Name: Luca Cordes, 444900
Name: Mahmut Can Dogan, 435714

Experimentalphysik II (SS 2023/2024)

Übung 11

Tutorium: 2 Abgabe: 30.06.2023

Aufgabe 1: Magentfeldberechnung

[siehe hinten]

Aufgabe 2: Induktionsstrombremse

$$
\begin{aligned}
U_{ind} &= -\frac{\mathrm{d}\phi}{\mathrm{d}t} \\
&= -B\frac{\mathrm{d}A}{\mathrm{d}t} \\
&= -Bdv \\
I_{ind} &= \frac{U_{ind}}{R} \\
&= -\frac{Bdv}{R}
\end{aligned}
$$

$$
\begin{aligned}
0 &= \Delta W_{pot} + \Delta W_{kin} + \Delta W_{el} \\
0 &= mgx + \frac{m}{2}v^2 + \int I_{ind} U_{ind} \,\mathrm{d}t \\
0 &= mgx + \frac{m}{2}v^2 + \frac{B^2d^2}{R}\int v^2 \,\mathrm{d}t \\
0 &= mgv + mva + \frac{B^2d^2}{R}v^2 \\
a &= -g - \frac{B^2d^2}{mR}v \\
F_{ges} &= \underbrace{-mg}_{F_g} + \underbrace{\frac{-B^2d^2}{R}v}_{F_{el}}
\end{aligned}
$$

$$
\begin{aligned}
v_{max} &\implies a = 0 \\
0 &= -g - \frac{B^2d^2}{mR}v_{max} \\
v_{max} &= -g\frac{mR}{B^2d^2} \\
&\approx -9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}\frac{45\,\mathrm{g}\cdot 4\,\Omega}{13^2\,\mathrm{T}^2\cdot 15^2\,\mathrm{cm}^2} \\
&\approx 0.464\frac{\mathrm{m}}{\mathrm{s}^2}
\end{aligned}
$$

Aufgabe 3: Ausschaltvorgang an einer Spule

$$\begin{aligned}
0 &= U_R + U_{ind} \\
0 &= IR + L\dot{I} \\
0 &= \dot{I} + \frac{R}{L} I \\
I(t) &= c \cdot e^{-\frac{R}{L}t} \\
&= I_0 \cdot e^{-\frac{R}{L}t}
\end{aligned}$$

Wobei I_0 der Strom ist, der vor Entfernen der Spannungsquelle geflossen ist.

Aufgabe 4: Dynamo

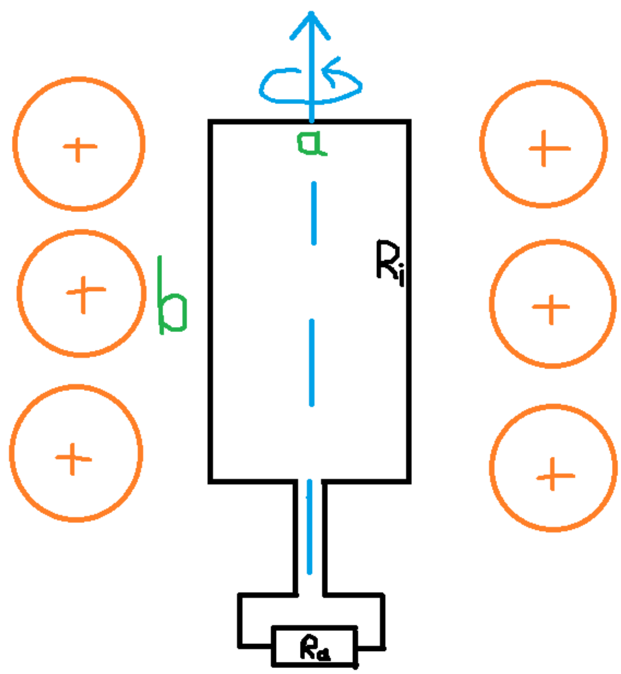

Figure 1: Versuchsaufbau

$$\begin{aligned}
U_{ind} &= -N \frac{\mathrm{d}\phi}{\mathrm{d}t} \\
&= -NB \frac{\mathrm{d}A}{\mathrm{d}t} \\
&= -NB \frac{\mathrm{d}}{\mathrm{d}t} \left(ab \cos \varphi \right) \\
&= -NB \frac{\mathrm{d}}{\mathrm{d}t} \left(ab \cos \left(\omega t \right) \right) \\
&= NabB\omega \sin \left(\omega t \right) \\
I_{ind} &= \frac{NabB\omega}{R_i + R_a} \sin \left(\omega t \right)
\end{aligned}$$

