Prof. Dr. K. Wüst
Fachbereich MNI

SS 2010
FH Gießen-Friedberg

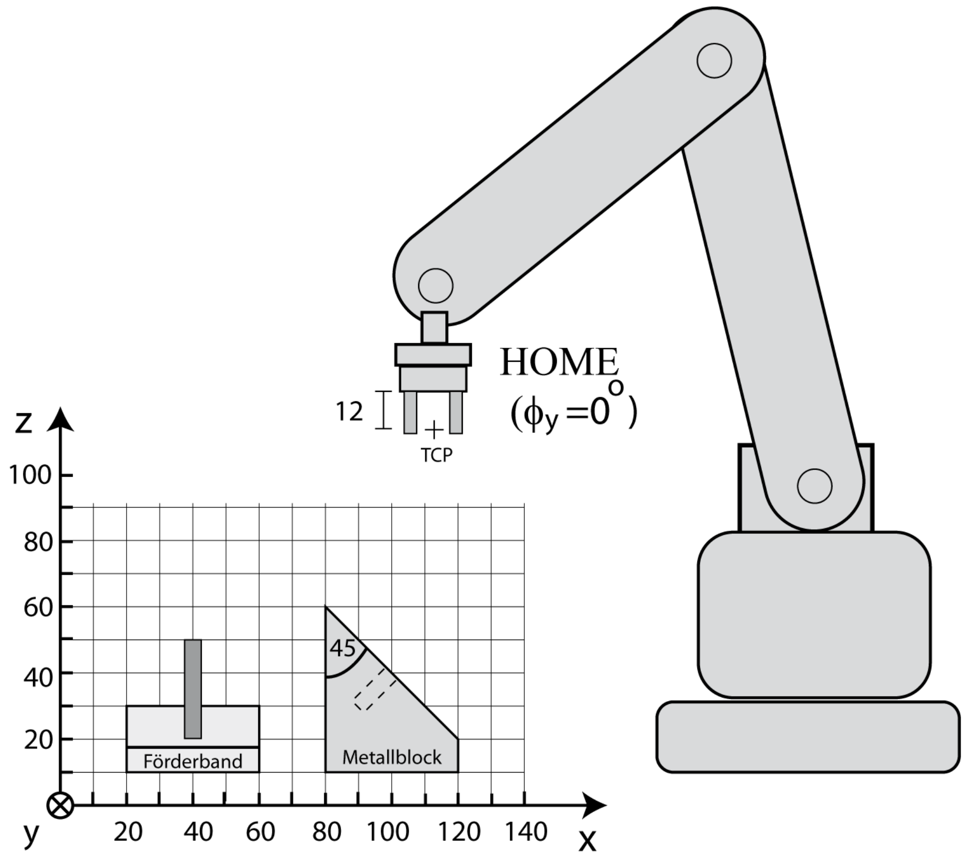
Klausur Robotik/Steuerungstechnik 1.7.2010

Nachname:
Vorname:
Matrikelnummer:

Aufgabe	Punkte	erreicht
1	30	
2	30	
3	40	
4	20	
Summe	120	

Mit Lösungen

Aufg.1)



Ein sechachsiger Industrieroboter soll einen Bolzen vom Band entnehmen und in eine Sackbohrung an einem Metallblock einsetzen. Wenn der Bolzen weit genug in die Bohrung geführt ist, gleitet er weiter bis zum Ende. Formulieren Sie ein Programm in Form einer Anweisungsliste für den Roboter. benutzen Sie eine Pseudo-Robotersprache. Gehen Sie davon aus, dass der Roboter zu Beginn in der Homeposition steht und fahren Sie ihn am Ende wieder in die HOME-Position. Der TCP liegt genau zwischen den Backen des Greifers. (siehe Bild) Für die Stopp-Position des Förderbandes und die Bohrung ist y=0. Geben Sie zu allen Befehlen und zu allen Bahnpunkten alle notwendigen Parameter an. Die Punkte können in einer separaten Punktliste beschrieben werden.

Zusatzfrage: Welche Koordinatensysteme sollten beim Einteachen benutzt werden?

