

# Octave / Matlab Projekt: Denavit & Hartenberg

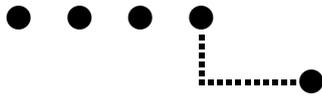
Wiedmer Simona  
Ducommun Thierry

Dozent: Andreas Stahel

16. Januar 2008

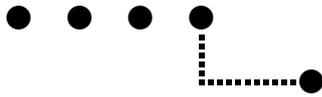
## **Zusammenfassung**

Das Matlab-Programm „DenavitHartenberg Berechnung.m“ dient zur raschen und bequemen Berechnung der Transformationsmatrix ab der Denavit & Hartenberg Tabelle.



## **Inhaltsverzeichnis**

<b>1 Einführung in die Robotik</b>	<b>3</b>
<b>2 Denavit &amp; Hartenberg</b>	<b>3</b>
2.1 Vorgehen zum bestimmen der D&H Parameter . . . . .	4
<b>3 Das Programm</b>	<b>5</b>
<b>4 Schlusswort</b>	<b>8</b>
<b>5 Epilog</b>	<b>8</b>



## 1 Einführung in die Robotik

Die Berechnung der Koordinatentransformation eines Roboters ist ein wesentlicher Schritt zur Realisierung von Trajektorien im Raum, um Objekte, von denen man die genaue Position im Raum kennt, zu ergreifen. Die Koordinatentransformation ergibt die Beziehung zwischen der Mathematik, die durch die Koordinaten der einzelnen Gelenke des Roboters und die Position jedes Organs entsteht, und der Endposition des Roboters (TCP<sup>1</sup>) im Raum.

Dabei gibt es mehrere Methoden die Koordinatentransformation zu machen. Zum Beispiel:

- Die direkte Koordinatentransformation
- Die Denavit & Hartenberg Methode
- Die alternative Methode

## 2 Denavit & Hartenberg

Die Denavit & Hartenberg Methode ermöglicht die systematische Berechnung der Koordinatentransformation jedes Roboters in „Serie“ indem bei jedem Gelenk einen intermediären Ursprung auf besondere Weise gesetzt wird.

Diese Methode besitzt die Annahme, dass jedes Gelenk nur einen Freiheitsgrad (DoF<sup>2</sup>) hat. Entweder ein Dreh- oder ein Schubgelenk. Diese Annahme ist überhaupt nicht restriktiv, da jedes Gelenk mit mehreren DoFs zu mehreren Gelenken mit je einem DoF zerlegt werden kann.

Hier ein Beispiel für einen Roboter mit 3 Gelenken und die bereits eingetragenen besonderen Ursprünge.

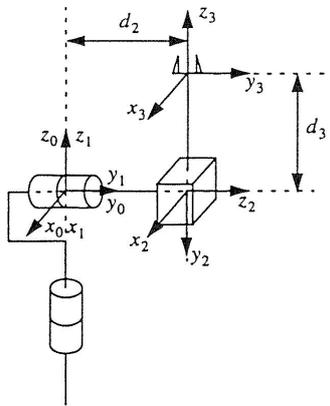
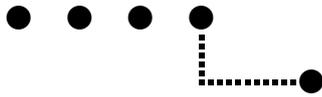


Abbildung 1: Beispiel eines Roboters mit 3 DoFs

Anschliessend gilt es mit Hilfe von vier Parametern den Ursprung  $\{ i \}$  im Ursprung  $\{ i-1 \}$  darzustellen.

<sup>1</sup>Tool Center Point

<sup>2</sup>Degree of Freedom



## 2.1 Vorgehen zum bestimmen der D&H Parameter

Wahl der Koordinatensysteme:

1. Koordinatenachsen  $z_i$  parallel zu den Gelankachsen wählen
2. Koordinatenachsen  $x_i$  parallel zum Lot von  $z_{i+1}$  auf  $z_i$  (d.h. senkrecht zu  $z_i$  und  $z_{i+1}$ )
3. Koordinatenachsen  $y_i$  senkrecht zu  $x_i, z_i$  wählen
4. Alle Koordinatenursprünge  $O_i$  kennzeichnen ( $O_i$ : Schnittpunkt des Lotes von  $z_{i+1}$  auf  $z_i$  mit  $z_i$ )
5. Spez: Koordinatensystem „O“ ist feststehend.  $z_0$  parallel  $z_1$  wählen
6. Spez: Koordinatensystem „n“ zur letzten Gelenkachse.  
falls Drehgelenk:  $x_n$  parallel  $x_{n-1}$  wählen  
falls Schubgelenk:  $x_n$  so wählen, dass  $\Theta_n = 0^\circ$

