

13/1 (2015), 21–39

tmcs@math.klte.hu
<http://tmcs.math.klte.hu>

Teaching
Mathematics and
Computer Science

Über die Verwendung von Maple für die Simulation von Mechanismen

ENGIN CAN

Abstract. Maple is used to do numerical computation, plot graphs and do exact symbolic manipulations and word processing. This paper demonstrates how Maple can be used for the simulation of mechanisms. This offers the possibility for students to become familiar with this particular section of mathematical modelling. The mechanism under consideration is a so-called F-mechanisms, i.e., a planar parallel 3-RRR robot with three synchronously driven cranks. It turns out that at this example it is not possible to find the poses of the moving triangle exactly by graphical methods with traditional instruments only. Hence, numerical methods are essential for the analysis of motions which can be performed by an F-mechanism.

Key words and phrases: Maple, scientific computing, mathematical modelling, planar mechanism, F-mechanism, planar parallel 3-RRR-robot.

ZDM Subject Classification: N60, P50, R20.

1. Einführung

F-Mechanismen wurden erstmals in [1] und [2] vorgestellt und analysiert. Es sind dies schnelllaufende ebene Mechanismen mit modifizierbaren Zwangläufen, die auf parallelen Robotern mit simultan angetriebenen Kurbeln basieren. Auch in [5] wird die Kinematik derartiger Roboter behandelt; die Website [3] bietet eine

Diese Arbeit wurde mit der Projektnummer 2014-17-02-001 durch den Forschungsfonds der Sakarya Universität unterstützt.

interaktiv steuerbare Simulation paralleler ebener Manipulatoren. Im folgenden wiederholen wir die charakteristischen Merkmale von F-Mechanismen:

1.1. Definition

Unter einem Fehrer-Mechanismus (kurz: F-Mechanismus) verstehen wir im folgenden eine 8-gliedrige kinematische Kette mit folgenden Merkmalen (siehe Abb.1):

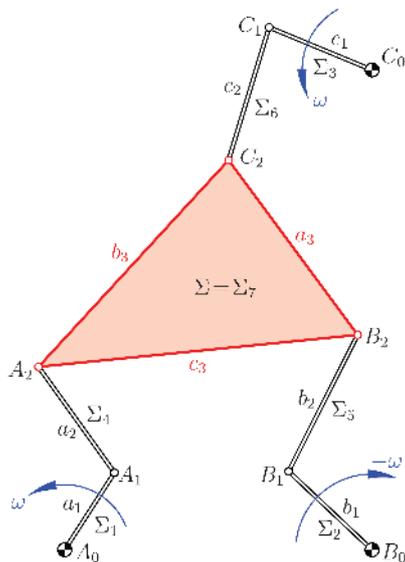


Abbildung 1. Schema eines (gegenseitigen) F-Mechanismus

- (1) Es gibt drei mit der gleichen Winkelgeschwindigkeit ω umlaufende Antriebskurbeln $\Sigma_1, \Sigma_2, \Sigma_3$ mit den Lagerpunkte A_0, B_0 bzw. C_0 und den Kurbelendpunkten A_1, B_1 bzw. C_1 . Deren Längen seien mit a_1, b_1 bzw. c_1 bezeichnet. Die Kurbeln Σ_1 und Σ_3 drehen stets in demselben (mathematisch positiven) Sinn.

Die Kurbel Σ_2 kann positiv oder negativ drehen. Bei negativem Drehsinn dieser Kurbel sprechen wir von einem gegenseitigen F-Mechanismus, sonst von einem gleichsinnigen.

- (2) Das Gangsystem $\Sigma = \Sigma_7$ enthält drei Punkte A_2, B_2, C_2 , die mit den Kurbelenden A_1, B_1 bzw. C_1 jeweils durch Stäbe Σ_4, Σ_5 bzw. Σ_6 der Längen a_2, b_2 bzw. c_2 verbunden sind.
- (3) Durch zwei kontinuierlich verstellbare Phasenverschiebungen δ_b und δ_c zwischen den Kurbeln Σ_2 und Σ_1 bzw. zwischen Σ_3 und Σ_1 sind während des Betriebes Modifikationen des Zwanglaufes Σ_7/Σ_0 möglich.

Dabei definieren wir δ_c als von der Zeit t unabhängige Differenz der Drehwinkel der Kurbeln $C_0C_1 \subset \Sigma_3$ und $A_0A_1 \subset \Sigma_1$, also $\delta_c = \varphi_{30} - \varphi_{10}$, und im gleichsinnigen Fall analog $\delta_b = \varphi_{20} - \varphi_{10}$. Im gegensinnigen Fall messen wir die Phasenverschiebung δ_b als jenen Winkel, den in der Ausgangslage $t := \varphi_{10} = 0$ die von A_0 nach rechts weisende Kurbel A_0A_1 mit der Kurbel B_0B_1 einschließt, also $\delta_b = \varphi_{20}(0) - \varphi_{10}(0)$.

Wir bezeichnen gelegentlich die Seitenlängen des Rastdreiecks mit a_0, b_0 bzw. c_0 und jene des Gangdreiecks mit a_3, b_3 bzw. c_3 .

BEMERKUNG.

- (1) Wir sprechen im folgenden vom Rast- oder Lagerdreieck $A_0B_0C_0$ und vom Gangdreieck $A_2B_2C_2$, obwohl diese Punktetripel auch kollinear sein dürfen.
- (2) Die Forderung nach gleicher Antriebsgeschwindigkeit für alle Kurbeln lässt sich natürlich auch mechanisch realisieren durch geeignete Riemen- oder Zahnradgetriebe oder im gleichsinnigen Fall auch durch Parallelkurbeln.