$$
\begin{aligned}
P_{mech} &= P_{el}\\
M\omega &= \frac{U_{ind}^2}{R_{ges}}\\
&= \frac{N^2a^2b^2B^2\omega^2\sin^2(\omega t)}{R_i+R_a}\\
M &= \frac{N^2a^2b^2B^2\omega}{R_i+R_a}\sin^2(\omega t)\\
&\approx 250^2\cdot 5^2\,\mathrm{cm}^2\cdot 3^2\,\mathrm{cm}^2\cdot 0.5^2\,\mathrm{T}^2\frac{\omega}{R_i+R_a}\sin^2(\omega t)\\
&\approx 3.52\cdot 10^{-2}\,\mathrm{V\,s\,m^2}\frac{\omega}{R_i+R_a}\sin^2(\omega t)\\
\vec{M} &= \frac{N^2a^2b^2B^2\omega}{R_i+R_a}\sin^2(\omega t)\,\vec{e}_\omega
\end{aligned}
$$

Aufgabe 5: Nichtperiodische Fourierzerlegung

$$
U(t) = \begin{cases} 0\,\mathrm{V} & \text{für} \quad t < 0\,\mathrm{s}\\ U_0 & \text{für} \quad 0\,\mathrm{s}\le t\le t_0\\ 0\,\mathrm{V} & \text{für} \quad t_0 < t\end{cases}
$$

(a)

$$
\begin{aligned}
F(U(t)\,;\,\omega) &= \int_{-\infty}^{\infty} U(t)\cdot e^{-i\omega t}\,\mathrm{d}t\\
&= U_0\int_0^{t_0} e^{-i\omega t}\,\mathrm{d}t\\
&= -\frac{U_0}{i\omega}e^{-i\omega t}\Big|_0^{t_0}\\
&= \frac{U_0}{i\omega}\left(1-e^{-i\omega t_0}\right)\\
&= \frac{iU_0}{\omega}\left(e^{-i\omega t_0}-1\right)\\
&= \frac{iU_0}{\omega}\left(\cos(\omega t_0)-i\sin(\omega t_0)-1\right)\\
\Re(F(U(t)\,;\,\omega)) &= \frac{U_0}{\omega}\sin(\omega t_0)\quad [= 2\pi\cdot a(\omega)]\\
\Im(F(U(t)\,;\,\omega)) &= \frac{U_0}{\omega}\left(\cos(\omega t_0)-1\right)\quad [= 2\pi\cdot b(\omega)]
\end{aligned}
$$

(b)

