ME7732 – Mechatronic Design and Automation

Topic: Mechanisms Design, Simulation and Analysis

Name: NILENDU SAHA

Stream: Mechatronic Systems

KU Number: K1943079

Date of Submission: 29/05/2020

Abstract:

The assignment is all about performing kinematics and dynamics analysis on proper well-known mechanisms. Such analysis is really required in order to get and idea about the robustness of any mechanism. This assignment contained problems based on only two bar and four bar mechanisms. Analysis is performed in two types of methods. One was done in Matlab, where theoretical equations were put in as code structures. This type of analysis is more of a theoretical kind. Another was a simulation-based analysis and it was performed in Siemens NX software. The parts of the mechanism are separately created and assembled more in a practical way. This process is much more realistic than the other process. At the end of each methods certain graph plots are obtained. These graph plots help to determine the robustness of the system for a particular task. The proceedings of the methods are mentioned in detail in the Methodology section of the report. The resultant graph plots are discussed in the Result and Discussion section of the report.

1. METHODOLOGY: -

This particular assignment consists of two methods of analysis. As mentioned in the abstract the first method is to perform the analysis on Matlab and the second method is supposed to be done on Siemens NX PLM Software. Two kinds of mechanism were set as a part of the task. The two kind of mechanism and their task are given below.

1.1. TWO BAR ROBOTIC ARM: -

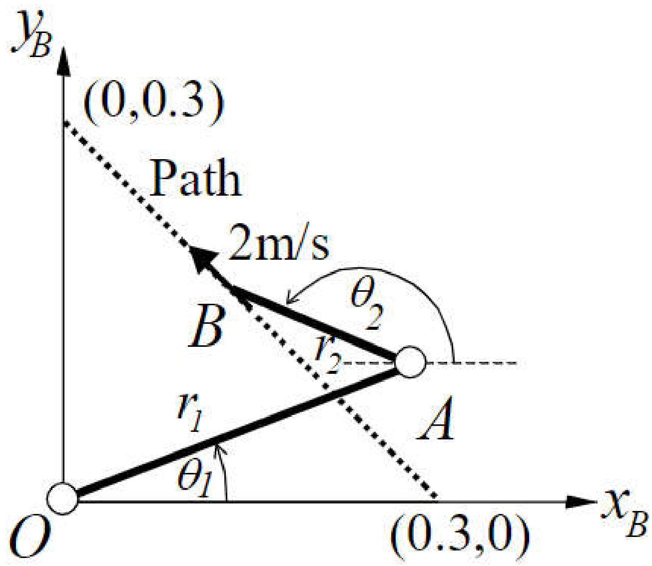

Figure 1.1

The above given figure 1.1 consists a schematic diagram of the two-bar robotic arm. Particularly the diagram depicts a two-link planar manipulator. It has two links OA and AB and an end effector at point B. The job of the manipulator is to weld a structure, as its end effector has the welding electrode. The angles θ_1 and θ_2 are controlled in such a way that the end effector moves in a straight path from point (0.3,0) to (0,0.3) so that it can perform the welding operation. The ultimate task is to perform kinematics analysis and plot the variation of angles θ_1 and θ_2. Also, to plot the angular velocities of the two arms OA and AB.

The above given problem can be sub-categorised as a problem of inverse kinematics. Here the path coordinates and the velocity of the end effector is given. But, the angles θ_1 and θ_2 are required in order to place the welding electrode in position for the entire operation. The problem has been solved in the following given ways.

1.1.1. MATLAB-

In order to write the code in matlab, a set of kinematic equations are required for the task. Position Analysis and Velocity Analysis is performed to get the required equations.

- **Position Analysis-**

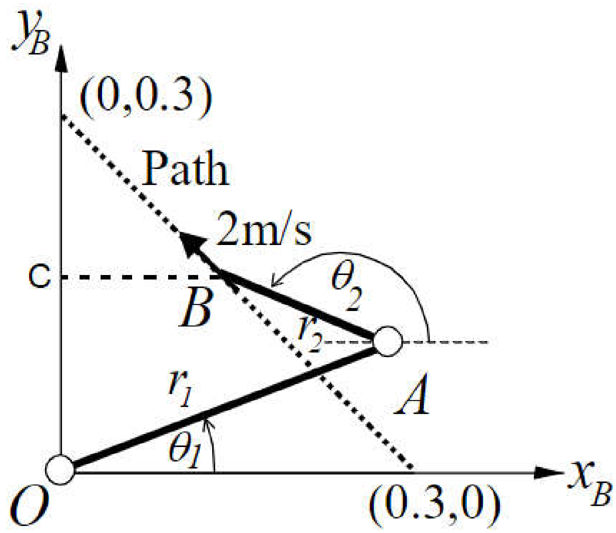

Figure 1.2

In reference to the above figure 1.2, the loop OABC should be considered in order to initiate the analysis. According to the loop the position equations are,

$r_1 \cos\theta_1 + r_2 \cos\theta_2 - x_B = 0$ ……… (i)

$r_1 \sin\theta_1 + r_2 \sin\theta_2 - y_B = 0$ ………(ii)

Equation (i) ⇒ $r_1 \cos\theta_1 = x_B - r_2 \cos\theta_2$ …….(iii)
Equation (ii) ⇒ $r_1 \sin\theta_1 = y_B - r_2 \sin\theta_2$ ……(iv)

Squaring and Adding of both sides of equations (iii) and (iv) can be written in the form of:

$A\cos\theta_2 + B\sin\theta_2 = C$

Where,

A = $2r_2x_B$, B = $2r_2y_B$ and C = $r_2^2 + X_B^2 + y_B^2 - r_1^2$

Let R = $\sqrt{A^2 + B^2}$

$\theta_2^{\pm} = ATAN2(B, A) + ATAN2(\sqrt{1 - \left(\frac{C}{R}\right)^2}, (C/R))$ ……….(v)

$\theta_1^{\pm}$ = ATAN2[($y_B - r_2 \sin\theta_2$), ($x_B - r_2 \cos\theta_2$)] ……….(vi)

Equations (v) and (vi) are the required equations for finding out the required angles θ_1 and θ_2. Equation (v) will have two solutions for θ_2. Hence the angle with the shortest path will only be accepted for the further velocity analysis.

- **Velocity Analysis-**

The above equations (i) and (ii) are required for initiating the analysis.

$r_1 \cos\theta_1 + r_2 \cos\theta_2 - x_B = 0$ ……… (i)

$r_1 \sin\theta_1 + r_2 \sin\theta_2 - y_B = 0$ ………(ii)

In order to get the velocity, the above equations should be differentiated w.r.t time,

$-r_1\dot{\theta}_1 \sin\theta_1 - r_2\dot{\theta}_2 \sin\theta_2 - \dot{x}_{\boldsymbol{B}} = 0$

$r_1\dot{\theta}_1 \cos\theta_1 + r_2\dot{\theta}_2 \cos\theta_2 - \dot{y}_{\boldsymbol{B}} = 0$

Eliminating $\dot{\theta}_2$ and putting $\dot{x}_{\boldsymbol{B}} = u_{\boldsymbol{B}}$ and $\dot{y}_{\boldsymbol{B}} = v_{\boldsymbol{B}}$

$\dot{\theta}_1 = \frac{u_{\boldsymbol{B}} \cos\theta_2 + v_{\boldsymbol{B}} \sin\theta_2}{r_1 \sin(\theta_2 - \theta_1)}$ ……………..(iii)

Eliminating $\dot{\theta}_1$,

$\dot{\theta}_2 = \frac{u_{\boldsymbol{B}} \cos\theta_1 + v_{\boldsymbol{B}} \sin\theta_1}{r_2 \sin(\theta_1 - \theta_2)}$ ………………(iv)

The above equations (iii) and (iv) are required to find out the angular velocities. The end effector is moving with a constant velocity of 2 m/sec. Hence, the values of $u_{\boldsymbol{B}}$ and $v_{\boldsymbol{B}}$ will the x and y component of the given constant velocity respectively.

