Übungsaufgaben 5

(Lerninhalte: *Listen, Funktionen, numpy, scipy.optimize.root, matplotlib*)

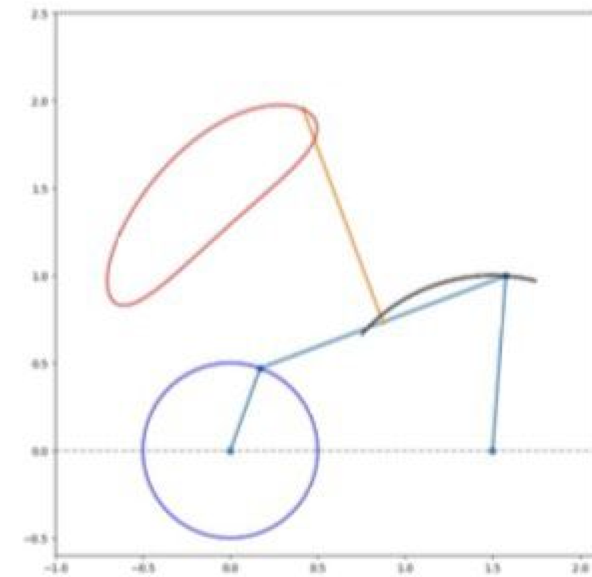
Berechnung der Bahnkurven einer Kurbelschwinge

Problemstellung

Eine Kurbelschwinge ist ein umlauffähiges Koppelgetriebe mit 4 Gelenken, dessen kürzestes Glied die Kurbel ist. Eine Drehbewegung/Umlauf (die blaue Kurve in der Skizze) der Kurbel wird in eine oszillierende Bewegung (das schwarze Kurvenstück in der Skizze) der Schwinge umgewandelt. In Anwendungen des Maschinenbaus, des Verkehrswesens oder der Verfahrenstechnik werden Kurbelschwingen zur Erzeugung von bestimmten Bewegungsabläufen genutzt.

Punkt P am Ende des Koppelauslegers (die orangene Strecke in der Skizze) ist der relevante Punkt, der die gewünschte Bewegung ausführt. Seine Bahnkurve nennt man Koppelkurve (die rote Kurve in der Skizze). Die Koppelkurve hängt stark von den konkret gewählten Parametern ab. Im Buch "Bewegungstechnik" von Fricke et al. (Fachbuchverlag Leipzig, 2015) sind verschiedene Koppelkurven gezeigt.

Es soll ein Programm entwickelt werden, das die Bahnkurve des Koppelausleger-Endpunkts P (die rote Kurve) berechnet und (statisch) darstellt. (Zum besseren Verständnis des Bewegungsablaufs siehe auch die zur Verfügung gestellte externe Animation.)



Motivation

Die Aufgabe ist ein klassisches Beispiel für die Analyse von Bewegungsabläufen mechanischer Systeme, wie sie in der Technischen Mechanik oder in der Getriebelehre vorkommen. Die Analyse des Bewegungsablaufs wird sowohl beim Entwurf des Systems benötigt als auch bei der späteren Auslegung. Beim Entwurf kommt es vor allem darauf an, dass das System die gewünschte Bewegung – also die Bewegung, die zur Funktionserfüllung notwendig ist – ausführt. Bei der Auslegung kommt es dann insbesondere auf die Beschleunigungen und die damit verbundenen Kräfte an. Mit diesen Informationen können dann die Lager und die Antriebe ausgelegt werden.

