Gekoppelte Harmonische Oszillatoren

November 12, 2022

0.1 Zwei gekoppelte harmonische Oszillatoren

Notebook erstellt am 08.11.2022 von C. Rockstuhl, überarbeitet von Y. Augenstein

In diesem Notebook wollen wir noch einen Schritt weiter gehen und zwei gekoppelte harmonische Oszillatoren untersuchen. Typische physikalische Situationen lassen sich einfach in Ergänzung der vorher bereits betrachteten Systeme finden:

- Zwei Massen m, die jeweils an einer Feder aufgehängt sind. Gleichzeitig sind die beiden Massen aber auch noch einmal durch eine dritte Feder miteinander verbunden. Es sind viele Konfigurationen vorstellbar, wir wollen aber davon ausgehen, dass diese Feder gerade so dimensioniert ist, dass wenn die beiden Massen sich in Ruhelage befinden, die sie verbindende Feder keine zusätzliche Kraft auf die beiden Massen ausüben sollen. Wird eine der beiden Massen so bewegt, dass sich der Abstand vergrößert, wird die zusätzliche Federkraft so auf die beiden Massen wirken, dass diese ihren Abstand wieder verringern. Das wäre eine anziehende Kraft, die auf beide Massen gleichzeitig wirkt. Bei einem kleineren Abstand als der Abstand der beiden Massen in Ruhelage wird die zusätzliche Federkraft so auf die beiden Massen wirken, dass diese ihren Abstand wieder vergrößern.
- Zwei mathematische Pendel, die z.B. mittels einer Feder oder einem Gummiband miteinander verbunden sind. Die Physik soll die gleiche sein wie bei den an Federn aufgehangenen Massen.
- Bei der Beschreibung der Ladung auf dem entsprechenden Kondensator in zwei nahe beieinander liegenden Schwingkreisen. Ein zeitlich veränderlicher Strom in einem der beiden Schwingkreise wird ein Magnetfeld induzieren, welches sich im ganzen Raum ausbreitetet. Dieses Magnetfeld wird in dem benachbarten Schwingkreis einen Strom induzieren. Auf diese Art sind die beiden Schwingkreise miteinander verbunden.

Im Vergleich zum vorhergehenden Beispiel des einfachen harmonischen Oszillators, betrachten wir im Folgenden also immer zwei Oszillatoren. Diese werden individuell und ohne Kopplung durch zwei identische Differentialgleichungen beschrieben, die einfach charakterisiert sind durch verschiedene Paramater wie Eigen- bzw. Resonanzfrequenz und einer Dämpfung. Wir würden im Folgenden gleich den Fall betrachten, dass die Parameter der beiden harmonischen Oszillatoren unterchiedlich sein können, aber numerisch können wir die Parameter auch identisch wählen, um identische Oszillatoren zu betrachten. Wir betrachten hier keine externe Kraft, um die Allgemeingültigkeit des Modells nicht auf die Spitze zu treiben und mit einer geringeren Anzahl an freien Paramatern die Dynamik in dem System zu studieren.

Die beiden Differentialgleichungen für die ungekoppelten harmonischen Oszillatoren lauten

$$
\begin{aligned}
\ddot{x}_1 + 2\gamma_1\dot{x}_1 + \omega_{01}^2 x_1 &= 0 \qquad (1)\\
\ddot{x}_2 + 2\gamma_2\dot{x}_2 + \omega_{02}^2 x_2 &= 0 \quad . \qquad (2)
\end{aligned}
$$

Alles, was bei Kopplung jetzt noch passiert, ist, dass ein zusätzlicher Kraftterm in jeder der beiden Gleichungen auftaucht, der proportional sein soll zur Differenz der beiden Ortskoordinaten. Wenn die beiden Massen sich in ihren Ruhepositionen befinden, wirkt so keine Kraft. Auch wenn die Differenz zwischen den Auslenkungen der beiden Massen verschwindet, wirkt ebenfalls keine Kraft. Nur bei Abweichungen soll eine Kraft wirken:

$$
\begin{aligned}
\ddot{x}_1 + 2\gamma_1\dot{x}_1 + \omega_{01}^2 x_1 - \kappa(x_2 - x_1) &= 0 \qquad (3)\\
\ddot{x}_2 + 2\gamma_2\dot{x}_2 + \omega_{02}^2 x_2 - \kappa(x_1 - x_2) &= 0 \quad . \qquad (4)
\end{aligned}
$$