Lösung Aufgabe 1

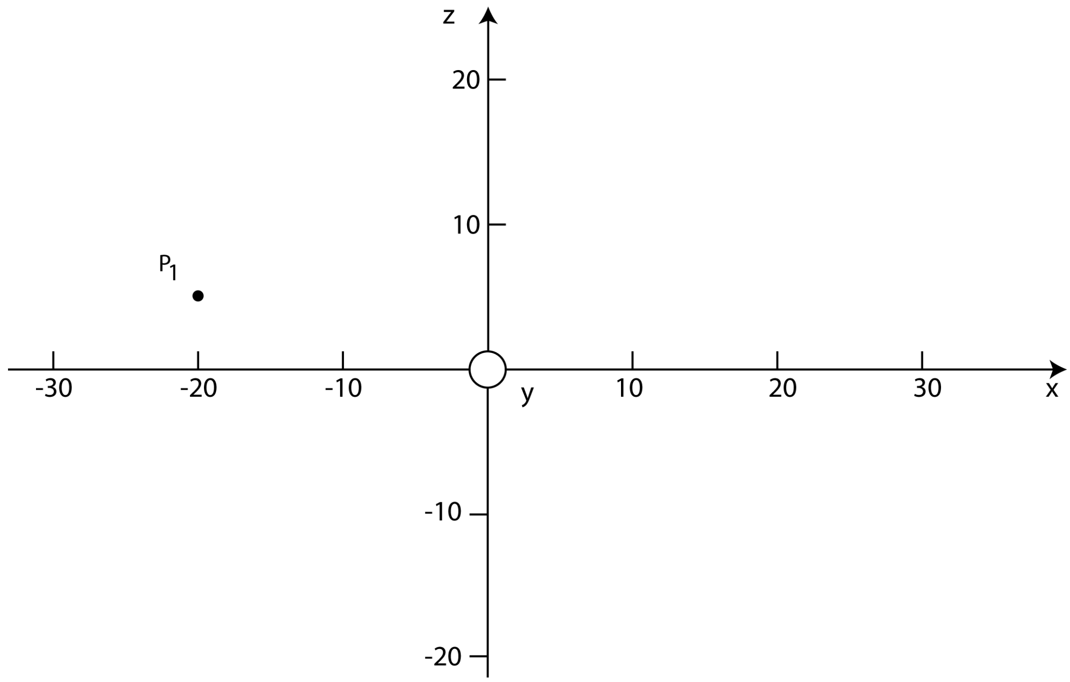
Wie immer muss auf die richtige Bewegungsart geachtet werden, in der Nähe des Werkobjekts oder innnerhalb der Bohrung nur CP linear. Der Bolzen muss weit genug aus der Öffnung gezogen werden, bevor von der Geraden Bahn abgewichen wird. Ebenfalls muss er beim ablegen ein Stück weit eingeführt werden, bevor der Greifer geöffnet wird. Punktabzug gibt es auch für jede unerwünschte Berührung. Der Vorpunkt vor dem Metallblock muss genau auf der Verlängerung der Bohrung liegen (45 Grad-Linie) sonst gibt es Verklemmungen. Alle Bewegungen mit Objekt im Greifer müssen langsam erfolgen (v10 – v30). Folgende Werte sind immer gleich 0: y, ϕ_x, ϕ_z

```
Greifer auf ; Falls noch zu
MoveJ VP1, vmax, z3 ; PTP
MoveL P10, v20, fine ; CP-linear
Greifer zu MoveL VP20, v10, z5 ; langsam
herausziehen
MoveJ UP10, v30, z5 ; Kollision vermeiden
MoveJ VP30, v30, z2 ; Vorpunkt vor Öffnung
MoveL P20, v10, fine ; langsam einführen
Greifer auf MoveL VP30, v20, z3 ; Greifer
zurückziehen
MoveJ HOME, vmax, z0 ; Homeposition anfahren
```

Punkt	x	z	ϕ_y
VP10	40	60	0
P10	40	40	0
VP20	40	70	0
UP10	80	90	-22
VP30	120	60	-45
P20	110	50	-45

Beim Einteachen können die im Bild dargestellten Koordinaten (Welt- oder Objektkoordinaten benutzt werden; für die Bewegungen an und in der Bohrung (45 Grad) können (falls vorhanden) Werkobjektkoordinaten benutzt werden, sonst hilfsweise Werkzeugkoordinaten.

