Denavit-Hartenberg

$$\boldsymbol{T}_{(i-1),i} = \left[H_{(i-1),i}\right] = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
H_joint = @(theta, theta_0, alpha, a, d) [cos(theta+theta_0) -cos(alpha)*sin(theta+theta_0)  s
                                 sin(theta+theta_0)  cos(alpha)*cos(theta+theta_0) -sin(alpha)
                                 0            sin(alpha)             cos(alpha)             c
                                 0               0                         0
```

```
H_joint =
    @(theta,theta_0,alpha,a,d)[cos(theta+theta_0),-cos(alpha)*sin(theta+theta_0),sin(alpha)*sin(theta+th
```

```
syms theta theta_0 alpha a d real
H_joint (theta, theta_0, alpha, a, d)        %% this is for visual verification
```

ans =

$$\begin{pmatrix} \cos(\theta_0+\theta) & -\sin(\theta_0+\theta)\cos(\alpha) & \sin(\theta_0+\theta)\sin(\alpha) & a\cos(\theta_0+\theta) \\ \sin(\theta_0+\theta) & \cos(\theta_0+\theta)\cos(\alpha) & -\cos(\theta_0+\theta)\sin(\alpha) & a\sin(\theta_0+\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
mm = 1e-3;
deg = pi/180;

%%               Base
delta_base = [-27.17; -155.5; 28]*mm;
H_base = [eye(3) delta_base
          0 0 0  1]
```

```
H_base = 4x4 double

    1.0000         0         0   -0.0272
         0    1.0000         0   -0.1555
         0         0    1.0000    0.0280
         0         0         0    1.0000
```

```
%%               Manipulator
syms theta_1 theta_2 theta_3 real
i_link = 1;
DH_par{i_link}.theta     = theta_1;
DH_par{i_link}.theta_0   = -90*deg;
DH_par{i_link}.d         =  41.68*mm;
DH_par{i_link}.a         =  1.55*mm;
DH_par{i_link}.alpha     =  90*deg;

i_link = 2;
```

```
DH_par{i_link}.theta     =  0;
DH_par{i_link}.theta_0   =  0*deg;
DH_par{i_link}.d         =  20.8*mm;
DH_par{i_link}.a         =  0*mm;
DH_par{i_link}.alpha     =  0*deg;

i_link = 3;
DH_par{i_link}.theta     = theta_2;
DH_par{i_link}.theta_0   =  180*deg;
DH_par{i_link}.d         =  -18.1*mm;
DH_par{i_link}.a         =  11.7*mm;
DH_par{i_link}.alpha     =  -90*deg;

i_link = 4;
DH_par{i_link}.theta     =  0;
DH_par{i_link}.theta_0   =  0*deg;
DH_par{i_link}.d         =  -18.4*mm;
DH_par{i_link}.a         =  0*mm;
DH_par{i_link}.alpha     =  0*deg;

i_link = 5;
DH_par{i_link}.theta     = -theta_3;
DH_par{i_link}.theta_0   = -180*deg;
DH_par{i_link}.d         = -47.5*mm;
DH_par{i_link}.a         =  17.5*mm;
DH_par{i_link}.alpha     =  90*deg;

i_link = 6;
DH_par{i_link}.theta     = 0;
DH_par{i_link}.theta_0   =  0*deg;
DH_par{i_link}.d         =  5.67*mm;
DH_par{i_link}.a         =  0*mm;
DH_par{i_link}.alpha     =  0*deg;

H_sym = H_base;
for i=1:length(DH_par)
    H_sym = H_sym * ...
        H_joint(DH_par{i}.theta, DH_par{i}.theta_0, DH_par{i}.alpha, DH_par{i}.a, DH_par{i}.d)
end
% J_sym = jacobian(H_sym, [theta_1; theta_2; theta_3])

H = matlabFunction(H_sym)
```

```
H =
    @(theta_1,theta_2,theta_3)reshape([cos(theta_3).*(cos(theta_2).*cos(theta_1-pi.*(1.0./2.0))-sin(thet
```

```
Rot = matlabFunction([eye(3) zeros(3,1)] * H_sym * [eye(3); 0 0 0])
```

```
Rot =
    @(theta_1,theta_2,theta_3)reshape([cos(theta_3).*(cos(theta_2).*cos(theta_1-pi.*(1.0./2.0))-sin(thet
```

```
Pos = matlabFunction( [eye(3) zeros(3,1)]* H_sym  * [0; 0 ; 0; 1])
```

```
Pos =
    @(theta_1,theta_2,theta_3)[cos(theta_1-pi.*(1.0./2.0)).*1.55e-3+sin(theta_1-pi.*(1.0./2.0)).*2.69999
```

Tracking on SO(3)

Koditschek (1988). Application of a new Lyapunov function to global adaptive attitude tracking. In *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on* (pp. 63-68). IEEE.