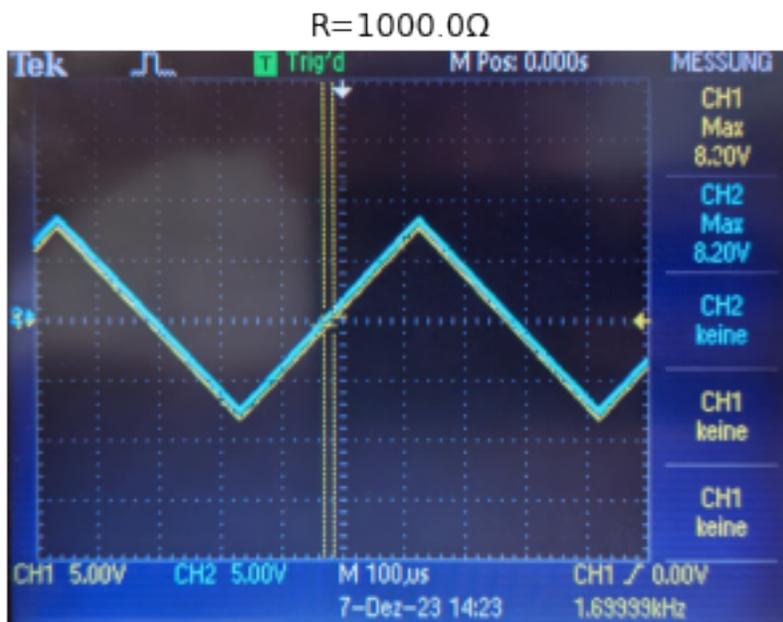


Auch für den zweitkleinsten Widerstand erzeugt der Tiefpass noch eine leicht parabelförmigen

Verlauf wobei die Parabeln verzerrt sind.



Mit weiter reduziertem Widerstand werden die Kanten weiter verzerrt und die Kurve fängt an sich der Eingangsspannung anzunähern.



Im Grenzwert $R \rightarrow 0$ nähert sich das Ausgangssignal der Eingangswelle an, ohne einen integrieren-

den Effekt. Hier ist jetzt $\omega \ll \omega_0$.

2.1 Aufgabe 2: Drosselkette

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-2.md](#).

2.1.1 Aufgabe 2.1: Bestimmung der charakteristischen Impedanz Z_0

Bestimmen Sie Z_0 für $\omega \ll \omega_0$.

```
[6]: #f = 20e3
#U_0 = 6
uR = ufloat(225, 10)

C = 2e-9
L = 96e-6
Z_A_theo = np.sqrt(L/C)

%m In diesem Versuch soll die charakteristische Impedanz  $Z_0$  der Drosselkette
↪ bestimmt werden. Der theoretische Erwartungswert errechnet sich aus der
↪ angegebenen Kapazität von 2nF und Induktivität von  $96\mu\text{H}$  der
↪ Drosselkette zu  $Z_{0\text{Theo}} = \sqrt{\frac{L}{C}} = [!Z_A_{\text{theo}}:.2f!]\backslash, \backslash\Omega$ . Wir
↪ haben bei einer Einstellung des Potentiometers am Ausgang der Drosselkette
↪ von  $[!uR.n:.2f!]\backslash, \backslash\Omega$  die geringsten Reflektionen erhalten. Zur
↪ Einstellung haben wir eine Rechteckwelle verwendet, da man hier besonders
↪ gut die Veränderungen durch eine Reflektion erkennen kann. Da das
↪ Potentiometer relativ viel Spiel hatte haben wir nur eine Unsicherheit von
↪  $[!uR.n:.2f!]\backslash, \backslash\Omega$ 
```

In diesem Versuch soll die charakteristische Impedanz Z_0 der Drosselkette bestimmt werden. Der theoretische Erwartungswert errechnet sich aus der angegebenen Kapazität von 2nF und Induktivität von $96\mu\text{H}$ der Drosselkette zu $Z_{0\text{Theo}} = \sqrt{\frac{L}{C}} = 219.09\Omega$. Wir haben bei einer Einstellung des Potentiometers am Ausgang der Drosselkette von 225.00Ω die geringsten Reflektionen erhalten. Zur Einstellung haben wir eine Rechteckwelle verwendet, da man hier besonders gut die Veränderungen durch eine Reflektion erkennen kann. Da das Potentiometer relativ viel Spiel hatte haben wir nur eine Unsicherheit von 225.00Ω

2.1.2 Aufgabe 2.2: Bestimmung der Grenzfrequenz ω_0

Bestimmen Sie ω_0 . Um diese Messreihe zügig aufnehmen zu können sollten Sie eine klare numerische Erwartung zur Frequenzabhängigkeit von $Z_0(\omega)$ haben.

```
[7]: %m In dieser Aufgabe soll die charakteristische Frequenz der Drosselkette
↪ bestimmt werden, indem die Ausgangsspannung  $U_5$  bei verschiedenen
↪ Eingangsfrequenzen  $\omega_0$  gemessen wird.
```

`%m` Um den regelbaren Widerstand entsprechen einstellen zu können muss vorher
 ↳ der Zusammenhang $Z_0(\omega)$ bestimmt werden. Hier werden die im
 ↳ Datenblatt angegebenen Werte für C und L verwendet um über den Zusammenhang
 ↳ $Z_0 = \sqrt{\frac{L}{C} \left(1 - \frac{\omega^2}{\omega_0^2}\right)}$ mit $\omega_0 =$
 ↳ \sqrt{LC} die nötige Ausgangsimpedanz $Z_0 = Z_A$ zu bestimmen.

```
C_theo = 2e-9
L_theo = 96e-6
```

```
#w = np.logspace(6.5, 6.7, 1000)
w = np.linspace(4.4e6, 4.7e6, 1000)
```

```
w_0 = 2/np.sqrt(L_theo*C_theo)
Z_0 = np.sqrt(L_theo/(C_theo*(1-w**2/w_0**2))+0j)
```

```
f = w/2/np.pi
```

```
plt.title("Theoretische Impedanz  $Z_0$ ")
plt.plot(f, np.real(Z_0), "--", label="Realteil")
plt.plot(f, np.imag(Z_0), "--", label="Imaginärteil")
plt.axvline(1/np.pi/np.sqrt(L_theo*C_theo), linestyle="--", color="k",
  ↳ label=" $w_0$ ")
plt.ylim((-100, 2000))
plt.ylabel("Complexer Widerstand in  $\Omega$ ")
plt.xlabel("Frequenz in Hz")
plt.legend()
plt.show()
```

```
n = 6
Z_L = 1j*w*L_theo
Z_C = 2/(1j*w*C_theo)
  = np.arccosh(Z_L/Z_C+1)
A = np.cosh(n*)-np.sinh(n*)
```

