King's College London

TOMSY Project Report on

The Kinematics of KCL Five Fingered Metamorphic Hand

Evangelos Emmanouil
Dr. Guowo Wei
Dr. Ketao Zhang
Prof. Jian S. Dai

Centre for Robotics Research
School of Natural & Mathematical Sciences
King's College London
Strand
WC2R 2LS

March 12, 2012

Contents

1 Description

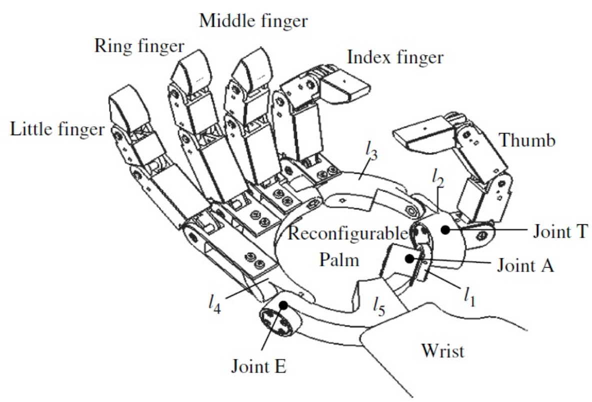

Figure 1: Five Fingered Metamorphic Hand.

The KCL five fingered metamorphic hand, shown in Fig. 1, comprises of a metamorphic palm and five fingers. Each finger is very simple, just some links connected by revolute joints whose axes are parallel. The interesting part is the palm. It is a spherical five-bar linkage. It is made out of five links in a circular configuration with every joint axis z_i passing through the centre of the sphere, as shown in Fig. 2.

Of the five joints θ_i on the palm, only θ_1 and θ_5 are actuated. The remaining joints, θ_2, θ_3 and θ_4 are rotating freely, based on the constraints imposed by the geometry of the spherical linkage. Each finger is actuated by only one tendon and move on it's own finger operation plane, as shown in Fig. 3.

2 Palm Kinematics

The coordinates of the points A, B, C, D and E as well as joint angles θ_2, θ_3 and θ_4 should be computed first. We start by assuming that point E is $p_E = [0, 0, 1]$. Then the coordinates of points A, B and D are computed from the known and actuated joint angles θ_5 and θ_1. Then, angle θ_3 can be computed by applying the cosine law for spherical triangles on the triangle $\triangle BCD$. Computing angle θ_4 can be done by adding together angles $\angle EDB$,

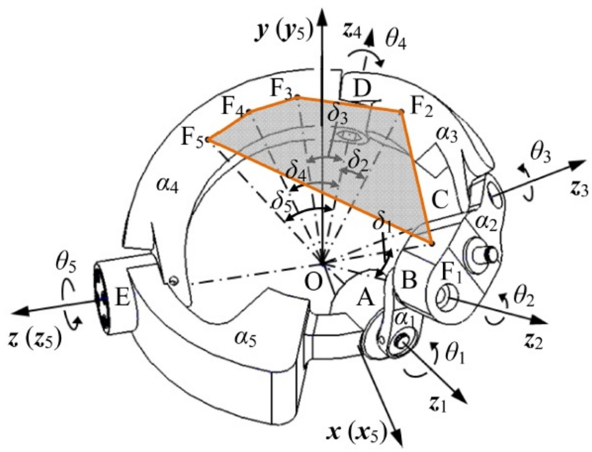

Figure 2: Metamorphic Palm and Finger Attachment Points.

$\angle BDC$ and subtracting π. Angle θ_3 can be computed in a similar way, by adding $\angle ABD$, $\angle DBC$ and subtracting π. This indicates that the distance $\|BD\|$ has to be computed.

2.1 Angles α_1 to α_5

The values for the angles α_1 to α_5 are as follows:

$$\alpha_1 = 25° \quad (1)$$
$$\alpha_2 = 40° \quad (2)$$
$$\alpha_3 = 70° \quad (3)$$
$$\alpha_4 = 112° \quad (4)$$
$$\alpha_5 = 113° \quad (5)$$

2.2 Points A, B, D, E

The coordinates for points A, B, D and E can be computed by performing the rotations described in equations 6, 7, 8 and 9.