In matlab this particular task can be done in two methods. Particularly both the methods are given as follows,

- **For loop method:**

```
for xb=0.3:-0.006:0              % step for every deg till 360
    yb=-xb+0.3;
    A=2*r2*xb;
    B=2*r2*yb;
    C=(r2)^2-(r1)^2+(xb)^2+(yb)^2;
    R=sqrt(A^2+B^2);
    Z=C/R;
    zz=sqrt(1-Z^2);
    i=i+1; % added a counter
    %Position Analysis
    theta2=atan2(B,A)+atan2(zz,Z);
    if theta2<0; theta2=theta2+2*pi;end   % done for shortest path
    theta2b=atan2(B,A)-atan2(zz,Z);
    if theta2b<0; theta2b=theta2b+2*pi;end
    theta1=atan2((yb-r2*sin(theta2)),(xb-r2*cos(theta2)));
    Xb(i)=xb; Yb(i)=yb; T1(i)=theta1; T2(i)=theta2; T2b(i)=theta2b; %T2 has two solutiuons
    %Velocity analysis
    ub=2*cos(2.3562);
    vb=2*sin(2.3562);
    w1=(ub*cos(theta2)+vb*sin(theta2))/(r1*sin(theta2-theta1));
    w2=(ub*cos(theta1)+vb*sin(theta1))/(r2*sin(theta1-theta2));
    W2(i)=w2; W1(i)=w1;
    point(i)=i;
end
```

Figure 1.3

The above figure 1.3 represents the screenshot of the main implemented code in for loop method. The equations from the above documented analysis are declared in a for loop. The for loop starts with $x_B = 0.3$ and ending at 0 as the first given point of welding is (0.3, 0). The resolution is kept as -0.006 in order to complete the task in 50 steps. The required graphs are discussed under the Results and Discussion section of this report.

- **Element by element method:**

```
%%
xb=[0.3:-0.006:0]
yb=-xb+0.3;

A=2*r2.*xb;
B=2*r2.*yb;
C=(r2)^2-(r1)^2+(xb).^2+(yb).^2;

R=sqrt(A.^2+B.^2);
Z=C./R;
zz=sqrt(1-Z.^2);

%Position Analysis
theta2=atan2(B,A)+atan2(zz,Z);
if theta2<0; theta2=theta2+2*pi;end   % done for shortest path
theta2b=atan2(B,A)-atan2(zz,Z);
if theta2b<0; theta2b=theta2b+2*pi;end
theta1=atan2((yb-r2*sin(theta2)),(xb-r2*cos(theta2)));

%Velocity analysis
ub=2*cos(2.3562);
vb=2*sin(2.3562);
w1=(ub*cos(theta2)+vb*sin(theta2))./(r1*sin(theta2-theta1));
w2=(ub*cos(theta1)+vb*sin(theta1))./(r2*sin(theta1-theta2));
```

Figure 1.4

The above figure 1.4 represents the screenshot of the main implemented code in element by element method. A range of values is set for x_B whose starting value is 0.3 and ending is 0. Here the resolution is also kept as -0.006 in order to complete the task in 50 steps. The required graphs are discussed under the Results and Discussion section of this report.

1.1.2. SIEMENS NX-

The similar two bar robot manipulator task has been implemented on Siemens NX. The task is completed in several stages. They are as follows:

- **Part Modelling-** This forms the first stage of the task. This is the most important stage as proper dimensions are needed to be used while constructing the parts. Two different lengthy link parts named as link OA and link AB, a cylindrical base and an end effector is required to build the mechanism. Techniques like sketching and extrusion are required to bring proper shape to the parts.

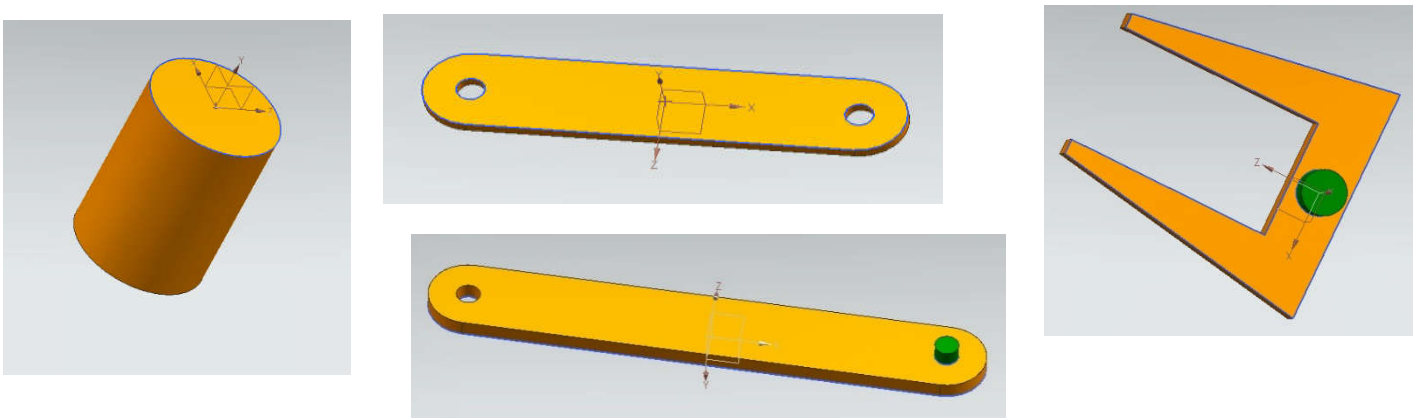

Figure 1.5

The above figure 1.5 shows all the models of parts required for the construction of the two bar robot manipulator.

- **Assembly-** This forms the second stage of the task where all the parts are assembled perfectly at their joints. Using Siemens NX, the parts are snapped into the workspace and are mated with concentric constraints. There are a total of 4 concentric constraints made. Then the assembled part is carried on to the motion section of the software. The joints are recognised and made automatically at this stage. Hence there are a total of four revolute joints formed. A fifth joint of type slider is created to enable the end effector to move. A vector is also set for the movement of the end effector.

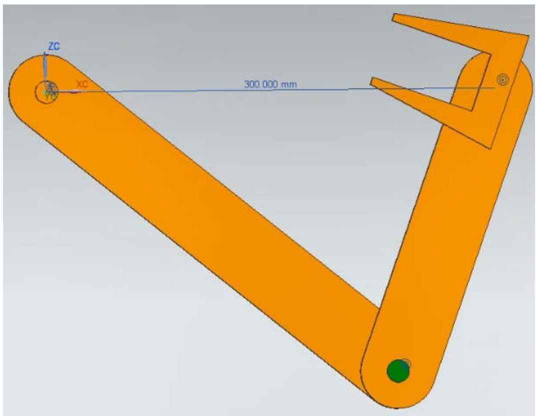

Figure 1.6

The above figure 1.6 represents the assembled model of the two bar robot manipulator in Siemens NX. The end effector is kept exactly 0.3m away from the base part, in order to mimic the starting position of the welding task.

- **Simulation-** At this stage the starting and ending points of the desired path is declared. The velocity of the slider joint is set to 2m/sec. The time of the simulation is set to 0.212 sec. This can be easily calculated as the distance of the path and the constant velocity is known. The number of steps of the operation is set to 50.

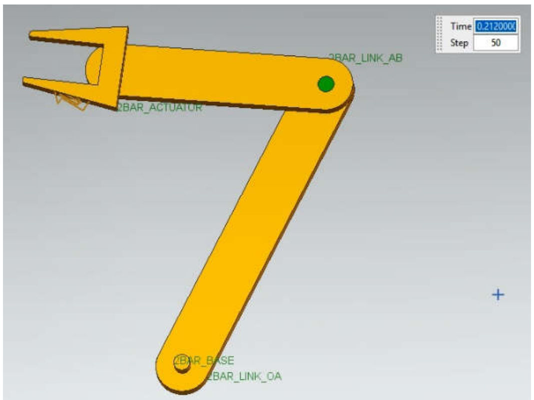

Figure 1.7

The above figure 1.7 represents the final position of the robotic arm in action. The simulation is solved, and the required results are plotted. One of the required graphs has been plotted using a simple Matlab code. Further discussion on the required graphs are made in the Results and Discussion section of the report.