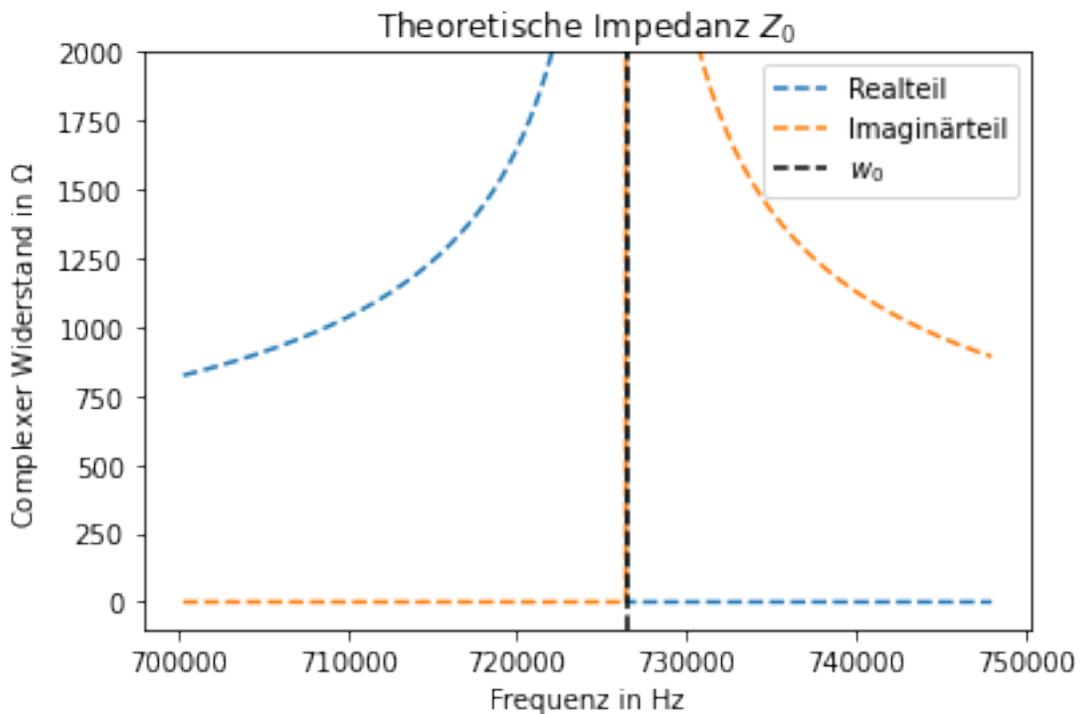
```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(f, np.abs(A), label=" $U_5/U_0$ ")
ax.axvline(1/np.pi/np.sqrt(L_theo*C_theo), linestyle="--", color="k",
  ↳ label=" $w_0$ ")
ax2 = plt.gca().twinx()
ax2.plot(f, np.angle(A), label="Phase", color="green")
ax.legend(loc="upper left")
ax2.legend(loc=0)
ax.set_ylabel(r"Verstärkung  $\left|\frac{U_5}{U_0}\right|$ ")
ax2.set_ylabel(r"Phasendifferenz  $\arg\left(\frac{U_5}{U_0}\right)$ ")
ax.set_xlabel("Frequenz in Hz")
```

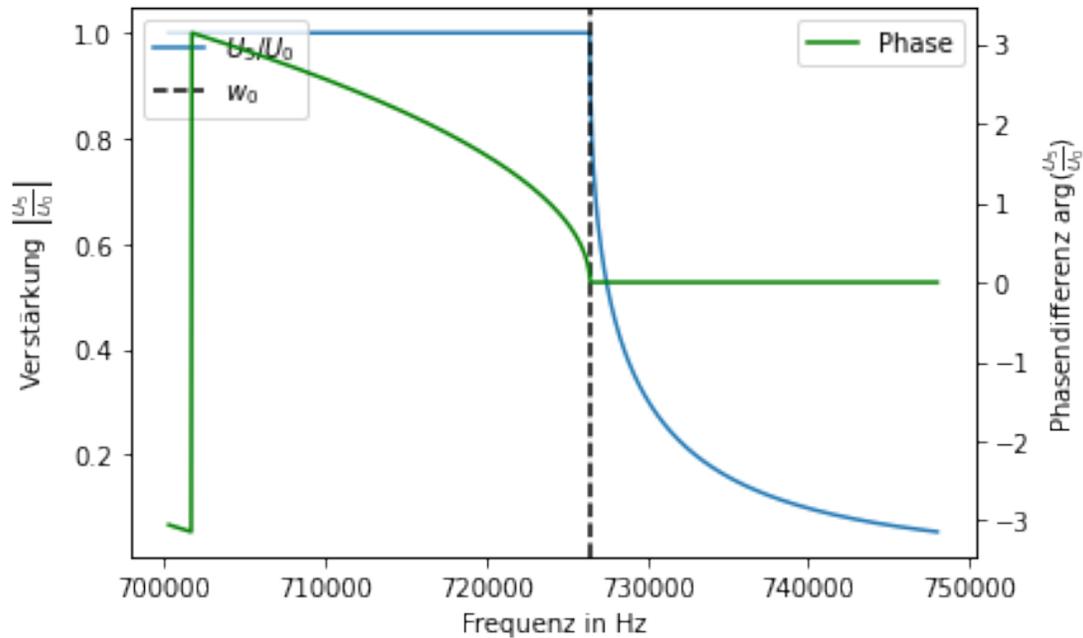
```
plt.show()
```

%m Es ist zu sehen, dass der Widerstand divergiert und auch schon für kleine
↪ Frequenzen weit oberhalb des Regelbereiches des regelbaren Widerstandes
↪ liegt. Es ist also mit dem Versuchsaufbau nicht möglich den Widerstand
↪ sinnvoll nachzuregeln. Deswegen haben wir einen konstanten, ausreichend
↪ großen Widerstand von $1.5\text{K}\Omega$ gewählt um Reflektionen in der Nähe der
↪ Grenzfrequenz möglichst zu vermeiden, da nur in diesem Bereich eine genaue
↪ Messung wichtig ist.

