

Guy JOSEPH

Elbow Manipulator

```
#include <stdio.h>
#include <math.h>
#define d1 0
#define d6 1
#define a2 1
#define a3 1
#define PI 3.14159265359
double RtoD(double T) {return (double)(180.0*T)/PI;}
double DtoR(double T) {return (double)(T*PI)/180.0;}

//Foward Kinematics
void FK_IO();
void FK(float T1, float T2, float T3, float T4, float T5, float T6, float *r11, float *r12, float *r13, float *r21, float *r22, float *r23, float *r31, float *r32, float *r33, float *x, float *y, float *z);
void PrintMatrix(float r11, float r12, float r13, float r21, float r22, float r23, float r31, float r32, float r33, float x, float y, float z);

//Inverse Kinematics
void IK_IO();
void InputMatrix(float *r11, float *r12, float *r13, float *r21, float *r22, float *r23, float *r31, float *r32, float *r33, float *x, float *y, float *z);
void IK(float *T1, float *T2, float *T3, float *T4, float *T5, float *T6, float r11, float r12, float r13, float r21, float r22, float r23, float r31, float r32, float r33, float x, float y, float z);

//Velocity
void V_IO();
void V(float T1, float T2, float T3, float T4, float T5, float T6, float dT1, float dT2, float dT3, float dT4, float dT5, float dT6, float *vx, float *vy, float *vz, float *wx, float *wy, float *wz);

void main()
{
    int choice;

    while(choice)
    {
        choice=10;
        printf("\n 1 - Foward kinematics\n");
        printf(" 2 - Inverse kinematics\n");
        printf(" 3 - Velocity\n");
        printf(" 4 - Inverse Velocity\n");
        printf(" 5 - Acceleration\n");
        printf(" 6 - Inverse Acceleration\n");
        printf(" 0 - Quit\n");
        printf("Your choice: ");
        scanf("%d",&choice);

        switch (choice)
        {
            case 1:
                FK_IO();
                break;
            case 2:
                IK_IO();
                break;
            case 3:
                V_IO();
                break;
        }
    }
}

void FK_IO()
{
    float T1,T2,T3,T4,T5,T6;
    float r11,r12,r13;
    float r21,r22,r23;
    float r31,r32,r33;
    float x,y,z;

    printf("Enter the joint angle (T1 to T6)\n");
    scanf("%f %f %f %f %f %f",&T1,&T2,&T3,&T4,&T5,&T6);
    FK(T1,T2,T3,T4,T5,T6,&r11,&r12,&r13,&r21,&r22,&r23,&r31,&r32,&r33,&x,&y,&z);
    printf("Result...\n");
    PrintMatrix(r11,r12,r13,r21,r22,r23,r31,r32,r33,x,y,z);
}

void FK(float T1, float T2, float T3, float T4, float T5, float T6, float *r11, float *r12, float *r13, float *r21, float *r22, float *r23, float *r31, float *r32, float *r33, float *x, float *y, float *z)
```

