/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROGRAM DESCRIPTION : THIS PROGRAM GIVE DIRECT KINAMETICS PARAMETER

OF TWO AXIS ARTICULATED ROBOT.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

#include<ctype.h>

#include<process.h>

# define pi 22.0/7.0

void main()

{

float p[4],t1,t2,l[4][4],l1[4][4],l2[4][4],l3[4][4],lf[4][4];

char message[100],choice;

int gd=DETECT,gm,n,a,b,y,z,x,m;

l[0][0]=0; l[0][1]=0;

initgraph(&gd,&gm,"c:\\tc\\bgi");

setcolor(MAGENTA);//setlinestyle(DOTTED\_LINE,1,3);

rectangle(50,40,580,455); rectangle(60,30,570,470);

setcolor(YELLOW);

settextstyle(3,0,4);outtextxy(60,70," DIRECT KINEMATICS PROBLEM OF");

outtextxy(80,110,"TWO AXIS ARTICULATED ROBOT");

getch();

// exit(0);

clearviewport();

setcolor(RED);settextstyle(7,0,3); outtextxy(30,30,"INSTRUCTIONS");

setcolor(RED);setlinestyle(SOLID\_LINE,1,3);line(30,60,210,60);

setcolor(WHITE);settextstyle(2,0,6);

outtextxy(30,100,"1. Enter the base and shoulder link lengths");

outtextxy(30,130,"2. Enter the base and shoulder angles");

outtextxy(30,160,"3. Draw the Link Coordinate Diagram of 2 axes Articulated Robot");

outtextxy(30,190,"4. Draw the Kinetic Parameter Table");

outtextxy(30,220,"5. Calculate the Link Coordinate Transformation Matrix");

outtextxy(30,250,"6. To check the matrix obtained is correct or not,evaluate it");

outtextxy(30,270," it at the Soft Home Position [SHP] by using last column of");

outtextxy(30,290," KP table in T");outtextxy(183,297,"0");outtextxy(183,282,"2");

outtextxy(30,320,"7. Output the Final Arm Matrix");

outtextxy(30,350,"8. Output the Position Matrix");

getch();

clearviewport();

// exit(0);

setcolor(BLUE); rectangle(60,90,580,440); setfillstyle(10,BLUE);

floodfill(3,3,1); setcolor(14);

rectangle(60,90,580,440); rectangle(63,93,577,437);

line(63,290,577,290); line(63,330,577,330);

floodfill(65,300,14);

settextstyle(1,0,3); setcolor(WHITE);

outtextxy(150,30,"Direct Kinematics Problem");

rectangle(290,170,350,210);

line(260,190,290,190);line(284,187,290,190);line(284,193,290,190);

line(320,210,320,240);line(317,217,320,210);line(323,217,320,210);

line(350,190,380,190);line(374,187,380,190);line(374,193,380,190);

settextstyle(SMALL\_FONT,0,4);

outtextxy(196,179,"JOINT"); outtextxy(186,189,"PARAMETERS");

outtextxy(393,179,"POSITION &"); outtextxy(395,189,"ORIENTATION");

outtextxy(280,245,"GEOMETRIC LINK"); outtextxy(280,258,"PARAMETERS");

setcolor(CYAN); settextstyle(TRIPLEX\_FONT,0,1); outtextxy(299,178,"D.K.P");

settextstyle(2,0,6); setcolor(11);

outtextxy(150,300,"THE LINK AND JOINT PARAMETERS ARE:-");

setcolor(WHITE); settextstyle(2,0,5);

outtextxy(80,350,"BASE ROTATION ANGLE -q1 BASE LINK LENGHT -a1");

outtextxy(80,365,"SHOULDER ROTATION ANGLE -q2 SHOULDER LINK LENGHT -a2");

settextstyle(SMALL\_FONT,0,5);

outtextxy(420,450,"PRESS ANY KEY TO CONTINUE....");

getch();

cleardevice();

settextstyle(1,0,3);

outtextxy(200,50,"ENTER PARAMETERS:");

setcolor(YELLOW);

settextstyle(2,0,5);

printf("\n\n\n\n\n\n\n\n\n");

for(n=1;n<3;++n)