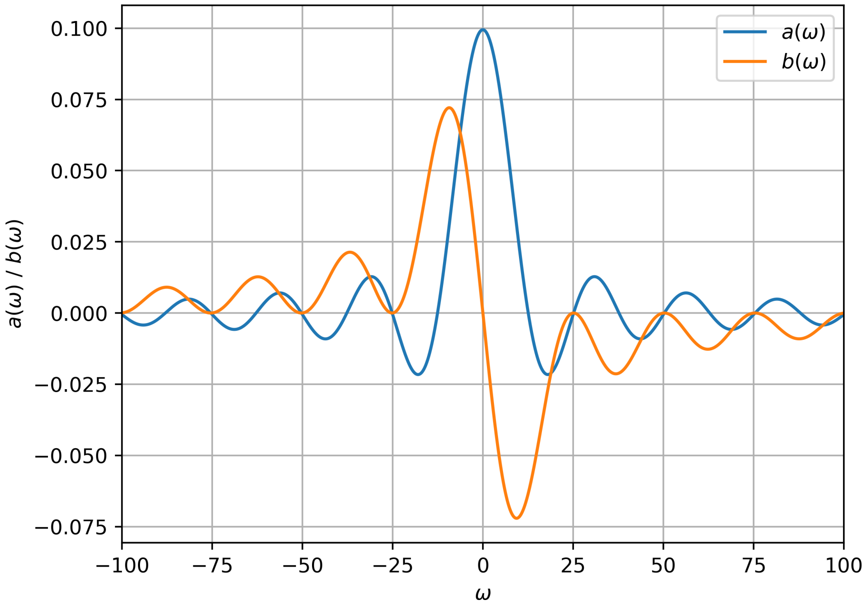

Figure 2: Plot von $a(\omega)$ und $b(\omega)$

(c)

$$U(t) = \int_{-\infty}^{\infty} (\cos(\omega t)a(\omega) + \sin(\omega t)b(\omega))\,\mathrm{d}\omega$$

$$U(t) = \frac{U_0}{2\pi}\int_{-\infty}^{\infty} \frac{\cos(\omega t)\sin(\omega t_0)}{\omega} + \frac{U_0}{2\pi}\int_{-\infty}^{\infty} \frac{\sin(\omega t)(\cos(\omega t_0) - 1)}{\omega}\,\mathrm{d}\omega$$

$$\int_{-\infty}^{\infty} \frac{\cos(\omega t)\sin(\omega t_0)}{\omega} \overset{(1)}{=} \frac{1}{2}\pi(\operatorname{sgn}(t_0 - t) + \operatorname{sgn}(t_0 + t))$$

$$\int_{-\infty}^{\infty} \frac{\sin(\omega t)(\cos(\omega t_0) - 1)}{\omega}\,\mathrm{d}\omega \overset{(2)}{=} -\frac{1}{2}\pi(\operatorname{sgn}(t_0 - t) - \operatorname{sgn}(t_0 + t) + 2\operatorname{sgn}(t))$$

$$\begin{aligned} U(t) &= \frac{U_0}{2\pi}\cdot\pi(\operatorname{sgn}(t_0 - t) + \operatorname{sgn}(t)) \\ &= \frac{U_0}{2}(\operatorname{sgn}(t_0 - t) + \operatorname{sgn}(t)) \\ &= \begin{cases} 0\,\mathrm{V} & \text{für} \quad t < 0\,\mathrm{s} \\ U_0 & \text{für} \quad 0\,\mathrm{s} \le t \le t_0 \\ 0\,\mathrm{V} & \text{für} \quad t_0 < t \end{cases} \end{aligned}$$

(1) : Integral mit Wolfram Alpha, nachdem wir es zwei Stunden lang erfolglos selbst versucht haben
(2) : "

Aufgabe 1: Magentfeldberechnung
(a/b)

11

June 27, 2023

1 Jupyter-Notebook zur Experimentalphysik II, SS 2023

von Dr. Markus Merschmeyer, III. Physikalisches Institut A, RWTH Aachen University

1.1 Übungsaufgabe: Magnetfeldberechnung (numerisch)

[1]:
```
import numpy as np                # numpy-Paket importieren
from math import *
%matplotlib inline
import matplotlib.pyplot as plt # Paket zur Diagrammerstellung
import scipy.constants as scc   # Naturkonstanten aus "scipy"
```

[2]:
```
mu0  = scc.mu_0 # magnetische Feldkonstante in N A^-2
print("magn. Feldkonstante:  ",mu0,"N A^-2")

# sonstige Parameter
R    = 0.03      # Radius der Leiterschleife in m
print("Radius Leiterschleife:",R,"m")
tmin = 0.0       # Startwert Kurvenparameter
tmax = 1.0       # Endwert Kurvenparameter
print("Kurvenparameter:      ",tmin,"bis",tmax)
```

```
magn. Feldkonstante:   1.25663706212e-06 N A^-2
Radius Leiterschleife: 0.03 m
Kurvenparameter:       0.0 bis 1.0
```

Die Leiterschleife soll durch die Funktion

$$\vec{s}(t) = \begin{pmatrix} R\cdot\cos(2\pi\,t) \\ -R\cdot\sin(2\pi\,t) \\ 0 \end{pmatrix}\,,\quad 0\leq t<1$$

beschrieben werden. Dabei ist R der Radius der Schleife und t ein Kurvenparameter zur Beschreibung des Weges entlang der Schleife.