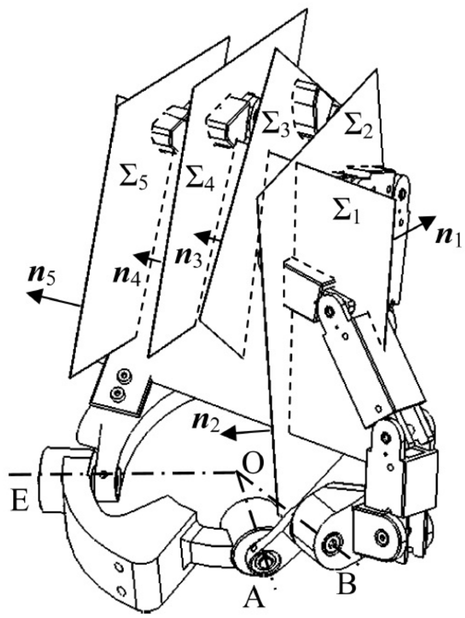

Figure 3: Finger Operational Planes.

$$\mathbf{p_A} = R(y_5, \alpha_5)\,\mathbf{k} \tag{6}$$
$$\mathbf{p_B} = R(y_5, \alpha_5)\; R(z_1, \theta_1)\,\mathbf{k} \tag{7}$$
$$\mathbf{p_D} = R(z_5, -\theta_5)\; R(y_4, -\alpha_4)\,\mathbf{k} \tag{8}$$
$$\mathbf{p_E} = \begin{bmatrix} 0 \\ 0 \\ R \end{bmatrix} \tag{9}$$

2.3 Joint Angles θ_2, θ_3, θ_4

First, the distance $\|BD\|$ and the angle α_{bd} are computed.

$$\mathbf{bd} = \mathbf{p_b} - \mathbf{p_d} \tag{10}$$
$$\|BD\| = \sqrt{\mathbf{bd}'\ \mathbf{bd}} \tag{11}$$
$$\alpha_{BD} = \arccos\left(1 - \frac{\|BD\|^2}{2}\right) \tag{12}$$

Then, θ_3 is computed by the spherical law of cosines.

$$\angle BCD = \arccos\left(\frac{\cos\alpha_{BD} - \cos\alpha_2\ \cos\alpha_3}{\sin\alpha_2\ \sin\alpha_3}\right) \tag{13}$$
$$\theta_3 = \angle BCD - \pi \tag{14}$$

Next, α_{BE} is computed and then used to compute θ_4.

$$\mathbf{be} = \mathbf{p_b} - \mathbf{p_e} \tag{15}$$
$$\|BE\| = \sqrt{\mathbf{be}'\ \mathbf{be}} \tag{16}$$
$$\alpha_{BE} = \arccos\left(1 - \frac{\|BE\|^2}{2}\right) \tag{17}$$

By using α_{BE}, θ_4 is computed.

$$\angle EDB = \arccos\left(\frac{\cos\alpha_{BE} - \cos\alpha_4\ \cos\alpha_{BD}}{\sin\alpha_4\ \sin\alpha_{BD}}\right) \tag{18}$$
$$\angle BDC = \arccos\left(\frac{\cos\alpha_2 - \cos\alpha_3\ \cos\alpha_{BD}}{\sin\alpha_3\ \sin\alpha_{BD}}\right) \tag{19}$$
$$\theta_4 = (\angle EDB + \angle BDC) - \pi \tag{20}$$

Finally, α_{AD} is computed and then used to compute θ_3. It must be noted that the angles of a spherical triangle don't add up to π. Because of this fact, angle θ_3 has to be computed in the same fashion as angles θ_2 and θ_4.

$$\mathbf{ad} = \mathbf{p_a} - \mathbf{p_d} \tag{21}$$
$$\|AD\| = \sqrt{\mathbf{ad}'\ \mathbf{ad}} \tag{22}$$
$$\alpha_{AD} = \arccos\left(1 - \frac{\|AD\|^2}{2}\right) \tag{23}$$

By using α_{AD}, θ_4 is computed.

$$\angle ABD = \arccos\left(\frac{\cos\alpha_{AD} - \cos\alpha_1\ \cos\alpha_{BD}}{\sin\alpha_1\ \sin\alpha_{BD}}\right) \tag{24}$$

$$\angle DBC = \arccos\left(\frac{\cos\alpha_3 - \cos\alpha_2\ \cos\alpha_{BD}}{\sin\alpha_2\ \sin\alpha_{BD}}\right) \tag{25}$$

$$\theta_2 = \pi - (\angle ABD + \angle DBC) \tag{26}$$

2.4 Point C

Point C can now be located in two different ways. One way is to move in a clockwise direction (E to D to C) and the other way is to move in a counter clockwise direction (E to A to B to C). The first way is chosen since it is the least computationally intensive. One can use both ways and see if the equations produce the same coordinates in order to verify their correctness.

$$\mathbf{p_C} = R(z_5, -\theta_5)\ R(y_4, -\alpha_4)\ R(z_4, -\theta_4)\ R(y_3, -\alpha_3)\ \mathbf{k} \tag{27}$$

3 MCP Joints

After determining angles θ_2, θ_3 and θ_4 from equations 26, 14 and 20, the points the fingers attach to the palm, as depicted in Fig. 4 have to be determined.