{

if(n==1)

sprintf(message,"ENTER BASE LINK LENGTH:");

else

{

if(n==2)

sprintf(message,"ENTER SHOULDER LINK LENGTH:");

}

outtextxy(210,115+(46\*n),message);

printf("\n\n\t\t\t\t ");

scanf("%f",&l[n][0]);

l[n][0]=l[n][0]\*(25);

}

gotoxy(0,0);

for(n=1;n<3;++n)

{

if(n==1)

sprintf(message,"ENTER BASE ROTATION ANGLE:");

else

{

if(n==2)

sprintf(message,"ENTER SHOULDER ROTATION ANGLE:");

}

outtextxy(200,200+(46\*n),message);

printf("\n\n\t\t\t\t ");

scanf

("%f",&l[n][1]);

if(n==3)

l[n][1]=l[n-1][1]+((l[n][1]+90)\*pi/180);

else

l[n][1]=l[n-1][1]+(l[n][1]\*pi/180);

}

z=l[1][0];

for(a=0;a<4;++a)

{

for(b=0;b<4;++b)

{

l1[a][b]=0;l2[a][b]=0;

lf[a][b]=0;

}

}

cleardevice();

l[1][0]=z;

settextstyle(2,0,6);

outtextxy(50,90,"BY FIRST PASS OF D-H ALGORITHM WE GET THE LCD AS FLLOWS");

rectangle(20,70,630,430); floodfill(5,5,14);

settextstyle(1,0,3);

setcolor(WHITE); outtextxy(150,20,"LINK COORDINATE DIAGRAM");

setcolor(BROWN); setlinestyle(0,0,2); line(150,50,450,50);

setcolor(WHITE);

settextstyle(2,0,5);

x=160; y=350;

circle(x,y,2.2);

setcolor(11); setlinestyle(SOLID\_LINE,0,THICK\_WIDTH);

line(x,y,x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3));

setcolor(WHITE);

outtextxy(x,y+20,"BASE");

setcolor(BROWN); outtextxy(x,y,"J1");

setcolor(11);

line(x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3),x+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3));

setcolor(WHITE);

circle(x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3),3.2);

outtextxy(x+l[1][0]\*cos(pi/3)-30,y-l[1][0]\*sin(pi/3)-70,"SHOULDER");

setcolor(BROWN);outtextxy(x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3),"J2");

setcolor(11); setlinestyle(1,0,3);

setcolor(11);

setcolor(WHITE);

circle(x+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3),3.2);

outtextxy(x+l[1][0]\*cos(pi/3)+l[2][0]-30,y-l[1][0]\*sin(pi/3)-70,"TOOL ");

setcolor(11);

outtextxy(x+l[1][0]\*cos(pi/3)/2,y-l[1][0]\*sin(pi/3)/2,"a1");

outtextxy(x+l[1][0]\*cos(pi/3)+l[2][0]/2,5+y-l[1][0]\*sin(pi/3),"a2");

setcolor(RED); setlinestyle(0,0,1);

line(x,y,x,y-50);line(x,y,x+50,y);

line(204,353,210,350); line(204,347,210,350); line(x-3,306,x,300); line(x+3,306,x,300);

outtextxy(x-19,y-45,"y0");

outtextxy(x+50,y+5,"x0");

line(x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3),x+(l[1][0]+50)\*cos(pi/3),y-(l[1][0]+50)\*sin(pi/3));

line(x+l[1][0]\*cos(pi/3),y-l[1][0]\*sin(pi/3),x+(l[1][0])\*cos(pi/3)-50\*cos(pi/6),y-(l[1][0])\*sin(pi/3)-50\*sin(pi/6));

outtextxy(x+(l[1][0]+50)\* cos(pi/3),y-(l[1][0])\*sin(pi/3)-50,"x1");

