SYNTHESIS OF 4-BAR FUNCTION GENERATOR WITH MINIMUM STRUCTURAL ERROR BY USING COMPUTATIONAL APPROACH

Prof Nitinchandra R. Patel¹, Prof Vinita B. Shah²

¹Mechanical Engineering department, G. H. Patel College of Engg and Technology, Vallabh Vidyanagar, Gujarat, India
vinitashah@gcet.ac.in
²Information Technology department, G. H. Patel College of Engg and Technology, Vallabh Vidyanagar, Gujarat, India
ntitinchangdrarpatel@gmail.com

Abstract — In the present work, an attempt has been made to design a four bar function generator, which will generate a given continuous function within a certain range with minimum structural error. While designing the four bar function generator, other critical parameters like transmission angle and a link ratio (ratio of maximum and minimum link lengths) has been taken as a set of constrains. This type of optimization problem with inequality constraint is difficult to solve by classical optimization method as well as by numerical optimization method. Also graphical methods are poor in accuracy and error is increased in making the scale of drawing to small or in determination of the point of interaction of two lines at a small angle to each other. Therefore there is still a need for the development of new techniques for solving these type of optimization problems. As they cannot ensure the global optimum always. In this investigation, one of the ways to solve this type of engineering problem is converting the range of function in to small incremental steps and implementing the whole process by applying computational method. The main advantage of this method is that the chance of its solutions for getting trapped in to local minimum is more. Moreover, a global minimum is found by applying a proper algorithm which is computationally efficient.

Keywords: Four bar function generator, Precision points, Chebishev’s Spacing, Frudenstein equation, Structural error, Cumulative error, Grubler’s criterion, Transmission angle, link ratio

I. INTRODUCTION

FREUDENSTEIN and Sendor used the precision point technique in which the on point’s structural error is assumed to be zero at some precision points. But the error is considered in between the precision point. This technique is simple but the accuracy of a curve generated by this approach is dependent on the number of precision point considered. The four point synthesis is a tedious geometrical approach and the five point synthesis uses matrix and determinant theory to solve the simultaneous linear equations. Later on, several investigators used optimization techniques to solve this problem. The least square technique is weakly convergent procedure and its final solution depends on the selection of initial parameter combinations. Whereas other investigators developed a nonlinear goal programming technique. The Graphical synthesis and H&N’s atlas of four bar coupler curves are useful references which contains number of coupler curves and defines the linkages geometry for each of its Grashoff linkages. Also a developed generalized overlay method enhances the graphical synthesis but the overlay technique is only utilized for two to five positions but for more than five positions, it becomes rather tedious and accuracy is generally sacrificed. All the above mentioned different approaches have limitations.

- Analytical methods are not able to convert every starting point towards the solution.
- Empirical method involves several trials.
- Graphical methods are poor in accuracy.

Therefore there is still a need for the developing a methodology in present technique for solving the problems. In the present work the extension of above topic has been developed based on two procedures.

(1) Synthesis of mechanism
(2) Implementing possible incremental steps within the span of working range for a four bar function generator.

The first is the transformation of the equations of synthesis (four bar linkages) into an appropriate form as precision point approach. Here the mathematical model is developed in order to obtain conveyance of the numerical iterations used in solving these equations. Secondly, the solutions are obtained by using this method through computer programming by getting iterations by considering the defined steps. In this attempt, it is the adequate to get suitable set of design variables for four bar function generator, which will give minimum structural error.

II. SYNTHESIS OF MECHANISM

- In four-bar linkage, the movement of output link creates a function y = f(x) with respect to movement of input link in the range (xs, xf)

@IJAERD-2015, All rights Reserved
Fig.1 Four-Bar Function Generator

- The mechanism fits generated function & desired function \( y = f(x) \) at a finite number of points in the interval \((x_s, x_f)\). Those points are called accuracy or precision points.
- The structural error \( E(x) \) is different between the generated function and desired function.
- To minimize \( E(x) \), Chebyshev's spacing of precision points is employed.
- Precision points are calculated by following formulas:

\[
x_j = \left( \frac{x_s + x_f}{2} \right) - \left( \frac{x_f - x_s}{2} \right) \cos \left( \frac{(2j - 1)\pi}{2n} \right) \quad j = \text{no. of Precision points: 1, 2, \ldots, n}
\]

\[
n = \text{Total no. of precision points (n = 3)}
\]

Fig.2 Four-Bar mechanism

- Figure 2 and 3 show linear relationships between angular changes and linear changes for both the input & output links.