1.2. Formenreichtum der Zwangläufen

Die folgenden zwei Beispiele sollen die Vielfalt der durch einen F-Mechanismus erzeugbaren Zwangläufe illustrieren:

BEISPIEL 1. *Abbildung 2 zeigt einen vollständig umlaufenden F-Mechanismus zu den Abmessungen*

$$\begin{aligned} \overline{A_0B_0} &= 300, & \overline{B_0C_0} &= 371.2, & \overline{A_0C_0} &= 404.0, \\ a_1 &= 84 \text{ (177}^\circ\text{)}, & b_1 &= 104 \text{ (103}^\circ\text{)}, & c_1 &= 107 \text{ (312}^\circ\text{)}, \\ a_2 &= 185, & b_2 &= 117, & c_2 &= 185, \\ \overline{A_2B_2} &= 205, & \overline{B_2C_2} &= 250, & \overline{A_2C_2} &= 117. \end{aligned}$$

Dieses Getriebe wurde mit Hilfe der Getriebeentwurf-Software SAM 6.1 der Firma ARTAS-Engineering Software analysiert. Das der Bildschirmausgabe entnommene Bild zeigt als Resultat die Bahnkurve des Punktes 7 der Gangebene.

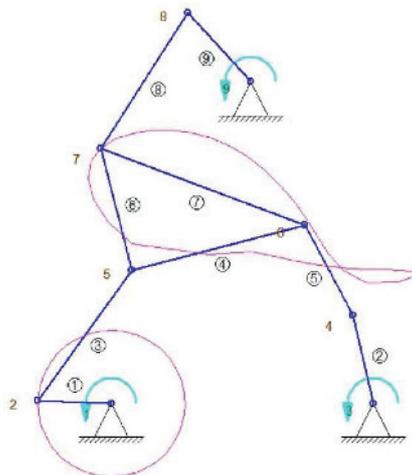


Abbildung 2. Beispiel eines gleichsinnigen F-Mechanismus (Ausgabe der Getriebeentwurf-Software SAM 6.1)

BEISPIEL 2. Allgemeiner Zwanglauf:

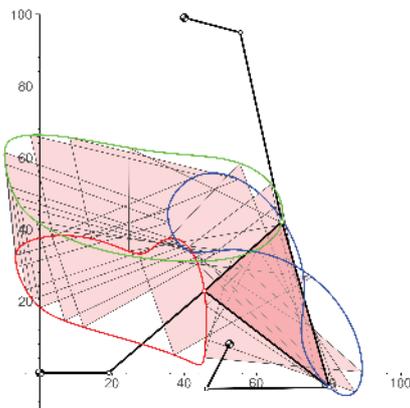


Abbildung 3. Zwanglauf des gegensinnigen F-Mechanismus aus dem Beispiel 2

Die Abbildung 3 zeigt den Zwanglauf eines gegensinnigen F-Mechanismus, welcher in der Folge mehrfach als Beispiel benutzt wird, z.B. in den Abbildungen 5, 7 und 8. Die zugehörigen Abmessungen lauten:

Rastdreieck: $A_0 = (0.0, 0.0)$, $B_0 = (52.5, 8.0)$, $C_0 = (40.0, 99.0)$,
 Kurbellängen: $a_1 = 19.0$, $b_1 = 14.0$, $c_1 = 16.0$,
 Phasenverschiebungen: $\delta_b = 243^\circ$, $\delta_c = -15^\circ$,
 Armlängen: $a_2 = 35.0$, $b_2 = 34.0$, $c_2 = 54.0$,
 Gangdreieck: $A_2 = (0.0, 0.0)$, $B_2 = (40.0, 18.0)$, $C_2 = (-7.0, 28.0)$.

2. Darstellung des Zwanglaufs Σ_7/Σ_0

Nun wenden wir uns dem Zwanglauf Σ_7/Σ_0 bei einem F-Mechanismus (siehe Abb.4) zu und diskutieren, wie wir dessen Positionen berechnen können. Die Bewegung eines beliebigen Punktes $X \in \Sigma_7$ gegenüber Σ_0 ist ansatzbar als

$$\mathbf{x}_0(t) = \mathbf{u}_0(t) + A(t)\mathbf{x}_7 \text{ mit } A(t) = \begin{pmatrix} \cos \varphi_{70} & -\sin \varphi_{70} \\ \sin \varphi_{70} & \cos \varphi_{70} \end{pmatrix} \quad (1)$$

Darin ist $\mathbf{u}_0(t)$ die im Rastkoordinatensystem (x_0, y_0) dargestellte und zunächst unbekannte Bahn des Ursprungs U des Gangkoordinatensystems (x_7, y_7) in Σ_7 . Mit $\varphi_{70}(t)$ ist der Drehwinkel des Gangkoordinatensystems gegenüber dem Rastkoordinatensystem in Σ_0 bezeichnet. \mathbf{x}_7 sind die Gangkoordinaten eines Punktes $X \in \Sigma_7$ und \mathbf{x}_0 die Rastkoordinaten des augenblicklich deckungsgleichen Punktes aus Σ_0 .

Für eine numerische Darstellung von Σ_7/Σ_0 setzen wir in der orthogonalen Matrix $A(t)$ die trigonometrischen Funktionen $\sin \varphi_{70}$ und $\cos \varphi_{70}$ in bekannter Weise als rationale Funktionen an mittels

$$f := \tan \frac{\varphi_{70}(t)}{2}$$

und daher

$$\sin \varphi_{70} = \frac{2f}{1+f^2}, \quad \cos \varphi_{70} = \frac{1-f^2}{1+f^2},$$

also

$$A(t) = \frac{1}{1+f^2} \begin{pmatrix} 1-f^2 & -2f \\ 2f & 1-f^2 \end{pmatrix}$$