1.2. FOUR BAR MECHANISM: -

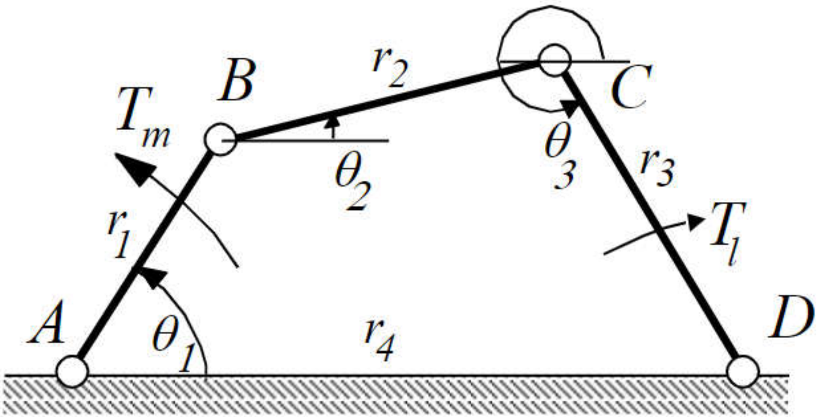

Figure 1.8

The above given figure 1.8 consists a schematic diagram of a four-bar mechanism. This mechanism has three links bolted on a ground where the ground is considered as the fourth link. As per the problem, crank AB is driven by a motor with a torque T_m, at an angular velocity of 100 rad/s and the follower CD carries load torque T_l. All the values of required lengths, masses of links and moment of inertia of the links are given. The ultimate task is to perform kinematics and dynamics analysis in order to plot the graph of variation of motor torque T_m and polar load diagrams for all the joints respectively.

The above given problem can be sub-categorised as a problem of inverse dynamics. Here all the physical properties of the mechanism are given and the torque and the forces are being expected to solved. The problem has been solved in the following given ways.

1.2.1. MATLAB-

In order to write the code in matlab and get the required results the steps of inverse kinematics should be implemented first. It involves the position analysis, then velocity analysis and ending with acceleration analysis.

- **Kinematics:**

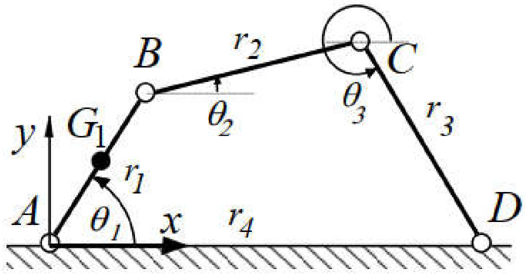

Figure 1.9

In reference to the above given figure 1.9, consider the link AB. Hence, the required kinematics equations are:

$x_{G1} = r_{G1}\, cos\, \theta_1$
$y_{G1} = r_{G1}\, sin\, \theta_1$

Differentiating the above equation w.r.t time to get the velocity equations

$\dot{x}_{G1} = -r_{G1}\dot{\theta}_1\, sin\, \theta_1$
$\dot{y}_{G1} = r_{G1}\dot{\theta}_1\, cos\, \theta_1$

Differentiating the above equation w.r.t time to get the acceleration equations

$\ddot{x}_{G1} = -r_{G1}(\ddot{\theta}_1 \sin\theta_1 + \dot{\theta}_1^2 \cos\theta_1)$ …………..(i)
$\ddot{y}_{G1} = r_{G1}(\ddot{\theta}_1 \cos\theta_1 + \dot{\theta}_1^2 \sin\theta_1)$ ……………(ii)

These are the required acceleration equations.

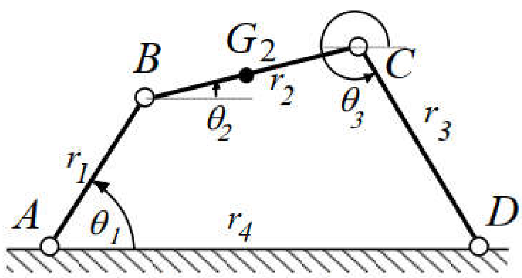

Figure 1.10

In reference to the above given figure 1.10, consider the link BC. Hence, the required kinematics equations are:

$x_{G2} = r_1\, cos\, \theta_1 + r_{G2}\, cos\, \theta_2$

$y_{G2} = r_1 \sin\theta_1 + r_{G2} \sin\theta_2$

Differentiating the above equation w.r.t time to get the velocity equations

$\dot{x}_{G2} = -r_1\dot{\theta}_1 \sin\theta_1 - r_{G2}\dot{\theta}_2 \sin\theta_2$
$\dot{y}_{G2} = r_1\dot{\theta}_1 \cos\theta_1 + r_{G2}\dot{\theta}_2 \cos\theta_2$

Differentiating the above equation w.r.t time to get the acceleration equations

$\ddot{x}_{G2} = -r_1(\ddot{\theta}_1 \sin\theta_1 + \dot{\theta}_1^2 \cos\theta_1) - r_{G2}(\ddot{\theta}_2 \sin\theta_2 + \dot{\theta}_2^2 \cos\theta_2)$ …………..(iii)
$\ddot{y}_{G2} = r_1(\ddot{\theta}_1 \cos\theta_1 - \dot{\theta}_1^2 \sin\theta_1) + r_{G2}(\ddot{\theta}_2 \cos\theta_2 - \dot{\theta}_2^2 \sin\theta_2)$ ………..(iv)

These are the required acceleration equations.

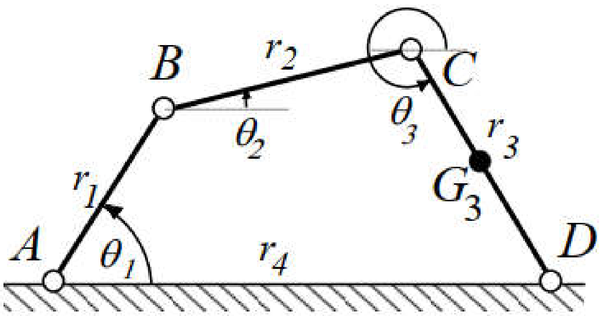

Figure 1.11

In reference to the above given figure 1.11, consider the link CD. Hence, the required kinematics equations are:

$x_{G3} = r_4 - (r_3 - r_{G3}) \cos\theta_3$
$y_{G3} = -(r_3 - r_{G3}) \sin\theta_3$

Differentiating the above equation w.r.t time to get the velocity equations

$\dot{x}_{G3} = (r_3 - r_{G3})\dot{\theta}_3 \sin\theta_3$
$\dot{y}_{G3} = -(r_3 - r_{G3})\dot{\theta}_3 \cos\theta_3$

Differentiating the above equation w.r.t time to get the acceleration equations

$\ddot{x}_{G3} = (r_3 - r_{G3})(\ddot{\theta}_3 \sin\theta_3 + \dot{\theta}_3^2 \cos\theta_3)$ …………..(v)
$\ddot{y}_{G3} = -(r_3 - r_{G3})(\ddot{\theta}_3 \cos\theta_3 - \dot{\theta}_3^2 \sin\theta_3)$ ………….(vi)

These are the required acceleration equations.