For input link (\( \theta_1, \theta_2, \theta_3 \))

\[
\left( \begin{array}{c}
\nabla \phi_1 + \nabla \theta_1 \\
\nabla \phi_2 + \nabla \theta_2 \\
\nabla \phi_3 + \nabla \theta_3
\end{array} \right) = \left( \begin{array}{c}
\nabla x_j - x_s \\
\nabla x_j - x_s \\
\nabla x_j - x_s
\end{array} \right)
\]

Freudenstein's equation is given below:

\[
K_1 \cos \phi - K_2 \cos \theta + K_3 = \cos(\theta - \phi)
\]

\[
K_1 = \left( \frac{LA}{L2} \right) \quad K_2 = \left( \frac{LA}{L3} \right) \quad K_3 = \left( \frac{L1^2 - L2^2 + L3^2 + L4^2}{2L1L3} \right)
\]

For output link (\( \phi_1, \phi_2, \phi_3 \))

\[
\left( \begin{array}{c}
\nabla y_j + \nabla \phi_1 \\
\nabla y_j + \nabla \phi_2 \\
\nabla y_j + \nabla \phi_3
\end{array} \right) = \left( \begin{array}{c}
\nabla y_j - y_s \\
\nabla y_j - y_s \\
\nabla y_j - y_s
\end{array} \right)
\]

There are three sets of (\( \theta, \phi \)):

\[
\begin{align*}
\text{Cos (01 \& \phi1):} & \quad (02 \& \phi2) : (03 - \phi3) \\
K_1 \cos \phi_1 - K_2 \cos \theta_1 + K_3 &= \cos(\theta_1 - \phi_1) \\
K_2 \cos \phi_2 - K_2 \cos \theta_2 + K_3 &= \cos(\theta_2 - \phi_2) \\
K_3 \cos \phi_3 - K_2 \cos \theta_3 + K_3 &= \cos(\theta_3 - \phi_3)
\end{align*}
\]
K1, K2, K3 are unknown link ratios can be solved by Cramer’s rule. 

- Assume L4 (Fixed link) = 1

\[
L1 = \frac{1}{K1} \quad L3 = \frac{1}{K2} \quad L2 = \left( L1^2 + L3^2 - 2K3L1L3 + 1 \right)^{\frac{1}{2}}
\]

### III. FORMULATION FOR OBJECTIVE FUNCTION

The objective function is the cumulative error i.e. sum of structural error

\[
Es = \sum_{i=1}^{n} \left[ K1 \cos \phi_i + K2 \cos \theta_i + K3 - \cos(\phi_i - \phi) \right]
\]

Where \( Es \) is minimization parameter.

- To Minimize \( Es \) following properties are taken as the constraints
  (I) Transmission angle (\( \lambda \))
  (II) Link ratio

- Transmission angle is an index of the force to be transmitted from coupler to output link.

![Transmission angle (\( \lambda \))](image)

- It is calculated from the link lengths by using cosine relationships (from Fig 4).

\[
\cos \lambda = \frac{L2^2 + L3^2 - L4^2 - L1^2 + 2L1L4 \cos \theta}{2L1L3}
\]

It is required to have optimum transmission angle in the range of \( 0^\circ \leq \theta \leq 2\pi \) of input link rotation.