Angles δ_1 to δ_5 determine points F_1 to F_5 where each MCP joint attaches on the palm and angles γ_2 to γ_5 are determined and fixed so that each finger, except for the thumb, is parallel and co-linear with the arm. γ_1 is 0.

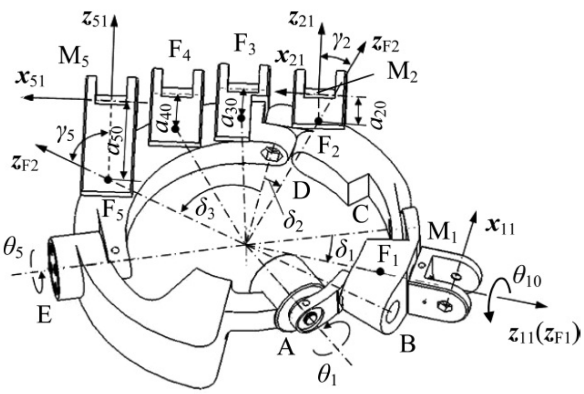

Figure 4: MetaCarpoPhalangeal Joints.

3.1 Angles δ_1 to δ_5

The values of the angles δ_1 to δ_5 are given by equations 28, 29, 30, 31 and 32.

$$\delta_1 = \alpha_2 \frac{1}{2} \tag{28}$$

$$\delta_2 = \alpha_3 \frac{1}{4} \tag{29}$$

$$\delta_3 = \alpha_4 \frac{1}{7} \tag{30}$$

$$\delta_4 = \alpha_4 \frac{3}{7} \tag{31}$$

$$\delta_5 = \alpha_4 \frac{5}{7} \tag{32}$$

3.2 Angles γ_1 to γ_5

The values of the angles γ_1 to γ_5 are given by equations 34, 34, 35, 36 and 37.

$$\begin{aligned}
\gamma_1 &= 0 && (33)\\
\gamma_2 &= -\delta_2 - (\alpha_4 - \frac{\pi}{2}) && (34)\\
\gamma_3 &= \delta_3 - (\alpha_4 - \frac{\pi}{2}) && (35)\\
\gamma_4 &= \delta_4 - (\alpha_4 - \frac{\pi}{2}) && (36)\\
\gamma_5 &= \delta_5 - (\alpha_4 - \frac{\pi}{2}) && (37)
\end{aligned}$$

3.3 Points F_i and M_i

In order to obtain the coordinates that describe the MCP joints, the homogeneous transform matrices to points F_i need to be determined.

$$\begin{aligned}
R_{F_1} &= R(y_5, \alpha_5)\ R(z_1, \theta_1)\ R(y_1, \alpha_1)\ R(z_2, \theta_2)\ R(y_2, \delta_1) && (38)\\
R_{F_2} &= R(z_5, -\theta_5)\ R(y_4, -\alpha_4)\ R(z_4, -\theta_4)\ R(y_3, \delta_2) && (39)\\
R_{F_3} &= R(z_5, -\theta_5)\ R(y_4, -(\alpha_4 - \delta_3)) && (40)\\
R_{F_4} &= R(z_5, -\theta_5)\ R(y_4, -(\alpha_4 - \delta_4)) && (41)\\
R_{F_5} &= R(z_5, -\theta_5)\ R(y_4, -(\alpha_4 - \delta_5)) && (42)
\end{aligned}$$

After obtaining the rotation matrices R_{F_i}, the coordinates of points F_i are easily obtained by eq. 43.

$$\mathbf{f_i} = R_{F_i}\ \mathbf{k} \tag{43}$$

Then, the homogeneous transform matrices for points F_i are given by eq. 44.

$$D_{F_i} = \begin{bmatrix} R_{F_i} & R_{F_i}\ \mathbf{k} \\ 0 & 1 \end{bmatrix} \tag{44}$$

Next, matrices D_{F_i} are multiplied by the homogeneous transform matrix D_{FM_i} given by eq. 45.