- **Dynamics:**

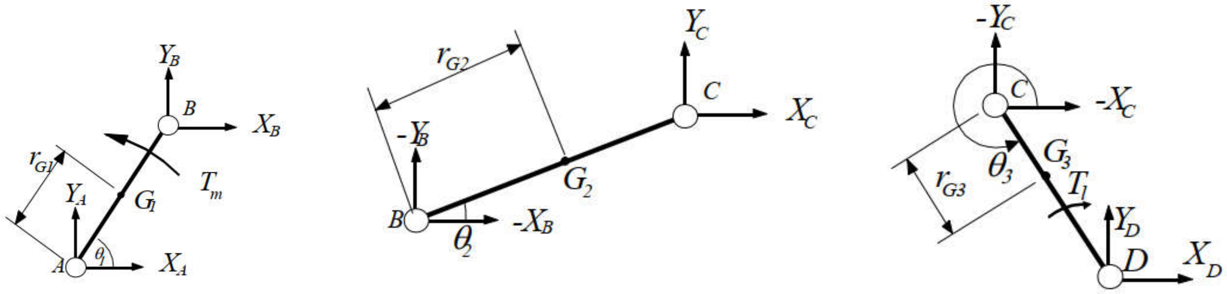

Figure 1.12

In reference to the above given figure 1.12, the required equations of dynamics are:

$X_A + X_B = m_1\ddot{x}_{G1}$ ……(1)
$Y_A + Y_B = m_1\ddot{y}_{G1}$ ……..(2)
$T_m +\ X_A r_{G1} \sin\theta_1 - Y_A r_{G1}\cos\theta_1 - X_B(r_1 - r_{G1})\sin\theta_1 + Y_B(r_1 - r_{G1})\cos\theta_1 =\ I_{G1}\ddot{\theta}_1$ ..(3)

$-X_B + X_C = m_2\ddot{x}_{G2}$ ……(4)
$-Y_B + Y_C = m_2\ddot{y}_{G2}$ ……..(5)
$-X_B r_{G2}\sin\theta_2 + Y_B r_{G2}\cos\theta_2 - X_C(r_2 - r_{G2})\sin\theta_2 + Y_C(r_2 - r_{G2})\cos\theta_2 =\ I_{G2}\ddot{\theta}_2$ ..(6)

$-X_C + X_D = m_3\ddot{x}_{G3}$ ……(7)
$-Y_C + Y_D = m_3\ddot{y}_{G3}$ ……..(8)

$-T_l - X_C r_{G3}\sin\theta_3 + Y_C r_{G3}\cos\theta_3 - X_D(r_3 - r_{G3})\sin\theta_3 + Y_D(r_3 - r_{G3})\cos\theta_3 =$
$I_{G3}\ddot{\theta}_3$…(9)

In the above written equations, $X_A, X_B, X_C, X_D, Y_A, Y_B, Y_C$ and Y_D are the forces at X and Y directions for the joints A,B,C and D respectively. T_m and T_l stands for motor torque and load torque respectively. T_l value is assumed to be zero.

To find out the required forces and motor torque the above dynamics equations can be written in matrices format. This method is only done after plugging in the acceleration due to the centre of mass equations from the kinematics analysis part, to equations (1),(2),(4),(5),(7) and (8) respectively.

[**A**][**X**] = [**B**]

The above-mentioned matrix multiplication is followed where:

- [**A**]- It is a matrix of resolution 9x9. It is formed by taking in the coefficients of the LHS of the above (1)-(9) equations.
- [**B**]- It is a matrix of resolution 9x1. It is formed by taking in the coefficients of the RHS of the above (1)-(9) equations.
- [**X**]- It is a matrix of resolution 9x1. It contains nine row elements, $X_A, X_B, X_C, X_D, Y_A, Y_B, Y_C, Y_D$ *and* T_m.

$[\mathbf{X}] = [\mathbf{A}]^{-1}[\mathbf{B}]$

The above matrix operation is performed in order to get the required forces and the motor torque.

```
%% Variable declaration
tic
r1=0.01; r2=0.035; r3=0.02; r4=0.03;  %in meters
w1=100; a1=0; i=0;   % rad/s
% For dynamics
rg1=0.005; rg2=0.0175; rg3=0.01;
m1=0.2; m2=0.6; m3=0.4;
Ig1=1 * 10^(-5); Ig2=4 * 10^(-4); Ig3=8 * 10^(-5);
T_L = 0;
%  Variation of THETA1 from 0 to 360

for TH1=0:0.0001:2*pi             % step for every deg till 360
    A=cos(TH1)-(r4/r1);
    B=sin(TH1);
    C=(r4/r3)*cos(TH1)-((r1^2-r2^2+r3^2+r4^2)/(2*r1*r3));
    R=sqrt(A^2+B^2);
    Z=C/R;
    zz=sqrt(1-Z^2);
    i=i+1; % added a counter
    %Position Analysis
    TH3=atan2(B,A)+atan2(zz,Z);
      if TH3<0; TH3=TH3+2*pi;end   % done for shortest path
      TH3b=atan2(B,A)-atan2(zz,Z);
      if TH3b<0; TH3b=TH3b+2*pi;end
    TH2=atan2((-r3*sin(TH3)-r1*sin(TH1)),(r4-r3*cos(TH3)-r1*cos(TH1)));
      T1(i)=TH1; T2(i)=TH2; T3(i)=TH3; T3b(i)=TH3b; %T3 has two solutiuons
    %Velocity analysis
    w3=w1*(r1*sin(TH1-TH2))/(r3*sin(TH2-TH3));
    w2=w1*(r1*sin(TH3-TH1))/(r2*sin(TH2-TH3));
    W3(i)=w3; W2(i)=w2;

    %Acceleration Analysis
    a3=(w1^2*r1/r3)*((1-(w2/w1))*cos(TH1-TH2)*sin(TH2-TH3)-((w2/w1)-(w3/w1))*cos(TH2-TH3)*sin(TH1-TH2))/(sin(TH2-TH3))^2;
    a2=-(w1^2*r1/r2)*((1-(w3/w1))*cos(TH1-TH3)*sin(TH2-TH3)-((w2/w1)-(w3/w1))*cos(TH2-TH3)*sin(TH1-TH3))/(sin(TH2-TH3))^2;
```

Figure 1.13

The above figure 1.13 represents the screenshot of the main implemented code in matlab. The equations from the above documented analysis are declared in a for loop. The for loop starts with $\theta_1 = 0$ to $\theta_1 = 2\pi$. Since, the calculations are done only for one full rotation of the crank AB. The resolution is kept as 0.0001 in order to complete the task. The required graphs are discussed under the Results and Discussion section of this report.

1.2.2. SIEMENS NX-

The similar four bar mechanism task has been implemented on Siemens NX. The task is completed in several stages. They are as follows:

- **Part Modelling-** This forms the first stage of the task. Again, this is the most important stage as proper dimensions are needed to be used while constructing the parts. Four different lengthy link parts are required to build the mechanism. Techniques like sketching and extrusion are required to bring proper shape to the parts.

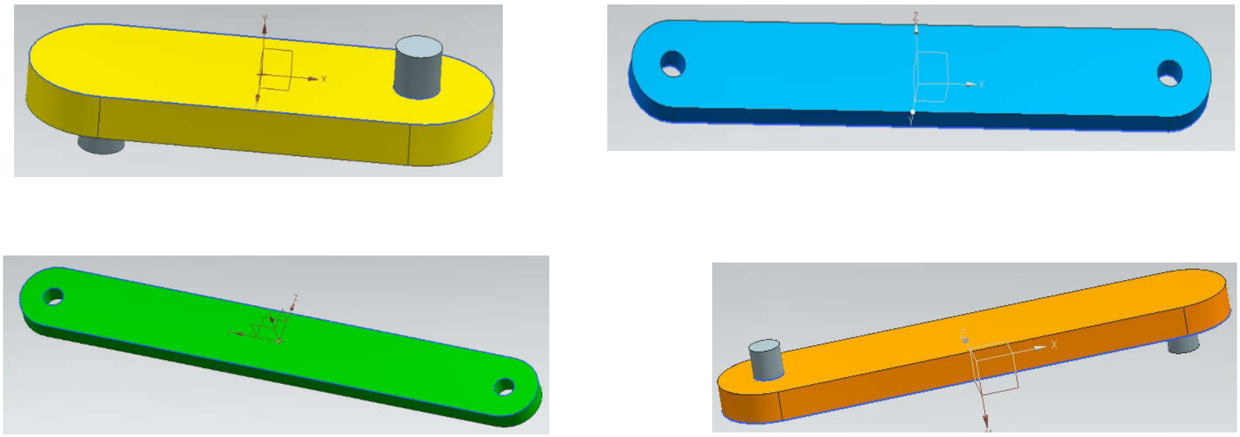

Figure 1.14

The above figure 1.14 shows all the models of parts required for the construction of the four bar mechanism.