Diese zwei gekoppelten Differentialgleichungen zweiter Ordnung benötigen wieder vier Anfangsbedingungen zur eindeutigen Lösung. Dies sind der Ort und die Geschwindigkeit der beiden Massen zu einem Anfangszeitpunkt $t_{\text{Anfang}} = t_0$. Numerisch formulieren wir diese beiden Gleichungen wieder genau wie beim einfachen harmonischen Oszillator in vier gekoppelte Differentialgleichungen erster Ordnung um, die wir wieder mit den üblichen Mitteln integrieren können.

Wir definieren also

$$
\mathbf{X}(t) = \begin{pmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{pmatrix} = \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} \quad .
$$

Die Differentialgleichung erster Ordnung für die zeitliche Entwicklung dieses Vektors lautet:

$$
\dot{\mathbf{X}}(t) = \begin{pmatrix} \dot{x}_1 \\ -2\gamma_1\dot{x}_1 - \omega_{01}^2 x_1 + \kappa(x_2 - x_1) \\ \dot{x}_2 \\ -2\gamma_2\dot{x}_2 - \omega_{02}^2 x_2 + \kappa(x_1 - x_2) \end{pmatrix} \quad .
$$

Diese werden wir im Folgenden lösen und die Ergebnisse wieder graphisch anzeigen. Beachten Sie bitte immer die Bedeutung der beiden Koordinaten x_1 und x_2. Diese entsprechen der Auslenkung der beiden Massen aus Ihrer entsprechenden Ruhelage.

0.1.1 Setup

Wie im vorangegangenen Notebook zum einfachen harmonischen Oszillator bereiten wir wieder den interaktiven Plot vor.

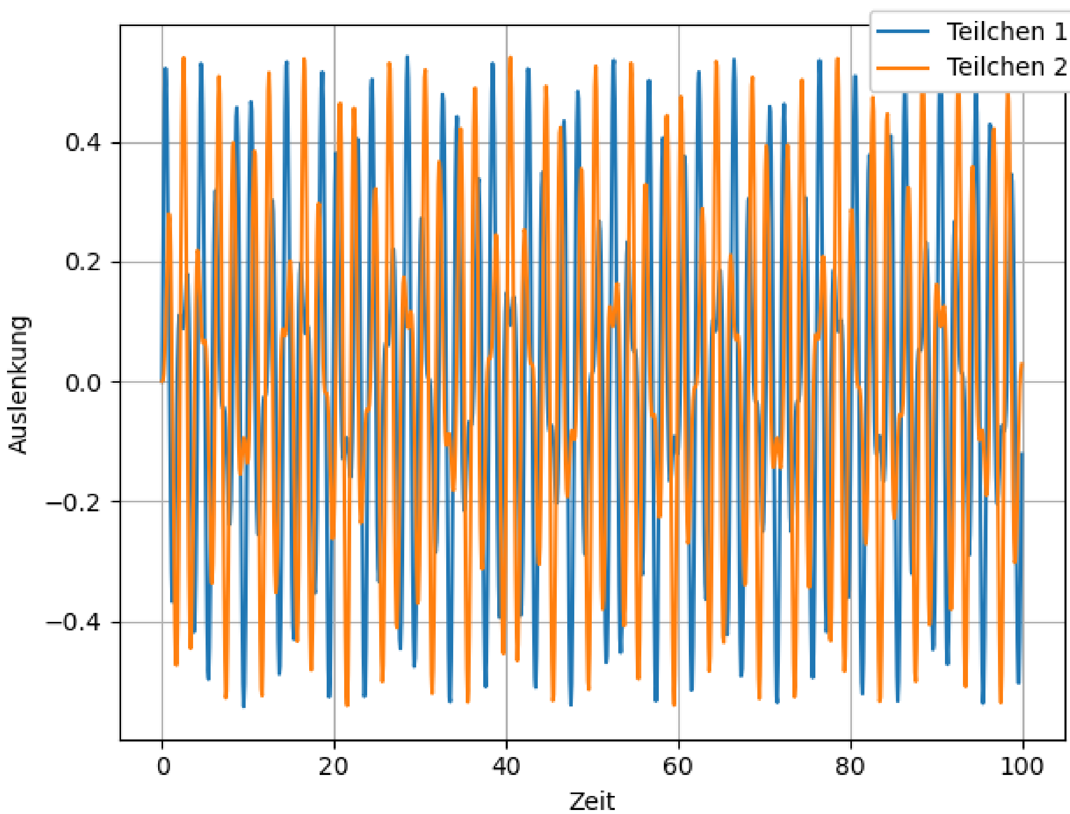
[1]:
```
%matplotlib ipympl
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp  # Solve Initial Value Problen, siehe␣
↪Notebook "DGL und Wurf"
from ipywidgets import interactive  # Interaktive Plots mit Schiebereglern


def solve_and_plot(ode_problem, x0, t):
    plt.close()
    fig, ax = plt.subplots(1, 1)
    (line0,) = ax.plot(t, np.zeros_like(t), c="tab:blue", label="Teilchen 1")
    (line1,) = ax.plot(t, np.zeros_like(t), c="tab:orange", label="Teilchen␣
↪2")

    ax.grid()
    ax.set_xlabel("Zeit")
    ax.set_ylabel("Auslenkung")
    fig.legend(framealpha=1)
    fig.tight_layout()

    def solve_and_draw(**kwargs):
        # Zunächst lösen wir die DGL
        sol = solve_ivp(ode_problem, (t[0], t[-1]), x0, t_eval=t,␣
↪args=kwargs.values())

        # Und zeichnen dann die relevaten Linien
        line0.set_ydata(sol.y[0])
        line1.set_ydata(sol.y[2])

        # Und als Letztes skalieren wir die Achsen neu, da diese an
        # die neuen Werte von y angepasst werden müssen
        ax.relim()
        ax.autoscale_view()

    return solve_and_draw
```