mit

$$f = f(t).$$

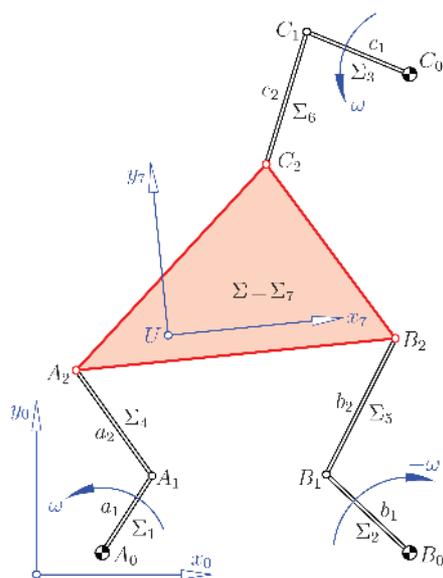


Abbildung 4. Schema eines (gegennigen) F-mechanismus

Wir bezeichnen die konstanten Gangkoordinaten der Punkte A_2 , B_2 bzw. C_2 in Σ_7 mit $\mathbf{a}_{2|7}$, $\mathbf{b}_{2|7}$ und $\mathbf{c}_{2|7}$. Damit können wir ihre Bahnen im Rastsystem darstellen als

$$\begin{aligned} \mathbf{a}_{2|0}(t) &= \mathbf{u}_0(t) + A(t)\mathbf{a}_{2|7}, \\ \mathbf{b}_{2|0}(t) &= \mathbf{u}_0(t) + A(t)\mathbf{b}_{2|7}, \\ \mathbf{c}_{2|0}(t) &= \mathbf{u}_0(t) + A(t)\mathbf{c}_{2|7}. \end{aligned}$$

Wegen der gegebenen Winkelgeschwindigkeiten der drei Antriebskurbeln sind die Bahnen der Kurbelenden A_1 , B_1 bzw. C_1 ansetzbar als

$$\mathbf{a}_{1|0}(t) = a_1 \begin{pmatrix} \cos \omega t \\ \sin \omega t \end{pmatrix}$$

$$\mathbf{b}_{1|0}(t) = b_1 \begin{pmatrix} \cos(\mp\omega t + \delta_b) \\ \sin(\mp\omega t + \delta_b) \end{pmatrix}$$

$$\mathbf{c}_{1|0}(t) = c_1 \begin{pmatrix} \cos(\omega t + \delta_c) \\ \sin(\omega t + \delta_c) \end{pmatrix}$$

mit a_1, b_1, c_1 als Kurbellängen (Abb. 4).

Die Bewegung Σ_7/Σ_0 wird nun festgelegt durch die konstanten Längen a_2, b_2 bzw. c_2 der Arme A_1A_2, B_1B_2 bzw. C_1C_2 . Dies ergibt die quadratischen Bedingungen

$$\begin{aligned} (a_{2|0}(t) - a_{1|0}(t))^2 &= a_2^2 = konst., \\ (b_{2|0}(t) - b_{1|0}(t))^2 &= b_2^2 = konst., \\ (c_{2|0}(t) - c_{1|0}(t))^2 &= c_2^2 = konst. \end{aligned} \tag{2}$$

Aus diesen drei Gleichungen müssen die Unbekannten (f, u_x, u_y) mit

$$f = \tan \frac{\varphi_{70}(t)}{2} \text{ und } \mathbf{u}_0 = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$$

berechnet werden.

Der geometrische Hintergrund dieses Eliminationsprozesses ist wie folgt: Bei vorübergehend festgehaltenem Antriebsparameter t liegt C_2 einerseits auf einem Kreis mit der Mitte C_1 , andererseits auf der Koppelkurve mit den Lagerpunkten A_1 und B_1 und der Koppel A_2B_2 . Im generischen Fall sind Koppelkurven trizirkuläre Kurven 6. Ordnung (Siehe 6, S.106). Nach dem Satz von Bézout gibt es abgesehen von den je dreifach zu zählenden absoluten Kreispunkten $6 \cdot 2 - 2 \cdot 3 = 6$ mögliche Punkte C_2 , wobei natürlich auch paarweise konjugiert Schnittpunkte auftreten können.

3. Eine Maple-Prozedur zur Berechnung des Zwangslaufs Σ_7/Σ_0

Zu Beginn werden die notwendigen Daten eingegeben, also die Koordinaten des Rastdreiecks und des Gangdreiecks sowie die Längen der Kurbeln und Arme, dann die Phasenverschiebungen samt Angabe, ob der F-Mechanismus gleichsinnig oder gegensinnig ist. Dazu kommen noch die Grenzen für den Antriebswinkel sowie die Anzahl der zu berechnenden Zwischenlagen. Aus programmiertechnischen Gründen sind einige Abweichungen von der früheren Bezeichnung unvermeidlich.

So wird z.B. der Tangens des halben Drehwinkels φ_{70} der Gangebene mit t oder t_{neu} bezeichnet, während die Antriebswinkel der einzelnen Kurbeln ff1, ff2 bzw. ff3 heißen.

```
> restart:
> with(linalg):with(plots):
-- Rastdreieck, Gangdreieck, Kurbel- und Armlängen:
> A0:=[0,0]: B0:=[52.5,8]: C0:=[40,99]:
> A2g:=[0,0]: B2g:=[40,18]: C2g:=[-7,28]:
> la1:=19: lb1:=14: lc1:=16:
> la2:=35: lb2:=34: lc2:=54:
-- Angabe, ob gleichsinnig (+1) oder gegensinnig (-1):
> omegab:=-1:
-- Phasenverschiebung in Grad bei B bzw. C:
> phasgradb:=243: phasgradc:=-15:
> phasb:=evalf(phasgradb*Pi/180): phasc:=evalf(phasgradc*Pi/180):
-- Grenzen für Antriebswinkel:
> pi:=evalf(Pi): x:=evalf(326*Pi/180):
> epsilon:=pi: mitte:=x:
> start:=mitte-epsilon: ende:=mitte+epsilon:
-- Anzahl der Zwischenlagen sowie w als Schrittweite für t:
> anz:=200:
> dif:=ende-start: w:=dif/anz:
-- Index der ersten Lösung:
> idx0:=1:
```