✓ O.K
modules

need
a nice interface
for project
completion
+ Real time
set up
on MoveMaster
or SIAI

```

t *z)
{
    double c1,c2,c3,c4,c5,c6,s1,s2,s3,s4,s5,s6;

    c1=cos(DtoR(T1));
    c2=cos(DtoR(T2));
    c3=cos(DtoR(T3));
    c4=cos(DtoR(T4));
    c5=cos(DtoR(T5));
    c6=cos(DtoR(T6));

    s1=sin(DtoR(T1));
    s2=sin(DtoR(T2));
    s3=sin(DtoR(T3));
    s4=sin(DtoR(T4));
    s5=sin(DtoR(T5));
    s6=sin(DtoR(T6));

    *r11=c6 *(c5 *(c4 *(c1 *c2 *c3-c1 *s2 *s3)+(-c1 *c3 *s2-c1 *c2 *s3) *s4)-s1 *s5)+(c4 *(-c1 *c3 *
s2-c1 *c2 *s3)-(-c1 *c2 *c3-c1 *s2 *s3) *s4) *s6;
    *r12=c6 *(c4 *(-c1 *c3 *s2-c1 *c2 *s3)-(-c1 *c2 *c3-c1 *s2 *s3) *s4)-c5 *(c4 *(c1 *c2 *c3-c1 *s
2 *s3)+(-c1 *c3 *s2-c1 *c2 *s3) *s4)-s1 *s5) *s6;
    *r13=c5 *s1+(c4 *(c1 *c2 *c3-c1 *s2 *s3)+(-c1 *c3 *s2-c1 *c2 *s3) *s4) *s5;
    *x =a2 *c2 *c1+a3*c3 *c1 *c2-c1 *s2 *s3 *a3+d6 *(c5 *s1+(c4 *(c1 *c2 *c3-c1 *s2 *s3)+(-c1 *c3
*s2-c1 *c2 *s3) *s4) *s5);
    *r21=c6 *(c5 *(c4 *(c2 *c3 *s1-s1 *s2 *s3)+(-c3 *s1 *s2-c2 *s1 *s3) *s4)+c1 *s5)+( c4 *(-c3 *s1
*s2-c2 *s1 *s3)-(-c2 *c3 *s1-s1 *s2 *s3) *s4) *s6;
    *r22=c6 *(c4 *(-c3 *s1 *s2-c2 *s1 *s3)-(-c2 *c3 *s1-s1 *s2 *s3) *s4)-c5 *(c4 *(c2 *c3 *s1-s1 *s2
*s3)+(-c3 *s1 *s2-c2 *s1 *s3) *s4)+c1 *s5) *s6;
    *r23=-c1 *c5+(c4 *(c2 *c3 *s1-s1 *s2 *s3)+(-c3 *s1 *s2-c2 *s1 *s3) *s4) *s5;
    *y =a2 *c2 *s1+a3 *c3 *c2 *s1-s1 *s2 *s3 *a3+d6 *(-c1 *c5+(c4 *(c2 *c3 *s1-s1 *s2 *s3)+(-c3 *s
1 *s2-c2 *s1 *s3) *s4) *s5);
    *r31=c5 *c6 *(c4 *(c3 *s2+c2 *s3)+(c2 *c3-s2 *s3) *s4)+(c4 *(c2 *c3-s2 *s3)-(-c3 *s2+c2 *s3) *s4)
*s6;
    *r32=c6 *(c4 *(c2 *c3-s2 *s3)-(-c3 *s2+c2 *s3) *s4)- c5 *(c4 *(c3 *s2+c2 *s3)+(c2 *c3-s2 *s3) *s4
) *s6;
    *r33=(c4 *(c3 *s2+c2 *s3)+(c2 *c3-s2 *s3) *s4) *s5;
    *z =a3 *c3 *s2+s2*a2+c2 *s3 *a3+d6 *(c4 *(c3 *s2+c2 *s3)+(c2 *c3-s2 *s3) *s4) *s5;

}

void IK_IO()
{
    float T1,T2,T3,T4,T5,T6;
    float r11,r12,r13;
    float r21,r22,r23;
    float r31,r32,r33;
    float x,y,z;
    InputMatrix(&r11, &r12, &r13, &r21, &r22, &r23, &r31, &r32, &r33, &x, &y, &z);
    IK(&T1, &T2, &T3, &T4, &T5, &T6, r11, r12, r13, r21, r22, r23, r31, r32, r33, x, y, z);
    printf("T1=%.2f T2=%.2f T3=%.2f T4=%.2f T5=%.2f T6=%.2f\n",T1,T2,T3,T4,T5,T6);
}

void IK(float *T1, float *T2, float *T3, float *T4, float *T5, float *T6, float r11, float r12, float
r13, float r21, float r22, float r23, float r31, float r32, float r33, float x, float y, float z)
{
    float px,py,pz;
    float D;
    double temp,c1,s1,tempTC,tempTS,t1,t2,t3,t4,t5,t6,tt1,tt2;

    px=x-d6*r13;
    py=y-d6*r23;
    pz=z-d6*r33;

    temp=atan2(px,py);
    t1 =temp;

    D=(float)((double)(px*px+py*py+(pz-d1)*(pz-d1)-a2*a2-a3*a3)/2.0*a2*a3);
    temp=atan2(D,sqrt(1-D*D));
    t3 =float(temp);

    tempTC=cos(t3);
    tempTS=sin(t3);

    tt1=pz-d1;
    tt2=atan2(sqrt(px*px+py*py),tt1);
    tt1=atan2(a3*tempTS,a2+a3*tempTC);
    temp=tt2-tt1;
    t2=temp;