$$D_{FM_i} = \begin{bmatrix} \cos\gamma_i & 0 & -\sin\gamma_i & -a_{i0}\sin\gamma_i \\ 0 & 1 & 0 & 0 \\ \sin\gamma_i & 0 & \cos\gamma_i & a_{i0}\cos\gamma_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{45}$$

The homogeneous transform matrices to points M_1 to M_5 are given by eq. 46 and eq. 47.

$$D_{M_1} = D_{F_1}\, D_{FM_1}\, D_{10} \tag{46}$$

$$D_{M_i} = D_{F_i}\, D_{FM_i} \tag{47}$$

where

$$D_{10} = \begin{bmatrix} \cos\theta_{10} & -\sin\theta_{10} & 0 & 0 \\ \sin\theta_{10} & \cos\theta_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{48}$$

Finally, the coordinates for the points M_1 to M_5 can be computed from eq. 49.

$$\begin{bmatrix} \mathbf{r_{i0}} \\ 0 \end{bmatrix} = D_{M_i} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{49}$$

References

[1] Guowu Wei, Jian S. Dai, Shuxin Wang, and Haifeng Luo (2011) Kinematic analysis and prototype of a metamorphic anthropomorphic hand with a reconfigurable palm. *International Journal Of Humanoid Robotics*, 8 (3), 459–479.

[2] Lei Cui and Jian S. Dai (2011) Posture, workspace and manipulability of the metamorphic multifingered hand with an articulated palm. *ASME Journal of Mechanisms and Robotics*, 3(2), 021001_1–7.

[3] Lei Cui and Jian S. Dai (2010) Singular-value decomposition based kinematic analysis of the metamorphic multifingered hand. pp. DETC2010–28653–1–6.

[4] H. Laimon and Jian S. Dai (2010) Wireless bluetooth remote control of a mlultifingered metamorphic hand. pp. DETC2010–28688–1–5.

[5] Lei Cui, Jian S. Dai, and De Lun Wang (2009) Workspace analysis of a multifingered metamorphic hand. *International Conference on Reconfigurable Mechanisms and Robots, London, ENGLAND 22-JUN-2009 - 24-JUN-2009 GENOVA: KC EDIZIONI*, pp. 619–625.

[6] Jian S. Dai, Delun Wang, and Lei Cui (2009) Orientation and workspace analysis of the multifingered metamorphic hand-metahand. *IEEE TRANSACTIONS ON ROBOTICS*, 25 (4), 942–947.

[7] Lei Cui, D. Wang, and Jian S. Dai (2009) Dimensional synthesis of palm of multifingered metamorphic dexterous hand. *Journal of Dalian University of Technology*, 49 (3), 380–386.

[8] Jian S. Dai and Delun Wang (2007) Geometric analysis and synthesis of the metamorphic robotic hand. *Journal of Mechanical Design*, 129 (11), 1191–1197.