In dieser Aufgabe soll die charakteristische Frequenz der Drosselkette bestimmt werden, indem die Ausgangsspannung U_5 bei verschiedenen Eingangsfrequenzen ω_0 gemessen wird.

Um den regelbaren Widerstand entsprechen einstellen zu können muss vorher der Zusammenhang $Z_0(\omega)$ bestimmt werden. Hier werden die im Datenblatt angegebenen Werte für C und L verwendet um über den Zusammenhang $Z_0 = \sqrt{\frac{L}{C(1-\frac{\omega^2}{\omega_0^2})}}$ mit $\omega_0 = \sqrt{LC}^{-1}$ die nötige Ausgangsimpedanz $Z_0 = Z_A$ zu bestimmen.





Es ist zu sehen, dass der Widerstand divergiert und auch schon für kleine Frequenzen weit oberhalb des Regelbereiches des regelbaren Widerstandes liegt. Es ist also mit dem Versuchsaufbau nicht möglich den Widerstand sinnvoll nachzuregeln. Deswegen haben wir einen konstanten, ausreichend großen Widerstand von $1.5\text{K}\Omega$ gewählt um Reflektionen in der Nähe der Grenzfrequenz möglichst zu vermeiden, da nur in diesem Bereich eine genaue Messung wichtig ist.

[8]: %m Wir haben uns davon überzeugt, dass die Eingangsspannung ausreichend
 ↳ unabhängig von der eingestellten Frequenz ist, indem wir einmal verschiedene
 ↳ Frequenzen eingestellt haben und festgestellt haben, dass sich die
 ↳ Eingangsspannung nicht sichtbar verändert.

```
U_0 = ufloat(6.08, 0.1)
```

%m Wir haben folgende Messwerte bei der Messung der Drosselkette erhalten:

```
a22 = ""
f,U_6
720E3,2.78
725E3,3.1
730E3,3.58
735E3,3.34
740E3,2.42
745E3,1.64
750E3,1.09
755E3,777E-3
758.6E3,608E-3
```

```

760E3,546E-3
763.96E3,430E-3
765E3,400E-3
770E3,299E-3
775E3,230E-3
780E3,180E-3
785E3,143E-3
790E3,116E-3
795E3,95E-3
800E3,73.2E-3
""" #keine Änderung mehr ab da, nur noch rauschen

f, U_6 = np.genfromtxt(StringIO(a22), delimiter=',')[1:].transpose()
w = f * (2 * np.pi)
U_6 = np.sqrt(2) * U_6 #effektiv zu anderer, unten bei Tabelee ncho machen

w_theo = np.linspace(min(f)*2*pi,max(f)*2*pi, 1000)
C_theo = 2e-9
L_theo = 96e-6
n=6
Z_L = 1j*w_theo*L_theo
Z_C = 2/(1j*w_theo*C_theo)
    = np.arccosh(Z_L/Z_C+1)
w_theo = np.linspace(np.min(w),np.max(w), 1000)
f_theo = w_theo/2/np.pi
A_theo = np.cosh(n* )-np.sinh(n* )

mpg = pd.read_csv(StringIO(a22))
mpg["U_6"] = np.sqrt(2) * mpg["U_6"]
mpg.rename(columns={"f": 'Frequenz in Hz',"U_6": "Ausgangsspannung in V"},
    inplace=True)
display(mpg)

%m Hier sind nun die Messwerte zusammen mit den theoretischen Werten
    aufgetragen.

w_0_theo = 2/np.sqrt(L_theo*C_theo)
index_theo = np.min(np.where((np.abs(A_theo) <= 0.1) == 1))
w_border_theo = (w_theo[index_theo] + w_theo[index_theo-1])/2

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(f_theo, np.abs(A_theo), label="$U_5/U_0$ (theo)")
uplot(f, U_6/U_0, fmt="x")
ax.axvline(w_0_theo/2/pi, linestyle="--", color="k", label="$w_0 (theo)$")
ax.axvline(w_border_theo/2/pi, linestyle="--", color="g", label="$\omega_{10\%}$
    (theo)$")

```

```

ax.legend(loc="upper left")
ax.set_ylabel(r"Verstärkung  $\left|\frac{U_5}{U_0}\right|$ ")
ax.set_xlabel("Frequenz in Hz")
x = np.linspace(f[0],f[len(f)-1])
ax.plot(x,0.1*np.ones(len(x)))
plt.show()

%m Es ist zu erkennen, dass die Kurvenformen nicht besonders ähnlich sind. Es
↳ spielen also noch andere Effekte eine Rolle, die unser Modell nicht
↳ darstellen kann. Das dürfte daran liegen, dass unser Modell die vorhandene
↳ Widerstände des Aufbaus vernachlässigt. Deswegen kann auch kein sinnvoller
↳ Fit an die Daten gemacht werden.

index = np.min(np.where((U_6/U_0 <= 0.1) == 1))
w_border = (w[index] + w[index-1])/2
Δw_border = abs((w[index] - w[index-1])/2) # Hier haben wir eine Unsicherheit
↳ von 100Hz
uw_boder = ufloat(w_border, Δw_border)

uw_boder = ufloat(763.96e3*2*pi, 100*2*pi)

index_theo = np.min(np.where((np.abs(A_theo) <= 0.1) == 1))
w_border_theo = (w_theo[index_theo] + w_theo[index_theo-1])/2

%m Wie in der Aufgabenstellung vorgegeben wird jetzt  $\omega_0$  so gewählt,
↳ dass das Modell bei  $\frac{U_6}{U_0} = 0.1$  auf die Daten passt. Dieses
↳ Verfahren finde ich persönlich nicht besonders sinnvoll, da hier Effekte eine
↳ großen Einfluss haben, die unser Modell nicht darstellen kann, was dafür
↳ sorgt, dass unsere Ergebnisse wenig mit der Realität zutun haben und einen
↳ großen unbekanntem systematischen Fehler aufweisen werden. Für die
↳ Aufgabenstellung führen wir die Auswertung hier trotzdem einmal vor.

%m Wir haben einmal einen groben Kurvenverlauf von 720KHz bis 800KHz
↳ aufgetragen um einen Überblick zu gewinnen. Danach haben wir die Frequenz so
↳ reguliert, das das Oszilloskop am Ausgang gerade 10% der Eingangsspannung,
↳ also 608mV angezeigt hat. Diese Einstellung konnten wir bis auf 100Hz genau
↳ machen. Es ergibt sich eine Grenzkreisfrequenz  $\omega_{10\%} = [!uw\_boder.n:$ 
↳  $.2f!)\, \mathrm{Hz}$  mit einer Unsicherheit von  $[!uw\_boder.s:.2f!$ 
↳  $)\, \mathrm{Hz}$ $.

w_0 = uw_boder/w_border_theo*w_0_theo