```

```

tempTC=cos(t2+t3);
tempTS=sin(t2+t3);
c1 =cos(t1);
s1 =sin(t1);
temp =atan2((c1*tempTC*r13+s1*tempTC*r23-tempTS*r23),(-c1*tempTS*r13-s1*tempTS*r23-tempTC*r33))
;
t4 =temp;

tempTC=s1*r13-c1*r23;
tempTC=tempTC*tempTC;
temp =atan2((s1*r13-c1*r23),sqrt(1-tempTC));
t5 =temp;

temp =atan2((s1*r11-c1*r21),(s1*r12+c1*r22));
t6 =temp;

*T1 = (float)RtoD(t1);
*T2 = (float)RtoD(t2);
*T3 = (float)RtoD(t3);
*T4 = (float)RtoD(t4);
*T5 = (float)RtoD(t5);
*T6 = (float)RtoD(t6);
}

void V_IO()
{
float T1, T2, T3, T4, T5, T6, dT1, dT2, dT3, dT4, dT5, dT6, vx, vy, vz, wx, wy, wz;
printf("Enter the joint angle (T1 to T6)\n");
scanf("%f %f %f %f %f %f",&T1,&T2,&T3,&T4,&T5,&T6);
printf("Enter the Velocity of the joint angle (dT1 to dT6)\n");
scanf("%f %f %f %f %f %f",&T1,&dT2,&dT3,&dT4,&dT5,&dT6);
V(T1, T2, T3, T4, T5, T6, dT1, dT2, dT3, dT4, dT5, dT6, &vx, &vy, &vz, &wx, &wy, &wz);
printf("Result...\n");
printf("vx=%.2f vy=%.2f vz=%.2f wx=%.2f wy=%.2f wz=%.2f\n",vx, vy, vz, wx, wy, wz);
}

void V(float T1, float T2, float T3, float T4, float T5, float T6, float dT1, float dT2, float dT3,
float dT4, float dT5, float dT6, float *vx, float *vy, float *vz, float *wx, float *wy, float *wz)
{
double c1,c2,c3,c4,c5,c6,s1,s2,s3,s4,s5,s6;

c1=cos(DtoR(T1));
c2=cos(DtoR(T2));
c3=cos(DtoR(T3));
c4=cos(DtoR(T4));
c5=cos(DtoR(T5));
c6=cos(DtoR(T6));

s1=sin(DtoR(T1));
s2=sin(DtoR(T2));
s3=sin(DtoR(T3));
s4=sin(DtoR(T4));
s5=sin(DtoR(T5));
s6=sin(DtoR(T6));

*vx=(c1*c5*d6-a2*c2*s1-a3*c2*c3*s1+a3*s1*s2*s3-c2*c3*c4*d6*s1*s5+c4*d6*s1*s2*s3*s5+c3*d6*s1*s2*s4*s5+c2*d6*s1*s3*s4*s5)*dT1 + (-a2*c1*s2-a3*c1*c3*s2-a3*c1*c2*s3-c1*c3*c4*d6*s2*s5-c1*c2*c4*d6*s3*s5-c1*c2*c3*d6*s4*s5+c1*d6*s2*s3*s4*s5)*dT2 + (-a3*c1*c3*s2-a3*c1*c2*s3-c1*c3*c4*d6*s2*s5-c1*c2*c4*d6*s3*s5-c1*c2*c3*d6*s4*s5+c1*d6*s2*s3*s4*s5)*dT3 + (-c1*c3*c4*d6*s2*s5-c1*c2*c4*d6*s3*s5-c1*c2*c3*d6*s4*s5+c1*d6*s2*s3*s4*s5)*dT4;
*vy=(a2*c1*c2+a3*c1*c2*c3+c5*d6*s1-a3*c1*s2*s3+c1*c2*c3*c4*d6*s5-c1*c4*d6*s2*s3*s5-c1*c3*d6*s2*s4*s5-c1*c2*d6*s3*s4*s5)*dT1 + (-a2*s1*s2-a3*c3*s1*s2-a3*c2*s1*s3-c3*c4*d6*s1*s2*s5-c2*c4*d6*s1*s3*s5-d6*s1*s2*s3*s4*s5)*dT2 + (-a3*c3*s1*s2-a3*c2*s1*s3-c3*c4*d6*s1*s2*s5-c2*c4*d6*s1*s3*s5-c2*c3*d6*s1*s4*s5+d6*s1*s2*s3*s4*s5)*dT3 + (-c3*c4*d6*s1*s2*s5-c2*c4*d6*s1*s3*s5-c2*c3*d6*s1*s4*s5+d6*s1*s2*s3*s4*s5)*dT4;
*vz=(c1*c1*c3*c4*c5*d6*s2+c3*c4*c5*d6*s1*s1*s2+c1*c1*c2*c4*c5*d6*s3+c2*c4*c5*d6*s1*s1*s3+c1*c2*c3*c5*d6*s4+c2*c3*c5*d6*s1*s4-c1*c5*d6*s2*s3*s4-c5*d6*s1*s2*s3*s4)*dT4;
*wx=s1*dT2+s1*dT3+s1*dT4+(c4*(-c1*c3*s2-c1*c2*s3)-(c1*c2*c3-c1*s2*s3)*s4)*dT5+(c5*s1+(c4*(c1*c2*c3-c1*s2*s3)+(-c1*c3*s2-c1*c2*s3)*s4)*s5)*dT6;
*wy=-c1*dT2-c1*dT3-c1*dT4+(c4*(-c3*s1*s2-c2*s1*s3)-(c2*c3*s1-s1*s2*s3)*s4)*dT5+(-c1*c5+(c4*(c2*c3*s1-s1*s2*s3)+(-c3*s1*s2-c2*s1*s3)*s4)*s5)*dT6;
*wz=(c4*(c2*c3-s2*s3)-(c3*s2+c2*s3)*s4)*dT5+((c4*(c3*s2+c2*s3)+(c2*c3-s2*s3)*s4)*s5)*dT6;
}

void InputMatrix(float *r11, float *r12, float *r13, float *r21, float *r22, float *r23, float *r31,
float *r32, float *r33, float *x, float *y, float *z)