outtextxy(5+x+(l[1][0])\* cos(pi/3)-50\*cos(pi/6),-14+y-(l[1][0])\*sin(pi/3)-50\*sin(pi/60),"y1");

line(x+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3),50+x+l[1][0]\*cos(pi/3)+l[2][0],y-(l[1][0])\*sin(pi/3));

line(x+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3),x+l[1][0]\*cos(pi/3)+l[2][0],-50+y-(l[1][0])\*sin(pi/3));

line(x+l[1][0]\*cos(pi/3)-3+l[2][0],-50+y-l[1][0]\*sin(pi/3)+6,x+l[1][0]\*cos(pi/3)+l[2][0],-50+y-(l[1][0])\*sin(pi/3));

line(x+l[1][0]\*cos(pi/3)+3+l[2][0],-50+y-l[1][0]\*sin(pi/3)+6,x+l[1][0]\*cos(pi/3)+l[2][0],-50+y-(l[1][0])\*sin(pi/3));

line(x+50+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3),x+44+l[1][0]\*cos(pi/3)+l[2][0],y-3-(l[1][0])\*sin(pi/3));

line(x+50+l[1][0]\*cos(pi/3)+l[2][0],y-l[1][0]\*sin(pi/3),x+44+l[1][0]\*cos(pi/3)+l[2][0],y+3-(l[1][0])\*sin(pi/3));

outtextxy(x+55+l[1][0]\*cos(pi/3)+l[2][0],y-5-l[1][0]\*sin(pi/3),"x2");

outtextxy(5+x+l[1][0]\*cos(pi/3)+l[2][0],-50+y-l[1][0]\*sin(pi/3),"y2");

outtextxy(420,450,"PRESS ANY KEY TO CONTINUE....");

getch();

cleardevice();

settextstyle(1,0,3);

setcolor(WHITE); outtextxy(150,50,"KINEMATIC PARAMETER TABLE ");

setcolor(BROWN); setlinestyle(0,0,2); line(150,80,490,80);

setcolor(1); setlinestyle(0,0,3); rectangle(50,200,600,350);

line(50,250,600,250); setlinestyle(1,0,2);

line(50,300,600,300);

line(120,200,120,350); line(207,200,207,350); line(280,200,280,350); line(360,200,360,350); line(440,200,440,350); line(520,200,520,350);

setcolor(11);

settextstyle(2,0,6); outtextxy(50,160,"BY SECOND PASS OF D-H ALGORITHM WE GET THE KPT AS FOLLOWS");

setcolor(WHITE); settextstyle(3,0,1);

outtextxy(65,215,"AXIS TYPE q(k) d(k) a(k) A(k) HOME");

sprintf(message," 1 BASE %0.f 0 %0.f 0 60 ",l[1][1]\*1260/22,l[1][0]/25);

outtextxy(65,265,message);

sprintf(message," 2 SHOULDER %0.f 0 %0.f 0 -60",(l[2][1]-l[1][1])\*1260/22,l[2][0]/25);

outtextxy(65,315,message);

outtextxy(400,450,"PRESS ANY KEY TO CONTINUE....");

getch();

cleardevice();

l1[1][1]=cos(l[1][1]); l1[1][2]=-sin(l[1][1]);

l1[2][1]=sin(l[1][1]); l1[2][2]=cos(l[1][1]);

l1[3][3]=1;l1[4][4]=1;l1[1][4]=z\*cos(l[1][1])/25; l1[2][4]=z\*sin(l[1][1])/25;

l2[1][1]=cos(l[2][1]-l[1][1]); l2[1][2]=-sin(l[2][1]-l[1][1]);

l2[2][1]=-l2[1][2]; l2[2][2]=l2[1][1];

l2[3][3]=1;l2[4][4]=1;l2[1][4]=l[2][0]\*cos(l[2][1]-l[1][1])/25; l2[2][4]=l[2][0]\*sin(l[2][1]-l[1][1])/25;

