UNIT:	**DYNAMICS OF PLANAR MECHANISMS**	
COURSE:	**MECHATRONICS**	
TITLE:	**LAB WORK 1**	
MEMBERS:	WASHINGTON KIGANI KAMADI	ENM221-0063/2017
	FELIX WANYOIKE GATERU	ENM221-0056/2017
	MICHAEL GITAU KIMANI	ENM221-0066/2017
	BROWN IRIA KIMUNYI	ENM221-0067/2017
DATE:	**November, 1st 2020**	

OBJECTIVES

1. To implement the Chebyshev spacing formula using MATLAB
2. To implement the Freudenstein's equation using MATLAB
3. To use Freudenstein's equation to design a four bar linkage with given design parameters.
4. To use the Least Square method to solve for the link lengths of the four bar linkage.
5. To compute the structural errors occurring in a given range of input angles.
6. To compute the transmission angle of the mechanism in a particular range of input angles.

THEORY

The synthesis of mechanism is the design or creation of a mechanism to produce a desired output motion for a given input motion. It is the determination of proportions of a mechanism for the given input and output motion. In designing a mechanism to generate a particular function, it is usually impossible to accurately produce the function at more than a few points. The points at which the generated and desired functions agree are known as precision points or accuracy points and must be located so as to minimize the error generated between these points. The best spacing of the precision points, for the first trial, is called Chebyshev's spacing. Denoted by the formula:

$$\boldsymbol{x_j = \frac{1}{2}(x_s + x_f) - \frac{1}{2}(x_f - x_s)\cos\left[\frac{\pi(2j-1)}{2n}\right]}$$

$\mathbf{x_j}$ = Precision points

The subscripts S and F indicate start and finish positions respectively.

Freudenstein's equation relates the input and output angles to the ratios of the lengths of the links denoted by K1, K2 and K3. It is expressed as:

$$\boldsymbol{cos(\theta - \phi) \ = \ k_1 cos\phi \ - \ k_2 cos\theta \ + \ k_3}$$

It is possible to design a mechanism to give least deviation from the specified positions. This is done by using least square technique denoted by:

$$\boldsymbol{k_1 \sum_{i=1}^{n} cos\phi_i \ + \ k_2 \sum_{i=1}^{n} cos\theta_i \ + \ k_3 \sum_{i=1}^{n} 1 \ = \ k_1 \sum_{i=1}^{n} cos(\theta_i - \phi_i)}$$

PROCEDURE

Our MATLAB program allowed for the user to input the length of the fixed link, and a range of input angles for a four-bar linkage. Using the range of input angles provided and the output angles calculated, Chebyshev's spacing was applied in order to find three precision points.

Each of these precision points was then fed into the Freudenstein's equation and solving these equations linearly allowed us to find the three ratios K1, K2 and K3. The lengths of the crank, coupler and follower were then subsequently calculated with the help of the three ratios calculated above.

The transmission angle was then calculated using the entered range of input angles in steps of 5 degrees. This was done by applying the cosine rule twice with the input angle being considered in the first instance and the transmission angle being considered the second time round. Since both of these angles share an opposite length, rearranging the equation allowed for the calculation of the transmission angles.

The variation of the input angles with the input angles was then plotted using the subplot command in MATLAB.

Structural errors were then calculated by rearranging Freudenstein's equation for the given range of input angles. The variation of structural errors as a function of input angles was then plotted.

Using Chebyshev’s spacing five precision points were calculated. We then used these precision points to re-evaluate the length ratios K1, K2 and K3. Using these new ratios, the lengths of the crank, coupler and follower were then calculated. Structural errors were also re-calculated using these newly calculated ratios for the given range of input angles at an increment of 5 degrees.

These structural errors were then plotted as a function of input angles on the same axis as the structural errors in the previous scenario.