Appendices

Appendix A MATLAB® Code

```
hold off
clear all
clc
syms R real
syms alpha1 alpha2 alpha3 alpha4 alpha5 real
syms theta1 theta2 theta3 theta4 theta5 real
syms a11 a12 a13 a21 a22 a23 a31 a32 a33 real
syms theta11 theta12 theta13 theta21 theta22 theta23 ...
    theta31 theta32 theta33 real
syms g1 g2 g3 g4 g5 real
syms a10 a20 a30 a40 a50 real
syms d1 d2 d3 d4 d5 real
%% Palm Geometry
q = pi/180;
% Palm radius
R = 1;

% Palm input angles
theta1 = 10 * q;
theta5 = 10 * q;

% Palm link angles
alpha1 = 25 * q;
alpha2 = 40 * q;
alpha3 = 70 * q;
alpha4 = 112 * q;
alpha5 = 113 * q;

%% Finger Geometry
% MCP joint length
a10 = 0.1;
a20 = 0.1;
a30 = 0.1;
a40 = 0.1;
a50 = 0.1;

% MCP joint on palm vector angle
d1 = alpha2/2;
```

```
d2 = alpha3*1/4;
d3 = alpha4*1/7;
d4 = alpha4*3/7;
d5 = alpha4*5/7;

% MCP joint to palm angle
g1 = 0;
g2 = -d2 - alpha4 + 90*q;
g3 = d3 - (alpha4 - 90*q);
g4 = d4 - (alpha4 - 90*q);
g5 = d5 - (alpha4 - 90*q);

% Finger link length
a11 = 0.5;
a12 = 0.5;
a13 = 0.5;
a21 = 0.5;
a22 = 0.5;
a23 = 0.5;
a31 = 0.5;
a32 = 0.5;
a33 = 0.5;

% Finger angle
theta11 = 0;
theta12 = 0;
theta13 = 0;
theta21 = 0;
theta22 = 0;
theta23 = 0;
theta31 = 0;
theta32 = 0;
theta33 = 0;

%% Palm Joint Coordinates (Points A, B, D and E)
% Z-axis Unit Vector
z = [0; 0; 1];
% Z-axis Vector including palm radious.
k = [0; 0; R];

% Rotation matrices, counter clockwise
[¬, RY1_ccw, RZ1_ccw] = Rotation(0, alpha1, theta1);
[¬, RY5_ccw, ¬]=Rotation(0, alpha5, 0);
% Rotation matrices, clockwise
[¬, RY4_cw, ¬] = Rotation(0, -alpha4, 0);
```

```
[¬, RY3_cw, ¬] = Rotation(0, -alpha3, 0);
[¬, ¬, RZ5_cw] = Rotation(0, 0, -theta5);

% Palm joint coordinates
pa_ccw = RY5_ccw * k;
pb_ccw = RY5_ccw * RZ1_ccw * RY1_ccw * k;
pd_cw = RZ5_cw * RY4_cw * k;
%pd_ccw = RY5_ccw * RZ1_ccw * RY1_ccw * RZ2_ccw * RY2_ccw ...
    * RZ3_ccw * RY3_ccw * k;
pe = k;

%% Joint Angles th2 and th4
ca1 = cos(alpha1);
ca2 = cos(alpha2);
ca3 = cos(alpha3);
ca4 = cos(alpha4);
ca5 = cos(alpha5);

sa1 = sin(alpha1);
sa2 = sin(alpha2);
sa3 = sin(alpha3);
sa4 = sin(alpha4);
sa5 = sin(alpha5);

BD_v = pb_ccw - pd_cw;
BD = sqrt(BD_v' * BD_v);
cBD = 1 - BD^2 / 2;
bd = acos( cBD );
sBD = sin(bd);

bcd = acos( (cBD - ca2*ca3) / (sa2 * sa3) );
theta3 = bcd - pi;

BE_v = pb_ccw - pe;
BE = sqrt( BE_v' * BE_v );
cBE = 1 - BE^2 / 2;
be = acos( cBE );
sBE = sin( be );

edb = acos( (cBE - ca4*cBD) / (sa4 * sBD) );
bdc = acos( (ca2 - ca3*cBD) / (sa3 * sBD) );

theta4 = (edb + bdc) - pi;

%
```

```
% Theta2 needs some work
%
AD_v = pa_ccw - pd_cw;
AD = sqrt(AD_v' * AD_v);
cAD = 1 - AD^2 / 2;
ad = acos( cAD );
sAD = sin( ad );

abd = acos( (cAD - ca1*cBD) / (sa1*sBD) );

dbc = acos( (ca3 - ca2*cBD) / (sa2*sBD) );

theta2 = -((abd + dbc) - pi);

%calc cont
[~, RY2_ccw, RZ2_ccw] = Rotation(0, alpha2, theta2);

[~, ~, RZ4_cw]=Rotation(0, 0, theta4);

pc_ccw = RY5_ccw * RZ1_ccw * RY1_ccw * RZ2_ccw * RY2_ccw * k;
pc_cw = RZ5_cw * RY4_cw * RZ4_cw * RY3_cw * k;

% verify th4 and th2