```

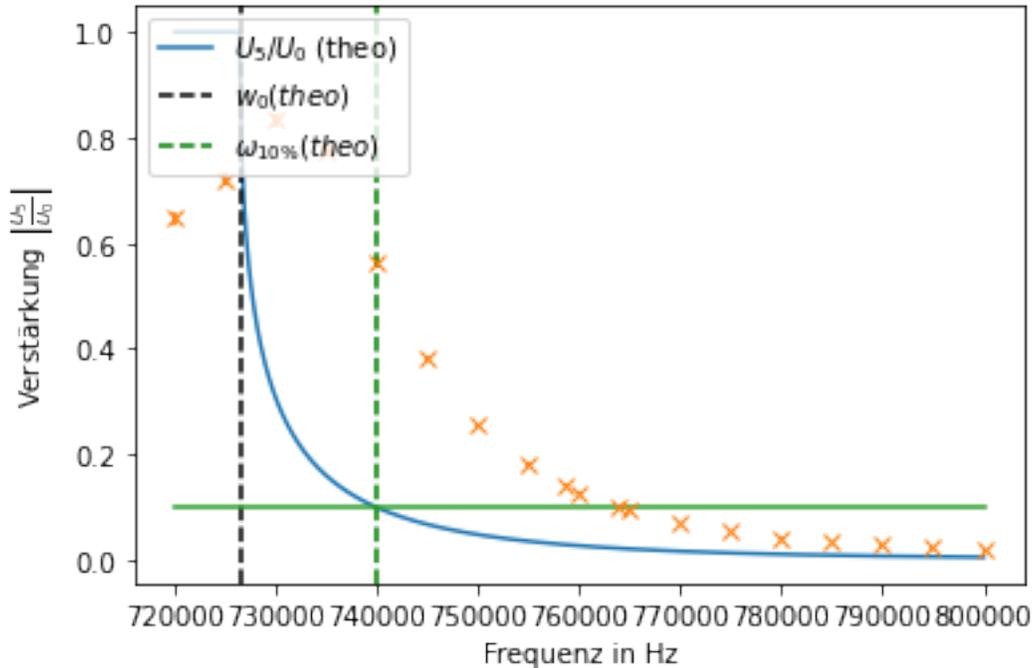
%m In der Kurvenverlauf der theoretischen Funktion gilt
 ↳ $\frac{\omega_{10\%}}{\omega_0} = \frac{w_{\text{border_theo}}}{w_{0_theo}}$ Da die
 ↳ theoretischen Werte nah genug an den echten Messwerten liegen kann in guter
 ↳ Näherung angenommen, dass dieses Verhältnis auch für die Messwerte gilt.
 ↳ Damit ergibt sich $w_0 = \pm 0.2 \text{ Hz}$ $\pm 0.2 \text{ Hz}$. Dieser Wert
 ↳ ist aufgrund der oben genannten Probleme mit dem Modell natürlich nicht mit
 ↳ dem theoretischen Wert vereinbar.

Wir haben uns davon überzeugt, dass die Eingangsspannung ausreichend unabhängig von der eingestellten Frequenz ist, indem wir einmal verschiedene Frequenzen eingestellt haben und festgestellt haben, dass sich die Eingangsspannung nicht sichtbar verändert.

Wir haben folgende Messwerte bei der Messung der Drosselkette erhalten:

	Frequenz in Hz	Ausgangsspannung in V
0	720000.0	3.931514
1	725000.0	4.384062
2	730000.0	5.062885
3	735000.0	4.723473
4	740000.0	3.422397
5	745000.0	2.319310
6	750000.0	1.541493
7	755000.0	1.098844
8	758600.0	0.859842
9	760000.0	0.772161
10	763960.0	0.608112
11	765000.0	0.565685
12	770000.0	0.422850
13	775000.0	0.325269
14	780000.0	0.254558
15	785000.0	0.202233
16	790000.0	0.164049
17	795000.0	0.134350
18	800000.0	0.103520

Hier sind nun die Messwerte zusammen mit den theoretischen Werten aufgetragen.



Es ist zu erkennen, dass die Kurvenformen nicht besonders ähnlich sind. Es spielen also noch andere Effekte eine Rolle, die unser Modell nicht darstellen kann. Das dürfte daran liegen, dass unser Modell die vorhandene Widerstände des Aufbaus vernachlässigt. Deswegen kann auch kein sinnvoller Fit an die Daten gemacht werden.

Wie in der Aufgabenstellung vorgegeben wird jetzt ω_0 so gewählt, dass das Modell bei $\frac{U_6}{U_0} = 0.1$ auf die Daten passt. Dieses Verfahren finde ich persönlich nicht besonders sinnvoll, da hier Effekte eine großen Einfluss haben, die unser Modell nicht darstellen kann, was dafür sorgt, dass unsere Ergebnisse wenig mit der Realität zutun haben und einen großen unbekanntem systematischen Fehler aufweisen werden. Für die Aufgabenstellung führen wir die Auswertung hier trotzdem einmal vor.

