ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Computer Aided Generalized Method of Synthesis of Four Bar Mechanism

Prashim K. Kamble*1, Dr. C.C.Handa2, P.N.Zode3
[*1]M.Tech, [2]H.O.D, [3]Assitant Proffesor, K.D.K.C.E., Nagpur, India
Prashim.friend@gmail.com

Abstract

A new computerised dimensional synthesis method is described in the paper. To synthesis of a four bar mechanism, knowledge of function, path, motion generation method is required. Synthesis of four bar mechanism depends upon the graphical as well as analytical methods. The graphical approach is suitable for finding out the length of links of a mechanism with two or three precession point. However, to find out the values of link of four bar using three position synthesis analytically at various position it required lots of calculation and it take more time. To get values of all link of a four bar mechanism at different positions of the crank, a computer program is developed. This paper deal with synthesis of a four bar mechanism to determine values of all four link of four bar mechanism at different positions of the crank using digital computer

Keywords: Mechanism, Four bar chain mechanism

Introduction

Synthesis and the investigation of coupler curve have been paid close attention by a fair number of mechanism researchers and a lot of technical papers were published. In general, 9 precise coupler-points can be realized for a four-bar synthesis mechanism. As to coupler-points more than 9 cannot be realized accurately. With the development of computer's calculating speed and memory capacity, many new and approximate synthesis methods were presented. In essence, almost all of them use a group parameters or functional formula to describe the shape characteristics of a coupler curve. Theoretically, a four-bar linkage with same dimensions can generate infinite coupler curves. Therefore, to achieve a relative precise synthesis, a very large number of coupler curves need to be stored in the coupler curves database.

To synthesis of a four bar mechanism, knowledge of function, path, motion generation method is required. Synthesis of four bar mechanism depends upon the graphical as well as analytical methods. Analytical and graphical method is used to synthesis of mechanism. For that, a computer program is prepared to solve this problem and to get the link length at different positions of the crank.

Programmable Calculation

For the synthesis of a four bar mechanism, we consider a problem of which, we have to find out solutions at an instant for various positions of the crank with some interval with the help of computer program in