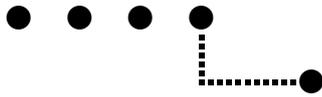
Bestimmung der Gelenkparameter  $i = 1$  bis  $n$ :

- $\alpha_{i-1}$ : Winkel zwischen  $z_{i-1}$  und  $z_i$  bei Drehung um  $x_{i-1}$
- $a_{i-1}$ : Abstand von  $z_{i-1}$  und  $z_i$  entlang  $x_{i-1}$
- $d_i$ : Abstand von  $O_{i-1}$  und  $O_i$  entlang  $z_i$
- $\theta_i$ : Winkel zwischen  $x_{i-1}$  und  $x_i$  bei Drehung um  $z_i$

Daraus folgend ergibt sich für den oben genannten Roboter folgende D&H Tabelle.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	$-90^\circ$	0	$d_2$	$\theta_2$
3	$90^\circ$	0	$d_3$	0

Abbildung 2: Beispiel einer D&H Tabelle



### 3 Das Programm

Im Programm geht es darum, aus dieser D&H Tabelle eine Transformationsmatrix zu erstellen, die die Endposition des Roboters TCP berechnet.  
Dafür muss die D&H Tabelle folgendermassen ins „DenavitHartenberg Berechnung.m“ File geschrieben werden. Dabei wird angenommen, dass  $\Theta_1 = 0$ ,  $\Theta_2 = 0$ ,  $d_2 = 2$  und  $d_3 = 3$ . Natürlich können diese Parameter auch irgend welche andere Werte haben.

```
***** D&H-Tabelle definieren *****%
%   alpha(i-1)  a(i-1)  d(i)  theta(i)      %
% 1              %
% 2              %
% 3              %
% :              %
% i              %
% :              %
% n              %
*****%
Tabelle=[0,0,0,0;-pi()/2,0,2,0;pi()/2,0,3,0];
```

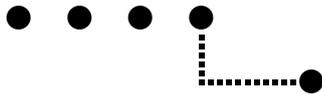
In einem weiteren Schritt wird ermittelt, wieviele Gelenke (Anzahl Parameter Zeilen) dieser Roboter besitzt und alle Parameter werden in eine einzige Matrix eingelesen, damit sie später weiterverarbeitet werden können.

```
***** Tabelle auswerten zu T verarbeiten *****%
G=size(Tabelle);
g=G(1,1);          % Anzahl Zeilen der D&H-Tabelle
T=zeros(1:g,1:4);
T(1:g,1:4)=Tabelle(1:g,1:4);
***** (end) Tabelle auswerten zu T verarbeiten *****%
```

Die Funktion „DoMatrix(var)“ wird im unteren Teil des Programms verwendet. Diese Funktion liest eine Zeile der Denavit & Hartenberg Tabelle nach der andern ein und ergibt die entsprechende Transformationsmatrix jedes Gelenks in folgender Form zurück.

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Abbildung 3: Transformationsmatrix für jedes einzelne Gelenk



```
function M = DoMatrix(T,i)
% Funktionsbeschreibung: Es wird eine Zeile von der D&H Tabelle "ubergeben
% und eine Transformationsmatrix zur"uckgegeben

% T: Ist die D&H Tabelle mit den Parametern alpha, a, d, theta
% i: Die zuberechnende Zeile der D&H Tabelle
% M: Transformationsmatrix der jeweiligen Zeile

M=zeros(4,4);

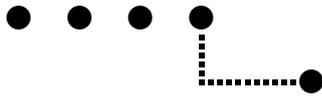
M=[cos(T(i,4)), (-sin(T(i,4))), 0, T(i,2);
(sin(T(i,4))*cos(T(i,1))), (cos(T(i,4))*cos(T(i,1))), (-sin(T(i,1))),((-sin(T(i,1)))*T(i,3));
(sin(T(i,4))*sin(T(i,1))), (cos(T(i,4))*sin(T(i,1))), cos(T(i,1)), (cos(T(i,1))*T(i,3));
0, 0, 0 ,1];
```