Aufg.2)



Ein Punkt P_1 befindet sich in Position (-20, 30, 10).

a) Markieren Sie in der oben stehenden Zeichnung die Richtung der y-Achse.

b) Geben Sie eine homogene Transformationsmatrix an, die eine Verschiebung (Translation) um den Vektor (15, -10, -5) bewirkt.

c) Transformieren Sie damit den Punkt P_1 in Punkt P_2.

d) Geben Sie eine homogene Transformationsmatrix an, die eine Drehung um die y-Achse um -90^o bewirkt.

e) Transformieren Sie damit den Punkt P_2 in Punkt P_3.

f) Geben Sie eine homogene Transformationsmatrix an, die die beiden vorhergehenden Transformationen in einem Schritt ausführt, also Punkt P_1 direkt in Punkt P_3 transformiert. Reihenfolge: Zunächst Verschiebung, dann Rotation.

g) Geben Sie eine homogene Transformation an, die genau die entgegengesetzte Transformation bewirkt, also Punkt P_3 wieder in einem Schritt in Punkt P_1 transformiert.

h) Zeichnen Sie die Punkte P_2, P_3 in der Abbildung ein. Zeichnen Sie auch die Transformationen symbolisch als gebogene Pfeile mit Namen zwischen den Punkten ein.

Lösung Aufgabe 2

a) da wir nur Rechtssysteme verwenden, muss die z-Achse aus der Papierebene zeigen und wird mit einem Punkt gekennzeichnet.

b)

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

c)

$$P_2 = T_1 P_1 = \begin{pmatrix} -5 \\ 20 \\ 5 \\ 1 \end{pmatrix} \tag{2}$$

d)

$$T_2 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

e)