Wir haben einmal einen groben Kurvenverlauf von 720KHz bis 800KHz aufgetragen um einen Überblick zu gewinnen. Danach haben wir die Frequenz so reguliert, das das Oszilloskop am Ausgang gerade 10% der Eingangsspannung, also 608mV angezeigt hat. Diese Einstellung konnten wir bis auf 100Hz genau machen. Es ergibt sich eine Grenzkreisfrequenz $\omega_{10\%} = 4800102.25$ Hz mit einer Unsicherheit von 628.32 Hz.

In der Kurvenverlauf der theoretischen Funktion gilt $\frac{\omega_{10\%}}{\omega_0} = 1.018418896535945$ Da die theoretischen Werte nah genug an den echten Messwerten liegen kann in guter Näherung angenommen, dass dieses Verhältnis auch für die Messwerte gilt. Damit ergibt sich $\omega_0 = 4713288.67\text{Hz} \pm 616.95\text{Hz}$. Dieser Wert ist aufgrund der oben genannten Probleme mit dem Modell natürlich nicht mit dem theoretischen Wert vereinbar.

2.1.3 Aufgabe 2.3: Bestimmung von C und L der in der Drosselkette verbauten Elemente

Aus Ihren Messergebnissen aus den Aufgaben 2.1 und 2.2 können Sie die Kapazität C und die Induktivität L der in der Drosselkette verbauten Elemente bestimmen. Pflanzen Sie die Unsicherheiten Ihrer Messungen fort und vergleichen Sie Ihre Ergebnisse mit den technischen Daten zu diesem Versuch (siehe [Datenblatt.md](#)).

```
[9]: Z_0 = ufloat(250,5)
f_0 = ufloat(720e3,1e3)

%m Wir haben für die Impedanz $Z_0$ einen Wert von $[!Z_0.n:.2f!]\,\,\Omega\pm[!
↪Z_0.s:.2f!]\,\,\Omega$ bestimmt und für die Frequenz $f_0$ einen Wert von $[!
↪f_0.n*1e-3:.2f!]\,\,kHz\pm[!f_0.s*1e-3:.2f!]\,\,kHz$.

C = 1/(np.pi*f_0*Z_0)
L = Z_0 / (np.pi*f_0)

%m Daraus ergeben sich mit Gausscher Unsicherheitsfortpflanzung $L=[!L.n*1e6:.
↪2f!]\,\,\mu H\pm[!L.s*1e6:.2f!]\,\,\mu H$ und $C=[!C.n*1e9:.2f!]\,\,nF\pm[!
↪.2f!]\,\,nF$.

%m Unter anbeacht der Probleme mit dem Modell liegen diese Werte noch recht
↪nah an den Werten im Datenblatt mit Abweichungen unter 15%. Allerdings sind
↪die Werte wie zu erwarten war nicht im Rahmen der statistischen
↪Unsicherheiten mit den Datenblattwerten vereinbar, da die systematischen
↪Unsicherheiten nicht bestimmt werden konnten.
```

Wir haben für die Impedanz Z_0 einen Wert von $250.00\,\Omega \pm 5.00\,\Omega$ bestimmt und für die Frequenz f_0 einen Wert von $720.00\,kHz \pm 1.00\,kHz$.

Daraus ergeben sich mit Gausscher Unsicherheitsfortpflanzung $L = 110.52\,\mu H \pm 2.22\,\mu H$ und $C = 1.77\,nF \pm 0.04\,nF$.

Unter anbeacht der Probleme mit dem Modell liegen diese Werte noch recht nah an den Werten im Datenblatt mit Abweichungen unter 15%. Allerdings sind die Werte wie zu erwarten war nicht im Rahmen der statistischen Unsicherheiten mit den Datenblattwerten vereinbar, da die systematischen Unsicherheiten nicht bestimmt werden konnten.

2.1.4 Aufgabe 2.4: Bestimmung der Phasendifferenz $\Delta\varphi$

- Stellen Sie den Verlauf von $\Delta\varphi(\omega)$ für mindestens 6 bis 8 Werte von ω dar. Führen Sie die gleiche Messreihe sowohl für die gesamte Drosselkette, als auch für ein einzelnes π -Glied aus der Drosselkette durch und überzeugen Sie sich von den folgenden Eigenschaften:
 - Für ein einzelnes π -Glied gilt $\Delta\varphi(\omega_0) = \pi$.
 - Bei vorgegebener Frequenz ist $\Delta\varphi(\omega)$ für die sechsgliedrige Drosselkette gegenüber des einzelnen π -Glieds 6-mal größer.

- Bestimmen Sie, in einer weiteren Messreihe, $\omega(\Delta\varphi)$ für fünf vorgegebene Werte von $\Delta\varphi = 1\pi, 2\pi, 3\pi, 4\pi, 5\pi$. Bestimmen Sie hieraus einen weiteren Messwert für ω_0 . Vergleichen Sie den so ermittelten Wert mit Ihrem Ergebnis aus Aufgabe 2.2.

```
[10]: a24 = """
f,DeltaT_1,DeltaT_6
10e3,3800e-9,104e-9
100e3,2700e-9,2.9e-6
300e3,2700e-9,3e-6
500e3,2800e-9,3e-6
600e3,2960e-9,3.12e-6
650e3,3240e-9,3.16e-6
690e3,3320e-9,3.44e-6
"""

f, DeltaT_1, DeltaT_6, = np.genfromtxt(StringIO(a24), delimiter=',')[1:].
    ↪transpose()
w = f * (2 * np.pi)

_1 = f*DeltaT_1*2*pi
_6 = f*DeltaT_6*2*pi

%m In diesem Versuch sollte die Phasendifferenz eines  $\pi$ -Glieds und der
    ↪Drosselkette untersucht werden. In der folgenden Tabelle sind unsere
    ↪Messwerte für verschiedene Frequenzen zusammen mit den daraus bestimmte
    ↪Phasenverschiebungen und dem Verhältnis der Phasenverschiebungen zwischen
    ↪dem einzelnen  $\pi$ -Glied und der Drosselkette.

mpg = pd.read_csv(StringIO(a24))
mpg.rename(columns={"f": 'Frequenz in Hz', "DeltaT_1": "\u0394t_1 in s", "DeltaT_6":
    ↪"\u0394t_6 in s"}, inplace=True)
mpg["_1"] = _1
mpg["_6"] = _6
mpg["_6/_1"] = _6/_1
display(mpg)

%m Leider können wir in den gemessenen Daten keinen der gesuchten Effekte
    ↪wiederfinden. Das Verhältnis der Phasenverschiebung sollte der Theorie nach
    ↪jeweils ungefähr 6 sein, ist aber in der Praxis für alle Frequenzen nahe 1.
    ↪Und auch die Phase nahe  $\omega_0$  ist weit entfernt von  $\pi$ . Es könnte
    ↪sein, dass etwas mit unserem Versuchsaufbau nicht gestimmt hat aber das ist
    ↪im Nachhinein schwer nach zu vollziehen. Es wurde auch explizit darauf
    ↪geachtet, dass keine Verschiebung um  $2\pi$  in der Phase übersehen wurde.
```