[3]:
```
def v_loop(R=0.03, t=0.0):
    return R * np.array([np.cos(2*np.pi*t),  # x-Koordinate in m,
                         -np.sin(2*np.pi*t),  # y-Koordinate in m,
                         0]) # z-Koordinate in m
```

Funktion zur numerischen Berechnung des Magnetfeldes einer Leiterschleife nach Biot-Savart:

[4]:
```
def biot_savart(R=0.03, t_min=0.0, t_max=1.0, n_step=10, I=10.0, x=0.0, y=0.0,␣
↪z=0.1):
    v_B     = np.array([x, y, z])         # Ortsvektor fuer die Feldberechnung␣
↪initialisieren
    v_Btot  = np.array([0.,0.,0.])  # initialisiere den Summenvektor des␣
↪B-Feldes
    d_t     = (t_max-t_min) / n_step  # Schrittlaenge entlang einer␣
↪Spulenwindung berechnen

    # berechne Liste mit den Vektor-Wegelementen d\vec{s} entlang der Spule
    v_ds = np.array([v_loop(R, t) - v_loop(R, t+d_t) for t in np.arange(t_min,␣
↪t_max, d_t)])

    # berechne Liste mit Abstandsvektoren vom jeweiligen Ort (v_loop()) auf der␣
↪Spule zum Ort (v_B),
    # an dem das B-Feld bestimmt werden soll: \vec{r} = v_B - v_loop()
    v_r = np.array([v_B - v_loop(R, t) for t in np.arange(t_min, t_max, d_t)])

    # berechne Liste der Kreuzprodukte \vec{r}xd\vec{s}
    crossp = np.cross(v_ds, v_r)
    scalar = mu0 / (4*np.pi) * I * np.sum(v_r**2,axis=1)**(-3/2)
    dB = np.array([scalar[i] * crossp[i] for i in range(n_step)])

    # bilde Summe ueber alle Vektoren -> Vektor des B-Felds am Ort \vec{r}
    v_Btot = np.sum(dB, axis=0)

    return v_Btot

biot_savart()
```

[4]:
```
array([ 4.23516474e-22, -9.52912066e-22,  4.64859219e-06])
```

Die Funktion zur analytischen Berechnung des Magnetfeldes einer Leiterschleife entlang der z-Achse lautet (Kap. 4.2.3, Gleichung 27):

$$B_z(z) = \frac{\mu_0}{2} \cdot I \cdot \frac{R^2}{\sqrt{z^2 + R^2}^3}$$

[5]:
```
def Bz_LS(R=0.03, I=10.0, z=0.05):
    return mu0/2 * I * R**2 / (z**2 + R**2)**(3/2)
```

Vergleich von analytischer und numerischer Berechnung des Bz-Feldes als Funktion des z-Abstands:

[6]:
```
fig, ax = plt.subplots(2,2)
fig.set_dpi(1000)
fig.set_figheight(10)
```

```
fig.set_figwidth(14)
ax = ax.flat
z = np.linspace(0,0.2,100)
for n in range(4):
    B = np.array([Bz_LS(z=i) for i in z])
    B_num = np.array([biot_savart(z=i,n_step=10*2**n)[2] for i in z])

    #ax[n].title(f"Numerisches vs. analytisches Biot-Savart-Gesetz\n Maximaler␣
 ↪relativer Fehler = {max(B_num/B) - 1:.3f}")
    ax[n].plot(z,B,label="$B(z)$ - analytisch")
    ax[n].plot(z,B_num,label=f"$B(z)$ - numerisch $(n_{{step}}={n})$")
    ax[n].set_xlabel("$z$ in Metern")
    ax[n].set_ylabel("$B(z)$ in Tesla")
    ax[n].set_xlim(0,0.2)
    ax[n].set_ylim(bottom=0)
    ax[n].legend()
fig.tight_layout()
plt.show()
```