For \( \lambda = 0^\circ \) \( Ft = 0 \) It will cause Mechanism to lock or jam

For \( \lambda = 90^\circ \) \( Fr = 0 \) It will max torque to transmit force form link L2 to L3

- For a safe design, \( \lambda \) is taken as more than 25°.
  i.e. \( \lambda > 25^\circ \)
- Link ratio is the ratio of the largest link to the smallest link.
- For a good design lr is taken less than 12
  i.e. \( lr \leq 12 \)
  - Mathematically, the formulation will be Minimize

\[
Es = \sum_{i=1}^{n} \left[ K1 \cos \phi_i + K2 \cos \theta_i + K3 - \cos(\phi_i - \phi) \right]
\]

Subject to

(i) \( \lambda_{\min} \geq 25^\circ \) (ii) \( lr \leq 12 \)

### IV. PROGRAMMING OF SYNTHESIS

// FOUR BAR FUNCTION GENERATOR -- SYNTHESIS //
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<dos.h>
#define c M_PI/180.0
void main()
{   clrscr();
    FILE *fp;
    fp=fopen("RUTSYS1.doc","w");

©IJAERD-2014, All rights Reserved 1388
int i,j,k;
float xs=1.0,xf=3.0,dth=120.0,dph=60.0;
float y,s,v,p,m,ths,phs,r,d1,d2,d3,d4,d5,d6,k1,k2,k3,colam,lr,lamda,sum,Plr,Plamda,temp1,temp2,temp3,temp4,temp5,temp6,temp7,temp8,temp;
float th[4],ph[4],x[4],y[4],l[5],Theta[1000],Phi[1000],Lr[1000],Lamda[1000],L[1000],L1[1000],L2[1000],L3[1000],L4[1000];
double colamda[1000],Es[1000];
printf("Enter following values:---------
");
printf(fp,"Enter following values:---------
");
printf("xs xf dth dph
");
fprintf(fp,"xs xf dth dph
");
k=1;
for(ths=0.0;ths<=dth;ths=ths+2.0)
for(phs=0.0;phs<=dph;phs=phs+2.0)
{
p=(xf+xs);
m=(xf-xs);
ys=sqrt(xs);
yf=sqrt(xf);
v=(yf-ys);
/* CHEBYSHEV'S SPECING OF THREE ACCURACY POINTS */
for(j=1;j<=3;j++)
{th[j]=(2.0*(float)j-1.0)/6.0;
r=(m/2.0)*(cos(M_PI*th[j]));
x[j]=(p/2.0)-r;
y[j]=sqrt(x[j]);
ths+((dth/m)*(x[j]-xs));
phs+((dph/v)*(y[j]-ys));
}
for(j=1;j<=3;j++)
{th[j]=th[j]*c;
ph[j]=ph[j]*c;
}
d1=cos(th[1]-ph[1])-cos(th[2]-ph[2]);
d2=cos(th[1]-ph[1])-cos(th[3]-ph[3]);
d3=cos(ph[1])-cos(ph[2]);
d4=cos(th[1])-cos(th[2]);
d5=cos(ph[1])-cos(ph[3]);
d6=cos(th[1])-cos(th[3]);
k1=((d1*d6)-(d2*d4))/((d3*d6)-(d5*d4));
k2=((d1*d5)-(d2*d3))/((d6*d3)-(d4*d5));
k3=((cos(th[1]-ph[1]))-(k1*cos(ph[1]))+(k2*cos(th[1])));
l[4]=1.0000000;
l[1]=l[4]/k1;
l[3]=l[4]/k2;
/* TRANSMISSION ANGLE-- LAMDA */
colam=(double)((pow(double)[2],2)+pow((double)[3],2)-pow((double)[1],2)-pow((double)[4],2)+
(2.0*l[4]*l[1]*cos(th*s))/((2.0*l[2]*l[3]));
if(colam>=1.0)
colam=colam;
else
break;
if(colam<1.0)
colamda[k]=colam;
else
break;
lambda=(double)acos((double)colamda[k]);
lambda=(float)lambda*(180.0/M_PI);
L1[k]=l[1];
L2[k]=l[2];
L3[k]=l[3];
L4[k]=l[4];