„C language.

```
include"iostream.h"
#include"conio.h"
#include"math.h"
#include"graphics.h"
#include"dos.h"
double inv[16], det, m[16],invOut[16],n[16],y[16];
int i;
char name[20];
void myline()
{       cout<<endl;
        for(i=1;i<=70;i++)
        {cout<<"=";
        }
        cout<<endl;
}
void login()
{       myline();
        cout<<"Logged User : "<<name;
        myline();
}
void calculate()
{
  inv[0] = m[5]  * m[10] * m[15] -
           m[5]  * m[11] * m[14] -
           m[9]  * m[6]  * m[15] +
           m[9]  * m[7]  * m[14] +
           m[13] * m[6]  * m[11] -
           m[13] * m[7]  * m[10];
 inv[4] = -m[4]  * m[10] * m[15] +
           m[4]  * m[11] * m[14] +
           m[8]  * m[6]  * m[15] -
```

```
            m[8]  * m[7]  * m[14] -
            m[12] * m[6]  * m[11] +
            m[12] * m[7]  * m[10];
inv[8] = m[4]  * m[9] * m[15] -
            m[4]  * m[11] * m[13] -
            m[8]  * m[5] * m[15] +
            m[8]  * m[7] * m[13] +
            m[12] * m[5] * m[11] -
            m[12] * m[7] * m[9];
inv[12] = -m[4]  * m[9] * m[14] +
             m[4]  * m[10] * m[13] +
             m[8]  * m[5] * m[14] -
             m[8]  * m[6] * m[13] -
             m[12] * m[5] * m[10] +
             m[12] * m[6] * m[9];
inv[1] = -m[1]  * m[10] * m[15] +
            m[1]  * m[11] * m[14] +
            m[9]  * m[2] * m[15] -
            m[9]  * m[3] * m[14] -
            m[13] * m[2] * m[11] +
            m[13] * m[3] * m[10];
inv[5] = m[0]  * m[10] * m[15] -
            m[0]  * m[11] * m[14] -
            m[8]  * m[2] * m[15] +
            m[8]  * m[3] * m[14] +
            m[12] * m[2] * m[11] -
            m[12] * m[3] * m[10];
inv[9] = -m[0]  * m[9] * m[15] +
            m[0]  * m[11] * m[13] +
            m[8]  * m[1] * m[15] -
            m[8]  * m[3] * m[13] -
            m[12] * m[1] * m[11] +
            m[12] * m[3] * m[9];
inv[13] = m[0]  * m[9] * m[14] -
            m[0]  * m[10] * m[13] -
            m[8]  * m[1] * m[14] +
            m[8]  * m[2] * m[13] +
            m[12] * m[1] * m[10] -
            m[12] * m[2] * m[9];
inv[2] = m[1]  * m[6] * m[15] -
            m[1]  * m[7] * m[14] -
            m[5]  * m[2] * m[15] +
            m[5]  * m[3] * m[14] +
            m[13] * m[2] * m[7] -
            m[13] * m[3] * m[6];
inv[6] = -m[0]  * m[6] * m[15] +
            m[0]  * m[7] * m[14] +
            m[4]  * m[2] * m[15] -
            m[4]  * m[3] * m[14] -
            m[12] * m[2] * m[7] +
            m[12] * m[3] * m[6];
inv[10] = m[0]  * m[5] * m[15] -
            m[0]  * m[7] * m[13] -
            m[4]  * m[1] * m[15] +
            m[4]  * m[3] * m[13] +
            m[12] * m[1] * m[7] -
            m[12] * m[3] * m[5];
inv[14] = -m[0]  * m[5] * m[14] +
             m[0]  * m[6] * m[13] +
             m[4]  * m[1] * m[14] -
             m[4]  * m[2] * m[13] -
             m[12] * m[1] * m[6] +
             m[12] * m[2] * m[5];
inv[3] = -m[1] * m[6] * m[11] +
            m[1] * m[7] * m[10] +
            m[5] * m[2] * m[11] -
            m[5] * m[3] * m[10] -
            m[9] * m[2] * m[7] +
            m[9] * m[3] * m[6];
inv[7] = m[0] * m[6] * m[11] -
            m[0] * m[7] * m[10] -
            m[4] * m[2] * m[11] +
            m[4] * m[3] * m[10] +
            m[8] * m[2] * m[7] -
            m[8] * m[3] * m[6];
inv[11] = -m[0] * m[5] * m[11] +
             m[0] * m[7] * m[9] +
             m[4] * m[1] * m[11] -
             m[4] * m[3] * m[9] -
             m[8] * m[1] * m[7] +
             m[8] * m[3] * m[5];
inv[15] = m[0] * m[5] * m[10] -
            m[0] * m[6] * m[9] -
            m[4] * m[1] * m[10] +
            m[4] * m[2] * m[9] +
            m[8] * m[1] * m[6] -
            m[8] * m[2] * m[5];
det = m[0] * inv[0] + m[1] * inv[4] + m[2] * inv[8] +
m[3] * inv[12];
if (det == 0)
{cout<<"invers is not possible";
}
else
{det = 1.0 / det;
        for (i = 0; i < 16; i++)
        {invOut[i] = inv[i] * det;
        }
}
cout<<"Inverse of the input matrix is: "<<"\n";
for(i=0;i<16;i++)
{cout<<"  "<<invOut[i];
        if((i+1)%4==0)
        {       cout<<endl;
                delay(50);
        }
}
void sides()
```

```
{y[0] = n[0]*invOut[0] + n[1]*invOut[1] + n[2]*invOut[2] + n[3]*invOut[3] ;
y[1] = n[0]*invOut[4] + n[1]*invOut[5] + n[2]*invOut[6] + n[3]*invOut[7] ;
y[2] = n[0]*invOut[8] + n[1]*invOut[9] + n[2]*invOut[10] + n[3]*invOut[11] ;
y[3] = n[0]*invOut[12] + n[1]*invOut[13] + n[2]*invOut[14] + n[3]*invOut[15] ;
}
//------------------------------------------------------------------------
----
void main()
{clrscr();
int A2,A3, B2,B3, C2,C3, D2,D3, p21,p31;
double wx,wy, zx,zy, ux,uy, sx,sy;
double w,z, u,s;
double theta,fi, theta1,fi1, sigma,sigma1, si,si1;
        int option;
        start:
        clrscr();
        cout<<"\t\t\tUSER LOGIN";
        myline();
        cout<<"Enter the User Name: ";
        cin>>name;
        clrscr();
        login();
cout<<(char)224<<"2 :";cin>>A2;
cout<<(char)224<<"3 :";cin>>A3;