```

```

{
    float tr11, tr12, tr13;
    float tr21, tr22, tr23;
    float tr31, tr32, tr33;
    float tx, ty, tz;

    printf("Enter row 1 (col 1 to 4)\n");
    scanf(" %f %f %f %f", &tr11, &tr12, &tr13, &tx);
    printf("Enter row 2 (col 1 to 4)\n");
    scanf(" %f %f %f %f", &tr21, &tr22, &tr23, &ty);
    printf("Enter row 3 (col 1 to 4)\n");
    scanf(" %f %f %f %f", &tr31, &tr32, &tr33, &tz);

    *r11=tr11;
    *r12=tr12;
    *r13=tr13;

    *r21=tr21;
    *r22=tr22;
    *r23=tr23;

    *r31=tr31;
    *r32=tr32;
    *r33=tr33;

    *x=tx;
    *y=ty;
    *z=tz;
}

void PrintMatrix(float r11, float r12, float r13, float r21, float r22, float r23, float r31, float
r32, float r33, float x, float y, float z)
{
    printf(" %9.2f %9.2f %9.2f %9.2f\n", r11, r12, r13, x);
    printf(" %9.2f %9.2f %9.2f %9.2f\n", r21, r22, r23, y);
    printf(" %9.2f %9.2f %9.2f %9.2f\n", r31, r32, r33, z);
    printf("      0      0      0      1\n");
}

