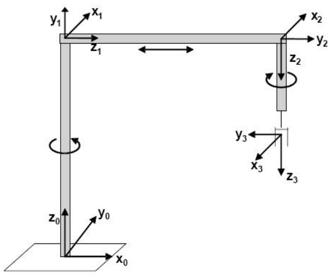
1 Szenario

Es soll eine Anwendung für die Klasse Matrix4D gezeigt werden. Man stelle sich vor man hat einen Roboterarm, ähnlich wie in Figur 1. Die Gelenke des Roboters sind wie folgt:

1. rotatorisch, Rotation mit θ_1
2. translatorisch, Translation mit d_2
3. rotatorisch, Rotation mit θ_3

Die Parameter θ_1, d_2 und θ_1 werden als Achsparameter bezeichnet. Ziel der sogenannten direkten Kinematik ist es nun aus dem gegebenen Vektor der Achsparameter des Roboters die Position und Orientierung des Werkzeugs im Bezug zum Koordinatensystem am Fußboden des Roboters anzugeben. Das Koordinatensysteme am Boden des Roboters wird mit $Base$, das am Werkzeug mit $Tool$ bezeichnet.



Joint i	θ_i	d_i	a_i	α_i
1	q_1	500	0	$\pi/2$
2	0	q_2	0	$\pi/2$
3	q_3	300	0	0

Abbildung 1: Ein Roboterarm

Um nun Koordinaten vom Werkzeugsystem *Tool* $[p]^{Tool}$ in das *Base* $[p]_{Base}$ System zu übertragen gilt:

$$[p]_{Base} = T_{Base}^{Tool} \cdot [p]^{Tool} \quad \text{dabei gilt}$$

$$T_{Base}^{Tool} = T_0^1 \cdot T_1^2 \cdot T_2^3 \quad \text{mit}$$

$$T_{k-1}^k = \begin{pmatrix} \cos\theta_k & -\cos\alpha_k \sin\theta_k & \sin\alpha_k \sin\theta_k & a_k \cos\theta_k \\ \sin\theta_k & \cos\alpha_k \cos\theta_k & -\sin\alpha_k \cos\theta_k & a_k \sin\theta_k \\ 0 & \sin\alpha_k & \cos\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Die Matrix T_{k-1}^k ist genau die Matrix, die Koordinaten vom System $k-1$ in das System k transferiert. Einsetzen in die Matrix kann man mit Hilfe der Tabelle der kinematischen Parameter, welche in obiger Zeichnung angegeben ist. Wie man diese bestimmt soll nicht Thema diese Tutorials sein.

2 Implementierung

Uns Interessieren jetzt die Koordinaten $[p]^{Tool} = [1, 1, 1]$ im Basiskoordinatenframe, wenn

$$\begin{aligned} \theta_1 &= \frac{\pi}{2} \\ d_2 &= 10cm; \\ \theta_3 &= \frac{\pi}{2} \end{aligned}$$

Dazu müssen wir folgende Schritte ausführen:

1. Die Matrizen $T_0^1 \quad , T_1^2 \quad T_2^3$ bestimmen
2. Die Matrix T_{Base}^{Tool} bestimmen
3. Den Punkt p mit T_{Base}^{Tool} multiplizieren

Anhand obigem Beispiel können wir mit Hilfe der Bibliothek sehr schön die Matrix T_{Base}^{Tool} bestimmen:

```
// Schritt 1
double t1[16]={0,0,1,0,1,0,0,0,0,1,0,500,0,0,0,1};
double t2[16]={1,0,0,0,0,0,-1,0,0,1,0,10,0,0,0,1};
double t3[16]={0,-1,0,0,1,0,0,0,0,0,1,300,0,0,0,1};
Matrix4D T01(t1);
Matrix4D T12(t2);
Matrix4D T23(t3);
//Schritt 2
Matrix4D T_toolBase = T01.mMult(T12.mMult(T23));
//Schritt 3
Vector3D v(1,1,1);
Vector4D p = v.to4D();
p_Base = p.matVecMult(T_toolBase); //[11,-1,199,1]
```

Als Hinweis, auch wenn es hier sehr einfach aussieht die Koordinaten zu berechnen, so ist es doch besser die Matrizenmultiplikation vorher im allgemeinen Fall zu lösen, denn damit spart man sich noch einmal erheblich viel Rechenaufwand