Danach beginnt die Initialisierung und die eigentliche Berechnung der Transformationsmatrix.

```
%***** Initialisieren von TFM, TF_TOT, 0 *****%
TFM=zeros(4,4*g);      % Transformationsmatrix pro Zeile
TF_TOT=zeros(4,4*g);  % Gesamte Transformationsmatrix
O=zeros(4,g+1);
%***** (end) Initialisieren von TFM, TF_TOT, 0 *****%

%***** Transformationsmatrix generieren *****%
i=1;                    %Integrator
while(i<=g)
    % Transformationsmatrix von i-1 bis i
    TFM(1:4,(4*i-3):4*i)=DoMatrix(T,i);
    % Transformationsmatrizen von 0 bis i
    if(i==1)
        TF_TOT(1:4,(4*i-3):4*i)=TFM(1:4,(4*i-3):4*i);
    else
        TF_TOT(1:4,(4*i-3):4*i)=TF_TOT(1:4,(4*(i-1)-3):4*(i-1))*TFM(1:4,(4*i-3):4*i);
    end
    i=i+1;
end
Transformationsmatrix=TF_TOT(1:4,(4*g)-3:4*g)
%***** (end) Transformationsmatrix generieren *****%
disp('done!')
```

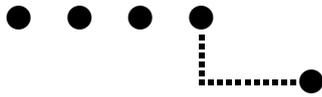
Schlussendlich wird die gesamte Transformationsmatrix ausgegeben  ${}^0_nT$ .



Um weiter Matrizen auszugeben, können in Matlab folgende Befehle ausgeführt werden, die als Variable des Programms gespeichert sind:

- TFM: Ergibt alle Transformationsmatrizen der einzelnen Gelenke.  
Dabei ergeben die ersten vier Spalten von TFM die erste Transformationsmatrix  ${}^0_1T$ , die nächsten vier Spalten ergeben  ${}^1_2T$  usw. bis  ${}^{n-1}_nT$ .
- TF\_TOT: Ergibt die Multiplikation aller bisherigen Transformationsmatrizen.  
Dabei ergeben die ersten vier Spalten von TF\_TOT die erste Transformationsmatrix  ${}^0_1T$ , die nächsten vier Spalten ergeben die Multiplikation der ersten beiden Transformationsmatrizen  ${}^0_1T \cdot {}^1_2T$ , weitere vier Spalten ergeben die Multiplikation der ersten drei Transformationsmatrizen  ${}^0_1T \cdot {}^1_2T \cdot {}^2_3T$  und ganz am Schluss  ${}^0_1T \cdot {}^1_2T \cdot {}^2_3T \cdot \dots \cdot {}^{n-1}_nT$

Das definitive Programm „Denavit & Hartenberg berechnung.m“ befindet sich auf dem Server unter:  
P:/fbf/Dozenten/Stahel/Octave/Projects/DenavitHartenberg



## 4 Schlusswort

Das Programm funktioniert soweit einwandfrei. Diese Version beinhaltet einzig die mathematische Berechnung der Transformationsmatrix aus der Denavit & Hartenberg Tabelle.

Vorgesehen war noch eine Version, die anhand der D&H Tabelle den Roboter mit Hilfe von Strichen im Raum darstellt und diese auch gemäss Rotationen aus der D&H Tabelle bewegt. Leider ist es uns nicht ganz gelungen, diese Animation des Roboters fehlerfrei zum laufen zu bringen.

Im selben Ordner ist unter P:/fbf/Dozenten/Stahel/Octave/Projects/DenavitHartenberg auch ein File mit der Bezeichnung „Incomplete DH Berechnung mit Plot“ abgelegt. Wie der Filename bereits hindeutet funktioniert dieses File noch nicht wie gewünscht. Falls es jemand unbedingt brauchen sollte, kann er dies übernehmen und es versuchen zum laufen zu bringen.

## 5 Epilog

Die Arbeit wurde mit bestem Wissen und Gewissen entwickelt. Sollten trotzdem noch Fragen offen bleiben, stehen wir gerne zur Verfügung. Wir sind unter den folgenden Adressen erreichbar:

Wiedmer Simona: [wieds5@bfh.ch](mailto:wieds5@bfh.ch)

Ducommun Thierry: [ducot1@bfh.ch](mailto:ducot1@bfh.ch)

Biel, den 16. Januar 2008

Wiedmer Simona

Ducommun Thierry