- **Assembly-** In the second stage of the task all the parts are assembled perfectly at their mating points. Using Siemens NX assembly environment, the parts are snapped into the workspace and are mated with concentric constraints. There are a total of 4 concentric constraints made. Then the assembled part is carried on to the motion section of the software. Here the joints are made manually at this stage. The joints are created at the obvious mating parts of the mechanism. Hence there are a total of four revolute joints formed. A fifth joint of type fixed is created to make the bottom link fixed to the ground. A driver is set at the first joint A. The four bar mechanism has a 1 DOF and so it requires one motor to drive the mechanism. The driver at joint A acts as the motor.

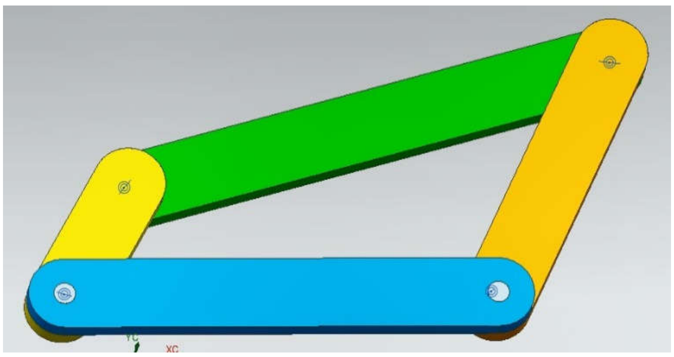

Figure 1.15

The above figure 1.15 represents the assembled model of the four bar mechanism in Siemens NX.
The joints are made manually, and the blue coloured link is assigned as the ground.

- **Simulation-** At this stage the angular velocity of the driver at joint A is set to 100 rad/sec. The time of the simulation is set to 0.063 sec. This can be easily calculated as the constant angular velocity is already known. The number of steps of the operation is set to 100.

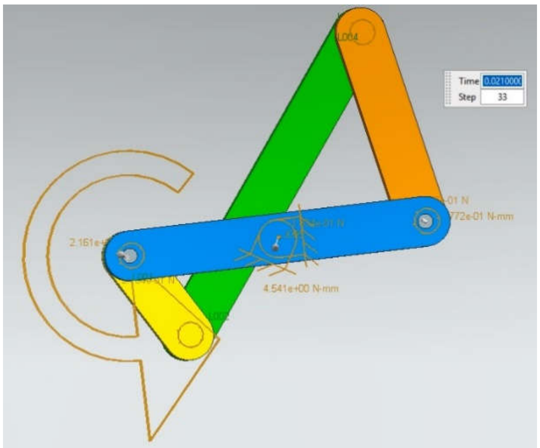

Figure 1.16

The above figure represents an instance from the simulation of the four bar mechanism in Siemens NX. The simulation is solved, and the required results are plotted. Further discussion on the required graphs are made in the Results and Discussion section of the report.

2. RESULTS AND DISCUSSION: -

2.1. TWO BAR ROBOTIC ARM: -

Two types of graphs were plot under the heading of two bar robotic manipulator system. The two types of graphs are given as follows.

- **Variation in Angles:**

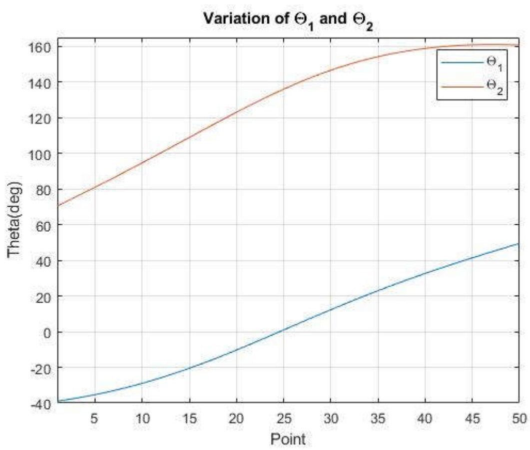

Figure 2.1

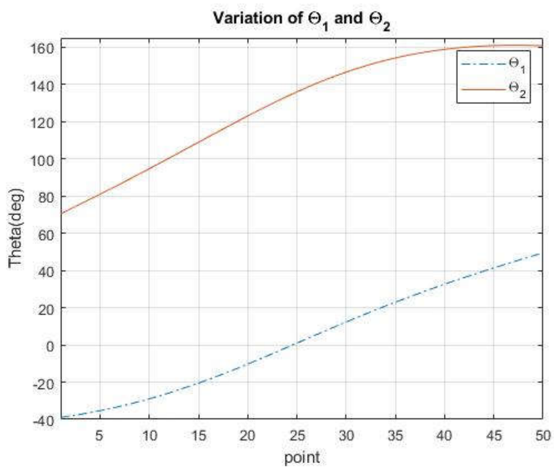

Figure 2.2

The above given figures 2.1 and 2.2 represents the variation of angles $\theta_1 and\ \theta_2$ by applying the for-loop method and the element by element method respectively. It is known that $\theta_1$and θ_2 are the angles made by link OA and AB respectively. Both the graphs have exact similar plots. But the second mentioned method took less execution time.If visualized correctly, the angle made by link OA starts off with an angle value lower than the other angle. The difference between both the angles at the initial stage is almost maintained till the last stage of the welding path. In easy words, the difference between the angles almost remained constant. Both the angle curve has same slope. Such slope is a result as the path of the welding operation is a straight path. Also, the second method is much suitable for many complex structures.