Nun setzen wir die Punkte A_1 , B_1 und C_1 in Abhängigkeit vom Tangens f_1 , f_2 bzw. f_3 der halben Antriebswinkel an. Daneben wird der Drehwinkel $t := \varphi_{70}$ eingeführt. Ferner führen wir den Schiebvektor $\text{trans} = (u, v) := \mathbf{u}_0$ ein sowie die orthogonale Matrix $\text{dreh} := A$ und setzen in Abhängigkeit davon die Ecken des Gangdreiecks an:

```
> Arm:=[((1-f^2)/(1+f^2)), (2*f/(1+f^2))]:
> A1:=A0+la1*(subs(f=f1, Arm)):
> B1:=B0+lb1*(subs(f=f2, Arm)):
> C1 :=C0+lc1*(subs(f=f3, Arm)):
-- Ansatz für Gangdreieck:
> trans:=[u,v]:
> co:=(1-t^2)/(1+t^2): si:=2*t/(1+t^2):
> dreh:=matrix(2,2,[co,-si,si,co]):
```

```
> A2:=simplify(matadd(evalm(dreh &* A2g),trans)):
> B2:=simplify(matadd(evalm(dreh &* B2g),trans)):
> C2:=simplify(matadd(evalm(dreh &* C2g),trans)):
```

Äquivalent zum Gleichungssystem (2) sind nun die Gleichungen $AG1 = 0$, $AG2 = 0$ und $AG3 = 0$:

```
> AG1:=numer(simplify(((A2[1]-A1[1])^2+(A2[2]-A1[2])^2)-1a2^2)):
> a1:=coeff(AG1,u^2):
> AG2:=numer(simplify(((B2[1]-B1[1])^2+(B2[2]-B1[2])^2)-1b2^2)):
> a2:=coeff(AG2,u^2):
> AG3:=numer(simplify(((C2[1]-C1[1])^2+(C2[2]-C1[2])^2)-1c2^2)):
> a3:=coeff(AG3,u^2):
```

Es folgt die Elimination von $\mathbf{u}_0^2 = u^2 + v^2$ durch Differenzenbildung. Wir berücksichtigen dabei nur die Koeffizienten a_i von u^2 . Es verbleiben zwei lineare Gleichungen eq[1] und eq[2], die wir nach u und v auflösen:

```
> eq[1]:=eval(numer(combine(a1*AG2-a2*AG1))):
> eq[2]:=eval(numer(combine(a1*AG3-a3*AG1))):
> G1:=collect(expand(eq[1]),[u,v]):
> G2:=collect(expand(eq[2]),[u,v]):
-- Aufstellung des linearen Gleichungssystems für u und v:
> P:=coeff(G1,u): Q:=coeff(G1,v):
> S:=coeff(G2,u): T:=coeff(G2,v):
> R:=-subs(u=0,v=0,G1): U:=-subs(u=0,v=0,G2):
> mat:=matrix(2,2,[P,Q,S,T]):
> vec:=[R,U]:
```

-- Lösung des linearen Gleichungssystems:

```
> loesung:=linsolve(mat,vec):
> uu:=loesung[1]:
> vv:=loesung[2]:
```

-- Diese Lösung wird in die linke Seite AG1 der ersten Gleichung eingesetzt:

```
> tt:=numer(simplify(subs(u=uu,v=vv,AG1))):
```

Hinsichtlich der Lösbarkeit dieses Gleichungssystems ist der Kommentar in [2] auf Seite 56 zu beachten.

3.1. Die Hauptschleife im Maple-Programm

Wir bezeichnen die einzelnen Werte der Antriebswinkel für die Kurbeln A_0A_1 , B_0B_1 bzw. C_0C_1 der Reihe nach mit $ff1[i]$, $ff2[i]$ bzw. $ff3[i]$, $i = 1, \dots, \text{anz}$, ($\text{anz}:=200$;) und beginnen mit der Hauptschleife des Programms:

```
> for i from 0 to anz do
> ff1[i]:=start+i*w:
> ff2[i]:=omegab*ff1[i]+phasc:
> ff3[i]:=ff1[i]+phasc:
> tt_werte[i]:=fsolve(evalf(subs(f1=tan(ff1[i]/2),f2=tan(ff2[i]/2),
    f3=tan(ff3[i]/2),tt)));
```

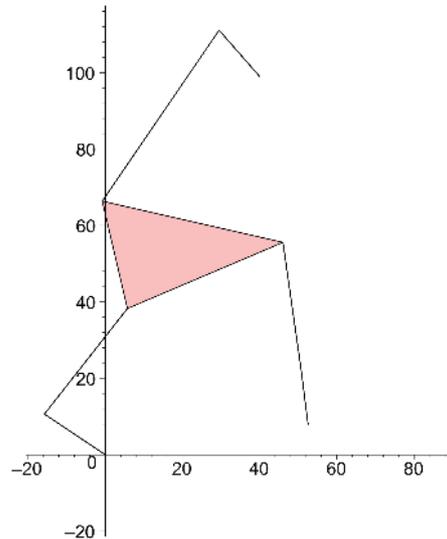


Abbildung 5. Eine mögliche Position der Gangebene Σ_4 des gegenseitigen F-Mechanismus zum Antriebswinkel $t = 146^\circ$, die Drehwinkel φ_{70} der Gangebene sind der Reihe nach $-75.688^\circ, -55.562^\circ, -18.443^\circ$ sowie -0.982° (siehe Abb.6 und Abb.7)

Bei $i = 0$ wählen wir eine der Nullstellen willkürlich aus (siehe Abb. 5).

```
> t_neu[0]:=tt_werte[0][idx0]:
```