cout<<(char)225<<"2 :";cin>>B2;
cout<<(char)225<<"3 :";cin>>B3;
cout<<(char)235<<"2 :";cin>>C2;
cout<<(char)235<<"3 :";cin>>C3;
cout<<(char)231<<"2 :";cin>>D2;
cout<<(char)231<<"3 :";cin>>D3;
cout<<"Enter the distance between the FIRST and SECOND image P21: ";
cin>>p21;
cout<<"Enter the distance between the FIRST and THIRD image P31: ";
cin>>p31;
myline();
m[0]=(cos((3.14/180)*B2)-1);
m[1]= -sin((3.14/180)*B2);
m[2]=(cos((3.14/180)*A2)-1);
m[3]= -sin((3.14/180)*A2);
m[4]=(cos((3.14/180)*B3)-1);
m[5]= -sin((3.14/180)*B3);
m[6]=(cos((3.14/180)*A3)-1);
m[7]= -sin((3.14/180)*A3);
m[8]=sin((3.14/180)*B2);
m[9]= (cos((3.14/180)*B2)-1);
m[10]=sin((3.14/180)*A2);
m[11]= (cos((3.14/180)*A2)-1);
m[12]=sin((3.14/180)*B3);
m[13]= (cos((3.14/180)*B3)-1);
m[14]=sin((3.14/180)*A3);
m[15]= (cos((3.14/180)*A3)-1);
n[0]=p21 * cos((3.14/180)*C2);
n[1]=p31 * cos((3.14/180)*C3);
n[2]=p21 * sin((3.14/180)*C2);
n[3]=p31 * sin((3.14/180)*C3);
calculate();
sides();
wx=y[0]; wy=y[1]; zx=y[2]; zy=y[3];
w=sqrt( floor(wx)*floor(wx) + floor(wy)*floor(wy) ) ;
z=sqrt( floor(zx)*floor(zx) + floor(zy)*floor(zy) ) ;
theta = abs( acos(wx/w) * 57.32 ) ;
theta1 = abs( asin(wy/w) * 57.32 ) ;
fi = abs( acos(zx/z) * 57.32 ) ;
fi1 = abs( asin(zy/z) * 57.32 ) ;
w=abs(w); z=abs(z);
myline();
cout<<"wx: "<<wx<<" wy: "<<wy<<" zx: "<<zx<<" zy: "<<zy<<endl;
delay(150);
cout<<"w: "<<w<<" z: "<<z<<endl;
delay(150);
if(theta<theta1)
{theta=theta1;
}
cout<<(char)233<<": "<<theta<<"  ";
if(fi<90 && fi1<90)
{if(fi<fi1)
        {fi=fi1;
        }
else if(fi1<fi)
{fi=fi1;
}
}
cout<<(char)237<<": "<<fi;
delay(150);
myline();
getch();
clrscr();
//cleardevice();
login();
m[0]= (cos((3.14/180)*D2)-1);
m[1]= -sin((3.14/180)*D2);
m[4]= (cos((3.14/180)*D3)-1);
m[5]= -sin((3.14/180)*D3);
m[8]= sin((3.14/180)*D2);
m[9]= (cos((3.14/180)*D2)-1);
m[12]= sin((3.14/180)*D3);
m[13]= (cos((3.14/180)*D3)-1);
calculate();
sides();
ux=y[0]; uy=y[1]; sx=y[2]; sy=y[3];
u=sqrt( floor(ux)*floor(ux) + floor(uy)*floor(uy) ) ;
```

```
s=sqrt( floor(sx)*floor(sx) + floor(sy)*floor(sy) ) ;
sigma = abs( acos(ux/u) * 57.32 ) ;
sigma1 = abs( asin(uy/u) * 57.32 ) ;
si = abs( acos(sx/s) * 57.32 ) ;
si1 = abs( asin(sy/s) * 57.32 ) ;
u=abs(u);  s=abs(s);
delay(150);
myline();
cout<<"ux: "<<ux<<" uy: "<<uy<<" sx: "<<sx<<" sy:
"<<sy<<endl;
delay(150);
cout<<"u: "<<u<<" s: "<<s<<endl;
delay(150);
if(sigma<90 && sigma1<90)
{if(sigma<sigma1)
        {sigma=sigma1;
        }
else if( abs(sigma1-90) <= abs(sigma-90) )
{sigma=sigma1;
}
}
cout<<(char)229<<": "<<sigma<<"  ";
if(si<si1)
{si=si1;
}
cout<<"si: "<<si;
delay(150);
myline();
getch();
clrscr();
//----------------------Grphics-------------------------
cout<<"u: "<<u<<"  ";
cout<<"sigma: "<<sigma;
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\turboc3\\bgi");
setbkcolor(BLUE);
        int point_x[10],point_y[10];
        int x=400,y=400,x1=0,y1=0;
        int rotate,new_x=0,new_y=0;
        int count=0;
        rotate=sigma;
        point_x[count]=x;
        point_y[count]=y;
        start1:
        for(i=1;i<=rotate;i++)
        {setcolor(YELLOW);
                x1 = u * cos( (3.14/180) * i );
                y1 = u * sin( (3.14/180) * i );
                line(x,y,x+x1,y-y1);
                delay(30);
                setcolor(BLUE);
                line(x,y,x+x1,y-y1);
                if(i==rotate)
                {new_x=x+x1;
                        new_y=y-y1;
                }
        }
        setcolor(YELLOW);
        line(x,y,x+x1,y-y1);
        count++;
        point_x[count]=new_x;
        point_y[count]=new_y;
        if(count==1)
        {x=new_x;y=new_y;
                rotate=si;
                x1=0;y1=0;
                new_x=0;new_y=0;
                u=s;
                goto start1;
        }
        int
xx1=100,yy1=100,xx2=xx1+z,yy2=100,dy,dx;
        int new_xx1,new_xx2,new_yy1,new_yy2;
        if(count==2)
        {rotate=fi;
                for(i=1;i<=rotate;i++)
                {setcolor(YELLOW);
                        xx2 = u * cos( (3.14/180) * i
);
                        yy2 = u * sin( (3.14/180) * i );
                        line(xx1,yy1,xx1+xx2,yy1-
yy2);
                        delay(30);
                        setcolor(BLUE);
                        line(xx1,yy1,xx1+xx2,yy1-
yy2);
                }
                setcolor(YELLOW);
                xx2 = xx2 + xx1;
                yy2 = yy1 - yy2;
                dy =  new_y - yy2 ;
                dx = new_x - xx2 ;
                new_xx2 = xx2 + dx ;
                new_yy2 = yy2 + dy ;
                new_xx1 = xx1 + dx ;
                new_yy1 = yy1 + dy ;
                delay(30);
        line(new_xx1,new_yy1,new_xx2,new_yy2);
        }
        point_x[count]=new_xx1;
        point_y[count]=new_yy1;
        count++;
        xx1=100,yy1=100,xx2=xx1+w,yy2=100,dy=0,d
x=0;
        if(count==3)
        {rotate=theta;
                for(i=1;i<=rotate;i++)
                {setcolor(YELLOW);
```