- **Variation in Angular Velocities:**

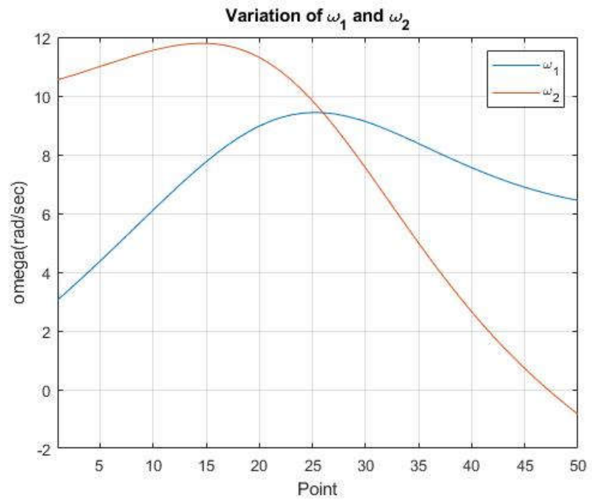

Figure 2.3

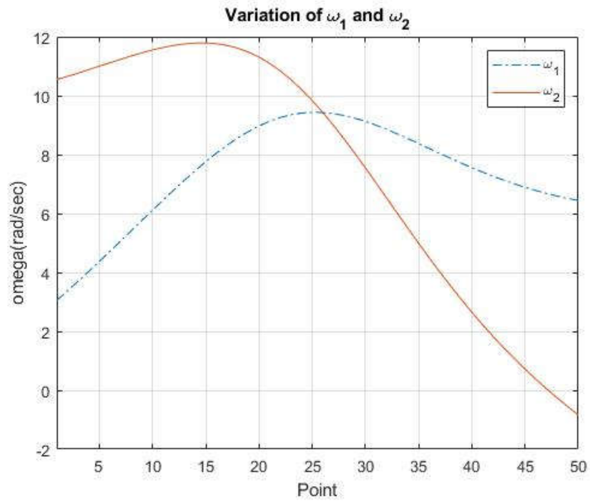

Figure 2.4

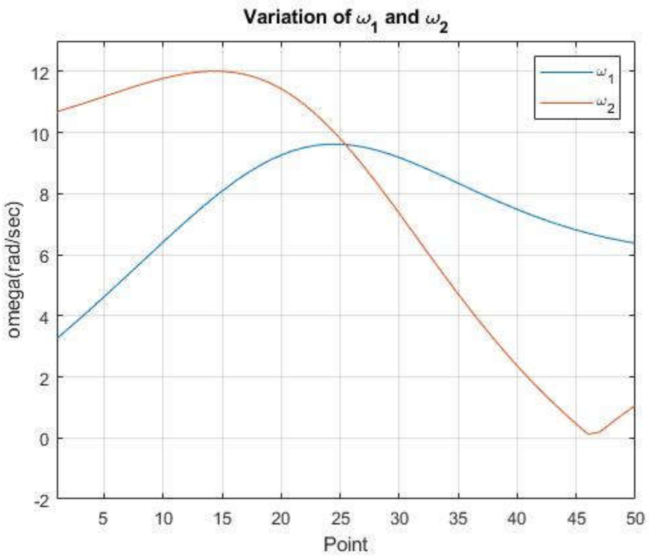

Figure 2.5

The above given figures 2.3, 2.4 and 2.5 represents the variation of angles $\omega_1 and\ \omega_2$ by applying the for-loop method, the element by element method and the NX simulation method respectively. For each of the graphs, the value of ω_2 achieves the maximum angular at 3/10th part of the entire path and then drastically decreases creating a steep slope. Whereas the value of ω_1 constantly decreases just after completing halfway through the journey. Both the angular velocities record the same value at 50% of the journey. The $\omega_1$does not have a steep slope. The ω_2 value is quite unstable this is because this angular velocity pertains to link AB, which is towards the open end of the mechanism. In any mechanism, the open ends are quite uncontrollable but don't have restricted movements as compared to the other links.

The graphs in figure 1 and 2 are almost same. Again, the graph made by the element by element is fetched in a much less time. The graph generated from the Siemens NX has a slightly different ending. This is generally obtained due to noise created during the simulation.

2.2. FOUR BAR MECHANISM: -

Five graphs were plot under the heading of four bar mechanism. The five types of graphs are given as follows.

- **Motor Torque Curve**

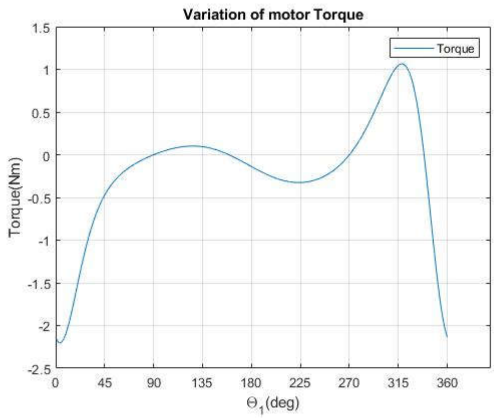

Figure 2.6

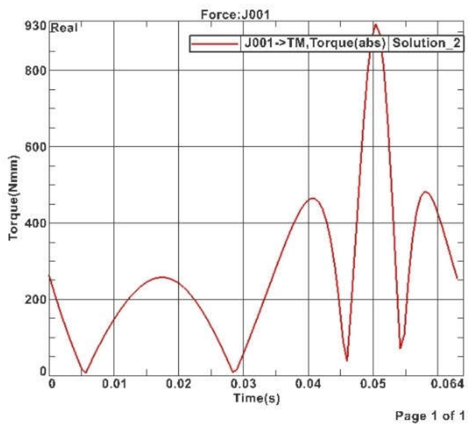

Figure 2.7

The above given figures a and b represents the variation of motor torque in four bar mechanism. The figure a is an output from the matlab code, while the later one is a result from the Siemens NX simulation. The angular velocity of the driver link is set to a constant value. This creates a large variation in motor torque.

- **Polar Load Diagram of all the Joints**

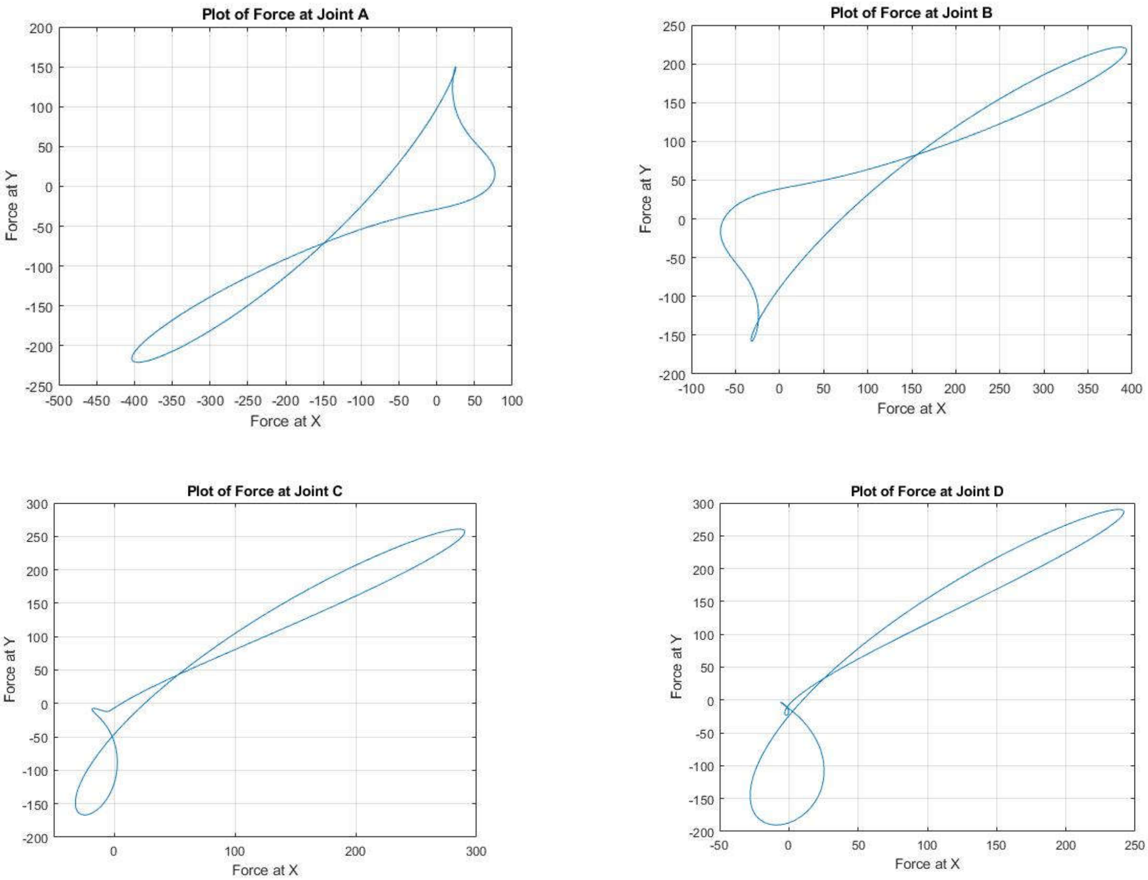

Figure 2.8

The above four graphs in figure 2.8 represents the polar diagram forces acting at all the four joints respectively. All the graphs are generated from the matlab code. Polar forces diagrams give a wonderful analysis of the magnitude of the forces bear by the joints. By the above given graphs, it can be commented that the joints A and B had to face a large amount of force in terms of magnitude. This is because the plots for joint A and B are quite spread out than the other two joints. Also, the area of the figures formed for the graph of joint A and B are much larger than the other joints. No type of noise is experienced in the graphs.