Andernfalls fragen wir die Anzahl der Nullstellen ab mittels

```
> nbr[i]:= nops(tt_werte[i]);
```

und suchen für festes i diejenige Lösung $tt_werte[i][j]$, $1 \leq j \leq \text{nbr}[i]$, welche dem aus der vorangegangenen Position berechneten Näherungswert $\text{naeh}[i] := t_{\text{neu}}[i-1] + dt_{\text{neu}}[i-1]$ am nächsten kommt. Dies geschieht wie folgt:

```
> naeh[i]:= t_neu[i-1] + dt_neu[i-1];
> idx[i]:= 1;
> for j from 2 to nbr[i] do
  if abs(tt_werte[i][j]-naeh[i])<abs(tt_werte[i][idx[i]]-naeh[i])
  then idx[i]:= j fi
od;
```

Danach setzen wir

```
> t_neu[i]:=tt_werte[i][idx[i]]:
```

Bei der hier vorgeschlagenen Auswahl der 'richtigen' Nullstelle sind vier Umstände zu beachten:

- $tt_werte[i][j]$ sowie $t_{neu}[i]$ sind Tangenswerte von halben Drehwinkeln φ_{70} , und diese werden bei $\varphi_{70} = 180^\circ$ unendlich groß und wechseln bei einer geringfügigen Überschreitung ihr Vorzeichen. Deshalb müssen bei $tt_werte[i][j] > 1$ oder $tt_werte[i][j] < -1$ die jeweiligen Kehrwerte, also die cot-Werte bei den Größenvergleichen zur Auswahl der nächstgelegenen Lösung herangezogen werden. Die in diesem Sinn verbesserte Version des Maple-Codes lautet nun

```
> idx[i]:= 1;
> if abs(naeh[i])<2 then
  for j from 2 to nbr[i] do
    if abs(tt_werte[i][j]-naeh[i]) <
      abs(tt_werte[i][idx[i]]-naeh[i]) then idx[i]:= j fi
  od
else
  for j from 2 to nbr[i] do
    if abs((1/tt_werte[i][j])-(1/naeh[i])) <
      abs((1/tt_werte[i][idx[i]])-(1/naeh[i]))
    then idx[i]:= j fi
  od
fi;
> t_neu[i]:=tt_werte[i][idx[i]];
```

- Ist man exakt in einer singulären Lage[2], so scheitert die nachstehend gezeigte Geschwindigkeitsanalyse wegen der verschwindenden Koeffizientendeterminante des zugehörigen linearen Gleichungssystems [2].

- Ist der verfolgte Zwanglauf nicht umlauffähig, so wachsen in der Nähe des Kurbelstillstandes — wenn also die Position von Σ_7 singular, aber nicht zweifach singular wird [2] — die Geschwindigkeiten über alle Grenzen. Daher wird die über die Geschwindigkeitsanalyse berechnete Näherungslösung keinen guten Vergleich ermöglichen.
- Wenn die Anzahl der Nullstellen zwischen zwei Schritten wechselt — was natürlich bedeutet, dass einer der Zwangläufe nicht umlauffähig ist —, dann wird das Programm zu einem anderen Zwanglauf wechseln, was sich in einem “Sprung” bei der Animation äußert. Bei der Getriebeanalyse mit der Getriebesoftware SAM bleibt der Mechanismus an derartigen Stellen überhaupt stehen — wie übrigens gar nicht so selten ohne erkennbaren Grund auch bei anderen Stellen, z.B., dann, wenn eine Kurbel und mit dem anschließenden Arm in einer Strecklage ist.

Als letzte Aktion innerhalb der Schleife berechnen wir einen Näherungswert für den Drehwinkel der nächsten Position. Dazu führen wir eine *Geschwindigkeitsanalyse* gemäß [2] durch:

```

> A1:=A0+la1*(subs(f=tan(ff1[i]/2),Arm));
> B1:=B0+lb1*(subs(f=tan(ff2[i]/2),Arm));
> C1:=C0+lc1*(subs(f=tan(ff3[i]/2),Arm));
> d1:=A1-A0: d2:=B1-B0: d3:=C1-C0:
-- Geschwindigkeitsvektoren der Kurbelenden:
> vA1:=w*[-d1[2],d1[1]]:
> vB1:=w*omegab*[-d2[2],d2[1]]:
> vC1:=w*[-d3[2],d3[1]]:
-- Ansatz für Geschwindigkeitsverteilung der Gangebene:
> X[i]:=[x1[i],x2[i]]:
> Y[i]:=matrix(2,2,[0,-x3[i],x3[i],0]):
-- Aufstellung des Gleichungssystems [1]:
> uu_werte:=evalf(subs(f1=tan(ff1[i]/2),f2=tan(ff2[i]/2),
    f3=tan(ff3[i]/2),t=t_neu[i],uu));
> vv_werte:=evalf(subs(f1=tan(ff1[i]/2),f2=tan(ff2[i]/2),
    f3=tan(ff3[i]/2),t=t_neu[i],vv));
> transm:=[uu_werte,vv_werte]:
> c2:=evalf(subs(t=t_neu[i],co)):
> s2:=evalf(subs(t=t_neu[i],si)):
> drehm:=matrix(2,2,[c2,-s2,s2,c2]):
> A2:=convert(evalf((matadd(evalm(drehm &* A2g),transm))),list):