Weitere Informationen können z. B. auf den Webseiten

en.wikipedia.org/wiki/Linkage_(mechanical)

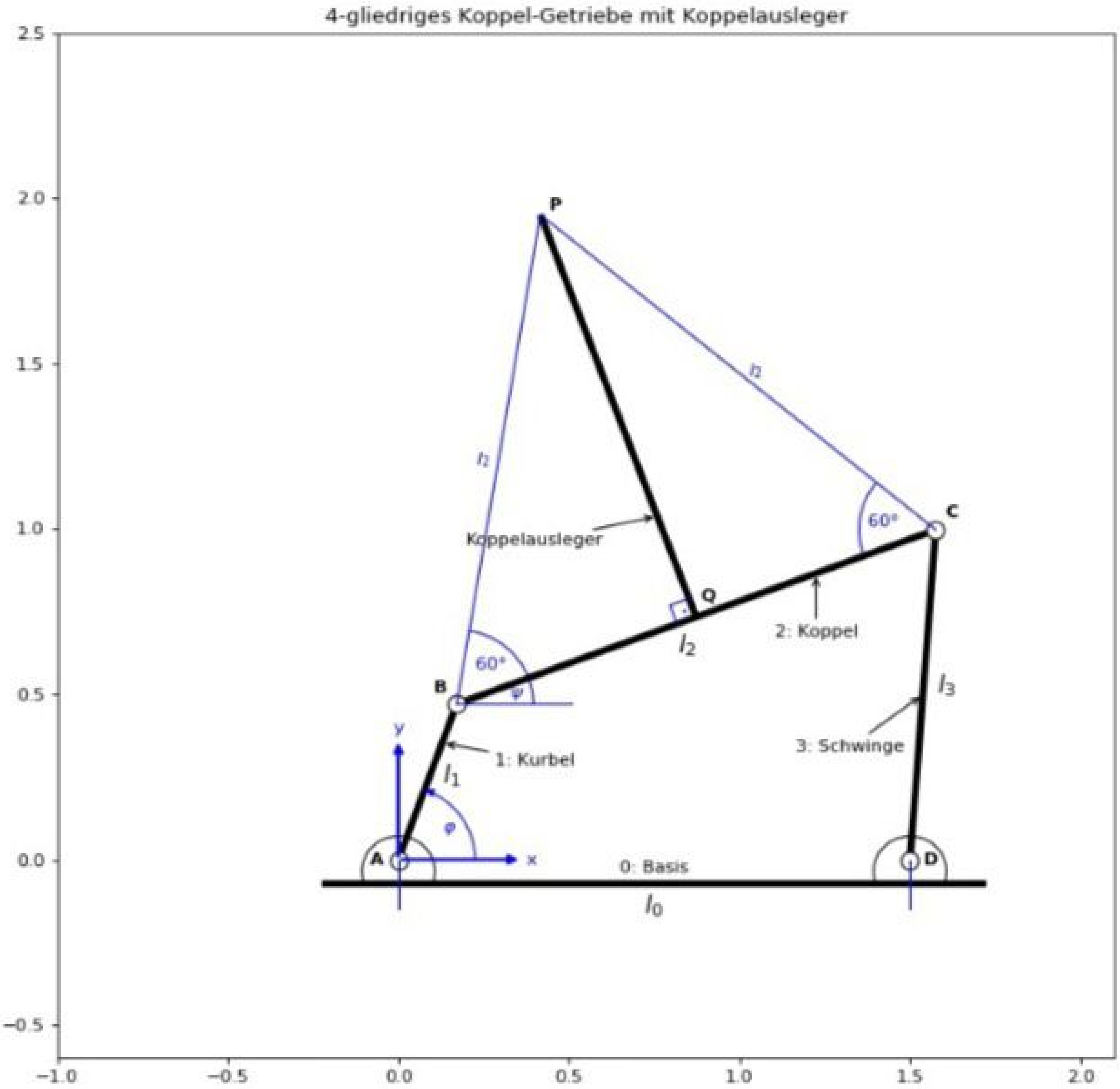
oder

en.wikipedia.org/wiki/Mechanism_(engineering)

gefunden werden.