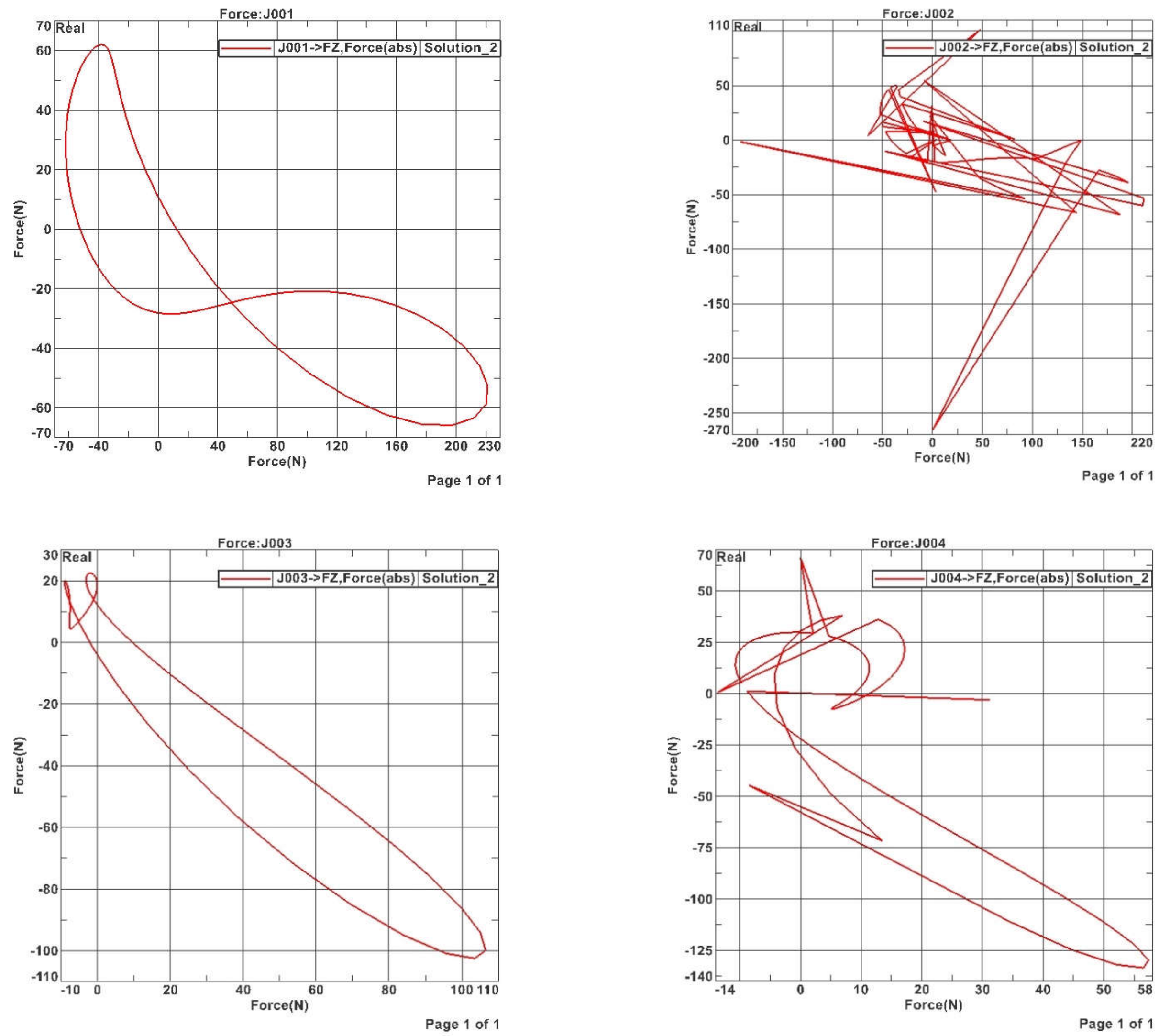

Figure 2.9

The above four graphs in figure 2.9 represents the polar diagram forces acting at all the four joints respectively. All the graphs are generated from the Siemens NX simulation. With respect to the results generated by the matlab method, these plots for polar force diagrams are not too satisfactory. The graphs involve a lot of noise as the simulation might not have proceeded smoothly.

Hence, this was the entire analysis of the two bar and the four bar mechanisms. Both tasks were an example under inverse kinematics and dynamics. In day to day life, majority number of engineering industries, deals with problems related to inverse kinematics and dynamics. A lot of characteristics about different types of mechanisms were able to be studied. Such type is analysis is really required in industries before actuating operations related to such fields.

RFERENCES: -

Motion simulation of a two-link manipulator using Siemens NX 10 SECTION I -Problem statement. (n.d.). Mechatronics Design and Automation Class Notes [Accessed 1 May 2020].

Shakouri, D.P. (n.d.). *Analysis of a four NX8-Simulation Motion*. Kingston University. Mechatronic Design and Automation Class Notes [Accessed 1 May 2020].

Zweiri, Y. (2019). *Mechatronic Design and Automation Lecture Notes* [Accessed 1 May 2020].