DATA PRESENTATION AND DISCUSSION

From the computer program's implementation of the Chebyshev spacing formula and the Freudenstein's equation, using three precision points, the values of K1, K2 and K3 were as follows;

- K1 = **-7.1003**
- K2 = **-3.4090**
- K3 = **-0.7100**

From the length ratios K1, K2 and K3, the lengths of the links were calculated as follows (With the length of the fixed link given as 400mm);

- Coupler ⇒ **442.44926mm**
- Follower ⇒ **120.26793mm**
- Crank ⇒ **57.74377mm**

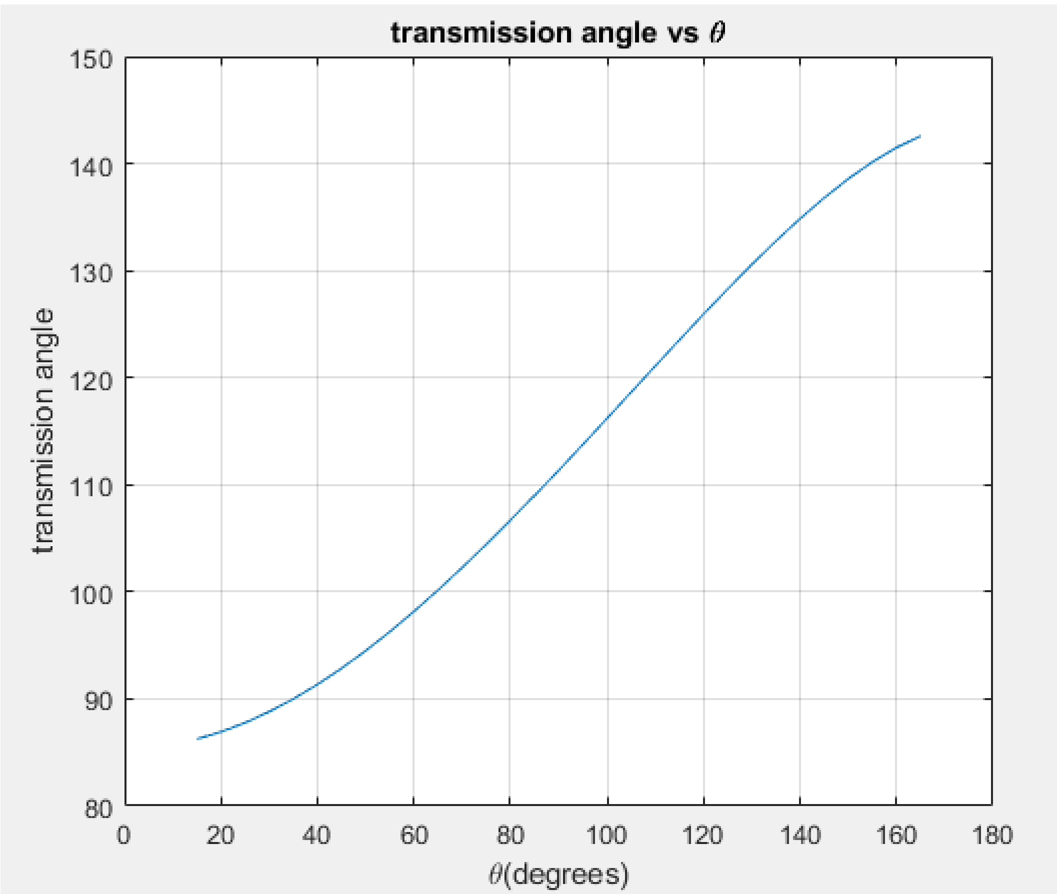
For a range of input angles ranging from 5° to 165° increasing by 5°, the transmission angles were computed using the formula below;

$$\cos \mu = \frac{(b^2 + c^2) - (a^2 + d^2) + 2ad \cos\phi}{2bc}$$

TABLE OF ANGULAR INPUT AND TRANSMISSION ANGLES

input	transmission
15.0000	86.2583
20.0000	86.9282
25.0000	87.7802
30.0000	88.8075
35.0000	90.0025
40.0000	91.3564
45.0000	92.8599
50.0000	94.5029
55.0000	96.2752
60.0000	98.1658
65.0000	100.1640
70.0000	102.2587
75.0000	104.4386
80.0000	106.6926
85.0000	109.0091
90.0000	111.3765
95.0000	113.7829
100.0000	116.2158
105.0000	118.6624
110.0000	121.1088
115.0000	123.5403
120.0000	125.9410
125.0000	128.2933
130.0000	130.5778
135.0000	132.7731
140.0000	134.8553
145.0000	136.7982
150.0000	138.5732
155.0000	140.1499
160.0000	141.4972
165.0000	142.5844

The values of transmission angles were plotted against the corresponding input angles and the graph is shown below.



The transmission angle is of good quality since it lies between the range **40° < ϑ < 140°**

Using the Chebyshev spacing equation to evaluate the values of five precision points, the values of K1, K2 and K3 were solved using the **Least Squares method** and their values found as below;

- K1 = **-1.7293**
- K2 = **-0.70609**
- K3 = **0.4632**

From the values of K1, K2 and K3 obtained from the Least Squares method, the lengths of the links were found as follows (With the length of the fixed link given as 400mm);

- Coupler ⇒ **658.743mm**
- Follower ⇒ **580.662mm**
- Crank ⇒ **237.090mm**

From the values of the link ratios obtained by using the two methods i.e. Freudenstein's equation and the Least Squares method, the values of structural errors were calculated by use of the formula below, while incrementing the input angle by 5 degrees in the range, **$15° < \vartheta < 165°$;**

$$\boldsymbol{e_i = k_1 cos\psi - k_2 cos\varphi + k_3 - cos(\varphi_i - \psi_i)}$$

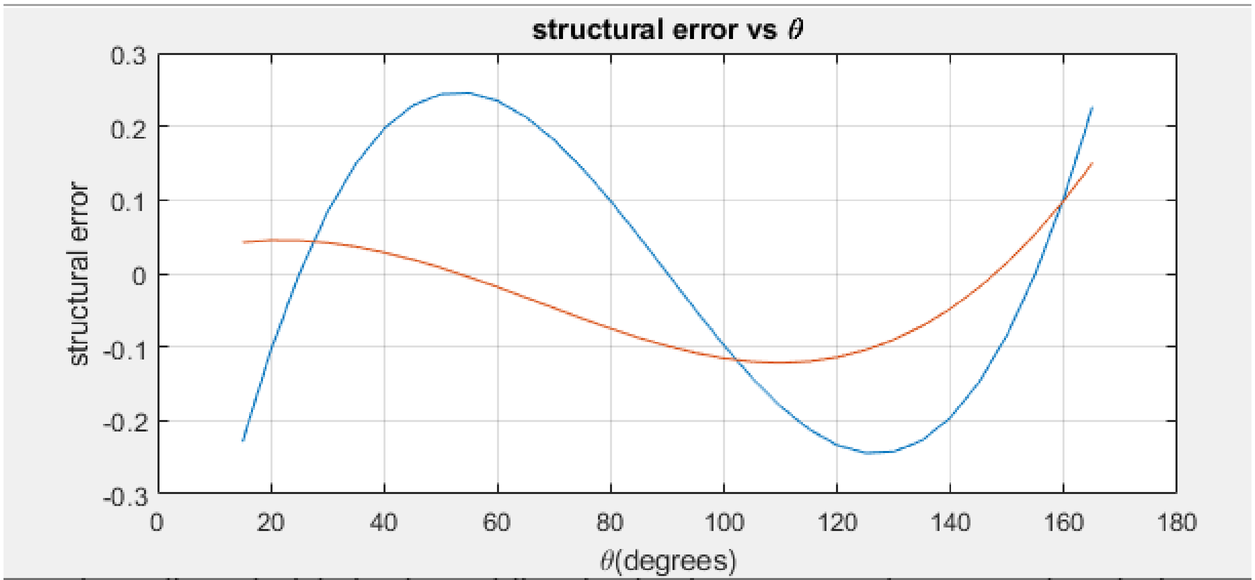
TABLE OF INPUT ANGLES AND STRUCTURAL ERRORS FOR PART A

input	error
15.0000	-0.2287
20.0000	-0.1048
25.0000	-0.0009
30.0000	0.0835
35.0000	0.1491
40.0000	0.1970
45.0000	0.2282
50.0000	0.2439
55.0000	0.2456
60.0000	0.2346
65.0000	0.2127
70.0000	0.1815
75.0000	0.1428
80.0000	0.0983
85.0000	0.0501
90.0000	-0.0000
95.0000	-0.0501
100.0000	-0.0982
105.0000	-0.1425
110.0000	-0.1810
115.0000	-0.2119
120.0000	-0.2336
125.0000	-0.2443
130.0000	-0.2425
135.0000	-0.2267
140.0000	-0.1955
145.0000	-0.1479
150.0000	-0.0827
155.0000	0.0009
160.0000	0.1036
165.0000	0.2261

TABLE OF INPUT ANGLES AND STRUCTURAL ERRORS FOR PART B

input	error
15.0000	0.0424
20.0000	0.0450
25.0000	0.0448
30.0000	0.0418
35.0000	0.0363
40.0000	0.0286
45.0000	0.0189
50.0000	0.0076
55.0000	-0.0050
60.0000	-0.0186
65.0000	-0.0327
70.0000	-0.0471
75.0000	-0.0614
80.0000	-0.0750
85.0000	-0.0876
90.0000	-0.0988
95.0000	-0.1082
100.0000	-0.1153
105.0000	-0.1198
110.0000	-0.1213
115.0000	-0.1194
120.0000	-0.1138
125.0000	-0.1041
130.0000	-0.0900
135.0000	-0.0713
140.0000	-0.0477
145.0000	-0.0190
150.0000	0.0150
155.0000	0.0544
160.0000	0.0994
165.0000	0.1499

The values obtained for the structural errors were plotted against the corresponding input angle for both a and b. The graph is included below;



From the calculated values of the structural error, a greater error is found when using Freudenstein's equation with three precision points as compared to the structural error found while using the **Least Square method**.