$$P_3 = T_2 P_2 = \begin{pmatrix} -5 \\ 20 \\ -5 \\ 1 \end{pmatrix} \tag{4}$$

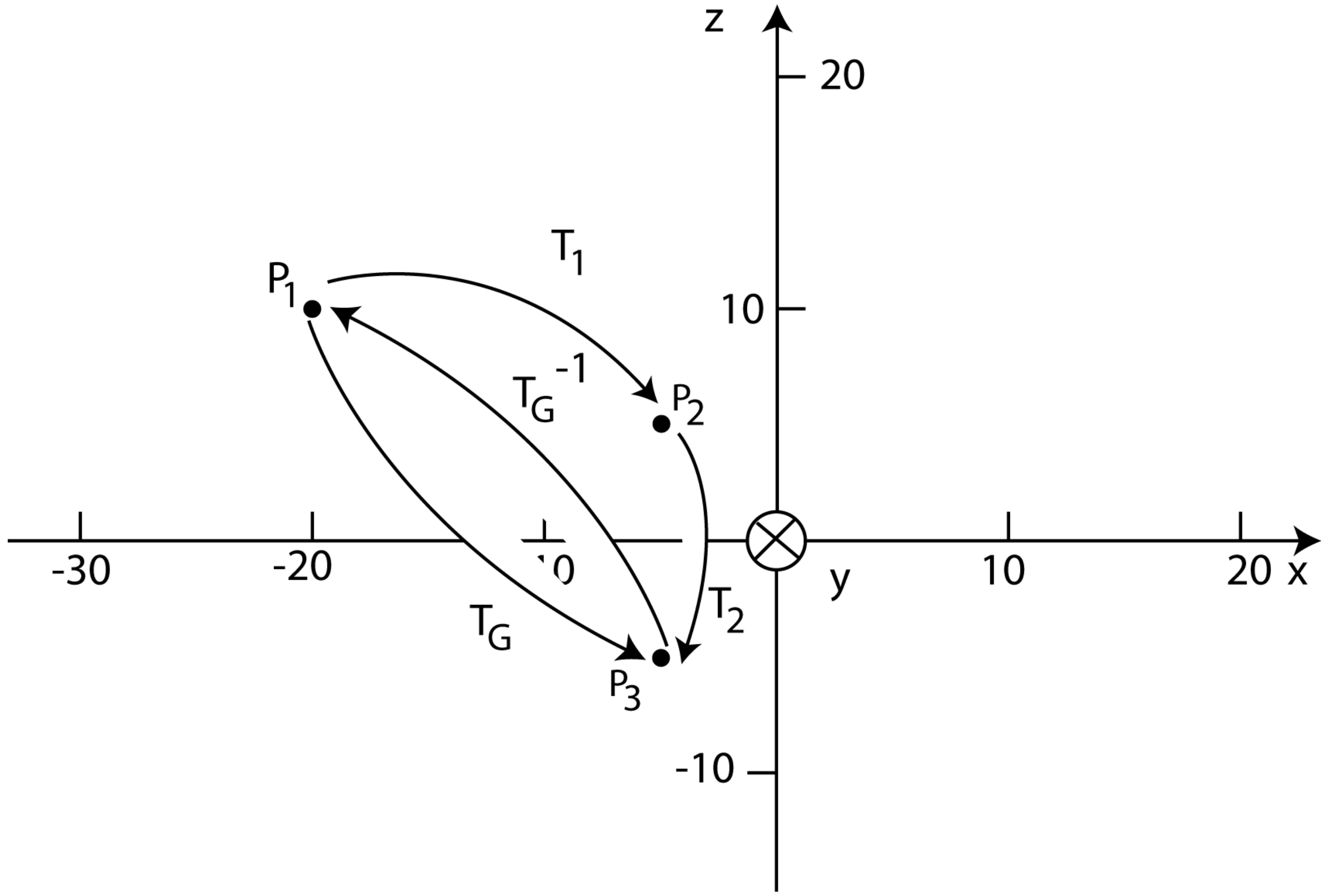
f)

$$T_2 T_1 = T_G = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 5 \\ 0 & 1 & 0 & -10 \\ 1 & 0 & 0 & 15 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Probe (freiwillig): $T_G P_1 = \begin{pmatrix} -5 \\ 20 \\ -5 \\ 1 \end{pmatrix}$

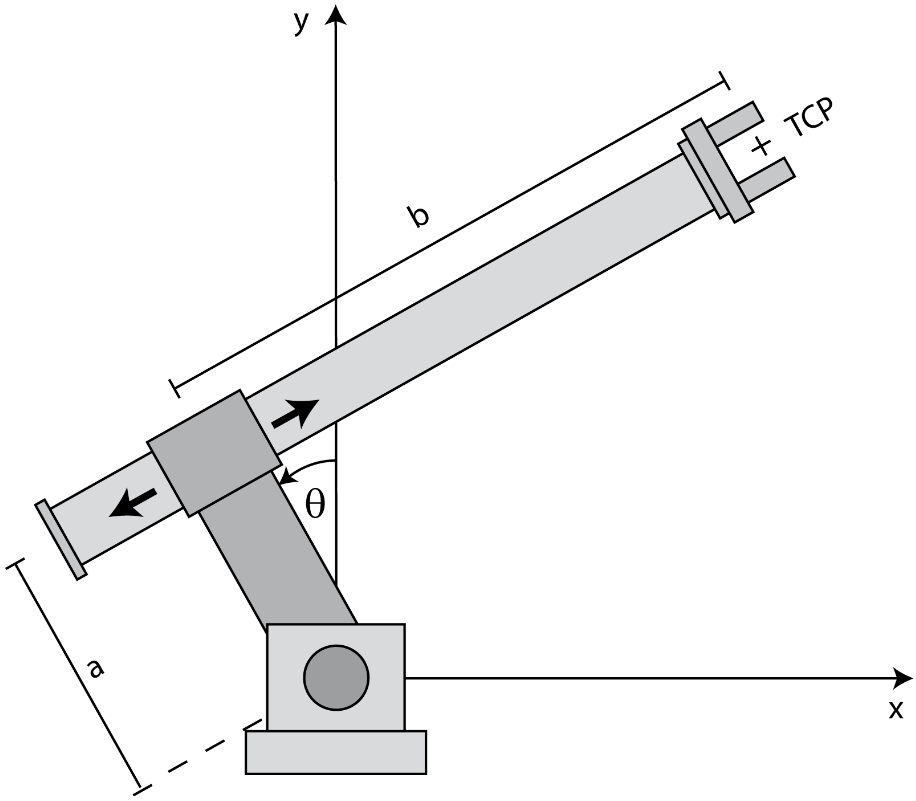
g)

$$T_G^{-1} = \begin{pmatrix} 0 & 0 & 1 & -15 \\ 0 & 1 & 0 & 10 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$



z
20
T_1
P_1
10
-1
T_G
P_2
-30
-20
-10
y
10
20 x
T_2
T_G
P_3
-10

Aufg.3) Ein Roboter hat den in der Abbildung gezeigten Aufbau: In einer Schwenkeinheit befindet sich eine Translationsachse (Linearachse), am Ende der Translationsachse befindet sich das Tool.



a) Bezeichnen Sie die Armteile mit AT0, AT1 usw.

b) Wie viele Freiheitsgrade (Degrees of Freedom) hat der Roboter? Nennen Sie die Gelenkkoordinaten (Achskoordinaten, Maschinenkoordinaten) und die kartesischen Koordinaten:

c) Geben Sie die Gleichungen der kinematischen Vorwärtstransformation für Position und Orientierung des TCP an.

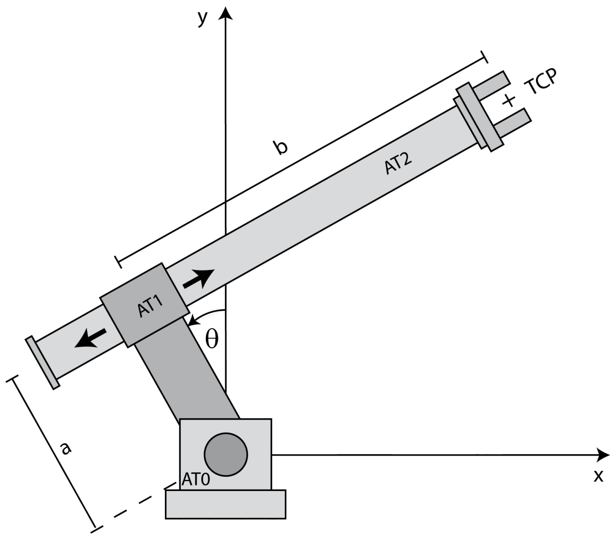
d)Wo befindet sich der TCP, wenn $a = 100\ mm, b = 200\ mm$ ist und $\theta = 30^o$ und wie ist seine Orientierung? (Bitte einzeichnen!)

e) Stellen Sie die Jacobi-Matrix für diesen Roboter auf. Wie ist der Geschwindigkeitsvektor des TCP in der Stellung $a = 100\ mm, b = 100\ mm$ ist und $\theta = 90^o$, wenn $\dot{\theta} = 0.2\ s^{-1}$ und $\dot{b} = 30\ mm/s$?

f) Welche singulären Stellungen gibt es bei dieser Geometrie? Begründen Sie Ihre Aussage rechnerisch oder logisch! Können diese Singularitäten bei dem im Bild dargestellten Roboter in der Praxis auftreten?

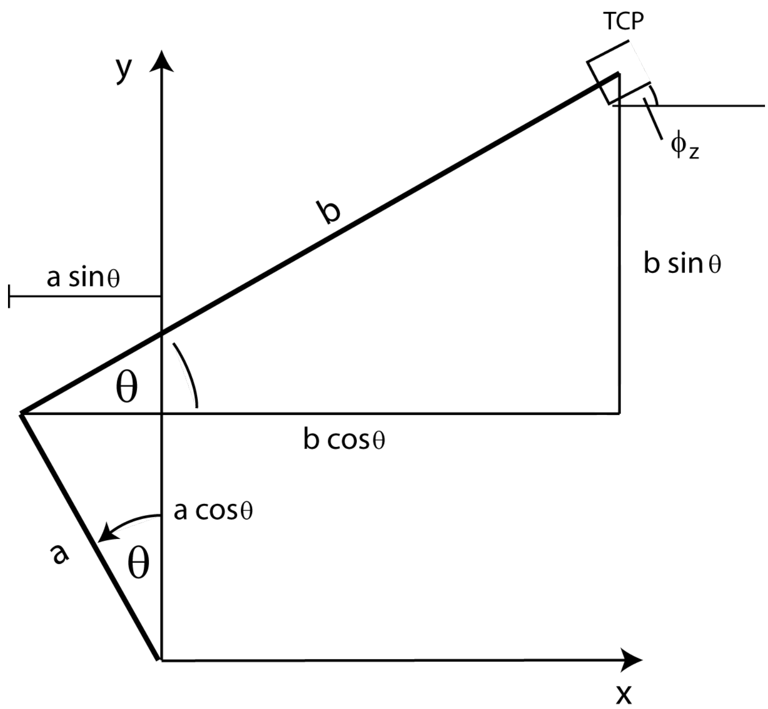
Lösung Aufgabe 3

a)



b) Der Roboter hat 2 Gelenke und somit 2 Freiheitsgrade. Die Gelenkkoordinaten sind θ, b
Die kartesischen Koordinaten sind x, y; ϕ_z ; von diesen drei kartesischen Koordinaten sind nur zwei frei wählbar, da nur zwei Gelenke vorhanden sind. Dazu kommen z, ϕ_x, ϕ_y die immer gleich Null sind.

c) Der Roboter kann im Linienbild so dargestellt werden:



Daraus liest man ab:

$$\begin{aligned} x &= b\cos\theta - a\sin\theta \\ y &= b\sin\theta + a\cos\theta \\ \phi_z &= \alpha \end{aligned}$$

d) Für das konkrete Beispiel $a = 100\ mm, b = 200\ mm$ und $\theta = 30^o$ befindet sich der TCP bei:

$$\begin{aligned} x &= 200\,mm\cos() - 100\,mm\sin(30^o) = 123.2\,mm \\ y &= 200\,mm\sin(30^o) + 100\,mm\cos(30^o) = 186.6\,mm \\ \phi_z &= 30^o \end{aligned}$$

e) Wenn die Vorwärtstransformation ergibt $x = f_1(\theta_1, \theta_2), y = f_2(\theta_1, \theta_2)$, wobei θ_1, θ_2 die Gelenkkoordinaten (Gelenkwinkel) sind, ist die Jakobi-Matrix in diesem konkreten Fall:

$$J_A = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial b} \\ \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial b} \end{pmatrix} = \begin{pmatrix} (-b\sin\theta - a\cos\theta) & \cos\theta \\ (b\cos\theta - a\sin\theta) & \sin\theta \end{pmatrix}$$

Die Geschwindigkeit des TCP ergibt sich aus den Gelenkgeschwindigkeiten mit J_A:

$$\dot{\vec{v}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = J_A \begin{pmatrix} \dot{\theta} \\ \dot{b} \end{pmatrix}$$

Zeilenweise geschrieben:

$$\begin{aligned} \dot{x} &= \dot{\theta}(-b\sin\theta - a\cos\theta) + \dot{b}\cos\theta \\ \dot{y} &= \dot{\theta}(b\cos\theta - a\sin\theta) + \dot{b}\sin\theta \end{aligned}$$

Im gefragten konkreten Fall ergibt sich:

$$\begin{aligned} \dot{(x)} &= -20\,mm/s \\ \dot{(y)} &= +10\,mm/s \end{aligned}$$