lf[1][1]=sin(l[2][1]); lf[1][2]=-cos(l[2][1]);

lf[2][1]= lf[1][2]; lf[2][2]= lf[1][1]; lf[3][3]=1; lf[4][4]=1;

lf[3][4]=l[2][0]/25;

lf[1][4]=l1[1][4]+l[1][0]\*cos(l[2][1])/25;

lf[2][4]= l1[2][4] + l[2][0]\*sin(l[2][1])/25;

settextstyle(1,0,3);

setcolor(LIGHTCYAN); outtextxy(175,130,"TRANSFORMATION MATRICES");

setcolor(WHITE);setcolor(BROWN);setlinestyle(0,0,1.5);line(175,160,490,160);

settextstyle(3,0,3);outtextxy(50,240,"T =");

outtextxy(350,240,"T =");

setcolor(LIGHTRED);settextstyle(2,0,4);

outtextxy(65,240,"1");outtextxy(65,263,"0");

outtextxy(365,240,"2");outtextxy(365,263,"1");

settextstyle(2,0,5);setcolor(WHITE);

for(a=1;a<5;++a)

{

for(b=1;b<5;++b)

{

sprintf(message,"%4.2f",l1[a][b]);

outtextxy(60+(50\*b),200+(20\*a),message);

}

}

setcolor(BROWN);

setlinestyle(0,0,1);

line(100,215,100,300);line(100,215,110,215);line(100,300,110,300);

line(295,215,295,300);line(285,215,295,215);line(285,300,295,300);

setcolor(WHITE);

for(a=1;a<5;++a)

{

for(b=1;b<5;++b)

{

sprintf(message,"%4.2f",l2[a][b]);

outtextxy(358+(50\*b),200+(20\*a),message);

}

}

setcolor(BROWN);

setlinestyle(0,0,1);

line(400,215,400,300);line(400,215,410,215);line(400,300,410,300);

line(595,215,595,300);line(585,215,595,215);line(585,300,595,300);

getch();

cleardevice();

settextstyle(1,0,4);

setcolor(LIGHTCYAN);

rectangle(150,150,490,300);

setfillstyle(10,1);floodfill(5,5,LIGHTCYAN);

setcolor(LIGHTCYAN);outtextxy(176,60,"FINAL ARM MATRIX");

setcolor(BROWN);setlinestyle(0,0,3);line(176,97,460,97);

setcolor(WHITE);settextstyle(3,0,3);outtextxy(200,210,"T =");

setcolor(LIGHTRED);settextstyle(2,0,4);outtextxy(205,205,"TOOL");outtextxy(205,240,"BASE");

setcolor(WHITE);settextstyle(2,0,5);lf[4][4]=1;

for(a=1;a<5;++a)

{

for(b=1;b<5;++b)

{

sprintf(message,"%4.2f",lf[a][b]);

outtextxy(220+(50\*b),170+(20\*a),message);

}}

setcolor(BROWN);

setlinestyle(0,0,1);

line(260,185,260,270);line(260,185,270,185);line(260,270,270,270);

line(455,185,455,270);line(445,185,455,185);line(445,270,455,270);

settextstyle(1,0,2);setcolor(WHITE);

outtextxy(60,320,"Position Matrix : \n");

settextstyle(1,0,1);setcolor(WHITE);

outtextxy(60,350,"P = ");

for(a=1;a<=4;++a){

sprintf(message,"%4.2f",lf[a][4]);

outtextxy(50+(70\*a),350,message);

}

line(105,350,105,375); line(105,375,115,375); line(105,350,115,350);

line(380,350,380,375); line(370,375,380,375); line(370,350,380,350);

setcolor(WHITE);settextstyle(2,0,5);

outtextxy(30,400,"This matrix maps frame L1 (Tooltip) coordinates into frame L0 (Base) coordinates");

outtextxy(30,420,"this is the solution of the Direct Kinematics problem,");

getch();

}