Berechnung

Betrachten Sie zur Berechnung folgende (für die u.g. Parameterwerte maßstäbliche) Skizze, die beispielhaft die Stellung des Kurbelgetriebes mit einem Kurbelwinkel $\varphi = 70°$ zeigt:



Mit A_x, B_x usw. werden im Folgenden die X-Koordinaten bezeichnet, mit A_y, B_y usw. die Y-Koordinaten. Bei gegebenem φ und mit zunächst unbekanntem ψ ergibt sich

$A_x = 0$
$A_y = 0$
$B_x = l_1 \cdot \cos \varphi$
$B_y = l_1 \cdot \sin \varphi$
$C_x = B_x + l_2 \cdot \cos \psi$
$C_y = B_y + l_2 \cdot \sin \psi$
$D_x = l_0$
$D_y = 0$

Als entscheidende Kopplungsbedingung (zur Festlegung des Ortes von C und damit von ψ), damit die Stäbe "geschlossen zusammenpassen", stellt man fest, dass $\overline{CD}$ immer die Länge l_3 haben muss, es muss also gelten:

$$(C_x - D_x)^2 + (C_y - D_y)^2 - l_3^2 = 0$$

Als Beispiel für mögliche Werte können Sie folgendes annehmen:

Parameter	Wert
l_0	1,5
l_1	0,5
l_2	1,5
l_3	1,0

Den Punkt P können Sie beispielsweise wie in der obigen Skizze durch die Spitze eines gleichseitigen Dreiecks BPC definieren. Dann ist

$$P_x = B_x + l_2 \cdot \cos(\psi + 60°)$$
$$P_y = B_y + l_2 \cdot \sin(\psi + 60°)$$

Programmierung

Entwickeln Sie Ihr Programm in folgenden Schritten:

1. Definieren Sie Konstanten für die Längen l_0 bis l_3 der Gelenkstäbe.
2. Überprüfen Sie als erstes, ob die gewählten Längen der Gelenkstäbe die *Grashof'sche Regel*[1] erfüllen, also ob überhaupt ein vollständig drehbares Gelenkviereck vorliegt (Hinweis: sortieren Sie die Längen l_0 bis l_3 in einer *list*). Beenden Sie das Programm mit einer entsprechenden Fehlermeldung, falls dies nicht der Fall ist.
3. Definieren Sie Konstanten für A_x, A_y, D_x, D_y.
4. Definieren Sie Funktionen `Bx(phi)`, `By(phi)`, die für ein gegebenes φ die Koordinaten von B zurückliefern.
5. Definieren Sie Funktionen `Cx(phi, psi)`, `Cy(phi, psi)`, die für gegebene φ und ψ die Koordinaten von C zurückgeben.
6. Definieren Sie eine Funktion `Kopplung(psi, phi)` (beachten Sie die Parameter-Reihenfolge hier!), die die linke Seite der Kopplungsbedingung berechnet, d.h. 0 liefert, wenn der Punkt C im Abstand l_3 zu D liegt.

Testen Sie Ihr Programm an dieser Stelle. Messen Sie dazu in einer maßstabgerechten Skizze (s.o.) die Positionen von B und C für ein gegebenes φ und vergleichen Sie es mit den Werten, die Ihre Funktionen berechnen.