In diesem Versuch sollte die Phasendifferenz eines π -Glieds und der Drosselkette untersucht werden. In der folgenden Tabelle sind unsere Messwerte für verschiedene Frequenzen zusammen mit den

daraus bestimmte Phasenverschiebungen und dem Verhältnis der Phasenverschiebungen zwischen dem einzelnen π -Glieder und der Drosselkette.

	Frequenz in Hz	Δt_{-1} in s	Δt_{-6} in s	$_{-1}$	$_{-6}$	$_{-6}/_{-1}$
0	10000.0	0.000004	1.040000e-07	0.238761	0.006535	0.027368
1	100000.0	0.000003	2.900000e-06	1.696460	1.822124	1.074074
2	300000.0	0.000003	3.000000e-06	5.089380	5.654867	1.111111
3	500000.0	0.000003	3.000000e-06	8.796459	9.424778	1.071429
4	600000.0	0.000003	3.120000e-06	11.158937	11.762123	1.054054
5	650000.0	0.000003	3.160000e-06	13.232388	12.905663	0.975309
6	690000.0	0.000003	3.440000e-06	14.393521	14.913769	1.036145

Leider können wir in den gemessenen Daten keinen der gesuchten Effekte wiederfinden. Das Verhältnis der Phasenverschiebung sollte der Theorie nach jeweils ungefähr 6 sein, ist aber in der Praxis für alle Frequenzen nahe 1. Und auch die Phase nahe ω_0 ist weit entfernt von π . Es könnte sein, dass etwas mit unserem Versuchsaufbau nicht gestimmt hat aber das ist im Nachhinein schwer nach zu vollziehen. Es wurde auch explizit darauf geachtet, dass keine Verschiebung um 2π in der Phase übersehen wurde.

[11]: `%m` In dieser Aufgabe soll nun über die Phasenverschiebung mit Lissajous-Figuren `↪` die Grenzfrequenz bestimmt werden. Wir haben folgende Messwerte erhalten:

```

a24_2 = """
n,f6
1,195e3
2,365e3
3,519e3
4,634e3
5,714.7e3
"""

n, f6, = np.genfromtxt(StringIO(a24_2), delimiter=',')[1:].transpose()
      = n*pi

mpg = pd.read_csv(StringIO(a24_2))
mpg.rename(columns={"n": 'Phasenverschiebung in rad', "f6": "Frequenz in
↪Hz"}, inplace=True)
display(mpg)

(m,), pcov = scipy.optimize.curve_fit(lambda x, m: m*x, , f6)

"""
plt.plot(, f6, "x")
plt.plot(np.linspace(0,16), np.linspace(0,16)*m)
plt.gca().set_ylim(bottom=0)
plt.show()
"""

```

```

def prop_model(x, m):
    return x*m

xy_data = kafe2.XYContainer(x_data= , y_data=f6)
xy_data.add_error(axis='x', err_val=0.05*pi)
xy_data.add_error(axis='y', err_val=10)
lin_fit = kafe2.Fit(data=xy_data, model_function=prop_model)

lin_fit.do_fit()
p = kafe2.Plot(lin_fit)
#p.x_label = 'Phase in radian'
p.y_label = 'Frequenz in Hz'
p.plot()
p.show()

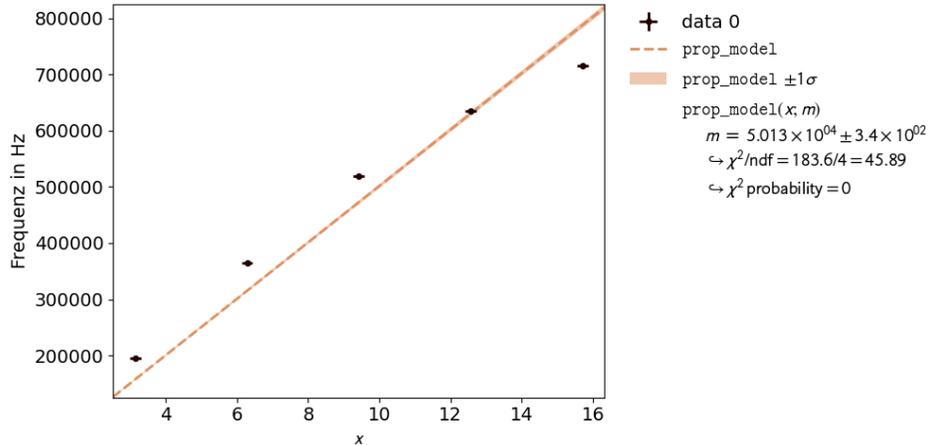
m, = lin_fit.parameter_values
Δm, = lin_fit.parameter_errors
um = ufloat(m, Δm)
w_0 = 6*pi*um

%m Aus dem proportionalen Fit ergibt sich für $_0$ ein Wert von [!w_0.n:.0f!]
↳Hz mit einer Unsicherheit von [!w_0.s:.0f!] Hz. Es wurde ein 5%
↳Ableseungenauigkeit für die Liassau-Figur angenommen, da es nicht so einfach
↳war zu erkennen, wann genau die Geradenform erreicht war und eine 10 Hz
↳Messunsicherheit auf der Frequenz. Es ist außerdem klar zu erkennen, dass
↳die proportionale gerade kein gutes Modell für die Daten ist, da ein
↳sichtbarer Block überhalb der Kurve liegt.