```

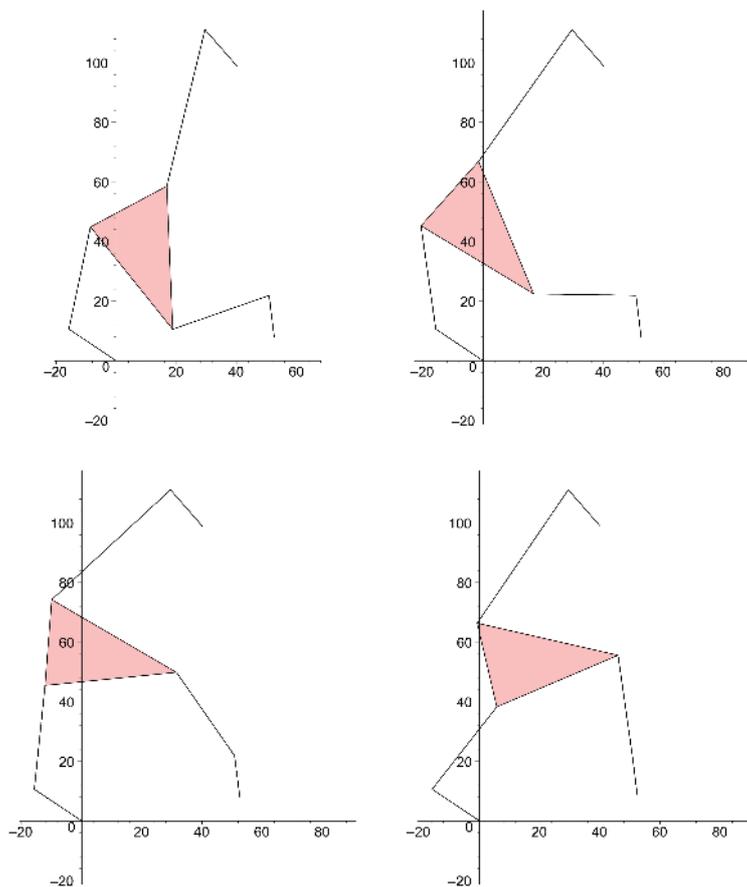


Abbildung 6. Vier verschiedene Positionen der Gangebene der Abb. 5

```

> B2:=convert(evalf((matadd(evalm(drehm &* B2g),transm))),list):
> C2:=convert(evalf((matadd(evalm(drehm &* C2g),transm))),list):
> vA2:=simplify(matadd(evalm(Y[i] &* A2),X[i])):
> vB2:=simplify(matadd(evalm(Y[i] &* B2),X[i])):
> vC2:=simplify(matadd(evalm(Y[i] &* C2),X[i])):
> V11:=matadd(vA2,-vA1): V12:=matadd(A2,-A1):
> V21:=matadd(vB2,-vB1): V22:=matadd(B2,-B1):
> V31:=matadd(vC2,-vC1): V32:=matadd(C2,-C1):
> GL1:=simplify(innerprod(V11,V12));
    
```

```

> GL2:=simplify(innerprod(V21,V22));
> GL3:=simplify(innerprod(V31,V32));
-- Koeffizientenmatrix und Absolutspalte des Gleichungssystems:
> P1:=coeff(GL1,x1[i]): P2:=coeff(GL2,x1[i]): P3:=coeff(GL3,x1[i]):
> Q1:=coeff(GL1,x2[i]): Q2:=coeff(GL2,x2[i]): Q3:=coeff(GL3,x2[i]):
> R1:=coeff(GL1,x3[i]): R2:=coeff(GL2,x3[i]): R3:=coeff(GL3,x3[i]):
> T1:=-subs(x1[i]=0,x2[i]=0,x3[i]=0,GL1):
> T2:=-subs(x1[i]=0,x2[i]=0,x3[i]=0,GL2):
> T3:=-subs(x1[i]=0,x2[i]=0,x3[i]=0,GL3):
> mat:=matrix(3,3,[P1,Q1,R1,P2,Q2,R2,P3,Q3,R3]):
> vec:=[T1,T2,T3]:
-- Bestimmung der Lösung:
> losung:=linsolve(mat,vec):
> x1[i]:= losung[1]: x2[i]:= losung[2]:
> x3[i]:= losung[3]:
> dt_neu[i]:= eval((1+t_neu[i]*t_neu[i])*x3[i]/2):
> od:

```

Hinter der vorletzten Programmzeile steht die folgende Überlegung:

Wir suchen eine Näherungswert für $t_{\text{neu}}[i+1]$ in der Form $t_{\text{neu}}[i] + dt_{\text{neu}}[i]$ nach dem Muster

$$f(x+h) \sim f(x) + hf'(x)$$

mit x als Antriebswinkel t und mit h als Schrittweite w , denn w war die ‘Winkelgeschwindigkeit’ der Antriebskurbeln im obigen Programmcode bei der Festsetzung der Geschwindigkeitsvektoren vA_1, \dots der Kurbelenden A_1, \dots

Die Funktion f ist $\tan \frac{\varphi}{2}$ mit $\varphi = \varphi_{70}(t)$, und dabei ist $h\dot{\varphi}_{70}$ gleich der Lösung x_3 in dem linearen Gleichungssystem zur Geschwindigkeitsanalyse. Daher ist

$$dt_{\text{neu}}[i] = h \frac{df}{dt} = h \frac{d}{dt} \left(\tan \frac{\varphi}{2} \right) = \left(1 + \tan^2 \frac{\varphi}{2} \right) \frac{h \dot{\varphi}}{2} = \left(1 + (t_{\text{neu}}[i])^2 \right) \frac{x_3[i]}{2}.$$

3.2. Verschiedene Ausgabemöglichkeiten

Welche Graphikausgabe auch immer gewünscht wird, stets beginnt man mit der Berechnung der einzelnen Positionen, und zwar derjenigen zu den vorhin gespeicherten Drehwinkeln $t_{\text{neu}}[i]$:

-- Berechnung der Positionen:

```

> for i from 0 to anz do

```

```

> AA1[i]:=A0+la1*(subs(f=tan(ff1[i]/2),Arm));
> BB1[i]:=B0+lb1*(subs(f=tan(ff2[i]/2),Arm));
> CC1[i]:=C0+lc1*(subs(f=tan(ff3[i]/2),Arm));
> uu_werte:=evalf(subs(f1=tan(ff1[i]/2),f2=tan(ff2[i]/2),
    f3=tan(ff3[i]/2),t=t_neu[i],uu));
> vv_werte:=evalf(subs(f1=tan(ff1[i]/2),f2=tan(ff2[i]/2),
    f3=tan(ff3[i]/2),t=t_neu[i],vv));
-- - Sollte die Koeffizientenmatrix des zugehörigen Gleichungssystems 0 sein[1], so
kommt es hier zu einer Division durch 0:
> trans2:=[uu_werte,vv_werte];
> c2:=evalf(subs(t=t_neu[i],co));
> s2:=evalf(subs(t=t_neu[i],si));
> dreh2:=matrix(2,2,[c2,-s2,s2,c2]);
> AA2[i]:=convert(evalf((matadd(evalm(dreh2&*A2g),trans2))),list):
> BB2[i]:=convert(evalf((matadd(evalm(dreh2&*B2g),trans2))),list):
> CC2[i]:=convert(evalf((matadd(evalm(dreh2&*C2g),trans2))),list):
-- - Die Schleife ist noch nicht abgeschlossen!
    Soll mit Maple eine Animation des Zwanglaufes erzeugt werden, muss man
nach der Berechnung der Positionen innerhalb der Schleife wie folgt vorgehen:
    -- - Es werden die Elemente der Animation definiert:
> Dreieck[i]:=polygonplot([AA2[i],BB2[i],CC2[i]],color=red):
> Gelenk_a01[i]:=polygonplot([A0,AA1[i]],thickness=2):
> Gelenk_a12[i]:=polygonplot([AA1[i],AA2[i]],thickness=2):
> Gelenk_b01[i]:=polygonplot([B0,BB1[i]],thickness=2):
> Gelenk_b12[i]:=polygonplot([BB1[i],BB2[i]],thickness=2):
> Gelenk_c01[i]:=polygonplot([C0,CC1[i]],thickness=2):
> Gelenk_c12[i]:=polygonplot([CC1[i],CC2[i]],thickness=2):
> od:
-- - Ende der Schleife:
> Konfiguration:=seq(display(Dreieck[i],Gelenk_a01[i],Gelenk_b01[i],
    Gelenk_c01[i],Gelenk_a12[i],Gelenk_b12[i],Gelenk_c12[i]),
    i=1..anz):
-- - Befehl zur Ausgabe der Animation:
> display(Konfiguration,scaling=constrained,insequence=true);

```

Die hier erfolgte Trennung der Streckenzüge $A_0A_1A_2$ usw. in jeweils zwei Zeilen hat den alleinigen Grund darin, dass die verfügbare Version von Maple hier zusätzliche Geraden einfügt, die dann den Betrachter eher verwirren. Der Befehl *'scaling=constrained'* garantiert ein maßstabsgetreues Bild.

Eine Darstellung der *Punktbahnen* (Abb. 3) gelingt bei der verwendeten Maple-Version (Maple 13) durch folgenden Trick: Wir speichern bei der Berechnung der Ecken $A_2[i], \dots$ des Gangdreiecks immer auch den jeweiligen Vorgänger mit ab und lassen anschließend die Verbindungssehnen zeichnen.

```
> AA2[i]:=convert(evalf((matadd(evalm(dreh2&*A2g),trans2))),list):
  DD2[(i+1)]:=AA2[i]:
> BB2[i]:=convert(evalf((matadd(evalm(dreh2&*B2g),trans2))),list):
  EE2[(i+1)]:=BB2[i]:
> CC2[i]:=convert(evalf((matadd(evalm(dreh2&*C2g),trans2))),list):
  FF2[(i+1)]:=CC2[i]:
-- Befehle zur Ausgabe der Plots:
> for i from 1 to anz do
  Bahn_a[i]:=polygonplot([DD2[i],AA2[i]],thickness=1,color=red):
  Bahn_b[i]:=polygonplot([EE2[i],BB2[i]],thickness=1,color=blue):
  Bahn_c[i]:=polygonplot([FF2[i],CC2[i]],thickness=1,color=green):
od:
> display({seq(Bahn_a[i],i=1..anz)},scaling=constrained,
  view = [-10..100,-10..100]);
> display({seq(Bahn_b[i],i=1..anz)},scaling=constrained,
  view = [-10..100,-10..100]);
> display({seq(Bahn_c[i],i=1..anz)},scaling=constrained,
  view = [-10..100,-10..100]);
-- Zur Darstellung einzelner Positionen des Gangdreiecks:
> display({seq(Dreieck[20*i],i=1..(anz/20))},scaling=constrained,
  view = [-10..100,-10..100]);
```

Auch hier wurden die Bahnen getrennt voneinander ausgegeben, um zu vermeiden, dass Maple zusätzliche Streckenbilder einfügt.

Will man eine Übersicht darüber haben, zu welchen Antriebswinkeln (hier mit $ff1[i]$ bezeichnet) wieviele Positionen der Gangebene Σ_7 , also wieviele Drehwinkel φ_{70} möglich sind, kann man die zu diskreten f_{f_1} -Werten gehörigen φ_{70} -Wert (hier $tt_werte[i][j]$) — jeweils im Gradmaß — in einem Diagramm zeichnen lassen (siehe Abb.7). Zuvor muss man allerdings $ff1$ auf das Intervall $[0, 2\pi]$ normieren:

```
> for i from 0 to anz do
  prov:=start+i*dif/anz:
  if prov>(2*pi)
  then ff1[i]:=prov-(2*pi) elif prov<0
  then ff1[i]:=prov+(2*pi)
```

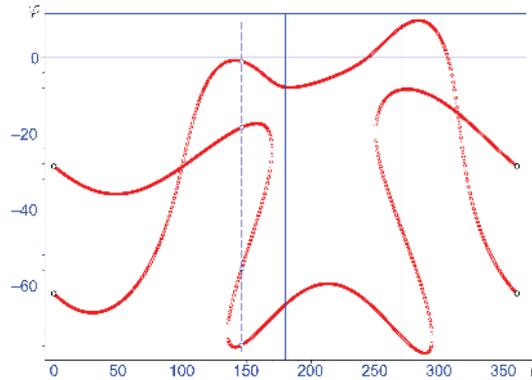


Abbildung 7. Drehwinkel φ_{70} als Funktion des Antriebswinkels t zu den Daten des gegenseitigen F-Mechanismus aus dem Beispiel 2 (siehe auch Abb. 6). Dort werden die vier Positionen des Gangdreiecks zum Antriebswinkel $t = 146^\circ$ gezeigt.

```

else ff1[i]:=prov
end if;
od:

```

Es war bei Maple 13 nicht möglich, das Diagramm direkt zeichnen zu lassen, wenn zu verschiedenen t -Werten unterschiedliche Lösungsanzahlen auftreten. Deshalb wurde die Maximalanzahl loesg der jeweils auftretenden Lösungen ermittelt, und dort, wo die tatsächlich Lösungsanzahl geringer war, wurde das Array der φ_{70} -Werte mit Nullen aufgefüllt:

```

> grd:=180/pi;
> for i from 1 to anz do
  AA1[i]:=A0+la1*(subs(f=tan(ff1[i]/2),Arm)):
  BB1[i]:=B0+lb1*(subs(f=tan(ff2[i]/2),Arm)):
  CC1[i]:=C0+lc1*(subs(f=tan(ff3[i]/2),Arm)):
  for j from 1 to nbr[i] do
    XX[i,j]:= [ff1[i]*grd,arctan(tt_werte[i][j])*2*grd];
  od;
  if nbr[i]<loesg then
  for j from (nbr[i]+1) to loesg do
    XX[i,j]:= [0,0];
  od
  end if;

```

```

od:
-- Wir wandeln das zweidimensionale Array der XX in ein eindimensionales:
> for i from 1 to anz do
  for j from 1 to loesg do
    YY[(i-1)*loesg+j] := XX[i,j];
  od
od;
> pointplot({seq(YY[i],i=1..(anz*loesg))},color=red,axes=boxed,
  symbol=CIRCLE,symbolsize=8);

```

Die Diagrammpunkte mit vertikalen Tangenten gehören zu Umkehrlagen der Antriebskurbeln, sofern nicht die Tangente die Diagrammkurve hier durchsetzt. Die Doppelpunkte (Selbstschnitte) der Diagrammkurve bestimmen Verzweigungslagen des F-Mechanismus, sofern nicht an dieser Stelle die Koeffizientendeterminante des linearen Gleichungssystems (2) für $\mathbf{u}_0 = (u_x, u_y)$ verschwindet.

Um so wie in Abb. 8 den Verlauf aller möglichen Positionen der Ecken des Gangdreiecks darstellen zu lassen, gibt es folgende Möglichkeit:

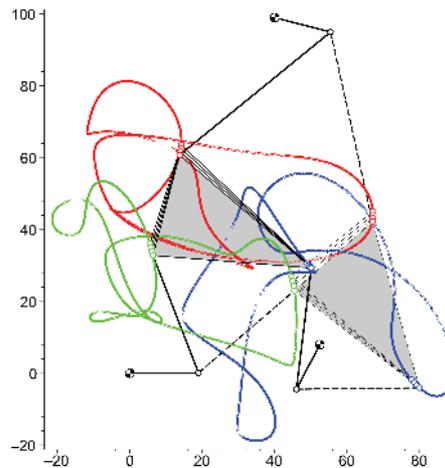


Abbildung 8. Bahnen der Ecken des Gangdreiecks bei den Positionen des gegensinnigen F-Mechanismus aus Beispiel 2 (siehe auch Abb.7). Die zum Antriebswinkel 0.5° gehörigen beiden Ausgangslagen sind besonders hervorgehoben. Die strichlierte Lage gehört zu dem umlauffähigen Zwanglauf, der auch in Abb.3 dargestellt ist.

```

> aa:=pointplot({seq(AA[i],i=1..(anz*loesg))}, color=blue, axes=boxed,
  symbol=CIRCLE, symbolsize=6,scaling=constrained):

```

```
bb:=pointplot({seq(BB[i],i=1..(anz*loesg))}, color=red, axes=boxed,
  symbol=CIRCLE, symbolsize=6,scaling=constrained):
cc:=pointplot({seq(CC[i],i=1..(anz*loesg))}, color=green, axes=boxed,
  symbol=CIRCLE, symbolsize=6,scaling=constrained):
> display(aa,bb,cc);
```

Die Abbildung 8 zeigt ein Beispiel dazu.

Zu einer Auswahl der jeweiligen Zusammenhangskomponenten und damit zum Zeichnen der eigentlichen Bahnkurven (so wie in Abb.3) kommt man nur, wenn man bei der numerischen Berechnung der nächsten Position die Geschwindigkeitsanalyse aus der vorhergehenden Lage heranzieht.

Literatur

- [1] E. Can and H. Stachel, A planar parallel 3-RRR robot with synchronously driven cranks, *Mech. and Mach. Theory* **79** (2014), 29–45.
- [2] E. Can, *Analyse and Synthese eines schnelllaufenden ebenen Mechanismus mit modifizierbaren Zwangsläufen*, PhD Thesis, Vienna University of Technology, 2012.
- [3] S. Ahamed, *Simulation of a 3-DOF 3-RRR Planar Parallel Robot*, www.parallemic.org/Java/3RRR.html.
- [4] W. Gander and J. Hrebicek, *Solving Problems in Scientific Computing Using Maple and Matlab*, Heidelberg Springer, Berlin, 2004.
- [5] S. Staicu, Kinematics of the 3-RRR planar parallel robot, *U.P.B. Sci. Bull.* **70**, no. 2, Ser. D (2008), 3–14.
- [6] W. Wunderlich, Ebene Kinematik, *BI-Hochschultaschenbücher, Bd. 447*, Bibliographisches Institut, Mannheim (1970).

ENGIN CAN
 KAYNARCA SCHOOL OF APPLIED SCIENCES
 SAKARYA UNIVERSITY
 TR54650 SAKARYA
 TURKEY

E-mail: ecan@sakarya.edu.tr

(Received June, 2014)