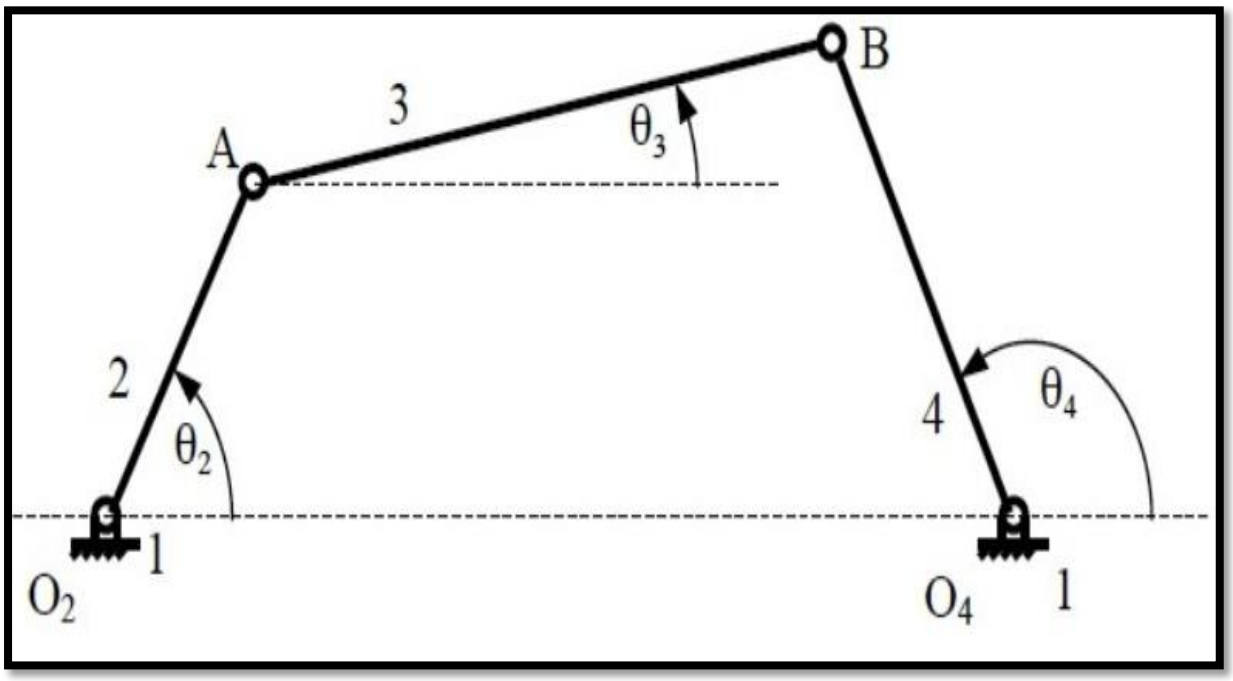
ME352A: Theory of Mechanisms and Machines
Bonus Assignment

Topic: Implementation of four-bar (4-R) linkage in MATLAB (Analysis of the configurations through animation)

- **Figure**:



Where –

- O_2O_4 = Frame = L_1
- O_2A = Input Link = L_2
- AB = Coupler = L_3
- O_4B = Follower = L_4

- T

4bar_l

4bar_

```
clc;
clear all;
count = 1;
while (count > 0)
    %% Input the dimensions for 4-bar linkage
    fprintf('\nInput the dimensions for grashof linkage\n');
    L1 = input('Frame (L1) = ');
    L2 = input('Input link (L2) = ');
    L3 = input('Coupler (L3) = ');
    L4 = input('Follower (L4) = ');
    %% Checking the grashof condition %
    ML= [L1 L2 L3 L4];        % input values into a matrix
    MS= sort(ML);             % sort the values in ascending order
    L = MS(1,4);
    S = MS(1,1);
    P = MS(1,2);
    Q = MS(1,3);
    %% checking the grashof condition
    if(L + S < P + Q)
        fprintf('\nGreat ! This is grashof linkage\n');
            if  L1 == S     % if L1 is shortest
                fprintf('This is a Double-Crank mechanism\n');
                count = 0;
            elseif L2 == S  % if L2 is shortest
                fprintf('This is a Crank-Rocker mechanism\n');
                count = 0;
            else
                fprintf('Please input grashof linkage\n');
                count = 1;
            end
    elseif(L + S == P + Q)
        fprintf('\nThis is a parallelogram type linkage\n');
        fprintf('Please input grashof linkage\n');
        count = 1;
    elseif(L + S > P + Q)
        fprintf('\nThis is a non-grashof linkage\n');
        fprintf('Please input grashof linkage\n');
        count = 1;
    end
```

```
C=input('\nAnglular speed ω = Cπ, enter C : ');
fprintf('\n');
t = 0:5/360:10;
omega = C*pi;
theta_1 = deg2rad(0);
theta_2 = omega*t;
end
for i=1:length(theta_2)
    theta_2_dash(i) = theta_2(i) - theta_1;
    lambda(i) = sqrt(L1^2 + L2^2 -2*L1*L2*cos(theta_2_dash(i)));
    beta(i) = acos( (L1^2 + lambda(i)^2 - L2^2) / (2*L1*lambda(i)));
    phi(i) = acos( (L3^2 + lambda(i)^2 - L4^2) / (2*L3*lambda(i)));
    delta(i) = acos( (L4^2 + lambda(i)^2 - L3^2) / (2*L4*lambda(i)));
    if(theta_2_dash<=pi)
        theta_3(i) = phi(i)-(beta(i)-theta_1);
        theta_4(i) = pi-delta(i)-(beta(i)-theta_1);
        gamma(i)= acos( (L3^2+L4^2-lambda(i)^2) / (2*L3*L4)) - pi/2;
    else
        theta_3(i) = phi(i)+(beta(i)+theta_1);
        theta_4(i) = pi-delta(i)+(beta(i)+theta_1);
        gamma(i)= acos( (L3^2+L4^2-lambda(i)^2) / (2*L3*L4)) - pi/2;
    end
F(i) = getframe(gcf);
A_x(i) = L2*cos(theta_2(i));
A_y(i) = L2*sin(theta_2(i));
B_x(i) = L2*cos(theta_2(i))+L3*cos(theta_3(i));
B_y(i) = L2*sin(theta_2(i))+L3*sin(theta_3(i));
B_o_x(i) = L1*cos(theta_1);
B_o_y(i) = L1*sin(theta_1);
plot([0 A_x(i)], [0 A_y(i)],'go-','LineWidth',5);hold on;                %L2
plot([A_x(i) B_x(i)], [A_y(i) B_y(i)], 'ro-','LineWidth',5); hold on;     %L3
plot([B_x(i) B_o_x(i)], [B_y(i) B_o_y(i)], 'bo-','LineWidth',5); hold on; %L4
plot([B_o_x(i) 0], [B_o_y(i) 0], 'ko-','LineWidth',5);hold off;           %L1
grid on
axis([-10 15 -10 10]);
pause(0.001);
end

video = VideoWriter('4-bar.avi', 'Uncompressed AVI');
open(video)
writeVideo(video, F);
close(video)
```

- **Example:** The following test case was run in the code as an example –

```
Input the dimensions for grashof linkage
Frame (L1) = 4
Input link (L2) = 6
Coupler (L3) = 7
Follower (L4) = 8

Great ! This is grashof linkage
This is a Double-Crank mechanism

Anglular speed ω = Cπ, enter C : 1/5

fx >>
```

- **Output:** The output animation generated has been attached with this email in video format. Here is a snapshot –