```

In dieser Aufgabe soll nun über die Phasenverschiebung mit Lissajous-Figuren die Grenzfrequenz bestimmt werden. Wir haben folgende Messwerte erhalten:

	Phasenverschiebung in rad	Frequenz in Hz
0	1	195000.0
1	2	365000.0
2	3	519000.0
3	4	634000.0
4	5	714700.0



Aus dem proportionalen Fit ergibt sich für ω_0 ein Wert von 944920 Hz mit einer Unsicherheit von 6395 Hz. Es wurde ein 5% Ableseungenauigkeit für die Liassau-Figur angenommen, da es nicht so einfach war zu erkennen, wann genau die Geradenform erreicht war und eine 10 Hz Messunsicherheit auf der Frequenz. Es ist außerdem klar zu erkennen, dass die proportionale gerade kein gutes Modell für die Daten ist, da ein sichtbarer Block überhalb der Kurve liegt.

2.1.5 Aufgabe 2.5: Reflexionen

Studieren Sie Reflexionen am Ende der Drosselkette qualitativ. Schließen Sie hierzu das Ende der Drosselkette (impedanzfrei) kurz und fügen Sie eine entsprechende Impedanz am Kettenanfang zu, um dort Reflexionen so weit wie möglich zu reduzieren. Schließen Sie eine rechteckförmige Wechselspannung an, beschreiben Sie, was Sie beobachten und diskutieren Sie, ob diese Beobachtungen Ihrer Erwartung entsprechen.

```
[12]: %m In diesem Versuch soll die Reflektion am Ende der Drosselkette untersucht
      ↪ werden. Wir haben eine Rechteckspannung mit 20KHz an den Eingang der
      ↪ Drosselkette angelegt.

      plt.title(f"Reflektion")
      pltimg("Fotos/Reflection_1.jpg")

      %m Hier ist keine sinnvolle Aussage über die Reflektion möglich, da keine genaue
      ↪ Abgrenzung der reflektierten Welle möglich ist. Es ist lediglich erkennbar,
      ↪ dass es Verzerrungen im Signal gibt. Dies könnte daran liegen, dass der
      ↪ Eingangswiderstand unserer Drosselkette defekt war und somit
      ↪ Mehrfachreflektionen am Eingang möglich waren, die das Signal verzerrt haben.
```

In diesem Versuch soll die Reflektion am Ende der Drosselkette untersucht werden. Wir haben eine Rechteckspannung mit 20KHz an den Eingang der Drosselkette angelegt.



Hier ist keine sinnvolle Aussage über die Reflektion möglich, da keine genaue Abgrenzung der reflektierten Welle möglich ist. Es ist lediglich erkennbar, dass es Verzerrungen im Signal gibt. Dies könnte daran liegen, dass der Eingangswiderstand unserer Drosselkette defekt war und somit Mehrfachreflektionen am Eingang möglich waren, die das Signal verzerrt haben.

2.2 Aufgabe 3: Koaxialkabel

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-3.md](#).

2.2.1 Aufgabe 3.1: Bestimmung der charakteristischen Impedanz Z_0

Bestimmen Sie Z_0 für $\omega \ll \omega_0$, analog zu Aufgabe 2.1. Hierzu können Sie eine rechteckförmige Spannung mit der Frequenz $\nu = 1.1$ MHz verwenden. (Die Grenzfrequenz von Koaxialkabeln liegt i.a. im GHz-Bereich.)

[13]: `uR = ufloat(48,5)`

```
%m In diesem Versuch soll die charakteristische Impedanz $Z_0$ des
↳Koaxialkabels bestimmt werden. Wir haben bei einer Einstellung des
↳Potentiometers am Ausgang der Drosselkette von $[!uR.n:.2f!]\,\Omega$ die
↳geringsten Reflektionen erhalten. Zur Einstellung haben wir eine
↳Rechteckwelle verwendet, da man hier besonders gut die Veränderungen durch
↳eine Reflektion erkennen kann. Da das Potentiometer relativ viel Spiel hatte
↳haben wir nur eine Unsicherheit von $[!uR.s:.2f!]\,\Omega$.
```

In diesem Versuch soll die charakteristische Impedanz Z_0 des Koaxialkabels bestimmt werden. Wir haben bei einer Einstellung des Potentiometers am Ausgang der Drosselkette von $48.00\ \Omega$ die geringsten Reflektionen erhalten. Zur Einstellung haben wir eine Rechteckwelle verwendet, da man hier besonders gut die Veränderungen durch eine Reflektion erkennen kann. Da das Potentiometer relativ viel Spiel hatte haben wir nur eine Unsicherheit von $5.00\ \Omega$.

2.2.2 Aufgabe 3.2: Bestimmung der Verzögerungszeit pro Länge τ'

Bestimmen Sie die **Verzögerungszeit pro Länge des Kabels** τ' . Tun Sie dies auf zwei verschiedene Arten:

- Indem Sie das Signal am Anfang mit dem Signal am Ende des Kabels vergleichen. Verwenden Sie hierzu ein geeignetes rechteckförmiges Signal und verwenden Sie das Oszilloskop im Zweikanalmodus.
- Bestimmen Sie die Laufzeit, indem Sie das Kabelende (impedanzfrei) kurzschließen und am Kabelanfang das am Kabelende reflektierte Signal mit dem Referenzsignal des Frequenzgenerators vergleichen. Wählen Sie auch hier eine rechteckförmige Wechselspannung.
- Vergleichen Sie beide Ergebnisse im Rahmen der bestimmten Unsicherheiten.

```
[14]: l = 10
      udt1 = ufloat(62e-9,10e-9)
      udt2 = ufloat(125e-9,3e-9)

      u 1 = udt1/l
      u 2 = udt2/2/l

      %m In dieser Aufgabe geht es darum die Verzögerungszeit im Kabel zu untersuchen.
      ↪ Das Kabel hat eine Länge von [!l!]m. Wir haben eine Verzögerung am Ende des
      ↪Kabels von [!udt1.n!]s±[!udt1.s!]s gemessen. Am Anfang des Kabels haben wir
      ↪eine Verzögerung von [!udt2.n!]s±[!udt2.s!]s gemessen. Damit ergeben sich
      ↪sich  $\tau_1 = [!u 1.n*10**9!]\frac{\text{ns}}{\text{m}} \pm [!u 1.s*10**9!]\frac{\text{ns}}{\text{m}}$  und
      ↪ $\tau_2 = [!u 2.n*10**9!]\frac{\text{ns}}{\text{m}} \pm [!u 2.s*10**9!]\frac{\text{ns}}{\text{m}}$ .
```

In dieser Aufgabe geht es darum die Verzögerungszeit im Kabel zu untersuchen. Das Kabel hat eine Länge von 10m. Wir haben eine Verzögerung am Ende des Kabels von $6.2e-08\text{s} \pm 1e-08\text{s}$ gemessen. Am Anfang des Kabels haben wir eine Verzögerung von $1.25e-07\text{s} \pm 3e-09\text{s}$ gemessen. Damit ergeben sich $\tau_1 = 6.2 \frac{\text{ns}}{\text{m}} \pm 1.0 \frac{\text{ns}}{\text{m}}$ und $\tau_2 = 6.25 \frac{\text{ns}}{\text{m}} \pm 0.15 \frac{\text{ns}}{\text{m}}$.

2.2.3 Aufgabe 3.3: Bestimmung der Dielektrizitätskonstanten ϵ

Bestimmen Sie ϵ aus den Werten, die Sie in den Aufgaben 4.1 und 4.2 bestimmt haben. Verwenden Sie hierzu alle Messungen, die Sie durchgeführt haben:

- Bestimmen Sie ϵ aus Ihrer Messung von Z_0 aus Aufgabe 4.1;
- Bestimmen Sie ϵ aus den beiden Messungen von τ' aus Aufgabe 4.2;
- Vergleichen Sie die von Ihnen bestimmten Werte für ϵ untereinander und vergleichen Sie sie mit der Erwartung aus den technischen Angaben zu diesem Versuch (siehe [Datenblatt.md](#)).

```

[15]: c = 299792458
u_0 = ufloat(1.25663706212e-6, 0.0000000000019e-6)
u_0 = ufloat(8.8541878128e-12, 0.000000000013e-12)
di = 0.5e-3
da = 1.75e-3

C = 925e-12

u = (2*pi*R**2*C)/(u_0*np.log(da/di))

u_1 = c**2*u_1**2
u_2 = c**2*u_2**2

%m Aus den zuvor bestimmten Werten für  $\epsilon$  lässt sich  $\epsilon$  mit folgender
↪ Formel bestimmen:  $\epsilon = \frac{c^2}{v^2 \mu}$ 
↪  $\frac{c^2}{v^2 \mu}$ .  $\mu$  ist in diesem Fall in guter Näherung gleich
↪ 1. Damit ergeben sich  $\epsilon_{\tau 1} = [!u_1.n:.2f!] \pm [!u_1.s:.2f!]$ 
↪  $\epsilon$  sowie  $\epsilon_{\tau 2} = [!u_2.n:.2f!] \pm [!u_2.s:.2f!]$ 

u_Z0 = u_0*np.log(da/di)**2/(4*pi**2*u_R**2*u_0)

%m Auch aus  $Z_0$  lässt sich  $\epsilon_{Z_0} = \frac{\mu_0 \mu}{4 \pi^2 Z_0^2 \epsilon_0} \left( \log\left(\frac{r_a}{r_i}\right) \right)^2$ 
↪  $Z_0^2 \epsilon_{Z_0} \left( \log\left(\frac{r_a}{r_i}\right) \right)^2$  ausrechnen. Damit
↪ ergibt sich  $\epsilon_{Z_0} = [!u_Z0.n:.2f!] \pm [!u_Z0.s:.2f!]$ .

= (u_1+u_2+u_Z0)/3

%m Damit ergibt sich ein gemittelter Gesamtwert von  $\epsilon_{Z_0} = [!.n:.2f!]$ 
↪  $\pm [!.s:.2f!]$ .

_tech = C/(2*pi*1*u_0)*np.log(da/di)

%m Über die technischen Angaben zum Versucht lässt sich  $\epsilon$  auch
↪ bestimmen mit einem Wert von  $\epsilon_{Z_0} = [!_tech.n:.2f!]$ .

%m Damit ist der von uns bestimmte Wert 2 weit vom angegebenen Wert entfernt.

```

Aus den zuvor bestimmten Werten für τ lässt sich ϵ mit folgender Formel bestimmen: $\epsilon = \frac{c^2}{v^2 \mu} = \frac{c^2 \tau^2}{\mu}$. μ ist in diesem Fall in guter Näherung gleich 1. Damit ergeben sich $\epsilon_{\tau 1} = 3.45 \pm 1.11$ sowie $\epsilon_{\tau 2} = 3.51 \pm 0.17$

Auch aus Z_0 lässt sich $\epsilon_{Z_0} = \frac{\mu_0 \mu}{4 \pi^2 Z_0^2 \epsilon_0} \left(\log\left(\frac{r_a}{r_i}\right) \right)^2$ ausrechnen. Damit ergibt sich $\epsilon_{Z_0} = 2.45 \pm 0.51$.

Damit ergibt sich ein gemittelter Gesamtwert von $\epsilon_{Z_0} = 3.14 \pm 0.41$.

Über die technischen Angaben zum Versucht lässt sich ϵ auch bestimmen mit einem Wert von

$$\epsilon_{Z_0} = 2.08.$$

Damit ist der von uns bestimmte Wert 2 weit vom angegebenen Wert entfernt.

[]: