Übungsblatt 6

Julius Auer, Thomas Tegethoff

Aufgabe 1 (Inverse Kinematik):

Die Rechnung auf den Folien ist leider nicht korrekt (?) - die Matrizen sind etwas komplexer:

$$\begin{aligned}
T_{2\to 0} &= \begin{pmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_2 & -s_2 & l_1 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{(soweit, so gut. Aber:)} \\
\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \begin{pmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_2 & -s_2 & l_1 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} l_2 \\ 0 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} l_2 \cdot c_2 + l_1 \\ l_2 \cdot s_2 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} l_1 \cdot c_1 + l_2 \cdot (c_1 \cdot c_2 - s_1 \cdot s_2) \\ l_1 \cdot s_1 + l_2 \cdot (s_1 \cdot c_2 + c_1 \cdot s_2) \\ 1 \end{pmatrix} \\
\frac{\delta}{\delta\Theta_1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \begin{pmatrix} -l_1 \cdot s_1 + l_2 \cdot (-s_1 \cdot c_2 - c_1 \cdot s_2) \\ l_1 \cdot c_1 + l_2 \cdot (c_1 \cdot c_2 - s_1 \cdot s_2) \\ 0 \end{pmatrix} \\
\frac{\delta}{\delta\Theta_2} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \begin{pmatrix} l_2 \cdot (-c_1 \cdot s_2 - s_1 \cdot c_2) \\ l_2 \cdot (-s_1 \cdot s_2 + c_1 \cdot c_2) \\ 0 \end{pmatrix} \\
J &= \begin{pmatrix} -l_1 \cdot s_1 + l_2 \cdot (-s_1 \cdot c_2 - c_1 \cdot s_2) & l_2 \cdot (-c_1 \cdot s_2 - s_1 \cdot c_2) \\ l_1 \cdot c_1 + l_2 \cdot (c_1 \cdot c_2 - s_1 \cdot s_2) & l_2 \cdot (-s_1 \cdot s_2 + c_1 \cdot c_2) \end{pmatrix} \\
&= \begin{pmatrix} -y & l_2 \cdot (-c_1 \cdot s_2 - s_1 \cdot c_2) \\ x & l_2 \cdot (-s_1 \cdot s_2 + c_1 \cdot c_2) \end{pmatrix}
\end{aligned}$$

Da die Jakobi-Matrix hier immer hübsch quadratisch ist, kann die Inverse direkt über die geschlossene Form berechnet werden:

$$J^{-1} = \frac{1}{l_1 \cdot l_2 \cdot (s_1^2 \cdot s_2 + c_1^2 \cdot s_2)} \cdot \begin{pmatrix} l_2 \cdot (-s_1 \cdot s_2 + c_1 \cdot c_2) & -l_2 \cdot (-c_1 \cdot s_2 - s_1 \cdot c_2) \\ -l_1 \cdot c_1 - l_2 \cdot (c_1 \cdot c_2 - s_1 \cdot s_2) & -l_1 \cdot s_1 + l_2 \cdot (-s_1 \cdot c_2 - c_1 \cdot s_2) \end{pmatrix}$$

Es seien nun: l_1, l_2 fest, $tf(\Theta_1, \Theta_2) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^2$ eine Funktion um wie gezeigt $T_{2\to 0}$ zu erzeugen, $j(\Theta_1, \Theta_2) : \mathbb{R}^{\nvDash} \to \mathbb{R}^{2\times 2}$ eine Funktion um wie gezeigt J^{-1} zu erzeugen, $f(\overrightarrow{x}) : \mathbb{R}^2 \to \mathbb{B}$ eine Funktion die bei Eingabe der aktuellen Position $\overrightarrow{x}$ entscheidet ob der Algorithmus terminiert und $d \in [0, 1]$ ein Dämpfungsfaktor. Dann lässt sich die inverse Kinematik folgenderweise berechnen:

(1) Initiale Winkel: Θ_1, Θ_2

(2) End-Effektor-Position: $\overrightarrow{x} = tf(\Theta_1, \Theta_2)$

(3) Ziel-Position: $\overrightarrow{x_t}$

(4) solange $f(\overrightarrow{x})$:

  (5) Prüfe und behandle ggf. Singularität (tritt hier nicht auf, falls Anfangsposition günstig gewählt)

  (6) $\begin{pmatrix} \Theta_1 \\ \Theta_2 \end{pmatrix} += d \cdot j(\Theta_1, \Theta_2) \cdot (\overrightarrow{x_t} - \overrightarrow{x})$

  (7) $\overrightarrow{x} = tf(\Theta_1, \Theta_2)$

Genauso implementiert liefert das die folgenden Ergebnisse:

(a) Plot (b) Trajektorie

Abbildung 1: 1 Iteration

b)

(a) Plot (b) Trajektorie

Abbildung 2: 10 Iteration

c)

(a) Plot (b) Trajektorie

Abbildung 3: 100 Iteration

d)

(a) Plot (b) Trajektorie

Abbildung 4: 1000 Iteration

Aufgabe 2 (Null-Space):