REFERENCES

1. Lecture notes by Mrs. Leila Mbagaya
2. J.K. Gupta & R.S. Khurmi. (1976). Theory of Machines. Eurasia Publishing House

THE CODE

```
clc;
%part a (Freudenstein's)
fixed = input('Input the length of the fixed link: ');
theta_2 = [];
theta_4 = [];
theta_not = input('Input the starting range of theta 2: ');
theta_f = input('Input the final value of the range of theta 2: ');
n = input('Input the number of precision points: ');
j = 1;
while (j < 4)
    precision_2 = 0.5*(theta_not+theta_f)- 0.5*(theta_f-
theta_not)*cosd(180*(2*(j)-1)/(2*n));
    precision_4 = (precision_2)*0.43 +65;
    theta_2 = [theta_2, precision_2];
    theta_4 = [theta_4, precision_4 ];
    j = j+1;
end

A = [cosd(theta_4(1,1)), -cosd(theta_2(1,1)), 1 ; cosd(theta_4(1,2)), -
cosd(theta_2(1,2)), 1 ; cosd(theta_4(1,3)), -cosd(theta_2(1,3)), 1];
B = [cosd(theta_2(1,1) - theta_4(1,1)) ; cosd(theta_2(1,2) - theta_4(1,2)) ;
cosd(theta_2(1,3) - theta_4(1,3)) ];
sol = linsolve(A,B);

K1 = sol(1,1);
K2 = sol(2,1);
K3 = sol(3,1);

fprintf('K1 = %.15g\n',K1);
fprintf('K2 = %.15g\n',K2);
fprintf('K3 = %.15g\n',K3);

crank_length = fixed /K1;
follower_length = fixed / K2;
coupler_length =
sqrt((crank_length*crank_length)+(follower_length*follower_length)+(fixed*fix
ed)-(K3*2*crank_length*follower_length));

fprintf('crank_length = %.15g\n',crank_length);
fprintf('follower_length = %.15g\n',follower_length);
fprintf('coupler_length = %.15g\n',coupler_length);

%transmssion_angle;
theta = 15 :5 :165;
transmission_angle = acosd(((coupler_length^2)+(follower_length^2)-
(crank_length^2)-
(fixed^2)+(2*crank_length*fixed*cosd(theta)))/(2*coupler_length*follower_leng
th));
disp(' ')
disp('TABLE OF ANGULAR INPUT AND TRANSMISSION ANGLES')
disp('   input    transmission    ')
disp([theta',transmission_angle'])
```

```
%subplot(1,1,1),plot(theta,transmission_angle),xlabel('\theta(degrees)'),ylab
el('transmission angle'),title('transmission angle vs \theta'),grid on
output = 65 + 0.43*theta;


%structural error part a)
structural_error = K1*cosd(output)- K2*cosd(theta)+K3-cosd(theta - output);
disp(structural_error)
disp('TABLE OF INPUT ANGLES AND STRUCTURAL ERRORS')
disp('   input    error   ')
disp([theta',structural_error'])
subplot(1,1,1),plot(theta,structural_error),xlabel('\theta(degrees)'),ylabel(
'structural error'),title('structural error vs \theta'),grid on

%part b (Least Square)
disp('LEAST   SQUARE')
n5 = input('Input the number of precision points: ');
theta_22 = [];
theta_42 = [];
p=1;
while (p < 6)
    precision_22 = 0.5*(theta_not+theta_f)- 0.5*(theta_f-
theta_not)*cosd(180*(2*(p)-1)/(2*n5));
    precision_42 = (precision_22)*0.43 +65;
    theta_22 = [theta_22, precision_22];
    theta_42 = [theta_42, precision_42 ];
    p = p+1;
end
ci = cosd(theta_22);
sumci = sum(cosd(theta_22));
co = cosd(theta_42);
sumco = sum(cosd(theta_42));
sumci2 = sum(cosd(theta_22).^2);
sumco2 = sum(cosd(theta_42).^2);
diff = cosd(theta_22-theta_42);
sumd = sum(diff);
e1 =     [
        sumco2,  - sum(co.*ci), sum(co);
        sum(co.*ci), -sumci2, sumci;
        sumco,-sumci,5];
e2 = [sum(co.*diff); sum(ci.*diff); sumd];
als = linsolve(e1,e2);

K1 = als(1);
K2 = als(2);
K3 = als(3);

fprintf('K1 = %.15g\n',K1);
fprintf('K2 = %.15g\n',K2);
fprintf('K3 = %.15g\n',K3);

crank_length = fixed /K1;
follower_length = fixed / K2;
coupler_length =
sqrt((crank_length*crank_length)+(follower_length*follower_length)+(fixed*fix
ed)-(K3*2*crank_length*follower_length));
```

```
fprintf('crank_length = %.15g\n',crank_length);
fprintf('follower_length = %.15g\n',follower_length);
fprintf('coupler_length = %.15g\n',coupler_length);

%structural error part b)
structural_error = K1*cosd(output)- K2*cosd(theta)+K3-cosd(theta - output);
disp('TABLE OF INPUT ANGLES AND STRUCTURAL ERRORS')
disp('   input    error   ')
disp([theta',structural_error'])
hold all
subplot(1,1,1),plot(theta,structural_error),xlabel('\theta(degrees)'),ylabel(
'structural error'),title('structural error vs \theta'),grid on
```