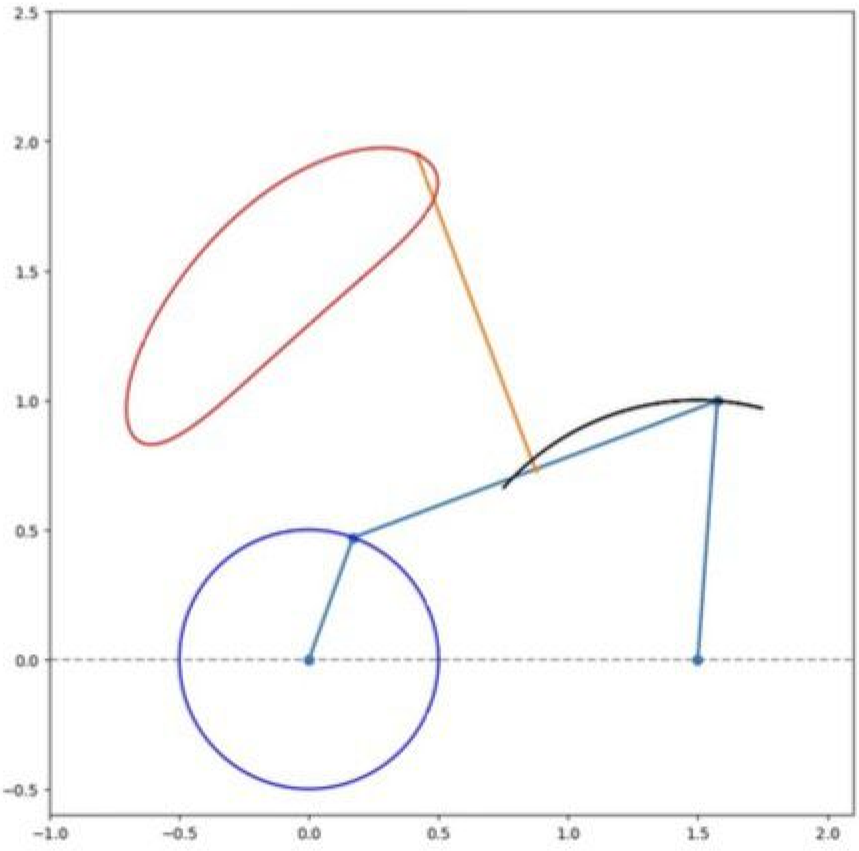
7. Definieren Sie (zunächst leere) Python-Listen px und py, die die Werte von P aufnehmen. Legen Sie zusätzlich auch Listen bx, by, cx und cy zur späteren Plot-Darstellung an.
8. Formulieren Sie eine Schleife, die die Werte für φ von 0 bis 2π durchläuft (z.B. 180 Werte, also einen Wert alle 2°).

1 s. Wikipedia: Grashofsche Regel

9. Definieren Sie eine Variable `psi_geraten`, die initial den Wert π/4 (=45°) hat.
10. Definieren Sie eine Funktion `Psi(phi, psi_geraten)`, die als Parameter φ und den aktuell geratenen Näherungswert von ψ hat. Sie wird in jedem Schleifendurchlauf (aus Schritt 8) aufgerufen und übernimmt den jeweils aktuellen Wert von φ aus der Schleife.
    Rufen Sie in dieser Funktion `scipy.optimize.root()`[2] (mit *method*='*krylov*') auf, um die Nullstelle von `Kopplung()` (aus Schritt 5) zu ermitteln. Verwenden Sie dabei als Näherungs-/Start-Wert das übergebene `psi_geraten`. Außerdem müssen Sie `root()` in `args=()` als zusätzlichen Parameter `phi` (zur Weitergabe an `Kopplung()`) übergeben. Sie erhalten im Attribut `x` des Rückgabeobjekts von `root()` einen berechneten Wert für ψ, den Sie als Ergebnis zurückgeben.
11. Aktualisieren Sie `psi_geraten` für den nächsten Schleifendurchlauf mit ψ.

Testen Sie Ihr Programm an dieser Stelle. Geben Sie in der Schleife die Werte von φ und ψ paarweise aus, und überprüfen stichprobenartig, ob die Werte plausibel sind.

12. Ergänzen Sie die Schleife, indem Sie die Werte von φ und ψ verwenden, um die Koordinaten P_x und P_y zu berechnen. Speichern Sie diese Koordinaten (per `list.append()`) am Ende der Listen px und py.
13. Verwenden Sie pyplot, um die Bahnkurve von P (rot) darzustellen. Lassen Sie zusätzlich die Bahnen von B (blau) und C (schwarz) darstellen. Zeichnen Sie die Punkte A, B, C, D einer ausgewählten Kurbel-Position (z.B. Kurbelwinkel $\varphi = 70°$) ein und verbinden diese als Polygonzug. Stellen Sie auch die sich in dieser Position ergebende Strecke PQ des Koppelauslegers dar.
    Es sollte sich damit in etwa folgende Darstellung ergeben.



[2] Dokumentation: [scipy.optimize.root()]()