/* LINK RATIO -- LR */
for(i=1;i<=3;i++)
  {
    for(j=1;j<=4-i;j++)
      {
        if(fabs(l[j])<fabs(l[j+1]))
          {
            temp=l[j];
            l[j]=l[j+1];
            l[j+1]=temp;
          }
        else
          continue;
      }
}

/* GRUBLER'S CRITERION */
  break;
else
  lr=(fabs(l[1])/fabs(l[4]));

/* CUMULATIVE ERROR */
sum=0.0;
for(i=1;i<=3;i++)
  {
    sum=sum+(k1*cos(ph[j])-(k2*cos(th[j]))+k3-(cos(th[j]-ph[j]));
  }
Plr=(float)pow(((fabs(lr)/12.0)-1.0),2.0);
Plamda=(float)pow(((25.0/(float)lambda)-1.0),2.0);
if((lr<=12.0)&&(&&(lambda>=25.0))
  {
    Es[k]=sum;
    Theta[k]=ths;
    Phi[k]=phs;
    Lamda[k]=lambda;
    Lr[k]=lr;
    k=k+1;
  }
else if((lr>12.0)&&(&&(lambda>=25.0))
    sum=sum+Plr;
else if((lr<=12.0)&&(&&(lambda<25.0))
    sum=sum+Plamda;
else
    sum=sum+Plr+Plamda;
}

printf("n CUMULATIVE ERROR          THETA     PHI");
fprintf(fp,"n CUMULATIVE ERROR         THETA     PHI");
for(i=1;i<=k-1;i++)
  {
    printf("n Es[%3d]=%12.9lf     %.2f     %.2f",  i,Es[i],Theta[i],Phi[i]);
    fprintf(fp,"n Es[%3d]=%12.9lf     %.2f     %.2f",  i,Es[i],Theta[i],Phi[i]);
  }

printf("n");
delay(50);
//getch();
}
for(i=1;i<=k-1;i++)
{
    for(j=1;j<=(k-i);j++)
    {
        if(Es[j]<Es[j+1])
        {
            temp=Es[j];
            temp1=Theta[j];
            temp2=Phi[j];
            temp3=Lamda[j];
            temp4=Lr[j];
            temp5=L1[j];
            temp6=L2[j];
            temp7=L3[j];
            temp8=L4[j];
            Es[j]=Es[j+1];
            Theta[j]=Theta[j+1];
            Phi[j]=Phi[j+1];
            Lamda[j]=Lamda[j+1];
            Lr[j]=Lr[j+1];
            L1[j]=L1[j+1];
            L2[j]=L2[j+1];
            L3[j]=L3[j+1];
            L4[j]=L4[j+1];
        }
        else
            continue;
    }
}

/* DESIGN VA LUES */
printf("\n\n DESIGN VA LUES OF MECHANISM: ");
printf("\n-----------------------------");
printf("Es(min)=%.9lf\n THETA=%.7f\n PHI=%.7f\n LAMDA=%.7f\n LR=%.7f", Es[i],Theta[i],Phi[i],Lamda[i],Lr[i]);
printf("\n-----------------------------");
printf("\n Es(min)=%.9lf\n THETA=%.7f\n PHI=%.7f\n LAMDA=%.7f\n LR=%.7f", Es[i],Theta[i],Phi[i],Lamda[i],Lr[i]);
printf("\n L1=%.7f\n L2=%.7f\n L3=%.7f\n L4=%.7f", L1[i],L2[i],L3[i],L4[i]);
if((Lr[i]<=12.0)&&(Lamda[i]>=25.0))
{
    printf("\n FOUR BAR MECHANISM SATISFIES ALL THE CONSTAINTS.");
}
getch();
V. RESULT AND DISCUSSION

The computational method has been used to solve various functions of four bar mechanism on TURBO-C. The execution time for the problem is found to be few seconds per run. Here, two different functions are taken for four bar function generator. The input and output angular values are taken as per its working range.