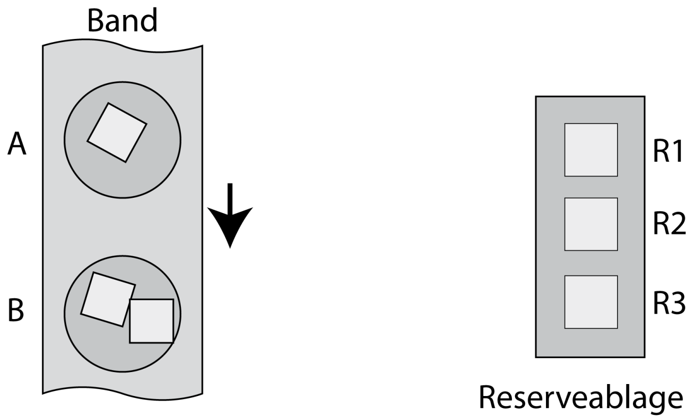
Der Geschwindigkeitsvektor ist also

$$\dot{\vec{v}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -20 \\ 30 \\ 0 \end{pmatrix} mm/s$$

f) Dieser Roboter kann nur dann Singularitäten aufweisen, wenn die mechanische Ausführung der Linearachse es konstruktiv ermöglicht, den TCP so weit nach hinten zu bewegen, dass $b = 0$ ist. Zwei mögliche Begründungen (eine genügt):

1. Wenn $b = 0$ ist, wird für $\theta = 90^o$ oder $\theta = 270^o$ die erste Zeile der Jacobi-Matrix 8siehe oben) komplett gleich Null, für $\theta = 0^o$ oder $\theta = 180^o$ die zweite.
2. Wenn $b = 0$ ist, kann – auch bei Bewegung beider Achsen – der TCP nur noch tangential bewegt werden, nicht mehr radial; hier ist also ein Freiheitsgrad verloren gegangen.

Aufg.4) Auf einem Montageband werden mit jedem Takt zwei Trägerschalen A und B in eine Roboterzelle gefahren. Auf jeder Trägerschale können 0, 1 oder 2 Würfel liegen. Ein Roboter mit Greifer soll nun dafür sorgen, dass beim Ausfahren aus der Zelle auf jeder Schale genau ein Würfel liegt. Zur Verfügung steht noch eine Reserveablage, auf der zeitweilig Würfel gelagert werden können. Der Abstand zur Reserveschale ist deutlich größer als der zwischen den Schalen A und B.



Eine Kamera meldet dem Robotersystem Anzahl und Position der Würfel auf jeder Schale, z.B. für den Zustand in der Abbildung A=1, B=2.

Geben Sie einen zeiteffizienten Algorithmus für die Arbeit des Roboters an. Für den Transport der Würfel verwenden Sie einfach die Befehle `mov(A nach B)`, `mov(B nach A)`, `mov(A nach R)`, `mov(B nach R)`, `mov(R nach A)` und `mov(R nach B)`,.

Formulieren Sie den Algorithmus als Entscheidungstabelle, Entscheidungsbaum, Flussdiagramm, Struktogramm oder Programmcode in einer Roboter- oder Programmiersprache.

Lösung Aufgabe 4 Lösungsidee und grobe Fallunterscheidung:

- Wenn auf einer Schale ein Würfel fehlt, wird er von der Reserveablage geholt
- Liegt auf einer Schale ein Würfel zu viel, wird er auf der Reserveablage abgelegt.
- Wenn auf einer Schale ein Würfel zu viel liegt und auf der anderen ein Würfle fehlt, wird ein Würfel direkt von Schale zu Schale bewegt (effizienter)
- Die Fehlersituationen müssen behandelt werden (Ablegen wenn Reserve voll, holen wenn Reserve leer)

	$n_B = 0$	$n_B = 1$	$n_B = 2$
$n_A = 0$	holen(A) holen(B)	holen(A)	mov(B nach A)
$n_A = 1$	holen(B)	–	ablegen(B)
$n_A = 2$	mov(A nach B)	ablegen(A)	ablegen(A) ablegen(B)

```
// n : Anzahl Würfel in Reserve
// R(1), R(2), R(3) : Array der Reserveplätze

holen(x) {
  wenn (n>0) {
    mov(R(n) nach x);
    n--;
    }
  sonst
    Fehler(1);
}

ablegen(x) {
  wenn (n<3) {
    n++;
    mov(x nach R(n));
    }
  sonst
    Fehler(2);
}
```