Und nun definieren und lösen wir die DGL, diesmal für zwei gekoppelte harmonische Oszillatoren.

[6]:
```
def coupled_harmonic_oscillator(t, X, omega01, omega02, gamma1, gamma2,␣
↪kappa):
    x1, dx1, x2, dx2 = X
    ddx1 = -2 * gamma1 * dx1 - omega01**2 * x1 + kappa * (x2 - x1)
    ddx2 = -2 * gamma2 * dx2 - omega02**2 * x2 + kappa * (x1 - x2)
    return [dx1, ddx1, dx2, ddx2]
```

```
# Anfangsbedingungen: [Auslenkung, Geschwindigkeit]
x0 = [0, 2, 0, 0]  # Hier beginnen wir mit einer endlichen Geschwindigkeit

t = np.linspace(0, 100, 1000)

interactive(
    solve_and_plot(coupled_harmonic_oscillator, x0, t),
    omega01=(0, 2 * np.pi, 0.01),  # Resonanzfrequenz Oszillator 1
    omega02=(0, 2 * np.pi, 0.01),  # Resonanzfrequenz Oszillator 2
    gamma1=(0, 1, 0.01),  # Dämpfungskoeffizient Oszillator 1
    gamma2=(0, 1, 0.01),  # Dämpfungskoeffizient Oszillator 2
    kappa=(0, 10, 0.1),  # Kopplungskoeffizient
)
```