*Table 1: input values applied for different functions*

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>( y = \sqrt{x} )</th>
<th>( y = \log_{10}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_s )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( X_f )</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>( \Delta \theta )</td>
<td>120</td>
<td>90</td>
</tr>
<tr>
<td>( \Delta \phi )</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 2: calculated output design values (local minimum) for different functions

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>( y = \sqrt{x} )</th>
<th>( y = \log_{10}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_s(\text{min}) )</td>
<td>-0.000001005</td>
<td>-0.000001120</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>32.0000000</td>
<td>68.0000000</td>
</tr>
<tr>
<td>( \phi )</td>
<td>28.0000000</td>
<td>54.0000000</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>25.8621330</td>
<td>53.4904213</td>
</tr>
<tr>
<td>( L_r )</td>
<td>9.8806238</td>
<td>9.6502247</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>0.1146753</td>
<td>0.1083761</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>1.1330636</td>
<td>1.0458534</td>
</tr>
<tr>
<td>( L_3 )</td>
<td>0.2617133</td>
<td>0.1490367</td>
</tr>
<tr>
<td>( L_4 )</td>
<td>1.0000000</td>
<td>1.0000000</td>
</tr>
</tbody>
</table>

The iterations and related structural errors are shown in a graph from which a minimum structural error is determined.

*Fig 5: Function: \( y = \sqrt{x} \), \( E_s \) v/s no of iterations*

The above graph of \( y = \sqrt{x} \) has 489 iterations, which give possible solutions. One of them gives minimum error.
i.e. \( E_s = -0.000001005 \) at \((32,28)\)

![Graph](image)

**Fig 6: Function:** \( y = \log_{10}(x) \), \( E_s \) v/s no of iterations

The above graph of \( y = \log_{10}(x) \) has 276 iterations, which give possible solutions. One of them gives minimum error i.e. \( E_s = -0.000001120 \) at \((68, 54)\)

**CONCLUSION**

It is better to find the solution of engineering optimization problem; initially local search is done by conventional method. Because of simulation with computer programming, iterative procedure becomes easy and having less complicity. It is to be noted that in this method the results of cumulative error are after satisfying both the constraints regarding minimum transmission angle and maximum link ratio. The results obtained are very much attractive and precise but not a very accurate design.

This is because of limitations in accurately generating highly nonlinear functions by three point precession method. However, if the constraints are relaxed, a more accurate design matching the designed function can be obtained. The result, as obtained indicate the much better solution for design of four-bar function generator.

**ACKNOWLEDGMENT**

I would like to express my thanks and gratitude to PROF. VINITA SHAH (Department of Information Technology) of G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar for devoting much of her precious & valuable time and kind cooperation in the said work.

**REFERENCES**

AUTHORS’ BIOGRAPHY

Prof. Nitinchandra R. Patel is an Assistant Professor in Mechanical Engineering Department of G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India. He completed Master degree in Machine Design in 2004 from Sardar Patel University, Vallabh Vidyanagar and Bachelor degree in Mechanical Engineering in 1997 from B.V.M. Engineering College, Sardar Patel University. He has 5 years industrial experience (2 years in Manufacturing & 3 years in Chemical Industries) and 13 years and 8 months teaching experience. He has presented 2 technical research papers in International conferences and published 11 technical research papers in International journals. He reviewed a book published by Tata McGraw Hill in 2012. He is a Member of Institute of Engineers (I) and Life member of ISTE. He is a reviewer in Editorial board of Peer-reviewed International Journal of Advance Engineering and Research Development. He is also recognized as a Chartered Engineer by Institute of Engineers (I) in Mechanical Engineering Division in 2012.

Prof. Vinita Bhavanshkumar Shah is an Assistant Professor in the Information Technology Department of G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India. She did Master degree in Information Technology in 2013 from G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat Technological University and Bachelor degree in Information Technology in 2010 from S.V.I.T, Vasad. She has total 4 years and 9 months teaching experience. She presented 2 technical research papers in National conferences, 1 in International conference and published 2 technical research papers in International journals.