[~, RY2_cw, RZ3_cw] = Rotation(0, -alpha2, theta3);
pb_cw = RZ5_cw * RY4_cw * RZ4_cw * RY3_cw * RZ3_cw * ...
    RY2_cw * k;

[~, RY3_ccw, RZ3_ccw] = Rotation(0, alpha3, -theta3);
pd_ccw = RY5_ccw * RZ1_ccw * RY1_ccw * RZ2_ccw * RY2_ccw * ...
    RZ3_ccw * RY3_ccw * k;


%% Finger Origins (Points Mi)
% Rotation matrices for location on palm
[~, RY4f5_cw, ~] = Rotation(0, -(alpha4-d5), 0);
[~, RY4f4_cw, ~] = Rotation(0, -(alpha4-d4), 0);
[~, RY4f3_cw, ~] = Rotation(0, -(alpha4-d3), 0);
[~, RY3f2_cw, ~] = Rotation(0, -d2, 0);
[~, RY2f1_ccw, ~] = Rotation(0, d1, 0);

% finger #1 (Thumb)
RF1 = RY5_ccw * RZ1_ccw * RY1_ccw * RZ2_ccw * RY2f1_ccw;
DF1 = [RF1 RF1*k; 0 0 0 1];
DFM1 = [cos(g1) 0 -sin(g1) -a10*sin(g1);...
        0 1 0 0;...
```

```
        sin(g1) 0 cos(g1) a10*cos(g1);...
        0 0 0 1];
DM1 = DF1*DFM1;
r10 = DM1*[0;0;0;1];
r10 = r10(1:3,1);
r11 = RF1*k;

% finger #2 (Index)
RF2 = RZ5_cw * RY4_cw * RZ4_cw * RY3f2_cw;
DF2 = [RF2 RF2*k; 0 0 0 1];
DFM2 = [cos(g2) 0 -sin(g2) -a20*sin(g2);...
        0 1 0 0;...
        sin(g2) 0 cos(g2) a20*cos(g2);...
        0 0 0 1];
DM2 = DF2*DFM2;
r20 = DM2*[0;0;0;1];
r20 = r20(1:3,1);
r21 = RF2*k;

% finger #3
RF3 = RZ5_cw * RY4f3_cw;
DF3 = [RF3 RF3*k; 0 0 0 1];
DFM3 = [cos(g3) 0 -sin(g3) -a30*sin(g3);...
        0 1 0 0;...
        sin(g3) 0 cos(g3) a30*cos(g3);...
        0 0 0 1];
DM3 = DF3*DFM3;
r30 = DM3*[0;0;0;1];
r30 = r30(1:3,1);
r31 = RF3*k;

% finger #4
RF4 = RZ5_cw * RY4f4_cw;
DF4 = [RF4 RF4*k; 0 0 0 1];
DFM4 = [cos(g4) 0 -sin(g4) -a40*sin(g4);...
        0 1 0 0;...
        sin(g4) 0 cos(g4) a40*cos(g4);...
        0 0 0 1];
DM4 = DF4*DFM4;
r40 = DM4*[0;0;0;1];
r40 = r40(1:3,1);
r41 = RF4*k;

%finger #5
RF5 = RZ5_cw * RY4f5_cw;
```

```
DF5 = [RF5 RF5*k; 0 0 0 1];
DFM5 = [cos(g5) 0 -sin(g5) -a50*sin(g5);...
        0 1 0 0;...
        sin(g5) 0 cos(g5) a50*cos(g5);...
        0 0 0 1];
DM5 = DF5*DFM5;
r50 = DM5*[0;0;0;1];
r50 = r50(1:3,1);
r51 = RF5*k;

%% Fingertip Coordinates (Points Ti)
% Transformation from finger base to finger tip
[RX11,RY11,RZ11]=Rotation(theta11,0,0);
[RX12,RY12,RZ12]=Rotation(theta12,0,0);
[RX13,RY13,RZ13]=Rotation(theta13,0,0);
T11=[RX11, RX11*[0; 0; a11]; 0,0,0,1];
T12=[RX12, RX12*[0; 0; a12];0,0,0,1];
T13=[RX13, RX13*[0; 0; a13];0,0,0,1];
T1=T11*T12*T13;

%% plots
palm = [pa_ccw'; pb_ccw'; pc_cw'; pd_cw'; pe'];
fingers = [r10'; r20'; r30'; r40'; r50'];
mcp = [r11'; r21'; r31'; r41'; r51'];

plot3(palm(:,1),palm(:,2),palm(:,3),'bo')
hold on
plot3(fingers(:,1),fingers(:,2),fingers(:,3),'ro')
plot3(mcp(:,1),mcp(:,2),mcp(:,3),'co')

plot3(pa_ccw(1), pa_ccw(2), pa_ccw(3),'mx')

plot3(pb_ccw(1), pb_ccw(2), pb_ccw(3),'bx')
plot3(pb_cw(1), pb_cw(2), pb_cw(3),'b+')

plot3(pc_cw(1), pc_cw(2), pc_cw(3),'cx')
plot3(pc_ccw(1), pc_ccw(2), pc_ccw(3),'c+')

plot3(pd_cw(1), pd_cw(2), pd_cw(3),'gx')
plot3(pd_ccw(1), pd_ccw(2), pd_ccw(3),'g+')

plot3(pe(1), pe(2), pe(3),'rx')

axis([-2 2 -2 2 -2 2])
xlabel('x')
```

```
ylabel('y')
zlabel('z')
```