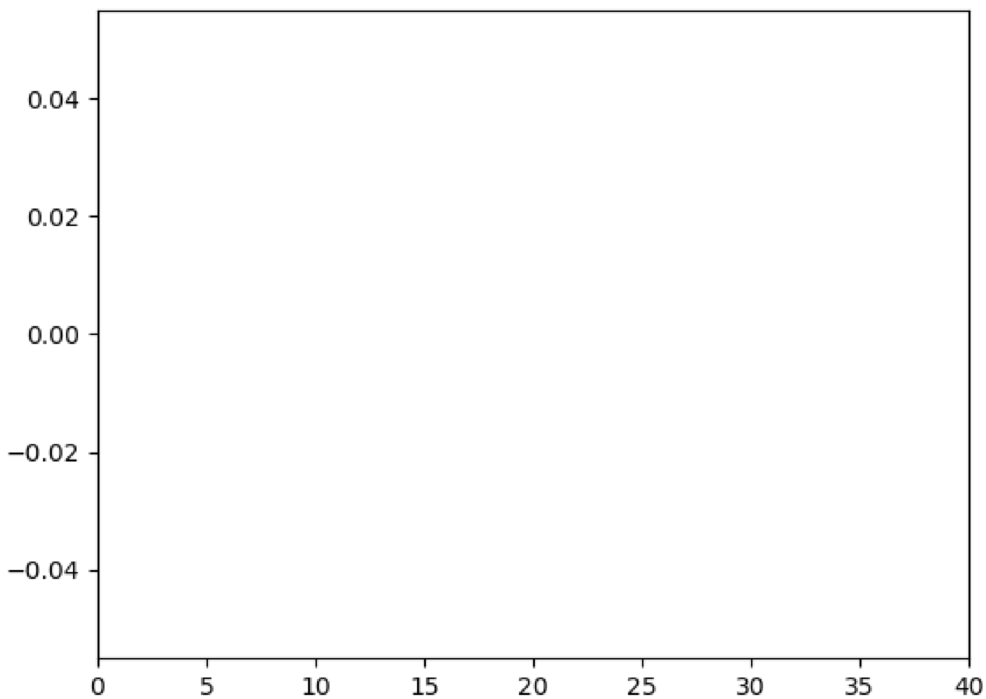
```
interactive(children=(FloatSlider(value=3.14, description='omega01', max=6.
↪283185307179586, step=0.01), FloatSâĂę
```



Um die Bewegung der beiden Teilchen etwas zu veranschaulichen, können wir sie auch zeitaufgelöst, also als Animation, darstellen.

[7]:
```python
from matplotlib import animation

# Anfangsbedingungen: [Auslenkung1, Geschwindigkeit1, Auslenkung2,␣
↪Geschwindigkeit2]
x0 = [0, 2, 0, 0]  # Teilchen 1 mit endlichee Geschwindigkeit


# Zunächst lösen wir die DGL, diesmal definieren legen wir die␣
↪Anfangsbedingungen programmatisch fest.
t = np.linspace(0, 40, 400)
args = (
    np.pi,  # omega01
    np.pi,  # omega02
    0.01,    # gamma1
    0.01,    # gamma2
    0.4,    # kappa
)
sol = solve_ivp(coupled_harmonic_oscillator, (t[0], t[-1]), x0, t_eval=t,␣
↪args=args)

# Und nun bereiten wir den Plot vor
plt.close()
fig, ax = plt.subplots(1, 1)
(line0,) = ax.plot([], [], c="tab:blue", label="Teilchen 1")
(line1,) = ax.plot([], [], c="tab:orange", label="Teilchen 2")
ax.set_xlim(t[0], t[-1])

def init():
    # Zur Initialisierung müssen die Inhalte der Achsen gelöscht werden
    line0.set_data([], [])
    line1.set_data([], [])
    return line0, line1

def animate(idx):
    # Zur Animation werden wir immer die Linien in den Plot-Achsen neu␣
↪zeichnen
    # bis zu einem bestimmten Zeitindex `idx`.
    line0.set_data(t[:idx], sol.y[0][:idx])
    line1.set_data(t[:idx], sol.y[2][:idx])
    ax.relim()
    ax.autoscale_view()
    return line0, line1

# Die matplotlib-Funktion `FuncAnimation` nimmt eine bestehende "Figure" und␣
↪zeichnet
```

```
# dort an jedem Schritt mittels der von uns definierten Funktion `animate`␣
↪die
# nötigen Linien ein.
# https://matplotlib.org/stable/api/_as_gen/matplotlib.animation.
↪FuncAnimation.html
anim = animation.FuncAnimation(
    fig, animate, init_func=init, frames=range(len(t)), interval=50,␣
↪blit=True, repeat=False
)
```



Spielen Sie bitte auch hier wieder mit den Parametern und vielleicht auch mit den Anfangswerten etwas rum. Für identische harmonische Oszillatoren können Sie die gekoppelten harmonischen Oszillatoren z.B. in der symmetrischen oder antisymmetrischen Eigenmode anregen. Welche Parameter müssen hierfür gewählt werden? Woran erkennen Sie, dass Sie das System nicht in einer der beiden Eigenmoden anregen? Was passiert, wenn Sie nur einen der beiden Oszillatoren zu Beginn anregen und die Dämpfung einmal sehr klein wählen? Öndern Sie dann einmal noch die Kopplung von einem sehr großen zu einem sehr niedrigen Wert!

Was Sie hier machen können, ist numerisch experimentieren!