ISSN: 2277-9655
Impact Factor: 1.852

```
                        xx2 = u * cos( (3.14/180) * i
);
                        yy2 = u * sin( (3.14/180) * i );
                        line(xx1,yy1,xx1+xx2,yy1-
yy2);
                        delay(30);
                        setcolor(BLUE);
                        line(xx1,yy1,xx1+xx2,yy1-
yy2);
                }
                setcolor(YELLOW);
                xx2 = xx2 + xx1;
                yy2 = yy1 - yy2;
                dy =  new_yy1 - yy2 ;
                dx = new_xx1 - xx2 ;
                new_xx2 = xx2 + dx ;
                new_yy2 = yy2 + dy ;
                new_xx1 = xx1 + dx ;
                new_yy1 = yy1 + dy ;
                delay(30);
        line(new_xx1,new_yy1,new_xx2,new_yy2);
        }
        point_x[count]=new_xx2;
        point_y[count]=new_yy2;
//--------------------First Shadow----------------------
        for(i=0;i<=4;i++)
        {cout<<"x"<<i<<":              "<<point_x[i]<<"
y"<<i<<": "<<point_y[i];
                cout<<endl;
        }
        x1=0;y1=0;
        for(i=1;i<=sigma+D2;i++)
        {setcolor(YELLOW);
                x1 = u * cos( (3.14/180) * i );
                y1 = u * sin( (3.14/180) * i );

        line(point_x[0],point_y[0],point_x[0]+x1,point_
y[0]-y1);
                delay(50);
                setcolor(BLUE);
        line(point_x[0],point_y[0],point_x[0]+x1,point_
y[0]-y1);
        }
        setcolor(YELLOW);
        line(point_x[0],point_y[0],point_x[0]+x1,point_
y[0]-y1);
        //line(x,y,point_x[0]+x1,point_y[0]-y1);
getch();
closegraph();
//-----------------------------------------------------
clrscr();
login();
        cout<<"\n\n\n\n\n";
        cout<<"Do you want to run the program
again:"<<endl;
        delay(150);
        cout<<endl<<"If Yes then [ Press : 1 ]"<<endl;
        delay(150);
        cout<<"If No then  [ Press : 2 ]";
        delay(150);
        myline();
        delay(150);
        cout<<"Enter Your OPTION and then press
ENTER: ";cin>>option;
        if(option==1)
        {goto start;
        }
        else
        {goto end;
        }
getch();
end:
}
```