APPENDICES: -

```
% analysis of task 1 using for loop method

clear
clc

tic
%% Variable declaration

r1=0.3; r2=0.2;  %in meters
i=0;

%%  Solution for the analysis

for xb=0.3:-0.006:0              % for loop for xb from 0.3 to 0
    yb=-xb+0.3;
    A=2*r2*xb;
    B=2*r2*yb;
    C=(r2)^2-(r1)^2+(xb)^2+(yb)^2;
    R=sqrt(A^2+B^2);
    Z=C/R;
    zz=sqrt(1-Z^2);
    i=i+1; % added a counter
    %Position Analysis
    theta2=atan2(B,A)+atan2(zz,Z);
    if theta2<0; theta2=theta2+2*pi;end
    theta2b=atan2(B,A)-atan2(zz,Z);
    if theta2b<0; theta2b=theta2b+2*pi;end
    theta1=atan2((yb-r2*sin(theta2)),(xb-r2*cos(theta2)));
    Xb(i)=xb; Yb(i)=yb; T1(i)=theta1; T2(i)=theta2; T2b(i)=theta2b; %T2 has two
    %Velocity analysis
    ub=2*cos(2.3562); %initial angle at the begining of the path is 135 degrees
    vb=2*sin(2.3562);
    w1=(ub*cos(theta2)+vb*sin(theta2))/(r1*sin(theta2-theta1));
    w2=(ub*cos(theta1)+vb*sin(theta1))/(r2*sin(theta1-theta2));
    W2(i)=w2; W1(i)=w1;
    point(i)=i;
end
```

```
%% PLOT Required Results for variation in angles

figure(1)
plot((point),(T1*180/pi),(point),(T2*180/pi))
grid
title('Variation of \Theta_1 and \Theta_2')
xlabel('Point')
ylabel('Theta(deg)')
legend('\Theta_1','\Theta_2')
xlim([1 50]);
ylim([-40 165]);


%% PLOT Required Results for variation in angular velocities

figure(2)
plot((point),(W1),(point),(W2))
grid
title('Variation of \omega_1 and \omega_2')
xlabel('Point')
ylabel('omega(rad/sec)')
legend('\omega_1','\omega_2')
xlim([1 50]);

toc
```

```
% analysis of task 1 using element by element method
clear
clc
tic
%% Variable declaration

r1=0.3; r2=0.2;  %in meters
i=1:51;


%%  Solution for the analysis

xb=[0.3:-0.006:0]      %Assigning the starting and end positions
yb=-xb+0.3;

A=2*r2.*xb;
B=2*r2.*yb;
C=(r2)^2-(r1)^2+(xb).^2+(yb).^2;

R=sqrt(A.^2+B.^2);
Z=C./R;
zz=sqrt(1-Z.^2);

%Position Analysis
theta2=atan2(B,A)+atan2(zz,Z);
if theta2<0; theta2=theta2+2*pi;end   % done for shortest path
theta2b=atan2(B,A)-atan2(zz,Z);
if theta2b<0; theta2b=theta2b+2*pi;end
theta1=atan2((yb-r2*sin(theta2)),(xb-r2*cos(theta2)));

%Velocity analysis
ub=2*cos(2.3562);
vb=2*sin(2.3562);
w1=(ub*cos(theta2)+vb*sin(theta2))./(r1*sin(theta2-theta1));
w2=(ub*cos(theta1)+vb*sin(theta1))./(r2*sin(theta1-theta2));

%% PLOT Required Results for variation in angles
figure(1)
plot(i,(theta1*180/pi),'-.'), grid
```

```
plot(i,(theta1*180/pi),'-.'), grid
hold on
plot(i,(theta2*180/pi))
xlabel('point')
ylabel('Theta(deg)')
legend('\Theta_1','\Theta_2')
title('Variation of \Theta_1 and \Theta_2')
xlim([1 50]);
ylim([-40 165]);

%% PLOT Required Results for variation in angular velocities

figure(2)
plot(i,(w1),'-.'), grid
hold on
plot(i,(w2))
xlabel('Point')
ylabel('omega(rad/sec)')
title('Variation of \omega_1 and \omega_2')
legend('\omega_1','\omega_2')
xlim([1 50]);

toc
```

```
% analysis of task 2 using for loop method
%%
clear
clc
tic
%% Variable declaration

r1=0.01; r2=0.035; r3=0.02; r4=0.03;  %length of links in meters
w1=100; a1=0; i=0;   % w1 is the angular velocity of 1st link
% For dynamics
rg1=0.005; rg2=0.0175; rg3=0.01;
m1=0.2; m2=0.6; m3=0.4;   % mass of links
Ig1=1 * 10^(-5); Ig2=4 * 10^(-4); Ig3=8 * 10^(-5);  % Moment of Inertia
T_L = 0;    % Load Torque
%%  Solution for the analysis

for TH1=0:0.0001:2*pi             % initiating for loop
    A=cos(TH1)-(r4/r1);
    B=sin(TH1);
    C=(r4/r3)*cos(TH1)-((r1^2-r2^2+r3^2+r4^2)/(2*r1*r3));
    R=sqrt(A^2+B^2);
    Z=C/R;
    zz=sqrt(1-Z^2);
    i=i+1; % adding a counter

    %Position Analysis
    TH3=atan2(B,A)+atan2(zz,Z);
      if TH3<0; TH3=TH3+2*pi;end
      TH3b=atan2(B,A)-atan2(zz,Z);
      if TH3b<0; TH3b=TH3b+2*pi;end
    TH2=atan2((-r3*sin(TH3)-r1*sin(TH1)),(r4-r3*cos(TH3)-r1*cos(TH1)));
      T1(i)=TH1; T2(i)=TH2; T3(i)=TH3; T3b(i)=TH3b;              %T3 has two s

    %Velocity analysis
    w3=w1*(r1*sin(TH1-TH2))/(r3*sin(TH2-TH3));
    w2=w1*(r1*sin(TH3-TH1))/(r2*sin(TH2-TH3));
    W3(i)=w3; W2(i)=w2;

    %Acceleration Analysis
```

```matlab
%Acceleration Analysis
a3=(w1^2*r1/r3)*((1-(w2/w1))*cos(TH1-TH2)*sin(TH2-TH3)-((w2/w1)-(w3/w1))*cos(TH2-TH3)*sin(TH1-TH2))/(sin(TH2-TH3))^2;
a2=-(w1^2*r1/r2)*((1-(w3/w1))*cos(TH1-TH3)*sin(TH2-TH3)-((w2/w1)-(w3/w1))*cos(TH2-TH3)*sin(TH1-TH3))/(sin(TH2-TH3))^2;
A3(i)=a3; A2(i)=a2;

%Acceleration due to centre of mass
acc_xg1= -(rg1 * ( (a1*sin(TH1)) + ((w1)^2 * cos(TH1)) ) );
acc_yg1= rg1 * ( (a1*cos(TH1)) + ((w1)^2 * sin(TH1)) );

acc_xg2= -( r1 * ((a1*sin(TH1)) + ((w1)^2 * cos(TH1))) + rg2 * ((a2*sin(TH2)) + ((w2)^2 * cos(TH2))) );
acc_yg2= (r1 * ((a1*cos(TH1)) - ((w1)^2 * sin(TH1)))) + (rg2 * ((a2*cos(TH2)) - ((w2)^2 * sin(TH2))));

acc_xg3= (r3-rg3) * ((a3*sin(TH3)) + ((w3)^2*cos(TH3)));
acc_yg3= -(r3-rg3) * ((a3*cos(TH3)) - ((w3)^2*sin(TH3)));

%B elements
B1 = m1*acc_xg1;
B2 = m1*acc_yg1;
B3 = Ig1*a1;
B4 = m2*acc_xg2;
B5 = m2*acc_yg2;
B6 = Ig2*a2;
B7 = m3*acc_xg3;
B8 = m3*acc_yg3;
B9 = (Ig3*a3) + T_L;
%B matrix
B = [B1; B2; B3; B4; B5; B6; B7; B8; B9];

%Preparing the A matrix and its inverse

A = [1 0 1 0 0 0 0 0 0; 0 1 0 1 0 0 0 0 0; rg1*sin(TH1) -rg1*cos(TH1) (r1-rg1)*sin(TH1) (r1-rg1)*cos(TH1) 0 0 0 0 1; 0 0 -1 0 1 0 0 0 (
inverse_A = inv(A);

%Finding X matrix

X = inverse_A * B;

%Recording the Answers
```

```
    %Recording the Answers
    T(i)=X(9,:);
    Joint_AX(i)=X(1,:);
    Joint_AY(i)=X(2,:);
    Joint_BX(i)=X(3,:);
    Joint_BY(i)=X(4,:);
    Joint_CX(i)=X(5,:);
    Joint_CY(i)=X(6,:);
    Joint_DX(i)=X(7,:);
    Joint_DY(i)=X(8,:);

end

%% PLOT Results

%%Plot of Theta2 and Theta3 vs Theta1
figure(1)
plot((T1*180/pi),(T2*180/pi),(T1*180/pi),(T3*180/pi))
grid
title('Variation of \Theta_2 and \Theta_3')
xlabel('\Theta_1(deg)')
ylabel('\Theta(deg)')
legend('\Theta_2','\Theta_3')
xlim([0 360]);

%Plot of OMEGA2 and OMEGA3 against THETA1
figure(2)
plot((T1*180/pi),W2,(T1*180/pi),W3)
grid
title('Variation of angular velocities')
xlabel('\Theta_1(deg)')
ylabel('\omega(rad/s)')
legend('\omega_2','\omega_3')
xlim([0 360]);

%Plot of alpha3 and alpha2 against Theta1
figure(3)
```

```
figure(3)
plot((T1*180/pi),A2,(T1*180/pi),A3)
grid
title('Variation of angular acc.')
xlabel('\Theta_1(deg)')
ylabel('\alpha(rad/s^2)')
legend('\alpha_2','\alpha_3')
xlim([0 360]);
%% Dynamics graphs
%Plot of motor torque against Theta1
figure(4)
plot((T1*180/pi),T)
grid
title('Variation of motor Torque')
xlabel('\Theta_1(deg)')
ylabel('Torque(Nm)')
legend('Torque')
xticks(0:45:360);

%Plot of Force in joint A
figure(5)
plot(Joint_AX,Joint_AY)
grid
title('Plot of Force at Joint A')
xlabel('Force at X')
ylabel('Force at Y')
xticks(-500:50:100);

%Plot of Force in joint B
figure(6)
plot(Joint_BX,Joint_BY)
grid
title('Plot of Force at Joint B')
xlabel('Force at X')
ylabel('Force at Y')
xticks(-100:50:450);
```

```
xticks(-100:50:450);

%Plot of Force in joint C
figure(7)
plot(Joint_CX,Joint_CY)
grid
title('Plot of Force at Joint C')
xlabel('Force at X')
ylabel('Force at Y')
xticks(-200:100:500);

%Plot of Force in joint D
figure(8)
plot(Joint_DX,Joint_DY)
grid
title('Plot of Force at Joint D')
xlabel('Force at X')
ylabel('Force at Y')
xticks(-100:50:500);

toc
```

```
%This set of code plots the variation in angular velocities in task 1 of data
%The simulation data is loaded from the spreadsheet.
%%
clear
clc
%%
T = readtable('twobar_solution_2_velocity.xlsx');

Step = T{:,1};
Link_AB = T{:,2};
Liink_OA = T{:,3};
AVelocity_Link_AB = T{:,2}*(pi/180);
AVelocity_Link_OA = T{:,3}*(pi/180);

%% PLOT Results velocities

figure(1)
plot((Step),(AVelocity_Link_OA),(Step),(AVelocity_Link_AB))
grid
title('Variation of \omega_1 and \omega_2')
xlabel('Point')
ylabel('omega(rad/sec)')
legend('\omega_1','\omega_2')
xlim([1 50]);
ylim([-2 13]);
```