```

Untitled

```

A1={{c1,0,s1,0},{s1,0,-c1,0},{0,1,0,0},{0,0,0,1}};
A2={{c2,-s2,0,c2a2},{s2,c2,0,a2s2},{0,0,1,0},{0,0,0,1}};
A3={{c3,-s3,0,c3a3},{s3,c3,0,a3s3},{0,0,1,0},{0,0,0,1}};
A4={{c4,0,-s4,0},{s4,0,c4,0},{0,-1,0,0},{0,0,0,1}};
A5={{c5,0,s5,0},{s5,0,-c5,0},{0,1,0,0},{0,0,0,1}};
A6={{c6,-s6,0,0},{s6,c6,0,0},{0,0,1,d6},{0,0,0,1}};
R1={{c1,0,s1},{s1,0,-c1},{0,1,0}};
R2={{c2,-s2,0},{s2,c2,0},{0,0,1}};
R3={{c3,-s3,0},{s3,c3,0},{0,0,1}};
R4={{c4,0,-s4},{s4,0,c4},{0,-1,0}};
R5={{c5,0,s5},{s5,0,-c5},{0,1,0}};
R6={{c6,-s6,0},{s6,c6,0},{0,0,1}};
k={0,0,1};
TO1=A1;
TO2=A1.A2;
TO3=A1.A2.A3;
TO4=A1.A2.A3.A4;
TO5=A1.A2.A3.A4.A5;
TO6=A1.A2.A3.A4.A5.A6;
Z0=k;
Z1=R1.k;
Z2=R1.R2.k;
Z3=R1.R2.R3.k;
Z4=R1.R2.R3.R4.k;
Z5=R1.R2.R3.R4.R5.k;
Oo = {0, 0, 0};
O1 = {0,0,0};
O2 = {c1 c2 a2,c2 a2 s1,a2 s2};
O3 = {c1 c2 a2+c1 c2 c3 a3-a3 s3 c1 s2,
      c2 a2 s1+c2 c3 a3 s1-a3 s3 s1 s2,a2 s2+a3 s3 c2+c3 a3 s2};
O4 = {c1 c2 a2+c1 c2 c3 a3-a3 s3 c1 s2,
      c2 a2 s1+c2 c3 a3 s1-a3 s3 s1 s2,a2 s2+a3 s3 c2+c3 a3 s2};
O5 = {c1 c2 a2+c1 c2 c3 a3-a3 s3 c1 s2,
      c2 a2 s1+c2 c3 a3 s1-a3 s3 s1 s2,a2 s2+a3 s3 c2+c3 a3 s2};
O6 = {c1 c2 a2+c1 c2 c3 a3-a3 s3 c1 s2+
      d6 (c5 s1+(c4 (c1 c2 c3-c1 s2 s3))+(-c1 c3 s2-c1 c2 s3) s4) s5),
      c2 a2 s1+c2 c3 a3 s1-a3 s3 s1 s2+
      d6 (-c1 c5+(c4 (c2 c3 s1-s1 s2 s3))+(-c3 s1 s2-c2 s1 s3) s4) s5),
      a2 s2+a3 s3 c2+c3 a3 s2+d6 (c4 (c3 s2+c2 s3)+(c2 c3-s2 s3) s4) s5}
;
O6mOo=O6;
O6mO1=O6-O1;
O6mO2=O6-O2;
O6mO3=O6-O3;
O6mO4=O6-O4;
O6mO5=O6-O5;
Cr0=Cross[Z0,O6mOo];MatrixForm[Cr0]
Cr1=Cross[Z1,O6mO1];MatrixForm[Cr1]
Cr2=Cross[Z2,O6mO2];MatrixForm[Cr2]
Cr3=Cross[Z3,O6mO3];MatrixForm[Cr3]
Cr4=Cross[Z4,O6mO4];MatrixForm[Cr4]

```

```

{
    float tr11,tr12,tr13;
    float tr21,tr22,tr23;
    float tr31,tr32,tr33;
    float tx,ty,tz;

    printf("Enter row 1 (col 1 to 4)\n");
    scanf(" %f %f %f %f",&tr11,&tr12,&tr13,&tx);
    printf("Enter row 2 (col 1 to 4)\n");
    scanf(" %f %f %f %f",&tr21,&tr22,&tr23,&ty);
    printf("Enter row 3 (col 1 to 4)\n");
    scanf(" %f %f %f %f",&tr31,&tr32,&tr33,&tz);

    *r11=tr11;
    *r12=tr12;
    *r13=tr13;

    *r21=tr21;
    *r22=tr22;
    *r23=tr23;

    *r31=tr31;
    *r32=tr32;
    *r33=tr33;

    *x=tx;
    *y=ty;
    *z=tz;
}

void PrintMatrix(float r11, float r12, float r13, float r21, float r22, float r23, float r31, float
r32, float r33, float x, float y, float z)
{
    printf(" %9.2f %9.2f %9.2f %9.2f\n",r11,r12,r13,x);
    printf(" %9.2f %9.2f %9.2f %9.2f\n",r21,r22,r23,y);
    printf(" %9.2f %9.2f %9.2f %9.2f\n",r31,r32,r33,z);
    printf("      0      0      0      1\n");
}

```