Example: Synthesis of four bar mechanism using three position with respect to given input using above program.

Solution:

Obtained using the program is given below:

Take input value α2=17, α3=33, β2=20, β3=50, δ2=186, δ3=200 , *p*21=30 , *p*31=61, ɤ2=21, ɤ3=31

Output values: *w = 34* , *z = 69*, θ= 119, φ=71, u=41, s=137, σ = 39, Ψ =105

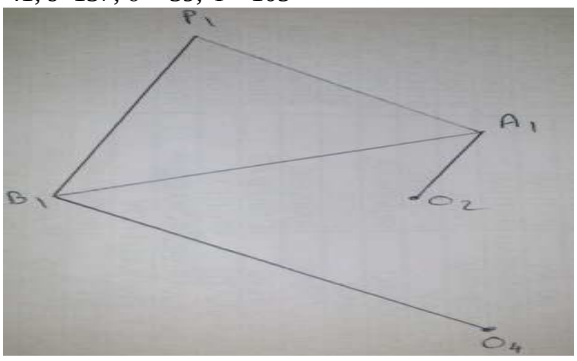

Fig. Four bar Mechanism

Advantages

Using this program we can minimize the time to synthesis a four bar mechanism. Also this program is more preferable than analytical or graphical way.

Disadvantages

This method is fail when determinate of matrix is zero because when the determinate is zero this program cannot find the value hence program does not show any output.

ISSN: 2277-9655
Impact Factor: 1.852

Conclusion

The proposed analytical method using computer programming is useful in determining the values of all link of four bar mechanism. On the basis of result and synthesis, it is concluded that this present method is very fast and less laborious and very efficient than graphical or analytical method. Also errors due to the analytical or graphical method are eliminated by this present method which gives better result.

References

[1] Shrinivas S. Bali, Satish Chand "Synthesis of a five-bar mechanism with variable topology for motion between extreme positions (SYNFBVTM)"

[2] R. Vinet ,A.A. Lakis , E.Haulin, "Optimal synthesis of a planar four – link mechanism used in a hand prosthesis" Huo Jingping, Cao Weiqing, Ruan Zhongtang, Synthesis of Path Generating four-bar linkage with use of fuzzy mathematics, Chinese Journal of Mechanical Engineering 25 (2) (1989) 61–66.

[3] T. Subbian, D.R. Flugrad, Four-Bar path generation synthesis by a continuation method, Journal of Mechanical Design 113 (3) (1991) 63–69.

[4] Chu Jinkui, Cao Weiquing, Synthesis of coupler curves of planar four-bar linkages through fast Fourier transform, Chinese Journal of Mechanical Engineering 29 (5) (1993) 117–122.

[5] Wang Zhixing, Chen Zhaobo, Jiang Lu, Study on planar linkage's path synthesis by coupler-angle curve, Chinese Journal of Mechanical Engineering 31 (1) (1995) 42–47.

[6] Wang Zhixing, Wu Qunbo, Li Jiansheng, A study on the path synthesis of four-bar linkage utilizing the method of rotation angle curve of links and on the evaluation of its multi-solutions, Machine Design 14 (11) (1997) 5–8.

[7] Wu Qunbo, Research on Intelligent-Virtual Theory and System of Mechanical Kinematics Scheme Doctor Dissertation, Harbin Institute of Technology, Harbin, 1999, pp. 71–78.

[8] Yu Hongying, Study on New Synthesis Methods of Planar Linkage and the Development of Software System Doctor Dissertation, Harbin Institute of Technology, Harbin, 2002, pp. 56–61.

[9] Z. Liu, J. Angeles, Optimization of planar, spherical and spatial function generators using input–output curve planning, Journal of Mechanical Design 116 (3) (1994) 915–919.

[10] J.A. Cabrera, A. Simon, M. Prado, Optimal synthesis of mechanisms with genetic algorithm, Mechanism and Machine Theory 37 (10) (2002) 1165–1177.

[11] A.-X. Liu, T.-L. Yang, Finding all solutions to unconstrained nonlinear optimization for approximate synthesis of planar linkages using continuation method, Journal of Mechanical Design 121 (3) (1999) 368–374.

[12] J.M. Jimenez, G. Alvarez, J. Cardenal, J. Cuadrado, A simple and general method for kinematic synthesis of spatial mechanism, Mechanism and Machine Theory 32 (3) (1997) 323–341.

[13] J. Mariappan, S. Krishnamurty, Using exact gradients in mechanism design, ASME Advances in Design Automation 44 (2) (1992) 53–59.

[14] R. Sancibrian, F. Viadero, P. Garcia, A. Fernandez, Gradient-based optimization of path synthesis problems in planar mechanisms, Mechanism and Machine Theory 39 (8) (2004)839–856.